Software engineering and Project management-Lab

Submitted by: Jyotiraditya Singh

Enroll Number: R2142220530

SAP ID: 500107065

Experiment-1

Aim: Study of UML diagrams

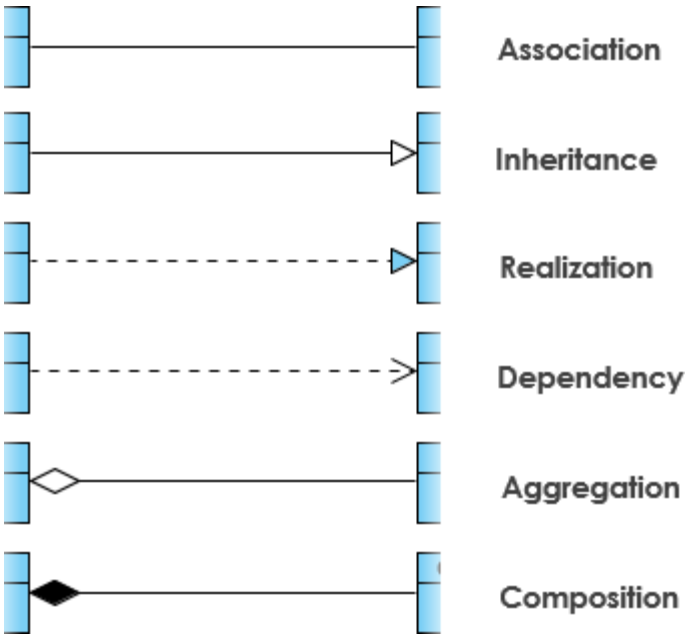
- A uml diagram is a way to visualize systems and software using Unified Modeling Language (UML).
- Software engineers create UML diagrams to understand the design, code architecture and proposed implementation of complex software.
- UML diagram simplifies the information about an application or software in a diagrammatic form that makes it easier for other team members to understand about the working of any software.
- UML diagrams can also be used in explaining the clients about the software they asked for.
- UML diagrams help in keeping track of relationships and hierarchies between important lines of code.

Types of UML diagrams

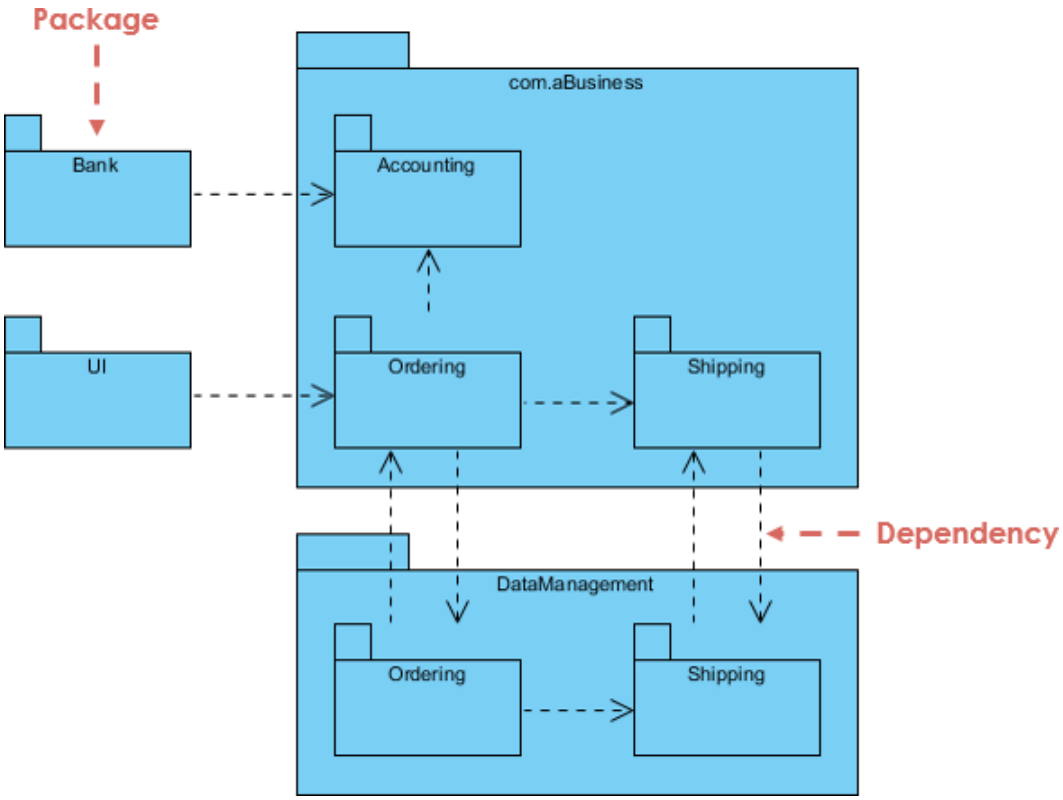
- There are two sub categories of UML diagrams:
  - Structural diagrams: depict the components that make up a system and the relationship between those two.
  - Behavioral diagrams: represent what happens within a system. They show how all the components interact with each other and with other systems or users.

