# Lesson-5: Use Cases

CS 438
Ali Aburas PhD

# Today’s Goals

- What is a use case?
- Introduce use cases and their role in brainstorming and explaining requirements.

# Use Cases

A use case is used to describe a scenario (i.e., an activity/interaction) between a system and external agents (human users or other systems)

A use case

– is the primary modeling tools defining the behavior of the system

– describes how the user interacts with the system to perform some activity, such as placing an order, making a reservation, or searching for information

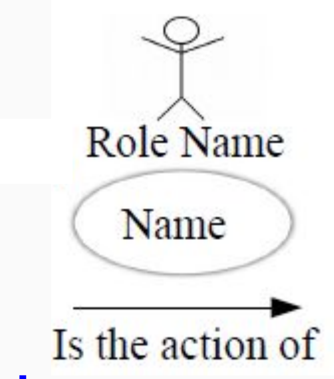
– represents a dialog between a user and the system, from the user’s point of view

– Is used to identify and to communicate the requirements for the system to the programmers who must write the system

# What is a use case?

- A use case is a written description of a user's interaction with the software system to accomplish a goal.
  - A use case is used to describe a scenario (i.e., an activity/interaction) between a system and external agents (human users or other systems)
    - It is an example behavior of the system
    - Written from an actor's point of view, not the system's
    - It describes how the user interacts with the system to perform some activity, such as placing an order, making a reservation, or searching for information
    - 3-9 clearly written steps lead to a "main success scenario"

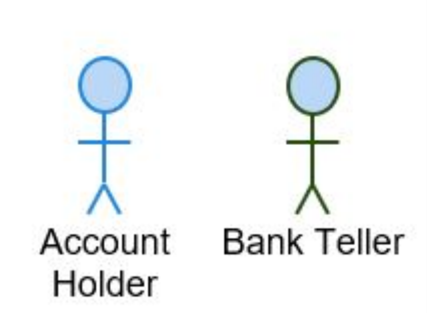
- **Terminology**
  - Actor: someone (or another system) interacting with the system
    - Stick man for user or another system
  - Goal: Ovals for use cases
    - desired outcome of the primary actor
  - Flow: interactive steps to achieve the goals.
    - Simple arrows when a use case "calls" another

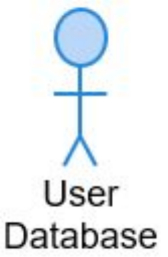


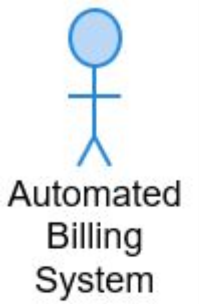
**Use cases capture functional requirements of a system!**

# What is an Actor?

- An **actor** is a role a user plays with respect to the system.
    - Actors carry out use cases. An actor can perform many use cases. A use case can involve multiple actors.
    - A single user can be multiple actors, depending on how they use a system.
    - Actors do not need to be human - can be an external system (hardware or software) that interacts with the system being built.

