

24th | Jan. | 2022

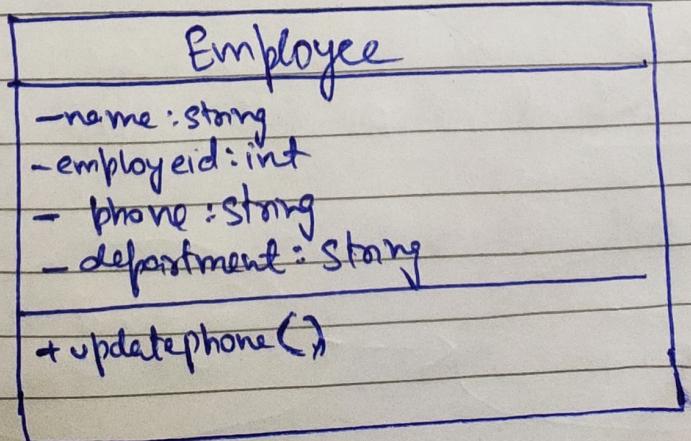
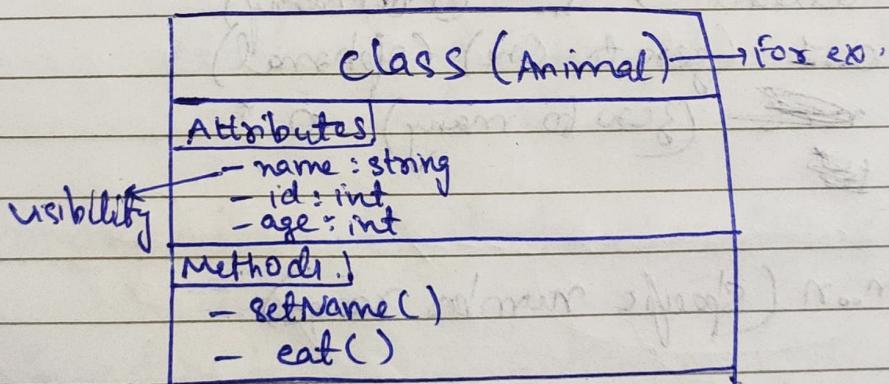
Date 1/1
Page _____
Puffin

Module -3 - UML CLASS DIAGRAM

(By UDAY)

Basics :-

- * Attributes :- A significant piece of data containing values that describe each instance of that class.
→ Also called as fields, Variables, properties.
- * Methods :- Also called operations or functions. Allow you to specify any behavioural feature of a class.
- * Visibility :- (-) private, (+) public, (*) protected, (~) package/default.

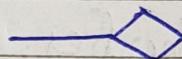


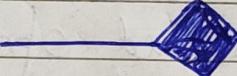
Relationships

D → Don't
 R → Repeat
 Y → Yourself.

→ * inheritance →

* Association —

* Aggregation — 

* Composition — 

Multiplicity:

- ~~o~~ specific number (n)
- at least → 1..* (many)
- 0..1 (zero to one) (optional)
- ~~o~~ (zero to many) 0..*
- m..n (specific number range)

UML Use Case Diagram Tutorial

We can breakdown use case diagram into few different elements

- Systems
- Actors
- Use cases
- Relationships

Systems. (whatever you are developing.)

It can be your website, software component,
~~or~~ business process application etc.

or any no. of other things.

→ Actor (eg customer that are going to user
your Software or your product)
(Person, an organization, another system,
or an external device)

* Primary Actor (Initiates the Use of System)
(Customer)

* Secondary Actor - Reactions (only after
customer proceed
with information)

→ Primary Actor should be on the left of the
System.

→ Secondary Actor should be on the right of
the System.

⇒ Use Case oval

Represents an action that accomplishes some sort of task within the System.

Final Element in the UML diagram
IT 2

⇒ Relationships

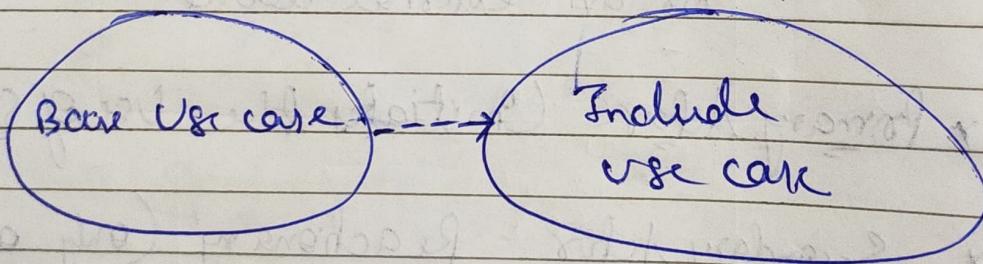
→ (solid line) → Association

→ Include (---→) <<include>>

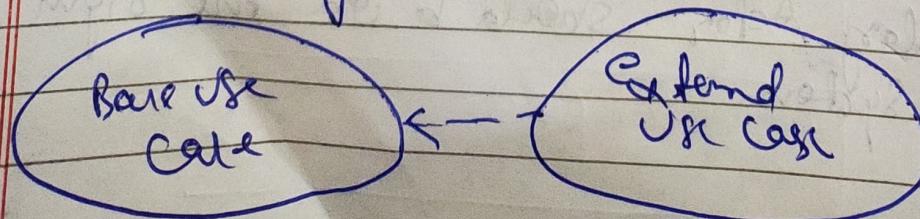
→ Extend (<---) <<extend>>

→ Generalization (→) from ~~base~~ children to parent
(also known as inheritance)

* ~~Include~~ & Every time the base use case is ~~executed~~, the include use case is executed as well



* Extend: when the base use case is executed, the extend use case will happen sometimes but not every



<< >>
(Double chevrons).

