المدرسة الوطنية للعلوم التطبيقية
ⵜⵜⵍⵎⵜⴰⵏⵓⵓⵜⵓⵙⴰⵍⵜⵙⵙⵙ

DÉDICACES

À mes très chers parents,

Pour leur amour inconditionnel, leurs sacrifices constants et leurs prières bienveillantes qui ont éclairé mon chemin tout au long de mes années d'études.
Puisse ce modeste travail être le témoignage de ma profonde gratitude et de mon éternel respect.

À mes frères et sœurs,

Pour leur soutien indéfectible, leurs encouragements permanents et leur présence chaleureuse dans les moments de doute comme dans les moments de réussite.

À mes enseignants et maîtres,

Qui m'ont guidé avec générosité et rigueur dans l'apprentissage du savoir et m'ont transmis la passion de l'ingénierie et de l'excellence.

À tous mes amis et camarades de promotion,

En souvenir des moments inoubliables partagés, des défis relevés ensemble et de la solidarité qui a marqué nos années de formation à l'ENSAF.

[Votre Prénom]

REMERCIEMENTS

Avant d'entamer le développement technique de ce rapport, il m'apparaît essentiel d'exprimer mes sincères remerciements à l'ensemble des personnes et des institutions dont la disponibilité, les conseils et la confiance ont permis la bonne réalisation et l'aboutissement de ce projet de fin d'année.

J'exprime tout d'abord ma profonde gratitude à l'organisme d'accueil **[Nom de l'organisme / Entreprise d'accueil]** pour m'avoir ouvert ses portes, accueilli au sein de ses structures et permis d'effectuer ce stage dans un cadre professionnel stimulant et propice à l'apprentissage.

Je tiens à témoigner ma vive et profonde reconnaissance à mes encadrants professionnels au sein de **[Nom de l'organisme / Entreprise d'accueil]** :

— **Dr. [Prénom & NOM Encadrant Entreprise 1]**, *Analyse de données & Ingénierie logicielle*, pour sa disponibilité constante, ses orientations méthodologiques et son accompagnement technique soutenu ;

Mes sincères remerciements vont également à mon encadrant académique à l'**École Nationale des Sciences Appliquées de Fès (ENSAF)**, **Prof. [Prénom & NOM Encadrant ENSAF]** pour son suivi pédagogique rigoureux, ses conseils méthodologiques et sa disponibilité tout au long du projet.

Je remercie chaleureusement l'ensemble de l'équipe technique, des ingénieurs et stagiaires de l'organisme pour leur coopération bienveillante, la richesse des échanges techniques et l'esprit de partage qui ont animé cette expérience professionnelle.

Mes remerciements s'adressent également au corps professoral et administratif du département **Génie Informatique** de l'ENSAF pour la qualité de la formation d'ingénieur d'État dispensée tout au long du cursus.

Enfin, j'adresse mes vifs remerciements aux honorables membres du jury pour l'honneur qu'ils me font en acceptant d'évaluer ce travail d'ingénierie et d'enrichir ce rapport par leurs observations avisées.

Membres du jury :

- **Prof. [Nom du Président du Jury]** — *Président (ENSAF)* ;
- **Prof. [Nom de l'Examinateur]** — *Examinateur (ENSAF)* ;
- **Dr. [Nom du Rapporteur / Invité]** — *Rapporteur / Membre invité (Entreprise d'accueil)* ;

RÉSUMÉ

Le présent rapport expose les travaux réalisés dans le cadre du Projet de Fin d'Année (PFA) au sein de **[Nom de l'organisme d'accueil]**. L'objectif principal de ce projet réside dans **[intitulé ou thématique générale du projet : ex. la conception et le développement d'une plateforme web distribuée / d'un système d'information / d'un module d'automatisation]**. Pour répondre aux défis identifiés — notamment **[décrire en une phrase la problématique clé : ex. l'optimisation des performances, la fiabilisation des flux de données ou la modernisation de l'architecture logicielle]** — une démarche rigoureuse d'ingénierie a été mise en œuvre.

La démarche méthodologique adoptée repose sur **[indiquer la méthodologie : ex. une approche itérative et agile combinant analyse des besoins, modélisation conceptuelle et développement modulaire]**. L'architecture développée s'appuie sur un ensemble de technologies modernes comprenant **[citer les technologies majeures : ex. Framework Front-End, Framework Back-End, Base de données et Outils de conteneurisation]**. Les résultats obtenus démontrent **[citer 2-3 résultats ou gains clés : ex. l'amélioration de l'efficacité des traitements, une couverture de tests complète et une expérience utilisateur fluide]**, répondant pleinement aux exigences fixées par le cahier des charges.

Mots-clés : [Mot-clé 1], [Mot-clé 2], [Mot-clé 3], [Mot-clé 4], [Mot-clé 5], Génie Logiciel, Architecture Système.

ABSTRACT

This report presents the engineering work conducted during the end-of-year internship (PFA) at **[Company / Host Organization Name]**. The project focuses on **[project title / core engineering topic : e.g., the design, development, and deployment of a distributed web application / an information management platform]**. To address the challenges identified within the organization — notably **[briefly describe the main problem : e.g., system scalability, process automation, or architectural modernization]** — a structured engineering approach was implemented.

