

## PROJET DE MODELISATION UML-XML

### Modélisation d'un site d'e-commerce :

Le but du projet vas être de réaliser la partie conceptuelle d'un site d'e-commerce, plus précisément de la vente de livre en ligne, afin d'y parvenir il faudra réaliser 4 diagrammes dont les suivants : diagramme de classes, diagramme Use Case, diagramme de séquence qui s'appuie sur une fonctionnalité précise du site et un diagramme de modèle relationnel.

Une fois cela fait il restera à faire un fichier XML suivi de son fichier DTD qui lui correspondra et pour finir une dizaine de requêtes XPATH.

### Diagrammes / I

#### 1) Diagramme Use case

Pour réaliser le diagramme Use Case il faut dans un premier temps déterminer les services et les acteurs qui vont avoir des rôles importants dans le bon fonctionnement du site e-commerce.

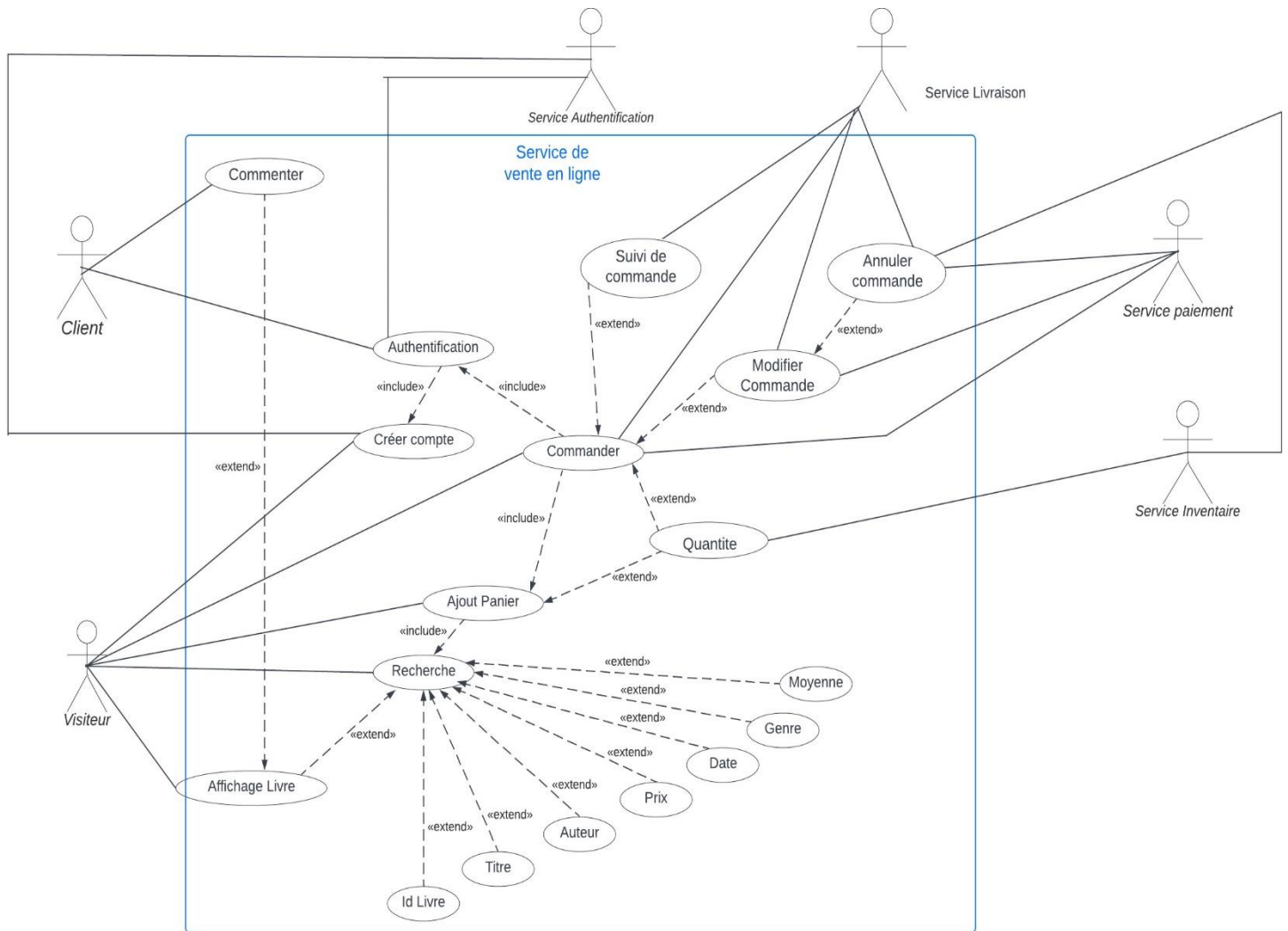
En tant que personne qui se rend pour la première fois sur le site elle sera libre de naviguer sans avoir besoin de se connecter a un compte jusqu'au moment de réaliser son paiement, c'est-à-dire que l'utilisateur nommé visiteur pourra rechercher des livres à l'aide des différentes fonctions de recherches ( Titre, Auteur, etc ), elle va pouvoir voir les informations de chaque livres présents dans le catalogue ( Prix, Titre, Commentaires, Note Moyenne, etc), elle va pouvoir sélectionner les livres de son choix et les ajouter à son panier au nombre qu'elle veut, elle pourra aussi modifier son panier et enfin viens le moment de passer la commande, à partir de ce moment la pour finaliser sa commande et passer à la transaction il sera demander au visiteur de s'authentifier ou de se créer un compte pour devenir client, c'est là qu'interviendra le service d'authentification.

