S6 - Garcia Inès, Donot Axelle, Bessonnaud Romain

CAHIER DES CHARGES

Conception Orienté-Objet

Uber Eats

Uber Eats

Sommaire :

1-Présentation du projet

Dans le cadre de notre module de Conception Orientée Objet, nous avons décidé de nous intéresser à l'entreprise de livraison de nourriture la plus utilisée de nos jours. En effet Uber Eats représente, environ 57% des livraisons de nourriture dans l'Hexagone. Étant des utilisateurs de ce service, nous avons pu en approfondissant nos connaissances existantes, produire différents diagrammes récapitulant le fonctionnement de cette entreprise.

2-Objectif du projet

L'objectif du projet de Conception Orientée Objet est de modéliser les différents diagrammes, correspondant au fonctionnement complet de l'application Uber Eats. Nous avons produit un cahier des charges (aussi appelé cahier de conception) pour détailler notre démarche et nos choix.

3-L'application

Uber Eats est un service de livraison de plats cuisinés. Depuis 2015, l'application lancée par Uber, livre des repas dans le monde entier (Amérique, Europe, Australie, Asie). Le service est simple, il faut commander un repas depuis l'application ou le site web, puis le restaurant s'occupe de préparer la commande. Quelques instants plus tard, un coursier récupère la commande, et vous l'apporte directement chez vous. Les coursiers sont indépendants, et peuvent se déplacer en vélo, scooter ou voiture.

4-Contenu du projet

Dans ce projet nous avons créé 5 types de diagrammes :

- Le diagramme de classes (Class Diagram), nous a permis de représenter les différentes classes et leurs relations.

- Le diagramme d'objet (Object Diagram), nous a permis d'instancier les différentes classes pour illustrer un cas concret.

- Les diagrammes de séquence (Séquence Diagram), nous ont permis de représenter les actions entre les différents acteurs, en fonction des méthodes à illustrer.

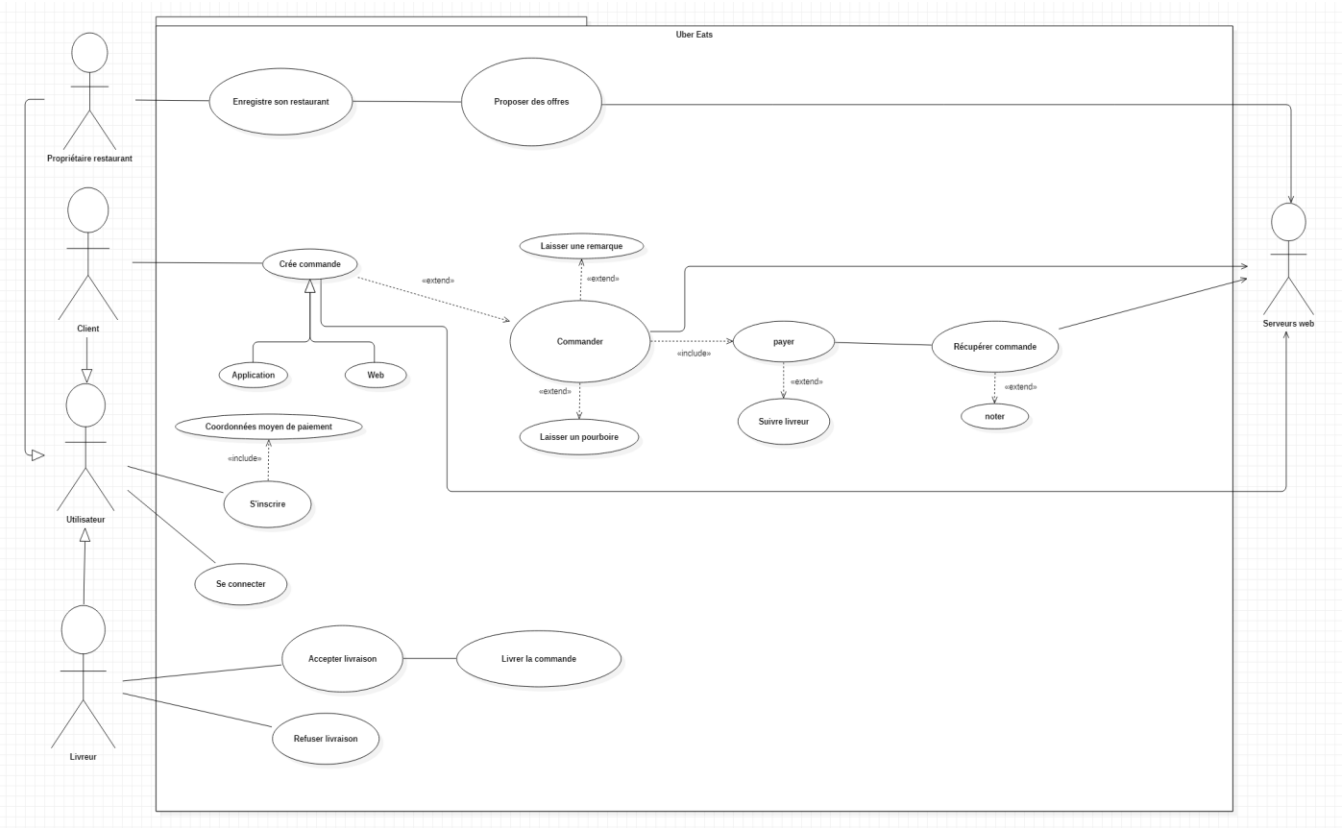
- Un diagramme de cas d'utilisation (Use Case Diagram), nous a permis de représenter les différentes utilisations possibles de l'application.

- Enfin, un diagramme d'états-transitions (State Transition Diagram), nous a permis de représenter les différents états que peut prendre l'objet.

