# BLOOD BANK MANAGEMENT SYSTEM

**PROJECT REPORT**

**18CSC202J/ 18AIC203J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY**

**(2018 Regulation)**

**II Year/ III Semester**

**Academic Year: 2022 -2023**

By

**ADITYA MISHRA (RA2111003011817)**

**SHIVAM KUMAR SINGH (RA2111003011798)**

Under the guidance of

**Dr . P. MURALI**

**Associate Professor**

**Department of Computing Technologies**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SCHOOL OF COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Kancheepuram**

**NOVEMBER 2022**

## BONAFIDE

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY project report** titled "**BLOOD BANK MANAGEMENT SYSTEM"** is the bonafide work of **ADITYA MISHRA (RA2111003011817) SHIVAM KUMAR SINGH (RA2111003011798)** who undertook the task of completing the project within the allotted time.

| | |
|---|---|
| **Signature of the Guide** | **Signature of the II Year Academic Advisor** |
| Dr. P. Murali | ---------------------------------------- |
| **Associate Professor** | **Professor and Head** |
| Department of CTECH, | Department of CTECH |
| SRM Institute of Science and Technology | SRM Institute of Science and Technology |

**About the course:-**

18CSC202J/ 8AIC203J - Object Oriented Design and Programming are 4 credit courses with **L T P C as 3-0-2-4** (Tutorial modified as Practical from 2018 Curriculum onwards)

**Objectives:**

The student should be made to:

- Learn the basics of OOP concepts in C++
- Learn the basics of OOP analysis and design skills.
- Be exposed to the UML design diagrams.
- Be familiar with the various testing techniques

**Course Learning Rationale (CLR): The purpose of learning this course is to:**

1. Utilize class and build domain model for real-time programs
2. Utilize method overloading and operator overloading for real-time application development programs
3. Utilize inline, friend and virtual functions and create application development programs
4. Utilize exceptional handling and collections for real-time object-oriented programming applications
5. Construct UML component diagram and deployment diagram for design of applications
6. Create programs using object-oriented approach and design methodologies for real-time application development

**Course Learning Outcomes (CLO): At the end of this course, learners will be able to:**

1. Identify the class and build domain model
2. Construct programs using method overloading and operator overloading
3. Create programs using inline, friend and virtual functions, construct programs using standard templates
4. Construct programs using exceptional handling and collections
5. Create UML component diagram and deployment diagram
6. Create programs using object oriented approach and design methodologies

**Table 1: Rubrics for Laboratory Exercises**

(Internal Mark Splitup:- As per Curriculum)

| | | |
|---|---|---|
| **CLAP-1** | 5=(2(E-lab Completion) + 2(Simple Exercises)( from CodeZinger, and any other coding platform) + 1(HackerRank/Code chef/LeetCode Weekend Challenge) | Elab test |
| **CLAP-2** | 7.5=(2.0(E-lab Completion)+ 2.0 (Simple Exercises)( from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge) | Elab test |
| **CLAP-3** | 7.5=(2.0(E-lab Completion(80 Pgms)+ 2.0 (Simborple Exercises)( from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge) | **2 Mark -** E-lab Completion **80 Program** Completion from 10 Session (Each session min 8 program)<br>**2 Mark -** Code to UML conversion GCR Exercises<br>**3.5 Mark - Hacker Rank** Coding challenge completion |
| **CLAP-4** | 5= 3 ( Model Practical) + 2( Oral Viva) | • **3 Mark** – Model Test<br>• **2 Mark** – Oral Viva |
| **Total** | 25 | |

## COURSE ASSESSMENT PLAN FOR OODP LAB

| S.No | List of Experiments | Course Learning Outcomes (CLO) | Blooms Level | PI | No of Programs in each session |
|---|---|---|---|---|---|
| 1. | Implementation of I/O Operations in C++ | CLO-1 | Understand | 2.8.1 | 10 |
| 2. | Implementation of Classes and Objects in C++ | CLO-1 | Apply | 2.6.1 | 10 |
| 3, | To develop a problem statement.<br>1. From the problem statement, Identify Use Cases and develop the Use Case model.<br>2. From the problem statement, Identify the conceptual classes and develop a domain model with a UML Class diagram. | CLO-1 | Analysis | 4.6.1 | Mini Project Given |
| 4. | Implementation of Constructor Overloading and Method Overloading in C++ | CLO-2 | Apply | 2.6.1 | 10 |
| 5. | Implementation of Operator Overloading in C++ | CLO-2 | Apply | 2.6.1 | 10 |
| 6. | Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams and Collaboration diagrams | CLO-2 | Analysis | 4.6.1 | Mini Project Given |
| 7. | Implementation of Inheritance concepts in C++ | CLO-3 | Apply | 2.6.1 | 10 |
| 8. | Implementation of Virtual function & interface concepts in C++ | CLO-3 | Apply | 2.6.1 | 10 |
| 9. | Using the identified scenarios in your project, draw relevant state charts and activity diagrams. | CLO-3 | Analysis | 4.6.1 | Mini Project Given |
| 10. | Implementation of Templates in C++ | CLO-3 | Apply | 2.6.1 | 10 |
| 11. | Implementation of Exception of Handling in C++ | CLO-4 | Apply | 2.6.1 | 10 |
| 12. | Identify the User Interface, Domain objects, and Technical Services. Draw the partial layered, logical architecture diagram with UML package diagram notation such as Component Diagram, Deployment Diagram. | CLO-5 | Analysis | 4.6.1 | Mini Project Given |
| 13. | Implementation of STL Containers in C++ | CLO-6 | Apply | 2.6.1 | 10 |
| 14. | Implementation of STL associate containers and algorithms in C++ | CLO-6 | Apply | 2.6.1 | 10 |
| 15. | Implementation of Streams and File Handling in C++ | CLO-6 | Apply | 2.6.1 | 10 |

