Chapter 3. Requirements Specification and Modelling with UML

Reyes Grangel Seguer

EI1030. Analysis of Inf. Systems / EI1032. Software Analysis / MT1046. Integrated Inf. Systems
Computer Engineering / Computational Mathematics
Dept. of Computer Systems and Languages

UNIVERSITAT JAUME•I

February 10, 2019

Outline

Outline

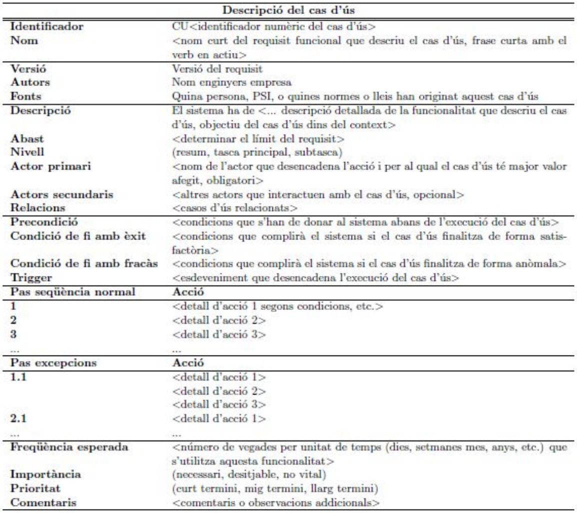
Objectives

- To document requirements
  - To show only external behaviours of the system
  - To show the vision of the future from user viewpoint
  - To describe exhaustively the conditions that needs to accomplish the system to achieve the defined objectives and scope, and taking into account restrictions

Basic Elements

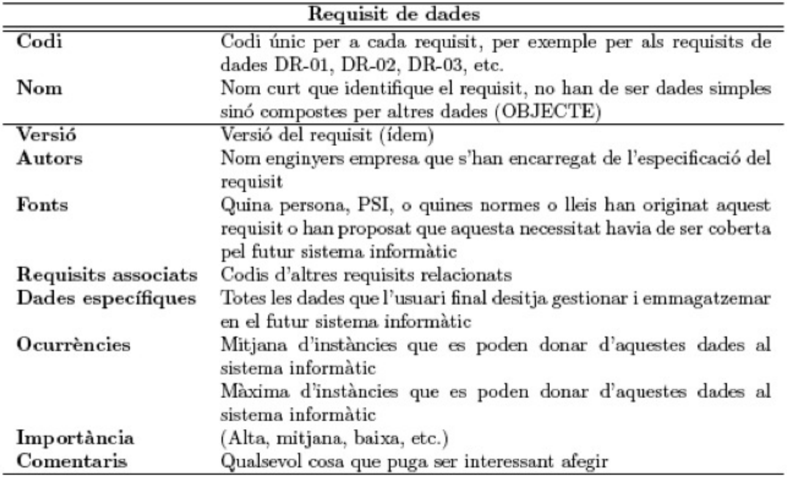
- **Identification**: name, objective, description, etc.
- **Description of situations**: trigger, actions, scenarios, etc.
- **Comments**

Template



Descripció del cas d'ús	
Identificador	CU<identificador numèric del cas d'ús>
Nom	<nom curt del requisit funcional que descriu el cas d'ús, frase curta amb el verb en actiu>
Versió	Versió del requisit
Autors	Nom enginyers empresa
Fonts	Quina persona, PSI, o quines normes o lleis han originat aquest cas d'ús
Descripció	El sistema ha de <... descripció detallada de la funcionalitat que descriu el cas d'ús, objectiu del cas d'ús dins del context>
Abast	<determinar el límit del requisit>
Nivell	(resum, tasca principal, subtasca)
Actor primari	<nom de l'actor que desencadena l'acció i per al qual el cas d'ús té major valor afegit, obligatori>
Actors secundaris	<altres actors que interactuen amb el cas d'ús, opcional>
Relacions	<casos d'ús relacionats>
Precondició	<condicions que s'han de donar al sistema abans de l'execució del cas d'ús>
Condició de fi amb èxit	<condicions que complirà el sistema si el cas d'ús finalitza de forma satisfactòria>
Condició de fi amb fracàs	<condicions que complirà el sistema si el cas d'ús finalitza de forma anòmala>
Trigger	<esdeveniment que desencadena l'execució del cas d'ús>
Pas seqüència normal	**Acció**
1	<detall d'acció 1 segons condicions, etc.>
2	<detall d'acció 2>
3	<detall d'acció 3>
...	...
Pas excepcions	**Acció**
1.1	<detall d'acció 1>
	<detall d'acció 2>
	<detall d'acció 3>
2.1	<detall d'acció 1>
...	...
Freqüència esperada	<número de vegades per unitat de temps (dies, setmanes mes, anys, etc.) que s'utilitza aquesta funcionalitat>
Importància	(necessari, desitjable, no vital)
Prioritat	(curt termini, mig termini, llarg termini)
Comentaris	<comentaris o observacions addicionals>

