

Register No: 19L31A0516

Experiment No: 1

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
		Signature of the Lab teacher	
Date			

AIM: To construct UML class diagram for the following case studies

Case Study 1:- Library Management System

Case Study 2:- Banking Application

Case Study 3:- Customer Support System

Class Diagram:-

Class diagrams are the most common diagram found in modeling object-oriented systems. A class diagram shows a set of classes, interfaces, and collaborations & their relationships.

We use class diagrams to model the static view of a system. For the most part, this involves modeling the vocabulary of the system, modelling collaborations, or modeling schemas.

## Contents:-

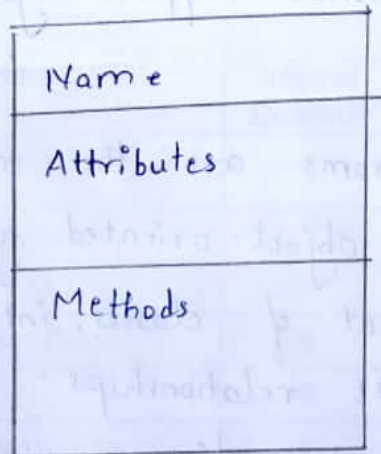
Class diagrams commonly contains the following things -

- Classes
- Interfaces
- Collaborations
- Dependency, generalization, association relationships.

## Classes:

A template for creating objects & implementing behaviour in a system. In UML, a class represents an object or a set of objects that share a common structure and behavior. They're represented by a rectangle that includes rows of the class diagram. In each row it consists of Name, Attributes, Methods.

Following is an example

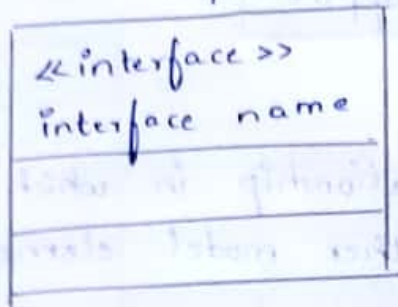


## Interfaces:-

An interfaces are model elements that define set of operations that other model elements, such as classes or components must implement. An implementing model element realizes an interface by overriding

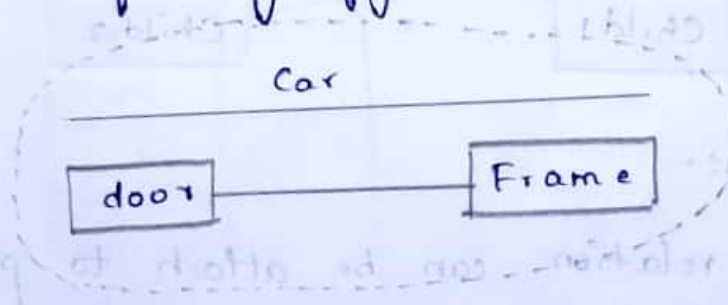


each of the operations that the interface declares.  
Following is the representation of interface



### Collaboration:-

Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send & receive messages. The goal of a collaboration can be to define the roles or the components of a classifier. The following figure illustrates a collaboration.



### Dependency:-

A dependency relationship in which one element depends on another element. It indicates the change in one element might require a change in another element. We can also use a dependency relationship to represent precedence where one model element must precede another.

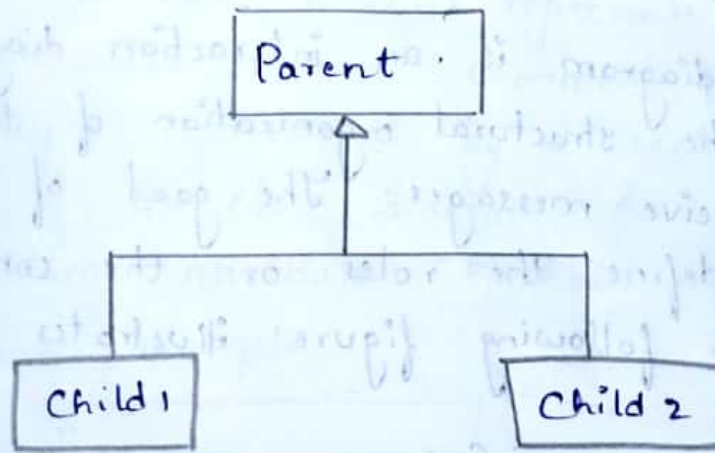
It is represented as follows :-



Generalization :-

A generalization is a relationship in which one model element is based on another model element.

It is represented as



Association :-

Association relation can be attach to provide additional information about the relationship. An association class is identical to other classes and can contain operations, attributes as well as other associations.

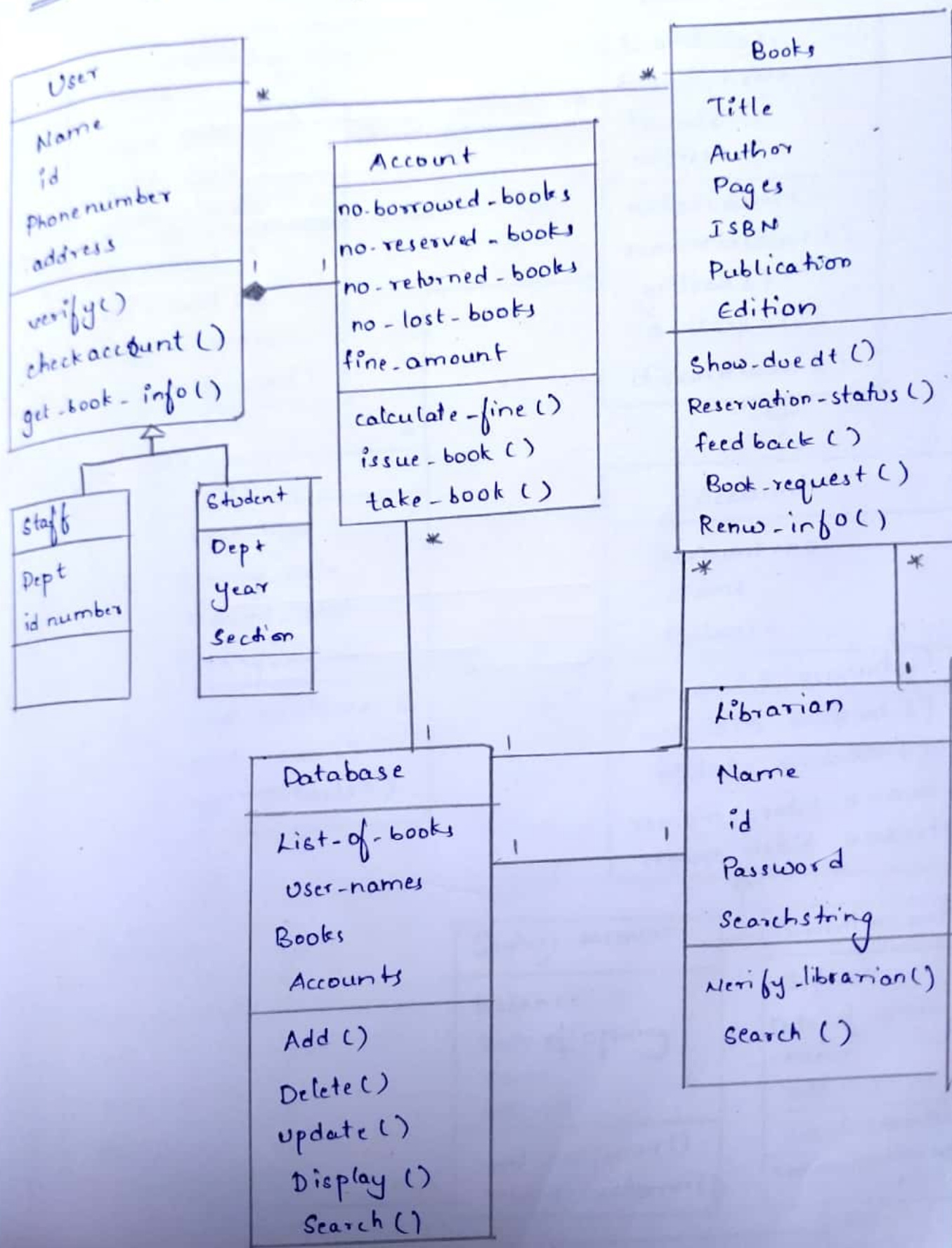


Register No: 19L31A0516

Experiment No: 1

Date:

# Base Study 1 : Library Management System

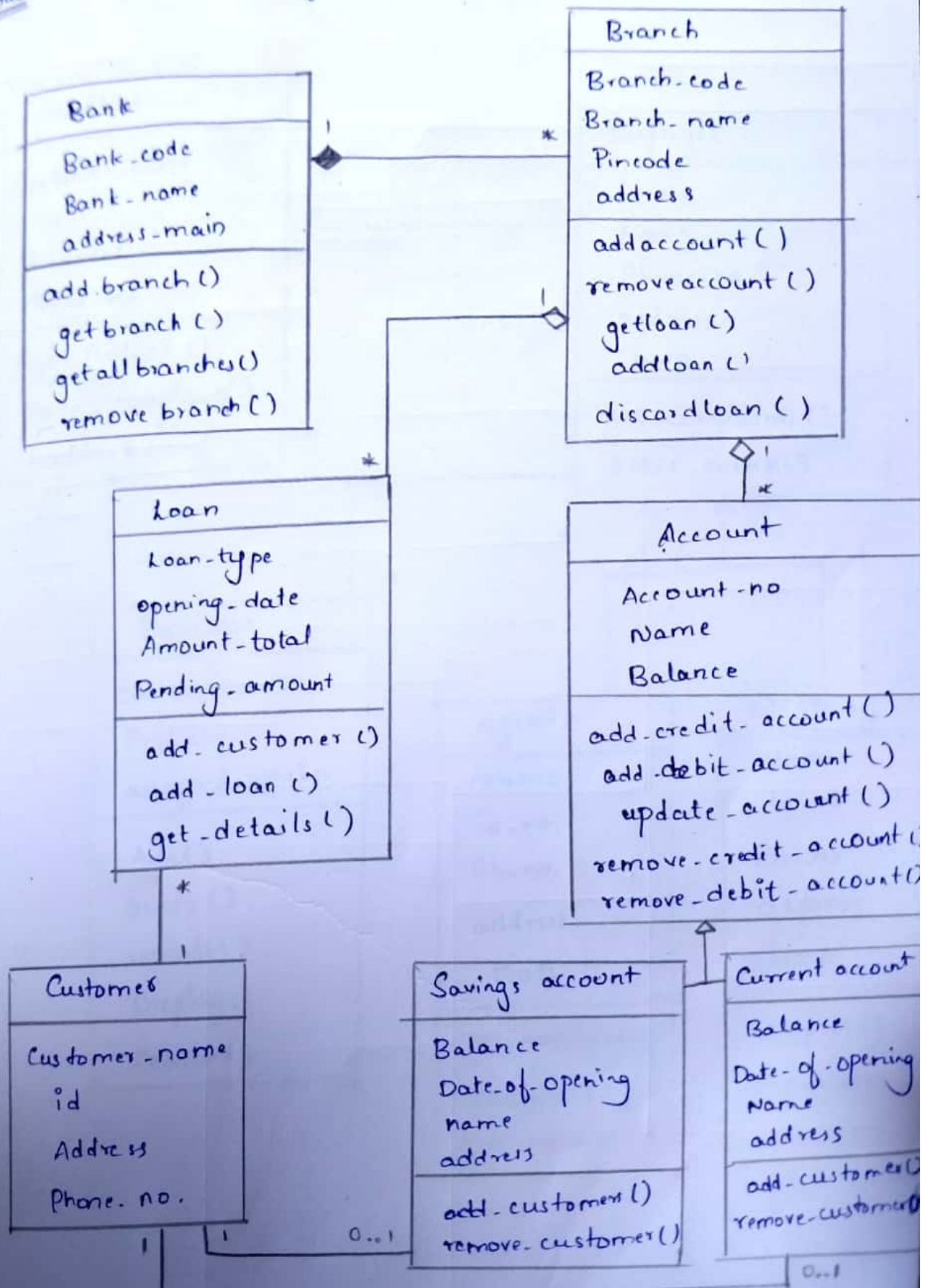


Register No: 19131A0516

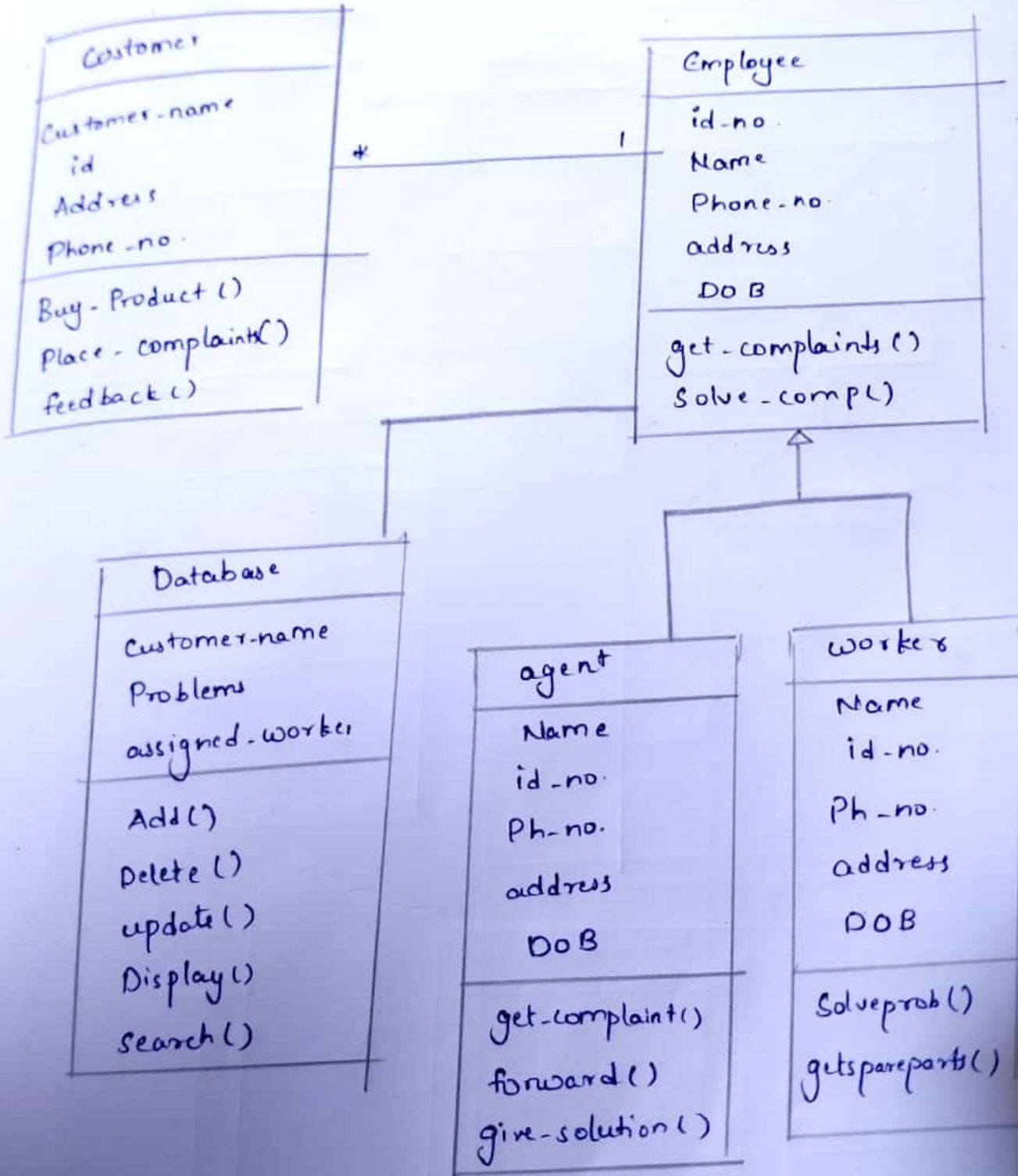
Experiment No: 1

Date:

## Case Study 2: Banking Application



## Case Study 3: Customer Support System





Register No: 19L31A0516

Experiment No: 2

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
		Signature of the Lab teacher	
Date			

AIM: To construct UML Object diagram for the following case studies.

Case Study 1 :- Library Management System

Case Study 2 :- Banking Application

Case Study 3 :- Customer Support System.

### Object Diagram:

Object diagrams model the instances of things contained in class diagram. An object diagram shows a set of objects and their relationships at a point in time. Object diagrams are important for visualizing, specifying, documenting structural models and for constructing the static aspects of systems through forward and reverse engineering.



## Contents:-

Object diagrams contain the following

→ Objects

→ Links

Objects:

Aim: To construct UML Object diagram for the following case studies.

- Case Study 1 :- Library Management System
- Case Study 2 :- Banking Application
- Case Study 3 :- Customer Support System

## Object Diagrams:

Object diagrams model the instance of things contained in class diagram. An object diagram shows a set of objects and their relationships at a point in time. Object diagrams are important for visualizing specifying, documenting structural models and for constructing the static aspects of systems through forward and reverse engineering.

