Object Oriented Analysis and Java

Nga Nguyen, nga.nguyen@eisti.eu, TG316
Florentin Bekier, florentin.bekier@eisti.eu

Course Schedule

Monday	Wednesday	Friday
10h40-12h10	10h40-12h40	9h-11h
Object Oriented Analysis (UML)	Software Engineering Project	Object Oriented Programming (Java)
Nga Nguyen	Nga Nguyen / Florentin Bekier	Florentin Bekier
Paper exam	Project (defense, report)	Programming exam
	Start date : 16th October	
	6 groups of 3 students	

Object Oriented Analysis

- Course 1 :
  - Introduction
  - Use Case Diagram

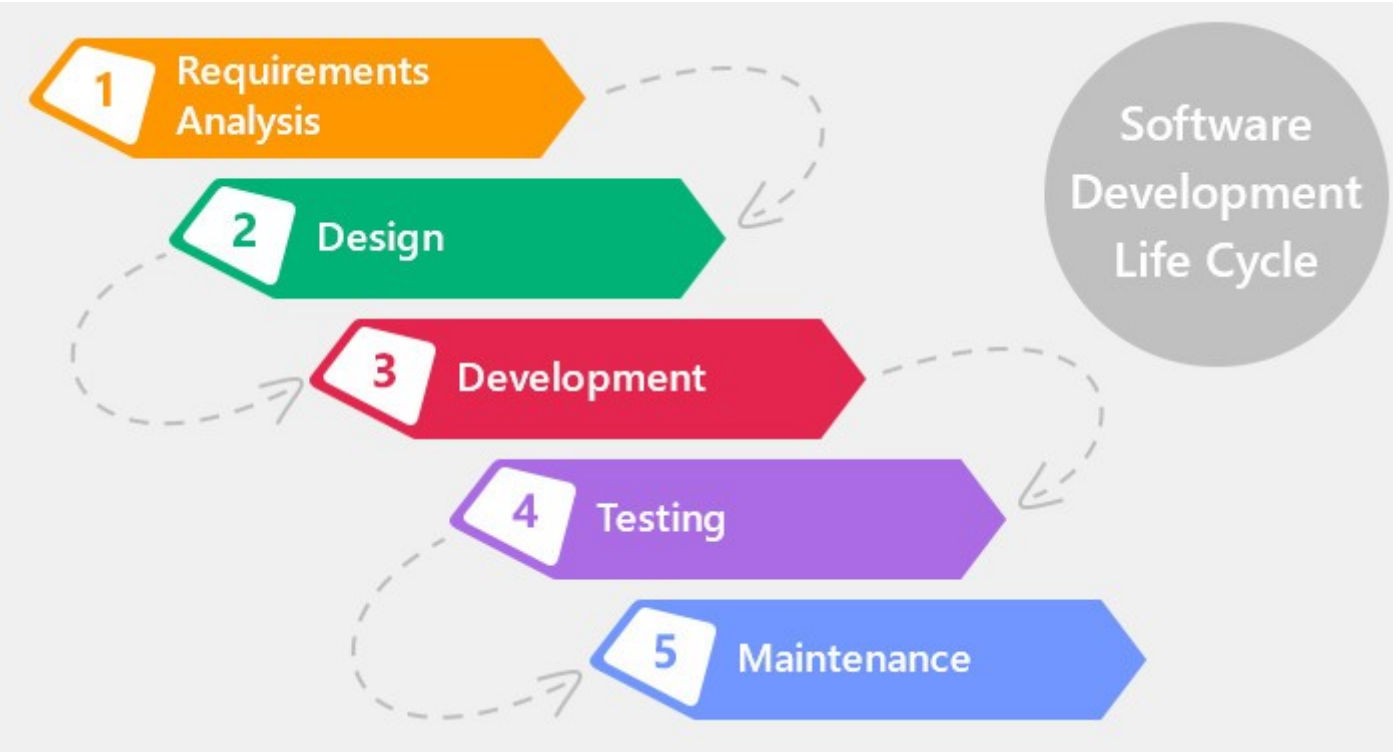
Introduction

- What are the Software Development Life Cycle phases?

