REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE : Mathématique et Informatique

DEPARTEMENT : Informatique

N °: ……

DOMAINE : Mathématique et Informatique

FILIERE : Informatique

OPTION : IA

Mémoire Présenté Pour L'obtention Du Diplôme De Master Professionnel Par :

- **BACHIR LINDA YASSAMINE**
- **TOUATI SAFWAT AMEL**

THEME

La réingénierie d'un logiciel

Soutenu devant le jury composé de :

Mr.	**MOKHTARI RABEH**	Université de M'sila	Président
Mr.	**BOUNIF MOHAMED**	Université de M'sila	Encadreur
Mr.	**BOUGHERARA SEDIK**	Université de M'sila	Examinateur

Année universitaire : 2023/2024

REMERCIEMENTS

Nous tenons à exprimer nos gratitudes envers dieu pour toutes les bénédictions
Qu'il nos avoir accordées dans la vie, Chaque jour nous sommes reconnaissants
Pour sa guidance pour réaliser notre mémoire.

Nous exprimons notre gratitude envers l'université de ***MOHAMED BOUDIAF***
Et le département de ***L'INFORMATIQUE*** *pour le soutien précieux qu'ils nous*
Ont apporté tout au long de notre parcours académique, notamment lors de
La réalisation de notre projet de fin d'études.

Merci à notre encadreur Dr : ***BOUNIF MOHAMED de*** *nous avoir*
Guidée au long de toute nos travaille et de son aide et le soutien qu'il nos apportée.

Merci à nos parents qui nous ont toujours soutenu et encouragé au cours
De la réalisation de ce mémoire.

Merci à nos chères collègues de la promo IA 2024 pour leur soutien
Et leur collaboration précieuse.

Enfin, nous exprimons notre gratitude envers tous ceux qui ont apporté
Leur contribution, que ce soit de près ou de loin.

LINDA YASSAMINE

&

SAFWAT AMEL

Dédicace

Je dédie ce travail à :

La femme qui a tout ***sacrifiée*** *pour nous et qui nous à donner la vie*

La femme la plus forte

La femme la plus douce

Mon espoir et mon esprit

Mon réconfort et ma certitude

La personne la plus chère ***à mes yeux***

Mon plus beau repère

Mon guide dans l'existence

Ma très chère mère ***ZAHIA***

Par toi j'ai grandi j'ai avoir la vie

Ta prière et ta bénédiction m'ont grandement apporté de l'aide pour réussir mes études.

Il n'y a pas de dédicace suffisamment éloquente pour exprimer ma gratitude pour

Tous les sacrifices que tu n'as cessé de me faire depuis ma naissance, pendant

Mon enfance et même à l'âge adulte.

Merci la plus adorables des mamans du monde.

À mes chères sœurs :

Khadidja, *Ma 2éme maman je voulais t'exprimer ma gratitude pour tout ce que tu accomplis pour moi. Ma vie est enrichie par ta présence, ton soutien et ton amour Inconditionnel. Merci d'être une source constante de joie et de bonheur dans ma vie Merci pour toutes les fois où tu as été ma saveur, Je t'en suis reconnaissante d'être ma sœur Je vous remercie pour tout*

Dina, *Mon ange, je suis reconnaissante de t'avoir comme petite sœur merci d'être toujours là pour moi, tu es un cadeau précieux dans ma vie, Merci pour tous les moments de bonheur que tu m'apportes*

Meriem, *Je te remercie du plus profond de mon cœur pour tout ce que tu fais pour moi*
Spécial dédicace à toi, ***Hamza,*** *qui a remplacé le père et le frère, je te remercie pour tous les moments où Tu as dessiné un sourire sur mon visage, pour tous les jours où tu as été le renfort de mes blessures et Pour toutes les fois où tu as répondu à mes appels sans hésitation*

À mon grand-père (Allah le bénisse) : Alioui Abderrahmane

*À mes chères cousines : **INES** ET **FADILA** pour m'avoir soutenu dans mes moments Difficiles*

*À mes chères copines : **ZINEB** ET **NOURHENE** est un baume pour mes blessures.*

*À ma moitié **Safwat Amel,** Pour tous les moments inoubliables qu'ont passé ensembles*

Tu es mon guide et mes bras, je t'aime.

À quelqu'un qui m'a aidé, je voulais remercier du fond du cœur pour tout ton soutien et Ton aide précieux.

LINDA YASSAMINE

Dédicaces

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

Mon cher père ***Touati Amhmed*** *Tu as toujours été pour moi un exemple de père respectueux, honnête et de personne méticuleuse, je tiens à honorer l'homme que tu es. Grâce à toi mon père, j'ai appris le sens du travail et de la responsabilité. Je sais que tu as souffert pour nous, Je tiens à vous remercier pour votre amour, votre générosité et votre compréhension je t'aime papa.*

À la femme qui a souffert sans me laisser souffrir ***Safi zahira****, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort, merci pour ton amour infini, ta force inébranlable et ta sagesse sans limite. Tu es mon roc, ma lumière et mon inspiration, Je t'aime plus que les mots ne puissent le dire, Merci pour tout maman.*

À ma sœur ***Dounia,*** *qui est plus qu'une sœur, mais aussi ma deuxième mère, même si je suis plus âgée qu'elle, Votre amour et votre soutien inconditionnels illuminent ma vie, Merci pour tout ce que vous faites, Je t'aime infinitive ment.*

À mon cher frère ***Anis****, mon partenaire de vie, que notre lien reste fort et inébranlable Avec tout mon amour et ma gratitude pour ta présence à mes côtés.*

À ma merveilleuse petite sœur ***Bouchra*** *et à mon petit frère* ***Mohamed Iyad****, qui savent toujours apporter joie et bonheur à toute la famille, je vous aime, mes enfants.*

Une dédicace spéciale pour ***un personne très important*** *dans ma vie, tu illumines ma vie de ta présence et de ton amour, mon bonheur et mon bras droit, Merci d'être toi toujours à mes côtés.*

À ma tante ***fifi*** *il n'y a pas de mots pour t'exprimer, merci pour tout.*

À tous les membres de ma famille et toute personne qui porte le nom ***Safi****, je dédie ce travail à tous ceux qui ont participé à ma réussi.*

À mes chers amis, que notre amitié perdure à jamais ***Chaïma, MALEK, Zineb****….*

Je n'oublie pas mon binôme ***Linda yassamine*** *préféré, merci d'être toujours là pour moi dans les bons moments comme dans les mauvais. Notre complicité est précieuse et je suis reconnaissante de t'avoir à mes côtés.*

Safwat Amel

Table des matières

Liste de figure

INTRODUCTION GÉNÉRALE :

Le génie logiciel est un domaine de l'informatique qui se concentre sur la conception, le développement et la maintenance de logiciels de qualité. Cette discipline fait appel à des méthodologies, des outils et des techniques spécifiques afin de gérer la complexité croissante des systèmes informatiques modernes. Les principaux enjeux du génie logiciel sont notamment d'assurer la fiabilité, la performance, la maintenabilité et l'évolutivité des applications. Les professionnels du secteur mettent en œuvre des processus de développement rigoureux, des pratiques de gestion de projet adaptées et des techniques d'analyse et de conception innovantes. Au-delà du développement initial, la maintenance et la réingénierie des logiciels jouent également un rôle essentiel dans cette discipline.

La maintenance des logiciels consiste à entretenir, à corriger et à améliorer les applications existantes afin de répondre aux besoins évolutifs des utilisateurs. Lorsque les défis de maintenance deviennent trop complexes ou que les technologies existantes atteignent leurs limites,

La réingénierie s'impose comme une solution stratégique. Ce processus de transformation en profondeur permet de moderniser les applications tout en conservant leurs fonctionnalités essentielles.

Il est bien connu que la majorité des logiciels subissent des modifications au cours de leur existence afin d'améliorer leur efficacité ou autres caractéristiques, ce qui a été le cas au sein de l'entreprise **CCLS MSILA** Où nous avons effectué notre stage, où nous avons été confrontés à des problèmes liés à l'utilisation d'une version ancienne du logiciel **Décompte**. Cette situation a mis en lumière les défis récurrents de la maintenance et de la réingénierie des applications existantes. Les logiciels vieillissants souffrent souvent d'une dette technique élevée, rendant leur évolution et leur adaptation aux nouveaux besoins particulièrement complexes. De plus, les coûts et les efforts nécessaires pour maintenir ces anciens systèmes peuvent devenir prohibitifs à long terme.

Cette problématique nous a motivés à trouver des solutions innovantes à l'aide de la réingénierie et de la maintenance pour ce logiciel Décompte. Ainsi, nous avons mis en place une approche innovante pour améliorer l'expérience utilisateur et les performances du système, tout en préservant les fonctionnalités essentielles.

Certains des problèmes associés à:

- Limitations de mémoire
- Absence de multitâche
- Pas d'interface graphique intégrée
- Limitations du système de fichiers
- Manque de sécurité

Bienvenue dans le monde fascinant du génie logiciel, où l'innovation et la créativité se conjuguent pour façonner les applications qui transforment nos vies numériques

Le projet nous permettra, les étudiants en informatique la maîtrise de nouveaux langages de Programmation, d'appliquer des méthodes théoriques sur des problèmes réels ainsi que de se Familiariser avec le monde des applications.

La structure est comme suit :

- Le premier chapitre s'agira d'une prise de connaissance concernât notre thème de mémoire donc on parle sur la réingénierie d'un logiciel plus la maintenance.
- Le second chapitre nous parlerons de l'entreprise dans laquelle nous avons fait un stage.
- Dans le troisième chapitre nous concentrons sur l'analyse du l'ancien logiciel.
- Dans le chapitre04 nous allons explorer en détail les méthodes utiliser pour la conception de notre nouveau logiciel.
- Le dernier chapitre c'est l'implémentation du logiciel donc nous présenterons également les outils Choisis et les diverses images du nouveau logiciel.

Chapitre 01 :
La réingénierie et la maintenance

INTRODUCTION :

Le génie logiciel et la réingénierie d'un logiciel sont des thèmes importants en informatique. En effet, la réingénierie d'un logiciel est la mise au point ou la modification d'un logiciel existant pour le rendre plus efficace, plus sûr ou plus facile à entretenir. Le génie logiciel est, lui, l'ensemble des procédés et des techniques de conception, développement et de maintenance de logiciels de haute qualité.

Nous allons examiner les liens étroits entre le génie logiciel et la réingénierie d'un logiciel dans ce chapitre. Nous examinerons comment les principes du génie logiciel peuvent être appliqués à la réingénierie pour améliorer la qualité et la fiabilité des logiciels existants. Nous évoquerons aussi les méthodes et les instruments qui peuvent être utilisés pour réaliser un projet de réingénierie, en insistant sur la planification et la gestion du processus. Nous allons parler sur plusieurs choses concernant le génie logiciel et la réingénierie d'un logiciel.

Les professionnels de l'informatique doivent affronter des défis et des enjeux lors de l'exécution d'un projet de réingénierie. Nous allons voir comment ces difficultés peuvent être résolues par une démarche systématique et rigoureuse, inspirée des bonnes pratiques en génie logiciel.

1. Génie logiciel

1.1 Historique :

À la fin des années 1960 éclate la « crise du logiciel », prise de conscience des difficultés que rencontre le développement des grands projets informatiques. De ce constat va naître une nouvelle discipline, le génie logiciel. Retour sur ses premières années d'existence. La fin des années 1950 voit la création des premiers langages de programmation de haut niveau. Dans la période euphorique qui suit, on pense que l'usage de ces nouveaux langages, qui viennent se substituer à l'assembleur, va résoudre les problèmes du développement des applications informatiques. Le terme de « programmation automatique », en vogue à l'époque, est représentatif de cet état d'esprit. Mais la réalité ne va pas tarder à se manifester…

1.2 Définition du GL :

Le Génie Logiciel (Software Engineering) est un domaine des sciences de l'ingénieur dont la finalité est la conception, la fabrication et la maintenance de systèmes logiciels complexes, sûrs et de qualité. C'est un ensemble de méthodes, techniques et outils pour la production et la maintenance de composants logiciels de qualité. Contrairement au développement artisanal (production individuelle d'un système simple), dans le cas du GL, il s'agit d'une production collective d'un système complexe concrétisée par un ensemble de documents de conception, de programmes et de jeux de tests avec souvent de multiples versions (multi-person construction of multi-version software).

1.3 Cycle de vie d'un logiciel :

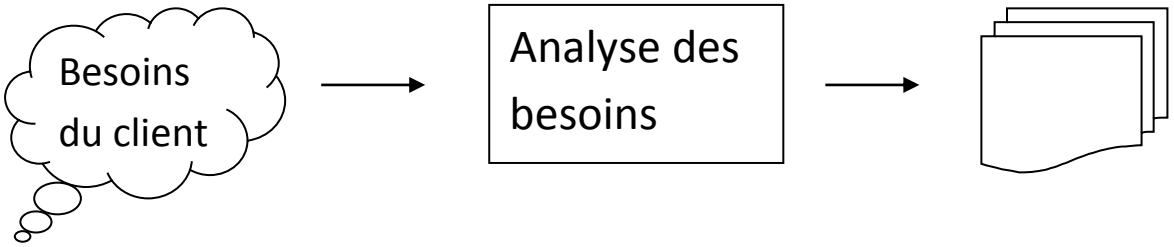
Le cycle de vie du logiciel modélise l'enchaînement des différentes activités du processus technique de développement du logiciel.

Une activité comprend : des tâches, des contraintes, des ressources, une façon d'être réalisée. Les grandes activités sont :

- Analyse des besoins
- Spécification globale
- Conception architecturel et détaillé
- Programmation
- Gestion de configuration et d'intégration
- Validation et vérification
- Livraison et maintenance

1.3.1 Analyse des besoins :

But : Eviter de développer un logiciel non adéquat. On va étudier le domaine d'application ainsi que l'état actuel et futur de l'environnement du système afin d'en déterminer : Les frontières, Le Rôle, Les ressources disponibles et requises, Les contraintes d'utilisation et performance … etc.



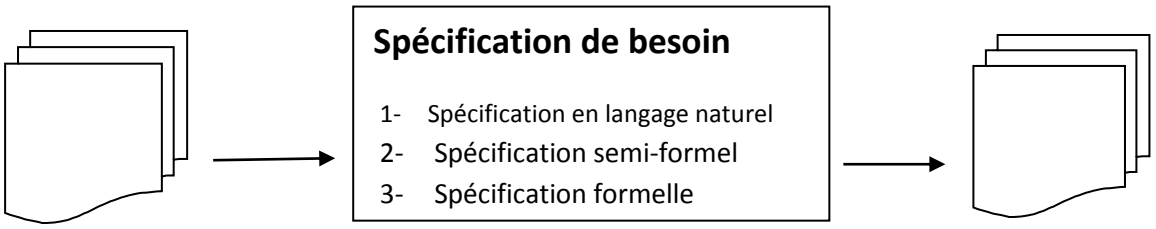
-Experts des domaines d'application
-Futur utilisateurs du système
-Observation de l'existence
-Etude de situation similaire

-Entretien
-Questionnaire
-préliminaire

- Cahier des charges
- Manuel d'utilisation

1.3.2 Spécification des besoins :

But : Etablir une première description du futur système. Pour produire une description de ce que doit faire le système mais sans préciser comment il le fait



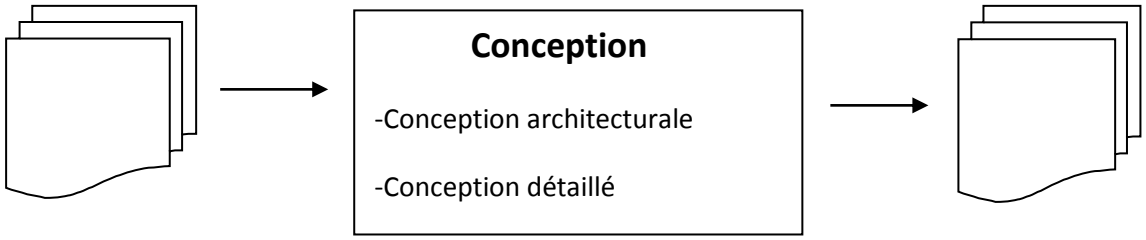
-Cahier des charges (CC)	- Modèle E/A	-CC fonctionnel
-Manuel d'utilisation	-Diagramme de flot de données	- Technique et faisabilité
Préliminaire	-Diagramme d 'état transition	
-Réseaux de pétri et les grafcet	- informatique	
	-Diagramme des méthodes O.O	

1.3.3 Conception :

But : Enrichir la description du logiciel de détails d'implémentation afin d'aboutir à une description très proche d'un programme (décrire le comment).

La conception architecturale (ou conception globale) a pour but de décomposer le logiciel en composants plus simples, définis par leurs interfaces et leurs fonctions (les services qu'ils rendent).

La conception détaillée fournit pour chaque composant une description de la manière dont les fonctions ou les services sont réalisés : algorithmes, représentation des données.

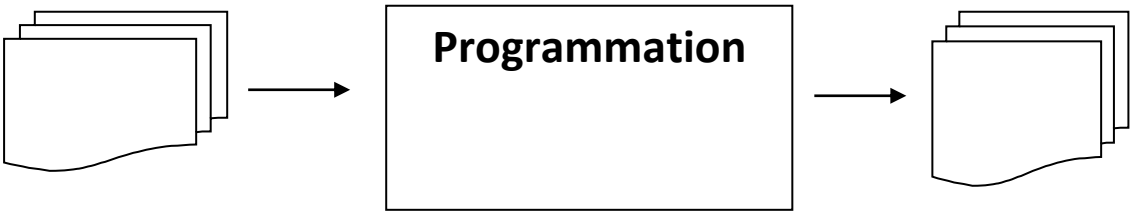


-Cahier des charges fonctionnel	- conception fonctionnelle et	- Dossier de conception :
-Technique et faisabilité	conception orienté objet	(Architecture et Algorithme)
Informatique	- Pour chaque composant	
- écrire les algorithmes		

1.3.4 Programmation :

Implantation de la solution conçue

Choix de l'environnement de développement, du/des langage(s) de programmation, de normes de développement...



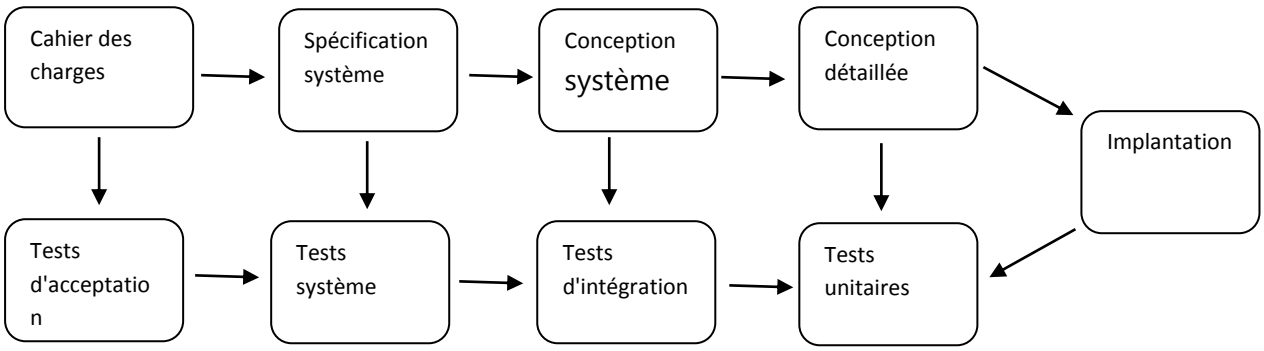
- Dossier de Conception:(Architecture et Algorithme)
- Code source
- Manuel final d'utilisateur
- Documentations

1.3.5 Gestion de configuration et intégration

- *La gestion de configurations* a pour but de maîtriser l'évolution et la mise à jour des composants tout au long du processus de développement.
- *L'intégration* a pour but de réaliser un ou plusieurs systèmes exécutables à partir des composants (Combiner les composants).

1.3.6 Validation et vérification

- *La validation* a pour but de répondre à la délicate question : a-t-on décrit le bon système, celui qui répond à l'attente des utilisateurs.
- *La vérification* répond à la question : le développement est-il correct par rapport à la spécification globale ? Ce qui consiste à s'assurer que les descriptions successives et le logiciel lui-même satisfont la spécification.



1.3.7 Maintenance :

Il s'agit d'apporter des modifications à un logiciel existant. C'est la phase la plus coûteuse (70% du coût total)

2. LA MAINTENANCE

2.1 Définition de la Maintenance :

La maintenance, ce n'est pas seulement la détection et la correction des erreurs trouvées dans un programme. On trouve différentes définitions de la maintenance logicielle. Des acteurs importants du domaine définissent l'activité de la façon suivante :

1-L'organisme de standardisation des logiciels, L'IEEE, définie dans la référence « software maintenance standard », IEEE STD 1219-1993, le terme de maintenance logiciel comme ceci : « la modification d'un logiciel après son entrée en production
Afin de corriger ses erreurs, d'améliorer ses performances et autres attributs, ou pour adapter le produit à son environnement ».

