KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine

Accredited by NAAC with 'A' grade
Autonomous | Affiliated to Anna University

(An ISO 9001:2015 and ISO 14001:2015 Certified Institution)

18PF15 PROFFESIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

TEAM ID : PNT2022TMID12582

TEAM MEMBERS	ROLL NO
1. PALANIKUMAR R	717819P324
2. BOOPATHI D	717819P305
3. SABARI VIGNESH M	717819P329
4. SATHEESH AJO A	717819P335

CONTENTS

CHAPTER-1

INDRODUCTION

With rapid increase in the usage of social networks sites across the world, there is also a steady increase in plasma donation requests as being noticed in the number of posts on these sites such as Face book and twitter seeking plasma donors.

Finding plasma donor is a challenging issue in almost every country. There are some plasma donor finder applications in the market such as Blood app by Red Cross and Blood Donor Finder application by Neologix.

1.1 PROJECT OVERVIEW

Several software technologies including languages and framework are used to develop our plasma-donor web application known as "**PLASMA DONOR APPLICATION**".

These technologies includes HTML, CSS along with PYTHON and IBM CLOUD for database are used. The python is computer programming language often used to create websites and software, automate task and conduct the data analysis.

Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problem.

1.2 PURPOSE

The main goal of our project is to design a user-friendly web application that is like a scientific vehicle from which we can help 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.

Your application helps patients who need plasma-derived biotherapies to improve or save their lives. Those in need are suffering from life-threatening conditions such as hemophilia, immune deficiencies, and other blood disorders.
Plasma is the essential ingredient in many medications and treatments.

CHAPTER -2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

There are a quite good number of software packages that exist for PLASMA DONOR APPLICATION system. But when I visited most plasma donor center system portal. I found that existing system is limited only to those particular plasma center.

Problem Found In Existing System

- At the present there is no software to keep any records in plasma center.
- It becomes difficult to provide any record immediately at times of emergency.
- Required more human efforts in maintaining the branch related information .
- Manually to keep the accounts is also tedious & risky job & to maintain those accounts in ledgers for a long period is also very difficult.
- Difficult to manage and maintain the files.
- Chance of damage of files, if the data is stored in the files for duration of time.
- Time consuming is retrieving, storing and updating the data.
- It is difficult to keep track the record about the donor & receiver he has donated or received the plasma at the last time.

2.2 REFERENCES

CASE STUDY - I

TITLE: Instant Plasma donar Recipient connector web application

AUTHOR: Kalpana Devi Guntoju, Tejaswini Jalli, Sreeja Uppala, Sanjay Mallisetti

YEAR: 2022

ABSTRACT:

The world is suffering from the COVID 19 crisis and no vaccine has been found yet, but there is another scientific way in which we can help reduce mortality or help people affected by COVID19 by donating plasma from recovered patients. In the absence of an approved antiviral treatment plan for a fatal COVID19 infection, plasma therapy is an experimental approach to treat COVID19-positive patients and help them faster recovery. Therapy is considered competent. In the recommendation system, the donor who wants to donate plasma can donate by uploading their COVID19 certificate and the blood bank can see the donors who have uploaded the certificate and they can make a request to the donor and the hospital can register/login and search for the necessary things. plasma from a blood bank and they can request a blood bank and obtain plasma from the blood bank.

CASE STUDY - II

TITLE: Determinants of plasma donation: A review of the literature

AUTHOR: A.Beurel , F. Terrade, J.-P.Lebaudy, B. Danic

YEAR: 2017

ABSTRACT:

The major contribution of Human Sciences in the understanding of the whole blood donation behavior has been through the study of individuals' motivations and deterrents to donate. However, if whole blood donation has been very widely studied in the last sixty years, we still know very little about plasma donation in voluntary non-remunerated environments. Yet, the need for plasma-derived products has been strongly increasing for some years, and blood collection agencies have to adapt if they want to meet this demand. This article aims to review the main motivations and deterrents to whole blood donation, and to compare them with those that we already know concerning plasma donation. Current evidence shows similarities between both behaviors, but also differences that indicate a need for further research regarding plasma donation.

CASE STUDY – III

TITLE: Developing a plasma donor application using Function-as-a-service in AWS

AUTHOR: Aishwarya R Gowri

YEAR: 2020

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 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. 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.

