Groupe 12

Etienne Audor

Gauthier Boissel(chef de groupe)

Naick Cher

Hameau Alexis

Rapport SAE JAVA

Université d'ORLÉANS

IUT ORLÉANS INFORMATIQUE

Sommaire

1 Introduction

Cette SAE a pour but de réaliser un projet en groupe. La thématique repose sur le développement d'un logiciel destiné à être utilisé par plusieurs entreprises vendant des livres en France.

En suivant attentivement la demande du client, nous procéderons à la réalisation complète du projet, de la traduction de la demande en étapes distinctes à la création d'un environnement de travail.

L'application est à destination de 3 types de personnes, tout d'abord un client classique (interface utilisateur classique avec les descriptions sur les différents articles ainsi qu'un aiguillage clair et efficace), un client professionnel (même description que le client mais avec des prix adaptés pour les entreprises, c'est-à-dire avec moins de taxes et plus d'options de livraison), un vendeur(doit pouvoir gérer les stocks du magasin pour les approvisionner).

Et ensuite un administrateur (doit être capable de régler les problèmes de gestion de l'employeur mais aussi des clients, il peut aussi accéder aux statistiques de vente de chaque magasin).

La réalisation de ce projet est expliquée dans plusieurs chapitres de ce rapport.

Le premier chapitre présente la partie conception de la base de données. Nous expliquons nos choix concernant le MCD et le MLD, puis les insertions effectuées dans la base, ainsi que les principales requêtes, comme l'affichage des ISBN des livres en stock.

Le deuxième chapitre concerne la conception UML, avec notamment le diagramme de classes et le diagramme de séquence. Nous expliquons pourquoi nous avons choisi de faire ces diagrammes de cette manière afin de mieux structurer notre programmation, avec les différentes classes et méthodes à implémenter.

Le troisième chapitre est consacré à l'élaboration de l'IHM, avec des maquettes, le diagramme de cas d'utilisation et les choix ergonomiques nécessaires pour garantir une navigation fluide.

Le quatrième et cinquième chapitre concernent l'implémentation du diagramme de classes et du diagramme de cas d'utilisation, qui servent à intégrer nos classes et méthodes en Java. Nous y aborderons aussi les outils de gestion de projet et GitHub, utilisés pour mieux répartir les tâches et nous organiser tout au long de la SAE.

Enfin, le sixième chapitre est dédié au bilan du projet. Nous y analysons notre organisation, nos difficultés, et ce que nous n'avons pas réussi à accomplir durant cette SAE.

2 Analyse de la base de données

L'intérêt de faire une analyse d'une base de donnée est de visualiser comment une base de donnée va interagir et stocker des données avec des données insérer par des utilisé via une application ou autre.

A) Explication du MCD/MLD

Un MCD (Modèle Conceptuel de Données) est une représentation graphique et logique des données d'un système d'information. Il est utilisé en modélisation des bases de données pour organiser et structurer les données surtout les attributs, entités et associations avant leur implémentation physique

Le MCD de cette SAÉ se penche sur le MCD de la SAE base de données. Le MCD sert à connaître, pour un client, quel livre commander avec ses auteurs, éditeurs et sa classification.

Tout d'abord, l'entité **Livre** sert à obtenir toutes les informations principales sur le livre, comme le titre, l'ISBN (qui est la clé primaire), le prix, le nombre de pages et enfin la date de publication.

Ensuite, il y a des associations entre les entités **Écrire**, **Éditer** et **Thèmes**, qui ont elles-mêmes des associations entre les entités **Auteur**, **Classification** et **Éditeur**, permettant de faire le lien entre ces entités et le livre. Un livre peut contenir 0 ou plusieurs auteurs, 0 ou plusieurs éditeurs et 0 ou plusieurs classifications, et inversement. Ces entités, comme **Écrire**, permettent de relier les clés primaires d' Auteur et de **Livre**, ce qui est utile pour les requêtes entre entités.

Par la suite, l'entité **Commande** contient des attributs comme la date de la commande et la date de livraison, tandis que l'entité **Magasin** contient le nom du magasin et le village où il se trouve. Ces deux entités sont liées. Un magasin peut connaître 0 ou plusieurs commandes, mais une commande ne peut être associée qu'à un seul magasin. Par conséquent, il n'est pas nécessaire de créer une entité faisant le lien entre **Commande** et **Magasin**. Il en est de même pour l'entité **Commande** et l'entité **Client**. On peut observer qu'entre les entités **Commande**, **DetailCommande** et **Livre**, il existe des

dépendances fonctionnelles qui servent à définir les clés primaires et, surtout, les clés étrangères des entités **DetailCommande**, comme on peut le voir avec les associations 1, 1 ou 1, N pour identifier les entités faibles.

Enfin, on trouve les entités **Livre**, **Magasin** et **Posséder**, qui permettent de connaître la quantité de livres présents dans chaque magasin. L'association entre ces entités suit un modèle 0, N, avec des clés primaires qui permettent de relier ces entités.

Ce MLD comporte presque tous les éléments du MCD, comme les clés primaires pour chaque entité. Toutefois, certaines entités ne figurent pas dans le MLD, comme les entités **Effectuer** et **Concerner**, qui ne sont pas utiles en SQL et ne sont donc pas présentes dans le MLD ci-dessous. Il y a quelques différences dans le MLD, notamment pour les entités mises en ligne avec les clés primaires en **gras**, les attributs en *italiques* et les clés étrangères marquées d'un #. Cela permet de visualiser plus simplement les différentes clés ou attributs dans les entités.

