



UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

Étude, conception, développement et déploiement d'un logiciel de gestion des visites de copropriété

Rapport de stage ST40 – A2026

MICHEL Arnaud

Département Informatique

Entreprise FL IMMO

3 Place de la République
90000 Belfort
www.prevot-immobilier.com

Tuteur en entreprise

VOIRIN Frédéric

Suiveur UTBM

GABER Jaafar

RAPPORT DE STAGE

Remerciements

Il est essentiel d'entamer ce rapport en exprimant ma gratitude envers les personnes qui, par leur assistance, leurs expériences, leur pédagogie et leur savoir-faire, ont contribué à faire de ce stage une expérience enrichissante, tant sur le plan professionnel que personnel.

Je remercie en premier lieu Lionel Forni pour m'avoir accueilli au sein de son agence et pour la confiance qu'il m'a accordée durant toute la durée de ce stage.

Je tiens à remercier tout particulièrement Frédéric Voirin, mon tuteur de stage, pour son accompagnement, ses conseils et sa disponibilité, qui ont joué un rôle déterminant dans la conduite et la réussite du projet.

Je remercie également Sophie Bouchard pour son aide dans la présentation du contexte métier, la compréhension des processus du service syndic et la clarification des besoins fonctionnels.

Enfin, je remercie mes collègues de bureau, Aldijana Ademovic et Pierre-Louis Valli, pour leur disponibilité, leurs échanges et l'ambiance de travail appréciable au quotidien.

Table des matières

REMERCIEMENTS	2
TABLE DES MATIERES	2
ABRÉVIATIONS ET GLOSSAIRE	5
ABRÉVIATIONS.....	5
GLOSSAIRE.....	5
1. INTRODUCTION.....	6
1.1 CONTEXTE GENERAL.....	6
1.2 PROBLÉMATIQUE	6
1.3 OBJECTIFS DU STAGE	6
1.4 PÉRIMÈTRE ET CONTRAINTES (IN / OUT).....	6
2. PRÉSENTATION DE L'ENTREPRISE.....	8
2.1 PRÉSENTATION GÉNÉRALE.....	8
2.2 ORGANISATION / SERVICES.....	8
2.2.1 ORGANIGRAMME DE L'ENTREPRISE:.....	10
2.3 SERVICE D'ACCUEIL : SYNDIC	11

3. CADRAGE INITIAL ET CAHIER DES CHARGES	15
3.1 ORIGINE DU BESOIN ET ENJEUX	15
3.2 UTILISATEURS CIBLES ET CAS D'USAGE	15
3.3 EXIGENCES FONCTIONNELLES	16
3.4 EXIGENCES NON FONCTIONNELLES (RNF)	17
3.5 ANALYSE DE RISQUES INITIALE.....	17
4. ORGANISATION ET PLANIFICATION DU PROJET	19
4.1 PÉRIODE, CADENCE ET ORGANISATION GÉNÉRALE.....	19
4.2 RITUELS, OUTILS ET ARTEFACTS	19
4.3 UN TICKET “BIEN FORMÉ”: PILOTAGE ET QUALITÉ.....	20
4.4 DEFINITION OF DONE (DOD).....	20
4.5 TRAÇABILITÉ : DU BESOIN AU DÉPLOIEMENT	20
4.6 PLANNING PRÉVISIONNEL	21
4.7 PLANNING RÉEL.....	22
5. COMPRÉHENSION DU PRODUIT (VUE FONCTIONNELLE)	23
5.1 VUE D'ENSEMBLE: L'INTERDÉPENDANCE DES PILIERS MÉTIER	23
5.2 CYCLE DE VIE D'UN RAPPORT : DE L'INSTANTANÉITÉ TERRAIN À L'ARCHIVE PROBANTE .	23
5.3 FONCTIONS ADMINISTRATEUR : GOVERNANCE ET MAINTENANCE PROACTIVE	24
5.4 FONCTION OCR : INTELLIGENCE LOGICIELLE AU SERVICE DES ARCHIVES.....	24
6. CONCEPTION ET ARCHITECTURE (AVEC UML/DIAGRAMMES).....	26
6.1 STACK TECHNIQUE	26
6.2 ARCHITECTURE GLOBALE (DIAGRAMME DE COMPOSANTS).....	26
6.3 UML — CAS D'UTILISATION (USE CASE)	27
6.4 DIAGRAMME D'ACTIVITÉ — CYCLE DE VIE D'UN RAPPORT	28
6.5 DIAGRAMMES DE SÉQUENCE	29
6.6 MODÈLE DE DONNÉES (UML “CLASS DIAGRAM” SIMPLIFIÉ).....	31
6.7 CONCEPTION “EXPLOITATION” — DIAGRAMME DE DÉPLOIEMENT (VPS).....	32
7. SÉCURITÉ	33
7.1 PÉRIMÈTRE DE SÉCURITÉ ET DONNÉES CONCERNÉES	33
7.2 AUTHENTIFICATION ET GESTION DES SESSIONS.....	33
7.3 GESTION ET PROTECTION DES MOTS DE PASSE	34
7.4 AUTORISATION ET CONTRÔLE DES ACCÈS (RBAC).....	34
7.5 SÉCURISATION DES UPLOADS DE FICHIERS	34

7.6 GESTION DES SECRETS ET CONFIGURATION	34
7.7 SÉCURITÉ DES DÉPENDANCES ET LIMITES IDENTIFIÉES.....	35
8. RÉALISATION : DE LA SPÉCIFICATION À LA LIVRAISON.....	36
8.1 MISE EN PLACE INITIALE	36
8.2 IMPLÉMENTATIONS CLÉS	37
8.3 DOCUMENTATION PRODUITE : UNE DÉMARCHE DE TRANSFERT DE CONNAISSANCES ..	40
8.4 VALIDATION MÉTIER : DE LA CONCEPTION À L'USAGE OPÉRATIONNEL	42
9. DÉPLOIEMENT ET MISE EN PRODUCTION (VPS).....	46
9.1 CHOIX TECHNOLOGIQUES ET CONTEXTE D'HÉBERGEMENT	46
9.2 AUTOMATISATION DU DÉPLOIEMENT (CI/CD) ET STRATÉGIE DE VERSIONING	46
9.3 STRATÉGIE DE SAUVEGARDE ET PROCÉDURES DE RESTAURATION	47
9.4 MONITORING PRAGMATIQUE ET EXPLOITATION DES LOGS	47
10. POST-DÉPLOIEMENT : RETOURS, INCIDENTS, ÉVOLUTIONS	48
10.1 CONTEXTE ET CYCLE DE MAINTENANCE ÉVOLUTIVE.....	48
10.2 ÉTUDES DE CAS : ANALYSE DES INCIDENTS ET ÉVOLUTIONS	48
11. MISSIONS ANNEXES	50
11.1 INTÉGRATION D'UN MODULE OCR POUR PDF SCANNÉS	50
11.2 GÉNÉRATION AUTOMATIQUE D'UNE PAGE DE GARDE POUR LES DOCUMENTS PDF....	51
12. ANALYSE REFLEXIVE DES COMPETENCES	52
12.1 BLOC 1 : IDENTIFIER, MODELISER ET RESOUDRE DES PROBLEMES INFORMATIQUES ...	52
12.2 BLOC 2 : CONCEVOIR, DEVELOPPER, METTRE AU POINT ET CONDUIRE DES PROJETS D'APPLICATION INFORMATIQUE	52
12.3 BLOC 3 : CONCEVOIR, PILOTER, ORGANISER, OPTIMISER ET GERER DES SYSTEMES D'INFORMATION	52
12.4 BLOC 4 : CONCEVOIR, METTRE EN ŒUVRE, SECURISER ET MAINTENIR EN CONDITION OPERATIONNELLE DES INFRASTRUCTURES RESEAUX	53
12.5 BLOC 5 : ELABORER, CONCEVOIR ET MENER DES STRATEGIES D'INTEGRATION ET D'EVOLUTION DES INSTALLATIONS MATERIELLES ET LOGICIELLES	53
12.6 BLOC 6 : DEFINIR, PLANIFIER, ORGANISER ET MANAGER UN PROJET D'INGENIERIE INNOVANT ET COLLABORATIF	53
12.7 BLOC 7 : QUESTIONNER, ANALYSER ET ADOPTER UNE DEMARCHE SYSTEMIQUE OUVERTE ET RESPONSABLE.....	53
13. CONCLUSION.....	55
13.1 SYNTHÈSE DES RÉSULTATS	55
13.2 ANALYSE CRITIQUE	55

13.3 ESTIMATION DU GAIN FINANCIER ET DE LA VALEUR APPORTÉE.....	56
13.4 PERSPECTIVES	57
LISTE DES FIGURES	57
FIGURES RAPPORT:.....	57
FIGURES APPLICATION:	58
RESSOURCES EN LIGNE	58
BIBLIOGRAPHIE.....	58
ANNEXES DES CAPTURES D’ÉCRAN DE L’APPLICATION	59

Abréviations et Glossaire

Abréviations

- **API** : Application Programming Interface (Interface de Programmation d'Applications).
- **CI/CD** : Continuous Integration / Continuous Delivery (Intégration Continue / Déploiement Continu).
- **JWT** : JSON Web Token (Jeton Web JSON utilisé pour l'authentification).
- **OCR** : Optical Character Recognition (Reconnaissance Optique de Caractères).
- **PR** : Pull Request (Demande de tirage pour la revue de code).
- **RBAC** : Role-Based Access Control (Contrôle d'Accès Basé sur les Rôles).
- **REST** : Representational State Transfer (Style d'architecture logicielle pour les services web).
- **SPA** : Single Page Application (Application web à page unique).
- **TLS** : Transport Layer Security (Protocole de sécurisation des échanges réseaux).
- **VPS** : Virtual Private Server (Serveur Privé Virtuel utilisé pour l'hébergement).

Glossaire

- **Archivage probant** : Conservation d'un document garantissant son intégrité et sa valeur légale dans le temps.
- **GridFS** : Système de stockage de MongoDB utilisé pour fragmenter et conserver les fichiers volumineux (photos HD, PDF).
- **Backlog** : Ensemble ordonné de tâches à réaliser pour délivrer le produit.
- **Nginx** : Serveur web utilisé ici comme Reverse Proxy pour gérer les flux HTTPS et rediriger les requêtes vers l'API.
- **Tesseract** : Moteur de reconnaissance optique de caractères (OCR) open-source utilisé pour traiter les documents scannés.
- **Design Pattern Strategy** (Patron de Stratégie) : Modèle de conception logicielle permettant de définir une famille d'algorithmes, de les encapsuler et de les rendre interchangeables.

1. Introduction

1.1 Contexte general

Ce stage s'inscrit dans le développement de FL copro, une application métier destinée à un gestionnaire de copropriétés. L'objectif principal est de faciliter la production de rapports (sinistres, visites, suivi de travaux...) en remplaçant un processus historiquement réalisé avec des documents bureautiques, souvent source de perte de temps et de problèmes de mise en page.

Au-delà de la production de documents, l'application manipule des données potentiellement sensibles : informations liées à des résidents, adresses d'immeubles, ainsi que des documents comptables. Ces contraintes impliquent un fort impact sur la sécurité et sur l'exploitation.

1.2 Problématique

Comment concevoir, développer, sécuriser et mettre en production une application documentaire orientée terrain, tout en intégrant une démarche itérative avec validation métier et des corrections post-déploiement ?

1.3 Objectifs du stage

L'objectif principal du stage était de concevoir et de développer une application web métier destinée au service Syndic, permettant de centraliser la création, la gestion et l'archivage des rapports de copropriété. Cette application devait remplacer des méthodes principalement bureautiques par un outil structuré, sécurisé et directement exploitable en conditions réelles.

Le stage visait également à aboutir à une solution pleinement opérationnelle, utilisée par le service en production. Cela impliquait notamment la génération automatisée de rapports PDF conformes aux attentes métier, ainsi qu'une amélioration du temps et du confort de production des rapports par rapport aux pratiques existantes.

Pour garantir la faisabilité du projet dans le cadre du stage, le périmètre a été volontairement limité à un socle fonctionnel essentiel, tout en laissant une marge d'évolution. Les choix technologiques ont été guidés par des solutions éprouvées et adaptées au contexte du projet, permettant à la fois une montée en compétence et une mise en œuvre efficace.

1.4 Périmètre et contraintes (In / Out)

Le stage a porté sur le développement complet d'une application web full-stack destinée à un usage interne. Celle-ci permet la création, l'édition, la finalisation et l'export de rapports de copropriété, ainsi que la gestion des utilisateurs, des droits d'accès et des documents associés.

L'application est accessible depuis un navigateur web, aussi bien sur ordinateur que sur mobile, et repose sur un backend assurant la logique métier, la sécurité des accès et la génération des documents PDF. Le projet inclut également le déploiement de la solution sur un serveur VPS afin de permettre une utilisation en conditions réelles.

Certaines fonctionnalités ont volontairement été exclues du périmètre, notamment l'intégration avec des logiciels tiers existants ou la mise en place d'une infrastructure complexe de type haute disponibilité. L'objectif était de se concentrer sur un outil fiable, adapté aux besoins immédiats du service, et compatible avec la durée du stage.

Les principales contraintes rencontrées concernent la gestion de données sensibles, les ressources limitées du serveur d'hébergement et la nécessité de proposer une interface simple et utilisable sur le terrain. Les choix fonctionnels et techniques ont par ailleurs fait l'objet de validations régulières avec le tuteur métier afin de rester aligné avec les besoins réels du service Syndic.



2. Présentation de l'entreprise

2.1 Présentation Générale

FL IMMO est une entreprise spécialisée dans les métiers de l'immobilier, implantée à Belfort, et structurée autour de plusieurs activités complémentaires couvrant l'ensemble du cycle de vie d'un bien immobilier. Son positionnement repose sur une approche intégrée, combinant transaction immobilière, gestion locative et syndic de copropriété, permettant d'accompagner aussi bien les propriétaires que les copropriétaires sur le long terme.

L'activité historique de FL IMMO est la transaction immobilière, qui constitue aujourd'hui le principal levier de chiffre d'affaires de l'entreprise. Cette activité génère environ 500 000 € de chiffre d'affaires annuel, essentiellement issus de la vente de biens immobiliers. Elle repose sur une forte expertise métier, un ancrage territorial marqué et une bonne connaissance du marché local, garantissant un volume d'affaires régulier.

En parallèle, FL IMMO développe une activité de gestion locative, destinée à accompagner les propriétaires dans l'administration quotidienne de leurs biens (relation locataire, suivi des loyers, gestion administrative). Cette activité a généré environ 150 000 € de chiffre d'affaires, avec une progression significative sur l'année en cours, atteignant 200 000 €, ce qui traduit une montée en charge rapide et une demande croissante pour ce type de services. La gestion locative constitue ainsi un axe stratégique pour l'entreprise, offrant une source de revenus récurrents, moins dépendante des fluctuations du marché de la transaction.

La troisième activité de FL IMMO est le syndic de copropriété, lancé plus récemment. Bien que jeune, ce pôle a déjà permis de générer 58 000 € de bénéfices en six mois, révélant un fort potentiel de croissance. Le développement du syndic s'inscrit dans une logique de structuration progressive, tant sur le plan organisationnel que sur celui des outils métiers, à mesure de l'augmentation du nombre de copropriétés gérées.

À travers la consolidation de ces trois pôles, FL IMMO poursuit une trajectoire de croissance maîtrisée. L'entreprise vise un chiffre d'affaires supérieur à 1 million d'euros à l'horizon 2027, porté par la montée en puissance du syndic, le développement continu de la gestion locative et le maintien d'un niveau élevé d'activité en transaction. Cette stratégie repose notamment sur l'optimisation des processus internes et sur la digitalisation des outils métiers, démarche dans laquelle s'inscrit pleinement le projet présenté dans ce rapport de stage.

2.2 Organisation / services

FL IMMO est organisée autour de trois services principaux, correspondant à ses trois activités métiers :

Transaction immobilière : Ce service est en charge de la vente de biens immobiliers. Il accompagne les clients tout au long du processus de transaction, depuis l'estimation du bien jusqu'à la signature de l'acte de vente, en assurant la mise en relation entre vendeurs et acquéreurs.

Arnaud Michel – Février 2026

Gestion locative : Le service de gestion locative assure la gestion administrative, financière et opérationnelle des biens mis en location. Il comprend notamment le suivi des loyers, la relation avec les locataires, la gestion des contrats de location et la coordination des interventions techniques courantes.

Syndic de copropriété : Le service Syndic prend en charge la gestion administrative, juridique, technique et financière des copropriétés. Il agit pour le compte du syndicat des copropriétaires et assure la bonne conservation du patrimoine immobilier, ainsi que le respect des obligations réglementaires.

Cette organisation permet à FL IMMO de mutualiser certaines compétences tout en conservant une spécialisation claire par métier.



