CSE334: Software Engineering

Major task report

RedPlus

Mustafa Usama Abdelrahmn	20p5556
Rawan Ahmad Muhammed	20p5251
Salma Mohamed Ahmed Elsoly	20p6375
Ahmed Ibrahim Eldera	20p2701
Kareem Moataz Naeem	20p4361
Ahmed Nezar Ahmed	20p4326
Yassin Khaled Abd Elsamie	20p2668
Mohamed Amin Mohamed	20p4699

Contents

Software Requirement Specification (SRS)

Introduction

A web based blood bank management system. Patients and medical expert have to register through entering their information, they can donate and request blood after logging in to their accounts. While the request process they must enter the blood type they want to receive. Medical expert have to give his approval according to blood types of the donators and receivers, as well as requesting and donating. A notification is send to the receivers when his request is approved. Medical expert can transfer the non-matching donors and receivers to the facility where they can donate and receive blood.

Feasibility analysis

The feasibility study is performed to determine whether the proposed system is applicable according to technical, operational and economical factor.

Technical feasibility:

The proposed system is developed using React/Redux, HTML/CSS, and Javascript as a front-end tool, Node.js as a backend, and MongoDB as the database. The proposed system needs a server equipped with linux based OS to serve the requests submitted by the users. The Web browser is used to view the web page that is available within the Windows operating system itself. The proposed system will run under any environment with a modern browser supporting AJAX. As Windows is very user friendly and GUI OS it is very easy to use. All the required hardware and software are readily available in the market. Hence the system is technically feasible.

Operational Feasibility:

The proposed system will help patients to get their required blood type easily, donors will communicate easily with near by requests. Patient will be served at his place. Based on this benefits, The proposed system is operationally feasible.

Economical factor:

The hardware and software required for this project are available in the market at low cost, the initial investment is the only cost incurred and does not need any further enhancements. Hence it is economically feasible.

System requirements

1. Functional requirements
    1. Patient and medical expert must register by entering their full name, email, phone number, password, diseases, location, role (user or expert) and blood type.
    2. User and expert should be able to login after registration by entering their phone number and password entered while registration.
    3. User must login to be able to donate and request blood as well as the medical expert.
    4. User and medical expert can request blood by entering full name, phone number and blood type (if he want to request for another patient in operation or surgery)
    5. User can donate to facility by selecting the facility name from the list as well as people by choosing one of the donating cards available to donate.
    6. Donating cards appear according to nearest location and blood type.

7. Donating cards take green color when it is eligible to donate while it take white card when it is not eligible to donate.
8. Donating card doesn't show any contact information for the default user about the receiver except blood type. Contact information appear only to the expert for privacy.
9. Medical expert approve pending blood transaction or transfer it to facility.
10. If their request still pending without donation eligible for it the medical expert has to transfer it to facility.

2. Non- functional requirements
   1. The system must be available 24/7 with no bandwidth issues.
   2. System should not register user unless creating strong password more than 4 characters.
   3. Runtime of page is less than one second.
   4. Extendable website in case there is new functional requirements need to be added.
   5. If issues happened in the system then it must be programmed in way that developer can serve it again.
   6. The system should be available for using in the whole city not a certain area.
   7. Implementation required for front- end is React framework with Redux library, back-end Node.js and database MongoDB (noSQL database).

Requirements validation

1. Source traceability matrix

	Patient	Facility	Medical expert	Developer
1	✓		✓	
2	✓		✓	
3	✓		✓	
4	✓		✓	
5	✓	✓		✓
6				✓
7				✓
8				✓
9	✓	✓	✓	
10	✓	✓	✓	
11	✓	✓	✓	✓
12				✓
13				✓
14				✓
15				✓
16				✓
17				✓

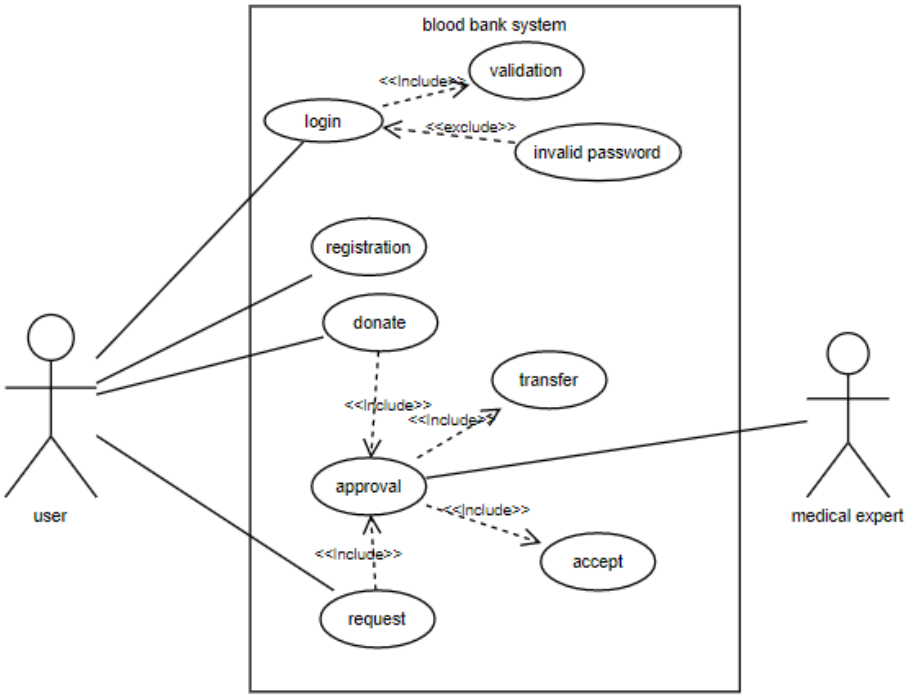
2. Requirement traceability matrix

	1	2	3	4	5	6	7	8	9	10
1		R								
2	D									
3	R	D		D	D					
4	R	R							D	D
5						D	D			
6				D	D					
7				D	D					
8				R					D	
9				D	D					
10				R	D					

Analysis and Design document

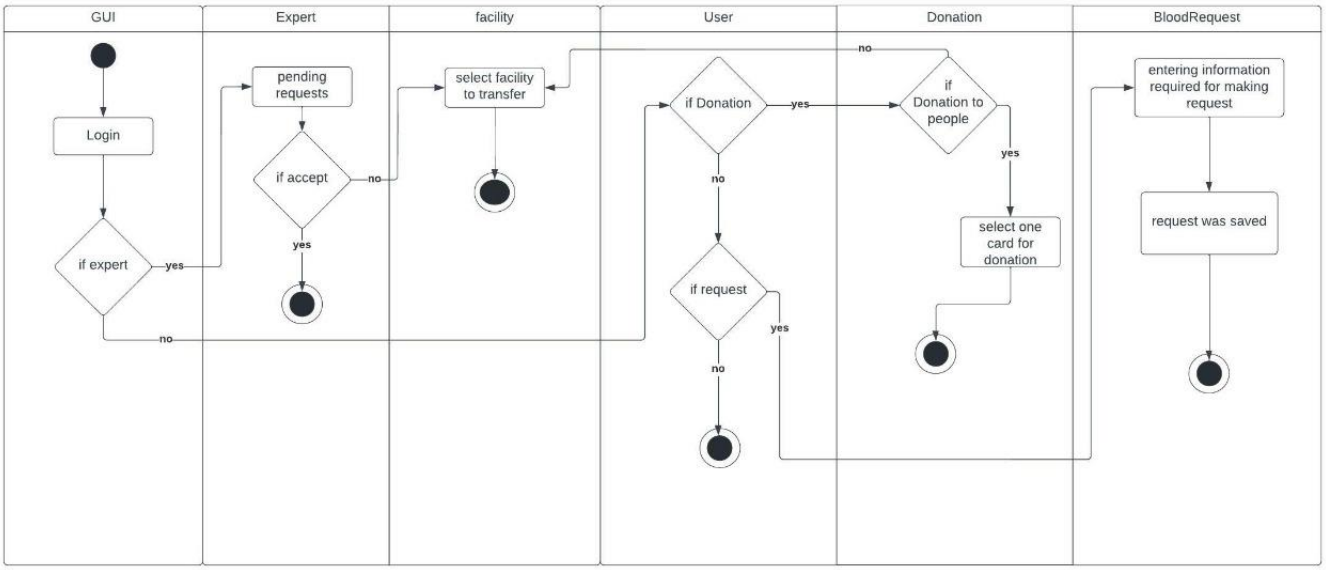
Use-Case Diagram and the Swimlane Diagram

1. Use - Case diagram

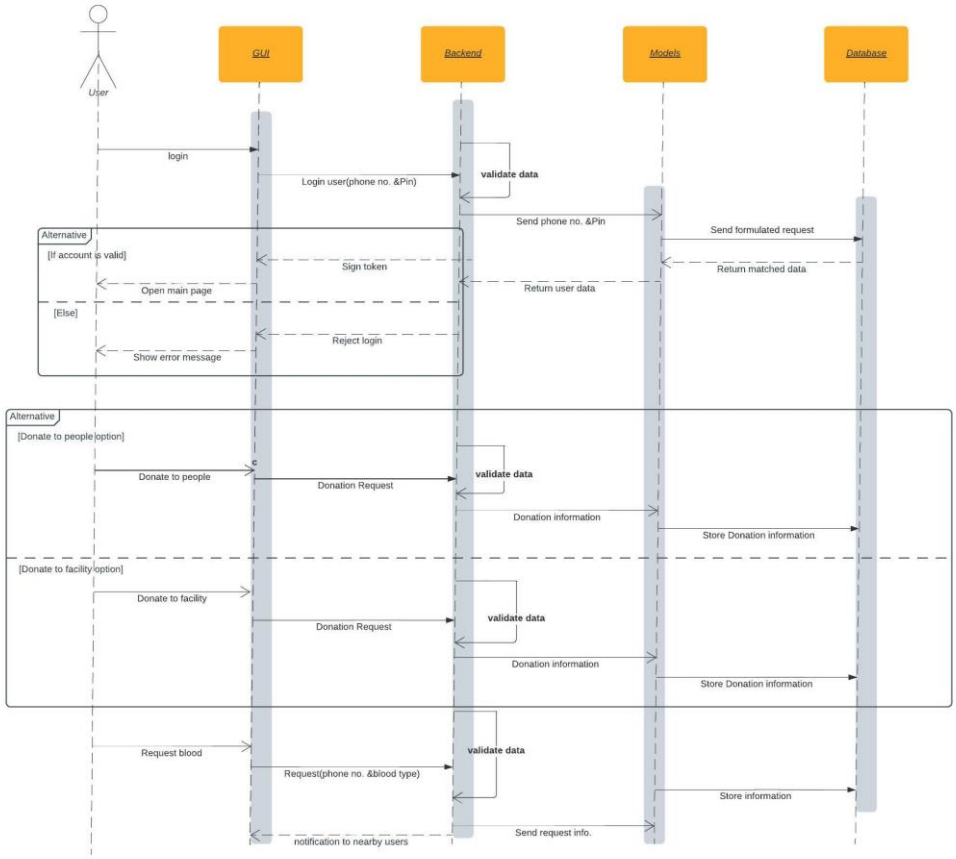


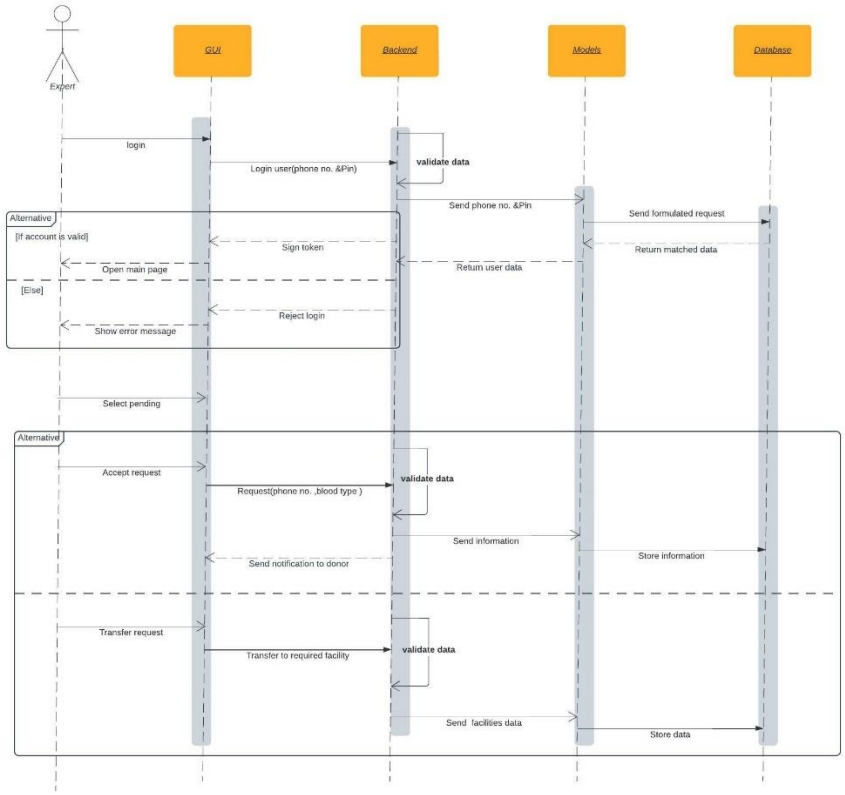
In this use-case diagram User is primary actor, he have to register at first the login to be able to request or donate. while medical expert in this case is secondary actor as he revise pending transactions, in the case of acceptance the transaction will disappear from pending and from the donating cards that appear while donating, in case of transfer the medical expert have to choose one of the listed facility and transfer the request to it.