The methodology applied is based on **[describe methodology : e.g., an agile iterative cycle integrating requirements engineering, architectural design, and modular implementation]**. The technological architecture leverages **[list key technologies : e.g., Frontend Framework, Backend Framework, Database Engine, and Containerization tools]**. The outcomes achieved demonstrate **[state 2-3 key achievements : e.g., enhanced processing efficiency, high test coverage, robust system reliability, and an intuitive user interface]**, successfully fulfilling all specifications.

Keywords : [Keyword 1], [Keyword 2], [Keyword 3], [Keyword 4], [Keyword 5], Software Engineering, System Architecture.

ملخص

يقدّم هذا التقرير وصفاً شاملاً لمشروع نهاية السنة المنجز لدى مؤسسة [/ Nom de l'organisme Entreprise d'accueil]، والمتمحور حول [TITRE DU PROJET : CONCEPTION ET RÉALISATION D'UNE SOLUTION D'INGÉNIERIE]. استجابةً للتحديات المرصودة، تم وضع وتنفيذ حلول هندسية متكاملة ومستدامة وفق أفضل المعايير التقنية المعمول بها.

اعتمدت المنهجية المتبعة على التحليل المنهجي للمتطلبات، والتصميم المعماري المعياري، مع اعتماد ممارسات التطوير الحديثة لضمان جودة الأداء وقابلية الصيانة والتوسع. وتُظهر النتائج المحققة نجاعة الحلول المقترحة ومطابقتها للمواصفات المسطرة.

كلمات مفتاحية : هندسة البرمجيات، بنية النظم، تطوير التطبيقات

LISTE DES ABRÉVIATIONS

Abréviation / Sigle	Signification Complète
API	Application Programming Interface
BDD / DBMS	Base de Données / Database Management System
CI/CD	Continuous Integration / Continuous Deployment
CSS	Cascading Style Sheets
ENSAF	École Nationale des Sciences Appliquées de Fès
HTML	HyperText Markup Language
HTTP / HTTPS	Hypertext Transfer Protocol / Secure
JSON	JavaScript Object Notation
MVC / MVVM	Model-View-Controller / Model-View-ViewModel
ORM	Object-Relational Mapping
PFA	Projet de Fin d'Année
PFE	Projet de Fin d'Études
REST	Representational State Transfer
SDK	Software Development Kit
SQL / NoSQL	Structured Query Language / Not Only SQL
UI / UX	User Interface / User Experience
UML	Unified Modeling Language

TABLE 1 – *Liste récapitulative des abréviations et sigles techniques*

TABLE DES FIGURES

1.1 Diagramme fonctionnel des flux du système 19
1.2 Diagramme de Gantt prévisionnel du déroulement du projet 21

2.1 Architecture globale du système proposé 25

3.1 Aperçu de l'interface du premier module développé 29
3.2 Aperçu de l'interface du deuxième module développé 29
3.3 Extrait de l'implémentation technique d'un composant métier 29
3.4 Synthèse graphique de l'amélioration des indicateurs de performance 30

