

École Nationale des Sciences Appliquées de Marrakech
Université Cadi Ayyad
Filière : Génie Informatique

RAPPORT DE STAGE PFA

Développement d'une solution de gestion intégrée

**développement d'un écosystème
interconnecté pour la transformation
digitale d'entreprise ABM maroc**

Réalisé par :

AHMAMO Hamza : Etudiant

AOUIR Aissam : Etudiant

HMIDOUCHE Abdessadk : Etudiant

ENCADRÉ PAR :

RACHID NIDALI

Année académique 2024-2025

Remerciements

Avant de présenter le contenu technique de ce rapport, nous souhaitons adresser nos plus sincères remerciements à toutes les personnes et institutions qui ont contribué, directement ou indirectement, à la réussite de ce projet de fin d'études.

Nous tenons en premier lieu à exprimer notre profonde gratitude à la **direction de l'École Nationale des Sciences Appliquées (ENSA)** pour la qualité de la formation dispensée et l'encadrement pédagogique constant tout au long de notre parcours. Cet accompagnement nous a dotés des compétences techniques, méthodologiques et humaines indispensables pour mener à bien ce stage PFA.

Nous remercions chaleureusement l'ensemble du corps professoral pour son engagement et la richesse de ses enseignements. Une mention particulière à **Monsieur Abdelghafour ATLAS** et **Monsieur BEKKARI Aissam** pour leurs conseils précieux, leur disponibilité et leur soutien tout au long de la réalisation de ce projet.

Côté entreprise, nous adressons nos vifs remerciements à toute l'équipe d'**ABM Maroc** pour son accueil chaleureux, sa confiance et sa collaboration active. Nous remercions tout particulièrement **Monsieur RACHID NIDALI**, Directeur Commercial, qui a été notre encadrant principal. Son expertise métier, ses retours pertinents et son implication quotidienne ont été déterminants pour l'orientation et la réussite de ce travail.

Nous exprimons également notre reconnaissance à **Monsieur BOULOUN**, Directeur Général d'ABM Maroc, pour avoir permis la réalisation de ce stage au sein de son entreprise et pour l'intérêt qu'il a porté à notre projet.

Nous n'oublions pas nos collègues de stage et l'ensemble des collaborateurs d'ABM Maroc qui nous ont facilité l'intégration et apporté leur aide au quotidien.

Ce travail est le fruit d'un effort collectif et n'aurait pas pu aboutir sans la contribution de chacun d'entre vous.

Résumé

Ce projet de fin d'études, réalisé au sein de l'entreprise **ABM Maroc** spécialisée dans le blanchissage et la location de linge professionnel, consiste en la conception et le développement d'un **ERP (Enterprise Resource Planning) complet et interconnecté** visant la transformation digitale globale de l'entreprise.

Le système couvre l'ensemble des processus métier, qui sont par nature **répétitifs, temporels et fortement interdépendants** :

- Prospection et acquisition de nouveaux clients - Gestion des commandes pour les clients en location - Gestion des stocks et des livraisons quotidiennes - Gestion du personnel (transversale) - Gestion des réclamations

L'**architecture fonctionnelle** de cet ERP est centrée sur ces flux interconnectés : la prospection alimente les commandes, les commandes initialisent les stocks, les livraisons et échanges quotidiens assurent le suivi en temps réel, tandis que la gestion du personnel et des réclamations interviennent à toutes les étapes.

Parmi les fonctionnalités clés figurent les formulaires dynamiques pour la collecte des besoins, la personnalisation des bons (création de types, upload de templates, association par client pour une adaptation précise), le suivi multi-niveaux des stocks (blanchisserie, lingerie, travail), la détection automatique des écarts avec notifications, et une interface multimédia intuitive pour les réclamations.

L'ERP génère ainsi le travail de manière fluide et automatisée : il digitalise l'ensemble des opérations quotidiennes de l'entreprise, élimine les tâches manuelles répétitives, centralise l'information en un point unique, réduit considérablement les erreurs humaines, renforce la traçabilité complète des articles et des opérations, et offre une vue unifiée et en temps réel de toutes les activités. Cela améliore significativement la réactivité face aux demandes clients, l'efficacité opérationnelle globale, la qualité de service et, in fine, la satisfaction des clients.

Ce projet illustre la faisabilité d'une digitalisation complète et intégrée des processus métier d'une entreprise du secteur des services B2B. Il pose les bases d'une solution moderne, évolutive et parfaitement adaptée aux contraintes spécifiques du blanchissage et de la location de linge, tout en ouvrant la voie à de futures extensions (facturation avancée, analyse prédictive, etc.).

Mots-clés : ERP, B2B, Transformation digitale, Blanchissage et location de linge,

Gestion intégrée, Flux métier interconnectés, Automatisation des opérations, Suivi temps réel, Personnalisation des bons.

Abstract

This end-of-studies project, carried out within the company **ABM Maroc**, specialized in professional linen laundering and rental, consists of the design and development of a **comprehensive and interconnected ERP (Enterprise Resource Planning)** aimed at the overall digital transformation of the company.

The system covers all business processes, which are by nature **repetitive, temporal and highly interdependent** :

- Prospecting and acquisition of new customers - Order management for rental clients
- Stock and daily delivery management - Personnel management (cross-functional) - Complaint management

The **functional architecture** of this ERP is centered on these interconnected flows : prospecting feeds orders, orders initialize stocks, daily deliveries and exchanges ensure real-time monitoring, while personnel management and complaints intervene at all stages.

Key features include dynamic forms for collecting needs, personalization of documents (creation of types, uploading templates, association per client for precise adaptation), multi-level stock tracking (laundry, linen room, in-use), automatic discrepancy detection with notifications, and a multimedia interface for complaints.

The ERP thus generates work in a fluid and automated manner : it fully digitalizes the company's daily operations, eliminates repetitive manual tasks, centralizes information in a single point, significantly reduces human errors, strengthens complete traceability of items and operations, and provides a unified and real-time view of all activities. This significantly improves responsiveness to client requests, overall operational efficiency, service quality and, ultimately, client satisfaction.

This project demonstrates the feasibility of a complete and integrated digitalization of business processes in a B2B service company. It lays the foundations for a modern, scalable solution perfectly adapted to the specific constraints of linen laundering and rental, while opening the way to future extensions (advanced invoicing, predictive analysis, etc.).

Keywords : ERP, Digital transformation, Linen laundering and rental, Integrated management, Interconnected business flows, Operations automation, Real-time monitoring, Personalized documents.

Table des matières

Remerciements	1
Résumé	2
Abstract	4
1 Introduction	8
1.1 Contexte général et présentation de l'entreprise	8
1.1.1 Présentation de l'entreprise ABM Maroc	8
1.1.2 Contexte général du projet	9
1.2 Problématique	9
1.3 Objectifs généraux et spécifiques	10
1.3.1 Objectif général	10
1.3.2 Objectifs spécifiques	11
1.4 Portée et contraintes	12
1.4.1 Portée du projet	12
1.4.2 Contraintes du projet	12
1.5 Organisation du rapport	13
2 Analyse des besoins	15
2.1 Acteurs du système	15
2.2 Exigences fonctionnelles (vue d'ensemble)	17
2.2.1 Exigences non fonctionnelles	19
2.3 Diagramme de cas d'utilisation (Use Case)	20
3 Conception	23
3.1 Architecture globale du système	23
3.1.1 Vue d'ensemble : backend centralisé, applications web/mobile, bases de données	23
3.1.2 Liaisons entre modules / flux	23
3.1.3 Diagramme d'architecture globale	24
3.2 Flux détaillés du système	25

3.2.1	Flux de prospection	25
3.2.2	Flux de commandes (clients location)	27
3.2.3	Flux de gestion des stocks et livraisons	29
3.2.4	Flux de gestion du personnel (transversal)	31
3.2.5	Flux de réclamations	33
3.3	Autres éléments de conception	35
3.3.1	Diagramme de classes métier	35
3.3.2	Diagramme d'activité global synthétique	36
3.3.3	Modélisation des données	37
4	Réalisation	38
4.1	Choix technologiques et outils	38
4.2	Architecture technique détaillée	38
4.2.1	Couches et composants principaux	39
4.2.2	Flux de communication et protocoles	39
4.2.3	Schéma synthétique de l'architecture technique	39
4.3	Méthodologie de travail et encadrement	40
4.4	Outils de développement	40
4.4.1	Environnements de développement intégrés (IDE)	41
4.4.2	Conteneurisation et déploiement	42
4.4.3	Outils de modélisation et de conception	43
4.4.4	Autres outils et bibliothèques	43
4.4.5	Synthèse des outils	43
4.5	Difficultés rencontrées et solutions	44
4.5.1	Flexibilité et variété des données métier	44
4.5.2	Solutions apportées	46
4.5.3	Autres difficultés et solutions	47
4.5.4	Bilan	48
5	Tests et Validation	49
5.1	Stratégie de tests	49
5.1.1	Types de tests réalisés	49
5.1.2	Méthodologie de tests	50
5.1.3	Outils utilisés	50
5.2	Résultats et corrections	50
5.2.1	Anomalies détectées	50
5.2.2	Processus de correction	50
5.2.3	Statistiques des tests	51
5.2.4	Améliorations post-tests	51
5.3	Validation avec l'encadrant	51

5.3.1	Validation finale	51
5.3.2	Retours de l'encadrant	51
5.3.3	Perspectives d'amélioration	51
6	Conclusion et Perspectives	52
6.1	Bilan du projet	52
6.2	Limites	53
6.3	Perspectives d'amélioration	53
	Bibliographie	55
A	Captures d'écran des interfaces	56
A.1	Dashboard Administrateur	56
A.2	Page de gestion de prospection	57
A.3	Page de gestion des stocks par bons	57
A.4	Page de création de commande	57
A.5	Interface Frontend Mobile	59
A.6	Interface de chat d'agent avec directeur	60
A.7	Interface de chat pour soumettre des recalmmations	61
A.8	Gestion de templates client pour les bons	62
B	Exemples de code source	63
B.1	Service Angular	63
B.2	Controller Spring Boot	63
B.3	Composant Dashboard Angular	63
C	Planning du projet	64
C.1	Phases du projet	64
C.2	Diagramme Gantt (simulé avec PlantUML)	65

Chapitre 1

Introduction

1.1 Contexte général et présentation de l'entreprise

1.1.1 Présentation de l'entreprise ABM Maroc

ABM Maroc est une entreprise spécialisée dans le **blanchissage industriel et la location de linge professionnel**. Implantée dans le **parc industriel de Benguerir***, commune de Sidi Bou Othmane (province de Rehamna, région Marrakech-Safi), elle fournit ses services à une clientèle diversifiée composée principalement d'établissements hôteliers, de restaurants, d'hôpitaux, de cliniques et d'autres structures nécessitant un approvisionnement régulier et fiable en linge de qualité (draps, serviettes, tenues de travail, nappes, uniformes, etc.).

L'activité de l'entreprise repose sur deux modèles complémentaires : - Le **blanchissage simple** : le client reste propriétaire de son linge et confie à ABM Maroc son nettoyage, son entretien, son repassage et sa livraison selon un calendrier défini. - La **location avec blanchissage** : ABM Maroc fournit le linge (elle en est propriétaire), assure son blanchissage, sa livraison, sa collecte du linge sale, son remplacement en cas d'usure et son entretien complet, offrant ainsi une solution clé en main au client.

Ces services exigent une gestion rigoureuse, précise et hautement répétitive : approvisionnement en linge neuf auprès de fournisseurs nationaux et internationaux, suivi minutieux des stocks et des variantes d'articles (dimensions, grammage, matière), organisation des tournées de livraison et de collecte, contrôle qualité à chaque étape, traçabilité complète des articles et gestion rapide des réclamations clients. L'entreprise emploie une équipe pluridisciplinaire composée d'agents de production en blanchisserie, d'agents terrain (présents chez les clients pour les échanges quotidiens), de chauffeurs pour les transports, et de personnel administratif chargé de la coordination et du suivi commercial.

Grâce à son implantation stratégique dans le parc industriel de Benguerir, ABM Maroc bénéficie d'infrastructures modernes et d'une proximité avec les grands axes de communication, ce qui lui permet de desservir efficacement une large zone géographique autour

de Marrakech et au-delà.

1.1.2 Contexte général du projet

Dans un secteur hautement concurrentiel et soumis à des exigences élevées en termes de qualité, d'hygiène, de réactivité et de traçabilité, ABM Maroc faisait face à une gestion majoritairement manuelle de ses processus opérationnels. Les tâches quotidiennes – prospection commerciale, traitement des commandes fournisseurs, suivi des stocks multi-niveaux, planning des livraisons et collectes, gestion des écarts et des réclamations – étaient gérées via des outils disparates (tableurs Excel, documents papier, échanges par e-mail et téléphone), entraînant des risques d'erreurs humaines, des pertes de temps importantes, une visibilité limitée sur l'activité globale et des difficultés à anticiper les besoins.

La direction de l'entreprise a identifié la nécessité d'une **transformation digitale profonde et structurée** pour : - Centraliser l'information et offrir une vue unifiée et en temps réel des opérations. - Automatiser les tâches répétitives et chronophages. - Améliorer la traçabilité complète du linge et réduire significativement les écarts. - Renforcer la réactivité face aux demandes et aux imprévus clients. - Optimiser la gestion des ressources humaines (planning, affectations) et des fournisseurs (historique, performances).

C'est dans ce cadre stratégique que s'inscrit notre projet de fin d'études : le développement d'un **ERP (Enterprise Resource Planning) interconnecté et sur mesure** dédié à la gestion intégrée des activités d'ABM Maroc. Ce système vise à digitaliser l'ensemble du cycle de vie du client et du linge, depuis la prospection et la contractualisation jusqu'à la fidélisation, en passant par les opérations quotidiennes répétitives et temporelles qui constituent le cœur de l'activité.

Le stage, réalisé sous l'encadrement direct de **Monsieur Rachid Nidali**, Directeur Commercial d'ABM Maroc, nous a offert une immersion complète dans les réalités métier de l'entreprise. Ses retours réguliers et son expertise ont permis d'aligner parfaitement la solution proposée sur les besoins opérationnels et stratégiques, tout en garantissant son adéquation avec les contraintes spécifiques du secteur du blanchissage et de la location de linge.

1.2 Problématique

Dans le secteur du blanchissage et de la location de linge professionnel, où les opérations sont hautement répétitives et interconnectées, ABM Maroc fait face à des défis majeurs liés à une gestion encore largement manuelle. Ces contraintes rendent impérative une transformation digitale pour assurer la survie, la compétitivité et la croissance de l'entreprise.

Tout d'abord, il devient **très nécessaire de digitaliser le travail** pour suivre le

rythme de l'évolution technologique et conforter la position concurrentielle sur le marché. Aujourd'hui, pour survivre et avancer, les entreprises doivent adopter la digitalisation : elle permet non seulement de mieux concurrencer les autres acteurs, mais aussi d'acquérir et de maintenir la confiance des clients. Contrairement à avant, où les méthodes traditionnelles suffisaient, le manque de digitalisation peut mener à une perte de clients au profit de concurrents plus modernes, qui offrent des services plus rapides, transparents et personnalisés.

Parmi les problèmes concrets identifiés : - La **difficulté de compter et de suivre tous les chiffres** (quantités d'articles, variantes, écarts de stocks) et de rassembler les bons (commandes, livraisons, collectes) de manière manuelle. Utiliser Excel ou des documents papier pour ces tâches entraîne des erreurs humaines fréquentes, des pertes de temps considérables et une traçabilité limitée. - La **communication déficiente** entre les acteurs (clients, fournisseurs, agents internes), souvent basée sur des e-mails ou des appels téléphoniques, cause des retards et des malentendus. Par exemple, un écart non détecté dans un stock peut mener à une livraison incomplète, provoquant la frustration du client et une fin potentielle de la relation commerciale. - Des problématiques récurrentes avec les fournisseurs (délais de livraison non respectés, quantités incomplètes) et les clients (réclamations non traitées en temps réel), amplifiées par l'absence de visibilité globale. Cela génère des pertes financières (coûts supplémentaires pour corrections d'erreurs) et des risques de rupture de contrats. - D'autres contraintes incluent le manque de scalabilité : avec une croissance de la clientèle, la gestion manuelle devient ingérable, augmentant les risques de saturation des ressources humaines et de pertes d'opportunités commerciales. De plus, la sécurité des données sensibles (contrats, détails clients) est compromise sans outils dédiés, exposant l'entreprise à des risques de fuites ou d'erreurs administratives.

