

UML Diagrams

UML stands for Unified Modeling language. It is a rich language to model software solutions, application structures, system behavior and business processes.

There are 14 UML diagrams types.

UML Diagram Types:-

There are two main categories ① Structure diagrams

② Behavioral diagrams.

① Structure diagrams:-

=> Structure diagrams show the things in the modeled system. In a more technical term,

=> They show different objects in a system.

Types:-

** class diagram

* Component diagram

** Deployment diagram

* Object diagram

* Package diagram

* Profile diagram

* Composite structure diagram.

② Behavioral diagrams

=> Behavioral diagrams show what should happen in a system.

=> They describe how the objects interact with each other to create a functioning system.

Types

** Use case diagram

* Activity diagram

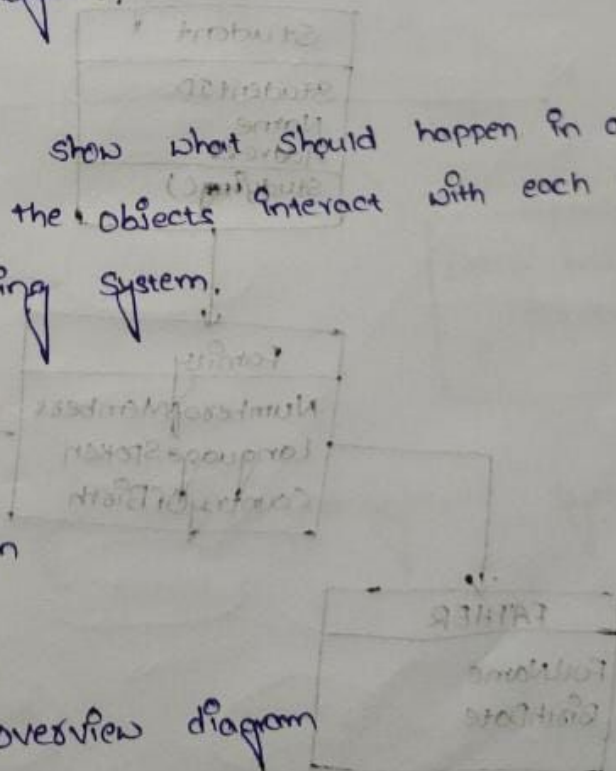
** State machine diagram

** Sequence diagram

* Communication diagram

* Interaction diagram overview diagram

* Timing diagram.



Class Diagram

Class diagrams are the main building block of any Object-Oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class.

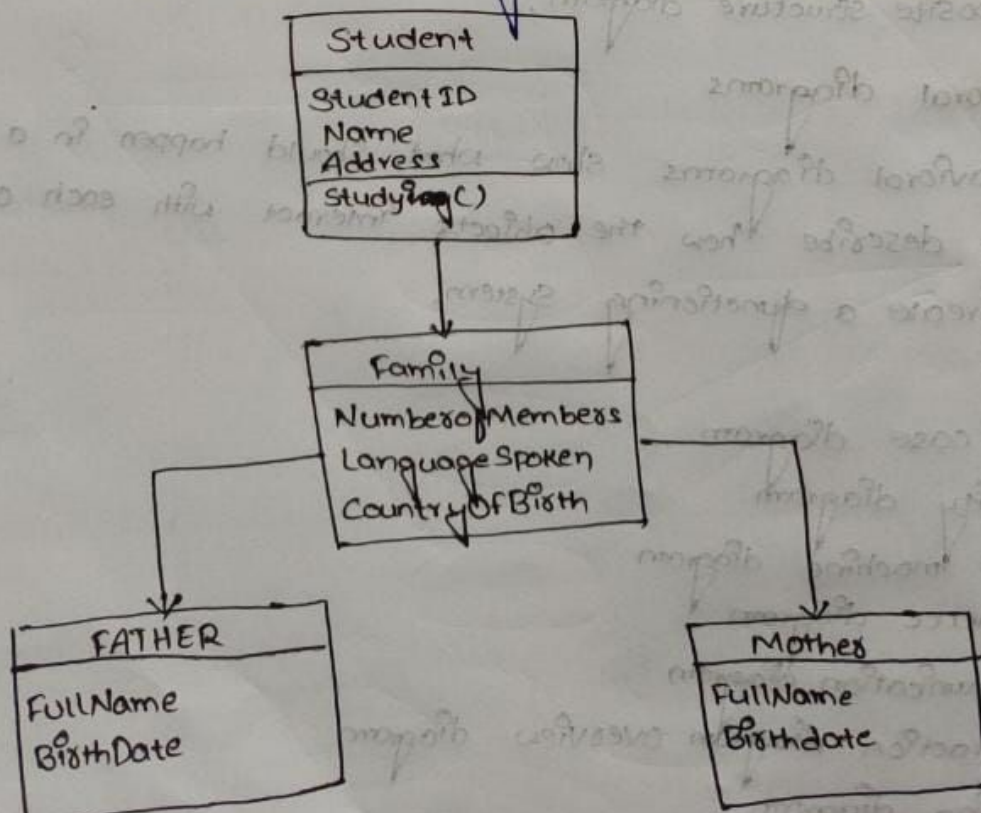
In most modeling tools, a class has three parts

