

Assignment 7

State Model Diagram

AIM - Design and implementation of the State model diagram.

PROBLEM STATEMENT -

Prepare a State Model.

Identify State and events of the system.

Study State transitions and define Guard conditions.

Draw State Model using UML2.0 Notations.

OBJECTIVE -

To identify state transitions ,events of the system.

To Draw a State Model.

To implement the State Model.

THEORY -

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 behavioral diagram and it represents the behavior 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 behavior 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 statechart 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 behavior of the system .
- To understand the reaction of objects/classes to internal or external stimuli.

Basic components of a statechart diagram –

1.Initial state – We use a black filled circle to represent the initial state of a System or a class.



Figure – initial state notation

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.



Figure – transition

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.



Figure – state notation

4.Fork – We use a rounded solid rectangular bar to represent a Fork notation with incoming arrows 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.

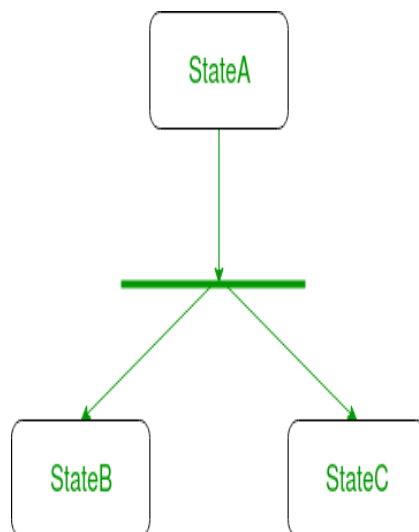


Figure – a diagram using the fork notation

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.

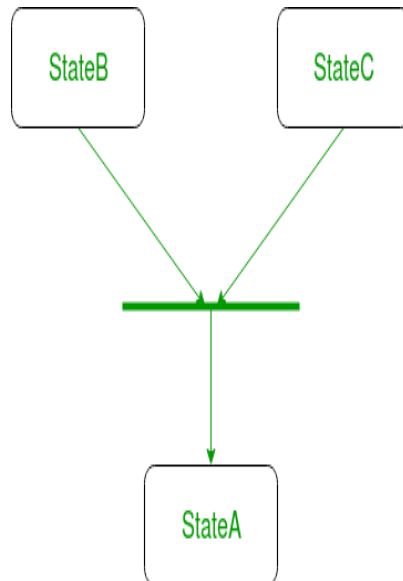


Figure – a diagram using join notation

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.

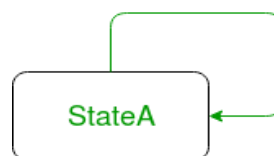


Figure – self transition notation

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.

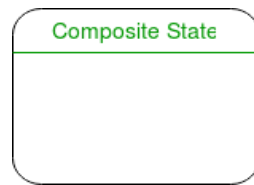
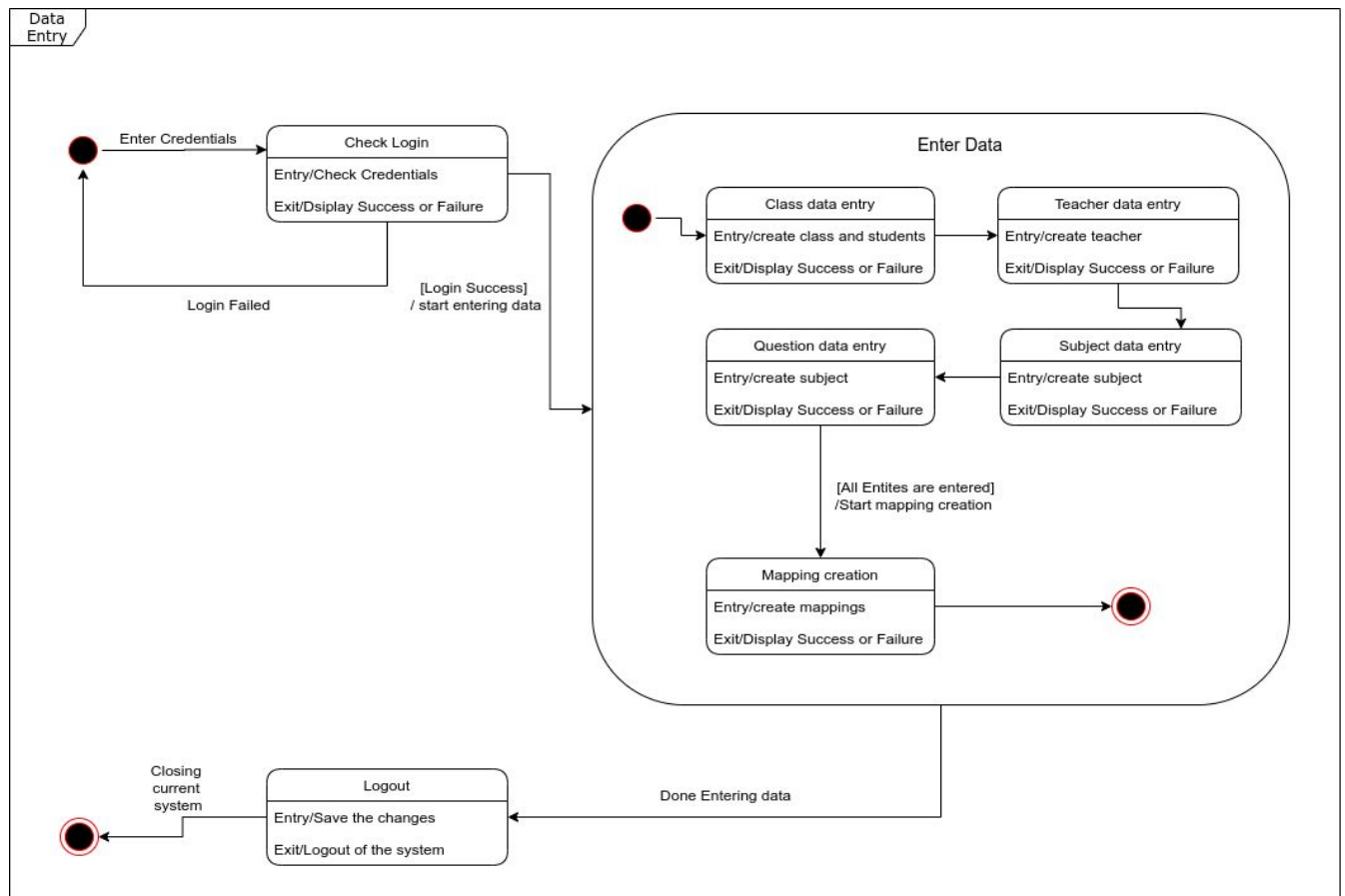


