Assignment 7

Title:

Prepare and Implement state model

Problem Statement:

- Prepare a State Model.
- Identify States and events for your system.
- Study state transitions and identify Guard conditions.
- Draw State chart diagram with advanced UML 2 notations.
- Implement the state model with a suitable OO language

Objective:

- To Identify States Transitions, events in the system flow.
- Draw State Diagram and Implement Model.

Theory:

State Machine Diagram:

State machine diagrams typically are used to describe state-dependent behavior for an object. **An object responds differently to the same event depending on what state it is in**. State machine diagrams are usually applied to objects but can be applied to any element that has behavior to other entities such as: actors, use cases, methods, subsystems systems and etc. and they are typically used in conjunction with interaction diagrams (usually sequence diagrams).

Characteristics of State Machine Notations

There are several characteristics of states in general, regardless of their types:

- A state occupies an interval of time.

- A state is often associated with an abstraction of attribute values of an entity satisfying some condition(s).
- An entity changes its state not only as a direct consequence of the current input, but it is also dependent on some past history of its inputs.

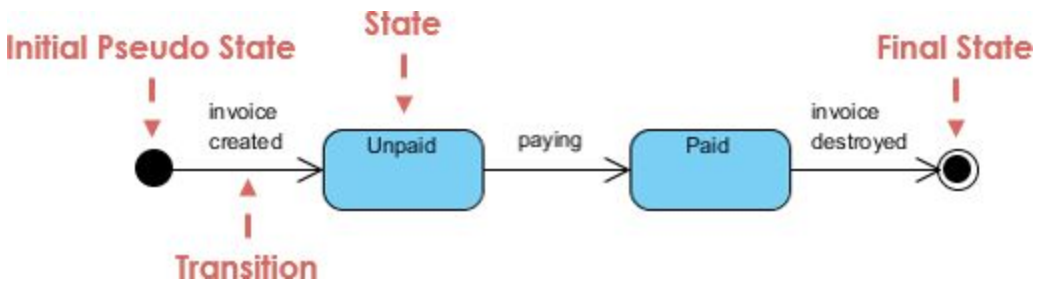
State

A state is a constraint or a situation in the life cycle of an object, in which a constraint holds, the object executes an activity or waits for an event.

A state machine diagram is a graph consisting of:

- States (simple states or composite states)
- State transitions connecting the states

Example:



Characteristics of State

- States represent the conditions of objects at certain points in time.
- Objects (or Systems) can be viewed as moving from state to state
- A point in the lifecycle of a model element that satisfies some condition, where some particular action is being performed or where some event is waited

Initial and Final States

- The **initial state** of a state machine diagram, known as an initial pseudo-state, is indicated with a solid circle. A transition from this state will show the first real estate.
- The **final state** of a state machine diagram is shown as concentric circles. An open loop state machine represents an object that may terminate before the system terminates, while a closed loop state machine diagram does not have a final state; if it is the case, then the object lives until the entire system terminates.

Events

An event signature is described as Event-name (comma-separated-parameter-list). Events appear in the internal transition compartment of a state or on a transition between states. An event may be one of four types:

1. Signal event - corresponding to the arrival of an asynchronous message or signal
2. Call event - corresponding to the arrival of a procedural call to an operation
3. Time event - a time event occurs after a specified time has elapsed
4. Change event - a change event occurs whenever a specified condition is met

Characteristics of Events

- Represents incidents that cause objects to transition from one state to another.
- Internal or External Events trigger some activity that changes the state of the system and of some of its parts
- Events pass information, which is elaborated by Objects operations. Objects realize Events
- Design involves examining events in a state machine diagram and considering how those events will be supported by system objects

Transition

Transition lines depict the movement from one state to another. Each transition line is labeled with the **event** that causes the transition.

- Viewing a system as a set of states and transitions between states is very useful for describing complex behaviors
- Understanding state transitions is part of system analysis and design
- A Transition is the movement from one state to another state
- Transitions between states occur as follows:

    1. An element is in a source state
    2. An event occurs
    3. An action is performed
    4. The element enters a target state
- Multiple transitions occur either when different events result in a state terminating or when there are guard conditions on the transitions
- A transition without an event and action is known as automatic transitions

Actions

Action is an executable atomic computation, which includes operation calls, the creation or destruction of another object, or the sending of a signal to an object. An action is associated with transitions and during which an action is not interruptible - e.g., entry, exit

