

Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique

\*\*\* \* \*\*\*

Université de Carthage

\*\*\* \* \*\*\*

**Institut National des Sciences  
Appliquées et de Technologie**



## **Projet de Fin d'Etudes**

Pour l'obtention du

### **Diplôme National d'Ingénieur En Sciences Appliquées et en Technologie**

Filière : Génie Logiciel

Sujet :

### **Refonte et extension de module de gestion commerciale SD-Leader (Sales & distribution System)**

*Réalisé par :* **Siwar GUEMRI**

Entreprise d'accueil :



*Soutenu le 19/06/2015*

Responsable à l'entreprise : Monsieur Tares FARES

Responsable à l'INSAT: Madame Wahiba BENFRADJ

**Année Universitaire : 2014/2015**

# Dédicaces

Au **Dieu** tout puissant mon créateur.

A mon père **Mohamed**, l'épaule solide, l'œil attentif compréhensif et la personne la plus digne de mon estime et de mon respect.

A ma mère **Fatma**, la personne qui m'a donné la vie, la tendresse et le courage pour réussir.

A mes très chères sœurs **Malak** et **Mariem** pour leurs soutiens.

A mon frère **Abd elKader** pour qui j'ai une grande admiration.

A tous mes proches, pour leur bienveillance et amour.

A tous ceux qui m'ont soutenu et cru en moi.

Je leur dédie ce modeste travail en leur souhaitant un immense bonheur et une joyeuse vie.

**Siwar GUEMRI**

# Remerciements

Ce travail n'aurait jamais pu se concrétiser sans l'aide et le soutien de plusieurs personnes que nous souhaitons vivement remercier et à qui nous dédions ce travail.

Madame **Wahiba BENFRADJ** notre encadrante, qui n'a pas cessé de nous prodiguer ses conseils et qui n'a épargné aucun effort pour contribuer à la réussite de notre travail.

Monsieur **Tarek FARES** et Monsieur **Abdessalam MARS**, pour leur accueil et la confiance qu'ils nous ont accordé au regard de ce projet.

Le personnel de **IT-Leaders**, pour l'expérience enrichissante et pleine d'intérêt qu'ils nous ont fait vivre durant la période du stage.

Tous nos professeurs et plus particulièrement les membres de jury qui ont accepté de juger notre travail.

Notre Institut qui nous a donnée l'occasion d'acquérir une formation professionnelle.

Toutes personnes ayant contribué de près ou de loin à l'élaboration de ce modeste travail.

# Table des matières

Introduction générale .....	1
Chapitre 1 : Contexte général du projet.....	3
Introduction : .....	4
1.1.    Organisme d'accueil : .....	4
1.1.1.    Présentation : .....	4
1.1.2.    Les domaines d'activités : .....	4
1.1.3.    Partenaires : .....	4
1.2.    Présentation du projet : .....	5
1.3.    Choix méthodologique : .....	6
1.3.1.    Les méthodes agiles : .....	6
1.3.2.    Méthodologie adoptée : .....	7
1.3.3.    Planning du projet : .....	9
Conclusion : .....	12
Chapitre 2 : Etude de l'existant .....	13
Introduction : .....	14
2.1. Analyse de l'existant : .....	14
2.2. Critiques de l'existant : .....	16
2.2.1. Gestion des autorisations: .....	16
2.2.2. Ergonomie : .....	16
2.2.3. Fonctionnalités : .....	18
Conclusion : .....	21
Chapitre 3 : Les processus métiers de la gestion commerciale .....	22
Introduction : .....	23
3.1. Gestion de processus métier: .....	23
3.2. La notation Business Process Model and Notation: .....	23
3.3. Etude des processus métiers: .....	24
3.3.1. Processus de gestion de devis : .....	24
3.3.2. Processus de gestion des bons de commandes : .....	25
3.3.3. Processus de génération d'un bon de livraison : .....	26
3.3.4. Processus d'exécution d'une livraison : .....	27
3.3.5. Processus de facturation : .....	28
3.3.6. Processus de paiement : .....	29
Conclusion : .....	30
Chapitre 4 : Analyse et spécification des besoins .....	31
Introduction : .....	32
4.1. Les acteurs du système : .....	32
4.2. Spécifications fonctionnelles détaillées : .....	32

4.2.1. Cas d'utilisation global :	32
4.2.2. Module gestion des utilisateurs :	35
4.2.3. Module gestion des clients :	36
4.2.4. Module gestion des devis :	38
4.2.5. Module gestion des bons de commande :	39
4.2.6. Module gestion des bons de livraisons :	41
4.2.7. Module gestion de stock :	42
4.2.8. Module Facturation :	44
4.2.9. Module Paiement :	45
4.3. Spécifications non-fonctionnelles :	46
Conclusion :	46
Chapitre 5 : Architecture et conception de la solution.....	47
Introduction :	48
5.1. Architecture :	48
5.1.1. Architecture technique :	48
5.1.2. Architecture applicative :	48
5.2. Conception :	49
5.2.1. Patrons utilisés :	49
5.2.2. Conception des modèles de données :	51
Conclusion :	58
Chapitre 6 : Implémentation de la solution.....	59
Introduction :	60
6.1. Environnement de travail :	60
6.1.1. Environnement matériel :	60
6.1.2. Environnement logiciel :	60
6.2. Implémentation et tests :	61
6.2.1. Sprint 1 : Module gestion de stock :	62
6.2.2. Sprint 2 : Module gestion des utilisateurs:	64
6.2.3. Sprint 3 : Module gestion des devis :	66
6.2.4. Sprint 4 : Module gestion des bons de commandes:	68
6.2.5. Sprint 5 : Module livraison :	70
6.2.6. Sprint 6 : Module gestion des bons de réceptions :	73
6.2.7. Sprint 7 : Module facturation:	75
6.2.8. Sprint 8 : Module paiement:	77
Conclusion :	79
Conclusion Générale .....	80
Bibliographie et Webographie.....	81
Annexe .....	82

# Acronymes

**MRP:** Material **R**equirement **P**lanning

**MRP II:** Manufacturing **R**equirement **P**lanning

**ERP :** Enterprise **R**essource **P**lanning

**PGI :** Progiciel de **g**estion **i**ntégré

**J2EE:** Java **E**nterprise **E**dition

**DAO:** **D**ata **A**ccess **O**bject

**INRIA :** Institut **n**ational de **r**echerche en **i**nformatique et en **a**utomatique

**BPM:** **B**usiness **p**rocess **m**anagement

**BPMN:** **B**usiness **P**rocess **M**odel and **N**otation

**BPMI:** **B**usiness **P**rocess **M**anagement **I**nitiative

**OMG:** **O**bject **M**anagement **G**roup

**UML:** Unifierd **M**odeling **L**anguage

**HTTP:** **H**yper **T**ext **T**ransfer **P**rotocol

**IoC:** Inversion of **C**ontrol

**JSF:** Java **S**erver **F**aces

**MVC :** **M**odel-**V**iew-**C**ontroller

# Table des figures

Figure 1 : Logo de la société IT-Leaders .....	4
Figure 2 : Logo de la société Biware Consulting .....	4
Figure 3 : Logo de la société SDA Consulting.....	5
Figure 4 : Cycle de vie de SCRUM [6] .....	9
Figure 5 : Diagramme de Gantt.....	11
Figure 6 : Architecture existante de SD-Leader .....	14
Figure 7 : Processus global de la gestion commerciale.....	20
Figure 8 : Processus de gestion des devis.....	25
Figure 9 : Processus de gestion des bons de commandes.....	26
Figure 10 : Processus de génération d'un bon de livraison.....	27
Figure 11 : Processus d'exécution d'une livraison.....	28
Figure 12 : Processus de facturation .....	29
Figure 13 : Processus de paiement .....	30
Figure 14: Diagramme de cas d'utilisation global.....	33
Figure 15 : Diagramme de cas d'utilisation « Gestion des utilisateurs ».....	35
Figure 16 : Diagramme de cas d'utilisation « Gestion des clients » .....	37
Figure 17 : Diagramme de cas d'utilisation « Gestion des devis ».....	38
Figure 18 : Diagramme de cas d'utilisation « Gestion des bons de commandes » .....	40
Figure 19 : Diagramme de cas d'utilisation « Gestion des bons de livraisons ».....	41
Figure 20 : Diagramme de cas d'utilisation « Gestion de stock ».....	43
Figure 21 : Diagramme de cas d'utilisation « Facturation ».....	44
Figure 22 : Diagramme de cas d'utilisation « Paiement » .....	45
Figure 23 : Architecture technique de l'application.....	48
Figure 24 : Architecture applicative de l'application .....	49
Figure 25 : Modèle d'accès aux données (DAO).....	50
Figure 26 : Modèle d'intégration des couches (IoC) .....	51
Figure 27 : Modèle des données pour le module de la gestion des clients .....	52
Figure 28 : Modèle des données pour le module gestion des devis .....	52
Figure 29 : Modèle des données pour le module gestion des bons de commandes .....	53
Figure 30 : Modèle des données pour le module gestion des bons de livraisons .....	54

Figure 31 : Modèle des données pour le module gestion des bons de réceptions .....	55
Figure 32 : Modèle des données pour le module gestion de stock.....	55
Figure 33 : Modèle des données pour le module de facturation.....	56
Figure 34 : Modèle des données pour le module gestion de paiement.....	57
Figure 35 : Modèle des données pour le module clôture de la caisse .....	57
Figure 36 : Modèle des données pour le module gestion des utilisateurs.....	58
Figure 37 : Diagramme de classes de module « Gestion de stock » .....	63
Figure 38 : Interface de gestion des opérations de stock.....	64
Figure 39 : Interface de l'historique des opérations .....	64
Figure 40 : Diagramme de classes de module « Gestion des utilisateurs » .....	65
Figure 41 : Interface de gestion des groupes.....	66
Figure 42 : Interface de gestion des utilisateurs.....	66
Figure 43 : Diagramme de classes de module « Gestion des devis ».....	67
Figure 44 : Interface de création d'un devis .....	68
Figure 45 : Interface de gestion des devis .....	68
Figure 46 : Diagramme de classes de module « Gestion des bons de commandes ».....	69
Figure 47 : Interface de création d'un bon de commande .....	70
Figure 48 : Interface de gestion des bons de commandes .....	70
Figure 49 : Diagramme de classes de module « Gestion des bons de livraisons » .....	71
Figure 50 : Interface de création d'un nouveau bon de livraison.....	72
Figure 51 : Interface de gestion des bons de livraisons.....	72
Figure 52 : Interface d'exécution d'une livraison.....	73
Figure 53 : Diagramme de classes de module « Gestion des bons de réceptions ».....	74
Figure 54 : Interface de gestion des bons de réception.....	75
Figure 55 : Diagramme de classes de module « Facturation ».....	76
Figure 56 : Interface de génération d'une nouvelle facture.....	77
Figure 57 : Interface de gestion des factures.....	77
Figure 58 : Diagramme de classes de module « Paiement ».....	78
Figure 59 : Interface de paiement.....	79
Figure 60 : Interface de gestion des opérations des paiements .....	79
Figure 61 : Exemple d'un Sprint Backlog.....	82



# Liste des tableaux

Tableau 1 : Tableau d'évaluation ergonomique .....	18
Tableau 2: Tableau des principaux cas d'utilisation et les acteurs correspondants .....	35
Tableau 3 : Tableau descriptif du cas d'utilisation « Gestion des utilisateurs » .....	36
Tableau 4 : Tableau descriptif du cas d'utilisation « Gestion des clients » .....	37
Tableau 5 : Tableau descriptif du cas d'utilisation « Gestion des devis » .....	39
Tableau 6 : Tableau descriptif du cas d'utilisation « Gestion des bons de commandes » .....	41
Tableau 7 : Tableau descriptif du cas d'utilisation « Gestion des bons de livraisons » .....	42
Tableau 8 : Tableau descriptif du cas d'utilisation « Gestion de stock » .....	43
Tableau 9 : Tableau descriptif du cas d'utilisation « Facturation » .....	45
Tableau 10 : Tableau descriptif du cas d'utilisation « Paiement » .....	45
Tableau 11 : Description de l'environnement logiciel utilisé .....	61
Tableau 12 : Tableau des sprints .....	61

# Introduction générale

Dans les années 70, l'informatique avait comme fonction l'automatisation des procédures. Ainsi chaque service avait son propre système d'information. Les applications étaient développées indépendamment les unes des autres. Ce qui au final s'avérera problématique pour les entreprises. Ainsi les données sont saisies à plusieurs reprises des systèmes d'information distincts. La naissance de MRP (Material Requirement Planning) et l'MRP II (Manufacturing requirement planning) a apporté une solution à ce problème. En fait, ils ont comme rôle principal de permettre la planification de la production de l'entreprise en fonction des ressources en personnel, en matières premières, en machines et en temps, par rapport à un besoin à une date ou un besoin de stock. Le MRP est l'ancêtre de l'ERP (Enterprise Ressources Planning). Les ERPs apportent une meilleure qualité de l'information permettant ainsi une meilleure décision. L'ERP gère l'ensemble des processus d'une entreprise en intégrant un ensemble des fonctions pour répondre rapidement aux besoins des entreprises. Avec l'évolution de la notion des MPRs et l'apparition des ERPs, il ne s'agit plus d'une gestion des activités/fonctions indépendantes de l'entreprise mais plutôt de processus de l'entreprise intégrant les fonctions de cette dernière. L'ERP est basé sur un ensemble de modules indépendants, qui partagent une base de données communes permettant d'assurer la communication des données inter modules.

Dans ce cadre, IT-leaders qui dispose d'un progiciel de gestion intégré déployé chez un ensemble de clients lance un projet de refonte et d'extension de cet ERP. La première étape de ce projet concerne le module SD-Leader (Sales & Distribution System) et le module MM-Leader (Maintenance Management System).

C'est dans ce contexte que la refonte de module SD-Leader a été proposée comme un projet de fin d'études. Notre projet se base sur l'étude des processus métiers de la gestion commerciale et l'utilisation de la méthodologie SCRUM.

Ce rapport présente les étapes de mise en œuvre de module SD-Leader. Notre démarche est organisée comme suit :

Le premier chapitre intitulé **Contexte générale du projet** est consacré à la présentation de l'environnement du stage au sein de la société **IT-Leaders** ainsi que la présentation du projet et la méthodologie adoptée.

Le chapitre suivant est intitulé **Etude de l'existant**. Dans ce chapitre, nous décrivons l'analyse et les critiques de l'application existante.

Le troisième chapitre s'intitule **Les processus métiers de la gestion commerciale** dans lequel nous détaillons l'étude des processus métiers de la gestion commerciale.

Le quatrième chapitre s'intitule **Analyse et spécification des besoins**. Ce chapitre est consacré à l'analyse et à la spécification des besoins fonctionnels et non-fonctionnels de la solution.

Le cinquième chapitre intitulé **Architecture et conception de la solution** est dédié à la présentation de l'architecture ainsi que la conception des patrons utilisés et des modèles de données.

Le dernier chapitre intitulé **Implémentation de la solution** présente les étapes de la réalisation du projet sous la forme des Sprints.

Le rapport s'achève par une **Conclusion Générale** dans laquelle nous exposerons les perspectives du projet.

---

# **Chapitre 1 : Contexte général du projet**

---

## Introduction :

Dans ce chapitre, nous abordons l'environnement du stage en présentant l'entreprise d'accueil **IT-Leaders**. Puis, nous présentons le projet en précisant la méthodologie adoptée. Nous clôturons ce chapitre par la description du plan de notre projet.

### 1.1. Organisme d'accueil :

#### 1.1.1. Présentation :

IT-Leaders [1] est une société d'ingénierie et de conseil qui offre son expertise en technologie de l'information dans le secteur des télécoms ainsi que l'industrie.



Figure 1 : Logo de la société IT-Leaders

#### 1.1.2. Les domaines d'activités :

Grace à l'expertise technologique de ses consultants et leurs connaissances métier dans les secteurs de télécoms, d'industrie et de services, IT-Leaders délivre un ensemble de services et de conseils dans les domaines suivants:

- Gestion des fraudes et assurance de revenus.
- Implémentation des projets Data-Warehouse et Business Intelligence.
- Développement spécifique selon les besoins du client.
- Assistance à la gestion des projets informatiques.

#### 1.1.3. Partenaires :

**Biware Consulting** est une société de services spécialisée dans la mise en place de solutions d'aide à la décision. [2]



Figure 2 : Logo de la société Biware Consulting

**SDA Consulting** est une société de services spécialisée dans l'audit, l'expertise comptable et le conseil. [3]



Figure 3 : Logo de la société SDA Consulting

## 1.2. Présentation du projet :

La société IT-leaders dispose d'un progiciel de gestion intégré déployé chez un ensemble de clients. La version actuelle de cet ERP présente quelques insuffisances en matière de fonctionnalités offertes ainsi que la technologie utilisée (architecture client/serveur en utilisant des clients lourds).

Afin de répondre aux besoins de ses clients, IT-Leaders lance un projet de refonte et extension de cet ERP. La première étape de ce projet concerne les modules suivants :

- SD-Leader: Sales & Distribution System.
- MM-Leader: Maintenance Management System.

Dans ce cadre, ce projet s'intitule **Refonte et extension du module de gestion commerciale SD-Leader (Sales & distribution System)**. SD-Leader est une application de gestion commerciale d'un progiciel de gestion intégré (PGI/ERP). Les modules qui constituent la nouvelle version de l'application sont :

- Le module de gestion de stock,
- le module de gestion des clients,
- le module de gestion des devis,
- le module de gestion des bons de commandes,
- le module de gestion des bons de livraisons,
- le module de gestion des bons de réceptions,
- le module de facturation,
- le module de paiement,
- le module de clôture de la caisse,
- le module de paramétrage,
- et le module de gestion des utilisateurs.

Ce travail rentre dans le cadre de notre projet de fin d'études qui vient conclure notre formation d'ingénieur en génie logiciel à l'Institut National des Sciences Appliquées et de Technologie (INSAT). Il s'intègre dans le cadre des projets de développement d'IT-Leaders.

### **1.3. Choix méthodologique :**

L'adoption d'une méthodologie de développement est une nécessité pour garantir un niveau de qualité acceptable et éviter tout débordement au niveau des délais.

#### **1.3.1. Les méthodes agiles :**

Une méthode agile est une approche itérative et incrémentale pour le développement de logiciel, réalisée de manière très collaborative par des équipes responsabilisées, en appliquant un cérémonial minimal, qui produisent, dans un délai contraint, un logiciel de grande qualité qui vise à répondre aux besoins changeants des utilisateurs. [4]

Le but d'une méthode agile est de maximiser la valeur ajoutée. Le développement s'effectue par itérations successives, il est possible, à la fin de chaque itération, de changer les priorités en faisant en sorte que les éléments apportant le plus de valeur soient réalisés a priori.

Les douze principes du manifeste agile sont :

- Satisfaire le client en livrant tôt et régulièrement des logiciels utiles, qui offrent une véritable valeur ajoutée.
- Accepter les changements, même tard dans le développement.
- Livrer fréquemment une application qui fonctionne.
- Collaborer quotidiennement entre clients et développeurs.
- Bâtir le projet autour de personnes motivées en leur fournissant environnement et support, et en leur faisant confiance.
- Communiquer par des conversations en face à face.
- Mesurer la progression avec le logiciel qui fonctionne.
- Garder un rythme de travail durable.
- Rechercher l'excellence technique et la qualité de la conception.
- Laisser l'équipe s'auto-organiser.
- Rechercher la simplicité.
- À intervalles réguliers, réfléchir aux moyens de devenir plus efficace.

### 1.3.2. Méthodologie adoptée :

Scrum signifie mêlée au rugby. Il exploite les valeurs et l'esprit du rugby et les adapte aux projets de développement. Comme le pack lors d'un ballon porté au rugby, l'équipe chargée du développement travaille de façon collective, soudée vers un objectif précis. Comme un demi de mêlée, le ScrumMaster aiguillonne les membres de l'équipe, les repositionne dans la bonne direction et donne le tempo pour assurer la réussite du projet. [5]

Scrum se base sur la théorie du contrôle empirique de processus. L'empirisme mentionne que les connaissances proviennent de l'expérience et d'une prise de décision basée sur des faits connus. Scrum utilise une approche itérative et incrémentale pour optimiser la prédictibilité et contrôler le risque.

Trois piliers soutiennent l'implémentation d'un contrôle empirique de processus : la transparence, l'inspection et l'adaptation.

**La transparence** requiert la définition d'un standard commun pour ces aspects afin que les observateurs partagent une compréhension commune de ce qui est observé.

**L'inspection** : les utilisateurs de Scrum doivent fréquemment inspecter les artefacts Scrum et l'état d'avancement par rapport à un objectif de Sprint (Sprint Goal) afin de détecter les écarts indésirables. La fréquence de ces inspections ne devrait pas gêner le travail en cours. Ces inspections sont bénéfiques lorsqu'elles sont effectuées de manière diligente sur les lieux du travail par les personnes qualifiées.

**L'adaptation** : si un inspecteur détermine qu'un ou plusieurs aspects du processus dérivent hors des limites acceptables, et que le produit qui en résulte sera inacceptable, le processus ou le matériel utilisé par le processus doit être ajusté. Un ajustement doit être fait dès que possible afin de minimiser le risque d'autres dérives.

L'équipe Scrum comprend trois rôles:

- **Le gestionnaire de produit (Product Owner)** qui a la responsabilité de décider du travail à réaliser.
- **Le maître SCRUM (ScrumMaster)** agit en tant que leader au service de l'équipe en aidant l'équipe et l'organisation à faire le meilleur usage de Scrum.
- **L'équipe de développement (Development Team)** est composée de professionnels qui travaillent pour fabriquer le produit de manière incrémentale avec une série de courtes périodes de temps appelées Sprints.



Scrum propose la création de trois artefacts essentiels :

- **Le carnet de produit (Product Backlog)** est une liste ordonnée des idées pour le produit, que l'on maintient dans l'ordre de fabrication attendu.
- **Le carnet d'itérations (Sprint Backlog)** constitue le plan détaillé du développement au sein du prochain Sprint.
- **L'incrément produit (Product Increment)** est le résultat attendu à l'issue de chaque Sprint. Il s'agit d'une version intégrée du produit, avec un niveau de qualité qui lui permet d'être déployé à la demande par le Product Owner.

Scrum prescrit cinq occasions formelles d'inspection et d'adaptation:

- **Le Raffinement du Backlog Produit (Product Backlog Refinement)** est une activité récurrente du projet Scrum pour maintenir le backlog produit ordonné en gérant ses items qui sont très souvent gros et étendus.
- **La Planification du Sprint (Sprint Planning)** est une réunion time boxée qui déclenche chaque sprint. Au cours de cette réunion, l'équipe Scrum collabore pour sélectionner et comprendre le travail à réaliser dans le Sprint à venir.
- **La Mêlée Quotidienne (Daily Scrum)** est une réunion qui a lieu aux mêmes endroits et en même temps tous les jours. Elle est utilisée par l'équipe de développement pour s'assurer qu'elle est convenable pour la situation d'atteindre l'objectif du Sprint.
- **La Revue de Sprint (Sprint Review)** est une réunion time boxée d'une heure par semaine où l'équipe Scrum et les parties prenantes examinent le résultat du Sprint.
- **La Rétrospective de Sprint (Sprint Retrospective)** est une réunion qui clôture chaque sprint dont l'objectif est d'examiner la façon dont les choses se sont déroulées vis-à-vis du processus, des relations entre les personnes et des outils.

La figure 4 représente le cycle de vie de SCRUM.



Figure 4 : Cycle de vie de SCRUM [6]

Le principe de Scrum est le suivant :

- Le product Owner crée une liste de tâches prioritaires appelée Product Backlog.
- Pendant le Sprint Planning, l'équipe prend une partie des tâches du Product Backlog pour préparer le Sprint Backlog et discuter la manière d'implémentation de ces tâches.
- L'équipe a une certaine période de temps de deux à quatre semaines (Sprint) pour achever le travail, mais elle se réunit chaque jour pour évaluer le progrès (Daily Scrum).
- Tout au long de la période, le Scrum Master essaie de maintenir l'équipe concentrée sur son objectif.
- À la fin de chaque sprint, le travail devrait être potentiellement livrable.
- Au départ du Sprint suivant, l'équipe prend une autre partie des tâches de Product Backlog et répéter les étapes précédentes.

Au-delà du Sprint, le cycle se répète jusqu'à ce que la liste des tâches du Product Backlog soit achevée ou le budget du projet est épuisé ou une date limite est atteinte. L'un de ces milestones marque la fin du travail sur le projet. Peu importe la cause de clôture, Scrum assure que le travail le plus important a été achevé lors de la fin du projet.

### 1.3.3. Planning du projet :

Dans ce projet, nous avons commencé par l'étude de l'existant puis la modélisation des processus métiers de la gestion commerciale. Par la suite, nous avons décomposé l'application en

plusieurs modules. Enfin, nous avons commencé le développement des modules. Chaque module représente un sprint de un à deux semaines. La préparation du rapport est une tâche qui a été effectuée tout au long du projet.

Ce stage a duré quatre mois à partir de février 2015. Nous avons commencé par une étude de l'existant et l'élaboration des processus métiers, cette phase a pris un mois qui est décomposé en deux tâches. La première tâche de 12 jours a été consacrée à l'étude de l'existant, la documentation sur les processus métiers et la rédaction d'un manuel d'utilisation pour l'application existante. La deuxième tâche de 14 jours a été dédiée à l'analyse de besoins et la modélisation des processus métiers de la gestion commerciale. La tâche d'élaboration du modèle conceptuel a duré une semaine. La phase de développement de l'application SD-Leader a duré neuf semaines.