Objects:

Objects are model elements that represent instance of a class. We can add objects to model for representing concrete and prototypical instance. A concrete instance represents an actual person or thing in the real world. A prototypical instance of a customer class contains data that represents a typical customer. It is represented as

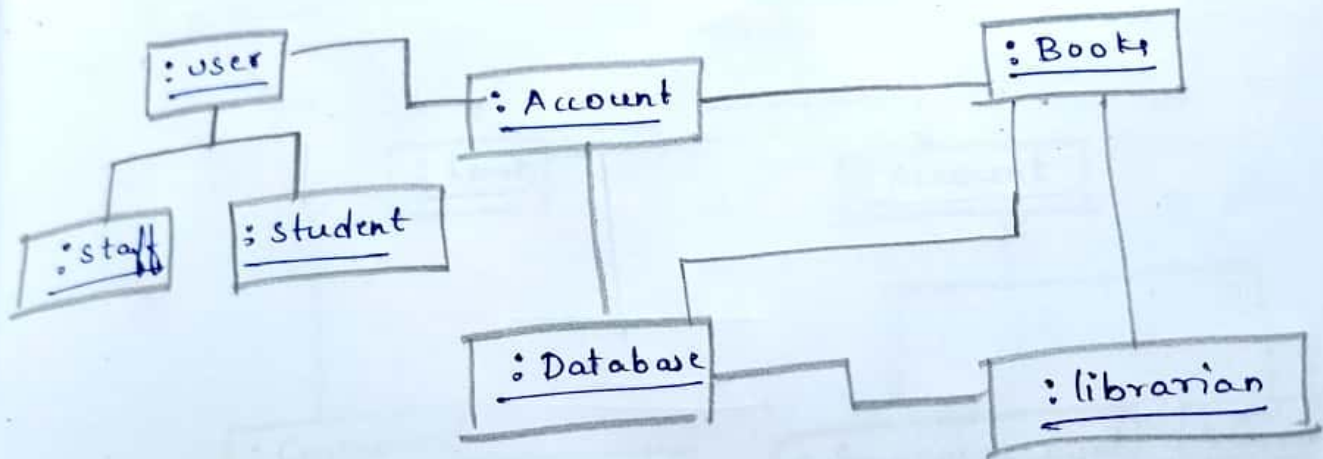
: object

Links :-

We use a link to represent a relationship between two objects. We represent the number of participants on the link for each end of the link. We use the term association for a relationship between two classifiers. The term link is used to specify a relationship between two objects.

Register No : 19L31A0516 \* Experiment No : 2 Date :

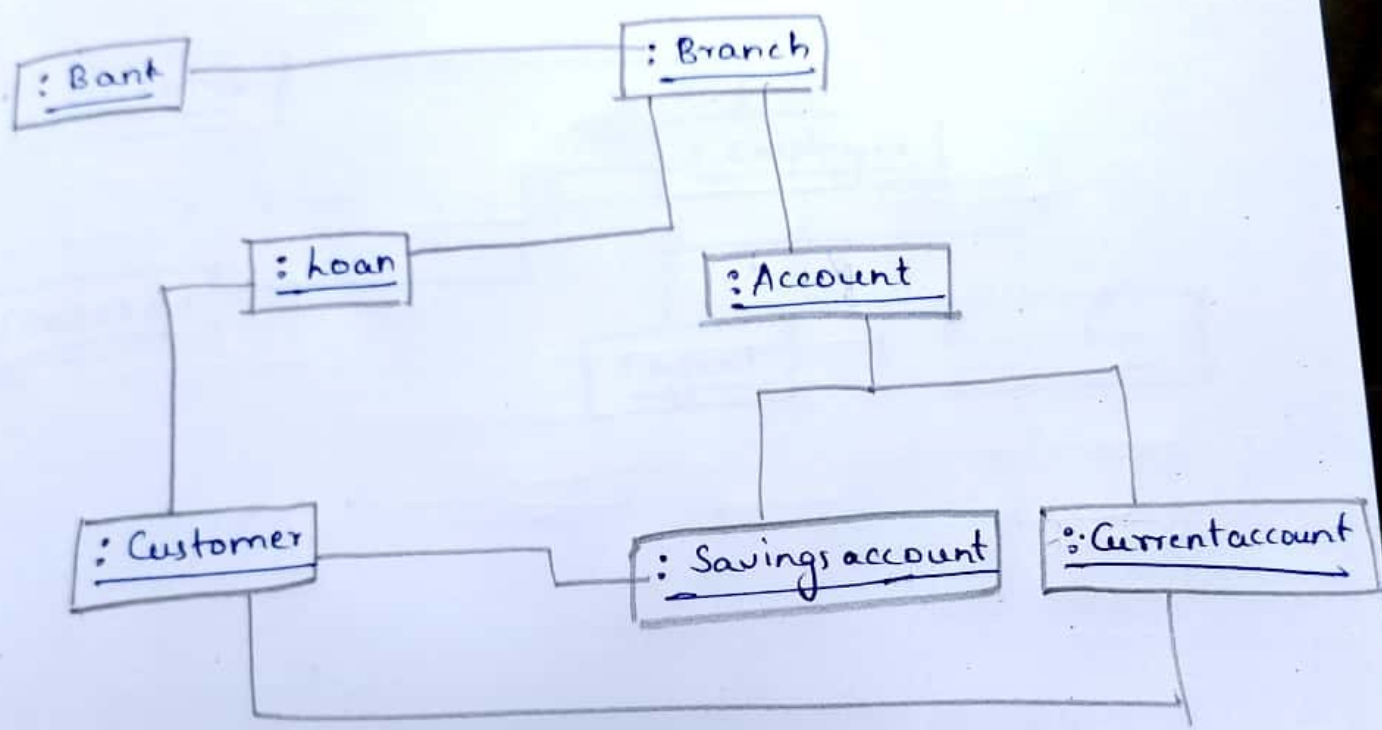
## Case Study 1 : Library Management System .





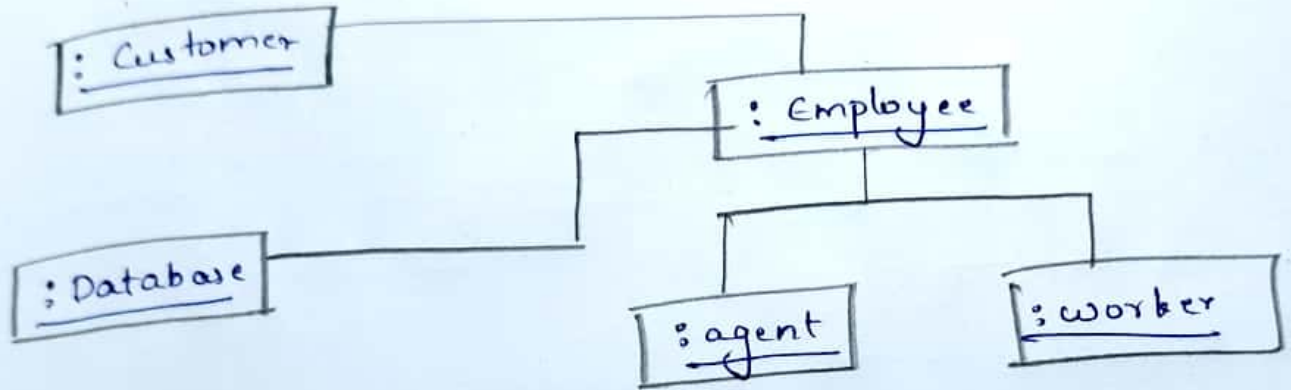
Register No: 19L31A0516 \* Experiment No: 2 Date:

Case study 2: Banking Application



Register No: 19L31A0516 • Experiment No: a Date:           

Case Study 3: Customer Support System



Register No: 19L31A0516

Experiment No: 3

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
Date	Signature of the Lab teacher		

AIM: To construct UML Use case diagram for the following Case Studies.

Case Study 1: Library Management System

Case Study 2: Banking Application

Case Study 3: Customer Support System

Use case diagram:

Use case diagrams are important for visualizing specifying, and documenting the behaviour of an element. They make systems, subsystems, and classes approachable and understandable by presenting an outside view of how those elements may be used in context. Use case diagrams are also important for testing executable systems through forward engineering and for comprehending executable system through reverse engineering.



## Contents :-

Use case diagrams commonly contain

→ Actors

→ Dependency, generalization and association.

## Actors :-

A use case diagram shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware or other systems. An actor is usually drawn as a named stick figure.



Customer

## Dependency :-

A dependency relationship in which one element depends on another element. It indicates the change in one element might require a change in another element. We can also use dependency to represent precedence, where one model element must precede another.

