PLASMA DONOR APPLICATION

1. INTRODUCTION

1.1 Project Overview

Blood is an important constituent of human body. Timely availability of quality blood is a crucial requirement for sustaining the healthcare services. In the hospital, in most of the cases, when blood is required, could not be provided on time causing unpleasant things. Though donor is available in the hospital, patient is unaware of it, and so is donor. To resolve this, a communication between hospital, blood bank, donor, and receptor is important. The system listed with following forecasting on price variations and stock handling, increase in number of blood type, increase in human accident Infrastructure, blood on various category to be managed. So we solve the problem using the android application. The system will make sure that in case of need, the blood will be made available to the patient. There will be android app to make this communication faster. It aims to create an information about the donor and organization that are related to donating the blood. The methodology used to build this system uses GPS. The Proposed system will be used in Blood banks, Hospitals, for Donors and Requester whoever registers to the system.

1.2 Purpose

Blood Donation System is an android based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and helps them to manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective document is a template.

The sole purpose of this project is to develop a computer system that will link all donors, control a blood transfusion service and create a database to hold data on stocks of blood in each area. Furthermore, people will be able to see which patients need blood supplies via the android application.

2. LITERATURE SURVEY

2.1 Existing Problem

YEAR	TITLE	AUTHOR(s)	TECHNIQUE(s)	PROS	CONS
2022	Instant Plasma Donor Recipient connector web application	Kalpana Devi Guntoju,Tejaswini Jalli, Sreeja Uppala, Sanjay Mallisetti	Web Technologies, API, Database	The Donor needs to upload their recovered COVID-19 Certificate and it required toverified by the blood bank. It is a user-friendly application. It will help people to find plasma easily.	This is system is closed for general plasma donation and mainly focused on COVID-19 patients for plasma donation
2021	BDoor App- BloodDonation Application using Android Studio	S Periyanayagi, A Manikandan, M Muthukrishnan,and M Ramakrishnan	Android, Flutter UI, Dart, Firebase, Decision tree algorithm	The Donor details are verified before they allow todonate and have to authorized by institution. TheVerification and validation are done in Emailbase.	The android mobile user will not be able to insert or view details if the server goes down. Thus, there is disadvantage of single point failure.
2020	Lifesaver E-BloodDonation App Using Cloud	Rishab Chakrabarti, Asha Darade,Neha Jadhav,Prof. S. M. Chitalkar	E-health, GPS, Blood bank database, Cloud Computing	Reduction in the errors ofblood bank using most eligible donor method. Direct Communication Between donor and the person in need of bloodDuring the Emergency situation.	The user given detailsare maintained unverified.
2020	Developing a plasma donor application using Function-as-a-service in AWS	Aishwarya R Gowri	Serverless, aws, plasma theory, covid19, dynamoDB, cloud	The efficient way of findingplasma donor for the infected people. AWS lambda function is usedand to deploy the application AWS EC2 service is used.	The user interface canbe better than now.
2019	D'WORLD: BloodDonation App Using Android	A. Meiyappan, K. Loga Vignesh, R. Prasanna,	Android, Global Positioning System (GPS), Mobile	When the giver gives the blood, it will naturally evacuate the contributor detail for	The user must have an device with android operating system

		T.Sakthivel	Computing	next three months.It additionally confirms with the Department of Health and Welfare to guarantee the benefactor medical case history.	with an active internet connection to interact with this application.
2018	Automated blood bank system usingRaspberry PI	Ashlesha C. Adsul, V. K. Bhosale, R. M. Autee	Raspberry Pi, EmbeddedBlood Bank, GSM, Android	When there is urgent need for blood then If this model is adopted the caller is immediately connected tothe donor	Tackling the fake users.
2018	Blood donation andlife saver-blood donation app	M.R. Anish Hamlin, J. Albert Mayan	Android, GPS, CloudComputing	One-Time Password (OTP) is used to verify the donor, once the donor accepts the request. Once the donor donates theblood it will automatically remove the donor detail fornext three months.	This application searches for donorsonly in the nearest areas.
2018	Android Based Health Application in Cloud Computingfor Blood Bank	Sayali Dhond, PradnyaRandhavan, Bhagyashali Munde, Rajnandini Patil	Cloud Computing, Global Positioning System (GPS), Web Technologies, Android.	Accessibility and availability are the criteria on which an application is designed for itssuccess in the IT market.	Requires the patientrecords to be accurate and accessible.
2016	mHealth: Blood donation application using android smartphone	Muhammad Fahim, Halil Ibrahim Cebe, Jawad Rasheed and Farzad Kiani	Android (operating system), medical computing, mobile computing	mHealth is one of the best possible concepts for the provision of healthcare services and improve qualityof life.	We have to utilize thecloud computing service for keeping the application data available, anywhere and anytime.
2015	An Android Application for Volunteer Blood Donors	Sultan Turhan	Distance Calculation, Web Services, GPS, Databases	This application helps healthcare centers to provide the blood as quick as possible when their stocks are insufficient. The application sends periodically actual location information of available donors to main system and the	If the stocks are insufficient, the only source of blood supply will be the people who come to the health center and donate the blood on avoluntary basis

				blood requests to the donors.	

2.2 References

1. Kalpana Devi Guntoju,Tejaswini Jalli, Sreeja Uppala, Sanjay Mallisetti,"Instant Plasma Donor Recipientconnector web application",(2022).
2. S Periyanayagi,A Manikandan,M Muthukrishnan,and M Ramakrishnan,"BDoor App-Blood Donation Application usingAndroid Studio",(2021).
3. Rishab Chakrabarti, Asha Darade, Neha Jadhav, Prof. S. M. Chitalkar,"Lifesaver E-Blood Donation App Using Cloud",(2020).
4. Aishwarya R, Gowri, "Developing a plasma donor application usingFunction-as-a- service in AWS",(2020).
5. A. Meiyappan, K. Loga Vignesh, R. Prasanna, T.Sakthivel,"D’WORLD: Blood Donation App Using Android",(2019).
6. Ashlesha C. Adsul, V. K. Bhosale, R. M. Autee,"Automated blood bank system using Raspberry PI",(2018).
7. M.R. Anish Hamlin, J. Albert Mayan,"Blood donation and life saver-blood donation app",(2018).
8. Sayali Dhond, PradnyaRandhavan, Bhagyashali Munde, Rajnandini Patil,"Android Based Health Application in Cloud Computing for Blood Bank",(2018).
9. Muhammad Fahim, Halil Ibrahim Cebe, Jawad Rasheed and Farzad Kiani,"mHealth: Blood donation application usingandroid smartphone",(2016).
10. Sultan Turhan,"An Android Application for Volunteer BloodDonors",(2015).

2.3 Problem Statement Definition

Plasma is commonly given to trauma, burn and shock patients, as well as people with severe liver disease or multiple clotting factor deficiencies. It helps boost the patient's blood volume, which can prevent shock, and helps with blood clotting. With the number of people affected by Covid-19 infection, the demand for the plasma of recovered patients has also gone up tremendously. The antibodies, which are present in our body, can help someone fight the infection and emerge victorious. Why plasma Donor application? Plasma donation saves lives, and the communication between blood/plasma centers and donors plays a vital role in this. Smart apps are now considered an important communication tool, and could be best utilized in plasma donation if they are designed to fit the users’ needs and preferences.

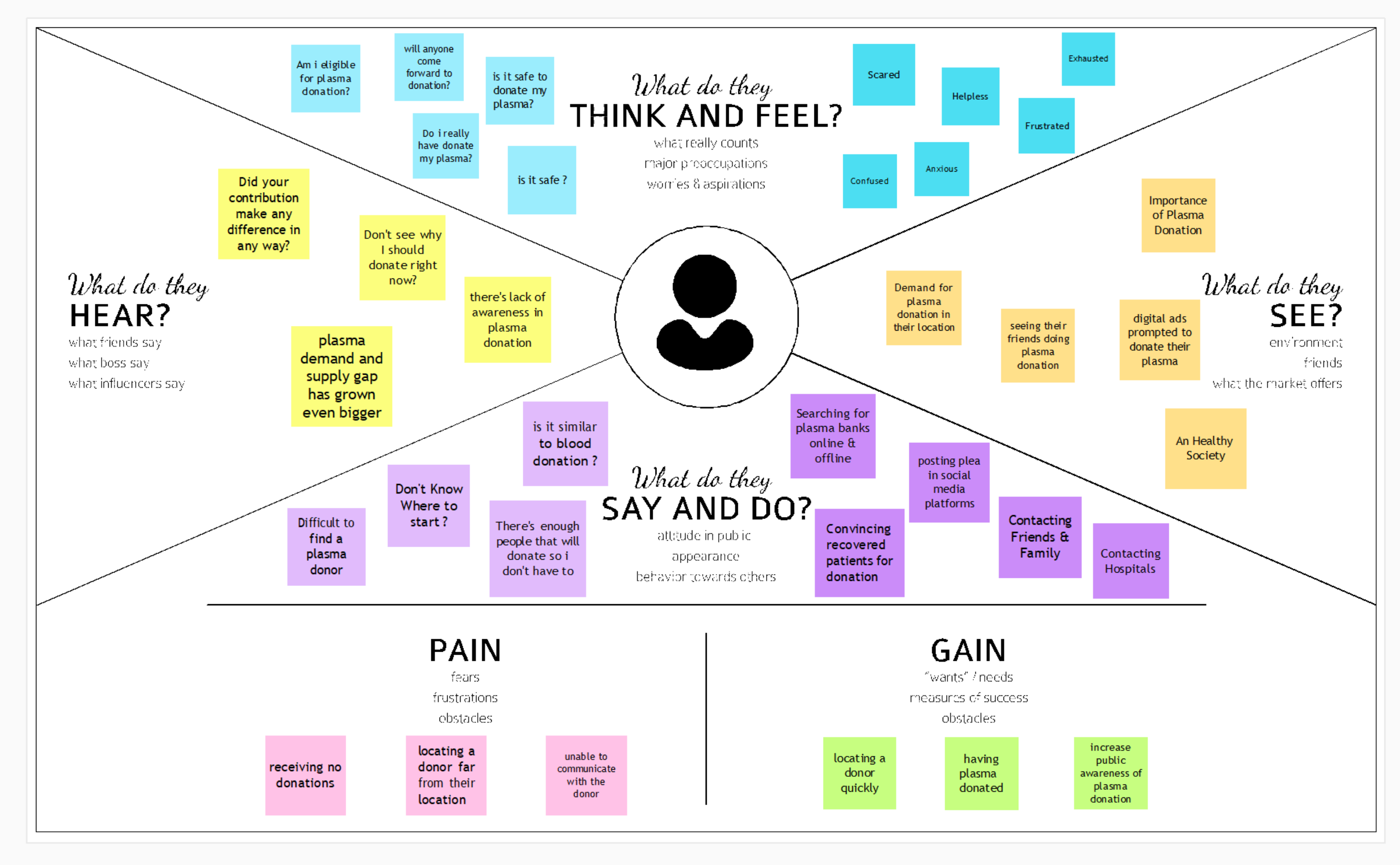
Our view: We plan to make a User-friendly application for users who are in need for plasma or who wish to donate plasma to anyone who are in need. However, areas of concern, including privacy and confidentiality, should be considered during design and development. Age was identified as a contributing factor that might decrease the likelihood of app usage among donors. The donation center staff focused on the educational features of the app and emphasized the importance of the app providing statistics and sending notifications and reminders to donors.

