**Architecture and Design**

**laboratory work 6**

**EXERCISE 01 – Creating dynamic UML diagrams**

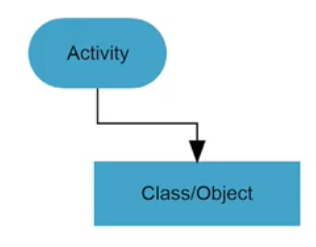
# **Overview**

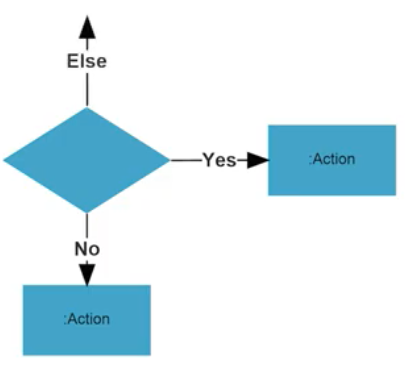
In this project, you will continue with the design of your project. You will include a variety of dynamic UML diagrams that will graphically depict your design.

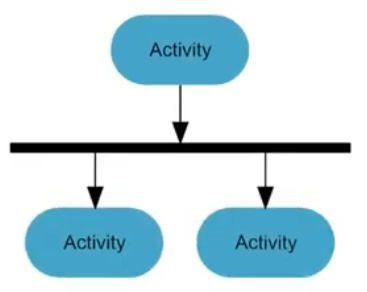
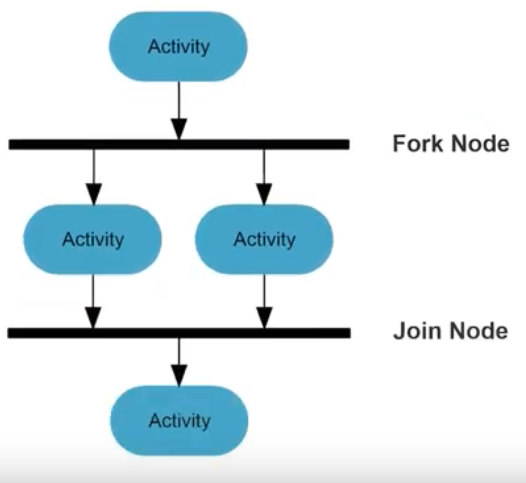
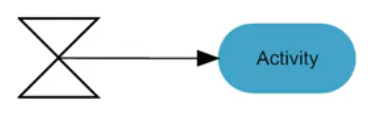
**IMPORTANT: PERFORM ALL TASKS ACCORDING TO YOUR OWN TOPIC AND FOLLOW STANDARDS**

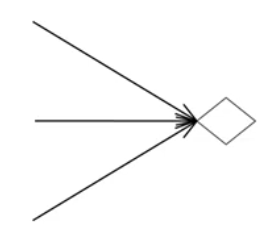
To successfully complete this exersize, you must:

1. Create an Activity Diagram that models what must happen in your system being modeled.
   * 1. Before you begin making an activity diagram, you should first understand its makeup:

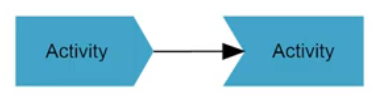
* Every an activity diagram starts with start point , which is located on the top left corner;
* An action state represents a non-interruptible action of objects 
* An action flow illustrates the transitions from one action state to another, it can be drawn with arrowed line ;
* An object flow refers to creation or modification of objects by activity. An object flow from activity to object means that the action creates or influences the object and object flow from object to arrow indicates that the action state uses the object;
* 
* Decision symbol is used when an activity requires a decision prior to moving on to the next activity ;
* Guard symbols can be drawn when decision must be true before moving to the next activity

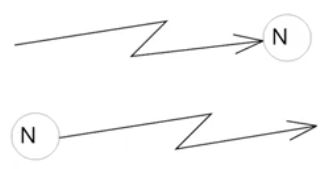


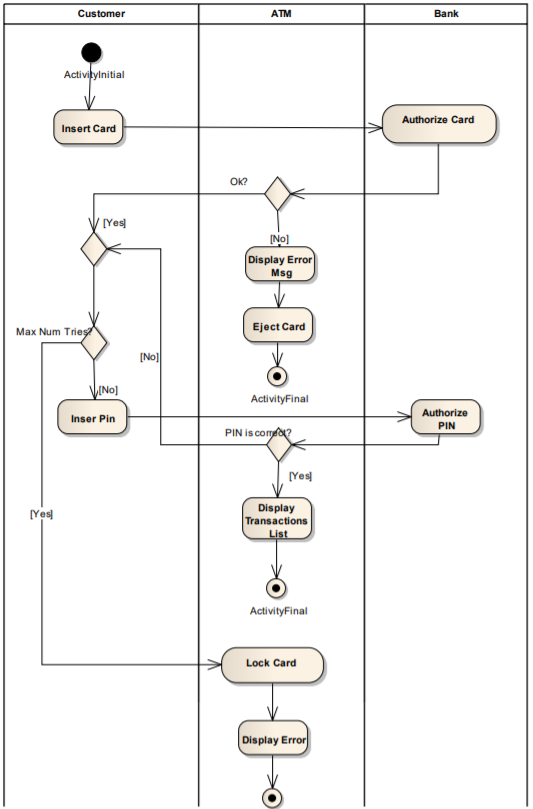
* The Fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as an straight slightly thicker line in an activity diagram; 
* A join node joins multiple concurrent flows back into a single outgoing flow
* The time event refers an events that stops the flow for period of time
* A merge event brings together multiple flows that are not concurrent



* Signals represent how activities can be modified from outside of the system. They usually appear in pairs of sent and received signals, because the state can’t change until a response is received;



* Interrupting edge is an event, that interrupts a flow 
* End flow symbol represents final action state



*Pic1. An activity diagram of ATM*

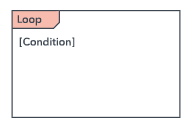
* + 1. Create a Sequence Diagram that shows how objects in a system or classes interact with each other (sequence of events).
* Firstly, think about all actions, that involved in your project.
* Draw your actors as the following figure

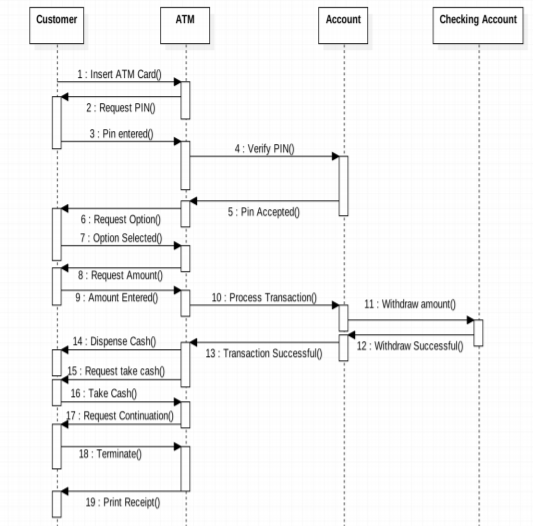


* Draw stick figures to represent each of the actors.
* Then alongside put objects, which can be represented as squares.
* Then add lifelines, which are vertical dash lines, that show the existence of objects or actors over time. Now we can start design interaction between actors and objects.
* This message symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply. 
* The return massage is represented by dashed arrow 
* Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences. To represent alternatives, use the labeled rectangle shape with a dashed line inside.

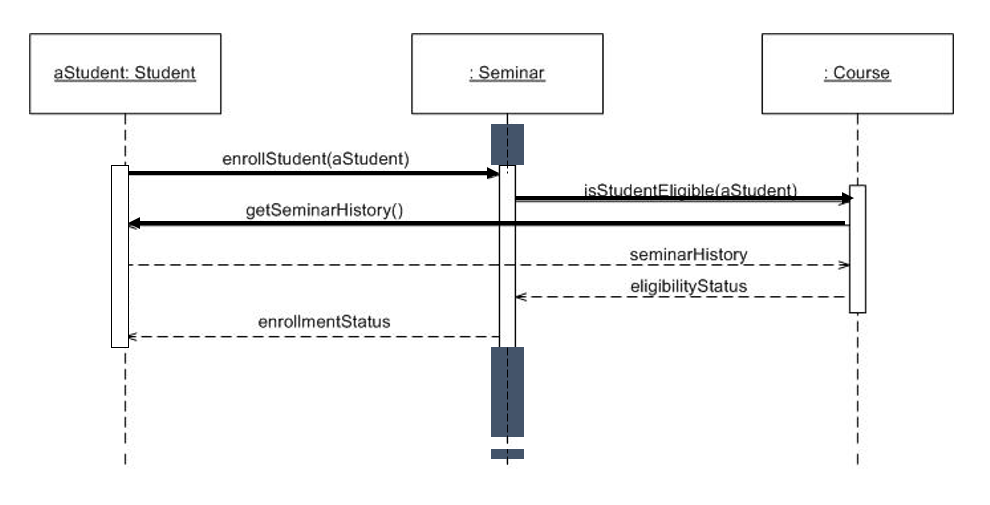


* Option loop symbol Used to model if/then scenarios, i.e., a circumstance that will only occur under certain conditions.





*Pic. 2- A sequence diagram of ATM*



*Pic 3. Another example of sequence diagram*