



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 Dépôt de Produit Laitier (GDPL) Conception et réalisation d'une application

Réalisé par :

- Melle. AHMED DADDA WASSILA

Présenté le 10 Juin 2014 devant la commission d'examination composée de MM.

- *Mr.BENMOUNA Y* (Encadreur)
- *Mme.BENMANSOUR F* (Examineur)
- *Mr.MERZOUG.F* (Examineur)

Année universitaire: 2013-2014

Remerciement

Je remercie dieu de m'avoir donné une grande patience et beaucoup de courage pour mener à bien et avec effort l'énorme travail que j'ai effectué

Je tiens à exprimer mes remerciements avec un grand plaisir et un grand respect à mon encadreur Mr BENMOUNA.Y, pour ses conseils, sa disponibilité et ses encouragements qui m'ont permis de réaliser ce travail dans les meilleures conditions.

J'exprime de même ma gratitude à Mr AHMED DADDA.B. qui m'a beaucoup aidé et permis d'approfondir mes connaissances en gestion de stock.

J'exprime de même ma gratitude à Mr SETTOUTI Ahmed Khalid Yassine, qui m'a beaucoup aidé à la conception de ce système

Merci aux membres du jury, qui m'ont fait l'honneur d'assister à la soutenance de ce mémoire.

Je ne peux nommer ici toutes les personnes qui de près ou de loin m'ont aidé et encouragé mais je les en remercie vivement.

Enfin je tiens à dire combien le soutien quotidien de ma famille a été important tout au long de ces quelques années, je leur dois beaucoup.

Dédicace

A ma mère, pour tous les moments d'énervement, de tendresse, de tensions, d'écoute.
Qu'elle sache que je l'aimerai toujours coûte que coûte, et que je lui dois la vie que j'ai

Aujourd'hui.

A mon père, ce roc, cette montagne inébranlable qui sera pour toujours mon unique
modèle en toutes circonstances et qui a fait de moi ce que je suis. Merci de tout cœur.

A ma chère sœur, vous occupez une place particulière dans mon cœur. Je vous
dédie ce travail en vous souhaitant un avenir radieux, plein de
bonheur et de succès.

A ma petite chère sœur FATIMA ZOHRA et sa mère RABIHA

A FATIHA, AFAFE, FAYZA, SELMA, et FAYZA

A la famille MEZOUGH LILA, SOUHILA, HOUARI

A mon très chère amie SAMIA en souvenir de nos éclats de rire et des bons moments.

En souvenir de tout ce qu'on a vécu ensemble.

A MOHAMED ABDELMAJID

A AMINE, HAMADA, KAMEL, CHAREF.M,

A tous mes amis et connaissances

Je dédie ce mémoire

Table des matières

Remerciement

Dédicace

Introduction générale.....	3
Chapitre I présentation générale de système existant	5
I.1 Introduction	6
I.2 Quelques définitions	6
I.3 L'organigramme de dépôt de produit laitier	7
I.4 Présentation de la méthode du travail	7
Chapitre II la modélisation et la conception du système	9
II.1 Introduction	10
II.2 Définition UML.....	10
II.3 Historique de modélisation objet	10
II.4 Les diagrammes d'UML	12
II.4.1 Définition	12
II.4.2 les 14 types de diagrammes UML.....	12
a. Les diagrammes comportementaux (Behavior Diagram)	12
b. Les diagrammes structurels ou statiques (Structure Diagram)	13
c. Les diagrammes d'interaction ou dynamiques (Interaction Diagram)	15
II.5 Processus UP.....	16
a. Définition.....	16
b. les caractéristiques du processus unifié.....	16
c. Les activités et les phases.....	17
II.6 La conception du système GDPL.....	17
II.6.1 Diagramme de cas d'utilisation.....	17

II.6.2 Les diagrammes de séquence	18
II.6.3 Diagramme de classe.....	24
II.7 Modèle logique de système GDPL.....	25
II.8 Conclusion.....	25
Chapitre III l'implémentation du système GDPL.....	26
III.1 Introduction.....	27
III.2 Définition Delphi.....	27
III.3 Vocabulaire.....	27
III.4 Delphi 7.....	28
III.5 Présentation de l'application.....	28
a. Page d'authentification	28
b. Menu principal	29
c. Interface gestion du produit, client, et fournisseur.....	30
d. Les interfaces des fichiers.....	30
d.1 Interface bon de commande.....	30
d.2 Interface facture.....	31
d.3 Interface bon de livraison	31
e. Interface recherche client par numéro.....	32
f. Interface recherche produit par nom.....	32
g. Interface de paramètres (changer mot de passe).....	33
III.6 Conclusion.....	33
Conclusion générale.....	35
Liste de figures	37
Listes des références.....	38

Introduction générale

Introduction générale

Nous vivons dans un monde moderne et technologiquement avancé. L'ère de l'informatique a conduit à la mondialisation et un accès rapide à l'information et à la technologie. Les ordinateurs sont devenus un élément important de ménages et d'une partie invariable de l'éducation, des affaires de la profession et de l'industrie, des services. L'informatisation favorise la croissance rapide, moins d'erreurs et un meilleur service dans chaque domaine de la vie.

L'informatique de gestion se caractérise par la conception, le développement et la mise en œuvre d'applications informatiques dédiées au management de l'entreprise, au suivi des clients et aux relations avec les fournisseurs (facturation, comptabilité...). Elle est étroitement liée à l'ingénierie du système d'information de l'entreprise et des organisations.

En générale, en domaine de commerce les commerçants ont toujours le mal gestion du stock, un problème qui complique le système du travail, c'est un système manuel, ce dernier est plus facile à comprendre, mais il est le moyen le plus sujet à erreur et plus inefficace de stocker et d'extraire des données.

L'objectif de mon projet fin d'étude est de concrétiser ce que j'ai appris pendant les trois ans d'étude, la réalisation d'un logiciel de gestion d'un dépôt de produit laitier(GDPL).

Dans un premier temps, étude de la méthode générale du travail de GDPL, dans le chapitre I.

La deuxième partie, propose la modélisation du système GDPL, et détaille l'étape de conceptualisation en utilisant le langage UML, dans le chapitre II.

La troisième partie traite, de la finalisation de l'application et de son utilisation en situation de terrain, dans le chapitre III.

Chapitre I

Présentation générale du système existant

I.1 Introduction

La gestion des stocks est un domaine d'activité consommateur de techniques quantitatives. Le succès d'une organisation est déterminé, entre autres, par sa capacité de proposer le bon produit (ou service) au bon moment et au bon endroit. Un stockage intelligent contribue de manière décisive à cet objectif stratégique. On distingue, les différents types de stockage :

- Stocks de produits finis ;
- Stocks d'entrants de fabrication (matières premières, pièces sous-traitées);
- Stocks de pièces de rechange et de produits pour la maintenance des équipements de production ;
- Stocks d'outillages et d'accessoires ;

Le stock de produit finis est le cas d'étude de mon projet, sur un dépôt de produit laitier (SARL HODNA LAIT) géré par le propriétaire et gestionnaire Mr Ahmed Dadda .B, en l'occurrence. Ce dernier m'a accueilli dans son bureau et m'a donné toutes les informations, et explications concernant la méthode et la gestion de son système de travail.

I.2 Quelques définitions

– **Le dépôt** : méthode de vente ou lieu de vente, dans lequel la marchandise est laissée en dépôt et où le produit de la vente est partagé entre le propriétaire de la marchandise et le dépositaire.

– **La Commande** : est une intention, soit verbale soit écrite, d'engager une transaction commerciale pour des produits ou services particuliers

– **La facture** : est un document qui atteste de l'achat ou de la vente de biens ou services. C'est une pièce comptable par laquelle un créancier (généralement un fournisseur) établit une créance vis-à-vis d'un débiteur (le plus souvent son client) en vue de déclencher le paiement d'une dette.