2. Swimlane diagram



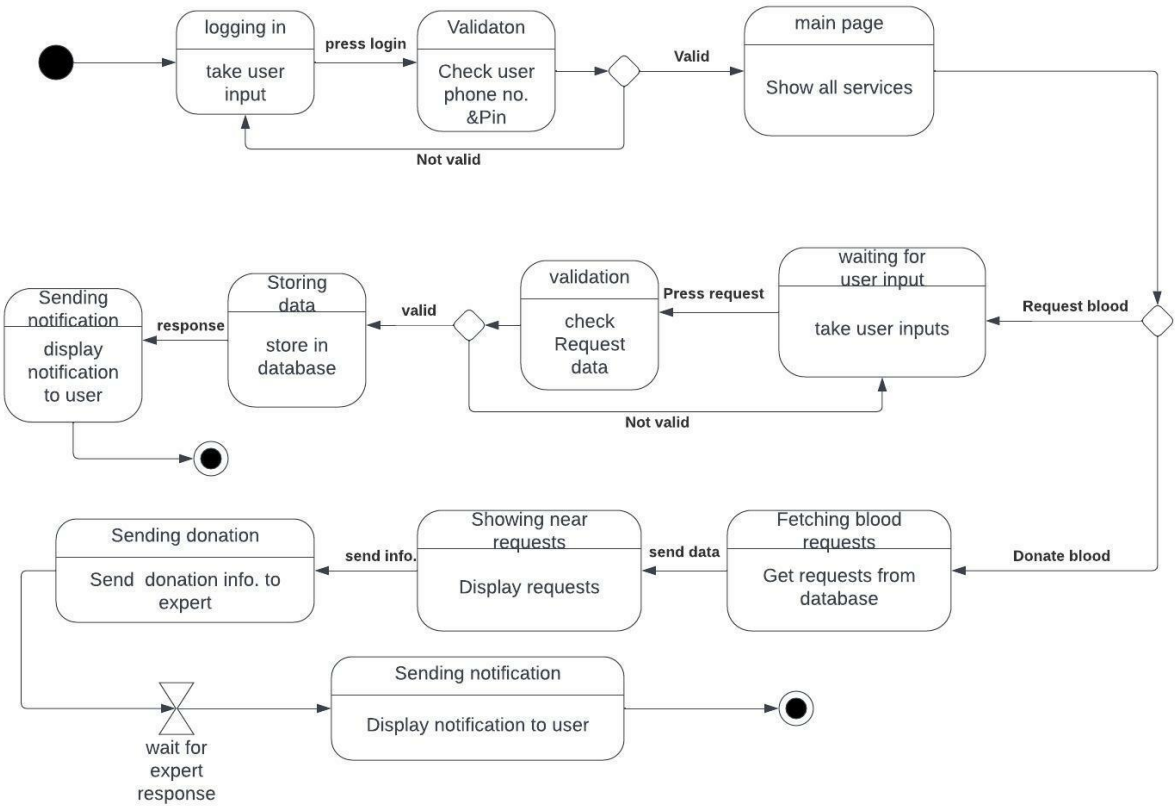
Interaction diagrams



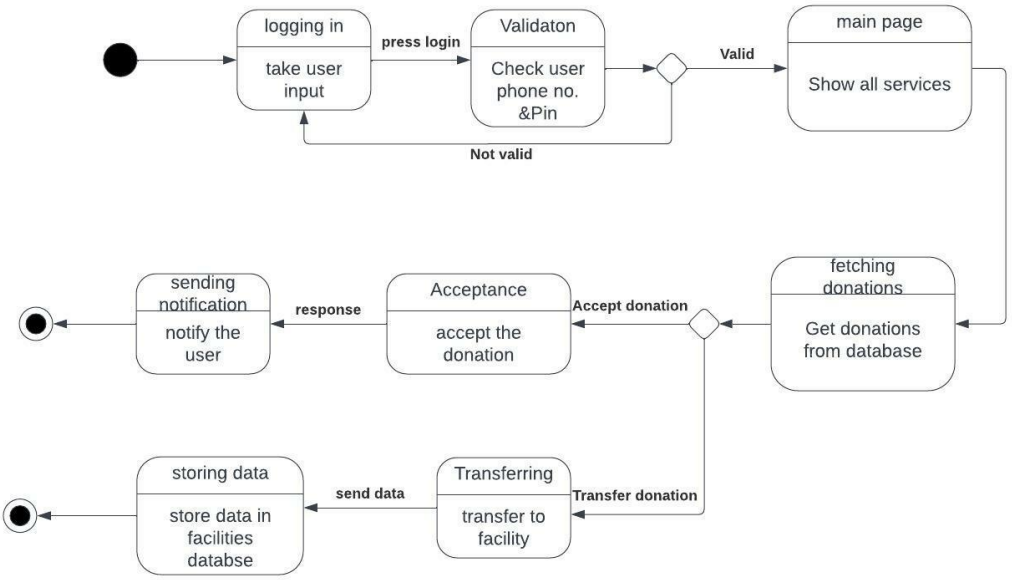


State diagram

1. User state diagram:

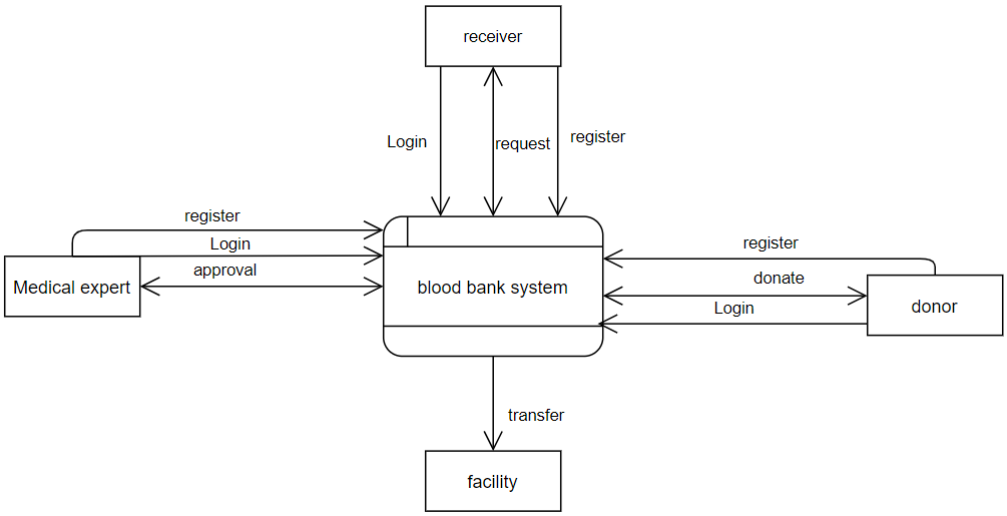


2. Expert state diagram:



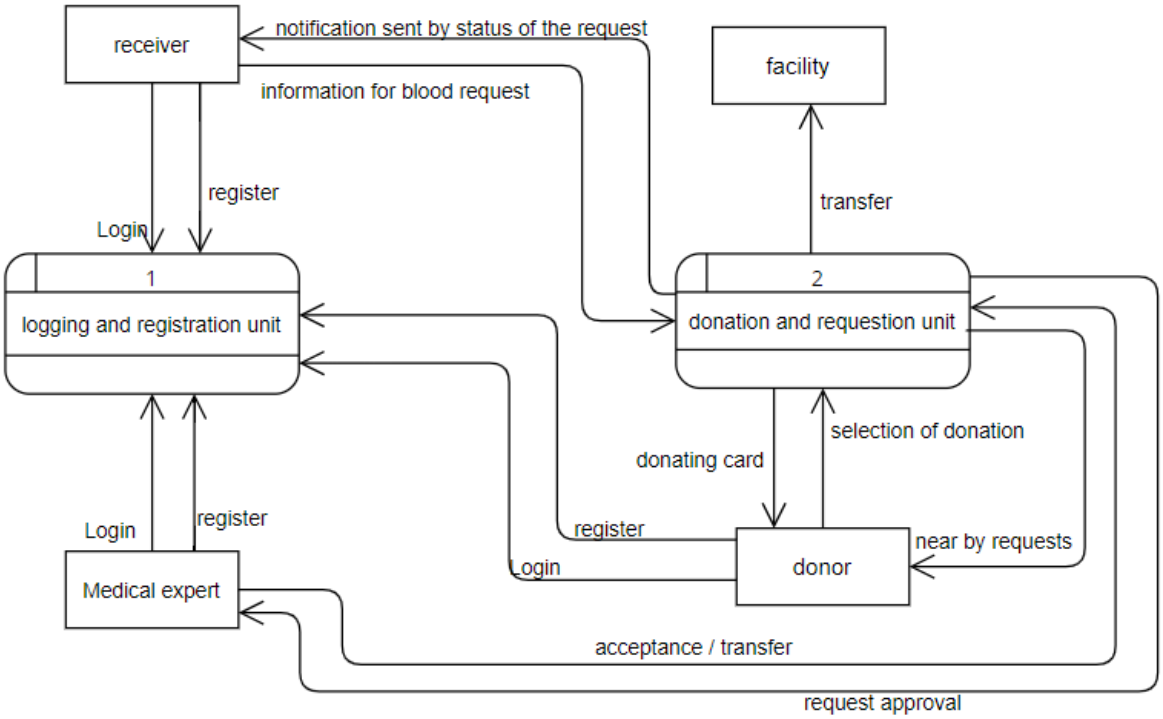
Data flow diagram

1. Context diagram (level 0)

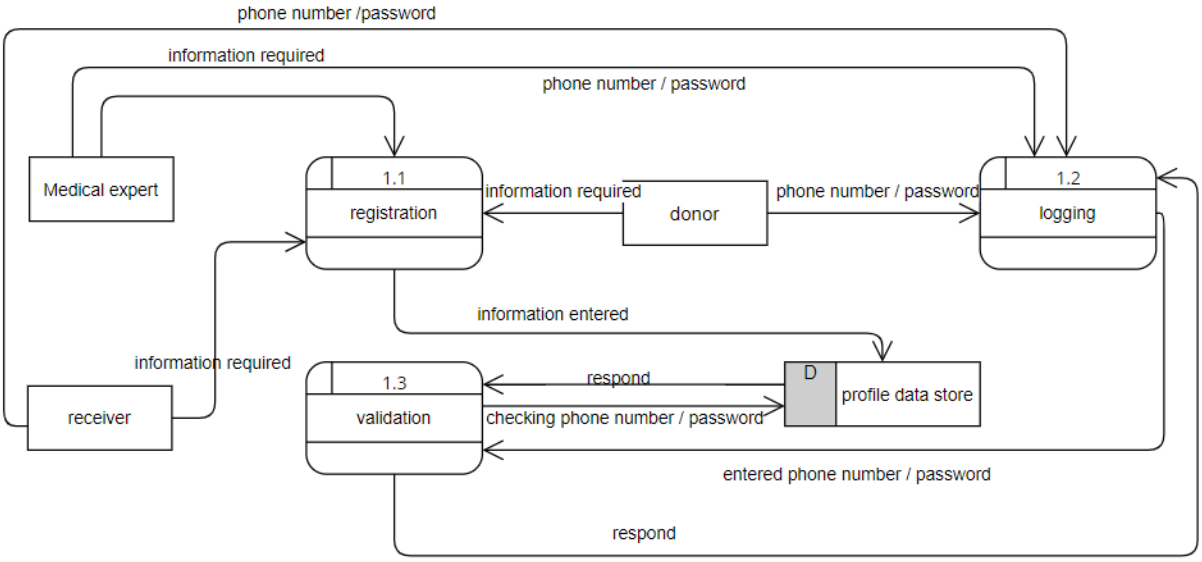


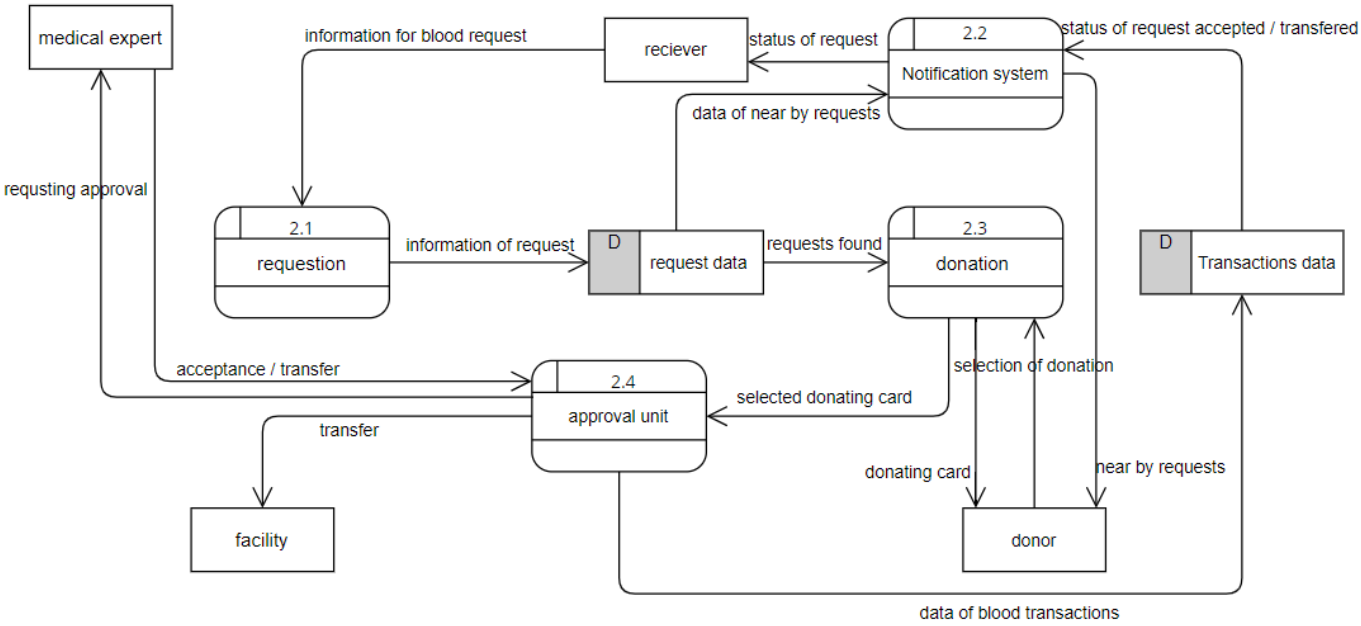
Assuming we have 3 entities Medical expert , user and facility. They have to register at first then login with information they used to register, so there is data flow from these 3 entities to the system. in case of donating and request there is exchangeable data flow between them and the system. the medical expert approve transactions between donors and receivers. facility receive from the system blood transaction which are not accepted by medical expert.

2. Level 1

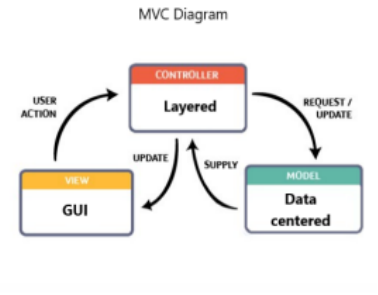


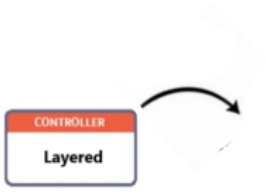
3. Level 2

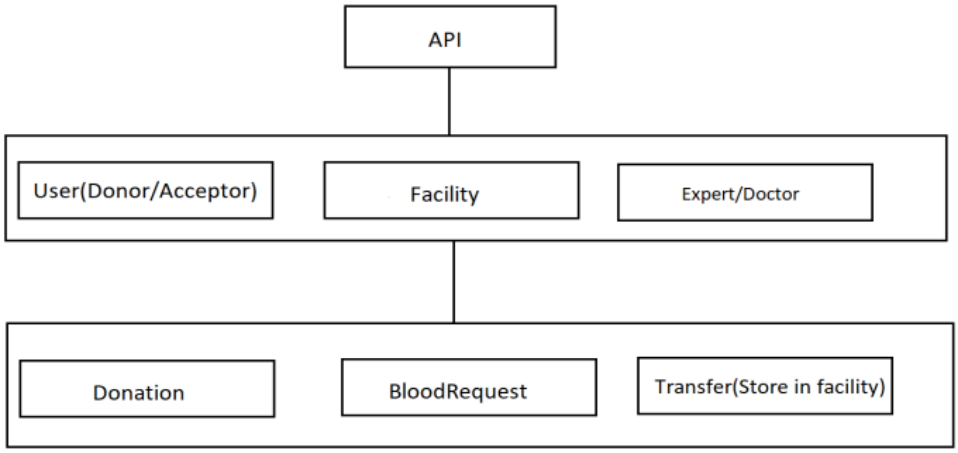

phone number /password
information required
phone number / password
Medical expert
1.1
registration
information required
donor
phone number / password
1.2
logging
information entered
information required
receiver
1.3
validation
respond
checking phone number / password
D
profile data store
entered phone number / password
respond


medical expert
information for blood request
reciever
status of request
2.2
Notification system
status of request accepted / transfered
data of near by requests
requsting approval
2.1
requestion
information of request
D
request data
requests found
2.3
donation
D
Transactions data
acceptance / transfer
2.4
approval unit
selected donating card
selection of donation
transfer
donating card
near by requests
facility
donor
data of blood transactions

