1 Use Case Analysis

1.1 Client Side Involvement

To identify all and good use cases, it's imperative to involve the users. This is usually very expensive: Around 30-50% of development costs are allocated towards requirements and use case analysis and validation.

Use cases usually are text stories used to discover and record requirements. These use cases complement requirments analysis and provide operational requirements as a basis for system design. They do not replace requirement analysis as they do not capture non-functional requirements.

Definitions of Constituents of Use Cases

- Actor
  - Someone or something with behaviour (person, computer system, organisation, etc.)
  - **Primary Actor:** The person who initiates the use case (requests a service)
- Scenario (Use Case Instance)
  - Specific sequence of actions and interactions between actors and system
  - One particular story using a system
- Use Case
  - Collection of related success and failure scenarios
  - Describe an actors usage of a system to achieve a goal

Different Kinds of Use Cases

- White Box vs. Black Box: With whom does interaction occur?
  - White Box (Transparent): Use cases provide details on internal interaction with the system
  - Black Box: Use cases describe only interactions with external actors
- Corporate vs. System
  - Corporate: Use cases describe business process
    (Usually white box)
  - System: Use cases are described with respect to the system
    (Usually black box)

Use Case Formats

- Brief: Short, one paragraph summary. Usually outlines main succes scenario
- Casual: Informal, Multiple paragraphs that cover multiple scencarios
- Fully Dressed: All steps and variations in detail. Includes supporting sections on preconditions, success guarantees etc.

Should be precise (detailed) and accurate (correct).

Fully Dressed Use Case Template

- Use Case Name
  - Start with a verb ("Accomplish this task")
- Scope
  - Corporate, system (name), subsystem
  - Design Scope: Boundaries of the system of the use case (whole corporation, (sub-)system name)
  - Function Scope: Limits functionality to be realized. Managed by a list of functions in and out of scope
- Level
  - User goal, summary goal, subfunction
  - User goal: Most important goal of the user
  - Summary goal
    - Multiple User Goals: Describe context of system
    - Life cycle sequence of related goals
    - Table of content for lower-level use cases
  - Subfunction: Use case that is part of user goal. Singled out on a by-need basis, reusable in multiple goals
- Primary Actor: Initiates use case
- Stakeholders and Interests: Who is interested in this and what do they want?
- Preconditions
  - What must be true or worth telling
  - Enforced by system and known to be true
  - Will not be checked again during execution
- Minimal Guarantee
  - Fewest promises the system makes to Stakeholders
  - Especially if the primary actors goal cannot be achieved
  - (MVP) Minimal Viable Product
- Success Guarantees
  - What must be true on successfull completion
  - States the satisfied interests of the stakeholders after successful completion
- Main Success Scenario
  - Representative Scenario of successful execution
  - Numbered list of steps executed
  - Each step may reference a sub use case
  - First step specifies trigger of use case
- Extensions
  - Alternative scenarios of success or failure
  - Refer to main success scenarios step, by explaining alternative scenario for each step as well as the condition or failure needed for the alternative
- Special Requirements: Related non-functional requirements
- Technology and Data Variation: Needed / Used Technology and Data formats
- Frequency of Occurence: How often does the use case occur?
- Miscellaneous: For example: open issues

For developing use cases one should proceed incrementally. Meaning that first relevant use cases should be accurately identified as a high level and then add precision gradually.

Recommended Workflow and Tips

1. List supported actors and goals
   Review list for accuracy and completeness
2. Write stakeholders, triggers and main success scenario for each use case
   Validate that the system delivers to important stakeholders
3. Identify and list failure conditions
4. Write Failure handeling

- Start simple and focus on intent
- Write black box use cases
- Focus on actors and users of a system and their goals

A well defined task in general should fulfill the following requirements:

- performed by one stakeholder in one place at one time
- model a business event
- add measurable business value
- leaves data in a consistent state
- be more than a single step

This is called the **Elementary Business Process (EBP)**.

1.2 UML Use Case Diagrams

The Unified Modeling Language is a visual, precise design notation for software development.

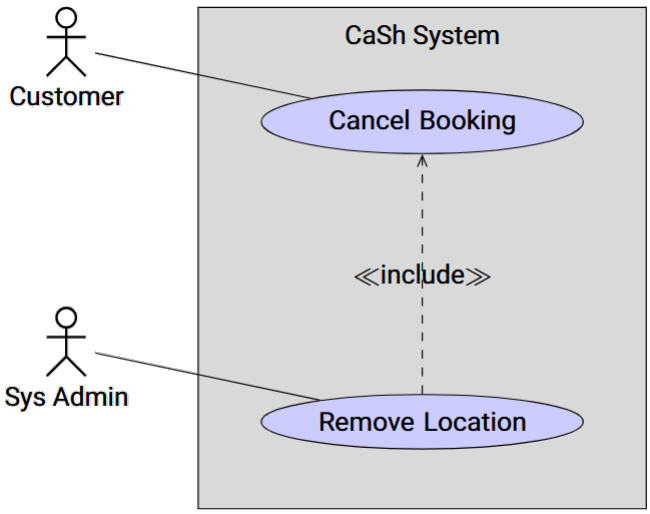
UML Use Case Diagrams Remarks

- UMLUCD is intentionally minimalist
- UMLUCD are an organizational method to improve communication and comprehension of use cases and to reduce text duplication
- UMLUCD provide a black-box view on system software
- Are only useful for early phases of use case analysis not suitable for fully dressed use cases

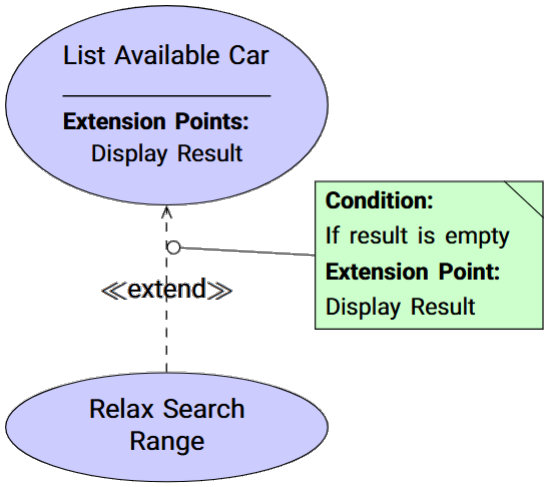
These diagrams essentially consist of system boundary (scope of the system), actors and use cases, as well as their relations.

UML Use Case Diagrams Relations

- «include»:
  - Factors out common behaviour between use cases into sub-function
  - Facilitates decomposition of large use cases and enables reuse
  - Included use cases are *always* executed
  - (Arrow goes from sub-function to base use case)
- «extend»:
  - Describes where and under what condition an extending use case extends the behaviour of the base use case
  - Most extensions do not qualify as seperate use cases Should only be used when really justified



UML Include



UML Extend