**PLASMA DONAR MANAGEMENT**

**INTRODUCTION**

* 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. **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 organisation, 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 donars 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 scanrequestforbloodbycallinginor 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

* 1. **REFERENCES**

[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

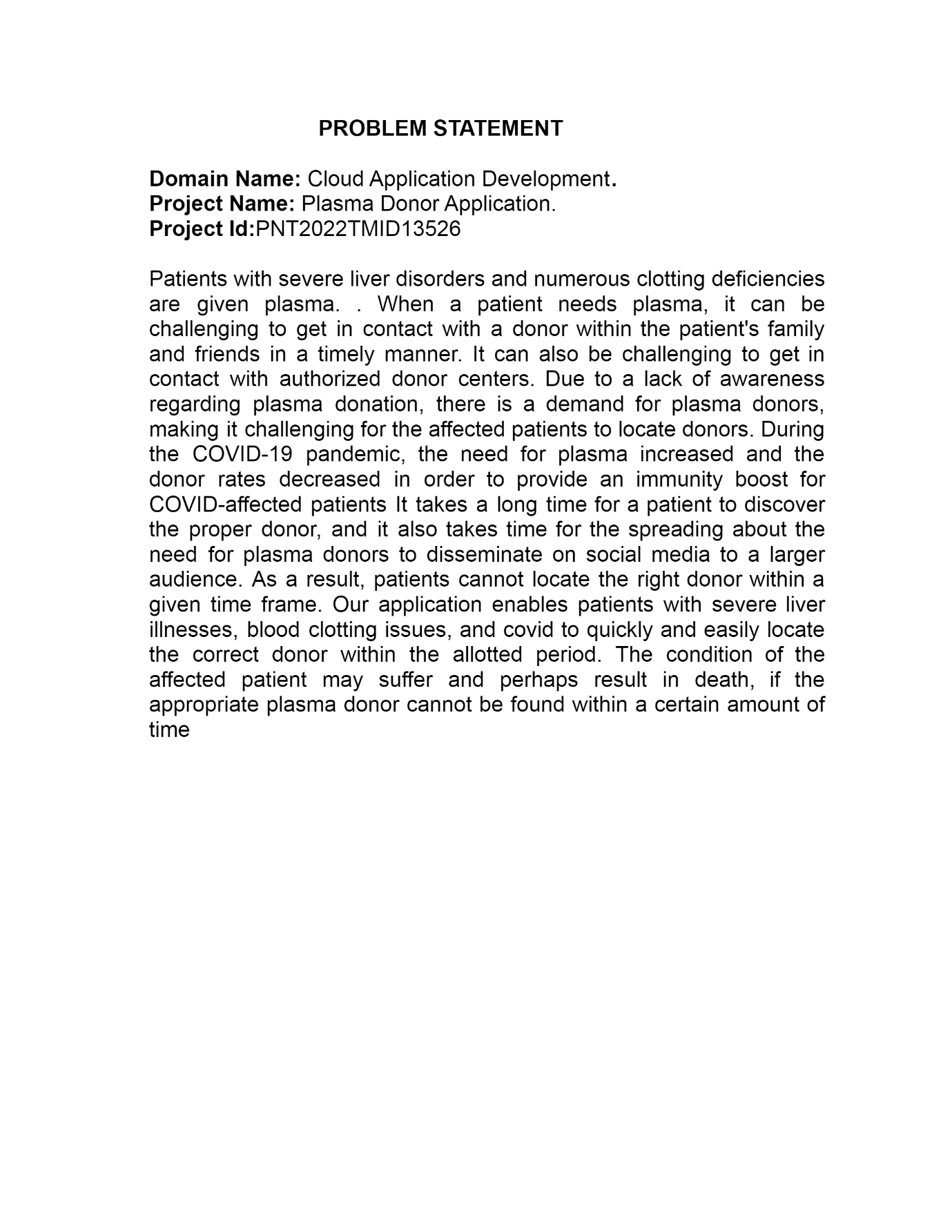
[7] Bagot KL, Bove LL, Masser BM, Bednall TC, Buzza M. Perceived deterrents to being a plasmapheresis donor in a voluntary, nonremunerated environment. Transfusion (Paris) 2013;53:1108–19

[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,

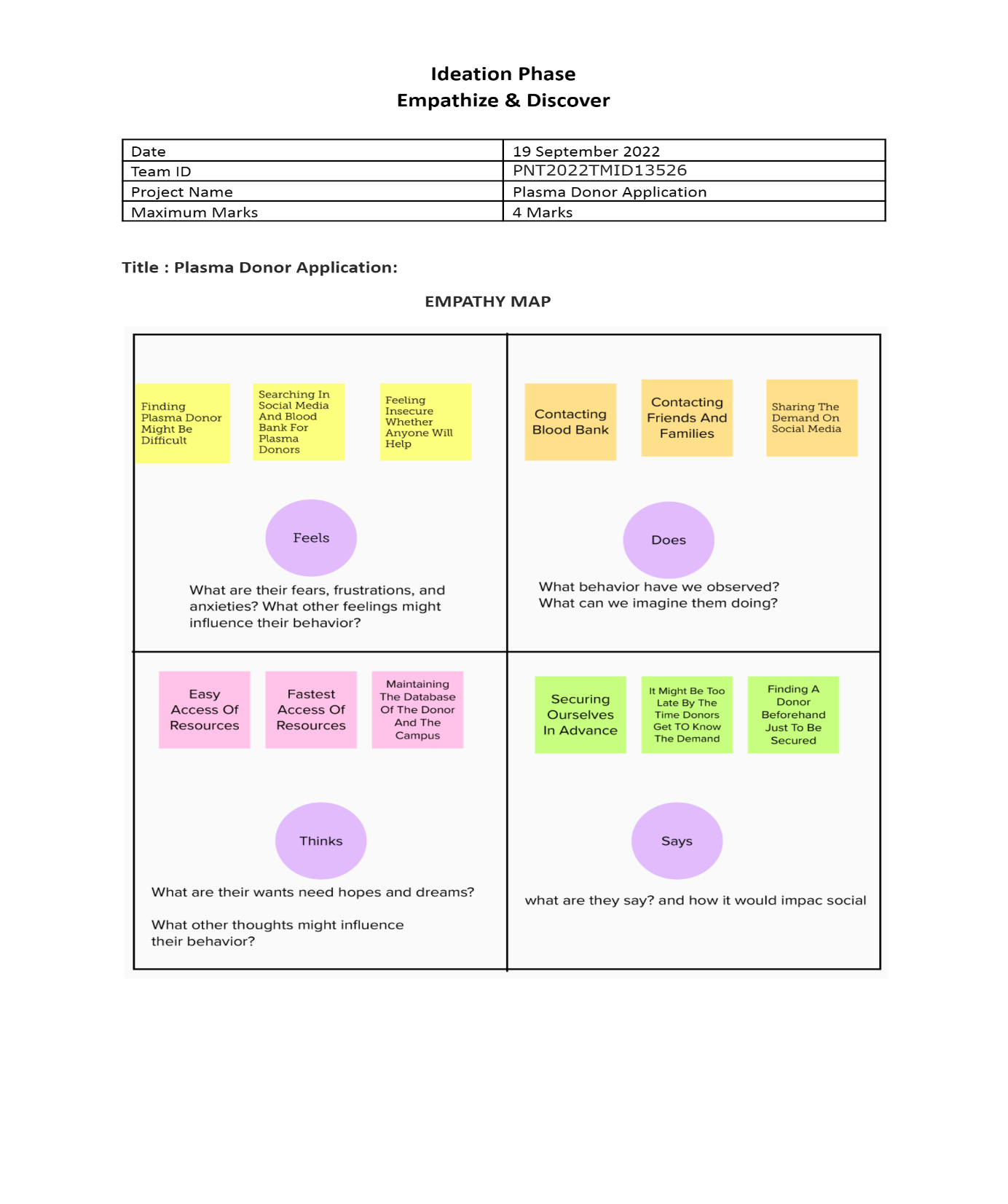
[9] Bove LL, Bednall T, Masser B, Buzza M. Understanding the plasmapheresis donor in a voluntary, nonremunerated environment. Transfusion (Paris) 2011;51:2411–24

