



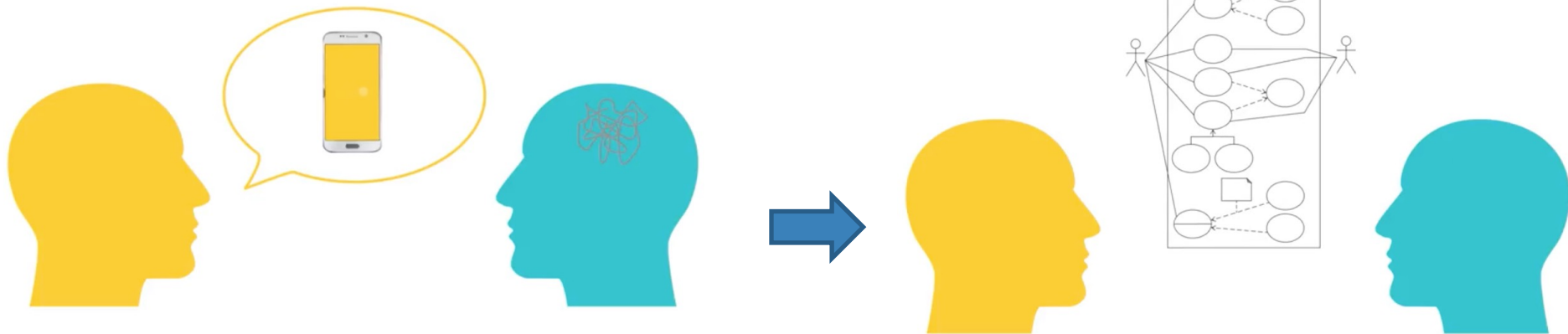
# *USE-CASE Modeling*

# Introduction

- ▷ A use case illustrates **the activities** that are performed **by the user** of system
- ▷ Use case can be used to document the **current system** (**As-Is system**) or the **new system** being developed (**To-Be system**)
- ▷ A key concept of use case modeling is that it helps us design a system from the **end user's perspective**.
- ▷ It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

# Introduction

- ▷ Each use case is **only one function** in which user interact with the system
- ▷ A use case may contain several “paths” that a user can take while interacting with the system ( **in web bookstore , the user might search by subject , by other , or by title**)
- ▷ Each path through the use case is referred to as a *scenario*.



# Introduction

- ▷ Define function in system (What)
- ▷ Black Box

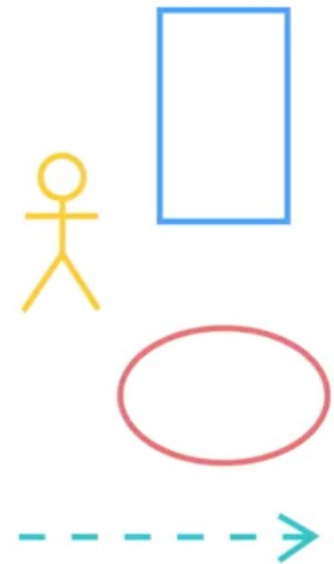
A use case diagram is usually simple. It does not show the detail of the use cases:

- It only summarizes **some of the relationships** between use cases, actors, and systems.
- It does **not show the order** in which steps are performed to achieve the goals of each use case.

# Component in Use Case Diagram

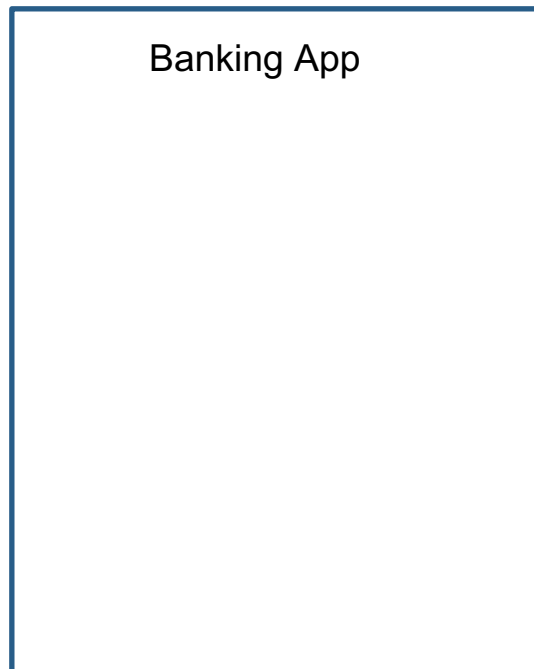
- ▷ Boundary System
- ▷ Actor
- ▷ Use Case
- ▷ Relationships

Systems  
Actors  
Use Cases  
Relationships



# Boundary System

- ▷ A collection of connected units organized to accomplish a purpose
- ▷ The system is the “Complete model”



## Systems

Website  
Software Component  
Business Process  
Application  
Etc.

# Actor

- ▷ An abstraction for entities outside a system, subsystem, or class that interact directly with the system. An actor participates in a use case or coherent set of use cases to accomplish an overall purpose.



