

Université Blida1
Faculté des Sciences
Département d'Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'obtention du diplôme de MASTER

Spécialité : Informatique
Option : Ingénierie des Logiciels

Thème

Conception et réalisation d'une plateforme intégrée pour la gestion
stratégique des contrats avec data visualisation.

Réalisé par : TEBANE Mohamed
HADJ HAMOU Riad

Promoteur : Mr. CHIKHI Nacim Fateh
Encadreur : Mr. BELAID Zahir

Devant les membres du jury :

Présidente : Mme. ABED Hafida

Examineur : Mr. BALA Mahfoud

Dédicace

“

*À nos très chers parents, qui sont toujours à nos côtés, que
Dieu Vous accorde une longue et heureuse vie,*

*À nos frères et sœurs pour qui, nous le savons, notre
réussite est très importante. Que Dieu vous paye pour tous
vos bienfaits,*

À toute notre famille,

À tous nos amis,

*À tous ceux qui ont contribué un jour à notre éducation et
formation,*

Merci.

”

Remerciements

Tout d'abord nous remercions Dieu le tout puissant de nous avoir accordé la connaissance, donner le courage, la patience et la santé pour réaliser notre projet de fin d'étude.

Nous adressons nos plus sincères remerciements à notre promoteur Monsieur Chikhi Nacim Fateh, Enseignant à l'Université Saad Dahlab de Blida, pour avoir suivi nos travaux, pour son aide, ses orientations pédagogiques dans l'élaboration de ce mémoire et tous les conseils qu'il nous a prodigués.

Nous tenons à exprimer nos vifs remerciements et notre profonde gratitude à notre encadreur, Monsieur Belaid Zahir, Ingénieur DSI chez SONATRACH, pour son aide précieuse et son accompagnement tout au long de ces travaux de PFE. Nous avons grandement apprécié ses compétences, son écoute attentive et sa disponibilité totale durant toute la préparation de ce mémoire.

Nous tenons à remercier chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de lire et d'évaluer ce mémoire.

Il va sans dire que nous exprimons notre reconnaissance particulière à l'ensemble de nos enseignants de l'Université Saad Dahlab de Blida, et tout spécialement à nos professeurs d'Informatique, pour les connaissances qu'ils nous ont transmises, leur disponibilité, et l'encadrement dont nous avons bénéficié.

Nous ne pouvons nommer ici toutes les personnes qui, de près ou de loin nous ont aidés et encouragés, mais nous les en remercions vivement.

Enfin, nous tenons à souligner l'importance du soutien quotidien de nos familles tout au long de ces années. Nous leur en sommes profondément reconnaissants.

Résumé

Ce mémoire présente le développement d'une nouvelle plateforme intégrée chez SONATRACH pour la gestion stratégique des contrats, bons de commande et avenants, intégrant des fonctionnalités avancées de Business Intelligence (BI) et de Data Visualization. L'objectif principal est d'améliorer l'efficacité opérationnelle et la transparence des processus de gestion contractuelle. Elle fournit également des tableaux de bord interactifs et des outils de visualisation de données pour une analyse approfondie des contrats.

Ce projet contribue significativement à l'évolution des pratiques de gestion chez SONATRACH en offrant une solution moderne et intégrée adaptée aux besoins spécifiques de l'entreprise.

Mots clés : SONATRACH, Gestion stratégique, Contrats, Bons de commande, Avenants, BI, Prise de décision, Tableaux de bord, Efficacité opérationnelle.

ملخص

تعرض هذه المذكرة تطوير منصة جديدة متكاملة في سوناطراك للإدارة الاستراتيجية للعقود وأوامر الشراء والتعديلات، مع دمج وظائف ذكاء الأعمال المتقدمة (BI) وتصور البيانات. الهدف الرئيسي هو تحسين الكفاءة التشغيلية والشفافية في عمليات إدارة العقود. كما توفر المنصة لوحات معلومات تفاعلية وأدوات لتصور البيانات من أجل تحليل متعمق للعقود.

يساهم هذا المشروع بشكل كبير في تطور الممارسات الإدارية في سوناطراك من خلال تقديم حل حديث ومتكامل يتكيف مع الاحتياجات الخاصة للشركة.

الكلمات المفتاحية: سوناطراك، الإدارة الاستراتيجية، العقود، أوامر الشراء، التعديلات، ذكاء الأعمال، اتخاذ القرار، لوحات المعلومات، الكفاءة التشغيلية.

Abstract

This thesis presents the development of an new integrated platform at SONATRACH for the strategic management of contracts, purchase orders and amendments, integrating advanced Business Intelligence (BI) and Data Visualization features. The main objective is to improve operational efficiency and transparency of contract management processes. It also provides interactive dashboards and data visualization tools for in-depth analysis of contracts.

This project contributes significantly to the evolution of management practices at SONATRACH by offering a modern and integrated solution adapted to the specific needs of the company.

Keywords : SONATRACH, Strategic management, Contracts, Purchase orders, Amendments, Business Intelligence (BI), Decision-making, Dashboards, Operational efficiency.

Table des matières

Dédicace	I
Remerciements	II
Résumé	III
IV	ملخص
Abstract	V
Introduction générale	1
1 organisme d'accueil	2
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	3
1.3 Les missions Principale de SONATRACH	3
1.4 Présentation de la structure d'accueil	4
1.4.1 Présentation de la DC-DSI	4
1.4.2 Mission de la DC-DSI	5
1.5 présentation de la DC-P&L	6
1.5.1 Mission de la DC-P&L	6
1.6 Conclusion	7
2 Analyse	8
2.1 Introduction	9
2.2 Procédure actuel de Reporting des Contrats	9
2.2.1 Définition des Éléments Clés de l'Étude	9
2.2.2 Champ d'application	9
2.2.3 Rôles et responsabilités	9
2.2.4 Banque de Données Contrat	11
2.2.5 Descriptif du Processus de Gestion des Données Contractuelles	11
2.2.6 Diagramme de classe d'analyse	14
2.2.7 Description des classes	16
2.3 Plateforme de gestion des contrats	17
2.3.1 Limites du Système Actuel	17
2.3.2 Besoins d'une Nouvelle Plateforme Intégrée	18
2.4 Conclusion	18
3 Conception	19

3.1	Introduction	20
3.2	Spécification des besoins du système	20
3.2.1	Identification des acteurs	20
3.2.2	Besoins fonctionnels	20
3.2.3	Besoins non fonctionnels	21
3.3	Modélisation des besoins fonctionnels	21
3.3.1	Diagrammes des cas d'utilisation	21
3.3.2	Diagrammes de séquence	23
3.3.2.1	Premier cas « Création d'un contrat »	23
3.3.2.2	Deuxième cas « Validation d'un contrat »	24
3.3.2.3	Troisième cas « Modification d'un contrat »	25
3.3.3	Diagramme de paquetage	28
3.3.4	Diagramme de classe	29
3.4	Business Intelligence et Data Visualization	29
3.4.1	Objectifs de la BI dans notre système	29
3.4.2	Conception des KPIs (Indicateurs de Performance Clés)	30
3.4.3	Conception des Tableaux de Bord	30
3.4.4	Conception de l'interaction	30
3.4.5	Intégration dans la plateforme	31
3.5	Conclusion	31
4	Implementation	32
4.1	Introduction	33
4.2	Vue globale de la plateforme	33
4.3	Mise en œuvre de la solution	34
4.3.1	Vue d'ensemble de l'architecture	34
4.3.2	Environnement de développement	34
4.3.2.1	Langages de programmations	35
4.3.2.2	Frameworks et bibliothèques	36
4.3.2.3	outils de développement	37
4.3.2.4	Outils de build	38
4.3.2.5	API	38
4.4	Méthodologie de développement appliquée	39
4.5	Stratégie de migration de données	40
4.5.1	Problèmes du schéma de l'ancienne base de données	40
4.5.2	Résolutions des problèmes	40
4.6	Migration	41
4.6.1	Diagramme de déploiement	42
4.7	Interfaces Graphiques	42
4.7.1	Administration	43
4.7.2	Paramétrage et référentiel	44
4.7.3	Gestion Contrats	46
4.7.4	Reporting	48
4.8	Conclusion	50
	Conclusion et perspectives	51

Annexes	54
A scripts de migration	55

Table des figures

1.1	Logo SONATRACH.	3
1.2	Organigramme de la macrostructure de SONATRACH.	4
1.3	Organigramme de la DC-DSI.	6
1.4	Organigramme de la DC-P&L.	7
2.1	Role et responsabilité des Structure.	10
2.2	Logigramme du flux d'informations.	11
2.3	Diagramme d'Activité « Flux de Saisie des Informations dans la BDC ». . .	13
2.4	Diagramme d'Activité « L'élaboration et la transmission du Reporting trimestriel ».	14
2.5	Diagramme de classe « Gestion contrats ».	15
3.1	Diagramme de cas d'utilisation générale du système.	22
3.2	diagramme de cas d'utilisation « Gestion du contrats ».	22
3.3	Diagramme de séquence « Création d'un contrat ».	24
3.4	Diagramme de séquence « Validation d'un contrat ».	25
3.5	Diagramme de séquence « Modification d'un contrat ».	27
3.6	Diagramme d'état transition « État contrat ».	28
3.7	Diagramme de paquetage globale.	28
3.8	Diagramme de classe.	29
4.1	Vue globale systeme.	33
4.2	Vue d'ensemble de l'architecture.	34
4.3	Outils de développement.	35
4.4	Architecture technique.	35
4.5	Flux de données Spring Boot et Oracle Database.	38
4.6	La structure des fichiers et répertoires du projet.	39
4.7	Diagramme de déploiement.	42
4.8	Login page.	43
4.9	Accueil.	43
4.10	Ajout utilisateur.	44
4.11	journal des activités.	44
4.12	Ajout groupement.	45
4.13	Entrepise participante dans un groupement.	45
4.14	Liste des contrats.	46
4.15	Formulaire ajout d'un contrat.	47
4.16	Validation et demande de modification.	47
4.17	Email création contrat.	48
4.18	Email demande modification contrat.	48

4.19	Email modification contrat.	48
4.20	ContratBI 1.	49
4.21	ContratBI 2.	49
4.22	ContratBI 3.	50
A.1	Création des IDs.	55
A.2	Extraction des fichier pdf.	56
A.3	Création du taux dans la table « ContratDevise ».	56
A.4	Modification de la Jointure entre Contrat et Bon de Commande.	57

Liste des tableaux

3.1	Description de cas « Création d'un contrat »	23
3.2	Description de cas « Validation d'un contrat »	25
3.3	Description de cas « Modification d'un contrat »	26

Liste des sigles et acronymes

DC	<i>Direction Centrale</i>
DSI	<i>Digitalisation et des Systèmes d'Information</i>
P&L	<i>Procurement & Logistique</i>
RHC	<i>Responsable Hiérarchique du Correspondant</i>
BDC	<i>Banque de Données Contrats</i>
SGBD	<i>système de gestion de base de données</i>
JDBC	<i>Java Database Connectivity</i>
IT	<i>Information Technology</i>
BI	<i>Business Intelligence</i>
UML	<i>Unified Modeling Language</i>
SQL	<i>Structured Query Language</i>
PL/SQL	<i>Procedural Language extensions to the Structured Query Language</i>
BLOB	<i>Binary Large Object</i>
PDF	<i>Portable Document Format</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>

Introduction générale

Aujourd'hui, dans un monde de plus en plus axé sur les données, les entreprises font face à un défi majeur : gérer efficacement leurs contrats, avenants et bons de commande dans un environnement concurrentiel. SONATRACH, l'un des leaders mondiaux de l'industrie pétrolière et gazière, n'échappe pas à cette réalité. Elle est toujours en train de travailler pour améliorer sa vision globale et en temps réel de ses contrats.

De ce fait, l'entreprise a déjà développé un système de gestion qui s'est révélé efficace par le passé. Cependant, avec ces nouvelles politiques, ce système montre certaines limites pour suivre ces nouvelles exigences. Parmi elles, le manque de fonctionnalités de visualisation des données, qui sont actuellement traitées manuellement. C'est pourquoi elle souhaite développer une nouvelle plateforme pour améliorer la gestion des contrats, avenants et bons de commande, et inclure le traitement des données à l'aide de tableaux de bord. Cette évolution permettra de faciliter la prise de décisions et l'optimisation des processus opérationnels.

La nécessité de mettre en place cette nouvelle plateforme soulève des questions essentielles :

- Quels défis entravent actuellement la gestion contractuelle de l'entreprise et comment affectent-ils ses opérations ?
- Quels avantages concrets peut offrir l'intégration de la Business Intelligence et de la visualisation des données dans cette plateforme ?
- Comment concevoir la nouvelle solution pour répondre à ces besoins et améliorer la gestion contractuelle de SONATRACH ?

L'objectif de ce travail est de concevoir et de mettre en œuvre une nouvelle plateforme intégrée, qui répond aux exigences de l'entreprise.

Ce travail est organisé comme suit :

- Le premier chapitre présente l'organisme d'accueil, SONATRACH, en mettant en avant son contexte et ses activités.
- Ensuite, nous aborderons une étude de l'existant en identifiant les limites du système actuel.
- Les troisième et quatrième chapitres traitent respectivement de la conception de notre solution en utilisant le langage UML, ainsi que du développement et de l'implémentation de notre application.
- Enfin, nous concluons par une synthèse générale et des perspectives.

Chapitre 1

organisme d'accueil

1.1 Introduction

Dans ce chapitre, nous allons en premier lieu présenter l'organisme d'accueil. Nous commencerons par une brève présentation de SONATRACH. Ensuite nous aborderons ses missions principales et sa structure organisationnelle. Nous présenterons la structure d'accueil DC-Digitalisation et Système d'information. Une attention particulière sera aussi accordée à la DC-Procurement & Logistique qui jouera un rôle essentiel dans notre étude.

1.2 Présentation de l'organisme d'accueil

SONATRACH (Société Nationale pour la **R**echerche, la **P**roduction, le **T**ransport, la **T**ransformation et la **C**ommercialisation des **H**ydrocarbures), a été établie en Algérie le 31 décembre 1963. Dès ses débuts, elle a joué un rôle crucial dans l'économie nationale, se distinguant comme une figure majeure du secteur en Afrique et se plaçant en tête des entreprises du continent.

Son histoire est intimement liée à la construction d'un oléoduc stratégique reliant Hassi Messaoud à Arzew. Aujourd'hui, la SONATRACH opère non seulement dans la recherche, la production, le transport, la transformation et la commercialisation des hydrocarbures, mais aussi dans d'autres domaines tels que la génération électrique, les énergies nouvelles et renouvelables, ainsi que le dessalement d'eau de mer. Ses activités s'étendent à l'échelle mondiale, cherchant des opportunités là où elles se présentent.[1]



FIG. 1.1 : Logo SONATRACH.[1]

1.3 Les missions Principale de SONATRACH

Sous l'autorité d'un directeur général, la SONATRACH a pour principales missions :

- Le développement, la préservation et la mise en valeur des réseaux énergétiques à travers l'ensemble du territoire national.
- La reconstitution et l'augmentation des réserves d'hydrocarbures.
- Promouvoir l'évolution, la gestion et l'entretien des installations de production, de transport et de conditionnement des hydrocarbures.

- Le développement des techniques modernes de gestion par le biais de la formation.
- La diversification des marchés et des produits destinés à l'exportation.
- Adapter les outils commerciaux aux demandes du marché énergétique afin de mieux comprendre ses mécanismes et d'optimiser les performances commerciales.
- La prise et la détention de tous portefeuilles d'actions, les prises de participation et autres valeurs mobilières dans toutes les sociétés existantes ou à créer en Algérie ou à l'étranger.[1]

Son organisation générale est illustrée dans l'organigramme de la figure 1.2 suivante :

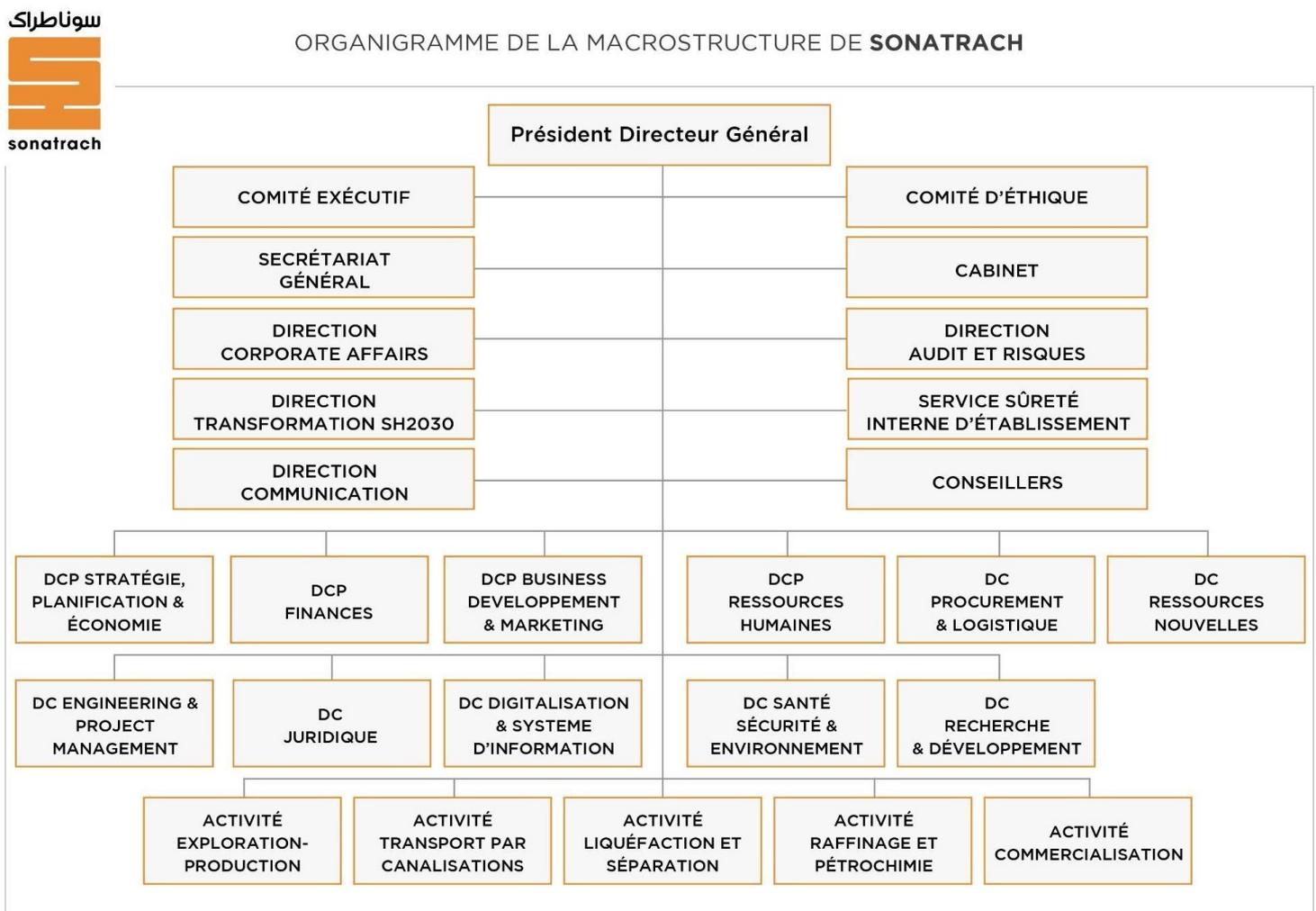


FIG. 1.2 : Organigramme de la macrostructure de SONATRACH.[2]

1.4 Présentation de la structure d'accueil

1.4.1 Présentation de la DC-DSI

La Direction Centrale de la Digitalisation et des Systèmes d'Information est un organe directement rattaché à la Direction Générale de SONATRACH. Son objectif principal

est de formuler et de superviser la mise en œuvre de la politique informatique et de la stratégie de digitalisation de l'entreprise, ainsi que de garantir la conformité aux normes et standards internationaux en matière de technologies et de systèmes d'information. De plus, elle est responsable de la gestion et du suivi des projets internes et externes liés à ces domaines.

