

OOD - Blaha?

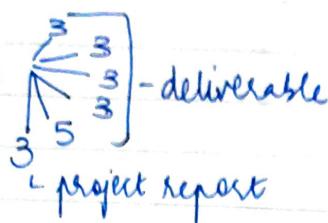
classmate

14/8/19

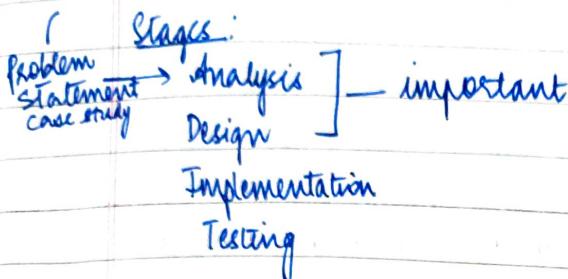
ISA1 40 to 15

ISA2 40 to 15

Project 20 to 10



client



Analysis:

- semantics of statement
- build model
- o/p: analysis model [ER diagram] easy to understand & explain
- changes are easy to adapt

Design:

- I/P: analysis model
 - o/p: design model - more specifications
 - inclusion of technical details
 - relational schema
- analysis
+ technical

Implementation:

- I/P: design model
- o/p: application
- platform, coding

Testing:

I/P: application + test case

O/P: final product

OODP

Modelling & Design
1 - 3

Design Patterns
4 & 5

3

design view

Implementation view

* not there in course

Usecase view

Interaction view

Deployment view

2

example for analysis model

ER

→ things
→ relationships
→ diagram

Unified modelling language
defines rules b) notations while
building analysis model.

real time usecase gives you information about:

- actors / end users / sellers / system admin
- roles played by actors
- one actor can play many roles

Interaction: object interaction

- : sequence diagram - done for every activity
- : activity diagram

design: class diagram & object diagram

public, private, protected

with

deployment: details of component & where to place
real time
entity + hardware

Design Patterns - choosing the pattern (among 21)

Structural | Behavioral

Creational

Creational: to create/initialise

Behavioral: how actors get involved

Structural: Structure of objects

9/8/19

- easy to communicate w/ end users.

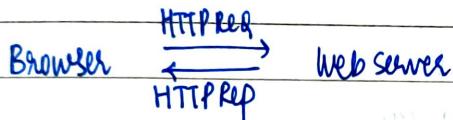
- Case Study - problem

 - solution - design model

 - analysis - concentrates on problem

 - doesn't have any technical aspects.

 - problem sector



 - System Conception → Analysis → System Design → Class Design → Implementation

 - functional & non-functional requirements are separated

 - NFR are not modelled

 - FR are input to analysis model

 - structural

 - Models

 - Behavioral

Identify entities → Identify attributes → Association

Structural: existence of an object

Behavioral: how objects interact & changes within objects

every notation has a meaning.

UML → things

- relationships

- diagrams [Behavioral / Structural]

Use Case View

yp: case study
op: use case model

(interactions)

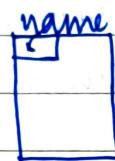
- visualise people who use the system & how they behave.
 - actors
 - primary - initiates
 - secondary - provide service upon request.
- client is a primary actor.
identify actors and categorise them - Step 1.



every actor must interact.

each interaction is a use case.

→ system - application

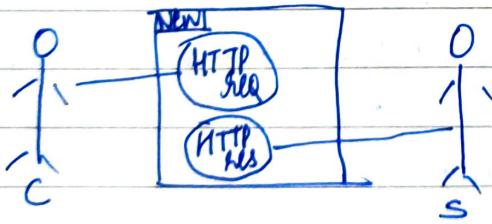


all use cases must be inside system boundary
actors cannot interact outside the system.

can be system or a real person.
not part of the system

on LHS of system, place actors who initiate.

on RHS of system, place actors who provide request.



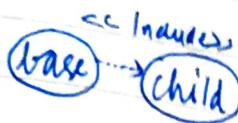
association b/w actor & use case is mentioned ~~using~~ using straight line

actors trigger use case.

one actor multiple use cases.

Relationships
 - include
 - extend
 generalization

Include:



Extend:



Generalisation - similar to inheritance



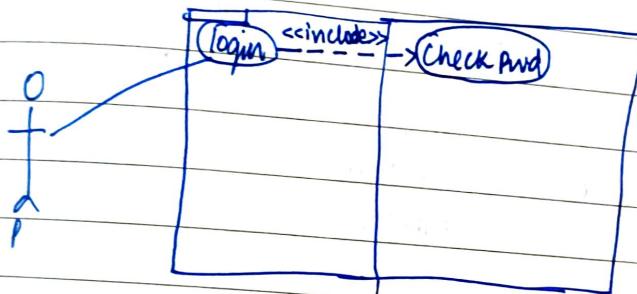
Include: base executes, child HAS to execute avoid describing something

Extend: base → child may/may not.

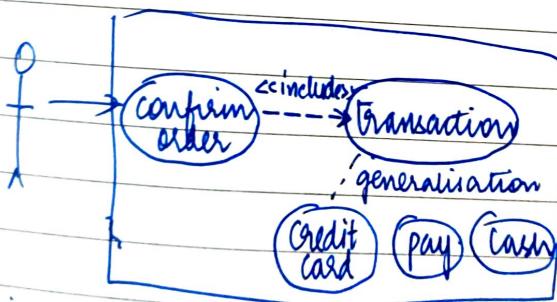
Generalisation: base → only 1 child HAS to execute both use-case & actor is only related to base class.

actor

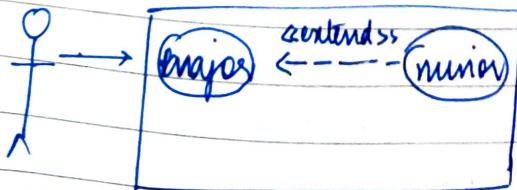
Log IN: Association



online shopping:



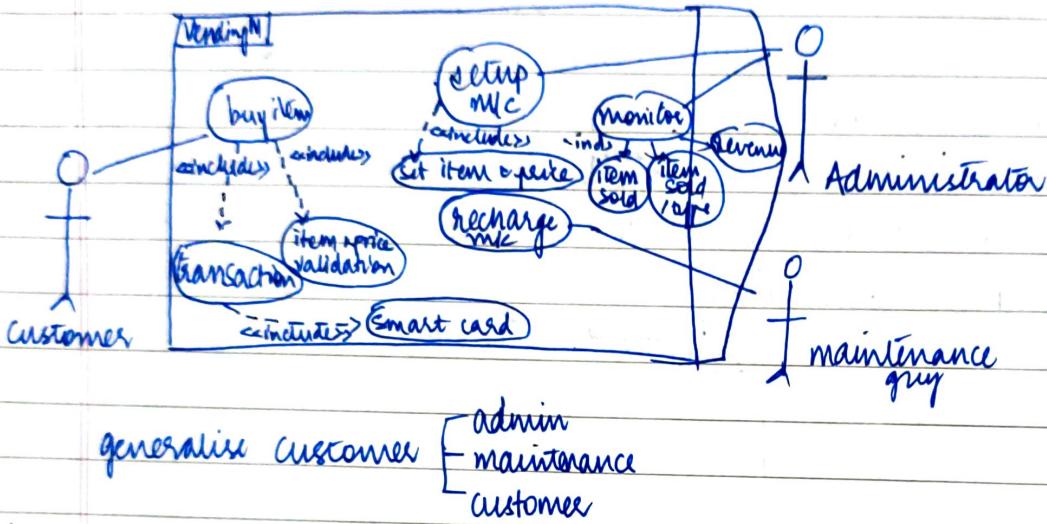
major:



explicit mention of server is unnecessary

every model has a documentation

- name - verb (start)
- summary about project [1 line]
- actors
- pre conditions: must exist before use case executes
- description - steps to execute . in detail
- exceptions
- post condition: after this, what?



26/8/19

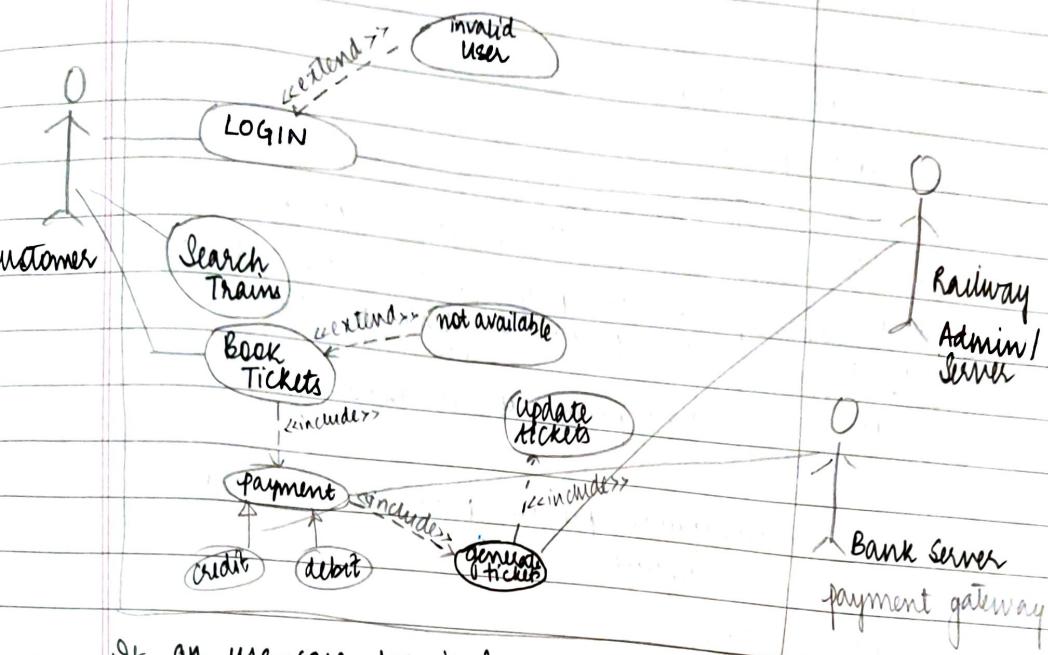
1. Draw an use case diagram for railway ticket booking system Identify at least 3 actors. Relationships: include & extend use-case: WHAT.
how something is done: class diagram

2. Hotel Management System.

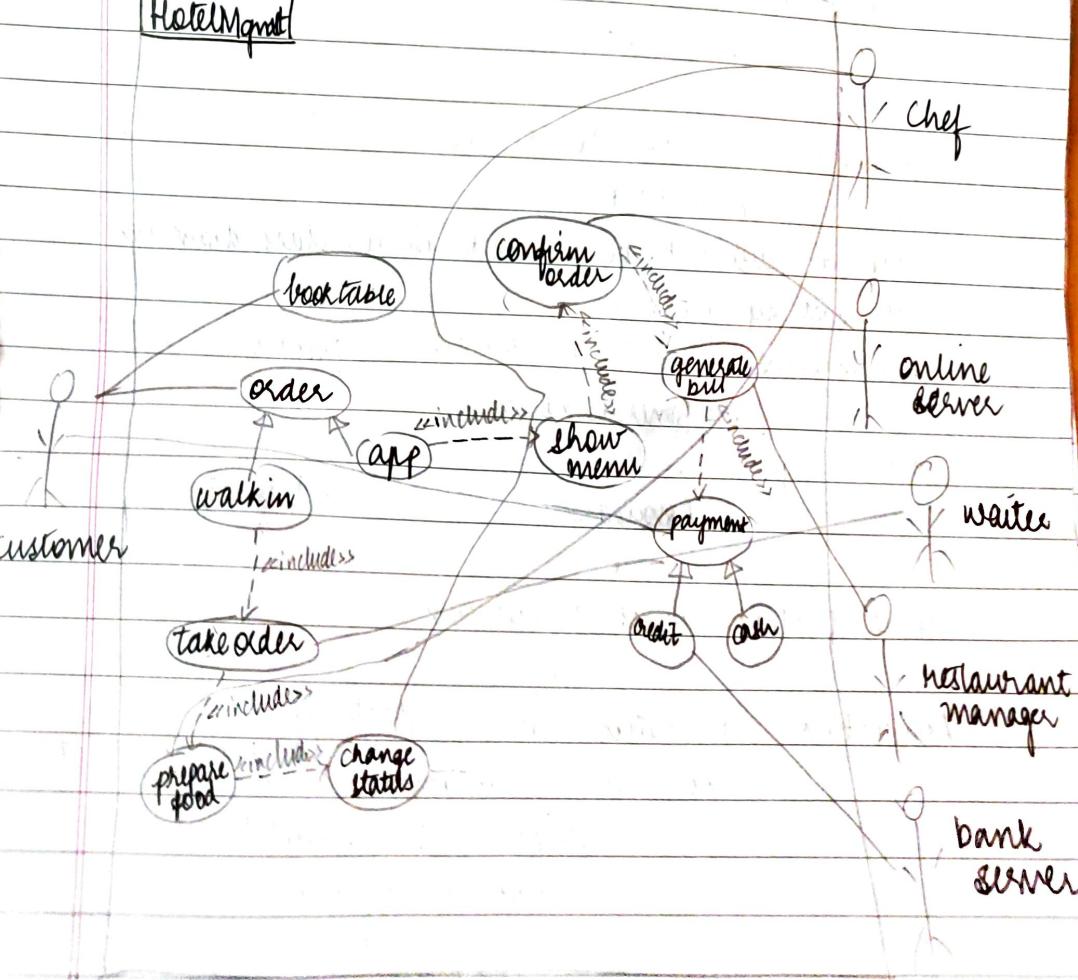
A customer books a table if ~~group size~~ group size > 6 or a customer walks in or a customer can order food online.

