

Scenario-Based Modeling

Activity Diagram : Control Flow

Analysis Modeling Approaches

- Structured analysis
 - Considers data and the processes that transform the data as separate entities
 - Data is modeled in terms of only attributes and relationships (but no operations)
 - Processes are modeled to show the 1) input data, 2) the transformation that occurs on that data, and 3) the resulting output data
- Object-oriented analysis
 - Focuses on the definition of classes and the manner in which they collaborate with one another to fulfill customer requirements

Elements of the Analysis Model

Object-oriented Analysis

Scenario-based modeling

Use case text
Use case diagrams
Activity diagrams
Swim lane diagrams

Class-based modeling

Class diagrams
Analysis packages
CRC models
Collaboration diagrams

static

Procedural/Structured Analysis

Flow-oriented modeling

Data structure diagrams
Data flow diagrams
Control-flow diagrams
Processing narratives

Behavioral modeling

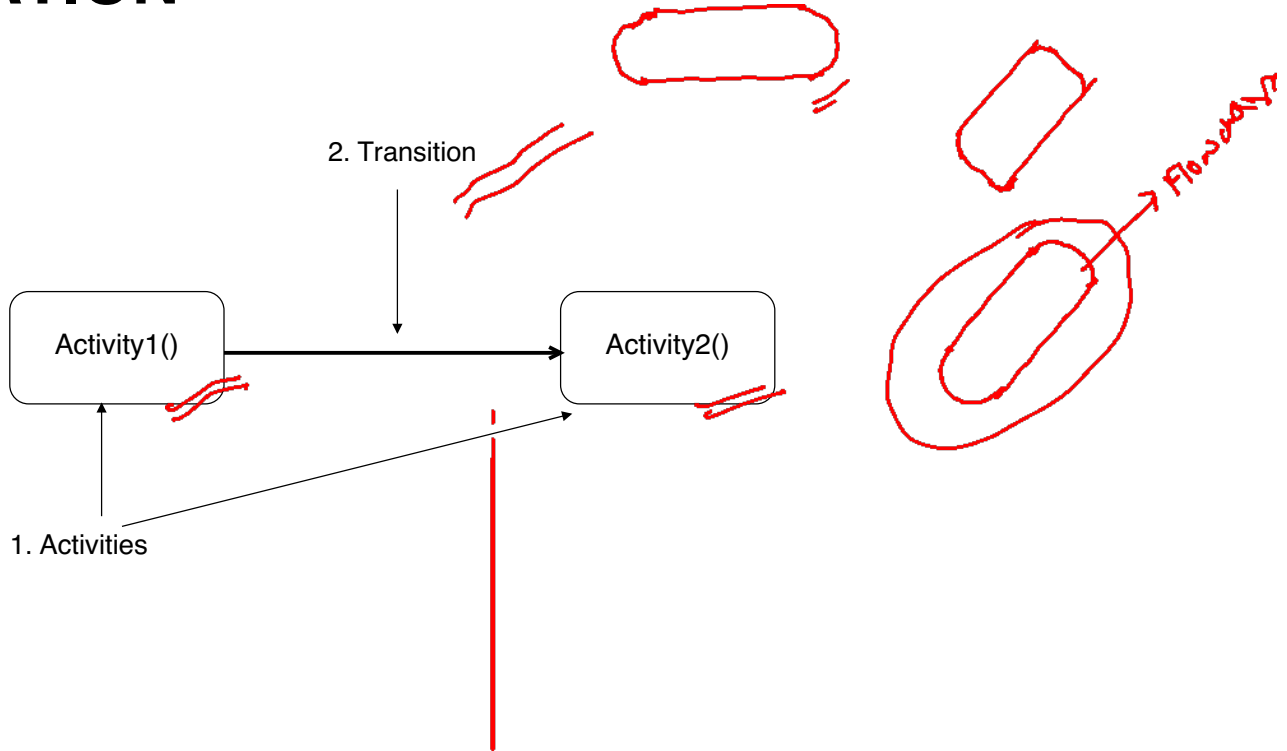
State diagrams
Sequence diagrams

Developing an Activity Diagram

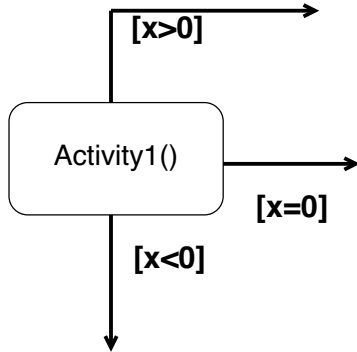
Activities

- Describes how activities are coordinated.
- Is particularly useful when you know that an operation has to achieve a number of different things, and you want to model what the essential dependencies between them are, before you decide in what order to do them.
- Record the dependencies between activities, such as which things can happen in parallel and what must be finished before something else can start.
- Represents the workflow of the process.

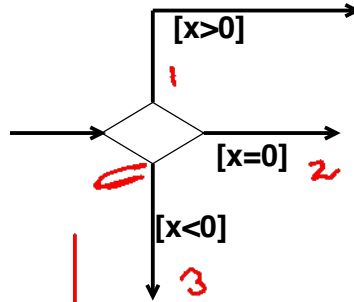
NOTATION



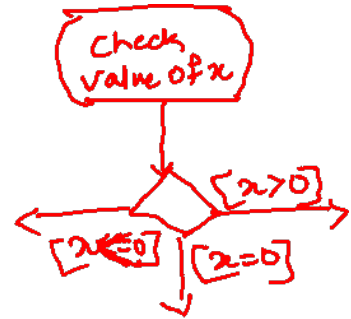
NOTATION - 2



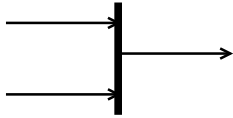
3. Decision Diamond



[]
Guard cond'n

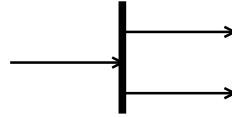


NOTATION - 3



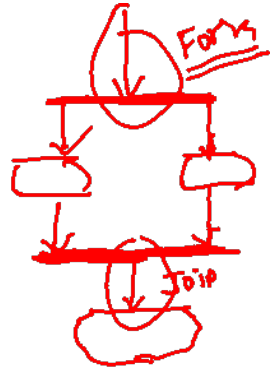
4.1 Synchron. Bar (Join)

