NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

Object Oriented Analysis & Design

Muhammad Nadeem || Awais Ahmed || Tooba Ali

Muhammad.nadeem@nu.edu.pk || awais.ahmed@nu.edu.pk || tooba.ali@nu.edu.pk

Lab Session # 03

Objectives:

- **To Understand Use Case Diagram**

Basic Definition

A use case is a description of how a person who actually uses that process or system will accomplish a goal.

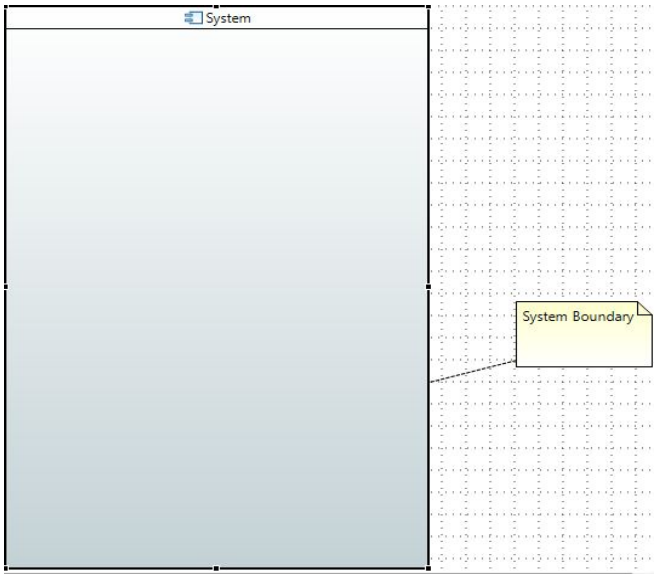
Basic Elements Of A Use Case

The basic elements that make up a use case are:

- System for which we are designing the use case for
- Actors: users who will interact with the system
- Use cases: Basic flow of what the system does
- Associations between actors and use case

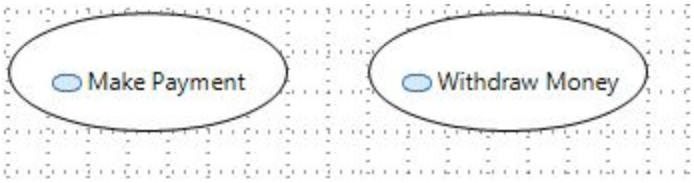
System:

System denotes the software or application for which you are designing the use case for. It is denoted by rectangular box with the system's name inside it. The boundaries of the rectangular box is called system boundary.



Use Cases:

The use cases are the discrete business tasks that are performed by the system, denoted by an ellipse shape. A use case may by connected to multiple actors.



Actors:

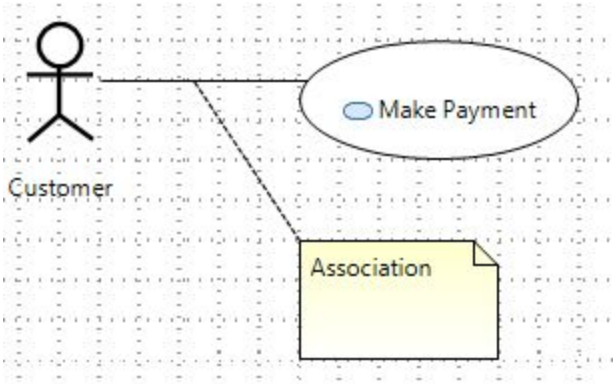
Actors are the users who will use the system to achieve certain goals. They are represented by stick man in the use case diagram.

Relationships in Use Case Diagram:

A relationship between two use cases is basically a link or dependency between two use cases or a use case and an actor. Use cases have 4 different kinds of relationships.

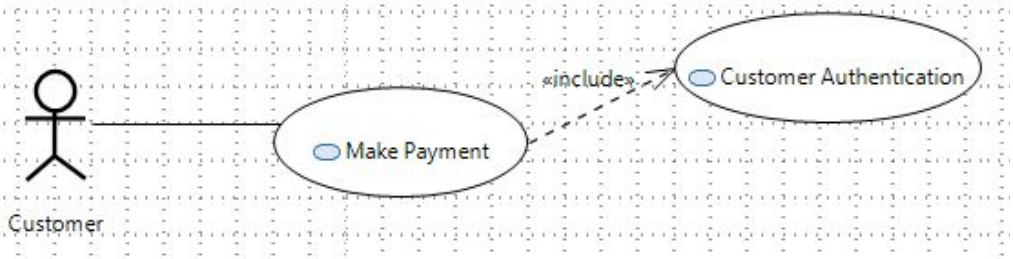
Association:

Association is a simple link between actor and a use case to show connection between them.