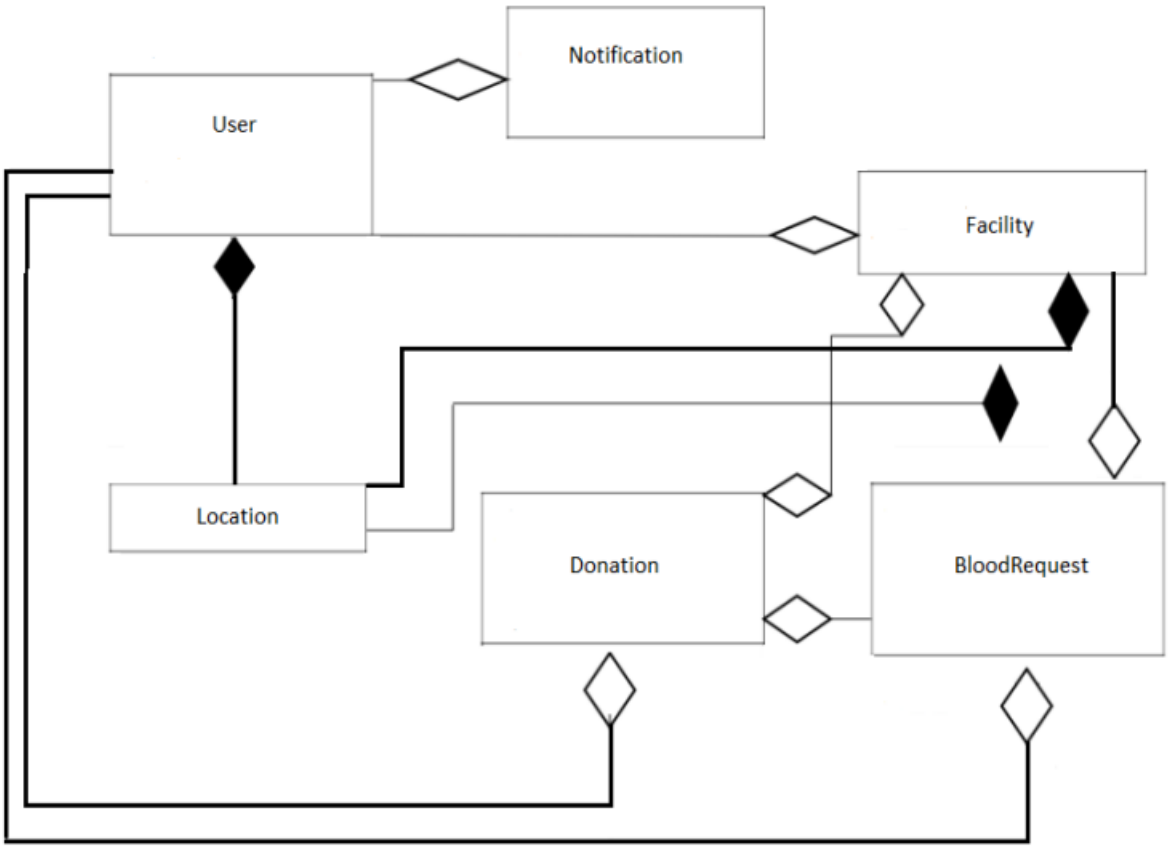
System architecture



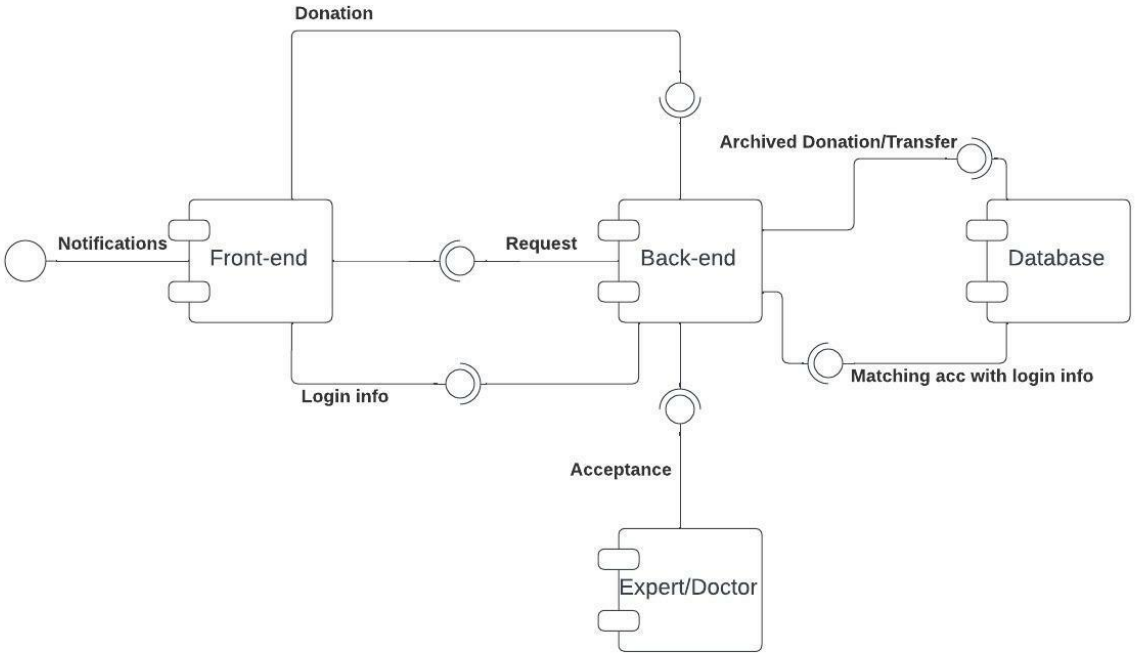




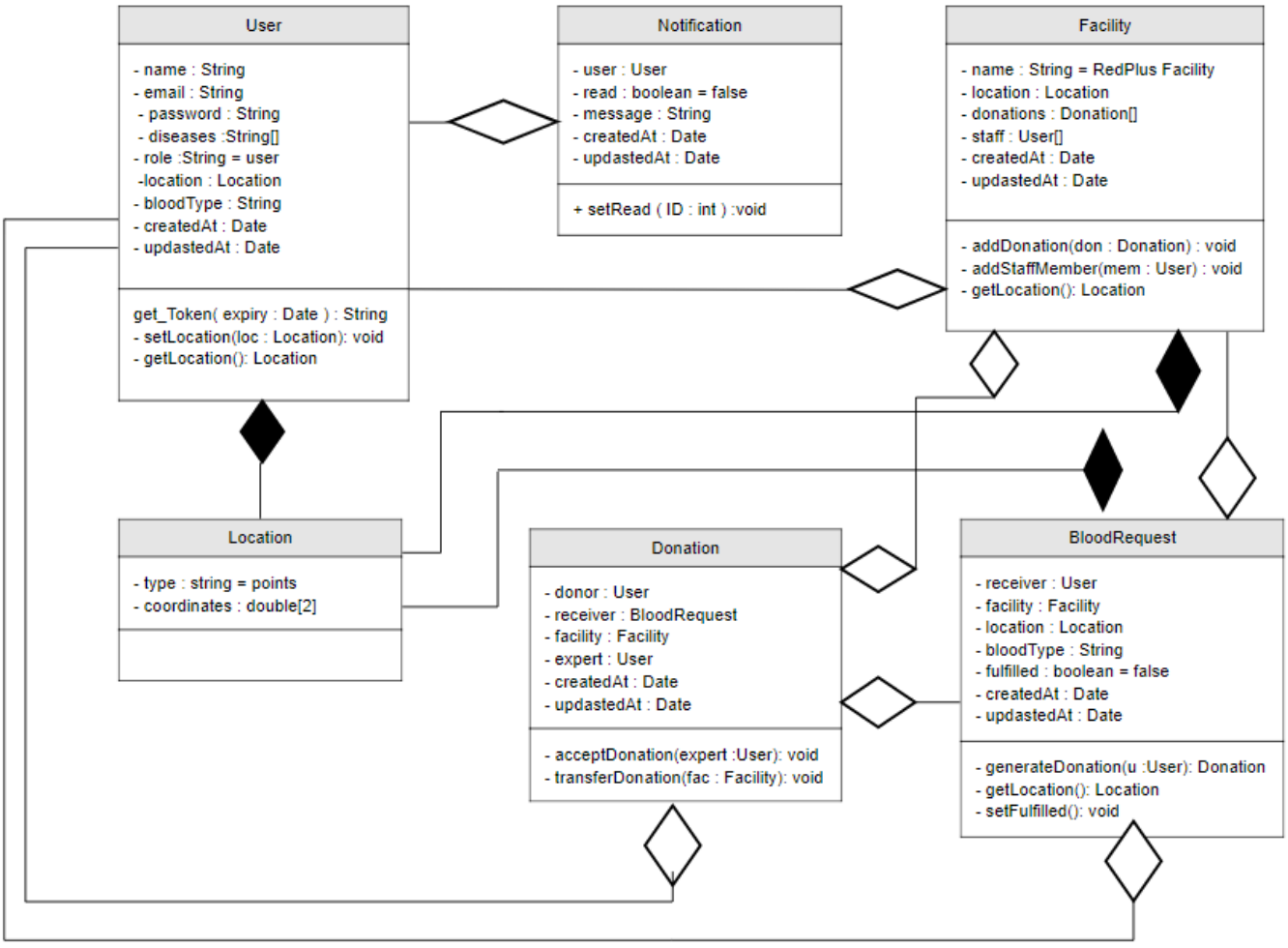

MODEL
Data
centered


Notification
User
Facility
Location
Donation
BloodRequest

Component diagram



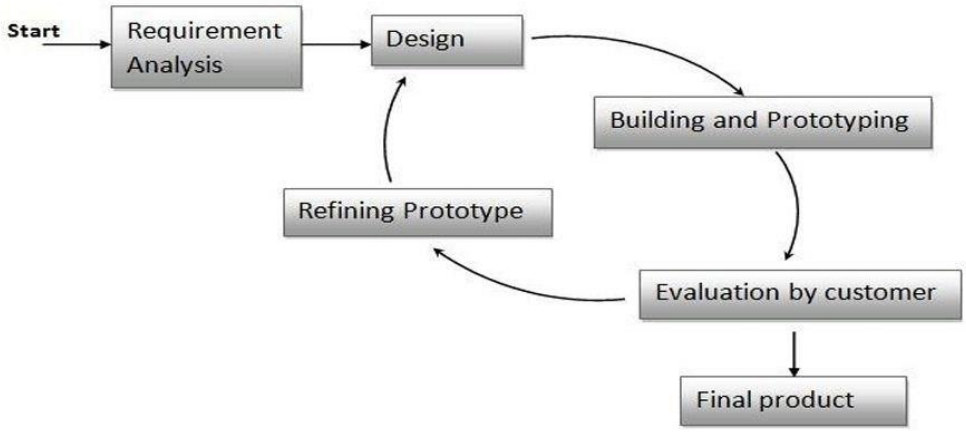
Class diagram



Software Development Methodology:

In choosing the development methodology of the product, there are various processes or methodologies that can be selected for the development of the research depending on the research's aims and goals. The models specify the various stages of the process and the order in which they are carried out. The researcher choose framework on prototyping model in developing the product. Figure 1 shows a prototyping model.

Water fall model



Justification Methodology Selection:

The researcher choose this type of prototyping because extreme prototyping is used in the web development domain. It also because of the desired system needs to have a lot of interaction with the end users and typically, online systems, web interfaces have a very high amount of interaction with end users. Since a working model of the system is displayed, the users get a better understanding of the system being developed. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user. Therefore, it can reduces time and cost as the defects can be detected much earlier and quicker user feedback is available leading to better solutions.

Prototyping Model:

Prototyping model consist of view phases, namely requirement analysis, design, building and prototyping, evaluation by customer, refining prototype, final product.

Requirement Analysis:

A prototyping model begins with requirements analysis and the requirements of the system are defined in detail. The research for development of this system is undertaken by the way of qualitative and quantitative approach in order to obtain the necessary data. The expert will be interview with regard to aspects of the blood donation process. Direct interview involved reaching out to people that is expert about blood donation process such as user from national blood center or hospital. This was in quest to find out the blood donation procedures and the existing of blood donation system. Interview question will be attach in appendix.

Design :