- **Structural diagrams**
  - Class diagram: fundamental building block of any oop solution. It depicts a static, object-oriented system, defining projects by classes, attributes and functions.

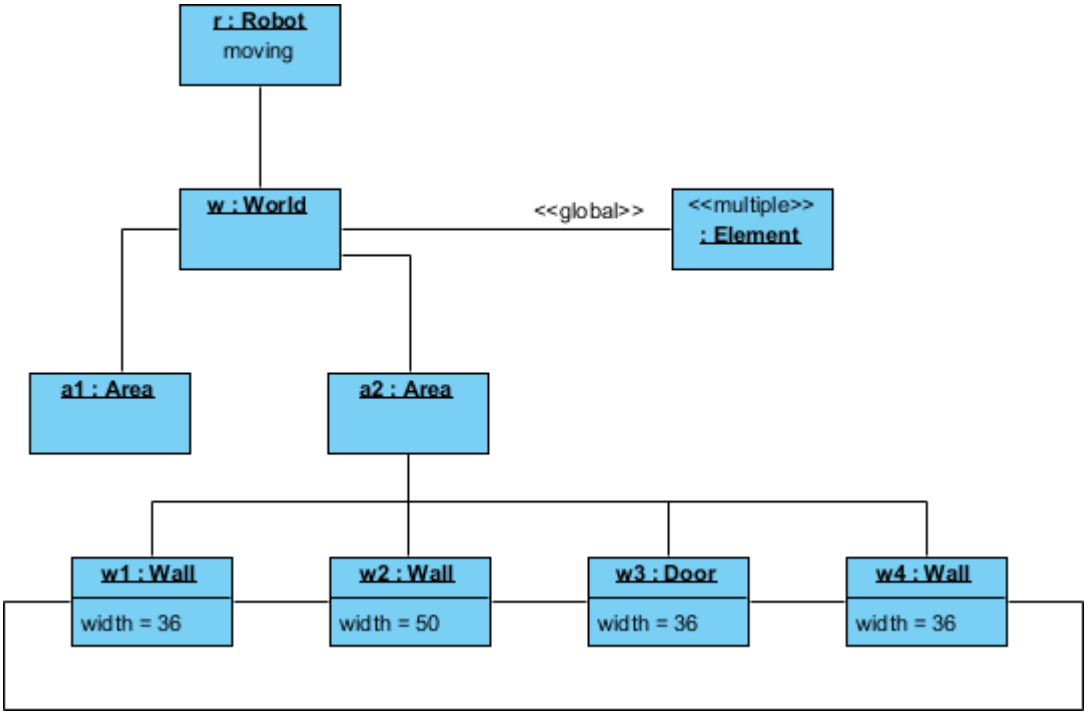
Relationships between classes are defined by:



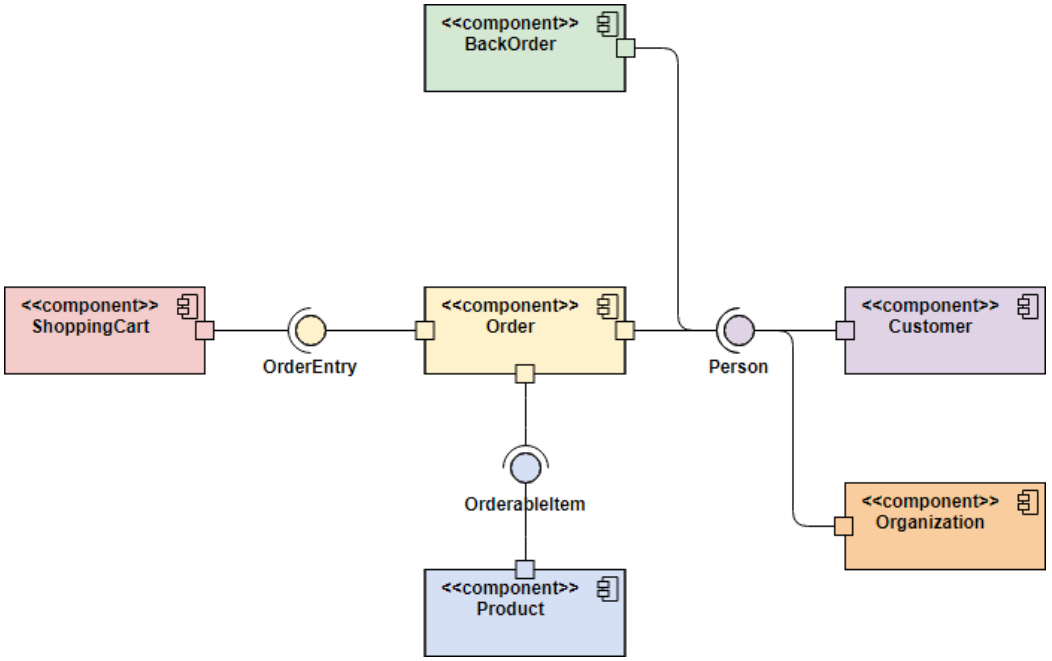
- Package diagram: It shows dependencies and relationships between packages in a system.



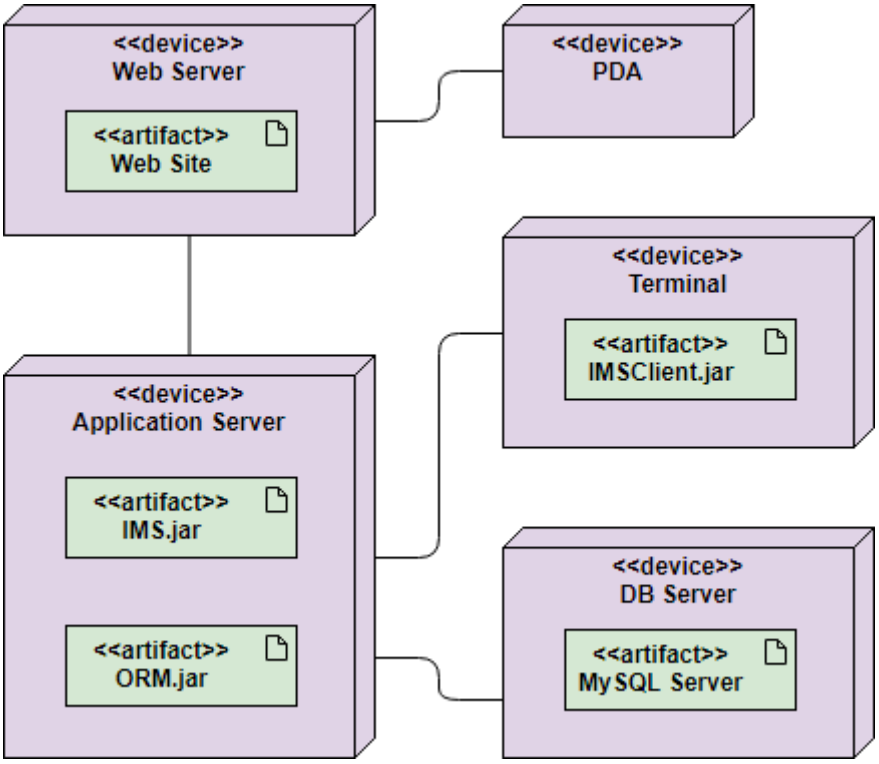
- Object diagram: Shows the attributes of different objects within a system. And like class diagrams they show the relationships between the objects in a piece of software.



