CHAPTER 8
Understanding user requirements

Objectives

- Understand two of the most commonly employed techniques for exploring user requirements: use cases and user stories
- Student could address all the functional requirements of a project by use case diagram and use case specification.

Contents

1. Use cases and user stories
2. The use case approach
3. Use cases and usage scenarios
4. Identifying use cases
5. Exploring use cases
6. Validating use cases
7. Use cases and functional requirements
8. Use case traps to avoid
9. Benefits of usage-centric requirements

Use cases and user stories

- A use case describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value
- The names of use cases are always written in the form of a verb followed by an object.
- Select strong, descriptive names to make it evident from the name that the use case will deliver something valuable for some user
- Users and actors

The use case approach

- Questions to consider
  - What will the actor use the system for?
  - Will the actor create, store, change, remove, or read data in the system?
  - Will the actor need to inform the system about external events or changes?
  - Will the actor need to be informed about certain occurrences in the system?
- Provide a brief description that elaborates the intent of the use case

Program Vacation Setting

Actor(s): Homeowner/programmer

Description: Homeowner/programmer sets lighting and alarm options for an extended stay away from home

Outline the Use Cases

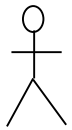
- Outline all the basic flows first
  - Basic flow is the most common path
  - What event starts the use case?
  - How does the use case end?
  - How does the use case repeat some behavior?
- Outline alternate flows
  - Are there optional situations?
  - What odd cases might happen?
  - What variants might happen?
  - What may go wrong?
  - What may not happen?
  - What kind of resources can be blocked?

Use Case Anatomy

- Mandatory elements
  - Name – Unique and descriptive
  - Brief description – Paragraph describing purpose
  - Actors – List all actors that interact with this use case
  - Flow of events – Basic flow and Alternate flows
- Optional elements
  - Pre-conditions – conditions that must be present in order for a use case to start
  - Post-condition – state of the system after a use case has run its course
  - System or subsystem – level of use case. System causes multiple subsystems to interact
  - Other stakeholder – non users
  - Special requirements – performance or throughput

Actors

- An actor is someone or something that interacts with the system
- Users
  - Homeowners are actors on HOLIS system
  - Authors are actors on a word processing system
- Other systems or applications
  - HOLIS Control Switch is an actor on the HOLIS Central Control Unit
- A device
  - Lights
  - Printer



Actor

Use cases and usage scenarios

- Use case labeling convention
- Preconditions and postconditions
- Normal flows, alternative flows, and exceptions
- Dressing the use cases
- Extend and include
- Aligning preconditions and postconditions
- Use cases and business rules

Use cases and usage scenarios

- Exploring use cases
- Validating use cases
- Use cases and functional requirements
  - Use cases only
  - Use cases and functional requirements
- Functional requirements only
- Use cases and tests
- Use case traps to avoid
- Benefits of usage-centric requirements

Use Case Specification Template*

Number	*Unique use case number*	
Name	*Brief noun-verb phrase*	
Summary	*Brief summary of use case major actions*	
Priority	*1-5 (1 = lowest priority, 5 = highest priority)*	
Preconditions	*What needs to be true before use case "executes"*	
Postconditions	*What will be true after the use case successfully "executes"*	
Primary Actor(s)	*Primary actor name(s)*	
Secondary Actor(s)	*Secondary actor name(s)*	
Trigger	*The action that causes this use case to begin*	
Main Scenario	**Step**	**Action**
	Step #	*This is the "main success scenario" or "happy path."*
	…	*Description of steps in successful use case* "execution"
	…	*This should be in a "system-user-system, etc." format.*
Extensions	**Step**	**Branching Action**
	Step #	*Alternative paths that the use case may take*
Open Issues	*Issue #*	*Issues regarding the use case that need resolution*

*Adapted from A. Cockburn, "Basic Use Case Template"

Use Case Specification Template Example

Number	1
Name	Withdraw Money
Summary	User withdraws money from one of his/her accounts
Priority	5
Preconditions	User has logged into ATM
Postconditions	User has withdrawn money and received a receipt
Primary Actor(s)	Bank Customer
Secondary Actor(s)	Customer Accounts Database

Continued …

Trigger	User has chosen to withdraw money	
Main Scenario	**Step**	**Action**
	1	System displays account types
	2	User chooses account type
	3	System asks for amount to withdraw
	4	User enters amount
	5	System debits user's account and dispenses money
	6	User removes money
	7	System prints and dispenses receipt
	8	User removes receipt
	9	System displays closing message and dispenses user's ATM card
	11	User removes card
	10	System displays welcome message
Extensions	**Step**	**Branching Action**
	5a	System notifies user that account funds are insufficient
	5b	System gives current account balance
	5c	System exits option
Open Issues	1	Should the system ask if the user wants to see the balance?