Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Virtuelle de Tunis

UVT
جامعة تونس الافتراضية
UNIVERSITÉ VIRTUELLE DE TUNIS

MEMOIRE

DE STAGE DE FIN D'ETUDES

Pour l'obtention du

«Mastère professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)»

Présenté par :

Hamza Bouchoucha

Implémentation et développement d'une plateforme pour la gestion du Back-office d'une solution SFA.

Soutenu le : Vendredi 07 Décembre 2018

Devant le jury :

Président : Mr BEN BRAIEK Ezzedine

Encadreur : Mme CHERIF Rim

Rapporteur : Mme AMMARI Imen

Année Universitaire : 2017 / 2018

J’autorise l’étudiant à faire le dépôt initial du mémoire ou de l’essai aux fins d’évaluation.

Encadrant professionnel, Bilel Belalouna

J’autorise l’étudiant à faire le dépôt initial du mémoire ou de l’essai aux fins d’évaluation.

Encadrante académique, RIM CHERIF

Dédicace

Tous ceux qui m'ont donné leur amour ainsi que leur soutien moral, puisse Dieu, le Très Haut, vous accorde une longue vie et faire en sorte que jamais je ne vous déçoive.

Mes chers parents, Chedli et Nadia, aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les efforts permanents que vous avez fournis pour mon instruction et mon bien être.

Je dédie également ce travail à toute ma famille envers laquelle je n'éprouve que de l'amour et du respect :

à mes enseignants qui m'ont assuré une bonne formation :

à tous mes amis en souvenir des plus beaux instants qu'on a passés ensemble :

à toute personne qui compte sur moi.

Hamza Bouchoucha

Remerciements

Au terme de ce projet j'ai le plaisir de remercier vivement tous ceux qui m'ont encouragé de près ou de loin pour l'accomplissement de ce travail.

Je tiens d'abord à remercier mon encadrante Madame Rim Cherif, maitre de conférences à l'institut supérieur des études technologiques en communication de Tunis, pour ses précieux conseils, pour son soutien et son encouragement qui m'ont permis de mener ce travail à bon port.

Mes remerciements vont également à SFBT qui m'a permis de réaliser mon stage dans de bonnes conditions.

Je remercie notamment Mr. Bilel Belalouna qui m'a encadré pendant ce stage. Les nombreuses connaissances qu'il nous a apportées, tout au long de ce projet, ont fortement contribué à son bon déroulement.

Mes remerciements s'adressent pareillement aux membres de jury qui m'ont fait l'honneur de juger mon travail.

Finalement, j'exprime ma gratitude à tous mes enseignants de l'Université Virtuelle de Tunis pour leurs aides et leurs soutiens.

Table de matières

Introduction générale 1

Chapitre 1 Cadre général du projet 3

Introduction 4

I. Cadre du projet 4

I.1 Présentation de l'organisme d'accueil 4

I.2 Contexte général du Projet 5

I.3 Problématique 5

I.4 Solution proposée 6

I.5 Choix de la méthode de travail 6

I.5.1. Méthodes agiles 6

I.5.2. Processus unifié 8

I.5.3. Comparatif des méthodologies de travail : 9

I.5.4. Méthode de travail adoptée : PU 10

I.5.5. Approche de la conception 11

Conclusion 12

Chapitre 2 Spécification des besoins 13

Introduction 14

I. Etudes des besoins 14

I.1 Besoins fonctionnels 14

I.2 Besoins non fonctionnels 15

II. Diagrammes de cas d'utilisation 16

II.1 Définition du cas d'utilisation 16

II.2 Identification des acteurs 16

II.3 Diagramme de cas d'utilisation globale 16

II.4 Raffinement de cas d'utilisation 17

II.4.1. Raffinement de cas d'utilisation du cas « Gestion des clients »..... 17
II.4.2. Raffinement cas d'utilisation du cas « Gestion des articles » 19
II.4.3. Raffinement de cas d'utilisation du cas « Gestion des visites »..... 20
II.4.4. Raffinement de cas d'utilisation du cas « Gestion des commandes » 21
II.4.5. Raffinement de cas d'utilisation du cas « Gestion des inspecteurs commerciaux » 22
II.4.6. Raffinement de cas d'utilisation du cas « Gestion des usines » 23
II.4.7. Raffinement de cas d'utilisation du cas « Gestion des catégories d'articles » 24
II.4.8. Raffinement de cas d'utilisation du cas « Gestion des familles d'articles » 25
II.4.9. Raffinement de cas d'utilisation du cas « Gestion des sous familles d'articles »26
III. Diagramme de Séquence système :..... 27
III.1. Diagramme de séquence système de l'opération « Planifier une visite »..... 27
III.2. Diagramme de séquence système de l'opération « Modifier un article »..... 28
III.3. Diagramme de séquence système de l'opération « Chercher un article » 29
Conclusion..... 29
Chapitre 3 Conception du système..... 30
Introduction 31
I. Spécification du style d'architecture..... 31
I.1 Organisation du modèle de spécification logicielle 31
I.2 Architecture physique de l'application 33
II. Diagrammes de séquence et de classes..... 34
II.1. Définition du diagramme de séquence 34
II.2. Définition du diagramme de classes..... 34
II.3. Diagrammes des séquences détaillés..... 35
II.3.1. Diagramme de séquence détaillé « Modifier un article » 35

II.3.2. Diagramme de séquence Détaillé « Ajouter un client » 36
II.3.3. Diagramme de séquence Détaillé « Chercher un article » 37
II.3.4. Diagramme de séquence Détaillé « Supprimer un article » 38
II.4. Conception du niveau de données 39
II.4.1. Diagramme de classe 39
II.4.2. Schéma relationnel 40
Conclusion 41
Chapitre 4 Réalisation 42
Introduction 43
I. Environnement de développement 43
I.1 Environnement matériel 43
I.2 Environnement Logiciel 43
II. Principales interfaces graphiques 45
II.1. Interface d'authentification 45
II.2. Interface d'accueil 46
II.3. Interface liste des clients 47
II.4. Interface d'ajout d'un client 50
II.5. Interface de modification d'un client 51
II.6. Interface de Liste des articles 53
II.7. Interface d'ajout d'article 53
II.8. Interface de d'ajout d'une visite 54
Conclusion 54
Conclusion générale 55
Bibliographie 56

Tables des illustrations

Figure 1 : Processus Agile 7
Figure 2: Axe de modélisation de système 11
Figure 3: Schéma de processus de modélisation 12
Figure 4: Diagramme de cas d'utilisation globale 17
Figure 5: Raffinement de cas d'utilisation du cas « Gestion des clients » 18
Figure 6: Raffinement cas d'utilisation du cas « Gestion des articles » 19
Figure 7: Raffinement de cas d'utilisation du cas « Gestion des visites » 20
Figure 8: Raffinement de cas d'utilisation du cas « Gestion des commandes » 21
Figure 9: Raffinement de cas d'utilisation du cas « Gestion des inspecteurs commerciaux» 22
Figure 10: Raffinement de cas d'utilisation du cas « Gestion des usines» 23
Figure 11: Raffinement de cas d'utilisation du cas « Gestion des catégories d'articles» 24
Figure 12: Raffinement de cas d'utilisation du cas « Gestion des familles d'articles» 25
Figure 13: Raffinement de cas d'utilisation du cas « Gestion des sous familles d'articles» 26
Figure 14: Diagramme de séquence système « Planifier une visite » 27
Figure 15: Diagramme de séquence système « Modifier un article » 28
Figure 16:Diagramme de séquence système « Chercher un article » 29
Figure 17: Organisation du modèle de spécication logicielle (diagramme de packages) ... 31
Figure 18: Architecture de l'application 32
Figure 19 : Diagramme de composants du système 33
Figure 20: Diagramme de séquence détaillé « Modifier un article » 35
Figure 21: Diagramme de séquence détaillé « Ajouter un client » 36
Figure 22: Diagramme de séquence détaillé « Chercher un article » 37
Figure 23:Diagramme de séquence Détaillé « Supprimer un article » 38
Figure 24: Diagramme de classe globale 39
Figure 25: Modèle de données des entités 41
Figure 26 : Page d'authentification 45
Figure 27 : Interface d'erreur lors de la saisie des mots de passe 46
Figure 28:Interface d'accueil 46
Figure 29: Liste des clients 47
Figure 30: Chercher un client Selon la première lettre du nom de client 48
Figure 31:Chercher un client selon son secteur 48

Figure 32:Echec d'un Recherche 49
Figure 33:Message de confirmation de suppression 49
Figure 34: La nouvelle liste après suppression d'un client 50
Figure 35:Interface d'ajout d'un client 50
Figure 36:Message de succès d'ajout 51
Figure 37: Interface de modification d'un client 51
Figure 38: Succès après modification d'un client 52
Figure 39: Liste après modification 52
Figure 40: Liste des articles 53
Figure 41: Ajout d'article 53
Figure 42: Ajout d'une visite 54

Tables des Tableaux

Tableau 1: Analyse des méthodes et des méthodologies de travail 9
Tableau 2: Description du cas d'utilisation «Gestion des client» 18
Tableau 3: Description du cas d'utilisation «Gestion des articles» 19
Tableau 4: Description du cas d'utilisation «Gestion des visites» 20
Tableau 5: Description du cas d'utilisation «Gestion des commandes» 21
Tableau 6: Description du cas d'utilisation «Gestion des inspecteurs commerciaux» 22
Tableau 7: Description du cas d'utilisation «Gestion des usines» 23
Tableau 8: Description du cas d'utilisation «Gestion des catégories d'articles» 24
Tableau 9: Description du cas d'utilisation «Gestion des familles d'articles» 25
Tableau 10: Description du cas d'utilisation «Gestion des sous familles d'articles» 26

Introduction générale

Dans le monde commercial, l'importance d'optimiser votre processus de vente, de retenir les clients existants et de développer l'entreprise est primordiale, c'est pourquoi l'automatisation de cycle de vente est aujourd'hui un enjeu crucial en B2B. Cette stratégie contribue à améliorer les performances de l'entreprise et de son service commercial. En effet, ces dernières années les sociétés tunisiennes s'intéressent de plus en plus à la SFA (Sales Force Automation), soit la gestion des forces de ventes. Cet intérêt s'explique par les multiples bénéfices qu'offre la SFA par rapport aux autres méthodes de vente telles que la vente par des échanges électroniques formalisés.

La gestion des forces de ventes permet la saisie des objectifs et des prévisions; l'élaboration d'un plan d'action et de diffusion personnalisée aux commerciaux ; la gestion des affaires depuis l'opportunité jusqu'à la commande ; le suivi des clients et des activités ; la gestion du planning des commerciaux : l'analyse des ventes et statistiques sous-forme de tableaux de bord notamment.

Notre projet au sein de la Société de Fabrication des Boissons de Tunisie (SFBT) consiste à concevoir et à implémenter une solution informatique qui automatise la vente en terrain et l'optimise. Pour cela l'équipe de développement a créé une application mobile de planification de tournées (sortie des agents sur le terrain) qui permettra d'automatiser les livraisons, ainsi que la gestion du stock des produits (visualisation de ses états).

Toutefois, cette application nécessite l'implémentation d'un backoffice qui permettra la gestion des articles, des inspecteurs commerciaux, des commandes, des clients, des factures, des usines, les tournées, etc.

Le présent rapport est organisé comme suit :

Le premier chapitre intitulé « Cadre général du projet » sera consacré à la présentation de l'organisme d'accueil, ainsi que la mise en contexte du projet. Nous présenterons aussi l'étude de l'existant qui consiste à décrire le fonctionnement de la plateforme.

Par ailleurs, un second chapitre rendra compte de l'étude fonctionnelle, suivie de l'étude technique. Ce chapitre proposera des diagrammes, des présentations des plateformes et technologies utilisées.

De surcroît, un troisième chapitre sera entièrement dédié à la conception du projet. Cela dit, il sera question de mettre en avant l'ensemble des diagrammes nécessaires à la bonne compréhension et à la cohésion du système ainsi que du diagramme de cas d'utilisation au diagramme de classe. Le tout accompagné des scénarios et descriptions nécessaires.

Le quatrième et le dernier chapitre sera voué à la description détaillée des outils utilisés pour développer l'application web, l'architecture du système et le matériel de déploiement de l'application.

Enfin, notre rapport se terminera par une conclusion générale résumant notre travail, présentant les acquis techniques et personnels et proposant quelques perspectives pour l'amélioration de notre application.

Chapitre 1

Cadre général du projet

Introduction

Le premier chapitre sera dédié pour la présentation générale du projet. Tout d'abord, nous allons présenter l'organisme d'accueil. Ensuite, nous allons décrire brièvement le sujet, la problématique à résoudre, ainsi que l'étude et la critique de l'existant. Enfin, nous allons clôturer par la solution proposée et la méthodologie de travail adoptée.

I. Cadre du projet

Le projet s'intitule «Implémentation et développement d'une plateforme pour la gestion du Back-office d'une solution SFA.». Il a été élaboré dans le cadre de la préparation du projet de fin d'études présenté en vue de l'obtention du diplôme Mastère professionnel en Nouvelles Technologies des Télécommunications et Réseaux sein de Université virtuelle de Tunis pour l'année universitaire 2017-2018.

I.1 Présentation de l'organisme d'accueil

Afin de situer notre travail dans son environnement de réalisation, nous présentons dans cette section l'organisme d'accueil.

La Société de Fabrication des Boissons de Tunisie (S.F.B.T.), connue jusqu'en 2012 sous l'appellation Société Frigorifique et Brasserie de Tunis, est un groupe tunisien du secteur de l'industrie agroalimentaire centré sur la production et la commercialisation de boissons. L'entreprise originelle fondée en 1889 sous le protectorat français s'est progressivement mutée en groupe et occupe une position de leader sur le marché tunisien avec (1) :

- environ 90 % du marché national de la bière pour les marques emblématiques Celtia et Stella et les produits sous licence 33 Extra Dry et Beck's (1),
- 90 % de celui des sodas essentiellement sous licence Coca Cola (1),
- 40 à 50 % du marché des eaux minérales avec des marques historiques comme Safia ou AïnGarci, ou plus récentes comme Marwa et Cristaline (1).

Le Groupe S.F.B.T est composé de plusieurs sociétés industrielles, opérant en Tunisie à travers deux domaines d'activités : boissons et autres. Le Groupe exerce ses activités sur tout le territoire tunisien, Nord, Centre et Sud (1).

Renseignements généraux

Dénomination sociale : Société de Fabrication des Boissons de Tunisie–S.F.B.T (2)

Date de Constitution : 03 Juin 1925 (2)

Forme juridique : Société Anonyme de droit commun (2)

Secteur d'activité : Alimentation: boissons gazeuses, Bière (2)

Registre de commerce N° : B 12956 1997 (2)

Siège social : 5, boulevard de la terre – Centre urbain nord 1082 Tunis (2).

I.2 Contexte général du Projet

Notre projet vise à mettre en place une nouvelle plateforme de gestion du backoffice au sein de la société à la place de l'ancienne plateforme.

Cette application web servira à organiser et unifier les processus de gestion de toutes les ventes de l'entreprise essentiellement ; la gestion des usines, la gestion des inspecteurs commerciaux, la gestion des clients, la gestion des articles, la gestion des commandes... tout en ayant un référentiel unique pour tous.

Il offrira aussi aux administrateurs un accès rapide, un contrôle des erreurs, une réduction des tâches quotidiennes et de saisie à effectuer, une interface web, beaucoup plus souple et ergonomique.

I.3 Problématique

De nos jours, les entreprises souhaitent s'ouvrir sur les nouvelles technologies et les utiliser pour avoir un meilleur rendement, et ainsi intégrer des solutions de gestion complètes, paramétrables et flexibles pour la gestion de tout ce qui se rapporte à leur métier.

Certes, il existe des nombreuses solutions sur le marché, mais une société qui vit une croissance comme dans notre cas, finit inévitablement par atteindre un niveau de complexité de son processus, chose qui oblige à implanter une nouvelle plateforme pour gérer le Back-office développer avec Microsoft .NET au lieu de l'ancienne plateforme qui est développée avec une technologie ancienne Visual Studio LightSwitch qui n'est plus supportée.

Ce dernier présente quelques insuffisances en matière de fonctionnalités offertes ainsi que la technologie utilisée :

- Difficultés de gérer les données de la base de données.
- la maintenabilité.
- Difficultés à trouver de l'information en temps réel (liste des articles, les fiches clients).

I.4 Solution proposée

L'objectif de ce travail est d'améliorer la qualité des services de l'entreprise, assurer la fiabilité des données et ceci en mettant en place une nouvelle application web permettant de gérer le backoffice de la solution SFA et de remédier aux différentes problématiques ressenties au sein de l'entreprise.

Cette application web permettra aussi de :

- Gérer de façon complète les entités de bases, à savoir les commandes, les clients, les articles, les visites, etc.
- Intégrer une solution complète et évolutive qui pourra être intégrée avec les autres solutions de l'entreprise.
- Mettre à jour, et paramétrer les données.

I.5 Choix de la méthode de travail

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en œuvre d'une planification organisationnelle tout au long de son cycle de vie. C'est ainsi qu'est apparue la notion de méthode. Une méthode, dans le contexte informatique, peut être dénie comme une démarche fournissant une méthodologie et des notations standards qui aident à concevoir des logiciels de qualité. Le but de cette partie est de montrer la méthode de travail que nous avons suivi tout au long du projet (3).

I.5.1. Méthodes agiles

Le développement de logiciels agile se réfère à un groupe de logiciels méthodologies de développement basé sur le développement itératif, incrémentale, où les exigences et les solutions évoluent grâce à la collaboration entre les équipes inter-fonctionnelles. Les méthodes agiles ou processus agiles favorisent généralement un processus de gestion de projet rigoureuse qui encourage les inspections fréquentes et l'adaptation, une philosophie de leadership qui favorise le travail d'équipe, l'auto-organisation et de responsabilité, un ensemble de meilleures pratiques

d'ingénierie destiné à permettre la livraison rapide des logiciels de haute qualité, et une approche d'a aires qui aligne le développement avec les besoins des clients et des objectifs de l'entreprise .