Client (**idcli**, *nomcli*, *prenomcli*, *adressecli*, *codepostal*, *villecli*)
Commande (**numcom**, *datecom*, *enligne*, *livraison*, #idmag, #idcli)
Magasin (**idmag**, *nommag*, *villemag*)
Posseder (**#isbn, #idmag**, qte)
DetailCommande (**numlig**, *qte*, *prixvente*, #isbn, #numcom)
Livre (**isbn**, *titre*, *nbpages*, *datepubli*, *prix*)
Auteur (**idauteur**, *nomauteur*, *anneenais*, *anneedeces*)
Ecrire (**#idauteur, #isbn**)
Editer (**#idedit, #isbn**)
Editeur (**idedit**, *nomedit*)
Themes (**#iddewey, #isbn**)
Classification (**iddewey**, *nomclass*)

Figure 1 : MLD Librairie

B) Choix des insertions

Dans la figure 2 on a fait ces choix d'insertions parce que on a pris les entitées les plus importantes pour en faire des insertions car l'entité Client n'est pas très utile dans ce cas car on n'a pas besoin de connaître le prénoms des client, ou aussi l'entité commande puisqu'on n'aura pas l'utilité de d'avoir la

date de la commande ou le numéro de la commande pareil avec l'entité detailcommande. Au final on a choisi ces entités là car cela impacte si on a bien le programme qui était demandé et qu'il soit correct.

```
INSERT INTO LIVRE(isbn, titre,nbpages,datepubli,prix) VALUES
    ('9782205054750', 'SQL pour les Nuls', 292, 2002, 33.5);

INSERT INTO EDITEUR(nomedit,idedit) VALUES
    ('First Interactive', 240);

INSERT INTO EDITER(isbn,idedit) VALUES
    ('9782205054750', 240);

INSERT INTO AUTEUR(idauteur, nomauteur,anneenais,anneedeces) VALUES
    ('OL246259A', 'Allen G. Taylor', NULL, NULL),
    ('OL7670824A', 'Reinhard Engel', NULL, NULL);

INSERT INTO ECRIRE(isbn,idauteur) VALUES
    ('9782205054750', 'OL246259A'),
    ('9782205054750', 'OL7670824A');

INSERT INTO POSSEDER(idmag, isbn, qte) VALUES
    (7, '9782205054750', 3);
```

Figure 2 : Insertion données

C) Explication des principales requêtes

On voit la figure 3 que nous avons choisi d'expliquer la requête palmarès qui consiste à connaître l'auteur qui a vendu le plus de livres dans ses librairies chaque année (sans prendre 2025 qui ne fait que commencer). on peut voir ci -dessous que nous avons fait un vue temporaire avec le WITH AS puis dans le select on à faire un YEAR pour avoir l'année de l'attribut datecom et un IFNULL dès quantité et si ces quantité la sont vide on mets 0 puis on a fait des jointure naturel qui correspond au MCD puis la condition que si l'anne est égal à 2025 on prendre pas les données qui correspond a cette année
Ensuite on voit dans la figure 4 la deuxième requêtes.Cela consiste à faire la facture de tous les clients qui ont pris des livres en fonction du mois et de l'année ici c'est fevrier 2020.On voit ici que on a refait une vue temporaire

avec tous les des entités Livres et commande puis on a fait les jointure naturel qui sont nécessaire et enfin une condition sur le mois de février 2020

```
WITH exemplairesVendusParAnnee AS(
    SELECT YEAR(datecom) annee, idauteur, nomauteur,
    IFNULL(SUM(qte), 0) exemplairesVendus FROM AUTEUR
    NATURAL LEFT JOIN ECRIRE
    NATURAL JOIN LIVRE
    NATURAL JOIN DETAILCOMMANDE
    NATURAL JOIN COMMANDE
    WHERE YEAR(datecom) != 2025
    GROUP BY annee, idauteur
    ORDER BY exemplairesVendus DESC
)

SELECT annee, nomauteur, MAX(exemplairesVendus) total FROM exemplairesVendusParAnnee
GROUP BY annee;
```

Figure 3 : requête Palmarès

```
WITH LivresVendus AS (
    SELECT numcom, isbn ISBN, titre Titre, qte, prixvente prix FROM DETAILCOMMANDE
    NATURAL LEFT JOIN LIVRE
    ORDER BY numcom
)
SELECT nomcli, prenomcli, adressecli, codepostal, villecli,
numcom, datecom, nommag, ISBN, Titre, qte, prix FROM LivresVendus
NATURAL LEFT JOIN COMMANDE
NATURAL JOIN CLIENT
NATURAL JOIN MAGASIN
WHERE MONTH(datecom) = 2 AND YEAR(datecom) = 2020
ORDER BY nommag, numcom, ISBN;
```

Figure 4 : requête Facture

D)Modification du MCD

Dans la figure 5 on peut voir qu'on a modifié le MCD parce qu' on a décidé dans notre UML d'ajouter des classes VENDEUR et ADMINISTRATEUR donc on a rajouté ces entités dans notre MCD. On a décidé d'ajouter ces entitées pour faciliter la tâches quand on fait le code et d'avoir plus de possibilités.