Introduction

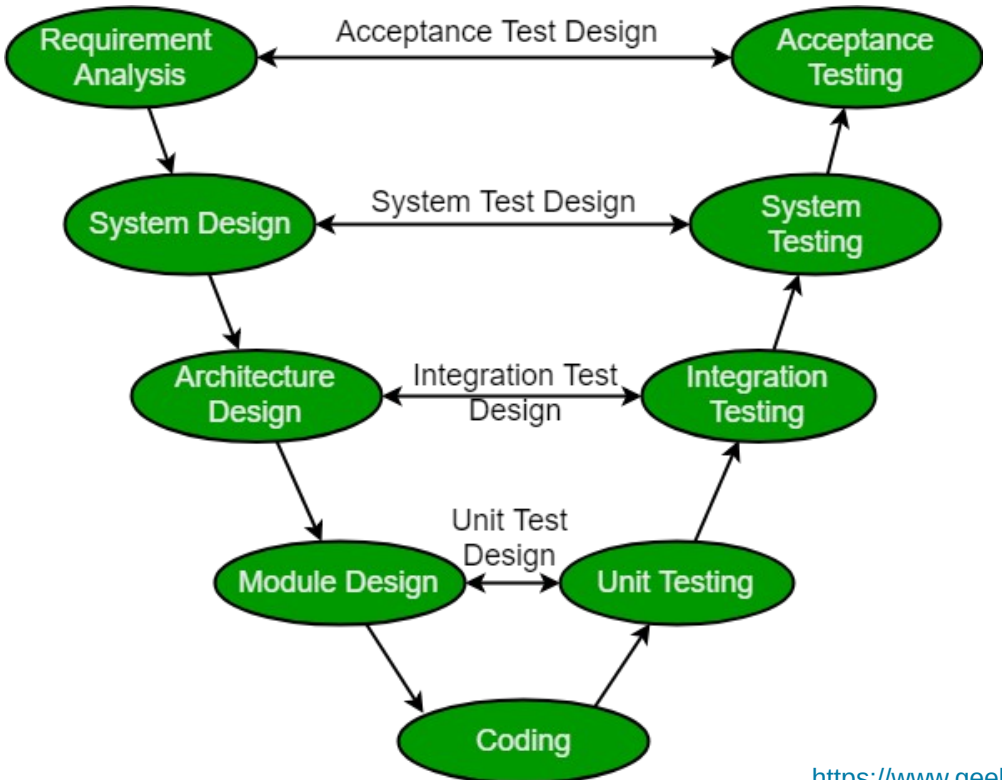
- What are the Software Development Life Cycle phases?
- Methodologies : Agile, Waterfall, Spiral, V Model, ...

Introduction



https://www.linkedin.com/pulse/what-software-development-life-cycle-sdlc-phases-private-limited/

