DATE
PAGE

OOMD

Unit 3

State Model - It describe the life of the object over a period of time.

- It is sequence of activities, events, actions.
- It is combination of several state diagram.

Components

① Events. are incident that take place at a particular time.

Types of Events

① Signal event.
→ Sending and receiving of information from one object to another.

② change event
occur when certain condition is satisfied

③ Time event
occur at a specified time or after some specified time.

State - It represent the value of the attribute of an object at a particular time.
→ situation of object at a particular time

transition and condition.

When object changes its current state to another state

The occurrence of the transition also depends on the guard condition.

Advanced State Modeling.

Nested State diagram. → Used to Model the complex problem.

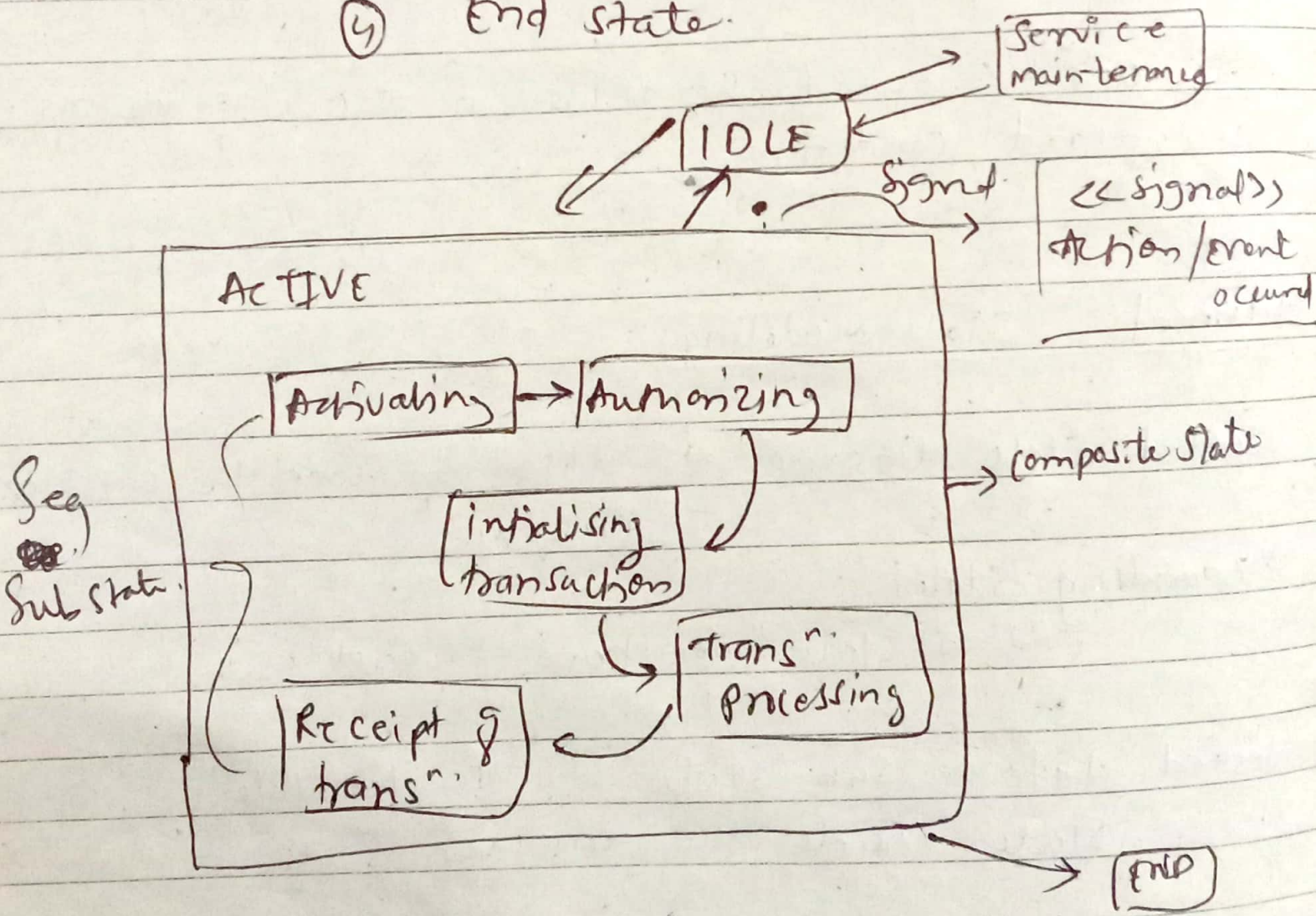
Expanding States
↳ a state have have Multiple Sub state.

