



République Islamique de Mauritanie  
Ministre de l'Enseignement Supérieur et de la  
Recherche Scientifique



## MÉMOIRE DE STAGE DE FIN D'ÉTUDES

Présenté en vue de l'obtention du  
Diplôme de Master Professionnel en Informatique  
Appliquée à la Gestion (MPIAG)

Par :

Mohamed Lemine Salem M'bedah (IE18689)

## Thème

---

CREATION D'UN SYSTEME DE GESTION DE  
RELATION CLIENT (CRM) ET LA  
CONNAISSANCE CLIENT (KYC)

---

Encadré par :

Dr. Cheikh Dhib

Mr. Mboirick Mohamed

Réalisé au sein de CADORIM



Année universitaire 2021-2022

# **DEDICACES**

Je dédie ce travail à :

**Mes frères et sœurs ;**

**Toute ma famille, qu'elle soit proche ou lointaine ;**

**Tous mes amis et camarades de classes pour leur aide et conseils ;**

**Mon tuteur pour son accueil chaleureux tout au long de mon  
cursus universitaire ;**

**Mon oncle depuis les Etats-Unis pour son soutien et ses  
encouragements durant tout mon cursus universitaire ;**

# REMERCIEMENTS

*Ce n'est pas parce que la tradition l'exige que cette page se trouve dans ce rapport ; mais par ce que les gens à qui s'adressent mes remerciements les méritent vraiment.*

*J'adresse mes remerciements les plus chaleureux envers :*

- *Dr. Cheikh Dhib, l'ancien coordinateur de MPIAG, ainsi que Dr. Emani Mohamed Sidi le nouveau coordinateur et l'ensemble du corps professoral et administratif de l'ISCAE. Et pour avoir suivi et dirigé l'évolution de mon travail, ainsi que les encouragements, conseils et orientations réguliers qu'il m'a prodigués. Je vous suis infiniment reconnaissant pour votre grande disponibilité et l'intérêt que vous avez porté à ce travail ;*
- *Mr. Mboirick Mohamed, responsable technique et mon maître de stage, pour ça disponibilité, sa gentillesse, son attention particulière à l'égard de ce travail. Et pour avoir mis les moyens nécessaires au bon déroulement de ce stage de six mois et à la réalisation de ce travail ; Je vous suis infiniment reconnaissant pour votre grande disponibilité et l'intérêt que vous avez porté à ce travail ;*

*Toutes les personnes qui de près ou de loin, ont contribué à la réalisation de ce travail.*

# **AVANT-PROPOS**

L’Institut Supérieur de Comptabilité et d’Administration des Entreprises est un établissement public d’enseignement supérieur et de recherche, placé sous la tutelle du Ministère en charge de l’Enseignement Supérieur, créé en 2009, par décret N° 2009-161 du 29 avril 2009.

L’ISCAE compte deux (2) départements et dans le cadre de leur formation, les étudiants qui sont en fin de cycle sont tenus d’effectuer un stage pratique au sein d’une entreprise ou d’un service informatique.

# RESUME

Le présent document est le fruit de notre travail dans le cadre du stage de fin d'études, pour l'obtention du diplôme de master Professionnel en informatique Appliquée à la gestion (MPIAG), qui a été réalisé au sein de la société CADORIM . Notre travail consiste à étudier les Framework Laravel et Flutter, afin de réaliser deux applications mobiles. L'une est une nouvelle version du projet mobile CADORIM permettant la gestion de transfert d'argent ,et identifier électroniquement des documents comme des pièces d'identité (carte d'identité, passeport et carte de séjour),et une fenêtre de discussion. Et l'autre application ce pour la gestion de service client.

Afin de suivre l'évolution des techniques de l'information et l'émergence du monde mobile, notre application est conçue pour les différentes plateformes mobiles à savoir :Android, IOS.

Nous avons commencé par l'élaboration d'une étude fonctionnelle qu'on a fini par l'élaboration d'une conception du projet.

Les principales fonctionnalités de CADORIM sont l'inscription des membres, le transfert d'argent, la communication avec le service client.

# **Abstract**

This document is the result of our work in the end study's internship, in order to obtain engineering degree in computer science applied to management, which has been done within company CADORIM. Our mission is to study two mobile development framework(Flutter,Laravel) and realize two mobile project, one is a new version of CADORIM application and other is for management of clients relationship.

To follow the evolution of information technology and the emergence of the mobile world, our application is designed for different mobile platforms namely Android, IOS.

# LISTE DES ABRÉVIATIONS

Je présente ici certains sigles et abréviations que nous utiliserons dans le document.

**ISCAE** : Institut Supérieur de Comptabilité et d'Administration des Entreprises

**MPIAG** : Master Professionnel en Informatique Appliquée à la Gestion

**API** : Application Programming Interface

**MVC** : Modèle Vue Contrôleur

**HTTP** : Hypertext Transfer Protocol

**HTTPS** : HyperText Transfer Protocol Secure

**UML** : Unified Modeling Language

**SQL** : Structured Query Language

**JSON** : JavaScript Object Notation

**XP** : eXtreme Programming

**IIS** : Internet Information Services

**DAO** : Data Access Object

**AAB** : Android App Bundles

**APK** : Android Package Kit

**IDE** : Integrated Development Environment (Environnement de Développement Intégré)

**MERISE** : Méthode d'Étude et de Réalisation Informatique par les Sous-Ensembles ou pour les Systèmes d'Entreprise

**SGBD** : Système de Gestion de Bases de Données

**SGBDR** : Système de Gestion de Bases de Données Relationnelle

**OCR** : Optical Character Recognition

**MRZ** : Machine-Readable Zone

**UX** : User Experience

**UI** : User Interface

# Table des matières

<b>Introduction générale</b>	<b>11</b>
<b>1 Contexte général du projet</b>	<b>13</b>
1.1 Présentation de l'organisme d'accueil . . . . .	13
1.1.1 Introduction . . . . .	13
1.1.2 Missions . . . . .	14
1.1.3 Organigramme . . . . .	14
1.2 Cadre général du projet . . . . .	15
1.2.1 Problématique . . . . .	15
1.2.2 Objectifs du projet . . . . .	15
1.2.3 Planification du projet . . . . .	16
<b>2 Analyse fonctionnelle et conceptuelle</b>	<b>18</b>
2.1 Analyse fonctionnelle . . . . .	18
2.1.1 Diagramme de cas d'utilisation . . . . .	18
2.1.2 Diagramme d'activité . . . . .	19
2.1.2.1 Création de compte . . . . .	20
2.1.2.2 Transfert d'argent . . . . .	21
2.1.2.3 Authentification . . . . .	22
2.2 Modélisation de la base de données . . . . .	23
2.2.1 Diagramme de classe . . . . .	23
<b>3 L'implémentation</b>	<b>24</b>
3.1 Environnement de travail . . . . .	24
3.1.1 Environnement logiciel . . . . .	24
3.1.1.1 Editeur de texte : VS Code . . . . .	24

