

## TD1: Cas d'utilisation CORRECTION

On s'intéresse ici aux cas d'utilisation textuels de l'approche d'A. Cockburn.

### Exercice 1

Un collègue vous envoie le cas d'utilisation suivant. A l'aide des conseils de rédaction vus en cours, renvoyez-lui un avis critique et des corrections :

#### **Cas d'utilisation 25 : ouvrir une session**

*Ce cas d'utilisation décrit le processus par lequel les utilisateurs se connectent au système de traitement des commandes. Il vise aussi à établir les autorisations d'accès aux différentes catégories d'utilisateurs (clients, employés, ...).*

#### **Scénario :**

1. *Le cas d'utilisation débute lorsque l'utilisateur démarre l'application*
2. *Le système affiche l'écran d'ouverture de session*
3. *L'utilisateur saisit un nom d'utilisateur et un mot de passe*
4. *Le système vérifie les informations*
5. *Le système définit les autorisations d'accès*
6. *Le système affiche l'écran principal*
7. *L'utilisateur sélectionne une fonction*
8. *Tant que l'utilisateur ne sélectionne pas la sortie, boucler :*
9. *Si l'utilisateur sélectionne Passer une commande, utiliser Passer une commande*
10. *Si l'utilisateur sélectionne Retourner le produit, utiliser Retourner le produit*
11. *Si l'utilisateur sélectionne Annuler la commande, utiliser Annuler la commande*
12. *Si l'utilisateur sélectionne Obtenir l'état de la commande, utiliser Obtenir l'état de la commande*
13. *Si l'utilisateur sélectionne Envoyer le catalogue, utiliser Envoyer le catalogue*
14. *Si l'utilisateur sélectionne Exécuter le rapport des ventes, utiliser Exécuter le rapport des ventes*  
*Fin Si*
15. *L'utilisateur sélectionne une fonction.*  
*Fin de boucle*
16. *Le cas d'utilisation prend fin*

### **CORRECTION**

Le CU ne concerne pas l'ouverture de sessions, malgré ce que laisse entendre son nom et son résumé. Il concerne le système de traitement des commandes. Le véritable CU auquel nous avons affaire ici est un CU de portée système, en boîte noire. Si les 6 premières étapes concernent bel et bien l'ouverture de sessions, il s'agit là d'un niveau d'objectif inférieur, qui devrait être mis à part. Une fois la chose faite, on remarque que l'utilisateur ouvre une session pour rentrer dans le système, mais qu'il n'en sort jamais !

*Boucler tant que l'utilisateur ne sélectionne pas la sortie, Fin Si et Fin de boucle* sont des constructions de programmeurs qui n'ont aucun sens pour les lecteurs potentiels du CU. La présence des énoncés en *si* alourdit l'écriture. Les étapes décrivent la conception de l'interface utilisateur, qui n'a pas à apparaître ici.

Le cas d'utilisation débute lorsque... et Le cas d'utilisation prend fin sont des étapes strictement ornementales, donc inutiles. La plupart des lecteurs supposeront naturellement que le CU débute avec l'étape 1 et s'achève à la fin de l'écrit.

L'autre style à prendre en considération est la formule suivante : L'utilisateur ..., utiliser Passer une commande. Dans cette expression, « utiliser » renvoie à la relation UML d'inclusion (auparavant appelée relation d'*utilisation*). Cette formulation obscurcit plus qu'elle n'éclaire le récit. On préférera employer la convention vue en cours pour renvoyer à un autre CU (nom du CU en souligné).

Si on souhaitait être complètement rigoureux, on écrirait les champs Acteur principal, Garanties, Niveau d'objectif, Portée ainsi que les extensions...

On peut identifier deux CU à placer à part : le CU de niveau utilisateur *Utiliser le système de traitement des commandes* et le CU de niveau sous-fonction *Ouvrir une session*.

### **Cas d'utilisation 25** **Utiliser le système de traitement des commandes**

...

#### **Scénario nominal :**

1. L'utilisateur ouvre une session
2. Le système présente les fonctions disponibles. L'utilisateur en sélectionne une parmi les suivantes et l'effectue :
  - Passer une commande
  - Retourner le produit
  - Annuler une commande
  - Obtenir un état de la commande
  - Envoyer un catalogue
  - Exécuter le rapport des ventes
3. Cette opération se répète jusqu'à ce que l'utilisateur choisisse de quitter
4. Le système ferme la session de l'utilisateur lorsque celui-ci choisit de quitter

## **Exercice 2**



Vous venez de perdre vos clés dans le bus et avez l'idée de développer un service d'objet trouvé sur Internet : *fffound*. Le service fournit des portes clés et des autocollants avec un identifiant et un QRcode unique à ses utilisateurs. Les utilisateurs peuvent coller les autocollants sur leurs objets (par exemple un téléphone) ou attacher leurs clés au porte clés.

Ils doivent ensuite lier ces identifiants à leurs comptes, pour que des personnes qui trouveraient les objets perdus puissent rentrer en contact avec eux, en rentrant l'identifiant de l'objet.