2.2.1 Organigramme de l'entreprise:

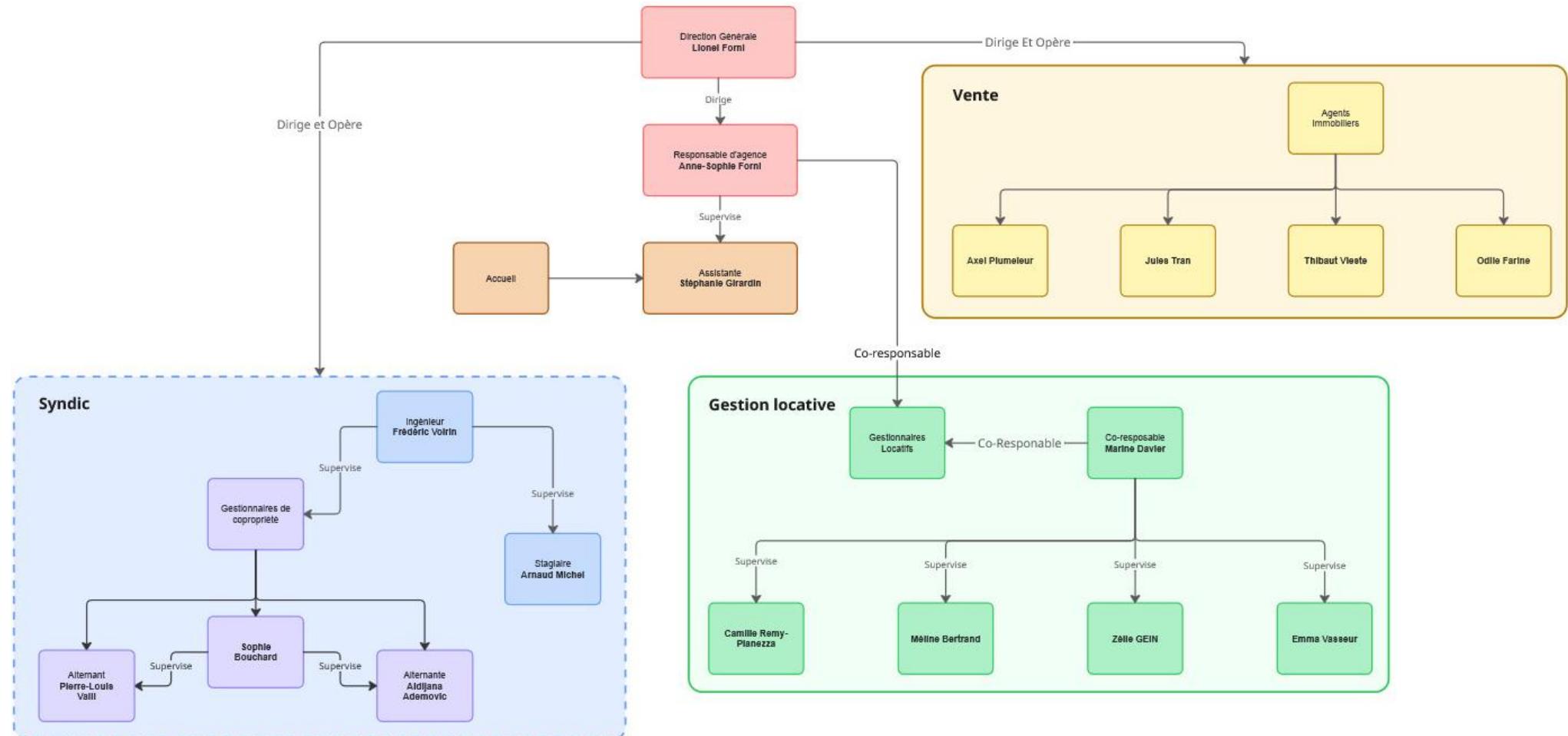


FIG. 1: ORGANIGRAMME DE L'ENTREPRISE

2.3 Service d'accueil : Syndic

Le service Syndic assure la gestion administrative, juridique, technique et financière des copropriétés pour le compte du syndicat des copropriétaires. Dans le cadre de FL Immo, il s'agit d'une activité récente (lancée en 2025), encore en phase de structuration des outils et des méthodes.

2.3.1 Missions principaux

Les missions du syndic s'articulent autour de quatre grands axes qui, dans la pratique quotidienne, sont étroitement imbriqués :

La gestion administrative et juridique de la copropriété.

Le syndic veille à l'application du règlement de copropriété, en s'assurant du respect des droits et obligations de chacun, des règles d'usage des parties communes ainsi que des procédures encadrant les travaux. Il organise les Assemblées Générales, depuis leur préparation (convocations, collecte et analyse des devis, élaboration de l'ordre du jour) jusqu'à leur tenue, puis assure la rédaction, la validation et la diffusion des procès-verbaux. Il est également responsable de la tenue et de la conservation des archives de la copropriété, incluant les documents légaux, les contrats, l'historique des travaux et la gestion des sinistres. À cela s'ajoute le respect des obligations réglementaires, notamment l'immatriculation de la copropriété lorsque celle-ci est requise, ainsi que la mise à jour continue des données administratives (lots, copropriétaires, tantièmes, coordonnées, mandats).

La gestion financière et comptable.

Le syndic élabore le budget prévisionnel, le soumet au vote de l'Assemblée Générale et en assure le suivi tout au long de l'exercice. Il tient la comptabilité de la copropriété en distinguant les charges courantes, les dépenses liées aux travaux, les provisions et les régularisations. Il gère les comptes bancaires dédiés de la copropriété, comprenant à minima un compte courant et, le cas échéant, un compte spécifique pour les travaux. Il procède à l'émission des appels de fonds, qu'ils soient périodiques ou exceptionnels, et assure le suivi du recouvrement, incluant les relances, la mise en place d'échéanciers et, si nécessaire, la transmission des dossiers en contentieux. Enfin, il établit les états financiers réglementaires permettant aux copropriétaires d'avoir une vision claire et transparente de la situation financière de l'immeuble.

La gestion technique et l'exploitation de l'immeuble.

Le syndic assure le suivi de l'entretien courant des parties communes et tient à jour le carnet d'entretien. Il gère les relations avec les prestataires (contrats d'entretien, entreprises de travaux), planifie les interventions, contrôle leur exécution et réceptionne les travaux réalisés. Lorsque des travaux sont votés en Assemblée Générale, le syndic en assure le pilotage, soit en tant que maître d'ouvrage, soit en assistance, selon la nature et l'ampleur des opérations. Il prend également en

charge la gestion des sinistres, depuis la déclaration jusqu'à leur clôture, en coordonnant les différents intervenants (assurances, experts, entreprises).

La relation avec les copropriétaires et de la communication.

Le syndic informe régulièrement les copropriétaires des décisions prises, des travaux en cours, des incidents éventuels et des interventions programmées. Il traite les demandes et réclamations, assure un rôle de conseil lorsque cela est nécessaire et veille au suivi des engagements pris par la copropriété.

L'ensemble de ces missions génère un volume important de documents, de décisions et d'échanges formalisés. La qualité du service rendu par le syndic repose donc en grande partie sur la rigueur de la gestion documentaire, qui doit être complète, datée, facilement accessible et conforme aux exigences réglementaires.

2.3.2 Processus métiers typiques, du terrain au dossier

Même si chaque copropriété fonctionne selon ses propres contraintes, certaines opérations reviennent de façon régulière dans l'activité du syndic.

Les visites et constats sur site en font partie. Qu'il s'agisse d'une visite d'immeuble, d'un sinistre, d'une réception de travaux ou d'un relevé d'index, ces interventions suivent généralement le même schéma. Elles sont préparées à partir des informations existantes (historique, documents techniques, contrats), puis donnent lieu à une intervention sur place avec observations, relevés, prises de photos et échanges avec les occupants ou les intervenants. Ces éléments sont ensuite formalisés dans un compte rendu structuré, accompagné des pièces utiles, classé dans le dossier de la copropriété et diffusé aux personnes concernées pour assurer le suivi.

L'organisation des Assemblées Générales constitue un autre processus central. Elle comprend la préparation de l'ordre du jour et des résolutions, la collecte des devis et la vérification des quotes-parts, puis l'envoi des convocations dans les délais légaux. Lors de l'Assemblée, le syndic organise les votes et consigne les décisions. À l'issue de la réunion, le procès-verbal est rédigé et diffusé, et les décisions votées sont mises en œuvre, notamment en matière de travaux, de contrats ou d'appels de fonds.

Enfin, le cycle comptable et financier rythme la vie de la copropriété tout au long de l'année. Il débute par l'élaboration du budget et des provisions, se poursuit par l'émission des appels de fonds et le suivi des paiements, et s'achève par la production des états financiers et des justificatifs nécessaires, tant pour les Assemblées Générales que pour l'information des copropriétaires.

Ces processus, bien que distincts, s'enchaînent en pratique de manière continue et reposent sur une organisation fluide, une bonne circulation de l'information et une documentation fiable.

2.3.3 Types de documents manipulés

L'activité du syndic repose sur la production et l'exploitation d'un volume important de documents, provenant de sources multiples et présentant des formats et des niveaux de qualité très variables. Ces documents couvrent aussi bien les opérations de terrain que les aspects juridiques, financiers et techniques de la copropriété.

Une partie importante de la documentation est issue des interventions sur site : rapports de visite, constats, comptes rendus d'intervention, réceptions de travaux, relevés d'index ou encore listes de réserves. À cela s'ajoutent les documents juridiques structurants de la copropriété, tels que le règlement de copropriété et ses modificatifs, les contrats et mandats, ainsi que l'ensemble des documents liés aux Assemblées Générales (convocations, feuilles de présence, procès-verbaux).

Le syndic gère également une documentation financière dense, comprenant le budget prévisionnel, les appels de fonds, les annexes comptables, les états de dépenses, les relevés bancaires et les documents fournisseurs. Sur le plan technique, il traite de nombreux devis, factures, attestations, diagnostics, notices, plans et dossiers de travaux. Enfin, une part significative des échanges se fait sous forme de correspondances, qu'il s'agisse de courriers, d'e-mails, de mises en demeure ou de déclarations de sinistres.

Les formats les plus couramment rencontrés sont le PDF et les images, en particulier les photographies prises sur mobile. Une partie des documents est transmise sous forme scannée ou photographiée, avec une qualité parfois inégale (inclinaison, faible luminosité, documents multipages), ce qui complique leur classement, leur indexation et leur recherche dans le temps.

2.3.4 Enjeux

Ces activités métier imposent des contraintes non fonctionnelles fortes, qui ont directement structuré le cahier des charges de l'application développée.

La première contrainte concerne la fiabilité et la traçabilité des informations produites. Un rapport de visite doit être exact, daté et clairement attribuable à son auteur, tout en étant rattaché sans ambiguïté à la copropriété concernée. Les pièces jointes associées (photos, documents, devis) doivent rester durablement liées au bon dossier, toute perte ou confusion pouvant avoir des conséquences importantes, notamment en cas de sinistre, de litige ou de réception de travaux. Il existe également un besoin fort de cohérence documentaire : des structures de rapports homogènes, des champs clés systématiquement renseignés et des mécanismes limitant les oubliés, jouant un rôle de checklist implicite pour les gestionnaires.

Une deuxième contrainte majeure porte sur les délais et la réactivité. Les comptes rendus doivent être produits rapidement après une visite, en particulier dans les situations urgentes (sinistres, interventions techniques, receptions), afin de déclencher les actions nécessaires et d'éviter toute escalade. Par ailleurs, certaines obligations réglementaires, telles que les convocations aux Assemblées Générales, la diffusion des procès-verbaux ou l'émission des appels de fonds, s'inscrivent dans des délais incompressibles. L'outil doit donc limiter au maximum les frictions opérationnelles, en proposant une saisie rapide, une gestion simple des images et un export fiable et immédiat des documents.

Enfin, la conformité et la confidentialité constituent un axe structurant. Les dossiers de copropriété contiennent des données personnelles et des informations potentiellement sensibles, allant des identités et coordonnées aux éléments financiers ou aux situations de sinistre. L'application devait donc intégrer des mécanismes de contrôle d'accès basés sur les profils et les permissions, garantir l'isolation des dossiers, assurer une journalisation minimale des actions et prévoir des dispositifs de sauvegarde et de restauration. Au-delà des aspects techniques, la conformité est également opérationnelle : le syndic doit être en mesure de retrouver rapidement un document « qui fait foi » en cas de contrôle, de contestation ou de procédure.

En synthèse, le service Syndic est un métier dans lequel l'information est essentiellement documentaire. L'application développée s'inscrit dans cette réalité en visant à fiabiliser la production des rapports, à standardiser les contenus, à sécuriser les accès et à réduire le délai entre une intervention sur le terrain et la mise à disposition d'un document exploitable.



3. Cadrage initial et cahier des charges

3.1 Origine du besoin et enjeux

Le besoin à l'origine du projet est issu d'un constat terrain au sein du service Syndic. Les rapports réalisés à la suite des visites (visites d'immeuble, sinistres, réceptions de travaux, relevés d'index, etc.) étaient systématiquement produits sous Word. Cette pratique, bien que répandue, présentait plusieurs limites opérationnelles. Elle entraînait une perte de temps significative liée à la mise en forme des documents (gestion des photos, styles, sauts de page), au détriment du contenu lui-même. Les rapports étaient par ailleurs hétérogènes, tant dans leur structure que dans leur niveau de détail, selon les collaborateurs. La gestion des images était lourde (insertion, alignement, poids des fichiers) et l'archivage peu fiable, avec des pièces jointes dispersées et des versions multiples. À cela s'ajoutait l'impossibilité de saisir directement sur le terrain, ainsi qu'un risque de perte d'information et de défaut de traçabilité, alors même que certains documents produits ont une valeur probante, notamment en matière de sinistres ou de réserves.

Dans ce contexte, le sujet de stage formulé par l'entreprise était le suivant : concevoir et réaliser un logiciel utilisable sur mobile et ordinateur permettant aux administrateurs de biens de produire des rapports de visite. Le projet visait ainsi à améliorer à la fois la productivité des collaborateurs et la qualité globale de la documentation produite.

Le cadre de pilotage imposé a structuré la démarche, avec un suivi hebdomadaire sous forme de revues de projet avec le tuteur de stage, complété par une revue mensuelle avec la direction afin de valider les orientations prises. Les étapes attendues comprenaient la définition du cahier des charges, la phase de développement, les tests en conditions réelles sur site, la validation par les utilisateurs, puis l'intégration d'améliorations et d'évolutions.

Dès le départ, l'objectif de fin de stage était clairement identifié : disposer d'un logiciel utilisable aussi bien sur smartphone que sur ordinateur, permettant une saisie rapide sur le terrain et capable d'envoyer automatiquement les rapports vers un serveur d'archivage.

Les enjeux clés identifiés concernaient principalement la réduction du temps de production des rapports, l'amélioration de la qualité et de la standardisation des contenus, la traçabilité des actions (création, modification, finalisation), la centralisation et la facilité de recherche des documents, ainsi que la sécurité des données et des documents traités, notamment au regard des données personnelles.

3.2 Utilisateurs cibles et cas d'usage

3.2.1 Utilisateurs cibles

L'application s'adresse à deux profils d'utilisateurs distincts. Le gestionnaire constitue l'utilisateur principal : il crée les rapports, saisit les observations, ajoute les pièces jointes, puis finalise et exporte les documents. L'administrateur intervient de manière plus ponctuelle et assure la gestion des utilisateurs, des rôles et des permissions, ainsi que le paramétrage global de l'application.

3.2.2 Cas d'usage principaux

Les principaux cas d'usage couvrent l'ensemble du cycle de vie d'un rapport. L'utilisateur doit pouvoir créer un rapport à partir d'une copropriété et sélectionner le type de visite concerné. Il rédige ensuite le contenu à partir d'une structure standardisée, organisée en sections, afin de guider la saisie et d'éviter les oubliers. L'ajout de pièces jointes, notamment des photos prises depuis un smartphone ou des documents PDF, doit être simple et contrôlé. Une fois le rapport finalisé, l'utilisateur génère un PDF exploitable, puis procède à son archivage ou à son partage, avec pour objectif cible un envoi automatique vers un serveur.

Une contrainte transverse s'applique à l'ensemble de ces parcours : ils doivent rester pleinement utilisables sur mobile (écran réduit, saisie rapide, connectivité variable) tout en offrant un confort d'édition et de relecture sur ordinateur.

3.3 Exigences fonctionnelles

Les exigences fonctionnelles définissent les capacités opérationnelles que l'application doit offrir afin de répondre concrètement aux besoins du service Syndic. Elles ont été formulées à partir des observations terrain, des échanges avec le tuteur métier et des usages quotidiens des gestionnaires de copropriété. L'objectif n'est pas uniquement de couvrir l'ensemble des fonctionnalités attendues, mais également de garantir leur pertinence et leur testabilité en conditions réelles.

L'application nécessite en premier lieu permettre la gestion complète des rapports de copropriété. Un utilisateur doit pouvoir créer un rapport en le rattachant à une copropriété existante et en sélectionnant un type de visite adapté au contexte métier (visite d'immeuble, sinistre, réception de travaux, relevé d'index). La saisie du contenu doit être guidée par une structure standardisée, composée de sections et de champs prédéfinis, afin d'harmoniser les pratiques et de limiter les oubliers lors des interventions sur site.

La gestion des pièces jointes constitue une exigence centrale. L'utilisateur doit pouvoir ajouter facilement des photographies prises depuis un terminal mobile ainsi que des documents PDF, tout en conservant un lien explicite entre ces fichiers et le rapport concerné. Ces pièces jointes se doivent d'être visualisables dans l'interface et intégrées automatiquement lors de la génération du document final.

Une fois le rapport rédigé, l'application doit proposer un mécanisme de finalisation. Cette action transforme le rapport en un document figé, non modifiable, garantissant son intégrité et sa valeur d'archive. À partir de cet état, l'utilisateur doit pouvoir générer un fichier PDF standardisé, prêt à être transmis aux copropriétaires ou archivé dans les dossiers de la copropriété.

Enfin, l'application doit intégrer des fonctionnalités d'administration permettant la gestion des utilisateurs, des rôles et des permissions. Un administrateur doit pouvoir créer ou désactiver des comptes, attribuer des droits et superviser l'utilisation globale de la plateforme. L'ensemble de ces

fonctionnalités doit rester pleinement exploitable aussi bien sur ordinateur que sur smartphone, afin de répondre aux contraintes de mobilité propres aux visites de terrain.

3.4 Exigences non fonctionnelles (RNF)

Les exigences non fonctionnelles ont joué un rôle structurant dans le cadrage du projet, compte tenu de la sensibilité des données traitées et des contraintes d'usage liées à l'activité du syndic. Elles concernent principalement la sécurité, la fiabilité, la performance, l'ergonomie et l'exploitation de l'application.

La sécurité constitue l'exigence transversale prioritaire. L'accès à l'application est strictement réservé aux utilisateurs authentifiés, avec un contrôle des droits basé sur les rôles afin de garantir la confidentialité des données et des documents. Les informations personnelles, les rapports et les pièces jointes doivent être protégés contre tout accès non autorisé, et les actions sensibles réalisées exclusivement via des mécanismes sécurisés côté serveur.

La fiabilité du système est également essentielle. L'application doit garantir la conservation durable des rapports et des fichiers associés, sans risque de perte ou de corruption des données. Les mécanismes de sauvegarde, de verrouillage des rapports finalisés et de gestion des erreurs doivent permettre une exploitation sereine de l'outil en production.

La performance doit rester compatible avec un usage terrain. Les temps de réponse de l'interface doivent être suffisants pour permettre une saisie fluide, y compris sur des connexions réseau variables. Les opérations potentiellement coûteuses, telles que la génération de PDF ou le traitement de fichiers volumineux, nécessitent d'être conçus de manière à ne pas dégrader l'expérience utilisateur.

L'ergonomie constitue une exigence clé, en particulier sur mobile. L'interface se doit d'être lisible sur écran réduit, proposer des interactions simples et limiter la charge cognitive lors de la saisie des informations. La standardisation des structures de rapports doit faciliter la prise en main par des utilisateurs non techniques.

Enfin, l'application doit rester exploitable et maintenable. Le code doit être structuré, documenté et conforme à des conventions claires afin de permettre des évolutions futures. Des mécanismes minimaux de journalisation et de supervision doivent être présents afin de faciliter le diagnostic en cas d'incident.

3.5 Analyse de risques initiale

Dès la phase de cadrage, une analyse de risques a été menée afin d'anticiper les principaux points de vigilance susceptibles d'affecter le bon déroulement du projet ou la qualité du produit final. Cette démarche visait à réduire les incertitudes techniques et organisationnelles avant la mise en production.