3.1.1.2	IDE : Android Studio . . . . .	25
3.1.1.3	Git . . . . .	25
3.1.1.4	Le SGBD Firebase . . . . .	26
3.1.1.5	SGBD MariaDB . . . . .	29
3.1.1.6	Logiciel de modélisation : Modelio . . . . .	29
3.1.1.7	Motifs d'architecture logicielle . . . . .	30
3.1.1.8	Serveur Web : Amazon Web Services (AWS) . . . . .	31
3.1.2	Conception . . . . .	32
3.1.2.1	Langage de modélisation : UML . . . . .	32
3.2	Choix des outils de travail . . . . .	32
3.2.1	Langages utilisés . . . . .	32
3.2.1.1	Dart . . . . .	32
3.2.1.2	PHP . . . . .	33
3.2.2	Frameworks utilisés . . . . .	33
3.2.2.1	Flutter . . . . .	33
3.2.2.2	Laravel . . . . .	35
3.3	Implémentation . . . . .	36
3.3.1	Etape de réalisation . . . . .	36
3.3.2	Interfaces Homme/Machine . . . . .	36
3.3.2.1	Application CADORIM . . . . .	36
3.3.2.2	Application ServiceClient . . . . .	45
<b>Conclusion</b>		<b>49</b>
<b>Bibliographie</b>		<b>50</b>

# Table des figures

1.1 CADOROM . . . . .	13
1.2 Organigramme du CADORIM . . . . .	14
1.3 Diagramme de Gantt du projet . . . . .	16
2.1 Diagramme de cas d'utilisation. . . . .	18
2.2 Diagramme d'activité : Création de compte . . . . .	20
2.3 Diagramme d'activité : Transfert d'argent . . . . .	21
2.4 Diagramme d'activité : Authentification . . . . .	22
2.5 Diagramme d'activité : Transfert d'argent . . . . .	23
3.1 VS Code . . . . .	24
3.2 Android Studio . . . . .	25
3.3 Git . . . . .	25
3.4 Firebase . . . . .	26
3.5 SGBD MariaDB . . . . .	29
3.6 Modelio . . . . .	29
3.7 MVC . . . . .	30
3.8 Amazon Web Services (AWS) . . . . .	31
3.9 UML . . . . .	32
3.10 Dart . . . . .	32
3.11 PHP . . . . .	33
3.12 Flutter . . . . .	33
3.13 Architecture d'une application Flutter. . . . .	34
3.14 Laravel . . . . .	35
3.15 Architecture de Laravel . . . . .	35
3.16 Première interface . . . . .	37

3.17 Interface d'authentification . . . . .	38
3.18 Interface d'inscription . . . . .	39
3.19 Processus de transactions . . . . .	41
3.20 Transferts . . . . .	42
3.21 Interface de discussion . . . . .	43
3.22 Profil de l'utilisateur . . . . .	44
3.23 Interface de menu . . . . .	45
3.24 Interface d'authentification . . . . .	46
3.25 Interface du compte administrateur . . . . .	47
3.26 Interface du compte administrateur . . . . .	48

# Introduction générale

Actuellement, à travers les progrès de la technologie, le smartphone est devenu un outil indispensable de travail qui peut apporter un plus dans la vie professionnelle et sociale de tout un chacun. D'ailleurs cela s'illustre parfaitement dans le domaine du Mobile Money qui est l'une des plus grandes évolutions dans le secteur consacré aux transfert d'argent de l'europe vers l'afrique où une grande partie de la population de l'afrique n'a pas de compte bancaire.

Au fur et à mesure on entend désormais partout, le processus KYC (Know Your Customer) se déploie dans de nombreux secteurs d'activité. Véritable atout pour combattre la fraude et l'usurpation d'identité, un processus KYC, lorsqu'il est correctement déployé, offre aujourd'hui une multitude d'avantages aux utilisateurs comme aux entreprises.

Dans notre société, l'information est devenue un élément à la fois stratégique pour développer les activités, et essentiel pour assurer un avantage concurrentiel (optimisation des coûts, meilleure satisfaction client...) aux entités qui savent l'utiliser. C'est ce constat qui explique pourquoi les entreprises cherchent aujourd'hui à mettre en place des systèmes de collecte et de traitement de données toujours plus performants. De même, la satisfaction du client est plus que jamais au centre des préoccupations des entreprises et se concrétise par une gestion personnalisée de la relation client : comprendre les clients et leurs attentes, les fidéliser, les inciter à consommer davantage. Le CRM, Customer Relationship Management (GRC en français) a pour objet d'identifier, attirer et conserver les meilleurs clients et d'en retirer chiffre d'affaire et rentabilité. Ainsi le CRM englobe l'ensemble des activités et des processus que doit mettre en place une entreprise pour interagir avec ses clients et ses prospects afin de leur fournir des produits et des services adéquats au bon moment. Les entreprises ont de plus en plus recours à une approche de type de CRM, afin de se différencier. En effet, la banalisation de l'offre, une exigence accrue du client conduisent les entreprises à faire évoluer leur offre dans le sens d'une plus grande person-

nalisation. Afin de parvenir à cet objectif, l'entreprise est tenue de s'adapter à la profusion des canaux d'accès parallèles et en particulier Internet. L'arrivée des Nouvelles Technologies de l'Information et de la Communication a en effet un impact très important sur les attitudes et les stratégies des entreprises face au CRM. Si bien que l'on peut se demander si l'E-CRM, la gestion de la relation client par Internet constitue une véritable révolution pour le CRM.

# Chapitre 1

## Contexte général du projet

Dans ce chapitre,nous présenterons le contexte général du projet qui sera décliné en deux parties : la première présentera la société d'accueil, et la seconde décrira le contexte,la problématique,l'objectif attendu du projet ainsi que les fonctionnalités à mettre un place et la planification de projet.