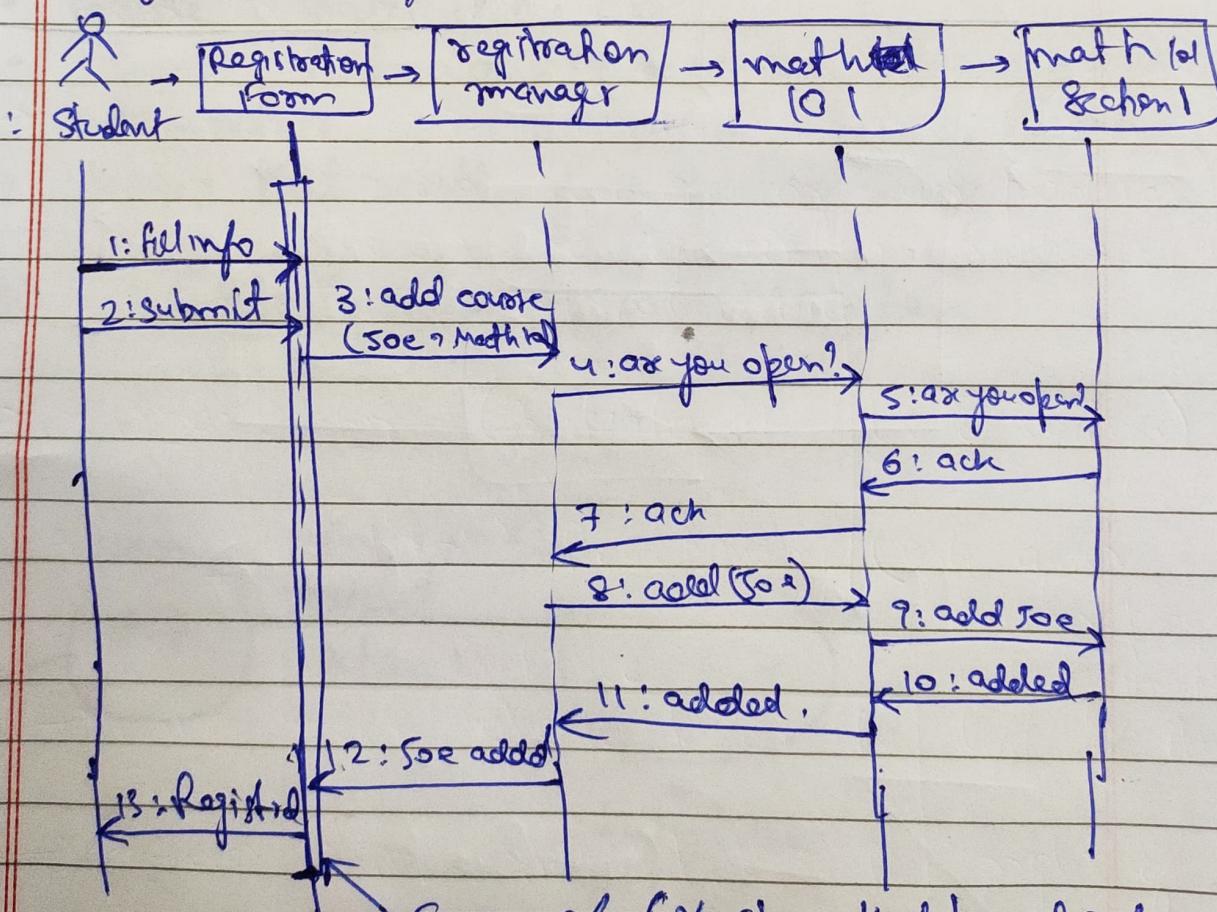
Date 1/1
Page _____
Puffin

- ⇒ Generalization: ($\leftarrow \rightarrow$) from children to parent
(General Use Case and Specialized Use Case.)
(Parent) and (child).

- ⇒ Extension points are just a detailed version of extend relationships.

Sequence Diagram

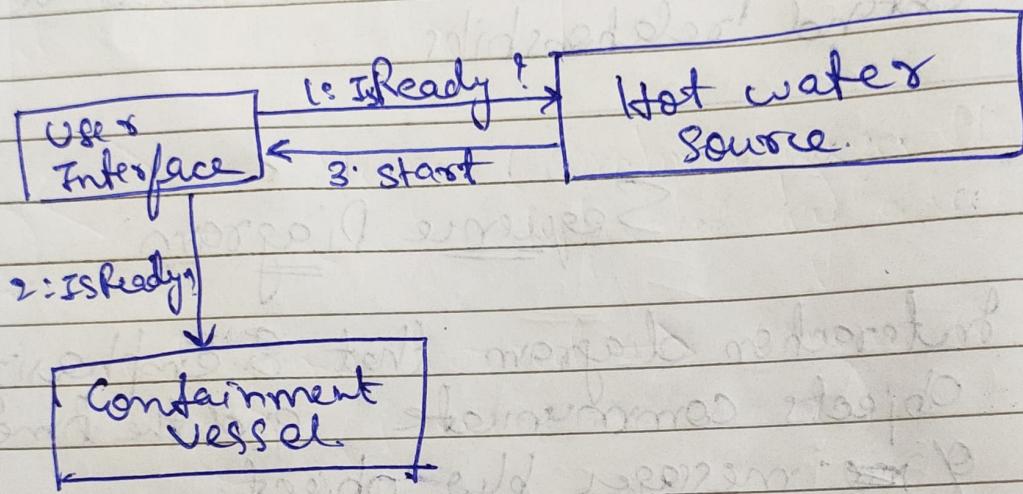
Interaction diagram that emphasizes how objects communicate and the time ordering of messages between objects.



