

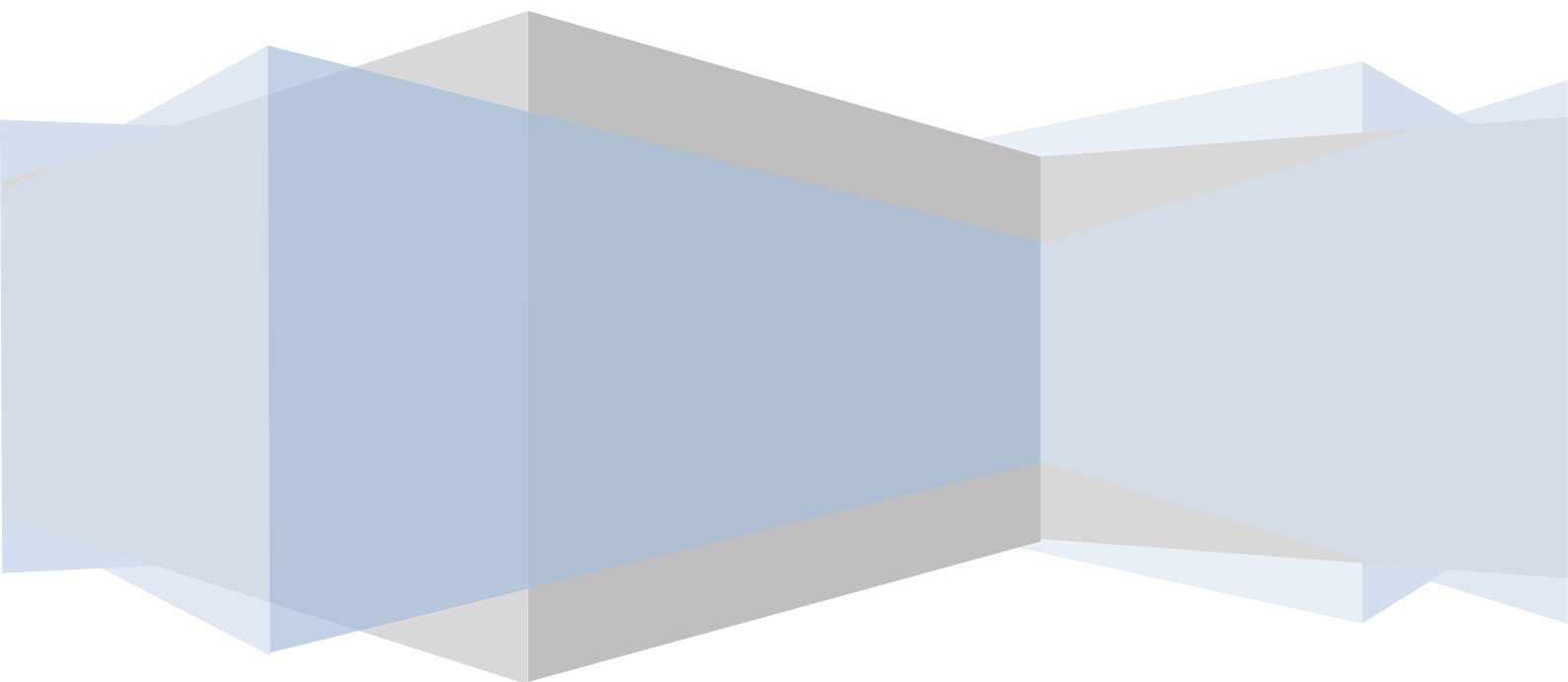


The Cookie Factory

Phase #1

Serveurs d'entreprise

Fernando GARRIGOS – Kaiwen YU



Sommaire

Contexte et vision du produit.....	3
Différents cas d'utilisation.....	3
Modèle de composants logiciels	9
Modèle de données et <i>mapping</i> objet-relationnel	10
Diagramme de déploiement	15

Contexte et vision du produit

The Cookie Factory (TCF), leader américain dans la fabrication de cookies artisanaux, souhaite mettre en place un service innovant pour ses clients, appelés « *Cookie On Demand* » (CoD). Ce système permettra aux clients de TCF de créer leurs cookies personnalisés à partir d'une large gamme d'ingrédients et/ou de choisir parmi les recettes préexistantes, de passer commande en ligne et de spécifier le magasin, la date et l'heure à laquelle ils viendront la récupérer au magasin.

TCF a fait appel à notre entreprise pour développer ce système de commande de cookies en ligne. La première phase de ce projet est de fournir un travail d'étude sur l'architecture à mettre en place pour servir de support au système *Cookie On Demand* (CoD).

Pour répondre à la demande de l'entreprise TCF, nous proposons de développer un produit dont l'architecture permet de gérer d'une part la gestion des clients et d'autre part la gestion du magasin.

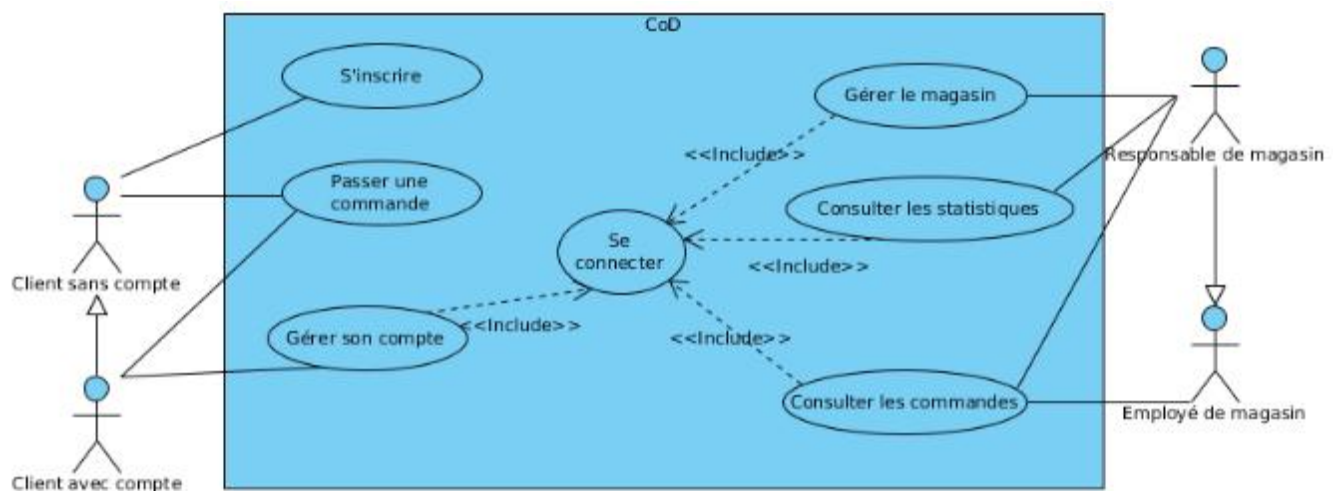
Différents cas d'utilisation

1. Cas d'utilisation CoD de haut niveau

Acteurs et fonctionnalités :

