

## Introduction

La conception constitue une phase fondamentale dans le cycle de vie d'une application. La réussite de ce dernier dépend de cette phase. L'un des soucis était d'avoir une idée globale en avance de ce que nous devons programmer. Pour ce faire, on a commencé par les diagrammes de cas d'utilisation qui permettent de donner une vue globale de l'application. En deuxième lieu, nous allons présenter les diagrammes de séquence. Et enfin, nous finirons par les diagrammes de classe.

### I. Diagramme de cas d'utilisation :

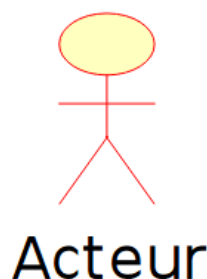
#### 1. Définition :

Chaque usage que les acteurs font du système est représenté par un cas d'utilisation. Chaque cas d'utilisation représente une fonctionnalité qui leur est offerte afin de produire le résultat attendu. Ainsi, « le diagramme de cas d'utilisation décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur ».

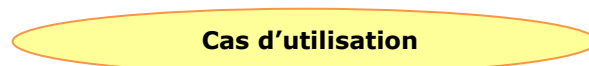
#### 2. Composition de diagramme de cas :

Le diagramme de cas se compose de trois éléments principaux :

**Un Acteur :** Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il se représente par un petit bonhomme avec son nom inscrit dessous.



**Un cas d'utilisation :** Un cas d'utilisation (« use case ») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.



**Les relations :** Trois types de relations sont pris en charge par la norme UML et sont graphiquement représentées par des types particuliers de ces relations. Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas issu. Une relation simple entre un acteur et une utilisation est un trait simple.

### 3. Les acteurs de notre projet :

**Le vendeur :** le rôle du vendeur repose sur présenter son produit faire le suivi de satisfaction de la clientèle fournir les informations demandées, faire la démonstration des articles si nécessaire

**Le Client :** cet acteur est un visiteur ayant déjà créer un compte sur notre site, il peut donc suivre le processus d'achat des produits en toute sécurité sachant que notre système doit être l'unique responsable de la confidentialité des données personnelles de ses clients.

**L'administrateur :** pour les sites web on l'appelle généralement « *le webmaster* ». C'est celui qui assure le dynamisme du site et veille sur les mises à jour des produits, de leurs prix, de leurs disponibilités, de la gestion des paiements et la gestion des livraisons.

