

# **420-N23-LA Introduction to IoT**

## IoT Process Design and Diagramming

---

# Design of IOT



A fun IOT Video

# IOT

---

- The Internet of Things is about extending the power of the internet beyond computers and smartphones to a whole range of other things, processes, and environments.
- **How do we do that? How do we extend the power to the internet?**
  - By putting components in the world around you which interact and sense the physical world and connecting those to the internet. These components are combinations of **sensors** and **actuators**.

# Basics of an IOT Activity

---

1. Sense
2. Connectivity
3. Process
4. Interface



IOT Fridge

# IOT Design Process

---

- ▣ IoT (Internet of Things) process design is the planning and structuring of how connected devices **collect data, communicate, and trigger actions**.
- ▣ It defines the steps: how sensors gather information → how data travels through the network → how it's analyzed → and how decisions or automated actions are made.
- ▣ Think of it like a blueprint for making sure smart devices, apps, and systems work together smoothly.

# We're Missing Juice

---

## ■ Sense

- User leaving home
- Sense no juice in the fridge

## ■ Process

- What do I do with this information?
- I should inform the user...
- I should order from Amazon based on set limits.

## ■ Communicate

- Inform the user Juice is missing
  - By cell phone
  - By adding to a grocery list automatically
  - By verbally announcing it via API to Google assistant.

# Sense Object – Juice Quantity

- What type of sensor?
- Is juice needed?
- No juice or 1/3 juice left? (Predicting)



# Use Cases

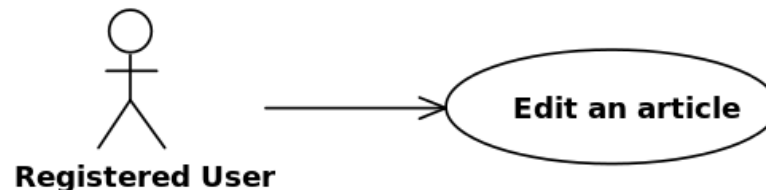
---

- A use case is a methodology used in system analysis to identify, clarify and organize system requirements.
- In software and systems engineering, a use case is a list of **actions** or event **steps** typically defining the interactions between a role (an actor) and a system to achieve a goal.
- The actor can be a human or other external system.

# Use Cases

---

- Use cases are done using a text template, then drawn using a actor-bubble notation to make it more understandable.
- We will look at use cases high level – you will revisit them in detail later in your program.