Figure – a state with internal activities

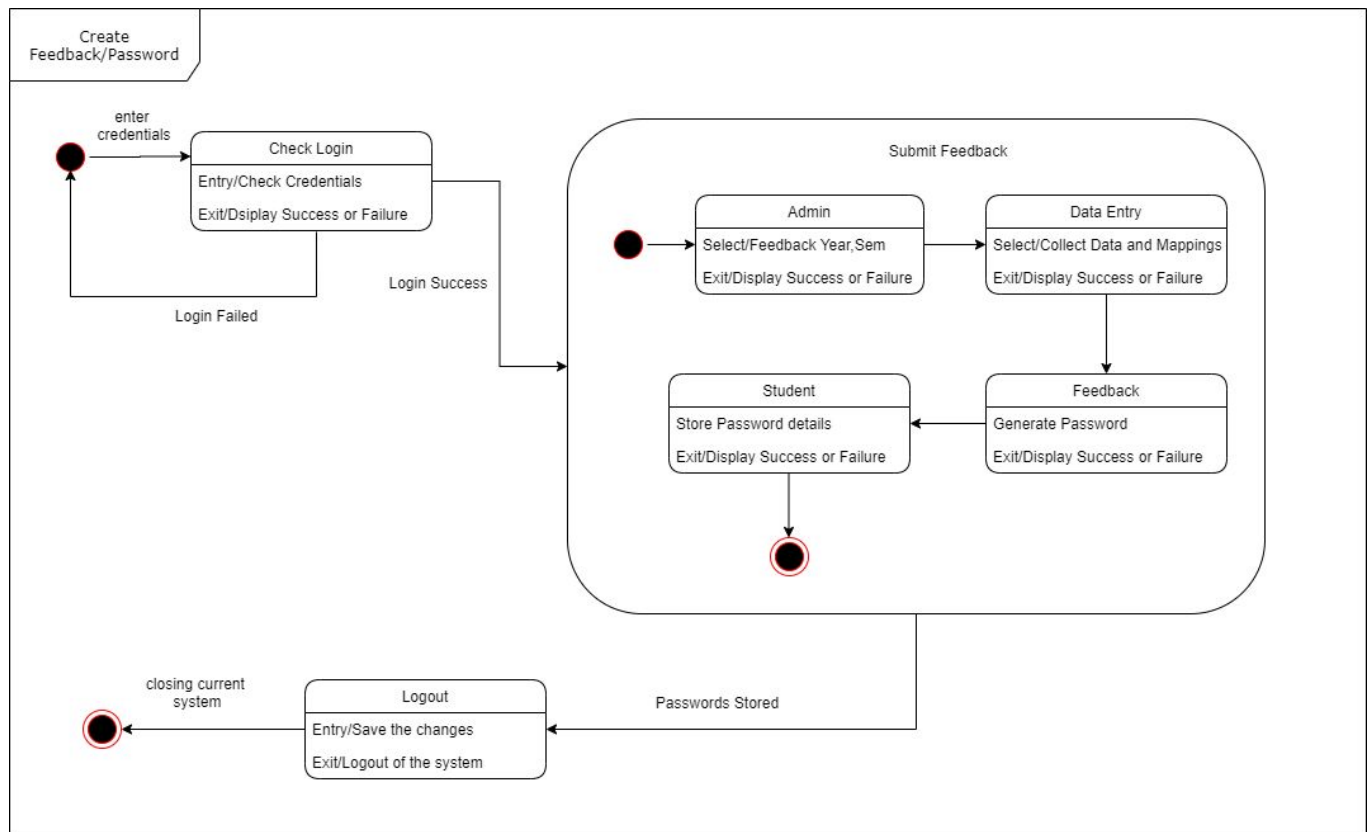
8.Final state – We use a filled circle within a circle notation to represent the final state in a state machine diagram.



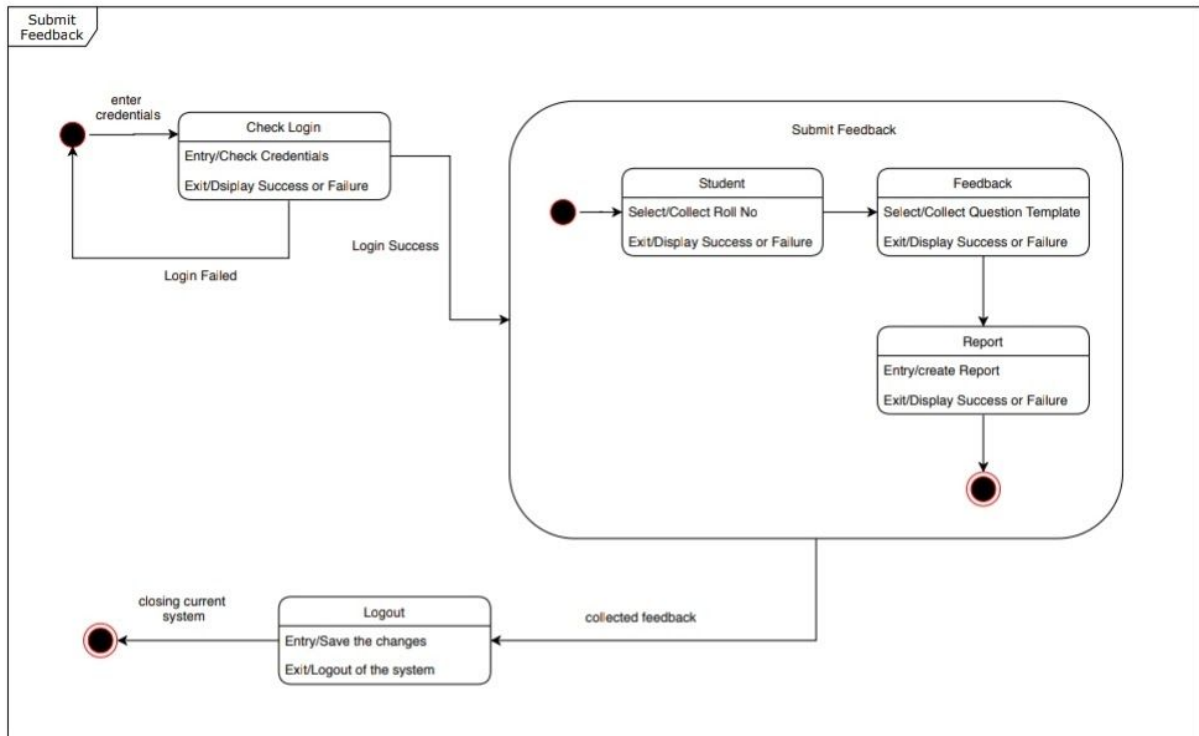
Figure – final state notation



In the data entry scenario, it starts from the login state. After login is successful the state is transitioned to a compound state. This compound state contains subsequent states to create and map the entities. After entering all of the data, the system gets transitioned to the final logout state. And at last the system is closed.



In the feedback/password generation scenario, Login state is the initial state. After successful login, the state is transformed to compound state, which contains subsequent states to enter feedback details and generate passwords. After creating feedback, the system gets transformed to the final logout state



In the submit feedback scenario, Login state is the initial state. After successful login, the state is transformed to compound state, which contains subsequent states to submit feedback and generate report. After submitting feedback, the system gets transformed to the final logout state.

CONCLUSION -

Thus we have prepared a State Model and studied in detail the working of the system/project. Identified different states ,events and used advanced relationships. We drew a State Model using UML2.0 Notations and implemented the State model with a suitable object-oriented language.