
بسم الله الرحمن الرحيم

Software Requirement Engineering

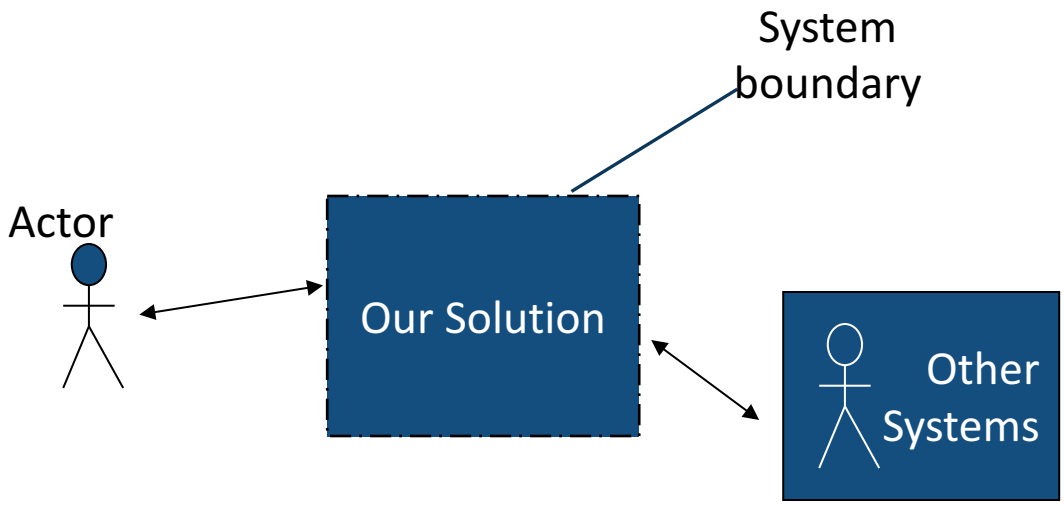
LECTURER: SYED HASNAIN ABBAS BUKHARI

Use Case Modeling

- **Use cases are a way of capturing and documenting software requirements**
- **Typical steps**
  - Find a candidate system boundary
  - Find actors and goals
  - Find Use cases
    - Specify the Use cases
    - Identify Key alternative flows

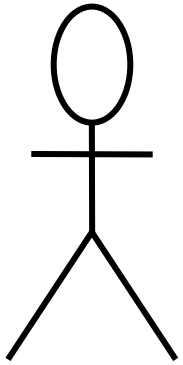
System Boundary

- **The System boundary defines the border between the solution and the real world that surround the solution**
  - Our system
  - Things that interact with our system
- **What is inside and what is outside?**



An Actor

- Is not part of the system
- Is a role a user of the system can play
  - Actor is class and users are an instances
- Can represent a human, a machine, or another system
- Can actively interchange information with the system
- Can be a giver of information
- Can be a passive recipient of information



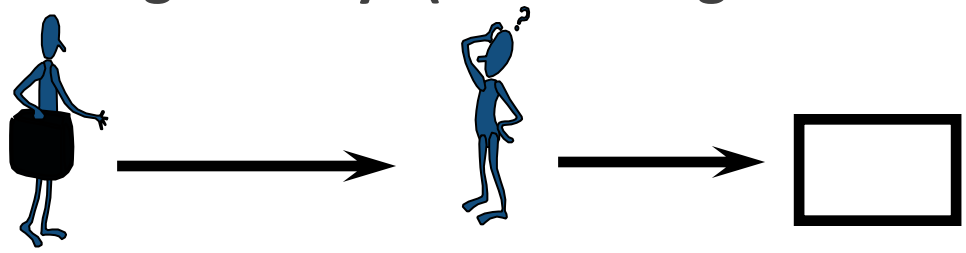
Actor

Secondary Actor

- Secondary actors are used by the system under design
- Helps the system to full fill its goal
- Will appear on the system context diagram
- May be mentioned in the Non-functional requirements
- Usually secondary actors are system roles

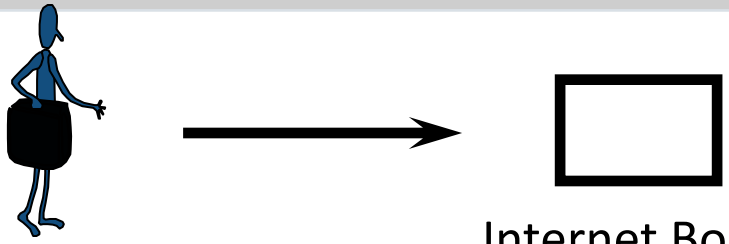
Who Is the Actor?

