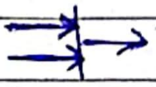
# "Activity Diagram"

Date ________

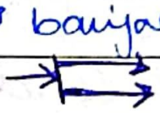
Activity Diagram:

Join → ये multiple inflow single outflow.



fork → 1 ⟶ [ 0.5 0.5 ( ek activity se 2 activities banjaengi)

Single inflow multiple outflows.



● Solid circle se start hoga hmesha ◉ End hoti activity diagram ispar.
↳ start marker.

Data Store: keyword : «datastore» Database.

data ka flow nhi state object k through define hota.

Object: we will show the object in rectangular box, feedback output, jo data ya response aaha hoga wo object hoga.

|→ partition → swimlane. puri diagram draw karke partition karenge or check karenge user ki activities or line draw kardenge.

Rakesymbol Activity ki andar sub activities hoti hai uain, Subactivities mein Rakesymbol use hota ⫪ Deliver order k andar or bhi activities aengi ikn Rakesymbol k through ham isko high level bnadenge.

Merge: multiple flows ko join karta. ◇

merge mein donon flows ka fullfill hona lazmi nhi hai or merg join mein ~~ek~~ donon flow chalenge tw ~~inflo~~ outflow chalega.

Signal: agr kisi activity ka wait karna hai we will define the time by using signal (Timer ⧖), response ka wait karna hai 5sec. ~~we~~ hm isko define karenge. / loading.

internal intervention (human / System) kisi bhi activity ki waja se rouss ko rakna pare. ∑ ] ye symbo signal use hoga. (Interruption)

Bright

Date →

highlevelactivitydiagram: main points ko define kar rhe hote bas.p ⊕

Regions of Activity Diagram.

"Design Principles"

⊕ Tonnel Vision → srf problem ko solve nhi karna hai baqi factors pr bhi dehan dena hai. / Gui pr dehan / nonfunctionalite means k architecture mein msla hai.

⊕ Effectively kaam kren existing solution ko modify kar skte na k Scratch se kaam shuru karen. (reinvent the wheel)

⊕ minimize the intelectual distance. (real world mein dikh tha ho)

⊕ uniformity honi chahye sem GUI sem design everthing should be same among the team.

⊕ changes ko accomodate kare apka design. ⊕ Error: srf particular part ko affect kare.

⊕· Design is solution but not a working one, cause design is blueprint

⊕ design ko evaluate karne k liye kuch quality guidelines hote.

⊕ Semantic errors → class diagram mein galat relationship bnehote.

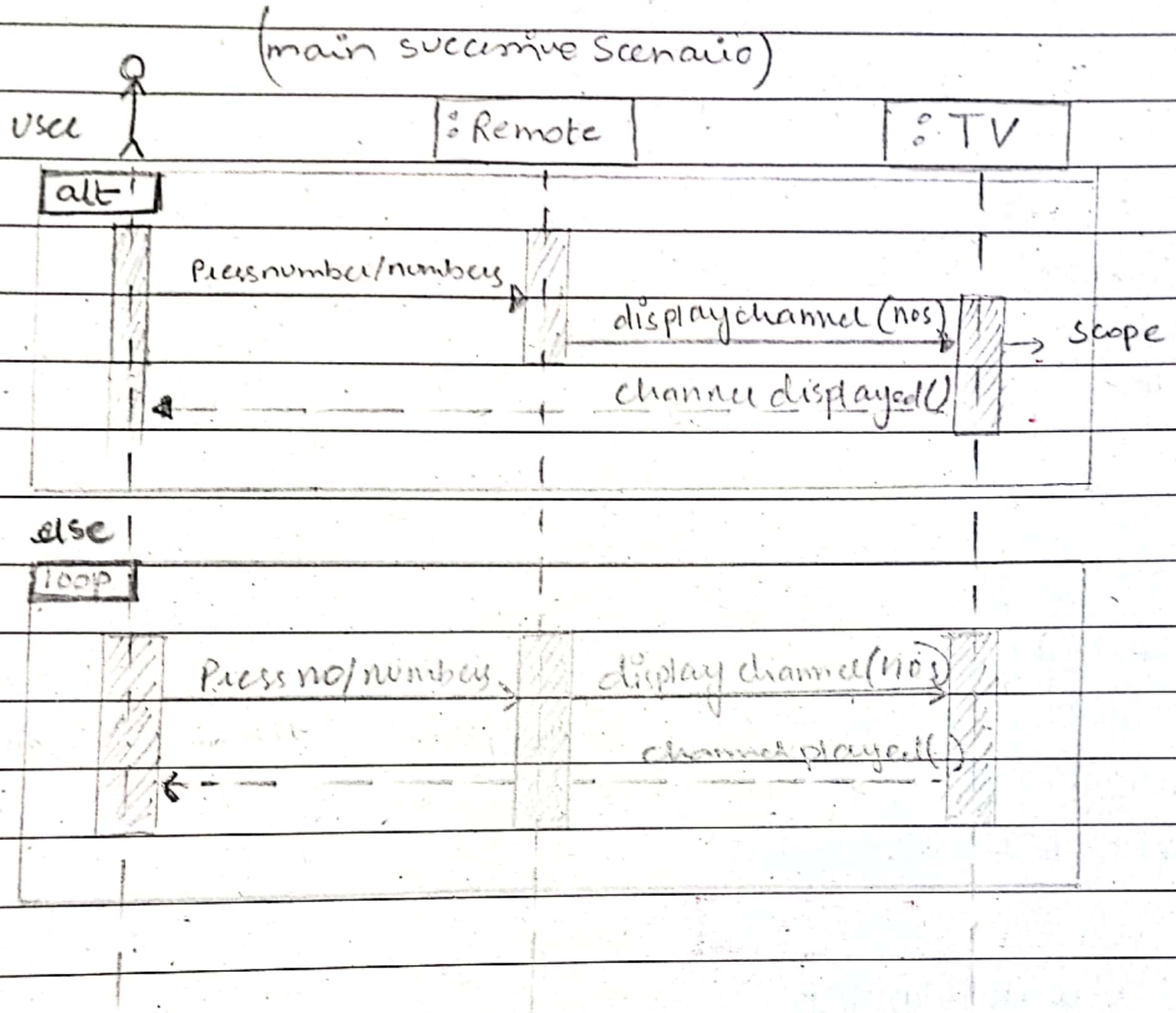
Quality Guideline: ftr → recorder, reviewer, producer.