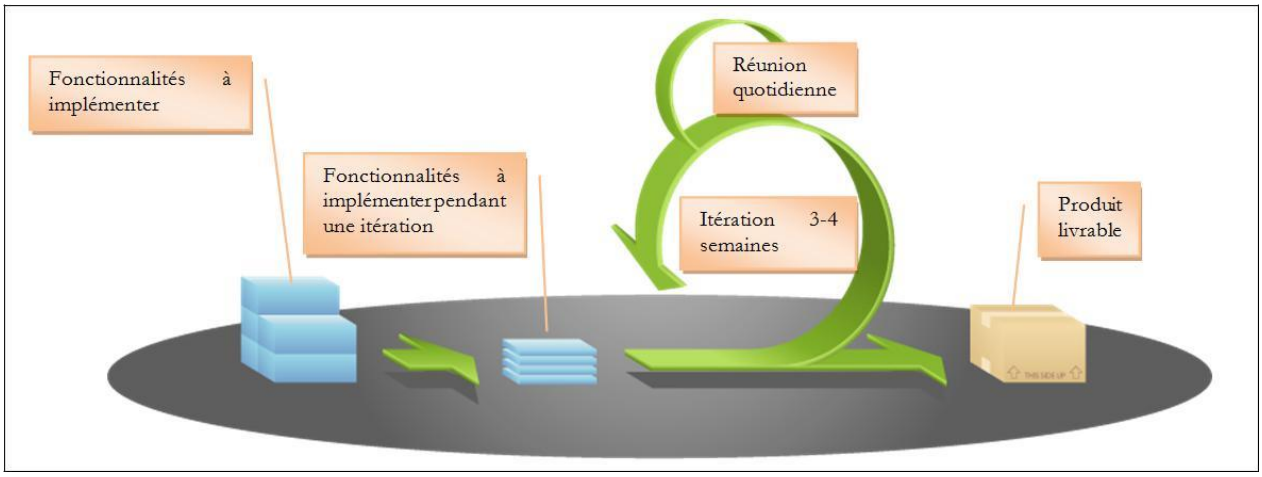


Figure 1 : Processus Agile

La Figure 1 montre que les méthodes agiles utilisent un principe de développement itératif qui consiste à scinder le projet en plusieurs étapes qu'on appelle itérations.

Ces itérations sont en fait des mini-projets définis avec le client en détaillant les différentes fonctionnalités qui seront développées en fonction de leur priorité. Le chef de projet établi alors une macro planning correspondant aux tâches nécessaires pour le développement de ces fonctionnalités. Chaque itération se déroule d'une à quatre semaines en effectuant chaque jour une réunion quotidienne pour obtenir un produit livrable.

Parmi les méthodes agiles nous citons :

- Extrem programming (Xp) :

Elle est orientée sur l'aspect réalisation d'une application. XP est adapté aux équipes réduites avec des besoins changeants. L'Extreme Programming repose sur des cycles rapides de développement (des itérations de quelques semaines) dont les étapes sont les suivantes :

Une phase d'exploration détermine les scénarios client qui seront fournis pendant cette itération.

- L'équipe transforme les scénarios en tâches à réaliser et en tests fonctionnels.
- Chaque développeur s'attribue des tâches et les réalise avec un binôme.
- Tous les tests fonctionnels passent alors le produit est livré (4).

- Scrum :

Le principe de base de la méthode de travail Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe d'une à quatre semaines, appelées sprints (5).

Chaque sprint possède un but à atteindre à partir duquel sont choisies les fonctionnalités à implémenter dans ce sprint. Un sprint aboutit toujours à la livraison d'un produit partiel fonctionnel (5). L'utilisation de la méthode Scrum offre la possibilité de développer uniquement les fonctionnalités qui apportent une valeur ajoutée au produit, en toute transparence, avec le souci de la qualité et du respect des délais et avec un retour rapide de la part du client (5).

I.5.2. Processus unifié

Le Processus unifié (PU) est une méthode de prise en charge du cycle de vie d'un logiciel et du développement pour les logiciels orientés objets. C'est une méthode générique, itérative et incrémentale. Parmi les méthodes du processus unifié (UP) nous citons :

❖ RUP (Rational Unified Process) :

RUP est l'une des plus célèbres implémentations de la méthode PU permettant de donner un cadre au développement logiciel, répondant aux exigences fondamentales préconisées par les créateurs d'UML :

- Une méthode de développement doit être guidée par les besoins des utilisateurs.
- Elle doit être centrée sur l'architecture logicielle.
- Elle doit être itérative et incrémentale.

❖ 2TUP (2 tracks unified process) :

2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent entre les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (6). Le processus s'articule ensuite autour de trois phases essentielles :

- Une phase de branche technique.
- Une phase de branche fonctionnelle.
- Une phase de conception et de la réalisation (6).

I.5.3. Comparatif des méthodologies de travail :

Il est très important de bien choisir la méthode de travail qui répond parfaitement à ces besoins et qui s'adapte le mieux aux exigences du client. Alors, il est nécessaire de faire une étude comparative entre les méthodes qu'on a citées avant, pour savoir sélectionner la meilleure qui s'adapte au projet.

Le tableau 1 ci-dessous, décrit les méthodes et les méthodologies de travail ainsi que les points forts et les points faibles de chacune. Le critère de comparaison se base sur le choix d'une méthode de travail ayant une équipe de développeurs à nombre réduit.

Tableau récapitulatif des différentes méthodologies de travail :

Tableau 1: Analyse des méthodes et des méthodologies de travail

Méthodologie	**Description**	**Points forts**	**Points faibles**
Méthodologie XP	• Ensemble de bonnes pratiques de développement (idéal pour le travail en groupe) Projets ciblés de moins de dix personnes. • Simple à mettre en œuvre.	• Simple à mettre en œuvre. • Applicable pour les projets complexes.	• XP ne peut s'adresser qu'à des projets petits ou moyens en raison de ses caractéristiques.
2TUP	• 2TUP propose un cycle de développement en Y. • Convient aux projets de toute taille.	• Itératif • Fait une large place à la technologie et à la gestion du risque. • Définit les profils des livrables, les Planning, les prototypes.	• Superficiel sur les phases situées en amont et en aval du développement • Ne propose pas de documents types.

RUP	• Méthodologie centrée sur l'architecture et couplée aux diagrammes UML. • Concerne des projets de plus dix personnes Processus complet assisté par des outils exhaustifs.	• Itératif spécifie le dialogue entre les différents intervenants du projet • Propose des modèles de document et des modèles pour des projets types Rôles bien définis	• Coûteux à personnaliser • Très axé processus, au détriment du développement Lourd, largement étendu, il peut être difficile à mettre en œuvre de façon spécifique • Convient pour les grands projets qui génèrent beaucoup de documentation.
SCRUM	• Méthode Agile Projet décomposé en « Sprints » • ScrumMaster,pas de chef projet.	• Itératif Incrémental ; Facile à mettre en œuvre. • La plus populaire des méthodes Agile.	Ne couvre aucune technique d'ingénierie du logiciel Pratiques de qualité du

D'après le tableau 1, nous constatons que toutes ces méthodologies et ces méthodes de travail proposent de travailler de façon itérative, que ce soit au niveau des plannings, des spécifications ou des développements... Si l'itératif s'est imposé, c'est parce qu'il réduit la complexité de réalisation des phases en travaillant par approches successives et incrémentales.

I.5.4. Méthode de travail adoptée : PU

PU se basant sur UML (Unified Modelling Langage) : Le langage de modélisation UML constitue un standard pour la modélisation des applications informatiques.

Tant qu'une bonne méthodologie de réalisation de logiciels suppose une bonne maîtrise de la distinction entre l'analyse et la conception, On a choisi la méthodologie PU basée sur UML qui permet d'exprimer le résultat des différents phases du développement de notre projet. En UML, les mêmes modèles peuvent être utilisés à différents niveaux d'abstraction du plus conceptuel à l'implantation (7).

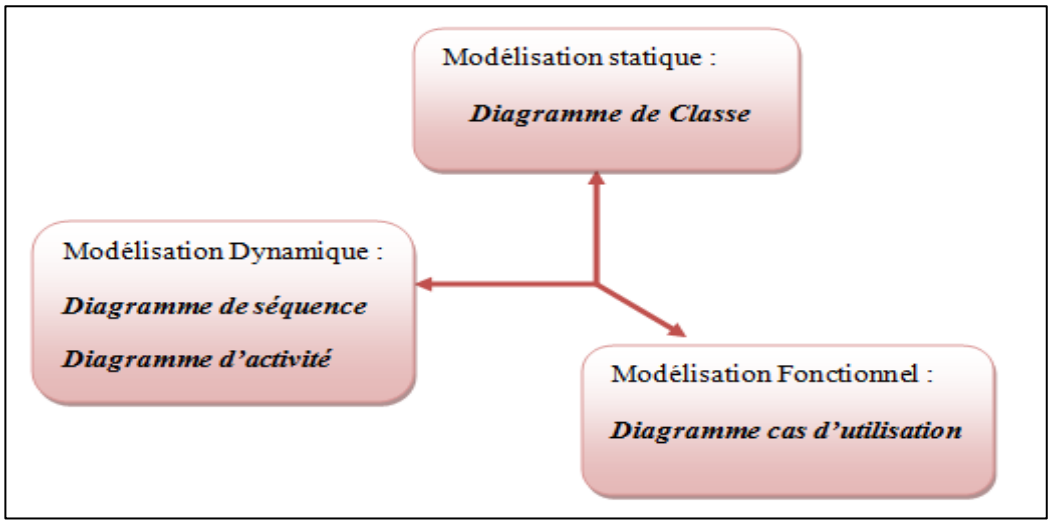


Figure 2: Axe de modélisation de système

Présentation du PU :

Le processus unifié — PU, ou UP (anglais : unified process) — est une méthode de développement pour les logiciels orientés objets. C'est une méthode générique, itérative et incrémentale, contrairement à la méthode séquentielle Merise (ou SADT).

PU vient compléter la systémique des modèles UML. Elle est le résultat final d'une évolution de l'approche d'Ericsson qui est au fondement d'une des premières méthodes de développement pour applications orientées objets, la méthode Objectory Process (1987). Objectory Process (version 1 à 3.8 en 1995) a elle-même servi de base à la société Rational pour la création de Rational Objectory Process (1997) (version 4.1), parente direct de RUP en 1998 (7).

I.5.5. Approche de la conception

La partie conception est la partie la plus compliquée, à partir de laquelle nous passons des besoins des utilisateurs au code source de l'application.

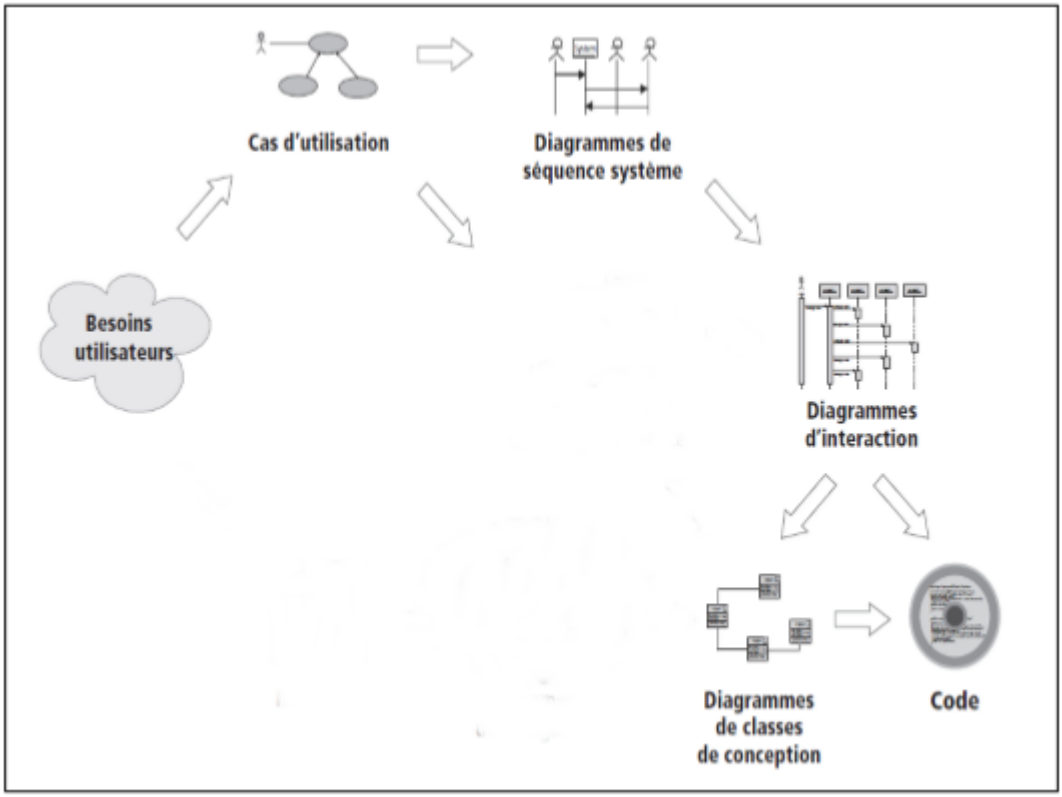


Figure 3: Schéma de processus de modélisation

La Figure 3 schématise les étapes nécessaires afin d'aboutir aux codes permettant la réalisation de l'application. En effet, à partir des besoins fonctionnels nous allons identifier les diagrammes de cas d'utilisation. Puis la description du scénario de chaque cas va nous permettre de définir les diagrammes de séquence systèmes.

Puis en se basant sur les diagrammes de séquences systèmes et la description détaillé des cas d'utilisation, nous allons proposer la modélisation des diagrammes de séquences détaillés qui aident à écrire le code à l'intérieur des opérations, ceci est un moyen utile à la définition des objets de chaque classe qui communiquent en s'envoyant des messages.

Finalement, en se basant sur ces derniers, la partie qui suit présente la modélisation des diagrammes de classes de conception qu'illustre la structure du code par ses classes, ses attributs, ses relations et ses méthodes (3).

Conclusion

Ce chapitre a été consacré à la présentation du cadre général de notre stage. Puis, nous avons défini le contexte global du projet, ainsi que les objectifs à atteindre. Nous avons exposé aussi la méthodologie que nous avons suivie. En vue de suivre un avancement logique dans ce rapport, une étude concernant l'analyse et la spécification feront l'objet du prochain chapitre.

Chapitre 2

Spécification des besoins

Introduction

Cette partie va servir à poser les bases du recueil des besoins du système à réaliser et aussi pour pouvoir clarifier les besoins des utilisateurs de notre application. Nous allons présenter les besoins fonctionnels ainsi que les besoins non fonctionnels puis nous allons identifier les acteurs.

I. Etudes des besoins

I.1 Besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système.

- **Gestion des articles :**

Le système doit offrir le moyen de créer des Articles Sous leur catégories, famille et sous familles spécifiques à eux. Il doit aussi offrir la possibilité de visualiser toute la liste des Articles organisés.

- **Planification des visites pour les inspecteurs commerciaux :**

La solution doit offrir un mécanisme de lancement des visites qui consiste à définir une liste de clients à visiter pour chaque date donnée.

- **Gestion des clients :**

Le système doit offrir le moyen d'ajouter des clients pour chaque secteur que l'inspecteur couvre.

- **Gestion des usines :**

Le système doit offrir le moyen d'ajouter des nouvelles usines de l'entreprise.

- **Gestion des commandes :**

Le système doit offrir le moyen de gérer les commandes effectuées par les inspecteurs.

- **Gestion des inspecteurs commerciaux :**

Le système doit offrir le moyen de gérer les inspecteurs commerciaux de chaque usine.

- **Gestion des catégories :**

Le système doit offrir le moyen de gérer les catégories des articles.

- **Gestion des familles d'articles :**

Le système doit offrir le moyen de gérer les familles des articles.

- **Gestion des sous familles d'articles :**

Le système doit offrir le moyen de gérer les sous familles des articles.

I.2 Besoins non fonctionnels

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate aux attentes des concepteurs des architectures. Notre application doit nécessairement assurer ces besoins :

- **Maintenabilité et extensibilité :**

Dans le cadre de ce travail, l'application devra être extensible par rapport aux besoins du marché. Le code de l'application doit être lisible et compréhensible.

- **Intégrité des données:**

Des contrôles de saisie doivent être réalisés pour assurer des données intégrées et homogènes.

- **Ergonomie et souplesse :**

L'application doit offrir une interface Home/Machine conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles.

- **La rapidité de traitement :**

Vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements soit la plus courte possible

- **Evolution :**

Le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins du marché.

II. Diagrammes de cas d'utilisation

II.1 Définition du cas d'utilisation

Les diagrammes de cas d'utilisation sont les diagrammes fonctionnels d'UML. Ils permettent de décrire le comportement de système de point de vue utilisateur. Ils permettent aussi de structurer les besoins des utilisateurs et les objectifs de système (8).

II.2 Identification des acteurs

Un acteur représente un élément externe qui interagit avec un système dans le but de le faire fonctionner et d'en tirer profit (8).

Administrateur : c'est l'acteur principal dans notre projet c'est à dire dans la partie qui contrôle le backoffice, il s'occupe de la vérification des opérations enregistrées par le FrontOffice et le contrôle de toute les opérations effectuées.

Les tâches citées ci-dessus seront présentées avec plus de détails dans la section suivante.

