**Chapitre II :**

**Etude conceptuelle**

**Introduction :**

La conception est une étape primordiale ayant une influence sur la qualité et la fiabilité des applications informatique. Dans ce cadre se situe le présent chapitre qui permettra de mettre en œuvre les diagrammes statiques du cas d’utilisation et des classes. Ensuite un diagramme dynamique qui est le diagramme de séquence.

**Outil de modélisation :**

Pour présenter l’architecture du système, on choisit **U.M.L** pour sa simplicité et lisibilité.

**U.M.L** : Le Langage de Modélisation Unifié, de l’anglais Unified Modeling Language, est un [langage](https://fr.wikipedia.org/wiki/Langage) de modélisation graphique à base de [pictogrammes](https://fr.wikipedia.org/wiki/Pictogramme) conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en [développement logiciel](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_logiciel) et en [conception orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet).

1. **Les diagrammes de cas d’utilisation :**
2. **Définition :**

Ce diagramme permet d’identifier les possibilités d’interaction entre le système et les acteurs (intervenants extérieurs au système). Il représente toutes les fonctionnalités que le système doit fournir.

Le diagramme de cas d’utilisation se compose de trois élément principaux :

1. **Acteur :**

C’est l’idéalisation d’un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.

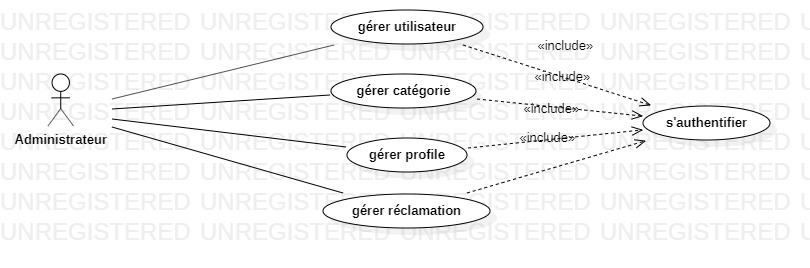
1. **Une cas d’utilisation :**

C’est une unité cohérente représentant une fonctionnalité visible de l’extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l’acteur qui l’initie.

1. **Les relations :**

Trois types de relations sont pris en charge par la norme **U.M.L** 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.

1. **Les diagrammes de cas :**
2. **Le diagramme de cas d’administrateur :**



**Figure : Diagramme de cas d’utilisation d’administrateur**

L’administrateur doit s’authentifier pour faire toute actions de gestion comme la gestion d’utilisateur, gestion des catégories, gestions de profiles et la gestion des réclamations.

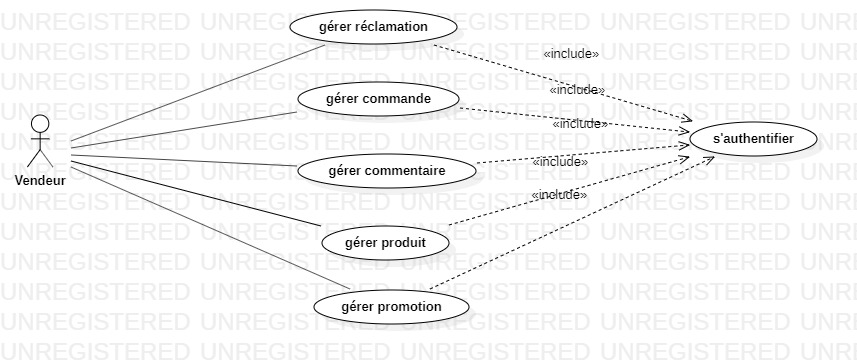
1. **Le diagramme de cas du client :**



**Figure : Diagramme de cas d’utilisation du client**

Le client doit s’authentifier pour faire toute action de gestion comme la gestion de commande et la gestion de commentaire. La gestion de commande début lorsque le client passe sa commande dont il peut contrôler sa commande et elle se termine lorsqu’il reçoit son produit. Alors que dans la gestion de commentaire le client peut ajouter ou supprimer des commentaires propres à son compte sous une publication.