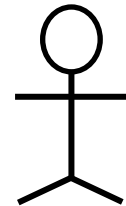
Actor

Person

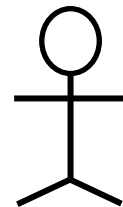
Organization

Another System

External Device



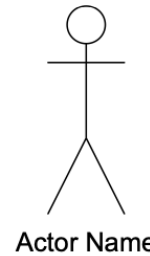
Customer



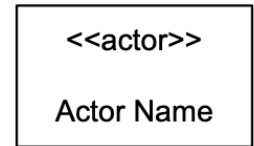
Bank

Banking App

# Actor



Primary & Stakeholder Actor



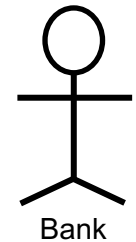
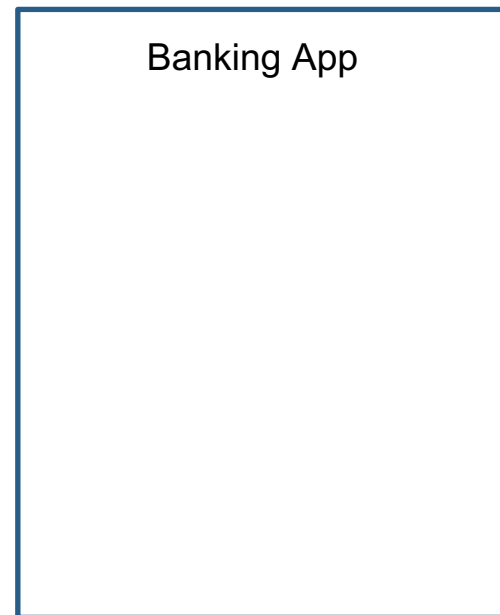
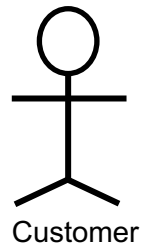
Primary & Stakeholder Actor

