

Sauriez-vous faire faire le travail de modélisation à ChatGPT ?

i.e., le guider, identifier ses erreurs, ...

Essayons.

Ce qui suit n'a

ni la prétention d'apprendre à diriger chatGPT,

ni d'être une bonne solution pour le faire (en anglais, il serait peut-être meilleur !).

Le principal objectif est de renforcer vos connaissances sur la modélisation en UML !

Warning! Blaming ChatGPT is an unacceptable excuse and reflects poorly on your skills as students.

You are always responsible for your production, regardless of the method used to obtain it.



Enoncé : étude de cas de l'an dernier

Il est possible pour n'importe qui de créer une commande, de choisir sa boutique de récupération et la date de rendez-vous puis de payer le montant dû par carte de crédit ou en utilisant ses points.

Une commande consiste en un ensemble de "cookies" ainsi que la quantité voulue, parmi les recettes préexistantes qui font la renommée de l'établissement (par exemple, le "Dark Temptation", le "Chocolalala" ...).

Chaque recette de cookie est définie par son type de pâte (`_dough_`), un arôme optionnel (`_flavour_`), jusqu'à trois garnitures (`_topping_`), un type de mélange (`_mix_`), et finalement un type de cuisson (`_cooking_`).

Il suffit ensuite de présenter le bon de commande (chaque recette a un prix donné) obtenu depuis le système pour récupérer sa commande en magasin.

Une nouvelle recette est proposée tous les mois et certaines recettes peuvent disparaître faute d'acheteurs.

En créant un compte utilisateur sur CoD, il est possible d'adhérer au "Loyalty program", qui offre après 30 cookies commandés un discount de 10% sur la prochaine commande.

Les responsables de magasin peuvent définir leurs horaires d'ouverture au niveau à un client de choisir un horaire de retrait qui n'entre pas dans les horaires de la

Les prix hors taxe des ingrédients sont les mêmes d'un magasin à un autre, mais magasin.

Le succès du CoD repose sur le bouche-à-oreilles.



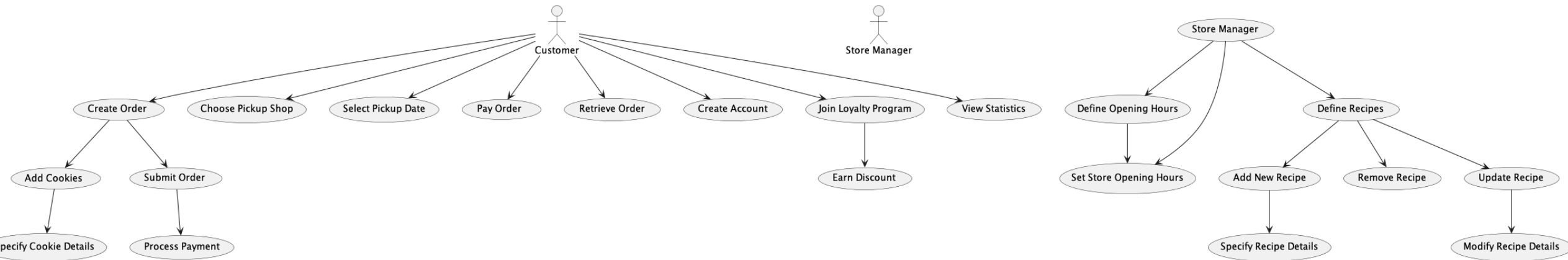
The Cookie Factory (TCF) is a major bakery brand in the USA. The Cookie on Demand (CoD) system is an innovative service offered by TCF to its customer. They can order cookies online thanks to an application, and select when they'll pick up their order in a given shop. The CoD system ensures to TCF's happy customers that they'll always retrieve their pre-paid warm cookies on time.

USE CASES



M

V0 : chatGPT feeds with the only previous statement and the request for a use case diagram in *PlantUML* : result



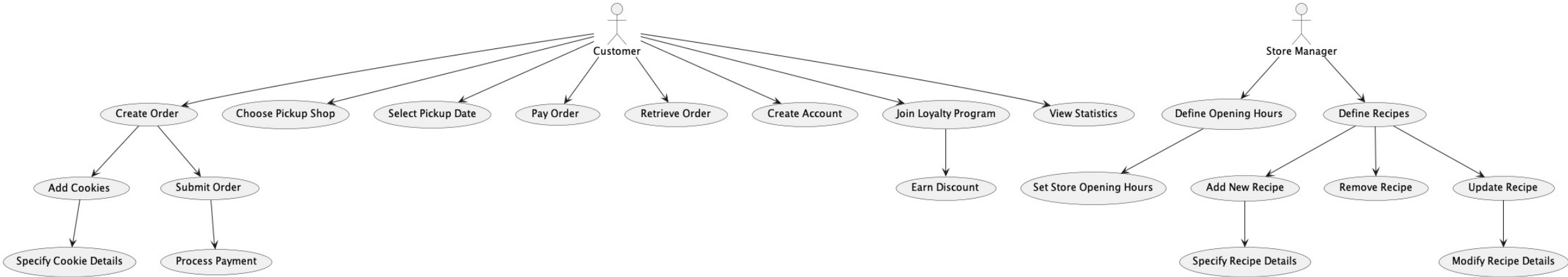
Attention ces diagrammes sont faux !



Tu confonds l'acteur "*Manager*" avec le Use case *store Manager* et tu n'as associé aucun cas d'utilisation au manager.



*Un UC sans acteur n'est pas un UC !
Et inversement*



A use case without an actor is not a use case!

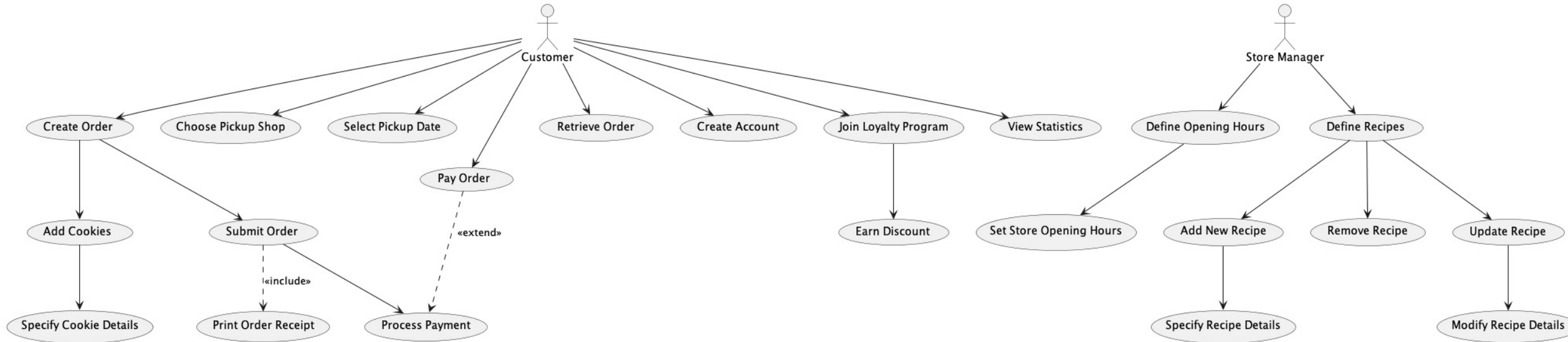
An actor not associated with a use case is not an actor

Attention ces diagrammes sont faux !

Les seules relations entre les use cases sont la spécialisation, l'extension, ou le include, est-ce que tu connais? Dois-je t'en dire plus pour que tu corriges en étiquetant correctement les relations entre cas d'utilisation ? *Une erreur très fréquente !!!*



Non, il ne connaît pas ...



Attention ces diagrammes sont faux !