LISTE DES TABLEAUX

1 Liste récapitulative des abréviations et sigles techniques 8

1.1 Domaines d'expertise de l'organisme d'accueil 18

1.2 Architecture technique du système existant 19

2.1 Synthèse comparative des choix technologiques 25

3.1 Tableau de synthèse des métriques de réalisation et de performance 30

A.1 Glossaire des concepts d'architecture et de méthodologie 37

A.2 Glossaire des technologies et outils de développement 37

TABLE DES MATIÈRES

Dédicaces **2**

Remerciements **3**

Résumé **5**

Abstract **6**

Résumé en langue arabe **7**

Liste des abréviations **8**

Introduction générale **14**

1 Cadre général du projet **16**

1.1 Introduction 17

1.2 Présentation de l'organisme d'accueil 17

1.2.1 Présentation générale et secteur d'activité 17

1.2.2 Positionnement et écosystème 17

1.2.3 Domaines d'expertise 17

1.2.4 Solutions et produits principaux 18

1.3 Analyse de l'existant 18

1.3.1 Contexte et périmètre du système existant 18

1.3.2 Architecture technique initiale 18

1.3.3 Acteurs et flux fonctionnels 19

1.4 Problématique et identification des besoins 19

1.5 Exposé de la mission confiée 20

1.5.1 Objectifs généraux 20

1.5.2 Périmètre du projet 20

1.5.3 Contraintes et critères d'évaluation 20

1.6 Planification temporelle et diagramme de Gantt 21

Conclusion 21

2 État de l'art et choix technologiques 22

2.1 Introduction 23

2.2 Approche méthodologique et organisation du projet 23

2.2.1 Cycle de développement et gestion de projet 23

2.2.2 Principes d'ingénierie logicielle appliqués 23

2.3 Étude comparative et sélection des technologies 23

2.3.1 Technologies Front-End (Interface Utilisateur) 23

2.3.2 Technologies Back-End (Logique Métier et API) 24

2.3.3 Gestion de la persistance des données 24

2.3.4 Synthèse des choix technologiques 24

2.4 Architecture globale et modélisation conceptuelle 25

2.4.1 Patron d'architecture logicielle 25

2.4.2 Modélisation des données et diagrammes de conception 25

2.5 Démarche d'assurance qualité et validation 26

Conclusion 26

3 Réalisation et analyse critique 27

3.1 Introduction 28

3.2 Environnement de développement et mise en place 28

3.2.1 Outils de développement et configuration 28

3.2.2 Organisation du projet et structure du code 28

3.3 Développement des modules principaux 28

3.3.1 Module 1 : [Nom du premier module / Interface principale] 28

3.3.2 Module 2 : [Nom du deuxième module / Traitement métier] 29

3.3.3 Implémentation technique et extrait de code 29

3.4 Résultats obtenus et indicateurs de performance 30

3.4.1 Indicateurs quantitatifs 30

3.4.2 Synthèse graphique de l'évolution 30

3.5 Discussion critique et évaluation des résultats 31

3.5.1 Adéquation aux objectifs et respect du cahier des charges 31

3.5.2 Difficultés rencontrées et solutions adoptées 31

3.5.3 Limites de la solution actuelle 31

Conclusion 31

Conclusion et perspectives **32**

A Glossaire des termes techniques **36**

INTRODUCTION GÉNÉRALE

Dans un contexte technologique en constante évolution, marqué par la digitalisation des processus et la complexification des systèmes d'information, la conception et le développement de solutions logicielles fiables, performantes et évolutives constituent un enjeu stratégique pour les entreprises. Ce projet de fin d'année (PFA), réalisé au sein de **[Nom de l'organisme d'accueil]**, spécialisé dans **[domaine d'activité : ex. l'ingénierie logicielle, les systèmes d'information, les télécommunications, etc.]**, s'inscrit dans cette dynamique.

Contexte et motivation :

Le projet répond à un besoin exprimé par l'organisme d'accueil concernant **[exposer le sujet : ex. la conception, le développement ou la modernisation d'une plateforme applicative spécifique]**. L'analyse initiale de l'environnement existant a permis d'identifier plusieurs défis majeurs : **[mentionner succinctement 2-3 défis : ex. limitation des fonctionnalités actuelles, nécessité d'optimisation des performances, automatisation des processus ou refonte architecturale]**.

Problématique :

Ce travail s'articule autour de la problématique centrale suivante :

« Comment concevoir, structurer et déployer une solution logicielle robuste et répondant aux exigences métier de [Nom de l'organisme], tout en garantissant la performance, la sécurité et la maintenabilité du système ? »

Objectifs assignés :

Pour apporter une réponse méthodique à cette problématique, les objectifs suivants ont été définis :

- Réaliser une étude préalable approfondie et spécifier le cahier des charges technique et fonctionnel ;
- Étudier l'état de l'art et sélectionner les technologies et architectures les plus adaptées ;
- Concevoir et modéliser l'architecture du système à travers les diagrammes appropriés ;
- Développer et intégrer les différents modules applicatifs selon les normes de qualité logicielle ;

— Effectuer les tests de validation, analyser les performances et valider les résultats obtenus.

Structure du rapport :

Le présent rapport est organisé en quatre chapitres :

— **Le Chapitre 1 (Cadre général du projet)** présente l'organisme d'accueil, analyse l'existant, définit le périmètre de la mission et détaille la planification temporelle via le diagramme de Gantt.

— **Le Chapitre 2 (État de l'art et choix technologiques)** expose l'approche méthodologique, l'étude comparative des outils et technologies sélectionnés, ainsi que les principes d'architecture logicielle.

— **Le Chapitre 3 (Réalisation et analyse critique)** décrit l'environnement de développement, détaille l'implémentation des modules réalisés, présente les résultats obtenus et propose une analyse critique.

— **Le Chapitre 4 (Conclusion et perspectives)** dresse le bilan global du projet, met en évidence les acquis d'ingénierie et trace les perspectives d'évolution à court, moyen et long terme.

1. Cadre général du projet

1.1 Introduction

Ce premier chapitre est consacré à la présentation du cadre général du projet de fin d'année. Nous introduisons tout d'abord l'organisme d'accueil, ses activités et ses pôles d'expertise. Nous menons ensuite une analyse méthodique de l'existant afin de cerner le contexte technique et opérationnel. Enfin, nous délimitons le périmètre précis de la mission confiée, ses objectifs, ses contraintes ainsi que la planification prévisionnelle illustrée par le diagramme de Gantt.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation générale et secteur d'activité

L'organisme d'accueil, **[Nom de l'organisme / Entreprise]**, est une structure spécialisée dans le domaine de **[secteur d'activité : ex. le développement logiciel, l'intégration de systèmes, la transformation digitale]**.

Fondée en **[Année]**, l'organisation propose des solutions innovantes adaptées aux besoins de ses clients et partenaires. Ses activités couvrent le conseil en technologies, la conception d'architectures logicielles et le déploiement de solutions métier.

1.2.2 Positionnement et écosystème

L'organisme collabore avec divers partenaires institutionnels et industriels **[citer des références : ex. entreprises partenaires, organismes publics, clients clés]**. Situé à **[Ville / Région]**, il dispose d'un positionnement stratégique lui permettant d'intervenir sur des projets à forte valeur ajoutée.

1.2.3 Domaines d'expertise

Les principales compétences et activités de l'organisme s'articulent autour de plusieurs axes complémentaires présentés dans le Tableau 1.1.

