CSY2030
Systems Design & Development
Use Case Diagrams

Use Cases

- Use Cases define what a proposed system should do from a user's perspective
  - They provide a basis for a contract between the customer and the developer
- Use Cases capture the functional requirements from the requirements document

Use Cases

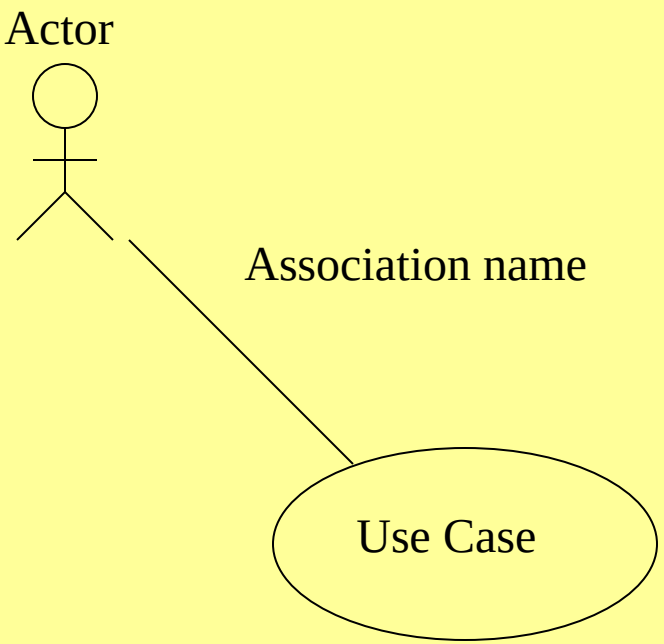
- These are the starting point for modelling and development.
  - it is a way of expressing user requirements.
- Defined as when a user performs “a behaviourally related sequence of transactions in a dialogue with the system”
- Should capture the user goals - not the system functions.
- Describes the external view of the system and its interactions with the outside world

Use Cases

- These are the starting point for modelling and development.
  - it is a way of expressing user requirements.
- Defined as when a user performs “a behaviourally related sequence of transactions in a dialogue with the system”
- Should capture the user goals - not the system functions.
- Describes the external view of the system and its interactions with the outside world

