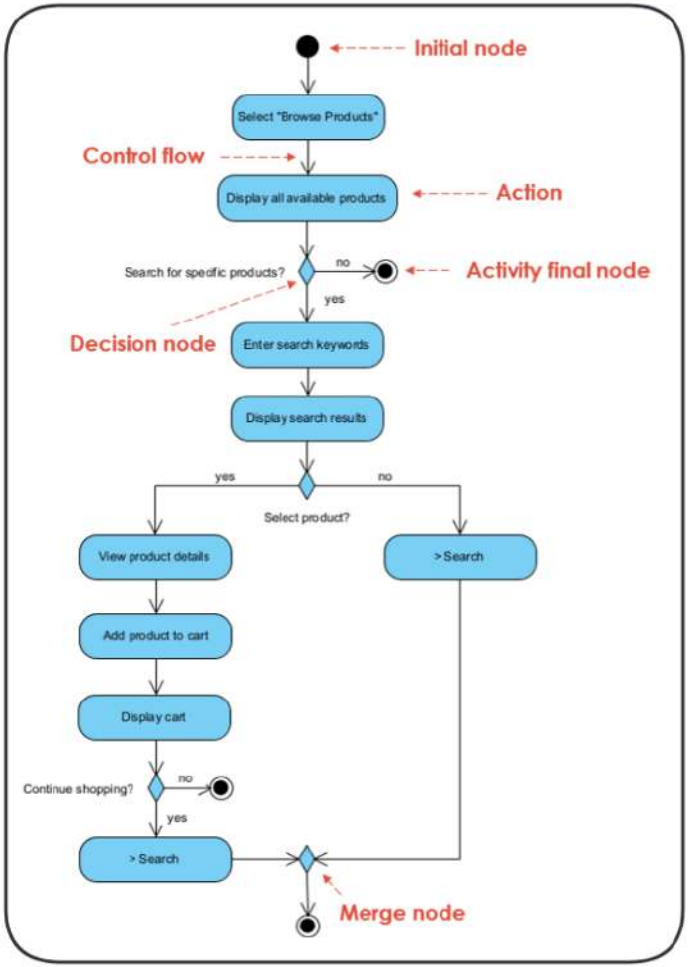
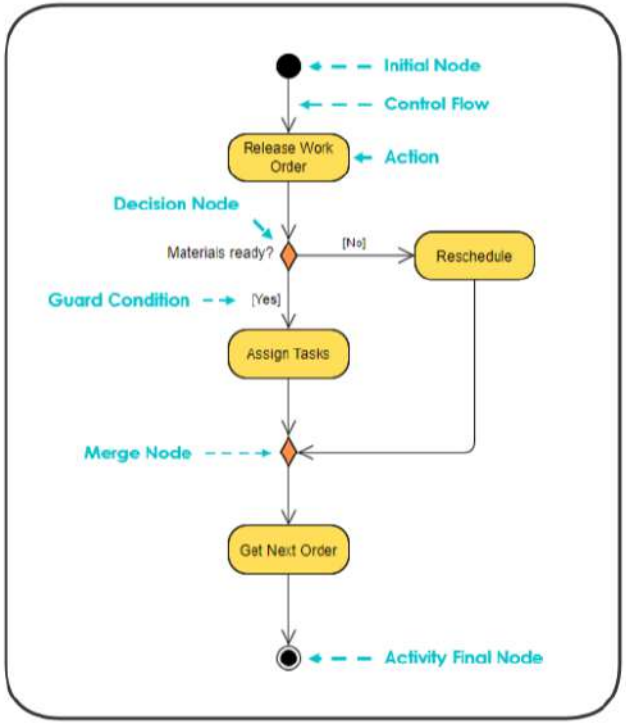
Activity Diagram

- *An activity diagram is a communication diagram that is used to show the dynamic aspects performed by a system.*
- *This diagram is used to represent a series of actions, similar to how they may appear in a* ***flowchart****.*
- *An activity diagram focuses on the condition of flow and the sequence in which it happens.*