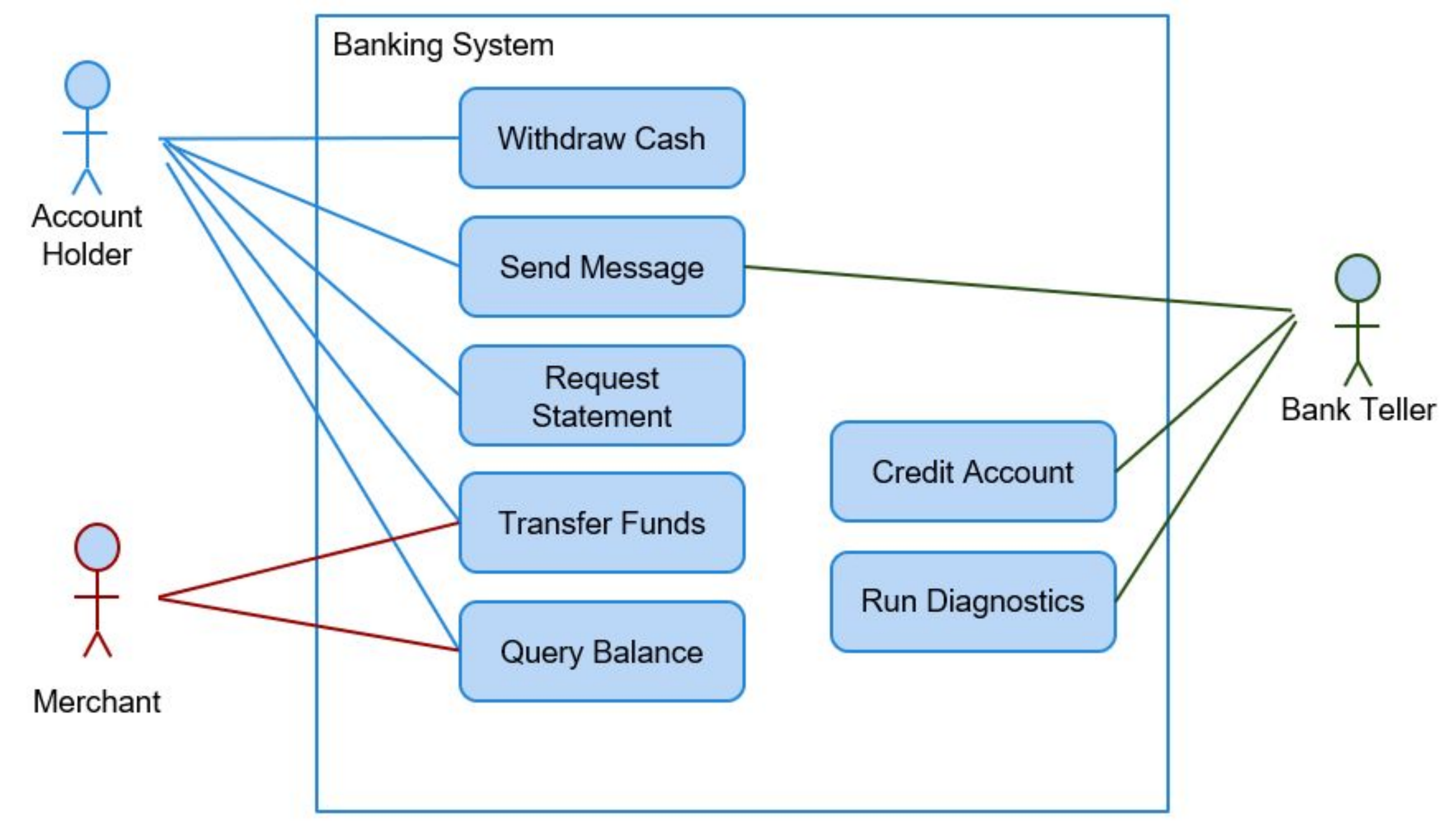




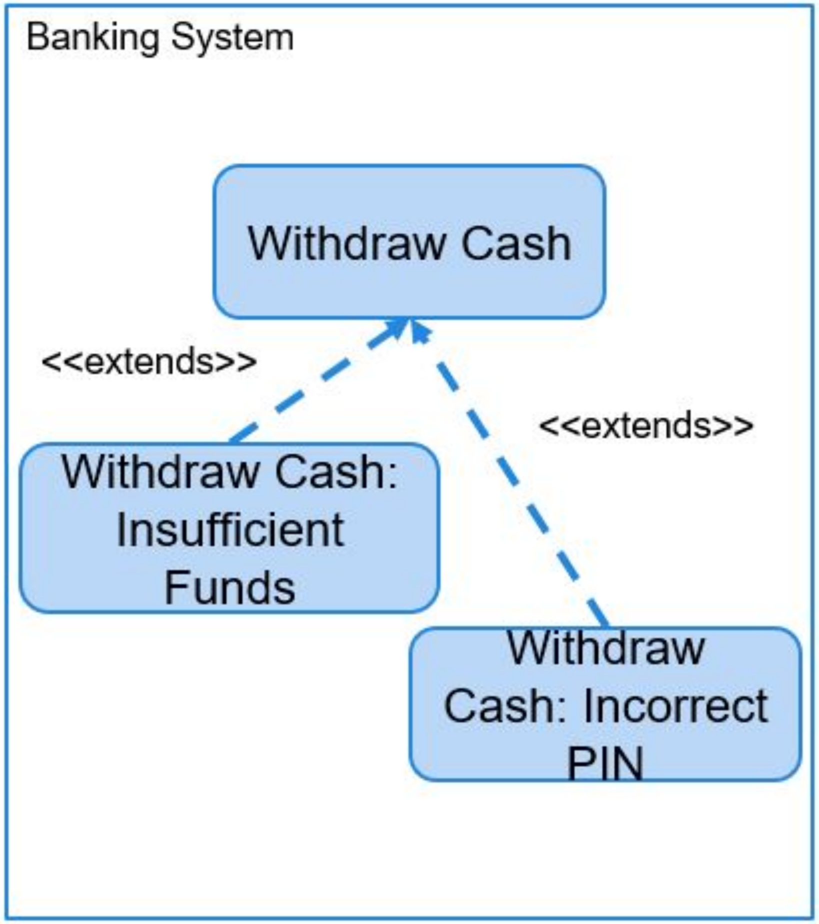


Manager

# Online Banking Use Case Diagram



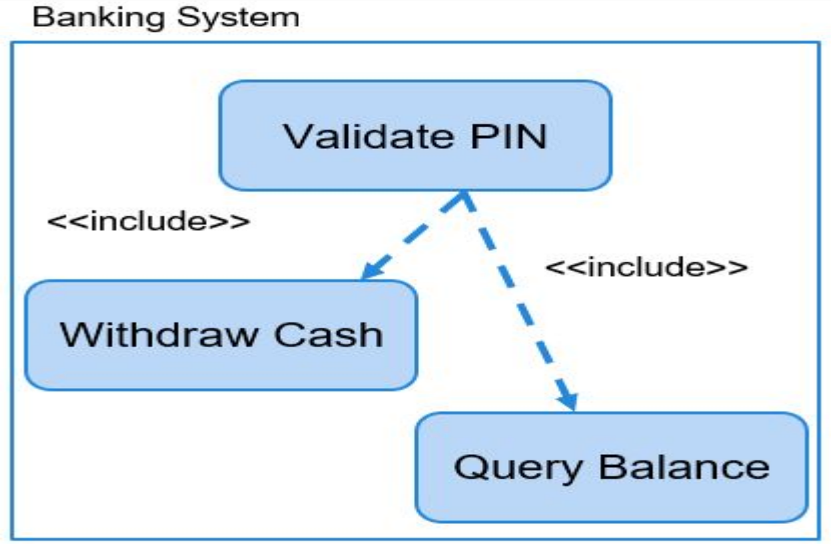
# Use Case Relationships: Extends

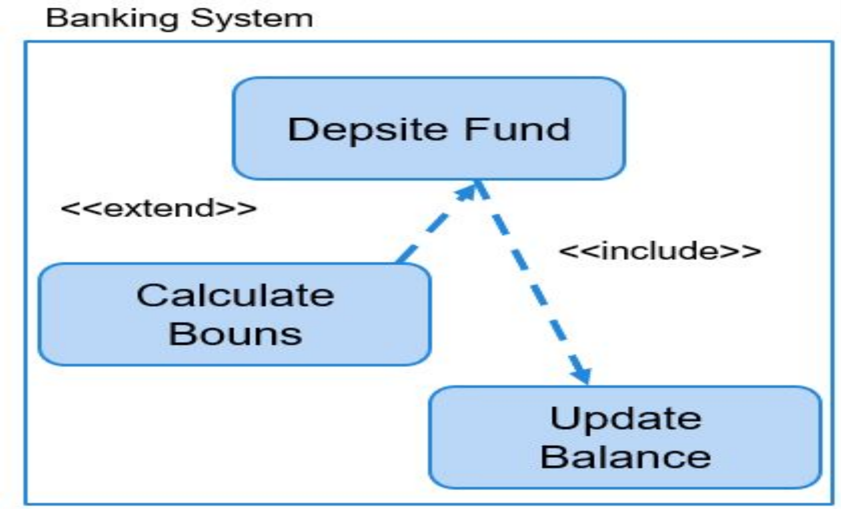

Banking System
Withdraw Cash
<<extends>>
<<extends>>
Withdraw Cash: Insufficient Funds
Withdraw Cash: Incorrect PIN

## Extends

- A use-case is similar to another, but does more or takes an alternate path.
- Put the normal behavior in one use-case and the exceptional behavior somewhere else:
  - Capture the normal behavior.
  - Try to figure out what went wrong in each step.
  - Capture the exceptional cases in separate use-cases
- Allows for easier to understand descriptions.