While booking/ordering, online application should show menu. When customer is assigned a table, waiter takes order. Manager generates bill & maintains the restaurant. The chef prepares food & once the food is prepared, status of ~~food~~ changes. Payment can happen using card or cash.

1

Railway ticket

If an use-case doesn't have an association, system handles it.

HotelMgmt

use case tells you what happens
class concentrates on how things are done

Class: represented by a
four parts

NOT:

composition [

class Name
declare variables
operational functions
responsibilities

- first letter must be CAPITAL.
- access specifier; variable name; datatype
- access specifier functionName(attributes): return type
(camel case)
- description about class

+ - # / (public, private, protected, derived)
default

+ name : string

+ phoneNumber (n: name) : phone number
 |
 |
 | default actual

derived doesn't exist. - doesn't come in implementation
implicit \rightarrow explicit details.

implicit must not be present in the class diagram.

generated by system

responsibility - contract/obligation to perform a service.

Relationships Generalization
Association

Generalization: parent \leftarrow sub class

Association: Aggregation vs. = part of - composition
Composition parent & sub exist independently. But assoc.

If parent class is deleted, subclass
cannot be accessed.

(browser window)
or chess board

Aggregation: ◊

composition: ◊

7

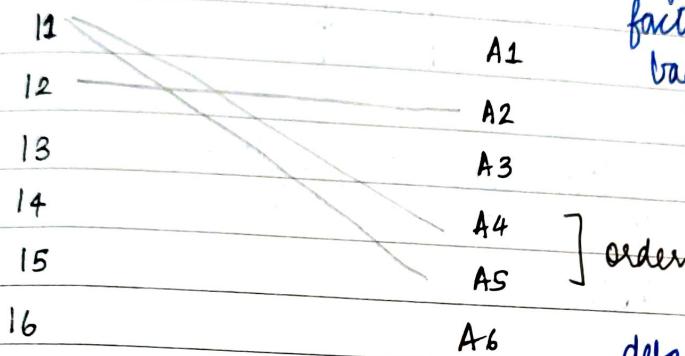
Association:

- between classes or within one
- multiplicity: how many instances of 2 classes are related to another.
 - : one one, one many, many many
 - : 2..8 [range]
 - * many

[bag] [ordered] [sequence]

C1 [role name] C2

along w/ multiplicity factor, there's either bag, ordered, sequence



default is bag.

without any lines: any order.

order is important.

order doesn't allow duplication. sequence can.

Dependency: association is "weak"

- dotted line -

19/19

Model ↗ Analysis

↗ Structural

between 2 states, there

must be an event

[conditions] in ()

Behavioral method: how objects interact w/ each other.

to jump from 1 state to another, there has to be a trigger.

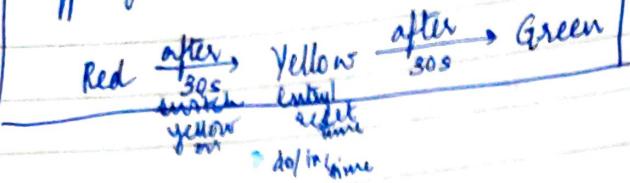
State Model

State
denoted by
□

Event/trigger Activity

- signal event - <> signal > - mouse clicks
- change event - when (condition) (graduation)
- time event - after (time) [traffic signal]
- when on (date)

Traffic signal



start : event

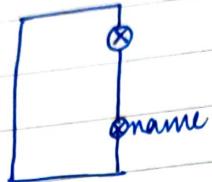
end:

or



more than one. can be zero.

or on the edge of a system



Activity: ontransition, do, entry, exit.

all four possible.

State 1 event/activity
event
[guard] / activity

State 2
activity
exit

state 2
do / activity
entry / "
exit / "

Red after 30s / switch on yellow Yellow

1st year student Take 2nd semESA / pay fees

2nd Year Student
entry / enroll subjects
do / write exam
exit

guard: boolean expression.

event + state are levels of abstraction.

State Description:

State :

Description

Sequence of events j:

to produce state j:

do / exit / obtain

new state

Conditions that characterize:

State

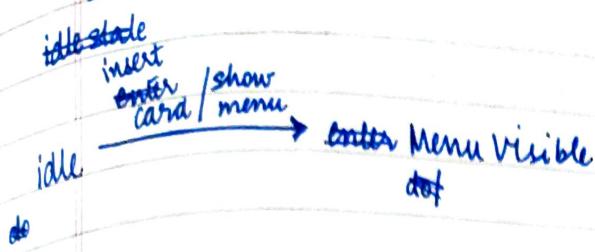
accepted Events of

current state.

State Diagram for ATM / PESSAT

Date _____
Page _____ 11

Withdraw



Order of evaluation cannot be shown in 'do'. In case of complex systems, where order matters, break that into sub

9/9/19 Advanced State Modelling

Nested State
Concurrent State

while exam pay fees → 2nd year entry exam subjects
dot white exams (ESA1-ESA2) } break into sub states

for eg: 2nd year → 3rd Sem
 |
 4th Sem

every sub state must have a name.

Nested State can have any number of states.

The transition from complex state to nested state must not be to the entire diagram but to a particular state.

Concurrent: System performs multiple tasks simultaneously.



A dotted line == concurrency.

