

Library Example

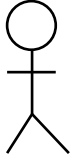
Before issuing the loan we need to check:

- 1) The borrower has no overdue fines
- 2) The borrower has not already reached the maximum number of loans that they are allowed.
- 3) None of the borrower's current loans are overdue

Developing Sequence Diagrams

- Identify the relevant objects involved in the computation
- Establish the role of each object
- Identify the controller
- Identify the collaborators
- Decide on the messages between objects

Sequence diagram notation

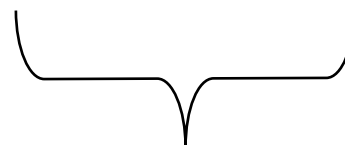


Actors

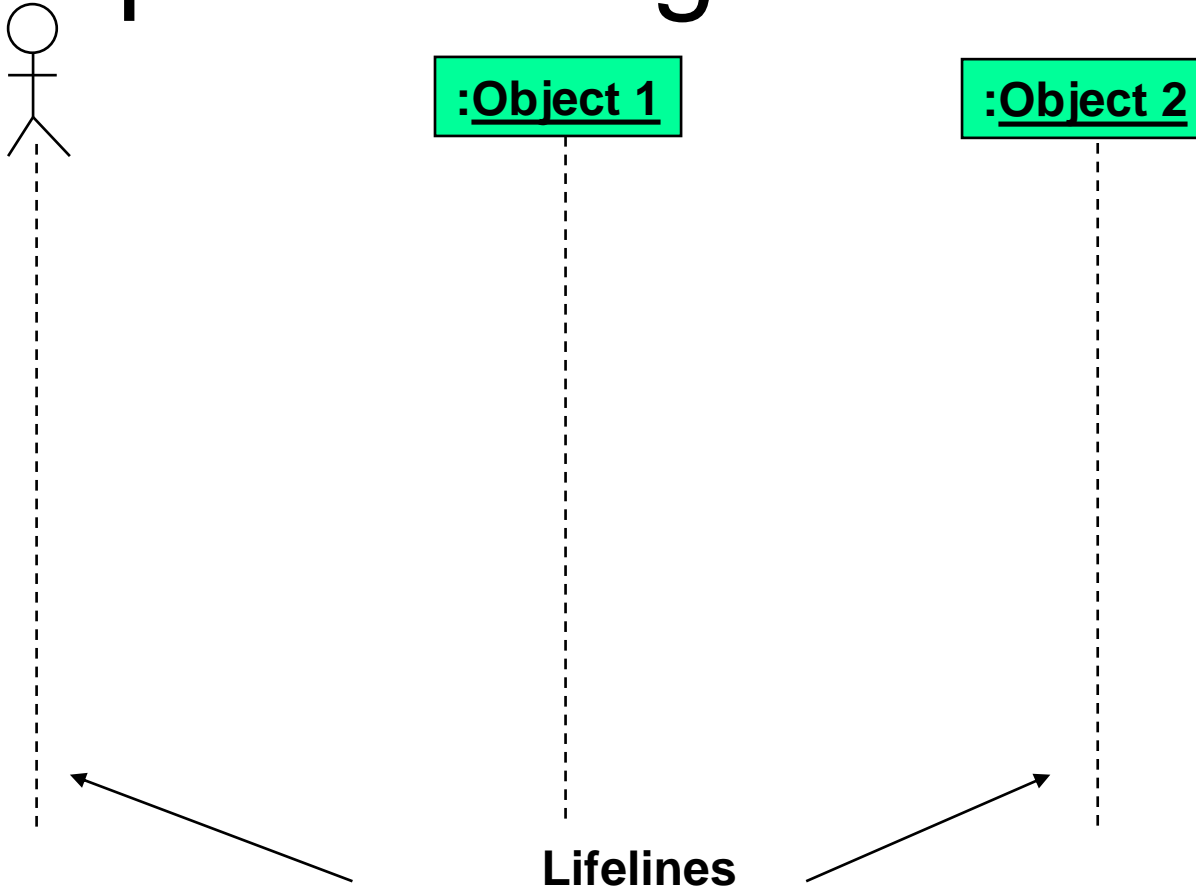
:Object 1

:Object 2

Objects

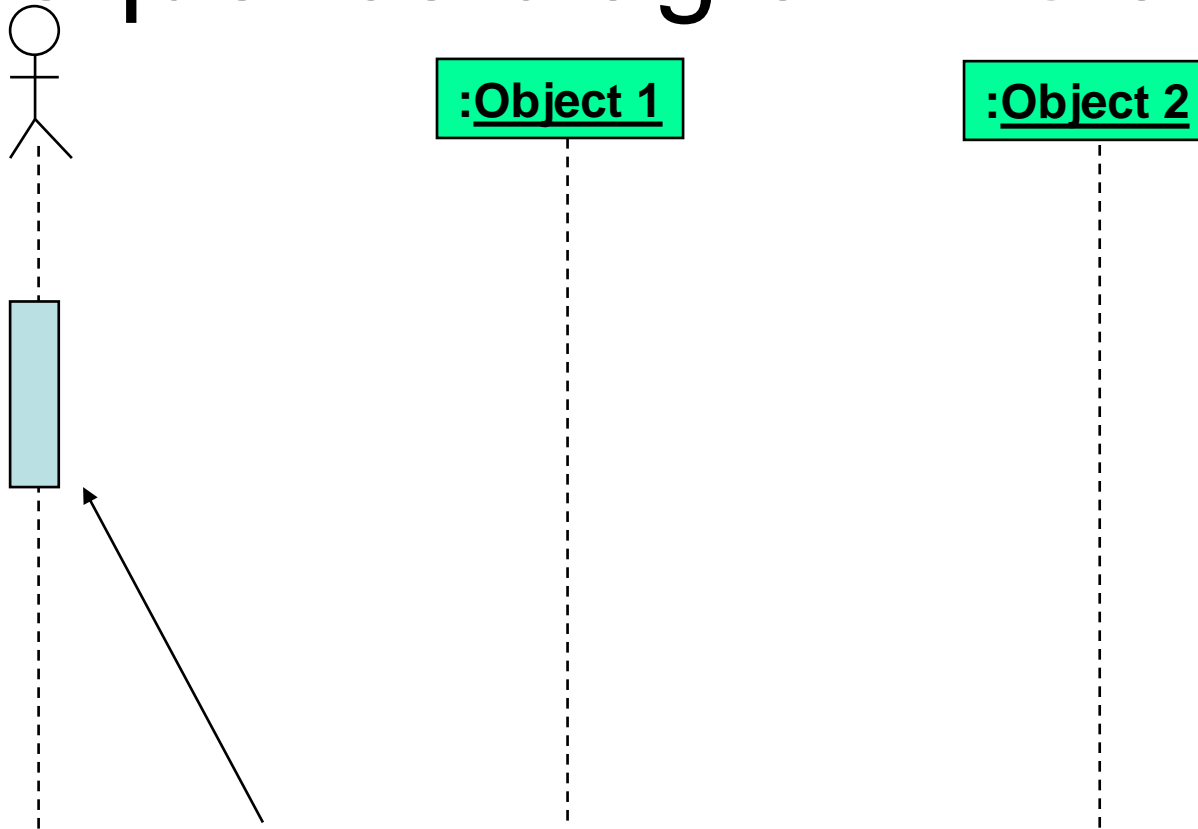


Sequence diagram notation



Identify the existence of the object over time.