1.4.2 Mission de la DC-DSI

- La définition des politiques et stratégies de la Société en matière de système d'information et de digitalisation, ainsi que leur diffusion et la veille à leur application.
- L'élaboration de la stratégie de sécurité du système d'information de SONATRACH et la veille à sa mise en œuvre.
- La planification de la mise en œuvre des projets structurants identifiés et inscrits dans la stratégie arrêtée.
- La dotation de la Société d'infrastructures IT modernes, sécurisées et évolutives pour supporter l'ensemble des composantes du SI de SONATRACH.
- La veille à la haute disponibilité du SI de la Société, ainsi qu'à sa maintenance, son intégration et son évolution.
- La valorisation et la préservation du patrimoine des données de la Société.
- La veille au développement et à la mise à niveau du capital humain en matière d'IT/IS en fonction des évolutions technologiques.
- La contribution en tant que vecteur de digitalisation, avec d'autres opérateurs algériens, à la promotion des programmes de modernisation initiés par les instances publiques.
- L'appui, le conseil et l'assistance aux entités du Groupe en matière de système d'information, de technologies de l'information, d'architecture SI d'entreprise et de solutions digitales.[3]

Son organisation est illustrée dans l'organigramme de la figure 1.3 suivante :

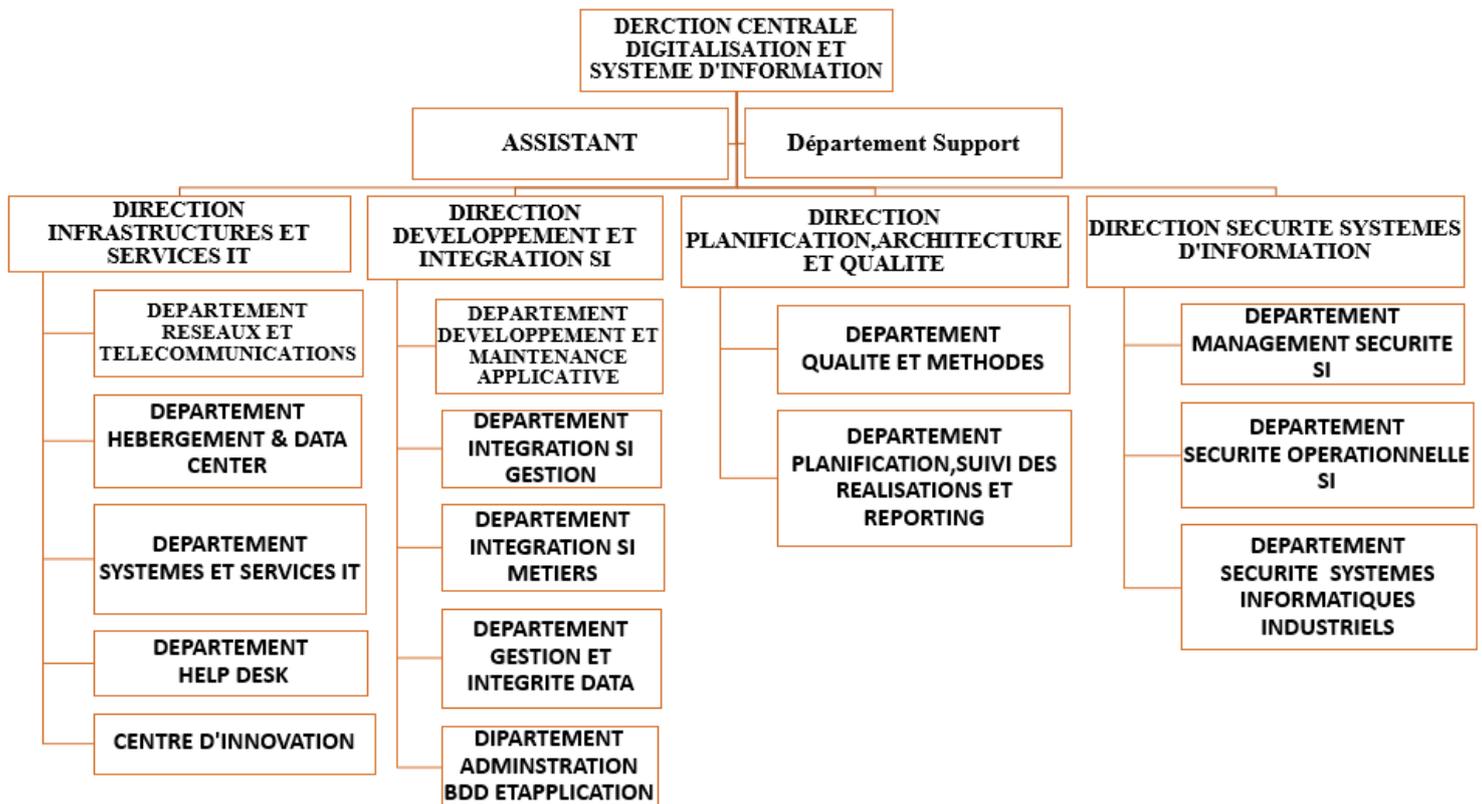


FIG. 1.3 : Oraganigramme de la DC-DSI.[3]

1.5 présentation de la DC-P&L

La Direction Centrale Procurement & Logistique est une entité clé au sein de SONATRACH, chargée de piloter les processus d'achats et de logistique pour l'ensemble de l'entreprise. Son rôle primordial consiste à optimiser ces processus afin d'assurer une gestion efficace des ressources tout en répondant aux besoins spécifiques de SONATRACH.

Récemment, la P&L a lancé une initiative visant à améliorer l'application de gestion des contrats, soulignant ainsi son engagement envers l'amélioration continue et la recherche de l'excellence opérationnelle au sein de l'entreprise.

1.5.1 Mission de la DC-P&L

- La définition, le pilotage et la mise en œuvre de la politique générale de la Société en matière de procurement et de logistique, en fonction de la stratégie de la Société.
- La définition et le suivi de la mise en œuvre des processus permettant l'optimisation de la chaîne d'approvisionnement en biens et services de la Société.
- Le lancement et le pilotage des projets majeurs de la fonction procurement et logistique, tels que la digitalisation, la codification, le e-procurement, l'accréditation des fournisseurs en ligne, etc.
- L'élaboration des études nécessaires concernant le procurement et la logistique

(études de marché des biens et services, études d'opportunité, études d'optimisation de la chaîne logistique, etc.).

- La réalisation des opérations de contrôle et d'inspection portant sur les processus Procurement et Logistique.
- La participation au développement et au déploiement des bases de données, des applicatifs et des outils informatiques ayant trait aux processus Procurement et Logistique (Codification, Performances, Reporting, ...) et ce, en coordination avec la Direction Centrale Digitalisation et Système d'Information.[4]

Son organisation est illustrée dans l'organigramme de la figure 1.4 suivante :

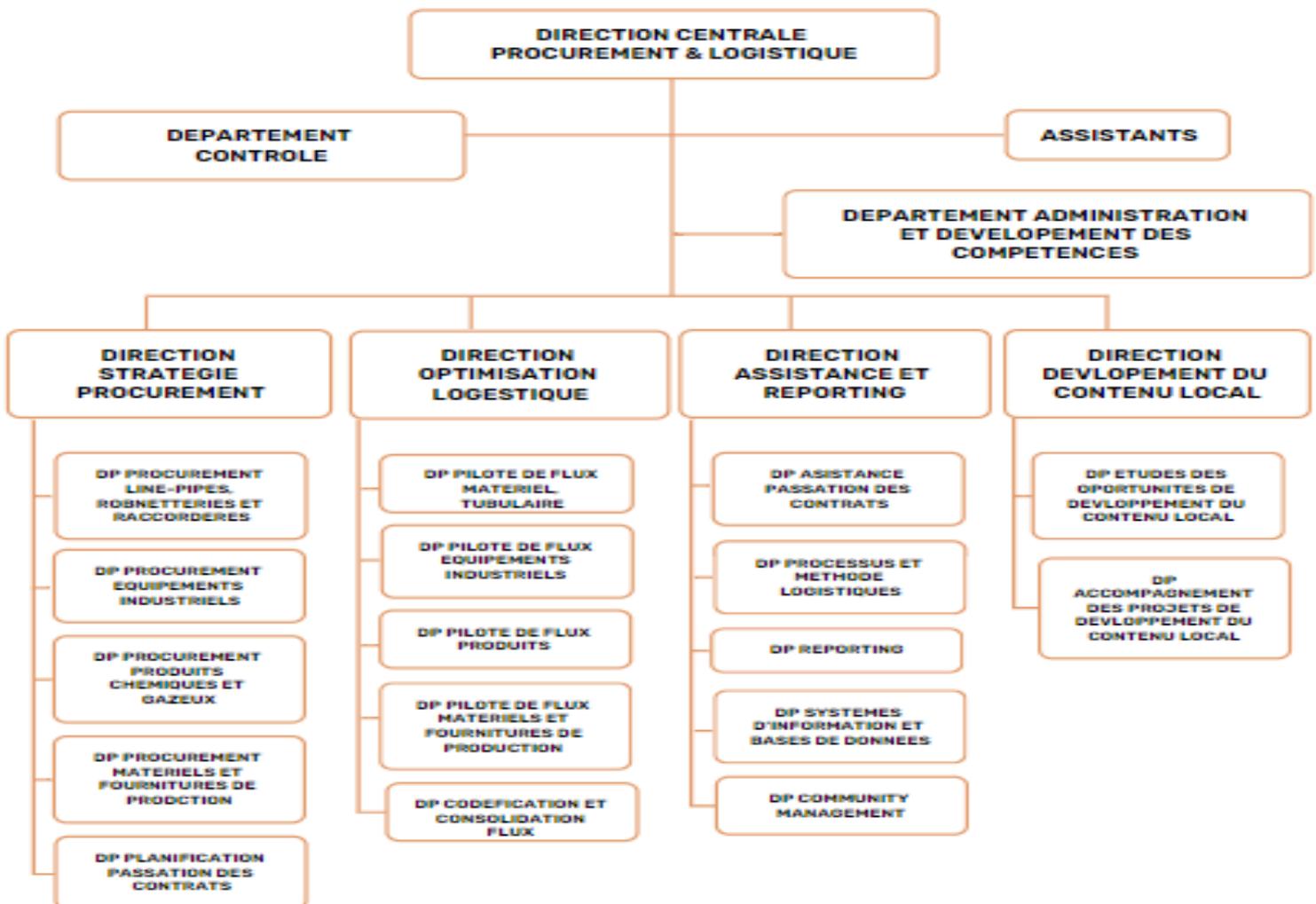


FIG. 1.4 : Organigramme de la DC-P&L.[4]

1.6 Conclusion

En somme, ce premier chapitre nous a fourni un aperçu général de l'organisme d'accueil SONATRACH, de ses missions clés et de son organisation interne, en particulier la Direction Centrale DSI et la Direction Centrale Procurement & Logistique qui sera au cœur de cette étude.

Chapitre 2

Analyse

2.1 Introduction

Dans ce chapitre, on va regarder de près comment SONATRACH gère ses contrats actuellement. Il est important de bien comprendre les procédures de gestion des contrats utilisées par l'entreprise. Si on saisit bien ce fonctionnement, on pourra plus facilement repérer les points faibles de l'application existante et expliquer pourquoi il faut une plateforme mieux conçue.

2.2 Procédure actuel de Reporting des Contrats

2.2.1 Définition des Éléments Clés de l'Étude

Avant de présenter ce que doit faire notre système et ses fonctionnalités, nous devons définir les éléments ciblés par cette étude :

1. **Contrat** : Le contrat est un accord de volontés entre deux ou plusieurs parties destiné à créer, modifier, transmettre ou éteindre des obligations.
2. **Avenant** : L'avenant est l'acte par lequel les parties à un contrat conviennent de modifier ou de compléter une ou plusieurs de ses clauses. Cette modification ne peut avoir ni pour objet ni pour effet de substituer au contrat initial un autre contrat. La seule exception à cette règle concerne les sujétions techniques imprévues rencontrées au cours de l'exécution du contrat.
3. **Bon de commande** : C'est un document reprenant les éléments d'une commande donnant lieu à une vente entre un vendeur et un acheteur. Ce document a donc une valeur d'engagement dès qu'il est signé par les deux parties, en effet le vendeur s'engage à fournir la commande et l'acheteur s'engage à la régler.

2.2.2 Champ d'application

La présente procédure s'applique à tous les contrats, y compris les avenants et bons de commandes, conclus par SONATRACH en effort propre et en association et concerne les marchés de fournitures, de travaux, de services et d'études et services de conseil quel que soit leur montant.

2.2.3 Rôles et responsabilités

Le diagramme de cas d'utilisation 2.1 résume les rôles et responsabilités des différentes entités impliquées dans la procédure de gestions des contrats au sein de SONATRACH. Il permet de visualiser les interactions entre les acteurs et les différentes tâches qu'ils accomplissent.

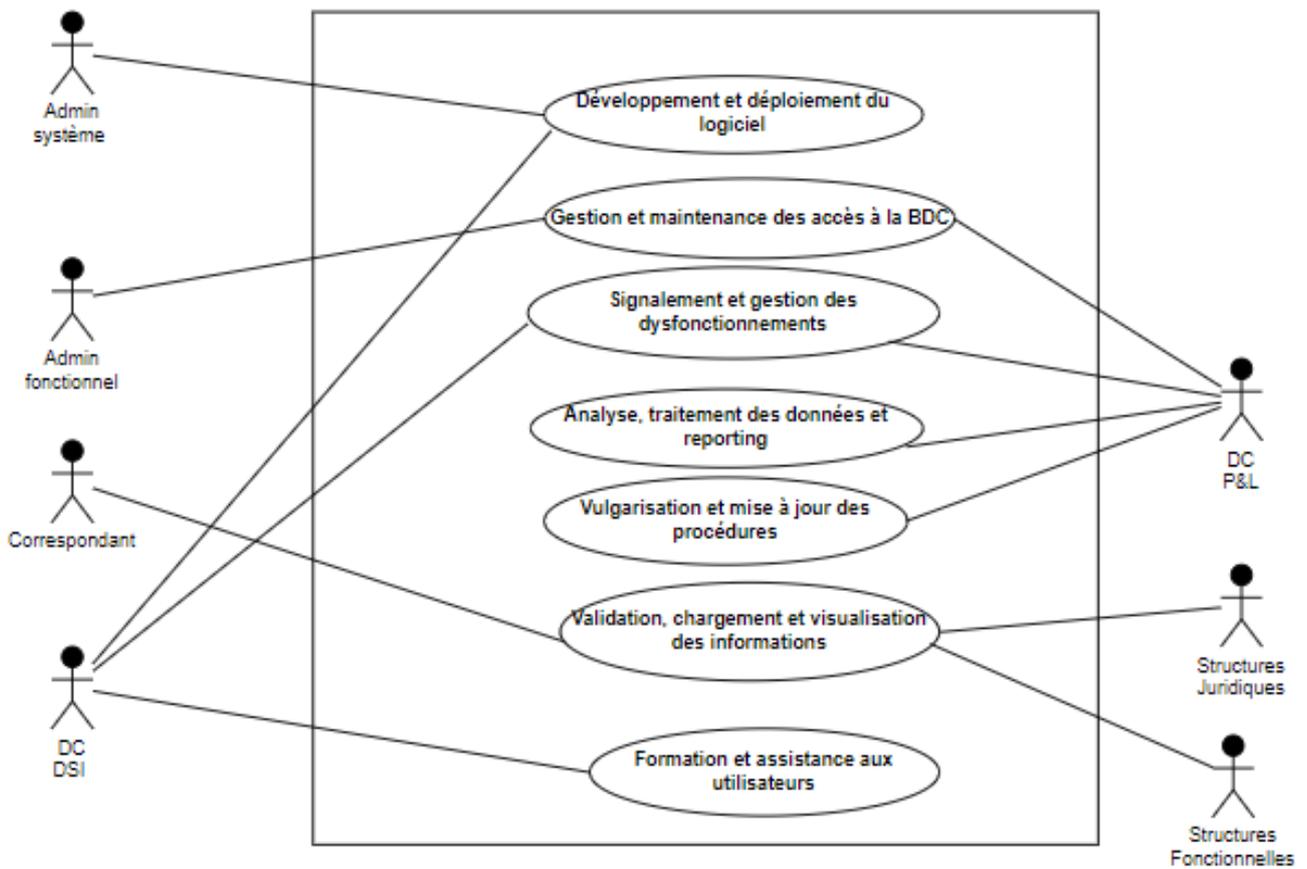


FIG. 2.1 : Role et responsabilité des Structure.

On peut illustrer les flux d'information entre les différentes entités impliquées dans la gestion de la Banque de Données Contrats (BDC) à travers le logigramme de la figure 2.2 ci-dessous.

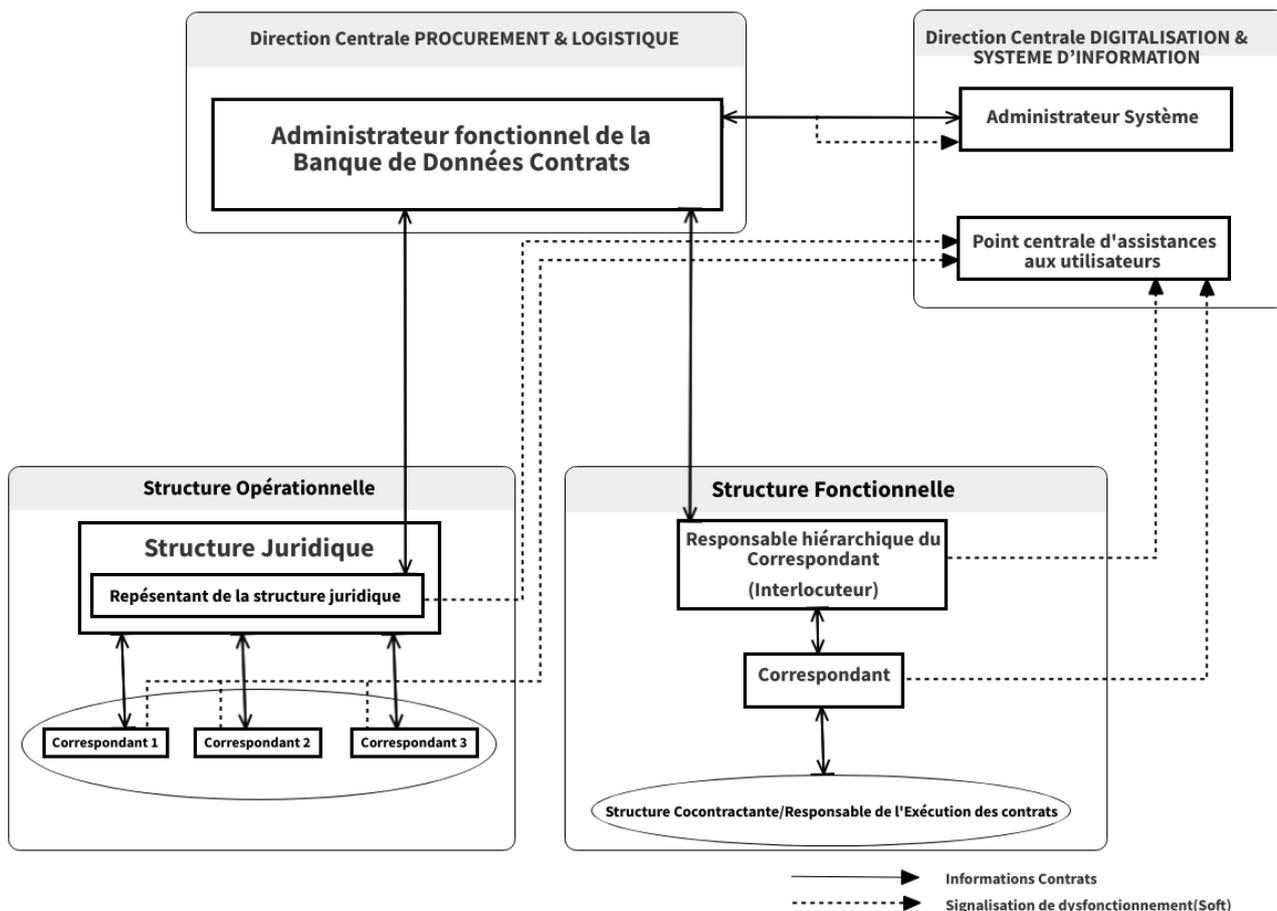


FIG. 2.2 : Logigramme du flux d'informations.[5]

2.2.4 Banque de Données Contrat

La Banque de Données Contrats contient les informations essentielles concernant les contrats, les avenants et les bons de commande liés à la présente procédure. Elle englobe également les données de paramétrage telles que les informations sur les entreprises, les groupements, les associations, etc.

