ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

Projet de Fin d'Études

Banking System avec IA pour la détection de fraudes

Réalisé par :

Hamza ADDAMI
Mohammed Amine KHAZRAJ

Encadré par :

M. BOUCHTIB Hicham
Mme ———

Année scolaire : 2024 – 2025

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à Dieu Tout-Puissant pour m'avoir donné la force, la patience et la persévérance nécessaires à la réalisation de ce projet de fin d'études.

Je remercie chaleureusement mon encadrant, Monsieur BOUCHTIB Hicham, pour ses conseils pertinents, son accompagnement et sa disponibilité tout au long de ce projet. Son expertise et ses encouragements ont grandement contribué à l'avancement et à la réussite de ce travail.

Je souhaite également remercier l'équipe pédagogique de l'établissement [Nom de ton établissement] pour la qualité de la formation dispensée tout au long de mon parcours.

Mes sincères remerciements vont également à mes collègues, amis et membres de ma famille, pour leur aide, leur soutien moral et leur présence bienveillante.

Résumé

Dans un contexte où la transformation numérique touche tous les secteurs, le domaine bancaire est particulièrement exposé aux risques liés à la sécurité des transactions. Ce projet vise à développer une application bancaire intégrant un module de détection de fraude basé sur l'intelligence artificielle.

La solution, intitulée **Banking System**, permet la gestion des comptes, la consultation de l'historique, et la détection d'anomalies dans les transactions. Elle repose sur une architecture moderne combinant **Spring Boot** pour le backend et **React.js** pour le frontend, avec une base de données relationnelle PostgreSQL.

L'approche adoptée s'appuie sur la modélisation UML pour garantir une conception robuste, ainsi que sur des algorithmes de machine learning pour détecter les comportements suspects en temps réel.

Mots-clés : Banque, Sécurité, Détection de fraude, Intelligence artificielle, Spring Boot, React.js, UML.

Table des matières

Chapitre 1

Contexte général du projet

1.1 Introduction

Dans un contexte de transformation digitale accélérée, les institutions financières doivent innover continuellement pour offrir des services rapides, accessibles et sécurisés. Les services bancaires en ligne sont devenus incontournables pour les clients, transformant radicalement la manière dont les opérations financières sont effectuées au quotidien. Cette évolution numérique, bien qu'apportant une commodité sans précédent, s'accompagne également d'une recrudescence alarmante des menaces de fraude financière.

Les statistiques récentes révèlent l'ampleur du problème : selon l'Association des Banquiers Américains, les pertes dues aux fraudes bancaires ont atteint plus de 28 milliards de dollars en 2023, soit une augmentation de 37% par rapport à 2020. En Europe, l'Autorité Bancaire Européenne rapporte que les incidents de fraude liés aux paiements électroniques ont augmenté de 43% sur la même période. Ces menaces, de plus en plus sophistiquées et automatisées, exploitent les vulnérabilités des systèmes traditionnels de détection, souvent incapables de s'adapter rapidement à l'évolution des techniques frauduleuses.

Face à cette réalité, les méthodes conventionnelles de détection de fraude, principalement basées sur des règles statiques et des seuils prédéfinis, montrent leurs limites. Elles génèrent un nombre élevé de faux positifs, mobilisant inutilement des ressources précieuses, tout en laissant passer certaines activités frauduleuses sophistiquées. C'est dans ce contexte que l'intelligence artificielle (IA) et plus particulièrement le machine learning émergent comme des technologies prometteuses pour révolutionner la détection et la prévention des fraudes bancaires.

Ce projet de fin d'études s'inscrit précisément dans cette dynamique d'innovation, avec pour ambition de développer une application bancaire complète intégrant un moteur d'intelligence artificielle capable de détecter les fraudes en temps réel, tout en minimisant les faux positifs et en garantissant une expérience utilisateur optimale.

1.2 Contexte du projet

Le développement des services financiers numériques a profondément modifié les comportements des utilisateurs et transformé l'écosystème bancaire mondial. Aujourd'hui, la majorité des opérations bancaires (consultation de solde, virements, paiements en ligne) se font sans interaction physique, à travers des applications mobiles ou des plateformes web. D'après la Banque des Règlements Internationaux, les paiements numériques ont doublé entre 2018 et 2023, atteignant plus de 1,5 trillion de transactions annuelles, conséquence directe de la digitalisation accélérée post-COVID.

Cette transformation numérique s'est accompagnée d'une évolution parallèle des méthodes de fraude. Les cybercriminels ont développé des techniques de plus en plus sophistiquées, allant

du phishing ciblé aux attaques par intelligence artificielle, en passant par l'usurpation d'identité biométrique. Selon le rapport mondial sur la cybersécurité de Cybersecurity Ventures, les coûts globaux liés à la cybercriminalité devraient atteindre 10,5 trillions de dollars annuellement d'ici 2025, avec le secteur financier comme cible privilégiée.

Les établissements bancaires se trouvent donc confrontés à un double défi : d'une part, répondre aux attentes croissantes des clients en matière de services numériques fluides et instantanés ; d'autre part, renforcer leurs dispositifs de sécurité pour contrer des menaces en constante évolution. Cette équation complexe ne peut être résolue par les approches traditionnelles de détection de fraude, qui présentent plusieurs limitations :

1. **Réactivité limitée** : Les systèmes basés sur des règles détectent les fraudes après leur occurrence, souvent trop tard pour empêcher les pertes financières.
2. **Manque d'adaptabilité** : Ces systèmes peinent à s'ajuster rapidement aux nouvelles techniques de fraude, créant une fenêtre d'opportunité pour les fraudeurs.
3. **Taux élevé de faux positifs** : Les alertes erronées génèrent des coûts opérationnels importants et affectent négativement l'expérience client.
4. **Incapacité à traiter les volumes massifs de données** : Avec l'explosion des transactions numériques, les systèmes traditionnels atteignent leurs limites en termes de capacité de traitement.

C'est dans ce contexte que les technologies d'intelligence artificielle, et particulièrement le machine learning, émergent comme des solutions prometteuses. Leur capacité à analyser de vastes volumes de données, à identifier des patterns complexes et à s'adapter continuellement aux nouvelles menaces en fait des outils particulièrement adaptés à la détection de fraude moderne.

Notre projet s'inscrit précisément dans cette dynamique d'innovation, visant à développer une solution bancaire intégrant nativement ces technologies avancées pour une détection de fraude plus efficace et proactive.