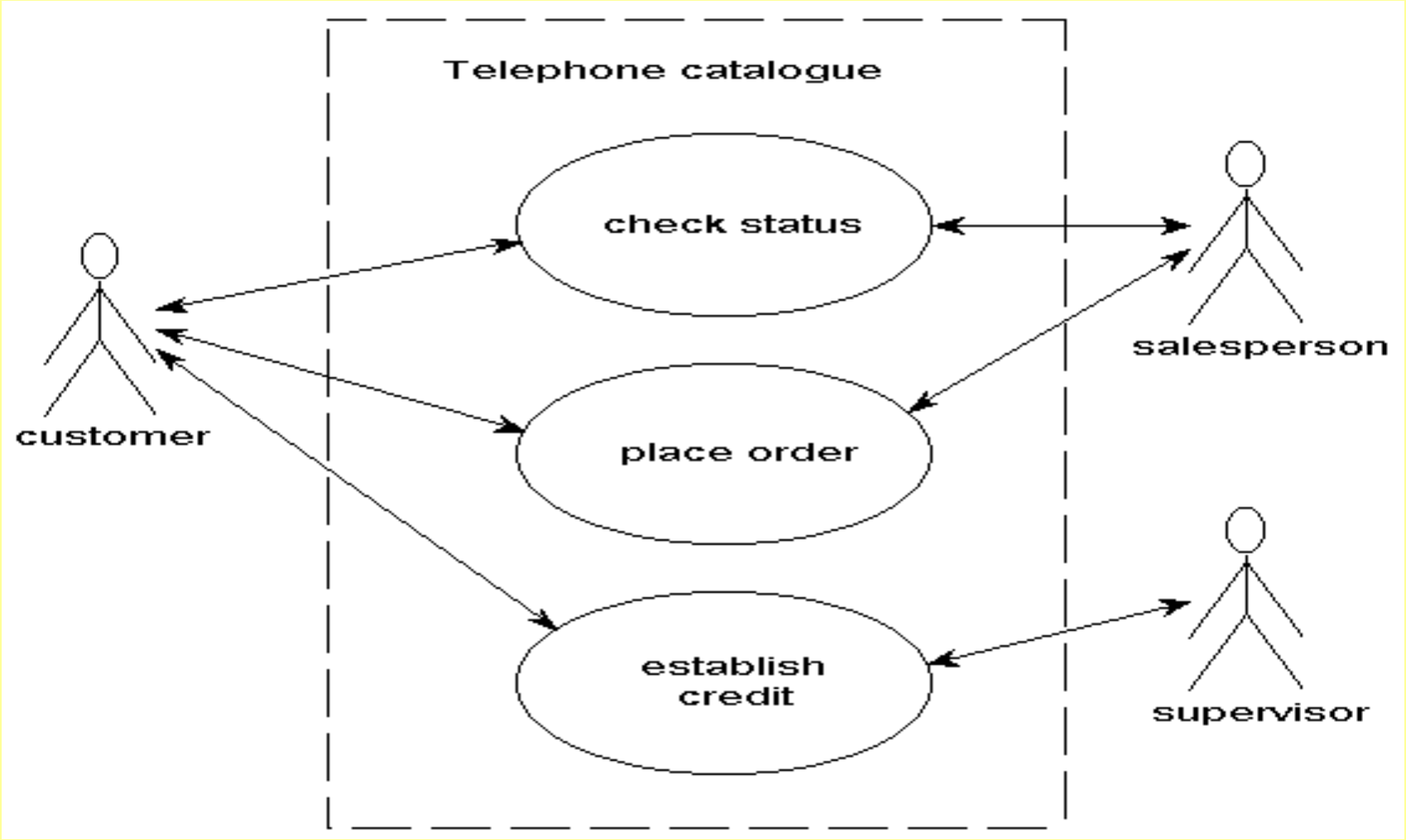
Use Cases

- Outside world is represented by Actors
- Actors are roles played by various people or other systems
  - One person may play many roles
  - A role may have many people playing in it
- Use Case is an interaction an actor has with the system - from the actor point of view
- Can have system boundaries to separate several subsystems