Focus of (10) shows that control flow
changes when an object is performing
an action, either directly or
indirectly.

Collaboration Diagram

It was like an object diagram - But the lines were the objects indicated our operations and the line was numbered indicating the order of the operations



State Transition Diagram :-

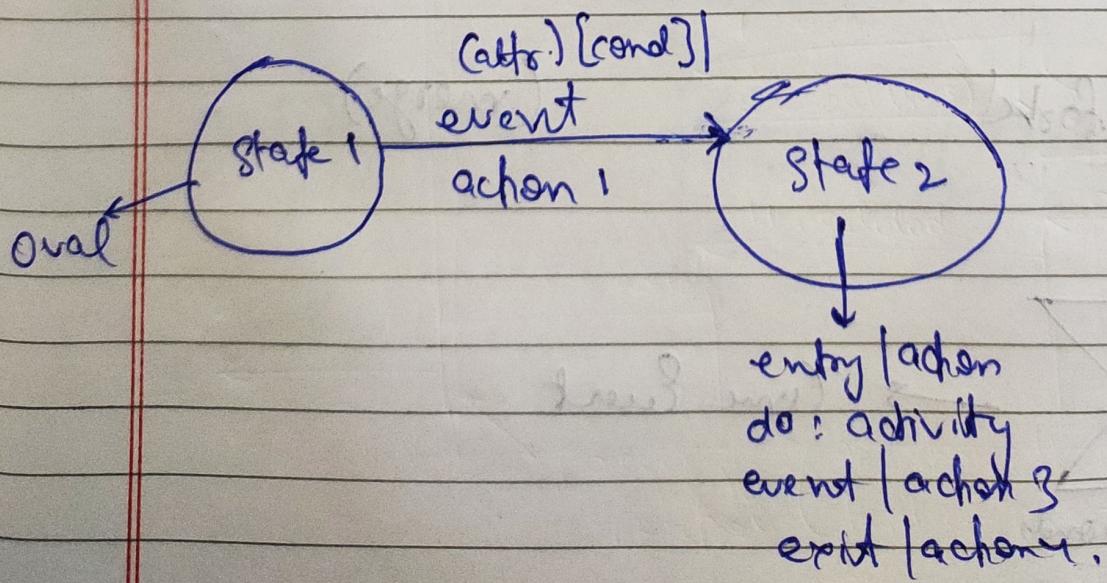
The state transition diagram is defined for each relevant class in the system and basically shows the possible live history of a given class or object - so

- Q- What does it mean to describe the live history ?
→ It means to describe the possible states of the class as defined by the values of the class Attribute.

for each relevant class :

- Possible states of the class
- Events that cause a transition from one state to another .
- Actions that result from a state change

State Transition Diagram: Notation:



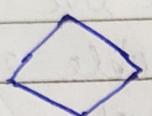
Activity Diagram

```

graph LR
    Start((Initial Node)) -- "Incoming Edge" --> Entry[entry]
    Start -- "Incoming Edge" --> Exit[exit]
    Entry -- "Outgoing edge" --> Action["Action  
(something happens)"]
    Exit -- "Outgoing edge" --> Action

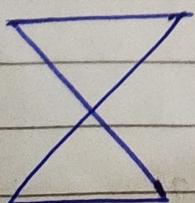
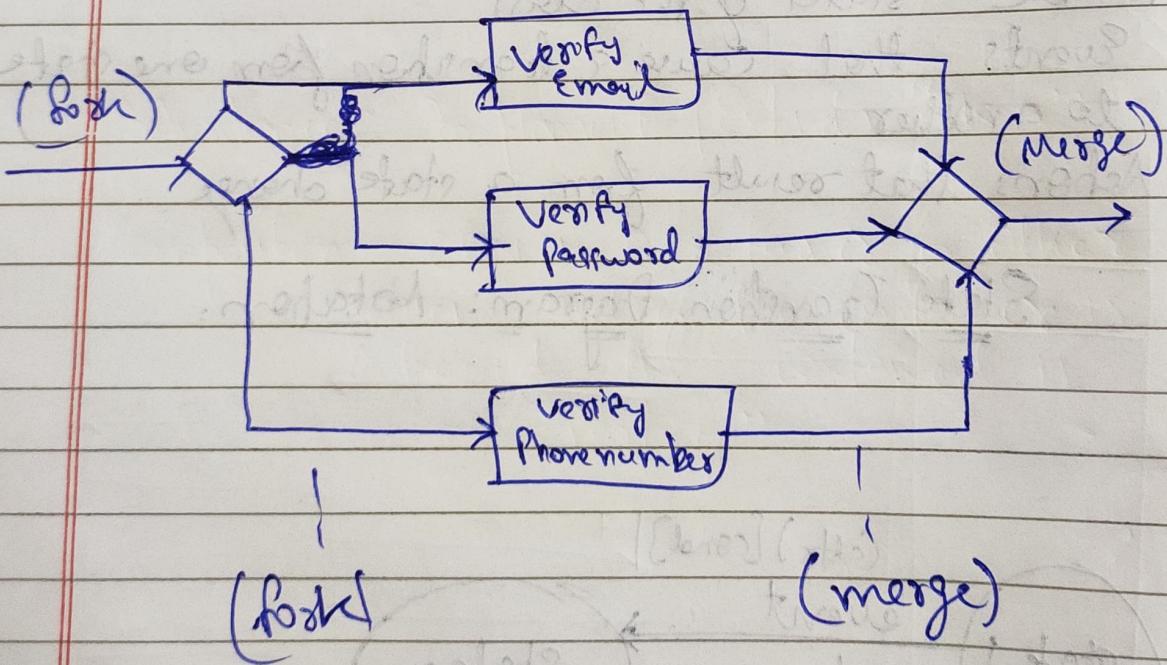
```