Nous avons mis en place un lien MediaFire pour télécharger nos diagrammes en format SVG pour une meilleure qualité d'image :
https://www.mediafire.com/file/j052acz4cn39sns/SVG_COO.zip/file

5-Présentation des diagrammes

Diagramme de cas d'utilisation :



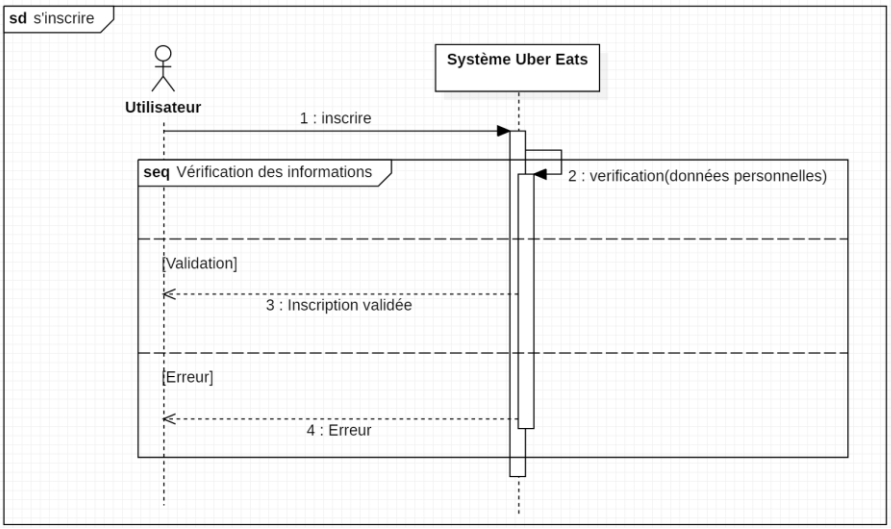
Pour commencer, nous avons représenté les 12 cas d'utilisation nécessaires au bon fonctionnement de l'application. Tout d'abord, l'utilisateur "Propriétaire de restaurant" enregistre ses restaurants, il ajoute les différentes offres qu'il souhaite proposer dans l'application.

Côté client, dès la première utilisation, les utilisateurs doivent obligatoirement s'inscrire, en entrant un moyen de paiement, un login et un mot de passe ("include"). Ensuite ils peuvent créer une commande, la valider, payer, choisir de laisser un pourboire et une remarque pour le livreur. Après le paiement, le client aura la possibilité de suivre le livreur en temps réel.

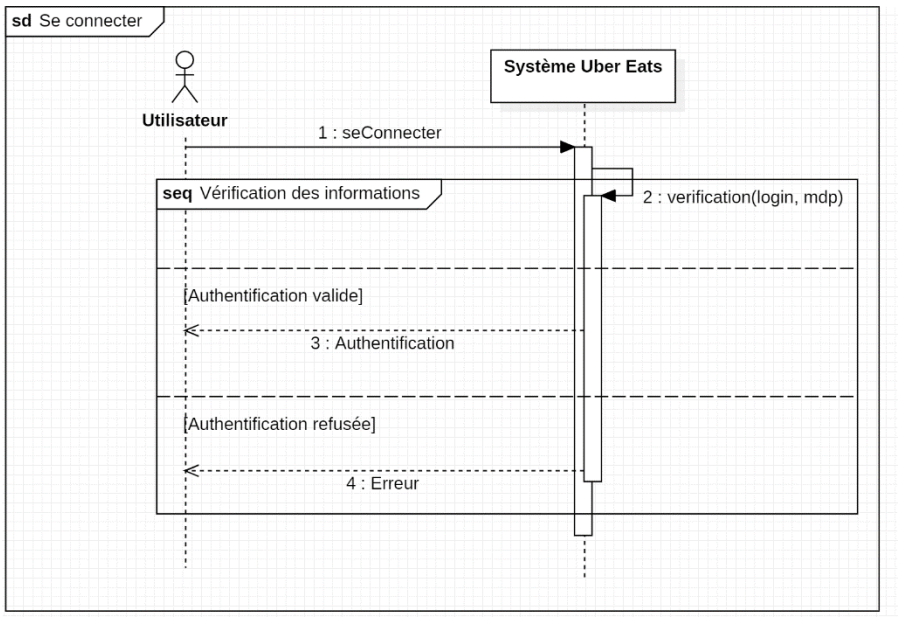
Chaque livreur a la possibilité d'accepter ou de refuser une commande en fonction de sa localisation par rapport à l'adresse de livraison et l'adresse du restaurant.

Nous avons fait le choix d'omettre des spécifications comme le moyen de locomotion, qui est normalement mentionné au moment de la prise en charge par le livreur.

