S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 7

Aim: Construct the Activity Diagram for the given problem definition.

Name of Student	**Shrutika Pradeep Bagdi**
Roll No	**CS22130**
Semester/Year	**5th / 3rd**
Academic Session	**2024-2025**
Date of Performance	
Date of Submission	

AIM: Construct the Activity Diagram for the given problem definition.

OBJECTIVE/EXPECTED LEARNING OUTCOME:

- Represent the above information pictorially using activities
- Identify activities representing basic units of work, and represent their flow

HARDWARE AND SOFTWARE REQUIRMENTS:

Hardware Requirement

- Processor : Dual Core
- RAM : 1GB
- Hard Disk Drive : > 80 GB

Software Requirement

- Operating System – Windows

THEORY

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.

Activity diagrams, however, cannot depict the message passing among related objects. As such, it can't be directly translated into code. 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. This could simply be invocation of a mathematical function, alter an object's properties and so on. An activity is represented with a rounded rectangle, as shown in table-01. A label inside the rectangle identifies the corresponding activity.

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 (table-01). Initial node represents the starting point of a flow in an activity diagram. There could be multiple initial nodes, which means that invoking that particular activity diagram would initiate multiple flows.

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, or to other types of components, as we will see below. A flow is often accompanied with a label, called the guard condition, indicating the necessary condition for the transition to happen. The syntax to depict it is [guard condition].

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. However, they can be omitted in obvious cases. The input edge could also have guard conditions. Alternately, a note can be attached to the decision node indicating the condition to be tested.

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. For example, a sales order is closed only when the customer has receive the product, and the sales company has received it's payment.

Note

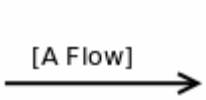
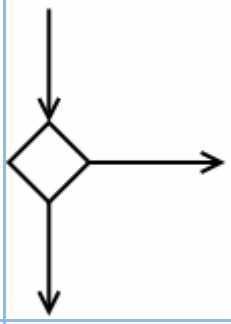
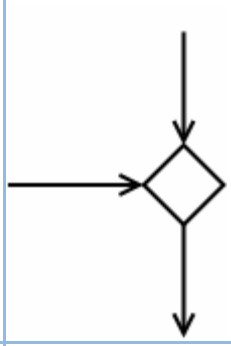
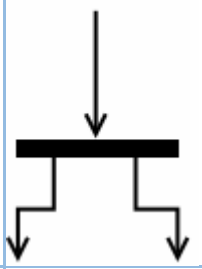
UML allows attaching a note to different components of a diagram to present some textual information. The information could simply be a comment or may be some constraint. A note can be attached to a decision point, for example, to indicate the branching criteria.

Partition

Different components of an activity diagram can be logically grouped into different areas, called partitions or swimlanes. They often correspond to different units of an organization or different actors. The drawing area can be partitioned into multiple compartments using vertical (or

horizontal) parallel lines. Partitions in an activity diagram are not mandatory.

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

Component	Graphical Notation
Activity	
Flow	
Decision	
	
Merge	
Fork	

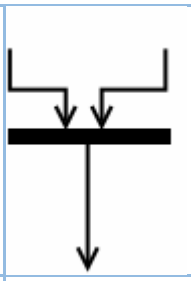
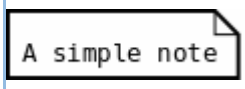
Join	
Note	

Table-01: Typical components used in an activity diagram

Apart from the above stated components, there are few other components as well (representing events, sending of signals, nested activity diagrams), which won't be discussed here. The reader is suggested to go through for further knowledge.

A Simple Example

Figure-04 shows a simple activity diagram with two activities. The figure depicts two stages of a form submission. At first a form is filled up with relevant and correct information. Once it is verified that there is no error in the form, it is then submitted. The two other symbols shown in the figure are the initial node (dark filled circle), and final node (outer hollow circle with inner filled circle). It may be noted that there could be zero or more final node(s) in an activity diagram.

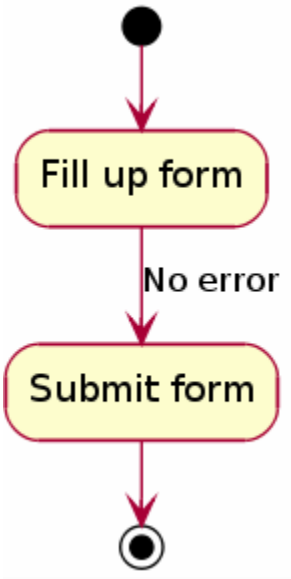


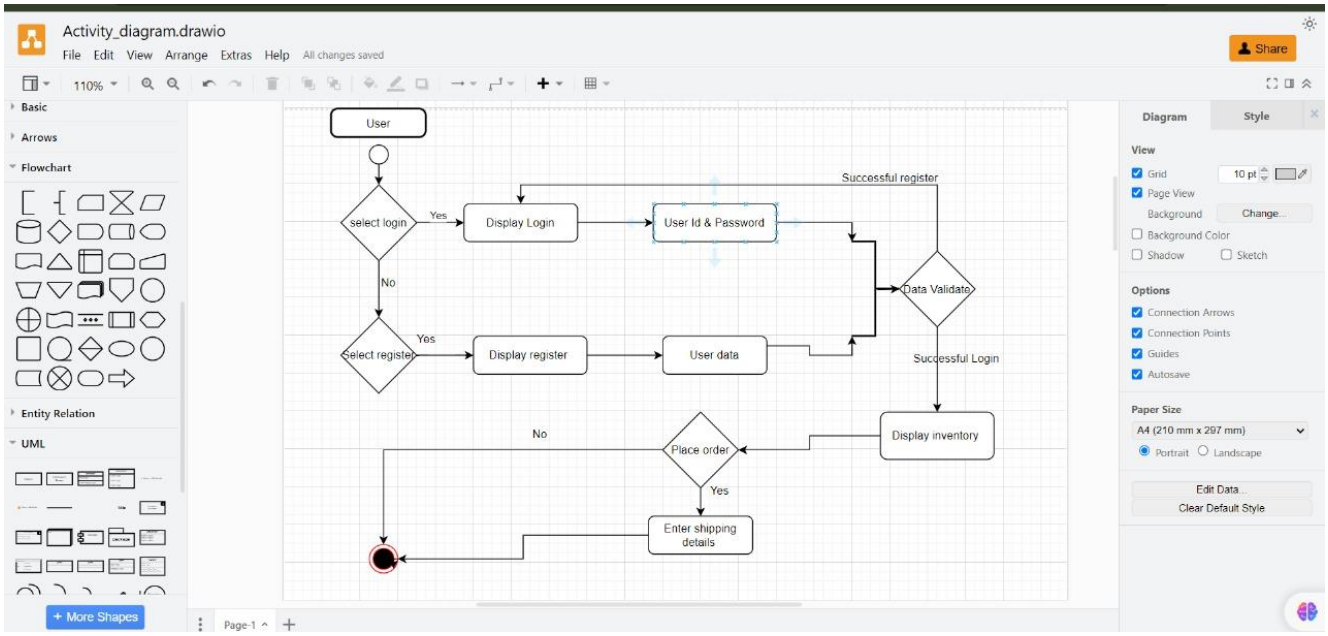
Figure-04: A simple activity diagram..Activity diagram

Guidelines for drawing an Activity Diagram

The following general guidelines could be followed to pictorially represent a complex logic.

- Identify tiny pieces of work being performed by the system
- Identify the next logical activity that should be performed
- Think about all those conditions that should be made, and all those constraints that should be satisfied, before one can move to the next activity
- Put non-trivial guard conditions on the edges to avoid confusion

OBSERVATION (Students should attach screenshot of assigned problem statement):



CONCLUSION:

DISCUSSION QUESTIONS?

1. What is the purpose of an activity diagram?

2. What are the 6 basic elements of an activity diagram?

3. What is the difference between flowchart and activity diagram?

4. What are the symbols used in activity diagram?

5. What are the common properties of activity diagram?

REFERENCES:

- http://vlabs.iitkgp.ernet.in/se/1/
- https://sites.google.com/view/ait-se/Home/practicals
- https://www.javatpoint.com/activity+diagram