**LIST OF EXPERIMNENTS FOR UML DESIGN AND MODELLING:**

**To develop a mini-project by following the exercises listed below.**

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant state charts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.

**Suggested Software Tools for UML:**

StarUML, Rational Suite, Argo UML (or) equivalent, Eclipse IDE and Junit

## ABSTRACT

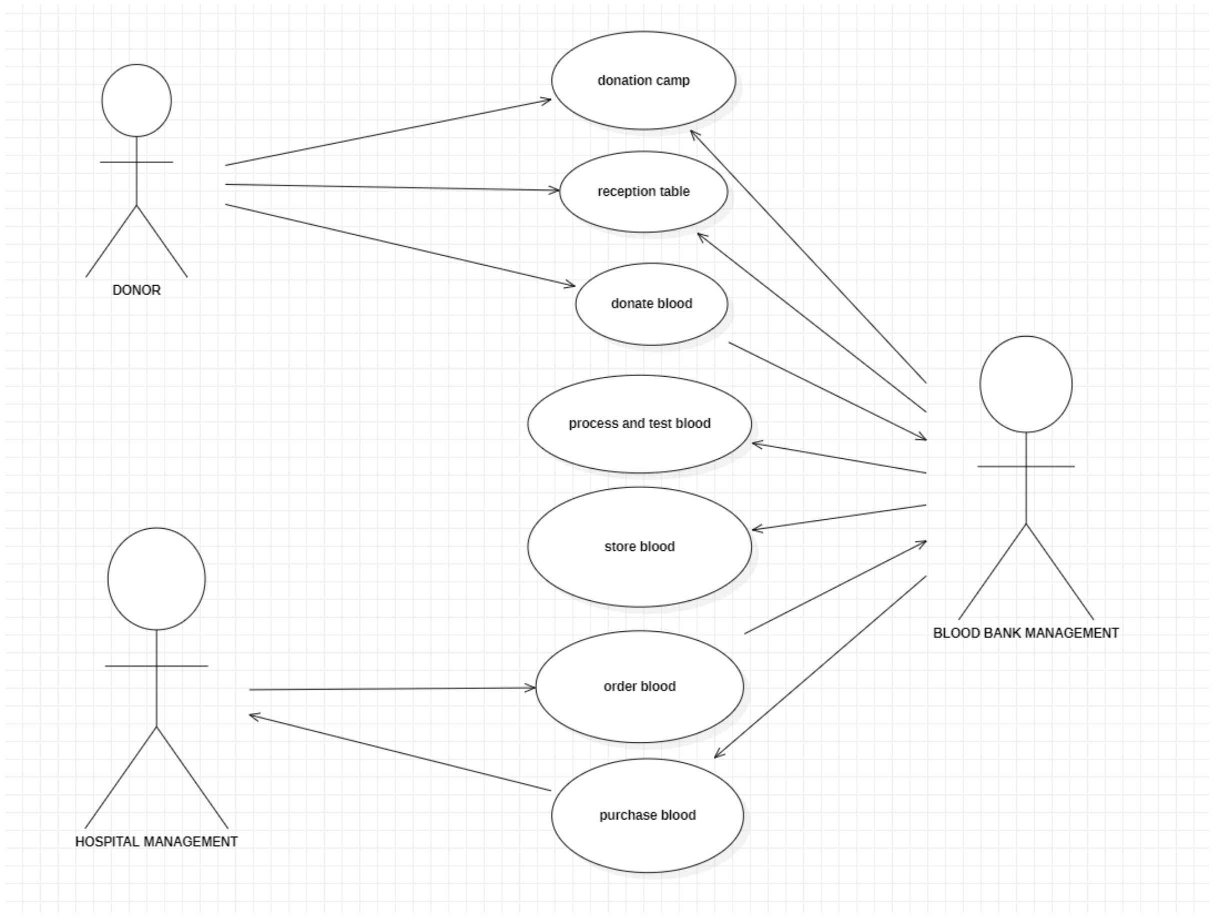
Blood bank management system UML DIAGRAMS are diagrams formed by the Unified Modeling Languages that acts as the blueprint for the project development. It shows the needed diagrams based in UML to guide you in building your Blood Bank Management System. These UML Diagrams is composed of Use Cases, Activity Diagrams, Class, Sequences and many more. The Blood Bank Management System is an application that stores, processes, retrieves, and analyzes data about blood bank administration. It also supervises the blood inventory management and other blood bank-related activities. The major goal of the blood bank management system is to keep track of blood, donors, blood groups, blood banks, and stock information. It keeps track of all information concerning blood, blood cells, patients, donor, and blood.

