
بسم الله الرحمن الرحيم

Software Requirement Engineering

LECTURER: SYED HASNAIN ABBAS BUKHARI

Activity Diagram

- An activity diagram portrays the control flow from a start point to a finish point, showing the various decision paths that exist while the activity is being executed.

Activity Diagram

- We can depict both sequential processing and concurrent processing of activities using an activity diagram.
- They are used in business and process modeling where their primary use is to depict the dynamic aspects of a system.
- An activity diagram is very similar to a flowchart. So let us understand if an activity diagrams or a flowcharts are any different.

Difference Between an Activity Diagram and a Flowchart

- Flowcharts were typically invented earlier than activity diagrams.
- Non programmers use Flow charts to model workflows.
- For example: A manufacturer uses a flow chart to explain and illustrate how a particular product is manufactured.
- We can call a flowchart a primitive version of an activity diagram.
- Business processes where decision making is involved is expressed using a flow chart.

Activity Diagram Notations

- Initial State – The starting state before an activity takes place is depicted using the initial state

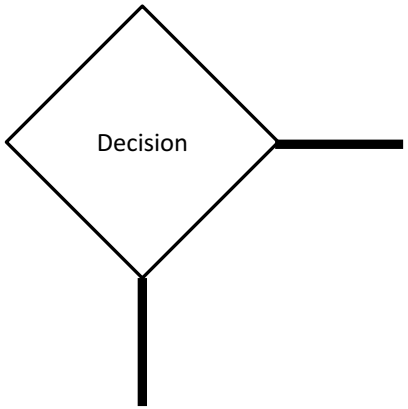
- Action or Activity State – An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners. Basically any action or event that takes place is represented using an activity.

Activity Diagram Notations

- Action Flow or Control flows – 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

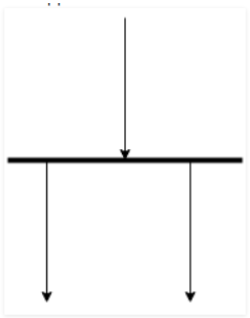
Activity Diagram Notations

- Decision node and Branching – When we need to make a decision before deciding the flow of control, we use the decision node.



Activity Diagram Notations

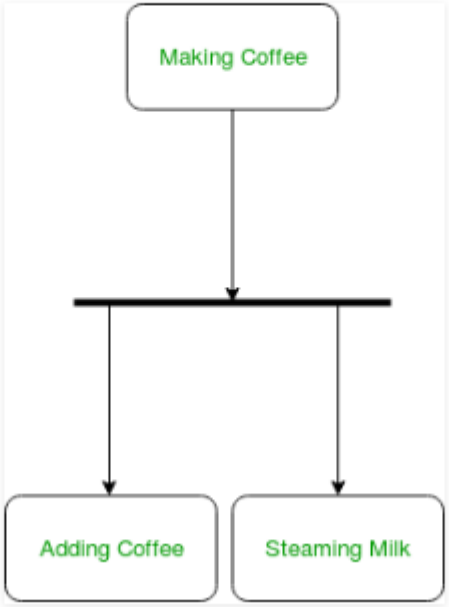
- Fork – Fork nodes are used to support concurrent activities



- We use a fork node when both the activities get executed concurrently i.e. no decision is made before splitting the activity into two parts.
- Both parts need to be executed in case of a fork statement.

