PLASMA DONOR APPLICATION

A PROJECT REPORT

Submitted by

Team Lead	**:**	**Udhayakumar L**
Team Member 1	**:**	**Vasudevan R**
Team Member 2	**:**	**Naveen Akshi D**
Team Member 3	**:**	**Vishnu S**

for the course

HX8001 – Professional Readiness for Innovation, Employment and Entrepreneurship

in

INFORMATION TECHNOLOGY

DMI COLLEGE OF ENGINEERING

PALANCHUR, CHENNAI – 600 123.

ANNA UNIVERSITY : CHENNAI 600 025

NOV/DEC-2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"PLASMA DONOR APPLICATION"** is the bonafide work of L.**Udhayakumar (210519205053), R.Vasudevan (210519205054), D.Naveen Akshi (210519205033) & S.Vishnu (210519205058)** who carried out the project work under my supervision.

SIGNATURE

Dr. B. Muthu Kumar M.E., Ph.D
HEAD OF THE DEPARTMENT
Department of Information
Technology

DMI College of Engineering
Palanchur, Chennai – 600 123.

SIGNATURE

J. Jennifer M.Tech
MENTOR
ASSISTANT PROFESSOR
Department of Information
Technology

DMI College of Engineering
Palanchur, Chennai – 600 123.

ABSTRACT

A plasma is a liquid portion of the blood, over 55% of human blood is plasma. Plasma is used to treat various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish antibodies that fights the infection.

In this project plasma donor application is being developed by using IBM Cloud, Flask. The services used are IBM Object Storage, IBM DB2, IBM Kubernets, IBM Watson, Python Flask, Docker, Sendgrid. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors.

Keywords: IBM Object Storage, IBM DB2, IBM Kubernets, IBM Watson, Python Flask, Docker, Sendgrid, Covid 19, Plasma Donor.

TABLE OF CONTENTS

S.NO	TITLE
1	**INTRODUCTION**
	1.1 Project Overview
	1.2 Purpose
2	**LITERATURE SURVEY**
	2.1 Existing problem
	2.2 References
	2.3 Problem Statement Definition
3	**IDEATION & PROPOSED SOLUTION**
	3.1 Empathy Map Canvas
	3.2 Ideation & Brainstorming
	3.3 Proposed Solution
	3.4 Problem Solution fit
4	**REQUIREMENT ANALYSIS**
	4.4 Functional requirement
	4.5 Non-Functional requirements
5	**PROJECT DESIGN**
	5.1 Data Flow Diagrams
	5.2 Solution & Technical Architecture
	5.3 User Stories
6	**PROJECT PLANNING & SCHEDULING**
	6.1 Sprint Planning & Estimation
	6.2 Sprint Delivery Schedule

6.3 Report from JIRA

7 CODING & SOLUTIONING

7.1 SendGrid

7.2 Database Schema

8 TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9 RESULTS

9.1 Performance Metrics

10 ADVANTAGES & DISADVANTAGES

11 CONCLUSION

12 FUTURE SCOPE

13 APPENDIX

13.1 Source Code

13.2 GitHub & Project Demo Link

CHAPTER 1: INTRODUCTION

1.1 Project Overview:

Patients with severe liver disease or numerous clotting factor deficits, as well as those who have undergone trauma, burns, or shock, frequently get plasma. The patient's blood volume is increased as a result, which can aids in blood coagulation and helps to prevent shock. The number of people with Covid-19 infection has increased, as has the demand for the plasma of patients who have recovered. The antibodies that are already in our systems can aid someone in overcoming the infection.

Plasma donation saves lives, and donors' and blood/plasma facilities' communicationis key to this. Smart apps are increasingly viewed as a crucial communication tool, and if they are created with the users' requirements and preferences in mind, plasma donation could make the best use of them.

1.2 Purpose:

In our opinion we intend to create an application that is user-friendly for people who require plasma or who wish to donate plasma to anyone who is in need.

However, during design and development, areas of concern including privacy and secrecy should be taken into account. Age was found to be a contributing factor that might reduce donors' propensity to use apps. 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.

CHAPTER 2: LITERATURE SURVEY

2.1 Existing problem:

In most of the existing plasma donor application then system is closed for general plasma donation and mainly focused on COVID-19 patients for plasma donation, the android mobile user will not be able to insert or view details if the server goes down and a disadvantage of single point of failure. Most of the user details remains unverified and it's difficult to track the fake users. The user interface of the application is not being user friendly and the user must have a device with android operating system with an active internet connection to interact with this application.

2.2 Reference:

Paper 1: Blood donor app usage behavior and perceptions: Considerations for a blood donation app (Andrea Potgieter, May 2022)

This article aimed to determine whether South African blood donor app usage behavior and perceptions were conducive to introduce a blood donation app, and what these behaviors and perceptions could reveal, to support South African Blood Donation Organizations in their recruitment and engagement endeavors. The research problem discussed in this article sought to highlight the app usage behavior of blood donors, and their perceptions about a proposed blood donation app. forming part of a larger sequential mixed-methods study, the data presented in this article were gathered through a quantitative online questionnaire involving 2154 South Africans respondents. The value of this research lies in the insight gained into the behavior and perceptions of South African blood donors, which can inform the conceptualization and design of a blood donation app, thereby improving its efficacy and subsequently supporting the strategy of employing such a technology to increase blood donation.

Paper 2: Evaluation of the Wateen App in the Blood-Donation Process in Saudi Arabia (Tourkiah Alessa, April 2022)

The aim of this research was to evaluate the usability, user satisfaction and perceived usefulness of this blood-donation app in Saudi Arabia. A mixed-method study was conducted comprising a quantitative questionnaire with donor and qualitative semi-structured interviews with healthcare professionals. Descriptive analysis was used for the quantitative data and a thematic approach for the qualitative data. Quantitative data analysis was conducted using SPSS software package 19 to calculate descriptive statistics. This blood-donation app is highly usable and acceptable among donors and healthcare professionals in Saudi Arabia, offering several benefits. Some accessibility issues were identified, along with possibilities for improving accessibility and expanding the app's functionality.

Paper 3: Location-based Mobile Application for Blood Donor Search (Fernando Alex Sierra-Linan, January 2022)

The research proposes the development of a location-based mobile application for blood donor search (DONAPE), for which the mobile application provides a direct location-based channel between blood seekers and blood donation centers. Achieving to increase the number of donors, improve the place of origin (geographical location) of donors and improve the search time. They chose to use the agile Scrum method to develop the project prototype. This method has 5 phases: initiation, planning and estimation, implementation, review and retrospective and launch, for the development of this project. In web and mobile applications were developed to manage blood donation, allowing to register, schedule, receive notifications and access information, synchronizing blood donation centers with emergency centers, to verify the availability of blood needed and to send a request to the nearest blood donation center.

Paper 4: A Cross-Platform Blood Donation Application with a Real-Time, Intelligent, and Rational Recommendation System (Rashik Rahman, September 2021)

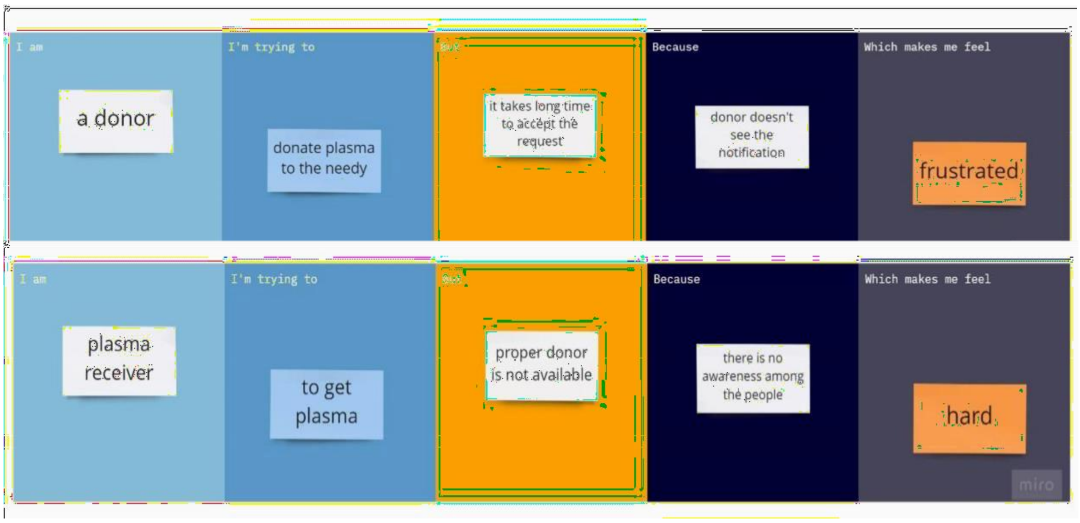
In this research work, they have designed a real-time, intelligent, and rational recommendation system using sentiment analysis of the user's feedback, response rate of the donor, and the current geo-location information and finally develop a cross-platform application for blood collection and distribution system. To process and generate features from the user feedback, they have designed a Bi-directional LSTM-based deep learning model. They chose the flutter framework to develop our cross platform applications. Firebase, a Google platform for mobile and web applications, has been used in the proposed application for authentication man. The quality of the recommendation of the potential donors has significantly improved. Moreover, they have conducted rigorous requirement analysis from real users and evaluated the performance of the application through both indoor and outdoor testing.

Paper 5: Preferences and features of a blood donation smart phone app: A multicentermixed-methods study in Riyadh, Saudi Arabia (Afaf Ali Batis, March 2021)

To identify the features and preferences of a blood donation smart phone app for blood donation centers and donors in Riyadh City, Saudi Arabia. This is a mixed-method study composed of a quantitative cross-sectional part (with donors, using a self-administered questionnaire), and a qualitative/quantitative part (with blood donation center staff, using semi-structured interviews). Data were collected between 15 November 2017 and 5 February 2018, from four blood donation centers in Riyadh City, Saudi Arabia. A descriptive analysis was used for the quantitative part and a thematic approach for the qualitative part.

2.3 Problem Statement Definition :

During COVID 19 crisis the requirement for plasma increased drastically as there were no vaccinations found in order to treat the infected patients. In such situation it was very difficult to find the plasma donor, check whether the donor was infected previously and was recovered, and which donor is eligible to donate plasma was a challenging task. As the plasma therapy was one of the ways to treat the infected patients getting the donor details played a major role.

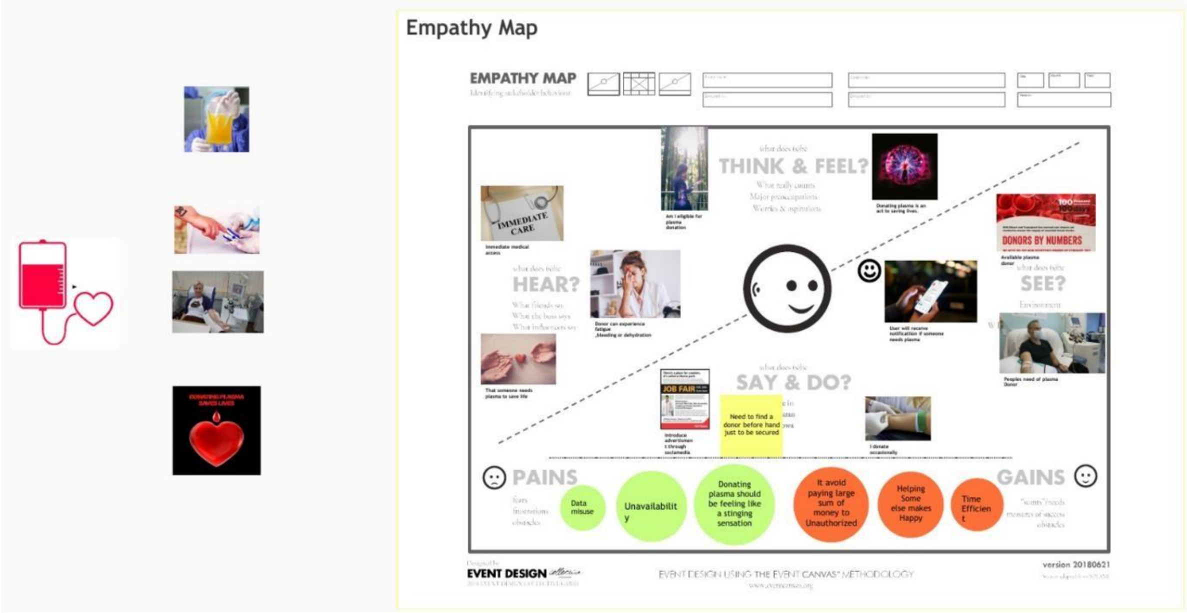


I am	I'm trying to	But	Because	Which makes me feel
a donor	donate plasma to the needy	it takes long time to accept the request	donor doesn't see the notification	frustrated
plasma receiver	to get plasma	proper donor is not available	there is no awareness among the people	hard

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a donor	donate plasma to the needy	it takes long time to accept the request	donor doesn't see the notification	frustrated
PS-2	plasma receiver	to get plasma	proper donor is not available	there is no awareness among the people	hard

CHAPTER 3: IDEATION PROPOSED SOLUTION

3.1 Empathy Map Canvas:



Empathy Map

EMPATHY MAP

what does he/she THINK & FEEL?

