

Université Abou Bekr Belkaid  
Tlemcen Algérie



جامعة أبي بكر بلقايد

تلمسان الجزائر

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en informatique

*Thème :*

**Conception et réalisation d'une application pour la  
gestion des stocks cas d'étude magasin de  
société GRTE.Spa, filiale du Groupe SONELGAZ  
Poste de TLEMEN**

Réalisé par :

- ❖ MERAHI Aissa Nadir
- ❖ LOUAHED Abdellatif

Présenté le 28-05-2015 devant la commission d'examination Composée par :

- Mr Benmouna M ( encadreur)
- Mr Matallah H (examineur)
- Mr Brikci A. (examineur)

*Année universitaire 2014-2015*

# *Remerciements*

*Louange à Dieu Miséricorde et Miséricordieux qui nous a donné la force, la*  
*volonté*

*et la patience durant toutes nos années d'études.*

*Nous remercions nos très chers parents qui nous ont accordé le courage, la*  
*volonté et le soutien pour bien réaliser ce travail.*

*nos remerciements vont à toutes les personnes qui de près ou de loin nous ont*  
*aidés par leurs encouragements.*

# Table des matières

Généralité .....	01
<b>Chapitre I : L'étude du système existant et le langage de modélisation UML.....</b>	<b>03</b>
I.1 introduction .....	04
I.2 présentation du société.....	05
I.3 Contexte du système: .....	06
I.3.1 Circuit du déroulement des activités relatives à la facture.....	06
I.3.2 Circuit du déroulement des activités relatives au bon de commande des équipes.....	07
I.3.3 Problématique.....	07
I.3.4 Proposition des solutions .....	07
I.4 Le langage de modélisation UML.....	08
I.4.1 Définition UML .....	08
I.4.2 La caractéristique d'ULM.....	08
I.4.3 Les diagrammes.....	09
I.4.3.1 Le besoin des utilisateurs. ....	10
I.4.3.2 L'aspect fonctionnel du logiciel.....	11
• Vue logique (1) .....	11
• Vue des processus (2) .....	13
I.4.3.3 L'aspect lié à l'architecture du logiciel.....	16
• Vue des composants (4) .....	16
• Vue de déploiement (5) .....	17
I.5 Processus UP.....	18
I.5.1 Définition du processus unifié .....	18
I.5.2 introduction .....	18
I.5.3 Les neuf disciplines d'UP .....	19
I.5.4 Phase du processus unifié .....	19
I.5.5 Construction.....	20

I.6 Conclusion .....	20
<b>Chapitre II Analyse et La conception Du Syst.</b> .....	21
II.1-Introduction .....	22
II.2- Modélisation du système .....	22
• a.Diagramme de cas d'utilisation .....	23
• b.Les diagrammes de séquence .....	24
• c- Diagramme de classe .....	34
II.3- Conclusion.....	34
<b>Chapitre III L'Implémentation du Système</b> .....	35
III.1- Introduction .....	36
III .2 Choix du langage de programmation.....	36
III .3 Objectifs de la programmation objet.....	36
III .4 Outils de développement.....	36
III.5-Description de l'application .....	38
III.5.1 Page d'authentification .....	38
III.5.2 Menu principal .....	39
III.5.3 Formulaire produit .....	39
III.5.4 Formulaire utilisateur.....	40
III.5.5 Formulaire commande .....	44
III.5.6 Formulaire Fournisseur.....	40
III.6- Conclusion .....	41
<b>Conclusion générale</b> .....	42
Références bibliographiques .....	42

# INTRODUCTION GENERALE

De nos jours, le rôle de l'informatique devient indispensable et sa présence de plus en plus répandue par son efficacité ainsi que son utilité. Ce fait s'explique par les applications importantes de l'informatique dans presque tous les domaines de l'entreprise et de la vie quotidienne. Une forte informatisation dans tous les secteurs de l'économie est un exemple le plus visible, y compris celui de la gestion des stocks auquel nous rattacherons d'ailleurs notre étude, et cela pour une meilleure gestion des différents traitements exigés par cette activité de gestion des stocks.

Nous avons pu constater, en effet, pendant nos recherches que l'ensemble des traitements au sein du magasin de société GRTE.Spa, filiale du Groupe SONELGAZ, poste de TLEMCEM se fait manuellement, ce qui engendre un certain nombre de problèmes tels que la lenteur dans l'accès aux données et le risque de perte d'informations, donc la meilleure solution pour pallier aux problèmes et l'informatisation afin d'assurer l'accès instantané aux données et une sécurisation de ces dernières, ce qui simplifie le travail administratif.

Pour la réalisation de cette tâche, notre choix s'est porté sur le Processus Unifié. En effet, le processus unifié est une solution de développement logiciel adaptée à tout type de projet. Ses traits distinctifs tiennent en trois notions : piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental.

Le langage de modélisation qu'on a utilisé est UML (Unified Modeling Language), qui est une partie intégrante de la démarche UP. Ses diagrammes sont largement utilisés dans chaque étape et phases de ce processus de développement.

Pour l'implémentation, le choix du langage de programmation a été dicté par le type de l'application qui devrait être réalisée, le choix s'est porté sur le langage de programmation JAVA et La base de données est implémentée avec MySQL qui est largement compatible avec JAVA.

Ayant présenté les outils et la méthode adoptée, nous allons maintenant exposer le plan du mémoire qui se subdivisera en quatre principaux chapitres.

Dans le premier chapitre intitulé « Généralités » nous présentons quelques concepts de base de la méthode et outils qui nous ont servi pour l'accomplissement de notre projet.

Puis, au sein de « Spécification des besoins », deuxième chapitre de ce travail, nous commençons à présenter l'organisme d'accueil, approfondir la compréhension du contexte du système par un processus continu de collecte d'information auprès des experts du domaine, ensuite déterminer les inconvénients majeurs de la gestion actuelle du service d'accueil, énumérer des suggestions informatiques qui peuvent remédier aux difficultés rencontrées, et enfin, tenant compte des moyens de la société, nous proposons la solution qui paraît la plus adéquate.

Au niveau de troisième chapitre intitulé « Conception et Analyse des besoins » nous analysons les principaux objectifs attendus du futur système à concevoir, et qui seront décrits par le diagramme des cas d'utilisation. et nous étendons les représenter en y intégrant les aspects techniques les plus proches des préoccupations physique.

Finalement dans le dernier chapitre qu'on a nommé « implémentation » nous présentons les outils de développement qui nous ont servi pour le développement de l'application, et enfin l'activité test qui consiste, justement, à la tester dans le but de s'assurer de son bon fonctionnement.

# **Chapitre I**

**L'étude du système existant**

**&**

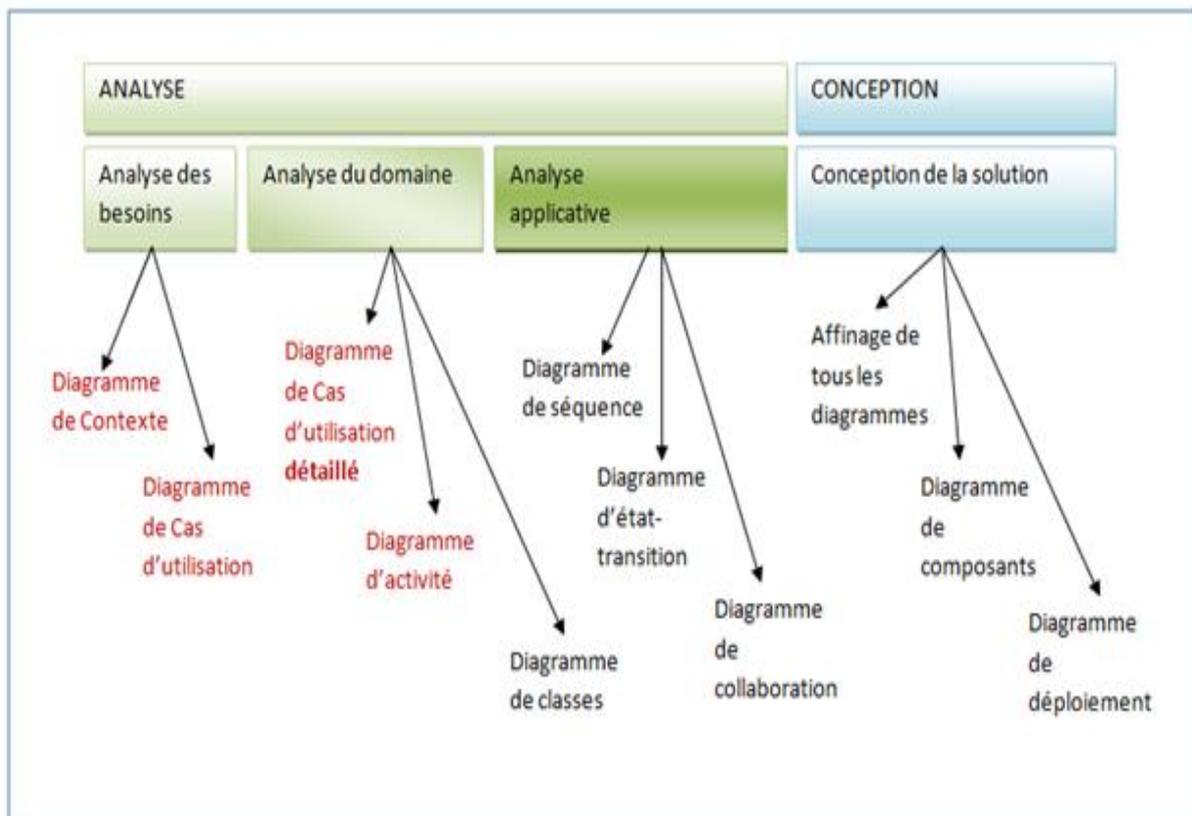
**Le langage de modélisation UML**

I.1 Introduction

Comme n'importe quel type de projet, un projet informatique nécessite une phase d'analyse, suivi d'une étape de conception.

Dans la **phase d'analyse**, on cherche d'abord à bien comprendre et à décrire de façon précise les besoins des utilisateurs ou des clients. Que souhaitent-ils faire avec le logiciel ? Quelles fonctionnalités veulent-ils ? Pour quel usage ? Comment l'action devrait-elle fonctionner ? C'est ce qu'on appelle « **l'analyse des besoins** ». Après validation de notre compréhension du besoin, nous imaginons la solution. C'est la partie **analyse de la solution**.

Dans la **phase de conception**, on apporte plus de détails à la solution et on cherche à clarifier des aspects techniques, tels que l'installation des différentes parties logicielles à installer sur du matériel.



**I.2 présentation du société :**

La loi n° 02-01 du 5 février 2002 relative à l'électricité et à la distribution du gaz par canalisations introduit l'ouverture progressive du marché électrique et impose la séparation des activités de Transport, Production, Distribution et règlemente l'accès des Tiers au réseau.

Le Gestionnaire du Réseau de Transport de l'Electricité, dénommée GRTE.Spa, filiale du Groupe SONELGAZ, a été créé le 1er janvier 2004, conformément à cette loi et enregistrée sous l'appellation « SONELGAZ Transport de l'Electricité, GRTE Spa » en septembre 2004. A cet égard, le Gestionnaire dispose d'une autorisation d'exploiter le réseau de transport délivrée par le ministère de l'énergie et des Mines, après avis de la Commission de Régulation de l'électricité et du gaz (CREG).

La raison sociale de l'entreprise devient en février 2009, « Société Algérienne de Gestion du Réseau de Transport de l'Electricité, dénommée GRTE Spa », Société par Actions au capital de 40.000.000.000 DA. GRTE est chargé de l'exploitation, de la maintenance et du développement du réseau de transport de l'électricité, en vue de garantir une capacité adéquate par rapport aux besoins de transit et de réserve

La société GRTE.Spa est administrée par un Conseil d'Administration, présidé par son Président Directeur Général. Elle est aussi dotée d'une Assemblée Générale. Les relations techniques entre les différents opérateurs (Producteurs, Sociétés de distribution (SDX) et Clients Haute tension) raccordés au réseau de transport d'électricité sont régies par l'arrêté ministériel du 28 février 2008, fixant les règles techniques de raccordement au réseau de transport et les règles de conduite du SPTE (Système de Production Transport de l'Electricité).

GRTE assure ses activités à travers des directions centrales et cinq directions de Régions Transport de l'Electricité : Alger, Oran, Sétif, Annaba, Hassi Messaoud. Ces régions, à travers 24 services de transport répartis sur le territoire national assurent une maintenance de proximité et la relation directe avec les clients.

Le réseau de transport est un ensemble d'ouvrages lignes et postes haute tension (60, 90, 150, 220 et 400 kV) équipé d'un réseau de télécommunication pour la surveillance, le contrôle et la télécommande des équipements.

Il est constitué d'un réseau interconnecté au nord du pays, avec des interconnexions internationales (Tunisie et Maroc) et d'un réseau isolé au sud.