## MODULE DESCRIPTION

The main activity involved in these UML Diagram of Blood Bank Management System are as follows: -

- Manage Recipients – The admin has access to the donor management. He can update the blood, can update camp details users, and can view the details of the customer who requested blood.
- Manage Donator – The admin has access to the donator of the blood. He can update the details of the donator.
- Login and Logout – By default one of the security features of this system is the secure login and logout system. The login and logout system of this Blood Bank Management System uses a session. It means that the user can only log in at once on the same browser.
- Donor Registration – For the donor registration, you will fill the forms. Such as your complete name, gender, date of birth, blood type, phone number, email address, home address and etc.
- Contact Information – For the contact us, you will be able to see their address, phone number and email address.
- Blood Letting Information – For the about us, you will be able see the goals and objectives of blood bank management system.

## Use case diagram with explanation

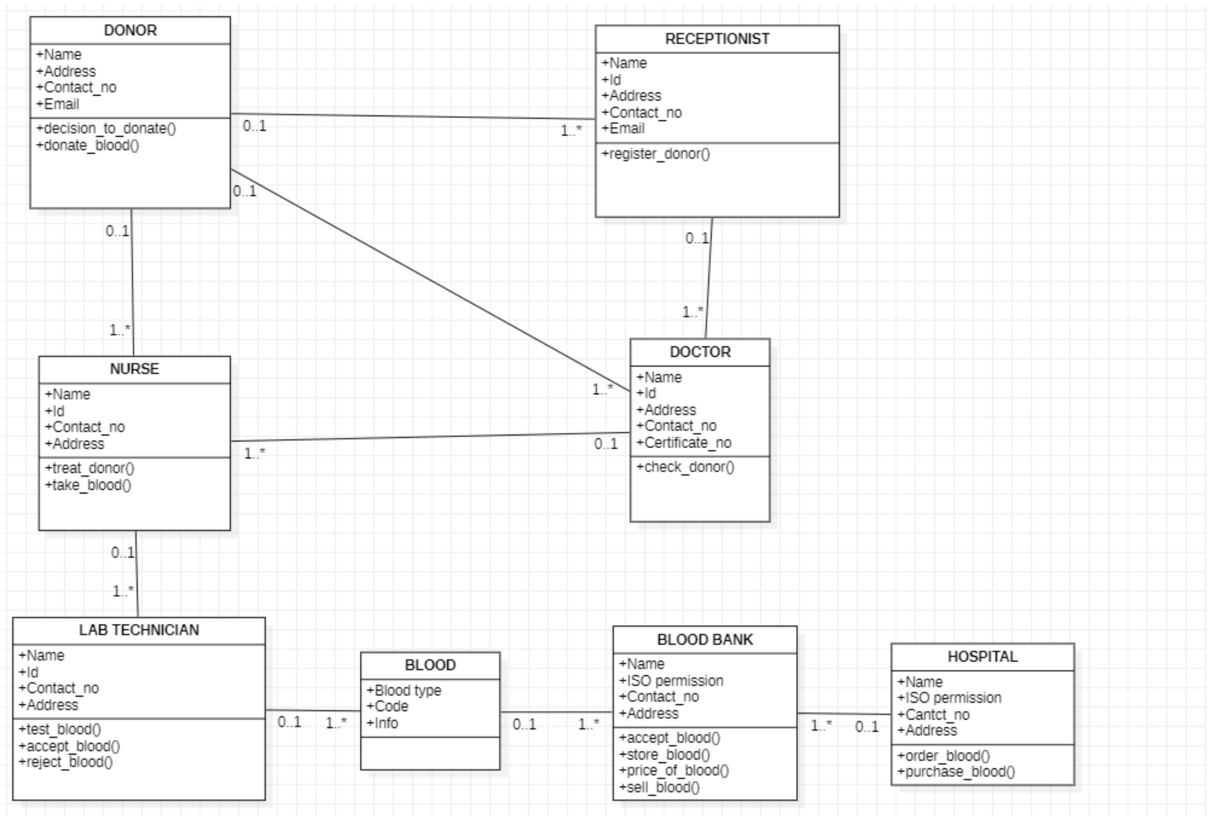


Report:

Cases included in following use-case diagram:

- Donation camp
- Reception table
- Donate blood
- Process & test blood
- Store blood
- Order blood
- purchase blood