The diagram illustrates a state transition model. It begins with an 'Initial Node' represented by a black circle. An 'Incoming Edge' leads to the node from both the left ('entry') and the top ('exit'). From the node, two 'Outgoing edges' lead to the right, each labeled '(something happens) Action'.



→ Merge. (combines edges that are separated by adhesion).

authorized, unauthorized \Rightarrow Guard Condition



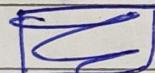
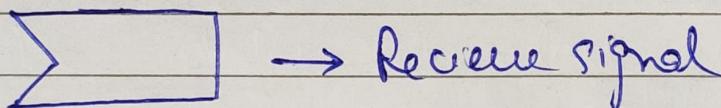
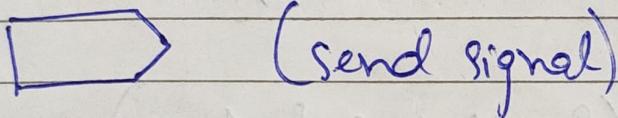
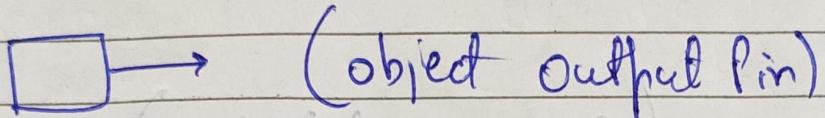
30 seconds

→ Time Event

2) Object Nodes (Information)

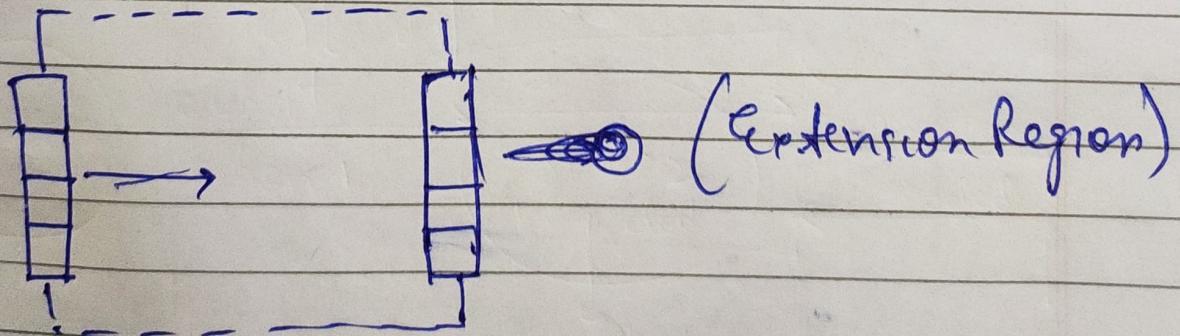
→ Indicating Event : (of user cancel).