II.3 Diagramme de cas d'utilisation globale

Dans le chapitre présent nous allons nous concentrer à capturer les besoins de notre projet qui consiste à réaliser la partie backoffice du projet globale. Nous avons regroupé l'ensemble de cas d'utilisation dans un seul diagramme qui est ci-dessous :

Figure 4: Diagramme de cas d'utilisation globale

Selon le diagramme figuré sur la figure 4, l'administrateur fait toutes les tâches dans la plateforme.

II.4 Raffinement de cas d'utilisation

II.4.1. Raffinement de cas d'utilisation du cas « Gestion des clients »

Nous détaillons le cas «Gestion des clients» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

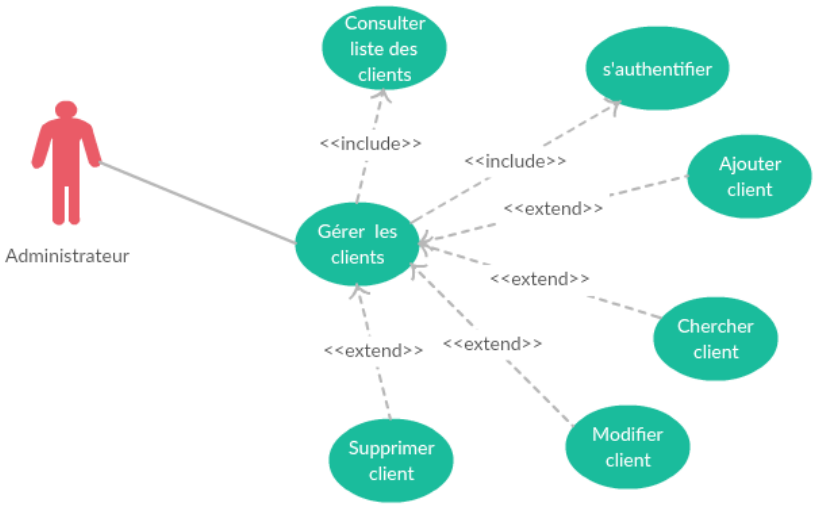


Figure 5: Raffinement de cas d'utilisation du cas « Gestion des clients »

La figure 5 représente le cas d'utilisation «Gestion des clients» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou consulter la liste des clients ou bien d'ajouter un nouveau client, le modifier ou le supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des client»

Tableau 2: Description du cas d'utilisation «Gestion des client»

Cas d'utilisation	Gestion des clients
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Client ajouté ou, Client modifié ou, Client supprimé à condition de confirmer la suppression.
Enchainement nominal :	La gestion des types ; ✓ La consultation de la liste des clients ✓ L'ajout des nouveaux clients ✓ La modification des clients existants, ✓ La suppression des clients existants avec une confirmation.

II.4.2. Raffinement cas d'utilisation du cas « Gestion des articles »

Nous détaillons le cas «Gestion des articles» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

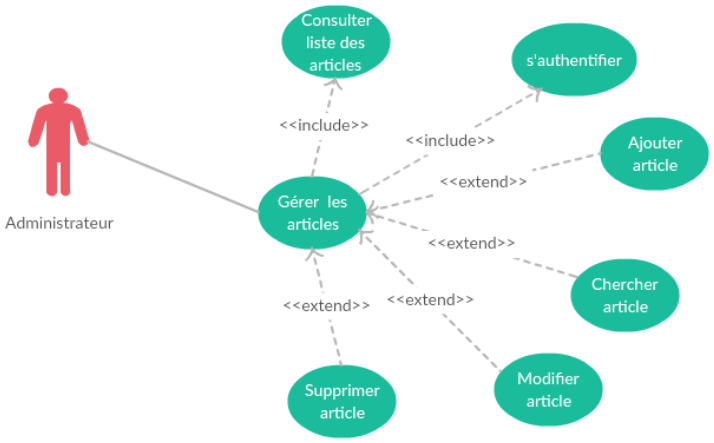


Figure 6: Raffinement cas d'utilisation du cas « Gestion des articles »

La figure 6 représente le cas d'utilisation «Gestion des articles» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou consulter la liste des articles ou bien d'ajouter un nouveau article, le modifier ou le supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des articles»

Tableau 3: Description du cas d'utilisation «Gestion des articles»

Cas d'utilisation	Gestion des articles
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Article ajouté ou, Article modifié ou, Article supprimé à condition de confirmer la suppression.
Enchainement nominal :	La gestion des types ; ✓ La consultation de la liste des articles ✓ L'ajout des nouveaux articles ✓ La modification des articles existants, ✓ La suppression des articles existants avec une confirmation..

II.4.3. Raffinement de cas d'utilisation du cas « Gestion des visites »

Nous détaillons le cas «Gestion des visites» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

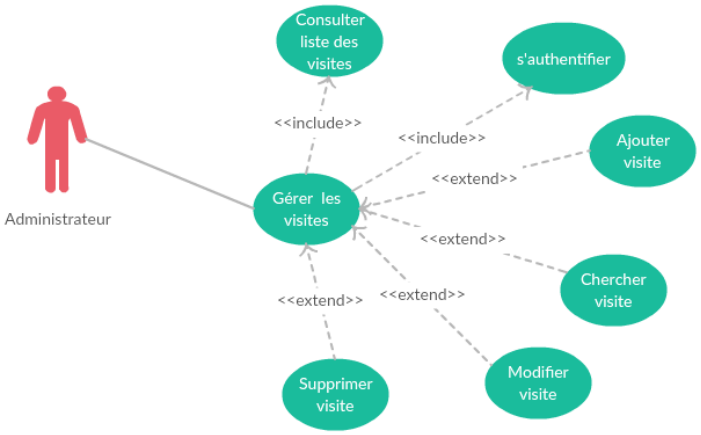


Figure 7: Raffinement de cas d'utilisation du cas « Gestion des visites »

La figure 7 représente le cas d'utilisation «Gestion des visites» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou consulter la liste des visites ou bien d'ajouter une nouvelle visite, la modifier ou la supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des visites»

Tableau 4: Description du cas d'utilisation «Gestion des visites»

Cas d'utilisation	Gestion des visites
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Visite ajouté ou, Visite modifié ou, Visite supprimé à condition de confirmer la suppression.
Enchainement nominal :	✓ La consultation de la liste des visites ✓ L'ajout des nouvelles visites ✓ La modification des visites existantes, ✓ La suppression des visites existantes avec une confirmation.

II.4.4. Raffinement de cas d'utilisation du cas « Gestion des commandes »

Nous détaillons le cas «Gestion des commandes» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

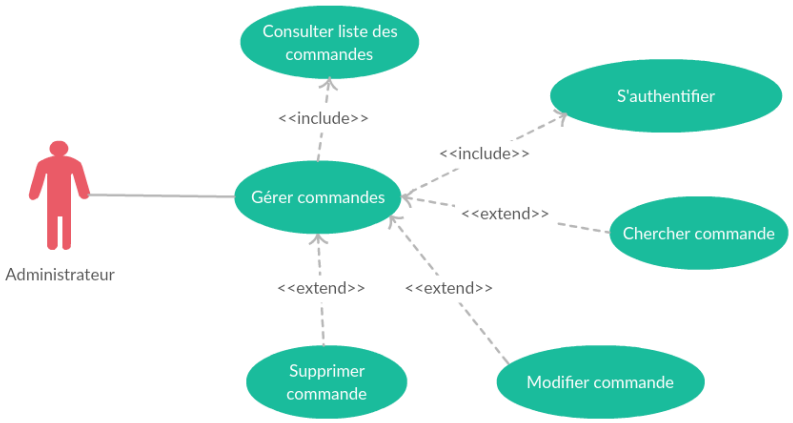


Figure 8: Raffinement de cas d'utilisation du cas « Gestion des commandes »

La figure 8 représente le cas d'utilisation «Gestion des commandes» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou consulter la liste des commandes ou modifier une commande ou la supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des commandes»

Tableau 5: Description du cas d'utilisation «Gestion des commandes»

Cas d'utilisation	Gestion des commandes
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	commande modifié ou, commande supprimé à condition de confirmer la suppression.
Enchainement nominal :	✓ La consultation de la liste des commandes ✓ La modification des commandes existantes, ✓ La suppression des commandes existantes avec une confirmation.

II.4.5. Raffinement de cas d'utilisation du cas « Gestion des inspecteurs commerciaux »

Nous détaillons le cas «Gestion des Inspecteurs commerciaux» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

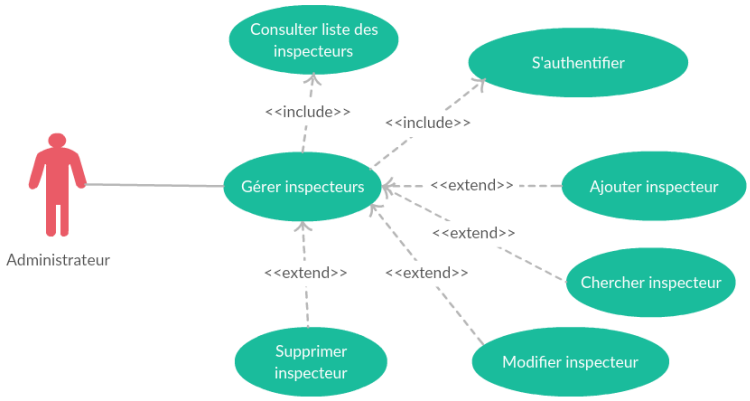


Figure 9: Raffinement de cas d'utilisation du cas « Gestion des inspecteurs commerciaux»

La figure 9 représente le cas d'utilisation «Gestion des inspecteurs commerciaux» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou bien ajouter un nouvel inspecteur, le modifier ou le supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des inspecteurs commerciaux»

Tableau 6: Description du cas d'utilisation «Gestion des inspecteurs commerciaux»

Cas d'utilisation	Gestion des inspecteurs commerciaux
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Inspecteur ajouté ou, Inspecteur modifié ou, Inspecteur supprimé à condition de confirmer la suppression.
Enchainement nominal :	✓ La consultation de la liste des Inspecteurs ✓ L'ajout des nouvelles Inspecteurs ✓ La modification des Inspecteurs, ✓ La suppression des Inspecteurs existante avec une confirmation.

II.4.6. Raffinement de cas d'utilisation du cas « Gestion des usines »

Nous détaillons le cas «Gestion des usines» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

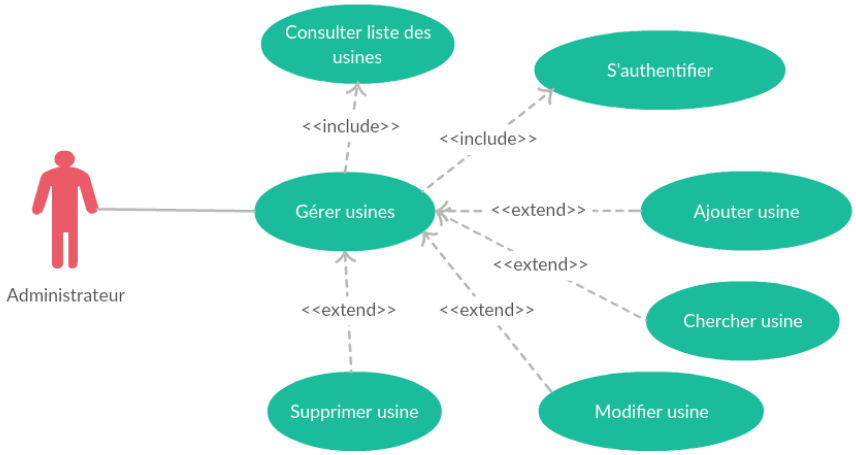


Figure 10: Raffinement de cas d'utilisation du cas « Gestion des usines»

La figure 10 représente le cas d'utilisation «Gestion des usines» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut ajouter une nouvelle usine, la modifier ou la supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des usines»

Tableau 7: Description du cas d'utilisation «Gestion des usines»

Cas d'utilisation	Gestion des usines
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Usine ajouté ou, Usine modifié ou, Usine supprimé à condition de confirmer la suppression.
Enchainement nominal :	✓ La consultation de la liste des Usines ✓ L'ajout des nouvelles Usines ✓ La modification des Usines, ✓ La suppression des Usines existantes avec une confirmation.

II.4.7. Raffinement de cas d'utilisation du cas « Gestion des catégories d'articles »

Nous détaillons le cas «Gestion des catégories d'articles» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

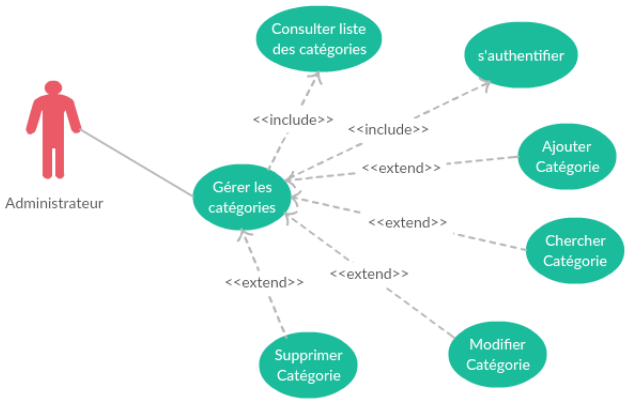


Figure 11: Raffinement de cas d'utilisation du cas « Gestion des catégories d'articles»

La figure 11 représente le cas d'utilisation «Gestion des catégories d'articles» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou bien ajouter une nouvelle catégorie, la modifier ou la supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des Catégories»

Tableau 8: Description du cas d'utilisation «Gestion des catégories d'articles»

Cas d'utilisation	Gestion des catégories d'articles
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Catégorie ajouté ou, Catégorie modifié ou, Catégorie supprimé à condition de confirmer la suppression.
Enchainement nominal :	✓ La consultation de la liste des Catégories ✓ L'ajout des nouvelles Catégories ✓ La modification des Catégories, ✓ La suppression des Catégories existantes avec une confirmation.

II.4.8. Raffinement de cas d'utilisation du cas « Gestion des familles d'articles »

Nous détaillons le cas «Gestion des familles d'articles» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

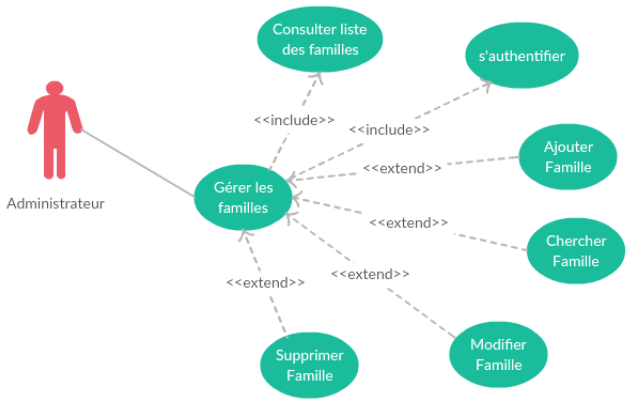


Figure 12: Raffinement de cas d'utilisation du cas « Gestion des familles d'articles»

La figure 12 représente le cas d'utilisation «Gestion des catégories d'articles» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou bien ajouter une nouvelle famille d'article, la modifier ou la supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion des familles d'articles»

Tableau 9: Description du cas d'utilisation «Gestion des familles d'articles»

Cas d'utilisation	Gestion des familles d'articles
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Famille ajouté ou, Famille modifié ou, Famille supprimé à condition de confirmer la suppression.
Enchainement nominal :	✓ La consultation de la liste des familles ✓ L'ajout des nouvelles familles ✓ La modification des familles, ✓ La suppression des familles existantes avec une confirmation.

II.4.9. Raffinement de cas d'utilisation du cas « Gestion des sous familles d'articles »

Nous détaillons le cas «Gestion des sous familles d'articles» par un sous diagramme de cas d'utilisation qui se présente dans la figure suivante:

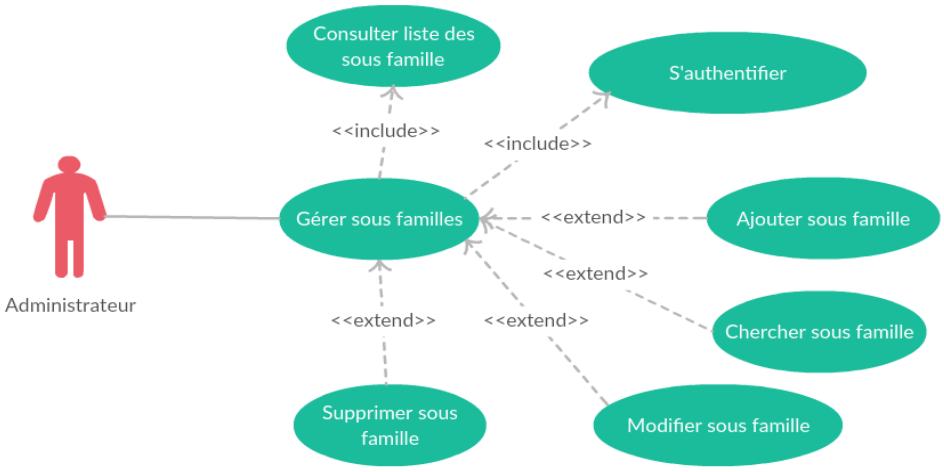


Figure 13: Raffinement de cas d'utilisation du cas « Gestion des sous familles d'articles»

La figure 13 représente le cas d'utilisation «Gestion des sous familles d'articles» raffiné. En effet, l'accès au système ne se fait qu'après une authentification. L'administrateur ainsi, peut chercher ou bien ajouter une nouvelle sous famille d'article, la modifier ou la supprimer.

Le tableau ci-dessous représente la description du cas d'utilisation «Gestion sous familles»

Tableau 10: Description du cas d'utilisation «Gestion des sous familles d'articles»

Cas d'utilisation	Gestion des sous familles d'articles
Acteur	Administrateur
Pré condition	L'Administrateur s'authentifie.
Post- condition	Sous famille ajouté ou, Sous famille modifié ou, Sous famille supprimé à condition de confirmer la suppression.
Enchainement nominal :	✓ La consultation de la liste des sous familles ✓ L'ajout des nouvelles sous familles ✓ La modification des sous familles, ✓ La suppression des sous familles existantes avec une confirmation.