#### 4. Diagrammes de notre projet :

##### a) Diagramme de cas d'utilisation général :

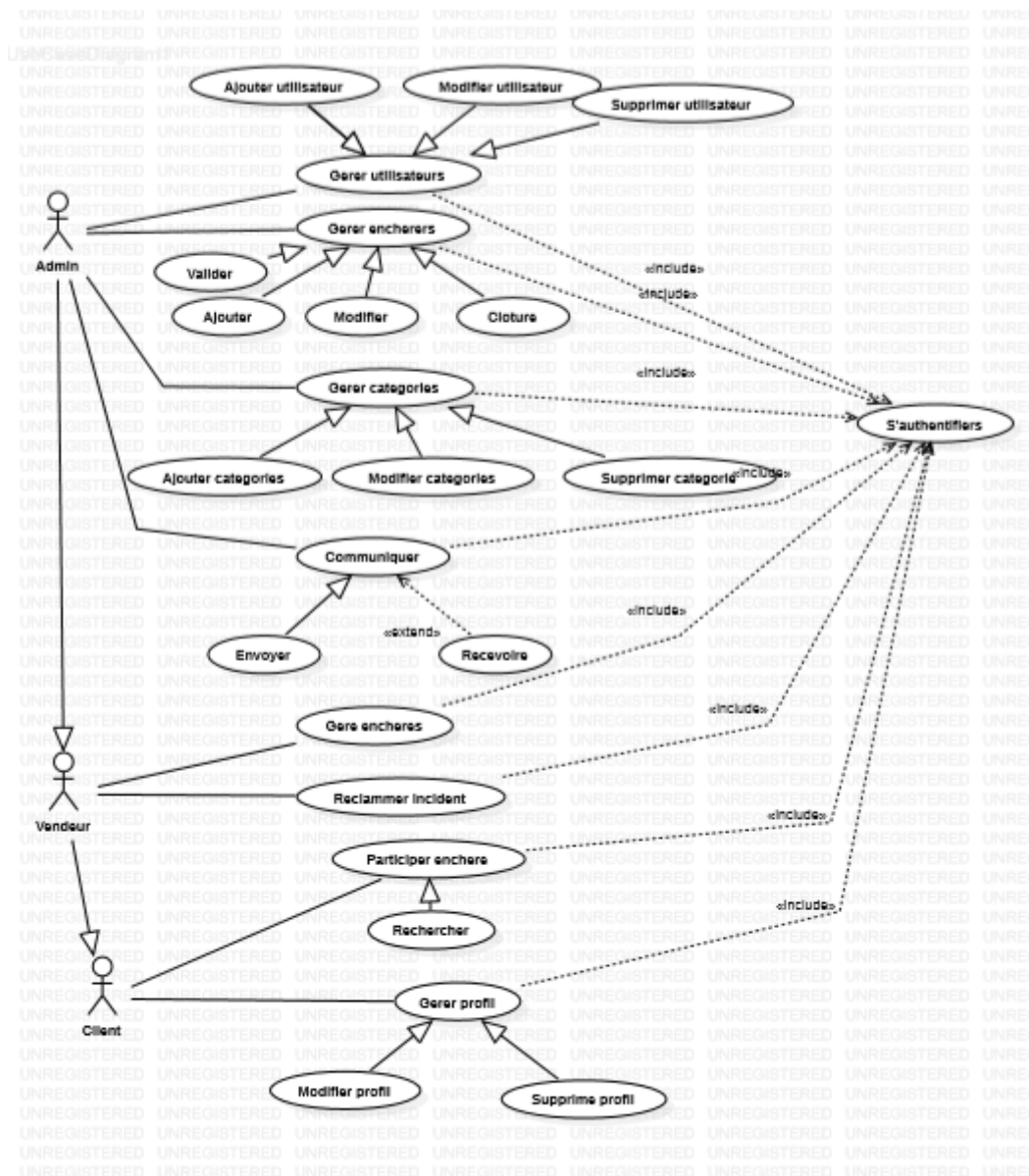


Diagramme 1 : Diagramme de cas d'utilisation générale

##### b) Diagramme de cas d'utilisation détaillée :

###### • Diagramme de cas de « vendeur » :

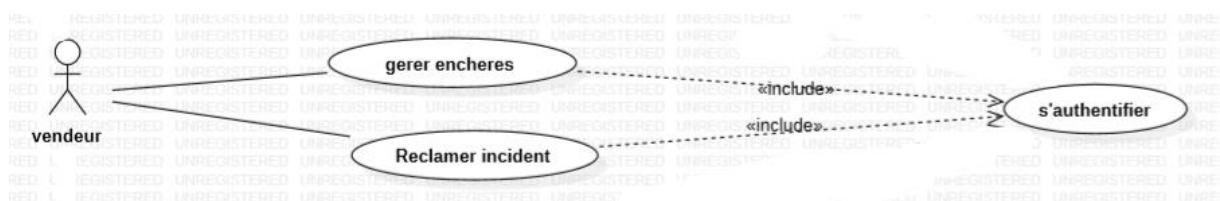


Diagramme 2 : Diagramme de cas d'utilisation « vendeur »

- **Diagramme de cas de « client » :**

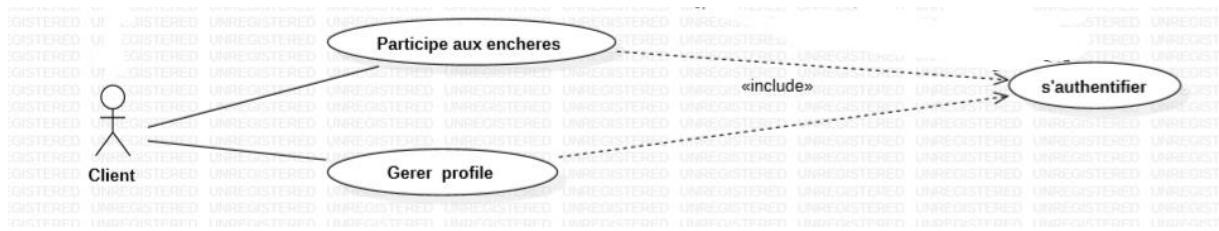


Diagramme 2 : Diagramme de cas d'utilisation « client »