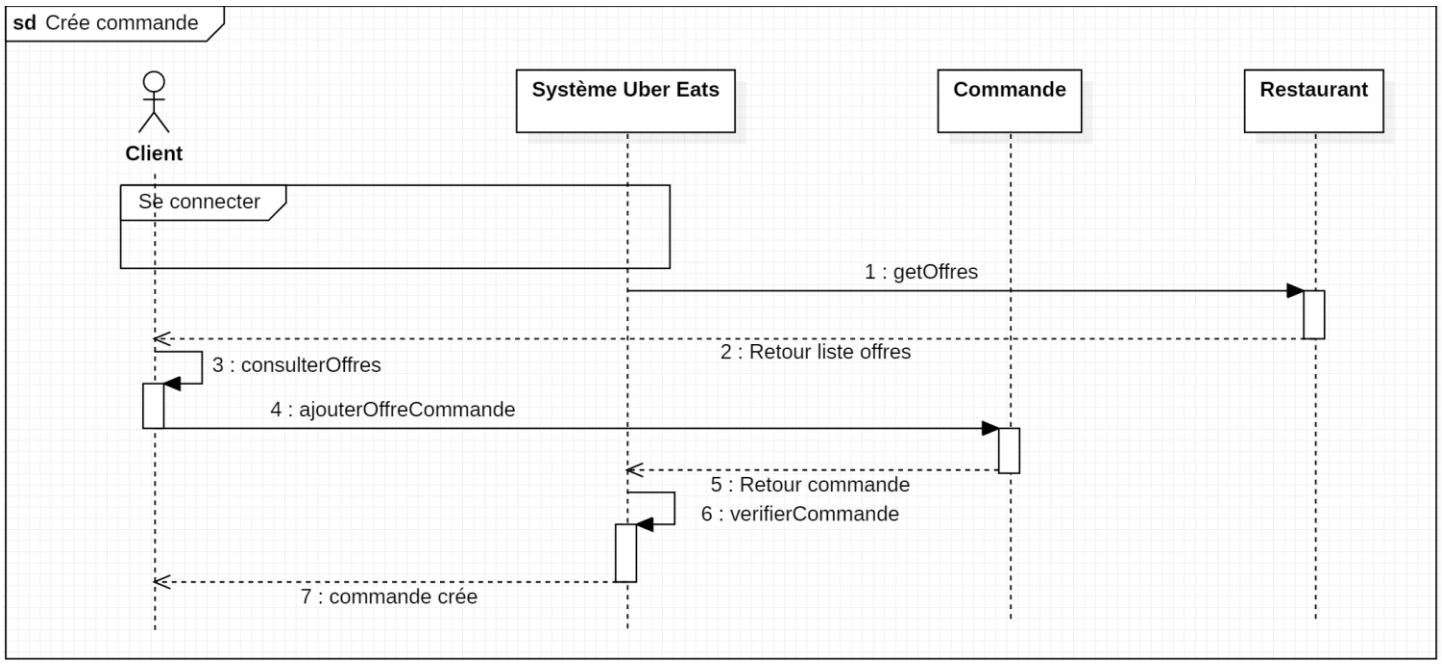
Diagrammes de séquence :



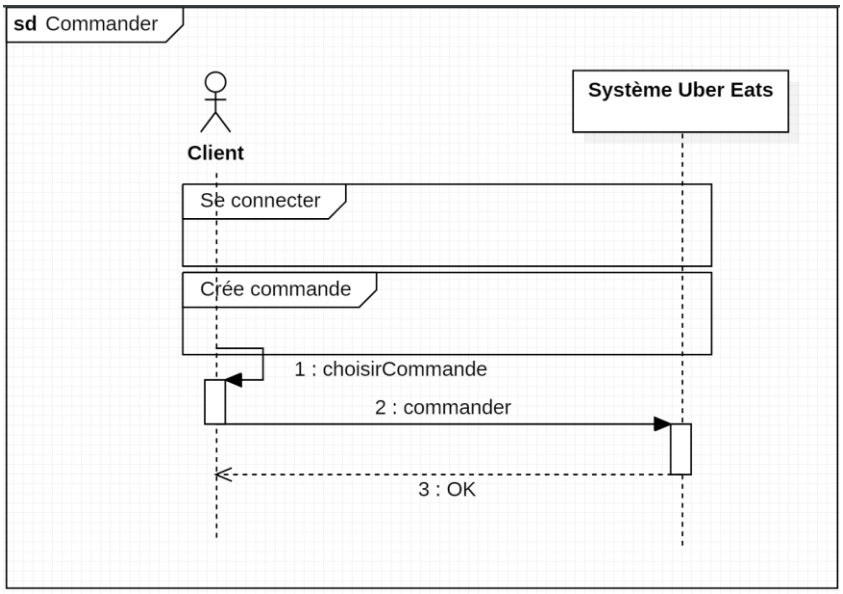
Pour s'inscrire l'utilisateur entre ses informations (adresse mail, adresse postale, numéro de téléphone, mot de passe, moyen de paiement...). Ces informations sont vérifiées par le système. Si ces informations sont valides, l'inscription est validée, sinon, le système indique un message d'erreur.



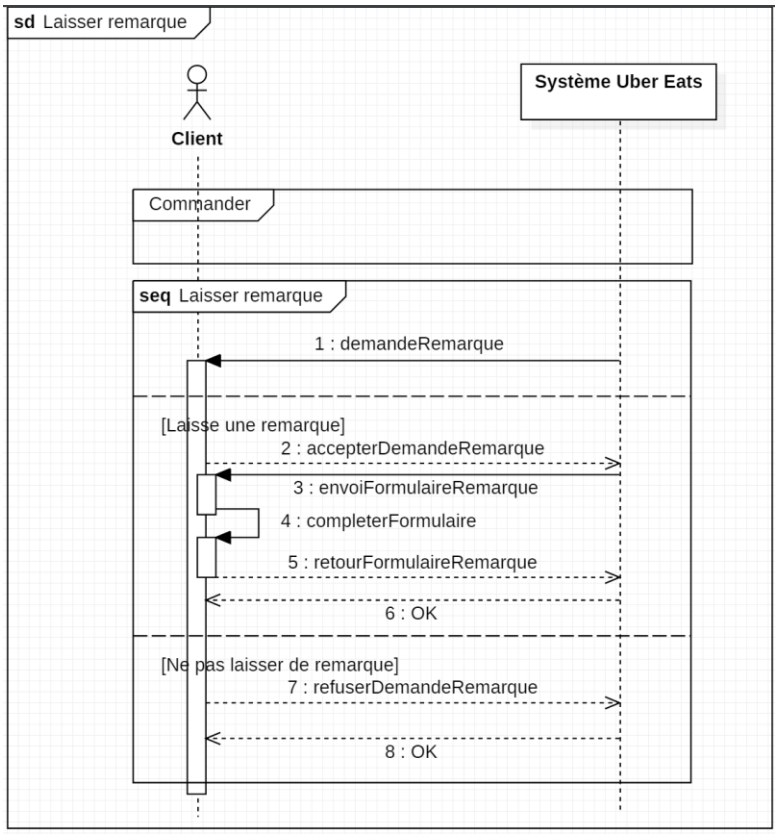
Lorsqu'un utilisateur souhaite se connecter, il entre son login et son mot de passe. Si le système les valide, alors le client est connecté, si le login ou le mot de passe sont erronés alors l'authentification est refusée et une erreur est renvoyée.