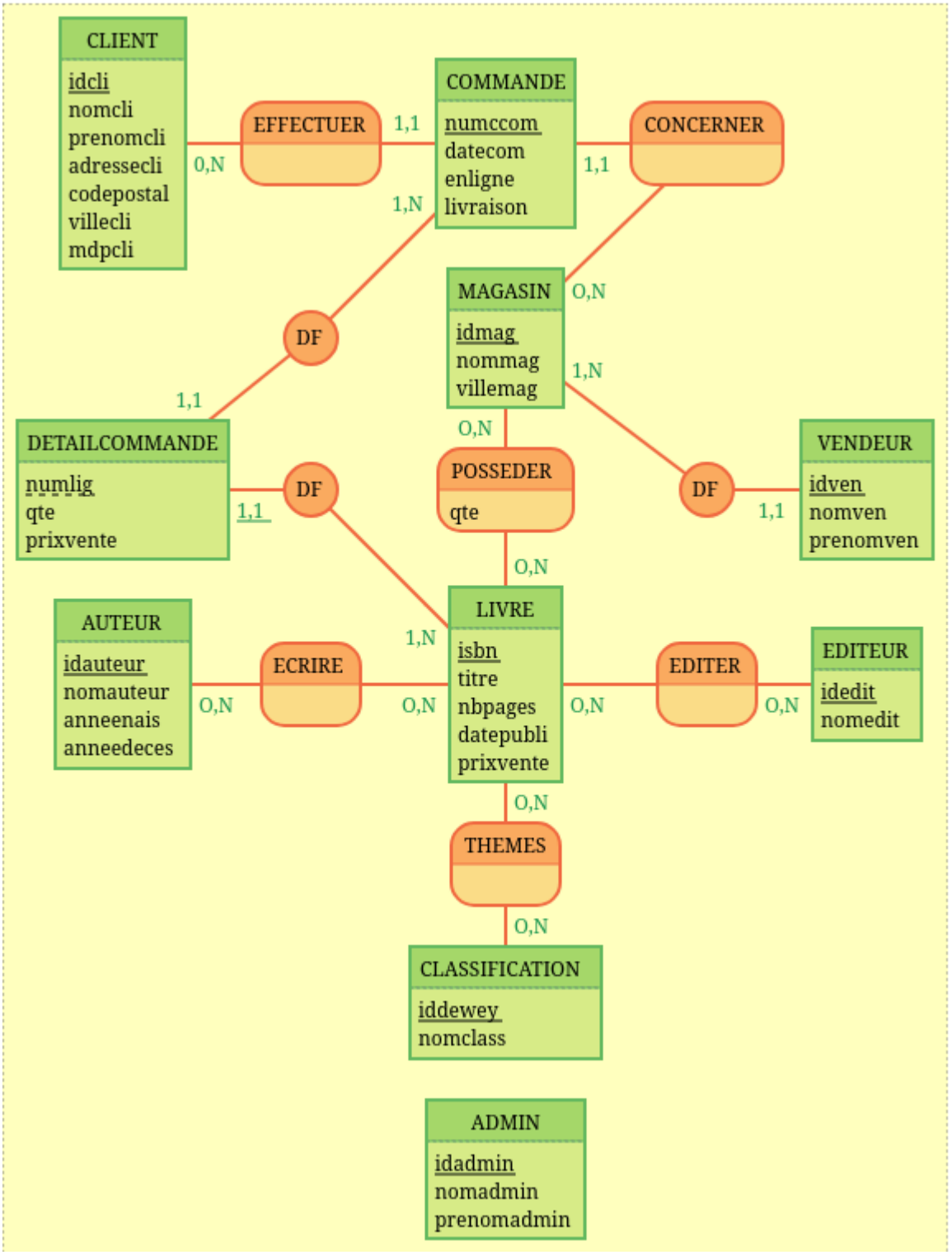


Figure 5 : MCD modifié

3 Analyse UML

A) Introduction de l'UML

Un UML sert à présenter la structure du code et connaître les attributs et méthodes qu'il faut implémenter pour plus facilement s'y retrouver au moment de coder et ne pas perdre trop de temps à chercher comment les implémenter. L'UML se traduit dans notre cas sous deux formes : Le diagramme de classes, montrant les principales classes que l'on va utiliser, avec leurs attributs, leurs méthodes et les liens entre les différentes classes, et le diagramme de séquence qui pour une méthode d'une classe donnée indique quelles autres méthodes elle utilise.