A partir de cette description, nous pouvons produire des cas d'utilisation ayant des portées, des visibilitées et des niveaux d'objectifs différents. Écrivez trois cas d'utilisation :

1. Un CU métier de niveau stratégique dont la portée est votre entreprise et l'acteur principal le client, en boîte noire.
2. Un CU de niveau utilisateur, dont l'acteur principal est la personne trouvant un trousseau perdu.
3. Un CU de niveau utilisateur, dont la portée est le service d'enregistrement d'objets.



## CORRECTION

**CU1**  **le client utilise fffound**   
**Portée** : fffound  
**Niveau** : stratégique  
**Acteur principal** : le client  
**Déclencheur** : le client découvre le service  
**Garanties minimales** : le client a un compte.  
**Garanties de succès** : le client a enregistré des objets sur le site.  
**Scénario nominal** :  

1. L'utilisateur s'inscrit sur le site
2. L'utilisateur commande des autocollants.
3. fffound envoie des autocollants.
4. L'utilisateur attache les autocollants à ses objets.
5. L'utilisateur enregistre les identifiants sur le site.

### Extensions

- 2a. Les autocollants sont trop chers ou ne correspondent pas aux attentes/besoins de l'utilisateur. Il attend une nouvelle version.
- 3a. fffound associe automatiquement les identifiants des autocollants au compte de l'utilisateur.
- 5a. L'utilisateur utilise son téléphone portable pour enregistrer les autocollants en scannant le QRcode.

**CU2**  **le client s'enregistre sur fffound**   
**Portée** : fffound  
**Niveau** : utilisateur  
**Acteur principal** : une personne découvrant un objet perdu  
**Acteur secondaire** : le propriétaire  
**Déclencheur** : une personne découvre un objet perdu enregistré sur fffound  
**Garanties minimales** : aucune  
**Garanties de succès** : le client retrouve son objet  
**Scénario nominal** :  

1. Le découvreur trouve l'objet
2. Le découvreur se rend compte que l'objet est associé à fffound
3. Le découvreur comprend que fffound permet de rentrer en contact avec le propriétaire de l'objet.

4. Le découvreur se rend sur le site de fffound.
5. L'utilisateur entre l'identifiant de l'objet sur le site.
6. fffound affiche le message du propriétaire au découvreur
7. Une page de discussion (anonyme ?) est établie entre le propriétaire et la personne ayant trouvé l'objet.
8. fffound enregistre pour référence l'IP du découvreur, sa localisation. <- boîte blanche
9. Un médiateur de fffound est prévenu pour vérifier éventuellement que la discussion se passe dans de bonnes conditions.
10. fffound envoie une alerte au propriétaire
11. Le propriétaire peut décider d'une récompense qui passera par fffound une fois l'objet remis.
12. Quand les deux personnes se rencontrent, elles valident l'échange.

#### **Extensions**

- 1a. Le découvreur ignore l'objet
- 3a. Le découvreur ignore le service fffound
- 4a. Le découvreur utilise son téléphone portable pour scanner le Qrcode. Il peut sauter l'étape 5.
- 11a. Le découvreur doit se créer un compte avec données bancaires.
- 11b. Il n'y a pas de récompense, fffound envoie gratuitement des autocollants à la personne qui a rendu l'objet.

#### **CU3**



#### **le client enregistre ses objets sur fffound**



**Portée :** fffound

**Niveau :** utilisateur

**Acteur principal :** le client

**Déclencheur :** le client découvre le service

**Garanties minimales :** le client a un compte.

**Garanties de succès :** le client a enregistré des objets sur le site.

**Scénario nominal :**

1. L'utilisateur cherche ses objets
2. L'utilisateur s'identifie sur le site
3. L'utilisateur enregistre l'identifiant de chacun de ses objets sur le site.
4. L'utilisateur ajoute une récompense à ses objets favori
5. L'utilisateur ajoute un moyen d'être contacté

#### **Extensions**

- 1a. A vous ...

### **Exercice 3 : coding style**

Le morceau de code suivant contient plusieurs violations de style (style Java Oracle). Lesquelles ?

```
package fr.univ_lyon1.info.m1.poneymon_fx;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.stage.Stage;
/*
 * Classe principale de l'application.
 *
 * S'appuie sur javafx pour le rendu.
```

```
*/  
public class App extends Application {  
  
    /**  
     * lance l'application.  
     */  
    @Override  
    public void Start(Stage STAGE) throws Exception  
    {  
        // Nom de la fenetre  
        STAGE.setTitle("Poneymon");  
  
        Group root = new Group();  
        Scene scene = new Scene(root);  
  
        // On cree le terrain de jeu et on l'ajoute a la racine  
        Field gameField = new Field(scene,600,600);  
        root.getChildren().add(gameField);  
  
        // On ajoute la scene a la fenetre et on affiche  
        STAGE.setScene(scene);  
        STAGE.show() ;  
    }  
}
```

**Correction :**

- Lignes blanches pour séparer package, import et définition de classe.
- Commentaire /\* ... \*/ pour une documentation d'API, devrait être /\*\* ... \*/
- Espace en trop entre public et class
- Accolade « { » seule sur une ligne, devrait être sur la ligne précédente.
- Indentation du contenu de la classe à 2 au lieu de 4.
- STAGE devrait s'écrire Stage
- Manque des espaces derrière , dans l'appel à new Field.
- Espace en trop entre show() et ;.