Nested state → Sub state, Composite State,
State inside the another State.

two types of Substate.

1) Sequential Substate → proper flow of activities actions or operation.

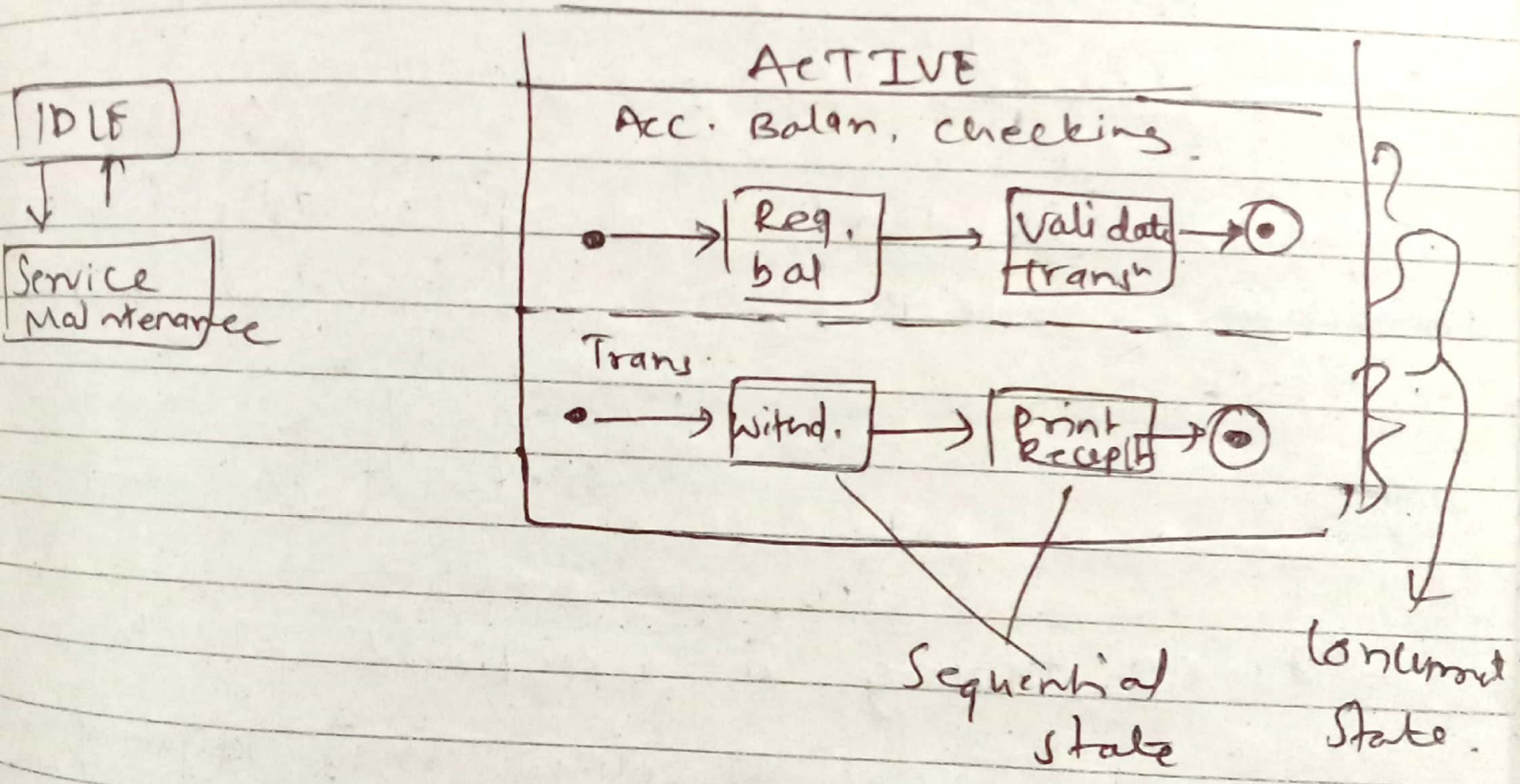
Any Software system normally execute in 4 basic steps
1. Idle State
2. Activating State
3. Active State
4. End state.