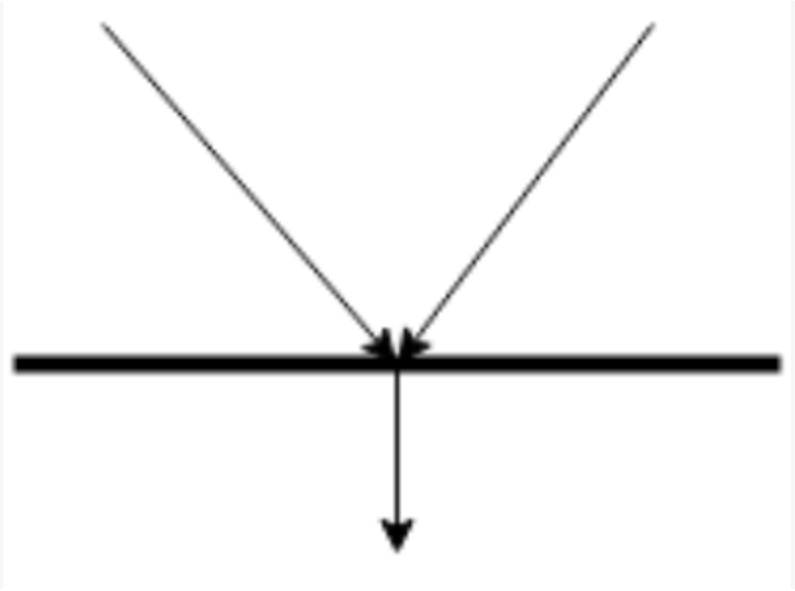
Example

For example: In the example below, the activity of making coffee can be split into two concurrent activities and hence we use the fork notation.



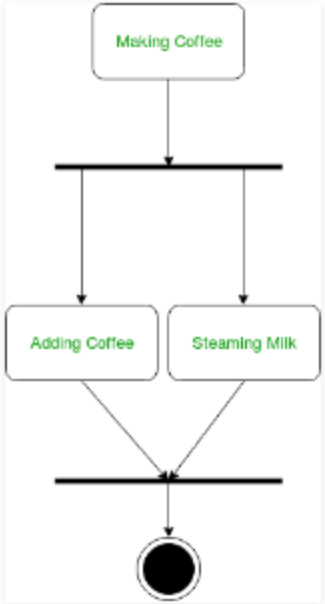
Activity Diagram Notations

- Join – Join nodes are used to support concurrent activities converging into one.
- For join notations we have two or more incoming edges and one outgoing edge.



Activity Diagram Notations

For example – When both activities i.e. steaming the milk and adding coffee get completed, we converge them into one final activity.

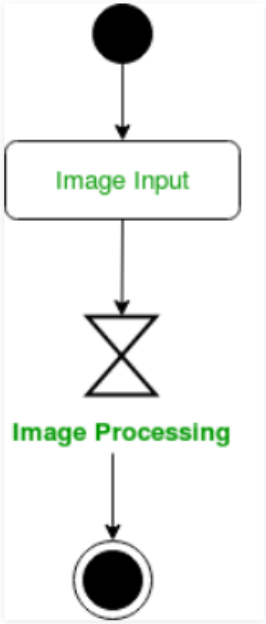


Activity Diagram Notations

- Time Event- We can have a scenario where an event takes some time to complete.
- We use an hourglass to represent a time event.