La figure 5 présente en détails le déroulement du projet à l'aide d'un diagramme de Gantt :

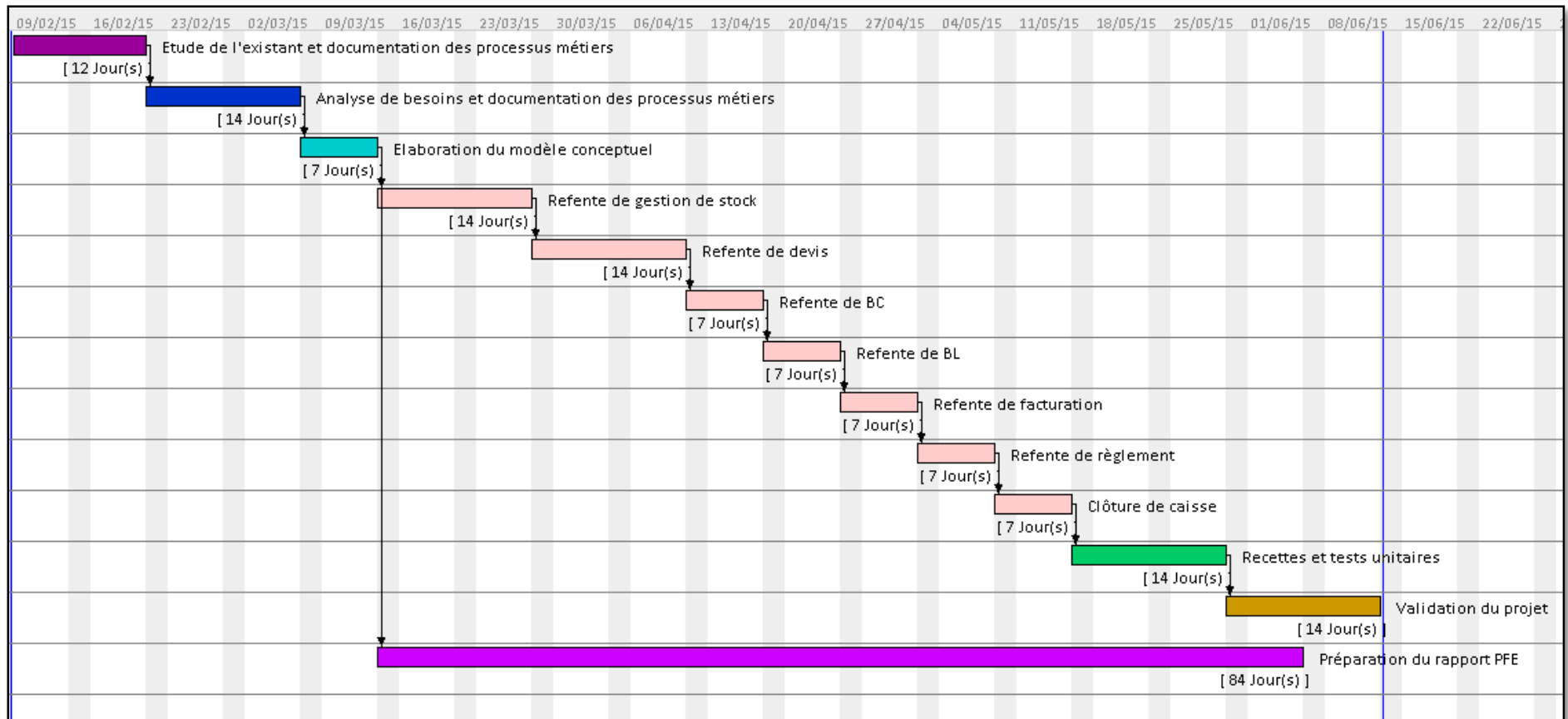


Figure 5 : Diagramme de Gantt

## **Conclusion :**

Ce chapitre a été consacré à la présentation de l'entreprise d'accueil **IT-Leaders**, au module de gestion commerciale SD-Leader sur le quel porte ce travail, à la méthodologie adoptée pour le projet à savoir SCRUM et au planning du travail. Dans Le chapitre suivant, nous allons aborder l'étude de l'existant.

---

## **Chapitre 2 : Etude de l'existant**

---

## Introduction :

Au cours de ce chapitre, nous décrivons l'étude que nous avons faite sur l'application existante. En premier lieu, nous présentons l'application à travers l'architecture et les fonctionnalités. En second lieu, nous citons les critiques de l'application.

### 2.1. Analyse de l'existant :

SD-Leader est un prototype du module de gestion commerciale d'un progiciel de gestion intégré développé par la société IT- Leaders.

SD-Leader est une application web à trois couches qui sont la couche présentation, la couche service et la couche accès aux données.

La couche présentation contient les différents pages xhtml de l'application qui représente les interactions avec l'utilisateur. La couche service contient les classes qui offrent les différents services et code métiers et elle contient aussi les composants métiers qui gère les différents fonctionnalités. En fin la couche accès aux données qui contient les classes persistantes. La figure 6 représente l'architecture de l'application existante.

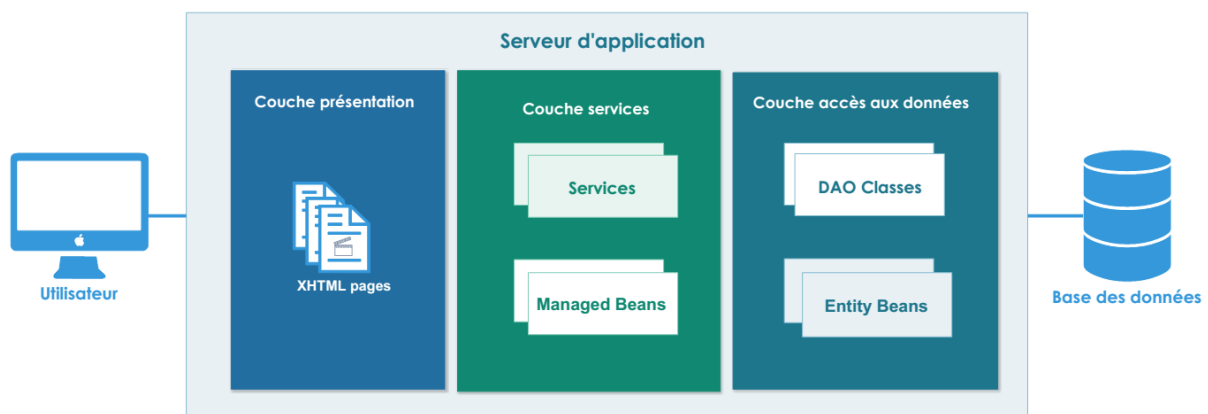


Figure 6 : Architecture existante de SD-Leader

Les modules constituant l'application existante sont:

- Le module de gestion des clients,
- le module de gestion des articles,
- le module de gestion des devis,
- le module de gestion des bons de commandes,
- le module de gestion des bons de livraisons,

- le module de facturation,
- et le module de paiement.

Pour mieux analyser l'application existante et découvrir les différentes fonctionnalités de l'existant, nous avons rédigé un manuel d'utilisation détaillé. Dans ce qui suit, nous allons présenter une brève description de chaque module de l'application.

#### **Gestion des clients :**

Les opérations de la gestion des clients sont la création d'un nouveau client, la recherche, la modification et la suppression.

#### **Gestions des articles :**

Les opérations de la gestion des articles sont la création d'un nouvel article, la recherche, la modification et la suppression.

#### **Gestion des devis :**

Les opérations de la gestion des devis sont la création d'un nouveau devis, la recherche, la modification et la suppression.

#### **Gestion des bons de commandes :**

Les opérations de la gestion des bons de commandes sont la création d'un nouveau bon de commande, la recherche, la modification et la suppression.

#### **Gestion des bons de livraisons :**

Les opérations de la gestion des bons de livraisons sont la création d'un nouveau bon de livraison, la recherche, la modification et la suppression.

#### **La facturation :**

Les opérations du module de facturation sont la génération d'une nouvelle facture, la recherche, la modification et la suppression.

#### **Le paiement :**

Les opérations du module de paiement sont la règlementation d'un paiement, la recherche, la modification et la suppression.



## 2.2. Critiques de l'existant :

SD-Leader est un prototype d'un module gestion commerciale d'un ERP qui englobe un ensemble de modules non développés et une interface Homme-Machine peu améliorée. Après l'évaluation de l'application et la rédaction d'un manuel d'utilisation, nous abordons les critiques de cette application qui concernent la gestion des autorisations d'accès à l'application, les fonctionnalités proposées par l'application et l'ergonomie.

### 2.2.1. Gestion des autorisations:

Nous avons remarqué que tous les utilisateurs ont l'accès aux mêmes modules d'où la notion des rôles n'a pas été considérée. Il est donc important d'ajouter un nouveau module pour gérer les utilisateurs et leurs privilèges. Ce module est accessible par l'administrateur de l'application. Les rôles sont définis selon les acteurs de l'application et les droits d'accès à l'application sont gérés selon ces rôles.

### 2.2.2. Ergonomie :

Nous avons évalué l'ergonomie de l'application selon les huit critères de Bastien et Scapin (INRIA) [7] qui sont :

- Le critère **Guidage** : c'est l'ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'application.
- Le critère **Charge de Travail** : il concerne l'ensemble des éléments de l'interface qui ont un rôle dans la réduction de la charge perceptive, mnésique ou physique des utilisateurs et dans l'augmentation de l'efficacité du dialogue.
- Le critère **Contrôle Explicite** : il concerne d'une part, la nécessité de prise en compte par le système des actions explicites des utilisateurs, et d'autre part, le contrôle dont disposent les utilisateurs sur le traitement de leurs actions.
- Le critère **Adaptabilité** : il concerne sa capacité d'un système à réagir selon le contexte, et selon les besoins et préférences des utilisateurs.
- Le critère **Gestion des Erreurs** : il concerne tous les moyens permettant d'une part d'éviter ou de réduire les erreurs, et d'autre part de les corriger lorsqu'elles surviennent.
- Le critère **Homogénéité/Cohérence** : il se réfère à la façon selon laquelle les choix de conception de l'interface (dispositifs, modalités, codes, dénominations, formats)

sont conservés pour des contextes identiques, et sont différents pour des contextes différents.

- Le critère **Signifiante des Codes et Dénominations** : il concerne l'adéquation entre l'objet, le comportement ou l'information présentée et son référent. Des codes et dénominations "signifiants" disposent d'une relation sémantique forte avec leur référent.
- Le critère **Compatibilité** : il se réfère à l'accord pouvant exister entre les caractéristiques des utilisateurs (mémoire, perceptions, anatomie, habitudes, compétences, âge, attentes, etc.) et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part.

Le tableau 1 représente l'évaluation de l'interface Homme-Machine de l'application:

Critères		Evaluation	Justification
Guidage	Incitation	★☆☆☆☆	Dans les formulaires d'ajout, il n'y a pas des indications sur les champs obligatoires à remplir.
	Groupe ment	★☆☆☆☆	Dans le menu « paramètre », les formulaires sont classés en séquence et non regroupé par catégorie.
	Feed-back immédiat	★★★★☆	Il y a des messages de feedback immédiat lors de l'ajout, la suppression ou lors de l'oubli d'un champ obligatoire à remplir.
	Lisibilité	★★★☆☆	Les caractères et les couleurs utilisés sont lisibles sur un fond blanc mais il y a utilisation du couleur gris qui est dédié aux champs désactivés.
Charge de travail	Bré vité : concision	★★★★☆	Les titres et labelles sont courts et descriptifs.
	Bré vité : action minimales	★★★★☆	A côté des éléments il y a des boutons modifier et supprimer ou des boutons pour réaliser les taches suivants.
	Densité d'information	★★★★☆	Il n'y a pas des textes descriptifs long ni des informations denses.

<b>Contrôle explicite</b>	Actions explicites	★★★★★	Les systèmes n'exécutent que les actions réalisées par l'utilisateur.
	Contrôle utilisateur	★★★☆☆	Il n'y a pas des boutons « annuler » ou « quitter » dans quelques formulaires.
<b>L'adaptabilité</b>	Flexibilité	★★★☆☆	Il n'y a pas une flexibilité totale car il y a un enchaînement pour réaliser une suite des tâches.
	Prise en compte de l'expérience de l'utilisateur	★★★☆☆	Il y a des restrictions sur les tâches. Les tâches sont dépendantes.
<b>Gestion des erreurs</b>	Prévention des erreurs	★★★★★	Présence des listes de choix pour éviter les fautes de saisie.
	Qualité des messages d'erreur	★★★★★	Les messages d'erreurs et d'avertissement décrivent correctement les erreurs.
	Correction des erreurs	★★★☆☆	Il n'y a pas des indications pour la correction des erreurs.
<b>Homogénéité cohérence</b>		★★★☆☆	Les éléments et les styles sont cohérents avec les tâches réalisées par le système.
<b>Signification des codes et dénominations</b>		★★★☆☆	Il n'y a pas des icônes descriptives et significatives.
<b>Compatibilité</b>		★★★★★	L'application est dédiée qu'aux agents de la gestion commerciale.

Tableau 1 : Tableau d'évaluation ergonomique

### 2.2.3. Fonctionnalités :

Un ERP doit se baser sur les processus métiers afin de répondre aux besoins des utilisateurs, or l'application existante SD-Leader n'utilise pas la notion des processus métiers. Donc, il est primordial de modéliser un processus générique pour la gestion commerciale

Ainsi, les fonctionnalités du module SD-Leader doivent impérativement respecter le processus métier de la gestion commerciale. Celui-ci doit être décomposé en plusieurs sous processus, correspondant chacun à une fonctionnalité du module. Ces sous-processus sont :

- Le processus de gestion des devis,
- le processus de gestion des bons de commandes,
- le processus de génération d'un bon de livraison,
- le processus d'exécution d'une livraison,
- le processus de facturation,
- et le processus de paiement.

Dans le chapitre suivant, nous allons voir en détails le processus métier de la gestion commerciale ainsi que ses différents sous processus.

La figure 7 représente le processus métier de la gestion commerciale, les liens entre les différents sous processus et les différents acteurs.

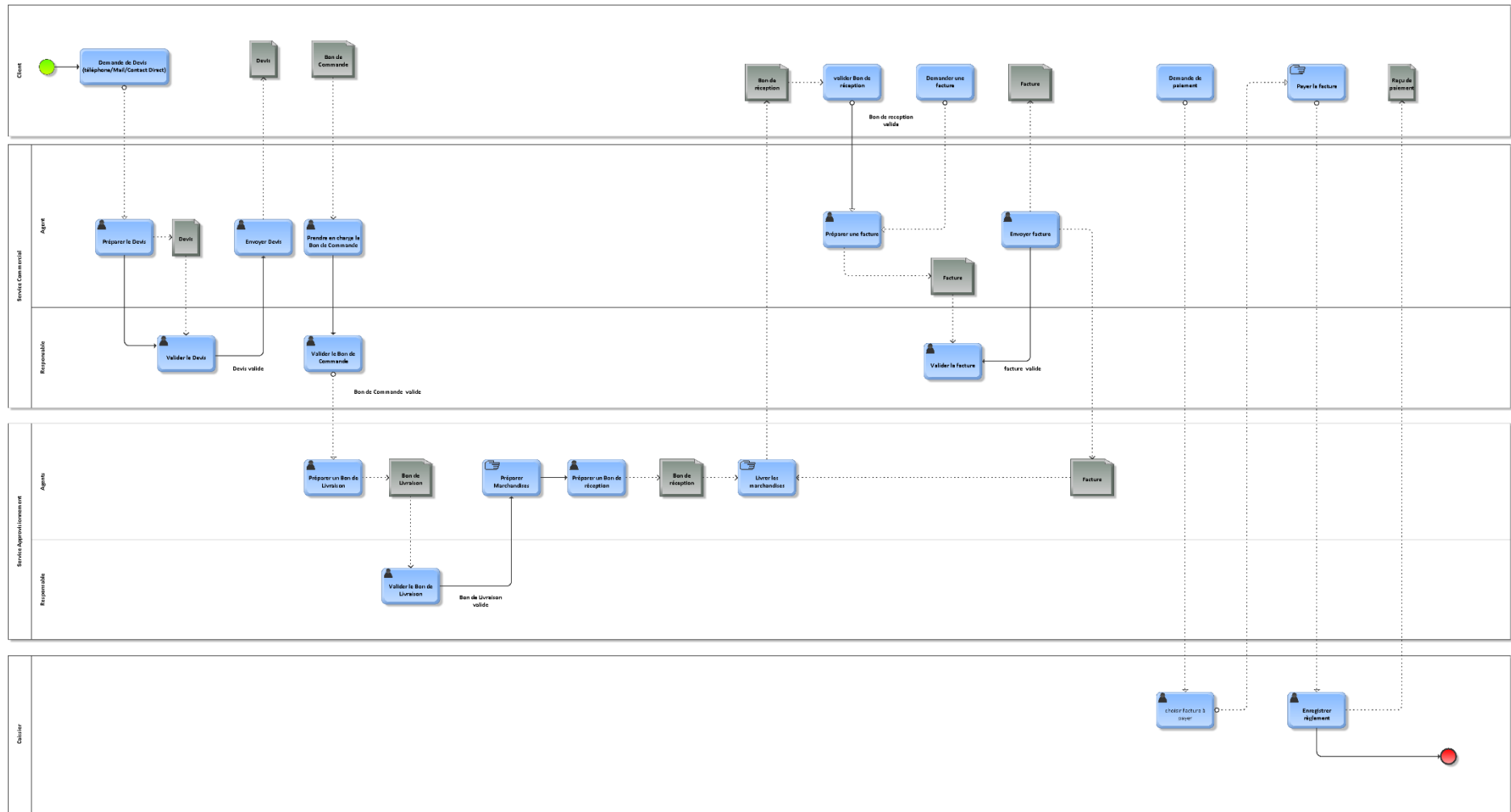


Figure 7 : Processus global de la gestion commerciale

## **Conclusion :**

Au cours de ce chapitre, nous avons analysé et critiqué l'application existante à travers son architecture et ses fonctionnalités. Dans le chapitre suivant, nous étudions les processus métiers de la gestion commerciale.

---

## **Chapitre 3 : Les processus métiers de la gestion commerciale**

---

## **Introduction :**

Ce chapitre est consacré à l'étude de processus métier de la gestion commerciale. L'étude de chaque processus commence par une description et finit par dévoiler quelques fonctionnalités de l'application.

### **3.1. Gestion de processus métier:**

Un processus est un flux séquentiel des activités aboutissant à un résultat déterminé. Il est déclenché par un évènement qui lui est externe et doit aboutir à un résultat qui est sa raison d'être. Un processus représente les interactions entre différents acteurs sous la forme des échanges d'informations. Ces acteurs peuvent être des humains, des applications, des services ou des processus tiers.

La gestion de processus métier ou Business Process Management (BPM) [8] est l'approche qui consiste à modéliser les processus métiers de l'entreprise dans leur aspect applicatif qu'humain.

L'objectif de cette démarche est d'aboutir à une meilleure vue globale de l'ensemble des processus métiers de l'entreprise et de leurs interactions afin d'être en mesure de les optimiser et de les automatiser au maximum à l'aide d'applications métiers.

Le BPM traite les processus dans leurs dimensions spatiale et temporelle. Dans leur dimension spatiale, le BPM considère les processus de bout en bout, traversant la chaîne de valeur de l'entreprise étendue, depuis la chaîne d'approvisionnement intégrant les fournisseurs jusqu'à l'intégration des clients. L'objectif du BPM est bien de donner à une entreprise une visibilité de bout en bout sur ses processus.

Dans leur dimension temporelle, le BPM englobe la totalité du cycle de vie des processus : modélisation, informatisation, déploiement, exécution, supervision et optimisation. L'objectif du BPM est bien de garantir à une entreprise que ses processus sont en permanence adaptés à son environnement en constante évolution.

### **3.2. La notation Business Process Model and Notation:**

La notation Business Process Model and Notation (BPMN) a été développée par la Business Process Management Initiative (BPMI), et est maintenant maintenue par l'Object



Management Group (OMG) depuis leur fusion en 2005. La spécification BPMN 1.0 a été libérée au public en mai 2004. La version actuelle de BPMN est la 2.0 depuis mars 2011.

L'objectif principal de développement de BPMN était de fournir une notation qui soit facilement compréhensible par tous les utilisateurs de l'entreprise, des analystes métiers qui créent les ébauches initiales des processus, les développeurs responsables de la mise en œuvre de la technologie qui va exécuter ces processus, et enfin, pour les gens d'affaires qui gèrent et contrôlent les processus. BPMN crée un pont normalisé entre la conception de processus métier et son implémentation. [9]

La notation BPMN permet notamment de découpler l'information métier de l'information technique afin de maximiser sa portabilité.

### **3.3. Etude des processus métiers:**

Le processus commercial décrit dans le chapitre précédent est décomposé en plusieurs sous processus pour mieux capter les différents besoins. Nous traitons ces sous processus un par un en commençant par le processus de gestion des devis, puis le processus de gestion des bons de commandes, le processus de génération d'un bon de livraison, le processus d'exécution d'une livraison, le processus de facturation et finalement le processus de paiement.

#### **3.3.1. Processus de gestion de devis :**

La figure 8 illustre le diagramme BPMN du processus de gestion des devis :

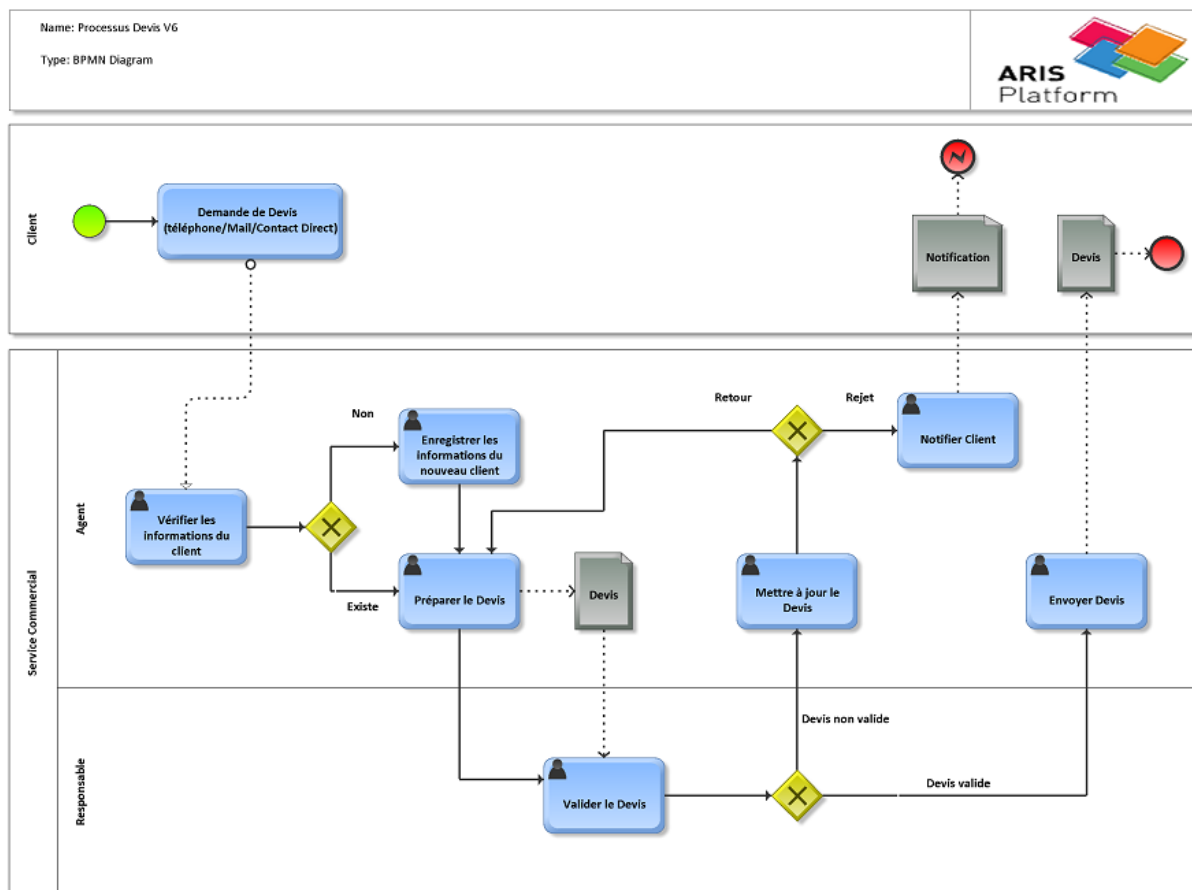


Figure 8 : Processus de gestion des devis

Les acteurs principaux de ce processus sont le client, l'agent commercial et le responsable commercial.

L'événement déclencheur du processus de gestion des devis est la réception d'une demande par un appel téléphonique ou par mail ou par un contact direct de la part d'un client. Après la création de devis, le responsable commercial doit le valider. Si le responsable rejette ce devis, l'agent doit notifier le client.

En se basant sur ce processus, nous constatons que plusieurs fonctionnalités doivent être ajoutées à la nouvelle version de l'application :

- L'ajout d'un nouveau client dans le module de gestion des devis.
- La validation d'un devis par un responsable.
- L'impression d'un devis.

### 3.3.2. Processus de gestion des bons de commandes :

La figure 9 illustre le diagramme BPMN du processus de gestion des bons de commandes :

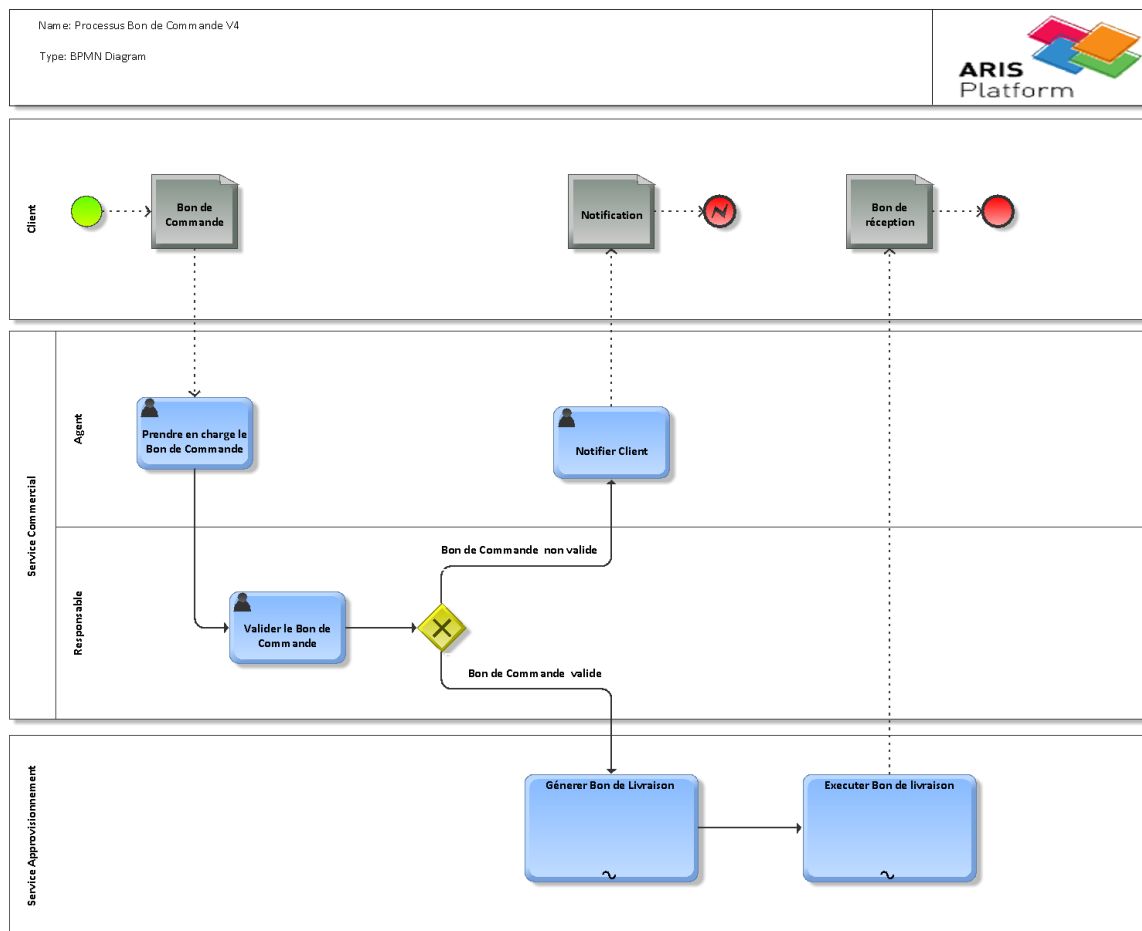


Figure 9 : Processus de gestion des bons de commandes

L'évènement déclencheur du processus est la réception d'une commande de la part du client. L'agent commercial prépare le bon de commande puis il l'envoi au responsable pour le valider. Si le bon est non valide, l'agent doit notifier le client que sa commande est rejetée. Si non le bon de commande est envoyé au service approvisionnement pour générer un bon de livraison et exécuter.