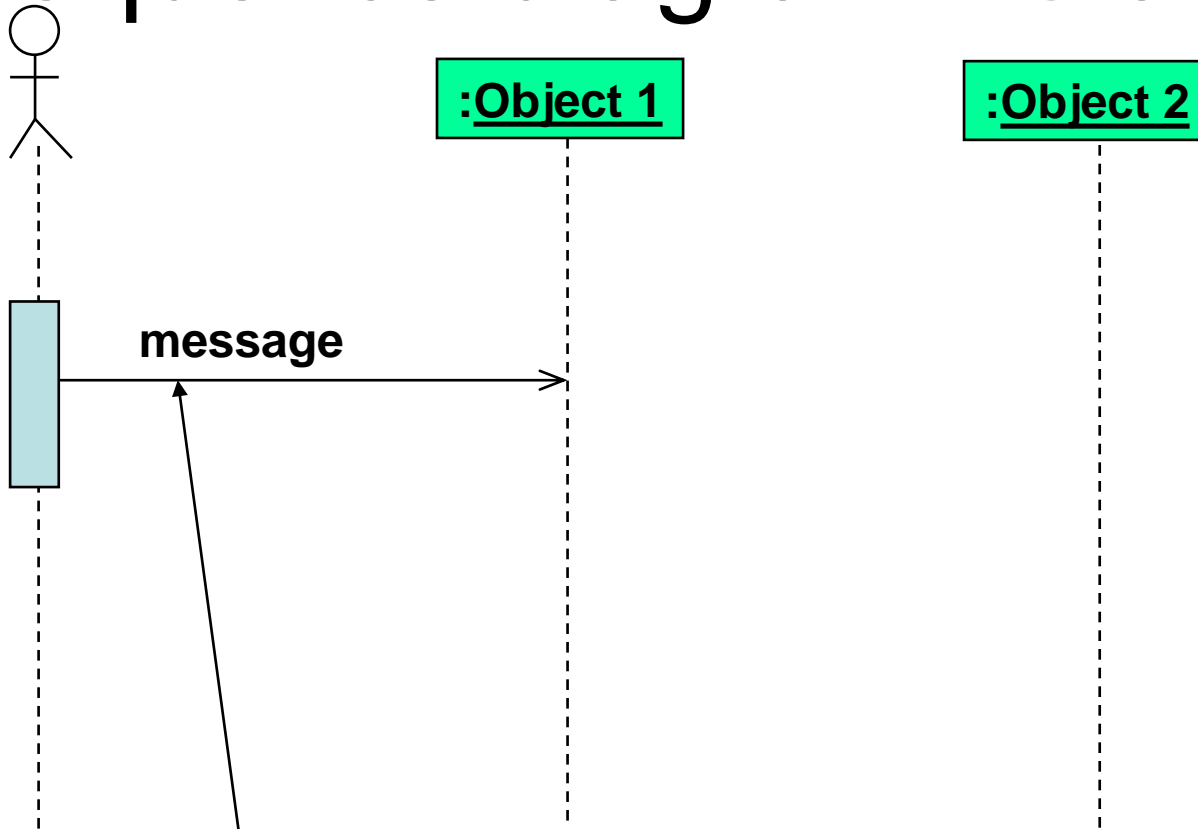
Sequence diagram notation



Activations

Indicate when an object is performing an action

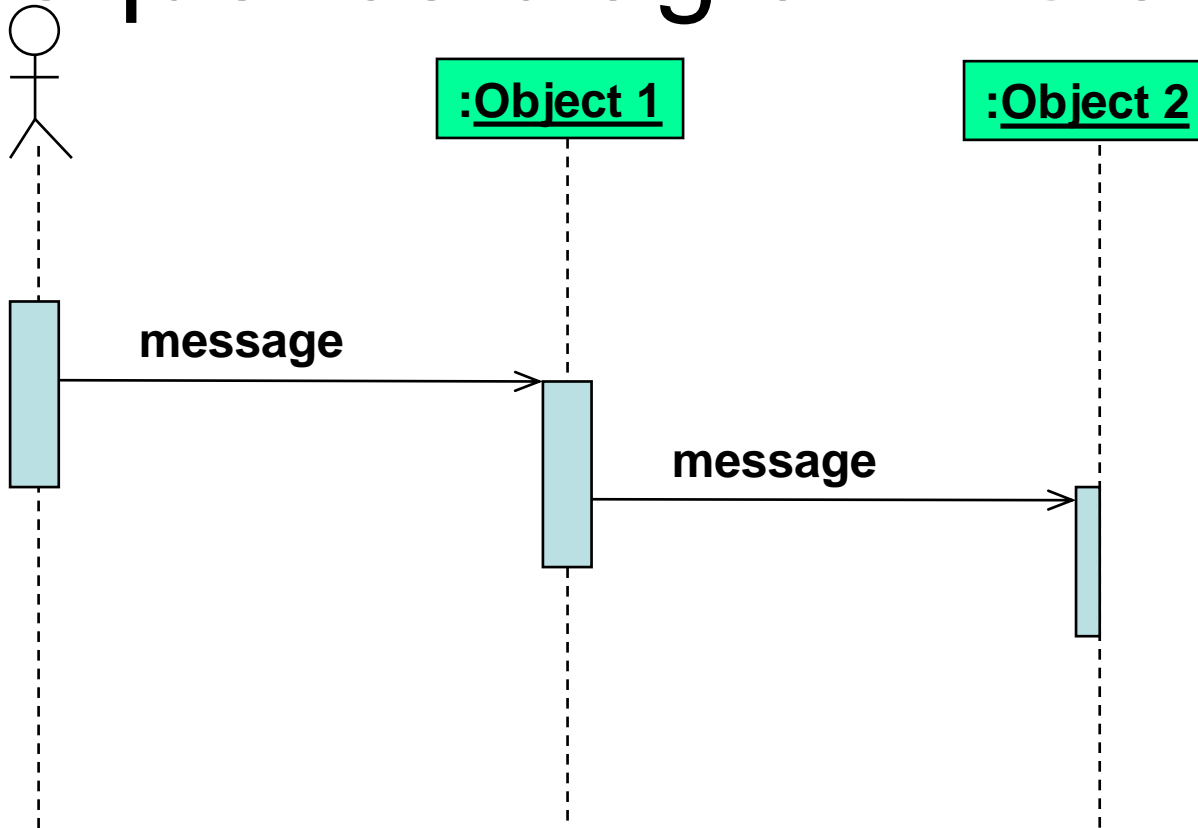
Sequence diagram notation



Messages

Indicate the communications between objects

Sequence diagram notation



Sequence

Vertical position signifies sequence – earlier messages appear nearer the top.

Use case elaboration

- We define use cases as sequences - primary and alternative paths
- Now we take sample sequences and build sequence diagrams
- This gives us the objects
- And it gives us the relationships
- And it gives us the operations

Invoicing use case (1)

Use Case Number: 99	Use Case Name: Invoice Customer
Brief Description: This is run daily to send invoices to customers. Items that have been delivered are billed all on the same invoice. Customers are only billed once a month.	
Actors: Daily batch run, customer (indirectly, through post)	
Frequency of Execution: Daily	
Scalability: Only one instance of this runs at any one time.	
Criticality: Essential. Every days delay to printing invoices affects the bank balance considerably. Not running this for 7 days could trigger a serious cash flow problem.	
Primary Path: The following sequence is carried out for every customer on the sales ledger who has not been billed in the last month: <ol style="list-style-type: none">1. Get sales items from the sales ledger.2. Get customer details from the customer file, covering billing address details.3. Get any credits that the customer has.4. Get discount details for customer.5. Print the invoice header6. Print the line items on the invoice7. Calculate any discounts8. Apply any credits9. Calculate and print the invoice total10. Calculate and print the VAT11. Mark items on sales ledger as invoiced	