Why Activity Diagram

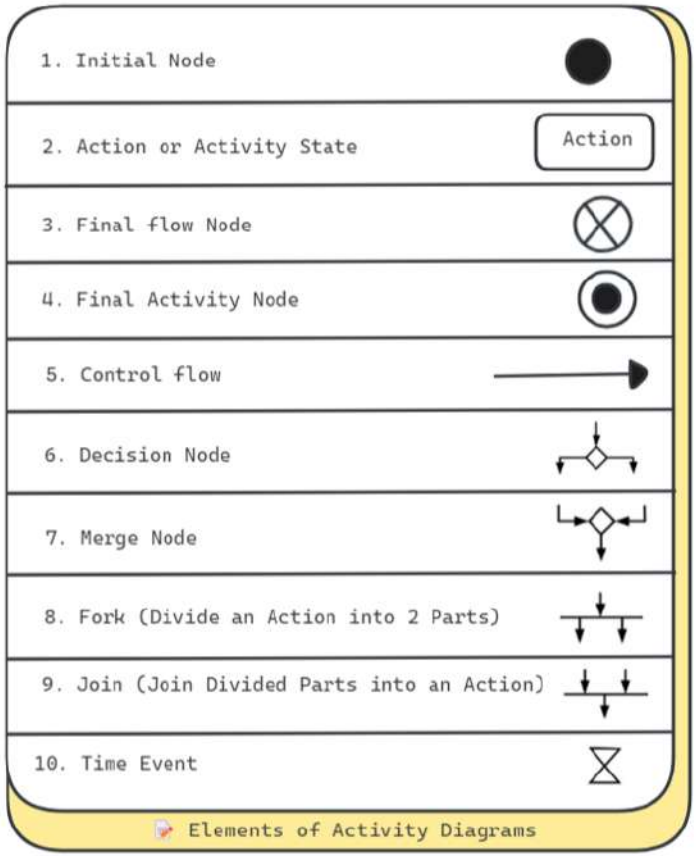
Activity diagrams are another type of diagram we use to show the order of events in a system, just like the other diagrams we talked about. But, activity diagrams are a bit different because ***they focus on how messages move from one activity to the next****. They might look like* ***flowcharts****, but they can show a lot more, like things happening at the same time, one after the other, or even splitting into different paths.*



Elements of Activity Diagram

In activity diagrams, the notations or elements are like visual symbols that help represent different elements and actions in a simple way.

1. *Initial Node*
2. *Action State, Activity State, or Behaviours Box*
3. *Final flow Node*
4. *Final Activity Node*
5. *Control flow, Sequence flow, Connector Edge, or Action Flow*
6. *Decision Node*
7. *Merge Node*
8. *Fork and Join*
9. *Time Event*

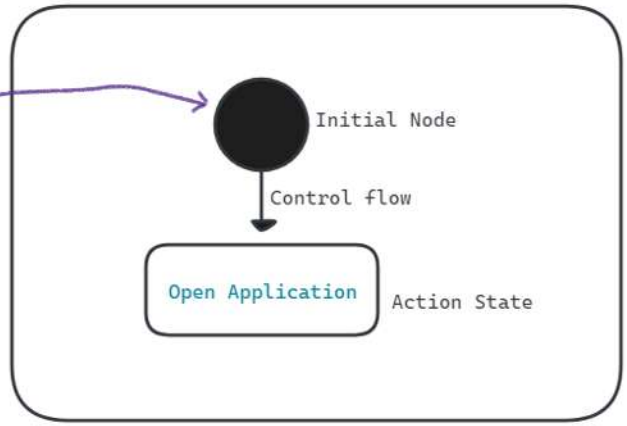


1. Initial Node

This represents the start of the workflow of the activity diagram.
They can be visualized as the node in a tree structure.

For example:

Here the initial state of the system before the application is opened.

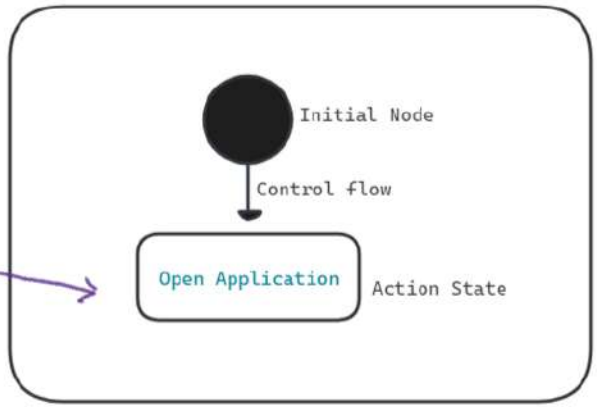


2. Action State, Activity State, or Behaviours Box

These are the main building blocks of an activity diagram and are used to show the activities that a modeled process is made of.

