Plasma Donor Application

Team Members:

Darshan Sasikumar

Gowtham Sriram

Kevin Nelson

Pranavh Harish

Team ID: PNT2022TMID12996

Chapter 1
Introduction

1.1 Project Overview:

A way in which one can help the COVID 19 affected people is by donating Plasma from recovered patients. With no approved antiviral treatment plan for the deadly COVID-19 infection, plasma therapy is an experimental approach to treat COVID positive patients and help them recover faster. The therapy is considered to be safe and promising. If a particular person is fully recovered from COVID 19 he/she is eligible to donate their plasma. In the proposed system, donors who need to donate plasma can donate by uploading covid-19 certificate and blood bank can view donors and can raise requests to donors and the hospital can register/login and can search for plasma, they can raise requests to blood bank and can get the plasma.

1.2 Purpose:

The proposed work aims to create an application and provide an interface with which donors can contact hospitals and hospitals can in turn find donors to meet the plasma requirements. This will allow hospitals to meet the demands of plasma for the patients undergoing treatment. Plasma from Covid recovered patients can be used to treat covid patients. This allows an easy way for donors to interact with hospitals and to donate plasma when needed.

Chapter 2

Literature Survey:

S.No	Title	Author	Abstract	Drawbacks
1.	Instant Plasma Donor Recipient Connector Web Application	Ripathi S Kumar V Prabhakar A	One way in which we can help the COVID 19 affected people is by donating Plasma from recovered patients. With no approved antiviral treatment plan for the deadly COVID-19 infection, plasma therapy is an experimental approach to treat COVID positive patients and help them recover faster. The therapy considered to be safe and promising. If a particular person is fully recovered from COVID 19 he/she is applicable to donate their plasma. In the proposed system, donors who need to donate plasma can donate by uploading covid-19 certificate and blood bank can view donors and can raise requests to donors and the hospital can register/login and can search for plasma, they can raise requests to blood bank and can get the plasma.	• Tedious work. • Expensive. • Requires more man power. • Time Consuming.
2.	Plasma Donation Website using MERN stack	Neha Soni	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. • Internet: It would require an internet	Internet: It would require an internet connection for the working of the website. AutoVerification: It cannot automatically verify the genuine users.

			connection for the working of the website. AutoVerification: It cannot Patients can directly call the donor by taking his/her contact number from the application. The user can also view the total active cases, recovered cases, vaccine centres in their area, hospital location, and helpline number.	
3.	Developing a plasma donor application using Function-as-aservice in AWS	Aishwarya R Gowri	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 AWS services. The services used are AWS Lambda, API gateway, DynamoDB, AWS Elastic Compute Cloud with the help of these AWS services, it eliminates the need of configuring the servers and reduces the infrastructural costs associated with it and helps to achieve serverless computing. Situations like if the donor count is very low, it is 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.	• It cannot auto verify user genuineness. • It requires an active internet connection.
4.	Plasma	Jenny Shersten	Motivation for further plasma	Internet

	Donation App		collection from donors for recipients, as well as fast communication with them. For both groups - always up-to-date information and the ability to follow statistics and data in the city and in the country	Connection is mandatory • Reports are not verified
5.	Plasma-DonorApp	Dheeraj Kotwani, Pragathi Verma, Sitam Sardar, Vatsal Kesarwani, Nakul Sharma, Nuh Koca , Harsh Rajgor	An Open-Source App which fills the gap between the patients and the Plasma Donors.	□ No search filter available □ Cannot login through Chrome
6.	Blood bag: A web application to manage all blood donation and transfusion processes.	Ali, R. S., Hafez, T. F., Ali, A. B, Abd-Alsabour, N.	This aims to help users fulfill their needs for a safe and reliable blood group by searching for and locating a specific blood group and plasma donor. The paper illustrates the . Hence, we proposed the Blood Bag web-based application that is connected to a centralized database using SQL to gather and organize the data from all blood banks and blood donation campaigns. The proposed application organizes and controls the whole critical processes related to blood donation, testing and storage of blood bags, and delivering it to the patient. The application is developed using C#.	Major UI Improvements □ Uses Donation Camps and is not instant access
7.	Nearest Blood & Plasma Donor Finding: A Machine Learning Approach	Nayan Das, MD. Asif Iqbal	This paper details the dangers of not finding a suitable donor and the impact such a reliable can create. The purpose is to build a platform with clustering algorithms which will jointly help to provide the quickest solution to find blood or plasma donor. The paper uses Machine	□ High Computational Requirements □ Depends on API's for database updation

			learning techniques and makes use of existing API's for faster data retrieval. Closest blood or plasma donors of the same group in a particular area can be explored within less time and efficiently	

Chapter 3

Ideation and Proposed Solution:

3.1 Empathy Map Canvas

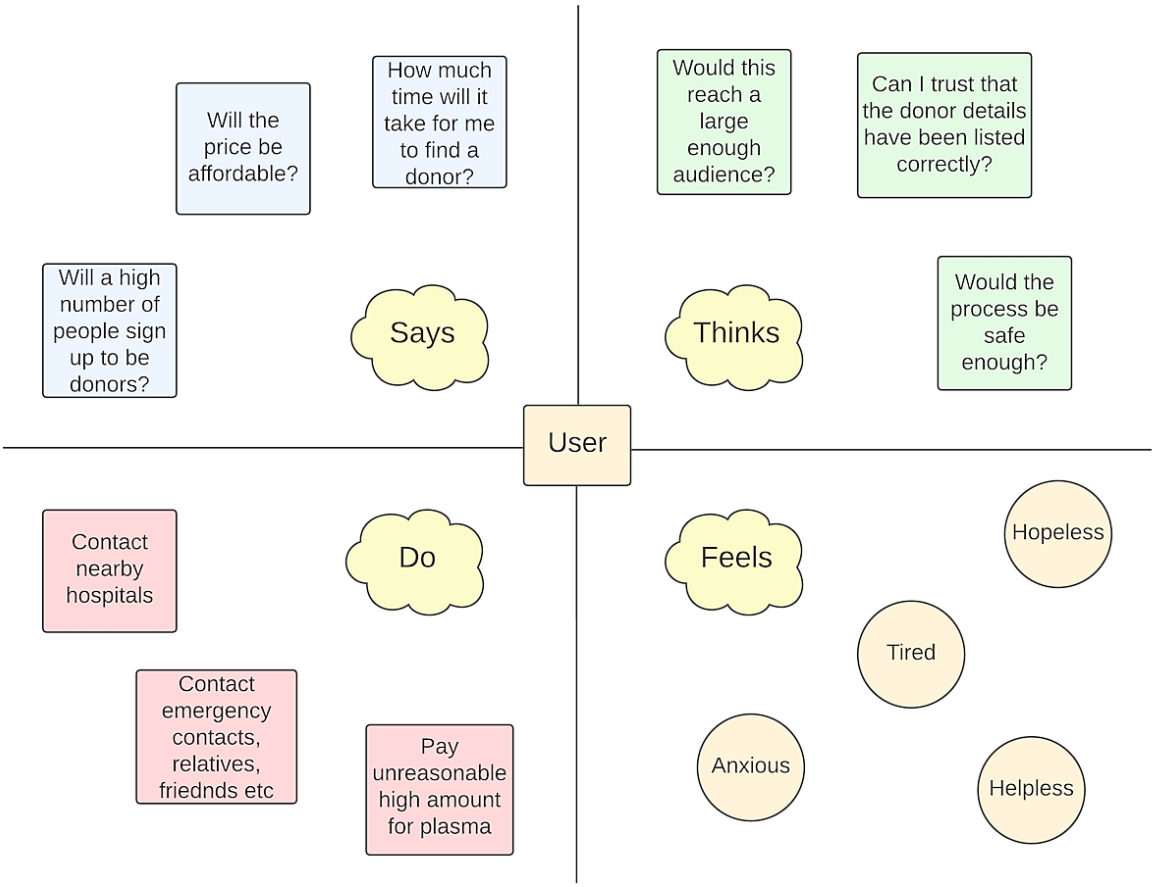


Fig 3.1 : Empathy Map

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to 1) create a shared understanding of user needs, and 2) aid in decision making. Fig 3.1 depicts the empathy map of the proposed work.

3.2 Ideation and Brainstorming

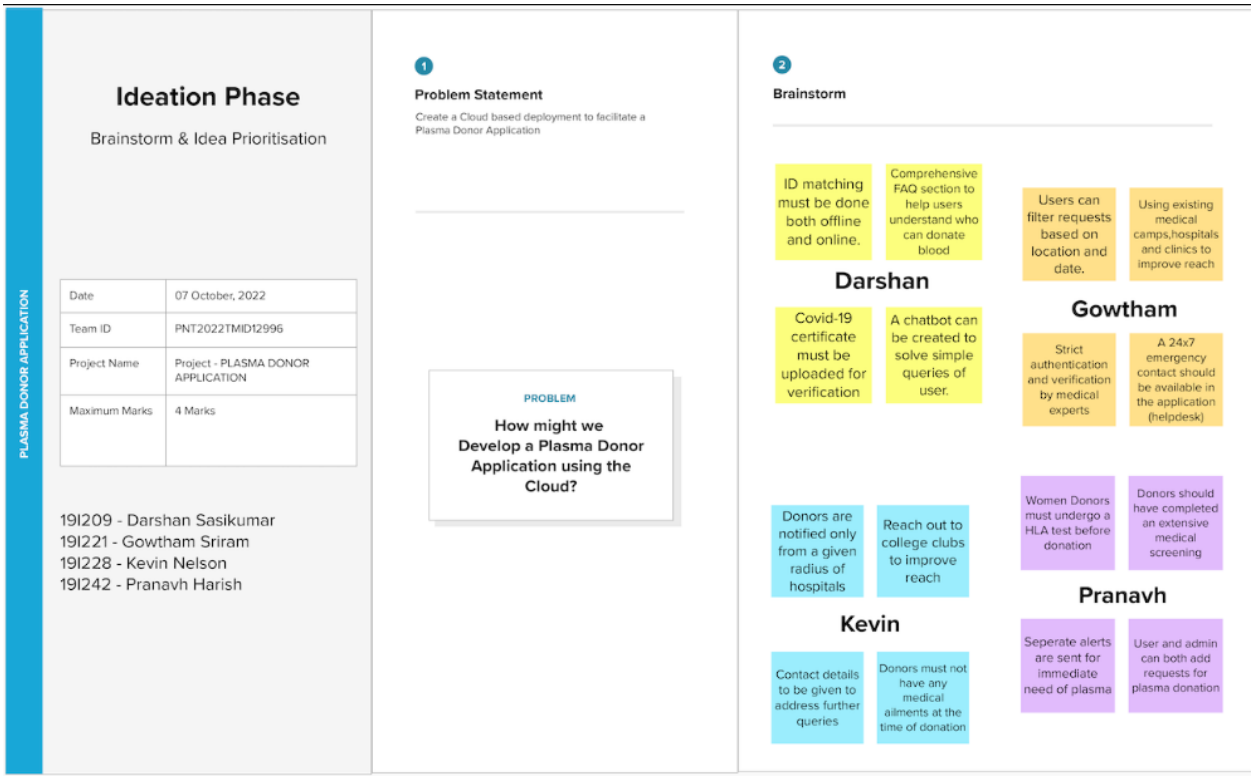


Fig 3.2: Ideation and Brainstorming Phase

Ideation is a creative process where designers generate ideas in sessions. The participants gather with open minds to produce as many ideas as they can to address a problem statement in a facilitated, judgment-free environment. Fig 3.2 depicts the ideation and brainstorm of the proposed work.