Les utilisateurs du réseau sont les centrales électriques, les sociétés de distributions de l'électricité et clients HT ainsi que pour les échanges internationaux à travers les interconnexions. GRTE exploite un réseau composé de :

- 25 147 km de lignes dont 2 547 km en 400 kV
- 283 postes (dont 12 en 400 KV) dotés d'une capacité de transformation de 48 806 MVA à travers 773 transformateurs et cabines mobiles

### **I.3 Contexte du système:**

Pour concevoir et réaliser le système logiciel de gestion de stock de magasin de la société GRTE.Spa, il nous était indispensable de collecter les informations nécessaires auprès des spécialistes du domaine. Après avoir structuré les informations collectées, nous avons remarqués que presque tout, se déroule autour de deux opérations principales qui sont :

- L'arrivée des produits facturés par le fournisseur.
- L'arrivée du bon de commande interne établi par un service demandeur.

Nous allons présenter le parcours du traitement de ces opérations de l'arrivée du bon de commande interne ou de la facture auprès du magasinier jusqu'à leurs accomplissements.

#### **I.3.1 Circuit du déroulement des activités relatives au bon de commande de société :**

A l'arrivée de la facture du fournisseur, ainsi que la marchandise commandée, le magasinier procède comme suit:

Il vérifie tout d'abord la conformité de la marchandise conformément au bon de commande ; ensuite le magasinier enchaîne avec ces deux opérations:

1. Etablir un bon d'entrée (bon de réception) sur lequel il saisit son numéro et sa date d'établissement. Par la suite, il note dans chaque ligne de bon: la désignation du produit réceptionné, le numéro de facture, date de réception, le nom du fournisseur et enfin la quantité livrée.

2. Enregistrer les produits réceptionnés dans deux registres différents ; il s'agit du registre d'entrée et celui d'inventaire. Le premier sert à enregistrer tous les produits réceptionnés étant consommables ou non consommables, en mentionnant la désignation des produits, le numéro de la facture, etc. Le deuxième sert à enregistrer uniquement les produits non consommables

en mentionnant les mêmes informations du registre d'entrée en rajoutant pour chaque produit un numéro d'inventaire.

**NB** : Lorsque le stock de sécurité d'un (des) produit(s) est atteint, le magasinier informe le chef de poste.

### **I.3.2 Circuit du déroulement des activités relatives au bon de commande des équipes**

Les équipes du poste peuvent commander des produits au Magasin en établissant un bon de commande (interne) qui sera adressé au magasinier.

En effet, à l'établissement du bon de commande (interne) le demandeur mentionne le numéro de bon de commande, la date de l'établissement, son nom, son prénom, sa fonction ; en suite il mentionne les produits à demander en précisant pour chacun : la désignation du produit et la quantité demandée. Et, il l'envoie au magasinier, signé par le responsable de l'équipes. A la réception, le magasinier traite la commande selon l'état de stock, et il affecte pour chaque produit demandé une quantité à accorder ( $\leq$  quantité demandée), et qui sera servie une fois eu l'avis favorable du chef de poste. A la livraison des produits, le magasinier établit un bon de sortie qu'est caractérisé par la quantité servie, Et pour les suivis de l'état de stock, le magasinier assigne pour chaque produit une fiche de stock.

### **I.3.3 Problématique :**

Le système d'information actuel de GRTE.Spa, poste de TLEMCEN et plus particulièrement le magasin de la est non encore automatisé, toute fois les tâches et procédures administratives de l'organisation qui contrôlent et traitent les mouvements des stocks sont manuelles, ce qui rend la tâche du magasinier fastidieuse et complexe et ne peut entièrement les assumer. La fatigue, l'oubli et l'erreur ont de grandes répercussions sur la qualité du travail et par conséquent, un effet négatif sur les résultats attendus de l'organisation.

### **I.3.4 Proposition des solutions :**

Pour faire à ces différents problèmes que rencontre le magasin dans ses activités journalières, un système informatisé est nécessaire pour remédier à ces problèmes. Ce

système devrait permettre au magasin de faire la saisie de toutes les entrées sorties du produit

Notre travail de recherche aboutira aux objectifs suivants :

- Suivi des commandes pour les fournisseurs et les équipes
- Eviter la perte des documents
- Faciliter le rangement du produit au stock.
- Faciliter l'inventaire périodique.
- Les statistiques du produit utilisé.

## **I.4 Le langage de modélisation UML**

### **I.4.1 Définition UML :**

UML, c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un **langage visuel** constitué d'un ensemble de schémas, appelés des **diagrammes**, qui donnent chacun une vision différente du projet à traiter. C'est un langage de modélisation qui permet de représenter graphiquement les besoins des utilisateurs : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

### **I.4.2 La caractéristique d'UML :**

UML est un langage formel et normalisé caractérisé par :

- Grande précision.
- Gage de stabilité.

- L'utilisation d'outils.
  - Encourage un support de communication performant
  - Il cadre l'analyse.
  - Il facilite la compréhension de diagramme ou de représentation abstraite complexe.
  - Son caractère polyvalent et souple en font un langage universel.
  - La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- ❖ Le processus (non couvert par UML) est une autre clé de la réussite d'un projet or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est tâche complexe et longue.

### **I.4.3 Les Diagrammes UML**

Le langage UML est constitué de diagrammes. À ce jour, il existe **13 diagrammes** « officiels ». Ces diagrammes sont tous réalisés **à partir du besoin des utilisateurs** et peuvent être regroupés selon les deux aspects suivants :

- **Les aspects fonctionnels** : Qui utilisera le logiciel et pour quoi faire ? Comment les actions devront-elles se dérouler ? Quelles informations seront utilisées pour cela ?
- **Les aspects liés à l'architecture** : Quels seront les différents composants logiciels à utiliser (base de données, bibliothèques, interfaces, etc.) ? Sur quel matériel chacun des composants sera installé ?

UML modélise donc le système logiciel suivant ces deux modes de représentation.

Nous allons donc dans un premier temps décrire ces différents aspects d'un logiciel grâce au schéma 4+1 vues et parcourir brièvement les différents diagrammes qui les composent.

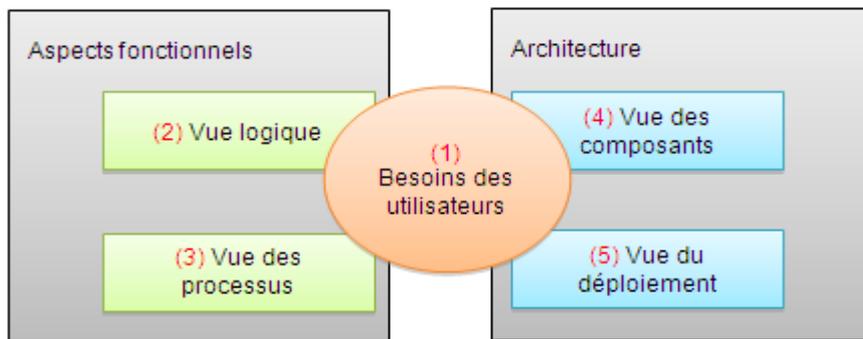
Comme nous avons le disais en introduction, la conception d'un logiciel est organisée autour des aspects fonctionnels et d'architecture. Ces deux aspects sont représentés par le schéma de 4 vues, axées sur les besoins des utilisateurs (parfois intitulé des cas d'utilisation), appelé

**4+1 vues.**

Une première décomposition d'une problématique ou système peut donc être faite à l'aide de

4+1 vues. Le schéma ci-dessous montre les différentes vues permettant de répondre au mieux

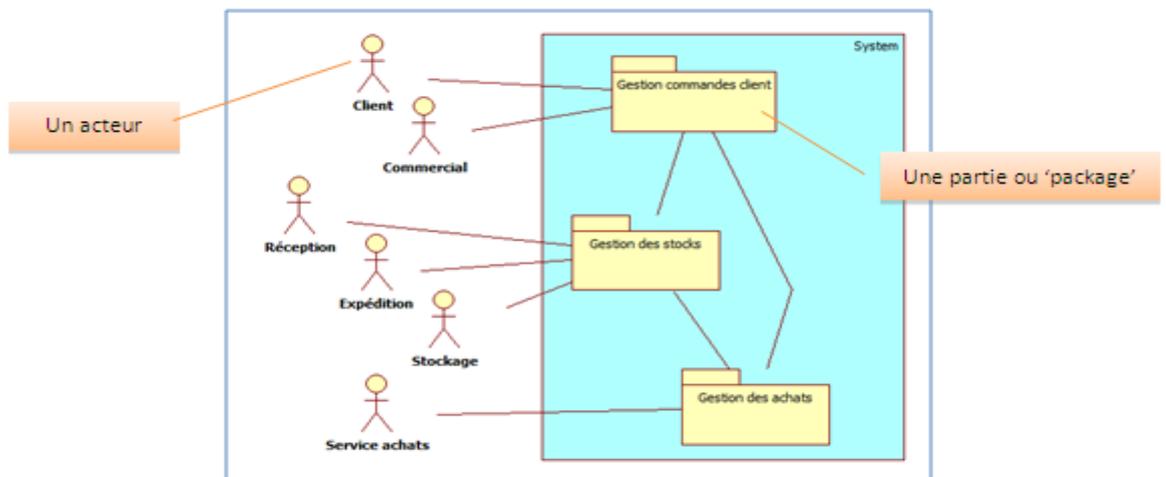
aux besoins des utilisateurs, organisées selon les deux aspects (fonctionnels et architecture). Chacune des vues est constituée de diagrammes.



Les 4+1 vues d'un système

**I.4.3.1 Le besoin des utilisateurs :** peut être décrit à l'aide de deux diagrammes.

- **Le diagramme de packages** permet de décomposer le système en catégories ou parties plus facilement observables, appelés « packages ». Cela permet également d'indiquer les acteurs qui interviennent dans chacun des packages.



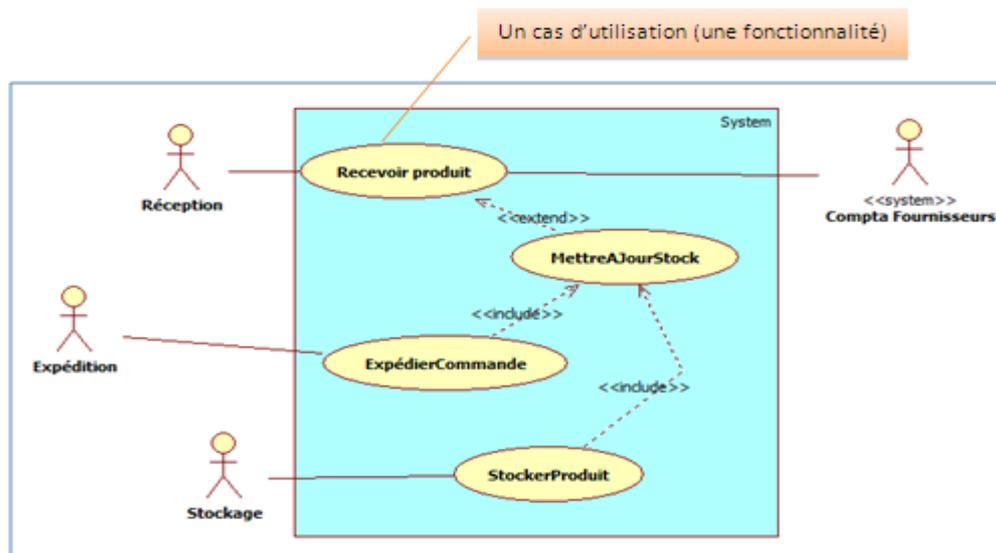
Un exemple de diagramme de package

Dans l'exemple précédent, nous voyons que le logiciel que nous concevons peut être divisé en trois parties (ou packages) observables séparément :

1. La gestion des commandes client
2. La gestion des stocks
3. La gestion des achats

La boîte qui entoure les packages (la boîte bleu) correspond au système (c'est-à-dire le logiciel) qui est analysé.

- **Le diagramme de cas d'utilisation** représente les fonctionnalités (ou dit cas d'utilisation) nécessaires aux utilisateurs. On peut faire un diagramme de cas d'utilisation pour le logiciel entier ou pour chaque package.



**UnExemple de diagramme de cas d'utilisation pour un package (Gestion des stocks)**

Étant donné que le diagramme de cas d'utilisation détaille le contenu d'un package, ici la boîte bleue correspond au package qui est détaillé.

### I.4.3.2 L'aspect fonctionnel du logiciel :

Pour rappel, cette partie du schéma 4+1 vues permet de définir qui utilisera le logiciel et pour quoi faire, comment les fonctionnalités vont se déroulement, etc.