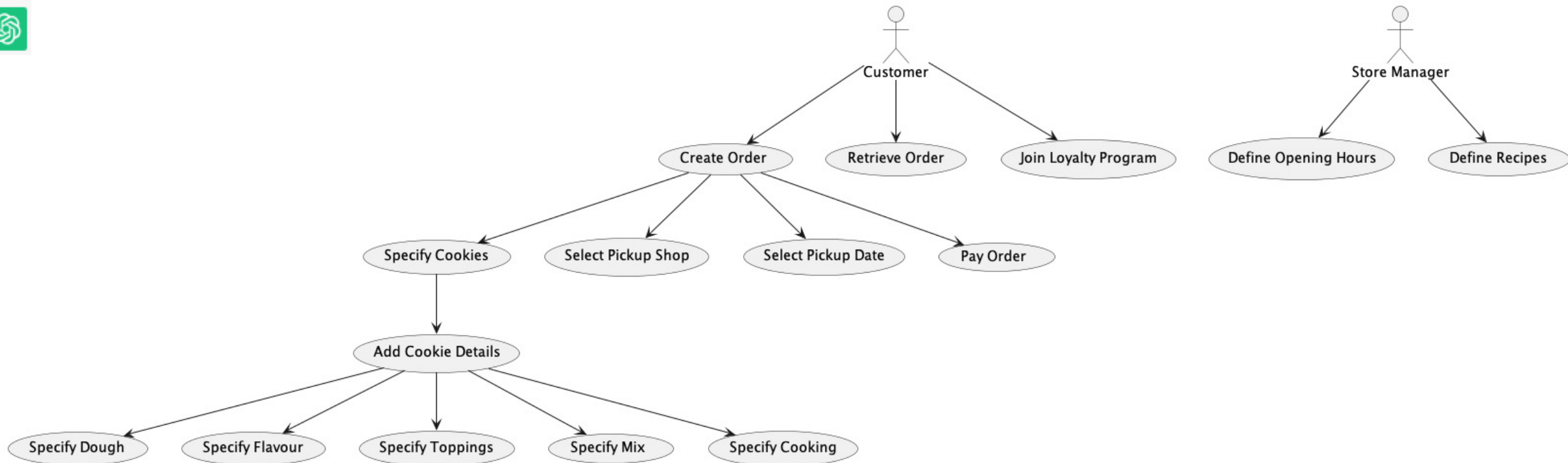
M

Tu t'es trop éloigné de l'énoncé et tu vas trop dans les détails des use cases, chaque cas d'utilisation doit avoir de la valeur pour l'acteur.



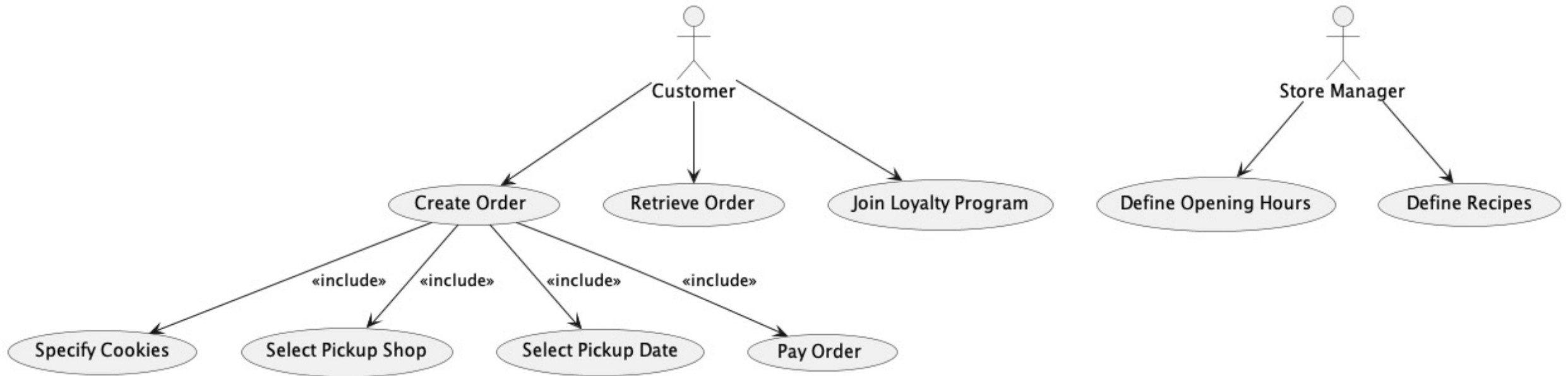
Chaque UC doit apporter de la Valeur à un acteur primaire.

Bon, il faut retravailler le prompt !



Attention ces diagrammes sont faux !

By your own



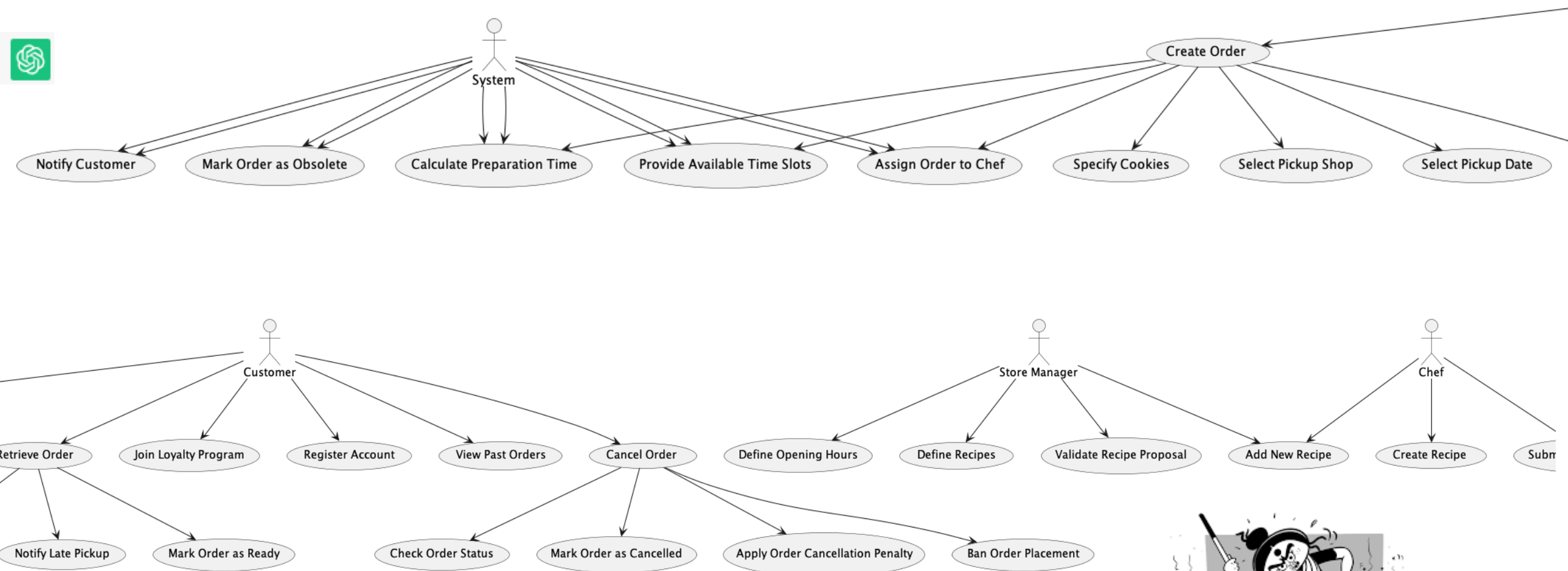
Attention ces diagrammes sont faux !

M On reprend en lui donnant notre solution puis on ajoute les besoins suivants :

- Le choix des ingrédients repose sur des entreprises partenaires qui proposent leurs nouveautés par l'intermédiaire de catalogues électroniques. Une recette ne peut donc contenir que des ingrédients présents dans ces catalogues.
- Les chefs de la Marque élaborent de nouvelles recettes en fonction des ingrédients puis les soumettent aux dirigeants. Ceux-ci valident ou invalident les propositions notamment en fonction du coût des ingrédients.
- Un internaute doit se voir proposer la possibilité de s'enregistrer pendant qu'il passe commande.
- Un client peut consulter les commandes passées.
- Quand on passe une commande
 - Le temps de préparation d'une commande est calculé comme étant la somme des temps de préparation de chaque cookie (un seul cuisinier par commande) + 15 minutes pour parer aux imprévus et intégrer le temps de cuisson du dernier cookie (pas de limite sur le nombre de cookies enfournées).
 - On découpe le temps en tranches de 15 minutes et on affecte un cuisinier par tranche de 15 minutes ...
 - Le système propose au client tous les horaires pour lesquels un cuisinier est disponible
- Dès que le client valide sa commande, elle est affectée à un cuisinier.
 - Un client peut annuler une commande tant qu'un cuisinier n'a pas commencé à travailler sur la commande ; il n'est alors plus possible de l'annuler. Un client qui annule deux commandes en 8 minutes, ne peut plus passer de commandes pendant 10 minutes.
 - Un client qui n'a pas retiré sa commande après 5 min, est notifié par sms qu'il peut venir la chercher, puis 1 heure après que nous gardons sa commande prête.
 - Une commande qui n'a pas été récupérée au bout de 2 h après l'heure de retrait prévue est considérée comme "obsolete".



