PLASMA DONAR MANAGEMENT

TEAM ID : PNT2022TMID46887

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

1.2 SCOPE

The scope of the work is plasma donar management scheme IBM cloud web management scheme is deployed for all admin details ,user details

2.LITEATURE SURVEY

In the literature survey the current blood banking and organ banking systems rely heavily on manual labor, which takes a significant amount of time and manpower. So, if the user wants to know the details of the BIO-donors, they must connect to the internet/network Collecting information about B.I.O.-donors or receivers takes more time and effort, Users do not have access to accurate information

In the current system, there are two sorts of processes: the blood donation process by donors and the blood request process by hospitals. An administrator is in charge of controlling the blood inventory in the blood bank in both procedures.

When a new donor comes to give blood, they are prompted to fill out their personal information during the registration process before making a donation.

Following donation, the donor is given an E-donor identity card that includes their name, blood type, and a barcode that may be used as a reference for future contributions. The barcode is used to access the donor's record, which includes personal information, medical history, and donation information, including blood results. Because the Android Application is only available for usage within the organization, only B.I.O.-bank administrators have access to the donor's information. Donors find it difficult to update their personal information in the system as a result. That is, in order for donors to update their personal information

Donors' Experiences with Blood and Organ Donation When a new donor arrives to give blood, they are asked to fill out their personal information. before making a gift, during the registration procedure Following donation, the donor is given an E-donor identity card that includes their name, blood type, and a barcode that may be used as a reference for future contributions. The barcode is used to access the donor's record, which includes personal information, medical history, and donation information, including blood results. Because the Android Application is only available for usage within the organization, only B.I.O.-bank administrators have access to the donor's information. Donors find it difficult to update their personal information in the system as a result. That is, in order for donors to update their personal information

Donors' Experiences with Blood and Organ Donation When a new donor arrives to give blood, they are asked to fill out their personal information.

Having a manual ID card, on the other hand There are several disadvantages, such as the loss or damage of the ID Card. We can identify donors to ensure their safety. Other credentials, such as username and password, can act as a safeguard if their donors ID. card is lost or damaged while using the Android Application. Donors' Blood and Organ Donation Process when a new donor When people arrive to give blood, they are asked to fill out personal information.

The blood will subsequently be entered into the inventory by the Administrator. future requests the need for blood and organs is growing. since there is no substitute for human blood and organ Every day, blood and organs are required in hospitals. Emergency care of sick patients in preparation for organ transplantation beneficiaries, as well as to save the lives of accident patients with the advancements in medical techniques and treatments, blood and blood products

Continue to grow Many individuals in India are dying as a result of a dearth of blood and organs; they are unable to obtain the blood and organs on time. The victims' relatives and friends begin looking for a donor to assist them, but there is no guarantee about the Donor's presence or health condition, and there are many people who are willing to help and donate others to save their lives. A variety of current methods have been progressively attempted to activate the blood and organ donation procedure. However, this is still ineffective today. We suggest using cutting-edge technology and techniques to develop a system that bridges the gap.

EXISTING PROBLEM

Web-based blood information system

A system called donation keeps track of donations, examines different research parameters, and offers online information. Blood is given to patients by contacting the donor over the phone or the internet. Only an android uses it and the blood donation time cannot be specified by the donor.

MPlus / Kerala Blood Bank is an Android application for Keralites that includes a blood donor bank of Kerala, the ability to send a request to MPlus users, and the ability to respond directly to needier The G.P.S. system is not used in this application, which is only for blood donation

Hospitals' Requests for Organs

MPlus / Kerala Blood Bank is an Android application for Keralites that includes a blood donor bank of Kerala, the ability to submit a request to MPlus users, and the ability to reply immediately to needier. The G.P.S. system is not used in this application, which is just for blood donation. Hospitals' Requests for Blood and Organs Hospital scanrequest for blood bycallinginor emailing the blood banks the type of blood and the quantity that is in need. The Administrator is responsible for checking the availability of the blood **This Blood Bank system** is suitable for a Single Blood Bank in "A Survey Paper on E-Blood Bank and an Idea to Use on Smartphone." All information, including Blood Group, Total Units of Blood Available, Donor Information, and so on, is maintained in a database. This system assigns a unique ID to contributors in order to maintain track of their records and retrieve information in the future. If a Seeker need blood, the Doctor will simply utilise this application on his/her Smartphone to obtain the necessary information from a Blood Bank about a certain Blood Type. The usage of GPS will improve and accelerate the search technique

2.1REFERENCES

[1] Danic B. Comprendre le don: l'apport des sciences humaines à l'activité de prélèvement. Transfus Clin Biol 2003;10:146–50,

[2] Bednall TC, Bove LL. Donating blood: a meta-analytic review of selfreported motivators and deterrents. Transfus Med Rev 2011;25:317–34,