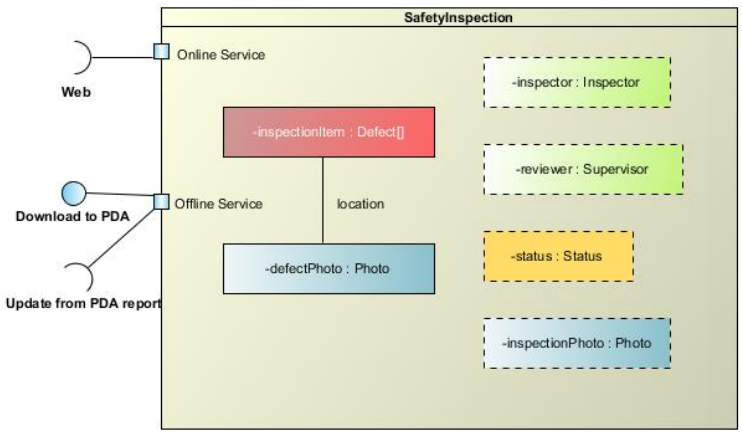
- Component diagram: This diagram breaks down a complex system into smaller components making it easier to visualize. Unlike other UML diagrams, it specifically outlines the relationship between these components.



- Deployment diagram: This shows the configuration of processing nodes and all the components that live on them. It depicts the physical arrangement of the nodes in a distributed system.

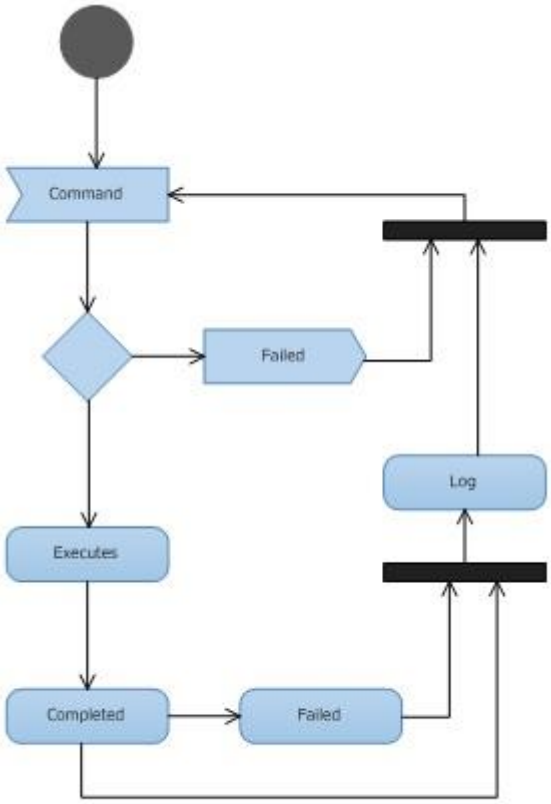


- Composite structure diagram: it shows the configuration of processing nodes and all the components that live on them. It depicts the physical arrangement of the nodes in a distributed system.