Who is pressing the keys (interacting with the system)?



Passenger Travel Agent Airline Booking system

The passenger never touches this system; the travel agent operates it. Or perhaps you are building an Internet application ...

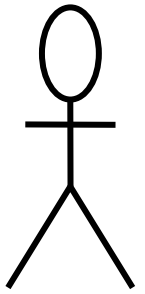


Passenger

Internet Booking system
(airline www page)

Useful Questions in Identifying Actors

- Who will supply, use, or remove information?
- Who will use this functionality?
- Who is interested in a certain requirement?
- Where in the organization is the system used?
- Who will support and maintain the system?
- What are the system's external resources?
- What other systems will need to interact with this one?



Actor

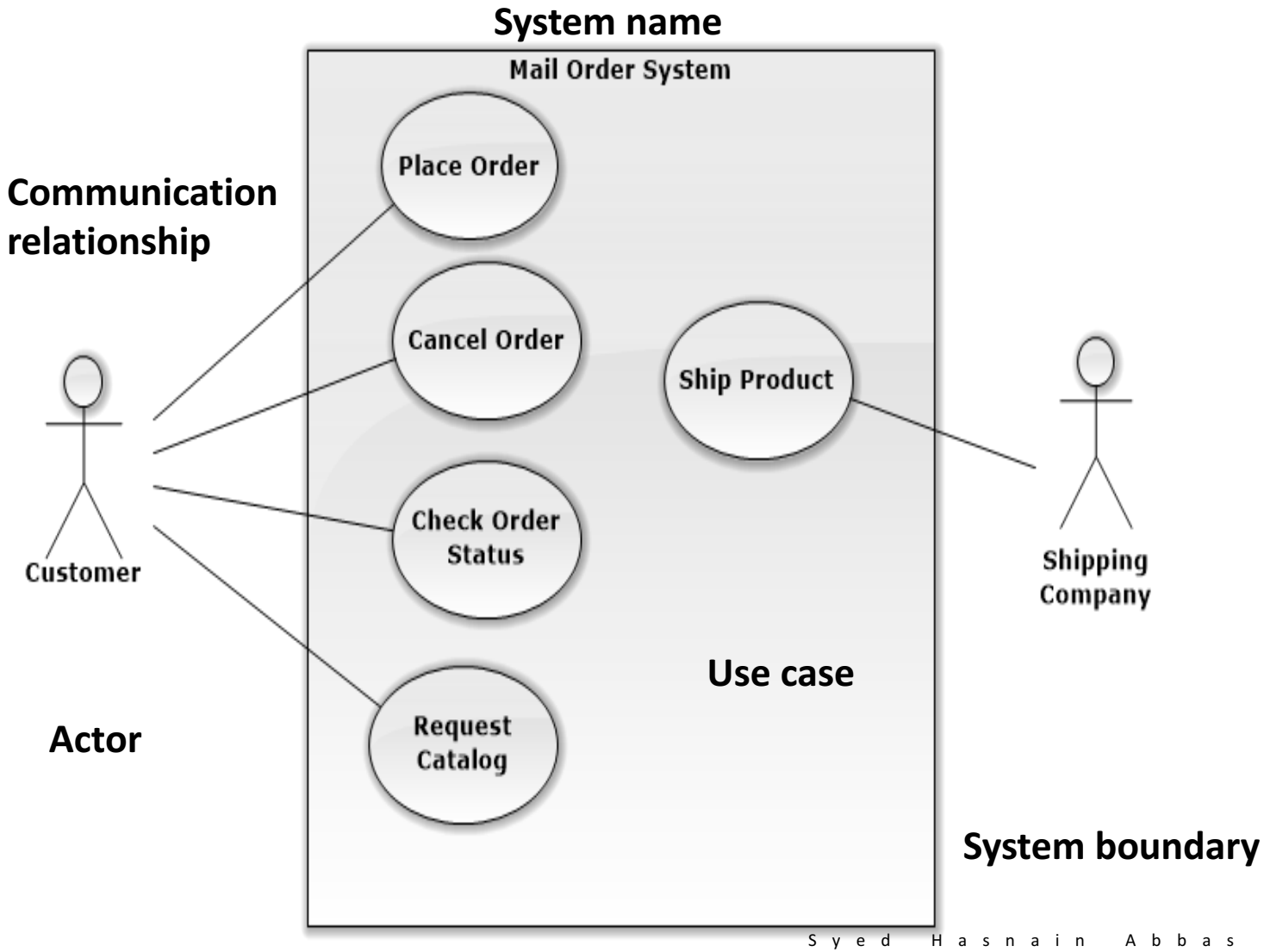
Use Cases are not Diagrams

- Use Cases may have a diagram associated with them, and a use case diagram is an easy way for an analyst to discuss a process with a subject matter expert (SME).

- But use cases are primarily text.

- The text is important. The diagram is optional.

Use case diagram (using UML)



Preconditions

- Anything that must always be true before beginning a scenario is a precondition.
- Preconditions are assumed to be true, not tested within the Use Case itself.
- Ignore obvious preconditions such as the power being turned on. Only document items necessary to understand the Use Case.

Post Condition (Success Guarantees)

- Success Guarantees or Post conditions state what must be true if the Use Case is completed successfully. This may include the main success scenario and some alternative paths.

- Stakeholders should agree on the guarantee.

Main success scenario

- a.k.a. Main flow