[3] Bagot KL, Murray AL, Masser BM. How can we improve retention of the first-time donor? A systematic review of the current evidence. Transfus Med Rev 2016; 30:81–91

[4] Bandura A. Self-efficacy: toward a unifying theory of behavioral change. Psychol Rev 1977; 84:191–215,

[5] Robert P. Global plasma demand in 2015. Pharm Policy Law 2009:359–67,

[6] Cataldo JF, Cohen E, Morganti JB. Motivation of voluntary plasmapheresis donors. Transfusion (Paris) 1976;16:375–9

[7] Bagot KL, Bove LL, Masser BM, Bednall TC, Buzza M. Perceived deterrents to being a plasmapheresis donor in a voluntary, nonremunerated environment. Transfusion (Paris) 2013;53:1108–19

[8] Bagot KL, Masser BM, White KM. A novel approach to increasing inventory with the current panel: increasing donation frequency by asking for a different blood product. Transfusion (Paris) 2015;55:1294–302,

[9] Bove LL, Bednall T, Masser B, Buzza M. Understanding the plasmapheresis donor in a voluntary, nonremunerated environment. Transfusion (Paris) 2011;51:2411–24

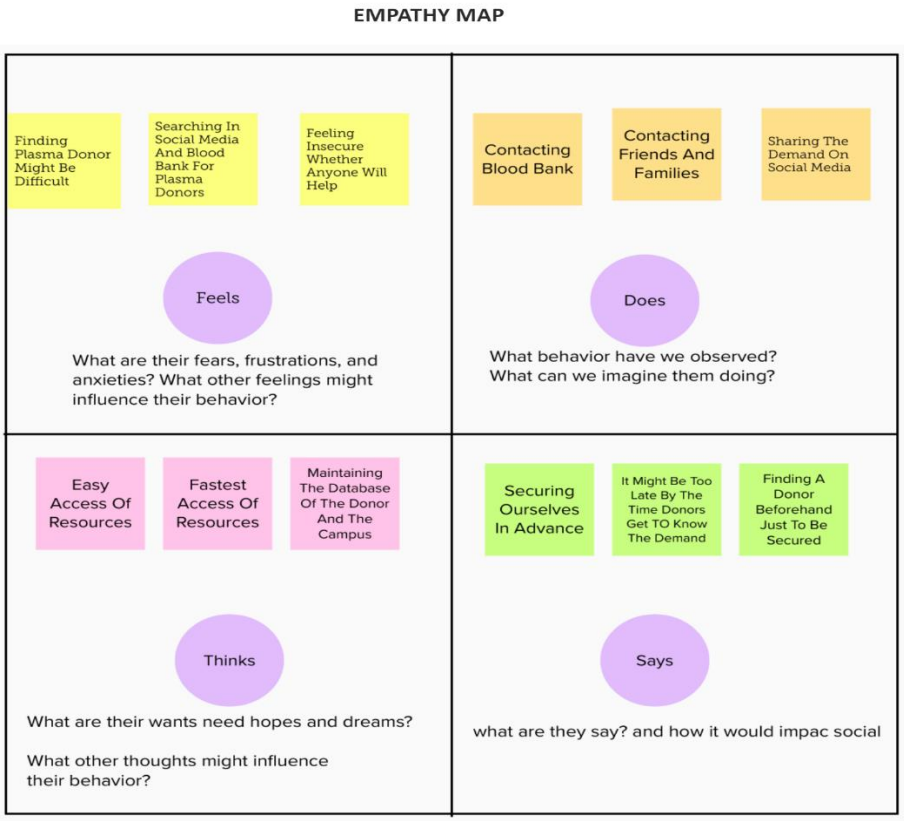
[10] Charbonneau J, Cloutier M-S, Carrier É. Whole blood and apheresis donors in Quebec, Canada: demographic differences and motivations to donate. Transfus Apher Sci 2015;53:320–8,

2.2PROBLEM STATEMENT DEFINITION

Patients with severe liver disorders and numerous clotting deficiencies are given plasma. . When a patient needs plasma, it can be challenging to get in contact with a donor within the patient's family and friends in a timely manner. It can also be challenging to get in contact with authorized donor centers. Due to a lack of awareness regarding plasma donation, there is a demand for plasma donors, making it challenging for the affected patients to locate donors. During the COVID-19 pandemic, the need for plasma increased and the donor rates decreased in order to provide an immunity boost for COVID-affected patients It takes a long time for a patient to discover the proper donor, and it also takes time for the spreading about the need for plasma donors to disseminate on social media to a larger audience. As a result, patients cannot locate the right donor within a given time frame. Our application enables patients with severe liver illnesses, blood clotting issues, and covid to quickly and easily locate the correct donor within the allotted period. The condition of the affected patient may suffer and perhaps result in death, if the appropriate plasma donor cannot be found within a certain amount of time

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAIN STORMING

Template

Brainstorm & idea prioritization

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A. Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B. Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C. Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

1. Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

To find the nearest available potential plasma donor and to maintain optimal level of blood bank reserves and a connecting platform for blood donors and blood blanks.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2. Brainstorm

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

10 minutes

Mahendra Kumar M

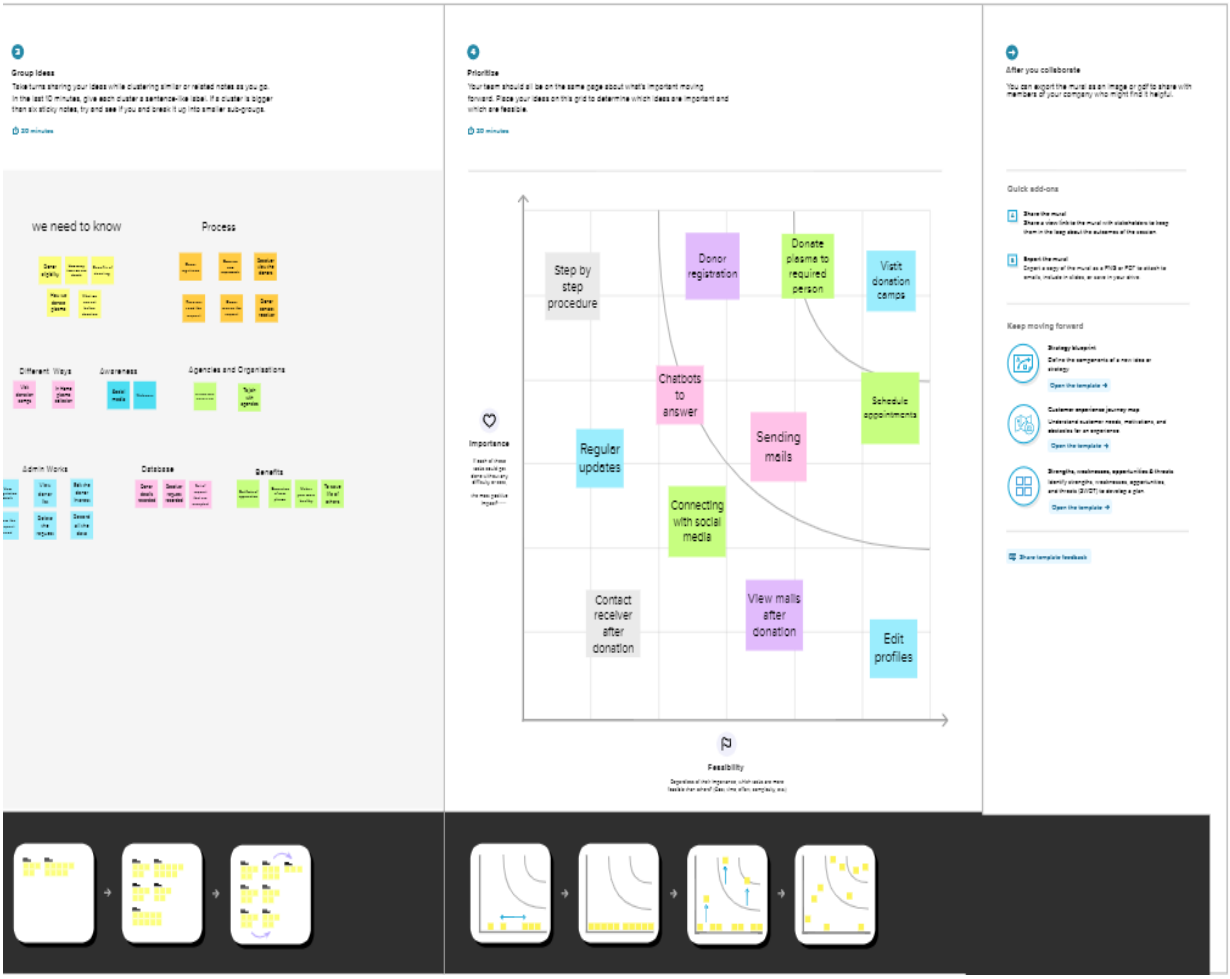
Check donor eligibility

Benefits of donating

Jithendran S

Kali Sathreesh B

Dharsan P

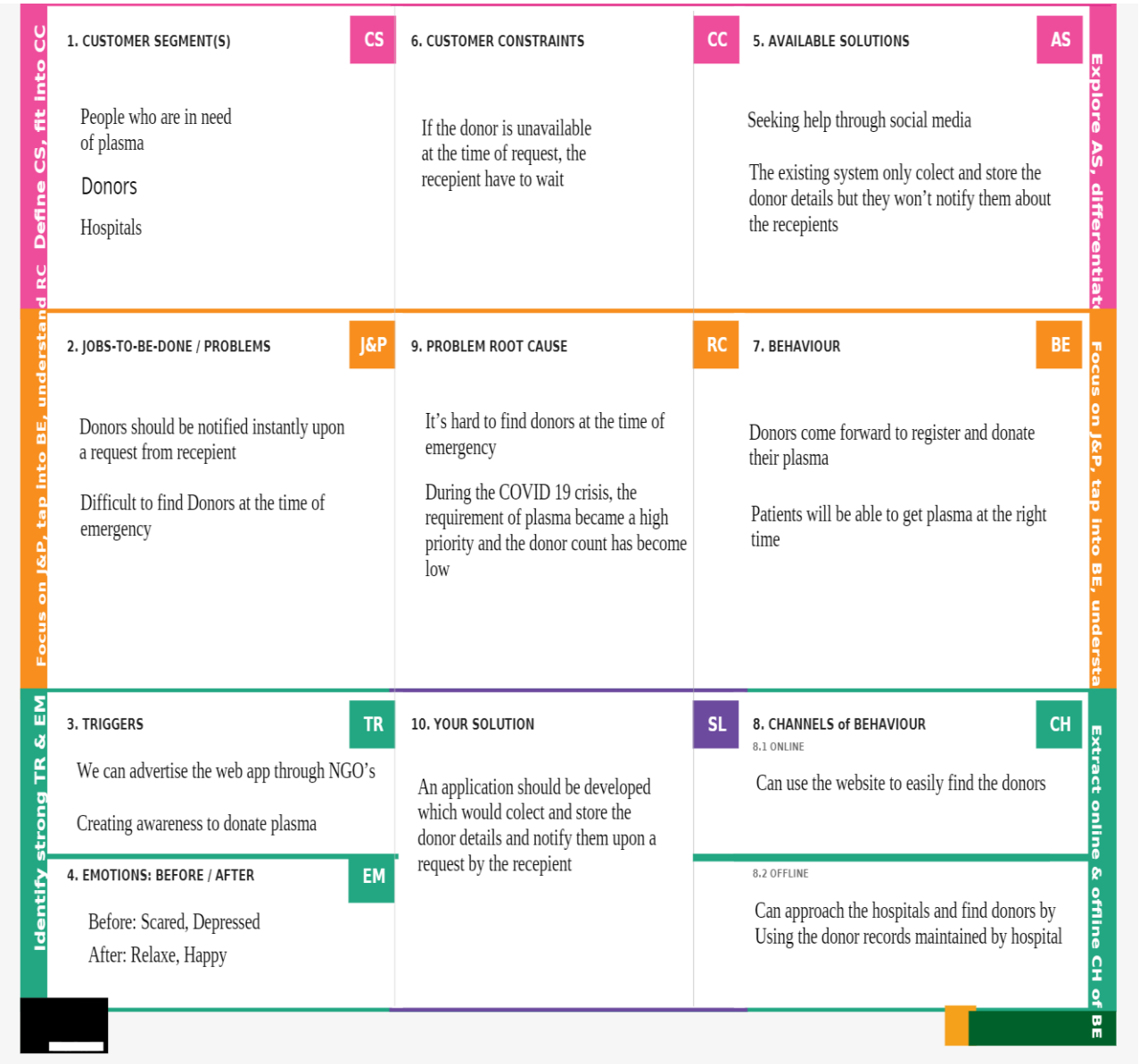


3.3 PROPOSED SOLUTIONS

To create an application for people who want to donate their plasma for the people who need it mostly in times of emergency. The application will enable people to register themselves in the portal for donating their plasma and the recipients who need it can see their details so that they can get the plasma. When the user request for plasma transmission if there is lack of plasma at the time of request, automatically user will be marked in hold back list. Later when there is availability of plasma, the receiver waiting in hold back list will be alerted via calling system. The application is user friendly and anyone with basic knowledge

can access it. The application seamlessly connects the donor and the person who need it and also hospitals who have availability of the plasma.

3.4 PROBLEM SOLUTION FIT



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Sign in / Login	Register with username, password
FR-2	Profile Registration	Register with username, password, email, blood type, age. This data will be stored in a database.
FR-3	User and donor dashboard	Display user requirements, requests for donor health certificate.
FR-4	Chatbot	A chat on the webpage to solve user queries and issues.
FR-5	Logout	Use logout option after completing job registration process.

