République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara - Boumerdès

Département d'Informatique



Domain : Informatique
Filière : Informatique
Spécialité : Système Informatique (SI)

Mémoire de fin d'études en vue de l'obtention du Diplôme de Licence Académique

Thème :

Le suivi et la gestion des réclamations clients versus Payment électronique

Présenté par :

- AMRANI YANIS
- DADDI HAMMOU SALAH
- OUARAB IMAD
- NOUI MOHAMMED RAYANE

Soutenu durant l'année 2025/2026 Devant le jury composé de :

Encadreur : Ad Meriem

Examinateur : Messaoud Ahmed-Nacer

DEDICATION

À mes chers parents, leur amour, leurs sacrifices, leurs encouragements et leur soutien constant durant tout mon parcours. Grâce à eux, j'ai pu avancer, surmonter les difficultés et arriver jusqu'à cette étape.

À mon grand frère dont l'aide et ses encouragements ont été une grande source de motivation pour moi

Salah.

À mes parents, mes amis et à toutes les personnes qui m'ont soutenu, merci pour votre amour, votre soutien et vos encouragements constants, ainsi que pour votre présence et vos conseils qui m'ont aidé à poursuivre mon parcours avec confiance et force.

Yanis.

À mes chers parents pour leur amour, leur patience et leur soutien durant tout mon parcours. Un grand merci surtout à mon père, qui m'a toujours accompagné et transporté à la gare avec patience et sacrifice afin que je puisse poursuivre mes études dans les meilleures conditions.

Rayan.

À mes parents qui m'ont toujours soutenu et ont cru en moi, sans votre motivation, je n'aurais jamais pu aller aussi loin ; À ma sœur bien-aimée, ton soutien a toujours été une source d'inspiration et ta confiance en mes capacités est inestimable ; et à mon frère, ton soutien, quelle que soit la distance, a toujours été un guide précieux.

Imad.

REMERCIEMENT

Nous remercions Allah, qui nous a accordé la santé, la force et la persévérance tout au long de notre parcours universitaire, que nous clôturons aujourd'hui avec ce projet de fin d'études.

Nous adressons nos remerciements à notre encadreur, Mme : **Ad Meriem**, pour son encadrement sa disponibilité ainsi que son soutien tout au long de la réalisation de ce mémoire, et pour le temps consacré à l'évaluation de notre travail.

Nous exprimons également notre profonde gratitude à notre professeur de l'année précédente, Mme **: Djeddai Selma** pour son aide, ses orientations et les connaissances qu'elle nous a apportées, qui ont grandement contribué à la réussite de notre projet.

Nos remerciements s'adressent aussi à notre encadreur au sein de l'entreprise Faderco, Mme **Bouachat Anfel** pour son accueil, son encadrement durant le stage, ainsi que pour son aide et ses conseils professionnels qui nous ont permis de mieux comprendre les besoins réels du projet.

Nous remercions également les membres du jury ainsi que notre examinateur M : **Messaoud Ahmed-Nacer** ,pour l'intérêt qu'ils ont porté à notre travail en acceptant d'évaluer ce mémoire.

Enfin, nous exprimons notre gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail.

TABLE DES MATIÈRES

Table des matières

DEDICATION ..... a

REMERCIEMENT ..... b

TABLE DES MATIÈRES c

LISTE DES FIGURES ..... e

Résumé ..... f

LISTE DES ABREVIATION ..... g

INTODUCTION GENERAL 1

Contexte et problématique 2

1.1 Introduction 2

1.2 présentation de l'entreprise Faderco 2

1.3 Présentation de l'application Presalio et API de paiement 3

1.4 Etude des postes de travail 5

1.5 Flux d'information 5

1.6 Problématique 6

1.7 Solution proposée 6

1.8 Objectif de la solution proposée 7

1.8.1 Objectifs généraux 7

1.8.2 Objectifs spécifique 7

1.8.3 publics cible 8

1.9 Conclusion 8

Spécification et conception 9

2.1 Introduction 9

2.2 Présentation de UML 9

2.3 Identification des acteurs 10

2.4 Diagramme des cas d'utilisation 10

2.5 Diagramme de séquence 11

- Diagramme de séquence d'authentification site web 11
- Diagramme de séquence d'authentification client 13

2.6 Diagramme des classes 13
2.7 Model relationnelle 15
2.8 Conclusion 15
Implémentation 16
3.1 introduction 16
3.2 Environnement de développement matériel 16
3.3 Environnement de développement logiciel 16
3.3.1 Langages de programmation 16
3.3.2 Outil de développement 18
3.4 Présentation de l'application mobile 20
3.5 : Présentation de l'application web 22
3.5.1 Présentation de l'interface login 23
3.5.2 Présentation de l'interface Administrateur 23
3.5.3 Présentation de l'interface agent 24
3.5.4 Présentation de l'interface finance 25
3.5.5 Présentations de l'interface caissier 26
3.6 Conclusion 27
Conclusion Générale 28
Bibliographie 29

LISTE DES FIGURES

Figure 1.1 – Les produits de faderco[1] 2
Figure 1.2 – Les marques de faderco[2]..... 3
Figure 1.3 – communication entre presalio et L'API de paiement Guiddini Epay[3].....4
Figure 2.1 – diagramme de cas d'utilisation[4]..... 11
Figure 2.2 – diagramme de séquence d'authentification site web[5] 12
Figure 2.3 – diagramme de séquence d'authentification client[6] 13
Figure 2.4 – diagramme de classe[7] 14
Figure 3.1 – l'interface de sign-up et login de l'application mobile[8]..... 20
Figure 3.2 – l'interface de demande de remboursement[9] 21
Figure 3.3 – demande envoyée[10] 21
Figure 3.4 – L'interface de la page d'accueil et de la page de Déconnexion[11] 22
Figure 3.5 – L'interface de login[12] 23
Figure 3.6 – l'interface de l'admin[13]..... 24
Figure 3.7 – L'interface de l'agent[14] 25
Figure 3.8 – L'interface de finance[15]..... 26
Figure 3.9 – L'interface de caissier[16].....27
Figure 3.10 – Fichier Excel des clients remboursés[17].....27

Résumé

Le développement des paiements électroniques a facilité les transactions commerciales, mais a également entraîné l'apparition de nouveaux problèmes, notamment les cas de double paiement. Ces erreurs peuvent provoquer des réclamations clients et compliquer la gestion des remboursements au sein des entreprises.

Dans ce cadre, notre projet réalisé au sein de Faderco consiste à développer une solution informatique dédiée à la gestion des demandes de remboursement liées aux doubles paiements. Le système met en place un processus de vérification à plusieurs niveaux impliquant les services agent, finance et caissier, tout en assurant la traçabilité complète des opérations depuis la soumission de la demande jusqu'au remboursement final.

À travers ce projet, nous avons cherché non seulement à résoudre un problème réel rencontré en entreprise, mais également à améliorer l'expérience client, renforcer la sécurité des traitements et optimiser le travail des différents services concernés.

LISTE DES ABREVIATION

API : Application Programming Interface

CSS : Cascading Style Sheets (Feuilles de style pour la mise en forme web).

CIB : Carte Interbancaire

CU: Cas d'utilisation

DCU : Diagramme de Cas d'Utilisation (Partie de la modélisation UML).

Epay :Electronic Payment

FMCG : Fast Moving Consumer Goods

JWT: JSON Web Token

JSON : JavaScript Object Notation

KPI : Key Performance Indicator

NPM: Node Package Manager.

QR Code : Quick Response Code

RBAC: Role-Based Access Control

SDK : Software Development Kit

SATIM : Société d'Automatisation des Transactions Interbancaires et de Monétique

SQL : Structured Query Language

SGBD : Système de Gestion de Base de Données

UML : Unified Modeling Language (Langage de modélisation unifié)

INTODUCTION GENERAL

Le développement rapide des technologies numériques a profondément transformé les méthodes de paiement utilisées par les entreprises et les consommateurs. Les paiements électroniques occupent aujourd'hui une place essentielle grâce à leur rapidité, leur simplicité et leur accessibilité. En Algérie, plusieurs entreprises ont adopté des solutions de paiement en ligne afin d'améliorer leurs services et de faciliter les transactions commerciales.

Cependant, malgré les avantages offerts par ces technologies, certaines anomalies peuvent apparaître lors des opérations de paiement électronique, notamment les cas de double paiement. Ce problème survient généralement lorsqu'un client effectue plusieurs paiements pour une même transaction à cause d'un retard de confirmation, d'une mauvaise connexion internet ou d'un dysfonctionnement technique. Ces situations entraînent des difficultés de gestion pour l'entreprise ainsi que des réclamations de la part des clients.

Dans ce contexte, l'entreprise Faderco a exprimé le besoin de mettre en place une solution informatique permettant de gérer efficacement les demandes de remboursement liées aux doubles paiements effectués via l'API de paiement Guiddini Epay intégrée à l'application Presalio.

L'objectif principal de notre projet est donc de concevoir et de développer un système de gestion des réclamations composé d'une application mobile destinée aux clients et d'une application web destinée aux différents services internes de l'entreprise (agent, finance, caissier et administrateur). Cette solution permet d'assurer le suivi, la vérification et le traitement des demandes de remboursement de manière organisée, rapide et sécurisée.

Le système proposé vise également à améliorer la traçabilité des opérations, réduire les erreurs de traitement et renforcer la confiance des clients envers les paiements électroniques. Pour atteindre ces objectifs, nous avons utilisé plusieurs technologies modernes ainsi qu'une architecture permettant une gestion claire et structurée des différentes étapes du processus de remboursement.