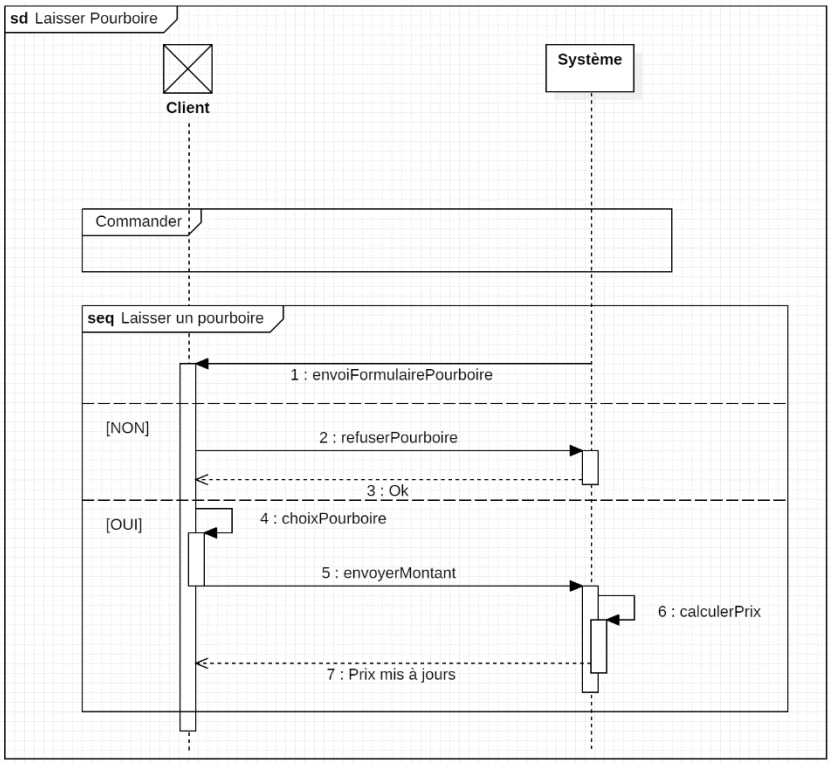
Un client connecté peut consulter les offres préalablement récupérées par le système. Si une offre lui plaît, il peut alors l'ajouter à sa commande. Si la commande est valide (quantité disponible, éligible à la livraison) alors le système crée la commande et la retourne au client.



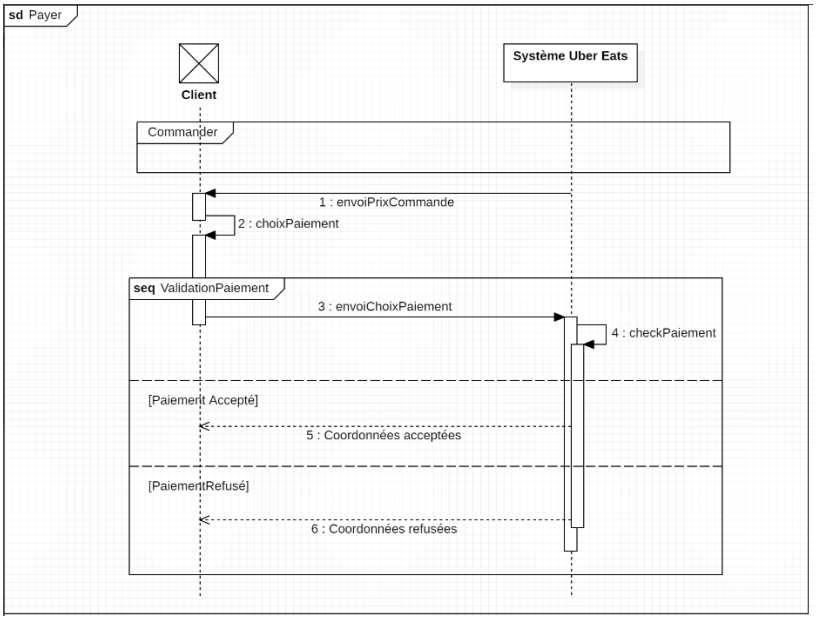
Pour pouvoir commander le client doit déjà être connecté et avoir créé sa commande, il va ensuite pouvoir commander. Si la commande est validée alors il reçoit un message du système qui lui annonce que sa commande est validée et qu'il doit passer au paiement.



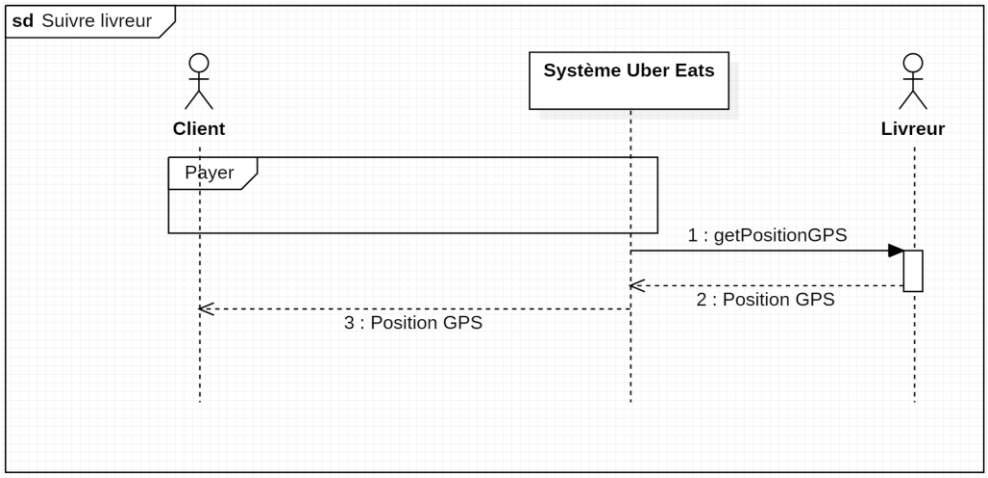
Après que le client ait commandé, il peut laisser une remarque : le système envoie alors une demande de remarque. Le client a deux choix : Soit il veut laisser une remarque alors le système lui envoie un formulaire à compléter, puis une fois complété il est récupéré par le système. Sinon le client refuse de laisser une remarque, alors il ne se passe rien.