### 1.1 Présentation de l'organisme d'accueil

#### 1.1.1 Introduction



FIGURE 1.1 – CADOROM

CADORIM est une société de transfert d'argent mauritanienne basée à Nouakchott, fondée fin 2018 par un entrepreneur mauritanien, titulaire d'un doctorat en mathématiques, CADORIM consiste a transférer de l'argent depuis n'importe quel pays dans le monde vers ses proches en Mauritanie. Notre objectif et de fourni une plateforme numérique permet à l'utilisateur de régler ses commandes en toute sécurité et confidentialité assurée par le service de PayPal qui est mondialement connu pour sa fiabilité et simplicité.t Pour effectuer un paiement il suffit d'une simple carte bancaire ou un compte PayPal . et une éventuelle possibilité de virement bancaire. CADORIM a été élu comme le champion

de Banque Centrale de Mauritanie (BCM) 1ère édition 2019 Fintech Challenge, Le siège social de CADORIM est situé à marche capital , Nouakchott, Mauritanie, immatriculée au registre du commerce.

### 1.1.2 Missions

CADORIM offre une large palette de prestations organisées autour des activités suivantes :

1. Maintenance et amélioration de leurs propres applications (CadoRim et MauriPay)
2. Développement des applications
3. Des agences des reçoivent d'argent et de service client

### 1.1.3 Organigramme

La structure organisationnelle de CADORIM comprend :

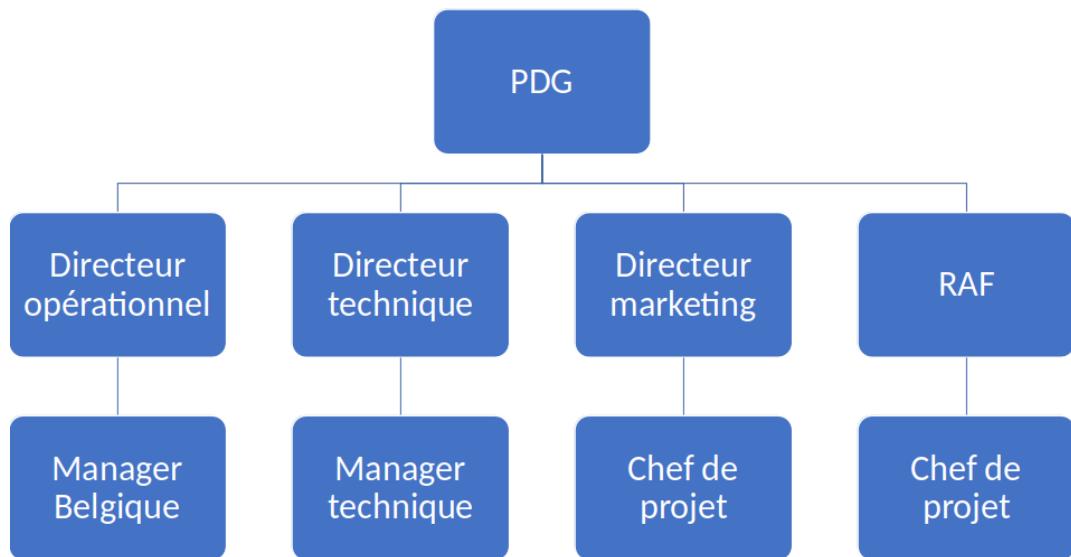


FIGURE 1.2 – Organigramme du CADORIM

## 1.2 Cadre général du projet

### 1.2.1 Problématique

Pour illustrer la couverture contre le risque de la fraude(de blanchiment, identité...),les sanctions(à cause du non respect de la réglementation),le risque juridique et fiscal. Et puisque que les sociétés réalisent facilement plus de ventes grâce à l'utilisation d'un system CRM(Gestion de la Relation Client).

Agissant avec lucidité et consciente des évolutions que connaît son domaine, la société CADORIM a décidé de développer deux systèmes l'un pour lutte contre les fraudes financières, les usurpations d'identité, le blanchiment et il permet aussi de mieux connaître ses clients et d'en analyser les risques. En plus il est aussi bien sûr un gage de sécurité supplémentaire offerte aux clients contre toute usurpation d'identité ou détournement d'un compte CADORIM. cette system quand il est bien implémenté, permet non seulement de prévenir les fraudes mais il va également améliorer l'expérience client en la rendant la vérification des informations plus simple et fluide, cette system est connue sous le nom KYC. Et l'autre collecte, enregistre et classifie les données liées aux clients d'une entreprise. Ses fonctionnalités permettent de gérer certains processus des fonctions ventes, clientèle et marketing. CRM permettant à une entreprise d'obtenir :

- \* Une base de données clients partagée.
- \* Une plate-forme de travail pour les services commerciaux et marketing.
- \* Une source d'indicateurs quantitatifs pour piloter la gestion de l'activité.

### 1.2.2 Objectifs du projet

L'objectif principal du projet est de concevoir et développer une solution mobile dont le but est de faciliter aux futurs utilisateurs la gestion de transfert d'argent. Le résultat de ce travail doit répondre aux objectifs fixés. Notre objectif général se décompose en différents objectifs spécifiques à savoir :

- \* **Faciliter le parcours d'inscription :** Le client renseigne un certain nombre d'informations via un formulaire. Ensuite, le client scan un document d'identité, via la caméra de son smartphone . Les informations sont ensuite analysées automatiquement.

- \* **Faciliter le transfert d'argent** : Avec son téléphone, l'utilisateur pourra envoyer l'argent à tout moment sans perdre du temps.
- \* **Réduire les coûts de transfert** : Avec une telle solution, l'idée est de pouvoir envoyer de petites sommes d'argent donc il n'y aura pas lieu de payer des frais de transfert exorbitants.
- \* **Participer à la diversification des services de transfert d'argent** : Vu le nombre de système de transfert d'argent déjà existant, l'idée n'est pas ici d'être leur concurrent mais plutôt un complément des services qui sont sur le marché pour une bonne satisfaction des clients.
- \* **Ajouter une fenêtre de discussion** : Pour améliorer l'expérience client .

### 1.2.3 Planification du projet

J'effectuais le diagramme de Gantt, pour avoir une meilleure compréhension de la chronologie des étapes de mon projet.