It is represented as follows -



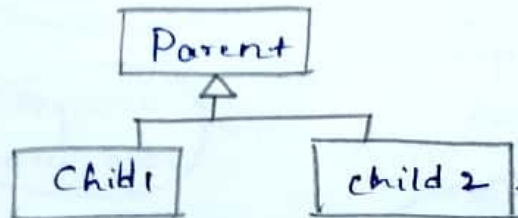
Register No: 19L31A0516

Experiment No: 3

Date:

### Generalization:-

A generalization is a relationship in which one model element is based on another model element. It is represented as:-



### Association:-

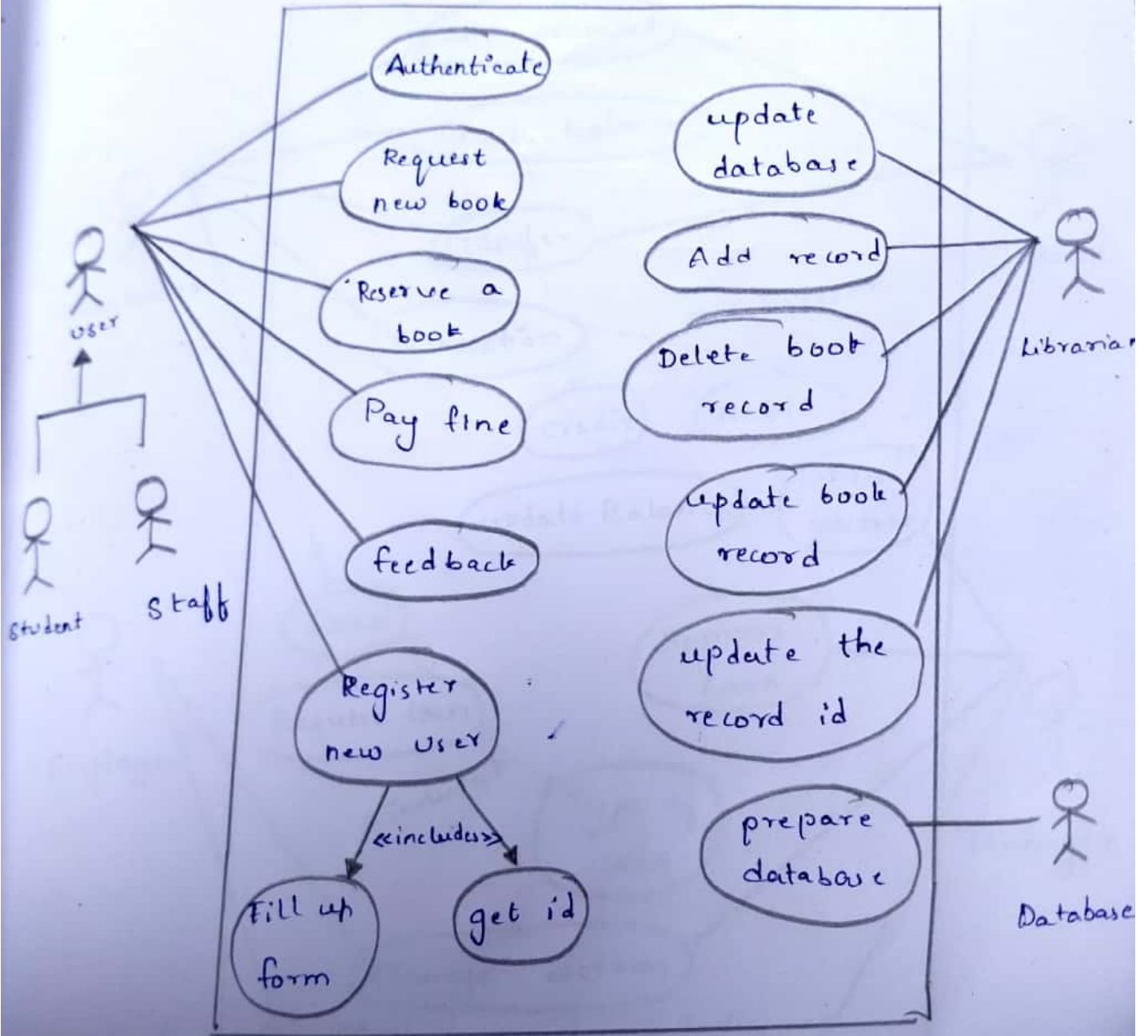
Association relation can be attach to provide additional information about the relationship. An association class is identical to other classes & can contain operations, attributes as well as other associations.

Register No: 19L31A0516

Experiment No: 3

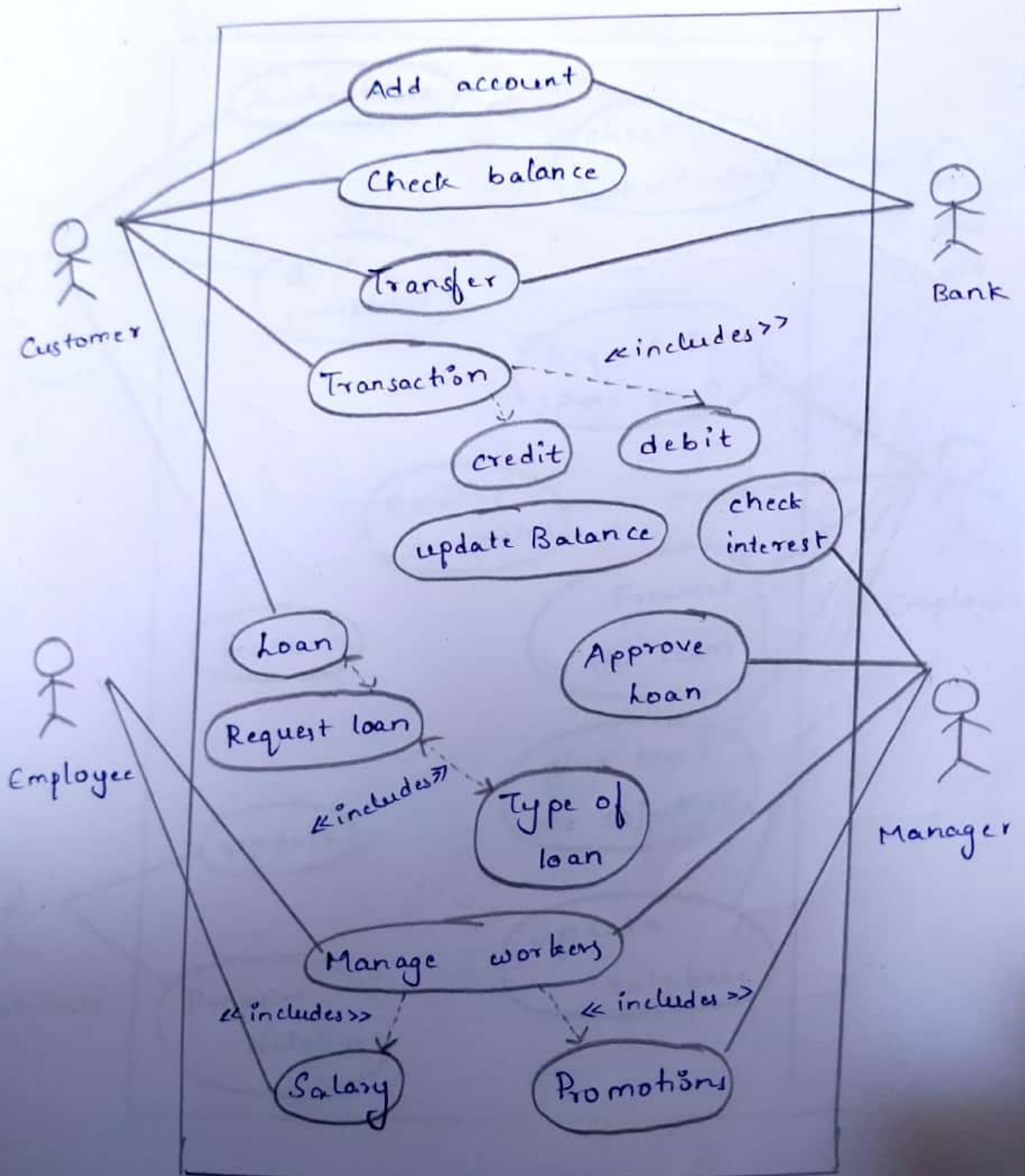
Date:

## Case Study 1: Library Management System

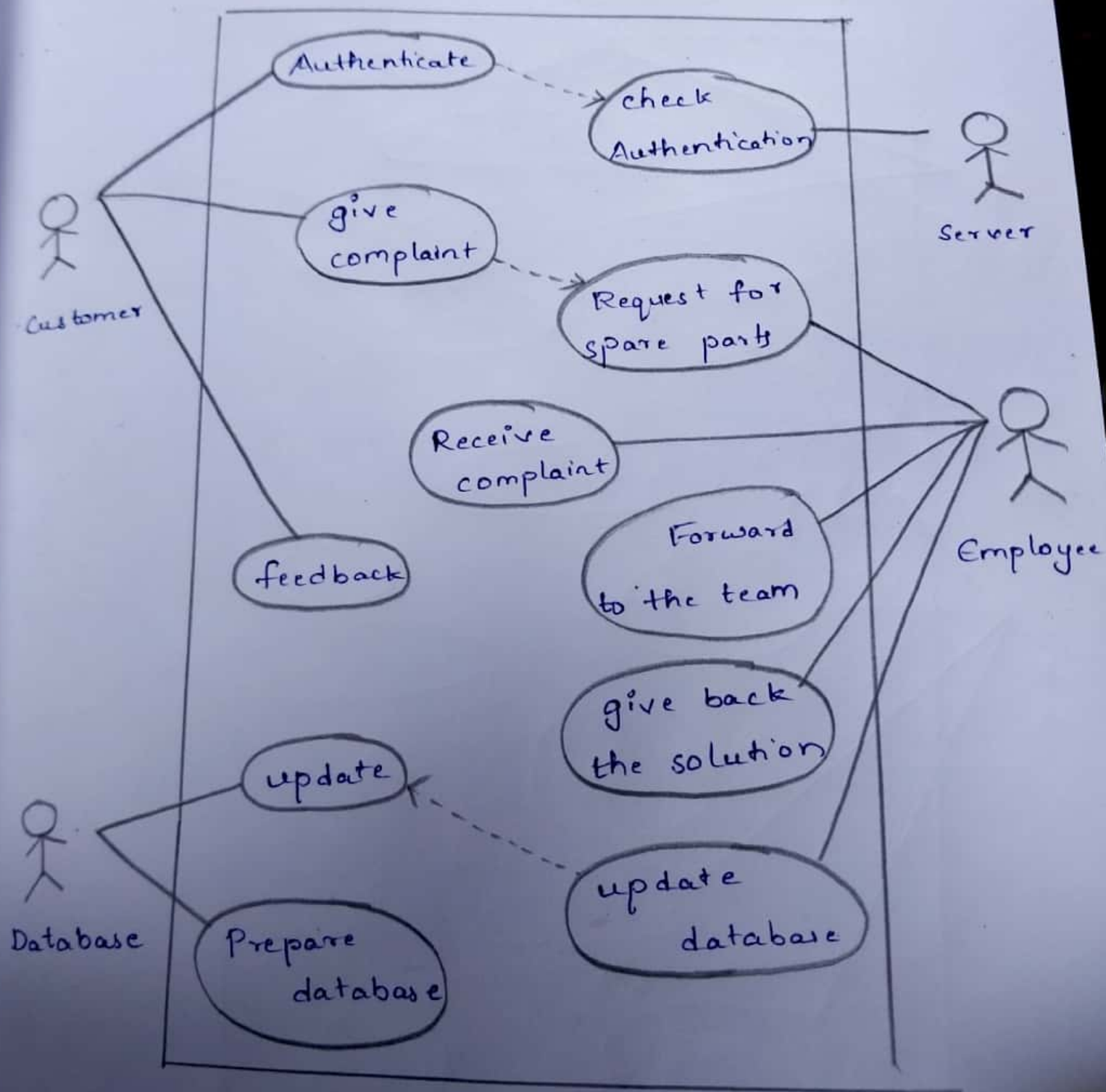




# Case Study 2 : Banking Application



Case Study 3: Customer Support System



Register No: 19L31AD516

Experiment No: 4

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
		Signature of the Lab teacher	
Date			

AIM: To construct UML sequence diagram for the following case studies

1. Library Management System
2. Banking Application
3. Customer Support system

Sequence Diagram:-

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part of the run time.

Notations of a Sequence diagram -

lifeline -

An individual participant in the sequence diagram is represented by a lifeline. It is positioned at top of the diagram.



Register No: 19L31A0516

Experiment No: 4

Date:

Lifeline

Actor:-

A role played by an entity that interacts with the subject is called as an actor. An actor may or may not represent a physical entity, but it purely depicts the role of an entity. Several distinct roles can be played by an actor or vice versa.



Actor

Activation:-

It is represented by a thin rectangle on the lifeline. It describes that time period in which an operation is performed by an element, such that the top & the bottom of the rectangle is associated with the initiation & the completion time each respectively.

Lifeline



Register No:

19L31A0516

Experiment No:

4

Date:

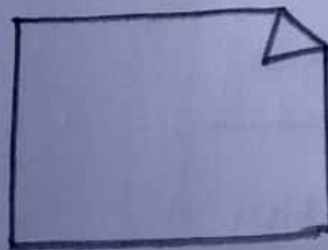
Messages:-

The messages depicts the information between the objects and are represented by arrows. They are in sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines.



Note -

A note is the capability of attaching several remarks to the element. It basically carries useful information for the modelers.

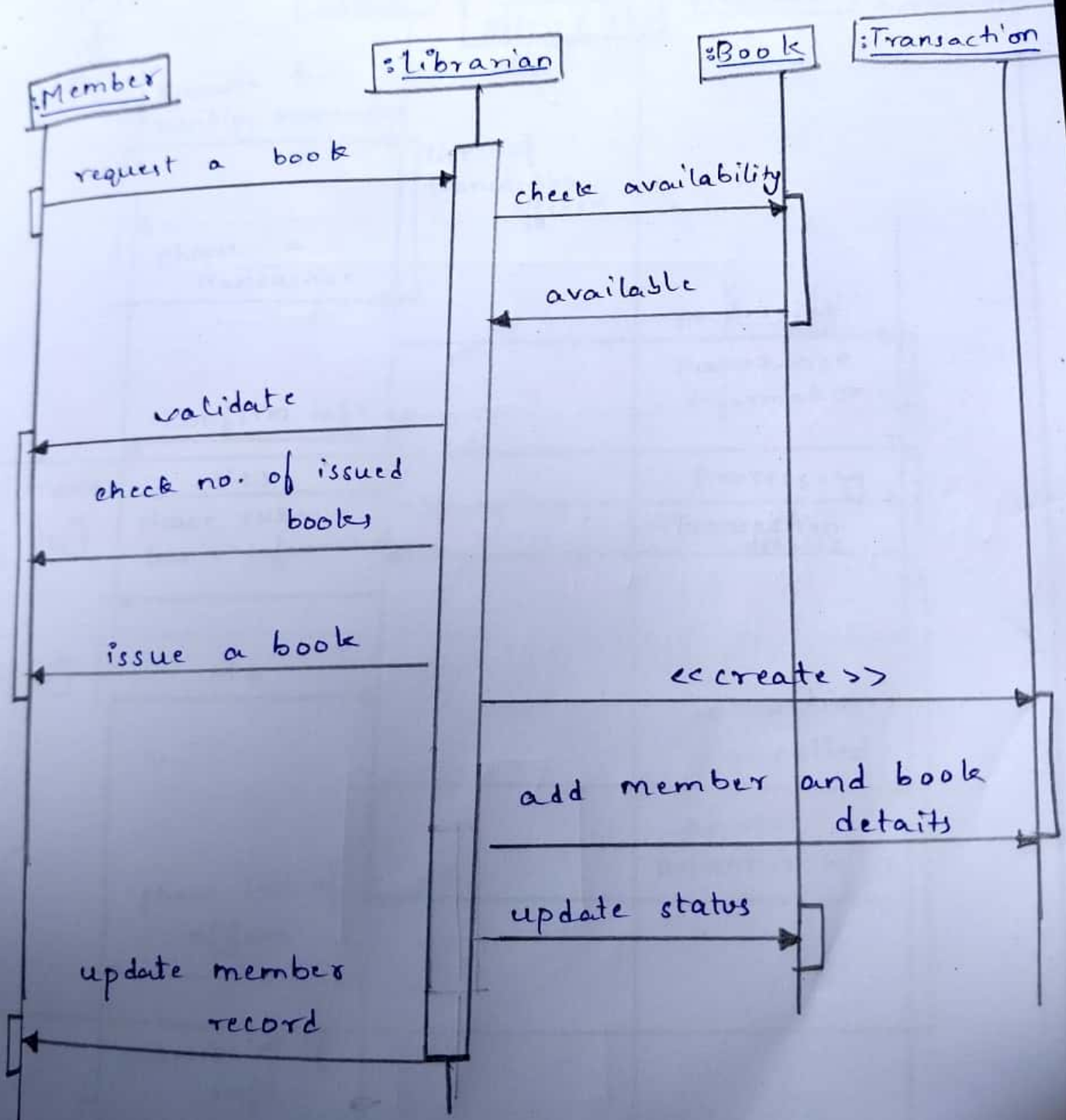


Register No: 19L31A0516

Experiment No: 4

Date:

## Base Study 1: Library Management System





Register No: 19131A0516

Experiment No: 4

Date:

## Case Study 2: Banking Application

Customer

:Banking  
system

:Banking  
Transaction  
offer db

:Transaction  
details

:Requirement  
database



Enquire for  
banking transaction

list of  
transaction  
offered

choose a  
transaction

processing

Transaction  
information

confirm info

Yes

Processing

please submit  
basic information

Transaction  
details

Alternative  
[if yes]