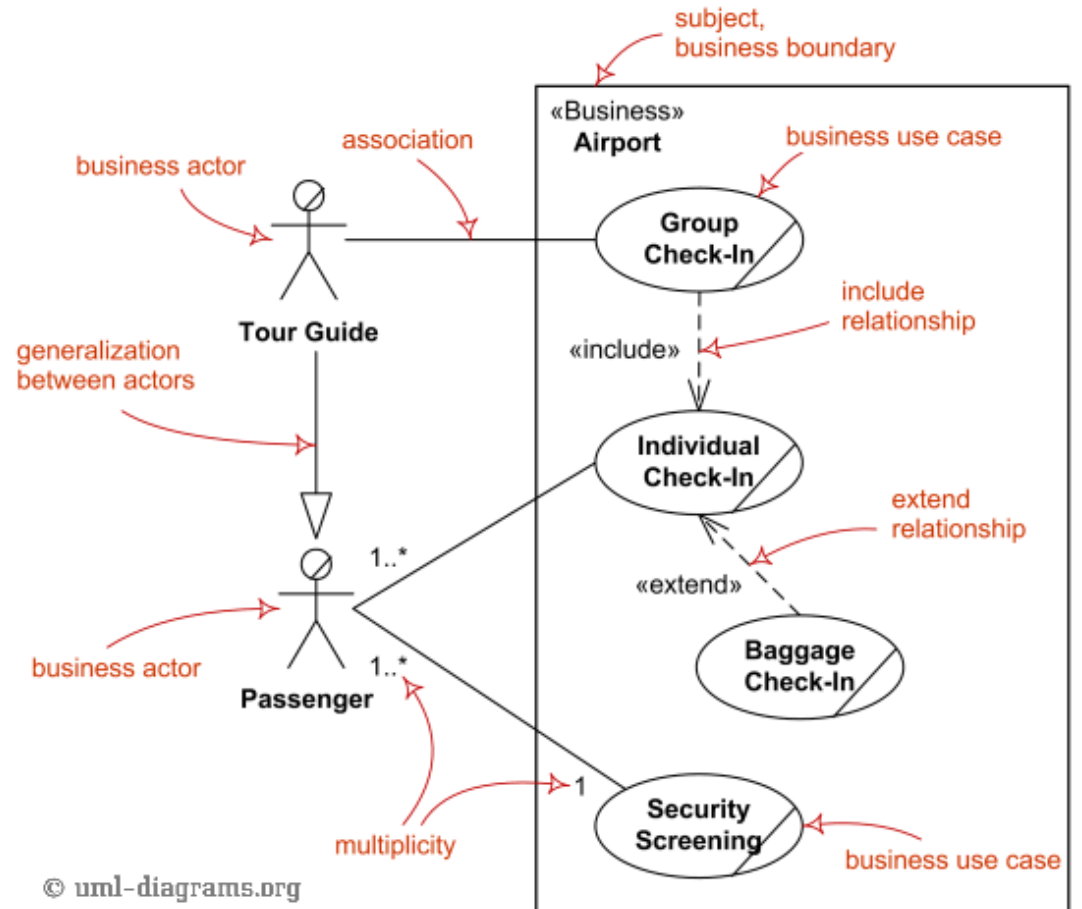


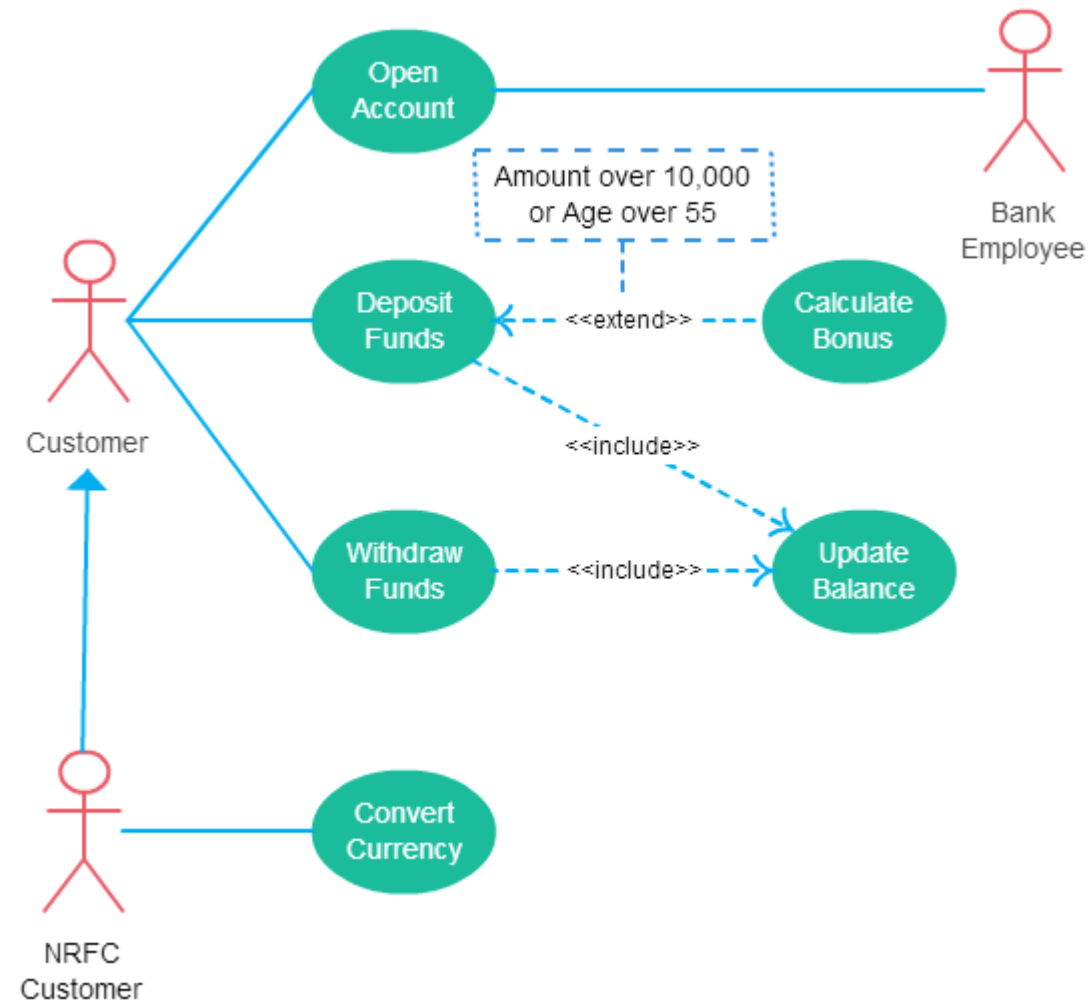
# Example of a Written Use-Case

Section	Function
<b>Use case name</b>	The name of a single function of the system. Think of this as “actor does thing”.
<b>Actor(s)</b>	The person or object (computer) doing the action.
<b>Trigger(s)</b>	What triggers the action? In computer speak, this might be an “event”.
<b>Preconditions</b>	What conditions or “state” needs to exist in order for this use-case to be valid, or to happen? Usually if the preconditions are not met, the use case shouldn’t happen. <i>Name these, PRE1, PRE2, PRE3 for example.</i>
<b>Postconditions</b>	How will the system be after the use case is done. What must change? <i>Name these, POST1, POST2, POST3, etc.</i>
<b>Normal Flow(s)</b>	This is the flow of the actions from A-Z. Avoid “IF” statements. If you feel there is an if statement, then consider making TWO use cases. Consider the example of paying Interac, by PIN or by Contactless Chip.
<b>Alternate Flow(s)</b>	Here, you might have a few. This is what happens when things don’t go as expected. Sometimes you can refer to other use cases here. <i>Name them A1, A2, A3, for example.</i>

# Example of a Graphical Use-Case Diagram

- Actor
  - Can be specialized (Think of Java and inheritance)
- System boundary
  - Limit all cases to a specific subject.
- Include
  - One function NEEDS another function to work.
- Extend
  - One function CAN USE another function if needed.