Le statut de client sera donné au visiteur qui aura rempli les informations requises à la création de compte ( adresse, tel , nom , prénom).. On se servira directement de ces informations pour livrer à la bonne adresse le client. Le client pouvant désormais finaliser ses achats aura le choix entre plusieurs types de paiement , une fois le paiement effectuer le client pourra suivre sa commande et la modifier en l'annulant, le client pourra aussi désormais poster des commentaires sur les ouvrages présents sur le site.

Qui dit paiement dit service de paiement qui se chargera d'effectuer que les transactions se seront bien effectuées et qui en cas de problème peut annuler la commande ou bien si il n't en a pas , changer l'état de la commande.

Un Service de Livraison sera forcément nécessaire, celui-ci actualisera le suivi de la commande, se permettra d'annuler la commande en cas de problème en passant par la modification.

Finalement interviendra le service d'inventaire qui se chargera de vérifier l'inventaire afin d'assurer le bon fonctionnement du service de livraison et du site, le service d'inventaire se permettra de vérifier les quantités de livre qui ont été commandés et en cas de problème il aura la possibilité d'annuler une commande.



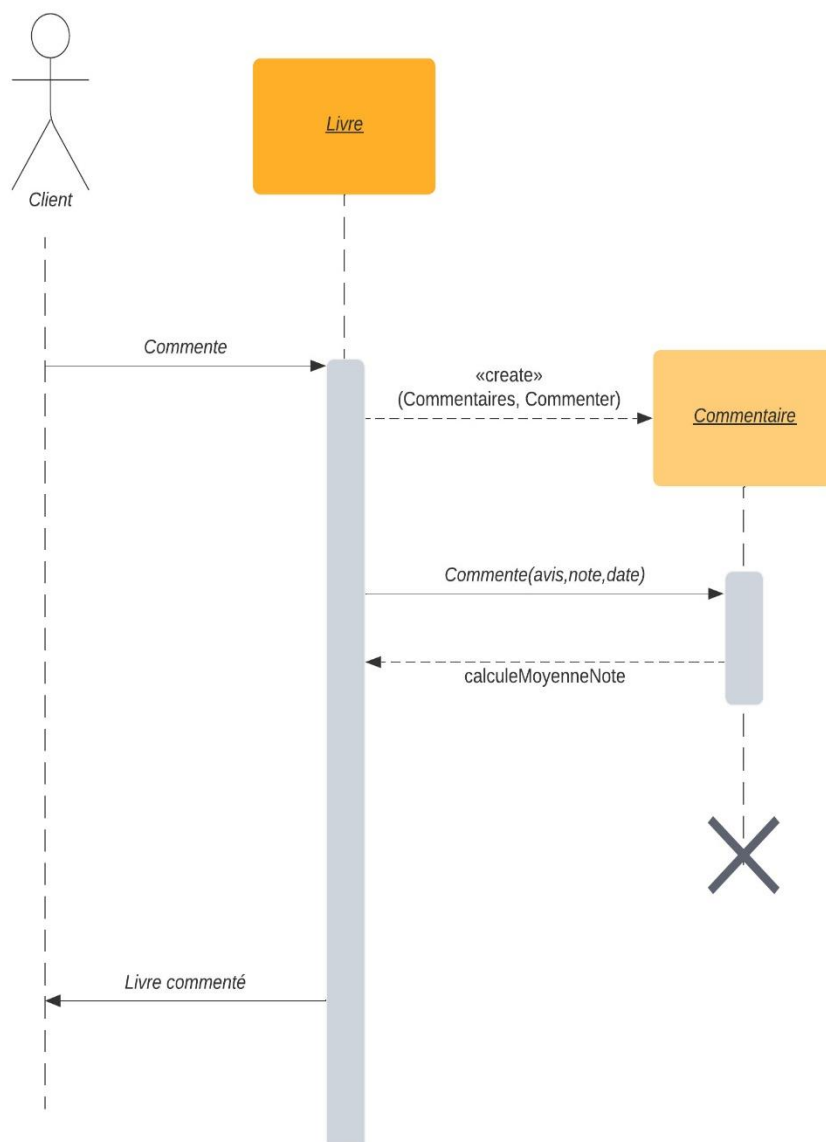
Voilà la représentation du diagramme Use Case avec les Acteurs Clients/Visiteurs, les différents services et le service de vente en ligne.

## 2) Diagramme de séquence

Pour le diagramme de séquence j'ai décidé de traiter et décrire la fonction commentaire. Sur le site le client seulement aura le droit d'écrire un commentaire sur les livres disponibles sur le catalogue, chaque commentaire requiert un avis écrit, une note sur 10 et une date qui sera prise automatiquement au moment de l'envoi du commentaire.