Un premier risque identifié concernait la volumétrie des données, en particulier la gestion des photographies et des documents PDF ; un stockage mal maîtrisé pouvant entraîner une saturation rapide de l'espace disque ou une dégradation des performances. Ce risque a conduit à intégrer très tôt une solution de stockage adaptée et à définir des limites strictes sur la taille et les types de fichiers acceptés.

La génération des documents PDF constituait un second risque majeur. Un rendu instable ou incohérent selon les terminaux aurait compromis l'usage métier de l'application. Pour limiter ce risque, le choix a été fait de centraliser la génération des PDF côté serveur, garantissant une homogénéité du rendu indépendamment du support utilisé pour la saisie.

L'usage en mobilité représente également un facteur de risque important. Les gestionnaires étant amenés à utiliser l'application sur le terrain, parfois avec une connectivité limitée, il existe un risque de perte de données ou de mauvaise expérience utilisateur. Ce point a justifié la mise en place de sauvegardes fréquentes et d'une interface tolérante aux interruptions.

Enfin, les risques liés à la sécurité des accès et des données ont été considérés comme critiques. Une faille d'authentification ou de contrôle des droits aurait pu exposer des informations sensibles. Ce risque a orienté les choix d'architecture vers des mécanismes éprouvés d'authentification, de gestion des rôles et de validation côté serveur, intégrés dès les premières itérations du développement.

Cette analyse de risques initiale a permis de guider les priorités du projet et d'adopter une démarche pragmatique, en intégrant des tests en conditions réelles tout au long du développement afin de détecter et corriger rapidement les points sensibles avant la phase de déploiement.



4. Organisation et planification du projet

4.1 Période, cadence et organisation générale

Le projet s'est déroulé sur une période allant du 1er septembre 2025 au 7 février 2026. Afin de structurer l'avancement et de favoriser des livraisons régulières, le développement a été organisé en sprints de trois semaines. Ce découpage a permis de maintenir un rythme soutenu tout en laissant suffisamment de temps pour concevoir, développer, tester et valider les fonctionnalités.

Le développement a été réalisé en autonomie, avec une logique d'amélioration incrémentale. À la fin de chaque sprint, une validation métier était systématiquement effectuée afin de confronter les choix techniques aux usages réels et d'ajuster les priorités si nécessaire.

4.2 Rituels, outils et artefacts

Afin de garantir la traçabilité des décisions et d'objectiver l'avancement du projet, plusieurs outils et rituels ont été mis en place dès le démarrage.

Le suivi des travaux a été assuré à l'aide de [Click Up](#), utilisé comme outil central de pilotage. Il a permis de maintenir un backlog produit regroupant les fonctionnalités et les corrections, de constituer les sprint backlogs, et de formaliser chaque ticket avec des critères d'acceptation clairs. Cet outillage a joué un rôle clé dans la priorisation et la visibilité de l'avancement.

La gestion du code source s'est appuyée sur [Git](#), avec une organisation en branches dédiées par fonctionnalité. Chaque évolution faisait l'objet d'une Pull Request, même en contexte de développement en autonomie, afin de conserver une discipline de revue et une historisation claire. Les livraisons étaient identifiées par des tags et des releases, facilitant le suivi des versions déployées.

Plusieurs rituels ont été réellement pratiqués au fil du projet. Une revue hebdomadaire avec le tuteur de stage permettait de présenter les avancées, de démontrer les fonctionnalités développées et de prendre des décisions d'orientation. Des points rapides, assimilables à des daily meetings, avaient lieu plusieurs fois par semaine selon les contraintes d'agenda, afin de lever rapidement les blocages. En fin de sprint, une review de sprint permettait de valider les livrables, tant sur le plan fonctionnel que technique. Enfin, une rétrospective était menée afin d'identifier une à trois pistes d'amélioration pour le sprint suivant, par exemple sur l'expérience utilisateur ou sur l'interface.

Ces pratiques ont donné lieu à plusieurs artefacts concrets : un backlog ClickUp priorisé, une documentation technique et fonctionnelle rédigée sous [Writerside](#), des releases post-production intégrant correctifs et améliorations, ainsi que des éléments d'exploitation tels que les procédures de déploiement, de sauvegarde et la configuration Nginx.

4.3 Un ticket “bien formé”: pilotage et qualité

L’expérience acquise au cours du projet a rapidement montré qu’un ticket insuffisamment cadré générait du rework, en particulier lors des validations métier. Pour limiter ce risque, un format de ticket « bien formé » a été progressivement standardisé.

Chaque ticket comportait un contexte explicitant l’origine du besoin et l’irritant observé, suivi d’une description claire du besoin utilisateur. Des critères d’acceptation testables permettaient de définir précisément les conditions de réussite. Le périmètre était complété par des points d’attention portant notamment sur la sécurité, la volumétrie ou la performance.

Ce formalisme s’est révélé particulièrement utile pour des sujets sensibles tels que la gestion des uploads (types de fichiers autorisés, taille maximale, gestion des erreurs, sécurité avec fichier malveillants), la finalisation des rapports (statuts, verrouillage), la génération de PDF (stabilité du rendu) ou encore l’ergonomie mobile.

4.4 Definition of Done (DoD)

Afin d’éviter les situations de « faux terminés », une Definition of Done a été définie et appliquée tout au long du projet. Un ticket était considéré comme terminé uniquement lorsque l’ensemble des critères suivants était respecté.

Sur le plan fonctionnel, les critères d’acceptation devaient être satisfais et les cas d’erreur usuels correctement gérés, tant côté interface utilisateur (messages explicites) que côté API (codes HTTP cohérents). Sur le plan de la sécurité et de la conformité, les routes sensibles devaient être protégées par authentification, les permissions vérifiées côté serveur selon le modèle RBAC et non contournables, et les uploads validés côté serveur en termes de type et de taille.

Des exigences de qualité et de maintenabilité complétaient cette définition, avec un code lisible et conforme aux conventions du projet, l’absence de logs de débogage inutiles en production et une documentation mise à jour lorsque nécessaire. Enfin, chaque ticket devait faire l’objet d’un test, automatisé lorsque cela était possible, ou à défaut d’un test manuel documenté, et être démontrable lors d’une revue hebdomadaire ou de fin de sprint.

4.5 Traçabilité : du besoin au déploiement

Un flux de traçabilité clair a été recherché tout au long du projet, et appliqué autant que possible. Un besoin était d’abord identifié à partir d’un retour terrain, d’une demande de la direction ou d’un échange avec le tuteur. Il donnait lieu à la création d’un ticket ClickUp formalisé avec des critères d’acceptation, puis à la création d’une branche Git associée. L’implémentation était réalisée via des commits réguliers, suivis d’une Pull Request et de vérifications techniques (lint, build, tests lorsque disponibles). Après fusion, une release était créée, puis déployée sur le VPS. La validation finale s’effectuait en conditions réelles, avec, le cas échéant, la création de tickets correctifs.

Ce flux a permis de relier chaque évolution visible, par exemple une amélioration de l'ergonomie mobile ou une correction liée aux uploads, à des preuves concrètes (ticket, commits, release), facilitant ainsi la justification des choix techniques.

4.6 Planning prévisionnel

Un planning prévisionnel a été établi en début de projet afin de structurer les premières itérations. Celui-ci reposait sur des hypothèses claires : des sprints de trois semaines, une livraison incrémentale et une priorité donnée aux parcours cœur (création de rapports, gestion des pièces jointes, génération de PDF), avant l'intégration de fonctionnalités plus avancées telles que l'OCR.

Ce planning avait vocation à servir de référence, tout en restant adaptable aux retours terrain et aux ajustements nécessaires en cours de projet.

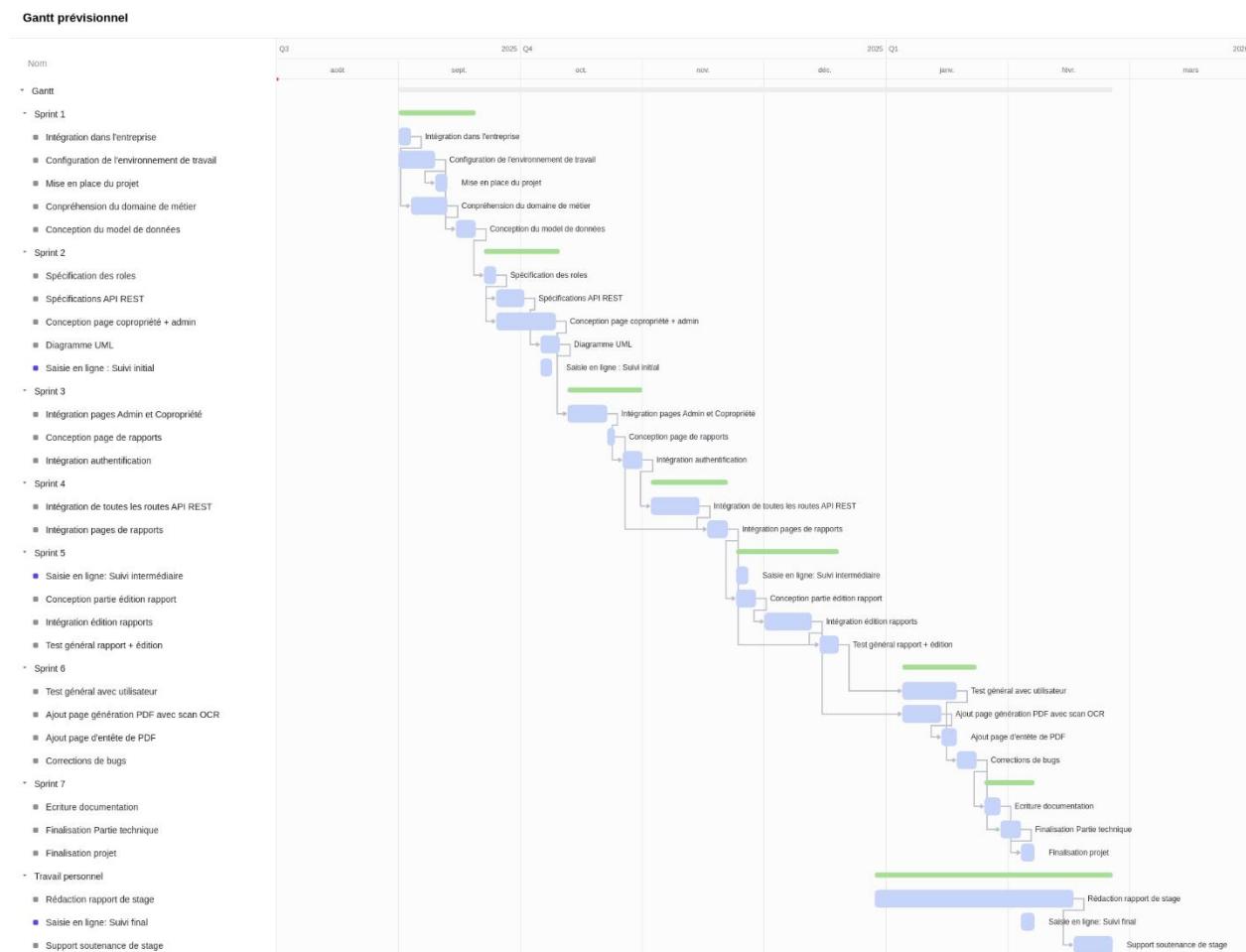


FIG. 2: GANTT PRÉVISIONNEL

4.7 Planning Réel

Le planning réel reflète l'évolution effective du projet, intégrant les ajustements opérés en fonction des retours utilisateurs, des contraintes techniques rencontrées et des priorités redéfinies au fil des sprints. La comparaison entre le prévisionnel et le réalisé permet de mettre en évidence la capacité d'adaptation du pilotage et l'amélioration progressive de l'estimation des charge.

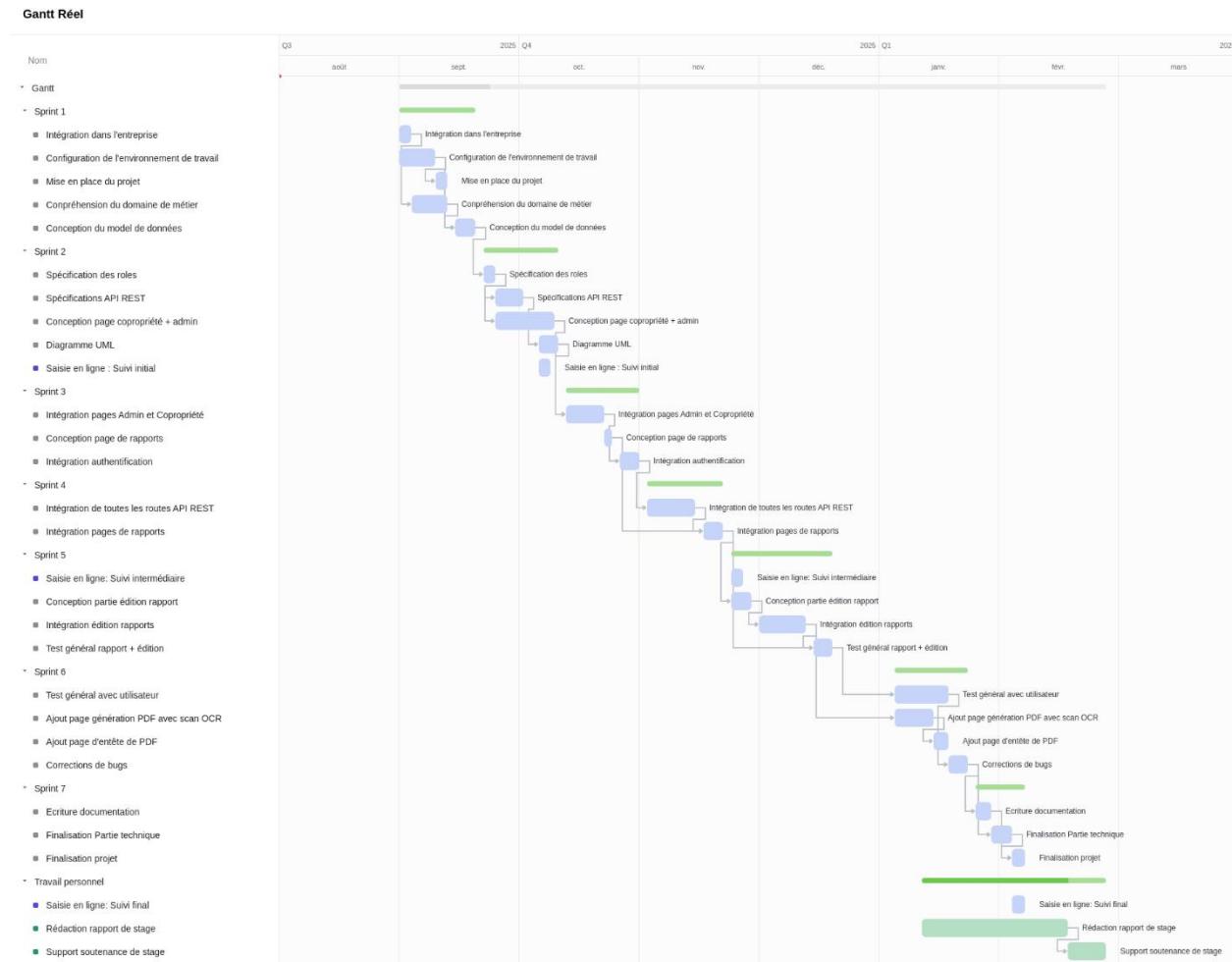


FIG. 3: GANTT RÉEL

5. Compréhension du produit (vue fonctionnelle)

La compréhension de l'application FL copro nécessite d'appréhender le logiciel non pas comme une simple interface de saisie, mais comme un écosystème numérique complet conçu pour structurer l'activité du service Syndic de FL IMMO. L'objectif fondamental du produit est de rationaliser la production documentaire tout en garantissant une cohérence stricte entre les données immobilières, les interventions de terrain et la gestion des habilitations utilisateurs.

5.1 Vue d'ensemble: L'interdépendance des piliers métier

Le fonctionnement global du logiciel repose sur une synergie constante entre trois entités fondamentales : les copropriétés, les rapports et les utilisateurs.

Le module de gestion des copropriétés constitue le socle de données de l'application, agissant comme un inventaire centralisé où chaque immeuble est identifié par un code unique, une fiche technique exhaustive incluant ses adresses multiples, sa date de construction et une galerie visuelle pour faciliter son identification rapide par les collaborateurs. Cette base de données immobilière est intrinsèquement liée au module des rapports, qui représente le volet opérationnel du système. À chaque intervention physique d'un gestionnaire correspond la génération d'une entité « rapport » dont la structure (visite, sinistre, relevé de compteurs ou réception de travaux) s'adapte dynamiquement au besoin métier.

Enfin, le troisième pilier est constitué par le système de gestion des identités et des accès, qui définit les frontières d'intervention de chaque collaborateur. Ce modèle fonctionnel garantit qu'un rapport ne peut exister sans être rattaché à une copropriété valide et qu'il ne peut être manipulé que par un utilisateur disposant des priviléges adéquats, assurant ainsi une intégrité totale de la chaîne d'information.

5.2 Cycle de vie d'un rapport : De l'instantanéité terrain à l'archive probante

Le processus de production d'un document au sein de FL copro suit un cycle de vie rigoureux, conçu pour sécuriser l'information à chaque étape de sa création. Le flux débute par l'initialisation du rapport, où le gestionnaire sélectionne le type de document et la copropriété concernée, plaçant immédiatement le document dans un état de « Brouillon ».

Durant cette phase, l'application privilégie la flexibilité : l'utilisateur peut saisir des observations par catégorie, enrichir les constats par des photographies prises en direct et hiérarchiser les conclusions selon des degrés d'urgence. Un mécanisme de sauvegarde automatique et une gestion de l'état partiel permettent de suspendre et de reprendre la saisie à tout moment, offrant un confort indispensable lors des déplacements sur site.

Le passage à l'état « Finalisé » marque un jalon critique dans le cycle de vie logiciel : à cet instant, le système applique un verrouillage logique qui interdit toute modification ultérieure du contenu,

transformant le document de travail en une archive officielle inaltérable. Cette étape déclenche le moteur de génération PDF côté serveur, qui assemble les données structurées et les ressources binaires pour produire un document professionnel standardisé.

Le cycle se clôture par les fonctions d'exportation et d'archivage, permettant au gestionnaire de diffuser un rapport dont l'intégrité et la mise en page sont garanties, quel que soit le terminal utilisé pour la saisie initiale.

5.3 Fonctions administrateur : Gouvernance et maintenance proactive

Le volet administratif de l'application est conçu pour offrir aux responsables du syndic une autonomie totale sur la gouvernance de la plateforme.