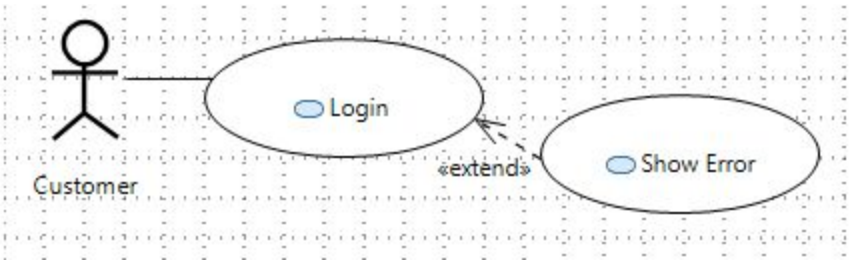
Include:

Include relationship shows relationship between base use case and included use case. We use include use case when the base use case requires the included use case in order to be complete. It is represented by a dotted line going towards include use case with include written in angular brackets.



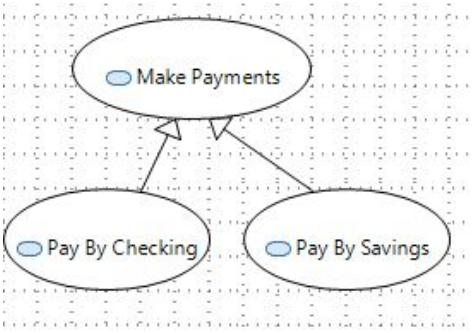
Extends:

In the extends relationship, the child use case adds to the functionality of another parent use case and may be executed in special conditions but not always. It is represented by a dotted line going towards the parent class with extend in angular brackets.



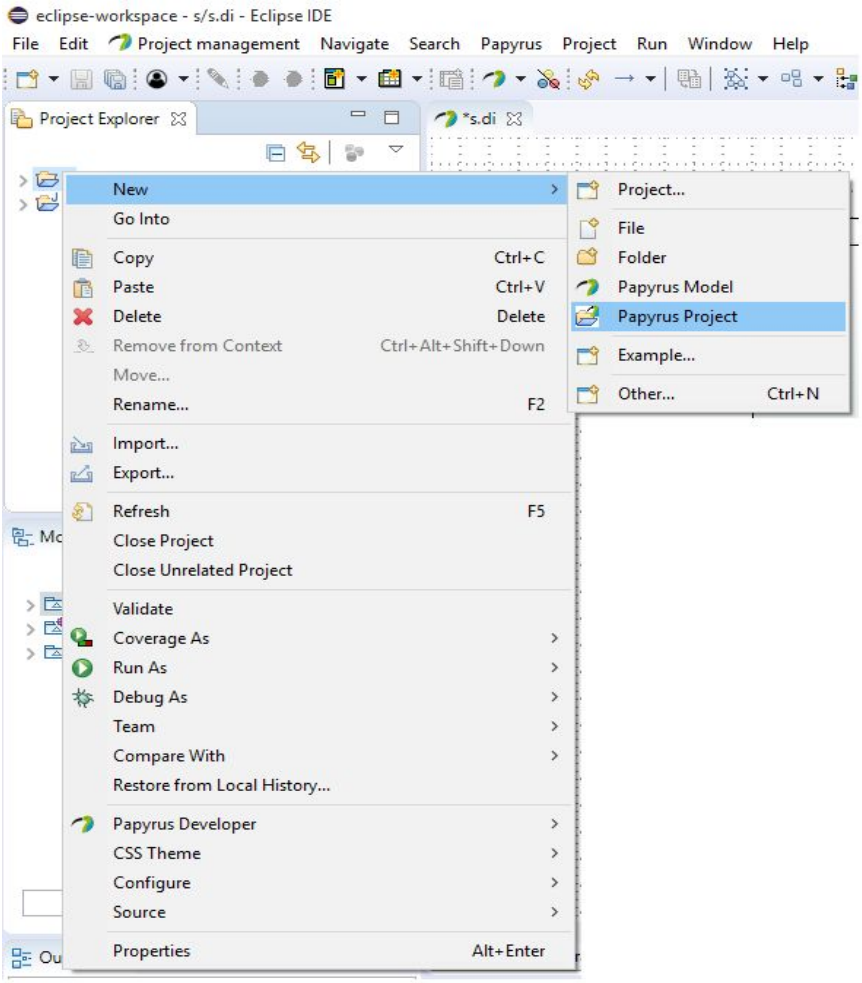
Generalization:

A generalization relationship is a relationship in which one model element (the child) is based on another model element (the parent).The model elements in a generalization relationship must be the same type. For example, a generalization relationship can be used between actors or between use cases; however, it cannot be used between an actor and a use case. Generalization is shown as a directed arrow with a triangle arrowhead.