→ ~~Flow final node~~

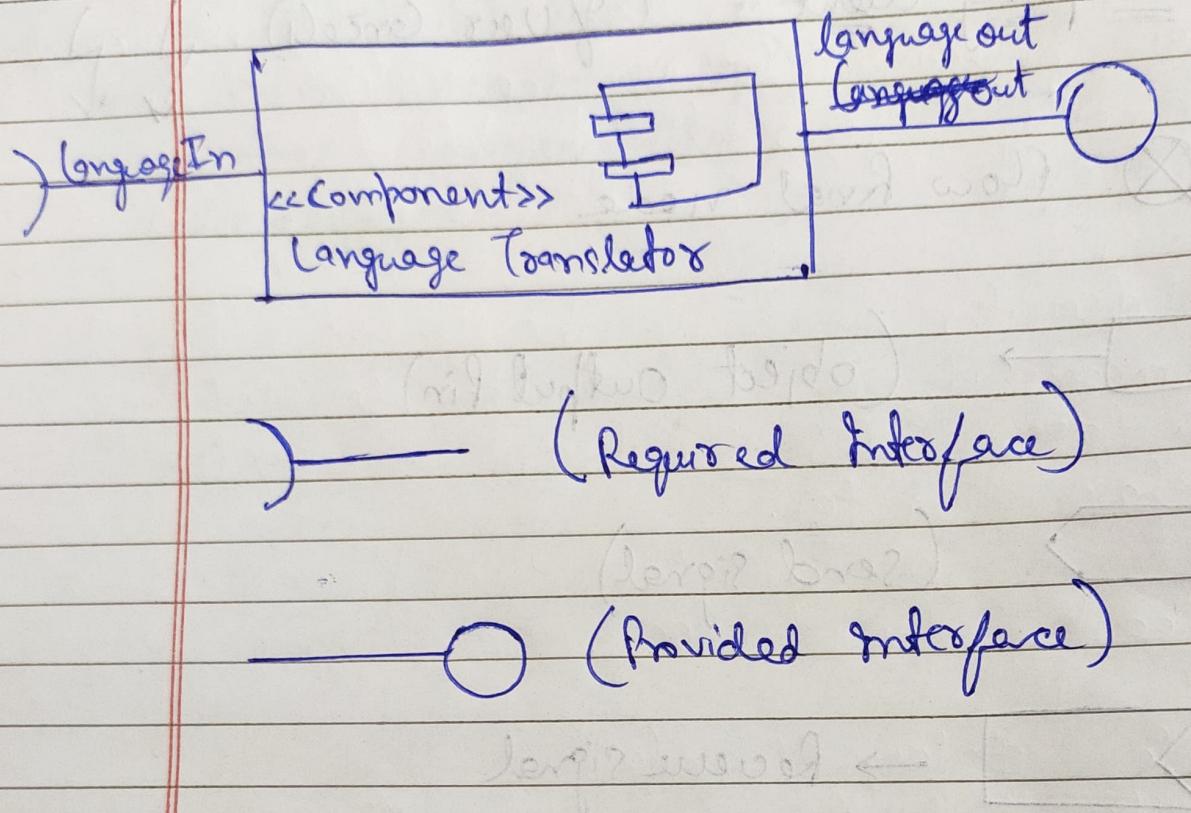


 Object Output pin).

c) → Connection



Component Diagrams & Composite Structures



Inheritance :- Implements | Extends "is a"

Composition : Class is a Container for other classes and if the container is destroyed the contained is also "Part of"

Component Diagram

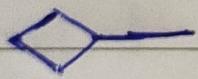
Ports & Delegation Diagram

- Ports & Delegation Diagram
- Show flow of data (Delegation).
- out (ports).

← Inheritance: (Implements or Extends a □)

--- Dependence (Class depend on something but it
isn't a member of the class
"Uses a".)

← Association: (Class contains an reference
to another class "Has a")



Aggregation

(Class is a container for other
classes, but if the container is
destroyed the contained is not
"Owns a".)



Composition

(Class is a container for other
classes and if the container is
destroyed the contained is also
"part of".)

Node Stereotypes

«executionEnvironment»
JSP Server
{Tomcat 7}

«JSP Server»
Tomcat 7

«database system»
MySQL 5.5



«database»
MySQL 5.5

«JSP Server»
Tomcat 7

«database»
MySQL 5.5

«database»
OS=Linux
vendor=MySQL
version = 5.5

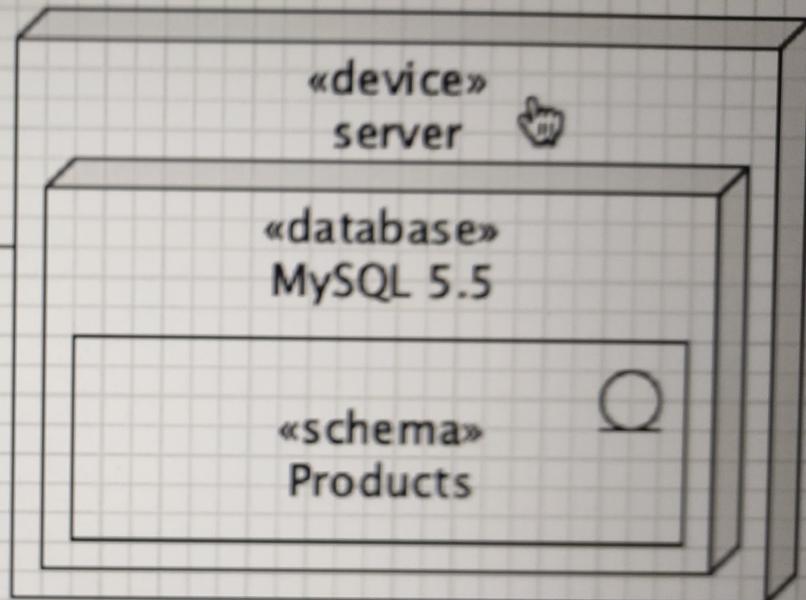
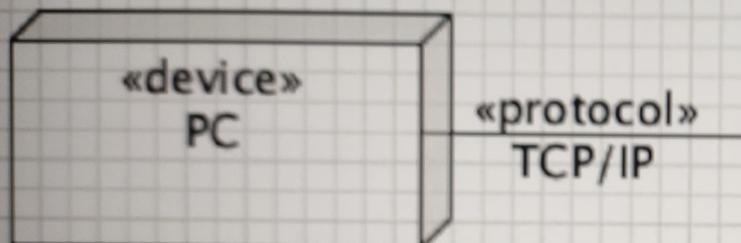
Named Nodes