[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,

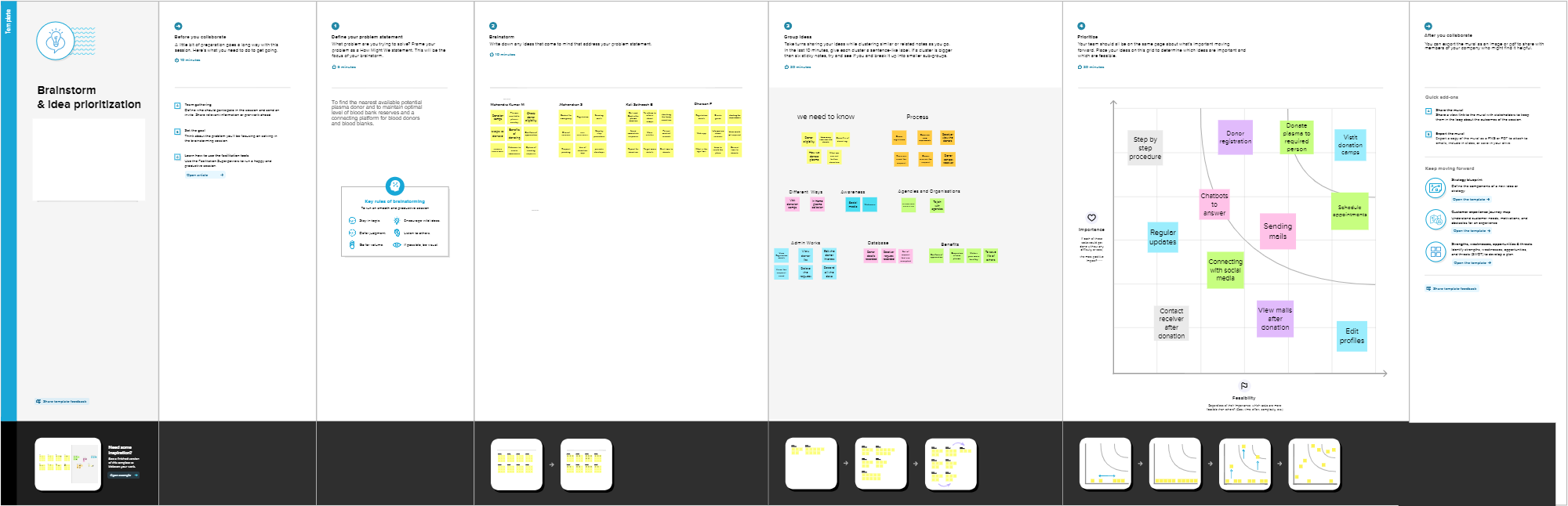
* 1. **PROBLEM STATEMENT DEFINITION**

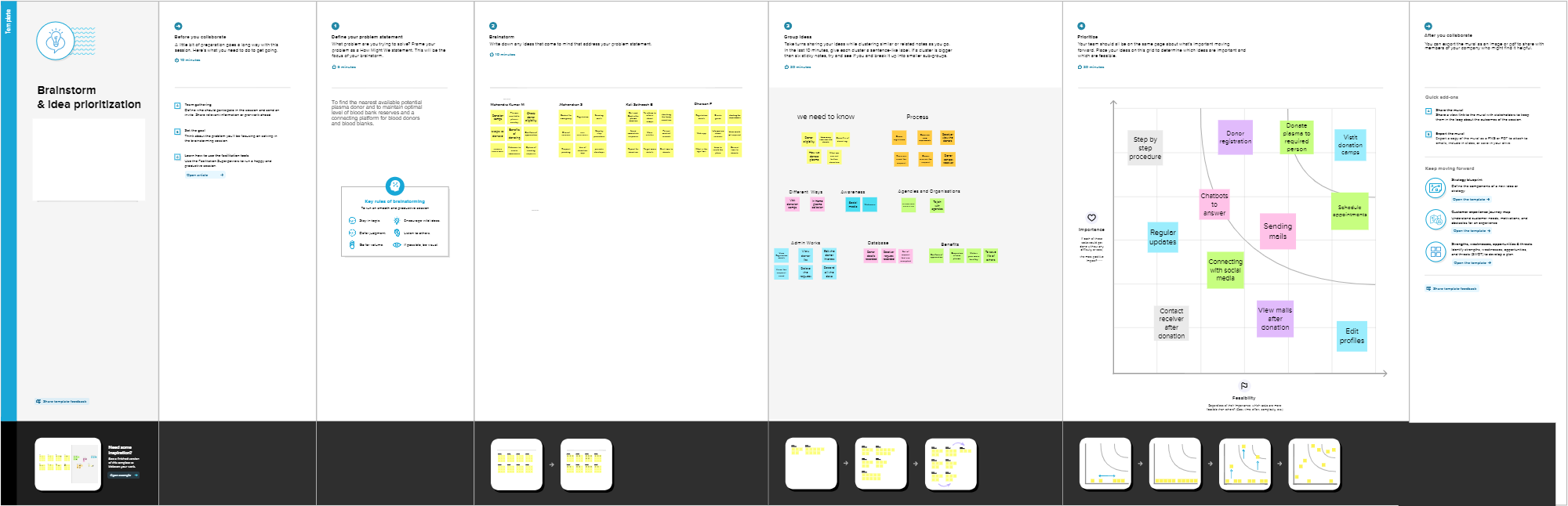
**3. IDEATION AND PROPOSED SOLUTION**

**3.1 EMPATHY MAP CANVAS**



**3.2 IDEATION AND BRAIN STORMING**

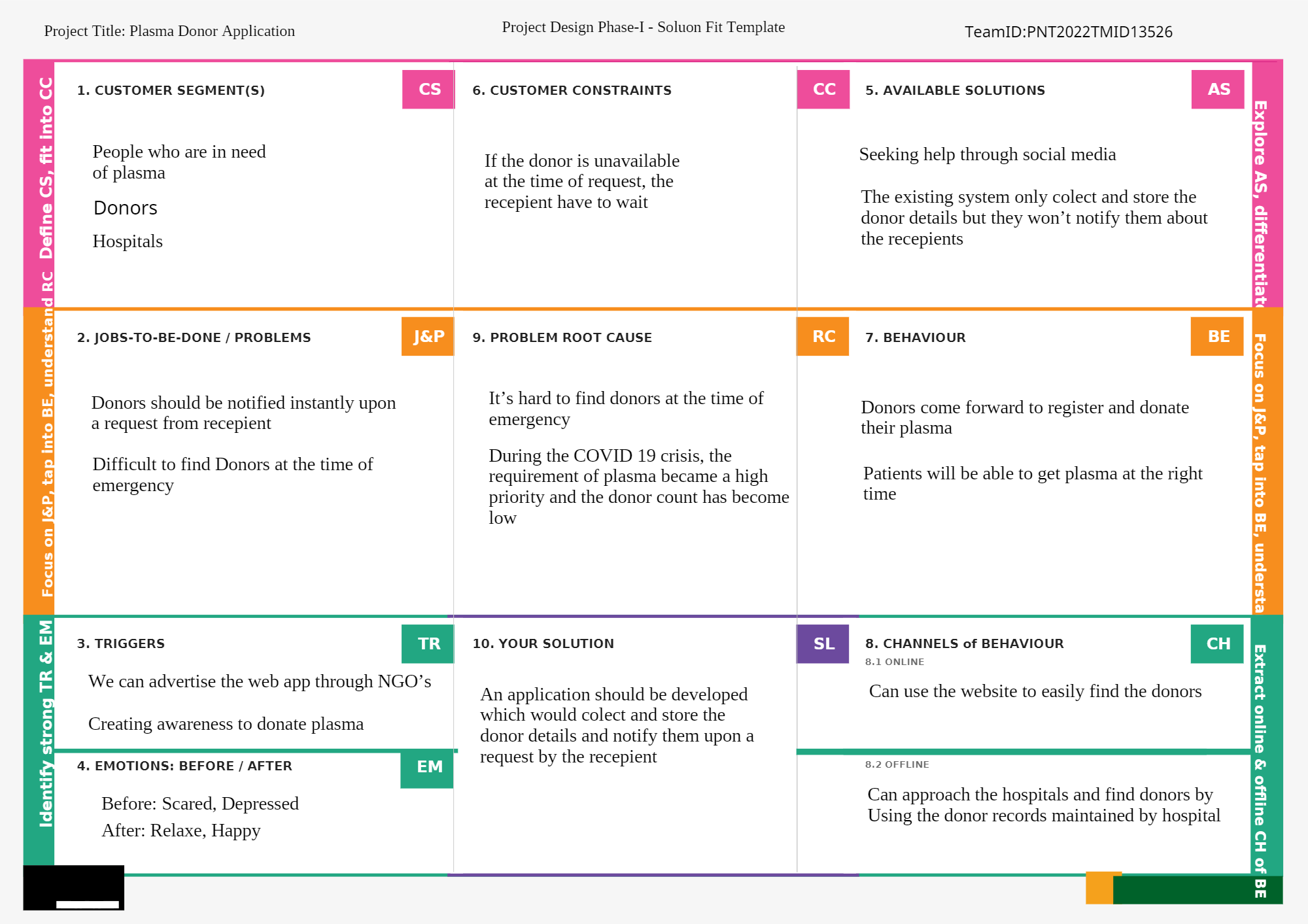




**3.3 PROPOSED SOLUTIONS**

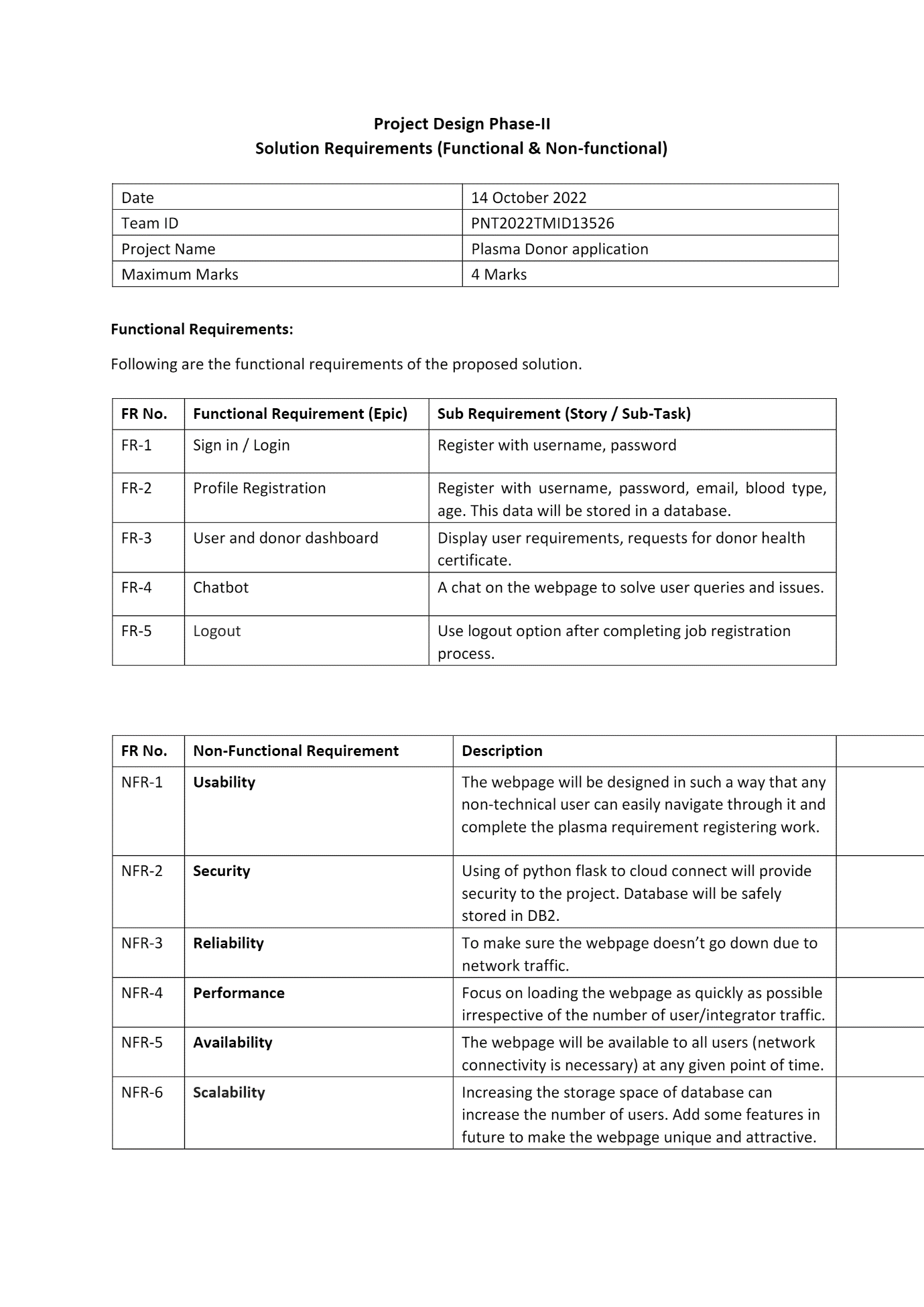
To create an application for people who want to donate their plasma for the people who need it mostly in times of emergency. The application will enable people to register themselves in the portal for donating their plasma and the recipients who need it can see their details so that they can get the 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**