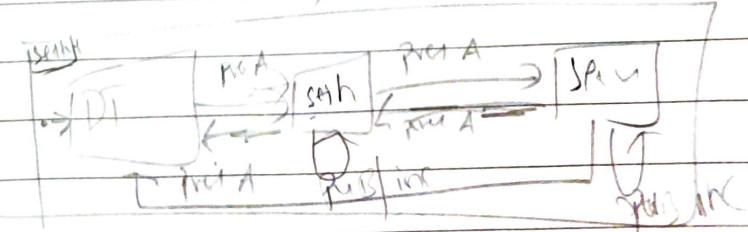
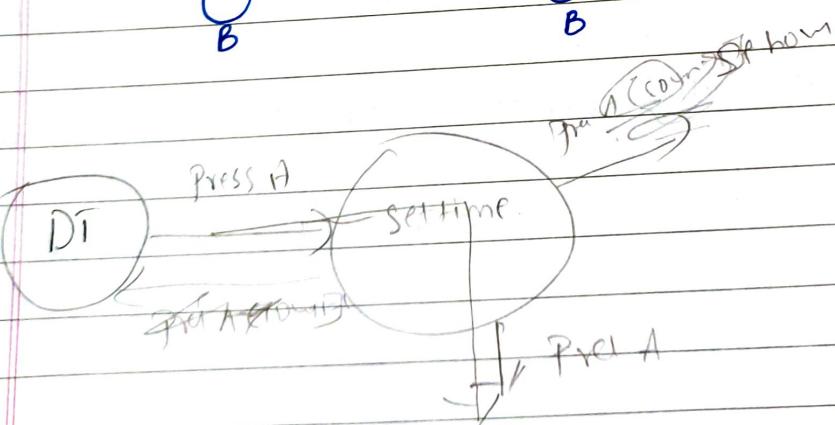
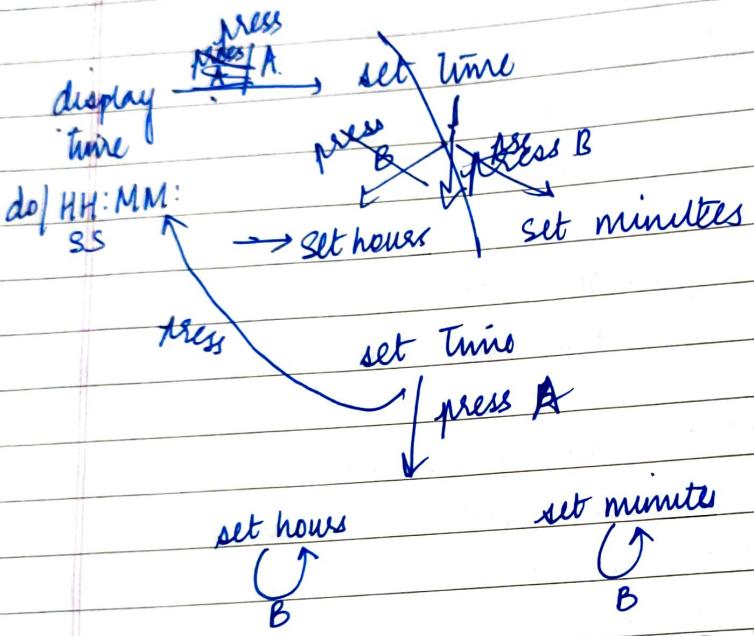
----- |

S2

A simple digital watch has a display & 2 buttons (A & B) to set it.

The watch has 2 modes of operation (display & set) time
or display: watch displays HH:MM:SS
set [] set hours
[] set minutes

A: selects modes each time - sequence: display → set H, set M
B: within submodes, B advances H/M once each time it is pressed. Button must be released before another event.



no loops in activity

classmate

add in sequence

13

11/9/19 State diagram fails to answer. order of activity
how activity is evaluated

activity diagram: order of evaluation

sequence diagram: how it is performed

activity diagram is similar to a flowchart

↳ must have one or more end points.

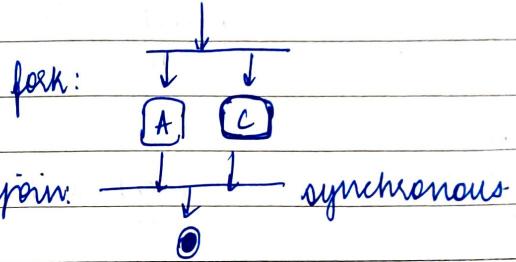
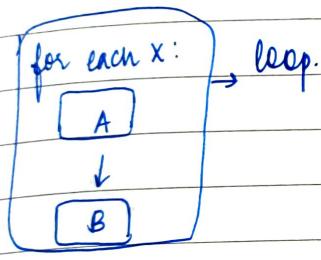
end must reflect the entry.

start: • & description about activity can be shown in activity diagram.

end: ◎ inclusion of decision points. There must be a guard (condition) Multiple outgoing edges

Exception path: not primary.

Concurrent process: A1 A2 - order of evaluation is unknown



non-synchronous: rhombus.

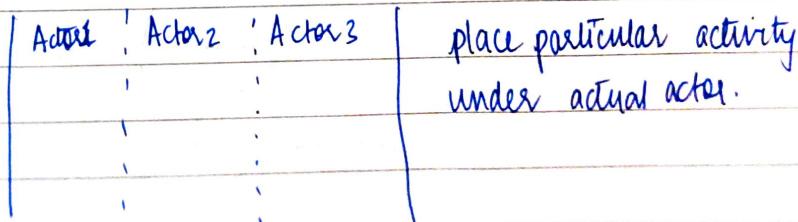
Class Diagram: has class name. not underlined: a noun