(e) Concurrent Substate.

↳ two or more State Machine that Execute concurrently. then we uses Concurrent State.

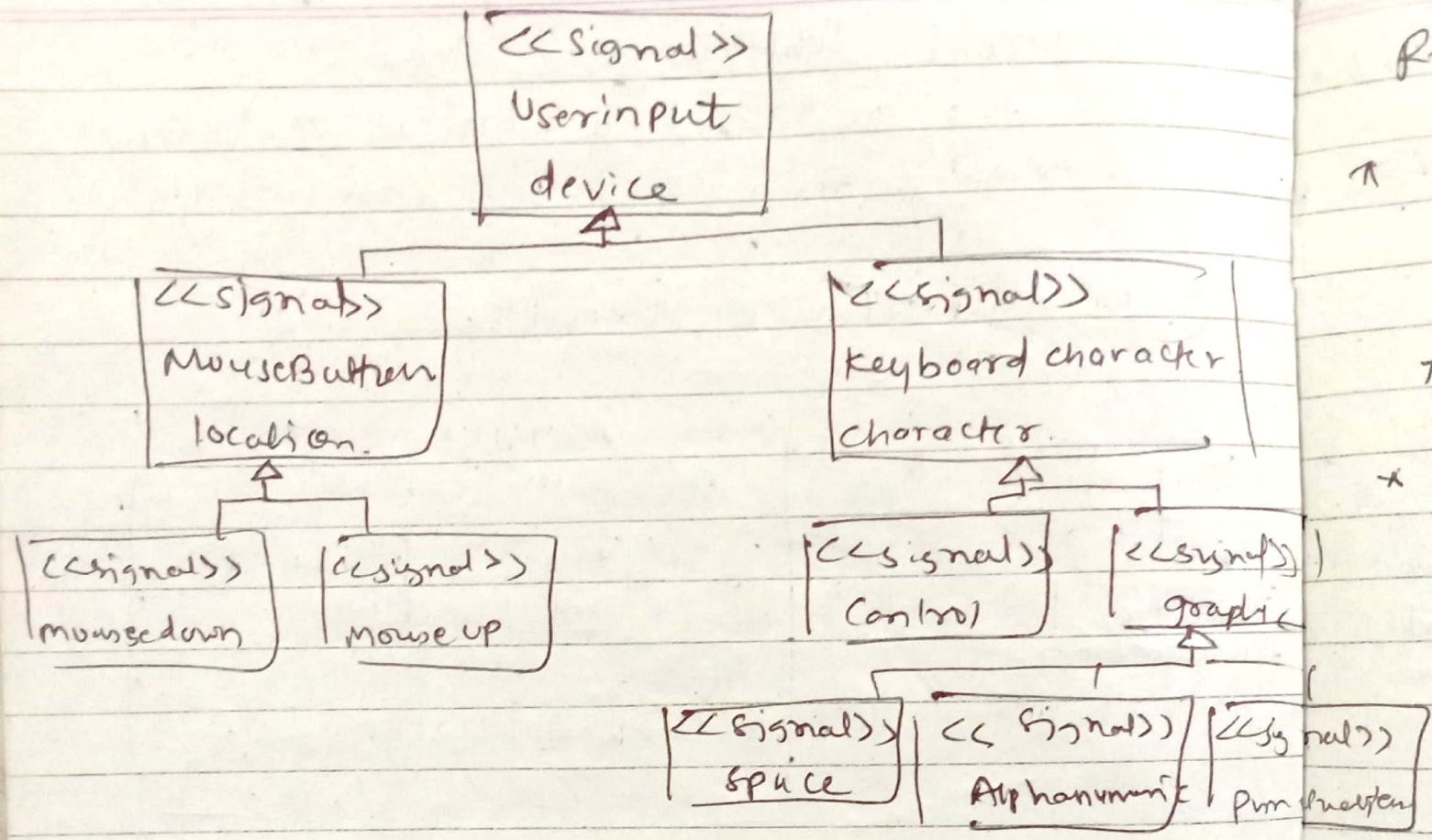
→ parallel flow of execution.



Signal Generalization.

→ We can organize Signal into a generalization hierarchy with inheritance of Signal attribute

→ level of abstraction.



eg on typing a in keyboard. will trigger Alphanumeric as well as signal keyboard character.

Concurrency

* State model support concurrency
* Object can act & change state independently one another
* Parallel execution.

