**Lab. 02**

**Title: Working with Use Case Diagrams**

**Lab Objective: the objective of this lab is to learn**

• What is UML

* Why the UML is necessary
* How to represent UML
* Familiarize with the concept underlining Use Case Diagram for requirement analysis.
* Exposure on all elements grouped under Use Case Diagram.
* Analyze functional requirements and problem statements of an intended system to be able to translate into use case modeling.

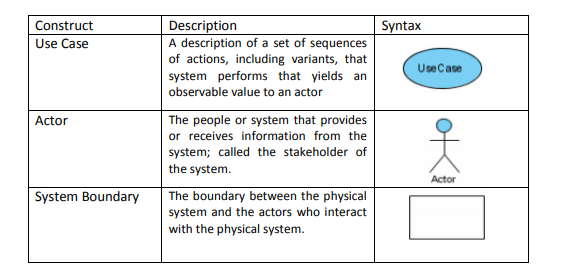
**Introduction:**

* The Unified Modeling Language (UML) is a visual modeling language that enables system builders to create blueprints that capture their visions in a standard, easy-to-understand way, and provides a mechanism to effectively share and communicate these visions with others.
* The purpose of UML diagrams is to present multiple views of a system; this set of multiple views is called a model. UML model describes what a system is supposed to do. It doesn’t tell how to implement the system
* A Use Case is a set of scenarios describing an interaction between a user and a system. It displays (in symbolic form) the relationship among actor and use cases besides these it also represents the system’s functionality and the requirements of the system from the user’s perspective.

**When to use Use-Cases:**

1. Use cases are used in almost every project
2. Very helpful in exposing requirements and planning of project
3. Usually done during the early stages of project

**UML Notation for Use Case Modeling:**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Association |  | The participation of an actor in a | | |  |
|  |  | use case. | | |  |
| Generalization |  | A relationship between a general | | |  |
|  |  | use case and a more specific use | | |  |
|  |  | case. | | |  |
|  |  |  | | |  |
| Extend |  | Relationship between the extension | | |  |
|  |  | use case and the base use case. | | |  |
|  |  | Specify how the behavior of the | |  |  |
|  |  | extension use case can be inserted | |  |  |
|  |  | into the behavior defined for the | |  |  |
|  |  | base use case. |  | |  |
| Include |  | Relationship between a base use | | |  |
|  |  | case and an inclusion use case, | | |  |
|  |  | specifying how the behaviors of the | |  |  |
|  |  | inclusion use case can be inserted | |  |  |
|  |  | into the behavior defined for the | |  |  |
|  |  | base use case. |  | |  |

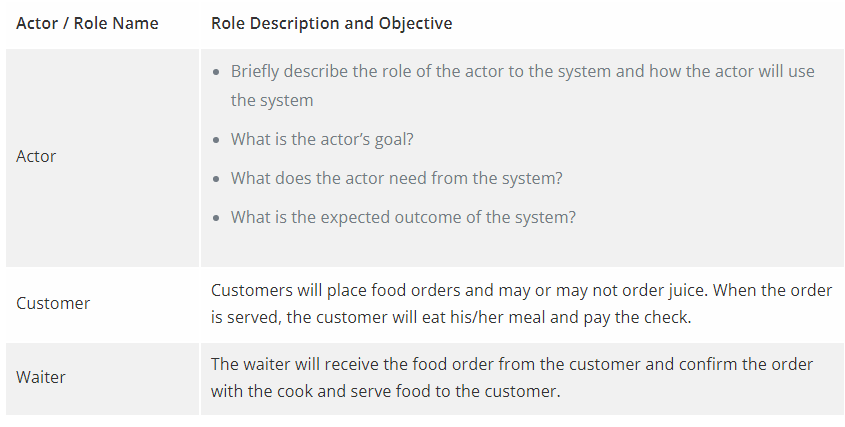


**Actors:**

Actors begin with noun

* Could be human beings or other system
* 2 types of actor classification:
  + **Primary Actors:** Actors that stimulate the system and the initiator of events.
  + **Secondary Actors:** Actors that only receive stimuli from the system.
  + **E.G For a criminal record system made for police****Primary actors:** The Employees who use the system , **Secondary or supporting actor**: The nadra system which may give input
* Actor Designing Consideration:
  + Who / what will be interested in the system
  + Who / what will want to change the data in the system
  + Who / what will want to interface with the system
  + Who / what will want information from the system