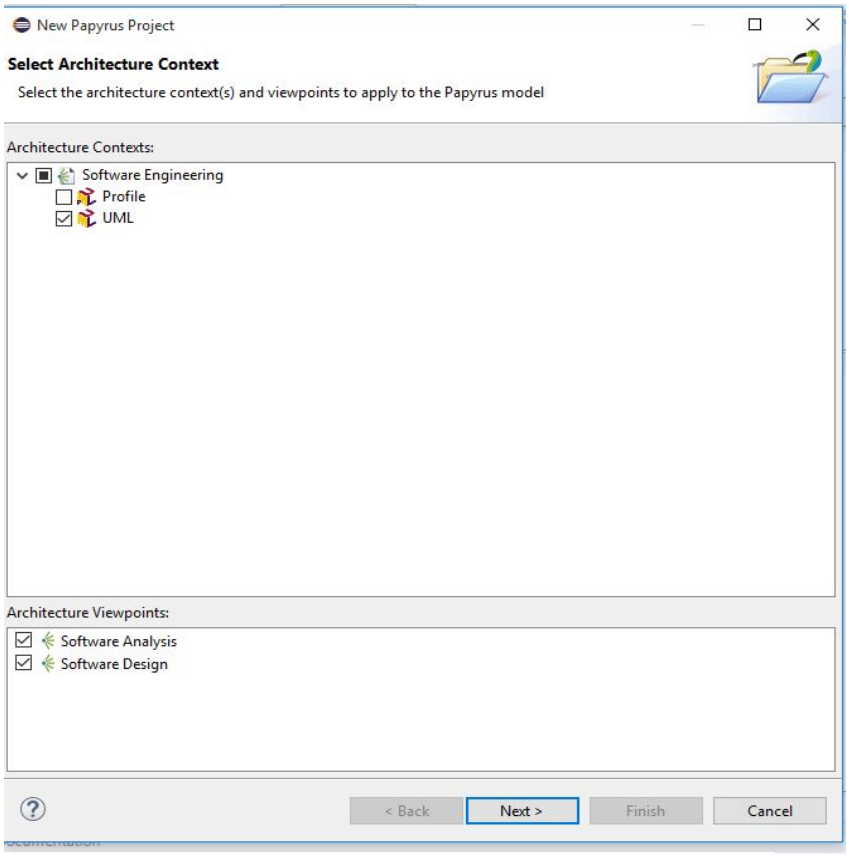


Use Case Diagram Tutorial In Papyrus

- To open create a use case diagram in papyrus, first create a empty java project.
- Right click on the java project and select New-> Project -> Papyrus project



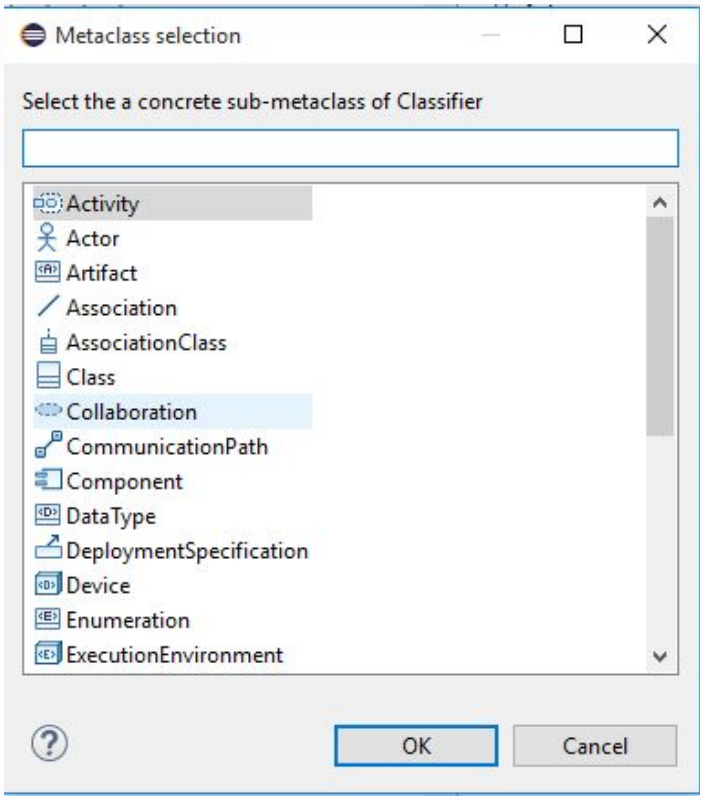
- In select architecture context window, select UML and click Next.



- Give the project a name in next and in the initialization information window, check on use case diagram and click finish.

- Once this is done, you can now create a use case diagram. A file called "model.di" will now be open and ready for editing under the Model Explorer. You may find elements relevant to you within the Palette window. Use Cases, Actors, and Subject Area (which will be used for the system) are all contained within the Nodes category and the arrows used to relate these elements are contained within the Links category.

Note: For system boundary, select subject area and once in the metaclass selection window, select component and click ok.


Metaclass selection
Select the a concrete sub-metaclass of Classifier
Activity
Actor
Artifact
Association
AssociationClass
Class
Collaboration
CommunicationPath
Component
DataType
DeploymentSpecification
Device
Enumeration
ExecutionEnvironment
OK
Cancel