- ✦ **Client:** client non inscrit dans le système CoD, qui peut passer une commande et s'inscrire.
- ✦ **Client avec compte :** client ayant un compte dans le système CoD, qui en plus de passer une commande, peut, une fois connecté, gérer son compte.
- ✦ **Employé de magasin :** employé de l'entreprise TCF ayant un compte dans le système CoD, qui peut, une fois connecté, consulter les commandes du magasin.
- ✦ **Responsable de magasin :** responsable d'un magasin TCF ayant un compte CoD, qui peut, une fois connecté, en plus de consulter les commandes du magasin, gérer le magasin et consulter les statistiques du magasin.

Figure 1. Diagramme de cas d'utilisation de haut niveau

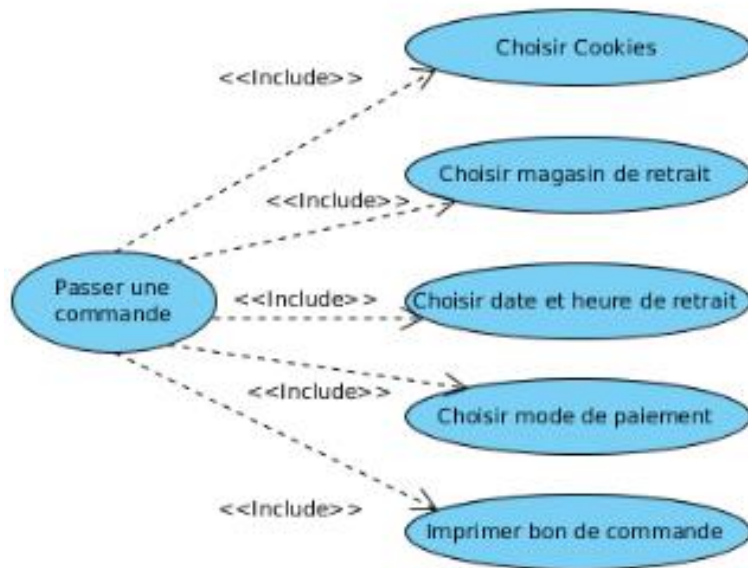


1.1. Cas d'utilisation « Passer une commande »

Acteurs et fonctionnalités :

- ★ **Client ou Client avec compte** : qu'il soit inscrit ou non, un client peut, pour passer commande : choisir les cookies, choisir le magasin où il souhaite retirer sa commande, choisir la date et l'heure du retrait de la commande et choisir le mode de paiement. Il peut alors imprimer son bon de commande.

Figure 2. Diagramme de cas d'utilisation « Passer une commande »

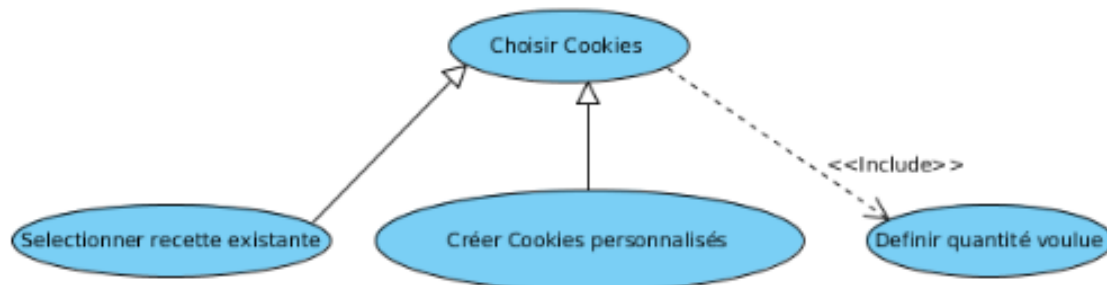


1.1.1. Cas d'utilisation « Choisir les cookies »

Acteurs et fonctionnalités :

- ✦ **Client ou Client avec compte** : le client a le choix entre sélectionner une recette existante ou créer ses cookies personnalisés. Il doit également définir la quantité de cookies voulue.

Figure 3. Diagramme de cas d'utilisation « Choisir les cookies »

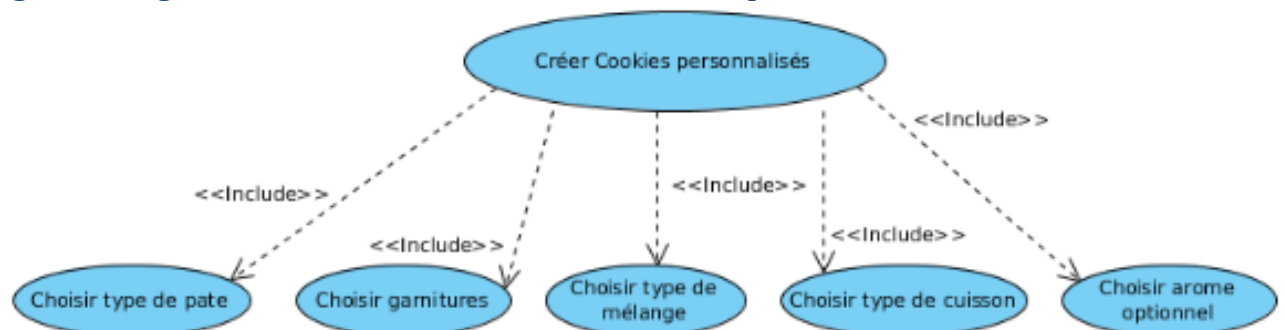


1.1.1.1. Cas d'utilisation « Créer cookies personnalisés »

Acteurs et fonctionnalités :

- ✦ **Client ou Client avec compte** : Pour créer des cookies personnalisés, le client doit choisir un type de pâte, jusqu'à trois garnitures, le type de mélange souhaité ainsi que le type de cuisson. Le client peut également choisir un arôme optionnel s'il le désire.