V2 -



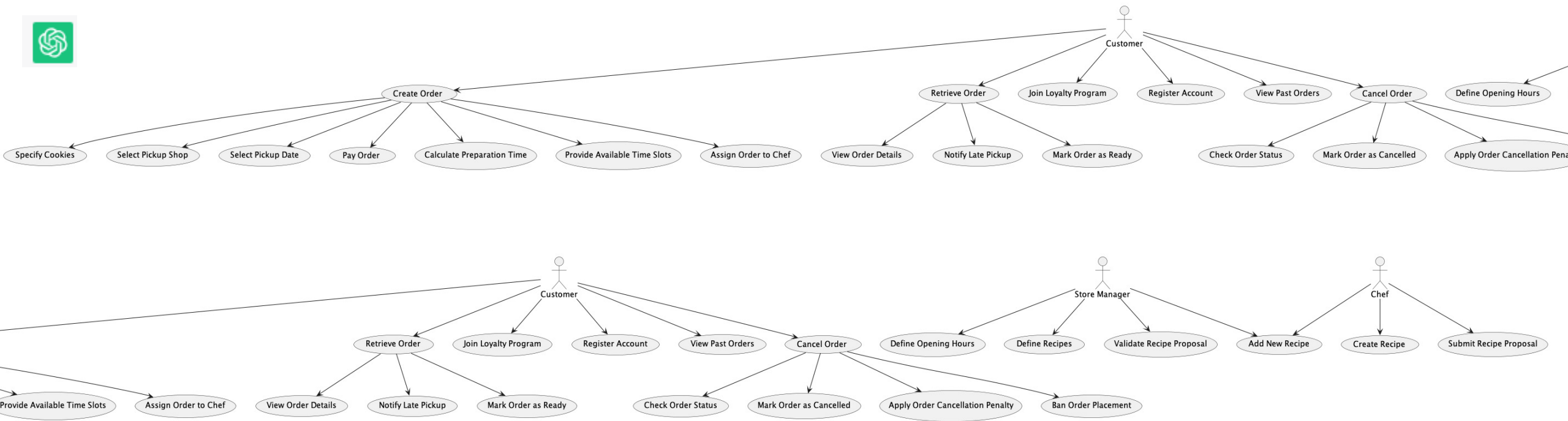
" Can your ChatGPT clean this up for me, please?? "

Attention ces diagrammes sont faux !



V2 - Le système n'est pas un acteur, c'est le système lui-même que l'on modélise.

Une erreur grave, on ne sait plus ce qui est modélisé.

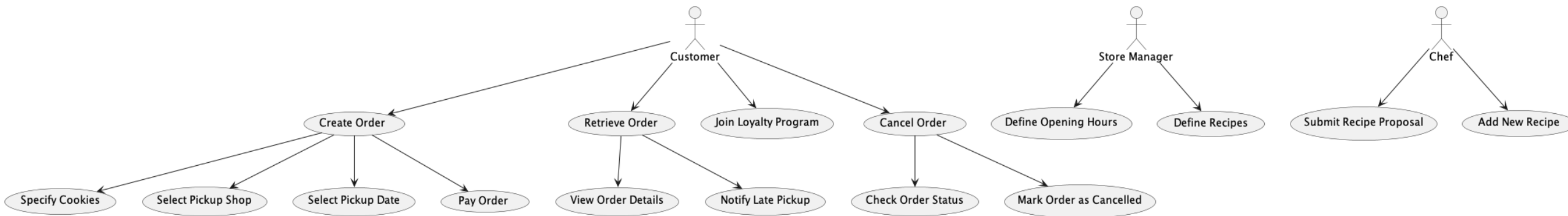


Attention ces diagrammes sont faux !



V2 - Peux-tu éliminer tous les cas d'utilisation où un acteur n'interagit pas avec le système?

Une autre règle évidente pour capturer les limites du systeme sans se perdre.



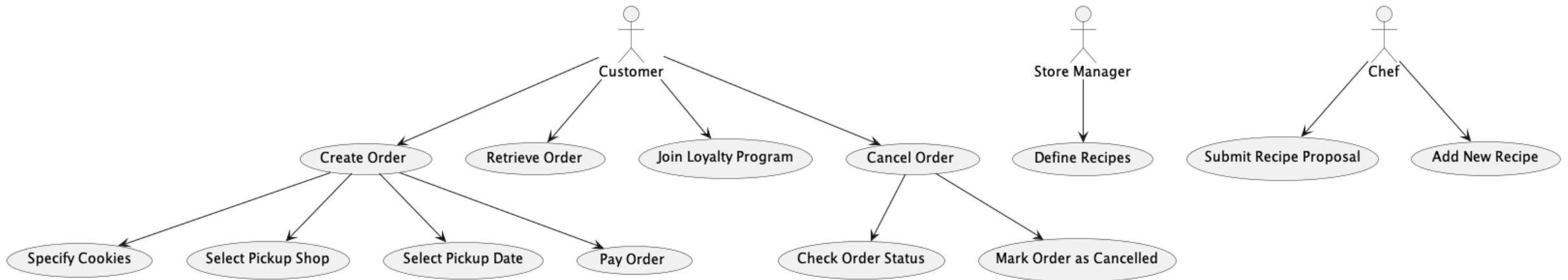
Attention ces diagrammes sont faux !

V2 - Enleve les cas d'utilisation qui sont trop simples au sens où ils représentent très peu de scénario différents.



Les UC doivent avoir de la Valeur en représentant au moins plusieurs scénarios possible.

Bon, il faut retravailler le prompt, si c'est possible ...



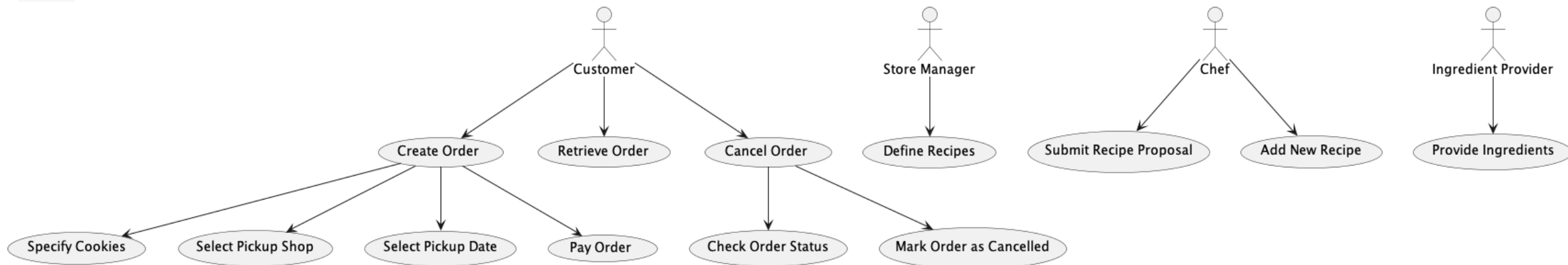
Attention ces diagrammes sont faux !

N'as-tu pas oublié des acteurs secondaires?



Identifier tous les acteurs est essentiel pour déterminer les limites du système à modéliser.

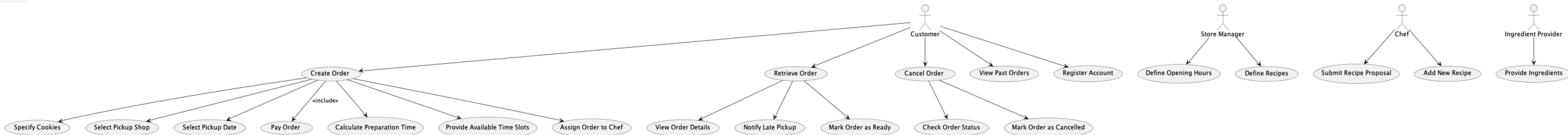
Outch, on ne peut vraiment pas lui faire confiance, Il n'en a identifié aucun !