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

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

I.3 L'organigramme de dépôt de produit laitier

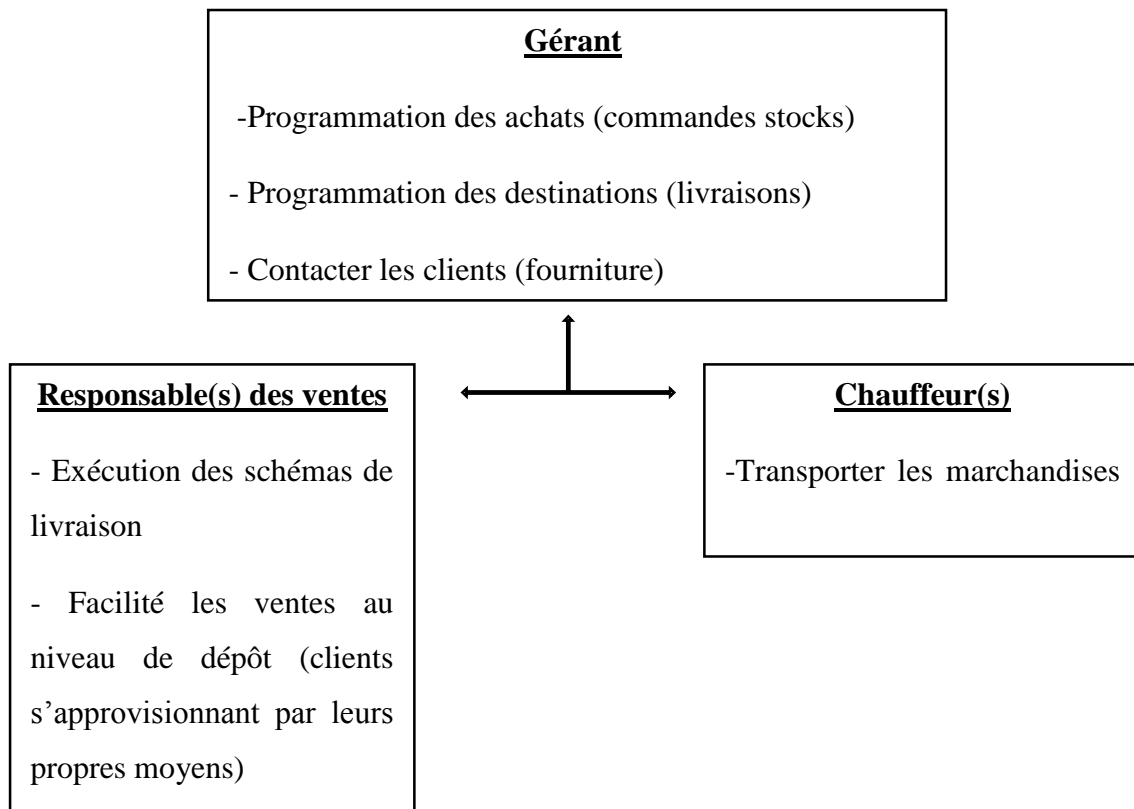


Figure I.1 l'organigramme de dépôt

I.4 Présentation de la méthode du travail

Adresse : cité El-Bordj n° 177. Wilaya de SAIDA.

Les employés :

Le propriétaire : Ahmed Dadda. B.

Les responsables des ventes, et les chauffeurs.

Dans ce dépôt, la méthode de travail se déroule comme suit :

Le gestionnaire, transmet un bon de commande au fournisseur (usine). Ce dernier procède à l'envoi de la marchandise commandée conformément au bon de commande émis.

A la livraison, la marchandise commandée est contrôlée par le gérant du stock, qui réceptionne suivant le bon de commande et la facture des produits transmise par le

fournisseur. Les produits manufacturés livrés sont stockés dans la chambre froide, par produit et par date.

Les clients, qui viennent s'approvisionner en produit laitier, doivent impérativement se présenter au niveau de la gestion des stocks et délivrer un bon de commande et prendre livraison de leurs besoins. Par ailleurs les clients, démunis de moyen de transports peuvent être satisfaits à domicile par les moyens du dépôt qui s'exécutera dès que le bon de commande est en sa possession. Le produit sera délivré en même temps que la facture et le bon de livraisons.

Si la méthode de travail décrite ci-dessus est très facile à mettre en œuvre, et très utilisée, elle est par contre très lente et présente certains aléas quand à la saisie, les calculs, les achats (bon), le stockage et les ventes (facturation).d'où la nécessité impérieuse d'utiliser les logiciels.

La réalisation d'un logiciel va nous permettre de mener à bon terme la gestion de notre dépôt aussi bien par :

- La saisie des informations
- Les calculs des résultats
- La gestion des achats, de stockage et des ventes

Ces derniers peuvent être exécutés dans un temps record grâce au logiciel développé.

Ce résultat net, rapide et précis nous montre la motivation du remplacement de la méthode initiale au profit du développement d'un logiciel.

Chapitre II

la modélisation et la conception du système

II.1 Introduction

Dans le cadre de ce chapitre, je vais définir quelques généralités portant sur la méthode et outils mettant en évidence la réalisation de notre projet. Nous allons commencer par présenter le langage de modélisation unifié UML (Unified Modeling Language).

II.2 Définition UML



Figure II.1 logo d'UML

UML (de l'anglais *Unified Modeling Language*), ou Langage de modélisation unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé en développement logiciel, et en conception orientée objet. **UML** est couramment utilisé dans les projets logiciels.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : **Booch**, **OMT**, **OOSE**. Principalement issu des travaux de **Grady Booch**, **James Rumbaugh** et **Ivar Jacobson**, il est à présent un standard défini par l'Object Management Group (**OMG**). 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. [1]

II.3 Historique des modélisations objet

La méthode Merise était parmi les premières à prendre en compte le point de vue des gestionnaires, en proposant des modèles pour représenter les aspects conceptuels (le sens des informations, la structuration du système de gestion) et organisationnels (choix d'informatisation, choix de répartition du travail...). Au début des années 90, les méthodes objets ont cherché à s'implanter. Succès mitigé : après avoir sérieusement

Investi sur une méthodologie, les entreprises freinent tout changement qui n'est pas solidement justifié. La donne commence à changer et UML (*Unified Modeling Language*, Langage de modélisation unifié) se présente aujourd'hui comme une alternative sérieuse.

De nombreux analystes ont mis au point des approches orientées objets, en 1994 il existait plus de 50 méthodes objet (**Booch, Classe-Relation, HOOD, OMT, OOA, OOD, OOM, OOSE etc...**). Toutefois seules 3 méthodes ont véritablement émergé :

- **OMT de James Rumbaugh** (Rational Software)

➤ fournit une représentation graphique des aspects statique, dynamique et fonctionnel.

- **OOD de Grady Booch** (General Electric)

➤ définie pour le département de défense, L'objectif était de préparer de façon rigoureuse la structuration des programmes écrits en langage ADA ou C++.

➤ Introduit le concept de package

- **OOSE d'Ivar Jacobson** (Ericsson)

➤ fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou "use cases").

L'unification a progressé par étapes :

– 1995, Booch et Rumbaugh (et quelques autres) se sont mis d'accord pour construire une méthode unifiée, "Unified Method 0.8"

– 1996, Jacobson les a rejoints pour produire UML 0.9 (noter le remplacement du mot "méthode" par le mot "langage").