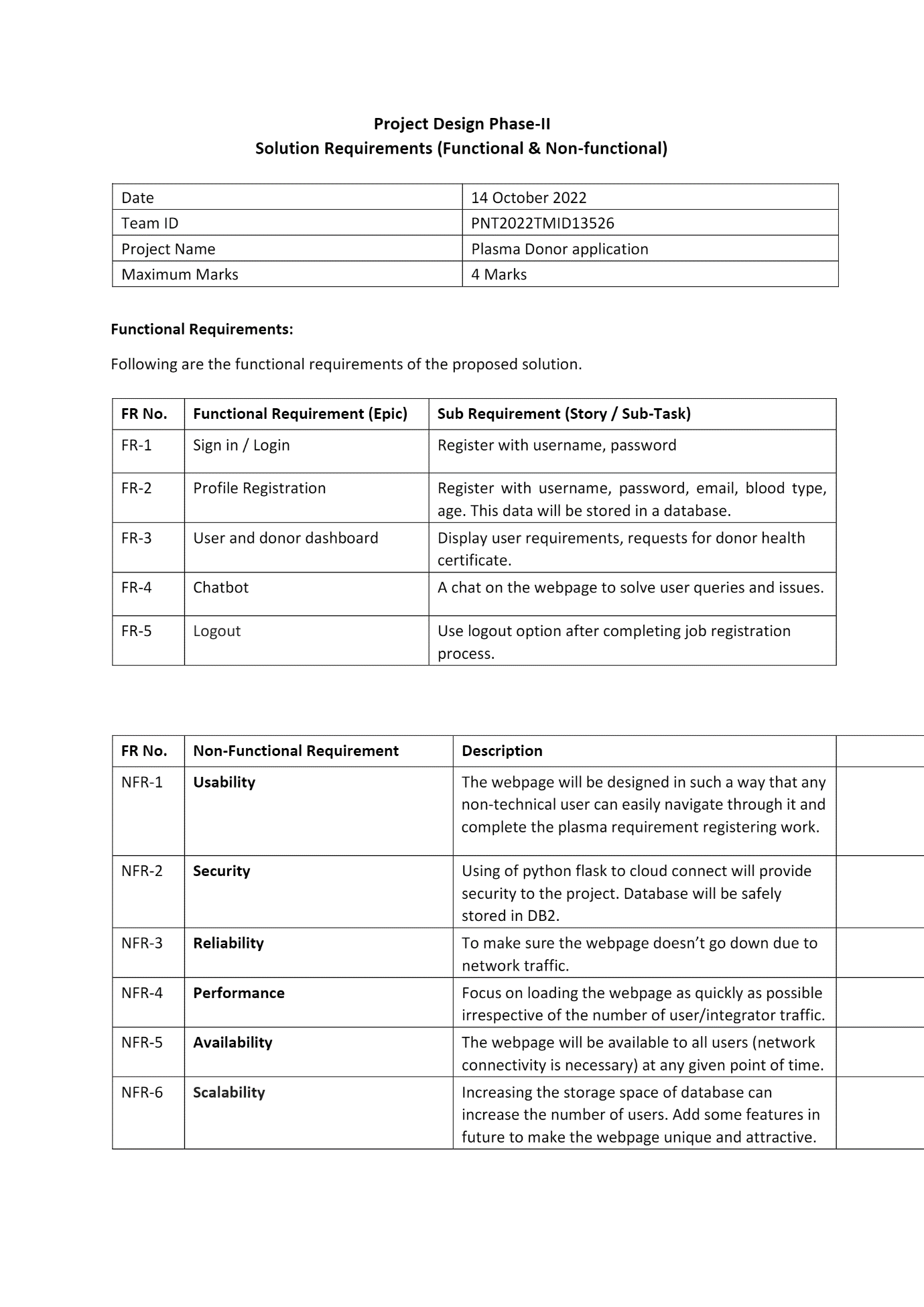
****

**4.REQUIREMENT ANALYSIS**

**4.1 FUNCTIONAL REQUIREMENT**

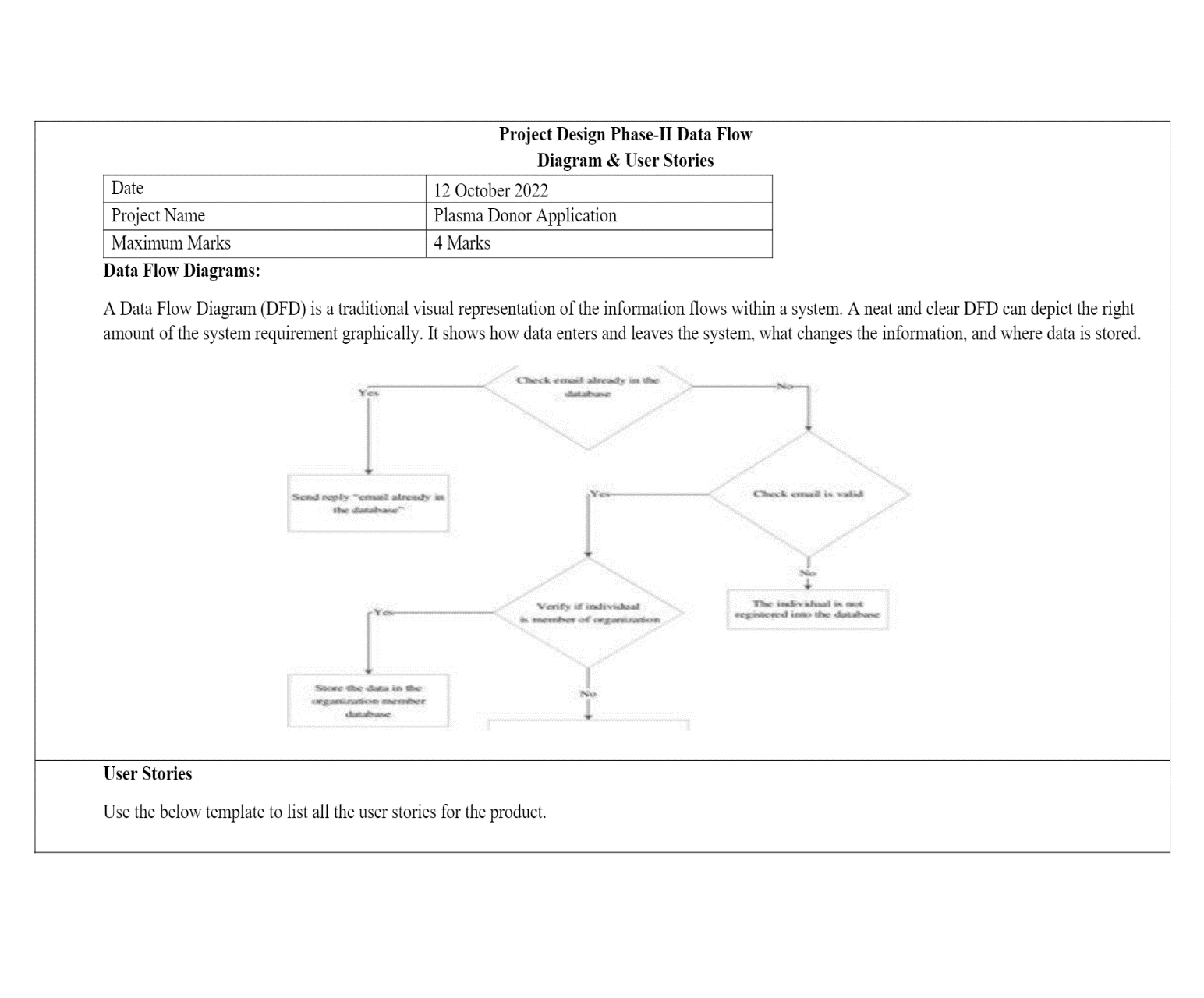
****

**4.2 NON -FUNCTIONAL REQUIREMENT**