Figure 4. Diagramme de cas d'utilisation « Créer cookies personnalisés »

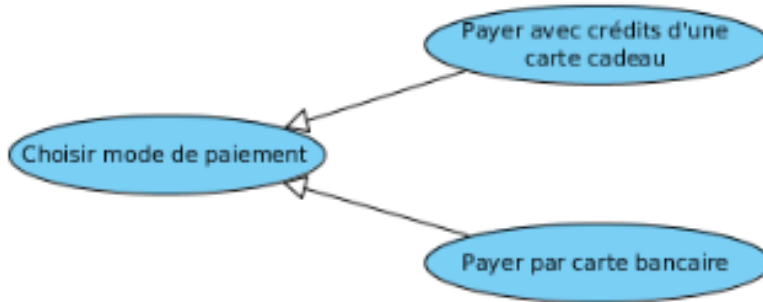


1.1.2. Cas d'utilisation « Choisir le mode de paiement »

Acteurs et fonctionnalités :

- ✦ **Client ou Client avec compte** : le client peut choisir de payer sa commande par carte bancaire ou d'utiliser des crédits provenant d'une carte cadeau.

Figure 5. Diagramme de cas d'utilisation « Choisir le mode de paiement »

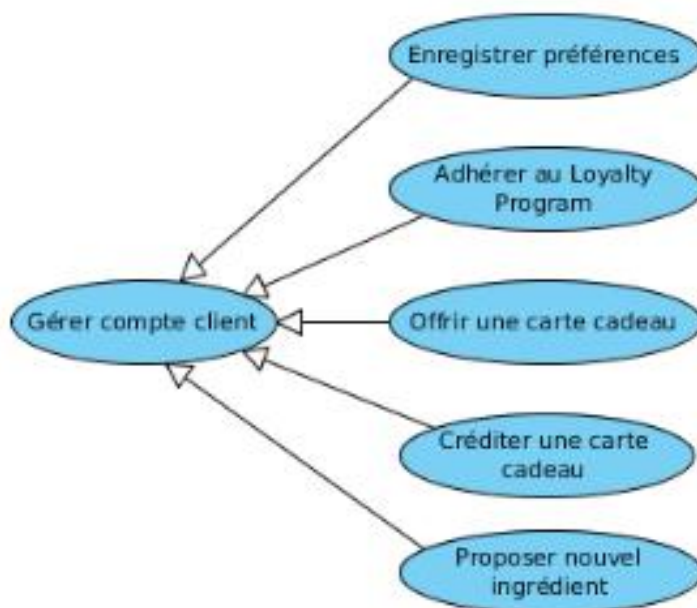


1.2. Cas d'utilisation « Gérer compte client »

Acteurs et fonctionnalités :

- ✦ **Client avec compte** : un client ayant un compte dans le système CoD peut enregistrer ses préférences, adhérer au programme de fidélité « Loyalty Program », offrir une carte cadeau à un autre client, créditer une carte cadeau (il pourra ensuite payer une commande avec les crédits accumulés), et proposer de nouveaux ingrédients.

Figure 6. Diagramme de cas d'utilisation « Gérer compte client »

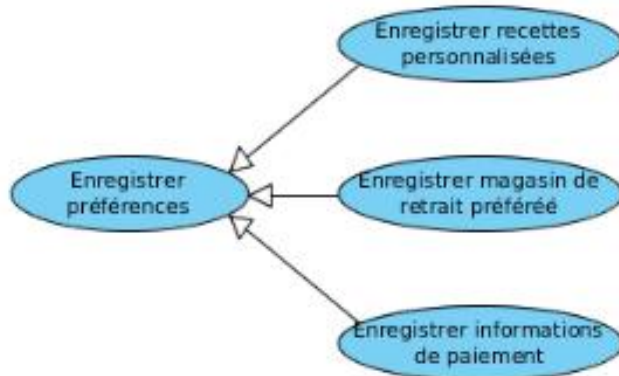


1.2.1. Cas d'utilisation « Enregistrer préférences »

Acteurs et fonctionnalités :

- ✦ **Client avec compte** : tout client inscrit peut enregistrer ses recettes de cookies personnalisées, ses magasins de retrait de commande favoris ainsi que ses informations de paiement.

Figure 7. Diagramme de cas d'utilisation « Enregistrer préférences »

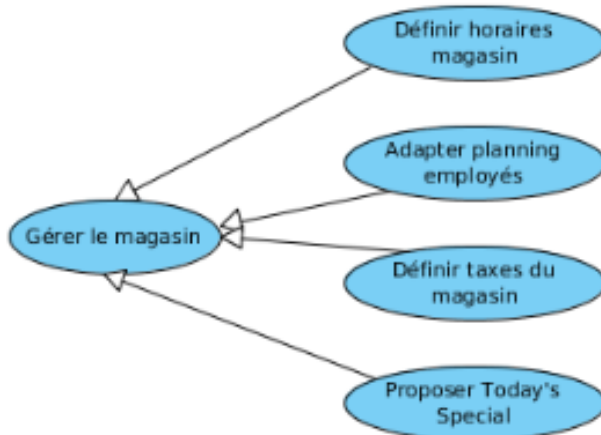


1.3. Cas d'utilisation « Gérer le magasin »

Acteurs et fonctionnalités :

- ✦ **Responsable de magasin** : le responsable du magasin en charge de la gestion du magasin, grâce au système CoD, peut: définir les horaires d'ouverture du magasin, adapter le planning des employés en atelier et en vente (en fonction des statistiques de commandes), définir les taxes à appliquer dans son magasin et proposer la recette du « today's special ».

