PLASMA DONOR APPLICATION
(TEAM ID:PNT2022TMID34869)

PROJECT REPORT

Submitted by

SHRRUTHI ND (962819104077)

VIGNESH MS (962819104088)

VIGNESH KUMAR A (962819104306)

RATHISHA M (962819104071)

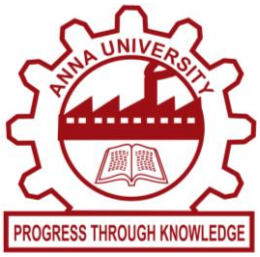
in partial fulfilment for the award of degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



UNIVERSITY COLLEGE OF ENGINEERING, NAGERCOIL

ANNA UNIVERSITY: CHENNAI 600 025

DECEMBER 2022

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

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. Regarding the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

The necessity of blood has become a significant concern in the present context all over the world. Due to a shortage of blood, people couldn't save themselves or their friends and family members. A bag of blood can save a precious life. Statistics show that a tremendous amount of blood is needed yearly because of major operations, road accidents, blood disorders, including Anaemia, Haemophilia, and acute viral infections like Dengue, etc. Approximately 85 million people require single or multiple blood transfusions for treatment. Voluntary blood donors per 1,000 populations of some countries are quite promising, such as Switzerland (113/1,000), Japan (70/1,000), while others have an unsatisfying result like India has 4/1,000, and Bangladesh has 5/1000. Recently a life-threatening virus, COVID-19, spreading throughout the globe, which is more vulnerable for older people and those with pre-existing medical conditions. For them, plasma is needed to recover their illness. Our Purpose is to build a platform with clustering algorithms which will jointly help to provide the quickest solution to find blood or plasma donor. Closest blood or plasma donors of the same group in a particular area can be explored within less time and more efficiently.

1.2 PURPOSE

In a plasma-only donation, the liquid portion of the donor's blood is separated from the cells. Blood is drawn from one arm and sent through a high-tech machine that collects the plasma. The donor's red blood cells and platelets are then returned to the donor along with some saline. The process is safe and only takes a few minutes longer than donating whole blood. Donated plasma is frozen within 24 hours of being donated to preserve its valuable clotting factors. It can be stored for up to one year and thawed for transfusion to a patient when needed. Red Cross donations are often used directly for hospital patient transfusions, rather than pharmaceutical uses. Only a small number of people living in the U.S. who are eligible to donate blood or source plasma donate. What's important is that we encourage all forms of donation from those who are eligible, so that they may contribute life-saving blood and source plasma to those in need.

The plasma protein therapeutics industry supports volunteerism donation in all of its forms. Source plasma donation and blood donation are critically important activities that contribute to saving lives. Source plasma and recovered plasma are used to produce therapies that treat people with rare, chronic diseases and disorders such as primary immunodeficiency, haemophilia and a genetic lung disease, as well as in the treatment of trauma, burns and shock. Whole blood donations most often are used locally in hospitals for transfusions required during surgery or other medical treatment. Find a donation centre near you! Plasma donation requires a commitment both in the amount of time for each donation and frequency of donation. Typically, it takes between one and three hours to donate source plasma, and plasma can be donated twice within a seven-day period. Whole blood donation takes less time—under 30 minutes—and donors donate less frequently—no more than once in eight weeks. The programs may fit into a donor's life differently at various times in the donor's life and are equally important in helping to fulfil a vital medical need. Doctors can use plasma to treat different kinds of serious health problems. Some of the elements in plasma, including the antibodies and chemicals that help your blood to clot, can help in medical emergencies like burns and trauma.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

[1]. DEVELOPING A PLASMA DONOR APPLICATION USING FUNCTION-AS-ASERVICE IN AWS

"Aishwarya R Gowri Jain University, Department of MCA, computer science 2021"

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 fight 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.

[2]. A WEB-BASED BLOOD BANK SYSTEM FOR MANAGING RECORDS OF DONORS AND RECEIPTS

"2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) Manvir Kaur,Nahida Nazir,Navneet Kaur,Syed Faraz Ali,Chirag Agarwal,Ujjwal Dubey,Varun Gupta,Abid Sarwar,Manik Rakhra,Omdev Dahiya."

The Online Blood Donation Management System, the purpose of which is to act as a bridge between a person who needs blood, a patient, and a blood donor. The design of an automatic blood system has become an integral part for saving the human lives, who needs blood under different situations. Since, there are various drawbacks of the pre-existing system like privacy issues for the donors, which are getting reflected directly on the interface. Thus, we have designed a robust system that will create a connection between different hospitals, blood banks etc to help the patient in any difficult situation.

[3]. BLOOD BAG: A WEB APPLICATION TO MANAGE ALL BLOOD DONATION AND TRANSFUSION PROCESSES

"2017 International Conference on Wireless Communications, Signal Processing and Networking (Wisp NET) Rehab S. Ali ,Tamer F. Hafez,Ali Badawey Ali,Nadia AbdAlsabour"

Many lives could be lost due to the difficulty in obtaining a proper blood bag, Therefore, this work aims to help citizens fulfil their needs for a safe and reliable blood group by searching for and locating a specific blood group. In this paper, we illustrate the problem of the blood bags shortage which is represented in the uncontrolled blood banks and parallel markets, lack of awareness and confidence, disappearance of the rare blood groups, and the difficulty in finding a specific blood group. Hence, we proposed the Blood Bag web-based application that is connected to a centralized database 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.

[4]. DETERMINANTS OF PLASMA DONATION: A REVIEW OF THE LITERATURE 2021

"Antoine Beurel ,Florence Terrade,J.-P. Lebaudy ,Bruno Danic"

The major contribution of Human Sciences in the understanding of the whole blood donation behaviour 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 behaviours, but also differences that indicate a need for further research regarding plasma donation.

[5]. A STUDY OF PRIVATE DONATION SYSTEM BASED ON BLOCKCHAIN FOR TRANSPARENCY AND PRIVACY

"2020 International Conference on Electronics, Information, and Communication (ICEIC) Junho Jeong,Donghyo Kim,Yangsun Lee,Jin-Woo Jung,Yunsik Son"

Donation is largely divided into sponsorship by individuals such as corporations and public administration. In the individual sponsorship, it is common to donate to a donation organization and to support the aid recipients by donation organization. Many people are reluctant to support to this donation because of the lack of transparency. In addition, many donation organizations lack transparent and formal administration due to lack of working capital. Therefore, this paper proposes a method to enhance personal transparency by enhancing the transparency of donation organizations and protecting the privacy of sponsors using block chain that is a Hyper ledger fabric.