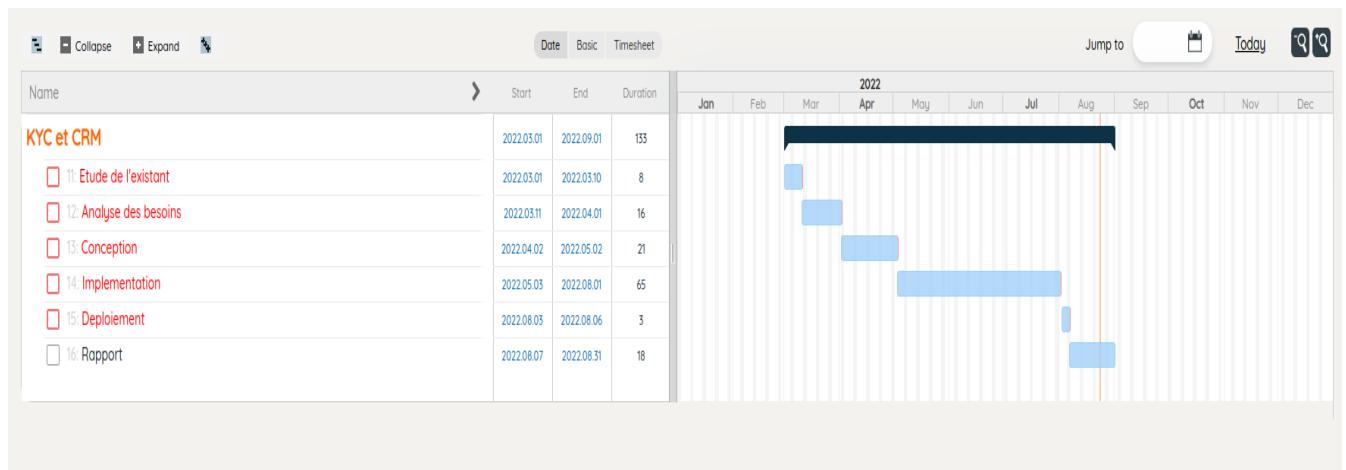


FIGURE 1.3 – Diagramme de Gantt du projet

Le projet est subdivisé en plusieurs phases.

1. Une phase comprenant l'étude de l'existant et analyse des besoins, en intervenant les différents acteurs du projet.
2. Une phase de conception consistant à modéliser et formaliser les données brutes du cahier de charge

3. Une phase d'implémentation consiste à traduire techniquement les données provenant de la conception.
4. Déploiement : externalisation des ressources.

# Chapitre 2

## Analyse fonctionnelle et conceptuelle

### 2.1 Analyse fonctionnelle

#### 2.1.1 Diagramme de cas d'utilisation

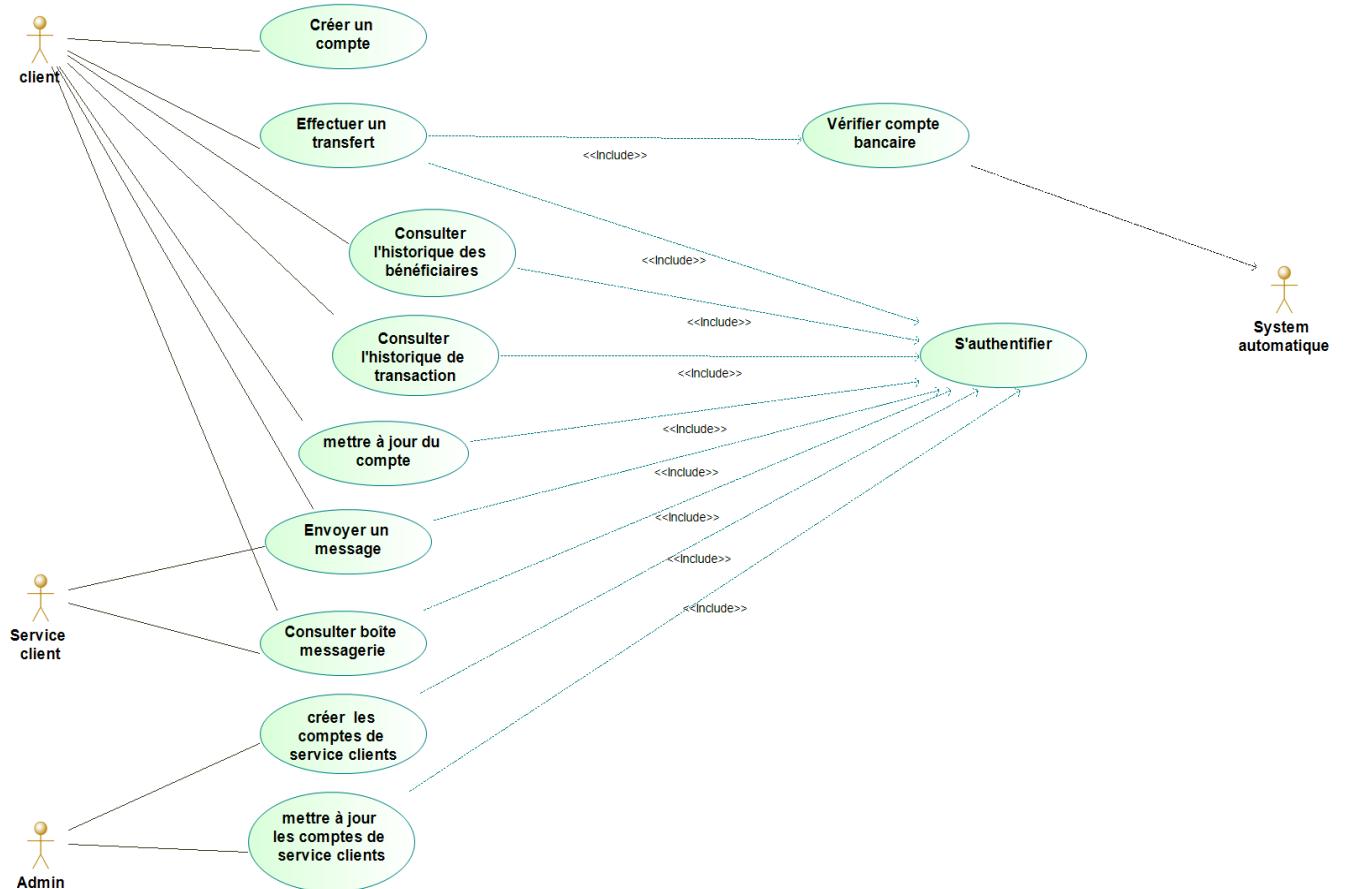


FIGURE 2.1 – Diagramme de cas d'utilisation.