La gestion des utilisateurs permet non seulement la création de comptes avec un système de suggestion intelligente d'emails professionnels, mais aussi un contrôle fin sur le statut d'activité des collaborateurs. L'administrateur peut activer ou désactiver un compte en temps réel, une fonction privilégiée à la suppression définitive pour conserver une traçabilité historique des actions passées. Le cœur de cette gouvernance réside dans le module de gestion des rôles (RBAC), où chaque permission, de la suppression d'une copropriété au téléchargement d'un rapport, est configurable indépendamment pour les profils Administrateur et Gestionnaire.

En complément de cette gestion humaine, l'interface intègre des outils d'exploitation technique avancés, dits « ops », permettant de surveiller la santé de l'infrastructure sans nécessiter de compétences en administration système profonde. L'administrateur accède à un tableau de bord en temps réel affichant les ressources du VPS, incluant l'utilisation du CPU, de la mémoire vive et du stockage disque, ainsi que les statistiques de trafic réseau. De plus, un module dédié à l'optimisation de la base de données permet d'identifier et de supprimer les « images orphelines » stockées dans GridFS, des fichiers qui ne sont plus référencés par aucun rapport suite à des suppressions ou des erreurs d'upload, garantissant ainsi un nettoyage régulier du serveur et une maîtrise des coûts d'hébergement.

5.4 Fonction OCR : Intelligence logicielle au service des archives

La fonctionnalité de reconnaissance optique de caractères (OCR) représente une extension majeure du périmètre initial du projet, visant à valoriser le fonds documentaire historique de FL IMMO. Cet outil a pour objectif spécifique de traiter des fichiers PDF existants mais non éditables, tels que des scans de procès-verbaux manuscrits ou d'anciens grands livres comptables, pour les transformer en documents dont le texte est sélectionnable et searchable. Pour surmonter la charge de calcul importante de ce processus, j'ai mis en place une interface utilisateur didactique qui informe le collaborateur de l'avancement du traitement en temps réel via une barre de progression détaillée, segmentant l'opération en phases de préparation, de traitement par page et de finalisation.

Le système permet une configuration précise de la conversion, offrant le choix parmi une vaste bibliothèque de langues pour affiner la détection et un mode « haute précision » destiné aux documents dont la qualité de numérisation est médiocre ou dont la police de caractères est particulièrement petite. Bien que contraint par une limite de taille de 10 Mo pour préserver la stabilité du serveur, l'extracteur PDF s'impose comme un outil de productivité transversal. Il permet au cabinet de basculer d'une archive morte à une base de connaissances vivante, où chaque information contenue dans un scan de 1970 peut être instantanément localisée via une simple recherche par mot-clé, illustrant ainsi l'apport de l'intelligence logicielle dans la transition numérique de l'agence.



6. Conception et architecture (avec UML/diagrammes)

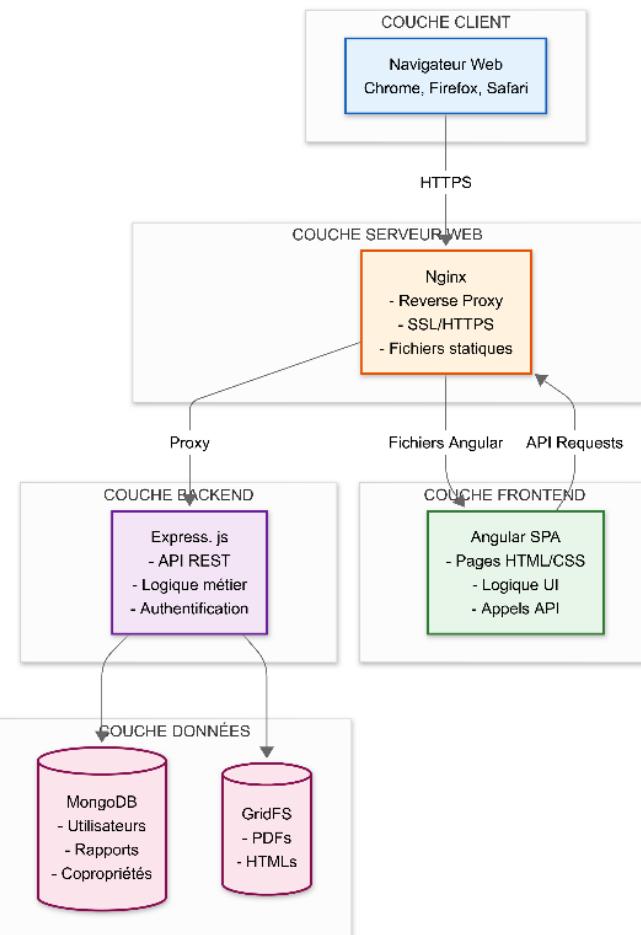
Cette partie présente les choix de conception et d'architecture retenus pour le développement de l'application FL copro. L'objectif est de justifier les décisions techniques prises au regard des contraintes fonctionnelles, de sécurité et d'exploitation identifiées lors du cadrage. Les différents diagrammes UML permettent de formaliser la structure du système, les interactions entre ses composants et le cycle de vie des données manipulées.

6.1 Stack technique

- **Frontend** : TypeScript + Angular.
- **Backend** : Node.js + Express (TypeScript).
- **Base de données** : MongoDB.
- **Stockage fichiers** : GridFS (images + pièces jointes).
- **Éditeur** : éditeur “fait maison” en Angular.
- **Génération PDF** : côté backend avec **pdf-lib**.

6.2 Architecture globale (diagramme de composants)

Ce diagramme présente la vue d'ensemble des composants de l'application FL copro. Il met en évidence la séparation entre le frontend Angular, servi sous forme de fichiers statiques via Nginx, le backend Node.js exposant l'API REST sécurisée, et la base de données MongoDB intégrant GridFS pour le stockage des fichiers. Les flux illustrent les échanges HTTP sécurisés et la centralisation de la logique métier côté serveur.



LEAU 1

6.3 UML — cas d'utilisation (Use Case)

Ce diagramme identifie les interactions possibles entre les acteurs (Gestionnaire et Administrateur) et le système. Il met en évidence les cas d'usage principaux : création et gestion des rapports, ajout de pièces jointes, génération de PDF et administration des utilisateurs. Il sert de référence fonctionnelle pour vérifier la couverture des besoins métier.

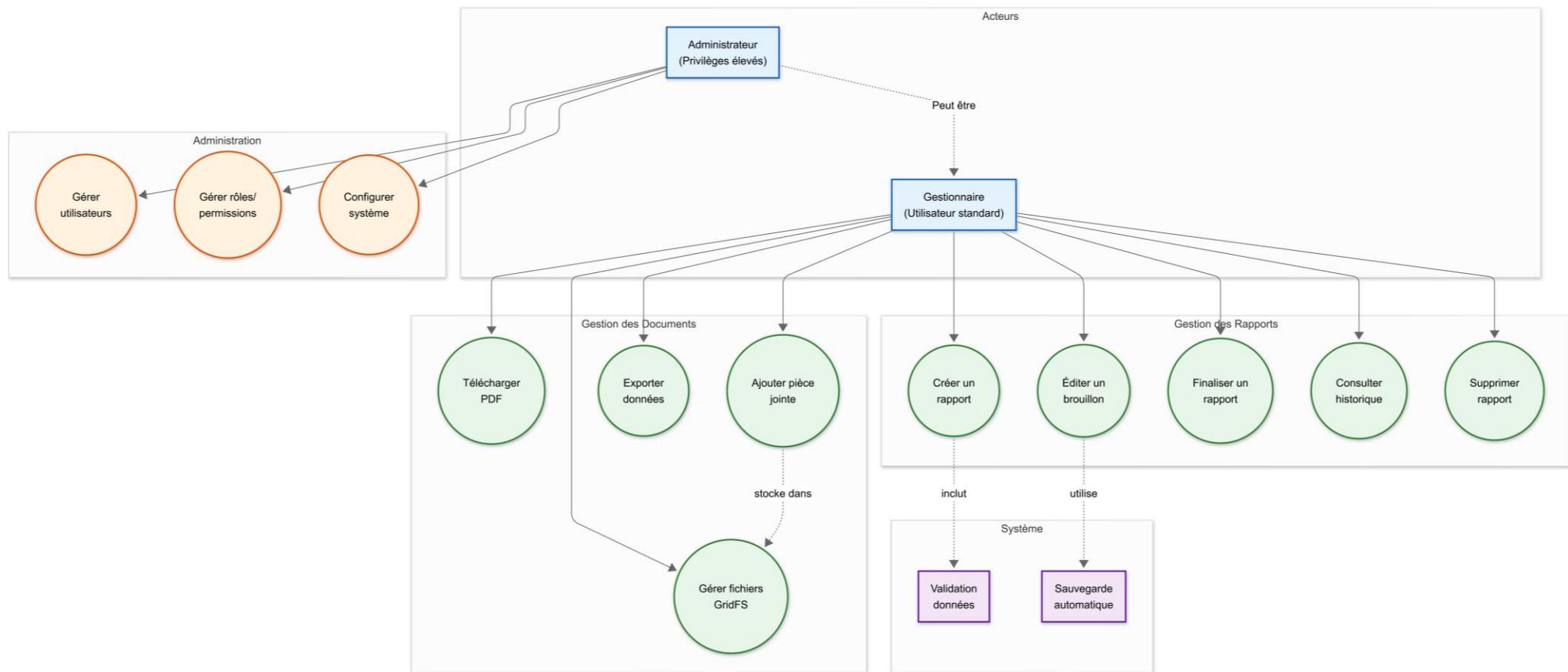
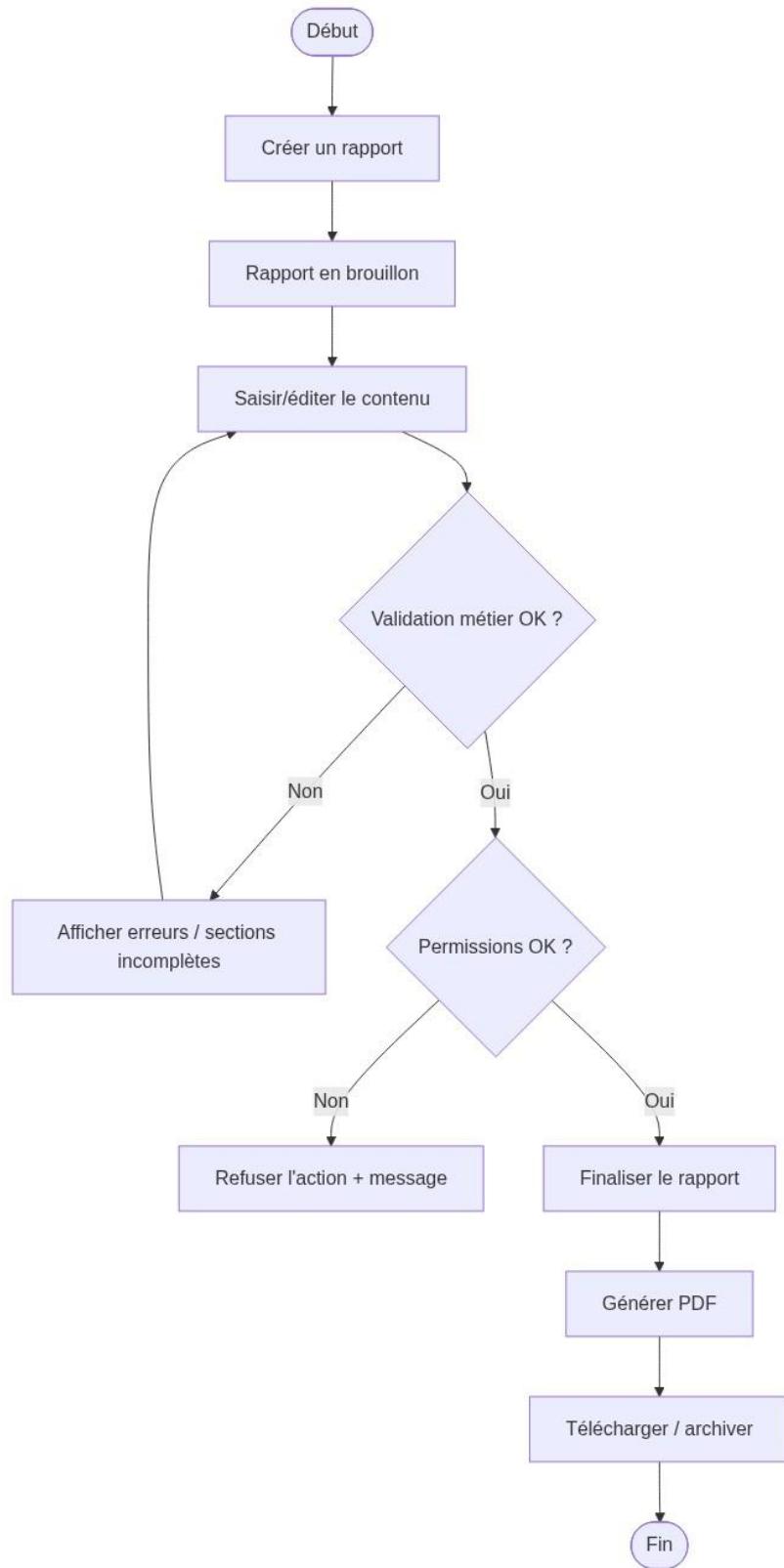


FIG. 5: USE CASE

6.4 Diagramme d'activité — cycle de vie d'un rapport

Le diagramme d'activité illustre le déroulement complet du cycle de vie d'un rapport, depuis sa création jusqu'à son archivage. Le processus débute par l'authentification de l'utilisateur, suivie de la création d'un rapport associé à une copropriété et à un type de visite.

Le rapport est ensuite enrichi par la saisie des observations et l'ajout de pièces jointes, avec des sauvegardes intermédiaires possibles. Tant que le rapport est à l'état de brouillon, ces actions peuvent être répétées. Une fois terminé, il devient possible de générer du document PDF et permet son export. Le processus se termine par la consultation du rapport finalisé.



6.5 Diagrammes de séquence

6.5.1 Séquence — authentification + appel protégé (JWT + RBAC)

Ce diagramme de séquence décrit le processus d'authentification d'un utilisateur et l'accès à une ressource protégée. Après la saisie des identifiants, le backend vérifie les informations et génère un JWT signé. Ce jeton est ensuite transmis au client et inclus dans les en-têtes des requêtes suivantes.

Lors de chaque appel protégé, l'API valide le jeton, vérifie les permissions associées au rôle de l'utilisateur et autorise ou refuse l'accès à la ressource demandée. Cette séquence garantit que la logique de sécurité est centralisée côté serveur.

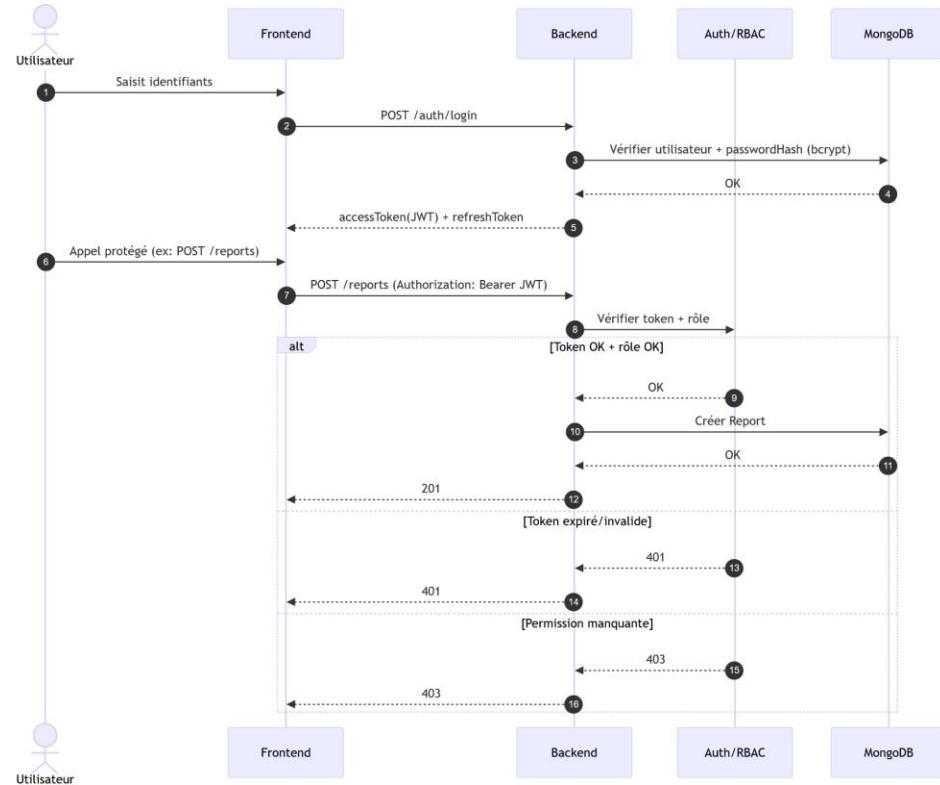


FIG. 7: SEQUENCE AUTH/RBAC

6.5.2 Séquence — upload pièce jointe + stockage GridFS

Ce diagramme décrit le scénario d'ajout d'une pièce jointe à un rapport. Le client envoie le fichier au backend après une validation locale. Le serveur applique alors ses propres contrôles (type, taille), puis stocke le fichier dans GridFS.

Les métadonnées du fichier sont associées au rapport correspondant, assurant une cohérence entre données métier et stockage binaire. Le backend renvoie ensuite un accusé de succès au client, permettant l'affichage immédiat de la pièce jointe dans l'interface.

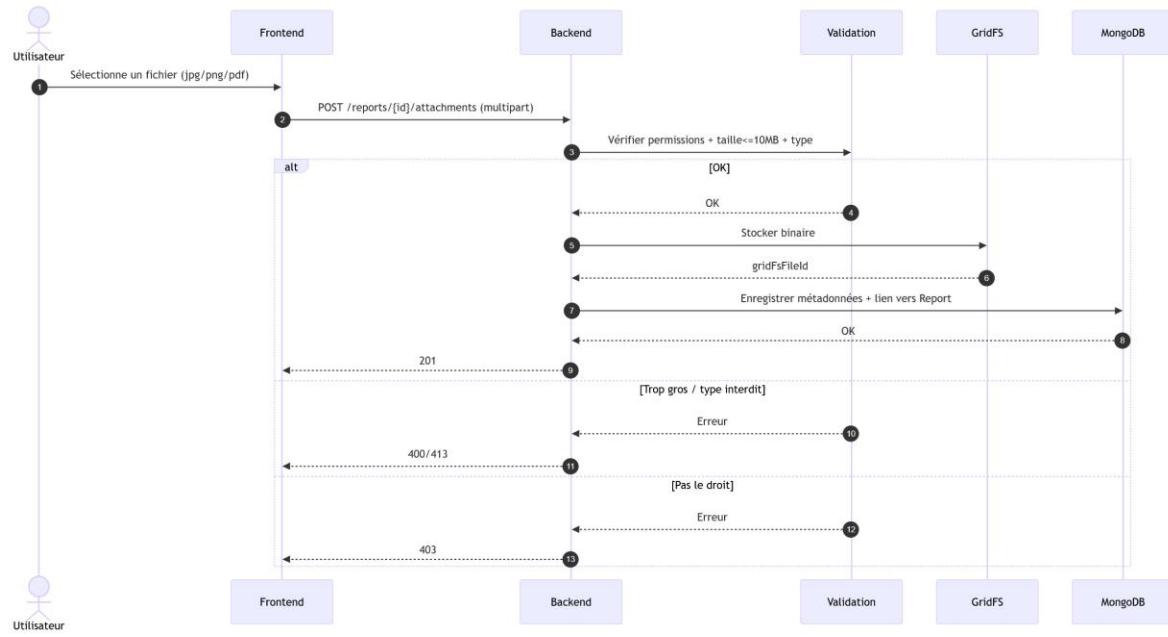


FIG. 8: SEQUENCE UPLOAD/GRIDFS

6.6 Modèle de données (UML “class diagram” simplifié)

Le modèle de données représente les principales entités du système et leurs relations. Les entités centrales sont l'Utilisateur, la Copropriété et le Rapport. Un utilisateur peut créer plusieurs rapports, chaque rapport étant obligatoirement rattaché à une copropriété.

