EXPERIMENT : No. _____ Page No. Date

PRACTICAL 1:- Study and implementation of class diagrams.

COMPONENTS

CLASS:-

- A class represents a relevant concept from the domain, a set of persons, objects, or ideas that are depicted in the IT system :
- Examples of classes are passengers, plane or tickets.

Class
Attribute

ATTRIBUTE:-

- An attribute of a class represents a characteristic of a class that is of interest for the user of the IT system.
- Characteristics of interest of a passenger, for example, are name and age.

Class
Attribute

Generalization:

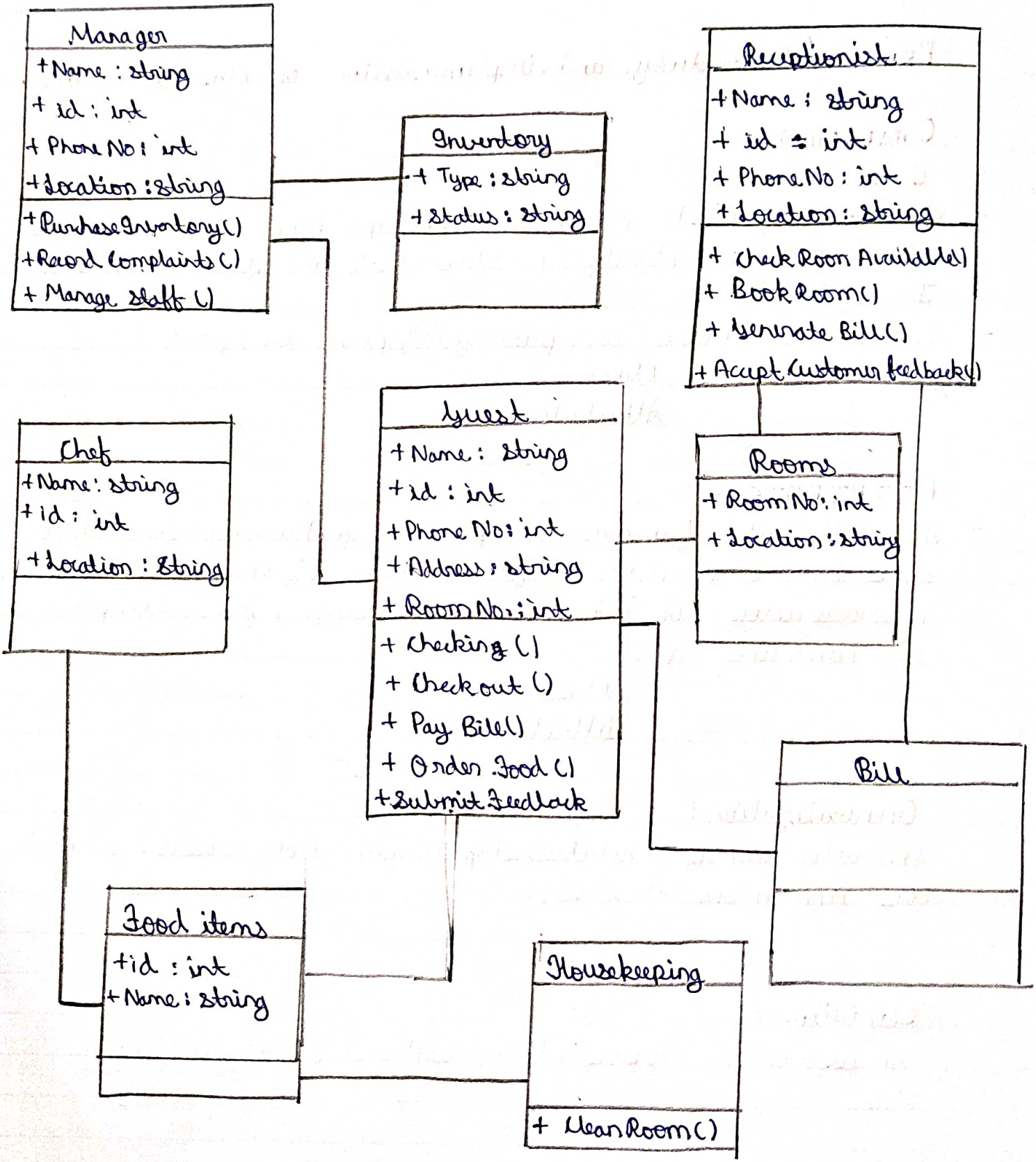
Generalization is a relationship between two classes: general class and a special class:

Association:-

An association represents a relationship between two class:

Sundaram

Teacher's Sign. : _____



Class Diagram of Hotel Management

EXPERIMENT : No.