– Les acteurs les plus importants dans le monde du logiciel s'associent à l'effort (IBM, Microsoft, Oracle, DEC, HP, Rational, Unisys etc.) et UML 1.0 est soumis à l'OMG ("Object Management Group", organisme international de normalisation en technologie objet qui rassemble les principales entreprises concernées). L'OMG adopte en novembre 1997 UML 1.1 comme langage de modélisation des systèmes d'information objet. Ce langage a passé par différents stades et est encore en évolution. [2]

Avantages d'UML

- Langage formel et normalisé
- Langage universel
- Gain de précision et de stabilité
- Utilisation d'outils (analyse cadrée)
- Facilité la compréhension des représentations abstraites et complexes (support de communication)

Inconvénients d'UML

- Apprentissage nécessaire (période d'adaptation)
- Ne constitue qu'une étape (n'est pas à l'origine des concepts objets)

II.4 Les diagrammes

II.4.1 Définition

Un diagramme est une représentation visuelle simplifiée et structurée des concepts, des idées, des relations, des données statistiques etc. employé dans tous les aspects des activités humaines pour visualiser et éclaircir la matière. Permet aussi de mettre en évidence des corrélations en certains facteurs ou de représenter des parties d'un ensemble.

II.4.2 14 types de diagrammes d'UML

a. Les diagrammes comportementaux (Behavior Diagram)

a.1 Les diagrammes de cas d'utilisation (Use Case Diagram)

Utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. [1]

✓ **Le cas d'utilisation** : sont des descriptions des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Les *use case* (cas d'utilisation) sont représentés par une ellipse sous-titrée par le nom du cas d'utilisation un acteur et un cas d'utilisation sont mis en relation par une association représentée par une ligne.

✓ **L'acteur** : est une entité externe au système, en interaction avec ce dernier. On distingue 4 catégories d'acteurs :

- Les acteurs principaux : Ceux qui vont réaliser le cas d'utilisation (ex : usager, client, etc.).
- Les acteurs secondaires : Ceux qui ne font que recevoir des informations à l'issue de la réalisation du cas d'utilisation (ex : opérateur de maintenance, administrateur, etc.).
- Le matériel externe : (capteur, imprimante, etc.).
- Les autres systèmes : (serveur, etc.).

✓ **Les relations** : Trois types de relations sont pris en charge par la norme UML et sont graphiquement représentées par des types particuliers de ces relations :

– **Les inclusions** : Dans ce type d'interaction, le premier cas englobe l'autre et son issue dépend souvent de la résolution du second. Elle est représentée par une flèche en pointillée et le terme « *include* ».

– **Les extensions** : représentent des prolongements logiques de certaines tâches sous certaines conditions. Elle est représentée par une flèche en pointillée avec le terme « *Extend* ».

– **Les généralisations** : est la relation de généralisation ou héritage, permet de définir la spécialisation d'un cas d'utilisation. Ces relations sont des traits pleins terminés par une flèche en triangle.

Il existe aussi :

Diagramme d'activité (Activity Diagram)

Diagramme états-transitions (State Machine Diagram)

b. Les diagrammes structurels ou statiques (Structure Diagram)

b.1 Le diagramme de classes

Est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. C'est l'outil essentiel permettant de décrire la structure d'un système.

✓ **Une classe** : décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe. Une classe est représentée par un rectangle séparée en trois parties :

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

✓ **Les attributs**

Un attribut est une propriété commune à tous les objets d'une classe. Pour chaque instance d'une classe ayant un attribut, l'instance possède cet attribut et cet attribut peut prendre une valeur pour une instance d'un objet.

✓ **Opérations et méthodes**

Une opération est associée à une classe ; c'est une fonction ou une transformation qui peut être appliquée aux objets d'une classe. Chaque opération possède un objet cible ou instance sur lequel elle s'applique.

✓ **la visibilité** : La visibilité d'une caractéristique détermine si d'autres classificateurs peuvent l'utiliser. En UML, on détermine 3 niveaux de visibilité :

-public : Toutes les autres classes ont accès à cet attribut.

-protected : Seules la classe elle-même et les classes filles (héritage) ont accès à cet attribut.

-private : Seule la classe elle-même a accès à cet attribut.

✓ **Les interfaces**

Une interface est un ensemble d'opérations utilisées pour décrire un service d'une classe ou d'un composant.

✓ **Les collaborations** : la collaboration désigne un ensemble de classes, d'interfaces et d'autres éléments qui travaillent ensemble et fournissent un comportement de coopération plus important que la somme de toutes ces parties.

✓ **Association** : une association est une relation générique entre deux classes. Elle est modélisée par une ligne reliant les deux classes.

✓ **Composition** : si une classe ne peut pas exister par elle-même, mais doit être un membre d'une autre classe, alors elle possède une relation de composition avec la classe contenant. Une relation de composition est indiquée par une ligne avec un "diamant" rempli

✓ **Dépendance** : quand une classe utilise une autre classe, par exemple comme membre ou comme paramètre d'une de ces fonctions, elle "dépend" ainsi de cette classe. Une relation de dépendance est représentée par une flèche pointillée.

✓ **Agrégation** : les agrégations indiquent une relation de contenant contenu. Elle est représentée par une ligne avec un "diamant" creux.

✓ **Généralisation** : une relation de généralisation est l'équivalent d'une relation d'héritage. Une relation de généralisation est indiquée par une flèche creuse se dirigeant vers la classe "parent".

– il y'a aussi :

Diagramme d'objets (Object diagram)

Diagramme de composants (Component diagram)

Diagramme de déploiement (Deployment diagram)

Diagramme des paquetages (Package diagram)

Diagramme de structure composite (Composite Structure Diagram)

Diagramme de profils (Profile diagram)

c. Les diagrammes d'interaction ou dynamiques (Interaction Diagram)

c.1 Diagramme de séquence (Sequence Diagram)

Représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs ou objets. Plusieurs types de messages (actions) peuvent transiter entre les acteurs et objets. (Message simple, avec durée de vie, message synchrone, message asynchrone, message déroband).

✓ **Cadre d'interactions** : pour les cas plus complexes, on peut intégrer des algorithmes dans les diagrammes de séquences. Par le biais de cadres d'interaction, on peut préciser les spécificités d'un ensemble de messages :

- **alt** : fragments multiple alternatifs (si alors sinon)
- **opt** : fragment optionnel
- **par** : fragment parallèle (traitements concurrents)
- **loop** : le fragment s'exécute plusieurs fois
- **region** : région critique (un seul thread à la fois)
- **neg** : une interaction non valable

- **ref** : référence à une interaction dans un autre diagramme
- **sd** : fragment du diagramme de séquence en entier

– On peut aussi voir :

Diagramme de communication (Communication Diagram)

Diagramme global d'interaction (Interaction Overview Diagram)

Diagramme de temps (Timing Diagram) [1]

II.5 processus UP

a. Définition

C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

b. les caractéristiques du processus unifié

UP est itératif, elle qualifie un traitement ou une procédure qui exécute un groupe d'opérations de façon répétitive jusqu'à ce qu'une condition bien définie soit remplie.

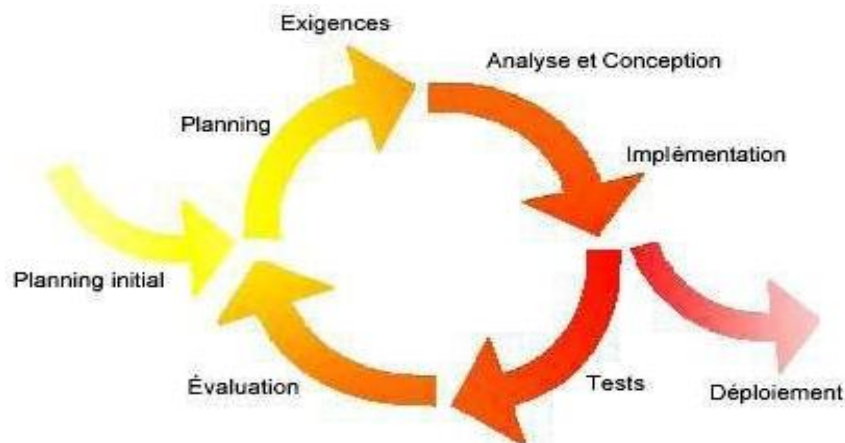


Figure II.2 le déroulement du processus UP

Il est centré sur l'architecture, décrit des choix stratégiques qui déterminent en grande partie les qualités du logiciel (adaptabilité, performance, fiabilité...).