La fonction Commentaire se chargera de recevoir les informations envoyées par le client, et de les renvoyer vers l'œuvre commentée et ensuite recalculer la moyenne de sa note, le client ensuite aura l'affichage de son commentaire sous l'œuvre commentée.

Diagramme Sequence Commentaire



### 3) Diagramme de classe

En s'inspirant du diagramme Use Case on va établir dans un premier temps dans notre diagramme de classe la classe Livre, une classe qui aura donc les attributs nécessaires à l'affichage des informations du livre, on y retrouve : l'id du livre , le titre , l'auteur, le prix , l'image de couverture, la moyenne , le nombre de pages, la date de parution, un résumé.

Le livre possède deux méthodes, une qui se nomme `présentationLivre()` qui permet d'afficher les informations du livre et l'autre `calculeMoyenneNote()` qui va donc calculer la moyenne globale des notes donnés par les clients, le livre est lié à une classe genre avec une multiplicité de 1 qui permet d'affilier un genre au livre. Le livre est aussi lié à la classe commentaire qui permettra d'afficher les commentaires envoyés sur celui-ci ainsi que la note qui en plus de s'afficher sur le commentaire, permettra de calculer la moyenne.

Dans un second temps on crée la classe catalogue qui n'aura pas d'attributs mais 3 méthodes , la méthode `recherche()` qui utilisera la classe Recherche afin de rechercher des livres, la méthode `methodeRecherche()` qui est un type « énuméré » du même nom liée à la classe Recherche et qui sert à faire des recherches beaucoup plus précises, et pour finir la méthode `approvisionnement()` qui va permettre de réapprovisionner le catalogue en cas de rupture de stock de livres.

