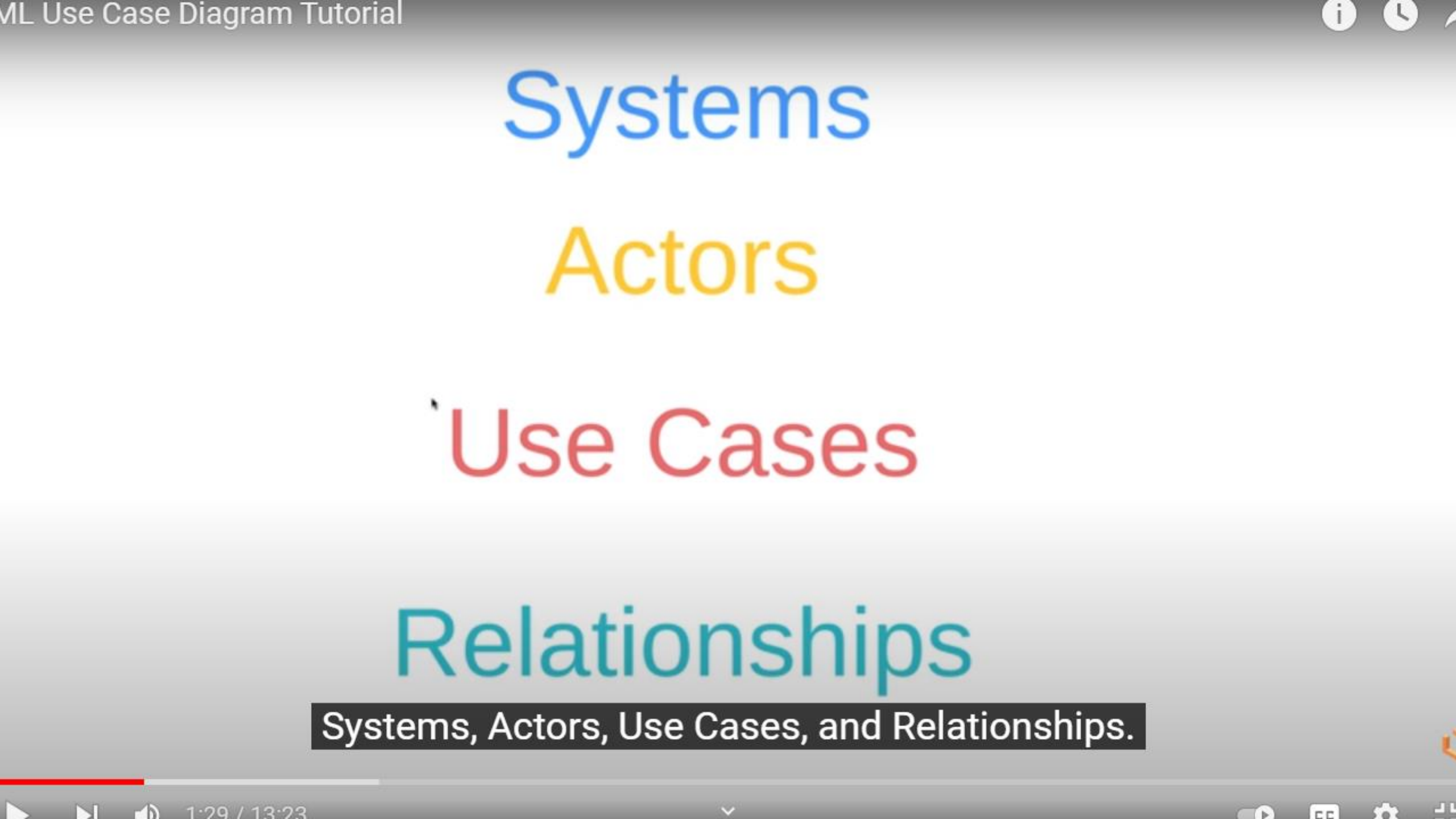
Use Case Diagram

ML Use Case Diagram Tutorial



Systems, Actors, Use Cases, and Relationships.

1:29 / 13:23

Banking App

Customer

Bank

Primary Actors

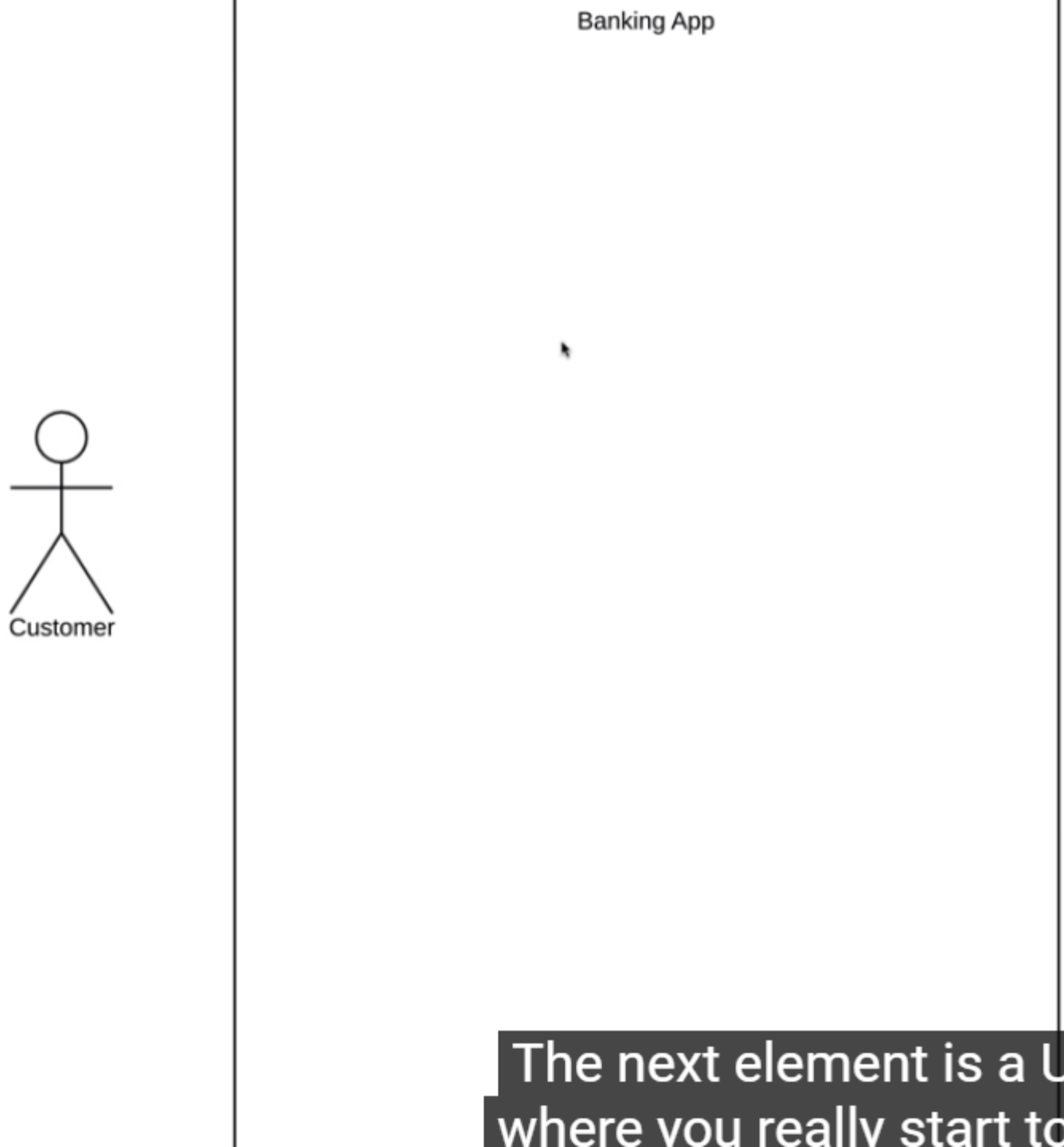
Initiates the use of the system

Secondary Actors

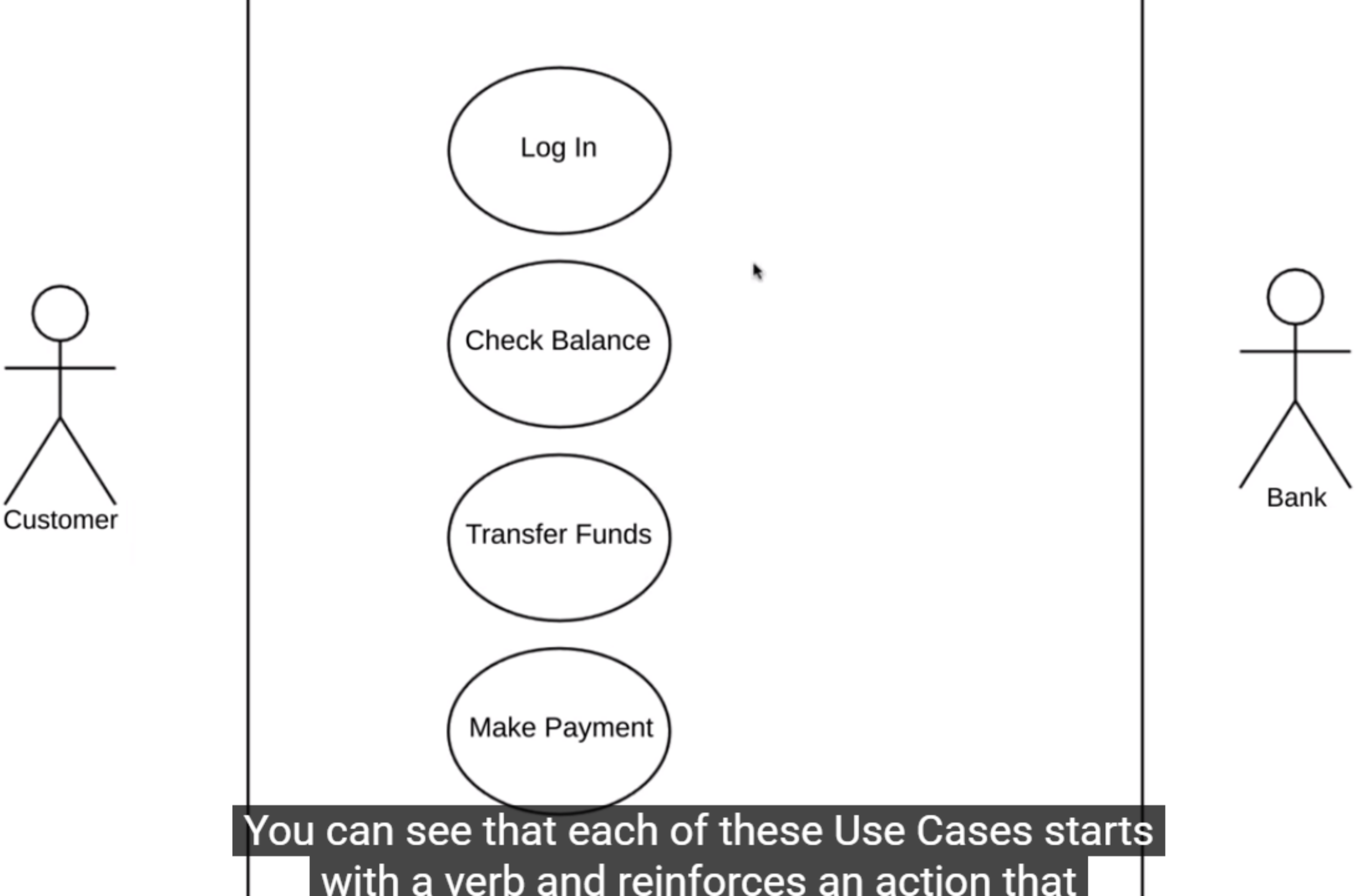
Reactionary

does the Bank engage with our system to provide the balance.

