

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 :

Gestion D'un Magasin De Pièces De Rechange
GMPR
(Conception et réalisation d'une application)

Réalisé par :

- ❖ M^{elle} KADA Fouzia
- ❖ M^{elle} ZAHAF Ahlem

**Présenté le 10 juin 2014 devant la commission d'examination
composée par :**

- M.BENMOUNA Youcef (encadreur)
- M^{me} BENMANSOUR.F (examineur)
- M^{me} CHAOUCHE. L (examineur)

Année universitaire 2013-2014

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.

Nous tenons à exprimer nos vifs remerciements à Mr BENMOUNA. Y, notre encadreur ; c'est sous sa direction scientifique que ce travail de mémoire à été réalisé. Nous tenons à le remercier pour ces conseils, ses encouragements et son orientation. Merci Monsieur l'encadreur pour votre gentillesse et votre patience. Qu'il trouve aussi ici l'expression de notre profonde gratitude d'avoir initié et dirigé ce travail.

Nous adressons toute notre gratitude à Mr BENMOUNA. Y, pour avoir très gentiment accepté d'examiner ce travail. On la suit reconnaissante pour tous les efforts qu'il a fourni pour ce diplôme. Merci Mr pour votre soutien humain et scientifique et votre gentillesse.

Nous remercions également M^{me} CHAUCHE. L et M^{me} BENMANSOUR. F examinateurs.

fait un plaisir d'être examinateurs de ce travail. Veuillez trouver ici de notre haute considération.

Nous exprimons aussi toute notre sympathie et notre reconnaissance à l'ensemble des membres du magasin que nous avons visité pour notre projet.

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

Dédicaces :

A l'aide de DIEU tout puissant, qui trace le chemin de ma vie, donc Je dédie ce travail :

A mes chers parents jamais je ne saurais m'exprimé quant aux Sacrifices et aux dévouements que vous consacrés à mon éducation et mes études. Les mots expressifs soient-ils restent faibles pour énoncer ma gratitude hautement profonde.

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance.

- ✚ A mes sœurs Aicha, Asmaa et mon frère amine.
- ✚ A mes cousines ,mes tantes ,mes oncles
- ✚ A mes collègues asmaa, fouzia,imene D ,imane H, sabiha .
- ✚ A tous mes amis.
- ✚ Et à toute la famille zahaf sans exception.

ZAHAF AHLEM

DEDICACES :

Je dédie ce modeste travail, aux deux êtres les plus chers à mon

Cœur auxquels je dois mon existence :

Mon père et ma mère ,vous qui étiez toujours à mes cotés pour

Me soutenir et m'encourager à me battre sans j'aimais m'arrêter à mi-chemin ;que dieu vous protéger .

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance :

- ❖ A mes sœurs Assia, Djazia, Sarah, et mon frère Zakaria
- ❖ A mon fiancé Ishak.
- ❖ A mes collègues Iméne H ,Imane d,ahlem et sabiha .
- ❖ ET à toute la famille kada .

Kada fouzia

Table des matières

Remerciements.....	01
Dédicaces	02
Introduction Générale	06
Introduction.....	07
ChapitreI :L'étude du système existant et le langage de modélisation UML.....	08
I.1 introduction.....	09
1 .I-présentation du magasin	09
2.I-Le système d'achat et de vente des pièces de rechange	10
3.I-Les définitions des mots ainsi que les documents utilisés	11
4 .I- Lacunes du système existant.....	11
5.I-Proposition des solutions.....	12
6 .I-Origine et histoire d'UML.....	12
7.I-Définition UML	13
8.I-La caractéristique d'ULM	13
9 .I-Les diagrammes UML.....	14
Les différents types de diagrammes	14
1.Diagramme structurelle ou statique.....	14
2.Diagramme de comportement	15
2- Processus UP.....	16
- Définition du processus unifié	16
-introduction.....	17
-Les neuf disciplines d'UP.....	17
-Phase du processus unifié.....	18
- Conclusion	19
Chapitre II La conception Du Système GMPR.....	20
II.1-Introduction	21
II.2- Modélisation du système GMPR.....	21
a.Diagramme de cas d'utilisation	21
b.Les diagrammes de séquence	23
c- Diagramme de classe	36
II.3- Le Model Logique du système GMPR	37
II.4- Conclusion.....	37

Chapitre III L'Implémentation du Système GMPR	38
III.1- Introduction.....	39
III.2- L'environnement du programmation Delphi	39
III.3-L'environnement de gestion de base de donné	39
III.4-Description de l'application.....	39
a- Page d'authentification.....	40
b- Menu principal	41
c- Formulaire d' Ajouter une pièce	42
d- Formulaire de chercher un Client (par Nom)	43
e- Formulaire de rechercher un Fournisseur (par Num_Four).....	44
f- Formulaire Bon de Livraison Client	45
g- Formulaire Facture Client.....	46
h- Formulaire de Bon de Commande Fournisseur	47
i- Formulaire modifier mot de passe	48
III.5- Conclusion	48
Conclusion générale	49
Références bibliographiques.....	51

INTRODUCTION GENERALE

Introduction :

Parmi, tous les thèmes proposés au niveau du département informatique nous à vous choisir « la gestion d'un magasin de pièces de rechange » pour de multiple raison.

On particulier :

L'évaluation des entreprises dans un environnement concurrentiel, variable et complexe. Le système de gestion d'achat et de vente des pièces de rechange rencontre certaines difficultés :

- La perte du document.
- L'oubli de faire les livraisons à temps.
- La redondance dans le saisi des données.
- Temps de réponse très longs ou lent.

En effet, pour bien assurer la bonne gestion dans une entreprise et avoir la compétitive avec les autres entreprises, il faut utiliser un système informatisé.

Dans cet objectif, noter travail consiste à chercher et de concevoir un logiciel appelé « GMPR » qui va contenir la base des données et les interfaces permettant la bonne gestion du magasin dans le domaine d'achat et de vente des pièces de rechanges.

Chapitre I
L'étude du système existant
&
Le langage de modélisation UML

1. Introduction :

L'analyse de l'existant est une étape importante dans le cycle de vie d'un système, il s'agit de connaître la situation actuelle de l'organisation pour pouvoir porter un jugement juste.

Ainsi, l'analyse du système existant doit nous fournir toute l'information nécessaire, afin d'établir une bonne solution

Pour faire l'analyse du système existant, nous avons procédé par plusieurs étapes. Nous avons utilisé des méthodes et techniques différentes pour avoir les informations complètes du système.

A ce qui concerne la documentation, nous avons utilisé les documents différents du magasin de pièces de rechange poids lourd « Abderrahimi Nouredine » qui décrivent le fonctionnement du magasin qui sont :

Des classeurs des documents comptables comme : les commandes, les bons de livraison, les fiches de stock et les factures.

Les documents qui sont en Excel sur l'ordinateur.

Les documents utilisés par l'agent de stock pour les entrées et sorties de stock (registre de stock).

Pour l'interview, nous avons interviewé les assistants du magasin qui sont :

Mlle h.chahrazed, la secrétaire.

Monsieur Abderrahimi noureddine, le patron.

1 .I-présentation du magasin :

- Raison social : « Abderrahimi Nouredine ».agent agréé par S.N.V.I.
- Date de création : 06 juillet 1991.
- Adresse : au 1025, les dahlias_kiffane Tlemcen.
- Nombre employé : 03.
- Activités : Garage : vente des pièces.
- Administration : établir les bons de commandes, les bons de livraison les fiches de stock et les factures.
- Dépôt : stockage des pièces.

Organigramme du magasin de P.R :

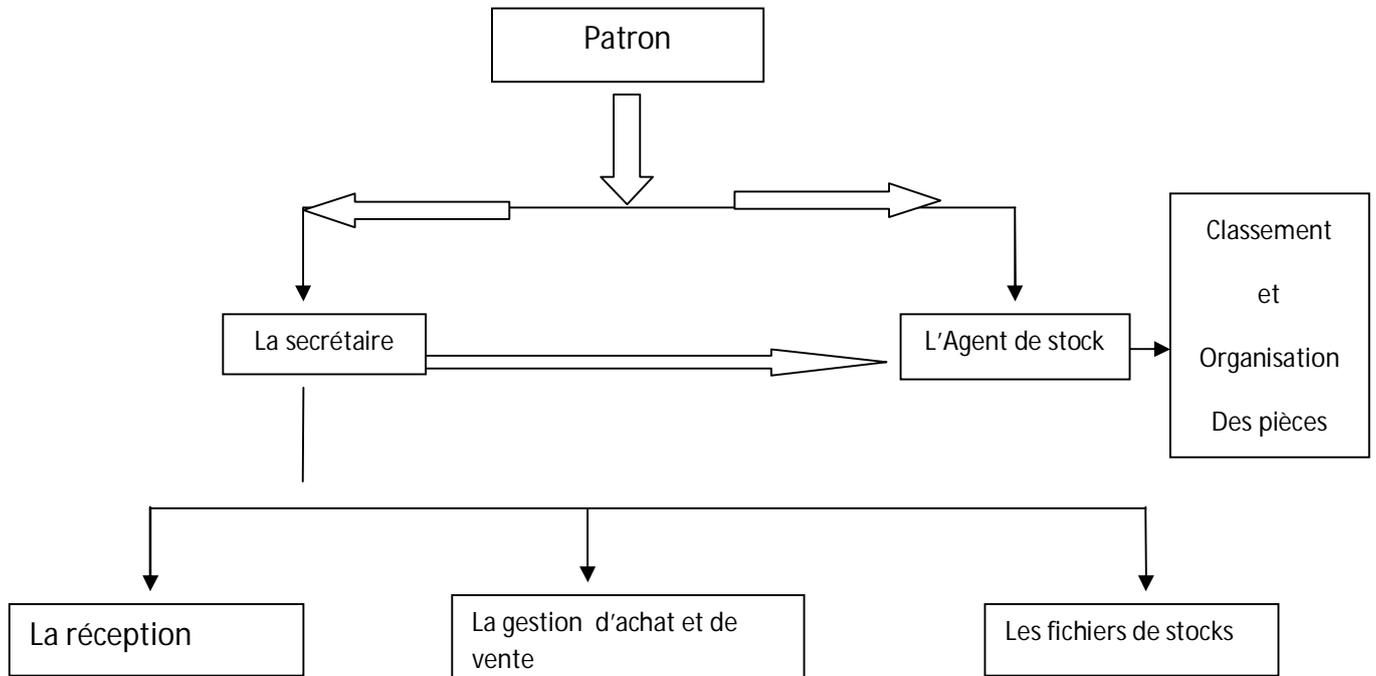


Figure 1 : l'organigramme du magasin.