### **2.1.2 Diagramme d'activité**

Cette section a pour objectif de mettre en surbrillance le processus quelques fonctionnalités de l'application pour voir les détails. J'ai choisi trois cas d'utilisation, à savoir, la création d'un compte, la transfert d'argent et l'authentification.

### 2.1.2.1 Cr ation de compte

Le diagramme d'activit  qu'illustre la figure 2.2 d crit le cas d'utilisation « Cr er un compte ».

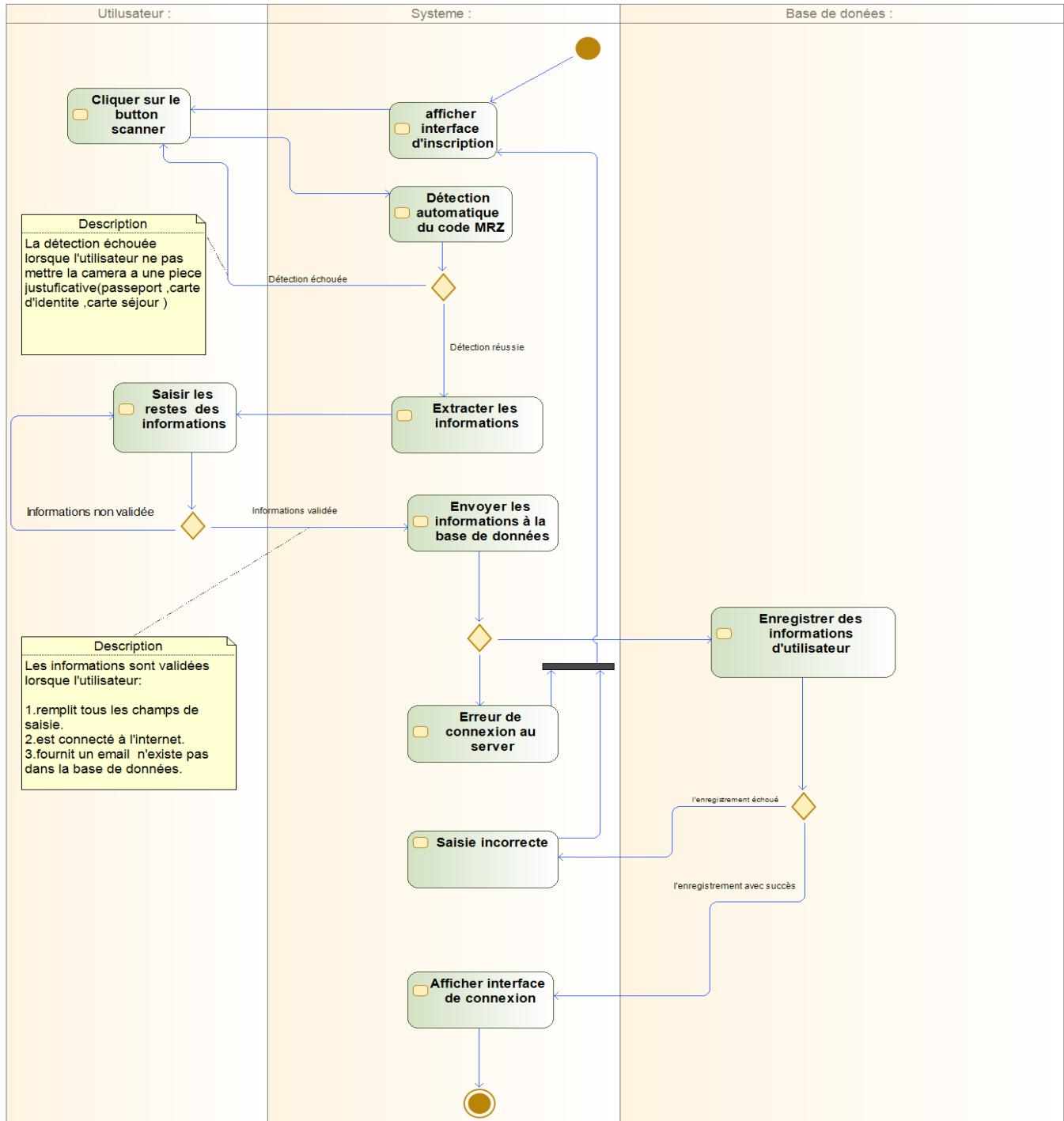


FIGURE 2.2 – Diagramme d'activit  : Cr ation de compte

### 2.1.2.2 Transfert d'argent

Le diagramme d'activité qu'illustre la figure 2.3 décrit les différentes actions ou enchainements effectués lors d'une opération de transfert d'argent.

