

# **R.M.K. ENGINEERING COLLEGE**

**(An Autonomous Institution)**

**R.S.M. Nagar, Kavaraipettai - 601 206, Gummidipoondi Taluk,  
Thiruvallur District.**

## **PROJECT**

### **PLASMA DONOR APPLICATION**

#### **DONE BY**

**TEAM ID: PNT2022TMID15585**

**Abbireddy Sai Jahnavi  
Anandhitha TR  
Chrissy Panfila V  
Sravya Sravani**

## INDEX

S.No.	Description	Page No.
1	INTRODUCTION	
1.1	Project Overview	
1.2	Purpose	
2.	LITERATURE SURVEY	
2.1	Existing System	
2.2	References	
2.3	Problem Statement	
3.	IDEATION AND PROPOSED SOLUTION	
3.1	Empathy Map Canvas	
3.2	Ideation and Brainstorming	
3.3	Proposed Solution	
3.4	Problem Solution Fit	
4.	REQUIREMENT ANALYSIS	
4.1	Functional Requirement	
4.2	Non-Functional Requirement	
5.	PROJECT DESIGN	
5.1	Data Flow Diagram	
5.2	Solution and Technical Architecture	
5.3	User Stories	
6.	Project Planning and Scheduling	
6.1	Sprint Planning and Estimation	
6.2	Sprint Delivery Schedule	
6.3	Report from JIRA	
7.	CODING AND SOUTION	
7.1		
7.2		
8.	TESTING	
8.1	Test Cases	
8.2	User Acceptance Testing	
9.	RESULTS	
9.1	Performance Metrics	
10.	ADVANTAGES AND DISADVANTAGES	
11.	CONCLUSION	
12.	FUTURE SCOPE	

# 1. INTRODUCTION

## 1.1 Project Overview

Traditionally, in order to find a plasma patient, one should look at hospital records and contacting donors who have been recovered. Unfortunately, sometimes those people may not be available at home or would have relocated. In this situation, it is difficult to find the required person in case of emergency requirement. If we consider the recent Covid19 calamity, it is required to find donors as early as possible. In this project, such a solution is presented. As the backup system is weak, this method is not considered as a rapid process to find plasma. The solution presented in this project will help the persons who want to donate plasma to upload their Covid19 traced certificate and can donate the plasma to the blood bank. On the other hand, the blood bank can invite for the donor and once the donor has accepted the request, the blood bank can add the units they required. At the same time, the hospital can also send the request to the blood bank that urgently needs the plasma for the patient and can take the plasma from the blood bank.

In this project, on receipt of the plasma from the donor, the blood bank verifies the donor's certificate and can make a request to the donor. If the donor accepts the request, they can add the required number of units. The hospital can send a request to the blood bank that needs the patient's emergency plasma and to get the plasma from the blood bank.

Following are the sequence of events that will happen during the function of this application.

1. On giving the URL to run the application, the system will ask for login details.
2. A new user needs to register for this and an old user can use the same credentials.
3. The blood bank Admin can view the existing and valid donor details and can make request to specific donors.
4. Donors also can view these requests from their login, and on the acceptance of any request, the blood bank checks the status of this request.
5. The status will change as "pending" or "approved" based on the acceptance and rejection of the request by the donor.
6. The blood bank acts as an interface between the hospital and the donor, and there is no direct communication between the donor and the hospital.

## 1.2 Purpose

The main objective of this application is to design an user-friendly web application. This application tries to reduce mortality or help those affected by Covid19 by donating plasma from patients who have recovered without approved antiretroviral therapy planning for a

deadly Covid19 infection. It should be noted that plasma therapy is an experimental approach to treat these Covid19-positive patients and help them recover faster. This therapy is considered reliable and safe. If a particular person has fully recovered from Covid19, they are eligible to donate their plasma.

## **2. LITERATURE SURVEY**

### **2.1 Existing System**

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. Traditionally, in order to find a plasma patient, one should look at hospital records and contacting donors who have been recovered. Unfortunately, sometimes those people may not be available at home or would have relocated. As the backup system is weak, this method is not considered as a rapid process to find plasma. In regard to the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

Cloud computing is nothing but internet-based computing which made revolution in today's world. It is the biggest innovation which uses advanced computational power and improves data sharing and data storing capabilities. Cloud is a large group of interconnected computers, which is a major change in how we store information and run application. Cloud computing is a shared pool of configurable computing resources, on demand network access and provisioned by the service provider. The advantage of cloud is cost savings. The prime disadvantage is security. The cloud computing security contains a set of policies, technology controls deployed to protect data, application the associated infrastructure of cloud computing.

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.

The following technology have been used:

- Frontend: HTML, CSS, React.JS
- Backend: NodeJs, Express.Js

- Database: MongoDB

The following are this project's objectives:

- To provide a common platform for Plasma, Mother's milk and Blood Donors, Recipients and Health care system.
- To track the location of Donors using Maps and Send quick messages via short message service.
- To advertise and effectively manage Blood/Plasma/Mother's milk camps, To provide scalability and ensure security
- To provide health tips and guidelines about PMB donation.

The planned system consists of the subsequent modules: Admin Module, PMB Bank Module, Donor Module, Recipient Module.

This project is based on Android OS version 6.0+ developed in Android Studio, which enables developers to create high quality applications for Android devices. It is an Android application that uses the Firebase & 000webhost.blogs (web cloud) real-time database to quickly and efficiently search, collect and sort data for each Plasma, mother's milk and blood donor and users.

## 2.2 References

The following are the references that were referred to gain knowledge:

1. In year 2015, an IEEE paper on A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor and Intelligent Medicine Box was authored by Geng Yang, Li Xie, Matti Mantysalo, Xiaolin Zhou, Zhibo Pang, Li Da Xu, Sharon KaoWalter, Qiang Chen, Lirong Zheng. In this paper, an intelligent home-based healthcare platform is proposed and implemented. It involves iMedBox with connectivity, iMedPack with communication capability enabled by RFID, Bio-Patch and SOC. It fuses with IoT. The body-worn Bio-Patch can detect and transmit the user's bio-signals to the iMedBox in real time. The only limitations are, comprehensive platform missing. And the Physical size, rigid nature and short battery become limitation for long term use.
2. In 2016, an IEEE paper was authored on Data Mining for Better Healthcare: A Path towards Automated Data Analysis? By Tania Cerquitelli, Elena Baralis, Lia Morra and Silvia Chiusano. This paper addresses the mining activity from the medical database perspective. The mining system should be able to devise which knowledge could be most interesting to the user extract actionable knowledge from large medical dataset with minimal user intervention. System should be capable of yielding actionable knowledge extracting manageable sets. Large parameter spaces need to be explored at abstraction level to envision a system capable of evaluating and comparing many data-mining technique configurations at a time.
3. In 2015, a IEEE paper on Mobile Based Healthcare Management using Artificial Intelligence was authored by Amiya Kumar Tripathy, Rebeck Carvalho, Keshav Pawaskar, Suraj Yadav, Vijay Yadav. In this paper, the health-care management system is proposed which will consist of mobile based heart rate measurement so that

the data can be transferred and diagnosis based on heart rate can be provided quickly with a click of button. The system will consist of video conferencing to connect remotely with doctor. The system will also consist of Doc-Bot and an online Blood Bank. In this implemented project, heart rate calculation differs from actual one due to noise present in input signal. So, the performance is not efficient in practical. Methodology used Clustering, Text Mining, Pattern Matching, Support Vector Machine, Partitioning Algorithm and DonorHART tool used in collecting donor reaction information. Limitations are Difficulty in handling emergency situation and no proper security for personal details misuse.