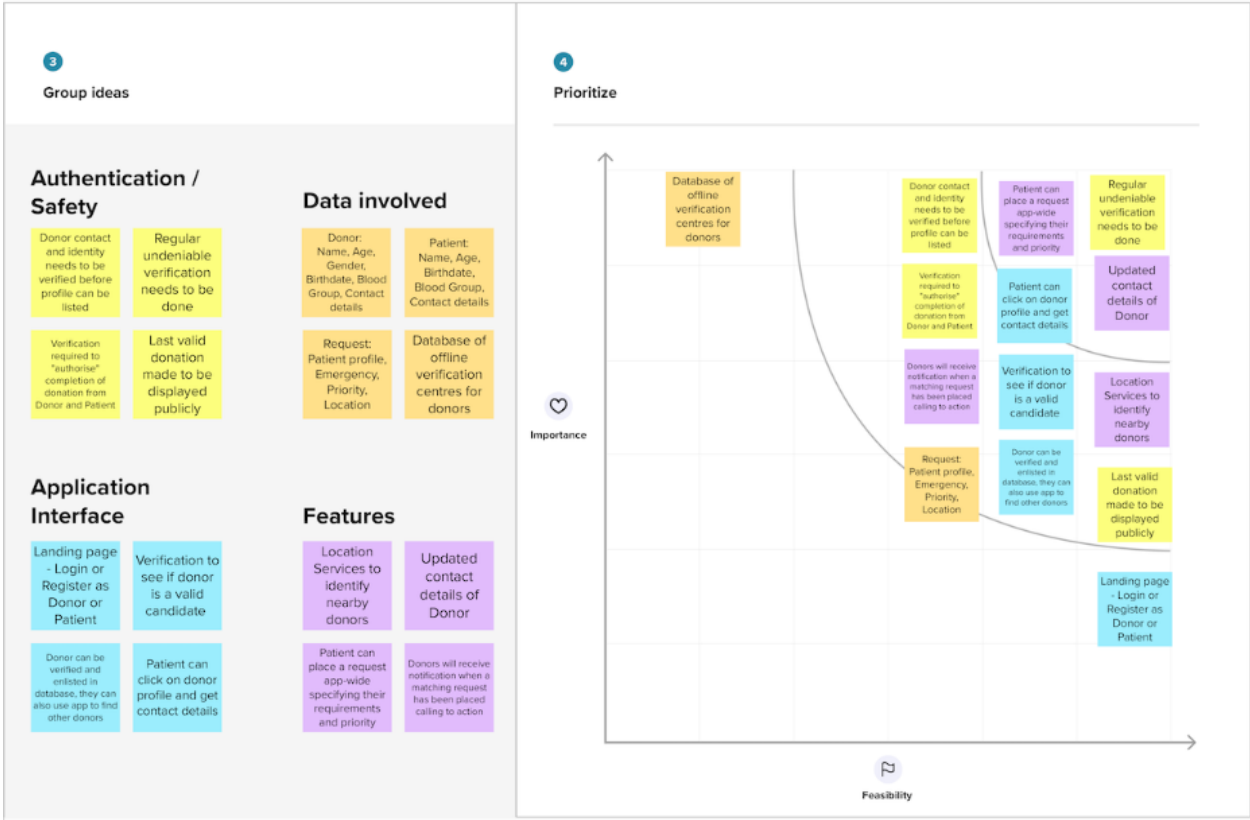


Fig 3.3: Group Ideas and Prioritisation

The ideas gathered in ideation phase are then gathered, and given a priority ranking, so as to select the high priority ideas during the implementation phase. Fig 3.3 depicts the group ideas and prioritisation of ideas.

3.3 Proposed Solution:

Our unified platform is aimed at all donors and requesters so they can communicate easily. Donors can be registered, verified and listed on the platform database. Patients or people in urgent need of plasma can use their location and other attributes to find a compatible plasma donor match right in time. Although this benefits the donors by offering them a universal location to list themselves, it poses a more significant impact to the patients who can find the resources they need in a way smaller timeframe. Patients and requesters don't have to be mentally pressured and frustrated about being able to find a compatible donor before it is too late.

Chapter 4

Requirements 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 Mobile Number Registration through Gmail
FR-2	**User Registration**	Confirmation via Email Confirmation via OTP
FR-3	**User Validation**	Donor – Check health conditions Patient – Check credibility of user
FR-4	**Chatbot**	Help the user understand the process and navigate the website
FR-5	**Search**	Patient enters details to search for compatible donors
FR-6	**Request**	Patient can send a donor request to obtain their plasma
FR-7	**Email notification**	Donor will get notified through email when a compatible patient places request
FR-8	**Donation completion**	Patient and Donor to confirm completion of donation

Table 4.1.1 Functional requirements

4.2 Non Functional Requirements

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

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user experience must be simple and the user must be able to perform all actions available on the platform
NFR-2	**Security**	The database layer and the logic layer are hosted using IBM services, hence the security of all data involved in the process is kept secure
NFR-3	**Reliability**	All processes involved in the platform must be uniform whenever accessed. It must also function without any bugs and errors
NFR-4	**Performance**	Immediate feedback from the platform is provided to user, so they are not discouraged to use the application
NFR-5	**Availability**	The application must be usable at all time, the database and servers need to be available and reachable anytime from anywhere
NFR-6	**Scalability**	The platform must adapt so that it can support a high volume of concurrent users. Meanwhile, the platform must also be loosely couples to ensure it can scale vertically too, by adding more functionality