Ces problématiques, extraites des observations et des échanges avec l'encadrement d'ABM Maroc, soulignent l'urgence d'une solution intégrée. Notre projet vise à résoudre ces défis en digitalisant les flux, en automatisant les tâches répétitives et en offrant une plateforme unifiée pour une meilleure efficacité et compétitivité.

1.3 Objectifs généraux et spécifiques

1.3.1 Objectif général

L'objectif général de ce projet de fin d'études est de **concevoir et développer un ERP (Enterprise Resource Planning) interconnecté et sur mesure** permettant la transformation digitale complète des processus métier d'ABM Maroc. Ce système vise à remplacer les méthodes manuelles par une solution intégrée, automatisée et centralisée, capable de gérer l'ensemble du cycle opérationnel de l'entreprise, depuis l'acquisition de nouveaux clients jusqu'à la fidélisation, en passant par les opérations quotidiennes répétitives et

temporelles.

En digitalisant les flux métier, cet ERP cherche à améliorer l'efficacité globale, réduire les erreurs, renforcer la traçabilité, optimiser l'utilisation des ressources et offrir une visibilité en temps réel, contribuant ainsi à la compétitivité et à la croissance durable de l'entreprise dans un secteur en pleine évolution.

1.3.2 Objectifs spécifiques

Pour atteindre cet objectif général, le projet s'articule autour des objectifs spécifiques suivants :

1. **Digitaliser la prospection et l'acquisition de clients** : mettre en place des formulaires dynamiques adaptés au type de client, automatiser la collecte des besoins (notamment via upload de fichiers CSV pour les clients location) et faciliter la validation rapide des contrats, afin de réduire les délais de transformation des prospects en clients fidèles.
2. **Automatiser la gestion des commandes** (pour les clients location) : analyser automatiquement les besoins, créer des commandes et sous-commandes par variante d'article, sélectionner les fournisseurs de manière optimale et générer des bons de commande personnalisés, pour minimiser les interventions manuelles et accélérer l'approvisionnement en linge neuf.
3. **Optimiser la gestion des stocks et des livraisons quotidiennes** : implémenter un suivi multi-niveaux des stocks (blanchisserie, lingerie, travail), automatiser les échanges de linge propre/sale, détecter les écarts en temps réel avec notifications, et personnaliser les bons (livraison, collecte) via des templates adaptés à chaque client et type d'opération, afin d'assurer une traçabilité complète et une réduction significative des pertes.
4. **Centraliser la gestion du personnel** : faciliter la création de contrats, les affectations aux postes et clients, le planning hebdomadaire et la gestion des absences, pour une meilleure allocation des ressources humaines et un soutien efficace à tous les flux opérationnels.
5. **Améliorer le traitement des réclamations** : proposer une interface multimédia intuitive pour l'envoi et la réponse aux réclamations, garantissant une résolution rapide et documentée, renforçant ainsi la satisfaction et la fidélisation des clients.
6. **Assurer une architecture fonctionnelle interconnectée et évolutive** : développer un système où les flux s'alimentent mutuellement (ex. : prospection → commandes → stocks), avec une vue unifiée pour l'administrateur, tout en posant les bases pour de futures extensions (facturation avancée, analyses prédictives, intégrations externes).

7. Garantir la sécurité, la performance et l'ergonomie : implémenter des contrôles d'accès précis, une interface multiplateforme (web et mobile) fluide et une automatisation fiable, pour une adoption facile par tous les acteurs (admin, agents, valle, clients).

Ces objectifs spécifiques, alignés sur les besoins exprimés par ABM Maroc, permettent de répondre directement aux problématiques identifiées (gestion manuelle, manque de traçabilité, communication déficiente) tout en offrant une solution scalable et adaptée aux contraintes du secteur du blanchissage et de la location de linge.

1.4 Portée et contraintes

1.4.1 Portée du projet

Le projet se concentre sur la **conception et le développement d'un ERP interconnecté** couvrant les processus métier principaux d'ABM Maroc. La portée fonctionnelle inclut les flux suivants :

- Prospection et transformation des prospects en clients (formulaires dynamiques, gestion des besoins via CSV). - Gestion des commandes pour les clients en location (analyse des besoins, sous-commandes, sélection fournisseurs, génération de bons dynamiques). - Gestion complète des stocks et des livraisons quotidiennes (suivi multi-niveaux des trois stocks, échanges propre/sale, détection d'écart, notifications automatiques, personnalisation des bons via templates). - Gestion du personnel (contrats, affectations, planning, absences). - Gestion des réclamations multimédia.

Le système est déployé sous forme de : - Un **backend centralisé** gérant l'ensemble des données et logiques métier. - Deux **applications web** (administrateur et prospect). - Deux **applications mobiles** (agent terrain et valle).

La portée technique inclut l'utilisation de technologies modernes (Spring Boot, Angular, Flutter, PostgreSQL, MongoDB) pour assurer une solution multiplateforme, sécurisée et évolutive.

Le projet a été réalisé dans le cadre d'un stage de fin d'études de trois mois, avec une validation progressive auprès de l'encadrant entreprise.

1.4.2 Contraintes du projet

Le développement a été soumis à plusieurs contraintes :

- **Contraintes temporelles** : durée limitée du stage (trois mois), imposant une méthodologie agile avec des itérations courtes et une priorisation rigoureuse des fonctionnalités essentielles.

- **Contraintes humaines** : équipe de trois étudiants, avec une répartition des tâches (backend, frontend web, frontend mobile) tout en maintenant une cohérence globale.
- **Contraintes techniques** : nécessité d'assurer la compatibilité multiplateforme (web et mobile Android/iOS), la gestion de données hybrides (relationnelles et non structurées), et une sécurité robuste pour des données sensibles (clients, contrats).
- **Contraintes métier** : respect strict des spécificités du secteur (traçabilité du linge, gestion des variantes d'articles, personnalisation des bons par client), avec une validation continue par l'encadrant pour garantir l'adéquation aux besoins réels.
- **Contraintes budgétaires et matérielles** : absence de budget dédié (projet académique), utilisation d'outils open-source et gratuits, tests réalisés principalement via émulateurs et environnements locaux.
- **Contraintes liées à l'accès aux données** : respect de la confidentialité des informations réelles de l'entreprise, utilisation de données de test pour les démonstrations.

Malgré ces contraintes, le projet a été mené à bien en respectant les délais et en livrant une solution fonctionnelle et opérationnelle, prête à être déployée et étendue en production.

1.5 Organisation du rapport

Le présent rapport est structuré de manière logique et progressive afin de présenter clairement le déroulement du projet, depuis l'analyse des besoins jusqu'à la conclusion et aux perspectives.

Le rapport est organisé comme suit :

- **Chapitre 1 – Introduction** : présente le contexte général, l'entreprise ABM Maroc, la problématique, les objectifs généraux et spécifiques, la portée et les contraintes du projet, ainsi que l'organisation du rapport lui-même.
- **Chapitre 2 – Analyse des besoins** : identifie les acteurs du système, détaille les exigences fonctionnelles et non fonctionnelles, et présente le diagramme de cas d'utilisation.
- **Chapitre 3 – Conception** : expose l'architecture globale du système, puis décrit en détail chaque flux métier (prospection, commandes, stocks et livraisons, personnel, réclamations), accompagnés des diagrammes d'activité correspondants. Cette partie inclut également les autres éléments de conception (diagramme de classes, modélisation des données).
- **Chapitre 4 – Réalisation** : détaille les choix technologiques, l'architecture technique mise en œuvre, la méthodologie de travail adoptée (Agile), les outils de

développement utilisés, ainsi que les difficultés rencontrées et les solutions apportées.

- **Chapitre 5 – Tests et Validation** : décrit la stratégie de tests adoptée, les résultats obtenus, les corrections effectuées et la validation finale avec l'encadrant.
- **Chapitre 6 – Conclusion et Perspectives** : dresse un bilan global du projet, met en lumière les limites observées et propose des perspectives d'amélioration et d'évolution future du système.
- **Bibliographie** : regroupe l'ensemble des références consultées.
- **Annexes** : contiennent les captures d'écran des interfaces, des exemples de code source, des diagrammes supplémentaires et le planning détaillé du projet.

Cette organisation permet au lecteur de suivre pas à pas le cheminement du projet, de la compréhension des besoins à la mise en œuvre concrète de la solution, tout en facilitant la consultation ciblée des différentes parties.

Chapitre 2

Analyse des besoins

2.1 Acteurs du système

Le système ERP développé pour l'entreprise ABM Maroc est destiné à être utilisé par plusieurs acteurs ayant des rôles et des responsabilités bien définis. Chaque acteur interagit avec le système selon ses missions métier et son niveau d'autorisation. L'identification claire de ces acteurs permet de définir les exigences fonctionnelles, les droits d'accès ainsi que les différents flux d'interaction.

Les principaux acteurs du système sont les suivants :

Administrateur

L'administrateur représente le cœur du système et dispose d'un accès complet à l'ensemble des fonctionnalités de l'ERP. Il est responsable notamment de :

- la gestion des prospects et leur transformation en clients ;
- la création et la gestion des formulaires dynamiques de prospection ;
- l'analyse des besoins clients (formulaires, fichiers CSV) ;
- la création des commandes et des sous-commandes fournisseurs ;
- la gestion des fournisseurs et du personnel ;
- la gestion globale des stocks (blanchisserie, lingerie, stock de travail) ;
- la génération des bons (livraison, collecte, réception) ;
- le suivi des écarts et la génération de notifications automatiques ;
- le traitement des réclamations ;
- la génération de rapports et le suivi global du système.

Prospect

Le prospect est un futur client potentiel de l'entreprise ABM Maroc. Il intervient durant la phase de prospection et peut :

- se connecter à un espace dédié à l'aide d'identifiants reçus par e-mail ;
- remplir un formulaire dynamique adapté à son type d'activité ;
- téléverser un fichier CSV contenant ses besoins (dans le cas des clients de location) ;
- valider les informations avant la signature du contrat.

Après validation par l'administrateur, le prospect devient un client officiel du système.

Client

Le client représente une organisation bénéficiant des services d'ABM Maroc. On distingue deux catégories de clients :

- **Client de blanchissage** : propriétaire de son linge, confiant uniquement le nettoyage à ABM Maroc ;
- **Client de location** : ABM Maroc fournit le linge, assure son entretien et sa gestion complète.

Le client peut consulter les livraisons, les collectes, les rapports d'inventaire et soumettre des réclamations via son représentant.

Valle (représentant du client)

La valle est le représentant du client au sein de son établissement. Il joue le rôle d'intermédiaire entre le client et ABM Maroc et peut :

- suivre les livraisons et les collectes ;
- signaler les écarts constatés ;
- envoyer des réclamations (texte, image ou audio) ;
- consulter les informations liées aux opérations quotidiennes.

Agent terrain

L'agent terrain est un employé d'ABM Maroc présent chez le client. Ses principales missions sont :

- le contrôle du stock de lingerie chez le client ;
- la validation des entrées et sorties de linge ;
- la saisie et la validation des bons via l'application mobile ;
- le signalement des anomalies et des écarts ;
- la participation aux inventaires périodiques.

Toutes ses actions sont tracées afin d'assurer la responsabilité et la traçabilité.

Agent de production

L'agent de production intervient au niveau de la blanchisserie. Il est chargé de :

- la réception du linge sale ;

- la validation des quantités traitées ;
- la préparation du linge propre pour la livraison ;
- la génération des bons liés aux opérations de lavage.

Chauffeur

Le chauffeur assure le transport logistique du linge entre la blanchisserie et les clients. Il est responsable de :

- la livraison du linge propre ;
- la collecte du linge sale ;
- la validation des opérations logistiques à l'aide des bons générés par le système.

Fournisseur

Le fournisseur est un acteur externe fournissant le linge neuf à ABM Maroc. Il intervient principalement pour :

- la réception des bons de commande envoyés par e-mail ;
- la livraison du linge en une ou plusieurs fois ;
- la fourniture des documents de livraison associés.

2.2 Exigences fonctionnelles (vue d'ensemble)

Les exigences fonctionnelles définissent l'ensemble des services que le système ERP doit offrir afin de répondre aux besoins opérationnels et organisationnels de l'entreprise ABM Maroc. Elles traduisent les processus métier identifiés lors de la phase d'analyse en fonctionnalités concrètes et exploitables par les différents acteurs du système.

Ces exigences couvrent l'intégralité du cycle de gestion du linge, depuis la phase de prospection et de contractualisation des clients jusqu'au suivi des stocks, des livraisons, du personnel et du traitement des réclamations. Afin d'assurer une meilleure lisibilité, les exigences fonctionnelles sont regroupées par modules correspondant aux principaux domaines fonctionnels du système.

Gestion de la prospection et des clients

Le système doit permettre une gestion structurée et centralisée de la relation client, en prenant en charge l'ensemble des étapes de la phase de prospection. À ce titre, il doit assurer :

- la création, la modification et le suivi des comptes prospects par l'administrateur ;
- l'envoi automatique des identifiants d'accès aux prospects par courrier électronique ;

- la conception de formulaires dynamiques adaptés au profil et au secteur d'activité du prospect ;
- la saisie et la soumission des formulaires par les prospects via un espace sécurisé ;
- la validation des données collectées et la transformation d'un prospect en client officiel après signature du contrat ;
- l'archivage ou la suppression des prospects non validés.

Gestion des commandes

Le système doit assurer une gestion complète et automatisée des commandes, en tenant compte des spécificités des services de blanchissage et de location. Il doit notamment permettre :

- la création et le suivi des commandes clients selon le type de prestation ;
- l'importation, l'analyse et le traitement des fichiers CSV de besoins pour les clients de location ;
- la génération automatique de commandes principales et de sous-commandes par article et par variante ;
- la sélection des fournisseurs en fonction de la disponibilité des articles et des contraintes définies ;
- la génération et l'envoi automatiques des bons de commande aux fournisseurs ;
- le suivi de l'état d'avancement des commandes et des livraisons associées.

Gestion des stocks

Le système doit garantir une gestion rigoureuse et en temps réel des stocks de linge.

Il doit permettre :

- la gestion de plusieurs types de stocks (blanchisserie, lingerie chez le client et stock de travail) ;
- l'initialisation des stocks lors de la première mise en service ou livraison ;
- la mise à jour automatique des quantités suite aux livraisons, collectes et opérations de lavage ;
- le suivi en temps réel des mouvements de linge tout au long de son cycle de vie ;
- la détection automatique des écarts de stock et l'historisation des mouvements.

Gestion des livraisons et des bons

Le système doit assurer une traçabilité complète des opérations logistiques à travers la gestion des bons. Il doit ainsi permettre :

- la génération automatique des bons de livraison, de collecte et de réception ;

- la personnalisation des modèles de bons selon le type de client et le contexte opérationnel ;
- la validation des bons via une application mobile dédiée ;
- l'enregistrement et la traçabilité de toutes les opérations de livraison et de collecte ;
- l'envoi automatique des bons validés aux parties concernées.

Gestion des réclamations

Le système doit offrir un canal de communication structuré pour le traitement des réclamations. À ce titre, il doit permettre :

- la soumission de réclamations par les agents et les représentants des clients ;
- l'intégration de contenus multimédias (texte, image et audio) dans les réclamations ;
- la réception, le suivi et le traitement des réclamations par l'administrateur ;
- la traçabilité des échanges et la clôture formelle des réclamations.

