EXPERIMENT NO: 6

AIM: To develop activity and state machine diagram.

Theory:

Activity diagram:

Activity diagram is another important behavioural diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modelling the flow from one activity to another activity. It focuses on the execution and flow of the behaviour of a system instead of implementation. It is also called **object-oriented flowchart**.

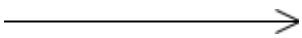
To draw an activity diagram, one must understand and explore the entire system. All the elements and entities that are going to be used inside the diagram must be known by the user. The central concept which is nothing but an activity must be clear to the user. After analysing all activities, these activities should be explored to find various constraints that are applied to activities. If there is such a constraint, then it should be noted before developing an activity diagram.

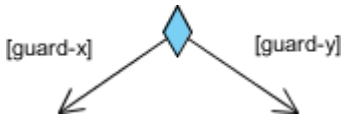
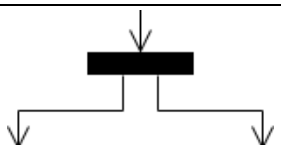
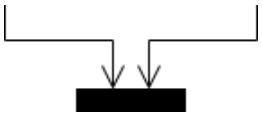
All the activities, conditions, and associations must be known. Once all the necessary things are gathered, then an abstract or a prototype is generated, which is later converted into the actual diagram.

Following rules must be followed while developing an activity diagram,

- All activities in the system should be named.
- Activity names should be meaningful.
- Constraints must be identified.
- Activity associations must be known.

Activity Diagram Notation Summary:

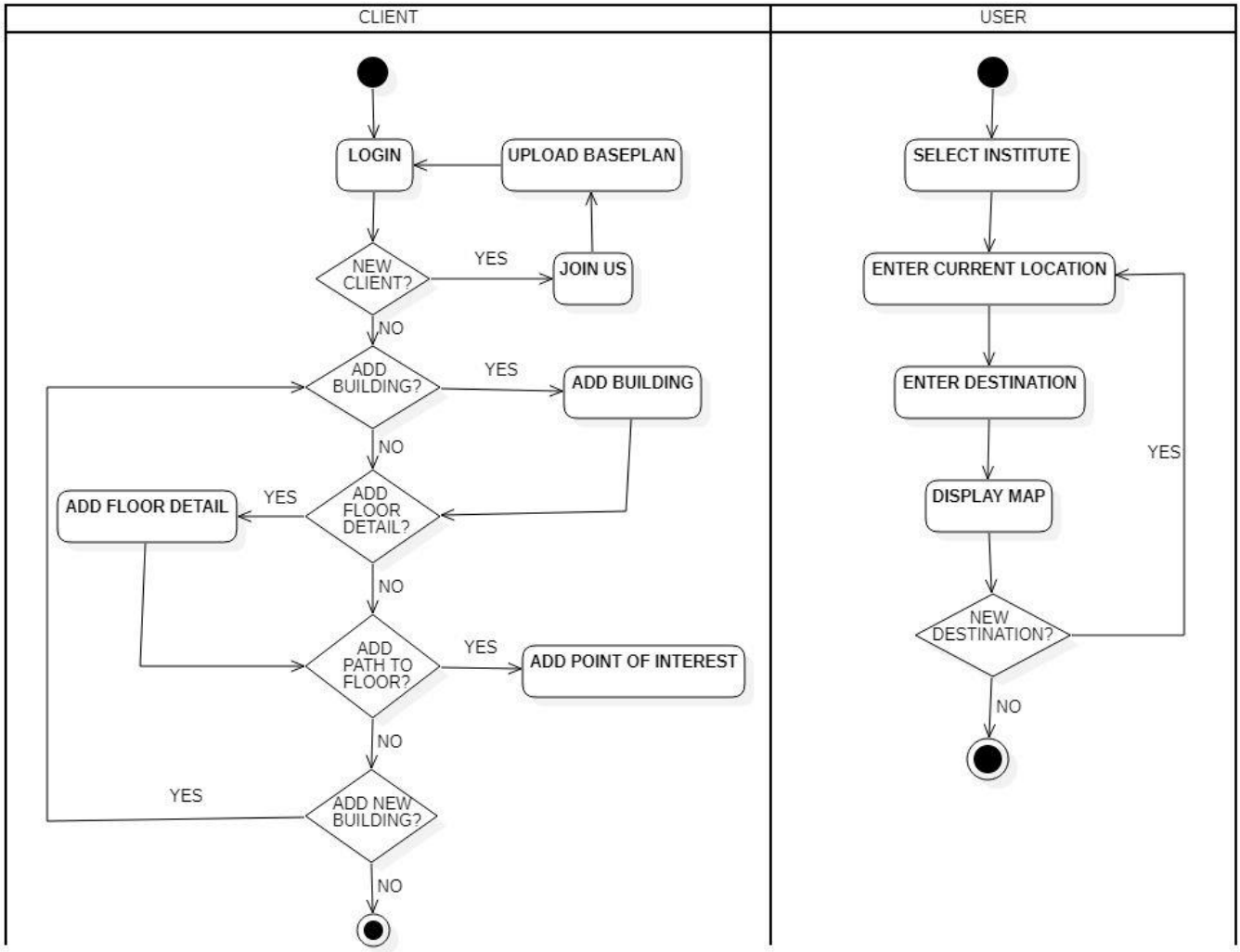
Notation Description	UML Notation
Activity: Is used to represent a set of actions.	
Action: A task to be performed.	
Control Flow/Object flow: Shows the sequence of execution OR Show the flow of an object from one activity (or action) to another activity (or action).	
Initial Node: Portrays the beginning of a set of actions or activities.	
Activity Final Node: Stop all control flows and object flows in an activity (or action).	