The Primary Path

The following sequence is carried out for every customer on the sales ledger who has not been billed in the last month:

1. Get sales items from the sales ledger.
2. Get customer details from the customer file, covering billing address details.
3. Get any credits that the customer has.
4. Get discount details for customer.
5. Print the invoice header
6. Print the line items on the invoice
7. Calculate any discounts
8. Apply any credits
9. Calculate and print the invoice total
10. Calculate and print the VAT
11. Mark items on sales ledger as invoiced

Print Invoice - class diagram



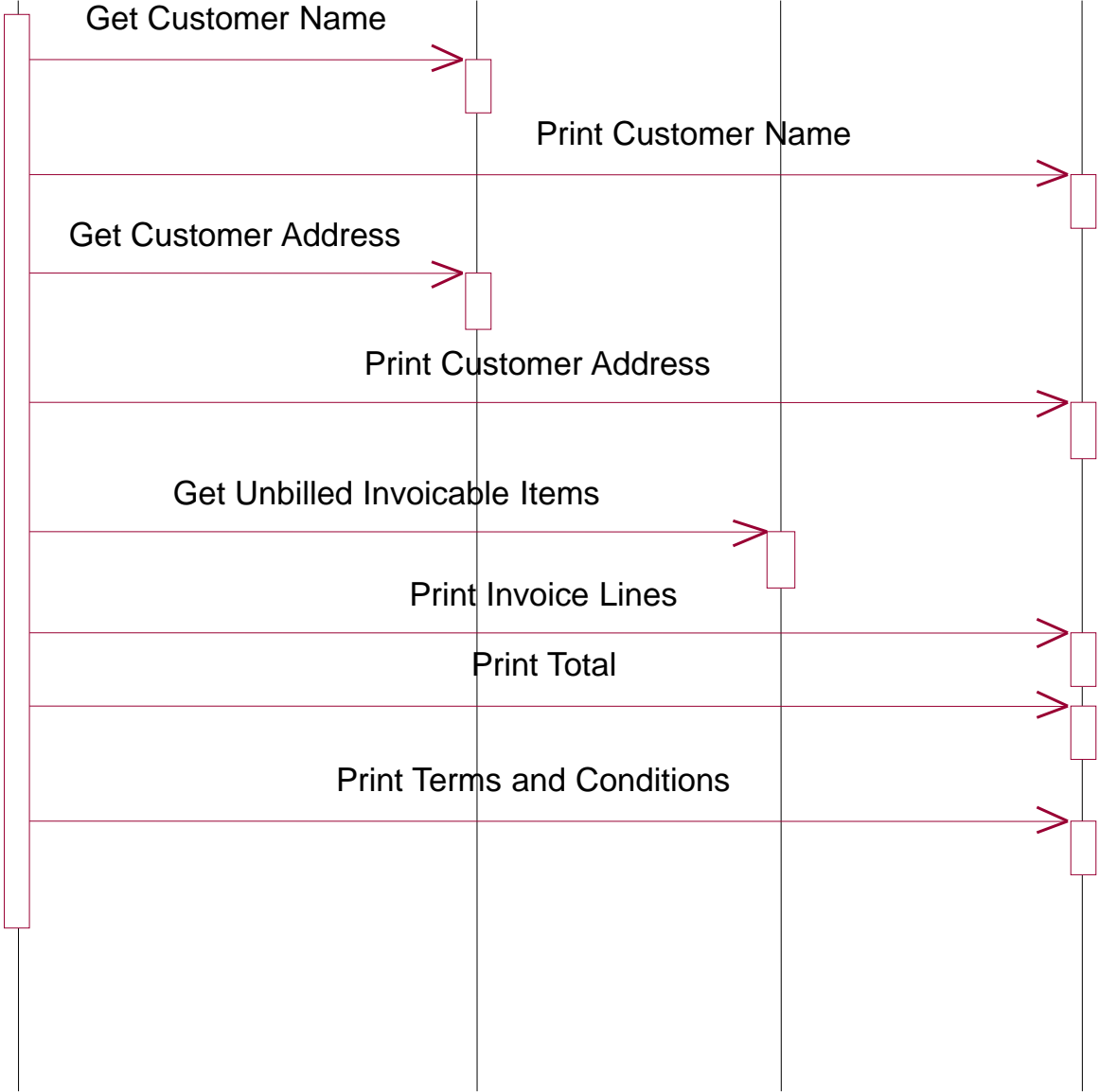
We can see that as the objects communicate we need a relationship between them.