La classe Recherche a elle trois méthodes, les méthodes `croissant()` `décroissant()` qui comme leurs noms l'indiquent permettent de trier les recherches par ordre croissant alphabétique et décroissant anti-alphabétique, et la méthode `resultat()` qui affiche les résultats de la recherche. La classe Recherche possède une liaison avec le type « énuméré » `MethodeRecherche` qui va permettre de sélectionner l'étiquette de la recherche ( auteur , titre , etc... ), la multiplicité est donc de 1.

Recherche est aussi lié à la classe Livre cependant cette liaison n'est pas obligatoire, car chaque livre peut être trouvé sans recherches, la multiplicité sera donc de 0..\*.

Vient désormais la structure de la commande en commençant par la classe Quantite qui a juste pour attribut un Integer du nom « quantite », ce qui permettra de choisir la quantité de livres. La classe est donc liée à la classe Catalogue, à la classe Livre mais aussi à la classe Panier, classe qui aura deux méthodes dont `AjouterLivre()` et `RetirerLivre()`.

Un lien triangulaire va se former entre la classe Panier , la classe Commande et la classe Client.

La classe Client possède elle six attributs, l'id client, le nom , le prenom , l'age, l'adresse , le telephone, ces attributs permettent de recueillir les informations nécessaires à la livraison. Il possède 3 méthodes dont parmi eux l'attribut `commander()` qui permet au client de commander le contenu de son panier en faisant appel à la classe commande, la méthode

commenter() qui comment on la vu précédemment va permettre au client de laisser un commentaire sur l'œuvre de son choix on faisant appel à la classe Commentaire avec une liaison non obligatoire et donc une multiplicité de 0..\*, et enfin la méthode modifierCommande() qui va permettre au client de modifier sa commande ou l'annuler via la classe Commande avec une multiplicité de 0..\* car le client peut passer 0 ou plusieurs commandes.

La classe commande possède 2 attributs , IDCommande et dateLivraison, ainsi que trois méthodes, modifierCommande(), annulerCommande() et etatCommande() qui permet de savoir l' état de la commande en cours. La classe est liée à la classe Panier par une multiplicité de 1 , ce qui permet de sélectionner les articles pour la commande mais elle est aussi liée à un type « énuméré » Etat avec une multiplicité de 1 qui liste les différents états d'une commande : attentePaiement, preparationEnCours, livraisonEnCours, commandeLivrée, commandeAnnulée.