Chapitre

1

Contexte et problématique

1.1 Introduction

Dans le premier chapitre, nous allons présenter une étude générale sur l'entreprise Faderco spa et ses activités. Ensuite nous verrons le problème rencontré et l'objectif de notre travail.

1.2 présentation de l'entreprise Faderco

Faderco est une entreprise industrielle algérienne créée en 1986 et spécialisée dans la fabrication de produits d'hygiène corporelle. Basée à Alger, elle s'est progressivement imposée comme l'un des leaders nationaux dans ce domaine grâce à la qualité de ses produits et à sa stratégie de développement. L'entreprise propose une large gamme de produits destinés à la vie quotidienne, notamment des couches pour bébés, des produits d'hygiène féminine, des articles pour adultes ainsi que des produits à base de papier et de coton. Au fil des années, Faderco a connu une croissance importante en développant ses activités à travers plusieurs filiales, comme la distribution et la production de matières premières. Elle repose sur des valeurs telles que l'innovation, l'intégrité et l'expertise, tout en cherchant à améliorer le bien-être des consommateurs. Aujourd'hui, Faderco joue un rôle clé dans l'industrie algérienne en contribuant à la production locale, à la création d'emplois et à la satisfaction des besoins du marché national.[1]



Figure 1.1 – Les produits de faderco [1]



Figure 1.2 – Les marques de faderco[2]

1.3 Présentation de l'application Presalio et API de paiement

L'application Presalio est une solution digitale utilisée principalement dans le domaine de la distribution et du commerce (FMCG). Elle permet aux entreprises de gérer efficacement leurs opérations commerciales, notamment la gestion des équipes de vente, le suivi des performances et l'organisation des activités sur le terrain. Grâce à ses fonctionnalités avancées, Presalio facilite la prise de commandes, la gestion des stocks, l'optimisation des tournées de livraison ainsi que le suivi des indicateurs de performance (KPI). L'application intègre également des outils d'intelligence artificielle qui aident à analyser les données, prévoir les ventes et améliorer la prise de décision. Elle offre en plus des fonctionnalités de communication, de gestion des promotions et de fidélisation des clients. Ainsi, Presalio permet aux entreprises d'améliorer leur efficacité opérationnelle, de renforcer la relation avec les clients et d'optimiser leurs résultats commerciaux.

L'API de paiement Guiddini Epay est une solution développée en Algérie permettant aux entreprises d'intégrer facilement le paiement en ligne dans leurs

sites web ou applications mobiles. Elle fonctionne comme un intermédiaire entre le commerçant, la banque et le système monétique nationale, en permettant d'accepter des paiements via les cartes CIB et EDAHABIA. SATIM joue un rôle essentiel dans ce système, car c'est l'organisme qui gère l'infrastructure de paiement électronique en Algérie et certifie les solutions comme Guiddini.

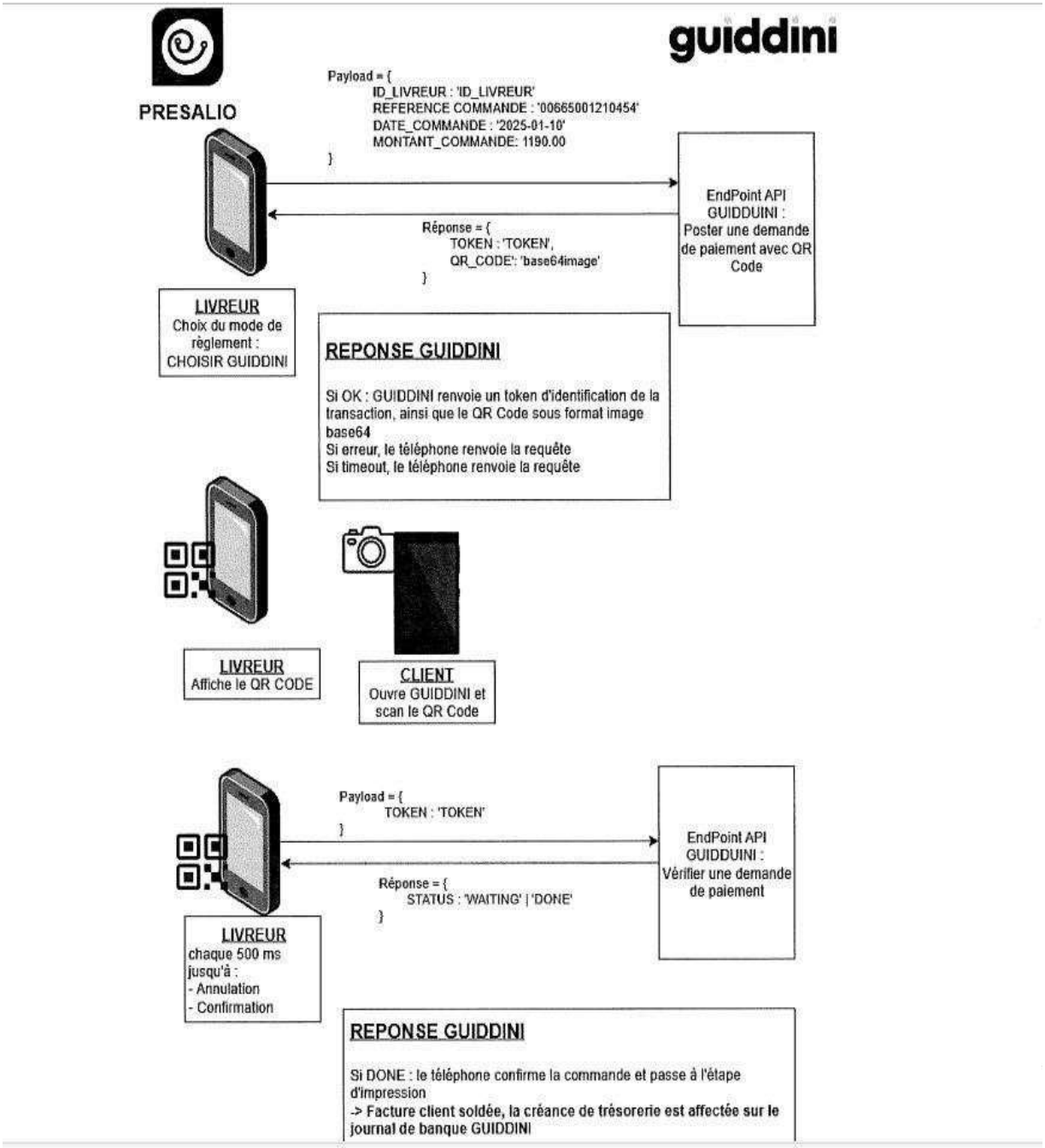


Figure 1.3 – communication entre presalio et L'API de paiement Guiddini Epay[3]

1.4 Etude des postes de travail

- **Le livreur** : utilise l'application presalio déjà existante afin de gérer les opérations de livraison et confirmer les paiements des clients.
- **Le Client** : l'utilisateur final qui effectuer un paiement et soumet une demande de remboursement via l'application mobile.
- **L'agent** : le responsable chargé de recevoir, analyser et traiter les demandes soumises par les clients.
- **Finance** : le département qui vérifie les demandes transmises par l'agent en les croisant avec les relevés bancaires, avant de les approuver ou de les rejeter.
- **Caissier** : le service chargé d'exécuter les remboursements approuvés et de notifier les clients concernés.
- **Administrateur :** L'administrateur assure la gestion globale du système

1.5 Flux d'information

Le processus débute lorsque le pré-vendeur (commercial terrain) effectue une visite chez un client (par exemple une supérette). À l'aide de l'application mobile Presalio, il saisit une commande comprenant les produits demandés (couches, papier, etc.) ainsi que les quantités correspondantes. Après validation, la commande est transmise au système d'information de Faderco pour traitement.

Par la suite, le livreur se rend chez le client afin d'assurer la livraison de la commande. Lors de cette étape, le client choisit le mode de paiement, soit en espèces, soit en ligne. Dans le cas d'un paiement en ligne, l'application génère automatiquement un code QR sécurisé via l'API de Guiddini Epay.

Le client procède alors au scan du code QR, ce qui le redirige vers une interface de paiement sécurisée. Il y renseigne les informations de sa carte CIB ou EDAHABIA. La transaction est ensuite traitée par le système monétique national, via SATIM, garantissant la sécurité, l'authenticité et la validation de l'opération.

Une fois le paiement effectué :

- Une confirmation est envoyée au client
- Le Livreur reçoit une notification dans son application
- Le système de Faderco met à jour le statut de la commande (payée)

1.6 Problématique

Le problème de double paiement survient lorsqu'un client effectue plusieurs transactions pour une même opération, généralement sans en avoir l'intention. Dans le cas des paiements par QR code, Cette situation peut se produire lorsque le livreur ne reçoit pas la confirmation de paiement après la première tentative de paiement effectué par le client. En raison des facteurs tels qu'un téléphone ancien, une mauvaise connexion internet ou un temps de réponse élevé, le client peut croire que la transaction n'a pas été prise en compte et décide alors de répéter l'opération ou de payer en espèces.

Cette situation crée une incohérence entre les transactions enregistrées et la réalité de l'achat, ce qui complique le suivi des paiements. Elle peut également entraîner des réclamations de la part des clients, qui constatent par la suite qu'ils ont été débités plusieurs fois pour un seul produit ou service.