Template

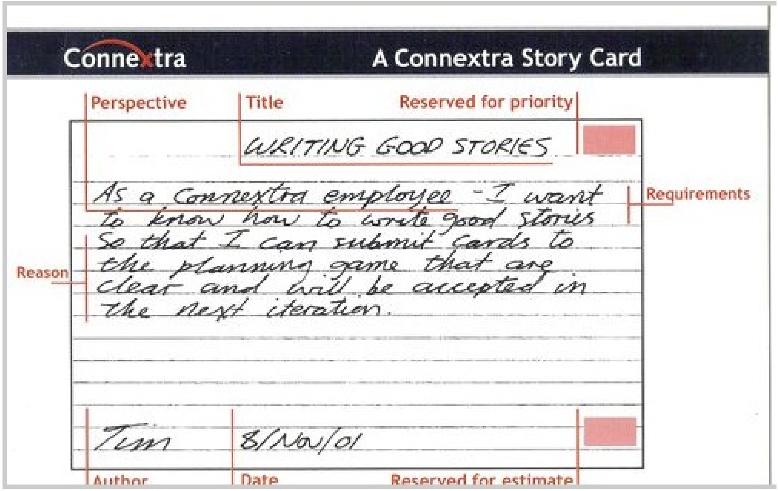


Requisit de dades	
Codi	Codi únic per a cada requisit, per exemple per als requisits de dades DR-01, DR-02, DR-03, etc.
Nom	Nom curt que identifique el requisit, no han de ser dades simples sinó compostes per altres dades (OBJECTE)
Versió	Versió del requisit (ídem)
Autors	Nom enginyers empresa que s'han encarregat de l'especificació del requisit
Fonts	Quina persona, PSI, o quines normes o lleis han originat aquest requisit o han proposat que aquesta necessitat havia de ser coberta pel futur sistema informàtic
Requisits associats	Codis d'altres requisits relacionats
Dades específiques	Totes les dades que l'usuari final desitja gestionar i emmagatzemar en el futur sistema informàtic
Ocurrències	Mitjana d'instàncies que es poden donar d'aquestes dades al sistema informàtic
	Màxima d'instàncies que es poden donar d'aquestes dades al sistema informàtic
Importància	(Alta, mitjana, baixa, etc.)
Comentaris	Qualsevol cosa que puga ser interessant afegir

Template

Requisit de qualitat	
Codi	Codi únic per a cada requisit, per exemple per als requisits de qualitat QR-01, QR-02, QR-03, etc.
Nom	Nom curt que identifique el requisit
Versió	Versió del requisit (ídem)
Autors	Nom enginyers empresa que s'han encarregat de l'especificació del requisit
Fonts	Quina persona, PSI, o quines normes o lleis han originat aquest requisit o han proposat que aquesta necessitat havia de ser coberta pel futur sistema informàtic
Requisits associats	Codis d'altres requisits relacionats
Descripció	El sistema ha de permetre ..., El sistema permetrà ..., El sistema ha de proporcionar ..., etc.
Importància	(Alta, mitjana, baixa, etc.)
Comentaris	Qualsevol cosa que puga ser interessant afegir

Origin: Connextra 2001



Connextra A Connextra Story Card

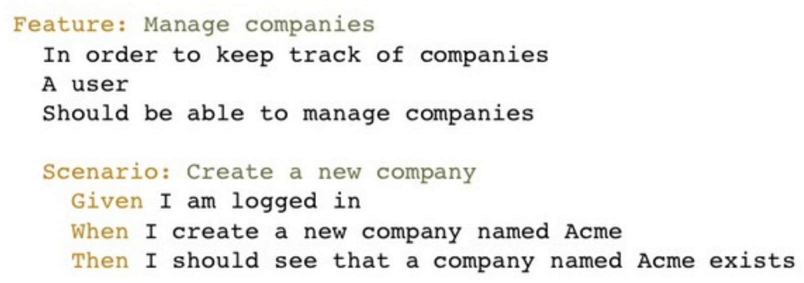
WRITING GOOD STORIES

As a Connextra employee - I want to know how to write good stories So that I can submit cards to the planning game that are clear and will be accepted in the next iteration.

Tim 8/Nov/01

Perspective
Title
Reserved for priority
Requirements
Reason
Author
Date
Reserved for estimate

Template: Gherkin Specification



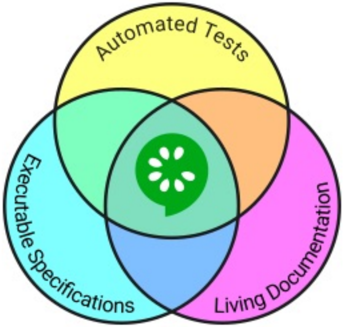
```
Feature: Manage companies
  In order to keep track of companies
  A user
  Should be able to manage companies

  Scenario: Create a new company
    Given I am logged in
    When I create a new company named Acme
    Then I should see that a company named Acme exists
```

Template: Gherkin Specification

- **Given**: to show the system in the desired state before the user (or external system) to interact with the system
- **When**: to describe the action taken by the user
- **Then**: to detail the results, these observations should be made from the viewpoint of business and the user value