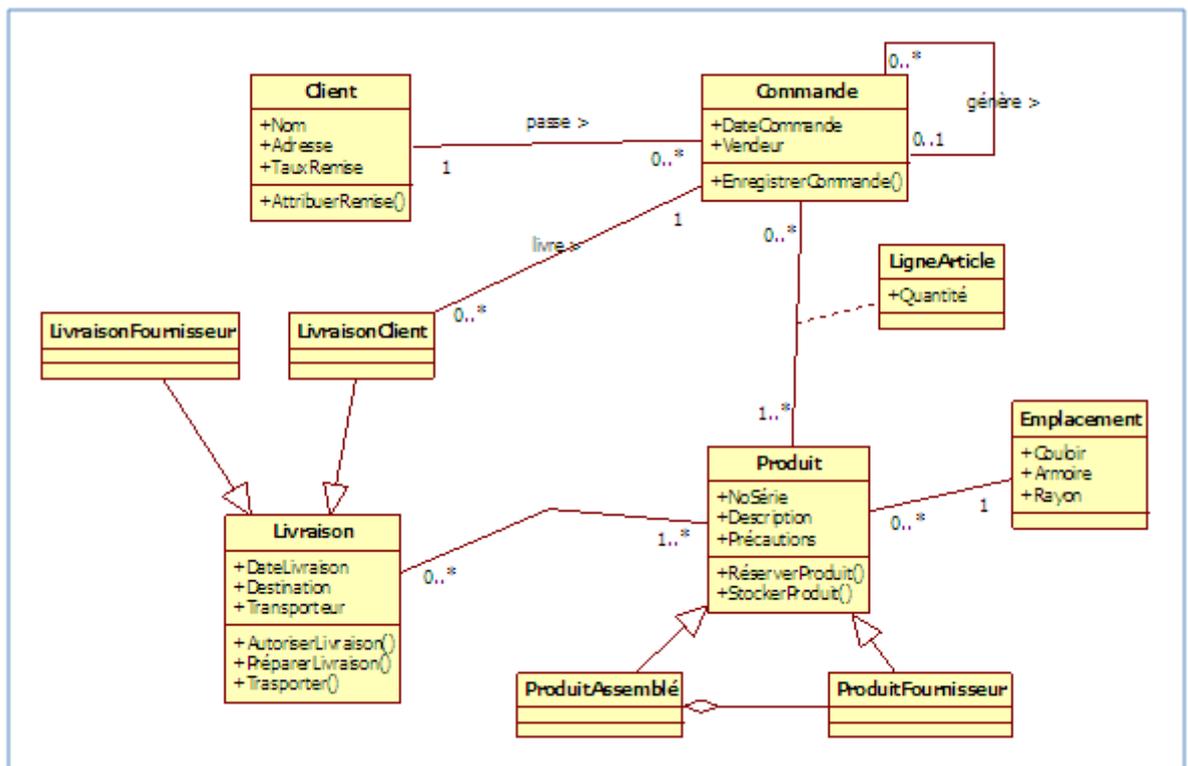
❖ **Vue logique (2)**

- **La vue logique** a pour but d'identifier les éléments du domaine, les relations et interactions entre ces éléments. Cette vue organise les éléments du domaine en « catégories ». Deux diagrammes peuvent être utilisés pour cette vue.

➤ **Le diagramme de classes**

Dans la phase d'analyse, ce diagramme représente les entités (des informations) manipulées par les utilisateurs.

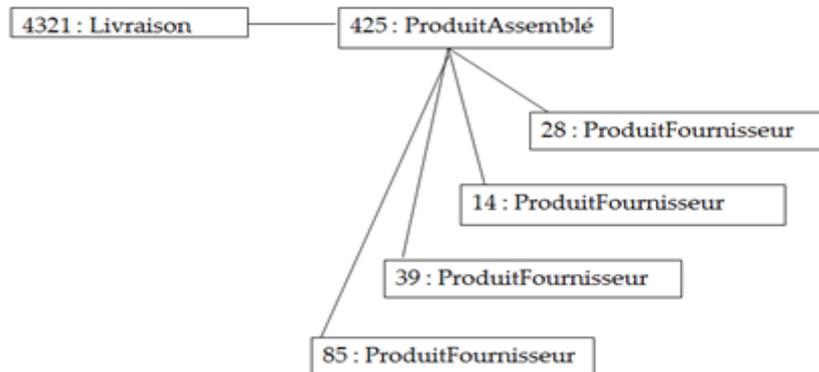
Dans la phase de conception, il représente la structure objet d'un développement orienté objet.



Un exemple de diagramme de classes (utilisé en phase d'analyse)

- **Le diagramme d'objets** sert à illustrer les classes complexes en utilisant des exemples d'instances.

Une instance est un exemple concret de contenu d'une classe. En illustrant une partie des classes avec des exemples (grâce à un diagramme d'objets), on arrive à voir un peu plus clairement les liens nécessaires.



Un exemple de diagramme d'objet

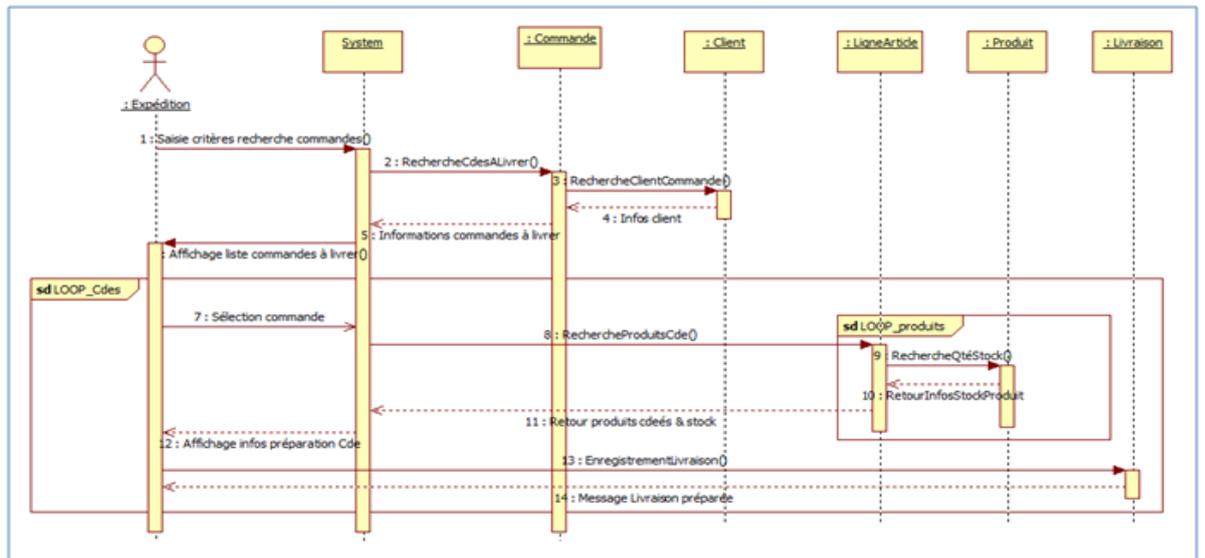
### ❖ Vue des processus (3)

La vue des processus démontre :

- la décomposition du système en processus et actions ;
- les interactions entre les processus ;
- la synchronisation et la communication des activités parallèles.

La vue des processus s'appuie sur plusieurs diagrammes.

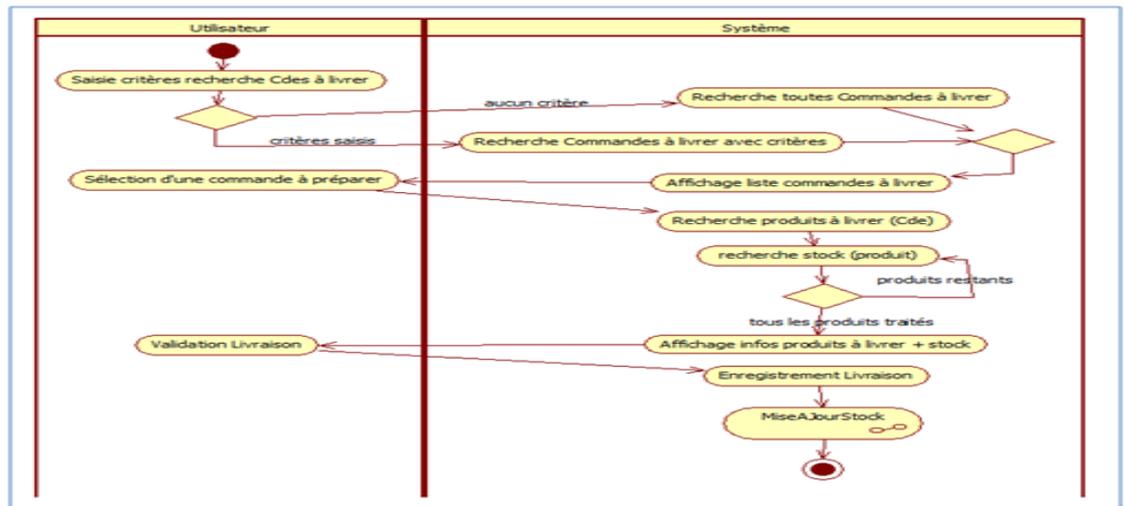
- Le **diagramme de séquence** permet de décrire les différents scénarios d'utilisation du système.



- 

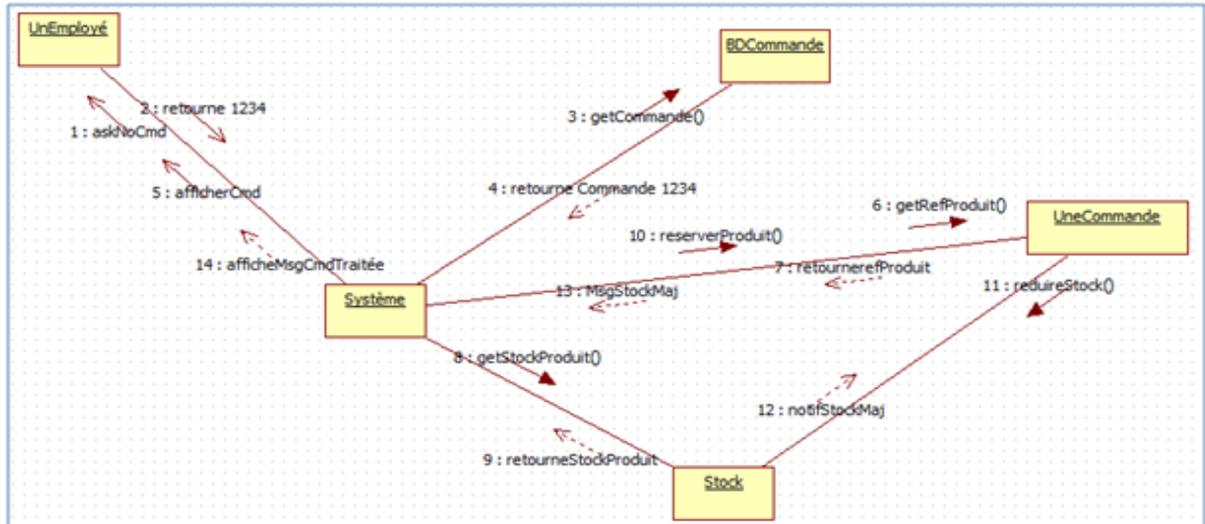
**Un exemple de diagramme de séquence**

- Le **diagramme d'activité** représente le déroulement des actions, sans utiliser les objets. En phase d'analyse, il est utilisé pour consolider les spécifications d'un cas d'utilisation.



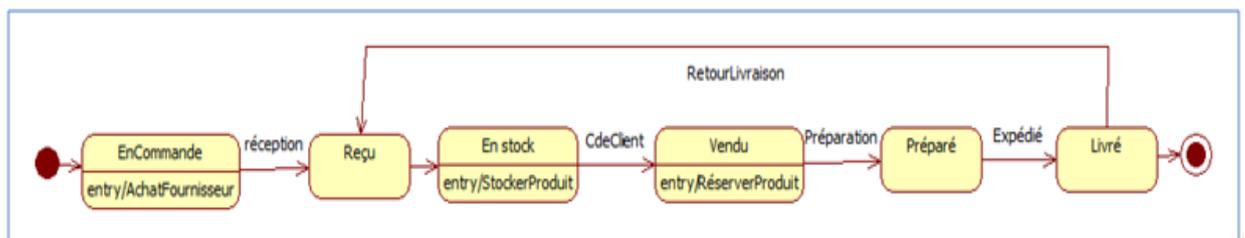
**Un exemple de diagramme d'activité**

- **Le diagramme de collaboration** (appelé également diagramme de communication) permet de mettre en évidence les échanges de messages entre objets. Cela nous aide à voir clair dans les actions qui sont nécessaires pour produire ces échanges de messages. Et donc de compléter, si besoin, les diagrammes de séquence et de classes.