Gestion du personnel

Le système doit prendre en charge la gestion des ressources humaines impliquées dans les opérations. Il doit permettre :

- l'ajout, la modification et la consultation des informations relatives aux employés ;
- la gestion des contrats de travail (CDD, CDI) ;
- l'affectation et la désaffectation des employés aux postes, aux clients et aux missions ;
- la gestion des plannings, des absences et des jours de repos ;
- le suivi des interventions et des activités des agents.

Reporting et supervision

Le système doit offrir des outils d'aide à la décision à travers des fonctionnalités de reporting et de supervision. Il doit permettre :

- la consultation de tableaux de bord synthétiques ;
- la génération de rapports d'inventaire périodiques ;
- le suivi en temps réel de la circulation du linge ;
- l'identification des responsabilités en cas d'écart constaté ;
- la génération de notifications automatiques.

2.2.1 Exigences non fonctionnelles

Les exigences non fonctionnelles définissent les contraintes et les critères de qualité que doit respecter l'application, indépendamment des fonctionnalités métier. Elles garantissent la performance, la sécurité, la fiabilité et la maintenabilité du système.

- **Performance** : Le système doit assurer un temps de réponse rapide pour les opérations courantes telles que l'authentification, la consultation des données et l'exécution des transactions. Les requêtes fréquentes doivent être traitées de manière optimisée afin de garantir une expérience utilisateur fluide.
- **Sécurité** : L'application doit assurer la confidentialité et l'intégrité des données sensibles des utilisateurs. L'accès aux fonctionnalités doit être protégé par des mécanismes d'authentification et d'autorisation. Les communications entre le frontend et le backend doivent être sécurisées.
- **Fiabilité** : Le système doit être stable et capable de fonctionner correctement même en cas de forte charge. Les erreurs doivent être gérées de manière contrôlée afin d'éviter toute perte de données ou interruption critique du service.
- **Disponibilité** : L'application doit être accessible de manière continue, avec un taux de disponibilité élevé. Les opérations critiques doivent rester accessibles même en cas de maintenance partielle du système.
- **Scalabilité** : Le système doit être conçu de façon à pouvoir évoluer facilement, en supportant l'augmentation du nombre d'utilisateurs, de transactions et de données, sans dégradation significative des performances.
- **Maintenabilité** : L'architecture logicielle doit être modulaire et bien structurée afin de faciliter la maintenance, les mises à jour et l'ajout de nouvelles fonctionnalités. Le code doit être lisible, documenté et conforme aux bonnes pratiques de développement.
- **Portabilité** : L'application doit pouvoir être déployée sur différents environnements sans modifications majeures, grâce à l'utilisation de technologies standards et multiplateformes.
- **Ergonomie et utilisabilité** : L'interface utilisateur doit être intuitive, claire et facile à prendre en main, afin de réduire le temps d'apprentissage et améliorer l'expérience utilisateur.

2.3 Diagramme de cas d'utilisation (Use Case)

Le diagramme de cas d'utilisation ci-dessous présente les différents acteurs du système ERP Blanchisserie & Location de Linge ainsi que leurs interactions avec les fonctionnalités principales (use cases). Chaque acteur représente un rôle dans le système et chaque cas d'utilisation correspond à une fonctionnalité offerte par le système.

Ce diagramme permet de visualiser rapidement :

- Les acteurs principaux : Directeur Commercial, Administrateur, Prospect, Client, Agent, Chauffeur, Agent Production, Fournisseur, etc.
- Les fonctionnalités principales du système : gestion de prospection, gestion des commandes, gestion des stocks, échanges de linge, génération de rapports, traite-

- ment des réclamations, gestion du personnel, etc.
- Les relations entre les acteurs et les cas d'utilisation ainsi que les inclusions («`include`») pour montrer les dépendances fonctionnelles.

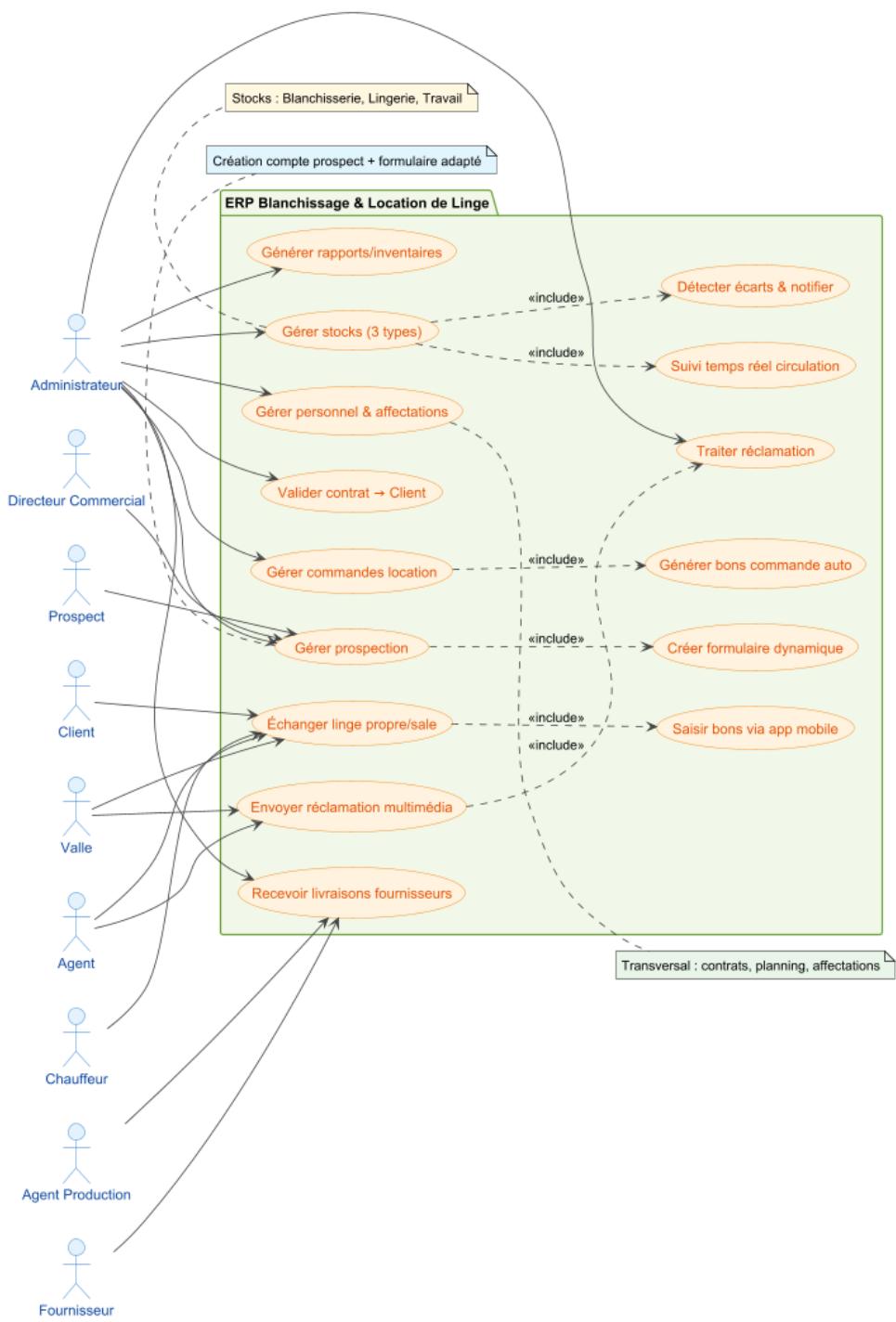


FIGURE 2.1 – Diagramme de cas d'utilisation de l'ERP Blanchisserie & Location de Linge.

Chapitre 3

Conception

3.1 Architecture globale du système

3.1.1 Vue d'ensemble : backend centralisé, applications web/mobile, bases de données

Le système ERP Blanchisserie & Location de Linge est basé sur une **architecture centralisée** composée de trois couches principales :

- **Backend centralisé** : un serveur Spring Boot qui gère la logique métier, les API REST, la sécurité et les traitements de données.
- **Applications web et mobile** : des interfaces front-end (Angular pour le web, applications mobiles pour la saisie et le suivi en temps réel) permettant aux différents acteurs d'interagir avec le système.
- **Bases de données** : stockage des données centralisées (MySQL/PostgreSQL pour les données relationnelles, MongoDB pour certaines collections flexibles) pour gérer les clients, fournisseurs, commandes, stocks et historiques.

Cette architecture permet une gestion cohérente et sécurisée des informations, une synchronisation en temps réel entre les différents modules et une évolutivité pour l'ajout de nouvelles fonctionnalités.

3.1.2 Liaisons entre modules / flux

Le système ERP Blanchisserie & Location de Linge est organisé en plusieurs modules interconnectés. Les flux principaux entre ces modules sont les suivants :

- **Prospection & Contrats** : Le module Prospection collecte les informations des prospects et transmet les données validées au module Contrats pour créer les contrats clients.
- **Commandes & Stocks** : Les commandes passées par les clients sont envoyées au module Stock pour vérifier la disponibilité et suivre la circulation du linge

(propre/sale).

- **Livraisons & Fournisseurs :** Les livraisons reçues des fournisseurs mettent à jour automatiquement les stocks et notifient les modules concernés (Production, Gestion de stocks).
- **Rapports & Analyses :** Les modules Rapports et Inventaires récupèrent les données de tous les autres modules pour générer des synthèses et indicateurs.
- **Réclamations & Suivi :** Les réclamations client sont transmises au module Gestion réclamations, qui coordonne avec les modules Agent, Production et Administration pour résoudre les incidents.

Ces liaisons garantissent un flux cohérent des informations entre tous les modules et assurent une traçabilité complète des opérations.

3.1.3 Diagramme d'architecture globale

Le diagramme suivant illustre l'architecture globale des flux métier de l'ERP Blan- chisserie & Location de Linge, en montrant comment les modules interagissent entre eux et avec le module de gestion du personnel, qui est transversal.

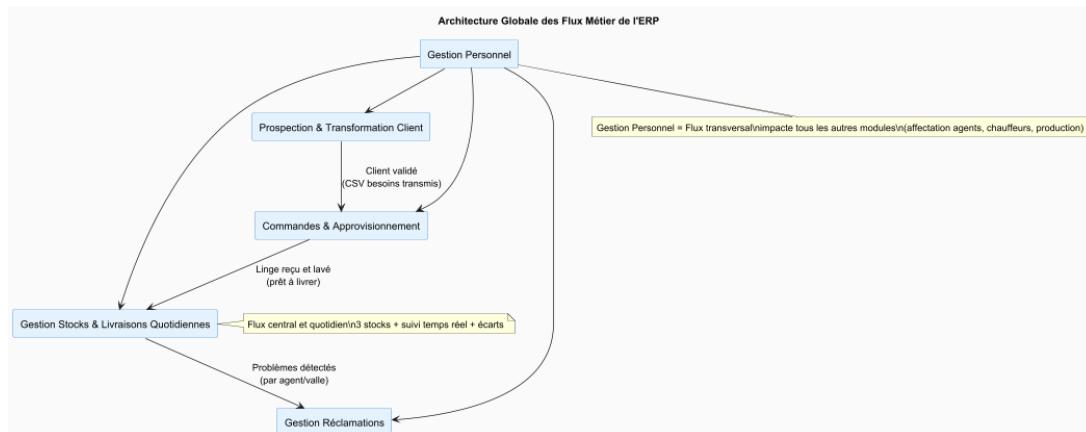


FIGURE 3.1 – Diagramme d'architecture globale des flux métier

Le diagramme montre :

- **Prospection & Transformation Client :** Gère la relation client et transmet les besoins validés au module de commandes.
- **Commandes & Approvisionnement :** Traite les commandes clients et organise l'approvisionnement des stocks.
- **Gestion des Stocks & Livraisons Quotidiennes :** Suit l'état des stocks, les lavages et la livraison du linge.
- **Gestion des Réclamations :** Prend en charge les problèmes signalés par les clients ou les agents.
- **Gestion du Personnel :** Module transversal impactant tous les autres modules pour l'affectation des agents, chauffeurs et personnel de production.

3.2 Flux détaillés du système

3.2.1 Flux de prospection

Le flux de prospection décrit le processus par lequel un prospect est identifié, suivi et transformé en client. Les principales étapes sont :

- Identification et saisie des informations du prospect.
- Création et gestion du compte prospect par le directeur commercial ou l'administrateur.
- Qualification du prospect et suivi des interactions.
- Validation du contrat une fois que le prospect devient client.
- Transmission des informations au module de commandes pour traitement ultérieur.

Ce flux permet de garantir une gestion structurée des prospects et assure la continuité vers les étapes de commande et de livraison.

Description et objectifs

Le flux de prospection a pour objectif de gérer efficacement l'acquisition de nouveaux clients pour l'ERP de blanchisserie et location de linge. Il décrit les étapes depuis l'identification d'un prospect jusqu'à sa transformation en client actif, en assurant un suivi structuré et une traçabilité des interactions.

Les principaux objectifs sont :

- Assurer la création et la gestion centralisée des prospects.
- Optimiser le suivi des interactions commerciales.
- Garantir la qualité et la fiabilité des informations transmises aux modules de commande et de livraison.
- Faciliter la conversion des prospects en clients avec contrats validés.

Acteurs impliqués

Les acteurs impliqués dans le flux de prospection sont :

- **Directeur Commercial** : supervise la prospection et valide les stratégies commerciales.
- **Administrateur** : gère les comptes prospects et configure les formulaires dynamiques.
- **Prospect** : personne ou organisation ciblée par les actions commerciales.

Processus étape par étape

Le flux de prospection se déroule en plusieurs étapes clés :

1. **Identification du prospect** : le Directeur Commercial ou l'équipe commerciale repère de nouveaux clients potentiels.
2. **Création du compte prospect** : l'Administrateur enregistre le prospect dans le système et complète les informations de contact.
3. **Remplissage du formulaire dynamique** : un formulaire adapté est généré pour collecter les besoins spécifiques du prospect.
4. **Suivi des interactions** : toutes les communications et actions commerciales sont enregistrées pour assurer un suivi structuré.
5. **Validation du contrat** : lorsque le prospect accepte les conditions, l'Administrateur transforme le prospect en client actif.
6. **Transmission aux modules suivants** : les informations validées sont envoyées au module de commandes et de livraison pour prise en charge.

Liens avec les autres flux

Le flux de prospection est directement lié aux autres flux du système :

- **Flux de commandes** : les prospects validés deviennent des clients, et leurs besoins sont transmis au module de gestion des commandes pour traitement.
- **Flux de gestion du stock et livraisons** : les informations sur les produits ou services demandés par le prospect alimentent le module de stock afin de préparer la livraison future.
- **Flux de gestion du personnel** : les affectations d'agents commerciaux pour suivre les prospects sont gérées via le module personnel.
- **Flux de réclamations** : si un prospect ou nouveau client signale un problème, cela sera traité via le flux de gestion des réclamations.

Ainsi, le flux de prospection sert de point de départ et de liaison pour les autres modules clés de l'ERP.

Intégration technique

Le flux de prospection est intégré techniquement à l'ERP via les composants suivants :

- **Backend (Spring Boot)** :
 - Gestion des entités Prospect, Client, Contrat.
 - API REST pour créer, lire, mettre à jour et supprimer les prospects.
 - Validation des données avant transformation en client actif.
- **Base de données** :
 - Stockage des prospects, clients et formulaires dynamiques dans des tables centralisées.
 - Relations entre prospects et commandes futures via clés étrangères.

- **Frontend (Web / Mobile) :**
 - Formulaires dynamiques pour la saisie des informations prospect.
 - Tableau de suivi et notifications pour les agents commerciaux.
- **Intégration inter-modules :**
 - Les prospects validés déclenchent des flux vers le module commandes.
 - Les affectations des agents sont gérées par le module personnel.
 - Les données sont synchronisées en temps réel via API REST et messages internes.

Cette intégration garantit que le flux de prospection fonctionne de manière cohérente avec tous les autres flux de l'ERP.

Diagramme d'activité dédié

Le diagramme d'activité ci-dessous illustre les étapes du flux de prospection depuis l'identification du prospect jusqu'à sa transformation en client actif.