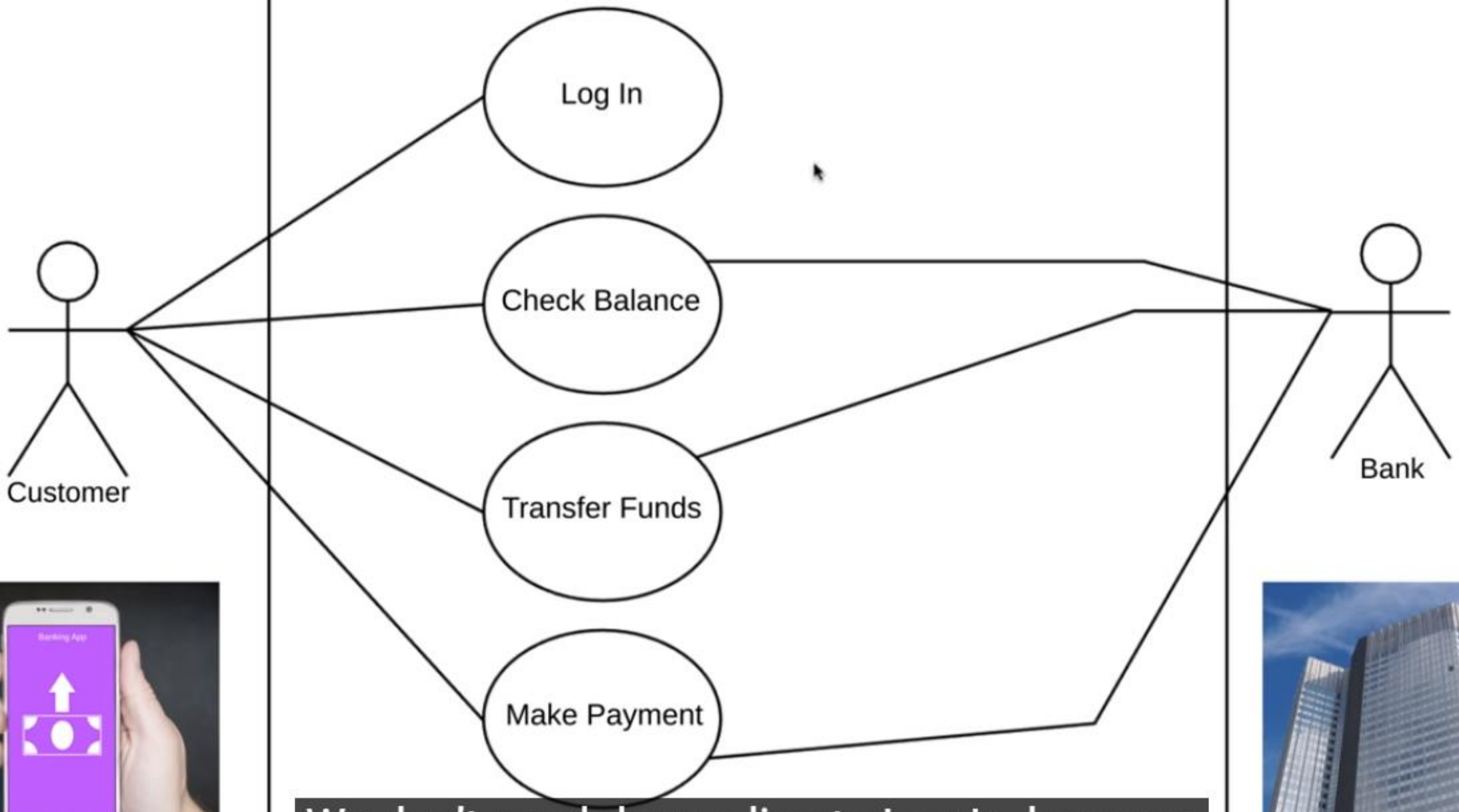
Use Case



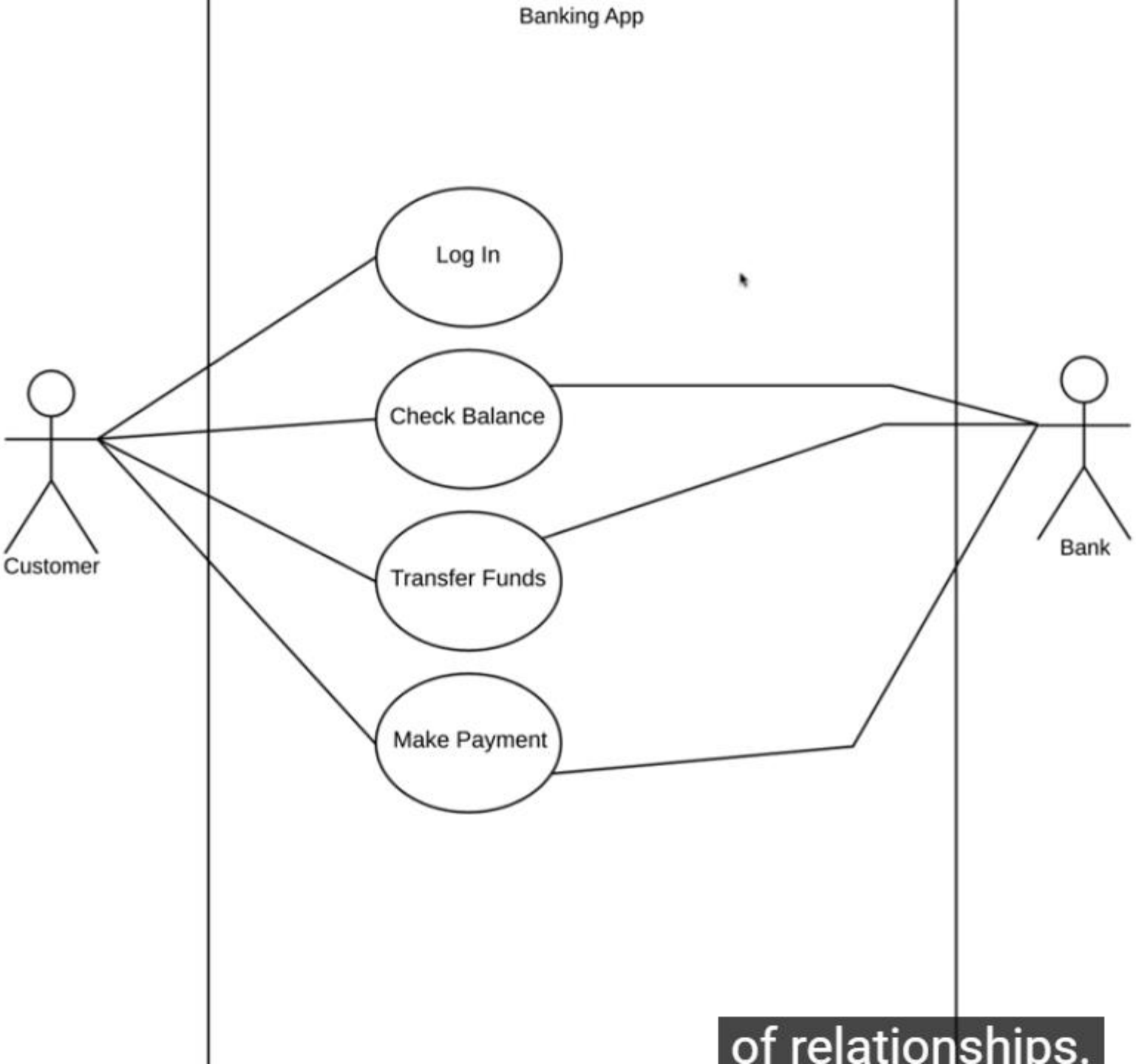
The next element is a Use Case and this is where you really start to describe what your



You can see that each of these Use Cases starts with a verb and reinforces an action that



We don't need draw a line to Log In, because that process happens within the Banking App.



Relationships

Association

Include

Extend

Generalization

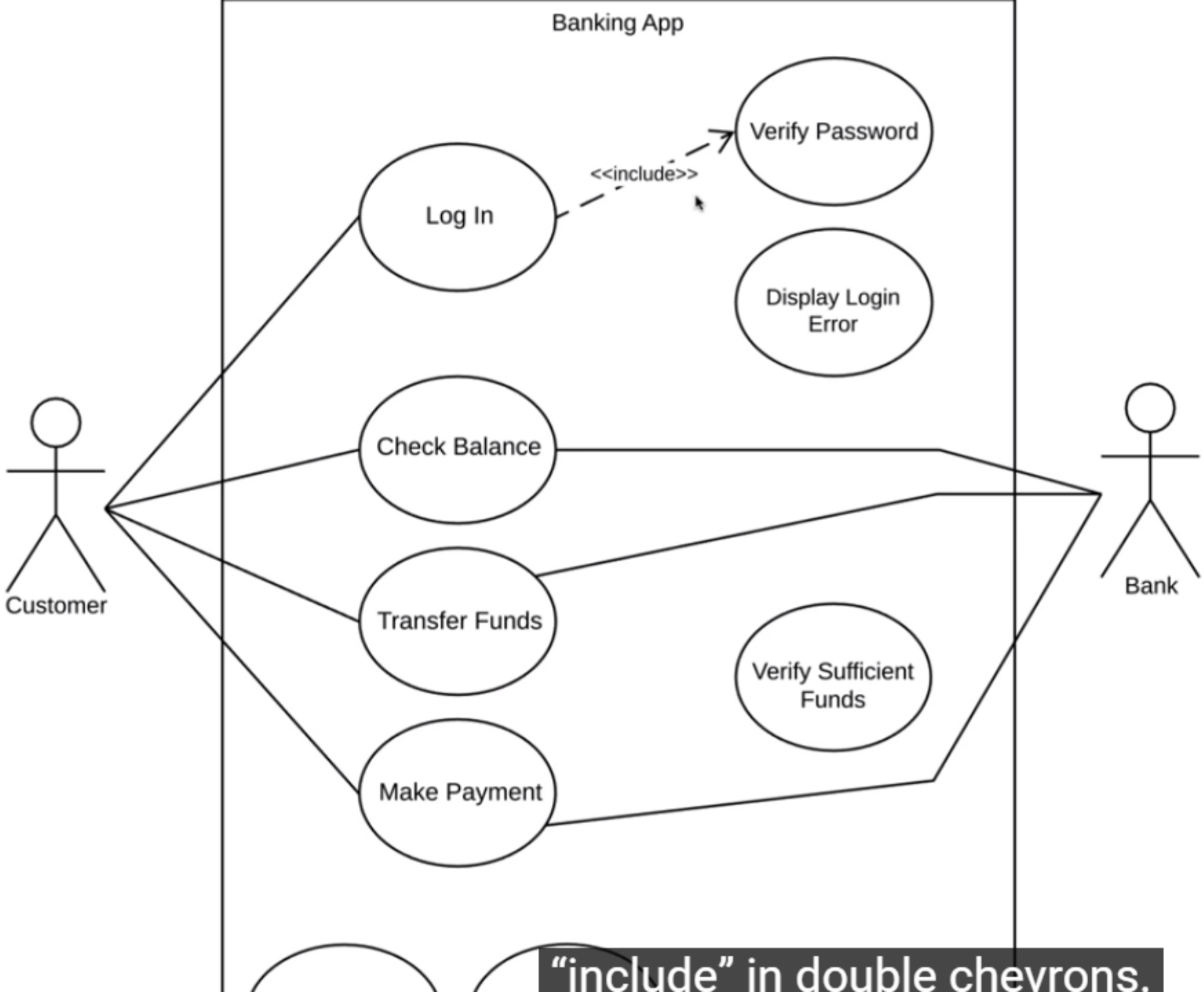
of relationships.