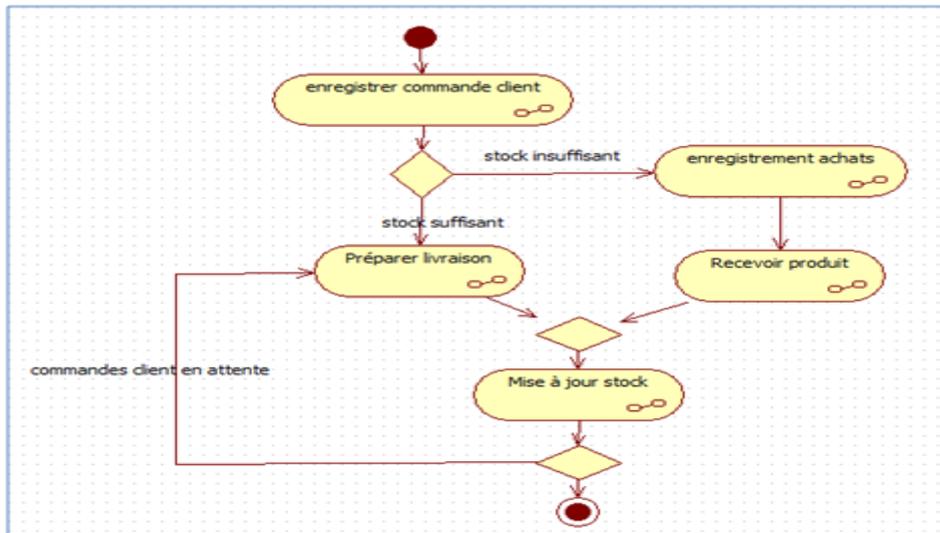
Un exemple de diagramme de collaboration (de communication)

- **Le diagramme d'état-transition** permet de décrire le cycle de vie des objets d'une class



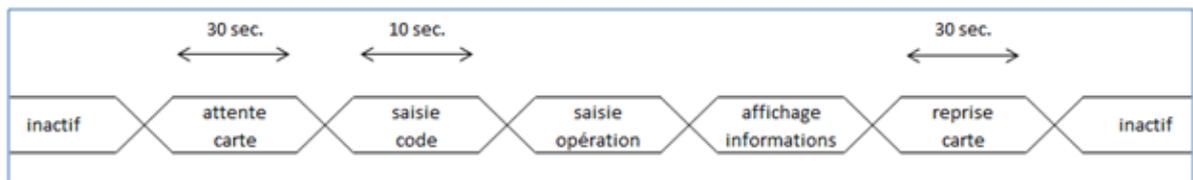
Un exemple de diagramme d'état-transition (objets de la classe produit)

- **Le diagramme global d'interaction** permet de donner une vue d'ensemble des interactions du système. Il est réalisé avec le même graphisme que le diagramme d'activité. Chaque élément du diagramme peut ensuite être détaillé à l'aide d'un diagramme de séquence ou d'un diagramme d'activité. Ce diagramme ne sera pas étudié dans ce projet.

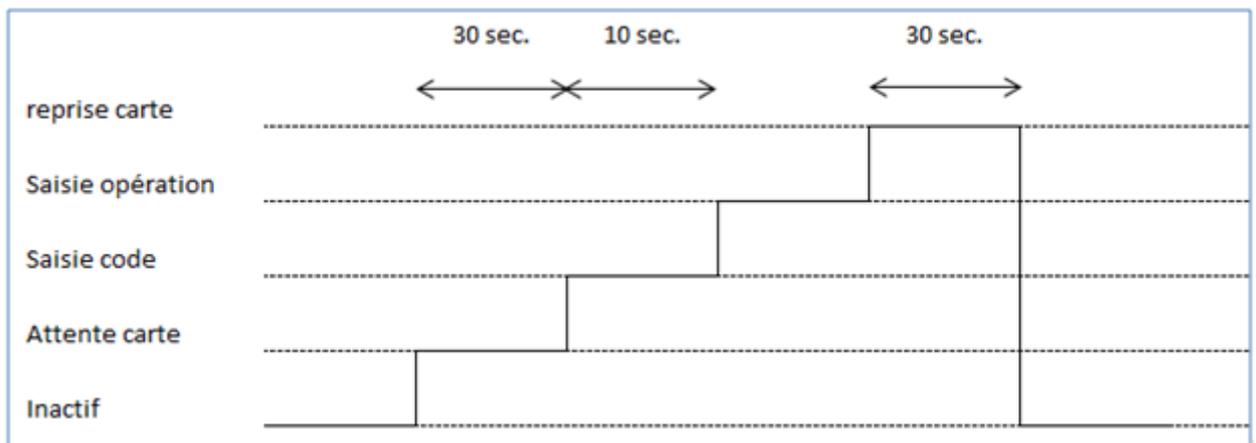


Un exemple de diagramme global d'interaction

- **Le diagramme de temps** est destiné à l'analyse et la conception de systèmes ayant des contraintes temps-réel. Il s'agit là de décrire les interactions entre objets avec des contraintes temporelles fortes. Ce diagramme ne sera pas étudié dans ce projet.



Un exemple de diagramme de temps avec un seul axe temporel



Un exemple de diagramme de temps avec un axe temporel par état

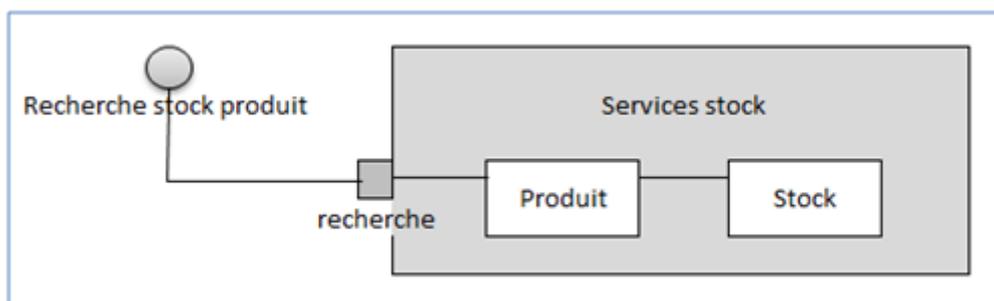
### I.4.3.3 L'aspect lié à l'architecture du logiciel :

Pour rappel, cette partie du schéma 4+1 vue permet de définir les composants à utiliser (exécutables, interfaces, base de données, bibliothèques de fonctions, etc.) et les matériels sur lesquels les composants seront déployés.

#### ❖ Vue des composants (4)

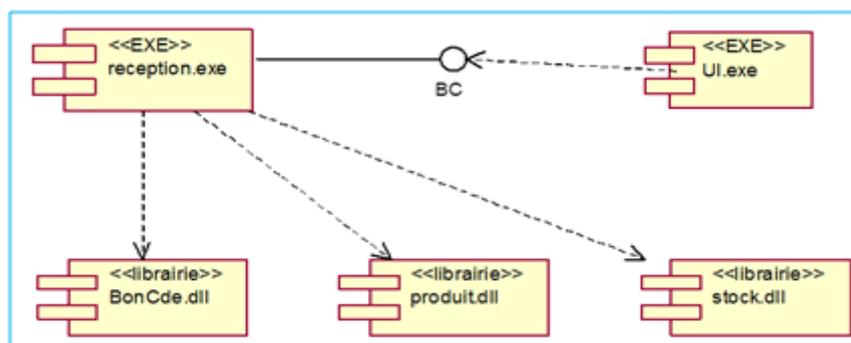
La **vue des composants** (vue de réalisation) met en évidence les différentes parties qui composeront le futur système (fichiers sources, bibliothèques, bases de données, exécutables, etc.). Cette vue comprend deux diagrammes.

- Le **diagramme de structure composite** décrit un objet complexe lors de son exécution. Ce diagramme ne sera pas étudié dans ce projet.



#### Un exemple de diagramme de structure composite

- Le **diagramme de composants** décrit tous les composants utiles à l'exécution du système (applications, bibliothèques, instances de base de données, exécutables, etc.). Ce diagramme ne sera pas étudié dans ce projet.

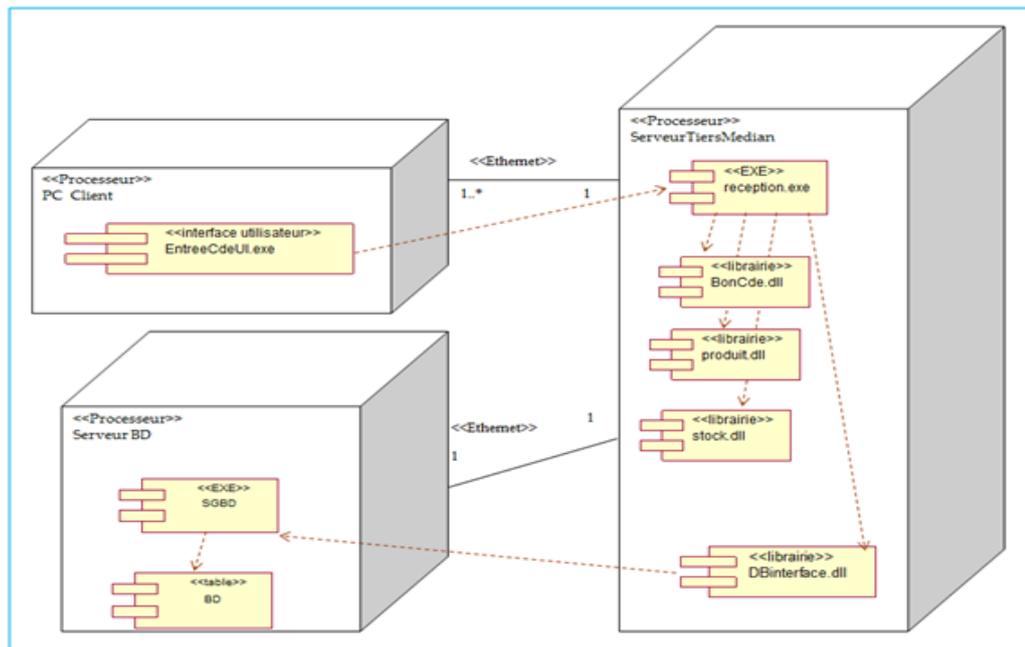


#### Un exemple de diagramme de composants

❖ **Vue de déploiement (5)**

La **vue de déploiement** décrit les ressources matérielles et la répartition des parties du logiciel sur ces éléments. Il contient un diagramme :

- **Le diagramme de déploiement** correspond à la description de l'environnement d'exécution du système (matériel, réseau...) et de la façon dont les composants y sont installés. Ce diagramme ne sera pas étudié dans ce projet.



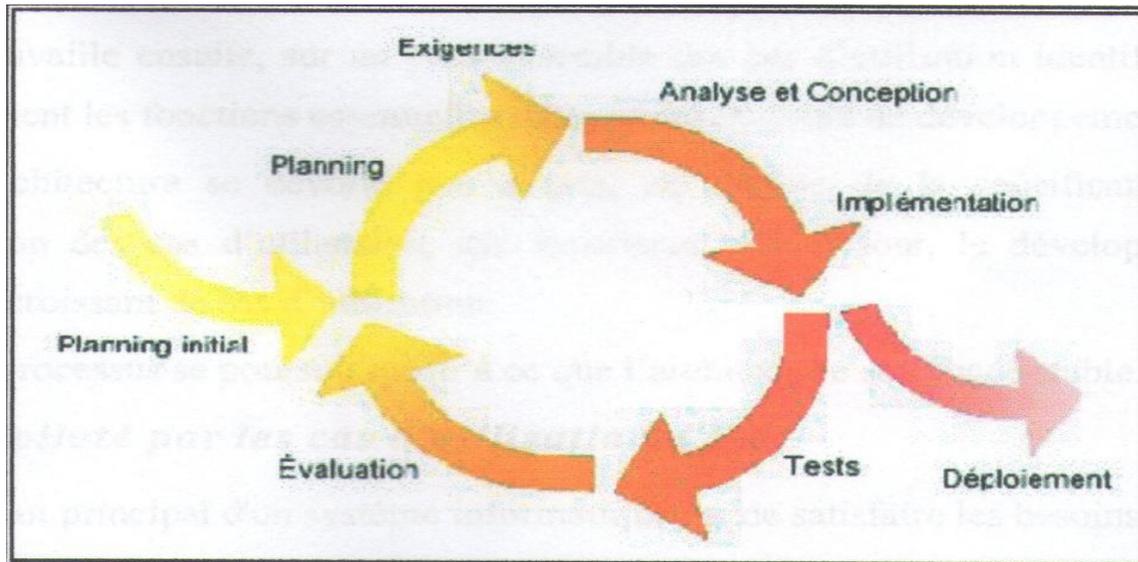
Un exemple de diagramme de déploiement

**I.5 Processus UP :**

**I.5.1 Définition du processus unifié :**

Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

C'est un patron de processus pouvant être adaptée à une large classe de système logiciel, A différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles d'entreprise.



Déroulement du processus UP

### I.5.2 Introduction :

Complètement idéal d'UML, le processus unifié (UP pour unified process) est le Processus de développement logiciel né de la fusion des travaux d'Ivar Jacobson, Grady Booch et de James Rumbaugh, les trois concepteurs du langage UML. Fruit des meilleures pratiques de l'ingénierie logicielle, le processus unifié décrit les activités qui permettent de traduire les besoins d'un utilisateur en un système logiciel.

Le processus unifié est décrit par ses auteurs comme une méthode pilotée par les cas d'utilisation, centrée sur l'architecture, itérative et incrémentale.

Discipline d'UP :

Les disciplines permettant de regrouper les différentes activités du PU en six disciplines en ingénierie (workflows du processus), directement liés au processus de développement, et trois disciplines de support (workflow de soutien).

### I.5.3 Les neuf disciplines d'UP sont :

#### 1- La modélisation du métier :

Il s'agit d'identifier les acteurs, les processus métier et la vision métier que le projet doit implémenter.

#### 2-La gestion exigences :

Basée sur les cas d'utilisation, est une discipline essentielle d'UP qui vise à capturer le plus fidèlement possible les exigences des utilisateurs de projet.

**3- Analyse de conception :**

Ont pour les objets de traduire les cas d'utilisation en autant de vues de l'architecture du logiciel que nécessaire. Cette discipline peut exploiter la palette des diagrammes UML Pour la production de ces différents vus.

**4 - L'implémentation :**

Visent le triple objectif de raffiner le modèle de conception, générer le code source et les tests unitaire associés, intégrer le travail d'implémentation des différents équipe travaillent en parallèle.