## Class diagram with explanation

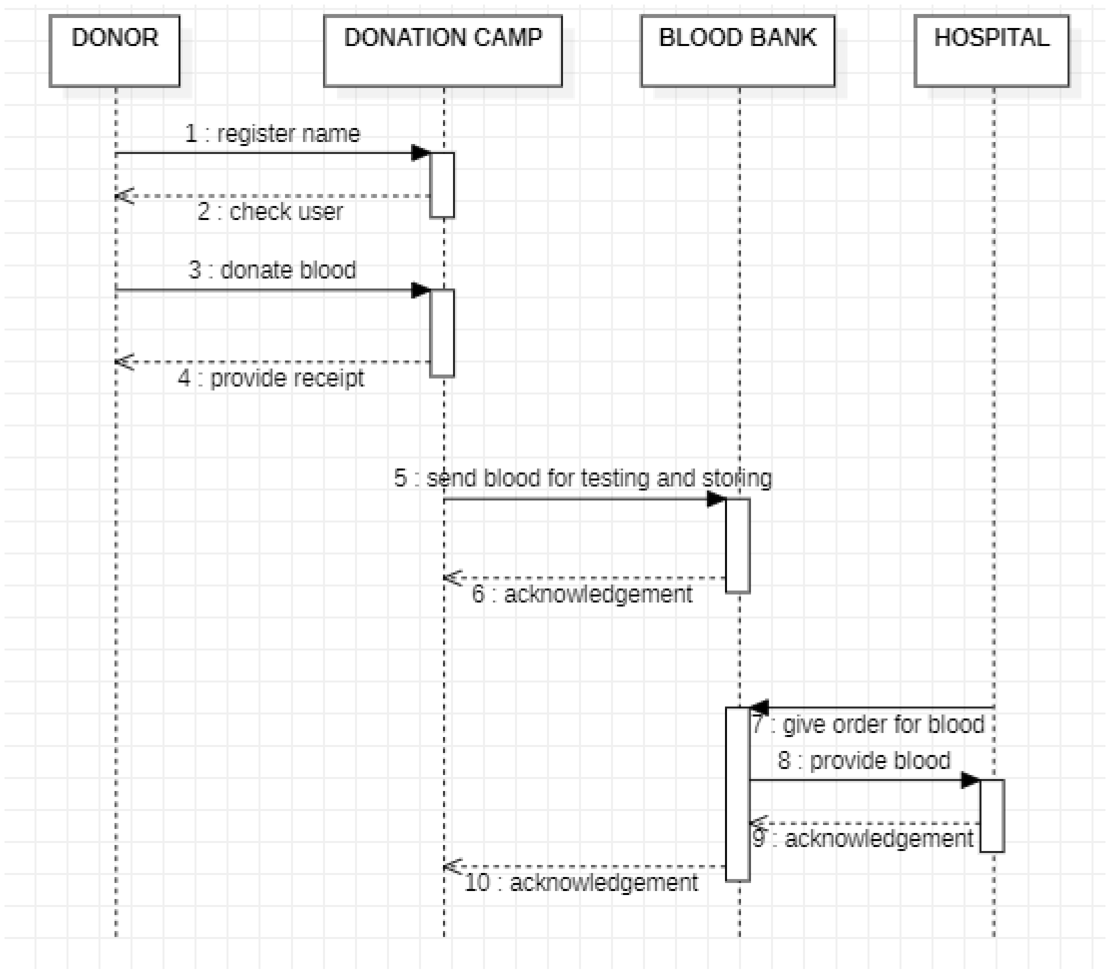


Report:

Classes included in the diagram:

- Donor: To hold details of the Donor.
- Receptionist: To register the Donor.
- Doctor: To check Donor.
- Nurse: To take blood and treat Donor.
- Lab technician: To test blood.
- Blood: To hold details of blood.
- Blood bank: To store blood database.
- Hospital: To purchase blood.