2.2 REFERENCE

1. http://ijiird.com/wp-content/uploads/050140.pdf
2. https://ieeexplore.ieee.org/document/9844389
3. https://ieeexplore.ieee.org/document/8300136
4. https://www.researchgate.net/publication/317771478_Determinants_of_plasma_donation_A_review_of_the_literature
5. https://ieeexplore.ieee.org/document/9051328

2.3 PROBLEM STATEMENT DEFINITION

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. Regarding the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

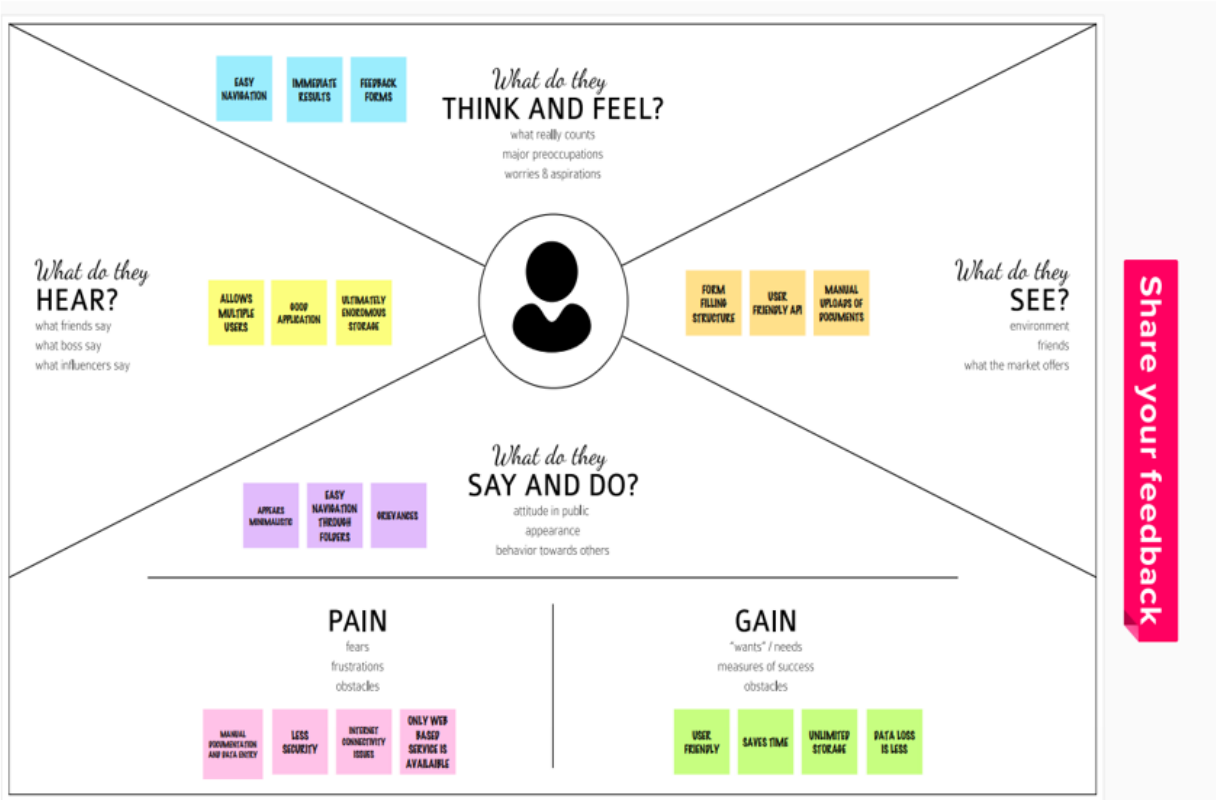
The proposed system implements a cloud-based web application as a solution to this problem. The details of the donors are fetched using a form filling structure and later embedded with the cloud storage. The receiver of the plasma has to make a request for a particular plasma in the application itself. The applicable donor and receiver are connected anonymously, and the plasma is donated. The donor and receiver are connected through a unique id which keeps the identity of both donor and receiver, hidden. The donor and receiver's information are hidden from each other providing unique Id for each user. Plasma donation can be facilitated easily through this application and hence reducing the death rates due to non-availability of plasma. The application can be linked with government hospitals and could be used in blood donation camps everywhere. As this is a web application and uses cloud storage, any further enhancements in technology can be incorporated within this application.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

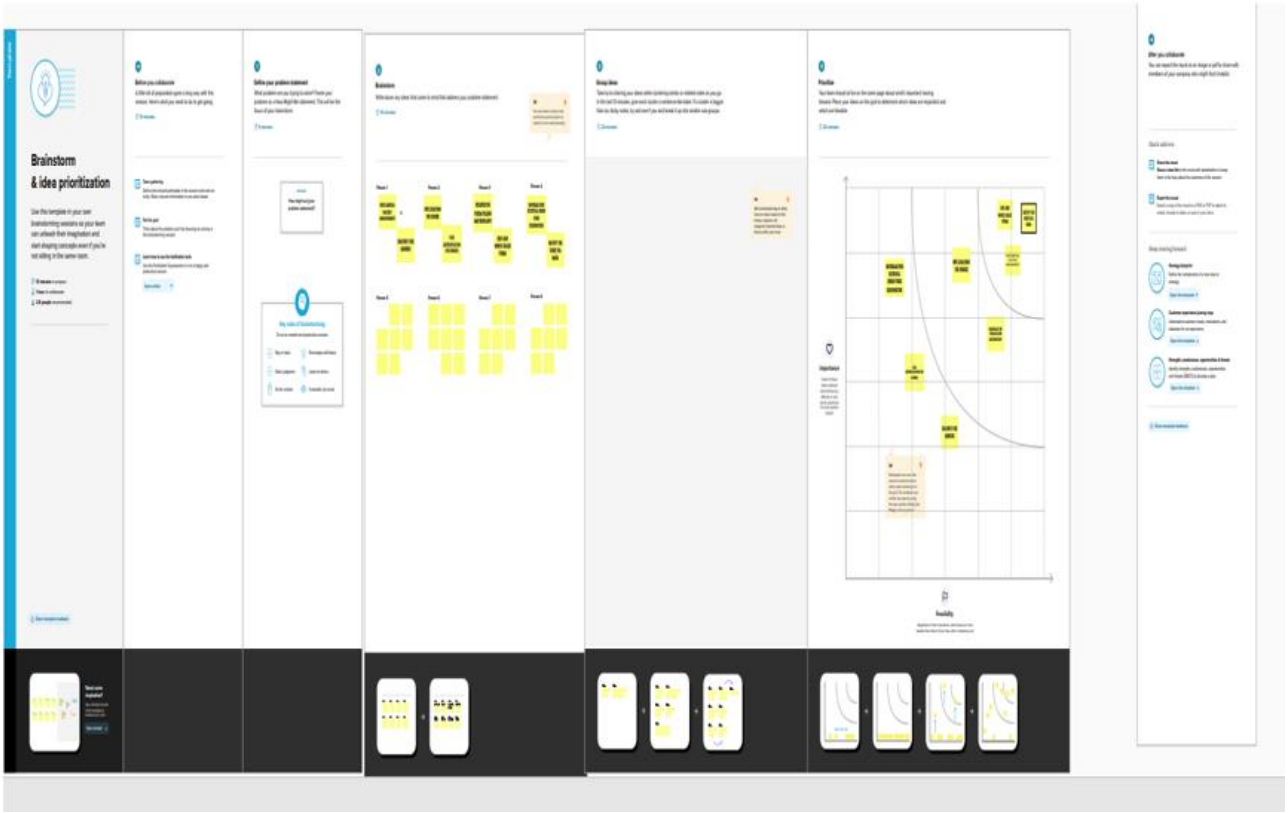
3.1 EMPATHY MAP CANVAS

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user person, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



3.2 IDEATION & BRAINSTORMING

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.

