

# INFO-H-302

## TPI: Use Cases

Analyse et Conception par Objets

# Specifications

- Description semi-formelle du système en cours de développement
- But : décrire le logiciel d'une façon compréhensible pour toutes les personnes impliquées
- Outils : Modèle conceptuel, Glossaire, Cas d'utilisations, Spécifications de protocoles

# Cas d'utilisations / Use Cases

- Scénarios décrivant les possibilités les plus courantes d'utilisation du logiciel
- Assez informels
- Compréhensibles par tous
- Une des premières étapes de spécification

# Exemple

Use case: Buy Items

Actors: Customer (initiator), Cashier

Description: A Customer arrives at a checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion the Customer leaves with the items.

# I) Liste des Use Cases

- Acteurs : Customer, Cashier, SysAdmin
- Use Cases :
  - Buy items
  - Exchange items
  - Start System
  - Stop System
  - Get Sales logs
  - Cashier Login
  - Cashier Logout
  - Manage Cashiers

# Niveaux de Use Cases

- Plus ou moins formels
- Plus ou moins détaillés
- On identifie deux niveaux :
  - High-Level Use Cases
  - Detailed Use Cases

## 2) High-Level Use Cases

- Décrire chaque Use Case de la liste en 2-3 phrases
- Obtenir une description haut-niveau mais complète du système
- But : Identifier les UCs à détailler

# Exemple

Use case: Buy Items

Actors: Customer (initiator), Cashier

Description: A Customer arrives at a checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion the Customer leaves with the items.



# 3) Detailed Use Case

- Description détaillée du cas d'utilisation
- Respecte un canevas standardisé
- Début du travail de conception

# Canevas

- Acteurs et buts
- Pre- et post-conditions
- Résumé
- Déclencheur
- Scénario de base
- Alternatives
- Erreurs

Name	Buy Items with Cash
Actors	Customer, Cashier
Goal	Cashier determines total purchase price. Customer pays quickly.
Pre-conditions	Cashier is logged-in
Post-conditions	Sale is properly logged, inventory is updated
Trigger	Customer arrives at checkout with goods

### Typical Course of Event

#### Actor Action

1. This use case begins when a Customer arrives at a POST checkout with items to purchase
2. The Cashier records the identifier from each item. If there is more than one of the same item, the Cashier can enter the quantity as well.
4. On completion of item entry, the Cashier indicates to the POST that the item entry is complete.
6. The Cashier tells the Customer the total.
7. The Customer gives a cash payment, possibly greater than the sale total.
8. The Cashier records the cash received amount.
10. The Cashier deposits the cash received and extracts the balance owing. The Cashier gives the balance owing and the printed receipt to the Customer.
12. The Customer leaves with the items purchased.