À long terme, ce type de problème peut nuire à l'image de l'entreprise, car les clients risquent de perdre confiance dans la fiabilité du service proposé. Cette perte de confiance peut avoir un impact direct sur la relation client et sur la fidélisation, notamment si ces incidents se répètent et ne sont pas traités efficacement.

1.7 Solution proposée

Afin de résoudre le problème de double paiement, il est nécessaire de mettre en place une solution basée sur un système structuré de gestion des demandes de remboursement, accompagné d'un mécanisme de vérification à plusieurs niveaux.

Dans un premier temps, une interface de demande de remboursement (application mobile) est mise à disposition du client. À travers cette interface, le client renseigne ses informations personnelles (nom, identifiant client, numéro de téléphone) ainsi que les détails de la transaction concernée (montant, numéro de transaction, raison). Une fois la demande soumise, elle est enregistrée dans une base de données dédiée et transmise à un agent pour traitement.

Dans un second temps, l'agent intervient pour analyser la demande. Il accède à une interface qui lui permet de consulter les demandes enregistrées ainsi que les transactions associées. Grâce à un système de filtrage basé sur des critères tels que le montant, l'identifiant du client et la date, il peut identifier les cas de

double paiement. Après vérification, l'agent décide soit de transmettre la demande au service financier, soit de la rejeter.

Dans un troisième temps, Le service financier vérifie les demandes validées par l'agent en les comparant aux relevés bancaires afin de confirmer le double paiement. Si l'erreur est confirmée, la demande est envoyée au caissier pour effectuer le remboursement, sinon elle est rejetée

Enfin, Le caissier rembourse ensuite le client et le système génère automatiquement un fichier Excel contenant la liste des remboursements effectués afin d'assurer le suivi et la traçabilité.

1.8 Objectif de la solution proposée

1.8.1 Objectifs généraux

Notre projet a pour but de développer une solution informatique composée d'une application mobile destinée aux clients et d'une application web destinée aux différents services internes de Faderco (agent, finance, caissier). Cette solution vise à résoudre le problème de double paiement survenant lors des transactions électroniques effectuées via l'API Guiddini EBay, en mettant en place un système structuré de détection et de gestion des demandes de remboursement.

1.8.2 Objectifs spécifique

- Détecter automatiquement les cas de double paiement en comparant les transactions enregistrées dans le système avec les relevés bancaires importés.
- Permettre au client de soumettre une demande de remboursement via une interface mobile simple et accessible, en renseignant ses informations personnelles et les détails de la transaction concernée.

- Permettre à l'agent de consulter, analyser et traiter les demandes de remboursement reçues, en les comparant avec l'historique des paiements disponible via Presalio.
- Permettre au service financier de vérifier les demandes transmises par l'agent en les croisant avec les relevés bancaires, puis de les approuver ou de les rejeter.
- Permettre au caissier d'exécuter les remboursements approuvés et de notifier les clients concernés.
- Assurer la traçabilité complète de chaque demande de remboursement, depuis sa soumission jusqu'à son traitement final.

1.8.3 publics cible

Le système développé s'adresse à quatre catégories d'utilisateurs :

- **Les clients** de Faderco, qui utilisent l'application mobile pour soumettre leurs demandes de remboursement en cas de double paiement détecté.
- **Les agents commerciaux**, qui accèdent à l'interface web pour analyser et valider les demandes reçues avant de les transmettre au service financier.
- **Le financier**, qui vérifie les demandes en les comparant aux relevés bancaires et prend la décision finale d'approbation ou de rejet.
- **Le caissier**, qui exécute les remboursements validés et assurent la communication avec les clients concernés.

1.9 Conclusion

Dans ce chapitre, nous avons présenté l'entreprise Faderco SPA et ses activités, puis nous avons exposé la problématique des erreurs de paiement électronique. Enfin, nous avons présenté la solution de remboursement proposée et défini l'objectif de notre travail.

Chapitre 2

Spécification et conception

2.1 Introduction

Après une étude approfondie de l'existant nous allons dans ce qui suit présenter l'analyse.

La conception qui est une phase importante avant la réalisation de tout projet, Cette phase nécessite des méthodes permettant en place un modelé sur lequel on va s'appuyer, et parmi ces Modèles se trouve :

Les diagrammes UML puis nous mettons en place les différents diagrammes (cas d'utilisation, séquences, et diagrammes de classe) afin de rendre le système et plus ressortir les points auxquels quels s'intéresse.

2.2 Présentation de UML

UML (Unified Modeling Language) est un langage de modélisation visuelle utilisé pour décrire et spécifier des systèmes logiciels. Il est largement utilisé dans l'industrie, Du développement de logiciels pour représenter les processus, les structures et les Comportements d'un système, Il existe différents types de diagrammes UML [2] :

1. Les diagrammes de comportement : Qui représentent les processus et les interactions entre Les objets il construire :

— **Diagramme de cas d'utilisation.**

— **Diagramme de séquence.**

2. Les diagrammes de structure : Qui représentent les éléments du système, leurs Propriétés, et leurs relations il construire :

— **Diagramme de classe**

2.3 Identification des acteurs

Notre système implique cinq acteurs principaux qui interagissent avec les différentes interfaces de l'application mobile et l'application web :

- **Le Client**
- **L'agent**
- **Le Finance**
- **Le Caissier**
- **Administrateur**

2.4 Diagramme des cas d'utilisation

Un cas d'utilisation décrit une fonctionnalité que le système exécute afin de répondre à l'objectif d'un acteur. Il doit produire un résultat observable et utile pour l'utilisateur du système. Dans notre cas, chaque acteur dispose d'un ensemble de cas d'utilisation qui lui est propre, reflétant son rôle et ses responsabilités au sein du processus de gestion des remboursements.

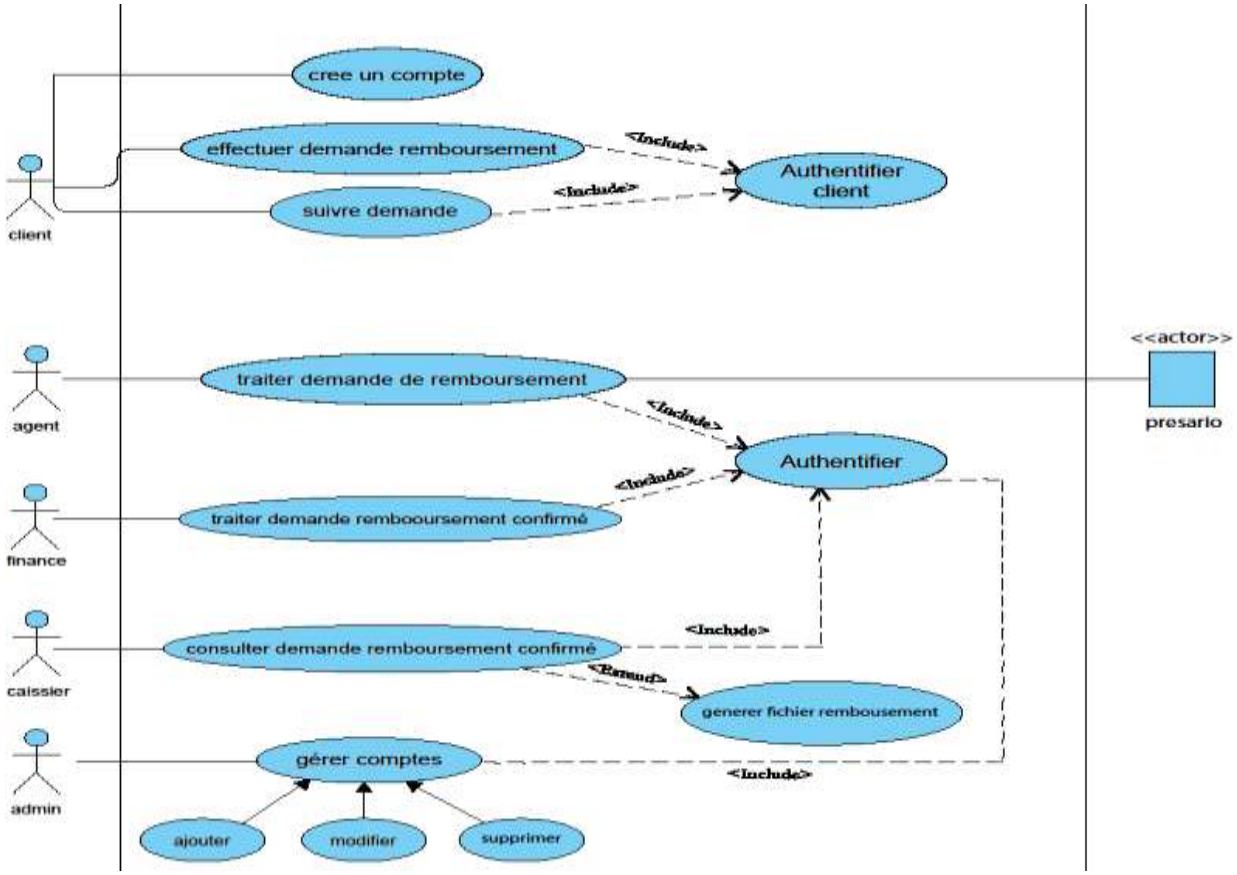


Figure 2.1 – diagramme de cas d'utilisation[4]

2.5 Diagramme de séquence

Un diagramme de séquences est une représentation graphique des interactions entre les acteurs et le système dans l'ordre chronologique, et il décrit la dynamique du système.