- * Name at the top
- * Attributes in the middle
- * Operations or methods at the bottom.

In a large system with many related classes, classes are grouped together to create class diagrams.

Different relationships between classes are shown by different types of arrows.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

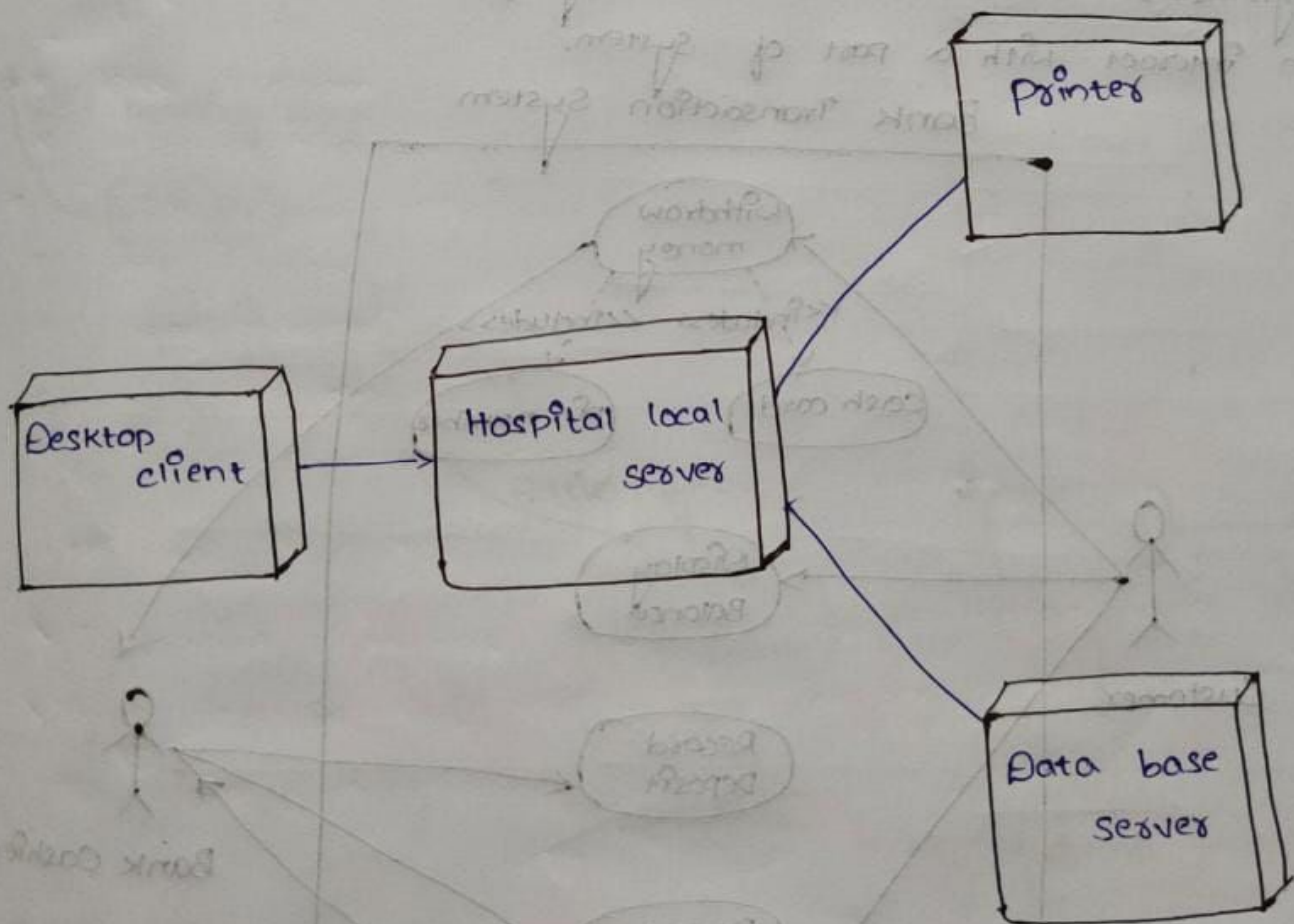


Deployment Diagram :-

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.