#### System Response

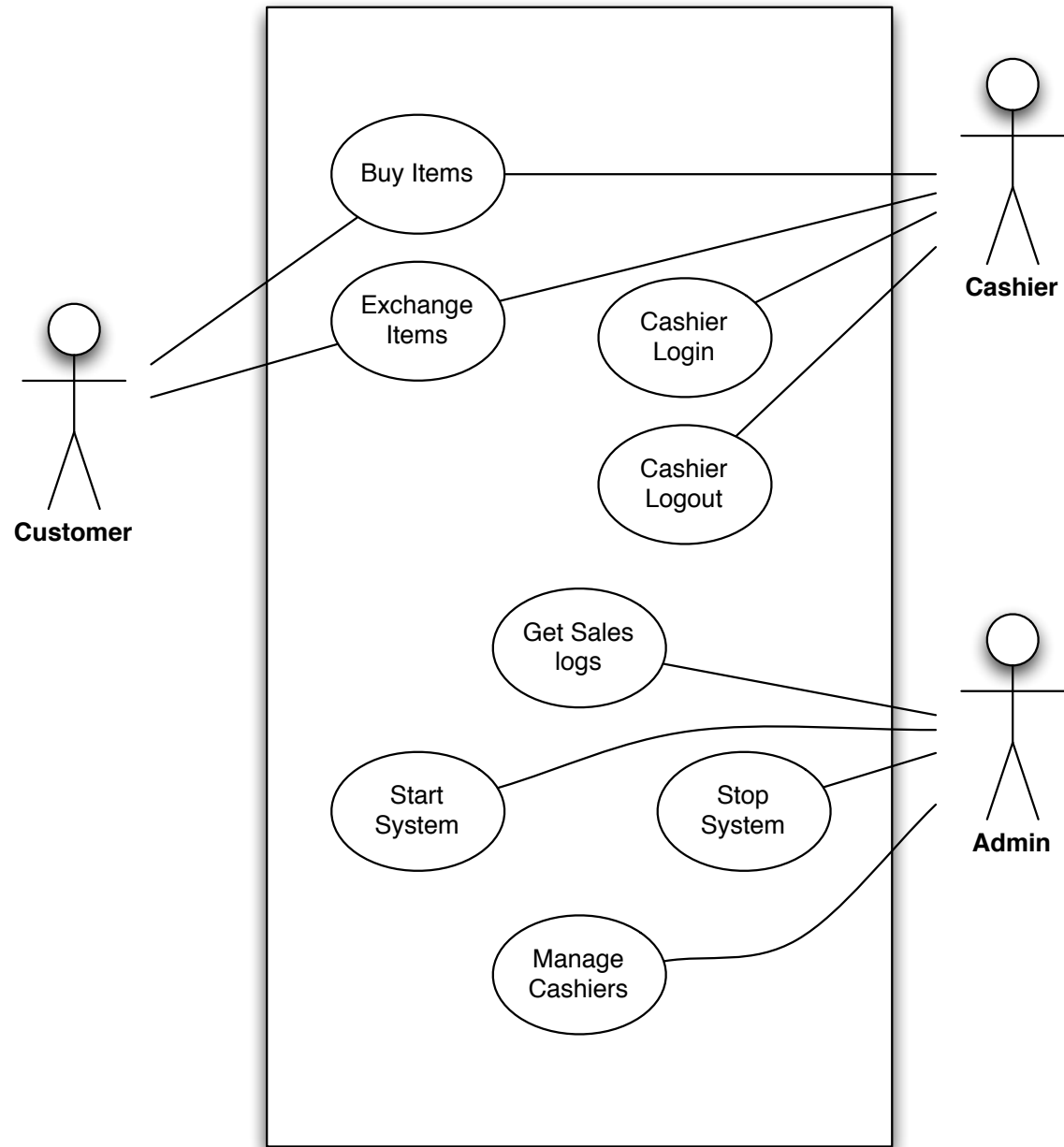
3. Determines the item price and adds the item information to the running sales transaction. The description and price of the current item are presented.
5. Calculates and presents the sale total.
9. Shows the balance due back to the Customer. Generates a receipt.
11. Logs the completed sale.

#### Alternative Courses

- Line 2: Invalid identifier entered. Indicate error.
- Line 7: Customer didn't have enough cash. Cancel sales transaction.