Template: Cucumber Tool



Types

- Epic
- Story

Best Practices to Write User Stories

INVEST principles

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testeable

http://www.intergrupo.com/blog/mobile/como-escribir-historias-de-usuario.aspx

Comparison with Use Cases

- **Shorter**: a user story is a brief description of functionality as viewed by the user, it is not a sequence of actions
- **Less specific**: a user story provides less details and information in its description that a use case
- **Written for non-specialised users**: a user story can be more easily written by a user or customer
- **Easier to maintain**: a user story is more readable than use cases
- ...
- http://www.agile-ux.com/2009/01/23/use-cases-user-stories-so-precious-but-not-the-same/

Web Links

- http://www.extremeprogramming.org/rules/userstories.html
- http://www.agilemodeling.com/artifacts/userStory.htm
- http://www.pmoinformatica.com/2012/10/plantillas-scrum-historias-de-usuario.html#more
- http://www.betterprojects.net/2011/03/user-story-template.html
- http://www.slideshare.net/bkeepers/behavior-driven-development-with-cucumber-presentation

Tools

- https://es.atlassian.com/software/jira
- https://www.pivotaltracker.com
- https://trello.com/
- https://github.com/
- https://cukes.info/

Outline

Origin

- **Use Cases**
    - Early 90
    - Ivar Jacobson
    - Initial idea of scenario
- **Use Case Diagram**: Behaviour Diagram in UML2

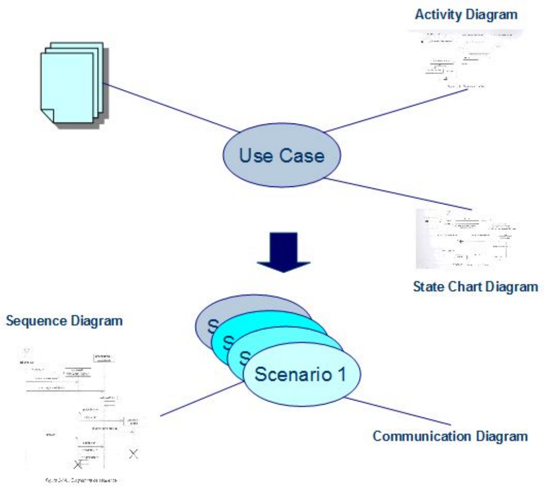
Objective

- To document the behaviour of a system
- From user point of view
- A user is someone or something external to the system
- e.g. a person, another computer system, a HW device, etc.

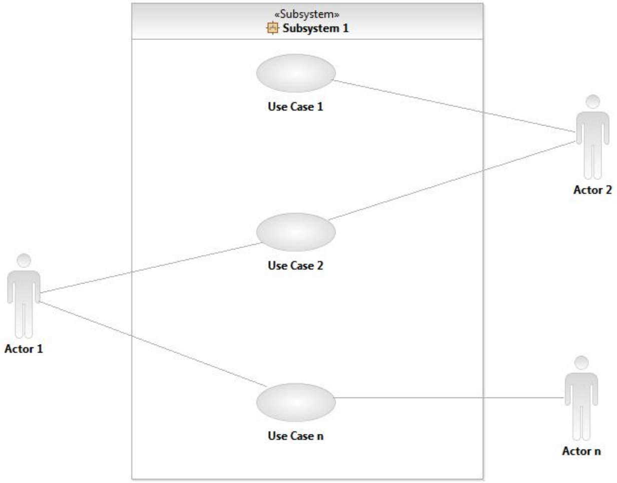
Features

- Easy to understand even by non-specialised users
- Very intuitive
- Simple

Refining the Use Case Diagram



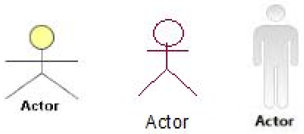
Example: Use Case Diagram



Actor

What does it represent?

A kind of role carried out by an external entity to the system that interact with it exchanging signals and data



Actor

- One role can match only one part of an entity
- One entity can adopt several roles
- One role can be adopted by different entities

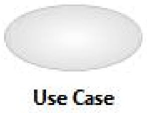
How to identify actors?

- Actor should interact directly with the system
- That means, to consult or to modify information of the system
- Actor is external to the system
- Actors can be
  - **Human actors**
  - **Non-Human actors**

Use Case

What does it represent?

One functionality that have to be supported by the computer system in development



Use Case

- It is named: **verb in INFINITIVE + OBJECT**
- It is triggered by the **main actor** (compulsory)
- It can have **secondary actors** (optional)
- One instance of a Use Case is called '**Scenario**'

Concept of Scenario

What is an scenario?

It is one possible and specific interaction between system and some people or devices in its specific roles

How to identify use cases?