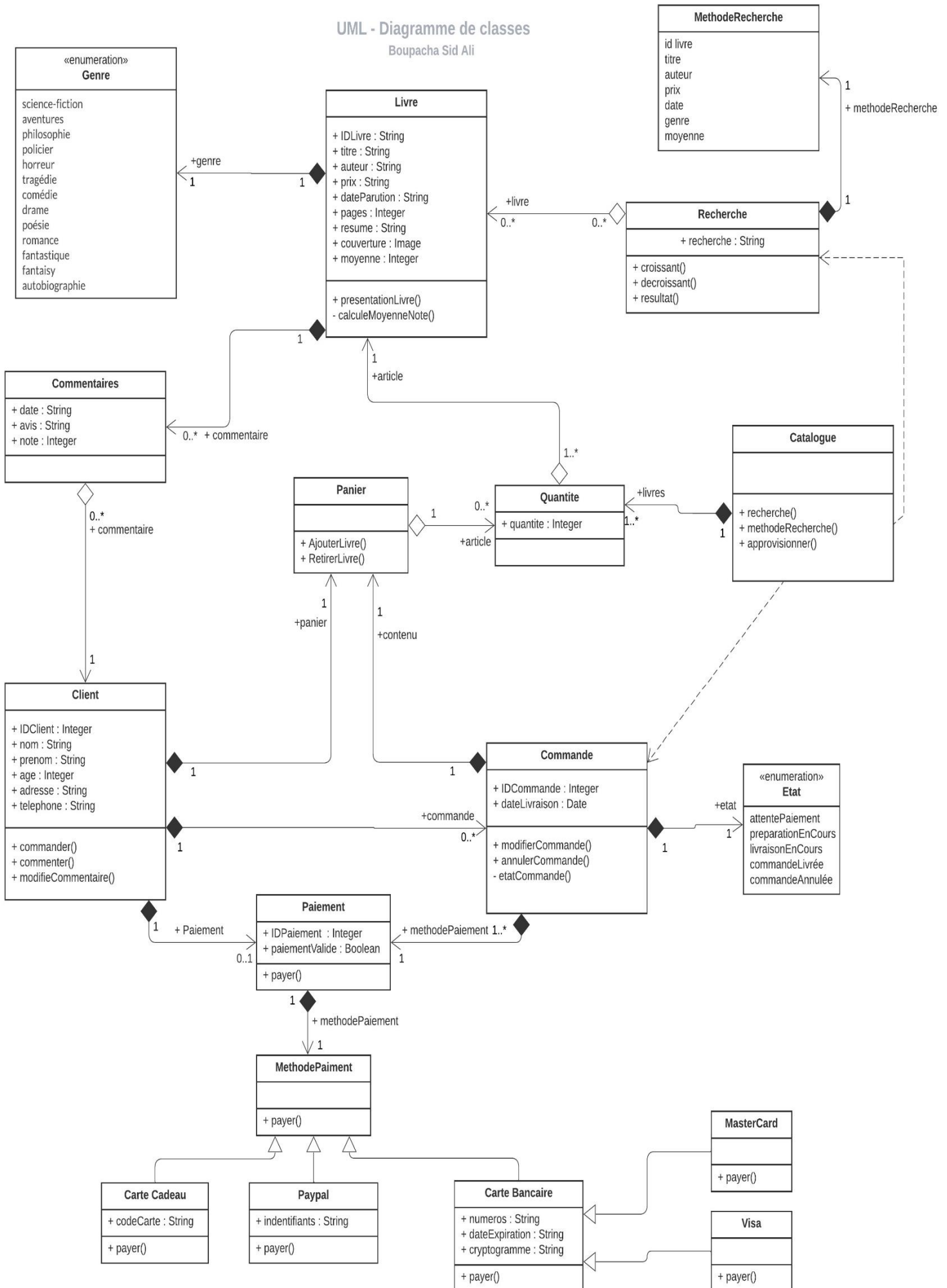
Commande et Client sont tout deux liée à la classe Paiement par une multiplicité de 0..1 avec le Client et une multiplicité de 1 avec la commande, Paiement possède IDPaiement et paiementValide qui est un boolean en tant qu'attributs, paiementValide sera true lorsque le paiement est validé et false lorsqu'il ne l'est pas. La classe à une méthode payer() qu'on va retrouver dans les classes liées ci-dessous.

Il nous faut donc une méthode de paiement et même plusieurs, on aura donc des classes différents pour chaque moyens de payement soit une pour la carte cadeau avec comme attribut le code de la carte, Paypal avec comme attribut les identifiants et une classe Carte Bancaire qui se compose en 2 classes MasterCard et Visa, Carte Bancaire possède trois attributs soit les numéros de la carte, la date d'expiration et le cryptogramme.

Le diagramme est donc désormais « complet » il manque encore des fonctionnalités à ajouter comme des réductions ou promotions , intégrer des recommandations en fonction des gens favoris du client, faire des livres avec plusieurs genre... Ceci dit la structure globale est terminée.