2.I-Le système d'achat et de vente des pièces de rechange :

Le magasin avec son système d'achat et de vente des pièces de rechange, utilise le fichier Excel pour le traitement des entrées et sorties.

- **Les étapes du fonctionnement du système sont les suivantes :**
- ✓ **Pour la demande :** Le client se présente à la secrétaire pour demander la pièce, le patron lui envoie à l'agent de stock. Le nouveau client est enregistré, établit le bon de livraison et la facture en Excel appelé « client » ou les clients sont enregistrées, après la vérification du bon de commande, ce dossier est classé dans un classeur des commandes de l'année concernée.

- La secrétaire vérifie dans le fichier Excel appelé « pièce » ou les pièces sont enregistrées.

Si la pièce existe, la secrétaire établit le bon de livraison et la facture en Excel pour le paiement, si la pièce n'existe pas, la secrétaire établit la liste des pièces à commander et l'envoi à un fournisseur (A.P.C, sociétés et entreprises privées).

- ✓ **Pour la commande envers le fournisseur:** Le fournisseur envoie un bon de commande ou il enregistre les pièces et la quantité à commander.
- ✓ **Arrivée d'une pièce :** Quand le fournisseur envoie les pièces commandées et le bon de livraison, l'agent de stock appelle le mécanicien pour contrôler si les pièces sont conformes à la commande ou les pièces ne sont pas endommagées, si les pièces sont conformes à la commande, l'agent de stock enregistre les pièces entrant dans le fichier Excel (fichier de stock) et il les arrange dans le magasin. Au cas contraire, l'agent de stock établit la fiche de pièces endommagées et l'envoie au directeur commercial pour le retour des pièces non conformes ou endommagées

Le système n'est pas capable de faire la gestion de tout le calcul, il y a même la redondance dans l'élaboration des rapports ou l'agent de stock fait le copier coller.

3.I-Les définitions des mots ainsi que les documents utilisés :

Le bon de commande : document comptable qui enregistre les données en rapport avec les commandes envoyées au fournisseur et aux magasins.

Le bon de livraison : est un document comptable utilisé pour l'enregistrement des données concernant les livraisons faites.

La facture : est un document comptable relatif à un achat ou un service rendu.

4 .I- Lacunes du système existant :

Le système actuel utilise des fichiers Excel pour la gestion d'achat et de vente des pièces de rechange.

Les différents inconvénients du système existant sont les suivants :

- Difficulté de contrôler des achats et de ventes
- Compte tenu des fichiers Excel utilisés dans le magasin, ce n'est pas facile de :

- Gérer les commandes
- Gérer les livraisons
- Avoir les informations directes concernant les clients et les fournisseurs
- Générer des rapports

La gestion inefficace des pièces :

Il est difficile de contrôler :

- Les pièces de rechange achetées.
- Les pièces de rechange vendues.
-

5.I-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 en rapport avec l'achat à la vente.

Notre travail de recherche aboutira aux objectifs suivants :

Suivi des commandes pour les fournisseurs et les clients

Eviter la perte des documents

Faciliter le rangement des pièces

Faciliter l'inventaire périodique.

La liste des pièces de rechange vendues et achetées (statique).

6 .I-Origine et histoire d'UML :

La modélisation objet consiste à créer une représentation informatique des éléments du monde réel auxquels on s'intéresse, sans se préoccupe de l'implémentation. Il s'agit donc de terminer les objets présents et d'isoler leurs données et les fonctions qui les utilisent.

Pour cela des méthodes ont été mises au point. Entre 1970 et 1990, de nombreuses analyses ont mis au point des approches orientées objets, si bien qu'en 1994 il existait plus de 50 méthodes objet. Toutefois seules 3 méthodes ont véritablement émergé :

OMT de James Rumbaugh (General Electric) qui fournissait une représentation graphique des statiques et fonctionnel d'un système .

OOD de Gready Booch (department of defense) qui introduisait le concept de paquetage (package) ;

OOSE de Jacobson (Ericson) qui fondait son analyse la description des besoins des utilisateurs (cas d'utilisation, ou use case).

Chaque méthode avait ses avantages et ses partisans. Bien que le nombre de méthodes s'était, la profession risquait encore de se diviser entre ces trois méthodes, créant autant de communautés qui auraient du mal à communiquer.

Dans un objectif de normalisation et de standardisation les trois constructeurs qui régnaient chacun sur l'une des trois méthodes se mirent d'accord pour définir une méthode commune qui fédérerait leurs apports respectifs (ils sont depuis surnommés <<the amigos>>).

UML (unified modeling language) pour langage de modélisation unifié est né de cet effort de convergence. Le terme langage est quant à lui employé pour souligner qu'UML à l'ambition d'être un langage et non celle d'être une méthode ;

7.I-Définition UML :

UML (sigle désignant l'unified modeling language ou <<langage de modélisation unifié >>) est un langage de modélisation graphique à base de pictogrammes, il est apparu dans le monde de génie logiciel, dans le cadre de la <<conception orientée objet >> UML est couramment utilisé dans les projets logiciel.

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet, UML offre un standard de modélisation, pour représenter l'architecture logicielle

8.I-La caractéristique d'UML :

UML est un langage formel et normalisé caractérisé par :

- Grain de précision.
- Gage de stabilité.
- L'utilisation d'outils.
- Encourage un support de communication performant

- Il cadre l'analyse.
- Il facilite la compréhension de diagramme ion de représentation abstraite complexe.
- Son caractère polyvalent et souplesse 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 une tâche complexe et longue.

9 .I-Les diagrammes UML :

Un type de diagramme UML véhicule une sémantique précis (un type de diagramme offre toujours la même vue d'un système).

Combinés, les différents types de diagramme UML offrent une vue complète des aspects statiques et dynamique d'un système.

Par extension et abus de langage, un diagramme UML est aussi un modèle (un diagramme modélisation un aspect du modèle globale).

❖ Le différent type de diagramme :

1. Diagramme structurelle ou statique :

a) Diagramme de classe :

Un diagramme des classes décrit le type des objets du système ainsi que la différente forme de relation statique qui retient entre eux.

On distingue classiquement deux types principaux de relation Les associations, bien connues des vieux modèles entité/association utilisé dans la conception des bases de données depuis les années 70

Les sous types, particulièrement en vogue conception orientée objets, puisqu'ils s'expriment très bien à l'aide de l'héritage en programmation.

b) diagramme d'objet :

Un objet est instance d'une classe, un diagramme d'objets est un ensemble d'objet respectant les contraintes du diagramme de classe, respect des cardinalités.

Chaque attribut d'une classe a une valeur affectée dans chaque instance de cette classe

c) diagramme de composant :

Il permet de montrer les composants du système d'un point d'un point de vue physique tels qu'ils sont mise en ouvre

d) diagramme de déploiement :

Il sert à représenter les éléments matériels et la manière dont les composant du système sont répartir sur ces élément matériels et interagissent entre eux.

e) Diagramme de paquetage :

Le diagramme de paquetages sert à représenter les dépendances entre paquetage. C'est-à-dire les dépendances entre ensemble de définition.

f) Diagramme de structure composite :

Permet de décrire sous forme de boite blanche les relation entre composants d'une classe.

2. Diagramme de comportement :

a) Diagramme de cas d'utilisation :

Un cas d'utilisation modélisation une interaction entre le système informatique à développer et utilisation ou un acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'action réalisée par le système qui produit un résultat observable pour un acteur.

Il ya en général deux types de description des use cases :

- ✓ Une description textuelle de chaque cas ;
- ✓ Le diagramme des cas d'utilisation qui présente une synthèse de l'ensemble des cas ;

▪ **Dépendances entre cas d'utilisation :**

Il est parfois intéressent d'utiliser des clients entre cas, UML en fournit deux types :

La relation utilisée (include) : est employée quand deux cas d'utilisation ont en commun une même fonctionnalité et que l'on souhait factoriser celle-ci en créant un sous-cas, ou cas intermédiaire, afin de marquer les différences d'utilisation.

La relation étend (ex-tend) : nous dirons qu'il y a extension d'un cas d'utilisation quand un cas est globalement similaire à un autre ou lorsque un cas doit être spécialisé ou adaptée.

b. Diagramme état-transition :

Permet de décrire sous forme de machine à état fini le comportement du système ou de ses composants.

c. diagramme d'activité :

Permet de décrire sous forme de flux ou d'enchaînement d'activité le comportement du système ou de ses composants.

1. Diagramme d'interaction dynamique :

a) Diagramme de séquence :

Représentation séquentielle du déroulement des traitements Et des interactions entre les éléments du système et /ou de ses acteurs.

b) Diagramme de communication :

Représentation simplifié d'un diagramme de séquence se concentrant sur les échanges de message entre les objets.

c) Diagramme globale d'interaction :

Permet de décrire les enchaînements possibles entre les scénarii préalablement identifiés sous forme de diagramme de séquences.

2-Processus UP :

- Définition du processus unifié :

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

C'est un patron de processus pouvant être adaptée à une large classe de système logiciel,

A différents domaine d'application, à différent type d'entreprises, à différent niveau de compétences et à différent taille d'entreprise.

❖ 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).

➤ 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 vues.

4 - L'implémentation :

Vise le triple objectif de raffiner le modèle de conception, générer le code source et les tests unitaires associés, intégrer le travail d'implémentation des différents équipes 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 avantagement 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....

2-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.

Construction :

C'est dans cette phase que la capture de les exigences doit être finalisée, mais aussi et surtout, que les différents incréments de l'application doivent être conçus et implémentés.

Transition :

Cette phase consiste à finaliser le produit et à effectuer la livraison du système auprès des utilisateurs finaux.