- How do they get it?
- Describes "perfect world" steps in a use case (everything goes as expected)

- Use case start when the customer selects "place order"

  The customer enters his or her name and address into the form

- Customer details are entered (bad – passive voice)
  - Who enters?
  - What is entered?
  - What is "customer details"?

Main success scenario

- The Success Scenario (or basic course) gives the best understanding of the use case
- Each step contains the activities and inputs of the actor and the system response
- If there are three or more items, create a list
- Label steps for configuration management and requirements traceability
- Use present tense and active voice
- Remember that User Interface designers will use this specification

Finding Extensions (alternative flows)

- **Each use case has a main flow and many have many alternative flows**
- **These flows capture errors, branches, and interrupts to the main flow**
- **For each step in the main flow look for**
  - Possible alternative to the main flow (how else?)
  - Errors that might be raised in the main flow
  - Interrupts that might occur at a particular point in the main flow
  - Interrupts that might occur at any point in the main flow
- **Only document most important alternative flows**

Use cases specification template

< ToDo: Create a table for each use case. Use cases must be numbers e.g., UC01, UC02…. />

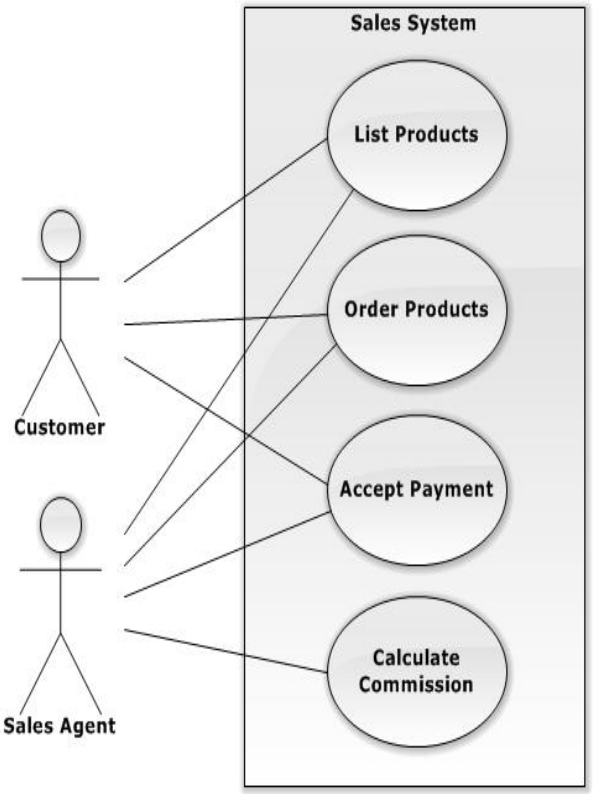
Use case Id: UC??	<Use case Title>
Brief Description	
Primary actors	
Preconditions:	
Post-conditions:	
Main Success Scenario:	
Actor Action	**System Response**
1.	2.
Alternative flows: ?.a.	