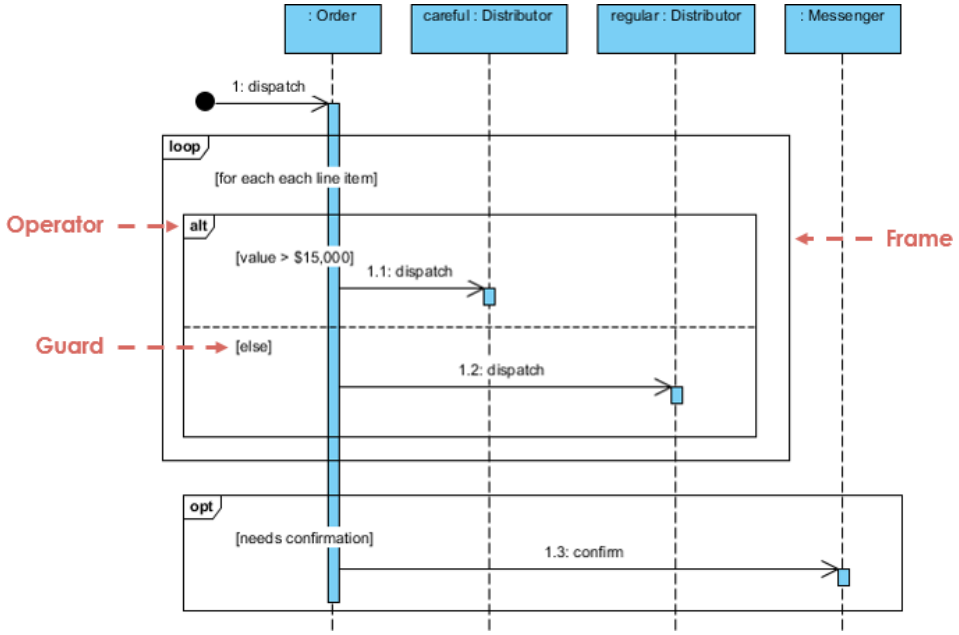


~ **Behavioral diagrams**

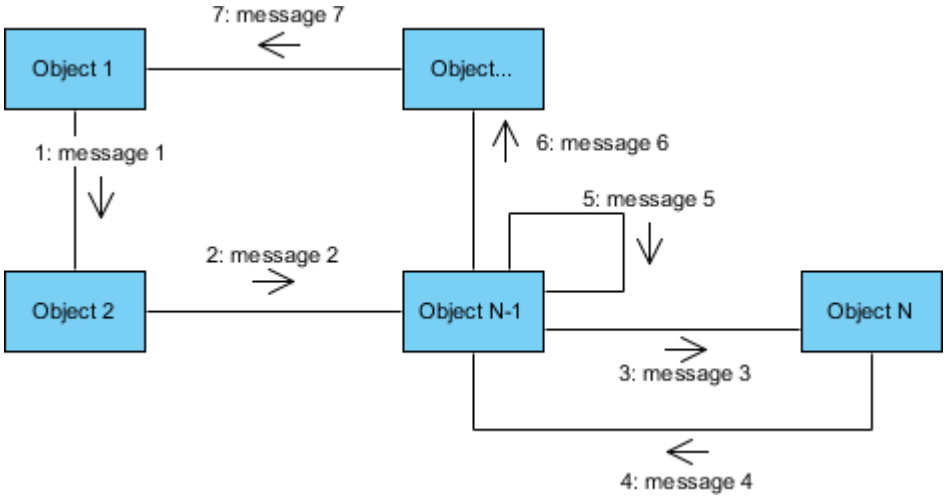
- Activity diagram: This diagram is a flowchart that outlines all the system's activities. It shows everything from start to finish.