ABSTRACT:

This system is used if anyone needs a Plasma Donor. This system comprises of Admin and User where both can request for a Plasma. Both parties can Accept or Reject the request. The person who wants to donate his/her plasma needs to register in our application providing required information which are name, age, blood group, phone number, and location, etc. Patients who need plasma can also fill the form to request the plasma. Patients can directly call the donor by taking his/her contact number from the application. User can also search based on location they are living. Just a single search allows anyone to reach maximum number of plasma donors in minimum possible time

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they THINK AND FEEL?

what really counts

major preoccupations

worries & aspirations

Am i eligible for plasma donation?

will anyone come forward to donation?

is it safe to donate my plasma?

Do i really have donate my plasma?

is it safe ?

Scared

Helpless

Exhausted

Frustrated

Confused

Anxious

What do they HEAR?

what friends say

what boss say

what influencers say

Did your contribution make any difference in any way?

Don't see why I should donate right now?

there's lack of awareness in plasma donation

plasma demand and supply gap has grown even bigger

What do they SEE?

environment

friends

what the market offers

Importance of Plasma Donation

Demand for plasma donation in their location

seeing their friends doing plasma donation

digital ads prompted to donate their plasma

An Healthy Society

What do they SAY AND DO?

attitude in public

appearance

behavior towards others

is it similar to blood donation ?

Don't Know Where to start ?

Difficult to find a plasma donor

There's enough people that will donate so i don't have to

Searching for plasma banks online & offline

posting plea in social media platforms

Convincing recovered patients for donation

Contacting Friends & Family

Contacting Hospitals

PAIN

fears

frustrations

obstacles

receiving no donations

locating a donor far from their location

unable to communicate with the donor

GAIN

"wants" / needs

measures of success

obstacles

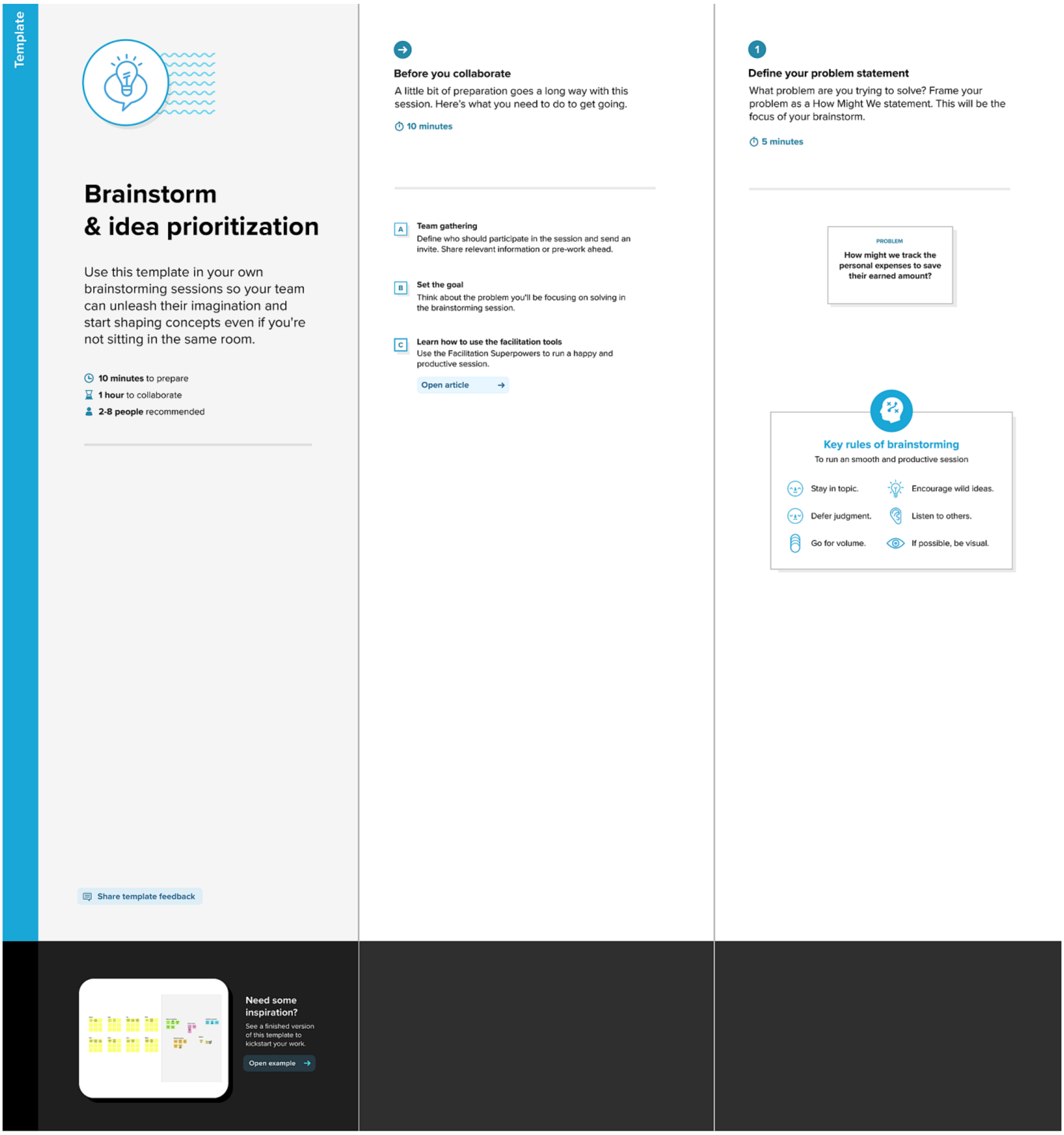
locating a donor quickly

having plasma donated

increase public awareness of plasma donation

3.2 Ideation & Brainstorming

Brainstorming



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Need some inspiration?

See a finished version of this template to kickstart your work.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

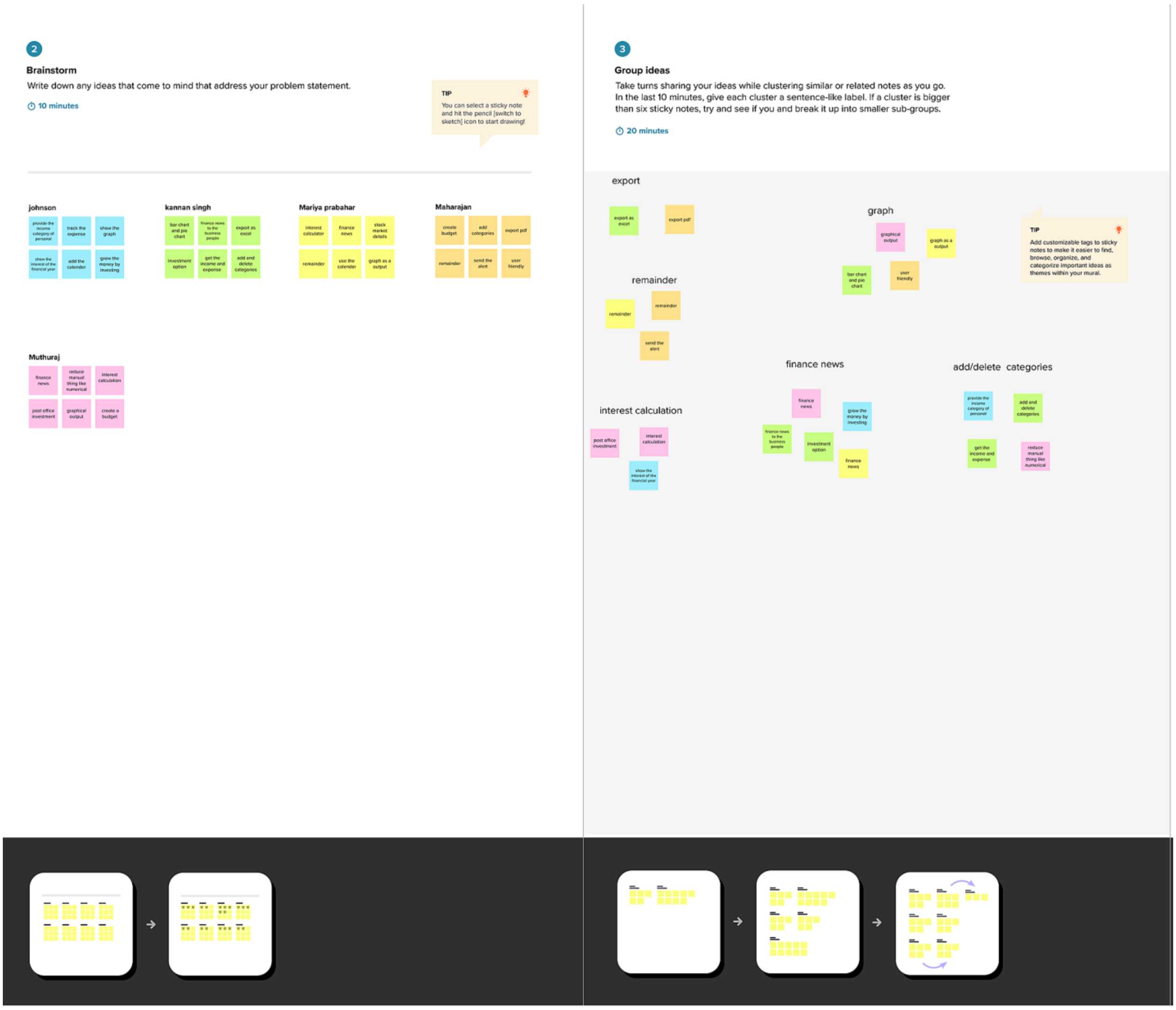
PROBLEM

How might we track the personal expenses to save their earned amount?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

johnson

provide the income category of personal	track the expense	show the graph
show the interest of the financial year	add the calender	grow the money by investing

kannan singh

bar chart and pie chart	finance news to the business people	export as excel
investment option	get the income and expense	add and delete categories

Mariya prabahar

interest calculater	finance news	stock market details
remainder	use the calender	graph as a output

Maharajan

create budget	add categories	export pdf
remainder	send the alert	user friendly

Muthuraj

finance news	reduce manual thing like numerical	interest calculation
post office investment	graphical output	create a budget