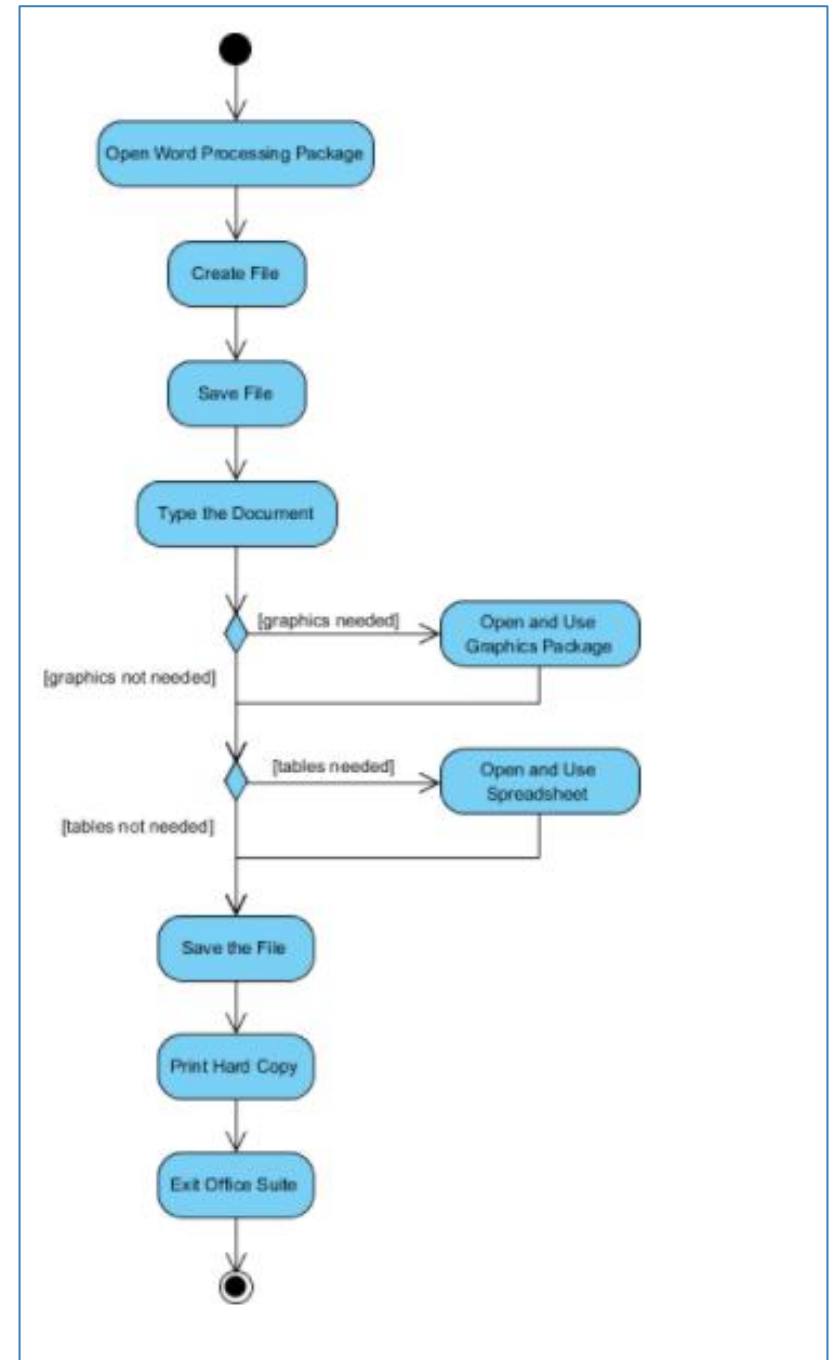


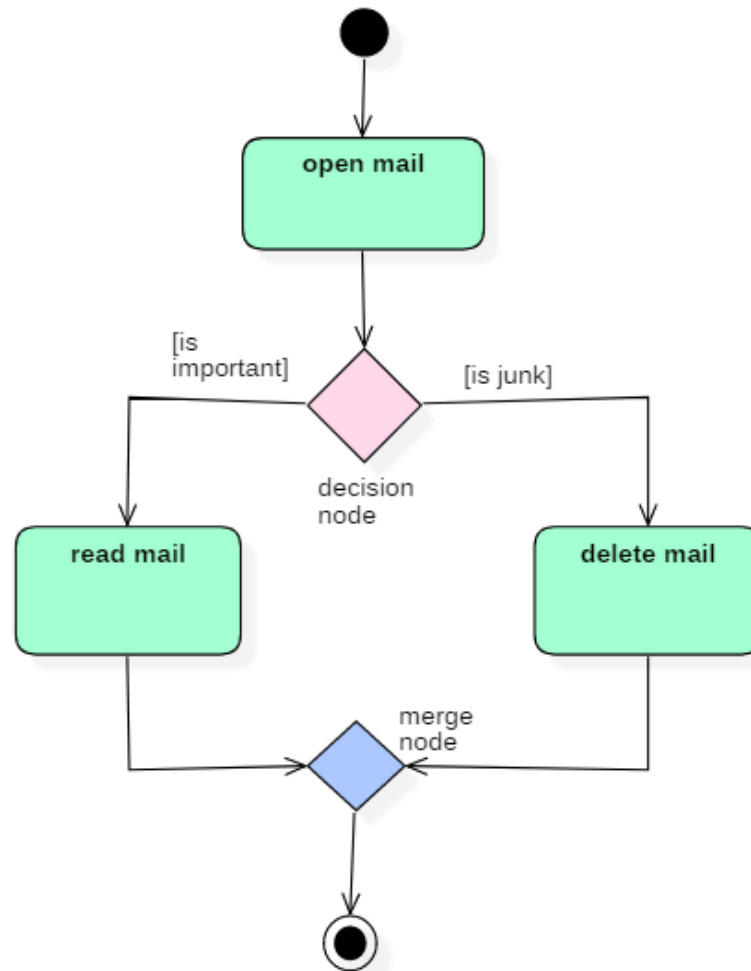


Example of a Bank System

# Activity Diagram

- We use **Activity Diagrams** to illustrate the **flow of control** in a system and refer to the steps involved in the execution of a use case.
- We model **sequential and concurrent** activities using activity diagrams.
- We basically **depict workflows visually** using an activity diagram.
- An activity diagram focuses on **condition of flow and the sequence in which it happens**.