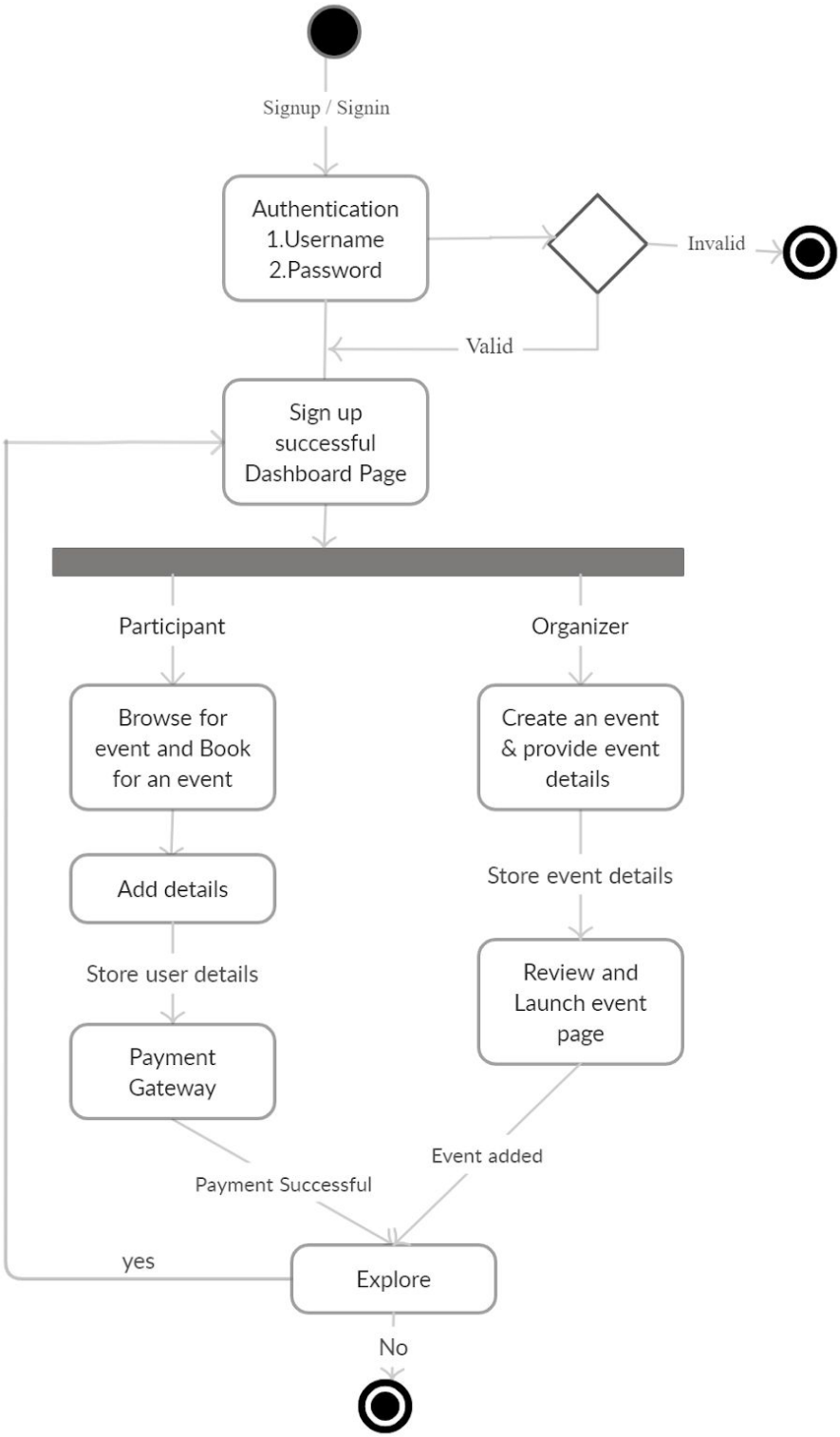
Activity

Activity is associated with states, which is a non-atomic or ongoing computation. Activity may run to completion or continue indefinitely. An Activity will be terminated by an event that causes a transition from the state in which the activity is defined

Characteristics of Action and Activities

- States can trigger actions
- States can have a second compartment that contains actions or activities performed while an entity is in a given state
- An action is an atomic execution and therefore completes without interruption
- Five triggers for actions: On Entry, Do, On Event, On Exit, and Include
- An activity captures complex behavior that may run for a long duration - An activity may be interrupted by events, in which case it does not complete occur when an object arrives in a state.

State Model for Event Registration System



Conclusion:

Thus, in this assignment, we learnt about the state diagram, how to draw the state diagram and implemented the same.