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

export

- export as excel
- export pdf

graph

- graphical output
- graph as a output
- bar chart and pie chart
- user friendly

remainder

- remainder
- remainder
- send the alert

finance news

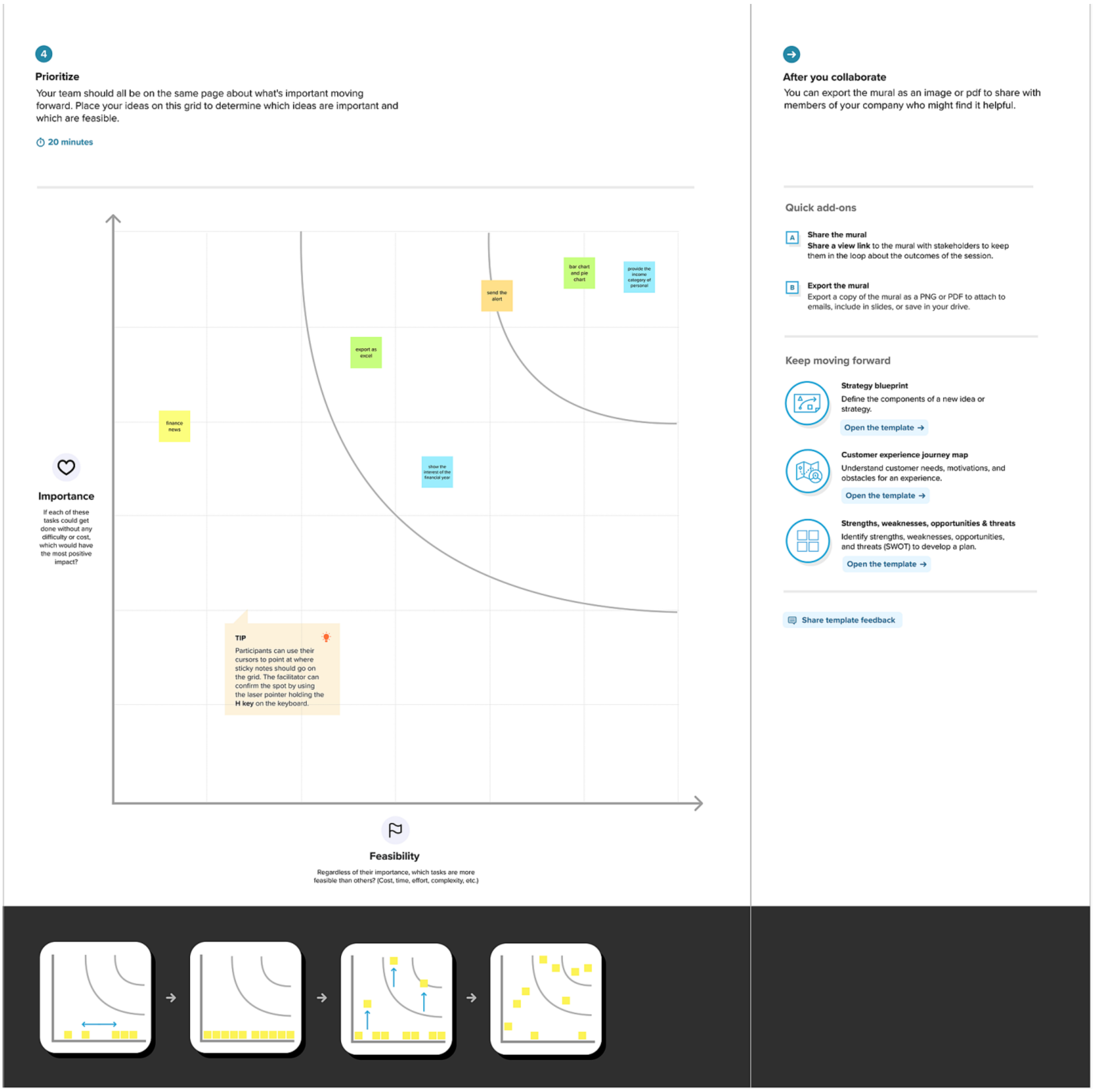
- finance news
- grow the money by investing
- finance news to the business people
- investment option
- finance news

add/delete categories

- provide the income category of personal
- add and delete categories
- get the income and expense
- reduce manual thing like numerical

interest calculation

- post office investment
- interest calculation
- show the interest of the financial year



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

send the alert
bar chart and pie chart
provide the income category of personal
export as excel
finance news
show the interest of the financial year

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

IDEATION

Ideation is the creative process of generating, developing, and communicating new ideas, where an idea is understood as a basic element of thought that can be either visual, concrete, or abstract. Ideation comprises all stages of a thought cycle, from innovation, to development, to actualization. Ideation can be conducted by individuals, organizations, or crowds. As such, it is an essential part of the design process, both in education and practice.

3.3 Proposed Solution

1. Problem Statement (Problem to be solved)

Plasma is used for the treatment of many serious health problems. This is why there are blood drives asking people to donate blood, plasma . Plasma is utilized to treat different irresistible sicknesses and it is one of the most established strategies known as plasma During treatment. Coronavirus emergency the necessity for plasma expanded radicallyas there were no immunization found to treat the contaminated patients, with plasmatherapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information aboutthe plasma donors. Saving the contributor data and tellingabout the ongoing givers would be someassistance as it can save time and assist the clients with finding the vital data about the contributors.

2. Idea / Solution description

This proposed system aims at connecting the donors & the patients by an online application. By using this application, the users can either raise a request for plasmadonation or requirements.

The basic solution is to create a centralized system to keep a track on the upcoming as well as past Plasma Donation Events. There commendation solution is as follows:

Application contains two roles:

- ➢ Admin
- ➢ User

User:

1. If the user wants to donateor receive they have to register with their personaldetails.
2. After successful registration of user .
3. A successful registration email issend to the user .
4. After successful registration userwill be directed to home page.
5. They will be asked to press whetherthey will be donor or receiver.
6. If the user is donor then he/she willfill the donation interest form which includes their Name, bloodgroup details, location, last time donateddate , phone number, email id.
7. After filling the donation form he/she will redirected to page in which he/shecan download the e-certificate.
8. If the user is receiver then he/she cansee the list of donors available and they can raise their request and contact donor directly.

Admin:

1. Admin can login using their credentials.
2. Admin can edit the request.
3. Admin can delete the request.
4. Admin can add volunteers.

3.Novelty / Uniqueness

A User Interface is simple for users tounderstand. We can use the applicationany where anytime. The user immediately need the plasma for their treatment but theplasma is not available in nearby hospitals, then user can use this application to raise request and directly contact the donor , request them to donate the plasma. Hospitals can also raise request donors for donation. Somebody wants to donate blood and plasma but they don't know the way to donate then they use this application which will simple to use and it will save lives of many people.Today many of them have mobile phones they can install this application and use it to save the lives of people.

4.Social Impact / Customer Satisfaction

We are living in a modern world and everything can be accessed online. Even though there are many application there is no proper application for plasma donation. Many of them wish to donate blood and plasma but they are unaware about donationand how they can donate.This application provides opportunity to those who want to donate plasma.

Donation of plasma arehappening in many places many of them come forward todonate but it is not available at right time for use. Sometimes there is a shortage of plasma of particular type. Additional facilities that we need is to access the patients information quickly before plasma transfusion. To solve this issue software applications are employed with Cloud computing and Internet of Things tool which enable features such as information retrieval and continuous data tracking with analytics. This application avoids circulating of wrong information. A single platform for maintaining genuine information and increase the trust of participants involvedin this activity .It increases the number of donors.

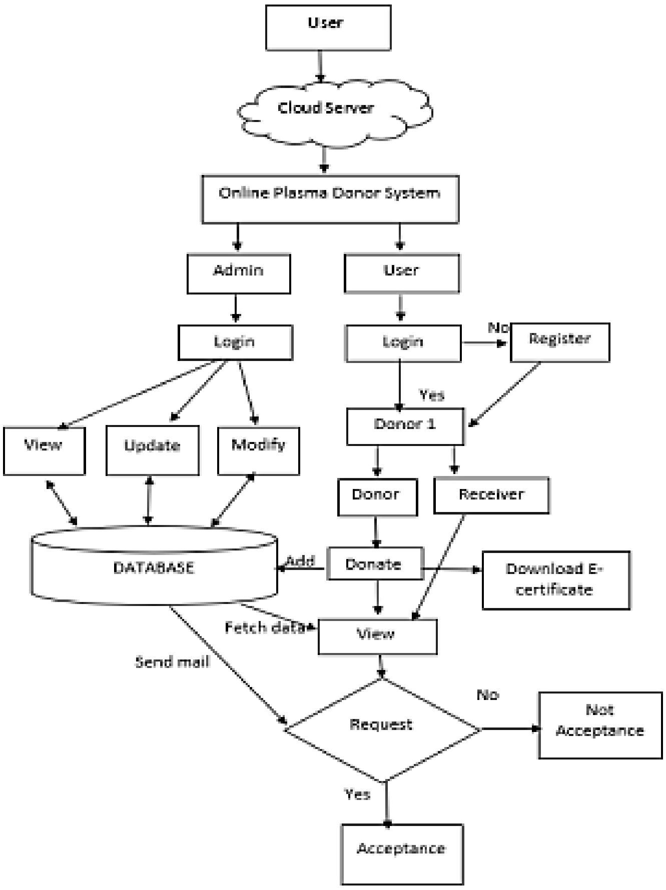
5.Business Model (Revenue Model)

This application is accessible by everyone . It is free.Because of the trouble in finding givers who match a specific blood bunch, this application empowers clients to enlist individuals who wish to give plasma and keep their data in a data set. Nowadays the need for plasma increases. Anyone with basic knowledge can access this app. This can be used anywhere anytime. working with the government we can utilize an application to help those needing plasma.

6.Scalability of the Solution

This application helps users to find plasma donors by sitting in home itself instead of searching donors everywhere. When there is a emergency then plasma request to send to everyone. Once the donor is ready to donate receiver is notified about donation. Receiver can contact the donor. With this app donor can know the eligibility to donate and making it easier to locate suitable donor at right time.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

1. Access Website:

Software operator should be capable to access web-application through either an application browser or similar service on the PC. There should not be any limitation to access web-application.

2. Software operator Registration:

Given that software operator has accessed web-application, then the software operator should be able to register through the web-application. The donor software operator must provide first name,gender, blood or plasma group, location, contact , software operator name and password.

3. New Releases:

When a new/update/revise version of the web-application is released, the appearance will be automatically appears when the software operator access the web-application.

4. Software operator log-in:

Given that the software operator has registered, then the software operator should be able to login to the web-application. The login information will be stored on the database for future use.

5. Search result in a list view:

Search result can be viewed in a list. Each element in the list represents a specific donor. Each elementshould include first name, gender, blood or plasma group,location, contact according to the software operator position.There should be maximum of ten result display.

6. Request Blood or plasma:

Software operator(Clinic) should be able to request for blood or plasma at emergency situation, software operator need to define blood or plasma group, location, required date, contact. The blood or plasma request requested will be sent to blood or plasma bank and then to the Inventory to check the availability. If available, the requested blood or plasma will be sent to the requested donor(Clinic).

7. View Request:

The Blood or plasma Bank should be able to view received request and then respond to them and can search requests by selecting two options select blood or plasma group and provision.

8. Search Blood or plasma Bank Stock:

Receiving the blood or plasma request from Clinic , the blood or plasma stock in the requested blood or plasma request. Thus matched blood or plasma units will be sent to the Clinic.

9. View Blood or plasma request Details*:*

The Clinic, Blood or plasma Bank should be able to view the Blood or plasma requestId, time of the blood or plasma request placed, name of the clinic, location and the address of the clinic. In addition to this an additional feature of tracking the distribution person which includes his location and the checkpoints passed.