Table 4.2.1 Non-functional requirements

Chapter 5

Project Design

5.1 Data Flow Diagrams:

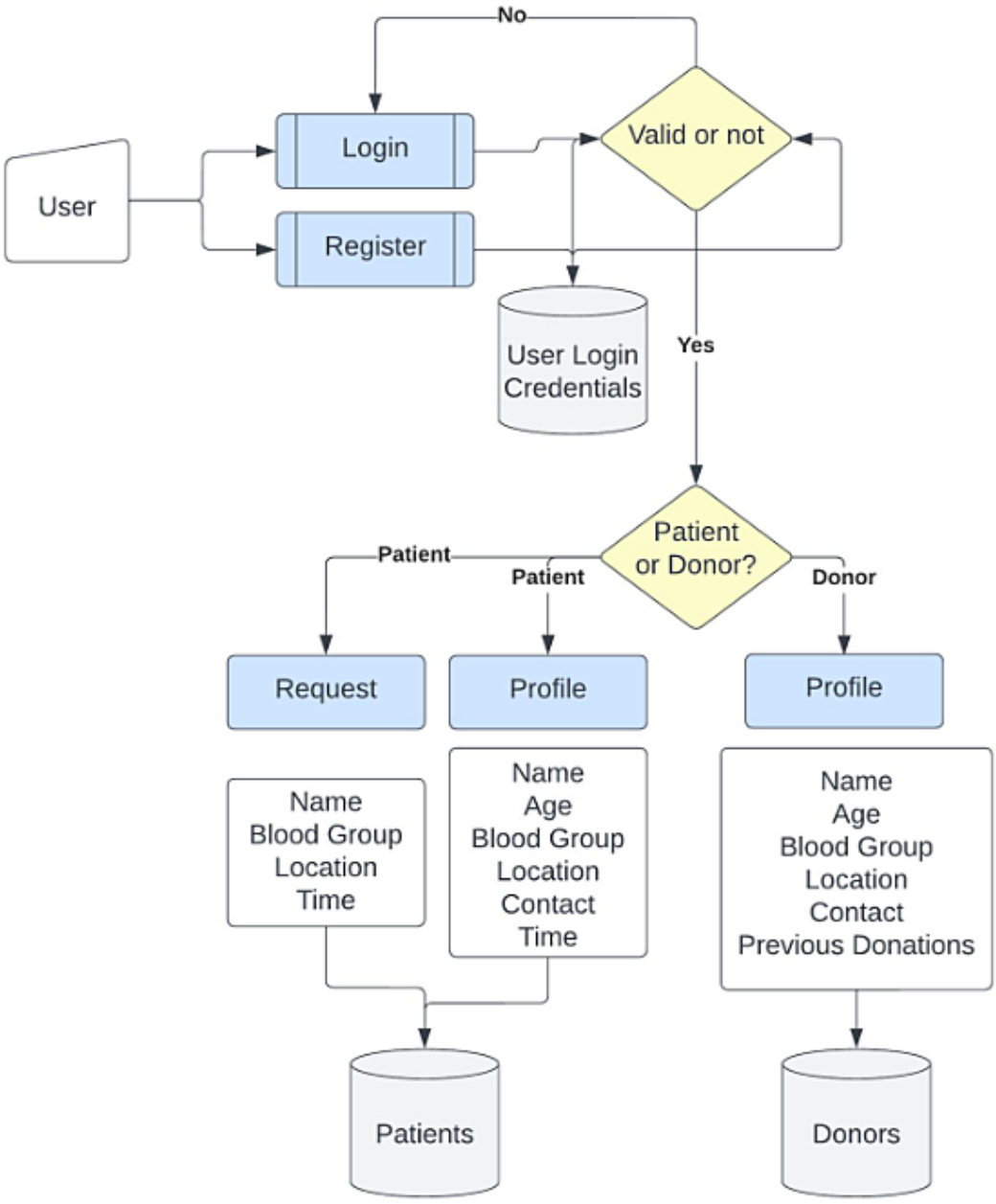
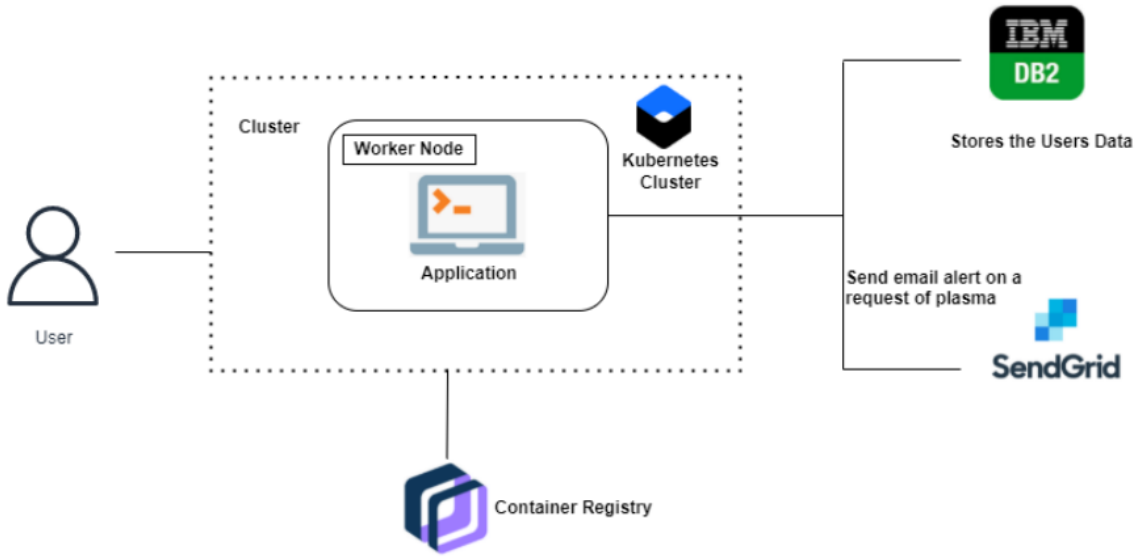


Fig: 5.1 Data Flow Diagram of Our Proposed System

5.2 Solution and Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the Table1 and Table 2:



S.No	Component	Description	Technology
1.	User Interface	User is presented with a website to interact with the platform, register, login, and place requests for plasma	HTML, CSS, JavaScript
2.	Application Logic-1	Login, Register as a patient or donor to the application	IBM DB2
3.	Application Logic-2	Search for donors by entering details of plasma required	IBM DB2
4.	Application Logic-3	Chatbot to help educate the users and navigate through the platform	IBM Watson Assistant

5.	Database	Used for data appending and retrieval from backend server by users	MySQL
6.	Cloud Database	Database Service on Cloud to store details about patients, donors, and history of requests and donations made	IBM DB2
7.	External API-1	SMTP mail is used to notify the donors if users have made requests for their specific plasma or compatible plasma	SMTP
8.	Infrastructure (Server / Cloud)	Deployed on container registry after containerizing image	Docker, Kubernetes, IBM Container Registry