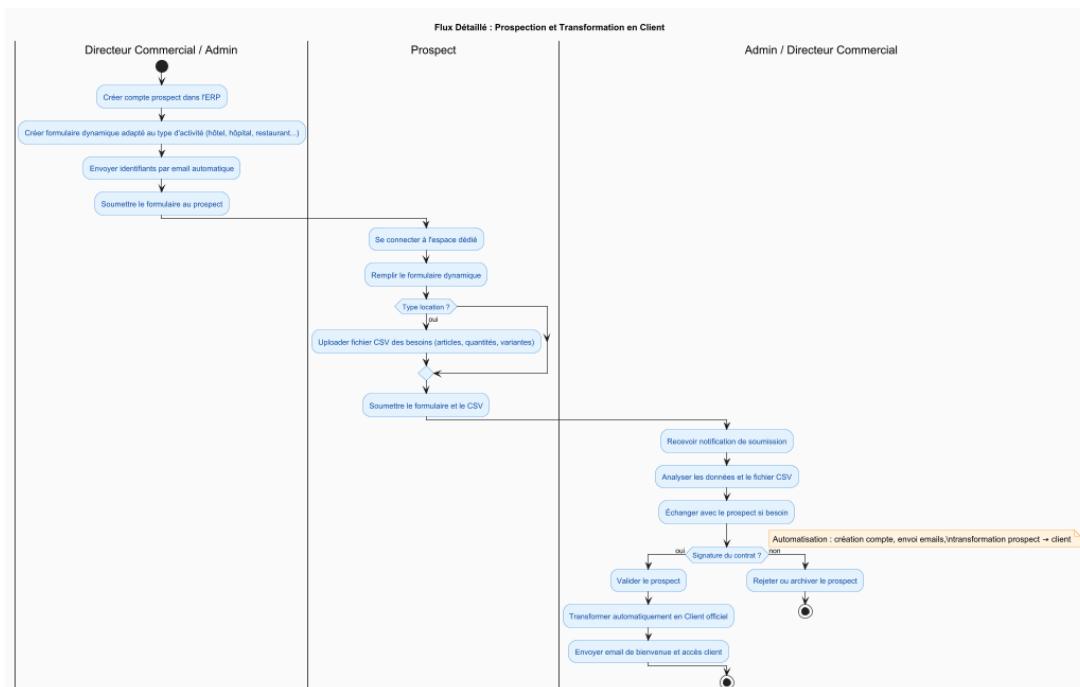


FIGURE 3.2 – Diagramme d'activité du flux de prospection

3.2.2 Flux de commandes (clients location)

Description et objectifs

Le flux de commandes a pour objectif de gérer l'ensemble du processus de commande pour les clients de location. Il couvre la création des commandes, la répartition entre fournisseurs, la génération de bons et la mise à jour des stocks.

Objectifs principaux :

- Assurer un traitement rapide et fiable des commandes clients.
- Optimiser la répartition des articles entre fournisseurs.
- Garantir la traçabilité des commandes et livraisons.
- Automatiser la génération et la personnalisation des bons (PDF) pour chaque type de commande.

Acteurs impliqués

- **Administrateur** : Crée et supervise les commandes.
- **Fournisseur** : Prépare et livre les articles commandés.
- **Agent de production** : Reçoit et enregistre les livraisons, planifie le traitement du linge.

Processus étape par étape

1. Accéder aux besoins du client (CSV prospect ou ERP).
2. Analyser quantités, articles et variantes.
3. Créer la commande principale et sous-commandes si nécessaire.
4. Rechercher et assigner les fournisseurs pour chaque variante.
5. Créer et associer les bons dynamiques aux commandes et clients.
6. Générer automatiquement les PDF des bons et les envoyer aux fournisseurs.
7. Suivre la livraison et la réception dans l'ERP.
8. Mettre à jour les stocks blanchisserie.
9. Planifier le traitement (lavage, livraison au client).

Gestion des bons (types, templates, personnalisation)

- **Types de bons** : Commande fournisseur, bon de livraison, etc.
- **Templates** : Uploadés dans l'ERP et associés à chaque type.
- **Personnalisation** : Champs dynamiques adaptés au client et au type de bon.
- **Traçabilité** : Historique des bons et des livraisons fournisseur.

Liens avec les autres flux

- Le flux de commandes dépend du flux de prospection pour obtenir les besoins clients validés.
- Les livraisons fournisseurs impactent le flux de gestion des stocks.
- Les anomalies ou problèmes de livraison peuvent générer des réclamations (flux réclamations).

Intégration technique

- Backend centralisé (Spring Boot) pour la gestion des commandes et bons.
- Base de données relationnelle pour stocker commandes, bons et stocks.
- Notification et génération PDF automatisées via services ERP.
- Interface web pour administrateurs et agents de production.

Diagramme d'activité dédié

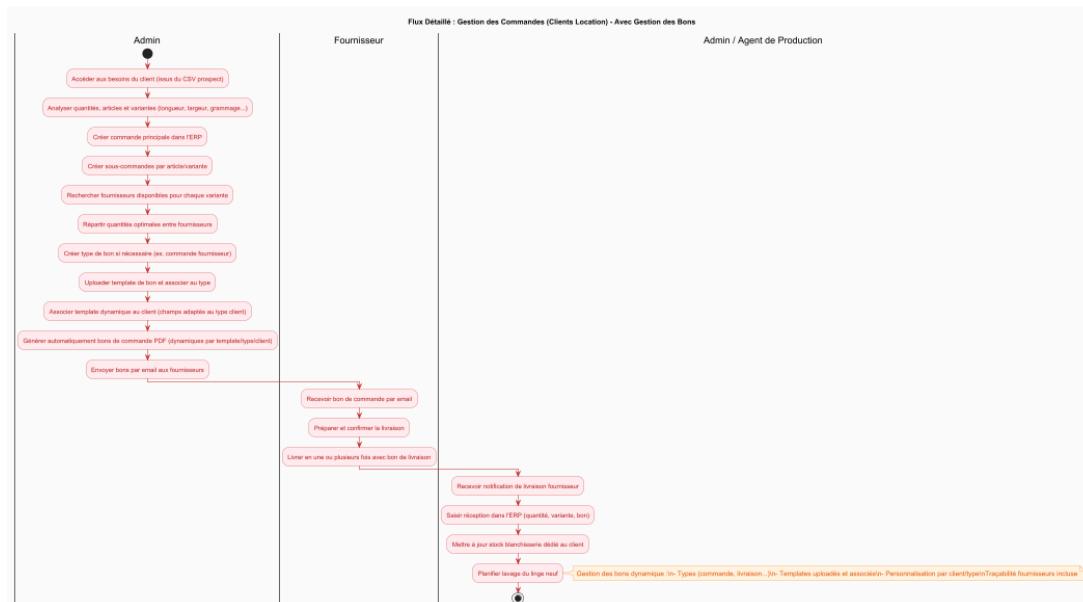


FIGURE 3.3 – Diagramme d'activité du flux de commandes (clients location)

3.2.3 Flux de gestion des stocks et livraisons

Description et objectifs

Le flux de gestion des stocks et des livraisons a pour objectif de suivre et contrôler le linge tout au long de sa circulation, depuis la blanchisserie jusqu'au client, et de gérer les échanges de linge sale/propre.

Objectifs principaux :

- Assurer le suivi des trois types de stocks : blanchisserie, linge, travail.
- Garantir la qualité et l'exactitude des livraisons aux clients.
- Optimiser la circulation du linge et la mise à jour des stocks en temps réel.
- Automatiser la génération et la personnalisation des bons dynamiques.
- Déetecter et notifier les écarts pour assurer une traçabilité complète.

Acteurs impliqués

- **Administrateur / Agent de production** : Prépare les bons et supervise les livraisons.

- **Chauffeur** : Livrer le linge propre et collecter le linge sale.
- **Agent chez le client** : Vérifie la réception et met à jour le stock lingerie.
- **Valle** : Participe à l'échange quotidien de linge avec l'agent.

Processus étape par étape

1. Créer le type de bon (livraison, collecte) si nécessaire.
2. Uploader et associer le template de bon dynamique.
3. Préparer la livraison du linge propre.
4. Générer automatiquement le bon PDF et l'envoyer au client.
5. Charger le camion et livrer le linge au client.
6. Contrôler la quantité et l'état du linge reçu par l'agent.
7. Signer et enregistrer la réception via l'application mobile.
8. Échanger quotidiennement le linge sale contre le linge propre et mettre à jour le stock de travail.
9. Collecter le linge sale et le retourner à la blanchisserie.
10. Suivi centralisé par l'administrateur : visualisation en temps réel, rapports et inventaires.
11. Déetecter et notifier automatiquement tout écart par email ou application.

Trois types de stocks

- **Blanchisserie** : Stock central dans l'entreprise.
- **Lingerie** : Stock chez le client, sous contrôle.
- **Travail** : Linge en cours d'utilisation chez le client.

Gestion avancée des bons dynamiques

- Bons PDF générés automatiquement selon le type et le client.
- Templates uploadés et associés dynamiquement aux bons.
- Historique et traçabilité des bons et échanges.

Liens avec les autres flux

- Les commandes clients alimentent les besoins de livraison et de stock.
- Les réclamations sont générées si des écarts ou anomalies sont détectés.
- Les agents et chauffeurs interagissent avec le module personnel pour les affectations et plannings.

Intégration technique

- Backend centralisé (Spring Boot) pour la gestion des stocks, livraisons et bons.
- Base de données relationnelle pour stocker les informations des stocks, bons et livraisons.
- Application mobile pour la saisie et validation des bons par les agents et chauffeurs.
- Notifications automatiques en cas d'écart via email ou application.

Diagramme d'activité dédié

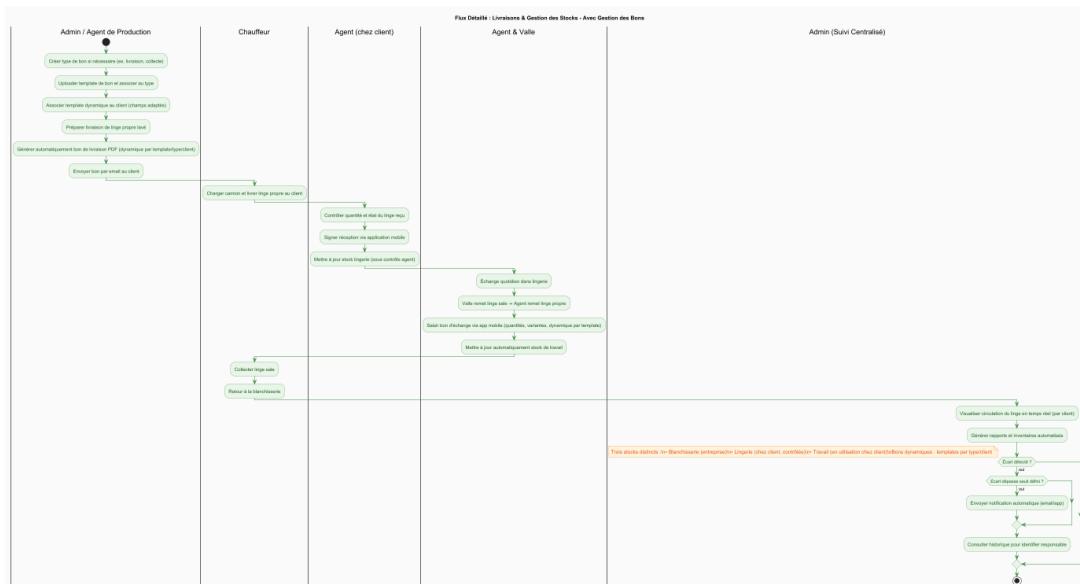


FIGURE 3.4 – Diagramme d'activité du flux de gestion des stocks et livraisons

3.2.4 Flux de gestion du personnel (transversal)

Description et objectifs

Le flux de gestion du personnel a pour objectif de coordonner toutes les actions des agents, chauffeurs, et du personnel interne afin de garantir le bon fonctionnement de l'ERP de blanchisserie et location de linge. Il couvre notamment la gestion des affectations, plannings et le suivi des interactions avec les autres flux (commandes, livraisons, réclamations).

Objectifs principaux :

- Assurer une organisation efficace du personnel et des ressources.
- Centraliser et tracer les décisions et interventions sur tous les flux.
- Faciliter la communication interne et la résolution rapide des problèmes.
- Suivre les performances et responsabilités des agents et chauffeurs.

Acteurs impliqués

- **Administrateur** : Supervise la gestion globale du personnel et affectations.
- **Agents internes / production** : Réalisent les tâches de blanchisserie et saisissent les informations dans l'ERP.
- **Chauffeurs** : Assurent la livraison et la collecte du linge.
- **Valle ou Agent chez client** : Communiquent avec l'ERP pour signaler les problèmes et gérer les échanges.

Processus étape par étape

1. Déetecter problème ou besoin sur site client.
2. Ouvrir le module réclamation dans l'application mobile.
3. Décrire le problème avec texte et joindre éventuellement photos ou enregistrements audio.
4. Envoyer la réclamation pour traitement.
5. Administrateur ou agent interne reçoit notification immédiate.
6. Ouvrir la conversation dédiée et consulter le contenu multimédia.
7. Répondre et échanger jusqu'à résolution complète.
8. Prendre décision appropriée (correction, remplacement, intervention).
9. Clore la réclamation avec statut final pour traçabilité.

Liens avec les autres flux

- Les réclamations impactent directement les flux de commandes et livraisons.
- La gestion du personnel transversal assure la disponibilité et coordination des agents, chauffeurs et production.
- Les notifications et historiques de réclamation sont centralisés dans l'ERP pour suivi global.

Intégration technique

- Backend centralisé (Spring Boot) pour la gestion des réclamations et suivi du personnel.
- Base de données relationnelle pour stocker les informations des réclamations, échanges et affectations.
- Application mobile pour la saisie et suivi des réclamations par agents et Valle.
- Notifications en temps réel pour assurer des interventions rapides.

Diagramme d'activité dédié

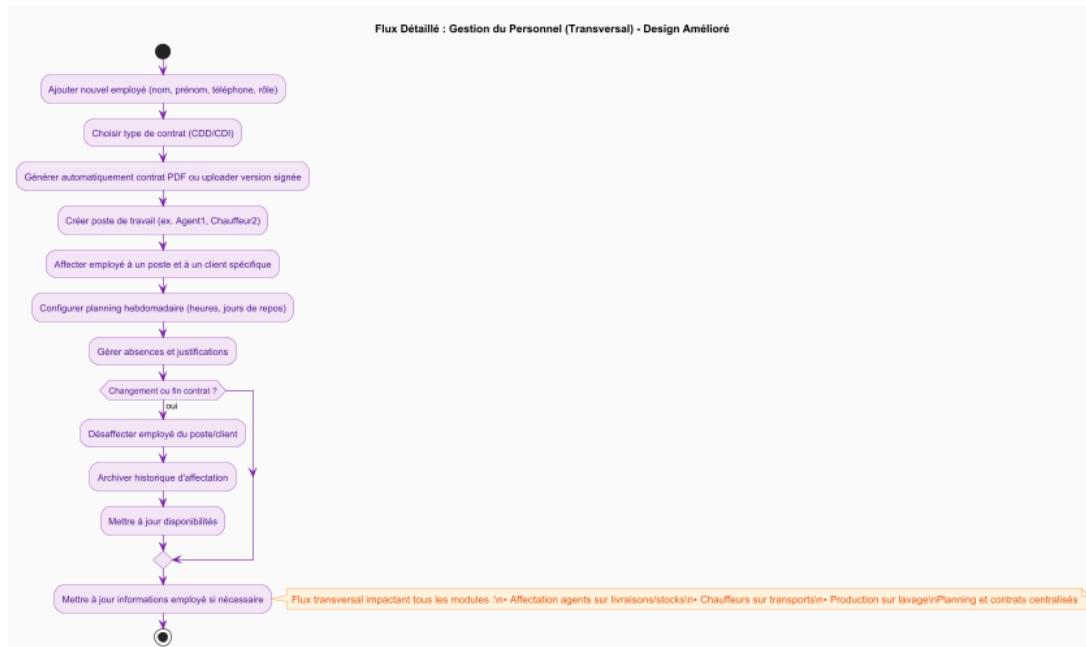


FIGURE 3.5 – Diagramme d'activité du flux de gestion du personnel et réclamations

3.2.5 Flux de réclamations

Description et objectifs

Le flux de réclamations a pour objectif de gérer efficacement les problèmes signalés par les clients ou les agents sur site. Il assure la traçabilité des incidents, la communication rapide entre les acteurs concernés et la prise de décision pour corriger ou résoudre les problèmes.