**5 - les tests :**

Jouent un rôle central dans le PU. Ils sont continuellement mis en œuvre pendant le processus de développement. Ils doivent également comprendre des tests de non régression essentielle dans tout processus incrémental.

**6 - Le déploiement :**

Prend en charge les activités de configuration et de conditionnement de système à livre.

**7 - La gestion de la configuration et des changements :**

Cette discipline utilisera avantageusement des systèmes de gestion de version comme de cvs (concurrent version system) par exemple.

**8 - La gestion de projet :**

Est une discipline critique du PU qui la responsable de planifier et de piloter l'ensemble des activités du projet.

**9 - L'environnement :**

Est la discipline responsable de la logistique au sens large du projet : normes, standards, environnement de développement, infrastructure matérielle....

**I.5.4 Phase du processus unifié :**

Les itérations d'UP s'inscrivent dans quatre phases successives dont la validation constitue des jalons importants du processus de développement :

- **Initialisation** : La Phase d'initialisation a pour objectif de trouver un compromis entre les exigences et contraintes.
- **Elaboration** : Cette phase devrait également à conduire à une révision et une précision du planning du projet.

**I.5.5 Construction :** Dans cette phase, il faut essayer de capturer tous les besoins restants car il n'est pratiquement plus possible de le faire dans la prochaine phase. Ensuite, continuer l'analyse, la conception et surtout l'implémentation de tous les cas d'utilisation. A la fin de cette phase, les développeurs doivent fournir une version exécutable du système.

### **I.6 Conclusion :**

A l'issu de ce chapitre, nous nous sommes familiarisé avec l'outil de modélisation UML d'une part et d'autre part, avec la démarche du processus de développement logiciel (Processus Unifié) qui va nous guider dans la réalisation des prochaines étapes de notre projet.

# **Chapitre II**

**L'analyse  
&**

**La conception Du Système**

**II 1-Introduction :**

L'étape de l'analyse des besoins est la deuxième phase de cycle de vie du Processus Unifié et l'une des étapes les plus importantes à considérer ; en effet si les besoins sont mal spécifiés et exprimés, ou mal analysés, toute la suite devra être refaite, d'où l'importance accordée à cette activité.

Notre objectif dans cette étape est donc d'exprimer les besoins attendus du futur système à développer. En va le détailler en trois étapes :

- Spécifier les besoins de notre système en définissant le diagramme de cas d'utilisation.
- Le diagramme de cas d'utilisation va être détaillé en plusieurs diagrammes de séquences
- Décrire la structure statique de notre système par le diagramme de classe.

**II .2 Modélisation du système :**

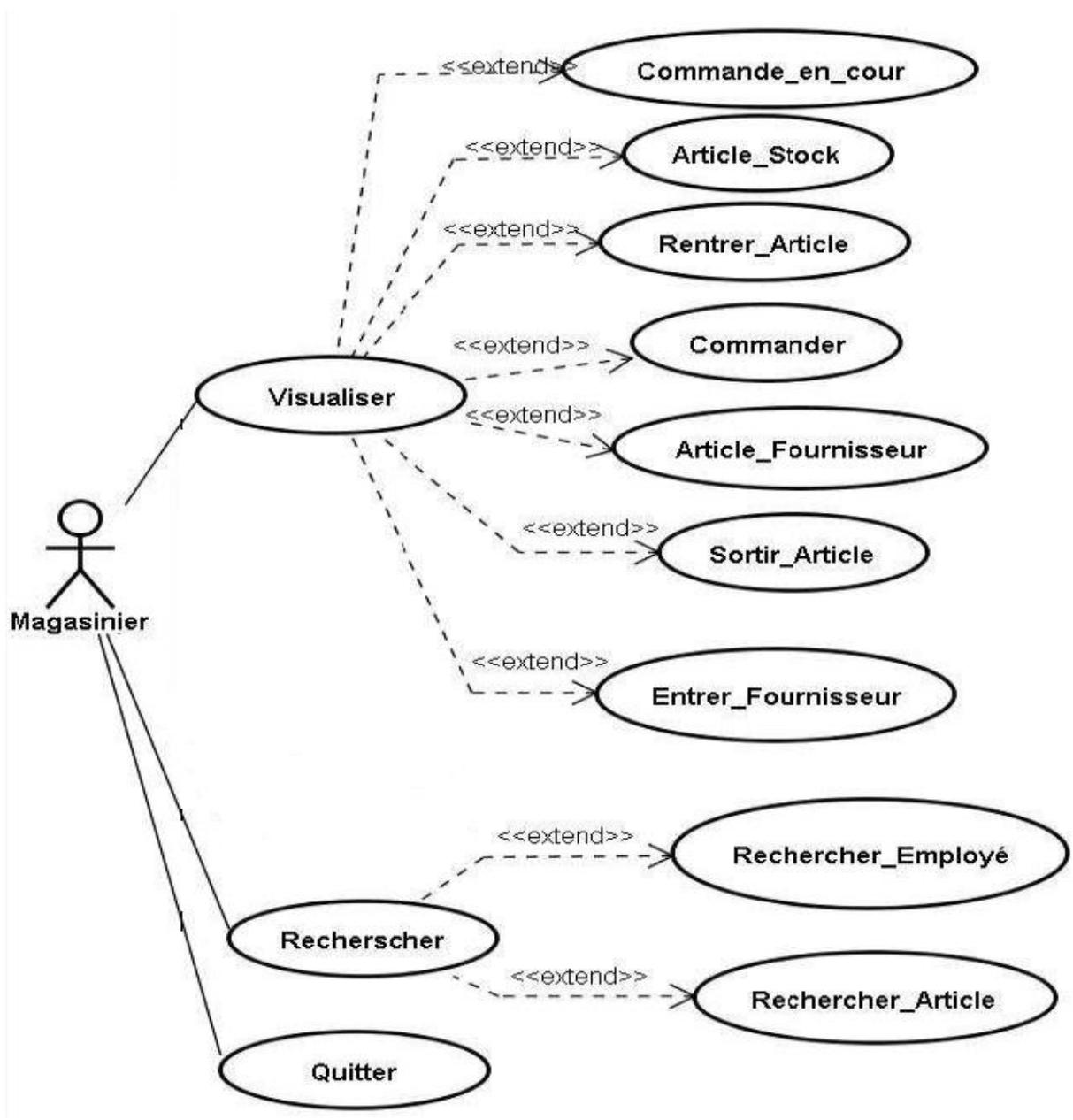
Le système à concevoir est régi par une application, permettant ainsi un accès aux différents services souhaitant solliciter le magasin.

De ce fait, les acteurs interagissant avec l'application sont :

- **Le magasinier:** l'administrateur de l'application.
- **Les services demandeurs:** utilisateurs ayant un accès limité à l'application.

Les principales fonctionnalités de l'application à concevoir sont érigées autour des besoins et exigences des différents acteurs. Elles sont illustrées dans le diagramme du cas d'utilisation de la gestion du stock suivant :

a-Diagramme de cas d'utilisation :



b- Les diagrammes de séquence

❖ **Authentifier** : Permet à un acteur de s'authentifier avant d'accéder à l'application

Le système vérifie le nom d'utilisateur et le mot de passe.

- Si les informations ne sont pas correctes le système affiche un message d'erreur sinon le système affiche le menu de l'application.

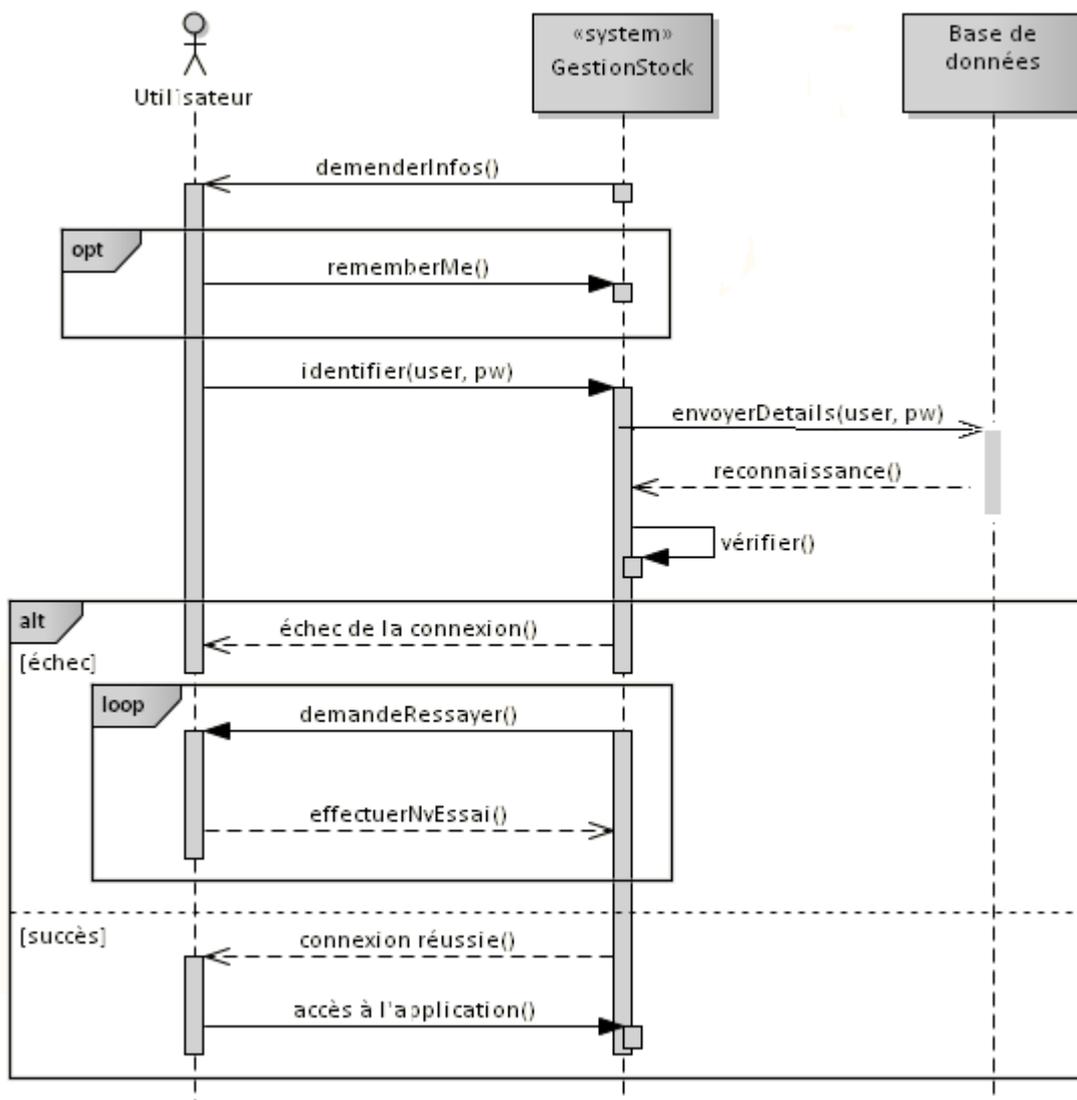


Diagramme de séquence(Authentication)

- ❖ **Consulter le stock:** Permet aux différents acteurs d'avoir une consultation sur les différentes fournitures du magasin.
- ❖ **Traitement des bons de commande :** Donne la possibilité aux services demandeurs d'exprimer leurs besoins envers le magasinier.