2- Martin et MC Colure, éminents théoriciens du domaine définissent la maintenance comme « Les changements effectués sur les programmes informatiques après livraison à leurs utilisateurs »

3- Von Mayrhauser : « La maintenance couvre le cycle de vie des systèmes logiciels à partir de leur mise en production jusqu'à la fin de leur utilisation.

Le thème commun de ces multiples définitions est que la maintenance est une activité qui poursuit la réalisation d'un travail. L'activité de maintenance commence dès la mise en production du logiciel….

2.2 Pourquoi la maintenance est nécessaire ?

Pour répondre à cette question, on se doit simplement d'observer la mise en production d'un logiciel livré à des utilisateurs. Lorsque les premiers utilisateurs vont utiliser les produits, ils vont trouver de nombreux problèmes, et disfonctionnements, et trouver d'innombrables fonctions à rajouter. Ces changements vont être demandés aux mainteneurs de l'application, qui vont corriger et améliorer le système. Puis les utilisateurs vont s'adapter aux changements et nouvelles fonctionnalités et à nouveau perturber les mainteneurs. Dans la plupart des cas, le cycle de maintenance d'un logiciel constitue la plus grosse part de son cycle de vie, en temps et en argent. Cette maintenance sans fin se justifie par le fait qu'une application doit sans cesse évoluée pour continuer à être utilisée, et ne pas mourir……

2.3 Les types de maintenance :

Il existe différents types de maintenance peuvent être combinés et adaptés en fonction des besoins spécifiques de chaque entreprise ou secteur d'activité.
Voici la figure suivante :

a) Maintenance préventive : consiste à effectuer des actions planifiées pour éviter les pannes et maintenir les équipements en bon état de fonctionnement.

b) Maintenance corrective : intervient après une panne pour réparer ou remplacer les équipements défectueux.

c) Maintenance prédictive : utilise des techniques de surveillance et d'analyse pour prédire les pannes et planifier les interventions avant qu'elles ne se produisent.

d)Maintenance conditionnelle : basée sur l'état réel des équipements, elle consiste à intervenir uniquement lorsque cela est nécessaire en fonction des données recueillies.

e) Maintenance curative : similaire à la maintenance corrective, elle vise à réparer rapidement les équipements en cas de panne critique.

f) Maintenance améliorative : vise à améliorer la performance et la fiabilité des équipements en mettant en place des actions d'amélioration continue.

3. LA REINGENIERIE d'un logiciel

3.1 Définition de la réingénierie (reengineering) :

La réingénierie d'un logiciel est le processus de modification, de reconstruction ou de réorganisation d'un logiciel existant afin d'améliorer sa qualité, sa performance, sa maintenabilité ou son évolutivité. Ce processus peut impliquer la refonte de l'architecture logicielle, la réécriture du code source, l'ajout de nouvelles fonctionnalités ou la suppression de fonctionnalités obsolètes.

L'objectif principal de la réingénierie logicielle est d'optimiser le logiciel pour répondre aux besoins actuels et futurs des utilisateurs tout en minimisant les coûts et les risques associés à ces modifications.

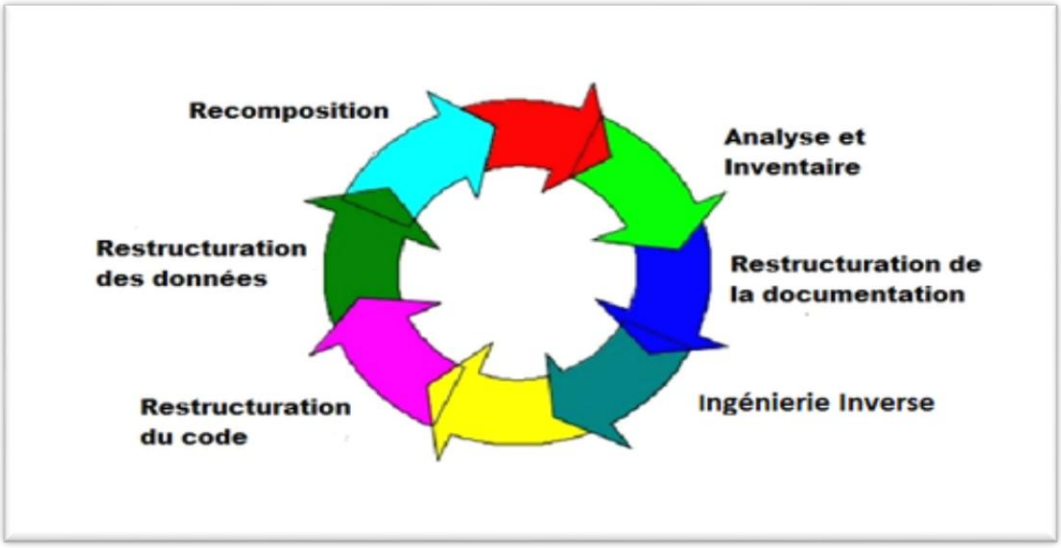


Figure 01 :la réingénierie

3.2 Définition de la rétro ingénierie :

Représente la première étape de la réingénierie qui consiste à effectuer le chemin inverse par rapport au processus de développement. On pourrait la définir comme le contraire de l'ingénierie autrefois nommée "forward engineering". Le reverse engineering est une partie indispensable de la maintenance logicielle, puisque celle-ci ne peut être réalisée sans une compréhension complète du système. Parfois, la compréhension d'un logiciel est si complexe en l'absence de toute documentation, que la tâche est presque impossible. C'est la raison pour laquelle la rétro-ingénierie est un processus qui a pour rôle de comprendre les différentes informations inconnues ou cachées du système logiciel…

3.3 La rétro ingénierie et réingénierie :

La rétro-ingénierie vise à automatiser l'extraction et l'exploitation d'informations à propos d'un système dans le but d'en améliorer la qualité et l'utilisation.

- Analyse et compréhension du code (générateurs de traces de donnée, de structures logiques ; ...etc.)
- Re documentation du code (aide à la modularisation; trace logique ou descriptive).
- Métriques (complexité, structure, control).
- Analyses de test (générateurs de données test; analyseurs de tests de couverture; tests non régressifs; debugger; comparateurs de code).
- Gestion de configuration (gestion des versions, du changement, etc.).

La réingénierie vise à automatiser l'extraction et l'exploitation d'information à propos d'un système dans le but de reconcevoir tout ou une partie d'un système existant.

- Restructuration et re modularisation (restruction logique des traitements, standardisation des noms de données et leur définition, reformatage du code).
- Migration vers de nouvelles technologies (reconception du code, reconception architecturale).
- Migration vers de nouvelles stratégies (formation, migration de plateforme, de systèmes, etc.).

3.4 Disciplines de la réingénierie :

3.4.1 La réingénierie du Système d'information :

Il s'agit d'une remise à plat de tout ou partie d'un système d'information, afin d'atteindre de meilleurs niveaux de performances globales

3.4.2 La ré ingénierie du processus :

Comme le système d'information à sa part de cette discipline les processus opérationnels vont donc à leur tour revu et, le plus souvent, refondus, réinventés à fin que les objectifs que
L'utilisateur s'est fixé soient atteints la tache des responsables en charge de ce projet, sera tout d'abord d'identifier quels processus sont concernés, les quels il va falloir simplifier, redéfinir, ou supprimer.

3.4.3 La réingénierie des logiciels :

La réingénierie des logiciels est l'ensemble des activités :

- Qui améliorent la compréhension que l'on a d'un logiciel
- Qui améliorent le logiciel lui-même, généralement dans le but

➔ D'accroître sa facilité de maintenance ou d'évolution,

➔ De faciliter la réutilisation de ses composants….

3.5 Les types de réingénierie :

Il existe plusieurs types de réingénierie de logiciel, Chaque type de réingénierie a ses propres avantages et inconvénients, et le choix de l'approche dépendra des objectifs spécifiques du projet et des contraintes techniques, dont voici quelques exemples :

A. La restructuration :

Est une technique de maintenance préventive qui implique des transformations structurelles relativement légères du code sans altérations comportementales du logiciel tel que vu par l'usager.

Un exemple de restructuration consiste à éliminer l'usage des énoncés "GOTO" en utilisant un style de programmation structurée.

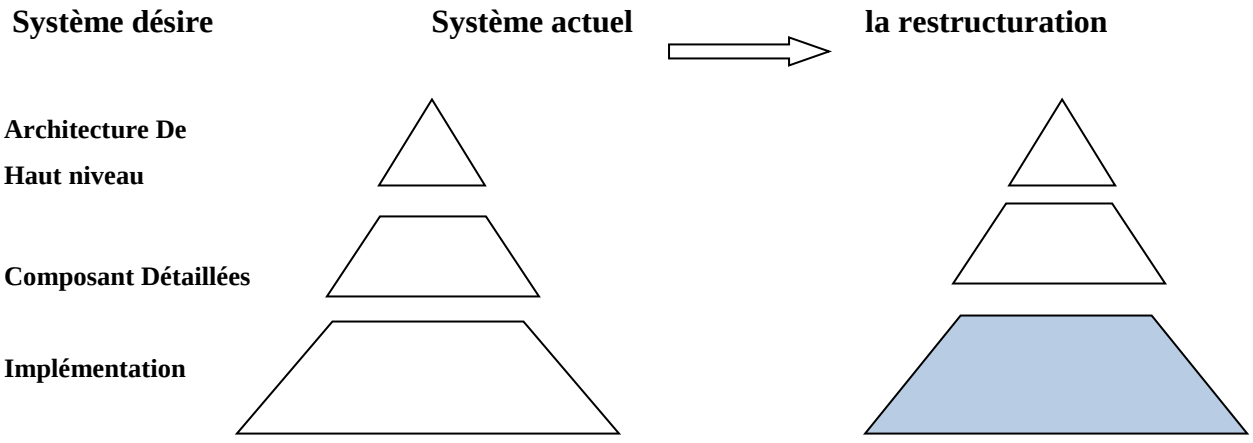


Figure 02 : réingénierie d'un système par restructuration

B. Le refactoring :

La refactoring ressemble restructuration a la différence près que les transformations qui sont effectuées sont généralement plus complexés et elles exigent une compréhension plus approfondie elle sémantique du code.

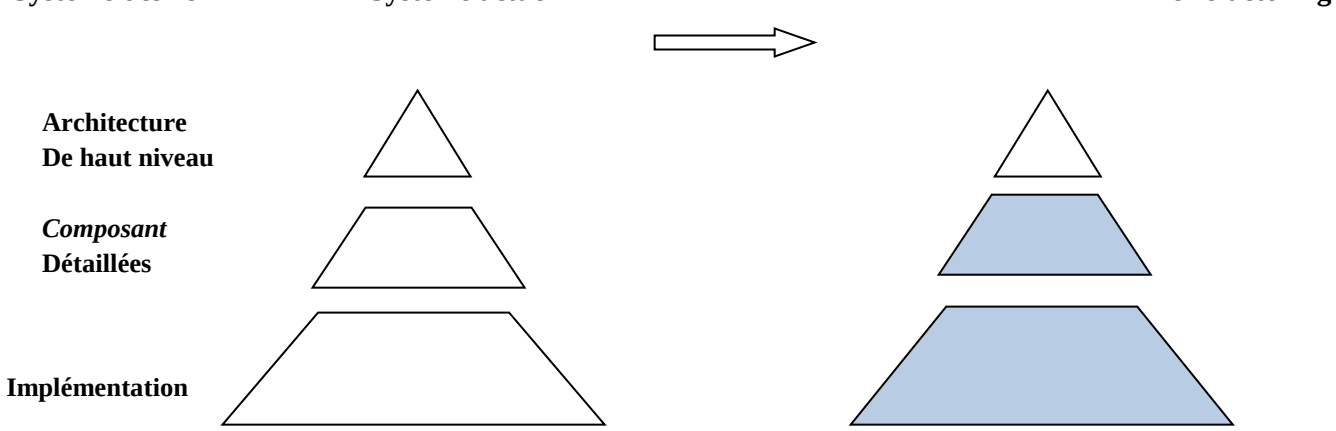


Figure 03 : réingénierie d'un système par refactoring

C. La migration :

Est activité de réingénierie caractérisée par l'évolution d'une propriété globale du logiciel. La propriété affectée peut être le langage de programmation, la plateforme (Migration macintosh a Windows), le style de programmation (migration impératif a orienté objet) la majorité des migrations impliquent de profondes transformations du logiciel la réingénierie comprend également une catégorie de transformations qui ressemblent ou a la migration mais elles études ou modifient la fonctionnement lité du logiciel.

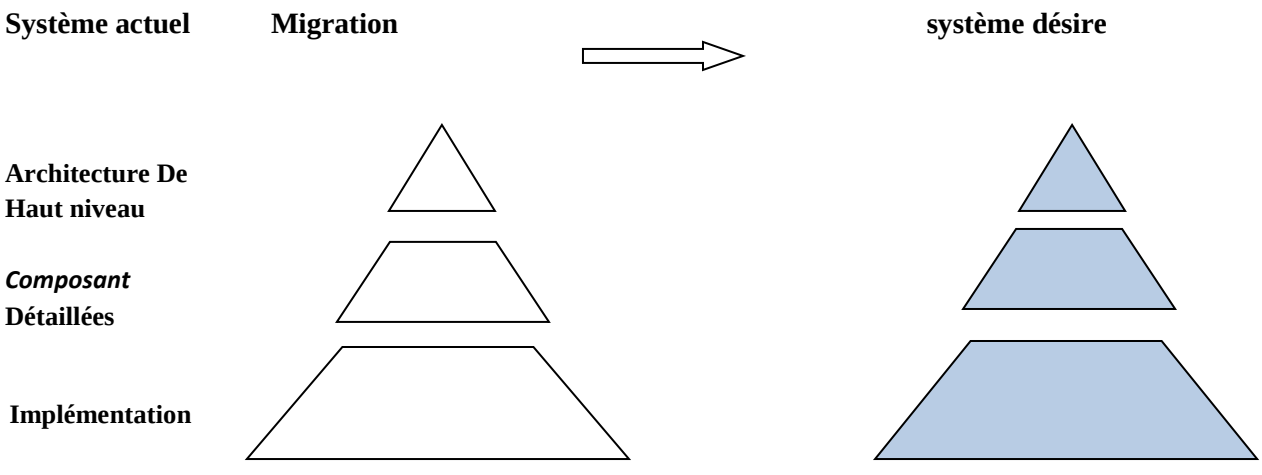


Figure 04 : réingénierie d'un système par migration

3.6 Les Etapes de réingénierie :

La réingénierie est un processus visant à repenser et à améliorer en profondeur les processus, les systèmes et les structures d'une organisation. Voici les étapes générales de la réingénierie :

A. La collecte de l'information :

C'est un rassemblement de toutes les informations possibles sur le programme. Les sources d'information peuvent comprendre le code source, les documents et les personnes connaissant le système.

B. L'étude de l'information :

Il s'agit de la revue des informations rassemblées. Cette étape permet aux intervenants de se familiariser avec le système et ses composantes. Un plan pour analyser le programme et enregistrer l'information récupérée, peut être formulé durant cette étape.

C. L'extraction de la structure :

Il s'agit de l'identification de la structure du programme et de la création d'un ensemble de diagrammes de la structure. Chaque nœud dans un diagramme de la
Structure correspond à une routine appelée dans le programme. Chaque arrête dans le diagramme représente les entrées/sorties des données dans les nœuds.

D. L'enregistrement des fonctionnalités :

Pour chaque nœud dans les diagrammes de structure, il faut enregistrer le traitement effectué dans le programme. La fonctionnalité peut être décrite en langage naturel ou en une notation plus formelle.

E. L'enregistrement du flux de données :

La structure du programme récupéré peut être analysée pour identifier les transformations des données dans le système. Ces étapes de transformation montrent le traitement des données réalisé dans le programme. Cette information est utilisée pour développer un ensemble de flux de données hiérarchique qui modélisent le système.

F. L'enregistrement du flux de contrôle :

Il s'agit de l'identification de la structure de contrôle de haut niveau qui affecte l'opération générale du système.

G. Passer en revue la conception récupérée :

C'est la vérification de l'exactitude des informations disponibles. Il faut détecter les items d'informations manqués et tenter de les récupérer, vérifier si l'information de conception récupérée représente correctement le programme.

H. La génération de la documentation :

C'est l'étape finale de la documentation de conception.

3.7 Les domaines de la réingénierie :

La réingénierie logicielle fait partie du domaine de la maintenance logicielle [5]. C'est un processus de transformation utilisé pour l'analyse des besoins (définition des spécifications des problèmes, des objectifs, des contraintes...)

Voici quelques domaines de réingénierie d'un logiciel :

3.7.1 Révision de l'architecture logicielle : revoir la structure globale du logiciel pour améliorer sa maintenabilité, sa performance et sa scalabilité.

3.7.2 Migration vers une nouvelle plateforme : adapter le logiciel pour qu'il fonctionne sur une nouvelle plateforme matérielle ou logicielle.

3.7.3 Réécriture du code source : restructurer le code source du logiciel pour le rendre plus lisible, maintenable et évolutif.

3.7.4 Amélioration des performances : optimiser les performances du logiciel en identifiant et en corrigeant les goulets d'étranglement.

3.7.5 Mise à jour des technologies obsolètes : remplacer les technologies obsolètes utilisées dans le logiciel par des technologies plus récentes et performantes.

3.7.6 Ajout de nouvelles fonctionnalités : Intégrer de nouvelles fonctionnalités au logiciel pour répondre aux besoins changeants des utilisateurs.

3.7.7 Correction des bugs et des failles de sécurité : identifier et corriger les bugs et les failles de sécurité présents dans le logiciel.

3.7.8 Adaptation aux normes et standards en vigueur: mettre à jour le logiciel pour qu'il respecte les normes et standards en vigueur dans l'industrie.

3.7.9 Documentation du code source : documenter le code source du logiciel pour faciliter sa compréhension et sa maintenance future.

3.7.10 Test et validation : effectuer des tests approfondis pour s'assurer que les modifications apportées au logiciel n'ont pas introduit de nouveaux problèmes ou erreurs.

3.8 Les avantages de la réingénierie :

Il est plus facile de modifier le code existant que de recommencer à nouveau. Ces options, parfois inéluctables, ont l'inconvénient d'être risquées, car de type **"big bang",** et coûteuses.

- On trouve dans le code existant des informations sur l'application qui pourraient ne pas exister ailleurs.
- La possibilité d'utiliser les outils CASE.
- On améliore le potentiel de réutilisation du code, des objets.
- On améliore souvent l'application en la rendant plus claire, plus fiable, plus maintenable et on augmente la qualité du programme.

- Des outils existent aujourd'hui, pour la rendre plus facile et automatique.
- Elle est très utile pour les outils stratégiques qui ont une longue durée de vie.
- Elle est indispensable pour la maintenance du système.

CONCLUSION :

Ce chapitre a abordé les concepts clés du génie logiciel et de la réingénierie des logiciels.

Nous avons d'abord examiné l'historique du génie logiciel, qui est devenu une discipline essentielle pour le développement de logiciels modernes. Nous avons ensuite défini le génie logiciel comme l'ensemble des méthodes, outils et processus utilisés pour concevoir, développer, tester et maintenir des logiciels de haute qualité.

Le cycle de vie d'un logiciel, comprenant les phases de conception, programmation, maintenance et réingénierie, a été exploré. La maintenance joue un rôle crucial dans ce cycle, impliquant des modifications et améliorations continues.

La réingénierie a été définie comme le processus de restructuration et de transformation d'un système logiciel existant tout en préservant sa fonctionnalité et ses performances. Les disciplines, types et étapes de la réingénierie ont été examinés en détail.

Enfin, les principaux domaines et avantages de la réingénierie, tels que l'amélioration de la qualité, des performances et de la maintenabilité, ont été identifiés.

Dans l'ensemble, ce chapitre a fourni une compréhension approfondie des concepts clés du génie logiciel et de la réingénierie, ouvrant la voie à une exploration plus approfondie de ces sujets.

Chapitre 02 :
Présentation de l'entreprise

INTRODUCTION :

Dans ce chapitre nous parlerons de l'entreprise dans laquelle nous avons fait un stage. Nous avons pu découvrir le fonctionnement et les activités de l'entreprise **CCLS M'SILA** lors de notre stage. Ce stage nous a donné l'occasion d'appliquer ce que nous avons appris pendant notre cursus universitaire et de développer nos connaissances dans le logiciel que nous allons étudier. Dans le cadre de ce mémoire, Nous aborderons divers aspects de l'entreprise **CCLS M'SILA** et compris son historique, son organisation interne, ses activités principales et les défis auxquels elle est confrontée. Nous allons également présenter le logiciel que nous avons étudié lors de notre stage et analyser son impact sur les opérations de l'entreprise. Nous pourrons mettre en avant les compétences acquises lors de notre stage et envisager les perspectives d'évolution dans ce domaine professionnel grâce à ce mémoire.