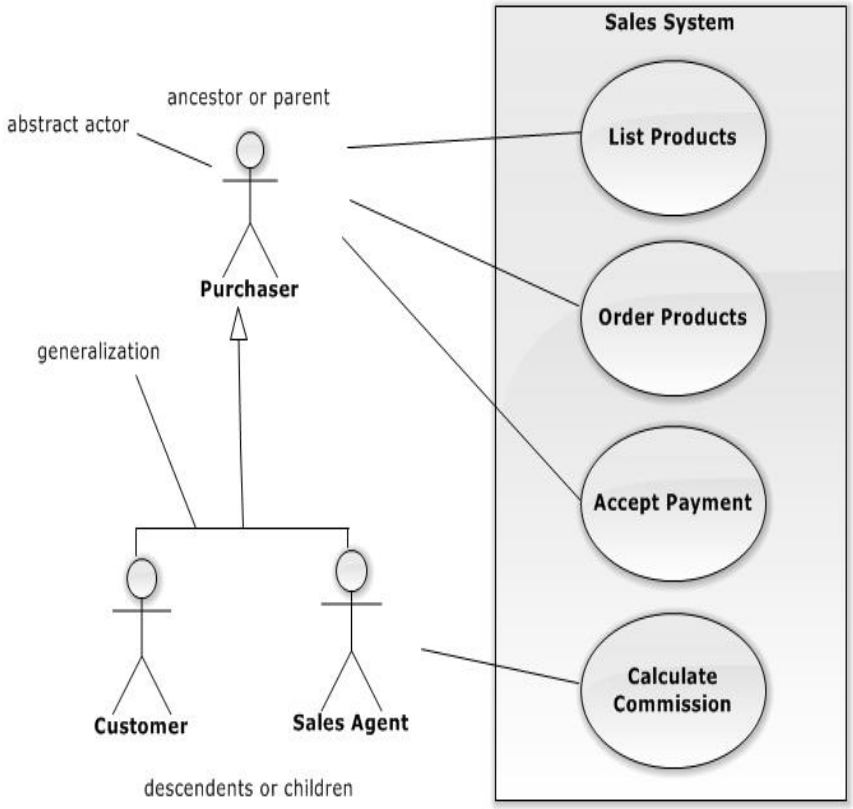
Example

Use case Id: UC01	**Login**
Brief Description	User login to the Payroll System
Primary actors	Payroll Officer and HR Manager
Preconditions: The user has a valid account.	
Post-conditions: If the use case was successful, the actor is logged into the system. If not, the system state is unchanged.	
Main Success Scenario:	
Actor Action	**System Response**
1. Enters username and password	2. The system validates the entered username and password and logs the user into the system (See 2.a. for alternative flow)
Alternative flows: **2.a. Invalid Username/Password** If the user enters an invalid username and/or password, the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the login, at which point the use case ends.	