- **Diagramme de cas de « admin » :**

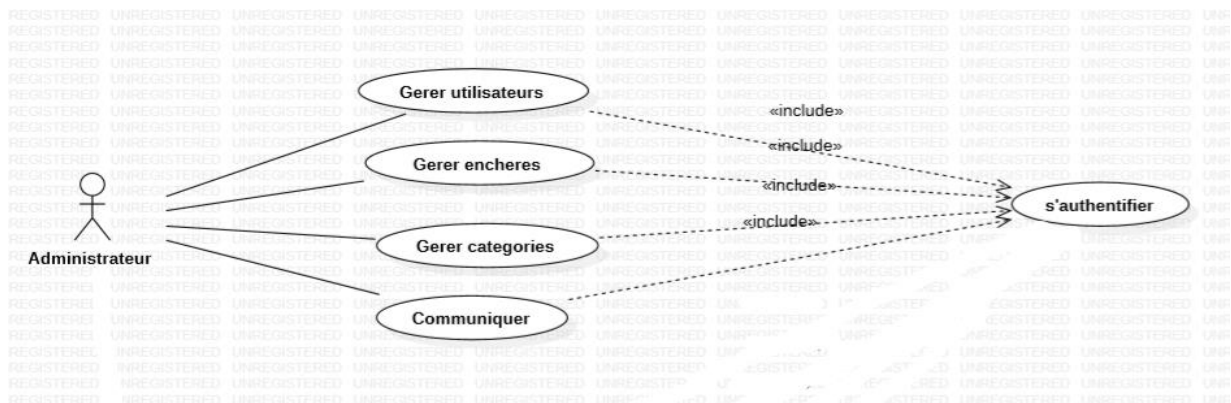


Diagramme 4 : Diagramme de cas d'utilisation « admin »

## II. Diagrammes des séquences :

### 1. Définition :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

### 2. Composition d'un diagramme de séquences :

Ce type des diagrammes est composé par les éléments suivants :

**Les lignes de vie :** Une ligne verticale qui représente la séquence des événements, produite par un participant, pendant une interaction, alors que le temps progresse en bas de ligne.

Ce participant peut être une instance d'une classe, un composant ou un acteur.

**Les messages** : deux types de messages dans le diagramme de séquences, le premier est dit message synchrone utilisé pour représenter des appels de fonction ordinaires dans un programme, le deuxième est appelé message asynchrone, étant utilisé pour représenter la communication entre des threads distincts ou la création d'un nouveau thread.

**Les occurrences d'exécution** : représente la période d'exécution d'une opération.

**Les commentaires** : Un commentaire peut être joint à tout point sur une ligne de vie.

**Les itérations** : représente un message de réponse suite à une question de vérification.

### 3. Diagrammes de notre projet :

- **Diagramme de séquences de cas d'authentification :**

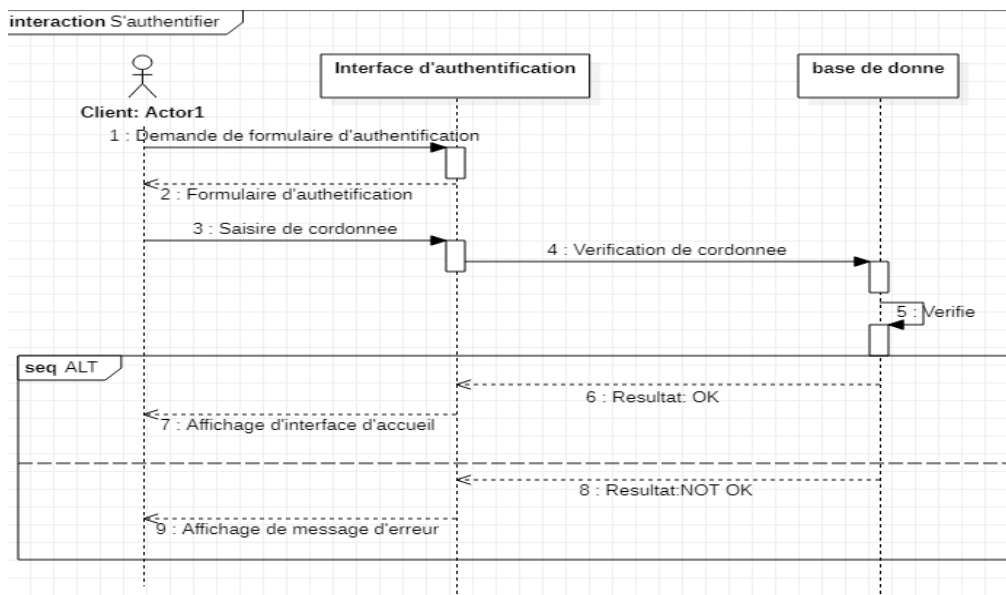


Diagramme 5 : Diagramme de séquence de cas d'utilisation « s'authentifier »