1. PRESENTATION DE L'ENTREPRISE :

1.1. OAIC :

L'Office algérien interprofessionnel des céréales est un organisme public dont la formulation administrative a été établie par décret présidentiel du 12 juin 1962 et qui est un instrument important de la politique céréalière de l'État algérien. Dans la mesure où elle joue un rôle essentiel dans la régulation et la modification du marché national d'une part et dans la réception et le stockage des céréales et légumes secs importés d'autre part, Il s'agit d'assurer un équilibre entre la production nationale et les besoins de la population en céréales et en légumes secs.

Dans ce cadre, l'Office s'est appuyé sur les moyens des Coopératives de Céréales et de Légumes Secs et de leurs Unions (CCLS et UCA), (infrastructure de stockage, transport et manutention).

1.2. LA CCLS M'SILA :

La coopérative des céréales et légumes secs de la Wilaya de M'SILA (C.C.L.S) est un organisme créé en 1975 et devenu opérationnel en 1987 dans le cadre de la création des CCLS des zones sahariennes et présahariennes.

La CCLS de M'sila à pour mission le stockage et la distribution des céréales ainsi que les orges pour la consommation du cheptel.

La CCLS de M'sila est sous tutelle d'autre organisme public national appelé: L'Office Algérien interprofessionnel des céréales (O.A.I.C), Ce dernier est régulateur du marché des céréales dans tout le territoire de l'Algérie.

Instrument de la politique céréalière de l'état, l'OAIC participe à l'importation, la réception, le stockage et le transport des graines de céréales et de légume secs, à la consultation et à la régulation des stocks, à la gestion des mécanismes de stabilisation et d'information des prix et à la définition des règles de commercialisation et de circulation des grains en Algérie.

1.3. Les fonctions de CCLS :

La coopérative de céréales et légumes secs est une organisation qui regroupe des producteurs de céréales et de légumes secs afin de mutualiser leurs ressources et leurs efforts pour mieux commercialiser leurs produits. Les principales fonctions de cette coopérative sont les suivantes :

- **Achat groupé :**
  La coopérative permet aux producteurs d'acheter en gros les intrants nécessaires à la production, tels que les semences, les engrais et les pesticides, à des prix avantageux grâce à la force du groupe.

- **Commercialisation** :
  La coopérative se charge de la vente des produits des membres sur le marché, en

  Négociant des contrats avec les acheteurs et en assurant une meilleure visibilité et

  Promotion des produits.

- **Stockage et logistique :**
  La coopérative met en place des infrastructures de stockage pour conserver les récoltes dans des conditions optimales, ainsi qu'un réseau logistique pour assurer la livraison des produits aux clients dans les délais.

- **Formation et conseil :**
  La coopérative propose des formations techniques aux membres pour améliorer leurs pratiques agricoles, ainsi que des conseils en matière de gestion d'exploitation et de développement durable.

- **Recherche et développement :**
  La coopérative investit dans la recherche pour améliorer les variétés de céréales et légumes secs cultivées par ses membres, ainsi que dans le développement de nouvelles techniques agricoles plus durables. En résumé, la coopérative de céréales et légumes secs joue un rôle essentiel dans l'accompagnement et le soutien des producteurs agricoles pour garantir une production de qualité, respectueuse de l'environnement, tout en assurant une meilleure rentabilité économique pour ses membres.

1.4. Structure de CCLS :

Différentes structures et services, liées les uns les autres, assurent harmonieusement la continuité de la **CCLS** et son bon fonctionnement :

-Administration : chargée des opérations administratives, secrétariat et gestion du personnel.

-Informatique : chargée de l'informatisation des services.

-Finance : chargée des opérations financières et de la comptabilité.

- ***Matière :*** comptabilité matière et gestion de stocks de produits commercialités.

-***Commerce :*** commercialisation des produit, suivi des clients et fournisses, facturation, …etc.

-***Exploitation :*** chargée des moyens de stockage, suivi et entretien.

-***Transport :*** gestion et maintenance du parc routier (camions), achat et gestion de pièces de rechanges (atelier).

- ***Moyens généraux :*** suivi des moyens généraux de la CCLS, notamment, fournitures de bureaux et travaux de mécanographie.

-***Appui à la production :*** gestion du parc agricole et suivi des producteurs de céréales.

1.5. Les tache de service commerciale :

Vente et facturation les céréales destinées à la consommation aux transformateurs.

Vente et facturation des légumes secs à différents clients.

Vente et facturation les semences et les engrais aux agriculteurs.

Vente et facturation des aliments (orge, avoine...) aux éleveurs.

Facturation de toutes les prestations fournies aux différents clients (location de matériel, Main d'œuvre, ...).

Acheter des céréales et des légumes secs aux agriculteurs (pendant la saison de récolte) et livrer un document de Décompte.

Délivrer les documents (quotidiens, décadaires, mensuels, …) et les envoyer à Office Professionnel des Céréales, à la Direction du Commerce et à la Direction des Services Agricoles.

Et bien d'autres tâches secondaires.

- Service Technique

1.6. Généralités sur le domaine d'études :

- **Concepts sur le sujet :**

Stockage : est la conservation des matières premières et des produits de divers types Lieux de stockage :

Ce sont les lieux où sont placés les matériaux et produits stockés et appartiennent à la Coopérative céréalière.

Légumes secs 19 centres de stockage, silos et entrepôts d'une superficie totale de 520 000 OX répartis sur plusieurs régions du territoire de l'Etat (Sidi Issa, Sidi Hegras , Al-Dis , Ain el hjal , Sidi Amer, M'sila) Approvisionnement :

Approvisionner l'inventaire en matériaux ou en produits Clients.

Il s'agit des types de clients avec lesquels l'entreprise traite, notamment :

Les unions coopératives, les coopératives.

Paysans, moulins, cimenteries Matériaux utilisés pour le stockage et la distribution.

Céréales : blé dur - blé tendre - orge - orge.

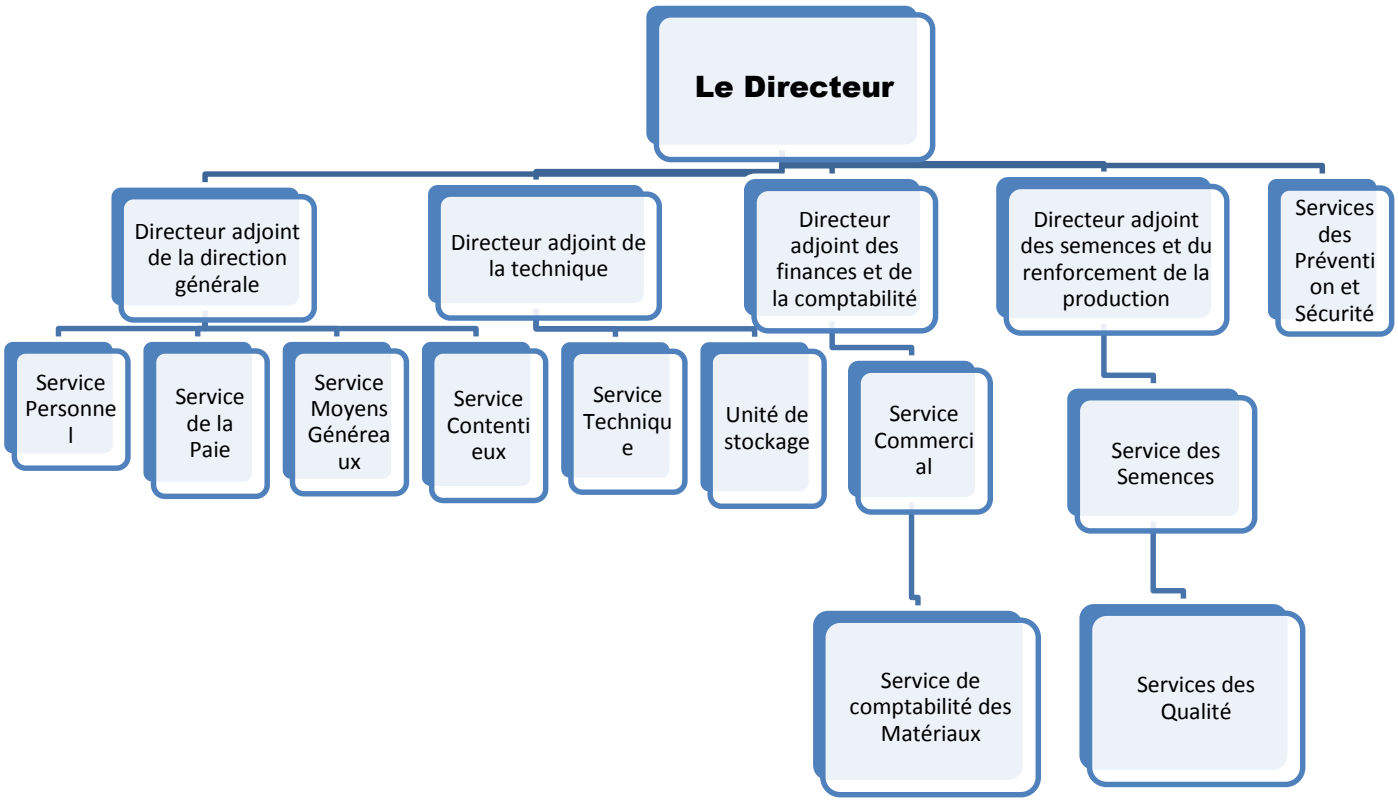
Légumes secs : pois chiches – lentilles – haricots. Engrais

Toutes sortes d'engrais.

Picots de graines pour céréales et fourrages de différents types. Origine du matériau …

Il est divisé en deux types : importé. Locale

1.7. LOGIGRAMME DE L'ENTREPRISE :



CONCLUSION :

Ce chapitre offre une compréhension approfondie de l'entreprise CCLS M sila, de son histoire, de son organisation interne et de ses activités principales, Cela permettra de mettre en avant les compétences acquises lors du stage et d'envisager les perspectives d'évolution dans ce domaine professionnel.

Chapitre 03 :
L’analyse de l’ancien système

INTRODUCTION :

Ce chapitre se concentre sur l'ancien logiciel "Décompte" utilisé par l'entreprise CCLS M'Sila avant sa réingénierie. Nous allons examiner en détail les principales fonctions de ce système, ainsi que ses différentes interfaces, afin de mieux comprendre ses capacités et ses limites. Cette analyse permettra de mettre en évidence les aspects du logiciel qui nécessitaient une amélioration, justifiant ainsi le processus de réingénierie entrepris par l'entreprise.

1. ETUDE ET ANALYSE DU LOGICIEL :

1.1. Fonction principale :

La fonction principale de ce logiciel est de gérer toutes les opérations bancaires et les opérations de caisse de l'entreprise, l'information est extraite sous forme de **« décompte »**

- Le logiciel exécute le travail automatiquement en introduisent les données.
- Facile à utiliser (pas seulement le concepteur ou l'informaticien).

1.2. Décompte :

Le décompte, un document délivré par la Coopérative de Céréales et Légumes Secs au profit de l'agriculteur, Il s'agit de la déduction du montant qui une dette envers l'agriculteur (des produits pris à crédit, comme les semences, les engrais, autres prestations…) sur le montant total de ses récoltes versées (vendues) à la coopérative pendant la saison de moisson et battage. Ce document, il contient également le détail de ces dettes et ventes.

1 .3. Captures d'écran sur le système :

Présentation du logiciel : Au lancement du logiciel, une fenêtre principale apparait dans l'écran, en général elle ressemble à celle qui est présentée par les figures suivantes {1,2} :

Figure 05 :la fenêtre principale

Figure06: l'authentification

Nous devons entrer un mot de passe pour accéder au logiciel que nous allons étudier. Après cela, il nous demande de choisir la cession correspondante et un menu principal apparaît, comme le montre la figure suivante.

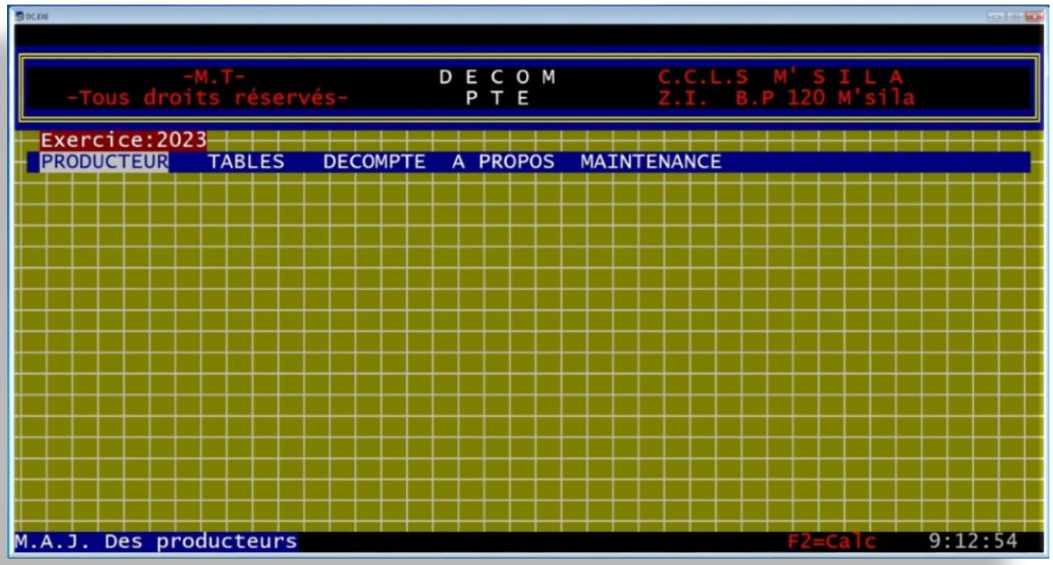


Figure07 : menu principal

Les commandes suivantes sont incluses dans ce menu :

- Producteur.
- Tables.
- Décompte.
- A propos.
- Maintenance.

I. Menu producteur :

La personne physique ou morale qui possède une certaine relation avec la société est appelée producteur. Il peut s'agir d'un client, d'un fournisseur, d'un employé ou d'une entreprise. Nous pouvons accéder à des commandes telles que :

- Création.
- Modification.
- Suppression.
- Consultation.
- Éditions.

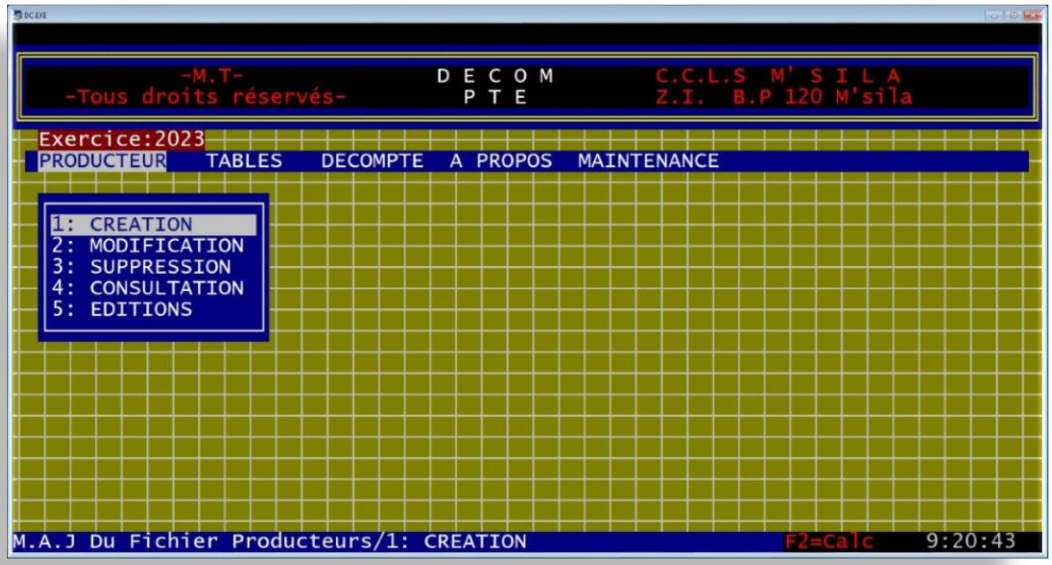


Figure 08 : menu de producteur

❖ **CREATION :**

Cette instruction permet d'enregistrer les informations nécessaires d'un opérateur, comme illustré dans la figure suivante.

Figure 09 :formulaire de création

❖ **MODIFICATION :**

Cette commande nous permet de modifier les informations enregistrées par un opérateur.

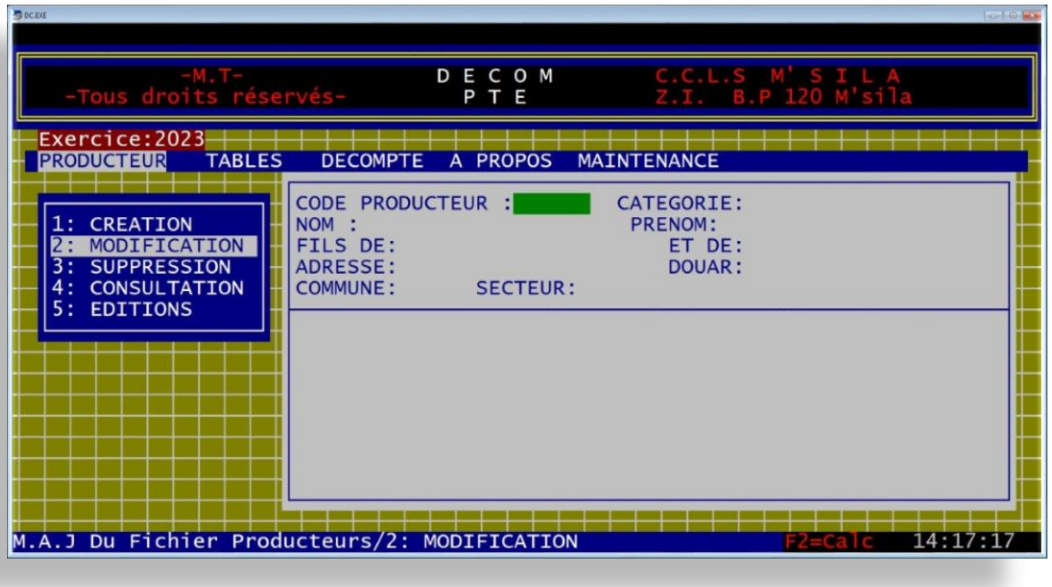


Figure 10 :la commande de modifier

❖ **SUPPRESSION :**

Le menu "Suppression" offre la possibilité de retirer un producteur.

Figure 11: formulaire de supprimer

❖ **CONSULTATION :**

Avec cette commande, nous pouvons consulter divers opérateurs. Il existe deux façons de les consulter :

- Par numéro de matricule
- Selon l'ordre alphabétique.

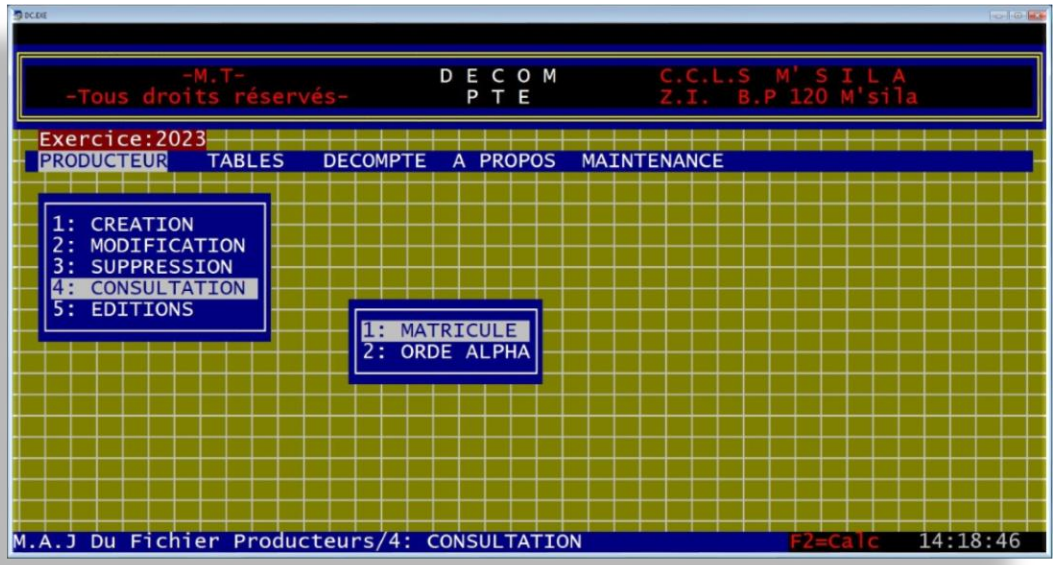


Figure 12:la commande consulter

✓ **CONSULTATION PAR MATRICULE :**

Pour accéder à un opérateur spécifique, il suffit de saisir son code.

Figure 13:la commande consulter par matricule

✓ **CONSULTATION PAR ORDRE ALPHABETIQUE :**

Pour accéder à plusieurs opérateurs. En choisissant cette option, une fenêtre apparaît qui nous demande d'indiquer l'intervalle alphabétique d'affectation des opérateurs.