10. View Distribution Status:

The Clinic, Blood or plasma Bank should be able to view the status of the distribution time. If the distribution seems tobe delayed then the clinic manager must to able to call the distribution person to get the update/revise on the distribution.

4.2 Non-Functional Requirements

Non functional requirements of Blood bank Management System Project

1. Maintainability:
2. Serviceability
3. Environmental
4. Data Integrity
5. Usability
6. ScalabilityReliability
7. Recoverability
8. Interoperability
9. Capacity
10. Performance
11. Security
12. Regulatory
13. Availability
14. Manageability

Maintainability:

The Blood bank Management System have must have high level of Maintainability.

Serviceability

If issue arises in the Blood bank Management System, then the project must be programmed in such a way that developer can service it again.

Environmental

The Blood bank Management System must be working in latest operating system environments like windows 7, windows 8, windows 10 and on linux.

Data Integrity

All the data in the Blood bank Management System must be accurate and reliable.

Usability

The Blood bank Management System must have a good looing user friendly interface.

Recoverability

The Blood bank Management System must have a proper data backup mechanism.

Interoperability

The Blood bank Management System must work with or use the parts or equipment of another system.

Capacity

The Blood bank Management System must fulfill on storage requirements, today and in the future. The Blood bank Management System must be scale up for increasing volume demands.

Performance

The Blood bank Management System must perform well in different scenarios.

Security

The Blood bank Management System must be secured with proper user name and passwords.

Regulatory

The Blood bank Management System must obey all the governmental requirements and constraints.

Availability

The Blood bank Management System must be available 24 hours a day with no bandwidth issues.

Manageability

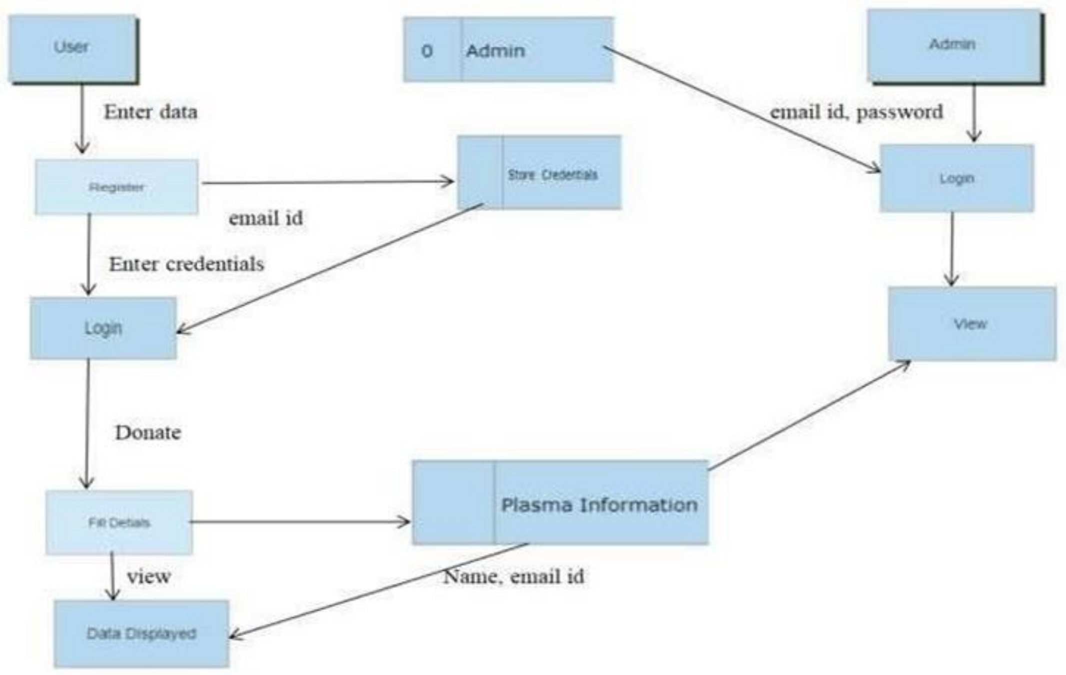
The Blood bank Management System must Alerts when the system suffers from a recoverable interruption.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFDcan depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changesthe information, and where data is stored.

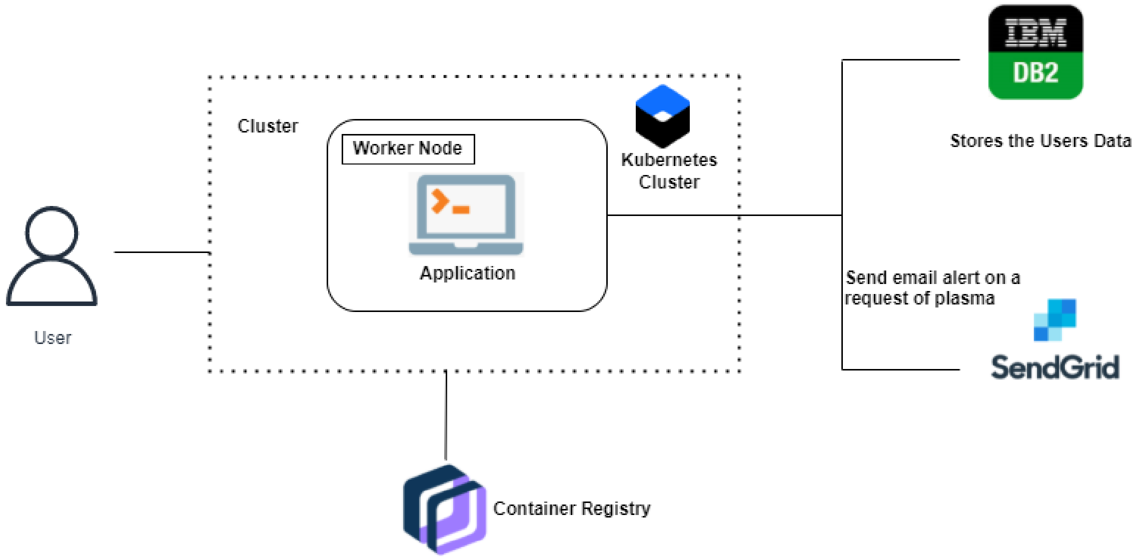
Dataflow Diagram(Plasma Donor Application):



5.2 Solution & Technical Architecture

Technical Architecture :

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



Guidelines:

1. Includeall the processes (As an application logic/ Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third partyAPI'setc.)
4. Indicate Data Storage components /services
5. Indicate interfaceto machine learningmodels (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	UserInterface	The user register and login.See the UI.	HTML, CSS, Python Flask
2.	Data maintenance	Store , maintain ,retrieve the user’s details.	MYSQL
3.	Chatbot	Clarify userqueries.	IBM Watson service
4.	Confirmation Email	Sending the confirmation email to usersthey have registered successfully.	SendGrid
5.	CloudDatabase	Cloud database to store plasma information and View Plasmainformation.	IBM DB2
6.	FileStorage	Filestorage requirements	IBM BlockStorage
7.	Infrastructure (Server/ Cloud)	To deploy the application on Local System	Kubernetes

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Python Flaskframeworks is used.	Python Flask
2.	Security Implementations	Mandatory Control(MAC) and kubernetes is used.	SHA-256, Encryptions, IAMControls, OWASP etc.
3.	Scalable Architecture	3-Tier Architecture is used.	Web server-HTML,CSS Application Server- Python Flask Database Server-IBMDB2
4.	Availability	UsingLoad Balancer to distribute networktraffic across Servers.	IBMLoad Balancer
5.	Performance	User Friendly UI. Request and Response is faster.	IBMContent Delivery Network

5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password.	I can accessmy account dashboard	High	Sprint-1
		USN-2	As a user,I will receiveconfirmation email once I have registered for the application	I can receivesuccessful message	High	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email &password	I can access into myProfile and view my dashboard	High	Sprint-1
	Dashboard	USN-4	As a user, I can login using my credentials and it will direct it to mydashboard	I can view and accesswhat are the features areprovided in dashboard	High	Sprint-1
Customer(Web user)		USN-5	As a user, I can login using my credentials and it will direct it to my dashboard	I can view and accesswhat are the features areprovided in dashboard	High	Sprint -1
Customer Care Execuive	Query	USN-6	As a user had an any query aboutthegiven requirements	I can view a query and rectify thegiven query	medium	Sprint-2
Administrator	Login	USN-7	As a admin,have credentials usingthat they can login	They can view and modify the data in database	medium	Sprint-2
	View	USN-8	As a admin I can viewplasma information	View and modify	High	Sprint-1
	Modify	USN-9	As a admin I can modify the plasma information.	Modify onlyif there is a false information/	Medium	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Project Tracker:

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story PointsCompl eted(as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov2022
Sprint-3	20	6 Days	07 Nov2022	12 Nov 2022	20	12 Nov2022
Sprint-4	20	6 Days	14 Nov2022	19 Nov 2022	20	19 Nov2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let'scalculate the team's average velocity (AV) per iteration unit (story pointsper day)

Sprint duration = 6 days
Velocity of the team = 20 points

Average velocity(AV) = Velocity/Sprint duration

AV = 20/6 = 3.34
Average Velocity = 3.34

6.2 Sprint Delivery Schedule

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story PointsCompleted(as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov2022
Sprint-3	20	6 Days	07 Nov2022	12 Nov 2022	20	12 Nov2022
Sprint-4	20	6 Days	14 Nov2022	19 Nov 2022	20	19 Nov2022

7. CODING & SOLUTIONING

7.1 Feature 1

User can search specifically for Donors

The donor list (found in your admin console under Donors > Donor List) has advanced search and filter controls to help you quickly gain insight into your donors and their donation history. From the Donor List, the Search & Filter box is at the top of the screen.

Here's a quick run-down of how each function works. Keep in mind that you can use multiple filters at the same time - selecting both a "Member" tag and a "New" checkbox will only show members who joined GivingFire in the last month.

Donor Name:

Search for donors based on their name. You can make this as specific or vague as you'd like, and search will look for the text you enter in both first names and last names. For example, if you enter "John" you might find results for both "John Doe" and "James Johnson." This function is identical to the Quick Search in the upper right corner of the navigation bar.

Donor Tags:

If your organization has added tags to certain donors, you can filter the list by those pre-existing tags. You can add more than one tag, in which case the filtered list will only show donors that have ALL of the tags. For example, you can create a filter with the "Member" and "Campus: Ballard" tags, which will only select members in the Ballard Campus (as opposed to members in other campuses, or non-members in the Ballard Campus.)

Zipcode:

You can filter donors in a certain geographic region by entering in specific zipcodes. Like the Donor Tag selection, an automatic dropdown will show pre-existing zipcodes in your database. Selecting multiple zipcodes will show donors that have ANY selected zipcode.

Badge Selection (Recurring/New/Error):

Selecting any of these checkboxes will return donors that have the specific badge. Recurring will return any donor that has an active recurring donation. New will return any donor that joined GivingFire in the past 30 days. Error will return any donor that has a active recurring donation that's currently being self-repaired by the donor.

Joined between:

This date range will return any donor that joined GivingFire or gave for the first time in between that range. You can choose to only select one date and leave the other blank -

for example, selecting Jan 1 for the Start Date and leaving the End Date blank will show all donors who joined from the beginning of the year through the present date.