2.2.5 Descriptif du Processus de Gestion des Données Contractuelles

Etape 1 : la collecte et la vérification de la cohérence des données

Dans cette étape, le correspondant a plusieurs responsabilités :

- Collecter, contrôler et traiter les données nécessaires pour l'alimentation de la BDC.
- Relancer, au besoin, les structures contractantes et/ou responsables d'exécution des contrats à l'effet de lui transmettre tout évènement nouveau concernant les contrats et/ou avenants, à titre indicatif : nouveaux, en cours, prorogés, achevés, résiliés.
- Demander aux structures contractantes et/ou responsables d'exécution des contrats

toutes informations supplémentaires et nécessaires au contrôle des données pour en assurer la fiabilité.

- Dans un souci de traçabilité, il reçoit des structures contractantes et/ou responsables d'exécution des contrats, les informations nécessaires à l'alimentation de la BDC par fax ou courriel (adresse e-mail professionnelle).

Etape 2: la saisie des informations collectées et contrôlées

- Le correspondant saisit les informations collectées et contrôlées dans la BDC.
- Il insère dans la BDC, pour les besoins de consultations, les copies numérisées (scannées) des contrats / avenants / bons de commande et visas des commissions des marchés.
- Après la saisie des données collectées et leur sauvegarde, le correspondant classe d'une manière méthodique et sécurisée les documents ou courriels ayant été exploités.

Etape 3 : Validation des Données Saisies

- Chaque fois que de nouvelles données contrôlées sont introduites dans la BDC, le correspondant demande systématiquement dans la journée une validation des données saisies (E-mail ou Fax) à la Structure Juridique relevant de la structure opérationnelle ou au responsable hiérarchique pour les structures fonctionnelles.
- A chaque demande de validation, les interlocuteurs des structures procèdent à un contrôle de la cohérence des données saisies dans la BDC par les correspondants et à leur validation au plus tard un jour suivant la demande.
- Une fois la validation effectuée, les données figurants dans la BDC seront considérées définitives pour les rubriques invariables et imputées à la période pour les rubriques à caractère évolutif.
- Les corrections à opérer après validation, sont signalées par la structure juridique relevant de la structure opérationnelle et par la structure fonctionnelle, à la Direction Centrale P&L, seule habilitée à autoriser des corrections sur une saisie antérieure validée.
- Les corrections, après validation, seront effectuées par le correspondant après autorisation de la Direction Centrale P&L et ce, sur la base de justificatifs.

Ces processus peuvent être synthétisés dans le logigramme 2.3 ci-dessous :

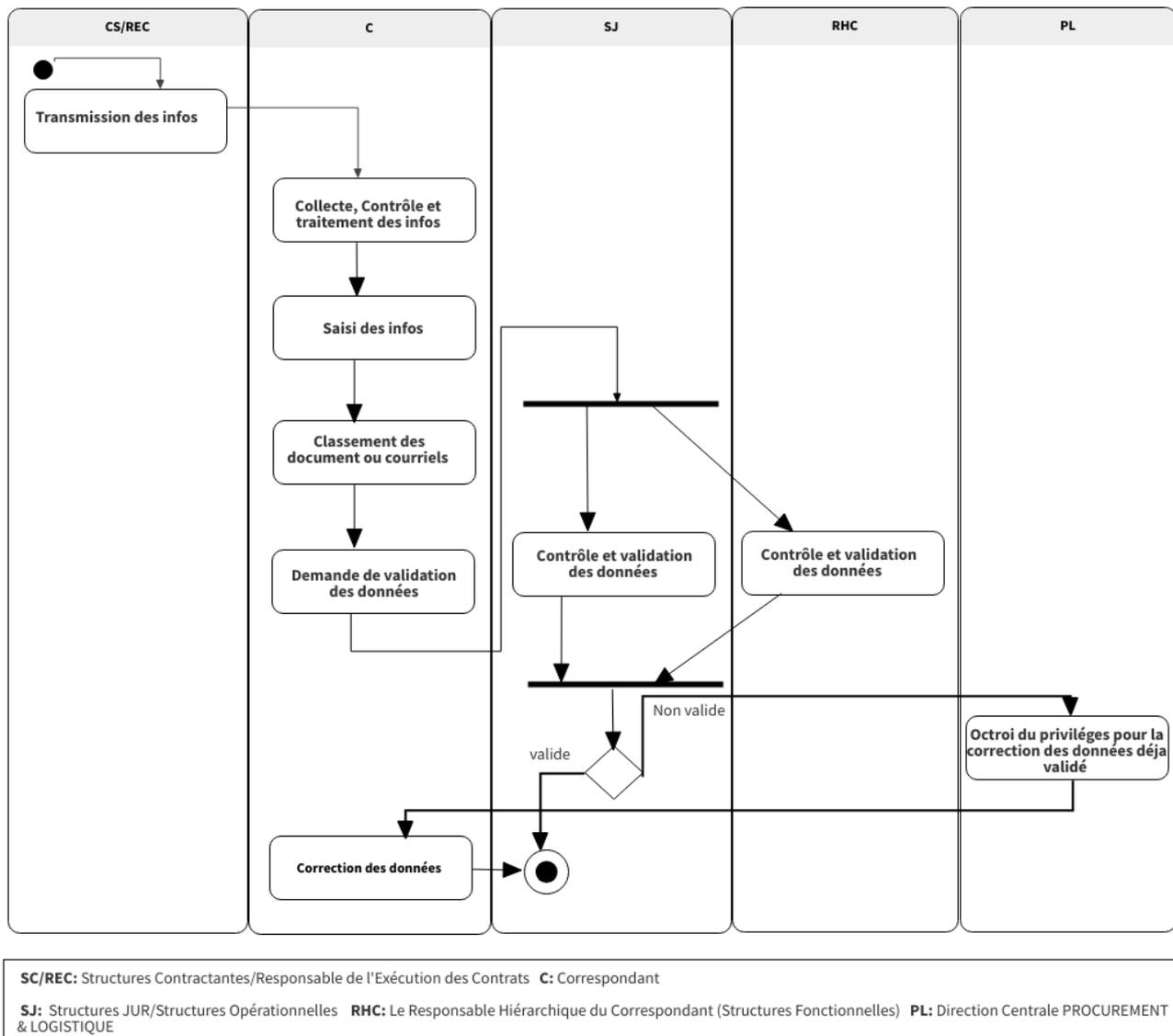


FIG. 2.3 : Diagramme d'Activité « Flux de Saisie des Informations dans la BDC ».

Etape 4 : l'élaboration et la transmission du Reporting trimestriel

Le responsable de la structure juridique relevant de la structure opérationnelle et le responsable hiérarchique du correspondant relevant de la structure fonctionnelle élaborent sur la base des informations contenues dans la BDC, un reporting trimestriel sous format papier, au plus tard le 5 du mois suivant le trimestre considéré, à transmettre :

- au Vice-président et à la Direction Centrale P&L.

La Direction Centrale P&L consolide les données à partir du reporting transmis par les structures juridiques relevant des structures opérationnelles et les structures fonctionnelles.

Elle élabore, après contrôle des données à partir de la BDC, un reporting trimestriel à transmettre à Monsieur le Président Directeur Général, au plus tard le 10 du mois suivant le trimestre considéré.

En sus d'indicateurs quantitatifs, tels que le nombre de contrats par structure, par

mode de passation, coût, avancement, analyse et explication des écarts et annexes, le reporting élaboré au niveau des structures et celui consolidé par la Direction Centrale P&L doivent inclure une synthèse sur les faits marquants de la période considérée ainsi qu'une analyse des données et d'éventuelles recommandations.

Ce processus peut être synthétisé dans le logigramme 2.4 ci-dessous :

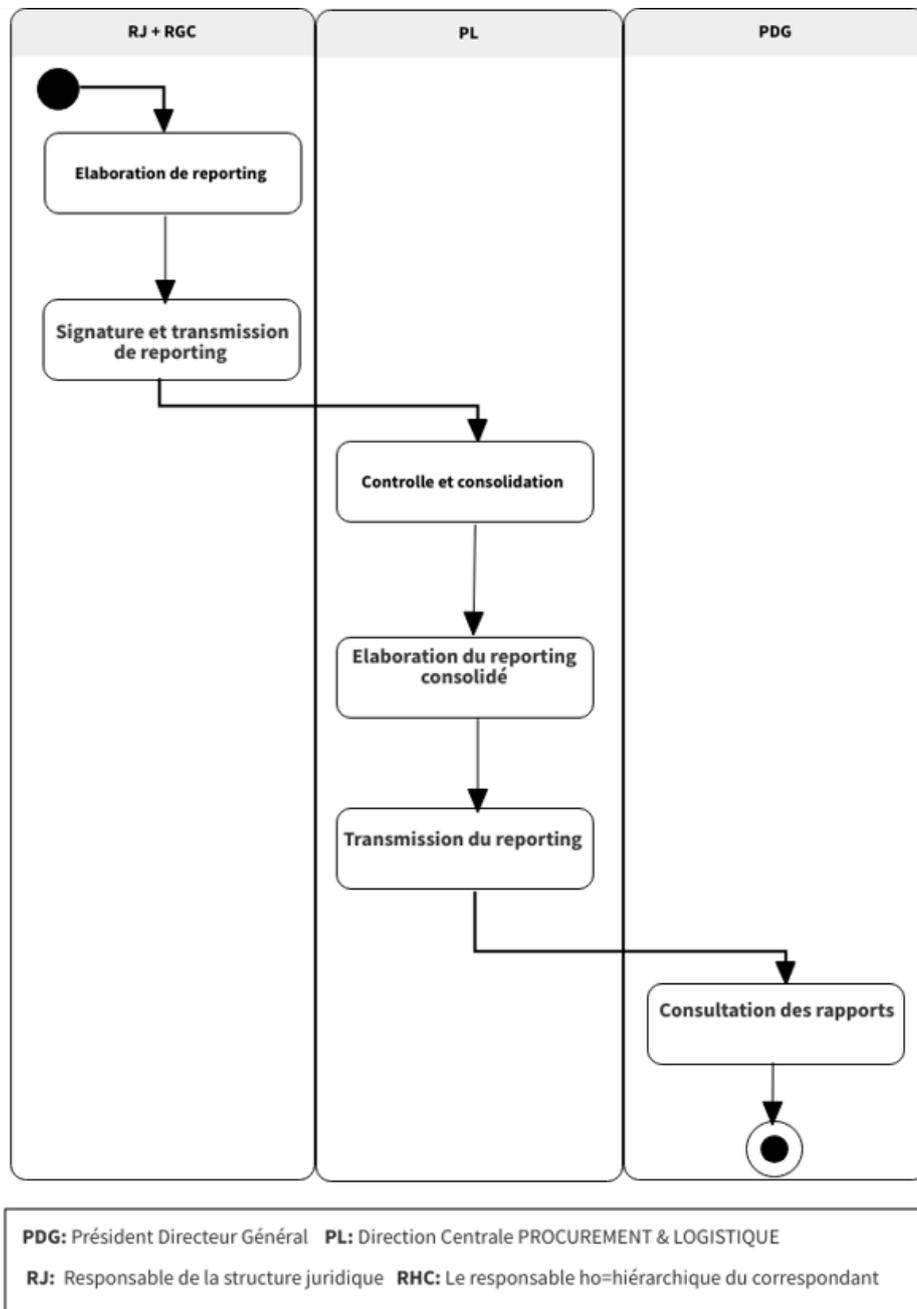


FIG. 2.4 : Diagramme d'Activité « L'élaboration et la transmission du Reporting trimestriel ».

2.2.6 Diagramme de classe d'analyse

Le diagramme de classes 2.5 suivante décrit l'ensemble des classes participantes à la fonctionnalité de gestion de contrat :

Selon la figure 2.5, la classe centrale contrat est associée à des classes telles que signataire, devise, mode de passation et nature budgétaire, etc. Des classes comme Entreprise, Association et Groupement représentent les différentes parties prenantes. D'autres classes fournissent des informations supplémentaires comme les types de documents, les statuts et les nationalités.

2.2.7 Description des classes

Classe	Description
Contrat	La classe Contrat représente la gestion des informations sur les contrats. Elle permet la création, la validation, la modification et l'affichage des détails et des listes de contrats.
Motif Gré à Gré	La classe Motif Gré à Gré représente la gestion des motifs spécifiques pour les contrats conclus sans appel d'offres formel, appelés contrats de gré à gré. Cette classe maintient les informations et l'état de ces motifs, permettant ainsi de contrôler et de valider les contrats basés sur ces motifs particuliers.
Mode Passation	La classe Mode Passation représente la gestion des différents modes de passation des contrats. Elle permet de définir et de modifier les modes utilisés pour passer des contrats.
Nature Budget	La classe Nature Budget représente la gestion des différentes natures de budget associées aux contrats. Elle permet de définir, modifier et obtenir des informations sur les types de budget.
Type Marché	La classe Type Marché représente la gestion des types de marché associés aux contrats. Elle permet de définir, modifier et obtenir des informations sur les types de marché.
Type Document	La classe Type Document représente la gestion des types de documents utilisés dans les contrats. Elle permet de définir, modifier et récupérer les types de documents nécessaires.
ContratTypeDocument	La classe ContratTypeDocument représente la gestion des documents spécifiques associés à chaque contrat. Elle maintient les liens entre les contrats et leurs documents.
Type Contrat	La classe Type Contrat représente la gestion des différents types de contrats. Elle permet de définir, modifier et obtenir des informations sur les types de contrats.
Structure	La classe Structure représente la gestion des structures organisationnelles. Elle permet de définir les structures parentales et de gérer les sous-structures.
Fonction Signataire	La classe Fonction Signataire représente la gestion des fonctions des signataires des contrats. Elle permet de définir et de modifier les fonctions des individus autorisés à signer des contrats.
ContratSignataire	La classe ContratSignataire représente la gestion des signataires spécifiques associés à chaque contrat. Elle maintient les liens entre les contrats et leurs Fonction Signataire.

Devise	La classe Devise représente la gestion des différents Devise associés aux contrats. Elle permet de définir, modifier et obtenir des informations sur les devises.
ContratDevice	La classe ContratDevice représente la gestion des dispositifs financiers spécifiques à chaque contrat. Elle inclut des informations sur les montants et les taux applicables.
Entreprise	La classe Entreprise représente la gestion des informations sur les entreprises. Elle inclut la dénomination sociale, l'adresse, le statut, la nationalité, etc. Elle permet aussi de modifier et de récupérer les entreprises.
Nationalité	La classe Nationalité représente la gestion des informations sur les nationalités. Elle permet de définir et de modifier les nationalités associées aux entreprises.
Statut	La classe Statut représente la gestion des statuts des entreprises. Elle permet de définir et de modifier les statuts pour indiquer des états tels qu'actif ou inactif.
Type Association	La classe Type Association représente l'énumération des types d'associations possibles pour les entreprises, comme COO ou regroupement.
Association	La classe Association représente la gestion des informations sur les associations d'entreprises. Elle inclut la dénomination, le type, et les taux d'association.
EnterpriseAssociation	La classe EnterpriseAssociation représente la gestion des associations spécifiques entre les entreprises et les taux d'association correspondants.
Groupement	La classe Groupement représente la gestion des types de groupements d'entreprises. Elle permet de définir et de modifier les types et les structures des regroupements.
EnterpriseGroupement	La classe EnterpriseGroupement représente la gestion des regroupements d'entreprises. Elle inclut les taux de regroupement.

2.3 Plateforme de gestion des contrats

2.3.1 Limites du Système Actuel

Le système actuel de gestion des contrats chez SONATRACH a bien rempli sa mission par le passé. Cependant, avec l'évolution constante des processus internes et des nouvelles exigences, il devient nécessaire de se doter d'un outil plus moderne et flexible. Si l'application existante reste fonctionnelle, elle commence à montrer ses limites pour s'adapter aux nouveaux besoins de l'entreprise. Par exemple, elle ne dispose pas d'une fonctionnalité clé comme la Business Intelligence et les tableaux de bord dynamiques, pourtant essentiels pour une gestion contractuelle efficace et proactive aujourd'hui. Un certain décalage commence à se créer entre les besoins grandissants de SONATRACH

et les fonctionnalités proposées. Cela peut rendre l'expérience utilisateur un peu moins fluide qu'attendu. Il devient aussi plus difficile pour les décideurs d'accéder aussi rapidement qu'ils le souhaiteraient aux informations clés pour prendre les meilleures décisions stratégiques.

En outre, le recours à une gestion manuelle des données, principalement à travers des rapports Excel et des graphiques élaborés manuellement, génère des inefficacités dans la transmission et l'analyse de l'information. Cette méthode entrave la réactivité de Sonatrach face aux fluctuations du marché et compromet son efficacité opérationnelle globale.

La nécessité d'une mise à niveau de l'application de gestion des contrats est impérative pour garantir la compétitivité et la performance opérationnelle de l'entreprise. En modernisant son infrastructure, SONATRACH pourra non seulement répondre aux exigences actuelles, mais aussi anticiper les besoins futurs, renforçant ainsi sa position sur le marché et assurant sa pérennité à long terme.

2.3.2 Besoins d'une Nouvelle Plateforme Intégrée

Pour répondre aux exigences croissantes de gestion des contrats, une nouvelle solution est nécessaire pour introduire une approche moderne et intuitive, mettant ainsi l'accent sur l'amélioration de l'expérience utilisateur et la transparence des opérations.

L'attention s'est portée sur deux aspects principaux : l'aspect fonctionnel et l'aspect analytique :

D'un point de vue fonctionnel, la solution fournira une plateforme flexible et conviviale pour la gestion des contrats, des avenants et des bons de commande. Des interfaces intuitives seront créées pour permettre aux utilisateurs de naviguer facilement dans le système, de saisir et de modifier les données en temps réel, tout en assurant un processus de validation fluide et efficace.

D'un point de vue analytique, la solution intégrera des fonctionnalités de Business Intelligence (BI) et de data visualization. Elle offrira aux décideurs des tableaux de bord dynamiques pour analyser les données contractuelles, identifier les tendances, les opportunités et les risques, et prendre des décisions stratégiques.

En résumé, la vision de la nouvelle solution est de créer un environnement numérique dynamique et efficace qui permettra à l'entreprise de relever les défis tout en ouvrant de nouvelles opportunités de croissance et d'innovation. Cela se fait en plaçant l'utilisateur au cœur de la conception et en mettant l'accent sur l'analyse des données.

2.4 Conclusion

Dans ce chapitre, nous avons présenté, de manière globale, le système de gestion des contrats utilisé par SONATRACH, mettant en évidence les nouveaux défis et justifiant la nécessité de mettre en place un nouveau système plus compatible avec les nouvelles exigences de l'entreprise.

Dans le prochain chapitre, nous aborderons la conception de notre projet.

Chapitre 3

Conception

3.1 Introduction

Dans ce chapitre, nous discuterons de l'analyse et de la conception de notre projet. Nous illustrerons l'architecture principale qui prend en charge notre solution informatique et examinerons les diagrammes UML pour représenter correctement les différents éléments et interactions de notre système. À partir de là, nous pourrions comprendre tous les besoins fonctionnels et non fonctionnels de notre projet.

3.2 Spécification des besoins du système

Dans cette section, nous allons identifier et séparer les exigences fonctionnelles et non fonctionnelles de notre système informatique. Cela est nécessaire pour comprendre exactement ce que le système doit réaliser et comment il doit fonctionner pour répondre aux attentes des utilisateurs.