Figure 8. Diagramme de cas d'utilisation « Gérer le magasin »

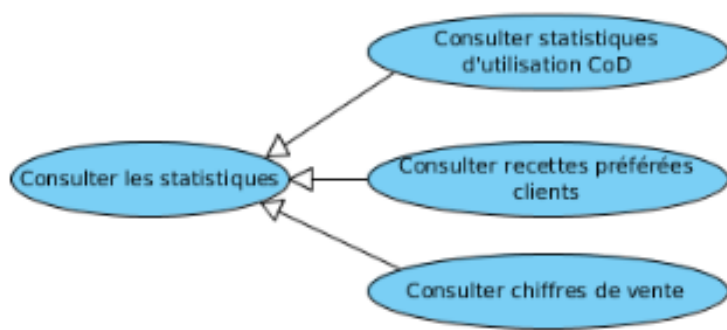


1.4. Cas d'utilisation « Consulter les statistiques »

Acteurs et fonctionnalités :

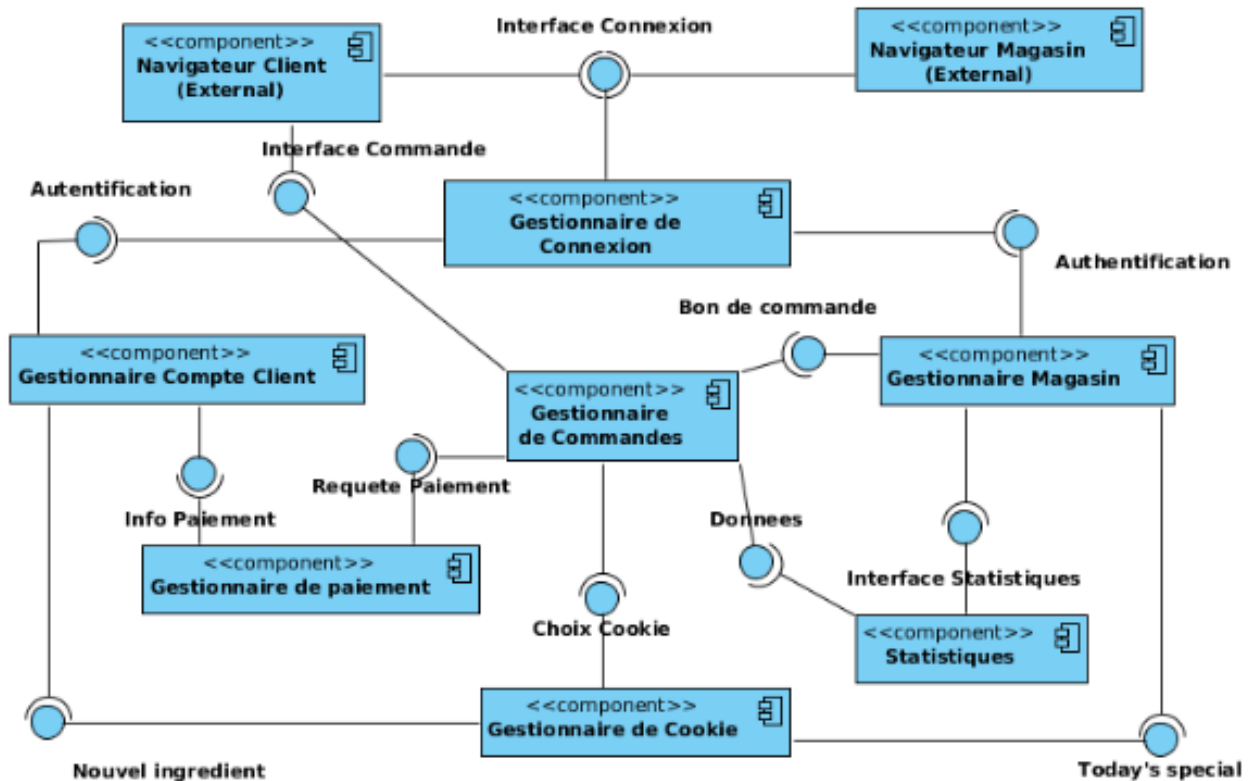
- ✦ **Responsable de magasin** : le gérant du magasin peut également consulter des statistiques relatives à l'utilisation de CoD, consulter les recettes préférées des clients et consulter les chiffres de vente annuels, mensuels, hebdomadaires ou quotidiens.

Figure 9. Diagramme de cas d'utilisation « Consulter les statistiques »



Modèle de composants logiciels

Figure 10. Diagramme de composants



Le diagramme de composants UML conforme ci-dessus distingue les différents composants du système CoD.

Le composant **Navigateur Client** permet à tout client d'accéder au **Gestionnaire de commandes** et d'effectuer ainsi toutes les opérations d'un Client (telles que définies dans les cas d'utilisations). Le **Navigateur Client** interagit également avec les **Gestionnaire de Connexion** qui permettra à un Client simple de s'inscrire et à un Client Avec Compte de se connecter et d'accéder au **Gestionnaire Compte Client**. Ce dernier est relié au **Gestionnaire de Cookies** afin que le Client Avec Compte puisse proposer de nouveaux ingrédients.

Le **Navigateur Magasin** est relié au **Gestionnaire de Connexion**, ce qui permet à un Employé et un Responsable de Magasin, une fois authentifiés, d'accéder au **Gestionnaire de Magasin** et d'effectuer les opérations qui leurs sont propres. Le **Gestionnaire de Magasin** interagit avec le **Gestionnaires de Commandes** pour récupérer les bons de commandes à préparer, avec le composant **Statistiques** (lui-même relié au **Gestionnaire de Commandes** pour récupérer les informations sur les commandes) afin de consulter les différentes statistiques du magasin et également avec le **Gestionnaire de Cookies** afin que le Responsable de Magasin puisse proposer le « Today's Special » quotidiennement.

Le **Gestionnaire de Commandes** est relié au **Gestionnaire de Paiement** afin d'autoriser le paiement des commandes et de récupérer les informations de paiement pour les Client Avec Compte. Pour cela, le **Gestionnaire de Paiement** est relié au **Gestionnaire Compte Client** dans lequel sont stockées les informations de paiement des Clients Avec Compte.