Include

An Include relationship shows dependency between a base use case and an included use case.

Include

towards the included use case.

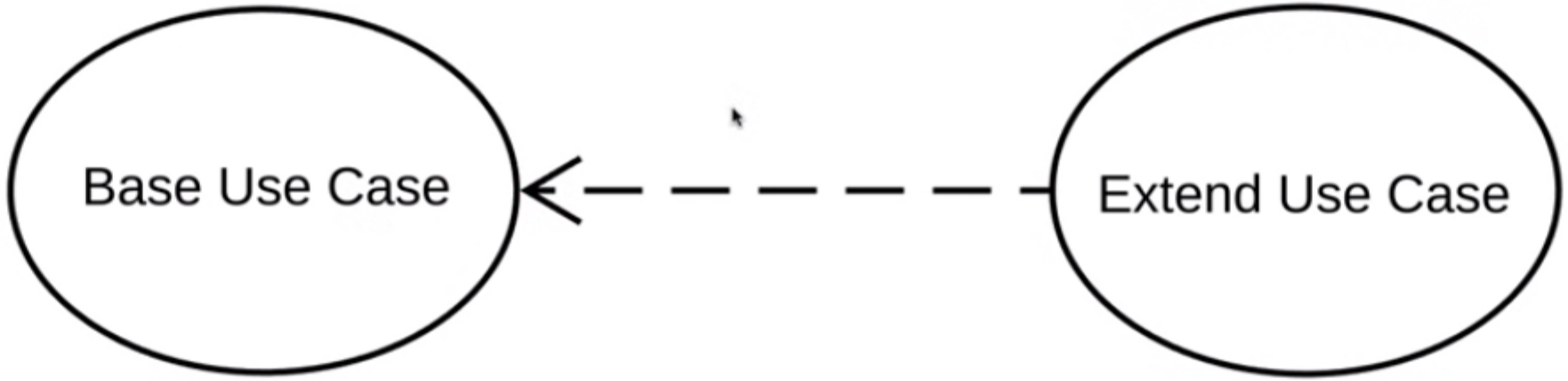


"include" in double chevrons.

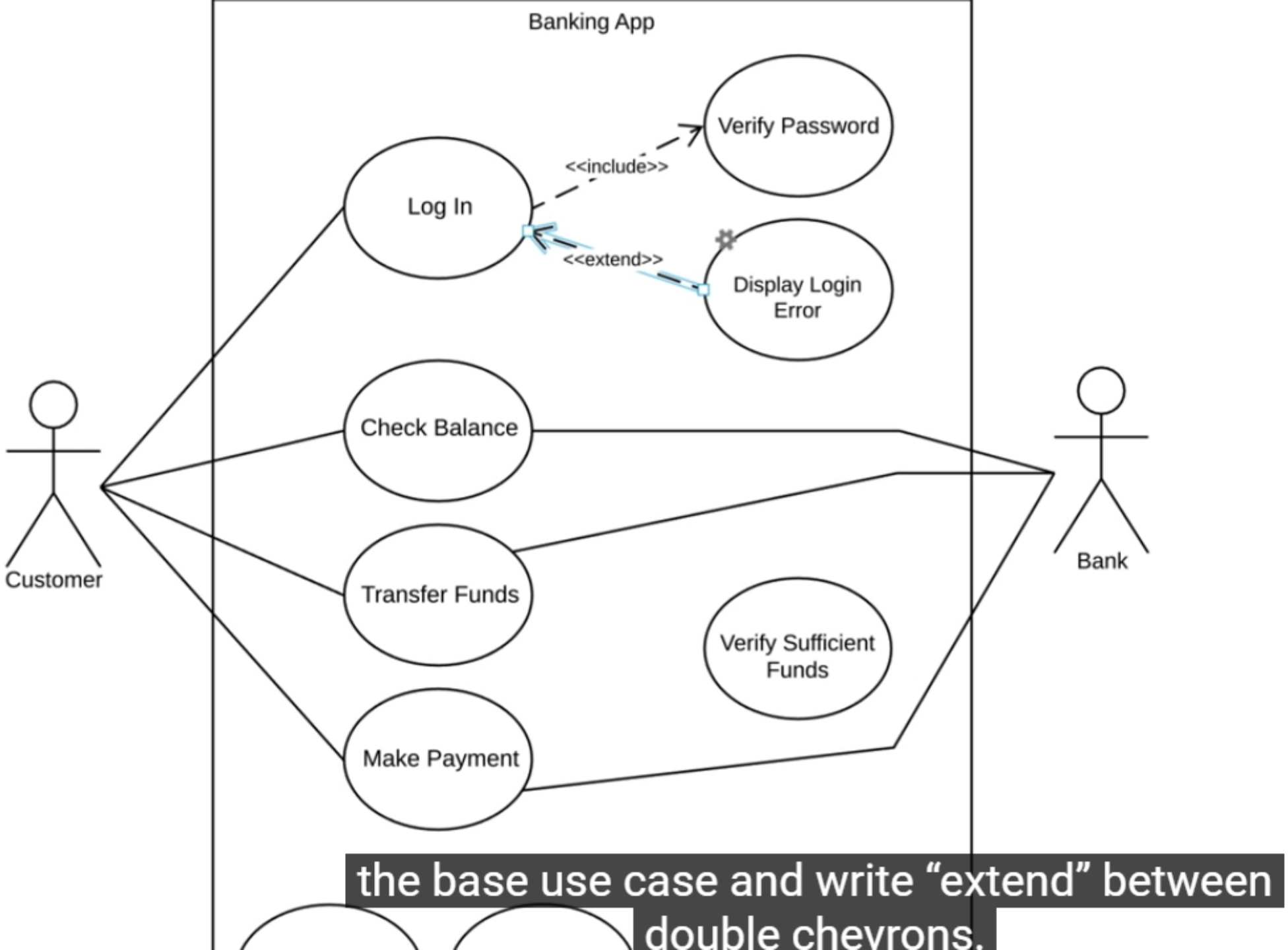
Extend

An extend relationship has a base use case and an extend use case.