III. Diagramme de Séquence système :

III.1. Diagramme de séquence système de l'opération « Planifier une visite »

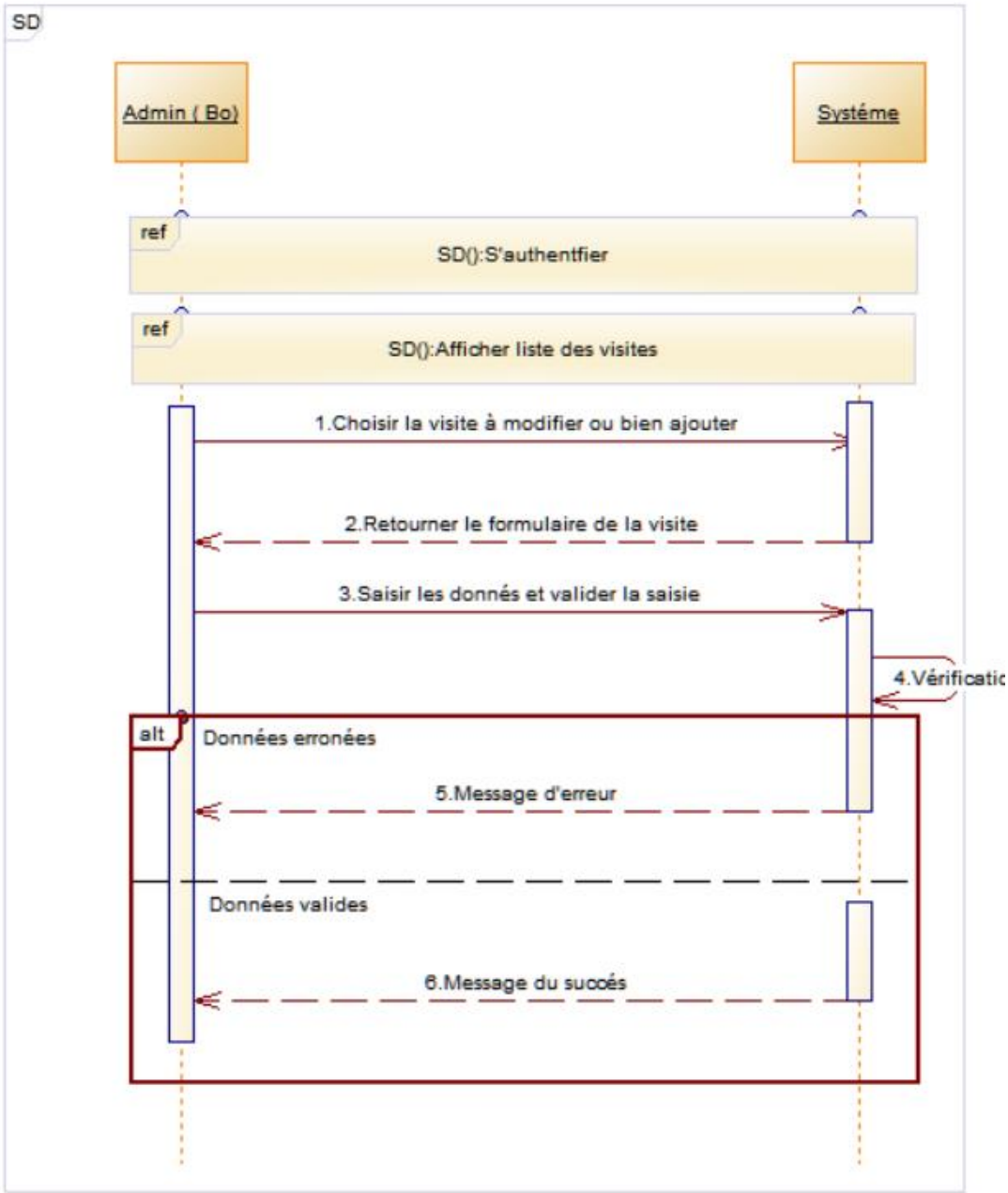


Figure 14: Diagramme de séquence système « Planifier une visite »

La figure 14 présente le diagramme de séquence système «Planifier une visite».

III.2. Diagramme de séquence système de l'opération « Modifier un article »

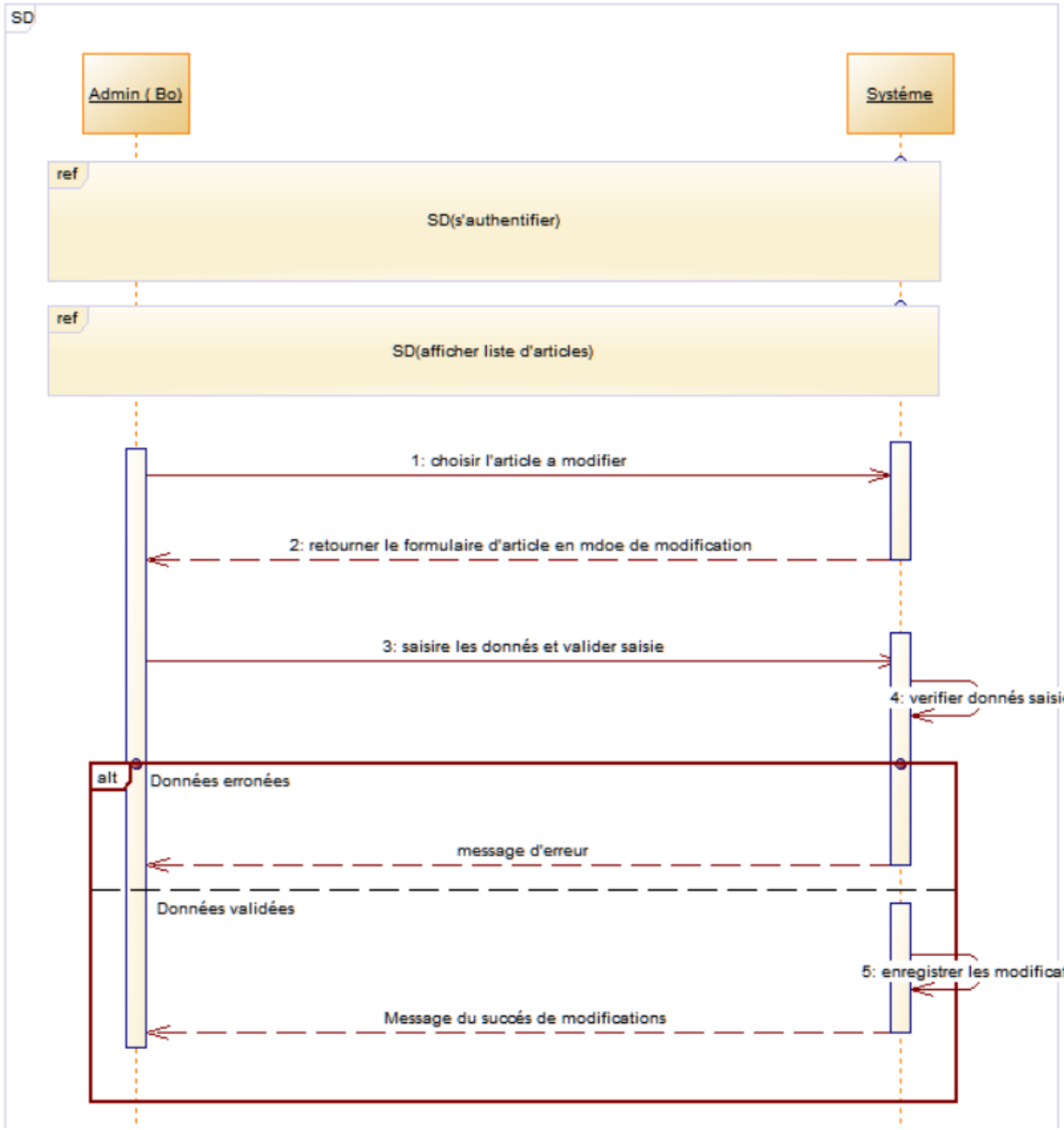


Figure 15: Diagramme de séquence système « Modifier un article »

La figure 15 présente le diagramme de séquence système «Modifier un article».

III.3. Diagramme de séquence système de l'opération « Chercher un article »

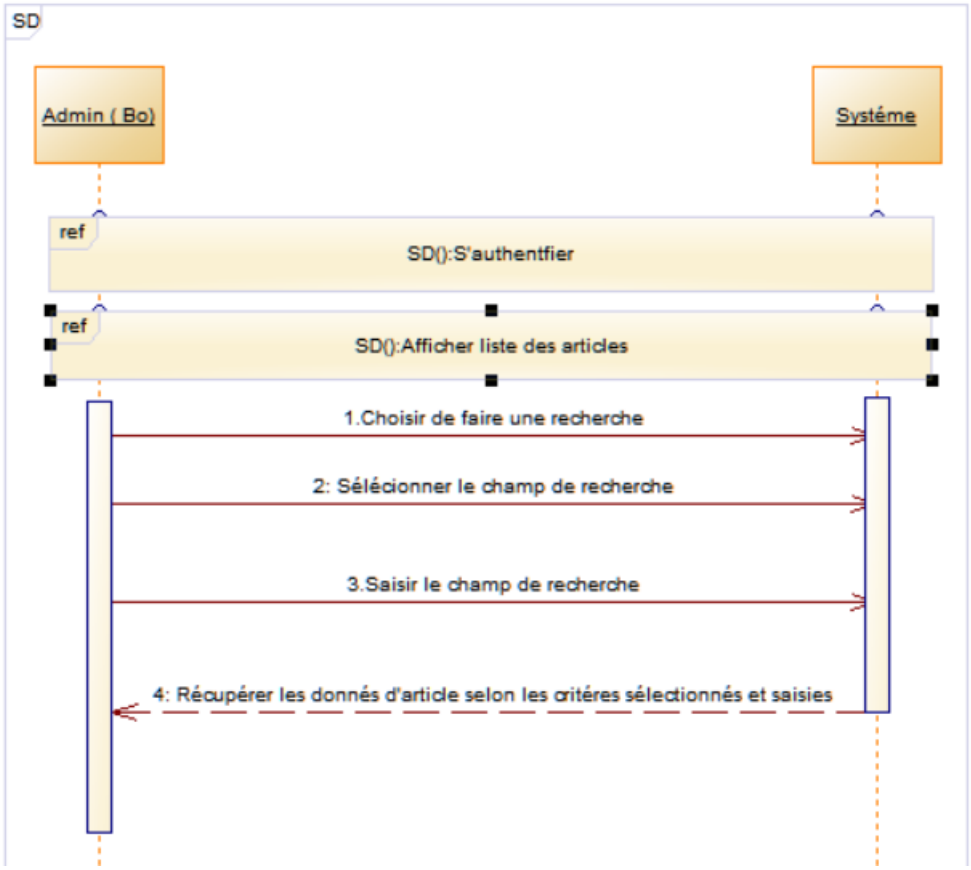


Figure 16:Diagramme de séquence système « Chercher un article »

La figure 16 présente le diagramme de séquence système «Chercher un article».

Conclusion

Dans ce chapitre, nous nous sommes concentrés, dans un premier temps, sur la détermination des besoins fonctionnels et non fonctionnels que l'application aura à satisfaire.

Dans un deuxième lieu, nous avons identifié l'acteur principal de l'application et nous avons détaillé les cas d'utilisation pour la gestion visite, articles et les clients et nous avons présenté des diagrammes de séquence système.

Dans le chapitre suivant, nous entamons la conception des systèmes.

Chapitre 3

Conception du système

Introduction

A travers cette partie, nous mettons en disposition une vue statique de notre produit représenté par le diagramme de classes et le diagramme de séquence, nous représentons par la suite le modèle objets qui présente le schéma de la base de données.

I. Spécification du style d'architecture

I.1 Organisation du modèle de spécification logicielle

Il est indispensable de penser à l'architecture du système dès le départ du projet, car c'est un élément maitre qui accompagne un travail de la phase de l'analyse des besoins jusqu'à sa réalisation. Son avantage principal est qu'il permet de garantir la qualité du produit dès le départ.

Dans le modèle UML, les couches logicielles correspondent à des packages. Pour préciser leur spécificité, nous avons introduit le stéréotype « layer ». Chaque couche produisant des services pour les niveaux supérieurs contient en conséquence des cas d'utilisation pilotés par les couches exploitantes. Les couches organisent les dépendances qui s'établissent entre elles.

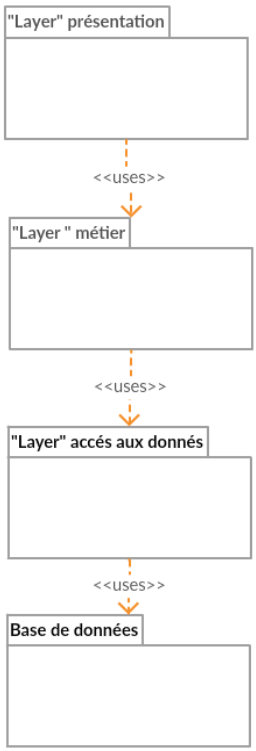


Figure 17: Organisation du modèle de spécication logicielle (diagramme de packages)

Nous constatons également dans la figure 17 ci-dessus que chaque couche de l'architecture représente un rôle précis. Ces rôles sont présentés comme suit :

- La couche présentation des données : correspondant à l'affichage, la restitution sur le poste de travail et le dialogue avec l'utilisateur.
- La couche métier des données : qui sert comme intermédiaire pour l'échange de données entre la couche de présentation et la couche accès aux données.
- La couche accès aux données: il gère la persistance des données dans la base de données (9).

La figure suivante représente l'architecture au niveau de notre application :

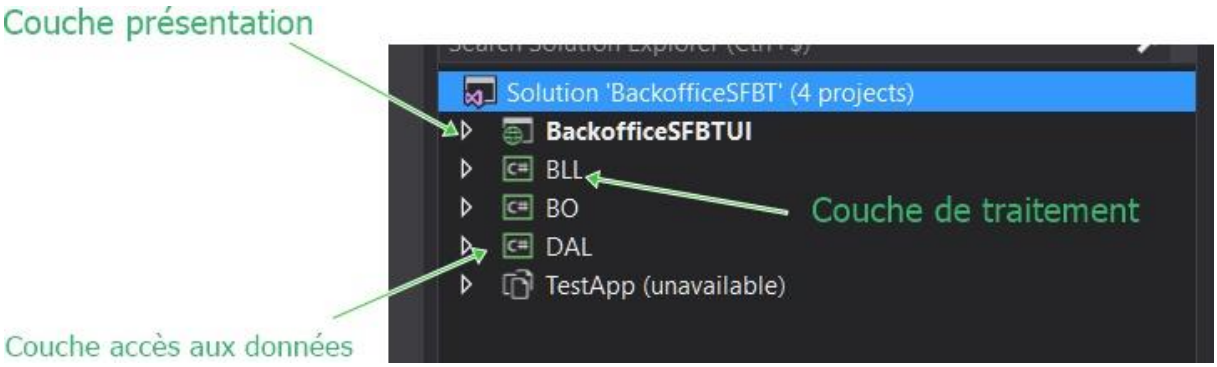


Figure 18: Architecture de l'application

L'avantage de ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur lors d'une maintenance ou une amélioration sur le projet. Cette flexibilité permet à une entreprise d'envisager une grande souplesse pour l'introduction de toutes nouvelles technologies.

I.2 Architecture physique de l'application

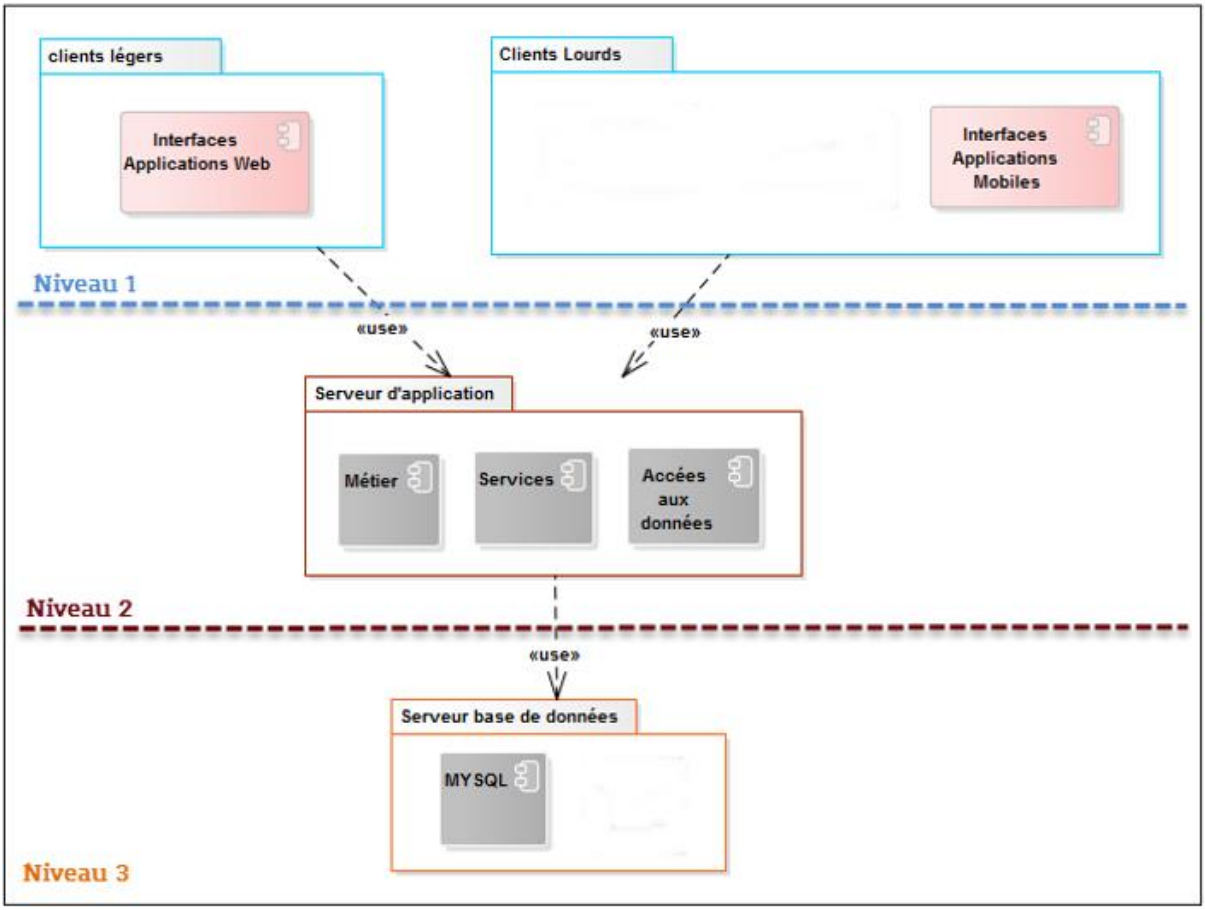


Figure 19 : Diagramme de composants du système

D'après la figure 19, nous remarquons que la spécification d'une architecture à composants métier 3-tiers implique la répartition des composants d'exploitation suivant les responsabilités. Cette architecture représente trois niveaux :

- Le niveau 1 : correspondant à la présentation des données.
- Le niveau 2 : représente le traitement métier des données.
- Le niveau 3 : correspondant à l'accès aux données persistantes.

