Software Engineering

Use Cases

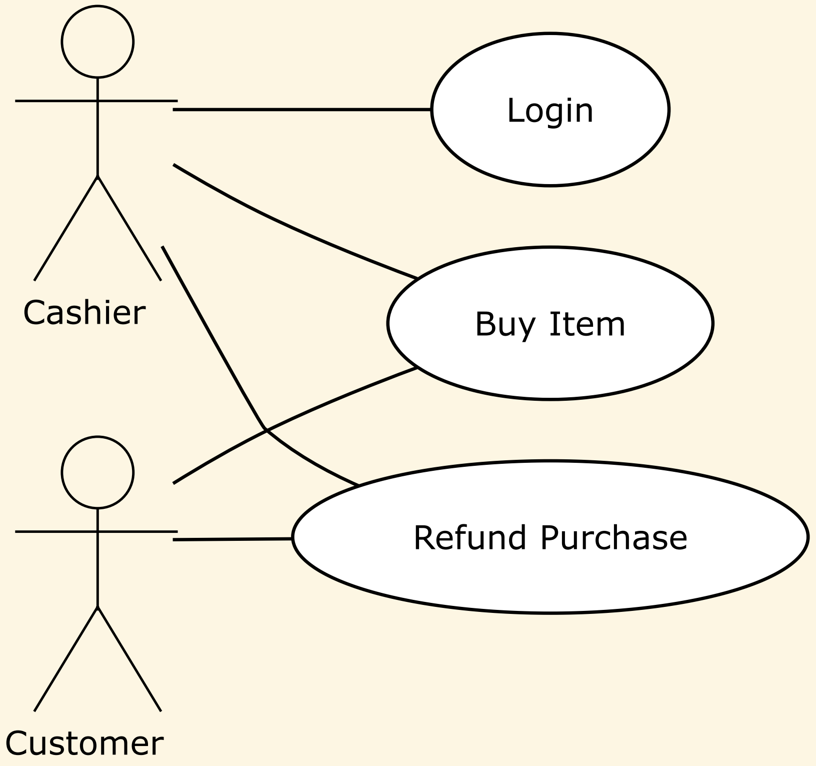
Michael L. Collard, Ph.D.

Department of Computer Science, The University of Akron

Requirements

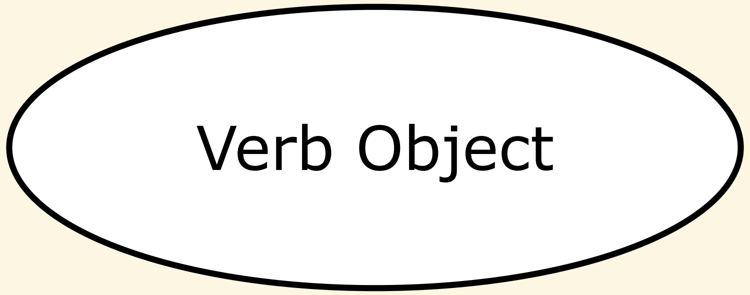
- Previously stated as *User Stories*
- Describes what the system is supposed to do
- Often is given in English (i.e., not in a formal language)
- If a system does not meet the actual requirements, then it has failed
- *Requirements Engineering*, *Requirements Elicitation*

UML Use Case Diagrams



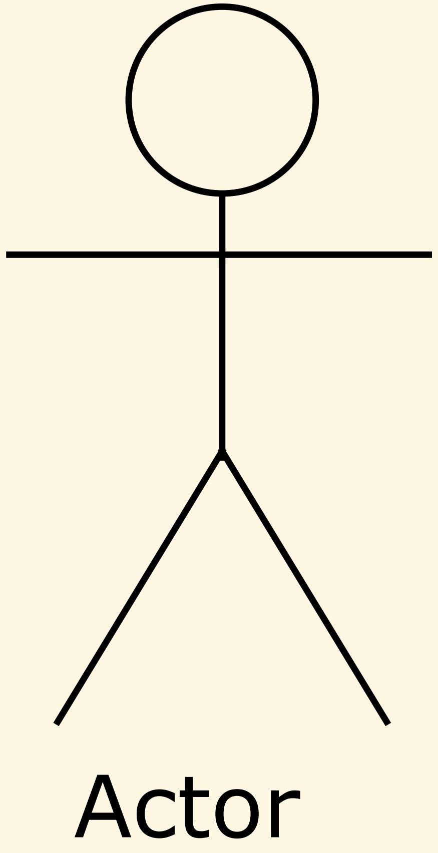
- Description of a system's behavior as it responds to a request that originates from outside of that system
- Describes a set of sequences
- Each sequence represents the interactions of things outside the system (actors) with the system itself (and fundamental abstractions)
- Use cases represent the functional requirements of the system (must state non-functional requirements elsewhere)