No

Transaction  
cancelled

[if no]

choose  
another offer

show list of  
offers

submission of  
info

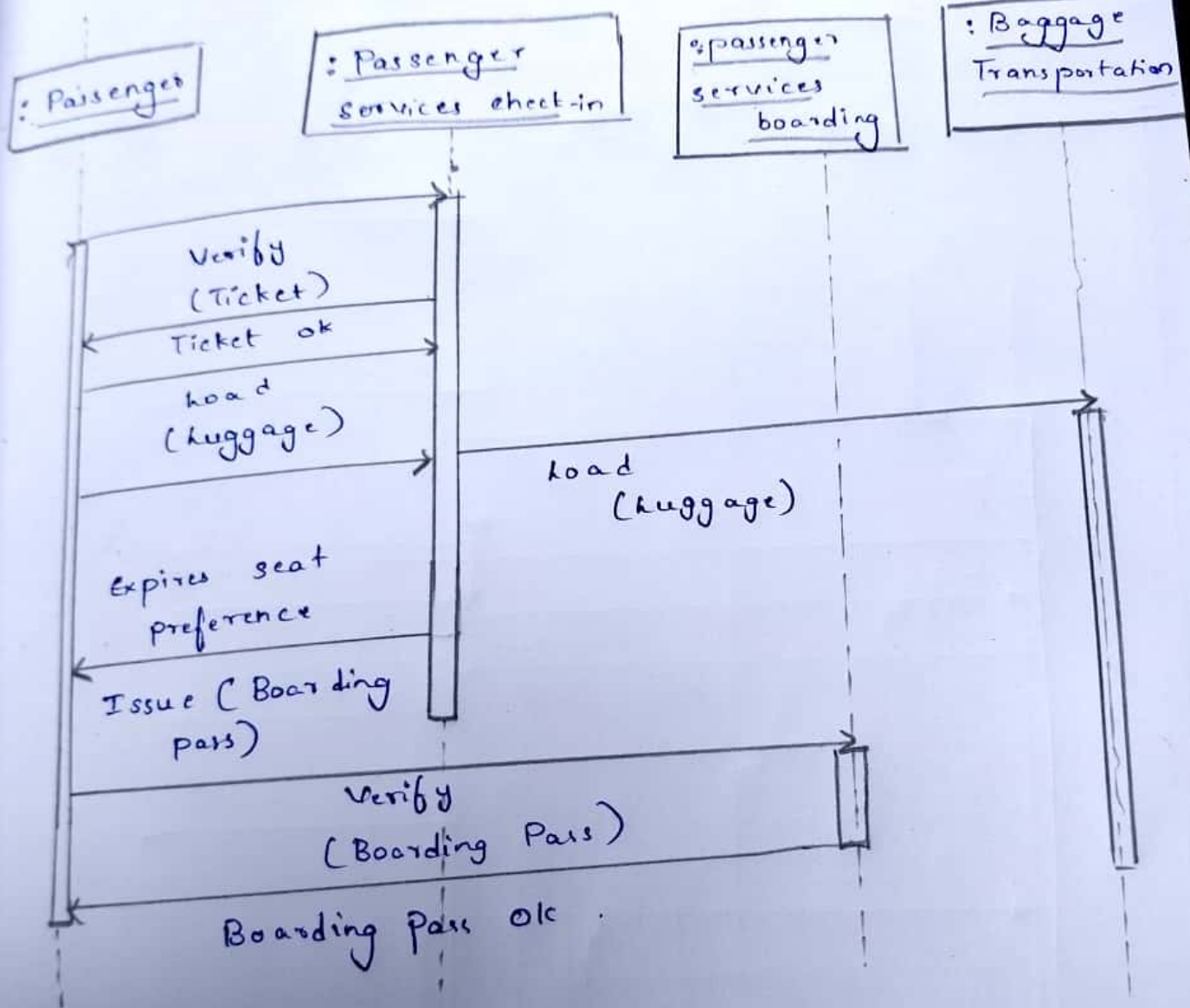
Processing

Processing

Transaction  
complete

Register No: 19L31A0516 Experiment No: 4 Date:

### Case Study 3: Customer Support System



Register No: 19L31A0516

Experiment No: 5

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
		Signature of the Lab teacher	
Date			

AIM: To construct UML collaboration diagram for the following case studies.

Case Study 1 :- Library Management System

Case Study 2 :- Banking application

Case Study 3 :- Customer Support System

Collaboration diagram :- The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information by differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. The collaboration diagram, which is also known as a communication diagram, is used to portray the objects' architecture in the system.

Notations of Collaboration diagram :-

Components :-

- objects
- Actors
- Links
- Messages

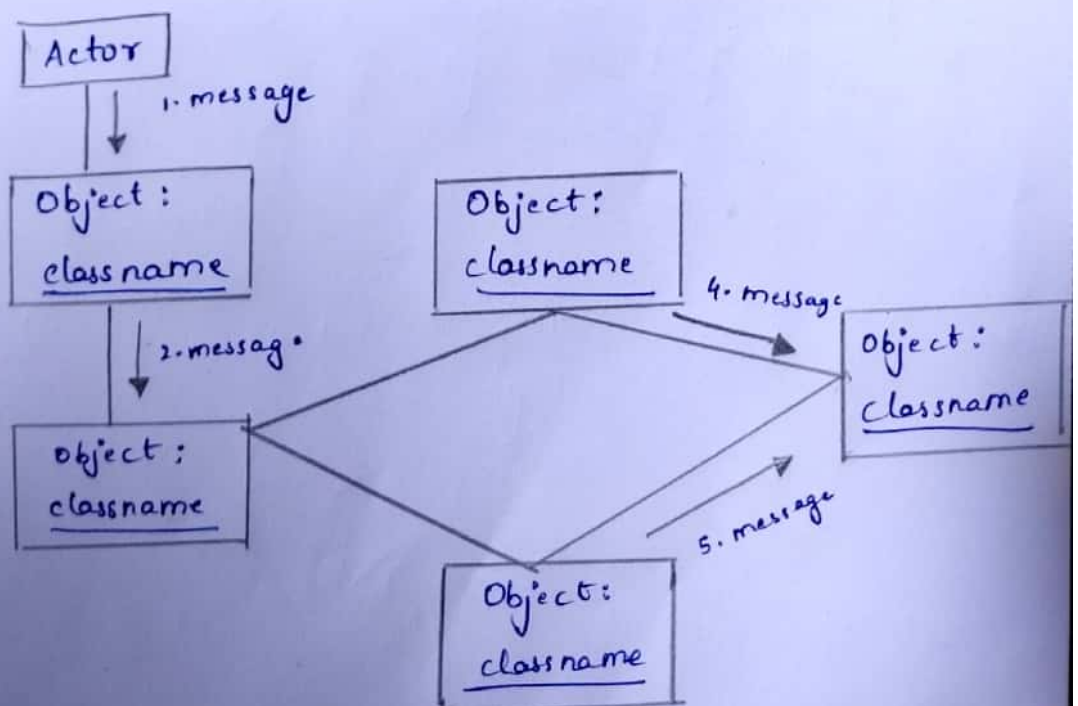


Objects:- The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. A class may constitute more than one object.

Actors:- In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective roles and name. In this, one actor initiates the use case.

Links:- The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line.

Messages:- It is the communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link.



Register No :

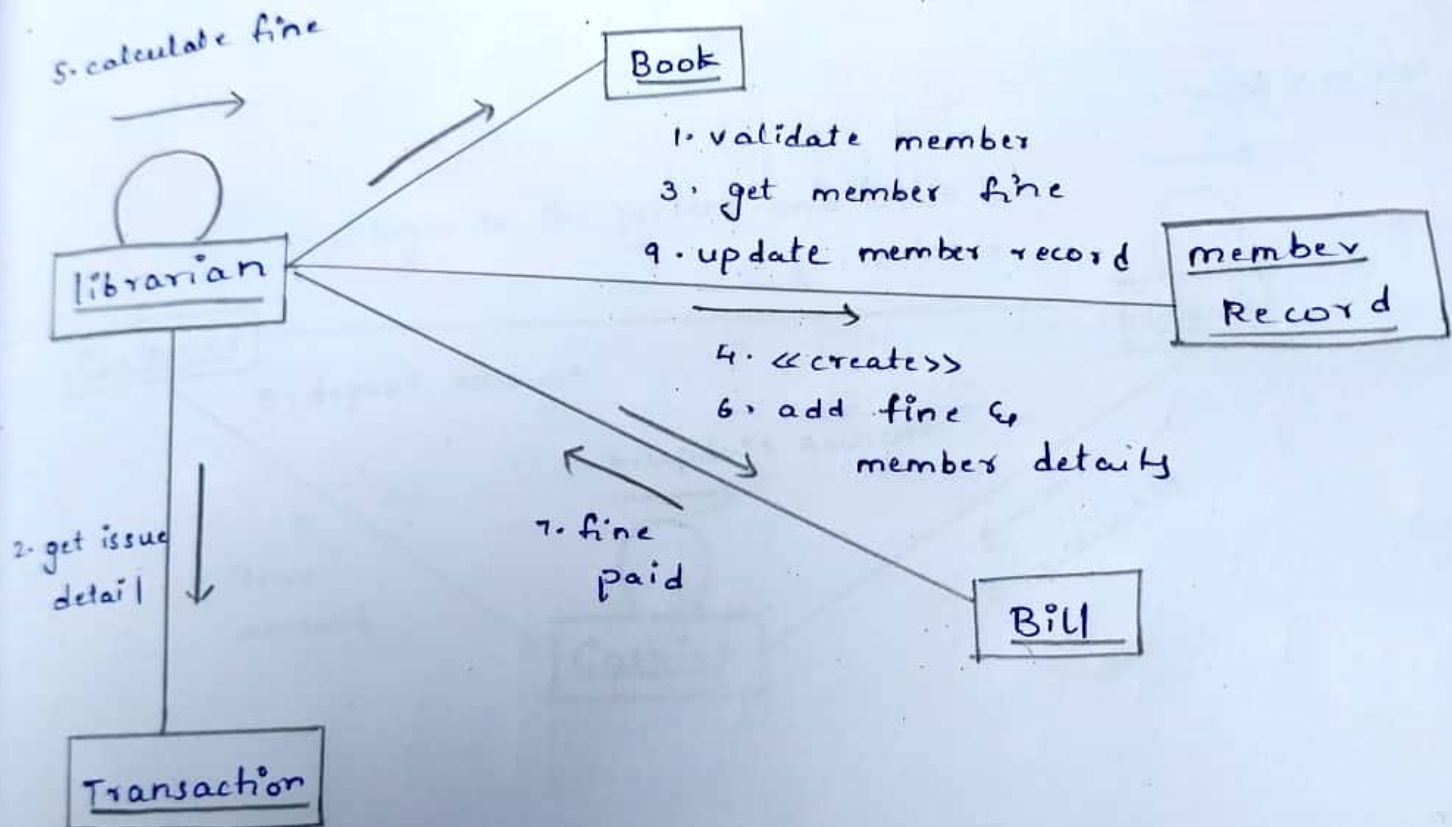
19L31A0516

Experiment No :

