Use Cases

PA14[13]5

Mikael Svahnberg[1]

2016-03-16

[1] Mikael.Svahnberg@bth.se

Use Case: Basic Notation

- Narrative Document
- Involves Actors and Events
- Illustrates requirements in a story, in a timeline.
- Considers the system as a black box
- Different levels:
  - Brief (High Level)
  - Fully Dressed (Expanded)

Example

Point Of Sale System

A point of sale system (PoS, Don't look it up in UrbanDictionary) is a computerised applicaion used to record sales and handle payments. It is typically used in a retail store. It includes hardware components such as a computer and a bar code scanner, and sofware to run the system.

Functions in Example

- Basic
  - Record the current sale
  - Calculate current sales total
  - Reduce inventory after sale
- Payment
  - Handle Cash Payment
  - Handle Credit Payment
  - Log credit Payment
- ...
- Don't forget quality attributes
  - Response Time (Price will appear within 5 secs when recording a sold item)
  - Fault Tolerance (Must record payments to accounts within 24h)
  - System Requirements (Windows 10 or later)
- Interface Requirements
  - Methaphore (Shopping Basket)
  - Infrastructure (Platform: Windows 10, Database: MySQL, Programming Language: C++)

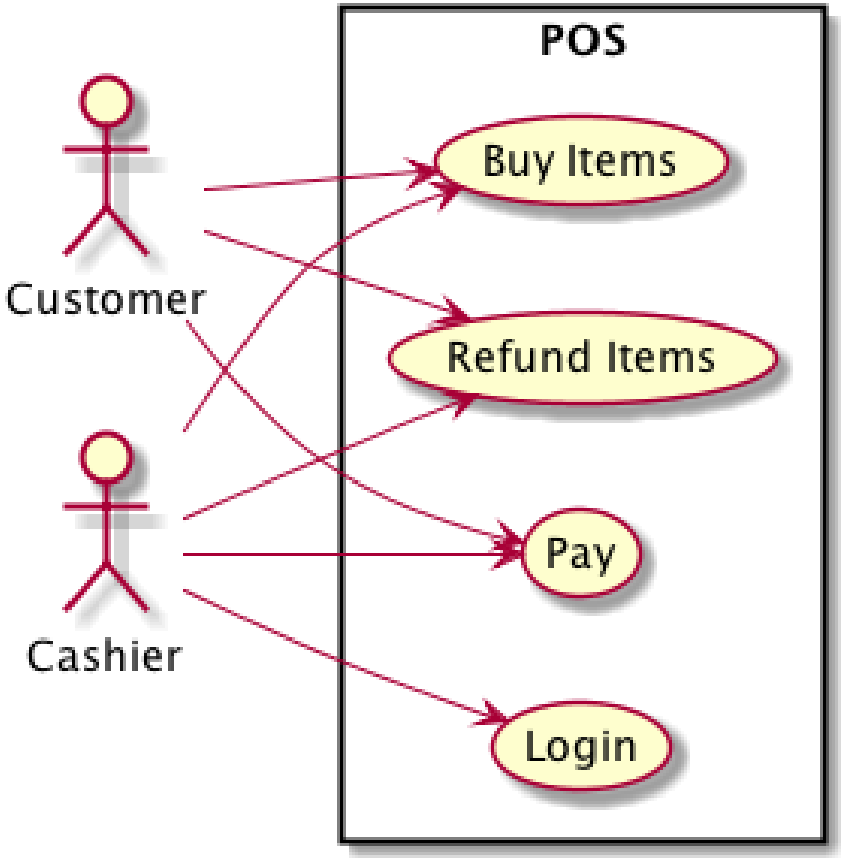
Example: Use Case

Use Case: Buy Items
Actors: Customer, Cashier
Description: A customer arrives at a checkout with items to purchase.
The cashier records the purchase item
The system presents the running total and line-item details.
The cashier collects the money and enters the payment information.
The system updates inventory.
The customer receives the receipt and leaves with the items

Actors

- Actors are
  - external to the system
  - participates in the story of a use case
- System Boundary
  - Hardware
  - Software
  - Organisation

Use Case Diagrams



Expanded Use Cases

Use Case: <unique name of the use case>
Primary Actor: <Actor initiating the use case>
Stakeholders: <List of actors and their interests>
Purpose: <Intention of the use case>

Precondition: <What must be true before the use case can start>
Postcondition: <Guaranteed Results>
Overview: <High-level use case or other summary>

Basic Flow: <Main successful scenario>
Alternative Flows: <branches (success or failure) of the main scenario>

Special Requirements:
Technology:
Open Issues:

Expanded Use Case Basic Flow

Main Successful Scenario

Actor Action	System Response
1. The cashier records the purchase items	
	2. The system presents the running total and line-item details
3. The cashier collects the money and enters the payment information	
	4. The System updates the inventory
5. The customer receives the receipt and leaves with the items.	

Alternative Flows

Line n: . . .

Line k: . . .

Example of Expanded Use Case

- Use Case: Buy Items with Cash
- Primary Actor: Cashier
- Stakeholders: Customer, Company, Gvt., Tax agency
- Purpose: Capture a sale and its cash payment
- Overview:

A customer arrives at a checkout with items to purchase. The cashier records the purchase items and collects payment. On completion, the customer leaves with the items.

- Precondition: cashier is identified
- Postcondition: sale is safe, receipt is generated, payment is recorded
- Basic Flow:

Actor Action	System Response
1. Customer arrives at a checkout with items to purchase.	
2. Cashier records identifier from each item.	
	3. The system determines the item price and adds item info into the sales transaction. Description and price of the current item are presented.

(Continues with more of the same)

- Alternate Flows:

Line	Flow
2	Invalid identifier is entered The System indicates an error.
7	Customer does not have enough cash The Cashier cancels the transaction

- Special Requirements:
  - Touch Screen UI
  - Language Internationalisation
- Technology:
  - Item identifier entered by barcode laser scanner
- Open Issues:
  - Can the customer pay by card?

Ranking Use Cases

Question

Which use case is the most important to begin with?

Rule

First implement use cases that *significantly influence* the core system architecture.
(Compare with Agile's *Minimum Viable Product (MVP)*)

Ranking

Increase ranking of a use case if it

- has direct impact on architectural design
  - example: adds classes to domain layer, require persistent services
- includes risky, time-critical, complex functions
- involves new research or technology
- represents primary business processes
- directly supports revenue or decreased costs

Ranking Techniques

- Scored (Numerical Weights)
- Discrete (High, Medium, Low)
- Simple Ordering (bubble sort?)
- MoSCoW (Must have, Should have, Could have, Won't have)
- Cumulative Voting