Il est piloté par les cas d'utilisation d'UML, Le but principal d'un système informatique est de satisfaire les besoins du client.

c. Les activités et les phases

c.1 Les activités

- Expression des besoins
- Analyse
- Conception
- Test

c.2 Les phases

- Analyse des besoins
- Elaboration
- Construction
- Transition

II.6 la conception du système GDPL

Dans cette section, nous présentons quelques diagrammes de notre modélisation en explicitant leurs rôles. Nous allons principalement présenter le diagramme des cas d'utilisation, le diagramme de séquences et le diagramme des classes.

II.6.1 Diagramme de cas d'utilisation

Les cas d'utilisation et les acteurs sont schématisés dans le diagramme suivant (**Figure II.3**).



Figure II.3 Diagramme de cas d'utilisation « gestion d'un dépôt de produit laitier »

II.6.2 Les diagramme de séquences

Détaille le fonctionnement de chaque cas d'utilisation trouvé dans le diagramme précédent

✓ le cas d'utilisation « connecter »

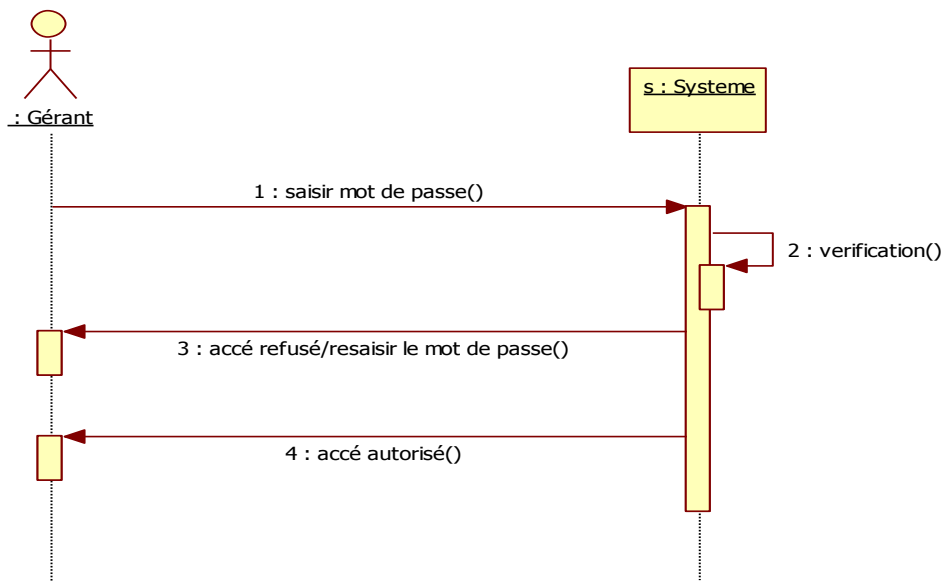


Figure II.4 diagramme de séquence « connecter »

✓ le cas d'utilisation « Manager client »

- Ajouter client

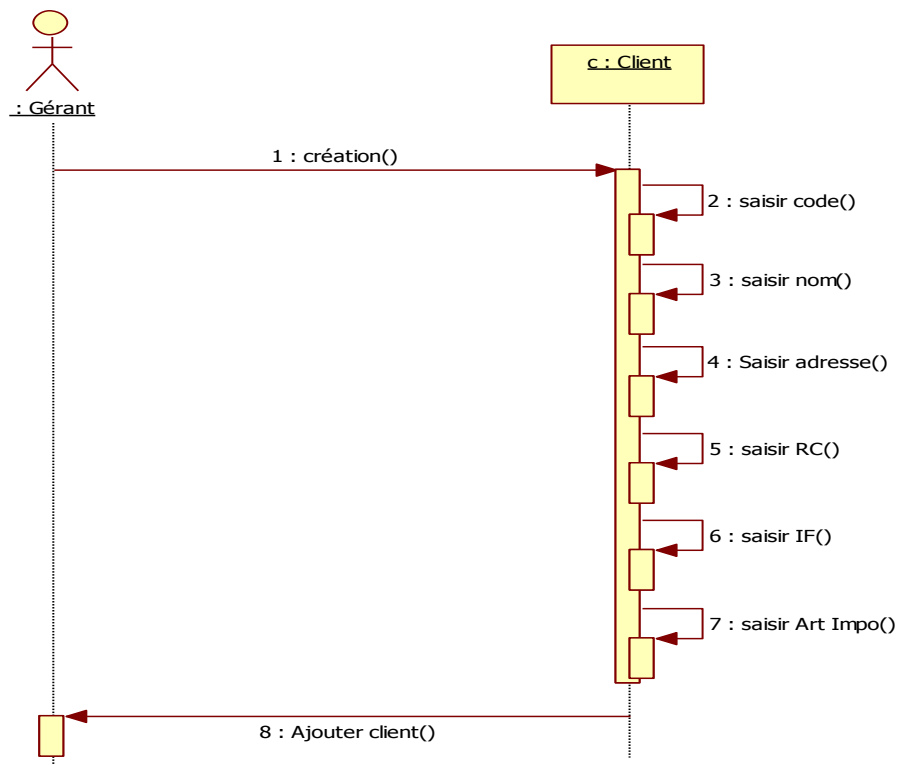


Figure II.5 diagramme de séquence « Ajouter client »

- « Modifier client »

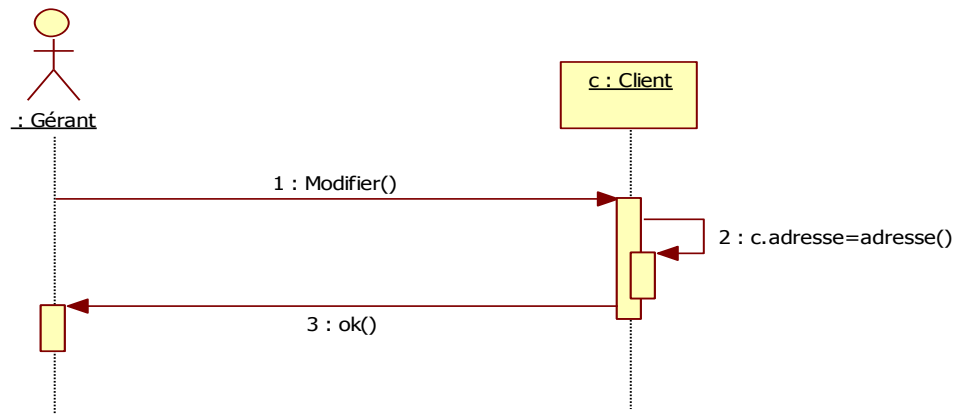


Figure II.6 diagramme de séquence « Modifier client »

✓ le cas d'utilisation « Manager fournisseur »

- « Ajouter fournisseur »