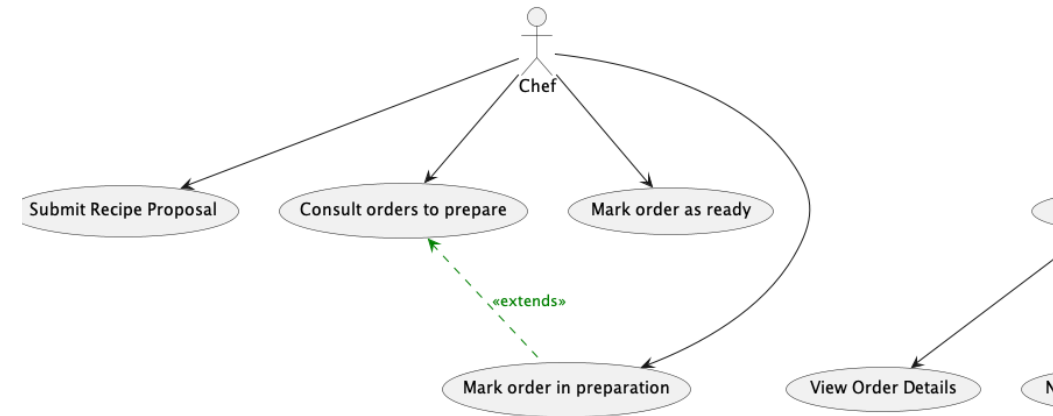
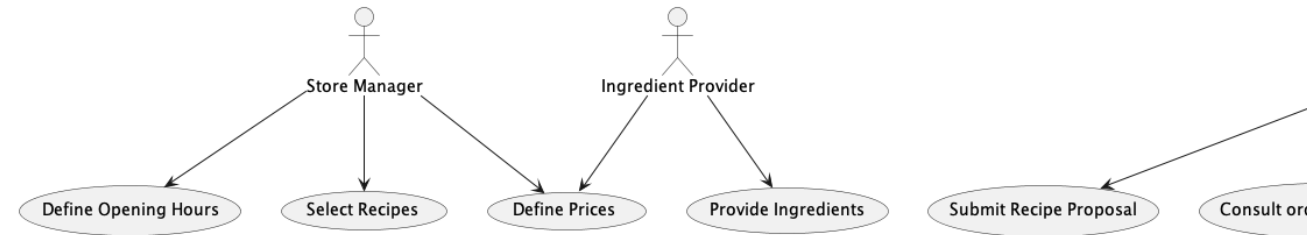
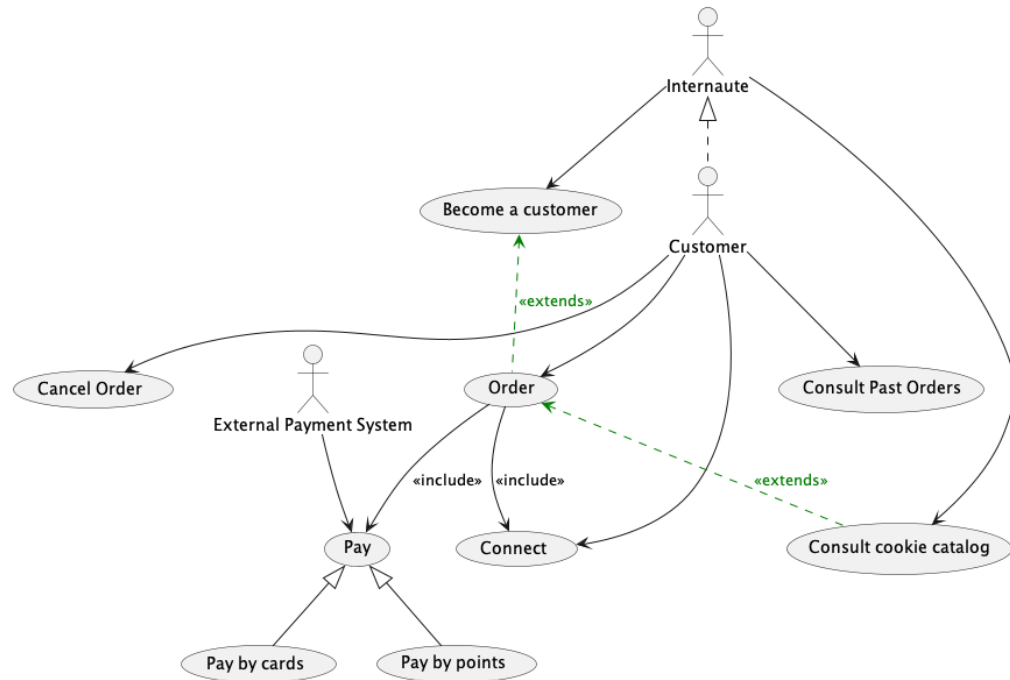
Attention ces diagrammes sont faux !

Erreurs fréquentes et "graves"

1. Le **"SI"** n'est ni un acteur, ni un cas d'utilisation (il est représenté par l'ensemble des cas d'utilisation).
2. La **base de données** n'est pas un acteur.
3. Pour chaque cas d'utilisation, les acteurs sont bien les personnes ou systèmes informatiques qui interagissent directement avec le SI du CoD (la personne qui "tape" sur le clavier, pas celle qui regarde par-dessus l'épaule!).
4. Si un cas d'utilisation ne fait pas intervenir le système informatique, il n'a pas lieu d'être (c'est une activité du métier qui n'est juste pas informatisée).
5. Les relations entre les cas d'utilisation ne sont jamais des relations d'ordre. Il n'y a qu'héritage, inclusion ou extension.
6. 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.



Based on last year's student submissions



Il peut probablement vous aider,
mais uniquement si vous savez le diriger, donc si vous savez faire..



