Groupe 13A

Démonstration de compétence

Desgranges Lucas

Dans ce projet, j'ai pris en charge plusieurs aspects clés pour construire l'application.
Tout d'abord, j'ai conçu le diagramme de séquence pour la fonctionnalité "Passer commande", ce qui m'a permis de modéliser le déroulement du processus sous un angle dynamique. Ce travail m'a aidé à mieux comprendre et préciser les interactions entre les différentes classes (Client, Magasin, Livre, etc.) et à clarifier la logique métier.
Ensuite, j'ai mis en place la structure de base de l'application en créant tous les fichiers Java nécessaires, avec leurs attributs et signatures de méthodes. Cette étape a servi de fondation pour le développement à venir et a aussi facilité la répartition des tâches au sein de l'équipe.
J'ai réalisé les diagrammes de séquence et j'ai ensuite développé la classe Client, Vendeur et une partie de Administrateur. J'y ai intégré plusieurs méthodes importantes :

- passerCommande(...) : pour enregistrer une commande dans la base de données, mettre à jour les stocks et générer automatiquement un numéro de commande.
- onVousRecommande() : conçue pour suggérer des titres à un client en comparant ses achats avec ceux d'autres clients aux goûts similaires.
- consulterCatalogue() : qui permet au client de voir les livres disponibles.
- ajouterLivre() : qui permet au vendeur d'ajouter un livre dans son magasin.
- livreDisponible() : qui permet au vendeur de voir si un livre est disponible dans son magasin.
- majQuantiteLivre() : qui permet au vendeur de mettre à jour la quantité d'un livre dans son magasin.
- transfertLivre() : qui permet au vendeur de transférer un livre de son magasin vers un autre.
- passerCommandePourClient() : qui permet au vendeur de passer une commande pour un client.
- creerLivre() : qui permet à l'administrateur d'ajouter un livre à la base de données.
- ajouterLibrairie() : qui permet à l'administrateur d'ajouter un magasin à la base de données.
- transfertLivreEntreMagasin() : qui permet à l'administrateur de transférer un livre de d'un magasin vers un autre.

Grâce à ce travail, j'ai appris à traduire un besoin fonctionnel en un code Java structuré, en respectant les principes de la programmation orientée objet. J'ai notamment représenté chaque entité (client, auteur, livre, commande) par une classe avec ses responsabilités propres.
La gestion de la commande m'a demandé d'adopter une démarche logique et organisée, en enchaînant plusieurs opérations : génération automatique d'un identifiant, insertion de données, mise à jour des stocks, récupération des prix, etc.
Cela m'a aussi donné l'occasion de m'exercer à la gestion des interactions avec une base de données SQL, en préparant et exécutant des requêtes avec PreparedStatement tout en veillant à bien libérer les ressources (comme les ResultSet et Statement).

Par ailleurs, j’ai pris l’habitude de tester régulièrement chaque fonctionnalité développée : je passais une commande, puis je vérifiais directement dans la base si les données avaient été correctement ajoutées ou modifiées.
Cette approche m’a permis de détecter et corriger des erreurs, comme des oublis de fermeture de requêtes ou des conditions SQL mal formulées.
Enfin, en structurant soigneusement mon code avec des noms explicites, des responsabilités bien définies et des méthodes courtes j’ai facilité le travail collaboratif. Mes collègues pouvaient ainsi comprendre, compléter ou utiliser mes classes sans difficulté.