2.3 PROBLEM SOLUTION DEFINITION

PROBLEM STATEMENT – I

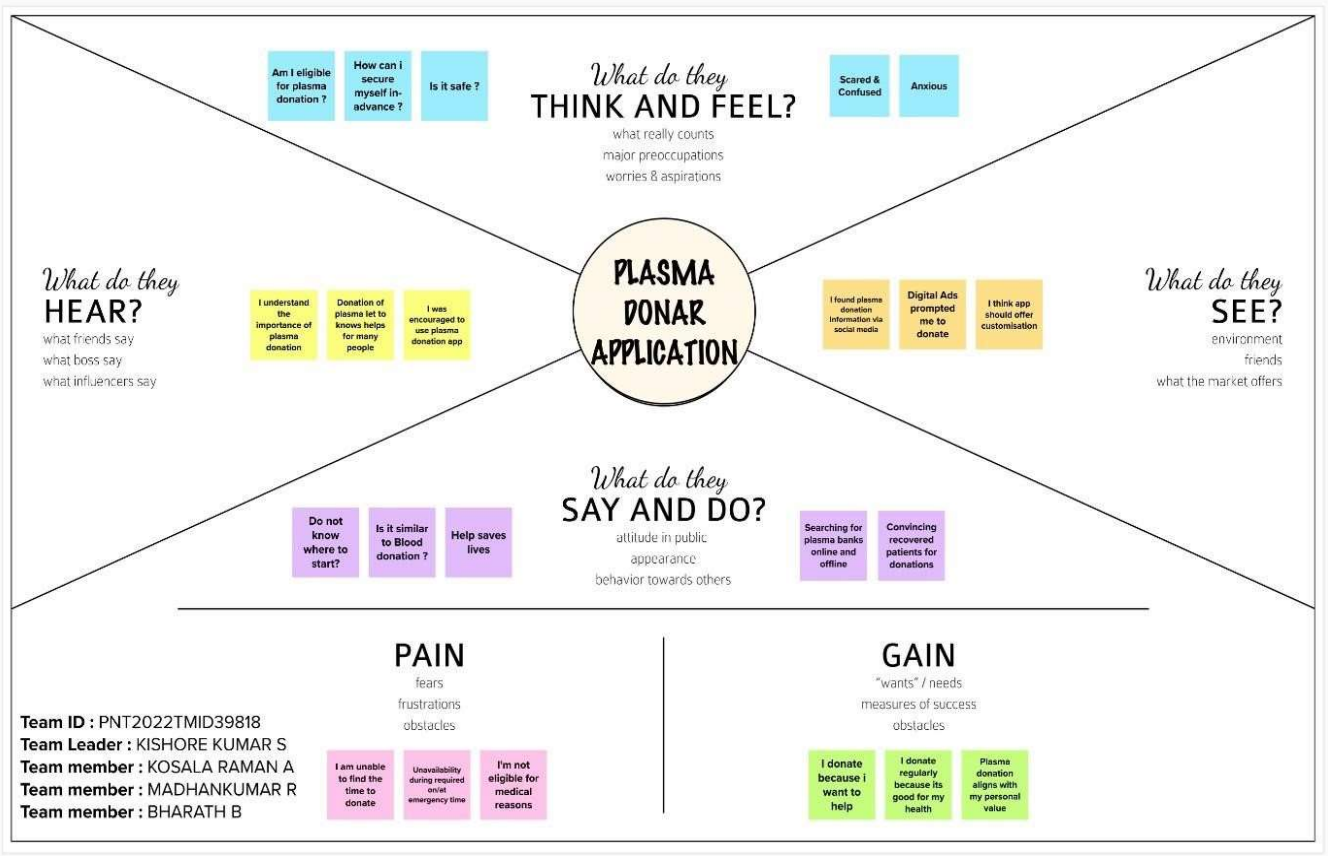
PROBLEM STATEMENT – II

PROBLEM STATEMENT – III

PROBLEM STATEMENT – IV

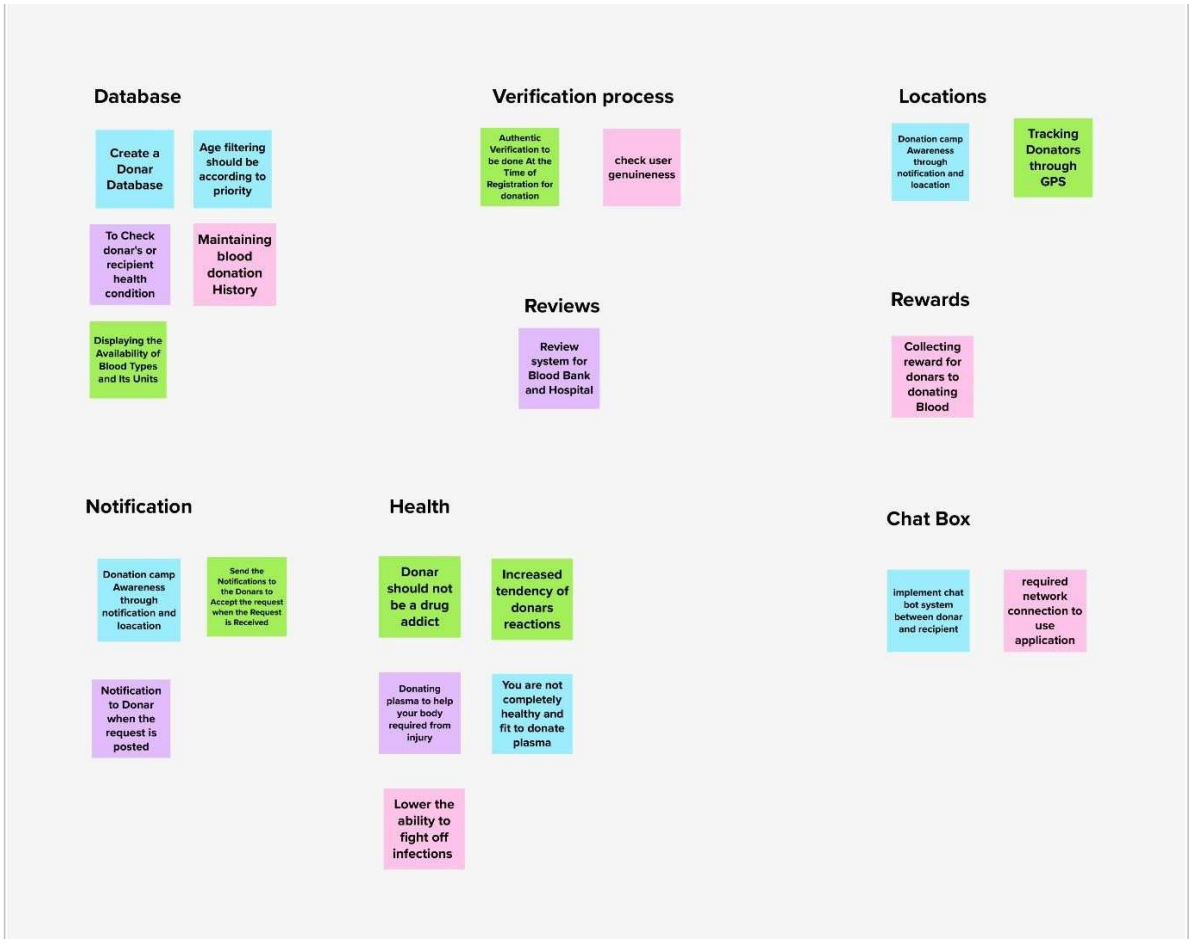
CHAPTER 3 IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING

Step-1: Brainstorm, Idea Listing and Grouping


Database
Create a Donar Database
Age filtering should be according to priority
To Check donar's or recipient health condition
Maintaining blood donation History
Displaying the Availability of Blood Types and Its Units
Verification process
Authentic Verification to be done At the Time of Registration for donation
check user genuineness
Reviews
Review system for Blood Bank and Hospital
Locations
Donation camp Awareness through notification and loacation
Tracking Donators through GPS
Rewards
Collecting reward for donars to donating Blood
Notification
Donation camp Awareness through notification and loacation
Send the Notifications to the Donars to Accept the request when the Request is Received
Notification to Donar when the request is posted
Health
Donar should not be a drug addict
Increased tendency of donars reactions
Donating plasma to help your body required from injury
You are not completely healthy and fit to donate plasma
Lower the ability to fight off infections
Chat Box
implement chat bot system between donar and recipient
required network connection to use application

Step-2: Idea Prioritization

3.3 PROPOSED SOLUTION

Proposed solution – I

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	I am Mr. Madhavan rao when I'm trying to use the plasma donar application but I'm not convenient to use the application because I'm expecting more specification, if more specifications are added, which makes me feel enthusiastic.
2.	Idea / Solution description	The suggestion which are said by the user will be noted and the apt suggestions will be added
3.	Novelty / Uniqueness	Everyone will have different ideas and different queries but the most important suggestions will be added upon the application.
4.	Social Impact / Customer Satisfaction	Almost Customer will be satisfied on the problem, if once again the problem occurs, it can be easily recovered.
5.	Business Model (Revenue Model)	On the revenue bases, this donar application will be profit for Hospital, NGO's and private organizations.
6.	Scalability of the Solution	The problem of the donar was solved and also as per the user flexibility the requirements can be modified.

Proposed solution – II

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	I am Stephen when I'm trying to donate plasma, But I can't donate Because 2 weeks before only I had donated the blood for plasma, but continuously I'm receiving the notification/mail for requirement which makes me feel Hatred.
2.	Idea / Solution description	Stephen needs to update his plasma donation details in the Application, if Still the issue occurs use "Contact Us" option in the application
3.	Novelty / Uniqueness	This problem rarely occurs to the users/donar and not a common problem. It will be rectified from the "technical team".
4.	Social Impact / Customer Satisfaction	The Customer will be more satisfied with the solution and if once again the problem occurs, it can be easily recovered.
5.	Business Model (Revenue Model)	On the revenue bases, this donar application will be profit for Hospital, NGO's and private organizations.
6.	Scalability of the Solution	The thought of the user/donar about the application will changed and also as per the user flexibility the requirements can be modified.