En se basant sur ce processus, nous constatons que plusieurs fonctionnalités doivent être ajoutées à la nouvelle version de l'application :

- La validation d'un bon de commande par le responsable commercial.
- Il faut prendre en considération que la commande peut être rejetée.

### 3.3.3. Processus de génération d'un bon de livraison :

Le processus de génération d'un bon de livraison traite les différentes étapes pré-livraison. Ce processus est totalement exécuté par le service approvisionnement. La figure 10 représente le processus de génération d'un bon de livraison.

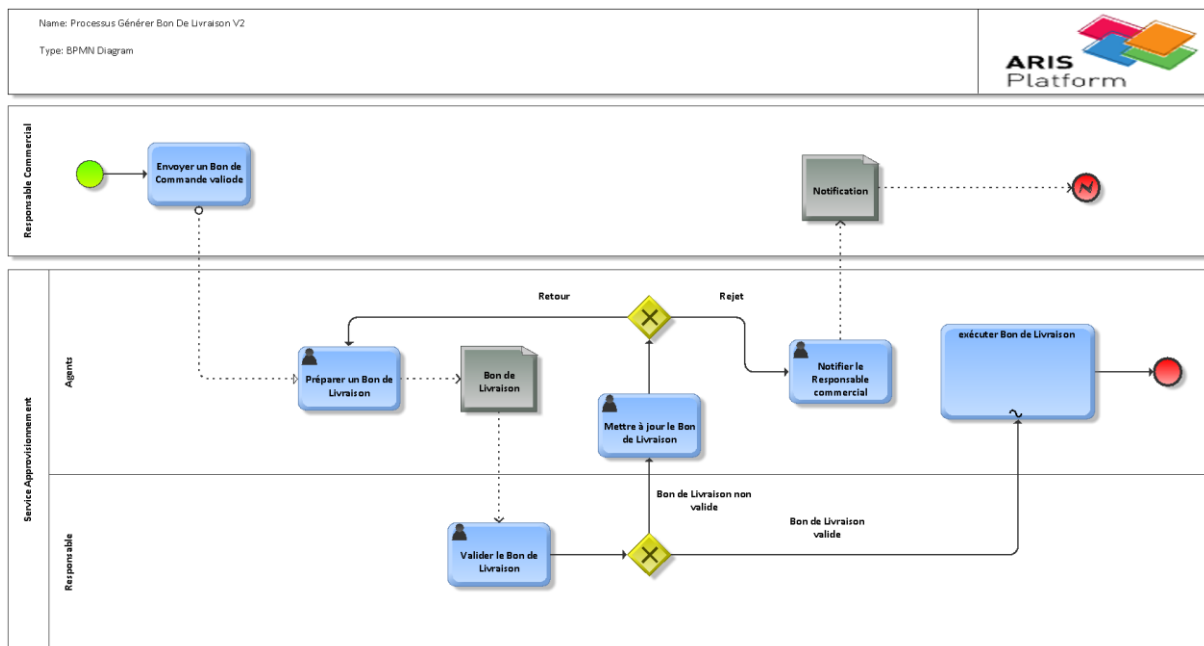


Figure 10 : Processus de g n ration d'un bon de livraison

L' v nement d clencheur de ce processus est la r ception d'un bon de commande valide de la part du service commercial. L'agent approvisionnement g n re le bon de livraison. Il l'envoie vers le responsable pour le valider. Si le bon de livraison est valide, le service commercial passe au processus d'ex cution de livraison.

En se basant sur ce processus, nous constatons que plusieurs fonctionnalit s doivent  tre ajout es   la nouvelle version de l'application :

- La validation d'un bon de livraison.
- Les fonctionnalit s de mise   jour et de rejet d'un bon de livraison.

### 3.3.4. Processus d'ex cution d'une livraison :

Le processus d'ex cution d'une livraison englobe le processus de pr paration des marchandises et le processus administratif qui consiste   pr parer un bon de r ception. Le diagramme BPMN dans la figure 11 d crit ce processus :

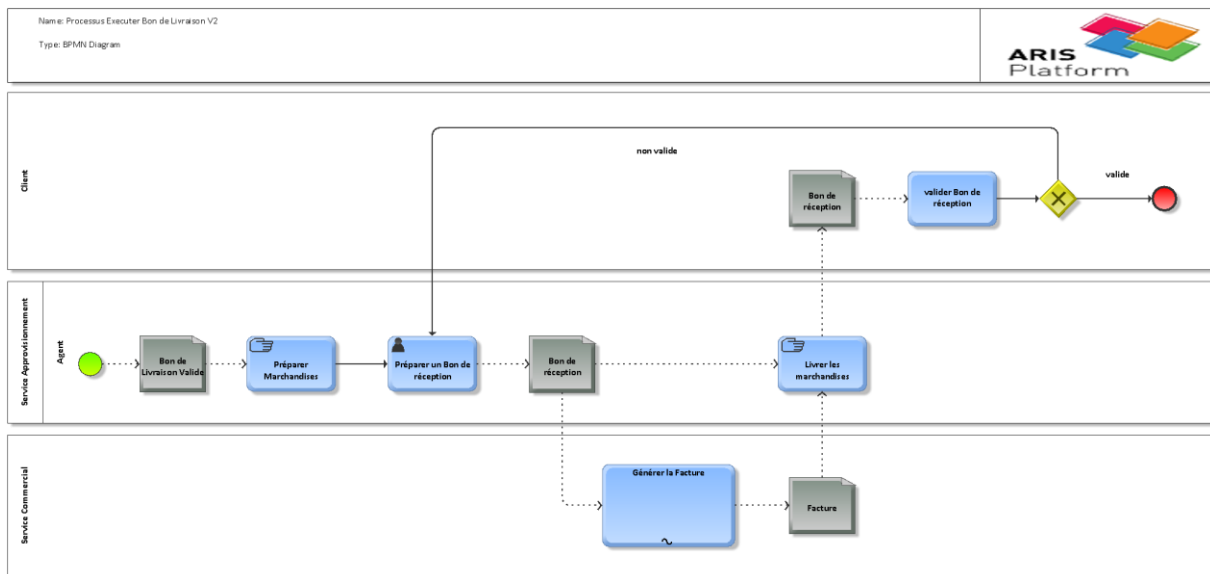


Figure 11 : Processus d'exécution d'une livraison

L'évènement déclencheur de ce processus est la validation du bon de livraison par le responsable approvisionnement. La préparation des colis des marchandises est une action manuelle. L'agent prépare un bon de réception et l'envoie au service commercial pour la génération d'une facture. Le bon de réception et le colis des marchandises sont livrés au client. Et enfin, le client valide le bon de réception.

Toutes les fonctionnalités de ce processus seront développées dans la nouvelle version de l'application.

### 3.3.5. Processus de facturation :

La figure 12 représente le diagramme BPMN du processus de facturation :

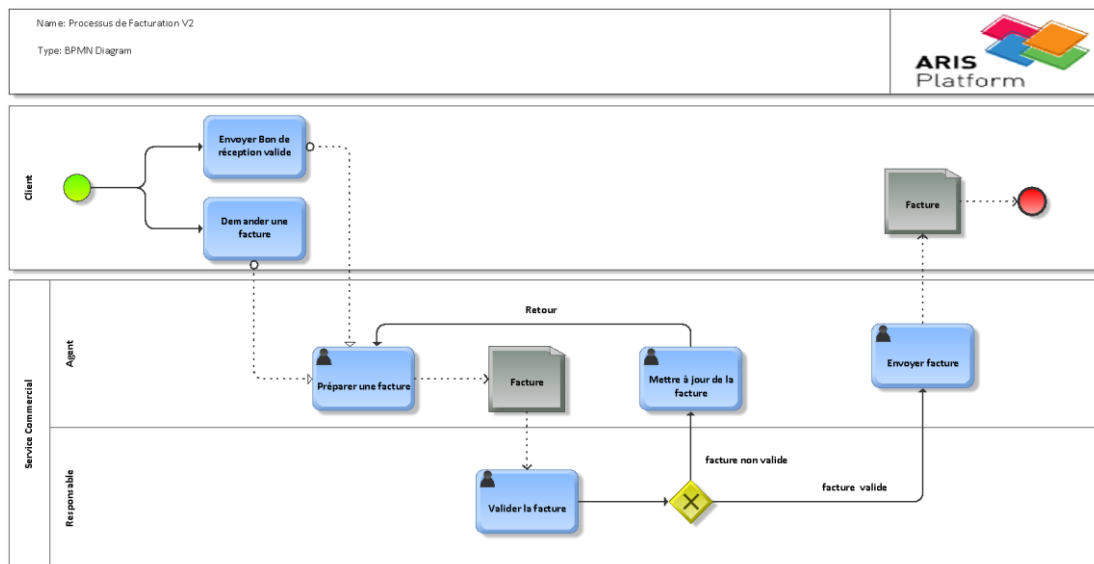


Figure 12 : Processus de facturation

Le processus de facturation est déclenché suite à la réception d'un bon de réception valide ou à une demande d'une facture de la part d'un client. Ensuite, l'agent génère la facture, avant que le responsable commercial la valide.

En se basant sur ce processus, nous constatons que plusieurs fonctionnalités doivent être ajoutées à la nouvelle version de l'application. Ces fonctionnalités sont:

- La validation d'une facture par un responsable commercial.
- L'impression d'une facture.

### 3.3.6. Processus de paiement :

Le module de paiement décrit les différentes étapes de paiement. La figure 13 représente le diagramme BPMN de ce processus :

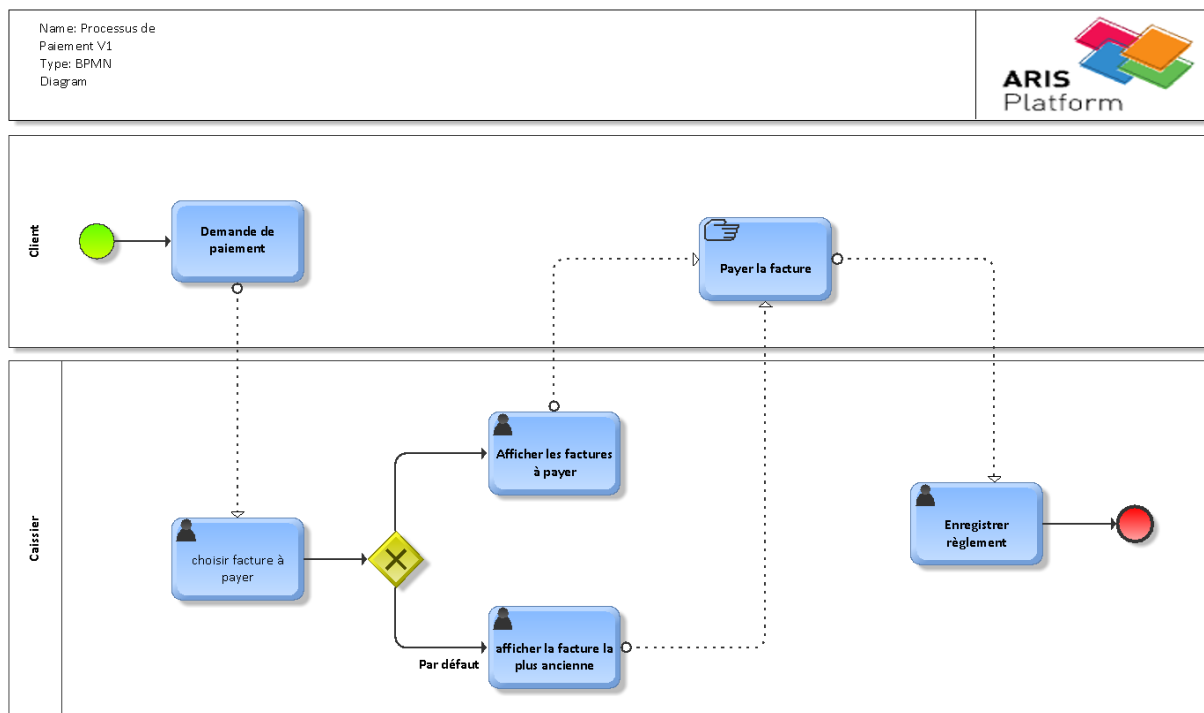


Figure 13 : Processus de paiement

L'évènement déclencheur du processus est la demande d'un client pour payer une facture. Après le paiement d'une facture, l'agent enregistre ce règlement dans le système et marque la facture comme payée.

En se basant sur ce processus, nous constatons que plusieurs fonctionnalités doivent être ajoutées à la nouvelle version de l'application. Ces fonctionnalités sont :

- Marquer la facture comme payée.
- Impression d'un reçu de paiement.

## Conclusion :

Ce chapitre a été consacré à l'étude en détails de tous les sous processus métiers de la gestion commerciale qui seront respectés dans notre nouvelle application. Le chapitre suivant sera consacré à l'analyse et à la spécification des besoins fonctionnels et non fonctionnels.

---

## **Chapitre 4 : Analyse et spécification des besoins**

---



## **Introduction :**

Ce chapitre est consacré à l'analyse et à la spécification des besoins fonctionnels et non fonctionnels de la solution qui est une étape primordiale pour la réalisation de notre projet.

### **4.1. Les acteurs du système :**

Avant de voir les besoins de notre système, nous allons déterminer Les principaux acteurs de celui-ci qui sont:

- **Administrateur :**

Cet acteur possède tous les droits d'accès qui lui permettent d'administrer le système. Sa fonction principale est la gestion des utilisateurs.

- **Service commercial :**

Ce service comprend un responsable et plusieurs agents. C'est le principal service de la gestion commerciale. Ce service est responsable de la gestion des devis, de la gestion des commandes et de la facturation. Le rôle « service commercial » sera attribué aux agents et le responsable de ce service.

- **Service d'approvisionnement :**

Le service d'approvisionnement est responsable de la gestion de stock, de la livraison et de la gestion des bons de réceptions.

- **Caissier :**

Les fonctions principales du caissier sont le paiement de la facture et la clôture de la caisse.

### **4.2. Spécifications fonctionnelles détaillées :**

La spécification des besoins sert à associer chaque acteur réactif du système à l'ensemble d'actions avec lesquelles il intervient. Nous utilisons les diagrammes des cas d'utilisations pour modéliser les besoins fonctionnels.

#### **4.2.1. Cas d'utilisation global :**

Le diagramme des cas d'utilisation dans la figure 14 donne une vue global sur le système:

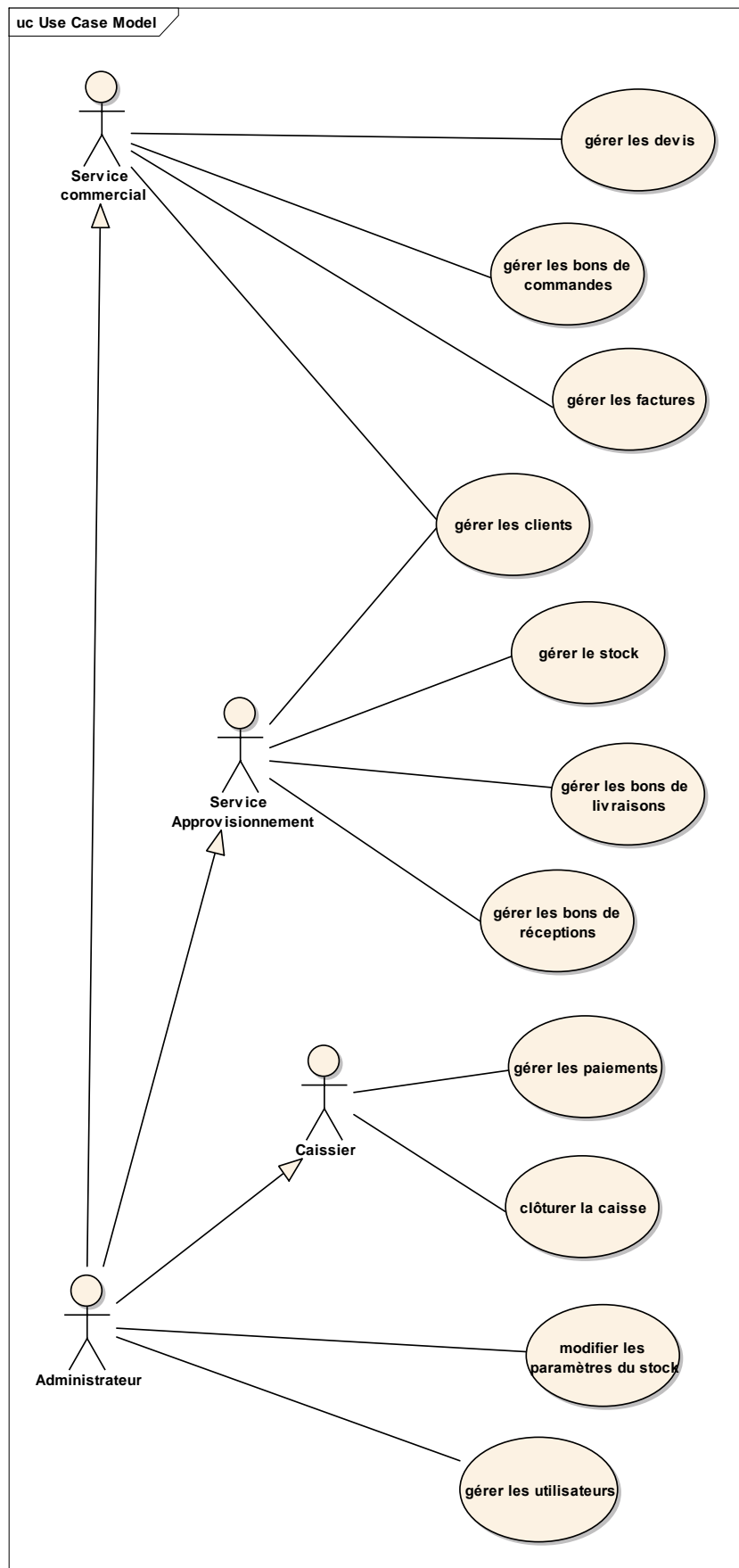


Figure 14: Diagramme de cas d'utilisation global

Le tableau 2 représente les différents cas d'utilisation avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Gérer les clients	Service commercial et service Approvisionnement	<b>Reçus</b> : Liste des clients <b>Emis</b> : Ajouter, modifier, supprimer, afficher détails, options
Gérer les devis	Service commercial	<b>Reçus</b> : Liste des devis, liste des clients <b>Emis</b> : ajouter, modifier, supprimer, afficher détails, imprimer
Gérer les bons de commandes	Service commercial	<b>Reçus</b> : Liste des bons de commandes, liste des devis <b>Emis</b> : ajouter, modifier, supprimer, afficher détails, imprimer.
Gérer les bons de livraisons	Service Approvisionnement	<b>Reçus</b> : Liste des bons de livraisons, liste des bons de commandes. <b>Emis</b> : ajouter, modifier, supprimer, afficher détails, imprimer
Gérer les bons de réceptions	Service Approvisionnement	<b>Reçus</b> : Liste des bons de réceptions, liste des bons de livraisons <b>Emis</b> : ajouter, modifier, supprimer, afficher détails, imprimer
Gérer le stock	Service Approvisionnement	<b>Reçus</b> : Liste des articles, liste des opérations. <b>Emis</b> : opération d'entrée, sortie, transfert du stock, consulter l'historique des opérations.
Gérer les factures	Service commercial	<b>Reçus</b> : Liste des factures, liste des bons de réceptions. <b>Emis</b> : ajouter, modifier, supprimer, afficher détails, imprimer
Gérer les paiements	Caissier	<b>Reçus</b> : Liste des paiements, liste des factures. <b>Emis</b> : régler paiement, modifier, supprimer, afficher détails, imprimer reçu.

Clôturer la caisse	Caissier	<b>Reçus</b> : liste des paiements <b>Emis</b> : vérifier caisse, clôturer.
Gérer les utilisateurs	Administrateur	<b>Reçus</b> : Liste des utilisateurs <b>Emis</b> : Ajouter, modifier, supprimer, afficher détails, modifier les privilèges
paramétrer	Administrateur	<b>Reçus</b> : Les différents variables de système. <b>Emis</b> : ajouter, modifier, supprimer, gérer utilisateurs.

Tableau 2: Tableau des principaux cas d'utilisation et les acteurs correspondants

#### 4.2.2. Module gestion des utilisateurs :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Gestion des utilisateurs » :

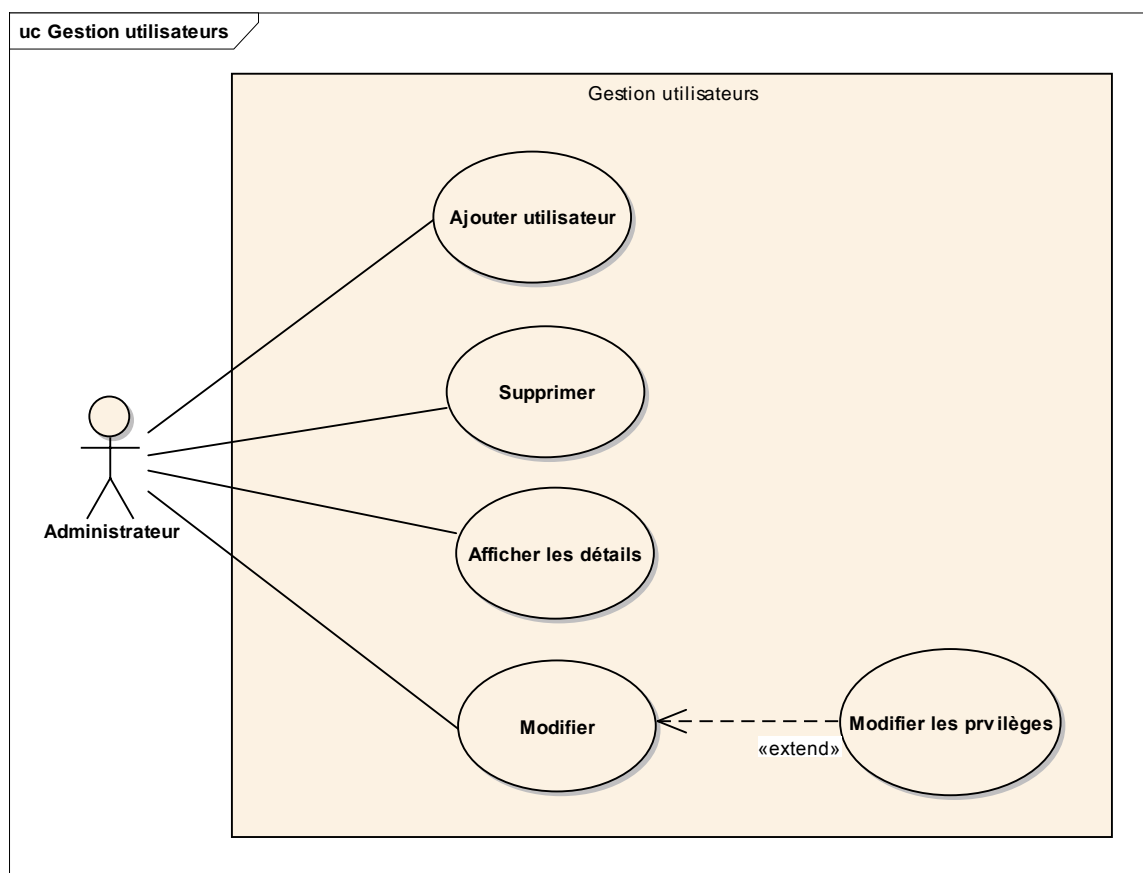


Figure 15 : Diagramme de cas d'utilisation « Gestion des utilisateurs »

Le tableau 3 représente les différents cas d'utilisation « Gestion des utilisateurs » avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Ajouter utilisateur	Administrateur	<b>Reçus</b> : Formulaire d'ajout <b>Emis</b> : nouvel utilisateur crée.
Afficher les détails	Administrateur	<b>Reçus</b> : liste des utilisateurs <b>Emis</b> : consulter les détails de l'utilisateur sélectionné
Modifier	Administrateur	<b>Reçus</b> : informations de l'utilisateur <b>Emis</b> : utilisateur modifié
Supprimer	Administrateur	<b>Reçus</b> : message de confirmation <b>Emis</b> : suppression d' l'utilisateur
Modifier les privilèges	Administrateur	<b>Reçus</b> : informations d'un utilisateur <b>Emis</b> : modifier les privilèges

Tableau 3 : Tableau descriptif du cas d'utilisation « Gestion des utilisateurs »

#### 4.2.3. Module gestion des clients :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Gestion des clients » :

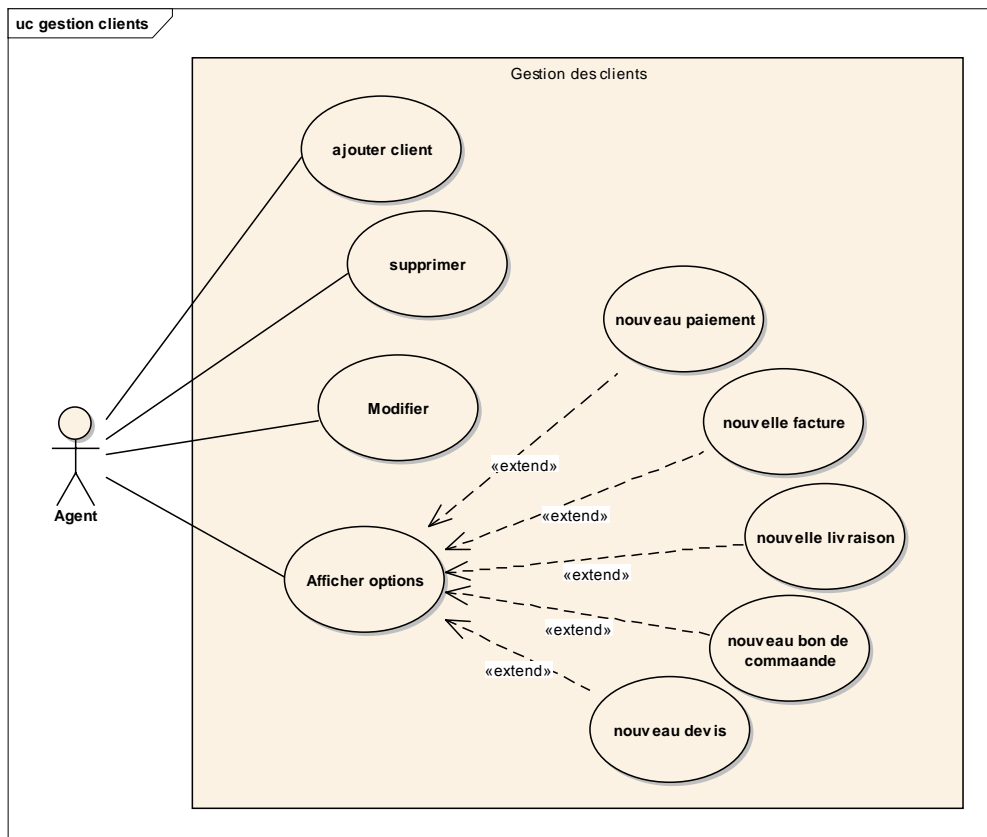


Figure 16 : Diagramme de cas d'utilisation « Gestion des clients »

Le tableau 4 représente les différents cas d'utilisation avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Ajouter client	agent	<b>Reçus</b> : formulaire d'ajout <b>Emis</b> : nouveau client ajouté
Supprimer	agent	<b>Reçus</b> : message de confirmation <b>Emis</b> : client supprimé
Modifier	agent	<b>Reçus</b> : informations d'un client. <b>Emis</b> : modifier les informations
Afficher les options	agent	<b>Reçus</b> : Liste des clients. <b>Emis</b> : ajouter devis, ajouter bon de commande, ajouter bon de livraison, ajouter paiement, ajouter facture.