1.3 Périmètre du projet

Le système développé dans le cadre de ce projet vise à fournir une plateforme bancaire en ligne complète, couplée à un module de détection de fraude basé sur des techniques d'apprentissage automatique. Le périmètre fonctionnel comprend :

- **Gestion des comptes bancaires** : Création, consultation, modification et suppression des comptes.
- **Opérations financières** : Réalisation de virements, retraits et dépôts, avec un suivi détaillé des transactions.
- **Détection de fraude par IA** : Mise en place d'un modèle d'apprentissage supervisé (Machine Learning) pour l'analyse en temps réel des transactions suspectes.
- **Interfaces différenciées** : Deux espaces distincts, l'un pour les clients (opérations et historique), l'autre pour les administrateurs (monitoring, alertes, statistiques).
- **Sécurité et performances** : Authentification sécurisée par JWT, chiffrement des données sensibles, performances assurées avec un temps de réponse inférieur à 200 ms par opération.

1.4 Problématique

La problématique centrale que nous abordons peut être formulée ainsi :

Comment concevoir une solution bancaire en ligne intelligente, performante et sécurisée, capable de détecter en temps réel des transactions frauduleuses tout en minimisant les faux positifs et en garantissant une expérience utilisateur optimale ?

Ce questionnement englobe plusieurs sous-enjeux :

1. Assurer une gestion fluide et fiable des opérations bancaires traditionnelles.
2. Développer un algorithme IA performant capable d'identifier les transactions anormales à partir d'un jeu de données d'apprentissage.
3. Intégrer des mécanismes de notification (emails, alertes UI) pour avertir les utilisateurs et les administrateurs lors de détection d'activités suspectes.
4. Maintenir un équilibre entre rigueur sécuritaire et confort utilisateur.

1.5 Objectifs du projet

Objectifs généraux :

— Développer une application bancaire web responsive basée sur les technologies **Spring Boot** (backend) et **React.js** (frontend).
— Concevoir et intégrer un moteur de détection de fraude basé sur un modèle de **Machine Learning supervisé** (algorithmes : arbre de décision, SVM ou régression logistique).
— Assurer la sécurité de la plateforme par l'implémentation de protocoles modernes : **JWT**, **HTTPS**, **BCrypt**, etc.

Objectifs spécifiques :

— Analyser des jeux de données réels ou simulés de transactions bancaires.
— Identifier les caractéristiques discriminantes entre les transactions normales et frauduleuses.
— Mettre en œuvre une architecture logicielle modulaire facilitant la maintenance et l'évolutivité.
— Implémenter un système de gestion des rôles pour contrôler les droits d'accès (admin/client).

1.6 Benchmark des solutions existantes

Un benchmark a été réalisé pour étudier les principales solutions du marché :

— **SAS Fraud Management** : Très complet et performant, mais complexe à déployer, coûteux et peu adapté à un usage éducatif ou à des petites structures.
— **FICO Falcon** : Référence du secteur, avec une forte capacité de détection, mais plateforme propriétaire, difficile à personnaliser.
— **RapidMiner** : Outil open-source adapté à l'analyse de données et au prototypage de modèles IA, mais manque d'intégration native avec des systèmes en temps réel.

Choix retenu : Une solution sur-mesure en full-stack, combinant performance, souplesse et intégration native du module IA, avec un coût de déploiement réduit et un contrôle total sur le code source.

1.7 Méthodologie de gestion de projet

Nous avons opté pour une démarche agile, selon le cadre de travail **Scrum**, afin d'assurer une livraison incrémentale, continue et itérative du projet.

— **Organisation en sprints** : Le projet est divisé en sprints de deux semaines. Chaque sprint commence par une réunion de planification (sprint planning) et se termine par une rétrospective.
— **Product backlog** : Maintenu et priorisé régulièrement, il permet de focaliser le travail sur les fonctionnalités les plus critiques (valeur métier).
— **CI/CD** : Une chaîne d'intégration et de déploiement continue est mise en place à l'aide de GitHub Actions pour assurer la stabilité du produit à chaque livraison.
— **Suivi Kanban** : Suivi des tâches via un tableau Trello ou Jira, pour visualiser l'état d'avancement en temps réel.

1.8 Diagramme de Gantt

Un planning prévisionnel a été établi pour organiser le déroulement du projet sur une période de 12 semaines :

Workstream	Task	Team	Duration	Start Date	End Date
Planning					
	Phase d'étude et planification - Étu	Developer	3	2025-04-21	2025-04-23
	Phase d'étude et planification - Rév	Developer	3	2025-04-24	2025-04-28
Coordination					
	Coordination - Réunions régulières	Developer	14	2025-04-21	2025-05-08
Design					
	Analyse et conception - Création de	Developer	3	2025-04-29	2025-05-01
	Analyse et conception - Définition d	Developer	3	2025-05-02	2025-05-06
Development					
	Développement backend - Mise en	Developer	10	2025-05-07	2025-05-20
	Développement backend - Implém	Developer	2	2025-05-21	2025-05-22
	Développement backend - Implém	Developer	2	2025-05-21	2025-05-22
	Développement backend - Implém	Developer	2	2025-05-21	2025-05-22
	Développement du module IA - Cré	Developer	5	2025-05-21	2025-05-27
	Développement frontend - Créatio	Developer	7	2025-05-21	2025-05-29
	Développement frontend - Intégrat	Developer	2	2025-05-30	2025-06-02
Testing					
	Tests - Validation des fonctionnalité	Developer	2	2025-05-23	2025-05-26
	Tests - Validation du module de dét	Developer	2	2025-05-28	2025-05-29
	Tests - Validation des interfaces util	Developer	2	2025-06-03	2025-06-04
Documentation					
	Documentation - Rédaction de la d	Developer	3	2025-06-05	2025-06-09