### 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 a 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.

Who does the problem affect?	Every citizen who wants to live disease-free.
What are the boundaries of the problem?	Boundaries can be set around every individual who wants to keep their health profile balanced.
What is the issue?	Requirement for plasma is high and trusting the donor is a Challenge.
When does the issue occur?	When the donor is fake and when the vaccination camp is not laid by the government well.
Where is the issue occurring?	With the application being less secure that every other individual can get in without any authentication.
Why is it important that we fix the problem?	To ensure individuals are getting information regarding plasma that they can trust blindly without second thought, our application helps in authentication of the donor.

People living in different regions find it difficult to get plasma from the potential donors within a stipulated time. So, after getting the details of all the available donors, we have to find the donors who live in close proximity to the person who requires the plasma.

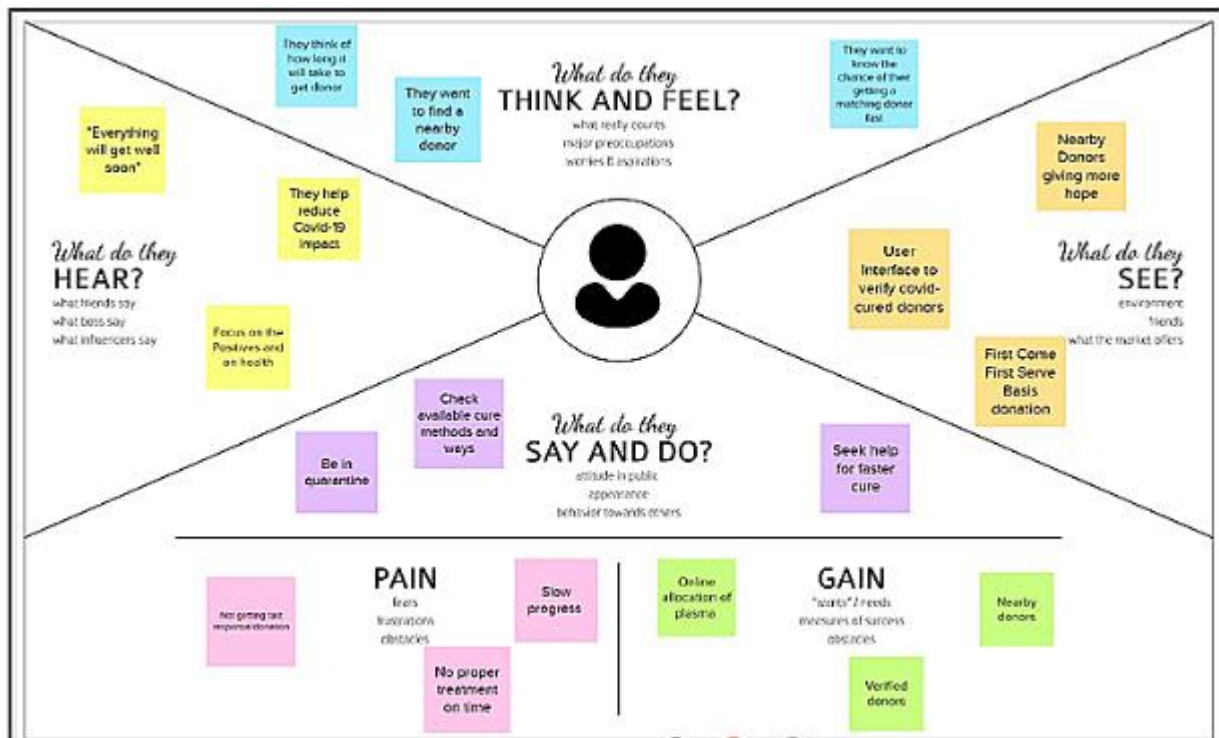
Who does the problem affect?	People who require plasma within a short duration of time.
What are the boundaries of the problem?	Boundaries can be set around people who need plasma.
What is the issue?	To find donors who live nearby to save time and resources.
When does the issue occur?	When the people who need plasma live in a remote region, the donor will not be able to reach out to them.
Where is the issue occurring?	Occurs when there is no proper information about the location of the donors.
Why is it important that we fix the problem?	To make it easier for the donors to reach the people who need plasma.

### 3. IDEATION & PROPOSED SOLUTION

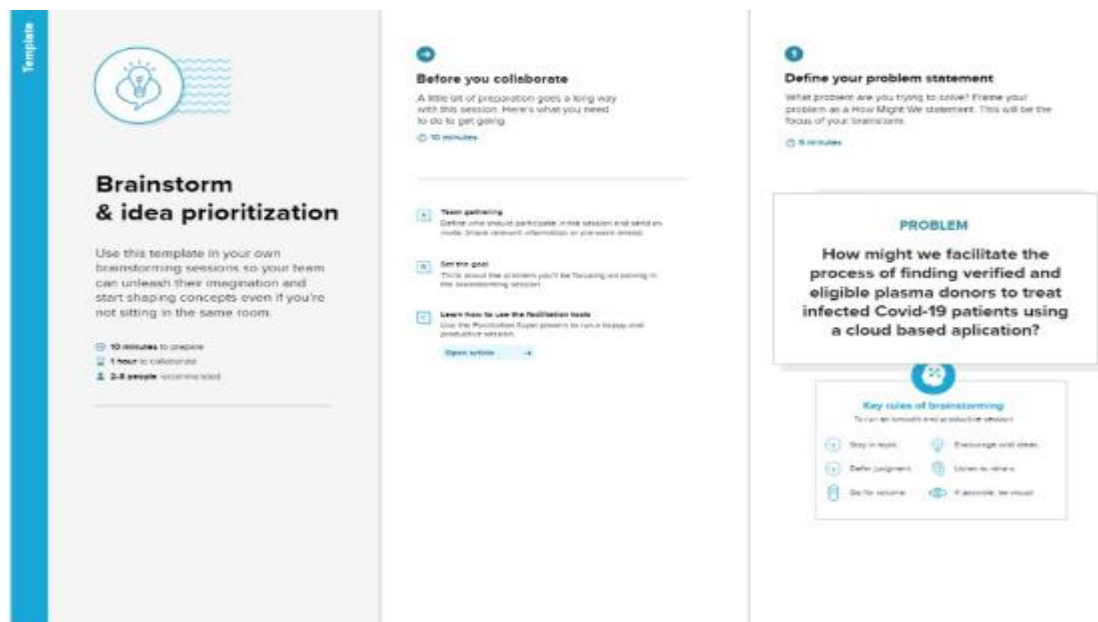
#### 3.1 Empathy Map Canvas

USER PERSONA: The user is tested positive for Covid-19 and is searching a donor (who is covid- cured or a matched donor) for plasma treatment, which is still experimental.

The following empathy map shows few details that has been developed by empathizing the user persona. How they feel, what they hear, what they gain from Plasma Donor Application, what are their fears- such details have been mentioned.



### 3.2 Ideation & Brainstorming





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 facilitate the process of finding verified and eligible plasma donors to treat infected Covid-19 patients using a cloud based application?



