ĐẠI HỌC FPT CẦN THƠ

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

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.

Use cases and user stories

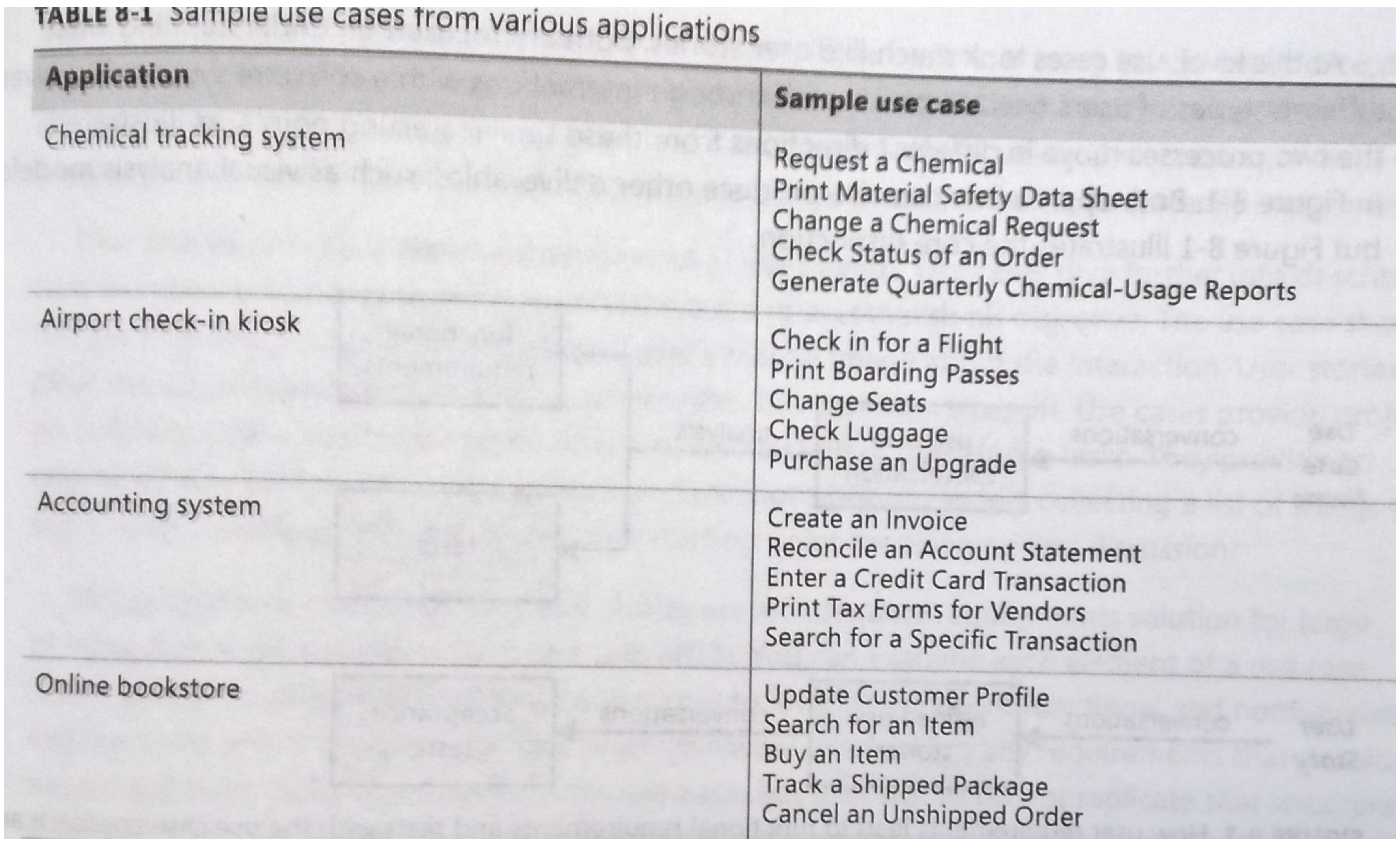


TABLE 8-1 Sample use cases from various applications

Application	Sample use case
Chemical tracking system	Request a Chemical Print Material Safety Data Sheet Change a Chemical Request Check Status of an Order Generate Quarterly Chemical-Usage Reports
Airport check-in kiosk	Check in for a Flight Print Boarding Passes Change Seats Check Luggage Purchase an Upgrade
Accounting system	Create an Invoice Reconcile an Account Statement Enter a Credit Card Transaction Print Tax Forms for Vendors Search for a Specific Transaction
Online bookstore	Update Customer Profile Search for an Item Buy an Item Track a Shipped Package Cancel an Unshipped Order

Use cases and user stories