Tableau 4 : Tableau descriptif du cas d'utilisation « Gestion des clients »

#### 4.2.4. Module gestion des devis :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Gestion des devis »:

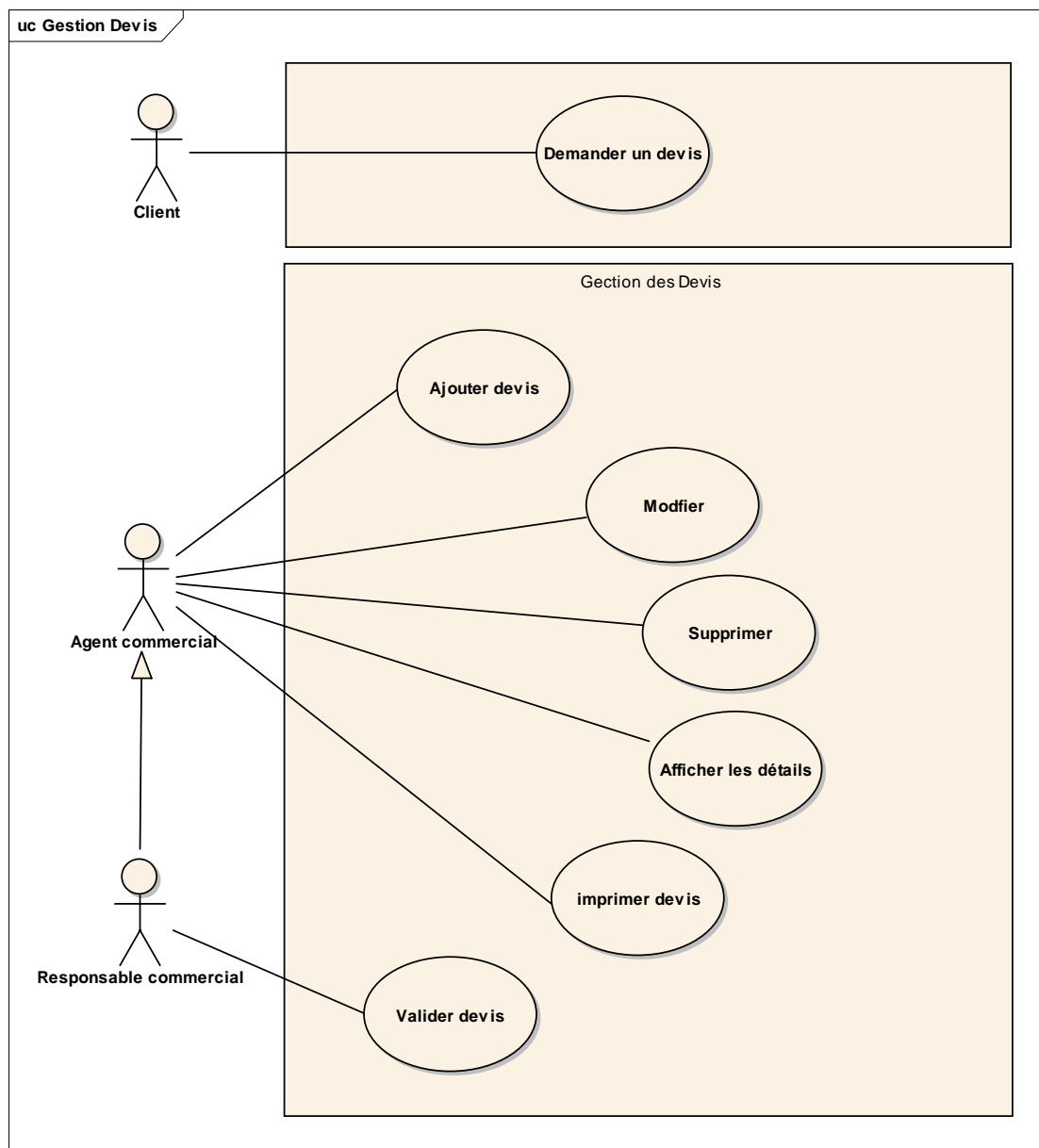


Figure 17 : Diagramme de cas d'utilisation « Gestion des devis »

Le tableau 5 représente les différents cas d'utilisation avec les messages reçus et émis:

Cas d'utilisation	Acteurs	Messages reçus/émis
Demander un devis	Client	<b>Reçus :</b> besoin d'un devis <b>Emis :</b> demande de devis

Ajouter devis	Agent commercial	<b>Reçus</b> : liste des clients, formulaire d'ajout <b>Emis</b> : nouveau devis ajouté
Modifier	Agent commercial	<b>Reçus</b> : informations d'un devis <b>Emis</b> : devis modifié
Supprimer	Agent commercial	<b>Reçus</b> : message de confirmation <b>Emis</b> : devis supprimé
Afficher les détails	Agent commercial	<b>Reçus</b> : Liste des devis <b>Emis</b> : informations du devis sélectionné
Imprimer devis	Agent commercial	<b>Reçus</b> : Liste des devis <b>Emis</b> : impression devis sélectionné
Valider un devis	Responsable commercial	<b>Reçus</b> : formulaire de modification <b>Emis</b> : modifier l'état du devis

Tableau 5 : Tableau descriptif du cas d'utilisation « Gestion des devis »

#### 4.2.5. Module gestion des bons de commande :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Gestion bon de commande » :



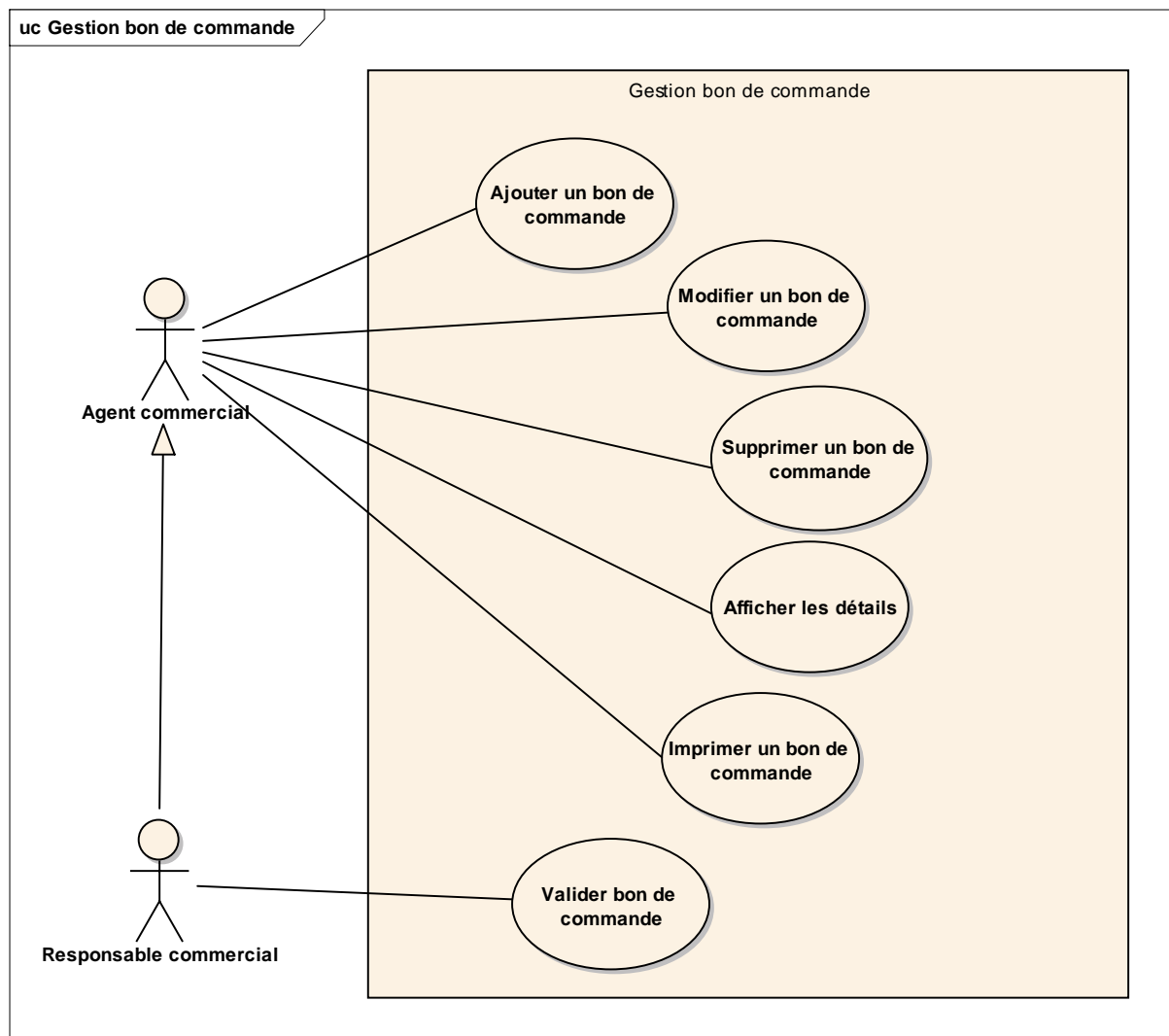


Figure 18 : Diagramme de cas d'utilisation « Gestion des bons de commandes »

Le tableau 6 représente les différents cas d'utilisation avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Ajouter un bon de commande	Agent commercial	<b>Reçus</b> : liste des devis, formulaire d'ajout <b>Emis</b> : nouveau bon de commande ajouté
Modifier	Agent commercial	<b>Reçus</b> : informations d'un bon de commande <b>Emis</b> : bon de commande modifié
Supprimer	Agent commercial	<b>Reçus</b> : message de confirmation <b>Emis</b> : bon de commande supprimé
Afficher les détails	Agent commercial	<b>Reçus</b> : Liste des bons de commandes

		<b>Emis</b> : informations du bon de commande sélectionné
Imprimer bon de commande	Agent commercial	<b>Reçus</b> : Liste des bons de commandes <b>Emis</b> : impression du bon de commande sélectionné
Valider un bon de commande	Responsable commercial	<b>Reçus</b> : formulaire de modification <b>Emis</b> : modifier l'état du bon de commande

Tableau 6 : Tableau descriptif du cas d'utilisation « Gestion des bons de commandes »

#### 4.2.6. Module gestion des bons de livraisons :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Gestion bon de livraison » :

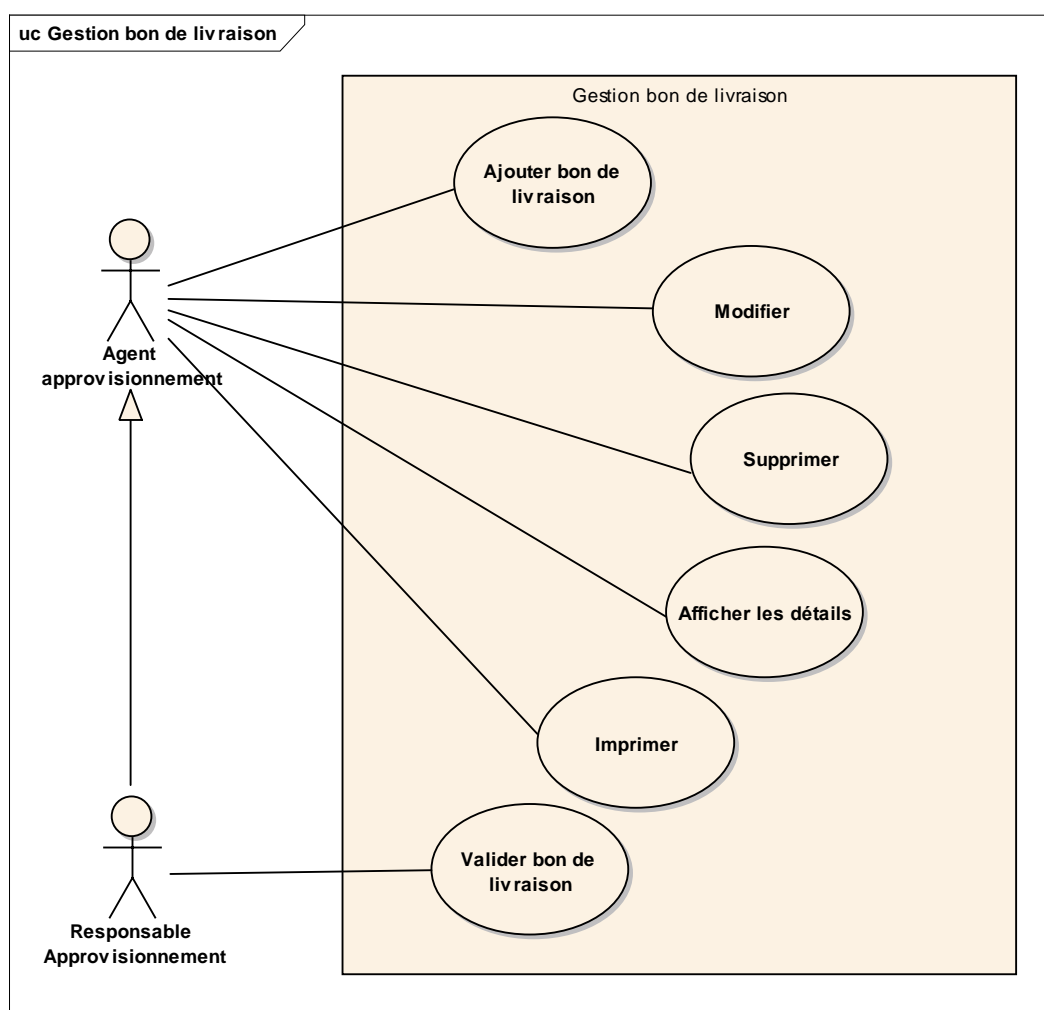


Figure 19 : Diagramme de cas d'utilisation « Gestion des bons de livraisons »

Le tableau 7 représente les différents cas d'utilisation avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Ajouter un bon de livraison	Agent Approvisionnement	<b>Reçus</b> : liste des bons de commandes, formulaire d'ajout <b>Emis</b> : nouveau bon de livraison ajouté
Modifier	Agent Approvisionnement	<b>Reçus</b> : informations d'un bon de livraison <b>Emis</b> : bon de livraison modifié
Supprimer	Agent Approvisionnement	<b>Reçus</b> : message de confirmation <b>Emis</b> : bon de livraison supprimé
Afficher les détails	Agent Approvisionnement	<b>Reçus</b> : Liste des bons de livraisons <b>Emis</b> : informations du bon de livraison sélectionné
Imprimer bon de livraison	Agent Approvisionnement	<b>Reçus</b> : Liste des bons de livraison <b>Emis</b> : impression du bon de livraison sélectionné
Valider un bon de livraison	Responsable Approvisionnement	<b>Reçus</b> : formulaire de modification <b>Emis</b> : modifier l'état du bon de livraison

Tableau 7 : Tableau descriptif du cas d'utilisation « Gestion des bons de livraisons »

#### 4.2.7. Module gestion de stock :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Gestion de stock » :

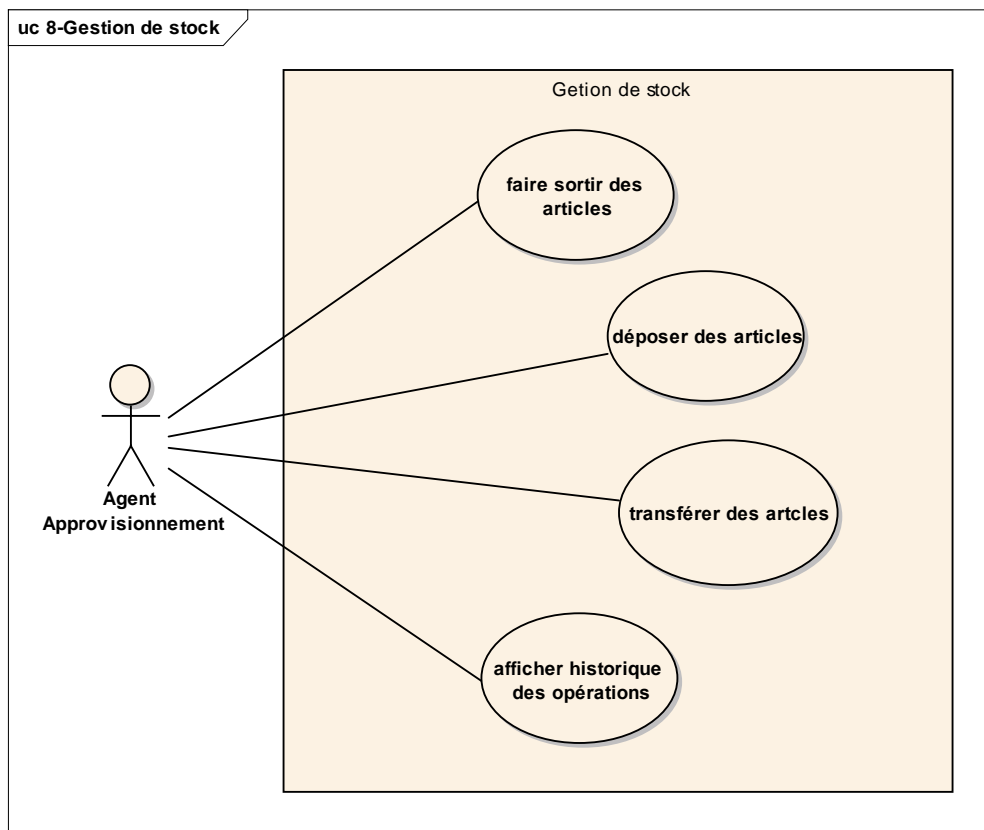


Figure 20 : Diagramme de cas d'utilisation « Gestion de stock »

Le tableau 8 représente les différents cas d'utilisation avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Faire sortir les articles	Agent Approvisionnement	<b>Reçus</b> : Liste des articles <b>Emis</b> : dépôt ajouté
Déposer des articles	Agent Approvisionnement	<b>Reçus</b> : Liste des articles <b>Emis</b> : retrait d'un article
Transférer des articles	Agent Approvisionnement	<b>Reçus</b> : Liste des articles <b>Emis</b> : article transféré entre deux magasins.
Afficher historique	Agent Approvisionnement	<b>Reçus</b> : Listes des opérations <b>Emis</b> : consulter les informations

Tableau 8 : Tableau descriptif du cas d'utilisation « Gestion de stock »

#### 4.2.8. Module Facturation :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Facturation » :

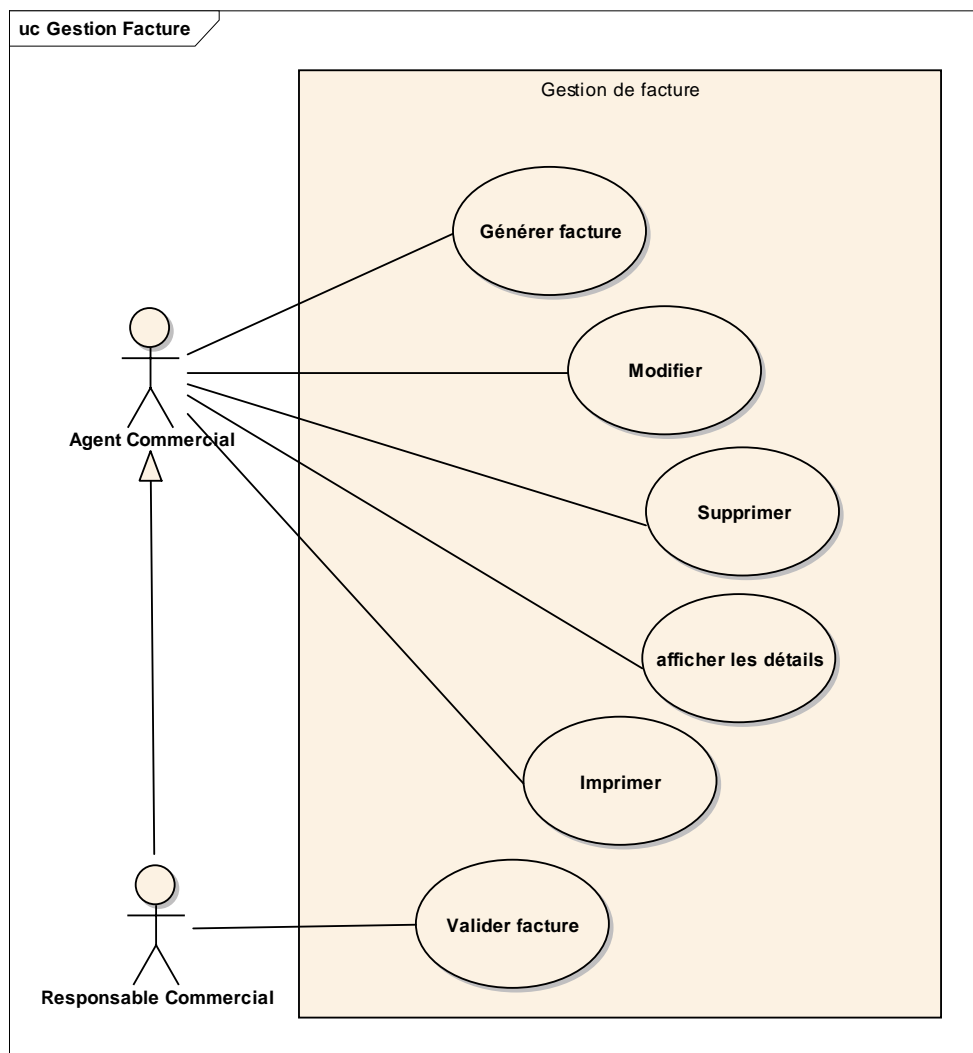


Figure 21 : Diagramme de cas d'utilisation « Facturation »

Le tableau 9 représente les différents cas d'utilisation avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Générer facture	Agent Commercial	<b>Reçus</b> : liste des bons de réceptions, formulaire d'ajout <b>Emis</b> : nouvelle facture créée
Modifier	Agent Commercial	<b>Reçus</b> : informations d'une facture <b>Emis</b> : facture modifiée

Supprimer	Agent Commercial	<b>Reçus</b> : message de confirmation <b>Emis</b> : facture supprimée
Afficher les détails	Agent Commercial	<b>Reçus</b> : Liste des factures <b>Emis</b> : informations de facture sélectionnée
Imprimer facture	Agent Commercial	<b>Reçus</b> : Liste des factures <b>Emis</b> : impression de la facture sélectionnée
Valider facture	Responsable Commercial	<b>Reçus</b> : liste des factures <b>Emis</b> : valider la facture

Tableau 9 : Tableau descriptif du cas d'utilisation « Facturation »

#### 4.2.9. Module Paiement :

Le diagramme des cas d'utilisation suivant donne une vue global sur le module « Paiement » :

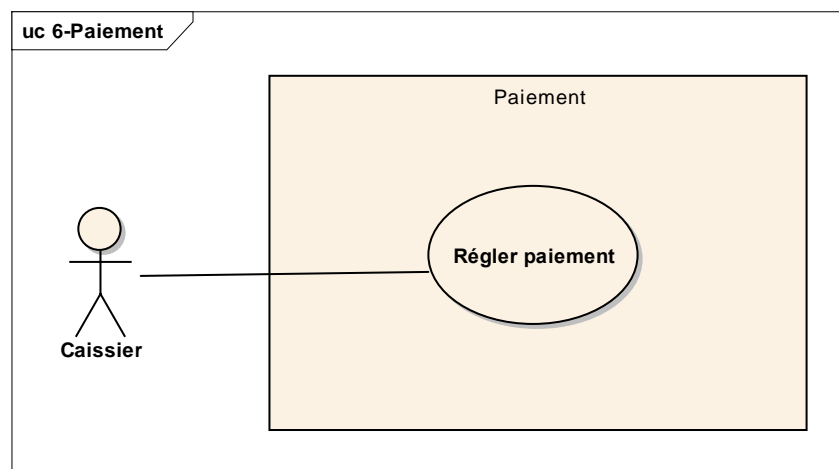


Figure 22 : Diagramme de cas d'utilisation « Paiement »

Le tableau 10 représente les différents cas d'utilisation avec les messages reçus et émis :

Cas d'utilisation	Acteurs	Messages reçus/émis
Régler paiement	Caissier	<b>Reçus</b> : Montants de paiement <b>Emis</b> : marquer la facture comme payées

Tableau 10 : Tableau descriptif du cas d'utilisation « Paiement »

### **4.3. Spécifications non-fonctionnelles :**

Les besoins non fonctionnels de notre système sont les suivants :

#### **Disponibilité :**

L'application constitue le cœur de l'activité de la gestion commerciale, il est indispensable que cette dernière soit disponible à tout moment.