FIGURE 1.1 – Planning prévisionnel du projet

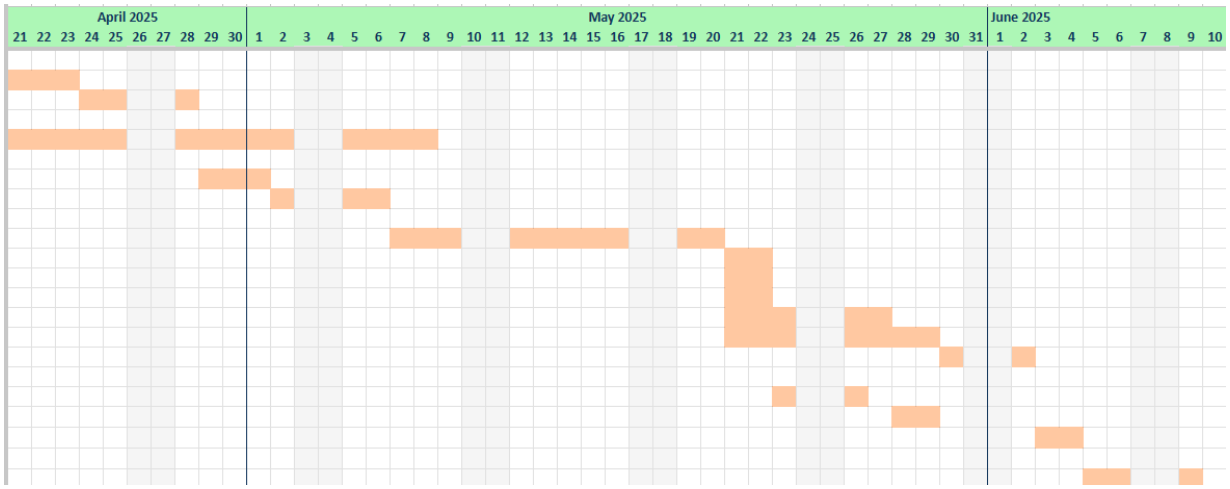

FIGURE 1.2 – Chart Gantt

1.9 Conclusion

Ce chapitre a permis d'établir le cadre global du projet. Nous avons présenté le contexte dans lequel il s'inscrit, les besoins auxquels il répond, les objectifs visés, ainsi que les choix techniques et méthodologiques. Cette première étape constitue la fondation sur laquelle s'appuiera la suite du rapport, à savoir l'analyse des besoins, la modélisation UML du système, l'implémentation, les tests, et enfin l'évaluation de la performance du module de détection de fraude.

Chapitre 2

Analyse et conception

2.1 Introduction

L'étape d'analyse et de conception est cruciale pour la réussite du projet, car elle permet de traduire les besoins exprimés en modèles abstraits guidant le développement. Ce chapitre présente une formalisation complète des besoins fonctionnels et non fonctionnels, suivie d'une modélisation UML du système : diagrammes de cas d'utilisation, d'activité, de séquence, et de classes.

2.2 Besoins fonctionnels

Les fonctionnalités essentielles attendues de l'application sont les suivantes :

- **Authentification sécurisée** : Inscription, connexion, et récupération de mot de passe avec vérification email.
- **Gestion des comptes** : Création, modification, consultation et suppression des comptes utilisateurs.
- **Transactions bancaires** : Virements internes, dépôts, retraits, avec gestion d'annulation dans un délai défini.
- **Historique** : Suivi détaillé et filtrage par période, type et statut de transaction.
- **Détection de fraude** : Analyse des transactions en temps réel à l'aide d'un modèle IA.
- **Notifications** : Alertes automatiques en cas d'anomalies, envoyées par email ou via l'interface.
- **Administration** : Gestion des utilisateurs, rôles (admin/client), et actions de blocage/déblocage.

2.3 Besoins non fonctionnels

Les exigences transverses garantissant la qualité globale du système sont :

- **Sécurité** : Chiffrement des données (AES-256), tokens JWT pour les sessions, protection CSRF.
- **Performance** : Temps de réponse API < 200 ms, avec gestion efficace des connexions concurrentes.
- **Scalabilité** : Architecture pouvant évoluer vers une approche microservices.
- **Ergonomie** : Interfaces adaptatives (responsive), accessibles (normes WCAG 2.1).
- **Testabilité et maintenance** : Code structuré, tests automatisés, documentation claire.

2.4 Diagramme d'activité

Ce diagramme décrit le processus métier principal : la réalisation d'une transaction. Il met en évidence les vérifications effectuées (solde suffisant, validation IA) avant de confirmer l'opération.

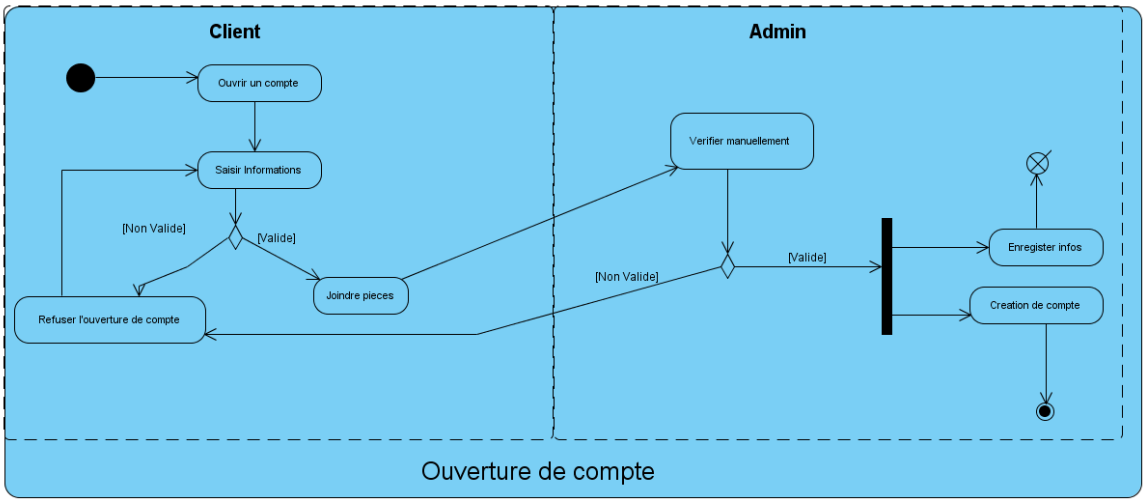

FIGURE 2.1 – Flux d'activité pour la réalisation d'une transaction

2.5 Diagrammes de cas d'utilisation

Les cas d'utilisation illustrent les interactions entre les utilisateurs (clients et administrateurs) et le système.

