

Requirement Analysis

Use Case Diagrams

Use Cases (UC)

- Use cases document *what* a system does/should do, not *how* it delivers/will deliver those functions
- UCs document system behavior from the user's point of view
 - Work inwards from user requirements
 - Thinking in terms of what the user wants to do with the system, not how the system will be implemented
- A use case is a sequence of actions that an actor performs within a system to achieve a particular goal
- “A use case captures a contract.. [that] describes the system's behavior under various conditions as the system responds to a request from one of its stakeholders.”

Use case example

- For library management system
 - Reserve book
 - Update catalog
 - Return a book
 - Search a book

- UC model is developed to help with:
 - Capturing system requirements
 - Maintaining traceability across iterations in system design/development
 - Creating system validation (testing) models
- UCs evolve; as understanding of users, user needs, system requirements, environment, etc change, UCs must reflect that change

Use Cases for requirements capture

- Result of Use Case modeling: all required system functionality is described in the UCs
- UCs provide a structured approach to requirements capture
 - Identify the actors
 - For each actor find out
 - What they need from the system; that is, what use cases there are which have value for them
 - Any other interactions they expect to have with the system, that is, which UCs they might take part in for someone else's benefit

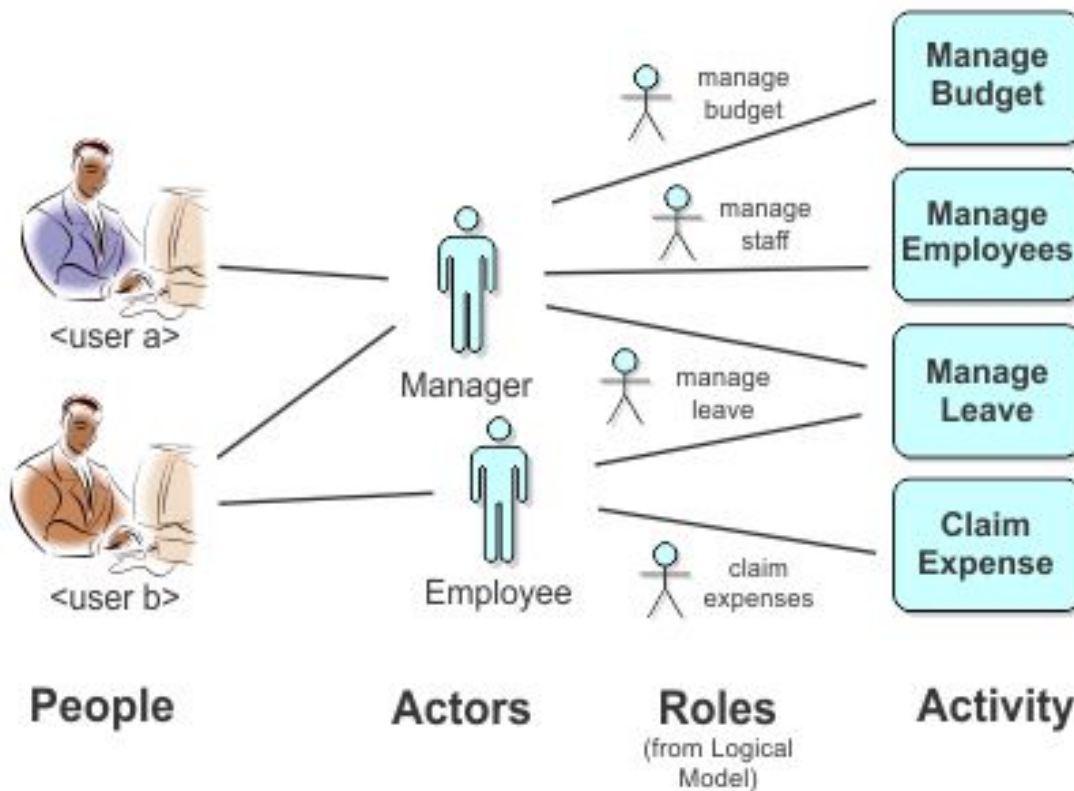
An Actor

- An actor represents an external entity that interacts with the system
 - An actor exchanges information with the system
 - A role that a user can play with regards to a system
 - An entity such as another system or a database (outside of the system being modeled)
 - Note, an actor and an end-user are not necessarily the same thing. End-users may play many roles. An actor plays one role in the context of a use-case

Identify actors

- Non-human actors can be less clear
 - What counts as an external system or device?
 - Do whatever seems to be most useful—show interactions with external systems:

User vs Actor



1. User is anyone who uses the system
2. Actor is a role that a user can play
3. A single person (user) can play more than one role (be an instance of more than one Actor)

Example: Actors in a library system

- Users:
 - Librarian
 - Library member
 - Non-member of the library
- Actors (roles):
 - Librarian
 - BookBorrower
 - JournalBorrower
 - Browser
 - Homeless



Developing the Use Cases

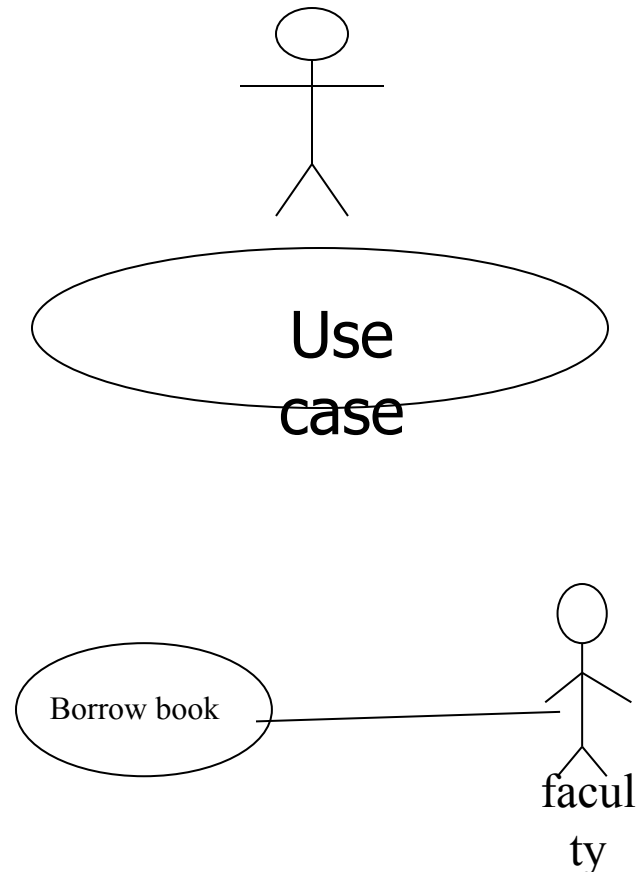
- Sit down with the Actors and identify their pertinent Use Cases
- Ask:
 - What are the main tasks (in terms of the organization) performed by each actor?
 - Will the actor read or update any information in the system?
 - Will the actor have to inform the system about changes outside the system?
 - Does the actor have to be informed about unexpected changes?