- Identifying ...
  - A sequence of actions executed by the system, which produce an observable result with added value for one actor
  - A service provided by the system
  - An interaction between actors and the system
  - A set of possible scenarios with the same objective
  - A functional requirement of the system to develop
  - Only essential behaviour of the system

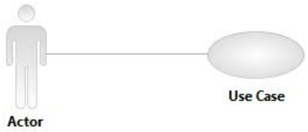
Problems identifying use cases

- **Use case too generic**: result not observable
- **Use case too specific**: not added value for the result
- To describe **how** and not **what** do the use case

Relationship

What does it represent?

The connection between one actor and one use case



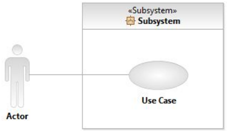
Relationship

- It is not named
- It can be navigable
- The arrow show the sense of the information flow

Boundary of the system

What does it represent?

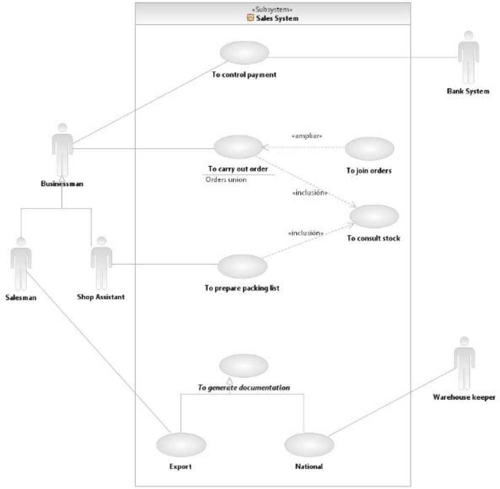
The scope of the system



Boundary of the system

- It is used for modelling complex systems
- To indicate the specific system that is modelled
- It is omitted in simple systems

Example: Use Case Diagram

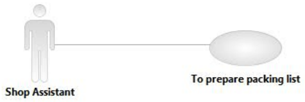


Kinds of Relationships

- Association
- Generalisation
- Dependency

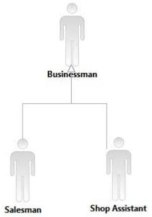
Association

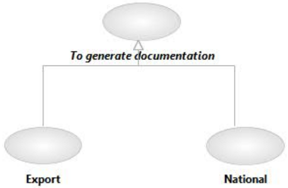
- Between one actor and one use case
- Represent communication between them
- To send and receive messages



Generalisation

- Between actors, or between use cases
- Represent an structural relationship
- To answer to the question ‘is a kind of’



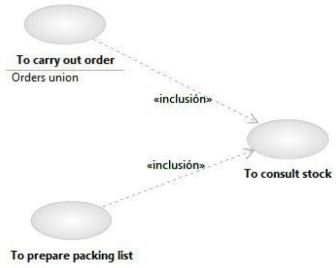


Dependency

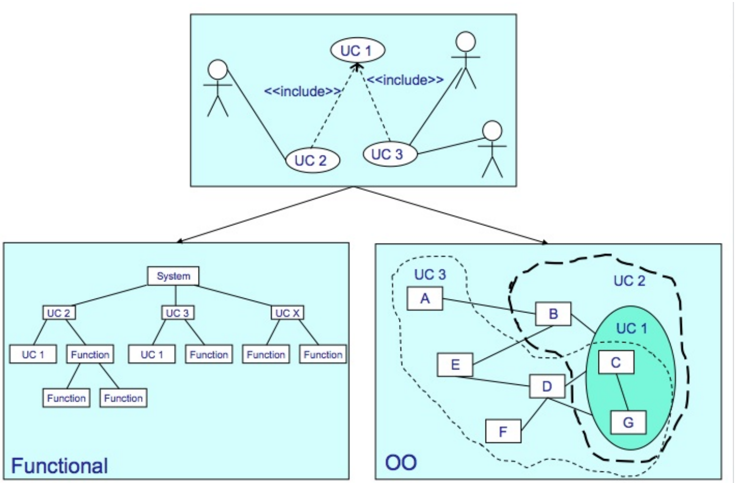
- Between use cases
- Represent a use relationship
- To show that one element depends on the other one
- There are some **stereotyped dependencies**
  - «include»
  - «extend»

«include»

- To represent that a base use case incorporates the behaviour of another use case
- To represent reuse
- To avoid repeat the same functionality inside different use cases

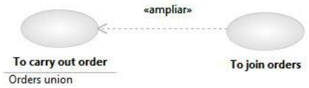


«include»


UC 1
<<include>>
<<include>>
UC 2
UC 3
System
UC 2
UC 3
UC X
UC 1
Function
UC 1
Function
Function
Function
Function
Function
Functional
UC 3
A
UC 2
B
UC 1
C
E
D
G
F
OO

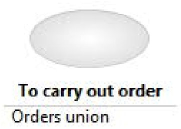
«extend»

- To represent a part of a use case that can be optional from user point of view
- To represent optionality
- Optional behaviour is inserted in an specific point called **extension point**