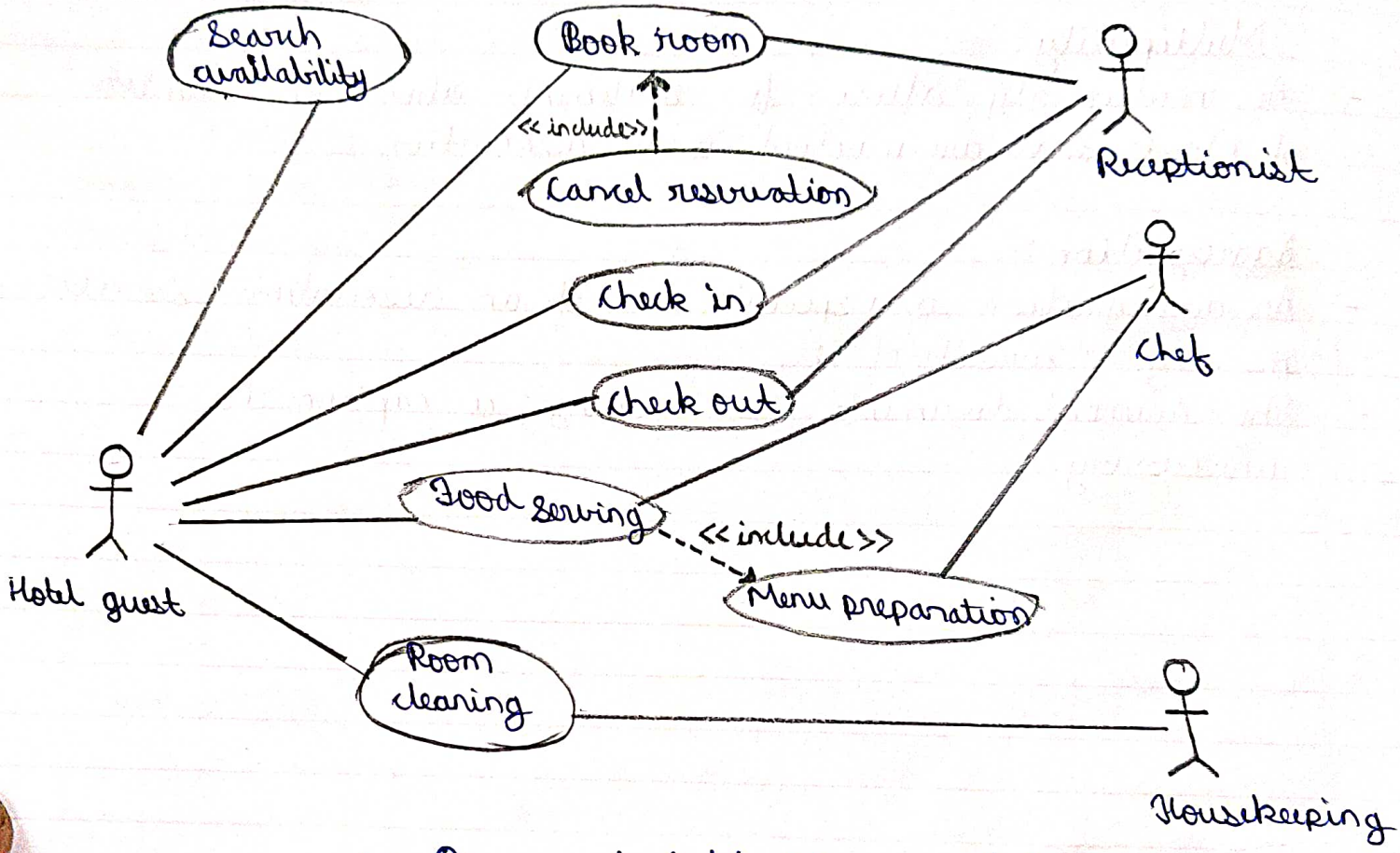
Page No.
Date

Multiplicity :

- A multiplicity allows for statements about the number of objects that are involved in an association.