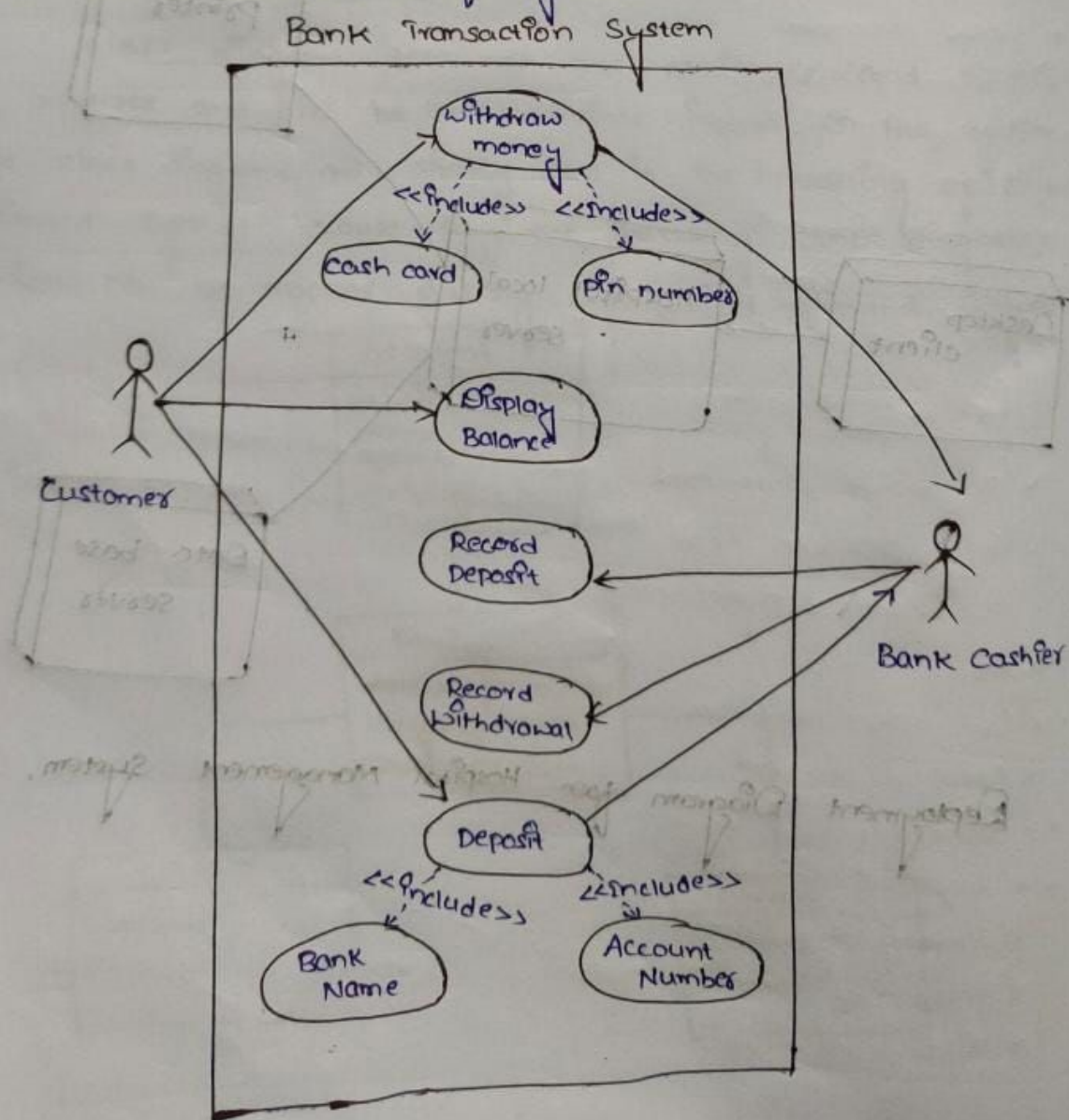
Deployment Diagram for Hospital Management System.

Use Case Diagram:-

A use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships.

It models the tasks, services, and functions required by a system or an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

→ The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirements which includes both internal as well as external influences. It involves persons, use case, and several things that involve the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of system.

