**EVENT MANAGEMENT SYSTEM GROUP TE C21**

**2103080 Hriday Kampani**

**2103072 Suyash kabra**

**Aim: To design state and activity diagram**

**State Chart Diagram**

State chart diagrams model the dynamic behavior of individual classes or any other kind of object. They show the sequences of states that an object goes through, the events that cause a transition from one state to another, and the actions that result from a state change.

State chart diagrams are closely related to activity diagrams. The main difference between the two diagrams is state chart diagrams are state centric, while activity diagrams are activity centric. A state chart diagram is typically used to model the discrete stages of an object’s lifetime, whereas an activity diagram is better suited to model the sequence of activities in a process.

Each state represents a named condition during the life of an object during which it satisfies some condition or waits for some event. A state chart diagram typically contains one start state and multiple end states. Transitions connect the various states on the diagram. As with activity diagrams, decisions, synchronizations, and activities may also appear on state chart diagrams.

# State

# Definition

A state represents a condition or situation during the life of an object during which it satisfies some condition or waits for some event. Each state represents the cumulative history of its behavior.

# Graphical Depiction

The state icon appears as a rectangle with rounded corners and a name (Wait). It also contains a compartment for actions:



# Naming

The name of a state should be unique to its enclosing class, or if nested, within the state. All state icons with the same name in a given diagram represents the same state.

# Actions

Actions on states can occur at one of four times:

· on entry

· on exit

· do

· on event.

An on-event action is similar to a state transition label with the following syntax:

event(args)[condition] : the Action

You must add actions through the Action Specification. States may also appear on activity diagrams.

# Start state

A start state (also called an "initial state") explicitly shows the beginning of a workflow on an activity diagram or the beginning of the execution of a state machine on a state chart diagram. You can have only one start state for each state machine because each workflow/execution of a state machine begins in the same place. If you use multiple activity and/or state chart diagrams to model a state machine, the same start state can be placed on the multiple diagrams. When you model nested states or nested activities, one new start state can be created in each context.

Normally, only one outgoing transition can be placed from the start state. However, multiple transitions may be placed on a start state if at least one of them is labeled with a condition. No incoming transitions are allowed.

You can label start states, if desired. State Specifications are associated with each start state.

# Graphical Depiction

The start state icon is a small, filled circle that may contain a name (Begin Process):



# End State

An end state represents a final or terminal state on an activity diagram or state chart diagram. Place an end state when you want to explicitly show the end of a workflow on an activity diagram or the end of a state chart diagram. Transitions can only occur into an end state; however, there can be any number of end states per context.

You can label end states, if desired. State Specifications are associated with each end state.

# Graphical Depiction

The end state icon is a filled circle inside a slightly larger unfilled circle that may contain a name (End Process):



# State Transition

# Definition

A state transition indicates that an object in the source state will perform certain specified actions and enter the destination state when a specified event occurs or when certain conditions are satisfied. A state transition is a relationship between two states, two activities, or between an activity and a state.

You can show one or more state transitions from a state as long as each transition is unique. Transitions originating from a state cannot have the same event, unless there are conditions on the event.

# Graphical Depiction

The icon for a state transition is a line with an arrowhead pointing toward the destination state or activity:



Naming

You should label each state transition with the name of at least one event that causes the state transition. You do not have to use unique labels for state transitions because the same event can cause a transition to many different states or activities.

Transitions are labeled with the following syntax:

event (arguments) [condition] / action ^ target.sendEvent (arguments)

Only one event is allowed per transition, and one action per event.

Events, conditions and actions must be added by editing the label or through the State Transition Specification.