II. Diagrammes de séquence et de classes

II.1. Définition du diagramme de séquence

La description de la vue dynamique de notre application est réalisée à travers les différents diagrammes de séquences. En effet, un diagramme de séquence décrit les interactions entre les objets du système. Nous devons, donc, utiliser les différents composants de l'architecture MVC pour mieux éclaircir les tâches, pour cela :

- L'administrateur émet une requête.
- Le contrôleur intercepte la requête de l'administrateur.
- Le contrôleur détermine quelle partie du modèle est concernée et quelle vue y est associée.
- Le modèle traite les interactions avec les données, applique les règles métier et renvoie les données au contrôleur.
- Le contrôleur sélectionne la vue et lui renseigne les données.
- La vue présente les données à l'utilisateur. (10)

II.2. Définition du diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orienté objet. Il s'agit d'une vue statique du fait qu'on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classe permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

II.3. Diagrammes des séquences détaillés

II.3.1. Diagramme de séquence détaillé « Modifier un article »

Le diagramme suivant présente le diagramme de séquence pour le cas détaillé précédemment :

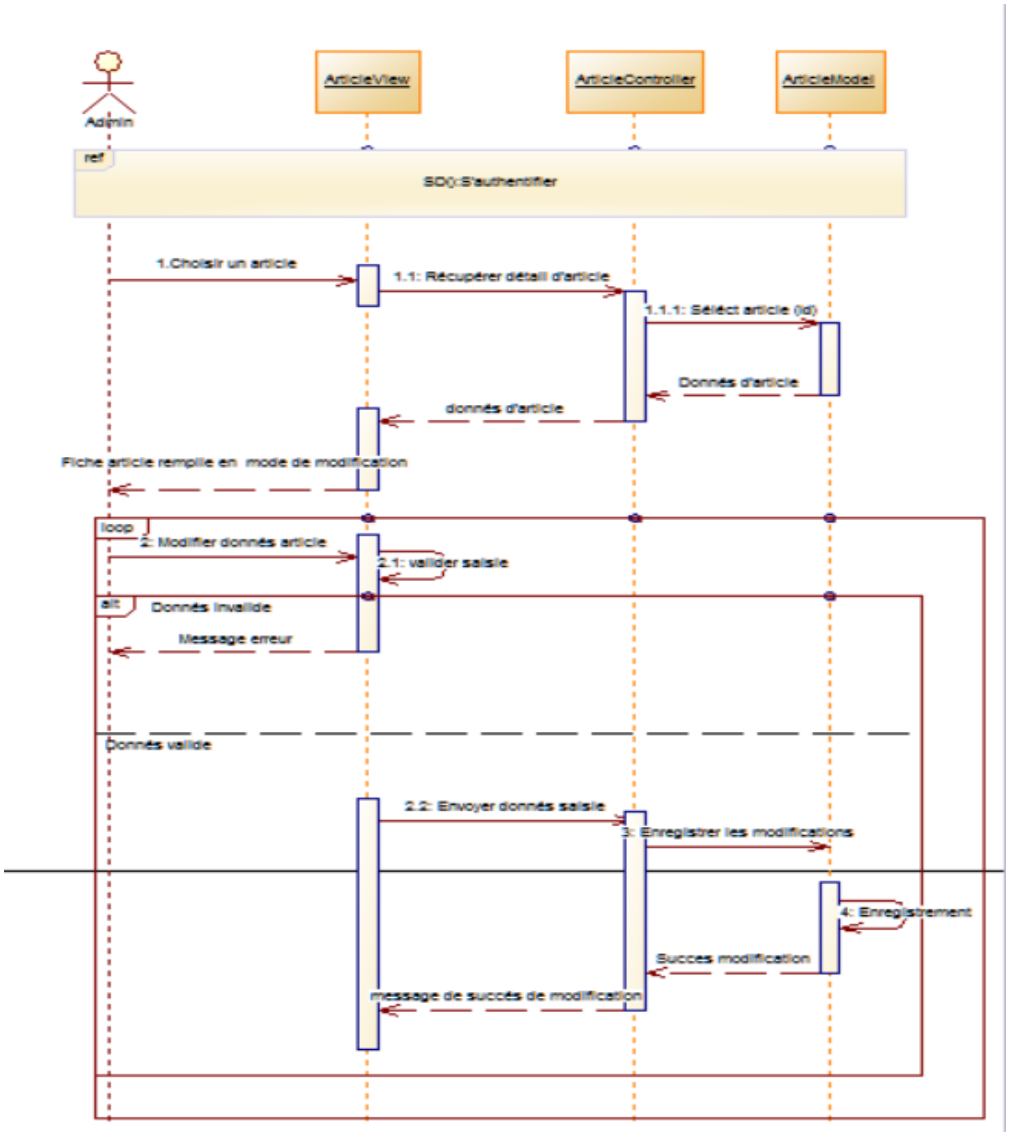


Figure 20: Diagramme de séquence détaillé « Modifier un article »

La figure 20 indique lorsque, l'administrateur choisit un article à modifier à partir de la liste affichée. Le contrôleur s'occupe de récupérer la fiche contenant les données relatives à l'article choisi en mode modification. L'utilisateur modifie les champs et confirme la saisie. Une vérification de la saisie sera faite au niveau de la vue. Le contrôleur récupère les données modifiées et se contente de mettre à jour l'entité Article et renvoie le résultat de la modification.

II.3.2. Diagramme de séquence Détaillé « Ajouter un client »

Le diagramme suivant présente le diagramme de séquence pour le cas détaillé précédemment :

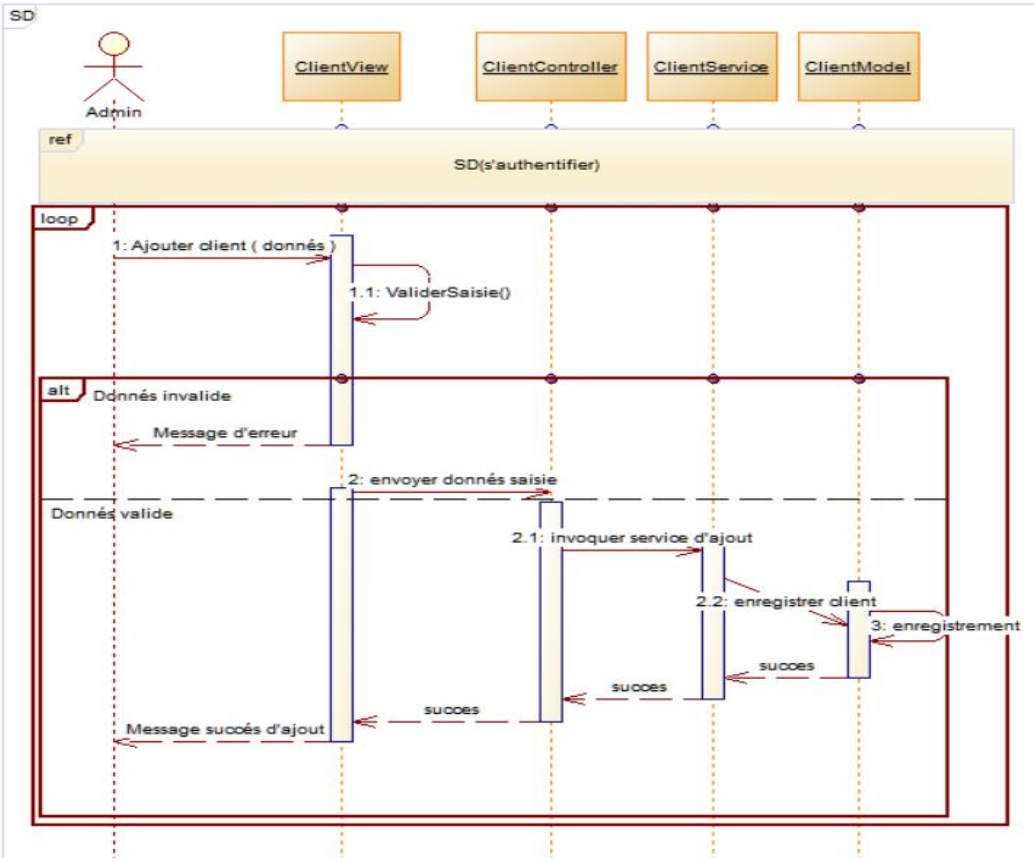


Figure 21: Diagramme de séquence détaillé « Ajouter un client »

En suivant l'enchaînement illustré sur la figure 21, l'administrateur remplit le formulaire d'ajout disponible via la vue « ClientView ».

Lorsque l'administrateur envoie les données, un contrôle de la saisie se fait au niveau de la vue pour vérifier la validité des données transmises. Si les données sont valides et complètes, elles seront transférées vers le contrôleur « ClientController », qui s'occupera de l'invocation du service « InsertClient ». Ce service va se contenter de l'ajout d'un client dans l'entité Client. Ensuite, il renvoie au contrôleur le résultat de l'opération, qui à son tour le renvoie à la vue.

Si les données sont invalides ou incomplètes, un message indiquant de vérifier les données saisies s'affichera à l'administrateur.

II.3.3. Diagramme de séquence Détaillé « Chercher un article »

Le diagramme suivant présente le diagramme de séquence pour le cas détaillé précédemment :

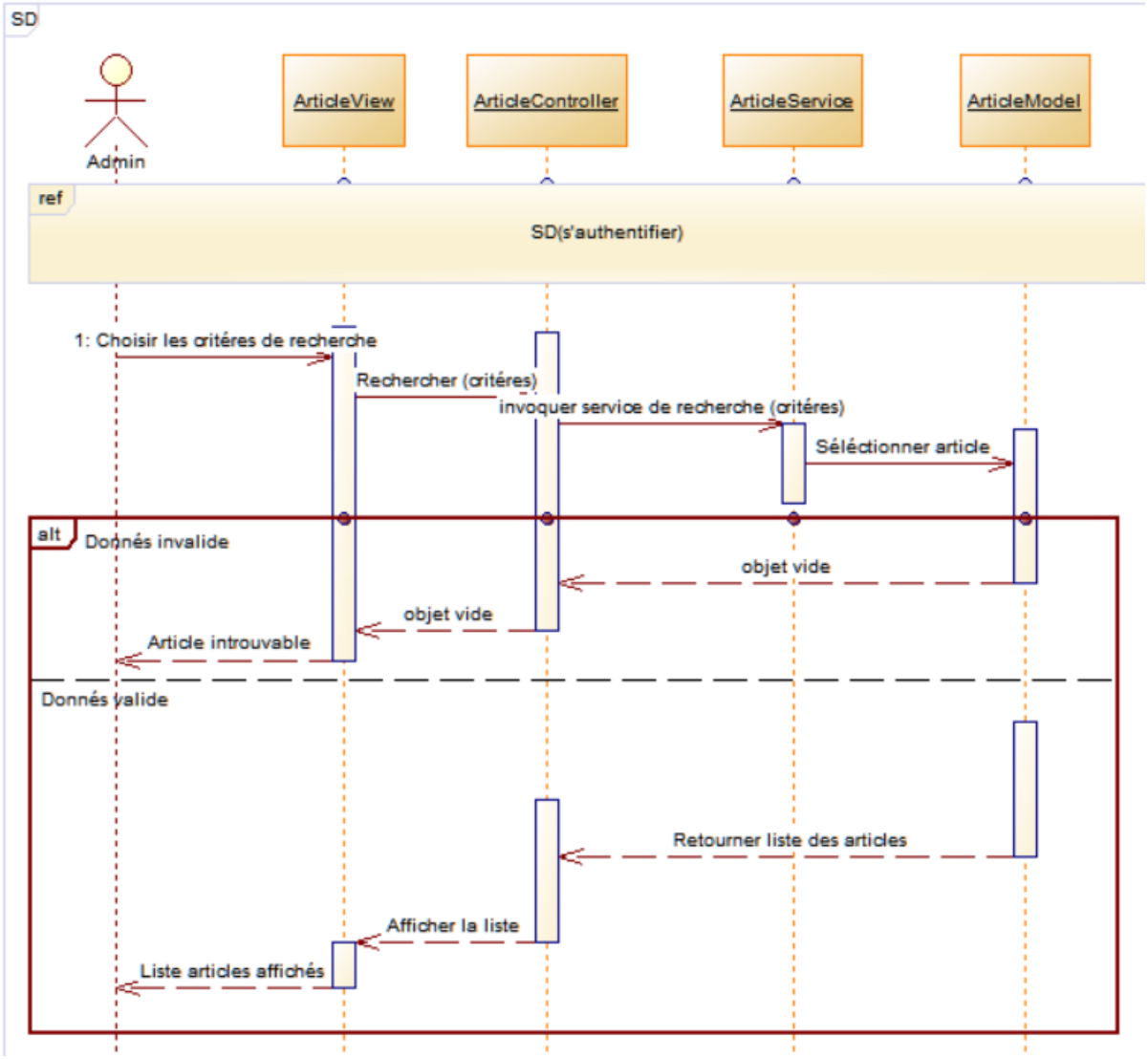


Figure 22: Diagramme de séquence détaillé « Chercher un article »

Le diagramme figuré sur l'image 22 explique l'enchaînement de la cherche d'un article. À cet effet, l'administrateur remplit le champ de recherche qui sera affiché par la vue « ArticleView». L'administrateur remplit les critères de recherche et transmet le formulaire et il envoie les données en cliquant sur le bouton « search ».Puis le contrôleur « ArticleController » récupère les données saisies et cherche l'utilisateur existant dans l'entité « ArticleModel », et répondant aux critères sélectionnés. Le résultat de la recherche sera affiché sur la vue « ArticleView». Si aucun résultat n'est trouvé, le système affiche une liste vide avec message d'erreur.

II.3.4. Diagramme de séquence Détaillé « Supprimer un article »

Le diagramme suivant présente le diagramme de séquence pour le cas détaillé précédemment :

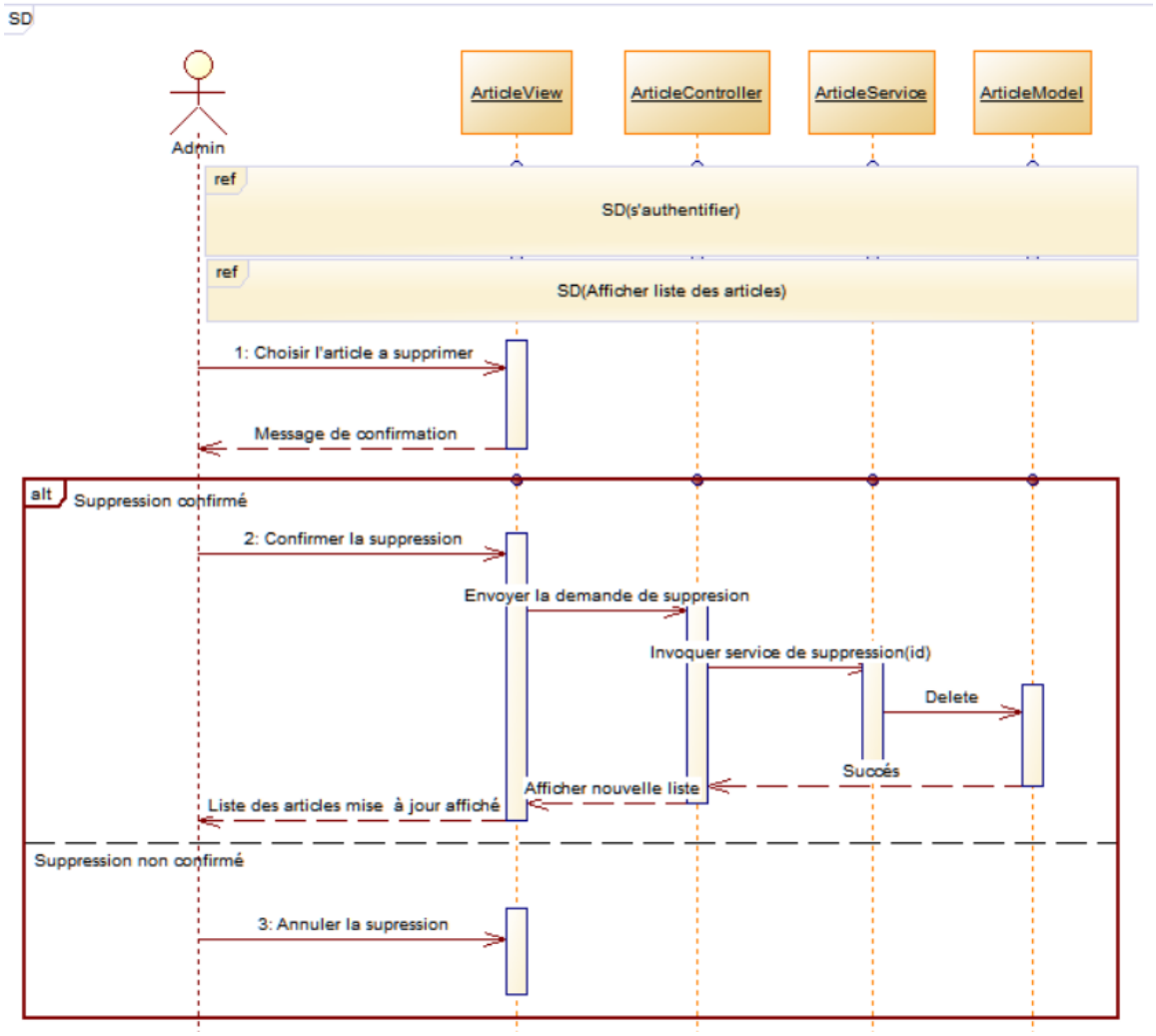


Figure 23:Diagramme de séquence Détaillé « Supprimer un article »

Comme le montre la figure 23, si l'administrateur souhaite supprimer un article, il choisit une à partir de la liste des articles affichée via la vue « ArticleView ». Par la suite, la vue « ArticleView » demande une confirmation auprès de l'administrateur. S'il valide son choix, la classe de contrôle « ArticleController » sera chargée de supprimer l'utilisateur à partir de l'entité « ArticleModel ». Dans le cas où l'administrateur annule la confirmation de la suppression du compte, il sera directement redirigé vers la vue « ArticleView ».