## ▷ **Primary Actors**

- Initiates the use of the system

## ▷ **Secondary Actors**

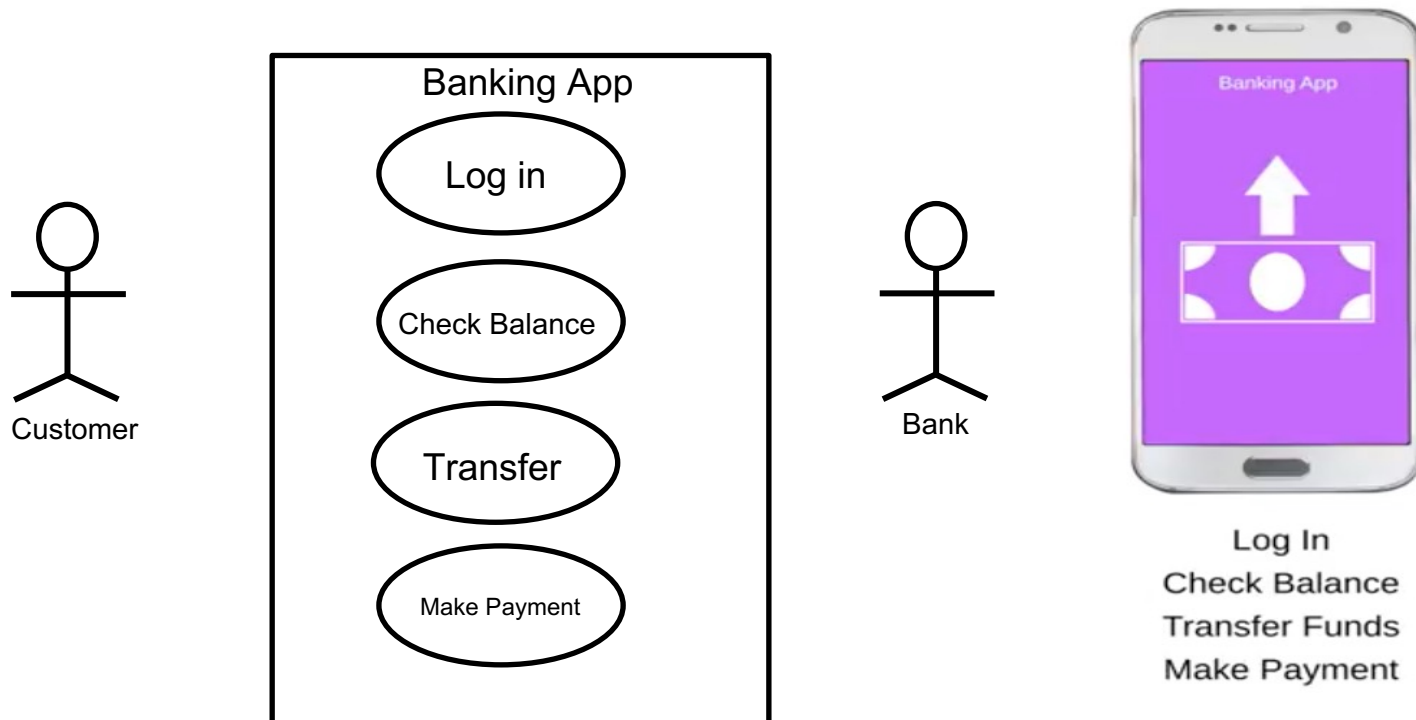
- Reactionary





# Use Case

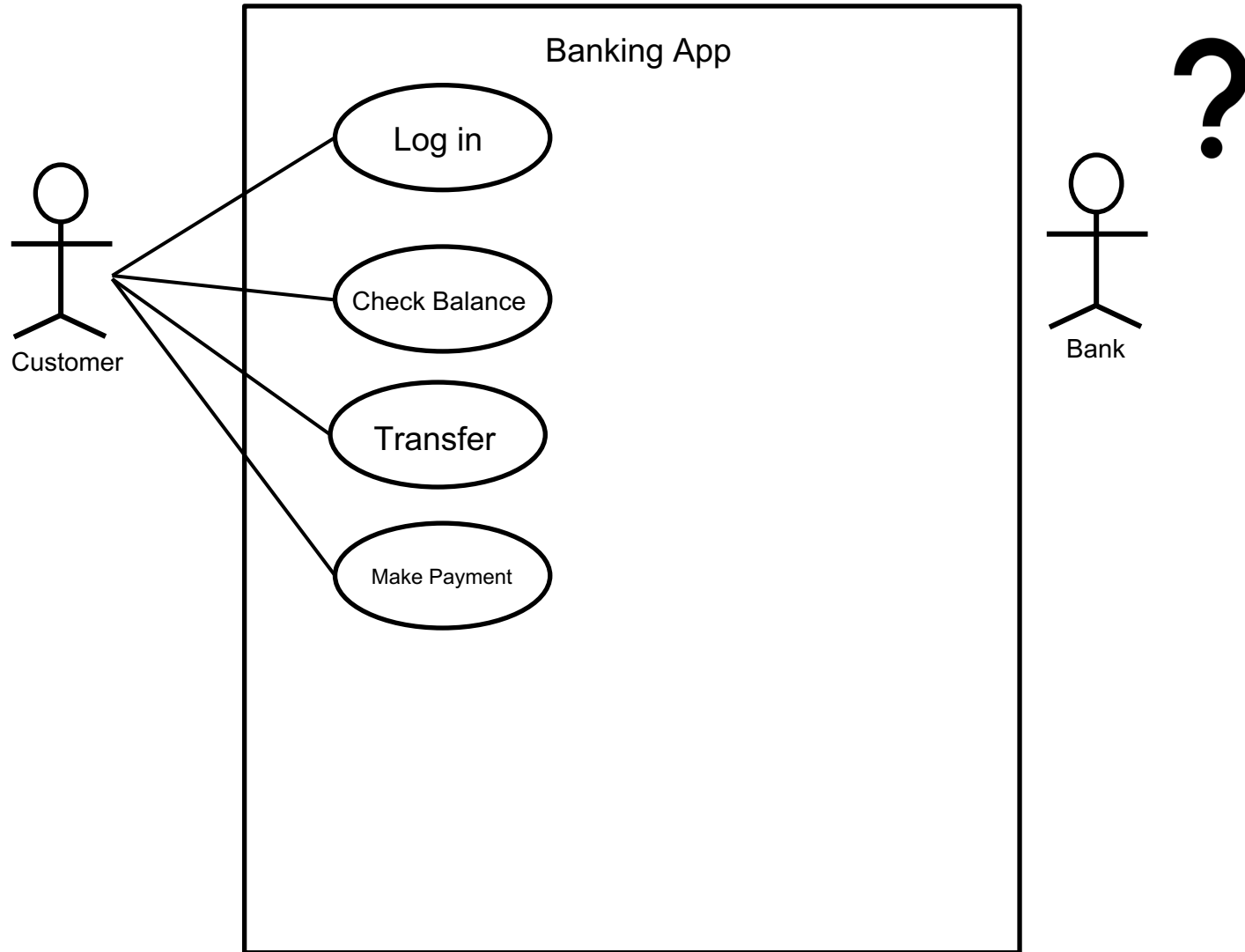
- ▷ The specification of **sequences of actions**, including variant sequence and error sequences, that a system, subsystem, or class can perform by interacting with outside actors