Introduction



https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

Main steps before coding

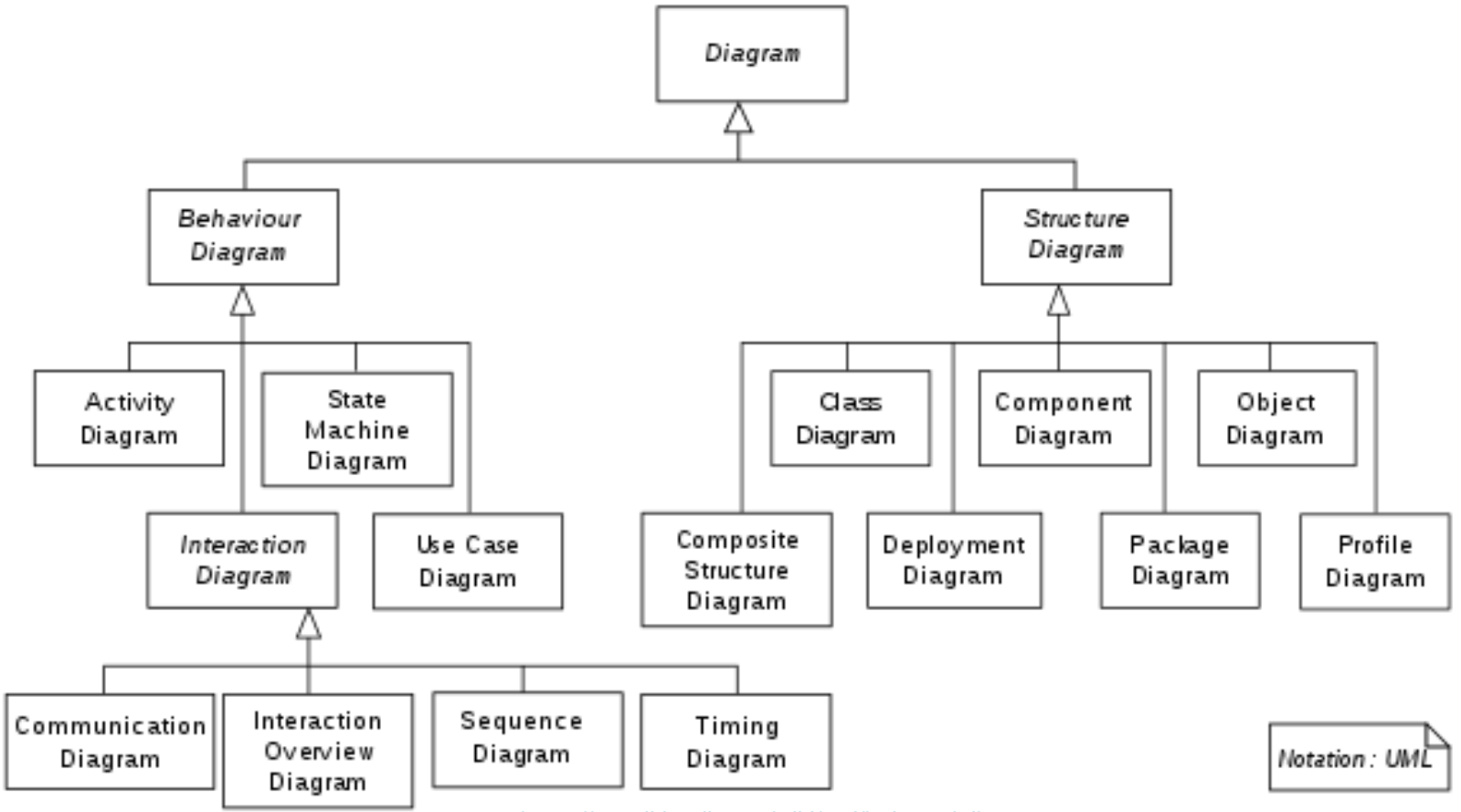
- Analysis :
  - WHO (actors) do WHAT (system) ?
- Design :
  - HOW ?

- Different approaches :
  - Structured approach vs Object-Oriented Approach

Unified Modeling Language (UML)

- Object Management Group (OMG) standard
- Fusion of 3 object-oriented modeling langages (Booch, OMT, OOSE) in 1995
- Latest version : UML 2.6.1 in 2017
- https://www.omg.org/spec/UML/About-UML/ : a specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems

Unified Modeling Language (UML)