B) Présentation du diagramme de classes

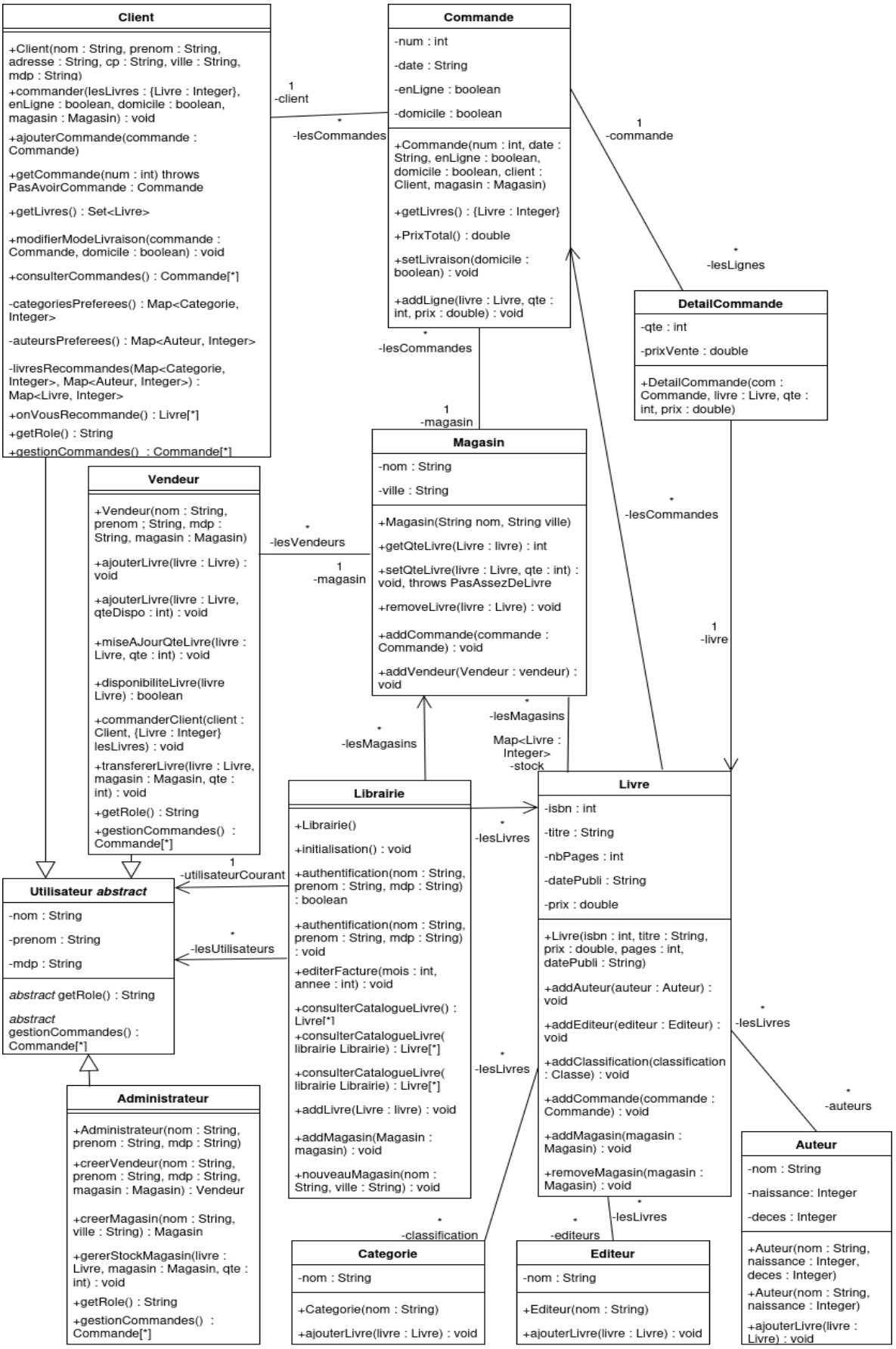


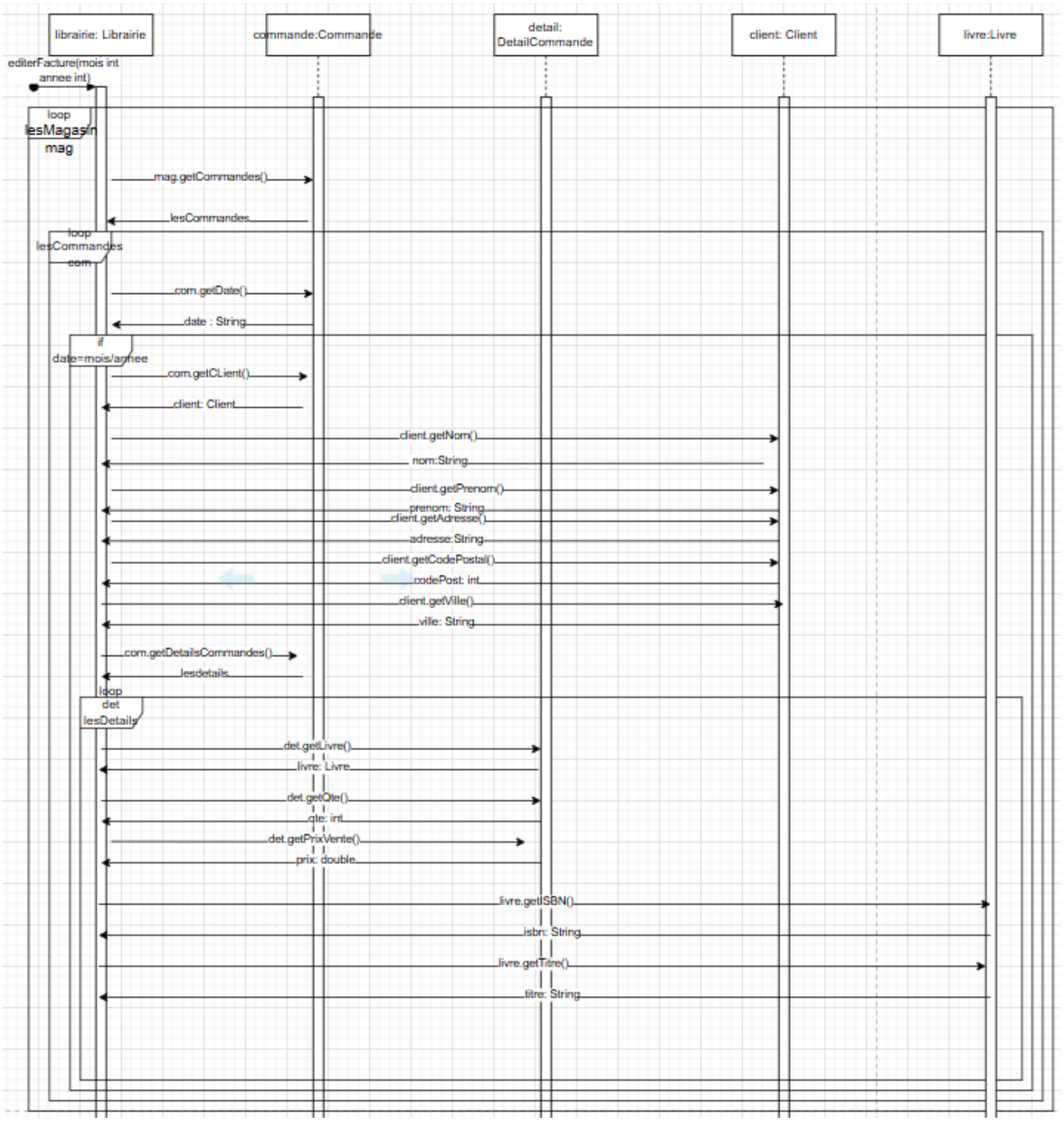
Figure 6 : Diagramme de classes

Pour concevoir notre diagramme de classes nous nous sommes basés sur la base de données avec le MCD modifié (voir *Figure 5*), plus particulièrement les dépendances.

Les dépendances dans le diagramme de classe sont définies comme suit :
-Si la dépendance d'une table dans la BD est de (0, N) ou (1, N), la classe représentant cette table aura comme attribut une liste comportant les instances de la classe représentant la table associée à la première table.
-Si la dépendance d'une table dans la BD est (1, 1), la classe représentant cette table aura comme attribut une instance de la classe représentant la table associée à la première table.
La seule exception est l'association entre Magasin et Livre, où la classe Magasin n'aura pas une liste de livres mais un dictionnaire ayant pour clés des livres et pour valeur leur stock en magasin en nombres entiers.