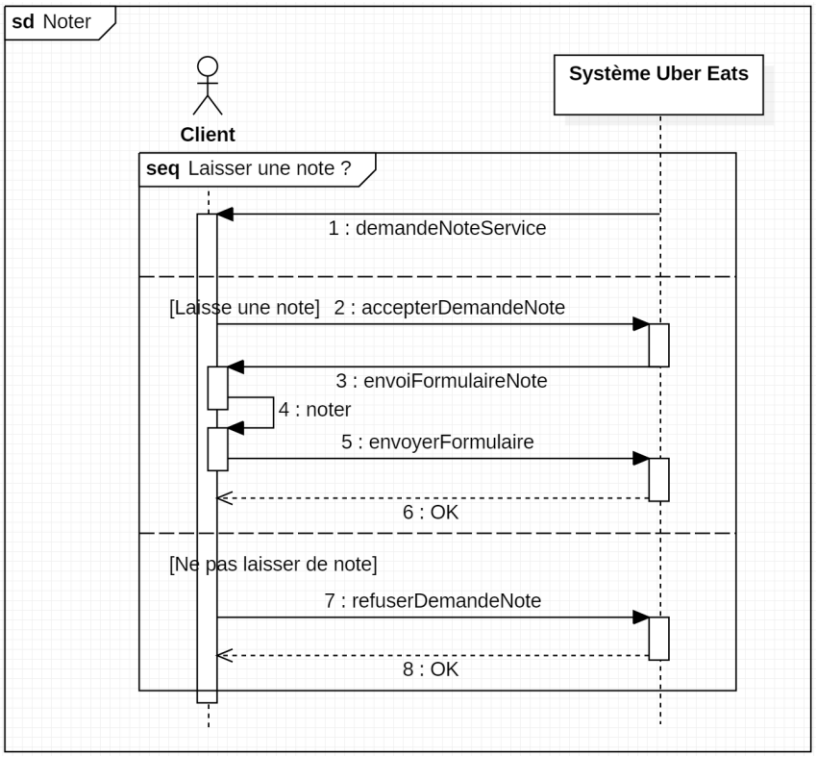
Une fois avoir commandé sur l'application UberEats, le système envoie le formulaire pour laisser un pourboire. Le client a deux choix : soit il accepte de laisser un pourboire, et le système récupère le montant laissé, et met à jour le prix en fonction du montant donné. Sinon il refuse et le système ne fait rien.



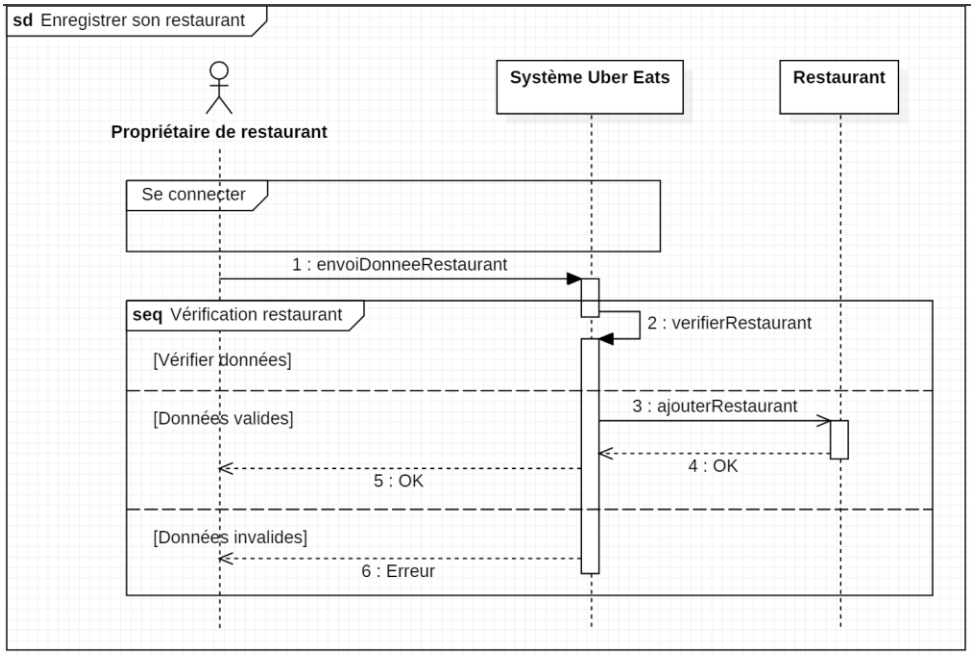
Payer implique déjà que le client ait passé une commande, le système envoie au client le prix de la commande. Le client choisit son moyen de paiement et procède à la validation. Si les informations bancaires sont valides (bon numéro et compte solvable) alors le paiement est accepté, sinon le paiement échoue.



