TD2- Conception Avancée en UML

I. Check-List avant de passer au TD 2

Avant de passer au problème du TD 2, vérifiez que votre solution au TD 1 répond bien à tous les critères suivants :

1. Erreurs fréquentes et "graves"

a. Le "SI" n'est ni un acteur, ni un cas d'utilisation (il est représenté par l'ensemble des cas d'utilisation).
b. La BD (base de données) n'est pas un acteur. La BD peut être un moyen d'implémentation du stockage des informations des objets.
c. Pour chaque cas d'utilisation, les acteurs sont bien les personnes ou systèmes informatiques qui interagissent directement avec le Système informatique, *i.e.*, la personne qui "tape" sur le clavier, pas celle qui regarde par-dessus l'épaule !
d. Si un cas d'utilisation ne fait pas intervenir le système informatique, ce n'est pas un cas d'utilisation (c'est une activité du métier qui n'est pas informatisée).
e. Les relations entre les cas d'utilisation ne sont jamais des relations d'ordre. Il n'y a qu'héritage, inclusion ou extension.
f. Attention vos cas d'utilisation ne doivent pas se recouvrir : si vous pensez qu'un UC est inclus dans un autre ou le spécialise vous devez l'expliciter.
g. Attention, un acteur n'est jamais un objet du système informatique (il n'est pas représenté par une classe). Son compte pour un humain, un proxy pour un acteur système est peut-être un objet du système, mais pas l'acteur lui-même.
h. Tout ce qui est représenté doit respecter la sémantique du langage, par exemples
    i. Sens et représentation des relations entre UC ou classes (héritage et association n'ont pas la forme !)
    ii. Placement des rôles et multiplicité dans les associations (même ChatGPT semble donner une réponse juste, vous devriez pouvoir faire juste).
    iii. Les messages dans le diagramme de séquence ne partent pas "tous seuls".

2. Bonnes Pratiques

a. Le glossaire doit essentiellement définir les termes du métier, donc ici ceux d'un système de commandes en ligne. Il permet de fixer le vocabulaire et d'éviter les ambiguïtés. S'il y a des termes différents pour parler d'un même concept dans la spécification, dans la modélisation, un seul terme fixé dans le glossaire est utilisé PARTOUT. On ne doit donc pas trouver dans les classes, les diagrammes de séquence ou les UC, un même concept sous des noms différents.
b. Vous avez une classe principale Facade (STEats ou un nom dans ce genre) qui sert de point d'entrée dans votre modèle objet. Ceci n'est pas une obligation en UML, c'est juste une facilité pour faire des diagrammes de séquence simples et cohérents (on reviendra sur cette notion de "Facade" plus tard). Attention, elle ne doit pas concentrer toute la logique, il faut distribuer intelligemment les responsabilités des traitements sur les autres objets (rappelez-vous les patrons GRASP vus l'année dernière sur lesquels on reviendra rapidement en cours).
c. Si, dans une classe, vous conservez un ID qui vous permet de retrouver un autre objet, c'est qu'il y a une association entre votre classe et la classe de cet autre objet (idem si vous conservez un tableau d'ID 😱). Attention vous anticipez peut-être beaucoup trop tôt sur votre représentation informatique du problème que vous êtes en train de modéliser. **Les attributs dans les classes UML ne devraient contenir que des types de base** (integer, String, Date). Toutes les autres informations sont

représentées par des associations. *Ne pas respecter cette bonne pratique aurait pu être référencé comme une erreur.*

3. Périmètre et cohérence

a. Une fois que vous avez des premiers cas d'utilisation couvrant le sujet, vérifiez que l'ensemble des cas soit cohérent et si possible complet. Il y a peut-être des cas non exprimés dans le sujet (comment le restaurant signale-t-il que la commande est prête ? Comment le livreur est-il notifié qu'il doit effectuer une livraison ?)
b. Le niveau de diagramme de cas d'utilisation doit permettre d'identifier les grandes fonctionnalités, pas les détails des scénarii, et inversement être suffisamment détaillé pour bien identifier les acteurs et les fonctionnalités.
c. Un diagramme de séquences est une vision du déroulement d'un des scenario d'un cas d'utilisation. On doit naturellement y retrouver les acteurs liés au cas d'utilisation comme déclencheur ou récepteur de messages. Assurez que vos diagrammes de séquence ne traversent pas plusieurs cas d'utilisation.
d. Dans un diagramme de séquences,
    i. les envois de message ne sont possibles qu'entre des instances de classes liées entre elles (par association, agrégation), ou des objets obtenus par retour de méthodes.
    ii. À l'inverse, les méthodes utilisées dans les diagrammes de séquences doivent faire apparaitre dans le diagramme de classes (dans les classes des objets récepteurs !).
e. Dans un diagramme d'activité,
    i. Seuls des acteurs définis par les cas d'utilisation peuvent intervenir ;
    ii. Les objets (instances de classes) manipulés dans le diagramme d'activité doivent être cohérents avec les classes définies dans les diagrammes de classes.