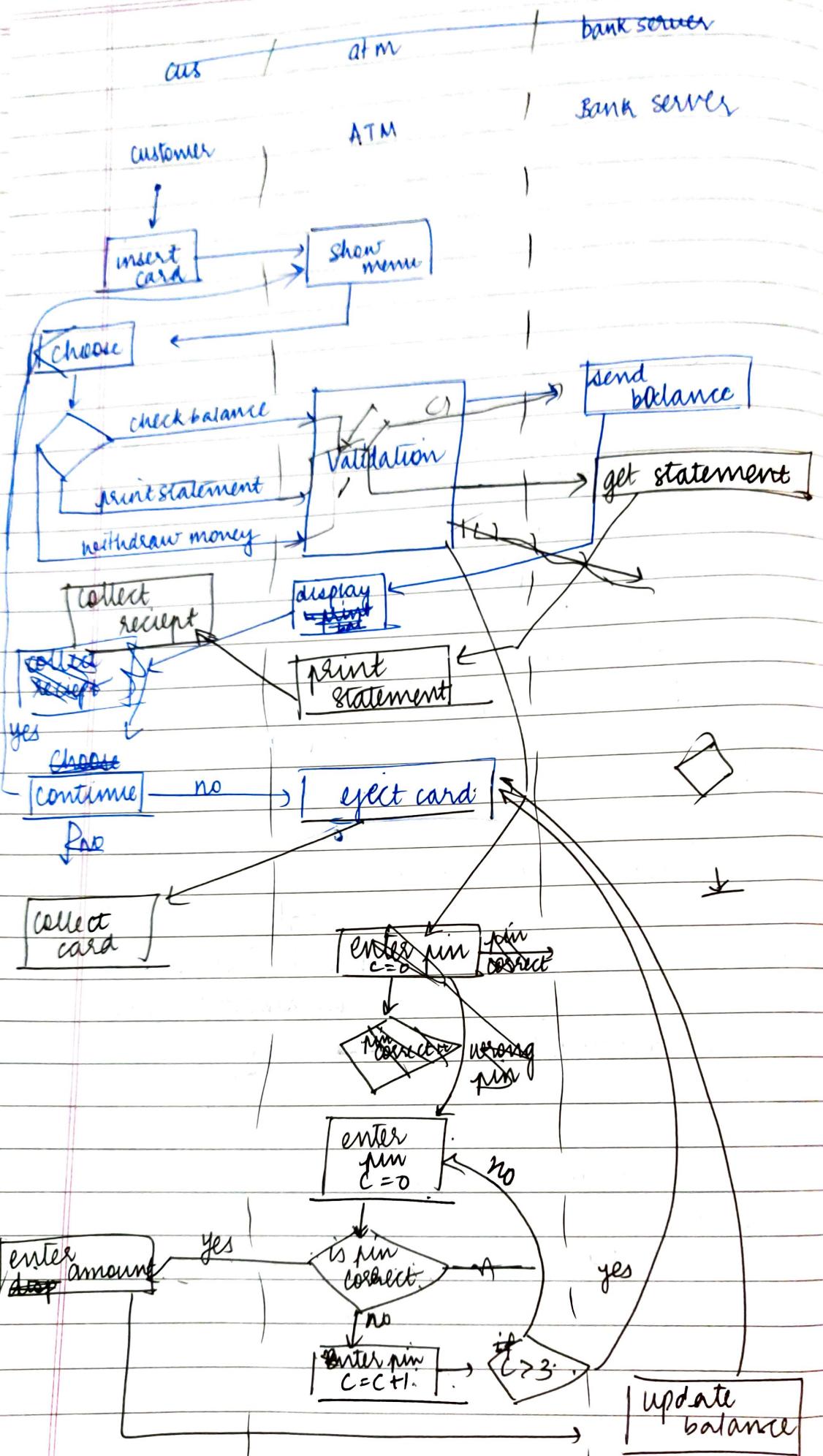
Object Diagram: has obj. name. class name: objname: CN

↳ show additional activities in a rectangle connected using a dotted line.

actors

Swimlanes: sometimes multiple objects or clients are required.





Patient:

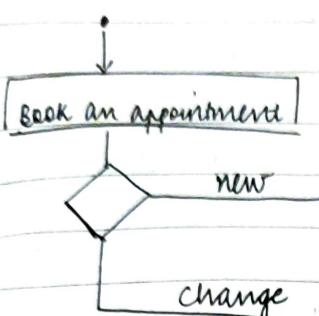
- fill form
- appointment

Receptionist

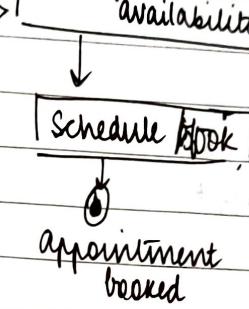
- check / schedule appointment
- send a reminder

division of activities can be done.

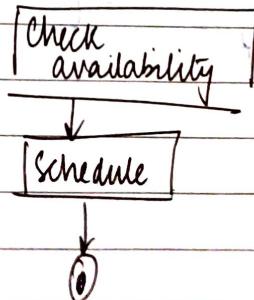
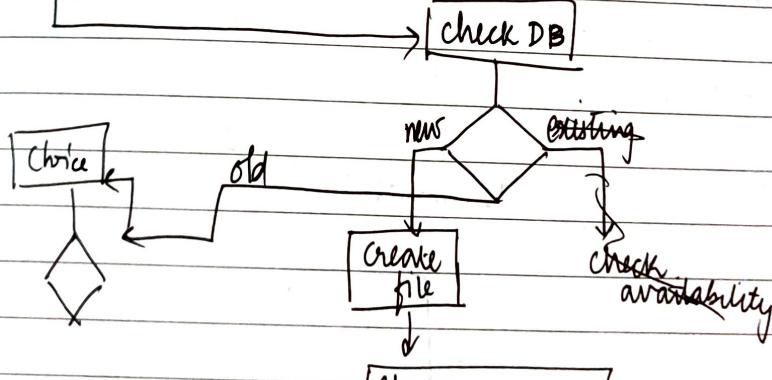
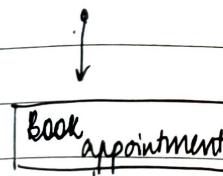
Patient



Receptionist



OR.





~~new?~~ [changes]