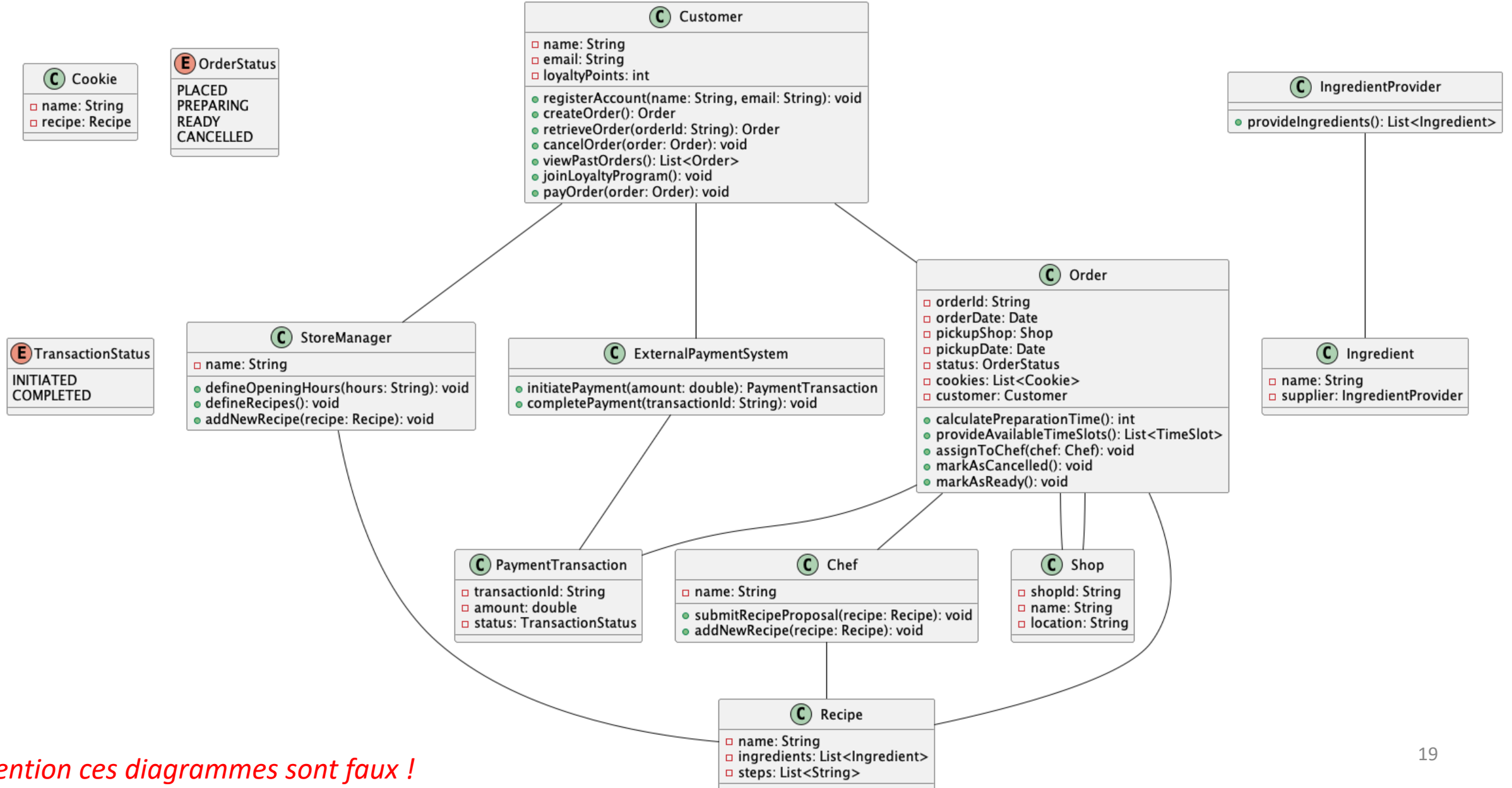
Back to the lesson



Domain Modeling



Domain modeling



Attention ces diagrammes sont faux !

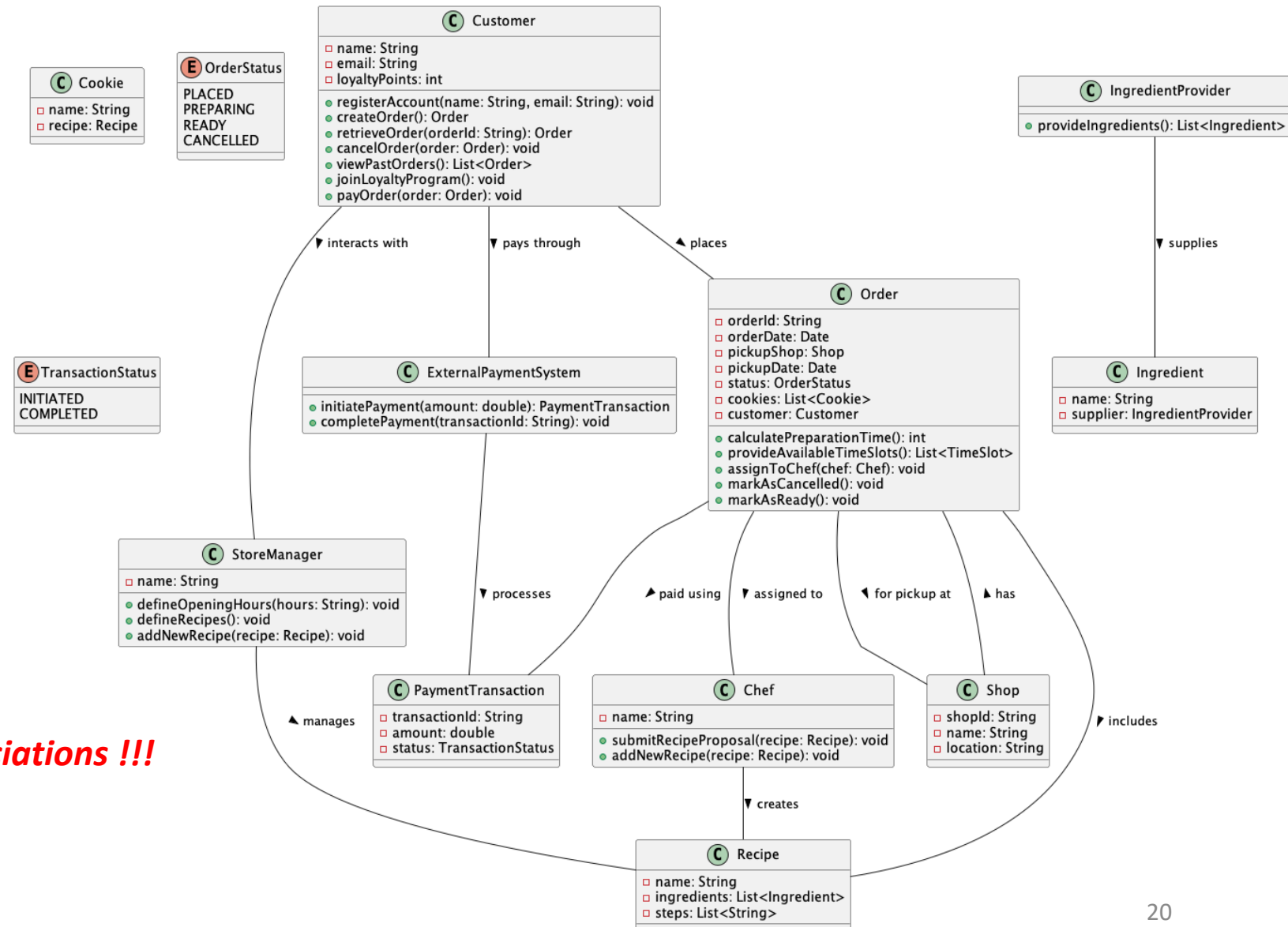
nomme les rôles et attribue des multiplicités à toutes les associations.



En l'absence de rôles et multiplicité sur les associations, il n'est pas possible de lire le diagramme, on ne peut qu'imaginer. Mais nous souhaitons modéliser..