Table 5.2.1 Components & Technologies

S.No	Component	Description	Technology
1.	Open-Source Frameworks	Flask framework of python is used to build the web application. Kubernetes is used to containerize the application, deploy and maintain it	Flask, Kubernetes
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Transport layer security, IBM Object Storage
3.	Scalable Architecture	A 3-tier architecture is used, with different layers for the database, logic and presentation	Docker
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Kubernetes
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Docker

Table 5.2.1: Application Characteristics

5.3 User Stories

User type	Functional Requirement	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Patient, Donor	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	5	High	Gowtham Pranavh
Patient, Donor		USN-2	As a user, I will receive a confirmation email which will successfully complete my registration procedure on the application	5	High	Kevin Darshan
Patient, Donor	Login	USN-3	As a user, I should be able to login with my existing credentials and my profile should be retained	5	High	Gowtham Darshan
Patient	Request creation	USN-4	As a patient, I can create requests for plasma donation for donors to see and respond to	10	High	Gowtham Darshan Kevin Pranavh
Donor	Donor creation	USN-5	As a donor, I should be able to enter my details and other health related information to show interest in donating plasma	5	High	Kevin Gowtham
Administrator	Request verification	USN-6	As a admin, I must verify the validity of each requests to ensure resources are not wasted, by contacting patient or hospital	5	High	Darshan Pranavh
Donor	Donor acceptance	USN-7	As a donor, I can accept requests that match my details and contact the patient or hospital to initiate donation	10	High	Gowtham Darshan Kevin Pranavh
Patient, Donor	Contact details sharing	USN-8	As a user, the donor contact will be shared to patient, and patient contact to donor for further communication	10	High	Gowtham Darshan Kevin Pranavh
Patient, Donor	Chatbot	USN-9	As a user of the application, I can avail guidance on how to use the app and the functionalities from the chatbot	4	Medium	Gowtham Kevin
Patient	Contact us	USN-10	As a user, I should have the option to provide feedback and contact the support team to convey the issues experienced while using the app	4	Medium	Darshan Pranavh

Table 5.3.1 User stories

Chapter 6

Project Planning and Scheduling

6.1 Sprint Planning & Estimation

Milestone	Description	Date
Literature Survey	Literature survey is important to the project because it helps in understanding the progress of the concept we are working on. It also identifies the best works of some of the experts that went in this field, and what their shortcomings were. This helps us fine tune our product further.	17 Sep 2022
Empathy Map	Empathy map is a visual representation of how the target audience, users, would perceive, understand and interact with our product.	17 Sep 2022
Brainstorming	This can be utilised to introduce new topics and ideas and see how feasible or ground breaking they are.	17 Sep 2022
Proposed Solution	The solution is made with all the requirements and the needs of the audience in mind.	19 Sep 2022
Problem-Solution Fit	We go through the solution again to verify if our product actually aligns with the customers mindset and needs. We can identify how well a certain aspect would work as a part of the product.	19 Sep 2022
Solution Architecture	The solution involves all the processes involved, similar to a flowchart, it shows in relation with other tasks how to whole application works	19 Sep 2022
Customer Journey	This document takes us through the process of discovering, entering, using and leaving the product from the perspective of a customer. It helps us see the end user experience clearly.	03 Oct 2022
Functional Requirement	These requirements describe how the methods of the application and the user experience intersect.	03 Oct 2022