- **Diagramme de séquences de cas d'inscription :**

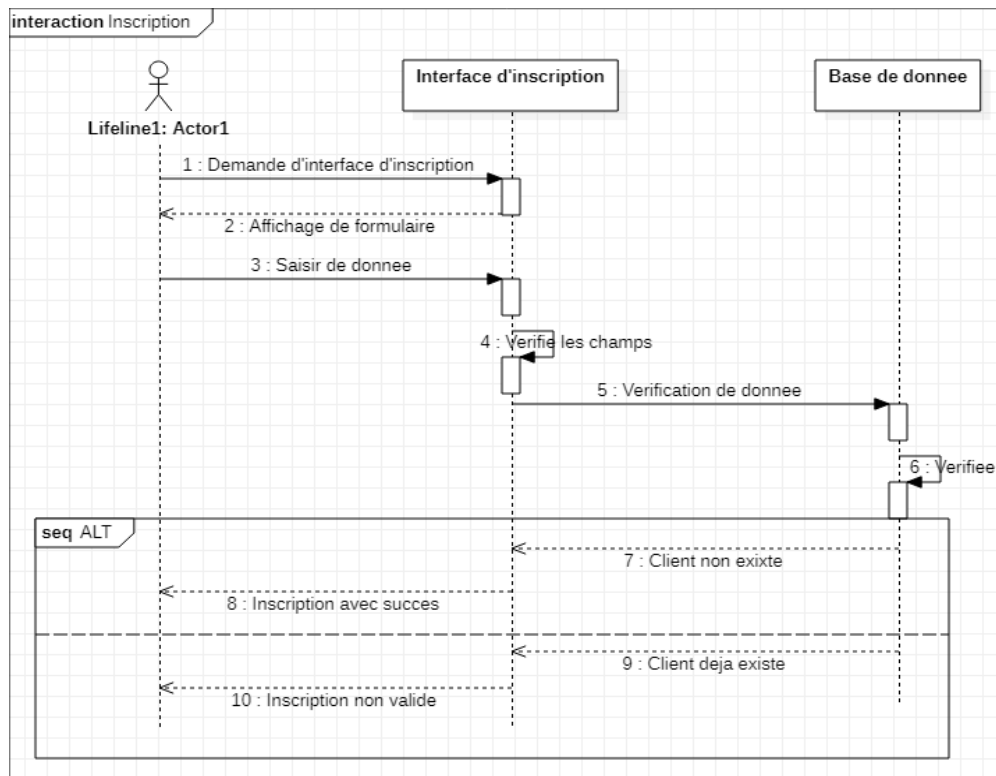


Diagramme 6 : Diagramme de séquence de cas d'utilisation « Inscription »