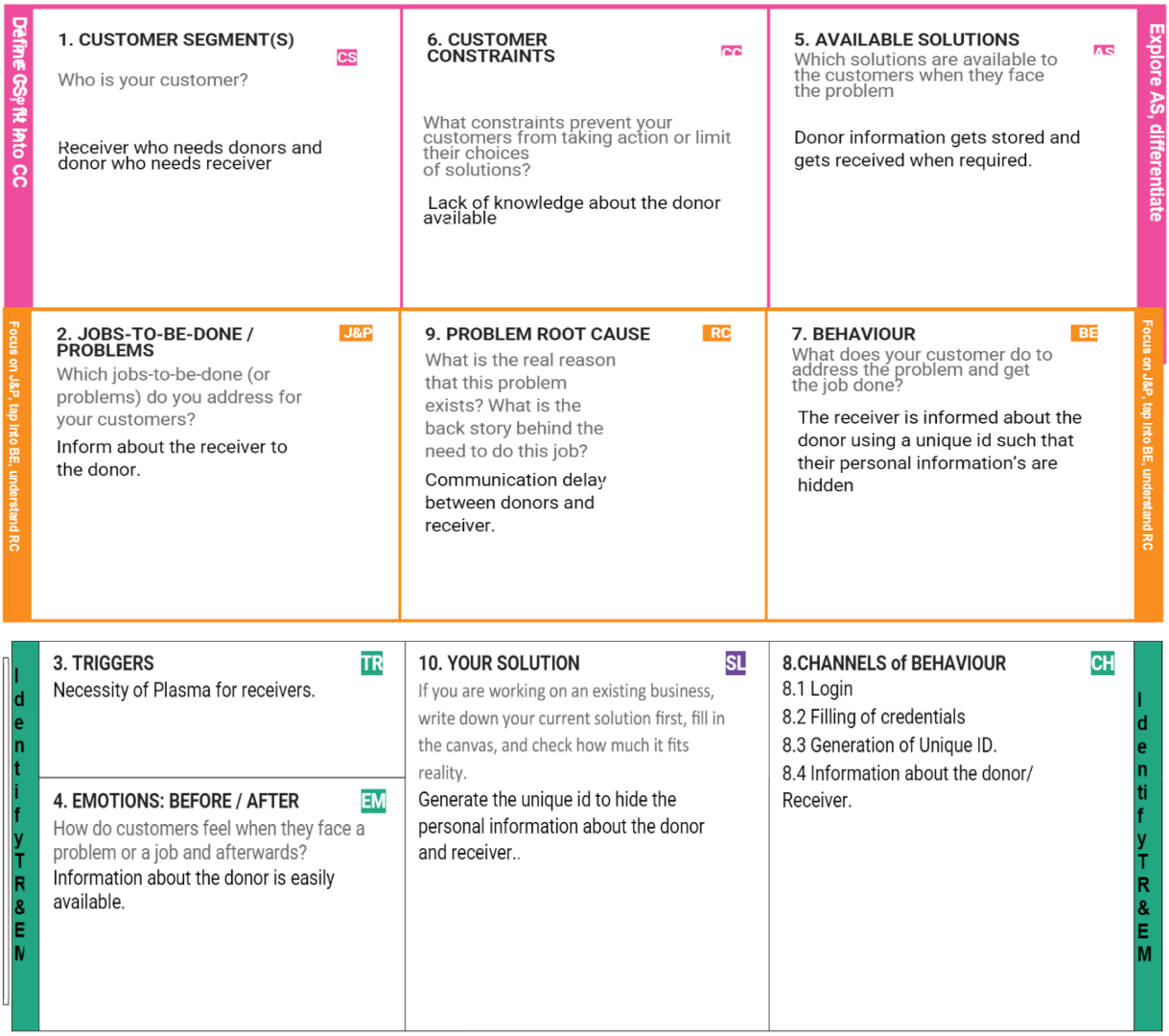


3.3 PROPOSED SOLUTION

S.NO.	PARAMETER	DESCRIPTION
1	Problem Statement	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. Regarding the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.
2	Idea/Solution Description	The proposed system implements a cloud-based web application as a solution to this problem. The details of the donors are fetched using a form filling structure and later embedded with the cloud storage. The receiver of the plasma has to make a request for a particular plasma in the application itself. The applicable donor and receiver are connected anonymously, and the plasma is donated. The donor and receiver are connected through a unique id which keeps the identity of both donor and receiver, hidden.
3	Novelty / Uniqueness	The donor and receiver's information are hidden from each other providing unique Id for each user.
4	Social Impact / Customer Satisfaction	Plasma donation can be facilitated easily through this application and hence reducing the death rates due to non-availability of plasma.
5	Business Model (Revenue Model)	The application can be linked with government hospitals and could be used in blood donation camps everywhere
6	Scalability of the Solution	As this is an web application and uses cloud storage, any further enhancements in technology can be incorporated within this application

3.4 PROBLEM SOLUTION FIT

Given below is the problem solution fit of the given project. It describes how the proposed system is suitable in various aspects such as customer constraints, available solution etc.



CHAPTER 4

REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed system.

FR NO.	FUNCTIONAL REQUIREMENT(EPIC)	SUB REQUIREMENT
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Unique ID Generation	Every user is provided with an Unique ID
FR-5	User Request	User in need requests for plasma
FR-6	User Donation	Plasma donation by suitable donor

4.2.NON-FUNCTIONAL REQUIREMENTS

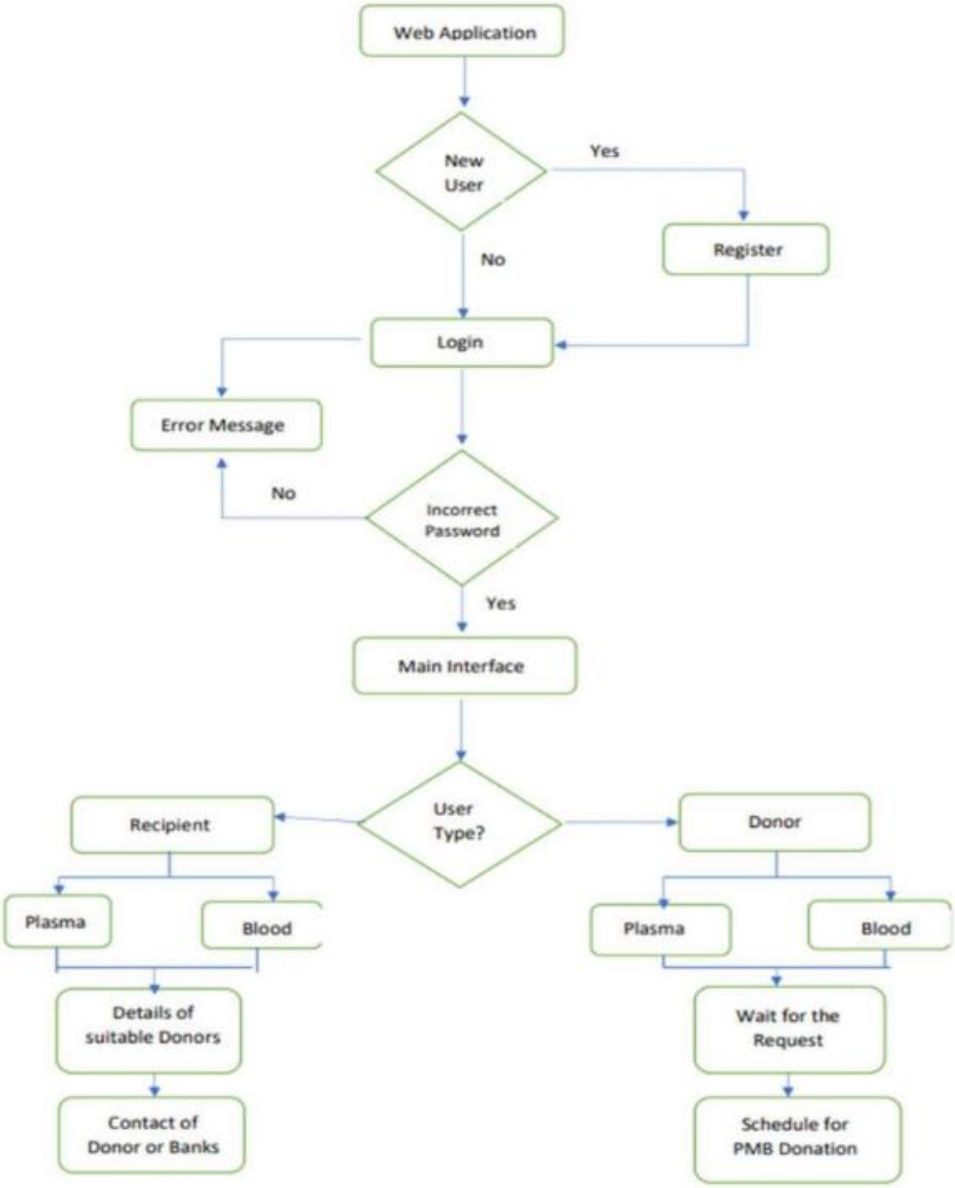
Following are the non-functional requirements of the proposed system