#### Key rules of brainstorming

To run a smooth and productive session

- Stay on topic
- Encourage wild ideas
- Defer judgment
- Listen to others
- Be fun and positive
- If possible, be visual

2

### Brainstorm

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

⌚ 10 minutes

**TIP**  
You can separate sticky notes by color or shape to make it easier to group them later.

#### ANANDHITHA

Secure system  
Eligible donors (with blood group)  
Stage-wise progress tracking  
Verified Donors

#### SAI JAHNVI

Nearby Donors  
Proper data update  
Requesting And Accepting donation online  
User Profile

#### CHRISSEY PANFILA

Certificate Of Covid tested Negative  
Filter by blood group  
Good recommendations  
Cloud Storage

#### SRAVYA SRAVANI

Plasma treatment awareness  
Plasma Donation history  
Connecting hospitals  
Filter based on age group

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 can 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.

#### Donor verification

Eligible donors (with blood group)  
Verified Donors  
Certificate Of Covid tested Negative

#### Performance Metrics

Secure system  
Cloud Storage  
Proper data update

#### Application Features

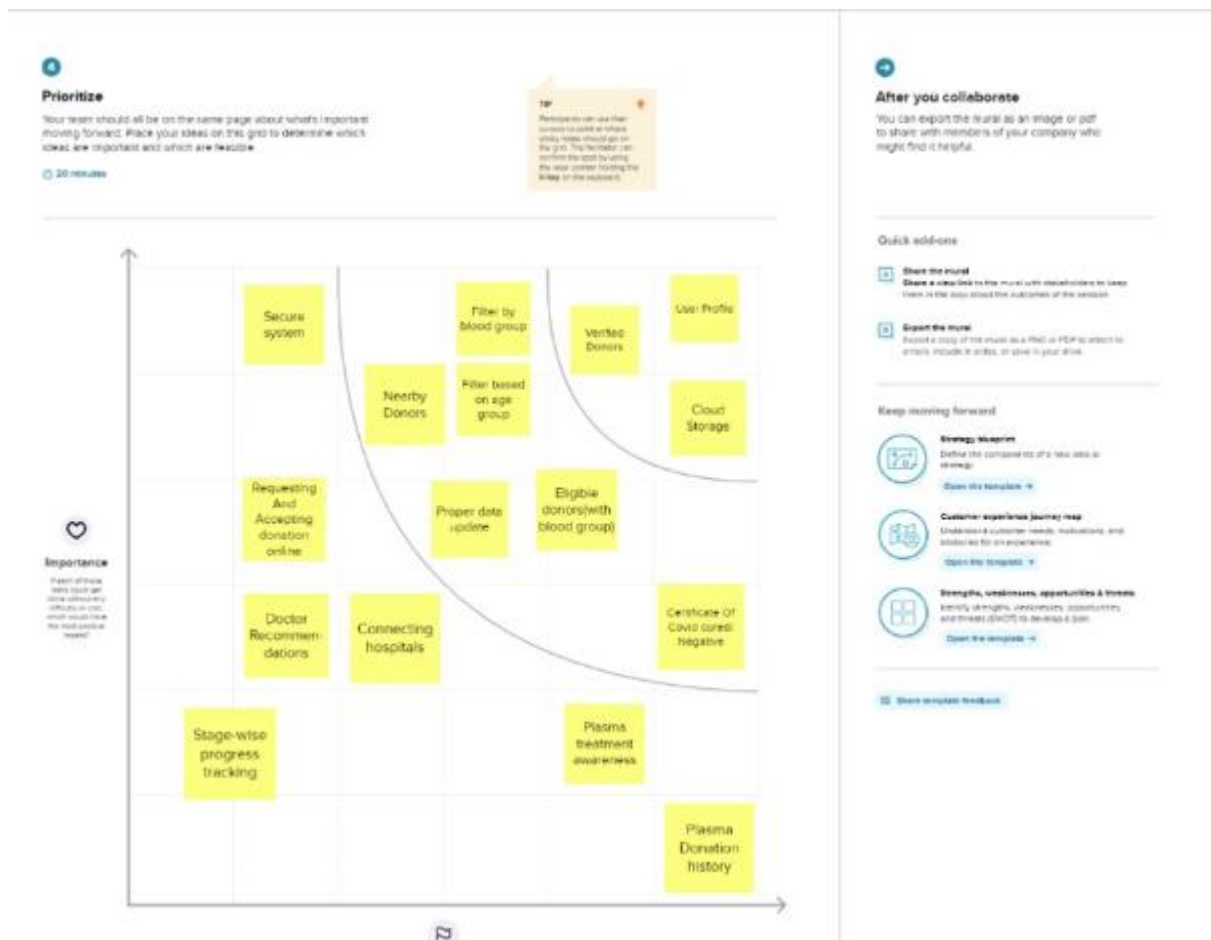
Stage-wise progress tracking  
Filter by blood group  
Connecting hospitals  
Requesting And Accepting donation online  
Filter based on age group

#### User Details

User Profile  
Plasma Donation history

#### Information

Nearby Donors  
Donor Recommendations  
Plasma treatment awareness



### 3.3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop a cloud based application to take plasma donor details,store them and inform them upon a request-facilitate finding verified and eligible donors.
2.	Idea / Solution description	The project idea is to store user details upon registration and using that find eligible, verified and nearby donors.Non-donor User can send request and donor-user can accept it only if eligible.May include hospital details to perform the donation and view the progress at all times.
3.	Novelty / Uniqueness	Nearby and verified donors are shown according to particular user. Progress of the donation process is shown.

4.	Social Impact / Customer Satisfaction	Facilitates more plasma treatment counts to reduce covid-19 . By the presence of only covid- negative certified donors, the user seeking plasma treatment has better trust and confidence.
5.	Business Model (Revenue Model)	Revenue generation can be through user registration under the non-donor field.
6.	Scalability of the Solution	Cloud Application allows flexible horizontal and vertical scaling.

### 3.4 Problem Solution fit

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> People who have been diagnosed with Covid-19. <b>CS</b> People who are looking for alternate treatment methods. Hospital management and health care facilities requiring plasma.	<b>6. CUSTOMER CONSTRAINTS</b> <b>CC</b> What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b> User can get donor through health care management and have to wait without any particular details.	Explore AS, understand RC
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>J&amp;P</b> Providing an intermediate between donor and the people in need for plasma. Availing appropriate details of donors based on location and availability.	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b> High demand for fast and promising treatment to covid-19. The solution solves the need for a fast acquirement of donors and sharing information easily.	<b>7. BEHAVIOUR</b> <b>BE</b> Customer looks for donors and verifies matching donors for covid-19 negative and all medical preferences. User requests for donor and waits.	
Focus on J&P, tap into BE, understand RC	<b>3. TRIGGERS</b> <b>TR</b> Looking for alternate and hopeful treatments for covid-19. Success stories of plasma treatment on News and from acquaintances.	<b>10. YOUR SOLUTION</b> <b>SL</b> By creating a web application that gives updated details of available donors nearby, we provide a intermediate for hospitals and patients and also indirectly give emotional support by providing true information.	<b>8. CHANNELS of BEHAVIOUR</b> <b>CH</b> <b>8.1 ONLINE</b> The user uses online website for searching and requesting plasma donors. They have option to view updated progress anytime. <b>8.2 OFFLINE</b> It is not very helpful as user needs to check in with hospital for any update.	
	<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b> <b>Before:</b> Anxious and impatient to get donor details. <b>After:</b> Focused and clear as they get clear ideas about the availability of donors.			

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional Requirements

The suggested solution's functional requirements are listed below.

FR No.	Functional Requirement	Sub Requirement
FR1	User Registration	This is a form that collects information from user.
FR2	Login	You will need to enter your username and password here.
FR3	Calendar	The user must be able to add the information of their blood group and can find people who can donate.
FR4	Plasma Application	Donors should be represented whether they have recovered from covid and only those should be allowed.
FR5	Report generation	The report must be represented in a Tabular Format.
FR6	Category	Users of this application will be able to add review donors.