create a file

check availability

schedule

booked

unavailable

Patient
visit office

enter office

Receptionist

check DR

complete form

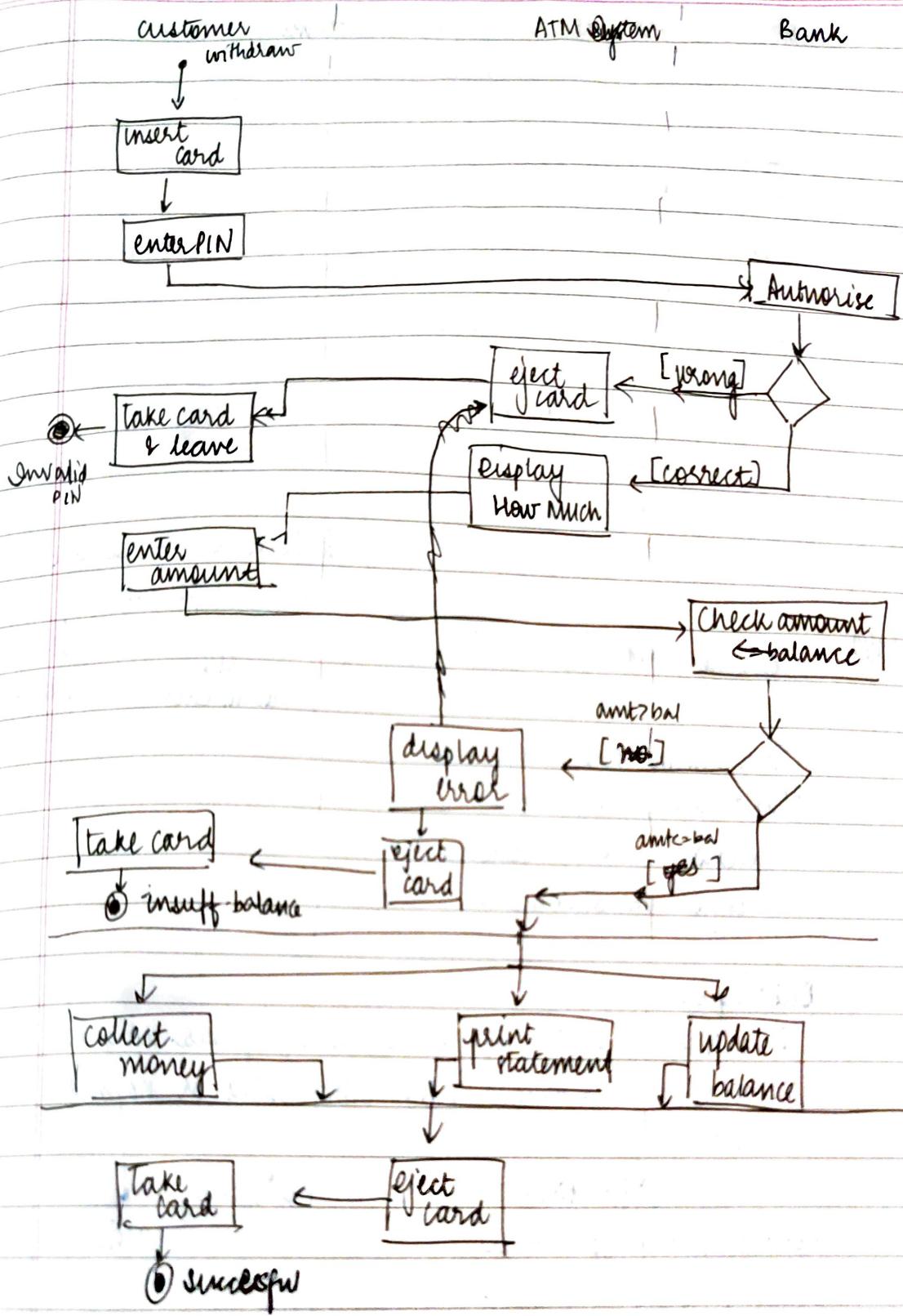
[new]

wait for turn

[old]

desc.

ATM Transaction



hockey league: at least 4 hockey teams
 hockey team: 6-12 players
 1 player is captain
 play games
 name record
 has a coach

normal coach captain-	Team: Name, record Players: Name, Address, Number, position Game: score locations
	Coach: level of accreditation years of exp. coach multiple name address

Sequence Modelling

Interaction - use case



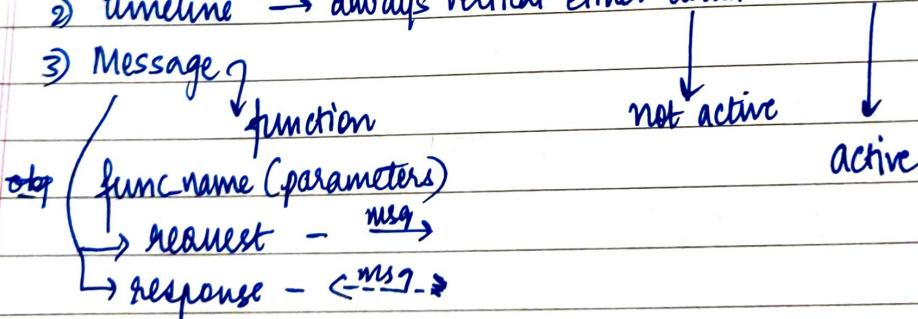
Each activity has a sequence
 "How" an activity works.

Scenario * Sequence

↳ english like statements

Building Blocks

- 1) object → denoted by on Top of seq. diagram
- 2) timeline → always vertical: either dotted or normal
- 3) Message ↴



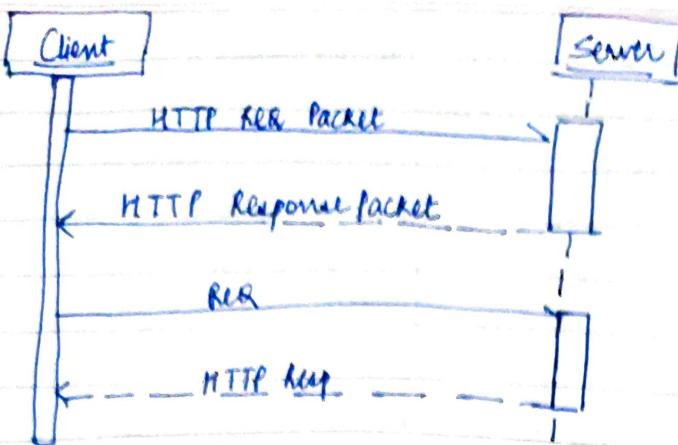
all interactions happen through a transition

Request: async: time unknown →

sync : time of message passing known

Response: async: ←---

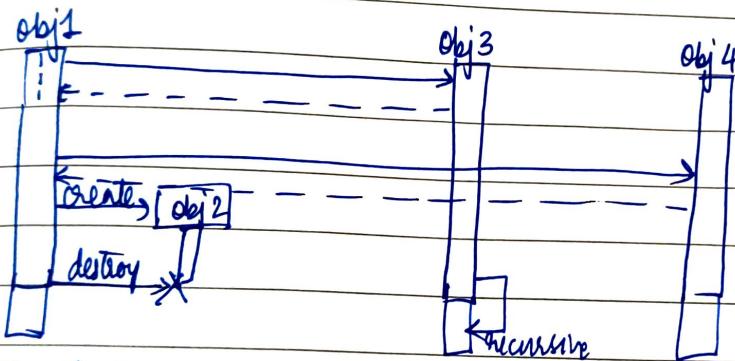
sync: ←---



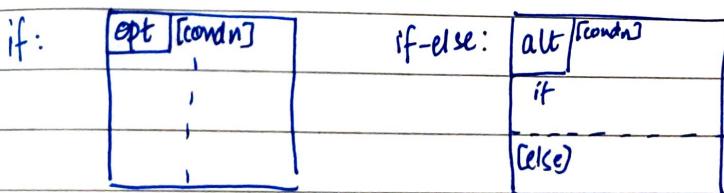
objects:
 active - for an object, if timeline is always active
 passive - if timeline varies [depending on `newest()`].
 transient - created when necessary. destroyed when job is done.

when objects are created, ~~if too~~ & instantiated as soon as scenario begins. destroyed only when scenario ends.