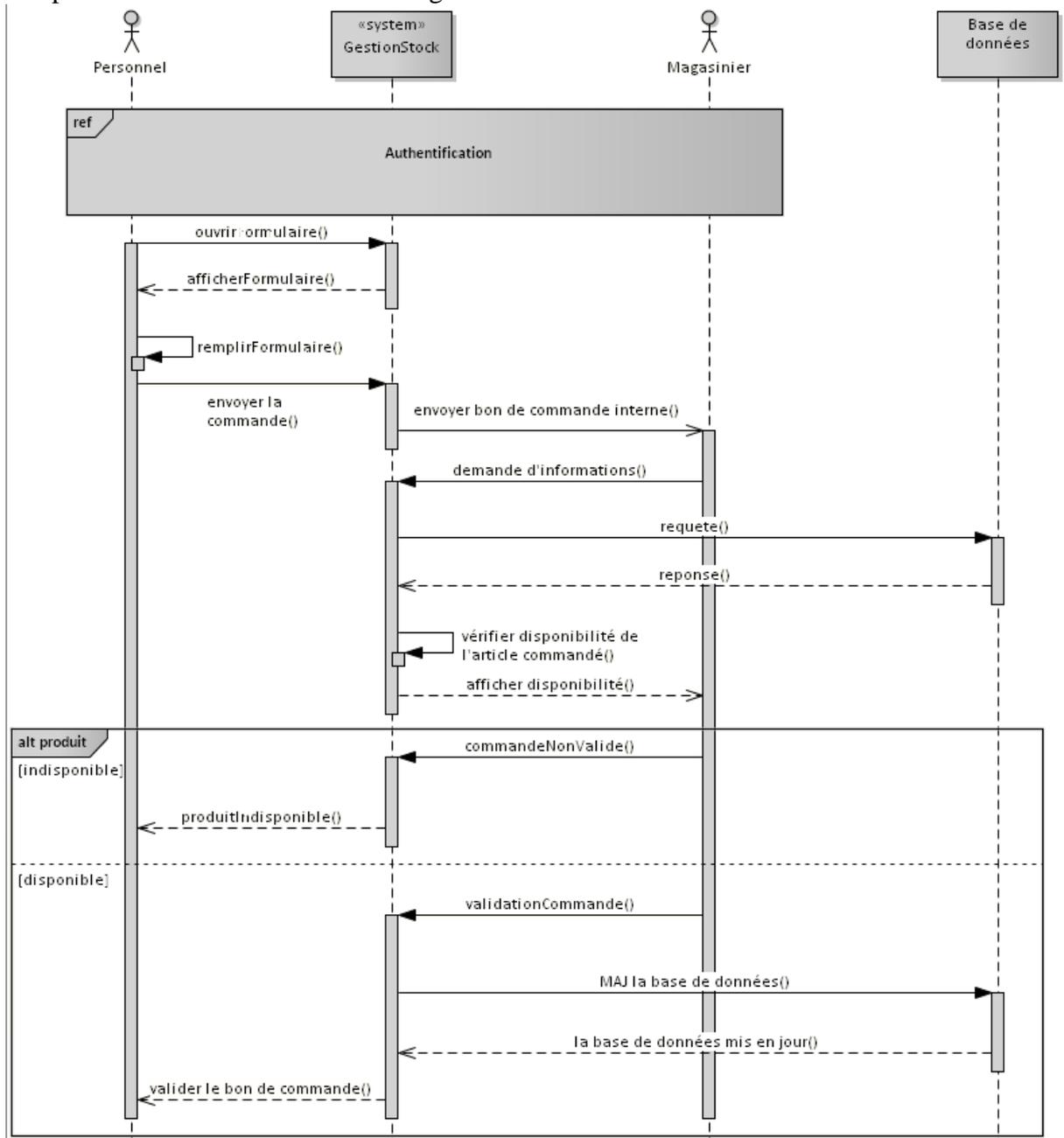
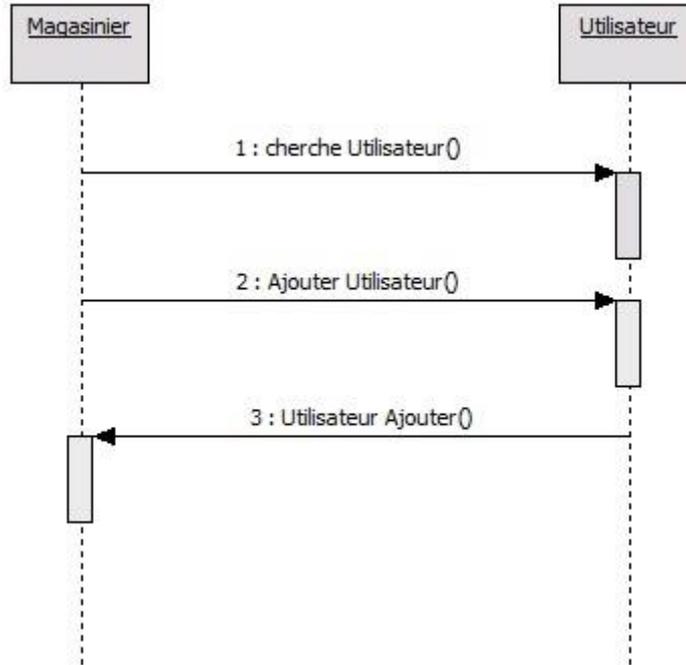


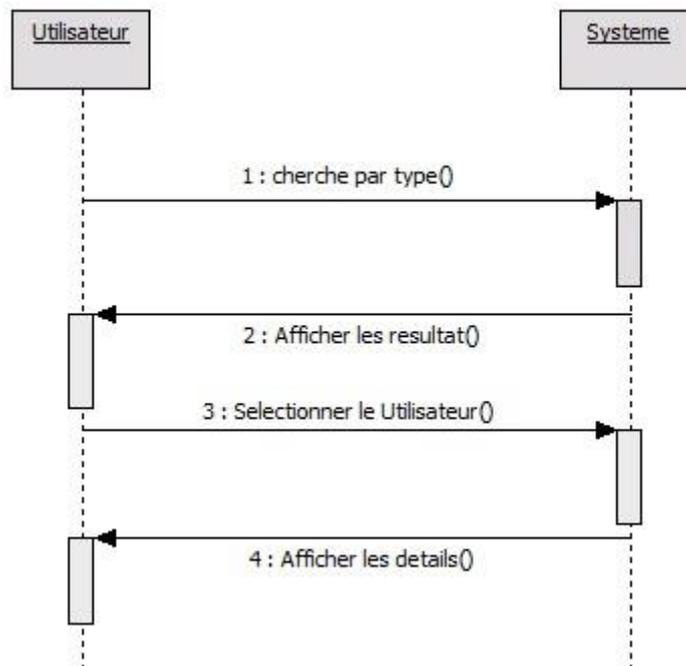
Diagramme de séquence Traitement des bons de commande

❖ Gérer les utilisateurs

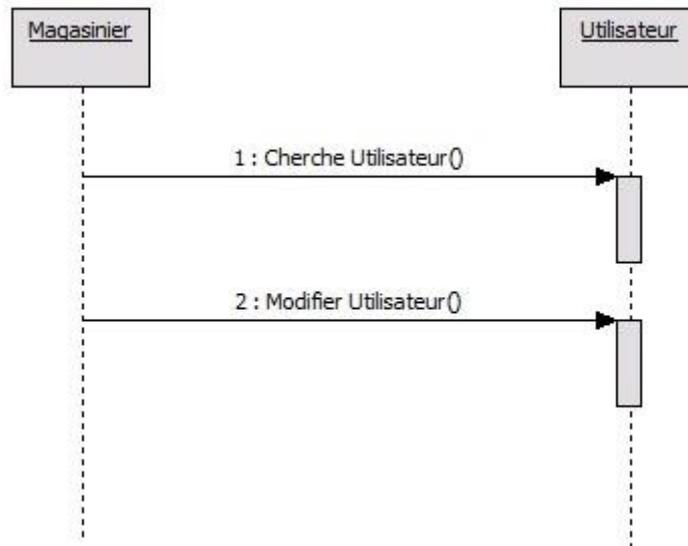
- **Ajouter Utilisateur** : Le système permet au magasinier d'ajouter des utilisateur de l'application



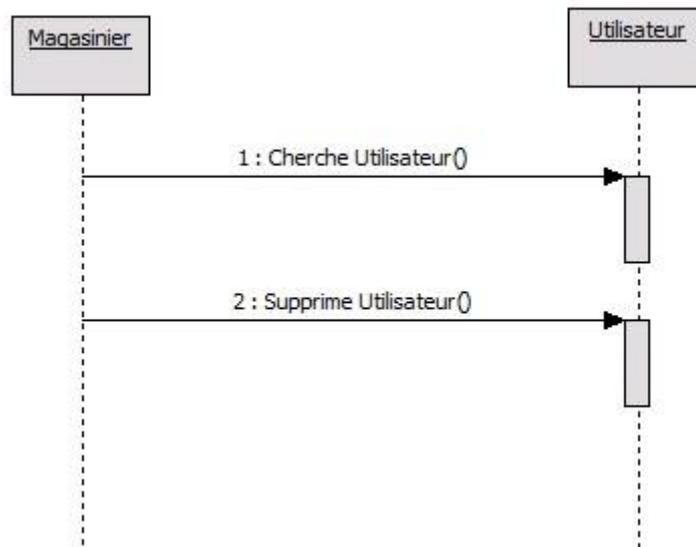
- **Rechercher utilisateur**: Permet de chercher un utilisateur grée l'application,



- **Modifier Utilisateur :** Le système offre la possibilité de modifier les informations des utilisateur

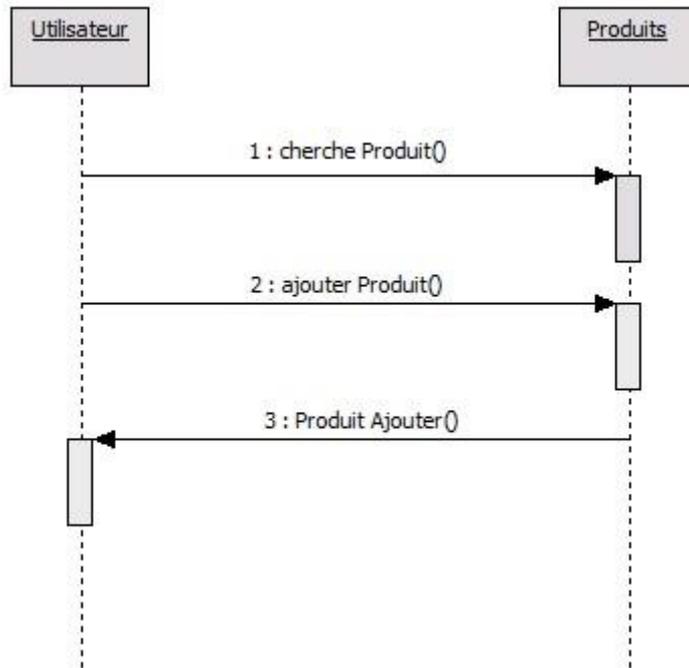


- **Supprimer Utilisateur :** Le système offre la possibilité au magasinier de supprimer les différentes entités créés.

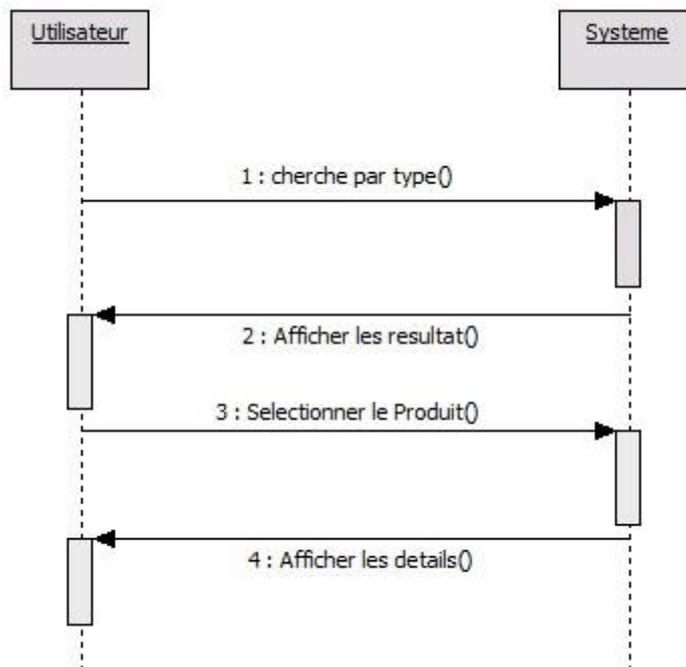


❖ **Gérer les Produits**

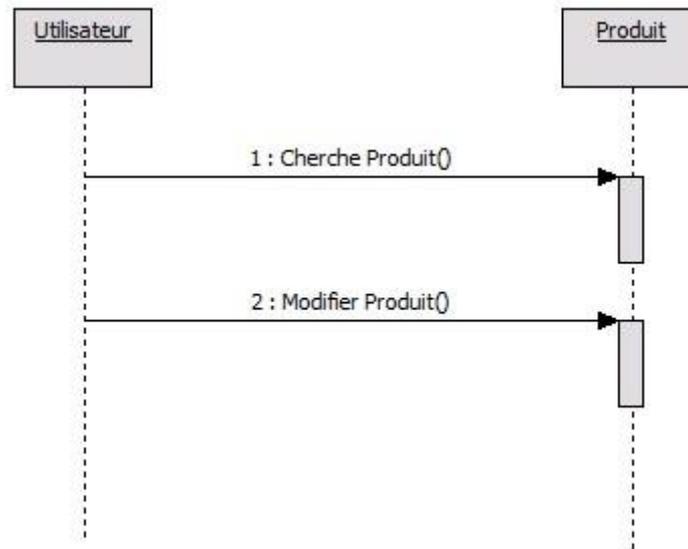
- **Ajouter produit** : Le système permet au magasinier d'ajouter des produits dans le stock.



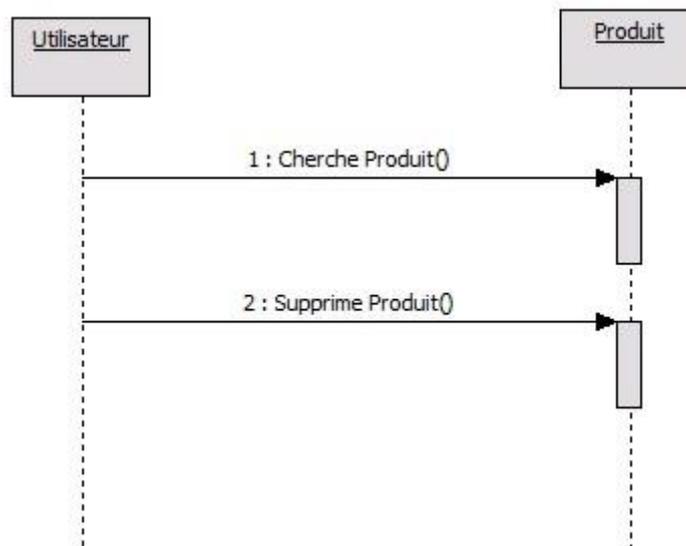
- **Rechercher produit** : Permet de chercher un produit dans le stock.