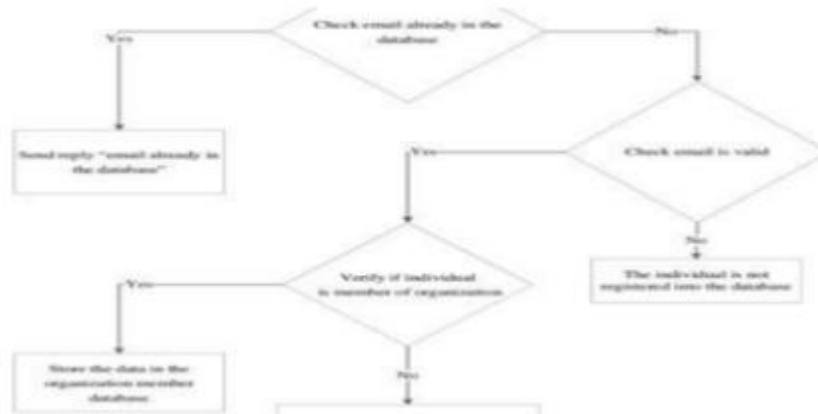
### 4.2 Non-functional requirements

The non-functional requirements of the proposed solution are as follows.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	keeps an accurate record of donors without any fake information.
NFR-2	Security	a detailed list of donors who are willing to.
NFR-3	Performance	There are large number of donors as well as an option. Because of lightweight database support, the system's throughput is increased.
NFR-4	Availability	The application must be completely operational at all places.
NFR-5	Scalability	The application must always function in its entirety.

## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams

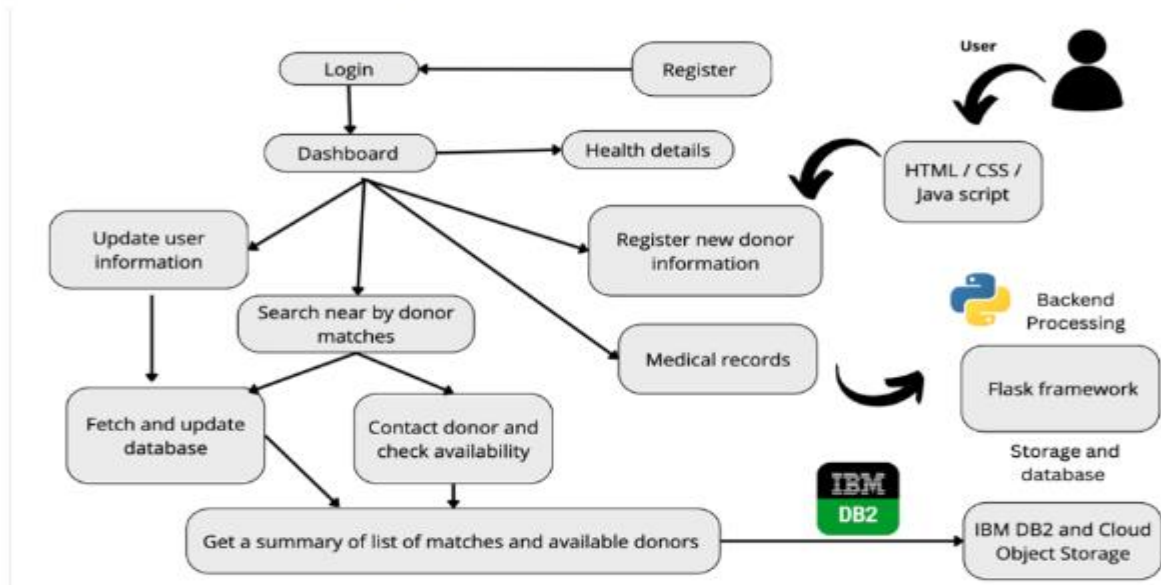


### 5.2 Solution & Technical Architecture

#### Solution Architecture:

- A new user registers in the portal by entering the personal information.
- The user can login and view the dashboard.
- From the dashboard the user can edit the personal information and update the health status.
- The available donors can be fetched and sorted based on the compatibility and location.
- The donors can also register in the website and enter their location and medical details.
- The progress can be tracked in the dashboard and any updates in the process can be notified through emails.

## Solution Architecture Diagram:



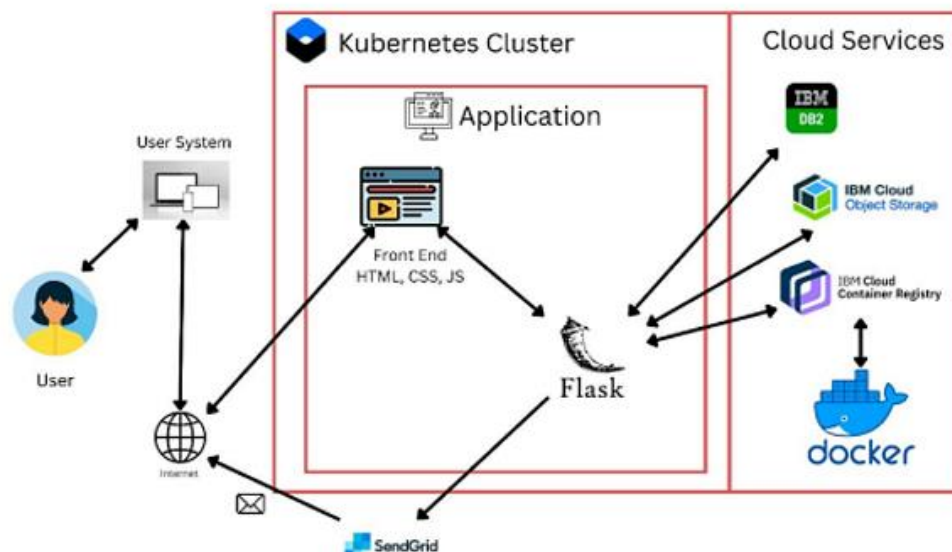
## Technical Architecture Diagram:

### Technical Architecture:

S.No	Component	Description	Technology
1.	User Interface	For user onboarding such as Login and Dashboard functions	HTML, CSS, JavaScript
2.	Verifying and filtering matching donors	Database operations to get data and perform operations and give user the appropriate details	Python
3.	IBM Watson	Chatbot to enable instant help for user	Watson Assistant by IBM
4.	Database	Stores all data including donor and user information	MySQL, NoSQL.
5.	Cloud Database	Cloud is used to store all the data in the database for elasticity and security	IBM DB2, IBM Cloudant.
6.	File Storage	File storage requirements have to be met here	IBM Cloud object storage
7.	External API: To send email SendGrid	Notifying users through e-mail when required to pass critical information	SendGrid
8.	Infrastructure (Server / Cloud)	For Application Deployment in Cloud	IBM - Docker – container, Cloud Foundry, Kubernetes container



## Technical Architecture Diagram:



## 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user & web user )	Registration	USN -1	As a user, I can register for the application by entering my email, and password, and confirming my password.	I can access my account/dashboard	High	Sprint -1
		USN -2	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm	High	Sprint -1
		USN -3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint -2
		USN -4	As a user, I can register for the application through a Google account.	I can register & access the dashboard with a Google Account login.	Medium	Sprint -1
	Login	USN -5	As a user, I can log into the application by entering my email & password	I can access the application.	High	Sprint -1
	Dashboard	USN -6	As a user, I can see the donors list up to date.	I can view the donors list and authenticate them.	High	Sprint -1
Customer Care Executive		USN -7	As a customer care executive, I can solve the problem that customers face.	I can provide support to customers at any time 24*7.	Medium	Sprint -1
Administrator	Application	USN -8	As an administrator, I can upgrade or update the application.	I can fix any bugs raised by customers and upgrade the application.	Medium	Sprint -1