Aggregation :-

- An aggregation is a special case of an association (see above) meaning "consists of" :
- The diamond documents this meaning ; a caption is unnecessary.



Case Diagram of Hotel Management System

EXPERIMENT : No.

Page No.
Date

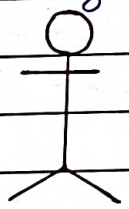
PRACTICAL 2: Study and implimentation of Use Case Diagrams

- A use case diagram is a graphic depiction of the interaction among the elements of a system.
- A use case is a methodology used in system analysis to identify, clarify and organize system requirements.
- The actors, usually individuals involved with the system defined according to their roles.

COMPONENTS :-

Actors :

- You can picture an actor as a user of the IT system, for example Mr. Steel or Mrs. Smith from check-in.
- Because individual persons are irrelevant for the model, they are abstracted. So the actors are called "check-in employee" or "passenger".



Use Case :

- Use cases describe the interactions that take place between actors and IT systems during the execution of business processes.

Use case

- A use case represents a part of the functionality of the IT system and enables the user (modeled as an actor) to access this functionality.

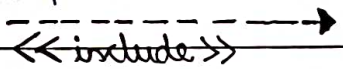
EXPERIMENT : No. _____ Page No. Date

Association

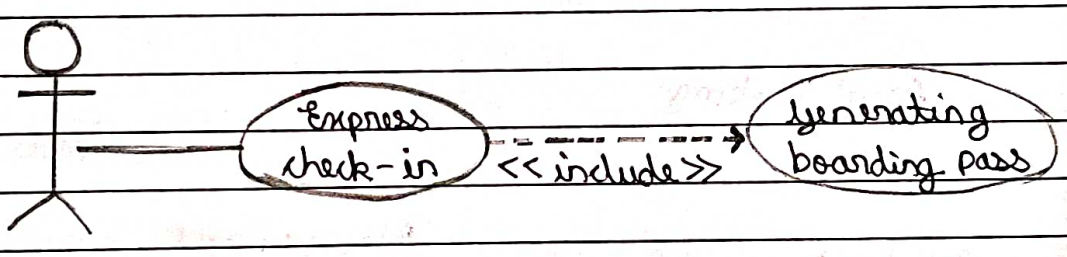
An association is a connection between an actor and a use case. An association indicates that an actor can carry out a use case. Several actors at one use case mean that use case on his or her own and not that the actors carry out the use case together.

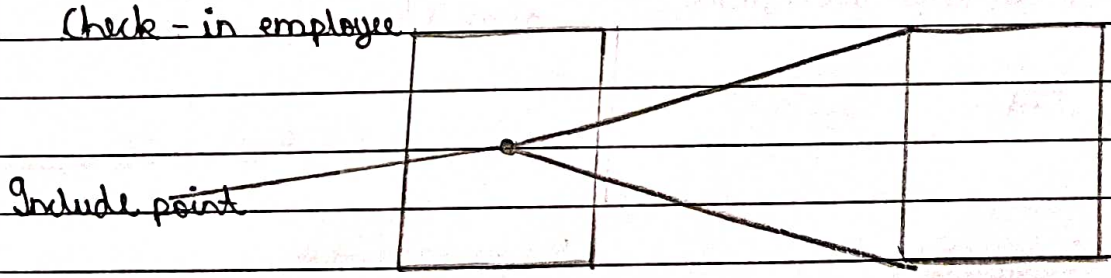
Include Relationships

An include relationship is a relationship between two use cases:



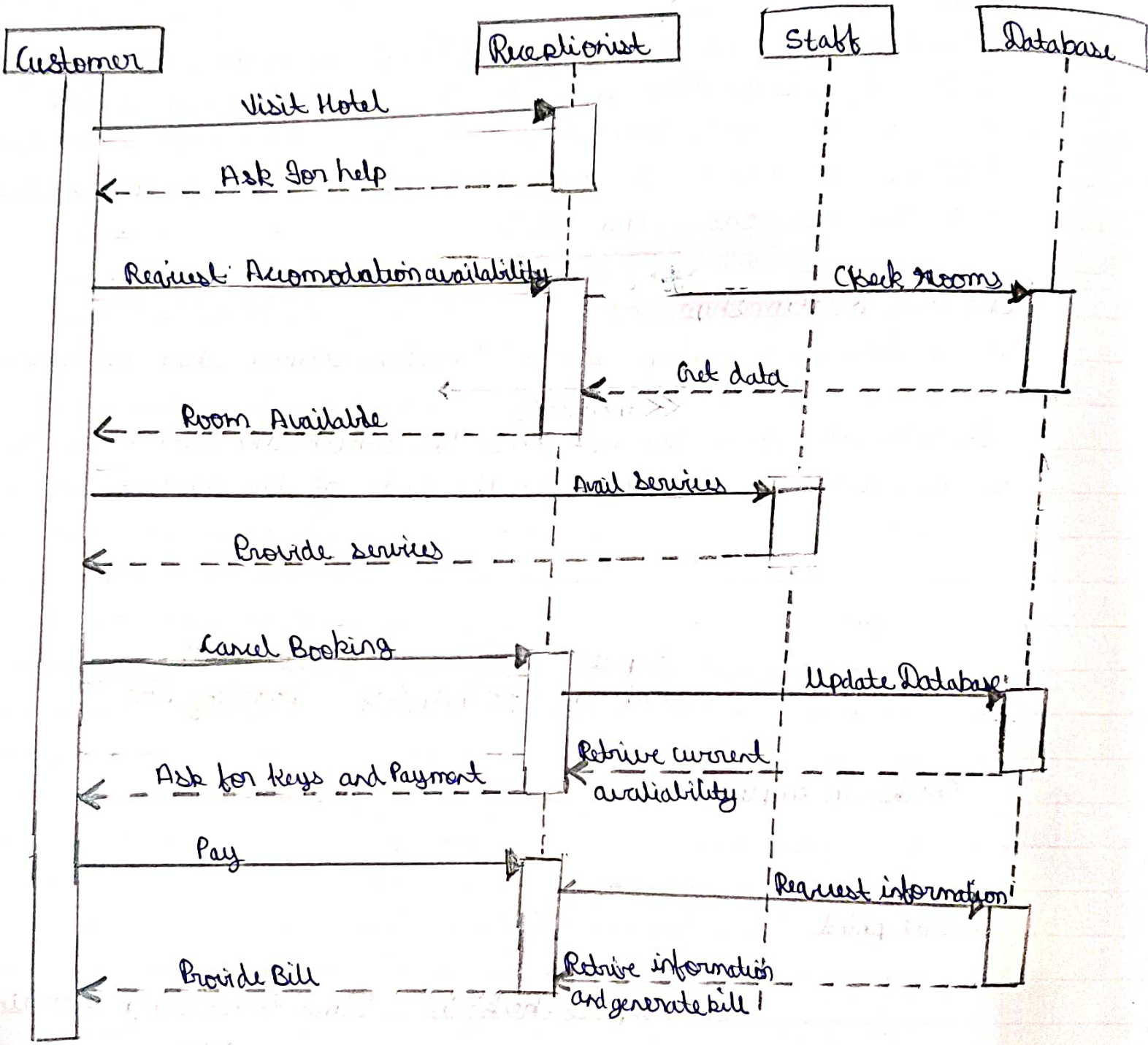
It indicates that the use case to which the arrow points is included in the use case on the side of the arrow.





Flow express check-in Flow Generating boarding pass

Teacher's Sign. : _____



Sequence Diagram for Hotel Management System

EXPERIMENT : No.

Page No.
Date

PRACTICAL 3 :- Study and implementation of Sequence Diagrams.

The sequence diagram is a good diagram to use to document a system's requirements and to flush out a system's design. The reason the sequence diagram is so useful is because it shows the interaction logic between the objects in the system in the time order that the interactions take place.

COMPONENTS:-

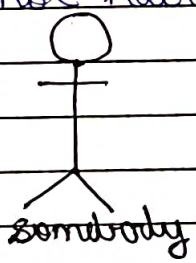
Comment:-

- The flow of a mutation event is documented with a combination of textual description and a sequence diagram.
- In comments, the flow logic is shown on the topmost level.

Delete Flight Number
Address Machine
...

Actor "Somebody"

- The actor "somebody" represents any actor from the use case diagram. Since the mutation event that is documented in a sequence diagram can be contained in several use cases, and since these use cases can have different actors, we use the actor somebody.
- This way, we do not have to decide on one, specific actor.



Sundaram

Teacher's Sign. : _____

EXPERIMENT : No.

Page No.
Date

Object :

- An object represents any object, meaning and undefined object of a class of the IT system.