Proposed solution –

III

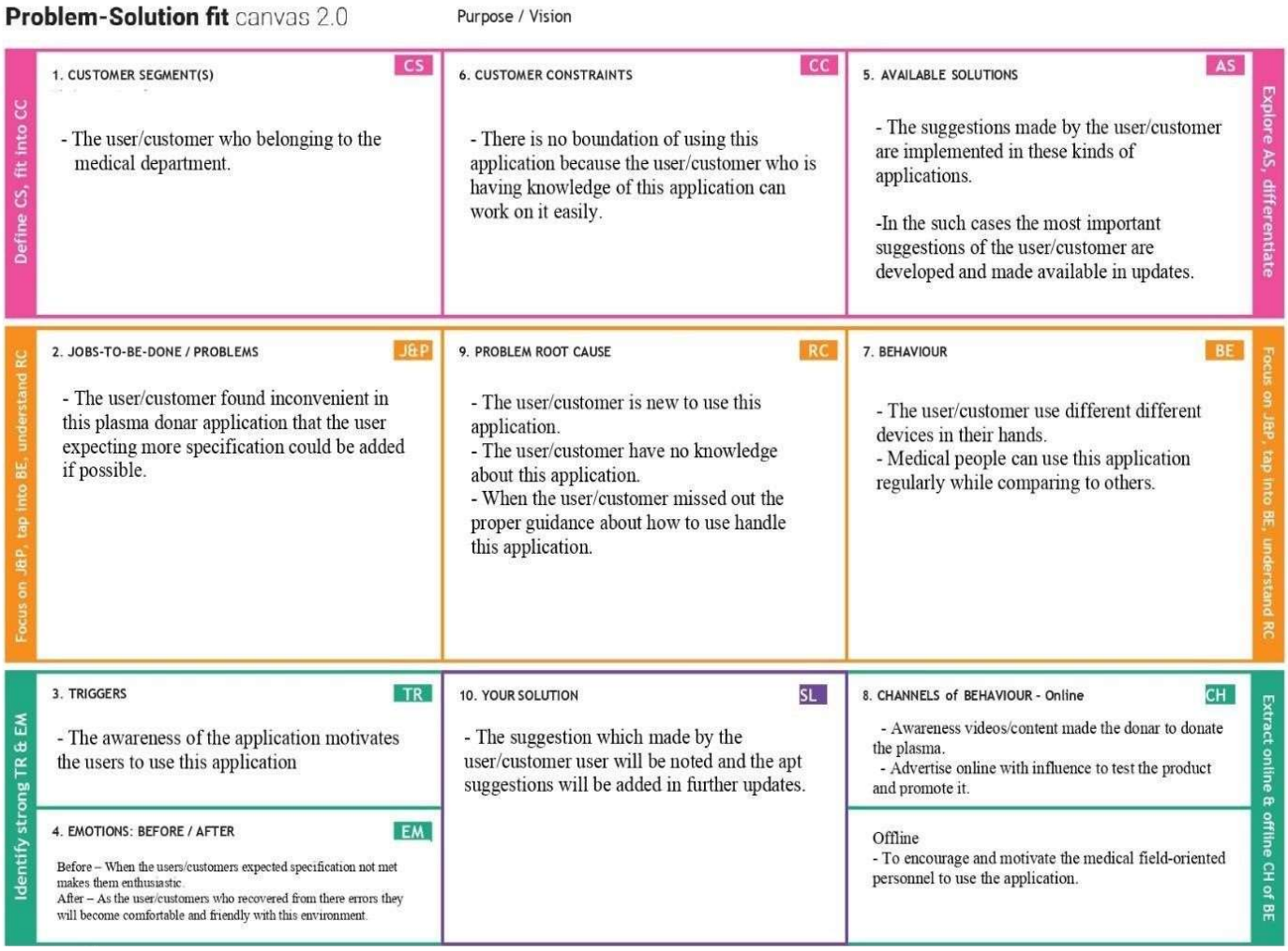
S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	I am a student I'm trying to use Plasma Donar Application Because I want to use the application, But I don't how to use the donar application & I never used before which makes me feel Anxiety.
2.	Idea / Solution description	The user should have basic knowledge about the application, read the user manual or else use the "Chat Bot" for the guidance to use the application efficiently.
3.	Novelty / Uniqueness	It is common problem face by the new users who are trying to use the application. If the user once learns how to use, then there will be no issue.
4.	Social Impact / Customer Satisfaction	The solution will be satisfied to the user. If any problem occurs once again it will be rectified.
5.	Business Model (Revenue Model)	On the revenue bases, this donar application will be profit for Hospital, NGO's and private organizations.
6.	Scalability of the Solution	The problem of the donar was solved and also as per the user flexibility the requirements can be modified.

IV

Proposed solution –

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	I am a student I'm trying to find Plasma Because in the need of emergency, But I don't whether the required unit of plasma is available or not which makes me feel disappointment.
2.	Idea / Solution description	The user should know the required unit of plasma and then check the availability of plasma in the application.
3.	Novelty / Uniqueness	It is not unique problem, these kind of problems faced by the new users or who don't know to how use the application
4.	Social Impact / Customer Satisfaction	The user will be more satisfied with the solution and there will be less chance of these kind of problem that repeat again.
5.	Business Model (Revenue Model)	On the revenue bases, this donar application will be profit for Hospital, NGO's and private organizations.
6.	Scalability of the Solution	The mindset of the user about the application will changed and also as per the user flexibility the requirements can be modified.

Proposed solution –



CHAPTER-4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

In software engineering and systems engineering, a functional requirement defines a function of a system or its components.

Access Website:

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

Software operator Registration:

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

New Releases:

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

Software operator log-in:

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

Search result in a list view:

Search result can be viewed in a list. Each element in the list represents a specific donor. Each element should include first name, gender, plasma group, location, contact according to the software operator position.

Request plasma:

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

View Request:

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

Search plasma Bank Stock:

Receiving the blood or plasma request from Clinic, the blood or plasma stock in the Blood or plasma Bank Inventory will be searched to match the requested blood or plasma request.

View Blood or plasma request Details:

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

View Distribution Status:

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

4.2 NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non- functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specifies behaviors.

Maintainability:

The plasma donar application System have must have high level of Maintainability.

Serviceability

If issue arises in the plasma donar application System, then then project must be programmed in such a way that developer can service it again.

Environmental

The plasma donar application System must be working in latest operating system environments like windows 7, windows 8, windows 10 and on Linux.

Data Integrity

All the data in the plasma donar application System must be accurate and reliable. **Usability**

The plasma donar application System must have a good looing user friendly interface.

Recoverability

The plasma donar application System must have a proper data backup mechanism.

Interoperability

The plasma donar application System must work with or use the parts or equipment of another system.

Capacity

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

Performance

The plasma donar application System must perform well in different scenarios.

Security

The plasma donar application System must be secured with proper user name and passwords.

Regulatory

The plasma donar application System must obey all the governmental requirements and constraints.

Availability

The plasma donar application System must be available 24 hours a day with no bandwidth issues.