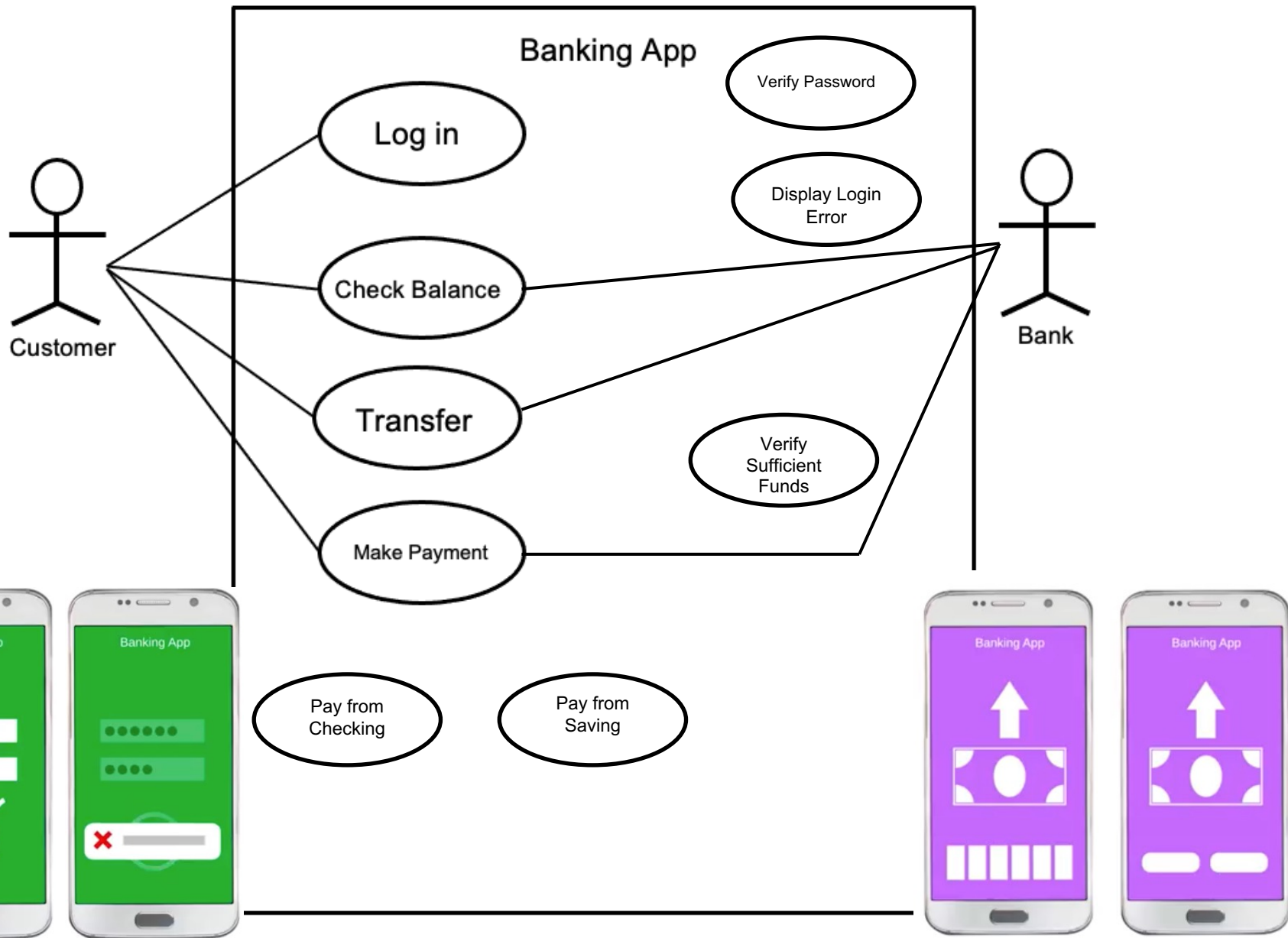


# Relationships

- ▷ Use case relationships explain how the user case is related to other use cases and user
- ▷ The four basic types of relationships are
  - Association
  - Include
  - Extend
  - Generalization

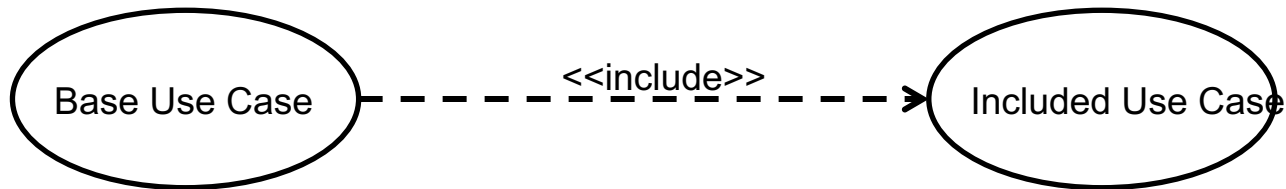
# Association Relationship





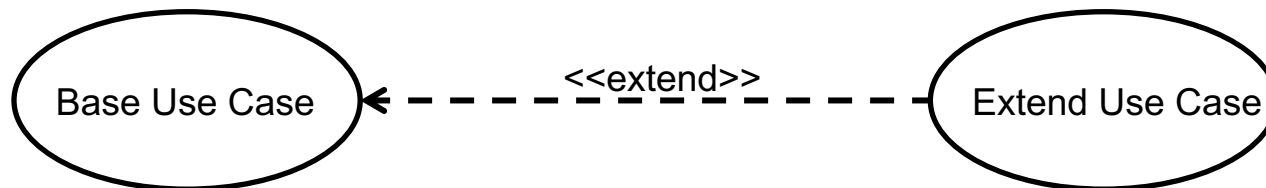
# Include Relationship

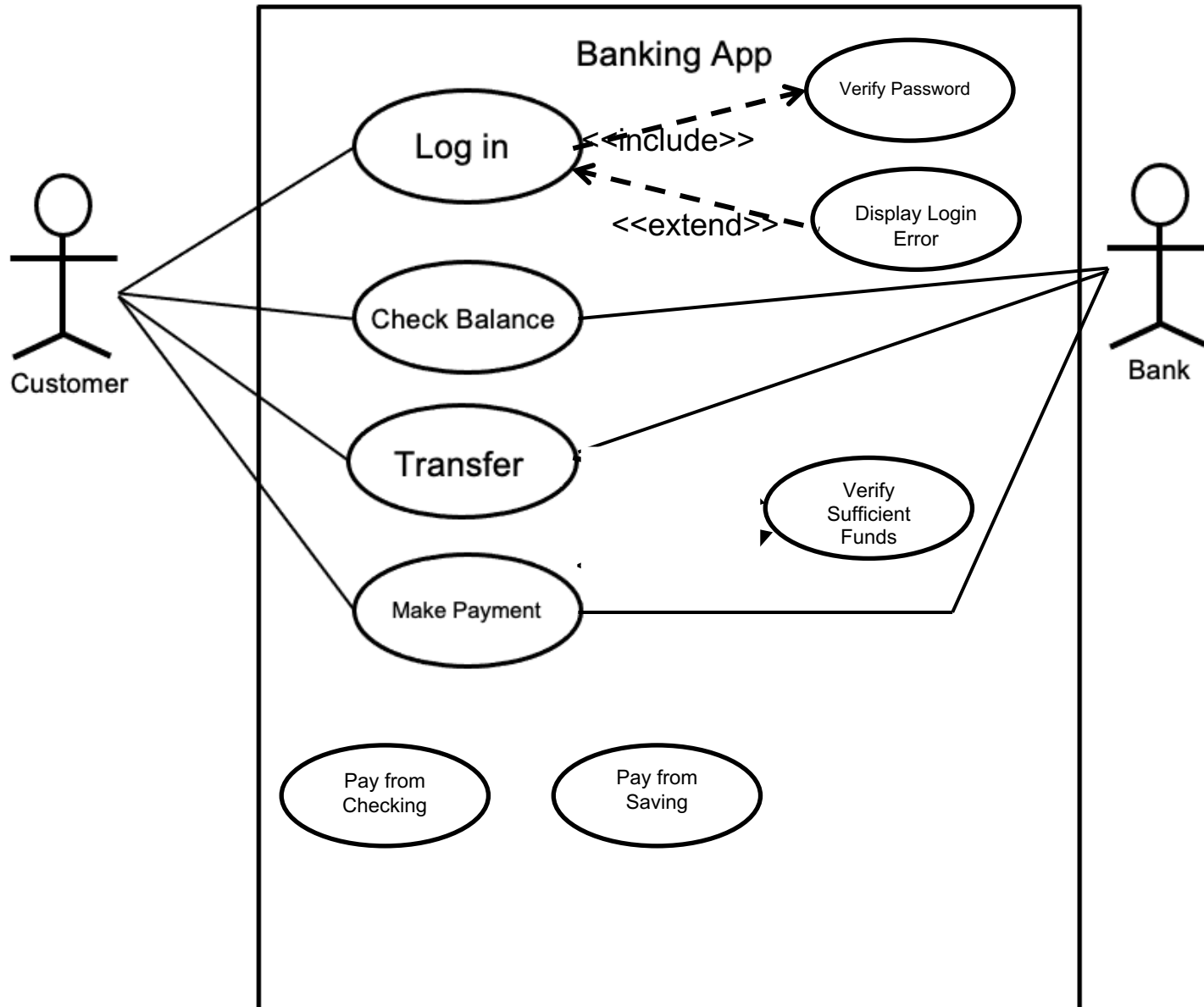
- ▷ Represents the mandatory inclusion of another use case.
- ▷ The include relationship enables *functional decomposition* – the breaking up of a complex use case into several simple ones