-conclusion :

Particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente, en effet nous avons choisi le processus UP qu'on utilisera pour modéliser notre système. Cette modélisation sera détaillée dans le chapitre suivant.

Chapitre II

La conception Du Système GMPR

II.1-Introduction :

Ce chapitre porte sur la conception du système notre choix s'est porté sur le processus UP, en conséquence, nous allons détailler 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 GMPR :

Pour modéliser le système GMPR nous allons commençons par partager ce système

En plusieurs cas d'utilisations chaque cas correspond à un ou plusieurs scénarios.

a-Diagramme de cas d'utilisation :

Suivant (Fig. les cas d'utilisation et les acteurs GMPR Sont schématisés dans le diagramme II.1).

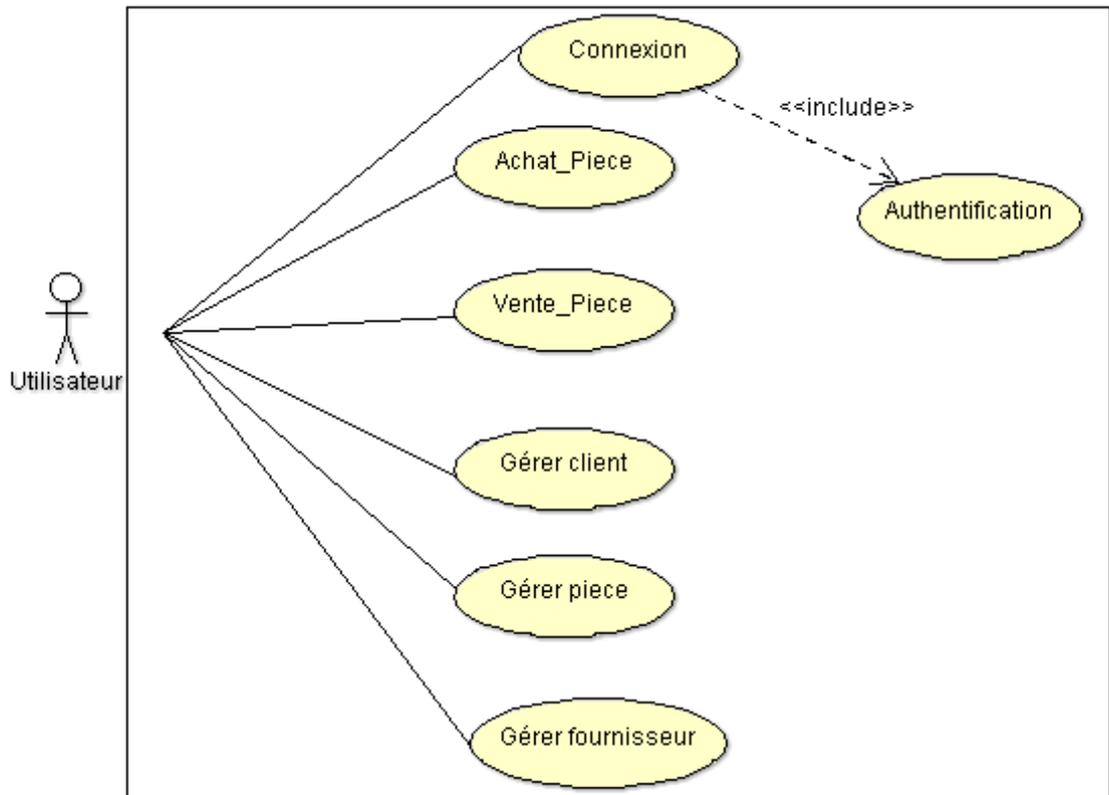


Fig . II.1 : Diagramme de cas d'utilisation « gestion d'achat et vente de pièces de rechanges ».

• **Identification d'acteur « Utilisateur » :**

L'utilisateur joue un rôle très important. Cet acteur est une personne qui travaille au magasin, dans notre cas c'est la secrétaire.

L'acteur interagit avec le système GMPR :

- ✓ Gestionnaire d'un magasin
- ✓ Commerçant de pièces de rechange

b- Les diagrammes de séquence :

❖ Le scénario de cas d'utilisation « Connecter » :

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.

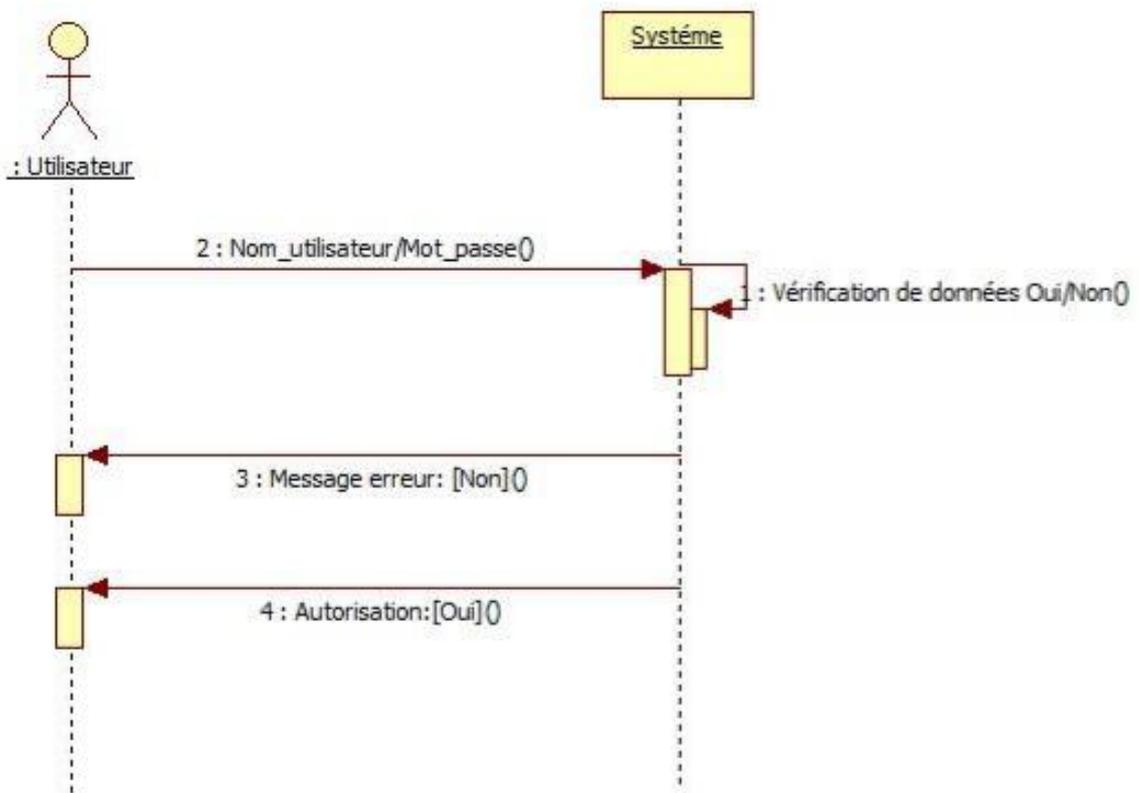


Fig. II.2 : Diagramme de séquence« Connecter ».

❖ Les scénarios de cas d'utilisation « Acheter pièces » :

➤ Gérer commande fournisseur :

Si le magasin a un manque de pièce, l'utilisateur doit créer un bon de commande au fournisseur.

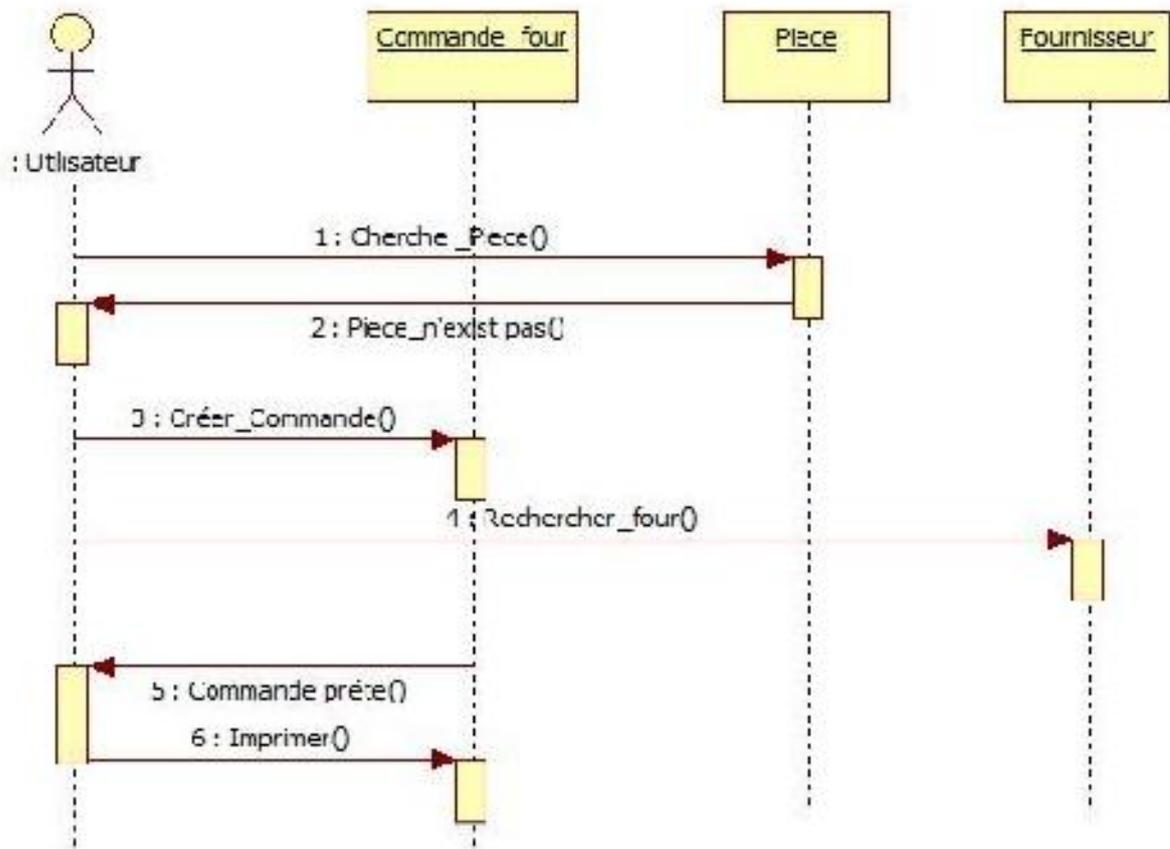


Fig. II.3 : Diagramme de séquence « gérer commande fournisseur ».