Generalization

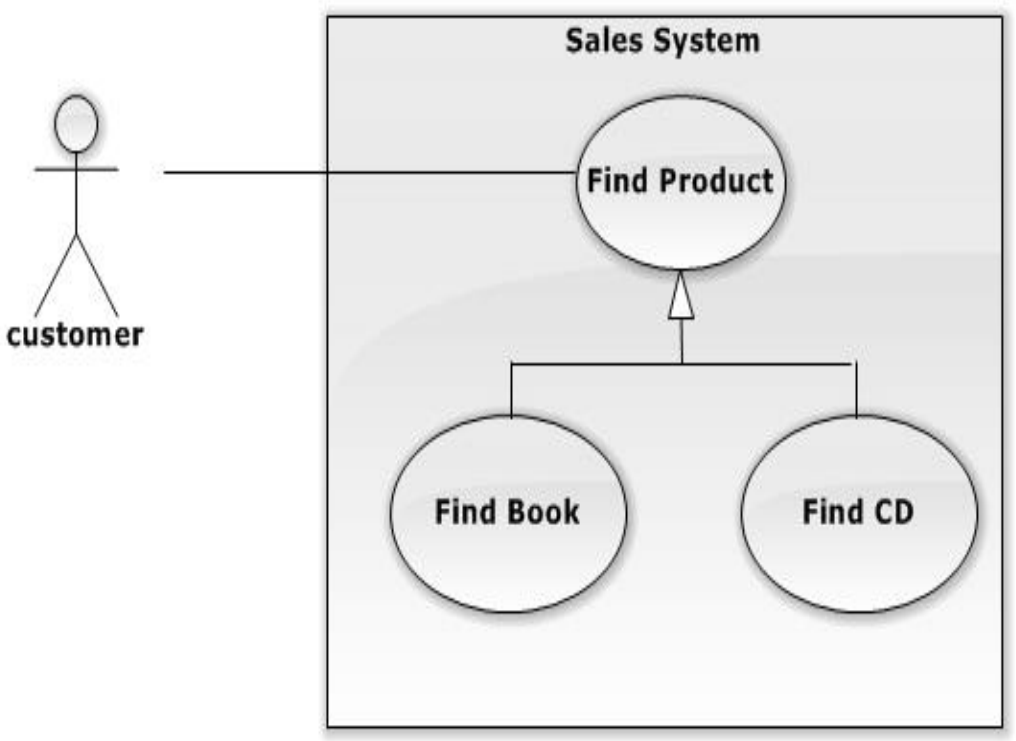
- Actor generalization factors out behavior common to two or more actors into a parent actor.