II.4. Conception du niveau de données

II.4.1. Diagramme de classe

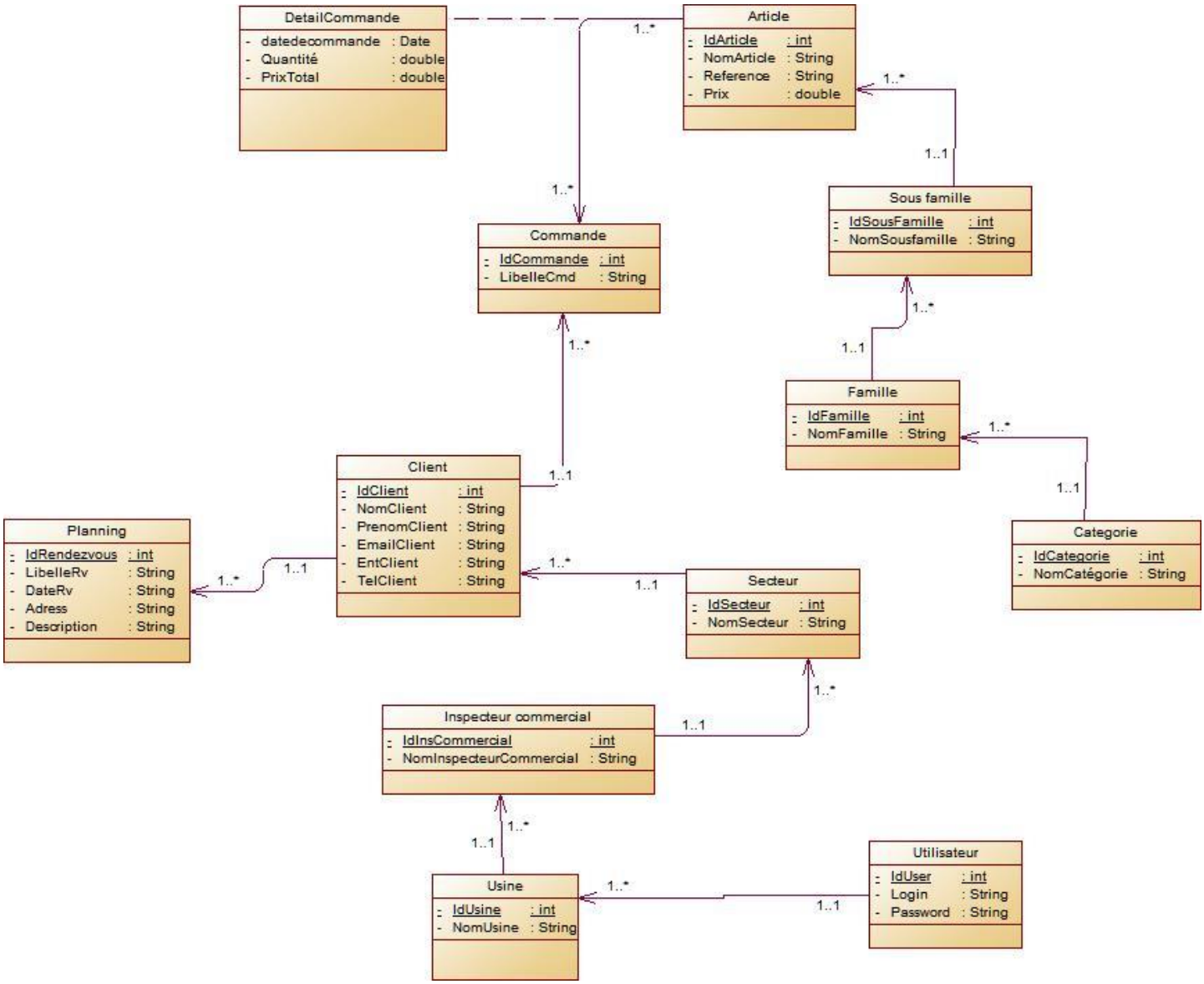


Figure 24: Diagramme de classe globale

Dans la figure 24, nous avons le diagramme de classe globale qui présente le modèle objet qui montre le schéma de notre base de données.

II.4.2. Schéma relationnel

- **Catégorie**(IdCategorie , NomCategorie) ;
- **Famille**(IdFamille , NomFamille,#IdCategorie) ;
- **SousFamill**e(IdSousfamille , NomSousfamille,#IdFamille) ;
- **Article**(IdArticle , NomArticle , Reference , Prix,#IdSousfamille) ;
- **Usine**(IdUsine , NomUsine) ;
- **InspecteurCommercial**(IdInsCommercial , NomInspecteurCommercial, #IdUsine)
- **Secteur**(IdSecteur , NomSecteur,#IdInsCommercial)
- **Client** (IdClient , NomClient , PrénomClient, EmailClient , EntClient, TelClient, #IdSecteur)
- **Commande**(IdCommande , LibelleCmd , Prix,#IdInsCommercial)
- **DetailCommande**(#IdArticle#Idcommande, Datecommande , Quantité,PrixTotal)
- **Planning**(IdRendezvous , LibelleRv , DateRv , Adresse , Description, #IdInsCommercial, #IdClient)
- **User**(IdUser , login , Password)

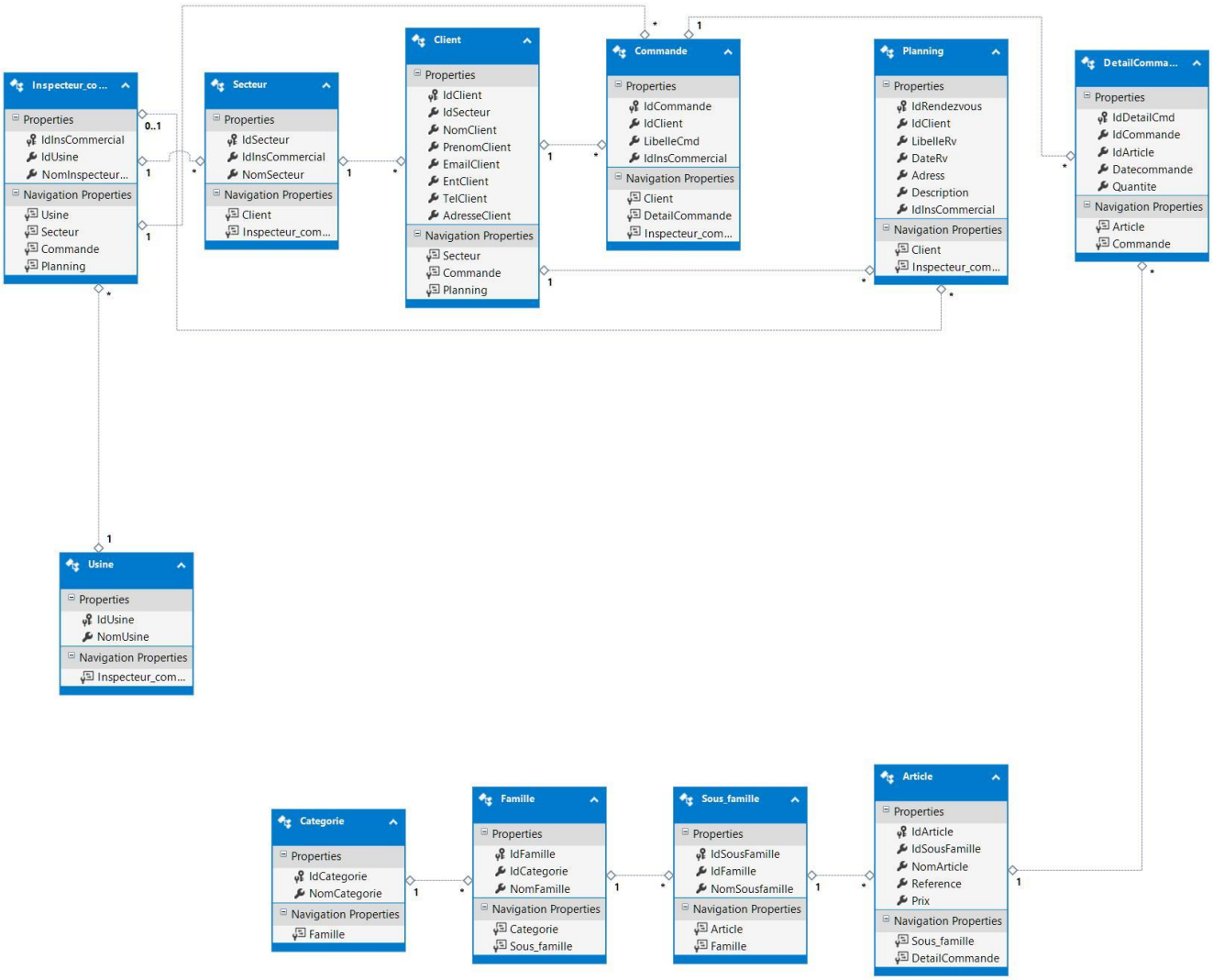


Figure 25: Modèle de données des entités

Conclusion

Ce chapitre nous a permis de définir les fonctionnalités de notre application, en passant par la phase de la conception pour assurer une bonne mise en œuvre d'un système fonctionnel répondant aux besoins utilisateurs afin de délimiter le cadre de notre travail et de préparer un terrain favorable pour la prochaine étape.

Maintenant, notre application est prête à être codée. Dans le chapitre suivant, nous allons nous intéresser à l'implémentation de notre système en se basant sur la conception détaillée de ce chapitre.

Chapitre 4

Réalisation

Introduction

Dans le chapitre précédent nous avons conçu notre système en préparant les diagrammes des cas d'utilisation et de séquences aussi le diagramme de classes.

Nous allons commencer par la description de l'environnement matériel et logiciel utilisé tout en donnant par la suite un aperçu sur notre application.

Dans cette réalisation, nous sommes passés par plusieurs jalons jusqu'à aboutir au produit final.

I. Environnement de développement

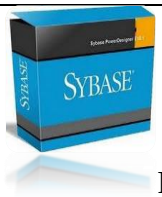
Chaque logiciel nécessite, lors de sa création et de son exécution, un appui matériel et un autre logiciel afin de réaliser les fonctionnalités pour lesquelles il a été conçu.

I.1 Environnement matériel

Ce projet a été développé sur un ordinateur portant la configuration suivante :

- Un ordinateur portable Intel i5,
- Disque dur 1 T,
- Mémoire vive 8 GO,
- Système d'exploitation Windows 10.

I.2 Environnement Logiciel

Conception	 Power AMC est un logiciel de modélisation et de gestion de métadonnées à la pointe de l'innovation, destinée aux architectures de données, aux architectures d'informations et aux architectures d'entreprise.

Implémentation	Visual Studio 2015 est une suite d'outils de développement basés sur des composants et des technologies destinés à la création d'applications puissantes et hautement performantes. En outre, Visual Studio est optimisé pour la conception, le développement et le déploiement en équipe de solutions d'entreprise. SQL Server 2012 est un système de gestion de bases de données relationnelles. Le stockage, la manipulation et l'analyse de ces données se font au sein de son moteur de bases de données. Ce service permet la réalisation de nombreuses applications, requêtes, et transactions, notamment grâce au langage T-SQL (Transact-SQL).

Le choix d'un environnement de développement, constitue une étape critique du projet puisque ce choix entraînera ou non des gains en temps et en effort, donc en coût de développement.

Nous avons aussi eu recours à des outils de développement à savoir ;

- ASP.NET est un Framework permettant de générer à la demande des pages web et utilisée pour mettre en œuvre des applications web. ASP.NET peut être utilisé avec n'importe quel langage de programmation pour la plateforme .NET (Visual Basic .NET, C#, JScript4…).
- EntityFramework est la solution de Mapping Objet-Relationnel proposée par Microsoft. Son but est de fournir la couche d'abstraction nécessaire aux développeurs pour qu'ils n'accèdent plus directement à la base de données

II. Principales interfaces graphiques

L'ergonomie, la simplicité et l'interactivité des interfaces graphiques sont des atouts consistants et primordiaux pour toute application informatique de qualité. En fait, on est souvent face à des cas où un non informaticien se lasse d'apprendre l'utilisation d'un logiciel à cause de la complexité des interfaces graphiques. Nous avons donc veillé à assurer le confort absolu de l'utilisateur en créant des interfaces graphiques ergonomiques, simples et non encombrées. En voici quelques exemples.

II.1. Interface d'authentification

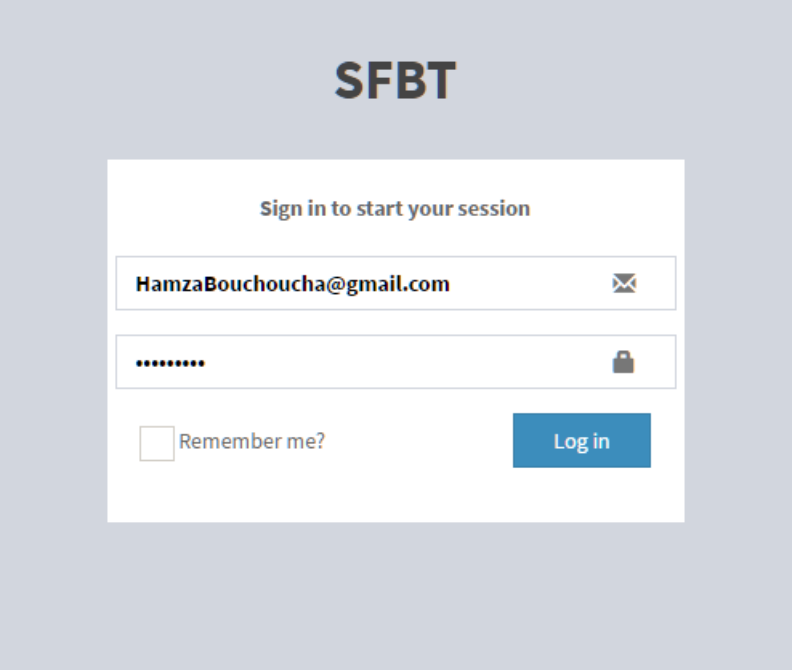


Figure 26 : Page d'authentification

La figure 26 représente l'interface d'authentification dans laquelle on doit choisir le nom d'administrateur et entrer le mot de passe pour commencer à utiliser le backoffice. A travers cette fenêtre l'administrateur s'authentifie pour utiliser l'application. Cette étape met en valeur l'aspect sécurité : nous vérifions la disponibilité du compte utilisateur et nous lui attribuons les droits et privilèges nécessaires.

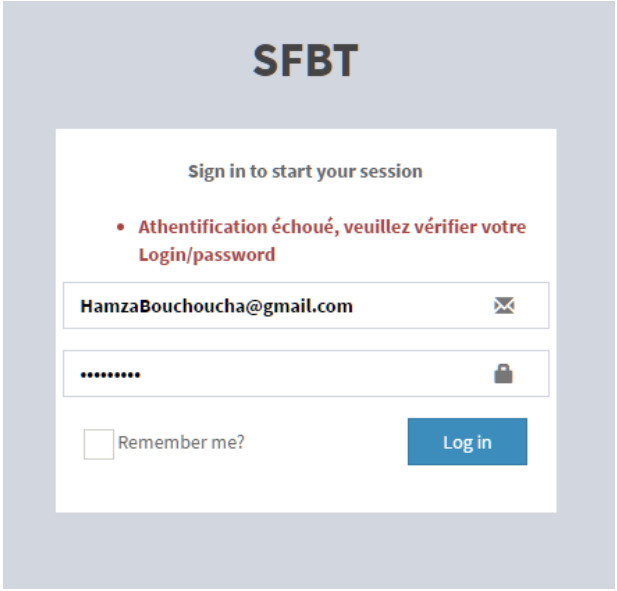


Figure 27 : Interface d'erreur lors de la saisie des mots de passe

Ce message dans la figure 27 s'affiche lorsqu'un utilisateur se trompe sur son mot de passe. Donc il doit entrer une autre fois le correct mot de passe pour pouvoir accéder à ses fonctionnalités.