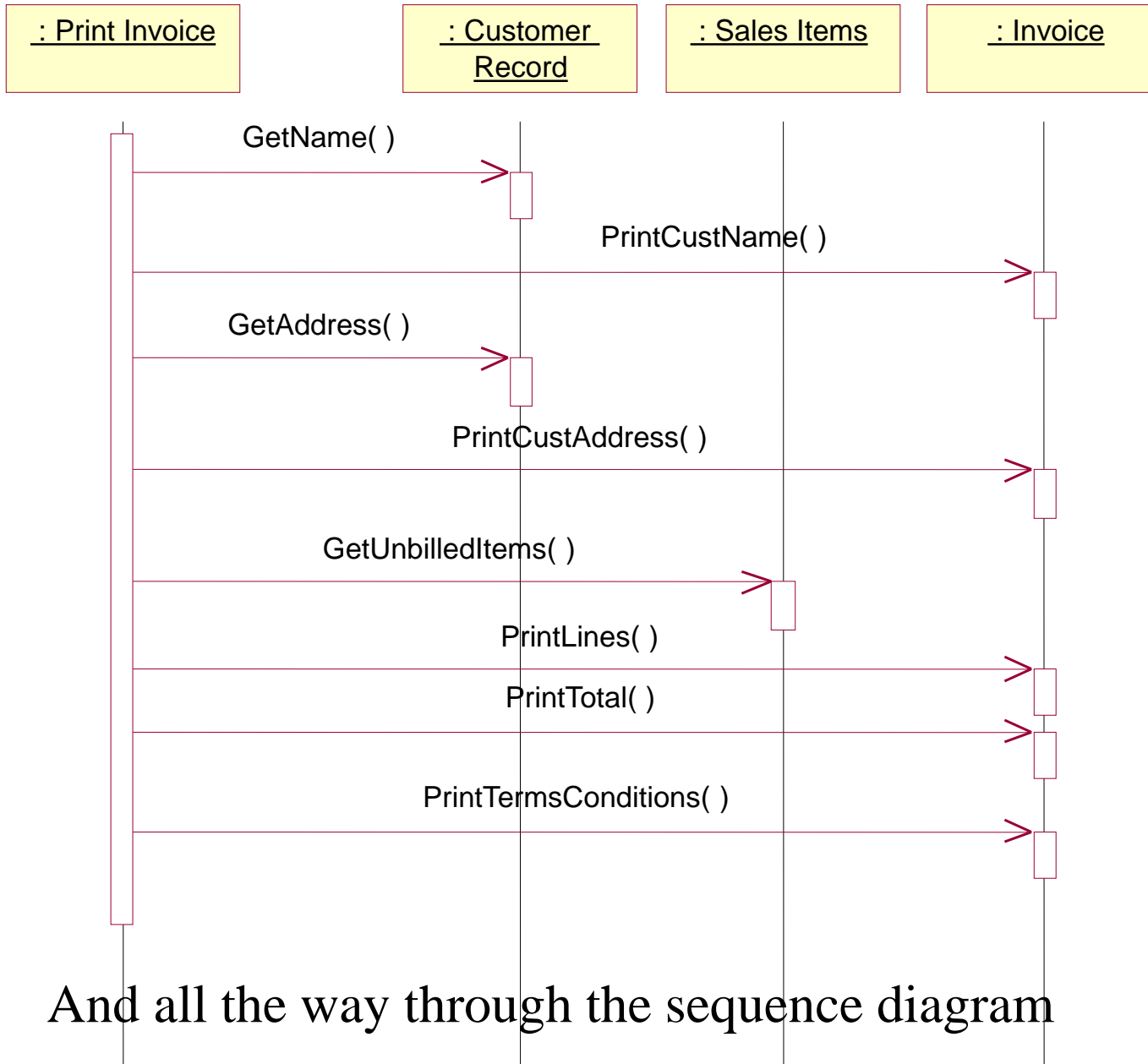
: Print Invoice

: Customer Record

: Sales Items

: Invoice





And all the way through the sequence diagram

Working from a scenario

Sending an email

1. Press “New ” email icon
2. Enter person’s name in “To” section
3. Type subject
4. Type contents
5. Press Send button
6. System looks up email address in address book
7. System submits the email to the email server

Developing a Sequence Diagram

Work through a scenario step by step

Make actors communicate with screens, icons, menus

Make the screen actions (etc) trigger actions with
objects

Convert the actions to operations

Need for Sequence Diagrams

- Tie use-cases and object models together
- Use the sequences in use cases
- Identify objects
- Identify relationships
- Identify operations

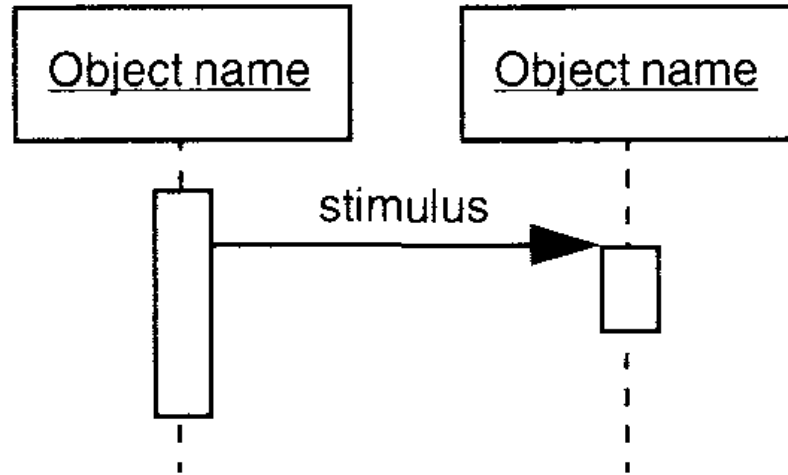
Sequence Diagrams

- Used to model *object interactions* on a time axis.
 - Dynamic aspects of a system.
 - How objects collaborate to realize a use case.
- Distribute use case behavior to classes.
 - Starting to look at *how* the system does something rather than just what is done.
- For now, high level interactions
 - Look at more detailed level later

Messages

- Interactions are represented by *messages* sent from one object to another.
- Long narrow vertical box on lifeline indicates *focus of control*.
 - When an object is active, either because it is doing something, or because it has sent a message to another object that is doing something on its behalf.
 - Not always shown.