Once the analysis is complete, the step of designing takes over, which is basically building the architecture of the project. It is not a detailed design and includes only the important aspects of the system, which gives an idea of the system to the user. This step helps remove possible flaws by setting a standard and attempting to stick to it.

Building and Prototyping:

The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed. While, the workarounds are used to give the same look and feel to the customer in the prototype developed.

Evaluation by Customer:

The prototype developed is then presented to the customer and the other important stakeholders in the research. The feedback is collected in an organized manner and used for further enhancements in the product under development.

Refining Prototype:

The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like – time and budget constraints and technical feasibility of the actual implementation. The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until the customer expectations are met. Prototypes can have horizontal or vertical dimensions. A Horizontal prototype displays the user interface for the product and gives a broader view of the entire system, without concentrating on internal functions. A Vertical prototype on the other side is a detailed elaboration of a specific function or a sub system in the product. The purpose of both horizontal and vertical prototype is different. Horizontal prototypes are used to get more information on the user interface level and the business requirements. It can even be presented in the sales demos to get business in the market. Vertical prototypes are technical in nature and are used to get details of the exact functioning of the sub systems. For example, database requirements, interaction and data processing loads in a given sub system.

Final Product:

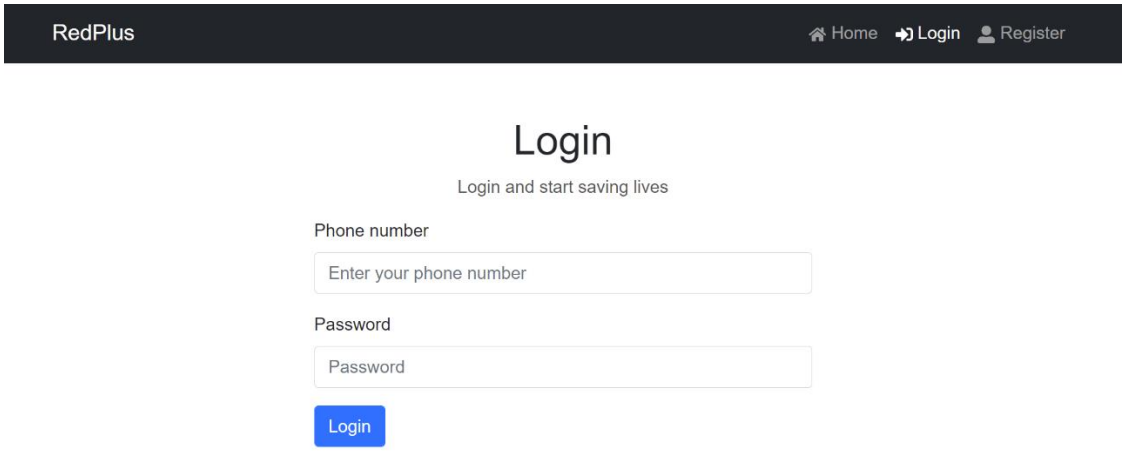
Once the requirements are completely met, the user accepts the final prototype. The final system is evaluated thoroughly followed by the routine maintenance on regular basis for preventing large-scale failures and minimizing downtime.

User interface design

Login page

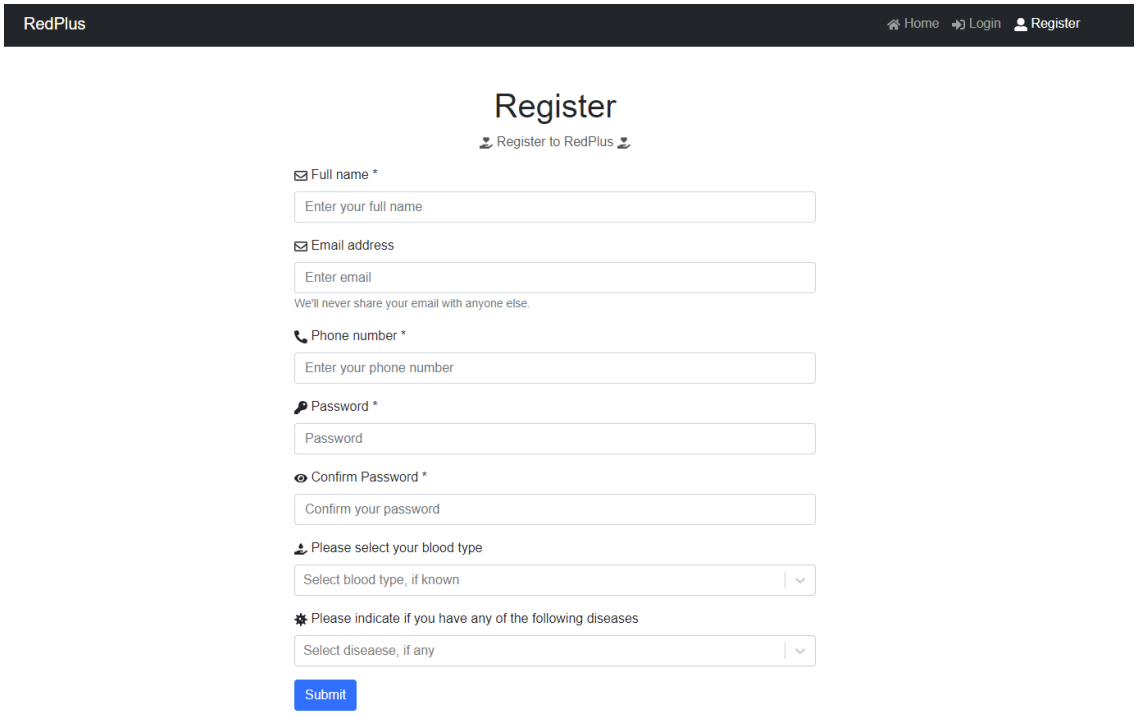
The app will be used by people who are in a hurry so a simple login page should be present.

The login requires only entering the phone number of the user and the password which can be as simple as a 4-digit pin and as complex as a 100-character wide password.



Register page

A simple and easy to use register page is present in the app. User has to enter a small number of fields in order to get registered.