# Use Case Relationships: Uses \ Include

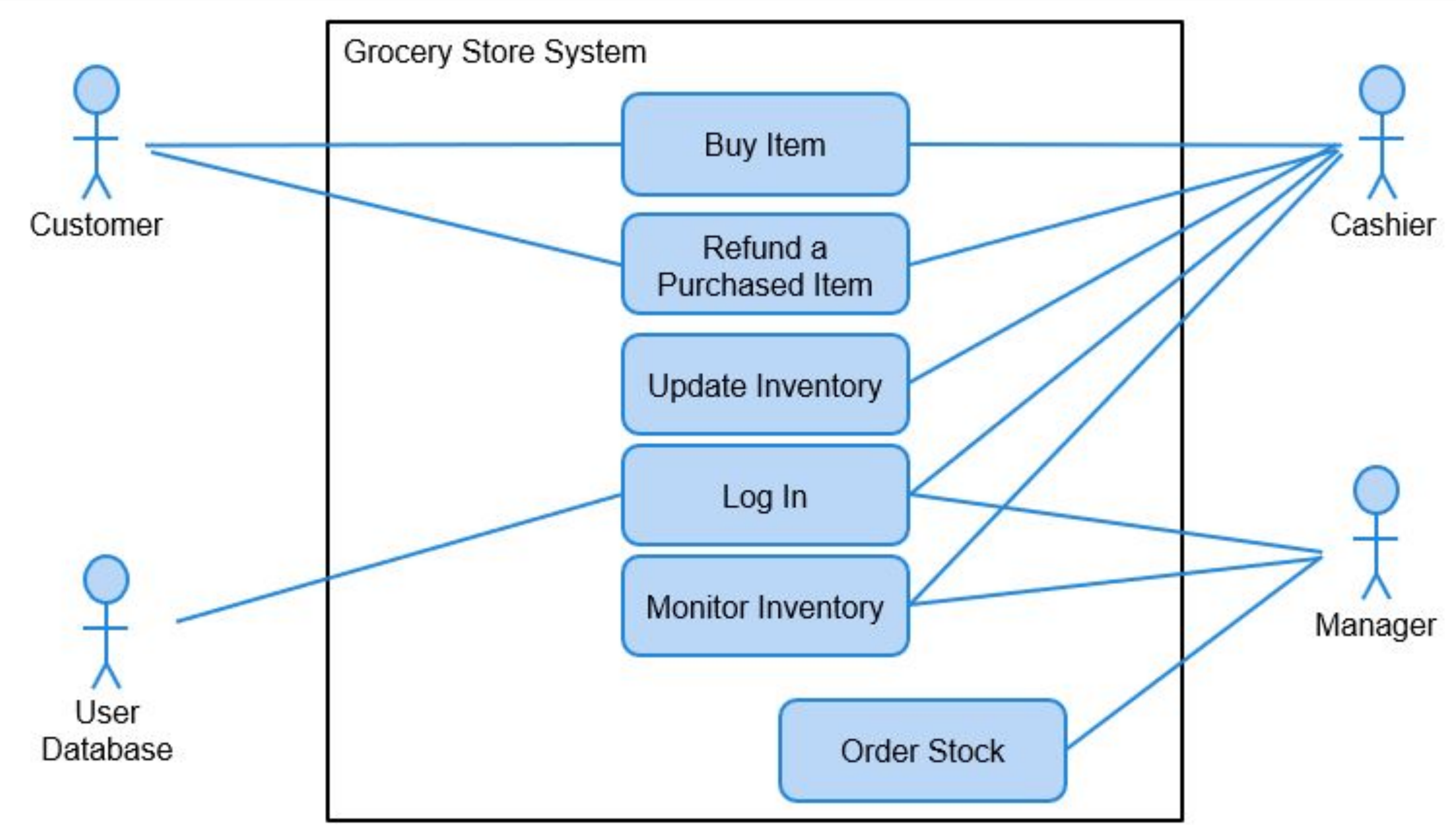




## Include

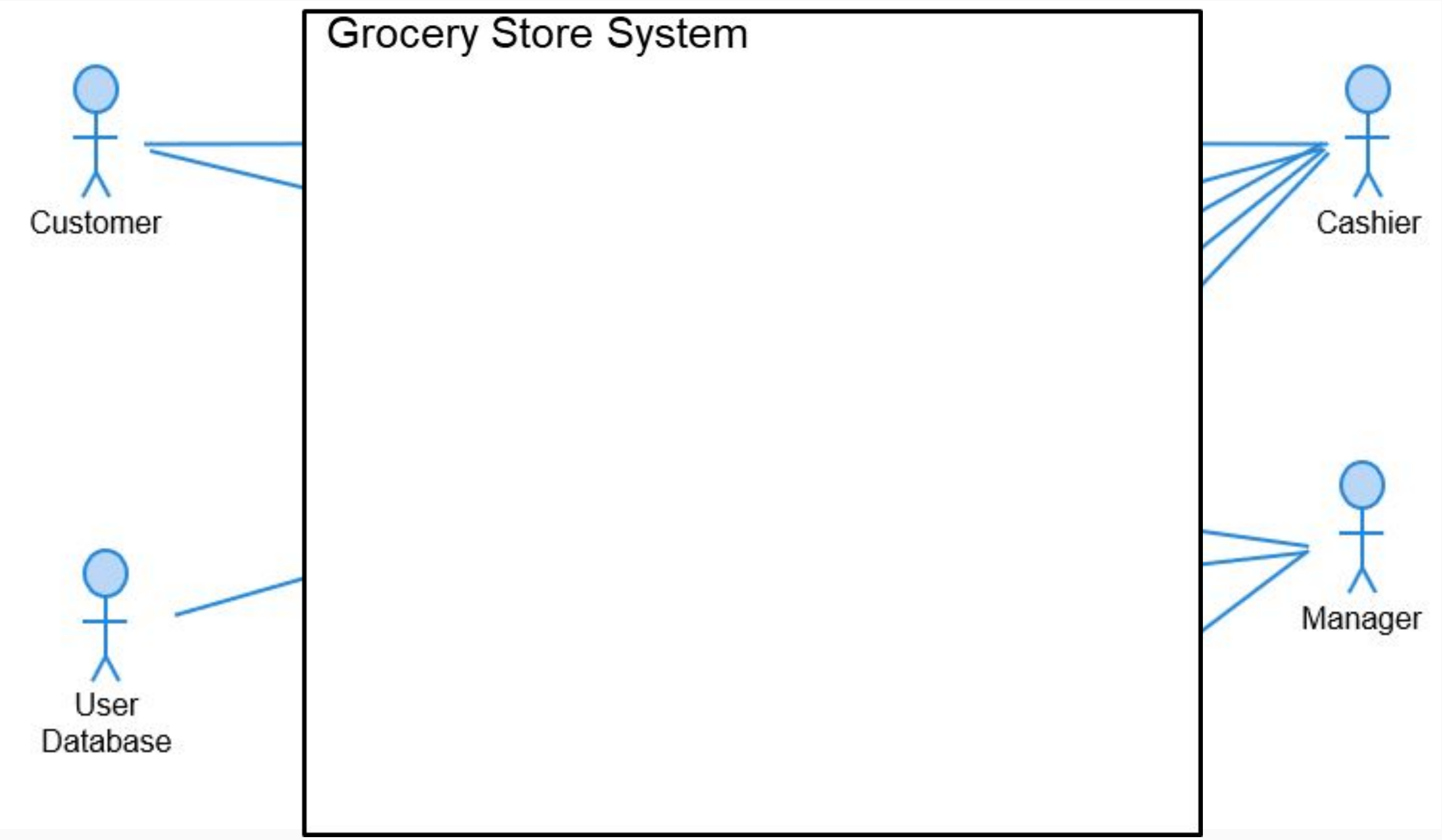
- You have a piece of behavior that is similar across many use-cases.
- Break this out as a separate use-case and let the others “use” it.
- Avoids repetition in written use-cases.
  - Step 1: Complete “Validate PIN” use-case.
  - Step 2: Select account.
  - ...