ATM State model

Relation of class Model & State Model.

↑ State Model Specifies allowable Sequences of changes to objects from the class model.

↑ a single object can have diff state over time.

* class model is attribute and behaviour of class

Interaction Modelling

① Use case Model.

→ How user interact with system in order to solve a problem.
→ It model the behaviour of system and, capture the requirement of system.
→ It describe the high level function & scope.

Basic Components

① Use case :- It is how the system is used
→ All the scenario that collectively work to achi

→ A simple use case is a single interaction blw actor & the system.

(2) Actors → External user of the system. who communicate with system.

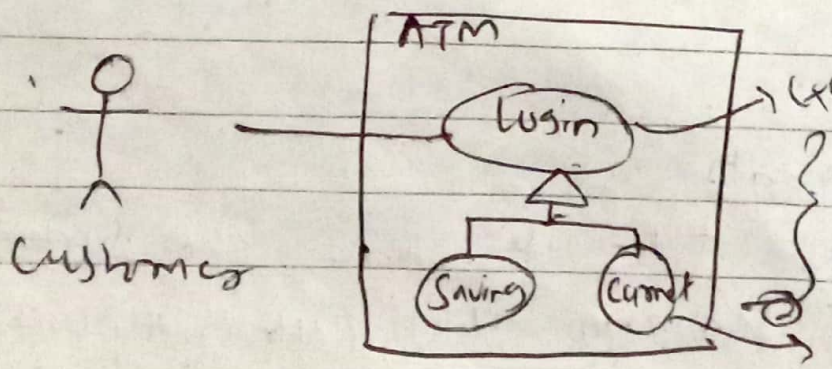
(3) System Boundary (subject)

* System designer should define the things which are external to the system and the things that are part of the system.

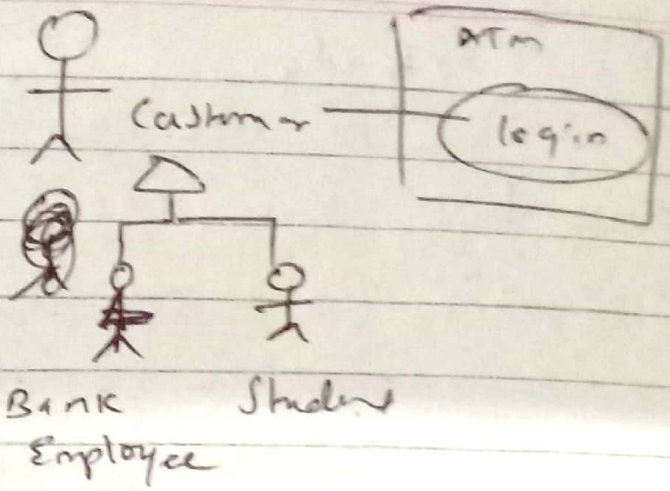
(4) Use case Relationship.

(1) Association → use case / Actor.
one actor can associated with one or more usecases.