Example

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



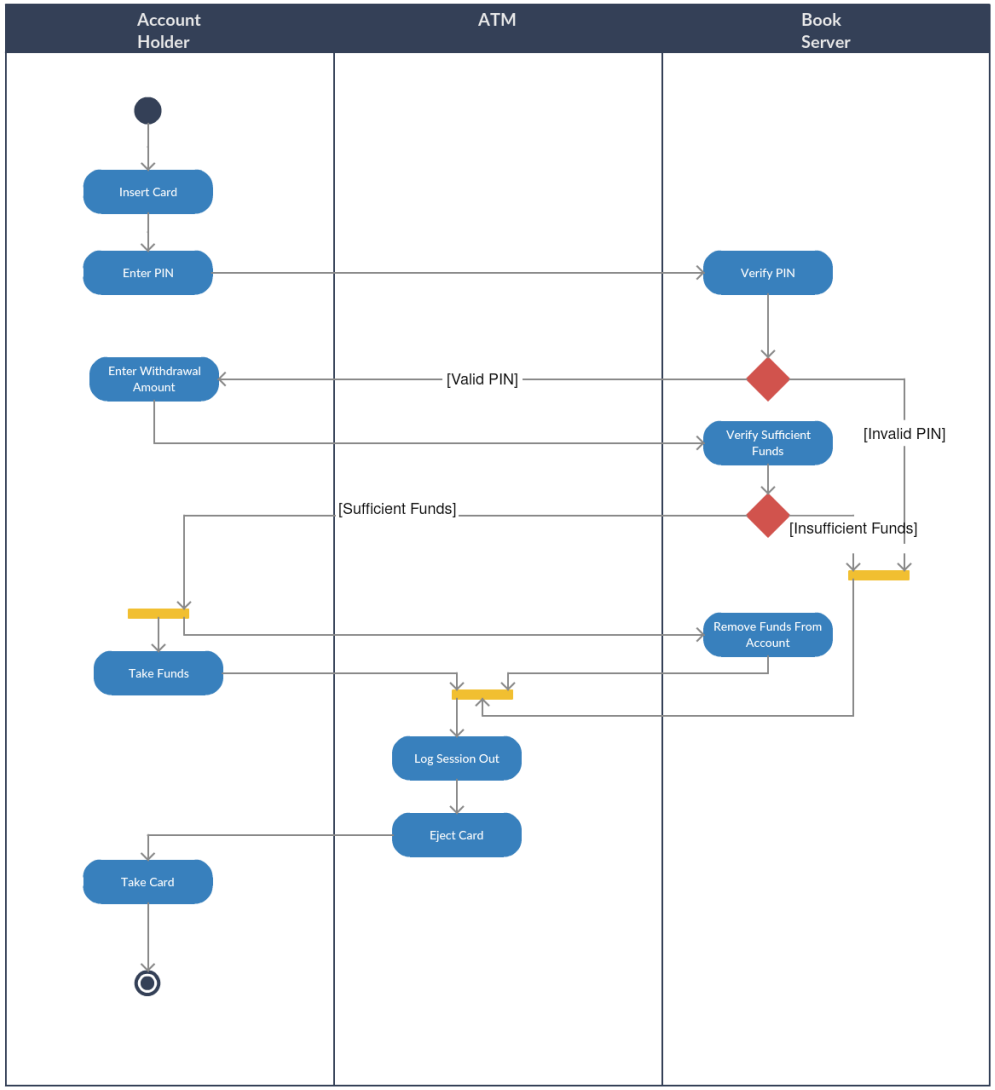
Activity Diagram Notations

- Final State or End State – The state which the system reaches when a particular process or activity ends is known as a Final State or End State.
- We use a filled circle within a circle notation to represent the final state in a state machine diagram.
- A system or a process can have multiple final states.

Activity Diagram Notations

Flow Final Node	
 *Flow final node.*	**Flow final node** is a control final node that terminates a flow. The notation for flow final node is small circle with X inside.

ATM SYSTEM for ABC BANK



How to Draw an activity diagram

- Identify the initial state and the final states.
- Identify the intermediate activities needed to reach the final state from the initial state.
- Identify the conditions or constraints which cause the system to change control flow.
- Draw the diagram with appropriate notations.

Uses of an Activity Diagram

- Dynamic modelling of the system or a process.
- Illustrate the various steps involved in a UML use case.
- Model software elements like methods, operations and functions.
- We can use Activity diagrams to depict concurrent activities easily.
- Show the constraints, conditions and logic behind algorithms.

Sequence Diagram

Sequence Diagram

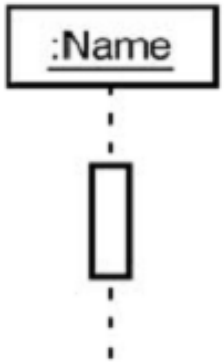
- Sequence diagram: an "interaction diagram" that models a single scenario executing in the system
- Describe the flow of messages, events & actions between objects
- Show time sequences that are not easily depicted in other diagrams
- Typically used during analysis and design to document and understand the logical flow of your system

Sequence Diagram

- The sequence diagram consists of objects represented in the usual way (as named rectangles with the name underlined), messages represented as solid-line arrows, and time represented as a vertical progression

Objects

- The objects are laid out near the top of the diagram from left to right..
- Extending downward from each object is a dashed line called the object's lifeline.
- Along the lifeline is a narrow rectangle called an activation. The activation represents an execution of an operation the object carries out. The length of the rectangle signifies the activation's duration.