Am I eligible for plasma donation

Donating plasma is an act to saving lives.

Immediate medical access

what does he/she HEAR?

Donor can experience fatigue ,bleeding or dehydration

That someone needs plasma to save life

DONORS BY NUMBERS

Available plasma donor

what does he/she SEE?

User will receive notification if someone needs plasma

Peoples need of plasma Donor

what does he/she SAY & DO?

Introduce advertisment through socialmedia

Need to find a donor before hand just to be secured

I donate occasionally

PAINS

Data misuse

Unavailability

Donating plasma should be feeling like a stinging sensation

GAINS

It avoid paying large sum of money to Unauthorized

Helping Some else makes Happy

Time Efficient

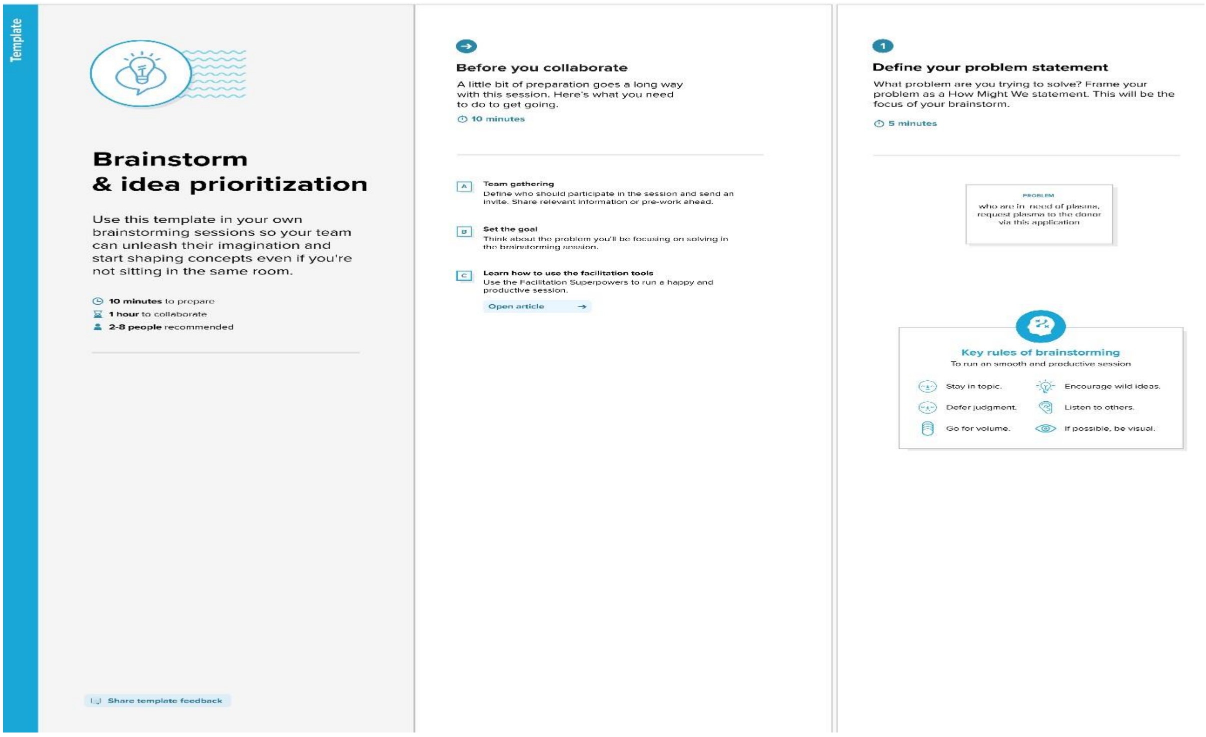
EVENT DESIGN

EVENT DESIGN USING THE EVENT CANVAS™ METHODOLOGY

version 20180621

3.2 Ideation & Brainstorming:

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

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

Share template feedback

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.

Open article →

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.

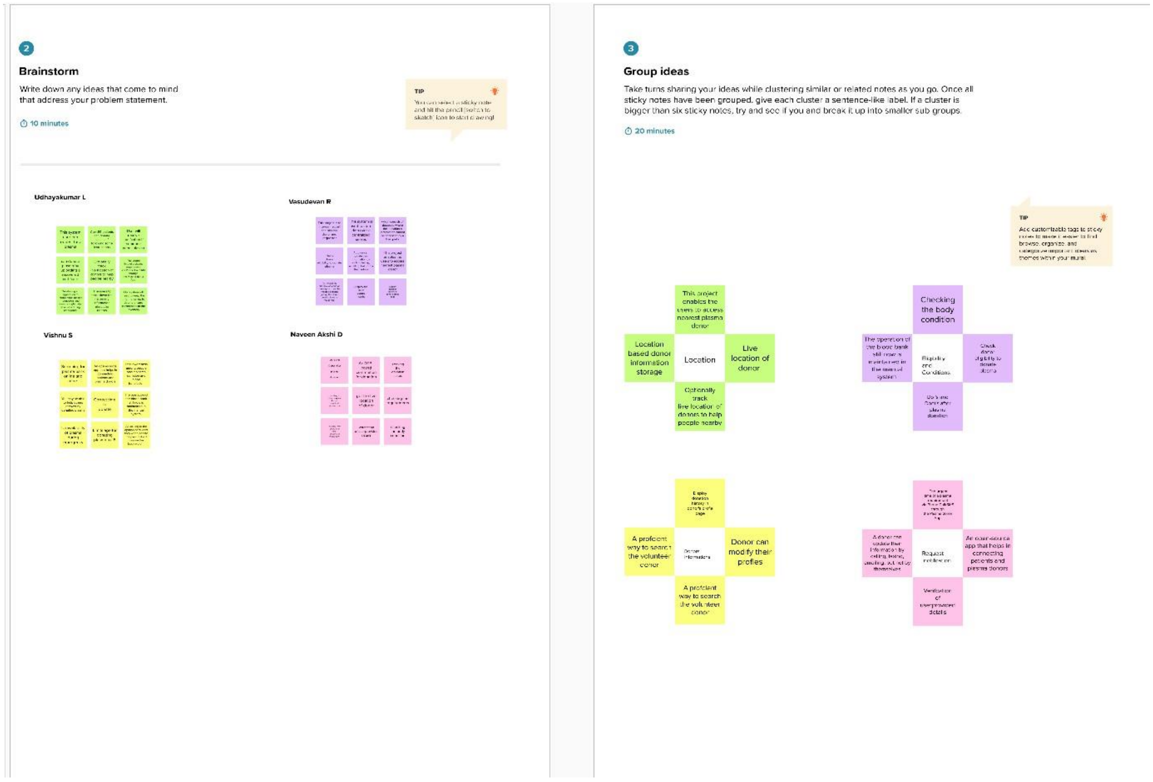
5 minutes

PROBLEM
who are in need of plasma, request plasma to the donor via this application

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.

Step-2: Brainstorm, Idea Listing and Grouping



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!

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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.

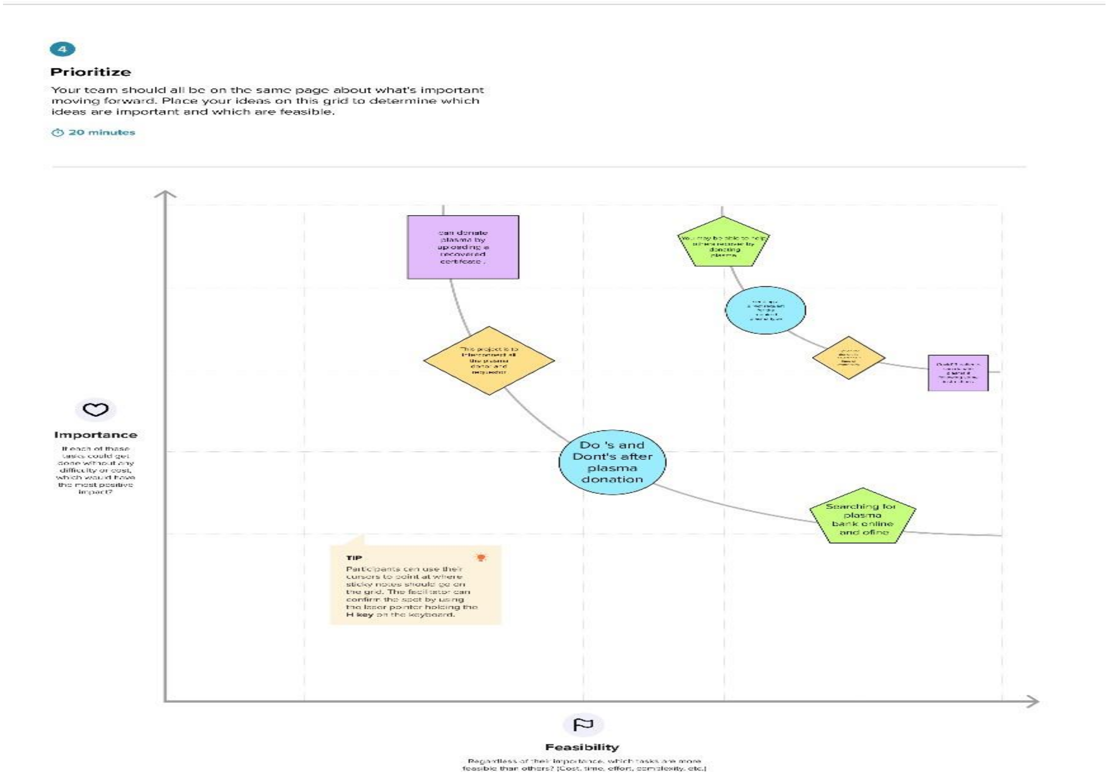
20 minutes

TIP

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

Udhayakumar L
Vasudevan R
Vishnu S
Naveen Akshi D
This project enables the users to access nearest plasma donor
Location based donor information storage
Location
Live location of donor
Optionally track live location of donors to help people nearby
Checking the body condition
The operation of the blood bank still now is maintained in the manual system
Eligibility and Conditions
Check donor eligibility to donate plasma
Do's and Don'ts after plasma donation
Display donation history in donor's profile page
A proficient way to search the volunteer donor
Donor Information
Donor can modify their profiles
A proficient way to search the volunteer donor
A donor can update their information by calling, texting, emailing, but not by themselves
Request notification
An open-source app that helps in connecting patients and plasma donors
Verification of userprovided details

Step-3: Idea Prioritization



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

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.

can donate plasma by uploading a recovered certificate .
You may be able to help others recover by donating plasma
This project is to interconnect all the plasma donor and requestor
Do 's and Dont's after plasma donation
Searching for plasma bank online and ofline
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.)