#### **Extensibilité :**

L'extensibilité est l'une des spécifications importantes d'un ERP. L'architecture doit supporter les extensions de nouvelles fonctionnalités sans pour autant la modifier énormément. Le code devra être ferme à la modification et ouvert à l'extension.

#### **Sécurité :**

L'ERP doit respecter certaines règles relatives à la sécurité des systèmes informatiques, nous devons avoir un système d'accès sécurisé basé sur l'authentification et la gestion des autorisations:

#### **Performance :**

La performance des services offerts est critique, notamment pour l'importance du facteur temps dans l'ERP qui vise un grand nombre d'utilisateurs.

#### **Portabilité :**

L'application doit être portable sur tous les environnements logiciels (Windows, Mac OS, Linux).

#### **Documentation :**

Lors de la livraison de l'application, nous devons fournir la documentation nécessaire pour les utilisateurs finaux ainsi que les futurs développeurs.

#### **Utilisabilité :**

Tous les standards d'ergonomies doivent être présents : interface utilisateur bien claire et simple dans l'utilisation.

### **Conclusion :**

Ce chapitre a été consacré à l'analyse et à la spécification des besoins fonctionnels et non-fonctionnels de la solution. Dans le prochain chapitre nous entamons la conception de la solution.

---

## **Chapitre 5 : Architecture et conception de la solution**

---



## Introduction :

Dans ce chapitre, nous exposons d'abord l'architecture technique et l'architecture applicative de la solution, avant d'entamer ensuite la phase de conception.

### 5.1. Architecture :

L'architecture est l'ensemble des aspects techniques et applicatifs qui sont importants pour un logiciel. Les choix architecturaux influent sur la réussite ou l'échec d'un projet. Nous exposons d'abord l'architecture technique cible de la solution ainsi que l'architectures applicative.

#### 5.1.1. Architecture technique :

L'architecture technique est l'environnement technique permettant l'exécution des composants informatiques et les échanges de données. Pour notre application, nous proposons l'architecture cible suivante :

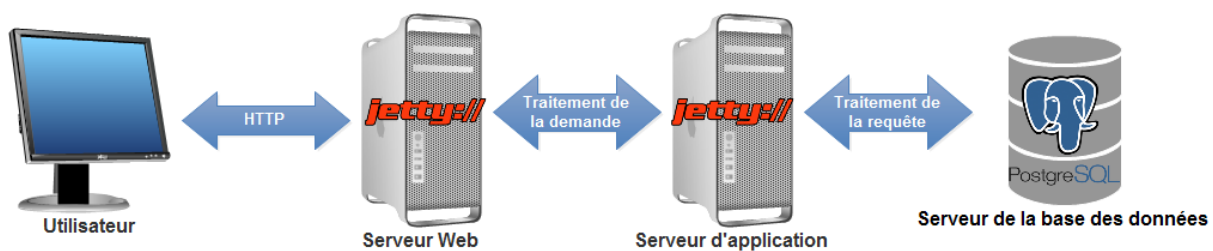


Figure 23 : Architecture technique de l'application

L'utilisateur se connecte à l'application à travers un navigateur web. Le navigateur permet d'envoyer des requêtes au serveur Web et d'en interpréter la réponse. Le navigateur et le serveur communiquent en utilisant le protocole http. La fonction du serveur Web étant d'envoyer le contenu des fichiers à l'utilisateur, des extensions peuvent y être ajoutées, permettant de faire appel à des services pour générer dynamiquement les informations à transmettre. Le serveur d'application traite les requêtes HTTP, interprète et exécute le code de l'application, puis génère une réponse qu'il renvoie au serveur web qui l'enverra au navigateur de l'utilisateur. Nous utilisons Jetty comme étant à la fois un serveur web et un serveur d'application. Le système de gestion de bases de données PostgreSQL permet d'interroger les données et de les mettre à jour.

#### 5.1.2. Architecture applicative :

Nous pouvons définir l'architecture applicative comme une organisation des données et des traitements qui mettent en œuvre les fonctions métiers. La figure 29 présente l'architecture applicative de la solution avec les frameworks utilisés.

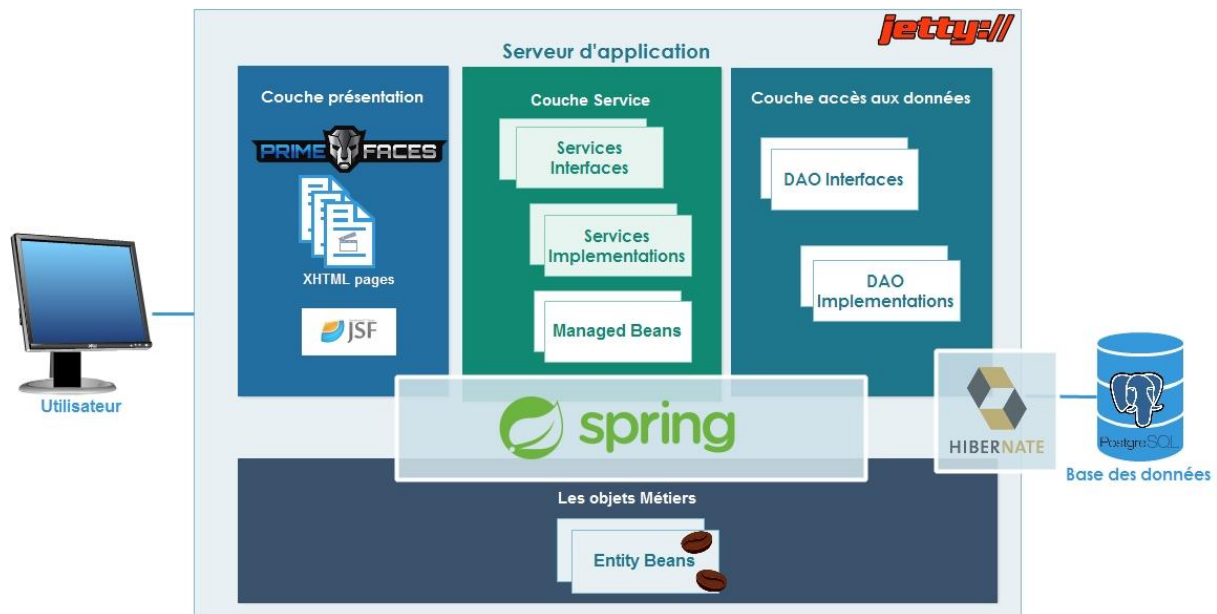


Figure 24 : Architecture applicative de l'application

Notre solution est une application n-tiers. Ses couches sont:

- **La couche accès aux données** sert à permettre à l'application d'interagir avec ses sources de données. il est important que cette couche soit la plus rapide possible. Pour cela, nous utilisons Hibernate [10] qui est un framework de persistance.
- **La couche service** contient l'ensemble du code métier de l'application. Elle organise et orchestre les accès à la couche accès aux données et ses aspects transactionnels. Ces différents aspects sont gérés par le framework Spring [11] qui est utilisé comme un framework d'intégration. Il implémente le patron IoC (Inversion de contrôle)
- **La couche présentation** est la couche d'entrée dans l'application du point de vue de l'utilisateur. Elle appelle les traitements de la couche Service en fonction des actions effectuées par l'utilisateur et récupère les données retournées. Elle met ensuite en forme ces données pour les afficher à l'utilisateur. Cette couche est réalisée grâce au Primefaces [12]

Par ailleurs les **objets métiers** sont utilisés contiennent les classes persistantes utilisés par les différentes couches de l'application.

## 5.2. Conception :

### 5.2.1. Patrons utilisés :

Un patron est un ensemble des pratiques éprouvées et considérées comme bonnes. C'est un moyen efficace de partager les connaissances. Le patron est une formalisation de solutions courantes à des problèmes récurrents. Dans notre projet, nous utilisons un patron d'accès aux données et un patron pour l'intégration des couches.

#### 5.2.1.1. Patron Accès aux données (DAO) :

Le DAO (Data Access Object) est un patron de conception qui découple entièrement le modèle objet et le modèle de donnée et permet d'encapsuler le traitement d'accès aux données. [13]

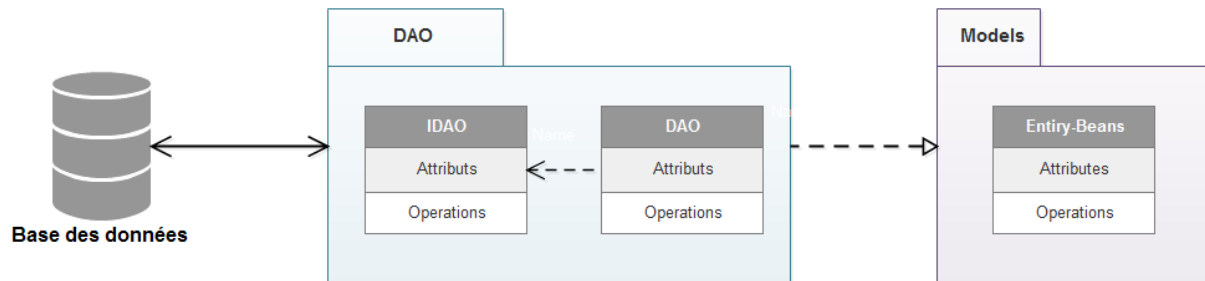


Figure 25 : Modèle d'accès aux données (DAO)

Pour implémenter le modèle d'accès aux données, nous utilisons le framework Hibernate. Pour chaque entité nous définissons une classe permettant de faire les opérations CRUD (Search, Create, Read, Update, Delete) sur chaque table de la base de données.

#### 5.2.1.2. Patron Inversion de contrôle (IoC):

L'inversion de contrôle (Inversion of Control, IoC) [14] est un patron d'architecture. Il fonctionne selon le principe que le flot d'exécution n'est plus sous le contrôle direct de l'application elle-même mais du framework. Le patron IoC assure l'intégration entre la couche d'accès aux données, la couche service et la couche présentation. Chaque couche expose des interfaces pour la couche supérieure. La figure 26 explique cette intégration :

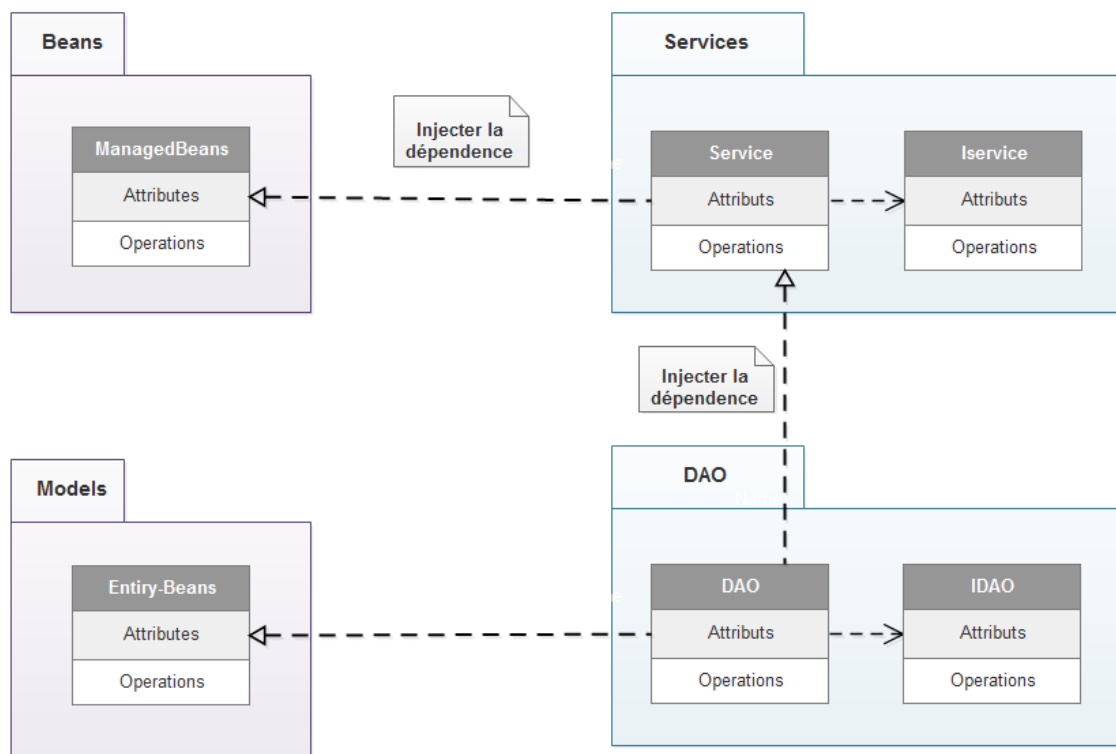


Figure 26 : Modèle d'intégration des couches (IoC)

Pour assurer l'intégration entre les différentes couches de l'application, nous adoptons le framework Spring qui implémente le patron IoC. Il fournit un conteneur d'injection pour gérer les dépendances entre les couches.

### 5.2.2. Conception des modèles de données :

Notre application nécessite que les différentes données doivent être stockées en permanence dans une base des données pour assurer le bon fonctionnement. Dans la suite, nous présentons les différentes tables de notre modèles de chaque module.

#### 5.2.2.1. Module « Gestion des clients » :

Nous présentons les tables des données du module gestion des clients dans la figure 27.

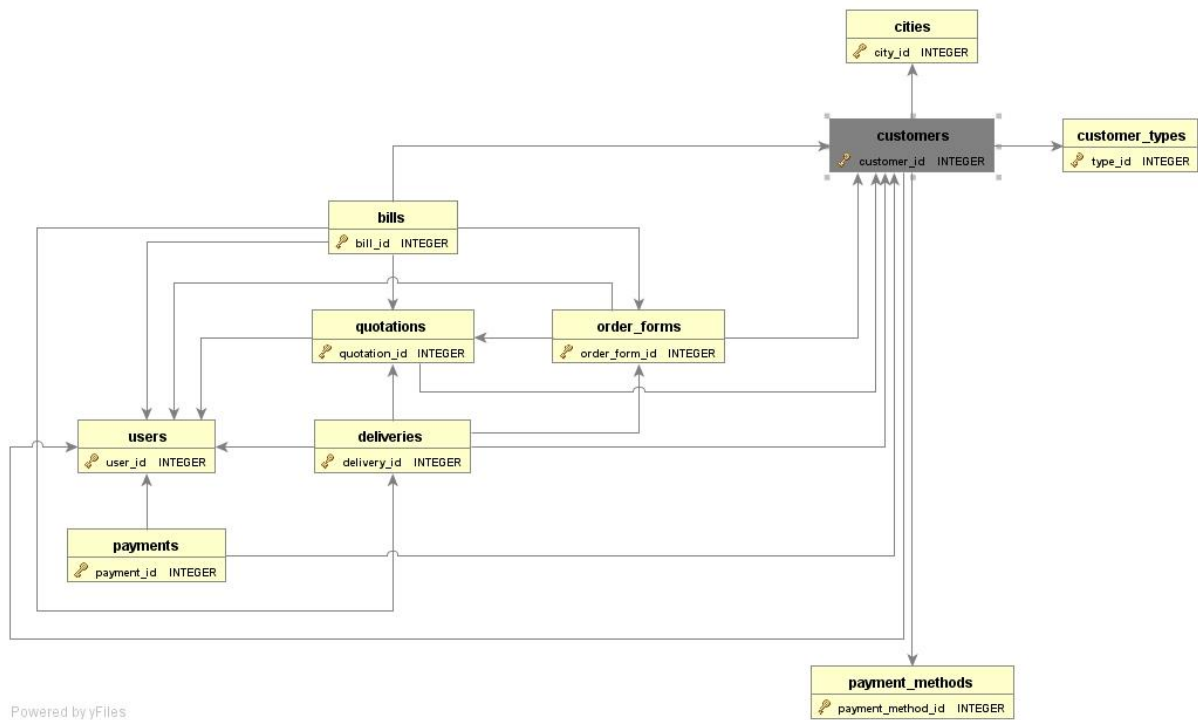


Figure 27 : Modèle des données pour le module de la gestion des clients

La table client (customers) contient les informations d'un client comme sa ville, sa méthode de paiements et l'identifiant de l'utilisateur qui a créé ce client. Le système offre à chaque client un code unique. Ce code est utilisé pour référencer ses devis, ses bons de commandes, ses bons de livraisons, ses factures et son historique de paiements.

#### 5.2.2.2. Module « Gestion des devis » :

Nous présentons les tables des données du module gestion des devis dans la figure 28.

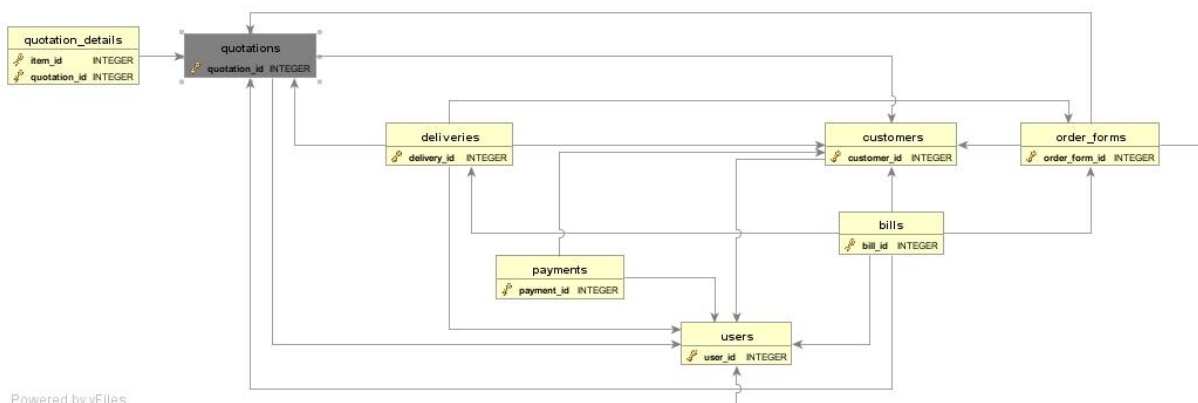


Figure 28 : Modèle des données pour le module gestion des devis

La table devis (quotations) contient les informations de devis comme le code de son client, sa date de validité, son état et l'identifiant de l'utilisateur qui lui a créé. Les détails de chaque devis

sont stockés dans la table détails de devis (quotation\_details). Le système génère un code unique pour chaque nouveau devis. Ce code sera l'identifiant de ce devis dans les bons de commandes, les livraisons, les factures et les paiements.

#### 5.2.2.3. Module « Gestion des bons de commandes » :

Nous présentons les tables des données du module gestion des bons de commandes dans la figure 29.

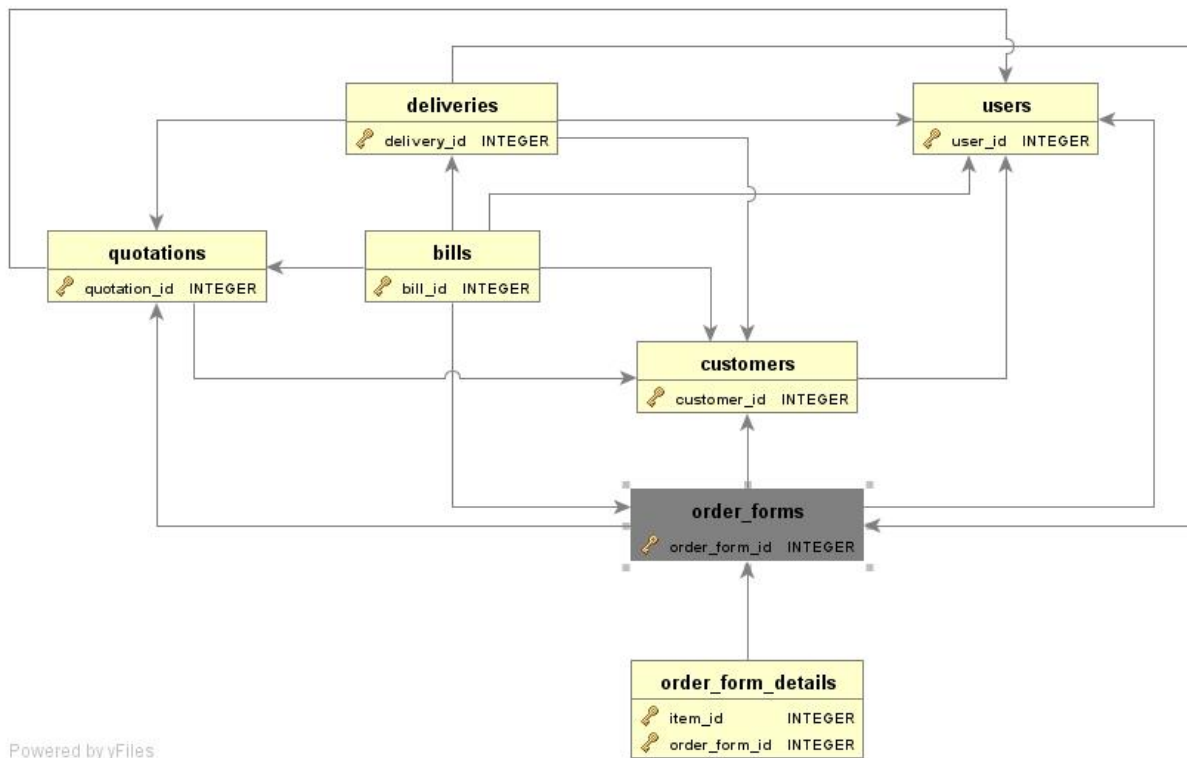


Figure 29 : Modèle des données pour le module gestion des bons de commandes

La table bon de commande (order\_forms) contient les informations d'un bon de commande comme le code de devis, l'utilisateur qui lui a créé, l'état du bon de commande et le code du client de ce commande. Les détails du bon de commande sont stockés dans la table à part. Ces détails sont les informations sur les articles de chaque commande comme la quantité, le TVA, remise. Le système génère un code unique pour chaque nouveau bon de commande.

#### 5.2.2.4. Module « Gestion des bons de livraisons » :

Nous présentons les tables des données du module gestion des bons de livraisons dans la figure 30.

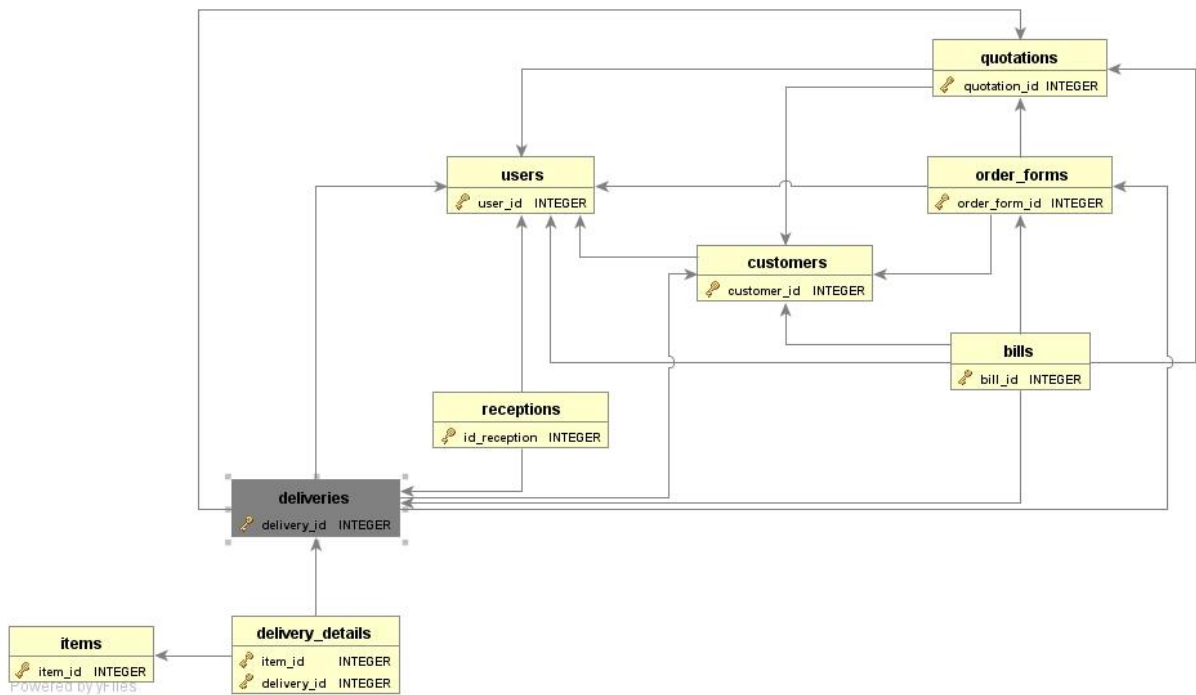


Figure 30 : Modèle des données pour le module gestion des bons de livraisons

La table livraison (*deliveries*) contient les informations d'un bon de livraison comme la date de livraison, l'état de bon de livraison, le code de commande, le code de son client, le code de devis et le code de l'utilisateur qui lui a créé. Les détails sur les différents articles de cette livraison sont stockés dans la table détails de livraison. Le system génère un code unique pour chaque nouvelle livraison.

#### 5.2.2.5. Module « Gestion des bons de réceptions » :

Nous représentons les tables des données du module gestion des bons de réceptions dans la figure 31.