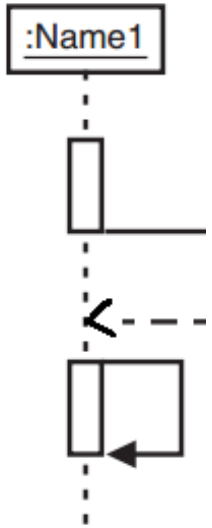


Messages

- A message that goes from one object to another goes from one object's lifeline to the other object's lifeline.

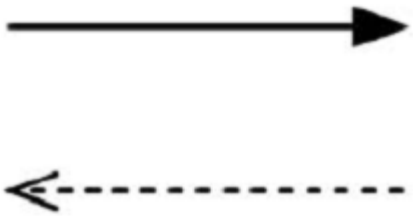
- An object can also send a message to itself—that is, from its lifeline back to its own lifeline.

- Type of Messages
  - Synchronous
  - Asynchronous



Synchronous Message

The sender waits for the receiver (that is, "synchs up" with the receiver), this message is also referred to as synchronous.



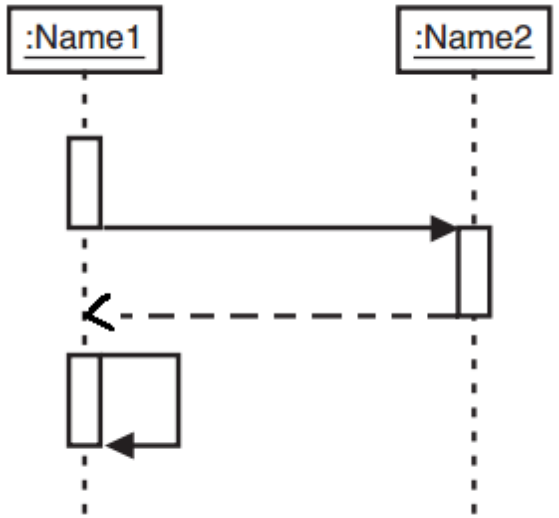
The UML symbol for a call and for a return

Asynchronous Message

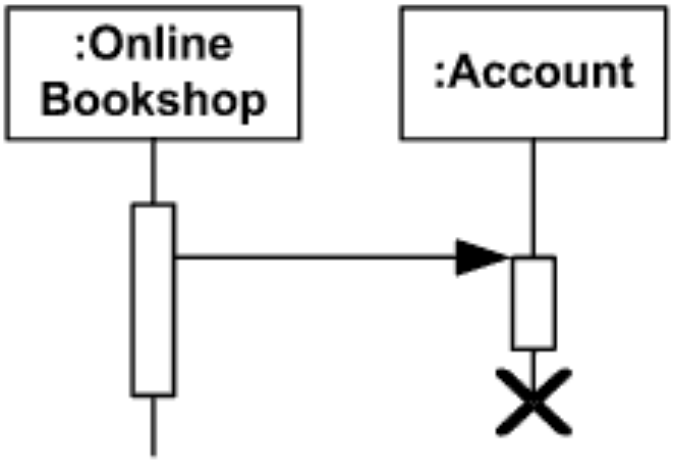
- With this one, the sender transfers control to the receiver and doesn't wait for the operation to complete