# Grocery Store System Diagram



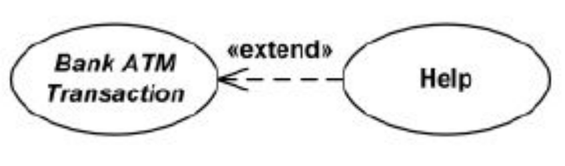
# Setting the System Boundary

- The system boundary will affect your actors and use-cases.

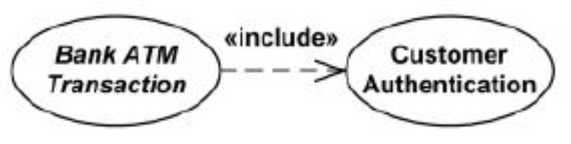


# Use Case Diagrams: Creation Tips

- Use name that communicate purpose
- Define one atomic behavior/scenario/activity supported by the system per use case
- Define flow of events clearly, and provide only essential details.
- Use case must be complete and understandable.
- Use case uses **relationships** for reducing the complexity in the use case model
  - **Extend Dependency**: Extending use case is optional.



  - **Include Dependency**: Included use case required, not optional.

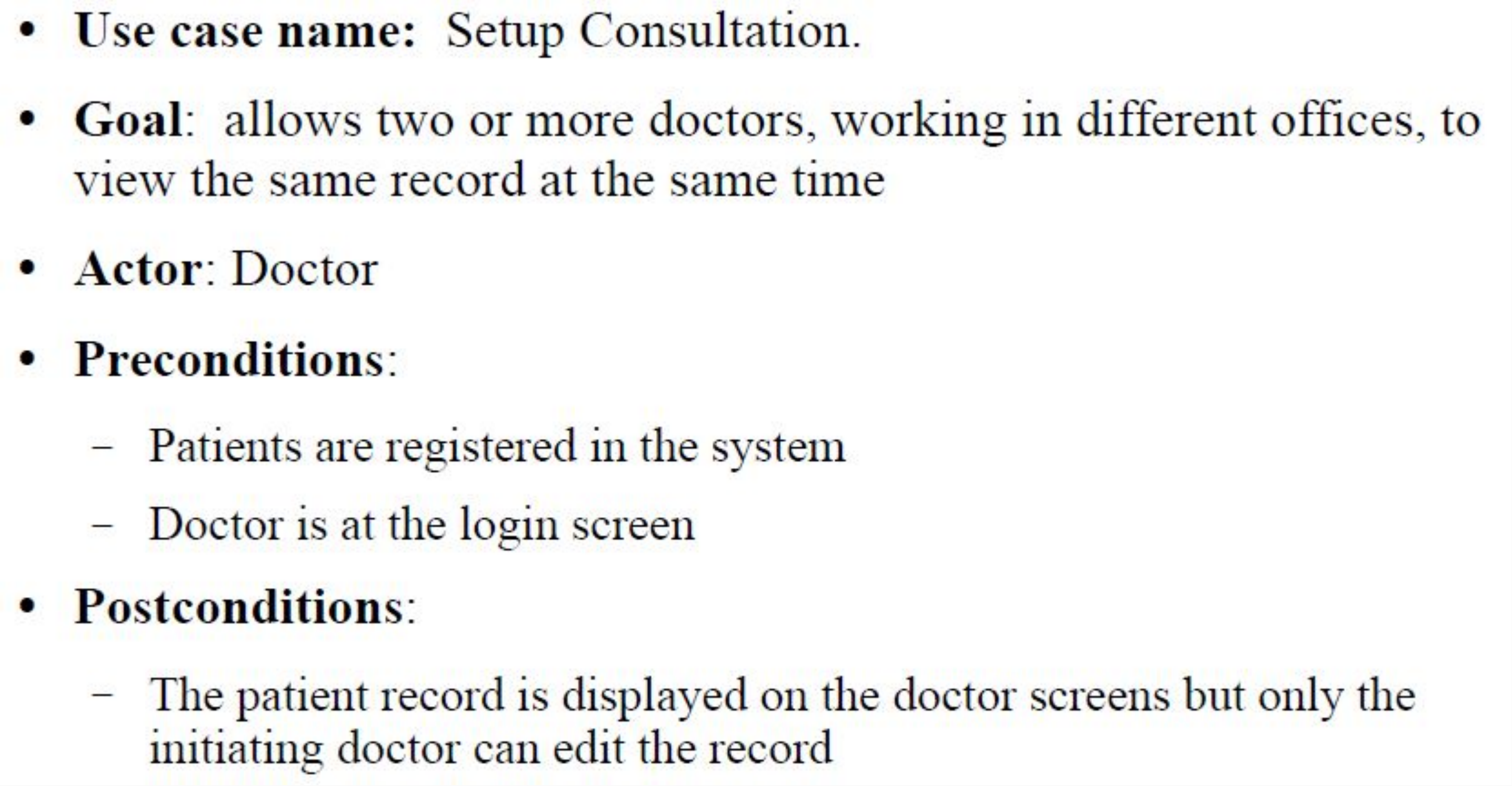


# Use Case Descriptions

Use case diagrams give a simple overview of an interaction,  and you need to add more detail for **complete interaction description**

1. **Use case name**: succinct and meaningful
2. **Goal Level**: a brief description of the goal the use case is supposed to deliver
3. **Actor**: who “does” the activity?
   - Can be human, device, another system.
   - May appear in more than one use case.
4. **Preconditions**: what conditions must be met before the scenario can start
5. **Postconditions**: what conditions must be met for a valid end of the scenario
6. **Flow of events**: what steps do the actor and the system perform during the scenario?