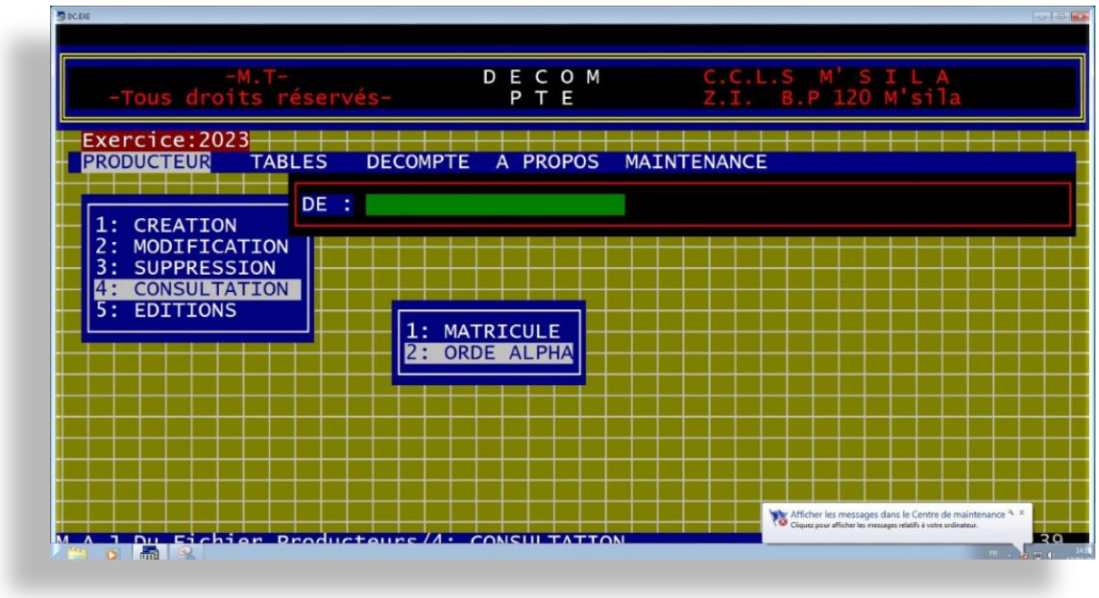


Figure 14 : la commande consulter par alpha

- **EDITION :**

C'est une instruction qui permet de nous donner soit une impression des opérateurs soit l'affichage sur l'écran. En tapant sur cette commande, une fenêtre apparaît comme le montre la figure suivante.

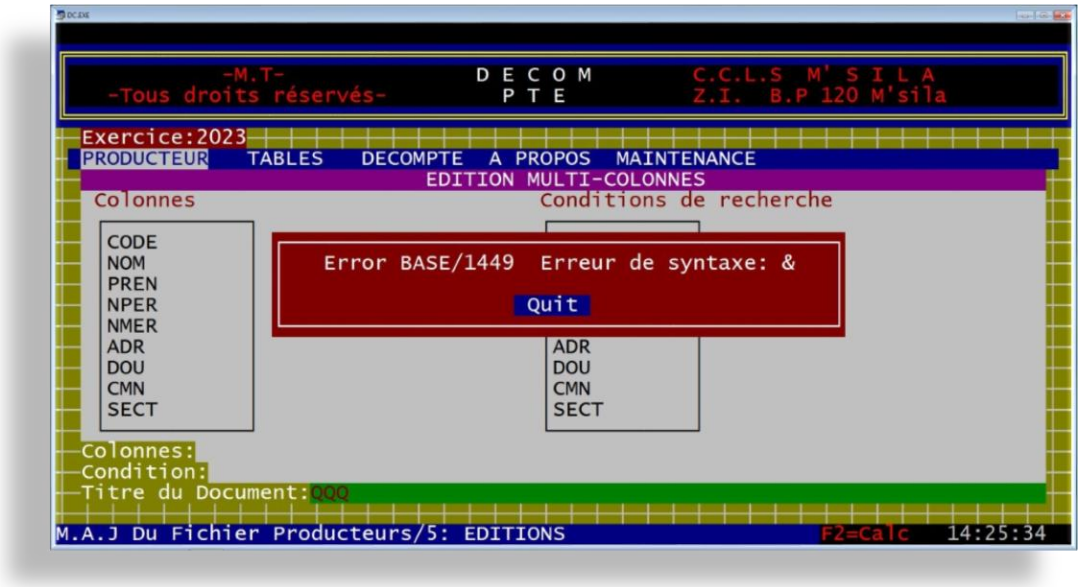


Figure 15:la commande éditions

Par conséquent, nous avons la possibilité de produire une impression en utilisant :

- Liste par commune.
- Liste par catégorie.
- Liste par catégorie et commune.

- Édition libre.

✓ **LA LISTE PAR COMMUNE :**

En cliquant sur cette commande, une liste de communes déjà enregistrées s'affiche. En sélectionnant l'une d'entre elles, le programme nous présente une représentation de tous les employés de la commune.

✓ **LISTE PAR CATEGORIE :**

En tapant sur cette commande, une table apparaît qui contient des catégories déjà enregistrées, et le logiciel nous donne une impression de tous les opérateurs de cette catégorie en choisissant l'une d'entre elles.

✓ **LISTE PAR COMMUNE ET CATEGORIE** :

La même chose, mais cette fois-ci, le logiciel offre une modification de tous les opérateurs en fonction de la commune et de la catégorie.

II. Menu tables :

Le menu Tables nous permet d'accéder à quêter commandes produits, communes, catégorie et magasins.

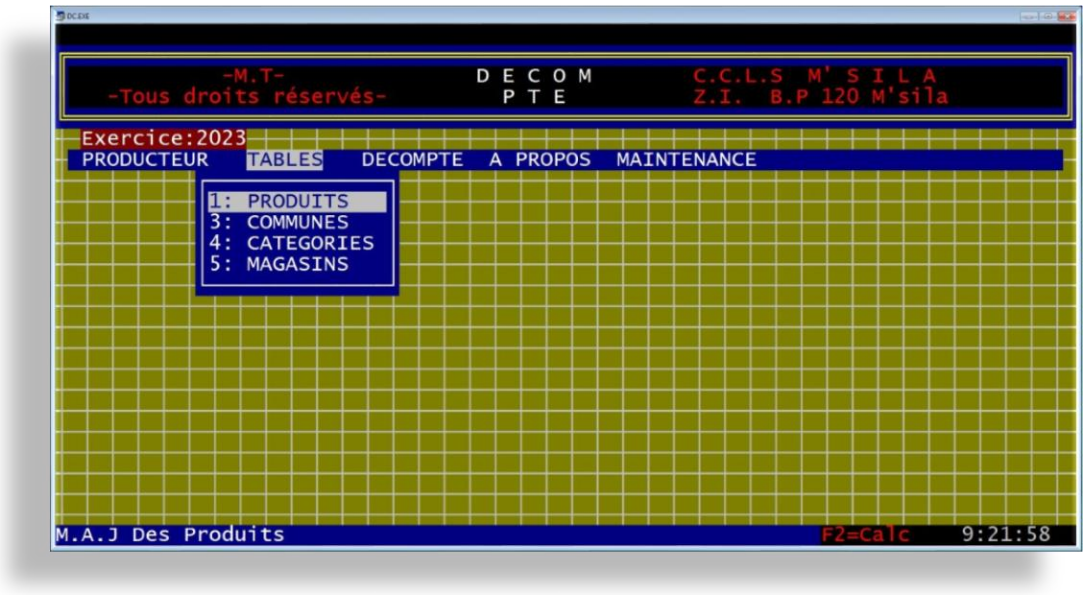


Figure 16 : menu de table

❖ **Produits** :

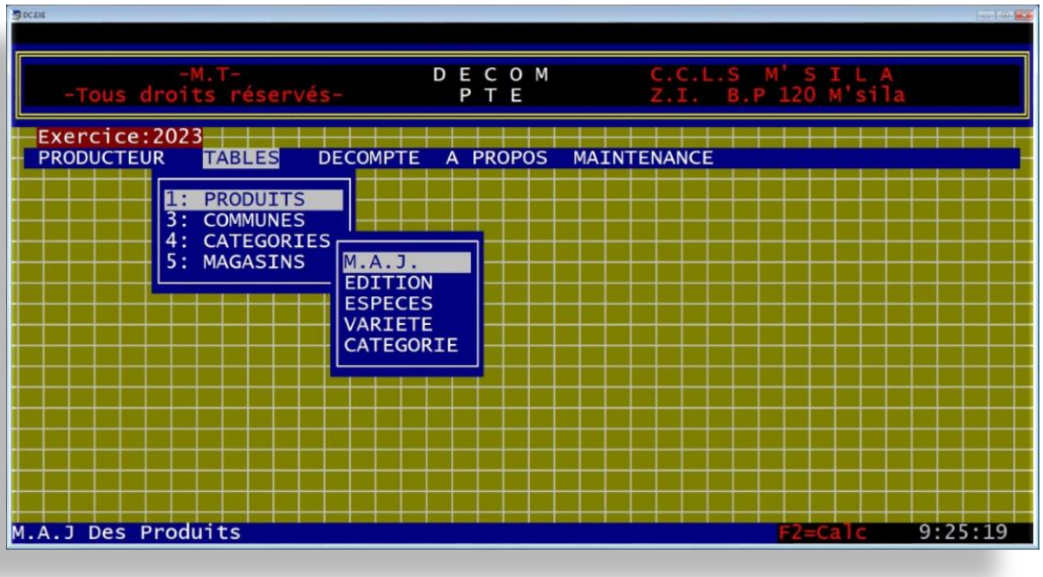


Figure 17 :la commande de produit

❖ **Commune** : En tapant sur cette commande, une fenêtre s'ouvre et indique à quelle région appartient l'opérateur. Elle contient :

- maj.
- Édition.

✓ M.A.J : Cette instruction permet de modifier cette table.

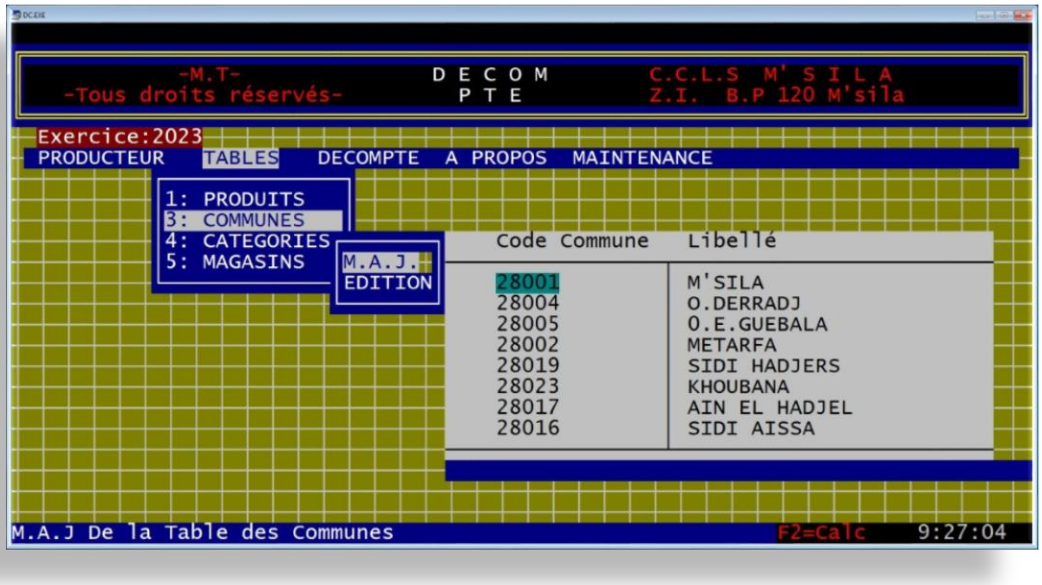


Figure 18 :la commande de communes

- Edition : Cette table est imprimée à l'aide de l'édition.

- Catégorie :

Elle nous indique à quelle catégorie appartient l'opérateur, en tapant sur cette instruction une fenêtre s'ouvre elle contient :

- ✓ M.A.J
- ✓ Édition.

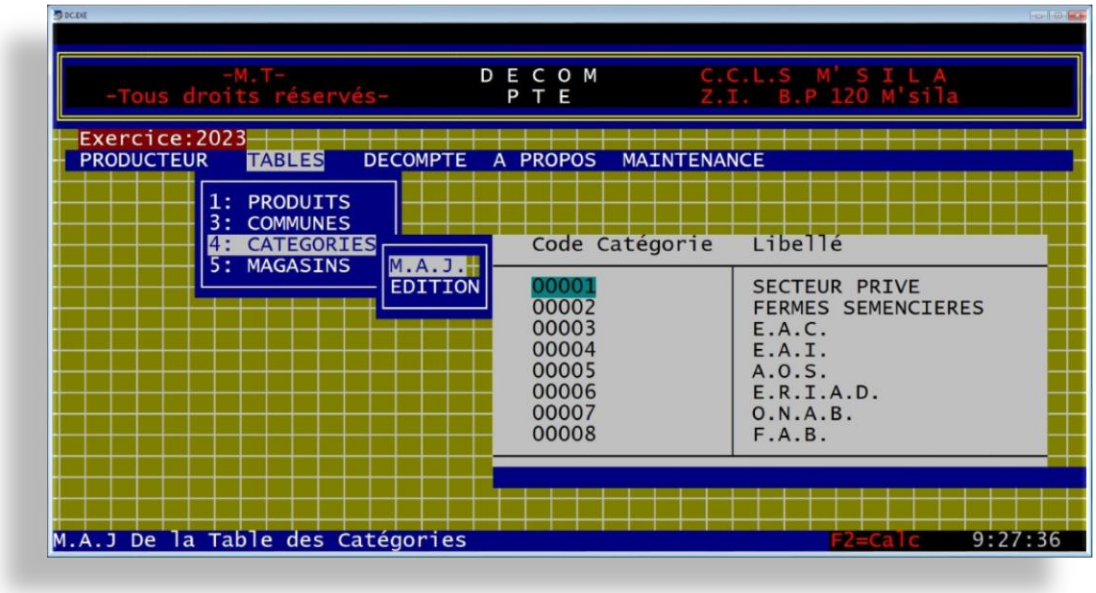


Figure 19:la commande de catégories

- ✓ **M.A.J :**

Cette instruction permet de faire la mise à jour de cette tabl.

- ✓ **Edition :**

Cette table est imprimée à l'aide de cette méthode d'édition.

- **Magasins.**

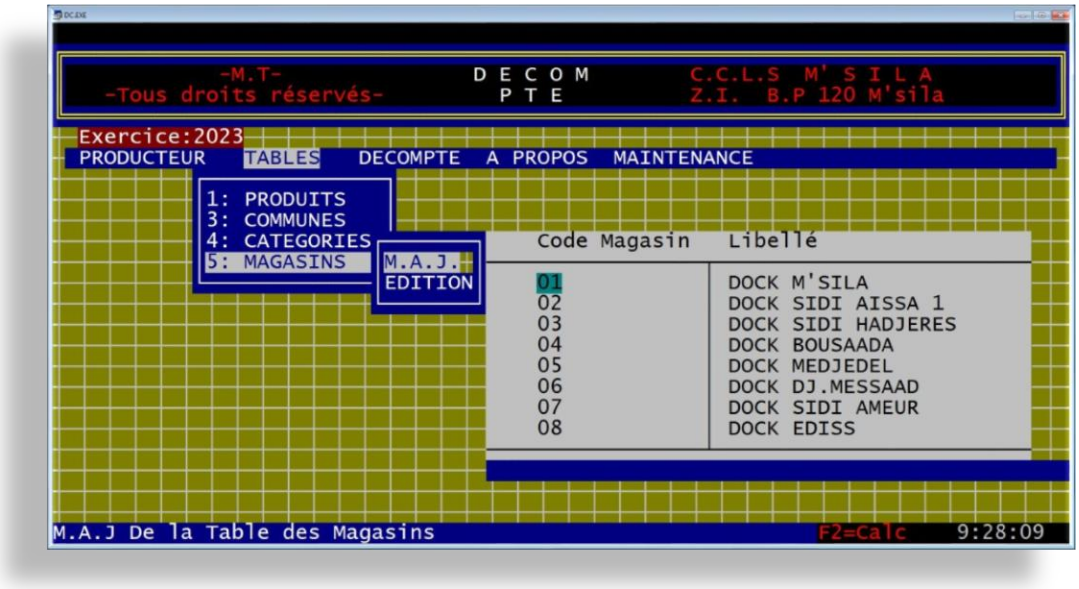


Figure20 :la commande de magasins

III. Menu de décompte

Figure 21 : menu de décompte

❖ **MODIFICATION**

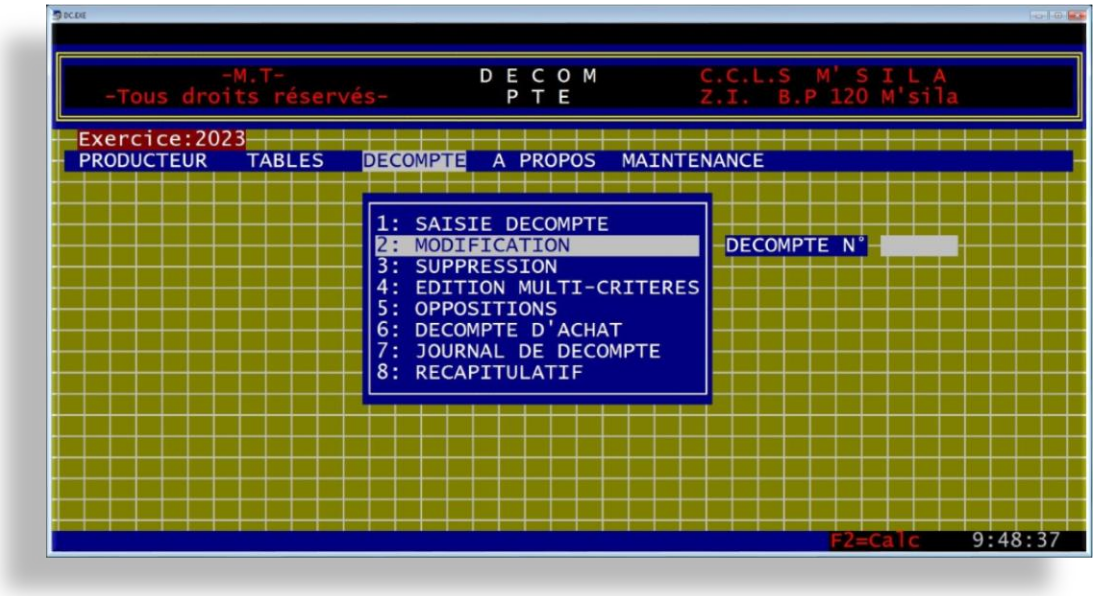


Figure 22:la commande de modifier

❖ **Suppression** :

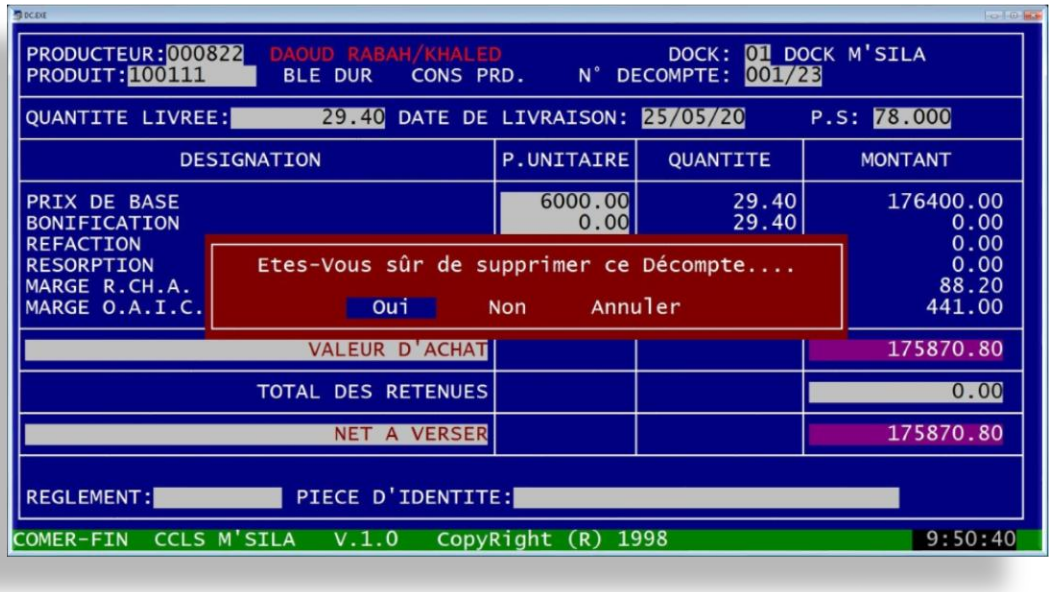


Figure 23: la commande de supprimer

❖ **Edition multicritères**.