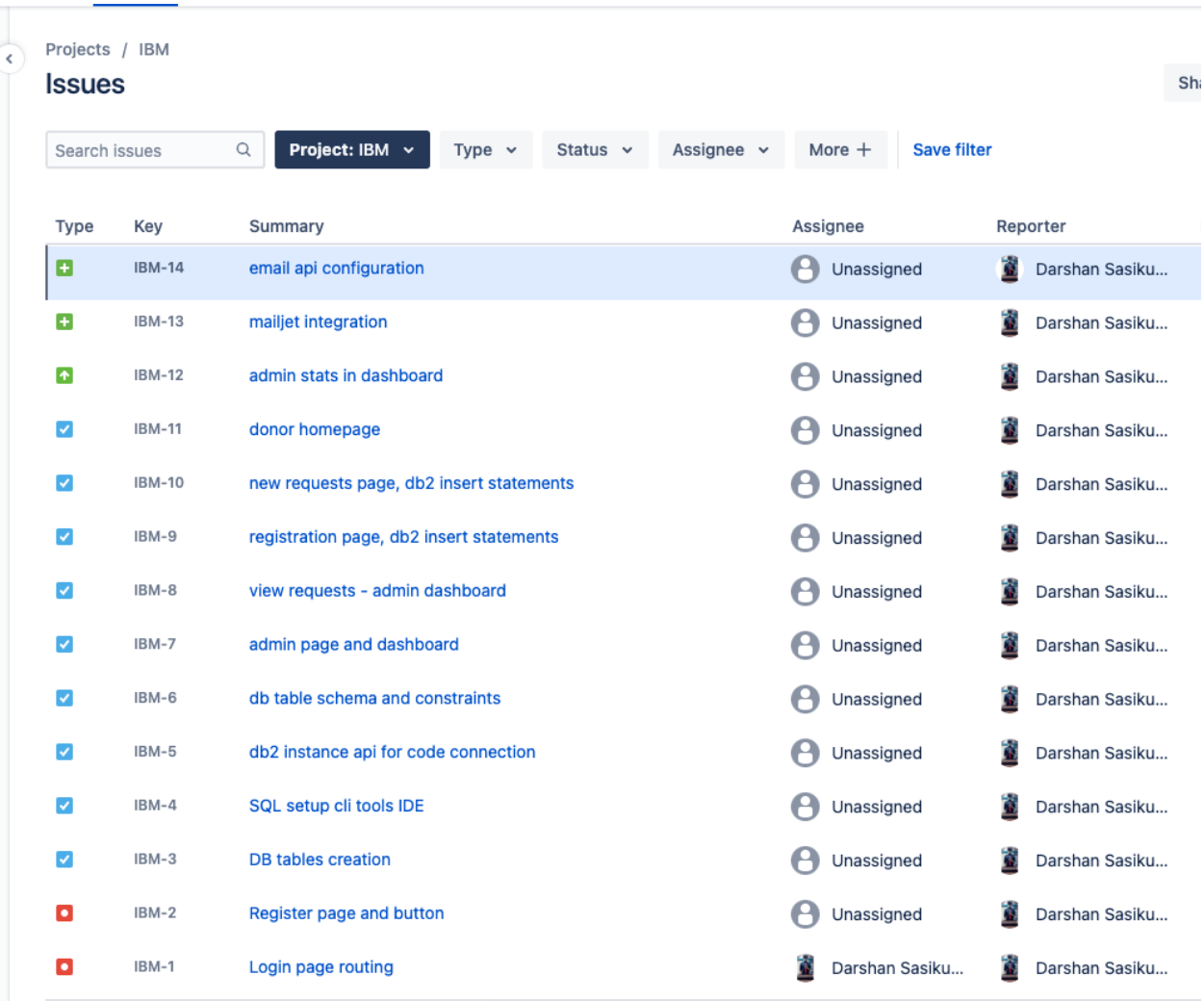
Data Flow Diagrams	The diagrams are pictorial visualisations of the transfer, storage and retrieval of data involved in the application flow	03 Oct 2022
Technology Architecture	The development of the architecture involves deciding the tools and technologies that the team has chosen to use for designing the product.	03 Oct 2022
Sprint Delivery Plan	The delivery plan helps efficiently plan and set targets to work towards. Tasks can be split into separate sprints and user stories can be given priorities to increase productivity	19 Oct 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement	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	5	High	Gowtham Pranavh
		USN-2	As a user, I will receive a confirmation email which will successfully complete my registration procedure on the application	5	High	Kevin Darshan
	Login	USN-3	As a user, I should be able to login with my existing credentials and my profile should be retained	5	High	Gowtham Darshan
Sprint 2	Request creation	USN-4	As a patient, I can create requests for plasma donation for donors to see and respond to	10	High	Gowtham Darshan Kevin Pranavh
	Donor creation	USN-5	As a donor, I should be able to enter my details and other health related information to show interest in donating plasma	5	High	Kevin Gowtham
	Request verification	USN-6	As a admin, I must verify the validity of each requests to ensure resources are not wasted, by contacting patient or hospital	5	High	Darshan Pranavh
Sprint 3	Donor acceptance	USN-7	As a donor, I can accept requests that match my details and contact the patient or hospital to initiate donation	10	High	Gowtham Darshan Kevin Pranavh
	Contact	USN-8	As a user, the donor contact will	10	High	Gowtham

	details sharing		be shared to patient, and patient contact to donor for further communication			Darshan Kevin Pranavh
Sprint 4	Chatbot	USN-9	As a user of the application, I can avail guidance on how to use the app and the functionalities from the chatbot	4	Medium	Gowtham Kevin
	Contact us	USN-10	As a user, I should have the option to provide feedback and contact the support team to convey the issues experienced while using the app	4	Medium	Darshan Pranavh

6.3 Reports from JIRA



Bug tracking template of Jira Atlassian was used to keep track of new features, completion of the tasks, improvements, features, bugs and assign them to team mambers appropriately.

Chapter 7

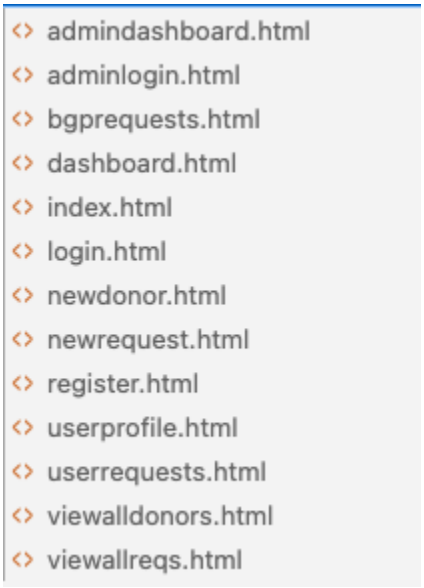
Coding and Solutioning

The application is based on the Flask framework of python used to develop web applications. The design of the pages are done using HTML, and CSS for styling them.

We used an IBM DB2 service to host the database to store all the information related to the application.

Container registery service was used to store and publish the containerized application, and Kubernetes to host this application on the cloud.

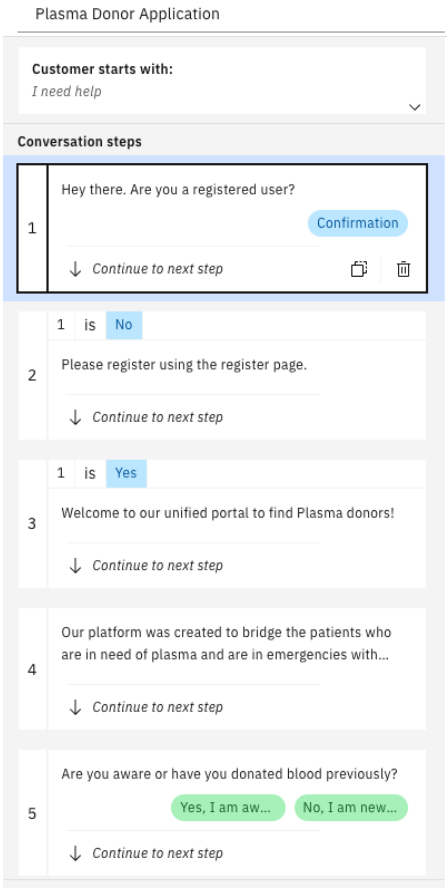
The mailing service which we are using is called the SMTP mailing package, it is used to share communication to users of the application, including donors and recipients.