3.2.1 Identification des acteurs

- **Administrateur fonctionnel de BDC** : Son rôle inclut l'interaction avec les structures fonctionnelles et les services juridiques des entités opérationnelles. Cela comprend la gestion des privilèges d'accès et la communication des autorisations à la base de données (BDC), aux correspondants et aux utilisateurs autorisés, garantissant ainsi une utilisation sûre et efficace du système.
- **Correspondant** : Premier acteur du processus d'alimentation de la Banque de Données.
- **Responsable hiérarchique du correspondant** : Est chargé de contrôler et valider les données saisies par le correspondant, assurant ainsi l'exactitude et la fiabilité des informations utilisées dans le système.
- **Décideur** : Son rôle implique la consultation des rapports et des tableaux de bord pour obtenir une vue d'ensemble de la performance de l'entreprise et surveiller les indicateurs clés.

3.2.2 Besoins fonctionnels

Dans cette section, nous représentons l'ensemble des besoins fonctionnels auxquels devrait répondre notre nouvelle solution.

Le système doit permettre :

1. **Gestion des Utilisateurs et des Rôles** : assurer l'accès sécurisé aux fonctionnalités de l'application selon des profils.
2. **Gestion des référentiels** : Mettre en place un module de gestion des référentiels pour centraliser et organiser les informations essentielles de manière efficace.

3. **Gestion des Contrats, Avenants et Bons de Commande** : Créer des interfaces conviviales pour la gestion des contrats, des avenants et des bons de commande, en permettant leur création, modification et suivi en temps réel, avec un workflow de validation.
4. **Intelligence d’Affaires et Data Visualization** : Intégrer des capacités de reporting à plusieurs niveaux, avec des tableaux de bord personnalisables, des visualisations interactives pour analyser les performances des contrats.
5. **Intégration et Importation Automatique** : mettre en place un mécanisme d’importation et d’intégration automatique des données.
6. **Historique des Modifications et Notifications** : Créer un système de suivi des modifications apportées aux contrats et mettre en place des notifications automatiques par e-mail pour les parties prenantes concernées.

3.2.3 Besoins non fonctionnels

Il s’agit des critères qui caractérisent le système. Ceux-ci incluent des besoins en matière de performance, de conception, ainsi que d’autres aspects qualitatifs du système.

L’ensemble des extensions à réaliser doivent respecter les exigences suivantes :

- **Ergonomie de l’interface** : L’application doit être facile à utiliser, les interfaces utilisateurs doivent être conviviales c’est-à-dire simples, ergonomiques et adaptées à l’utilisateur.
- **Fiabilité** : Les informations apportées par l’application doivent être fiables et sûres
- **Disponibilité** : L’application doit être disponible à tout instant pour être utilisée
- **Sécurité** : L’application comporte des informations personnelles et sensibles, donc elle doit respecter les règles relatives à la sécurité des systèmes informatiques.
- **La performance** : L’application doit être performante c’est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d’une manière optimale

3.3 Modélisation des besoins fonctionnels

3.3.1 Diagrammes des cas d’utilisation

Dans cette section, nous allons détailler les diagrammes de cas d’utilisation réalisés.

La figure 3.1 ci-dessous, représente le diagramme des cas d’utilisation général de notre système.

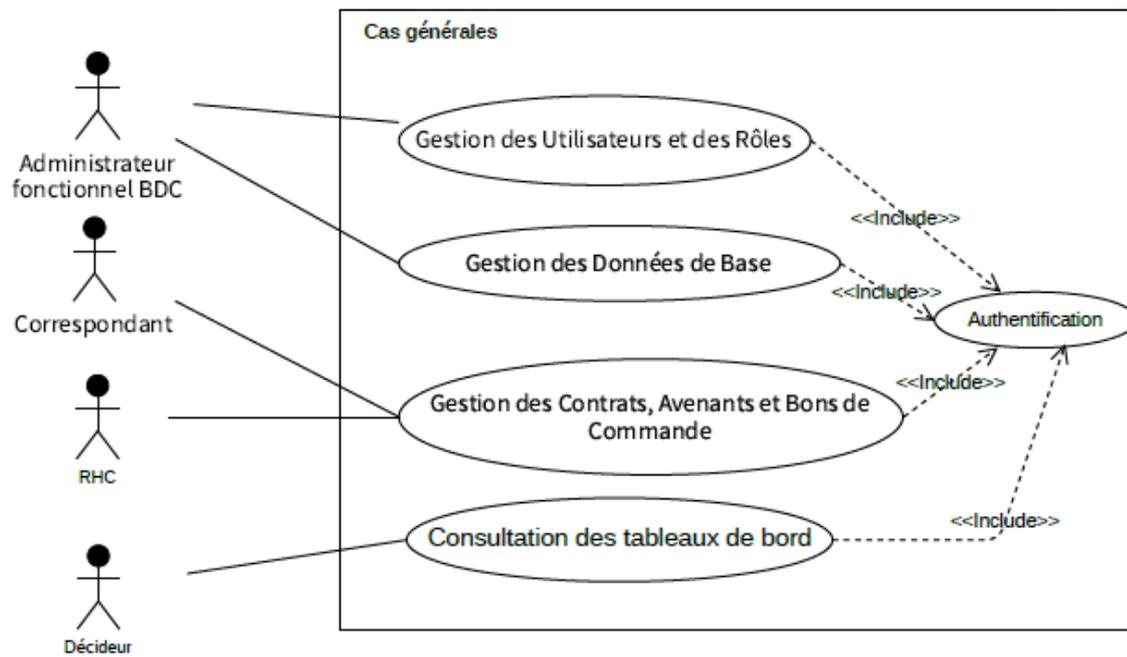


FIG. 3.1 : Diagramme de cas d'utilisation générale du système.

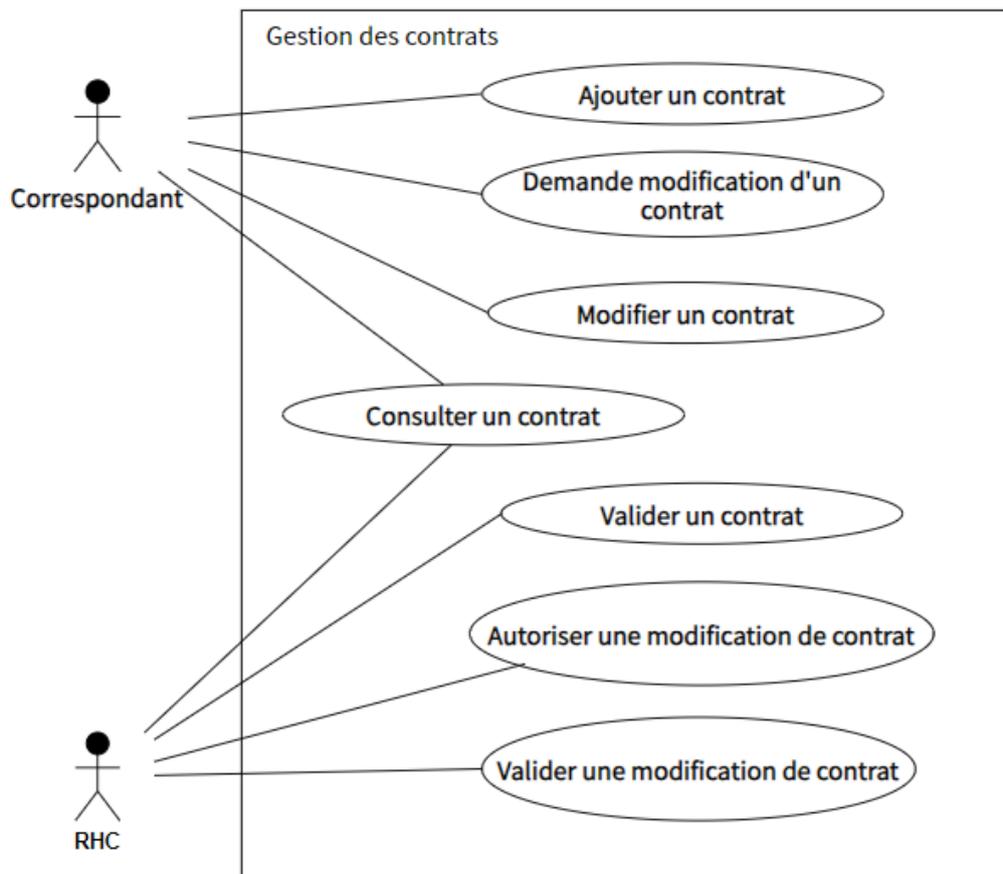


FIG. 3.2 : diagramme de cas d'utilisation « Gestion du contrats ».

La figure 3.2 ci-dessus, illustre le diagramme des cas d'utilisation de gestion du contrat

3.3.2 Diagrammes de séquence

3.3.2.1 Premier cas « Création d'un contrat »

Ce cas d'utilisation illustre comment le correspondant crée un nouveau contrat dans le système, en saisissant les informations essentielles telles que les parties impliquées et les conditions contractuelles.

Description textuelle :

Le tableau représente la description textuelle du cas d'utilisation « Création d'un contrat »

Description de cas « Création d'un contrat »
Titre : Création d'un contrat.
Résumé : Le correspondant saisit les informations essentielles pour créer un nouveau contrat dans le système.
Acteurs : Correspondant
Préconditions : Le correspondant doit être authentifié et avoir accès aux fonctionnalités de création de contrat, et les informations nécessaires pour la création du contrat doivent être disponibles et valides dans le système.
Postconditions : Un nouveau contrat est enregistré dans le système avec les informations fournies par le correspondant, prêt à être validé par le responsable hiérarchique.
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1. Le correspondant s'authentifie auprès du système de gestion des contrats. 2. Une fois authentifié, le correspondant crée un nouveau contrat en fournissant les informations nécessaires. 3. Le système vérifie que les informations obligatoires sont présentes dans le contrat. 4. Si les données sont valides, le système ajoute le contrat à la base de données. 5. Une notification est envoyée pour informer le responsable hiérarchique du correspondant de la création du contrat.
Scénarios alternatifs : Si des informations essentielles sont manquantes lors de la création du contrat, le système génère une alerte pour informer le correspondant.

TAB. 3.1 : Description de cas « Création d'un contrat ».

Diagramme de séquence « Création d'un contrat »

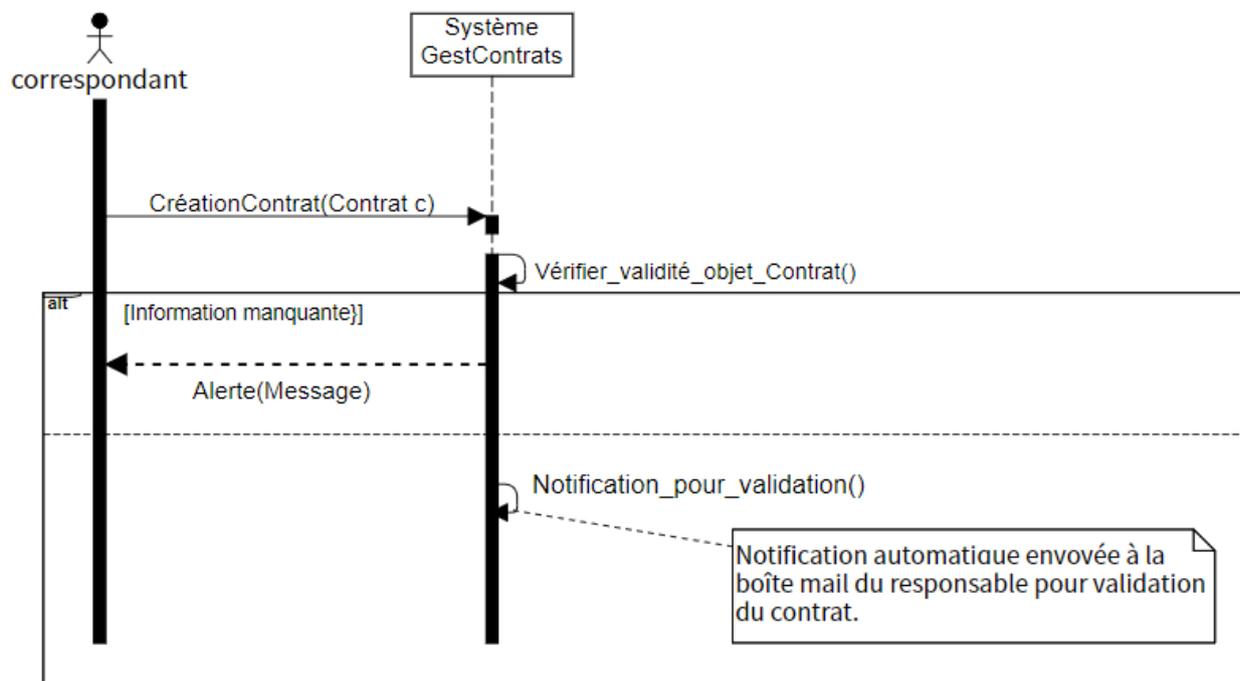


FIG. 3.3 : Diagramme de séquence « Création d'un contrat ».

3.3.2.2 Deuxième cas « Validation d'un contrat »

Ce cas d'utilisation explique comment le responsable hiérarchique du correspondant approuve ou refuse un contrat soumis par celui-ci. Il précise les différentes étapes à suivre pour étudier le contrat, prendre une décision de validation et informer le correspondant de la décision prise.

Description textuelle

Le tableau représente la description textuelle du cas d'utilisation « Validation d'un contrat »

Description de cas « Validation d'un contrat »
Titre : Validation d'un contrat.
Résumé : Ce cas d'utilisation décrit le processus par lequel le responsable hiérarchique du correspondant valide ou rejette un contrat soumis par le correspondant.
Acteurs : Responsable hiérarchique du correspondant.
Préconditions : le responsable hiérarchique doit être authentifié auprès du système de gestion des contrats. De plus, le responsable hiérarchique doit avoir l'autorisation de validation. Le contrat soumis, qui nécessite la validation du responsable hiérarchique, doit être disponible dans le système.
Postconditions : Le contrat est validé par le responsable hiérarchique du correspondant, selon les critères définis.

Scénario nominal :

1. Le responsable hiérarchique examine les détails du contrat.
2. Si le contrat est jugé valide, le responsable hiérarchique le valide.
3. Après la validation, l'état du contrat est mis à jour dans la base de données pour indiquer qu'il est désormais valide.

Scénarios alternatifs : Si le contrat nécessite des corrections, le responsable hiérarchique peut demander des corrections au correspondant.

TAB. 3.2 : Description de cas « Validation d'un contrat »

Diagramme de séquence « Validation d'un contrat »

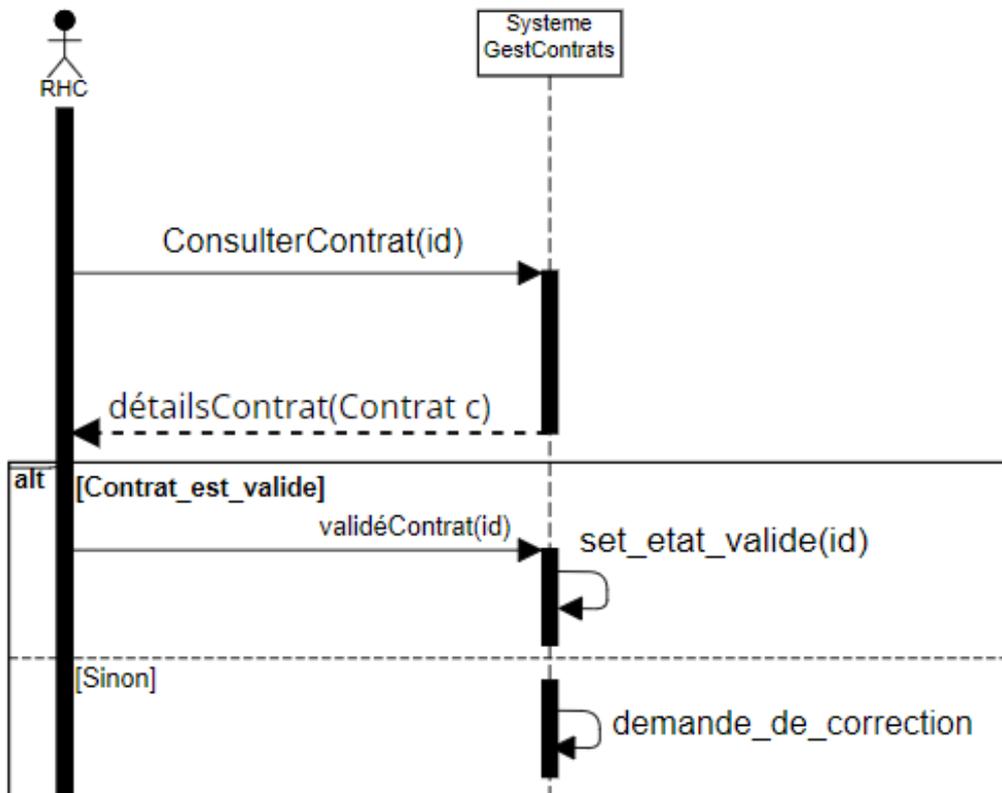


FIG. 3.4 : Diagramme de séquence « Validation d'un contrat ».

3.3.2.3 Troisième cas « Modification d'un contrat »

Ce cas d'utilisation décrit comment le correspondant modifie un contrat existant, y compris l'identification des modifications, leur soumission et leur validation par le responsable hiérarchique.

Description textuelle

Le tableau représente la description textuelle du cas d'utilisation « Modification d'un contrat »

Description de cas « Modification d'un contrat »
Titre : Modification d'un contrat.
Résumé : Ce cas d'utilisation décrit le processus de modification d'un contrat existant dans le système de gestion des contrats.
Acteurs : Correspondant
Préconditions : Le correspondant doit être authentifié et avoir accès aux fonctionnalités de modification de contrat. Le contrat à modifier doit être présent dans le système.
Postconditions : Les modifications demandées sont enregistrées dans le système et soumises à la validation du responsable hiérarchique du correspondant.
Scénario nominal : <ol style="list-style-type: none">1. Le correspondant consulte le contrat qu'il souhaite modifier.2. Après examen, le correspondant demande la modification du contrat.3. Le système enregistre la demande de modification et notifie le responsable hiérarchique du correspondant.4. Le responsable hiérarchique examine la demande de modification.5. Si la demande est autorisée, le correspondant modifie le contrat conformément à la demande.6. Une fois les modifications effectuées, le correspondant soumet les changements pour validation.7. Le responsable hiérarchique examine les modifications proposées.8. Si les modifications sont validées, le système enregistre les changements dans la base de données.
Scénarios alternatifs : Si les modifications proposées ne sont pas validées par le responsable hiérarchique, elles sont rejetées.