2



4.2 Splitting Bar (Fork)

1



Notation - 3



5. Start & Stop Markers

Notation - 4

- Simple Activity diagram

- Swimlane diagram ✓

↓
Actors

Developers

Testers

Markers

Swimlane

Swimlane

Swimlane

Application/Department/Group/Role Boundaries

Activity Diagrams

- Activity diagrams commonly contain
 - Activity states and action states
 - Transitions
 - Objects

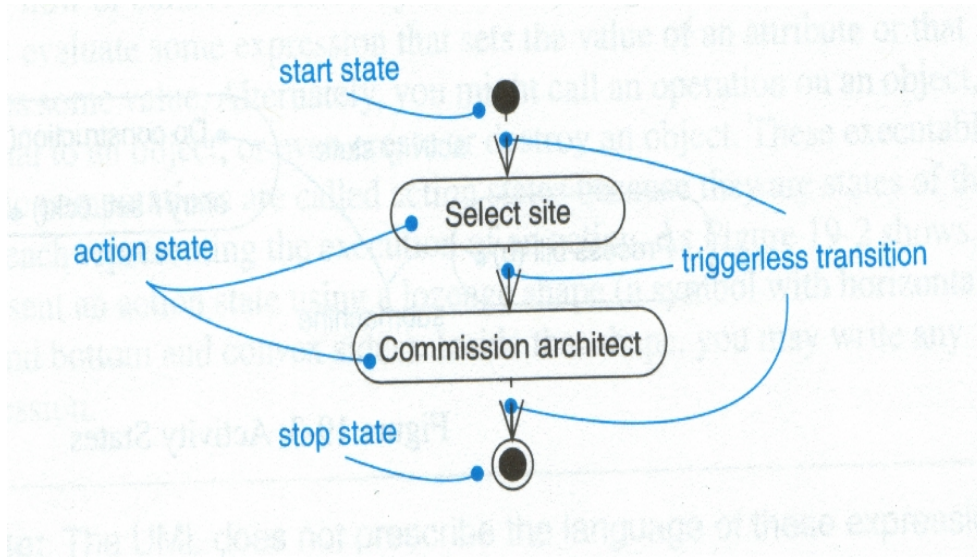
Action States and Activity States

- Action states are atomic and cannot be decomposed
 - Work of the action state is not interrupted
- Activity states can be further decomposed
 - Their activity being represented by other activity diagrams
 - They may be interrupted

Transitions

- When the action or activity of a state completes, flow of control passes immediately to the next action or activity state
- A flow of control has to start and end someplace
 - initial state -- a solid ball — Start
 - stop state -- a solid ball inside a circle Stop

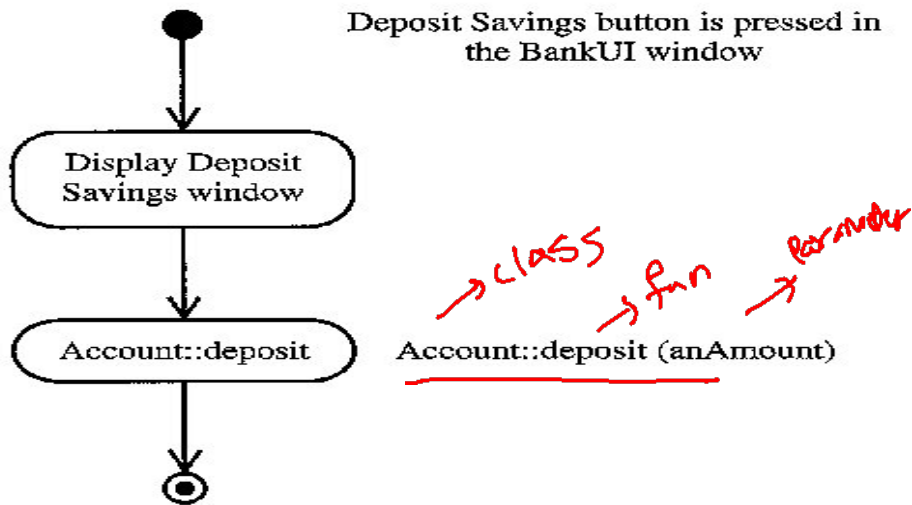
Transitions



Activity Diagram: Example (1)