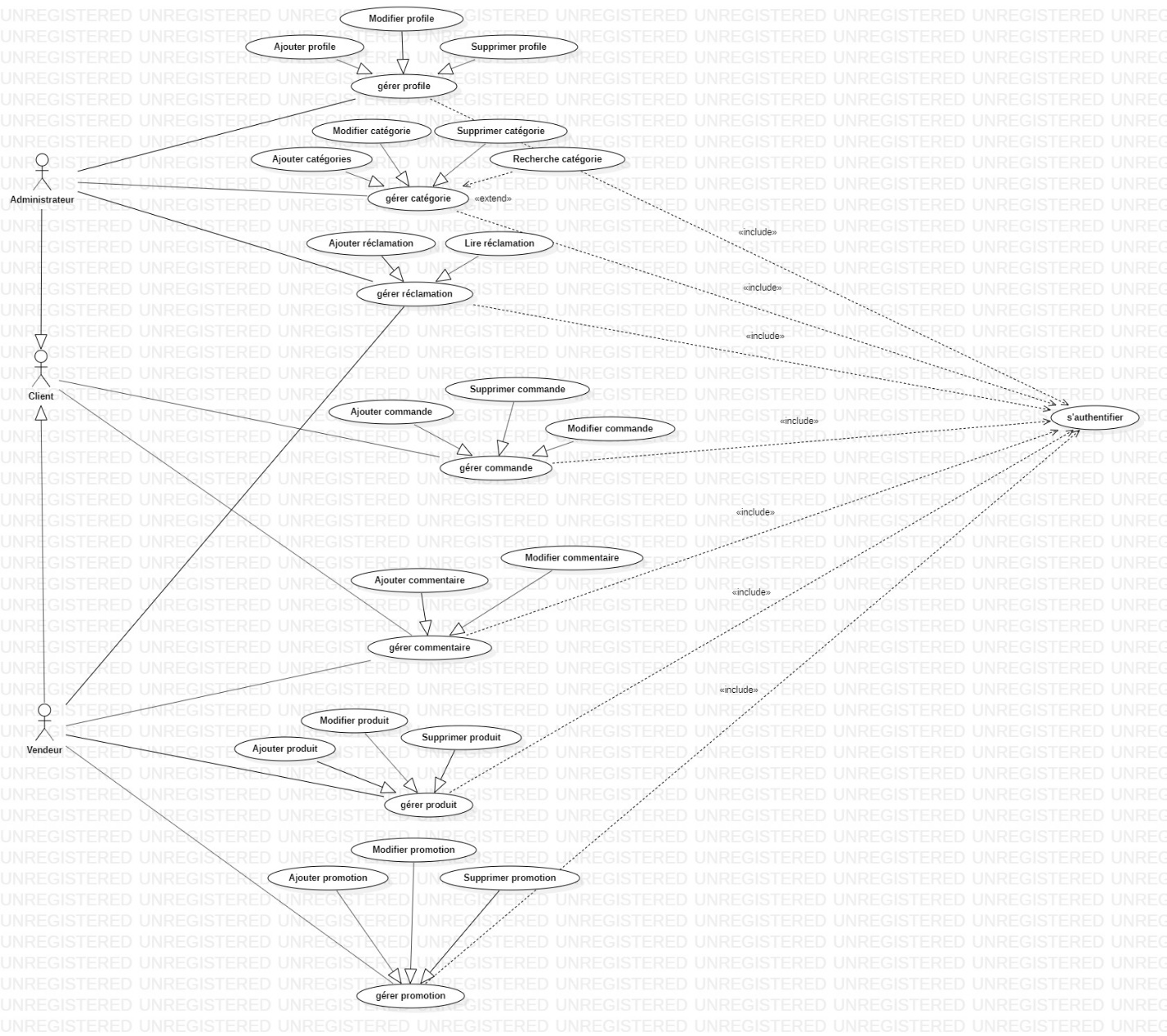
1. **Le diagramme de cas du vendeur :**



**Figure : Diagramme de cas d’utilisation du vendeur**

Le vendeur doit s’authentifier afin de faire toute action de gestion comme la gestion de réclamation, gestion de commande, gestion de commentaire, gestion de produit et la gestion de promotion. Et pour garantir qu’il est la bonne personne à travers l’interface d’application. La gestion de réclamation permet de vérifier si le produit est délivré ou si le produit délivré ne conforme pas le produit commandé. Pendant que la gestion de commentaire pour répondre à l’existence de produit dans sa boutique.

1. **Le diagramme de cas générale :**



**Figure : Diagramme de cas d’utilisation générale**

Ce diagramme représente la fonctionnalité de chaque acteur :

**L’administrateur :** Permet de gérer les utilisateurs, les catégories, les profile et les réclamations mais il ne peut pas effectue ces tâches qu’après une authentification réussie.

**Le client :** Permet de gérer les commandes et les commentaires d’après l’authentification.

**Le vendeur :** Est autorisé à exécuter les tâches suivantes : gérer les réclamations, les commandes, les commentaires, les produits et les promotions et l’authentification qui est nécessite pour accéder au système.

* L’administrateur possède tous les droits du l’utilisateur.
* Le vendeur peut être aussi un utilisateur.

1. **Les diagrammes de classes :**
2. **Définition :**

Le diagramme de classes exprime la structure statique du système en terme des classes et des relations entre ces classes. Il permet de représente l’ensemble des informations finalisées qui sont gérées par le domaine. Ces informations son structurées, c’est-à-dire qu’elles sont regroupées dans classes. Le diagramme de classes met en évidence d’éventuelles relations entre ces classes.

On a trois élément principaux du diagramme de classes :

1. **Une classe :**

Une classe est représentée par un rectangle séparé en trois parties :

* La première partie contient le nom de la classe
* La seconde contient les attributs de la classe
* La dernière contient les méthodes de la classe

La seconde et la dernière représentent le comportementde la classe.