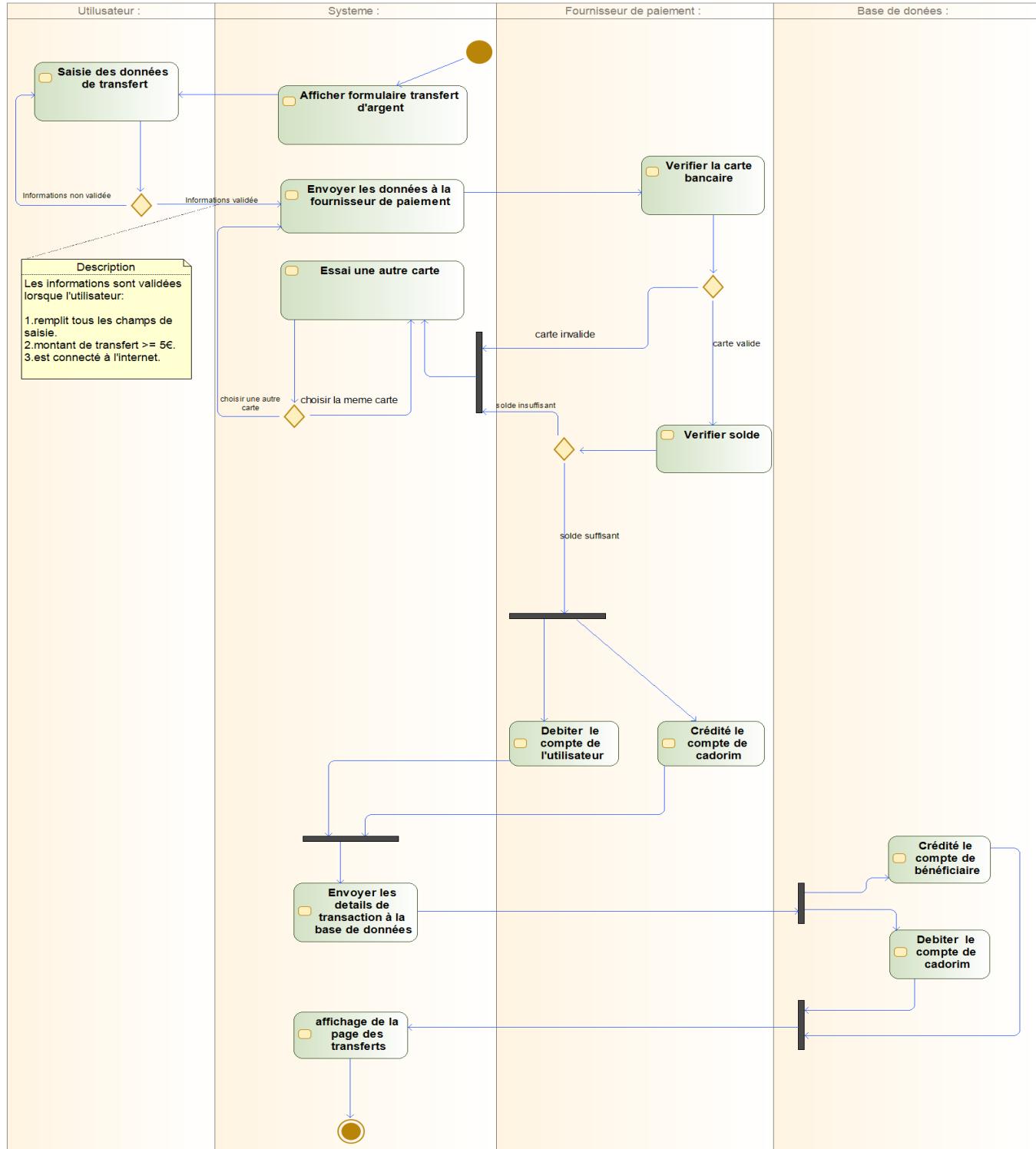


FIGURE 2.3 – Diagramme d'activité : Transfert d'argent

### 2.1.2.3 Authentification

Le diagramme d'activité qu'illustre la figure 2.4 décrit le cas d'utilisation « Authentification ».