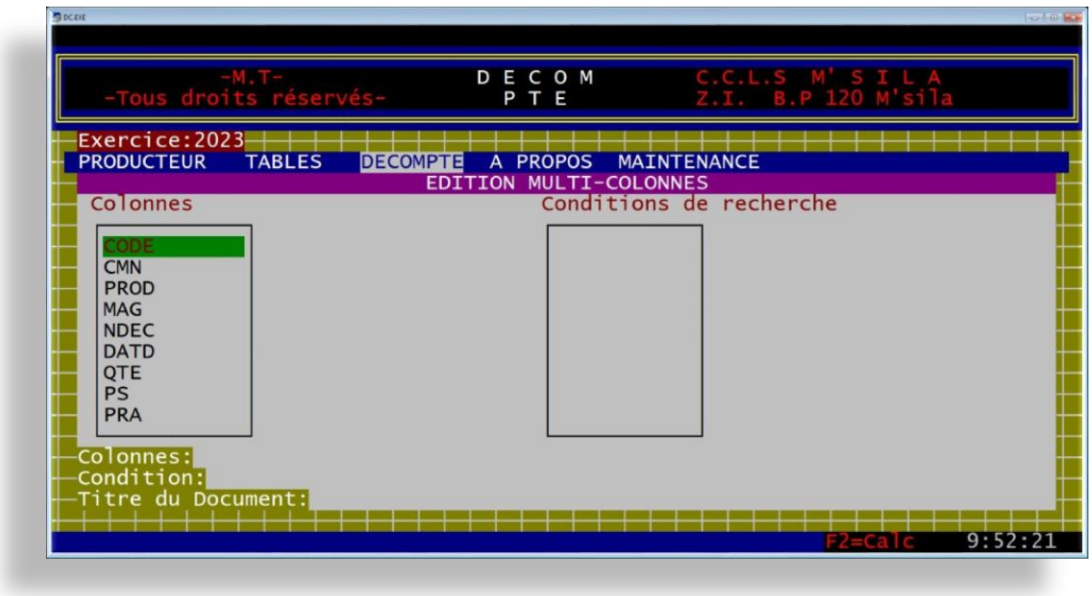


Figure 24 :la commande de Edition multi

❖ **Journal de décompte** :

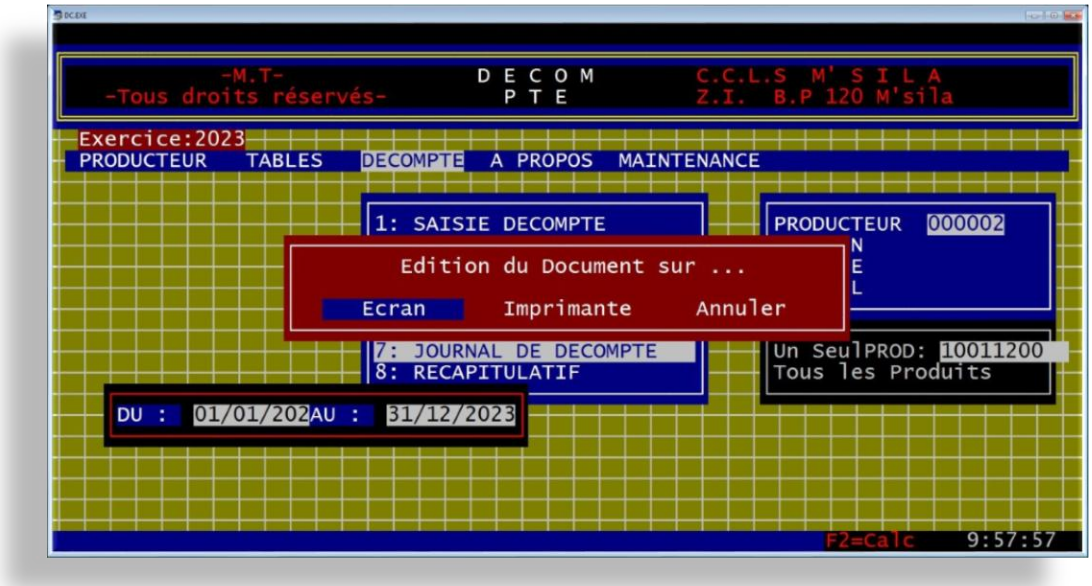


Figure 25: la commande journal de décompte

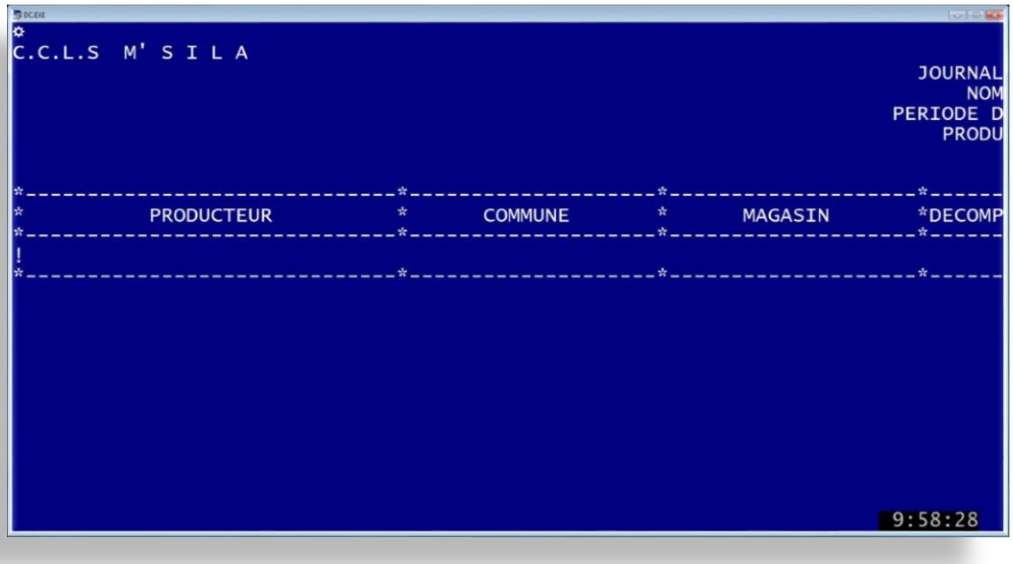


Figure 26:résultats journal de décompte

- **Récapitulatif.**

Figure 27 :la commande récapitulatif

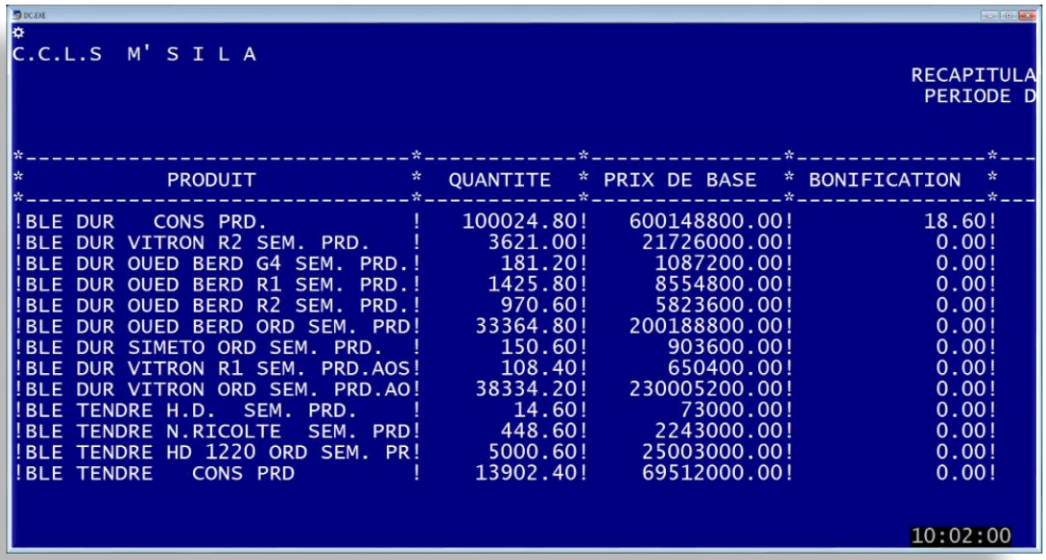


Figure 28:résultats de récapitulatif

- **décompte d'achat :**

Figure 29: la commande décompte d'achat

Figure 30 :résultats de décompte d'achat

IV. A propos :

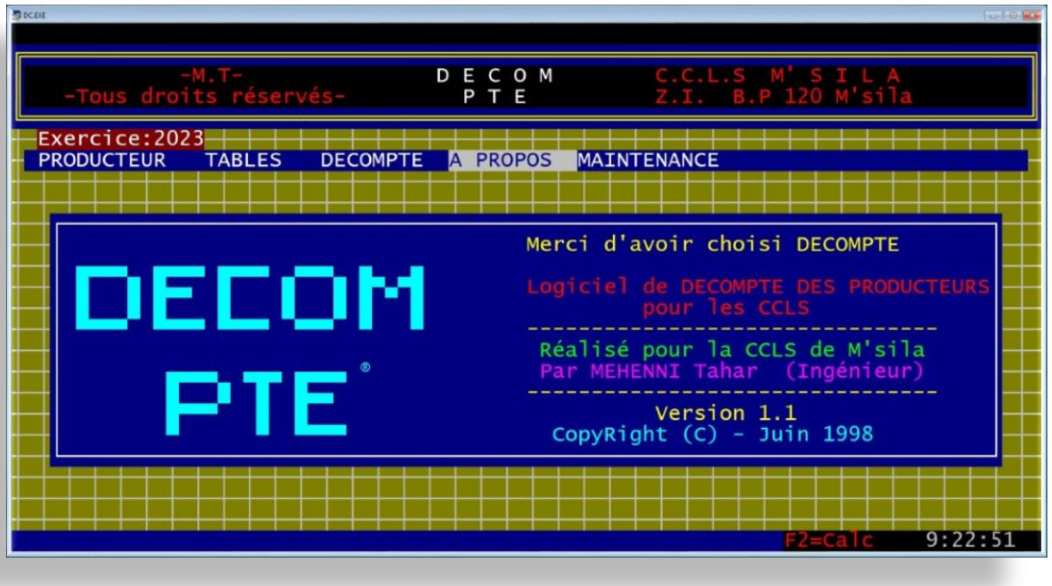


Figure 31 :A-propos

V. Menu de Maintenance :

Elle est valide en indiquant la date de début jusqu'à la date de fin quelques opérations sont disponibles dans ce menu afin de maintenir le logiciel, comme illustré dans la figure ci-dessous.

Figure 32: menu de maintenance

- **Réindexer :**

Cette commande nous permet de conserver les mises à jour effectuées sur l'un des éléments suivants (producteur, tables, décompte, A propos, Maintenance).

- **Sauvegarder :**

Cette procédure permet de stocker des fichiers sur une disquette.

- **Mettre à Zéro :**

Cette commande permet d'initialiser les tables à zéro selon notre préférence, comme illustré dans la figure ci-dessous.

- **Mot de passe :**

Le mot de passe peut être modifié grâce à cette commande.

1.4.LA FACTURE DE DECOMPTE FINAL :

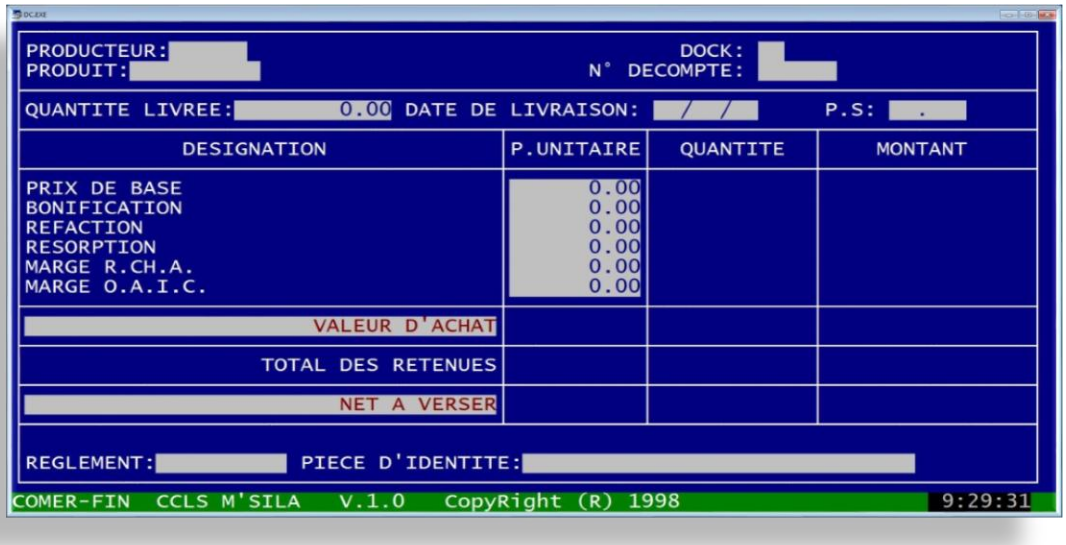


Figure33:la facture de décompte

CONCLUSION :

L'analyse approfondie de l'ancien logiciel "Décompte" a mis en lumière ses capacités et ses limites. Bien que remplissant des fonctions importantes pour l'entreprise, le logiciel présentait des lacunes et des défis qui ont motivé la décision de procéder à sa réingénierie. La compréhension détaillée de l'ancien système servira de base pour examiner les améliorations apportées par le nouveau logiciel développé à la suite de ce processus de réingénierie, qui sera abordé dans les chapitres suivants.

Chapitre 04 :
Conception de nouveau logiciel

INTRODUCTION :

La modélisation avec le langage UML est un outil essentiel dans le domaine de l'informatique et du développement logiciel. Dans ce chapitre nous allons explorer en détail les principes fondamentaux de la modélisation avec UML, en mettant l'accent sur ses différents diagrammes et leur utilisation dans le processus de développement logiciel. Enfin, nous examinerons quelques études de cas concrets pour illustrer l'application pratique de la modélisation avec UML dans des projets réels, en mettant en lumière les avantages qu'elle peut apporter en termes de compréhension, communication et qualité du code produit.

1. Unified Modeling Language

1.1 L'HISTOIRE D'UML :

UML ou Unified Modeling Language, est un langage de modélisation graphique utilisé dans le domaine du génie logiciel pour représenter visuellement le design et l'architecture des systèmes logiciels. Il a été créé dans les années 1990 par Grady Booch, Ivar Jacobson et James Rumbaugh, trois experts en génie logiciel qui ont uni leurs forces pour développer un langage de modélisation commun. L'objectif principal de l'UML était de fournir un langage standardisé et universellement compris pour la modélisation des systèmes logiciels. En utilisant des diagrammes et des notations graphiques, les développeurs peuvent communiquer plus efficacement sur la conception d'un système, en facilitant la compréhension et la collaboration entre les membres de l'équipe. Depuis sa création, l'UML est devenu un outil essentiel dans le processus de développement logiciel, largement utilisé par les professionnels du secteur pour concevoir des systèmes complexes. Il a également évolué au fil du temps pour s'adapter aux nouvelles technologies et aux besoins changeants du secteur. Aujourd'hui, l'UML est largement reconnu comme un standard de facto dans le domaine du génie logiciel et continue d'être utilisé par des milliers de développeurs à travers le monde pour concevoir et modéliser des systèmes logiciels.

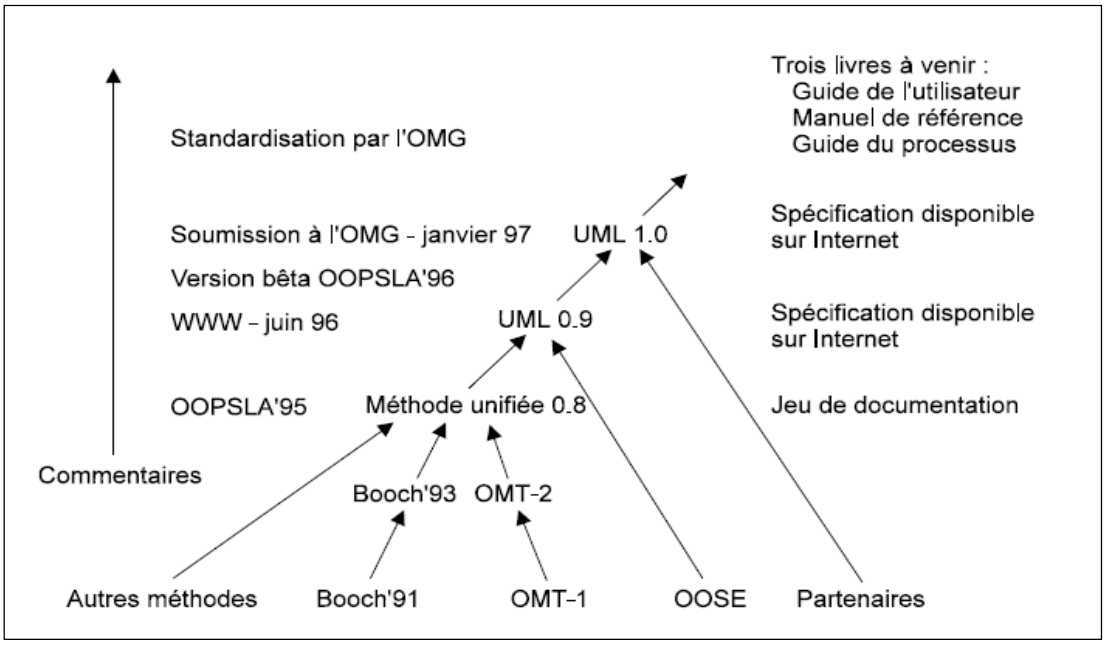


Figure 34 : La naissance d'UML

1.2 Définition UML :

L'UML (Unified Modeling Language ou Langage de modélisation unifiée en français) est un langage graphique de modélisation informatique. Ce langage est désormais la référence en modélisation objet, ou programmation orientée objet. Cette dernière consiste à modéliser des éléments du monde réel (immeuble, ingrédients, personne, logos, organes du corps...) ou virtuel (temps, prix, compétence...) en un ensemble d'entités informatiques appelées « objet ».

L'UML est constitué de diagrammes qui servent à visualiser et décrire la structure et le comportement des objets qui se trouvent dans un système. Il permet de présenter des systèmes logiciels complexes de manière plus simple et compréhensible qu'avec du code informatique. L'UML a des applications dans le développement logiciel, mais aussi dans l'industrie (pour modéliser les flux de processus par exemple), dans l'ingénierie ou le marketing. L'UML 1.0 a été adopté comme standard par l'Object Management Group (OMG) en janvier 1997. Il est issu de la fusion de trois méthodes orientées objet issues des travaux de Grady Booch, de Jim Rumbaugh et d'Ivar Jacobson. Des versions successives ont ensuite été validées, la dernière en date étant l'UML 2.5.1.

1.3 Les logiciels de modélisation UML :

Il existe de nombreux logiciels de modélisation UML, certains gratuits (GitMind, Gliffy, Draw.io, Moqups...), d'autres payants (Lucidchart, MagicDraw, StarUML, EdrawMax...) ou d'autres intégrés à des plateformes (IBM Rational Rhapsody, Microsoft Visio...). Certains permettent d'exporter les diagrammes UML dans différents formats ou d'importer du code pour l'afficher sous forme de diagramme. Attention, certains logiciels introduisent parfois des notations non conformes et donc non compatibles avec d'autres modélisations UML.

2. Les Diagramme UML

2.1 DEFINITION DIAGRAMME UML

Un diagramme UML est un moyen de visualiser des systèmes et des logiciels à l'aide du langage de modélisation unifié (UML). Les ingénieurs logiciels créent des diagrammes UML pour comprendre la conception, l'architecture du code et la mise en œuvre proposée de systèmes logiciels complexes. Les diagrammes UML sont également utilisés pour modéliser des flux de travail et des processus d'entreprise. Les diagrammes UML permettent de garder une trace des relations et des hiérarchies entre des lignes de code importantes. Bien que ces diagrammes ressemblent à certains arbres de décision ou diagrammes de flux, ils possèdent des attributs uniques. Les diagrammes UML divisent également les composants et les sous-composants qui sont essentiels à la construction d'un logiciel.

2.2 Caractéristiques des diagrammes UML :

- Les diagrammes UML supportent l'abstraction. Leur niveau de détail caractérise le niveau d'abstraction du modèle.
- La structure des diagrammes UML et la notation graphique des éléments de modélisation est normalisée (document "UML notation guide").
- la sémantique des éléments de modélisation et de leur utilisation est définie par le méta modèle UML (document "UML sémantiques") **[J.Steffe, 05].**

2.3 Avantages d'utilisation des diagrammes UML :

- Ils transforment du code complexe en un diagramme visuel.
- Ils permettent aux développeurs d'avoir une vue d'ensemble d'un système.
- Ils aident les non-programmeurs à comprendre les processus et les fonctionnalités des logiciels.