Nous avons aussi choisi de rendre les classes Vendeur, Administrateur et Client comme étant des enfants d'une classe abstraite Utilisateur, car les 3 classes ont des attributs proches/en communs, en plus nous souhaitons que chacun des 3 types d'utilisateurs puissent se connecter, ce qui a été rendu plus simple en liant ces 3 classes à une classe abstraite.

C) Présentation du diagramme de séquence

Diagramme sequence fonction editerFacture();



Explication:

Ce diagramme représente l'interaction et l'appel des différentes fonctions dans les différentes bases.

Cette fonction est de type void, donc elle ne renvoie rien. Elle crée ou modifie seulement un fichier .txt contenant les factures de chaque magasin pour un mois et une année donnés (il s'agit de la même fonction que dans la SAE base de données).

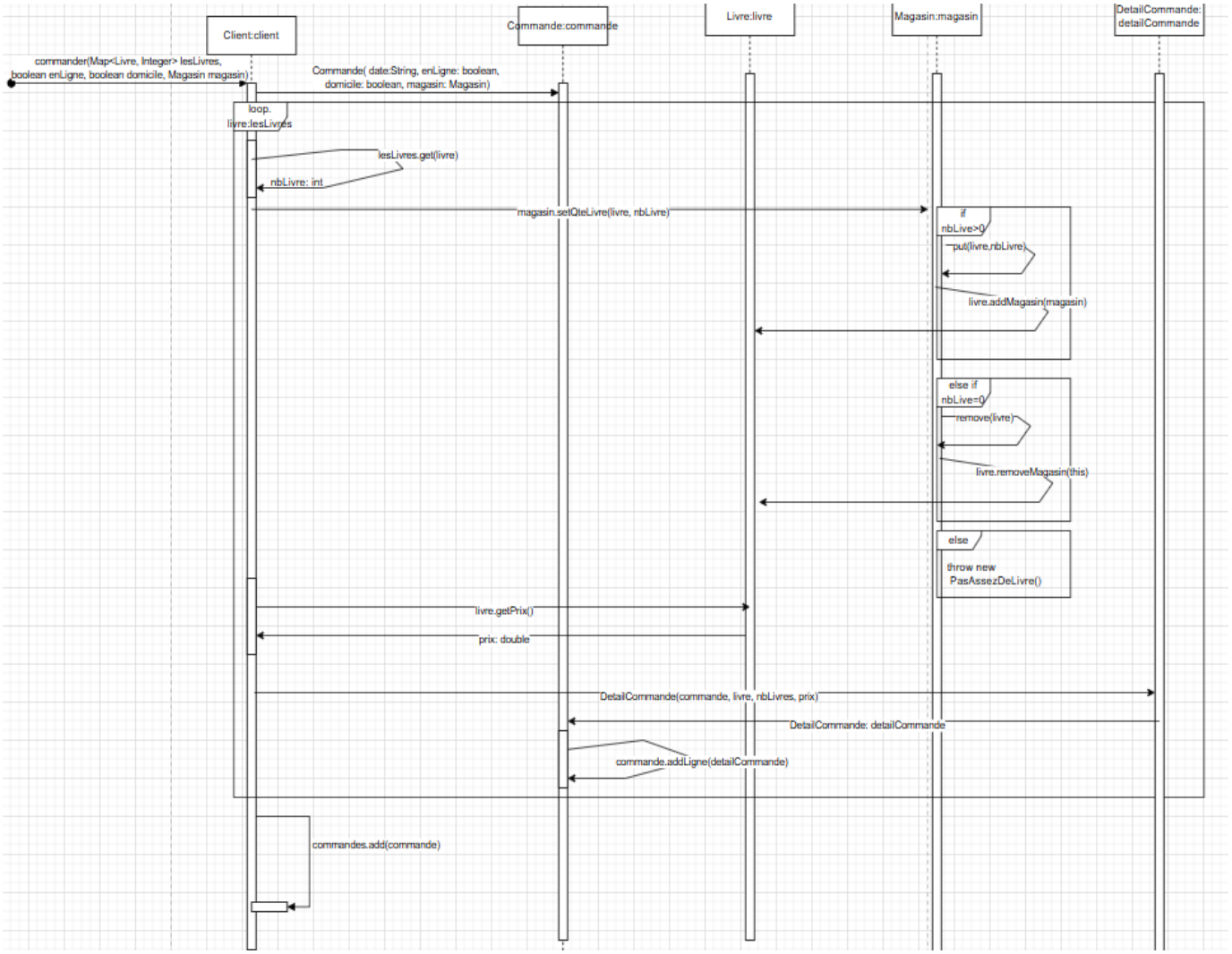
Pour ce qui est du fonctionnement, je récupère l'ensemble des magasins, et pour chaque magasin, l'ensemble des commandes réalisées. J'effectue une vérification sur la date : en cas d'égalité avec l'entrée (mois et année), je parcours les detailsCommandes pour consulter le contenu, avec le nom du livre, la quantité, le prix, etc.