- **Modifier produit** : Le système offre la possibilité de modifier les informations des Produits

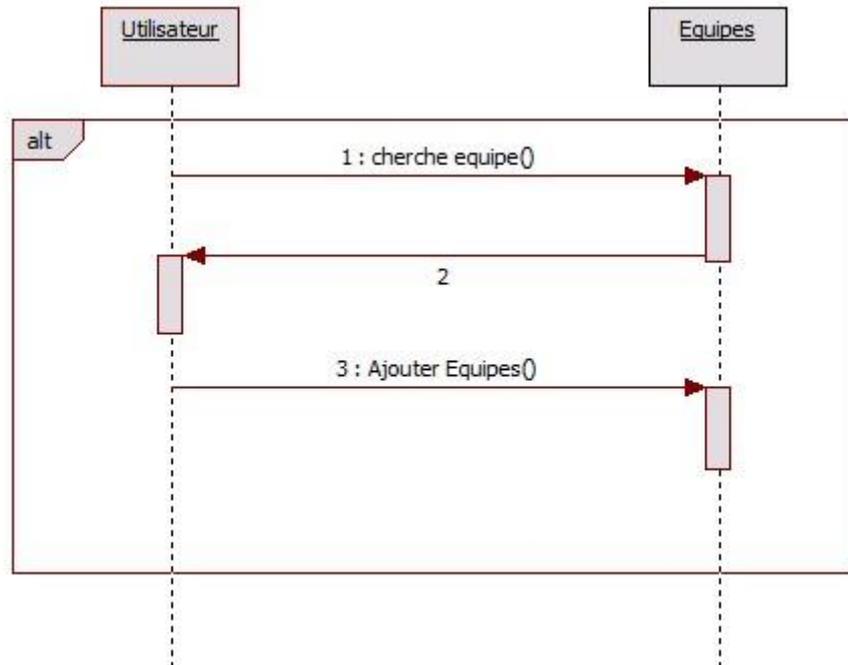


- **Supprimer produit** : Le système offre la possibilité au magasinier de supprimer les différentes entités créées.

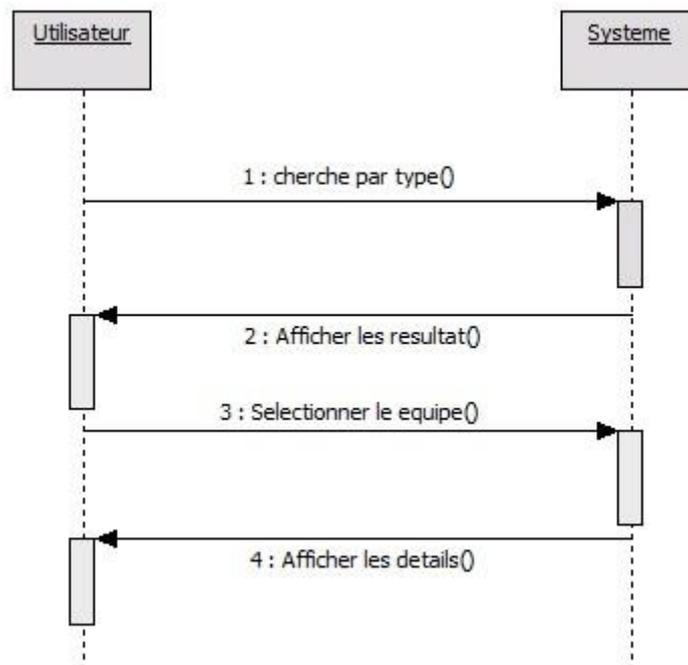


❖ Gérer les Equipes

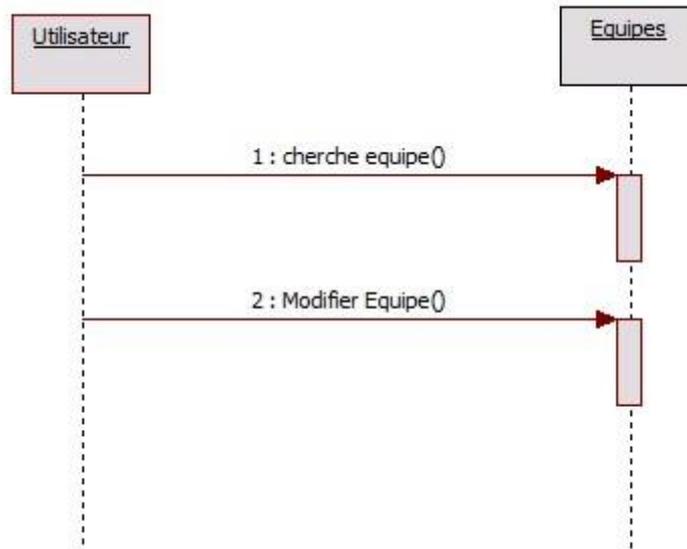
- Ajouter Equipes : Le système permet au magasinier d'ajouter des équipes



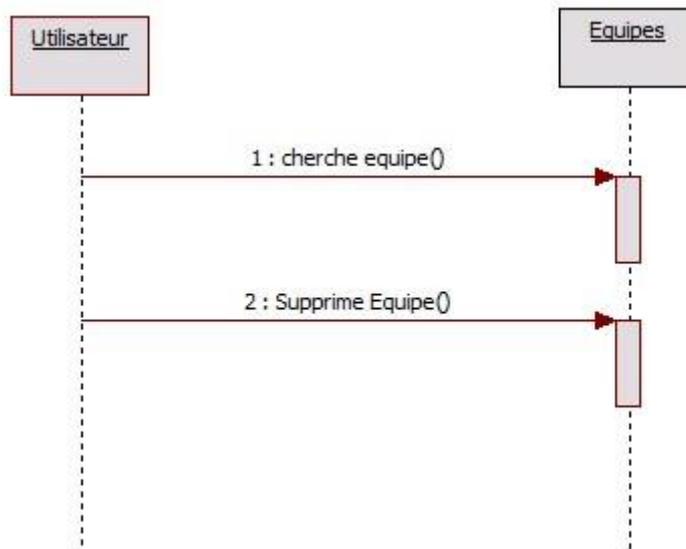
- Rechercher Equipes : Permet de chercher un équipe



- **Modifier Equipes** : Le système offre la possibilité de modifier les informations des Equipes

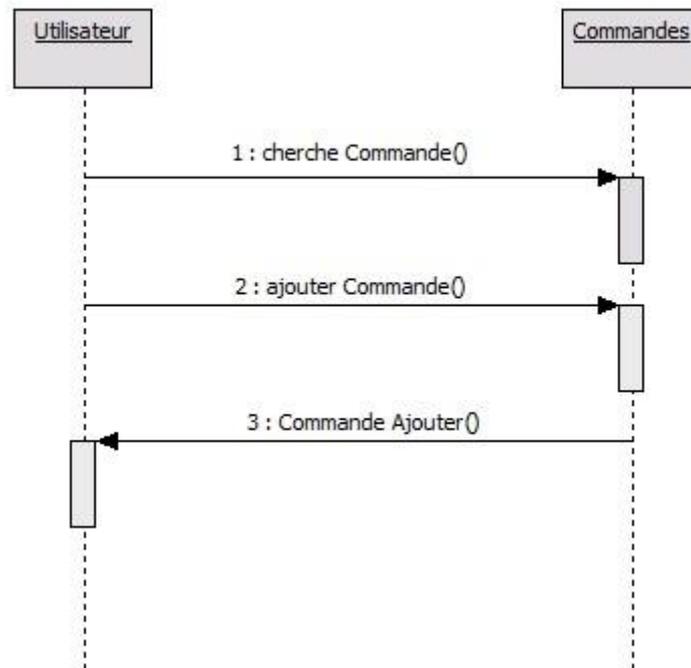


**Supprimer Equipes** Le système offre la possibilité au magasinier de supprimer les différentes entités créés.

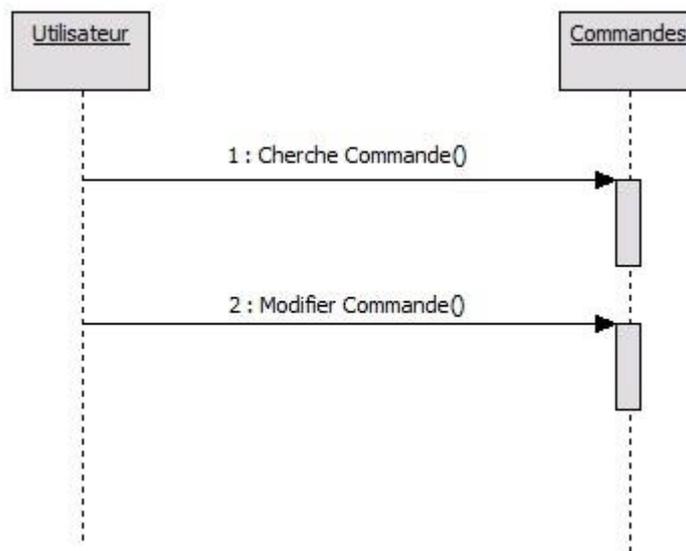


❖ Gérer les Commandes

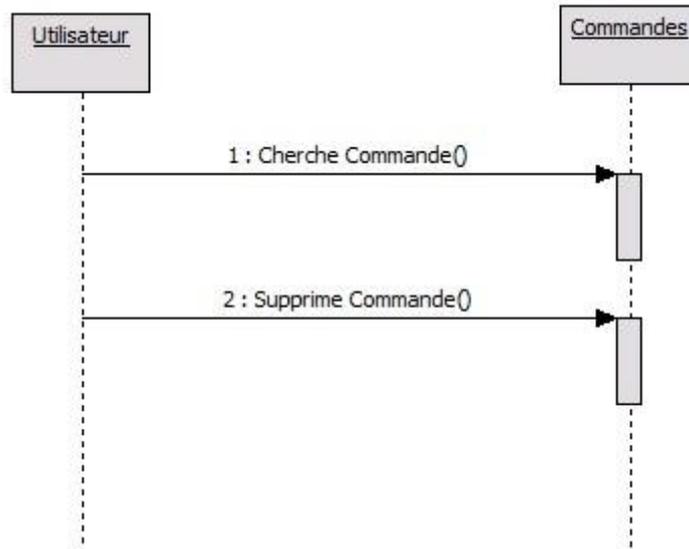
- **Ajouter Commande** : Le système permet au magasinier d'ajouter des Commande dans le bon commande.



- **Modifier Commande** : Le système offre la possibilité de modifier les informations des Commandes.



- **Supprimer Commande :** Le système offre la possibilité au magasinier de supprimer les différentes entités créés.





# **Chapitre III**

## **L'Implémentation du Système**

### **III .1 Introduction**

Dans la démarche de Processus Unifié, la phase de conception suit immédiatement la phase d'Analyse, par ailleurs la conception de logiciel est un art qui nécessite de l'expérience, et elle consiste à traduire les besoins en spécifiant comment l'application pourra les satisfaire avant de procéder à sa réalisation. En effet, dans ce chapitre nous essayons d'étendre la représentation des diagrammes effectués au niveau de l'analyse en y intégrant les aspects techniques plus proches des préoccupations physiques.

### **III .2 Choix du langage de programmation :**

Pour réaliser notre application, nous avons utilisé le langage de programmation JAVA dédié à la création des applications, celui-ci nous l'avons manipulé dans un environnement de développement intitulé NetBeans IDE 8.0.2, qui est largement compatible avec JAVA.

Par ailleurs, il faut noter que La plate forme que nous avons adoptée est WampServeur version 2.5 qui inclut tous les outils nécessaires, Apache-2.4.9-Mysql-5.6.17-php5.5.12.

### **III .3 Objectifs de la programmation objet :**

- Lier les données et les fonctions qui les manipulent afin d'éviter des accès aux données par des fonctions non autorisées.
- Obtenir une meilleure abstraction en cachant l'implémentation des techniques utilisées et en ne rendant visible que des points d'entrée. Ainsi, si l'implémentation change, le code utilisateur n'est pas affecté.
- Réutiliser l'existant dans un souci de productivité.
- Traiter les erreurs localement au niveau des objets sans que cela ne perturbe les autres parties du programme.
- Faciliter la maintenance.

### **III .4 Outils de développement**

#### **a\_langage de programmation JAVA :**

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems aujourd'hui racheté par Oracle

Une de ses plus grandes forces est son

excellente portabilité : une fois votre programme créé, il fonctionnera automatiquement sous (Windows, Mac, Linux, etc.....) grâce à la machine virtuelle. On peut faire de nombreux types de programmes avec Java : Des applications, sous forme de fenêtre ou de console. Des applets, qui sont des programmes Java incorporés à des pages Web. Des applications pour appareils mobiles, comme les Smartphones, avec J2ME (Java2 Micro Edition). Des sites web dynamiques, avec J2EE (Java 2 Enterprise Edition, maintenant JEE) ; Et bien d'autres : JMF (Java Media Framework), J3D pour la 3D. Comme on peut le voir, Java permet de réaliser une très grande quantité d'applications différentes. On peut distinguer trois grandes phases dans la vie d'un code Java : la phase d'écriture du code source, en langage Java. la phase de compilation du code. la phase d'exécution. Ces phases sont les mêmes pour la plupart des langages compilés (C, C++. . .). Par contre, ce qui fait la particularité de Java, c'est que le résultat de la compilation n'est pas directement utilisable par l'ordinateur. Les langages mentionnés ci-dessus permettent de faire des programmes directement compréhensibles par votre machine après compilation, mais avec Java, c'est légèrement différent. Avec Java, c'est un programme appelé la machine virtuelle qui va se charger de retranscrire le résultat de la compilation en langage machine, interprétable par celle-ci. Nous n'avons pas à nous préoccuper des spécificités de la machine qui va exécuter notre programme, la machine virtuelle Java s'en charge pour nous.