Objectifs principaux :

- Permettre aux clients et agents de signaler facilement tout problème.
- Centraliser et suivre toutes les réclamations dans l'ERP.
- Faciliter la communication multimédia (texte, photo, audio) entre les acteurs.
- Garantir la documentation et le suivi pour la traçabilité et l'analyse future.

Acteurs impliqués

- **Valle ou Agent chez client** : Signale les problèmes et saisit les réclamations via l'application mobile.
- **Administrateur / Agent interne** : Reçoit les réclamations, échange avec les utilisateurs et prend des décisions.
- **Client** : Peut également initier une réclamation via l'interface prévue.

Processus étape par étape

1. Détection d'un problème par Valle, Agent ou Client.
2. Ouverture du module réclamation dans l'application mobile.
3. Description du problème avec texte, photos et/ou enregistrement audio.
4. Envoi de la réclamation à l'ERP.
5. Notification immédiate à l'administrateur ou agent interne.
6. Consultation de la réclamation et des contenus multimédia.
7. Échange avec l'expéditeur pour clarification si nécessaire.
8. Prise de décision (correction, remplacement, intervention sur site).
9. Clôture de la réclamation avec mise à jour du statut.

Liens avec les autres flux

- Les réclamations peuvent impacter directement les flux de commandes et de livraisons.
- Elles sont liées à la gestion du personnel pour coordonner les interventions.
- Les notifications et historiques sont centralisés dans l'ERP pour analyse et reporting.

Intégration technique

- Backend centralisé (Spring Boot) pour gérer les réclamations et l'historique.
- Base de données relationnelle pour stocker le détail des réclamations et les échanges multimédias.
- Application mobile pour la saisie et le suivi des réclamations par Valle et agents.
- Système de notifications en temps réel pour assurer une réactivité maximale.

Diagramme d'activité dédié

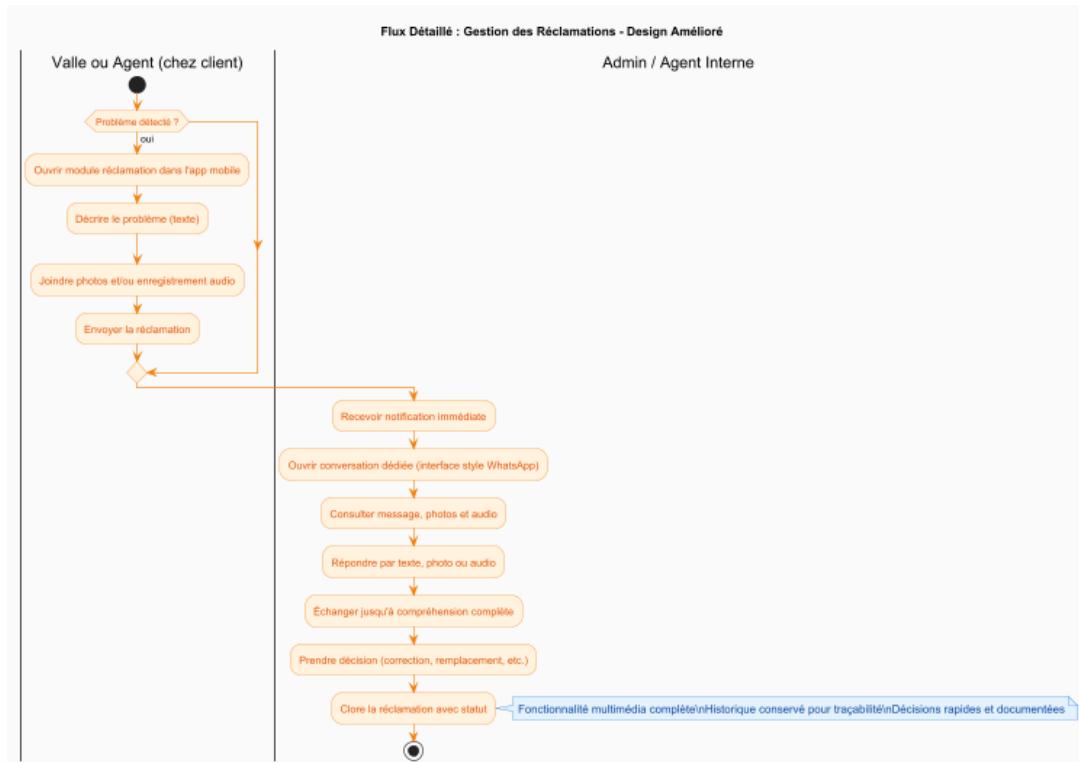


FIGURE 3.6 – Diagramme d'activité du flux de réclamations

3.3 Autres éléments de conception

3.3.1 Diagramme de classes métier

Le diagramme de classes métier présente la structure des principales entités de l'ERP de blanchisserie et location de linge. Il permet de visualiser les relations entre utilisateurs, clients, prospects, articles, commandes, stocks, livraisons et réclamations.

Ce diagramme illustre :

- Les **Utilisateurs et rôles** (Admin, Client, Prospect, Agents internes, Personne) héritant d'une classe abstraite Users pour l'authentification et la gestion des droits.
- La **Gestion du personnel** avec Contrat, Affectations, Absences.
- Les **Prospections et formulaires dynamiques** (CustomForm,FormField, FormSubmission, ProspectDetails).
- Les **Articles et fournisseurs** (Category, Article, Variant, Fournisseur) et leurs relations.
- Les **Commandes et livraisons** (Commande, DemandeArticle, Demande, LivraisonFournisseur, LivraisonClient).
- Les **Stocks et bons dynamiques**, avec calculs et gestion des écarts.

- La **Gestion des réclamations et chat**, incluant les échanges multimédias entre clients et agents.

Remarques techniques :

- Les classes sont liées par des associations indiquant la multiplicité et la direction des relations.
- Les données clients sont stockées à la fois dans PostgreSQL (relationnel) et MongoDB (documents, formulaires, chat).
- Les bons dynamiques et les calculs permettent de suivre les stocks de blanchisserie, lingerie et clients.

Diagramme

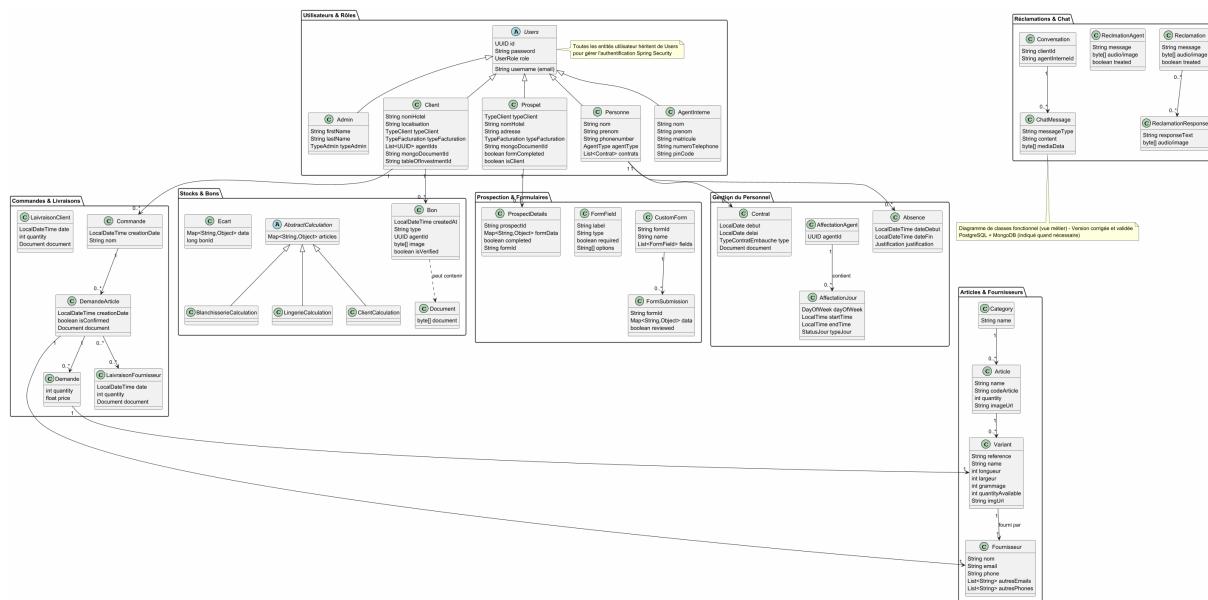


FIGURE 3.7 – Diagramme de classes métier de l'ERP

3.3.2 Diagramme d'activité global synthétique

Le diagramme ci-dessous présente une vue synthétique et globale des flux de l'ERP de blanchisserie et location de linge. Il illustre les interactions principales entre les différents acteurs (Prospects, Clients, Admin, Fournisseurs, Agents, Chauffeurs, Valle) et les étapes clés des processus : prospection, validation client, commandes, livraison, gestion des stocks et traitement des réclamations.

Remarques :

- Les flux sont représentés horizontalement avec des swimlanes par acteur.
- Chaque étape métier majeure est incluse : formulaire prospect, validation contrat, création de commandes, livraison, lavage, échanges linge propre/sale, traitement des réclamations, gestion du personnel.

- Le suivi en temps réel et la notification automatique des écarts sont également représentés.

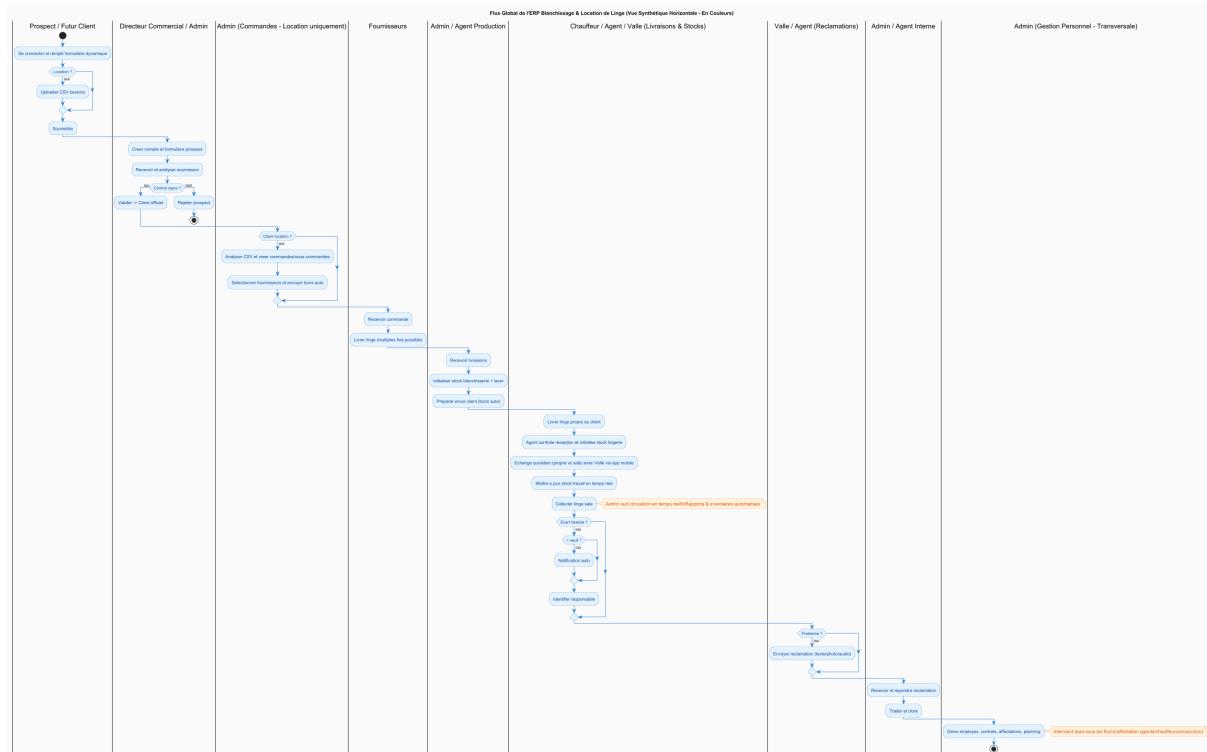


FIGURE 3.8 – Diagramme d'activité global synthétique de l'ERP

3.3.3 Modélisation des données

L'ERP utilise deux types principaux de stockage de données afin d'assurer performance et flexibilité :

- **PostgreSQL (bases relationnelles)** : Stocke les entités structurées et fortement liées, telles que :
 - Utilisateurs (Admin, Client, Prospect, Agents)
 - Commandes et sous-commandes
 - Articles et variantes
 - Stocks et bons dynamiques
 - Contrats et affectations du personnel
- **MongoDB (bases NoSQL / documents)** : Stocke les données non structurées ou semi-structurées, telles que :
 - Formulaires dynamiques soumis par les prospects/clients
 - Historique des échanges multimédia (réclamations, chats)
 - Documents PDF ou images associées aux bons ou contrats

Cette modélisation permet de gérer efficacement les différents flux de l'ERP tout en conservant traçabilité, fiabilité et souplesse pour les documents non structurés.

Chapitre 4

Réalisation

4.1 Choix technologiques et outils

-  **Java 17** : Choisi pour sa robustesse, sa maturité, sa compatibilité avec Spring Boot et sa sécurité.
-  **Spring Boot** : Permet de créer rapidement un backend RESTful, gérant l'authentification, la sécurité et la persistance avec PostgreSQL/MongoDB.
-  **Angular** : Framework moderne pour le développement d'interfaces web dynamiques et réactives côté frontend.
-  **Flutter** : Choisi pour le développement mobile cross-platform, permettant d'avoir une seule base de code pour iOS et Android.
-  **PostgreSQL** : Base relationnelle robuste et performante, adaptée à la gestion des utilisateurs, commandes et stocks.
-  **MongoDB** : Base NoSQL pour stocker des données flexibles, notamment les formulaires dynamiques et les historiques de chat/réclamations.
-  **Docker** : Conteneurisation de l'application pour faciliter le déploiement et assurer la cohérence entre environnements.
-  **GitHub Actions** : Automatisation du CI/CD pour tester, construire et déployer automatiquement l'application.
-  **PlantUML** : Création rapide de diagrammes UML pour la conception et la documentation technique.

4.2 Architecture technique détaillée

L'architecture technique de l'ERP Blanchissage Location de Linge repose sur un modèle **centralisé et modulaire**, permettant de séparer clairement les différentes couches de l'application tout en assurant leur communication fluide.

4.2.1 Couches et composants principaux

- **Frontend Web** : Applications Angular et React pour l'interface utilisateur (tableau de bord, gestion des commandes et des stocks).
- **Frontend Mobile** : Application Flutter pour les interactions clients, agents et Valle, avec saisie des bons et gestion des réclamations.
- **Backend** : Spring Boot (Java 17) pour la logique métier, les services REST, la gestion des utilisateurs et l'orchestration des flux.
- **Bases de données** :
 - **PostgreSQL** : Gestion des données relationnelles (clients, commandes, stocks, contrats).
 - **MongoDB** : Gestion des données NoSQL flexibles (formulaires dynamiques, historique chat et réclamations).
- **Conteneurisation et orchestration** : Docker pour encapsuler les services backend et frontend, garantissant un déploiement cohérent.
- **CI/CD** : GitHub Actions pour automatiser le test, la construction et le déploiement de l'application.