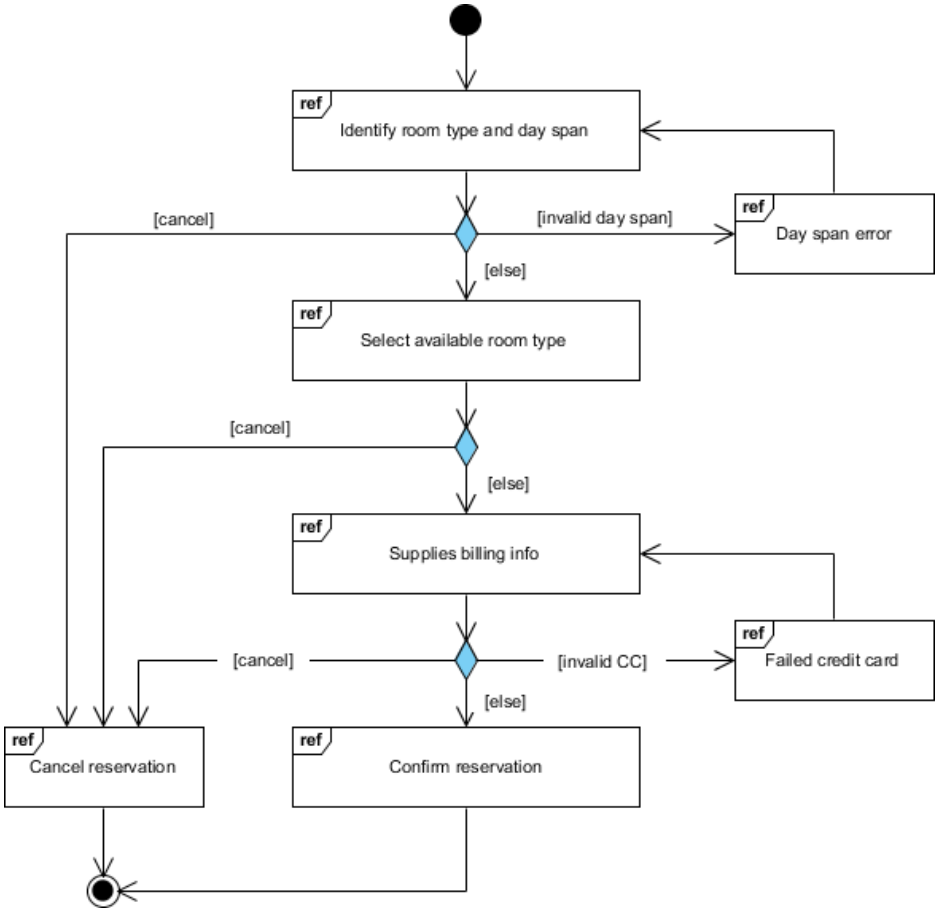
- Sequence diagram: This shows the order in which the objects interact.



- Communication diagram: These are also called collaboration diagrams. They are relatively similar to sequence diagrams, focusing on the messages passed between different objects.



- Interaction overview diagram: These diagrams visualize the flow of activity and the sequence of those activities.



- Timing diagram: These are useful for deployment, as they depict the behavior of specific objects within an explicit time frame. These diagrams are usually fairly straightforward, but when dealing with more than one object, they show the interactions between many different and important sequences within that time frame.
- Use case diagram: This diagram provide graphic overview of the actors involved in a software system. By illustrating system functionality and outlining a system's expected behavior, they help the developers in analyzing the relationships between use cases and personas.
- State machine diagram: Also knows as state chat diagrams, UML state machine diagram show the behaviors of different components in a system.

Benefits of UML Diagram:

Makes it easier to understand the code and complex ideas

It turns complex codes into a visual diagram

It keeps everyone in the team on the same page

Allows developers to see the bigger picture of a system

It helps non-programmers to understand the processes and functionalities of software

Uses a common notation so it means any programmer can understand the working of a project