TAB. 3.3 : Description de cas « Modification d'un contrat »

Diagramme de séquence « Modification d'un contrat »

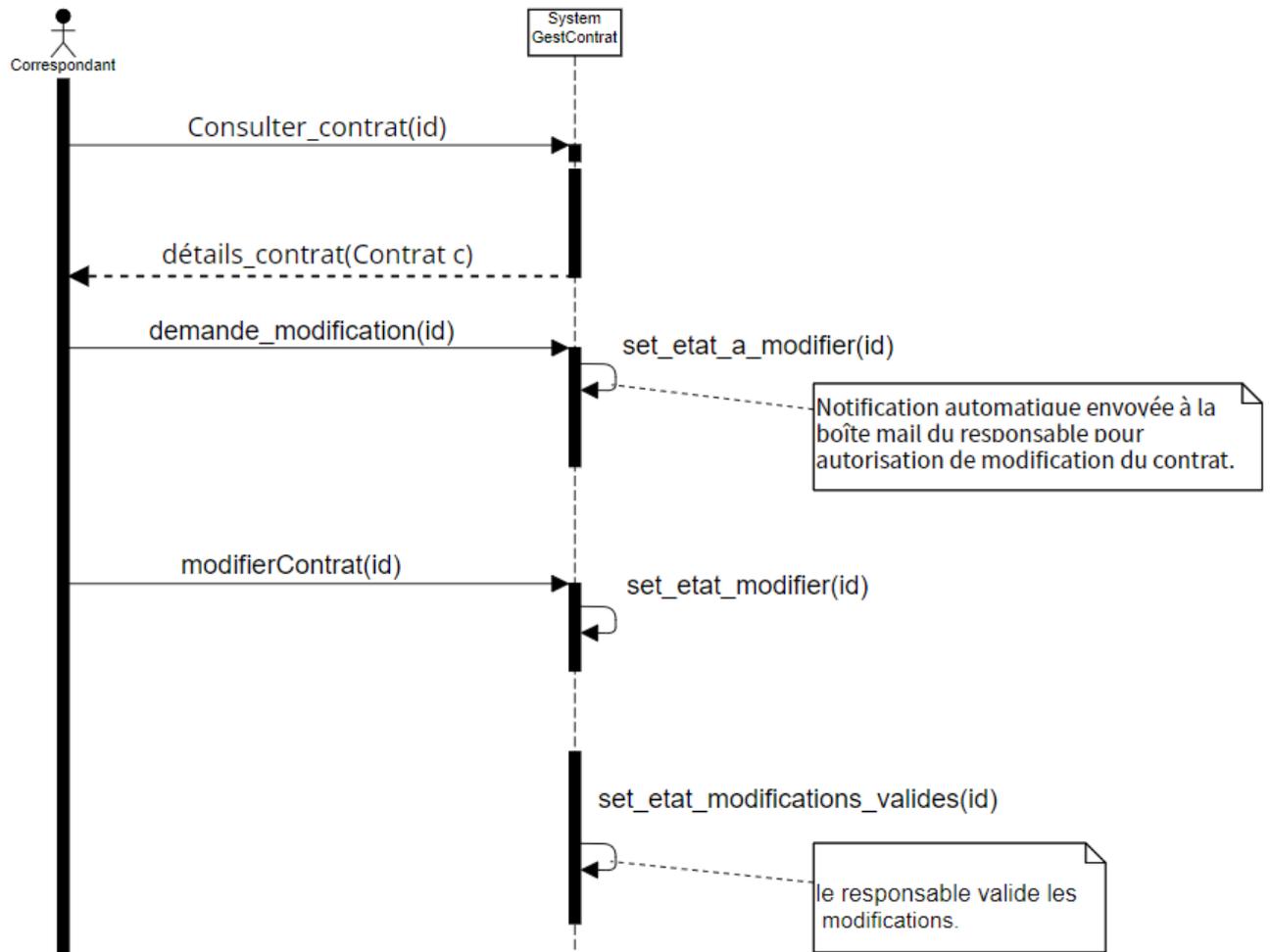


FIG. 3.5 : Diagramme de séquence « Modification d'un contrat ».

Comme le montrent les diagrammes de séquences précédents. Un contrat suit tout un processus avant d'être définitivement validé. Il passe par différents stades, avec parfois des retours en arrière pour des ajustements.

Remarque : Les diagrammes de cas d'utilisation et de séquences sont illustrés pour le cas des contrats. Il est important de noter que les processus et interactions décrits pour les contrats sont également applicables aux avenants et aux bons de commande, avec des adaptations mineures dues à la nature similaire de ces entités dans le système de gestion.

Le diagramme d'état 3.6 ci-dessous résume les différents états que peut prendre un contrat, de sa création initiale jusqu'à sa version finale. Ça permet de bien visualiser le cheminement complet.

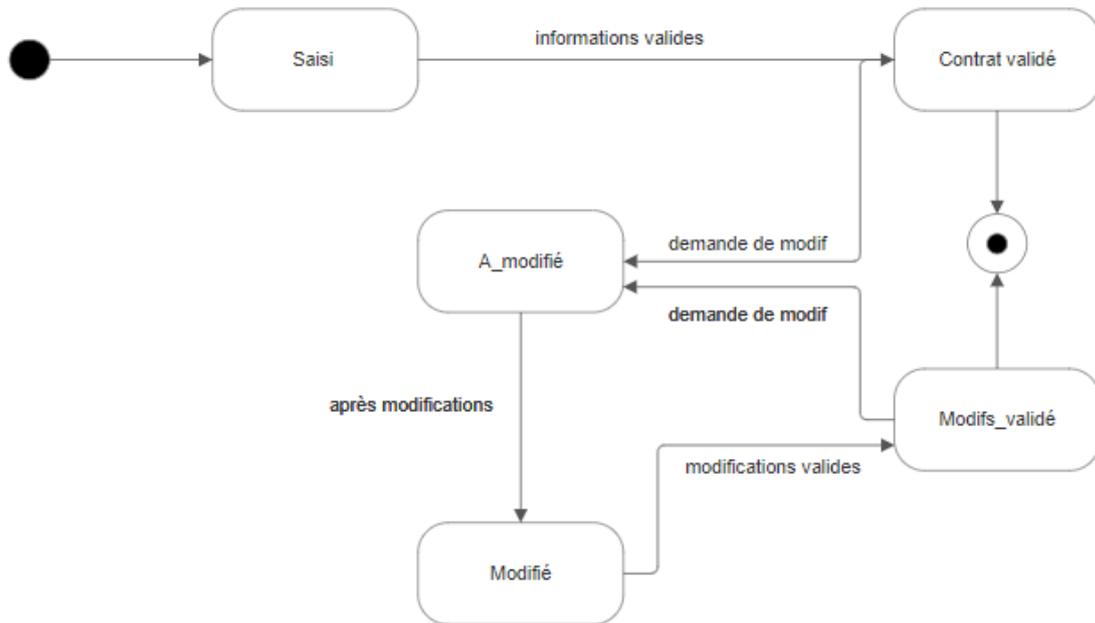


FIG. 3.6 : Diagramme d'état transition « État contrat ».

3.3.3 Diagramme de paquetage

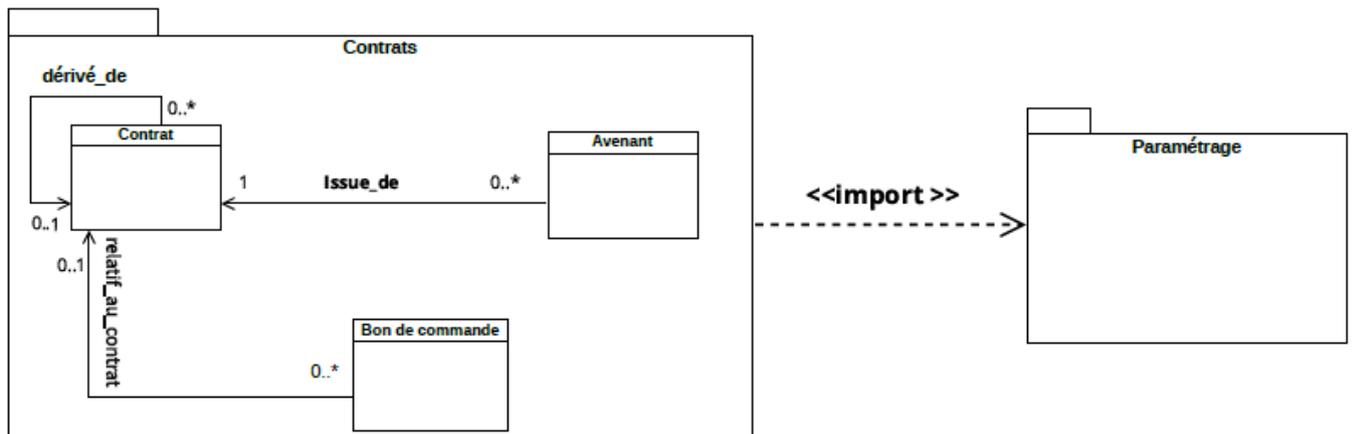


FIG. 3.7 : Diagramme de paquetage globale.

À partir de ce diagramme de paquetage 3.7, nous voyons la disposition des classes contrat, avenant et bon de commande, ainsi que le package paramétrage, offrant une vue claire de l'architecture globale du système.

3.3.4 Diagramme de classe

Le diagramme de classe 3.8 suivant montre les différentes méthodes métiers utilisées pour la nouvelle application de gestion des contrats.

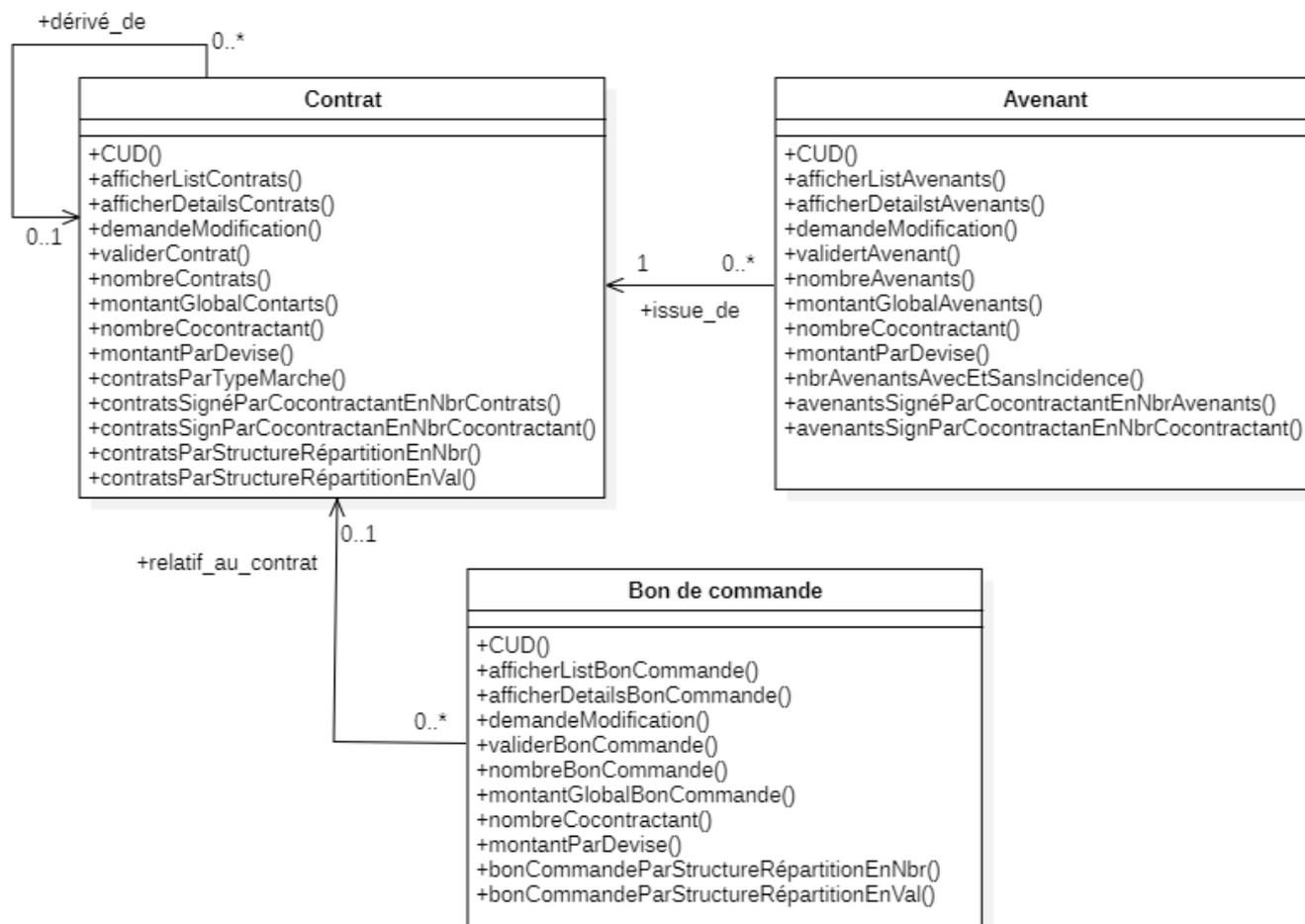


FIG. 3.8 : Diagramme de classe.

3.4 Business Intelligence et Data Visualization

La Business Intelligence (BI) est un ensemble de processus et de technologies utilisés pour transformer les données brutes en informations utiles à la prise de décision. Dans le contexte de notre projet, la BI joue un rôle crucial en fournissant des informations précieuses qui permettent de gérer efficacement les contrats, avenants et bons de commande.

3.4.1 Objectifs de la BI dans notre système

Les principaux objectifs de la BI dans notre système sont :

- **Optimisation des processus** : Améliorer l'efficacité des processus de gestion des contrats, avenants et bons de commande en fournissant des informations claires et précises.

- **Prise de décision informée** : Aider les décideurs à prendre des décisions basées sur des données concrètes et à jour.
- **Transparence** : Augmenter la transparence des opérations de gestion des contrats grâce à des visualisations claires des données.

3.4.2 Conception des KPIs (Indicateurs de Performance Clés)

La conception de nos KPIs s'appuie sur les objectifs stratégiques de l'organisation et les besoins des utilisateurs. Voici les étapes clés de la conception des KPIs :

- **Identification des besoins** : Recueillir les besoins des utilisateurs finaux pour déterminer quelles métriques sont les plus pertinentes.
- **Définition des KPIs** : Définir les KPIs en fonction des besoins identifiés. Par exemple :
 - Nombre de contrats signés et leur montant global.
 - Répartition des contrats par type marché et par structure.
 - Nombre d'avenants par incidence financière.
 - Taux des bons de commande en dinars vs en devises.
 - Nombre de cocontractants.

Ainsi que d'autres KPIs pertinents pour l'analyse et la gestion des contrats.

3.4.3 Conception des Tableaux de Bord

Sur la base de ces besoins, nous avons conçu une architecture modulaire de tableaux de bord comprenant :

- Des tableaux de bord dédiés au suivi détaillé des contrats.
- Des tableaux de bord pour le suivi des bons de commande.
- Des tableaux de bord pour le suivi des avenants.

Pour chaque tableau de bord, nous avons déterminé les visualisations les plus appropriées (camemberts, barres, courbes, jauges, etc.) en fonction des données à représenter et des cas d'utilisation identifiés.

3.4.4 Conception de l'interaction

Au-delà de la représentation visuelle des données, ces tableaux de bord seront dotés d'interactions permettant aux utilisateurs d'explorer de manière intuitive :

- Navigation par onglets entre les différents tableaux de bord.
- Affichage des détails au survol des éléments visuels.
- Fonctionnalités de filtrage.

3.4.5 Intégration dans la plateforme

Les tableaux de bord s'intègrent de manière transparente dans l'architecture globale de la plateforme, offrant ainsi une expérience utilisateur cohérente et unifiée.

3.5 Conclusion

En somme, ce chapitre a permis d'examiner les différents aspects d'analyse et de conception de solutions logicielles. Grâce aux diagrammes UML, nous avons pu créer une modélisation complète des exigences fonctionnelles et non fonctionnelles. À partir de là, les bases d'un système efficace, sécurisé et évolutif ont été posées pour commencer l'implémentation.

Chapitre 4

Implementation

4.1 Introduction

Ce chapitre porte sur les aspects pratiques liés à la réalisation concrète. Il présente les diverses ressources employées pour bâtir notre solution informatique, détaillant leur rôle à l'aide d'illustrations explicatives.

4.2 Vue globale de la plateforme

Avant de commencer à détailler les étapes de programmation par lesquelles notre système est passé, nous devons présenter le schéma qui comprend les différentes parties qui composent notre solution logicielle. Il donne un aperçu de la structure et des principaux modules de la plateforme de gestion des contrats, facilitant ainsi la compréhension de l'ensemble du système.

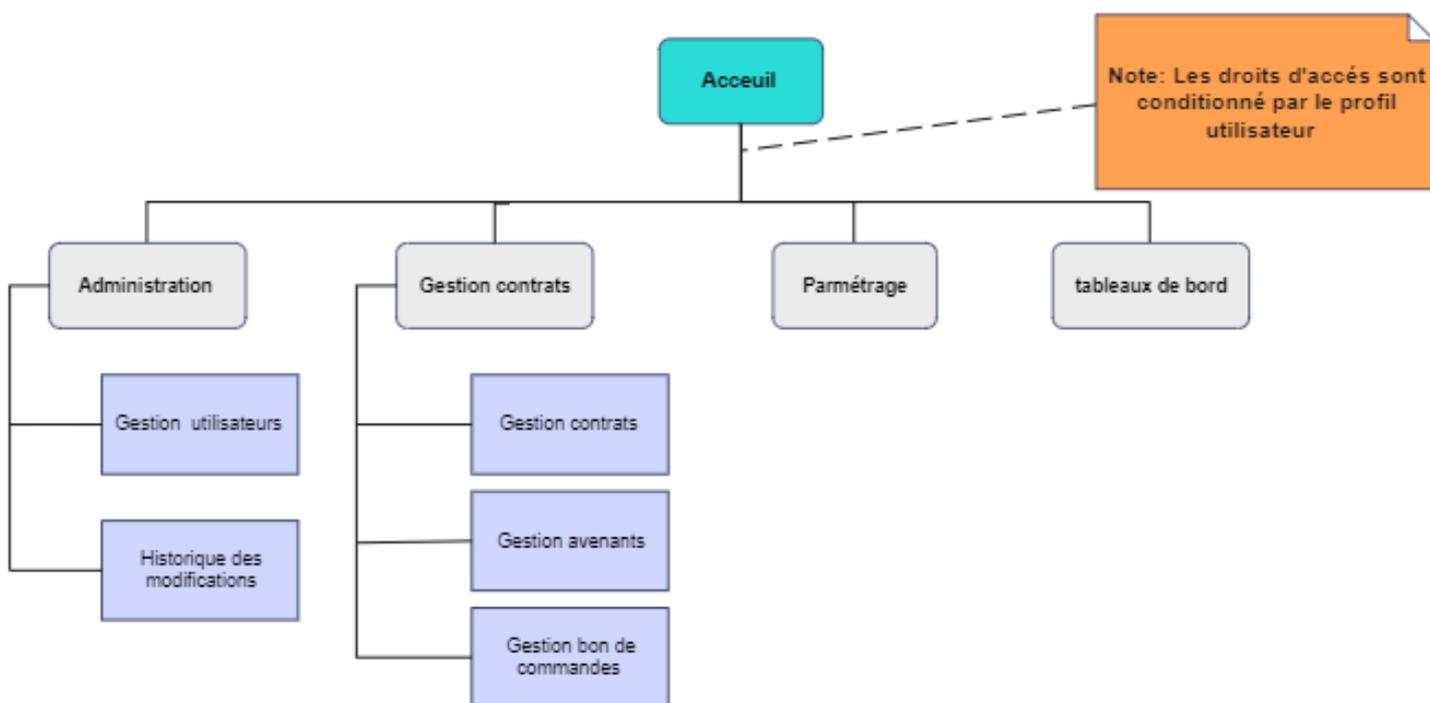


FIG. 4.1 : Vue globale systeme.

Le graphique de La figure 4.1 montre les différentes fonctionnalités disponibles pour les utilisateurs une fois connectés à la page d'accueil. Les droits d'accès dépendent du profil de l'utilisateur qui détermine ce qu'une personne peut faire dans le système, en fonction de son rôle.

Le système se décompose en quatre modules :

1. **Administration** : Gestion des utilisateurs, de leurs comptes/rôles, de leur historique des connexions/créations et du journal d'activité des contrats.
2. **Gestion des contrats** : Module central dédié à la gestion des contrats, la gestion des avenants et des bons de commande associés.

3. **Paramétrage** : Regroupe l'ensemble des informations nécessaires à la gestion des contrats (entreprises, classes contrats, etc.).
4. **Tableaux de bord** : Fournissent des visualisations graphiques affichant les mesures et indicateurs clés.

Ainsi, le processus de maintenance du système et son évolutivité est réalisé grâce à la séparation claire des responsabilités incluses dans cette architecture standard. Une fois authentifié, toutes les fonctionnalités sont accessibles depuis la page d'accueil, mais selon les autorisations pour assurer l'intégrité et la sécurité des données. Cela permet une gestion optimale des accès aux données.