Messages



Arrows are labeled with the name of the message, or stimulus, that they represent.

Sequence Diagram vs Activity Diagram

- When do I use a sequence diagram and when do I use an activity diagram?
 - How do I decide which one is appropriate?
- Ans: First of all, you don't have to choose. You can do both.
- Depends on what you want to show.
- Activity diagrams focuses on the sequence of actions.
 - Doesn't show *why* an object does something.
- Sequence diagrams show “flow of information” (Who says what to whom).

Class Activity

- Draw a sequence chart for the vending machine use case “Customer purchases soft drink with credit card.”
- Active objects:
 - Customer
 - Vending Machine
 - Credit Card Processing Center
- Use case description on next slide.

Vending Machine Use Case

<i>Use case name</i>	Customer purchases soft drink with credit card
<i>Participating actor</i>	Initiated by Customer Credit Card Processing Center
<i>Flow of events</i>	<ol style="list-style-type: none">1. The customer swipes his credit card.2. Vending machine sends message to processing center.3. Processing center confirms card and provides available credit.4. Vending machine indicates that customer can select product.5. Customer presses button to select product.6. Vending machine sends charge to processing center.7. Processing center confirms charge8. Vending machine dispences selected product.9. Customer removes product.
<i>Entry condition</i>	The customer stands in front of a soft drink vending machine that accepts credit cards. Customer has a valid credit card and wants to purchase a soft drink using it.
<i>Exit condition</i>	Customer has soft drink. Credit card is charged.

Use-Case template

- A use case typically includes the following information:
- **Name:** The name of the use case
- **Brief Description:** A brief description of the role and purpose of the use case
- **Flow of Events:** A textual description of what the system does in regard to a use case scenario (not how specific problems are solved by the system). Write the description so that the customer can understand it. The flows can include a basic flow, alternative flows, and subflows.
- **Key scenarios:** A textual description of the most important or frequently discussed scenarios
- **Special Requirements:** A textual description that collects all of the requirements of the use case that are not considered in the use-case model, but that must be taken care of during design or implementation (for example, non-functional requirements)
- **Preconditions:** A textual description that defines a constraint on the system when the use case starts
- **Post-conditions:** A textual description that defines a constraint on the system when the use case ends
- **Extension points:** A list of locations within the flow of events of the use case at which additional behavior can be inserted by using the extend-relationship

Use-Case template (Sample)

Use case Id: 20

Use case name: Withdraw Cash

Primary actors:

- Customer
- ATM

Stakeholders and their interest:

- **Customer** (end user who retrieves money)
- **Security administrator** (monitors security issues on ATM)
- **IT administrator** (installs patches of the application)
- **Bank manager** (advertise other financial products)
- **Developers** (maintains nice development environment)