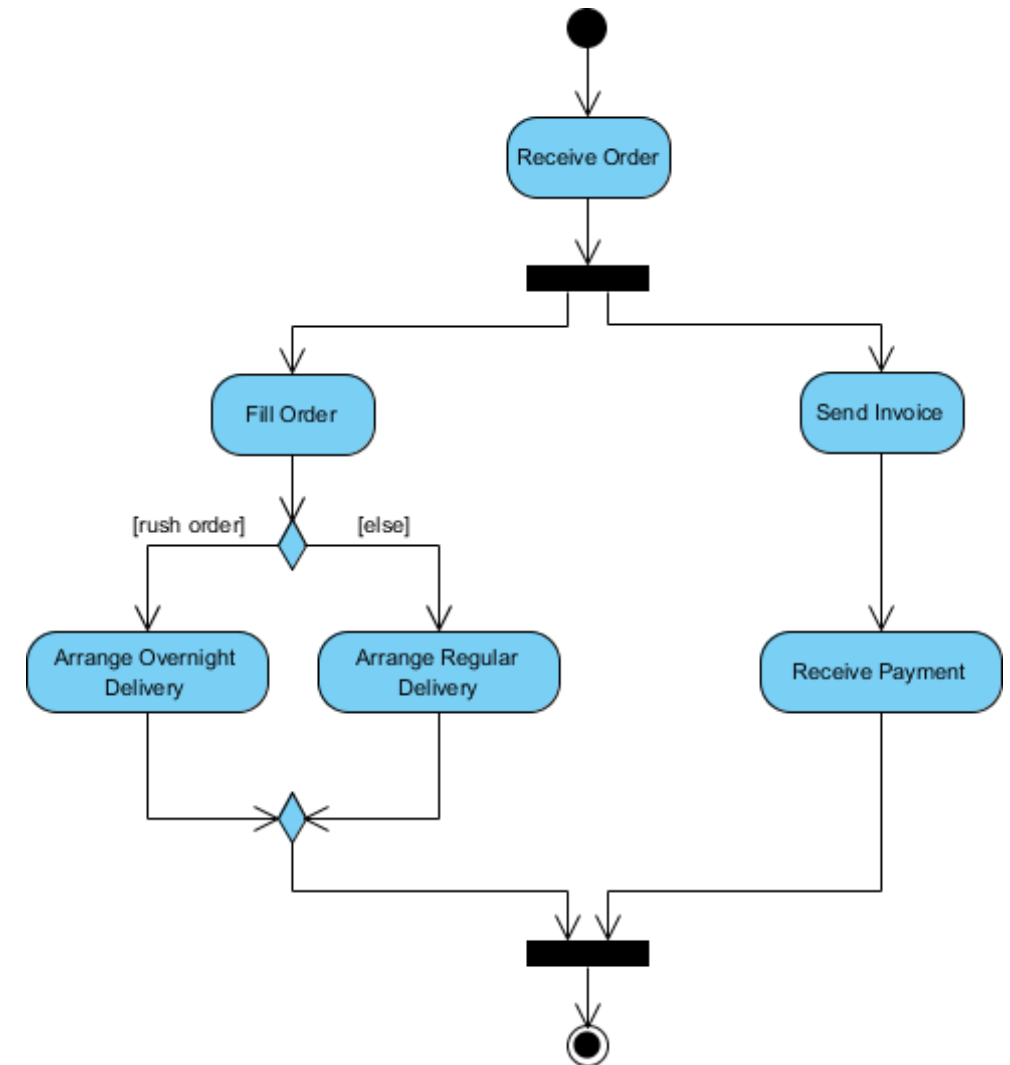


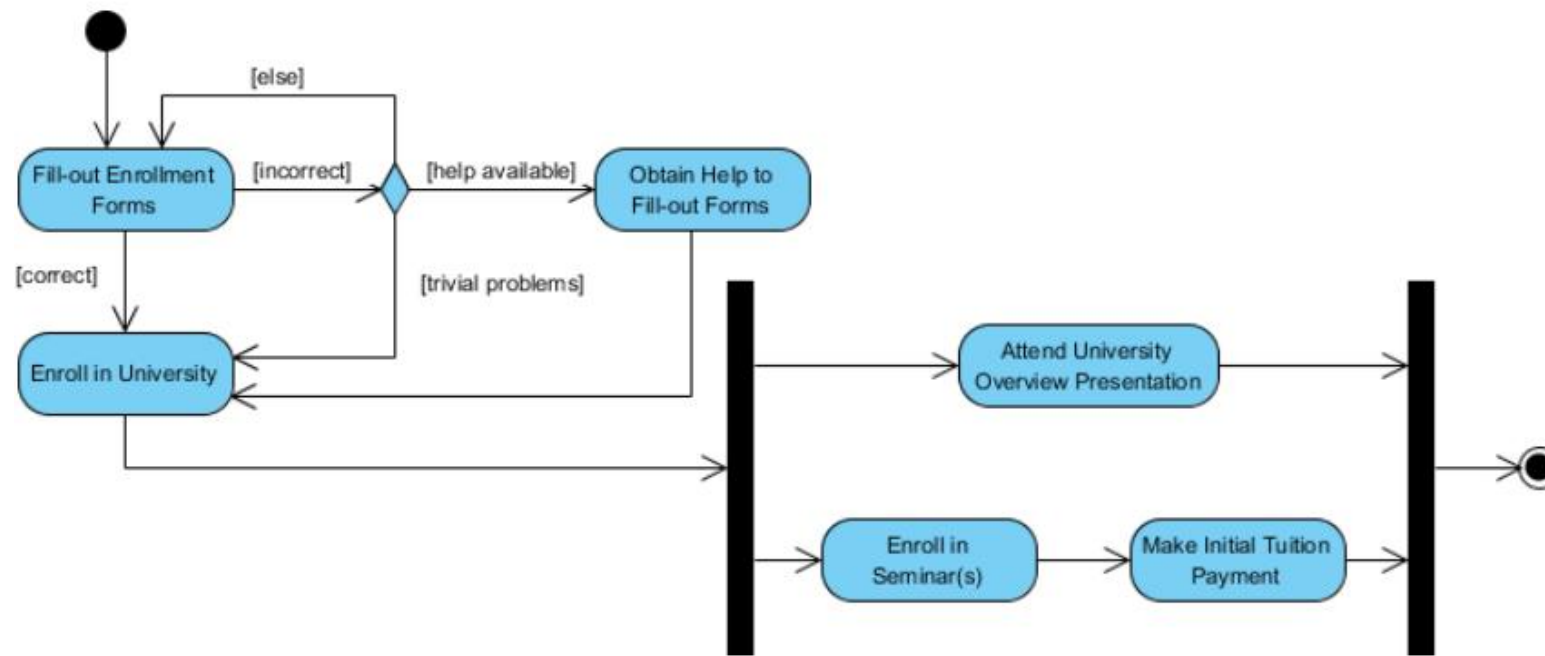


Examples of Activity Diagrams

### Process Order - Problem Description

- Once the order is received, the activities split into two parallel sets of activities.
- One side fills and sends the order while the other handles the billing.
- On the Fill Order side, the method of delivery is decided conditionally.
- Depending on the condition either the Overnight Delivery activity or the Regular Delivery activity is performed.
- Finally the parallel activities combine to close the order.