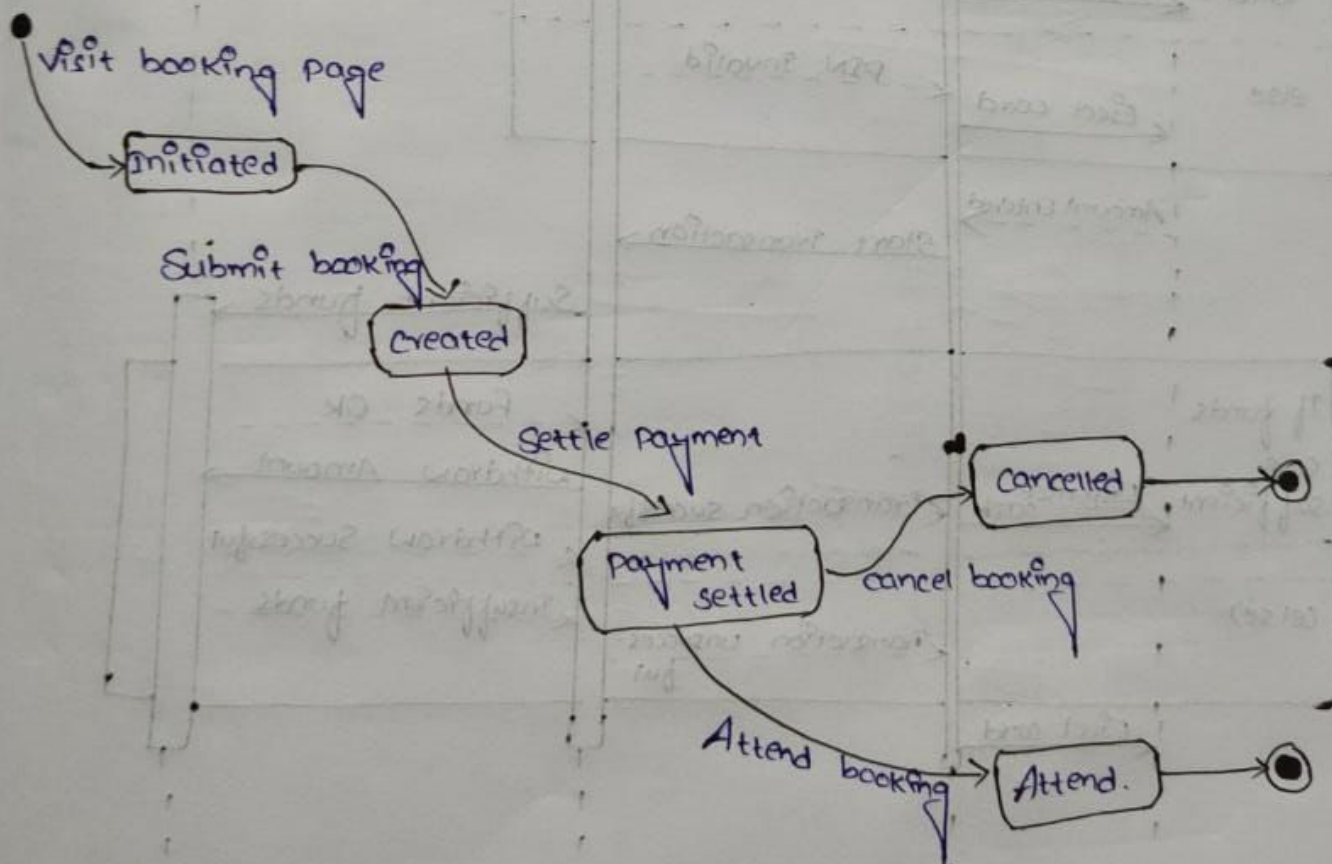


State Diagram:-

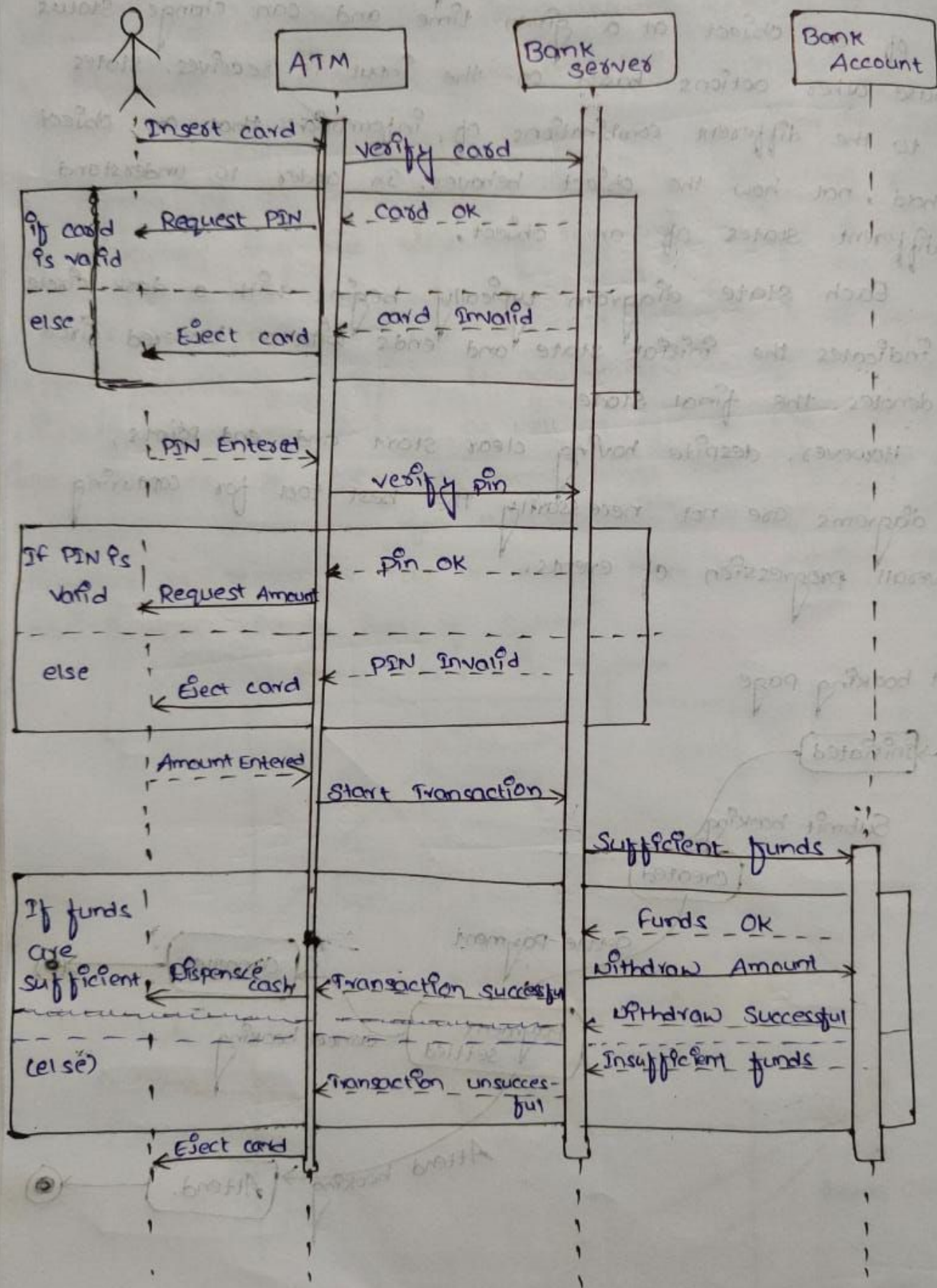
A state machine is any device that stores the status of an object at a given time and can change status or cause other actions based on the input it receives. States refer to the different combinations of information that an object can hold, not how the object behaves. In order to understand the different states of an object,

Each state diagram typically begins with a dark circle that indicates the initial state and ends with a bordered circle that denotes the final state.