4.2.2 Flux de communication et protocoles

- Les frontends (web et mobile) communiquent avec le backend via des API REST sécurisées (JWT pour l'authentification).
- Le backend interagit avec PostgreSQL et MongoDB selon le type de données à traiter.
- Les notifications temps réel et la mise à jour des stocks peuvent utiliser WebSocket ou polling côté client.
- Les documents PDF (bons de commande/livraison) sont générés côté backend et transmis aux clients/fournisseurs par email.

4.2.3 Schéma synthétique de l'architecture technique

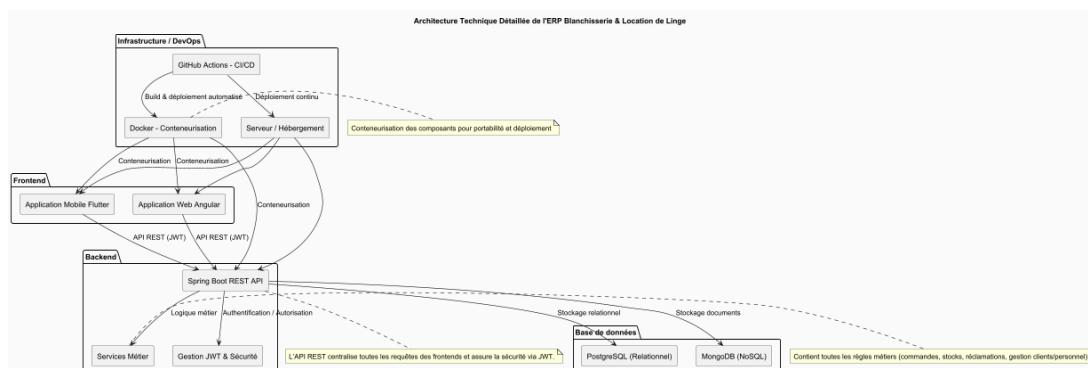


Figure : Schéma de l'architecture technique détaillée de l'ERP montrant les interactions

entre frontends, backend, bases de données et services auxiliaires.

4.3 Méthodologie de travail et encadrement

Le projet a été réalisé dans un contexte professionnel réel : nous avons travaillé en mode **client – équipe de développement**, notre encadrant jouant le rôle de **directeur commercial chez ABM Maroc** (client réel). Cette immersion nous a permis une compréhension approfondie des enjeux métier et une adaptation continue de la solution aux besoins de l'entreprise.

Nous avons adopté une **méthodologie Agile inspirée de Scrum**, particulièrement adaptée à un projet complexe aux exigences évolutives. Les itérations courtes (sprints de 1 à 2 semaines) ont permis une livraison progressive et une validation régulière.

Les pratiques clés mises en œuvre étaient :

- Réunions régulières (hebdomadaires ou bi-hebdomadaires) avec l'encadrant pour recueillir/prioriser les besoins, valider les livrables du sprint précédent et intégrer les retours.
- Définition du backlog du sprint suivant à l'issue de chaque réunion.
- Développement, tests internes et démonstration lors de la revue de sprint.
- Cycle répétitif : planification → développement → revue → rétrospective.

Cette approche a assuré une grande réactivité, une réduction des risques et une alignement parfait avec les attentes de l'entreprise.

La collaboration en équipe (trois membres) a été facilitée par **Git/GitHub** (branches par fonctionnalité, pull requests, revues de code), **Docker** pour la conteneurisation et **GitHub Actions** pour les pipelines CI/CD automatisés.

L'encadrement de **Monsieur Rachid Nidali** a été déterminant : ses retours métier réguliers ont guidé les choix fonctionnels et garanti l'adéquation de la solution aux contraintes opérationnelles d'ABM Maroc.

4.4 Outils de développement

Le développement de cet ERP a nécessité l'utilisation d'un ensemble d'outils professionnels et performants, couvrant à la fois le développement frontend (web et mobile), le développement backend, la gestion de bases de données, ainsi que la conception et la modélisation. Cette section présente les principaux outils utilisés et justifie leur choix.

4.4.1 Environnements de développement intégrés (IDE)

IntelliJ IDEA

IntelliJ IDEA a été utilisé pour le développement backend avec Spring Boot (Java 17). Ce choix s'explique par plusieurs raisons :

- Support avancé et natif pour les projets Spring Boot (autocomplétion intelligente, détection automatique des annotations, navigation facilitée dans le code).
- Outils de débogage puissants et intégrés, essentiels pour identifier rapidement les problèmes dans la logique métier complexe.
- Gestion automatique des dépendances Maven/Gradle avec suggestions intelligentes.
- Refactoring avancé et détection proactive des erreurs de code.
- Interface fluide et ergonomique permettant une productivité élevée.

IntelliJ IDEA nous a permis de nous concentrer pleinement sur la logique métier et les flux interconnectés de l'ERP, tout en minimisant les erreurs de configuration et les problèmes techniques.

Visual Studio Code

Visual Studio Code (VS Code) a été retenu pour le développement des applications web avec Angular. Ses principaux atouts sont :

- Légèreté et rapidité d'exécution, même sur des projets volumineux.
- Écosystème riche d'extensions pour Angular (Angular Language Service, TSLint, Prettier, etc.).
- Terminal intégré facilitant l'exécution des commandes Angular CLI.
- Support natif de TypeScript avec autocomplétion et vérification des types en temps réel.
- Intégration facilitée avec les outils de versioning.

VS Code a permis un développement frontend rapide et efficace, avec un contrôle précis sur le code TypeScript et les templates Angular.

Android Studio

Android Studio a été utilisé pour le développement des applications mobiles avec Flutter. Bien que Flutter dispose de son propre SDK, Android Studio offre :

- Support complet et stable des émulateurs Android et iOS.
- Outils de débogage Flutter intégrés (Flutter Inspector, DevTools).
- Gestion facilitée des configurations d'émulateurs (versions OS, résolutions d'écran variées).
- Performance optimale lors des tests sur émulateurs multiples.
- Interface familière pour le développement mobile multiplateforme.

Nous avons créé et utilisé plusieurs émulateurs (Android 10, 11, 12 et iOS via simulateurs) pour tester les applications mobiles dans différents scénarios d'utilisation (agent terrain, valle) et garantir une compatibilité maximale.

4.4.2 Conteneurisation et déploiement

Docker

Docker a été utilisé pour conteneuriser les différents services de l'application (backend Spring Boot, bases de données PostgreSQL et MongoDB). Les avantages de cette approche sont :

- Reproductibilité des environnements : chaque développeur et chaque serveur de déploiement dispose du même environnement, éliminant les problèmes classiques du type « ça marche sur ma machine ».
- Facilité de déploiement : les conteneurs Docker peuvent être déployés rapidement sur n'importe quel serveur supportant Docker.
- Isolation des services : chaque service (backend, PostgreSQL, MongoDB) fonctionne dans son propre conteneur, évitant les conflits de dépendances.
- Gestion simplifiée des dépendances et des configurations via les fichiers `Dockerfile` et `docker-compose.yml`.

Docker a considérablement simplifié le déploiement et la mise en production de l'ERP, tout en assurant une cohérence entre les environnements de développement, de test et de production.

GitHub Actions (CI/CD)

Pour automatiser les processus de tests et de déploiement, nous avons mis en place des pipelines **CI/CD** avec **GitHub Actions**. Ces pipelines permettent :

- L'exécution automatique des tests unitaires et d'intégration à chaque modification du code.
- La compilation et le *build* automatiques des applications (backend et frontend).
- La détection précoce des erreurs et des régressions avant la validation du code.
- Le déploiement automatisé vers les environnements de staging ou de production après validation.

Cette approche a permis de maintenir une qualité de code élevée tout au long du projet et d'accélérer les cycles de livraison.

4.4.3 Outils de modélisation et de conception

StarUML

Pour la phase de conception et de modélisation, nous avons utilisé **StarUML**. Ce choix s'explique par :

- Support complet de tous les diagrammes UML nécessaires au projet (diagrammes de cas d'utilisation, de classes, d'activités, de séquence).
- Interface intuitive et professionnelle facilitant la création rapide de diagrammes complexes.
- Export de haute qualité des diagrammes (PNG, PDF) pour intégration dans le rapport et les présentations.
- Capacité à gérer des modèles volumineux et interconnectés, essentielle pour un ERP de cette envergure.

Étant donné la complexité et l'interconnexion des flux métier de cet ERP (prospection, commandes, stocks, livraisons, personnel, réclamations), une conception solide et rigoureuse était indispensable. StarUML nous a permis de modéliser l'ensemble du système de manière claire, cohérente et communicable, facilitant ainsi la compréhension commune au sein de l'équipe et avec l'encadrant.

4.4.4 Autres outils et bibliothèques

En complément des outils principaux, plusieurs bibliothèques et services ont été intégrés :

- **Postman** : pour tester les API REST du backend de manière interactive et valider le bon fonctionnement des endpoints.
- **DBeaver** : client de bases de données universel utilisé pour gérer et visualiser les données dans PostgreSQL et MongoDB.
- **iText** : bibliothèque Java pour la génération dynamique de documents PDF personnalisés (bons de commande, bons de livraison, bons de collecte, contrats).
- **AG Grid** : bibliothèque JavaScript pour la création de tableaux de données avancés et interactifs dans les applications Angular, permettant le tri, le filtrage, la pagination et l'export de données.
- **API Gmail** : pour l'envoi automatique d'emails (identifiants de connexion, bons de commande, notifications d'écart).

4.4.5 Synthèse des outils

Le tableau ci-dessous récapitule les principaux outils utilisés et leur rôle dans le projet :

Outil	Rôle
IntelliJ IDEA	Développement backend (Spring Boot, Java 17)
Visual Studio Code	Développement frontend web (Angular, TypeScript)
Android Studio	Développement mobile (Flutter, Dart) et gestion des émulateurs
Docker	Conteneurisation des services (backend, bases de données)
GitHub Actions	CI/CD : automatisation des tests, builds et déploiements
StarUML	Conception et modélisation UML (use case, classes, activités, séquence)
Postman	Tests des API REST
DBeaver	Gestion et visualisation des bases de données (PostgreSQL, MongoDB)
iText	Génération dynamique de bons PDF personnalisés
AG Grid	Tableaux de données avancés et interactifs (Angular)
API Gmail	Envoi automatique d'emails

TABLE 4.1 – Récapitulatif des principaux outils de développement utilisés

Cette combinaison d'outils professionnels et complémentaires a permis de mener à bien un projet d'envergure, en garantissant qualité, performance, maintenabilité et évolutivité de l'ERP développé.

4.5 Difficultés rencontrées et solutions

Le développement d'un ERP complet et interconnecté pour une entreprise de blanchisserie et de location de linge a présenté des défis techniques et conceptuels majeurs. Cette section détaille les principales difficultés rencontrées ainsi que les solutions apportées pour les surmonter.

4.5.1 Flexibilité et variété des données métier

Nature du problème

La principale difficulté rencontrée au cours de ce projet résidait dans la **flexibilité et la variété des données** à gérer. En effet, chaque client possède des besoins spécifiques et des exigences particulières qui varient considérablement selon son secteur d'activité :

- Un hôtel n'a pas les mêmes besoins qu'un hôpital ou qu'un restaurant.
- Les formulaires de prospection doivent être entièrement dynamiques et personnalisables selon le type de client.
- Les bons (livraison, collecte, réception) doivent être adaptés à chaque client, avec des templates différents et des champs variables.

- Les articles de linge comportent de multiples variantes (dimensions, grammage, matière, couleur) qui diffèrent selon les fournisseurs et les clients.
- Les workflows opérationnels (prospection, commandes, livraisons, stocks) doivent s'adapter aux particularités de chaque contrat client.

Cette **extrême variabilité et ce dynamisme** des données selon chaque client et chaque prospect constituaient un défi majeur, car ils imposaient de concevoir un système à la fois rigoureux et suffisamment souple pour s'adapter à des contextes métier très différents.

Complexité de la modélisation hybride

La difficulté était amplifiée par le fait que ces données, bien que flexibles et dynamiques, devaient être gérées dans un **cadre relationnel strict**. En d'autres termes, il fallait :

- Gérer des données **non relationnelles** (formulaires dynamiques avec champs variables, templates de bons personnalisés, soumissions de prospects aux structures hétérogènes) dans un environnement structuré et fortement relationnel (PostgreSQL).
- Assurer la **cohérence et l'intégrité des données** malgré leur nature hétérogène.
- Maintenir des **relations complexes** entre entités (clients, commandes, articles, variantes, stocks, fournisseurs, personnel) tout en permettant une grande flexibilité au niveau des attributs et des configurations.
- Éviter la prolifération de tables ou de colonnes inutiles, tout en garantissant l'évolutivité du système.

Cette problématique s'apparentait à **gérer des données non relationnelles dans une boîte relationnelle**, ce qui représentait un paradoxe architectural nécessitant une réflexion approfondie et des compromis techniques judicieux.

Impact sur la phase de conception

Cette complexité a eu un impact majeur sur la **phase de conception**, qui a pris **beaucoup de temps** — bien plus que prévu initialement. Les principales raisons étaient :

- **Modélisation des entités** : il a fallu identifier les entités stables et structurées (clients, articles, commandes) tout en prévoyant des mécanismes pour gérer les parties dynamiques (champs de formulaires, attributs variables).
- **Choix architectural** : définir une architecture hybride combinant PostgreSQL (pour les données relationnelles) et MongoDB (pour les données flexibles et non structurées) tout en assurant la cohérence globale.
- **Diagrammes UML complexes** : la création des diagrammes de classes, d'activités et de séquence a nécessité plusieurs itérations pour capturer correctement toutes les relations et tous les flux métier interconnectés.
- **Validation avec l'encadrant** : chaque décision de conception devait être validée

avec le client (encadrant) pour s'assurer qu'elle répondait bien aux besoins réels et aux contraintes métier.

Cette phase de conception intensive, bien que chronophage, s'est révélée **absolument indispensable** pour la réussite du projet. Elle a permis d'éviter des erreurs coûteuses en phase de développement et de poser des bases solides pour un système évolutif et maintenable.

4.5.2 Solutions apportées

Pour surmonter ces difficultés, plusieurs solutions techniques et méthodologiques ont été mises en œuvre :

Architecture hybride de bases de données

Nous avons adopté une **architecture hybride** combinant deux systèmes de gestion de bases de données complémentaires :

- **PostgreSQL** pour toutes les données relationnelles et fortement structurées : clients, commandes, articles, variantes, stocks, personnel, affectations, contrats, fournisseurs, etc. Cette base assure l'intégrité référentielle et la cohérence des relations complexes.
- **MongoDB** pour toutes les données dynamiques, flexibles et non structurées : formulaires dynamiques de prospection, soumissions de formulaires par les prospects, templates de bons personnalisés, historique des conversations et réclamations multimédia.

Cette séparation a permis de bénéficier du meilleur des deux mondes : la rigueur relationnelle de PostgreSQL pour le cœur métier stable, et la flexibilité schemaless de MongoDB pour les données variables.

Modélisation orientée métier avec entités génériques

Pour gérer la variabilité des formulaires et des bons, nous avons mis en place des **entités génériques et configurables** :

- **Formulaires dynamiques** : un système de métadonnées décrivant la structure des formulaires (types de champs, validations, options) stocké dans MongoDB, permettant de générer dynamiquement les formulaires côté frontend sans modifier le code.
- **Templates de bons** : des templates uploadés par l'administrateur et associés à des types de bons et à des clients spécifiques, avec injection dynamique des données lors de la génération PDF via iText.
- **Gestion des variantes d'articles** : modélisation des variantes comme entités à part entière liées aux articles, avec attributs configurables (dimensions, grammage,

matière), permettant une grande flexibilité sans multiplier les tables.

Conception itérative et validation continue

Pour éviter les erreurs conceptuelles coûteuses, nous avons adopté une **approche itérative** avec validation continue :

- Réalisation de plusieurs versions des diagrammes UML (classes, activités, séquence) avec révisions successives.
- Validation systématique de chaque décision de conception lors des réunions hebdomadaires avec l'encadrant.
- Prototypage rapide de certaines fonctionnalités critiques (formulaire dynamiques, génération de bons) pour valider la faisabilité technique avant la modélisation complète.
- Documentation rigoureuse des choix architecturaux et des compromis effectués.