Use Case



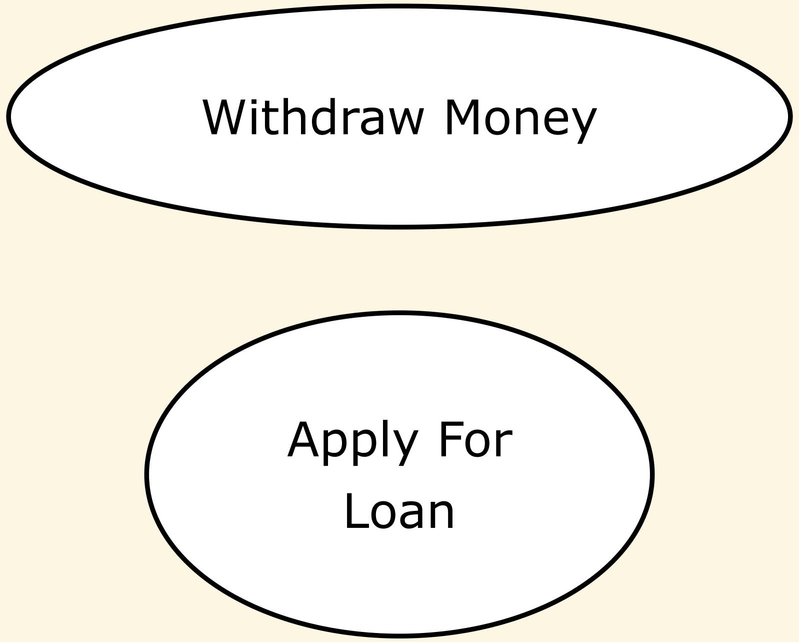
- Each use case has a descriptive name
- Describes what a system does but not how it does it.
- Names must be unique within a given package

Actor



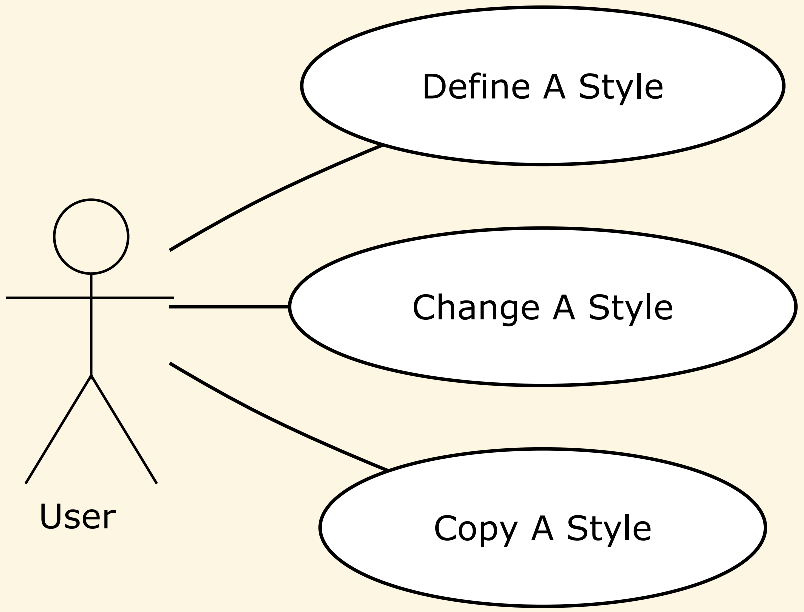
- Actors have a name
- An actor is a set of roles that users of use cases play when interacting with the system
- They are external entities, e.g., people, other systems
- Examples: Customer, Loan officer

What is a Use Case?