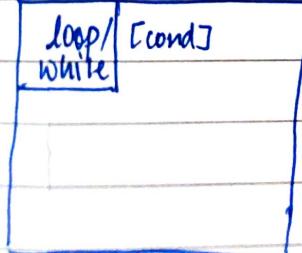
only object that creates can destroy. other objects can interact.
 destroy: X. If not destroy, it'll get destroyed when the creator is destroyed

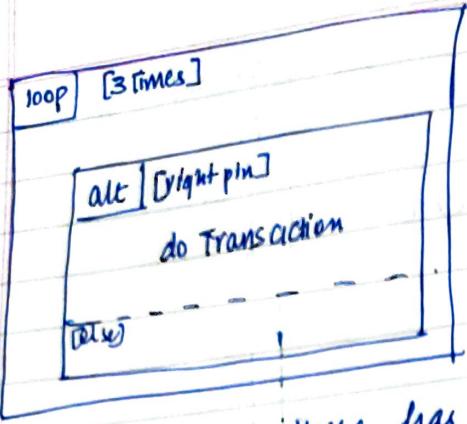


for, if, if-else:



for/while:



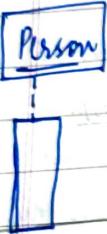


for every req, there has to be a resp.

withdraw funds:

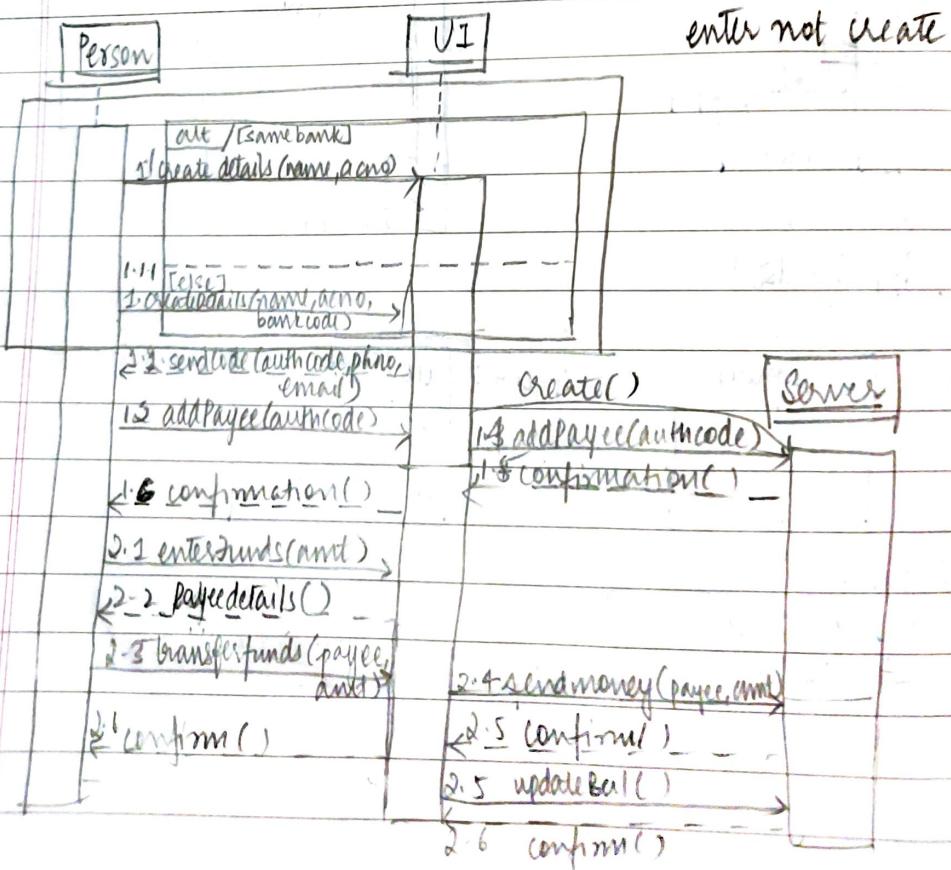
Person

ATM



Store

Payment Gateway



CLASS DIAGRAM

Date _____
Page 21

Increase Generalization

Class diagram must match pattern

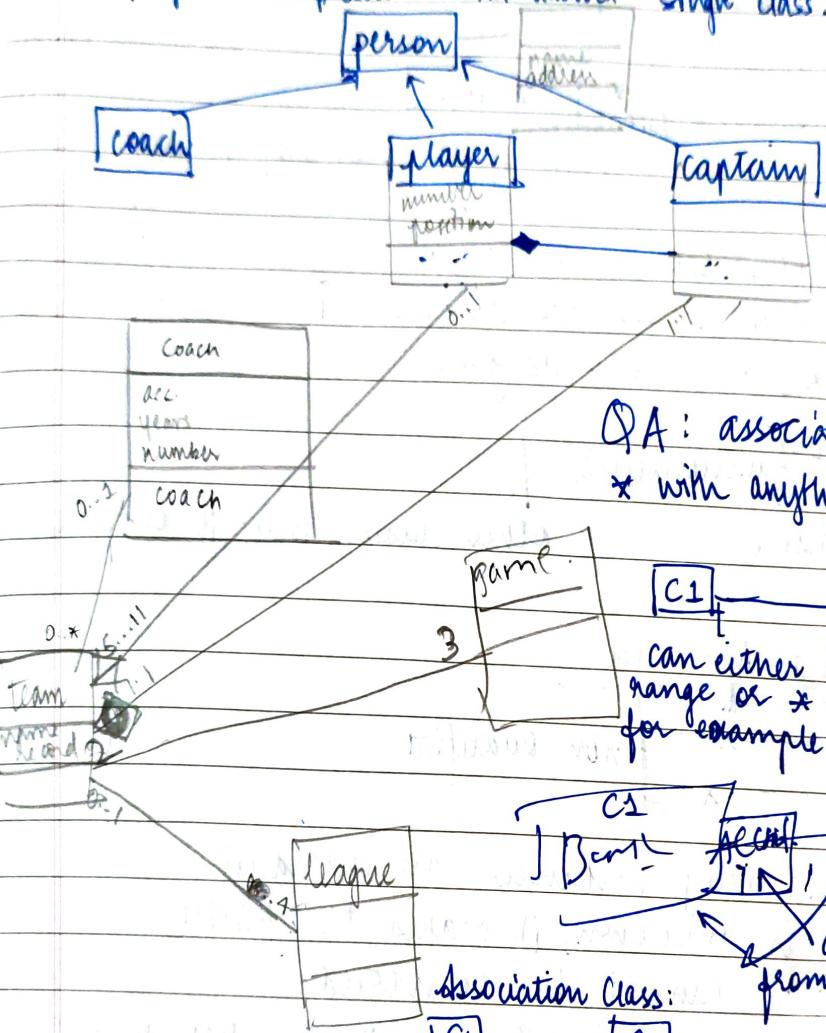


high dependency
because direct association

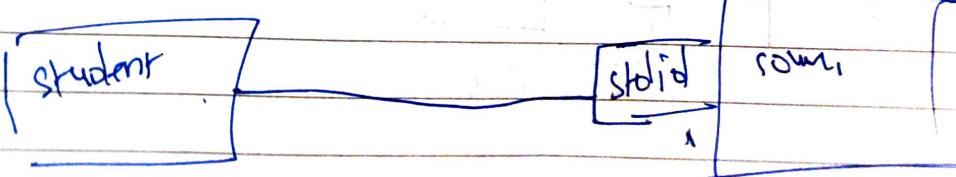
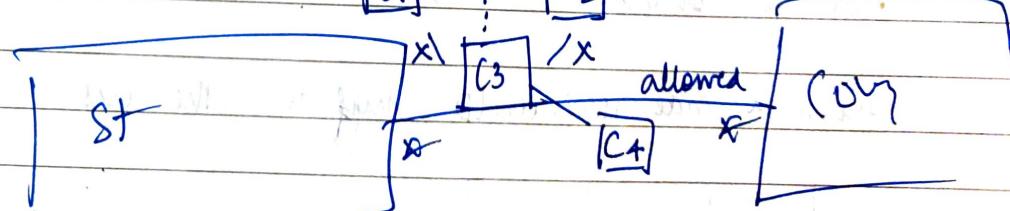
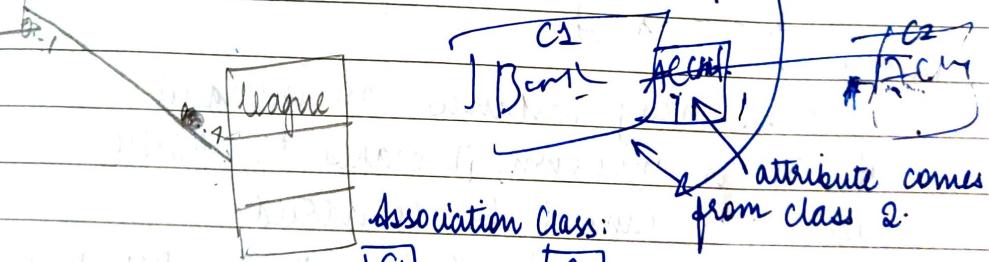
aggregation
composition] part of

from IC:

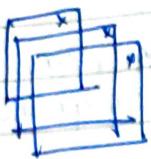
player + captain : not allowed. single class responsibility.



`C1` ————— `C2`
can either be a number,
range or *.
for example,