Pour modéliser notre système, il est difficile de représenter toute la dynamique du système sur un seul diagramme. La dynamique globale sera également représentée par un ensemble de diagrammes de séquence, chaque diagramme de séquence étant généralement associé à une fonction du système.

- Diagramme de séquence d'authentification site web

Ce diagramme représente le flux d'authentification et d'autorisation basé sur JWT et RBAC, permettant de rediriger les utilisateurs vers leurs dashbords respectifs selon leurs rôles (Agent, Finance, Cassier,Admin), tout en sécurisant l'accès aux routes backend.

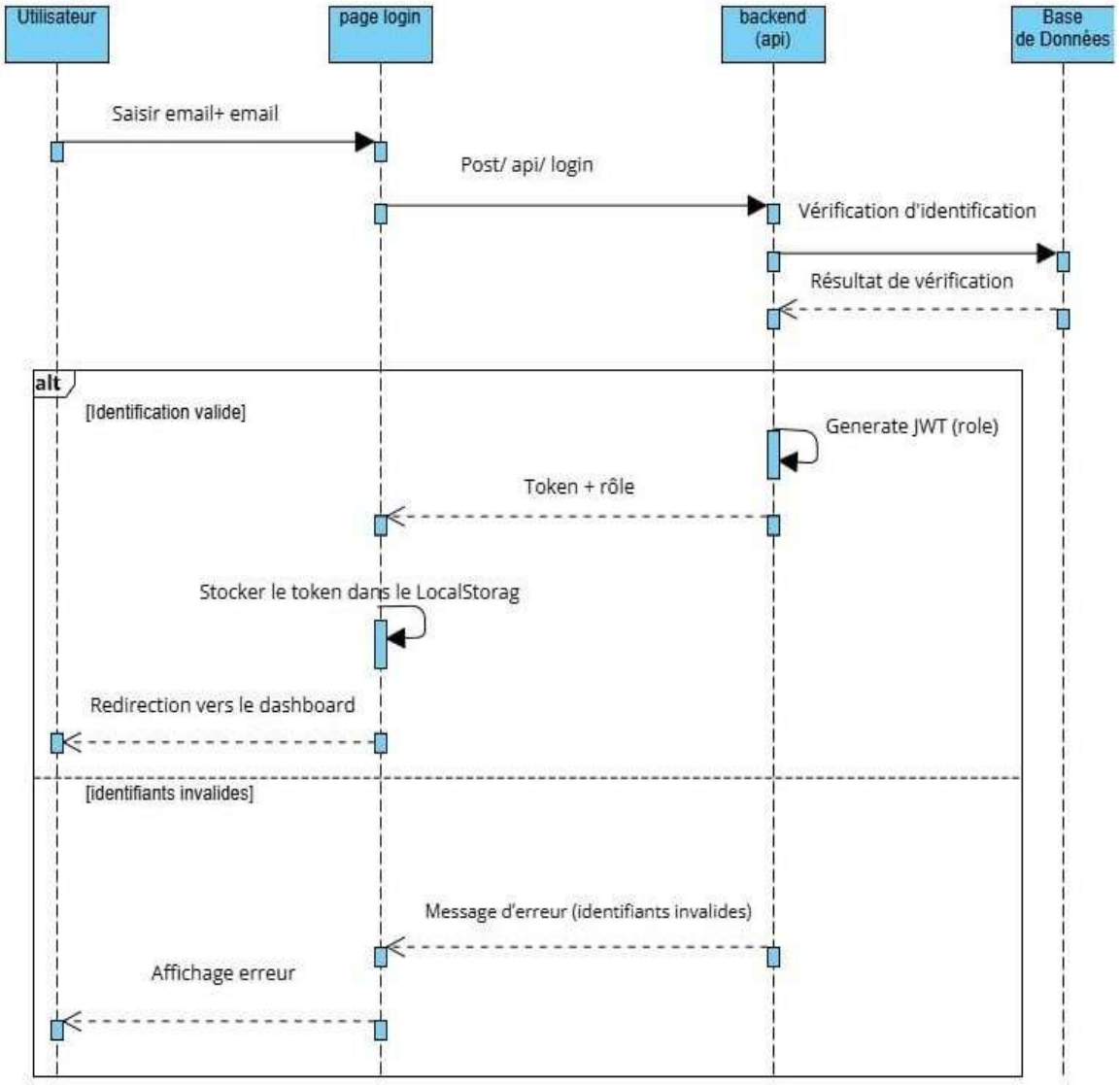


Figure 2.2 – diagramme de séquence d'authentification site web[5]

- Diagramme de séquence d’authentification client

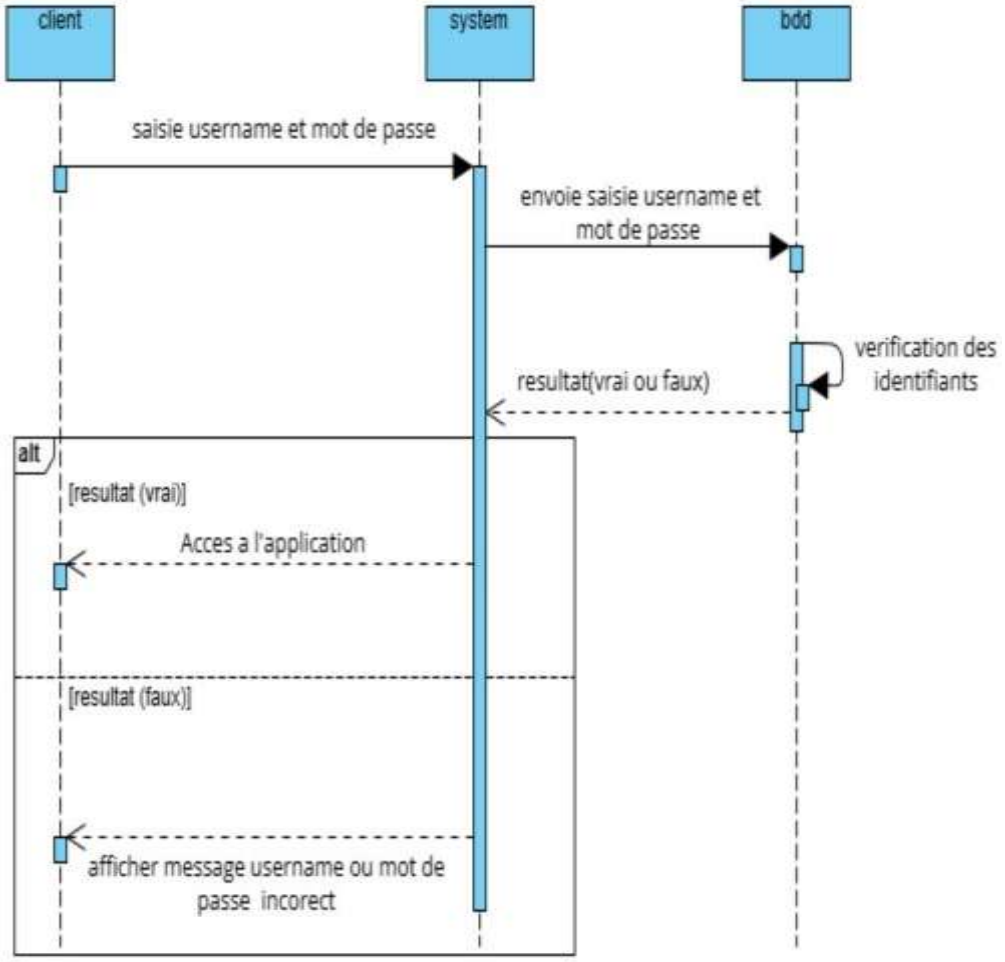


Figure 2.3 – diagramme de séquence d’authentification client[6]

2.6 Diagramme des classes

Un diagramme de classe est un diagramme statique du langage UML qui permet de représenter la structure d’un système en montrant les classes, leurs attributs, leurs méthodes ainsi que les relations entre ces classes (association, héritage, composition, etc….)