# UML - Diagramme de classes

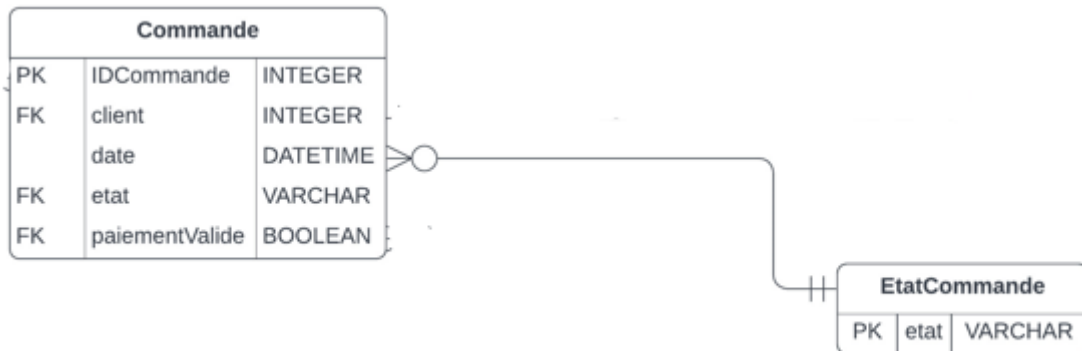
Boupacha Sid Ali



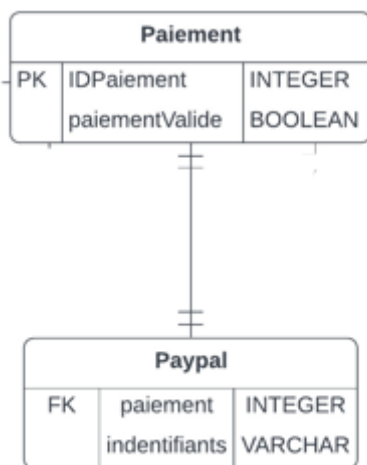
#### 4) Diagramme modèle relationnel

Enfin pour le dernier diagramme il aura fallut transformer notre modélisation du site en modèle relationnel pour pouvoir l'intégrer dans une base de donnée, le diagramme est donc réalisé dans le modèle Entité Relation , pour les clés primaire on utilise PK qui veut dire Primary Key et pour les clés étrangères on utilise FK qui veut dire Foreign Key.

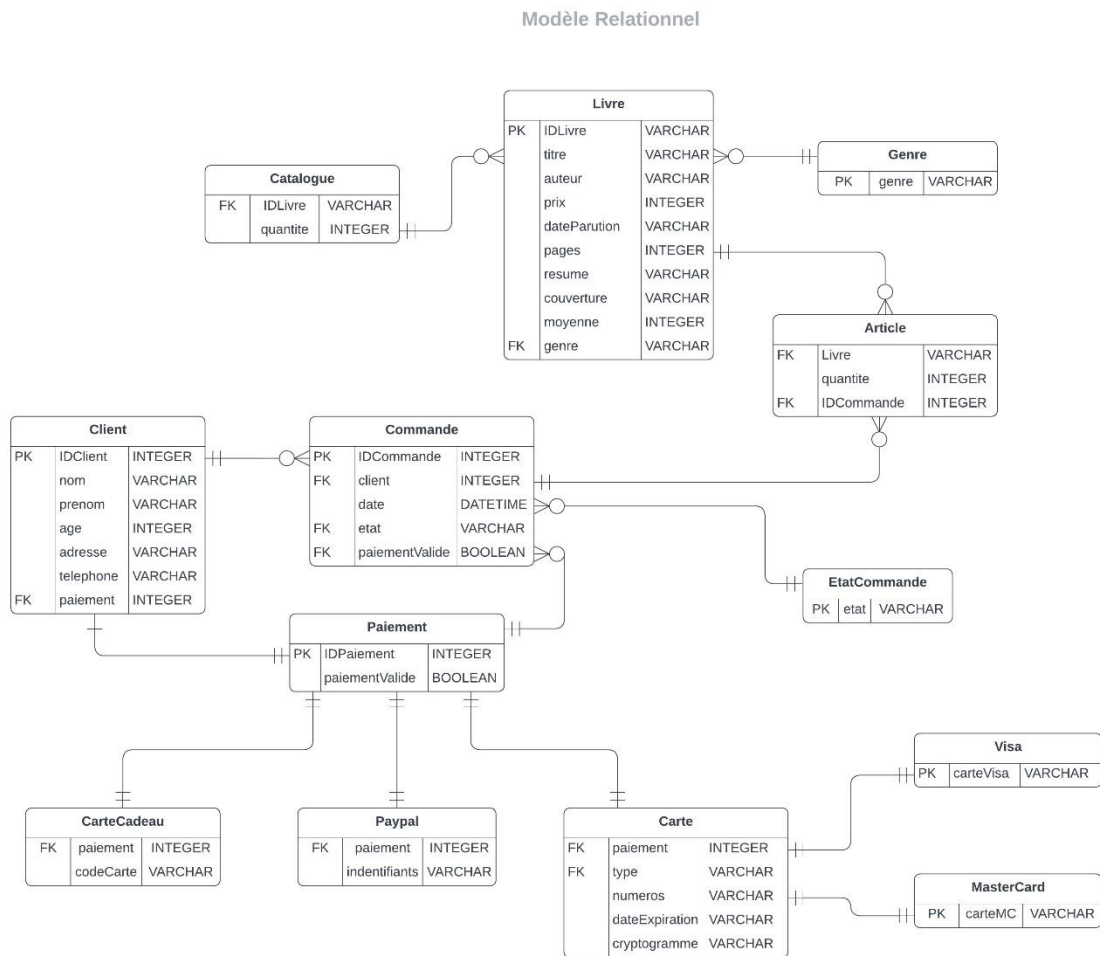
Voici une relation entre une table ( Commande ) et une entitée (EtatCommande)