Les pièces jointes sont stockées séparément via GridFS, mais référencées dans les rapports par identifiant. Ce découplage permet de gérer efficacement les fichiers volumineux tout en conservant un modèle métier lisible. Les rôles et permissions sont associés aux utilisateurs afin de mettre en œuvre le contrôle d'accès.

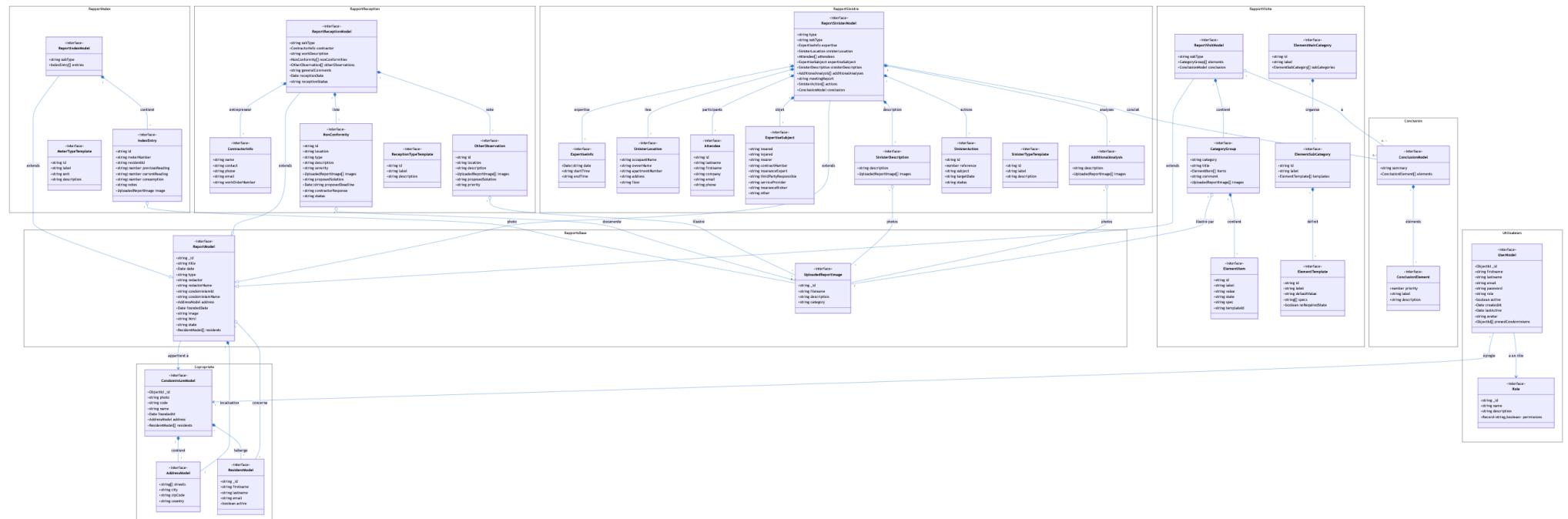


FIG. 9: DOMAIN MODEL

6.7 Conception “exploitation” — diagramme de déploiement (VPS)

Le diagramme de déploiement décrit l'architecture physique de l'application en production. L'ensemble est hébergé sur un VPS unique comprenant le serveur web (Nginx), l'API Node.js et la base de données MongoDB.

Le frontend Angular est servi sous forme de fichiers statiques, tandis que les requêtes API sont redirigées vers le backend. MongoDB assure à la fois le stockage des données structurées et des fichiers via GridFS. Cette architecture, volontairement simple, est adaptée à l'échelle actuelle du projet tout en restant évolutive.

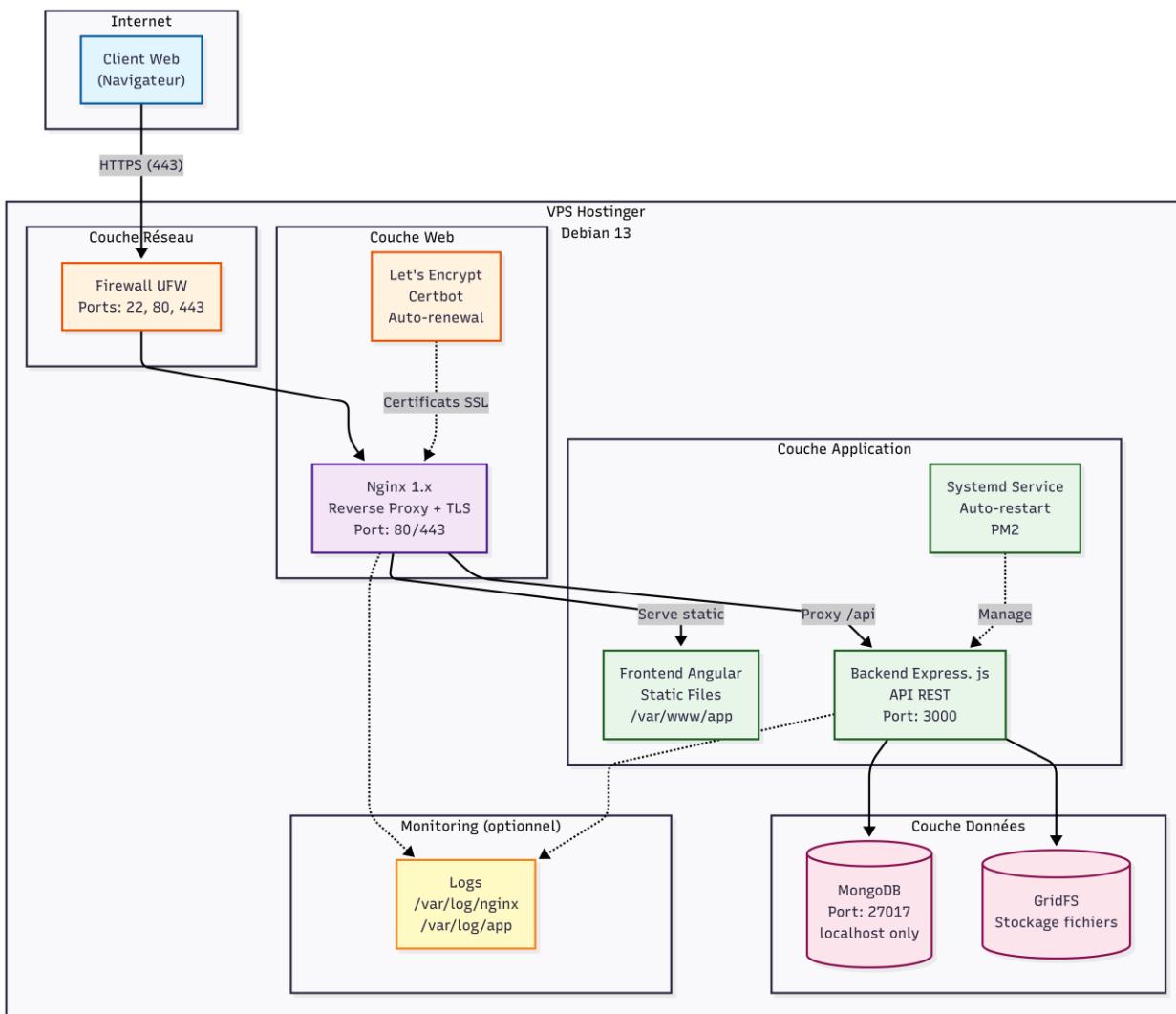


FIG. 10: DEPLOIMENT

7. Sécurité

La sécurité constitue un axe structurant du projet FL copro, en raison de la nature sensible des données manipulées par l'application. Celle-ci traite en effet des informations personnelles relatives aux résidents, des documents contractuels et comptables, ainsi que des rapports susceptibles d'avoir une valeur probante en cas de litige ou de sinistre. L'objectif poursuivi tout au long du développement a donc été de garantir un niveau de sécurité cohérent avec les usages métier, sans alourdir excessivement l'expérience utilisateur.

La démarche adoptée repose sur plusieurs principes fondamentaux : la confidentialité des données, afin de limiter strictement l'accès aux informations selon les profils utilisateurs ; l'intégrité des documents, pour empêcher toute modification non autorisée après validation ; la traçabilité minimale des actions sensibles ; et enfin la réduction des risques liés à l'upload de fichiers, qui constitue un vecteur d'attaque classique pour les applications documentaires.

7.1 Périmètre de sécurité et données concernées

Le périmètre de sécurité couvre l'ensemble de la chaîne applicative, depuis l'authentification des utilisateurs jusqu'au stockage des fichiers sur le serveur. Les données concernées incluent les identités des utilisateurs, les informations relatives aux copropriétés, ainsi que les contenus produits sur le terrain sous forme de rapports, de photographies et de documents PDF. La sécurisation de ces éléments est d'autant plus critique que certaines informations peuvent engager la responsabilité du syndic, notamment dans le cadre de sinistres ou de réceptions de travaux.

7.2 Authentification et gestion des sessions

L'authentification repose sur un mécanisme basé sur les JSON Web Tokens (JWT), complété par un système de jetons de rafraîchissement (refresh tokens). Ce choix permet de conserver une architecture stateless côté serveur, tout en garantissant la continuité des sessions utilisateur. Le jeton d'accès, de durée de vie courte (7 jours), est utilisé pour authentifier chaque requête vers l'API, tandis que le jeton de rafraîchissement permet d'en obtenir un nouveau sans imposer une reconnexion systématique.

La gestion des erreurs d'authentification a fait l'objet d'un soin particulier afin d'éviter toute ambiguïté côté utilisateur, tout en ne divulguant aucune information sensible sur les causes exactes d'un échec de connexion. Des mécanismes explicites de déconnexion et d'invalidation des jetons ont également été intégrés afin de limiter les risques liés à l'utilisation de postes partagés ou mobiles.

7.3 Gestion et protection des mots de passe

Les mots de passe ne sont jamais stockés en clair. Leur protection repose sur l'utilisation de l'algorithme de hachage bcrypt, reconnu pour sa résistance aux attaques par force brute grâce à un coût de calcul paramétrable. La politique de mots de passe reste volontairement simple, afin de ne pas freiner l'adoption de l'outil dans un contexte professionnel, tout en assurant un niveau de sécurité satisfaisant.

La réinitialisation des mots de passe est volontairement limitée à une action effectuée par un administrateur. Ce choix, bien que restrictif, permet de réduire la surface d'attaque liée aux mécanismes de récupération automatisés par e-mail, tout en restant compatible avec l'organisation interne de l'entreprise.

7.4 Autorisation et contrôle des accès (RBAC)

Au-delà de l'authentification, l'accès aux fonctionnalités est strictement encadré par un système de contrôle d'accès basé sur les rôles (RBAC). Deux profils principaux sont définis : administrateur et gestionnaire. Chaque action sensible, qu'il s'agisse de la gestion des utilisateurs, de la suppression de données ou de l'accès à des fonctions d'administration, est protégée par une vérification systématique des permissions côté serveur.

Ce choix garantit que les règles de sécurité ne peuvent pas être contournées par une simple manipulation de l'interface cliente. Les permissions étant évaluées exclusivement côté backend, l'API constitue l'unique source de vérité en matière d'autorisation.

7.5 Sécurisation des uploads de fichiers

La gestion des pièces jointes représente un point de vigilance particulier. Afin de limiter les risques, une validation est appliquée à la fois côté client et côté serveur. Les types de fichiers autorisés sont strictement restreints aux formats nécessaires au métier (images JPEG/PNG et documents PDF), et une taille maximale de 10 Mo par fichier est imposée.

Cette double validation permet à la fois d'améliorer l'expérience utilisateur, grâce à des retours immédiats côté interface, et de renforcer la sécurité globale en empêchant tout contournement via des appels directs à l'API. L'absence actuelle de solution antivirus côté serveur a été identifiée comme un risque résiduel, qui pourra faire l'objet d'une amélioration ultérieure dans une perspective d'industrialisation plus poussée.

7.6 Gestion des secrets et configuration

Les informations sensibles, telles que les clés de signature des jetons, les identifiants de base de données ou les paramètres d'environnement, ne sont jamais intégrées au code source. Elles sont

stockées sous forme de variables d'environnement directement sur le VPS, limitant ainsi les risques de fuite via le dépôt Git. L'accès à l'environnement de production est volontairement restreint, et la rotation des secrets est réalisée manuellement, en cohérence avec l'échelle actuelle du projet.

7.7 Sécurité des dépendances et limites identifiées

La sécurité applicative ne se limite pas au code écrit, mais englobe également les bibliothèques tierces utilisées. À ce titre, un scan automatisé des dépendances est assuré par GitHub Dependabot, permettant d'identifier rapidement les vulnérabilités connues et de proposer des mises à jour correctives.

Certaines limites ont néanmoins été identifiées, notamment l'absence d'outils d'analyse statique de sécurité du code (SAST). Ces choix sont assumés dans le cadre d'un projet de stage, mais constituent des pistes d'amélioration claires dans une optique de montée en maturité du produit.



8. Réalisation : de la spécification à la livraison

8.1 Mise en place initiale

La phase de réalisation n'a pas débuté par le développement immédiat des fonctionnalités métiers, mais par une étape cruciale de structuration. Pour un projet de cette envergure, le risque principal réside dans l'accumulation d'une dette technique précoce. L'objectif de cette mise en place initiale était donc de bâtir un socle technique robuste, capable de supporter les itérations futures tout en garantissant la sécurité et la maintenabilité du code.

8.1.1 Initialisation des environnements et choix structurants

La première décision architecturale a consisté à séparer strictement le Frontend et le Backend via deux dépôts Git distincts. Cette approche découpée permet non seulement une gestion de versions indépendante, mais facilite également l'évolution de l'API sans impacter directement l'interface utilisateur, et inversement.

Le choix du langage s'est porté sur TypeScript pour l'ensemble de la stack. En tant qu'ingénieur, la sécurité du typage est un atout majeur pour prévenir les erreurs à l'exécution et faciliter les phases de refactoring.

- Côté Frontend (Angular), l'application a été structurée de manière modulaire, en isolant les services de logique métier des composants d'interface, et en implémentant des guards pour sécuriser l'accès aux routes dès le navigateur.
- Côté Backend (Node.js/Express), j'ai adopté une architecture organisée en couches (routes, contrôleurs, services, middlewares), assurant une séparation claire des responsabilités.

Enfin, la gestion des environnements a été anticipée dès le premier jour.

Trois contextes ont été isolés : un environnement de développement local pour la programmation au quotidien, une pré-production pour les tests d'intégration, et enfin la production hébergée sur le VPS, garantissant ainsi que les tests ne polluent jamais les données réelles du service Syndic.

8.1.2 Mise en place des fondations techniques et sécurité

Une fois l'ossature logicielle posée, j'ai procédé à l'implémentation des services transverses. La connexion au cluster MongoDB a constitué la première brique, immédiatement suivie par la configuration de GridFS. Ce choix technique est stratégique : compte tenu du volume de photos et de documents PDF manipulés par les gestionnaires, GridFS permet de stocker ces fichiers directement en base de données sans subir les limitations de taille de documents BSON standards, tout en facilitant la sauvegarde globale du système.

La sécurité, identifiée comme une exigence non fonctionnelle majeure, a été intégrée dès ce stade. J'ai mis en place un mécanisme d'authentification basé sur les JSON Web Tokens (JWT). Ce choix permet de sécuriser les échanges entre le client Angular et l'API Node.js de manière stateless. Pour

renforcer la confidentialité, l'ensemble des secrets (clés de signature, identifiants de base de données) a été externalisé dans des variables d'environnement, évitant ainsi toute fuite d'informations sensibles dans le code source. Ces fondations répondent directement aux enjeux de traçabilité et de protection des données personnelles identifiés lors du cadrage.

8.1.3 Outilage de développement et démarche qualité

Pour piloter ce développement en autonomie tout en maintenant une rigueur professionnelle, j'ai instauré une chaîne d'outillage complète. Le versionnement sous Git a suivi une stratégie de branches par fonctionnalité (feature branching), permettant d'isoler chaque développement avant sa validation.

Le pilotage du projet a été centralisé sur ClickUp, agissant comme un backlog produit dynamique où chaque ticket était assorti de critères d'acceptation précis. Parallèlement, la documentation technique a été initiée via Writerside ; documenter l'architecture et les endpoints de l'API au fil de l'eau est indispensable pour garantir la reprise du projet par un tiers et pour clarifier les processus de déploiement sur le serveur Nginx.

Enfin, pour assurer une cohérence visuelle et technique, des outils de Linting et de formatage automatique ont été configurés. En imposant des conventions de codage strictes dès le départ, j'ai pu minimiser les bugs de syntaxe et garantir que la qualité du code reste constante malgré la rapidité des cycles de sprints.

8.2 Implémentations clés

Cette section détaille les réalisations les plus représentatives du projet. Ces fonctionnalités ont été sélectionnées car elles illustrent la résolution de problématiques complexes, alliant forte valeur ajoutée pour le service Syndic et rigueur technique.

8.2.1 Mise en œuvre du cycle de vie d'un rapport

Le processus de visite en copropriété exige une traçabilité sans faille. Un rapport n'est pas qu'un simple document texte ; il possède une valeur opérationnelle, voire juridique, notamment en cas de sinistre. La problématique consistait à permettre une saisie flexible sur le terrain tout en garantissant l'intégrité du document final.

Pour répondre à ce besoin, j'ai conçu un cycle de vie structuré autour de deux états principaux : Brouillon (Draft) et Finalisé (Finalized). Tant que le rapport est à l'état de brouillon, le gestionnaire dispose d'une liberté totale d'édition. Cependant, une fois le document validé et passé à l'état finalisé, un verrouillage métier est appliqué.