FIGURE 12-24

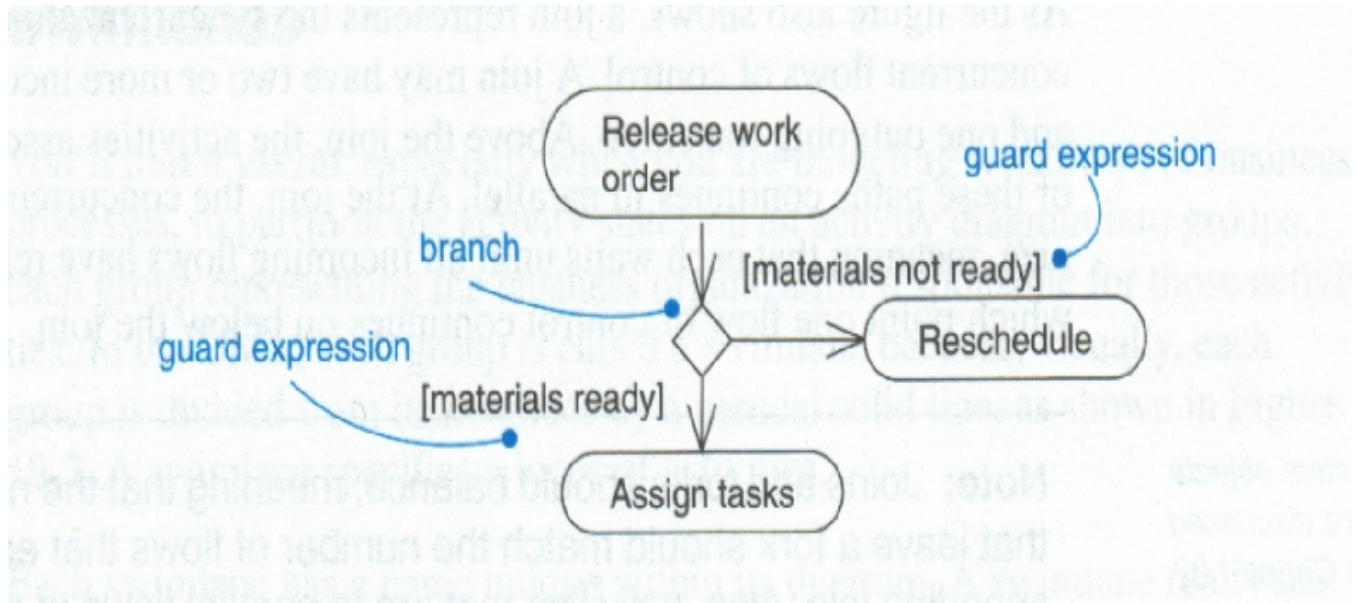
Activity diagram for processing a deposit to a savings account.



Branching

- A branch specifies alternate paths taken based on some Boolean expression
- A branch may have one incoming transition and two or more outgoing ones

Branching



Activity Diagram: Example

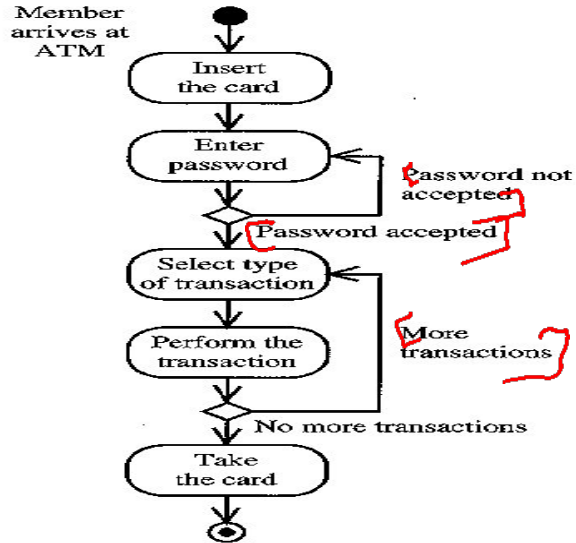


FIGURE 6-8

Activities involved in an ATM transaction.

Forking and Joining

- Use a synchronization bar to specify the forking and joining of parallel flows of control
- A synchronization bar is rendered as a thick horizontal or vertical line

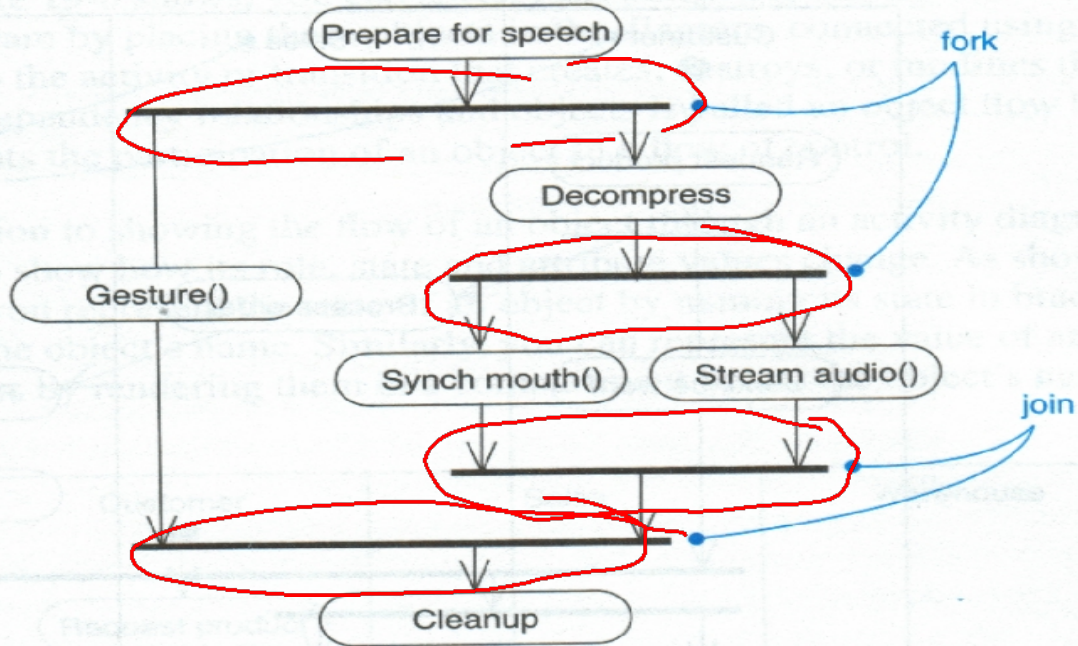
Fork

- A fork may have one incoming transitions and two or more outgoing transitions
 - each transition represents an independent flow of control
 - conceptually, the activities of each of outgoing transitions are concurrent
 - either truly concurrent (multiple nodes)
 - or sequential yet interleaved (one node)