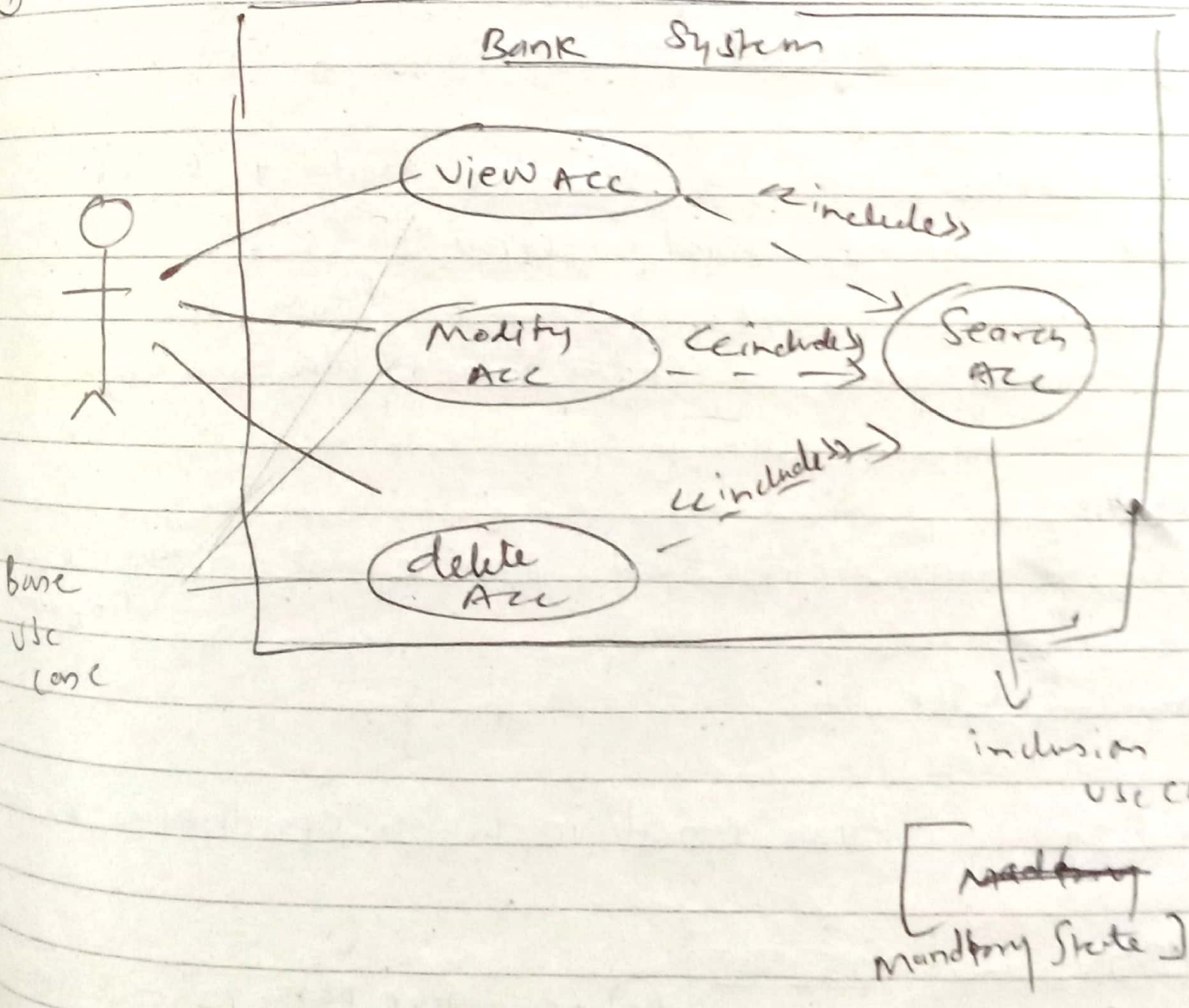
(2) Use case Generalization.
↳ use for simplification of use cases



③ Actor Generalization

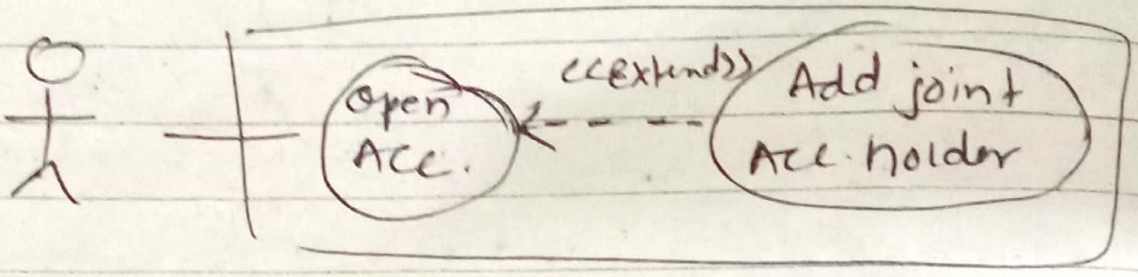


④ <<include>>



(5) <<extends>>

The use case is optional and come after the base use case



Sequence Model → ordering of MSg.

interaction Model → Sequence Model + Collaboration Model
- Set of objects, relationships and message sent amongst them.
- Communication & coordination b/w two objects.

Notation

Lifeline - an individual participant in the sequence diagram

Actor - User

Activation :- time period in which operation is performed

Message - interaction

1. call MSg
2. return MSg
3. Self MSg
4. Recursive MSg.
5. Create MSg
6. Destroy MSg
7. Duration MSg.

Activity Model

→ demostrate the flow of control within the system rather than implementation
→ flowchart of object oriented system
→ workflow from one activity to another

① Activities.
↳ categorization of behaviour into one or more actions.

② Activity partition.
③ forks
④ Join