# 4) Use Case Diagram

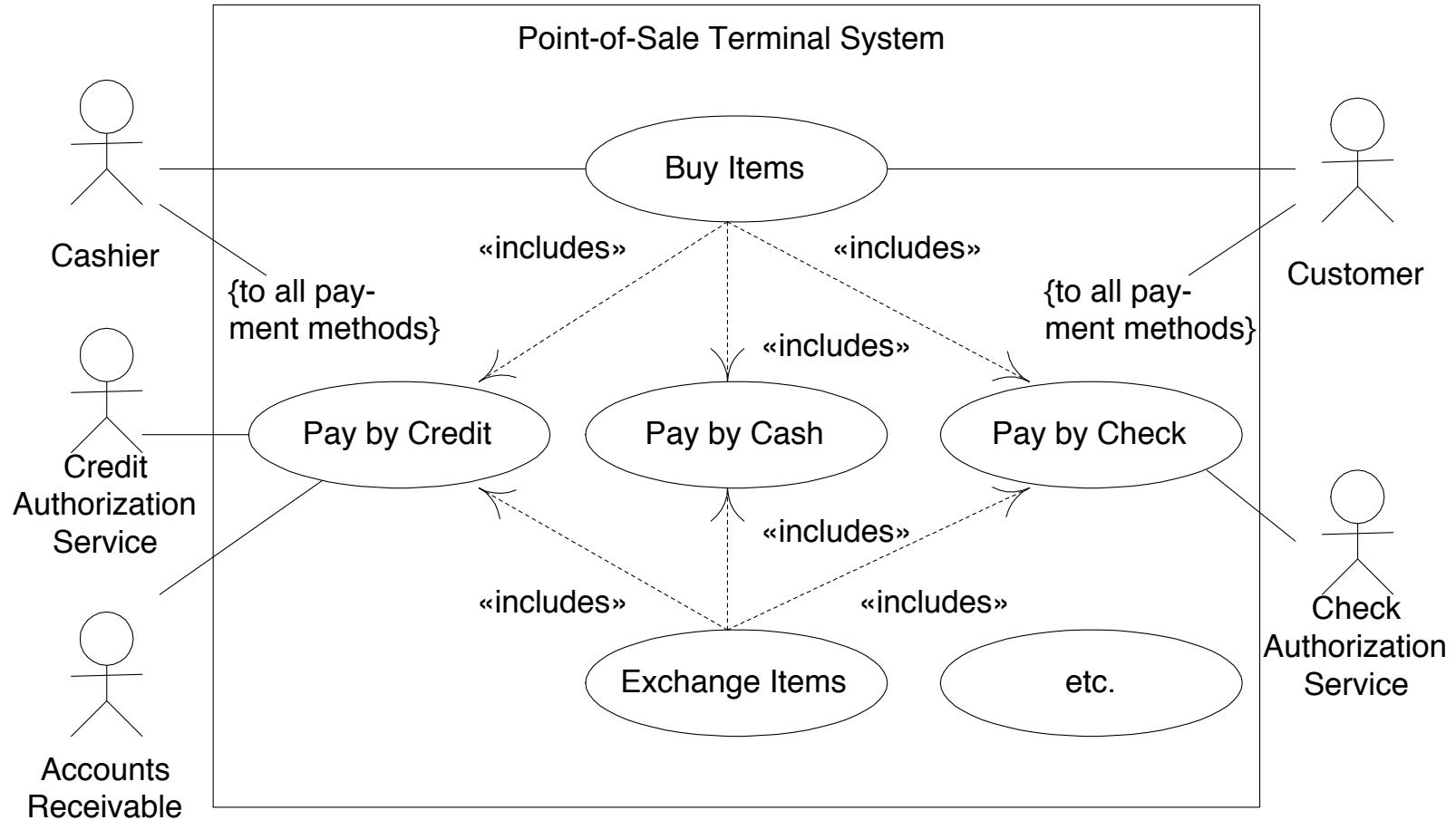
- Représentation visuelle sommaire des liens entre Use Cases et acteurs
- Vue globale d'un ensemble de Use Cases



# Use Case diagrams avancés

- Trois relations particulières
  - Spécialisation : cas particuliers
  - Inclusion : englobe un autre use case
  - Extension : prolonge un use case

# Point-of-Sale Application: Relating Use Cases





# Lister Use Cases

- Penser par Acteur
- Penser par Événement
- Ne pas s'arrêter aux étapes mais penser scénarios complets
  - *Printing the receipt vs Buy Items*

# Pour finir...

- Qu'est-ce qui peut se passer ?
- Comment le système doit réagir ?
- Le client choisit de payer par carte, puis la transaction est refusée...
- Décider *maintenant* comment le système réagira