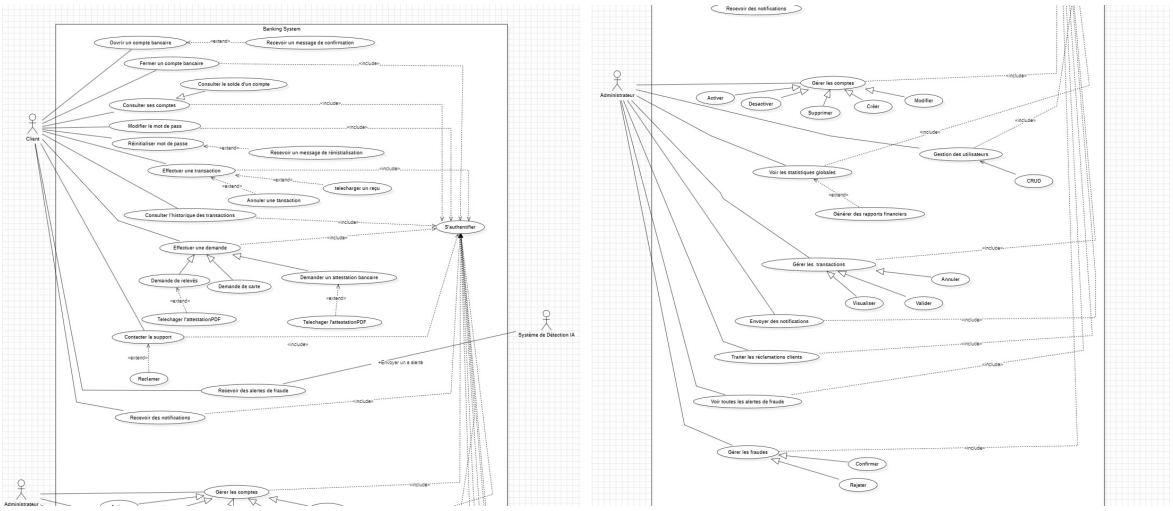

FIGURE 2.2 – Cas d'utilisation principaux

Scénario nominal : Effectuer un virement

1. L'utilisateur authentifié accède à l'espace transactionnel.
2. Il initie un virement en saisissant les informations nécessaires.
3. Le backend enregistre la transaction en état `PENDING`.
4. Le module IA est invoqué pour évaluer le risque.

5. Si le score de risque est faible, le statut devient `COMPLETED`, sinon, une alerte est générée.
6. Le client est notifié, et l'opération est visible dans son historique.

2.6 Diagrammes de séquence

Ce diagramme détaille l'échange entre le frontend, le backend, la base de données et le module IA lors d'une transaction.

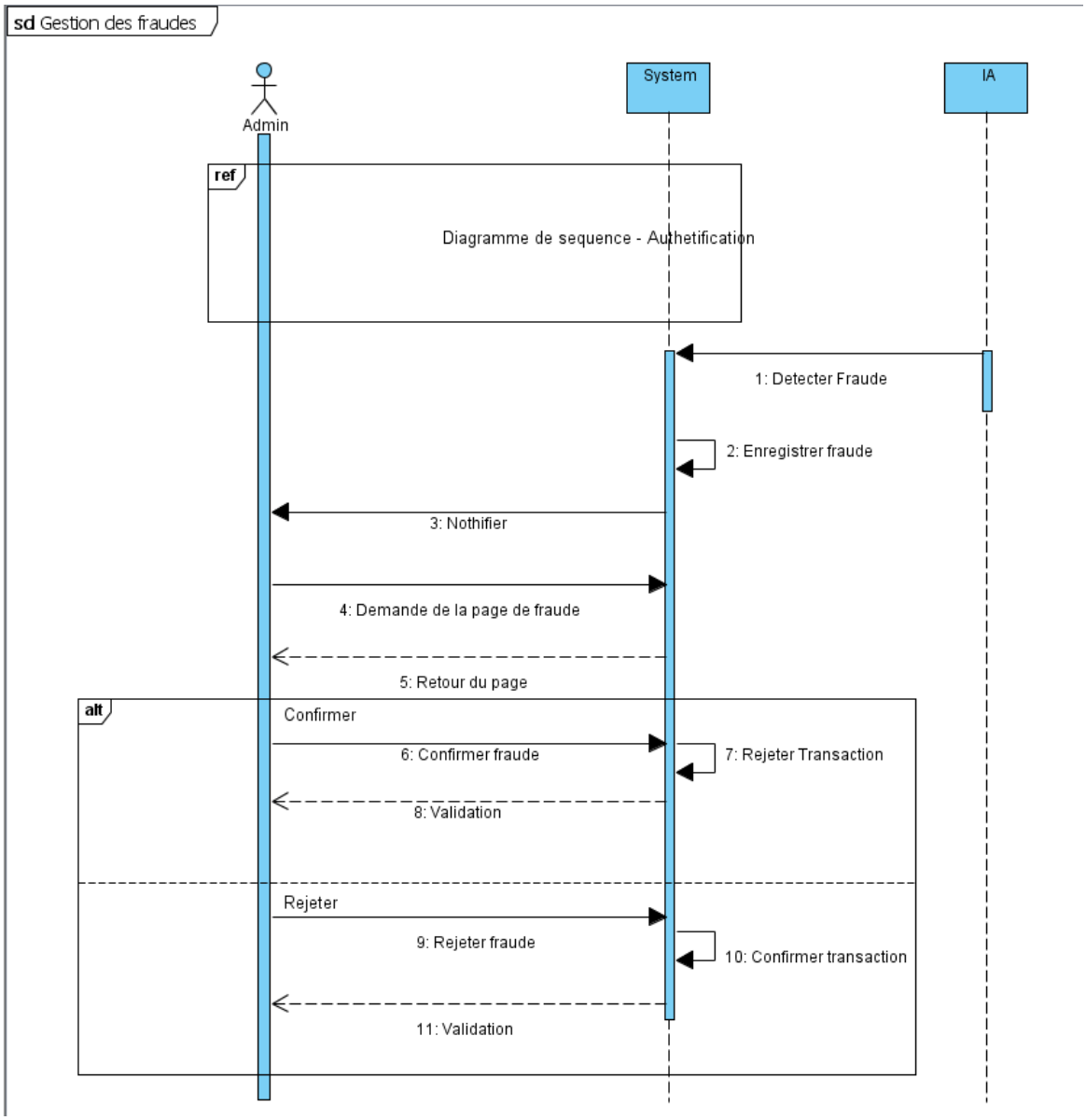

FIGURE 2.3 – Séquence d'une transaction et appel au module IA

2.7 Diagramme de classes

Il représente la structure statique du système, incluant les entités clés comme les utilisateurs, comptes, transactions, et alertes.