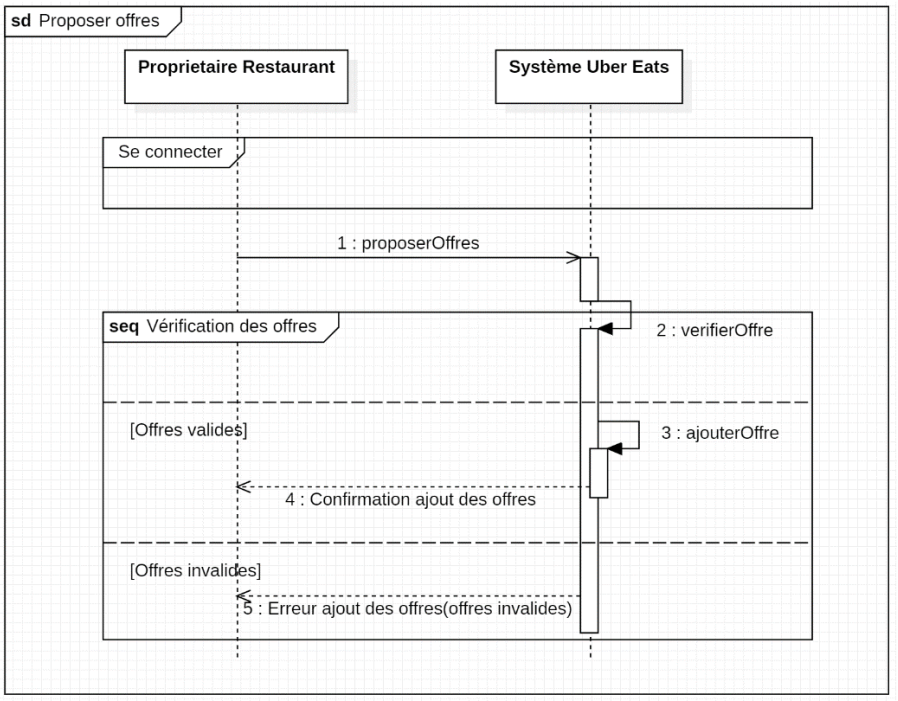
Une fois la commande payée, le système récupère la position GPS du livreur pour la rendre consultable sur l'application du client.



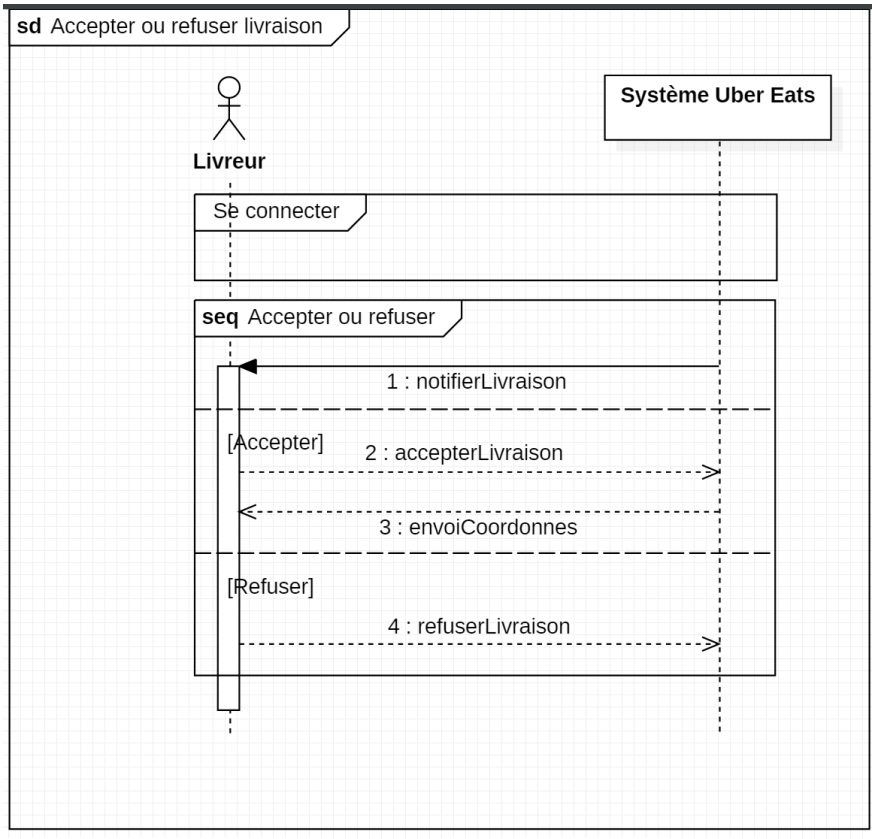
Pour noter, le système envoie une demande de note du service (sous forme de notification). Le client a deux choix : Soit il accepte la demande de notation, et le système envoie le formulaire de note, soit il refuse et le système ne fait rien. Dans le cas où l'utilisateur accepte la demande, il note le service, et le système récupère le formulaire.



Une fois connecté le propriétaire peut enregistrer son restaurant. Il va premièrement envoyer les données de son restaurant (nom, adresse, …). Le système vérifie les données, si les données sont validées le restaurant est ajouté à la base de données. Sinon les données ne sont pas validées, un message d'erreur est envoyé au propriétaire du restaurant et son restaurant n'est pas ajouté à la base de données.