Time

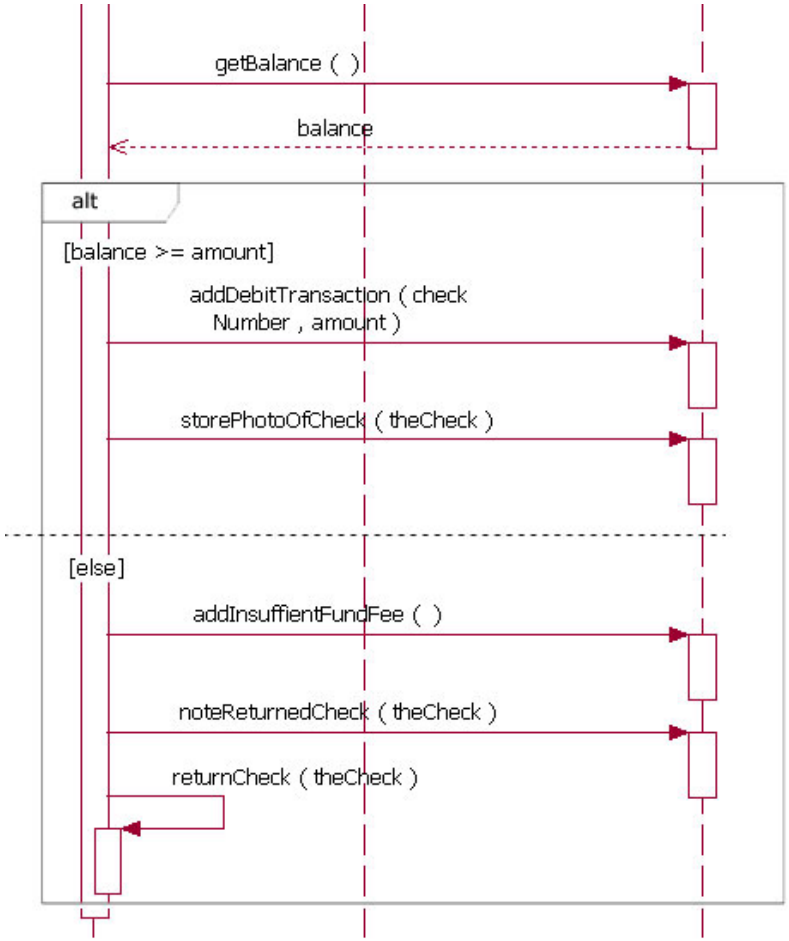
- The diagram represents time in the vertical direction: Time starts at the top and progresses toward the bottom.
- Thus, the sequence diagram is two-dimensional. The left-to-right dimension is the layout of the objects, and the top-to-bottom dimension shows the passage of time.