3.3 Proposed Solution :

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	During COVID 19 crisis the requirement for plasma increased drastically as there were no vaccinations found in order to treat the infected patients. In such situation it was very difficult to find the plasma donor, check whether the donor was infected previously and was recovered, and which donor is eligible to donate plasma was a challenging task. As the plasma therapy was one of the ways to treat the infected patients getting the donor details played a major role.
2.	Idea / Solution description	This proposed system aims at connecting the donors & the patients by an online application. By this creating application with UI to interact with the user for getting the donor details, who need it can see their details providing them upon the recipient's request so that they can get the plasma.
3.	Novelty / Uniqueness	Our application allows the user to request and donate the plasma. The person needs the plasma immediately or pre request. You have plasma immediately then give emergency request, then all registered member on the application to get voice alert.

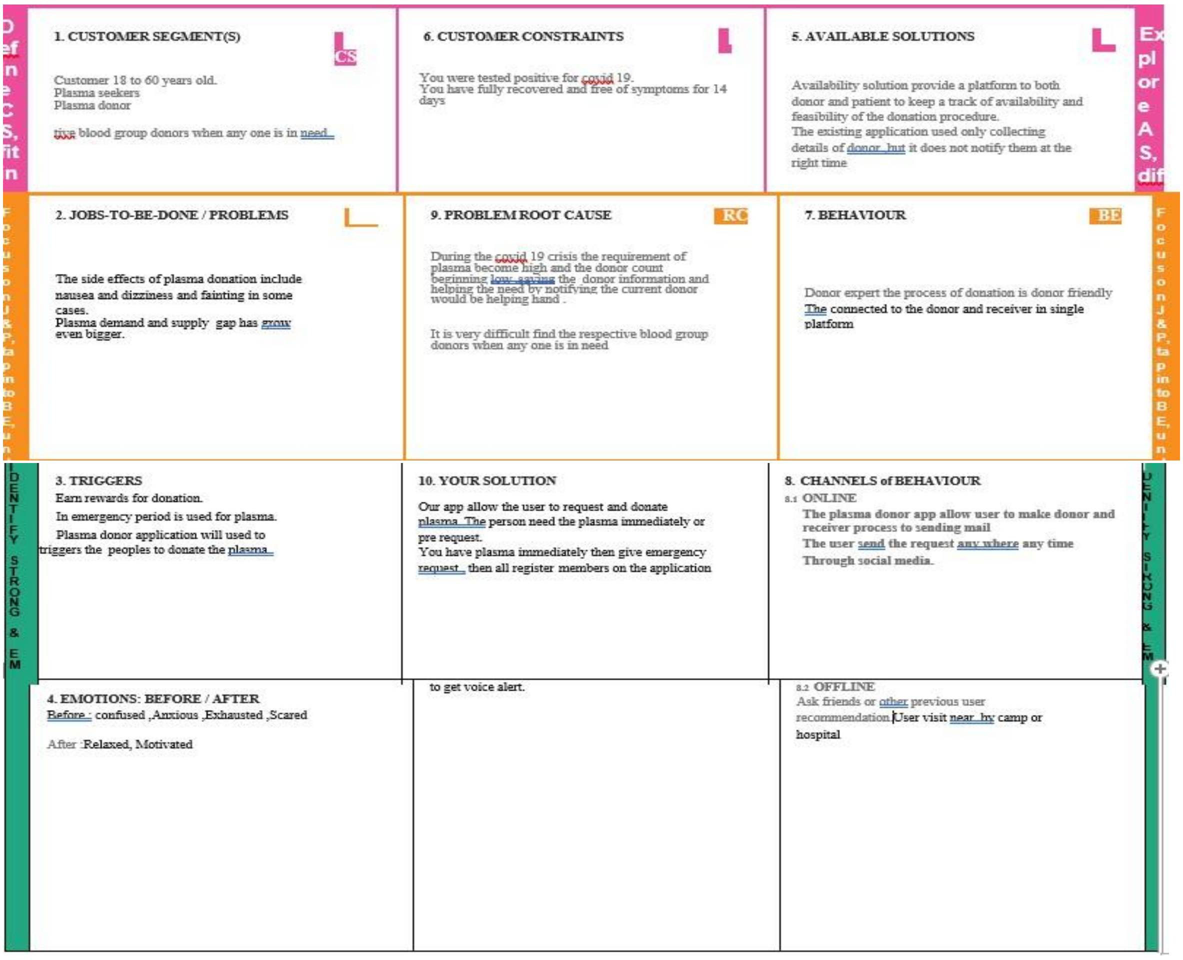
4.	Social Impact / Customer Satisfaction	In this covid19 period the requirement for plasma needs high and the donor count has low, so using this application provides opportunity come forward to donate plasma. we have predicted that effect of donor motivation on donor relationship satisfaction and loyalty change.
5.	Business Model (Revenue Model)	The application is user friendly and can be easily used. User Data can be stored in IBM DB2 in cloud which reduces the overall cost incurred for developing the application. This application is accessible by everyone. This can be used anywhere anytime.
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 an 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:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work an

Purpose:

- ❖ Solve complex problems in a way that fits the state of your customers.
- ❖ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- ❖ Sharpen your communication and marketing strategy with the right triggers and messaging.
- ❖ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- ❖ Understand the existing situation in order to improve it for your target group


Define CS, fit in
Explore AS, dif
Focus on J&P, tap into BE, un
IDENTIFY STRONG & EM

1. CUSTOMER SEGMENT(S) CS Customer 18 to 60 years old. Plasma seekers Plasma donor tive blood group donors when any one is in need.	6. CUSTOMER CONSTRAINTS You were tested positive for covid 19. You have fully recovered and free of symptoms for 14 days	5. AVAILABLE SOLUTIONS Availability solution provide a platform to both donor and patient to keep a track of availability and feasibility of the donation procedure. The existing application used only collecting details of donor. but it does not notify them at the right time
2. JOBS-TO-BE-DONE / PROBLEMS The side effects of plasma donation include nausea and dizziness and fainting in some cases. Plasma demand and supply gap has grow even bigger.	9. PROBLEM ROOT CAUSE RC During the covid 19 crisis the requirement of plasma become high and the donor count beginning low. saving the donor information and helping the need by notifying the current donor would be helping hand . It is very difficult find the respective blood group donors when any one is in need	7. BEHAVIOUR BE Donor expert the process of donation is donor friendly The connected to the donor and receiver in single platform
3. TRIGGERS Earn rewards for donation. In emergency period is used for plasma. Plasma donor application will used to triggers the peoples to donate the plasma.	10. YOUR SOLUTION Our app allow the user to request and donate plasma. The person need the plasma immediately or pre request. You have plasma immediately then give emergency request. then all register members on the application	8. CHANNELS of BEHAVIOUR 8.1 ONLINE The plasma donor app allow user to make donor and receiver process to sending mail The user send the request any where any time Through social media.
4. EMOTIONS: BEFORE / AFTER Before : confused ,Anxious ,Exhausted ,Scared After :Relaxed, Motivated	to get voice alert.	8.2 OFFLINE Ask friends or other previous user recommendation.User visit near by camp or hospital

CHAPTER 4: REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login using Registered email Id
FR-4	Sent Request	If plasma is required, the receiver will contact the donor
FR-5	Contact Donor	Contact the donor directly if a phone number is given
FR-6	View donation camps	View the list of donation camps happening nearby

4.2 Non Functional requirements:

Following are the non-functional requirements of the proposed solution.

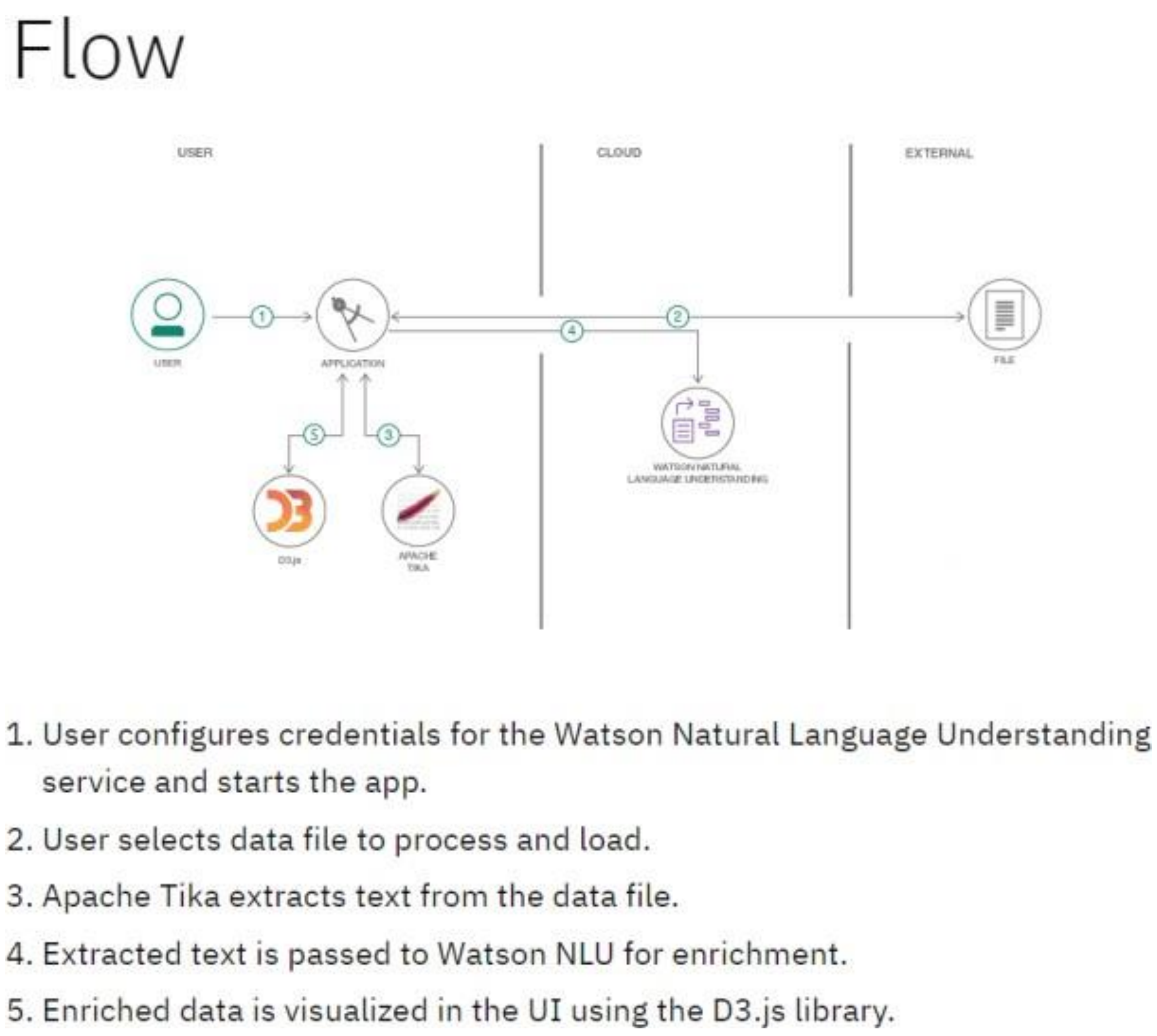
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface of the plasma donor system must be well-designed and welcoming.
NFR-2	**Security**	Data storage is required by security systems, just like it is by many other applications. Databases are able to keep all the donor information that is viewed by applications. It must be secured with email Id and password.
NFR-3	**Reliability**	The system has the ability to work all the times without failures apart from network failure. A donor can have the faith on the system. The authorities will keep the privacy of all donors in a proper manner
NFR-4	**Performance**	The Plasma donor System must perform well in different scenarios. The system is interactive and delays involved are less.
NFR-5	**Availability**	The system, including the online components, should be available 24/7.
NFR-6	**Scalability**	The system offers the proper resources for issue solutions and is designed to protect sensitive information during all phases of operation.