Ingénierie Logicielle	Conseil & Expertise	Innovation & R&D
• Conception d'architectures	• Audit et sécurité	• Veille technologique
• Développement Web & Mobile	• Gestion de projets IT	• Développement de PoC
• Intégration de systèmes	• Accompagnement digital	• Intelligence Artificielle

TABLE 1.1 – *Domaines d'expertise de l'organisme d'accueil*

1.2.4 Solutions et produits principaux

Dans le cadre de ses prestations, l'organisme conçoit et maintient différentes solutions logicielles :

— **[Solution / Plateforme 1]** : [Brève description de la solution principale ou du produit associé au projet] ;

— **[Solution / Plateforme 2]** : [Brève description d'autres produits développés au sein de l'entreprise].

1.3 Analyse de l'existant

1.3.1 Contexte et périmètre du système existant

L'intervention réalisée porte sur **[Nom du système / de l'application existante]**. Ce système a pour finalité de **[décrire la mission opérationnelle du système existant]**.

1.3.2 Architecture technique initiale

Le Tableau 1.2 résume la composition de la pile technologique initiale du système.

Composant / Couche	Technologies Utilisées	Rôle Opérationnel
Couche Présentation	[Technologies Front : ex. HTML/CSS, Framework UI]	Interface utilisateur et interactions
Couche Métier	[Technologies Back : ex. Langage, Framework API]	Logique métier et traitement des flux
Couche Persistance	[Système de BDD : ex. SQL / NoSQL]	Stockage et gestion des données
Infrastructure	[Outils : ex. Serveur web, Conteneurs, Outils CI]	Hébergement et exécution

TABLE 1.2 – *Architecture technique du système existant*

1.3.3 Acteurs et flux fonctionnels

Le fonctionnement de la plateforme implique différents rôles d'utilisateurs :

— **[Acteur 1 : ex. Administrateur]** : [Rôle, responsabilités et droits d'accès] ;

— **[Acteur 2 : ex. Gestionnaire / Utilisateur métier]** : [Opérations principales et tâches quotidiennes] ;

— **[Acteur 3 : ex. Client / Utilisateur final]** : [Consultation des informations et interactions avec le système].

[Insérer ici le diagramme de cas d'utilisation ou d'architecture fonctionnelle de l'existant]

FIGURE 1.1 – *Diagramme fonctionnel des flux du système*

1.4 Problématique et identification des besoins

L'analyse de l'existant et les échanges avec l'équipe encadrante ont permis de dégager plusieurs limitations :

1. **Limitations fonctionnelles** : [Décrire les manques fonctionnels ou processus manuels à automatiser] ;
2. **Contraintes techniques** : [Décrire les contraintes d'architecture, de scalabilité ou de performance] ;
3. **Besoins d'amélioration** : [Décrire les attentes prioritaires des utilisateurs et de l'organisme].

1.5 Exposé de la mission confiée

1.5.1 Objectifs généraux

Pour répondre aux besoins identifiés, la mission confiée dans le cadre de ce stage consiste à **[définir l'objectif majeur du projet : ex. concevoir et développer un module métier complet, refondre l'architecture applicative ou créer une nouvelle plateforme]**.

1.5.2 Périmètre du projet

— **Tâches et fonctionnalités incluses** :

Tâche 1 : ex. Analyse détaillée des besoins et rédaction des spécifications ;

Tâche 2 : ex. Conception de l'architecture logicielle et de la base de données ;

Tâche 3 : ex. Développement et intégration des composants Front-End et Back-End ;

Tâche 4 : ex. Mise en œuvre des tests de validation et déploiement .

— **Fonctionnalités hors périmètre** :

Exclusion 1 : Fonctionnalités secondaires reportées aux versions futures ;

Exclusion 2 : Intégrations matérielles ou déploiements externes hors portée .

1.5.3 Contraintes et critères d'évaluation

Le projet est soumis aux contraintes suivantes :

— **Délai d'exécution** : Respect de la durée impartie pour le stage ([nombre] semaines) ;

— **Qualité et sécurité** : Respect des standards de codage, de sécurité des données et de modularité ;

— **Maintenabilité** : Production d'un code documenté, testable et aisément réutilisable.

1.6 Planification temporelle et diagramme de Gantt

Afin de structurer le déroulement du projet et d'assurer le respect des délais, un planning prévisionnel a été établi. La Figure 1.2 présente la répartition temporelle des principales phases du projet.