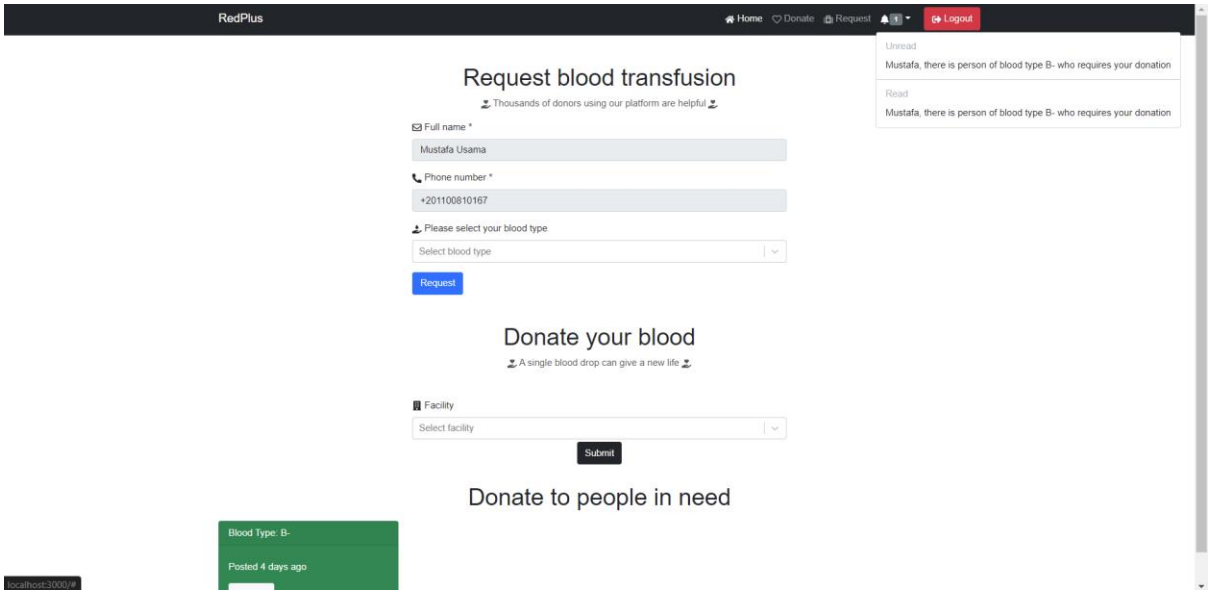


Dashboard

Dashboard, either for a user or a medical expert contains all the functionality of the app in a simple way that the user can develop a memory of the user interface very easily.

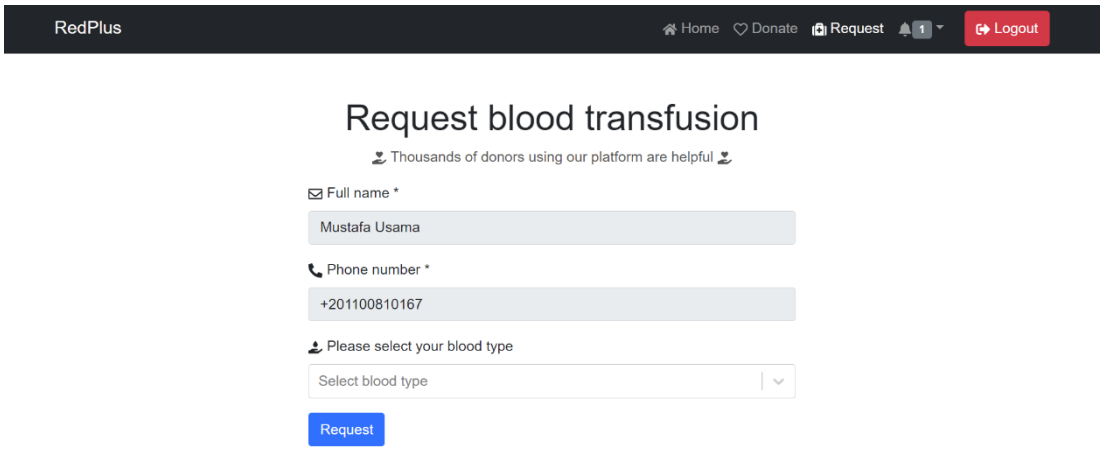
Notifications appear in a non-distracting way. Their details do not show except if the user clicked in the notification bell.

Requesting and donating appear on the dashboard in order to ease the process.



Request blood

Blood request form contains the info of the requester, and requires the requester to only enter the blood type of the request.

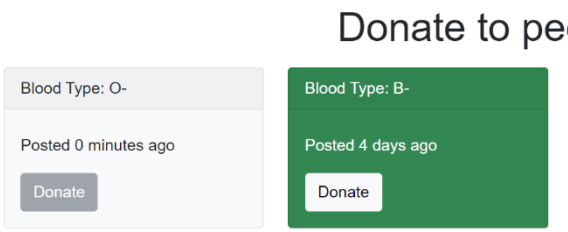


Donate

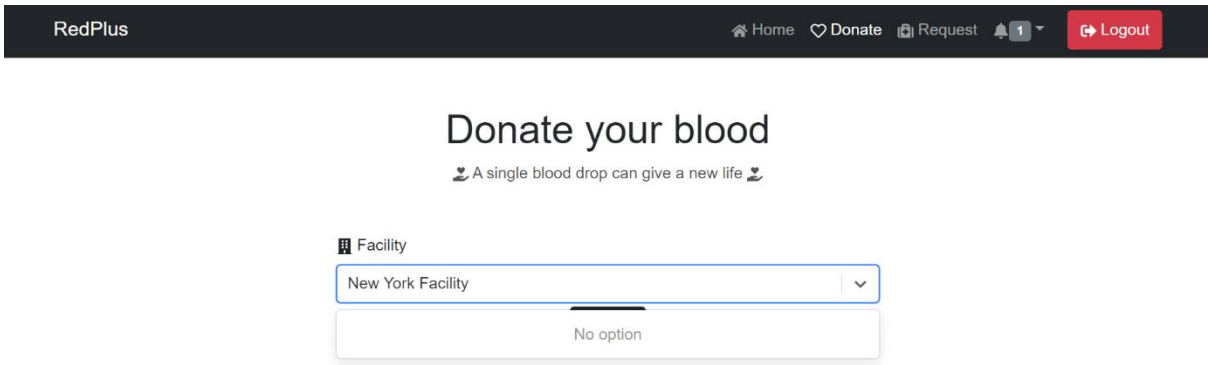
The donate interface appears in a very simple and understandable way. The card color indicates the possibility of a donation where a green color indicates a possible donation and a grey color indicates impossibility of a donation.

The postdate appears in a “minutes ago” or “days ago” format. The blood type appears as a card header.

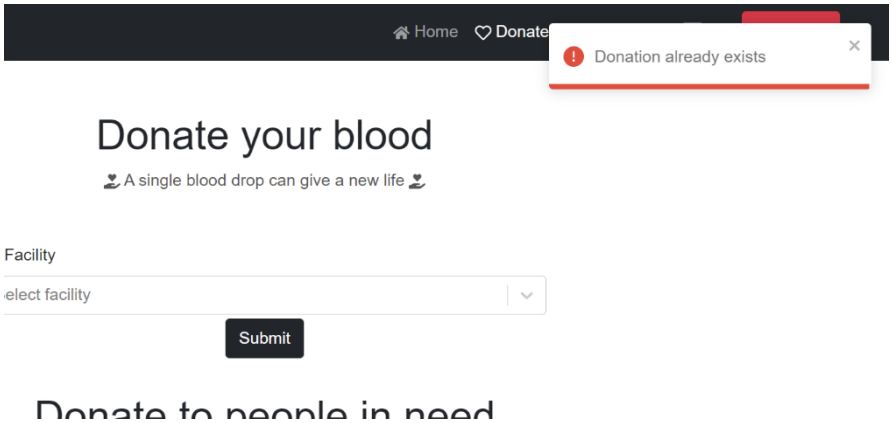
And the Donate button is only enabled in the donations that can be done using the user’s blood type.



Donation to a facility is done easily. The select box will be pre-loaded with all the nearby facilities (2 km from the user's location.) If the user wants to donate to a facility that is not nearby, they only need to enter the facility name or description and it will appear if it exists.

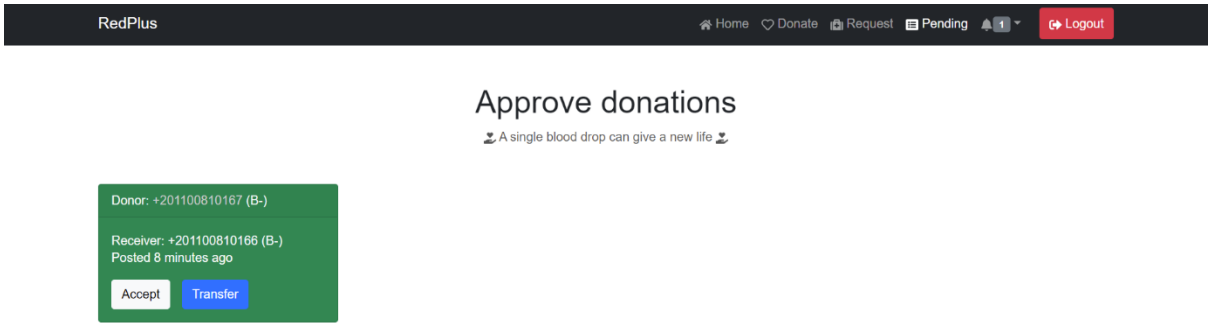


Error messages throughout the system appear in a non-distracting and a friendly way.



Expert area

Expert area (Pending page) has a very simple user interface where the donations that need approval appear as cards. Expert can either accept a donation by just clicking Accept or transfer the donation.



The transfer has a simple and easy interface for the expert to transfer a donation to a specific facility. A popup form appears once the expert clicks Transfer. The facility form in this popup works in the same way the form works in the Donation to Facility for the ease of use for both the user and the expert.

End-User guide

Preparing the local environment

The web application is well-suited to be deployed and run on a server. However, there is always a way to run it locally on any device to use it on a small scale for testing and development.

Setting up the required software

There is a set of software that are required to run the application locally on any machine

1. Node.js

The web application is based entirely based on the Node.js framework. So, we need to install the LTS (life-time support) version of node.js: https://nodejs.org/en/

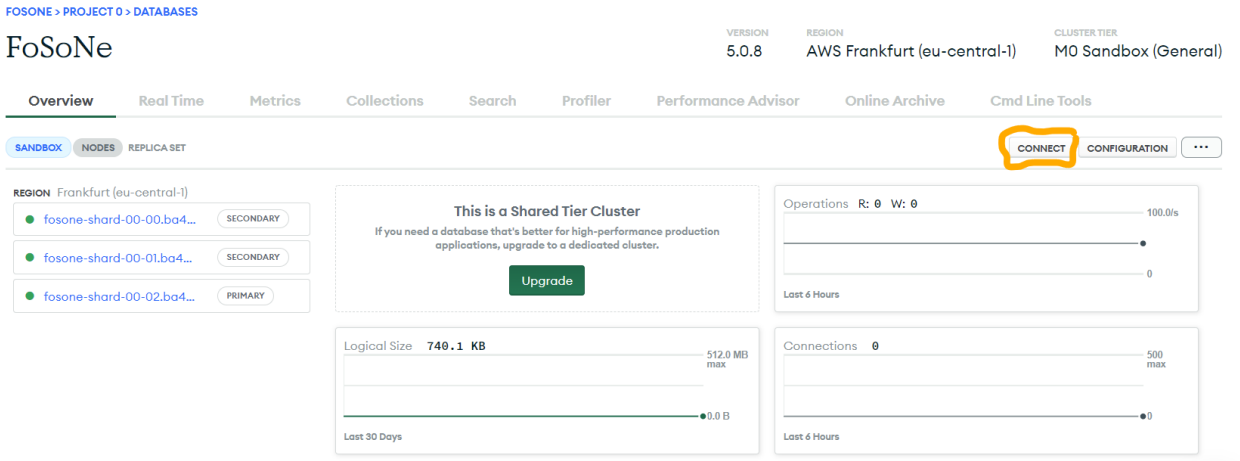
2. MongoDB

RedPlus uses a noSQL database called MongoDB which stores the information in documents of json format. You can use MongoDB either on an online hosting or locally on your machine.

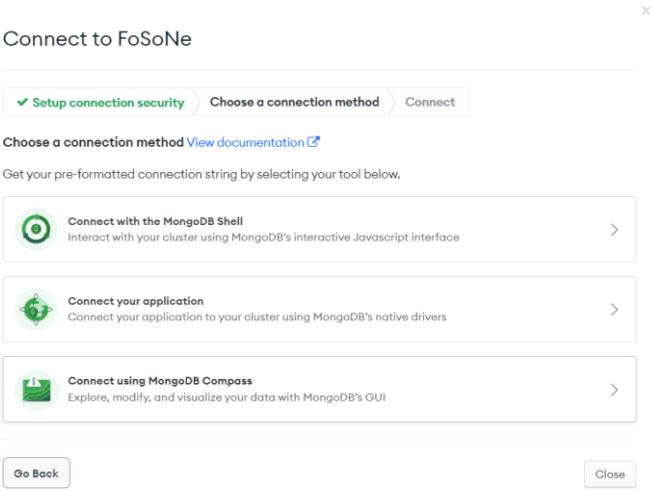
Online choice

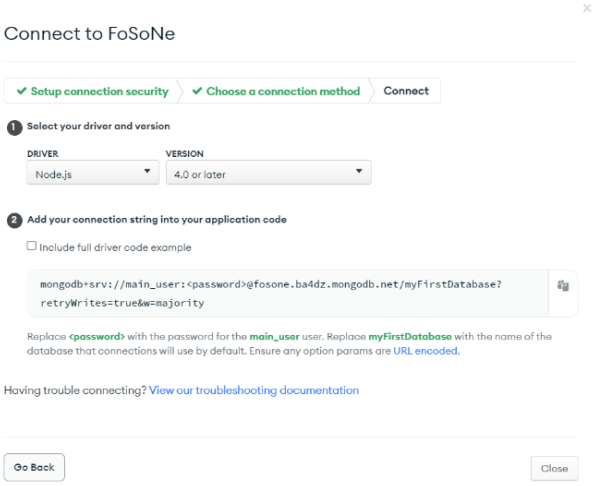
MongoDB can be used online by creating an account on any MongoDB online hosting service and creating a free cluster. Also create a database user and save its credentials for later use.