➤ Facture Fournisseur :

A la réception de la facture d'un fournisseur, l'Utilisateur

1. Cherche le fournisseur (par nom ou numéro).
2. Enregistre la facture.
3. Fait le mis à jour du fichier de la pièce avec la quantité mentionnée dans le bon de commande et le bon de livraison.

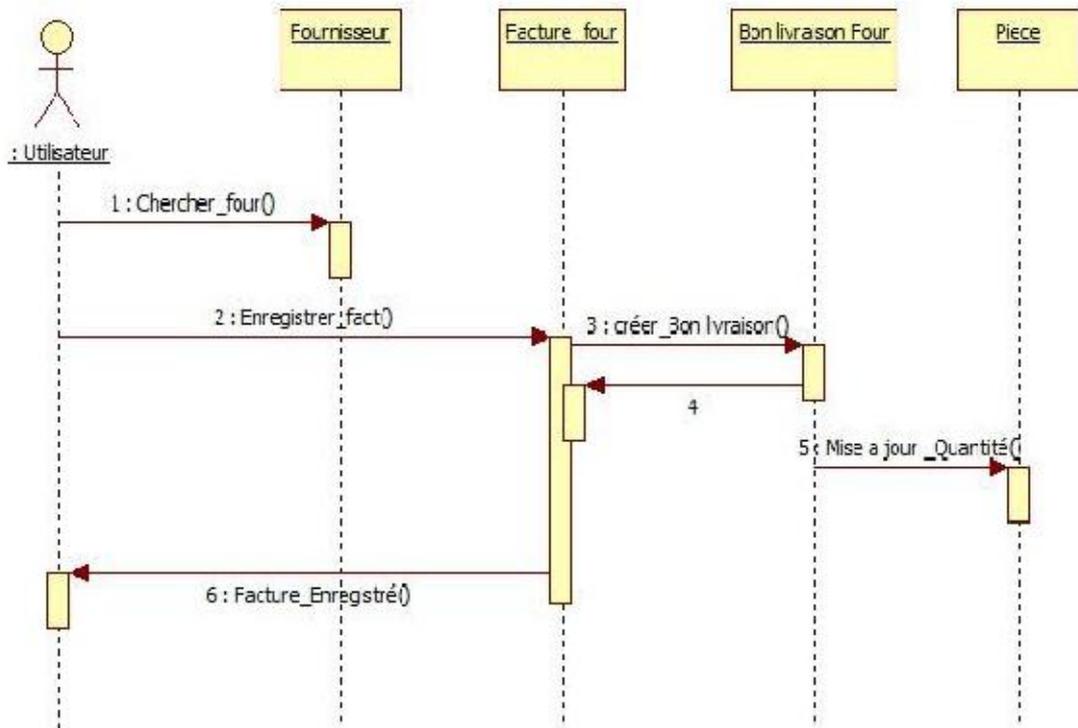


Fig. II.4 : Diagramme de séquence « Facture fournisseur ».

❖ Les scénarios de cas d'utilisation « Vente pièces » :

➤ Traitement des bons de commande :

L'utilisateur enregistre le bon de commande reçu de client.

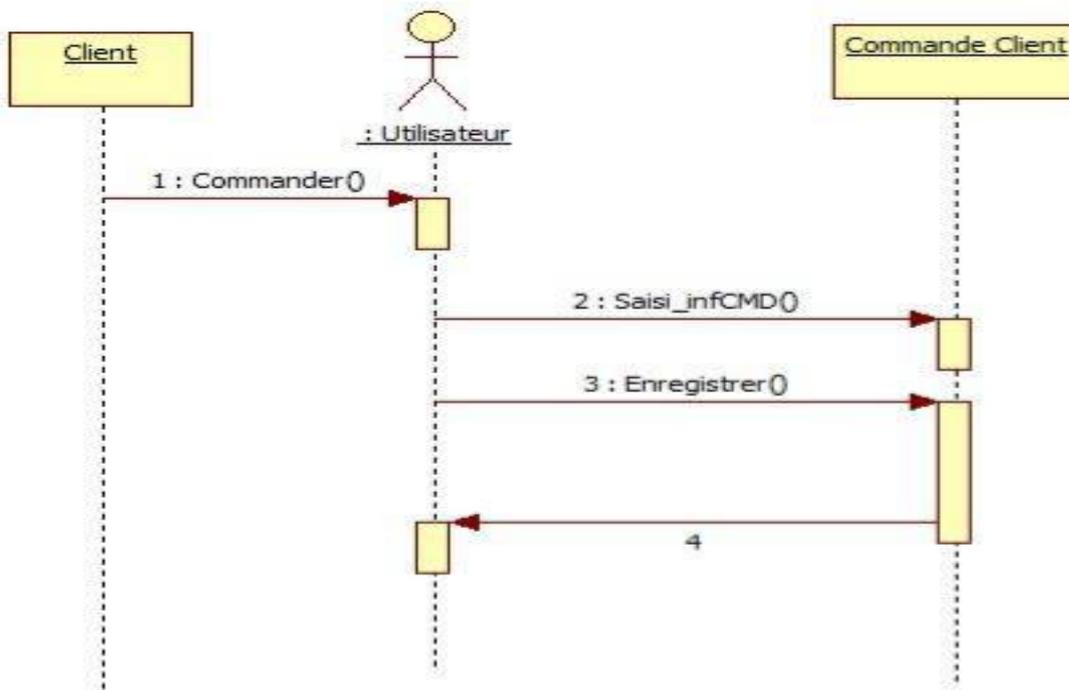


Fig. II.5 : Diagramme de séquence « Traitement des bons de commande ».

➤ Facture Client :

A la réception de bon de commande d'un client, l'Utilisateur :

1. Recherche le client dans la base de données si le client a été trouvé alors il édite une facture.
2. Ajoute les pièces avec leur quantité demander à partir de bon de commande de client, pour chaque pièces ajouter le système automatiquement le total montant par pièces ainsi que le total acheter et le TTC.

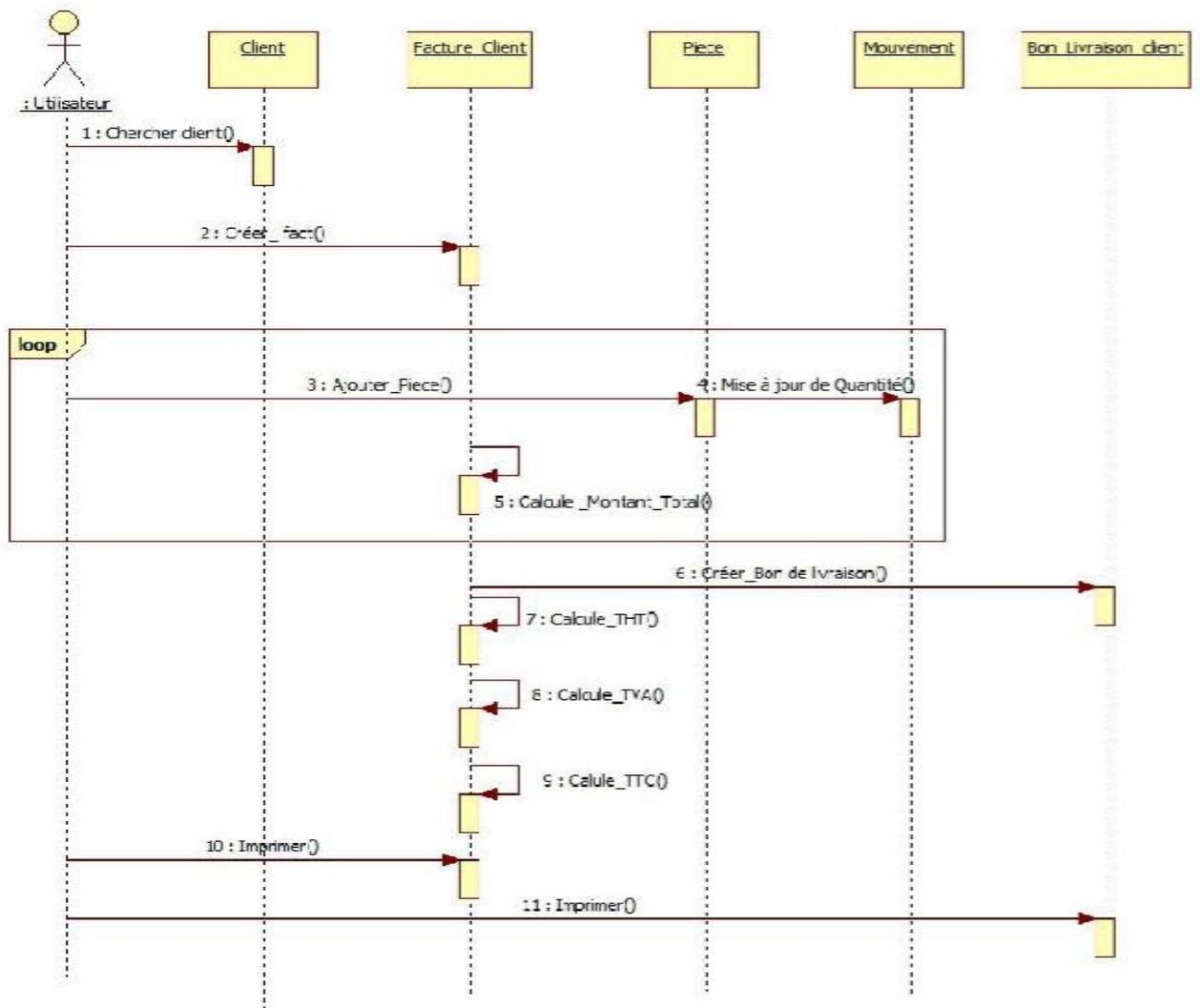


Fig. II.6 : Diagramme de séquence « Facture Client ».