fundamental Ind. → high cohesion↑ low coupling↓ → Ideal Solution.

Cohesion:

# "Sequence Diagram"

Date



Fragment

- loop
- alt
  if, else
- opt
  if

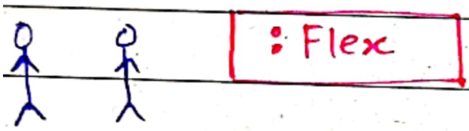
## "Generate Sequence Diagram"

Date

jahan se request jati woheen add karte.

System Sequence Diagram :

system ka naam box mein likh den.

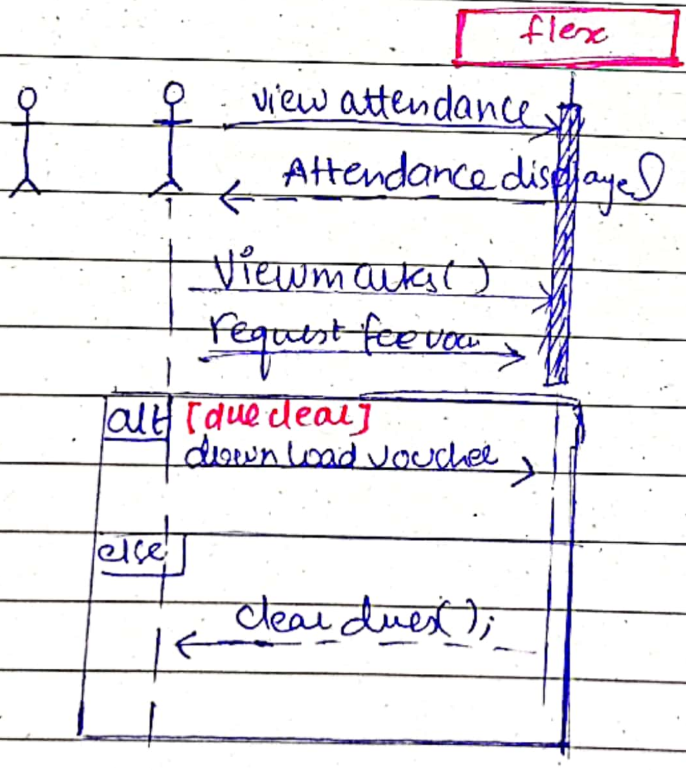


- SSD -> System Seq. Diagram.

multiple Actors are allowed.

Show high level communication.

functionalities define nhi karte.



## (SSD)

Date________

○ system sequence diagram hai usecase ka alg banta hai.

○ System is a block box.

○ Scope aa bhi sakta or nhi bhi.
mandatory in ~~system~~ sequence diagram not in SSD

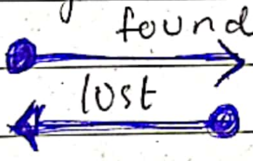
○ Clearity dene k liye kisi message ko verb se ~~ko~~ start karna. for ex.
—. makepayment.

○ message should not be specific but generic. for example.
enterItem ✓ → scan ✗

static → high+low level
behavioural → activity, sequence, scenario.