«device»
cardVerify:Server

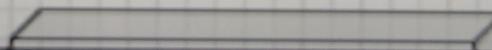
«device»
fundsVerify:Server



Node Communication Lines



Artifacts



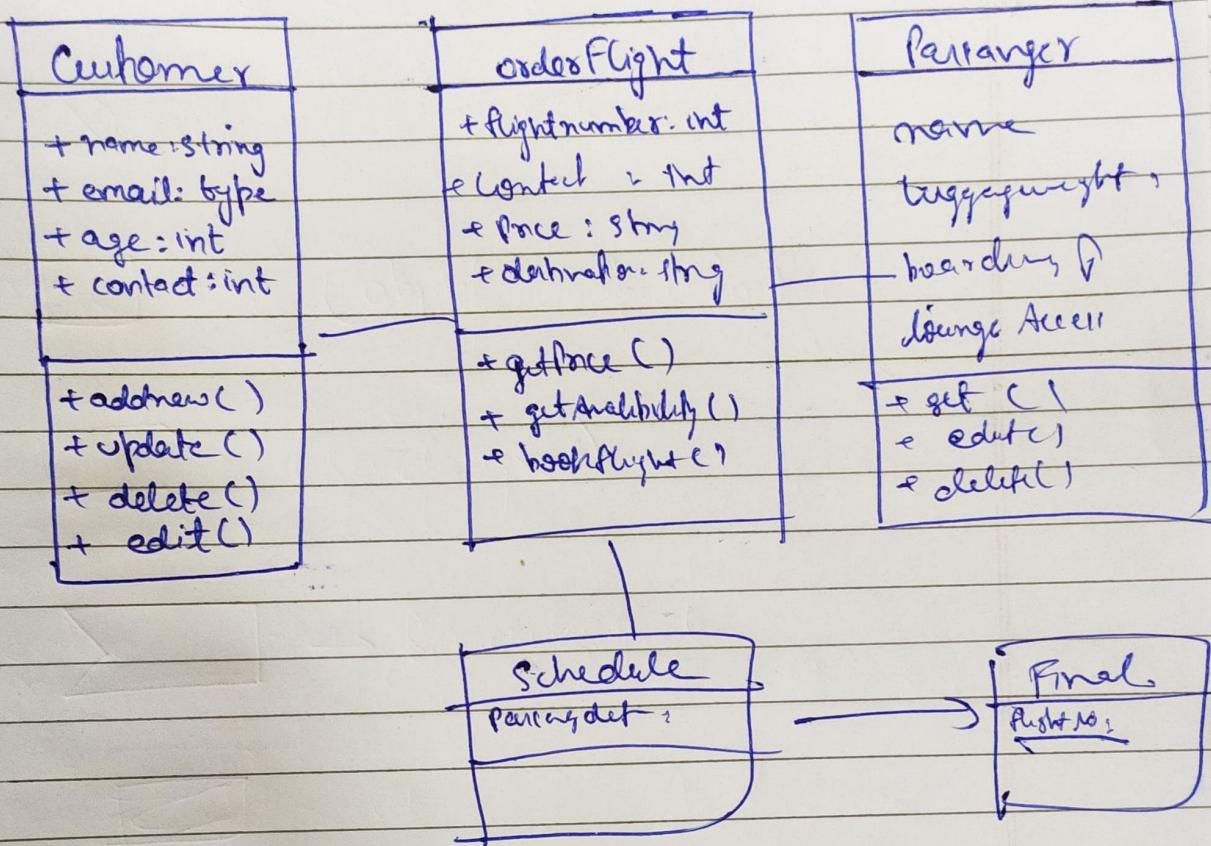