Object Node: Represent an object that is connected to a set of Object Flows.	
Decision Node: Represent a test condition to ensure that the control flow or object flow only goes down one path.	
Merge Node: Bring back together different decision paths that were created using a decision-node.	
Node: Split behaviour into a set of parallel or concurrent flows of activities (or actions).	
Join Node: Bring back together a set of parallel or concurrent flows of activities (or actions).	
Swimlane and Partition: A way to group activities performed by the same actor on an activity diagram or to group activities in a single thread.	

Applications:
Activity diagram is used to model business processes and workflows. These diagrams are used in software modelling as well as business modelling. Most commonly activity diagrams are used to,

- Model the workflow in a graphical way, which is easily understandable.
- Model the execution flow between various entities of a system.
- Model the detailed information about any function or an algorithm which is used inside the system.
- Model business processes and their workflows.
- Capture the dynamic behaviour of a system.
- Generate high-level flowcharts to represent the workflow of any application.
- Model high-level view of an object-oriented or a distributed system.

ACTIVITY DIAGRAM:



State Diagram:

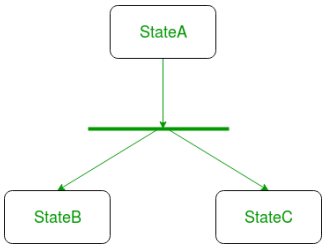
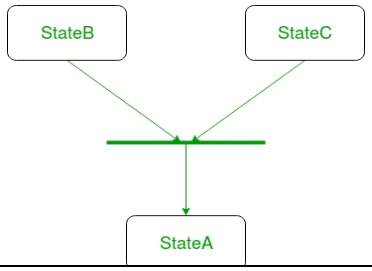
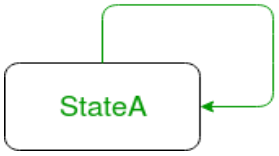
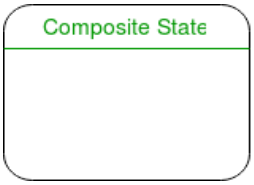
A **state diagram** is used to represent the condition of the system or part of the system at finite instances of time. It's a **behavioural** diagram and it represents the behaviour using finite state transitions. State diagrams are also referred to as **State machines** and **State-chart Diagrams**. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behaviour of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams. We prefer to model the states with three or more states.

Uses of state chart diagram –

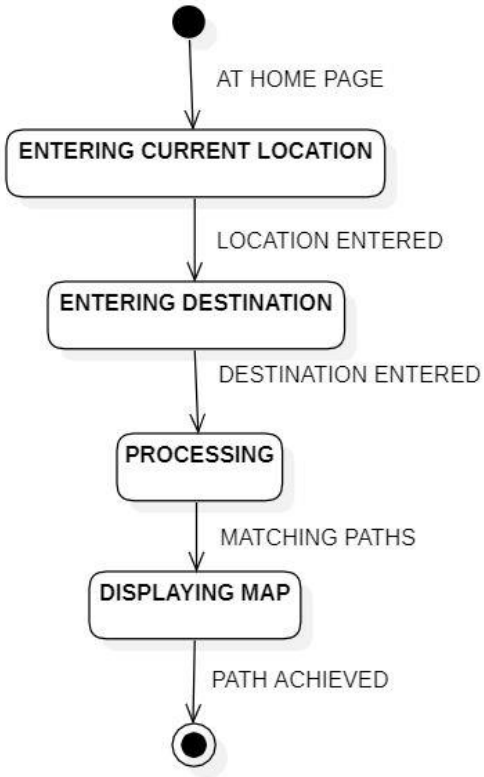
- We use it to state the events responsible for change in state (we do not show what processes cause those events).
- We use it to model the dynamic behaviour of the system.
- To understand the reaction of objects/classes to internal or external stimuli.