Utilisation d'outils de modélisation performants

L'utilisation de **StarUML** a été essentielle pour gérer la complexité de la modélisation. Cet outil nous a permis de :

- Créer des diagrammes de classes volumineux et interconnectés avec une vue d'ensemble claire.
- Maintenir la cohérence entre les différents diagrammes (classes, activités, séquence).
- Communiquer efficacement les décisions de conception au sein de l'équipe et avec l'encadrant.
- Exporter des diagrammes de qualité professionnelle pour le rapport et les présentations.

4.5.3 Autres difficultés et solutions

Au-delà de la problématique principale de flexibilité des données, d'autres difficultés ont été rencontrées :

Gestion de la complexité des flux interconnectés

Les flux métier (prospection, commandes, stocks, livraisons, personnel, réclamations) sont fortement interdépendants, ce qui a nécessité :

- Une coordination rigoureuse entre les membres de l'équipe pour assurer la cohérence des interfaces et des API.
- Des tests d'intégration poussés pour vérifier le bon fonctionnement des interactions entre modules.
- Une documentation précise des API REST et des contrats d'interface.

Performance et optimisation

Certaines opérations (calcul des stocks en temps réel, génération de rapports d'inventaire, détection automatique des écarts) nécessitaient des optimisations :

- Mise en place d'indexés appropriés dans PostgreSQL pour accélérer les requêtes complexes.
- Mise en cache de certaines données fréquemment consultées.
- Optimisation des requêtes avec agrégations et jointures multiples.

Compatibilité multiplateforme (mobile)

Le développement des applications mobiles Flutter pour Android et iOS a nécessité :

- Des tests approfondis sur émulateurs et dispositifs réels pour garantir la compatibilité.
- L'adaptation des interfaces pour différentes tailles d'écran et résolutions.
- La gestion des permissions (appareil photo, microphone) pour les réclamations multimédia.

4.5.4 Bilan

Malgré les difficultés rencontrées, notamment la complexité de gérer des données flexibles et dynamiques dans un cadre relationnel rigoureux, les solutions mises en œuvre se sont révélées efficaces. La phase de conception, bien que longue et exigeante, a permis de poser des bases solides pour un système robuste, évolutif et parfaitement adapté aux besoins métier d'ABM Maroc. Le temps investi en conception a largement été compensé par une phase de développement plus fluide et moins sujette aux erreurs structurelles.

Chapitre 5

Tests et Validation

La phase de tests et de validation est cruciale pour tout système d'information, et particulièrement pour l'ERP développé pour ABM Maroc. Elle a permis de vérifier la conformité du système aux exigences, de corriger les anomalies et de valider la solution auprès de l'encadrant.

5.1 Stratégie de tests

Une approche pragmatique a été adoptée, privilégiant des **tests manuels rigoureux** plutôt que des frameworks automatisés complexes, afin de couvrir les aspects fonctionnels et d'intégration.

5.1.1 Types de tests réalisés

Tests fonctionnels

Les tests fonctionnels ont vérifié le bon fonctionnement de chaque module :

- **Prospection** : création de comptes, formulaires dynamiques, upload CSV, transformation prospect → client.
- **Commandes** : création de commandes et sous-commandes, sélection des fournisseurs, génération et envoi de bons PDF.
- **Stocks et livraisons** : gestion des différents stocks, génération et validation des bons, détection des écarts.
- **Personnel** : gestion des employés, contrats, plannings et absences.
- **Réclamations** : création, traitement et archivage des réclamations multimédia.

Tests d'intégration

Ces tests ont vérifié la cohérence et l'interconnexion des modules, incluant :

- Flux complet prospection → commandes → stocks.

- Intégration personnel modules opérationnels.
- Synchronisation mobile web.
- Génération et envoi automatique de documents.

Tests système

Évaluation globale du système dans des conditions proches de la réalité :

- Scénarios métier complets (journées types, commandes, livraisons, réclamations).
- Tests de charge simples (multi-applications simultanées).
- Cohérence et traçabilité des données.
- Vérification sécurité et autorisations.

5.1.2 Méthodologie de tests

- Tests manuels à la fin de chaque sprint.
- Cahiers de tests détaillés pour chaque module.
- Environnement dédié avec données fictives réalistes.
- Tests sur émulateurs et dispositifs réels.
- Validation croisée par les membres de l'équipe.

5.1.3 Outils utilisés

Postman, DBeaver, émulateurs Android Studio, navigateurs multiples et outils de développement web pour inspecter les requêtes, erreurs et performances.

5.2 Résultats et corrections

5.2.1 Anomalies détectées

Critiques : synchronisation des stocks, envoi d'emails volumineux, CSV mal formés, crash application mobile. **Majeures** : lenteur des tableaux, incohérence des dates, non-raffraîchissement des données, templates PDF incorrects. **Mineures** : affichage sur petits écrans, messages d'erreur peu explicites, accès à tort à certaines fonctions, incohérences visuelles.

5.2.2 Processus de correction

Identification → Priorisation → Analyse → Développement → Revue de code → Tests de régression → Déploiement et validation.

5.2.3 Statistiques des tests

Indicateur	Valeur
Cas de tests fonctionnels	127
Scénarios d'intégration	38
Anomalies critiques	12
Anomalies majeures	24
Anomalies mineures	41
Taux de correction critiques	100%
Taux de correction majeures	100%
Taux de correction mineures	95%
Cycles tests-corrections	4

TABLE 5.1 – Synthèse des tests et corrections

5.2.4 Améliorations post-tests

Barres de progression, messages de confirmation clairs, raccourcis clavier, optimisation du cache frontend, gestion partielle hors ligne pour mobile.

5.3 Validation avec l'encadrant

Validation progressive et finale grâce à des réunions de revue de sprint, échanges continus par email et messagerie, et démonstrations intermédiaires.

5.3.1 Validation finale

Tests exhaustifs des flux prospection, commandes, stocks, personnel, réclamations et fonctionnalités transversales. Scénarios métier complets validés avec succès. Vérification des exigences non fonctionnelles : performance, sécurité, ergonomie, fiabilité et traçabilité.

5.3.2 Retours de l'encadrant

Satisfaction globale, avec points positifs : adéquation aux besoins, flexibilité, interconnexion des flux, ergonomie, qualité des documents générés et vision stratégique pour l'évolution future.

5.3.3 Perspectives d'amélioration

- Module de facturation avancée.
- Dashboards analytiques.
- Application mobile pour les clients.
- Intégration avec des systèmes tiers.

Chapitre 6

Conclusion et Perspectives

6.1 Bilan du projet

Le projet de fin d'études a permis la conception et le développement d'un ERP interconnecté pour l'entreprise **ABM Maroc**, spécialisé dans le blanchissage et la location de linge professionnel. L'ensemble des processus métier principaux a été digitalisé : prospection, gestion des commandes, suivi des stocks multi-niveaux, livraison et collecte du linge, gestion du personnel et traitement des réclamations.

L'application repose sur une architecture modulaire et sécurisée, comprenant un **backend centralisé** sous Spring Boot, deux **applications web** (administrateur et prospect) et deux **applications mobiles** (agent terrain et valle). Cette structure a permis de garantir une communication fluide entre les différents flux, une traçabilité complète des opérations et une automatisation efficace des tâches répétitives.

Parmi les réussites majeures du projet, on peut citer :

- La **digitalisation complète** des flux métier, réduisant les erreurs humaines et accélérant le traitement des commandes et des livraisons.
- L'**automatisation et personnalisation** des bons (création de types, upload de templates, association aux clients et flux spécifiques).
- La **traçabilité en temps réel** des stocks et des écarts, permettant de suivre le linge depuis sa réception jusqu'à son utilisation chez le client.
- L'**interface multimédia pour les réclamations**, favorisant une communication rapide et efficace entre clients, agents et administrateurs.
- Une solution **scalable et évolutive**, prête à accueillir de futures fonctionnalités comme l'analyse prédictive ou la facturation avancée.

Globalement, le projet a atteint ses objectifs généraux et spécifiques, offrant une solution opérationnelle qui répond aux besoins identifiés et aux contraintes métier, techniques et temporelles.

6.2 Limites

Malgré les résultats satisfaisants obtenus, certaines limites ont été constatées lors de la réalisation et de la validation du projet :

- **Durée limitée du stage** : certaines fonctionnalités avancées n'ont pu être implémentées pleinement, notamment les modules de reporting avancé et d'analyse prédictive.
- **Données réelles limitées** : pour respecter la confidentialité des clients et des fournisseurs, les tests ont été réalisés sur des données de simulation, ce qui ne permet pas de valider toutes les interactions en conditions réelles.
- **Interface mobile encore perfectible** : certaines interactions de l'application mobile peuvent être optimisées pour l'ergonomie et la fluidité, notamment pour les agents terrain.
- **Intégration avec des services externes** : le système n'inclut pas encore de connecteurs avec des solutions tierces (ex. systèmes comptables, ERP externes ou services bancaires), ce qui pourrait limiter certaines automatisations futures.
- **Scalabilité en très grande échelle** : l'architecture a été testée sur un environnement local ou de petite production ; un déploiement à grande échelle nécessiterait une optimisation des performances et une infrastructure cloud dédiée.

Ces limites sont principalement liées aux contraintes de temps, de ressources et à la confidentialité des données utilisées lors du stage, mais elles ouvrent la voie à des améliorations futures.

6.3 Perspectives d'amélioration

Plusieurs axes d'évolution peuvent être envisagés pour renforcer et enrichir l'ERP développé :

- **Ajout de fonctionnalités analytiques et prédictives** : par exemple, la prédiction des besoins en linge selon la saisonnalité ou les historiques de consommation, et l'analyse des performances fournisseurs.
- **Amélioration de l'interface mobile** pour les agents et les clients, en optimisant l'expérience utilisateur et en introduisant des notifications en temps réel.
- **Intégration avec des systèmes externes** tels que la comptabilité, la facturation électronique ou les outils CRM, afin de centraliser encore davantage les opérations de l'entreprise.
- **Optimisation des performances et montée en charge** : mise en place d'une infrastructure cloud scalable, avec des bases de données optimisées pour gérer un grand nombre de clients et d'articles.
- **Extension des modules de personnalisation des bons** : permettre des règles

de génération plus avancées selon les types de clients, les flux, les périodes ou les évènements spécifiques.

- **Renforcement de la sécurité et de la conformité** : implémentation de mécanismes de cryptage avancés, authentification multi-facteurs et traçabilité renforcée pour répondre aux exigences légales et métier.
- **Automatisation avancée des notifications et alertes** : par e-mail, SMS ou application mobile pour anticiper les écarts de stock, les retards de livraison ou les réclamations clients.

Ces perspectives montrent que le projet constitue une base solide et modulaire, permettant un développement progressif vers un ERP complet, moderne et pleinement adapté aux besoins futurs de l'entreprise ABM Maroc.

Bibliographie

Annexe A

Captures d'écran des interfaces

A.1 Dashboard Administrateur

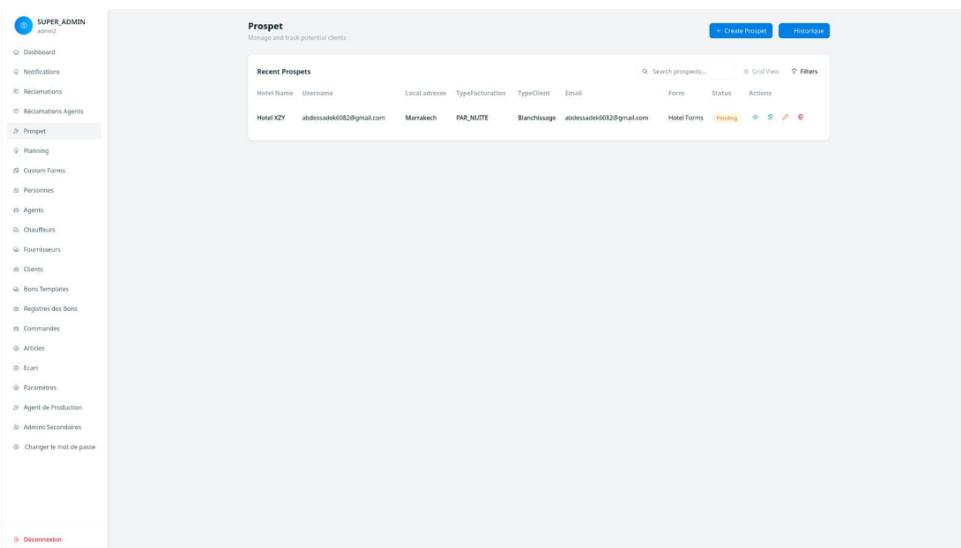


FIGURE A.1 – Interface du dashboard de l'administrateur.

A.2 Page de gestion de prospection

The screenshot shows the 'Prospect' management interface. On the left, a sidebar menu for 'SUPER_ADMIN' includes options like Dashboard, Notifications, Réclamations, Réclamations Agents, Prospect, Planning, Custom Forms, Personnes, Agents, Chauffeurs, Fournisseurs, Clients, Bons Templates, Registres des Bons, Commandes, Articles, Ecart, Paramètres, Agent de Production, Admins Secondaires, and Changer le mot de passe. Below this is a 'Déconnexion' link. The main area is titled 'Prospect' with the subtitle 'Manage and track potential clients'. It features a search bar and filter buttons for 'Grid View' and 'Filters'. A table lists recent prospects with columns for Hotel Name, Username, Local address, TypeFacturation, TypeClient, Email, Form, Status, and Actions. One entry is shown: Hotel KZY, abdessaadek6082@gmail.com, Marrakech, PAR_NUITE, Blanchissage, abdessaadek6082@gmail.com, Hotel Forms, Pending.

FIGURE A.2 – Exemple de gestion des prospects.

A.3 Page de gestion des stocks par bons

A.4 Page de création de commande

The screenshot shows the 'Gestion des commandes' (Command Management) page. The sidebar is identical to Figure A.2. The main area has a search bar with fields for 'Nom de la commande', 'Client' (set to 'Tous les clients'), 'Date exacte' (empty), 'Date minimum' (mm/dd/yyyy), 'Date maximum' (mm/dd/yyyy), 'Tri par date' (empty), and 'Aucun tri'. Buttons for 'Réinitialiser les filtres' and 'Rechercher' are at the bottom. Below this is a table titled 'Liste des commandes' (List of commands) with columns ID COMMANDE, NOM COMMANDE, DATE COMMANDE, CLIENT, and ACTION. One command is listed: 3038091402-4021-a219-148808C929, Commande K, 29/10/2025, AMBASSADEUR, with three small icons in the ACTION column.

FIGURE A.4 – Exemple de création de commande par un utilisateur.