- **Diagramme de séquences de cas clôture enchère :**

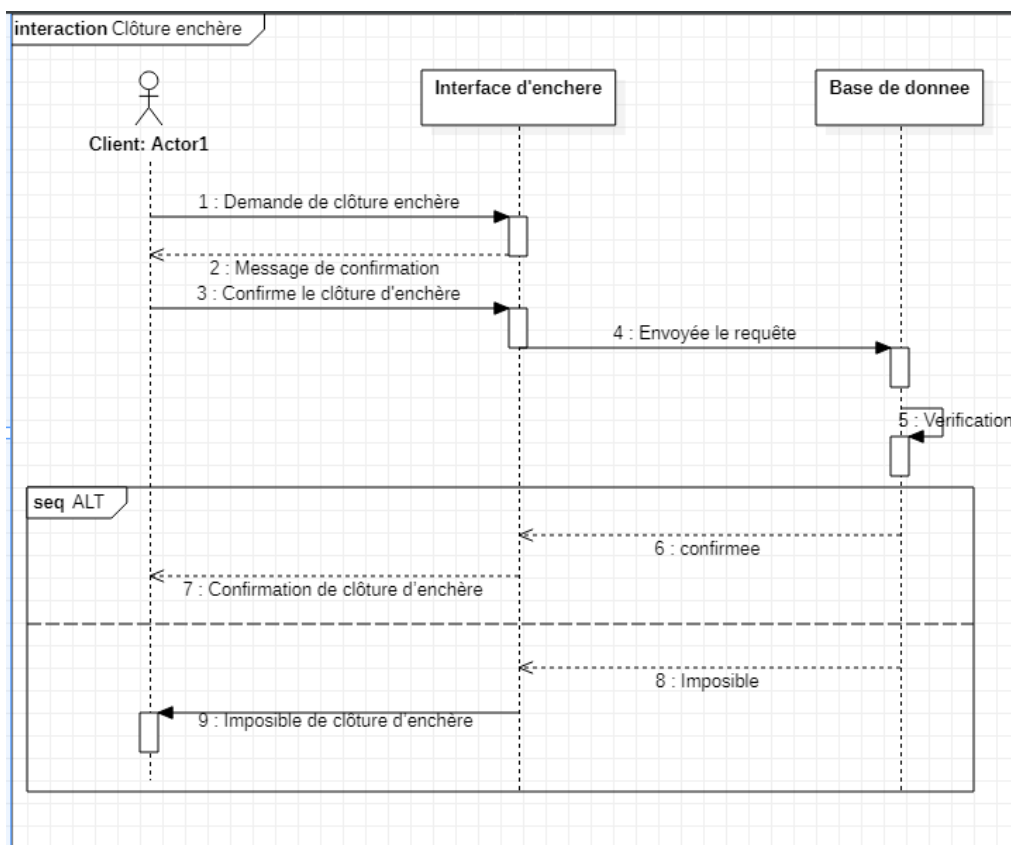


Diagramme 7 : Diagramme de séquence de cas d'utilisation « Clôture enchère »

- **Diagramme de séquences de cas participe aux enchères:**

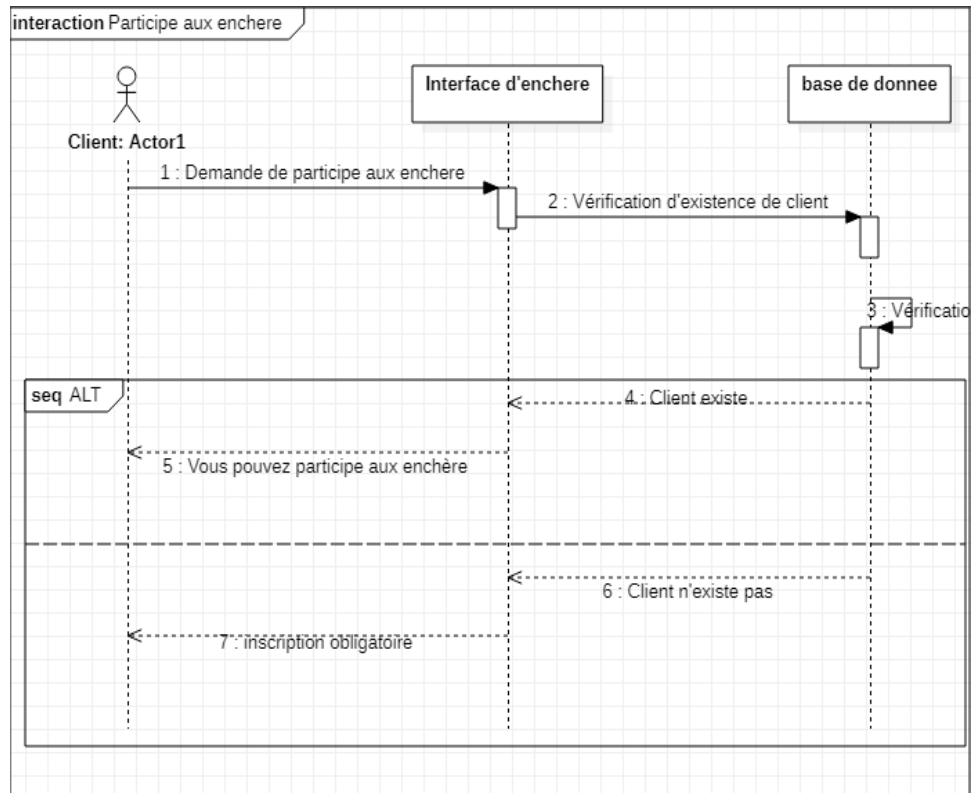


Diagramme 8 : Diagramme de séquence de cas d'utilisation «Participe aux enchère »

### III. Diagramme de classes :

#### 1. Définition

Un diagramme de classes UML décrit les structures d'objets et d'informations utilisées sur notre site web, à la fois en interne et en communication avec ses utilisateurs. Il décrit les informations sans faire référence à une implémentation particulière. Ses classes et relations peuvent être implémentées de nombreuses manières, comme les tables de bases de données, les nœuds XML ou encore les compositions d'objets logiciels.