4.2 NON -FUNCTIONAL REQUIREMENT

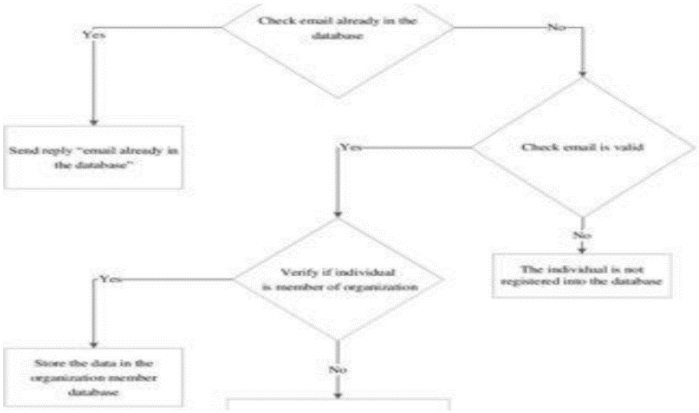
FR No.	Non-Functional Requirement	Description	
NFR-1	**Usability**	The webpage will be designed in such a way that any non-technical user can easily navigate through it and complete the plasma requirement registering work.	
NFR-2	**Security**	Using of python flask to cloud connect will provide security to the project. Database will be safely stored in DB2.	
NFR-3	**Reliability**	To make sure the webpage doesn't go down due to network traffic.	
NFR-4	**Performance**	Focus on loading the webpage as quickly as possible irrespective of the number of user/integrator traffic.	
NFR-5	**Availability**	The webpage will be available to all users (network connectivity is necessary) at any given point of time.	
NFR-6	**Scalability**	Increasing the storage space of database can increase the number of users. Add some features in future to make the webpage unique and attractive.	

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

Data Flow Diagrams:

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



5.2 USER STORIES

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

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, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register the app with Gmail login.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register & access the dashboard with Gmail Login	High	Sprint-1
	Dashboard	USN-6	As a user, I can search the blood group for which I need plasma.	I can get perfectly-matched plasma through filters.	High	Sprint-2
Customer (Web user)	Dashboard	USN-7	As a user, I can see login page and registration page for which the user logins and searches for the required blood group plasma.	I can login through Gmail and Facebook and register for my required blood group plasma.	Medium	Sprint-2
Customer Care Executive	Dashboard	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 see who are the persons registered here and their mail ids.	Medium	Sprint-4
	Dashboard	USN-10	As an Administrator, I can view how many members need what kind of blood group for plasma.	I can count the number of requirements.	Low	Sprint-4
ChatBot	Dashboard	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 that are related to our app.	Medium	Sprint-4

5.3 SOLUTIONS AND TECHNICAL ARCHITECTURE

Solution Architecture:

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

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

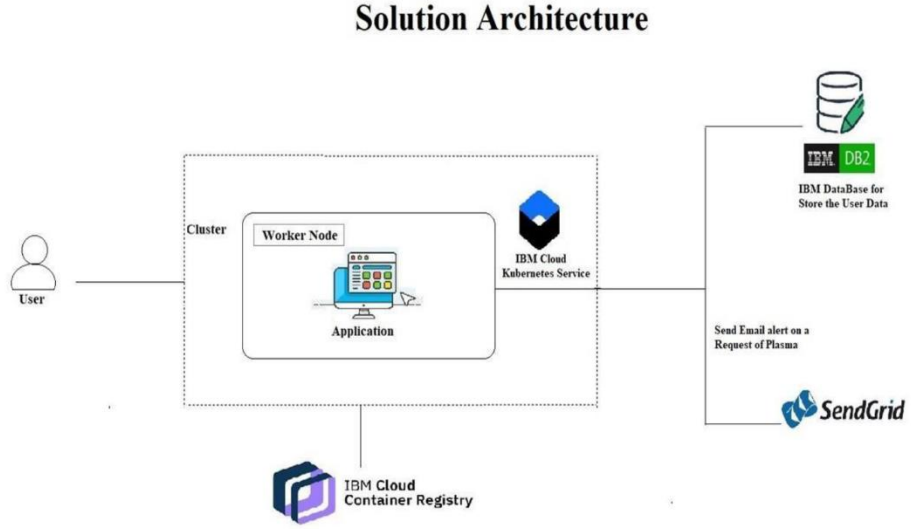


Figure 1: Architecture and data flow of the plasma donor application

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user can interact with our application with the help of Mobile app,Website.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	The user can login with our application, but he/she should registered earlier in our web app.	Java / Python

3.	Application Logic-2	They can also register with the help of website.	IBM Watson STT service
4.	Cloud Database	The user data will be stored and retrieved with help of this database.	IBM DB2, IBM Cloudant etc.
5.	File Storage	The user documents like photos,records and much more will be stored in cloud bucket,etc..	IBM Block Storage or Other Storage Service or Local Filesystem
6.	External API	With the help of API,the user can search the profile Based on their need.	IBM API, etc.
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud.	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description
1.	Is it scalable?	It follows highly scalable technologies that allows application to handle increase in large user data's and workload.
2.	Is it modifyable?	It is highly modifyable and maintenance requires low cost,compared to other application.
3.	Is the System Robust?	It does not disturb the performance of the computer by not affecting the operating system.It works in minimal hardware systems.