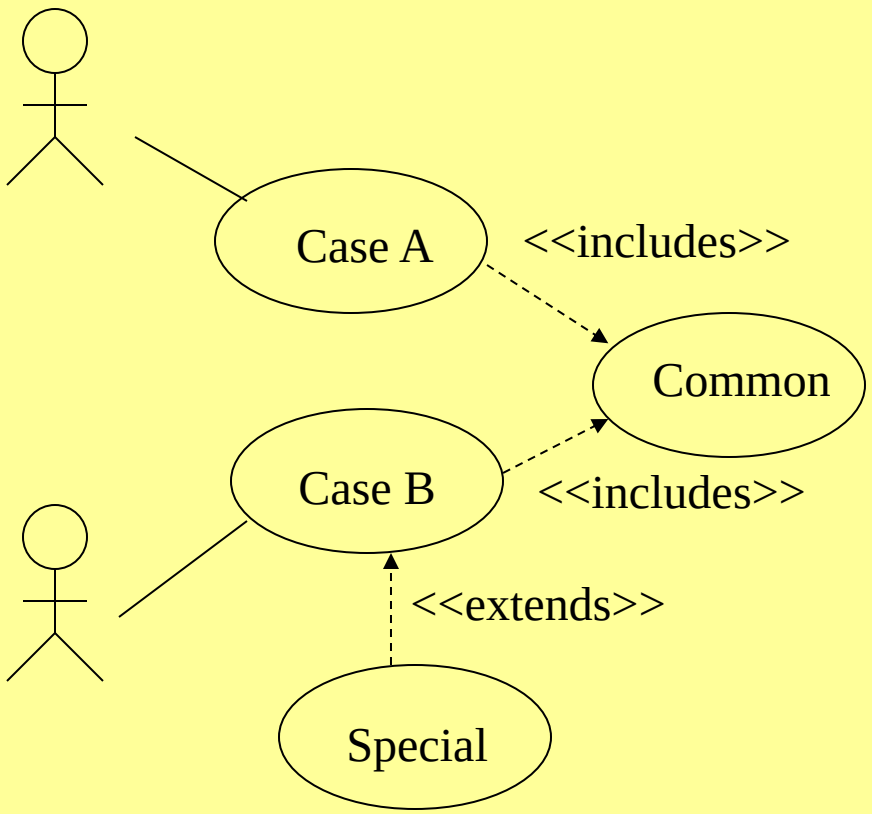
Use Cases

Example



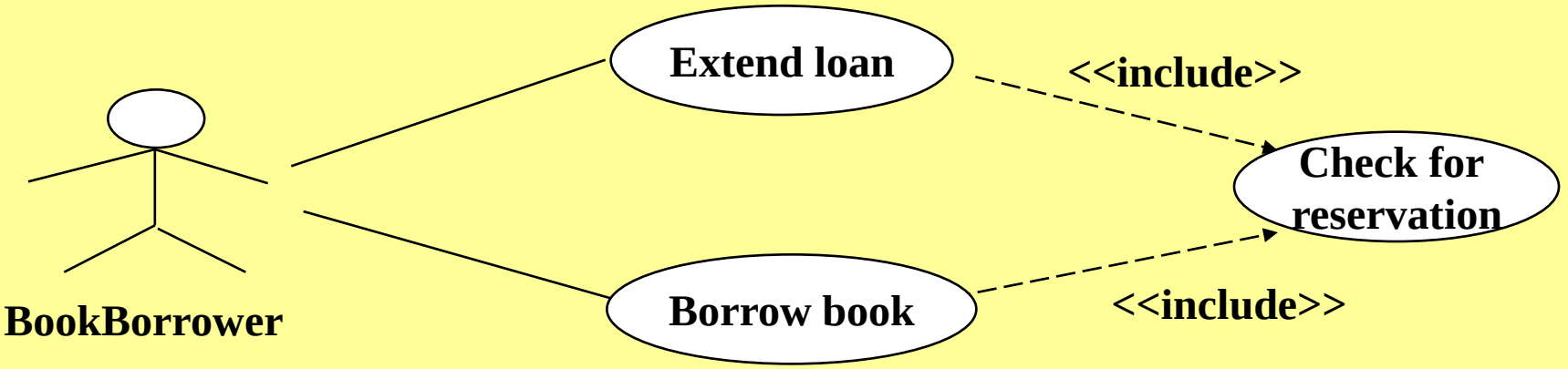
Use Cases - Stereotypes

- In UML you can have special types of relationship a.k.a **stereotypes** i.e **includes** and **extends**
- The **includes** relationship allows common behaviour to pulled out into a separate use case
- **extends** relationship shows behaviour which is exceptional, optional or special

