**CASE STUDY: Online Blood Bank**

**Scenario:**

**Problem Statement:** Any person who needs blood of a particular blood group has to search blood donors in offline blood banks like hospitals and etc. Also the blood donor has to visit some blood bank to get registered as a donor which is obviously a hectic task for donor as well as patient who needs blood. Also the particular patient can contact only his/her area’s donor only from the blood bank. Donor or patient might feel a hassle-free platform to work

**Proposed System**: Blood Bank Management System (BBMS) is a browser and in future may become mobile based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank.

This project aims at maintaining all the information pertaining to blood donors, different blood groups available in blood bank and help them manage in a better way.

Aim is to provide transparency make the process of obtaining blood from a blood bank hassle free and make the system of blood bank management effective.

**Use Cases:**

**End User: Donor and Client**

1. Donor can register to donate blood with all the required details
2. Clients can view all the donors available.
3. Clients can view donors of a Specific Group
4. Clients can view donors of Specific Group who are eligible to donate, eligible donor are those who have donated blood six months before the current date
5. Clients can also search donors according to location
6. Donor can update their info.
7. Donor can un register when he/she not ready to donate blood.
8. Donors can view the Donation Camps being organized in their Area
9. Donors can register themselves for the Donation Camps
10. Client can view list of Donors for a Particular Camp

General Requirements

* Most services use REST request/response-based inter-process communication mechanism and JSON as the message format.
* Client applications (browser or smartphone) don’t send requests to the services directly but only through API Gateway.
* Client Application is Built with Spring MVC Using Controllers, jsp and HTML
* Eureka is used as service registry and it has the details of available service instances.
* Hystrix is used enable fault tolerance in the application when underlying service is down or throwing error permanently.
* Need to fallback to different path of program execution automatically.
* Donor and Donation Camp Details can be stored in a MySql Database

**Software Resources Required:**

Miroservice – Spring boot

Java Rest API services

MYSQL database

WEB UI – Spring MVC

My SQL

My SQL

Donor Service

Donation Camp Service

API GATEWAY

Eureka Registry