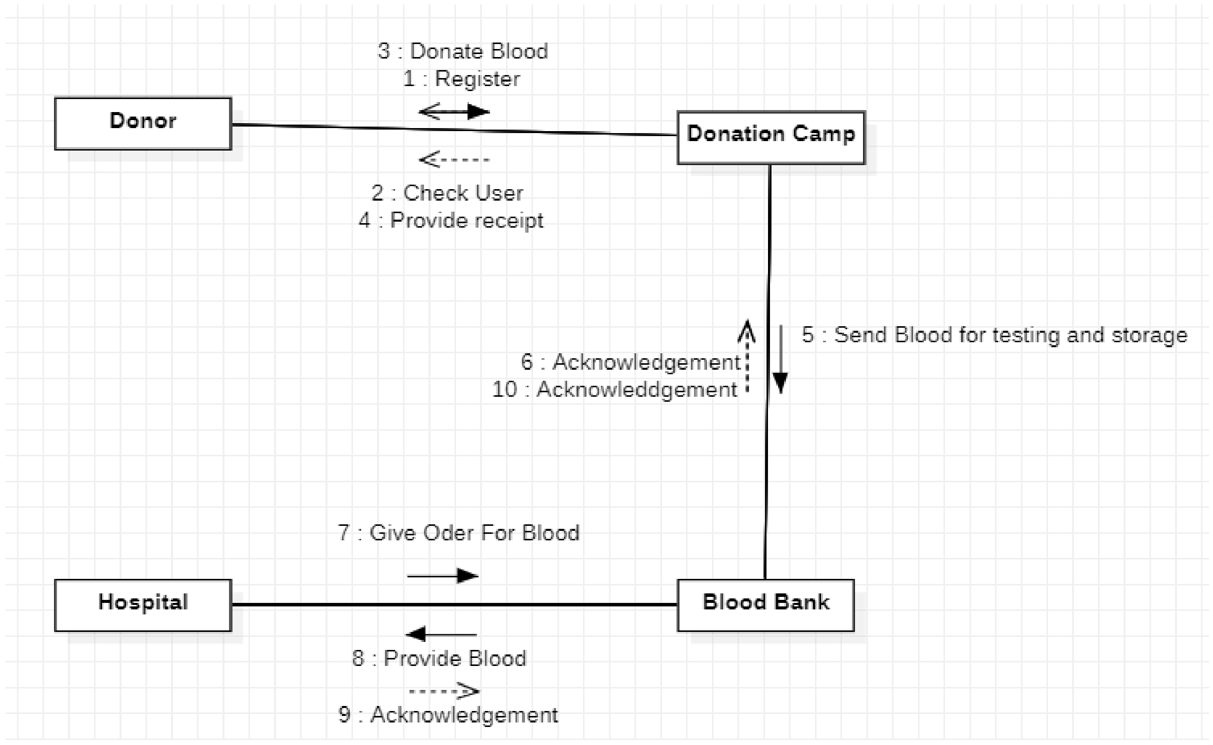
## Sequence diagram with explanation



Report:

- Donor registers name
- User is checked by donation camp
- Donor donates blood
- Donation camp provides receipt
- Donation camp sends blood for testing
- Blood bank replies with an acknowledgement to Donation camp
- Hospital orders blood
- Blood bank provides blood
- Hospital replies with an acknowledgement to Blood bank
- Blood bank replies with an acknowledgement to Donation camp

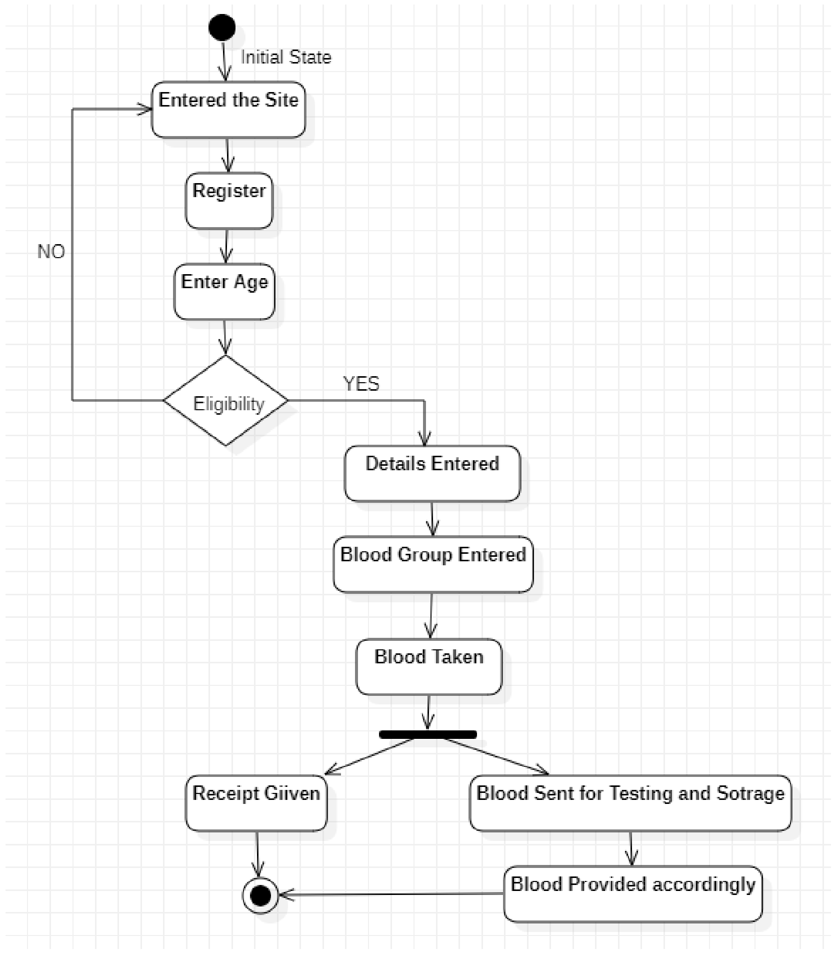
## Communication diagram with explanation



Report:

- Donor registers name
- User is checked by donation camp
- Donor donates blood
- Donation camp provides receipt
- Donation camp sends blood for testing
- Blood bank replies with an acknowledgement to Donation camp
- Hospital orders blood
- Blood bank provides blood
- Hospital replies with an acknowledgement to Blood bank
- Blood bank replies with an acknowledgement to Donation camp