4.3 Mise en œuvre de la solution

4.3.1 Vue d'ensemble de l'architecture

La figure 4.2 ci-dessous résume de façon graphique l'architecture de notre application :

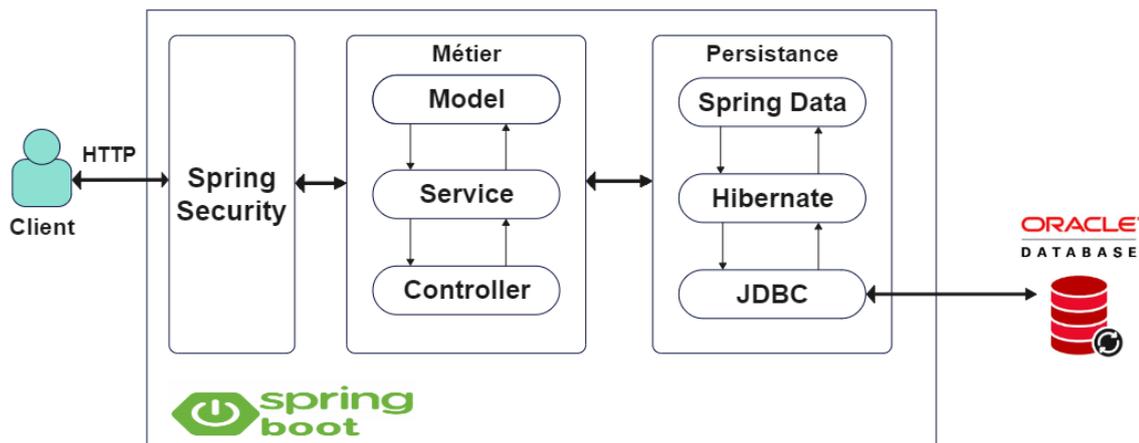


FIG. 4.2 : Vue d'ensemble de l'architecture.

4.3.2 Environnement de développement

La figure 4.3 montre les principaux outils utilisés dans le développement de notre système :



FIG. 4.3 : Outils de développement.

4.3.2.1 Langues de programmations



JAVA

Java est un langage de programmation et une plate-forme de calcul lancé par Sun Microsystems en 1995. Depuis ses débuts modestes, Java a beaucoup évolué. A l'heure actuelle, de nombreux services et applications reposent sur cette plate-forme fiable.[6]

Les principales caractéristiques de Java sont :

- Approche orientée objet.
- Portabilité grâce à la machine virtuelle Java.
- Robustesse avec une gestion automatique de la mémoire.
- Support du multi-tâche (threads).
- Riche écosystème de bibliothèques et frameworks.[7]

Java a été choisi en raison de son vaste écosystème de développement, de son support multimodal avancé et de sa sécurité renforcée, répondant ainsi aux exigences de SONATRACH en termes d'évolutivité et de fiabilité du développement pour sa plateforme.

HTML, CSS et JavaScript

Pour développer l'interface utilisateur, nous avons utilisé HTML, CSS et JavaScript.



FIG. 4.4 : Architecture technique.

4.3.2.2 Frameworks et bibliothèques



SpringBoot

Spring Boot est un framework open source basé sur Java, visant à simplifier le processus de configuration et de déploiement des applications Spring.

Ses principales caractéristiques sont :

- Configuration automatique des dépendances et composants Spring.
- Démarrage rapide grâce à un serveur web embarqué (Tomcat, Jetty, etc.).
- Outil de monitoring et de gestion de l'application en production.
- Support du rechargement à chaud facilitant le développement itératif.
- Architecture basée sur le principe de "convention plutôt que configuration".
- Vaste écosystème de "starters" permettant d'inclure facilement des bibliothèques tierces.[8]

Spring Boot a été sélectionné pour son approche de configuration automatique, sa facilité de déploiement et son écosystème riche en "starters". Son adoption répond aux besoins de SONATRACH en matière de développement agile et de gestion simplifiée des applications.



Thymeleaf

Thymeleaf est un moteur de template web moderne côté serveur pour les environnements web et autonomes en Java.

L'objectif principal de Thymeleaf est d'apporter des templates naturels et élégants au flux de travail de développement du HTML pouvant être correctement affiché dans les navigateurs et également servir de prototypes statiques, permettant une meilleure collaboration au sein des équipes de développement.[9]



Bootstrap

Bootstrap est l'un des frameworks les plus populaires pour le développement de sites Web. L'outil fournit des modèles pour CSS et HTML avec lesquels il est possible de mettre en place une conception Web moderne de manière relativement simple. La mise en page, les polices, les boutons et les divers éléments de navigation peuvent être créés et positionnés très simplement.

Développé à l'origine pour Twitter, le framework est disponible depuis un certain temps déjà en open source gratuitement pour tous.[10]



Amcharts

AmCharts est une bibliothèque JavaScript permettant de créer des graphiques modernes et interactives, telles que : des graphiques à barres, des lignes, des

diagrammes circulaires, des cartes, etc. Développée depuis 2006, elle propose une riche palette d'éléments visuels prêts à l'emploi et personnalisables. Cette bibliothèque facilite l'intégration de données avancées dans des applications Web, quelle que soit leur complexité.[11]

Les principales caractéristiques de AmCharts sont :

- Sa modélisation en fonction du tableau de bord du projet.
- C'est un choix adaptable.
- Les préférences de prise en charge multi-navigateurs.
- Les options de personnalisation élargies et le simple processus d'insertion.

4.3.2.3 outils de développement



IntelliJ IDEA

IntelliJ IDEA est un environnement de développement intégré (IDE) pour Java et Kotlin conçu pour maximiser la productivité des développeurs. Il effectue les tâches routinières et répétitives à leur place en fournissant une complétion de code intelligente, une analyse de code statique et des refactorisations, ce qui le rend non seulement productif mais aussi une expérience agréable.[12]



Oracle Database 21c

Oracle Database, un SGBDR performant et populaire, qui accompagne les entreprises de toutes tailles dans la gestion de leurs données.

La version 21c (2023) offre :

- Performances et évolutivité accrues pour les charges de travail exigeantes.
- Sécurité et protection des données renforcées contre les menaces actuelles.
- Analyses avancées pour une meilleure exploitation des données.
- Support du cloud hybride et multicloud pour une flexibilité optimale.[13]

Nous avons sélectionné Oracle Database en raison de sa réputation de performance, de sécurité et de flexibilité, alignée avec les exigences et les standards de SONATRACH.



Postman

Postman est une plateforme permettant de concevoir, tester et documenter des requêtes API. Elle facilite leur développement en offrant une interface pour envoyer des requêtes et visualiser les réponses. On peut collaborer sur des collections d'API et elle propose des outils de test et de débogage. En bref, un outil complet pour la gestion du cycle de vie des API.[14]

4.3.2.4 Outils de build

Maven

Maven, mot yiddish signifiant "accumulateur de connaissances", est un outil pour construire et gérer des projets Java. Son objectif principal est de permettre au développeur de comprendre l'état complet d'un effort de développement rapide. Il facilite le processus de construction, fournit un système uniforme et des informations de qualité sur le projet. Il encourage l'adoption de meilleures pratiques de développement, comme l'exécution et le rapport des tests unitaires. Il fournit aussi des directives sur la structure de répertoires du projet pour faciliter la navigation. Bien que Maven soit flexible, il ne peut pas s'adapter à toutes les situations sans compromettre ses objectifs.[15]

4.3.2.5 API

JDBC

c'est une API (Application Programming Interface) qui permet l'accès à des données de type table de bases de données SGBD relationnelles, permettant :

- D'ouvrir une connexion avec le SGBD.
- D'envoyer des requêtes SQL au SGBD.
- De récupérer les données renvoyées par les requêtes SQL.
- Et de traiter ces données "tableaux".
- De gérer les erreurs renvoyées par les requêtes au SGBD.[16]

Dans notre cas, nous utilisons le pilote JDBC spécifique à Oracle pour permettre une intégration harmonieuse et des opérations adaptées au système SGBD.

JPA

Java Persistence API, ou JPA, est une API standard Java qui fournit une interface pour gérer la persistance de données dans une application Java. Il présente un moyen commun et systématisé pour le développement d'applications de bases de données orientées objet. Il simplifie l'accès aux banques de données relationnelles pour les applications Java en fournissant un système objet.[17]

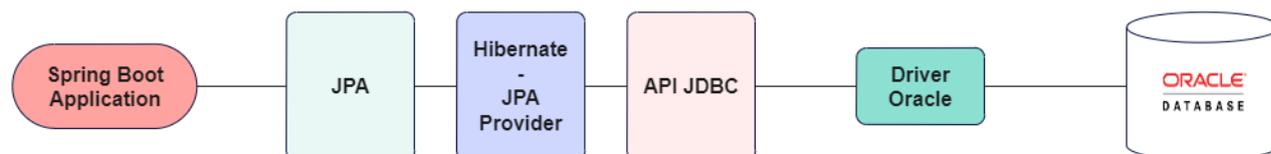


FIG. 4.5 : Flux de données Spring Boot et Oracle Database.

JavaMail

JavaMail est une API utilisée pour créer, écrire et lire des messages électroniques (e-mails). Il fournit un cadre indépendant du protocole et de la plate-forme pour l'envoi et la réception d'e-mails.

4.4 Méthodologie de développement appliquée

1. **Initialisation du projet** : Nous avons configuré le projet en utilisant Spring Initializr et ajouté les dépendances clés pour notre application de gestion de contrats. Les dépendances essentielles comprennent Spring Boot Starter Data JPA pour la gestion des données, Spring Boot Starter Thymeleaf pour les vues dynamiques, etc. Cette configuration assurera le bon fonctionnement des principales fonctionnalités de notre application.
2. **Configuration de la structure** : Mise en place de la structure du projet avec les répertoires pour les contrôleurs, les modèles, les vues, les services, etc., comme représenté dans la figure 4.6 ci-jointe.

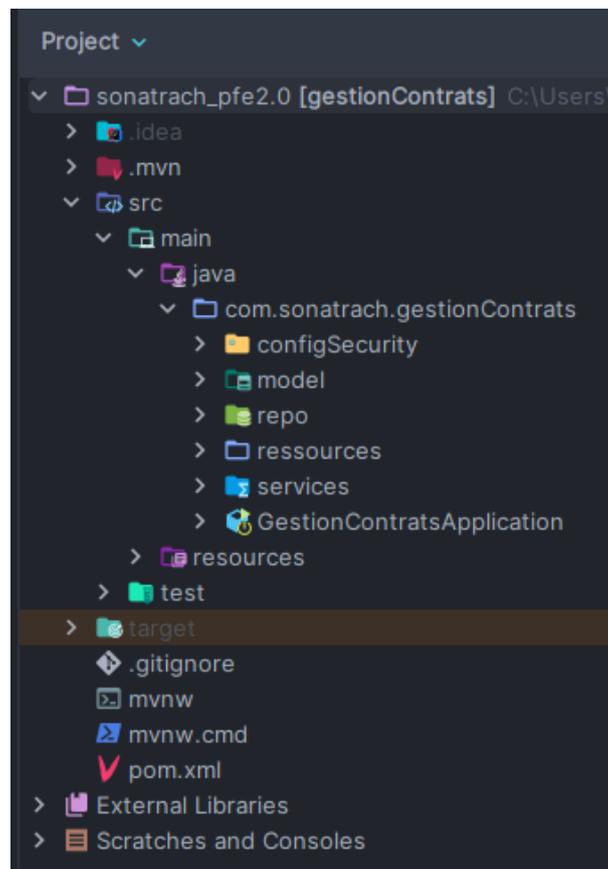


FIG. 4.6 : La structure des fichiers et répertoires du projet.

3. **Création des modèles** : Définir des modèles de données pour représenter les contrats, avenants, bons de commande, etc. à l'aide de classes Java.
4. **Implémentation des services** : Écriture des services Spring pour gérer la logique métier, l'accès aux données et les opérations sur les contrats.

5. **Développement des contrôleurs** : Création des contrôleurs Spring Boot pour gérer les requêtes HTTP et interagir avec les services.
6. **Tests Unitaires et d'Intégration** : Effectuer des tests approfondis pour chaque composant de l'application pour garantir son bon fonctionnement. Les tests sont réalisés sous Postman afin de simuler les requêtes HTTP et de vérifier les réponses attendues.
7. **Gestion des Vues avec Thymeleaf** : Création des vues HTML avec Thymeleaf pour afficher les données et interagir avec l'utilisateur.

4.5 Stratégie de migration de données

Afin de pouvoir utiliser la nouvelle application, une migration de données est nécessaire pour pouvoir l'exploiter. Cette action importante sera appliquée sur plusieurs étapes qui seront expliquées en détail dans ce qui suit, en commençant tout d'abord par les problèmes de l'ancien schéma :

4.5.1 Problèmes du schéma de l'ancienne base de données

1. **Clés primaires non automatiquement incrémentées** : Certaines tables utilisent des champs tels que la référence comme clé primaire, ce qui peut entraîner des contrats avec la même référence.
2. **Stockage des fichiers PDF** : Dans l'ancienne version de l'application gestion des contrats, les fichiers PDF des contrats y compris les avenats et les bons de commande étaient stockés dans la base de données comme des blobs. Certains contrats peuvent avoir une taille énorme (plus de 400 Mo), ce qui causait un problème de saturation rapide de l'espace alloué à la base de données, et sur la performance de l'application elle-même.
3. **Tables associatives et données JSON** : Certaines relations plusieurs-à-plusieurs étaient stockées comme des données JSON dans les tables, comme la relation entre Entreprise et Groupement. Cela rendait le schéma de la base de données plus complexe et difficile à maintenir.
4. **Ajouts de données pour l'enrichissement et le BI** : Des données telles que le taux de conversion de devises a été ajoutées pour enrichir l'application et permettre l'utilisation de la business intelligence, une fonctionnalité qui n'existait pas dans la version précédente.

4.5.2 Résolutions des problèmes

1. Toutes les tables de la nouvelle base de données auront un champ 'id' ajouté comme clé primaire, et la règle d'unicité sera appliquée sur la référence.

2. Les fichiers seront stockés dans des dossiers sur le serveur et non pas dans la base de données, ça permettra d'améliorer les performances de l'application et de minimiser la taille consommée dans la base de données.
3. Certaines tables associatives ont été créées pour remplacer le stockage de ces données JSON dans les tables. Par exemple, la table associative entre Entreprise et Groupement. La création de ces tables associatives permettra d'avoir un schéma simple et facilement maintenable, et de faciliter la génération des tableaux de bord.

4.6 Migration

La migration des données se fera en plusieurs étapes comme suit :

1. **Copie de l'ancienne base de données** : avant de commencer la migration des données une copie de la base de données production a été réalisée dans une base de test où cette nouvelle application sera connectée.
2. **Tables intouchables** : étant donné que certaines tables de l'ancienne base de données sont les mêmes dans la nouvelle base (elles ont les mêmes champs), ces tables ne subiront aucun changement, cela concerne généralement les tables de paramétrage.
3. **L'ajout d'un champ "id"** : dans les tables telles que contrats, avenants et bons de commande, qui serait automatiquement généré et incrémenté, permettrait une meilleure traçabilité et une identification unique de chaque élément.
4. **Chargement des fichiers** : Afin de pouvoir extraire les fichiers PDF de la base de données où ils sont stockés en tant que BLOB, puis conserver une trace de leur emplacement, nous suivons ces étapes :
 - (1) Création d'un dossier contrat qui contiendra tous les fichiers PDF extraits de la base.
 - (2) Création d'une colonne chemin fichier qui contiendra le chemin du fichier contrat.
 - (3) Extraire le fichier PDF du contrat qui est stocké dans un champ blob dans la table contrat.
 - (4) Enregistrer le nouveau chemin dans la table contrat.

Ces étapes peuvent être réalisées avec des scripts PL/SQL.

5. **Ajout du taux dans la table 'ContratDevise'** : on garde la même table telle qu'elle, mais en ajoutant un champ "taux_conversion" nullable, puisqu'on n'a pas encore les informations sur les taux déjà élaborés auparavant.

Après avoir effectué toutes ces modifications ainsi que bien d'autres changements que nous n'avons pas mentionné, et après avoir supprimé tous les champs et restrictions inutiles. Nous pourrons connecter notre application à la base de données pour commencer à l'utiliser.

4.6.1 Diagramme de déploiement

Le déploiement de la solution sera effectué en suivant le diagramme de déploiement de La figure 4.7 suivante :

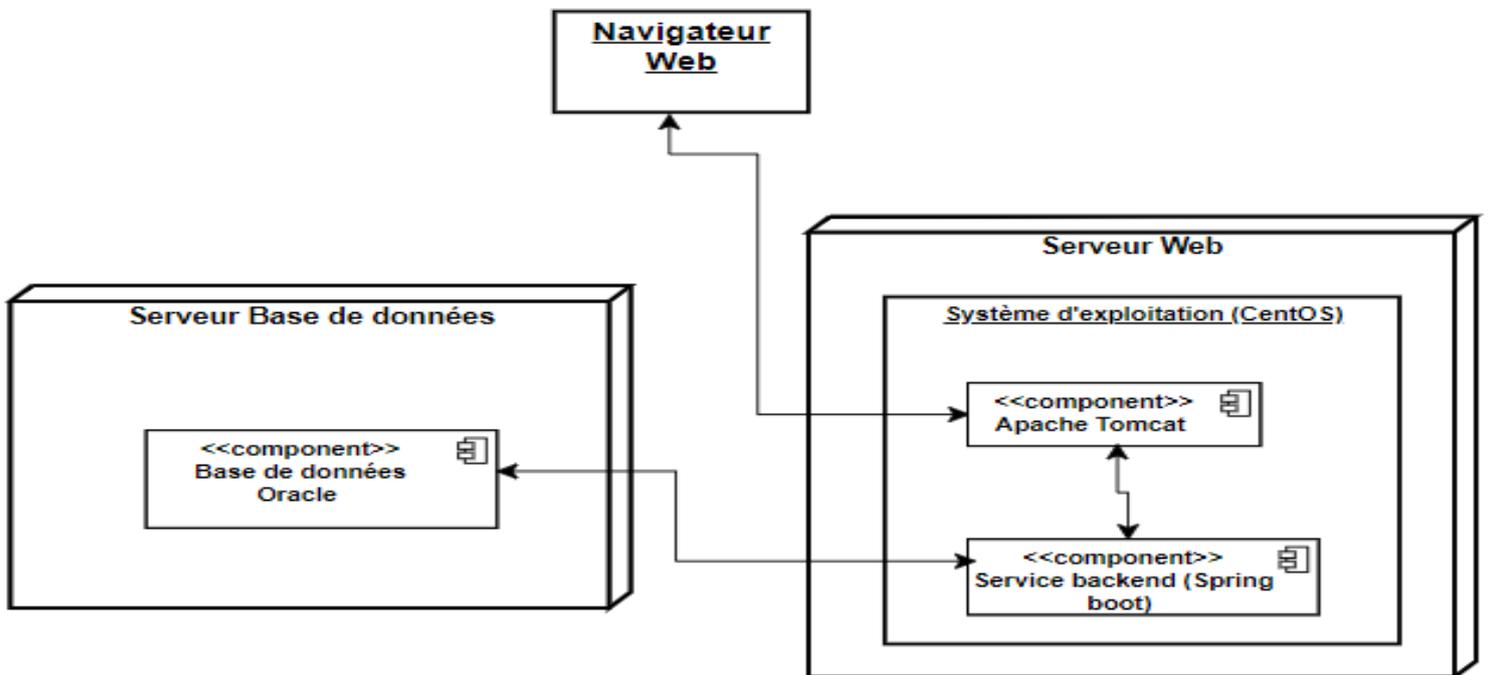


FIG. 4.7 : Diagramme de déploiement.

4.7 Interfaces Graphiques

Dans ce qui suit, nous allons présenter quelques interfaces de notre application. Il convient de souligner que ces interfaces ne sont qu'un aperçu parmi plus de 40 interfaces réalisées.

- Lors du lancement de l'application, la première page qui s'affiche est celle de connexion, contenant un espace réservé à l'identification de l'utilisateur.



FIG. 4.8 : Login page.

- Une fois l'utilisateur authentifié, une page d'accueil s'affiche.