Basic components of a state chart diagram –

Notation Description	UML Notation

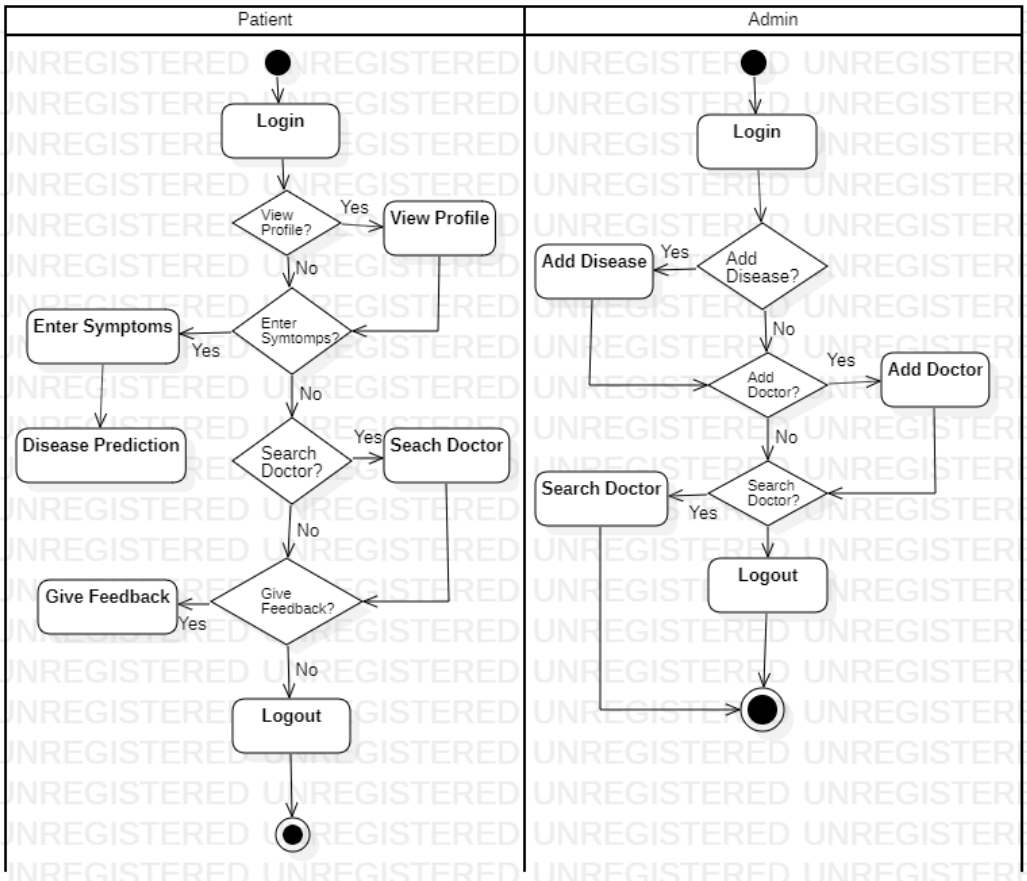
Initial state – We use a black filled circle represent the initial state of a System or a class.	
Transition – We use a solid arrow to represent the transition or change of control from one state to another. The arrow is labelled with the event which causes the change in state.	
State – We use a rounded rectangle to represent a state. A state represents the conditions or circumstances of an object of a class at an instant of time.	
Fork – We use a rounded solid rectangular bar to represent a Fork notation with incoming arrow from the parent state and outgoing arrows towards the newly created states. We use the fork notation to represent a state splitting into two or more concurrent states.	
Join – We use a rounded solid rectangular bar to represent a Join notation with incoming arrows from the joining states and outgoing arrow towards the common goal state. We use the join notation when two or more states concurrently converge into one on the occurrence of an event or events.	
Self-transition – We use a solid arrow pointing back to the state itself to represent a self-transition. There might be scenarios when the state of the object does not change upon the occurrence of an event. We use self-transitions to represent such cases.	
Composite state – We use a rounded rectangle to represent a composite state also. We represent a state with internal activities using a composite state.	
Final state – We use a filled circle within a circle notation to represent the final state in a state machine diagram.	

STATE DIAGRAM:



CONCLUSION: Thus we have developed Activity and State Diagram successfully.

ACTIVITY DIAGRAM :



STATE MACHINE DIAGRAM :