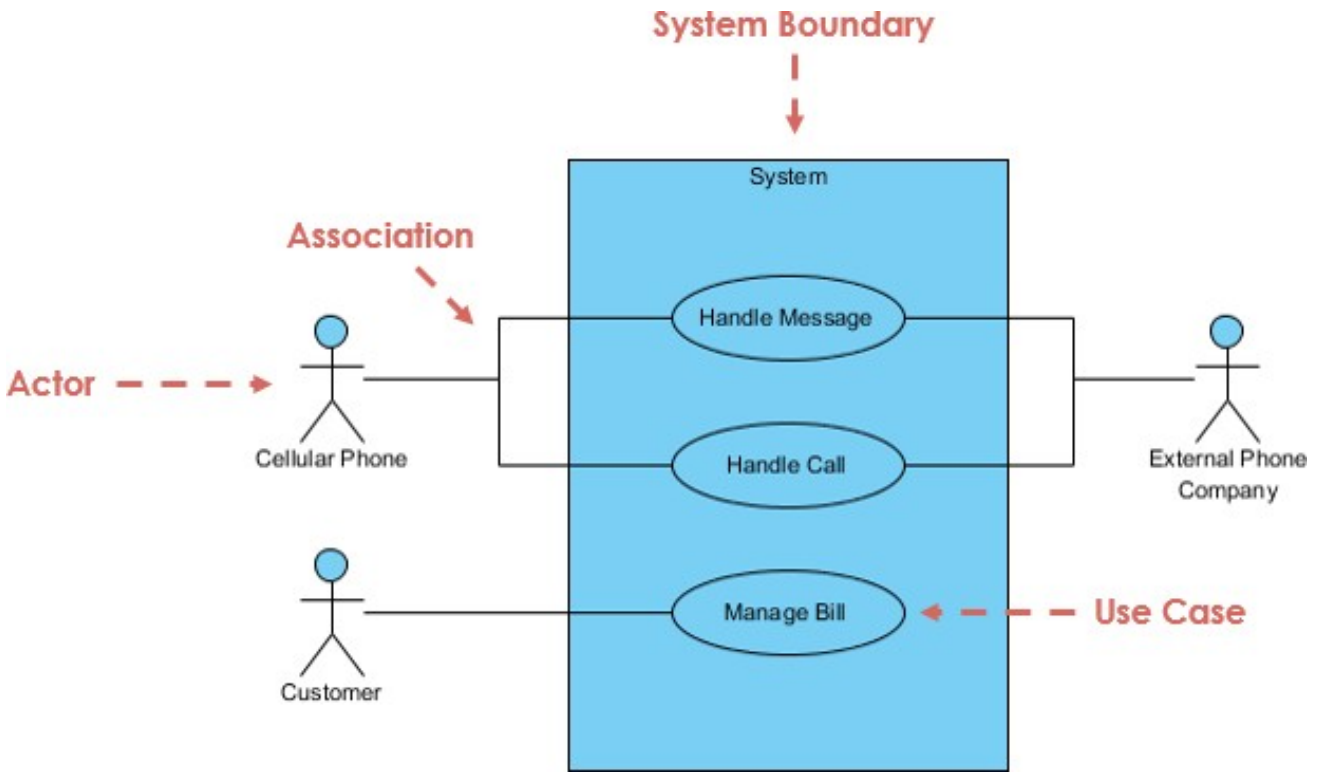
https://en.wikipedia.org/wiki/Unified_Modeling_Language

Analysis

- The system ?
- The actors who interact with the system ?
- The actions of these actors ?