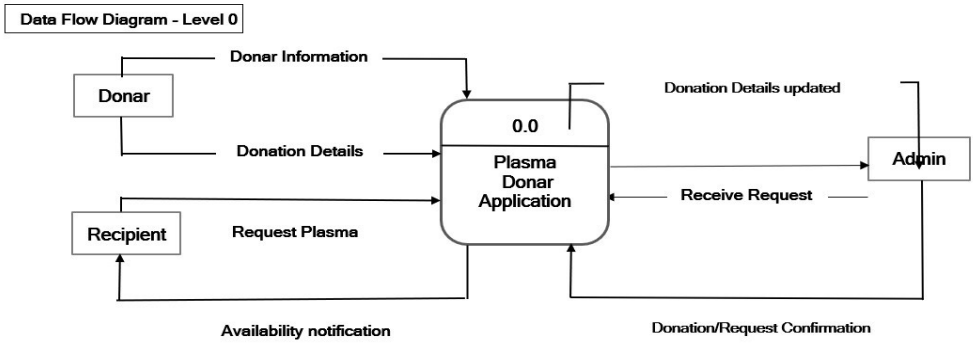
Manageability

The plasma donar application System must Alerts when the system suffers from a recoverable interruption.

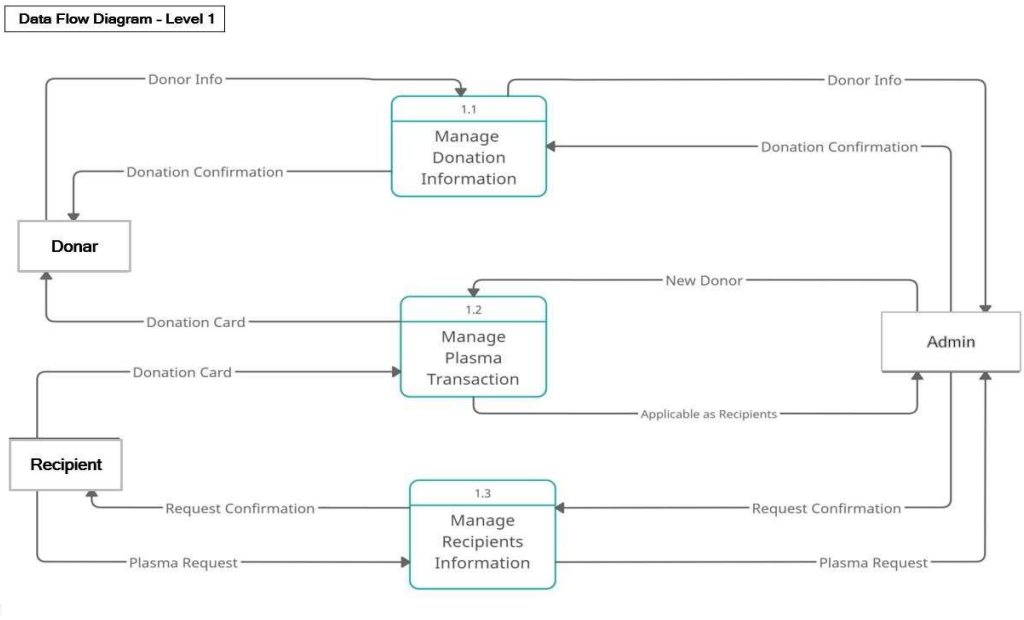
CHAPTER-5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



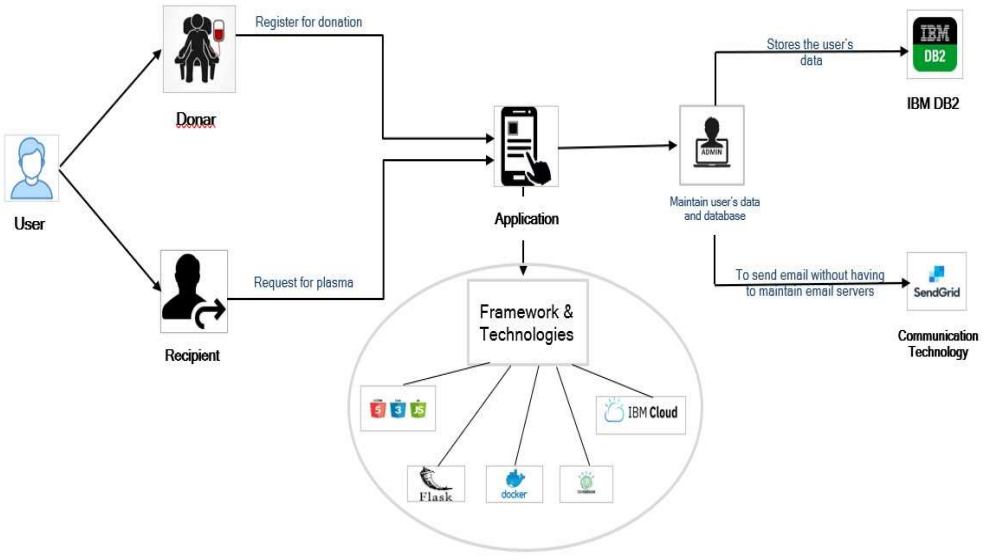
Data Flow Diagram – Level 0



Data Flow Diagram – Level 1

5.2 SOLUTION AND TECHNOLOGY ARCHITECTURE

Technical Architecture:



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, phone number, password.	I can access my account / profile.	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive verification email for confirmation.	High	Sprint-1
		USN-3	As a user, I can register for the application through social media site/account.	I can register & access my account/profile with social media account.	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail, Yahoo mail, Outlook…	I can register the app with email account.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password.	I can register & access user profile/account with Gmail account.	High	Sprint-1
	Requesting/ recipient	USN-6	As a recipient, I can request the blood group for which I need plasma.	I can get plasma through Donation center while plasma is available.	High	Sprint-2
Customer (Web user)	Profile	USN-7	As a user, I can see registration page, login page and chat bot for which the user can access to donate and to request for the required blood group plasma.	I can login through email and social media account for registration.	Medium	Sprint-2
Customer Care Executive	Help desk /User support	USN-8	As a customer care executive, I can solve the queries of the users.	I can reply to their queries and solve their related problems.	High	Sprint-3
Administrator	Registration	USN-9	As an Administrator, I can view the database of the registered users.	I can check and verify the persons who are the registered their mail Id's and information's.	Medium	Sprint-4
	Dashboard	USN-10	As an Administrator, I can view how many members requested for what kind of blood group for plasma.	I can check the number of requirements and monitor the availability.	Low	Sprint-4
Chabot	User-Interface	USN-11	In addition to the customer care executive, I can solve all the queries of the donor as well as the recipient.	I can reply to all the Questions which are asked by the users that are related to the service we provided.	Medium	Sprint-4

CHAPTER-6

6.1 SPRINT PLANNING AND ESTIMATION

Project Tracker:

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

Velocity: Sprint – I to 4

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

$$\text{average velocity (AV)} = \frac{\text{Velocity}}{\text{Sprint duration}}$$

$$AV = 20/6 = 3.34$$

Average Velocity = 3.34

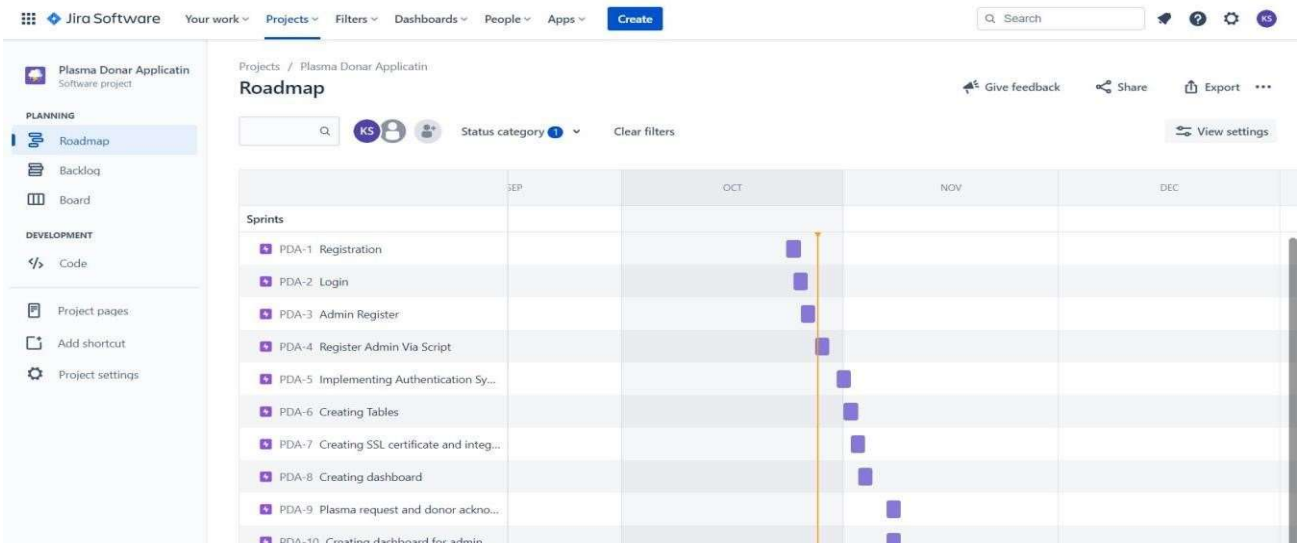
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	A User can register and create the user account.	6	High	Palanikumar, Boopathi
Sprint-1	Login	USN-2	A User can sign-in to the application by entering the registered email id and password.	6	High	Sa
Sprint-1	Admin Register	USN-3	An admin can register through the admin registry.	4	Medium	Kishore kumar S Bharath B
Sprint-1	Register Admin Via Script	USN-4	Creating an Admin Account using a python script. As for security reasons we should implement a separate python script.	4	High	Bharath B Kosalaraman A Madhankumar R
Sprint-2	Implementing Authentication System	USN-5	creating an authentication system for both admin and users using flask application	6	High	Kishore kumar S Kosalaraman A
Sprint-2	Creating Tables	USN-6	Creating Db2 account and creating the tables in DB2 in IBM cloud db2	4	Medium	Madhankumar R Bharath B

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Creating SSL certificate and integrating with python code	USN-7	Creating the SSL certificate to connect db2 via python code.	6	High	Kishore kumar S Kosalaraman A
Sprint-2	Creating dashboard	USN-8	Admin and Donor can interact with our application.	4	Medium	Bharath B
Sprint-3	Plasma request and donor acknowledge feature	USN-9	Admin can create plasma requests which will be shown in the user portal.	6	High	Kosalaraman A Madhankumar R
Sprint-3	Creating dashboard for admin	USN-10	Admin dashboard, admin can view the total request has been requested for plasma by the recipient/user.	6	High	Kishore kumar S Kosalaraman A
Sprint-3	Integrating the Watson chat bot	USN-11	Users can use the chatbot for basic clarification using the chatbot.	4	Medium	Madhankumar R Kishore kumar S
Sprint-3	Integration with SendGrid.	USN-12	The source/verification mail for both user(donar and recipient) .	4	Medium	Madhankumar R Bharath B
Sprint-4	Docker installation	USN-13	Installing Docker CLI	4	Low	Bharath B