- Use case captures some user-visible functionality
- The granularity of functionality depends on the level of detail in your model
- Each use case achieves a discrete goal for the user
- Use Cases are generated through requirements elicitation

Goal vs. Interaction



- Goal - something the user wants to achieve

  Format a document

  Ensure consistent formatting of two documents

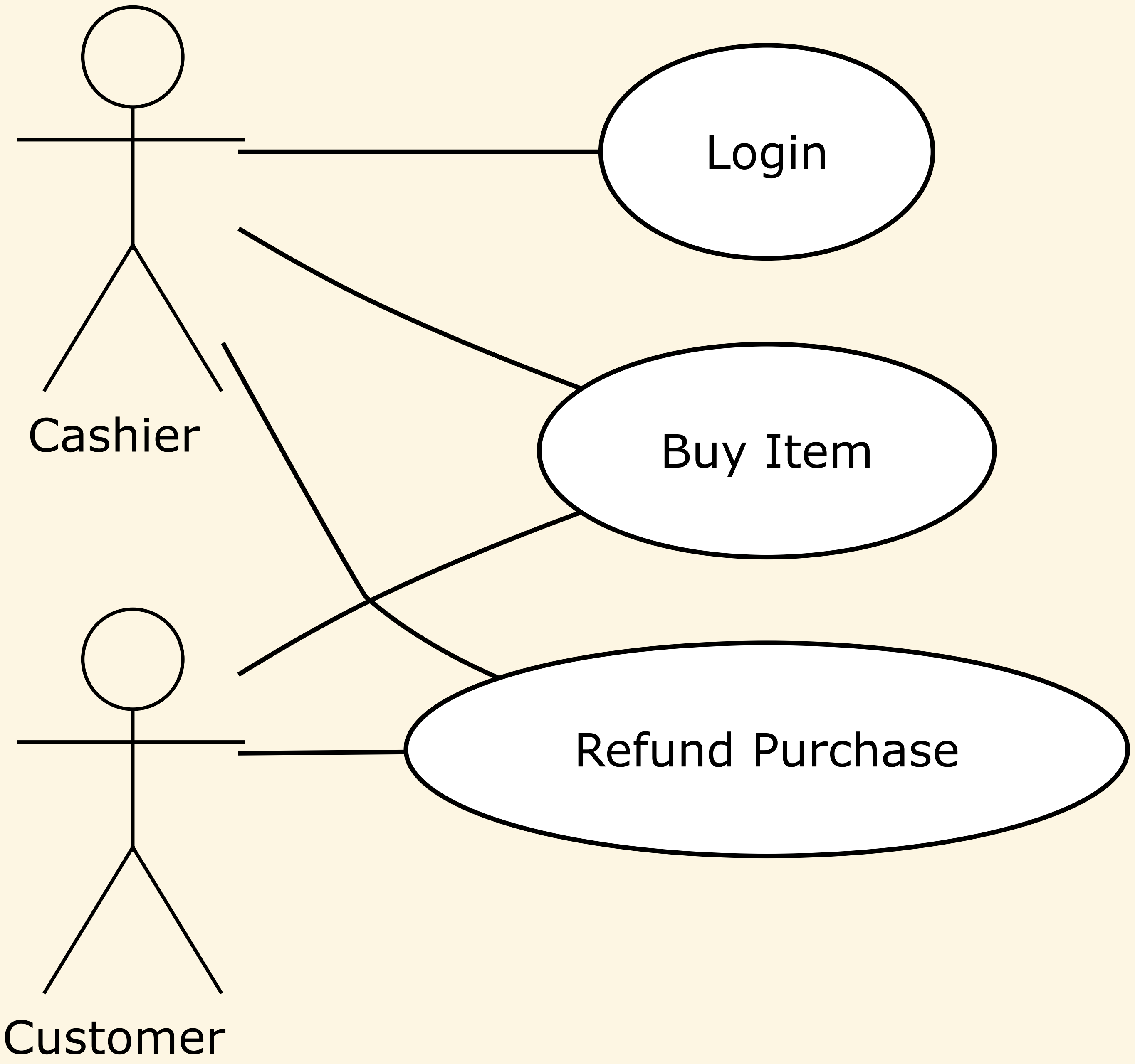
- Interaction - things the user does to achieve the goal