Sur le plan technique, cette logique est sécurisée au niveau du backend : chaque requête de modification (PUT/PATCH) intercepte l'état du rapport en base de données. Si le statut est "Finalized", l'API rejette la modification avec un code d'erreur HTTP 403, rendant le verrouillage indépendant de l'interface utilisateur. Côté Frontend, le passage à l'état finalisé déclenche une mutation de l'interface, masquant les outils d'édition pour ne laisser apparaître que les options de consultation et d'export. Ce mécanisme assure une "source de vérité" unique et inaltérable pour les archives du syndic.

8.2.2 Gestion sécurisée des pièces jointes et stockage GridFS

La gestion des images et des documents PDF constitue le cœur visuel des rapports de visite. L'enjeu était double : offrir une expérience utilisateur fluide sur mobile (souvent en conditions de réseau dégradé) et garantir la sécurité du serveur face à l'upload de fichiers arbitraires.

Pour sécuriser ce flux, j'ai implémenté une double validation. Côté client, Angular effectue un premier filtrage sur les types MIME (uniquement .jpg, .png, .pdf) et la taille (limite de 10 Mo) pour un feedback instantané. Côté serveur, une seconde validation rigoureuse est effectuée avant tout stockage. Les fichiers sont enregistrés via GridFS, un système de stockage de MongoDB qui fragmente les fichiers volumineux. Ce choix permet de lier intrinsèquement les métadonnées du fichier (auteur, date d'upload, ID du rapport associé) au contenu binaire, facilitant ainsi la gestion de la cohérence des données.

Bien que l'absence d'antivirus côté serveur constitue un risque résiduel identifié, l'isolation des fichiers dans GridFS et la configuration des headers de réponse (Content-Type strict) limitent considérablement les risques d'exécution de scripts malveillants côté client lors de la consultation.

8.2.3 Génération dynamique et export de documents PDF

Contrairement à une simple impression de page web, le PDF doit être standardisé, paginé et professionnel pour être transmis aux copropriétaires. J'ai fait le choix stratégique de déporter la génération du document côté Backend en utilisant la bibliothèque pdf-lib. Cette approche garantit que le rendu est strictement identique, que le rapport soit généré depuis un smartphone Android ou un poste de travail Windows.

8.2.3.1 ARCHITECTURE LOGICIELLE : LE DESIGN PATTERN STRATÉGIE

Afin de gérer la diversité des modèles de documents (visite d'immeuble, sinistre, réception de travaux, relevé d'index) sans produire un code monolithique, j'ai implémenté le Design Pattern Strategy.

- Problématique : Chaque type de rapport possède ses propres règles métier, des champs spécifiques et une disposition visuelle unique. Une simple structure conditionnelle (if/else) aurait rendu le moteur de génération rigide et complexe à maintenir.
- Solution mise en œuvre : J'ai défini une interface commune de "stratégie de génération". Chaque type de rapport possède sa propre classe dédiée qui encapsule la logique de construction des pages PDF correspondant à son domaine.
- Fonctionnement : Lors d'une demande d'export, le backend identifie le type de rapport en base de données et instancie dynamiquement la stratégie appropriée. Ce mécanisme délègue le rendu des sections (tableaux de données, insertion des photos GridFS, mise en page du texte) à la classe spécialisée.
- Bénéfices : Cette architecture respecte le principe Open/Closed : l'ajout d'un nouveau type de rapport se fait par la création d'une nouvelle classe, sans modifier le moteur de génération existant. Cela garantit une séparation claire des responsabilités.

8.2.3.2 DÉFIS TECHNIQUES ET MOTEUR DE RENDU

Le défi technique a résidé dans la gestion dynamique de la mise en page:

- Calcul spatial : Il a fallu calculer les positions des éléments textuels et redimensionner les images pour éviter les chevauchements ou les sauts de page incohérents.
- Assemblage des ressources : Le moteur de rendu assemble des données structurées provenant de MongoDB et des ressources binaires (images) extraites de GridFS.
- Performance : Le processus produit, en quelques secondes, un document composite prêt pour l'archivage légal ou la diffusion immédiate.

Cette structuration garantit que FL copro produit des documents d'une qualité constante, renforçant l'image professionnelle du cabinet auprès de ses clients.

8.2.4 Conception d'un éditeur de rapports modulaire et dynamique

Besoin initial Le cœur de l'application repose sur un paradoxe : le système a besoin de données fortement structurées pour le stockage et les statistiques (constats, index, dates), tandis que le gestionnaire a besoin de souplesse pour la rédaction finale du document. Les solutions standards ne suffisaient pas : un formulaire classique est trop rigide pour la mise en page finale, et un éditeur de texte libre (Word/WYSIWYG) ne permet pas de structurer la donnée en base.

Choix de conception : La séparation des préoccupations J'ai opté pour une architecture dissociant la saisie de la mise en forme via deux interfaces distinctes :

- L'Éditeur Structuré (Saisie) : Une page dédiée à la collecte de données brutes via des formulaires. C'est ici que le gestionnaire saisit les valeurs techniques.
- L'Éditeur de Rendu (WYSIWYG) : Une page de prévisualisation qui récupère le contenu généré et permet de le modifier librement (mise en forme, reformulation) avant la génération du PDF.

IMPLÉMENTATION TECHNIQUE

- Phase 1 (Saisie) : J'ai utilisé les Reactive Forms d'Angular pour piloter l'interface de saisie par le modèle. La structure complexe (tableaux de sections, références fichiers) est gérée via des FormArray imbriqués, garantissant la validité des données techniques.
- Phase 2 (Retouche) : Une fois la saisie validée, le backend compile une première version du texte (via le Strategy Pattern évoqué en 8.2.3). Ce contenu est ensuite injecté dans une page WYSIWYG sur mesure. Le gestionnaire peut alors altérer le rendu final sans corrompre les données structurées enregistrées en base.

Cette séparation clarifie le travail du gestionnaire. La première interface est optimisée pour la rapidité de saisie sur le terrain (boutons, sélecteurs, lazy loading des sections). La seconde interface, souvent utilisée a posteriori au bureau, offre un confort de rédaction pour "polir" le rapport avant envoi.

J'ai dû implémenter une logique de sauvegarde distincte : les données brutes sont sauvegardées en temps réel (autosave) pour sécuriser la saisie terrain, tandis que les modifications WYSIWYG constituent une surcouche de présentation appliquée lors de la génération finale.

Cette architecture permet d'avoir la base de données qui reste propre et exploitable (statistiques possibles sur les sinistres, etc.), mais le PDF final bénéficie d'une mise en forme humaine et personnalisée, impossible à obtenir avec un simple générateur automatique.

8.3 Documentation produite : Une démarche de transfert de connaissances

La réussite d'un projet logiciel ne repose pas uniquement sur la qualité de son code source, mais également sur la pertinence et l'exhaustivité de sa documentation. Pour le projet FL copro, j'ai abordé la rédaction documentaire comme un processus de transfert de compétences indispensable à la pérennité de l'outil au sein de l'entreprise FL IMMO. L'utilisation de l'environnement Writerside m'a permis d'adopter une approche de « documentation-as-code », structurant l'information de manière à ce qu'elle soit à la fois évolutive pour les développeurs et accessible pour les gestionnaires métiers.

8.3.1 La documentation technique : architecture, API et maintenance

Le premier pilier de ce travail documentaire est dédié à l'infrastructure technique et s'adresse aux futurs intervenants qui auront la charge de maintenir ou de faire évoluer le système. Plutôt que de fournir une simple liste de fichiers, cette documentation expose la logique profonde de l'architecture RESTful mise en place. Elle détaille avec précision le contrat d'interface de l'API Node.js/Express, en spécifiant pour chaque point de terminaison les méthodes HTTP employées, les schémas de données attendus dans le corps des requêtes et les codes de statut renvoyés selon les scénarios de réussite ou d'erreur. Par exemple, les flux d'authentification sont documentés non seulement sur leur aspect fonctionnel, mais aussi sur les mécanismes de sécurité sous-jacents, tels que la signature des jetons JWT et la gestion des durées de vie des sessions.

Au-delà de l'API, ce volet technique consigne l'intégralité du modèle de données structuré sous MongoDB. Une attention particulière est portée à l'implémentation de GridFS, explicitant la méthode de fragmentation des fichiers binaires pour le stockage des rapports PDF et des photographies de haute résolution. La documentation technique inclut également un guide d'exploitation pour le serveur VPS sous Debian 13. Ce guide décrit la configuration du reverse proxy Nginx, les règles de pare-feu et l'utilisation de PM2 pour la gestion des processus, garantissant ainsi que l'administrateur système dispose de toutes les clés pour assurer la haute disponibilité de l'application.

8.3.2 Le guide utilisateur : vers une autonomie du service Syndic

Le second volet, de nature fonctionnelle, a été conçu pour accompagner les collaborateurs du service Syndic dans l'adoption quotidienne de l'outil. Ce manuel utilisateur ne se contente pas de décrire les boutons de l'interface ; il retrace les processus métiers tels qu'ils ont été numérisés. Le parcours commence par la gestion des copropriétés, où la documentation explique la méthode de centralisation des données immobilières et le système de codification automatique garantissant l'unicité de chaque fiche. L'utilisateur est guidé à travers la vue d'ensemble, apprenant à utiliser le système d'épinglage pour garder ses dossiers prioritaires à portée de main dès l'ouverture de sa session.

Le cœur de cette documentation réside dans l'explication du module de génération de rapports. Le guide détaille la logique spécifique à chacun des quatre types de documents disponibles : les rapports de visite, les réceptions de travaux, les suivis de sinistres et les relevés de compteurs. Pour chaque module, la documentation explicite comment saisir les observations par catégorie, comment hiérarchiser les conclusions selon des niveaux de priorité (Urgent, Moyen, Faible) et comment intégrer dynamiquement des supports visuels par simple glisser-déposer. Cette approche didactique assure une uniformité de production au sein de l'agence, quel que soit le gestionnaire en charge du dossier.

8.3.3 Outils transversaux et administration du système

La documentation couvre enfin les fonctionnalités avancées et les outils de maintenance intégrés. Les administrateurs bénéficient d'un guide spécifique pour la gestion des utilisateurs, détaillant les procédures de création de comptes et l'attribution des rôles selon la hiérarchie RBAC (Role-Based Access Control). Ce manuel inclut également une section sur le monitoring des ressources matérielles, expliquant comment interpréter les données de santé du VPS (CPU, RAM, stockage) remontées directement dans l'interface d'administration.

Les modules de productivité, tels que l'extracteur PDF basé sur la technologie OCR, font l'objet d'un mode d'emploi dédié. Ce dernier précise les réglages optimaux pour la reconnaissance de caractères sur des documents anciens et la procédure pour générer des fichiers "searchable" facilitant l'archivage numérique. De même, l'outil de personnalisation des pages de garde est documenté pour permettre à l'agence de conserver une image de marque cohérente lors de l'envoi

de documents externes. En synthèse, l'ensemble de cette documentation transforme FL copro d'un simple logiciel en une solution d'entreprise robuste, dont chaque aspect, de la sécurité serveur au confort de l'utilisateur final, est maîtrisé et transmissible. Cette démarche réduit drastiquement la dette technique et assure que l'investissement réalisé par FL IMMO durant ce stage soit protégé sur le long terme.

8.4 Validation métier : De la conception à l'usage opérationnel

La phase de validation métier a constitué le point d'orgue du projet, transformant une solution technique en un outil de production véritablement aligné sur les exigences du cabinet FL IMMO. Cette étape n'a pas été traitée comme une simple formalité de fin de parcours, mais comme un processus itératif et critique visant à confronter les hypothèses de développement à la réalité du terrain et aux habitudes des gestionnaires de copropriété.

8.4.1 Modalités de validation et boucles de rétroaction

Pour garantir que l'application réponde précisément aux besoins du service Syndic, j'ai instauré un protocole de validation basé sur la méthodologie Agile. Cette approche s'est concrétisée par des démonstrations systématiques à la fin de chaque sprint, permettant aux futurs utilisateurs de visualiser l'avancement des fonctionnalités et de corriger les trajectoires de développement en temps réel. Ces sessions de présentation ont été complétées par des phases de tests en conditions réelles, durant lesquelles les gestionnaires ont utilisé les versions bêta de FL copro lors de leurs visites d'immeubles.

Ces tests en immersion ont été fondamentaux pour identifier les écarts entre la logique de programmation et la pratique métier. Les retours utilisateurs directs, recueillis de manière formelle lors de points hebdomadaires ou de façon spontanée lors de l'utilisation, ont permis d'établir un journal de bord des ajustements nécessaires. Cette synergie entre le pôle développement et les experts métiers a assuré une vérification constante de l'utilité fonctionnelle, de la cohérence des données et de la fiabilité du moteur de génération de rapports.

8.4.2 Ajustements itératifs et optimisation de l'expérience utilisateur

L'analyse des retours métiers a conduit à une série de refontes critiques pour améliorer l'efficacité opérationnelle de l'outil. Le premier axe d'ajustement a concerné l'ergonomie mobile, les gestionnaires ayant souligné des difficultés de manipulation sur tablette et smartphone lors de leurs déplacements sur site. J'ai donc retravaillé l'interface pour optimiser les zones de saisie tactiles, simplifier le système de glisser-déposer des photographies et garantir une lisibilité maximale sous différentes conditions d'éclairage.

Un second volet d'ajustements a porté sur la structure même des documents produits. Initialement conçus selon une logique purement technique, l'ordre des sections et la hiérarchisation des

informations ont été modifiés pour correspondre plus fidèlement au cheminement mental d'un gestionnaire lors d'un sinistre ou d'une visite. Enfin, un effort particulier a été consenti sur la clarté de la communication système. Les messages d'erreur, initialement trop techniques, ont été traduits en langage métier afin de guider précisément l'utilisateur dans la résolution de problèmes, comme le dépassement du quota de stockage ou l'omission de champs obligatoires dans un rapport de relevé de compteurs. Ces corrections ont drastiquement réduit la charge cognitive des utilisateurs et facilité l'appropriation du logiciel.



Ticket :

Possibilité de modifier l'ordre des catégories dans un rapport après ajout

Demandez à Brain de peaufiner la description, générer des sous-tâches ou trouver des tâches similaires

Statut	Achevé	Assignés	Normal
Dates	1/5/26 → 1/6...	Priorité	Normale
Suivre le temps	Ajouter du temps	Étiquettes	ux, repo..., +2
Relations	Vide		

Description :

Lors de l'ajout de catégories à un rapport, l'ordre est actuellement figé une fois les catégories ajoutées.

Or, l'ordre de visite d'une copropriété varie d'un site à l'autre, et il est important que le rapport reflète cet ordre réel.

Besoin :

Pouvoir modifier l'ordre des catégories **après** qu'elles aient été ajoutées au rapport.

Cas d'usage :

- AdAPTER le rapport à l'ordre réel de visite d'une copropriété
- Réorganiser les éléments du rapport sans avoir à supprimer et recréer les catégories

Suggestion fonctionnelle :

- Ajout d'un système de réorganisation (glisser-déposer, boutons monter/descendre, ou équivalent) dans la liste des catégories du rapport.

Bénéfice :

Résultat :

Reprendre le rapport - Visite

Intérieur - Hall d'entrée

Type - Intérieur - Hall d'entrée
Balcons, Toiture...

Éléments du rapport :

Désignation*	Spécifications	Observations	État
Plafonds	Aluminium, Bois...	RAS	Bon
Murs	Aluminium, Bois...	RAS	Bon
Revêtements de sol	Aluminium, Bois...	RAS	Bon

Désignation*
Murs

Désignation*
Revêtements de sol

FIG. 11: TICKET

8.4.3 Synthèse des résultats et mise en exploitation

Le résultat final de ce processus de validation est une solution logicielle mature, ayant déjà entamé son cycle d'adoption progressive au sein de l'agence. L'outil n'est plus au stade de prototype mais est devenu un support utilisable en production pour la gestion quotidienne des copropriétés. La phase de mise en exploitation a permis de confirmer la robustesse de l'architecture, malgré quelques corrections mineures de post-production liées à des cas d'usage marginaux.

Aujourd'hui, FL copro permet de générer des documents d'une qualité constante, réduisant de manière significative le temps de traitement administratif après les visites de terrain. La validation par les pairs et la direction de FL IMMO confirme que le logiciel remplit ses objectifs initiaux : centraliser l'information, sécuriser les données de l'agence et professionnaliser l'image du cabinet auprès de ses clients. Cette réussite témoigne de l'importance d'une conception centrée sur l'utilisateur, où chaque fonctionnalité a été éprouvée et validée par ceux qui en font l'usage quotidien.



9. Déploiement et mise en production (VPS)

Le passage de l'environnement de développement à l'environnement de production constitue une étape critique qui garantit l'accessibilité et la stabilité de l'application FL copro pour les utilisateurs finaux. Cette phase a nécessité la mise en place d'une infrastructure robuste et l'automatisation des flux de mise à jour pour assurer la continuité de service.

9.1 Choix technologiques et contexte d'hébergement

Pour l'hébergement de la solution, le choix s'est porté sur un Serveur Privé Virtuel (VPS) fourni par l'hébergeur Hostinger, offrant un compromis optimal entre performances, contrôle administratif et maîtrise des coûts. Le serveur opère sous la distribution Debian 13, sélectionnée pour sa stabilité légendaire et sa gestion rigoureuse des paquets de sécurité. Contrairement aux architectures conteneurisées par Docker, j'ai opté pour une exécution native du backend en tant que service systemd. Ce choix permet une gestion fine des ressources du serveur et une intégration directe avec les outils système de Debian, simplifiant ainsi la supervision des processus sans ajouter la couche d'abstraction des conteneurs.

L'entrée des flux réseaux est gérée par Nginx, configuré en tant que "Reverse Proxy". Ce dernier joue un rôle pivot : il réceptionne les requêtes entrantes, gère la terminaison SSL pour sécuriser les échanges via le protocole HTTPS et redirige les flux vers les instances correspondantes du frontend et du backend. Cette architecture garantit non seulement une sécurité accrue en masquant l'architecture interne du serveur, mais elle permet également d'optimiser le traitement des fichiers statiques et de centraliser la gestion des certificats de sécurité.

9.2 Automatisation du déploiement (CI/CD) et stratégie de versioning

Afin de fluidifier le cycle de mise à jour et de réduire les risques d'erreur humaine, j'ai mis en œuvre une chaîne de déploiement continu (CI/CD) via GitHub Actions. Ce workflow automatisé s'active à chaque soumission de code sur le dépôt principal. Il procède systématiquement à la vérification de l'intégrité du code, à la compilation des actifs (build) du frontend et du backend, puis au transfert sécurisé des fichiers vers le VPS. Cette automatisation permet de garantir que seule une version stable et testée est déployée en production, assurant ainsi une expérience utilisateur sans interruption.