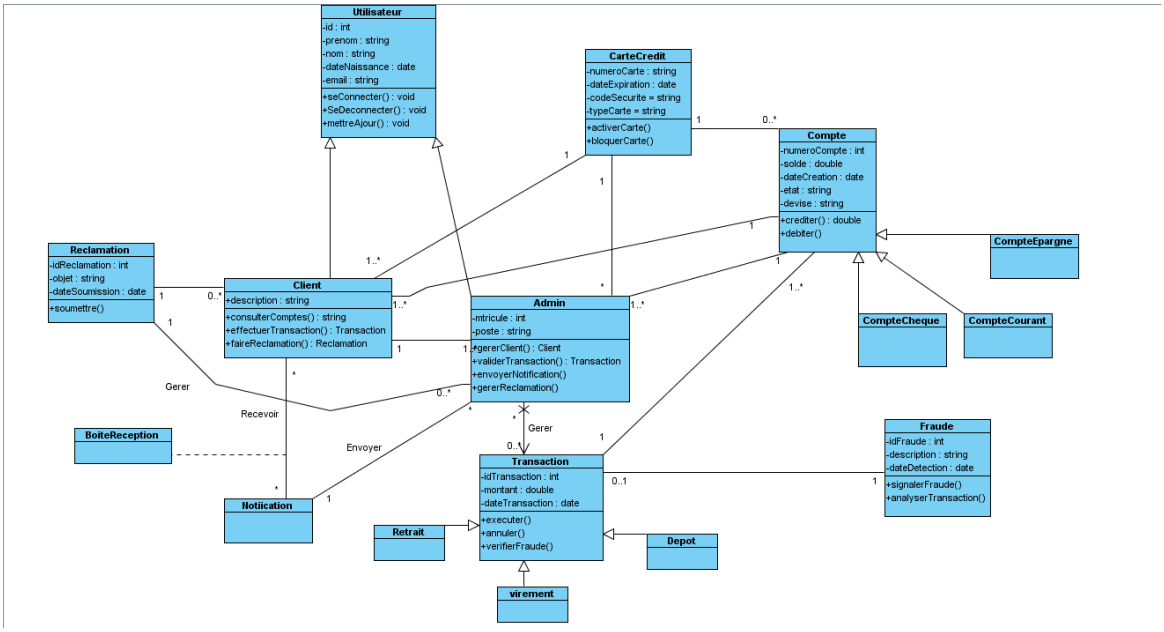

FIGURE 2.4 – Structure statique du système (diagramme de classes)

2.8 Conclusion

L'analyse des besoins et la modélisation UML fournissent une vision claire et structurée du système. Ces artefacts servent de fondation technique pour les étapes de développement, en assurant une correspondance fidèle avec les attentes métier.

Chapitre 3

Réalisation

3.1 Introduction

Ce chapitre décrit la mise en œuvre technique des modèles conçus dans le chapitre précédent. Nous y présentons l'architecture adoptée, les outils et technologies utilisés, ainsi qu'un aperçu des interfaces principales développées dans le cadre de ce projet.

3.2 Architecture de la solution

Le système repose sur une architecture en trois couches bien distinctes, favorisant la séparation des préoccupations et facilitant la maintenance :

- **Frontend** : Développé avec `React.js`, en combinaison avec `Tailwind CSS` pour le design responsive et `Shadcn UI` pour les composants réutilisables.
- **Backend** : Basé sur le framework `Spring Boot` (Java), exposant une API REST sécurisée. Le module de détection de fraude repose sur `Flask` (Python), permettant l'intégration du modèle IA.
- **Base de Données** : Utilisation de `PostgreSQL` pour le stockage des données structurées, assurant robustesse et évolutivité.

Les échanges entre les couches s'effectuent via des appels API REST, sécurisés à l'aide de jetons JWT (JSON Web Token).

3.3 Outils et technologies utilisés

Le développement a nécessité l'usage d'un ensemble d'outils pour garantir une productivité optimale, une gestion efficace du code source et une validation rigoureuse des fonctionnalités.

- **Environnements de développement** :
  - `IntelliJ IDEA` pour le backend Java.
  - `Visual Studio Code` pour le frontend React.js.
- **Contrôle de version** : `Git` et `GitHub` pour la gestion collaborative du code et le suivi de l'évolution du projet.
- **Tests d'API** : `Postman`, utilisé à la fois pour les tests manuels et l'automatisation de scénarios à travers des collections.

3.4 Interfaces de l'application

Les interfaces utilisateurs ont été conçues dans un souci d'ergonomie, de clarté et d'accessibilité, conformément aux standards du responsive design.