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Easily used by everyone
NRF-2	Security	High end security to users' data
NRF-3	Reliability	Increased reliability
NRF-4	Performance	Developed for better performance
NRF-5	Availability	Available to anyone registered in the app
NRF-6	Scalability	Improved scalability

CHAPTER 5

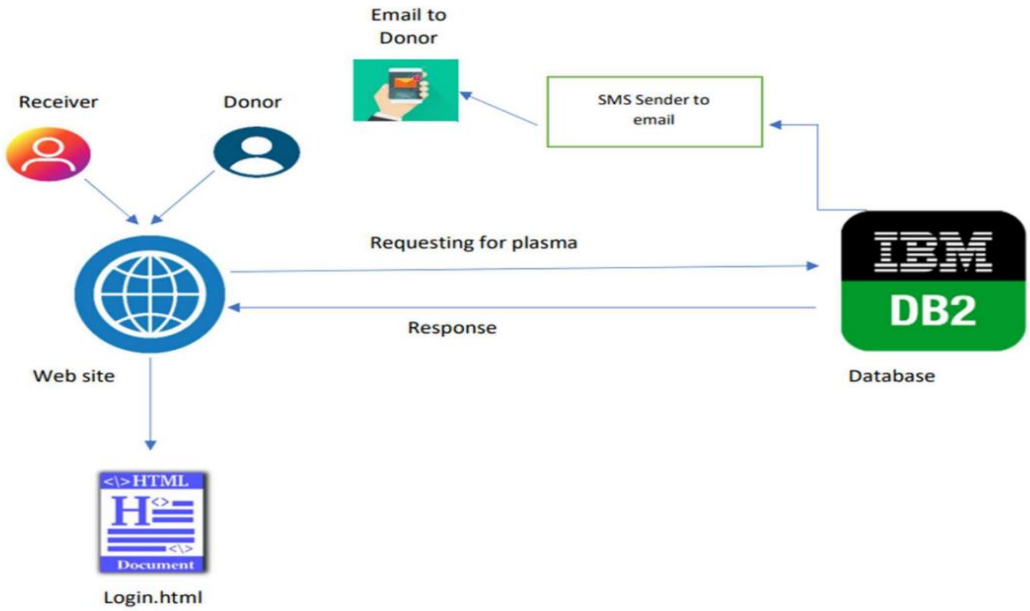
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

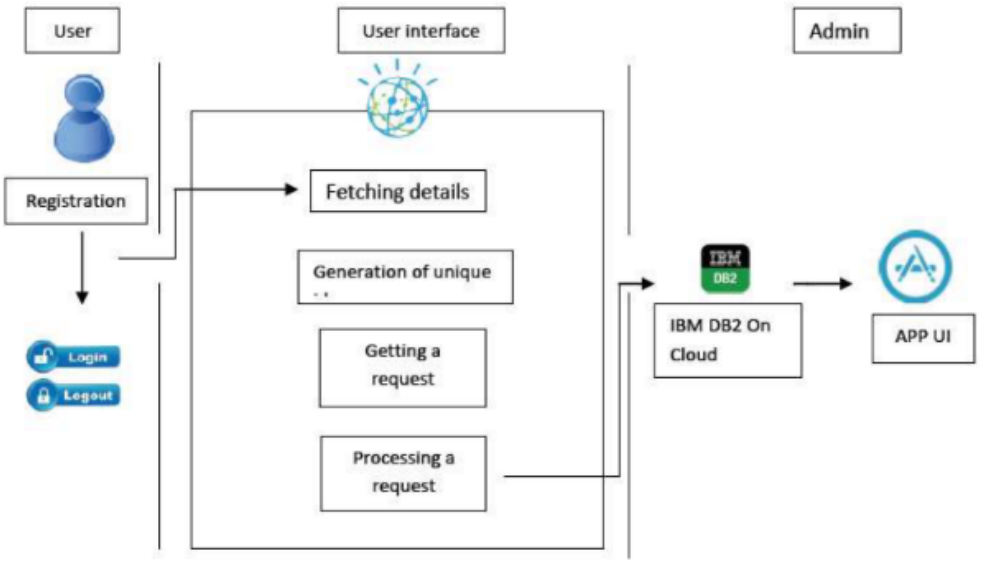


5.2 SOLUTION & TECHNICAL ARCHITECTURE

5.2.1. SOLUTION ARCHITECTURE



5.2.2. TECHNICAL ARCHITECTURE



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Numbe.	User Story / Task	Acceptance criteria	Priority	Release
Applicant (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Applicant (Web user)	Email confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Applicant (Web user)	Login	USN-3	As a user, I can log into the application by entering email & password	I can login into the website for further details	High	Sprint-1
Donor (Web user)	Details	USN-4	As a donor, I can fill my personal details like blood type	I can submit my personal details	High	Sprint-1
Receiver (Web user)	Details	USN-5	As a receiver, I can request for the particular blood type	I can request for a particular type of blood	High	Sprint-2
Administrator	Unique id generation	USN-6	As an administrator, I can generate a unique id for each user	I can generate a unique id for each user	High	Sprint-2
Administrator	Cloud Database	USN-7	As an administrator, I can store the donor details into the cloud database	I can store the data into cloud database	High	Sprint-2
Administrator	Cloud Database	USN-8	As an administrator, I can store the receiver details into the database	I can store the details into cloud database	High	Sprint-3
Receiver (Web user)	Details	USN-9	As a receiver, I can request the website for a particular donor	I can request for a plasma donor of my blood type	High	Sprint-3
Administrator	Cloud Database	USN-10	As an administrator, I can match the particular donor with receiver with unique id	I can match the donor to the receiver	High	Sprint-3
Administrator	Email	USN-11	As an administrator, I can send a mail to the donor	I can send an email to donor	High	Sprint-4
Donor (Web user)	Email	USN-12	As a donor, I can confirm the email received	I can confirm the email	High	Sprint-4
Receiver (Web user)	Email	USN-13	As a receiver, I can receive the plasma of particular type	I can receive the plasma	High	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