# (Example) Use case: Setup Consultation

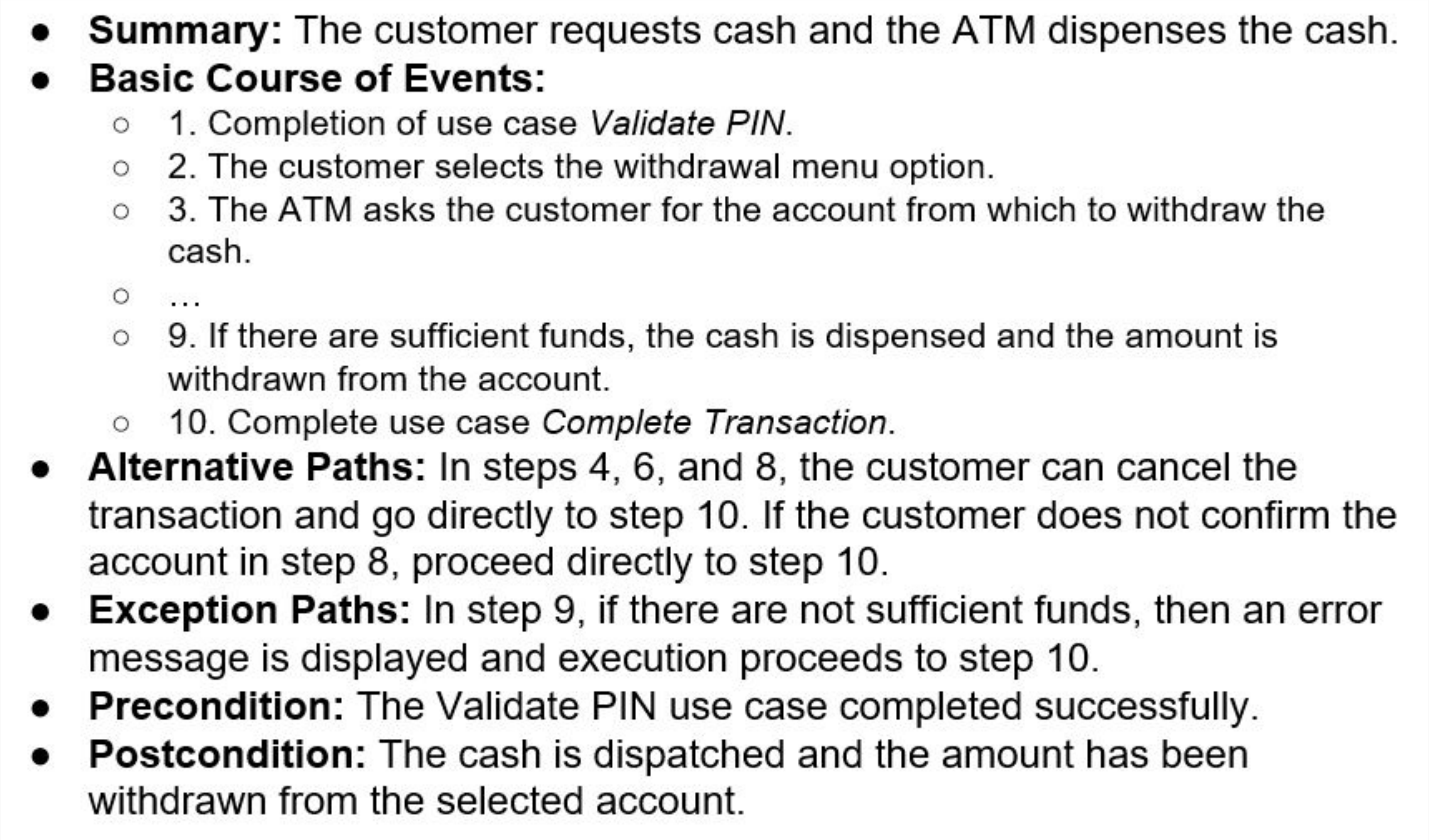


- **Use case name:** Setup Consultation.
- **Goal**: allows two or more doctors, working in different offices, to view the same record at the same time
- **Actor**: Doctor
- **Preconditions**:
  - Patients are registered in the system
  - Doctor is at the login screen
- **Postconditions**:
  - The patient record is displayed on the doctor screens but only the initiating doctor can edit the record