- As used on agile project, a user story is a "short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system."
- User stories are often written by *"As a <type of user>, I want <some goal" so that <some reason>.*

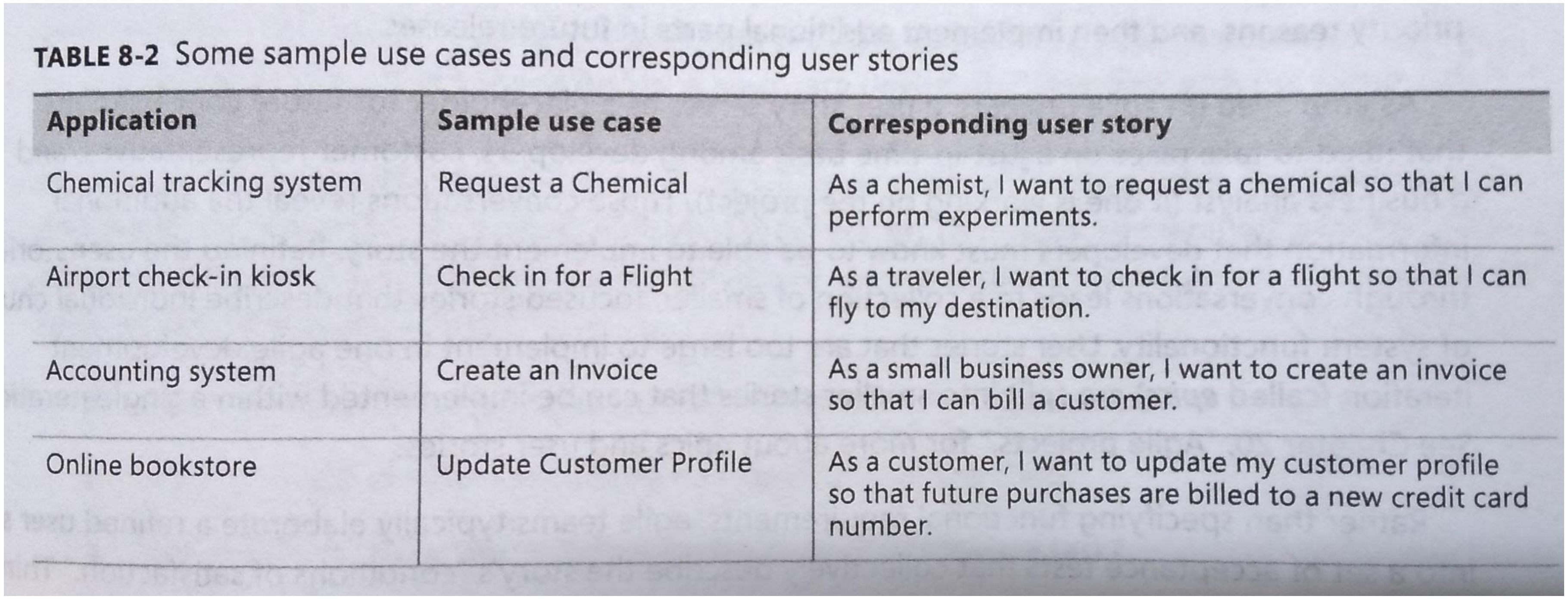


TABLE 8-2 Some sample use cases and corresponding user stories

Application	Sample use case	Corresponding user story
Chemical tracking system	Request a Chemical	As a chemist, I want to request a chemical so that I can perform experiments.
Airport check-in kiosk	Check in for a Flight	As a traveler, I want to check in for a flight so that I can fly to my destination.
Accounting system	Create an Invoice	As a small business owner, I want to create an invoice so that I can bill a customer.
Online bookstore	Update Customer Profile	As a customer, I want to update my customer profile so that future purchases are billed to a new credit card number.