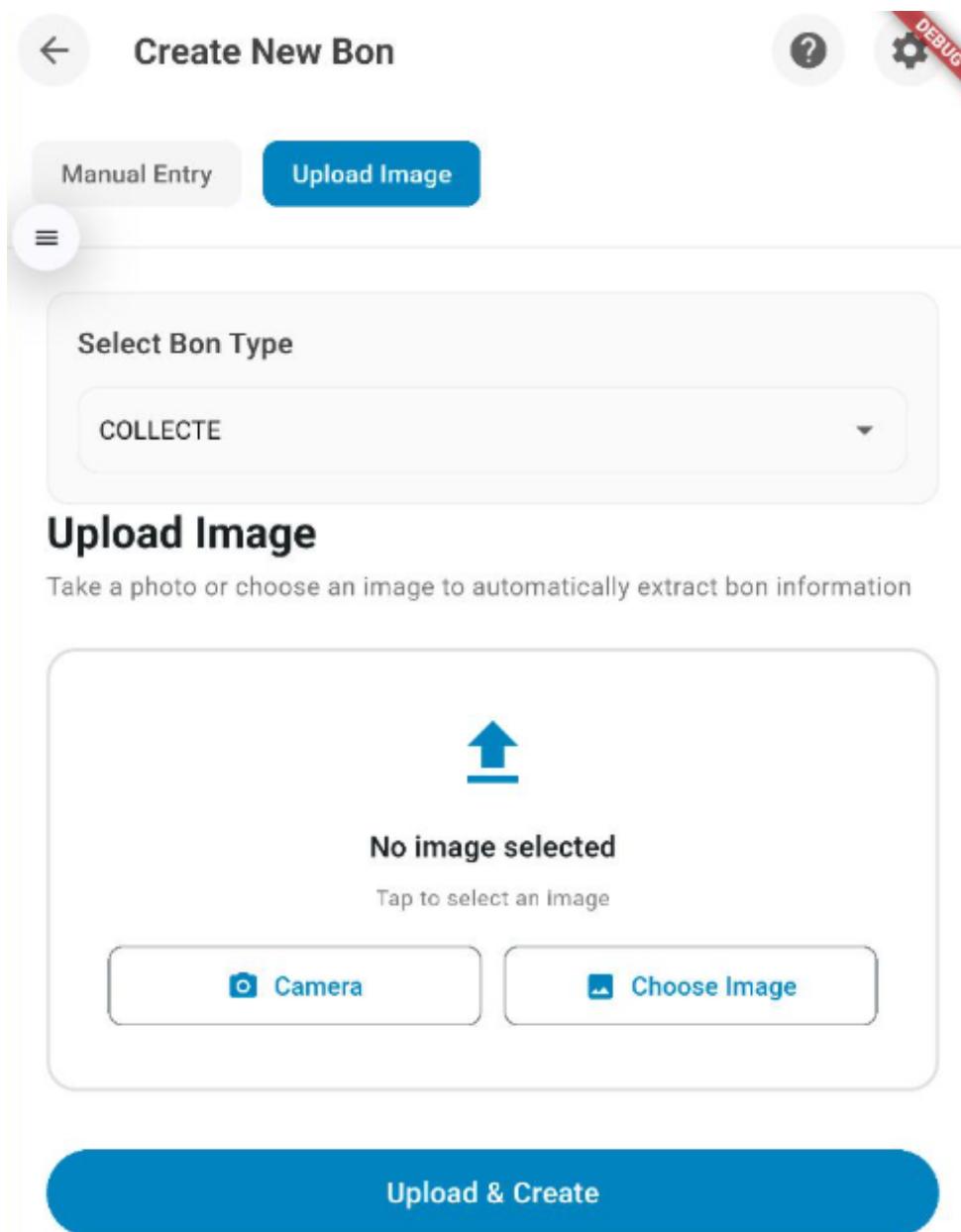


FIGURE A.3 – Interface de gestion des stocks permettant l'ajout des bons ainsi que le suivi des articles.

A.5 Interface Frontend Mobile

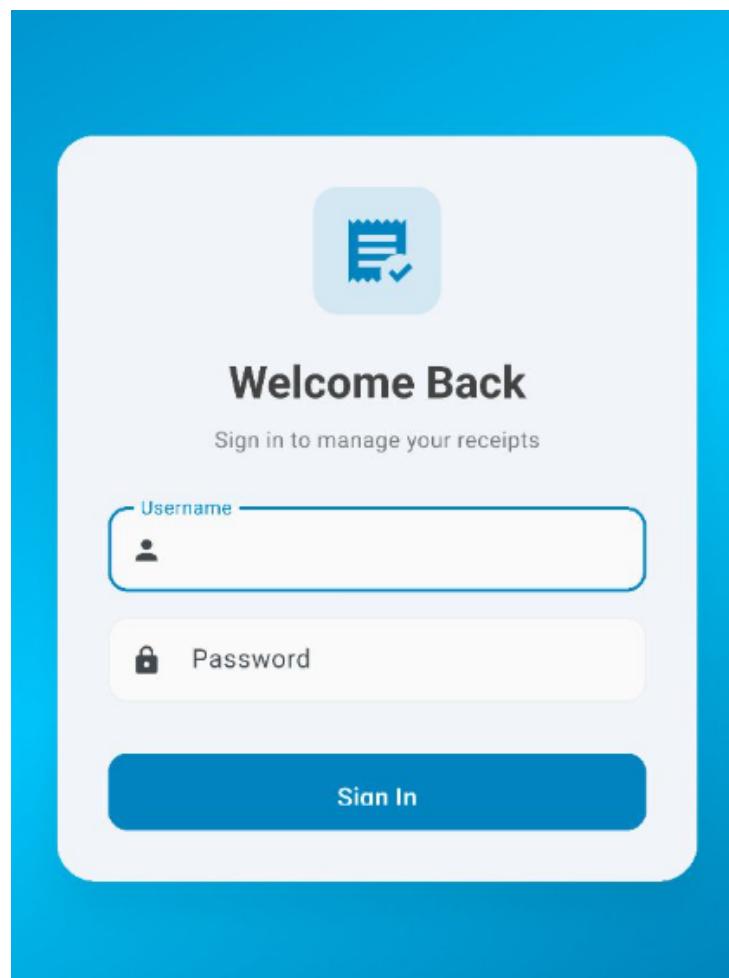


FIGURE A.5 – Interface mobile pour le client ou l'administrateur.

A.6 Interface de chat d'agent avec directeur

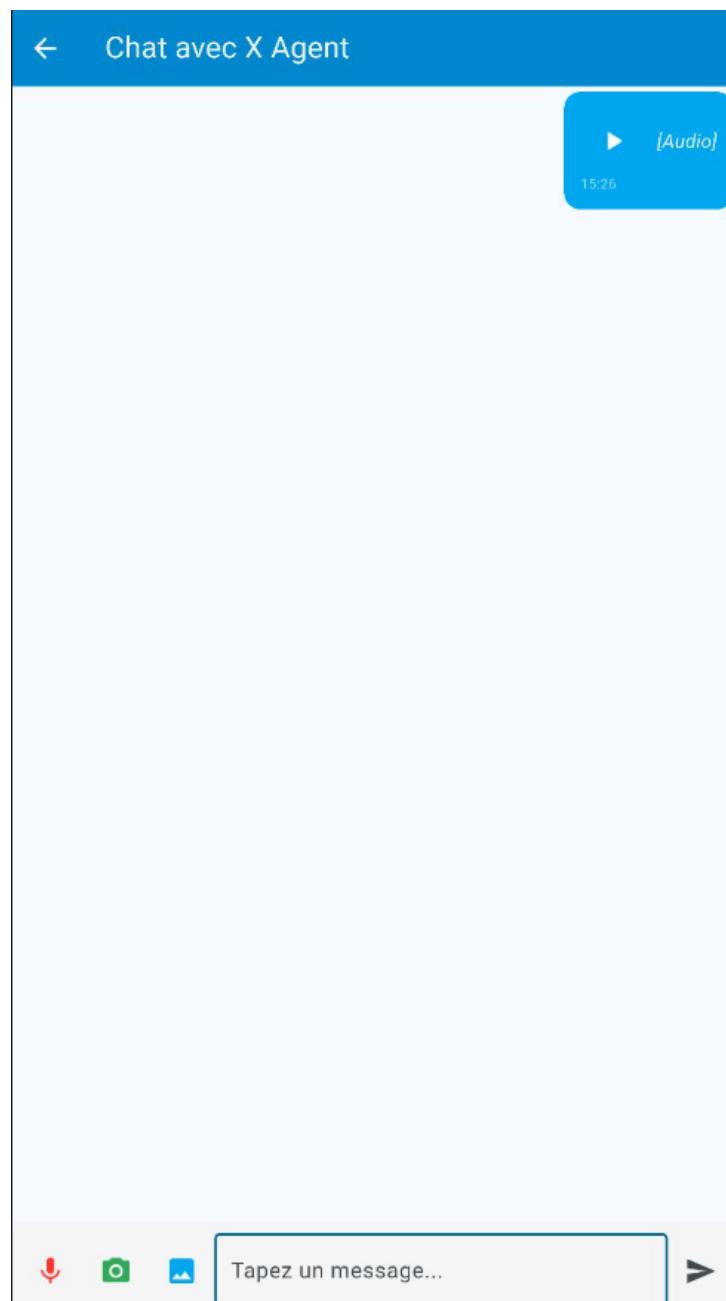


FIGURE A.6 – Interface mobile de chat d'agent avec l'administrateur.

A.7 Interface de chat pour soumettre des reclammmations

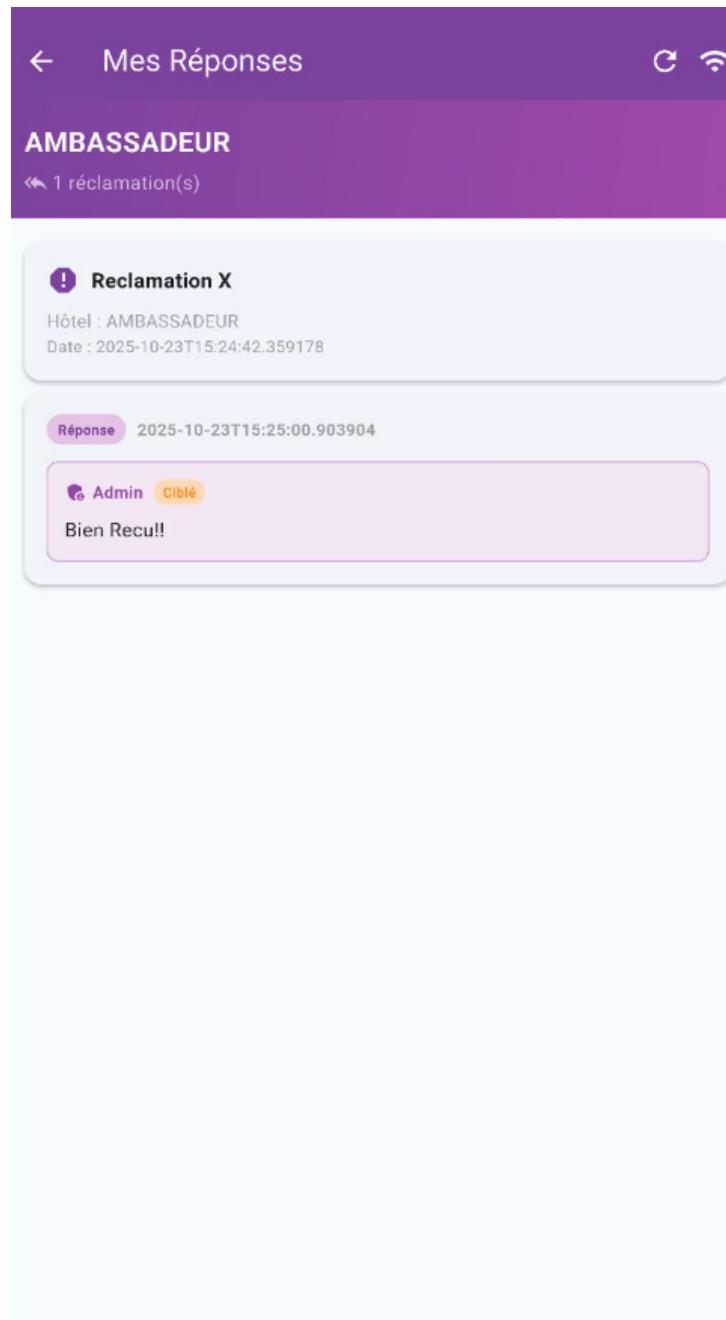


FIGURE A.7 – Interface de chat pour soumettre des reclammmations.

A.8 Gestion de templates client pour les bons

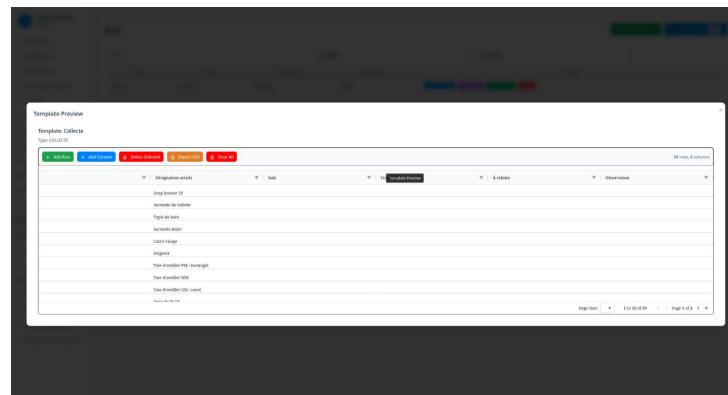


FIGURE A.8 – Interface mobile pour gestion des templates des bons client.

Annexe B

Exemples de code source

B.1 Service Angular

Code source du service Angular pour la gestion des clients.

B.2 Controller Spring Boot

Code source du controller Spring Boot pour la gestion des clients.

B.3 Composant Dashboard Angular

Code source du composant Angular affichant le dashboard et les statistiques.

Annexe C

Planning du projet

Le projet de fin d'études pour le développement de l'ERP d'ABM Maroc a été réalisé sur une période de **trois mois**, du **25 juillet 2025 au 25 septembre 2025**. Le planning a été structuré en plusieurs phases, avec des tâches clairement définies, afin d'assurer une progression cohérente et l'atteinte des objectifs fixés.

C.1 Phases du projet

- **Analyse des besoins** : 25/07/2025 – 07/08/2025 Comprend l'étude de l'existant, l'identification des flux métier, des acteurs et des exigences fonctionnelles et non fonctionnelles.
- **Conception du système** : 08/08/2025 – 28/08/2025 Définition de l'architecture globale, des diagrammes de classes et d'activités, ainsi que des modèles de données et interfaces.
- **Réalisation Backend** : 15/08/2025 – 11/09/2025 Développement du serveur Spring Boot, des services métier et des API pour la gestion des flux (prospection, commandes, stocks, personnel, réclamations).
- **Réalisation Frontend Web** : 15/08/2025 – 04/09/2025 Création des applications web pour l'administrateur et le client prospect, avec Angular et intégration des fonctionnalités dynamiques.
- **Réalisation Frontend Mobile** : 22/08/2025 – 11/09/2025 Développement des applications mobiles pour les agents terrain et la valle, avec Flutter et connexion aux API backend.
- **Tests et validation** : 12/09/2025 – 18/09/2025 Mise en place des scénarios de tests unitaires et fonctionnels, corrections des anomalies et validation finale avec l'encadrant.
- **Documentation et rapport** : 19/09/2025 – 25/09/2025 Rédaction du rapport, intégration des diagrammes UML, captures d'écran et finalisation de la présentation.

C.2 Diagramme Gantt (simulé avec PlantUML)

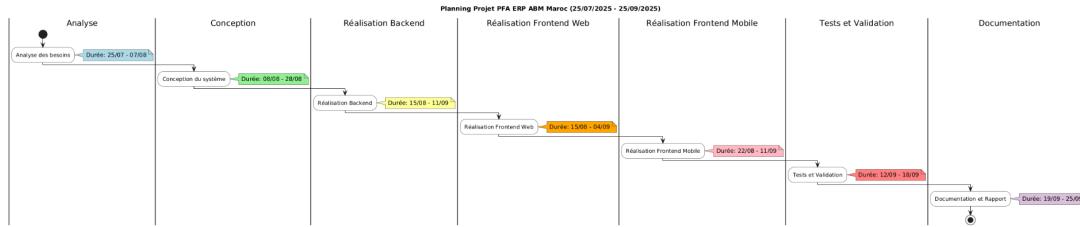


FIGURE C.1 – Diagramme de Gantt représentant le planning global du projet ERP (25/07/2025 – 25/09/2025).

Pour représenter visuellement le planning, nous avons utilisé un diagramme de type Gantt simulé avec **PlantUML**, avec des couleurs distinctes pour chaque phase afin de faciliter la lecture et la compréhension de la répartition temporelle des tâches.

Ce diagramme permet de visualiser rapidement les phases principales du projet et leur chevauchement éventuel, tout en facilitant la planification et le suivi de l'avancement des tâches.