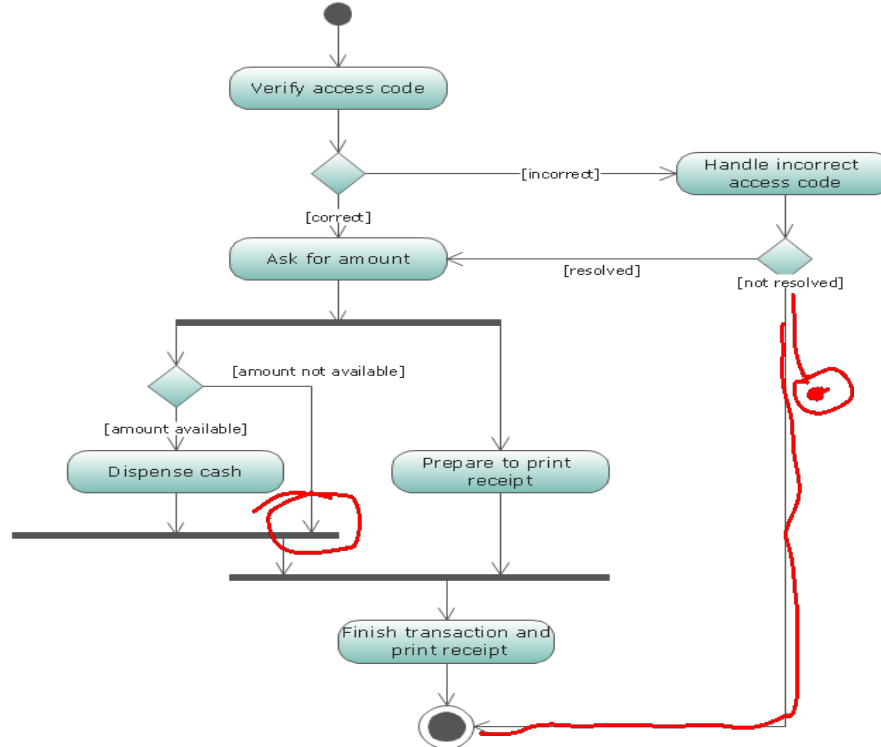
Join

- A join may have two or more incoming transitions and one outgoing transition
 - above the join, the activities associated with each of these paths continues in parallel
 - at the join, the concurrent flows synchronize
 - each waits until all incoming flows have reached the join, at which point one flow of control continues on below the join







Fork and Join Example



UML Activity Diagram



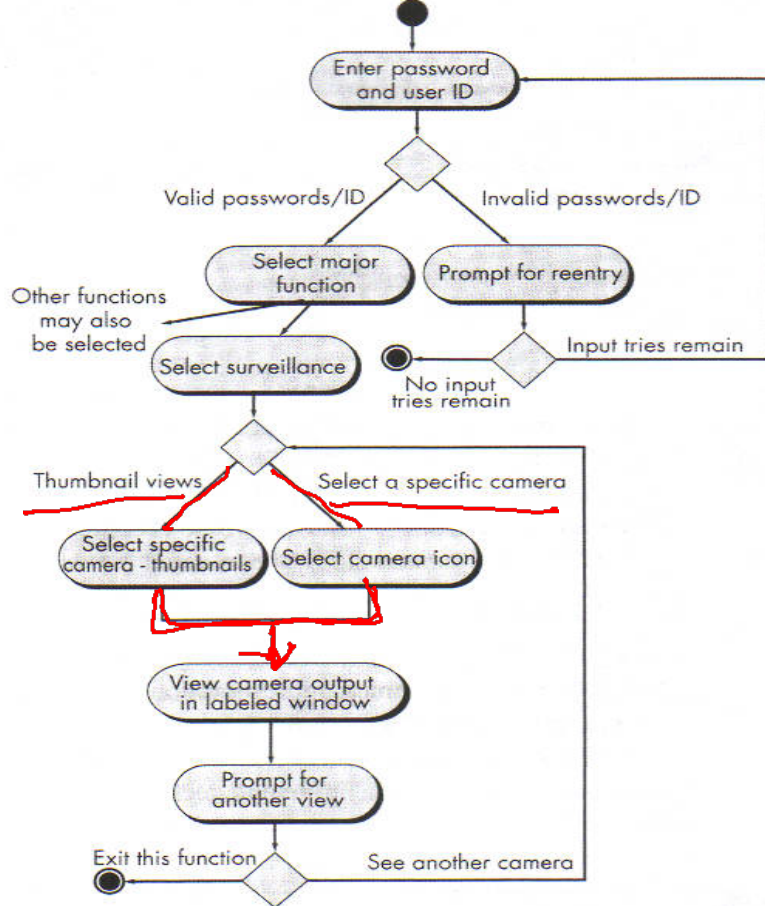
Revision: Activity Diagram

- **Rounded rectangle**  actions.
- **Diamond**  decisions/merge
- **Bars**  The start (fork) or end (join) of concurrent activities.
- **Black circle**  start (initial state)
- **Encircled black**  end (final state)
- **Arrow**  flow of control

Scenario:

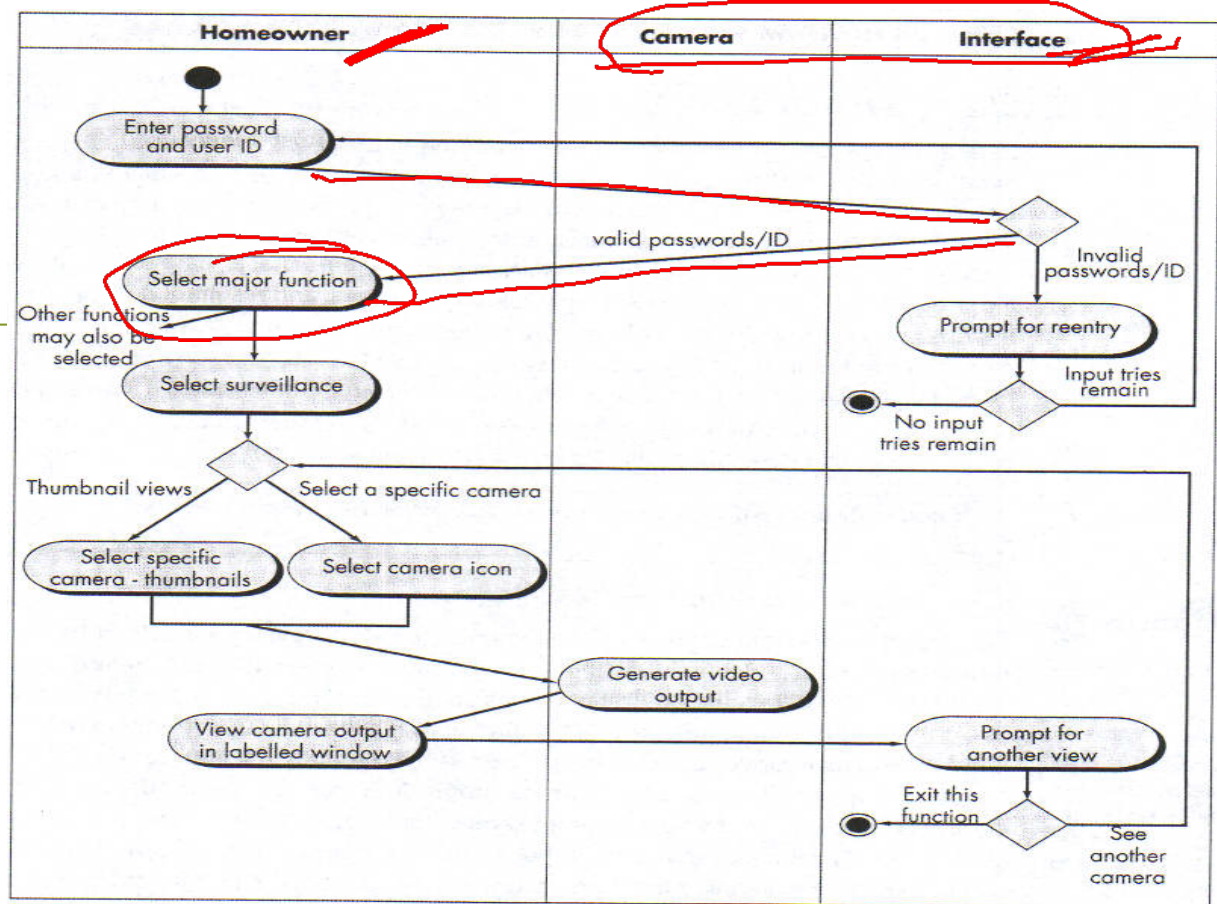
1. The homeowner logs onto the *SafeHome Products* Web site.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the "view" button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