The use case approach

- *Use case* describes a sequence of interactions between a system and an external actor that results in some outcome that provides value to the actor. An *actor* is a person (or another SW system) that interacts with the system to perform a use case.
- 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

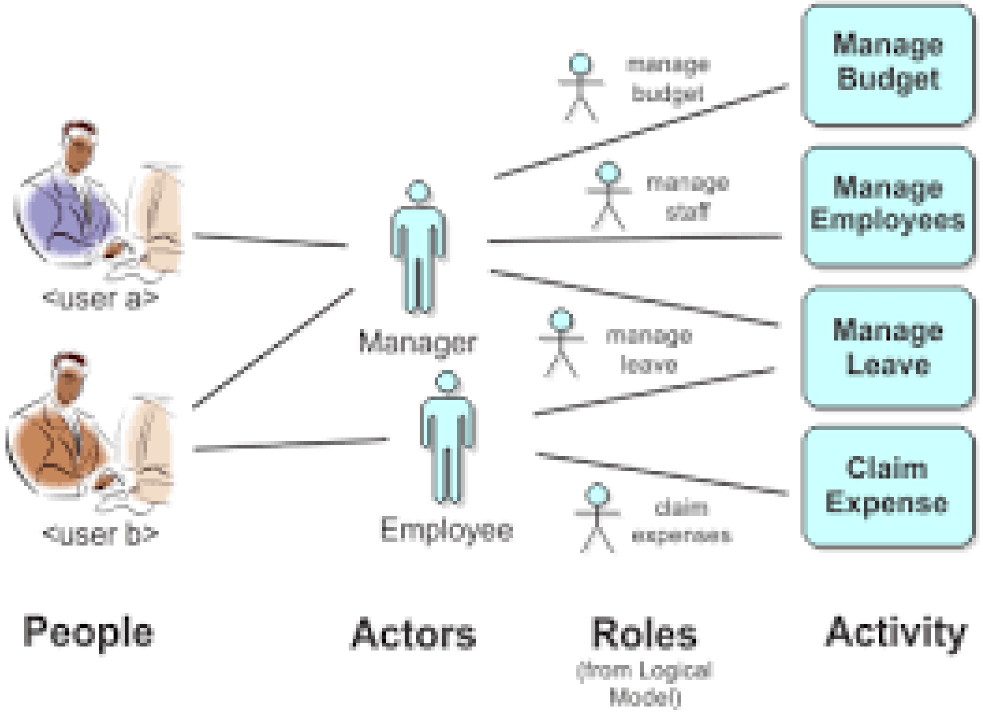
Actor(s): Homeowner/programmer

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

FPT Education
FPT UNIVERSITY

Users and Actors

- Users:
  - A human user has a collection of hats available, each label with the name of an actor that the system will recognize as participating in certain use cases.
- Actors
  - When the user want to perform a certain action with the system, he puts on the appropriate hat. The system will recognize that person as the labeled actor when he launches whatever use case he's interested in performing.



Use Case Diagrams

- *Use case diagrams* provide a high-level visual representation of the user requirements.
- *Arrows* from an actor connect to the use cases.
- An arrow from an actor to a use case indicates that he is the *primary actor* for the use case.
- *The primary actor* initiates the use case and derives the main value from it. An arrow goes from a use case to a *secondary actor,* who participates somehow in the successful execution of the use case.
- *The arrows* in the use case diagrams simply indicate the connections between actors and use cases in which they participate.