Use Case Diagrams

- In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.
- These help in
 - Representing the goals of system-user interactions
 - Defining and organizing functional requirements in a system
 - Specifying the context and requirements of a system

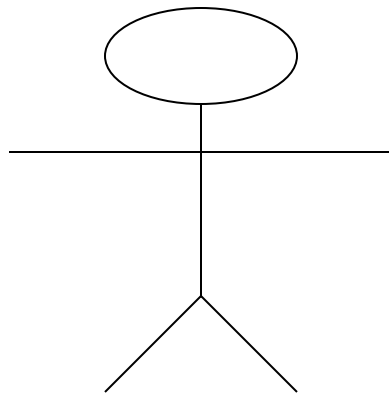
Use case diagram components

- A set of **ACTORS** : roles the users can play in interacting with the system.
- A set of **USE CASES**: each describes a possible kind of interaction between an actor and the system.
- A number of **RELATIONSHIPS** between these entities (Actors and Use Cases).
 - Relationships are simply illustrated with a line connecting actors to use cases.

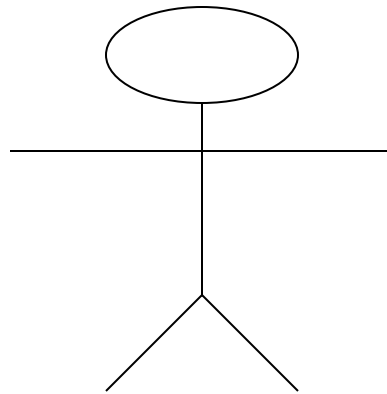


Use Case Diagrams - Actors

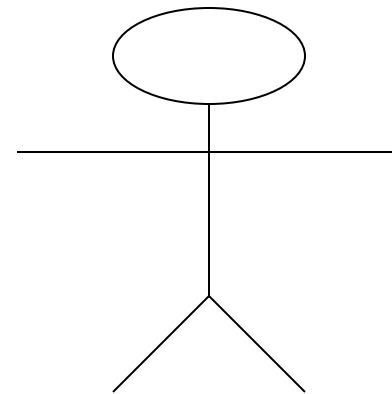
- Actor is shown with a stick figure.



employer



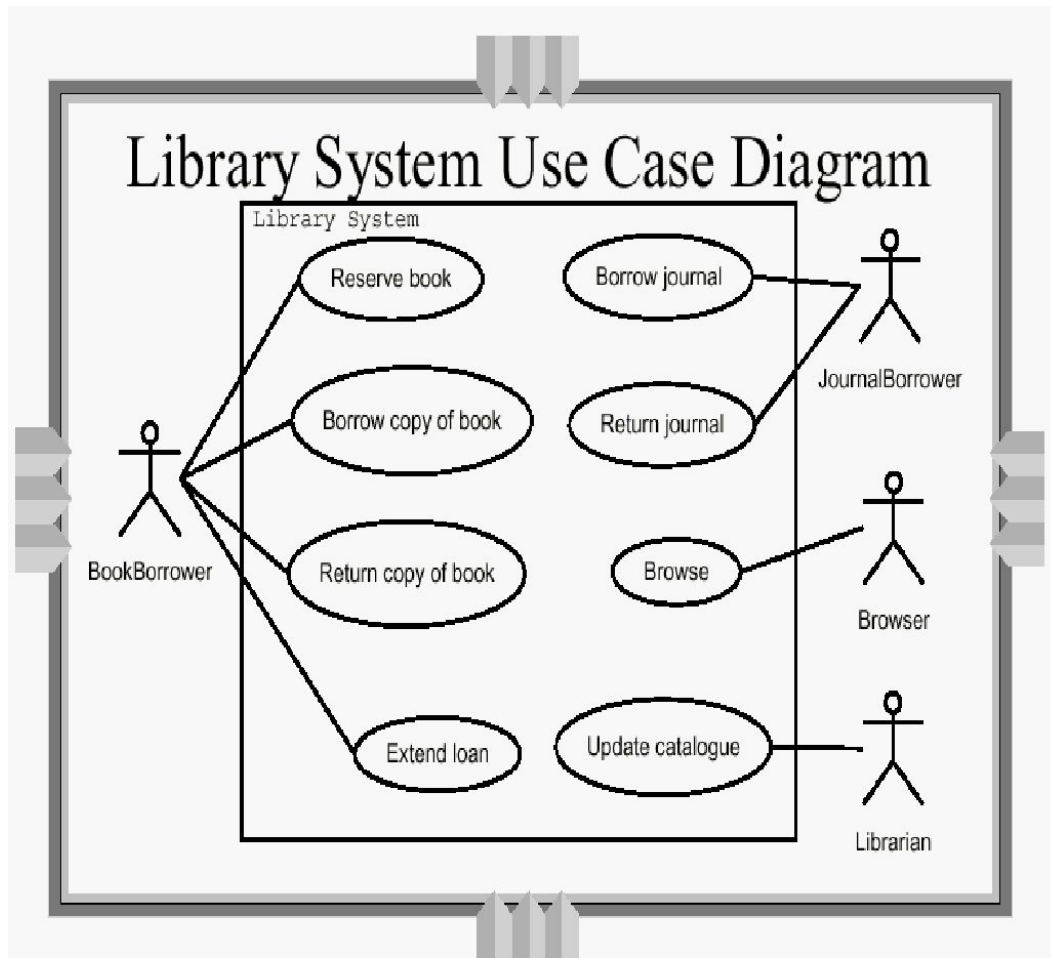
employee



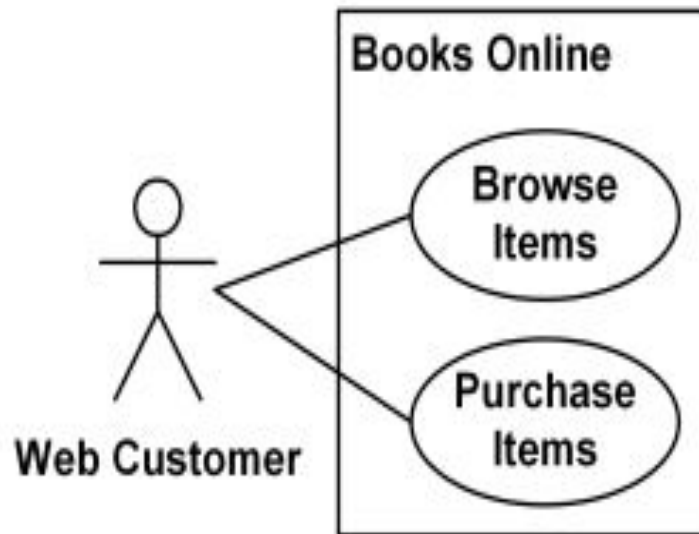
client

Use Cases in a library system: Diagram

- BookBorrower
 - Reserve book, Borrow copy of book, Return copy of book, Extend loan
- JournalBorrower
 - Borrow journal, Return journal
- Browser
 - Browse
- Librarian
 - Update catalog, ...
- Homeless
 - Special case of Browser



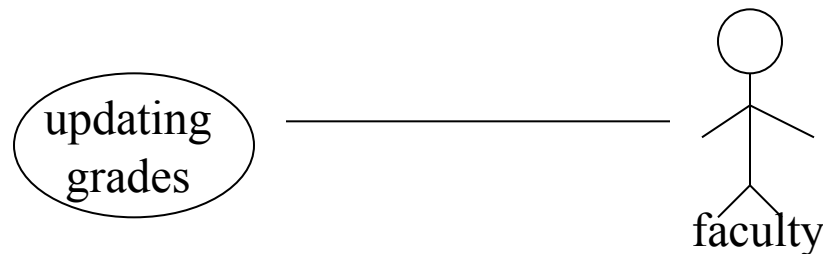
Use Case Subject



Books Online (subject) with applicable use cases and Web Customer actor.