❖ Les scénarios de cas d'utilisation « Gérer Client » :

➤ Ajouter Client :

Si l'utilisateur a un nouveau client, il doit l'ajouter.

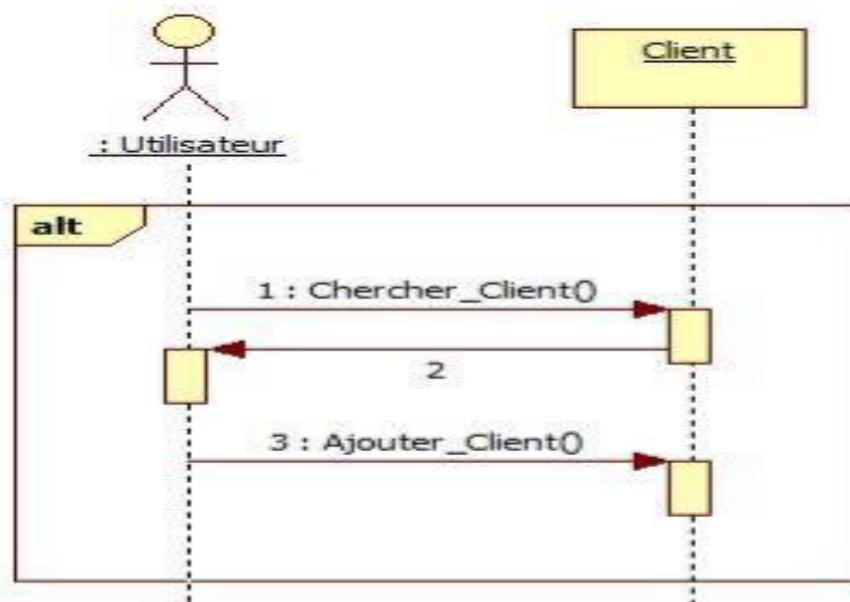


Fig. II.7 : Diagramme de séquence « Ajouter Client ».

➤ Modifier Client :

Si l'utilisateur a des modifications dans les informations des clients (changement de leur adresse ou numéro de téléphone), il va les modifier.

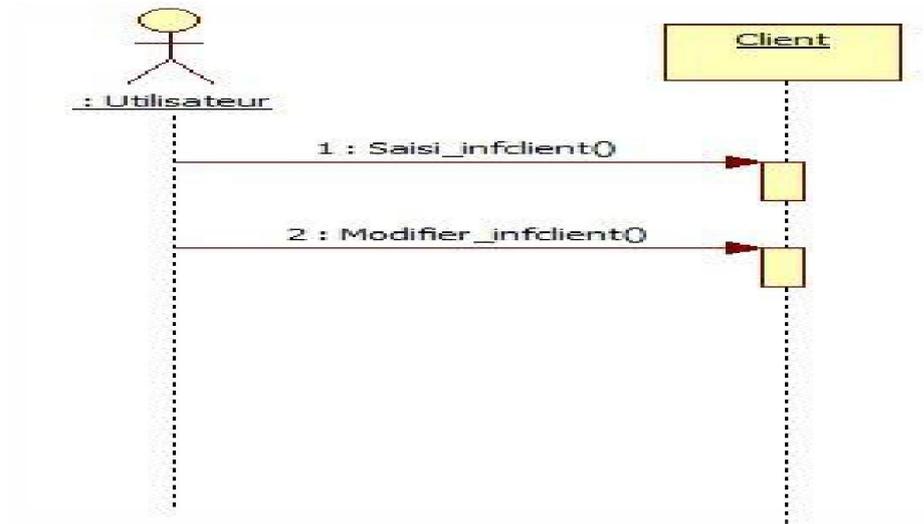


Fig. II.8 : Diagramme de séquence « Modifier Client ».

➤ Supprimer Client :

Si un client arrête de commander des pièces pendant une longue durée, l'utilisateur doit le supprimer.

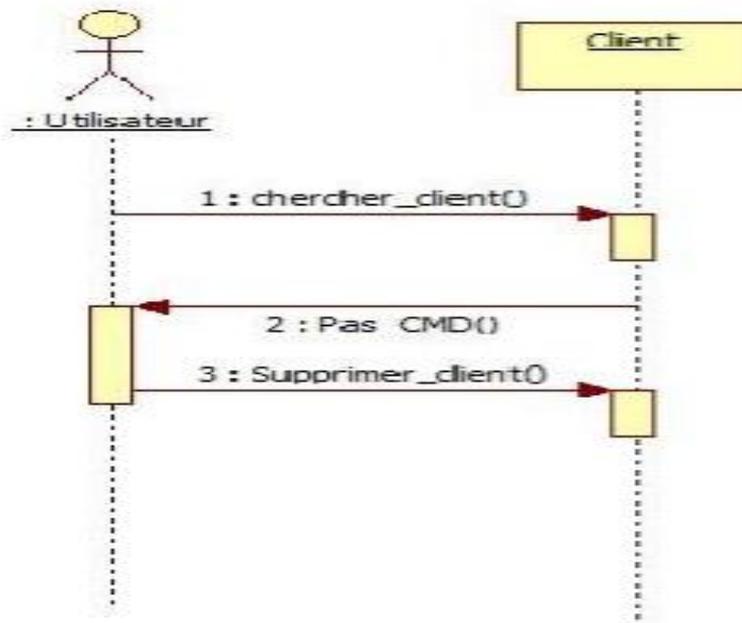


Fig. II.9 : Diagramme de séquence « Supprimer Client ».

❖ Les scénarios de cas d'utilisation « Gérer Pièces »

➤ Ajouter Pièce

Si l'utilisateur a une nouvelle pièce, il va créer une nouvelle pièce.

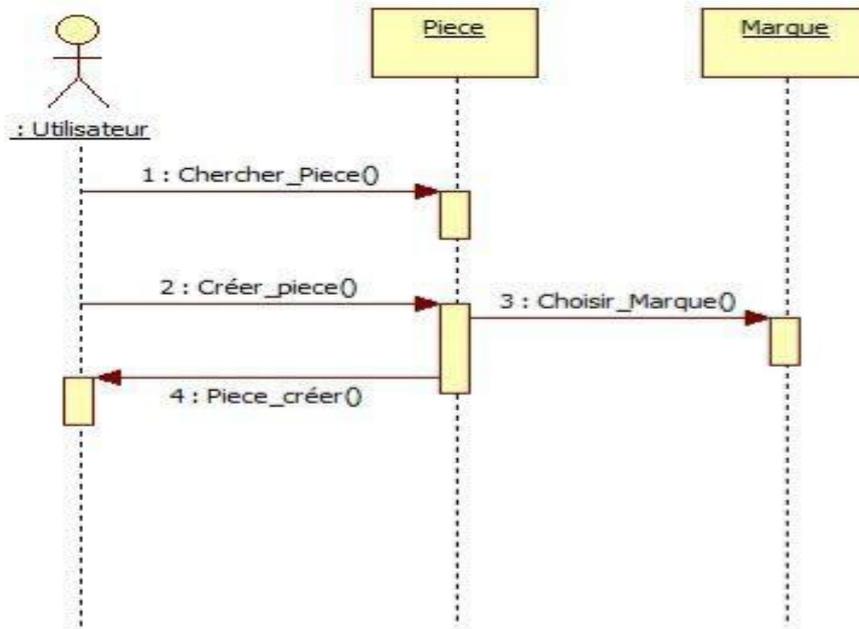


Fig.II.10 : Diagramme de séquence « Ajouter pièce ».

➤ Modifier Pièce :

Si la quantité de la pièce est changée, l'utilisateur va la modifier

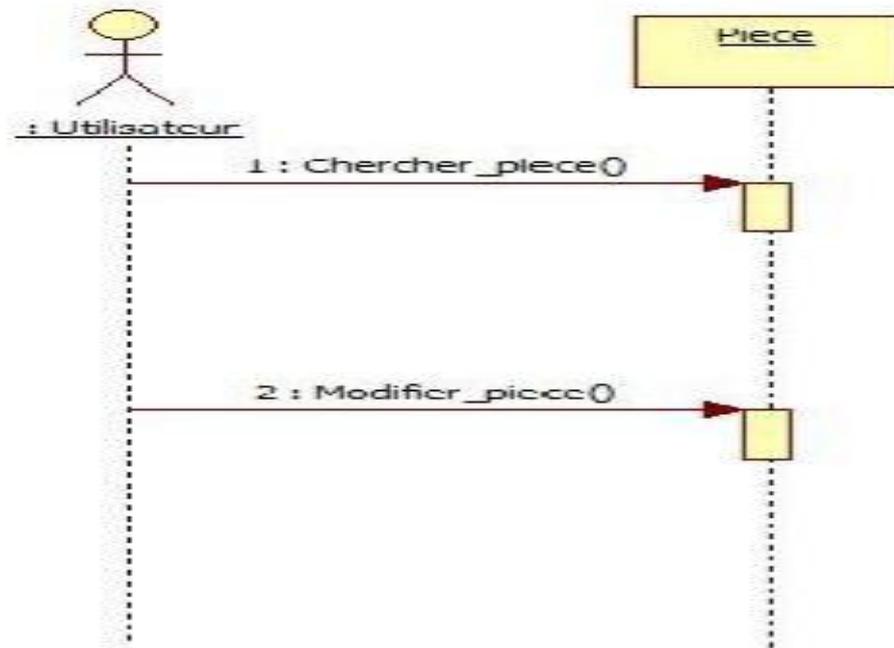


Fig. II.11 : Diagramme de séquence « Modifier Pièce ».

➤ Supprimer Pièce

L'utilisateur doit supprimer la pièce s'elle dépasse la date d'expérimentation.