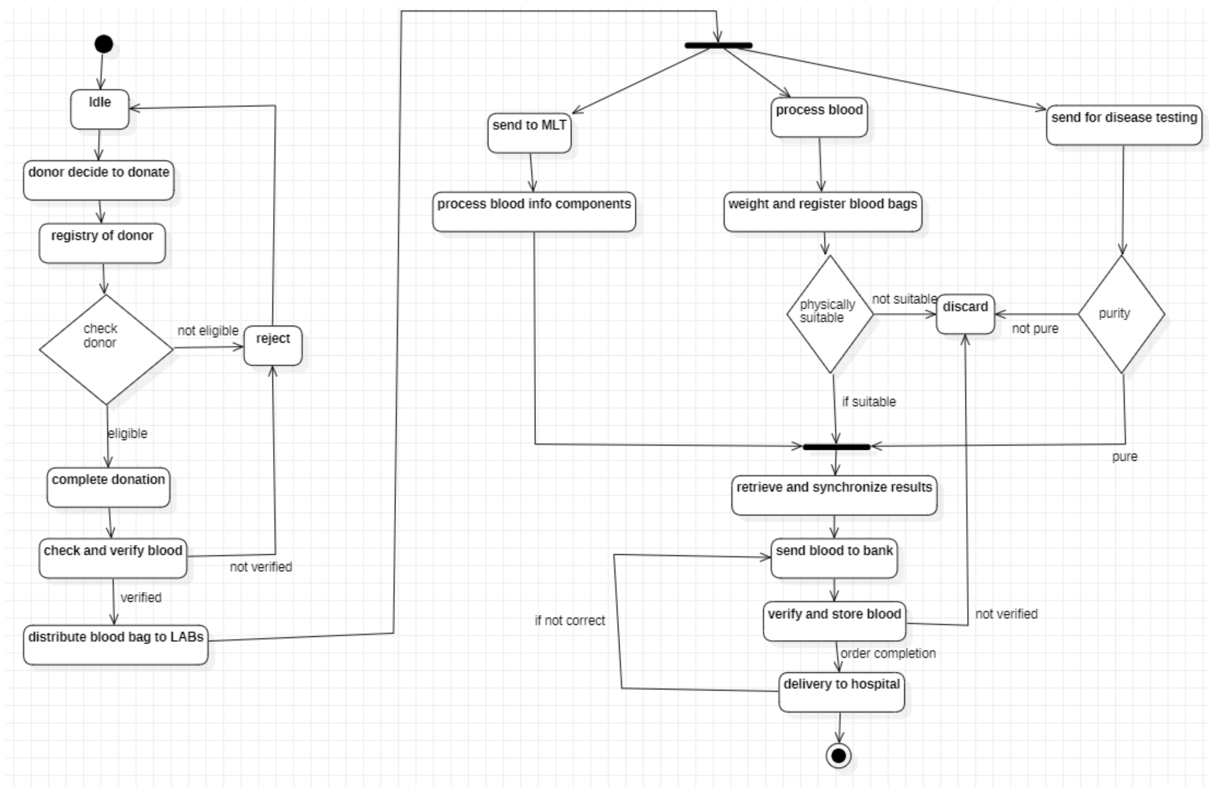
## State chart diagram with explanation



Report:

- User enters the hospital’s website.
- User registers their Id.
- User enters their age, if they are eligible, they can move with further process else they are redirected to website homepage.
- User enters their details related to blood.
- Blood is taken, receipt is provided to the patient, blood is sent for testing and sorting and is provided to the needed accordingly.