Relationships between Use Cases and Actors

- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

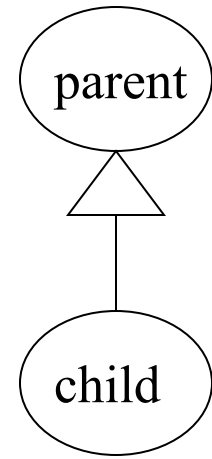


Relationships between Use Cases

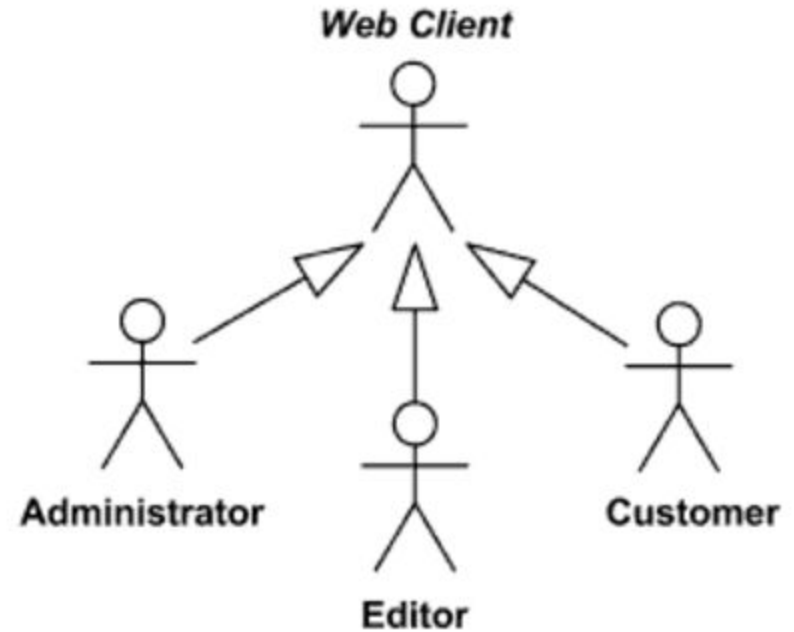
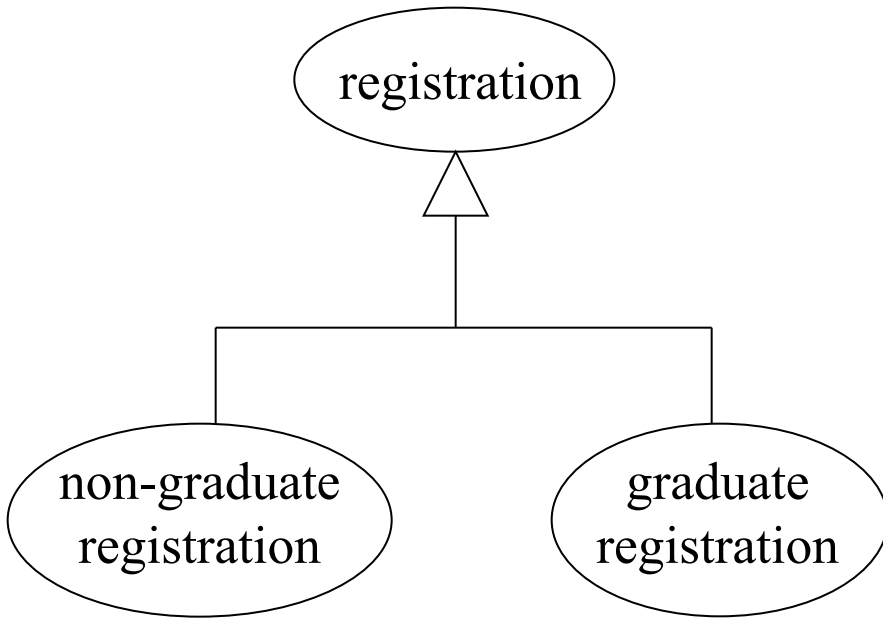
1. Generalization - use cases that are specialized versions of other use cases.
2. Include - use cases that are included as parts of other use cases. Enable to factor common behavior.
3. Extend - use cases that extend the behavior of other core use cases. Enable to factor variants.

1. Generalization

- The child use case inherits the behavior and meaning of the parent use case.
- The child may add to or override the behavior of its parent.



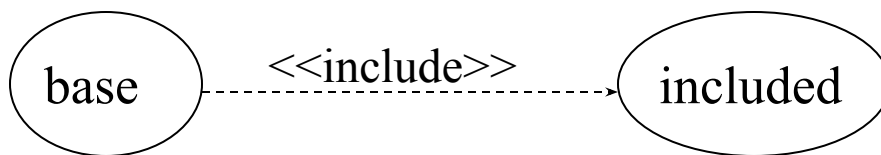
More about Generalization



Web Client actor is abstract superclass for Administrator, Editor, and Customer actors.

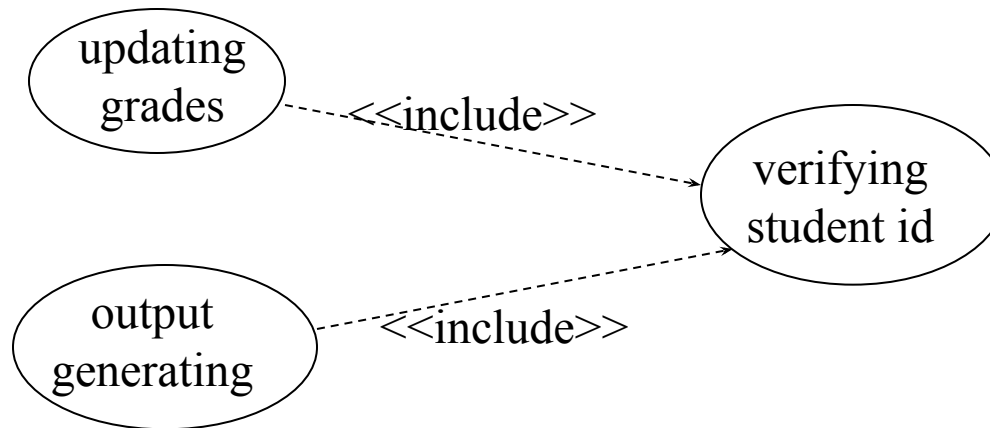
2. Include

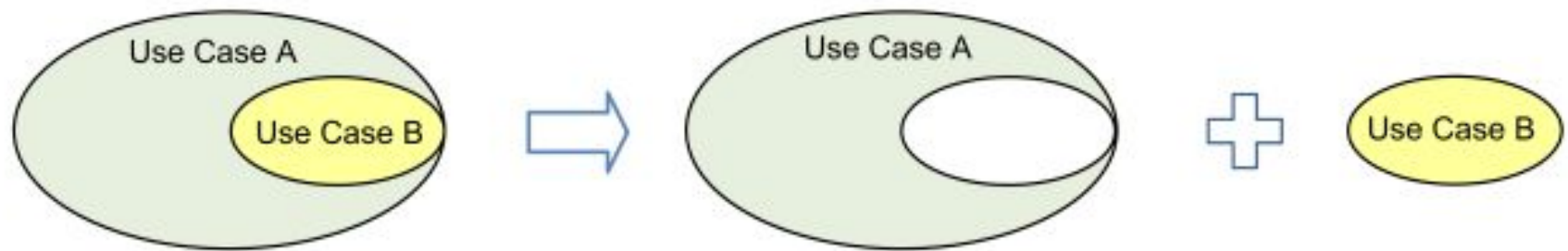
- The base use case explicitly incorporates the behavior of another use case at a location specified in the base.
- The included use case never stands alone. It only occurs as a part of some larger base that includes it.