Au fur et à mesure de l'exécution de la fonction, j'incrémente une variable de type String avec les informations récupérées. Une fois terminé, je place cette variable dans un fichier .txt.

Le fichier devrait ressembler plus ou moins à : (capture du sujet de la SAE BD)

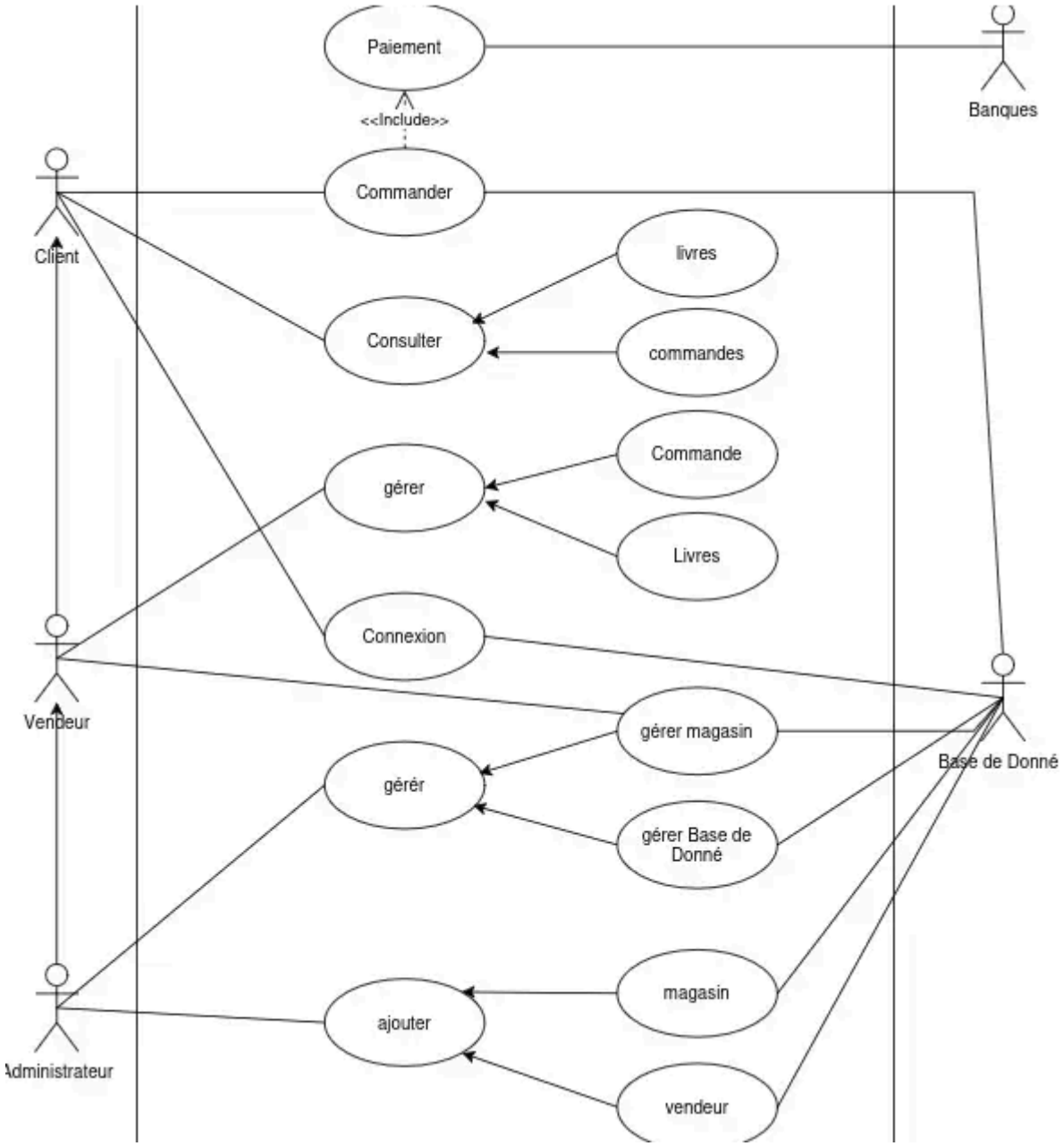
```
Factures du 2/2020
Edition des factures du magasin La librairie parisienne
--------------------------------------------------------------------------------
Ben Ali Francesca
115 allée du Musée
51100 Reims
                        commande n°51 du 11/02/2020
        ISBN                          Titre                   qte   prix    total
  1 9782020339032 Leçons de philosophie                       1  16.01    16.01
  2 9782012383869 Beurres aromatisés                          1  39.90    39.90
  3 9782747034692 Le secret du Lorrain                        1  15.00    15.00
                                                                       --------
                                                                Total    70.91
--------------------------------------------------------------------------------
Durand Louis
196 quai de la Colline
06000 Nice
                        commande n°60 du 20/02/2020
        ISBN                          Titre                   qte   prix    total
  1 9782020069625 Fenêtre jaune cadmium, ou, Les dessous de   2  32.50    65.00
                                                                       --------
```