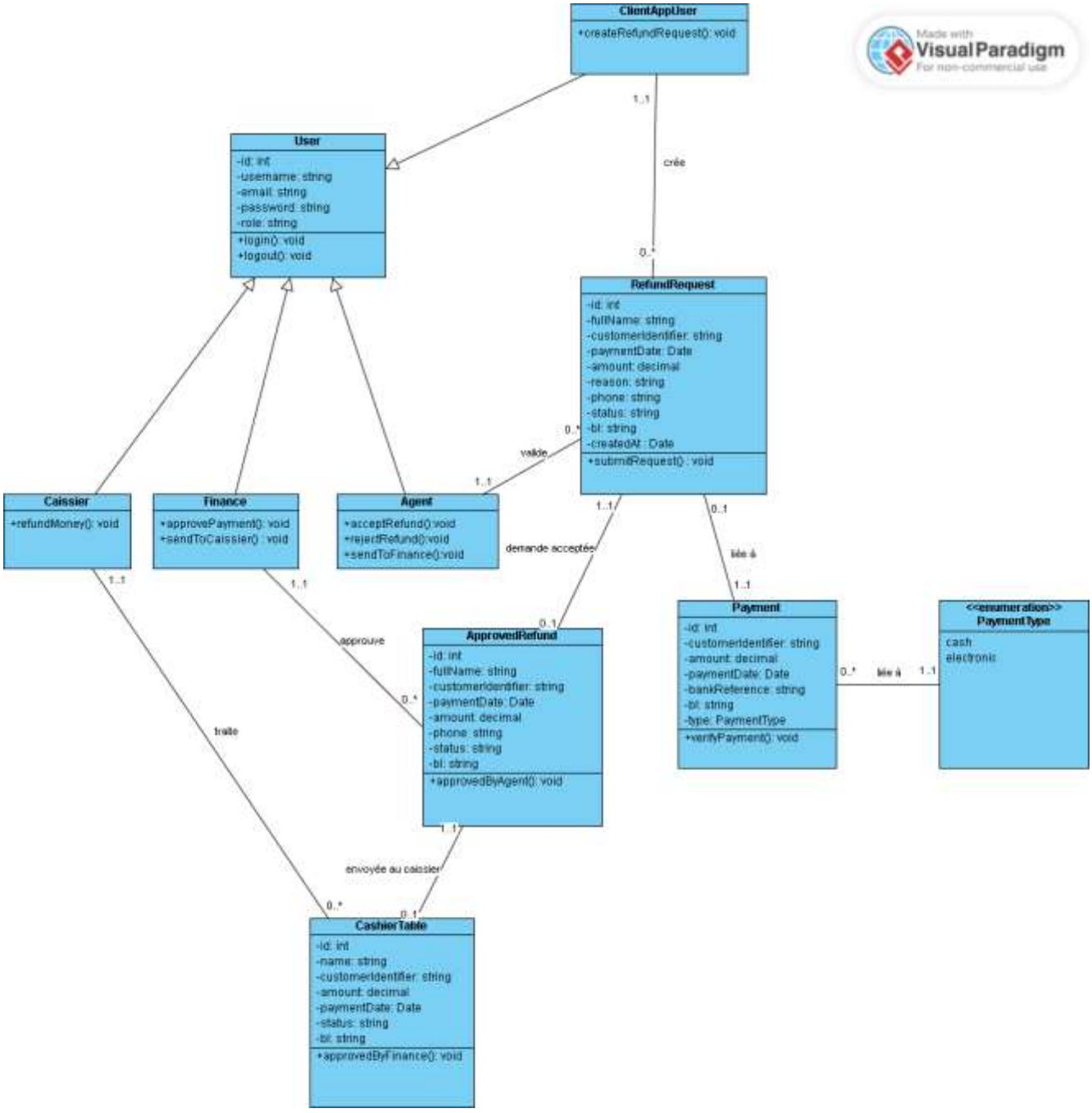


Figure 2.4 – diagramme de classe[7]

2.7 Model relationnelle

Modèle Relationnel – Refund Management System :

- user (id*, customer_identifier, username, password, role)
- user_App (id*, full_name, username, password)
- Payments(id*,#customer_identifier,amount,payment_date, bank_reference,bl,type)
- Refund_Request (id*, full_name, #customer_identifier, payment_date, amount, reason, phone, created_at, status, bl)
- Approved_Refund (id*, full_name, #customer_identifier, payment_date, amount, phone, created_at, status, bl)
- Caissier_Table (id*, name, #customer_id, amount, payment_date, status, bl)

Légende :

-Clé primaire : attribut * ;

-Clé étranger : #attribut ;

2.8 Conclusion

Dans ce chapitre nous avons commencé par la présentation de la méthode de modélisation UML puis l'élaboration des différents diagrammes (cas d'utilisation, séquence, classe) plus la réalisation du dictionnaire de données pour regrouper et classer les données de notre projet et finalement nous sommes passés du modèle conceptuel vers le modèle relationnel de données.

Chapitre 3

Implémentation

3.1 introduction

Dans ce dernier chapitre nous allons présenter la réalisation de notre projet. la présentation des différentes fonctionnalités de notre plate forme et les outils utiliser pour développer cette application

3.2 Environnement de développement matériel

Nom de la Machine	Lenovo Thinkpad L440	Dell Latitude 5480	HP EliteBook 845 G10	Dell Vostro 3400
Processeur	Intel(R) Core(™)i5-4300M @ 2.60GHz	Intel(R) Core(™)i5-7200U@ 2.50GHz	AMD Ryzen 5 PRO 7540U 3.20 GHz	intel(R) Core(™)i5-1135G7@ 2.40GHz
Mémoire RAM	12 GB	8 GB	16 GB	8 GB
Carte graphique	Intel(R) HD Graphics 4600	Intel(R) HD Graphics 620	AMD Radeon 740M Graphics	Intel(R) Iris(R) Xe Graphics
Système d'exploitation	Windows 10 Pro	Windows 11	Windows 11	Windows 11

3.3 Environnement de développement logiciel

3.3.1 Langages de programmation

- **PostgreSQL :** base de données SQL open-source qui stocke les données sous forme de tableaux.[3]



- **Express :** utilisé pour le développement backend des sites Web et est utilisé avec Node.js. Il gère toutes les fonctions comme le reroutage, le HTTPS et la gestion des erreurs.[4]



- **ReactJS :** React.js est une bibliothèque JavaScript développée par Meta (ex-Facebook). Elle sert à construire des interfaces utilisateur interactives [5].



- **Node.JS :** Node.js est un environnement d'exécution JavaScript open-source qui permet aux développeurs d'exécuter du code JavaScript côté serveur. [6]



- **JavaScript :** JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web.[7]



- **CSS** (cascading style sheets) : Est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par « feuilles de style en cascade ». Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML [8]



- **Npm (Node Package Manager) :** est le gestionnaire de paquets officiel de Node.js. Il permet aux développeurs d'installer, de partager et de gérer facilement des bibliothèques et des dépendances nécessaires au développement des applications JavaScript côté serveur.[9]



- **Dart () :** Dart est un langage de programmation développé par Google pour créer des applications web, mobiles et desktop, Il est principalement utilisé avec Flutter afin de développer des applications multiplateformes avec un seul code source. [10]



Flutter : Flutter est un kit de développement logiciel (SDK) d'interface utilisateur open-source créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code.[11]



3.3.2 Outil de développement

- **Visual Studio Code :** l'éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOs.[12]



- **PgAdmin 4 :** outil de gestion Open Source pour PostgreSQL. Il fournit une interface graphique pour simplifier la création, la maintenance et l'utilisation des objets de votre base de données.[13]



- **GitHub** : GITHUB est une plateforme web d'hébergement et de gestion de développement de logiciel cette dernière utilise GIT comme logiciel de gestion, elle est utilisée comme un dépôt de données sécurisé pour des projets de développement logiciel prive ou publique [14]



- **GIT** : est un logiciel de gestion de versions (Version Control System) qui suit l'évolution des fichiers sources et garde les anciennes versions de chacun d'eux sans rien écraser. Cela permet de retrouver les différentes versions d'un fichier ou d'un lot de fichiers connexes et ainsi éviter des problèmes.[15]



- **Android Studio :** Android Studio est l'environnement de développement officiel pour Android, qui fournit tous les outils nécessaires pour créer des applications destinées à différents types d'appareils Android.[16]



- **Visual Paradigm Online :** est une plateforme en ligne permettant de créer et gérer des diagrammes UML (classes, cas d'utilisation, etc.) directement depuis un navigateur, sans installation.[17]



3.4 Présentation de l'application mobile :

- **Sign up et log in**



Figure 3.1 – l'interface de sign-up et login de l'application mobile[8]

- **Formule de demande**

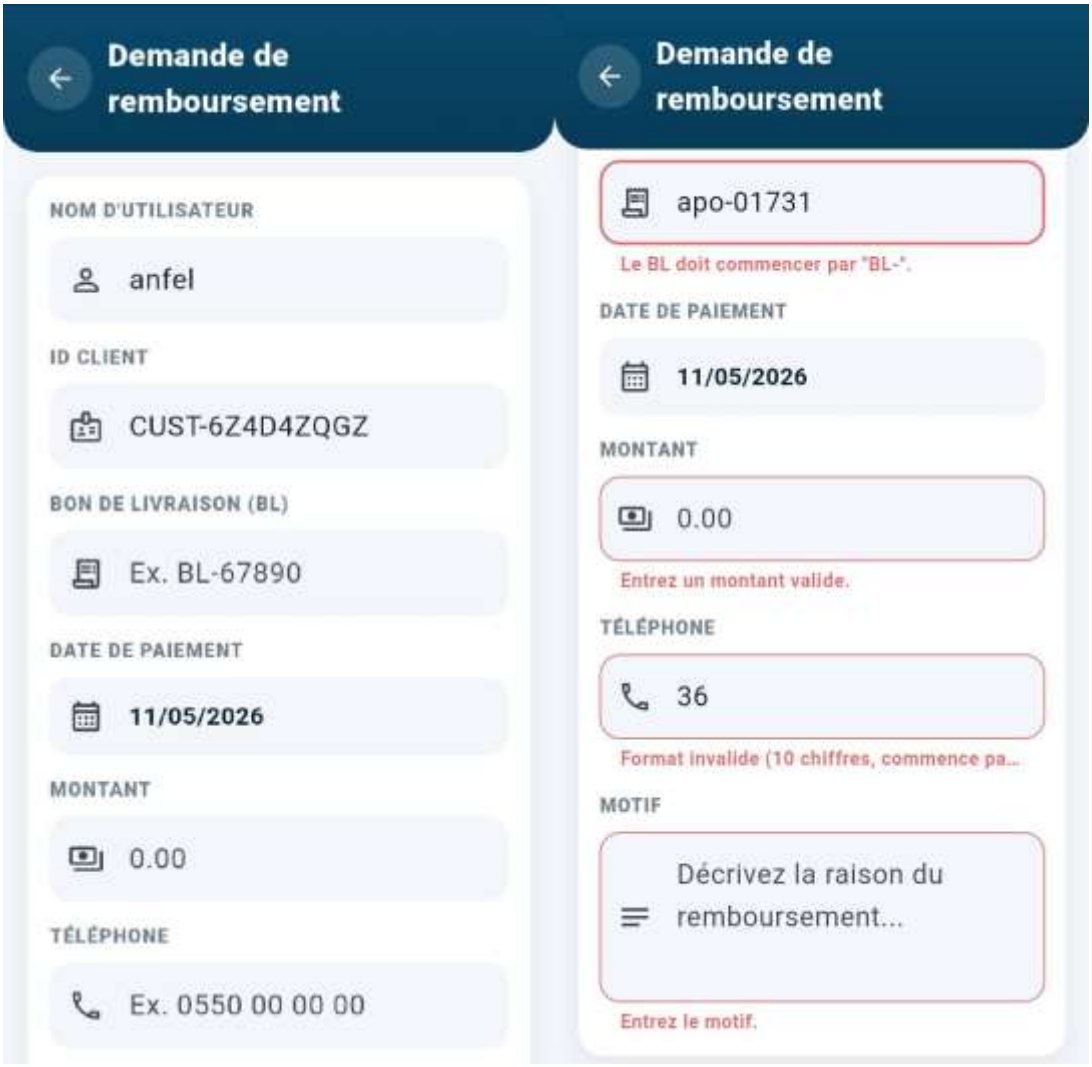


Figure 3.2 – l'interface de demande de remboursement[9]