- An iteration indicates that all objects to which a relationship exists receive the event, for example all the flights of a flight number.

Lifeline :

- The lifeline of an object represents a life (over the course of time). The rectangle, meaning the "thick part" of the lifeline shows when the object is active.

EXPERIMENT : No. _____ Page No. Date

PRACTICAL 4 : Study and implementation of State Transition Diagrams.

A state diagram is a diagram used in computer science to describe the behavior of a system considering all the possible states of an object when an event occurs. This behavior is represented and analyzed in a series of events that occurs in one or more possible states.

Components:

Initial State : The initial state represents the source of all objects.

It is not a normal state, because objects in this state do not yet exist.

State

- The state of an object is always determined by its attributes and associations. States in state chart diagrams represent a set of those value combinations, in which an object behaves the same in response to events:

State

- Therefore, not every modification of an attribute leads to a new state.

Transition

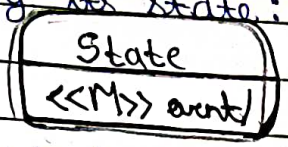
A transition represents the change from one state to another.

EXPERIMENT : No.

Page No.
Date

Internal Transition

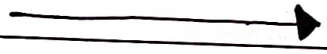
- An internal transition is a transition from one state to itself. This means that the object handles the event without changing its state:



- The events that initiate the internal transition are listed in the lower part of the state symbol. For instance, a frequent flyer card object in the state normal remains in the state normal when the event <<M>> add miles occurs

Mutation event

A mutation event in the initiator of a transition from one state to another, or for an internal transition, where the state remains the same:



Action

An action is the activity of an object that is initiated by an event:

An action describes what the object does in response to the event. This description can be textual or formalized.

<<M>> Event / Action

EXPERIMENT : No.

Page No.
Date

Guard condition:

- A guard condition is a condition that has to be met in order to enable the transition to which it belongs.
- Guard conditions can be used to document that a certain event, depending on the condition, can lead to different transitions.

[Guard Condition]

Final State :

- The final state represents the end of an object's existence.
- The final state is not a real state, because objects in this state do not exist anymore.

EXPERIMENT : No.

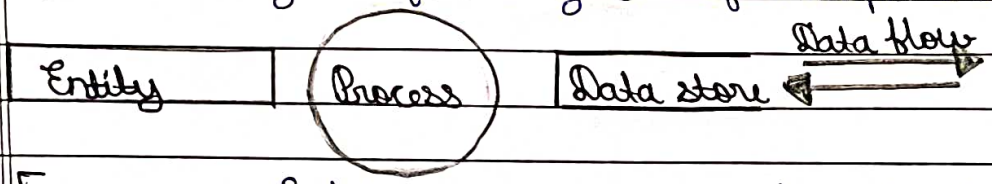
Page No.
Date

PRACTICAL 5:- Study and implementation of Data Flow Diagrams.

A data flow diagram (or DFD) is a graphical representation of the flow of data through an information system. It shows how information is input to and output from the system, the sources and destinations and where that information is stored.

COMPONENTS:-

DFD can represent source, destination, storage and flow of data using the following set of components -



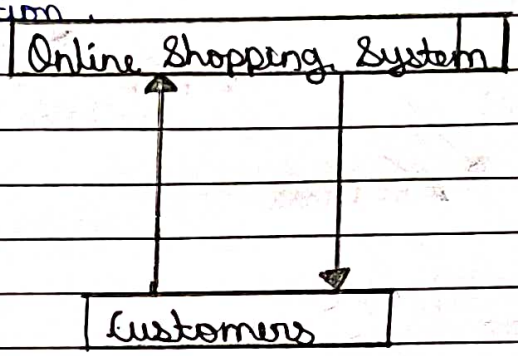
- ENTITIES: Entities are source and destination of information data. Entities are represented by a rectangles with their respective names.
- PROCESS: Its Activities and action taken on the data are represented by Circle or Round-edged rectangles.
- DATA STORAGE: There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.
- DATA FLOW: Movement of data is shown by pointed arrow. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

EXPERIMENT : No.