## 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	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Chrissy Panfila V
Sprint-1	Registration verification	USN-2	As a user, I will receive confirmation email once I have registered for the application	2	Medium	Anandhitha TR
Sprint-2	Registration from Facebook	USN-3	As a user, I can register for the application through Facebook	2	Low	Anandhitha T R
Sprint-2	Registration from Gmail	USN-4	As a user, I can register for the application through Gmail	2	Low	Chrissy Panfila V
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email &	2	High	Chrissy Panfila V



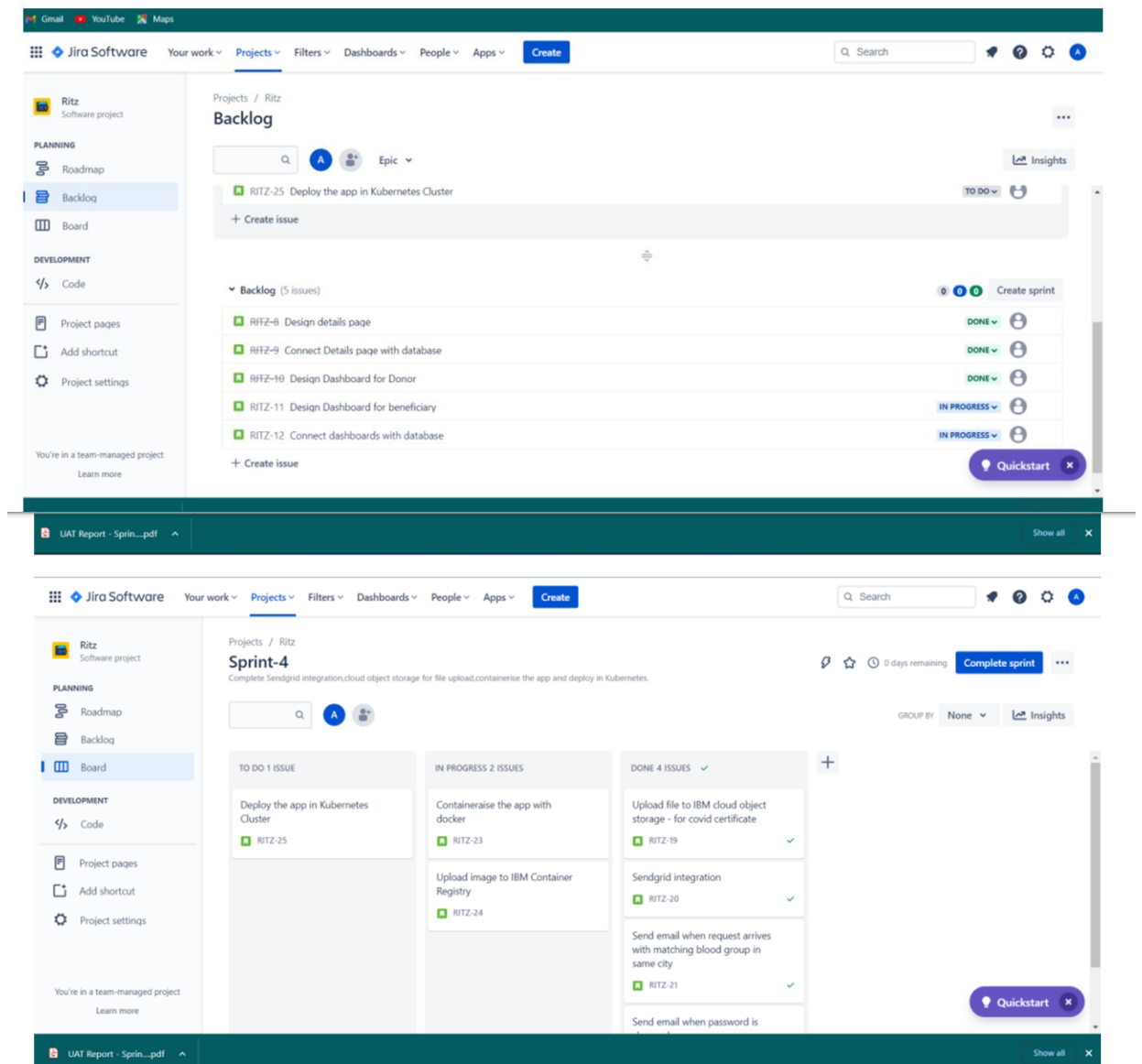
			password			
Sprint-3	Dashboard	USN-6	As a user, I can view donor details in dashboard	5	Very High	Chrissy Panfila V, Anandhitha TR
Sprint-3	Add to list	USN-7	As a user, I can add donors to list and send request to them	2	High	Sai Jahnvi, Sravya a Sravan Chrissy Panfila V, Anandhitha TR
Sprint-4	Notify request	USN-8	As a user, I can get email notification when any request arrives for me.	5	Very high	Sai Jahnvi, Sravya Sravani
Sprint-3	Dashboard	USN-9	As a user, I can view donors by filtering based on location and age	2	High	Anandhitha TR, Chrissy Panfila
Sprint-2	Profile	USN-10	As a user, I can edit my details from profile	2	High	Chrissy Panfila V, Sai Jahnvi
Sprint-3	Bot	USN-11	As a user, I can get frequently asked questions answered by a bot.	2	Medium	Sravya Sravani, Anandhitha TR

## 6.2 Sprint Delivery Schedule

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	29 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	23 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	28 SEPTEMBER 2022

<b>Customer Journey</b>	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
<b>Functional Requirement</b>	Prepare the functional requirement document.	8 OCTOBER 2022
<b>Data Flow Diagrams</b>	Draw the data flow diagrams and submit for review.	9 OCTOBER 2022
<b>Technology Architecture</b>	Prepare the technology architecture diagram.	10 OCTOBER 2022
<b>Prepare Milestone &amp; Activity List</b>	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
<b>Project Development - Delivery of Sprint-1, 2, 3 &amp; 4</b>	Develop & submit the developed code by testing it.	WORK IN PROGRESS

### 6.3 Reports From JIRA:



## 7. CODING AND SOUTION

### 1.LOGIN PAGE

Sign In

Email

Password

Submit

New User? [SignUp](#)



2.SIGNUP PAGE

Sign Up

Name

Email

Password

Retype Password

I am a:

Beneficiary

Donor

Submit

Already a User? [Login](#)

X Close

Hi,I'm Laura!Can I help you ?

### 3.PROFILE – ACCOUNT SETTINGS PAGE


Donor App

Logout

HOME DASHBOARD PROFILE ABOUT CONTACT

Login successful..

## Account Settings



Janu

- Account
- Details

### Account Settings

First Name

Janu

Last Name

Email


janu@gmail.com

Phone number

8765423623

Hi,I'm Laura!Can I help you ?

Close



Janu

- Account
- Details
- Password
- Notification

### Account Settings

First Name

Janu

Last Name

Email

janu@gmail.com

Phone number

8765423623

Address

None

City

Tiruchy

Bio

None


Submit

Hi,I'm Laura!Can I help you ?