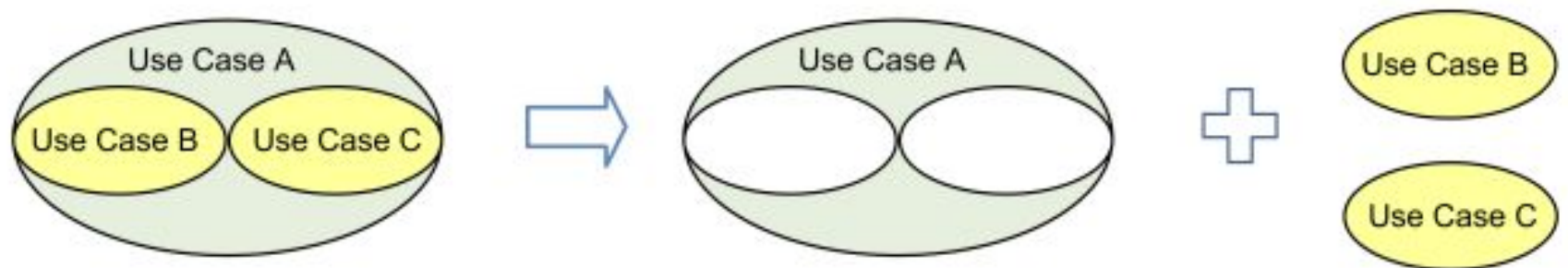
More about Include

- Enables to avoid describing the same flow of events several times by putting the common behavior in a use case of its own.

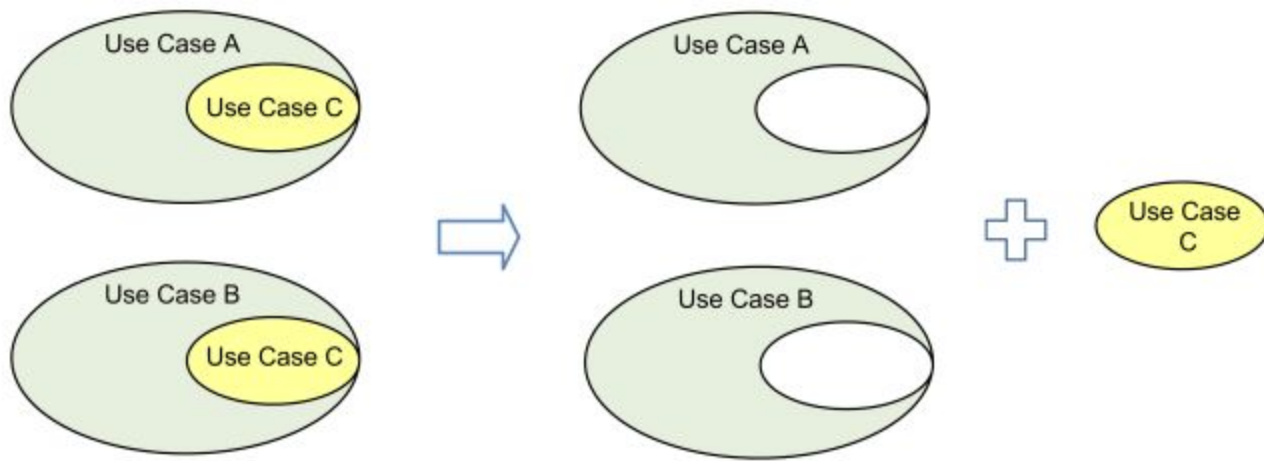




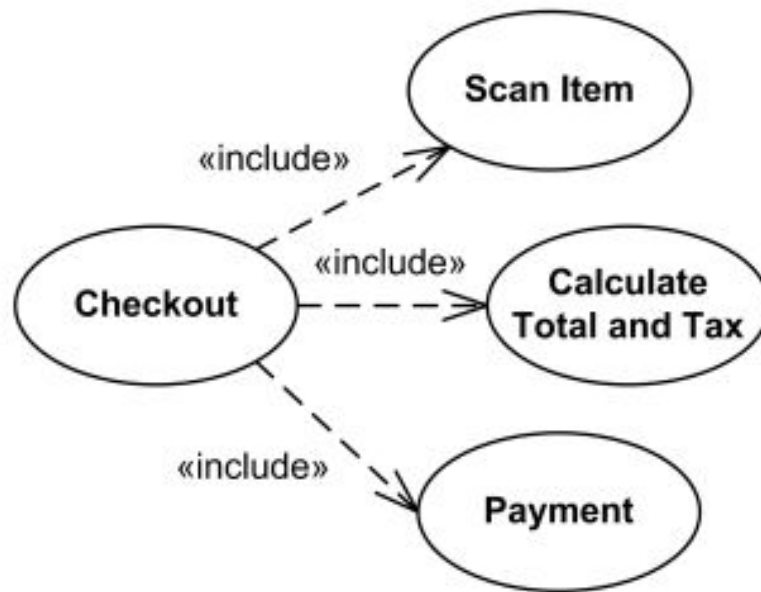
Use case B is extracted from larger use case A into a separate use case.



Use cases B and C are extracted from larger use case A into separate use cases.



Use case C is extracted from use cases A and B to be reused by both use cases using UML include relationship.



Checkout use case includes several use cases - Scan Item, Calculate Total and Tax, and Payment

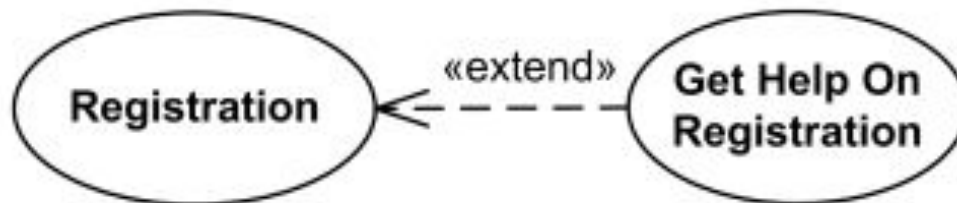
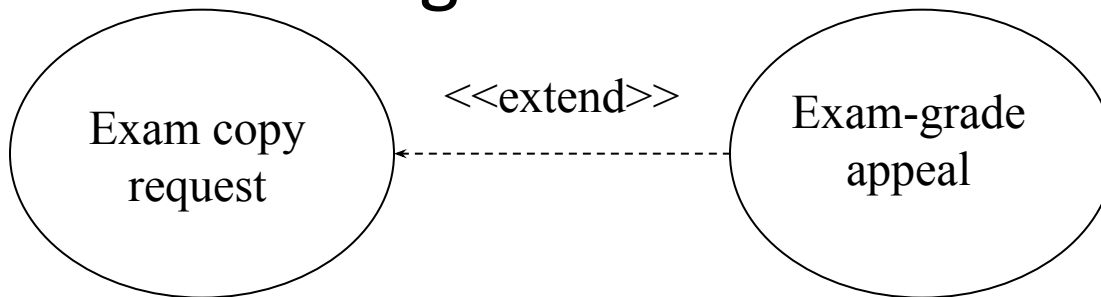
3. Extend

- The base use case implicitly incorporates the behavior of another use case at certain points called extension points.
- The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.