- **Demande envoyée**



Figure 3.3 – demande envoyée[10]

- **Page d’accueil** **Déconnexion**

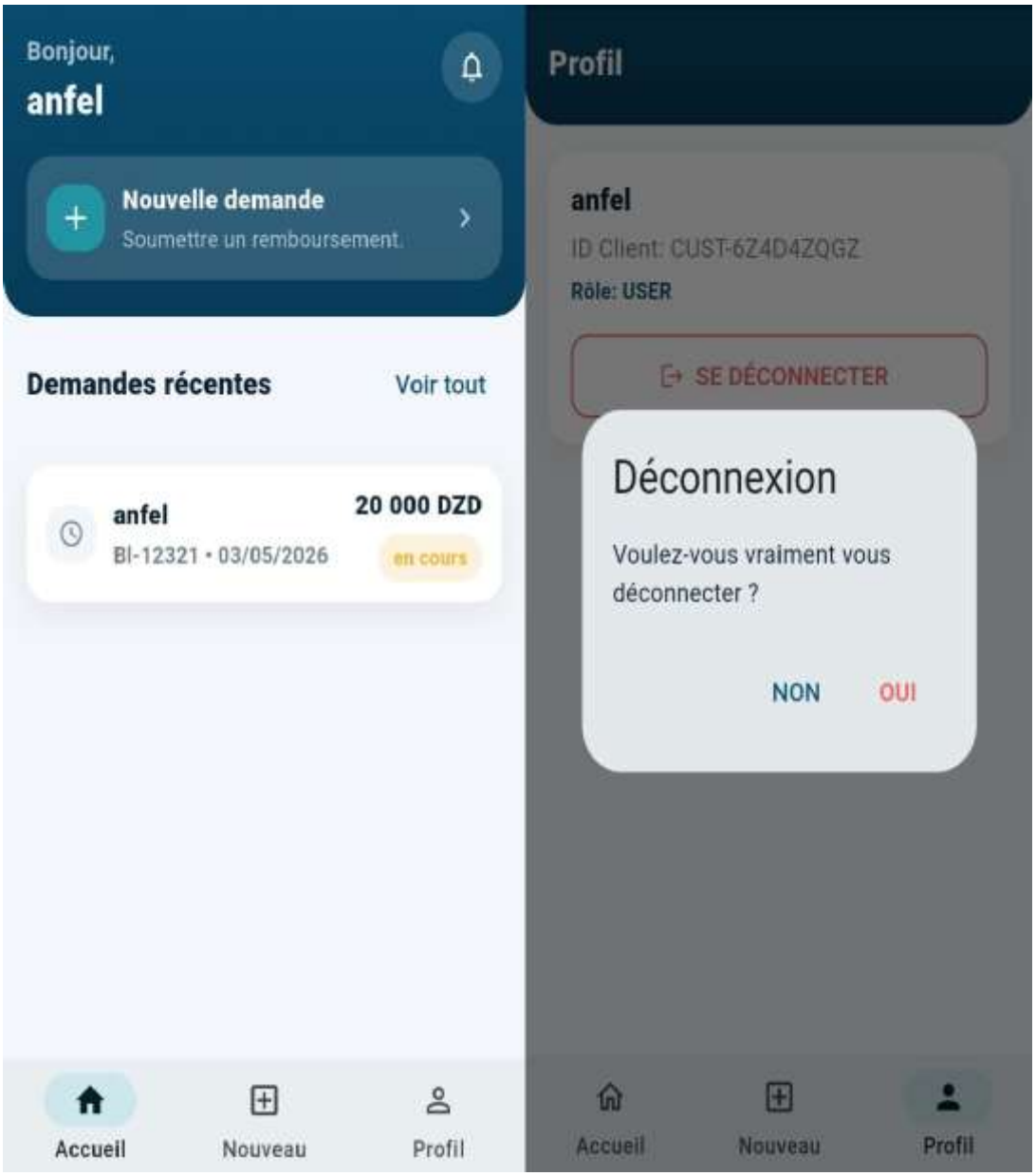


Figure 3.4 – L’interface de la page d’accueil et de la page de déconnexion[11]

3.5 : Présentation de l’application web

3.5.1 Présentation de l’interface login

La figure suivante montre la page de connexion du site web, utilisée par les différents utilisateurs (finance, agent, caissier et administrateur) pour s’authentifier.

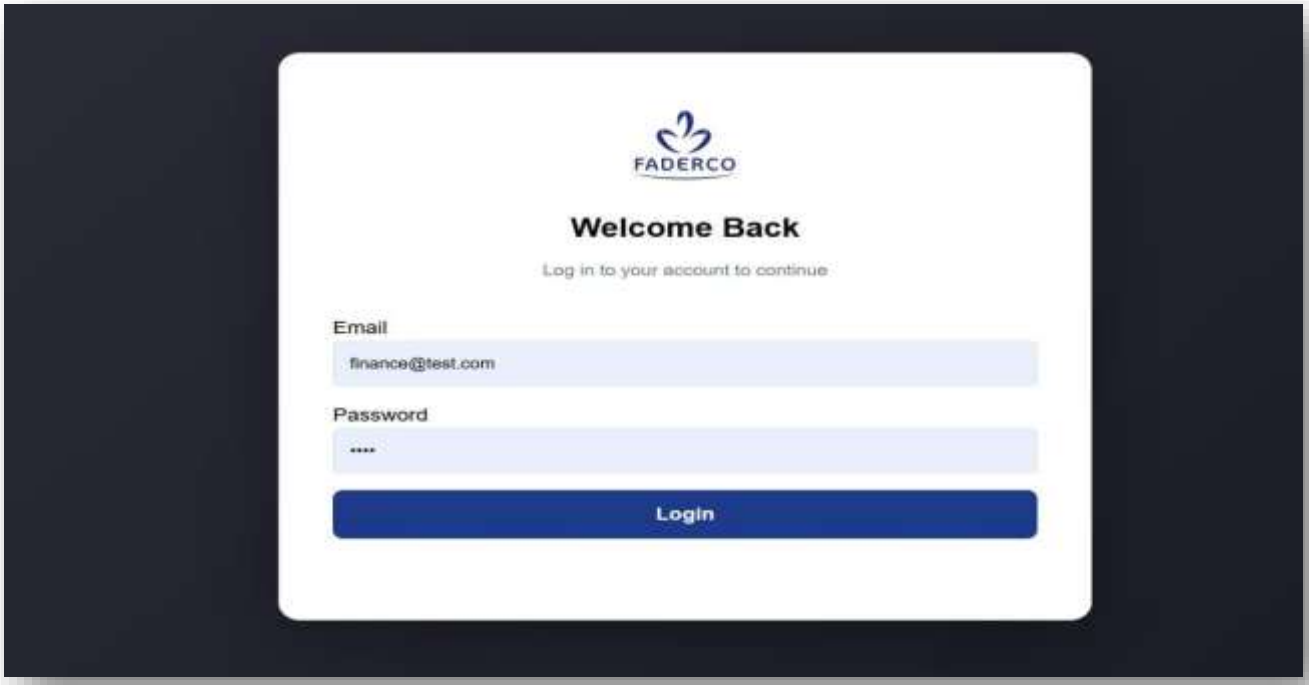


Figure 3.5 – L’interface de login[12]

3.5.2 Présentation de l’interface Administrateur

L’interface administrateur est une plateforme web destinée à la gestion des comptes utilisateurs.

Grâce à cette interface, l’entreprise peut centraliser les informations, améliorer le traitement des demandes et assurer une meilleure traçabilité des opérations de remboursement.