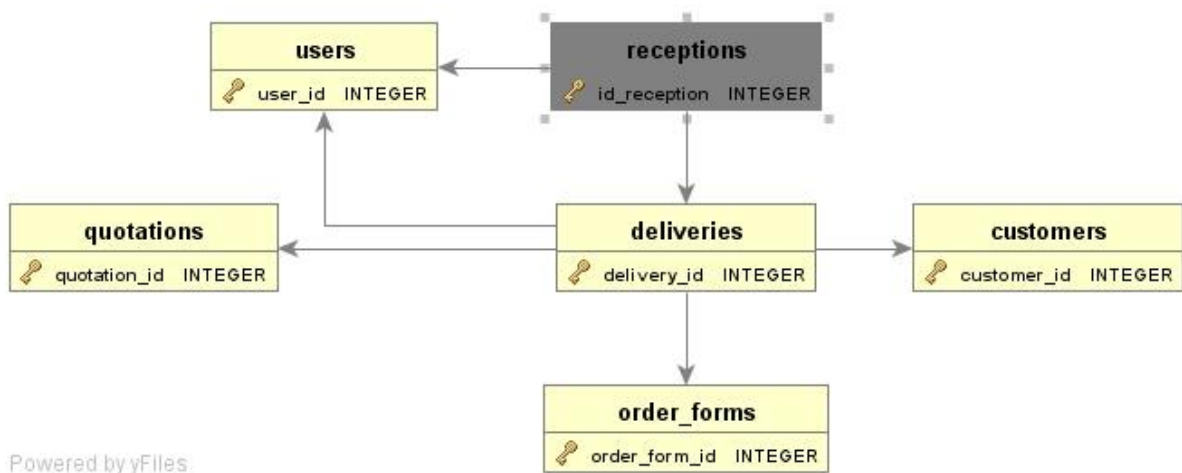


Figure 31 : Modèle des données pour le module gestion des bons de réceptions

La table bon de réception (receptions) contient les informations d'un bon de réception comme la date de création, l'état du bon de réception, la date de validation, le code de livraison et le code de l'utilisateur qui a créé. Le bon de réception est une image sur le bon de livraison. Le système génère un code unique pour chaque nouveau bon de réception. Ce code référence ce bon de réception pour garder une trace.

#### 5.2.2.6. Module « Gestion de stock » :

Nous présentons les tables des données du module gestion de stock dans la figure 32.

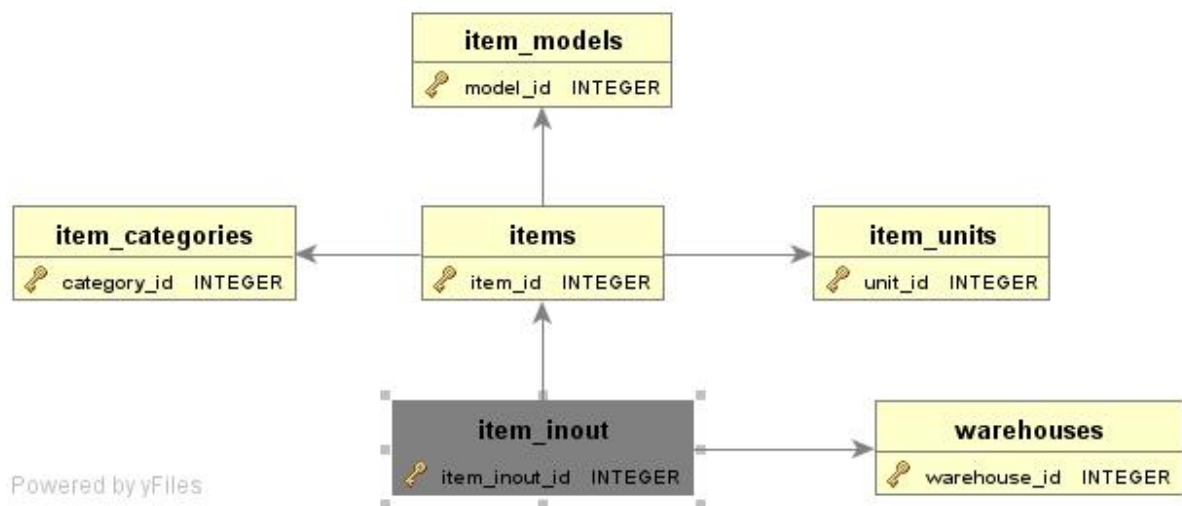


Figure 32 : Modèle des données pour le module gestion de stock

La table stock (item\_inout) contient les informations de opérations de stock comme le type d'opération (entrée, sortie, transfert), l'identifiant du magasin et l'identifiant de l'article.



#### 5.2.2.7. Module « Facturation » :

Nous présentons les tables des données du module de facturation dans la figure 33.

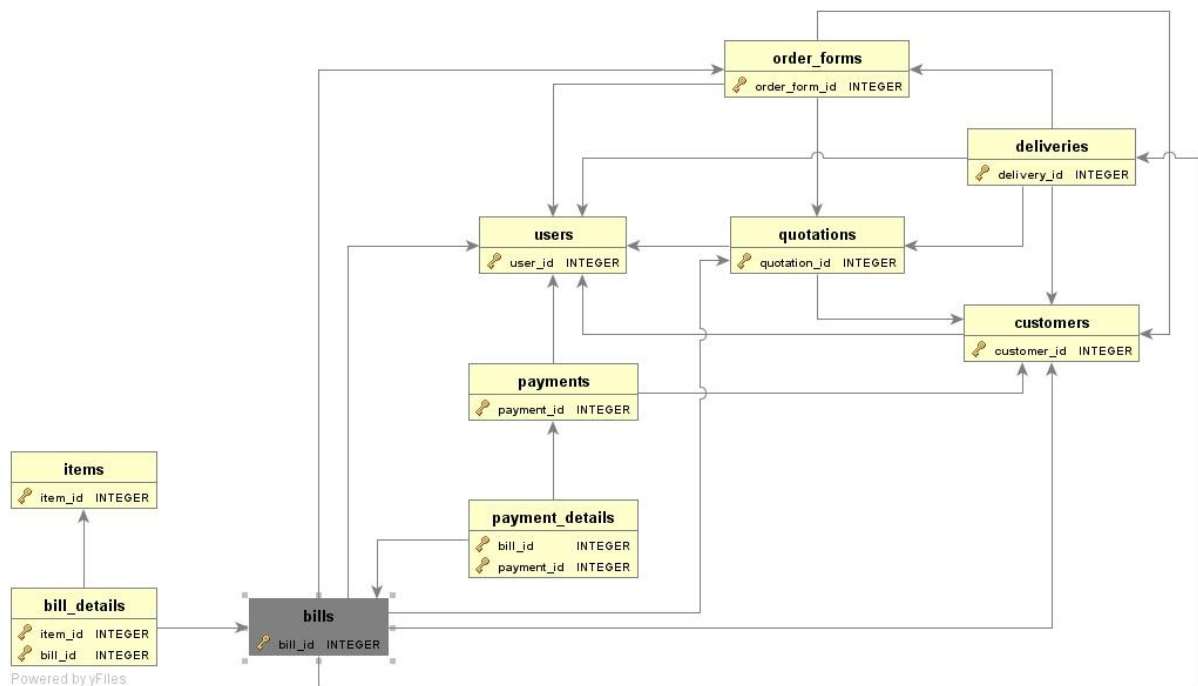


Figure 33 : Modèle des données pour le module de facturation

La table facture (bills) contient les informations d'une facture comme la date de création, la somme totale, le code de livraison, le code de commande, le code de client, le code de devis, l'état de facture et le code de l'utilisateur qui a créé. La table détails de facture (Bills\_details) contient les articles de cette facture. Le système génère un code unique pour chaque nouvelle facture. Ce code référence cette facture dans la table de paiement.

#### 5.2.2.8. Module « Paiement » :

Nous présentons les tables des données du module gestion de paiement dans la figure 34.

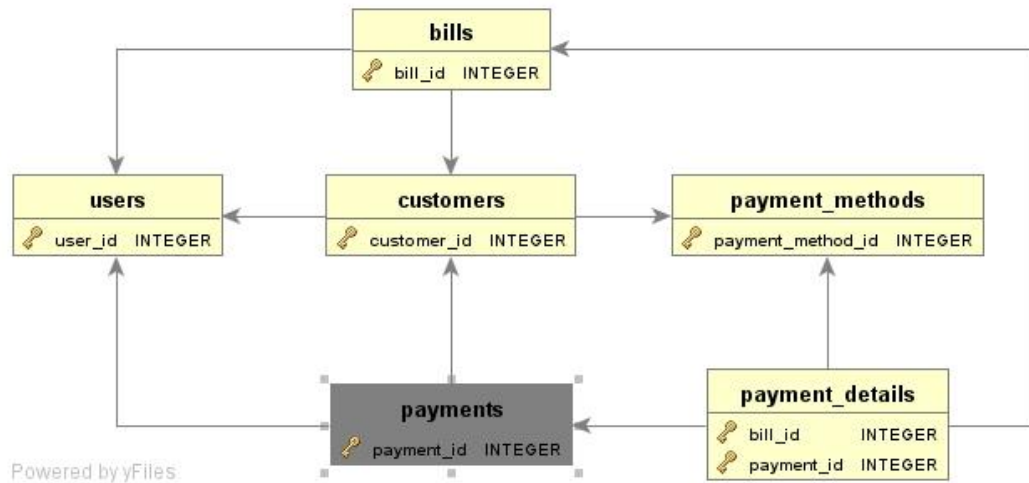


Figure 34 : Modèle des données pour le module gestion de paiement

La table paiement (payments) contient les informations d’une opération de paiement comme la date de paiement, le code de client et le code de l’utilisateur qui a créé. La table détails de paiement (Payment\_details) est table d’association entre le paiement, la facture et la méthode de paiement de la facture. Le système génère un code unique pour chaque nouveau paiement.

#### 5.2.2.9. Module « Clôture de la caisse » :

Nous présentons les tables des données du module clôture de la caisse dans la figure 35.

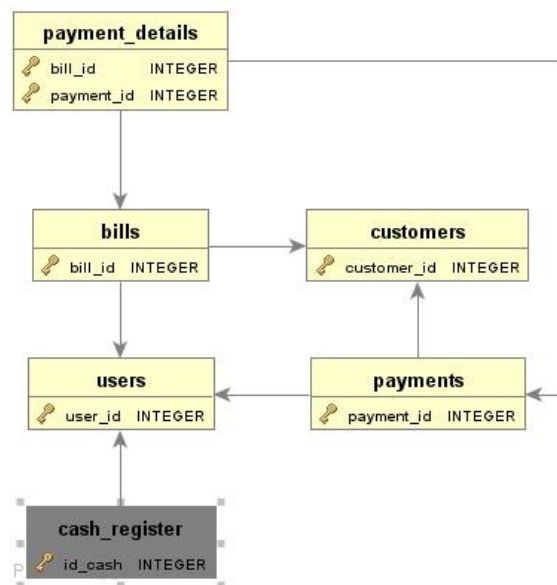


Figure 35 : Modèle des données pour le module clôture de la caisse

La table caisse (cash\_register) contient des informations concernant le revenu quotidien. Elle contient un champ date, un champ revenu et un code qui référence l'utilisateur qui a effectué l'opération de la clôture.

#### 5.2.2.10. Module « Gestion des utilisateurs » :

Nous présentons les tables des données du module gestion des utilisateurs dans la figure 36.

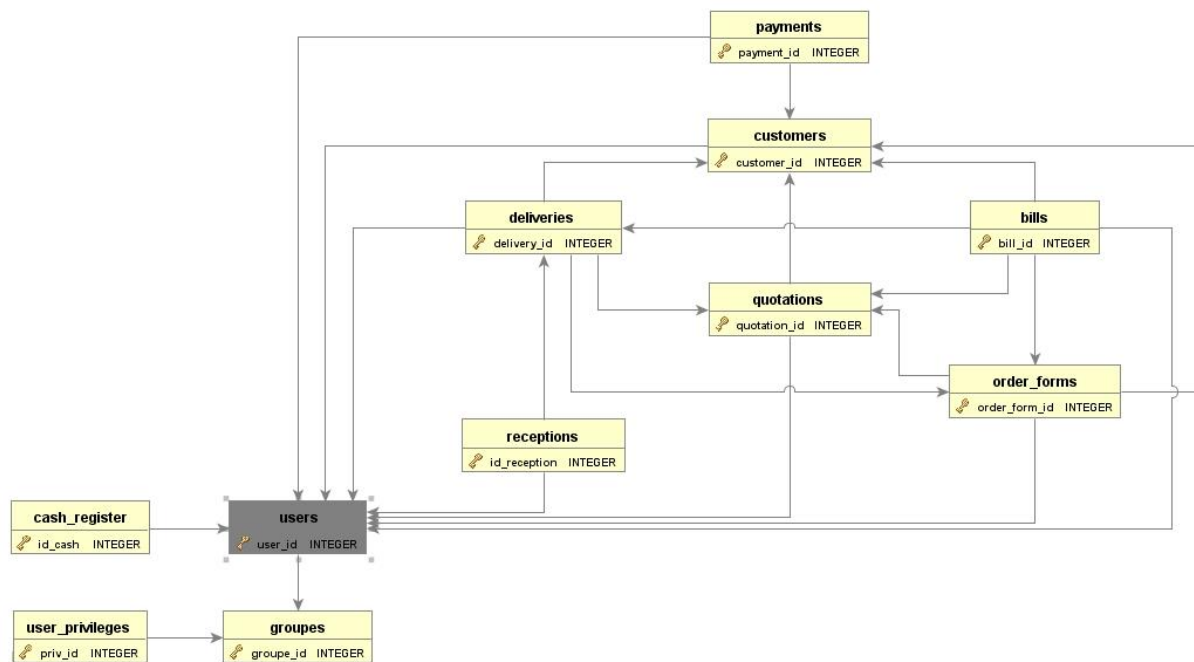


Figure 36 : Modèle des données pour le module gestion des utilisateurs

La table utilisateur (user) contient les informations d'un utilisateur de l'application comme l'identifiant d'un groupe. Chaque groupe contient plusieurs privilèges pour gérer les rôles des utilisateurs. Le système génère un code unique pour chaque nouvel utilisateur. Ce code sera la référence dans les différentes opérations effectuées par cet utilisateur.

## Conclusion :

Ce chapitre a été consacré à la description des architectures et à la conception de la solution, à travers la présentation de patrons utilisés et des modèles de données. Dans le chapitre suivant nous détaillons les étapes de la réalisation de l'application.

---

## **Chapitre 6 : Implémentation de la solution**

---

## Introduction :

Nous commençons ce chapitre par la présentation de l'environnement matériel et logiciel utilisé, par la suite, nous décrivons les itérations (Sprints) de réalisation de l'application.

### 6.1. Environnement de travail :

#### 6.1.1. Environnement matériel :

L'environnement matériel, a été mis en place pour le développement qui est un PC HP. Ce dernier dispose de la configuration suivante :

- Processeur : Intel Core i3 2330M @ 2.20GHz.
- RAM: 6.00 Go Canal-Double DDR3 @ 665 MHz.
- Système d'exploitation : Windows 7 64-bit SP1.

#### 6.1.2. Environnement logiciel :

L'environnement logiciel utilisé pour réaliser notre projet est structuré dans le tableau 11:

Outils	Description
JDK 1.7	Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java.
Eclipse Kepler	Environnement de développement intégré (IDE)
Apache Maven 3.2.5	Apache Maven est un outil pour la gestion et l'automatisation de production des projets logiciels Java en général et Java EE en particulier. [15]
Jetty 8.1	Jetty est un serveur HTTP et un moteur de servlet entièrement fondé sur la technologie Java. [16]
JSF 2.2.5	Java Server Faces est un framework de développement d'applications Web en Java permettant de respecter le modèle d'architecture MVC [17]
Primefaces 4.0	PrimeFaces est une puissante librairie de composants JSF qui s'appuie sur la librairie Ajax jQuery. [12]

Spring 3.1.1	Spring est un framework libre pour construire et définir l'infrastructure d'une application Java EE [11]
Hibernate 4.1	Hibernate est un framework open source. Il gère la persistance des objets en base de données relationnelle. [10]
PostgreSQL 9.3	PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). [18]
PgAdmin III	PgAdmin est un outil d'administration graphique pour PostgreSQL [19]
DbVisualizer 9.2.1	DbVisualizer est un outil de visualisation des modèles des données qui accèdent à une large gamme des bases des données. [20]
Entreprise Architect 7.5	Entreprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. [21]
ArisExpress 2.4	ArisExpress est un outil libre de modélisation des processus métiers. Il prend en charge plusieurs notations telles que le BPMN. [22]

Tableau 11 : Description de l'environnement logiciel utilisé

## 6.2. Implémentation et tests :

Nous avons réparti le travail en plusieurs itérations (Sprints). Le sprint est un bloc de temps durant lequel un incrément du produit sera réalisé. Tous les sprints ont une durée constante et ne se chevauchent jamais. Tout au long de cette partie, nous traitons les sprints pour produire un incrément potentiellement livrable. Le tableau 12 représente les sprints (en jours) de notre projet :

Sprint	Durée	Sprint	Durée
Gestion de stock	14	Gestion des livraisons	7
Gestion des utilisateurs	7	Gestion des bons de réception	7
Gestion des devis	14	Facturation	7
Gestions des bons de commandes	7	Paiement	7

Tableau 12 : Tableau des sprints

Avant de commencer un sprint, il faut lister les tâches incluses dans le backlog du sprint. Un exemple de sprint Backlog est en annexe.

Dans un sprint nous dégageons quatre activités principales qui sont successivement : la spécification fonctionnelle, la conception, le codage et le test. Les spécifications fonctionnelles que nous avons définis dans le chapitre quatre seront décomposées en sprints. Du coup pour chaque module il y'aura un sprint. Du fait, nous avons un sprint pour la gestion de stock, un sprint pour la gestion des utilisateurs, un sprint pour la gestion des devis, un sprint pour la gestion des bons de commandes, un sprint pour la livraison, un sprint pour la gestion des bons de réceptions, un sprint pour la facturation et un sprint pour le paiement. A chaque sprint correspond à un diagramme de classes participantes qui seront implémentés dans notre application avant d'être testés.

Le diagramme de classes participantes représente les classes conçues et implémentées pour un sprint, afin de réaliser les fonctionnalités d'un module. Dans un diagramme des classes participantes, nous pouvons distinguer quatre types de classes qui sont :

- Les classes ManagedBean qui exécutent la logique métier et gèrent la navigation entre les pages.
- Les classes Services sur lesquelles nous grefferons la gestion des transactions avec la base de données
- Les classe DAOs qui accèdent à la base de données à travers un gestionnaire de session.
- Les classes Models qui représentent les table dans des modèles objets.

Nous allons reprendre le diagramme de classes participantes pour chaque sprint.

#### **6.2.1. Sprint 1 : Module gestion de stock :**

**Diagramme de classes :**

La figure 37 illustre le diagramme de classes de module « Gestion de stock ».

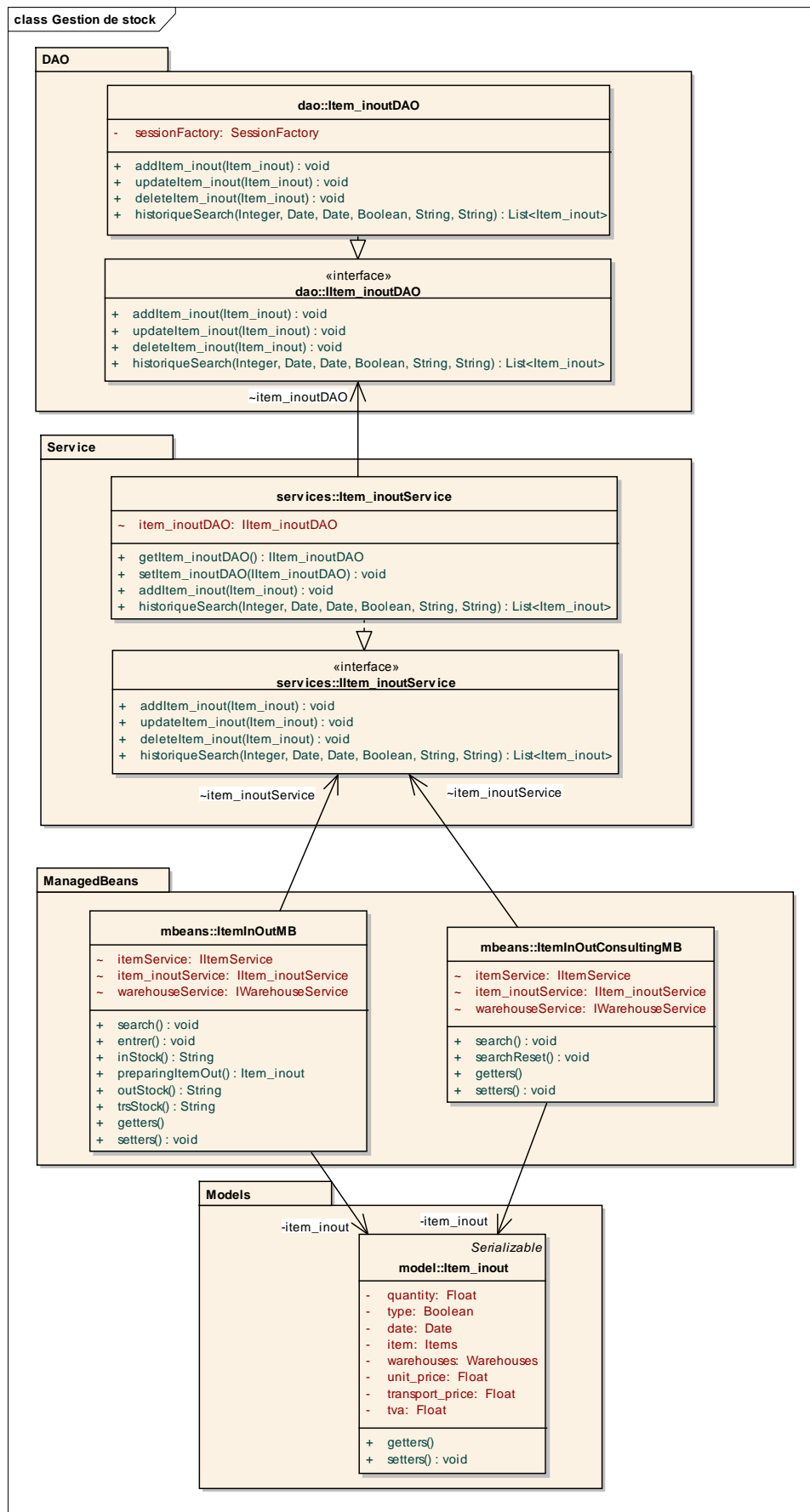


Figure 37 : Diagramme de classes de module « Gestion de stock »



Les classes principales sont «Item\_inOutMB» et «ItemIonOutConsultingMB». Le premier gère les opérations de dépôt, de retrait et de transfert des articles. Le deuxième gère les historiques des opérations.

### Test :

Les fonctionnalités offertes par le module « Gestion de stock » sont le retrait, le dépôt et le transfert des articles. La figure 38 représente l'interface de gestion des opérations de stock.

**SD - LEADER**  
Sales & Distribution System

Accueil Clients Gestion de stock Devis Bons de commande Bons de livraisons Executer Livraison Factures Paiements Caisse Paramètres m. mars

### Gérer les opérations du stock

Zone de recherche des articles

Désignation : Référence : Famille : Choisir une famille Catégorie : Choisir une catégorie

Marque : Choisir une marque Modèle : Choisir un modèle Disponibilité : choix par disponibilité

Chercher Effacer Nouvel article

**Liste des articles**  
(1 of 2)

Désignation	Référence	Catégorie	Modèle	Quantité	Unité	Disponibilité	Action
LG50d	reflqd042015	EverVivid Q50	L45-HD	11,000	pièce	Disponible	E S T
microProcesseur	refmp042015	S2000	LG50-HD	6,000	pièce	Disponible	E S T
HUAWEI	reth042015	EverVivid Q50	Alu-a	17,000	pièce	Disponible	E S T

Figure 38 : Interface de gestion des opérations de stock

La figure 39 représente l'interface de consultation de l'historique des opérations.

**SD - LEADER**  
Sales & Distribution System

Accueil Clients Gestion de stock Devis Bons de commande Bons de livraisons Executer Livraison Factures Paiements Caisse Paramètres m. mars

### Historiques des Entrées/Sorties

Zone de recherche

Désignation : Référence : Opération : Choisir Magasin : Choisir un magasin

Debut période : Fin période :

Chercher Effacer

**Liste des opérations**  
(1 of 3)

Désignation	Référence	Quantité	Unité	Prix de l'unité (TND)	Prix de transport (TND)	TVA (%)	Date	Opération	Magasin
HUAWEI	reth042015	10,0	pièce	330,000	150,000	12,00	24-05-2015	Entrée	AluMAG
microProcesseur	refmp042015	4,0	pièce	11,000	11,000	6,00	22-05-2015	Sortie	AluMAG
LG50d	reflqd042015	5,0	pièce	100,000	10,000	6,00	22-05-2015	Sortie	localMAG
HUAWEI	reth042015	9,0	pièce	330,000	150,000	12,00	22-05-2015	Sortie	localMAG

Figure 39 : Interface de l'historique des opérations

## 6.2.2. Sprint 2 : Module gestion des utilisateurs:

### Diagramme de classes:

La figure 40 illustre le diagramme de classes de module « Gestion des utilisateurs ».