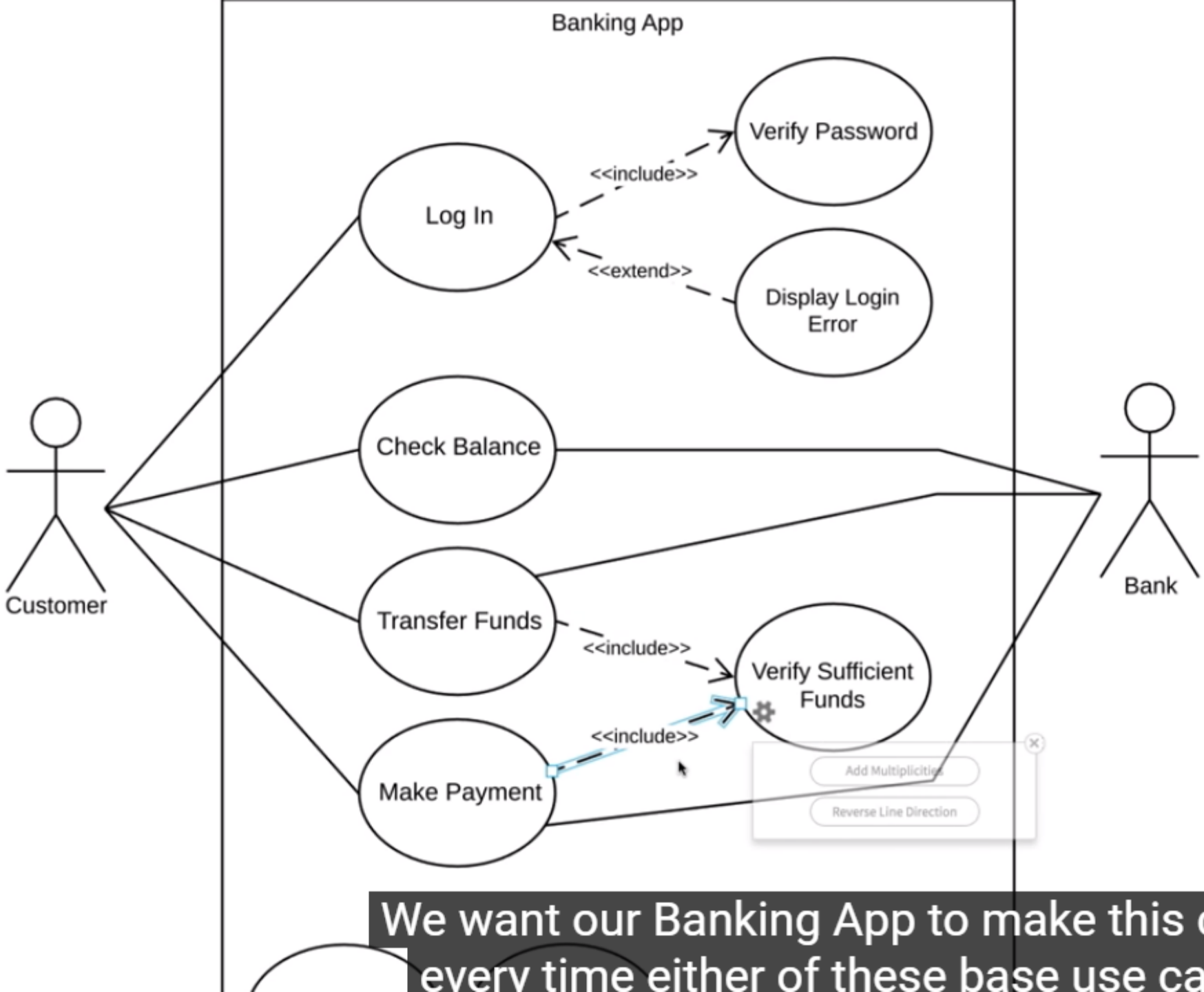
Extend



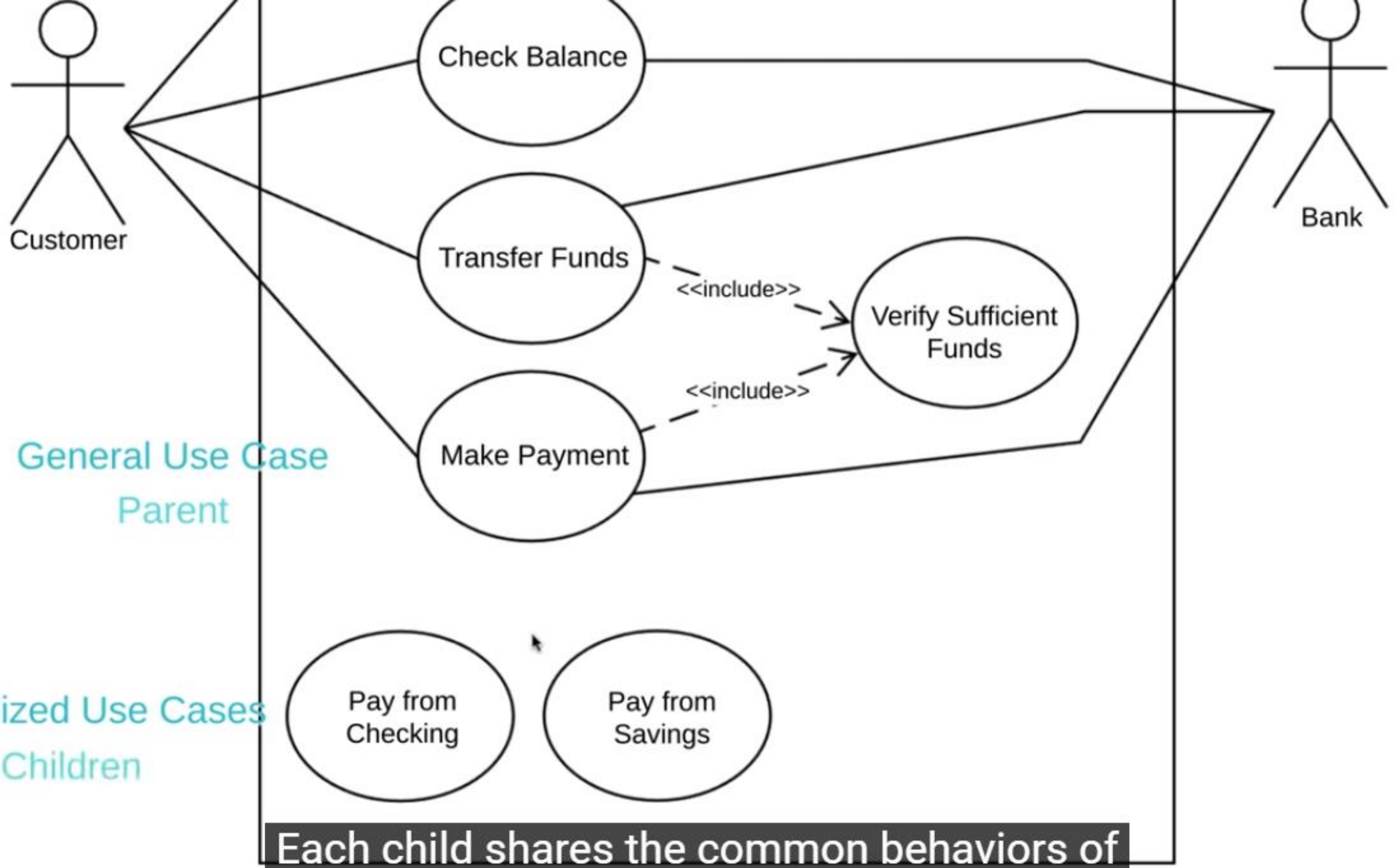
towards the base use case.



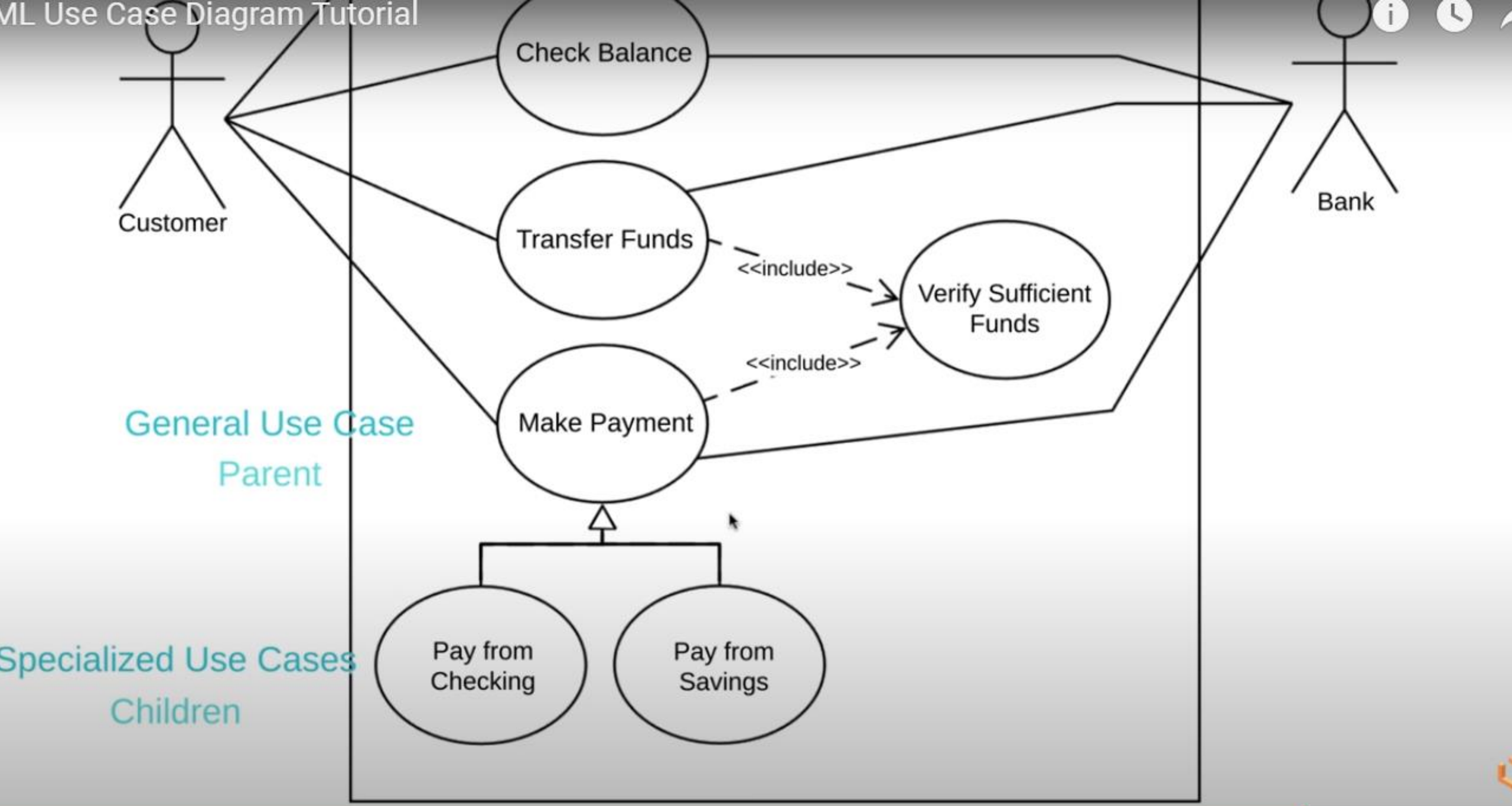
the base use case and write “extend” between double chevrons.

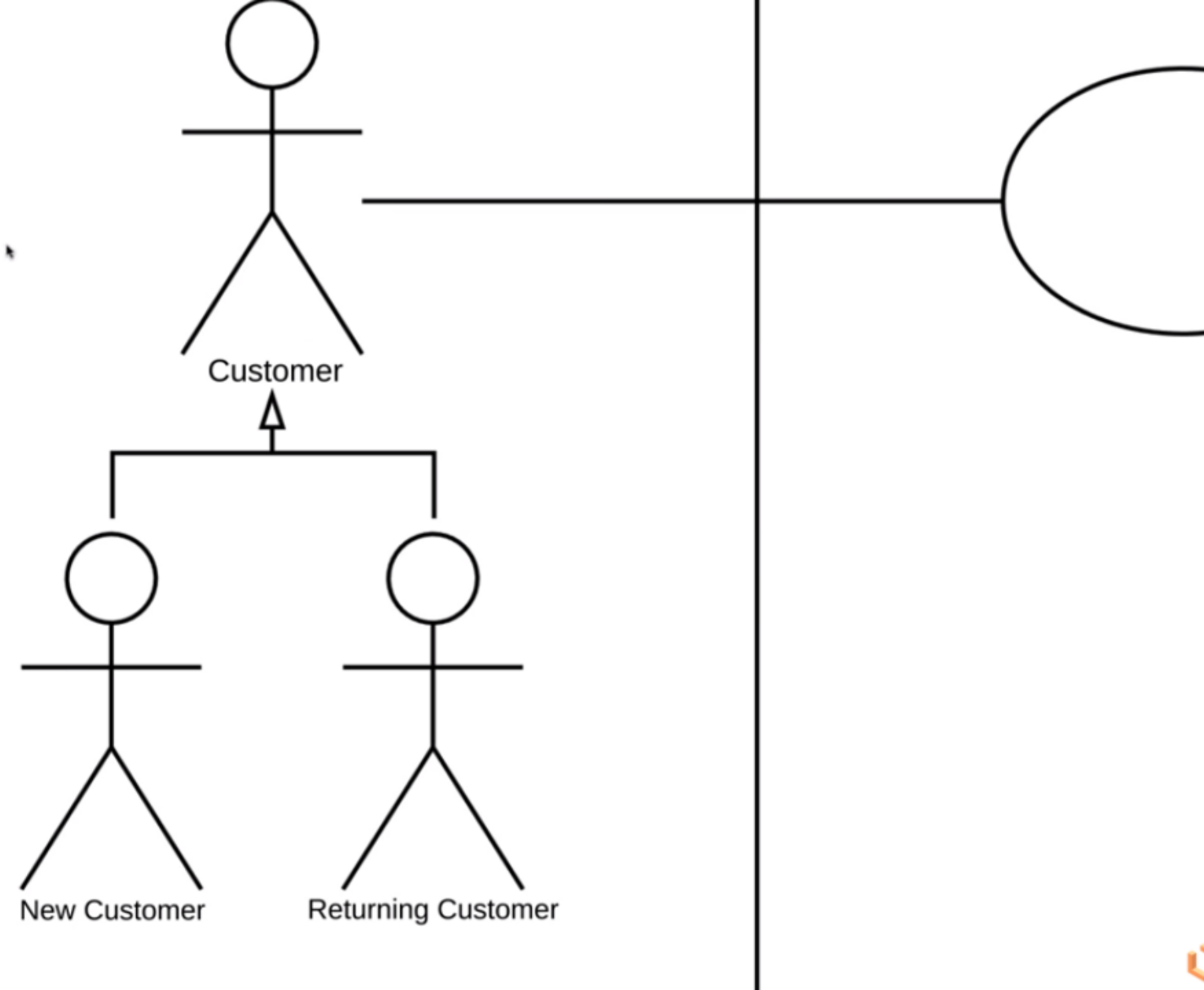


We want our Banking App to make this check every time either of these base use cases