2.4 Les différents types de diagrammes UML :

L'UML définit 14 types de diagrammes divisés en deux catégories :

1. Les diagrammes de structure représentent les éléments du système, leurs propriétés et leurs relations entre eux
   - Diagramme de classes.
   - Diagramme d'objets.
   - Diagramme de composants.
   - Diagramme de structure composite.
   - Diagramme d'ensemble.
   - Diagramme de déploiement.
   - Diagramme de profil.
2. Les diagrammes de comportement représentent les processus et les interactions entre les objets :
   - Diagramme de cas d'utilisation.
   - Diagramme d'activité.
   - Diagramme d'état-transition.
   - Diagramme de séquence.
   - Diagramme de communication.
   - Diagramme de temps.
   - Diagramme d'aperçu d'interaction

2.5 Vues statiques du système :

La conceptualisation vise à comprendre et à structurer les besoins du client. Il ne s'agit pas de rechercher l'exhaustivité, mais d'expliciter, trier et structurer les besoins. Une fois identifiés et structurés, ces besoins :

- Définissent le contour du système à modéliser (ils précisent le but à atteindre),
- Permettent d'identifier les fonctionnalités principales (critiques) du système.
- Le modèle conceptuel doit permettre une meilleure compréhension du système.
- Le modèle conceptuel doit servir d'interface entre tous les acteurs du projet.
- Les besoins des clients sont des éléments de traçabilité dans un processus intégrant UML.
- Le modèle conceptuel joue un rôle central, il est capital de bien le définir …. **[J. Steffe , 05].**

2.5.1 Diagramme de classe : Étant donné que de nombreux logiciels reposent sur la programmation orientée objet, dans laquelle les développeurs définissent différentes fonctions pouvant être utilisées, les diagrammes de classes sont le type de diagramme UML le plus couramment employé. Ils exposent la structure statique d'un système, notamment les classes, leurs attributs et leurs comportements, ainsi que les liens entre chacune d'elles. Une classe est représentée par un rectangle contenant trois compartiments empilés verticalement. Le compartiment supérieur contient le nom de la classe et est indispensable, tandis que les deux compartiments inférieurs fournissent des détails sur les attributs et les opérations ou comportements de la classe.

2.5.2 Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation modélisent la manière dont les utilisateurs, représentés sous forme de figurines appelées « acteurs », interagissent avec le système. Ce type de diagramme UML est une vue d'ensemble des relations entre les acteurs et les systèmes, ce qui en fait un excellent outil pour présenter votre système à un public non technique.

2.5.3 Diagramme de séquence :

Un diagramme de séquence, parfois appelé diagramme d'événements ou scénario d'événements, montre l'ordre dans lequel les objets interagissent. Ils vous permettent ainsi de représenter visuellement des scénarios d'exécution simples.

3. CONCEPTION DU NOUVEAU SYSTEME :

– **Diagramme de classe** :

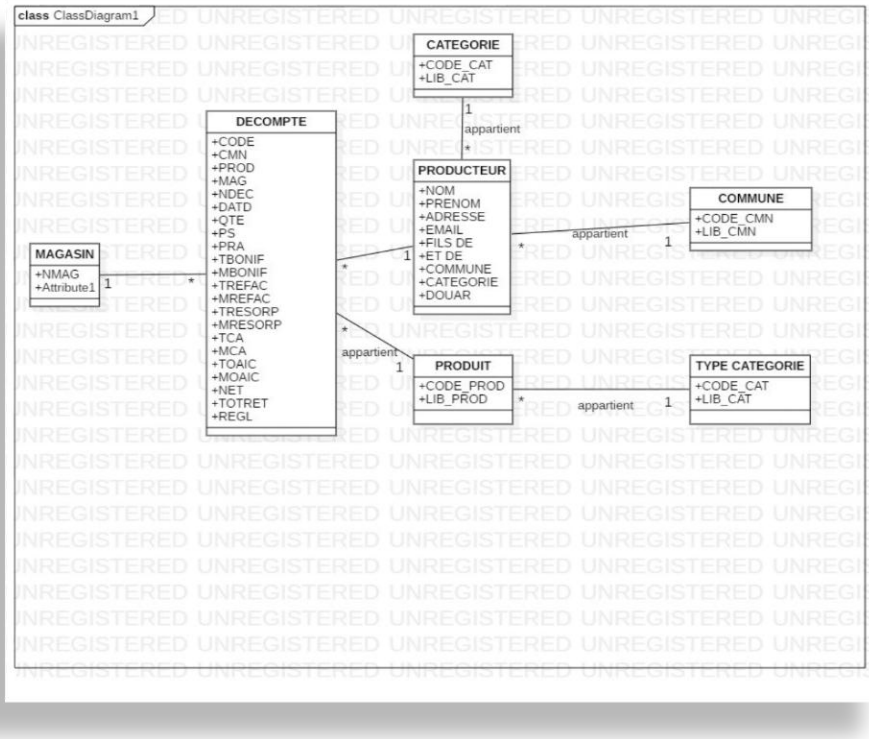


Figure 35 : diagramme de classe

– **Diagramme de cas d'utilisation :**

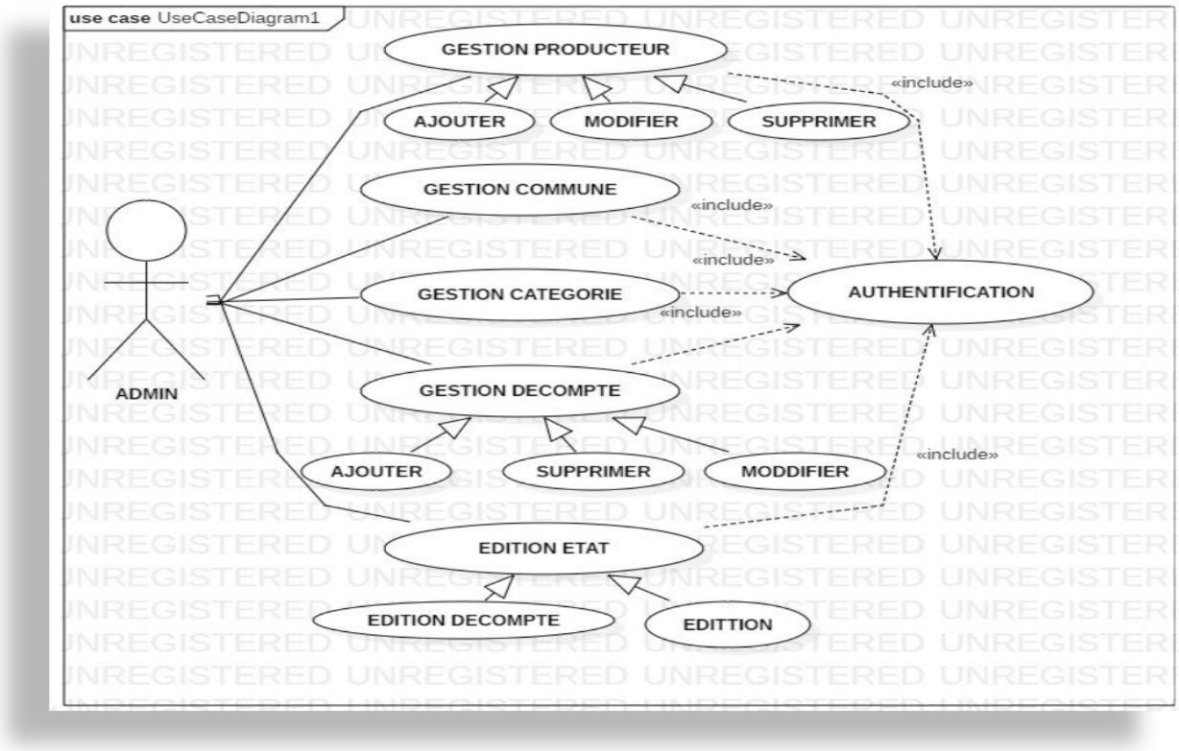


Figure 36 : diagramme de cas d'utilisation

- **Diagramme de séquence : {1} {2} {3} {4} {5}**

➢ **Pour l'authentification :**

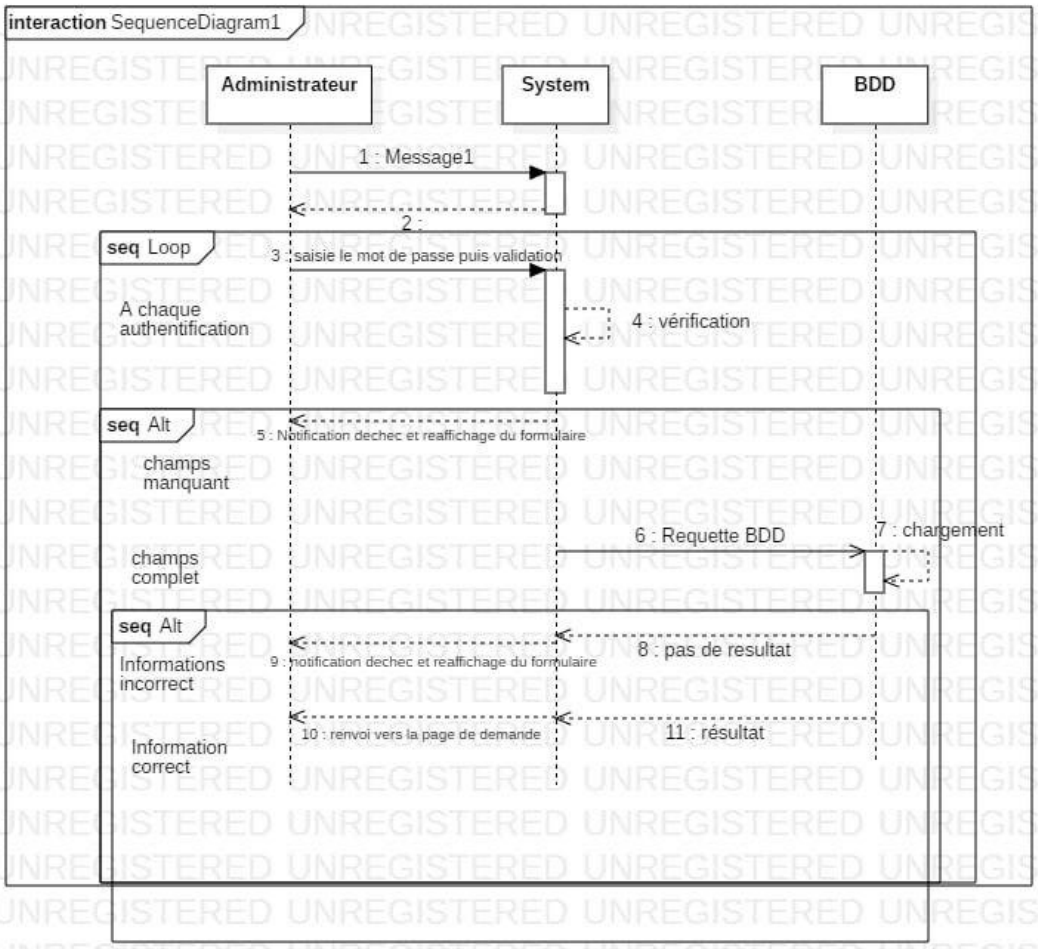


Figure 37: diagramme de l'authentification

- **Pour la recherche :**

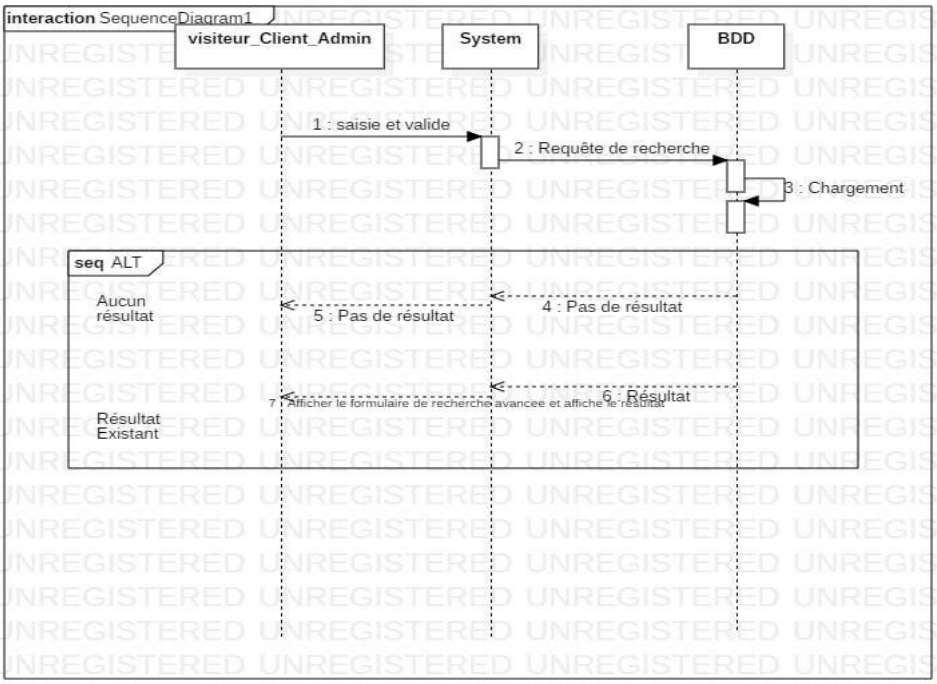


Figure 38 : diagramme de séquence pour recherche

- **Pour ajouter un producteur :**

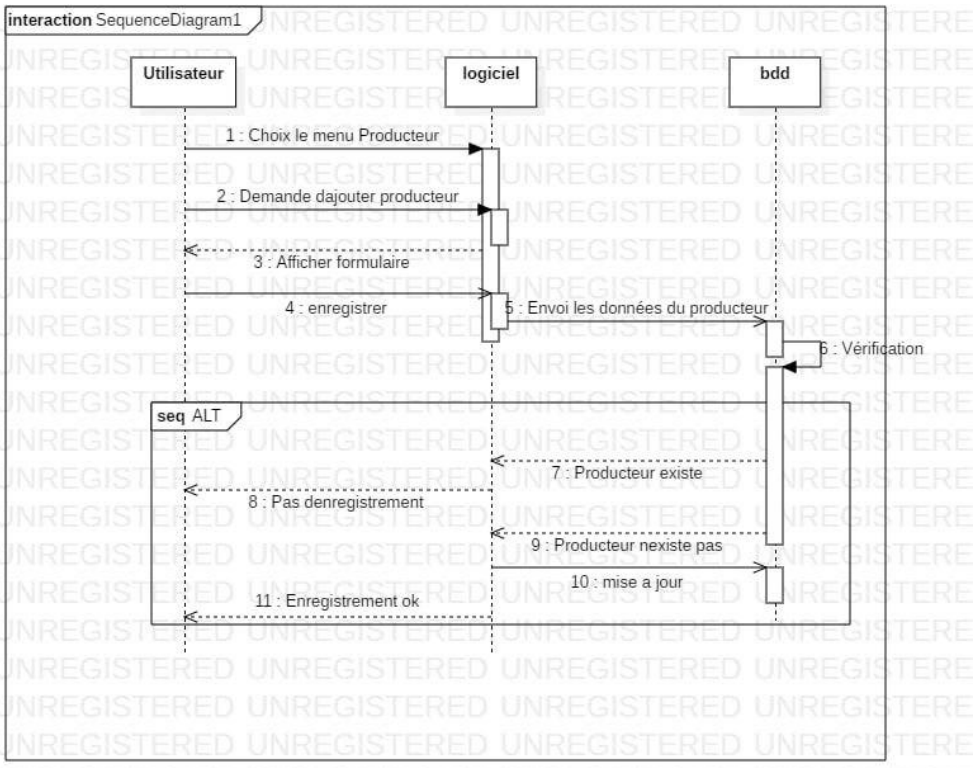


Figure 39: diagramme de séquence pour ajouter un producteur

- **Pour décompte :**

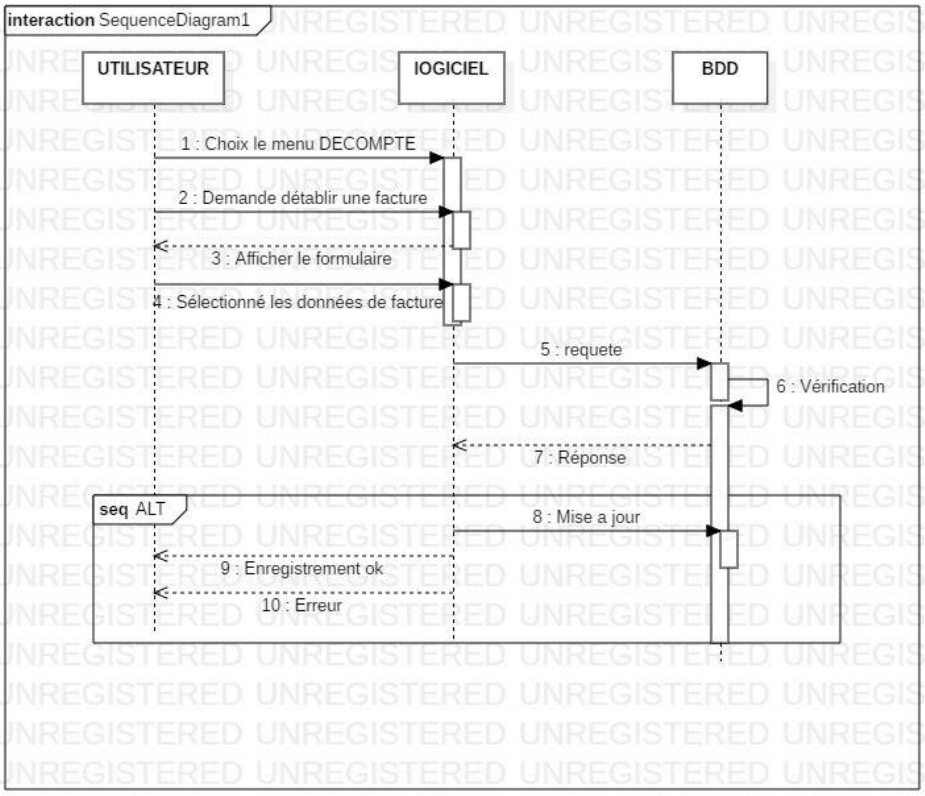


Figure 40: diagramme de séquence pour décompte

- **Pour ajouter un produit :**

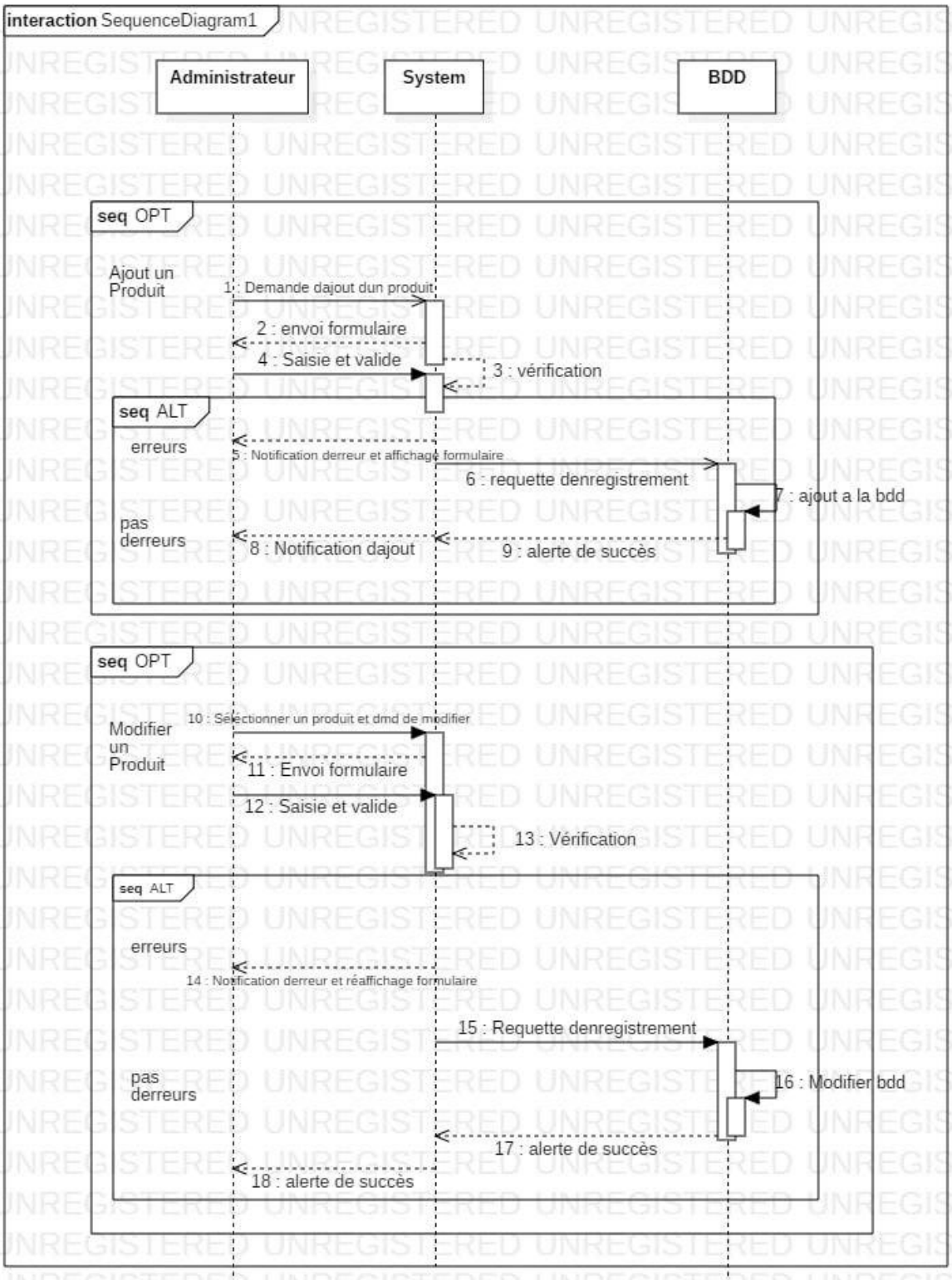


Figure 41: diagramme de séquence pour ajouter ,modifier un produit

Conclusion :

À travers ce chapitre, nous avons exploré en détail les principes de la modélisation avec le langage UML et son application dans le cadre du projet "Décompte". Les diagrammes UML présentés ont permis de donner une vision globale de l'architecture et du fonctionnement du système, facilitant ainsi la compréhension et la communication entre les parties prenantes. Cette approche de modélisation s'est avérée essentielle pour appréhender la complexité du logiciel "Décompte", offrant une compréhension d'ensemble du projet.