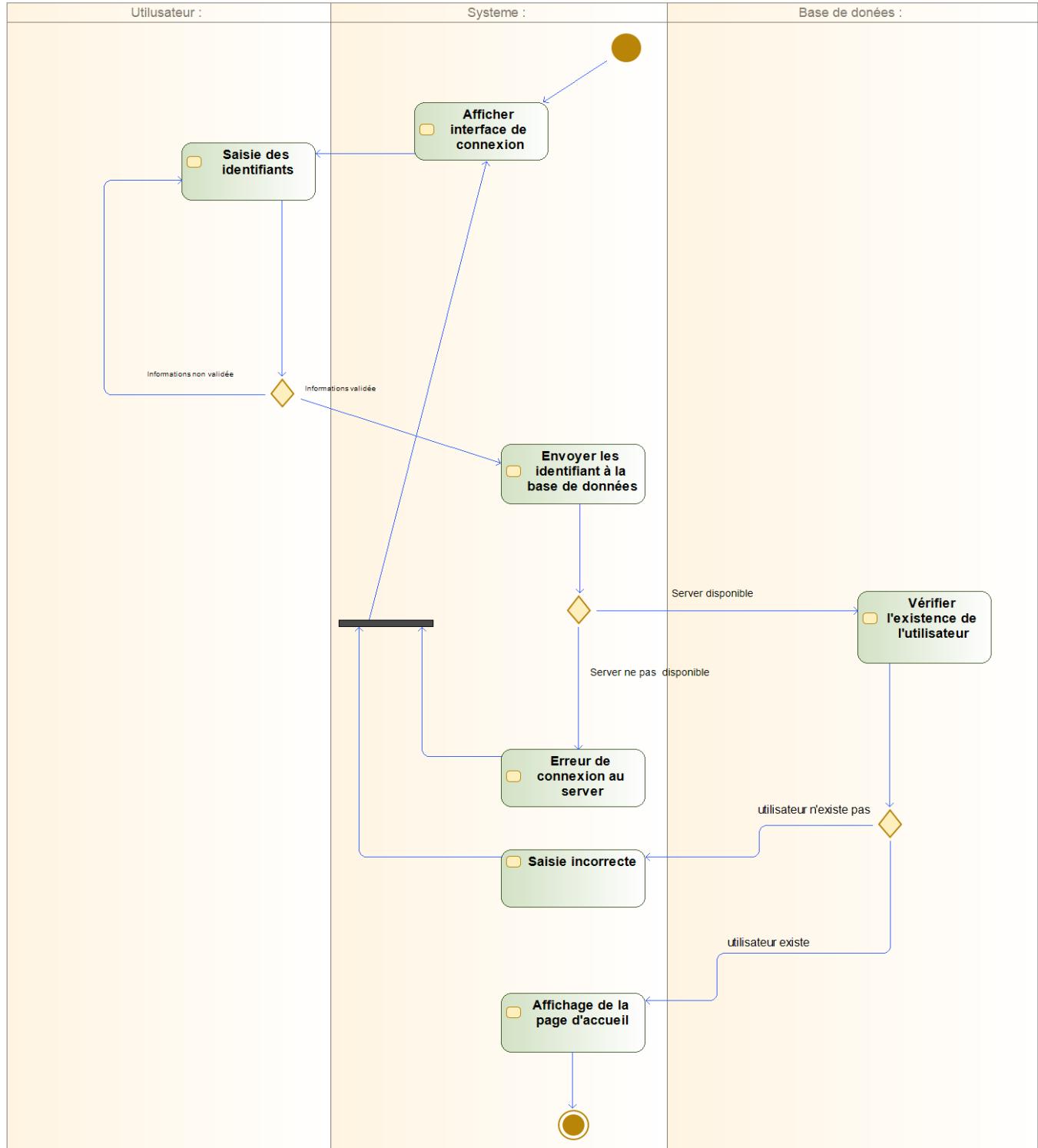


FIGURE 2.4 – Diagramme d'activité : Authentification

## 2.2 Modélisation de la base de données

### 2.2.1 Diagramme de classe

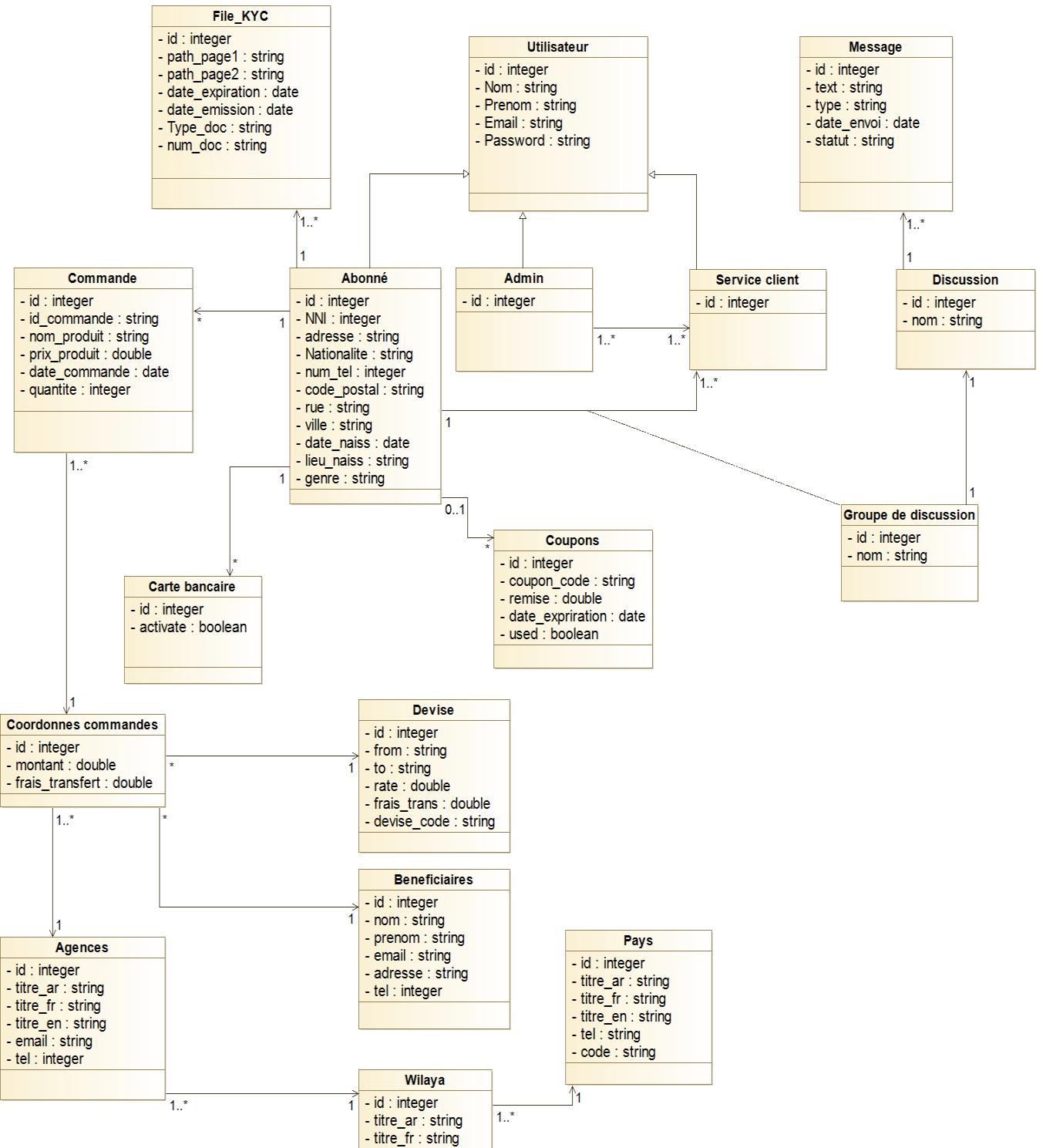


FIGURE 2.5 – Diagramme d’activité : Transfert d’argent

# Chapitre 3

## L'implémentation

### 3.1 Environnement de travail

Durant la réalisation de ce projet, nous avons essayé d'utiliser différents outils de développement, d'une part afin de rendre la tâche de la réalisation plus facile, d'autre part pour que notre système soit robuste et répond parfaitement à nos besoins , et que nos interfaces soient claires et faciles à utiliser.

#### 3.1.1 Environnement logiciel

##### 3.1.1.1 Editeur de texte : VS Code



FIGURE 3.1 – VS Code

VS Code (Visual Studio Code) est un éditeur de code source réalisé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la coloration syntaxique, la saisie semi-automatique intelligente du code, les extraits de code, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Pour en savoir plus, veillez visiter le lien : <https://en.wikipedia.org/wiki/VisualStudioCode>

### 3.1.1.2 IDE : Android Studio



FIGURE 3.2 – Android Studio

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.

Pour en savoir plus, veillez visiter le lien : <https://en.wikipedia.org/wiki/AndroidStudio>.

### 3.1.1.3 Git



FIGURE 3.3 – Git

Git est un système de contrôle de version distribué gratuit et Open Source conçu pour gérer des petits projets aux très grands projets avec rapidité et efficacité. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2 [GPLv2](#). Depuis 2016, il s'agit du logiciel de gestion de versions le plus populaire. Git possède deux structures de données : une base d'objets et un cache de répertoires. Il existe quatre types d'objets :

- \* l'objet **blob** (pour binary large object désignant un ensemble de données brutes) qui représente le contenu d'un fichier ;

- \* l'objet **tree** (mot anglais signifiant arbre) qui décrit une arborescence de fichiers. Il est constitué d'une liste d'objets de type blobs et des informations qui leur sont associées telles que le nom du fichier et les permissions. Il peut contenir récursivement d'autres objets trees pour représenter les sous-répertoires ;
- \* l'objet **commit** (résultat de l'opération du même nom signifiant « valider une transaction ») qui correspond à une arborescence de fichiers (tree) enrichie de métadonnées comme un message de description, le nom de l'auteur, etc. Il pointe également vers un ou plusieurs objets commit parents pour former un graphe d'historiques ;
- \* l'objet **tag** qui est une manière de nommer arbitrairement un commit spécifique pour l'identifier plus facilement. Il est en général utilisé pour marquer certains commits.