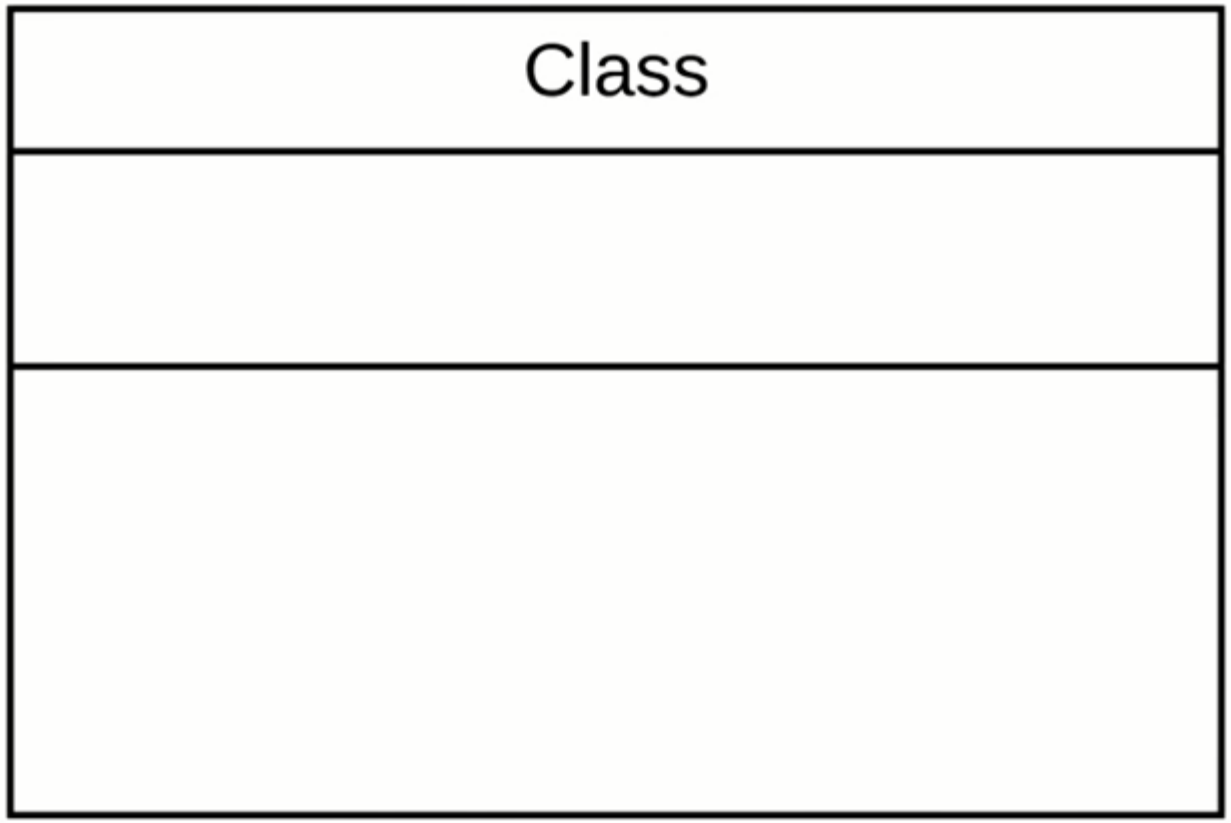


Each child shares the common behaviors of the parent, but each child adds something


ML Use Case Diagram Tutorial
Check Balance
Customer
Bank
Transfer Funds
<<include>>
Verify Sufficient Funds
<<include>>
Make Payment
General Use Case
Parent
Specialized Use Cases
Children
Pay from Checking
Pay from Savings
11:26 / 13:23