Chapitre 05 : Implémentation

INTRODUCTION :

Avant de commencer à coder l'application, nous nous concentrons sur la phase de spécification pour définir et clarifier les principales fonctionnalités de notre application.

Ce chapitre vise à fournir une définition précise des exigences fonctionnelles de notre applications, ainsi que des objectifs visés.

Nous présenterons également les outils choisis, discuterons des technologies utilisées dans la conception et le développement de notre système, les langages de programmation utilisés, les bases de données. De plus, nous présenterons les fonctionnalités de ce système, y compris les écrans de notre application.

Nous commençons par présenter les caractéristiques de notre système informatique utilisé pour ce projet :

Les caractéristiques du PC utilisé :

-Intel(R) Core (TM) i5-5200U CPU @ 2.20GHz (4 cœurs), ~2.2GHz
-Mémoire : 8G RAM
-Intel R HD graphics 5500
-Disque dur SSD 120G

1. Les outils et langages de programmation utilisés pour développer notre desktop :

Flutter : Un Framework open-source de Google pour le développement d'applications multiplateformes, utilisant le langage de programmation Dart.

Dart :Dart est un langage de programmation développé par Google, conçu pour créer des applications web, mobiles et serveur. Il est utilisé notamment avec le Framework Flutter pour le développement d'applications multiplateformes.

Android Studio : un environnement de développement intégré (IDE) officiel pour le développement d'applications Android. Il fournit des outils avancés pour la conception, le codage, le débogage et le déploiement d'applications Android. Source : Site officiel d'Android Studio

Visual Studio Code (VS Code) : est un éditeur de code source gratuit et open-source développé par Microsoft. Il offre des fonctionnalités avancées telles que la coloration syntaxique, l'autocomplétions intelligente, le débogage intégré, la gestion de versions avec Git, des extensions personnalisables et bien plus encore. VS Code est largement utilisé par les développeurs pour divers langages de programmation et technologies

Laravel : est un Framework web open-source écrit en PHP, conçu pour le développement rapide d'applications web. Il offre une architecture élégante et des fonctionnalités puissantes telles que la gestion des routes, les migrations de base de données, l'authentification

2. Justification de ces choix technologique :

Ce stack technologique est un choix judicieux pour ce projet car il permet de :

Créer des applications multiplateformes en utilisant une base de code unique, des performances élevées, une communauté de développeurs étendue et un soutien de Google (flutter et Dart) . Proposant des outils sophistiqués pour concevoir des applications Android natives (Android studio). Facilité l'utilisation, la solidité, aussi son architecture MVC claire et ses fonctionnalités avancées (laravel). Apprécié son large éventail de langages de programmation, ses extensions personnalisables et sa simplicité d'utilisation.

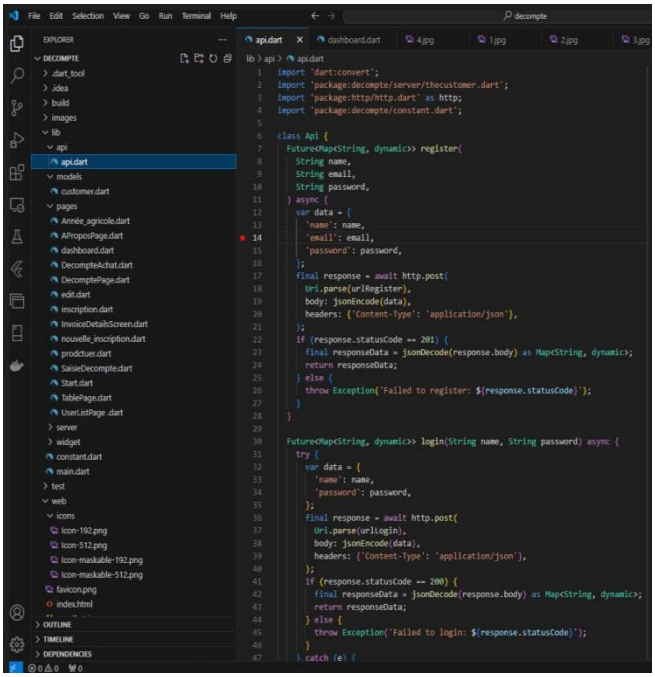
3. Les caractéristiques de notre base de données :

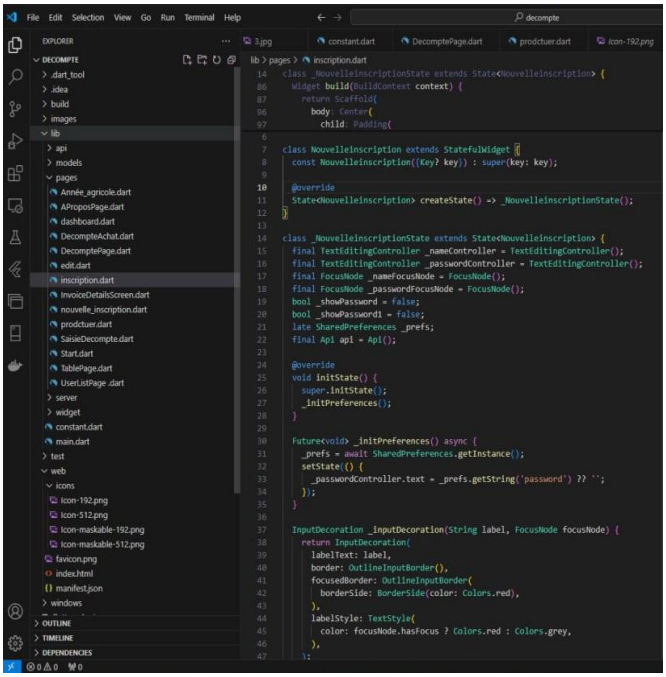
- Base de données relationnelle gérée par Microsoft SQL
- Version SGBD : SQL server 2022
- le nombre des tables 11.
- La taille de la base de données (16 MO)

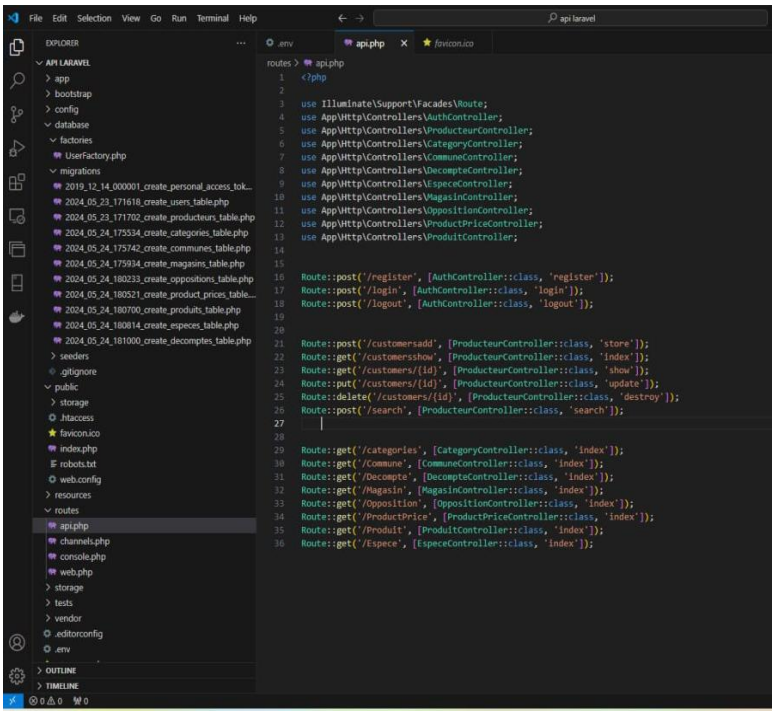
Nous avons choisi d'utiliser la même base de données que dans l'ancienne version DECOMPTE, car elle présente des qualités appréciables : sa qualité, sa disponibilité, sa fiabilité et sa facilité d'accès. Cela nous a permis de gagner du temps dans la progression de notre nouveau projet DECOMPTE pour ça 2éme version, en nous appuyant sur une source de données déjà connue et maîtrisée.

4. Extrait de notre code d'application :

Dans cette section, nous allons vous présenter un extrait du code source du logiciel "Décompte". Cet exemple vous permettra d'avoir un aperçu de la manière dont certaines fonctionnalités clés de l'application ont été implémentées.







5. Les interfaces de notre application desktop :

Nous avons choisi le nom **(DECOMPTE V1.0.0)** pour notre application dans sa deuxième version et dans cette section, nous présenterons quelques captures :