Donated between:

This date range will return any donor that gave any donation in between that range. Like all of the range filters, you can opt to only enter one date and leave the other open-ended.

YTD Giving:

This number range will return any donor that's given in between that range since January 1 of the current year. Like all of the range filters, you can opt to only enter one number and leave the other open-ended.

Lifetime Giving:

This number range will return any donor that's given in between that range. Like all of the range filters, you can opt to only enter one number and leave the other open-ended.

7.2 Feature 2

Donor and Beneficiary Stories feature aims to create a sense of belonging to the community. Donors will be able to view and share personal experiences about their donation; Beneficiaries can share their experiences of receiving blood transfusion which contributed to their improved health and lives. We hypothesise that this will help Jon to be reminded of the reasons he donate blood in the first place.

Live Blood Bank Levels feature aims to bring forth the information for the constant need for blood donation. We hypothesise that this will prompt a sense of urgency for Jon to take action, and make a booking appointment.

Live Check-in Process feature aims to provide a better experience with regards to the waiting time when Jon is in the process of donation. We hypothesise that a more efficient experience will help Jon look forward to his blood donation appointments.

8. TESTING

8.1 Test Cases

Prototype 1.0 — Low-Fidelity

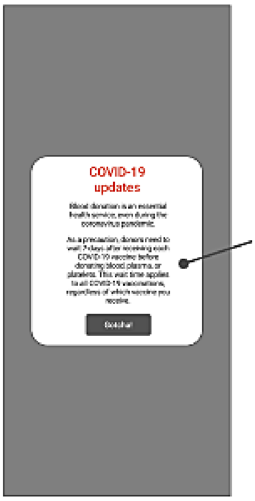
Given the current COVID-19 restrictions and our remote working locations, we created a low-fidelity prototype on Figma to allow us to work together collaboratively and to ensure consistency across testing.

Prototype 1.0 Testing

- The aim of this round of testing was to understand:How successful the changes are to existing features.
- To identify and prioritise a minimum viable product feature, steering the team in the right direction to focus our efforts building a version of the final product with just enough features that solves Jon’s problem.

Prototype 1.0 Feedback

- The Live Blood Bank Levels is a favourite feature for most users, and helped create a sense of need to take action.
- What this informs us? Testing Prototype 1.0 helped give us a clear indication that the Live Blood Bank Levels feature is the MVP. It is the most effective feature in addressing Jon’s problem of reconnecting with his initial purpose for donating.

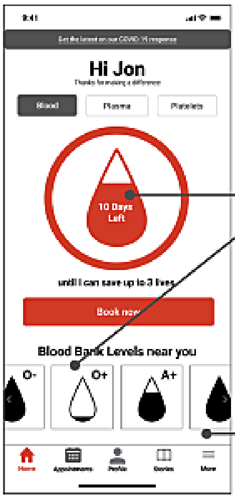

COVID-19 updates

Blood donation is an essential health service, even during the coronavirus pandemic.

As a precaution, donors need to wait 7 days after receiving each COVID-19 vaccine before donating blood plasma, or platelets. This wait time applies to all COVID-19 vaccinations, regardless of which vaccine you receive.

3 out of 5 thought the message was too long or unclear

COVID Pop-up message



Hi Jon

Thanks for making a difference

until I can save up to 3 lives

Blood Bank Levels near you

"It's positive reinforcement to show when I can book again"

3 out of 5 were confused with graphic inconsistency between days left to donate and blood bank levels

" You feel important, responsible and needed"

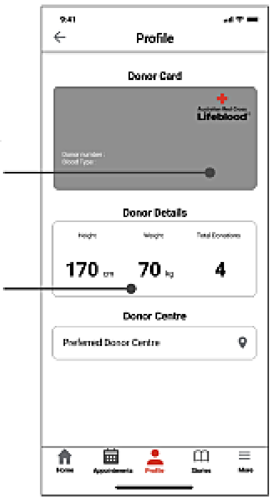
4 out of 5 admitted the blood bank levels were memorable and created a sense of urgency to take action

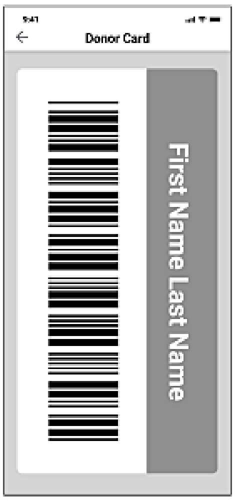
Home Page

5 out of 5 users were unsure about where to tap to check in

1 out of 5 wanted Donor Details and Total Donations to be separated

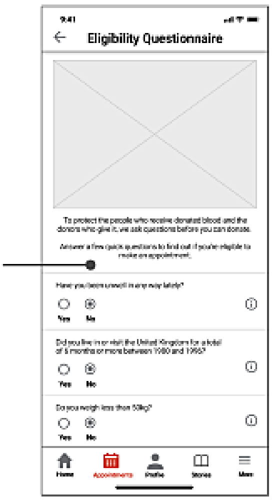
"You want to be proud of it" (Your donations)





Donor Card

5 out of 5 users found the questionnaire to be *"pretty straightforward"*



To protect the people who receive donated blood and the donors who give it, we ask questions before you can donate.

Answer a few quick questions to find out if you're eligible to make an appointment.

Have you been unwell in any way lately?

○ Yes ● No

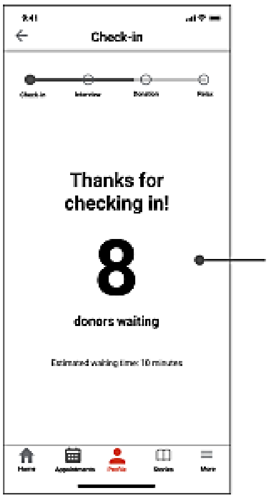
Did you live in or visit the United Kingdom for a total of 6 months or more between 1980 and 1996?

○ Yes ● No

Do you weigh less than 50kg?

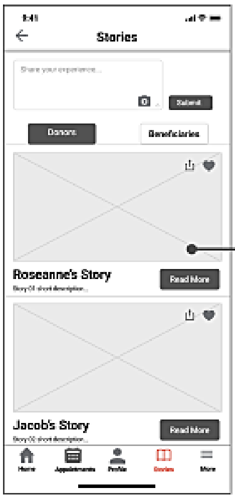
○ Yes ● No

Eligibility Questionnaire



Live Check-in Process

3 out of 5 users did not find this feature essential



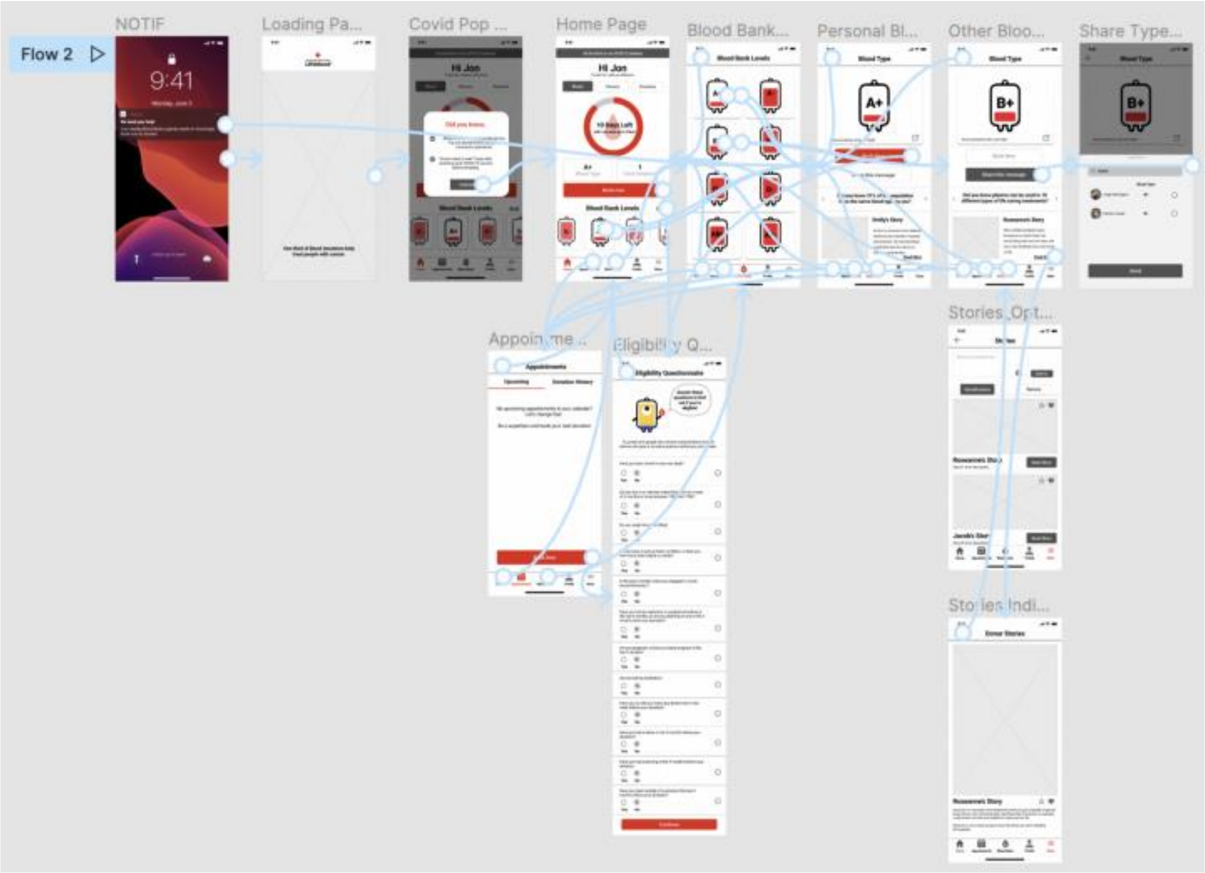
Donor & Beneficiary Stories

"It's a positive feeling seeing that I make an impact"

4 out of 5 users mentioned they would like to see other peoples' experiences, but

3 out of 5 also mentioned it would not be a regular feature to explore

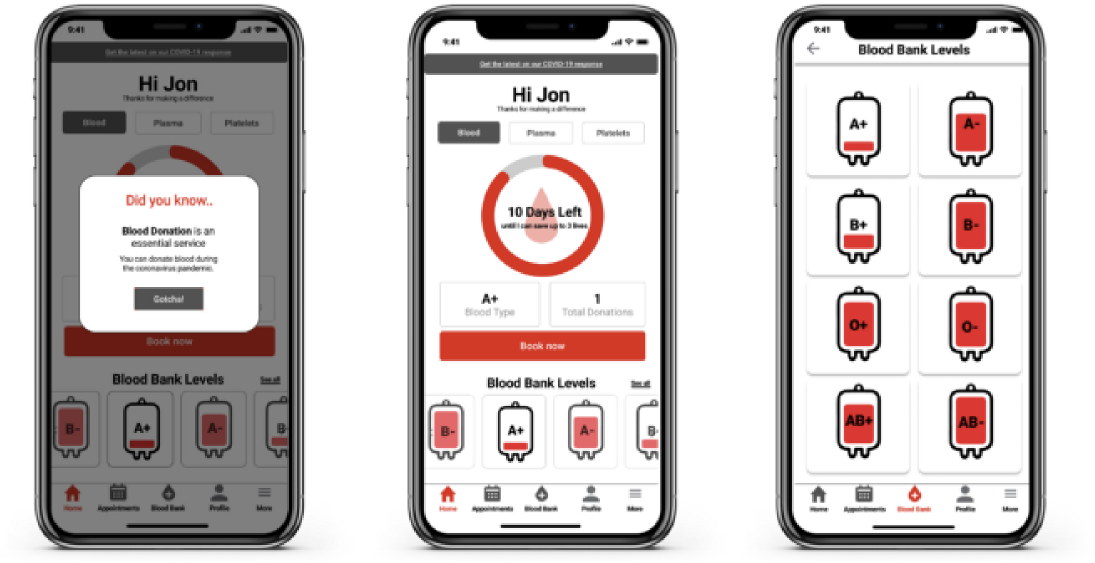
Prototype 2.0 — Mid-Fidelity — The MVP



We proceeded to create a mid-fidelity clickable prototype on Figma, this helps us to expand our testing to include the interactive flow of the design.