Activity
diagram for
Access
camera
surveillance—
display
camera views
function

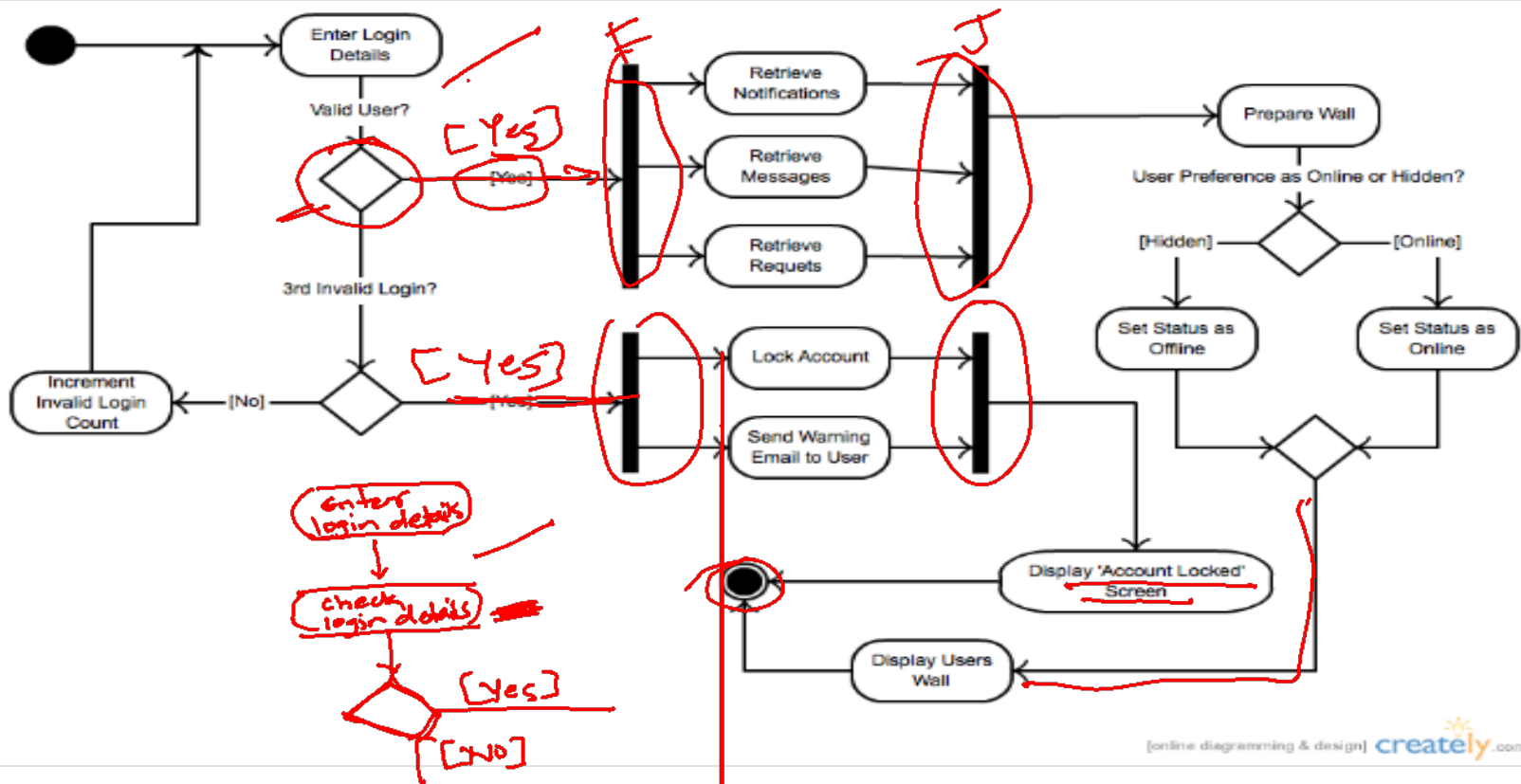


Swimlane Diagrams

- A variation of activity diagram
- Indicate which actor has responsibility for the action described by an activity rectangle
- A UML swimlane diagram represents the flow of actions, decisions and indicates which actors perform each



Facebook is an online social media website, where the users must register before using the it. Once the user is authenticated by site, he can retrieve notifications, messages and requests. Further, in the next step the user can change the user preferences as online or hidden. Finally the status will be displayed in user's wall. User has option to login by entering credentials for three times. Third unsuccessful login will lock the account, send warning email of the user and display 'account locker' message on the screen. Else the system will prompt for login credentials again.

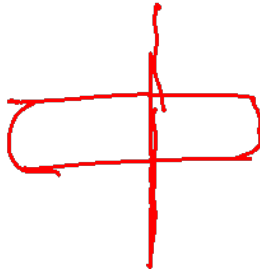


Swimlanes (1)

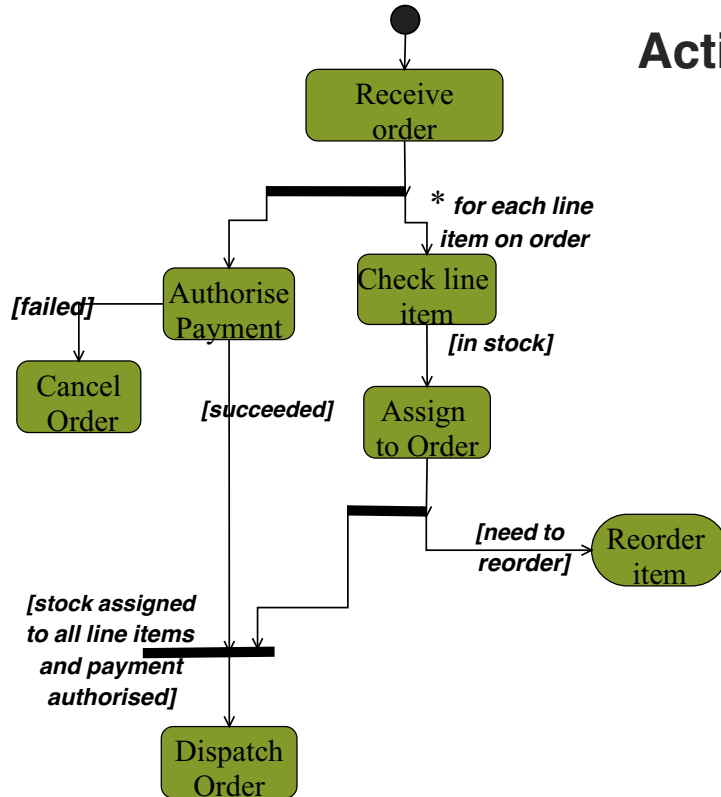
- A swimlane specifies a locus of activities
- To partition the activity states on an activity diagram into groups
 - each group representing the business organization responsible for those activities
 - each group is called a swimlane
- Each swimlane is divided from its neighbor by a vertical solid line

Swimlanes (2)

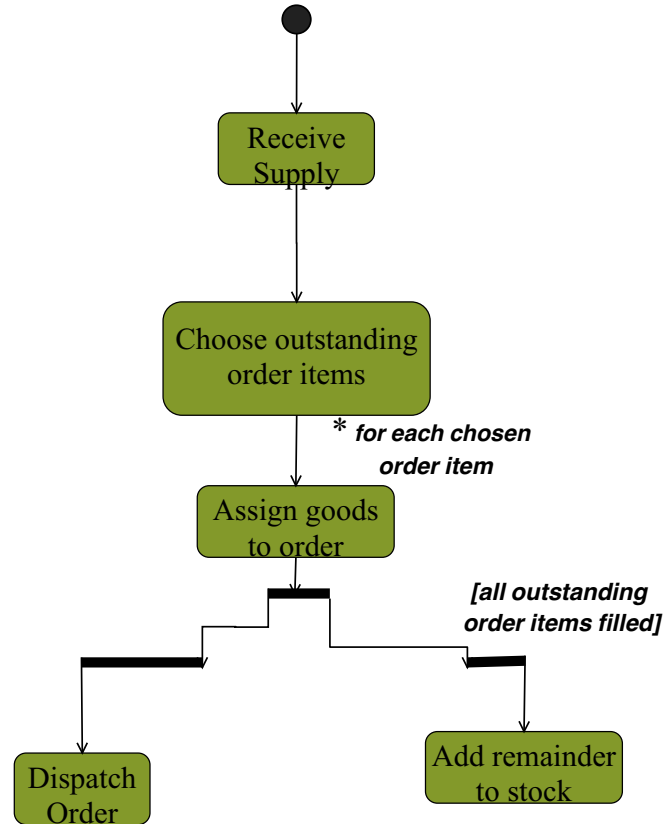
- Each swimlane has a name unique within its diagram
- Each swimlane may represent some real-world entity
- Each swimlane may be implemented by one or more classes
- Every activity belongs to exactly one swimlane, but transitions may cross lanes