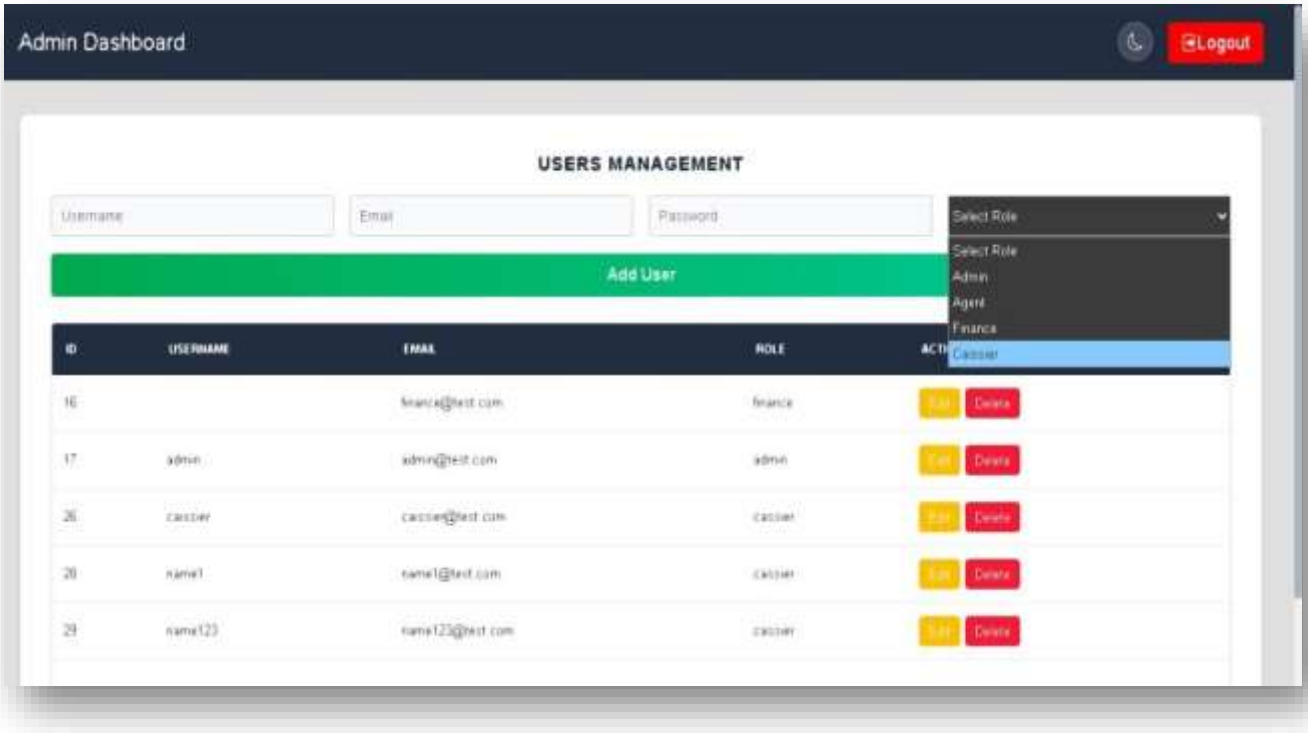


Figure 3.6 – l'interface de l'admin[13]

3.5.3 Présentation de l'interface agent

L'agent reçoit les demandes de remboursement envoyées par les clients.

Les demandes sont affichées dans le tableau de gauche de l'interface.

Le système compare ensuite les informations de la demande avec les données enregistrées dans la base de données de l'application Presalio afin de vérifier l'existence d'un éventuel double paiement.

Si un double paiement est détecté, les transactions concernées sont affichées dans le tableau de droite pour permettre à l'agent de les analyser.

Après vérification, l'agent peut :

- Approuver la demande à l'aide du bouton « Send to Finance » afin de transmettre au service financier ;
- Rejeter la demande à l'aide du bouton « Reject » si la demande est invalide.

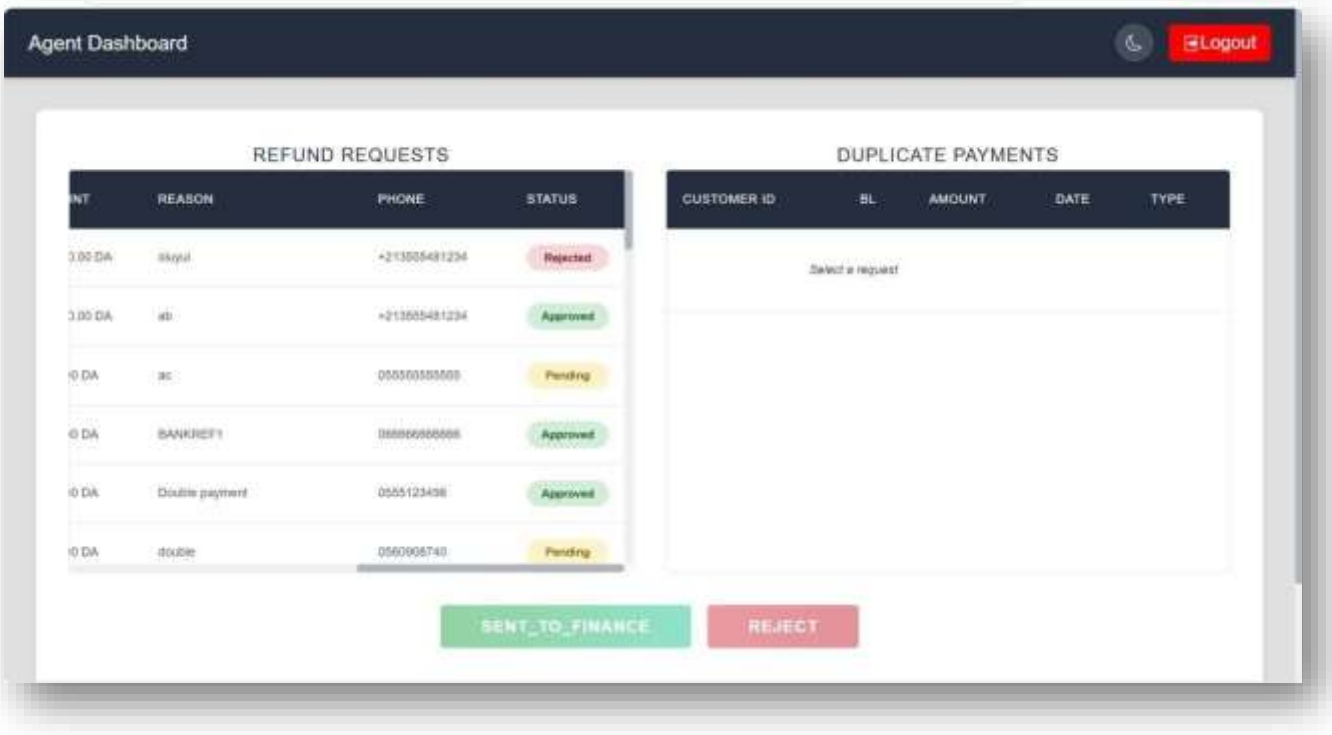


Figure 3.7 – L'interface de l'agent[14]

3.5.4 Présentation de l'interface finance

Le service financier reçoit les demandes de remboursement approuvées par l'agent de traitement.

Les demandes sont affichées dans le tableau de gauche de l'interface financière.

Le système compare ensuite les informations des transactions avec le fichier bancaire transmis quotidiennement par la banque afin de vérifier l'existence d'un éventuel double paiement dans les opérations enregistrées.

Si un double paiement est détecté, les transactions concernées sont affichées dans le tableau de droite afin de permettre au service financier de les analyser et de confirmer la validité de la demande.

Après vérification, le service financier peut :

- Approuver la demande à l'aide du bouton « Approve » afin de la transmettre au service de caisse ;
- Rejeter la demande à l'aide du bouton « Reject » si la demande est jugée invalide.

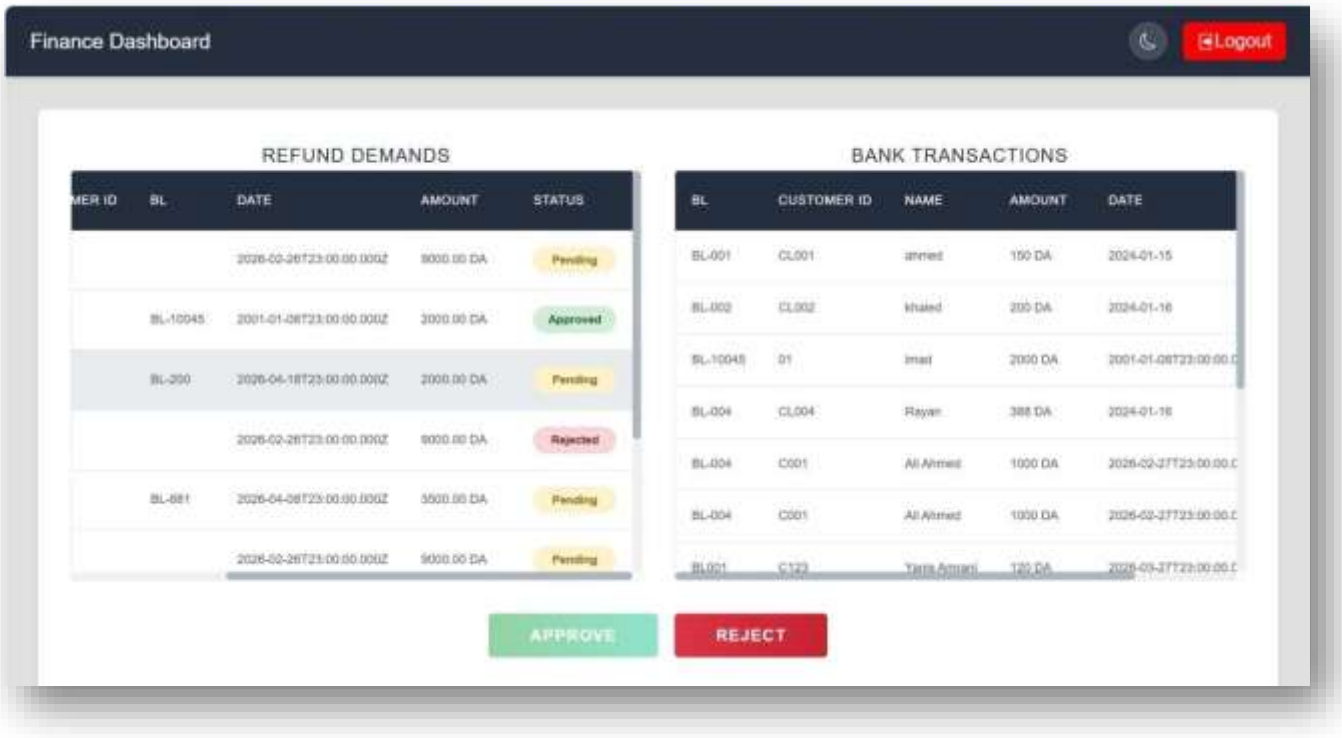


Figure 3.8 – L'interface de finance[15]

3.5.5 Présentations de l'interface caissier

Le caissier est responsable de l'exécution manuelle des opérations de remboursement des clients.