Prototype 2.0 Testing

To ensure consistency across testings, we chose to conduct this round of usability test with the same 5 users who tested Prototype 1.0.

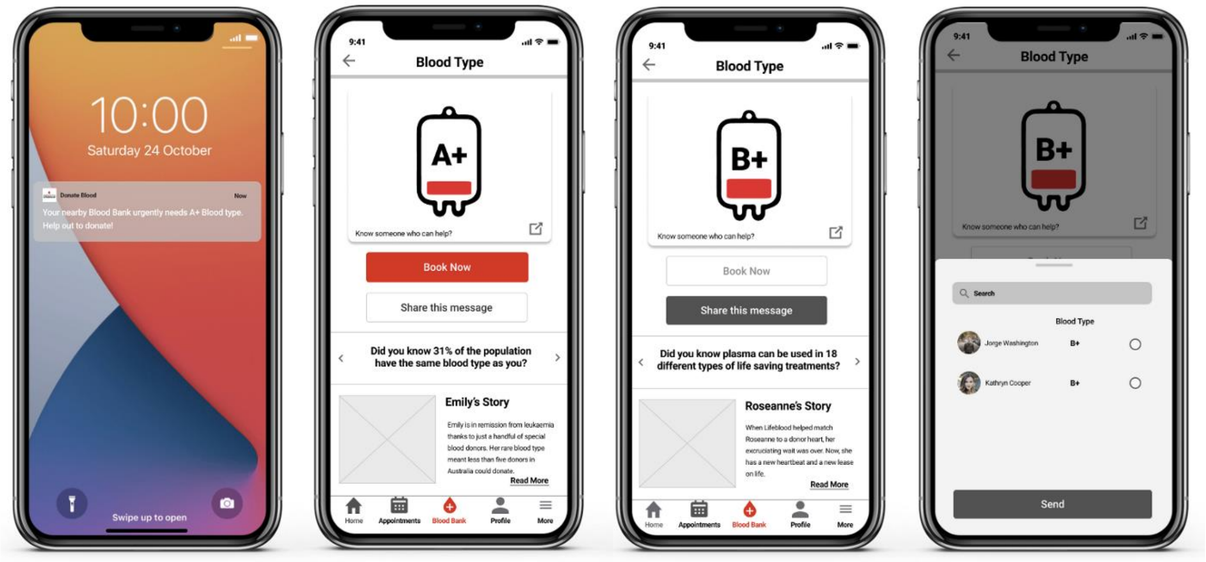


Did you know..

Blood Donation is an essential service

You can donate blood during the coronavirus pandemic.

- ➤ Upon opening the app, there’s a short and clear pop-up message about donating during COVID-19 — 4 out of 5 users found this feature informative and read the information before closing the pop-up.
- ➤ Leading on to the home page, the count down shows the number of days until Jon can donate again. Jon can also see his blood type, total number of donations, and a clear call to action button to book an appointment now — 3 out of 5 users found this count down graphic visually intuitive, easier to understand than the blood drop graphic in Prototype 1.0.
- ➤ And below the home page, we have the new feature, the Live Blood Bank Levels. From here, Jon is able to explore and see the levels of all the different blood types — 4 out of 5 users found this feature informative, and were prompted to find out more.
- ➤ Jon can click through each blood type levels to view more information.



Your nearby Blood Bank urgently needs A+ Blood type. Help out to donate!

Did you know 31% of the population have the same blood type as you?

Emily's Story

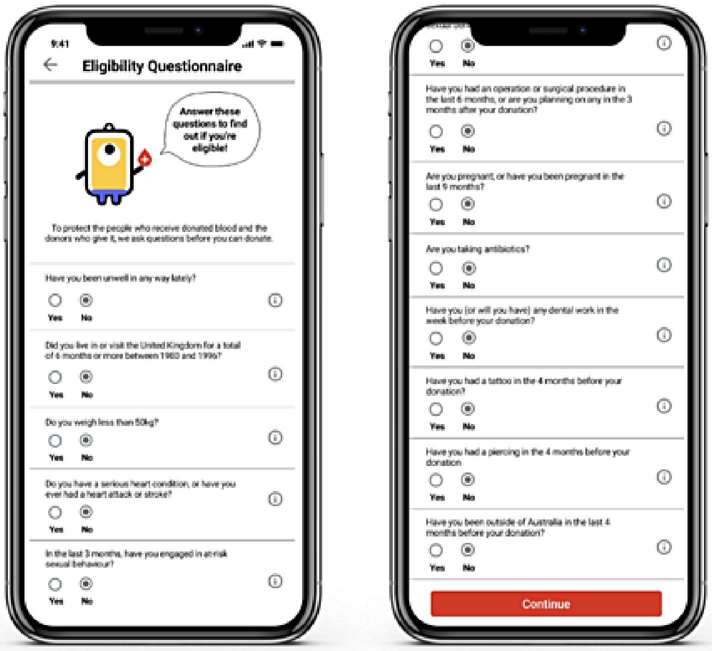
Emily is in remission from leukaemia thanks to just a handful of special blood donors. Her rare blood type meant less than five donors in Australia could donate.

Did you know plasma can be used in 18 different types of life saving treatments?

Roseanne's Story

When Lifeblood helped match Roseanne to a donor heart, her excruciating wait was over. Now, she has a new heartbeat and a new lease on life.

- ➢ Another way to access the Live Blood Bank Levels feature, is through a push notification when Jon’s blood type is low and in need. By clicking the notification, he can directly access his blood type page, and book an appointment — 3 out of 5 users felt a sense of urgency to donate.
- ➢ On the blood type page, Jon can learn more about the current blood bank levels, read facts and explore stories about how his donation can help others — 3 out of 5 users stated that they enjoy reading the facts; 2 out of 5 users found the stories to be relevant in this section in comparison to a separate section in Prototype 1.0.
- ➢ For the other blood type pages, there is the ability to share the urgent need message with others who might be able to help — 3 out of 5 users expressed that they will share the message.



9:41
Eligibility Questionnaire
Answer these questions to find out if you're eligible!
Continue

To protect the people who receive donated blood and the donors who give it, we ask questions before you can donate.

Have you been unwell in any way lately?
○ Yes ● No

Did you live in or visit the United Kingdom for a total of 6 months or more between 1980 and 1996?
○ Yes ● No

Do you weigh less than 50kg?
○ Yes ● No

Do you have a serious heart condition, or have you ever had a heart attack or stroke?
○ Yes ● No

In the last 3 months, have you engaged in at-risk sexual behaviour?
○ Yes ● No

○ Yes ● No

Have you had an operation or surgical procedure in the last 6 months, or are you planning on any in the 3 months after your donation?
○ Yes ● No

Are you pregnant, or have you been pregnant in the last 9 months?
○ Yes ● No

Are you taking antibiotics?
○ Yes ● No

Have you (or will you have) any dental work in the week before your donation?
○ Yes ● No

Have you had a tattoo in the 4 months before your donation?
○ Yes ● No

Have you had a piercing in the 4 months before your donation
○ Yes ● No

Have you been outside of Australia in the last 4 months before your donation?
○ Yes ● No

- And finally, the updated Eligibility Questionnaire, which has a clear Yes/No radio buttons. From the previous round of testing, all users said it is “pretty straightforward”, visual designs were made to this mid-fidelity prototype.

8.2 User Acceptance Testing

User tests

Before launching the app into the market, we knew we needed to test its usability and performance. We wanted to be sure that we are giving users the product that will work for their best interest, and will help them in donating blood. Besides, we also believe that being engaged in the process of app delivery impacts users’ loyalty. Satisfied users are more likely to recommend the app to friends, which could also support the major goal of the app we always kept in mind. We were aware that the more people will have and like the app, the more blood will be donated.

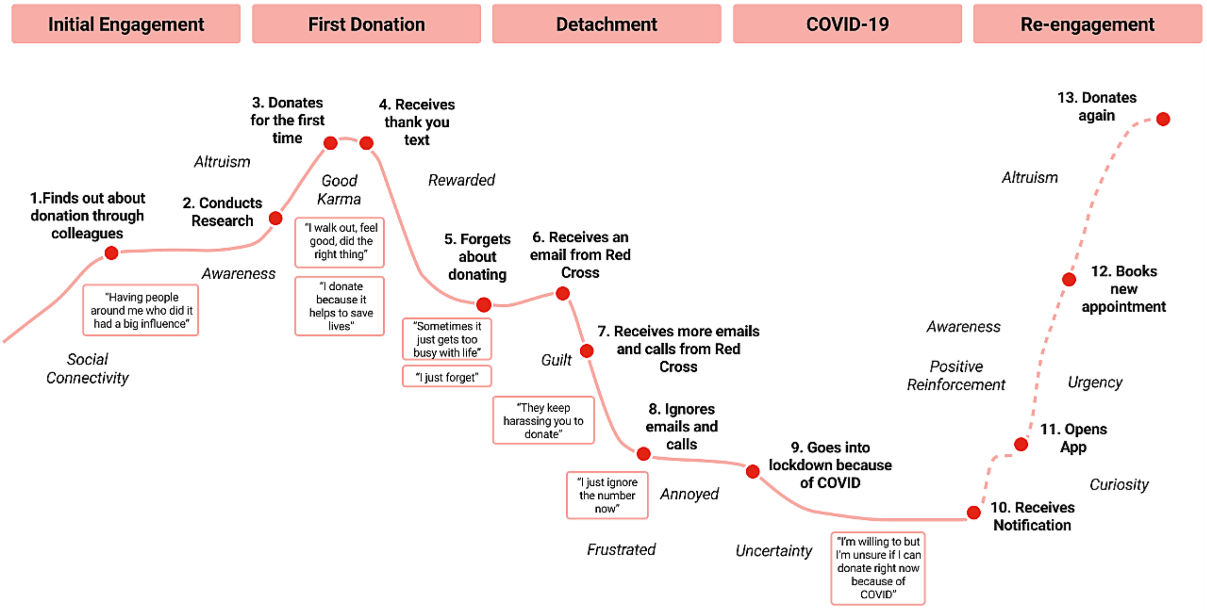
Our testing goal was to ask the real users (donors) how they feel about the app, investigate the potential usability problems, and then introduce changes if needed. The big plus was to test the app on actual devices, so we could identify real problems and gather constructive feedback. Again, the good relation with the client was the helpful aspect to get to the real users – blood donors. We introduced the app in the store in developer mode and conducted the semi-closed tests on future app users. As a result of the tests, we made changes to improve usability and user experience.