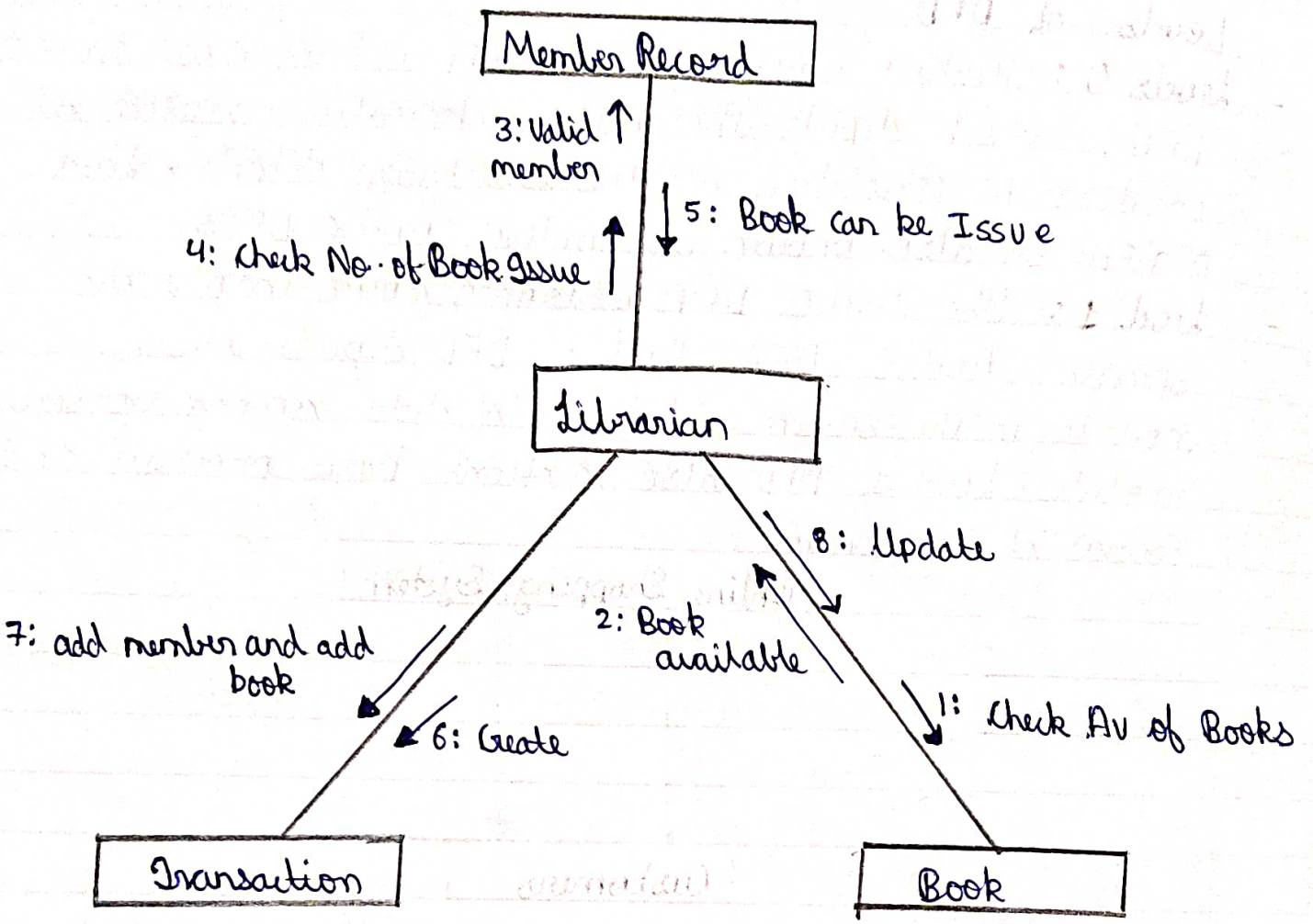
Page No.
Date

Levels of DFD

- Levels 0 : Highest abstraction level DFD is known as level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.
- Level 1 : The level 0 DFD is broken down into more specific, level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.



- Level 2 : At this level, DFD show data flows inside the modules mentioned in level 1. Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.



Collaboration diagram for Libary management

EXPERIMENT : No. Date

PRACTICAL 6: Study and implementation of Collaboration Diagrams.

Collaboration diagram also called a communication diagram or interation diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling language (UML).

COMPONENTS:

- **OBJECT:-**
The interaction between objects takes place in a system. An object is depicted by a rectangle with the name of the object, preceded by a colon and underline.
- **RELATION / ASSOCIATION:**
Association among objects is linked by connecting them. The cardinality can be depicted by placing qualifiers on either ends.
- **MESSAGES:**
An arrow that commencing from one object to the destination object. This dipicts the interaction between objects. The sequence or order of the interaction is depicted by the number.