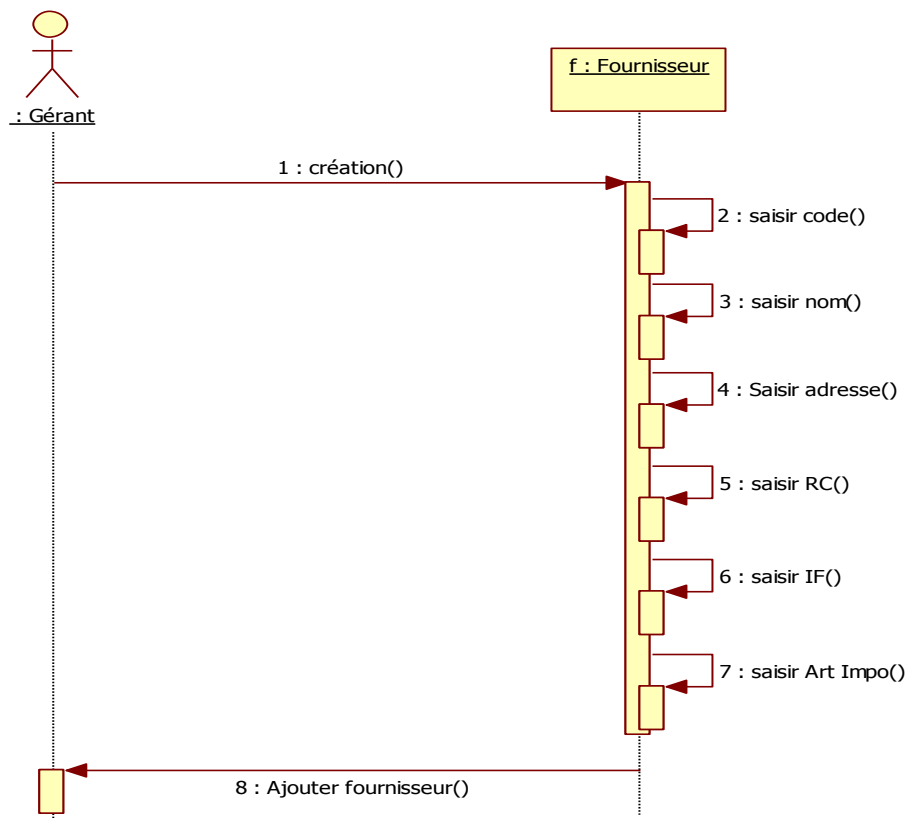


Figure II.7 diagramme de séquence « Ajouter fournisseur »

- « Modifier fournisseur »

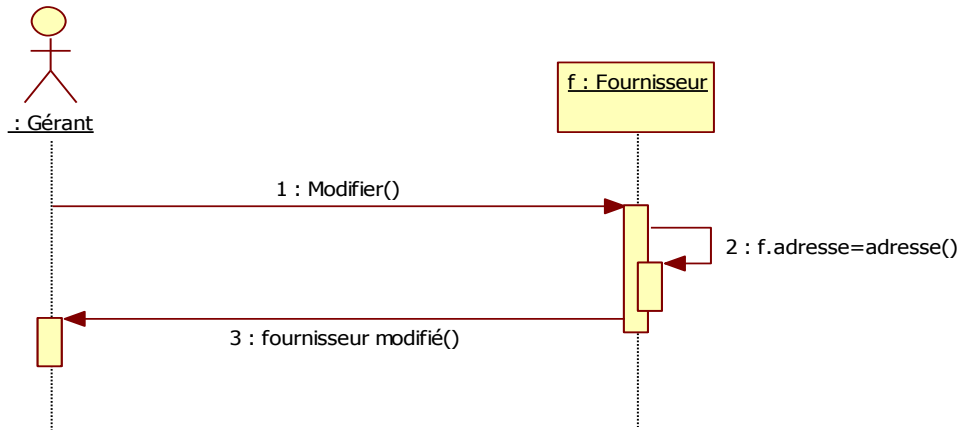


Figure II.8 diagramme de séquence « modifier fournisseur »

✓ Le cas d'utilisation « Manager produit »

- « Ajouter produit »

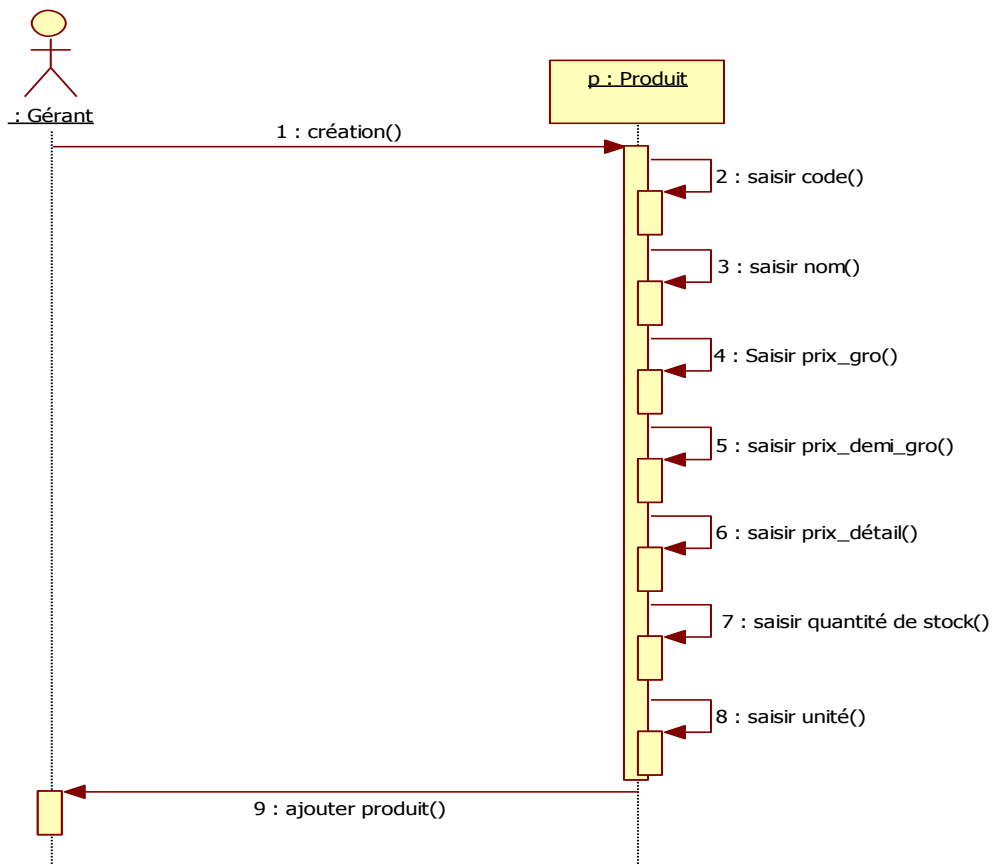


Figure II.9 digramme de séquence « ajouter produit »

- « Saisir prix »

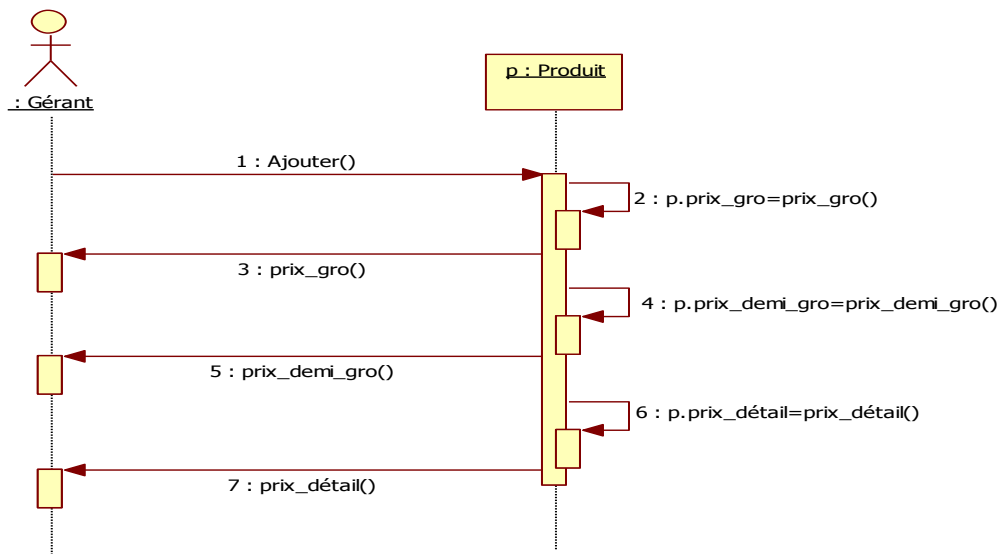


Figure II.10 diagramme de séquence « saisir prix »

- « Modifier produit »

- « Modifier quantité »