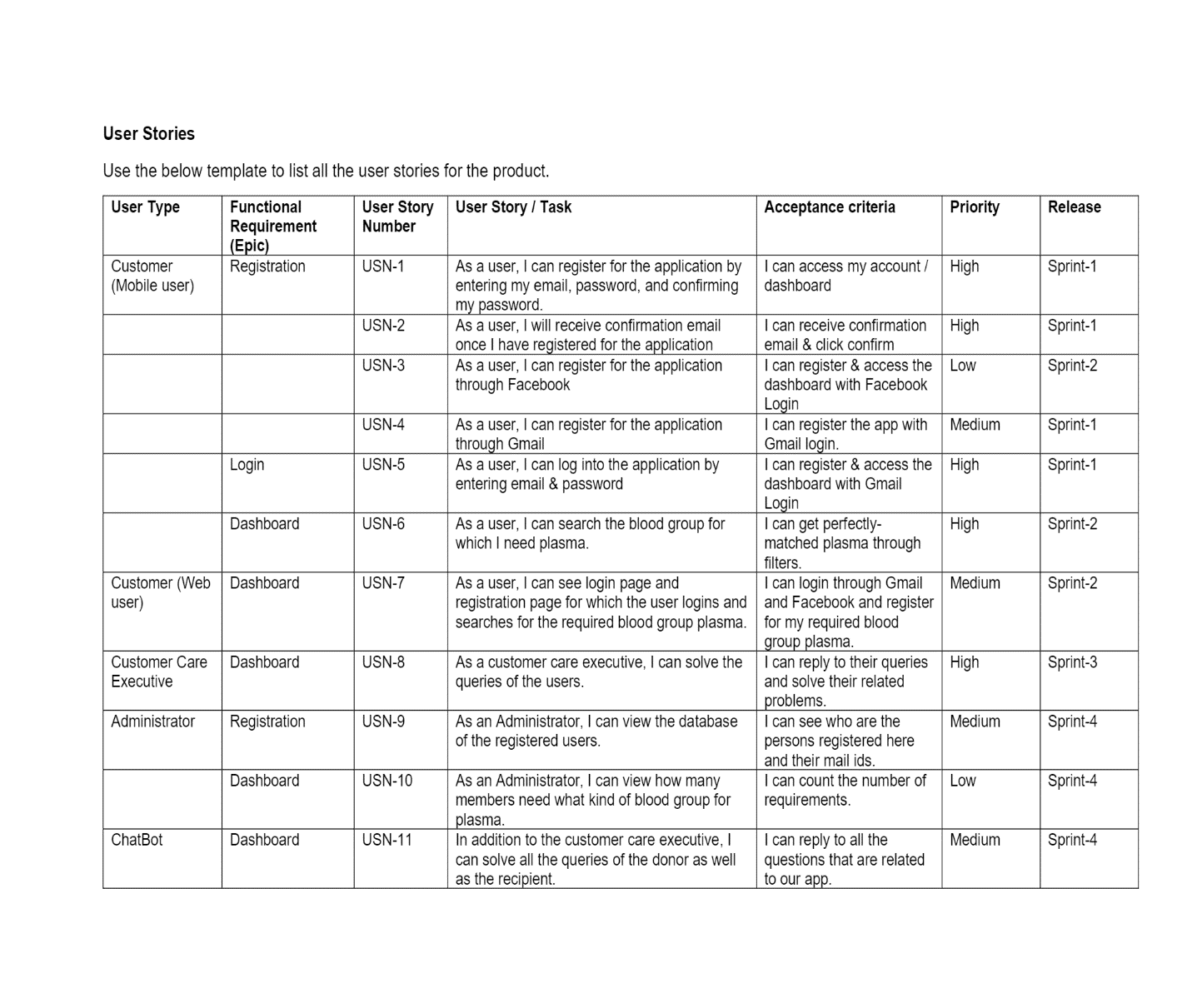
****

**5.PROJECT DESIGN**

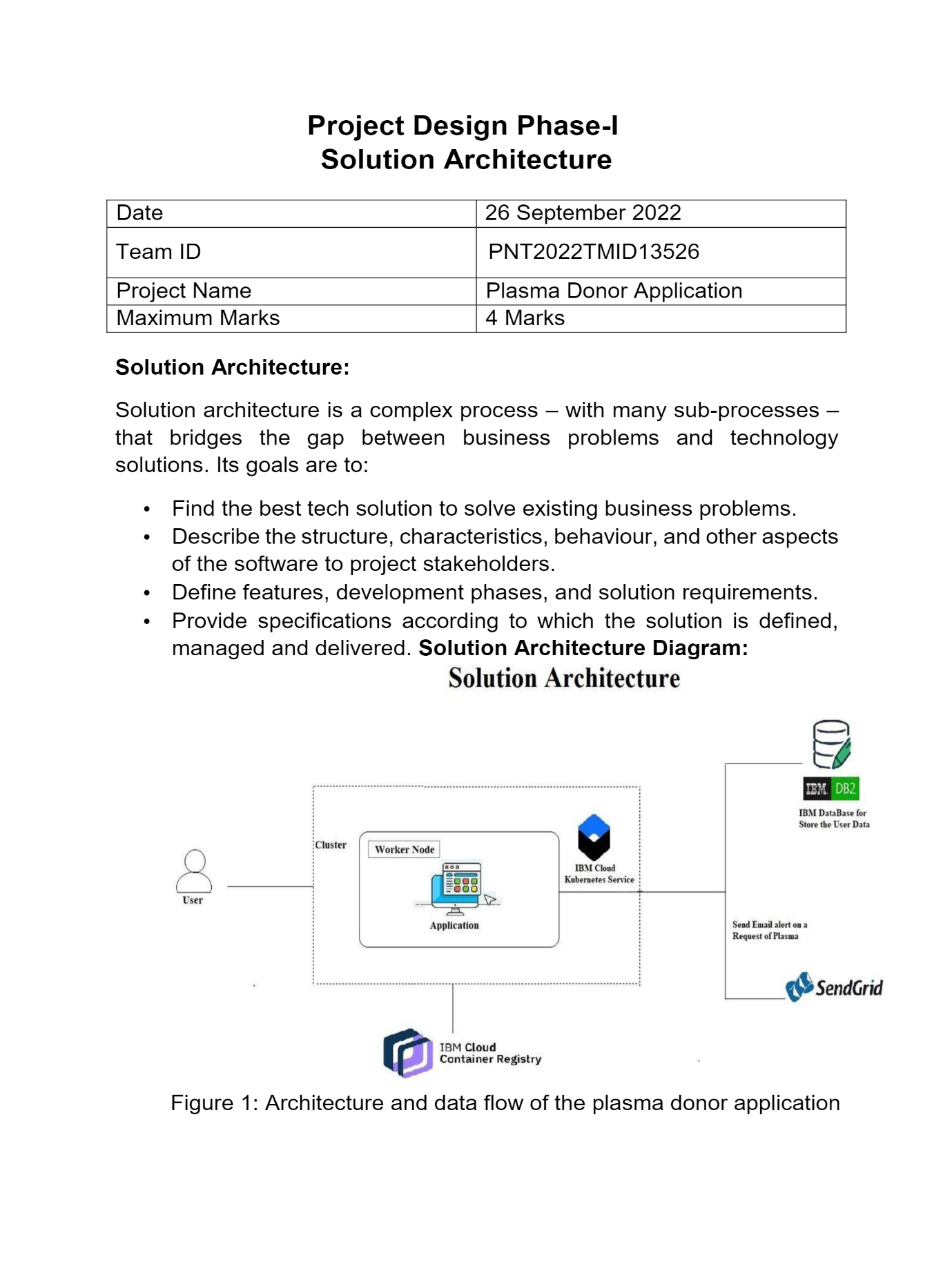
**5.1 DATA FLOW DIAGRAMS**

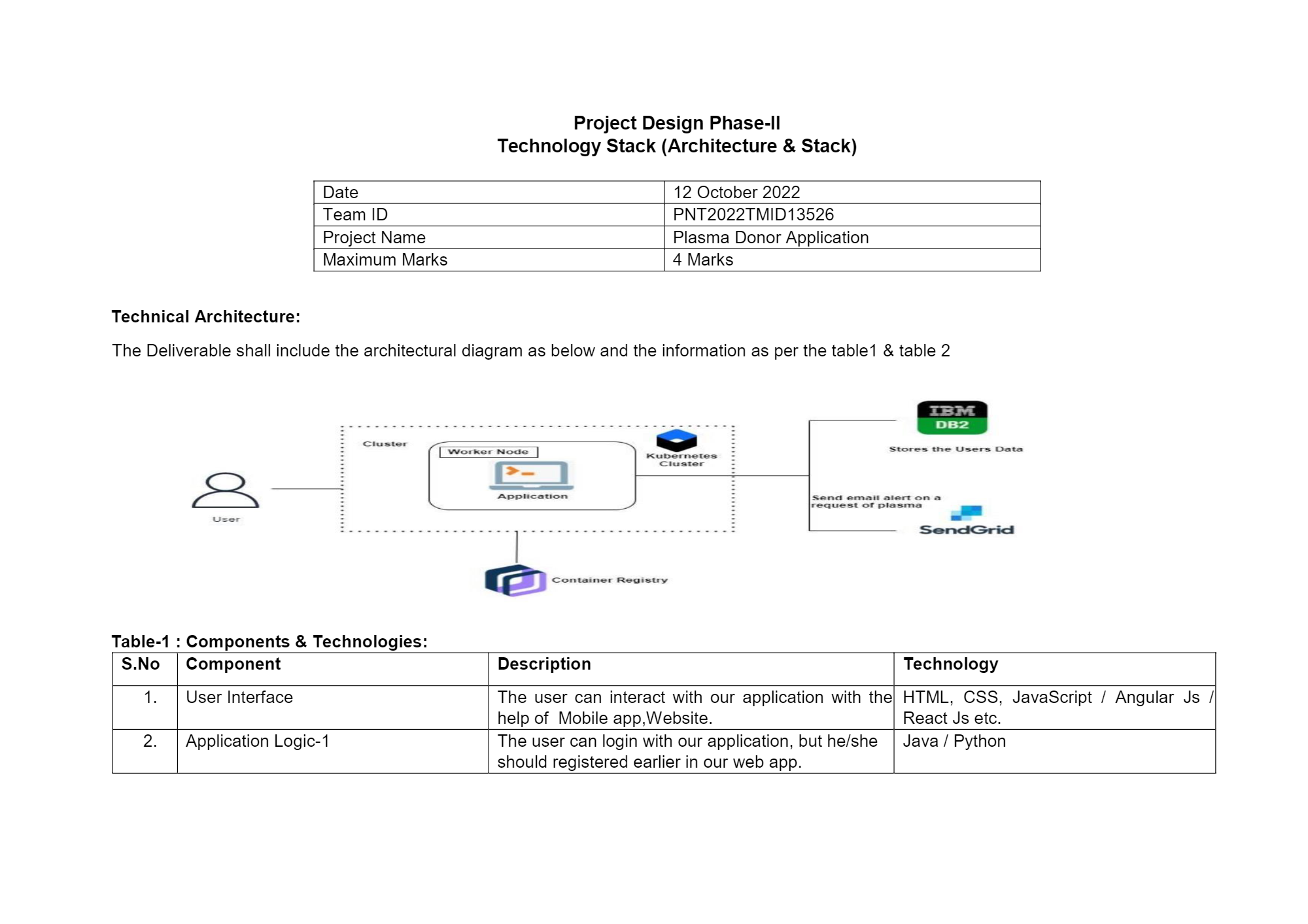
****

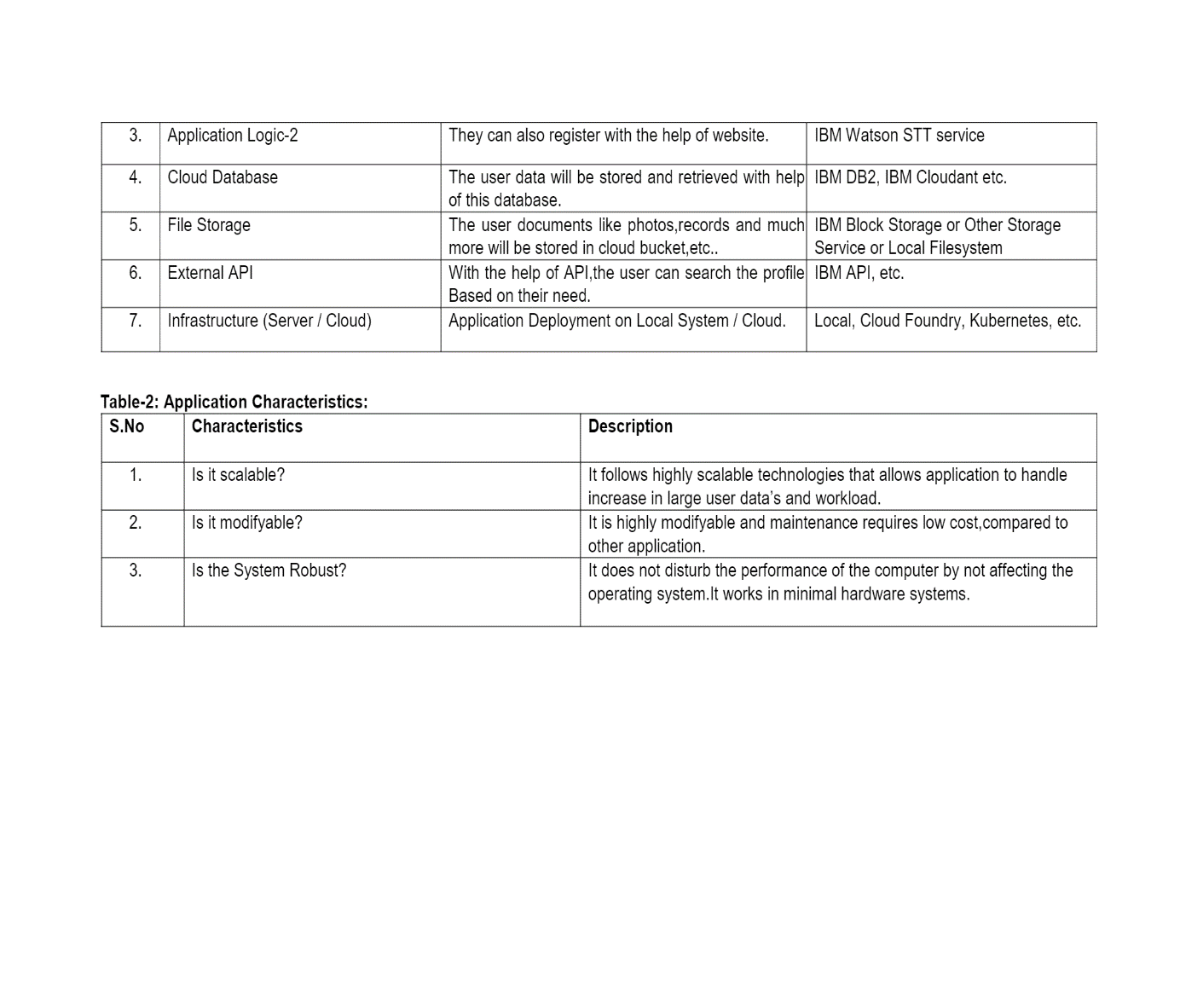