Customer
New Customer
Returning Customer

Zoo System

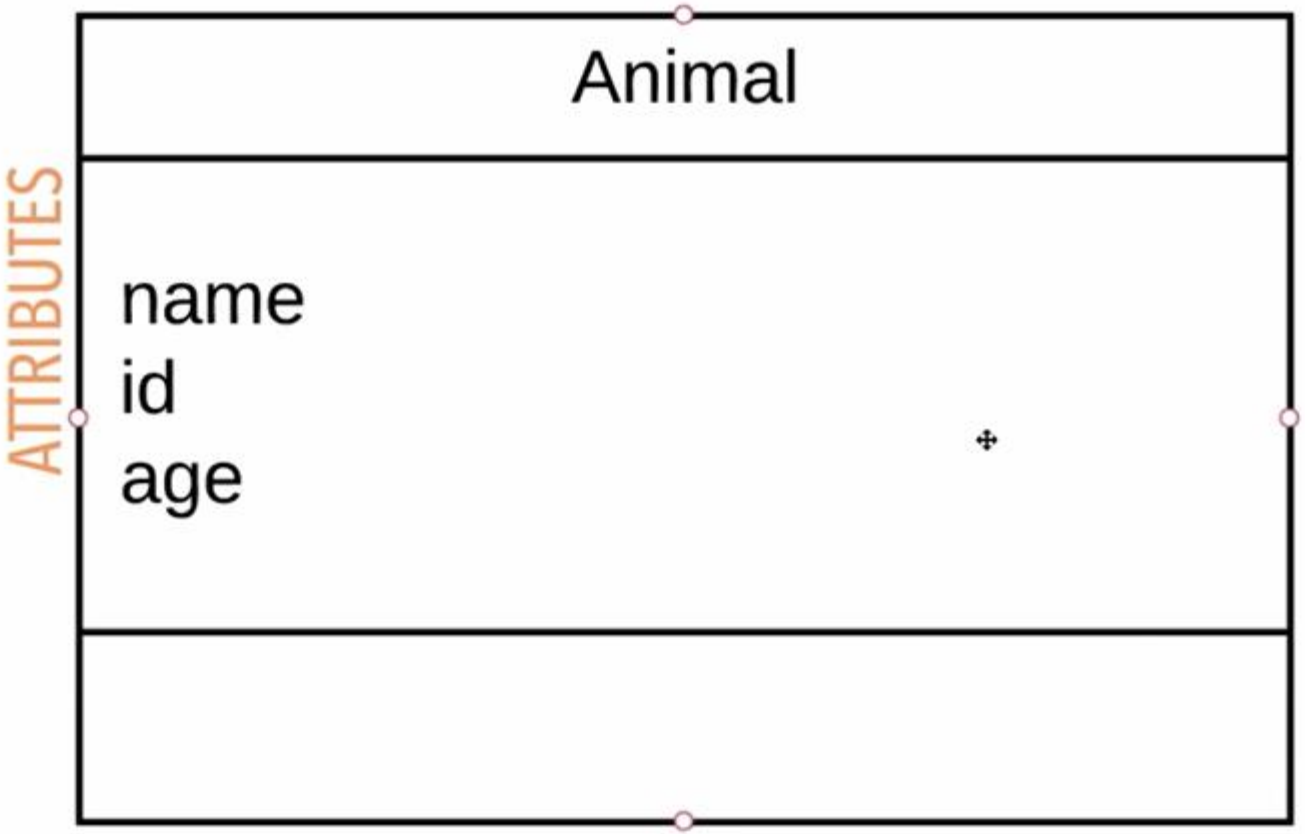


Zoo System

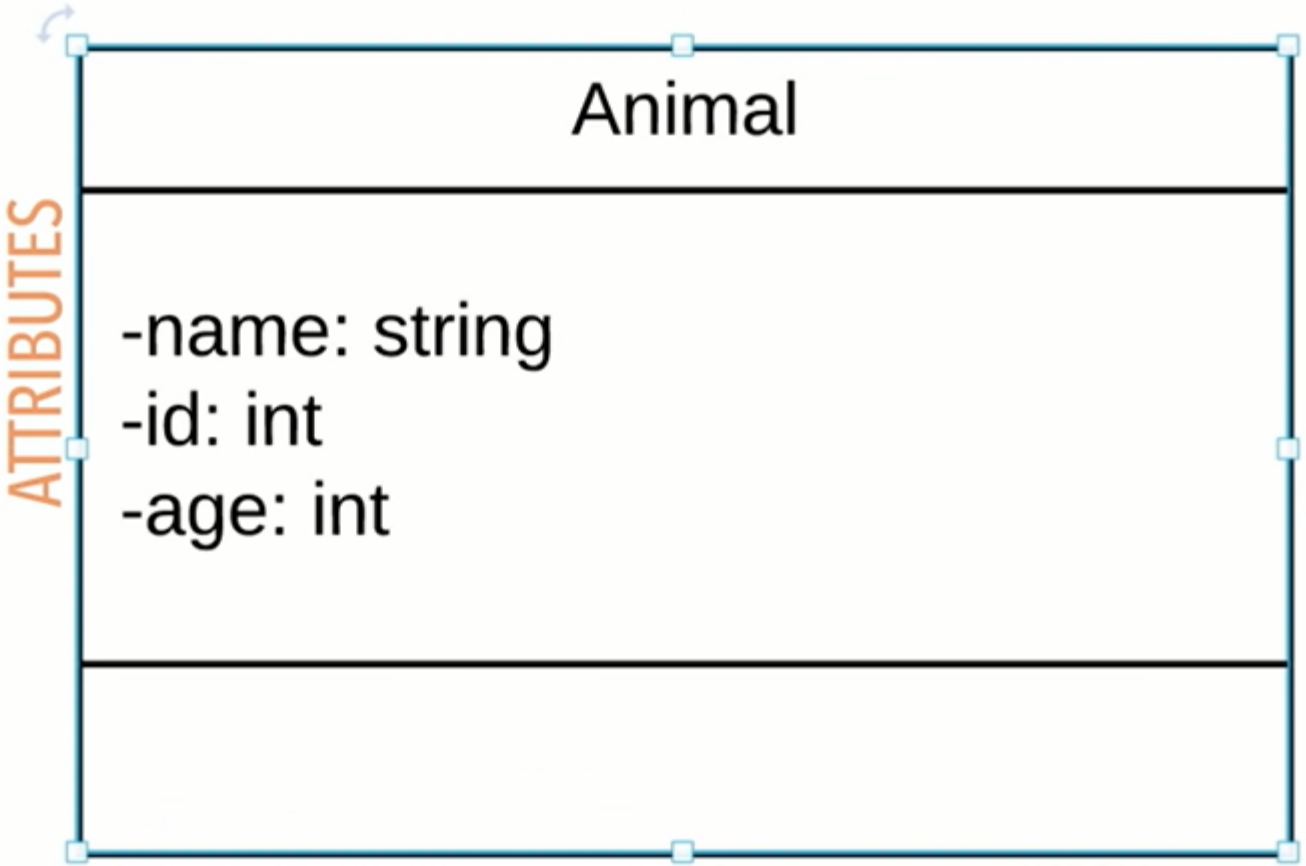
Animal

Instances of Animal class

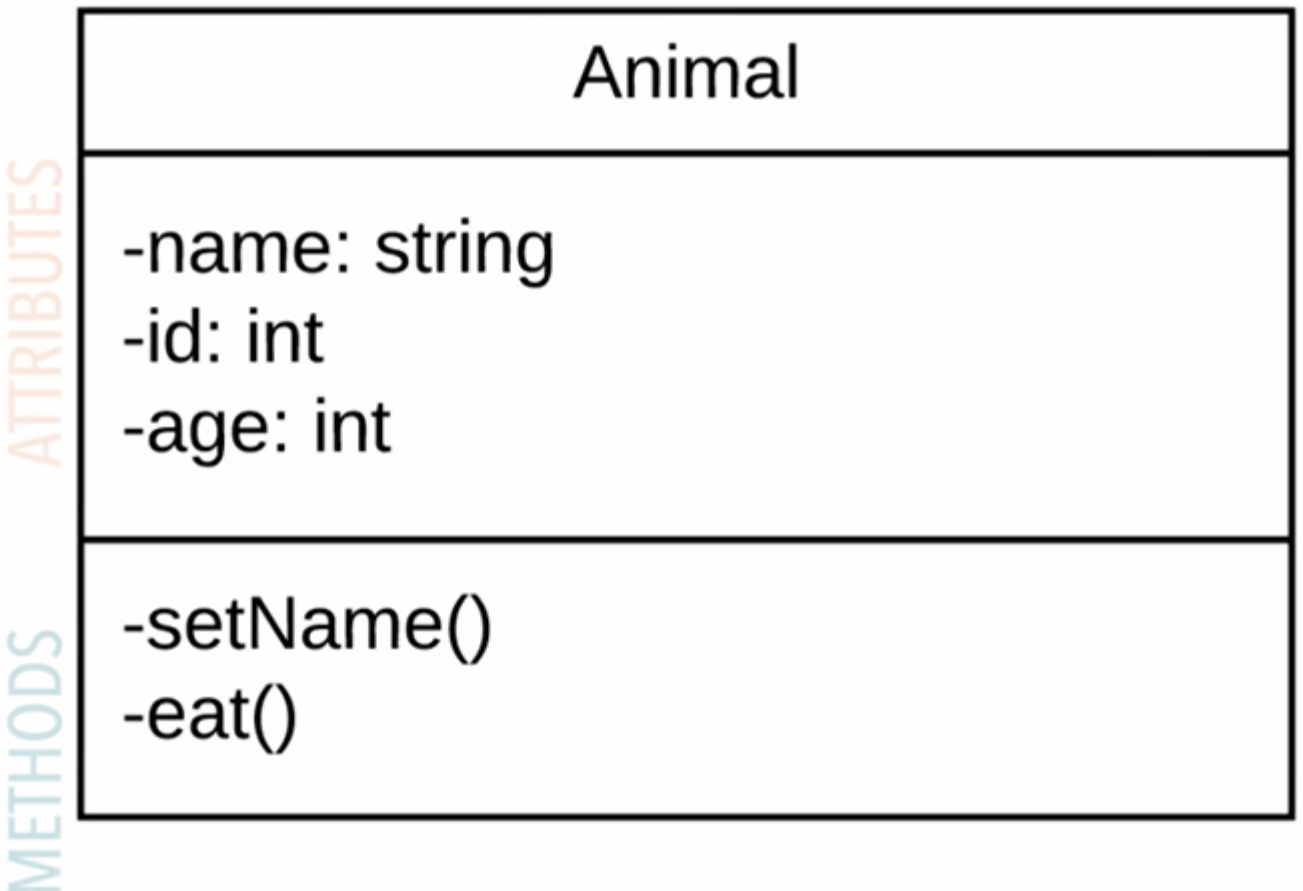
Zoo System



Zoo System



Zoo System


Animal
ATTRIBUTES
-name: string
-id: int
-age: int
METHODS
-setName()
-eat()

Zoo System

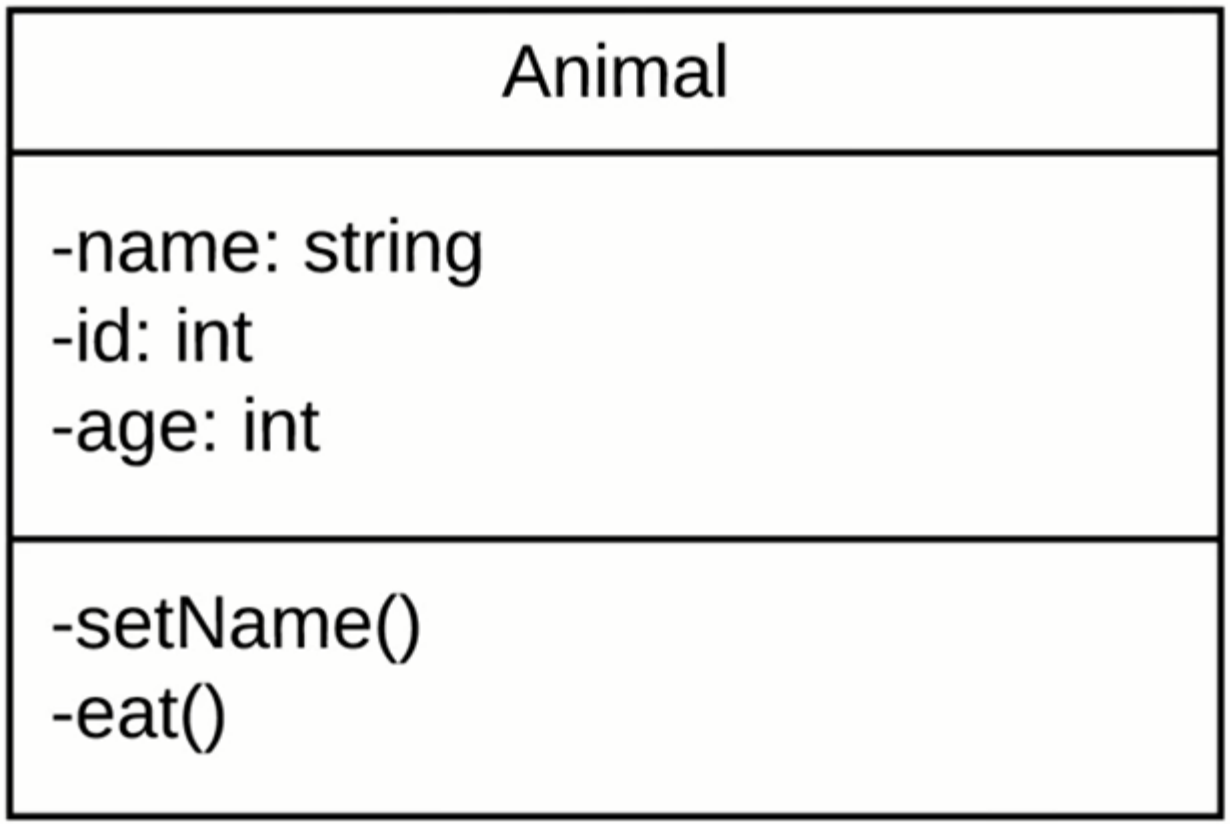
Animal
#name: string #id: int #age: int
#setName() #eat()