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Creating docker image	USN-14	Setting up the docker environment and creating the docker image file	6	High	Bharath B Madhankumar R
Sprint-4	Kubernetes	USN-15	creating pods in Kubernetes and uploading it in IBM cloud	6	Medium	Kishore kumar S Madhankumar R Kosalaraman A
Sprint-4	End-to-End Testing	USN-16	Implementing End-to-End testing	6	High	Kishore kumar S Kosalaraman A

6.2 REPORT FROM JIRA



CHAPTER-7

CODING AND SOLUTIONING

7.1 FEATURE CODE

```
<!DOCTYPE html>

<html>


<head>

   <meta charset="UTF-8">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>REGISTRATION PAGE</title>

</head>

<style>

   body {

      background-image: linear-gradient(92.7deg, rgb(59, 170, 201) 8.5%,
rgb(246, 244, 198) 90.2%);

      font-family: 'Times New Roman', Times, serif;

   }


   input:hover {

      border-color: rgb(25, 20, 20);

   }


   button:hover {

      background-color: darkgray;
```

```
        border-color: black;
    }

    h1 {
        font-family: 'Courier New', Courier, monospace;
        color: darkolivegreen;
    }
    #qwerty {
        margin-top: 15em;
    }
</style>

<body>
    <center id="qwerty">
        <H1>REGISTRATION FORM</H1>
        <!--  -->
        <form action="http://localhost:5000/register" method="POST">
            <table>
                <tr>
                    <td><label for="text">USERNAME</label></td>
                    <td> </td>
                    <td><input type="text" placeholder="Enter Username"
name="username" id="username"></td>
                </tr>
```

```
<tr></tr>

<tr></tr>

<tr>

   <td><label for="text">EMAIL ID</label></td>

   <td> </td>

   <td><input type="text" placeholder="Enter email_id"
name="email_id" id="email_id"></td>

</tr>

<tr></tr>

<tr></tr>

<tr>

   <td><label for="text">PHONE NUMBER</label></td>

   <td> </td>

   <td><input type="text" placeholder="Enter PHONE Number"
name="phone_no" id="phone_no"></td>

</tr>

<tr></tr>

<tr></tr>

<tr>

   <td><label for="text">PASSWORD</label></td>

   <td> </td>

   <td><input type="text" placeholder="Enter PASSWORD"
name="password" id="password"></td>

</tr>

<tr></tr>
```

```
            <tr></tr>
        </table>
        <br>
        <center><button onclick="asd()" type="submit">Submit</button>
        </center>
    </form>
  </center>
</body>
<script>
  function asd() {
    var username1 = document.getElementById("username");
    var email_id = document.getElementById('email_id');
    var phone_no = document.getElementById('phone_no');
    var password = document.getElementById('password');
    if (username1.value == "" || phone_no.value == "" || password.value ==
"") {
        username.style.borderColor = "red";
    }
    else if (email_id.value == "") {
        email_id.style.borderColor = "red";
    }
    else if (phone_no.value == "") {
        phone_no.style.borderColor = "red";
    }
```

```
        else if (phone_no.value == "") {
            password.style.borderColor = "red";
        }
    }
</script>
</html>
```

Login Page

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    body {
        background-image: linear-gradient(92.7deg, rgb(243, 178, 178) 8.5%,
rgb(246, 244, 198) 90.2%);
        font-family: Georgia, 'Times New Roman', Times, serif;
    }
    button:hover {
        background-color: darkgray;
        border-color: black;
```

```
}
h1 {
    font-family: 'Courier New', Courier, monospace;
    color: darkolivegreen;
    top: 10em;
} .container1 {
    border: 2px solid black;
    border-color: black;
    border-radius: 10px;
    width: 400px;
}
.top {
    margin-top: 100px;
}
input:hover {
    border-color: rgb(25, 20, 20);
}
a {
    text-decoration: none;
}
a:link {
    color: #006600;
    text-decoration: none;
}
```

```
    a:visited {
        color: rgb(215, 117, 92);
        text-decoration: none;
    }

    a:hover {
        color: rgb(128, 105, 255);
        text-decoration: none;
    }

    a:active {
        color: rgb(202, 99, 75);
        text-decoration: none;
    }
</style>

<body>
    <center>
        <h1 class="top">IBM</h1>
        <div class="container1">
            <br>
            <h1>LOGIN</h1>
            <form action="http://localhost:5000/login" method="POST">
```