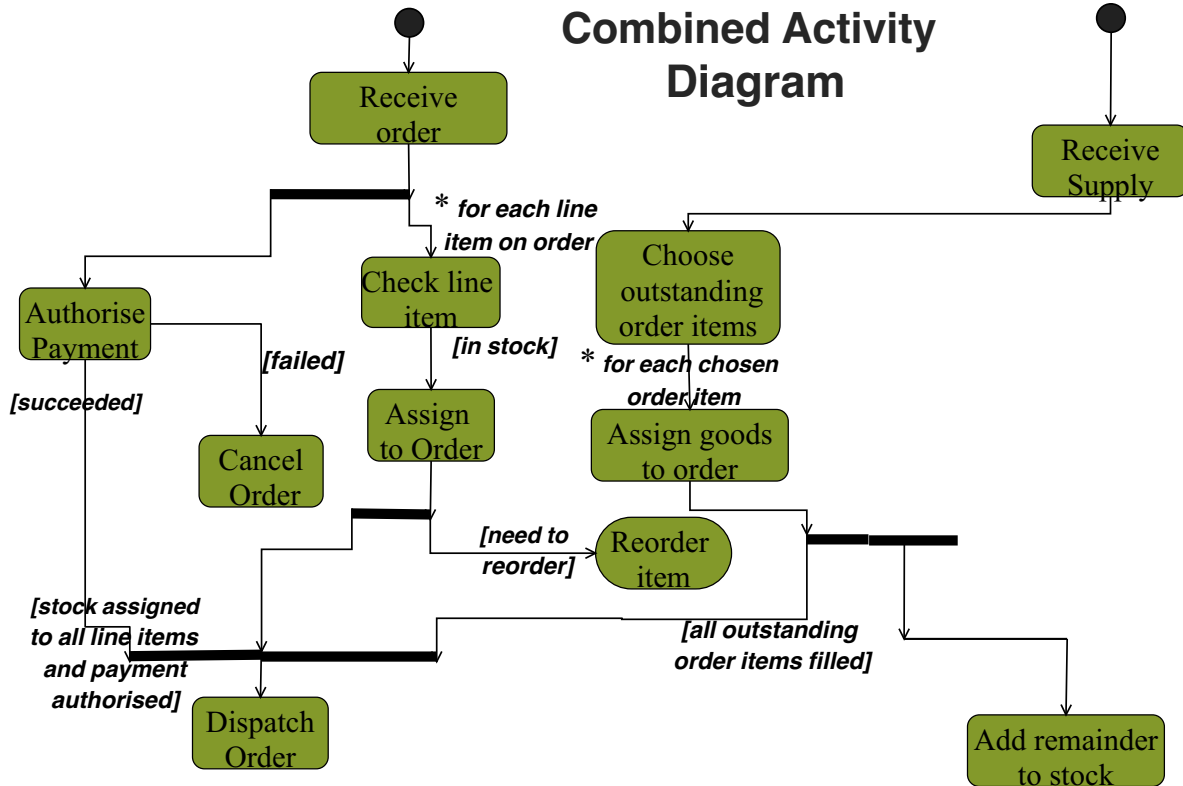
Activity Diagram for Receiving an Order

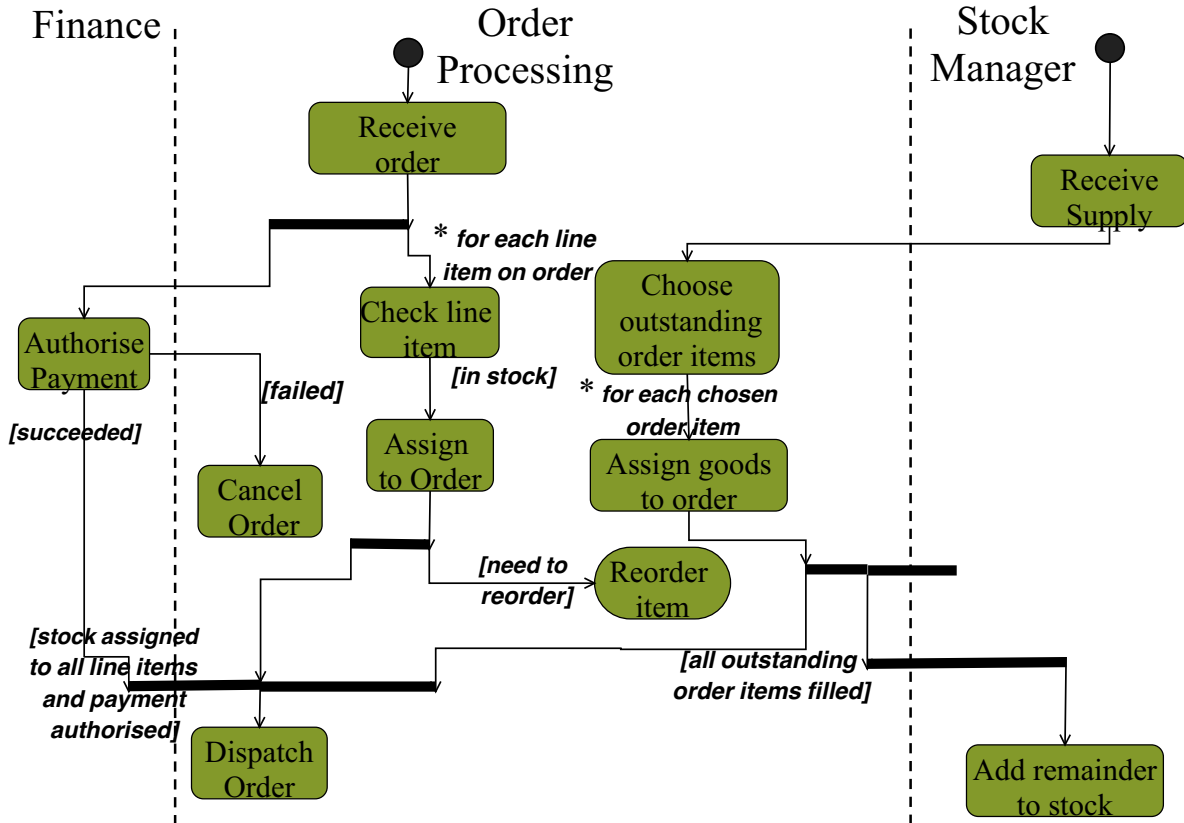


Activity Diagram for receiving Supply

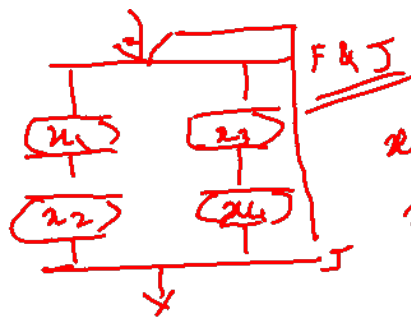


Combined Activity Diagram

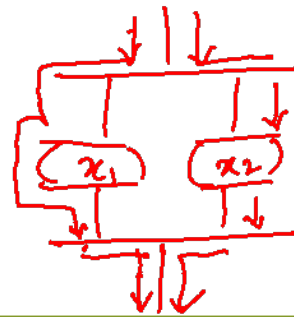




With Swimlanes



x_1, x_2, x_3, x_4
 x_3, x_4, x_1, x_2

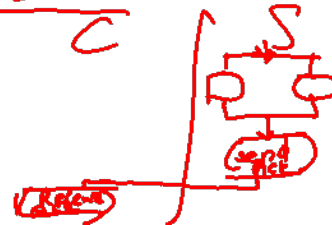


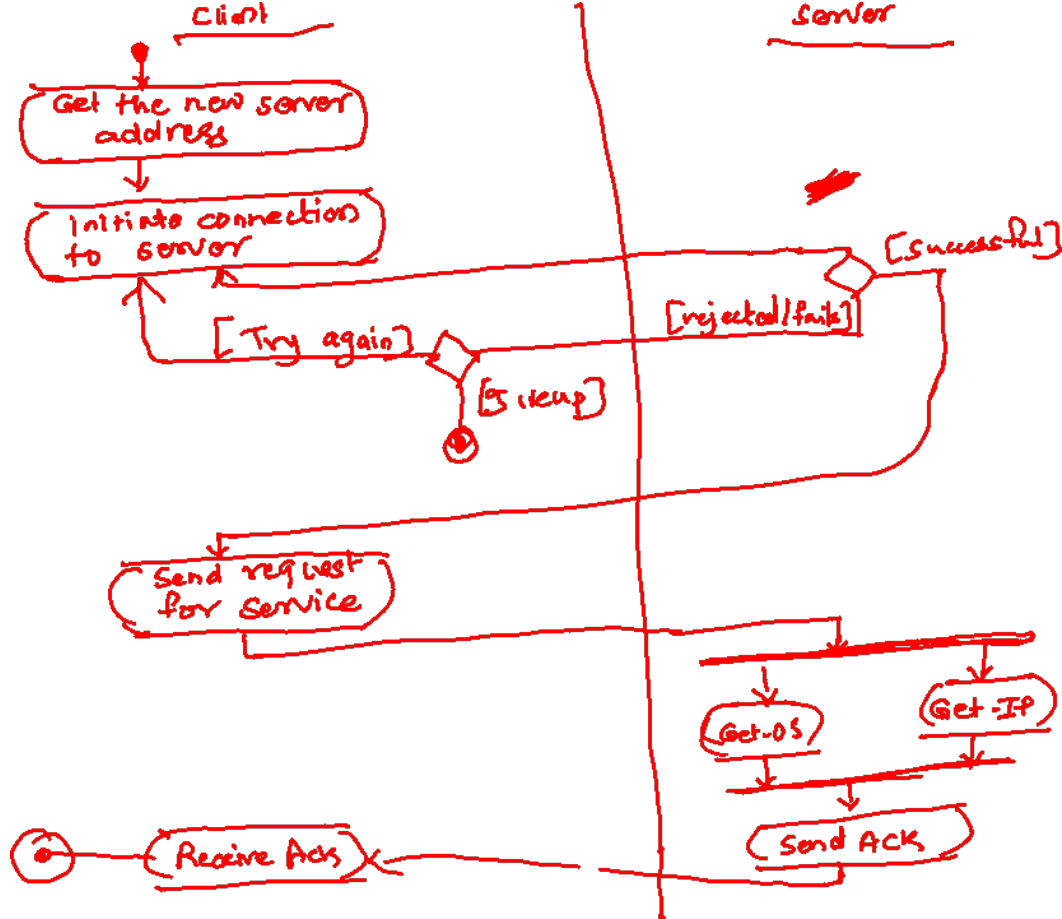
x_1, x_2
 x_2, x_1

- The client needs to know the whole address of a server for the communication. Client performs some work which includes, making a decision to initiate a connection to a server. If connection to the server fails, or the server rejects the connection, the client may try again or may give up. On successful connection, the client reads the request to execute the services. GET_OS and GET_IP are the services which will execute parallel and acknowledgement will be sent to the client by the server.

Draw an activity diagram.

Swimlane





- The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions.
- The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned. The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed.
- If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back. If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction.

- The ATM will also maintain an internal log of transactions to facilitate resolving ambiguities arising from a hardware failure in the middle of a transaction. Entries will be made in the log when the ATM is started up and shut down, for each message sent to the Bank (along with the response back, if one is expected), for the dispensing of cash, and for the receiving of an envelope.
- Log entries may contain card numbers and amounts, but for security will never contain a PIN. To avail ATM facility, a customer is required to open/have an account in the bank and apply for the ATM card. A customer can have one or more accounts and for each account, only one ATM card will be provided.
- The bank also provides SMS updates for every transaction of customer's account. To obtain SMS updates, customer is required to register his/her mobile number against his account in the bank.