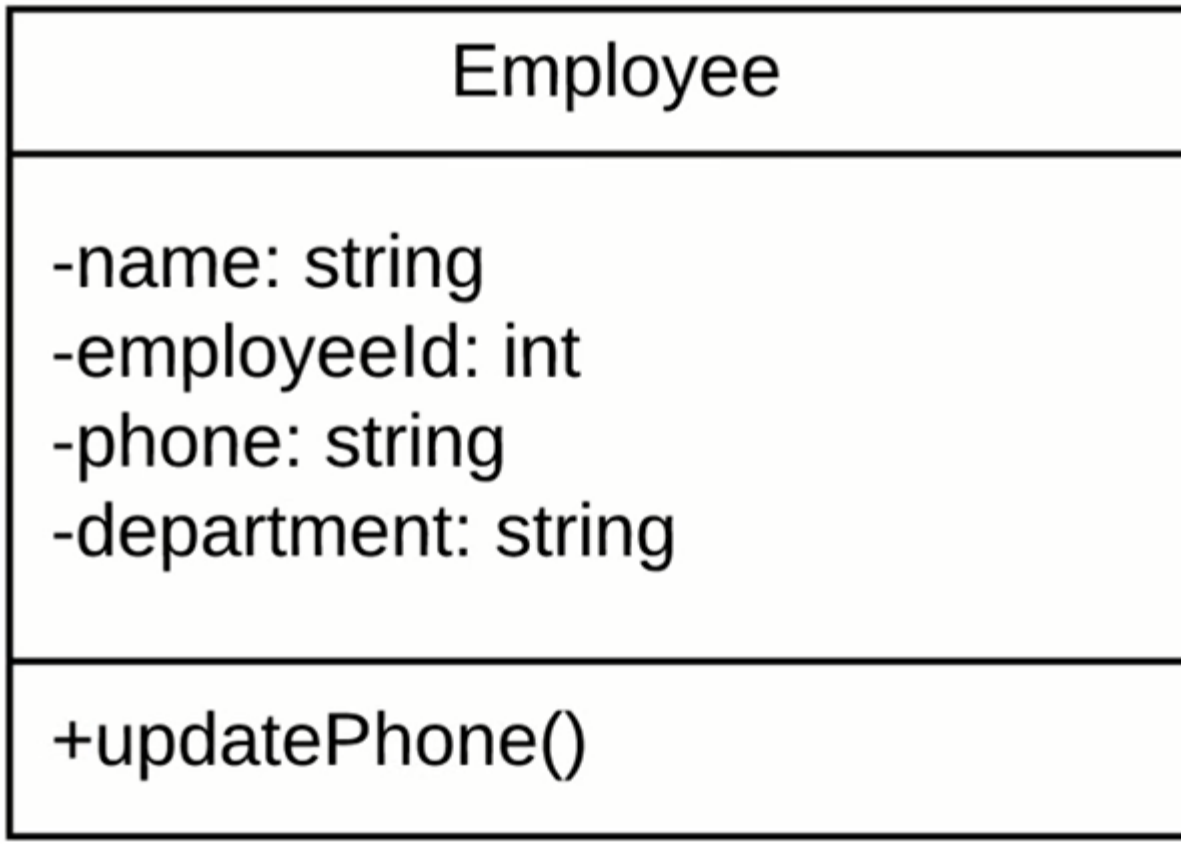
Visibility

- \- private
- \+ public
- \# protected

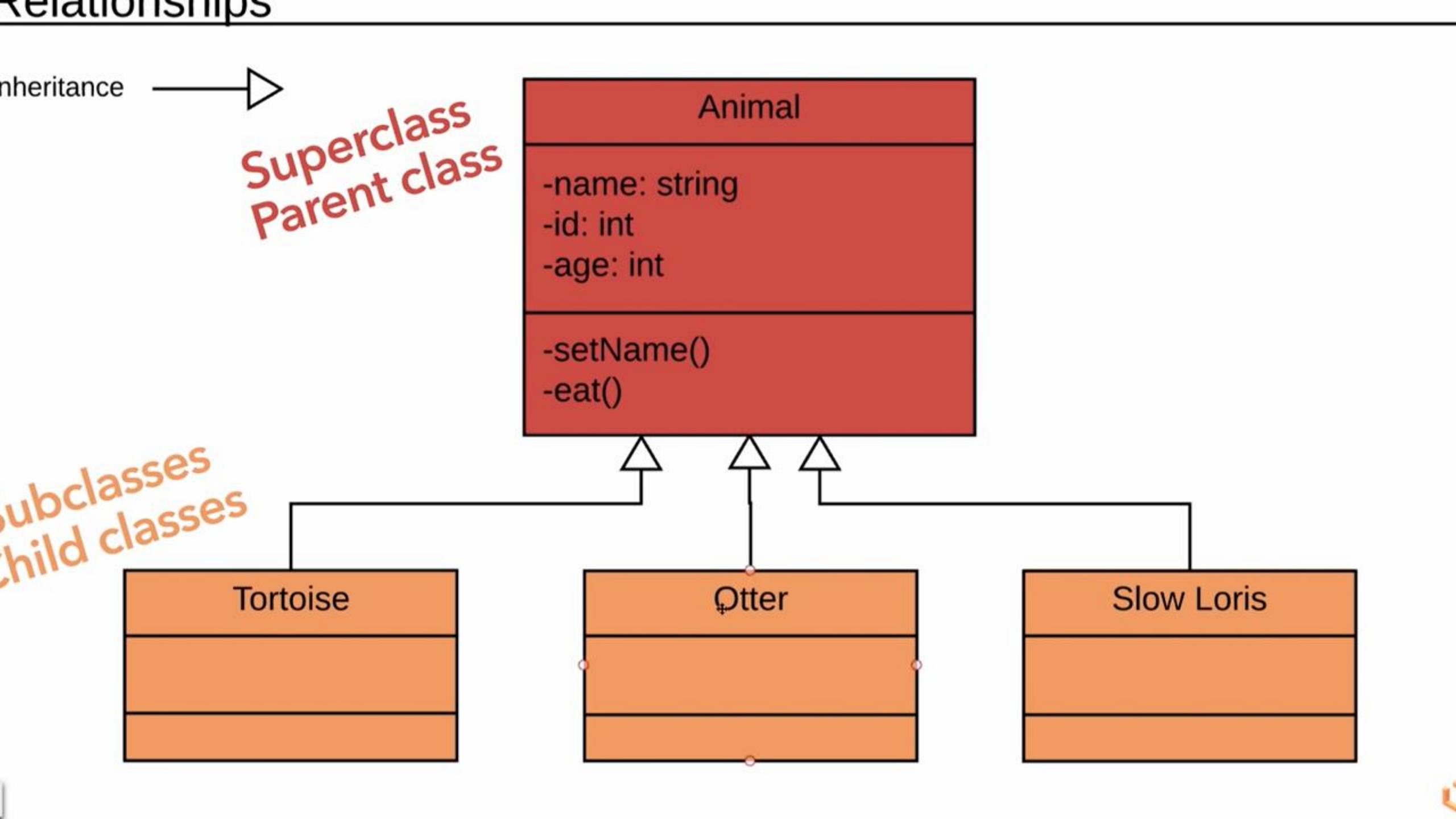
same class or subclass

System

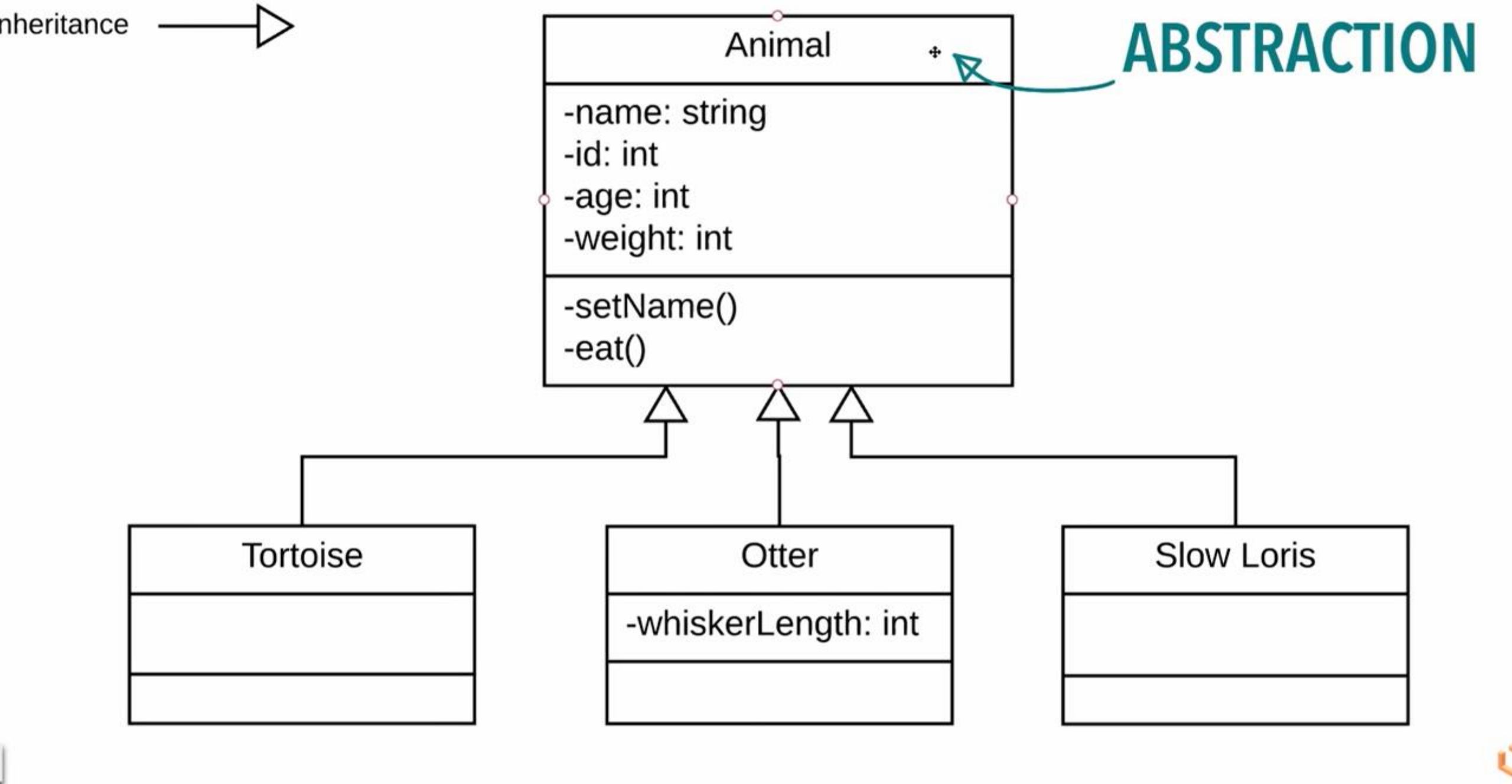

Animal
-name: string
-id: int
-age: int
-setName()
-eat()

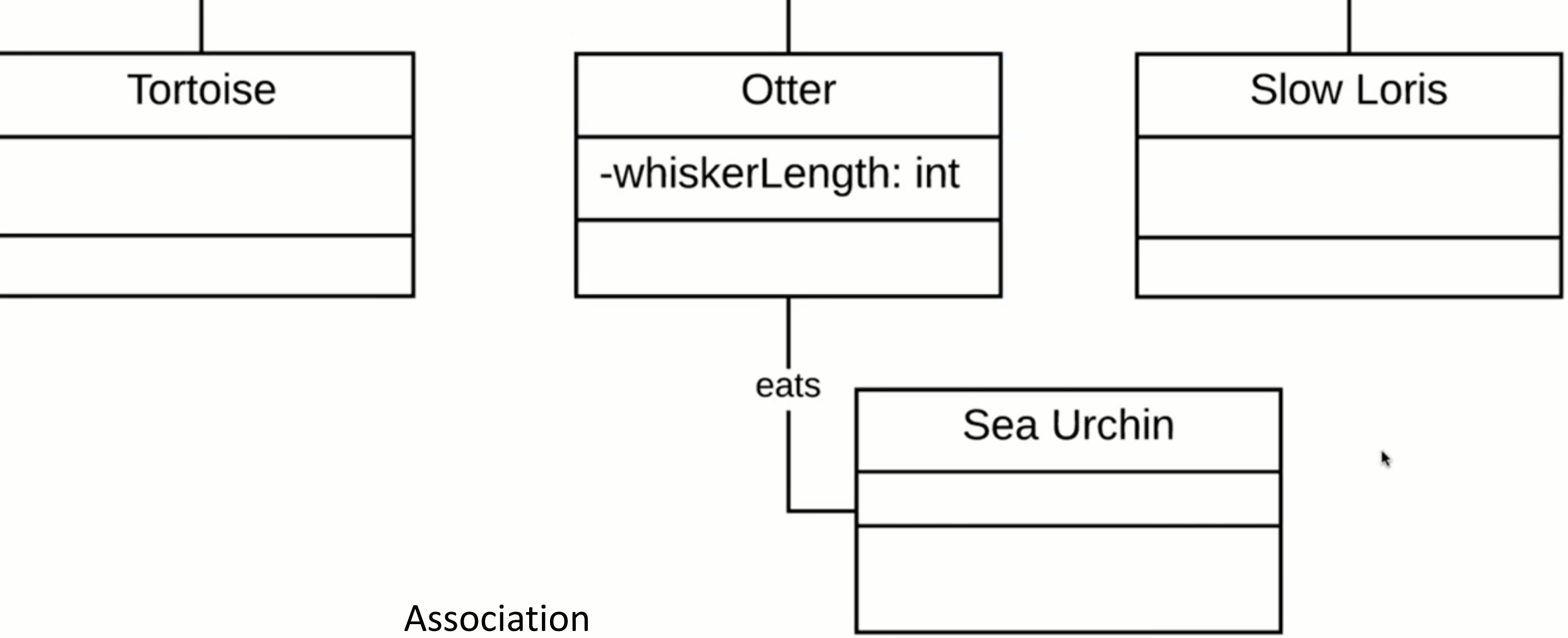

Employee
-name: string
-employeeId: int
-phone: string
-department: string
+updatePhone()

Relationships



Relationships

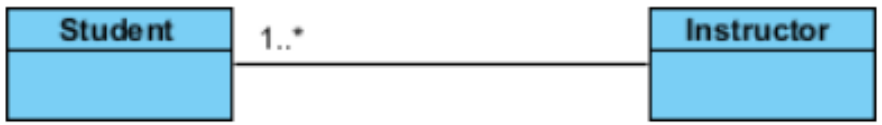




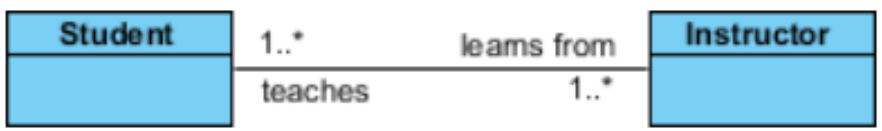
Association

A single student can associate with multiple teachers:

The example indicates that every Instructor has one or more Students:



We can also indicate the behavior of an object in an association (i.e., the role of an object) using role names.