Close

## 4.PROFILE-DETAILS PAGE

### Account Settings



Janu

[Account](#)  
[Details](#)  
[Password](#)  
[Notification](#)

#### Details

I am a : Beneficiary


Height	Weight
<input type="text" value="None"/>	<input type="text" value="None"/>
Blood Group:O+	Age
<input type="text" value="Select"/>	<input type="text" value="34"/>

Update

Choose File


No file chosen

Submit



## 5.PASSWORD CHANGE PAGE

### Account Settings



Janu

- Account
- Details
- Password**
- Notification

#### Password Settings

Old password

New password

Confirm new password

Submit

## 6.DASHBOARD-FOR BENEFICIARIES

HOME DASHBOARD PROFILE ABOUT CONTACT

Welcome Janu bene

Beneficiary Dashboard

My Details

Name	Janu
Blood Group	O+
Email	janu@gmail.com
Location	Tiruchy
Phone	8765423623

Request Donors



## 7.DASHBOARD FOR DONORS

Donor App

Logout

**HOME   DASHBOARD   PROFILE   ABOUT   CONTACT**

Welcome Test donor

All Requests

Saved Requests

My-Dashboard

Requests

First Name	Last Name	EMAIL	PHONE	ADDRESS	CITY	AGE	BLOODGROUP	Covid-Certificate-uploaded
------------	-----------	-------	-------	---------	------	-----	------------	----------------------------



## 8.TESTING

### 8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
LoginPage_TC_OO1	Functional	Login page	Verify user is able to see the Login page	1.Enter URL and click go. 2.Verify login page is displayed or not.	<a href="http://127.0.0.1:5000/">http://127.0.0.1:5000/</a>	Login page should display	Working as expected	Pass	Abbireddy Sai Jahnavi
LoginPage_TC_OO2	UI	Login page	Verify the UI elements in Login page	1.Enter URL and click go. 2.Verify login page with below UI elements: a.email text box b.password text box c.Submit button d.New user? Signup link	<a href="http://127.0.0.1:5000/">http://127.0.0.1:5000/</a>	Application should show below UI elements: a.email text box b.password text box c.Submit button d.New user? Signup	Working as expected	Pass	Anandhitha TR
LoginPage_TC_OO3	Functional	Login page	Verify user is able to log into application with Valid credentials	1.Enter URL and click go. 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button	Username: testuser@gmail.com password: testuser-1	User should navigate to user account homepage	Working as expected	Pass	Chrissy Panfila V

LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Invalid credentials	1.Enter URL and click go 2.Enter Invalid username/email in Email text box 3.Enter valid password in password text box 4.Click on login button	Username: abc@gmail.com password: testuser-1	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Sravya Sravani
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with Invalid credentials	1.Enter URL and click go 2.Enter Valid username/email in Email text box 3.Enter Invalid password in password text box 4.Click on login button	Username: testuser@gmail.com password: test1234	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Abbireddy Sai Jahnavi
LoginPage_TC_OO6	Functional	Login page	Verify user is able to log into application with Invalid credentials	1.Enter URL and click go 2.Enter Invalid username/email in Email text box 3.Enter Invalid password in password text box 4.Click on login button	Username: abc@gmail.com password: test1234	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Chrissy Panfila V
LoginPage_TC_OO7	Functional	Login Page	Check if signup link takes the	1.Enter URL and click go 2. Click on the signup		The signup page should be displayed	Working as expected	Pass	Anandhitha TR

			user to the signup page	link					
LoginPage_TC_OO8	UI	Signup page	Verify the UI elements in Signup page	1.Enter URL and click go 2. Click on the signup link 3. Check if the following UI elements are present: a. Name textbox b. email textbox c. password textbox d. retype password textbox e. Beneficiary/Donor radio button f. Submit button g. Already a user? Login link		Application should show below UI elements: a. Name textbox b. email textbox c. password textbox d. retype password textbox e. Beneficiary/Donor radio button f. Submit button g. Already a user? Login link	Working as expected	Pass	Chrissy Panfila V
LoginPage_TC_OO9	Functional	Signup page	Verify if the user is able to signup with invalid email	1.Enter URL and click go 2. Click on the signup link 3. Enter an invalid email address 4. Fill other details 5. Click submit	name: Test User email: abc@gmail.com password: test1234 retype-password: test1234	An error message stating the email is invalid should be displayed	Working as expected	Pass	Sravva Sravani
LoginPage_TC_O10	Functional	Signup page	Verify if the user is able to	1.Enter URL and click go 2. Click on the signup	name: Test User email:	An error message stating the passwords do not	Working as expected	Pass	Anandhith a TR

			signup with passwords that do not match	link 3. Type a password 4. Type a different password in retype-password field	abc@gmail.com password: test1234 retype-password: test123456	match should be displayed			
LoginPage_TC_O11	Functional	Signup page	Check if the users details are uploaded correctly in the database	1.Enter URL and click go 2. Click on the signup link 3. Enter all required details 4. Click on submit	name: Test User email: abc@gmail.com password: test1234 retype-password: test1234	The users signup details should be entered in the database	Working as expected	Pass	Sravva Sravani
LoginPage_TC_O12	Functional	Signup page	After signup the user should be redirected to login page	1.Enter URL and click go 2. Click on the signup link 3. Enter all required details 4. Click on submit		The user should be redirected to login page after signup	Working as expected	Pass	Abbireddy Sai Jahnavi
LoginPage_TC_O13	Functional	Home page	After login the user should be directed to home page	1.Enter URL and click go. 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button.		The user should be directed to home page after login	Working as expected	Pass	Chrissy Panfila V
LoginPage_	UI	Home page	Verify the	1.Enter URL and click		Application should	Working	Pass	Abbireddy

TC_O14		ge	UI elements in Home page	go 2. Login into the website 3. Check if the following UI elements are present: a.App name b. logout button c.Navigation bar		show below UI elements: a.App name b. logout button c.Navigation bar	as expected		Sai Jahnnavi
LoginPage_ TC_O15	Functional	Home page	Check if the logout button logs the user out	1.Enter URL and click go. 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button. 5. Click on logout button in homepage.		The user should be logged out of the application and redirected to login page	Working as expected	Pass	Sravva Sravani
LoginPage_ TC_O16	Functional	Dashboard	Check if navigation link to dashboard is working	1.Enter URL and click go. 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button. 5. Click on dashboard in navbar		The dashboard should be displayed	Working as expected	Pass	Chrissy Panfila V
LoginPage_ TC_O17	Functional	Profile page	Check if navigation	1.Enter URL and click go.		The profile should be displayed	Working as	Pass	Anandhitha TR

			link to profile page is working	2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button. 5. Click on profile in navbar			expected		
LoginPage_ TC_O18	Functional	Contact page	Check if navigation link to contact page is working	1.Enter URL and click go. 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button. 5. Click on contact in navbar		The contact page should be displayed	Working as expected	Pass	Sravva Sravani
LoginPage_ TC_O19	Functional	About page	Check if navigation link to about us page is working	1.Enter URL and click go. 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button. 5. Click on about in navbar		The about page should be displayed	Working as expected	Pass	Chrissy Panfila V
LoginPage_ TC_O20	Functional	Profile page	Check if the password	1.Go to profile page 2. Go to password tab	Old password: test1234	The password should not change and display error	Working as expected	Pass	Anandhitha TR