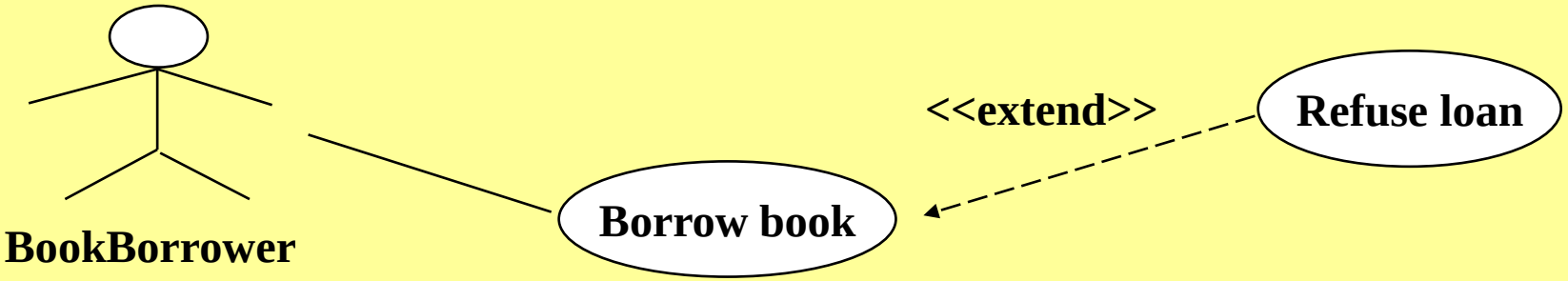


Use Cases - stereotypes

Example - include

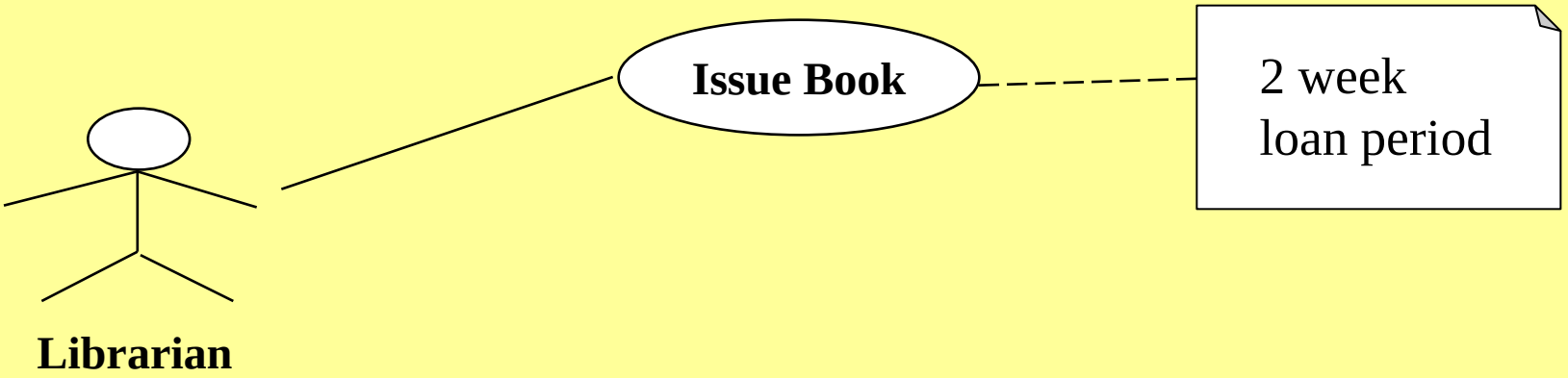


Example - extend



Use Cases - Notes

Example



Complex Use Cases Diagrams

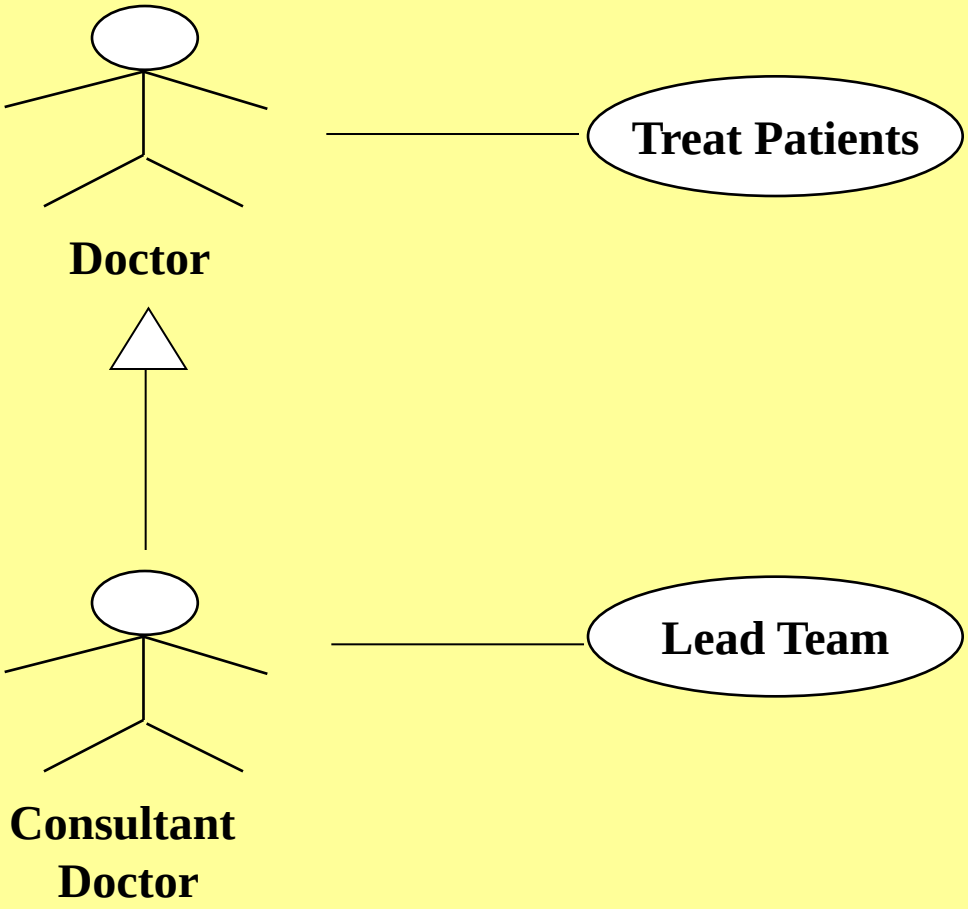
- Use Case Diagrams can be more complex i.e
  - Can have the following inheritance:
    - Actors
    - Use cases
- We shall look at each in turn