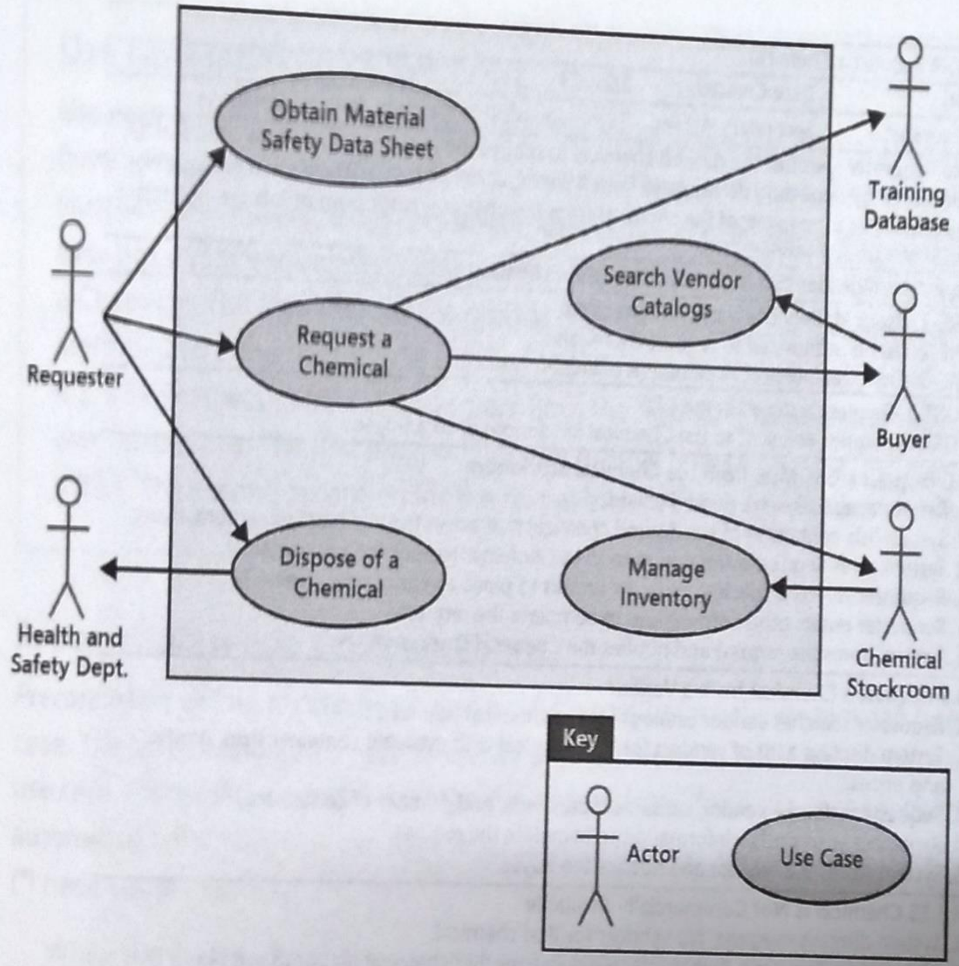