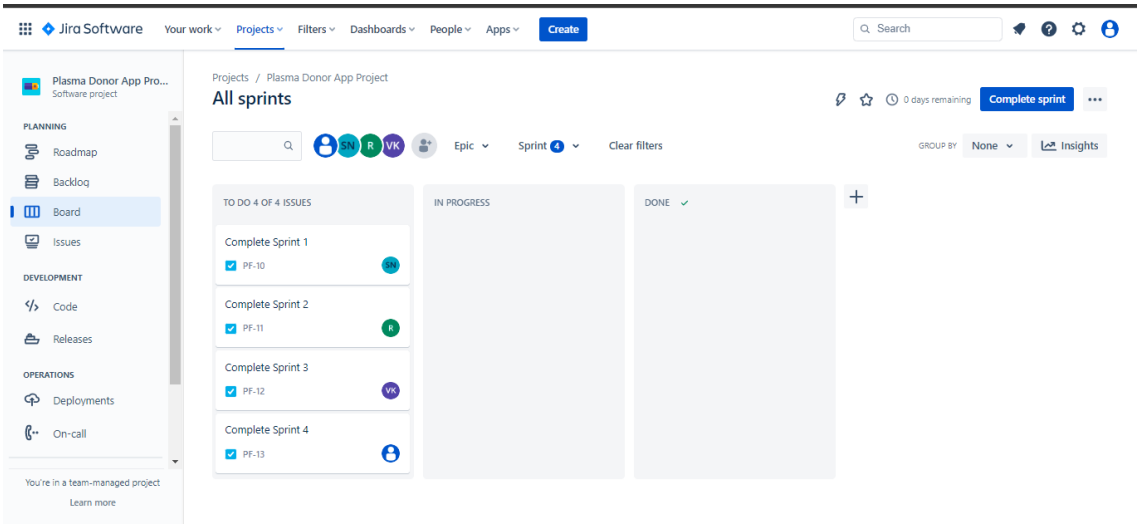
6.1 SPRINT PLANNING & ESTIMATION

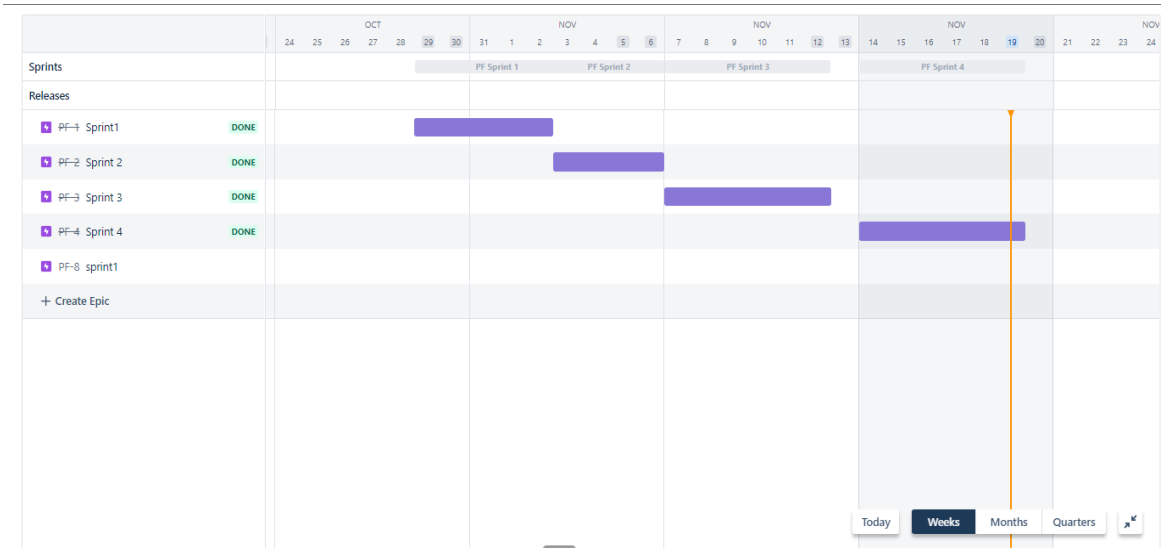
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	Accessing user account / dashboard	High	TM-3
1	Email confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	Receive confirmation Email	High	TM-2
1	Login	USN-3	As a user, I can log into the application by entering email & password	Login into the website for further details	High	TM-1
2	Details	USN-4	As a donor, I can fill my personal details	Submit personal details	High	TL
2	Details	USN-5	As a receiver, I can request for the particular blood	Request for a particular type of blood	High	TL
2	Unique id generation	USN-6	As an administrator, I can generate a unique id	Generate a unique id	High	TM-1
3	Cloud Database	USN-7	As an administrator, I can store the donor details into the cloud database	Store the data into cloud database	High	TM-2
3	Details	USN-8	As a receiver, I can request the website for a donor	I can request for a plasma donor of my blood type	High	TM-3
3	Cloud Database	USN-9	As an administrator, I can match the particular donor to receiver	I can match the donor tothe receiver	High	TM-3
4	Email	USN-10	As an administrator, I can send a mail to the donor	I can send an email todonor	High	TM-2
4	Email	USN-11	As a donor, I can confirm the email received	I can confirm the email	High	TM-1
4	Email	USN-12	As a receiver, I can receive the plasma	I can receive the plasma	High	TL

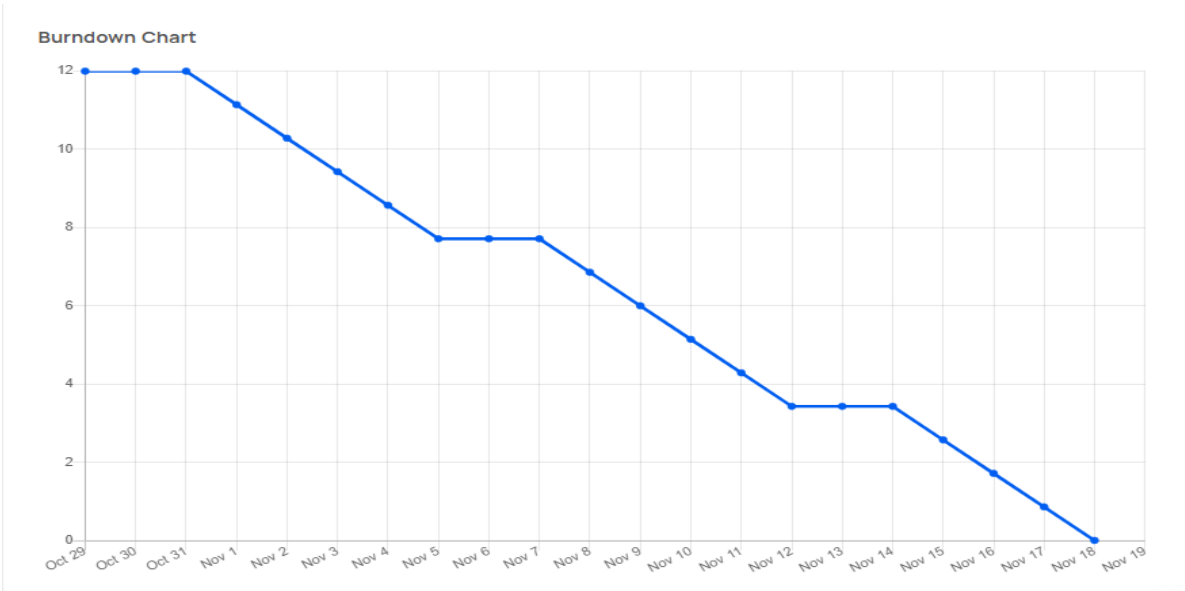
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Data (Actual)
Sprint-1	20	6 Days	29 Oct 2022	02 Nov 2022	20	03 Nov 2022
Sprint-2	20	6 Days	03 Nov 2022	06 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA





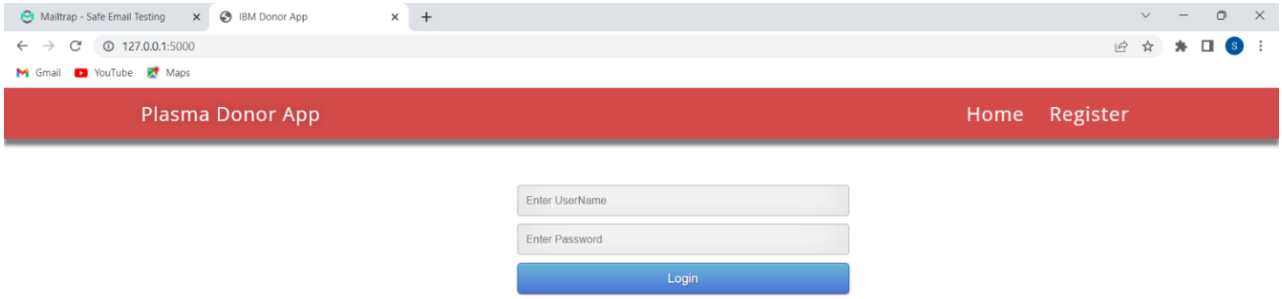


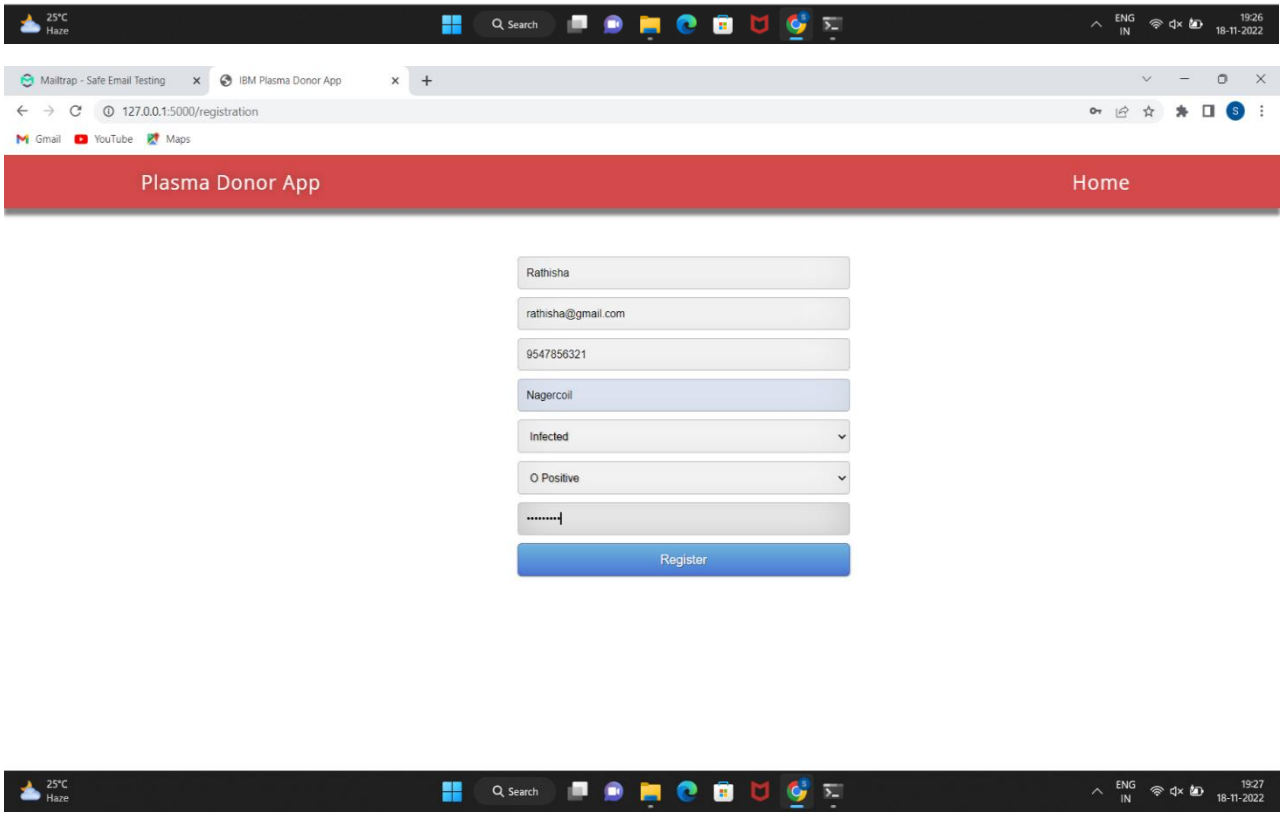
CHAPTER 7

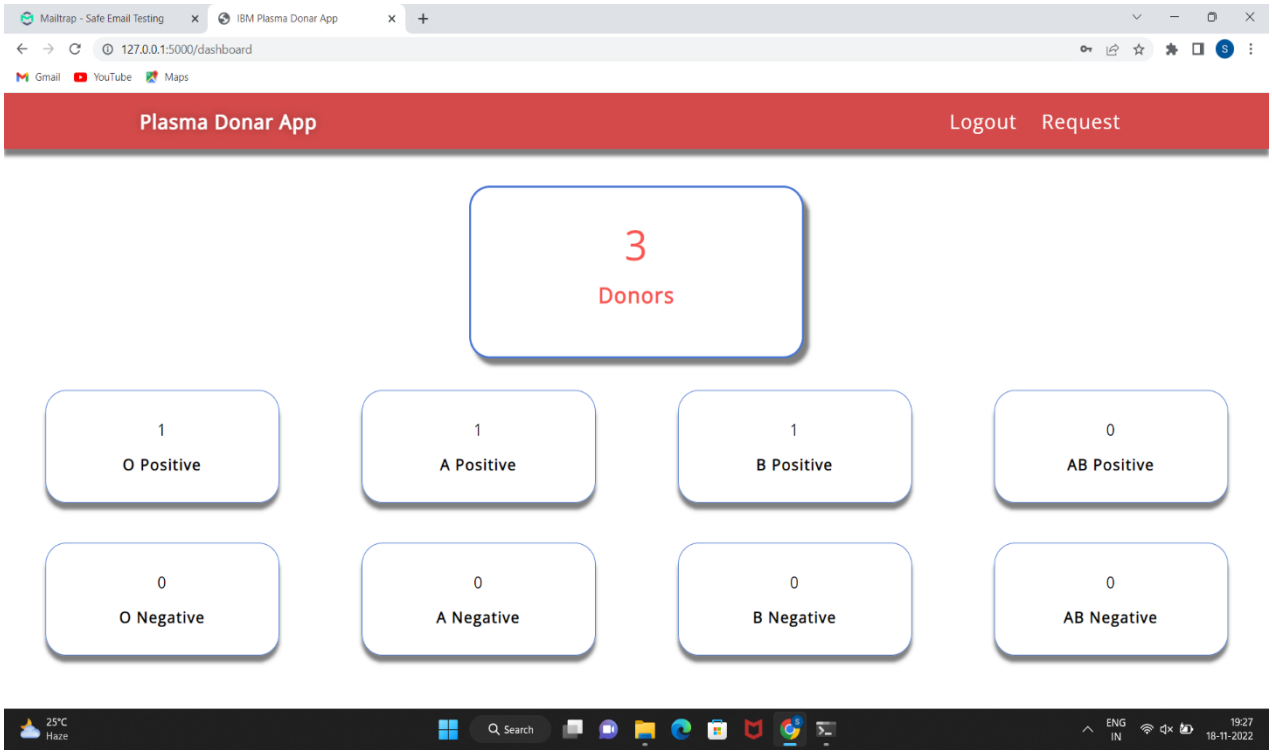
CODING & SOLUTIONING

7.1 FEATURE 1

Feature 1 describes the front end of the whole project including the login page, register page and the dashboard