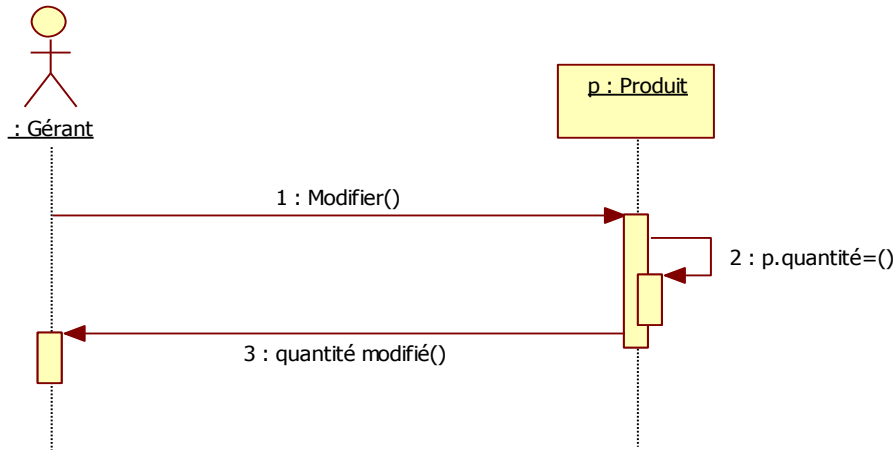


Figure II.11 diagramme de séquence « Modifier quantité de produit »

- « Modifier prix »

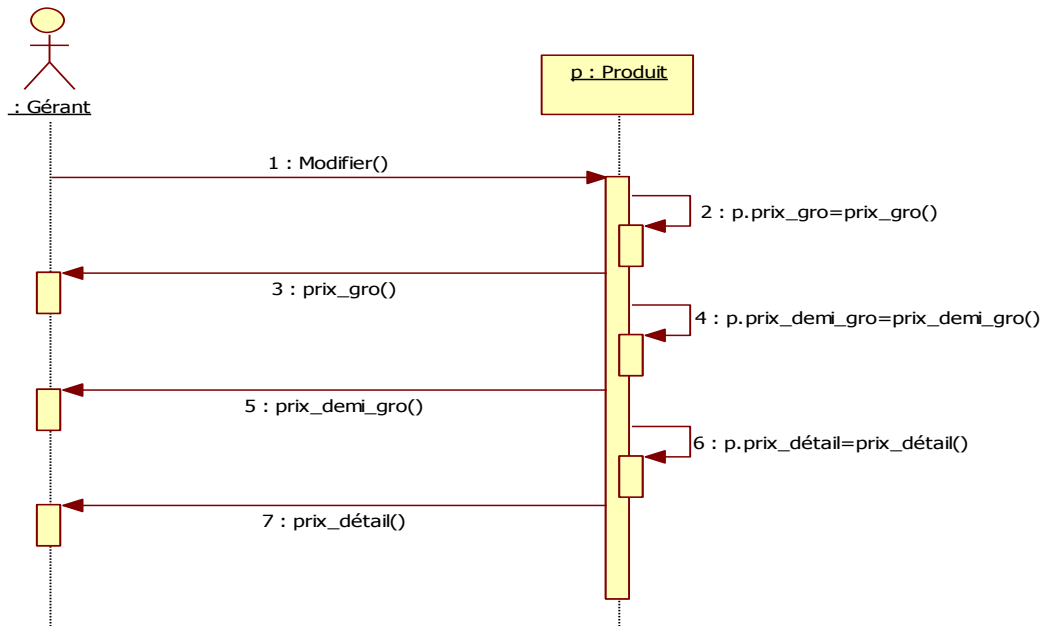


Figure II.12 diagramme de séquence « modifier prix de produit »

✓ le cas d'utilisation « commander »

- « création de commande »

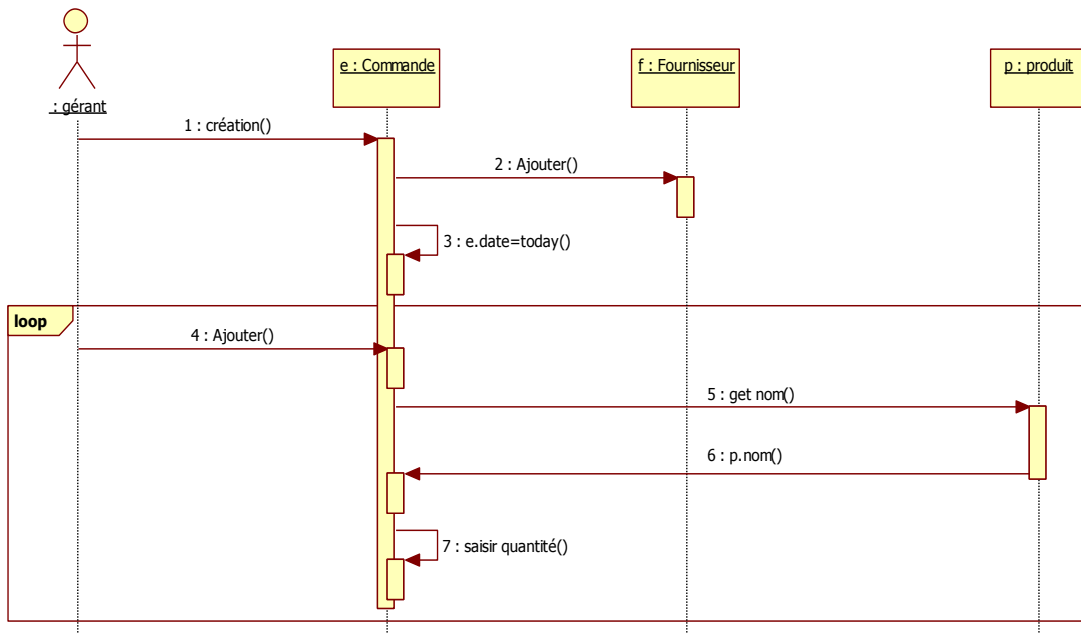


Figure II.13 diagramme de séquence « création de commande »

✓ Le cas d'utilisation « vendre »

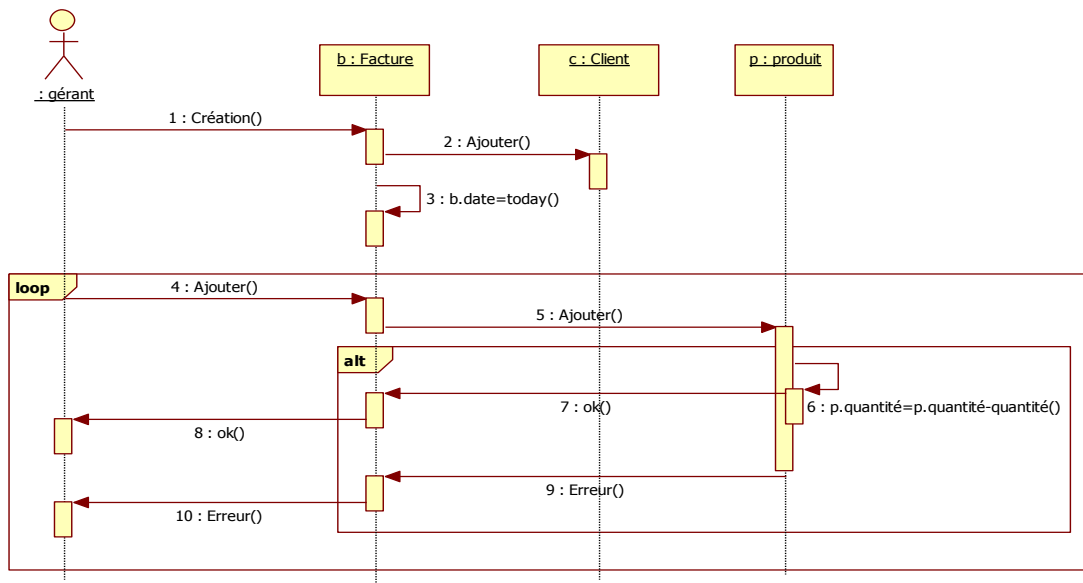


Figure II.14 digramme de séquence « création de facture »