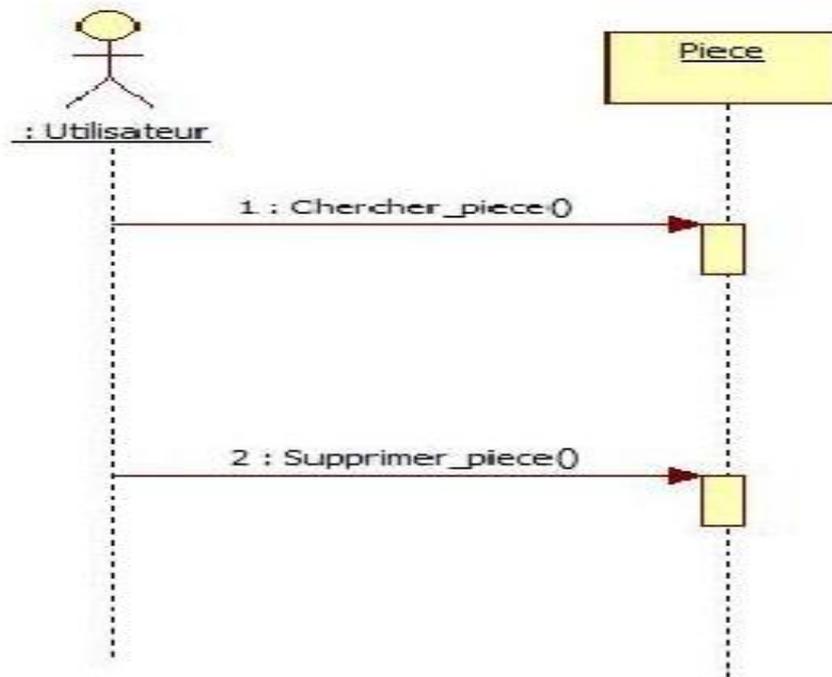


Fig. II.12 : Diagramme de séquence « Supprimer Pièce ».

❖ Les scénarios de cas d'utilisation « Gérer fournisseur »

➤ Ajouter Fournisseur

Si l'utilisateur a un nouveau fournisseur, il doit l'ajouter.

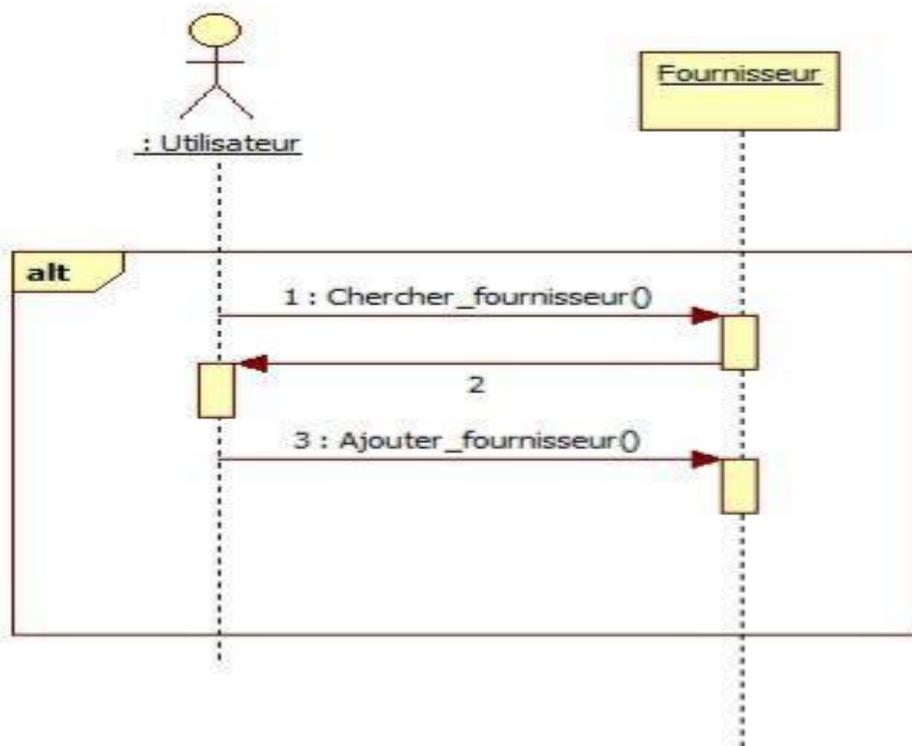


Fig. II.13 : Diagramme de séquence « Ajouter Fournisseur ».

➤ **Modifier Fournisseur :**

Si l'utilisateur a des modifications dans les informations de fournisseur (changement de leur adresse ou leur numéro de téléphone), il les va modifier.

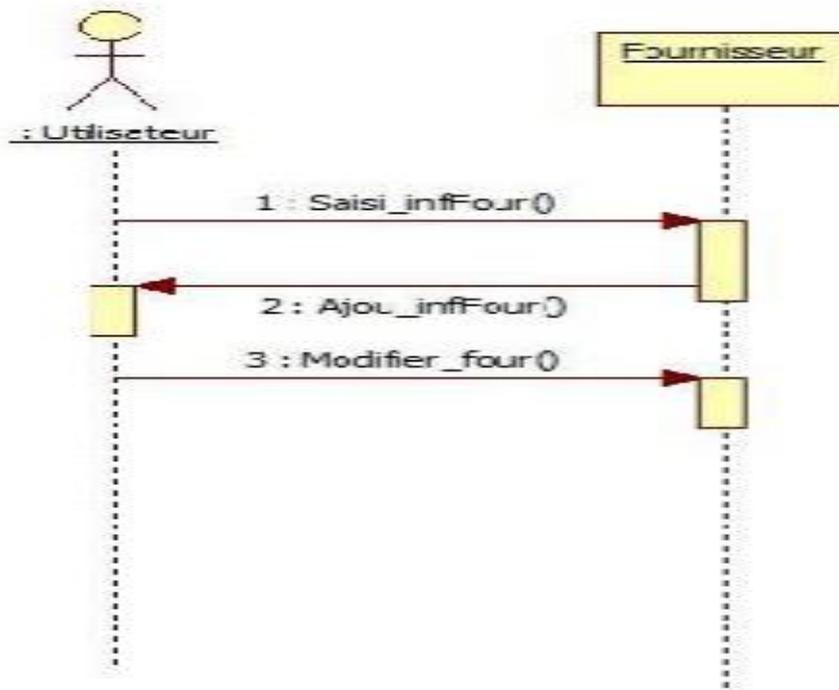


Fig. II.14 : Diagramme de séquence « Modifier Fournisseur».

➤ **Supprimer Fournisseur :**

S'il y a un fournisseur qui s'arrête de livrer des pièces pendant une longue durée, l'utilisateur doit le supprimer.

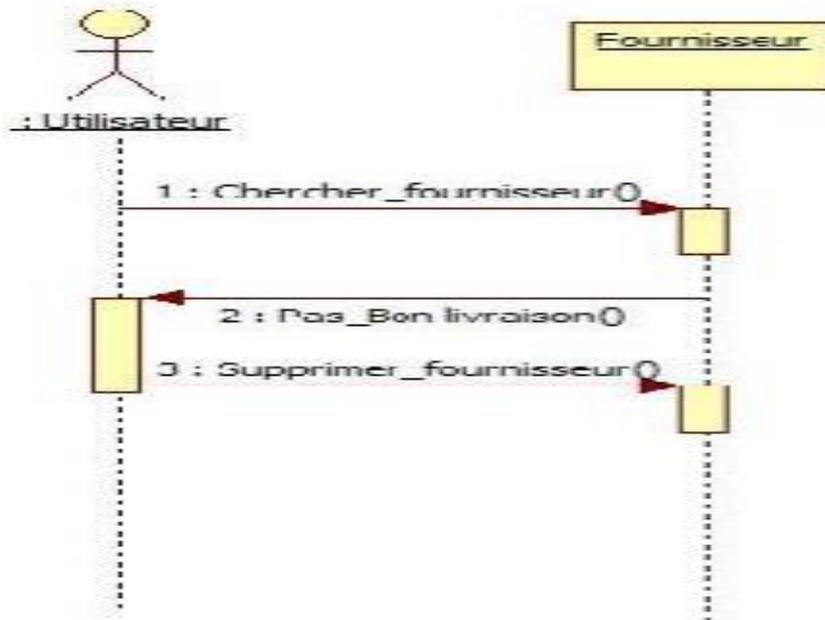


Fig.II.15: Diagramme de séquence « Supprimer Fournisseur ».

Chapitre II :La conception Du Système GMPR

c- Diagramme de classe :

Le diagramme de classes identifie les classes de notre système et les associations entre elles. Ce diagramme contient neuf classes