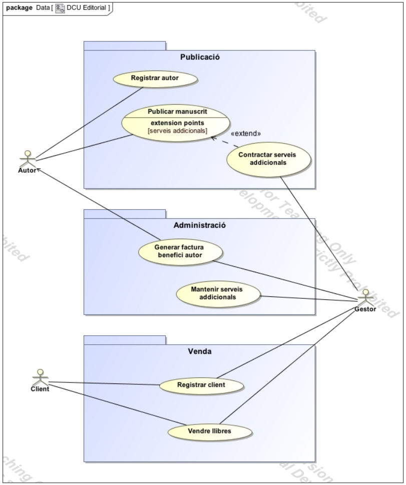


Extension Point

- It is a reference in a use case, where it is possible to insert the behaviour of another use case
- It should have a unique name inside the use case
- **Notation**: <extension point> ::= <name> [: <explanation>]



Example


package Data [DCU Editorial]
Publicació
Registrar autor
Publicar manuscrit
extension points
[serveis addicionals]
«extend»
Contractar serveis addicionals
Autor
Administració
Generar factura benefici autor
Mantenir serveis addicionals
Gestor
Venda
Client
Registrar client
Vendre llibres

Concept

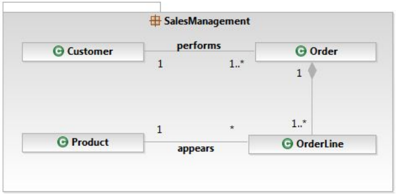
What is a package?

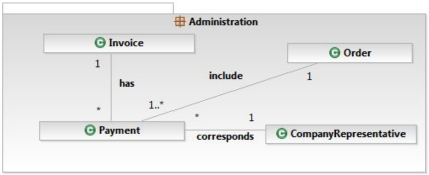
A set of elements of the model and a names space for these elements

- Used for grouping elements with several purposes
- Used any UML diagram for grouping for example
  - **Class Diagram**: classes
  - **Use Case Diagram**: use cases
  - **Components Diagram**: components
  - **Deployment Diagram**: nodes

Notation: elemental name

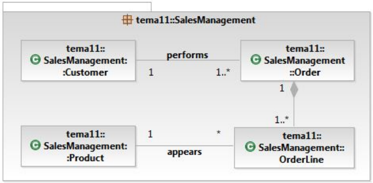
Non qualified name or incomplete name

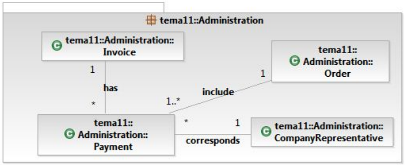




Notation: names space

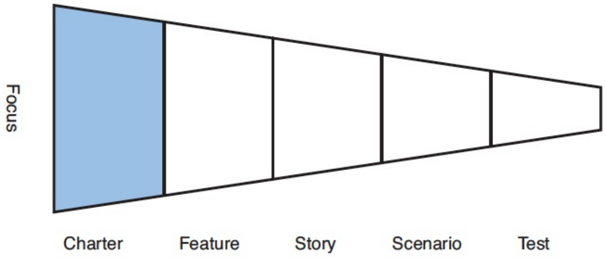
Path name, qualified name or complete name


tema11::SalesManagement
tema11::SalesManagement::Customer
performs
1
1..*
tema11::SalesManagement::Order
1
1..*
tema11::SalesManagement::Product
1
*
appears
tema11::SalesManagement::OrderLine

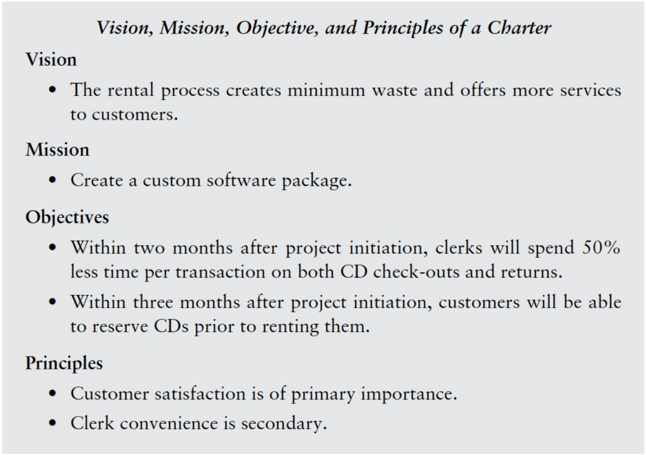

tema11::Administration
tema11::Administration::Invoice
1
has
*
1..*
include
1
tema11::Administration::Order
tema11::Administration::Payment
*
1
corresponds
tema11::Administration::CompanyRepresentative

Outline

Charter



Charter: Example



Vision, Mission, Objective, and Principles of a Charter

Vision

- The rental process creates minimum waste and offers more services to customers.

Mission

- Create a custom software package.