Inheritance of Actors

- Some actors can do everything that other actors can do, and more
- Rather than repeating associations you can use inheritance of actors
  - Show specialisation of actors
  - Achieved graphically in the same way as inheritance of classes

Inheritance of Actors cont.

Example

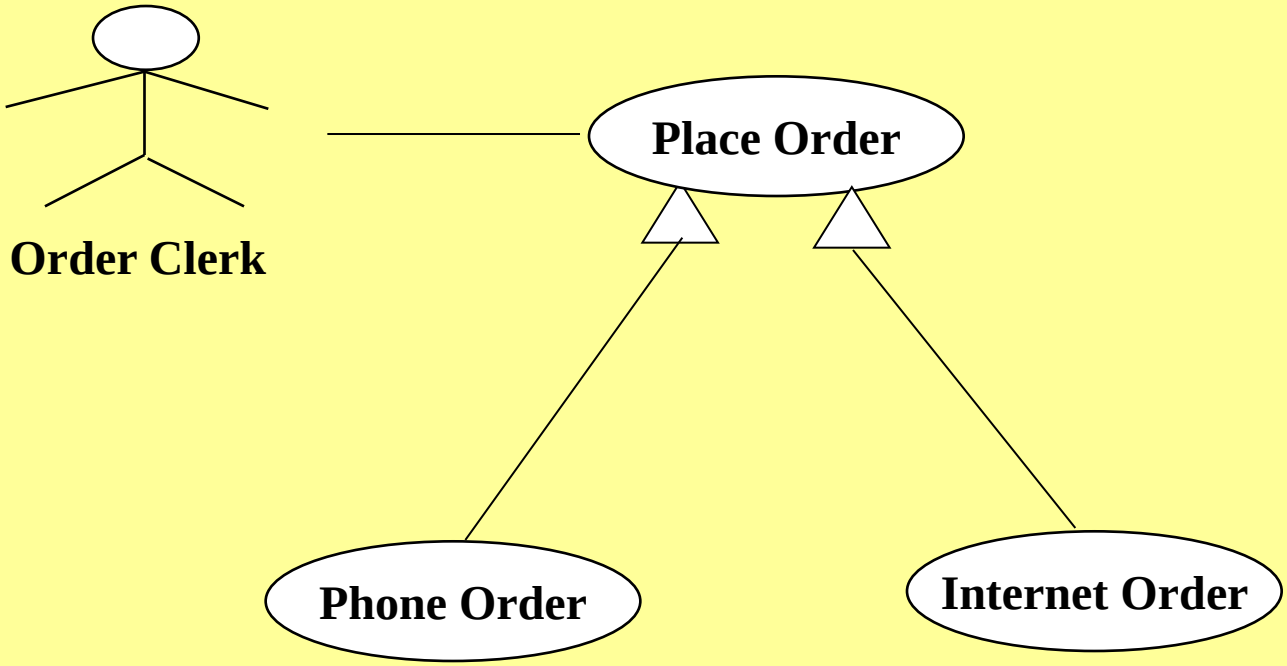


Inheritance of Use Cases

- Some use cases have common functionality with other use cases
  - Need to abstract out the common functionality into a ‘super-use case’ and use inheritance for specific use cases
  - Achieved graphically in the same way as inheritance of classes

Inheritance of Use Cases cont.