#### 2. La composition d'un diagramme de classes :

En général un diagramme de classe peut contenir les éléments suivants :

**Les classes :** une classe représente la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes. Elle est représentée en utilisant un rectangle divisé en trois sections.

La section supérieure est le nom de la classe, la section centrale définit les propriétés de la classe alors que la section du bas énumère les méthodes de la classe.

**Les associations :** une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc que des liens peuvent exister entre des instances des classes associées.

**Les attributs :** les attributs représentent les données encapsulées dans les objets des classes. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l'attribut doit être unique dans la classe.

### 3. Notre diagramme des classes :



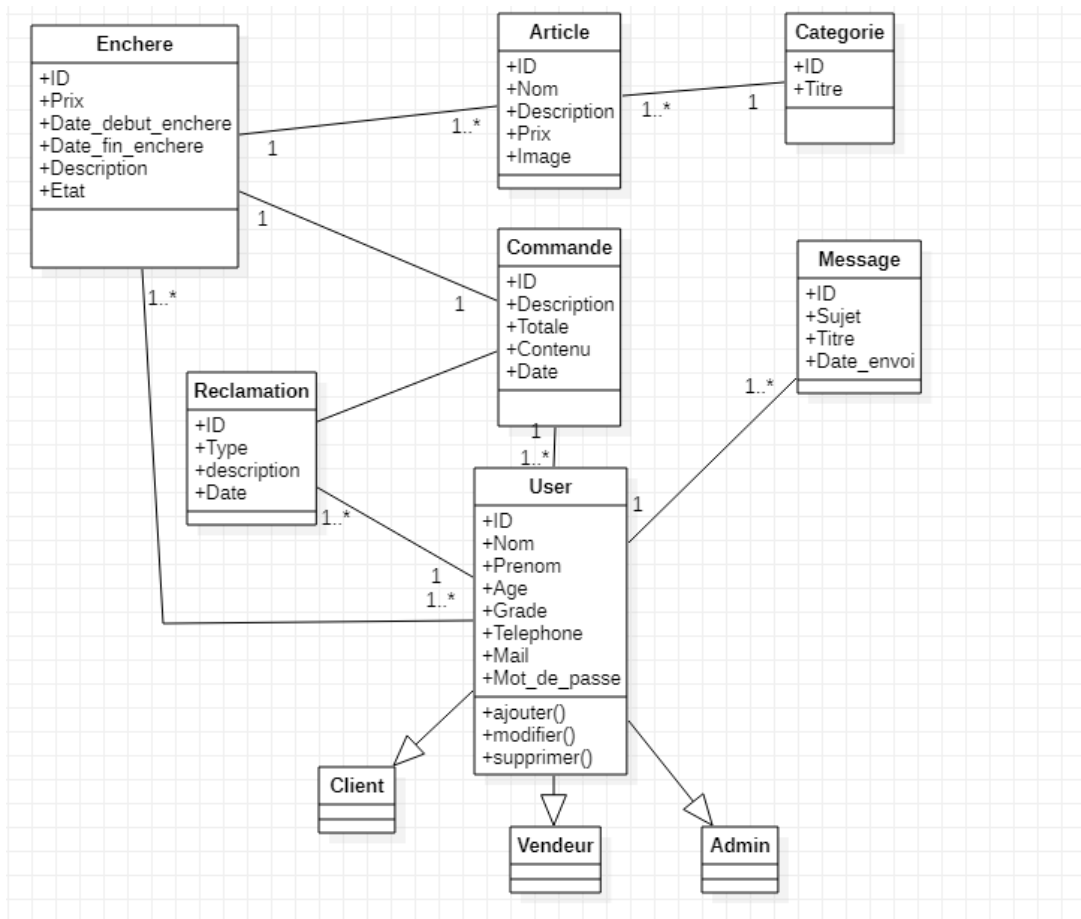


Diagramme 9 : Diagramme de classe général

## Conclusion :

Dans ce chapitre, on a étudié la conception de cette application. A ce propos, on a adopté la méthode de conception UML pour l'élaboration des diagrammes de cas d'utilisation, de classe et de séquence.

Dans le chapitre suivant on aborder la dernière partie qui représente la partie réalisation de notre site web.