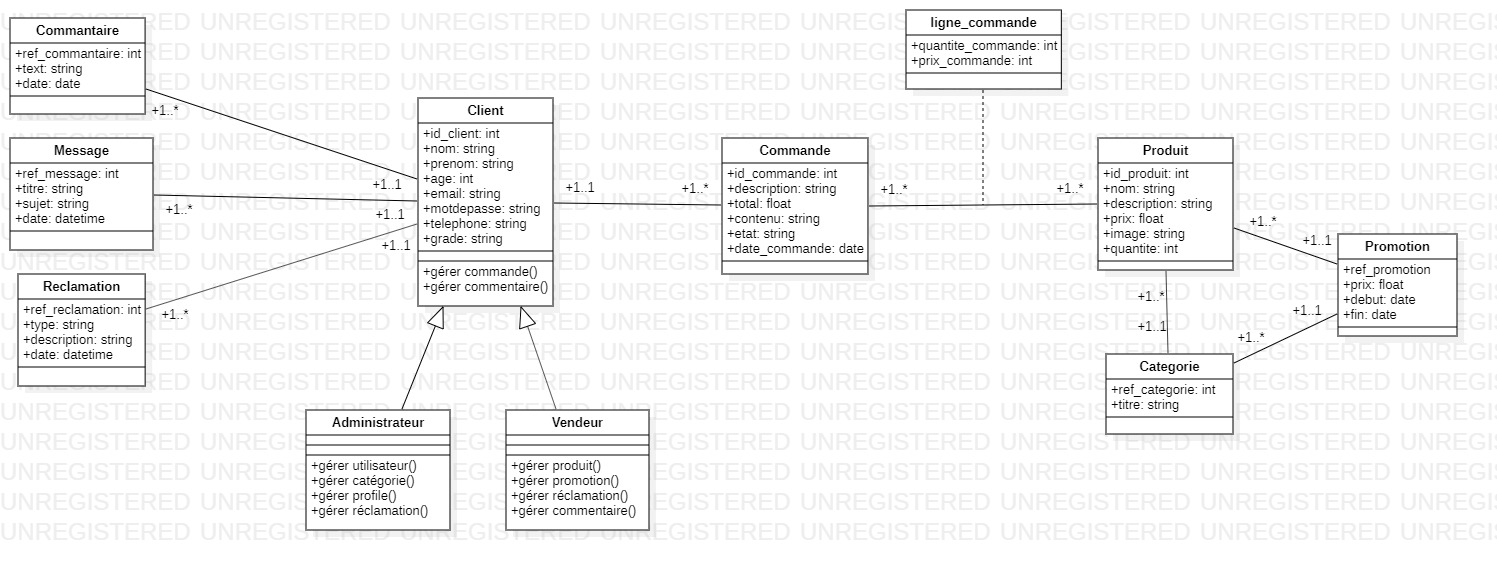
1. **Les relations :**

Ces relations ne sont pas exclusives au diagramme de classes, elles peuvent également s'appliquer à l'ensemble des diagrammes statiques.

On a plusieurs types de relation entre les classes :

* **Héritage** : Est un principe de division par généralisation et spécialisation, représenté par un trait reliant les deux classes et dont l'extrémité du côté de la classe mère comporte un triangle.
* **Association** : Est une connexion sémantique entre deux classes (relation logique). Une association peut être nommée. L'invocation d'une méthode est une association. Elle peut être binaire, dans ce cas elle est représentée par un simple trait, les classes sont reliées à un losange par des traits simples. Ces relations peuvent être nommées. L'association n'est utilisée que dans les diagrammes de classe.
* **Agrégation** : Est une association avec relation de subordination, représentée par un trait reliant les deux classes et dont l'origine se distingue de l'autre extrémité (la classe subordonnée) par un losange vide. Une des classes regroupe d'autres classes. L'objet T utilise une instance de la classe T’.
* **Composition** : Est une agrégation avec cycle de vie dépendant : la classe composée est détruite lorsque la classe mère disparait. L'origine de cette association est représentée par un losange plein.