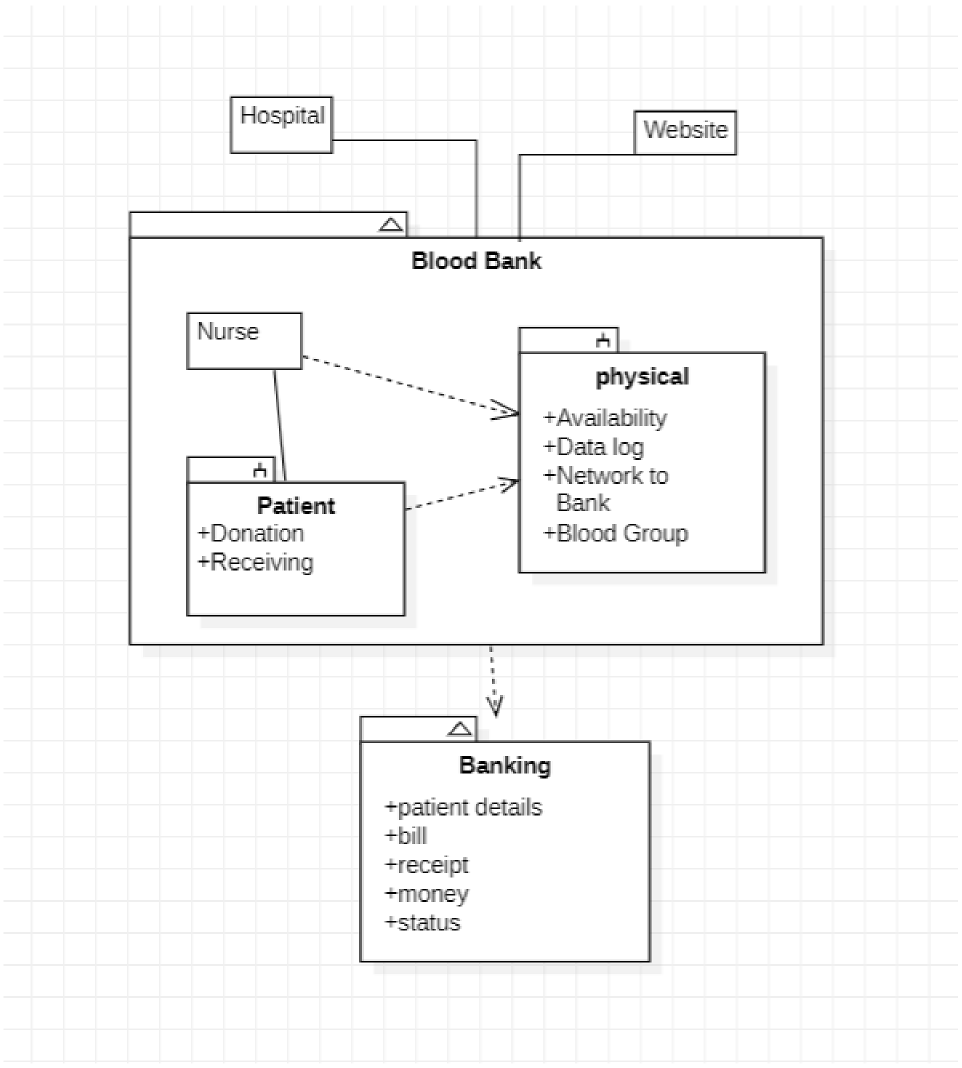
## Activity diagram with explanation



Report:

- Patient is ready and registers for donation.
- Patient is check for eligibility criteria, if eligible they are allowed to proceed with donation else donation is rejected
- Donation is completed and blood is verified, if verified the blood bags are sent to the lab else, they are rejected.
- The verified blood sent for disease testing, processing and to MLT. Blood is processed in the labs, if it passes any of the eligibility, it is retrieved and sent to the blood bank else it is discarded.
- The eligible blood is verified and stored and deliver to hospital.

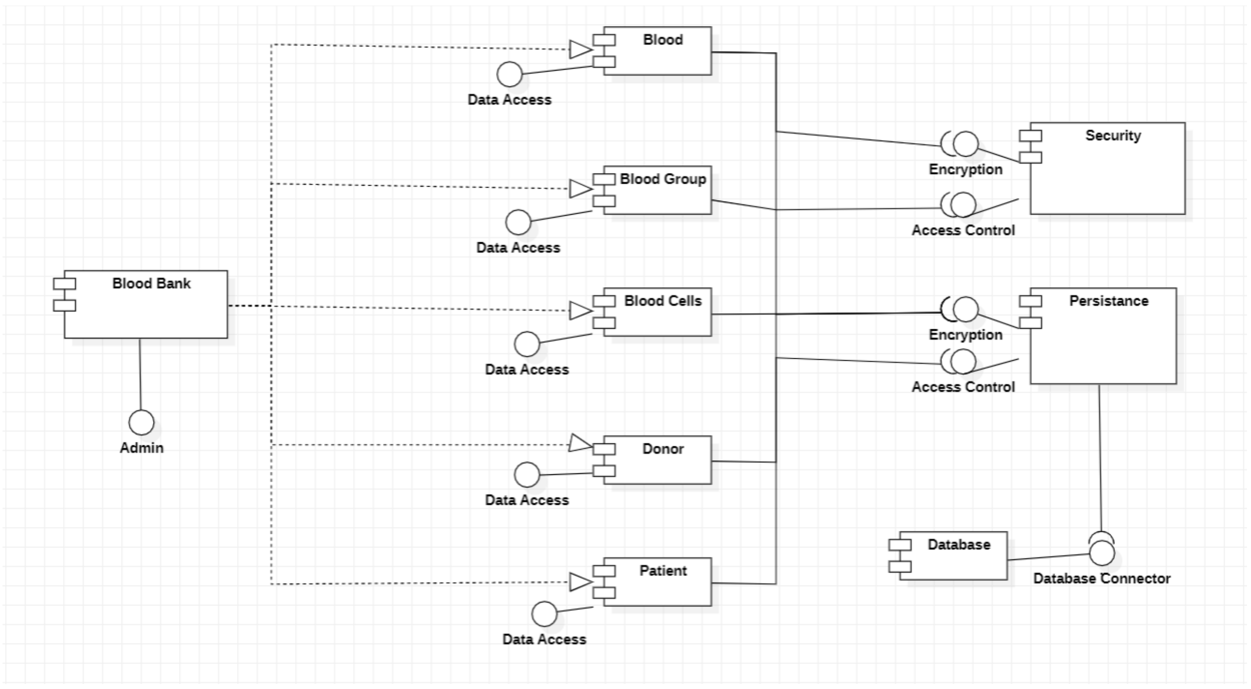
## Package diagram with explanation



Report:

- Hospital and Hospital’s website, both are linked to the Blood Bank package.
- The package contains the Nurse, who interacts with the patients accordingly.
- Nurse also keeps the track of the physical elements required which are under the physical sub package.
- Blood Bank is depended on the Banking package which contains the details of patients and their activity.