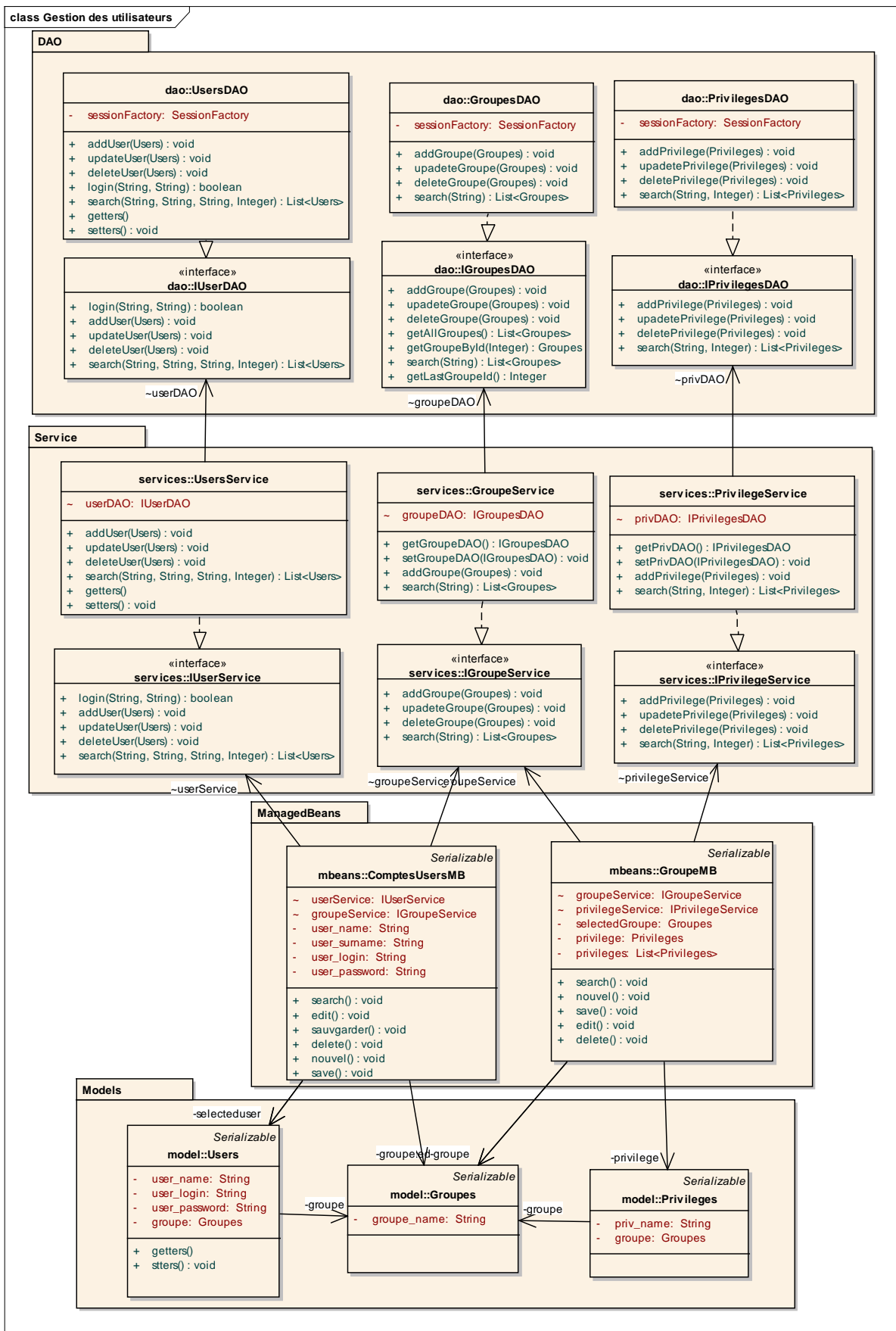


Figure 40 : Diagramme de classes de module « Gestion des utilisateurs »

La classe principale est «ComptesUsersMB». Elle gère les opérations de gestion des utilisateurs et leurs rôles. Pour gérer les rôles, il faut créer des groupes des utilisateurs qui ont les mêmes privilèges.

### Test :

L'une des fonctionnalités offerte par le module «Gestion des utilisateurs» est la gestion des groupes et leurs privilèges. La figure 41 représente l'interface de gestion des groupes.

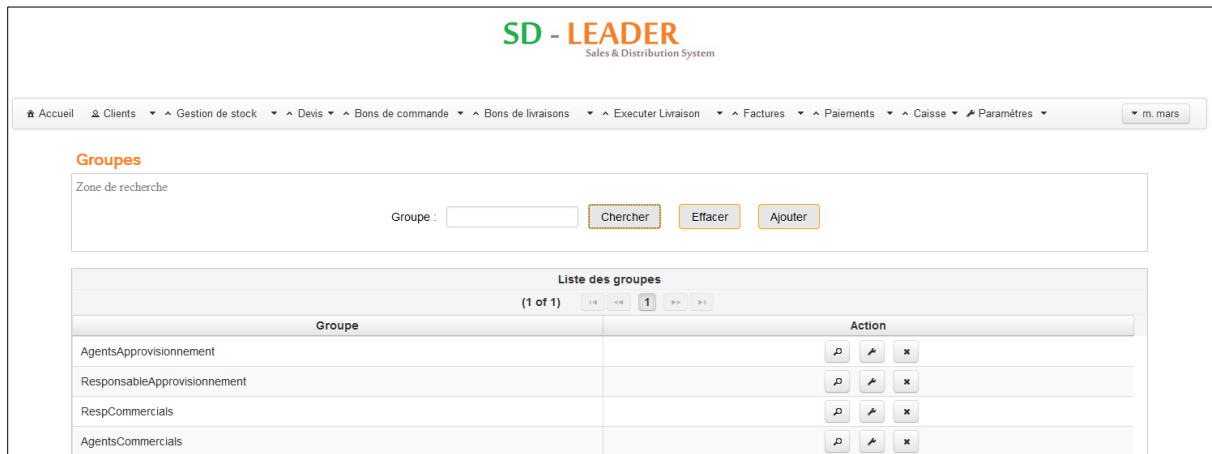


Figure 41 : Interface de gestion des groupes

Après la création d'un groupe, l'application permet d'ajouter des utilisateurs à ce groupe. La figure 42 illustre l'interface de gestion des utilisateurs.

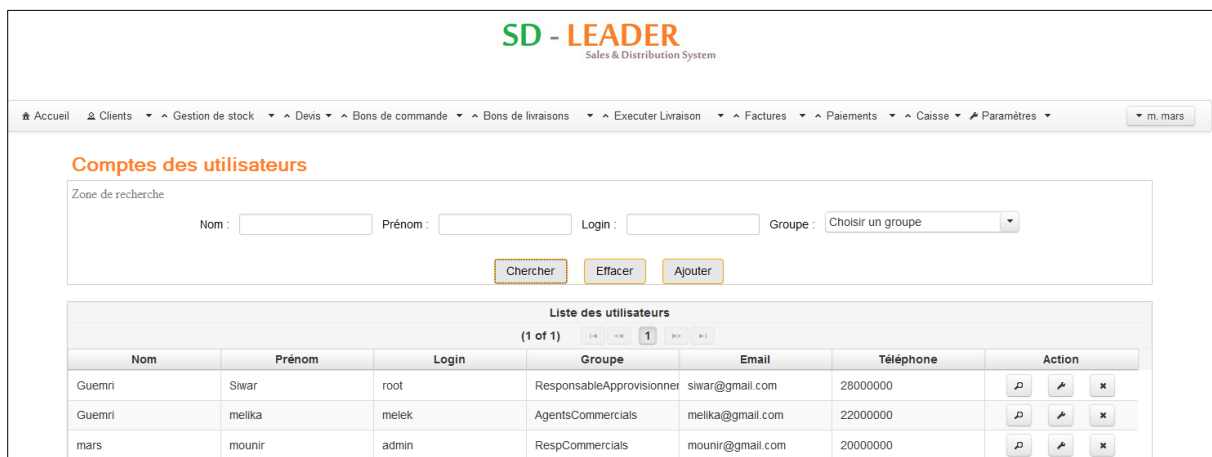


Figure 42 : Interface de gestion des utilisateurs

### 6.2.3. Sprint 3 : Module gestion des devis :

#### Diagramme de classes :

La figure 43 illustre le diagramme de classes de module « Gestion des devis ».

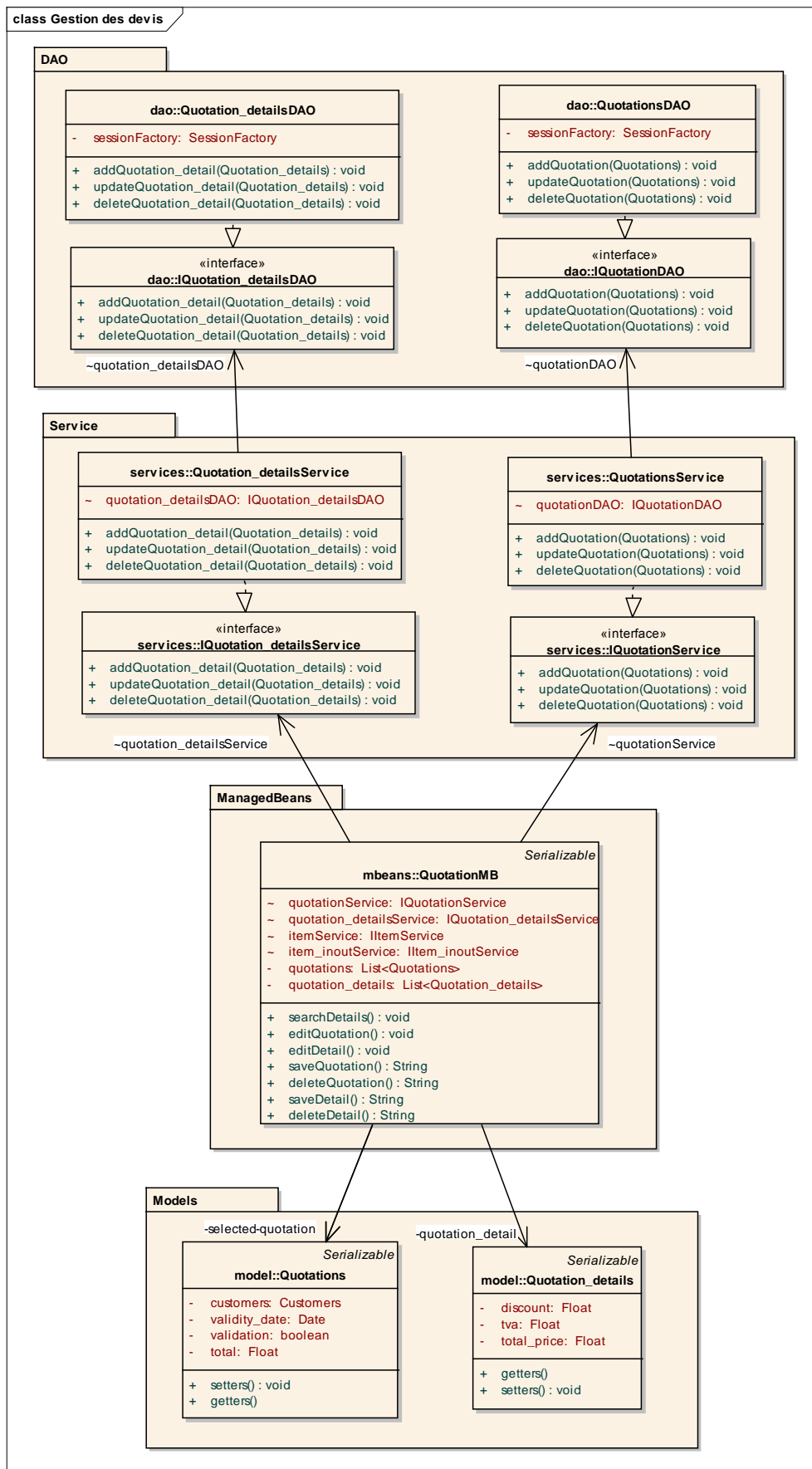


Figure 43 : Diagramme de classes de module « Gestion des devis »

La classe principale est le managedBean «QuotationsMB». Il gère les opérations de gestion des devis et leurs détails.

### Test :

La figure 44 illustre l'interface de création d'un nouveau devis.

**Nouveau devis**

Recherche des clients

Nom / Raison sociale :  Code du client :  Choisir un code Ville :  Choisir une ville Type de client :  Choisir un type

Matricule fiscal :  Registre du commerce :

**Liste des clients**  
(1 of 1)

Nom / Raison sociale	Code client	Matricule fiscal	Registre du commerce	Tél.Fixe	Tél.Mobile	Fax	Email	Action
fatma Innov	CP20	mffm21042015	rgfi21042015	750000000	54000000	75000000	fatma@gmail.com	<input type="button" value="Nouveau devis"/>
client2	CP19	MMSG120354kl	RCSG120354kbh11					<input type="button" value="Nouveau devis"/>
siwar	CP14	MMSG1203546897	RCSG1203546897	75	28	75	istabrou9a@gmail.co	<input type="button" value="Nouveau devis"/>

Figure 44 : Interface de création d'un devis

Les opérations de gestion des devis sont la recherche, la modification, la suppression, la consultation des détails et l'impression d'un devis valide. La figure 45 illustre l'interface de gestion des devis.

**Gérer les devis**

Zone de recherche

Nom de client :  Code du client :  Choisir un code Code de devis :  Choisir un code Date de validité :

Etat :  Choisir

**Liste des devis**  
(1 of 3)

Code de devis	Nom du client	Code du client	Date de validité	Total	Etat	Action
DC20155179	fatma Innov	CP20	10-06-2015	4093.3	Valide	<input type="button" value="D"/> <input type="button" value="E"/> <input type="button" value="X"/> <input type="button" value="P"/>
DC20155178	fatma Innov	CP20	31-05-2015	69.96	Non valide	<input type="button" value="D"/> <input type="button" value="E"/> <input type="button" value="X"/> <input type="button" value="P"/>
DC20155176	siwar	CP14	07-05-2015	3887.32	Valide	<input type="button" value="D"/> <input type="button" value="E"/> <input type="button" value="X"/> <input type="button" value="P"/>
DC20155175	siwar	CP14	05-05-2015	363.58	Non valide	<input type="button" value="D"/> <input type="button" value="E"/> <input type="button" value="X"/> <input type="button" value="P"/>

Figure 45 : Interface de gestion des devis

## 6.2.4. Sprint 4 : Module gestion des bons de commandes:

### Diagrammes de classes :

La figure 46 illustre le diagramme de classes de module « Gestion des bons de commandes».

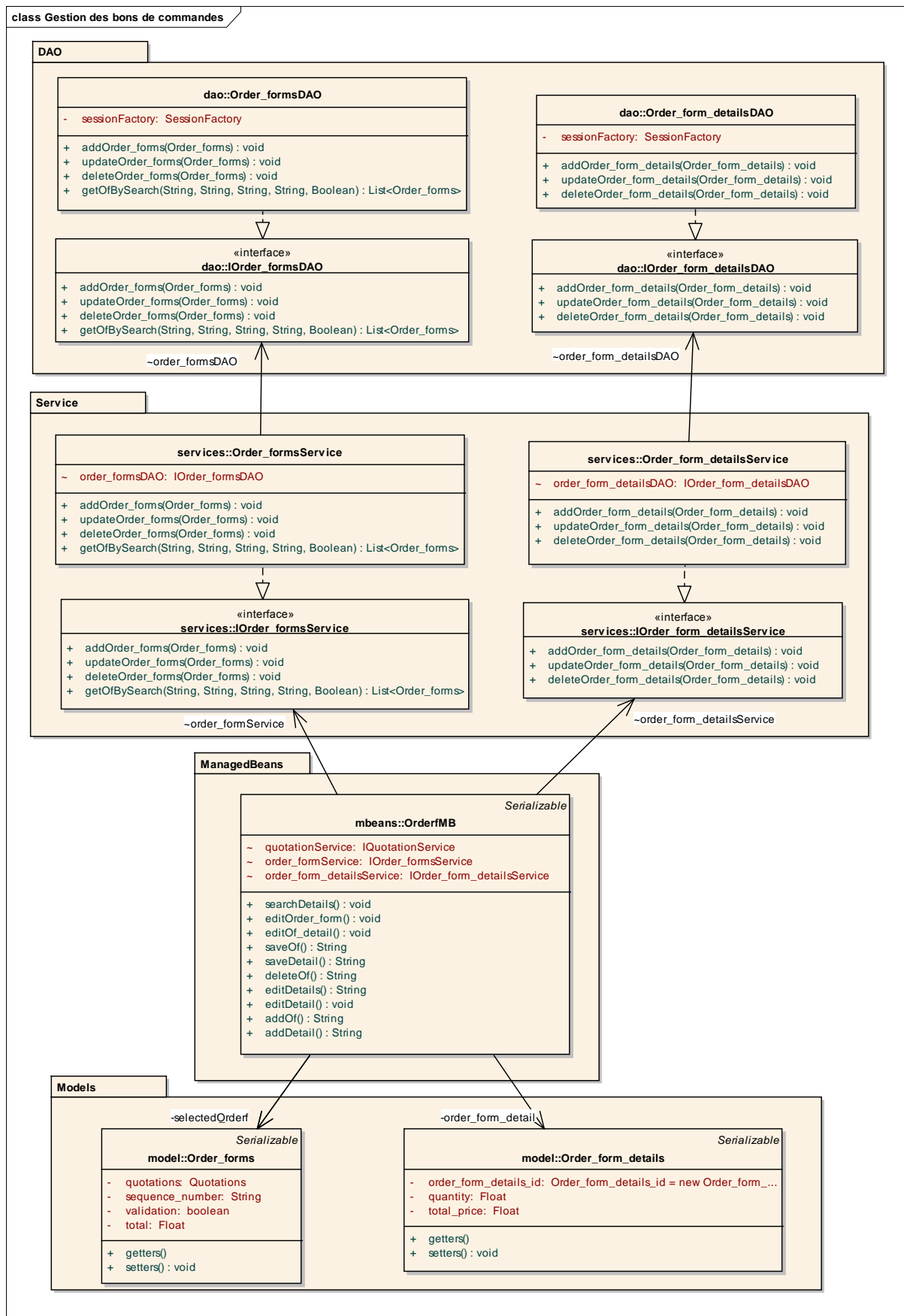


Figure 46 : Diagramme de classes de module « Gestion des bons de commandes »

La classe principale est le managedBean «OrdrefMB». Il gère les opérations de gestion des bons de commandes et leurs détails.

### Test :

La figure 47 illustre l'interface de création d'un nouveau bon de commande qui est associé à un devis valide.

**Nouveau bon de commande**

Chercher un devis

Nom de client :  Code du client : Choisir un code Code de devis : Choisir un code Date de validité :

Etat de devis : Choisir

Chercher Effacer

Liste des devis						
(1 of 3)						
Code de devis	Nom du client	Code du client	Date de validité	Total	Etat	Action
DC20155179	fatma Innov	CP20	10-06-2015	4093.3	Valide	Nouveau BC
DC20155178	fatma Innov	CP20	31-05-2015	69.96	Non valide	Nouveau BC
DC20155176	siwar	CP14	07-05-2015	3887.32	Valide	Nouveau BC

Figure 47 : Interface de création d'un bon de commande

L'application offre la possibilité de gérer les bons de commandes. Les opérations de gestion sont la recherche, la modification, la suppression, la consultation des détails et l'impression des bons de commandes valides. La figure 48 représente l'interface de gestion des bons de commandes.

**Gérer les bons de commande**

Zone de recherche

Nom de client :  Code du client : Choisir un code Code de devis : Choisir un code Code de bon de commande : Choisir un code

Etat de Bon de commande : Choisir

Chercher Effacer

Liste des Bons de Commande						
(1 of 2)						
Code du BC	Nom du client	Code du client	Code du devis	Date de création	Total	Etat
BC2015579	fatma Innov	CP20	DC20155179	22-05-2015	4093.3	Valide
BC2015578	siwar	CP14	DC20155176	07-05-2015	23.3201	Non valide
BC2015577	fatma Innov	CP20	DC20154169	05-05-2015	3864.0	Non valide

Figure 48 : Interface de gestion des bons de commandes

## 6.2.5. Sprint 5 : Module livraison :

### Diagrammes de classes :

La figure 49 illustre le diagramme de classes de module « Gestion des livraisons».

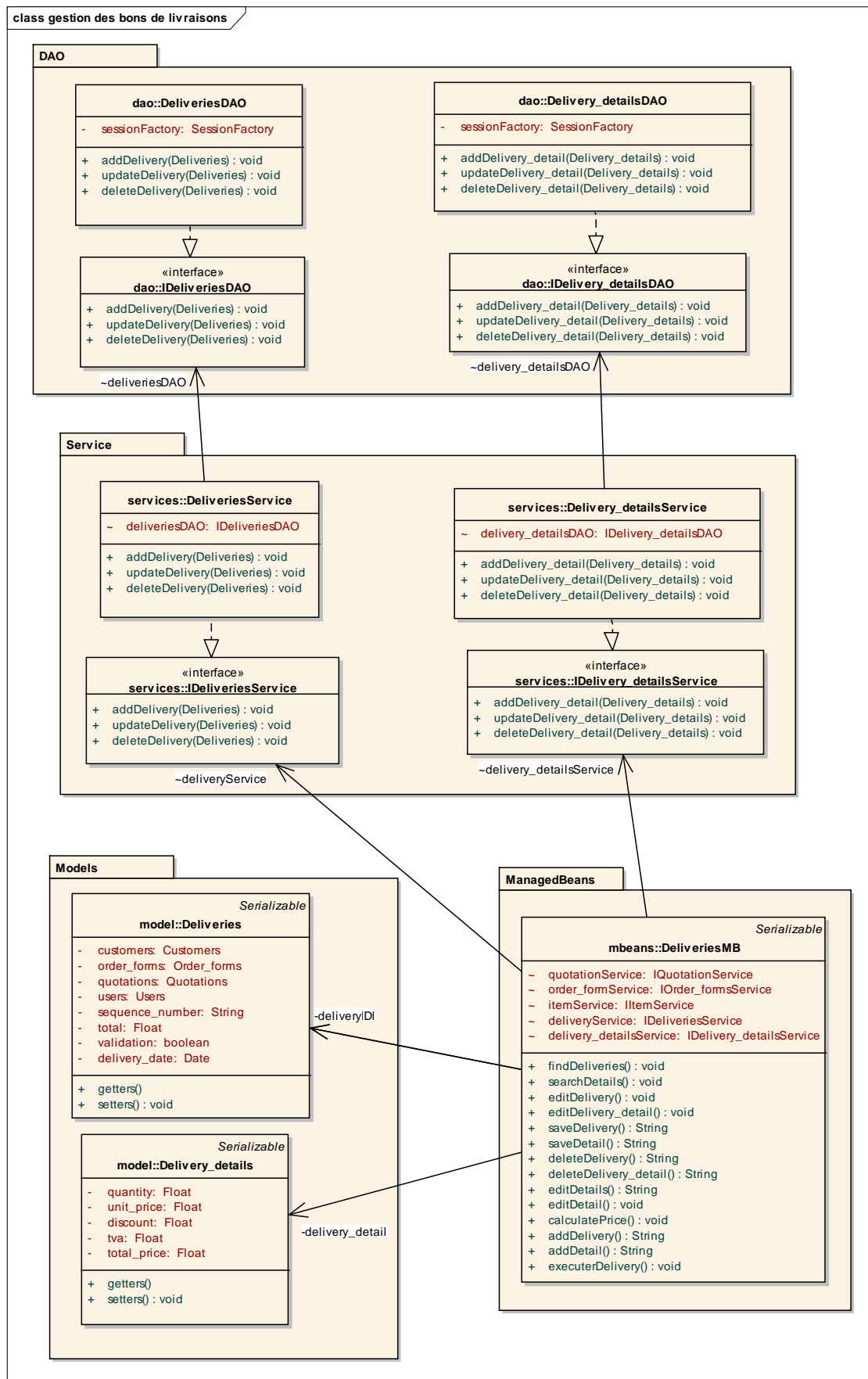


Figure 49 : Diagramme de classes de module « Gestion des bons de livraisons »



La classe principale est le managedBean « deliveriesMB ». Il gère les opérations de gestion des bons de livraisons et l'exécution d'une livraison.

### Test :

La figure 50 illustre l'interface de création d'un nouveau bon de livraison qui est associé à un bon un bon de commande valide.

**SD - LEADER**  
Sales & Distribution System

Accueil Clients Gestion de stock Devis Bons de commande Bons de livraisons Executer Livraison Factures Paiements Caisse Paramètres m. mars

### Nouvelle livraison

Recherche des bon de commande

Nom de client :  Code du client : Choisir un code Code de devis : Choisir un code Code de bon de commande : Choisir un code

Etat de Bon de commande : Choisir

Chercher Effacer

**Liste des bons de commandes**  
(1 of 2)

Code du BC	Nom du client	Code du client	Code du devis	Total	Etat	Action
BC2015579	fatma Innov	CP20	DC20155179	4093.3	Valide	Nouvelle livraison
BC2015578	slwar	CP14	DC20155176	23.3201	Non valide	Nouvelle livraison
BC2015577	fatma Innov	CP20	DC20154169	3864.0	Non valide	Nouvelle livraison
BC2015575	fatma Innov	CP20	DC20154169	4934.6	Non valide	Nouvelle livraison
BC2015574	client2	CP19	DC20154171	1341.8	Valide	Nouvelle livraison

Figure 50 : Interface de création d'un nouveau bon de livraison

Suite à la création, nous pouvons gérer les bons de livraisons existants. Les opérations de gestion sont la recherche, la modification, la suppression, la consultation des détails et l'impression des bons de livraisons valides. La figure 51 représente l'interface de gestion des bons de livraisons.

**SD - LEADER**  
Sales & Distribution System

Accueil Clients Gestion de stock Devis Bons de commande Bons de livraisons Executer Livraison Factures Paiements Caisse Paramètres m. mars

### Gérer les livraisons

Zone de recherche

Nom du client :  Code du client : Choisir un code Code de devis : Choisir un code Code de BC : Choisir un code

Code de Bon de livraison : Choisir un code Etat de Bon de livraison : Choisir Date de livraison :

Chercher Effacer

**Liste des livraisons**  
(1 of 2)

Code de livraison	Nom du client	Code du client	Code du BC	Code du BC	Date de livraison	Etat	Total	Action
LC2015536	fatma Innov	CP20	BC2015579	BC2015579	30-06-2015	Valide	4093.3	
LC2015534	client2	CP19	BC2015574	BC2015574	03-06-2015	Valide	1276.8	
LC2015537	fatma Innov	CP20	BC2015579	BC2015579	31-05-2015	Non valide	0.0	
LC2015535	client2	CP19	BC2015574	BC2015574	28-05-2015	Non valide	907.2	
LC2015122	MONOPRIX	CP2	BC2015149	BC2015149	01-01-2015	Non valide	0.0	

Figure 51 : Interface de gestion des bons de livraisons

Après la validation des bons de livraisons, nous pouvons exécuter les livraisons correspondantes. La figure 52 illustre l'interface d'exécution d'une livraison.