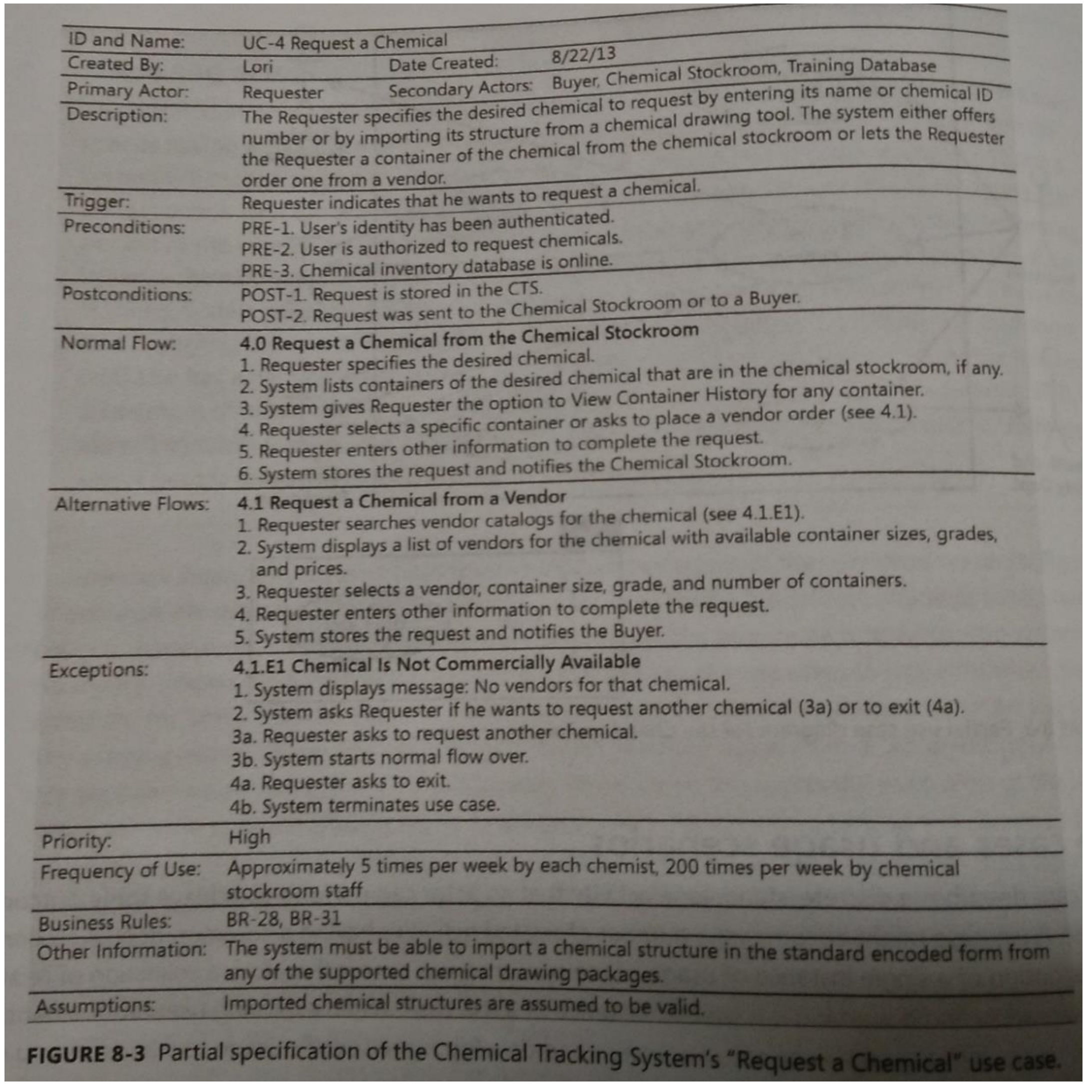