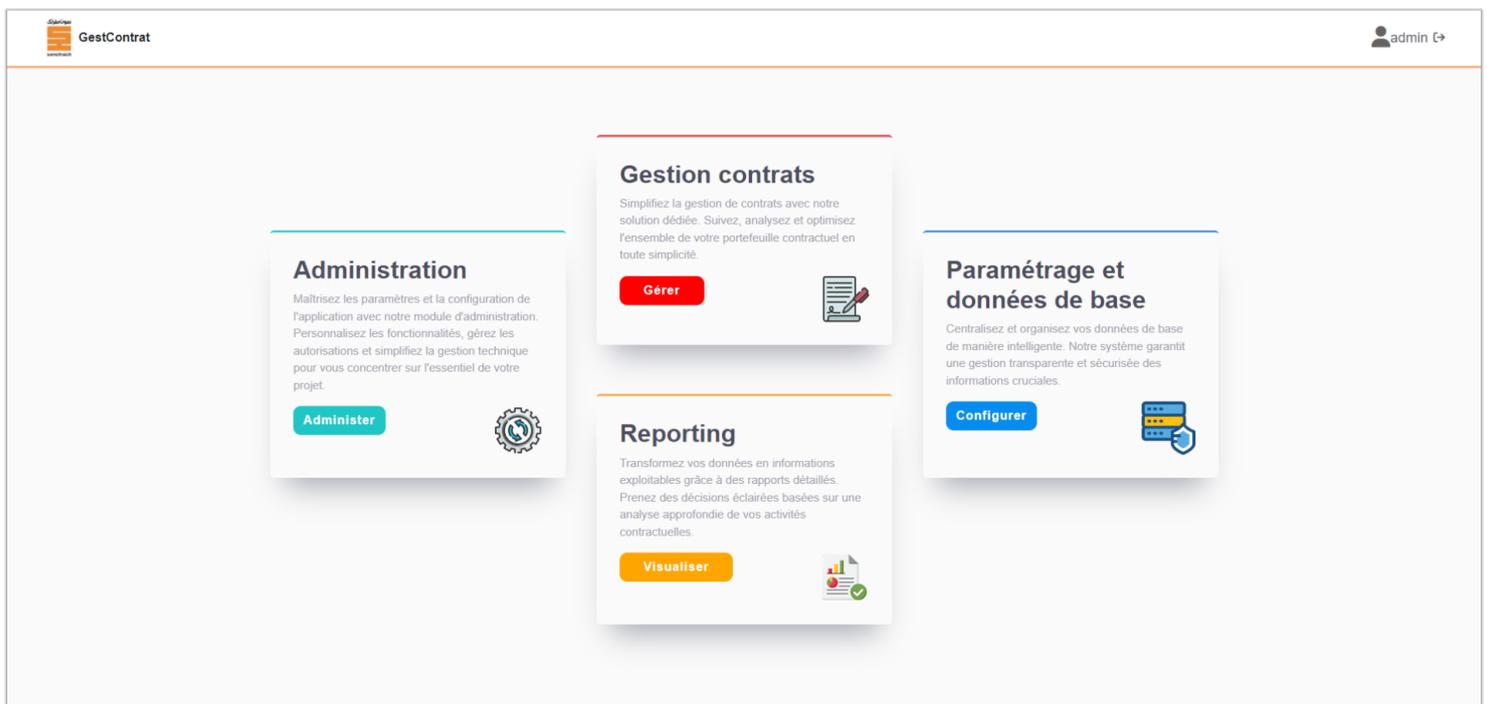


FIG. 4.9 : Accueil.

4.7.1 Administration

Les deux interfaces suivantes, parmi d'autres disponibles, sont exclusivement accessibles aux utilisateurs possédant le rôle d'administrateur :

- Cette interface permet l'ajout d'un utilisateur par l'administrateur.

The screenshot shows the 'Page Admin' interface. At the top left is the 'GestContrats' logo. Below it is a sidebar with 'Table d'administration' and 'Administration' selected. The main content area is titled 'Page Admin' and features a search bar labeled 'Rechercher' and a button 'Ajouter un utilisateur'. Below the search bar is a table with the following data:

Nom d'utilisateur	Email	Structure	Rôle	Historique de connexion
admin	riadto2000@gmail.com	DSI	ADMIN	[Icon]
riad	snestest@gmail.com	DSI	CORRESPONDANT	[Icon]
rhc	aladinsadj@gmail.com	DSI	RHC	[Icon]
anis	hadjkarim200@gmail.com	PL	CORRESPONDANT	[Icon]
chef	hadjriad452@gmail.com	DSI	DECIDEUR	[Icon]

At the bottom of the table are navigation buttons 'Précédent' and 'Suivant'.

FIG. 4.10 : Ajout utilisateur.

- Cette interface représente le journal des activités.

The screenshot shows the 'Journal des Activités' interface. At the top left is the 'GestContrats' logo. Below it is a sidebar with 'Table d'administration' and 'Administration' selected. The main content area is titled 'Journal des Activités' and features a search bar labeled 'Rechercher'. Below the search bar is a table with the following data:

Référence	Type	Utilisateur	Date	Opération
CPS-2023-070	CONTRAT	riad	14/06/2024 22:43	CREATE
CPS-2023-059	CONTRAT	riad	14/06/2024 22:46	UPDATE
CPS-2023-059	CONTRAT	riad	14/06/2024 22:41	CREATE
CPS-2024-001	CONTRAT	riad	14/06/2024 22:46	CREATE
CPS-2023-070	CONTRAT	riad	14/06/2024 22:47	UPDATE

At the bottom of the table are navigation buttons 'Précédent' and 'Suivant'.

FIG. 4.11 : journal des activités.

4.7.2 Paramétrage et référentiel

Les deux interfaces suivantes, parmi d'autres disponibles, sont exclusivement accessibles aux utilisateurs possédant le rôle d'administrateur :

- Cette interface permet l'ajout d'un groupement par administrateur

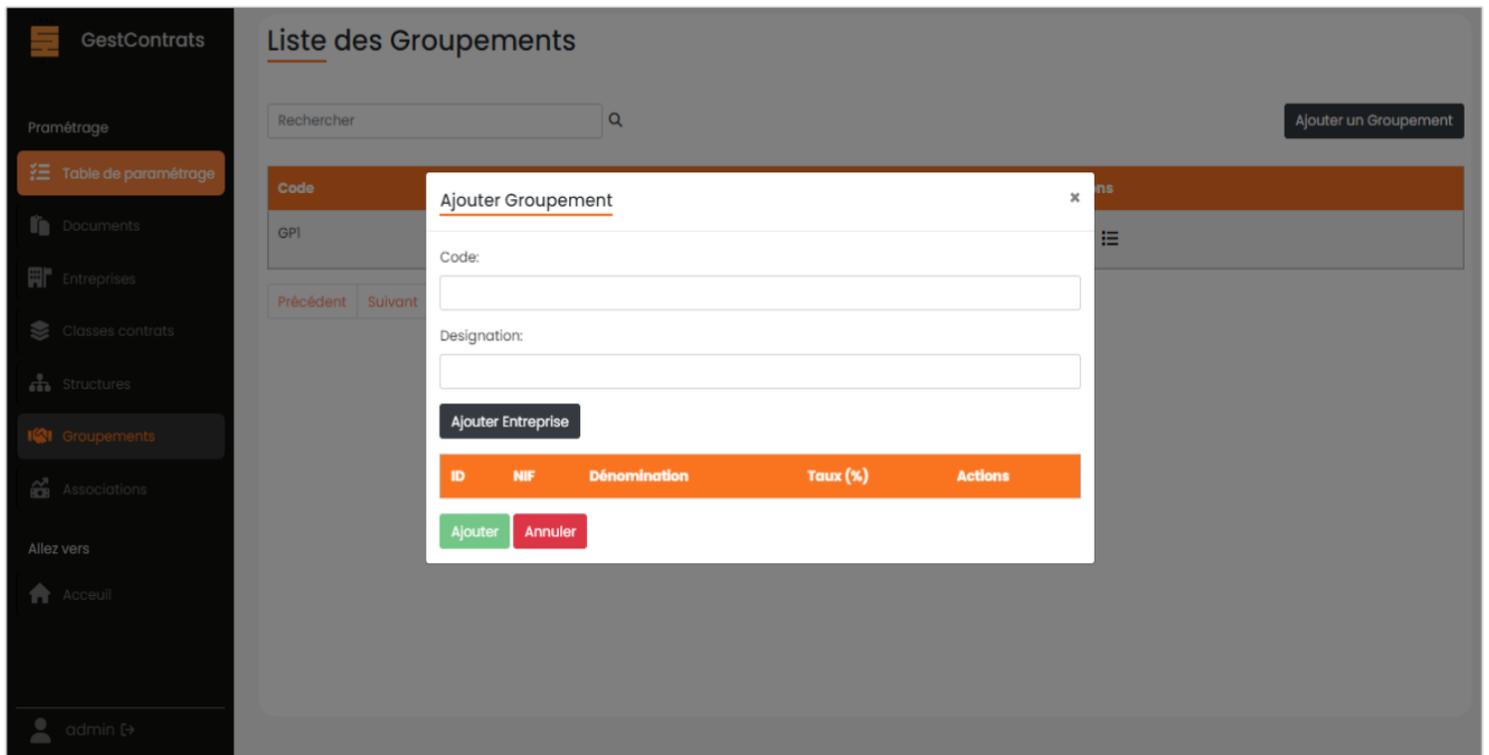


FIG. 4.12 : Ajout groupement.

- Cette interface représente les entreprises incluse dans un groupement

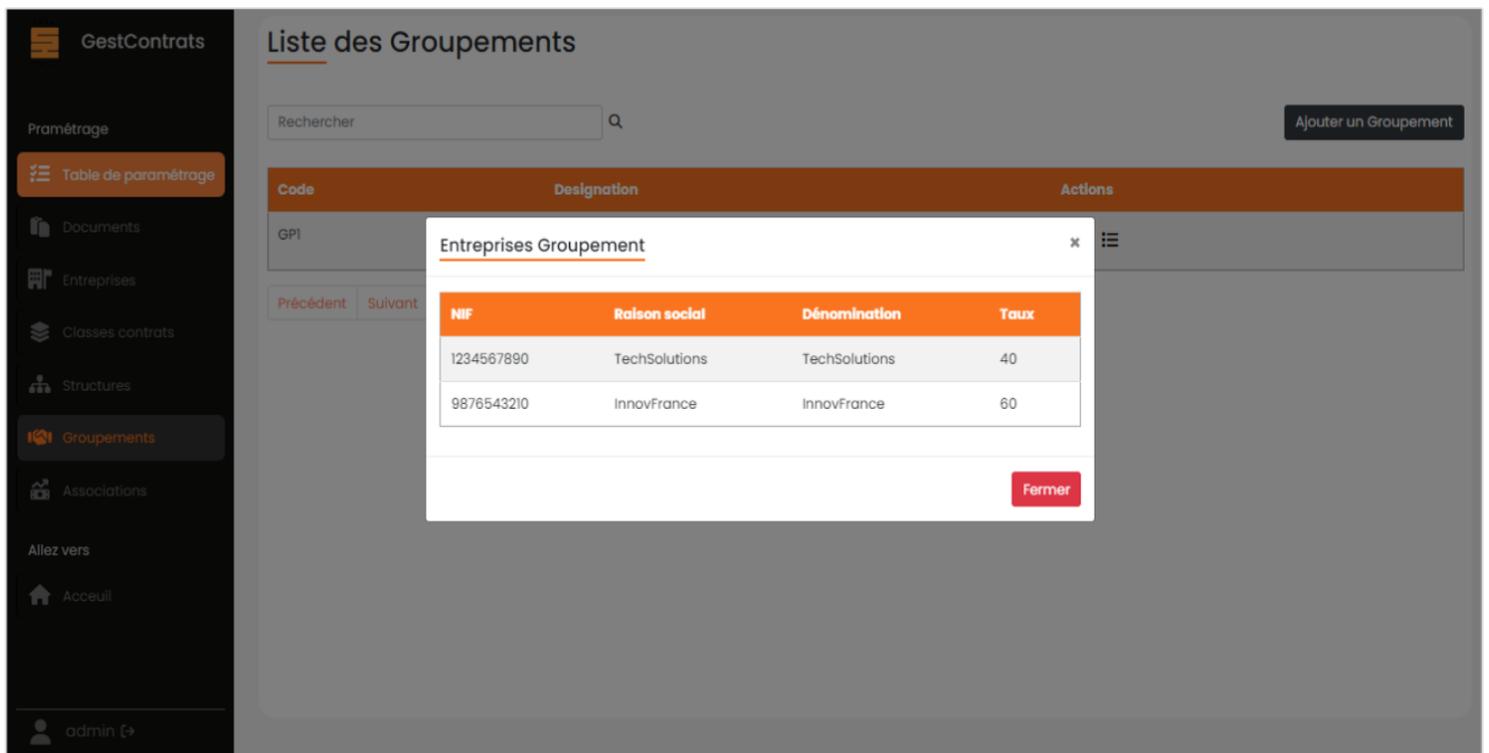
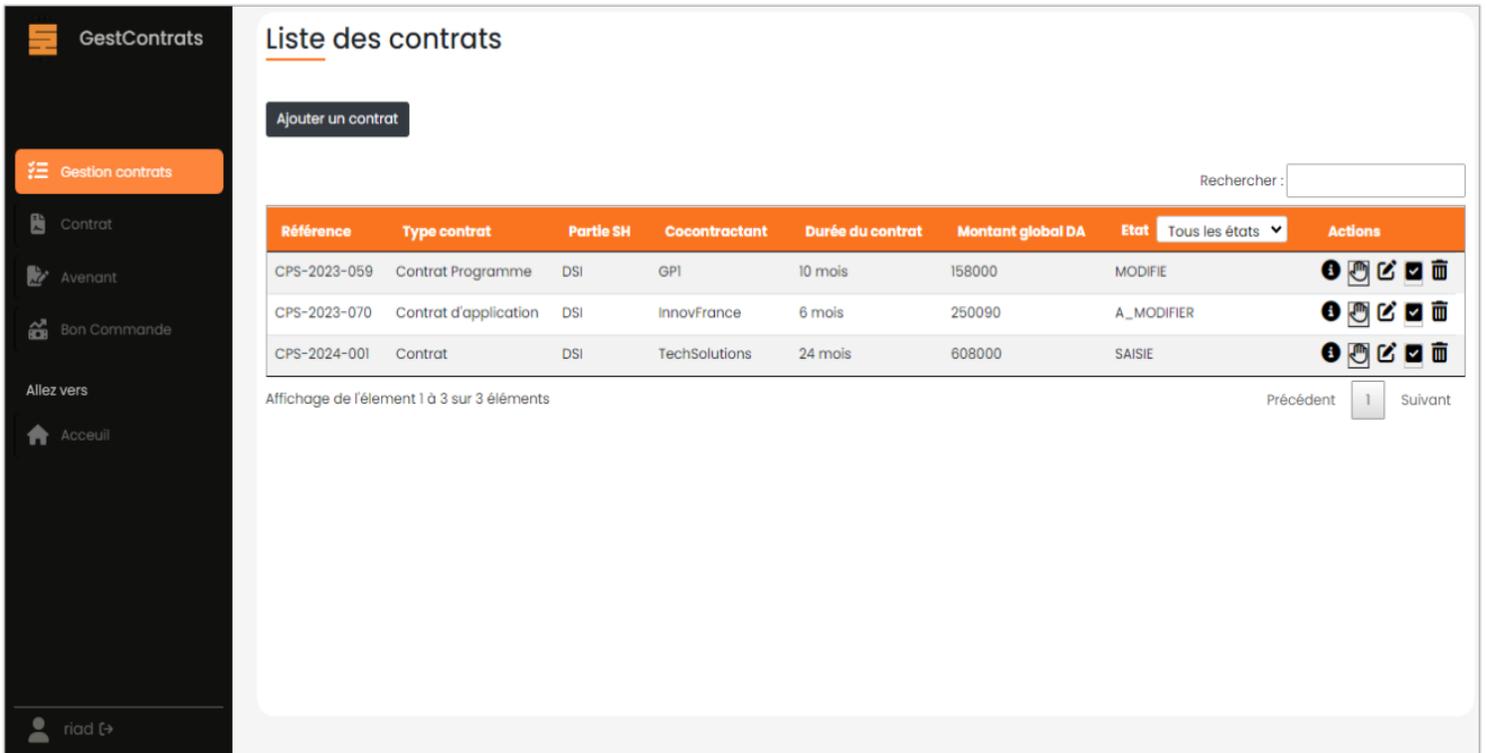


FIG. 4.13 : Entreprise participante dans un groupement.

4.7.3 Gestion Contrats

Les deux interfaces suivantes sont uniquement accessibles aux utilisateurs possédant le rôle correspondant, nous avons pris le cas des contrats, les mêmes interfaces sont disponibles pour les avenants et les bons commande :

- Cette interface montre la liste des contrats



GestContrats

Liste des contrats

Ajouter un contrat

Rechercher:

Référence	Type contrat	Partie SH	Cocontractant	Durée du contrat	Montant global DA	Etat	Tous les états	Actions
CPS-2023-059	Contrat Programme	DSI	GPI	10 mois	158000	MODIFIE		    
CPS-2023-070	Contrat d'application	DSI	InnovFrance	6 mois	250090	A_MODIFIER		    
CPS-2024-001	Contrat	DSI	TechSolutions	24 mois	608000	SAISIE		    

Affichage de l'élément 1 à 3 sur 3 éléments

Précédent Suivant

Allez vers

Accueil

riad

FIG. 4.14 : Liste des contrats.

- Cette interface représente le formulaire d'ajout d'un contrat

Nouvelle contrat/convention

Référence * Type de contrat Indépendant Issue d'un autre contrat

Type de contrat * Participation SH En effort propre En association

Partie SH Cocontractant *

Objet

Délais d'exécution * Durée contrat

Date de signature Date d'entrée en vigueur *

Date d'ordre de service Commission *

Nature Budget * Montant de budget *

Montant globale DA * Montant en dinar uniquement Montant en devise

Type de marché * Mode de passation *

Site d'exécution Allotissement

Droit applicable *

Ajouter Signataires +

Grade	Désignation	Nom	Prénom	Action
Sélectionner un fichier de contrat : <input type="text" value="Choisir un fichier"/> Aucun fichier choisi				

Ajouter Type de Document +

Type documents	Référence	Date	Fichier	Action
Ajouter Annuler				

FIG. 4.15 : Formulaire ajout d'un contrat.

- la figure présente les icônes responsables de la validation et de la demande de modification des contrats.

Référence	Type contrat	Partie SH	Cocontractant	Durée du contrat	Montant global DA	Etat	Tous les états	Actions
CPS-2023-059	Contrat Programme	DSI	GPI	10 mois	158000	MODIFIE		
CPS-2023-070	Contrat d'application	DSI	InnovFrance	6 mois	250090	A_MODIFIER		
CPS-2024-001	Contrat	DSI	TechSolutions	24 mois	608000	SAISIE		

Affichage de l'élément 1 à 3 sur 3 éléments

Demande de modification

Validation création/modification

FIG. 4.16 : Validation et demande de modification.

- Une fois que le contrat est saisi par le correspondant, une notification automatique par e-mail est envoyée au responsable hiérarchique pour validation.



FIG. 4.17 : Email création contrat.

- Dès qu'une demande de modification de contrat est émise par le correspondant, une notification automatique par e-mail est reçue par le responsable hiérarchique.



FIG. 4.18 : Email demande modification contrat.

- Et lorsque le correspondant effectue une modification sur un contrat, une notification automatique par e-mail est envoyée au responsable hiérarchique pour validation.



FIG. 4.19 : Email modification contrat.

4.7.4 Reporting

Les trois interfaces suivantes sont uniquement accessibles aux utilisateurs possédant le rôle de décideur. Il y a cinq tableaux de bord. L'exemple pris est celui des tableaux de bord des contrats.