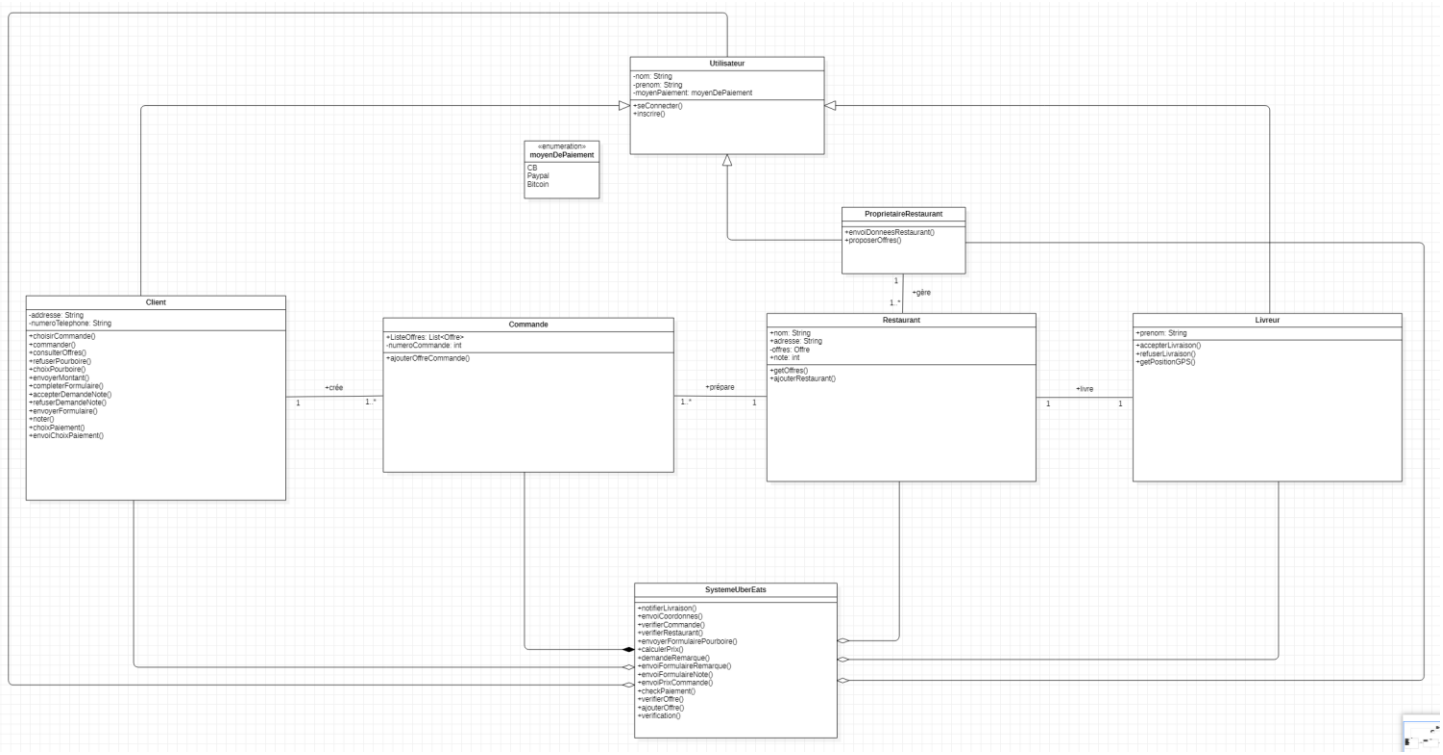


Une fois un propriétaire de restaurant connecté, il peut rentrer les offres qu'il propose dans son restaurant. Les offres sont vérifiées, si elles sont valides elles sont ajoutées et une confirmation d'ajout est envoyée sinon une erreur est retournée.



Le livreur une fois connecté, reçoit des notifications de livraison. Le livreur fait alors le choix d'accepter ou non la course. Dans le cas où il accepte, il reçoit alors les coordonnées correspondantes. Sinon il ne se passe rien.

Diagramme de classes :



Pour le diagramme de classes, nous avons décidé de représenter la classe Utilisateur, qui est une généralisation des classes Client, ProprietaireRestaurant, et Livreur. Nous avons aussi les classes Commande, Restaurant, et SystemeUberEats.

Chaque Client est défini par les attributs privés suivants : un nom, un prénom, une adresse, un numéro de téléphone et un moyen de paiement.

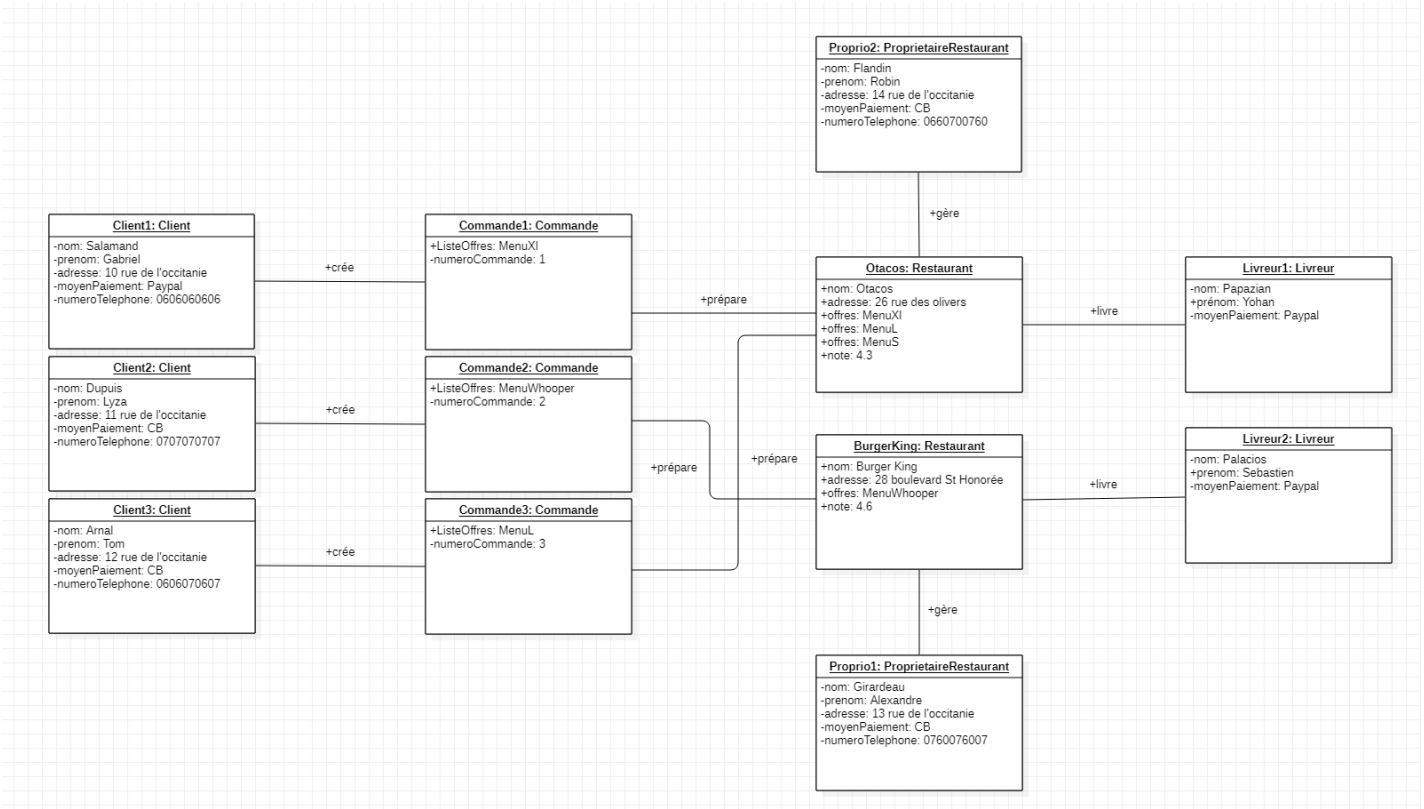
Une commande est définie par une liste d'offres et un numéro de commande.