Et voici une relation entre deux entitées (Paiement et Paypal).



Voilà donc le modèle Relationnel.



## XML , DTD et XPATH / II

Pour la partie XML je me suis inspiré directement de la structure d'un site de e-commerce c'est-à-dire qu'on retrouve les livre dans le catalogue , les attributs du livre dans le livre un a un et j'en ai fais des balises, les seuls attributs dans les balises seront IDLivre dans la balise livre, quantite dans la balise LivreQuantite et nom ainsi que prenom dans la balise commentaire.

J'ai pour illustré ma structure utilisé de vrai livres qui peuvent se retrouver avec cette même structure dans un site d'e-commerce.

Voici le fichier xml :



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE catalogue SYSTEM "Catalogue.dtd">
```

```
<catalogue>
```

```
  <livreQuantite quantite="4">
```

```
    <livre IDLivre="89274">
      <titre>Les Fleurs Du Mal</titre>
      <auteur>Charles Baudelaire</auteur>
      <prix>3.50</prix>
      <pages>304</pages>
      <dateParution>2019</dateParution>
      <resume>blabla blablabla</resume>
```

```
      <commentaires nom="Leblanc" prenom="Fred">
        <avis>Très bon recueil, un classique!</avis>
        <date>12/07/2020</date>
        <note>8.5</note>
      </commentaires>
```

```
      <commentaires nom="Lebleu" prenom="Alphonse">
        <avis>J'ai adoré chaque poèmes.</avis>
        <date>09/02/2021</date>
        <note>9</note>
      </commentaires>
```

```
      <couverture img="../Couverture_89274.jpeg" />
      <genre>poésie</genre>
      <moyenne>8.75</moyenne>
    </livre>
```

```
  <livre IDLivre="698304">
    <titre>La Promesse de l'aube</titre>
    <auteur>Romain Gary</auteur>
    <prix>5.00</prix>
    <pages>379</pages>
    <dateParution>1960</dateParution>
    <resume>blabla blablabla</resume>
```

```
    <commentaires nom="LeRouge" prenom="Grégoire">
      <avis>Magnifique Autobiographie, très émouvant.</avis>
      <date>18/07/2020</date>
      <note>7.5</note>
    </commentaires>
```

```
    <couverture img="../Couverture_698304.jpeg" />
    <genre>autobiographie</genre>
    <moyenne>7.5</moyenne>
  </livre>
```

```
  <livre IDLivre="42979">
    <titre>Le bonheur commence maintenant</titre>
    <auteur>Thomas Raphaël</auteur>
    <prix>5.99</prix>
    <pages>522</pages>
```