6.PROJECT PLANNING AND SCHEDULING

TITLE	DESCRIPTION	DATE
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.,	23rd September
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements.	9th September
Problem Solution Fit	Prepare problem – solution fit document.	28th September
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring to technical papers, research publications etc.,	3rd September
Solution Architecture	Prepare a solution architecture document.	26th September
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	12th September
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	12th October
Functional Requirement	Prepare the functional requirement document.	14th October
Technology Architecture	Prepare the technology architecture diagram	15th October
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project	21th October
Project Development - Delivery of Sprint-1,2,3 & 4	Develop & submit the developed code by testing it.	In Progress

CODE

```
from flask import Flask, render_template, flash, request,session
from flask import Flask, render_template, request, jsonify
import datetime
import re




import ibm_db
import pandas
import ibm_db_dbi
from sqlalchemy import create_engine

engine = create_engine('sqlite://',
                       echo = False)

dsn_hostname = "9938aec0-8105-433e-8bf9-
0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"
dsn_uid = "slg84898"
dsn_pwd = "sQLhssDgMcqDZ0uR"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "bludb"
dsn_port = "32459"
dsn_protocol = "TCPIP"
dsn_security = "SSL"

dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
```

```
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol,
dsn_uid, dsn_pwd,dsn_security)



try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)


except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )




app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'



@app.route("/")
def homepage():


    return render_template('index.html')



@app.route("/AdminLogin")
```

```
def AdminLogin():

    return render_template('AdminLogin.html')

@app.route("/Register")
def Register():

    return render_template('Register.html')
@app.route("/UserLogin")
def UserLogin():

    return render_template('UserLogin.html')

@app.route("/Donor")
def Donor():

    return render_template('Donor.html')



@app.route("/RNewUser", methods=['GET', 'POST'])
def RNewUser():
    if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
        address = request.form['address']
        pnumber = request.form['phone']
        uname = request.form['uname']
        password = request.form['psw']
```

```
        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" +
email + "','" + pnumber + "','" + password + "','" + uname + "','" + address + "')"
        insert_table = ibm_db.exec_immediate (conn, insertQuery)
        print(insert_table)




    return render_template('userlogin.html')


@app.route("/RNewDonor", methods=['GET', 'POST'])
def RNewDonor():
    if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        blood = request.form['bgrp']
        address = request.form['address']
        pnumber = request.form['phone']
        uname = request.form['uname']
        password = request.form['psw']

        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "INSERT INTO dotb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" + blood
+ "','" + pnumber + "','" + password + "','" + uname + "','" + address + "')"
```

```
        insert_table = ibm_db.exec_immediate (conn, insertQuery)
        print(insert_table)









    return render_template('userlogin.html')


@app.route("/Request")
def Request():


    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)


    selectQuery = "SELECT * from dotb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)


    dataframe.to_sql('Employee_Data',
                     con=engine,
                     if_exists='append')


    # run a sql query
    print(engine.execute("SELECT * FROM Employee_Data").fetchall())


    return render_template('ViewProduct.html', data=engine.execute("SELECT * FROM
Employee_Data").fetchall())
```

```
@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():
    error = None
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['uname'] = request.form['uname']



        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)


        selectQuery = "SELECT * from regtb where uname='" + username + "' and password='" +
password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)


        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:
            print("Login")
            selectQuery = "SELECT * from regtb where uname='" + username + "' and password='" +
password + "'"
            dataframe = pandas.read_sql(selectQuery, pd_conn)




            dataframe.to_sql('Employee_Data',
                    con=engine,
                    if_exists='append')
```

```
        # run a sql query
        print(engine.execute("SELECT * FROM Employee_Data").fetchall())

        return render_template('UserHome.html', data=engine.execute("SELECT * FROM
Employee_Data").fetchall())




def main():
    app.run(debug=True, use_reloader=True)


if __name__ == '__main__':
    main()
```