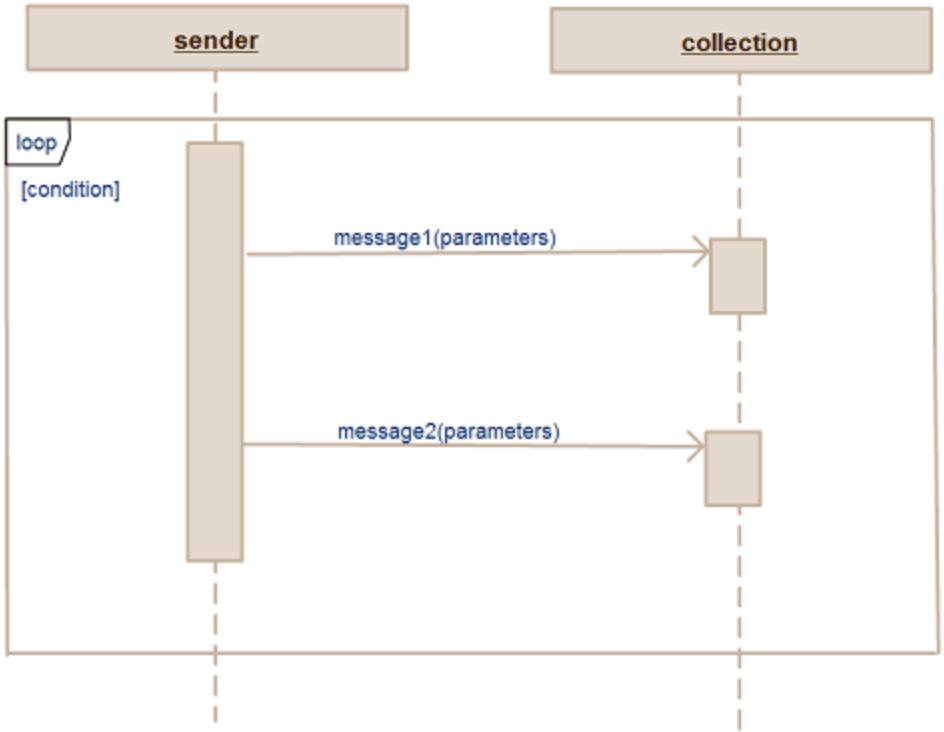
Destroy/Destruction



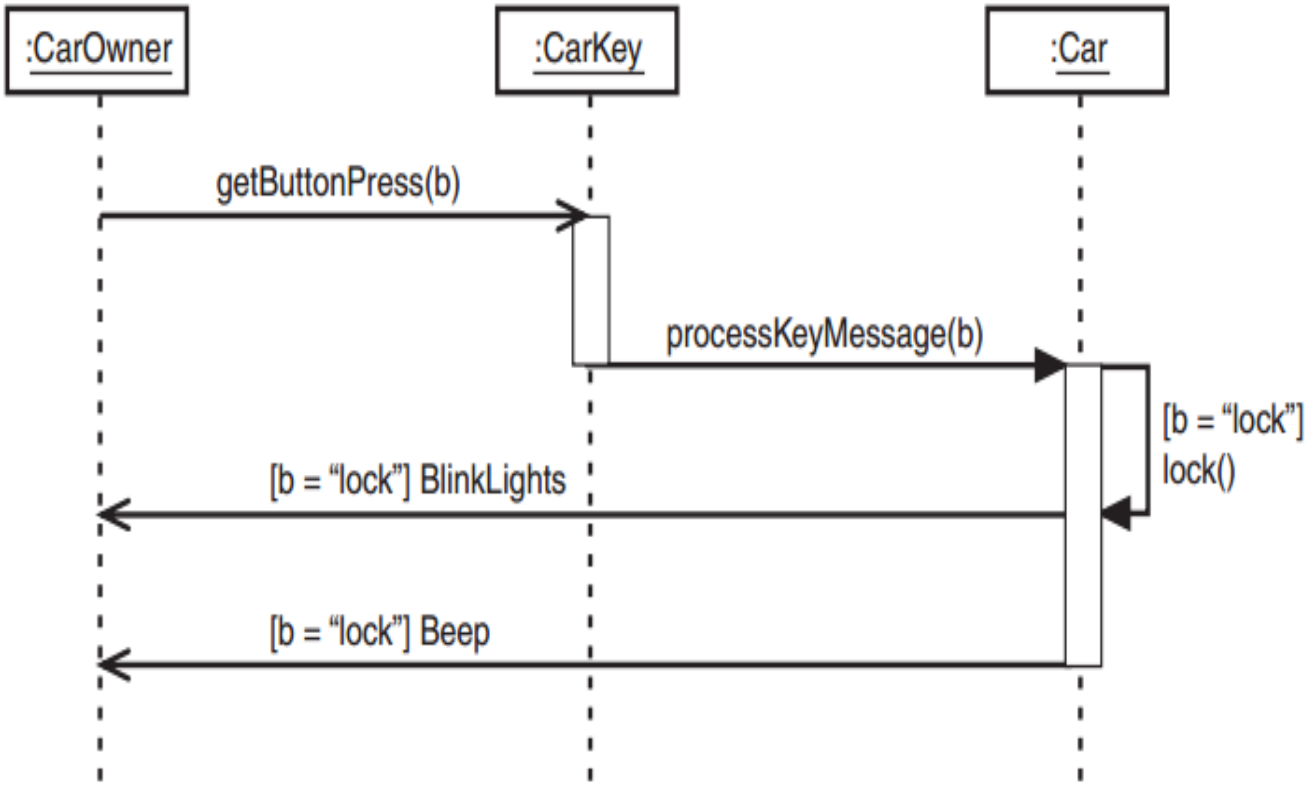
Alt/Else


getBalance ( )
balance
alt
[balance >= amount]
addDebitTransaction (check Number , amount)
storePhotoOfCheck (theCheck)
[else]
addInsuffientFundFee ( )
noteReturnedCheck (theCheck)
returnCheck (theCheck)