More about Extend

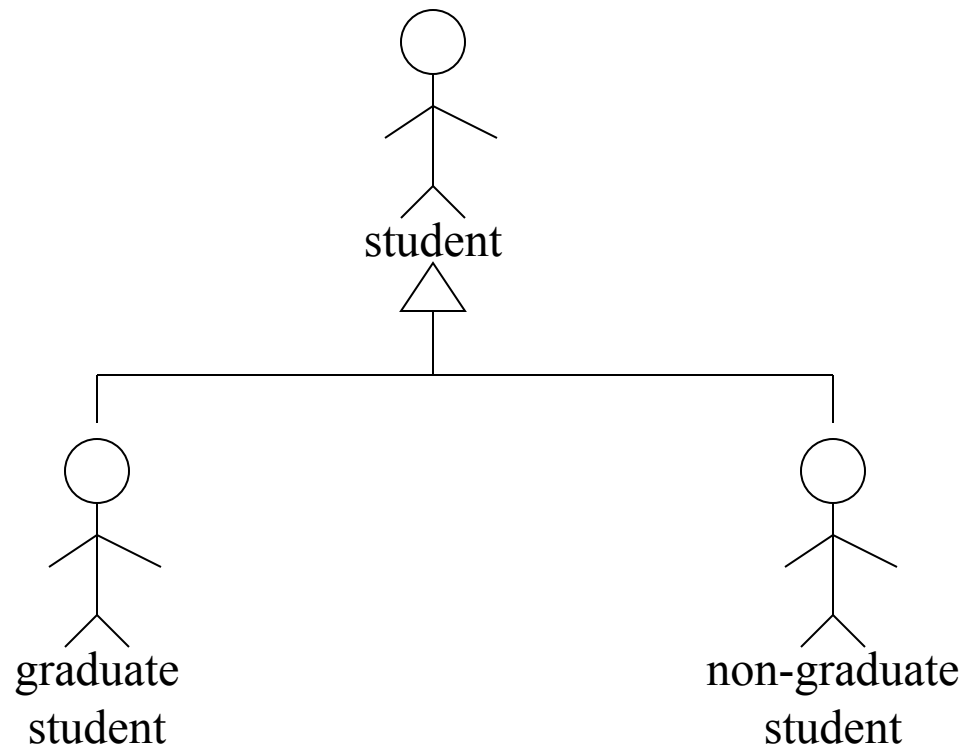
- Enables to model optional behavior or branching under conditions.



*Registration use case is complete and meaningful on its own.
It could be extended with optional **Get Help On Registration** use case.*

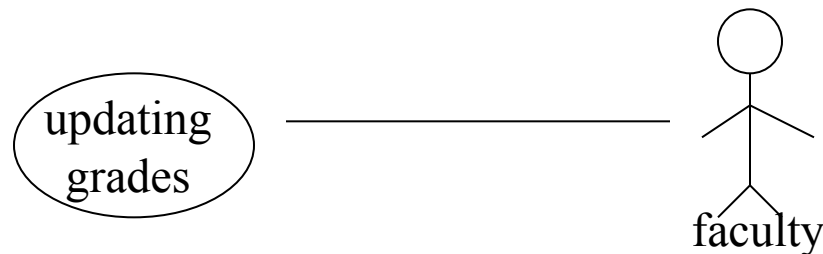
Relationships between Actors

- Generalization.