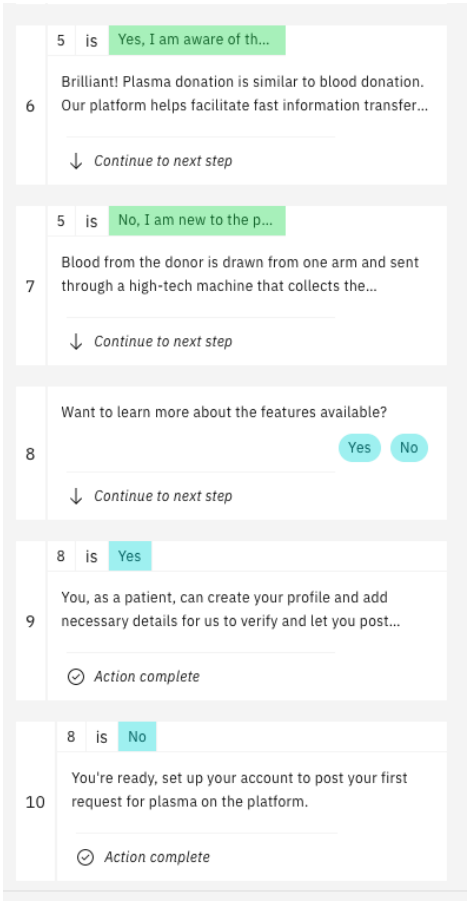


- Admin dashboard - view number of donors, patients and requests live
- Admin login - Admin access to the application
- BGPrequests - Requests made by patients
- Dashboard - User can register as donor, request plasma, view past requests, view admin requests

- Index - landing page
- Login - main login page, admin can click on admin login and enter credentials
- New Donor - Form details for donor registration
- New Request - Form details for requesting plasma from patient side
- Register - creating a new account
- User Profile - shows details of the user account entered in forms
- User requests - see past requests placed by patient account
- View all donors - admin can see all donors registered on platform
- View all requests - admin can view all requests placed for plasma

The landing page (index.html) also houses our own chatbot created to assist users by simulating a conversation with a virtual support assistant.





Database Schema

Donor

```
create table donor(
username varchar(20),
name varchar(20),
age integer,
gender varchar(1),
phone varchar(10),
addr varchar(40),
city varchar(20),
state varchar(20),
blood varchar(5),
don date
)
```

Users

```
create table users (
username varchar(20),
email varchar(50),
password varchar(20)
)
```

Requests

```
create table requests(
username varchar(20),
pname varchar(20),
phone varchar(10),
paddress varchar (40),
city varchar(20),
state varchar(20),
blood varchar(5)
)
```

Approved requests

```
create table approved(
pname varchar(20),
phone varchar(10),
state varchar(20),
blood varchar(5)
)
```

Admins

```
create table admins(
username varchar(20),
password varchar(20)
)
```

Chapter 8

Testing

Testing was managed on a platform called Testlodge. This website helps us create test suites and individual test cases for accounting purposes. We can describe individual test cases, their pass and fail criteria. These test cases can be then assigned to individual team mebers for them to carry out the test and report the result on the platform.

We performed unit testing, integration testing and acceptance testing on our final application.

Unit testing was done by creating testcases for individual modules to work in separation from the rest of the application. Each of the html pages, their scripting and their input and output was tested.

Integration testing was performed to check if a complete logical process can be done without any errors in the workflow. As some of our functions and features work together, they need to be assessed if the testcases pass.

8.1 Test cases:

8.2 Acceptance Testing:

Acceptance testing is testing the overall service flow and checking if each logical function works with all ranges of input from the perspective of an end user. This is similar to blackbox testing, as the user doesn't know the internal functioning of the application. They only provide their input and we need to validate if the necessary output has been acquired as a result.

Load testing was performed on the hosted application using Jmeter. Jmeter is an application used for testing web applications and websites performance metrics. HTML GET requests are simulated with thread groups on Jmeter to test throughput, deviation, load etc.