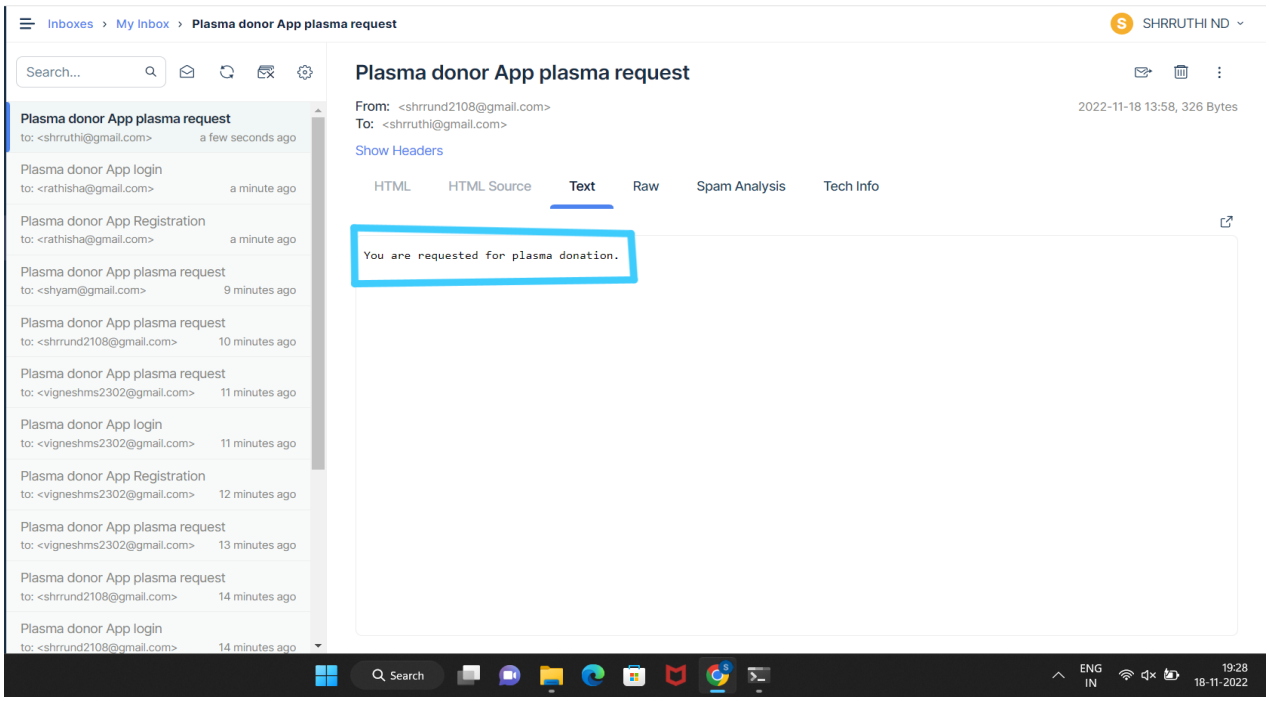




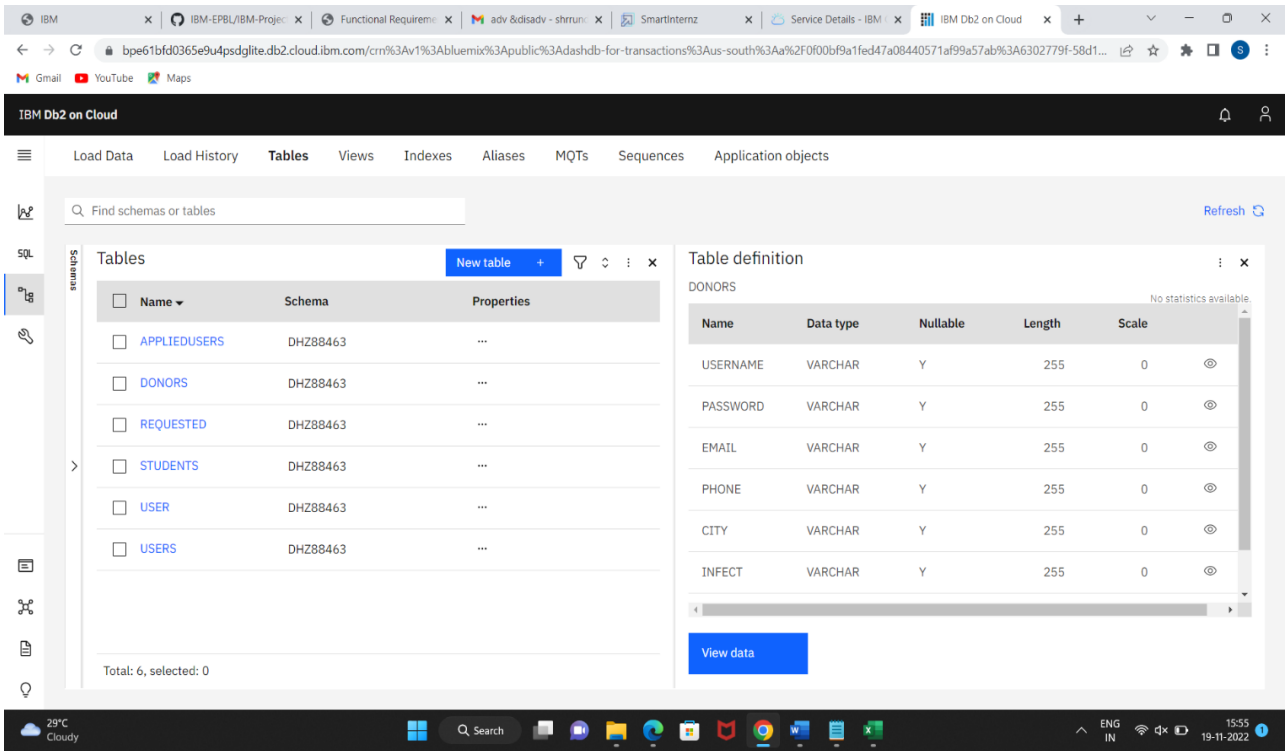


7.2 FEATURE 2

Feature 2 describes the linking of the application with email services



7.3 DATABASE SCHEMA



CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL and click go 2.Verify login/Singup popup displayed or not	http://127.0.0.1:5000	Login/Signup popup should display	Working as expected	Pass				
LoginPage_TC_OO2	UI	Home Page	Verify the user with Login/Signup	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	http://127.0.0.1:5000	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 e.Last password? Recovery password link	Working as expected	Pass				
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box	Username: Shruthi password: 123456	User should navigate to user account homepage	Working as expected	Pass				
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box	Username: Shruthi password: 123456	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				

8.2 USER ACCEPTANCE TESTING

8.2.1. Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	1	1	0	2
Duplicate	0	0	0	0	0
External	0	1	0	0	1
Fixed	10	5	2	3	20
Not Reproduced	1	0	0	2	3
Skipped	0	0	0	0	0
Won't fix	0	0	1	0	1
Total	11	7	4	5	27

8.2.2 Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	20	0	0	20
Security	2	0	0	2
Exception Reporting	5	0	0	5
Final Report Output	7	0	0	7