# Use Case Templates

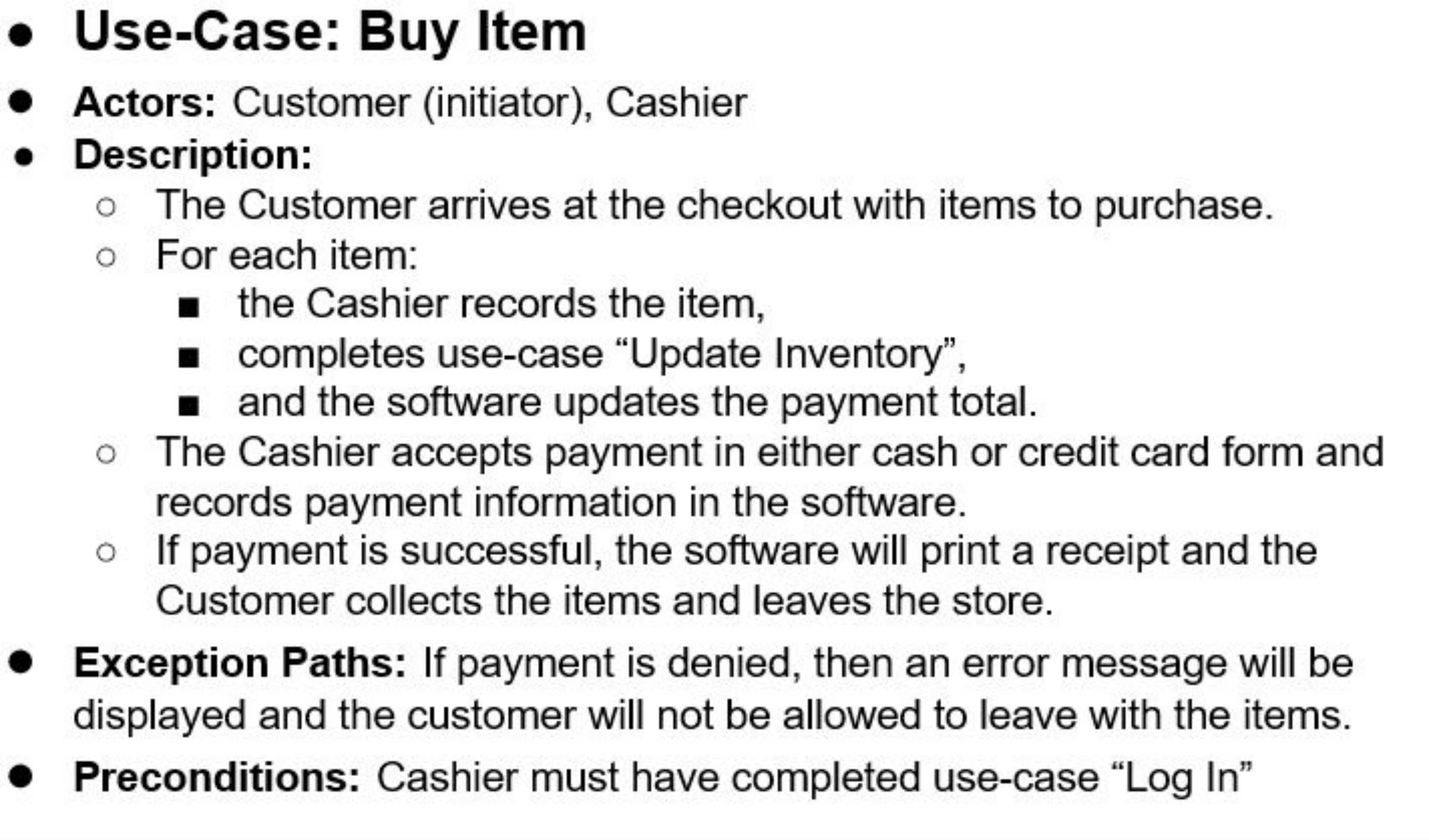
- **Identifier:** Unique ID number
- **Iteration:** Version number
- **Summary:** User goal being fulfilled
- **Actors:** What users/databases/external systems are involved?
- **Basic Course of Events:** Sequence of user interactions.
- **Alternative Paths:** Alternate sequences of events that stem off from certain points.
- **Exception Paths:** Error sequences that stem off from certain points.
- **Extension Points:** Use-cases that resume from the end of this use-case.
- **Trigger:** Rationale, what causes this interaction sequence to begin.
- **Assumptions:** Constraints assumed on this use-case.
- **Precondition:** Conditions that must hold for this use-case to take place, may list other use-cases that need to be completed first.
- **Postcondition:** Side-effects of this use-case.
- **Author:** Who wrote this use-case.
- **Date:** When was this last modified?

# Withdraw Cash (In Sample Template)



- **Summary:** The customer requests cash and the ATM dispenses the cash.
- **Basic Course of Events:**
  - 1. Completion of use case *Validate PIN*.
  - 2. The customer selects the withdrawal menu option.
  - 3. The ATM asks the customer for the account from which to withdraw the cash.
  - …
  - 9. If there are sufficient funds, the cash is dispensed and the amount is withdrawn from the account.
  - 10. Complete use case *Complete Transaction*.
- **Alternative Paths:** In steps 4, 6, and 8, the customer can cancel the transaction and go directly to step 10. If the customer does not confirm the account in step 8, proceed directly to step 10.
- **Exception Paths:** In step 9, if there are not sufficient funds, then an error message is displayed and execution proceeds to step 10.
- **Precondition:** The Validate PIN use case completed successfully.
- **Postcondition:** The cash is dispatched and the amount has been withdrawn from the selected account.

# Grocery Store System Use Case



- **Use-Case: Buy Item**
- **Actors:** Customer (initiator), Cashier
- **Description:**
  - The Customer arrives at the checkout with items to purchase.
  - For each item:
    - the Cashier records the item,
    - completes use-case "Update Inventory",
    - and the software updates the payment total.
  - The Cashier accepts payment in either cash or credit card form and records payment information in the software.
  - If payment is successful, the software will print a receipt and the Customer collects the items and leaves the store.
- **Exception Paths:** If payment is denied, then an error message will be displayed and the customer will not be allowed to leave with the items.
- **Preconditions:** Cashier must have completed use-case "Log In"

# When should we use Use Cases?

- **In short… Always!**
- **Requirements specification is the hardest part of software development. Use cases are a powerful tool to understand**:
    - Who your users are (including non-human systems).
    - What functions your system should provide.
    - How these functions work (at a high level).

I'LL DESIGN THE SYSTEM AS SOON AS YOU GIVE ME THE USER REQUIREMENTS.
www.dilbert.com scottadams@aol.com
BETTER YET, YOU COULD BUILD THE SYSTEM, THEN I'LL TELL YOUR BOSS THAT IT DOESN'T MEET MY NEEDS.
3/21/03 © 2003 United Feature Syndicate, Inc.
I DON'T MEAN TO FRIGHTEN YOU, BUT YOU'LL HAVE TO DO SOME ACTUAL WORK.
THAT'S CRAZY TALK.