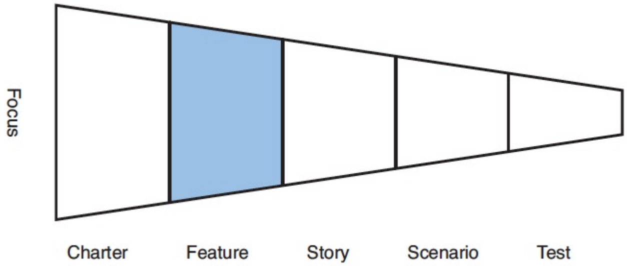
Objectives

- Within two months after project initiation, clerks will spend 50% less time per transaction on both CD check-outs and returns.
- Within three months after project initiation, customers will be able to reserve CDs prior to renting them.

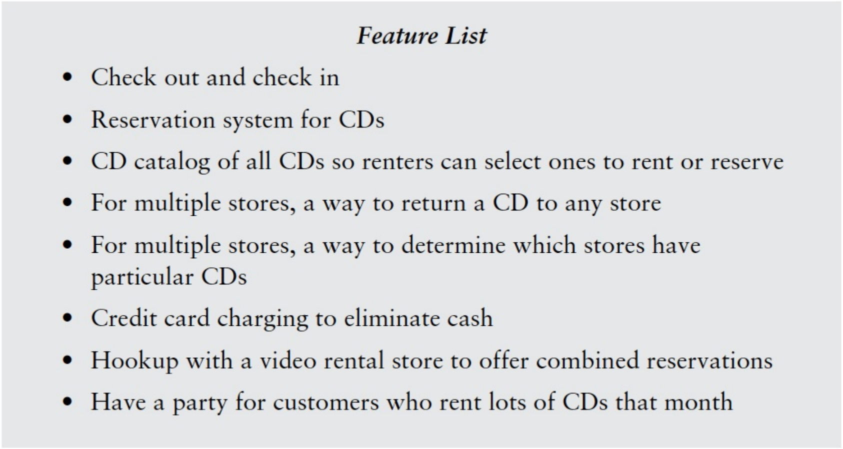
Principles

- Customer satisfaction is of primary importance.
- Clerk convenience is secondary.

Features



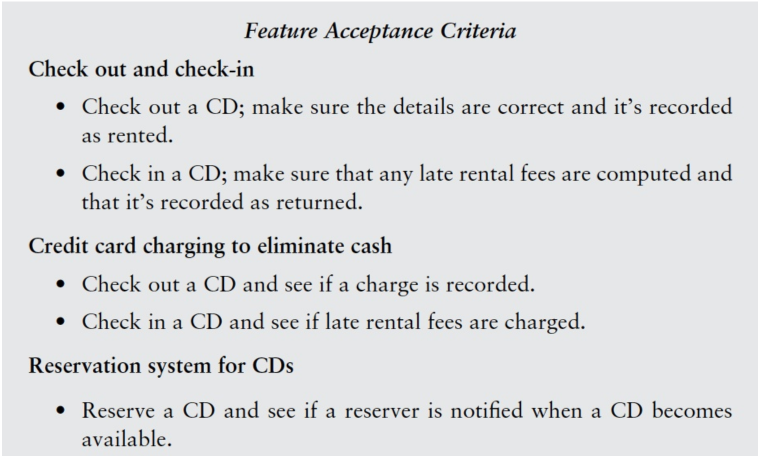
Features: Example



Feature List

- Check out and check in
- Reservation system for CDs
- CD catalog of all CDs so renters can select ones to rent or reserve
- For multiple stores, a way to return a CD to any store
- For multiple stores, a way to determine which stores have particular CDs
- Credit card charging to eliminate cash
- Hookup with a video rental store to offer combined reservations
- Have a party for customers who rent lots of CDs that month

Features: Test



Feature Acceptance Criteria

Check out and check-in

- Check out a CD; make sure the details are correct and it's recorded as rented.
- Check in a CD; make sure that any late rental fees are computed and that it's recorded as returned.

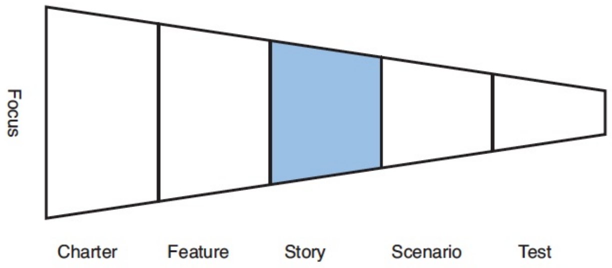
Credit card charging to eliminate cash

- Check out a CD and see if a charge is recorded.
- Check in a CD and see if late rental fees are charged.

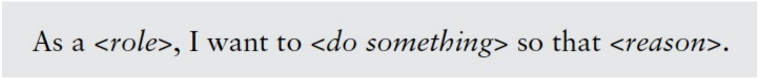
Reservation system for CDs

- Reserve a CD and see if a reserver is notified when a CD becomes available.