Les demandes affichées dans l'interface du caissier correspondent uniquement aux demandes validées au préalable par le service financier.

Après avoir effectué le remboursement, le caissier peut sélectionner les clients concernés puis confirmer l'opération à l'aide du bouton « Approve ».

L'interface permet également au caissier de consulter la liste complète des remboursements effectués. En cliquant sur le bouton « Download Excel », le système génère automatiquement un fichier Excel contenant les informations des clients remboursés ainsi que les détails des opérations réalisées.

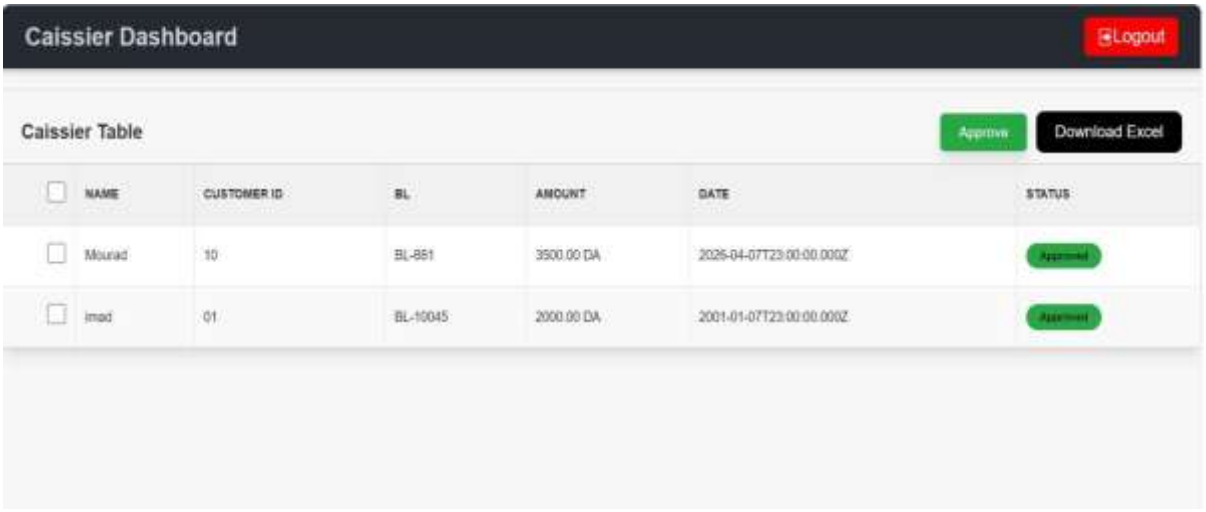


Figure 3.9 – L'interface de caissier[16]

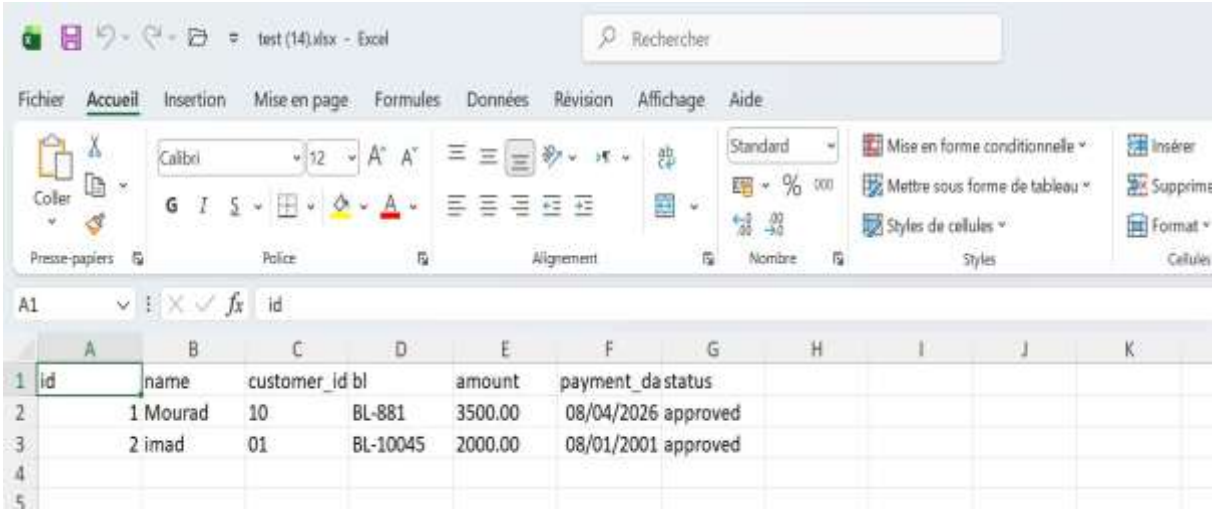


Figure 3.10 – Fichier Excel des clients remboursés[17]

3.6 Conclusion

Dans ce dernier chapitre, nous avons présenté les différentes outils utiliser pour développer notre application ensuite on a présenter les interfaces résumant notre projet de manière visuelle à travers quelques résultats graphiques avec une brève description de chaque photo on arrive à la fin de notre mémoire de fin d'étude en élaborant une dernière conclusion générale pour notre travail

Conclusion Générale

Au terme de ce travail, nous pouvons affirmer que les objectifs fixés au départ ont été atteints. En effet, ce projet nous a permis de concevoir et de développer une solution informatique répondant à une problématique concrète rencontrée au sein de l'entreprise Faderco, à savoir la gestion des cas de double paiement survenant lors des transactions électroniques effectuées via l'API Guiddini Epay. Grâce à la mise en place d'un système de vérification à plusieurs niveaux, d'un mécanisme d'authentification sécurisé basé sur JWT et RBAC, ainsi que d'une traçabilité complète de chaque demande depuis sa soumission jusqu'à son traitement final, la solution développée garantit fiabilité, transparence et efficacité dans le traitement des réclamations clients. Elle permet également de faciliter la communication entre Agent → Finance → Caissier, tout en réduisant le travail manuel et le désordre entre les différents services. Sur le plan personnel, ce projet a représenté une expérience enrichissante qui nous a permis de mettre en pratique les connaissances acquises tout au long de notre formation. Il nous a confrontés aux réalités du développement logiciel en entreprise, depuis la phase d'analyse et de conception jusqu'à la réalisation et le déploiement des interfaces, en passant par la modélisation UML et l'intégration de technologies modernes.

Bibliographie

[1] Faderco, Présentation de l'entreprise Faderco". des systèmes logiciels

Disponible sur : https://www.faderco.dz/

[2] Booch, G., Rumbaugh, J., & Jacobson, I. (2005).

The Unified Modeling Language User Guide (2nd Edition). Addison-Wesley.

➤ Livre de référence sur UML, écrit par ses créateurs, utilisé pour modéliser des systèmes logiciels

[3] PostgreSQL Global Development Group, "PostgreSQL Documentation".

Disponible sur : https://www.postgresql.org/docs/

[4] Express.js, "Express - Node.js Web Application Framework".

Disponible sur : https://expressjs.com/

[5] Meta Platforms, "React – A JavaScript library for building user interfaces".

Disponible sur : https://react.dev/

[6] Node.js Foundation, "Node.js Documentation".

Disponible sur : https://nodejs.org/en/docs/

[7] JavaScript, "Guide de référence du langage JavaScript.

Disponible sur : https://developer.mozilla.org/fr/docs/Web/JavaScript

[8] CSS "Feuilles de style en cascade (Cascading Style Sheets)".

Disponible sur : https://developer.mozilla.org/fr/docs/Web/CSS

[9] npm, "npm Documentation".

Disponible sur : https://docs.npmjs.com/

[10] Référence : https://dart.dev/overview

[11] Flutter, "Documentation technique de Flutter".

Disponible sur : https://fr.wikipedia.org/wiki/Flutter_(logiciel)

[12] Microsoft, "Visual Studio Code Documentation".

Disponible sur : https://code.visualstudio.com/docs

[13] pgAdmin Development Team, "pgAdmin 4 Documentation".

Disponible sur : https://www.pgadmin.org/docs/

[14]GitHub, “GitHub Documentation”.

Disponible sur : https://docs.github.com/

[15] Git, “Git Documentation”.

Disponible sur : https://git-scm.com/docs

[16] : chapter1 de module application mobile L3 si 2025-2026

[17] : https://online.visual-paradigm.com