II.2. Interface d'accueil

Figure 28:Interface d'accueil

La figure 28 représente l'interface pilote de notre backoffice après le succès de l'authentification. Elle permet d'accéder aux différentes fonctionnalités offertes par l'application (Liste des articles, des clients, des commandes, etc.).

II.3. Interface liste des clients

Dans le menu, quand l'administrateur choisit la rubrique liste des clients, une liste sera affichée.

Il aura la possibilité de filtrer la liste selon des mots clés.

Dans cette interface, l'administrateur peut :

- Cliquer sur le bouton d'ajouter un client pour aller à la page d'ajout
- Choisir un critère pour effectuer une recherche
- Organiser les données de la liste par ordre alphabétique.
- Cliquer sur le bouton ' Supprimer ' Pour effacer un client après la confirmation.
- Cliquer sur le bouton d'ajouter un client pour aller à la page de modification.

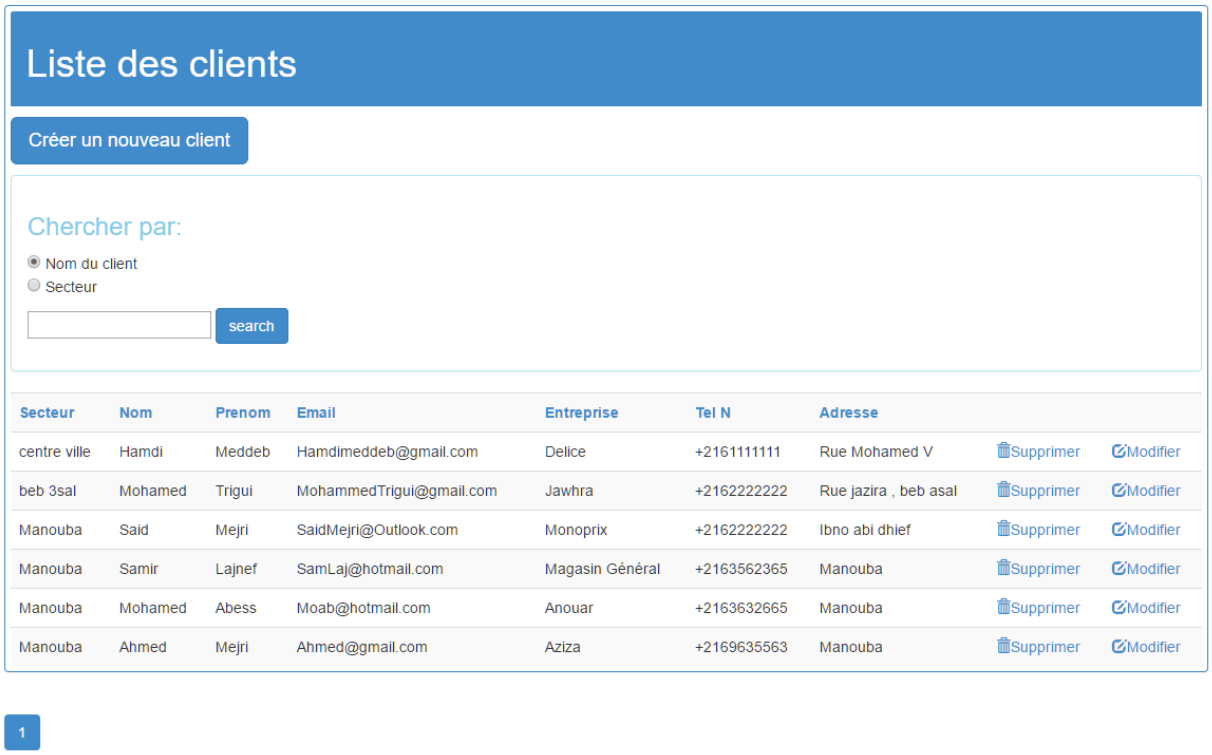


Figure 29: Liste des clients

La Figure 29 illustre une capture de la liste des clients de notre système.

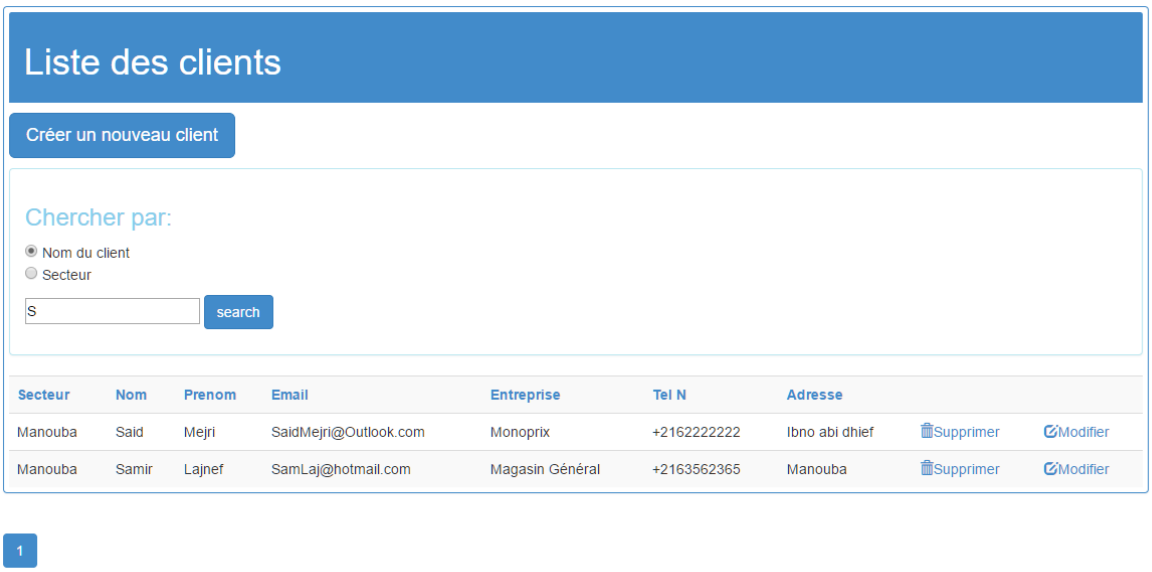


Figure 30: Chercher un client Selon la première lettre du nom de client

Dans la Figure 30 nous avons l'interface de la recherche des clients qui permet à l'administrateur de saisir le nom de client dans le champ « search ». Puis, il obtient la liste des clients dont les noms commencent par le critère de recherche.

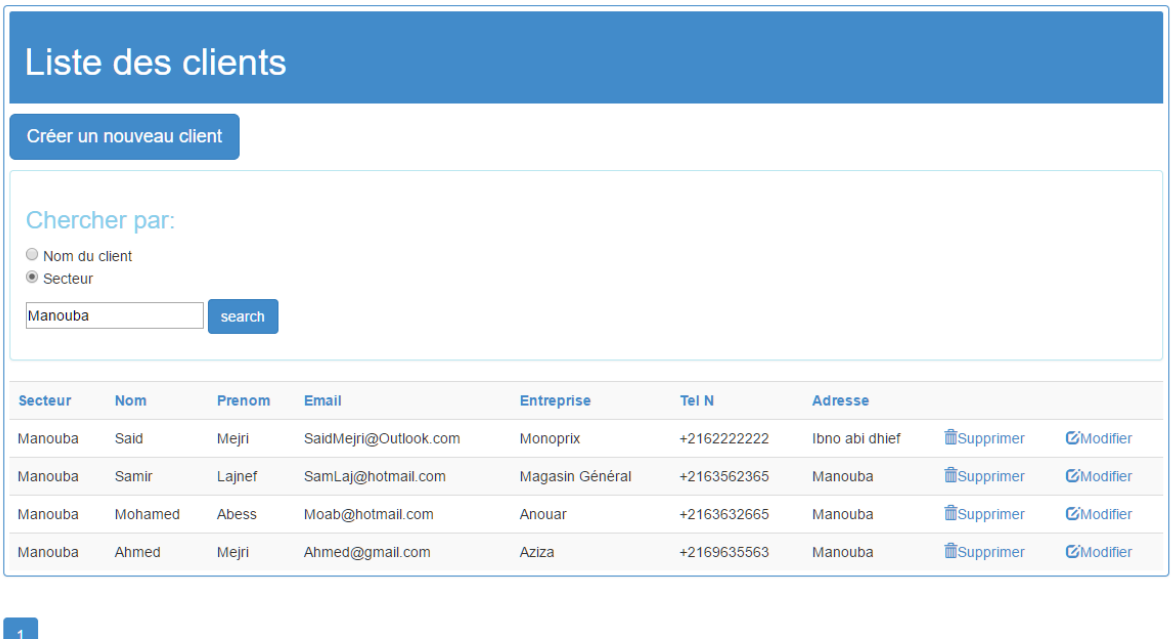


Figure 31:Chercher un client selon son secteur

Dans la Figure 31 nous avons l'interface de recherche des clients qui permet à l'administrateur de saisir le secteur de client dans le champ « search ». Puis, il obtient la liste des clients dont leurs secteurs commencent par le critère de recherche.

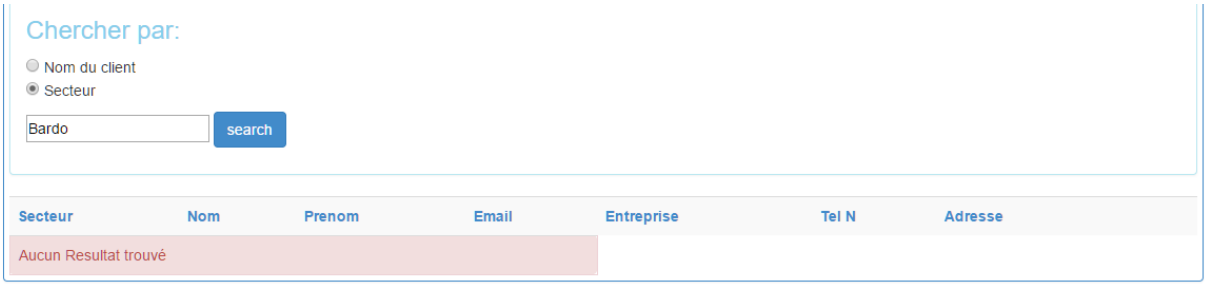


Figure 32:Echec d'un Recherche

La Figure 32 illustre le message du résultat de recherche si aucun résultat n'est trouvé.

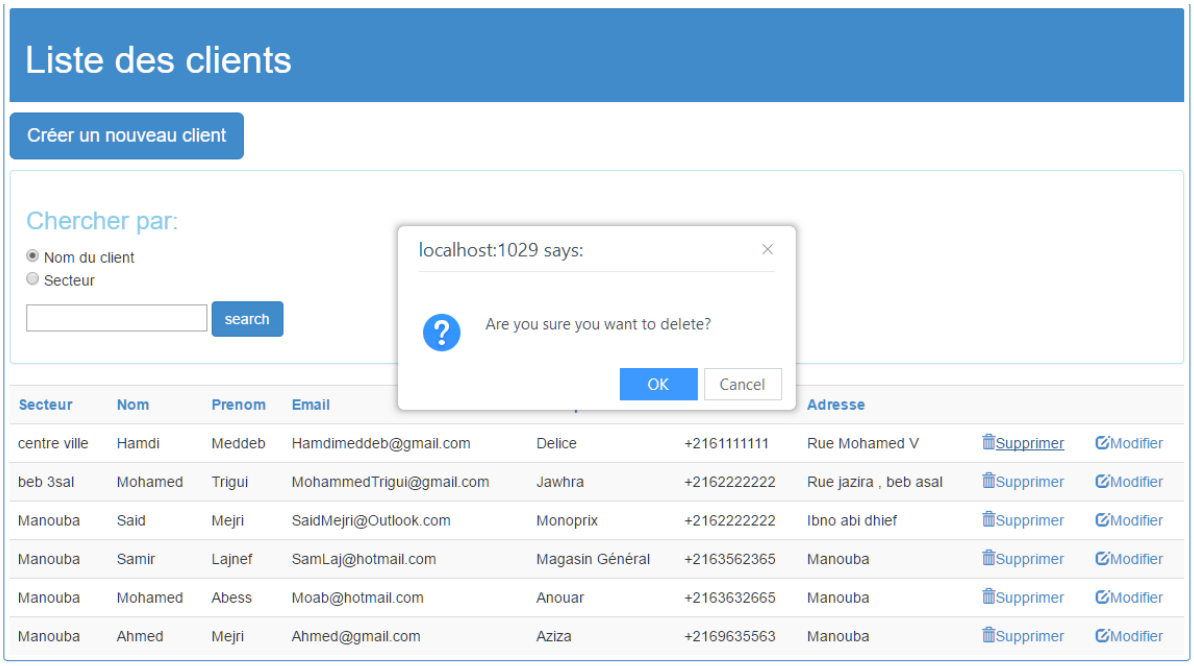


Figure 33:Message de confirmation de suppression

La figure 33 illustre le message de confirmation de la suppression de client.

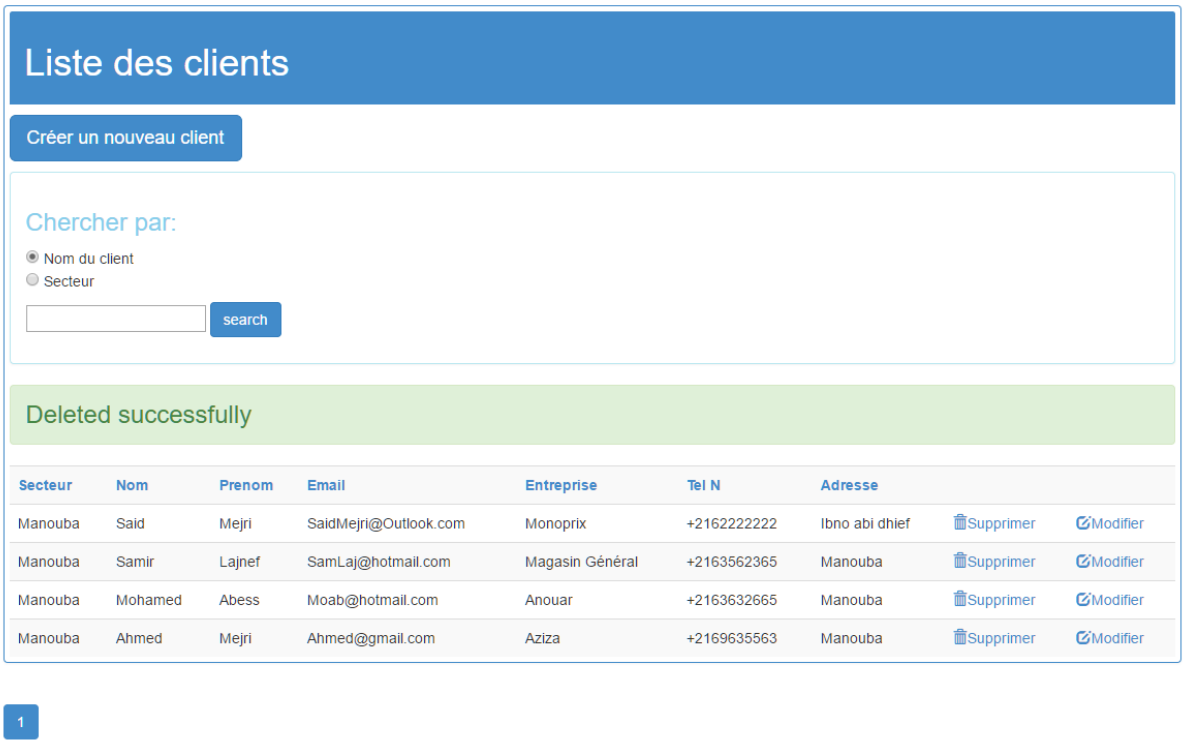


Figure 34: La nouvelle liste après suppression d'un client

La Figure 34 illustre le message du résultat de suppression et l'affichage de la nouvelle liste des clients.

II.4. Interface d'ajout d'un client

Un message de succès apparait après l'ajout et le client sera ajouté dans la liste.

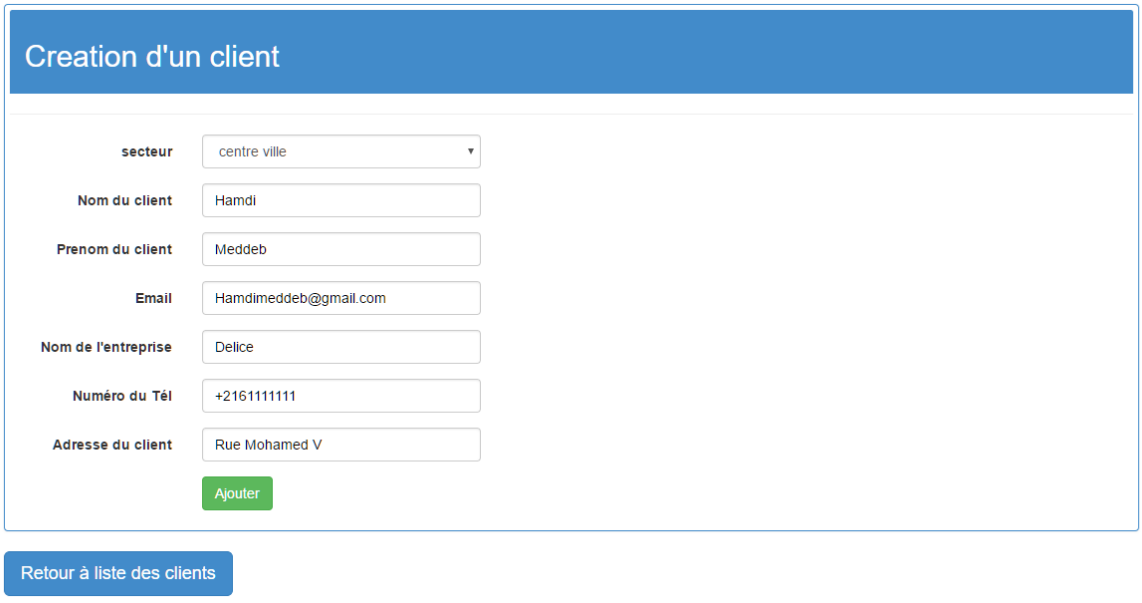


Figure 35:Interface d'ajout d'un client

La figure 35 représente l'interface permettant à l'administrateur d'ajouter un nouveau Client.