Sundaram

Teacher's Sign. : _____

EXPERIMENT : No.

Page No.
Date

PRACTICAL 7: Study and implementation of Component Diagrams.

UML Component Diagrams. Component diagram shows components, provided and required interfaces, ports and relationships between them. This type of digrams is used in Component-Based Development (CBD) to describe systems with Service-Oriented Architecture (SOA).

COMPONENTS:

NODES:

Nodes are hardware or software objects, which are of a bit higher level than components. Boxes represent nodes in Lucidchart.

INTERFACE:

Show input or materials that a component either receives or provides. Interfaces can be represented with textual notes or symbols such as the lollipop, socket and ball-and-socket shape

PORT:

Symbolized with a small square, ports specify a separate interaction point between the component and the environment

Sundaram

Teacher's Sign. : _____

EXPERIMENT : No. _____ Page No. Date

PACKAGE:
Groups together multiple elements of the system. Packages are represented by file folders in lucidchart. Just as file folders group together multiple sheets, packages can be drawn around several components.

DEPENDENCY:
Show that one part of your system depends on another. Dependencies are represented by dashed lines linking one component (or element) to another.

Teacher's Sign. : _____

EXPERIMENT : No.

Page No.
Date

PRACTICAL 8:- Study and implementation of Deployment Diagrams.

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to development targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

COMPONENTS:

- **ARTIFACT:**
  A product develop by the software, symbolized by a rectangle with the name and the word "artifact" enclosed by double arrows.

- **ASSOCIATION:**
  A line that indicates a message or other type of communication between nodes.

- **COMPONENT:**
  A rectangle with two tabs that indicates a software element.

- **DEPENDENCY:**
  A dashed line that ends in a arrow, which indicates that one node or component is dependent on another.

Sundaram

Teacher's Sign. : _____

EXPERIMENT : No.