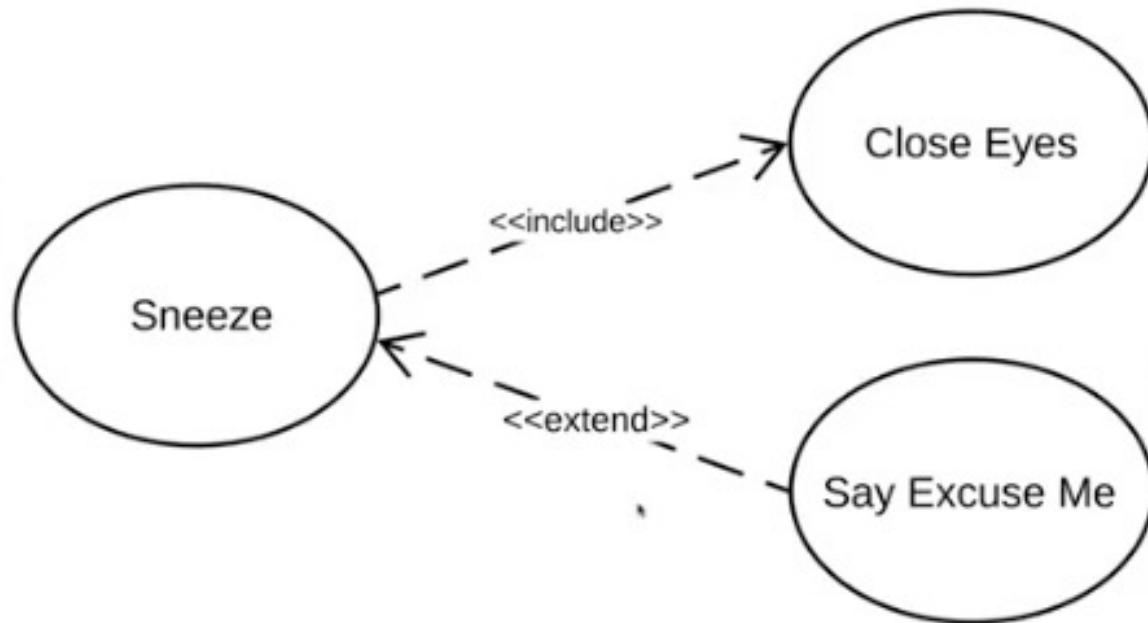


# Extend Relationship

- ▷ Represents the extension of the functionality of the use case to incorporate optional behavior
- ▷ It is not part of the normal flow of events and should be modeled with an extend relationship and an alternate/exception flow