Chapitre 4. Implementation

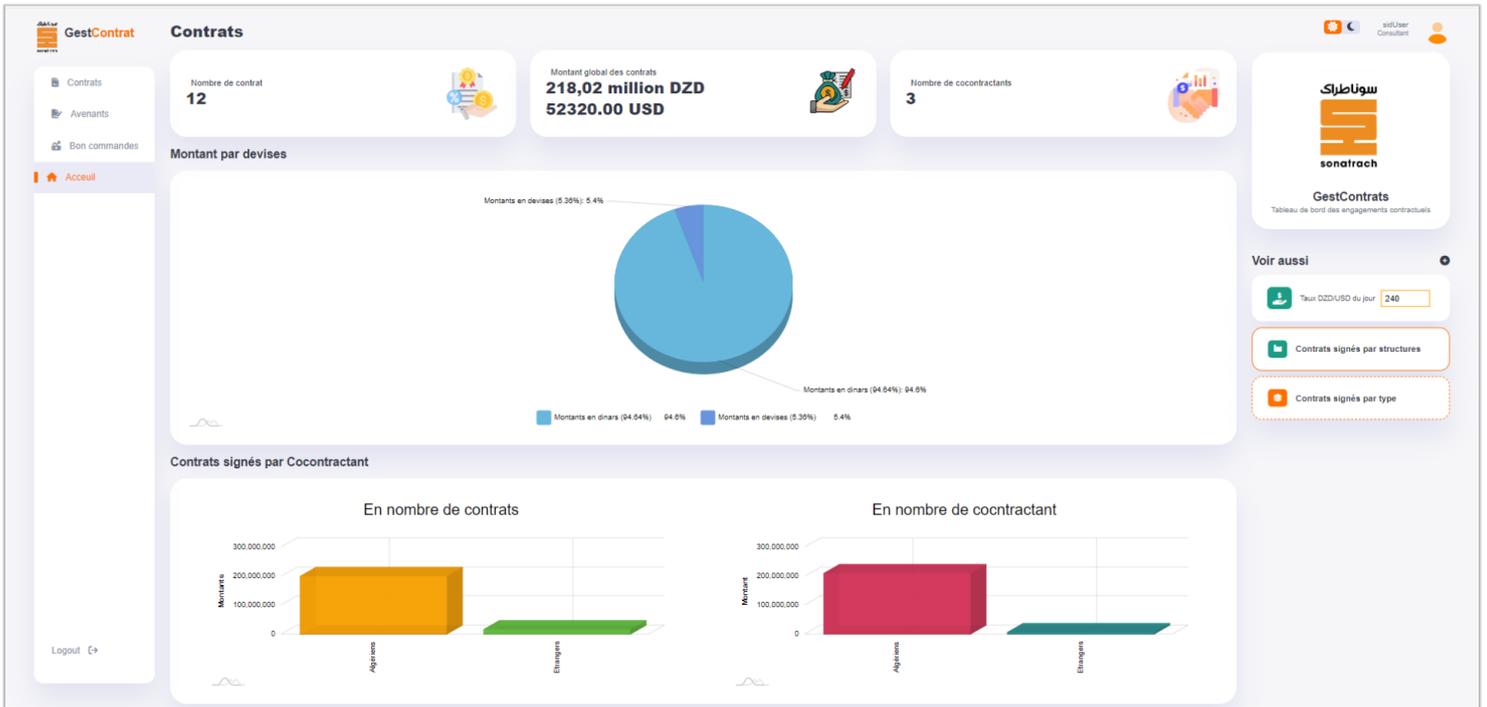


FIG. 4.20 : ContratBI 1.

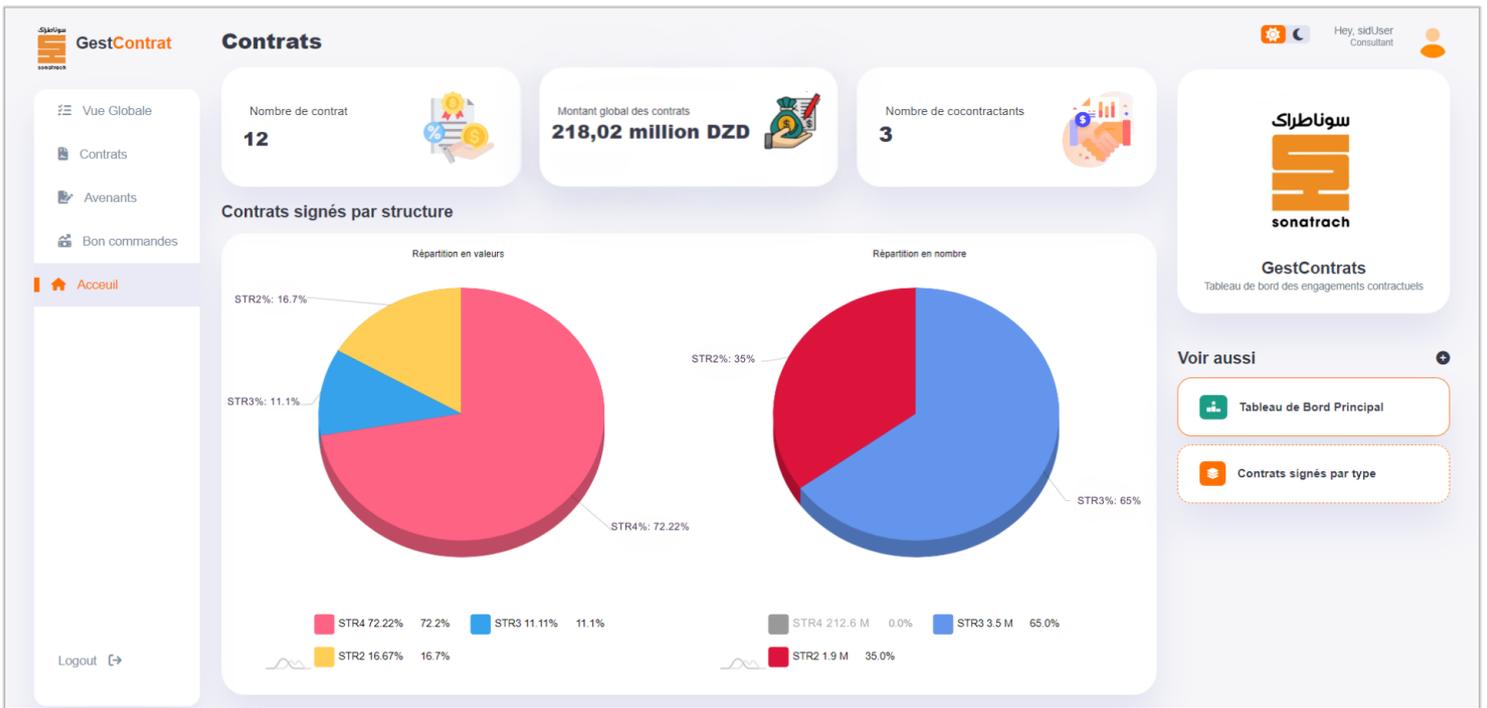


FIG. 4.21 : ContratBI 2.

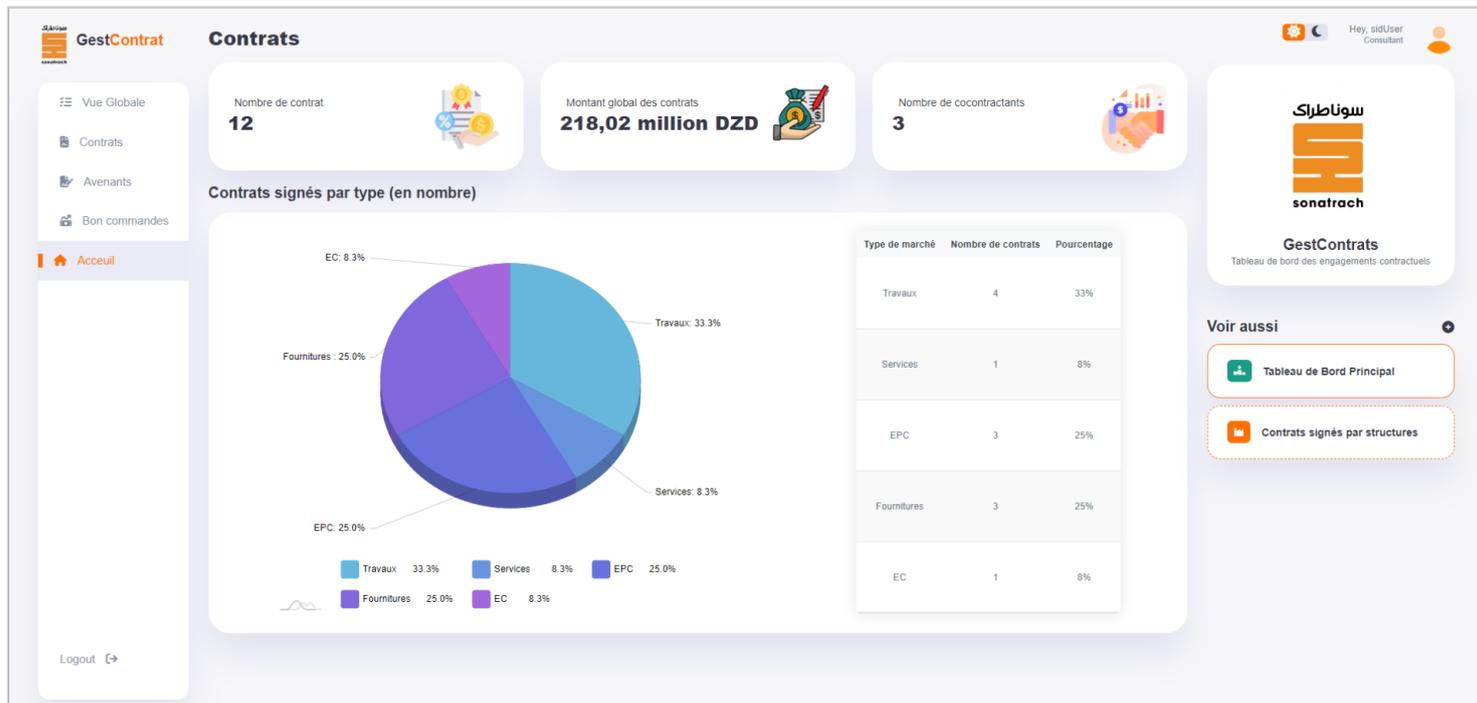


FIG. 4.22 : ContratBI 3.

4.8 Conclusion

Dans ce chapitre, nous avons présenté les outils utilisés dans le développement de notre application et ses principales fonctionnalités, ainsi que certaines interfaces qui illustrent le déroulement de la nouvelle plateforme.

Conclusion et perspectives

Au terme de ce travail, nous pouvons dire que la mise en place de cette nouvelle plateforme représente une grande avancée pour SONATRACH.

Grâce à une étude approfondie du contexte du sujet, nous avons pu mettre en évidence les défis actuels en matière de gestion des contrats. Une analyse du système existant a révélé ses limites, justifiant la nécessité d'évoluer vers une solution plus efficace et adaptée aux nouveaux besoins de l'entreprise.

La mise en œuvre de ce projet a été organisée en plusieurs étapes. Dans un premier temps, nous avons présenté le contexte et les activités de l'entreprise. Nous avons ensuite procédé à une analyse approfondie du système en place, afin d'identifier les axes d'amélioration et les opportunités de développement. Cela nous a permis de commencer à concevoir la nouvelle plateforme en utilisant la modélisation UML pour définir précisément les fonctionnalités de la solution. Enfin, nous avons travaillé sur le développement et la mise en œuvre.

Pour terminer, on peut dire que notre travail a permis de relever le défi de moderniser et d'améliorer les processus de gestion. Cependant, à l'avenir, il serait préférable de penser à d'autres ajouts. Par exemple :

- Intégrer la plateforme à plusieurs autres systèmes d'information de l'entreprise.
- Développer une cellule de réclamations pour le processus de gestion, ce qui permettrait une meilleure maîtrise des risques contractuels.
- Ajouter des fonctionnalités d'intelligence artificielle pour enrichir les fonctionnalités et optimiser les processus de gestion des contrats, avenants et bons de commandes.

Bibliographie

- [1] R. GOUFFI, *Conception et réalisation d'une application web du suivi des stagiaires*, Mémoire de fin de formation, Technicien Supérieur, Spécialité : Développeur d'applications Multiplateformes, Organisme d'accueil : Sonatrach Direction Général, Algérie, sept. 2022.
- [2] *SONATRACH_Organigramme.pdf*. adresse : https://sonatrach.com/wp-content/uploads/2019/09/SONATRACH_Organigramme.pdf.
- [3] *AMENDEMENT DE LA DECISION RELATIVE À L'ORGANISATION DE LA DIRECTION CENTRALE DIGITALISATION ET SYSTEME D'INFORMATION*, Decision No. 332/DG, Classement : 1.021.1, Référence : A-1020 (R1).
- [4] *AMENDEMENT DE LA DECISION RELATIVE À L'ORGANISATION DE LA DIRECTION PROCUREMENT & LOGISTIQUE*, Decision No. 223/DG, Classement : 1.0.17.1, Référence : A-1229(R2).
- [5] *PROCÉDURE DE REPORTING DES CONTRATS*, Decision No. 246/DG, Classement : 0.29.5, Référence : E-036.
- [6] *Qu'est-ce que Java et pourquoi en ai-je besoin ?* Adresse : https://www.java.com/fr/download/help/whatis%5C_java.html.
- [7] *Qu'est-ce que Java ?—Guide du débutant sur Java | Microsoft Azure*. adresse : <https://azure.microsoft.com/fr-ca/resources/cloud-computing-dictionary/what-is-java-programming-language>.
- [8] *Spring Boot*. adresse : <https://spring.io/projects/spring-boot#overview>.
- [9] *Thymeleaf :: Spring Framework*. adresse : <https://docs.spring.io/spring-framework/reference/web/webmvc-view/mvc-thymeleaf.html>.
- [10] *About • Bootstrap*. adresse : <https://getbootstrap.com/docs/4.1/about/overview/>.
- [11] *JavaScript Charts Maps - amCharts*. adresse : <https://www.amcharts.com/>.
- [12] *IntelliJ IDEA overview | IntelliJ IDEA Documentation*. adresse : <https://www.jetbrains.com/help/idea/discover-intellij-6ea.html>.
- [13] *Database | Oracle*. adresse : <https://www.oracle.com/database/>.
- [14] *About Postman*. adresse : <https://www.postman.com/company/about-postman/>.
- [15] *Maven – Introduction*. adresse : <https://maven.apache.org/what-is-maven.html>.
- [16] *Développons en Java - JDBC*. adresse : <https://www.jmdoudoux.fr/java/dej/chap-jdbc.htm>.

- [17] *Java Persistence API (JPA) - Documentation IBM*. adresse : <https://www.ibm.com/docs/fr/was-liberty/nd?topic=liberty-java-persistence-api-jpa>.

Annexes

Annexe A

scripts de migration

Ces scripts SQL représentent les solutions proposées pour résoudre les problèmes rencontrés lors de la migration des données mentionnées dans le chapitre précédent.

- Les requêtes SQL présentées permettent d'ajouter une colonne `id_contrat` dans la table `contrat`, de générer des valeurs auto-incrémentées à l'aide d'une séquence, et de définir cette colonne comme clé primaire. De manière similaire, des identifiants peuvent être ajoutés aux autres tables et de les configurer de la même manière pour garantir l'unicité et la gestion appropriée des données.

```
/* Ajouter une nouvelle colonne id_contrat */
ALTER TABLE contrat
ADD id_contrat NUMBER;

/* Créer une séquence pour générer des valeurs auto-incrémentées */
CREATE SEQUENCE seq_id_contrat
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;

/* Mettre à jour la table contrat pour remplir id_contrat avec les val de séquence */
UPDATE contrat
SET id_contrat = seq_id_contrat.NEXTVAL;

/* Supprimer l'ancienne contrainte de clé primaire sur referenceContrat */
ALTER TABLE contrat
DROP CONSTRAINT pk_referenceContrat;

/* Ajouter une nouvelle contrainte de clé primaire sur id_contrat */
ALTER TABLE contrat
ADD CONSTRAINT pk_contrat
PRIMARY KEY (id_contrat);
```

FIG. A.1 : Création des IDs.

- Ce script permet d'extraire les fichiers PDF de la base de données où ils sont stockés en tant que BLOB, puis de conserver une trace de leur emplacement.

```
/* Ajouter une nouvelle colonne chemin_fichier à La table Contrat */
ALTER TABLE Contrat ADD chemin_fichier VARCHAR2(200);

DECLARE
  /* Déclaration des variables */
  v_Nouveau_fichier BLOB;
  v_Destination VARCHAR2(200); -- Chemin de destination du fichier
  v_Nom_FICHER VARCHAR2(100); -- Nom du fichier
BEGIN
  /* Boucle pour traiter chaque ligne de La table Contrat */
  FOR c IN (SELECT id, CONTRAT_FICHER FROM Contrat) LOOP
    /* Sélectionner Les données de La colonne BLOB */
    v_Nouveau_fichier := c.CONTRAT_FICHER;

    /* Générer un nom de fichier unique */
    v_Nom_FICHER := c.id || '.pdf';

    /* Définir Le chemin de sortie */
    v_Destination := '/chemin/dossier/Contrat/' || v_Nom_FICHER;

    /* Écrire Le contenu du BLOB dans un fichier */
    DBMS_LOB.FILEOPEN(FILENAME => v_Destination, OPTIONS => DBMS_LOB.FILE_CREATE);
    DBMS_LOB.LOADFROMFILE(DEST_LOB => v_Nouveau_fichier, SRC_LOB => v_Destination, AMOUNT
=> DBMS_LOB.GETLENGTH(v_Nouveau_fichier));
    DBMS_LOB.FILECLOSE(FILENAME => v_Destination);

    /* Mettre à jour La table avec Le chemin du fichier */
    UPDATE Contrat
    SET chemin_fichier = v_Destination
    WHERE id = c.id;
  END LOOP;
END;
/

/* Vérifier que tous Les enregistrements ont un chemin valide */
SELECT COUNT(*) FROM Contrat WHERE chemin_fichier IS NULL;

/* Supprimer La colonne BLOB contenant Les fichiers PDF */
ALTER TABLE Contrat DROP COLUMN CONTRAT_FICHER;

/* Ajouter une contrainte NOT NULL sur La colonne chemin_fichier */
ALTER TABLE Contrat MODIFY chemin_fichier VARCHAR2(200) NOT NULL;
```

FIG. A.2 : Extraction des fichier pdf.

- Ce script permet la création du champ "taux_conversion" dans la table "Contrat-Devise".

```
ALTER TABLE contratdevise
ADD taux_conversion NUMBER NULL;
```

FIG. A.3 : Création du taux dans la table « ContratDevise ».

- Ce code SQL modifie la jointure entre les tables contrat et bonCommande, en utilisant le nouveau champ id_contrat. Il peut être adapté pour d'autres paires de tables.

```
/* Créer une table de mappage temporaire pour stocker les associations entre reference et id_contrat */
CREATE TABLE temp_mapping (
  reference VARCHAR2(50),
  id_contrat NUMBER
);

INSERT INTO temp_mapping (reference, id_contrat)
SELECT reference, id_contrat
FROM contrat;

/* Supprimer la contrainte de clé étrangère existante dans la table bonCommande */
ALTER TABLE bonCommande
DROP CONSTRAINT fk_bonCommande_refengagement;

/* Ajouter une nouvelle colonne dans bonCommande pour stocker la nouvelle clé étrangère id_contrat */
ALTER TABLE bonCommande
ADD id_contrat_ref NUMBER;

/* Mettre à jour la nouvelle colonne id_contrat_ref avec les valeurs correspondantes à partir de la table de mappage */
UPDATE bonCommande bc
SET bc.id_contrat_ref = (
  SELECT tm.id_contrat
  FROM temp_mapping tm
  WHERE bc.reference = tm.reference
);

/* Ajouter la nouvelle contrainte de clé étrangère dans bonCommande faisant référence à id_contrat dans contrat */
ALTER TABLE bonCommande
ADD CONSTRAINT fk_bonCommande_idcontrat
FOREIGN KEY (id_contrat_ref) REFERENCES contrat (id_contrat);

/* Supprimer l'ancienne colonne reference de bonCommande */
ALTER TABLE bonCommande
DROP COLUMN reference;

/* Supprimer la table de mappage temporaire temp_mapping */
DROP TABLE temp_mapping;
```

FIG. A.4 : Modification de la Jointure entre Contrat et Bon de Commande.