=> USE CASE DIAGRAM

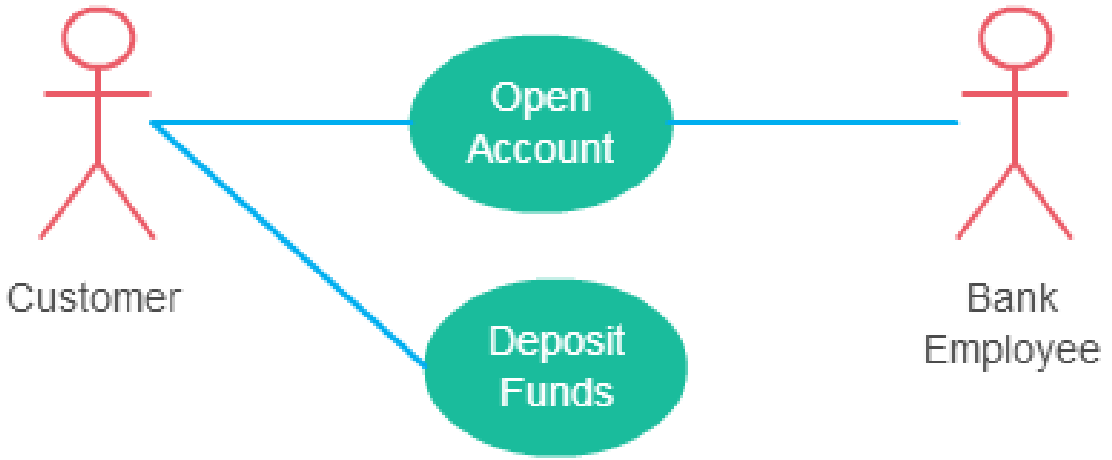
Use Case Diagram Elements



https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

Actors

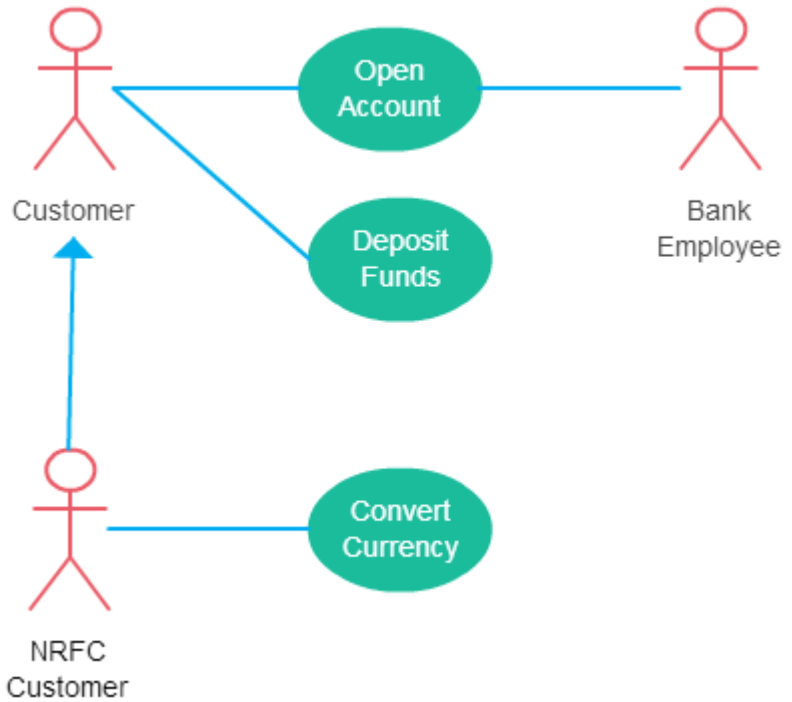
- An actor is an external entity (person, printer, server, database, …) that interacts with the system
  - Primary actor : use the system to achieve a goal
  - Secondary actor : the system needs assistance from to achieve the primary actor’s goal.



https://creately.com/blog/diagrams/use-case-diagram-relationships/

Relationship between Actors : Generalization

- The descendant inherits all the use cases of the ancestor.
- The descendant has one or more use cases that are specific to that role.



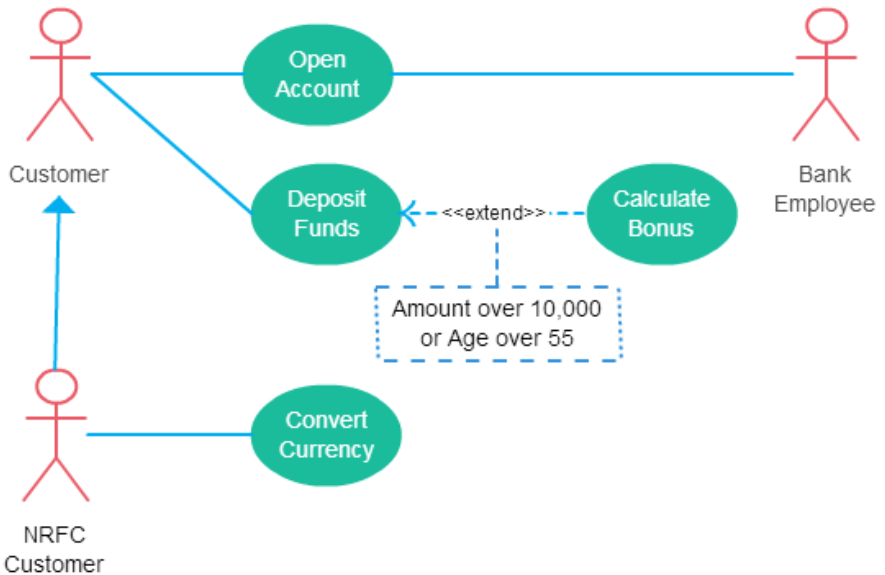
https://creately.com/blog/diagrams/use-case-diagram-relationships/