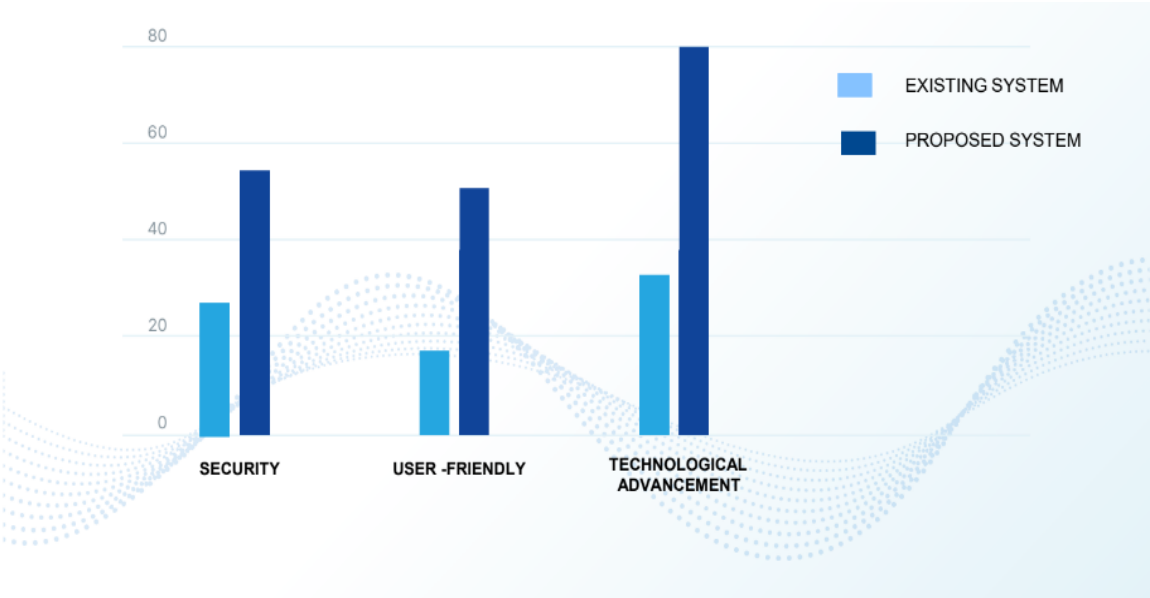
CHAPTER 9

RESULTS

9.1 PERFOMANCE METRICS

The performance metrics of a particular project can be calculated by comparing it with the existing solution. The proposed solution must have advancements in its applications.

S.NO	EXISTING SYSTEM	PROPOSED SYSTEM
1	User data is not secure	User data is secured
2	Uses traditional database	Uses cloud database
3	Not so user friendly	User friendly



CHAPTER 10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- It is a useful tool to find compatible blood donors who can receive blood request posts in their local area. Clinics can use this web application to maintain blood donation activity
- It connects plasma donors and recipients through a single and scalable platform
- The proposed system has a login page that allows only the registered user to login and thereby preventing unauthorized access.
- It is a user-friendly platform

10.2 DISVADANTAGES

- It requires a active internet connection
- The application cannot verify users' genuineness
- The server may slow down due to overload

CHAPTER 11

CONCLUSION

The report shows an approach for developing an plasma donor application integrating IBM cloud services. Nowadays everything is getting digitalized hence this Application suits best for the scenario. Donating blood helps save lives and has positive benefits for donors, such as improving their emotional and physical health. People who donate blood may also experience side effects, such as minor bruising or feeling lightheaded. This project aims to create a mobile app for android mobiles. the main aim of this project is to develop a computer system that will link all donors and blood banks. The system helps to control a blood transfusion service and create a database to hold data on donors and blood banks in each area or city. furthermore, people will be able to register as donors and thus receive a call from their local clients who needs blood to donate blood in cases of need. The app will help develop public awareness amongst its visitors of the hospitals' need for blood in order to supply the appropriate donors.

CHAPTER 12

FUTURE SCOPE

The main objective of this project is to provide the recipient with a donor who is in good form with no health ailments to donate blood of the corresponding blood group. This project provides quick access to donors for an immediate requirement of blood. In case of an emergency/surgery, blood procurement is always a major problem which consumes a lot of time. This helps serve the major time-lapse in which a life can be saved. In future this project can be integrated with governmental websites and can be used in all the plasma donation centres. The scope clearly defines the boundaries of the proposed system. The functional areas of this application that lies under the scope of the proposed system are the management of the availability of donors, hospitals, and blood banks to the user or member at any time. Upgrading the UI that is more user-friendly will help many users to access this app and also ensures that many plasma donors can be added to the community. Increasing a few features helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible downtime

CHAPTER 13

APPENDIX

13.1 SOURCE CODE

app.py

```
from distutils.log import debug

# from sendgridmail import sendmail

from flask import Flask, render_template, request, redirect, url_for, session

from flask_mail import Mail, Message

import re

import os

import ibm_db

from dotenv import load_dotenv

load_dotenv()

app = Flask(__name__)

app.secret_key = 'a'

print("Try to connect to Db2")

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;UID=dhz88463;
SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PWD=ePdzIYfMID5noyij",
","")

print("Connected Successfully")

app.config['MAIL_SERVER']='smtp.mailtrap.io'

app.config['MAIL_PORT'] = 2525

app.config['MAIL_USERNAME'] = '2f9fceb61f547b'

app.config['MAIL_PASSWORD'] = 'b6e70eb6747bec'

app.config['MAIL_USE_TLS'] = True

app.config['MAIL_USE_SSL'] = False

mail = Mail(app)

@app.route('/')

@app.route('/login')
```

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

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

```
        ibm_db.bind_param(prep_stmt, 6, infect)
        ibm_db.bind_param(prep_stmt, 7, blood)


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


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


@app.route('/requester')
def requester():
    if session['loggedin'] == True:
```

```
        return render_template('request.html')
    else:
        msg = 'Please login!'
        return render_template('login.html', msg = msg)

@app.route('/requested',methods=['POST'])
def requested():
    bloodgrp = request.form['bloodgrp']
    address = request.form['address']
    name=  request.form['name']
    email=  request.form['email']
    phone= request.form['phone']
    insert_sql = "INSERT INTO  requested VALUES (?, ?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, bloodgrp)
    ibm_db.bind_param(prep_stmt, 2, address)
    ibm_db.bind_param(prep_stmt, 3, name)
    ibm_db.bind_param(prep_stmt, 4, email)
    ibm_db.bind_param(prep_stmt, 5, phone)
    ibm_db.execute(prep_stmt)
    index(email,'Plasma donor App plasma request','You are requested for plasma donation.')
    return render_template('request.html', pred="Your request is sent to the concerned people.")
def index(usermail,subject,content):
  msg = Message(subject, sender = 'shrrund2108@gmail.com', recipients = [usermail])
   msg.body = format(content)
   mail.send(msg)
   return "Sent"
@app.route('/logout')
def logout():
   session.pop('loggedin', None)
   session.pop('id', None)
```

```
    session.pop('username', None)
    return render_template('login.html')
if __name__ == '__main__':
    app.run(host='0.0.0.0',debug='TRUE')
```

13.2 GITHUB AND PROJECT DEMO LINK

13.2.1. GitHub Link

https://github.com/IBM-EPBL/IBM-Project-23929-1659933689

13.2.2. Project Demo Link

https://drive.google.com/file/d/1ayy1Dsovl-k79XqjXkFv0pZPEOG-hvbf/view