# To understand Use Case Diagram

# Blood Bank Management System


**Prepared by**

| | |
|---|---|
| Abhishek Girkar | VU4F1718022 |
| Mohit Khambayat | VU4F1718072 |
| Ajay Waghmare | VU2T4S1718028 |
| Rasika Mahadik | VU4F1718004 |

**Instructor:** Mr Vinod sapkal
**Course** : SEPM
**Class:** TE-IT(A)/Batch A

## Experiment 3

## Aim: To understand Use Case Diagram

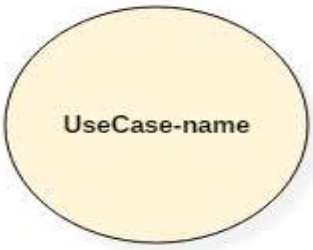
### Theory:

**Use Case Diagram** captures the system's functionality and requirements by using actors and use cases. Use Cases model the services, tasks, function that a system needs to perform. Use cases represent high-level functionalities and how a user will handle the system. Use-cases are the core concepts of Unified Modelling language modeling.

### Use Case Diagram Notation

### Use-case:

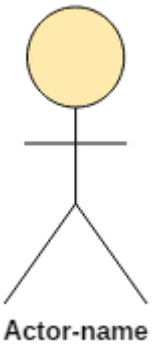
Use cases are used to represent high-level functionalities and how the user will handle the system. A use case represents a distinct functionality of a system, a component, a package, or a class. It is denoted by an oval shape with the name of a use case written inside the oval shape. The notation of a use case in UML is given below:




UML UseCase Notation

### Actor:

It is used inside use case diagrams. The actor is an entity that interacts with the system. A user is the best example of an actor. An actor is an entity that initiates the use case from outside the scope of a use case. It can be any element that can trigger an interaction with the use case. One actor can be associated with multiple use cases in the system. The actor notation in UML is given below.




UML Actor Notation

## System:

It is used to draw system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

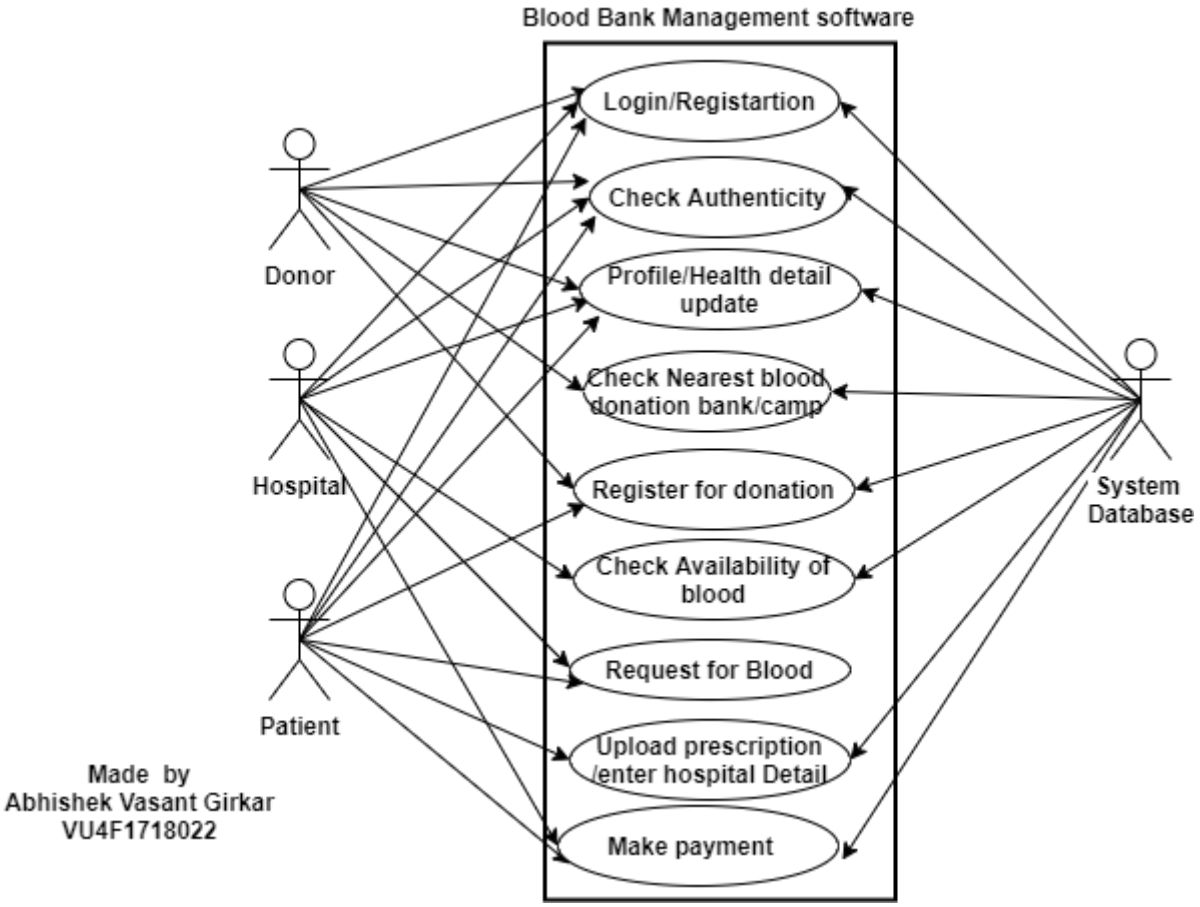
**Relationships**

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.

<<include>>

<< exclude>>

**Diagram**




**Conclusion:** Hence, We studied and understood Use Case Diagram