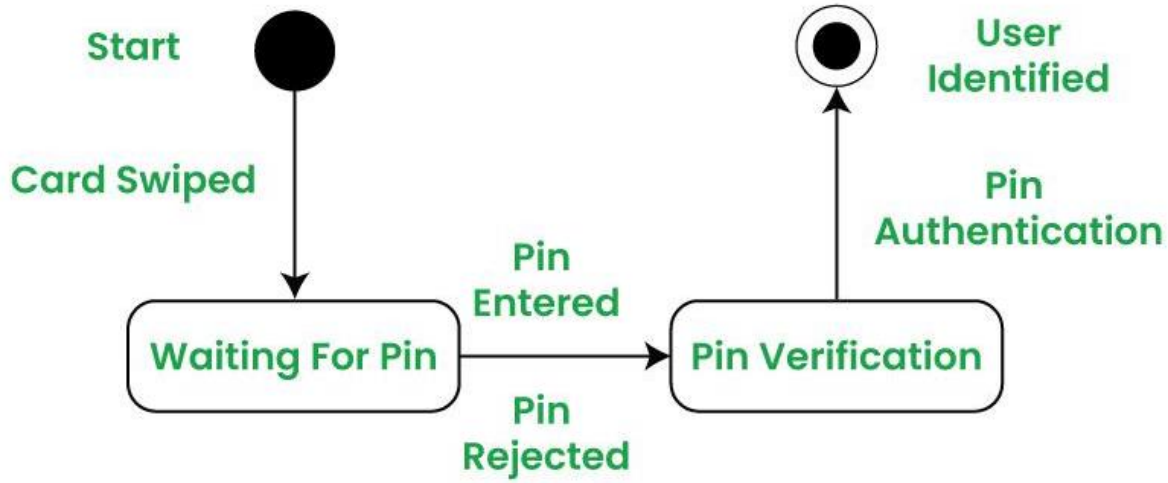
State Machine Diagram

A State Machine Diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behaviour using finite state transitions. It is a UML diagram which is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions.

- State Machine diagrams are also known as **State Diagrams** and **State-Chart Diagrams**. These both terms can be used interchangeably.
- A state machine diagram is used to model the dynamic behaviour of a class in response to time and changing external stimuli (events that cause the system to change its state from one to another).
- We can say that every class has a state but we don't model every class using State Machine diagrams.

Example:



The State Machine Diagram above shows the different states in which the verification sub-system or class exists for a particular system.

Basic Components and Notations of a State Machine Diagram

Below are the basic components and their notations of a State Machine Diagram:

1. Initial state

We use a black filled circle represent the initial state of a System or a Class.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. Composite state

We use a rounded rectangle to represent a composite state also. We represent a state with internal activities using a composite state.

8. Final State

We use a filled circle within a circle notation to represent the final state in a state machine diagram.