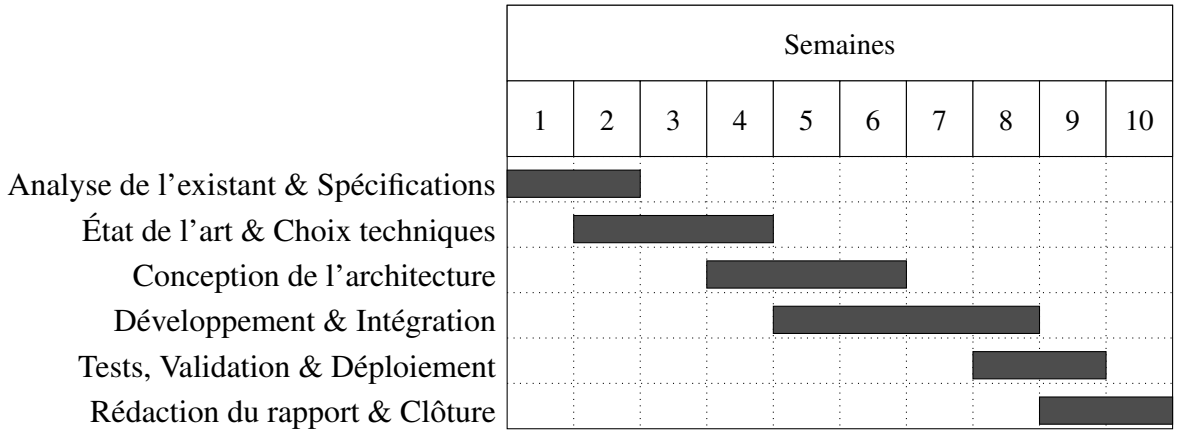


FIGURE 1.2 – *Diagramme de Gantt prévisionnel du déroulement du projet*

Conclusion

Ce premier chapitre a présenté l'organisme d'accueil, analysé l'existant et défini les objectifs et la planification du projet. Le deuxième chapitre est consacré à l'étude de l'état de l'art, à la démarche méthodologique et à la justification des choix technologiques et architecturaux.

2. État de l’art et choix technologiques

2.1 Introduction

La réussite d'un projet d'ingénierie logicielle repose sur l'adoption d'une démarche méthodologique rigoureuse et le choix d'un écosystème technologique adapté aux exigences du cahier des charges. Ce deuxième chapitre est dédié à la présentation de l'approche méthodologique retenue, à l'étude comparative des différentes technologies envisageables, ainsi qu'à la conception architecturale globale du système à développer.

2.2 Approche méthodologique et organisation du projet

2.2.1 Cycle de développement et gestion de projet

Pour mener à bien la réalisation de ce projet, une méthodologie de développement **[nom de la méthodologie : ex. Agile / Scrum / Cycle itératif]** a été adoptée. Cette démarche privilégie :

— L'itération continue et la validation régulière des fonctionnalités développées ;

— La flexibilité face aux ajustements éventuels des besoins du projet ;

— Une communication fluide et régulière avec les encadrants académique et professionnel.

2.2.2 Principes d'ingénierie logicielle appliqués

Le développement de la solution respecte les principes fondamentaux de qualité logicielle :

— **Séparation des préoccupations (Separation of Concerns)** : Découplage net entre la présentation, la logique métier et l'accès aux données ;

— **Modularité et réutilisabilité** : Conception de composants indépendants, facilement testables et maintenables ;

— **Gestion rigoureuse des versions** : Utilisation de Git avec une convention de branches claire (`main`, `develop`, `feature/*`).

2.3 Étude comparative et sélection des technologies

2.3.1 Technologies Front-End (Interface Utilisateur)

Pour la réalisation de la couche interface, une comparaison entre différents frameworks a été menée :

— **[Framework 1 : ex. Angular / React / Vue.js]** : [Avantages, inconvénients et spécificités] ;

— **[Framework 2 : ex. Solution alternative]** : [Comparaison et limites].

Choix retenu : [Nom de la technologie front-end retenue], en raison de **[justification : ex. son architecture composant, sa robustesse, sa performance]**.

2.3.2 Technologies Back-End (Logique Métier et API)

Pour le développement de la logique serveur et de l'API REST :

— **[Technologie 1 : ex. Spring Boot / Node.js / Django]** : [Forces, écosystème et facilité de prise en main] ;

— **[Technologie 2 : ex. Solution alternative]** : [Comparaison technique].

Choix retenu : [Nom de la technologie back-end retenue], justifiée par **[justification : ex. son écosystème riche, sa sécurité intégrée et sa stabilité]**.

2.3.3 Gestion de la persistance des données

Le stockage des données nécessite un choix adapté à la nature des informations manipulées :

— **Bases de données relationnelles (SQL : ex. PostgreSQL / MySQL)** : Gestion rigoureuse des transactions ACID et intégrité référentielle ;

— **Bases de données NoSQL (ex. MongoDB / Redis)** : Flexibilité de schéma et gestion de volumes ou de caches rapides.

Choix retenu : [Nom du SGBD retenu], répondant aux besoins de **[justification du choix]**.

2.3.4 Synthèse des choix technologiques

Le Tableau 2.1 synthétise les technologies sélectionnées pour l'ensemble du projet.

Composant	Technologie Retenue	Avantages Principaux	Critère Décisif
Front-End	[Technologie Front]	[Modularité, réactivité, composants]	Ergonomie UI
Back-End	[Technologie Back]	[Performance, sécurité, API RESTful]	Fiabilité métier
Base de Données	[SGBD Retenu]	[Intégrité, performance, flexibilité]	Cohérence données
Environnement	[Outils : ex. Docker, Git]	[Portabilité, gestion de versions]	Reproductibilité

TABLE 2.1 – *Synthèse comparative des choix technologiques*

2.4 Architecture globale et modélisation conceptuelle

2.4.1 Patron d'architecture logicielle

L'architecture adoptée s'articule autour d'un modèle en couches (**[ex. N-Tiers / Microservices / MVC]**), permettant d'isoler clairement les responsabilités. La Figure 2.1 illustre l'organisation des modules et les flux de communication.