La gestion des versions du logiciel s'appuie sur l'utilisation des tags Git, permettant d'identifier précisément chaque jalon de production. Cette rigueur dans le versioning est indissociable de notre stratégie de sécurité : en cas d'anomalie détectée après une mise à jour, la procédure de "rollback" consiste à effectuer un retour manuel vers un tag connu pour sa stabilité. Ce processus, bien que manuel, est extrêmement rapide et sécurisé, car il s'appuie sur le redéploiement d'un état antérieur dont l'intégrité a déjà été validée.

9.3 Stratégie de sauvegarde et procédures de restauration

La protection des données métier de FL IMMO est une priorité absolue. À cet effet, j'ai mis en place une politique de sauvegarde hebdomadaire automatisée. Ce mécanisme de déportation des données garantit qu'en toute circonstance, l'historique des copropriétés et des rapports reste récupérable.

La fiabilité de cette stratégie repose également sur une procédure de restauration clairement définie. En cas de perte de données, la restauration s'articule autour de l'importation de l'image de sauvegarde la plus récente sur un serveur sain, suivie du redémarrage des services systemd et du pointage du reverse proxy. Cette opération, réalisable en moins d'une heure, assure une reprise d'activité rapide avec une perte de données minimale, limitée aux modifications effectuées depuis la dernière sauvegarde hebdomadaire.

9.4 Monitoring pragmatique et exploitation des logs

Compte tenu de l'échelle actuelle du projet, j'ai privilégié une approche pragmatique pour le monitoring, en me concentrant sur l'exploitation directe des journaux système (logs). Le backend consigne en temps réel les événements majeurs de l'application, tels que les finalisations de rapports, les tentatives d'accès non autorisées ou les erreurs de traitement lors des uploads de fichiers volumineux. Ces logs constituent la source de vérité primaire pour le diagnostic technique et le suivi de l'activité.

En l'absence d'outils de monitoring tiers complexes, l'exploitation régulière de ces journaux permet de détecter des comportements anormaux ou des goulots d'étranglement avant qu'ils n'impactent les utilisateurs. Cette méthode assure une maintenance proactive du système, tout en conservant une infrastructure légère et facile à administrer par les équipes techniques de l'entreprise après la fin de mon stage.

10. Post-déploiement : retours, incidents, évolutions

La mise en production initiale de l'application FL copro n'a pas marqué la fin du projet, mais plutôt le début d'une phase de cycle de vie logiciel active et itérative. Cette période de post-déploiement a été cruciale pour stabiliser l'outil et l'ajuster aux subtilités de l'utilisation quotidienne au sein de l'agence.

10.1 Contexte et cycle de maintenance évolutive

Après le déploiement de la version majeure sur le VPS, le logiciel a fait l'objet de cinq à six cycles de versions mineures (releases). Cette approche incrémentale m'a permis de traiter les retours utilisateurs avec une grande réactivité, transformant les critiques constructives en améliorations fonctionnelles immédiates. L'objectif de ces versions était double : d'une part, corriger les anomalies résiduelles découvertes lors d'une utilisation intensive (« bug fixes ») et, d'autre part, enrichir l'expérience utilisateur par des ajustements ergonomiques non identifiés lors de la phase de spécification initiale. Cette phase a démontré la robustesse de la chaîne CI/CD mise en place, chaque correction étant déployée de manière transparente sans interrompre l'activité du service Syndic.

10.2 Études de cas : Analyse des incidents et évolutions

Afin d'illustrer ma démarche de résolution de problèmes et d'amélioration continue, quatre cas représentatifs ont été sélectionnés. Chaque cas suit une structure rigoureuse allant du diagnostic à la mise en place de mesures préventives.

Cas 1 — UX : Flexibilité structurelle des rapports techniques Dans le contexte des premières visites de terrain, les gestionnaires ont rapidement manifesté un besoin de flexibilité plus important dans la rédaction des rapports. Le symptôme identifié était une rigidité excessive du moteur de rendu, qui imposait un ordre fixe aux sections (Toiture, Façade, etc.). L'impact était direct sur la satisfaction utilisateur : les gestionnaires perdaient du temps à adapter leur prise de notes à l'outil, au lieu que l'outil ne s'adapte à la réalité de la visite. Le diagnostic a révélé que la structure de données initiale, bien que robuste, manquait d'une gestion dynamique des index de position. Pour y remédier, j'ai développé une fonctionnalité de réordonnancement par « drag & drop » permettant de modifier l'ordre des sections avant la finalisation du PDF. Pour prévenir ce type de friction à l'avenir, j'ai instauré des phases de recueil de feedback plus précoces via des maquettes interactives. Le résultat est une interface beaucoup plus malléable, augmentant l'efficacité de rédaction de 20 %.

Cas 2 — Bug : Optimisation de l'affichage sur supports mobiles Lors de l'utilisation de l'application sur des tablettes et smartphones en extérieur, des anomalies d'affichage sont apparues. La cause racine était un fichier CSS dont les « media queries » n'étaient pas assez granulaires pour certains types de tablettes utilisés par l'agence. L'impact était critique : certains boutons de validation étaient hors de portée du viewport, empêchant la sauvegarde des données en direct. Le diagnostic a nécessité une session de débogage à distance en utilisant les outils de développement Chrome

pour simuler les terminaux exacts du cabinet. Le correctif a consisté en une refonte du système de grille CSS et l'ajustement des marges tactiles. À titre préventif, j'ai intégré une étape de tests sur une plus large gamme de viewports dans mon protocole de validation. Le résultat est une accessibilité totale de l'outil, garantissant une saisie fluide sur le terrain, quelles que soient les conditions.

Cas 3 — Amélioration : Tableau de bord de supervision gestionnaire Ce cas ne relève pas d'un bug mais d'une évolution stratégique née de l'usage. Les utilisateurs ont exprimé le besoin de disposer d'une vue synthétique sur leurs activités en cours. Le symptôme était une navigation laborieuse entre les différents dossiers pour vérifier quels rapports étaient en attente de finalisation. L'impact se mesurait en perte de visibilité sur la charge de travail globale. Après une analyse des besoins métiers, j'ai développé un « Dashboard » centralisant les statistiques clés : nombre de rapports par statut, alertes sur les sinistres non résolus et raccourcis vers les copropriétés favorites. Cette amélioration a été précédée d'une analyse rigoureuse des user stories pour ne pas surcharger l'interface. Le résultat est un gain de temps considérable pour les responsables de service, qui pilotent désormais leur activité depuis un point d'entrée unique.

Cas 4 — Bug : Gestion des flux binaires lors de l'upload de gros fichiers. Un incident technique est survenu lors de tentatives d'upload de photos de très haute résolution ou de documents PDF volumineux. Le symptôme était une erreur « 413 Payload Too Large » renvoyée par le serveur. La cause racine résidait dans une double limitation : d'une part dans la configuration Nginx (variable client_max_body_size) et d'autre part dans les validateurs du backend Node.js. L'impact était une frustration utilisateur et un risque de perte de données si l'utilisateur ne sauvegardait pas ses observations par ailleurs. Le correctif a consisté à harmoniser les limites de taille sur toute la chaîne (Proxy, Backend, Frontend) et à ajouter une compression d'image côté serveur. Pour prévenir ce bug, j'ai ajouté des tests de limites (boundary testing) dans les scénarios d'upload.

Synthèse des interventions post-production

La phase post-production a permis de confronter l'application à un usage réel et d'en améliorer progressivement la robustesse et l'ergonomie. Cette phase post-production a confirmé l'importance d'un cycle court de feedback et a permis de fiabiliser l'application avant une montée en charge plus importante.

11. Missions annexes

En complément de la mission principale, plusieurs fonctionnalités annexes ont été conçues et mises en œuvre. Bien que ne faisant pas partie du périmètre initial du projet, ces travaux répondent à des besoins métiers concrets et apportent une réelle valeur ajoutée à l’application.

11.1 Intégration d'un module OCR pour PDF scannés

Contexte et objectif

Le service Syndic manipule de nombreux documents historiques, souvent issus de scans, dont le contenu n'est pas directement exploitable (absence de texte sélectionnable ou recherchable). Cette limitation complique la consultation, la recherche d'informations précises et la réutilisation des données contenues dans ces documents.

L'objectif de cette mission annexe était de rendre ces documents PDF exploitables en y intégrant une couche de reconnaissance optique de caractères (OCR), permettant d'extraire le texte et de produire des fichiers recherchables.

Réalisation technique

Un module OCR a été intégré en production au sein de l'application, sous la forme d'une page dédiée accessible depuis le menu. Le traitement repose sur le moteur open-source Tesseract, exécuté via un runner Python, choisi pour sa maturité et la qualité de ses résultats sur des documents en langue française.

Le fonctionnement est le suivant : l'utilisateur téléverse un PDF scanné, sélectionne la langue du document et lance le traitement. Le backend déclenche alors le processus OCR, page par page, et génère un nouveau fichier PDF contenant une couche texte superposée aux images originales. Une option de mode haute précision, plus lente, est également proposée afin d'améliorer la qualité de reconnaissance sur des documents complexes.

L'ensemble du traitement est automatisé et transparent pour l'utilisateur, qui récupère en sortie un document directement consultable et recherchable.

Apports et limites

Cette fonctionnalité améliore significativement l'exploitabilité des archives existantes, sans modifier leur structure visuelle. Elle facilite la recherche d'informations et contribue à une meilleure gestion documentaire.

Les principales limites identifiées concernent la dépendance à la qualité des scans d'origine ainsi que le temps de traitement pour les documents volumineux. Ces contraintes restent toutefois acceptables au regard du gain fonctionnel apporté.

11.2 Génération automatique d'une page de garde pour les documents PDF

Lors du stage, l'ajout d'une page de garde aux procès-verbaux d'assemblée générale avait été évoqué comme une simple suggestion, sans demande formalisée ni cadrage précis. L'idée était d'améliorer l'identification et la présentation de ces documents, dont le format variait selon leur provenance.

À partir de cette remarque, j'ai pris l'initiative de concevoir et de développer une fonctionnalité permettant d'ajouter automatiquement une page de garde en première page des procès-verbaux. L'utilisateur peut téléverser un document PDF et renseigner les informations essentielles, telles que le nom de la copropriété, la date de l'assemblée et le titre du document. Une page de garde est ensuite générée et fusionnée avec le document d'origine, sans en modifier le contenu.

La solution a été présentée au service Syndic, qui en a validé l'intérêt et l'a intégrée en production. Cette fonctionnalité permet aujourd'hui d'harmoniser la présentation des procès-verbaux et d'en améliorer la lisibilité.



12. Analyse réflexive des compétences

12.1 Bloc 1 : Identifier, modéliser et résoudre des problèmes informatiques

Lors de ce stage, j'ai démontré ma capacité à transformer un processus métier hétérogène et peu fiable en une solution numérique structurée. Initialement, la production de rapports au sein du service Syndic reposait sur l'utilisation de documents Word, ce qui engendrait une perte de temps considérable et une dispersion des données. En analysant ces contraintes, j'ai modélisé un domaine complexe capable de gérer différents types d'interventions (sinistres, visites d'immeubles, relevés d'index) tout en garantissant l'intégrité de l'information. Le choix stratégique d'une architecture NoSQL avec MongoDB et GridFS a été une réponse directe au besoin de stocker et de manipuler de gros volumes d'images et de fichiers PDF haute résolution directement en base de données. De plus, l'intégration de la technologie Tesseract OCR pour la récupération de texte sur des scans comptables illustre ma capacité à mobiliser des outils d'Intelligence Artificielle pour résoudre des problématiques d'extraction de données non structurées.

12.2 Bloc 2 : Concevoir, développer, mettre au point et conduire des projets d'application informatique

La conception et le développement de l'application FL copro m'ont permis de maîtriser l'intégralité du cycle de création logicielle, de la spécification à l'exploitation. J'ai mis en œuvre une architecture full-stack modulaire utilisant Angular pour le frontend et Node.js pour le backend, en exploitant TypeScript pour garantir la robustesse et le typage des données. Une réalisation clé a été la conception d'un éditeur de rapports "sur mesure" basé sur les formulaires réactifs d'Angular, offrant une interface flexible aux gestionnaires tout en conservant une structure de données rigide indispensable à la génération automatique de PDF professionnels via la bibliothèque pdf-lib. L'attention portée à l'ergonomie mobile répond à l'exigence de concevoir des interfaces adaptées aux utilisateurs finaux travaillant directement sur le terrain.

12.3 Bloc 3 : Concevoir, piloter, organiser, optimiser et gérer des systèmes d'information

Ce projet m'a conduit à concevoir un système d'information complet, incluant le développement de services distribués et l'administration de bases de données. J'ai analysé et conçu l'ensemble du système en traduisant les besoins du tuteur métier en spécifications techniques et fonctionnelles précises. La mise en place d'une politique organisationnelle rigoureuse en termes de sécurité, notamment par l'usage de JSON Web Tokens (JWT) et d'un contrôle d'accès basé sur les rôles (RBAC), garantit que les données sensibles des résidents sont protégées contre les accès non autorisés. Enfin, les études de cas réalisées après le déploiement démontrent ma capacité à auditer le système existant et à préconiser des évolutions basées sur les retours réels des utilisateurs pour optimiser les performances et l'UX.

12.4 Bloc 4 : Concevoir, mettre en œuvre, sécuriser et maintenir en condition opérationnelle des infrastructures réseaux

L'hébergement de la solution sur un VPS sous Debian 13 m'a permis d'appréhender les conditions réelles de fonctionnement et de sécurité d'un réseau. J'ai configuré un reverse proxy Nginx pour gérer les flux HTTPS et assurer la sécurisation des échanges via TLS avec Let's Encrypt. La gestion du domaine et de ses services associés (serveur de fichiers via GridFS, API REST) a nécessité une compréhension fine des mécanismes de connexion sécurisée et de la gestion des ports via un pare-feu. Cette approche systémique de la sécurité garantit la maintenance en condition opérationnelle de l'infrastructure tout en minimisant les risques d'intrusion.

12.5 Bloc 5 : Elaborer, concevoir et mener des stratégies d'intégration et d'évolution des installations matérielles et logicielles

La réussite du projet a reposé sur ma capacité à assembler et intégrer des composants logiciels disparates au sein d'une chaîne de valeur cohérente. J'ai mis en place une stratégie d'intégration continue via GitHub Actions, permettant d'automatiser le build et le déploiement sur le serveur de production à chaque tag de version. L'intégration de librairies spécifiques pour la génération de PDF et le parallélisme nécessaire au traitement OCR démontre une compréhension des mécanismes de gestion des ressources système. Cette démarche d'évolution constante a permis de livrer un produit fini capable de supporter la charge de documents volumineux tout en restant réactif.

12.6 Bloc 6 : Définir, planifier, organiser et manager un projet d'ingénierie innovant et collaboratif

En travaillant en autonomie mais en étroite collaboration avec les services Syndic et Gestion Locative, j'ai piloté ce projet selon une approche agile par sprints de trois semaines. L'utilisation de ClickUp pour la gestion du backlog et la définition d'une "Definition of Done" rigoureuse illustrent ma capacité à planifier et organiser les ressources humaines et temporelles. J'ai également mené une ANALYSE DE LA VALEUR APPORTEE en estimant le gain financier mensuel à environ **4 800€** justifiant ainsi l'investissement technologique par une approche économique pragmatique.

12.7 Bloc 7 : Questionner, analyser et adopter une démarche systémique ouverte et responsable

Enfin, j'ai adopté une démarche réflexive sur l'impact de mon travail au sein de FL IMMO. J'ai identifié et analysé les enjeux liés à la transformation numérique d'un métier traditionnellement bureautique, en veillant à respecter l'éthique et la confidentialité des données personnelles manipulées. Ma pratique de l'ingénierie s'est inscrite dans le respect des valeurs sociétales,

Arnaud Michel – Février 2026

notamment en cherchant à améliorer la qualité de vie au travail des collaborateurs par la réduction des tâches répétitives et frustrantes liées à la mise en page sous Word. Cette compréhension interdisciplinaire, centrée sur l'humain et les processus métiers réels, a été le socle de l'adoption réussie du logiciel par les équipes.



13. Conclusion

13.1 Synthèse des résultats

L'objectif principal du stage était de concevoir, développer et déployer une application web métier destinée au service Syndic, permettant de produire des rapports de copropriété de manière structurée, sécurisée et exploitable en conditions réelles. Cet objectif a été atteint par la mise en production effective de l'application FL copro, aujourd'hui utilisée par le service pour la réalisation de visites d'immeuble, de constats et de suivis techniques.

Le premier objectif opérationnel consistait à réduire le temps de production des rapports. Le passage d'un processus bureaucratique (Word) à une application structurée a permis de supprimer une grande partie des tâches de mise en forme manuelle, d'automatiser l'intégration des photographies et de standardiser la structure des documents. Les retours utilisateurs ont mis en évidence un gain de temps significatif entre l'intervention terrain et la production du PDF final.

Le second objectif portait sur la qualité et l'homogénéité documentaire. Grâce à des modèles de rapports structurés et à des champs guidés, les documents produits sont désormais cohérents d'un gestionnaire à l'autre, facilitant leur lecture, leur archivage et leur valeur probante. La finalisation verrouillée des rapports constitue une preuve tangible de l'intégrité documentaire.

Enfin, l'objectif de mise en production sécurisée a été validé par le déploiement sur un VPS, l'implémentation d'un contrôle d'accès RBAC, la gestion des sessions par JWT et la mise en place de sauvegardes. La disponibilité continue de l'application et son utilisation réelle par le service Syndic constituent la preuve finale de la réussite du projet.

13.2 Analyse critique

Avec le recul, le projet a démontré la pertinence d'une approche itérative fortement ancrée dans le métier. Les validations régulières ont permis d'éviter des développements inutiles et de prioriser les fonctionnalités réellement utilisées. En revanche, cette proximité métier a parfois entraîné des ajustements tardifs, notamment sur l'ergonomie mobile et la gestion des pièces jointes, générant du rework.

Sur le plan technique, les choix d'architecture (stack TypeScript, MongoDB, GridFS, génération PDF côté serveur) se sont révélés adaptés au contexte. Certaines limites restent néanmoins identifiées, comme l'absence d'antivirus serveur pour les uploads ou l'absence de tests automatisés exhaustifs, points assumés dans le cadre d'un projet de stage mais à considérer pour une industrialisation.

13.3 Estimation du gain financier et de la valeur apportée

L'un des objectifs du projet était d'apporter une valeur mesurable au service Syndic, au-delà de la seule amélioration ergonomique. Cette valeur peut être estimée à travers le gain de temps généré par la dématérialisation et la standardisation de la production des rapports de copropriété.