- An applicant wants to enroll in the university.
- The applicant hands a filled out copy of Enrollment Form.
- The registrar inspects the forms.
- The registrar determines that the forms have been filled out properly.
- The registrar informs student to attend in university overview presentation.
- The registrar helps the student to enroll in seminars.
- The registrar asks the student to pay for the initial tuition.

# Sequence Diagrams

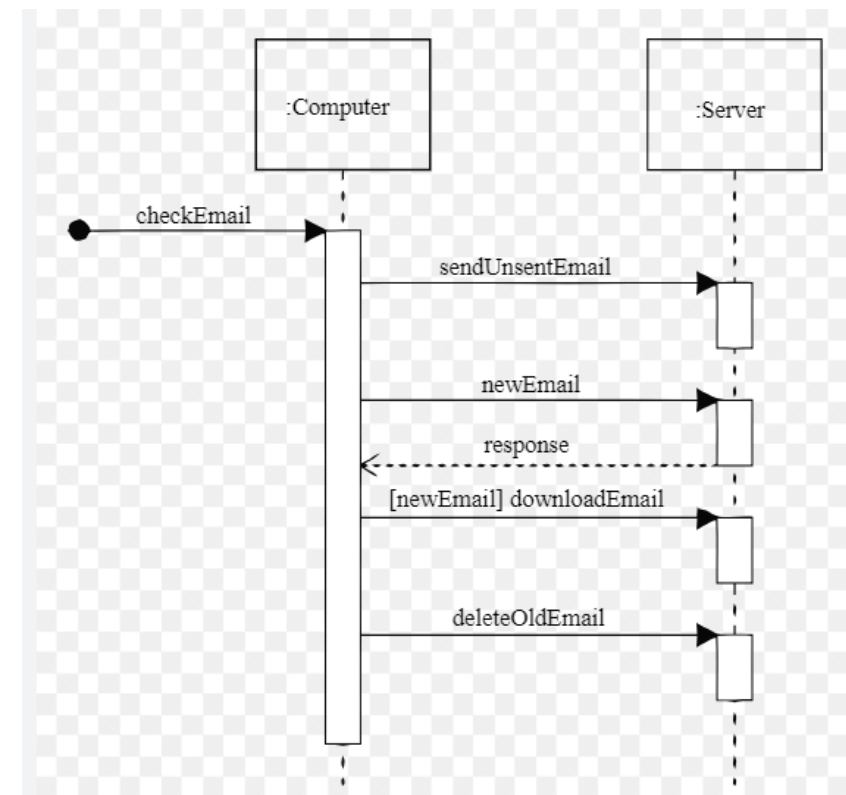
---

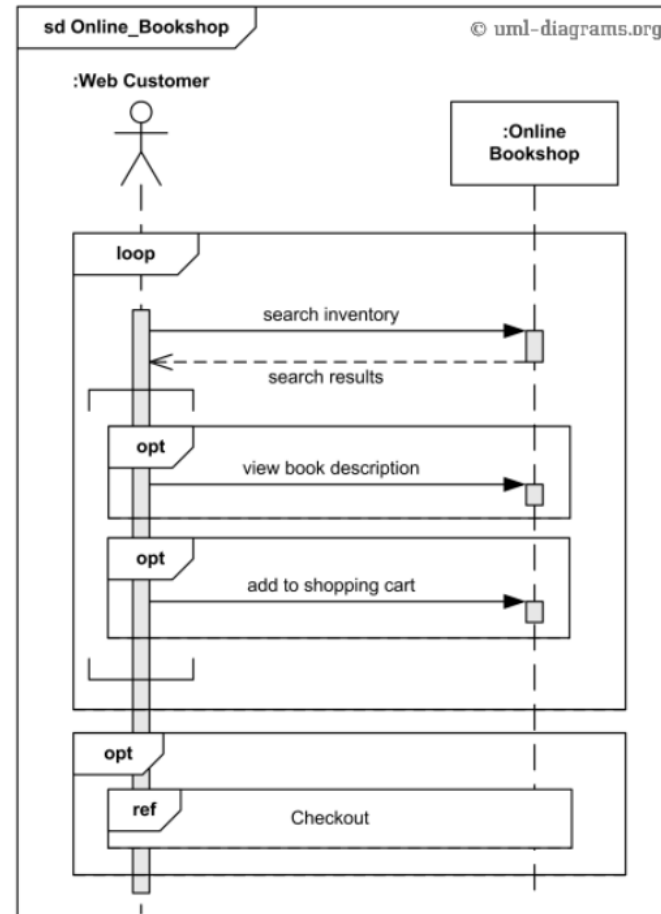
- A sequence diagram is a **Unified Modeling Language (UML) diagram** that illustrates the sequence of messages between objects in an **an interaction**.
- A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