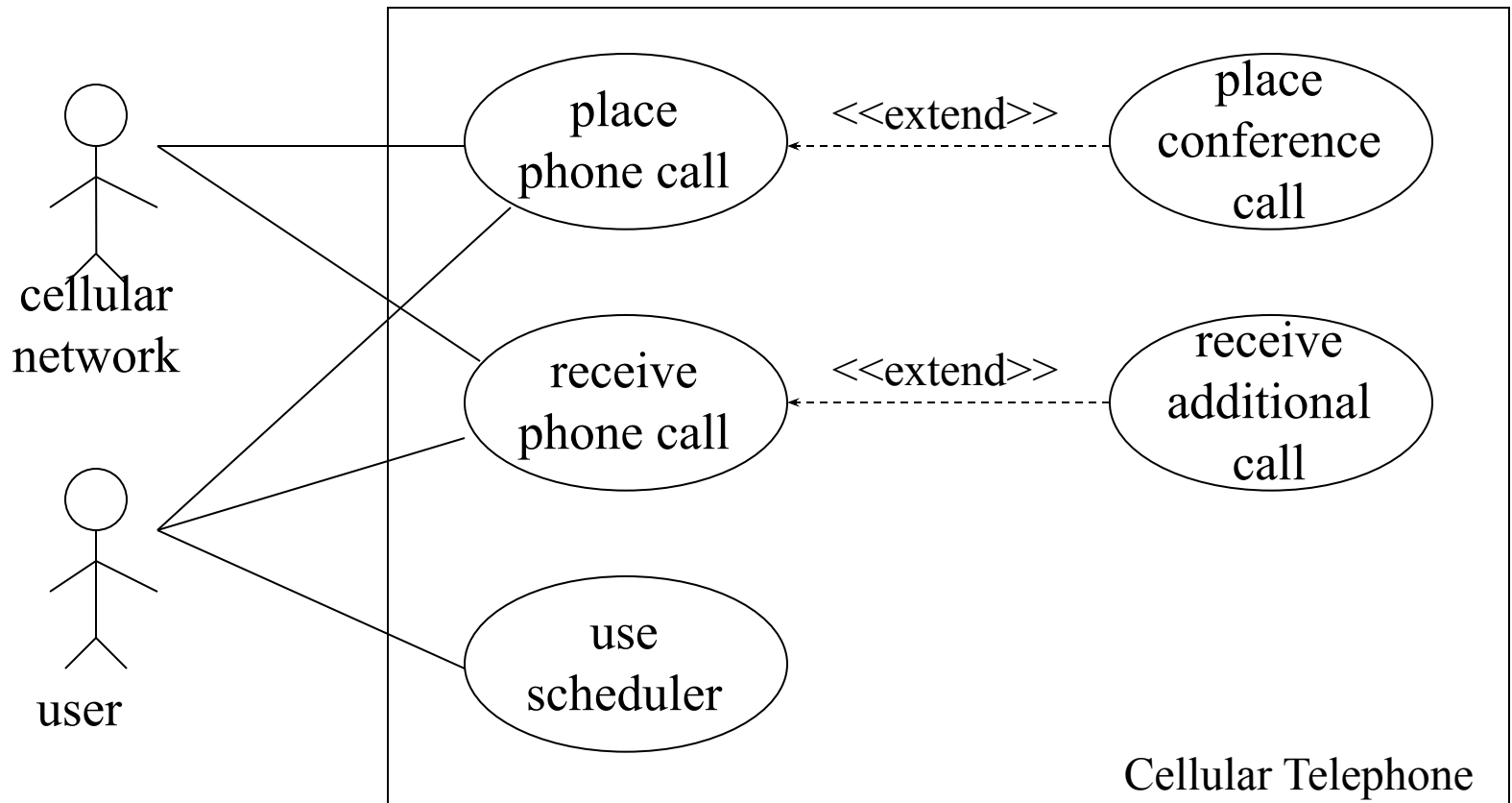


Relationships between Use Cases and Actors

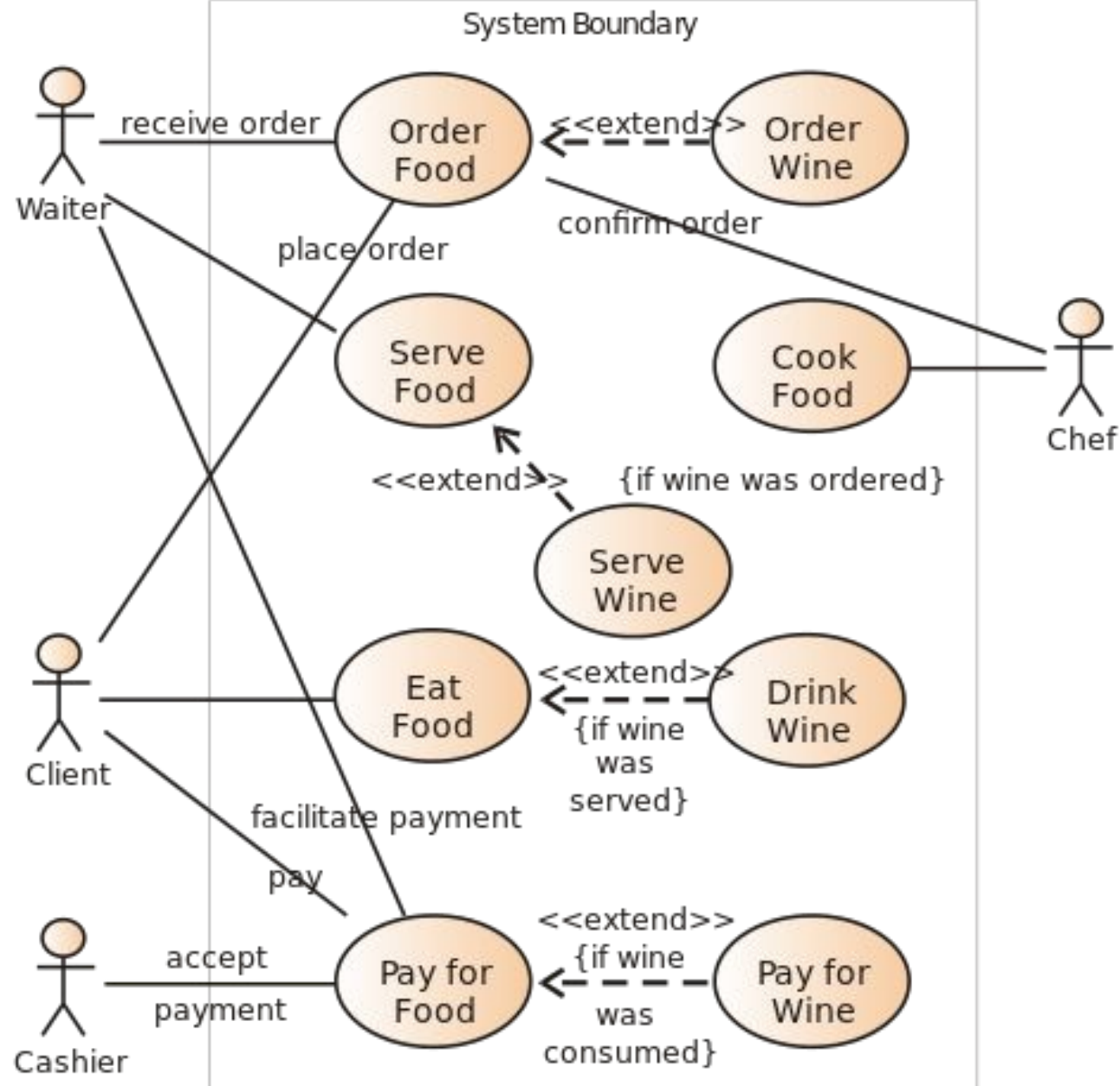
- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

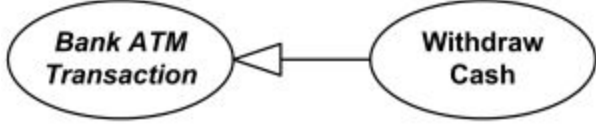

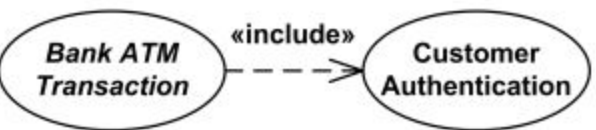


Example



uc Use Cases



Generalization	Extend	Include
 <pre> graph LR A([Bank ATM Transaction]) < -- B([Withdraw Cash]) </pre>	 <pre> graph LR A([Bank ATM Transaction]) -.-> «extend» B([Help]) </pre>	 <pre> graph LR A([Bank ATM Transaction]) -.-> «include» B([Customer Authentication]) </pre>
Base use case could be <u>abstract use case</u> (incomplete) or concrete (complete).	Base use case is complete (concrete) by itself, defined independently.	Base use case is incomplete (<u>abstract use case</u>).
Specialized use case is required, not optional, if base use case is abstract.	Extending use case is optional, supplementary.	Included use case required, not optional.
No explicit location to use specialization.	Has at least one explicit extension location.	No explicit inclusion location but is included at some location.
No explicit condition to use specialization.	Could have optional extension condition.	No explicit inclusion condition.

Use Cases description

- Story like description easy to understand by both users and analysts
- They do not form the complete SRS, only the functionality part.
- Use case description consists of scenarios.

Use case description-Documenting Use Cases

- Detail of each UC must be written down
 - Use third person, active voice English
 - Use terms from the problem domain
 - Say what the system does (in terms of the business; what it accomplishes for the business), not implementation detail or how
- The **basic course of action** (main success scenario) is the main start-to-finish path the user will follow under normal circumstances
- An **alternative course of action(exception scenario)** can represent an infrequently used path, an exception, or an error

...documenting

- Create a simple UC template:
 - Actor(s) that initiate the UC
 - Basic course
 - Alternate courses
- Eliciting Use Cases from system stakeholders:
 - Ask “what happens?” This gets the basic course of action started.
 - Ask “and then what happens?” Keep asking this question until you have all details for the basic course on paper
 - Be relentless. “what else can happen? Are there any other things that can happen?” Keep asking until you have a set of alternative courses written down.
 - Alternative courses usually hardest to elicit—hard to think of unusual situations, odd occurrences, ...