Les livreurs et les propriétaires de restaurant sont des spécialisations de l'utilisateur car ils peuvent effectuer des actions spécifiques. Par exemple, un propriétaire de restaurant peut envoyer les informations sur son établissement et répertorier ses offres afin de les référencer sur Uber Eats.

Nous avons déterminé que la classe Commande était une composition du SystemeUberEats car il ne peut pas exister de commande sans le système.

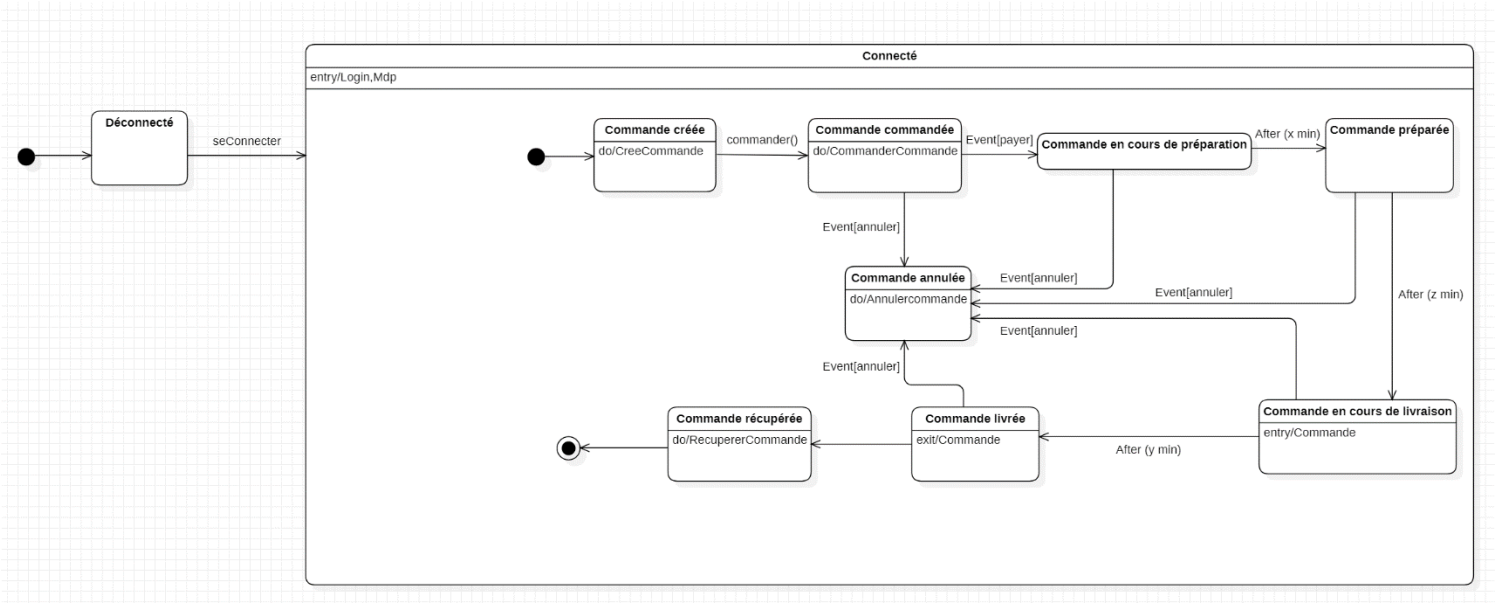
Cependant les autres classes sont des agrégations car elles peuvent exister sans le SystemeUberEats.

Diagramme d’objet :



Ici, avec ce diagramme d’objet, nous illustrons un cas concret, avec la création de 3 Clients, 3 Commandes, 2 Propriétaires de restaurant, 2 Restaurants et 2 Livreurs.

Diagramme d’état transitions :



L’état initial est l’état déconnecté. Une fois connecté, le client peut créer une commande, et s’il le souhaite la valider. Une fois payée, la commande passe en cours de préparation par le restaurant. Après un certain temps, la commande est prête et est en attente de livreur. Une fois récupérée par un livreur, elle est alors en cours de livraison. Après le temps de trajet, la commande est livrée et récupérée par le client. De l’état “commande commandée” à l’état “commande livrée”, il est possible que la commande soit annulée à tout moment pour différentes raisons comme: impossibilité d'accès du livreur, problème dans le restaurant, ...