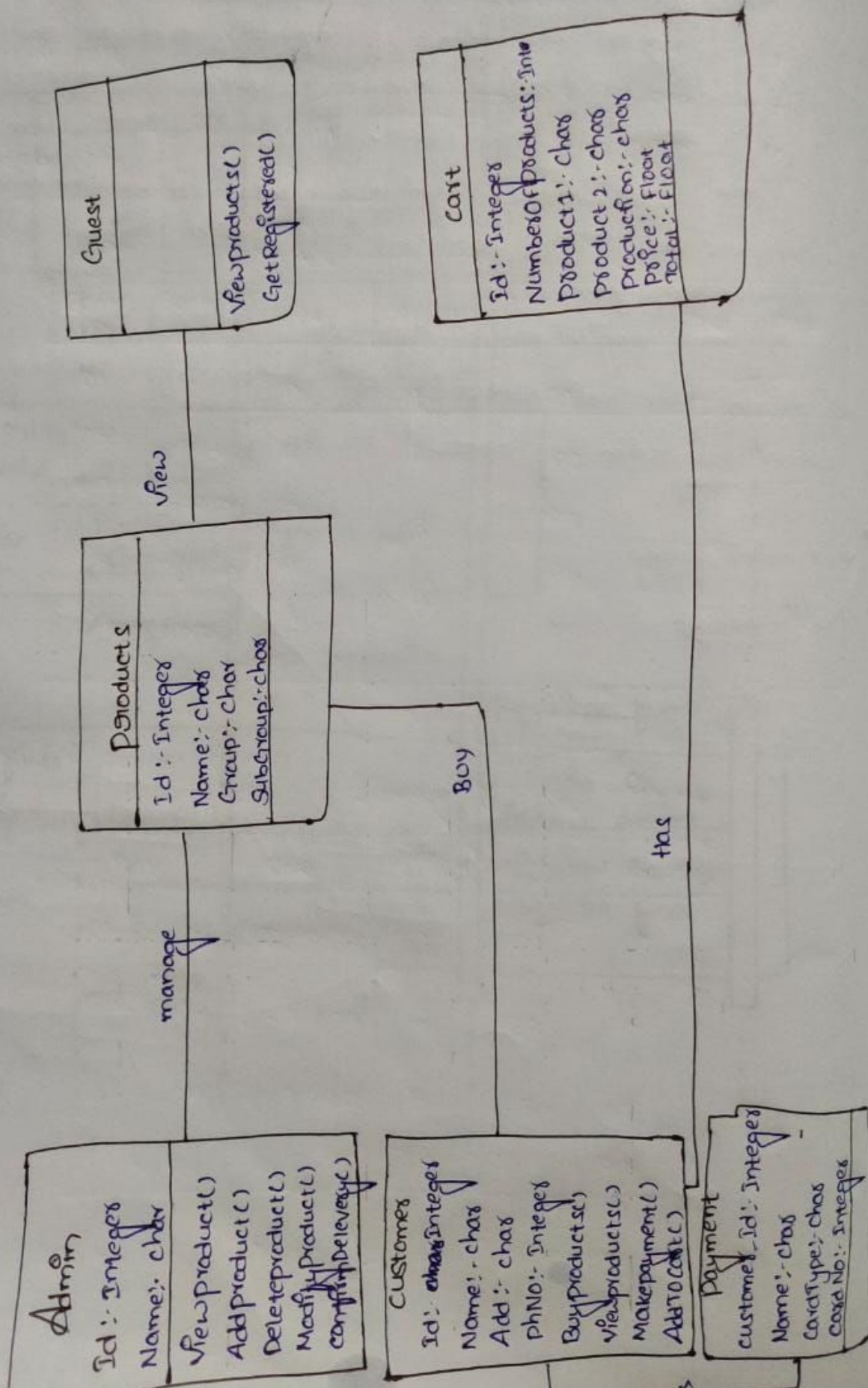
However, despite having clear start and end points, state diagrams are not necessarily the best tool for capturing an overall progression of events.



② Sequence Diagram:-



Class diagram for Shopping mart:-



(1)

Normalization

* Normalization We are using to splitted the bigger table into smaller tables.

* With the help of normalization we are able to remove the

* Data redundancy

* Anomalies problem

DML statements should not come multiple time based on the column record.

Types of Normalization:-

* 1NF (1st normal form)

* 2NF (2nd Normal form)

* 3NF (3rd Normal form)

* BCNF (Boy's Codd normal form)

* 4NF (4th Normal form)

* 5NF (5th Normal form)

1NF (1st normal form) :-

St ID	Student Name	mail	Subjects	Subject code	Book ID	BOOK NAME	Author DOI	DOR
1	A	A@gmail	C, C++	1, 2	01, 05	C, C++	09-01-17	20-3-18
2	B	B@gmail	C++, Java	2, 3	05, 07	C++, Java	10-1-18	15-04-18
3	C	C@gmail	Python	4	08	Python	4-3-18	5-06-19
4	D	D@gmail	SQL	5	09	SQL	5-3-19	7-4-20

St-ID	Student name	mail	Subjects	Subject code	Book ID	BOOK NAME	DOI	DOR
1	A	A@gmail	C	1	01	C	09-01-17	01-3-18
1	A	A@gmail	C++	2	05	C++	09-1-17	20-3-18
2	B	B@gmail	C++	2	05	C++	10-1-18	15-04-18
2	B	B@gmail	Java	3	07	Java	10-1-18	15-04-18
3	C	C@gmail	Python	4	08	Python	4-3-18	5-06-19
4	D	D@gmail	SQL	5	09	SQL	5-3-19	7-4-20

2NF (2nd normal form):-

Foreign-key Student Table

Student ID	Student name	mail	phone
1	A	A@gmail	97003..
2	B	B@gmail	6281..
3	C	C@gmail	9441..
4	D	D@gmail	9014..