			change is working with wrong password	3.Enter wrong password in password field 4.Fill other details 5.Click on Submit button	New password: test-abcd Confirm password: test-abcd	message			
LoginPage_TC_O21	Functional	Profile page	Check if the password change is working with correct credentials	1.Go to profile page 2. Go to password tab 3.Enter old password in password field 4.Type new password and confirm 5.Click on Submit button	Old password: testuser-1 New password: test-abcd Confirm password: test-abcd	The password should be changed and updated in database	Working as expected	Pass	Abbireddy Sai Jahnavi
LoginPage_TC_O22	Functional	Contact page	Check if the message entered in contact page is delivered correctly	1.Enter URL and click go. 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button. 5. Click on contact in navbar	Hello I have a problem with contacting the donor	The message should be stored in database	Working as expected	Pass	Chrissy Panfila V
ProfilePage_TC_OO1	Functional	Profile page	Verify user is able to see the Profile page	1.Enter URL and click go. 2.Login with valid credentials. 3. Click on profile tab	<a href="http://127.0.0.1:5000/">http://127.0.0.1:5000/</a>	Profile page should display	Working as expected	Pass	Abbireddy Sai Jahnavi
ProfilePage_TC_OO2	Functional	Profile page	Check if the user is	1.Enter URL and click go.	Firstname: Test	All the details should be updated	Working as	Pass	Anandhitha TR

			able to update the details in account settings	2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in account settings. 5. Click on submit.	Lastname: User Phone:999 9999999 Address: 20, First Lane, Chromepet City: Chennai	in the accounts page	expected		
ProfilePage_TC_OO3	Functional	Profile page	Check if the details are stored in the database	1.Enter URL and click go. 2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in account settings. 5. Click on submit.	Firstname: Test Lastname: User Phone:999 9999999 Address: 20, First Lane, Chromepet City: Chennai	All details should be available in the database	Working as expected	Pass	Chrissy Panfila V
ProfilePage_TC_OO4	Functional	Profile page	Check if the user is able to update the details in details tab of account settings	1.Enter URL and click go. 2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in details tab of account settings. 5. Click on update.	Height: 160 Weight: 55 Blood Group: O+ Age: 22 Illness: None	All the details should be updated in the details tab of accounts page	Working as expected	Pass	Sravya Sravani
ProfilePage_TC_OO5	Functional	Profile page	Check if the data in	1.Enter URL and click go.	Height: 160 Weight: 55	All details of details tab should be	Working as	Pass	Abbireddy Sai

			details tab of account settings is updated in the database.	2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in details tab of account settings. 5. Click on update.	Blood Group: O+ Age: 22 Illness: None	available in the database	expected		Jahnvi
ProfilePage_TC_OO6	Functional	Profile page	Check if the user is able to update the details in account settings with invalid data	1.Enter URL and click go. 2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in account settings. 5. Enter invalid phone number. 5. Click on submit.	Firstname: Test Lastname: User Phone:999 9999 Address: 20, First Lane, Chromepet City: Chennai	Application should show 'Incorrect phone number ' validation message.	Working as expected	Pass	Chrissy Panfila V
ProfilePage_TC_OO7	Functional	Profile page	Check if the user is able to update the details in details tab of account settings with invalid data.	1.Enter URL and click go. 2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in details tab of account settings. 5. Enter invalid height. 5. Click on update.	Height: 1600 Weight: 55 Blood Group: O+ Age: 22 Illness: None	Application should show 'Invalid height ' validation message.	Working as expected	Pass	Anandhitha TR
ProfilePage_TC_OO8	Functional	Profile page	Check if the user is	1.Enter URL and click go.	Height: 160 Weight:	Application should show 'Invalid	Working as	Pass	Chrissy Panfila V

			able to update the details in details tab of account settings with invalid data.	2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in details tab of account settings. 5. Enter invalid weight. 5. Click on update.	5500 Blood Group: O+ Age: 22 Illness: None	weight ' validation message.	expected		
ProfilePage_TC_OO9	Functional	Profile page	Check if the user is able to update the details in details tab of account settings with invalid data.	1.Enter URL and click go. 2.Login with valid credentials. 3. Click on profile tab. 4. Enter the details in details tab of account settings. 5. Enter invalid age. 5. Click on update.	Height: 160 Weight: 55 Blood Group: O+ Age: 2201 Illness: None	Application should show 'Invalid age ' validation message.	Working as expected	Pass	Shravya Shrivani
Dashboard_TC_OO1	Functional	Dashboar d	Verify if the donor dashboard is displayed for donors correctly.	1.Enter URL and click go. 2.Login as donor with valid credentials. 3. Click on dashboard tab		donor dashboard is displayed for donors	Working as expected	Pass	Anandhitha TR
Dashboard_TC_OO2	Functional	Dashboar d	Verify if the beneficiary dashboard is displayed for beneficiari	1.Enter URL and click go. 2.Login as beneficiary with valid credentials. 3. Click on dashboard tab		beneficiary dashboard is displayed for beneficiaries	Working as expected	Pass	Shravya Shrivani

			es correctly.						
Dashboard_TC_OO3	UI	Dashboar d	Verify if the donor details given are displayed correctly and verify if the request button is visible	1.Enter URL and click go. 2.Login as donor with valid credentials. 3. Click on dashboard tab 4.Check the details 5. Check for Request button.		Donor details are displayed correctly and request button is visible.	Working as expected	Pass	Abbireddy Sai Jahnavi
Dashboard_TC_OO4	UI	Dashboar d	Verify if the beneficiary details given are displayed correctly and verify if the accept button is visible	1.Enter URL and click go. 2.Login as beneficiary with valid credentials. 3. Click on dashboard tab 4.Check the details 5. Check for Accept button.		Beneficiary details are displayed correctly and accept button is visible.	Working as expected	Pass	Chrissy Panfila V
Dashboard_TC_OO5	Functional	Dashboar d	Verify if the request is stored in the database after the request button is clicked.	1.Enter URL and click go. 2.Login as beneficiary with valid credentials. 3. Click on dashboard tab 4. Click on Request button.		Request should be stored in database	Working as expected	Pass	Abbireddy Sai Jahnavi

Dashboard_TC_OO6	UI	Dashboar d	Verify if the stop requesting button is present after the request is made.	1.Enter URL and click go. 2.Login as beneficiary with valid credentials. 3. Click on dashboard tab 4. Click on Request button. 5.Check for stop requesting button.		Stop requesting button should be visible	Working as expected	Pass	Chrissy Panfila V
Dashboard_TC_OO7	UI	Dashboar d	Verify if all requests button is present in the donor dashboard.	1.Enter URL and click go. 2.Login as beneficiary with valid credentials. 3. Click on dashboard tab 4. Check if all requests button is present in the dashboard.		The all requests button should be present.	Working as expected	Pass	Chrissy Panfila V
Dashboard_TC_OO8	UI	Dashboar d	Verify if saved requests button is present in the donor dashboard.	1.Enter URL and click go. 2.Login as beneficiary with valid credentials. 3. Click on dashboard tab 4. Check if saved requests button is present in the dashboard.		The saved requests button should be present.	Working as expected	Pass	Anandhitha TR
Dashboard_TC_OO9	Functional	Dashboar d	Verify if the request is stored in the database after the request button is clicked.	1.Enter URL and click go. 2.Login as beneficiary with valid credentials. 3. Click on dashboard tab 4. Click on Request button.		All requests for	Working as expected	Pass	Abbireddy Sai Jahnavi

