**Practical No. 6**

**Aim:** [State chart and Activity Modeling](http://vlabs.iitkgp.ernet.in/se/6/)

**Theory:**

1. **Statechart Diagrams**

A statechart diagram is a pictorial representation of a system, with all it's states, and different events that lead transition from one state to another.

To illustrate this, consider a computer. Some possible states that it could have are: running, shutdown, hibernate. A transition from running state to shutdown state occur when user presses the "Power off" switch, or clicks on the "Shut down" button as displayed by the OS. Here, clicking on the shutdown button, or pressing the power off switch act as external events causing the transition.

Statechart diagrams are normally drawn to model the behaviour of a complex system. For simple systems this is optional.

**Building Blocks of a Statechart Diagram**

**State**

A state is any "distinct" stage that an object (system) passes through in it's lifetime. An object remains in a given state for finite time until "something" happens, which makes it to move to another state.  All such states can be broadly categorized into following three types:

**Initial:** The state in which an object remain when created

**Final:** The state from which an object do not move to any other state [optional]

**Intermediate:** Any state, which is neither initial, nor final

As shown in figure-01, an initial state is represented by a circle filled with black. An intermediate state is depicted by a rectangle with rounded corners. A final state is represented by a unfilled circle with an inner black-filled circle.

IMG_256

**Transition**

Transition is movement from one state to another state in response to an external stimulus (or any internal event). A transition is represented by a solid arrow from the current state to the next state.

1. **Activity Diagrams**

Activity diagrams fall under the category of behavioural diagrams in Unified Modeling Language. It is a high level diagram used to visually represent the flow of control in a system. It has similarities with traditional flow charts. However, it is more powerful than a simple flow chart since it can represent various other concepts like concurrent activities, their joining, and so on. These kind of diagrams are suitable for confirming the logic to be implemented with the business users. These diagrams are typically used when the business logic is complex. In simple scenarios it can be avoided entirely.

**Components of an Activity Diagram**

Below we describe the building blocks of an activity diagram.

**Activity**

An activity denotes a particular action taken in the logical flow of control. An activity is represented with a rounded rectangle.

There are two special type of activity nodes: initial and final. They are represented with a filled circle, and a filled in circle with a border respectively. Initial node represents the starting point of a flow in an activity diagram. There could be multiple initial nodes.

A final node represents the end point of all activities. Like an initial node, there could be multiple final nodes. Any transition reaching a final node would stop all activities.

**Flow**

A flow (also termed as edge, or transition) is represented with a directed arrow. This is used to depict transfer of control from one activity to another.

**Decision**

A decision node, represented with a diamond, is a point where a single flow enters and two or more flows leave. The control flow can follow only one of the outgoing paths. The outgoing edges often have guard conditions indicating true-false or if-then-else conditions. **Merge**

This is represented with a diamond shape, with two or more flows entering, and a single flow leaving out. A merge node represents the point where at least a single control should reach before further processing could continue.

**Fork**

Fork is a point where parallel activities begin. For example, when a student has been registered with a college, he can in parallel apply for student ID card and library card. A fork is graphically depicted with a black bar, with a single flow entering and multiple flows leaving out.

**Join**

A join is depicted with a black bar, with multiple input flows, but a single output flow. Physically it represents the synchronization of all concurrent activities. Unlike a merge, in case of a join all of the incoming controls must be completed before any further progress could be made.

**Note**

UML allows attaching a note to different components of a diagram to present some textual information. The information could simply be a comment or may be some constraint.

The following table shows commonly used components with a typical activity diagram.

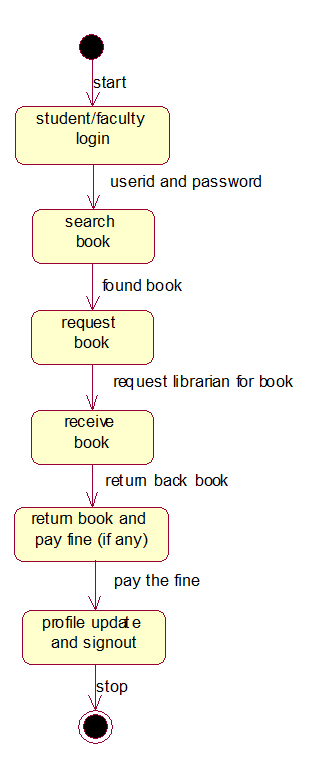
| **Component** | **Graphical Notation** |
| --- | --- |
| Activity | IMG_256 |
| Flow | IMG_257 |
| Decision | IMG_258 |
| Merge | IMG_259 |
| Fork | IMG_260 |
| Join | IMG_261 |
| Note | IMG_262 |

Table-01: Typical components used in an activity diagram

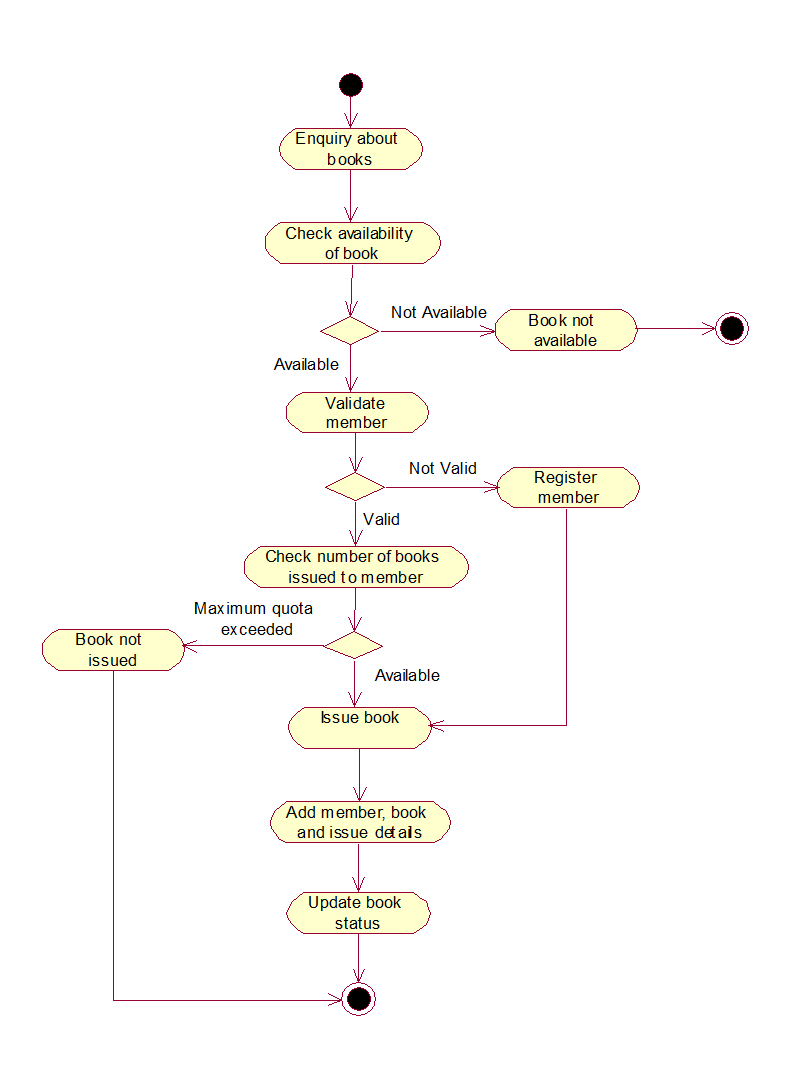
**Case Study:** Draw Statechart diagram and activity diagram for Library Management System.

**Answer:**

## Statechart diagram



## Activity diagram



**Conclusion:** Thus we have studies how to draw statechart diagram and activity diagram