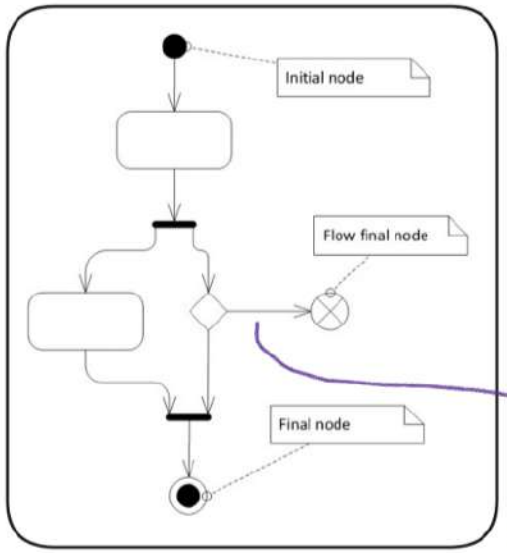
For example:

Consider the previous example of opening an application, opening the application is an activity state in the activity diagram.



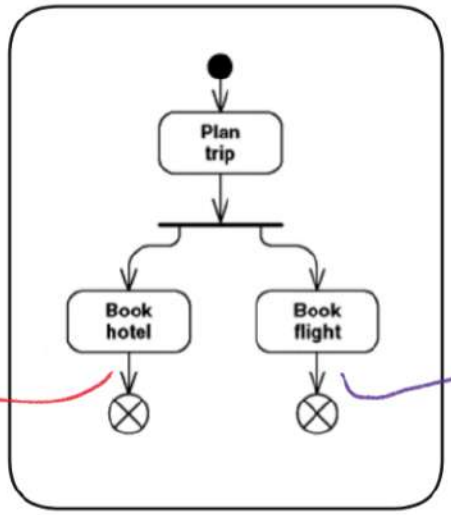
3. Flow Final Node

This represents the end of a single path in the activity diagram.
They can be visualized as a leaf in a tree structure.



END OF A SINGLE PATH

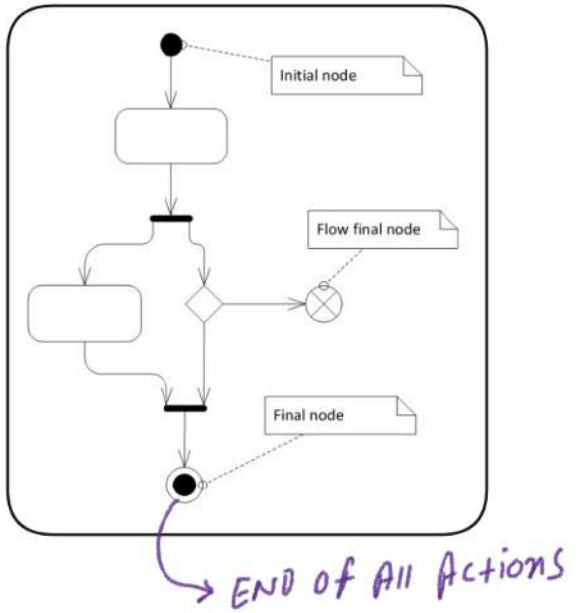
For example: *Booking a trip*



When Booked the Hotel means End of this path

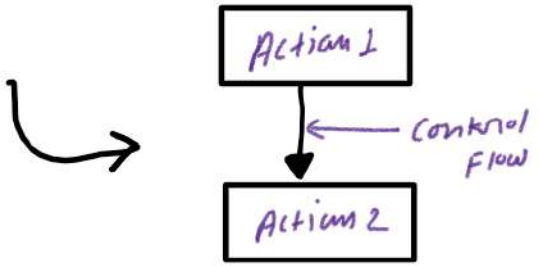
When Booked the flight means End of this path.

4. Final Activity Node *(Final State or End State)*
This represents the end of all the activities in the activity diagram.