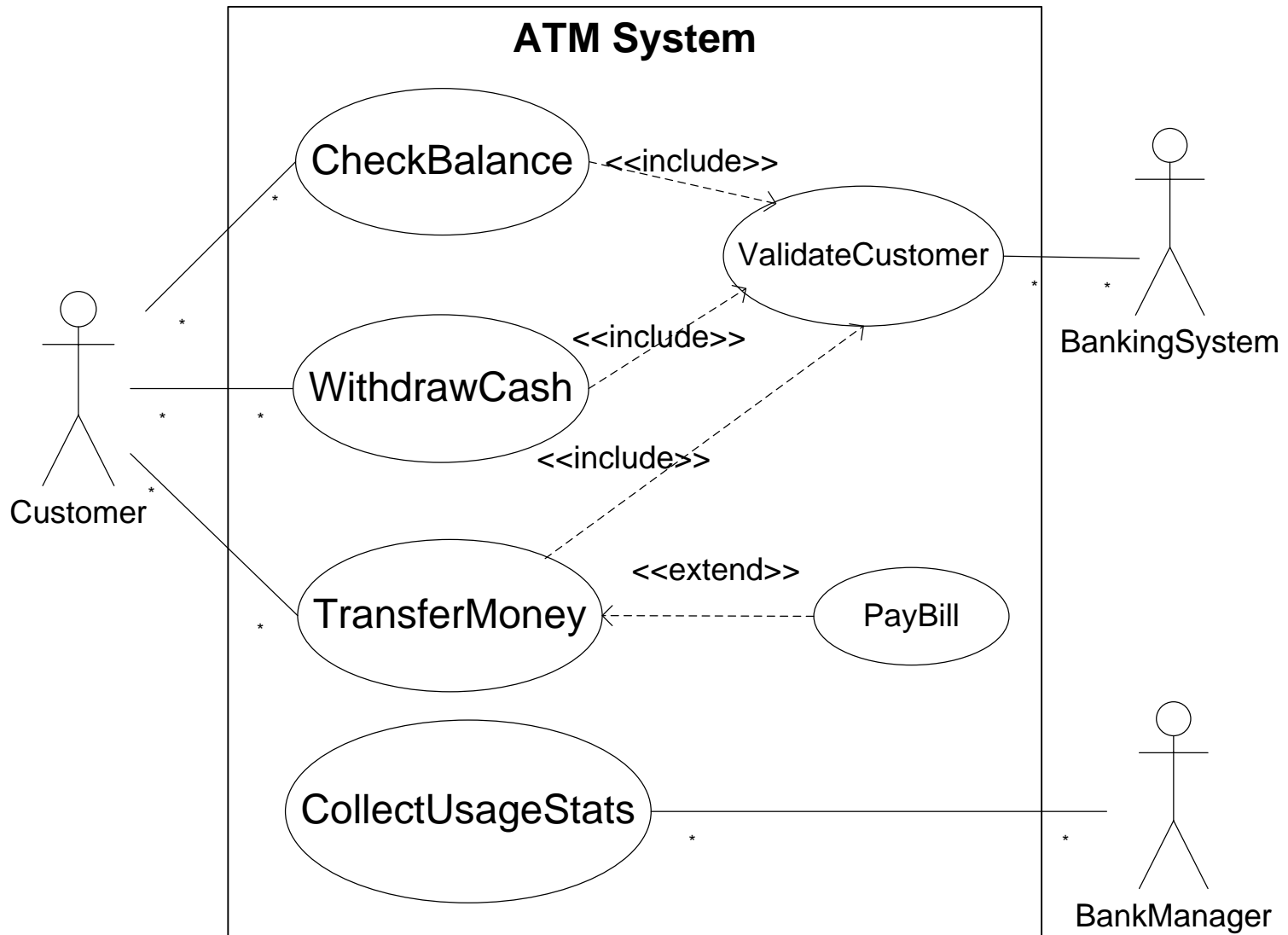
Pre-condition:

- Customer must possess a bank card.
- Network connection to the bank system must be active.
- System must have at least some cash.
- Cash withdrawal service must be available.

Main success scenario:

- Ask user withdraw amount
- Check if that amount is available
- User receives that due amount