bpm

Artifacts

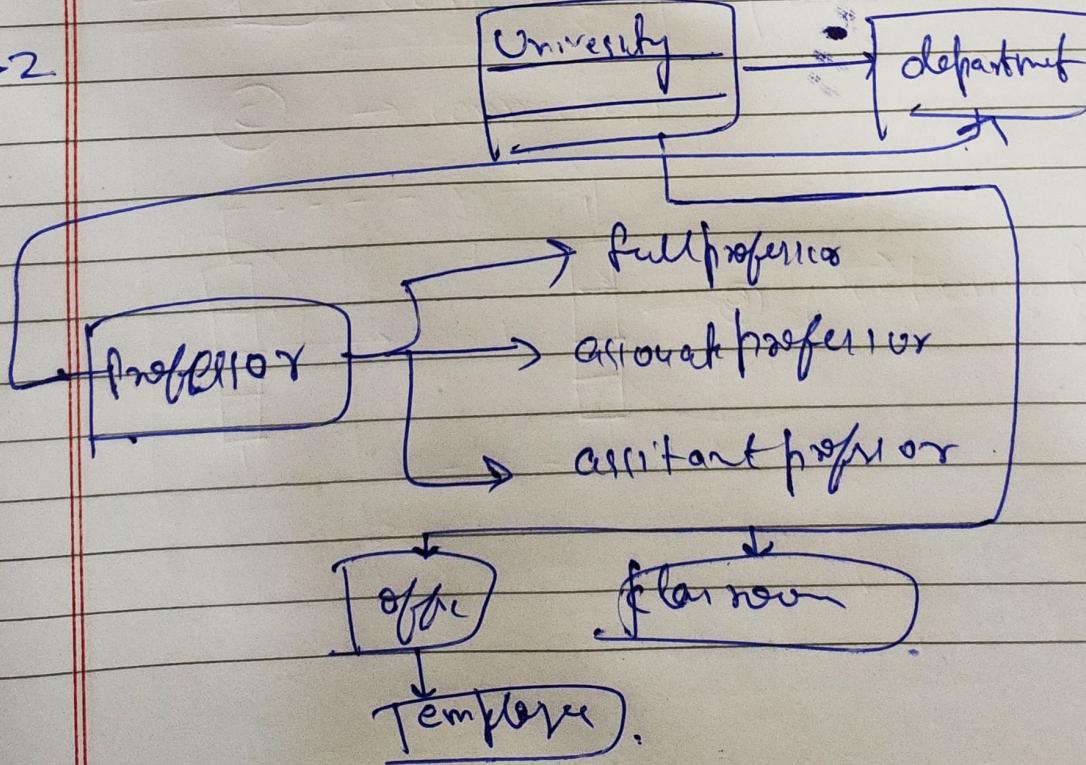
Assignment - UML

Date / /
Page _____
Puffin

Ques 1. UML Class Diagram

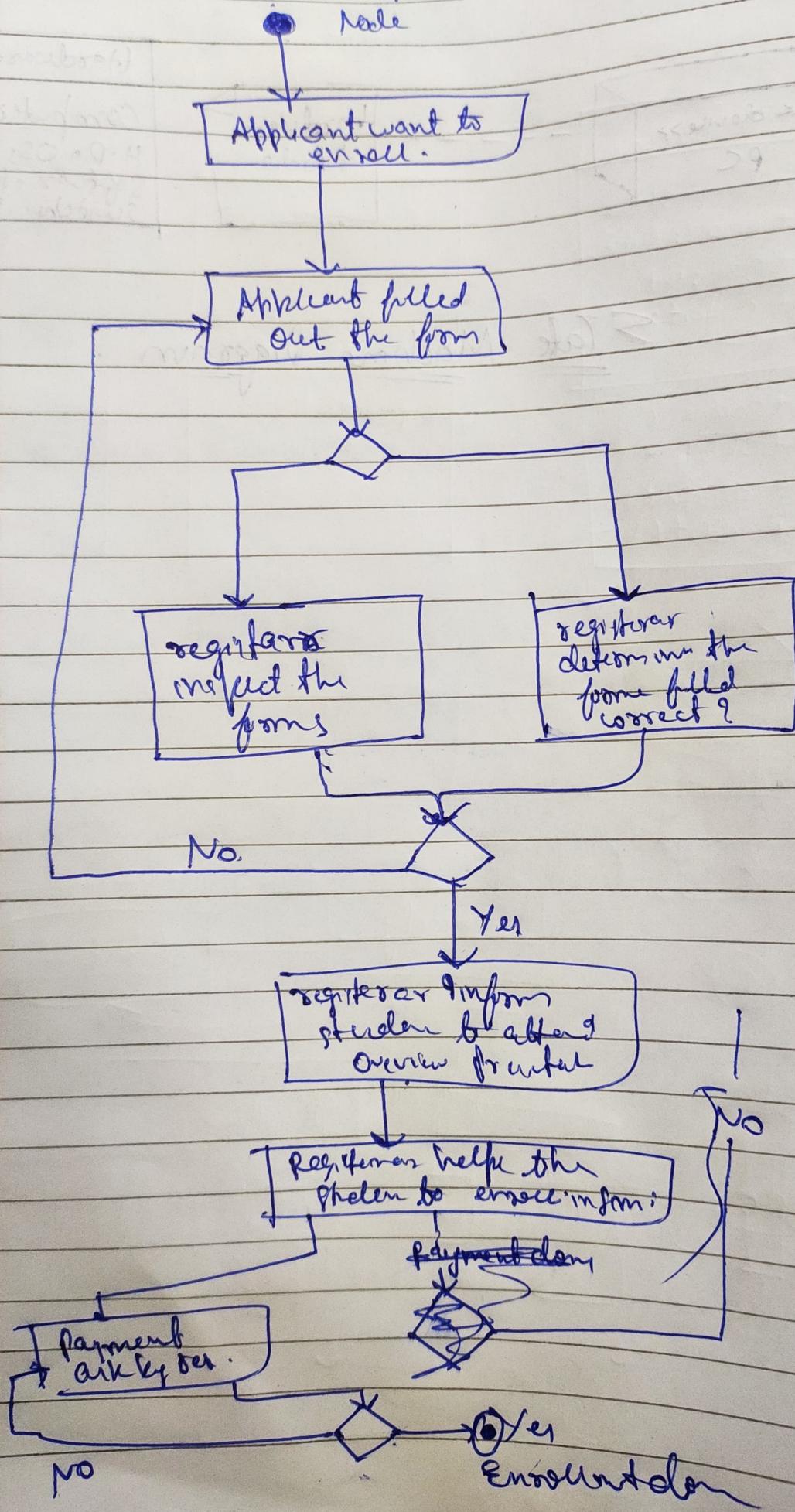


Ques 2.