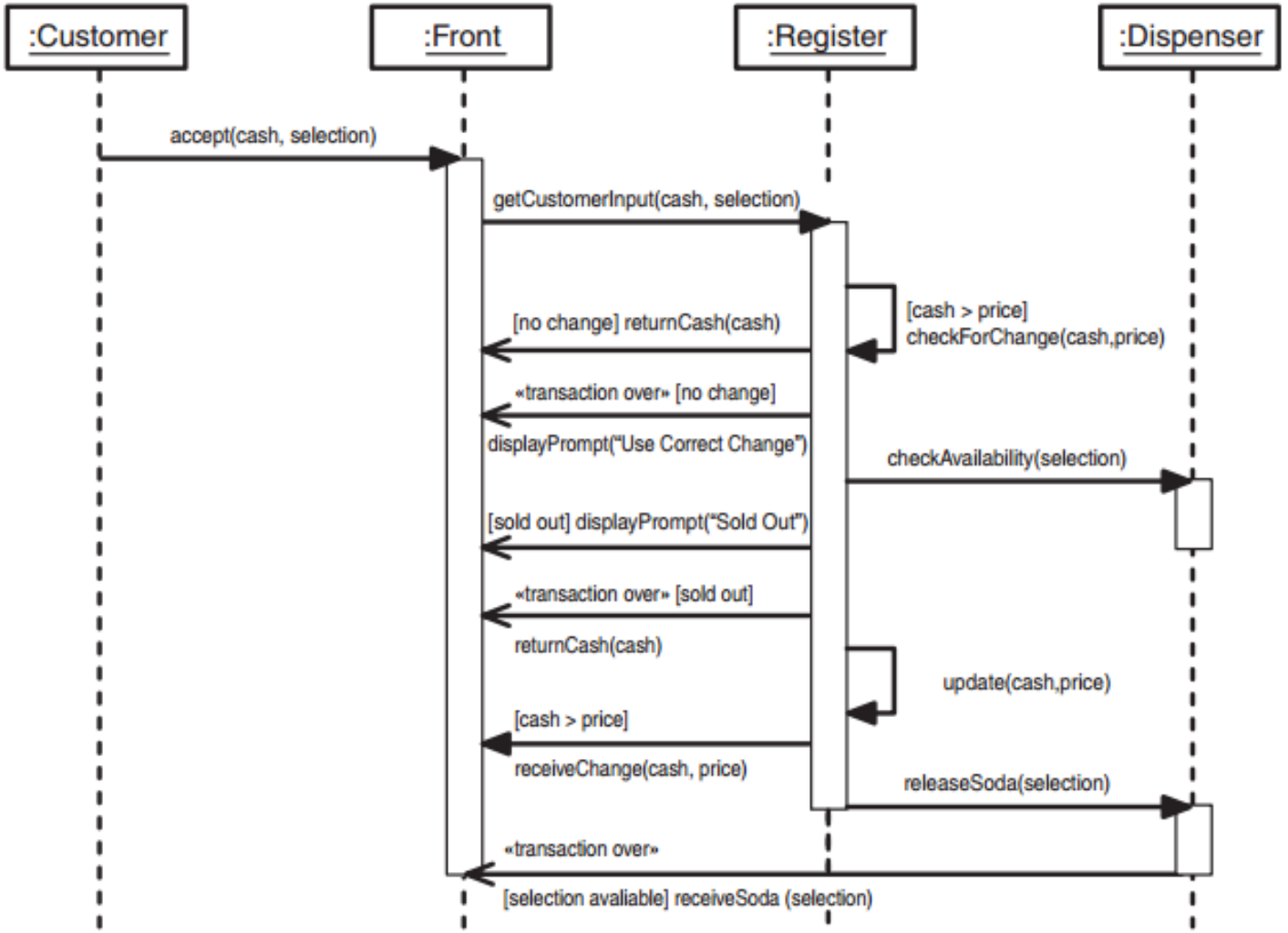
Loop

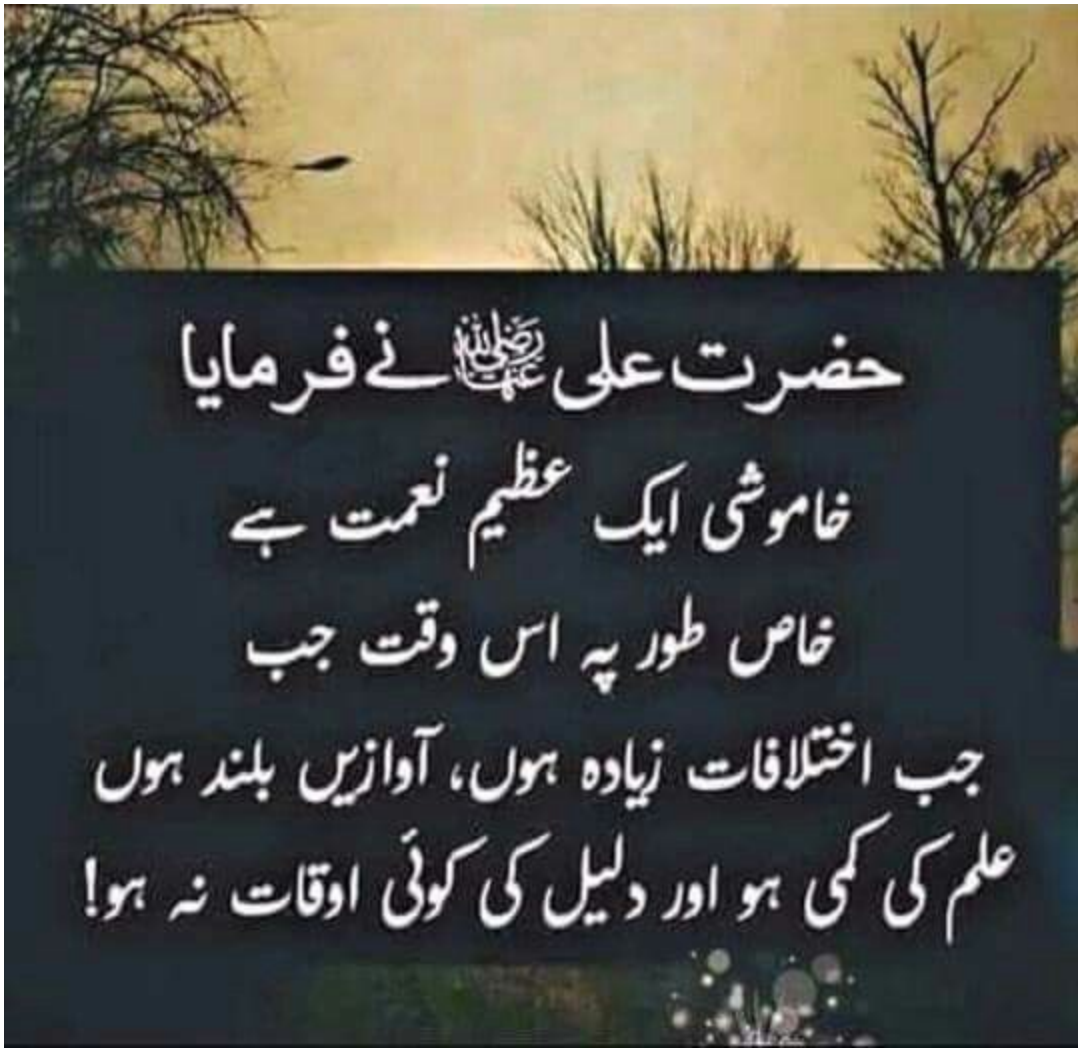

sender
collection
loop
[condition]
message1(parameters)
message2(parameters)

Example


:CarOwner
:CarKey
:Car
getButtonPress(b)
processKeyMessage(b)
[b = "lock"]
lock()
[b = "lock"] BlinkLights
[b = "lock"] Beep

A generic sequence diagram of buying drink




حضرت علی رضی اللہ عنہ نے فرمایا
خاموشی ایک عظیم نعمت ہے
خاص طور پہ اس وقت جب
جب اختلافات زیادہ ہوں، آوازیں بلند ہوں
علم کی کمی ہو اور دلیل کی کوئی اوقات نہ ہو!