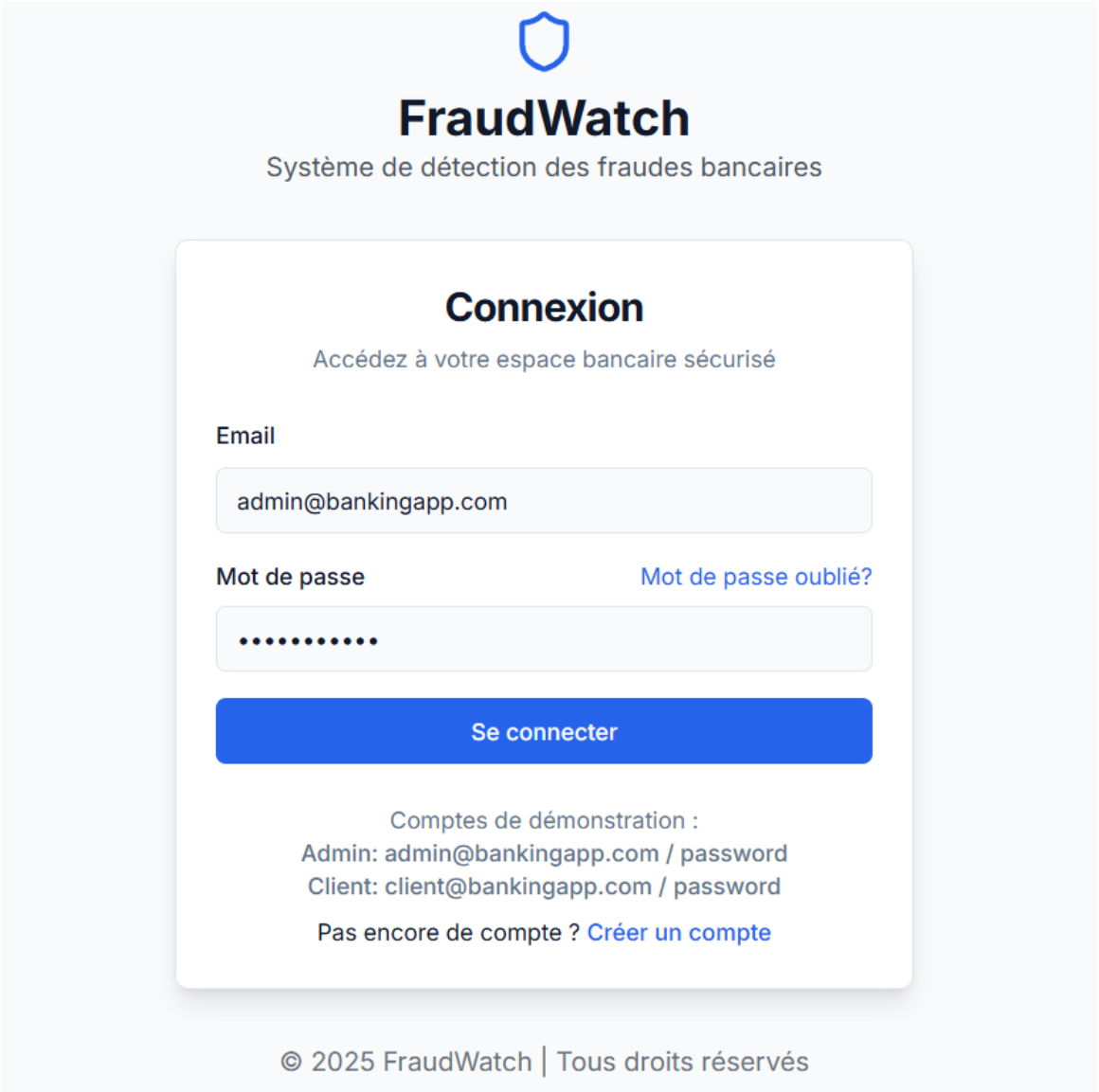

FIGURE 3.1 – Page de connexion sécurisée

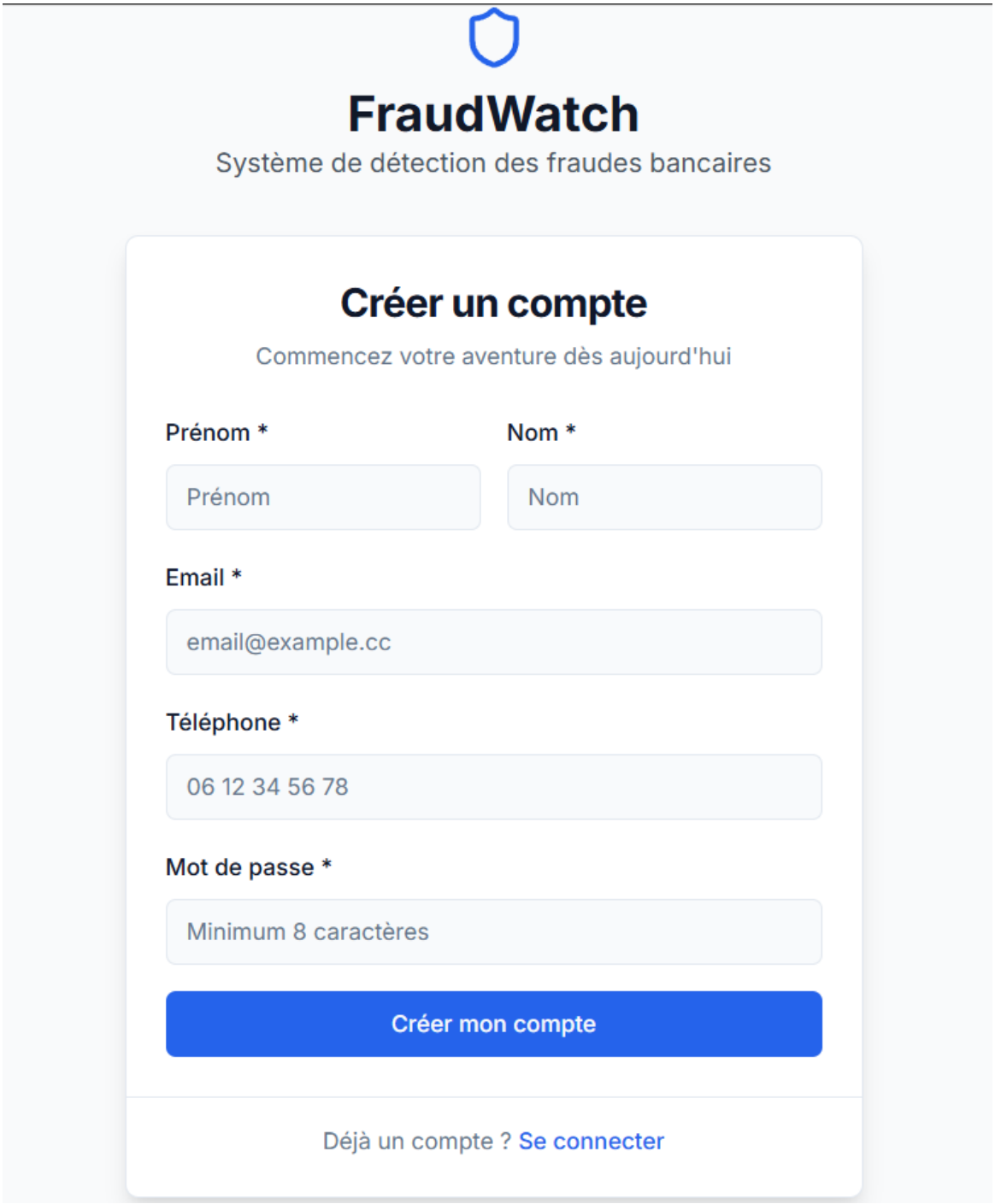

FIGURE 3.2 – Page d'inscription sécurisée

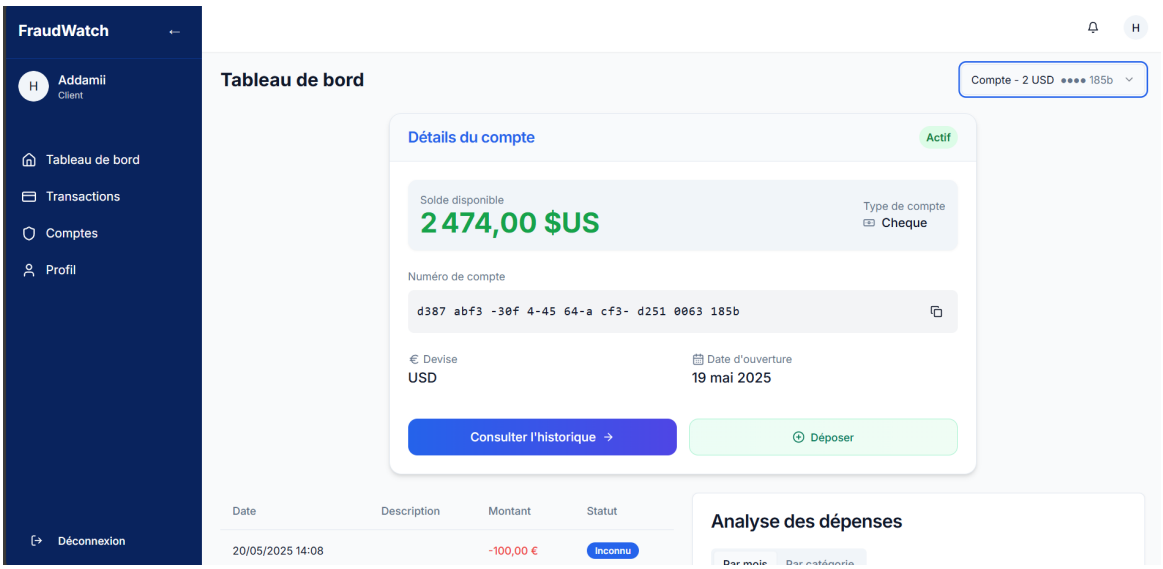

FIGURE 3.3 – Vue du tableau de bord pour client

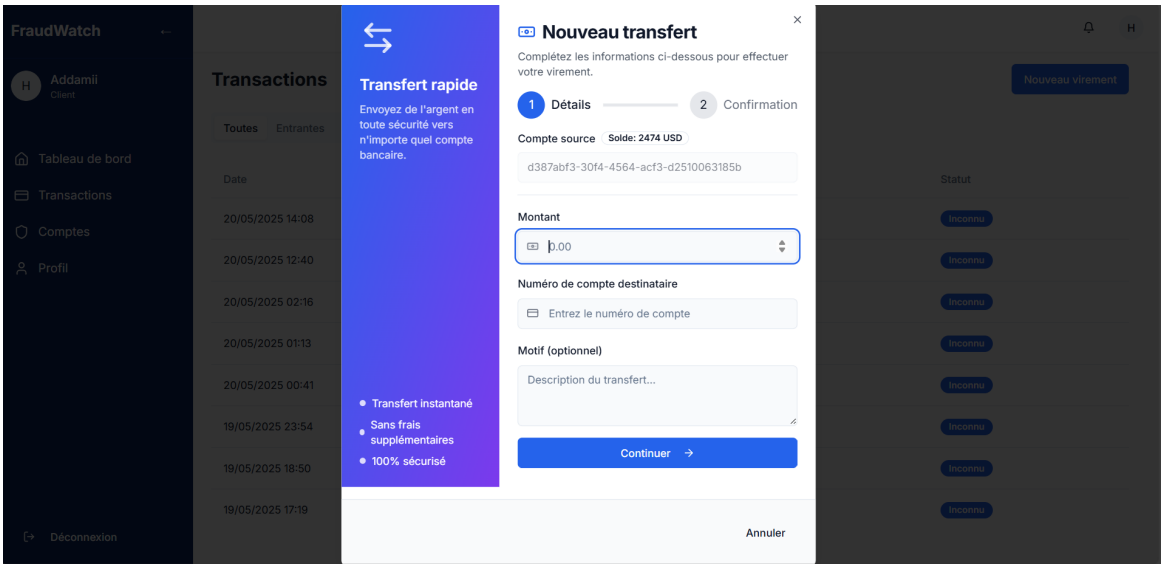

FIGURE 3.4 – Vue pour effectuer u transfert

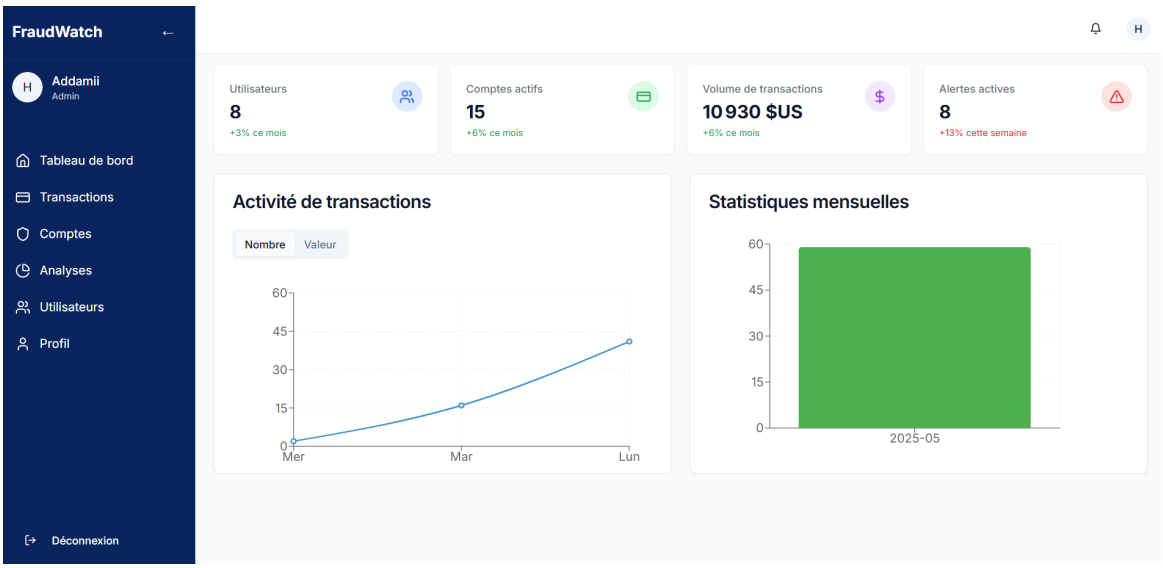

FIGURE 3.5 – Vue du tableau de bord pour administrateur

FIGURE 3.6 – Vue d'analyse 1

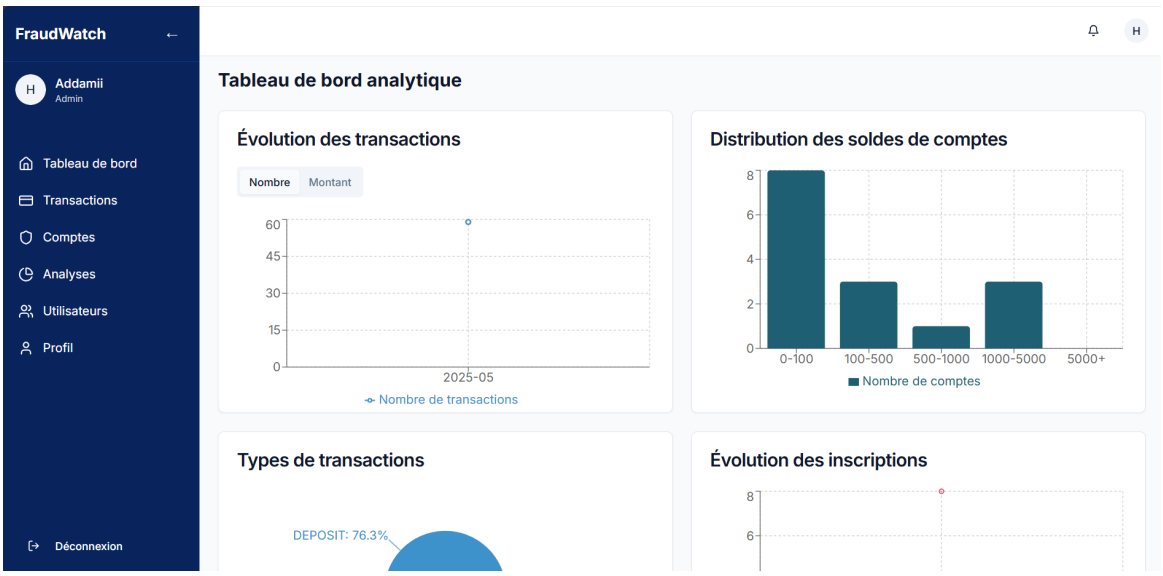

FIGURE 3.7 – Vue d'analyse 2

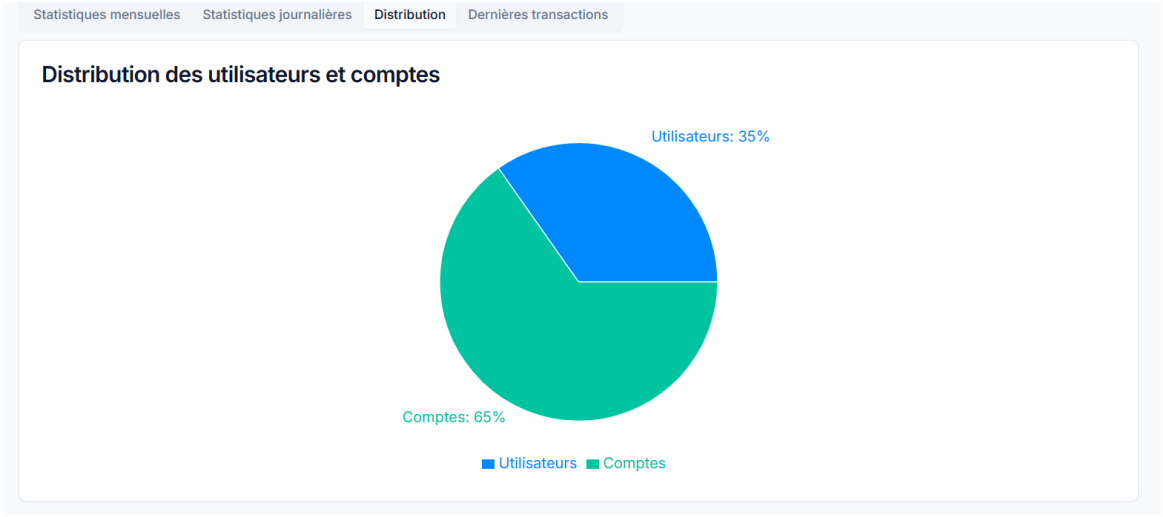

FIGURE 3.8 – Vue d'analyse 3