9. RESULTS

9.1 Performance Metrics

The Result

Using a journey map to illustrate how our solution helps to solve Jon's problem, a Re-engagement phase for current and past donors:


Initial Engagement
First Donation
Detachment
COVID-19
Re-engagement
1.Finds out about donation through colleagues
"Having people around me who did it had a big influence"
Social Connectivity
2. Conducts Research
Awareness
Altruism
3. Donates for the first time
Good Karma
"I walk out, feel good, did the right thing"
"I donate because it helps to save lives"
4. Receives thank you text
Rewarded
5. Forgets about donating
"Sometimes it just gets too busy with life"
"I just forget"
6. Receives an email from Red Cross
Guilt
7. Receives more emails and calls from Red Cross
"They keep harassing you to donate"
8. Ignores emails and calls
"I just ignore the number now"
Annoyed
Frustrated
9. Goes into lockdown because of COVID
Uncertainty
"I'm willing to but I'm unsure if I can donate right now because of COVID"
10. Receives Notification
Curiosity
11. Opens App
Urgency
Awareness
Positive Reinforcement
12. Books new appointment
Altruism
13. Donates again

By introducing this new Live Blood Bank Levels feature, we are able to re-engage our user, Jon (and others like him) by evoking feelings of curiosity, urgency and awareness. Upon receiving a push notification our MVP aims to prompt Jon to open the app and book an appointment to donate once again, experiencing initial feelings of altruism and emotionally rewarded.

Eventually, we hope that Jon (and others like him) will feel encouraged to become a regular donor, where the re-engagement phase replaces the detachment phase that current and past donors face, addressing both the problem statement and Red Cross's business goal of increasing the number of repeat donors.

10. ADVANTAGES & DISADVANTAGES

Benefits of Donating Plasma

- Earn Up to $4,000 per Year. What attracts many people to plasma donation is the fact that you can earn a substantial amount of money every time you donate. ...
- Make an Impact. ...
- Boost Your Mood. ...
- Maintain a Healthy Diet. ...
- Reduce Cholesterol Levels. ...
- Lower Blood Pressure.

Disadvantages

- weakness.
- dizziness.
- feeling faint.
- lightheadedness.
- nausea.
- bleeding from the needle prick.
- bleeding under the skin or bruising.

11. CONCLUSION

Plasma donor application was developed to search for blood donors, using the Scrum methodology, Google maps service to locate the donor and blood donation centers in real time.

After the implementation of the project, there was a significant increase in the number of blood donors by 39.58%, as well as a reduction of 53.2% in the time required to search for blood donors.

The results show that donors from different cities in the interior of the country such as Piura, Trujillo, Chimbote, Cuzco and Tarapoto, representing 39.58%, however, it is important to note that the city of Lima continues to concentrate the largest number of blood donors, equivalent to 60.42% according to the case study.

The scientific contribution of this article is fundamental for the development of future work related to blood donation campaigns. With the data collected through the mobile application, it is possible to perform analytical processing, develop strategies with reward incentives to motivate blood donation, and add new functionalities so that doctors and patients can communicate.

Finally, it was demonstrated that the solution developed, manages to increase the number of blood donors, reduce the time to get donors, has greater population reach and in turn has access to the information provided by the donor, and with this to perform actions in favor of those concerned.

12. FUTURE SCOPE

Scope of the Project

The system functions and features of our system will include the following:

Registration

This function allows the donor and administrator to register as a user to interact with the system. The system requires the user to login before viewing and editing any information.

Data is input by the Administrators

The donor's information and donation records can be sent from the hospital to the administrator by calling or e-mail. The administrator is responsible for keying the received data into the system. Recording donation records. The system can record data of whole blood which is sent from the hospital.

Manage blood inventory

The system uses a First-In-First-Out stock management, where the blood stock that is checked-in to the system first will be the first one given to the hospital when requested. When the blood stock is expired, the administrator is responsible for removing the stock from the inventory and updating the system.

Blood requests

The hospital can request blood via e-mail and by calling to the blood bank.

Notify by E-mail

The donor's account and generated password will be sent via e-mail, following by their blood result of the previous donation sent in a separated e-mail. Hospitals can also receive e-mail responding to their requested blood whether it is available in our stock or not.

Summary report

The system can generate a report to summarize all records including blood donation, blood requests and blood stock for the administrator.

13. APPENDIX

Source Code

login.html

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
<meta charset="UTF-8">
<title>IBM Donor App</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
        type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
        rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
<style>
.login{
top: 20%;
}
</style>
</head>
<body>
<div class="header">
<div>Plasma Donor App</div>
<ul>
<li><a href="/registration">Register</a></li>
<li><a class="active" href="/login">Home</a></li>
</ul>
</div>
<div class="login" >
<div>
</div>
```

```
<!-- Main Input For Receiving Query to our ML -->

<form action="{{ url_for('loginpage')}}"method="post">

<input type="text" name="username" placeholder="Enter UserName" required="required"
        style="color:black" />

<input type="password" name="password" placeholder="Enter Password" required="required"
        style="color:black" />

<button type="submit" class="btn btn-primary btn-block btn-large">Login</button>

</form>

<br>

<br>

<div style="color:black">

</div>

</div>

</body>

</html>
```

register.html

```
<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

<meta charset="UTF-8">

<title>IBM Plasma Donor App</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet'
        type='text/css'>

<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

<style>

.register{

top: 20%;

}

</style>

</head>

<body>
```

```
<div class="header">

<div>Plasma Donor App</div>

<ul>

<li><a class="active" href="/login">Home</a></li>

</ul>

</div>

<div class="login">

<!-- Main Input For Receiving Query to our ML -->

<form action="{{ url_for('register')}}"method="post">

<input type="text" name="username" placeholder="Enter Your Name" required="required"
        style="color:black"/>

<input type="email" name="email" placeholder="Enter Email" required="required"
        style="color:black"/>

<input type="text" name="phone" placeholder="Enter 10-digit mobile number" required="required"
        style="color:black"/>

<input type="city" name="city" placeholder="Enter Your City Name" required="required"
        style="color:black"/>

<select name="infect">

<option value="select" selected>Select COVID infection status</option>

<option value="infected">Infected</option>

<option value="uninfected">Uninfected</option>

</select>

<select name="blood">

<option value="select" selected>Choose your blood group</option>

<option value="O Positive">O Positive</option>

<option value="A Positive">A Positive</option>

<option value="B Positive">B Positive</option>

<option value="AB Positive">AB Positive</option>

<option value="O Negative">O Negative</option>

<option value="A Negative">A Negative</option>

<option value="B Negative">B Negative</option>

<option value="AB Negative">AB Negative</option>

</select>

<input type="password" name="password" placeholder="Enter Password" required="required"
        style="color:black"/>

<button type="submit" class="btn btn-primary btn-block btn-large">Register</button>

</form>

<br><br>
```

```
<div style="color:black">

</div>

</div>

</body>

</html>
```

request.html

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
<meta charset="UTF-8">
<title>IBM Plasma Donor App</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
        rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
<style>
.login{
top: 20%;
}
</style>
</head>
<body>
<div class="header">
<div>Plasma Donor App</div>
<ul>
<li><a href="/requester">Request</a></li>
<li><a href="/registration">Register</a></li>
<li><a class="active" href="/dashboard">Home</a></li>
</ul>
</div>
<div class="login">
<div>
</div>
<!-- Main Input For Receiving Query to our ML -->
<form action="{{ url_for('requested')}}"method="post">
<input type="text" name="name" placeholder="Enter Name" required="required"
        style="color:black" />
<input type="email" name="email" placeholder="Enter Email" required="required"
        style="color:black"/>
<input type="text" name="phone" placeholder="Enter 10-digit mobile number"
        required="required" style="color:black"/>
<select name="bloodgrp">
<option value="select" selected>Choose your blood group</option>
<option value="O Positive">O Positive</option>
<option value="A Positive">A Positive</option>
<option value="B Positive">B Positive</option>
<option value="AB Positive">AB Positive</option>
<option value="O Negative">O Negative</option>
```

```
<option value="A Negative">A Negative</option>
<option value="B Negative">B Negative</option>
<option value="AB Negative">AB Negative</option>
</select>
<textarea rows="4" placeholder="Enter the address" required="required" style="color:black"
            name="address"></textarea>
<button type="submit" class="btn btn-primary btn-block btn-large">Submit the
            request</button>
</form>
<br><br>
<div style="color:black">
</div>
</div>
</body>
</html>
```

dashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>IBM Plasma Donar App</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js">
</script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<style>
            .big{
            top:70;
            background-color:white;
            margin-top:80px;
            margin-left:550px;
            margin-right:550px;
            height:200px;
            border-radius: 25px;
            border: 3px solid #4a77d4;
            box-shadow: 6px 8px 4px grey;
            text-align:center;
            }
            .row{

            height:150px;

            }
            .col{
                    margin:10px;
                    margin-left:50px;
                    margin-right:50px;
```

```
                border-radius: 25px;
                border: 1px solid #4a77d4;
                box-shadow: 0px 8px 4px grey;
                text-align:center;
        }
        .ext{
        margin-top:25px;
        line-height:40px;
        }
        .ext1{
        margin-top:40px;
        line-height:50px;
        font-size:25px;
        color:#f95450;
        }
</style>
<body>
<div class="container-fluid">
<div class="header">
<div><b>Plasma Donar App</b></div>
<ul>
<li><a href="/requester">Request</a></li>
<li><a class="active" href="/logout">Logout</a></li>
</ul>
</div>
<br>
<div class="big">
<div class="box">
<div class="ext1"><font size="20px">{{b['1']}}</font><br><b>Donors</b></div>
</div>
</div>
<br>
<div class="row">
<div class="col" >
<div class="ext">{{b['2']}}<br><b>O Positive</b></div>
</div>
<div class="col" >
<div class="ext">{{b['3']}}<br><b>A Positive</b></div>
</div>
<div class="col" >
<div class="ext">{{b['4']}}<br><b>B Positive</b></div>
</div>
<div class="col" >
<div class="ext">{{b['5']}}<br><b>AB Positive</b></div>
</div>
</div>
<br>
<div class="row">
<div class="col" >
<div class="ext">{{b['6']}}<br><b>O Negative</b></div>
</div>
<div class="col" >
<div class="ext">{{b['7']}}<br><b>A Negative</b></div>
</div>
<div class="col" >
```