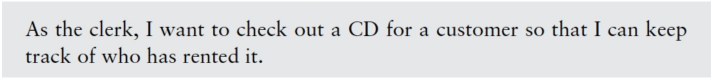
User Stories



User Stories: Template

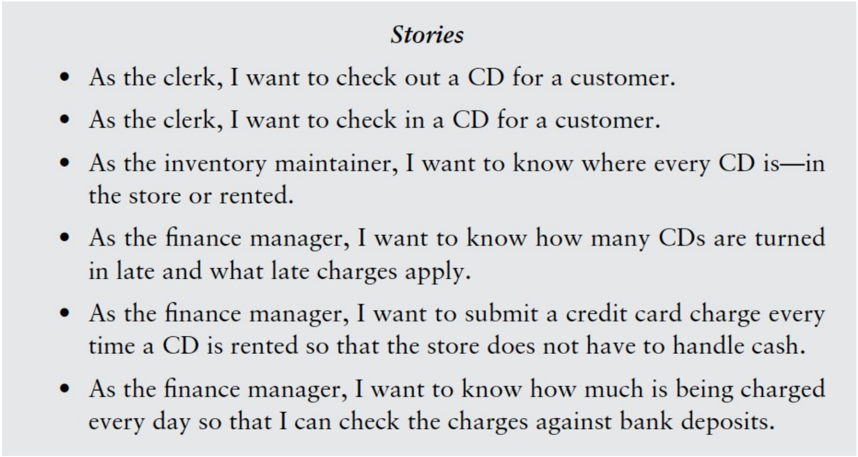

As a <*role*>, I want to <*do something*> so that <*reason*>.

User Stories: One Complete Example



As the clerk, I want to check out a CD for a customer so that I can keep track of who has rented it.

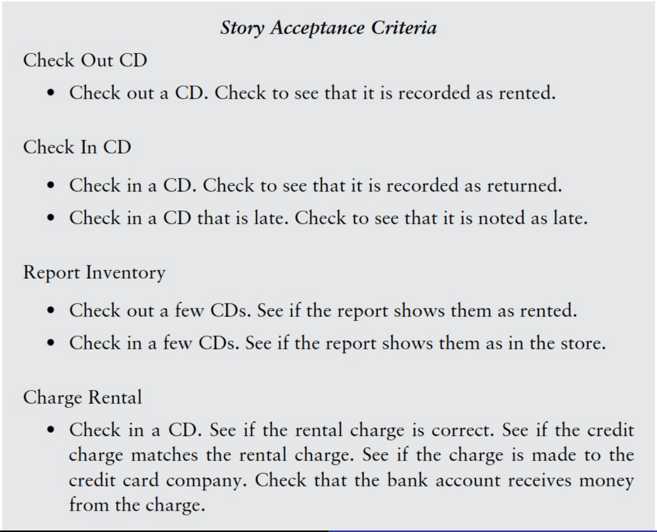
User Stories: Some Incomplete Examples



Stories

- As the clerk, I want to check out a CD for a customer.
- As the clerk, I want to check in a CD for a customer.
- As the inventory maintainer, I want to know where every CD is—in the store or rented.
- As the finance manager, I want to know how many CDs are turned in late and what late charges apply.
- As the finance manager, I want to submit a credit card charge every time a CD is rented so that the store does not have to handle cash.
- As the finance manager, I want to know how much is being charged every day so that I can check the charges against bank deposits.

User Stories: Test



Story Acceptance Criteria

Check Out CD

- Check out a CD. Check to see that it is recorded as rented.

Check In CD

- Check in a CD. Check to see that it is recorded as returned.
- Check in a CD that is late. Check to see that it is noted as late.

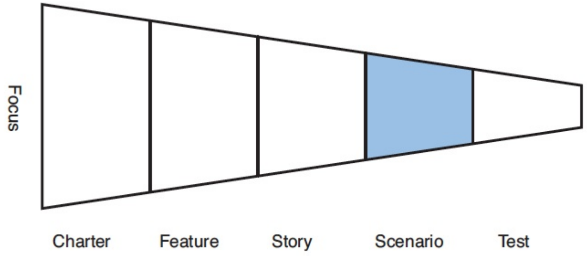
Report Inventory

- Check out a few CDs. See if the report shows them as rented.
- Check in a few CDs. See if the report shows them as in the store.

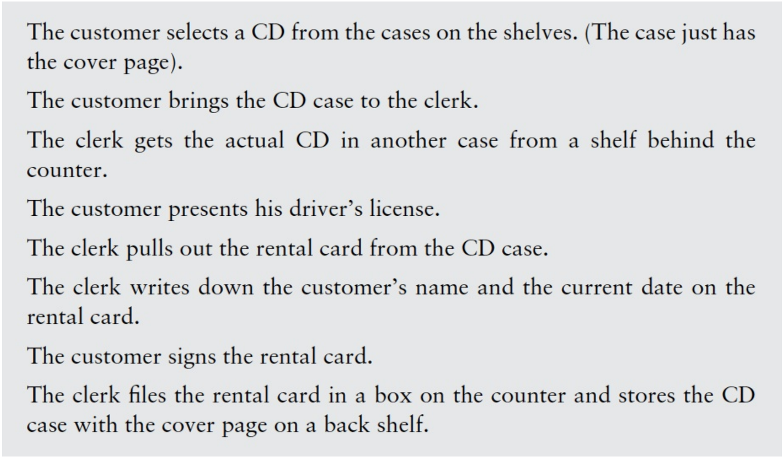
Charge Rental

- Check in a CD. See if the rental charge is correct. See if the credit charge matches the rental charge. See if the charge is made to the credit card company. Check that the bank account receives money from the charge.