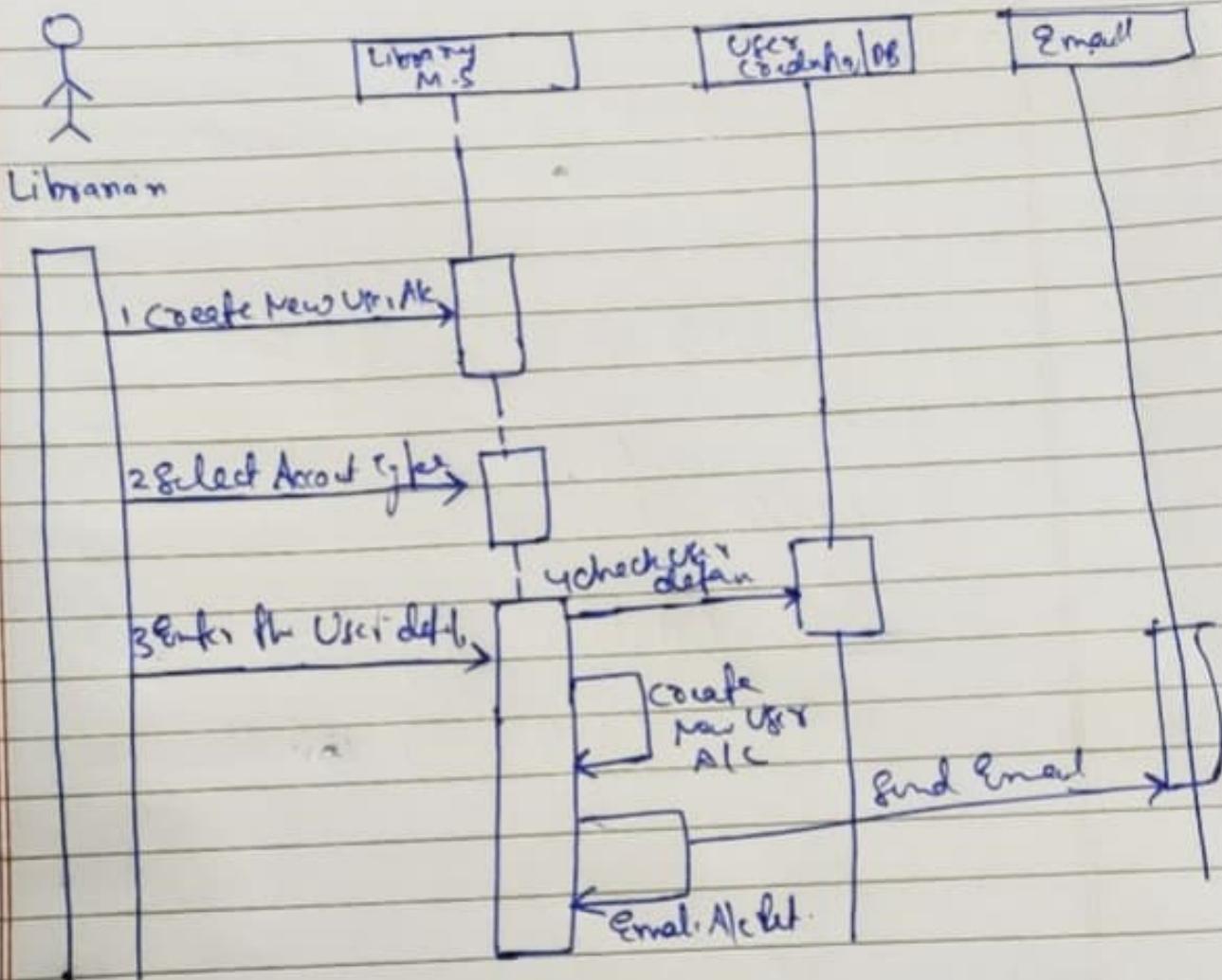
Sequence diagram

Initial Node

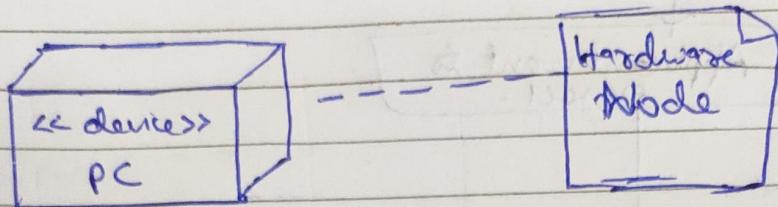


De-4 (Blue front) Sequence Diagram

Date / /
Page _____
Puffin



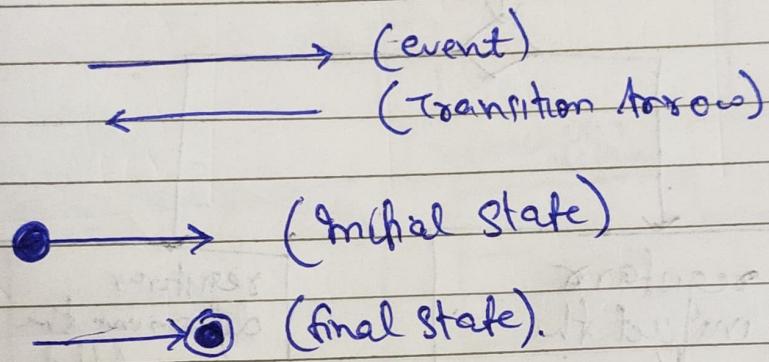
Deployment Diagrams



Hardware Nodes

Computer & Server
H.D., OSs, Servers
Software, Router,
Switcher, Firewall

State Machine Diagram



State Internal Behaviour

Choice Pseudo states.

Diagramming Signals

Package Diagram