Liste des livraisons

(1 of 5)

Code de livraison	Nom du client	Code du client	Code du BC	Code du BC	Date de livraison	Etat	Total	Action
LC2015536	fatma Innov	CP20	BC2015579	BC2015579	30-06-2015	Valide	4093.3	Executer
LC2015534	client2	CP19	BC2015574	BC2015574	03-06-2015	Valide	1276.8	Executer
LC2015537	fatma Innov	CP20	BC2015579	BC2015579	31-05-2015	Non valide	0.0	Executer

(1 of 5)

Lignes de livraison

(1 of 1)

Code de livraison	Désignation de l'article	Quantité	Unité	prix de l'unité (Dinars)		TVA (%)	Remise (%)	Total
LC2015536	LG50d	5.0	pièce	100.0		6.0	0.0	540.6
LC2015536	microProcesseur	4.0	pièce	11.0		6.0	0.0	58.3
LC2015536	HUAWEI	9.0	pièce	330.0		12.0	0.0	3494.4

(1 of 1)

Liste des articles

(1 of 1)

Désignation	Référence	Catégorie	Modèle	Quantité	Unité	Disponibilité	Action
LG50d	reflgd042015	EverVivid Q50	L45-HD	11,000	pièce	Disponible	S
microProcesseur	refmp042015	S2000	LG50-HD	6,000	pièce	Disponible	S
HUAWEI	refh042015	EverVivid Q50	Alu-a	17,000	pièce	Disponible	S

(1 of 1)

Figure 52 : Interface d'exécution d'une livraison

L'opération d'exécution d'une livraison est associée à un bon de livraison valide. En exécutant la livraison, les lignes de livraison s'affichent dans le deuxième tableau et les articles s'affichent dans le troisième tableau. Dans ce tableau, nous pouvons retirer les articles du stock.

## 6.2.6. Sprint 6 : Module gestion des bons de réceptions :

### Diagrammes de classes :

La figure 53 illustre le diagramme de classes de module « Gestion des bons de réceptions »

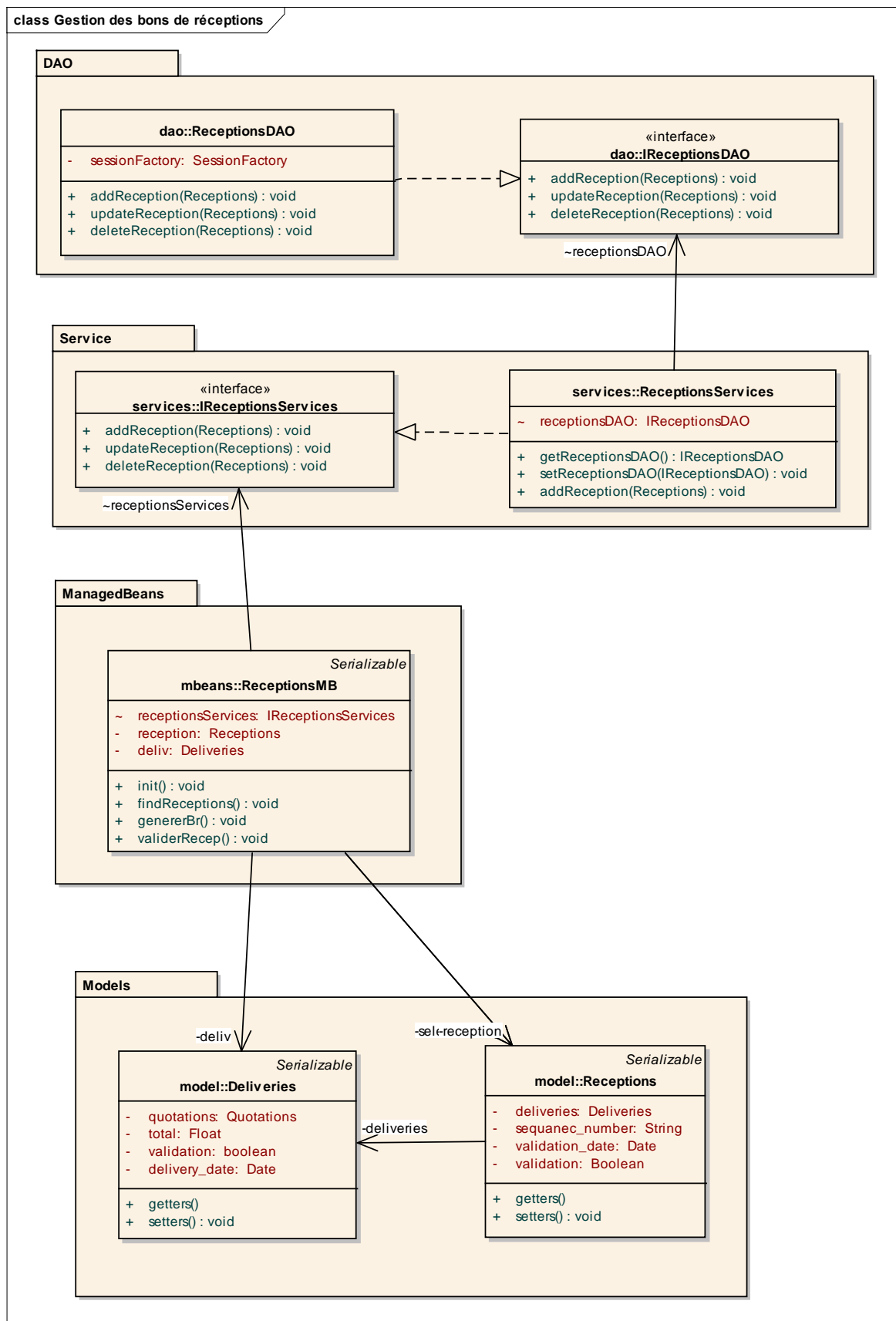


Figure 53 : Diagramme de classes de module « Gestion des bons de réceptions »

La classe principale est le managedBean « ReceptionsMB ». Il gère les opérations de gestion des bons de réceptions.

### Test :

Le module de gestion des bons de réceptions permet de générer un nouveau bon de réception pour un bon de livraison valide. Il permet de valider ou imprimer les bons de réceptions existants. La figure 54 illustre l'interface de gestion des bons de réceptions.

### Gérer les Bons de receptions

Recherche des livraisons

Nom du client : 
Code du client : Choisir un code
Code du devis : Choisir un code
Code de BC : Choisir un code

Code de Bon de livraison : Choisir un code
Etat de Bon de livraison : Choisir
Date de livraison :

Chercher
Effacer

Liste des livraisons
(1 of 5)
1 2 3 4 5

Code de livraison	Nom du client	Code du client	Code du BC	Code du BC	Date de livraison	Etat	Total	Action
LC2015536	fatma Innov	CP20	BC2015579	BC2015579	30-06-2015	Valide	4093.3	Générer BR
LC2015534	client2	CP19	BC2015574	BC2015574	03-06-2015	Valide	1276.8	Générer BR
LC2015537	fatma Innov	CP20	BC2015579	BC2015579	31-05-2015	Non valide	0.0	Générer BR

(1 of 5)
1 2 3 4 5

Recherche des Bons de réceptions

Code de bon de réception : Choisir un code
Etat de Bon de réception : Choisir
Code de bon de livraison : Choisir un code
Date de livraison :

Chercher
Effacer

Liste des bons de réceptions
(1 of 1)
1

Code de réception	Nom du client	Code du client	Code de Livraison	Date de livraison	Etat	Action
RC201556	fatma Innov	CP20	LC2015536	30-06-2015	Non valide	✓
RC201555	fatma Innov	CP20	LC2015536	30-06-2015	Valide	✓

Figure 54 : Interface de gestion des bons de réception

## 6.2.7. Sprint 7 : Module facturation:

### Diagrammes de classes :

La figure 55 illustre le diagramme de classes de module «Facturation»

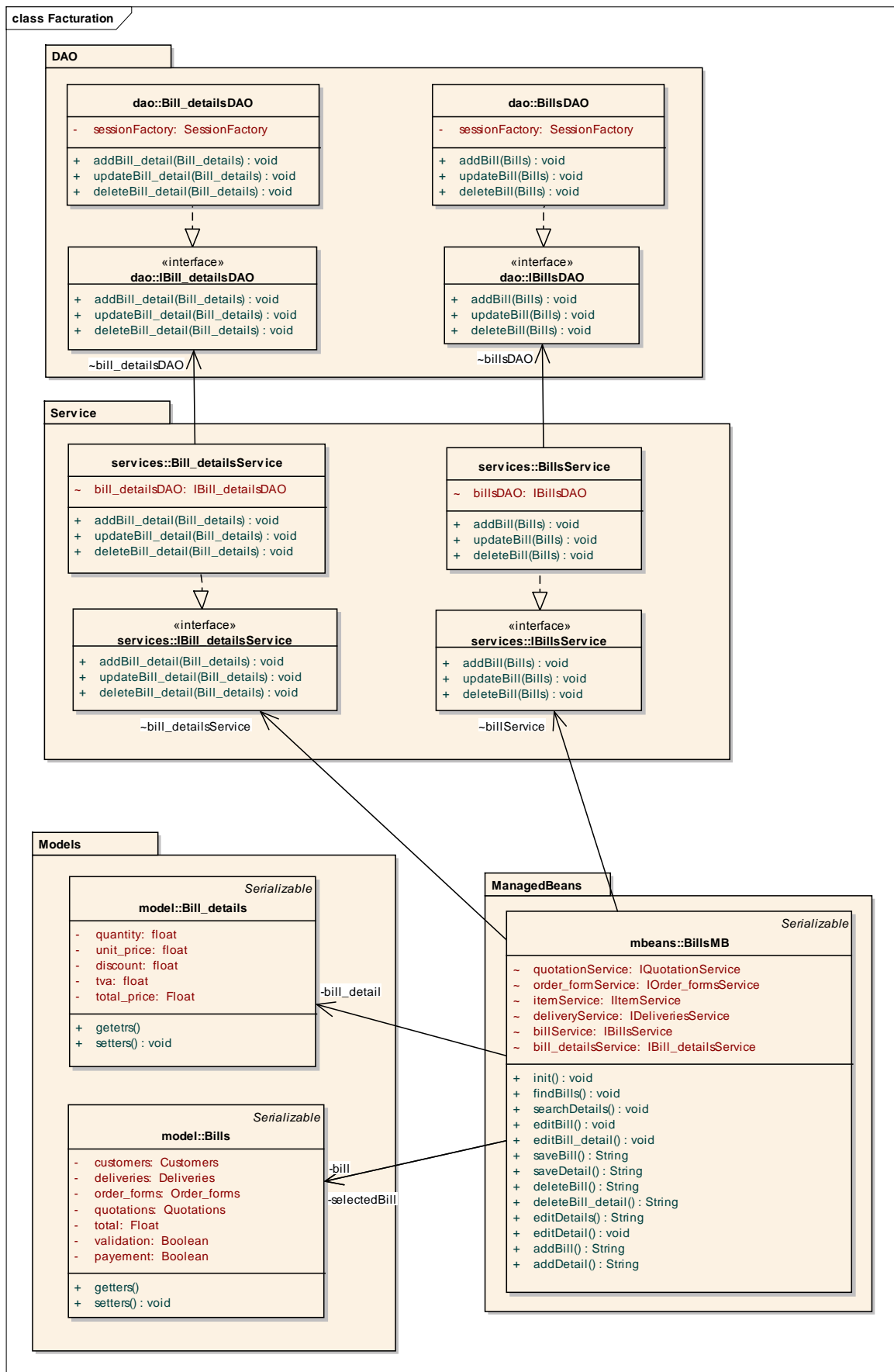


Figure 55 : Diagramme de classes de module « Facturation »

La classe principale est le managedBean « BillsMB » Il gère les opérations de gestion des factures et la génération d'une nouvelle facture.

### Test :

La figure 56 représente l'interface de génération d'une nouvelle facture qui est associée à un bon de réception validé par le client.

**Nouvelle facture**

Recherche des Bons de réceptions

Code de bon de réception :  Choisir un code Etat de Bon de réception :  Choisir Code de bon de livraison :  Choisir un code Date de livraison :

**Liste des bons de réceptions**

(1 of 1)

Code de réception	Nom du client	Code du client	Code de Livraison	Date de livraison	Etat	Action
RC201556	fatma innov	CP20	LC2015536	30-06-2015	Non valide	<input type="button" value="Nouvelle facture"/>
RC201555	fatma innov	CP20	LC2015536	30-06-2015	Valide	<input type="button" value="Nouvelle facture"/>
RC201554	client2	CP19	LC2015534	03-06-2015	Valide	<input type="button" value="Nouvelle facture"/>

Figure 56 : Interface de génération d'une nouvelle facture

Après la génération, nous pouvons gérer les factures existantes. Les opérations de gestion sont la recherche, la modification, la suppression, la consultation des détails et l'impression des factures valides. La figure 57 illustre l'interface de gestion des factures.

**Gérer les factures**

Recherche des factures

Nom du client :  Code du client :  Choisir un client Code du devis :  Choisir un devis Code de BC :  Choisir un BC

Code de livraison :  Choisir une livraison Code de facture :  Choisir une facture Etat de facture :  Choisir

**Liste des factures**

(1 of 2)

Code de facture	Nom du client	Code du client	Code de livraison	Etat	Total	Action
FC2015536	fatma innov	CP20	LC2015536	Non valide	3494.4	<input type="button" value="D"/> <input type="button" value="A"/> <input type="button" value="X"/> <input type="button" value="I"/>
FC2015535	fatma innov	CP20	LC2015536	Valide	3494.4	<input type="button" value="D"/> <input type="button" value="A"/> <input type="button" value="X"/> <input type="button" value="I"/>
FC2015534	fatma innov	CP20	LC2015536	Valide	540.6	<input type="button" value="D"/> <input type="button" value="A"/> <input type="button" value="X"/> <input type="button" value="I"/>
FC2015533	new customer	CP9	LC2015226	Valide	134.4	<input type="button" value="D"/> <input type="button" value="A"/> <input type="button" value="X"/> <input type="button" value="I"/>

Figure 57 : Interface de gestion des factures

## 6.2.8. Sprint 8 : Module paiement:

### Diagrammes de classes :

La figure 58 illustre le diagramme de classes de module « Paiement »

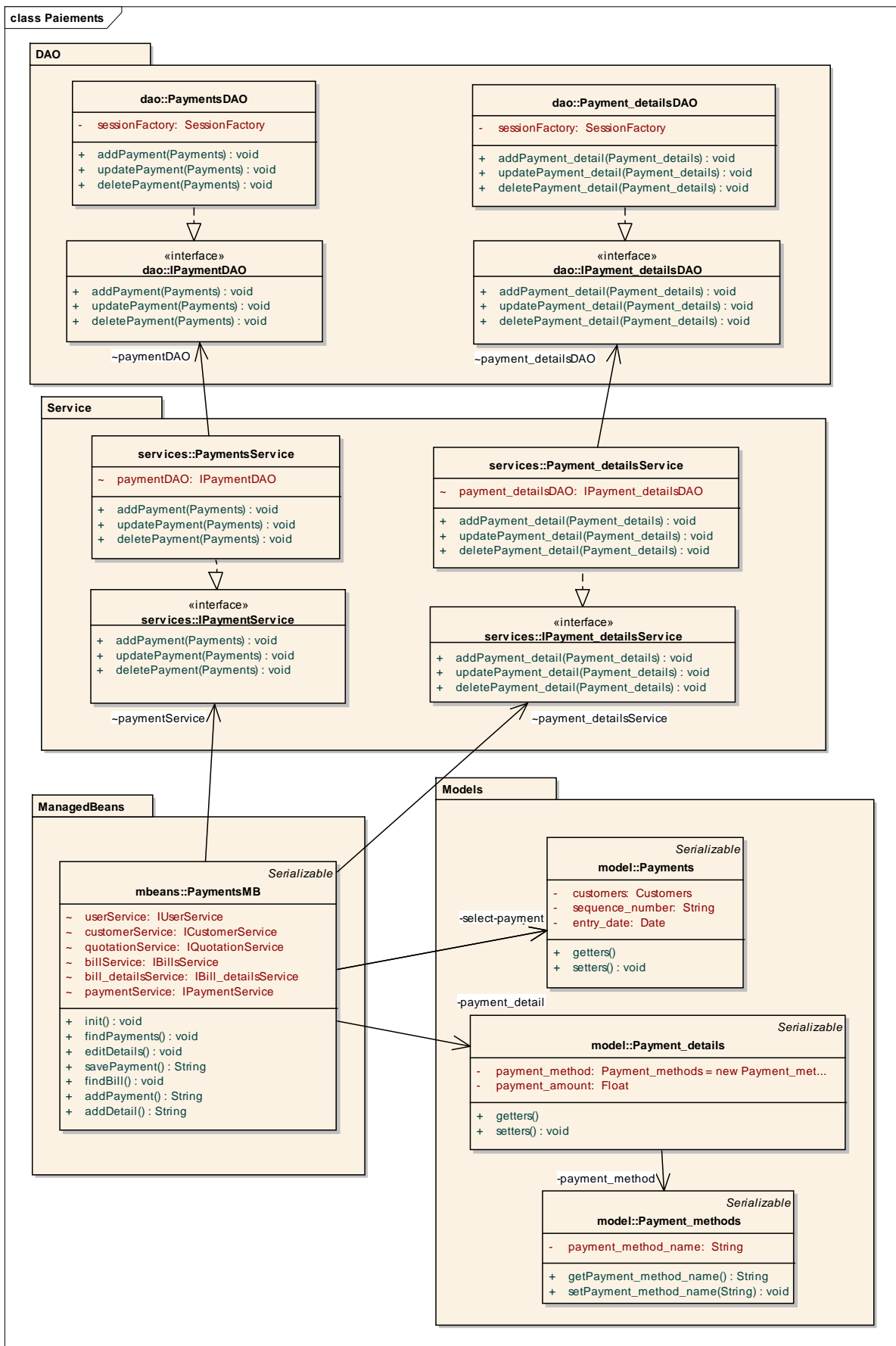


Figure 58 : Diagramme de classes de module « Paiement »

La classe principale est le managedBean «PaymentMB». Il gère les opérations de gestion des paiements et de leurs détails comme la méthode de paiement et la facture à payer.

### Test :

La figure 59 illustre l'interface de paiement d'une facture.

**Nouveau paiement**

Chercher une facture

Nom du client :  Code du client : Choisir un client Code de facture : Choisir une facture

Chercher Effacer

**Liste des factures**  
(2 of 2)

Nom du client	Code du client	Code du facture	Etat	Total	Action
new customer	CP9	FC2015533	Payé	134.4	Payer
fatma innov	CP20	FC2015530	Payé	4093.3	Payer
fatma innov	CP20	FC2015535	Non Payé	3494.4	Payer

Figure 59 : Interface de paiement

Après la réglementation de paiement, nous pouvons gérer les opérations de paiements existantes. Les opérations de gestion sont la recherche, la modification, la suppression et l'impression d'un reçu de paiement. La figure 60 illustre l'interface de gestion des opérations des paiements.

**Gérer des paiements**

Nom du client :  Code du client : Choisir un client Code du paiement : Choisir un paiement

Chercher Effacer

**Liste des paiements**  
(1 of 2)

Nom du client	Code du client	Code du paiement	Date du paiement	Action
fatma innov	CP20	PC20155128	22-05-2015	✎ ✕ 🗑
new customer	CP9	PC20155127	22-05-2015	✎ ✕ 🗑
client2	CP19	PC20155126	22-05-2015	✎ ✕ 🗑

Figure 60 : Interface de gestion des opérations des paiements

### Conclusion :

Dans ce chapitre, nous avons présenté la réalisation de notre projet à travers l'illustration des différents sprints.



# Conclusion Générale

Dans le cadre de notre projet de fin d'études, nous avons conçu et développé une application de gestion commerciale intitulée SD-Leader. Le présent rapport couvre l'étude, la conception et l'implémentation de la solution. En appliquant la méthodologie Scrum, nous avons développé notre application sprint par sprint.

Nous avons commencé dans un premier lieu par étudier le contexte général de notre application en rédigeant un manuel d'utilisation pour l'application existante ainsi que l'étude des processus métiers de la gestion commerciale. Nous avons préparé par la suite un planning de travail en respectant les priorités de nos besoins.

Les difficultés rencontrées lors de ce travail concernent principalement l'étude du métier existant et l'élaboration d'un manuel pour l'application qui a été précédemment développée et qui n'a pas été richement documentée. Ce travail nous a permis d'approfondir nos connaissances dans les bonnes pratiques de programmation et la documentation. Au-delà de l'aspect technique, ce projet a été abordé à fin d'assurer une formidable expérience pour notre parcours, grâce aux contacts des membres de la société IT-Leaders qui nous ont fait bénéficier de leur expérience. En effet, prendre en charge un tel projet dans un cadre professionnel, nous permet assurément de développer notre esprit d'analyse, de réflexion et de décision.

Finalement, notre travail ne s'arrête pas à ce niveau, en effet il est possible d'ajouter des nouvelles fonctionnalités à notre application, citons comme exemple le système basé sur les lecteurs codes à barres des produits qui permet de faciliter les opérations de gestion de stock. En outre nous pouvons développer un module de sécurité pour enrichir les fonctionnalités de gestion des utilisateurs et pour qu'il soit accessibles par les autres modules de l'ERP.

# Bibliographie et Webographie

- [1] **IT-Leaders** [www.it-leaders.net](http://www.it-leaders.net), consulté le 25/05/2015
- [2] **Biware Consulting** [www.biware-consulting.com](http://www.biware-consulting.com) consulté le 25/05/2015
- [3] **SDA Consulting** [www.sda-consulting.net](http://www.sda-consulting.net) consulté le 25/05/2015
- [4] **S.Ambler**, Une tentative de définition, [www.agilemodeling.com/essays/agileSoftwareDevelopment.htm](http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm) consulté le 06/05/2015
- [5] **A. Claude**, SCRUM : le guide pratique de la méthode agile, 2eme Edition DUNOD
- [6] **SCRUM Alliance**, <https://www.scrumalliance.org/why-scrum> consulté le 06/05/2015
- [7] **C. BACH, D. L. Scapin**, Critères Ergonomiques pour les Interactions Homme-Environnements Virtuels : définitions, justifications et exemples, version 2007
- [8] **M. Owen, J. Raj**, BPMN and Business Process Management Introduction to the New Business Process Modeling Standard
- [9] **S. White**, Introduction to BPMN, Version May 2004
- [10] **Hibernate**, <http://www.hibernate.org/> consulté le 11/05/2015
- [11] **Spring**, <http://www.springsource.org/> consulté le 11/05/2015
- [12] **Primefaces**, <http://www.primefaces.org/> consulté le 11/05/2015
- [13] **D. Alur, J. Crupi, D. Malks**, Core J2EE™ Patterns: Best Practices and Design Strategies, Second Edition
- [14] **M. Fowler**, InversionOfControl, <http://martinfowler.com/bliki/InversionOfControl.html> consulté le 25/05/2015
- [15] **Maven**, <https://maven.apache.org> consulté le 11/05/2015
- [16] **Jetty**, <http://eclipse.org/jetty/> consulté le 11/05/2015
- [17] **JSF**, <http://javaserverfaces.java.net> consulté le 11/05/2015
- [18] **PostgreSQL**, <http://www.postgresql.org/> consulté le 11/05/2015
- [19] **Pgadmin**, <http://www.pgadmin.org/> consulté le 11/05/2015
- [20] **DbVisualizer**, <https://www.dbvis.com/> consulté le 11/05/2015
- [21] **Enterprise Architect**, <http://www.sparxsystems.com.au/> consulté le 11/05/2015
- [22] **ArisExpress**, <http://www.ariscommunity.com/aris-express> consulté le 11/05/2015

# Annexe

SD-Leader Project					
Date	Time		Activities	Type	Time Summary
lun - 30/mars/2015	08:30	12:30	Exception Method Not found : Operation "Depôt des articels dans le stock"	Dév: gestion de stock	4:00:00
	13:30	17:30	Exception Method Not found : Operation "Depôt des articels dans le stock"	Dév: gestion de stock	4:00:00
mar - 31/mars/2015	08:30	12:30	Exception No insertion dans BD : Operation "Depôt des articels dans le stock"	Dév: gestion de stock	4:00:00
	13:30	17:30	Exception No insertion dans BD : Operation "Depôt des articels dans le stock"	Dév: gestion de stock	4:00:00
mer - 01/avr/2015	08:30	12:30	Operation "Depôt des articels dans le stock"	Dév: gestion de stock	4:00:00
	13:30	17:30	contrôle Operation "Retrait des articels du stock"	Dév: gestion de stock	4:00:00
jeu - 02/avr/2015	08:30	12:30	operation "Retrait des articels du le stock"	Dév: gestion de stock	4:00:00
	13:30	17:30	contrôle Operation "Transfert des articels"	Dév: gestion de stock	4:00:00
ven - 03/avr/2015	08:30	12:30	Operation "Transfert des articels "	Dév: gestion de stock	4:00:00
	13:30	17:30	contrôle sur l'affichage et l'interface	Dév: gestion de stock	4:00:00
Sub-total					40:00:00

Figure 61 : Exemple d'un Sprint Backlog

يتمثل العمل المقدم في مشروع ختم الدروس في تصميم و تطوير نسخة جديدة لبرمجية إدارة الأعمال التجارية إستنادا على دراسة العمليات المهنية. يتمثل هذا المشروع كخطوة أولى لتطوير برنامج تخطيط موارد الشركات المعد من قبل شركة IT-Leaders الذي يشكو من عديد القصور الوظيفية. و تعتمد هذه البرمجية بالأساس على تكنولوجيا J2EE و على النموذج القياسي BPMN

**المفاتيح :** برنامج تخطيط موارد الشركات ، العمليات المهنية ، BPMN ، إدارة الأعمال التجارية ، SCRUM

---

## Refonte et extension de module de gestion commerciale SD-Leader (Sales & distribution System)

---

Le travail réalisé dans ce projet de fin d'études consiste à concevoir et implémenter une nouvelle version d'une application de gestion commerciale en se basant sur l'étude de processus métier. Ce travail fait partie d'un projet de refonte et d'extension de l'ERP développé par la société IT-Leaders qui souffre de quelques insuffisances en matière de fonctionnalités. L'application développée est fondée sur la technologie J2EE et basée sur les normes de notation graphique BPMN de gestion des processus.

**Mots-clés :** ERP, processus métier, BPMN, gestion commerciale, SCRUM

---

## Redesign and upgrade of a sales and distribution management application SD-Leader (Sales & Distribution System)

---

The work achieved in this graduating project, consists in designing and implementing a new version of sales and distribution management application within a business process management approach. This project is a part of redesign and upgrade ERP developed by IT-Leaders. This ERP is suffered from some functional deficiencies. The application is based on J2EE technology and BPMN notation standards.

**Key Words:** ERP, business process, BPMN, sales and distribution management, SCRUM

---

### Intitule et adresse complète de l'entreprise :

---

Entreprise : IT-Leaders

Adresse : Espace Tunis, Bureau N° B-4-3, Montplaisir, 1002 Tunis Tunisie

Tél. : 71 90 81 82

Fax : 71 90 81 82

Email: [contact@it-leaders.net](mailto:contact@it-leaders.net)