CHAPTER 5: PROJECT DESIGN

5.1 Data Flow Diagrams:

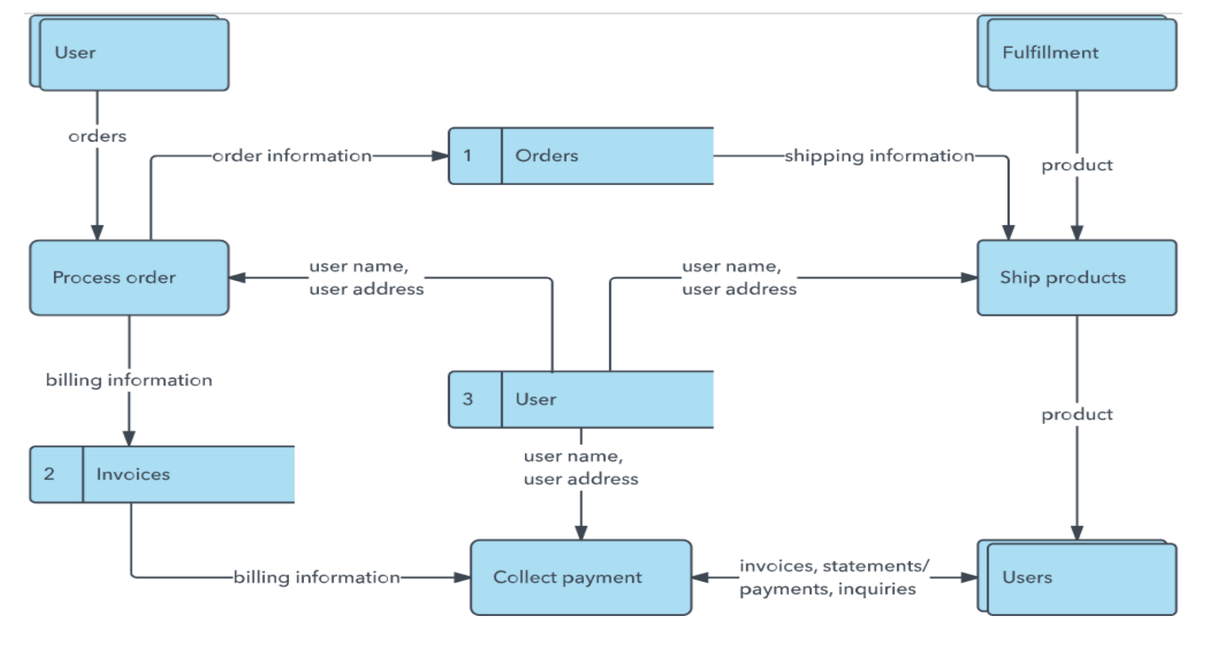
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: (Simplified)

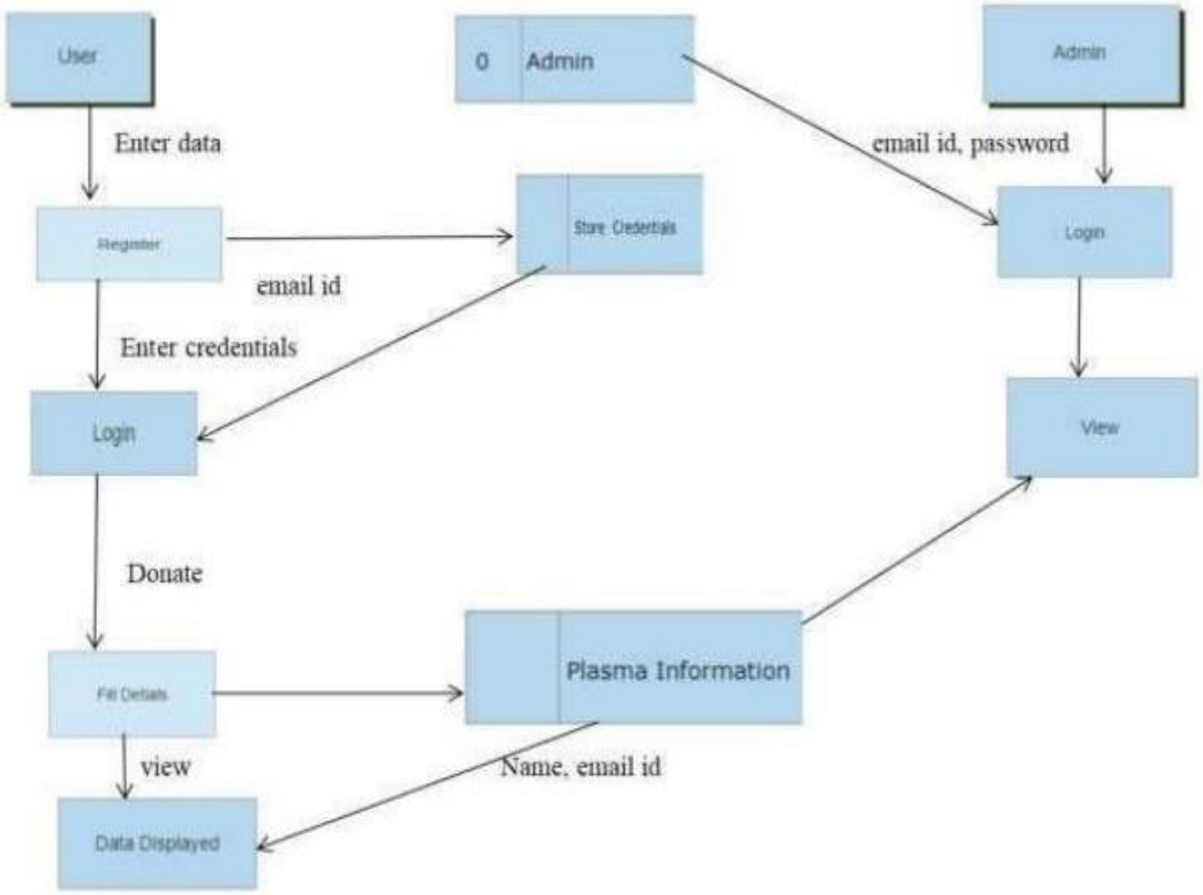


1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

Example: DFD Level 0 (Industry Standard)



Dataflow Diagram (Plasma Donor Application):

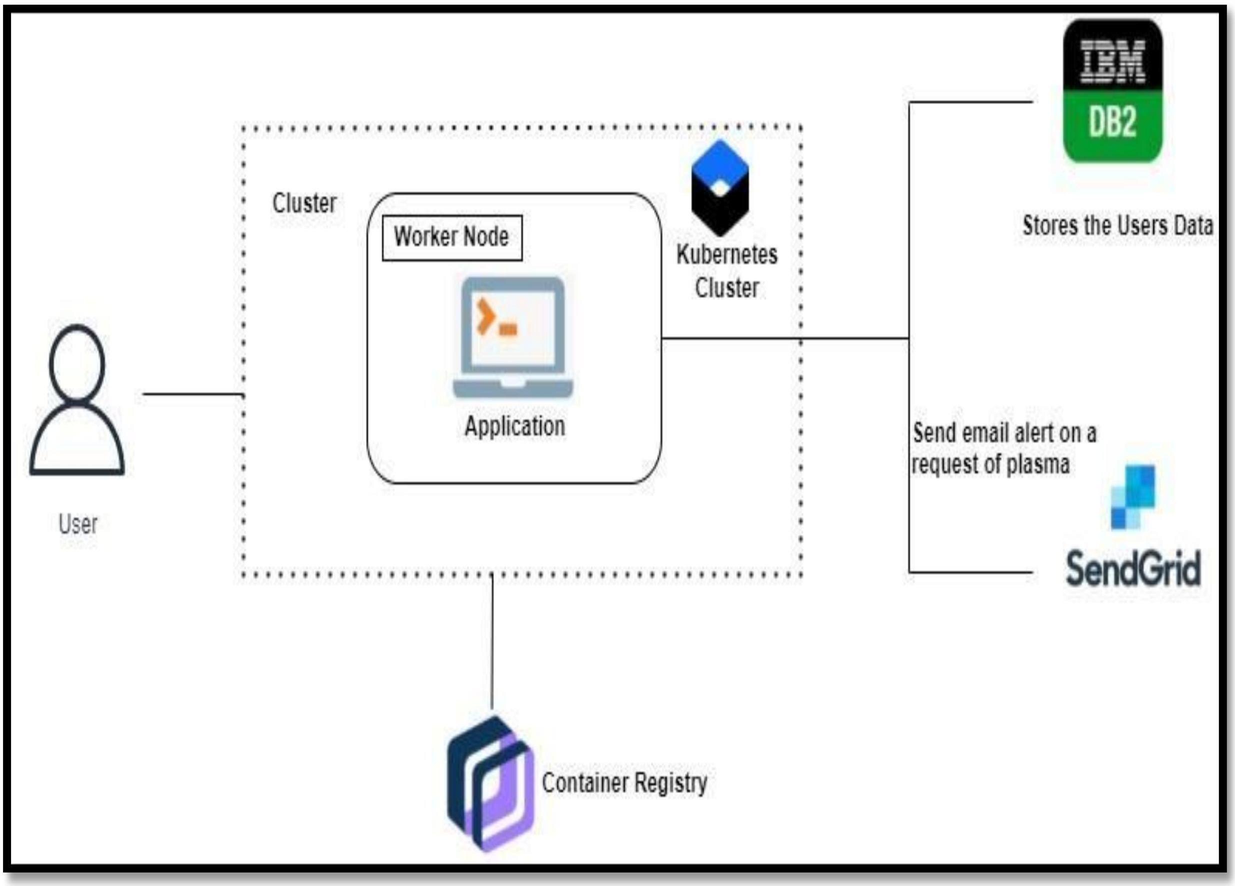


5.2 Solution & Technical Architecture:

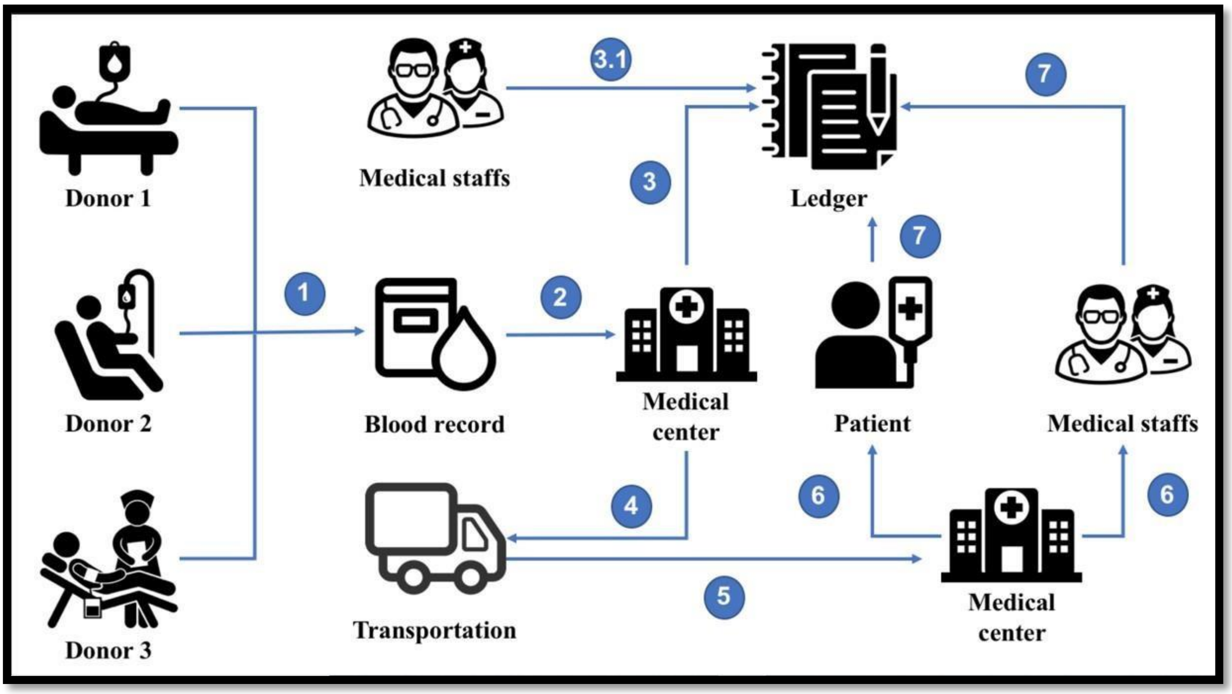
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed,and delivered.