Diagramme sequence fonction editerFacture():



Explication:
Ce diagramme représente le fonctionnement de la fonction commander de la classe Client, plus précisément lorsqu'un client décide de commander un livre. Tout d'abord, le client choisit son panier, que je vais modéliser par un dictionnaire avec le livre en clé et la quantité en valeur. Je fais ensuite un parcours par clé pour vérifier la quantité de livres restante dans le magasin : soit il y en a assez, et je mets à jour la nouvelle valeur du stock, soit il en manque, et je renvoie une exception indiquant que la commande est impossible.
Une fois terminé, j'ajoute un detailCommande avec le livre, la quantité et le prix. Pour finir, je stocke la commande dans la classe Client.

Diagramme cas d’utilisation:



Explication :
Pour chaque type d’utilisateur :

Client :
Dans notre application, le client peut soit commander des livres en créant son panier, auquel cas il peut accéder à un espace de paiement avec un lien vers sa banque. Il peut aussi consulter les livres ainsi que toutes ses commandes. Et bien sûr, il peut accéder à un espace de connexion avec un champ pour renseigner un identifiant et un mot de passe propre à chaque utilisateur.

Vendeur :
Tout d’abord, le vendeur a accès aux mêmes paramètres que ceux du client. Il dispose cependant de fonctionnalités propres à son rôle, c’est-à-dire qu’il peut gérer les commandes ou les livres (l’existence d’un livre ou non). Il peut également gérer les magasins, c’est-à-dire les stocks de livres. Par conséquent, il peut réapprovisionner son magasin en fonction des stocks restants.

Administrateur :
Ce rôle a accès à l’intégralité des fonctionnalités des utilisateurs définis ci-dessus. Il a cependant un rôle clé dans l’administration de l’application. En particulier, il a un accès direct à la base de données, c’est-à-dire qu’il peut la gérer et effectuer toutes les requêtes (INSERT, CREATE, UPDATE, …). Par conséquent, il peut ajouter des magasins mais aussi associer de nouveaux vendeurs à un magasin.

4 Implémentation : explication de la démarche

A) Présentez les maquettes et les choix ergonomiques

L'objectif principal de l'interface hommes machines est de simplifier au maximum l'utilisation de l'application sans pour autant limiter les actions de l'utilisateur. Nous avons donc mis en place un menu simple et accessible.
On peut voir dans la figure 7 une interface classique qui va permettre au client de ne pas se perdre dans l'application, en gros on voit le texte de la librairie puis à côté un bouton connecter ou se créer un compte
puis à gauche on a un barre de recherche pour des chercher des livres ou auteur