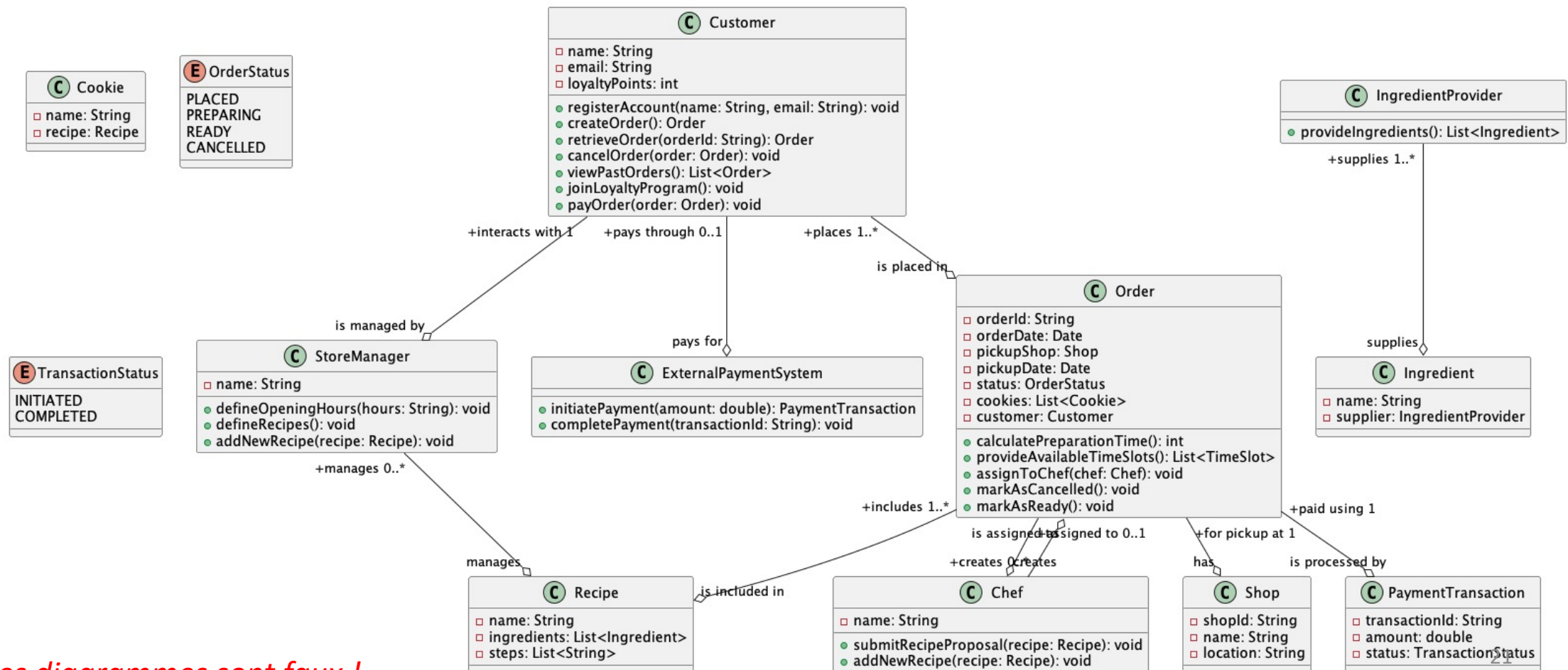


**C'est hyper faux !
Il confond avec des acteurs !!
Ce n'est pas du tout des associations !!!**



Pour exprimer une multiplicité et un rôle comme par exemple une Company a 0..* employees et un employé est employé par 0 ou une company on écrit : `Person "+employees 0..*" <-- "+employer 0..1` Company : employs note on link: [employs] is the name of the association. Peux-tu revoir le diagramme en tenant compte de cette information?

Oh! Le prompt n'est pas assez clair... Mais c'est trop long.



Attention ces diagrammes sont faux !

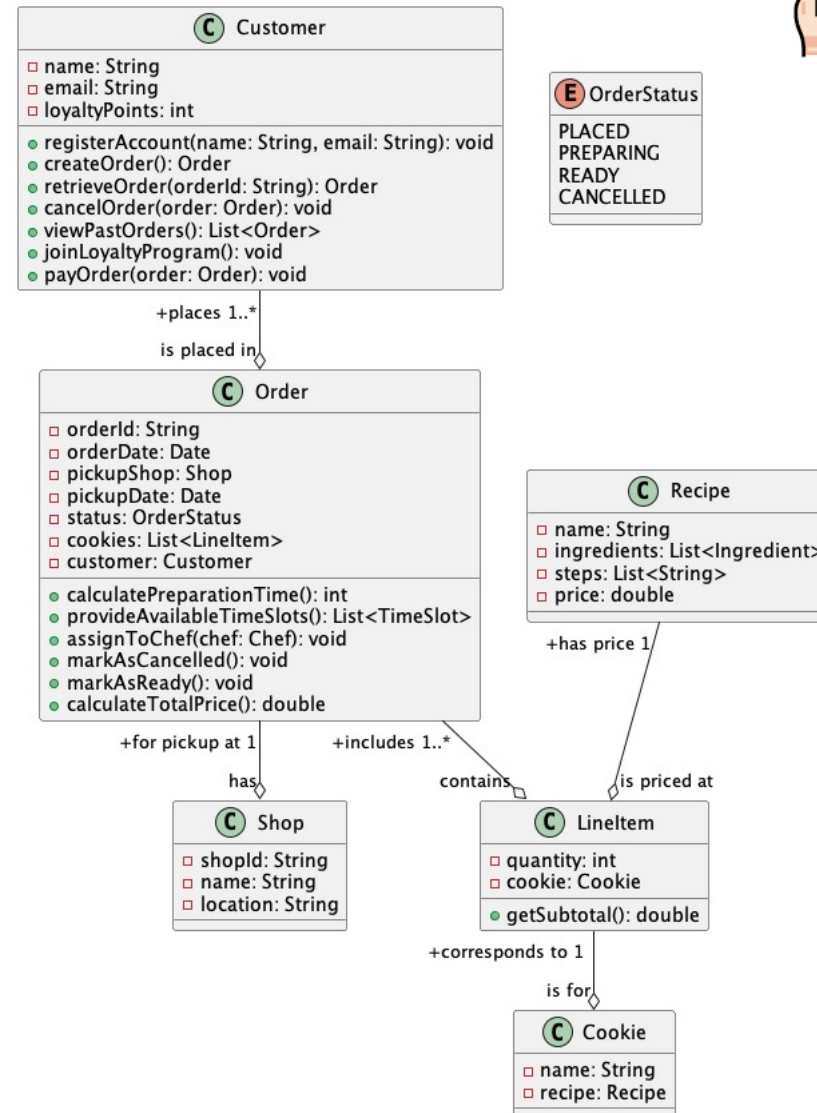
Soyons incrémental, décomposons-lui le problème.

plusieurs cookies correspondant à la même recette et chaque recette a un prix. Peux-tu me donner le diagramme correspondant uniquement à une commande en incluant ces informations?



Divide and Conquer

C'est bien mieux, modulo les associations...



Attention ces diagrammes sont faux !

pourquoi as-tu gardé une classe Cookie à quoi sert-elle ?

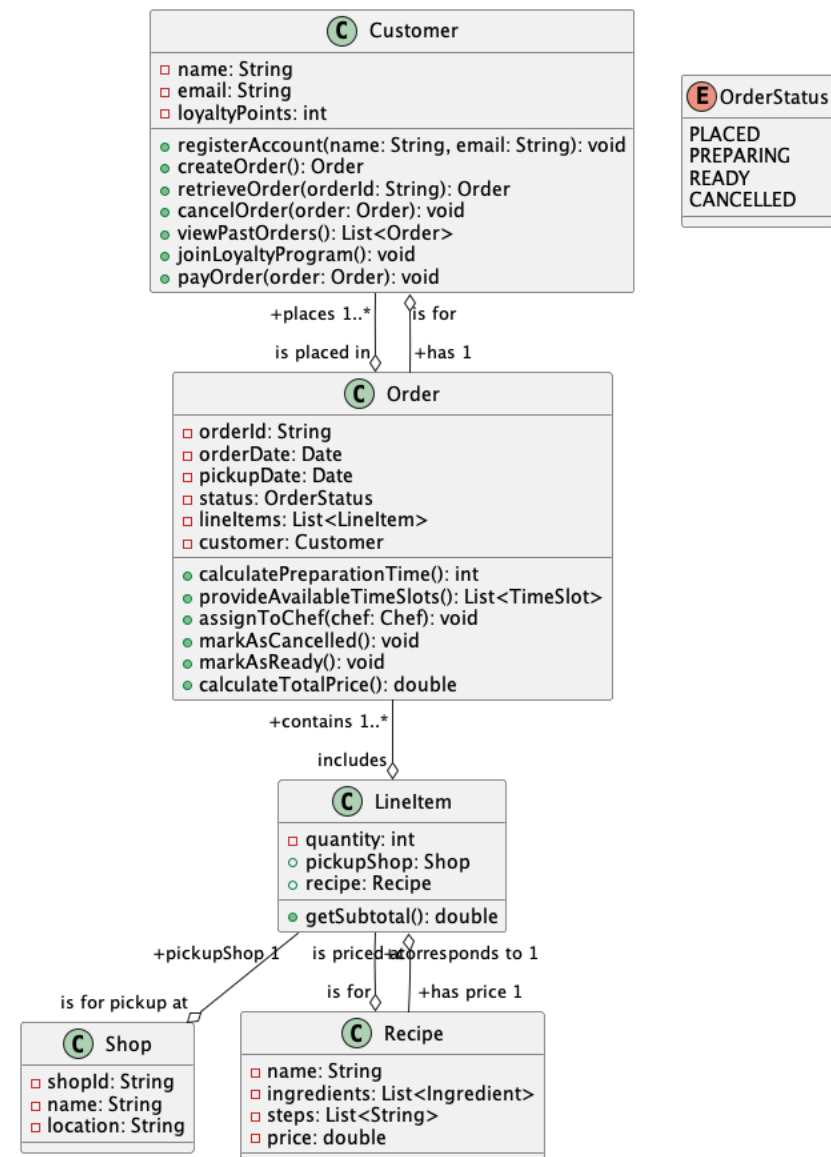


Rester coherent; éviter les duplications d'information

Oh... shop s'est décallé!!!