[Insérer ici le schéma d'architecture globale / modèle en couches du système]

FIGURE 2.1 – *Architecture globale du système proposé*

2.4.2 Modélisation des données et diagrammes de conception

La conception du modèle de données repose sur la modélisation UML (diagramme de classes ou modèle relationnel) détaillant les entités clés, leurs attributs et leurs associations.

2.5 Démarche d'assurance qualité et validation

Pour garantir la qualité des livrables et la robustesse du code :

— **Stratégie de tests** : Tests unitaires pour valider les fonctions métier critiques, tests d'intégration pour vérifier les points de terminaison d'API ;

— **Conventions de codage** : Respect des règles de nommage, linters et revue systématique du code ;

— **Sécurité** : Validation rigoureuse des entrées, gestion des habilitations et sécurisation des communications.

Conclusion

Ce deuxième chapitre a détaillé la méthodologie, les choix technologiques et les choix d'architecture logicielle. Le troisième chapitre présente la mise en œuvre pratique, les réalisations techniques développées ainsi que l'analyse des résultats obtenus.

3. Réalisation et analyse critique

3.1 Introduction

Ce troisième chapitre est consacré à la présentation concrète des réalisations techniques effectuées au cours du projet. Il expose l'environnement de développement mis en place, détaille le développement et l'intégration des principaux modules applicatifs, présente les résultats et indicateurs de performance obtenus, et propose une analyse critique des résultats et des choix opérés.

3.2 Environnement de développement et mise en place

3.2.1 Outils de développement et configuration

La phase initiale a consisté à préparer l'environnement de travail et les outils de développement :

- **Environnement d'exécution** : Configuration des runtimes, gestionnaires de paquets et conteneurs ;
- **Gestionnaire de versions** : Initialisation du dépôt Git et configuration des flux de travail collaboratifs ;
- **Base de données locale** : Déploiement et initialisation du schéma de données de développement.

3.2.2 Organisation du projet et structure du code

Le code source est structuré de manière modulaire afin de faciliter la maintenance et la collaboration, en séparant nettement les composants d'interface, les services métier, les modèles de données et la configuration de l'application.

3.3 Développement des modules principaux

3.3.1 Module 1 : [Nom du premier module / Interface principale]

Ce module a pour rôle de **[description du module : ex. la gestion de l'authentification et du tableau de bord utilisateur]**. La Figure 3.1 présente une capture d'écran de l'interface développée.

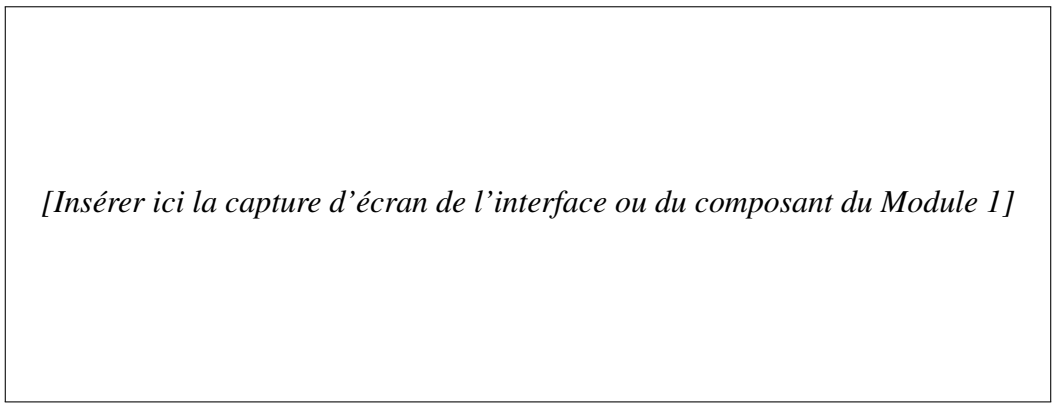


FIGURE 3.1 – *Aperçu de l'interface du premier module développé*

3.3.2 Module 2 : [Nom du deuxième module / Traitement métier]

Ce module assure **[description du rôle du module : ex. le traitement des données, la gestion des requêtes ou les opérations CRUD métier]**. La Figure 3.2 illustre l'interface ou le flux associé.

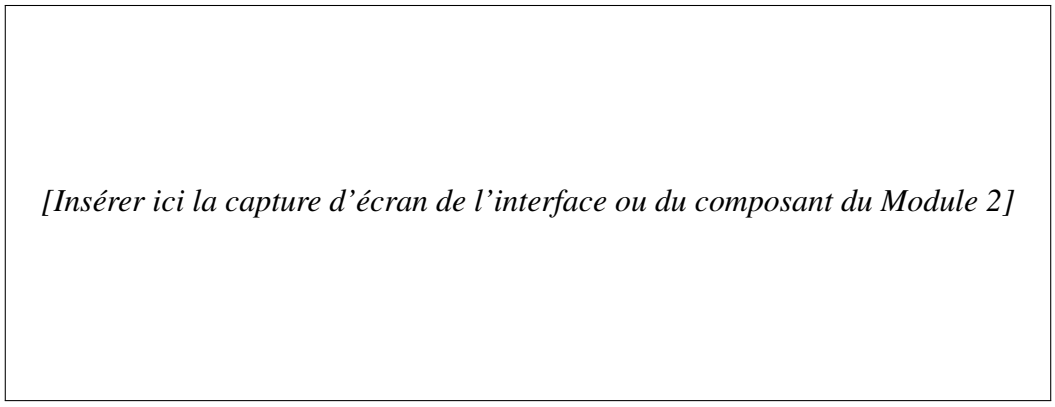


FIGURE 3.2 – *Aperçu de l'interface du deuxième module développé*

3.3.3 Implémentation technique et extrait de code

Le Listing ci-dessous donne un exemple de structure ou de service implémenté au sein de l'application :