Technical Architecture:



Solution Architecture-Donor



5.3 User Stories:

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

User Type	Functional Requirement (Epic)	User Story No	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 access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email & password	I can register & access the dashboard with Facebook Login	High	Sprint-1
	Dashboard	USN-4	As a user, I can register for the application through Gmail	I can view and access what are the features are provided 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 access what are the features are provided in dashboard	High	Sprint-1
Customer Care Executive	Query	USN-6	As a user had an any query about the given requirements	I can view a query and rectify the given query	Medium	Sprint-2
Administrator	Login	USN-7	As an admin, have credentials using that they can login	They can view and modify the data in database	Medium	Sprint-2
	View	USN-8	As an admin I can view plasma information	View and modify	High	Sprint-1
	Modify	USN-9	As an admin I can modify the plasma information.	Modify only if there is a false information	Medium	Sprint-1

CHAPTER 6: PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	A user can register for the application by entering their email, password, and confirming the password.	3	High	Udhayakumar L
	Email verification	USN-2	A user will receive confirmation e-mail once they have registered for the application.	3	High	Vasudevdan R
	Login	USN-3	A user can log in and enter the application by proceeding with email &password.	2	High	Naveen Akshi D
	Donor/Recipient Profile	USN-4	A user is able to register themselves as verified plasma donor or a recipient.	3	High	Vishnu S
Sprint-2	Virtual Certificate	USN-5	A user will get a virtual donation certificate after a verified successful Plasma donation.	2	Mediu m	Udhayakumar L
	Plasma Request	USN-6	A verified clinic can make a Plasma request through the application.	3	High	Vasudevan R
	Verification of Donor's details	USN-7	We the administrators will verify the details provided by the donor/ recipient are correct for a genuine Plasma donation or plasma	2	Mediu m	Naveen Akshi D

			recipient.			
Sprint-3	Accept the donation request	USN-8	A user and a registered donor will get a notification to accept the plasma request for their specific blood type.	3	High	Vishnu S
	Communication Channel	USN-9	A patient is able to communicate with the donor personally within the application.	3	Medium	Udhayakumar L
	Share Location	USN-10	A user and a registered donor is able to share their location with the recipient after accepting their Plasma request.	3	Medium	Vasudevan R
	Administrator	USN-11	An admin will store the registered donors and recipients details after their verification into the database.	3	High	Naveen Akshi D
Sprint-4	Support	USN-12	A user is able to ask basic question Related to plasma donation/ Plasma recipient with the help of chat-bot.	2	Medium	Vishnu S

	About	USN-13	A new user can use the About page to know information and details related to Plasma donation and benefits. Also the details of Plasma.	2	Medium	Udhayakumar L Vasudevan R

	Administrator	USN-14	An admin will approve all the Plasma transaction in the application after every proper verification and will update the Plasma availability and donors count periodically.	3	High	Naveen Akshi D Vishnu S

6.2 Sprint Delivery Schedule:

Project Tracker, Velocity & Burn down Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (ason Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	18	03 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	09 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	18	15 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	18	21 Nov 2022

Velocity:

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

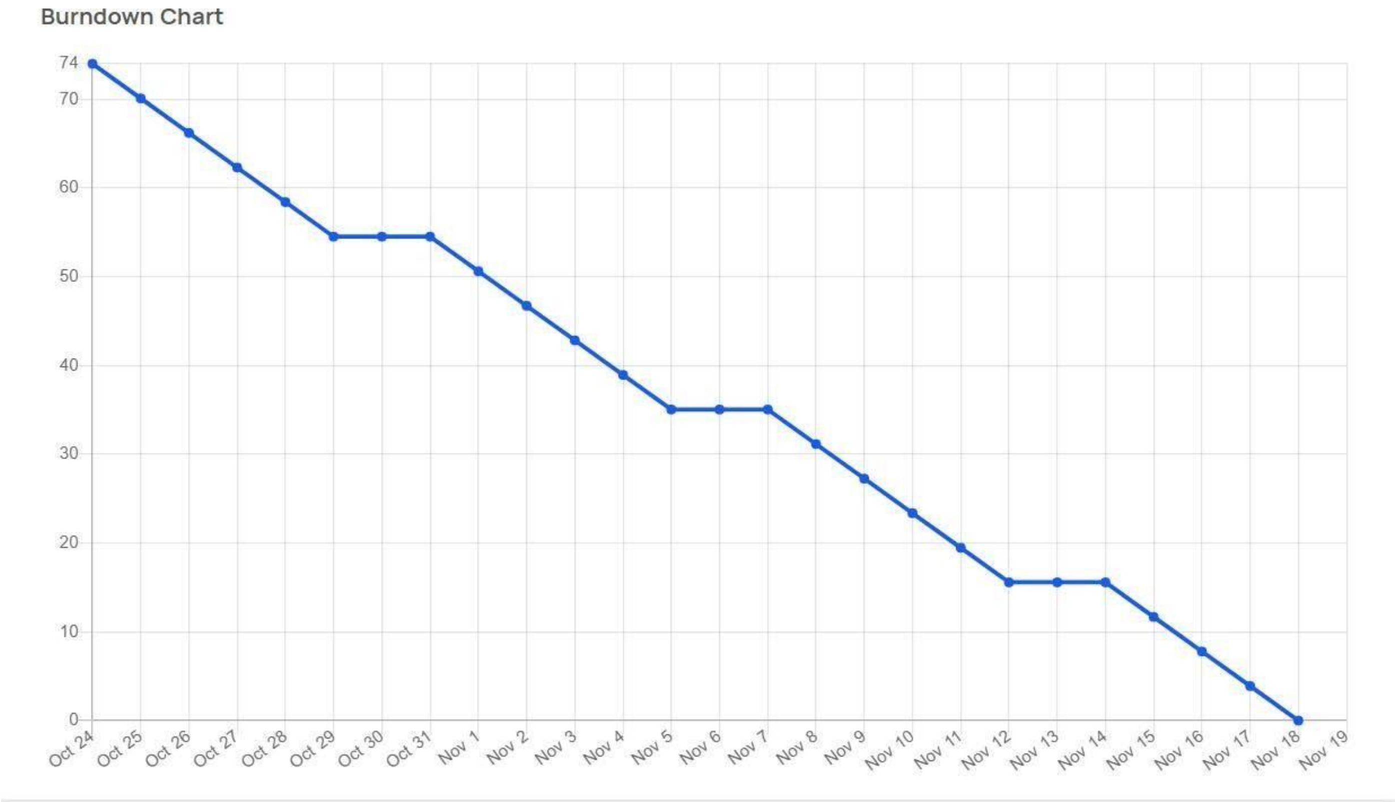
AV= Sprint Duration/ Velocity=24/20= 1.2

Total number of days = sprint1 +sprint2+ sprint3+ sprint4 = 6 +6+ 6+6 = 24 Total number of story points = 18 +20+ 18+ 18 = 74

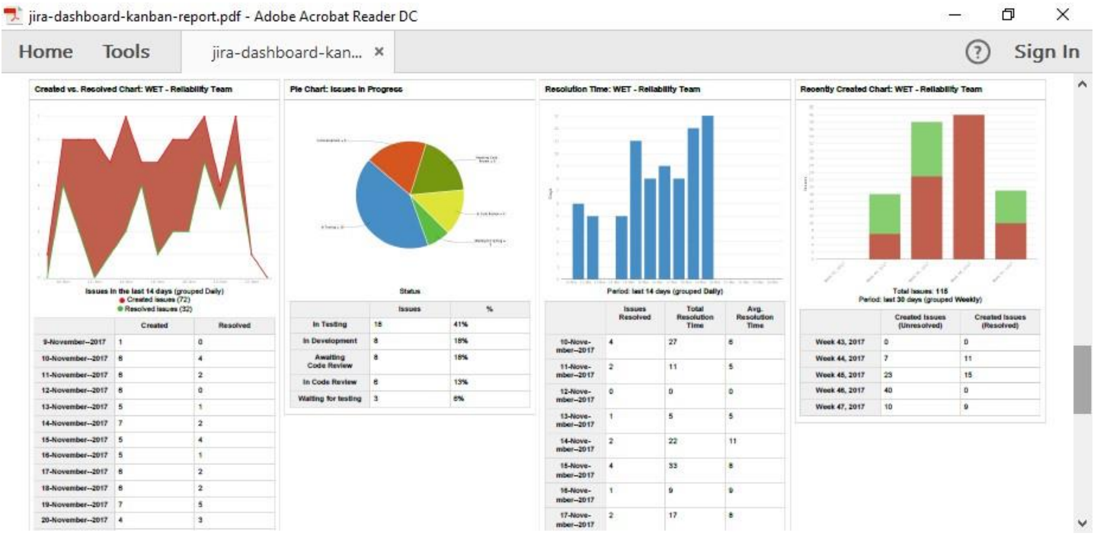
Average velocity per sprint = 74/24

~ = 3.083333 = 3

Burn down Chart: A burn down chart is a graphical representation of work left to do versus time . It is often used in agile software development methodologies such as Scrum . However, burn down charts can be applied to any project containing measurable progress overtime.



6.3 Report from JIRA



Created vs. Resolved Chart: WET - Reliability Team

Issues in the last 14 days (grouped Daily)
● Created issues (72)
● Resolved issues (32)

	Created	Resolved
9-November--2017	1	0
10-November--2017	6	4
11-November--2017	6	2
12-November--2017	6	0
13-November--2017	5	1
14-November--2017	7	2
15-November--2017	5	4
16-November--2017	5	1
17-November--2017	6	2
18-November--2017	6	2
19-November--2017	7	5
20-November--2017	4	3

Pie Chart: Issues In Progress

Status

	Issues	%
In Testing	18	41%
In Development	8	18%
Awaiting Code Review	8	18%
In Code Review	6	13%
Waiting for testing	3	6%

Resolution Time: WET - Reliability Team

Period: last 14 days (grouped Daily)

	Issues Resolved	Total Resolution Time	Avg. Resolution Time
10-November--2017	4	27	6
11-November--2017	2	11	5
12-November--2017	0	0	0
13-November--2017	1	5	5
14-November--2017	2	22	11
15-November--2017	4	33	8
16-November--2017	1	9	9
17-November--2017	2	17	8

Recently Created Chart: WET - Reliability Team

Total Issues: 115
Period: last 30 days (grouped Weekly)

	Created Issues (Unresolved)	Created Issues (Resolved)
Week 43, 2017	0	0
Week 44, 2017	7	11
Week 45, 2017	23	15
Week 46, 2017	40	0
Week 47, 2017	10	9

CHAPTER 7: CODING & SOLUTIONING

7.1 SendGrid

SendGrid is a cloud-based SMTP provider that allows you to send email without having to maintain email servers. SendGrid manages all of the technical details, from scaling the infrastructure to ISP outreach and reputation monitoring to whitelist services and real time analytics.

SendGrid provides two ways to send email: through our SMTP relay or through our Web API. SendGrid provides client libraries in many languages. This is the preferred way to integrate with SendGrid. If you choose to use SendGrid without a client library, the Web API is recommended in most cases as it is faster, provides some benefit with encoding, and tends to be easier to use. SMTP provides many features by default, but is harder to setup.

Web API:

- The Web API has some advantages over SMTP:
- If your ISP blocks all outbound mail ports and your only option isHTTP.
- If there is high latency between your site and ours, the Web API mightbe quicker since it does not require as many messages between the client and server.
- If you do not control the application environment and cannot installand configure an SMTP library.
- If you build a library to send email, developing against a web APIprovides quicker development.