Example



Documenting Use Cases

- As well as the use case diagram we need to document use cases textually
  - This will us to transform the user requirements into more detailed software requirements
  - We propose the following template – see next slide

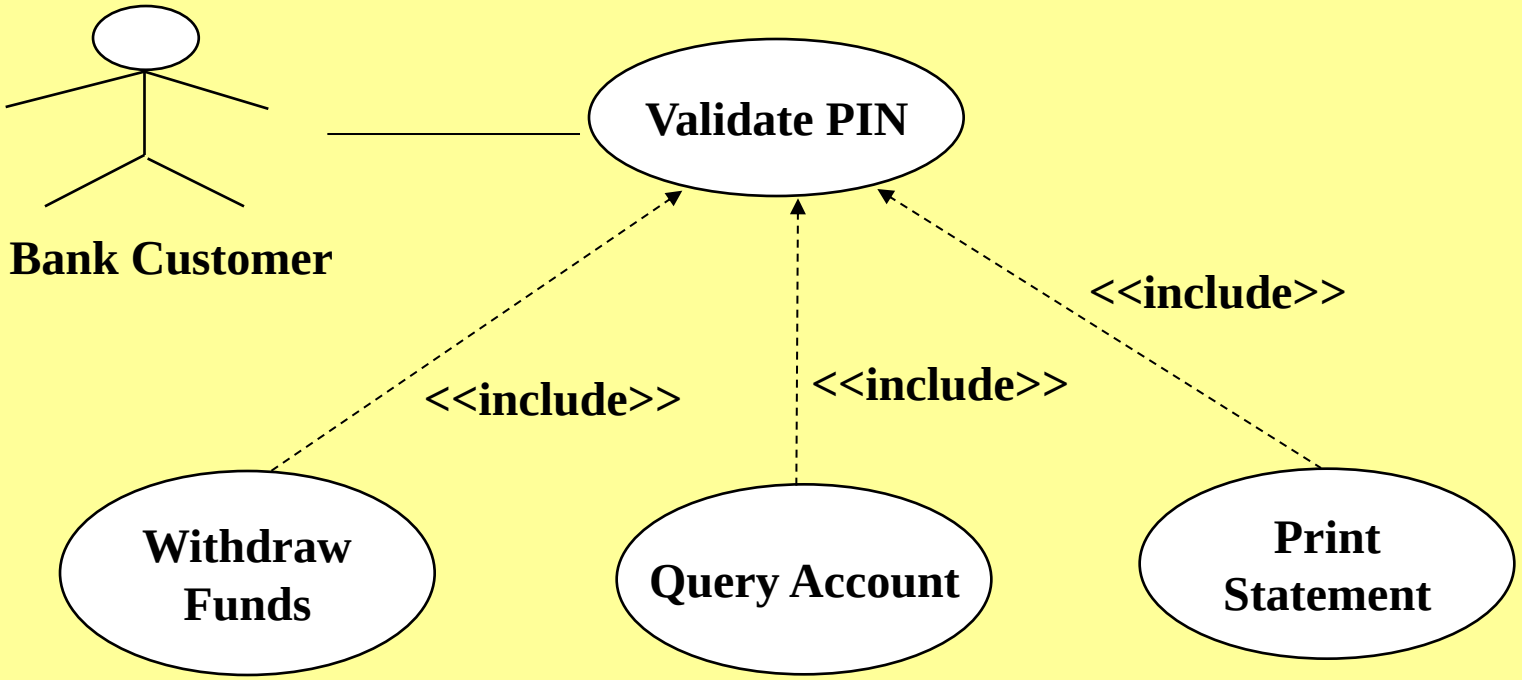
Documenting Use Cases cont.

Identifier and name: <state use case id and use case>
Initiator: <state actor who initiates use case>
Goal: <state what use case aims to achieve>
Pre-condition: <state conditions to be satisfied prior to carrying out this use case>.
Post-condition: <state conditions to be satisfied after carrying out this use case>.
Assumptions: <state what assumptions you make about use case>

Main success scenario: <state series of steps that demonstrates the success of the use case>
Extensions or Includes: <state any extended or included behaviour>

Documenting Use Cases cont.

Example



Documenting Use Cases cont.

Identifier and name: UC1 Validate PIN
Initiator: Bank Customer
Goal: Bank Customer accesses his/her account
Pre-condition: ATM is idle, displaying a "Welcome" message
Post-condition: Bank Customer performed desired actions on his/her account
Assumptions: Bank Customer is a valid customer of the Bank

Main success scenario:

1. Customer inserts his/her ATM card into reader
2. If system recognises card, it reads the card number
3. System prompts customer for PIN
4. Customer enters PIN
5. System checks for expiration date and whether lost or stolen
6. If valid then system check the PIN number
7. If PIN valid then system offers customer to withdraw, query or print statement

Includes: *Withdraw Funds*, *Query Account* or *Print Statement*