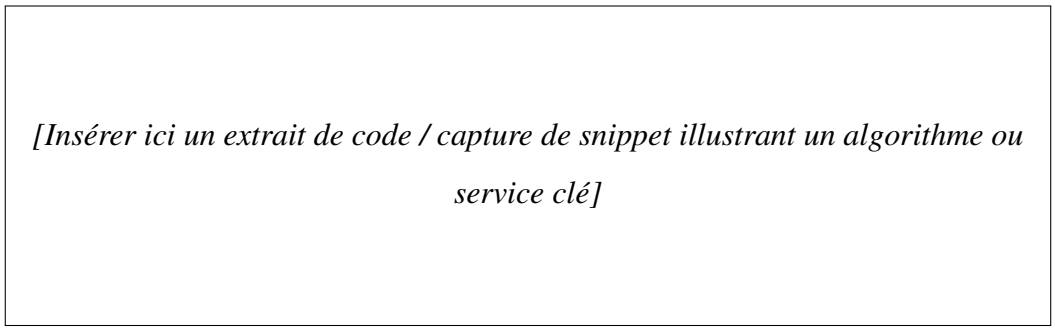


FIGURE 3.3 – *Extrait de l'implémentation technique d'un composant métier*

3.4 Résultats obtenus et indicateurs de performance

3.4.1 Indicateurs quantitatifs

Afin d'évaluer objectivement l'impact des travaux réalisés, plusieurs métriques ont été mesurées avant et après la réalisation du projet. Le Tableau 3.1 présente la synthèse de ces indicateurs.

Indicateur d'Évaluation	**Situation Initiale**	**Résultat Obtenu**	**Gain / Évolution**	**Unité**
Couverture fonctionnelle	[Valeur init.]	[Valeur finale]	[+X %]	% du cahier
Temps de réponse moyen	[Valeur init.]	[Valeur finale]	[-X %]	Millisecondes
Couverture de tests	[0 %]	[Valeur finale]	[+X %]	% du code
Stabilité des déploiements	Instable	100 %	Résolu	Statut

TABLE 3.1 – *Tableau de synthèse des métriques de réalisation et de performance*

3.4.2 Synthèse graphique de l'évolution

La Figure 3.4 met en perspective l'évolution des indicateurs clés tout au long du projet.

[Insérer ici le graphique synthétique d'évolution des résultats / métriques]

FIGURE 3.4 – *Synthèse graphique de l'amélioration des indicateurs de performance*

3.5 Discussion critique et évaluation des résultats

3.5.1 Adéquation aux objectifs et respect du cahier des charges

L'ensemble des fonctionnalités développées répond aux exigences fonctionnelles et techniques spécifiées lors de la phase de cadrage. La solution déployée apporte des gains substantiels en termes d'efficacité opérationnelle et de convivialité.

3.5.2 Difficultés rencontrées et solutions adoptées

Au cours de la réalisation, plusieurs défis ont été surmontés :

— **[Difficulté 1 : ex. Intégration entre différents services]** : [Solution mise en œuvre pour la résoudre] ;

— **[Difficulté 2 : ex. Optimisation des temps de réponse de requêtes]** : [Démarche d'optimisation appliquée].

3.5.3 Limites de la solution actuelle

Bien que les résultats soient conformes aux attentes, certaines limites méritent d'être soulignées :

— **[Limite 1]** : [Description d'une contrainte technique ou fonctionnelle actuelle] ;

— **[Limite 2]** : [Points nécessitant des améliorations futures].

Conclusion

Ce troisième chapitre a détaillé les réalisations techniques, les résultats obtenus et leur analyse critique. Le chapitre suivant conclut le présent rapport en dressant le bilan général des acquis et en dégageant les perspectives d'évolution.

CONCLUSION ET PERSPECTIVES

Ce projet de fin d'année effectué au sein de **[Nom de l'organisme d'accueil]** a constitué une expérience particulièrement enrichissante et professionnalisante. Il a permis de mettre en application et de consolider les compétences théoriques et pratiques acquises durant la formation d'ingénieur à l'ENSAF, tout en se confrontant aux exigences réelles d'un environnement professionnel.

Bilan synthétique des réalisations :

Les objectifs fixés au début du stage ont été atteints avec succès :

- **Cadrage et analyse** : Réalisation d'une étude préalable détaillée de l'existant et spécification précise du cahier des charges ;
- **Conception et architecture** : Choix raisonné d'une pile technologique moderne et modélisation d'une architecture logicielle modulaire et évolutive ;
- **Implémentation et intégration** : Développement des différents modules applicatifs dans le respect des bonnes pratiques de génie logiciel ;
- **Validation et qualité** : Mise en place de protocoles de tests garantissant la stabilité, la sécurité et la performance de la solution livrée.