Chaque utilisateur dispose d'une interface adaptée à son rôle, avec des fonctionnalités spécifiques selon qu'il soit client, administrateur ou superviseur sécurité.

3.5 Conclusion

La phase de réalisation a permis de concrétiser les spécifications établies lors de l'analyse. L'architecture en couches s'est révélée efficace pour une répartition claire des responsabilités. Les choix technologiques adoptés ont permis de répondre aux exigences fonctionnelles et non fonctionnelles en termes de sécurité, de performance, de maintenabilité et d'ergonomie. L'intégration fluide du module d'intelligence artificielle témoigne de la modularité et de la flexibilité de l'architecture mise en œuvre.

Bibliographie

[1] Rod Johnson et al., *Spring in Action*, Manning Publications, 6th Edition, 2022.

[2] Alex Banks, Eve Porcello, *Learning React*, O'Reilly Media, 2nd Edition, 2020.

[3] RFC 7519, *JSON Web Token (JWT)*, IETF, May 2015. Disponible sur : `https://datatracker.ietf.org/doc/html/rfc7519`

[4] OWASP Foundation, *OWASP Top Ten 2023*, Disponible sur : `https://owasp.org/www-project-top-ten/`

[5] Banque des Règlements Internationaux, *Rapport annuel sur les systèmes de paiement*, 2023. Disponible sur : `https://www.bis.org/statistics/payment_stats.htm`

[6] FICO, *Falcon Fraud Manager Overview*, White Paper, 2022. Disponible sur : `https://www.fico.com/`

[7] GitHub, *Spring Boot + React Banking System Example*, Repos open-source consultés entre 2024 et 2025.