Use Cases



Use Cases: Business Example



The customer selects a CD from the cases on the shelves. (The case just has the cover page).

The customer brings the CD case to the clerk.

The clerk gets the actual CD in another case from a shelf behind the counter.

The customer presents his driver's license.

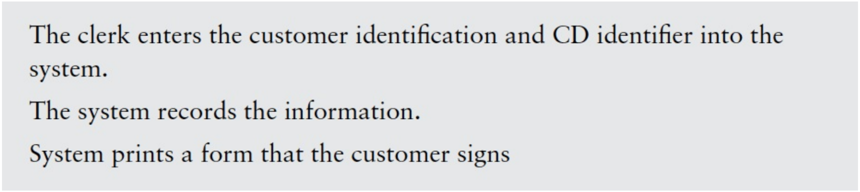
The clerk pulls out the rental card from the CD case.

The clerk writes down the customer's name and the current date on the rental card.

The customer signs the rental card.

The clerk files the rental card in a box on the counter and stores the CD case with the cover page on a back shelf.

Use Cases: Computer Example

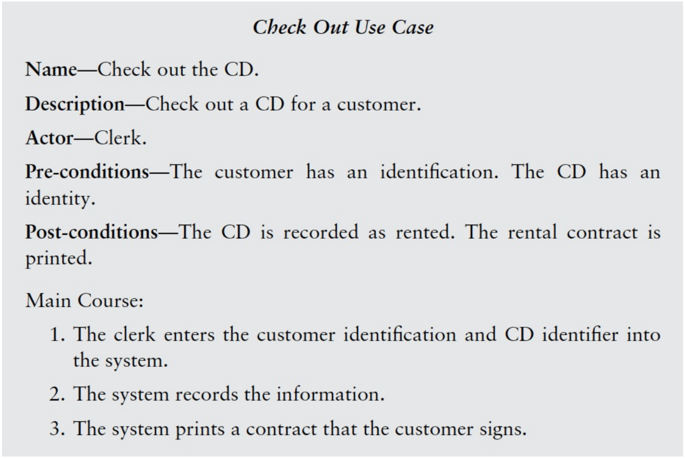


The clerk enters the customer identification and CD identifier into the system.

The system records the information.

System prints a form that the customer signs

Use Cases: Simple Template



Check Out Use Case

Name—Check out the CD.

Description—Check out a CD for a customer.

Actor—Clerk.

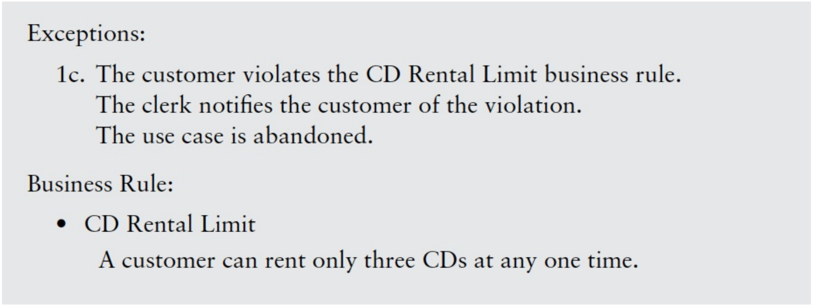
Pre-conditions—The customer has an identification. The CD has an identity.

Post-conditions—The CD is recorded as rented. The rental contract is printed.

Main Course:

1. The clerk enters the customer identification and CD identifier into the system.
2. The system records the information.
3. The system prints a contract that the customer signs.

Use Cases: Template with more Information



Exceptions:

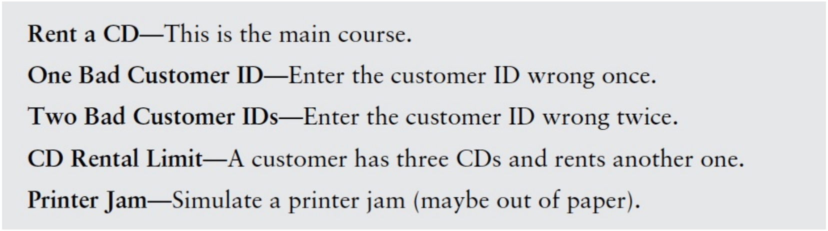
1c. The customer violates the CD Rental Limit business rule.
The clerk notifies the customer of the violation.
The use case is abandoned.

Business Rule:

- CD Rental Limit

  A customer can rent only three CDs at any one time.

Use Cases: Test



Rent a CD—This is the main course.

One Bad Customer ID—Enter the customer ID wrong once.

Two Bad Customer IDs—Enter the customer ID wrong twice.

CD Rental Limit—A customer has three CDs and rents another one.

Printer Jam—Simulate a printer jam (maybe out of paper).