- DEVELOPING USE CASES

- Questions to ask about actors:

- ◆ Who are the actor(s)?
- ◆ What are the actor's goals?
- ◆ What preconditions should exist before the story begins?
- ◆ What exceptions might be considered as the story is described?
- ◆ What variations in the actor's interaction are possible?
- ◆ What type of system information will the actor acquire, produce, or change?
- ◆ Will the actor have to inform the system about changes in the external environment?
- ◆ What information does the actor desire from the system?
- ◆ Does the actor wish to be informed about unexpected change?

Use Case Description

:Each use case may include all or part of the following

- **Actors** - A person or a system which uses the system being built for achieving some goal.
- **Purpose** – Explain the purpose
- **Primary actor** - The main actor for whom a use case is initiated and whose goal satisfaction is the main objective of the use case.
- **Pre Conditions**- must be true to allow execution
- **Post Conditions**- will be set when completes normally
- **Main success scenario** – Describes the interaction if nothing fails and all steps in the scenario succeed.(normal flow of control)
- **Exceptional scenario** - Describes the system behavior if some of the steps in the main scenario do not complete successfully(unusual situations)

The **basic course of action** is the main start-to-finish path the user will follow under normal circumstances

An **alternative course** of action can represent an infrequently used path, an exception, or an error

Template for use case description

Use Case:

Actors:

Pre Condition:

Post Condition:

Main success scenario:

Step 1 :

Step 2 : -----

Exception scenario:

Step 1 :

Step 2 : -----

Example- Money Withdraw (cont.)

- Use Case: Withdraw Money
- Actors: Customer
- Pre Condition:
 - The ATM must be in a state ready to accept transactions
 - The ATM must have at least some cash on hand that it can dispense
 - The ATM must have enough paper to print a receipt for at least one transaction
- Post Condition:
 - The current amount of cash in the user account is the amount before the withdraw minus the withdraw amount
 - A receipt was printed on the withdraw amount
 - The withdraw transaction was audit in the System log file

Example- Money Withdraw (cont.)

main success scenario

- 1. Begins when a Customer arrives at ATM**
- 2. Customer inserts a Credit card into ATM**
- 3. System verifies the customer ID and status**
- 4. System asks for an operation type**
- 5. Customer chooses "Withdraw" operation**
- 6. System asks for the withdraw amount**
- 7. Customer enters the cash amount**
- 8. System checks if withdraw amount is legal**
- 9. System dispenses the cash**
- 10. System deduces the withdraw amount from account**
- 11. System prints a receipt**
- 12. System ejects the cash card**
- 13. Customer takes the cash and the receipt**

Example- Money Withdraw (cont.)

- Exception scenario:
 - 3 a: Customer authorization failed. Display an error message, cancel the transaction and eject the card.
 - 8 a: Customer has insufficient funds in its account. Display an error message, and go to step 6.
 - 8 b: Customer exceeds its legal amount. Display an error message, and go to step 6.
 - 9 a: if Power failure in the process of the transaction before step 9, cancel the transaction and eject the card

USE CASE DESCRIPTION TEMPLATE(pressman book)



Use case: Access camera surveillance via the Internet—display camera views (ACS-DCV)

Iteration: 2, last modification: January 14 by V. Raman.

Primary actor: Homeowner.

Goal in context: To view output of camera placed throughout the house from any remote location via the Internet.

Preconditions: System must be fully configured; appropriate user ID and passwords must be obtained.

Trigger: The homeowner decides to take a look inside the house while away.

Scenario:

1. The homeowner logs onto the *SafeHome Products* website.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects the “surveillance” from the major function buttons.
6. The homeowner selects “pick a camera.”
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the “view” button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

Exceptions:

1. ID or passwords are incorrect or not recognized—see use case **Validate ID and passwords**.
2. Surveillance function not configured for this system—system displays appropriate error message; see use case **Configure surveillance function**.
3. Homeowner selects “View thumbnail snapshots for all camera”—see use case **View thumbnail snapshots for all cameras**.
4. A floor plan is not available or has not been configured—display appropriate error message and see use case **Configure floor plan**.
5. An alarm condition is encountered—see use case **Alarm condition encountered**.

Priority: Moderate priority, to be implemented after basic functions.

When available: Third increment.

Frequency of use: Moderate frequency.

Channel to actor: Via PC-based browser and Internet connection.

Secondary actors: System administrator, cameras.

Channels to secondary actors:

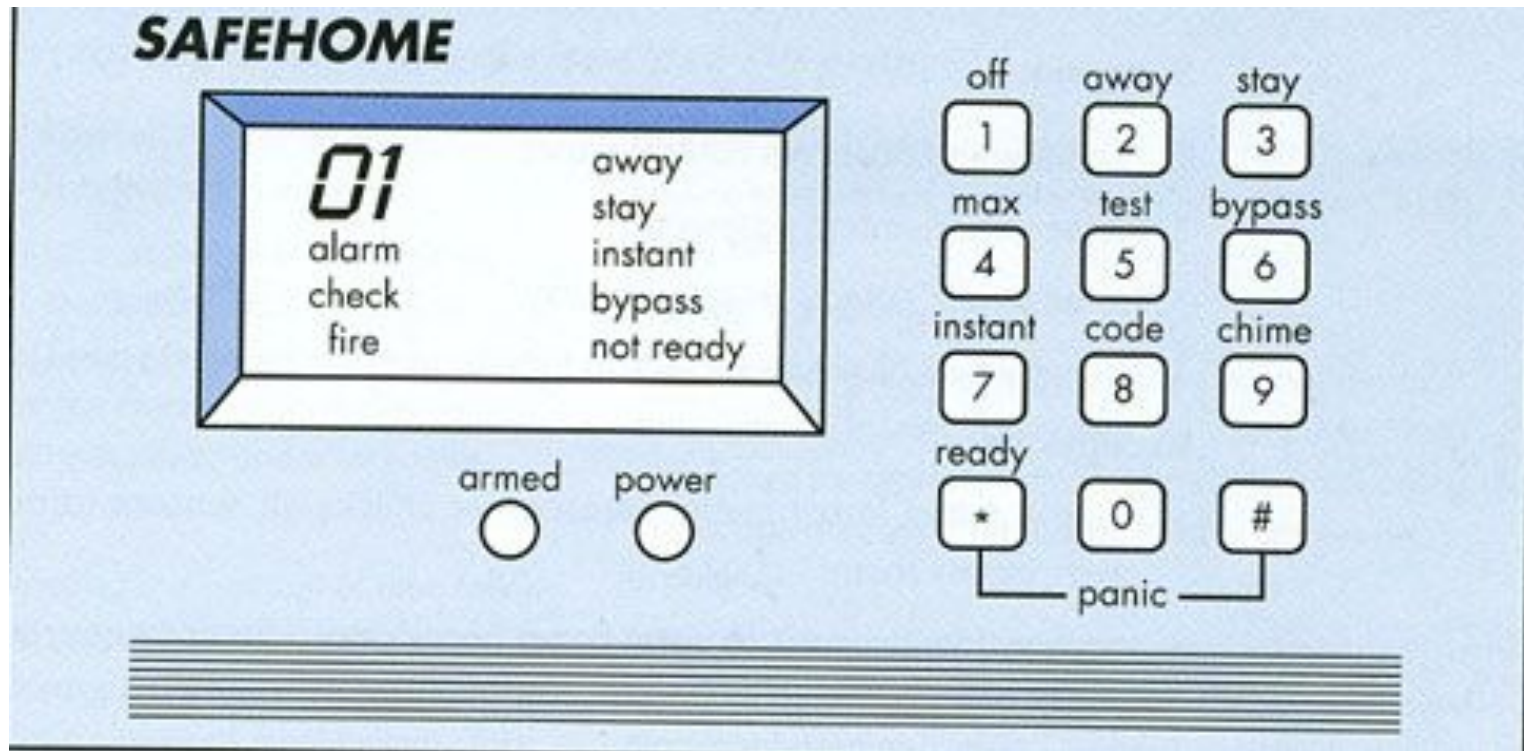
1. System administrator: PC-based system.
2. Cameras: wireless connectivity.