1. **Le diagramme des classes :**



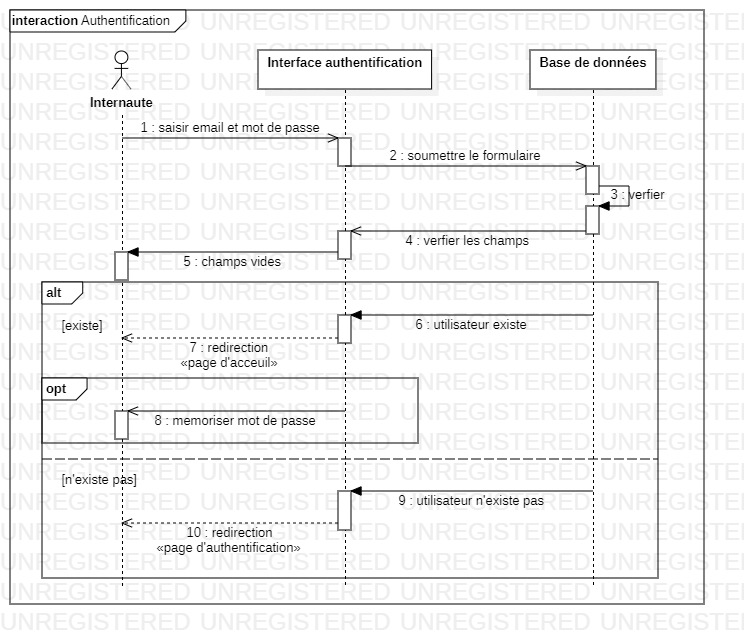
**Figure : Diagramme de classes**

* **Client :** Cette classe représente l’utilisateur qui peut être soit un admin soit un vendeur ou bien un client.
* **Commentaire :** Chaque publication peut être commentée par les utilisateurs.
* **Message :** Chaque utilisateur peut envoyer un message.
* **Réclamation :** Les réclamations de dérangement envoyés par le client.
* **Commande :** Chaque commande est envoyée par un client et elle constitué par plusieurs produits.
* **Produit :** Chaque produit a une identification unique.
* **Catégorie :** Chaque catégorie a un ensemble des produits différents.
* **Promotion :** La promotion fait peut-être dans chaque produit ou bien dans la catégorie.

1. **Les diagrammes de séquences :**
2. **Définition :**

Le diagrammes de séquences est la représentation graphique des [interactions](https://fr.wikipedia.org/wiki/Unified_Modeling_Language) entre les [acteurs](https://fr.wikipedia.org/wiki/Acteur_(UML)) et le système selon un ordre chronologique dans la formulation **UML**.

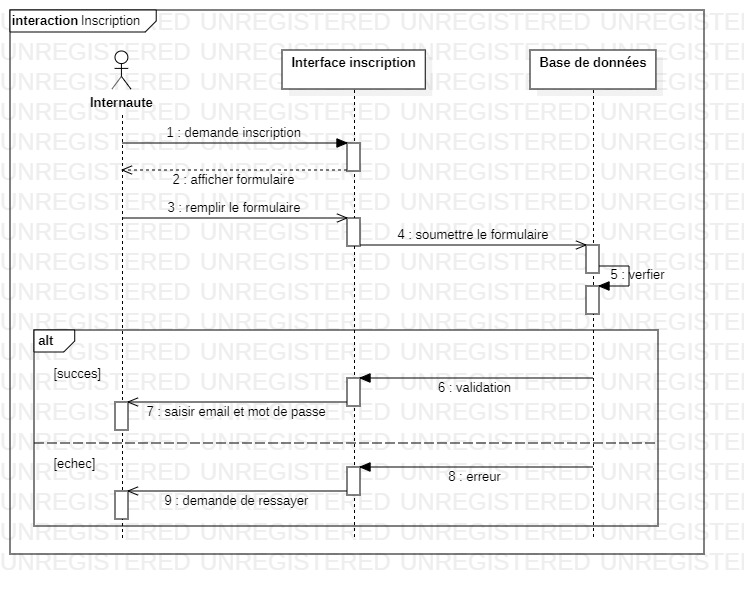
1. **Les diagrammes de séquence :**
2. **Authentification :**



**Figure : Diagramme de séquence authentification**

* **Acteur :** Admin/Vendeur
* **Objectif :** La sécurité et la confidentialité de l’accès à l’application
* **Description :** Pour accéder à l’application, l’utilisateur doit s’identifier par son email et son mot de passe le système vérifie si les champs correctes, l’utilisateur peut accéder à l’application. Sinon il doit vérifier ses champs.