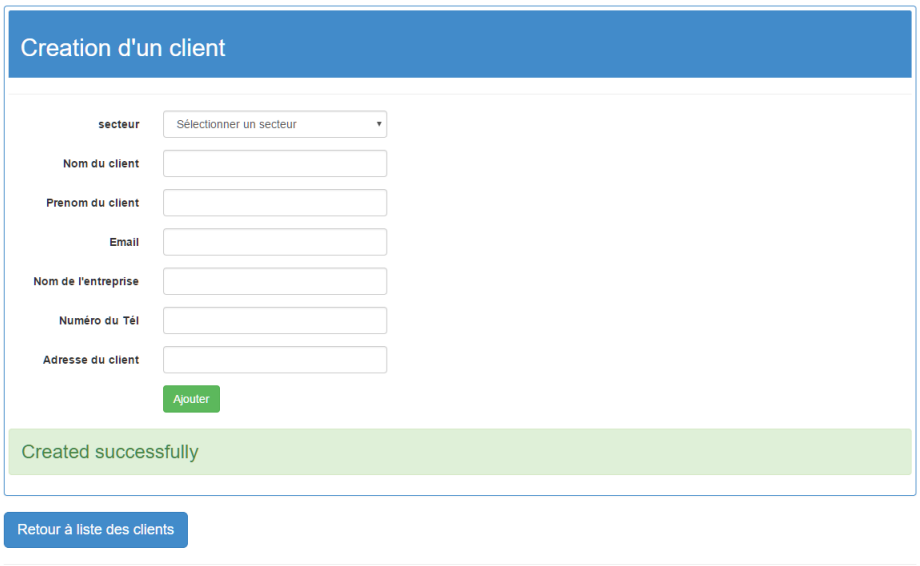


Figure 36:Message de succès d'ajout

La figure 36 illustre le message du succès de l'ajout d'un nouveau client.

II.5. Interface de modification d'un client

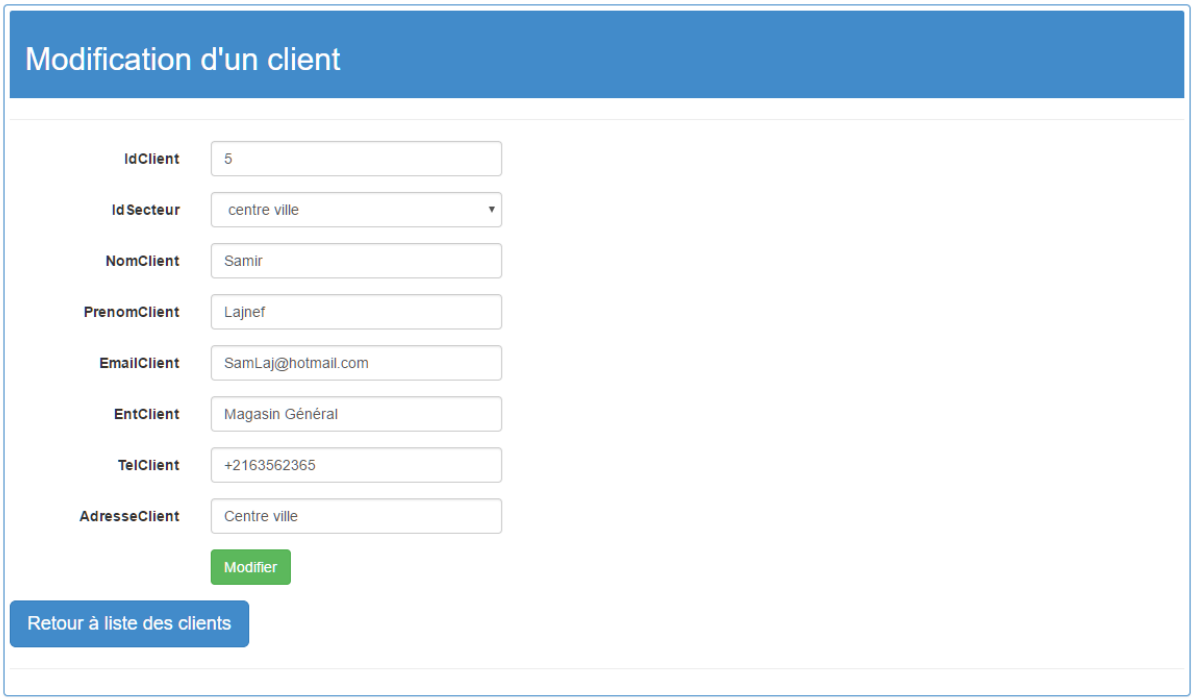


Figure 37: Interface de modification d'un client

Dans la figure 37 nous remarquons qu'une fois l'utilisateur choisit le point relais à modifier, un formulaire contient les données s'affiche en mode modification. Il modifie les champs souhaités puis il valide.

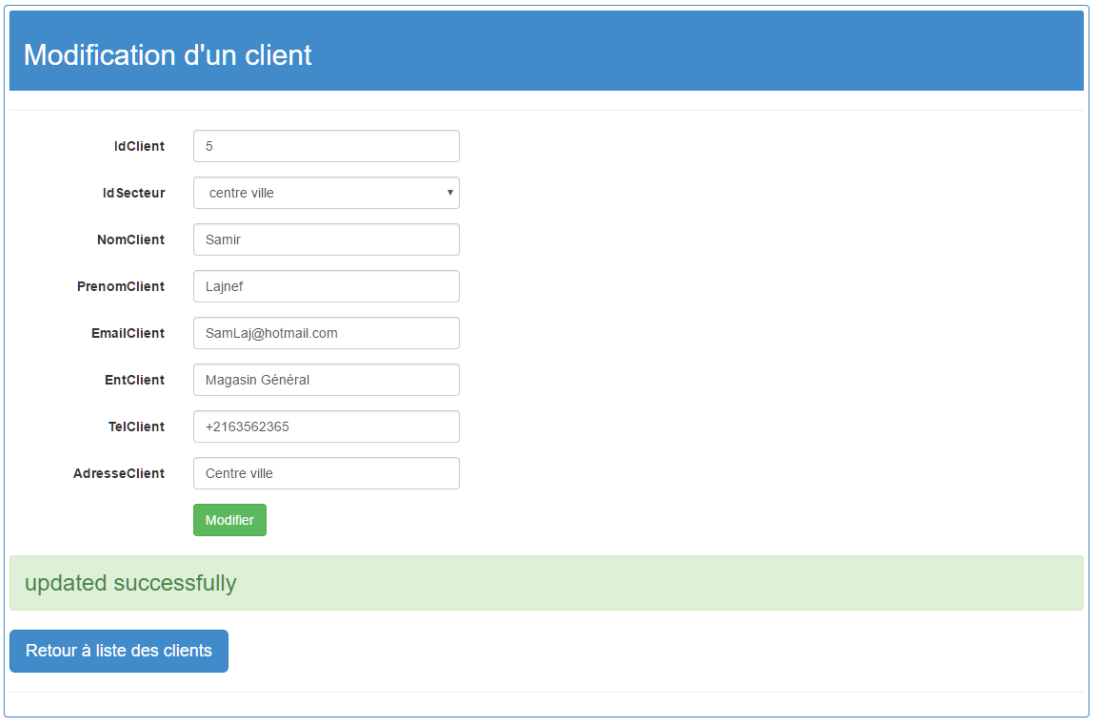
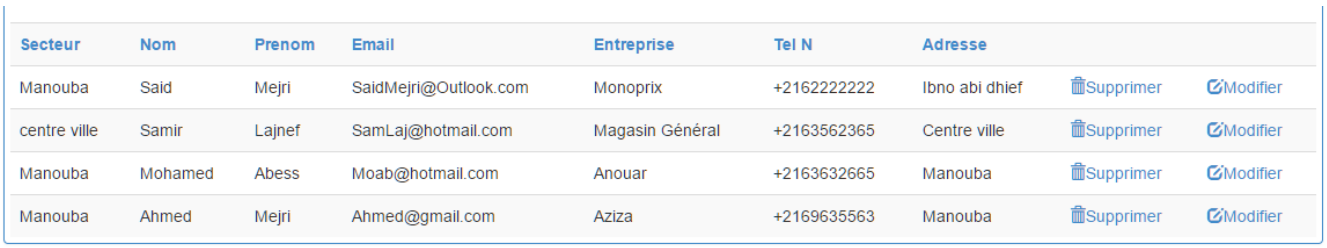


Figure 38: Succès après modification d'un client

La figure 38 illustre le message du succès de la modification d'un client existant dans la liste.



Secteur	Nom	Prenom	Email	Entreprise	Tel N	Adresse		
Manouba	Said	Mejri	SaidMejri@Outlook.com	Monoprix	+2162222222	Ibno abi dhief	Supprimer	Modifier
centre ville	Samir	Lajnef	SamLaj@hotmail.com	Magasin Général	+2163562365	Centre ville	Supprimer	Modifier
Manouba	Mohamed	Abess	Moab@hotmail.com	Anouar	+2163632665	Manouba	Supprimer	Modifier
Manouba	Ahmed	Mejri	Ahmed@gmail.com	Aziza	+2169635563	Manouba	Supprimer	Modifier

Figure 39: Liste après modification

La figure 39 montre que la modification des données du client ont été bien enregistrés dans la liste des clients.

II.6. Interface de Liste des articles

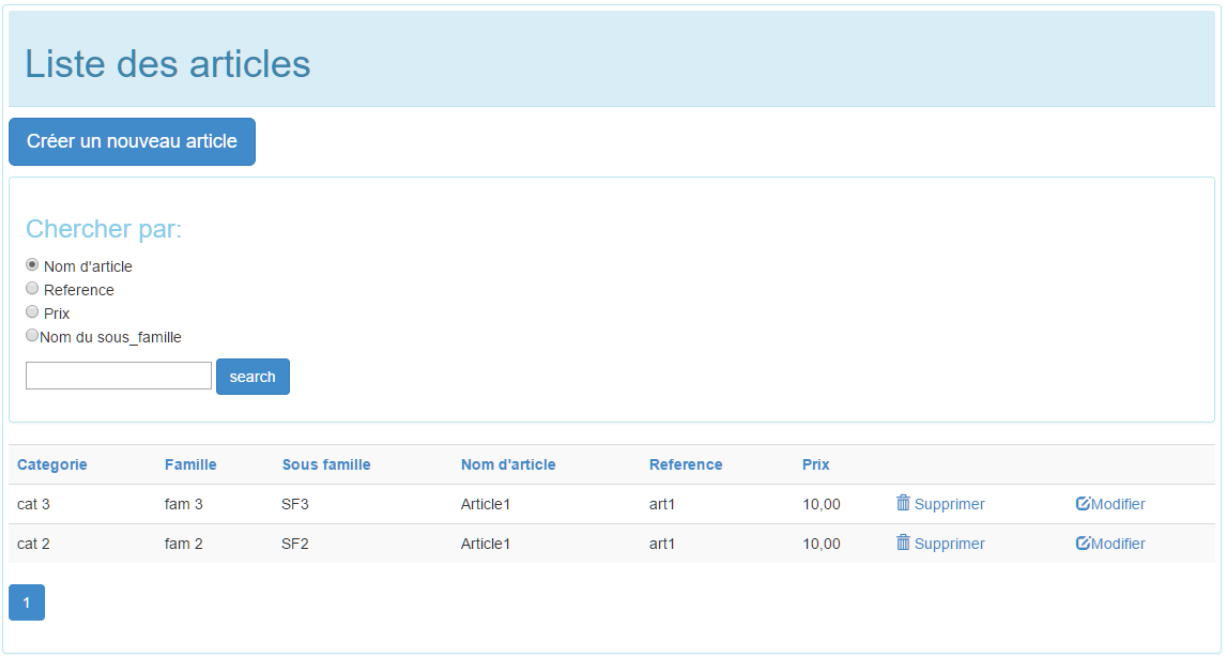


Figure 40: Liste des articles

La figure 40 illustre l'interface web qui représente la liste des articles. A partir de cette interface l'administrateur peut faire l'ajout, la modification, la recherche ou la suppression de l'article.

II.7. Interface d'ajout d'article

Figure 41: Ajout d'article

La figure 41 représente l'interface permettant à l'administrateur d'ajouter un nouvel Article.

II.8. Interface de d'ajout d'une visite

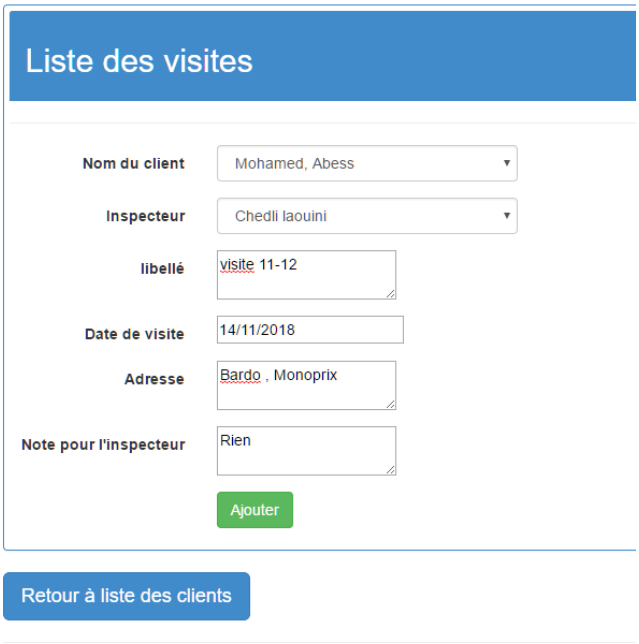


Figure 42: Ajout d'une visite

La figure 42 représente l'interface permettant à l'administrateur d'ajouter une nouvelle visite.

Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel et logiciel utilisé dans la réalisation. Puis nous avons fourni quelques captures d'écrans décrivant des interfaces de notre application web. Aussi nous avons essayé d'expliquer à travers ces interfaces le fonctionnement de notre système. Dans ce qui suit, nous allons conclure et donner quelques perspectives.

Conclusion générale

Dans ce rapport nous avons présenté notre projet qui consiste à implémenter et développer une plateforme pour la gestion du Back-office d'une solution SFA susceptible d'assurer la bonne gestion de l'administrateur avec les données stockées dans la base des données. Pour ce faire, nous avons commencé par présenter le cadre du stage, ensuite nous sommes passés à la spécification des besoins, puis nous avons détaillé notre conception et finalement, nous avons présenté le dernier aspect de la création d'une application web par la partie réalisation.

Nous avons réussi à améliorer nos performances et connaissances depuis la conception jusqu'à la réalisation d'un projet en pratiquant nos acquis et en dépassant toute sorte de nouveaux problèmes, quel que soit leur aspects techniques ou morales. Cela nous a notamment permis d'approfondir nos compétences au niveau du développement sur l'environnement Visual Studio 2015 en se basant sur les bases de données SQL Server. Un apprentissage de l'Asp.net a été au rendez-vous pour combler nos attentes du stage.

Concernant les perspectives d'avenir de ce projet, nous prévoyons d'ajouter des nouvelles fonctionnalités qui aident à la création de rapports d'activité. Nous comptons également finaliser quelques tâches qui manquent à notre application telles que l'établissement des statistiques.

Bibliographie

1. SFBT_CoP_n°3.pdf. *unglobalcompact.* [En ligne] 19 7 2014. [Citation : 17 11 2018.] https://www.unglobalcompact.org/system/attachments/cop_2014/95231/original/SFBT_CoP_n%C2%B03.pdf?1404998427.

2. SFBT - African Manager. *africanmanager*. [En ligne] [Citation : 19 11 2018.] https://africanmanager.com/documents/info_bourse/SFBT.pdf.

3. **Roques, P.** *UML 2 : Modéliser une application web.* 4e. s.l. : Eyrolles, 2008. pp. 1-21;50-164.

4. **Pillou, Jean-François.** Méthodes agiles (RAD, XP). *commentcamarche.* [En ligne] [Citation : 17 11 2018.] https://www.commentcamarche.com/contents/477-methodes-agiles-rad-xp.

5. Méthodologie agile SCRUM. *conceptit.* [En ligne] [Citation : 19 11 2018.] http://www.conceptit.fr/?p=1043.

6. le langage UML et le processus unifié. [En ligne] http://dspace.univ-tlemcen.dz/bitstream/112/5500/5/chapitre1.pdf.

7. **Sophnouille.** [En ligne] [Citation : 17 11 2018.] https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/.

8. [auteur du livre] P. ROCK. *UML 2 en action de l'analyse des besoins à la conception.* s.l. : EYROLLES, 2007.

9. **Cariou, Eric.** Architectures N-tiers. [En ligne] Octobre 2016. http://ecariou.perso.univ-pau.fr/cours/web/cours-architecture.pdf.

10. Conception et développement d'une application mobile de vente flash sous android. *Mémoire Online.* [En ligne] [Citation : 11 19 2018.] https://www.memoireonline.com/11/13/7773/m_Conception-et-developpement-d-une-application-mobile-de-vente-flash-sous-android17.html.