Je m'excuse pour la confusion. En effet, la classe Cookie n'est pas nécessaire dans ce contexte, car elle ne semble pas avoir de rôle distinct par rapport à la classe LineItem qui représente déjà les éléments de commande. Voici le diagramme de classes mis à jour en supprimant la classe Cookie :



Attention ces diagrammes sont faux !

Peux-tu éliminer les attributs qui correspondent à des associations comme pickupShop et donner comme nom au rôle (ici has) plutôt le nom de l'attribut c'est à dire pickupShop dans cet exemple

Nouvelles tentatives sur les associations...

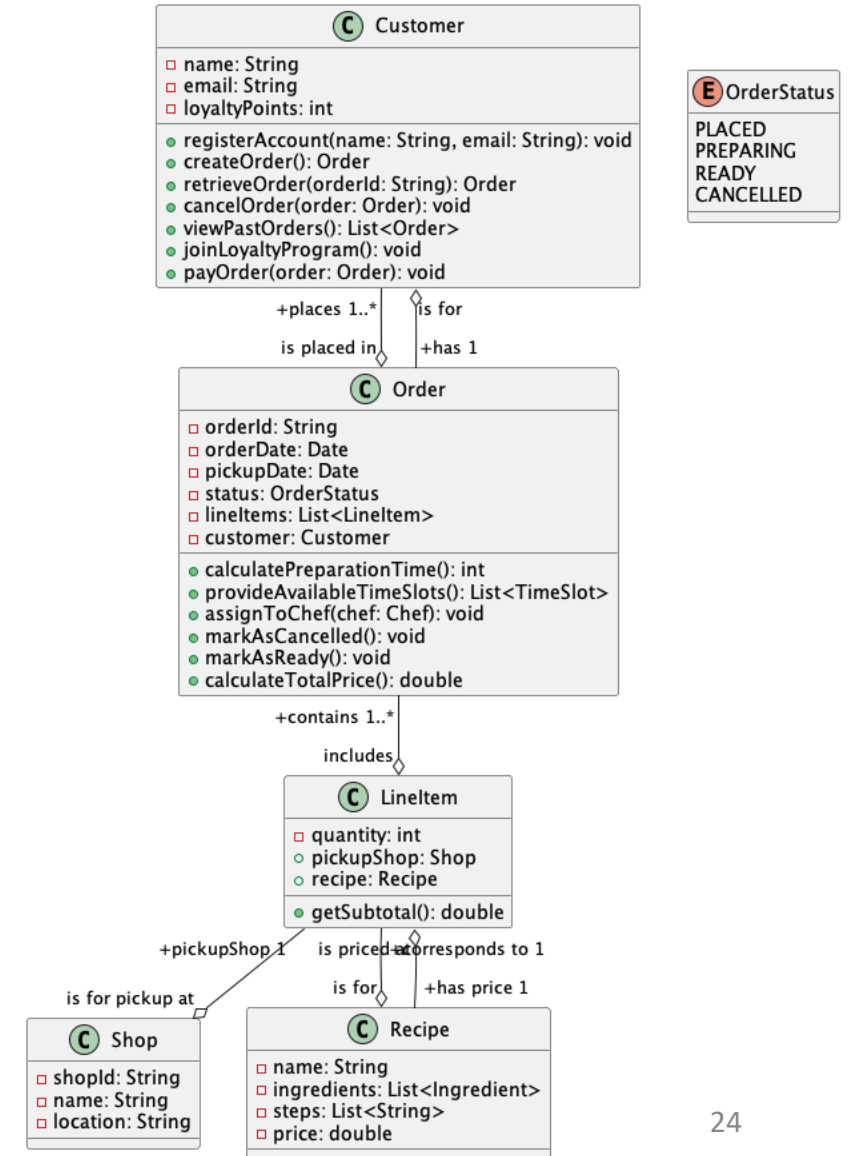


Les associations capturent les dépendances.

- Utiliser des attributs c'est passer à l'implémentation (est-il logique de travailler avec une liste d'items ?)
- Incohérence : y-a-t-il un attribut `lineltems` et un attribut `includes` ?

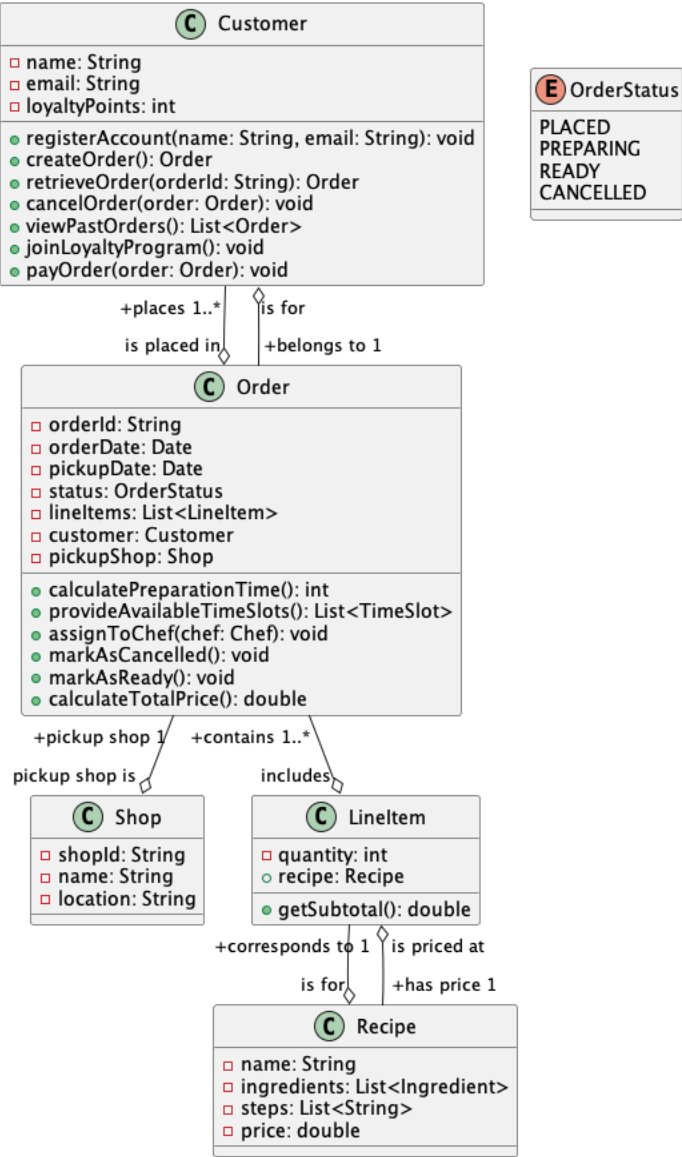


Je comprends, vous voulez utiliser le nom de l'attribut comme nom de rôle. Voici le diagramme de classes mis à jour en tenant compte de vos demandes :



Les multiplicités sont aussi à l'envers un Order a un seul pickupShop (1 du côté Shop) et plusieurs bons de commandes peuvent correspondre au même shop (0..* du côté Order).

Ultimes tentatives sur les associations...