Open issues:

1. What mechanisms protect unauthorized use of this capability by employees of *SafeHome Products*?
2. Is security sufficient? Hacking into this feature would represent a major invasion of privacy.
3. Will system response via the Internet be acceptable given the bandwidth required for camera views?
4. Will we develop a capability to provide video at a higher frames-per-second rate when high-bandwidth connections are available?

Use case description

Think about a home security system.



Contd...

Three actors exist: homeowner/configuration manager, sensors, and the monitoring subsystem.

Let's look at the homeowner. The homeowner interacts with the home security function in a number of different ways using either the alarm control panel or a PC:

1. Enters a password to allow all other interactions.
2. Inquires about the status of a security zone.
3. Inquiries about the status of a sensor.
4. Presses the panic button in an emergency.
5. Activates/deactivate the security system

DEVELOPING USE CASES

Questions to ask about actors:

1. Who are the actor(s)?
2. What are the actor's goals?
3. What preconditions should exist before the story begins?
4. What exceptions might be considered as the story is described?
5. What variations in the actor's interaction are possible?
6. What type of system information will the actor acquire, produce, or change?
7. Will the actor have to inform the system about changes in the external environment?
8. What information does the actor desire from the system?
9. Does the actor wish to be informed about unexpected change?

Use case: Initiate Monitoring

Primary Actor: Homeowner

Goal in context: To set the system to monitor sensors when the homeowner leaves the house or remains inside.

Preconditions: System was programmed for a password and to recognize various sensors.

Trigger: The homeowner decides to “set” the system, i.e. turn on the alarm function.

Scenario:

1. Homeowner observes control panel.
2. Homeowner enters password.
3. Homeowner selects ‘stay” or “away”
4. Homeowner observes red alarm light to indicate that SafeHome has been armed

Exceptions:

- 1a. Control panel is not ready: Homeowner checks all sensors to determine which are open and closes them.
- 2a. Password is incorrect (control panel beeps once) homeowner enters correct password.
- 2b. Password not recognized: monitoring and response subsystem must be contacted to reprogram password
- 1. Stay is selected: control panel beeps twice and a stay light is lit; perimeter sensors are activated
- 2. Away is selected. Control panel beeps three times and an away light is lit; all sensors are activated

Priority: Essential, must be implemented

When available: First Increment

Frequency of use: Many times a day

Channel to actor: via control panel

Example

1. Purchaser buy stocks
2. Customer buy product

Example : Purchaser buy stocks

Use Case 1: Buy stocks

Actor: Purchaser

Purpose : Goals of Stakeholders:

Purchaser: wants to buy stocks

Company: wants full transaction info

Precondition: User already has an account

Example (Purchaser buy stocks)...

- **Main Success Scenario**

1. User selects to buy stocks
2. System gets name of web site from user for trading
3. Establishes connection
4. User browses and buys stocks
5. System intercepts responses from the site and updates user portfolio
6. System shows user new portfolio

Example (Purchaser buy stocks)...

Exception Scenario

- 2a: System gives err msg, asks for new suggestion for site, gives option to cancel
- 3a: Web failure. 1-Sys reports failure to user, backs up to previous step. 2-User exits or tries again
- 4a: web site does not ack purchase
- 5a: web site does not return needed info

Example 2- Buy a product

- Use Case 2: Buy a product
- Primary actor: buyer/customer
- Goal: purchase some product
- Precondition: Customer is already logged in

Example 2 (Buy a product)

- **Main Scenario**
 1. Customer browses and selects items
 2. Customer goes to checkout
 3. Customer fills shipping options
 4. System presents full pricing info
 5. Customer fills credit card info
 6. System authorizes purchase
 7. System confirms sale
 8. System sends confirming email

Example 2 (Buy a product) ...

Exception Scenario

- 6a: Credit card authorization fails
 - Allows customer to reenter info
- 3a: Regular customer
 - System displays last 4 digits of credit card no
 - Asks customer to OK it or change it
 - Moves to step 6

Example- An auction site

- Use case 0 : Auction an item
- Use case 1 : Put an item for auction
- Use case 2 : Make a bid
- Use case 3 : Complete auction of an item

Example – summary-level Use Case

- ***Use Case 0 : Auction an item***
- ***Primary Actor:*** Auction system
- ***Scope:*** Auction conducting organization
- ***Precondition:*** None
- ***Main Success Scenario:***
 - Seller performs put an item for auction
 - Various bidders make a bid
 - On final date perform Complete the auction of the item
 - Get feed back from seller; get feedback from buyer; update records

Example – An auction site

- **Use Case1:** Put an item for auction
- **Primary Actor:** Seller
- **Precondition:** Seller has logged in
- **Main Success Scenario:**
 1. Seller posts an item (its category, description, picture, etc.) for auction
 2. System shows past prices of similar items to seller
 3. System specifies the starting bid price and a date when auction will close
 4. System accepts the item and posts it
- **Exception Scenarios:**
 - 2 a) There are no past items of this category
 - * System tells the seller this situation

Example – auction site..

- **Use Case2:** Make a bid
- **Primary Actor:** Buyer
- **Precondition:** The buyer has logged in
- **Main Success Scenario:**
 1. Buyer searches or browses and selects some item
 2. System shows the rating of the seller, the starting bid, the current bids, and the highest bid; asks buyer to make a bid
 3. Buyer specifies bid price, max bid price, and increment
 4. Systems accepts the bid; Blocks funds in bidders account
 5. System updates the bid price of other bidders where needed, and updates the records for the item

- ***Exception Scenarios:***

- 3 a) The bid price is lower than the current highest

- * System informs the bidder and asks to re-bid

- 4 a) The bidder does not have enough funds in his account

- * System cancels the bid, asks the user to get more funds

Example –auction site..

- **Use Case3: Complete auction of an item**
- **Primary Actor:** Auction System
- **Precondition:** The last date for bidding has been reached
- **Main Success Scenario:**
 - Select highest bidder; send email to selected bidder and seller informing final bid price; send email to other bidders also
 - Debit bidder's account and credit seller's account
 - Transfer from seller's account commission amount to organization's account
 - Remove item from the site; update records
- **Exception Scenarios:** None

Thank you