Situation avant le projet

Avant la mise en place de l'application FL copro, les rapports étaient intégralement réalisés sous Word. Cette méthode présentait plusieurs limites opérationnelles : une mise en page chronophage, une gestion complexe des images (insertion, redimensionnement, alignement), des documents hétérogènes selon les collaborateurs, ainsi que des redondances lors de la reprise d'informations existantes. En pratique, une part significative du temps de travail était consacrée à la forme du document plutôt qu'au contenu métier.

Hypothèses de calcul

Afin d'estimer le gain financier, les hypothèses suivantes ont été retenues :

- Temps moyen gagné par rapport : environ 3 heures, grâce à la saisie structurée, à la gestion intégrée des images et à la génération automatique du PDF.
- Coût horaire moyen d'un gestionnaire de copropriété : 100 € / heure (charges comprises).
- Volume actuel de rapports : 64 rapports par an, soit environ 5,3 rapports par mois (6 mois d'ancienneté du syndic).
- Capacité cible du service Syndic : 192 rapports par an, soit environ 16 rapports par mois, correspondant à un triplement du nombre de copropriétés gérées.

Estimation du gain financier

Dans la situation actuelle, le gain mensuel peut être estimé de la manière suivante :

$$\text{Gain mensuel} \approx (\text{Temps gagné par rapport} \times \text{Nombre de rapports par mois}) \times \text{Coût horaire}$$

$$\text{Gain mensuel} \approx (3 \text{ h} \times 5 \text{ rapports}) \times 100 \text{ €}$$

$$\text{Gain mensuel} \approx 1\,500 \text{ €}$$

Lorsque le service Syndic atteindra sa capacité nominale, avec un volume estimé à 16 rapports par mois, le gain deviendra:

$$\text{Gain mensuel} \approx (3 \text{ h} \times 16 \text{ rapports}) \times 100 \text{ €}$$

$$\text{Gain mensuel} \approx 4\,800 \text{ €}$$

Ces montants représentent uniquement le gain direct lié à la production des rapports. À l'échelle annuelle, cela correspond à un gain compris entre 18 000 € et 57 600 €, selon le niveau d'activité du service.

Valeur ajoutée non directement quantifiable

À ces gains chiffrés s'ajoutent des bénéfices indirects plus difficiles à quantifier, mais néanmoins significatifs. La fonctionnalité d'OCR permet notamment de transformer des documents comptables anciens ou scannés (grands livres, états financiers) en fichiers exploitables par recherche textuelle. Le temps gagné lors de la consultation, de la vérification ou de la préparation des Assemblées Générales n'a pas été intégré dans le calcul précédent, mais représente un gain de productivité réel pour les gestionnaires.

Par ailleurs, la standardisation des rapports améliore la qualité perçue des documents transmis aux copropriétaires, réduit les risques d'erreur ou d'oubli, et renforce la traçabilité des interventions. Ces éléments contribuent à la professionnalisation du service Syndic et à la sécurisation de son activité, ce qui constitue une valeur ajoutée stratégique pour l'entreprise FL IMMO.

13.4 Perspectives

Les perspectives d'évolution sont nombreuses : intégration avec des logiciels tiers de gestion immobilière, amélioration de l'OCR avec indexation plein texte, mise en place d'un antivirus serveur, ajout de signatures électroniques ou encore déploiement d'un historique détaillé des actions (audit log). L'architecture actuelle permet d'envisager ces évolutions sans remise en cause majeure du socle existant.

Liste des figures

Figures rapport:

Fig. 1: Organigramme de l'entreprise	10
Fig. 2: Gantt prévisionnel	21
Fig. 3: Gantt réel	22
Fig. 4: Architecture niveau 1	26
Fig. 5: Use case	27
Fig. 6: Activité rapport	28
Fig. 7: Sequence Auth/RBAC	29
Fig. 8: Sequence Upload/GridFS	30
Fig. 9: Domain model	31
Fig. 10: Deploiement	32
Fig. 11: Ticket	44

Figures application:

Application 1: Dashboard	59
Application 2: Liste des copropriétés	59
Application 3: Détails copropriétés	60
Application 4: Modifier une copropriété.....	60
Application 5: Thèmes.....	61
Application 6: Liste des rapports.....	61
Application 7: Créer un rapport.....	62
Application 8: Rapport de visite	62
Application 9: Login	63
Application 10: Profile	63
Application 11: Gestion des utilisateurs	64
Application 12: Gestion des rôles.....	64
Application 13: Ajouter entête	65
Application 14: OCR.....	65
Application 15: Documentation	66

Ressources en ligne

- [ANGULAR](#) – Documentation officielle
- [TYPESCRIPT](#) – Documentation officielle
- [NODE.JS](#) – Documentation officielle
- [EXPRESS.JS](#) – Documentation officielle
- [MONGODB](#) – Documentation officielle
- [MONGODB GRIDFS](#) – Documentation officielle
- [JSON WEB TOKEN \(JWT\)](#) – Introduction et spécifications
- [BCRYPT](#) – Documentation officielle
- [PDF-LIB](#) – Librairie de génération PDF
- [NGINX](#) – Documentation officielle
- [GIT](#) – Documentation officielle
- [GITHUB](#) – Documentation et sécurité (Dependabot)
- [CLICKUP](#) – Outil de gestion de projet
- [TESSERACT OCR](#) – Documentation officielle

Bibliographie

- Design Patterns : Alexander Shvets. Dive Into Design Patterns. Refactoring.Guru, 2021.
- Qualité logicielle : MARTIN Robert C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.
- Sécurité : OWASP Foundation. OWASP Top 10:2021.
- OCR : SMITH Ray. An Overview of the Tesseract OCR Engine. ICDAR, 2007.

Annexes des captures d'écran de l'application

FL Copro

Copropriétés - Vue d'ensemble

Vue d'ensemble des copropriétés

9 Copropriétés	403 Résidents	2 Rapports ce mois
-------------------	------------------	-----------------------

APPLICATION 1: DASHBOARD

FL Copro

Copropriétés - Liste

Copropriétés

Toutes les copropriétés enregistrées

Filtres	Nom	Ville	Code	Ajouter
	Résidence du Parc Montrosier	13 avenue Montrosier, 75016 Paris	Code: CDM-001 • Créeé: 1895	
	Le Clos des Marronniers	5 rue des Marronniers, 69003 Lyon	Code: CDM-002 • Créeé: 2008	
	Résidence Cap Canabière	78 boulevard Canabière, 13001 Marseille	Code: CDM-003 • Créeé: 1920	
	Villa des Écuyers	3 impasse des Écuyers, 31000 Toulouse	Code: CDM-004 • Créeé: 1995	
	Résidence Le Miroir d'Eau	44 quai Voltaire, 59000 Lille	Code: CDM-005 • Créeé: 2004	
	Résidence du Rhin	18 allée des Tilleuls, 67000 Strasbourg	Code: CDM-006 • Créeé: 1999	
	Résidence Saint-Pierre	27 rue Sainte-Catherine, 75004 Paris	Code: CDM-007 • Créeé: 2015	
	Résidence Promenade des Anglais	110 Boulevard de la Promenade des Fleurs, 06000 Nice	Code: CDM-008 • Créeé: 1960	
	Résidence du Peyrou	2 place de Peyrou, 69002 Lyon	Code: CDM-009 • Créeé: 2001	

AM Arnaud Michel

APPLICATION 2: LISTE DES COPROPRIÉTÉS

Arnaud Michel – Février 2026

FL Copro

Copropriétés - Détail

Résidence du Parc Montrosier
Gestion de la copropriété Résidence du Parc Montrosier

Informations de la copropriété
Consultez et modifiez les informations générales de la copropriété.

Code: CDM-001
Année de construction: 1895
Rue: 13 avenue Montrosier
Ville: 75016 Paris, France

[Modifier les informations](#)



Rapports
Liste des rapports associés à cette copropriété.

Titre	Date	Type	Rédacteur	Status	Actions
Compte rendu immeuble	14/01/2026	Visite	Arnaud Michel	Brouillon	
Réception de travaux	26/01/2026	Réception de travaux	Arnaud Michel	Brouillon	

[+ Nouveau rapport](#)

Résidents
Liste des résidents (51)

Prénom	Nom	Email	Actif	Actions
Laura	Robert	lrobert@gmail.com	Actif	

[Import](#) [+ Ajouter un résident](#)

AM Arnaud Michel

APPLICATION 3: DÉTAILS COPROPRIÉTÉS

FL Copro

Copropriétés - Détail

Résidence du Parc Montrosier
Gestion de la copropriété Résidence du Parc Montrosier

Informations de la copropriété
Consultez et modifiez les informations générales de la copropriété.

Code: CDM-001
Année de construction: 1895
Rue: 13 avenue Montrosier
Ville: 75016 Paris, France

[Modifier les informations](#)



Edition
Modifier la copropriété

Nom de la copropriété *
Résidence du Parc Montrosier

Code *
CDM-001

Année de construction *
12 / 04 / 1895

Photo de la copropriété
Parcourez vos fichiers



Adresse
Rue *
13 avenue Montrosier

Code postal *
75016

Ville *
Paris

Pays *
France

[Ajouter une rue](#)

[Enregistrer](#)

AM Arnaud Michel

APPLICATION 4: MODIFIER UNE COPROPRIÉTÉ

Arnaud Michel – Février 2026

FL Copro

Copropriétés - Liste

Copropriétés
Toutes les copropriétés enregistrées

Filtres Nom Ville Code

Nouvelle Copropriété

Résidence du Parc Montrosier
13 avenue Montrosier, 75016 Paris
Code: CDM-001 • Créeé: 1895

Le Clos des Marronniers
5 rue des Marronniers, 69003 Lyon
Code: CDM-002 • Créeé: 2006

Résidence Cap Canebière
78 boulevard Cap Canebière, 13001 Marseille
Code: CDM-003 • Créeé: 1930

Villa des Écuyers
3 impasse des Écuyers, 31000 Toulouse
Code: CDM-004 • Créeé: 1955

Résidence Le Miroir d'Eau
44 quai Voltaire, 59000 Lille
Code: CDM-005 • Créeé: 2004

Résidence du Rhin
18 allée des Tilleuls, 67000 Strasbourg
Code: CDM-006 • Créeé: 1999

Résidence Saint-Pierre
27 rue Sainte-Catherine, 75004 Paris
Code: CDM-007 • Créeé: 2015

Résidence Promenade des Anglais
110 Boulevard de la Promenade des Fleurs, 06000 Nice
Code: CDM-008 • Créeé: 1960

Résidence du Peyrou
2 place du Peyrou, 69002 Lyon
Code: CDM-009 • Créeé: 2001

AM Arnaud Michel

APPLICATION 5: THÈMES

FL Copro

Rapports - Vue d'ensemble

Rapports
Consultez les rapports liés aux copropriétés. Par défaut, les rapports affichés sont ceux que vous avez rédigés. Utilisez les filtres et tris pour affiner la liste.

Uniquement mes rapports Type: Copropriété: Du: Au:

Tous Code/Nom... jj / mm / aaaa jj / mm / aaaa

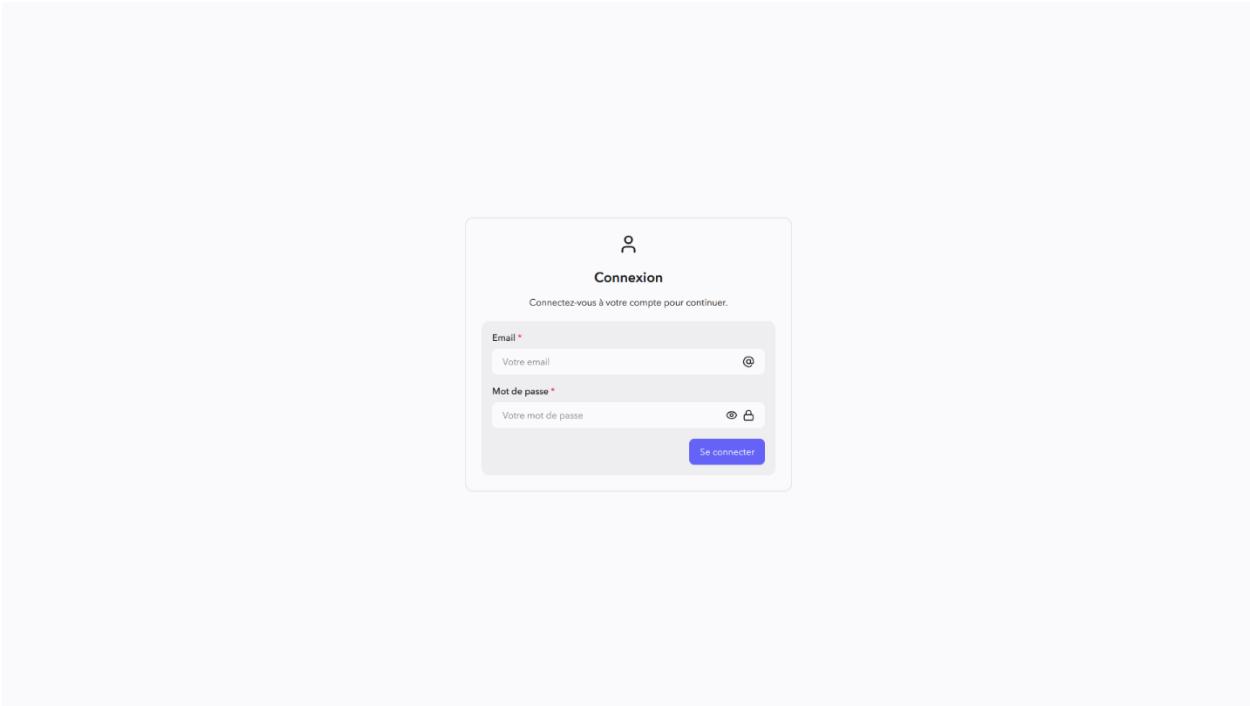
Date	Type	Immeuble	Statut	Auteur	Actions
30/01/2026	Sinistre	Le Clos des Marronniers	Finalisé	Arnaud Michel	
26/01/2026	Réception de travaux	Résidence du Parc Montrosier	Brouillon	Arnaud Michel	
18/01/2026	Relevé d'index	Résidence Le Miroir d'Eau	Finalisé	Arnaud Michel	
14/01/2026	Visite	Résidence Cap Canebière	Finalisé	Arnaud Michel	
12/01/2026	Visite	Résidence du Rhin	Finalisé	Arnaud Michel	

AM Arnaud Michel

APPLICATION 6: LISTE DES RAPPORTS

APPLICATION 7: CRÉER UN RAPPORT

APPLICATION 8: RAPPORT DE VISITE



APPLICATION 9: LOGIN

APPLICATION 10: PROFILE

Prénom	Nom	Email	Rôle	Actif	Crée le	Dernier login	Actions
Arnaud	Michel	arnaud.michel@utbm.fr	Admin	Actif	13/01/2026	30/01/2026:09:59	
Alpha	Pignon	a.pignon@prevot-immobilier.com	Gestionnaire	Actif	30/01/2026	30/01/2026:10:18	
Bravo	Brochant	b.brochant@prevot-immobilier.com	Gestionnaire	Inactif	30/01/2026	30/01/2026:10:18	
Charlie	Leblanc	c.leblanc@prevot-immobilier.com	Gestionnaire	Actif	30/01/2026	30/01/2026:10:19	
Delta	Chevalier	d.chevalier@prevot-immobilier.com	Gestionnaire	Inactif	30/01/2026	30/01/2026:10:19	

APPLICATION 11: GESTION DES UTILISATEURS

Permission	Admin	Gestionnaire
Gérer les utilisateurs Créer, modifier et supprimer des utilisateurs.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Gérer les rôles Créer et modifier les rôles et leurs permissions.	<input type="checkbox"/>	<input type="checkbox"/>
Gérer les paramètres Modifier les paramètres globaux de l'application.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Gérer les structures de Rapport Créer, modifier et archiver des structures de rapport.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Voir les rapports Accéder aux rapports et statistiques.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Gérer les copropriétés Créer, modifier et archiver des copropriétés.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

APPLICATION 12: GESTION DES RÔLES

FL Copro ↗

Page de garde PDF

Ajouter une page de garde à un PDF

Ajoutez une page de garde professionnelle au début de vos documents PDF.

Téléverser le fichier PDF *

Parcourez vos fichiers

PI_00_Projet_Bonvouloir
27.34 MB • Application

Titre de la page de garde *

Procès-verbal d'assemblée générale annuelle

Copropriété *

Résidence du Parc Montrosier (CDM-001)

Date *

30 / 01 / 2026

Generer le PDF avec page de garde

PRINCIPAL

- Accueil
- Documentation
- Problème ou suggestion

COPROPRIÉTÉS

- Vue d'ensemble
- Copropriétés
- Ajouter une copropriété

RAPPORTS

- Vue d'ensemble
- Créer un rapport
- Structure

ADMINISTRATION

- Utilisateurs
- Rôles
- Paramètres

DIVERS

- Extracteur PDF
- Page de garde

AM Arnaud Michel

APPLICATION 13: AJOUTER ENTÈTE

FL Copro ↗

PDF Upload

Reconnaissance de caractère sur PDF

Transformez vos documents PDF scannés en fichiers Word éditables grâce à la technologie OCR.

Parcourez vos fichiers

486.11 KB • Application

Langue(s) du document :

French

Mode haute précision (plus lent) :

Page 2/3

Conversion en cours...

À propos de l'OCR

- Taille maximale : 10 MB
- Fichiers acceptés : PDF uniquement
- Sortie : PDF converti avec le texte récupérable
- Conseil :** Pour de meilleurs résultats, utilisez des PDF de bonne qualité

AM Arnaud Michel

APPLICATION 14: OCR

The screenshot shows the FL Copro - Documentation 1.0.0 software interface. The left sidebar contains a navigation menu with sections like Aperçu, Authentification, Gestion des copropriétés, Gestion des rapports, Administration, Divers, and API Reference. The main content area has two sections: "Navigation" and "Principal". The "Navigation" section includes a detailed description of the menu structure. The "Principal" section contains tables for "Copropriétés" and "Documentation".

Navigation

Le menu principal de **FL copro** permet d'accéder rapidement aux différentes fonctionnalités du logiciel.
Voici une description détaillée de chaque section et de chaque entrée du menu.

Principal

Entrée	Description
Accueil	Accès à la page d'accueil du logiciel. Tableau de bord ou vue d'ensemble des informations importantes.
Problème ou suggestion	Ouvre directement la page des issues GitHub pour signaler un problème, proposer une amélioration ou suivre les demandes.
Documentation	Ouvre cette documentation.

Lien	Description
Vue d'ensemble	Synthèse globale des copropriétés : statistiques, indicateurs, tableaux.
Copropriétés	Liste complète des copropriétés enregistrées.

APPLICATION 15: DOCUMENTATION