**5.2 USER STORIES**

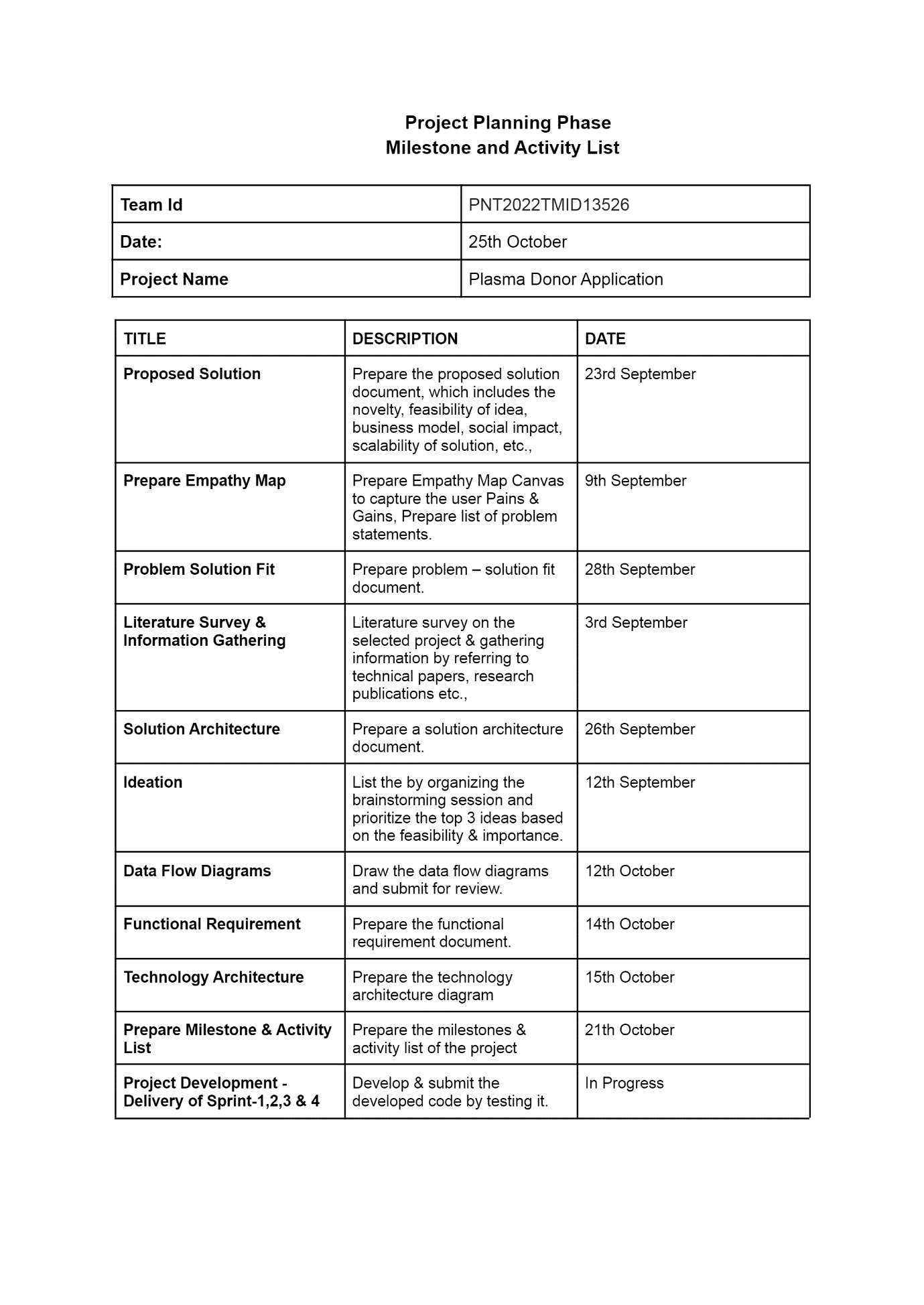
****

**5.3 SOLUTIONS AND TECHNICAL ARCHITECTURE**





**6.PROJECT PLANNING AND SCHEDULING**