FIGURE 8-2 Partial use case diagram for the Chemical Tracking System.

Use cases and usage scenarios

- Use case describes a discrete, standalone activity that an actor can perform to achieve some outcome of value.
- A use case might encompass a number of related activities having a common goal.
- A *scenario* is a description of a single instance of usage of the system.



ID and Name:	UC-4 Request a Chemical		
Created By:	Lori	Date Created:	8/22/13
Primary Actor:	Requester	Secondary Actors:	Buyer, Chemical Stockroom, Training Database
Description:	The Requester specifies the desired chemical to request by entering its name or chemical ID number or by importing its structure from a chemical drawing tool. The system either offers the Requester a container of the chemical from the chemical stockroom or lets the Requester order one from a vendor.		
Trigger:	Requester indicates that he wants to request a chemical.		
Preconditions:	PRE-1. User's identity has been authenticated. PRE-2. User is authorized to request chemicals. PRE-3. Chemical inventory database is online.		
Postconditions:	POST-1. Request is stored in the CTS. POST-2. Request was sent to the Chemical Stockroom or to a Buyer.		
Normal Flow:	**4.0 Request a Chemical from the Chemical Stockroom** 1. Requester specifies the desired chemical. 2. System lists containers of the desired chemical that are in the chemical stockroom, if any. 3. System gives Requester the option to View Container History for any container. 4. Requester selects a specific container or asks to place a vendor order (see 4.1). 5. Requester enters other information to complete the request. 6. System stores the request and notifies the Chemical Stockroom.		
Alternative Flows:	**4.1 Request a Chemical from a Vendor** 1. Requester searches vendor catalogs for the chemical (see 4.1.E1). 2. System displays a list of vendors for the chemical with available container sizes, grades, and prices. 3. Requester selects a vendor, container size, grade, and number of containers. 4. Requester enters other information to complete the request. 5. System stores the request and notifies the Buyer.		
Exceptions:	**4.1.E1 Chemical Is Not Commercially Available** 1. System displays message: No vendors for that chemical. 2. System asks Requester if he wants to request another chemical (3a) or to exit (4a). 3a. Requester asks to request another chemical. 3b. System starts normal flow over. 4a. Requester asks to exit. 4b. System terminates use case.		
Priority:	High		
Frequency of Use:	Approximately 5 times per week by each chemist, 200 times per week by chemical stockroom staff		
Business Rules:	BR-28, BR-31		
Other Information:	The system must be able to import a chemical structure in the standard encoded form from any of the supported chemical drawing packages.		
Assumptions:	Imported chemical structures are assumed to be valid.		

FIGURE 8-3 Partial specification of the Chemical Tracking System's "Request a Chemical" use case.

Use Case Anatomy

- Name – Unique and descriptive.
- Brief description – Paragraph describing purpose.
- Actors – List all actors that interact with this use case.
- Descriptions describe the purpose of the use case.
- Trigger initiates execution of the use case.
- 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
- Flow of events – Normal flow and Alternate flows

Use Case Anatomy

- ***Normal flow*** of events is also called the *main flow, primary flow, basic flow, normal course, primary scenario,…* It's written as a numbered list of steps, indicating which entity – the system or a specific actor – performs each step.
- ***Alternative flow*** or *secondary flow* deliver the same biz outcome as the normal flow but represents less common or lower-priority variations in the specifics of the task or how it is accomplished.
- ***Exception:*** conditions that have the potential to prevent a use case from succeeding are called *exceptions.* Exceptions describe anticipated error conditions that could occur during execution of the use case and how they are to be handled. In some cases, the user can recover, but in some other cases, the use must terminate the execution without reaching its success conditions.