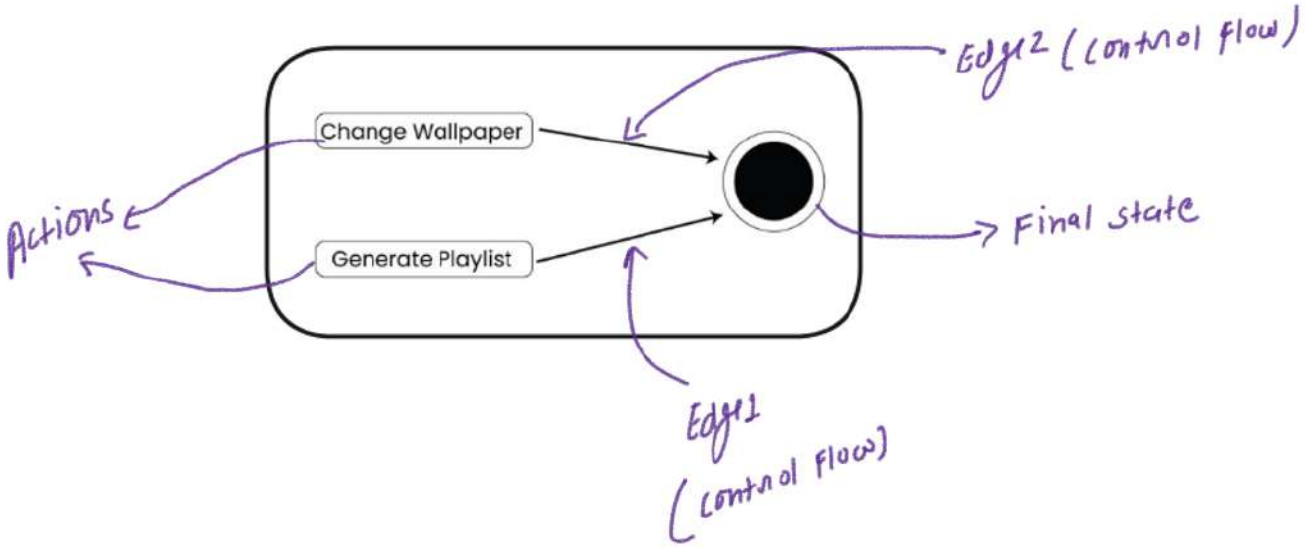


5. Control flow *(Sequence flow, Connector Edge, or Action Flow)*
This shows the directional flow of the diagram. This exists as a connector between one action and another.

- *In other words, Action flows or Control flows are also referred to as paths and edges.*
- *They are used to show the transition from one activity state to another activity state.*

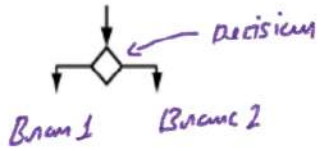


For example:
Here both the states transit into one final state using action flow symbols i.e. ***arrows****.*



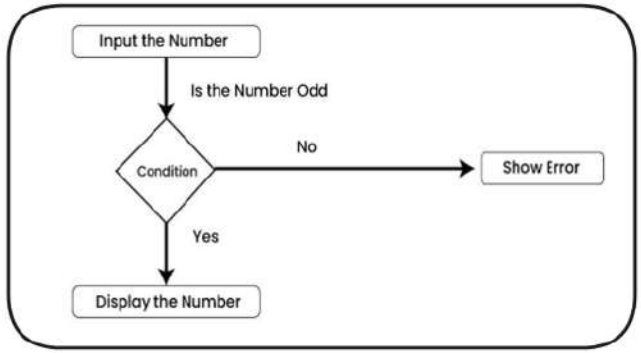
6. Decision Node *(Branching)*
This is used to represent ***multiple options*** *that are possible in a system. They appear as a* ***branch*** *alongside the text describing the condition for the path.*

When we need to make a decision before deciding the flow of control, we use the decision node.

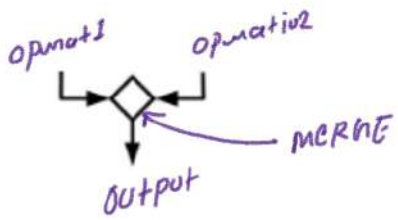


For example:
We apply the conditions on input number to display the result :

- *If number is odd then display the number.*
- *If number if even then display the error.*