II.6.3 diagramme de classe

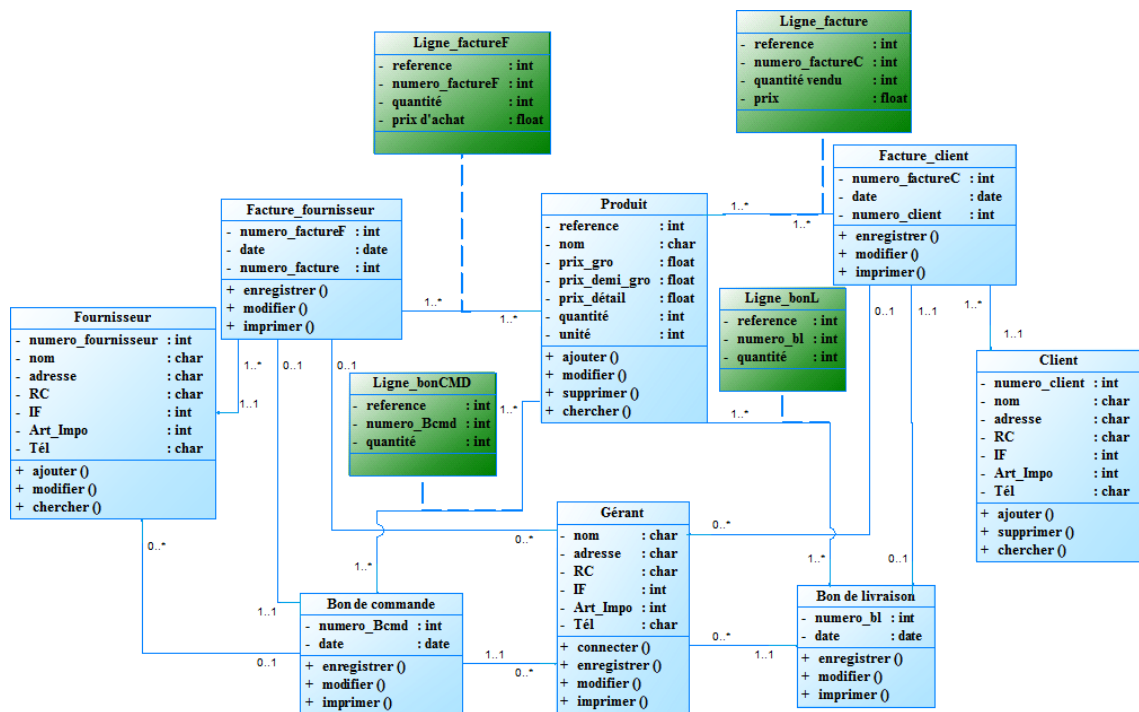


Figure II.15 Diagramme de classe

II.7 le modèle logique de système GDPL

Produit (Reference, Nom, Prix_gro, Prix_demi_gro, Prix_détail, Quantité, Unité)

Gérant (Nom, Adresse, RC, IF, Art_Impo, Tél)

Fournisseur (Numéro_fournisseur, Nom, Adresse, RC, IF, Art_Impo, Tél)

Client (Numéro_client, Nom, Adresse, RC, IF, Art_Impo, Tél)

Bon_De_Commande (Numéro_Bcmd, Date, Numéro_factureF*, Nom*)

Bon_de_livraison (Numero_bl, Date, Numéro_factureC*, Nom*)

Facture_fournisseur (Numéro_factureF, Date, Numéro_fournisseur*)

Facture_client (Numéro_factureC, Date, Numéro_client*)

Ligne_bonCmD (Numéro_Bcmd,Reference, quantité)

Ligne_bonL (Numéro_bl, Reference, Quantité)

Ligne_factureF (Numéro_factureF,Reference, Quantité, Prix d'achat)

Ligne_factureC (Numéro_factureC,Reference, Quantité vendu, Prix)

II.8 Conclusion

Tout au long de ce chapitre j'ai défini le langage de modélisation UML et traité la phase de conception et je l'ai décrite dans la plus part de ses facettes. Ceci est à travers des diagrammes de cas d'utilisation (Use Case), de séquences et de classes. Une fois la conception est faite, nous arrivons à la phase de développement et de réalisation de l'application qui doit respecter les directives de la conception.

Chapitre III

L'implémentation du système GDPL

III.1 Introduction

Dans ce chapitre je vais m'intéresser à la présentation de l'environnement matériel et logiciel utilisés pour assurer la réalisation de l'application. Il s'agit en plus de décrire les étapes de mise en œuvre de l'application ainsi que les différentes interfaces permettant l'interaction entre l'utilisateur et le système à développer et décrivant les différentes phases suivies pour la réalisation. Cette dernière est réalisée par le langage de programmation pascal objet en utilisant le delphi7.

III.2 Définition Delphi

Delphi est le nom d'un logiciel actuellement largement employé pour créer des logiciels. Il permet d'utiliser le langage Pascal. Pascal et Delphi NE SONT PAS une seule et même chose :

- Pascal est un langage informatique,
- Delphi est un logiciel destiné à créer des logiciels avec ce langage.

Delphi n'est qu'un enrobage, une enveloppe de confort autour de Pascal, c'est-à-dire qu'il simplifie de nombreuses tâches liées à la programmation en langage Pascal. Et est destiné à écrire des programmes fonctionnant exclusivement sous Windows.

Delphi est un environnement graphique de développement et de maquettage rapide.

La notion de projet est essentielle : un projet permet de regrouper le programme principal ainsi que toutes les fiches (fenêtres) et unités utilisées. Un projet Delphi contient au minimum un programme principal. A la création du projet le programme principal est automatiquement accompagné d'une fiche (une fenêtre vide) et d'une unité (le code associé à la fiche). [3]

III.3 Vocabulaire

« **Programme** » : texte écrit dans un langage informatique, comportant dans notre cas des instructions structurées.

« **Développer en Delphi** » : écrire des programmes en utilisant Pascal.

« **Application** » : Logiciel fonctionnant sous Windows.

« **Projet** » : c'est la base d'une application. Sous Delphi, pour créer une application, on constitue d'abord un projet, constitué de divers morceaux.

« **Code** », « **Code Pascal** », « Code source » : morceau de programme, texte d'un programme écrit en Pascal.

« **Interface (utilisateur)** » : la partie d'un logiciel qui est visible par l'utilisateur, à l'opposé du code source, invisible à l'utilisateur.

« **Fiche** » : fenêtre à l'état non compilé. Les fiches sont les alter egos sous Delphi des fenêtres sous Windows. [3]

III.4 Delphi 7

La version 7 utilisée pour écrire les exemples est la dernière disponible sur Windows, mais tous les exemples sont écrits avec les fonctionnalités générales de Delphi ce qui permet de les compiler sur n'importe quelle version de Delphi depuis la version 5.

III.5 Présentation de l'application

Dans cette partie, je présente les principales fonctionnalités de mon application à travers les interfaces réalisées.

a. Page d'authentification

Un utilisateur ne pourra accéder au système que s'il s'identifie en indiquant son mot de passe. Une fois l'utilisateur s'est authentifié,



Figure III.1 interface d'authentification

Si le mot de passe est incorrect le système affiche un message : "le mot de passe incorrect" fig. sinon le système affiche le menu principal.



Figure III.2 Interface d'authentification « cas mot de passe incorrect »

b. Menu principal

Cette interface représente la page d'accueil de l'application offrant à l'utilisateur les 6 principaux choix : fichier, client, produit, fournisseur, recherche, et paramètres.