Use Case Anatomy

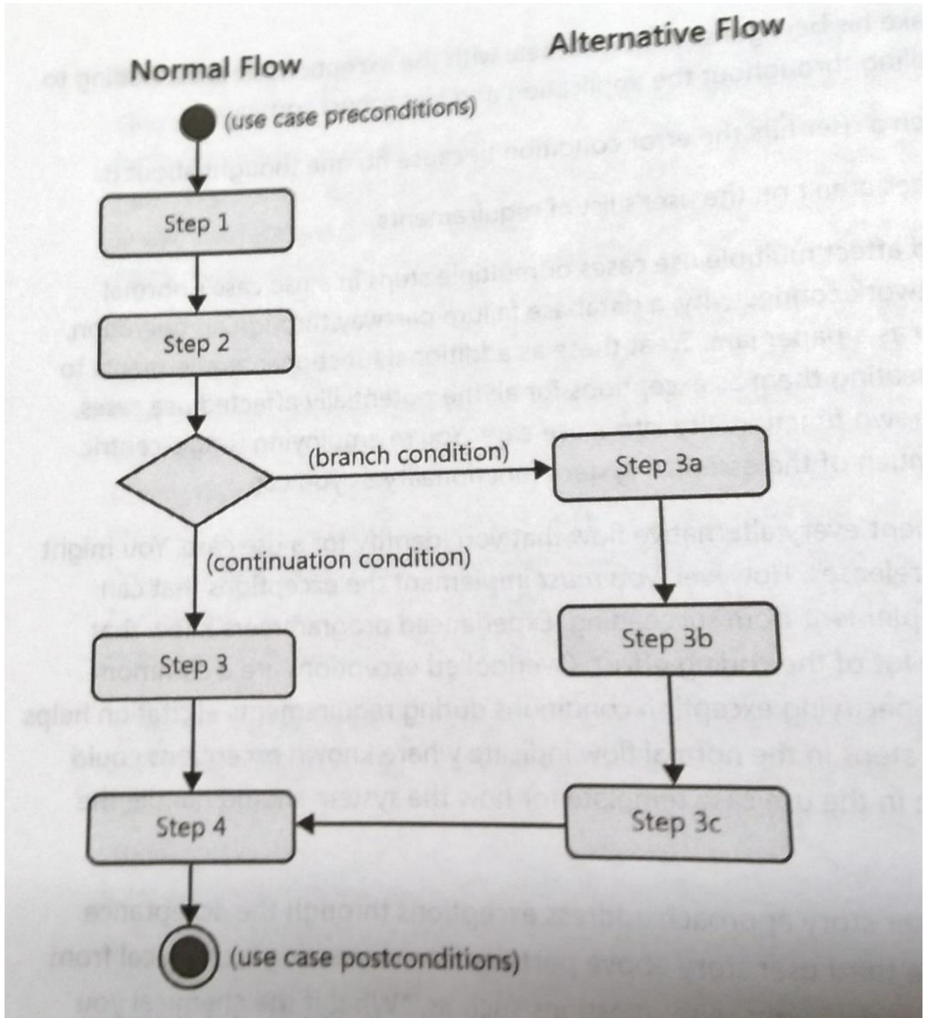


FIGURE 8-4 An activity diagram illustrating the step sequence in the

Use cases and usage scenarios

- Exploring use cases: Identify the actors → Estimate the frequency of using system → Define the preconditions/postconditions → The resulting sequence of actor actions and system responses became the flow of events → BA captures the actor actions and their corresponding system responses on sticky notes in the meeting/workshop to discuss about use cases.

- Validating use cases: 1 day or 2 after each workshop, the BA send the use cases and functional requirements to the workshop participants to review. These informal reviews reveal many errors: undiscovered alternative flows, new exceptions, incorrect functional requirements,… The mental relaxation that comes after a day or 2 away allows people to examine their earlier work from a fresh perspective.

Use cases and usage scenarios

- Use cases and functional requirements: *Big different!*
  - Use cases only: Use cases describe the user's perspective, looking at the externally visible of the system. They don't contain all the information that a developer need to write the SW.
  - Many functional requirements fall right out of the dialog steps between the actor and the system.

- Use cases and tests: If you write both detailed use cases specifications and functional requirements, you might notice some duplication, particularly around the normal flow. You also have to write the acceptance tests to determine if the system properly handles the basic behavior of the use case, alternative success paths, and the various things that could go wrong.

Use cases and usage scenarios

- Use case traps to avoid: There many ways to go astray when applying the use case approach:
  - Too many use cases: You’ll typically have many more use cases than business requirements and features, but many more functional requirements than use cases.
  - Highly complex use cases: If you review a use case with 4 dense pages of dialog steps, with a lot of embedded logic and branching conditions, it’s incomprehensible.
  - Including design in the use case: Use cases should focus on what the users need to accomplish with the system’s help, not on how the screen will look.
  - Including data definitions in the use cases: Use case explorations naturally stimulate data discussions, thinking about what data elements serve as inputs and outputs during the interaction.
  - Use cases that users don’t understand: If users can not relate a use case to their biz processes or goals, there’s a problem.

Use cases and usage scenarios

- Benefits of usage-centric requirements: The usage-centric approach leads to functionality that will allow the user to perform certain known tasks. That helps prevents “orphan functionality” that seems like a good idea but that no one uses because it doesn’t relate directly to user goals.
- A use case or user story could be of high priority for several reasons:
  - It describes part of a core biz process that the system enables.
  - Many users will use it frequently.
  - A favored user class requested it.
  - It’s required for regulatory compliance.
  - Other system functions depend on its presence.

QS STARS RATED FOR EXCELLENCE 2012

FPT Education

FPT UNIVERSITY

THE END
THANK YOU!