Subject Table

Sub code	Subject name	Student ID
01	C	1
02	C++	2
03	Java	3
04	Python	4
05	SQL	

PK

Books

Book ID	Book Name	Book	Student ID	Subject Code	DOI	DOR
01	C	D	1	01	—	—
05	C++	E	2	02	—	—
07	Java	G	3	03		
08	Python	H	4	04		
09	SQL	I		05		

3NF (3rd normal form):-

Library

PK

PK

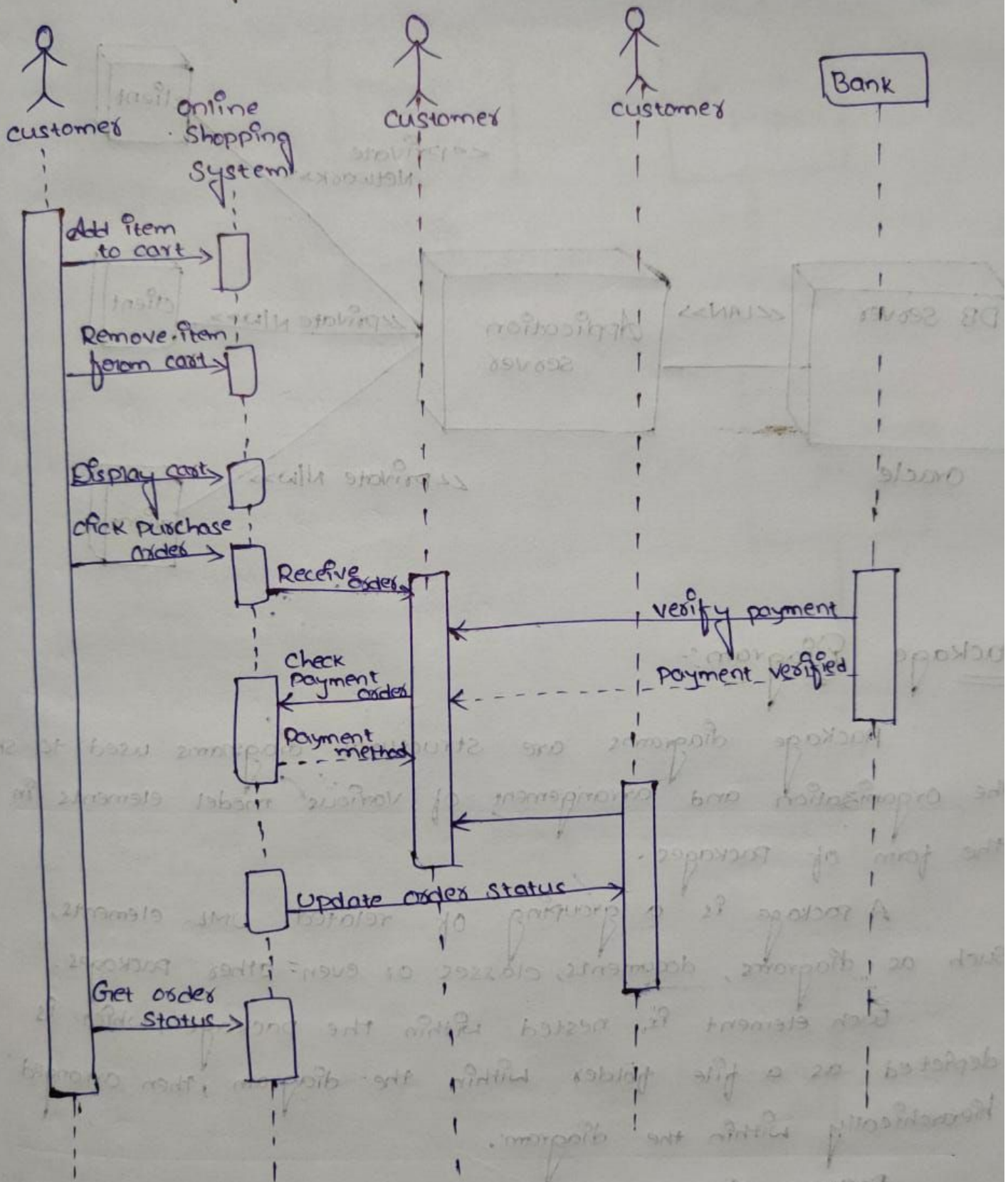
PK

DOI	DOR	Student ID	Subject code	Book ID
01-3-19	02-4-19	1	01	01
04-3-19	6-7-19	2	02	05
5-3-19	6-9-20	3	03	07
5-3-20	6-3-21	4	04	08