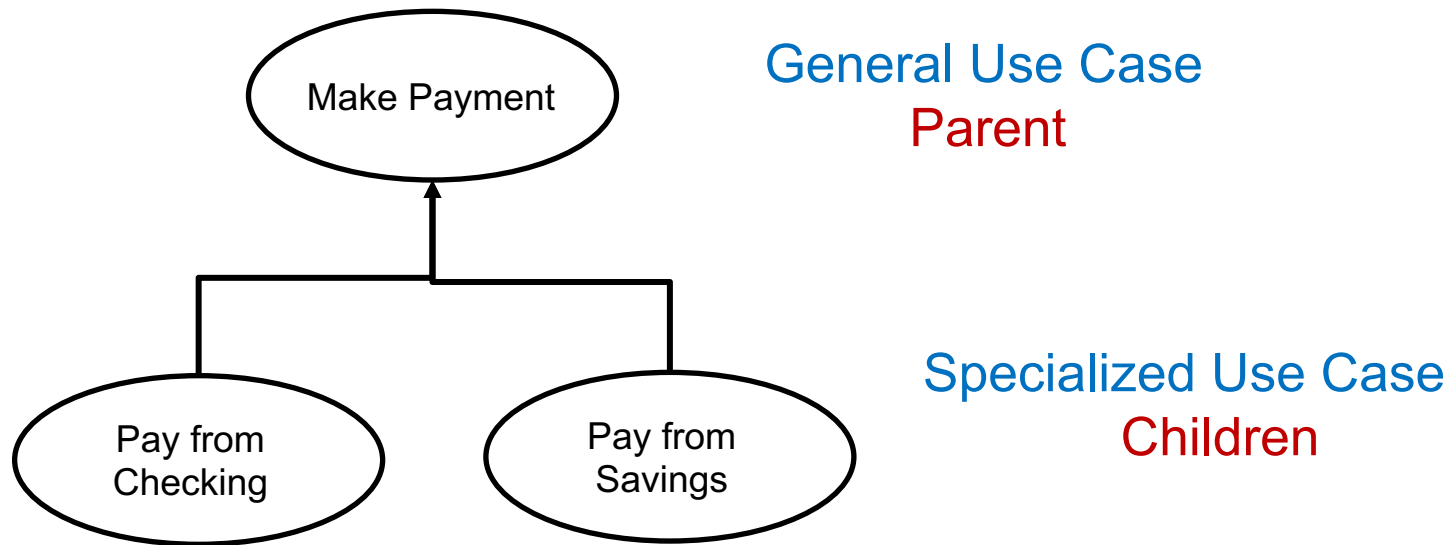


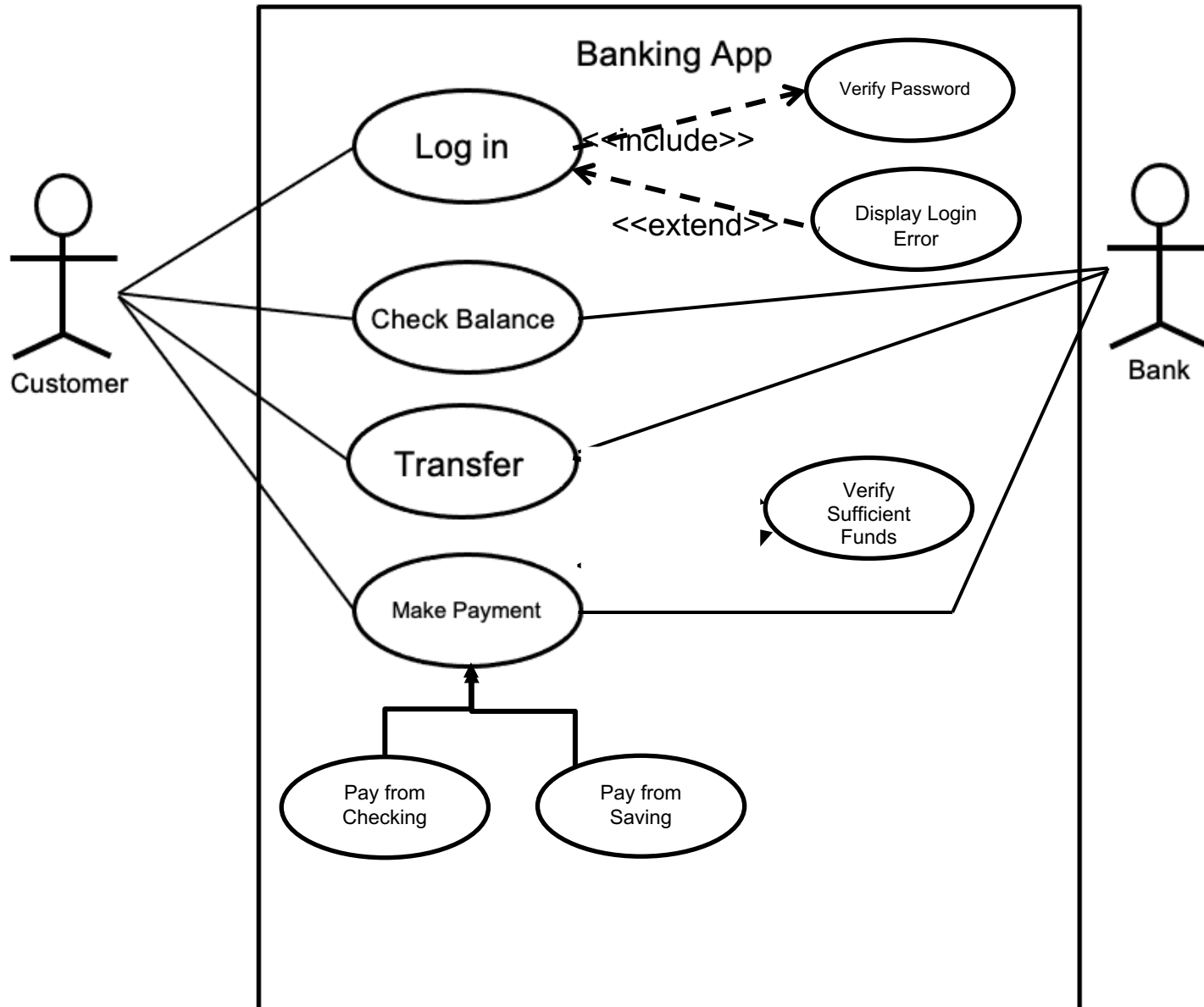


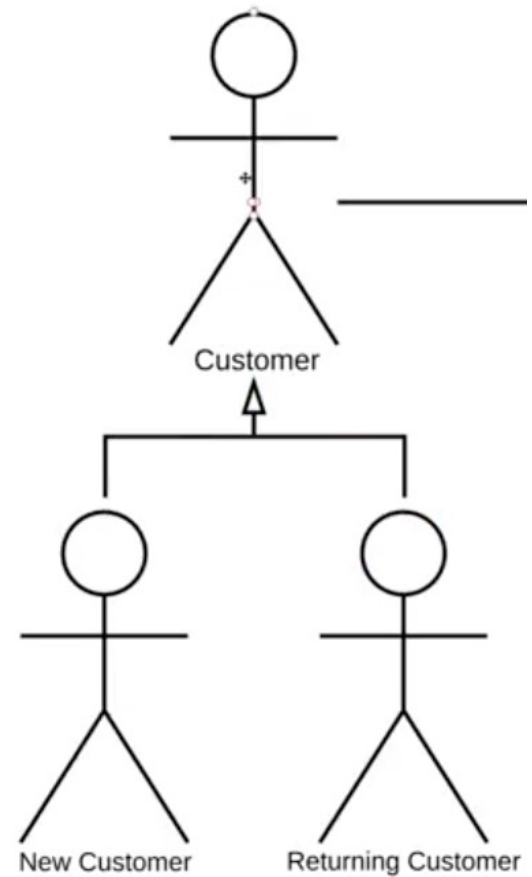
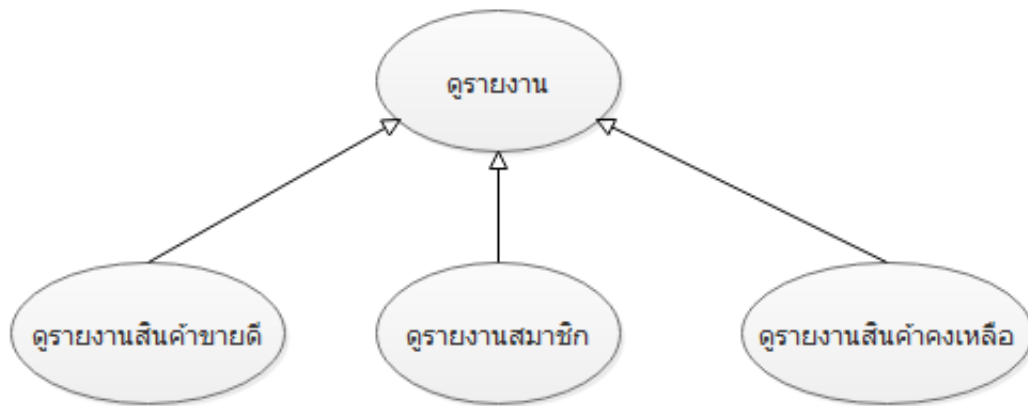