SMTP Relay

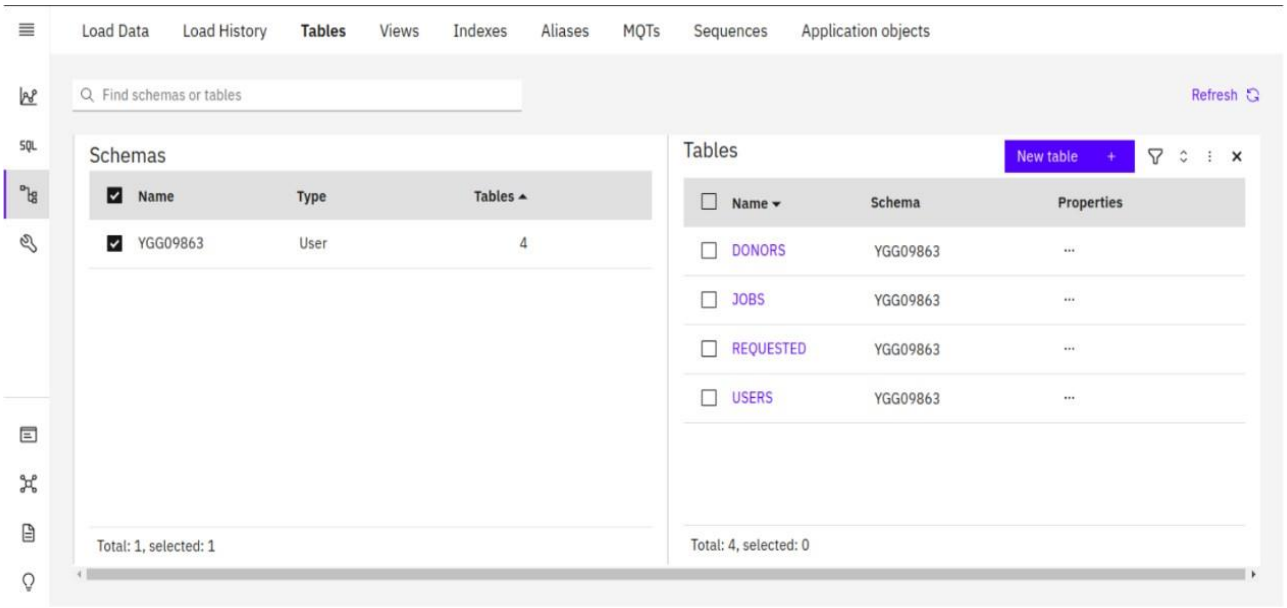
- If you are integrating SendGrid with an existing application, setting upthe application to use our SMTP relay is easiest, as it only requires modifying SMTP configuration.

- Change your SMTP username and password to your SendGridcredentials.
- Set the server host name to smtp.sendgrid.net
- Use ports 25 or 587 for plain/TLS connections and port 465 for SSL connections.

Code:

```
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='udhayakumar24001@gmail.coml',to_emails=usermail,subject=
subj ect,
html_content='<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)
```

7.2 Database Schema:



Schemas

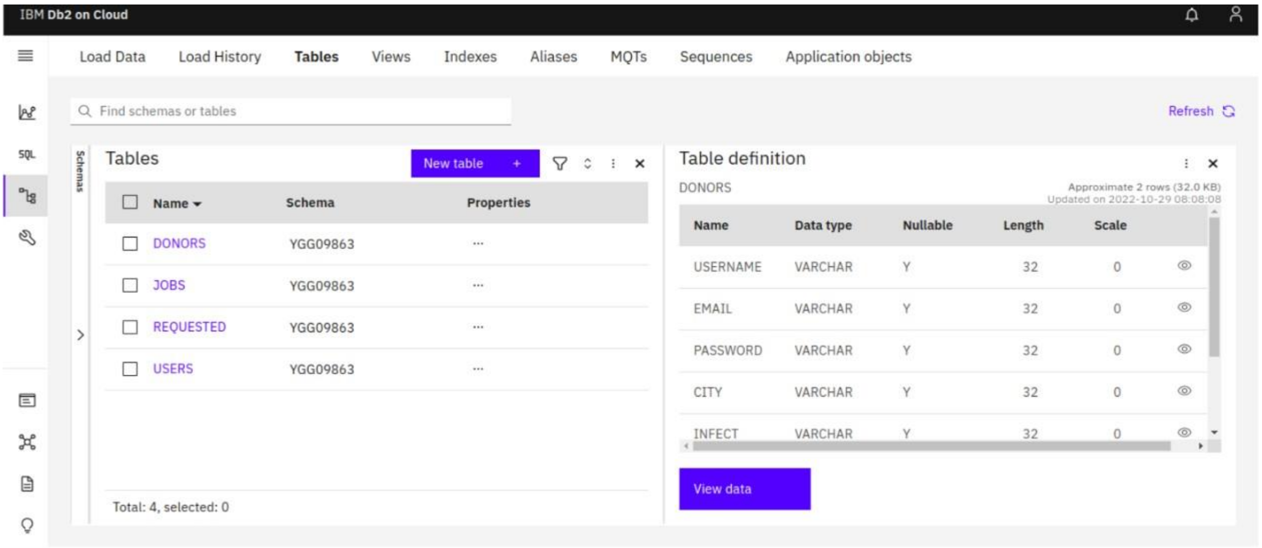
☑ Name	Type	Tables ▲
☑ YGG09863	User	4

Total: 1, selected: 1

Tables

□ Name ▾	Schema	Properties
□ DONORS	YGG09863	···
□ JOBS	YGG09863	···
□ REQUESTED	YGG09863	···
□ USERS	YGG09863	···

Total: 4, selected: 0



Tables

□ Name ▾	Schema	Properties
□ DONORS	YGG09863	···
□ JOBS	YGG09863	···
□ REQUESTED	YGG09863	···
□ USERS	YGG09863	···

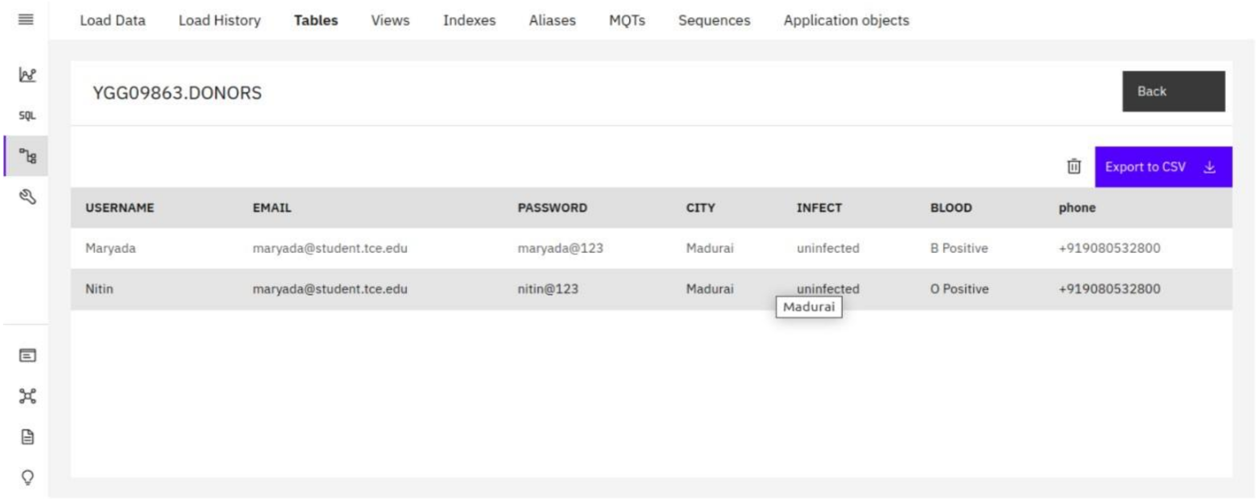
Total: 4, selected: 0

Table definition

DONORS

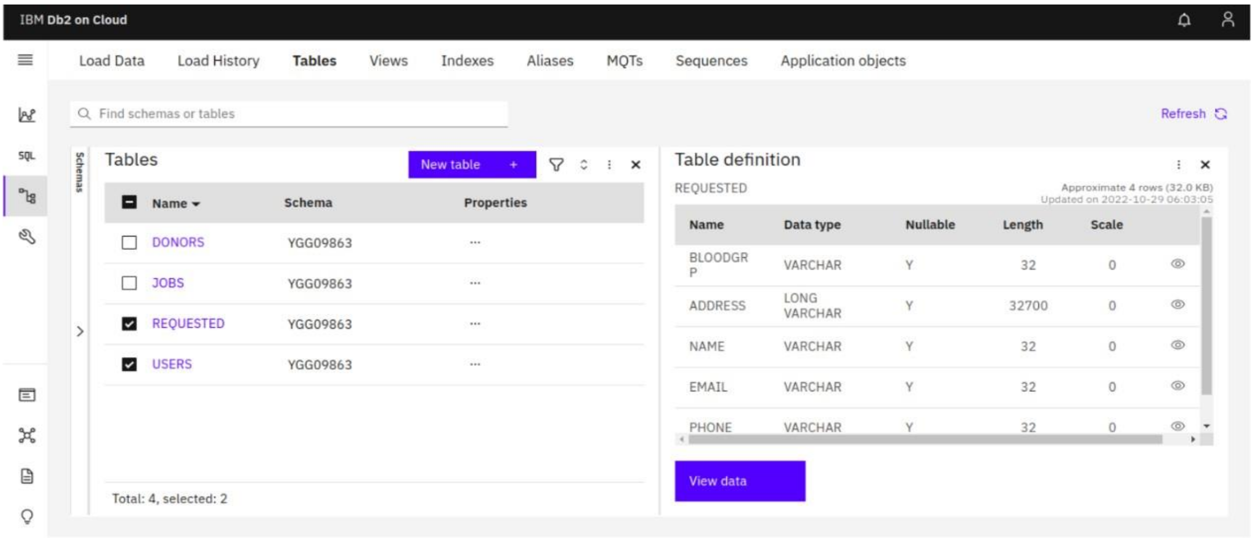
Approximate 2 rows (32.0 KB)
Updated on 2022-10-29 08:08:08

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0
CITY	VARCHAR	Y	32	0
INFECT	VARCHAR	Y	32	0



YGG09863.DONORS

USERNAME	EMAIL	PASSWORD	CITY	INFECT	BLOOD	phone
Maryada	maryada@student.tce.edu	maryada@123	Madurai	uninfected	B Positive	+919080532800
Nitin	maryada@student.tce.edu	nitin@123	Madurai	uninfected	O Positive	+919080532800



Tables

☐	Name	Schema	Properties
☐	DONORS	YGG09863	···
☐	JOBS	YGG09863	···
☑	REQUESTED	YGG09863	···
☑	USERS	YGG09863	···

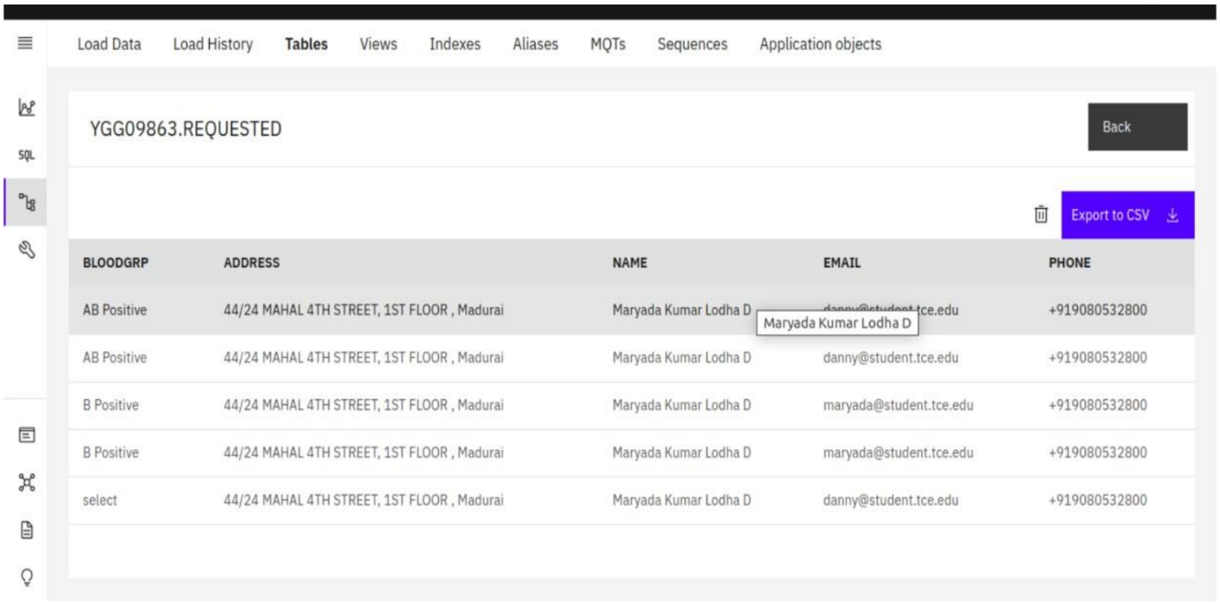
Total: 4, selected: 2

Table definition

REQUESTED

Approximate 4 rows (32.0 KB)
Updated on 2022-10-29 06:03:05

Name	Data type	Nullable	Length	Scale
BLOODGRP	VARCHAR	Y	32	0
ADDRESS	LONG VARCHAR	Y	32700	0
NAME	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
PHONE	VARCHAR	Y	32	0