```
<dateParution>2013</dateParution>
<resume>blabla blablabla</resume>
```

```
<couverture img="../../Couverture_42979.jpeg" />
<genre>Comédie</genre>
<moyenne>?</moyenne>
</livre>
```

```
<livre IDLivre="72306">
  <titre>1991</titre>
  <auteur>Franck Thilliez</auteur>
  <prix>9.99</prix>
  <pages>504</pages>
  <dateParution>2021</dateParution>
  <resume>blabla blablabla</resume>
```

```
<commentaires nom="LeJaune" prenom="Julie">
  <avis>Le suspense et les dénouements m'ont rendue très
nerveuse, je recommande !</avis>
  <date>20/02/2022</date>
  <note>9.5</note>
</commentaires>
```

```
<couverture img="../../Couverture_72306.jpeg" />
<genre>Policier</genre>
<moyenne>9.5</moyenne>
</livre>
```

```
</livreQuantite>
</catalogue>
```

```
<!--
```

## REQUETES XPATH

Le nombre d'article :

```
/catalogue/livreQuantite/@quantite
```

Les titres de chaque livres dans le catalogue :

```
/catalogue/livreQuantite/livre/titre
```

Les ID de chaque livres :

```
/catalogue/livreQuantite/livre/@IDLivre
```

Le titre des livres avec un prix supérieur à 7 euros :

```
/catalogue/livreQuantite/livre[prix>7]/titre
```

Le titre des livres apparus avant l'an 2000 :

```
/catalogue/livreQuantite/livre[dateParution<2000]/titre
```

Le titre des livres apparus après et pendant l'an 2000 :

```
/catalogue/livreQuantite/livre[dateParution>=2000]/titre
```

La photo de couverture d'un livre :  
/catalogue/livreQuantite/livre/couverture/@img

Les ID des livres contenant plus d'un commentaire:  
/catalogue/livreQuantite/livre[count (commentaire)>1]/@IDLivre

Le titre des livres avec une note moyenne supérieure ou égale à 9 :  
/catalogue/livreQuantite/livre[moyenne>=9]/titre

Les ID des livres avec un nombre de page inférieur ou égal a 400 :  
/catalogue/livreQuantite/livre[pages<=400]/@IDLivres

-->

Comme vous pouvez le voir j'ai directement mit les 10 Requettes XPATH en commentaire dans le fichier XML.

Voici le fichier valide DTD :

```
1.      <!ELEMENT catalogue (livreQuantite*)>
2.      <!ELEMENT livreQuantite (livre*)>
3.
4.      <!ATTLIST livreQuantite quantite CDATA #REQUIRED>
5.
6.      <!ELEMENT livre (titre, auteur+, prix+, pages, dateParution, resume, commentaire*, couverture,
genre, moyenne)>
7.
8.      <!ATTLIST livre ISBN CDATA #REQUIRED>
9.
10.     <!ELEMENT titre (#PCDATA)>
11.     <!ELEMENT auteur (#PCDATA)>
12.     <!ELEMENT prix (#PCDATA)>
13.     <!ELEMENT pages (#PCDATA)>
14.     <!ELEMENT dateParution (#PCDATA)>
15.     <!ELEMENT resume (#PCDATA)>
16.     <!ELEMENT commentaire (avis, date, note)>
17.
18.     <!ATTLIST commentaire nom CDATA #REQUIRED>
19.     <!ATTLIST commentaire prenom CDATA #REQUIRED>
20.
21.     <!ELEMENT avis (#PCDATA)>
22.     <!ELEMENT date (#PCDATA)>
23.     <!ELEMENT note (#PCDATA)>
24.     <!ELEMENT couverture EMPTY>
25.
26.     <!ATTLIST couverture img CDATA #IMPLIED>
27.
28.     <!ELEMENT genre ( science-fiction | aventures | philosophie| policier | horreur | tragédie |
comédie | drame | poésie | romance | fantastique | fantasy | autobiographie )>
29.     <!ELEMENT moyenne (#PCDATA)>
```