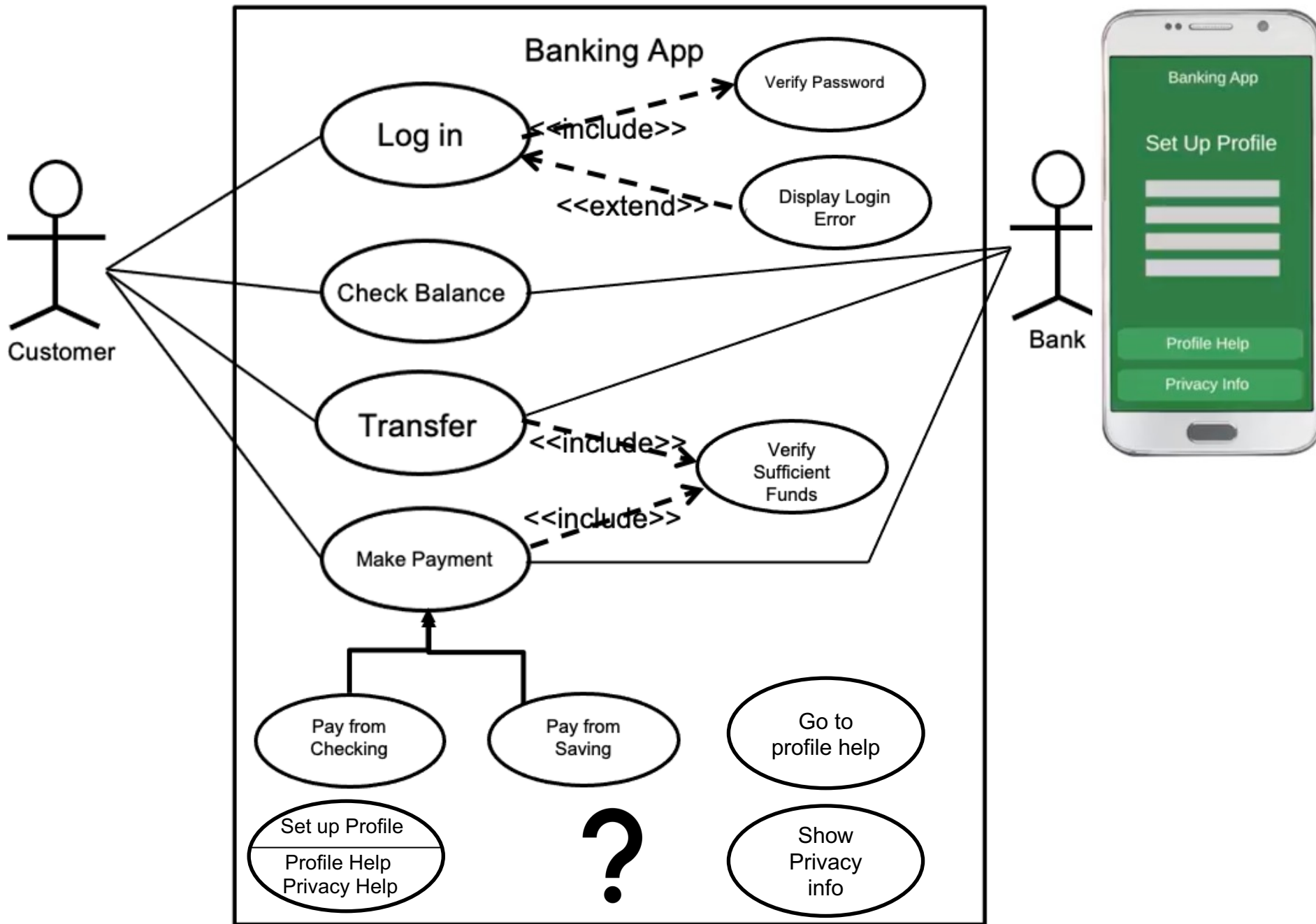
# Generalization Relationship

- ▷ Allows use cases to support *inheritance*









## ตัวอย่างระบบห้องสมุด

- ▷ บรรณารักษ์ให้บริการยืม-คืนหนังสือให้แก่นิสิตได้ โดยที่ถ้ามีการคืนไม่ตรงตามกำหนดจะมีการคิดค่าปรับวันละ 5 บาทต่อเล่ม
- ▷ นิสิตสามารถยืมหนังสือต่อผ่านทางระบบอินเทอร์เน็ต และค้นหาหนังสือที่ต้องการผ่านระบบอินเทอร์เน็ต โดยค้นหาตามชื่อหนังสือ ชื่อผู้แต่ง
- ▷ บรรณารักษ์สามารถบันทึกข้อมูลและเรียกดูข้อมูลของหนังสือใหม่ได้
- ▷ นิสิตสามารถเรียกดูข้อมูลหนังสือใหม่ได้
- ▷ หัวหน้าห้องสมุดสามารถเรียกดูรายงานของหนังสือที่ถูกยืมมากที่สุด รายการของค่าปรับในแต่ละเดือนได้
- ▷ หัวหน้าห้องสมุดสามารถทำฟังก์ชันที่บรรณารักษ์ทำได้