Le **Gestionnaire de Commandes** fait également appel au **Gestionnaire de Cookies** pour connaître le choix de cookies disponible.

Voici quelques exemples de méthodes appelées :

- ✦ **Authentification :**
boolean seConnecter(String nom, String prenom, String motDePasse)
boolean sInscrire(String nom, String prenom, String motDePasse)
- ✦ **Statistiques :**
Statistiques consulterStatistiques(ResponsableDeMagasin r)
- ✦ **Création de commandes :**
boolean commander(Client c)

Modèle de données et *mapping* objet-relationnel

A partir de nos diagrammes d'utilisation, nous avons défini les classes principales du système CoD : la classe Utilisateur, la classe Commande, la classe Magasin, la classe Cookie, et la classe Gestionnaire de Paiement. Nous avons représenté ces classes dans 5 diagrammes de classe différents, présentés ci-dessous.

Figure 11. Diagramme de classe Utilisateur

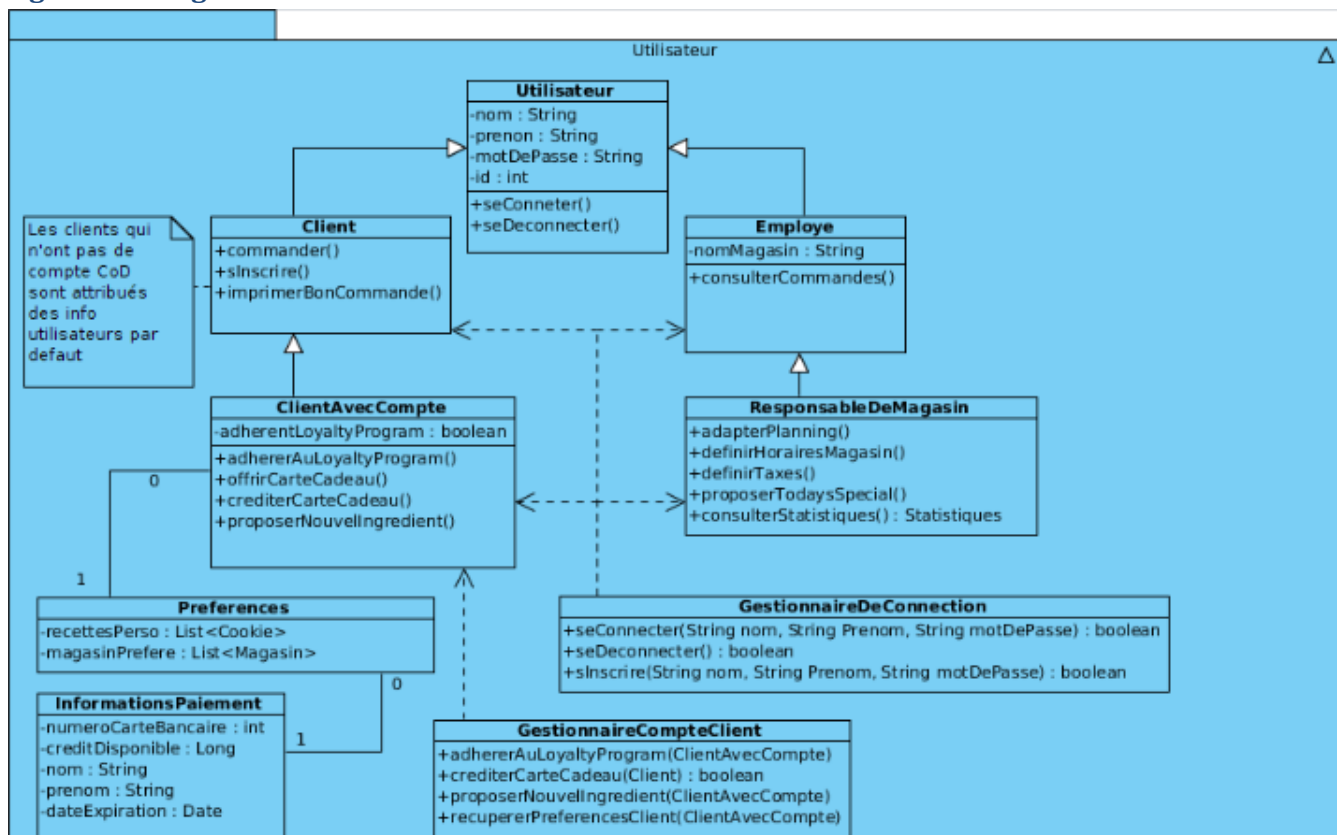


Figure 12. Diagramme de classe Commande

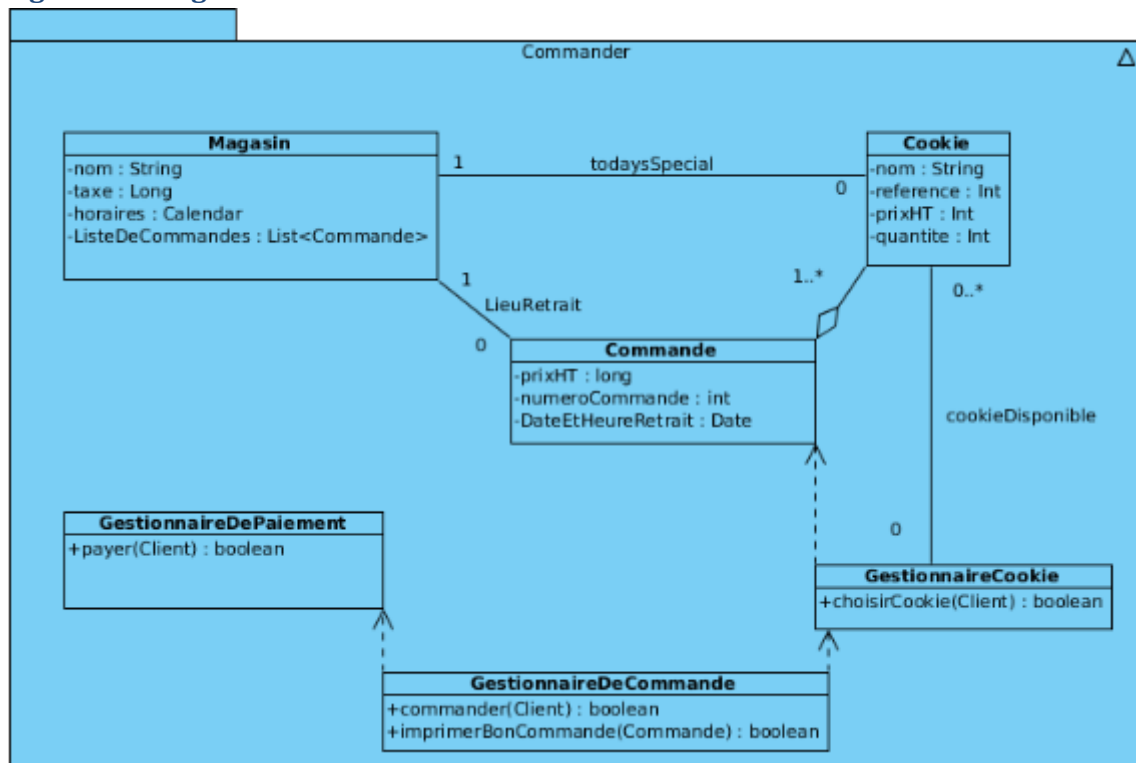


Figure 13. Diagramme de classe Magasin

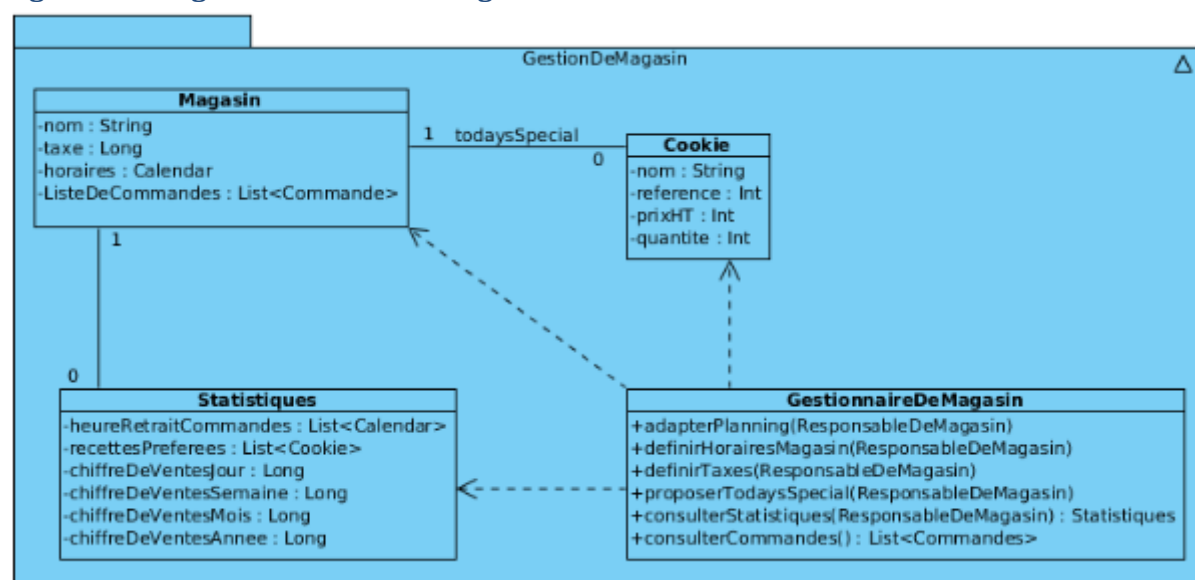


Figure 14. Diagramme de classe Cookie

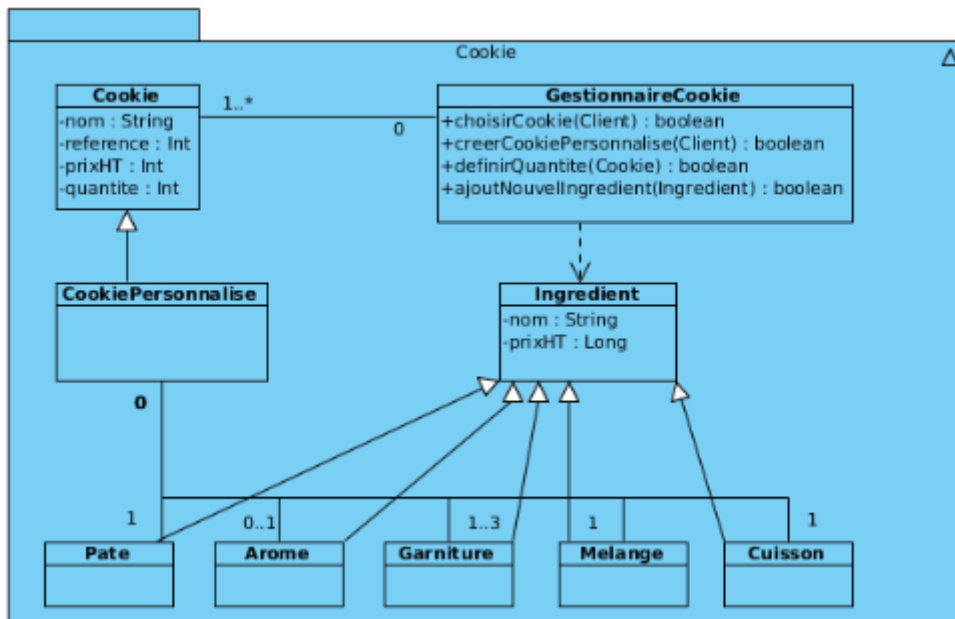
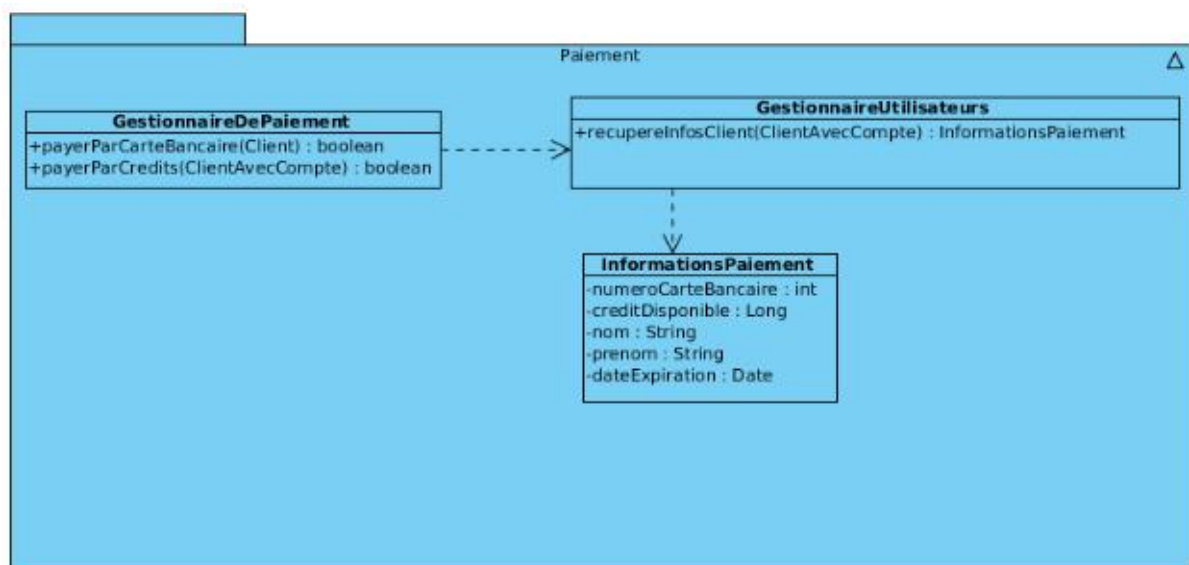