After creating a cluster, choose connect (this example is on MongoDB atlas hosting)



Connect your application





Copy the link and change ***main_user*** to the user name of your account, change ***<password>*** to your password, and change ***myFirstDatabase*** to the name of the database that you wish to use in RedPlus

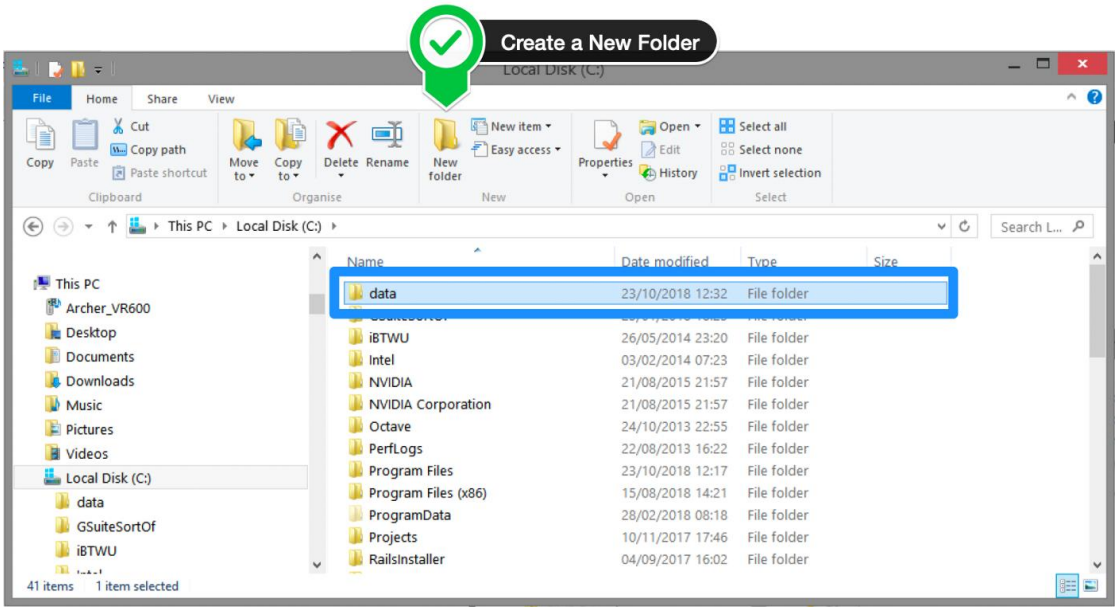
Paste the final URI in **ATLAS_URI** variable in the **.env.example** file, and change the file name to **.env**

Local choice
Local MongoDB Installation can be found here:
https://www.mongodb.com/try/download/community

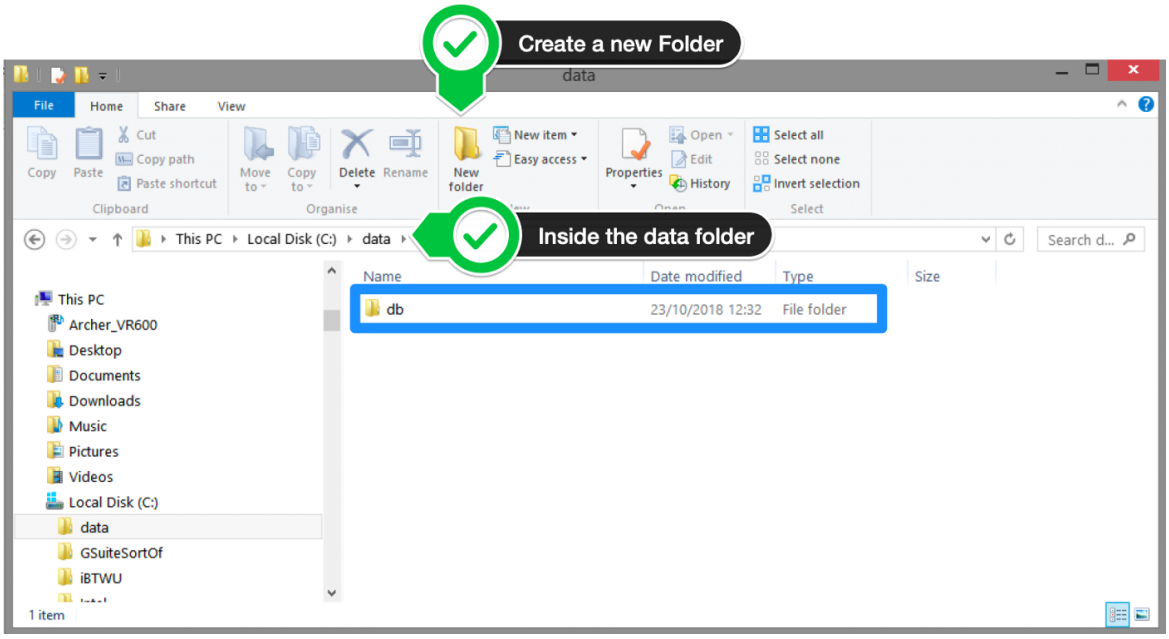
Git bash is also required to correctly follow the following steps. You can install it by installing the GIT (version controlling system) software: https://git-scm.com/downloads

The next page embeds a guide copied from How to Download & Install MongoDB on Windows | by London App Brewery | Medium that we followed during our local database installation.

1. Navigate to the **C Drive** on your computer using Explorer and create a new folder called **data** here.



2. Inside the **data** folder you just created, create another folder called **db**.



Setup Alias Shortcuts for Mongo and Mongod

Once installation is complete, we'll need to set up MongoDB on the local system.

A. Open up your Hyper terminal running Git Bash.

B. Change directory to your home directory with the following command:

```
cd ~
```

C. Here, we're going to create a file called .bash_profile using the following command:

```
touch .bash_profile
```

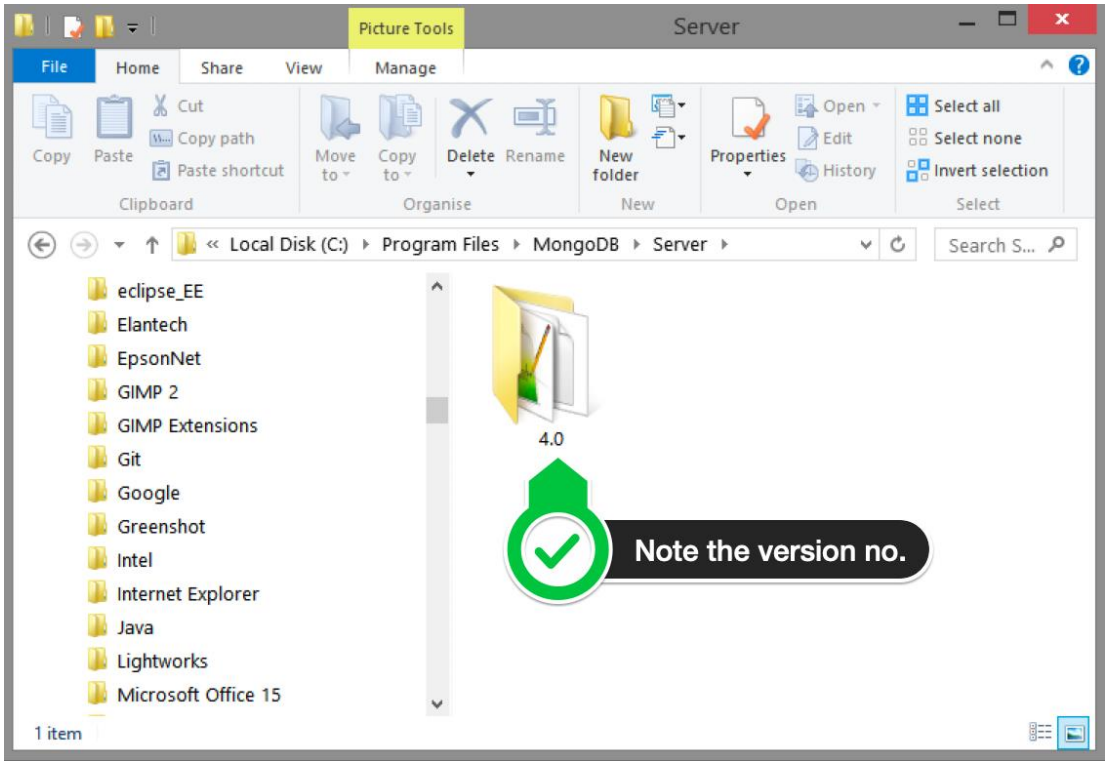
D. Open the newly created .bash_profile with vim using the following command:

```
vim .bash_profile
```

E. In vim, hit the **I** key on the keyboard to enter insert mode.

F. In your explorer go to C → Program Files → MongoDB → Server

Now you should see the version of your MongoDB.



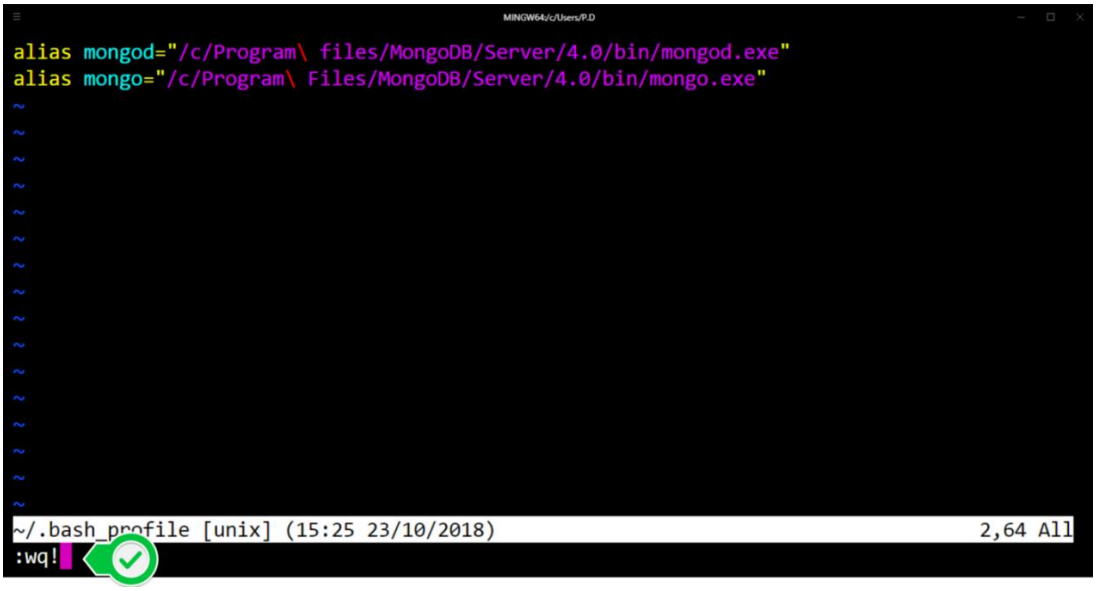
G. Paste in the following code into vim, make sure your replace the 4.0 with your version that you see in explorer

```
alias mongod="/c/Program\ files/MongoDB/Server/4.0/bin/mongod.exe"
alias mongo="/c/Program\ Files/MongoDB/Server/4.0/bin/mongo.exe"
```

H. Hit the Escape key on your keyboard to exit the insert mode. Then type

```
:wq!
```

to save and exit Vim. (screenshot next page)



Verify That Setup was Successful

A. Close down the current Hyper terminal and quit the application.

B. Re-launch Hyper.

C. Type the following commands into the Hyper terminal:

```
mongo --version
```

Once you've hit enter, you should see something like this:

This means that you have successfully installed and setup MongoDB on your local system!

If you see something that looks like bash mongo command not found, then make sure you check back at all the steps above and follow it step-by-step making sure there are no typos and you haven't missed any of the steps.