# Sequence Diagrams

- A sequence diagram consists of:
  - Actors (on top)
  - Actions listed in sequence from top to bottom.
  - A solid vertical line is a process execution.
- More Information:
  - <https://www.uml-diagrams.org/sequence-diagrams.html>

## Components of a Sequence Diagram, Illustrated





*An example of UML sequence diagram for online bookshop.*

## Sequence Diagram, Example 2

# User Interfaces for IoT

IoT “UI’s”

# User Interfaces (for IoT)

---

- The information needs to be available to the end-user in some way.
  - Triggering alarms on their phones
  - sending them notification through email or text message.
- The user sometimes might need an interface which actively checks their IOT system.
  - The user has a camera installed in his home. He wants to access video recording and all the feeds with the help of a web server.

# User Interfaces (for IoT)

---

- Depending on the IoT application and complexity of the system, the user may also be able to perform an action which may create cascading effects.
  - if a user detects any changes in the temperature of the refrigerator, with the help of IOT technology the user should be able to adjust the temperature with the help of their mobile phone.

# User Interfaces (for IoT)

---

## ▣ Voice based.

- ▣ Eliminate the need for applications (apps).
- ▣ Focused on AI and deep learning.

## ▣ Edge-Computing

- ▣ IOT devices are small, they can not process the commands you give it.
- ▣ They send the commands to a server (think Google assistant) and the server processes your command(s).

# User Interfaces (For IoT)

- Web Based Interfaces
  - Usually dashboards for displaying aggregate data (sums, totals, averages, alerts).
- This web interface reads values (temperature, humidity)
- This interface also allows the user to turn on IO ports (to control devices such as lights).

ESP8266 Demo - www.projetsdiy.fr - Mozilla Firefox

ESP8266 Demo - www.p... x +

192.168.1.21

Rechercher

## Demo Webserver ESP8266 + Bootstrap

Mini station météo

Température 23.90 Humidité 37.80 Pression atmosphérique 971.85

Capteur	Mesure	Valeur	Valeur précédente
DHT22	Température	23.90°C	-
DHT22	Humidité	37.80%	-
BMP180	Pression atmosphérique	971.85mbar	-

### GPIO

D5 OFF	ON	OFF
D6 On	ON	OFF
D7 OFF	ON	OFF
D8 OFF	ON	OFF

www.projetsdiy.fr

# Specialized Interfaces

## ▣ Nest Display

- ▣ Rotary control via console (built-in screen)
- ▣ Can interact via voice with Google assistant or Alexa
- ▣ Android application
- ▣ Web access (for user)
- ▣ Web access (for Google)



# Alerts & Notifications

---

- Active vs. Passive
  - Alert = Active, Dashboard = Passive
- Triggers
  - Alerts happen on triggers, such as, movement detected, water detected, etc...
- One trigger Many notifications
  - It's possible for one trigger to produce >1 notification. Example, a **text message** and also a **push notification** on your phone.
- Interesting Application : IFTTT (If this then that)
  - <https://www.youtube.com/watch?v=p5McvkJYL2s>