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

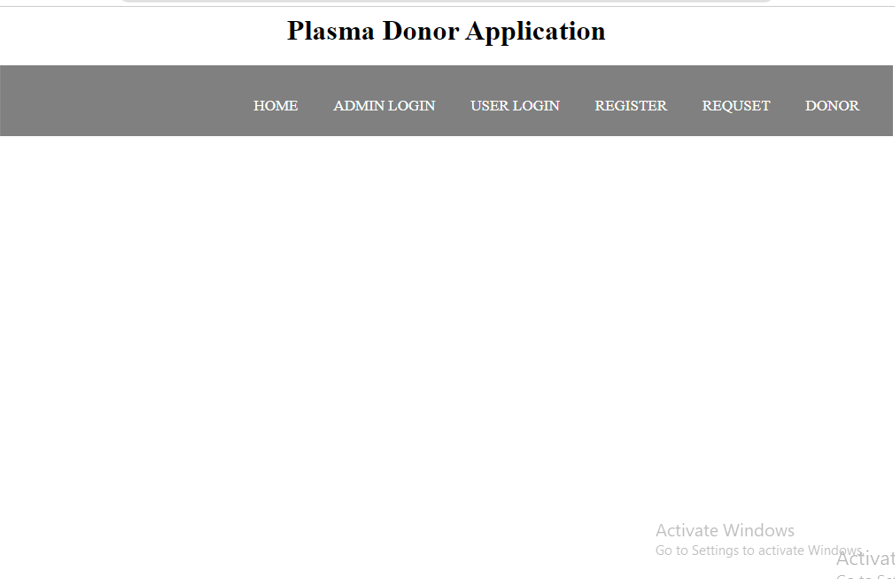
2. Identify and formulate the problem

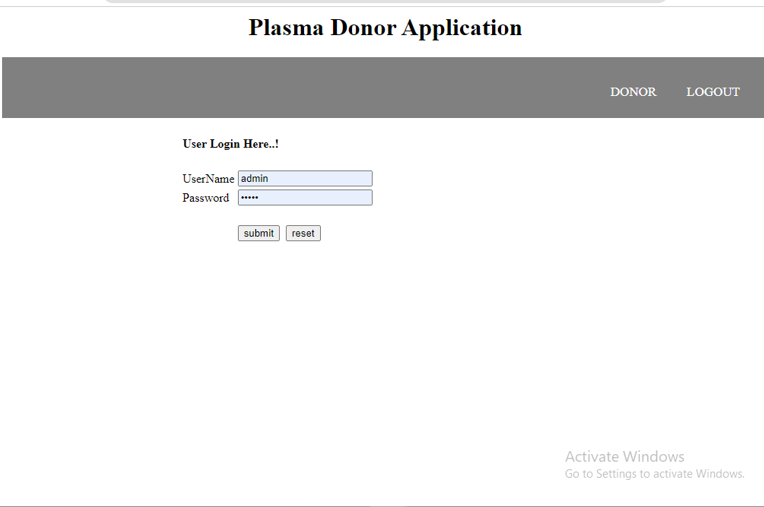
3. Identify the requirements, objective, and preferences