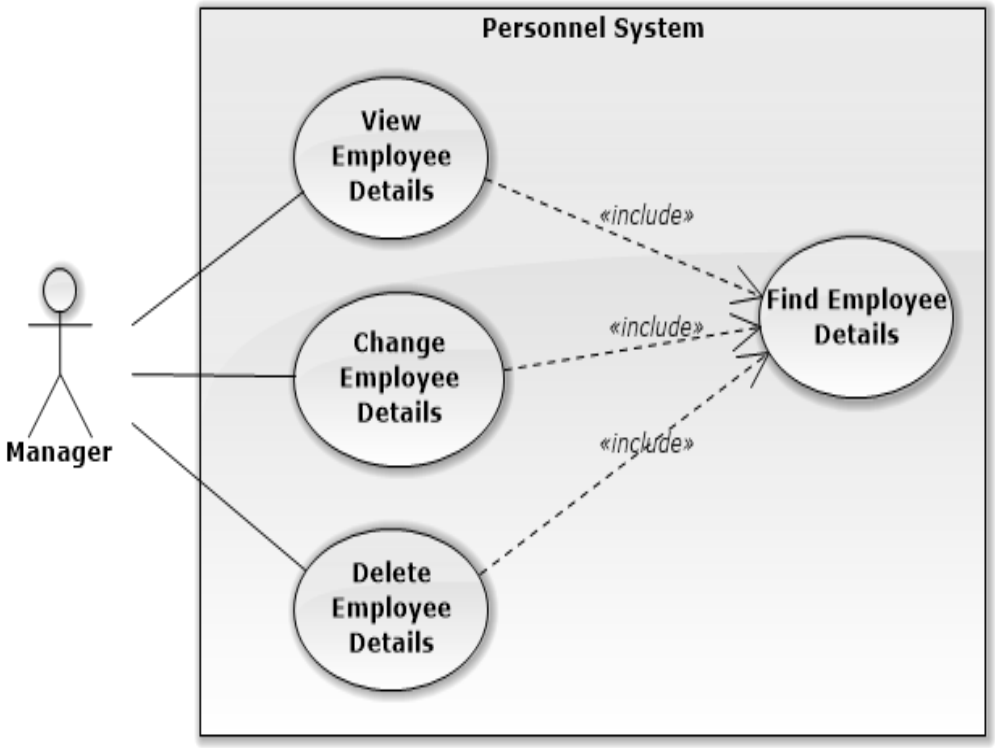
Generalization

- **Use case generalization factors out behavior common to one or more use cases into a parent use case.**



<<Include>>

- Writing use cases can be repetitive at times
- Factors out steps common to several use cases into a separate use case that is then included
- The base use case executes until the point of inclusion in reached, then execution passes over to the inclusion use case.
- When the inclusion uses case finishes, control returns to the base case



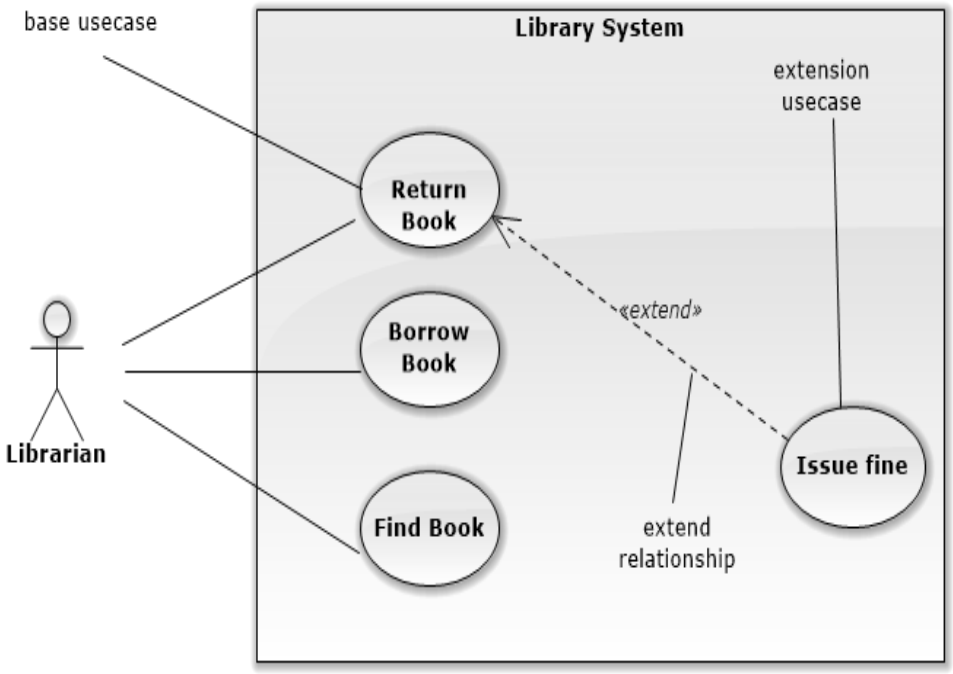
UC specification - Include

Use case: Change Employee Details
ID: 1
Brief Description: The Manager changes the employee details
Primary actors: Manager
Secondary actors: None
Preconditions: The Manager is logged on to the system
Main flow: 1. Include (Find Employee Details) 2. The System displays the employee details …
Postconditions: The employee details have been changed
Alternative flow: None

Use case: Find Employee Details
ID: 4
Brief Description: The Manager finds the employee details
Primary actors: Manager
Secondary actors: None
Preconditions: 1. The Manager is logged onto the system
Main flow: 1. The Manager enters the employee's ID 2. The system finds the employee details.
Postconditions: 1. The system has found the employee details.
Alternative flow: None

<<extend>>

- It is a way of inserting new behavior into an existing use case
- The base case provides a set of extension points that are hooks where new behavior may be added, and
- The extension use case provides a set of insertion segments that can be inserted into the base use case at these hooks



Conditional Extensions