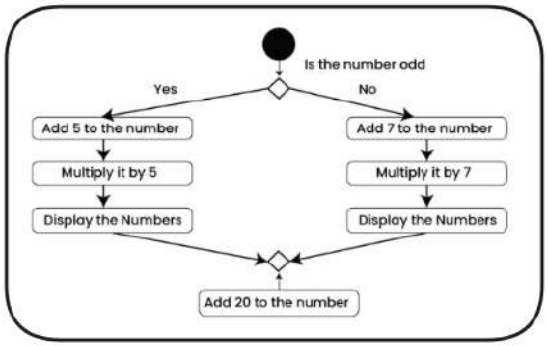


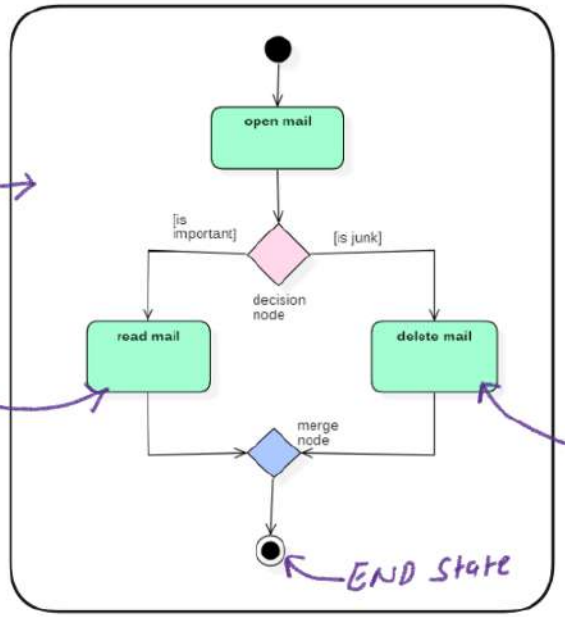
7. Merge Node *(Merge Event)*
This uses the same symbol as a decision. However, this shows that ***multiple options join at this node,*** *but* ***leads to a single output.***



For example:
In the diagram below: we can't have both sides executing concurrently, but they finally merge into one.

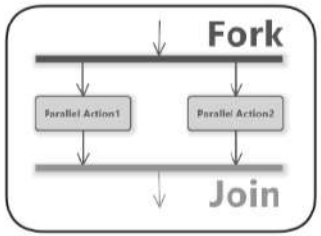
- *A number can't be both odd and even at the same time.*
- *We can not read and delete the same mail at the same time.*





8. Fork and join

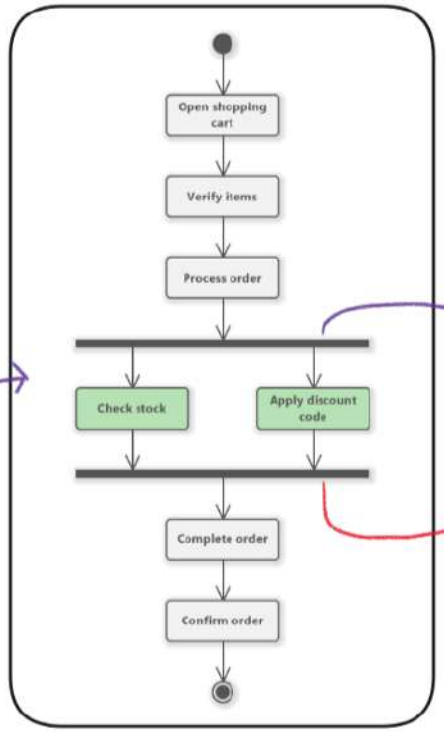
*The **fork** node represents a single activity that is split into two concurrent activities happening alongside each other. On the other hand, the **join** node joins two concurrent activities together to lead to a single activity.*



Fork (Divide an Action into 2 Parts)

Join (Join Divided Parts into an Action)

For example:



FORK — process (order) divided into two parts (Action)

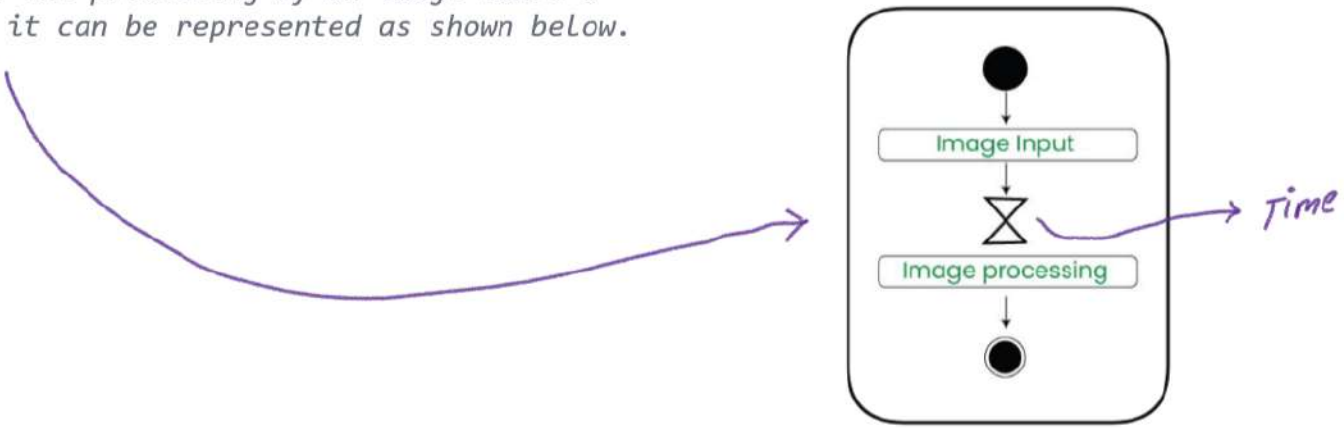
join — join node is joining two parts into single part (Action)

9. Time Event

This refers to an event that stops the flow for a time; an hourglass depicts it. We can have a scenario ***where an event takes some time to completed.***

For example:

Let us assume that the processing of an image takes a lot of time. Then it can be represented as shown below.



How to Draw an Activity Diagram

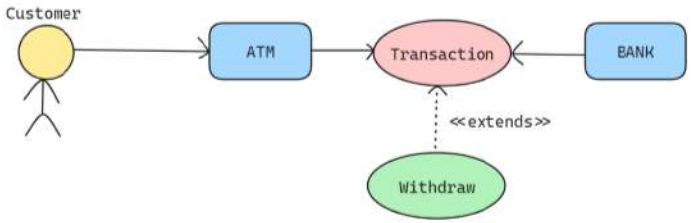
Step 1: Determine the actions of the system
Step 2: Finding the flow of each activity
Step 3: Create the diagram

Step 1: *Determine the actions of the system*

Here, we need to understand each step and how they connect with the rest in the system. Using examples, like stories of what people want to do, can help make this clearer. Let's use the same story as before, where a customer is taking out money from an ATM.

*Figure out **who** is involved and **what** they do, then write down all the **people** and **things** that take part in the whole process. They are:*

1. *Customer*
2. *ATM*
3. *Transaction*
4. *Account*
5. *Cash dispenser*
6. *Bank*
7. *Withdraw*

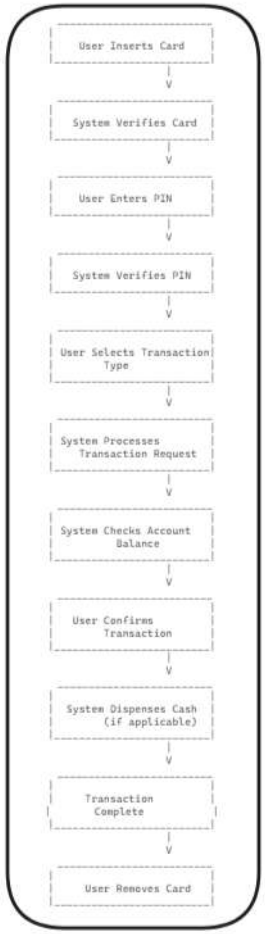


Step 2: *Finding the flow of each activity*

First, we figure out the steps and their order. We also check how some steps happen at the same time as others. We look at what conditions cause certain things to happen. And we note if some steps can only start after the one before is finished.

Rough Flow of Activities in ATM System:

1. *User Inserts Card*
2. *System Verifies Card*
3. *User Enters PIN*
4. *System Verifies PIN*
5. *User Selects Transaction Type*
6. *System Processes Transaction Request*
7. *System Checks Account Balance*
8. *User Confirms Transaction*
9. *System Dispenses Cash (if applicable)*
10. *Transaction Complete*
11. *User Removes Card*



Step 3: Create the diagram

Now, we will use these steps to build a sample activity diagram.

What is SWIMLANES element:

- We use Swimlanes for grouping related activities in one column.
- Swimlanes group related activities into one column or one row.
- Swimlanes can be vertical and horizontal.
- Swimlanes are used to add modularity to the activity diagram.
- It is not mandatory to use swimlanes. They usually give more clarity to the activity diagram.
- It's similar to creating a function in a program.
- It's not mandatory to do so, but, it is a recommended practice.

ATM Activity Diagram with SWIMLANES