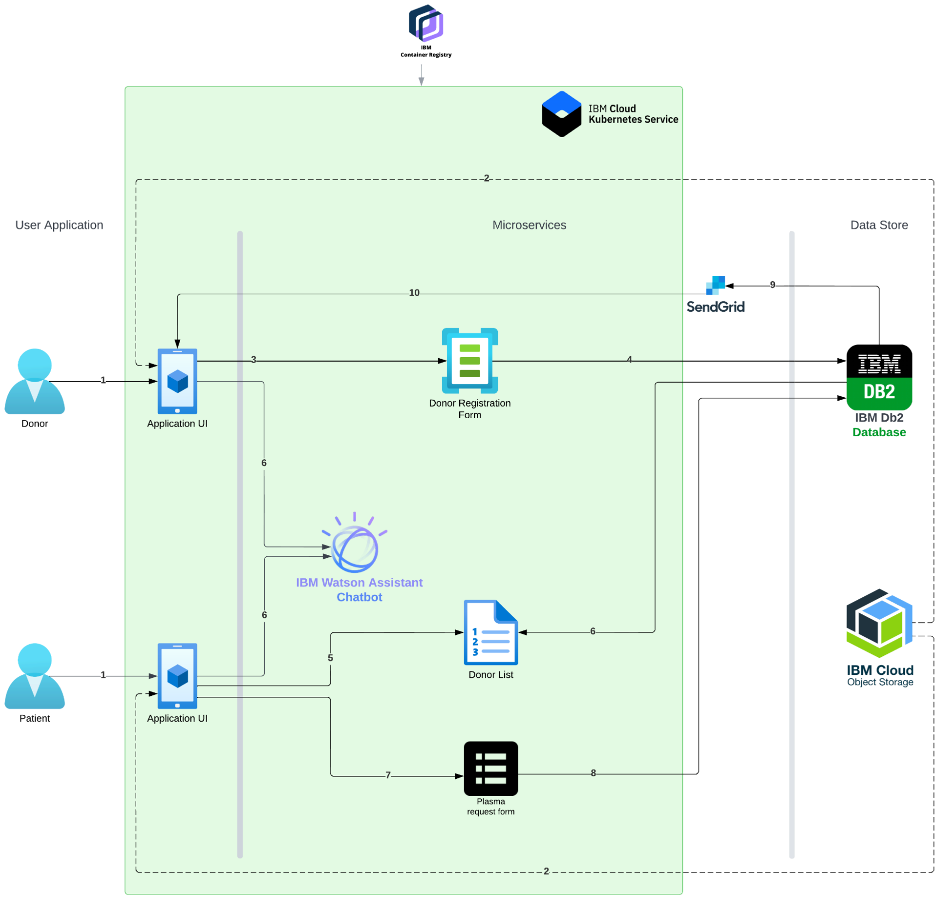
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	22 October 2022
Team ID	PNT2022TMID06713
Project Name	Plasma Donor Application
Maximum Marks	4 Marks

Technical Architecture:



Steps

1. Users open the web application on their devices
2. The required assets for the application are instantly loaded from the IBM Cloud Object Storage
3. The donor registers his/her medical and contact information
4. These details are stored in the IBM Db2 Database
5. The patient accesses the list of donors available
6. The donor list is updated real-time from the database
7. The patient then submits a plasma donation request
8. The request details are stored in the database
9. The SendGrid API is provided with the required request details
10. The SendGrid API sends an email alert to the donor's device

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application (Web UI, Chatbot etc.)	HTML, CSS and JavaScript
2.	User Registration	Sign up users to the web application using Email and Password and store credentials in database	IBM Db2 and Python Flask
3.	User login	Authorize users by comparing credentials with the registered accounts in the database	IBM Db2 and Python Flask
4.	Resolve user queries	Allow users to resolve their queries using the chatbot	IBM Watson Assistant
5.	Database	The IBM Db2 is a relational and fully managed SQL database	RDBMS, SQL
6.	Cloud Database	Database Service on Cloud to store application data	IBM Db2
7.	File Storage	The files required for the application are stored in the cloud object storage for fast retrieval	IBM Cloud Object Storage
8.	Email Notification	Send immediate email alerts to donors upon a request from a patient	SendGrid API
9.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	IBM Container Registry, IBM Cloud Kubernetes Cluster

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The application is primarily developed in Flask and containerized as Docker images	Python Flask and Docker
2.	Security Implementations	The flask app is protected in various to prevent attacks and only registered users can access the data in the database	IBM Db2, HSTS, CSP, XSS Protection, TLS
3.	Scalable Architecture	The services provided by IBM Cloud and Kubernetes can handle huge amount of users seamlessly and are scalable to a large extent	IBM Cloud Platform and Kubernetes
4.	Availability	The application can handle any number of users and is available all the time	Docker and Kubernetes
5.	Performance	The IBM Cloud services ensure fast data delivery and Kubernetes ensures extremely fast performance of the containerized application	IBM Cloud Object Storage, IBM Db2, IBM Container Registry, IBM Cloud Kubernetes Cluster