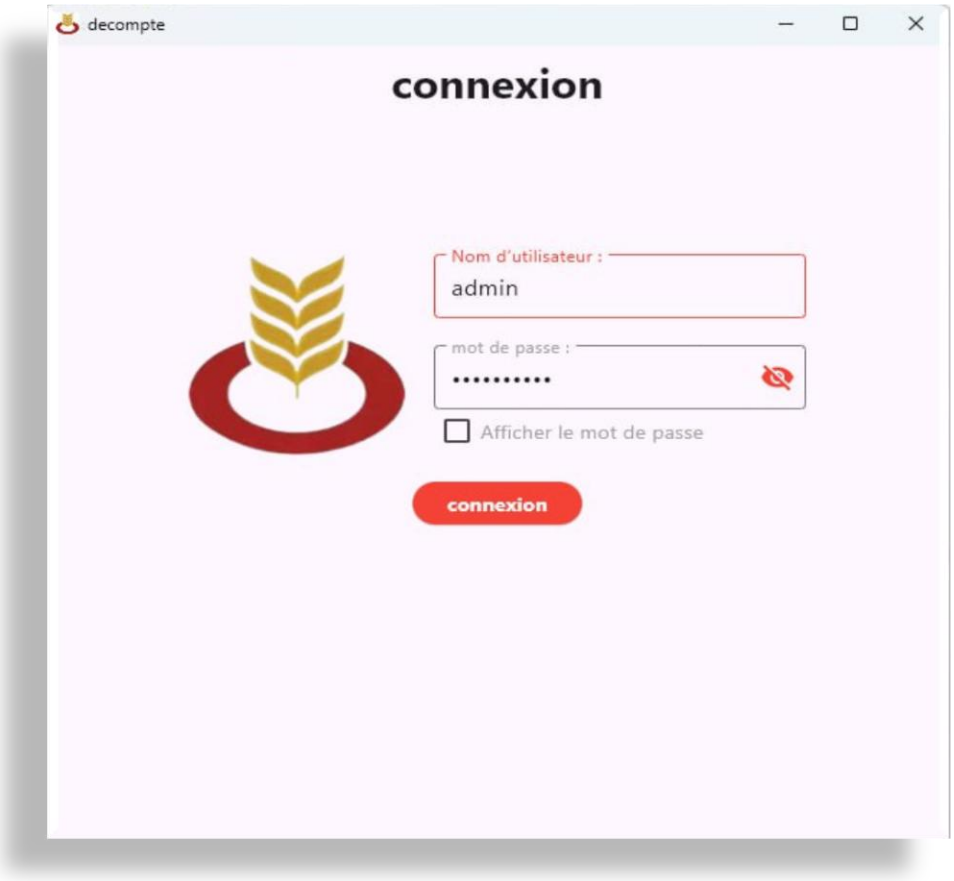


Figure 42 : Écran de connexion de l'application

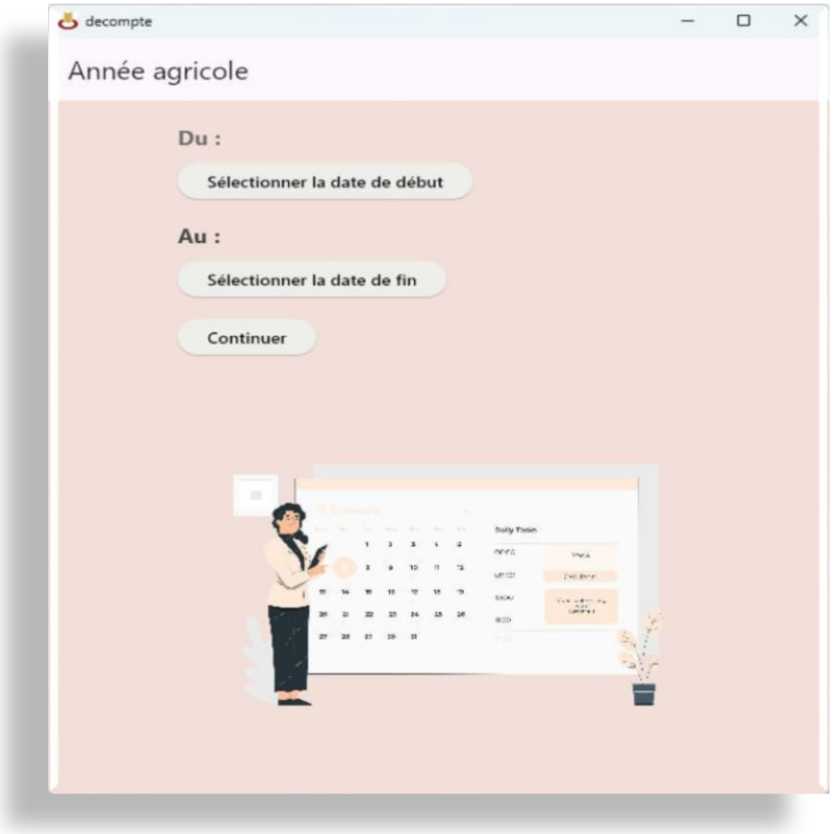


Figure 43 :Saisie année agricole

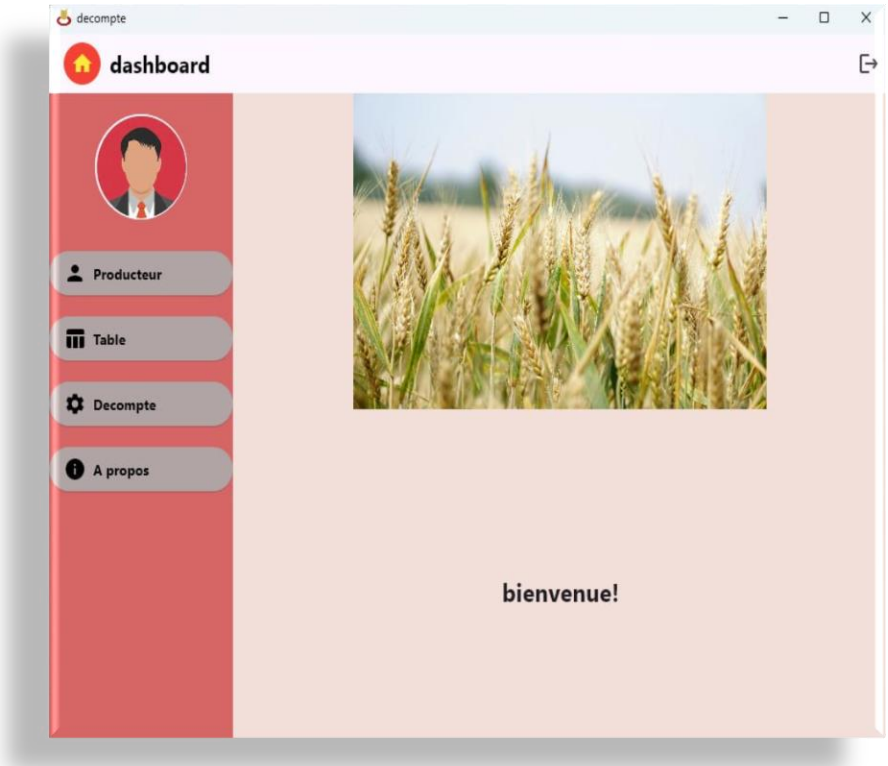


Figure 44 :Tableau de bord de l'application

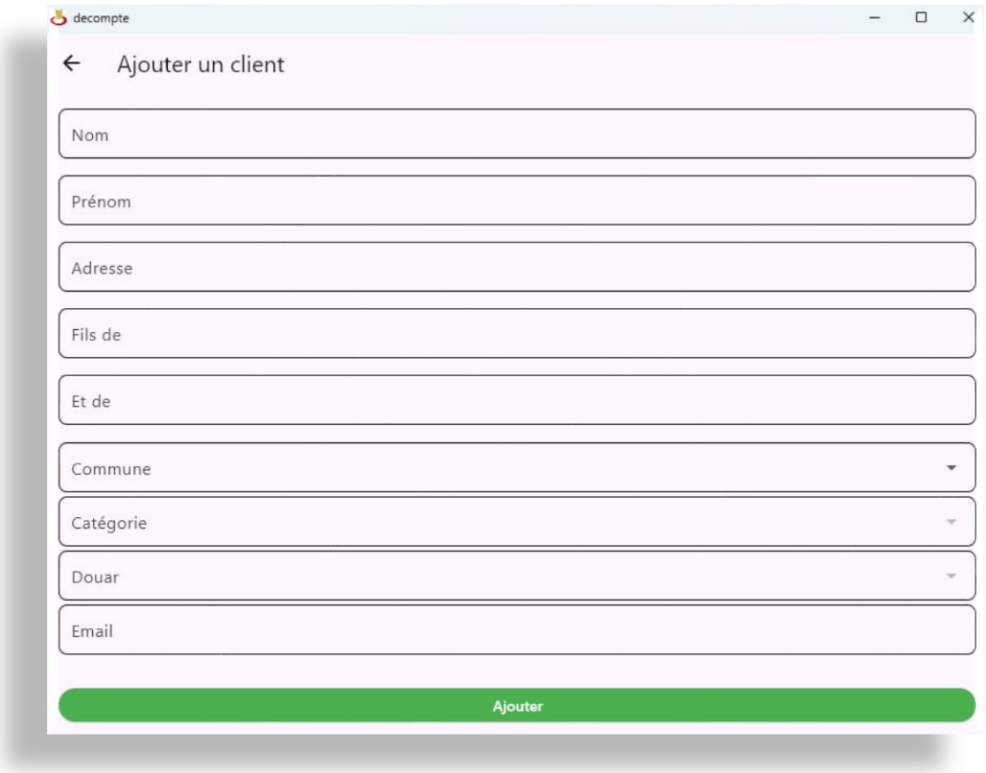


Figure 45 : Écran de création de compte client

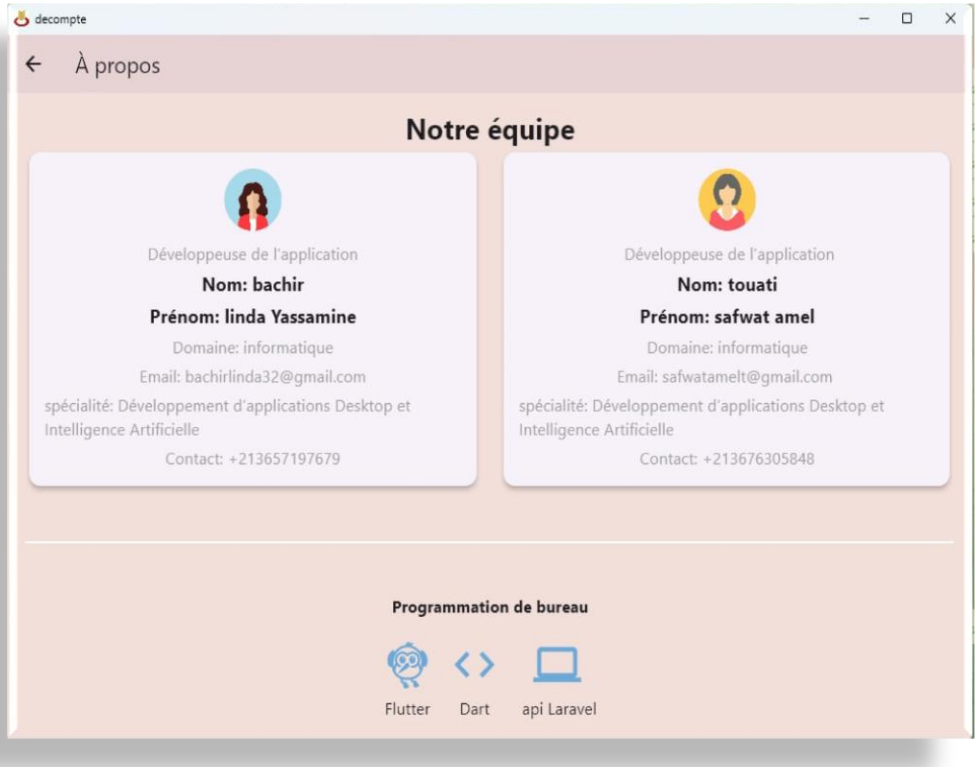


Figure 46: Écran d'information sur l'équipe

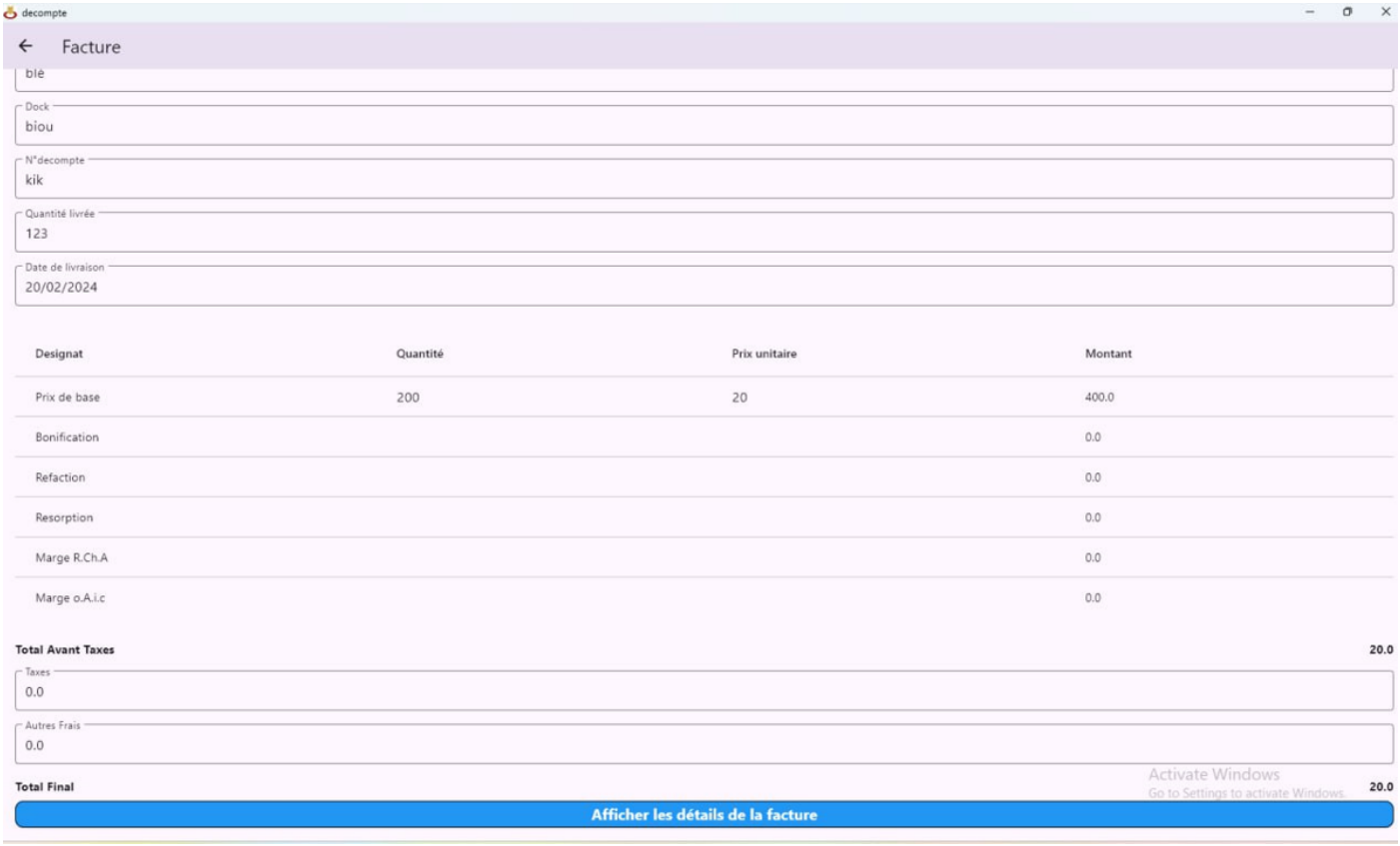


Figure 47: facture de DECOMPTE

6. Travail Supplémentaire : Développement de l'application mobile "Décompte Mobile"

Nom de l'application : **Décompte Mobile V 1.0.0**

Afin de renforcer notre projet, nous avons développé une application mobile dédiée. Cette application a été conçue pour apporter un support supplémentaire et faciliter certains aspects clés de notre activité.

Les principales fonctionnalités implémentées dans cette application mobile sont les suivantes :

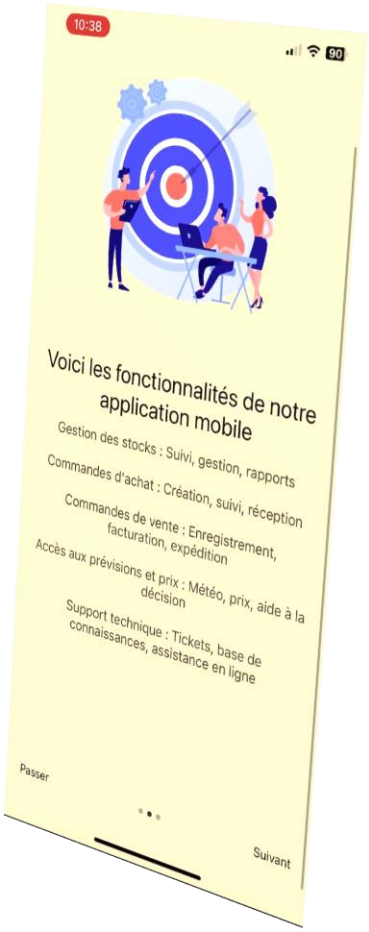
- Gestion des stocks : Suivez en temps réel les niveaux de stocks, effectuez des entrées/sorties et générez des rapports détaillés.
- Gestion des commandes d'achat et de vente : Gérez facilement vos commandes, du bon de commande à la facturation, le tout dans une interface intuitive.
- Accès aux prévisions météorologiques et aux prix du marché : Restez informé des tendances et évolutions pour mieux anticiper vos activités.
- Intégration d'un chabot basé sur l'intelligence artificielle : Bénéficiez d'une assistance personnalisée et réactive pour répondre à vos questions.
- Cette application mobile vient ainsi compléter notre système et offrir de nouvelles capacités à nos équipes sur le terrain.
- Elle a été développée comme un complément essentiel à notre projet principal, dans le but de rendre l'utilisation de nos services encore plus accessible et agréable.

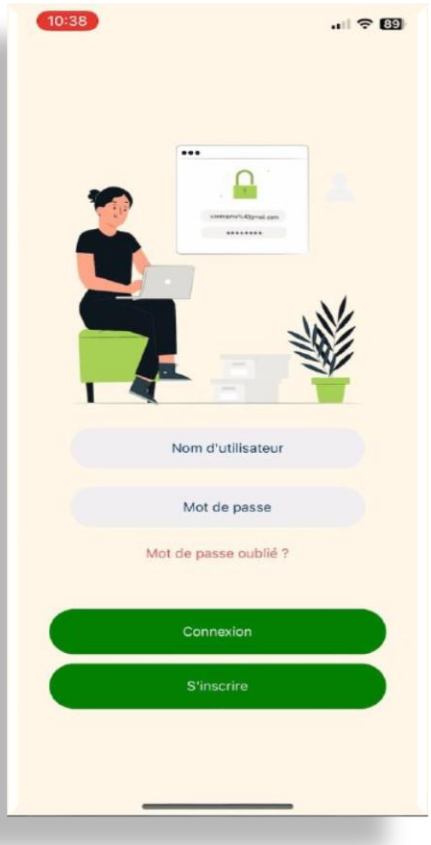
6.1 Avantages pour l'utilisateur :

- Gestion centralisée de toutes vos activités métier
- Gain de temps et d'efficacité grâce à l'automatisation
- Prise de décision éclairée avec des données en temps réel
- Assistance personnalisée et réactive avec le chabot IA Sécurité et confidentialité des données garanties.

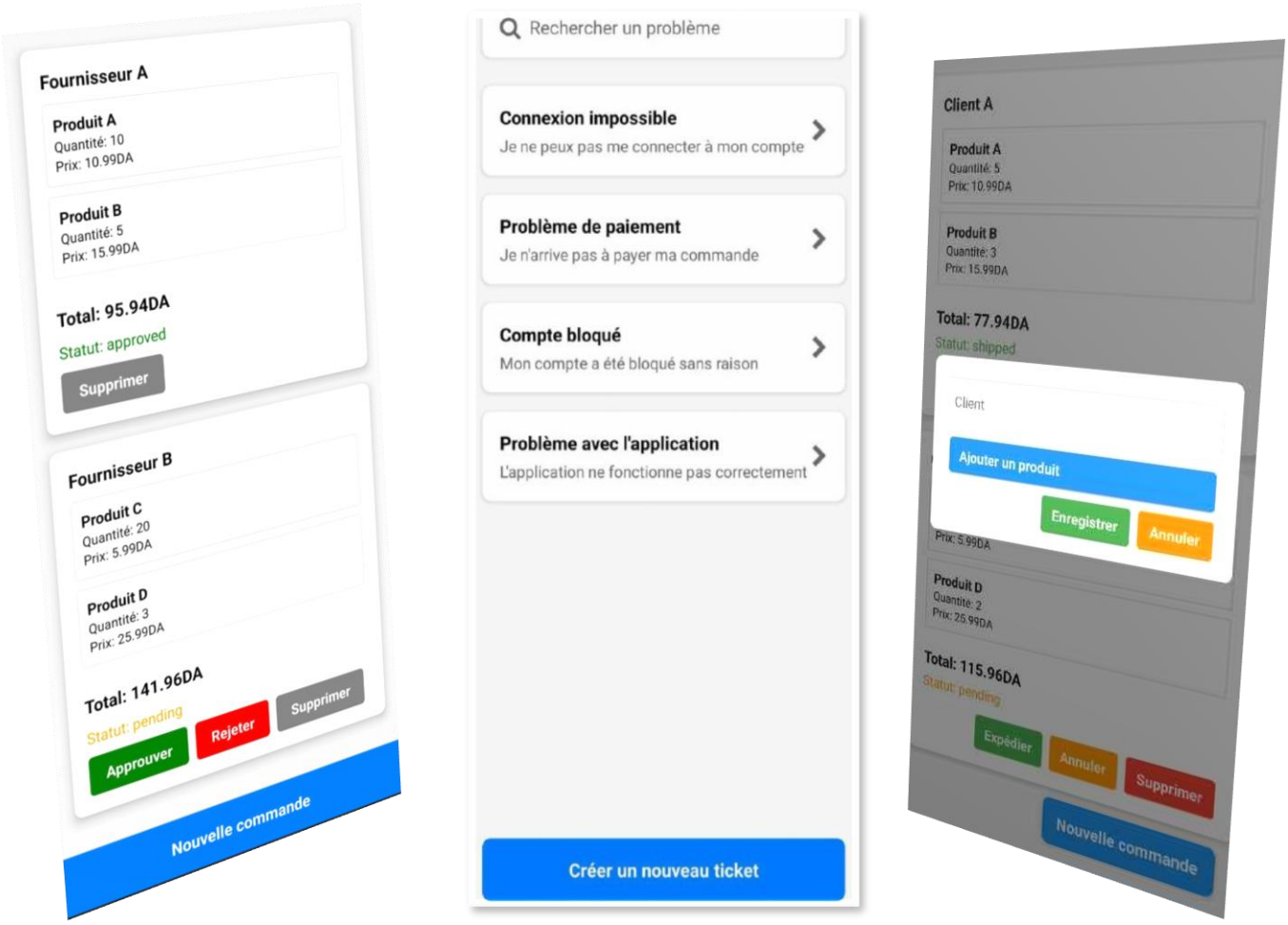

10:38
Présentation de notre application
Notre application mobile est une solution complète de gestion d'entreprise qui regroupe plusieurs fonctionnalités essentielles pour faciliter vos activités quotidiennes. Avec cette application, vous aurez accès à une gestion des stocks, des commandes d'achat et de vente, ainsi qu'à des informations météorologiques et sur les prix du marché. De plus, un module de support technique vous permet d'obtenir de l'aide rapidement en cas de besoin.
Suivant


10:38
Notre équipe
"Notre équipe de développement est composée de Linda, Yassamine, Bachir et Safwat. Ensemble, nous formons une équipe de développeurs full stack expérimentés, spécialisés dans la création d'applications de bureau et d'applications mobiles de haute qualité. "
Cliquez ici pour accéder à l'équipe
Entrer

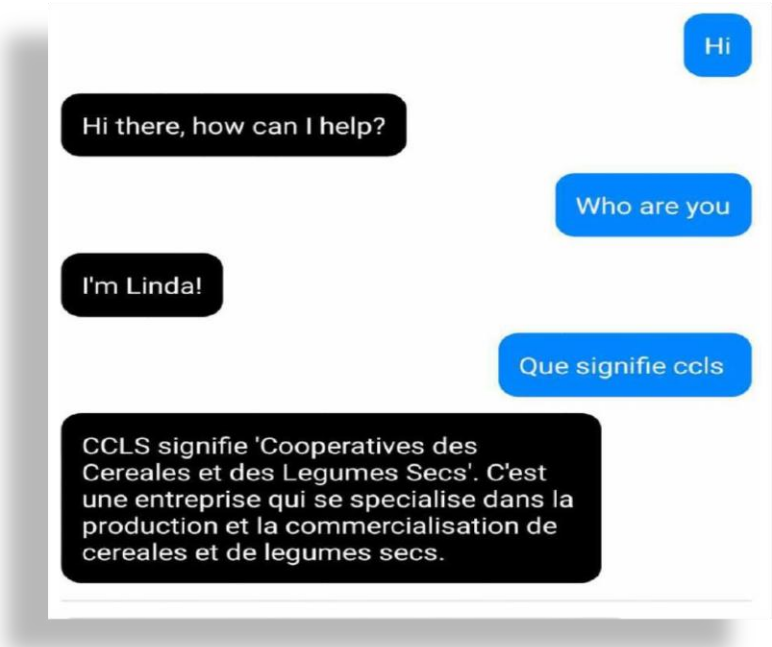

10:38
Voici les fonctionnalités de notre application mobile
Gestion des stocks : Suivi, gestion, rapports
Commandes d'achat : Création, suivi, réception
Commandes de vente : Enregistrement, facturation, expédition
Accès aux prévisions et prix : Météo, prix, aide à la décision
Support technique : Tickets, base de connaissances, assistance en ligne
Passer
Suivant


10:38
Nom d'utilisateur
Mot de passe
Mot de passe oublié ?
Connexion
S'inscrire


10:38
Mon Application Agricole
Principales Fonctionnalités
Gestion des Stocks
Commandes d'Achat
Commandes de Vente
Accès Rapide
Prévisions Météorologiques
Prix du Marché
Support Technique
Nous Contacter
© 2024 Mon Application Agricole. Tous droits réservés.



Exemple concernant assistance personnalisé avec le chatbot IA :



Conclusion :

Grâce à cette étape de spécification, nous avons pu établir de manière précise les principales fonctionnalités et objectifs de notre application. Les outils, technologies et langages de programmation requis pour le développement du système ont également été choisis. Nous passerons ensuite à la phase de développement en utilisant ces bases solides établies lors de la spécification.

CONCLUSION GÉNÉRALE :

Projet de réingénierie et de maintenance du logiciel DECOMPTE a été l'occasion de mettre en pratique les principes fondamentaux du génie logiciel. À travers cette expérience, nous avons pu apprécier toute la complexité liée à la gestion du cycle de vie des applications existantes.

En résumé nos travaux ont permis de moderniser l'interface utilisateur, d'optimiser les performances du système et d'intégrer de nouvelles fonctionnalités tout en préservant les fonctionnalités essentielles du logiciel. Ces améliorations ont eu impact positif sur l'expérience des utilisateurs finaux rendant le logiciel plus ergonomique et efficace.

Cependant, le projet n'a pas été sans défis. Nous avons dû faire face à une importante dette technique accumulée au fil des années, ce qui a complexifié certaines phases de la réingénierie. De plus, la nécessité de maintenir la compatibilité avec les systèmes existants a parfois limité nos possibilités d'innovation. Les défis liés à la migration de la base de données depuis l'ancien logiciel fonctionnant sous MS-DOS vers le nouvel outil ont également nécessité des efforts supplémentaires, notamment en raison de la non-disponibilité de certaines données historiques.

Malgré ces obstacles, nous sommes fiers du résultat obtenu. Ce projet nous a permis d'approfondir nos connaissances en génie logiciel et de développer de nouvelles compétences en matière de maintenance et de réingénierie d'applications.

En perspective, nous envisageons de poursuivre l'amélioration du logiciel Décompte en intégrant de nouvelles technologies et en automatisant davantage certains processus. L'objectif sera de garantir une évolutivité à long terme et de faciliter les futures mises à jour. Le défi sera de trouver un équilibre optimal entre innovation et préservation des fonctionnalités existantes, afin de répondre aux besoins changeants des utilisateurs tout en minimisant les coûts et les efforts de maintenance.

Nous prévoyons également d'inclure une application mobile synchronisée avec le logiciel desktop, ainsi qu'un système de paiement en ligne des factures pour éviter la collusion et alléger la charge au niveau de l'entreprise. Au-delà de ces améliorations techniques et fonctionnelles, nous envisageons également d'explorer les possibilités offertes par l'intelligence artificielle (IA) pour enrichir davantage notre solution. L'IA pourrait notamment être utilisée pour automatiser certaines tâches récurrentes, développer des fonctionnalités d'assistance intelligente, mettre en place des systèmes de prédiction et d'aide à la décision, ou encore améliorer continuellement l'expérience utilisateur.

CONCLUSION GÉNÉRALE :

L'intégration de L'IA nécessitera une réflexion approfondie sur les aspects éthiques, la protection des données personnelles et la transparence des processus, afin d'exploiter le potentiel de ces technologies de manière responsable et en adéquation avec les valeurs de notre organisation.

RÉSUMÉS :

La réingénierie est une approche qui se concentre sur la création et la restauration du logiciel. Nous avons procédé à la maintenance de l'interface afin de la rendre accessible et facile d'utilisation pour les utilisateurs, en respectant les critères d'ergonomie. Nous avons également innové au niveau de la base de données en travaillant avec des outils offrant une haute performance, une bonne scalabilité et en conservant les fonctionnalités essentielles. L'étape suivante était de tenter de mettre en œuvre le logiciel sur Windows et de faire migrer l'ancienne base de données vers un autre serveur SQL pour créer un environnement nouveau et complet pour le logiciel. Enfin, nous avons entrepris la conception de modèles conceptuels du système à l'aide de la notation UML, en nous basant exclusivement sur le programme exécutable et les tables de la base de données existante.

ABSRACT:

Reengineering is an approach that focuses on creating and restoring the software. We maintained the interface to make it accessible and easy to use for users, while respecting ergonomic criteria. We also innovated with the database, working with tools offering high performance and scalability, while retaining essential functionality. The next step was to attempt to implement the software on Windows and migrate the old database to another SQL server to create a new, complete environment for the software. Finally, we set about designing conceptual models of the system using UML notation, based exclusively on the executable program and the existing database tables.

ملخص:

إعادة الهندسة هو نهج يركز على إنشاء البرنامج واستعادته، لقد حافظنا على الواجهة لجعلها سهلة الاستخدام للمستخدمين، مع احترام المعايير المريحة. كما قمنا أيضاً بالابتكار فيما يتعلق بقاعدة البيانات، حيث عملنا بأدوات توفر أداءً عالياً وقابلية جيدة للتوسع، مع الاحتفاظ بالوظائف الأساسية. كانت الخطوة التالية هي محاولة تنفيذ البرنامج على نظام ويندوز وترحيل قاعدة البيانات القديمة إلى خادم SQL آخر لإنشاء بيئة جديدة وكاملة للبرنامج. أخيراً، شرعنا في تصميم نماذج مفاهيمية للنظام باستخدام ترميزUML ، استناداً إلى البرنامج القابل للتنفيذ والجداول الموجودة في قاعدة البيانات الحالية حصرياً.