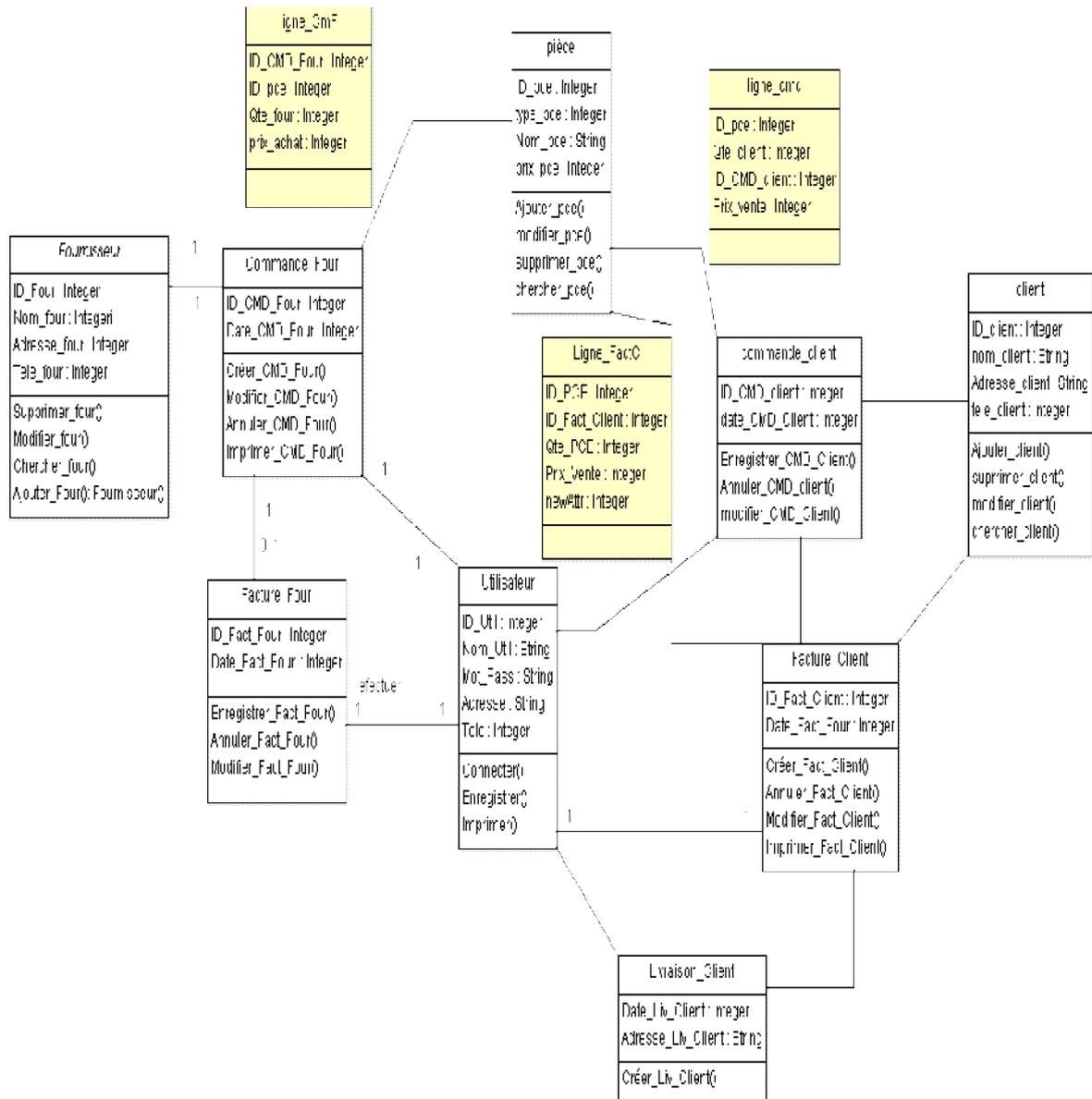


Figure II.16 : Diagramme de classes.

II.3- Le Model Logique du système GMPR

Piece (ID_Pce, Nom_Pce, Type_Pce, Prix_Pce)

Utilisateur (ID_Util, Nom_Util, Mot_Pass, Adresse, Tele)

Commande_Four (ID_CMD_four, Date_CMD_four, ID_Four*, ID_Util*)

Commande_Client (ID_CMD_Client, Date_CMD_Client, ID_Util*, ID_Client*) .

facture_Four (ID_fact_four, Date_fact_four, ID_CMD_four*, ID_Util*, ID_Four*)

Livraison_Client (Date_Liv_Client, Adresse_Liv_Client, ID_Util*, ID_fact_client*)

Client (ID_Client, Nom_Client, Adresse_Client, Tele_Client)

Fournisseur (ID_Four, Nom_Four, Adresse_Four, Tele_Four)

Ligne_CmF(ID_CMD_Four*,ID_Pce*,Qte_Four,Prix_Achat)

Ligne_FactC (ID_Pce*,ID_fact_client,Qte_Pce,Prix_Achat).

Ligne_CmC (ID_CMD_Client*,ID_Pce*,Qte_Client,Prix_Vente)

II.4- Conclusion :

Nous avons traité dans ce chapitre la conception du système GMPR. En utilisant le langage UML, cette conception est une phase nécessaire pour pouvoir créer notre base de données et réaliser l'application, le chapitre suivant détaillera les phases de l'implémentation de notre système

Chapitre III

L'Implémentation du Système GMPR

III.1-Introduction :

Ce chapitre est consacré à l'implémentation de notre application qui s'appuie sur la modélisation présentée dans le chapitre précédent, pour l'implémentation nous avons utilisé Delphi version 5.

III.2-L'environnement de programmation Delphi :

Delphi est un environnement de programmation visuel orienté objet pour le développement rapide d'application (**RAD**). Il incarne la suite logique de la famille turbo pascal. C'est un outil moderne qui fait appel à une conception visuelle des applications. De plus, il prend en charge le maintien automatique d'une partie du code source.

Voici quelques unes des caractéristiques de Delphi :

- ✚ Il supporte le langage pascal orienté objet.
- ✚ Utilisation visuelle bidirectionnelle.
- ✚ Compilateur produisant du natif.
- ✚ Traitement complet des exceptions.
- ✚ Possibilité de créer des programmes exécutables et des DLL.
- ✚ Bibliothèque des composants extensibles.
- ✚ Déboguer graphique intégré.
- ✚ Support de tous les API de Windows : OLE2, DDE, VBX, OCX.....

III .3-L'environnement de gestion de base de donné :

Nous avons utilisé le module de base de données version 5 (BDE) de Delphi pour la création de différentes tables de notre base de donné.

Ce module permet de créer une base de données dans différents formats, tel que les bases de données dans différents format, tel que les bases de données Accès, oracle Sybase, c'est un environnement graphique qui permet la création, la modification et la suppression des tables, ainsi que la définition des requêtes SQL permettant de retourner différent type de résultat a partir des tables.

III.4- Description de l'application :

Cette partie concerne la description de notre application, la gestion d'un magasin de pièces de rechange.

Elle est composée d'une base de données qui contient les fichiers pièces, clients et fournisseurs.

a- Page d'authentification :

Cette forme permet à l'utilisateur de s'introduire au menu de l'application. Il devra saisir le nom d'utilisateur, le mot de passe.



Fig. III.1 : Page d'authentification.

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 principal.

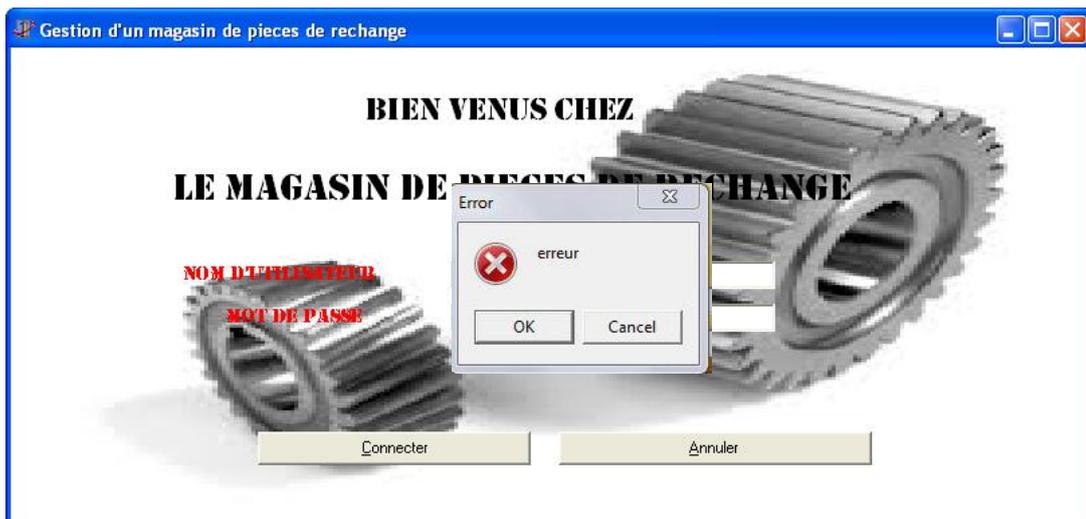


Fig.III.2 : Page d'authentification « cas d'erreur ».

b- Menu principal :

La forme de menu principal contient des différents menus permettant d'accéder aux différents formulaires.

Elle comprend six menus principaux : Pièce, Client, Fournisseur, Traitement, Recherche et Aide.

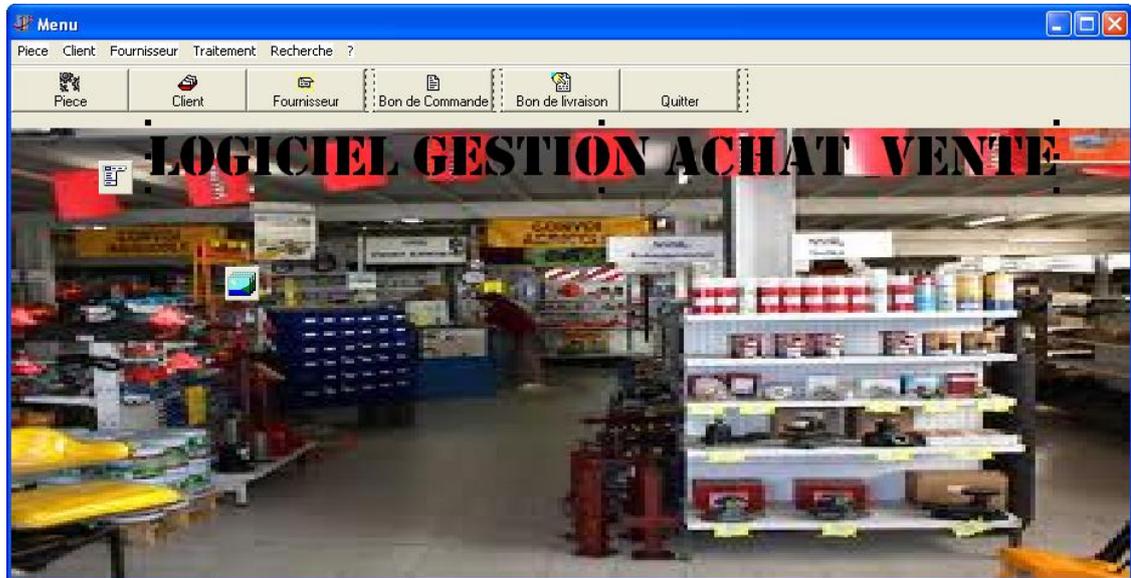


Fig. III.3 : Menu Principal.

c- Formulaire d'Ajouter une pièce :

Cette forme permet d'ajouter une nouvelle pièce

The screenshot shows a software window titled "Nouveau Piece". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main content area has a light beige background with a dotted pattern. At the top left, there is a small image showing various mechanical parts like filters and belts. To the right of this image, the text "AJOUTER PIECE" is written in bold red letters. Below the image and text, there is a form with several input fields. Each field is labeled with a green text and followed by a text box containing "DBE dit" followed by a number. The labels and their corresponding text boxes are: "Identifiant" (DBE dit1), "Nom" (DBE dit2), "Type" (DBE dit3), "Désignation" (DBE dit6), "Reference" (DBE dit4), "Prix Unitaire" (DBE dit5), and "Unité" (DBE dit7). To the right of these fields, there is a panel titled "Execution" containing six buttons: "Ajouter", "Supprimer", "Modifier", "Valider", "Enregistrer", and "Fermer". Below this panel, there is a button labeled "Liste de Pieces".