5

Date :

## Case Study 1 : Library Management System

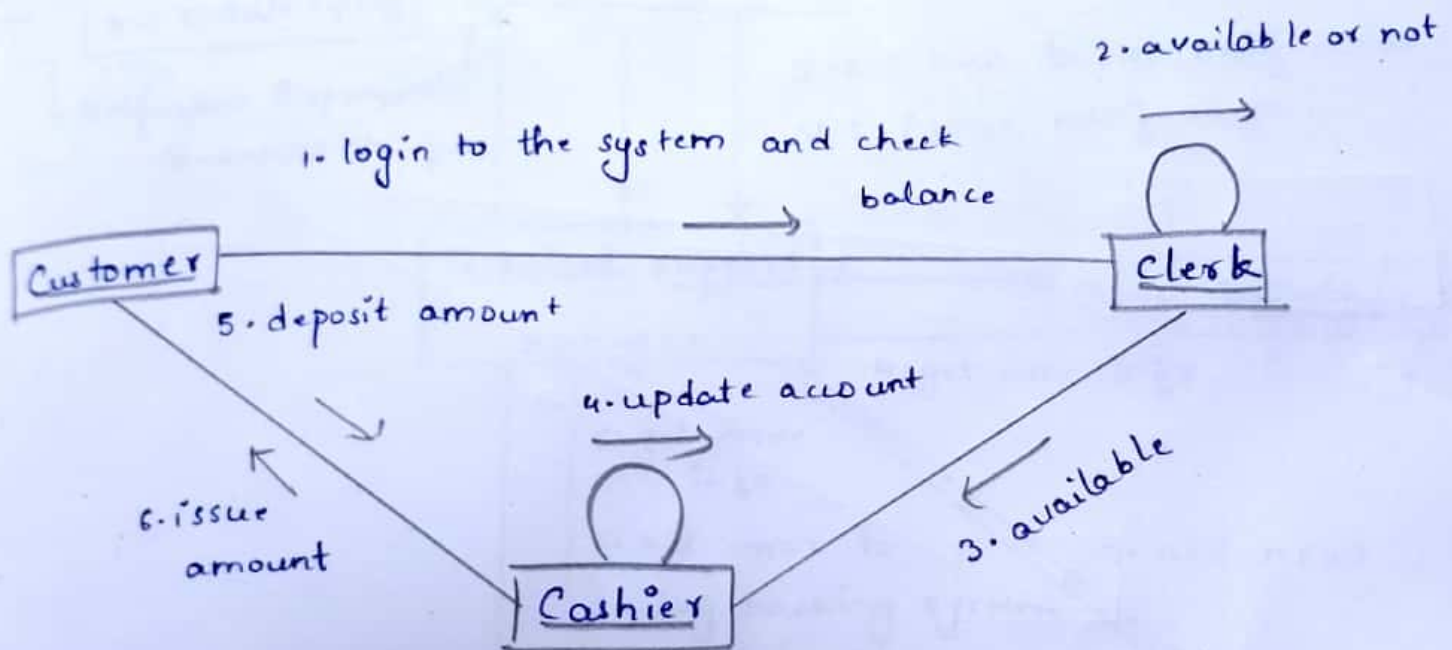


Register No: 19L31A0516

Experiment No: 5

Date:

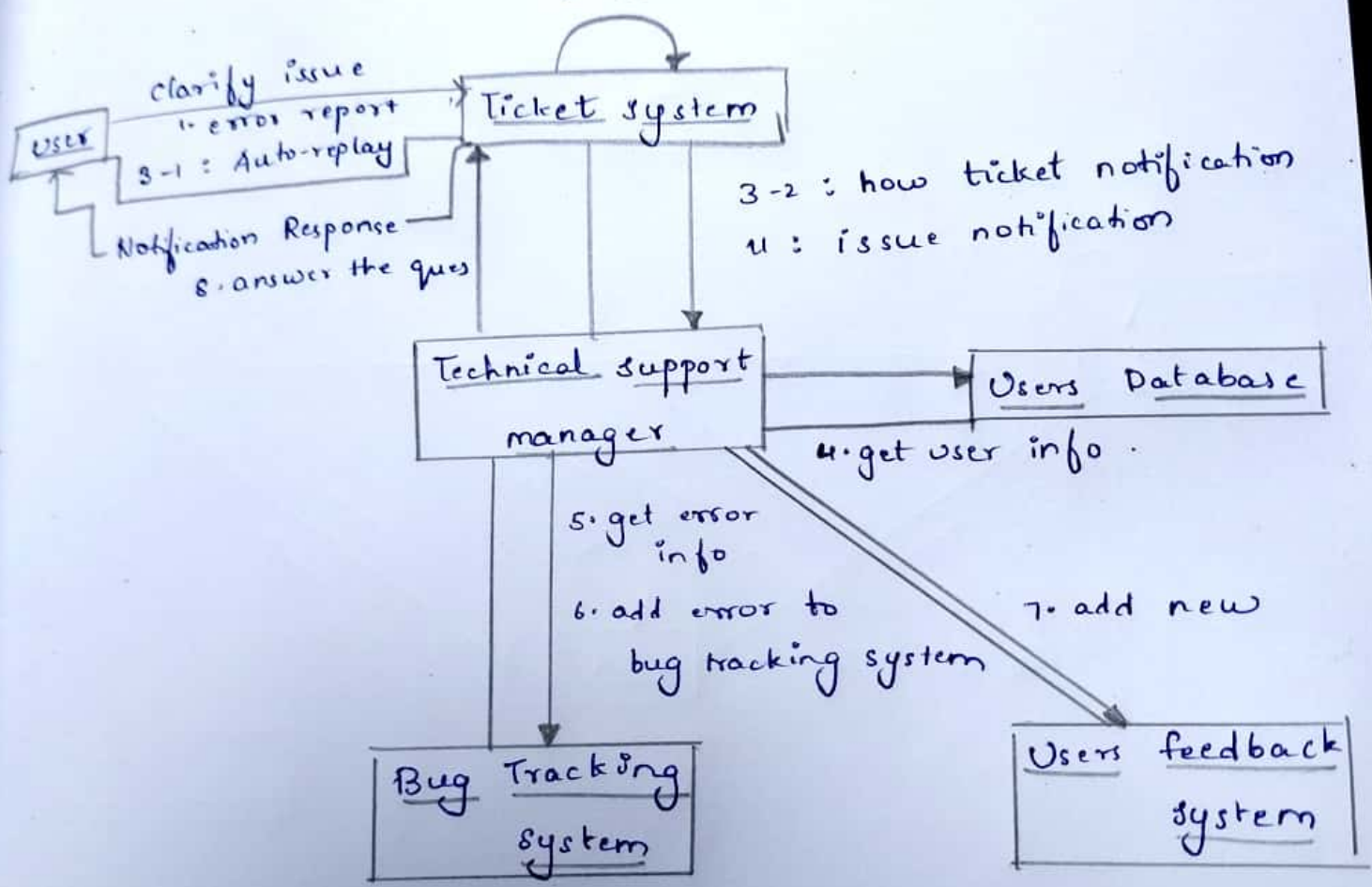
## Case Study 2: Banking application





Case Study 3: Customer Support System

2. record in database



Register No: 19L31A0516

Experiment No: 6

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
		Signature of the Lab teacher	
Date			

AIM: To construct UML activity diagram for the following case studies.

Case Study 1:- Library Management System.

Case Study 2:- Banking application.

Case Study 3:- Customer Support System.

Activity diagram:- In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow as the order in which it occurs. The flow can be sequential, branched or concurrent and to deal with such kinds of flows, the activity diagram has come with a fork, join etc.

Components of activity diagram:-

→ Activities

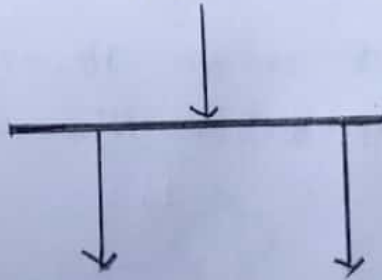
→ forks

→ join nodes

Activities:- The categorization of behaviour into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. The control flow of activity is represented by control nodes & object nodes that illustrates the object used within an activity.