Figure 7 : maquette de l'application

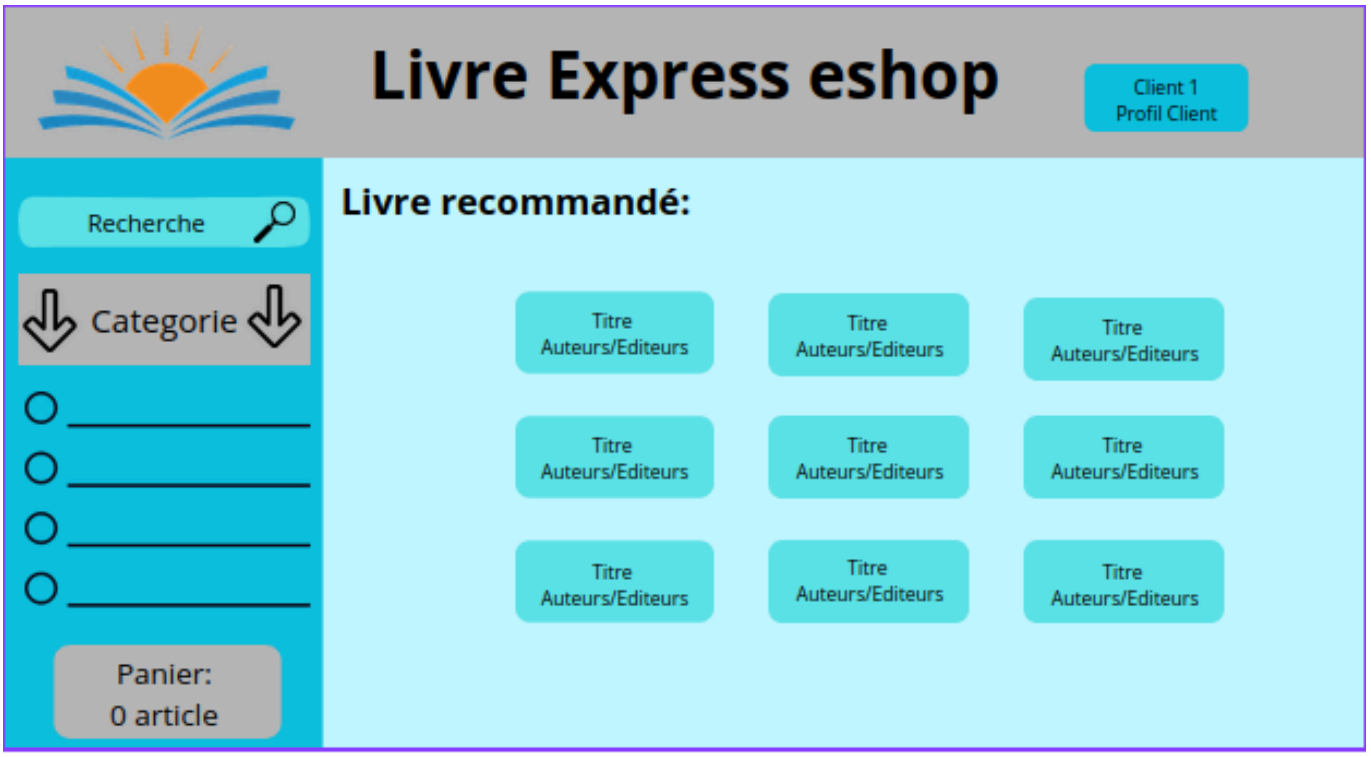


Figure 8 : maquette client

5 Elaboration des Interfaces Homme Machine

A) Expliquez la démarche (développement du back-end : traitements, calculs)

Pour implémenter notre projet, nous avons créé plusieurs parties, afin de fractionner le travail. Tout d'abord, la création d'une branche "utilisateur" sur notre repository nous a permis de gérer la façade utilisateur avec les clients et les administrateurs. Une autre branche a été créée pour implémenter le côté "Livre" , avec le détail des livres mais aussi des magasins, et sans oublier les commandes effectuées. Avec ce choix d'implémentation, nous pouvons travailler de notre côté sans gêner les autres membres du groupe et par conséquent travailler efficacement. Grâce à cela, nous pouvons merger les différentes branches vers le "main" pour former une nouvelle version complète et fonctionnelle de notre projet. Et cela dans le but d'avoir une version 1, 2 … jusqu'à la version finale. Ce qui nous permet de corriger directement les bugs et les possibles conflits entre les classes.

6 Organisation du travail de groupe

A) Comment vous êtes-vous organisés tout au long de cette SAE ?

Pour réaliser l'application, nous avons principalement travaillé en même temps, afin d'avoir une meilleure communication. Cela nous permet à chaque début de séance de faire le point sur l'avancement du projet mais aussi de répartir ou de continuer les tâches.Puis on se repartie les tâches comme faire le diagramme de classe, continuer le rapport ou implémenter les méthodes en java.

B) Présentez vos outils de gestion de projet (Diagramme de Gantt, matrice RACI)

Pour ce qui est de la communication, nous avons créé en parallèle un serveur sur Discord, avec un canal vocal et textuel. En cas de retard ou de manque d'information, ou même pour donner de l'aide, les autres membres du groupe peuvent donc intervenir avec presque les mêmes conditions que dans les cours dédiés aux SAE.

C) Présentez GitHub et son utilité et expliquez la manière dont vous l’avez utilisé

Pour ce qui est du support de travail, nous avons construit un dépôt sur GitHub. Nous l'avons par la suite paramétré ce même dépôt pour préparer notre futur travail. Plus précisément, nous avons construit les différentes branches avec les dossiers communs mais aussi un fichier.gitignore pour faciliter les commits.Puis on a fait des branches pour chaque partie importantes de la SAE comme sur la figure 7, cela nous ai très utile pour pas travailler sur la même branche en même temps.

Figure 7 : les Branches

7 Bilan du groupe

A) Qu'est-ce qui fonctionne, qu'est-ce qui ne fonctionne pas, et pourquoi ?

Pour résumer l'avancée de nos travaux, nous sommes plutôt fiers de l'ensemble de notre projet. C'est-à-dire de l'interaction entre l'homme, l'application et la base de données.
Dans l'ensemble, toutes les fonctions obligatoires définies dans le sujet ont été implémentées. Nous avons même pu créer des fonctions supplémentaires, telles que la fonction "onVousRecommande".
Cependant, nous avons rencontré quelques problèmes lors de la réalisation et de l'application des tests JUnit.

B) Bilan sur l'organisation du groupe et du projet

Grâce à notre choix de support (GitHub), ainsi qu'à une très bonne communication via des réseaux sociaux (Discord), nous avons eu une excellente organisation. Cela nous a permis de travailler efficacement à l'IUT comme chez nous, dans des conditions idéales.
En ce qui est de l'organisation, notre chef de groupe s'est occupé de la prise de notes sur l'avancée du projet, avec une liste de tâches à réaliser en fonction de leur priorité. Cette organisation a permis à notre projet d'avancer sans temps mort, avec toujours des tâches à effectuer.

C) Bilan sur les principaux acquis

Pour finir, cette SAE nous a donné une vraie leçon, surtout au niveau de l'organisation, car il s'agit jusqu'à maintenant du plus gros projet que nous ayons réalisé, qui plus est avec un groupe de 4 personnes.
De plus, celle-ci nous a permis de gagner en rigueur sur les conventions à utiliser (que nous appliquerons plus tard dans le milieu professionnel), mais aussi de nous familiariser avec des méthodes telles que JDBC, et enfin avec des outils/technologies comme MariaDB (SQL) et Java.