ML Class Diagram Tutorial

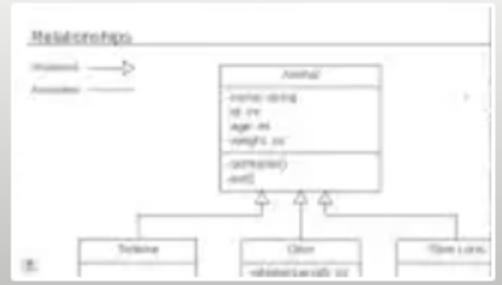
Liberation Sans 8 pt B I U A 2 px None None

-eat

Aggregation

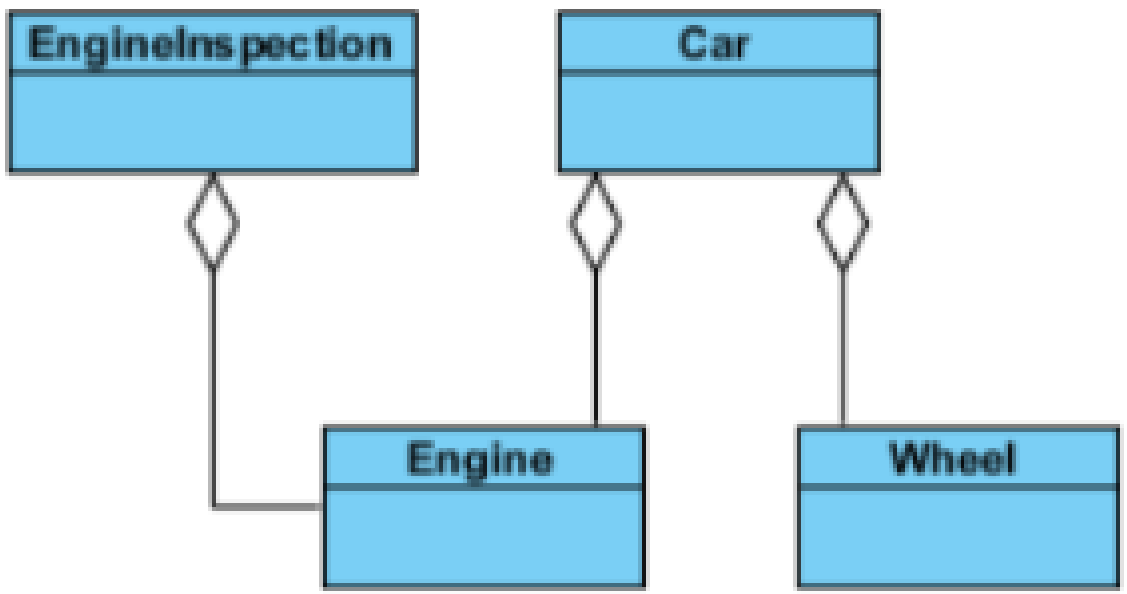
Creep

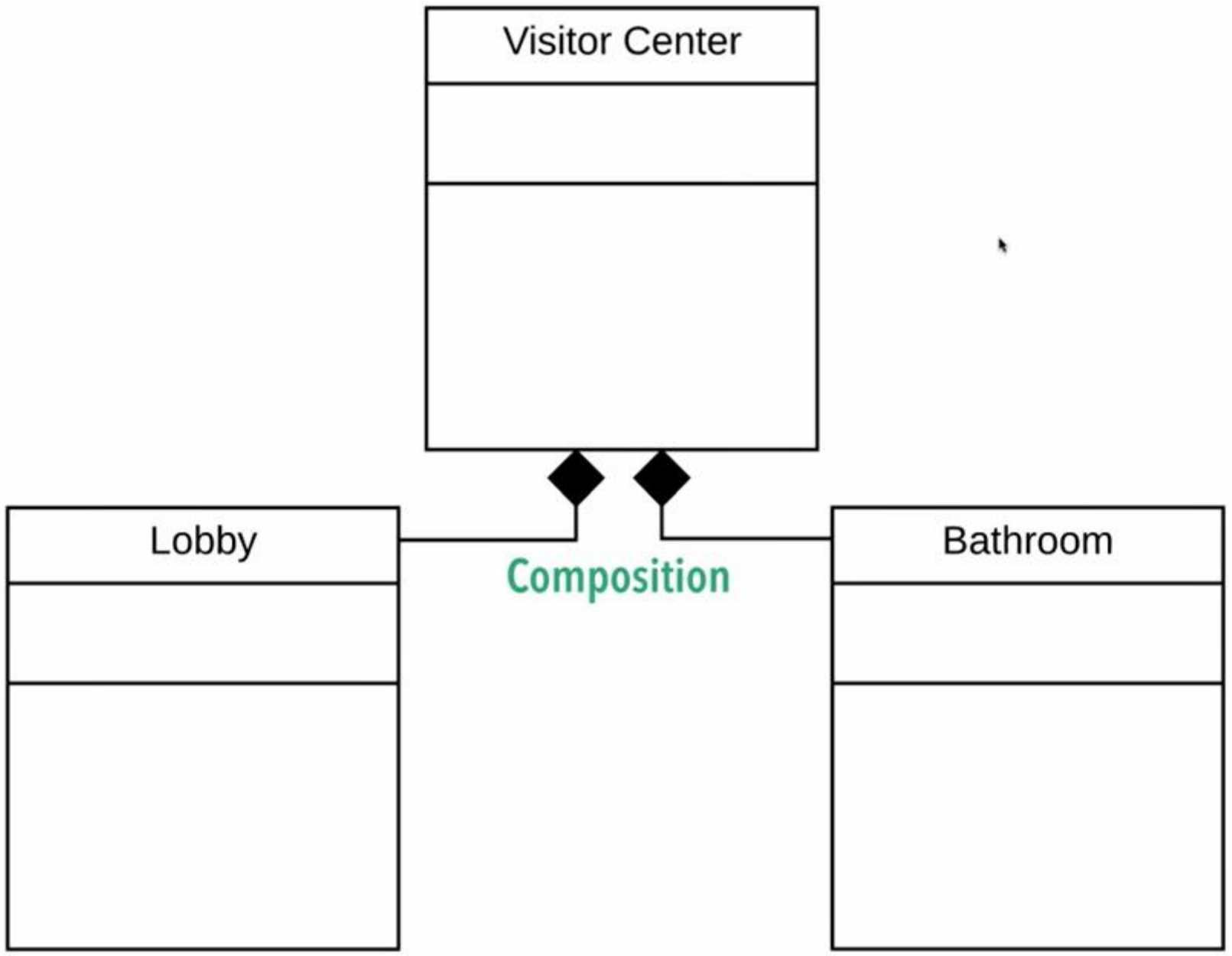
Tortoise



5:56

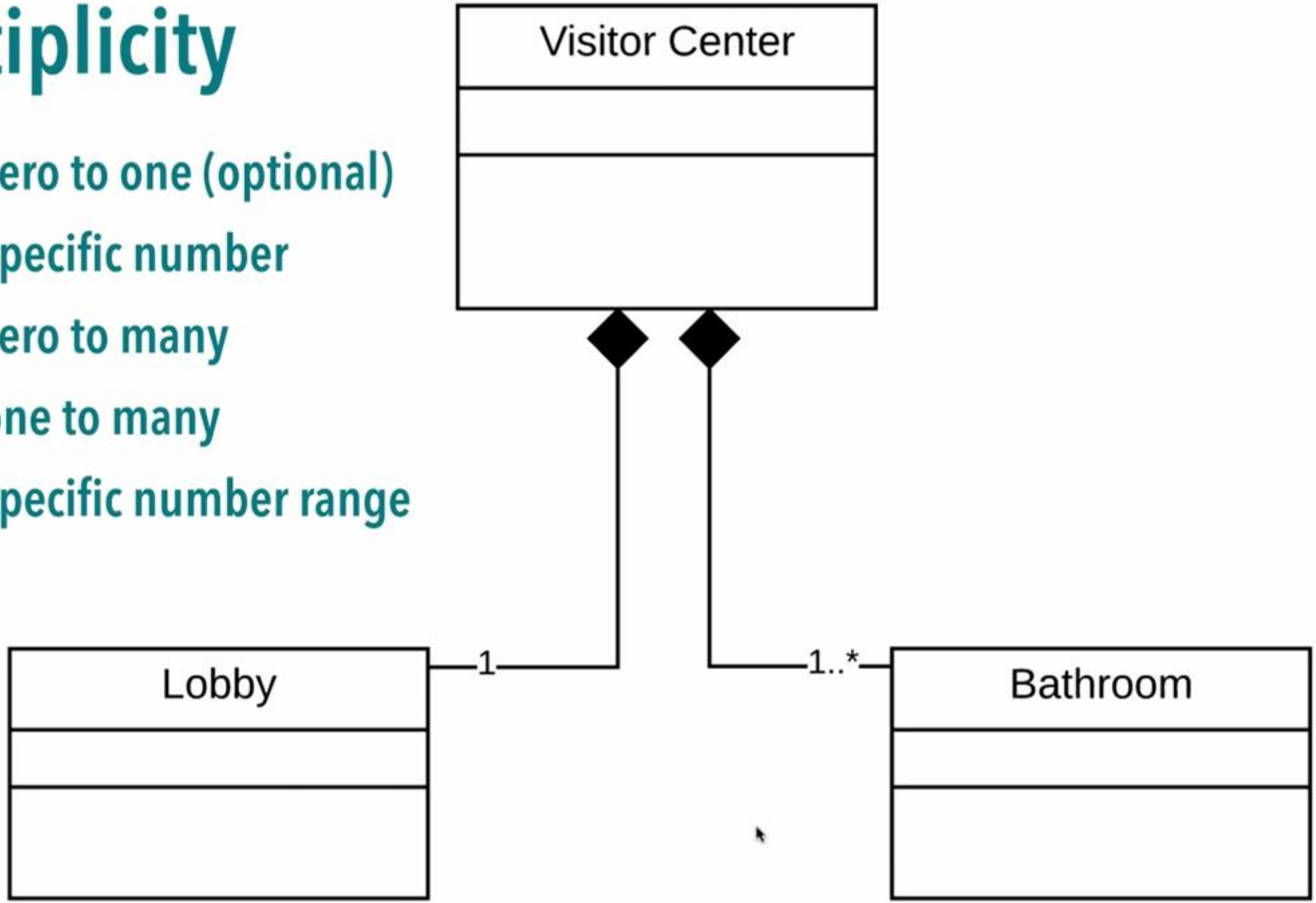
6:30 / 10:16


EngineInspection
Car
Engine
Wheel


Visitor Center
Lobby
Bathroom
Composition

Multiplicity

0..1	zero to one (optional)
n	specific number
0..*	zero to many
1..*	one to many
m..n	specific number range



View this example

UML Tutorial

Edit View Insert Arrange Share Help Saved

Feature Find Present