Relationship between Use Cases

- <<include>>
- <<extend>>
- generalization

Extend Relationship between Two Use Cases

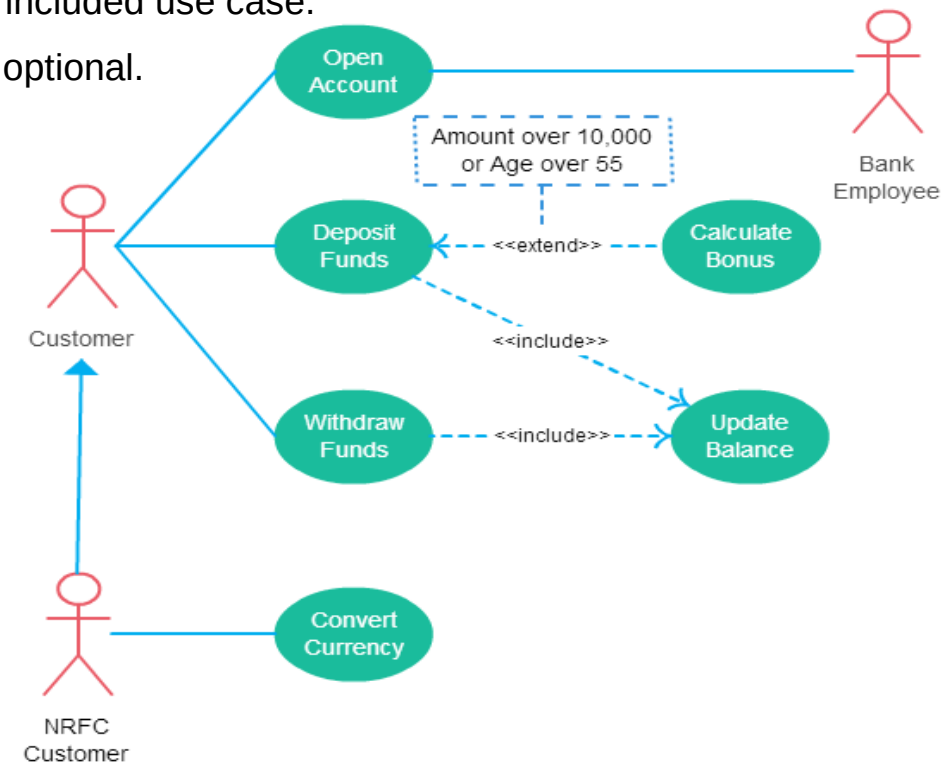
- Extend the base use case and add more functionality to the system.
  - The extending use case is dependent on the extended (base) use case.
  - The extending use case is usually optional and can be triggered conditionally.
  - The extended (base) use case must be meaningful on its own.



https://creately.com/blog/diagrams/use-case-diagram-relationships/

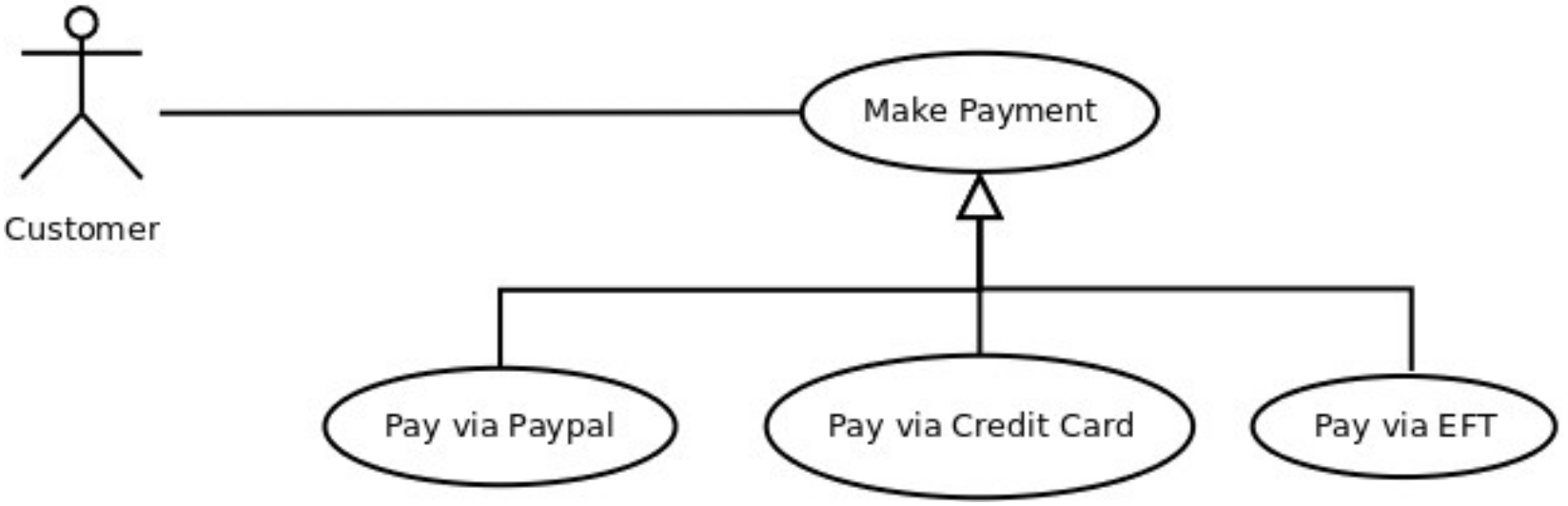
Include Relationship between Two Use Cases

- Reuse common actions across multiple use cases. Decompose complex behavior.
  - The base use case is incomplete without the included use case.
  - The included use case is mandatory and not optional.