II. Travail à réaliser

1. Vérifiez vos modèles

Commencer par vérifier que vous n'avez commis aucune des erreurs présentes dans la check-list.

2. Complétez vos diagrammes de cas d'utilisation

Déterminer les ***relations entre les cas d'utilisation*** pour en particulier bien capturer les éléments communs (*cas d'héritage*), les facilités de navigation entre cas d'utilisation (*extends*) et les exigences (*include*). N'en ajoutez pas artificiellement.

3. Complétez votre diagramme de séquence

Focalisez-vous sur le diagramme de séquence qui réalise l'opération : ***passer une commande.***
Le niveau de détail de ce diagramme doit permettre de visualiser au moins :

a. Le choix des créneaux de livraison en *« fonction des restaurants »*, en attribuant la responsabilité de rechercher les créneaux à un objet dédié (donc pas "System" ou STEats)
b. Le choix des menus en fonction du créneau de livraison sélectionné
c. La création et l'enregistrement de la commande (pas System ou STEats)
d. L'appel au service externe de paiement

Ne détaillez pas les cas d'erreur, mais posez des notes pour les identifier.
N'intégrez pas la connexion au Système qui est une précondition à ce cas d'utilisation.
Ne traversez pas d'autres cas d'utilisation que celui modélisé, mais n'oubliez pas de notifier les éléments de votre système qui ont besoin de savoir qu'une nouvelle commande a été passée.

4. Mettez à jour le glossaire si nécessaire

5. Complétez votre diagramme de classe au niveau du domaine.

6. Réalisez un diagramme d'activité

Pour simplifier, nous nous limitons au diagramme suivant qui modélise uniquement l'enchainement des cas d'utilisation de **la prise de commande à la livraison du point de vue unique du client.** Vous prendrez soin de préciser les objets et leur état. Vous exprimez donc uniquement l'ordre des activités qui le concernent.
Vous n'avez pas besoin de beaucoup de notations. L'objectif est de vous aider ici à formaliser les objets qui doivent être transmis.
Un diagramme d'activité qui visualise les interactions entre tous les acteurs de la prise de commande à la livraison est gros. Si vous le souhaitez, vous pouvez introduire cette vision plus globale, mais ne vous perdez pas.

7. Identifier les limites additionnelles

S'il y en a que vous ne traiterez pas (**G.6**) ou les **hypothèses** que vous avez fait sur le projet (E.4)

8. Préciser les responsabilités de chacun dans l'équipe (P.1)

III. RENDU D1

Vos réponses aux points 2 à 8 sont à rendre sur Moodle : **D1 et E1 : Rendu de la modélisation UML initiale et Évaluation par les pairs.**

Nous mettons ici ce qui est attendu dans ce rendu.

La structure du rapport en pdf doit suivre le plan suivant :

1. Le nom des étudiants impliqués dans l'équipe sur la première page avec leur rôle dans l'équipe
2. Une section d’une page maximum résumant
    1. vos hypothèses de travail (E.4)
    2. les limites identifiées (G.6)
    3. et si certains points restent non étudiés.
3. Une section pour chacun des éléments suivants, avec éventuellement des explications supplémentaires propres aux diagrammes, si cela est nécessaire. Si vous le souhaitez, vous pouvez ajouter des diagrammes.
    1. Le glossaire
    2. Le diagramme de cas d'utilisation
    3. Le diagramme de classes
    4. Le diagramme de séquences
    5. Un diagramme d'activité.

Le périmètre du rapport à rendre correspond à l’ensemble des fonctionnalités demandées dans les TD 1 et TD 2. N’oubliez pas de vérifier les différents éléments de la check-list fournie avec le sujet du TD 2.