Acquis méthodologiques et personnels :

Au-delà des livrables techniques, ce projet a permis de développer des compétences clés :

- **Rigueur d'ingénierie** : Gestion rigoureuse des versions de code, documentation technique structurée et respect des normes de développement ;
- **Autonomie et résolution de problèmes** : Capacité à diagnostiquer méthodiquement des anomalies complexes et à formuler des solutions adaptées ;
- **Travail collaboratif** : Intégration active au sein de l'équipe technique et communication régulière sur l'avancement des tâches.

Perspectives d'évolution :

Pour poursuivre et valoriser les résultats de ce travail, plusieurs perspectives d'amélioration peuvent être envisagées :

- **À court terme** : Étendre la couverture des tests automatisés et intégrer l'application

dans un pipeline d'intégration et de déploiement continus (CI/CD) ;

— **À moyen terme** : Développer de nouveaux modules fonctionnels répondant à des besoins métier émergents ;

— **À long terme** : Optimiser le passage à l'échelle sur une infrastructure cloud et explorer l'intégration de fonctionnalités intelligentes basées sur l'analyse de données.

Bilan personnel :

Cette expérience au sein de **[Nom de l'organisme]** a permis de consolider les compétences techniques en génie informatique et constitue une assise solide pour aborder la suite du cursus et la future carrière d'ingénieur d'État.

BIBLIOGRAPHIE & SITOGRAPHIE

[1] ATLASSIAN. *Agile Coach : Comprehensive Guide to Scrum, Sprints, and Estimation*. 2026. URL : `https://www.atlassian.com/agile/scrum` (visité le 25/08/2026).

[2] Scott CHACON et Ben STRAUB. *Pro Git : Everything you need to know about Git*. 2nd. New York, NY, USA : Apress, 2014.

[3] Edsger W. DIJKSTRA. "On the Role of Scientific Thought : Separation of Concerns". In : *Selected Writings on Computing : A Personal Perspective* (1982), p. 60-66.

[4] DOCKER INC. *Docker Documentation : Architecture and Best Practices for Containerization*. 2026. URL : `https://docs.docker.com/` (visité le 25/08/2026).

[5] Roy T. FIELDING et Richard N. TAYLOR. "Principled Design of the Modern Web Architecture". In : *ACM Transactions on Internet Technology (TOIT)* 2.2 (2002), p. 115-150.

[6] Martin FOWLER. *Refactoring : Improving the Design of Existing Code*. 2nd. Boston, MA, USA : Addison-Wesley Professional, 2018.

[7] Erich GAMMA et al. *Design Patterns : Elements of Reusable Object-Oriented Software*. Reading, MA, USA : Addison-Wesley Professional, 1994.

[8] *ISO/IEC 25010 :2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. Geneva, Switzerland : International Organization for Standardization, 2011.

[9] Avraham LEFF et James T. RAYFIELD. "Web-Application Development Using the Model/View/Controller Design Pattern". In : *Proceedings of the Fifth IEEE International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE. 2001, p. 118-127.

[10] Robert C. MARTIN. *Clean Code : A Handbook of Agile Software Craftsmanship*. Upper Saddle River, NJ, USA : Prentice Hall, 2008.

[11] Roger S. PRESSMAN et Bruce R. MAXIM. *Software Engineering : A Practitioner's Approach*. 8th. New York, NY, USA : McGraw-Hill Education, 2014.

[12] Ian SOMMERVILLE. *Software Engineering*. 10th. Boston, MA, USA : Pearson Education, 2016.

[13] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL 16 Documentation : Relational Database Management System*. 2026. URL : https://www.postgresql.org/docs/ (visité le 25/08/2026).

A. Glossaire des termes techniques

Ce glossaire récapitule les définitions des principaux termes techniques, concepts architecturaux et acronymes employés dans le rapport :

Architecture et méthodologies

Terme / Concept	Définition et Contexte
API REST	Interface de programmation applicative reposant sur le protocole HTTP, permettant la communication découplée entre client et serveur.
CI/CD	Pratiques d'intégration et de déploiement continus automatisant le cycle de vie logiciel (build, tests, mise en production).
GitFlow	Modèle structuré de gestion des branches Git garantissant la stabilité du code en production.
MVC / MVVM	Patrons d'architecture logicielle séparant les données, la logique de traitement et la couche de présentation.
SoC	Séparation des préoccupations (*Separation of Concerns*), principe de conception visant à modulariser les responsabilités.

TABLE A.1 – *Glossaire des concepts d'architecture et de méthodologie*

Technologies et développement

Terme / Outil	Définition et Rôle Technique
Conteneurisation	Mécanisme d'isolation logicielle empaquetant le code et ses dépendances pour garantir la portabilité.
JWT	Jeton sécurisé d'authentification (*JSON Web Token*) transmettant des informations signées numériquement.
ORM	Outil faisant le pont entre les classes objets du code source et les tables relationnelles d'une base de données.
Swagger / OpenAPI	Outil de documentation interactive et de test standardisé des routes d'une API REST.

TABLE A.2 – *Glossaire des technologies et outils de développement*