Figure III.3 interface de menu principal

c. Interface Gestion de produit, client, et fournisseur

Cette interface permet à l'utilisateur d'effectuer les différentes opérations : ajout, modification, enregistrement, annulation, suppression, et impression.

The screenshot shows the 'Gestion de produit' window. On the left is the Hodna Lait logo. On the right, there are input fields for product details:

- Référence: 2
- Nom: FRIANDISE 100g
- Prix gros: 12,39
- Prix demi gro: 12,25
- Prix détail: 12
- Prix d'achat: 12,39
- Quantité de stock: 6000
- Unité: 48

Below these fields is a table with the following data:

Reference_article	Nom_article	Prix_gro	Prix_demi_gro	Prix_détail	Prix_d_achat	Quantité_de_stock
2	FRIANDISE 100g	12,39	12,25	12	12,39	
3	FLANIA CARAMEL 100 g	10,04	10	10	10,04	
4	CREME DESSERT 80g	7,05	7	6,5	7,05	

At the bottom, there are buttons for 'enreg', 'annuler', 'supprimer', and 'fermer'.

Figure III.4 Interface de gestion de produit

d. Les interfaces des fichiers

d.1 Interface bon de commande

The screenshot shows the 'Fichier' window with tabs for 'Bon de commande', 'Facture', and 'Bon de livraison'. The 'Bon de commande' tab is active. It contains the following fields:

- Emis par: AHMED DADDA BAGHDADI
- Adresse: CITE AOUI MILOUD N°40 SAID.
- Tél: 0553090012
- FOURISSEUR: HODNA LAIT
- N° B C: 2
- DATE B C: 27/05/2014

Below these fields is a table with the following data:

N° cmd four	DESIGNATION	Quantité	Unité	TOTAL
*	2 FLANIA CARAMEL 100 g	100	40	4000

At the bottom, there are buttons for 'Enreg', 'Annuler', 'Supprimer', 'Imprimer', and 'Quitter'.

Figure III.5 Interface de bon de commande

d.2 Interface facture

Bon de commande | Facture | Bon de livraison

AHMED DADDA MOHAMMED
HAI SALAM 2 SAIDA

13A1542789
215468436542845
12469646476

FACTURE N°: 165
SAIDA LE : 29/05/2014

AHMED DADDA BAGHDADI
CITE AOUI MILOU N°40 SAIDA

3720217A05
198346230068322
20800910872

NUMERO	REFERENCE	DESIGNATION	QUANTITE	P.U	PRIX TOT
165	2	FRIANDISE 100g	90	12	1080
165	3	FLANIA CARAMEL 100 g	100	10	1000
165	4	CREME DESSERT 80g	80	6,5	520

MONTANT HT : 2600
TVA 17% : 442
TIMBRE : 2440
NET A PAYER : 5482

ENREG ANNULER SUPP Imprimer FERMER

Figure III.6 Interface de facture

d.3 Interface Bon de livraison

Bon de commande | Facture | Bon de livraison

BON DE LIVRAISON

N° 56
DATE DE LIVRAISON 26/05/2014
LIVRE A Mr: AHMED DADDA MOHAMMED

Reference_article	DESIGNATION	Quantité	P.U	TOTAL
2	FRIANDISE 100g	90	12	1080
3	FLANIA CARAMEL 100 g	120	10	1200

Enreg Annuler Supprimer Imprimer Quitter

Figure III.7 interface de bon de livraison

e. Interface recherche client par numéro

Numero_cl	Nom_cl	Adresse	RC	I.F
1	MAEROUF SAID	L1 N°42 SAIDA	1426478A94	26
2	SAHRAOUI MOHAMM	CITE SID CHIKH N°21	254587F06	12
3	AHMED DADDA MOE	HAI SALAM 2 SAIDA	13A1542789	21

Figure III.8 Interface de recherche client par numéro

f. Interface recherche produit par nom

Reference_article	Nom_article	Prix_gro	Prix_demi_gro	Prix_détail	Prix_d_achat	Quantité_de
3	FLANIA CARAMEL 100 g	10,04	10	10	10,04	

Figure III.9 interface recherche de produit par nom

g. Interface de paramètres (changer mot de passe)

The screenshot shows a web browser window titled "changer mot de passe". The interface includes three text input fields for password entry. The first field, labeled "ENTRER LE MOT DE PASSE ACTUEL", contains the text "wassi". The second field, labeled "ENTRER LE NOUVEAU MOT DE PASSE", contains "2014". The third field, labeled "CONFIRMER LE NOUVEAU MOT DE PASSE", also contains "2014". Below the input fields are two buttons: "CONFIRMER" and "ANNULER". The background of the page features a white hand holding a bunch of blueberries with green leaves. In the bottom left corner, there is a logo for "Hodna lait" and in the bottom right corner, the URL "www.hodna-lait.com" is visible.

Figure III.10 interface changement de mot de passe

III.6 Conclusion

Tout au long de ce chapitre, j'ai présenté les différents outils de travail, les principales interfaces relatives aux principales fonctionnalités de mon système.

Conclusion générale

Conclusion générale

Au cours de cette expérience professionnelle, j'ai démontré que j'étais apte à m'adapter à tout type de tâche informatique. De plus j'ai su rapidement m'intégrer dans une équipe ou dans un nouvel environnement de travail.

Mon travail est débuté par la compréhension du contexte de mon projet. Ensuite, j'ai réalisé une étude générale du système existant, ce qui m'a permis de fixer les aléas à éviter et les objectifs à réaliser pour avoir un système satisfaisant.

Puis, j'ai accédé à l'étude conceptuelle de mon application selon une approche orientée objet tout en me basant sur le langage UML et le processus UP. Ces derniers ont été largement commentés et leur utilité pour la mise en œuvre dans la modélisation des diagrammes (diagramme de cas d'utilisation, diagrammes de séquence, diagramme de classe).

Par la suite, j'ai effectué le codage et l'implémentation de l'application.

Liste des figures

CHAPITRE I

Figure I.1 : l'organigramme de dépôt.....7

CHAPITRE II

Figure II.1 : Logo d'UML.....10

Figure II.2 : Déroulement du processus UP.....16

Figure II.3 : Diagramme de cas d'utilisation « gestion d'un dépôt de produit laitier »...18

Figure II.4 : Diagramme de séquence « connecter ».....19

Figure II.5 : Diagramme de séquence « Ajouter client ».....19

Figure II.6 : Diagramme de séquence « Modifier client »20

Figure II.7 : Diagramme de séquence « Ajouter fournisseur ».....20

Figure II.8 : Diagramme de séquence « modifier fournisseur ».....21

Figure II.9 : Diagramme de séquence « ajouter produit ».....21

Figure II.10 : Diagramme de séquence « saisir prix »22

Figure II.11 : Diagramme de séquence « Modifier quantité de produit ».....22

Figure II.12 : Diagramme de séquence « modifier prix de produit ».....23

Figure II.13 : Diagramme de séquence « création de commande ».....23

Figure II.14 : Diagramme de séquence « création de facture ».....24

Figure II.15 : Diagramme de classe.....24

CHAPITRE III

Figure III.1 : Interface d'authentification.....28

Figure III.2 : Interface d'authentification « cas mot de passe incorrect ».....29

Figure III.3 : Interface de menu principal.....29

Figure III.4 : Interface de gestion de produit.....30

Figure III.5 : Interface de bon de commande.....	30
Figure III.6 : Interface de facture.....	31
Figure III.7 : Interface de bon de livraison.....	31
Figure III.8 : Interface de recherche client par numéro.....	32
Figure III.9 : Interface recherche de produit par nom.....	32
Figure III.10 : Interface changement de mot de passe.....	33

Listes des références

[1] : http://fr.wikipedia.org/wiki/UML_%28informatique%29.

[2] : <http://www.uml.org/>

[3] : <http://repo.zenk-security.com/Programmation/Presentation%20delphi.pdf>