28/8/19 by default: ordering is bag.1 unordered w/ duplications



monitor, $\xrightarrow{\text{ordered}}$ close

sequence: order + duplication



if it's ordered, 1-way

time table: sequence.

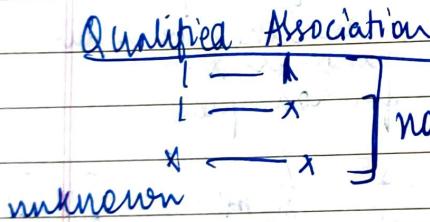
use case desc: use case diagram

UML: class diagram



name

other classes that it associates with



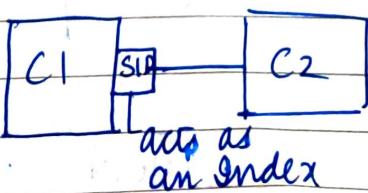
how many instances are interacting

If 'n' is unknown, it makes it difficult.

All 'n' cannot be converted.

try to convert 1-*+, *-* to some number

from * side, unique ID shift to other class



Obj- Constraint language:

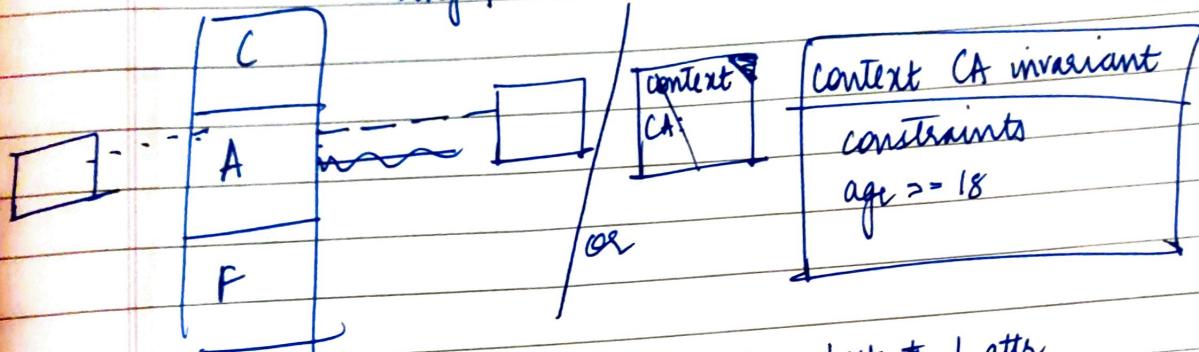
add constraint & navigate through class.

Can all classes be reached? If no, there's a mistake in Assoc.

constraint - all functions (or 1)

class

single/all attributes



constraint not added to entire class but to attr

~~filters~~ variables

~~class~~ filters (SQL like commands)

sum

count

Navigate

variables

methods

association

won't come
in exams

done by asking questions