TC_O09		rd	the all requests button displays all the donor requests.	go. 2.Login as donor with valid credentials. 3. Click on dashboard tab 4. Click on all requests button present in the dashboard.		donor should be displayed.	as expected		Sai Jahnavi
Dashboard_TC_O10	Functional	Dashboa rd	Verify if the saved requests button displays all the donor requests.	1.Enter URL and click go. 2.Login as donor with valid credentials. 3. Click on dashboard tab 4. Click on saved requests button present in the dashboard.		All requests saved by the donor should be displayed.	Working as expected	Pass	Anandhith a TR
Dashboard_TC_O11	Functional	Dashboa rd	Check if the requests from beneficiary is displayed to the correct donor based on given criteria.	1.Enter URL and click go. 2.Login as donor with valid credentials. 3. Click on dashboard tab 4. Click on all requests button present in the dashboard.		The displayed beneficiaries should match the given criteria.	Working as expected	Pass	Chrissy Panfila V
Dashboard_TC_OO12	Functional	Dashboa rd	Verify if the request is cancelled	1.Enter URL and click go. 2.Login as donor with		The cancelled request should not be displayed.	Working as expected	Pass	Sravva Sravani

			when stop requesting button is clicked.	valid credentials. 3. Click on dashboard tab 4. Click on all requests button present in the dashboard.					
Dashboard_TC_O13	Functional	Dashboa rd	Check if save button saves the request in the database.	1.Enter URL and click go. 2.Login as donor with valid credentials. 3. Click on dashboard tab 4. Click on save button near a request.		Check if the request is saved in the database.	Working as expected	Pass	Abbireddy Sai Jahnavi
Dashboard_TC_O14	Functional	Dashboa rd	Check if saved requests are displayed when save button is clicked.	1.Enter URL and click go. 2.Login as donor with valid credentials. 3. Click on dashboard tab 4. Click on save button near a request. 5. Click on saved beneficiaries button.		All the saved requests should be displayed.	Working as expected	Pass	Chrissy Panfila V
IBMAssistant_TC_OO1	UI	Signup Page	Check if the IBM Assistant can be viewed in signup page.	1. Go to signup page.		IBM Assistant is displayed in the signup page.	Working as expected	Pass	Anandhith a TR



IBMAssistant_TC_OO2	UI	Login Page	Check if the IBM Assistant can be viewed in login page.	1. Go to login page.		IBM Assistant is displayed in the login page.	Working as expected	Pass	Chrissy Panfila V
IBMAssistant_TC_OO3	UI	Home Page	Check if the IBM Assistant can be viewed in home page.	1. Enter URL and click go. 2. Login as donor with valid credentials.		IBM Assistant is displayed in the home page.	Working as expected	Pass	Sravya Sravani
IBMAssistant_TC_OO4	UI	Dashboard	Check if the IBM Assistant can be viewed in dashboard page.	1. Enter URL and click go. 2. Login as donor with valid credentials. 3. Click on dashboard tab		IBM Assistant is displayed in the dashboard page.	Working as expected	Pass	Anandhitha TR
IBMAssistant_TC_OO5	UI	Profile	Check if the IBM Assistant can be viewed in profile page.	1. Enter URL and click go. 2. Login as beneficiary with valid credentials. 3. Click on profile tab		IBM Assistant is displayed in the profile page.	Working as expected	Pass	Sravya Sravani
IBMAssistant_TC_OO6	UI	Contact	Check if the IBM Assistant can be viewed in contact us page.	1. Enter URL and click go. 2. Login as beneficiary with valid credentials. 3. Click on contact tab		IBM Assistant is displayed in the contact page.	Working as expected	Pass	Abbireddy Sai Jahnvi

## 8.2 User Acceptance Testing

### Defect Analysis

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

#### **SPRINT – 1**

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	4	1	3	16
Duplicate	1	1	3	0	5
External	2	3	2	1	8
Fixed	12	2	3	19	36
Not Reproduced	0	0	1	0	1
Skipped	0	1	1	1	3
Won't Fix	0	5	2	0	7
Totals	23	16	13	24	76

## SPRINT – 2

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	5	0	3	15
Duplicate	2	0	3	1	6
External	2	2	1	1	6
Fixed	15	3	1	12	31
Not Reproduced	0	1	0	0	1
Skipped	0	1	3	1	5
Won't Fix	0	4	0	3	7
Totals	26	16	8	21	71

## SPRINT – 3

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	3	1	1	10
Duplicate	2	1	4	0	7
External	0	2	0	3	5
Fixed	14	4	0	18	36
Not Reproduced	0	0	1	0	1
Skipped	0	1	2	0	3
Won't Fix	1	1	0	1	3
Totals	22	12	8	23	65

## SPRINT – 4

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	6	3	1	15
Duplicate	0	1	2	0	3
External	2	1	1	1	5
Fixed	10	3	2	15	30
Not Reproduced	0	0	1	0	1
Skipped	0	1	1	0	2
Won't Fix	1	3	1	2	7
Totals	18	15	11	19	63

## Test Case Analysis

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

## SPRINT – 1

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	60	0	0	60
Security	4	0	0	4
Outsource Shipping	3	0	0	3
Exception Reporting	7	0	0	7
Final Report Output	4	0	0	4
Version Control	2	0	0	2

## SPRINT – 2

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	55	0	0	55
Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	6	0	0	6
Final Report Output	3	0	0	3
Version Control	2	0	0	2

## SPRINT – 3

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	3	0	0	3
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	4	0	0	4
Final Report Output	3	0	0	3
Version Control	2	0	0	2

## SPRINT – 4

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5ss
Client Application	27	0	0	27
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	7	0	0	7
Final Report Output	3	0	0	3
Version Control	2	0	0	2

## 10. ADVANTAGES AND DISADVANTAGES

The main advantage of this application is the centralized platform for the interaction of donors, blood bank and hospital. Further it provides an authentication phase in addition to the decision options by the donor and the hospital. Also the blood bank can provide this service to other hospitals also.

In spite of many advantages, this application comes with some disadvantages also. This application is not able to verify the genuineness of the user automatically. That is, it requires various entities and phases by the users to verify their integrity. Furthermore, it requires an active internet connection throughout its operation, which is considered as a constraint for this application.

## 11. CONCLUSION

Plasma donor application provides a reliable platform to connect local plasma donors with patients. This application creates a communication channel through authenticated hospitals whenever a patient needs plasma donation. It is a useful tool to find compatible blood donors who can receive plasma request posts in their local area. Hospitals can use this web application to maintain the blood donation activity.

## 12. FUTURE SCOPE

Plasma donor application can be further improved by providing a provision for user accessibility via integrating this application with various social network application program interface. Consequently, users can login and signup using various social networks. This would increase number of donors and enhances the process of plasma donation. User interface can be improved in future to accommodate global audience by supporting different languages across countries. Data scrapping can be done from different social networks and can be shown in the plasma request feeds.

## 13. APPENDIX

### Source code

```
window.watsonAssistantChatOptions = {  
    integrationID: "0b26d6c6-f0b0-4523-b321-521b3a1b752d", // The ID of this  
integration.  
    region: "eu-gb", // The region your integration is hosted in.  
    serviceInstanceID: "bef30b86-322f-496b-8084-24f492d6c1ba", // The ID of  
your service instance.  
    onLoad: function(instance) { instance.render(); }  
};  
setTimeout(function(){  
    const t=document.createElement('script');  
    t.src="https://web-  
chat.global.assistant.watson.appdomain.cloud/versions/" +  
(window.watsonAssistantChatOptions.clientVersion || 'latest') +  
"/WatsonAssistantChatEntry.js";  
    document.head.appendChild(t);  
});
```

## **14. GitHub Link and Project Demo Link**

GitHub Link:

<https://github.com/IBM-EPBL/IBM-Project-27658-1660062291>

Project Demo Link: