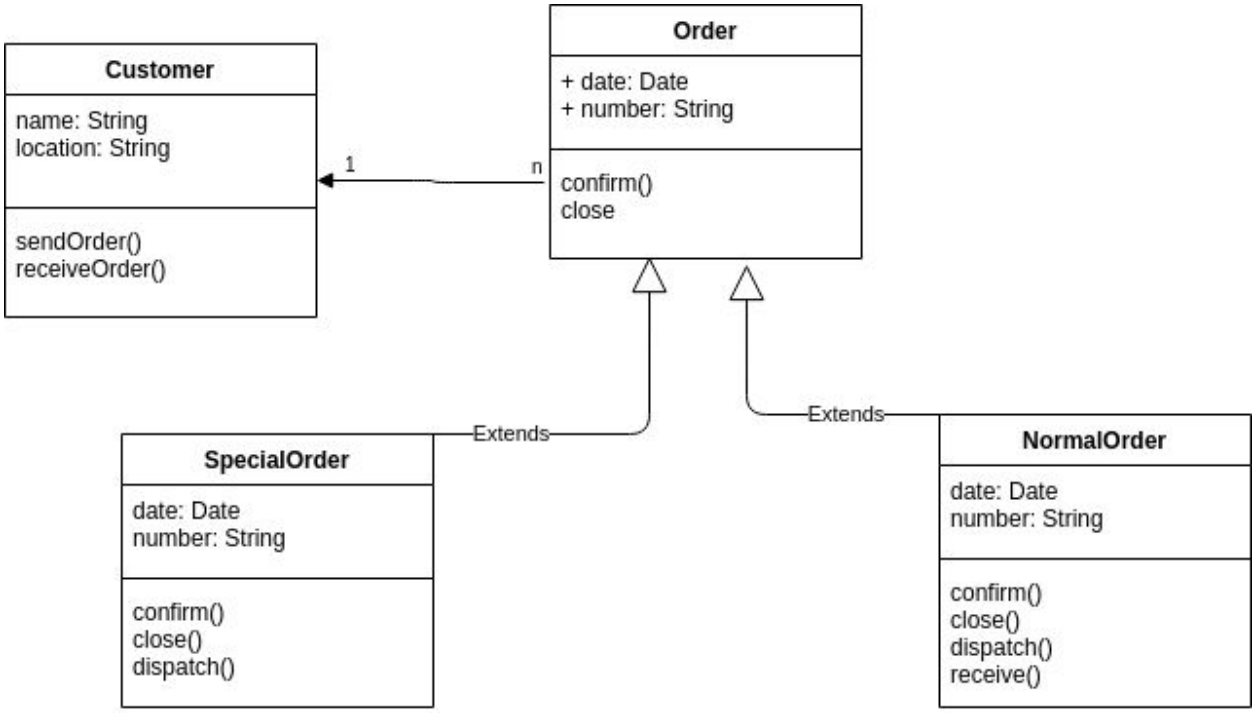
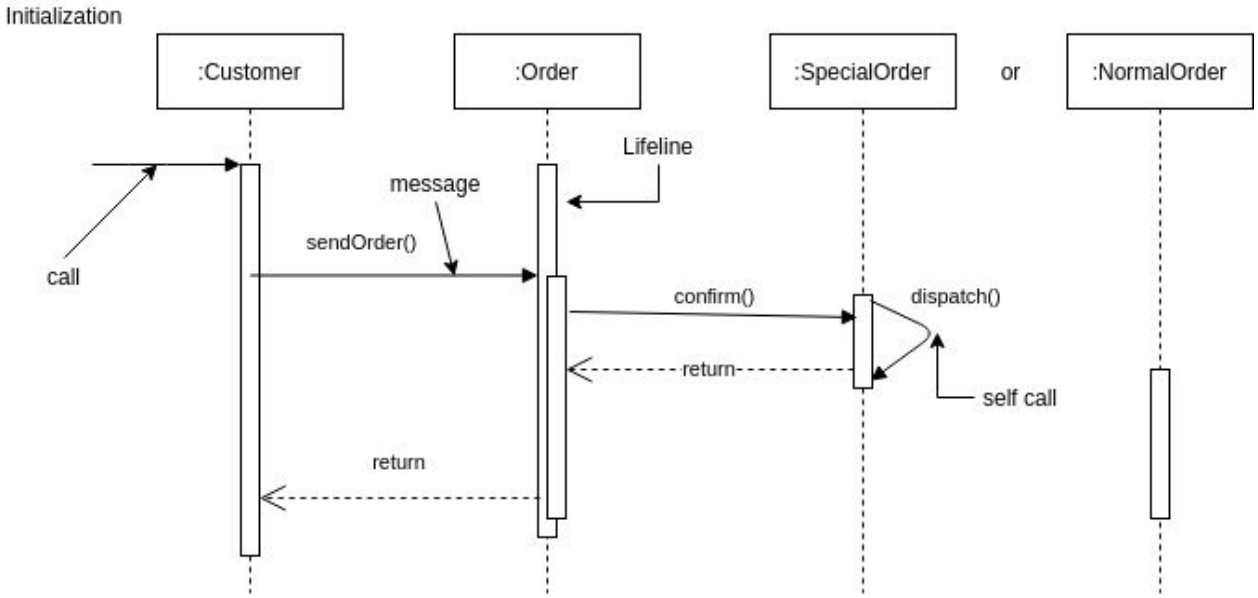
1. Class diagram



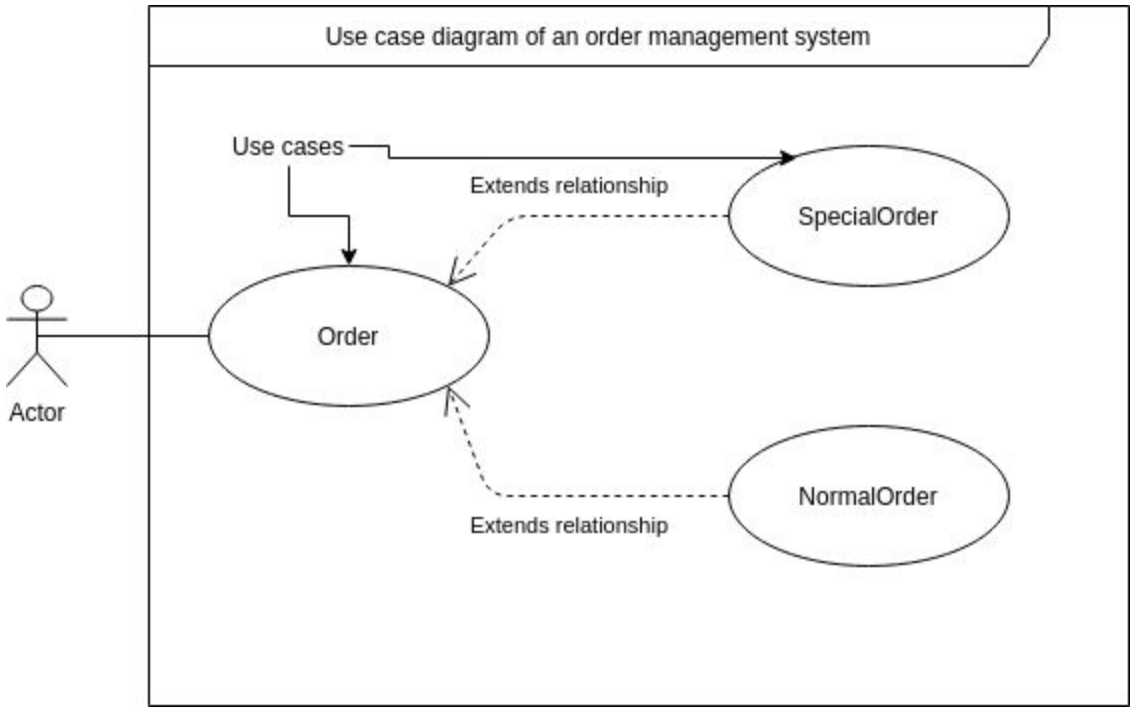
The diagram represents a class diagram and it shows the relation between various classes involved.
The class: Order has entities Date and Name and member functions confirm() and close(). Clearly the class stores the details of the order placed. The order has two other classes connected to it, SpecialOrder and NormalOrder, the relationship is Generalization here, hence the subclasses inherit property parents class as well, apart from that both classes have other member functions in them.
And the class: Customer is connected by many to one connection to Class:Order, which means an order can have many customers, but a customer can have only one order.

2.



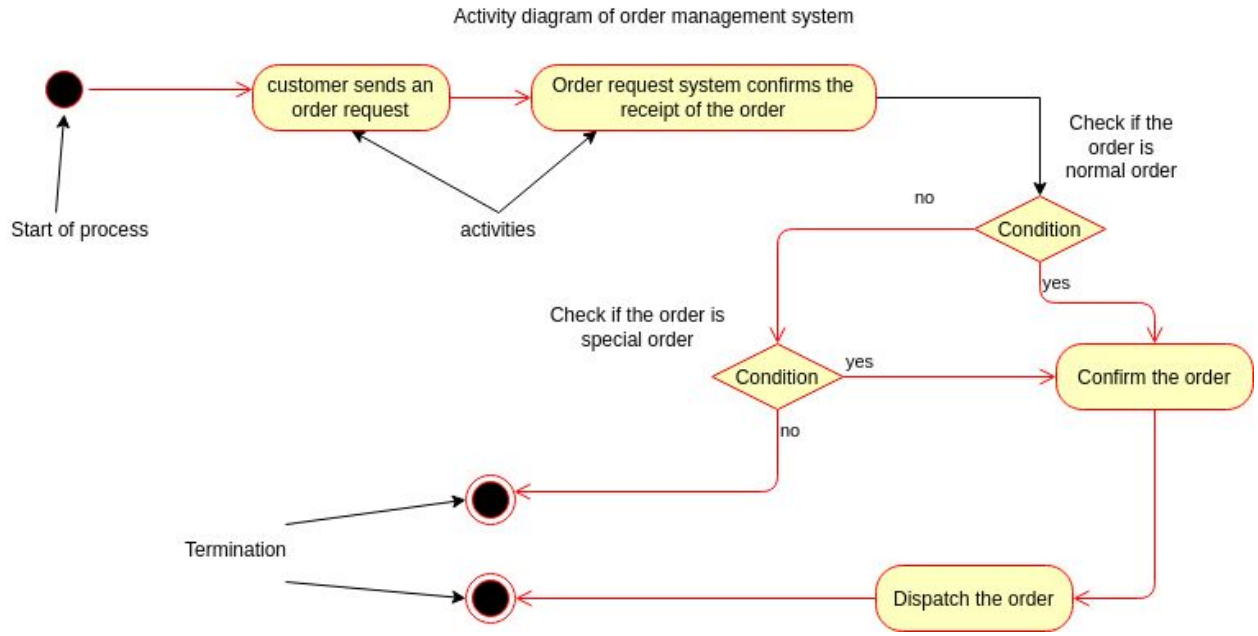
The diagram is a sequence diagram, the diagram starts with input from the customer(which is a call from the customer). The call initiates the process, the sendOrder() message is passed tot :Order. The order is confirmed and a message is passed to :SpecialOrder if the order is a special order. The dispatch() message is passed and is activated as a self call. The same applies for :NormalOrder.

3.



The diagram is a use case diagram. It simply shows the relationship between entities involved in the order management system. Order, SpecialOrder and NormalOrder are use cases here. The actor(here customer) is connected to the order use case, the order use case is related to SpecialOrder and NormalOrder.

4.



The diagram is an activity diagram. The process starts with a call from the customer, the request system is confirmed by the system. Then the system checks if it's normal order. For a normal order it confirms and dispatches the order and terminates. If it's not a normal order, the system checks if it's a special order, if it's a special order, it confirms and dispatches the order and terminates. If it's not a special order it terminates.