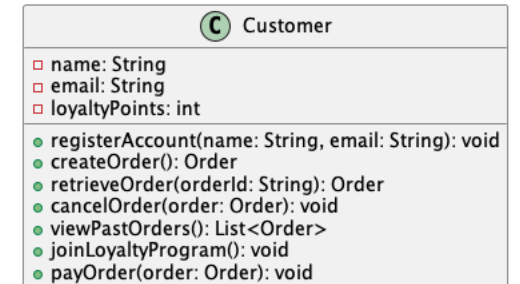
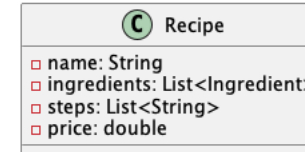
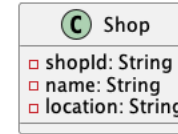
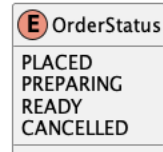
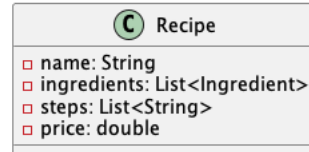
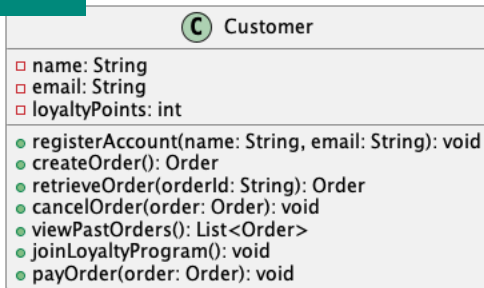


Ultime tentative

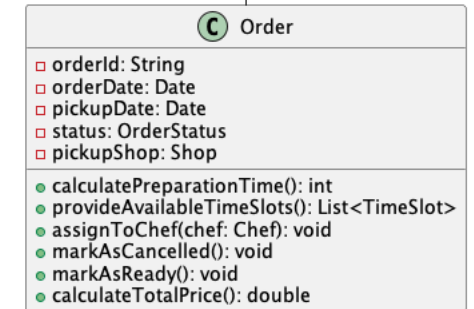
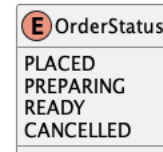
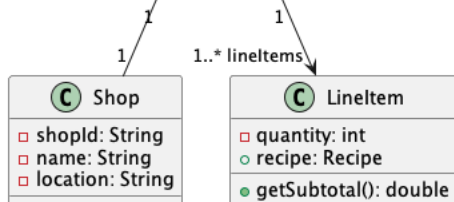
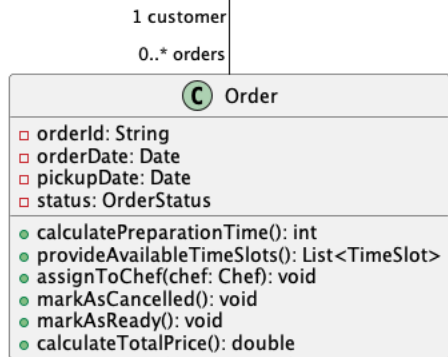
A la main on lui donne le diagramme à gauche et il génère le diagramme de droite



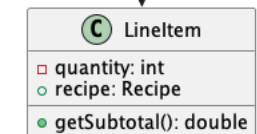
M



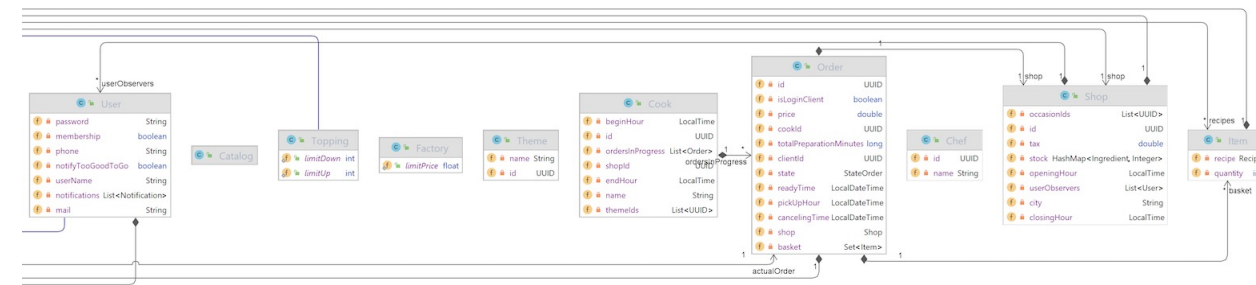
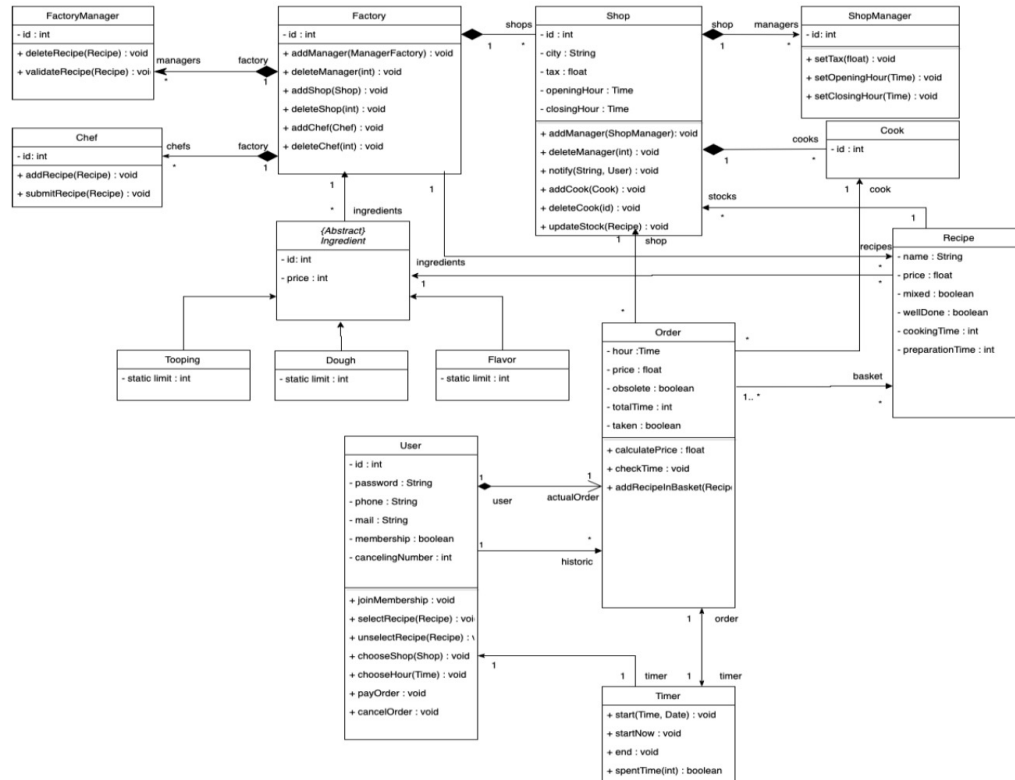
1 customer
0..* orders



1
1..* lineItems



Based on last year's student submissions (same team : begin – end)



ChatGPT peut probablement vous aider, mais uniquement si vous savez le diriger, donc si vous savez faire..

Laissons-lui le mot de la fin 😊



Voici une suggestion pour le contenu de la conclusion de votre présentation, mettant en avant l'utilisation de mon intervention pour générer des diagrammes UML :

- **Conclusion : Génération de Diagrammes UML avec l'Assistance de ChatGPT**

En conclusion, nous avons vu comment l'assistance de ChatGPT peut être un outil précieux pour générer des diagrammes UML. Voici comment vous pouvez l'utiliser efficacement :

- 1. Définir Vos Besoins** - Commencez par clarifier vos besoins en matière de diagrammes UML. Identifiez les classes, les relations, les cas d'utilisation ou tout autre élément que vous souhaitez représenter.
- 2. Formuler Vos Demandes** - Utilisez des instructions claires et concises pour formuler vos demandes à ChatGPT. Par exemple, vous pouvez demander : "Pouvez-vous générer un diagramme de classe UML pour la classe Course dans Java avec ses attributs et méthodes principales ?"
- 3. Analyser et Personnaliser** - Examinez attentivement la sortie générée par ChatGPT. Vous pouvez personnaliser davantage le diagramme en demandant des modifications spécifiques ou en ajoutant des détails supplémentaires.
- 4. Intégration dans Vos Documents** - Une fois satisfait de la sortie, intégrez facilement le diagramme UML dans vos documents, présentations ou rapports. Vous disposez d'une représentation visuelle claire de votre concept ou de votre modèle.
- 5. Collaboration Simplifiée** - L'utilisation de ChatGPT pour générer des diagrammes UML facilite la collaboration au sein de l'équipe, car vous pouvez rapidement produire des diagrammes pertinents pour les discussions techniques.

En somme, l'assistance de ChatGPT peut accélérer et simplifier le processus de création de diagrammes UML, en vous permettant de vous concentrer davantage sur la conception et la communication de vos idées. N'oubliez pas de réviser soigneusement les résultats pour vous assurer qu'ils correspondent à vos besoins spécifiques, et utilisez cet outil comme un atout précieux dans votre boîte à outils de développement logiciel.

Back to the lesson