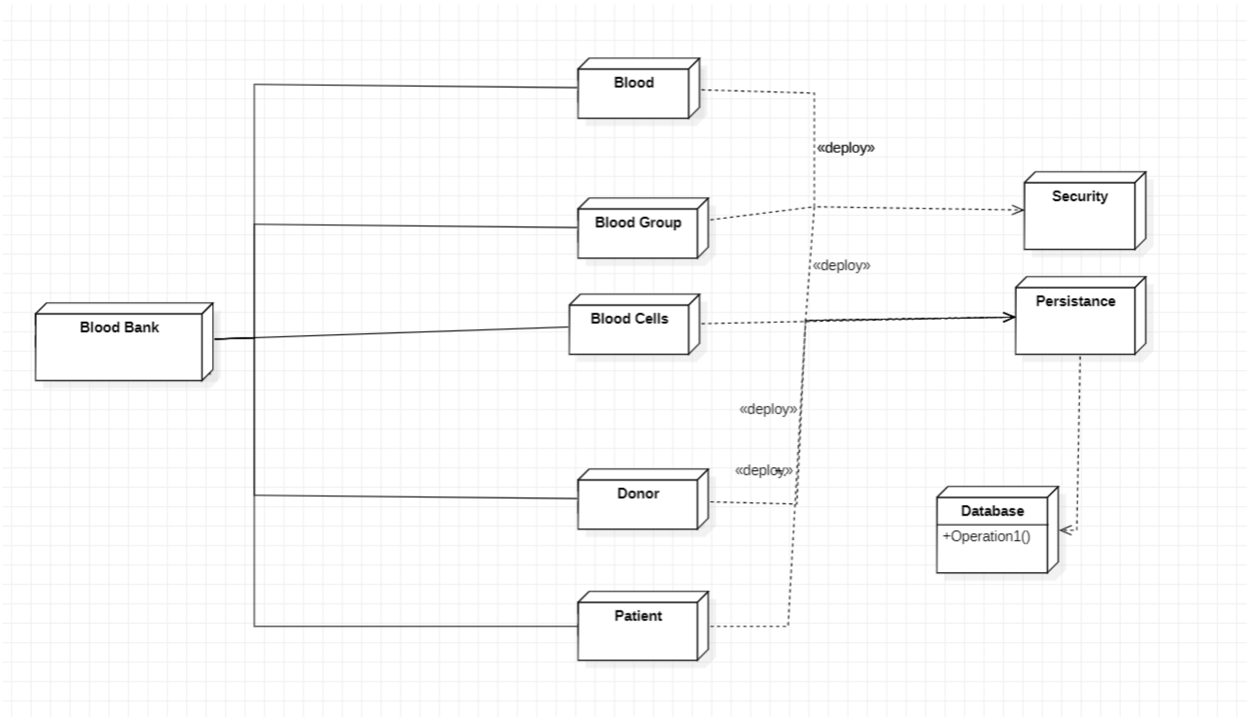
## Component diagram with explanation



Report:

- Admin has access to everything in the Blood Bank through interface.
- Blood, Blood Group, Blood Cells, Donor and Patient are all stored under the Blood Bank. The Admin has access to all of them.
- Security and Persistence are dependent upon Blood, Blood Group, Blood cells, Donor and Patient. The data of everything is encrypted and can be accessed only by the Admin.
- Persistence is dependent on Database which contains the data of Blood Bank.

## Deployment diagram with explanation



Report:

- Blood, Blood Group, Blood Cells, Donor and Patient are all linked with Blood Bank.
- Security and Persistence are linked with Blood, Blood Group, Blood cells, Donor and Patient.
- Persistence is linked with Database which contains the data of Blood Bank.

## Conclusion

As a whole, the UML Diagrams works together to achieve the most desired functions of a Blood Bank Management Project System. All of these were designed to guide programmers and beginners about the behavior and structure of Blood Bank Management System.

By completing all the given Diagrams, the Blood Bank Management Project System development would be much easier and attainable. So those UML diagrams were given to teach you and guide you through your project development journey. You can use all of the given UML diagrams as your reference, or have them for your project development. The ideas presented in UML Diagrams were all based on Blood Bank Management System requirements.

## References

https://www.wikipedia.org/

https://plantuml.com/

https://www.uml.edu/

https://www.lucidchart.com

https://www.studocu.com