## loss and found:

receiver or sender identify na hon ya missing hon tu agr
● ye ho tu yani msg found



## Collaboration Diagram.

· link connects two objects (——), every message has its own seq n

message → The arrow shows the direction of flow. between objec

→ messages also contain parameters. but it is not necessary all the time.

⊙ system sequence diagram hai usecase ka alg banta hai.

⊙ System is a block box.

⊙ Scope aa bhi sakta or nhi bhi.
mandatory in ~~system~~ sequence diagram not in SSD

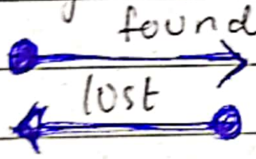
⊙ Clarity dene k liye kisi message ko verb se ~~ko~~ start karna. for ex.
—. makepayment.

⊙ message should not be specific but generic. for example.
enterItem ✓ → scan ✗

Static → high + low level
behavioural → activity, sequence, scenario.

loss and found:

receiver or sender identify na hon ya missing hon tw agr
● ye ho tw yani msg found.



Collaboration Diagram.

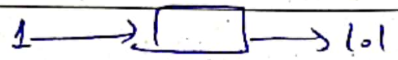
· link connects tw objects (——), every message has its own seq no.
$\overrightarrow{\text{message}}$ → The arrow shows the direction of flow. between objects
⇒ messages also contain parameters. but it is not necessary all the time.

③ Identify the object.

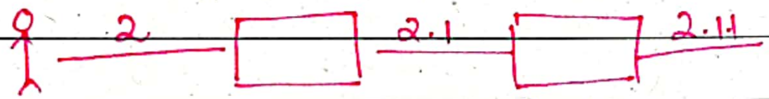
⊙ Pass the msg between objects.

③ Show direction through arrow.

⇒ Sequence number. agr 1 hai or uska flow hi continue horha ho "object se jo msg niklega wo usi ka subpart hoga.



⊙ Input message ka output bhi prependent hoga.



first (main) msg ko number nhi karte hain.

⊙ how to "make Payment" collaboration Diagram:

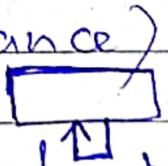
- jis Object ki tarah msg input ka arrow jarha hota hai wo us obj ko call karo nhi hoti.

⊙ jahan kch create horaha yani object instantiate horaha.

⊙ Sale → represents class , :Sale → object of class sale

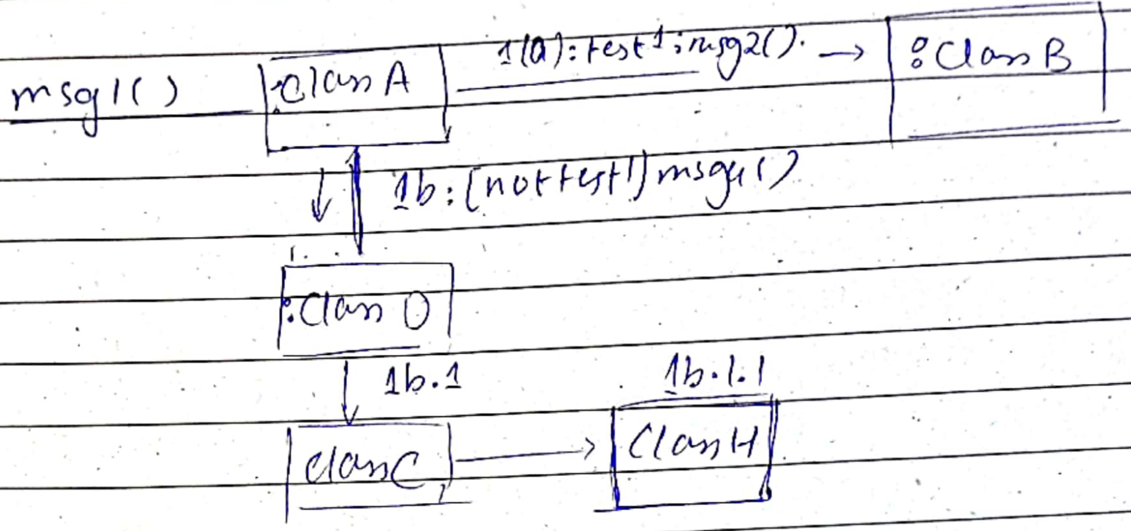
⊙ S1: Sa → class ka ek object hai or uska ek specific object bna rahe. (named istance)

recursion → itself calling → self call



⊙ Collaboration diagram mein lost and found ki notation nhi banti.

## Collabaration Diagram Notation Conditional Messages.



## Mutually Enclusive Conditional Paths.

⊙ Basic Scenaio

⊙ Theory