### Activity

Forks:- Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters. Whenever a data is received at an inward edge, it gets copied & split crossways various outward edges.



Join nodes:- Join nodes are the opposite of fork nodes. A logical AND operation is performed on all of the inward edges as it synchronizes the flow of input across one signal output edge.



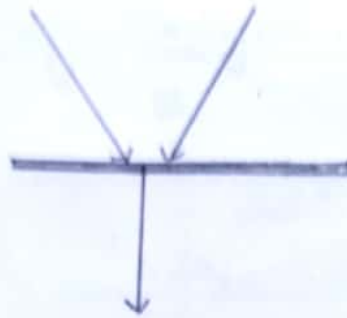
Register No:

18131A0316

Experiment No:

6

Date:



Notation of an activity diagram:-

Initial state:- It depicts the initial stage or beginning of the set of actions



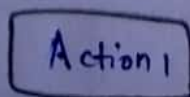
Final state:- It is the stage where all the control flow and object flows end.



Decision Box:- It makes sure that the control flow or object flow will follow only one path.



Action Box:- It represents the set of actions that are to be performed.

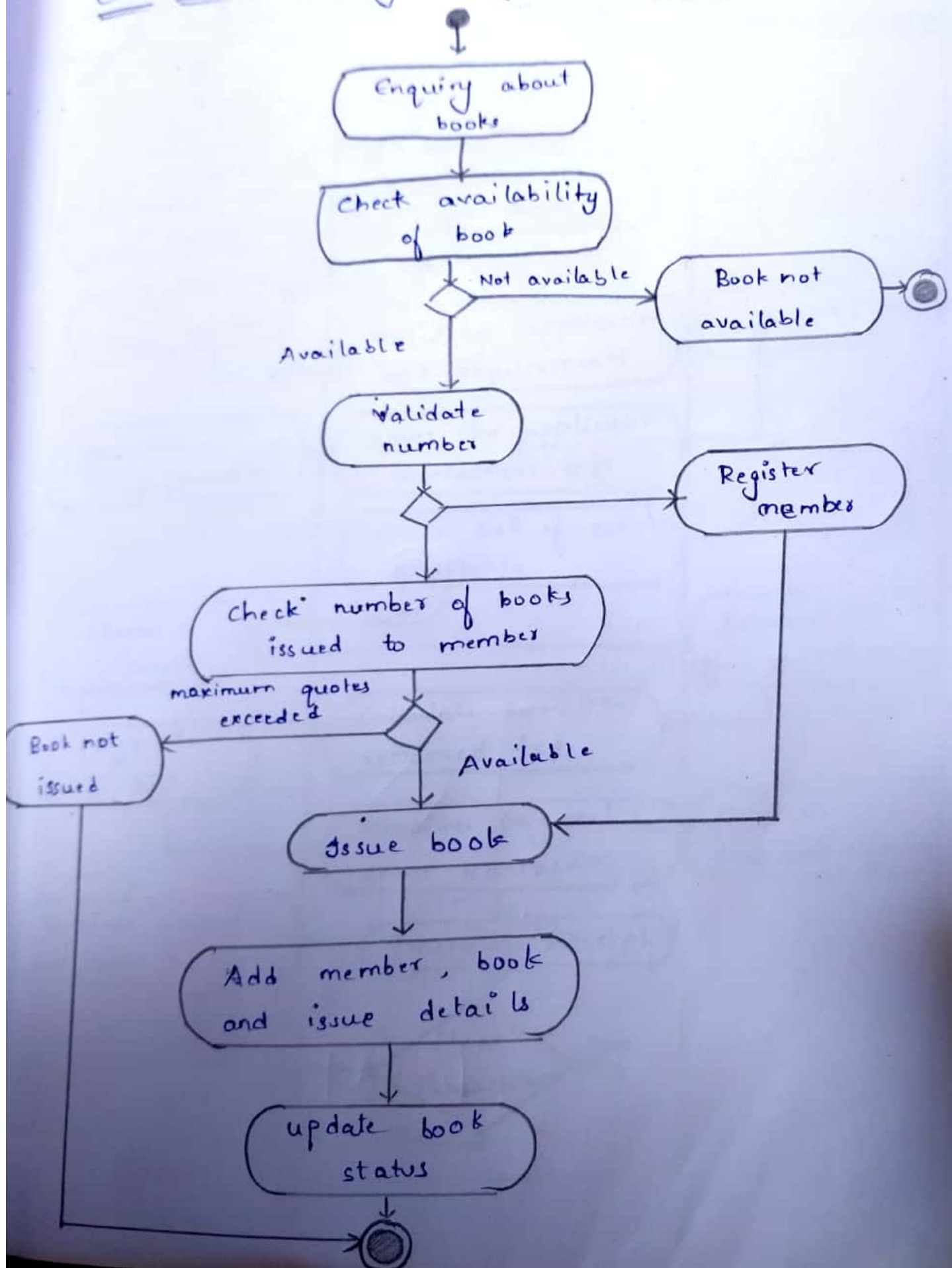


Register No: 1AL31A0514

Experiment No: 6

Date:

## Case Study 1: Library Management System

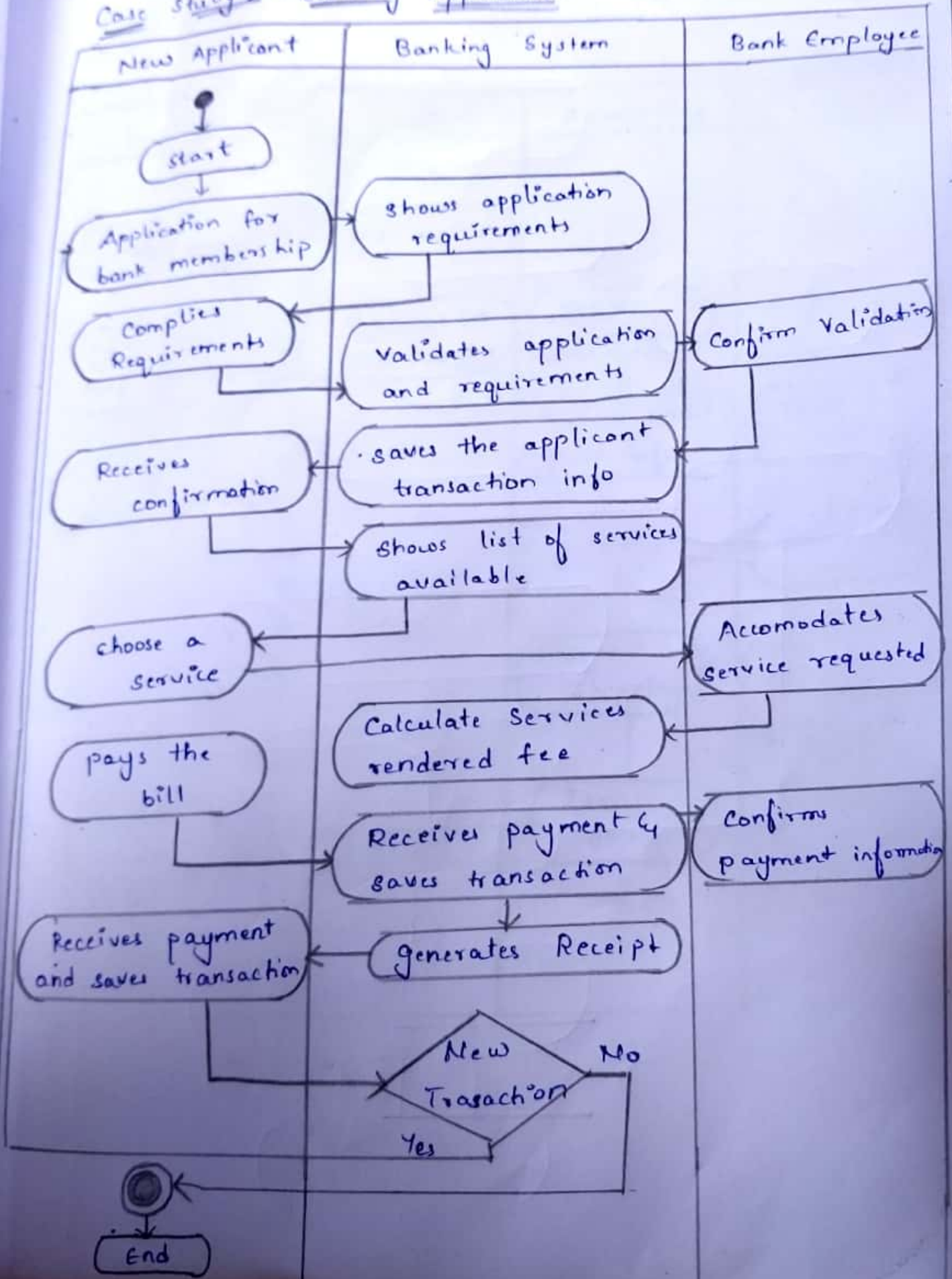


Register No: 19L31A0514

Experiment No: 6

Date:

Case Study 2: Banking Application





ster No: 19L31A0516 Experiment No: 6 Date:

# Case Study 3: Customer Support System