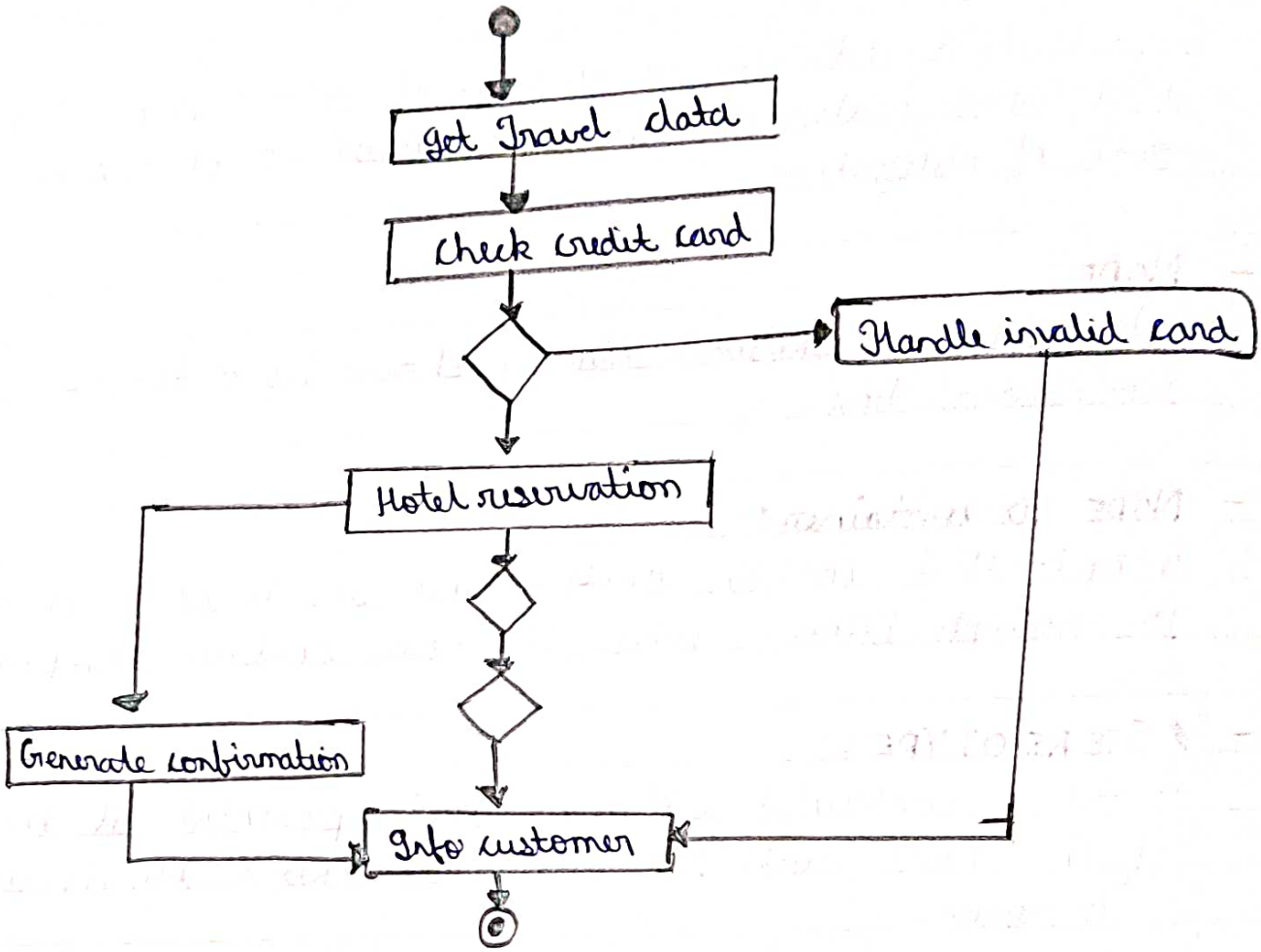
Page No.
Date

- INTERFACE:
A circle that indicates a contractual relationship; those objet that realize the interface must complete some sort of obligation.

- NODE:
A hardware or software objet, shown by a three-dimensional box.

- NODE as container:
A node that contains another node inside of it such as in the example below, where the nodes contain components.

- STEREOTYPE:
A device contained within the node, presented at the top of the node, with the name bracketed by double arrows.


Get Travel data
Check credit card
Handle invalid card
Hotel reservation
Generate confirmation
Info customer

EXPERIMENT : No. _____ Page No. Date

PRACTICAL 9: Study and implementation of Activity Diagrams.

- Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.
- Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

COMPONENTS :-

INITIAL NODE:

- The initial node is the starting point of an activity. An activity can have more than one initial node; in this case several flows start at the beginning of an activity:

- It is also possible that an activity has no initial node, but is initiated by an event (action : accepting an event).

Activity Final Node:

- The activity final node indicates that an activity is completed. An activity diagram can have more than one exist in the form of activity final nodes:

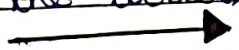
- If several parallel flows are present within an activity, all flows are stopped at the time the activity final node is reached.

EXPERIMENT : No.

Page No.
Date

Edge (Control flow)

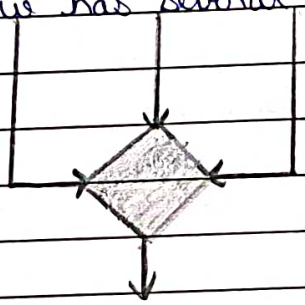
- Edges, represented by arrows, connect the individual components of activity diagrams and illustrate the control flow of the activity.



- Within the control flow an incoming arrow starts a single step of an activity. after the step is completed the flow continues along the out going arrow. A name can be attached to an edge (close to the arrow)

Merge Node:

The diamond below has several inputs and only one output:



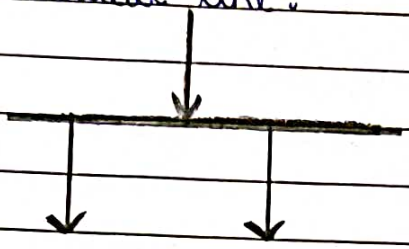
Its purpose is the merging of flows. The inputs are not synchronized; if a flow reaches such a node it proceeds at the output without waiting for the arrival of other flows.

EXPERIMENT : No.

Page No.
Date

Fork :

For the branching of flows in two or more parallel flows we use a synchronization bar, which is depicted as a thick horizontal or vertical line.



Branching allows parallel flows within activities. A fork has no one input and two or more outputs.

Action :

- An action is an individual step within an activity, for example, a calculation step that is not deconstructed any four further. That does not necessarily mean that the action cannot be subdivided in the real world, but in this diagram will not be refined any further.

(Action)