Setting up the app with dependencies

After unzipping the files open the terminal in the folder where you can find package.json file (the main folder)

Setup the required dependencies

RedPlus is build using some main and development dependencies. All of them are defined with their required versions in the packages.json file. You can install them by running

```
npm install && cd frontend && npm install && cd ..
```

in the terminal.

Change environment variables

The first step in changing the environment variables was in adding the database URI. This second step involves changing the **JWT_SECRET** key which is the signature for encrypting Json Web Tokens that are required in the authentication process. You can change it or keep it as it is.

After making sure the environment variables are set correctly. Change the file name to **.env** if you have not already changed it.

Running the application

The whole application

There is a command for running the front-end (in dev mode) and back-end (in dev mode) concurrently.

You can start the application by running (in the terminal)

```
npm run dev
```

Front-end

You can start the front-end only (dev mode) by running (in the terminal)

```
npm run client
```

You can build the front-end (although you will not need this in a local environment) by running:

```
cd frontend && npm run build && cd ..
```

Back-end

You can run the back-end only by running (in the terminal) for watch mode:

```
npm run server
```

Or, for production mode

```
npm run start
```

The application will be run by default on http://localhost:3000 (if you have not changed the port)

Adding an expert account

In order to test for the expert functionalities, a developer needs to register as a normal user http://localhost:3000/register and then manually change the “role” field of the user in the database to “expert” instead of “user”.

This can be done by typing a terminal script to fetch all users from the “users” collections of the MongoDB client, if it installed locally.

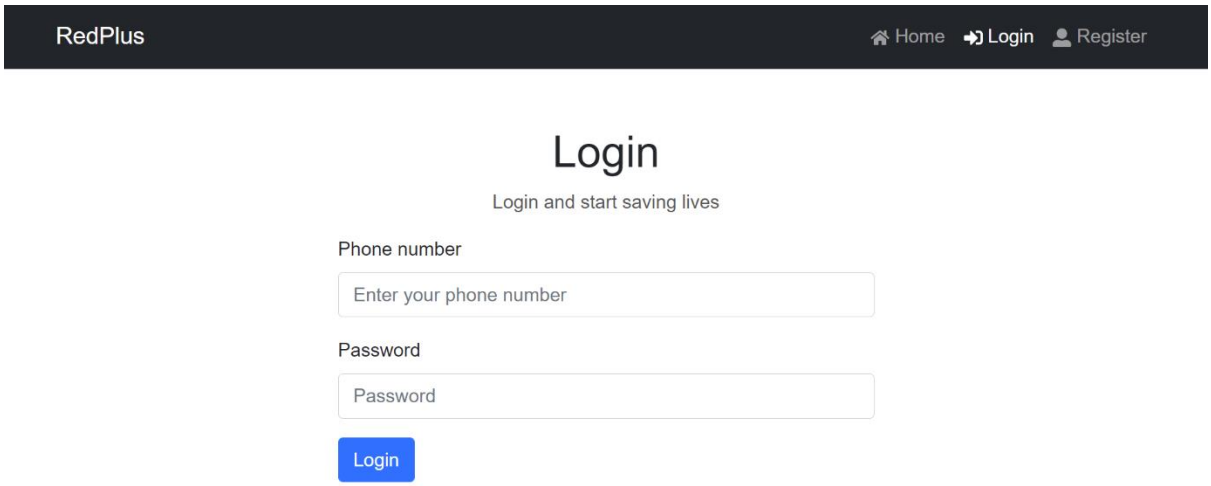
A MongoDB GUI can be used for achieving this goal easily. MongoDB Compass is a tool for showing the databases and manipulating them without writing a script, whether the database is working locally or online on a cluster.

You can install it, enter the database URI for connection, and make changes to the database without a scripting language.

RedPlus user manual

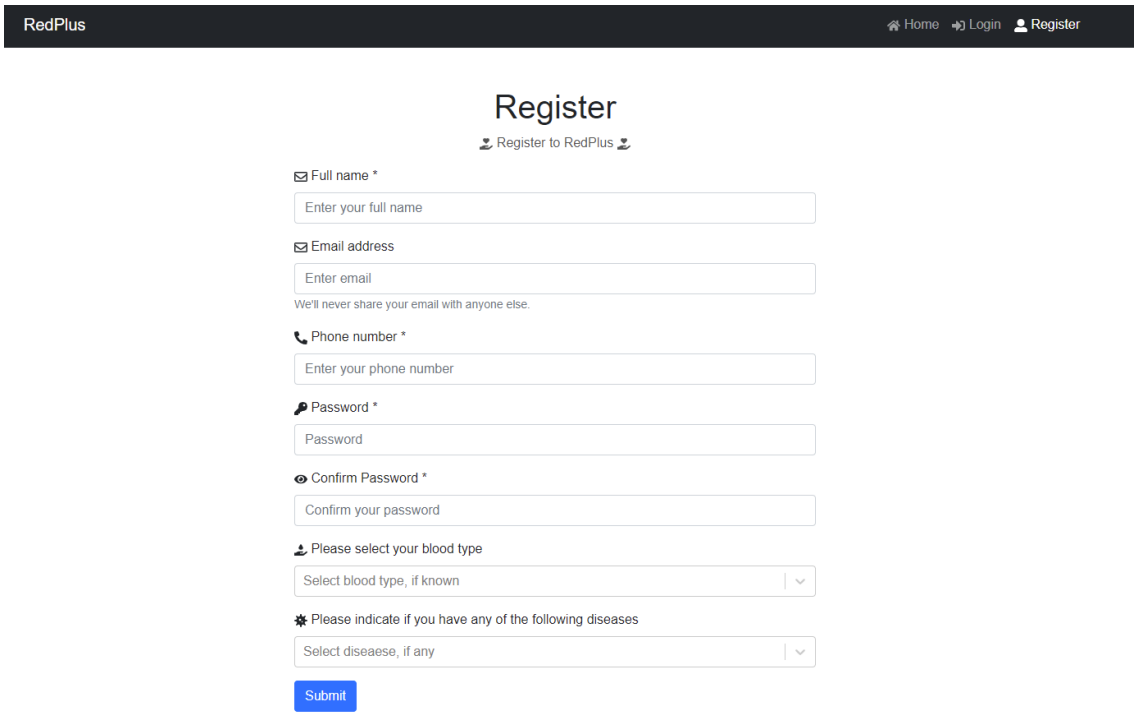
First use

When you first use RedPlus you will be redirected to the Login page. If you have an account please enter your phone number and password.



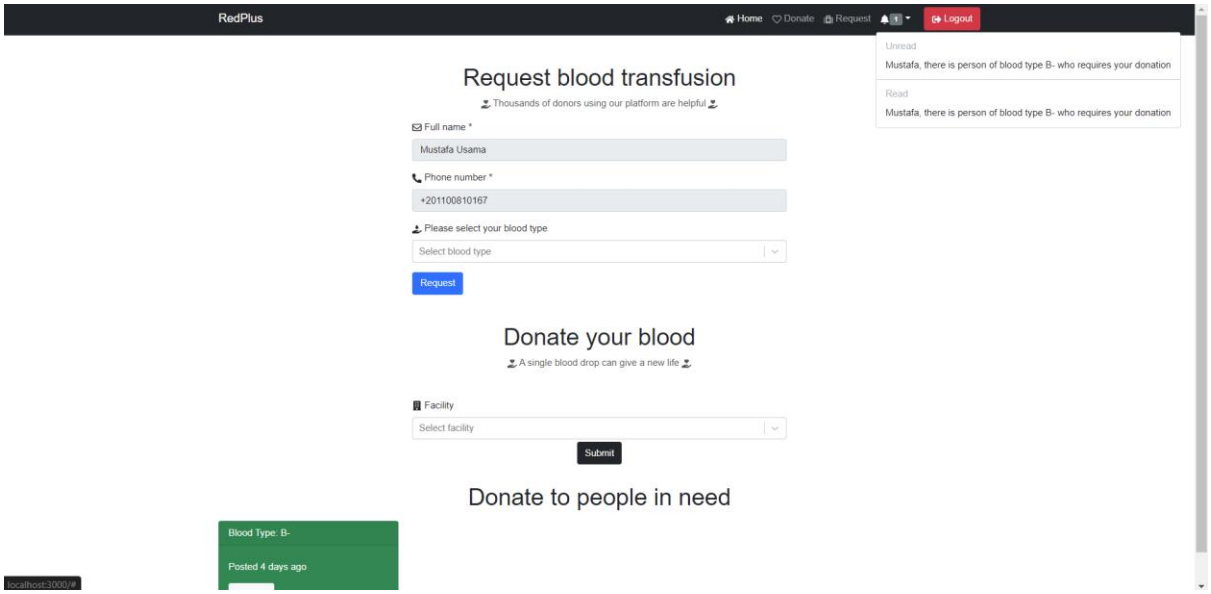
Otherwise press Register in the navbar, and fill your information. Fields that have * are required.

After registering, navigate back to Login and login with your credentials.



Logged in as a user

After you get into the main page, you will be see the various options of the applications.



Notifications

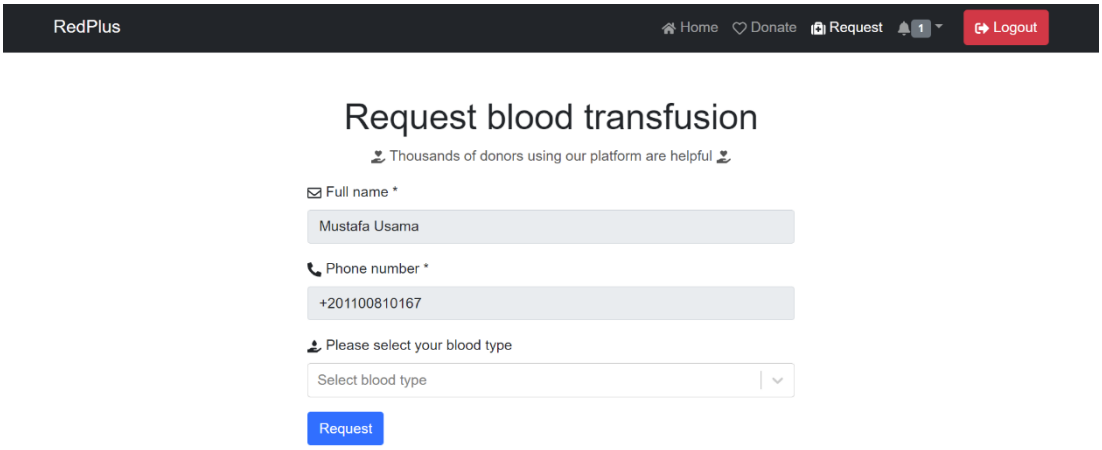
You will see your Unread and Read notifications, you can read an unread notification by simply clicking on it.

Notifications are pushed to you whenever there is someone 2 kilometers or less that are in need for a blood donation that your blood type is suitable for.

You can see all the nearby blood requests by simply navigating to your main page (dashboard) or clicking Donate in the navbar.

Request blood

You can request blood either in the Dashboard or by clicking on Request in the navbar.

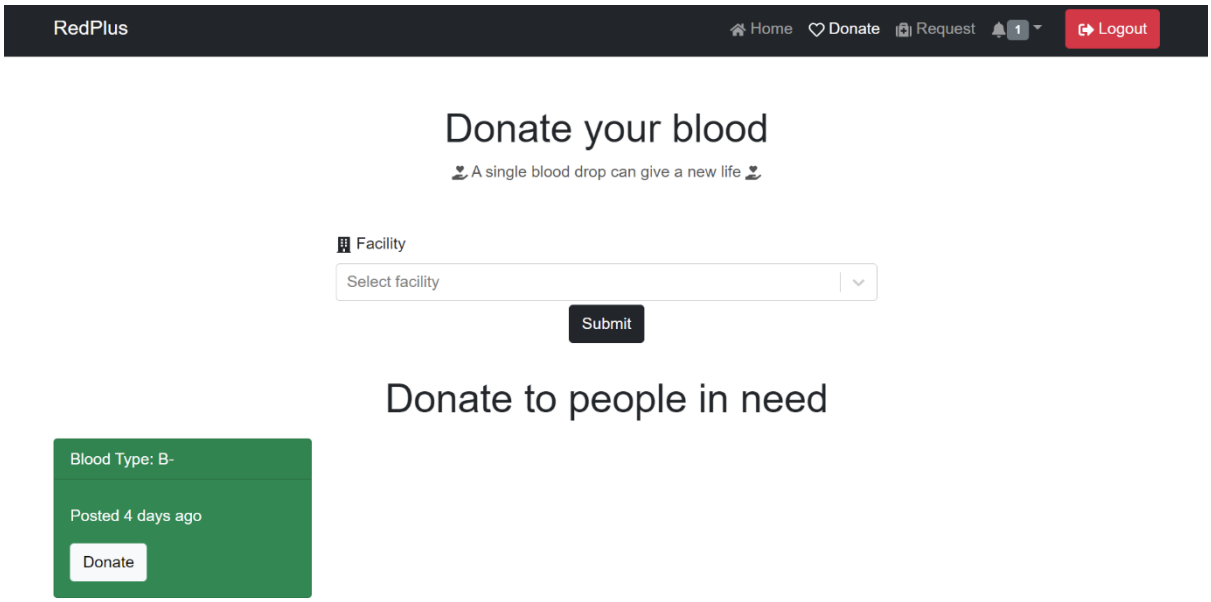


When you choose to request blood. The form will have your full name and your phone number as the requester contact information. You cannot change them. Yet you can select a blood type to request a blood transfusion for, because it is possible and frequent to request blood on behalf of a patient.