YGG09863.REQUESTED

BLOODGRP	ADDRESS	NAME	EMAIL	PHONE
AB Positive	44/24 MAHAL 4TH STREET, 1ST FLOOR , Madurai	Maryada Kumar Lodha D	danny@student.tce.edu	+919080532800
AB Positive	44/24 MAHAL 4TH STREET, 1ST FLOOR , Madurai	Maryada Kumar Lodha D	danny@student.tce.edu	+919080532800
B Positive	44/24 MAHAL 4TH STREET, 1ST FLOOR , Madurai	Maryada Kumar Lodha D	maryada@student.tce.edu	+919080532800
B Positive	44/24 MAHAL 4TH STREET, 1ST FLOOR , Madurai	Maryada Kumar Lodha D	maryada@student.tce.edu	+919080532800
select	44/24 MAHAL 4TH STREET, 1ST FLOOR , Madurai	Maryada Kumar Lodha D	danny@student.tce.edu	+919080532800

CHAPTER 8: TESTING

8.1 Test Cases

With a concerted effort, I conducted research on general well-being to have a rudimentary grasp on health management, as well as the existing nutrient tracking apps in order to get an understanding of what is already existing in the market, the characteristics, specialties, and usability.There are a considerable number of nutrition tracking apps existing in the market. They aim to track daily calories/macros intake by logging meals to achieve users' preset goals. To log meals, users can input the food in the app, or scan the barcode of a package. Most apps allow users to connect with associated activities apps to track exercise progress. With a premium upgrade, users can get access to tailor-made recipes according to health goals or specified diets.In order to build a realistic initial target group, I wanted to conduct some usability tests with 5 users that regularly engage in physical activity and food tracking, including both first-time and regular users of meal planning and fitness apps. I asked these individuals to perform tasks related to general usage of the MyFitnessPal, Lifesum, and Nutrition Coach apps (such as food logging, searching, and checking their caloric breakdown.)

Test Description	Expected	Result
Connect to Desktop site	Login page should be displayed	Passed
Login with wrong password	Should fail to login	passed
Login with correct password	Should show Dashboard	passed
Search for diet plans	Should show plans	passed

Enroll into a plan	Should be enrolled	passed
Again enroll	Should not enroll	passed

8.2 User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Plasma Donor Application project at the time of the release to User Acceptance Testing (UAT).

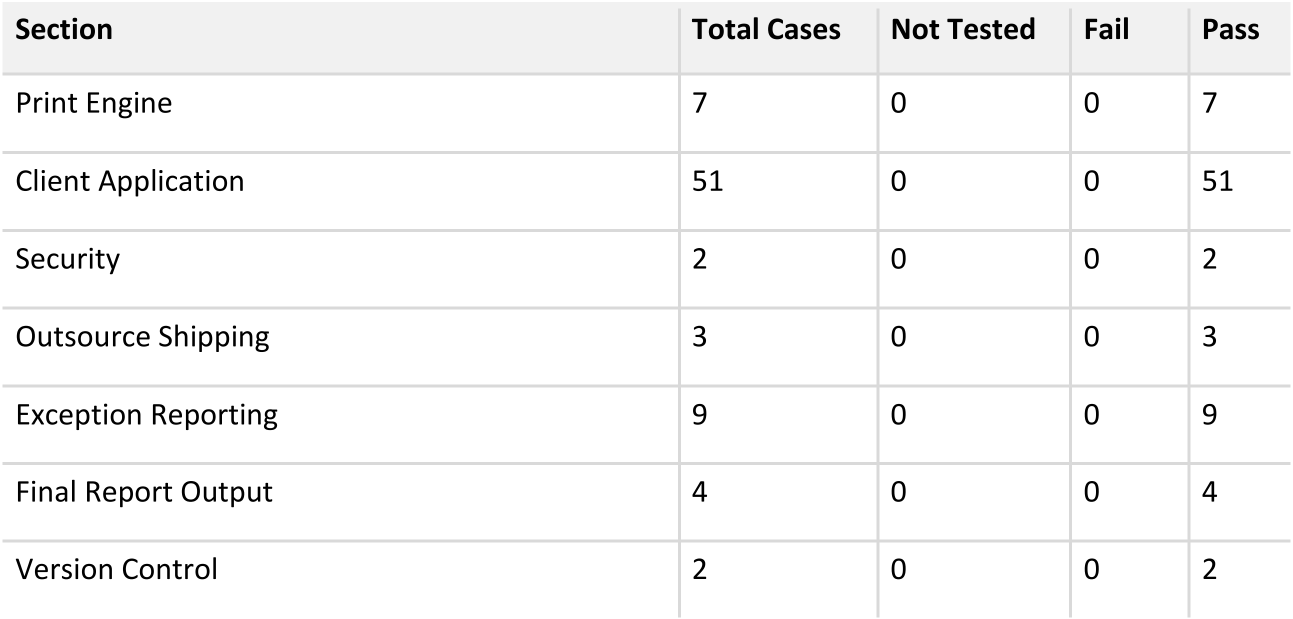
2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested



Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

CHAPTER 9: RESULT

9.1 Performance Matrix:

Authentication Module

- ❖ Sign Up

New user or donor can create an account to use in the blood/plasmadonor application and create a password for account verification and create an identity.

- ❖ Sign In

Donor Sign In to the account for viewing or editing location detailsand any other personal information.

- ❖ Account Verification

If donor changes their password or if they forget the password then wehave to verify their account using mail verification.

Service Provider Module

- ❖ Add New Donor

User can be able to register to add donor details.

- ❖ List All Donor

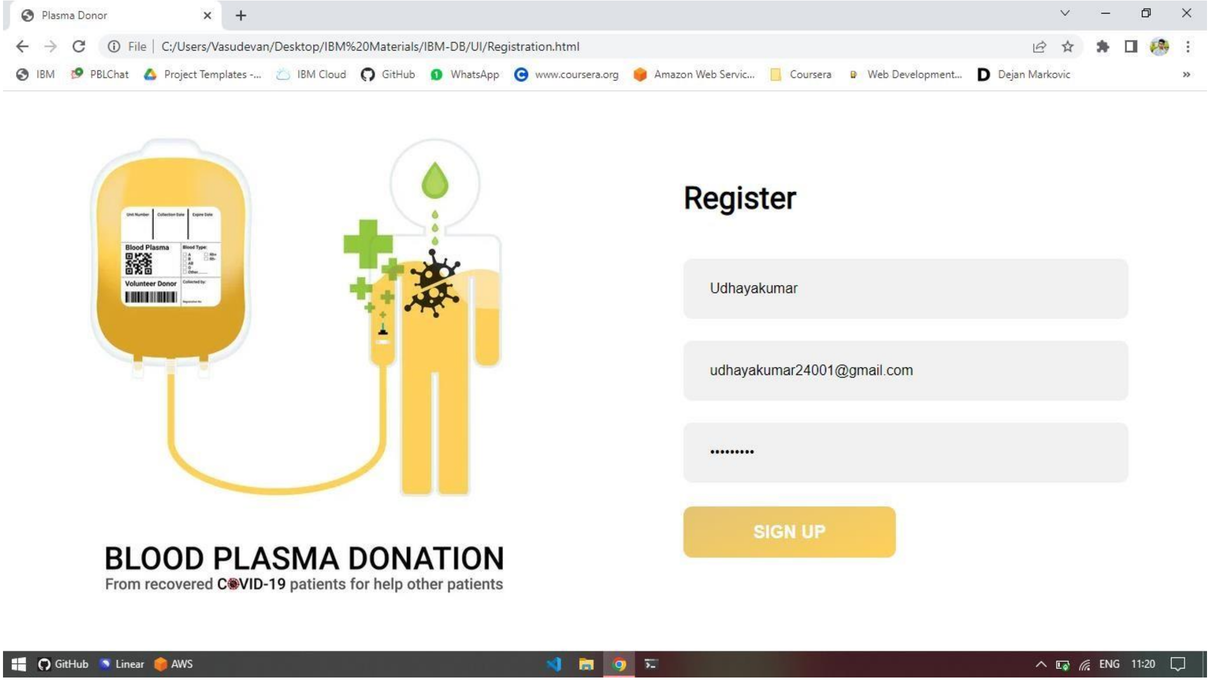
User can be able to view all Donor who all use our Plasma Donor Application.

- ❖ Edit Customer Plan Details

User can be able to edit the existing Donor details as the Donor wish.

Screenshots:

Registration Phase



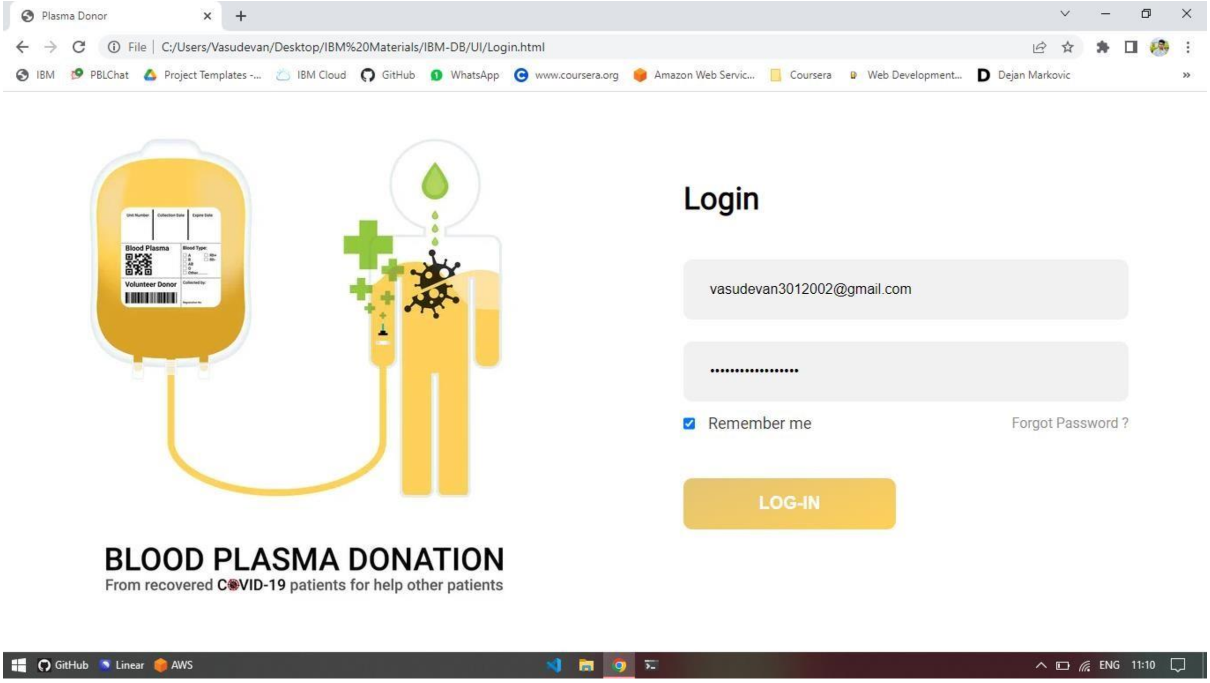
Register

Udhayakumar

udhayakumar24001@gmail.com

•••••••••

Login Page



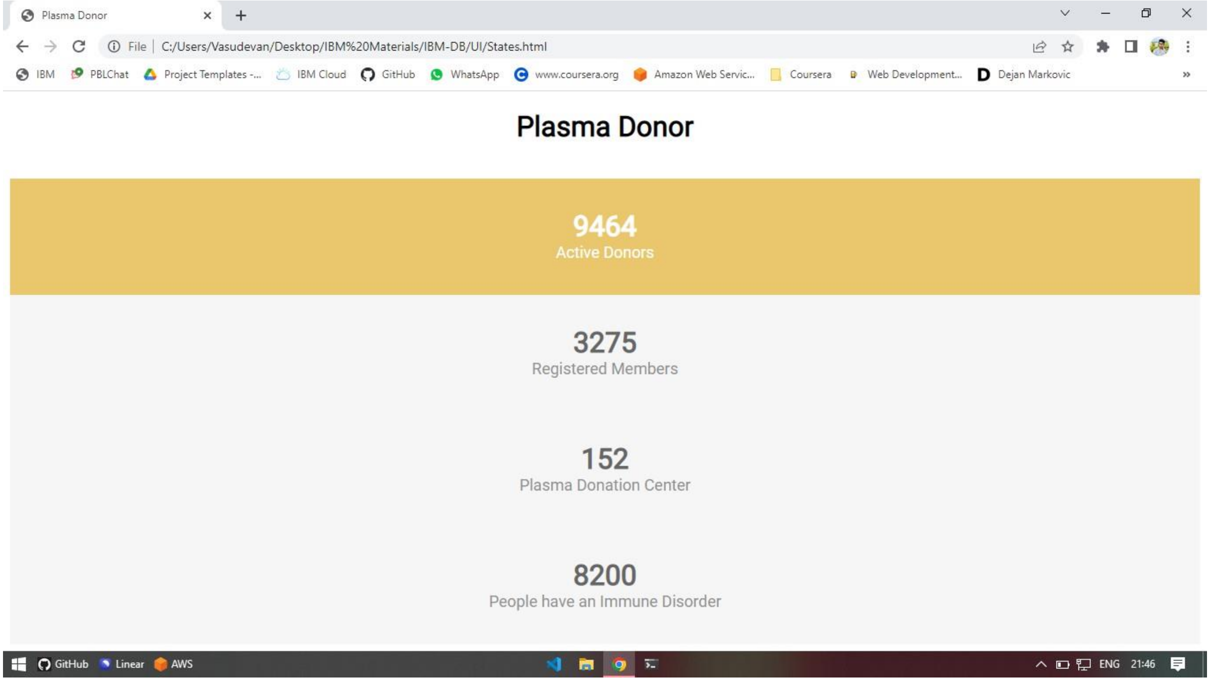
Login

vasudevan3012002@gmail.com

••••••••••••••••••

☑ Remember me

Stats Page to display the count:



Plasma Donor

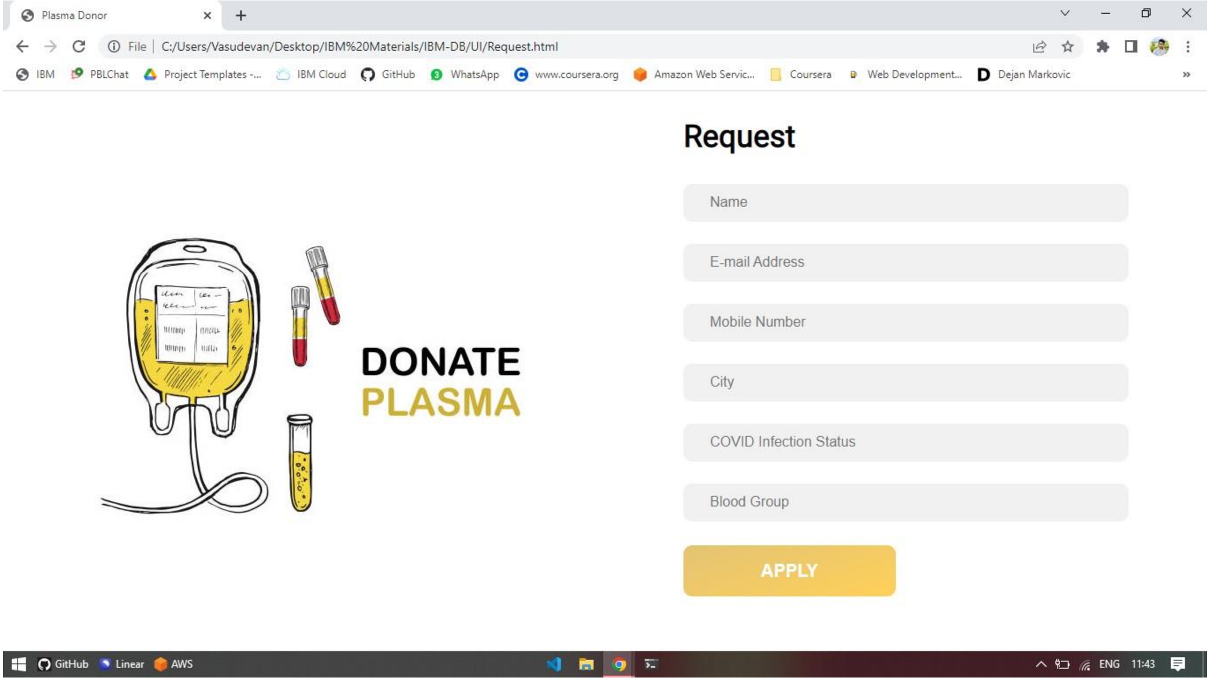
9464
Active Donors

3275
Registered Members

152
Plasma Donation Center

8200
People have an Immune Disorder

Request Page



Request

Name

E-mail Address

Mobile Number

City

COVID Infection Status

Blood Group

APPLY

CHAPTER 10: ADVANTAGES & DISADVANTAGES

Advantages

- Speed

  This website is fast and offers great accuracy as compared to manual registered keeping.

- Maintenance

  Less maintenance is required

- User Friendly

  It is very easy to use and understand. It is easily workable and accessible for everyone.

- Fast Results

  It would help you to provide plasma donors easily depending upon the availability of it.

Disadvantages

- **Internet**

  It would require an internet connection for the working of the website.

- Auto- Verification

  It cannot automatically verify the genuine users.

CHAPTER 11: CONCLUSION

Although the government is carrying out Covid vaccination campaigns on a large scale, the number of vaccines produced is not enough for all the population to get vaccinated at present. And with the corona positive cases rising every day, saving lives has become the prime matter of concern. As per the data provided by WHO more than 3 million people have died due to the coronavirus. However, apart from vaccination, there is another scientific method by which a covid infected person can be treated and the death risk can be reduced. This plasma therapy is an experimental approach to treat corona- positive patients and help them recover. This plasma therapy is considered to be safe & promising. A person who has recovered from Covid can donate his/her plasma to a person who is infected with the coronavirus.

This system proposed here aims at connecting the donors & the patients by an online application. By using this application, the users can either raise a request for plasma donation or requirement. Both parties can Accept or Reject the request. User has to Upload a Covid Negative report to be able to Donate Plasma. This system is used if anyone needs a Plasma Donor Blood and Plasma donation is a kind of citizen's social responsibility in which an individual can willingly donate blood/plasma via our app. This Application has been created with the concept and has sought to make sure that the donor gives blood/plasma to community. This model is made user friendly so anybody can view and maintain his/her account. This application will break the chain of business through blood/plasma and help the poor to find donor at free of cost. This project will help new blood/plasma banks improve their services and progress from traditional to user-friendly frameworks.

CHAPTER 12: FUTURE SCOPE

Plasma Application can be developed to further improve user accessibility via integrating this application with various social networks application program interfaces (APIs). Consequently, users can login and sign up using various social networks. This would increase number of donors and enhances the process of blood donation.

User interface (UI) can be improved in future to accommodate global audience by supporting different languages across countries. Data scraping can be done from different social networks and can be shown in the Blood/Plasma Request Feeds. Appointments can be synchronized with Google and Outlook calendars for the ease of users.

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.

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

CHAPTER 13: APPENDIX

13.1 Source Code

App.py:

```
from distutils.log import debug
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'),"","")
conn= ibm_db.connect(" DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dxj24137;PWD=pGm0FgyMqS3Jy4JV ",",")
@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')
```

flaskmail.py:

```
# importing libraries
from flask import Flask
from Flask-Mail import Mail, Message
app = Flask(__name__)
mail = Mail(app) # instantiate the mail class
```

```
# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'vasu.awsconsole1@gmail.com'
app.config['MAIL_PASSWORD'] = 'awsaccountno1'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
# message object mapped to a particular URL �/�
@app.route("/")
def index():
msg = Message(
        'Hello',
        sender ='vasu.awsconsole1@gmail.com',
        recipients = ['receiver�vasu3012002@gmail.com']
      )
msg.body = 'Hello Flask message sent from Flask-Mail'
mail.send(msg)
return 'Sent'
if __name__ == '__main__':
app.run(debug = True)
```

Dockerfile:

```
FROM python:3.6
WORKDIR /app
ADD . /app
COPY requirements.txt /app
RUN python3 -m pip install -r requirements.txt
EXPOSE 5000
CMD ["python","app.py"]
```

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"></scrip
t>
  <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>
```

Login.html

```
doctype html
html(lang="es", dir="ltr")
    head
        meta(name="viewport", content="width=device-width, user-scalable=no, initial-scale=1.0")
        meta(charset="utf-8")
        link(rel="stylesheet", type="text/css", href="main.css")
        <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700;800&display=swap" rel="stylesheet">
    body

        div.main
            div.container.a-container#a-container
                form(id="a-form", class="form" method="", action="")
                    h2.form_title.title Create Account
                    div.form__icons
                    img.form__icon(src="data:image/svg+xml", alt="")
                    img.form__icon(src="data:image/svg+xml")
                    img.form__icon(src="data:image/svg+xml")
                    span.form__span or use email for registration
                    input.form__input(type="text", placeholder="Name" )
                    input.form__input(type="text", placeholder="Email")
                    input.form__input(type="password", placeholder="Password")
                    button().form__button.button.submit SIGN UP

            div.container.b-container#b-container
                form(id="b-form", class="form" method="", action="")
                    h2.form_title.title Sign in to Website
                    div.form__icons
                    img.form__icon(src="data:image/svg+xml ", alt="")
                    img.form__icon(src="data:image/svg+xml")
                    img.form__icon(src="data:image/svg+xml")
                    span.form__span or use your email account
                    input.form__input(type="text", placeholder="Email")
                    input.form__input(type="password", placeholder="Password")
                    a().form__link Forgot your password?
                    button().form__button.button.submit SIGN IN

            div.switch#switch-cnt
                div.switch__circle
                div.switch__circle.switch__circle--t
                div.switch__container#switch-c1
                    h2.switch__title.title Welcome Back !
                    p.switch__description.description To keep connected with us please
login with your personal info
                    button().switch__button.button.switch-btn SIGN IN

                div.switch__container#switch-c2.is-hidden
                    h2.switch__title.title Hello Friend !
```

```
          p.switch__description.description Enter your personal details and start journey with us
          button().switch__button.button.switch-btn SIGN UP
    script(src="main.js")
```

Resgister.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 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">
 {{ msg }}</div>
 </div>
</body>
</html>
```

13.2 Github & Project demo link

Github link : https://github.com/IBM-EPBL/IBM-Project-50460-1660910166

Project demo link : https://drive.google.com/file/d/1CPOUmyIHkwP8TM8JxiRmN36_fEeB1VEd/view?usp=drivesdk