Figure 15. Diagramme de classe Gestionnaire de Paiement



Afin d'interagir avec les objets persistants, nous utilisons des « **Table-Data Gateway** » selon les exemples détaillés ci-après :

Figure 16. Table-Data Gateway « Commande »

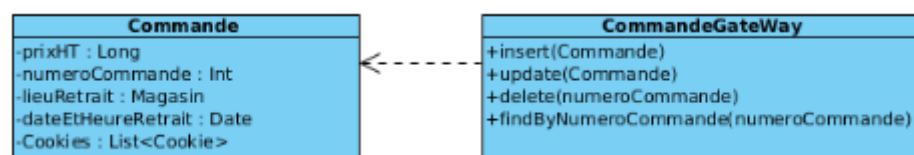


Figure 17. Table-Data Gateway « Preferences »

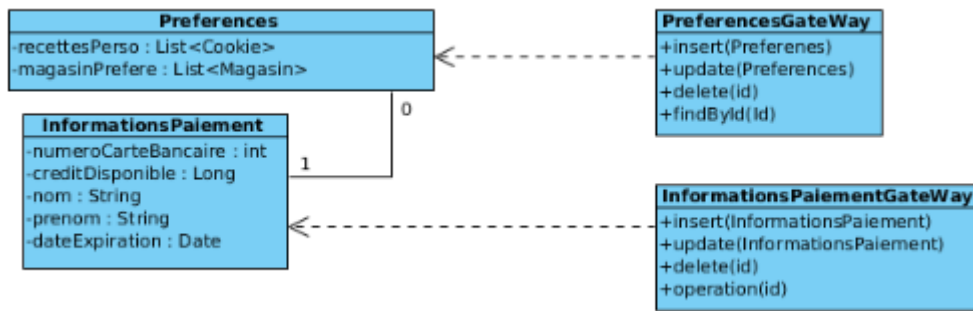
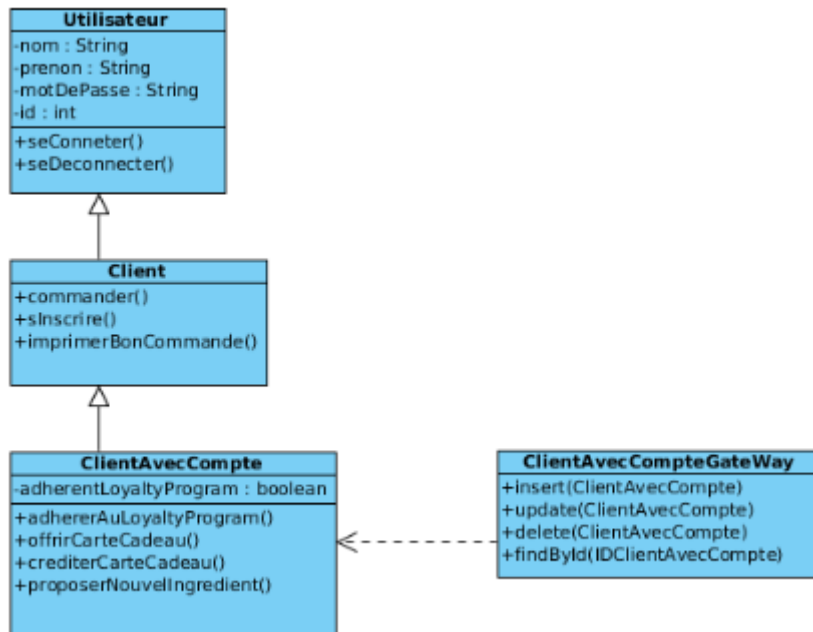