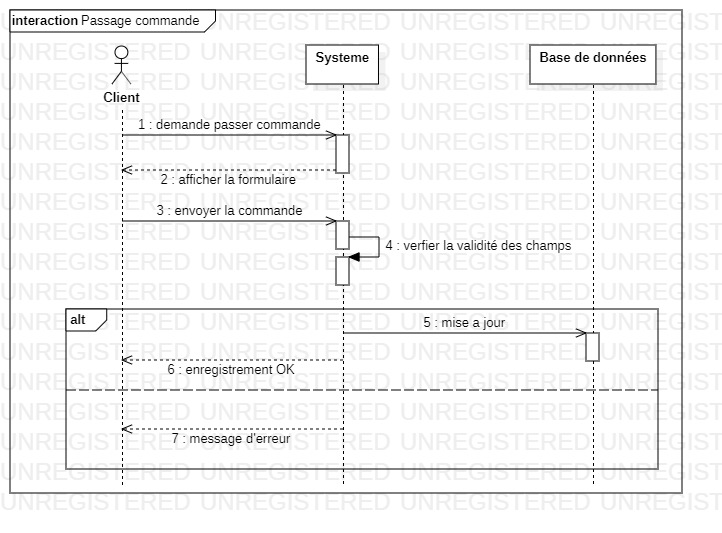
1. **S’inscrire :**



**Figure : Diagramme de séquence inscription**

* **Acteur :** Admin/Vendeur
* **Objectif :** Créer un compte
* **Description :** Pour créer un compte, l’utilisateur doit remplir le formulaire. Le système vérifie si les champs ne sont pas enregistrés, une fois les vérifications terminées avec sucées sinon un message d’erreur et doit ressayer les champs.

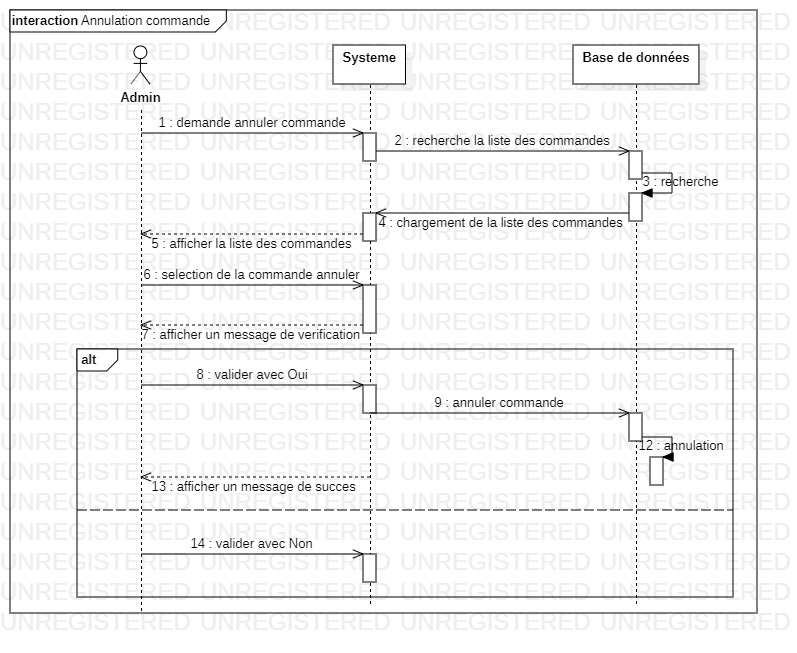
1. **Passage du la commande :**



**Figure : Diagramme de séquence passage commande**

* **Acteur :** Client
* **Objectif :** Passer une commande
* **Description :** Pourpasser une commande, le client doit remplir le formulaire.Le passage du la commande enregistre si les champs sont vrais. Sinon le système envoie un message d’erreur au client pour lui informer que son commande n’a pas enregistrer.

1. **Annulation du la commande**



**Figure : Diagramme de séquence annulation commande**

* **Acteur :** Admin
* **Objectif :** Annuler une commande
* **Description :** Afin d’établir l’annulation du la commande, l’administrateur doit sélectionner la commande approprie. Le système fait la vérification et par la suite un message de validation affichée pour la confirmation de l’annulation du la commande.

**Conclusion :**

Dans ce chapitre je présente ma conception avec une vue d’ensemble et modélisation **U.M.L** faite à travers des vues statiques et des vues dynamiques. Les vues statiques ont été les diagrammes du cas d’utilisation et de la classe alors que les vues dynamiques ont été les diagrammes de séquence. Cette partie conduit à une réalisation qui sera dans le chapitre prochain.