When you click Request, your blood request will be added to our database, a notification will be sent to every nearby user with a blood type that can donate to the requested type.

Donate

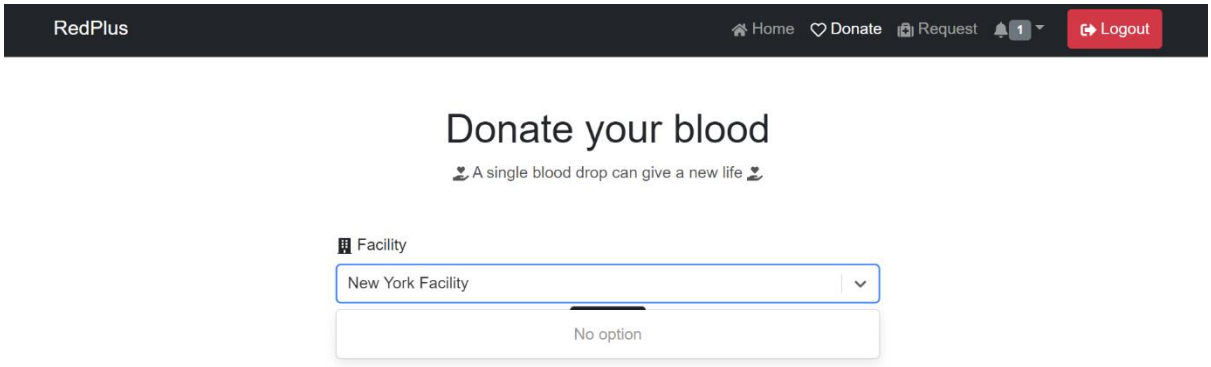
You can donate blood either in the Dashboard or by clicking on Request in the navbar.



When you choose to donate your blood, you can either donate to a facility for storage or donate to a nearby patient in need.

Donating to facility

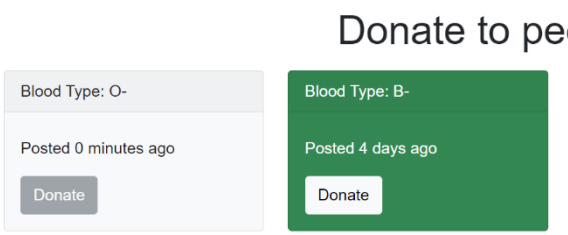
All nearby facilities will be listed in the Select facility box. However, you can select any facility to donate to if you wish.



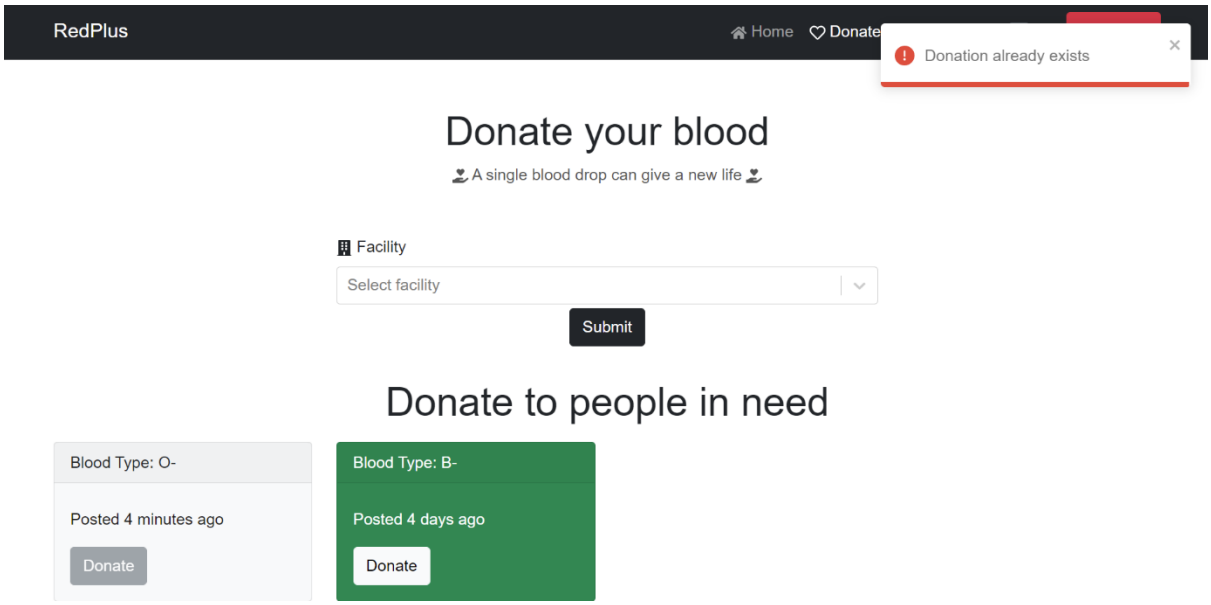
If you search by a keywork, all facilities matching the keyword appear. Otherwise, a “No option” message appears. After selecting an existing facility, you can press Submit to place a donation request. A facility member will be in touch with you as soon as they receive your donation request.

If you choose to donate to nearby patients. All nearby blood requests will appear in cards format with the blood type and the post date. A green card means your blood type is suitable

for that donation. A light-colored card means your blood type cannot be donated to them, but you can help find somebody with a suitable type.



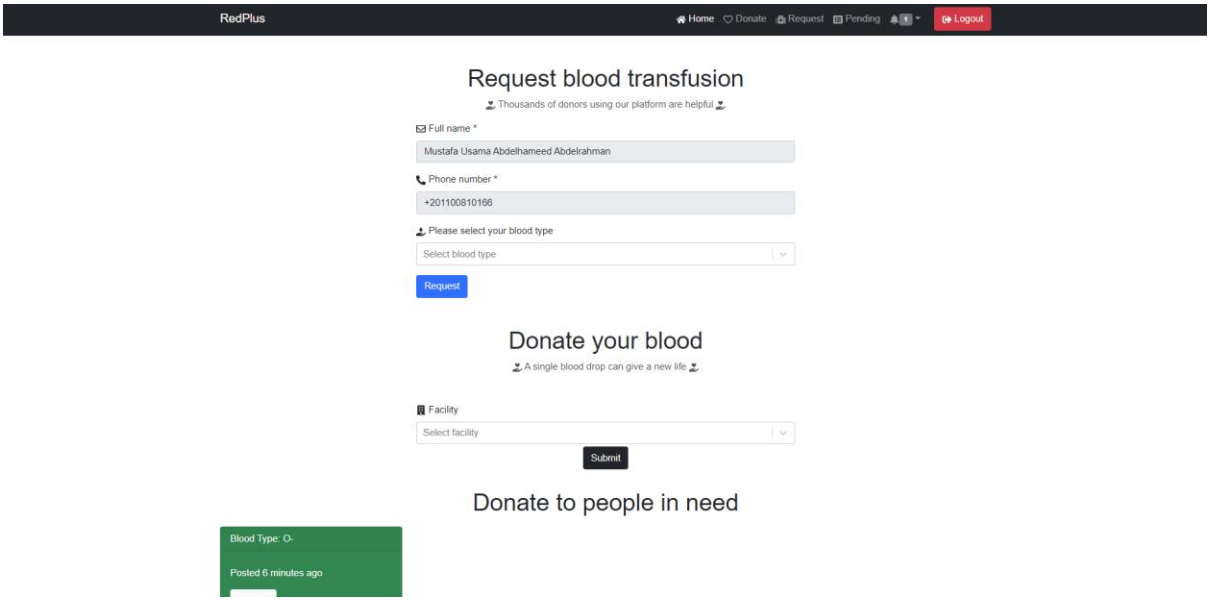
If you click donate, a donation request will be placed, and a success message appears at the top-right corner. However, if you click donate again to the same blood requests an error message will appear.



Logged in as a medical expert

Functionalities of a user

A medical expert user will have all the permissions and functionalities of an app user. Notifications will appear for nearby people in need. Blood request and donation request are available for medical experts.



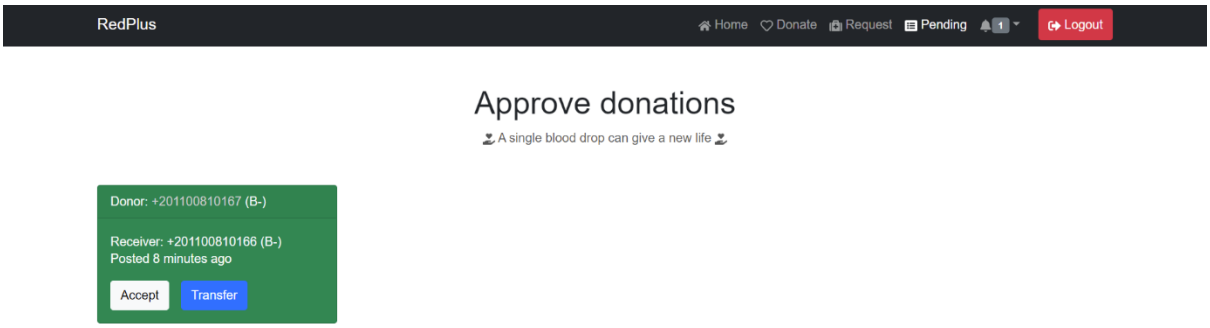
Functionalities of an expert

In addition, a medical expert can:

- see all donation requests placed.
- see donor's and receiver's information.
- accept a donation by a user to a patient or to a facility.
- transfer a donation by a donor from a user to a facility or from a facility to another facility.

Viewing pending donations

A medical expert can access their protected dashboard by clicking on Pending in the navbar. A pending section is where all pending donations reside waiting for an expert acceptance or transfer.



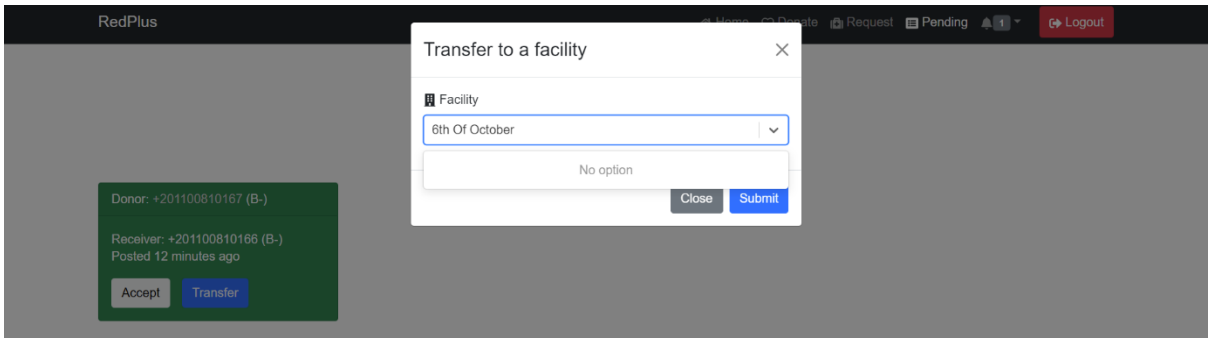
Accepting donations

Since there was a donation created by a user in the previous example. The donation now appears in the expert Pending dashboard as a green card, with a "Receiver" field if the donation is for a specific blood request by a patient, or a "Facility" field if the donation is for a specific facility.

The expert can click Accept to sign the donation as accepted and place himself responsible for that donation. The donor will receive a notification about that acceptance.

Transferring donations to a facility

The expert can also transfer the donation to a facility by clicking on the Transfer button.



A popup will appear requiring the expert to enter the facility name, and if it exists, the expert can select it and press Submit to transfer the donation to that facility and the facility database collection will have the donation appended to it.

The expert is then responsible for contacting the donor to coordinate for the donation process.