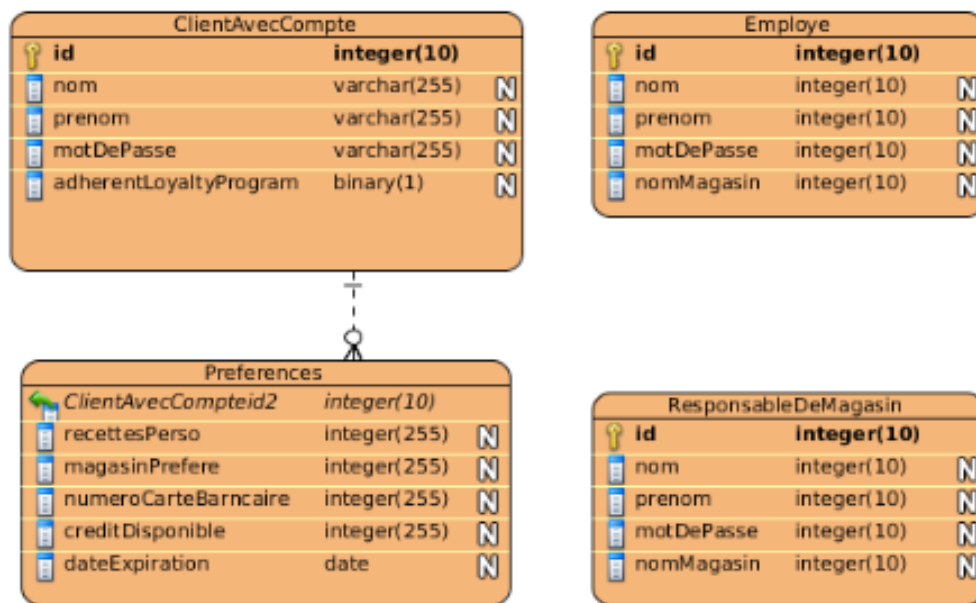
Figure 18. Table-Data Gateway « ClientAvecCompte »



Ces modèles seront à répéter pour tous les objets persistants du système CoD, c'est-à-dire « Cookie », « Ingredient », « ClientAvecCompte », « Employé », « Responsable de Magasin », « Preferences », « Magasin » et « Commande ».

Il s'agit ensuite de définir les tables dans lesquelles seront stockées les données du système CoD. Pour les données relatives aux utilisateurs, nous avons choisi le type de table « Concrete-Table », comme illustré dans la figure ci-dessous. Nous avons fait ce choix car cela permet de créer des tables indépendantes entre chaque utilisateur et évite les « goulots » dans la hiérarchie.

Figure 19. Concrete-Table Utilisateurs



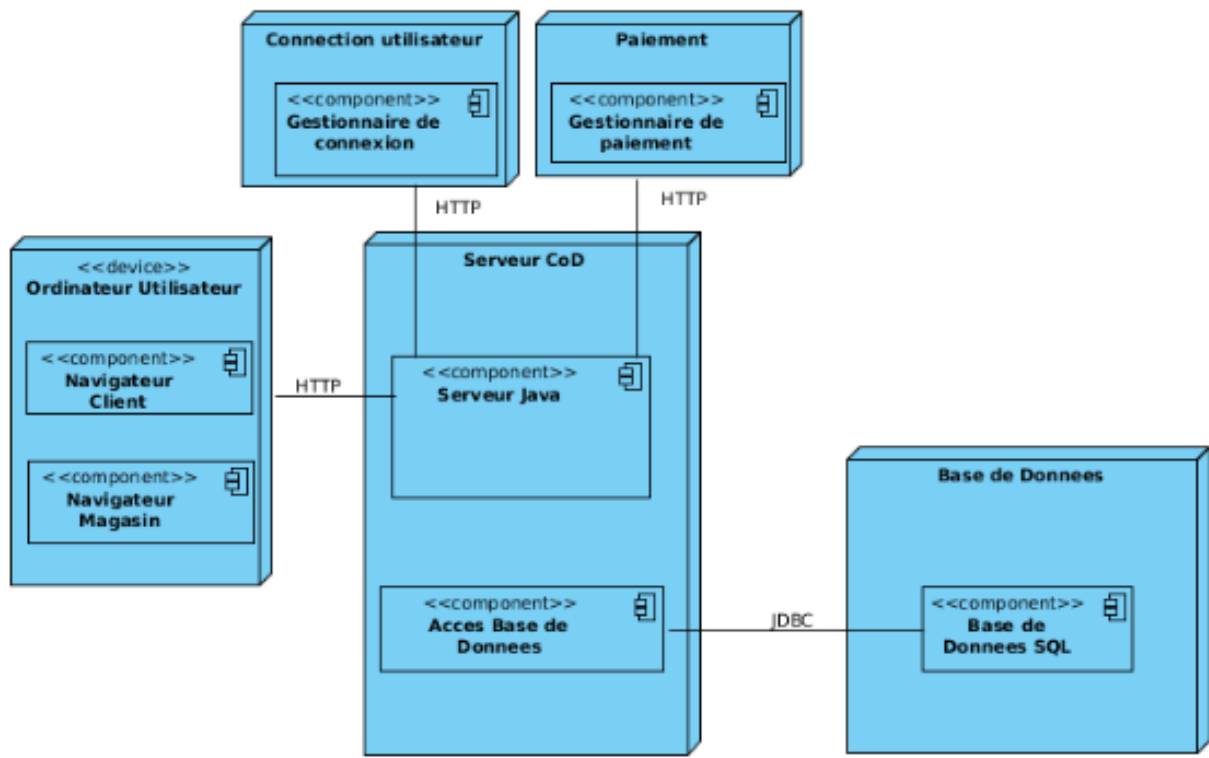
Pour les données relatives à la commande d'un client, le système CoD garde toutes les données dans différentes tables ayant des relations entre elles, comme le montre la figure 20.

Figure 20. Tables relatives à la commande



Diagramme de déploiement

Figure 21. Diagramme de déploiement



Afin de positionner le système, nous avons élaboré un diagramme de déploiement composé de 5 entités, dans lequel on retrouve certains composants définis dans le Diagramme de composants précédemment (voir figure 10).

Tout d'abord, nous avons l'**Ordinateur Utilisateur** qui regroupe les composants « Navigateur Client » et « Navigateur Magasin » et qui communique directement avec le Serveur Java du **Serveur CoD**.

L'entité **Connexion Utilisateur**, qui héberge le composant Gestionnaire de connexion, est également reliée au Serveur Java du **Serveur CoD**, de même que l'entité **Paieement**.

La **Base de Données** est reliée au **Serveur CoD** par le composant « Acces Base de Donnees ».