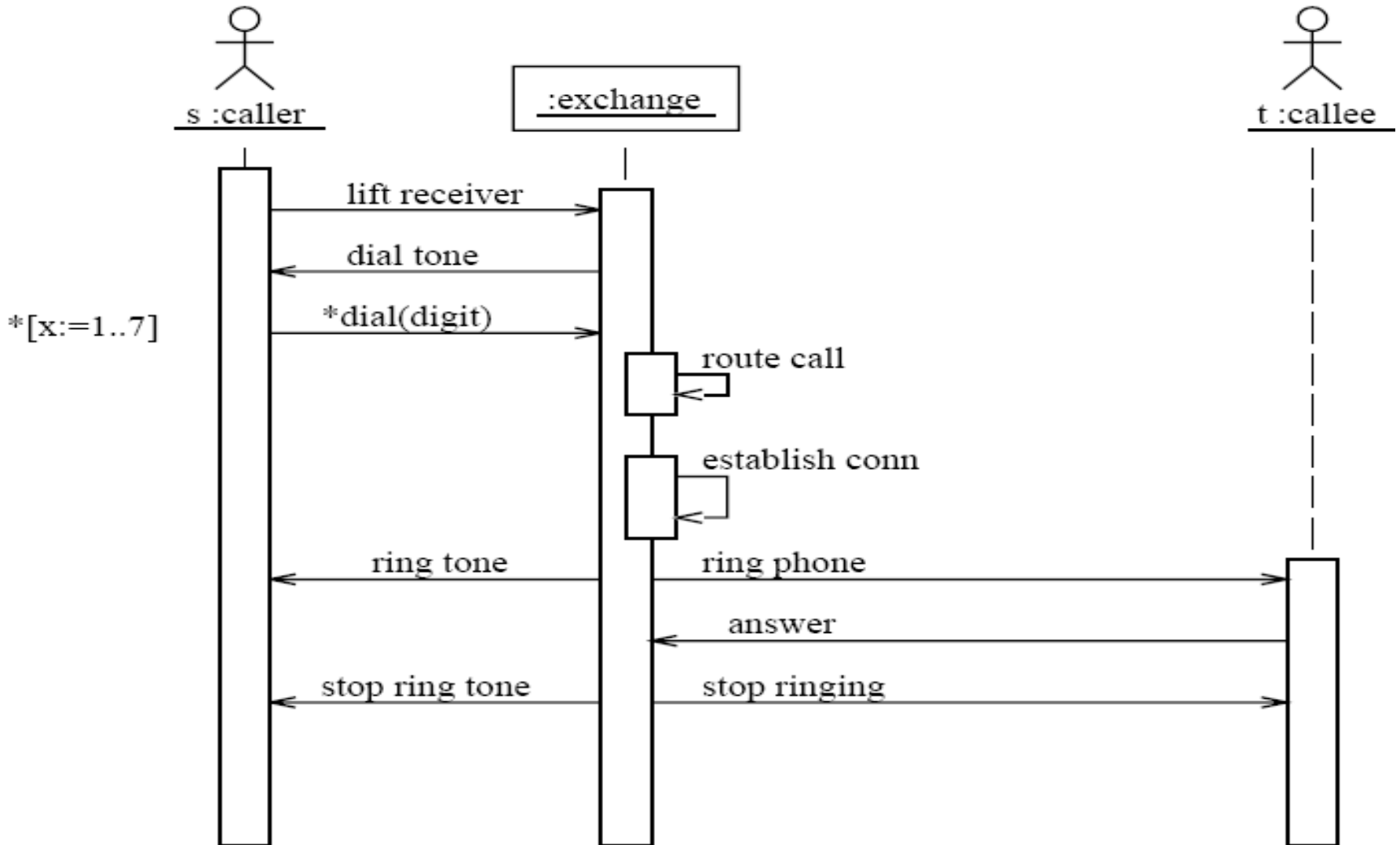
Elaborated Use Case Diagram for ATM



Use Case Diagrams

- Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact.
- Describes a set of sequences.
- Each sequence represents the interactions of things outside the system (*actors*) with the system itself (and key abstractions)

Sequence Diagram – An Example



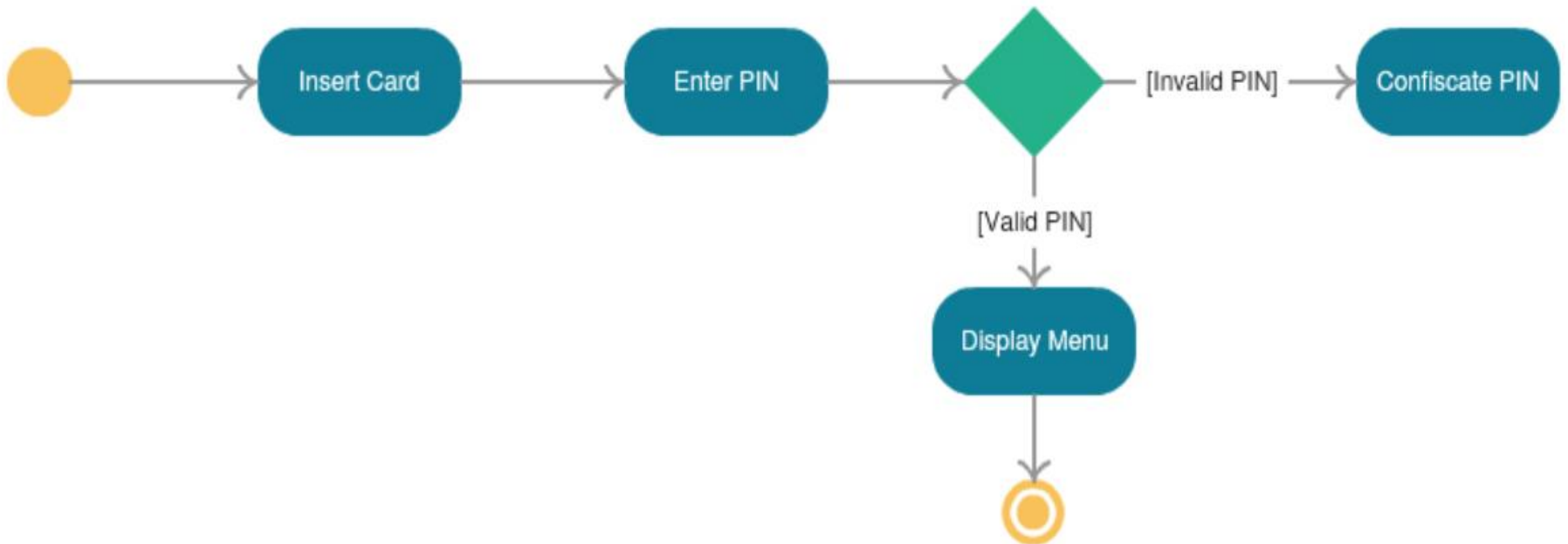
Sequence Diagrams

- X-axis is objects
 - Object that initiates interaction is left most
 - Object to the right are increasingly more subordinate
- Y-axis is time
 - Messages sent and received are ordered by time
- Object life lines represent the existence over a period of time
- Activation (double line) is the execution of the procedure.

Activity Diagram

- Activity diagrams represent workflows in a graphical way.
- They can be used to describe the business workflow or the operational workflow of any component in a system.
- It also enables developers to figure out constraints and conditions that cause particular events.

Activity Diagram



Summary

- In this chapter we have discussed software requirements. The need of requirements and some types.
 - Functional requirements.
 - Non-functional requirements.
 - User-requirements.
 - System requirements.
- We have seen some diagrams through which we can represent our system/ website in different modules.
- We have discussed how different use-cases can show different sequence of actions performed in system.

Thank you