ADVANTAGES AND DISADVANTAGES

1.user friendly

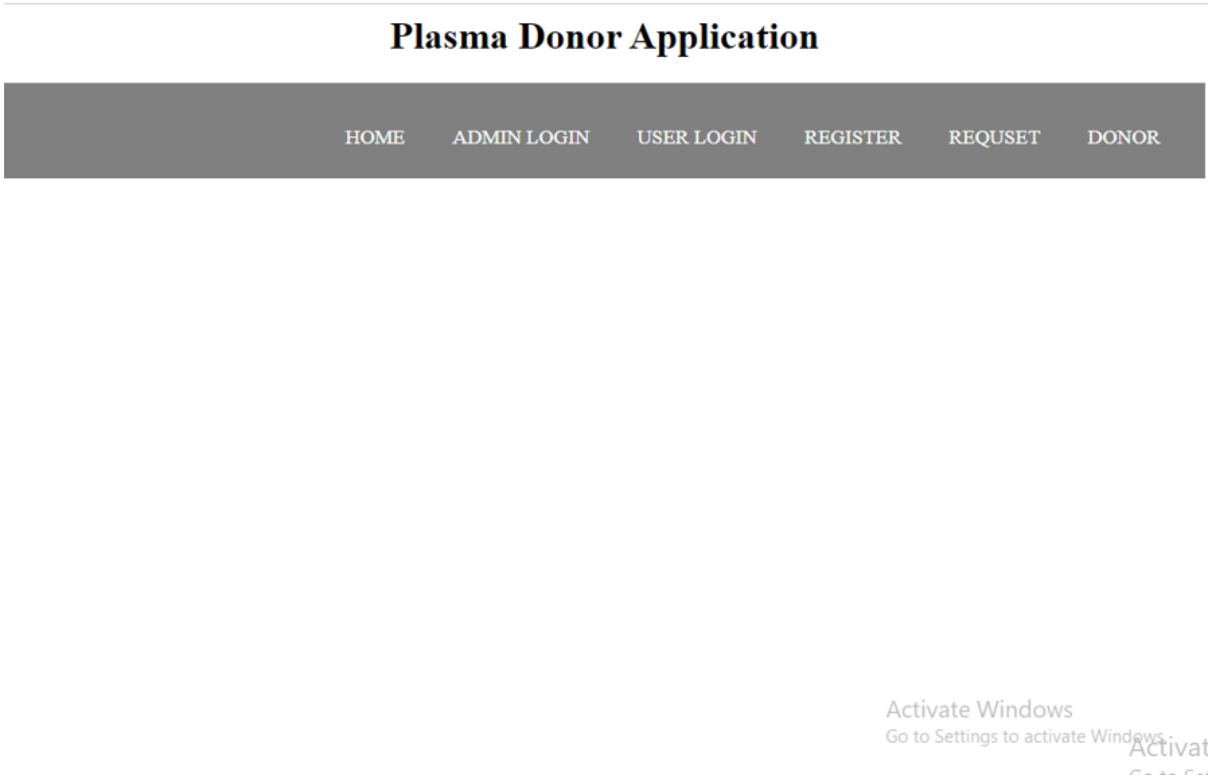
2. Identify and formulate the problem

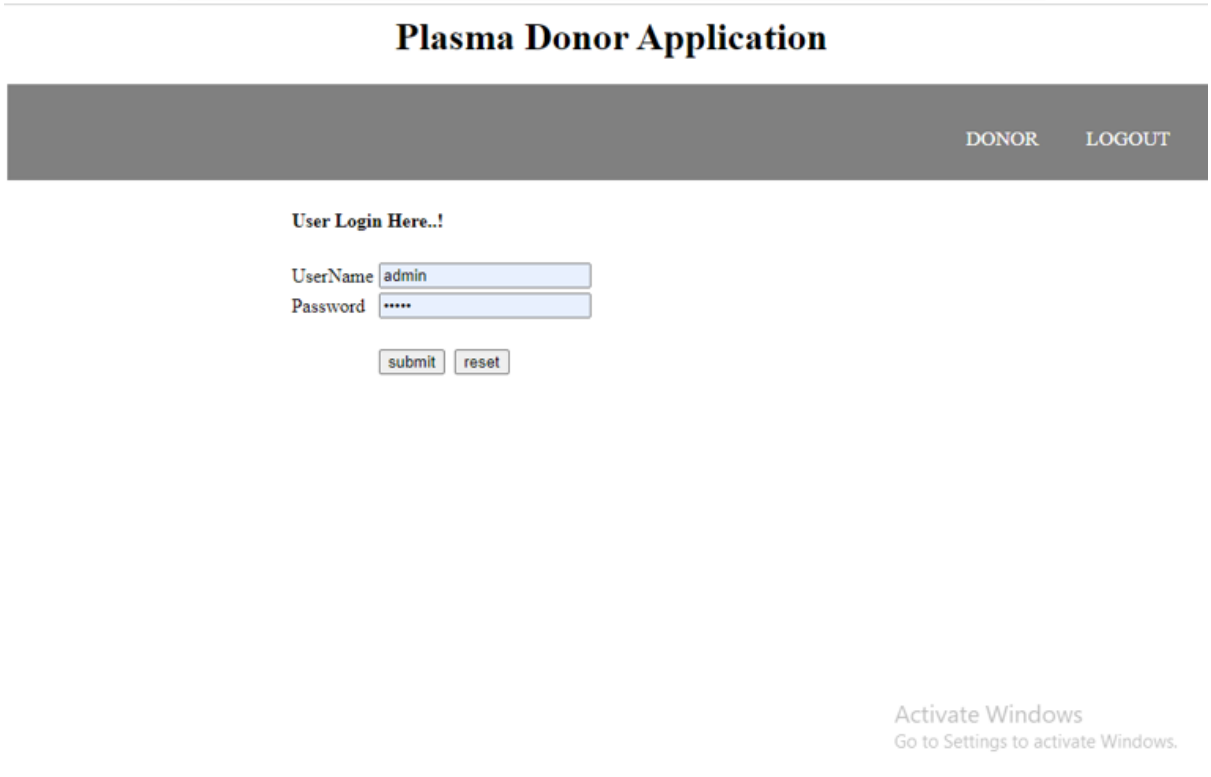
3. Identify the requirements, objective, and preferences