```
            <table>
                <tr>
                    <td><label for="text">USERNAME</label></td>
                    <td><input type="text" name="username"
placeholder="ENTER USERNAME" /></td>
                </tr>
                <tr>
                    <td><label for="text">PASSWORD</label></td>
                    <td><input type="text" name="password"
placeholder="ENTER PASSWORD"></td>
                </tr>
            </table>
            <br>
            <button type="submit">SUMBIT</button>
        </form>
        <br>
    </div>
    <br>
    <label for="text">New User!! <br> <br> <b><a
href="http://http://localhost:5000">SIGN UP</a></b></label>
  </center>
</body>
</html>
```

Welcome page

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <title>WELCOME</title>
</head>
<style>
    h1 {
        font-family: 'Courier New', Courier, monospace;
        color: darkolivegreen;
        top: 10em;
    }
    body {
        background-image: linear-gradient(92.7deg, rgb(59, 170, 201) 8.5%, rgb(246, 244, 198) 90.2%);
        font-family: 'Times New Roman', Times, serif;
    }
    a:hover {
        color: rgb(128, 105, 255);
        text-decoration: none;
```

```
    }

    .font {
        color: rgb(141, 18, 100);
        font: bold;
        font-size: 27px;
    }
    #top {
        margin-top: 200px;
    }
</style>

<body>
    <center>
        <H1 id="top">LOGIN SUCCESSFUL</H1>
        <table>
            <tr class="font">
                <td>WELCOME</td>
                <td> </td>
                <td>{{username}}</td>
            </tr>
            <!-- <tr class="font">
                <td>EMAIL_ID</td>
                <td> </td>
```

```
            <td>{{email_id}}</td>
        </tr>
        <tr class="font">
            <td>PHONE NO</td>
            <td> </td>
            <td>{{phone_no}}</td> -->
        <!-- </tr> -->
    </table>
    <table>
        <tr>
            <h1>CONNECT TO</h1>
        </tr>
        <tr>
            <td>
                <label"><a href="https://in.linkedin.com/company/ibm"><i
class="fa fa-linkedin-square"
                        style="font-size:36px"></i></a></label>
            </td>
        </tr>
        <tr>
            <td><a href="https://www.instagram.com/ibm/?hl=en"><i class="fa
fa-instagram"
                    style="font-size:36px"></i></a></td>
        </tr>
```

```
        <tr>
            <td><a href="https://www.facebook.com/IBM/"><i class="fa fa-facebook" style="font-size:36px"></i></a>
            </td>
        </tr>
        <tr>
            <td><a href="https://twitter.com/ibm?lang=en"><i class="fa fa-twitter" style="font-size:36px"></i></a>
            </td>
        </tr>
    </table>
    </table>
  </center>
</body>
</html>
```

CHAPTER-8

TESTING

8.1 TEST CASE

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup page whenever get into the application		1.Enter URL and click go 2.Click on the login/signup page 3.Verify login/Signup by entering the details	-	Login/Signup page should display	Working as expected	Pass				kosalaraman
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup page		1.Enter URL and click go 2.Click on Login/signup and get into next respective page. 3.Verify login/Signup page with below UI elements: a.email text box b.password text box c.Login button d.New User? Create account link	-	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link	Working as expected	Pass				Kishore kumar
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL and click go 2.Click on login button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: demo@gmail.com password: 12345678	User should navigate to Donar/Recipient requesting page	Working as expected	pass				Madhankumar
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL and click go 2.Click on login button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: demo@gmail password: Testing123	Application should show 'Incorrect email or password' validation message.	Working as expected	pass				Bharath
LoginPage_TC_005	Functional	Login page	Verify Admin is able to log into application with Valid credentials		1.Enter URL and click go 2.Click on login button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: adminrrr@gmail.com password: admin@rrr	Admin should navigate to Donar/Recipient requesting page	Working as expected	pass				Kishore kumar
LoginPage_TC_006	Functional	Login page	Verify Admin is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: adminrrr@gmail.com password: Adminrrr@	Application should show 'Incorrect email or password' validation message.	Working as expected	pass				kosalaraman

8.2 USER ACCEPTANCE TEST

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] 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	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	4	2	4	20
Duplicate	1	0	1	0	2
External	2	2	1	1	6
Fixed	4	1	1	10	16
Not Reproduced	0	0	0	0	0
Skipped	1	1	0	1	3
Won't Fix	0	2	2	0	4
Totals	18	10	7	16	51

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	9	0	0	9
Client Application	10	0	0	10
Security	1	0	0	1
Outsource Shipping	0	0	0	0
Exception Reporting	9	0	0	9
Final Report Output	9	0	0	9
Version Control	1	0	0	1

CHAPTER-9

RESULT

9.1 PERFORMANCE METRICS

- **Formal code metrics** - Such as Lines of Code (LOC), code complexity, Instruction Path Length, etc. In modern development environments, these are considered less useful.

- **Developer productivity metrics**—Such as active days, assignment scope, efficiency and code churn. These metrics can help you understand how much time and work developers are investing in a software project.

- **Agile process metrics**—Such as lead time, cycle time and velocity. They measure the progress of a dev team in producing working, shipping-quality software features.

- **Operational metrics**—Such as Mean Time Between Failures (MTBF) and Mean Time to Recover (MTTR). This checks how software is running in production and how effective operations staff are at maintaining it.

- **Test metrics**—Such as code coverage, percent of automated tests, and defects in production. This measures how comprehensively a system is tested, which should be correlated with software quality.

- **Customer satisfaction**—Such as Net Promoter Score (NPS), Customer Effort Score (CES) and Customer Satisfaction Score (CSAT). The ultimate measurement of how customers experience the software and their interaction with the software vendor.

CHAPTER – 10

ADVANTAGES AND 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

The efficient way of finding plasma donor for the infected people is implemented using the plasma donor website that is hosted on Aws platform. To ensure the smooth functioning of the website operations.

I have hosted the website in AWS platform to make sure the operations are running successfully AWS lambda function is used and to deploy the application AWS EC2 service is used.

CHAPTER-12

FURTURE SCOPE

Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures that many plasma donors can be added into the community.

Using elastic load balancer, it helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible downtime.

CHAPTER-13

APPENDIX

SOURCE CODE:

For source code Check out the below link:

https://github.com/IBM-EPBL/IBM-Project-9437-1659006030/tree/main/PROJECT%20DEVELOPMENT%20PHASE

GitHub and Project demo link GitHub link:

https://github.com/IBM-EPBL/IBM-Project-9437-1659006030

Project demo link:

https://drive.google.com/drive/folders/14GpxrfY6oSbb1x_pOdw1yVD-oqOcgBsK