Figure III.4 : Formulaire d'ajouter une pièce.

d- Formulaire de chercher un Client (par Nom)

Cette forme permet de chercher et afficher si un client existe ou pas par son nom.

ID_Client	Nom_Client	Téle_Client	Adresse_Client
6	APC SIDI OUARACHE	9865434	AIN TEMOUCHANTE
5	APC SIDI YAEKOUB	876554	SIDI BELABASS
9	SOCIETE DENNOUNI ESTPH	897654	TLEMCEN
7	SOCIETE NAFTAL	4355636	REMCHI
10	SOCIETE SARL INTER ENTREPRISE	4309876	TLEMCEN
8	SOCIETE STARR	430876	TLEMCEN

Fig. III.5 : Formulaire de recherche Client.

e- Formulaire de rechercher un Fournisseur (par Num_Four) :

Cette forme permet de chercher et d'afficher, si le fournisseur existe ou pas par son identifiant.

ID_four	Nom_four	Téle_four	Adresse_four
1	SARL COMPAN''	2345678	CRAN
2	DELMI RAL-ID	12345678	AIN MULA
3	D.AFCUD MOHAMED	987654	SIDI EELA3ASS

Fig. III.6 : Formulaire recherche Fournisseur.

f- Formulaire Bon de Livraison Client :

Cette forme va aider le magasin à connaître si la livraison est conforme à la commande du client.

The screenshot shows a software window titled "Facturation Client" with a sub-tab "Bon Livraison | Facture_Client". The form contains the following fields and values:

- N°B/Livraison:
- Client:
- Adresse: TLEMCEN
- Quantité: 4
- Désignation: K-20
- Date: 21/06/2013
- Prix Unitaire: 11514
- Montant:

Summary section (TOTAL):

- T.HT:
- T.TVA:
- T.TC:

ID_Iv_Client	Quantité	Désignation	Prix_unitaire	Date_Iv_Client	Adresse_Iv_Client	Mont	THT
	4	K120	11514	12/03/2013	TLEMCEN		

Buttons: Imprimer, Ajouter, Supprimer, Modifier, Valider, Fermer

Fig. III.7 : Formulaire de Bon De Livraison.

g- Formulaire Facture Client :

Cette fenêtre représente le traitement de facture de client chez le magasin ;

Facturation Client

Bon Livraison | Facture_Client

N°Facture: 1 | Date: 21/06/2013 | TOTAL: T.HT 25000

Client: APC AIN FEZZA | N°B/C: | Prix_Unitaire: 5000 | T.TVA: 1.17

Adresse: TLEMCCEN | Montant: 25000 | Quantite: | T.TC: 29250

Designation: 5 | Reference: |

ID_Fact_Client	Date_Fact_Client	ID_Client	ID_CMD_Client	CLIENT
1	14/08/2012	1	1	APC AIN FEZZA
2	10/05/2012	2	2	APC BENI BEHDEL
3	06/06/2012	1	3	APC AIN FEZZA
4	01/01/2013	4	4	APC HASSI ZEHANA
5	05/06/2013	5	5	APC SIDI YAEKOUB

ID_Pce	Nom_Pce	Type_Pce	Reference
2	AMORTISSEUR	AV-AR	19090015
3	ALTERNATEUR	24V	1128762
4	AMORTISSEUR	ARRIERE	1090011
6	ARBRE	PRINCIPALE	134782

ID_Fact_Client	ID_Pce	Qte_Pce	Prix_Vente	PIECE
1	1	5	5000	
1	3	33	23	ALTERNATEUR

Recherche La piece

Fig.III.8 : Formulaire Facture Client.

h- Formulaire de Bon de Commande Fournisseur :

Cette forme représente le traitement de Bon de Commande imprimé au fournisseur

The screenshot displays a software window titled 'Facturation Fournisseur' with a sub-tab 'Facture Fournisseur'. The interface includes several input fields and a data table. The fields are: 'N°B/C' with value '1', 'Date' with value '21/06/2013', 'Fournisseur' (dropdown), 'Designation' with value 'K66', 'Adresse' (empty), 'Reference' with value '16E 627', and 'Quantité' with value '3'. Below these fields is a table with the following data:

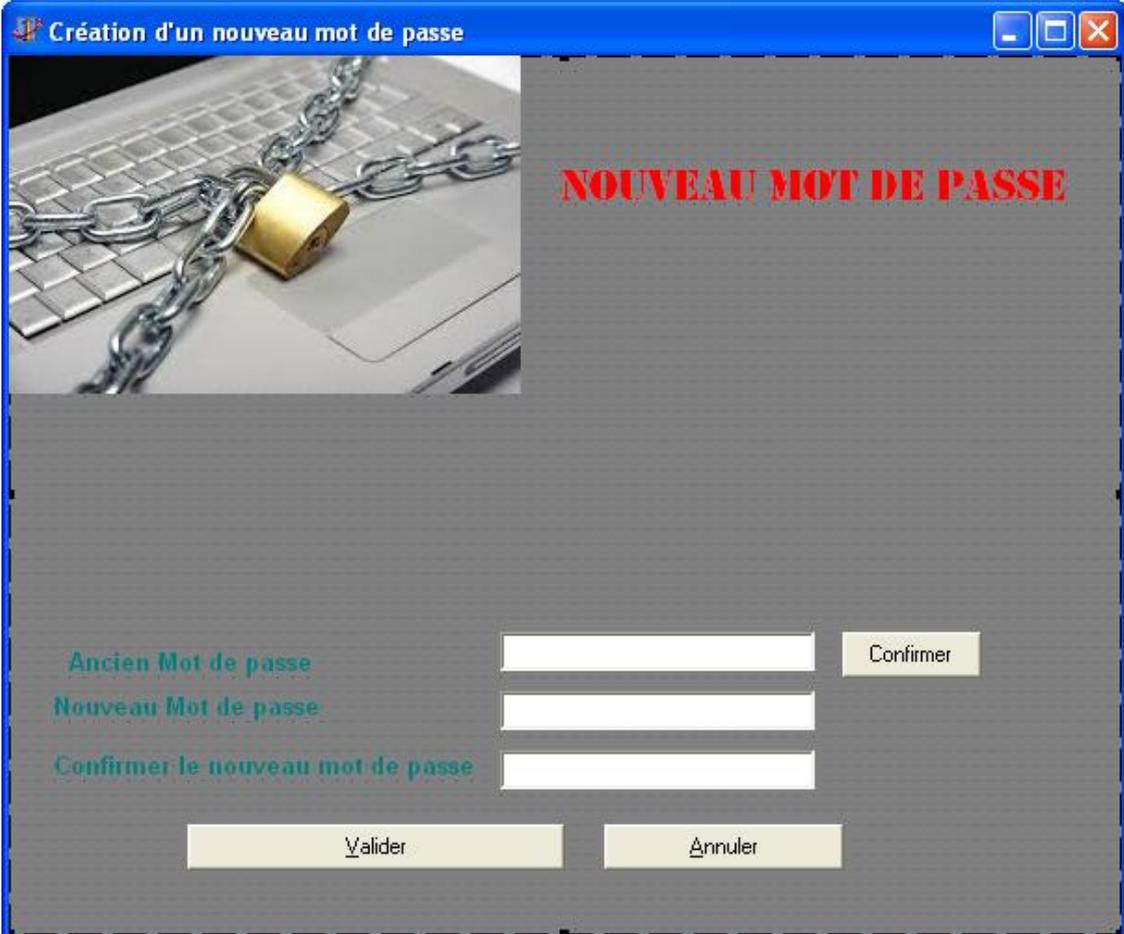
ID_Four_Commande	Désignation	Reference	Qte	FEF Rectifiée	Prix Unitaire	Date_CMD_four
1	K66	16E 627	3		3531	13/04/2013

At the bottom of the window, there are six buttons: 'Imprimer', 'Ajouter', 'Supprimer', 'Modifier', 'Valider', and 'Fermer'.

Fig. III.9 : Formulaire Bon de Commande Fournisseur

i- Formulaire modifier mot de passe :

Cette fenêtre permet de modifier le mot passe de l'application.



The screenshot shows a window titled "Création d'un nouveau mot de passe" with a blue header bar. The main content area has a grey background. On the left, there is a photograph of a laptop keyboard with a silver chain and a gold padlock. To the right of the photo, the text "NOUVEAU MOT DE PASSE" is displayed in large, bold, red capital letters. Below this, there are three input fields for passwords, each with a corresponding label in teal text: "Ancien Mot de passe", "Nouveau Mot de passe", and "Confirmer le nouveau mot de passe". To the right of the first input field is a "Confirmer" button. At the bottom of the form, there are two buttons: "Valider" and "Annuler".

Fig.III.10: Formulaire modifier mot de passe.

III.5- Conclusion

Dans ce dernier chapitre, nous avons présenté la partie réalisation de notre projet, et nous avons présenté les interfaces les plus importantes.

Conclusion Générale

Conclusion Générale

Conclusion général :

Le travail qui nous a été confié consiste à réaliser le système de gestion d'un magasin de pièce de rechange, l'analyse, la spécification, la conception sont orienté objet et décrites en UML en se basant sur le processus UP.

L'implémentation de l'application a été réalisée par l'environnement Delphi
Cette application va faciliter les activités de gestion d'achat et de vente des pièces de rechange 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 Delphi.

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] : Olivier Sigaud, Introduction à la modélisation orientée objets avec UML
Edition 2005-2006.
- [2] : Laurent Piechocki, cours UML.
- [- [3] : Frédéric Beaulieu, guide pascal et Delphi, Edition 2000, Mis à jour 2008.
- [4] : www.Developpez.com.
- [5] : Jérôme Darmont, Programmation sous Delphi, Edition 1999-2000.
- [6] : <http://fbeaulieu.developpez.com/guide>.
- [7] : <http://fr.wikipedia.org>.