**Actor goal list**



**Use Case:**

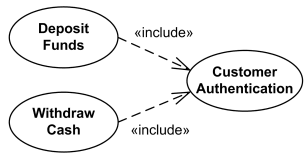
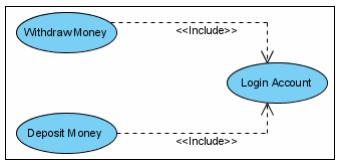
* Use case should ideally begin with a verb.

**Actor / Use Case Generalization:**

* Used when you find 2 or more use cases that have commonalities in behavior, structure and purpose.
* A parent use case may be specialized into 1 or more child use cases that represent more specific form of the parent.
* **For example**, suppose the ATM system can be used to pay bills. Pay bills has two child use cases: Pay Credit Card Bill and Pay Utility Bill

**Include Relationship:**

* Used when 2 or more use cases share some common portion in a flow of events.
* The common portion is grouped and extracted to form an inclusion use case to be shared among the use cases related
* Specifies that the source use case explicitly incorporates the behavior of another use case at a location specified by the source
* **For example, in the ATM system example, such as Withdraw Money, Deposit Money or Check Balance, all share the inclusion use case Login Account**



**Extend Relationship:**

* Extend relationship is used when 2 use cases are similar, but one does a bit more than the other
* A base case under extend relationship may develop an alternative flow invoking an alternative use case
* The extension use case adds extra behavior to the base use case



**Example 1**

**A user clicks the search button on an application’s user interface. The application sends an SQL query to a database system. The database system responds with a result set. The application formats and displays the result set to the user**

**In this scenario:**

* The user is a **primary acto**r because he initiates the interaction with the system (application).
* The database system is a **secondary actor** because the application initiates the interaction by sending an SQL query.

**Case study example 2**

Computerized application used to record sales and handle payments and is Used in retail store

It includes hardware and software It also interfaces to other applications, such as a third-party tax calculator and inventory control and system should work even if access to external services is down (i.e. at least allow checkout with cash if credit card processing software goes down) .Needs to support Multiple clients-side interfaces i.e. web browser , touch screen and wireless phone etc Commercial POS

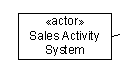
**Step 1**

**Actor goal list**

|  |  |
| --- | --- |
| Actor | goal |
| Cashier ,customer | Process sales, handle returns, cash in cash out |
| Manager | Start up, shut down ,handle returns |
| System administrator | Add users  Modify users  Delete users  Manage security  Manage system tables |
| Sales activity system | Analyze sales and performance data |

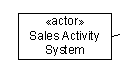
**Step 2**

**Use case diagram**



<actor>

Sales Activity System



**Question 1**

Do a comparative analysis of UML tools

**Question 2**

Which diagrams comes under static and dynamic UML diagrams also give reasoning

**Question 3**

**Consider following scenario**

At the beginning of each semester students may request a course catalogue from registrar containing a list of course offering for the semester. Information about each course, such as professor, department and prerequisites will be maintained by registrar to help students make informed decisions. The new on-line registration system will allow students to perform registration and select four course offerings for the coming semester. In addition, each student will indicate two alternative choices in case a course offering becomes filled or canceled. No course offering will have more than ten students. No course offering will have fewer than three students. A course offering with fewer than three students will be canceled. Once the registration process is completed for a student, the registration system sends information to the billing system, so the student can be billed for the semester. Professors must be able to access the on-line system to indicate which courses they will be teaching. Professors will also need to view course offering roster to see which students signed up for their course offering. For each semester, there is a period of time that students can change their schedules. Students must be able to access the on-line system during this time to add or drop courses. The billing system will credit all students for courses dropped during this period of time. The registrar would also maintain course curriculum and update it in the system

**Lab Task:**

Q1 Identify primary and secondary Actors of the system

Q2 Create actor goal list for above scenario

Q2 Identify Use Cases that appears as verbs.

Q3 Use Include and Extend relationship where needed

Q4 Draw a complete Use Case diagram of the system