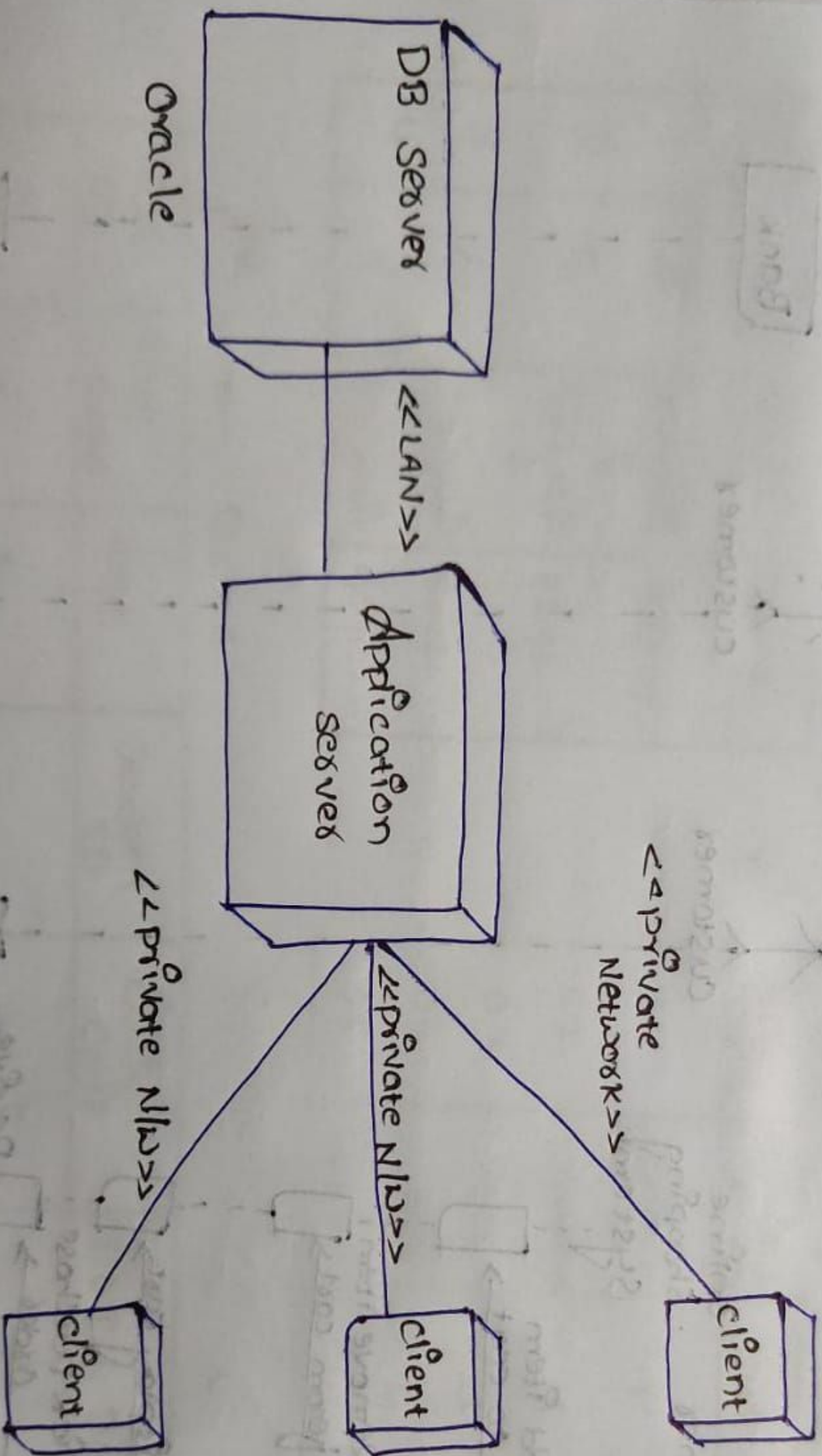
In a single table we have 3 primary key P_s called as "composite key". primary key".

* Sequence Diagram for the shopping mast:

⑥



Deployment diagram for book store



⑦ Package Diagram:

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages.

A package is a grouping of related UML elements, such as diagrams, documents, classes or even other packages.

Each element is nested within the package, which is depicted as a file folder within the diagram, then arranged hierarchically within the diagram.

Package diagrams are most commonly used to provide a visual organization of the layered architecture within any UML classified, such as a software system.