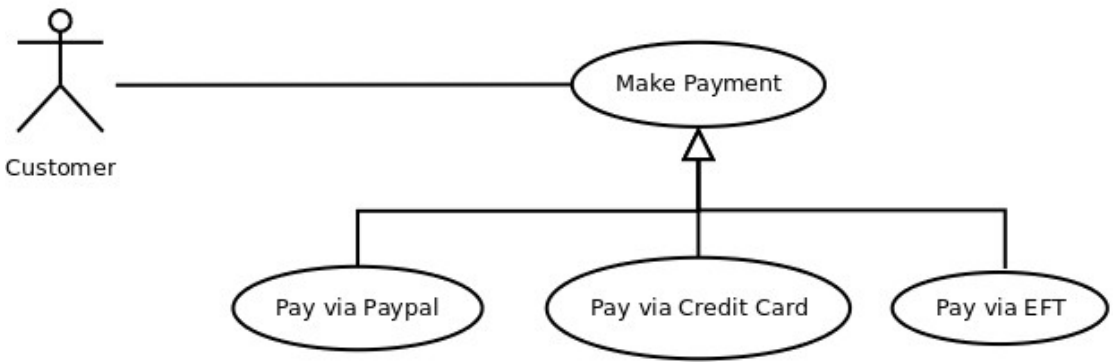
https://creately.com/blog/diagrams/use-case-diagram-relationships/

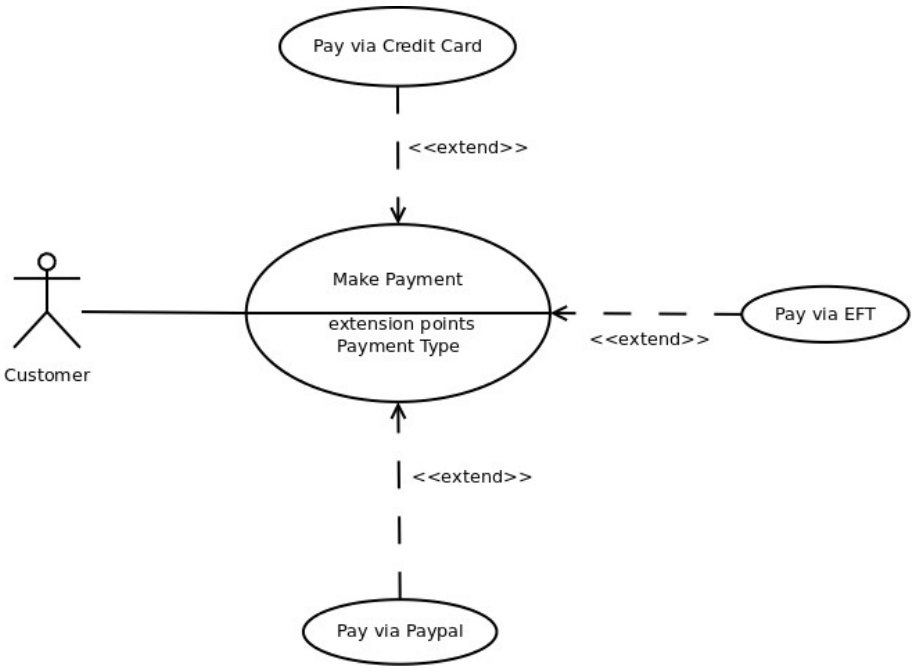
Generalization of a Use Case



https://stackoverflow.com/questions/15133595/use-case-generalization-versus-extension

What are the differences ?





https://stackoverflow.com/questions/15133595/use-case-generalization-versus-extension

Conclusions

- Identify the actors
- Identify the use cases
  - Name use cases with verbs
  - 5-7 rule
- Use different diagrams for the same project :
  - Decompostion
  - Level of abstraction
- NO TEMPORAL ASPECT IN A USE CASE DIAGRAM