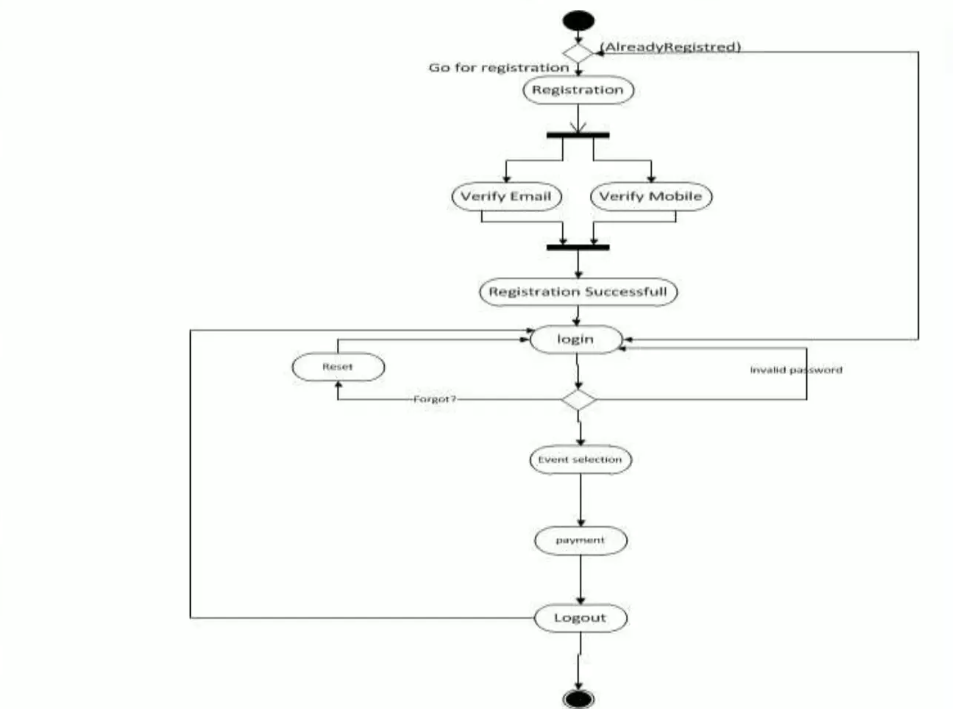
# Nested States

States may be nested to any depth level. Enclosing states are referred to as super states, and everything that lies within the bounds of the super state is referred to as its contents. Nested states are called sub states.

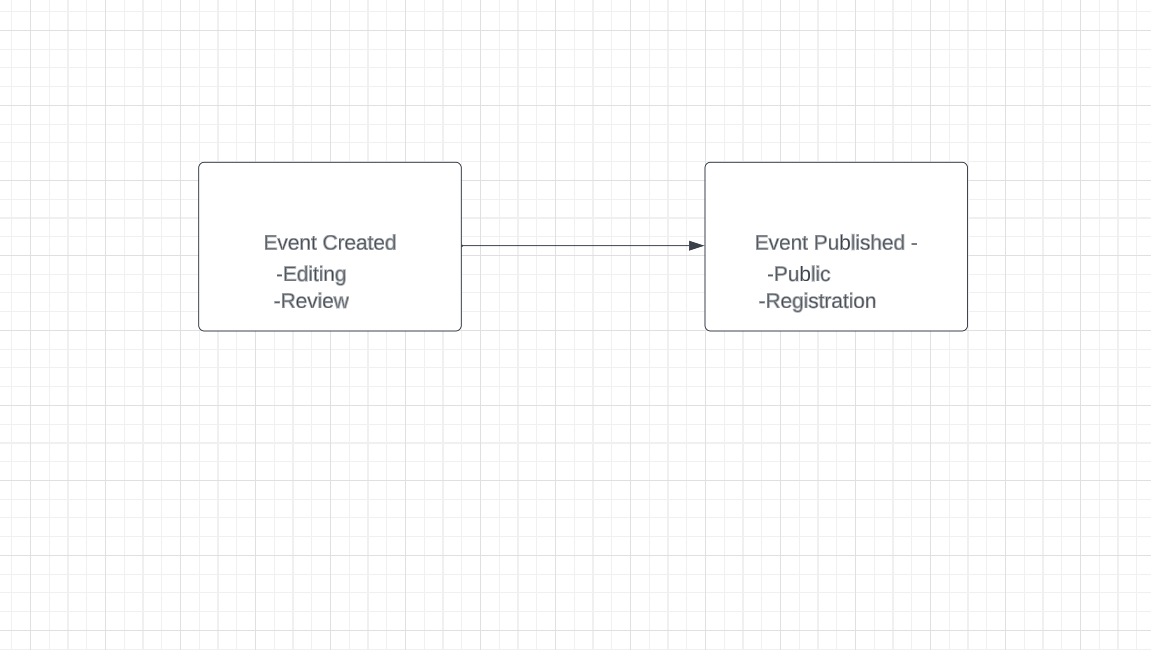
**In** this way we have studied state chart diagram.

**Activity Diagram**

Activity diagrams provide a way to model the workflow of a business process. You can also use activity diagrams to model code-specific information such as a class operation. Activity diagrams are very similar to a flowchart because you can model a workflow from activity to activity. An activity diagram is basically a special case of a state machine in which most of the states are activities and most of the transitions are implicitly triggered by completion of the actions in the source activities. The main difference between activity diagrams and state charts is activity diagrams are activity centric, while state charts are state centric. An activity diagram is typically used for modeling the sequence of activities in a process, whereas a state chart is better suited to model the discrete stages of an object’s lifetime.



**STATE DIAGRAM**



**Procedure:**

Activity diagrams can model many different types of workflows. For example, a company could use activity diagrams to model the flow for an approval of orders or to model the paper trail of invoices. An accounting firm could use activity diagrams to model any number of financial transactions. A software company could use activity diagrams to model a software development process.

Each activity represents the performance of a group of actions in a workflow. Once the activity is complete, the flow of control moves to the next activity or state through a transition. If an outgoing transition is not clearly triggered by an event, then it is triggered by the completion of the contained actions inside the activity. A unique activity diagram feature is a swim lane that defines who or what is responsible for carrying out the activity or state. It is also possible to place objects on activity diagrams. The workflow stops when a transition reaches an end state.

**Conclusion:**

Thus,The activity diagram shows use case as it progress from start to finish.Activity diagrams are used to show workflow in parallel and conditionally.  They are useful when working out the order and concurrency of a sequential algorithm, when analysing the steps in a business process and when working with threads.