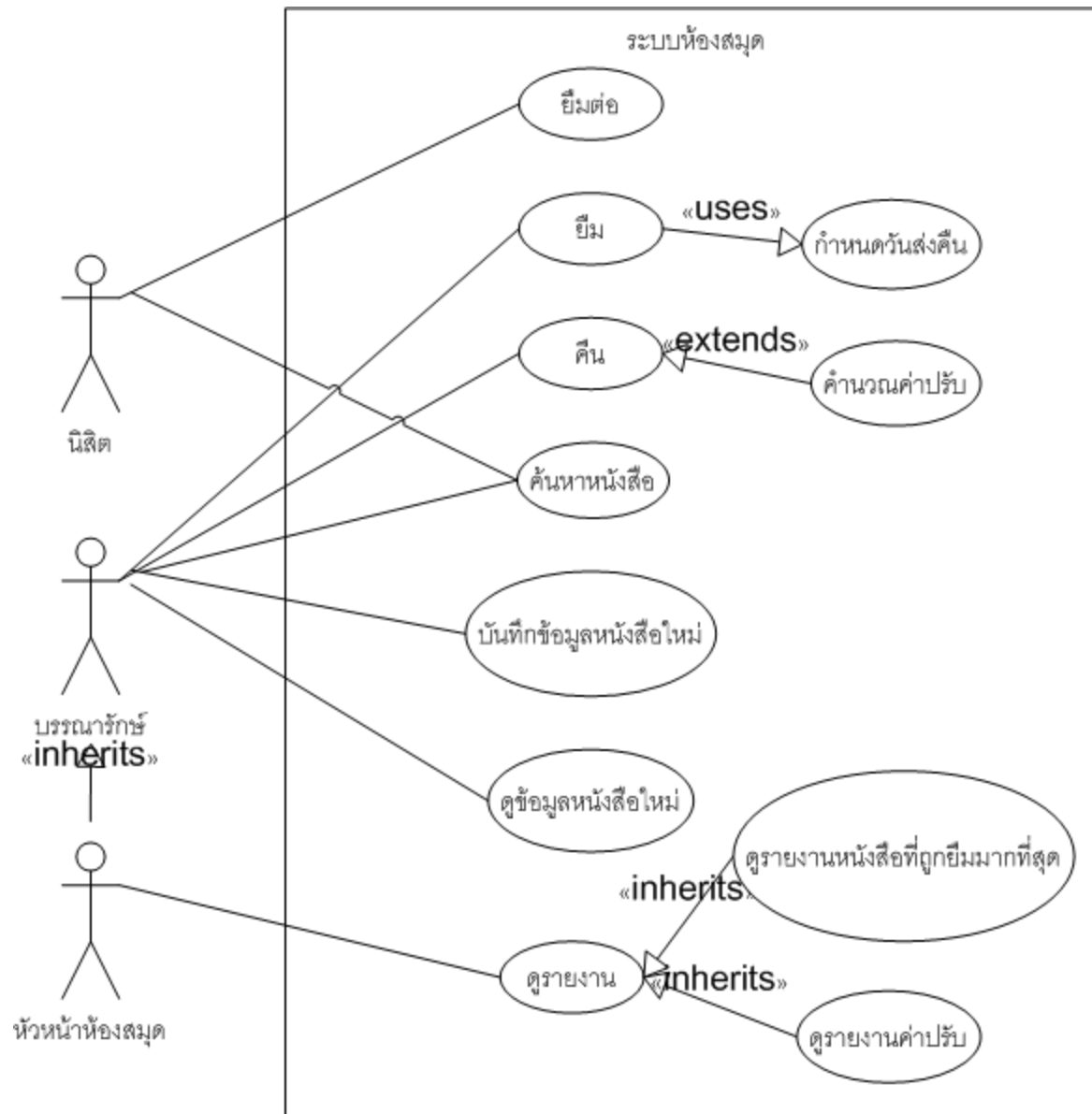
# วิเคราะห์ Actor

- ▷ บรรณารักษ์
- ▷ นิสิต
- ▷ หัวหน้าห้องสมุด

# วิเคราะห์ use case

- ▷ ยืม
- ▷ คืน
- ▷ คิดค่าปรับ
- ▷ ยืมต่อ
- ▷ ค้นหาหนังสือ
- ชื่อหนังสือ
- ชื่อผู้แต่ง
- ▷ บันทึกข้อมูลหนังสือใหม่
- ▷ ดูข้อมูลหนังสือใหม่
- ▷ ดูรายงาน

# Use case Diagram ระบบห้องสมุด





## คำอธิบายยูสเคส (Use Case Description)

- ▷ Use Case Name
- ▷ Participating Actor
- ▷ Entry Condition
- ▷ Flow of events
- ▷ [Alternative]
- ▷ Exit Condition

# ตัวอย่างคำอธิบายยูสเคส (Use Case Description) ยืม

Use Case Name	ยืม
Participating Actor	บรรณารักษ์
Entry Condition	บรรณารักษ์เลือกฟังก์ชันการยืม
Flow of events	<ol style="list-style-type: none"><li>1. บรรณารักษ์ใส่รหัสหนังสือ</li><li>2. บรรณารักษ์ใส่รหัสหนังสือ</li><li>3. กำหนดวันคืน</li></ol>
Exit Condition	ระบบแจ้งว่าทำการบันทึกข้อมูลเรียบร้อยแล้ว

More Information: <https://gla.reru.ac.th/UML/Use-Case-Diagram1.pdf>

Thanks!

**Any questions?**