### **b\_WampServer 2.5 :**

WampServer 2.5 est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi qu'une administration pour les deux bases SQL PhpMyAdmin et SQLiteManager.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray-icon (icône près de l'horloge de Windows).

### **c\_NetBeans :**

est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Développment and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons Conçu en Java,

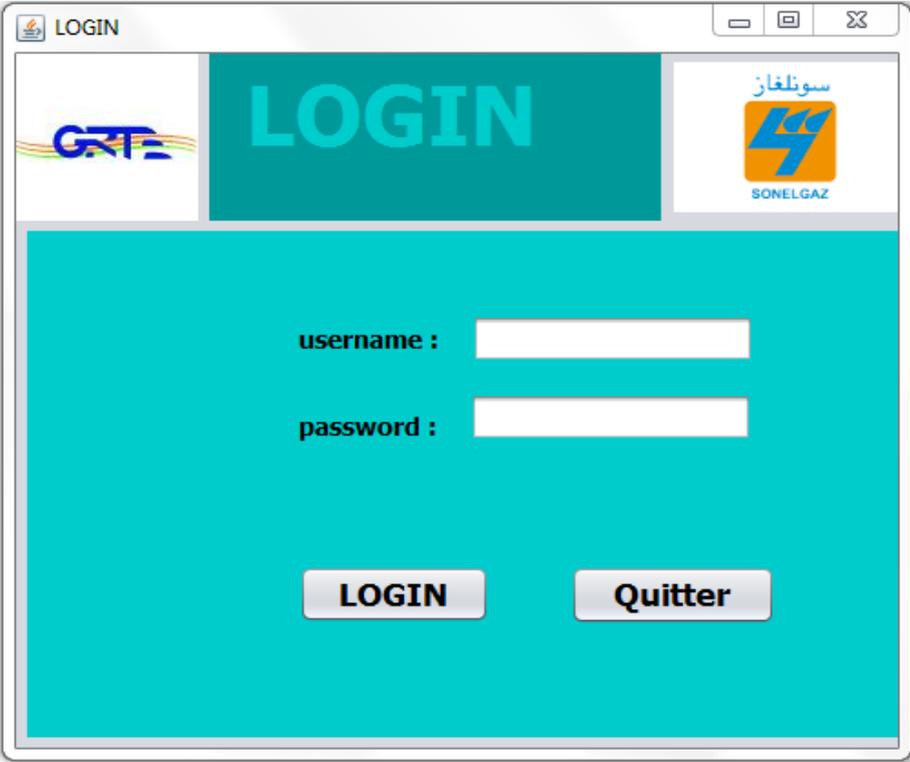
NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Développment KitJDK est requis pour les développements en Java. NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

### III .5 Description de l'application

Dans ce qui suit, nous allons présenter quelques interfaces de notre application gestion de stock.

#### III .5 .1 Page d'accueil

On retrouve l'interface d'authentification permettant aux différents utilisateurs d'accéder à leurs sessions. Il devra saisir le nom d'utilisateur, le mot de passe.



The image shows a Java Swing window titled "LOGIN". The window has a teal header bar with the word "LOGIN" in large, white, bold letters. On the left side of the header is the GTE logo, and on the right side is the SONELGAZ logo, which includes the text "سونلغاز" in Arabic script above a stylized lightning bolt icon and the text "SONELGAZ" below it. The main content area of the window is teal and contains two white input fields. The first field is labeled "username :" and the second is labeled "password :". Below these fields are two buttons: "LOGIN" and "Quitter". The window has standard window controls (minimize, maximize, close) in the top right corner.

\*

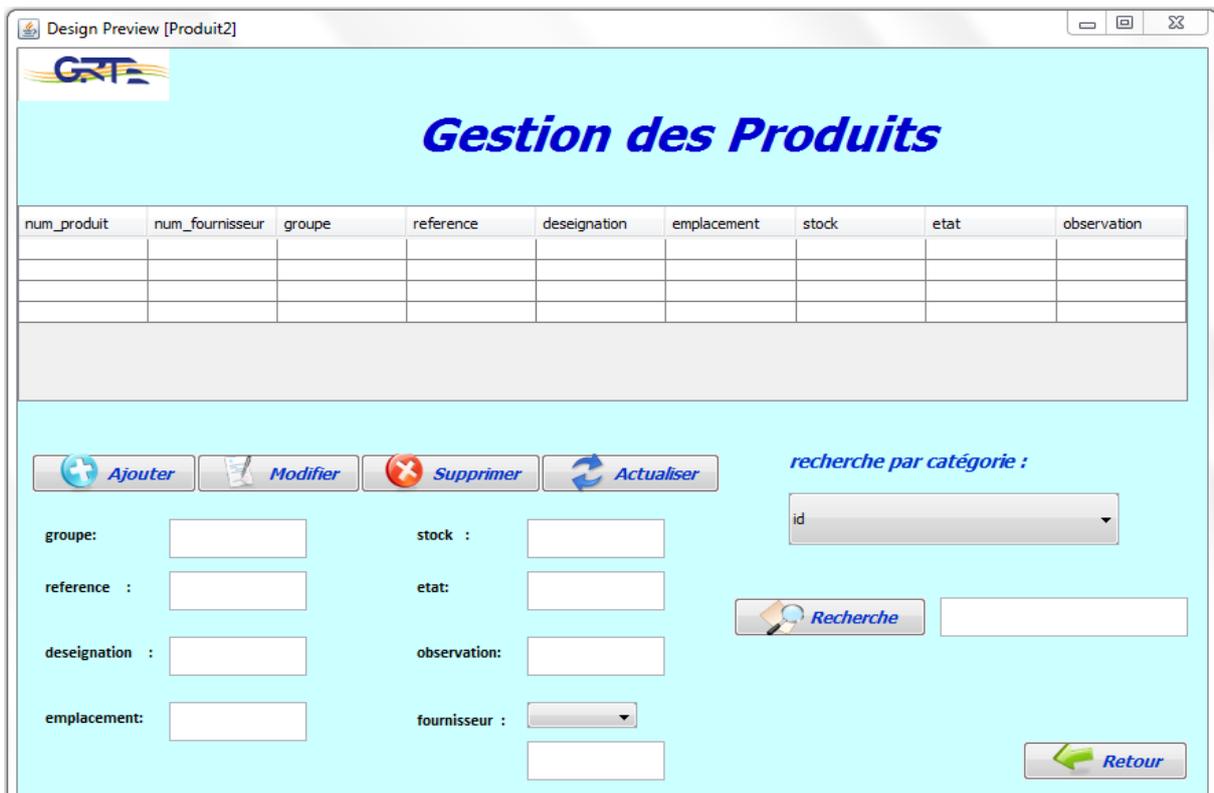
### III .5 .2 Menu principal

Cette interface permet au superviseur d'avoir un contrôle total sur l'application, lui permettant ainsi d'accéder à toutes ses fonctionnalités.



### III .5 .3 Formulaire Produit :

Cette interface permet au superviseur d'éditer une fiche de stock pour un produit bien spécifique.



### III .5 .4 Formulaire Utilisateur :

Cette interface permet au superviseur d'éditer une fiche d'un nouvel utilisateur de l'application.

**Gestion des Utilisateur**

id	username	password	type

**id\_user :**   
**username :**   
**password :**   
**type :**

**recherche par catégorer :**

### III .5 .5 Formulaire de Bon de Commande :

Cette interface représente le traitement de Bon de Commande imprimé à l'équipe.

**Gestion des Commandes**

num_equipe	nom__equipe	membre

**nom d'equipes**   
**membres**

**recherche par catégorie :**

num_prd_comd	num_commande	designation	date	nom_equipe	quantite

**numéro de commande**  **designati...**   
**Date**   
**nom de equipe**  **quantite**

**recherche historique de equipe**  
**Nom d 'equipe**

**cherche la commande**

### III .5 .6 Formulaire Fournisseur :

Cette interface permet au superviseur d'éditer une fiche d'un nouvel fournisseur .

num_fournisseur	nom_fournisseur	adresse	TEL

Ajouter Modifier Supprimer actualiser

Nom Fournisseur:

Adresse:

TEL:

recherche par catégorer : nom\_fournisseur

recherche

Retour

### III .6 Conclusion :

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application Gestion de stock en spécifiant l'environnement de développement, l'implémentation de la base des données et la démarche suivie pour la réalisation. En effet, nous avons achevé l'implémentation et les tests de tous les cas d'utilisation, tout en respectant la conception élaborée. En d'autres termes, nous détenons la version finale du logiciel, installée dans notre environnement de développement. Ainsi que nous avons prévu la plate forme sous laquelle le système sera installé dans l'environnement des utilisateurs

## **Conclusion Générale**

Le travail qui nous a été confié consiste à réaliser une application de gestion de stock Du magasin de société GRTE.Spa, filiale du Groupe SONELGAZ, l'analyse, la spécification, la conception sont orientée objet et décrites en UML en se basant sur le processus UP. L'implémentation de l'application a été réalisée par l'environnement Netbeans Cette application va faciliter les activités de gestion de stock en suivant tous les dossiers.

Pendant la réalisation de notre travail nous avons acquis beaucoup de connaissances concernant le langage de modélisation UML et l'environnement Netbeans e nous avons souhaité d'avoir plus de temps pour mieux traiter le sujet proposé. mais nous espérons que notre travail sera évolué et amélioré par autre promotion et qu'il sera un aide pour eux.

# Références bibliographiques

[1] *N.ABDAT et L.MAHDAOUI, UML Outil du génie logiciel.*

[2] *Laurent AUDIBERT, 'UML 2', Institut Universitaire de Technologie de Villetaneuse - Département Informatique, Édition 2007-2008.*

[3] *Ivar Jakobson, Grady Boosh, James Rambaugh, Le processus unifié de développement logiciel, Editions Eyrolles, 1999.*

[4] *Ivar Jakobson, Manus Cristerson, Patrick Jonsson, Gunar Overgaard, Le génie logiciel orienté objet, Editions Addison Wesley, 2000.*

[5] *F. Juliard UML Unified Method Language, Journal Université de Bretagne Sud UFR SSI-IUP Vannes, 2001-2002.*

[6] *Joseph Gabay et David Gabay, Mise en oeuvre guidée avec études de cas, édition Dunod, Paris 2008.*

[7] *Nathalie Lopez, Jorge Migueis et Emmanuel Pichon, Intégrer UML dans vos projets, Editions Eyrolles, 2000.*

[9] *Pascal ROQUES, UML 2 par la pratique étude de cas et exercices corrigés, ÉDITIONS EYROLLES, Septembre 2006.*

[10] *Mémoire de fin d'études Licence en Informatique (2013/2014).*

## **Webliographie**

[www.commentcamarche.com/initiation/concept](http://www.commentcamarche.com/initiation/concept)

[www.Developpez.com](http://www.Developpez.com).

<http://fr.wikipedia.org>.

## **Résumé**

Aujourd'hui, l'informatique a atteint une prodigieuse évolution technologique dans différents domaines (réseaux informatiques, bases de données, le Web, etc.). Cette évolution est nécessaire pour remédier aux problèmes rencontrés dans la vie actuelle.

Le dynamisme est l'une des caractéristiques les plus essentielles de l'informatique. C'est ceci qui nous a poussés à créer une application web dynamique, accessible par des utilisateurs dans un réseau informatique que ce soit intranet ou local.

Chaque création nécessite une modélisation avec un langage universel bien spécifié tel qu'UML, la réalisation quant à elle nécessite des outils de développements bien adaptés au contexte de l'application. Pour les bases de données, l'utilisation d'un SGBD tel que MySQL, ORACLE, PostgreSQL, etc. est indispensable.

Notre travail consiste à concevoir une application en utilisant une base de données, pour la gestion des stocks de société GRTE.Spa, filiale du Groupe SONELGAZ, poste de TLEMCEN. L'application a été développée en utilisant différents logiciels informatiques tel que, WampServer, , etc. Le langage de programmation utilisé est le JAVA.

## **Abstract**

Today, data processing reached an extraordinary technological development in various fields (data-processing networks, databases, the Web, etc.). This evolution is necessary to cure the problems encountered in the current life.

The dynamism is one of the most essential characteristics of data processing. It is that which pushed us to create dynamic Web application, accessible by users in a data-processing network that it is Intranet or local Network.

Each creation requires a modeling with a universal language good specified such as UML, the realization as for it requires development tools adapted well to the context of the application. For the databases, the use of a DBMS such as MySQL, ORACLE, PostgreSQL, etc is essential.

Our work consists in conceiving a application by using a database for the inventory control of société GRTE.Spa, SONELGAZ, poste of TLEMCEN. The application was developed by using various data-processing software such as, WampServer., etc. The programming language used is the JAVA.