  Define a style

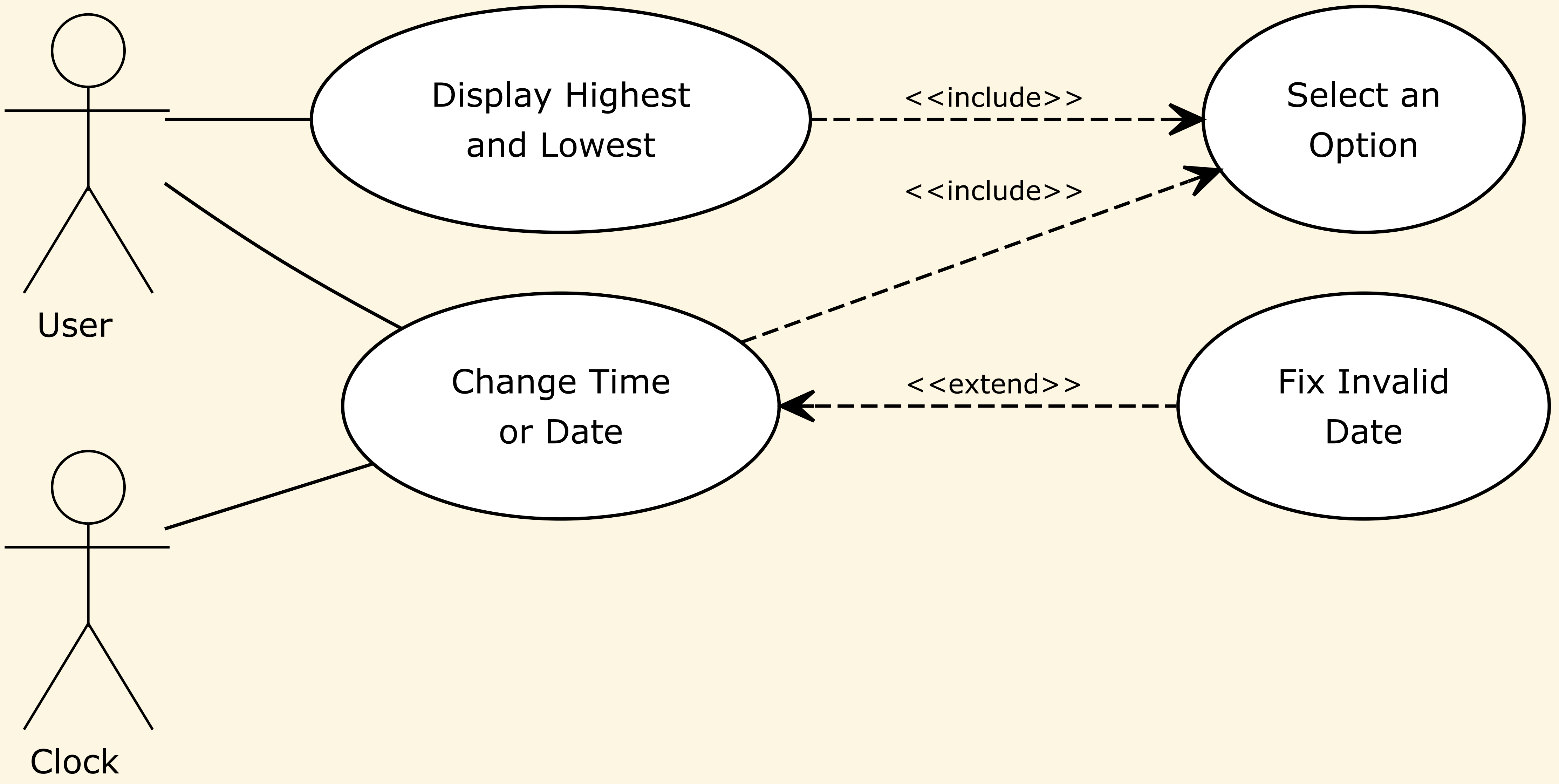
  Change a style

  Copy a style

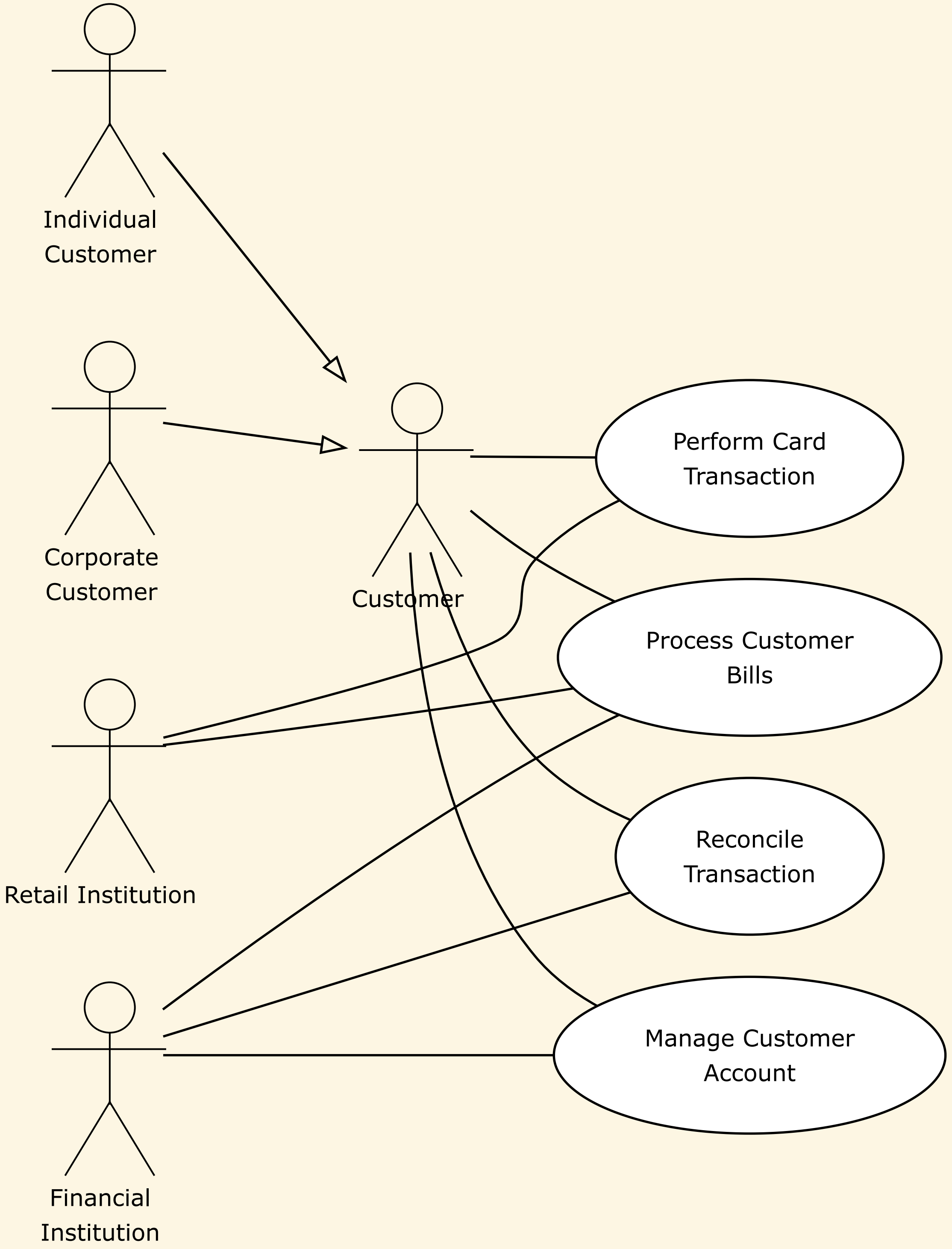
Point of Sale Terminal



Extend and Include



Generalization



Use Case Diagram

Create Use Case Diagram

Developing Use Cases

Start with goals and refine them into interactions

- Capture the goals - Understand what the system must do
- Capture user interactions - Understand how the user must interact to achieve the goals
- Identify sequences of user interactions

Casual Contents Cockburn'01

- **Title** (goal)
- **Primary Actor**
- **Scope**
- **Level**
- (Story): Paragraph or two of text in the body of the use case

Fully-Dressed Contents Cockburn'01

- Title
- Primary Actor
- Goal in Context
- Scope
- Level
- Stakeholders and Interests
- Precondition
- Minimal Guarantees
- Success Guarantees
- Trigger
- Main Success Scenario
- Extensions
- Technology & Data Variations List
- Related Information

Use Case Example

UML Use Case Example in Markdown

Feature	User Stories	Use Cases
Definition	Short, informal descriptions of one or more aspects of a software feature from an end-user perspective	Formalized, detailed descriptions of system interactions, often including actors, scenarios, preconditions, and postconditions
Format	As a [role], I want to [action/desire] so that [benefit/reason].	Typically structured with title, main success scenario, extensions, preconditions, and postconditions
Length	Typically one to three sentences	Can be several pages long
Detail	High-level and often focuses on value or benefit.	Detailed, specifying step-by-step interactions
Purpose	Convey the value of a feature or requirement to the development team.	Describe how a system will behave under various conditions
Usage	Agile development methodologies, especially Scrum	Traditional software development, requirements gathering, and detailed system design
Stakeholders	Mainly product owners, developers, and end-users	Business analysts, system architects, testers, and sometimes end-users

User Stories vs. Use Cases

User Stories	**Use Cases**
needs of the software	*behavior* in the software to meet those needs
Easy for *users* to read	Describe the *complete interaction*
A single-user story …	may map to *several* use cases