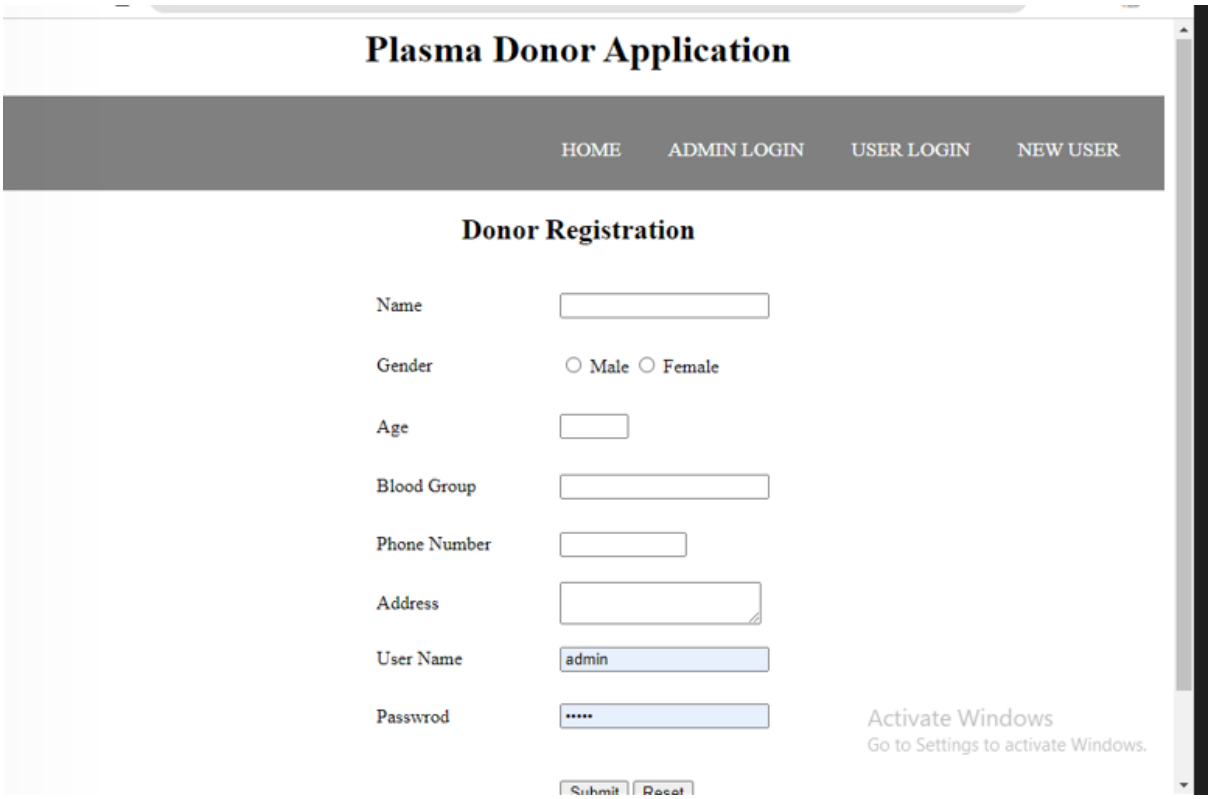
4.Determine project plan

5. Determine the feasibility of plan

RESULTS


Plasma Donor Application
HOME
ADMIN LOGIN
USER LOGIN
REGISTER
REQUSET
DONOR
Activate Windows
Go to Settings to activate Windows.


Plasma Donor Application
DONOR
LOGOUT
User Login Here..!
UserName admin
Password
submit
reset
Activate Windows
Go to Settings to activate Windows.


Plasma Donor Application
HOME
ADMIN LOGIN
USER LOGIN
NEW USER
Donor Registration
Name
Gender
Male
Female
Age
Blood Group
Phone Number
Address
User Name
admin
Passwrod
Activate Windows
Go to Settings to activate Windows.

CONCLUSION

We design the plasma donar project for web based application using flask technology Plasma is the most essential thing to save a life. By donating blood, we can save many lives. It is also important to remember that any one of us may need blood at some point in our lives, making blood donation is an essential duty of our citizenry. In today's world where people are busy with their lifestyle and those who are eager to donate blood but are not able to, can plan to donate blood by sitting at home just by one click with our application This application will make revolutionary changes to the medical system as people will be able to donate blood and serve mankind. It can also help people to know about the benefits about blood donation and that their small contribution can help one person to save his/her lives as soon as possible in a quick and well managed manner.

Project Demo Video : https://youtu.be/-G4rVYUOs0E

GitHub Link : https://github.com/IBM-EPBL/IBM-Project-53721-1661490331