Chapter 9
Results

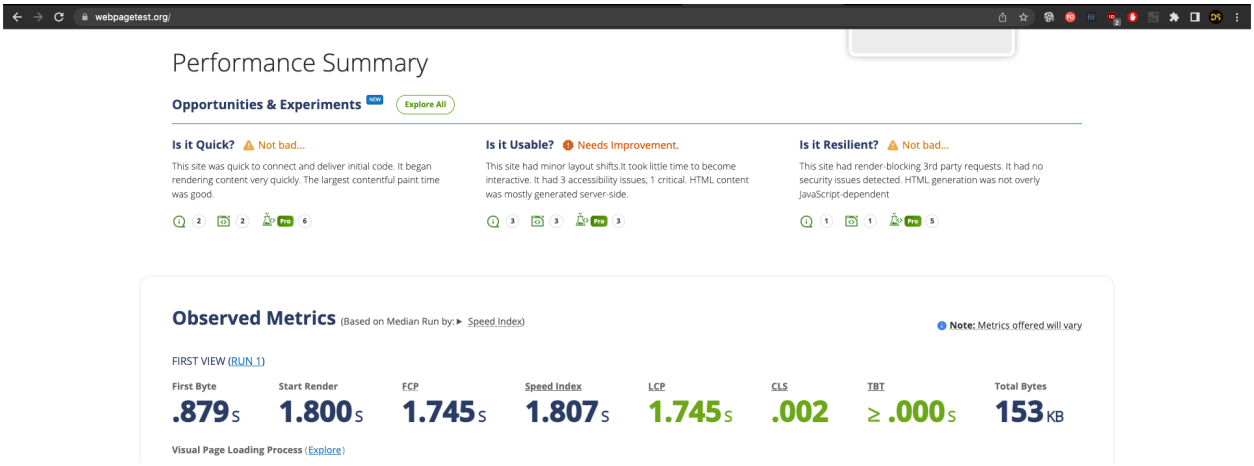
9.1 Software Test metrics

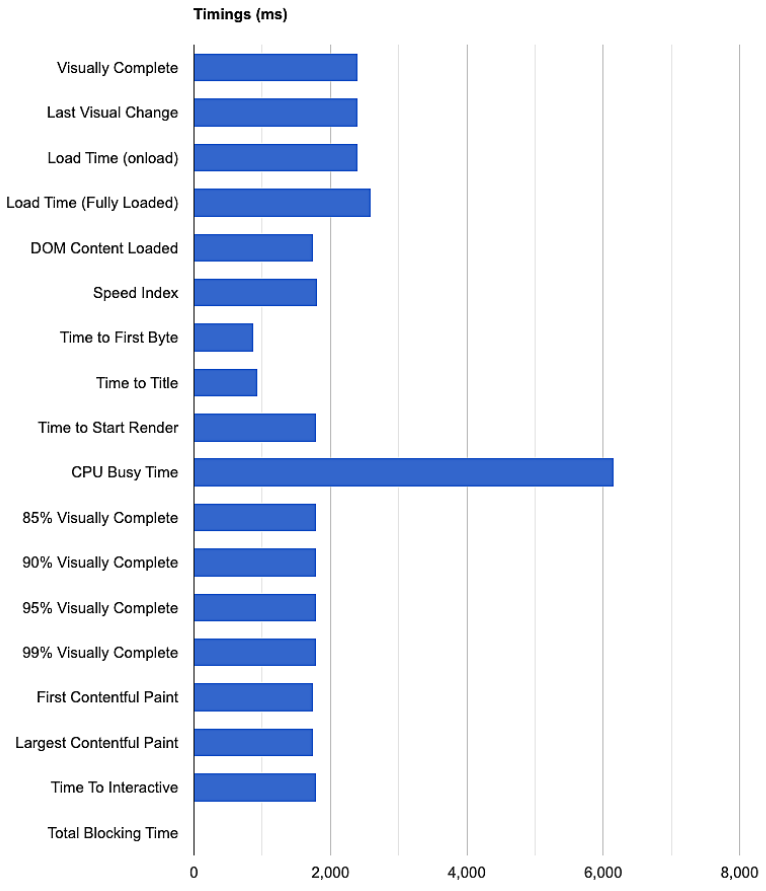
1. Test design efficiency = Number of tests designed /Total time
    - 25 Tests designed / 10 Hours = **2.5 Tests designed per hour**
2. Passed Test Cases Percentage = (Number of Passed Tests/Total number of tests executed) X 100
    - 25 passed Test cases / 25 Test cases x 100 = **100%**
3. Failed Test Cases Percentage = (Number of Failed Tests/Total number of tests executed) X 100
    - 0 failed Test cases / 25 Test cases x 100 = **0%**
4. Schedule slippage = (Actual end date – Estimated end date) / (Planned End Date – Planned Start Date) X 100
    - (19/11/2022 - 17/11/2022) / (17/11/2022 - 10/09/2022) x 100 = 3 days / 70 days x 100 = **4.28%**

9.2 Performance metrics

Testing done using online tool - https://www.webpagetest.org/

Metric	Measure
First Byte	0.879s
Start Render	1.800s
First Contentful Paint (FCP)	1.745s
Speed Index	1.807s
Largest Contentful Paint (LCP)	1.745s
Cumulative Layout Shift (CLS)	0.002
Total Blocking Time (TBT)	> 0.000s
Total Bytes	153KB





Our project uses the IBM Lite plan for all the cloud services like DB2, Watson Assistant, Container Registry, Kuberenetes which provides minimal access and performance for free of cost.

There are performance constraints for the deployed application. All requests by the user must be processed and rendered on screen before 5 seconds of initiation. If this is not possible due to errors from any side, a buffering screen must be displayed.

Database needs to be online at all times to facilitate information transfer whenever a user logs on and accesses the platform.

Chapter 10

Advantages & Disadvantages

The vision of this project is to provide a unified platform where all the requests and information can be shared for higher visibility and faster correspondence. By using an application to serve both donors and patients who are in need of plasma, we are bridging the gap in between them. Donors can register themselves so they may be contacted when a certain request is placed. Patients can also place a request on the platform so their needs can be fulfilled as soon as possible.

Disadvantages of this platform include a transparency about sensitive information provided by both patients and donors. Administrators who are responsible for managing requests would be able to see information regarding the users health. Verification also needs to be implemented to make sure that the credibility of both patients and donors are upheld, to increase the reliability of the platform for public usage.

Chapter 11
Conclusion

In conclusion, we have created an application and provided an interface with which donors can contact hospitals and hospitals can in turn find donors to meet the plasma requirements. This will allow hospitals to meet the demands of plasma for the patients undergoing treatment. Plasma from Covid recovered patients can be used to treat covid patients. This allows an easy way for donors to interact with hospitals and to donate plasma when needed.

In existing solutions, requests are accepted from patients and are batch processed. They are not tended to in real time. Existing solutions also feature only work with accounted plasma that is already present in a blood bank, but our solution is available to the public to share real time information and obtain plasma as quickly as they need it.

Donors who need to donate plasma can donate by uploading their details and blood bank can view donors and can raise requests to donors and the hospital can register/login and can search for plasma, they can raise requests to blood bank and can get the plasma.

People need not have feelings of anxiety and fear that they might not receive the help they seek. Using our platform they can gather information about the donors available near them and compatible so the donation can be done.

Such an application will be beneficial to the public who currently don't have a common and dedicated platform to share such emergencies and contact details.

Chapter 12

Future Scope

There is scope for more features to be integrated in our application for a better user experience and more efficient process.

A messenger service can be included on the platform for the donors and patients to enable direct communication without the need for a medium.

Verification using any photo ID can be done as to automate the process by integrating government service to increase the credibility and reliability of the application.

A ticketing system can be used to collect, track and manage user queries and problems they face during the usage of the application.

Although the app is already cross platform supported due to it being a website, a mobile application can be designed and published for users to have instant access to our application on their smartphones.

Chapter 13
Appendix

Github link: https://github.com/IBM-EPBL/IBM-Project-16927-1659624974

Demo link: https://www.youtube.com/embed/3qSXd0qArak