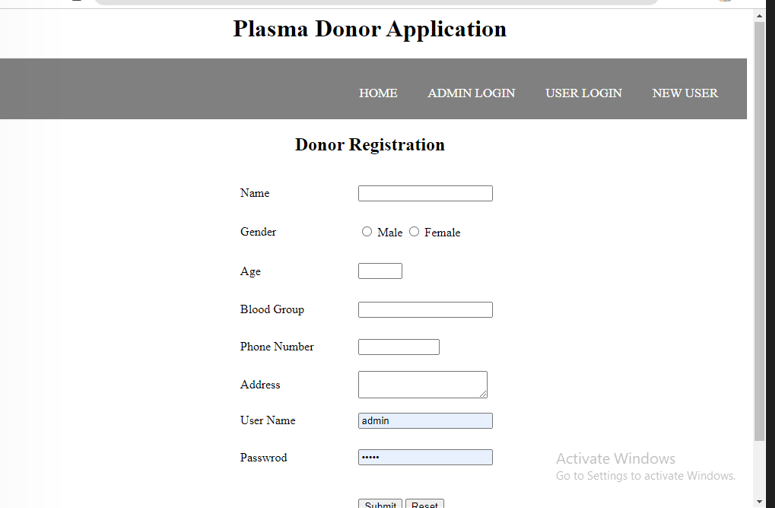
4.Determine project plan

5. Determine the feasibility of plan

**RESULTS**







**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.