```
<div class="ext">{{b['8']}}<br><b>B Negative</b></div>
</div>
<div class="col" >
<div class="ext">{{b['9']}}<br><b>AB Negative</b></div>
</div>
</div>
<div style="height:200px"></div>
</div>
</body>
</html>
```

style.css

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn {
  display: inline-block;
  *display: inline;
  *zoom: 1; padding:
  4px 10px 4px;
  margin-bottom: 0;
  font-size: 13px;
  line-height: 18px;
  color: #333333;
  text-align: center;
  text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
  vertical-align: middle;
  background-color: #d70c0c;
  background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));
  background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -o-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: linear-gradient(top, #ffffff, #e6e6e6);
  background-repeat: repeat-x;
  filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff,
  endColorstr=#e6e6e6, GradientType=0);
  border-color: #e6e6e6 #e6e6e6 #e6e6e6;
  border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);
  border: 1px solid #e6e6e6;
  -webkit-border-radius: 4px;
  -moz-border-radius: 4px;
  border-radius: 4px;
  -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  cursor: pointer; *margin-left: .3em;
  }

.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }

.btn-large {
  padding: 9px 14px;
  font-size: 15px;
  line-height: normal;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  border-radius: 5px;
  }

.btn:hover {
  color: #333333;
  text-decoration: none;
  background-color: #e6e6e6;
  background-position: 0 -15px;
  -webkit-transition: background-position 0.1s linear;
  -moz-transition: background-position 0.1s linear;
  -ms-transition: background-position 0.1s linear;
  -o-transition: background-position 0.1s linear;
  transition: background-position 0.1s linear;
  }

.btn-primary, .btn-primary:hover {
  text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);
  color: #ffffff;
  }

.btn-primary.active { color: rgba(255, 255, 255, 0.75); }

.btn-primary {
  background-color: #d70c0c;
  background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4));
  background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: -o-linear-gradient(top, #6eb6de, #4a77d4);
  background-image: linear-gradient(top, #6eb6de, #4a77d4);
  background-repeat: repeat-x;
  filter: progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de, endColorstr=#4a77d4, GradientType=0);
  border: 1px solid #3762bc;
  text-shadow: 1px 1px 1px rgba(0,0,0,0.4);
  box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5);
  }

.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled] {
  filter: none;
  background-color: #d70c0c
  }

.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box; }

html { width: 100%; height:100%; overflow:hidden; }

body {
  width: 100%;
  height:100%;
  font-family: 'Open Sans', sans-serif;
  color: #000000;
  font-size: 18px;
  text-align:center;
  letter-spacing:1.2px;


}
.header {
                        top:0;
                        margin:0px;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        background: #d44a4a;
                        color: black;
                        box-shadow: 0px 8px 4px grey;
                        overflow: hidden;
                        padding: 15px;
                        font-size: 1.5vw;
                        width: 100%;
                        text-align: center;
              }
.login {
  position: absolute;
  top: 70%;
  left: 50%;
  margin: -25px 0 0 -150px;
  width:400px;
  height:400px;
}

.header div { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; float:left; padding-left:150px;}

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  padding-right:150px;
  overflow: hidden;
 }


li {
  float: right;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 0px 15px;
  text-decoration: none;
}


input {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(255,255,255,255);
  border: none;
  outline: none;
  padding: 10px;
  font-size: 13px;
  color: black;
  text-shadow: black;
  border: 1px solid rgba(0,0,0,0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
  -webkit-transition: box-shadow .5s ease;
  -moz-transition: box-shadow .5s ease;
  -o-transition: box-shadow .5s ease;
  -ms-transition: box-shadow .5s ease;
  transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }


textarea {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(255,255,255,255);
  border: none;
  outline: none;
  padding: 10px;
  font-size: 13px;
  color: black;
  text-shadow: black;
  border: 1px solid rgba(0,0,0,0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px   rgba(255,255,255,0.2);
  -webkit-transition: box-shadow .5s ease;
  -moz-transition: box-shadow .5s ease;
  -o-transition: box-shadow .5s ease;
  -ms-transition: box-shadow .5s ease;
  transition: box-shadow .5s ease;
}
textarea:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }

select {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(255,255,255,255);
  border: none;
  outline: none;
  padding: 10px;
  font-size: 13px;
  color: #000000;
  text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
  border: 1px solid rgba(0,0,0,0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
  -webkit-transition: box-shadow .5s ease;
  -moz-transition: box-shadow .5s ease;
  -o-transition: box-shadow .5s ease;
  -ms-transition: box-shadow .5s ease;
  transition: box-shadow .5s ease;
}
```

app.py

```
#code for login and register
from distutils.log import debug
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__)
app.secret_key = 'a'
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-
                0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32459;Secu
                rity=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=ygg09863;PWD=d
                QXLECjtaCqXsJUt;","","")
@app.route('/')
@app.route('/login')
def login():
 return render_template('login.html')
@app.route('/loginpage',methods=['GET', 'POST'])
def loginpage():
 global userid
 msg = ''
 if request.method == 'POST' :
     username = request.form['username']
     password = request.form['password']
     sql = "SELECT * FROM donors WHERE username =? AND password=?"
     stmt = ibm_db.prepare(conn, sql)
     ibm_db.bind_param(stmt,1,username)
     ibm_db.bind_param(stmt,2,password)
     ibm_db.execute(stmt)
     account = ibm_db.fetch_assoc(stmt)
     print (account)
     if account:
         session['loggedin'] = True
         session['id'] = account['USERNAME']
         userid=  account['USERNAME']
         session['username'] = account['USERNAME']
         msg = 'Logged in successfully !'
         return redirect(url_for('dash'))
     else:
        msg = 'Incorrect username / password !'
     return render_template('login.html', msg = msg)
@app.route('/registration')
def home():
return render_template('register.html')
@app.route('/register',methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' :
       username = request.form['username']
       email = request.form['email']
       password = request.form['password']
       phone = request.form['phone']
       city = request.form['city']
       infect = request.form['infect']
       blood = request.form['blood']
       sql = "SELECT * FROM donors WHERE username =?"
       stmt = ibm_db.prepare(conn, sql)
       ibm_db.bind_param(stmt,1,username)
       ibm_db.execute(stmt)
       account = ibm_db.fetch_assoc(stmt)
       print(account)
       if account:
          msg = 'Account already exists !'
       elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
          msg = 'Invalid email address !'
       elif not re.match(r'[A-Za-z0-9]+', username):
          msg = 'name must contain only characters and numbers !'
       else:
          insert_sql = "INSERT INTO  donors VALUES (?, ?, ?, ?, ?, ?, ?)"
          prep_stmt = ibm_db.prepare(conn, insert_sql)
          ibm_db.bind_param(prep_stmt, 1, username)
          ibm_db.bind_param(prep_stmt, 2, email)
          ibm_db.bind_param(prep_stmt, 3, password)
          ibm_db.bind_param(prep_stmt, 4, city)
          ibm_db.bind_param(prep_stmt, 5, infect)
          ibm_db.bind_param(prep_stmt, 6, blood)
          ibm_db.bind_param(prep_stmt, 7, phone)
          ibm_db.execute(prep_stmt)
          msg = 'You have successfully registered !'
      elif request.method == 'POST':
      msg = 'Please fill out the form !'
return render_template('register.html', msg = msg)
```

app.py

Aim: As a user, I can log into the application by entering email & password

```
from distutils.log import debug
from sendgridmail import sendmail
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
import os
from dotenv import load_dotenv
```

```
load_dotenv()


app = Flask(__name__)

app.secret_key = 'a'

conn=ibm_db.connect(os.getenv('DB_KEY'),"","")

@app.route('/')
@app.route('/login')
def login():
    return render_template('login.html')


@app.route('/loginpage',methods=['GET', 'POST'])
def loginpage():
    global userid
    msg = ''

    if request.method == 'POST' :
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM donors WHERE username =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin'] = True
            session['id'] = account['USERNAME']
            userid=  account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            sendmail(account['EMAIL'],'Plasma donor App login','You are successfully logged in!')
            return redirect(url_for('dash'))
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

@app.route('/registration')
def home():
    return render_template('register.html')

@app.route('/register',methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        phone = request.form['phone']
        city = request.form['city']
        infect = request.form['infect']
        blood = request.form['blood']
        sql = "SELECT * FROM donors WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        else:
            insert_sql = "INSERT INTO  donors VALUES (?, ?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.bind_param(prep_stmt, 4, city)
            ibm_db.bind_param(prep_stmt, 5, infect)
            ibm_db.bind_param(prep_stmt, 6, blood)
            ibm_db.bind_param(prep_stmt, 7, phone)

            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered !'
            sendmail(email,'Plasma donor App Registration','You are successfully Registered
             {}!'.format(username))
   elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)

@app.route('/dashboard')
def dash():
    if session['loggedin'] == True:
        sql = "SELECT COUNT(*), (SELECT COUNT(*) FROM DONORS WHERE blood= 'O
Positive'), (SELECT COUNT(*) FROM DONORS WHERE blood='A Positive'), (SELECT
COUNT(*) FROM DONORS WHERE blood='B Positive'), (SELECT COUNT(*) FROM
DONORS WHERE blood='AB Positive'), (SELECT COUNT(*) FROM DONORS WHERE
blood='O Negative'), (SELECT COUNT(*) FROM DONORS WHERE blood='A Negative'),
(SELECT COUNT(*) FROM DONORS WHERE blood='B Negative'), (SELECT COUNT(*)
FROM DONORS WHERE blood='AB Negative') FROM donors"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        return render_template('dashboard.html',b=account)
    else:
        msg = 'Please login!'
        return render_template('login.html', msg = msg)

@app.route('/requester')
def requester():
    if session['loggedin'] == True:
        return render_template('request.html')
    else:
        msg = 'Please login!'
        return render_template('login.html', msg = msg)

@app.route('/requested',methods=['POST'])
def requested():
    bloodgrp = request.form['bloodgrp']
    address = request.form['address']
    name=  request.form['name']
    email=  request.form['email']
    phone= request.form['phone']
    insert_sql = "INSERT INTO  requested VALUES (?, ?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, bloodgrp)
    ibm_db.bind_param(prep_stmt, 2, address)
    ibm_db.bind_param(prep_stmt, 3, name)
    ibm_db.bind_param(prep_stmt, 4, email)
    ibm_db.bind_param(prep_stmt, 5, phone)
    ibm_db.execute(prep_stmt)
    sendmail(email,'Plasma donor App plasma request','Your request for plasma is recieved.')
    return render_template('request.html', pred="Your request is sent to the concerned people.")


@app.route('/logout')

def logout():
   session.pop('loggedin', None)
   session.pop('id', None)
   session.pop('username', None)
   return render_template('login.html')

if __name__ == '__main__':
   app.run(host='0.0.0.0',debug='TRUE')
```

sendgridmail.py

```
# using SendGrid's Python Library
# https://github.com/sendgrid/sendgrid-python
import os
from dotenv import load_dotenv

load_dotenv()
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

def sendmail(usermail,subject,content):
   message =
   Mail(from_email='maryada@student.tce.edu',to_emails=usermail,subject=subject,html_conte
   nt='<strong> {} </strong>'.format(content))
    try:
        sg = SendGridAPIClient(os.getenv('SENDGRID_API_KEY'))
```

```
response = sg.send(message)
print(response.status_code)
print(response.body)
print(response.headers)
except Exception as e:
print(e.message)
```

GitHub & Project Demo Link

GITHUB