veillez visiter le lien : <https://git-scm.com/s>

### 3.1.1.4 Le SGBD Firebase



FIGURE 3.4 – Firebase

Firebase est une plateforme d'hébergement pour n'importe quel type d'application (mobiles, Web, etc.) qui fournit aux développeurs une pléthore d'outils et de services pour les aider à développer des applications de haute qualité, à élargir leur base d'utilisateurs et à générer davantage de profits. Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github) et des notifications, ou encore des services tels que par exemple un serveur de communication en temps réel [1]. Lancé en 2011 sous le nom d'Envolve, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet (maison-mère de groupe qui regroupe une majorité des activités du groupe comme YouTube, Android, la recherche et les publicités). L'objectif premier de Firebase est de vous libérer de la complexité de création et de

la maintenance d'une architecture serveur, tout en vous garantissant une scalabilité (capacité de l'application à s'adapter à un changement d'ordre de grandeur de la demande, en particulier sa capacité à maintenir ses fonctionnalités et ses performances en cas de forte demande) à toute épreuve (plusieurs milliards d'utilisateurs) et une simplicité dans l'utilisation. Pour cela, Firebase a été décomposée en plusieurs produits extrêmement riches et adaptés au monde du mobile, dont la liste suivante :

#### **3.1.1.4.1 Firebase Realtime Database :**

Firebase Realtime Database est une base de données NoSQL hébergée dans le cloud qui vous permet de stocker et de synchroniser des données entre vos utilisateurs en temps réel [1], [2]. Realtime Database est vraiment juste un gros objet JSON que les développeurs peuvent gérer en temps réel. La synchronisation en temps réel permet à vos utilisateurs d'accéder facilement à leurs données depuis n'importe quel appareil, que ce soit sur le Web ou sur un appareil mobile. La base de données en temps réel permet également à vos utilisateurs de collaborer les uns avec les autres. Un autre gros avantage de Realtime Database est qu'elle est livrée avec des SDK mobiles et Web, vous permettant de créer vos applications sans avoir besoin de serveurs. Lorsque vos utilisateurs sont hors ligne, les SDK de base de données en temps réel utilisent le cache local sur l'appareil pour servir et stocker les modifications. Lorsque l'appareil est en ligne, les données locales sont automatiquement synchronisées.

#### **3.1.1.4.2 Firebase Authentication :**

Firebase Authentication fournit des services backend, des SDK faciles à utiliser et des bibliothèques d'interfaces utilisateur prêtes à l'emploi pour authentifier les utilisateurs de votre application [1], [2]. Vous pouvez authentifier les utilisateurs de votre application à l'aide des méthodes suivantes :

- \* Email et mot de passe
- \* Numéro de téléphone
- \* Compte Google
- \* Compte Facebook
- \* Compte Twitter
- \* Etc.

L'utilisation de Firebase Authentication facilite la création de systèmes d'authentification sécurisés, tout en améliorant l'expérience de connexion et d'intégration pour les utilisateurs finaux.

#### **3.1.1.4.3 Firebase Cloud Messaging (FCM) :**

Firebase Cloud Messaging (FCM) fournit une connexion fiable et à faible consommation de batterie entre votre serveur et vos périphériques, vous permettant d'envoyer et de recevoir gratuitement des messages et des notifications sur iOS, Android et sur le Web [1], [2]. Vous pouvez envoyer des messages de notification (limite de 2 Ko) et des messages de données (limite de 4 Ko). En utilisant FCM, vous pouvez facilement cibler les messages en utilisant des segments prédéfinis ou créer les vôtres, en utilisant les données démographiques et comportementales. Vous pouvez envoyer des messages à un groupe d'appareils abonnés à des rubriques spécifiques, ou vous pouvez obtenir des informations aussi détaillées qu'un seul appareil. FCM peut envoyer des messages instantanément, ou à un moment ultérieur dans le fuseau horaire local de l'utilisateur. Vous pouvez envoyer des données d'application personnalisées, telles que la définition des priorités, des sons et des dates d'expiration, ainsi que le suivi des événements de conversion.

#### **3.1.1.4.4 Cloud Firestore :**

Cloud Firestore est une base de données de documents NoSQL qui vous permet de facilement stocker, synchroniser et interroger des données pour vos applications mobiles et Web à l'échelle mondiale [1], [2]. Bien que cela puisse ressembler à quelque chose de similaire à la base de données en temps réel, Firestore apporte beaucoup de nouvelles choses à la plateforme qui est en fait quelque chose de complètement différent de Realtime Database. Là où Realtime Database stocke des données sous la forme d'un arbre JSON géant, Cloud Firestore adopte une approche beaucoup plus structurée. Firestore conserve ses données dans des objets appelés documents. Ces documents sont constitués de paires clé-valeur et peuvent contenir n'importe quel type de données, depuis les chaînes jusqu'aux données binaires en passant par des objets qui ressemblent à des arbres JSON (Firestore l'appelle des maps). Les documents, à leur tour, sont regroupés en collections. La base de données Firestore peut se composer de plusieurs collections qui peuvent contenir des documents pointant vers des sous-collections. Ces sous-collections peuvent à nouveau contenir des documents qui pointent vers d'autres sous-collections, et ainsi de suite. Vous

pouvez créer des hiérarchies pour stocker les données associées et récupérer facilement les données dont vous avez besoin à l'aide de requêtes. Toutes les requêtes peuvent évoluer en fonction de la taille de votre jeu de résultats. Votre application est donc prête à évoluer depuis le premier jour.

### 3.1.1.5 SGBD MariaDB



FIGURE 3.5 – SGBD MariaDB

Un gestionnaire de base de données libre. ce projet est assurée par la fondation Maria DB, et sa maintenance par la société Monty Program AB, créateur du projet, il confère au logiciel l'assurance de rester libre. MariaDB a plusieurs et différentes versions. Elles s'articulent sur le code source de MySQL de la version 5.1 aux versions plus récentes (comme la 5.6 fin 2012). Un serveur qui stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Du coup il améliore la rapidité et la souplesse de l'ensemble.

### 3.1.1.6 Logiciel de modélisation : Modelio



FIGURE 3.6 – Modelio

Modelio est un logiciel open source et multiplateforme permettant, entre autres, la modélisation UML et Business Process Model and Notation (BPMN). Pour en savoir plus, veuillez visiter le lien <https://www.modelio.org/about-modelio/features.html>. Sans doute, les logiciels de modélisation UML sont nombreux, à savoir, Visual Paradigm, Eclipse Papyrus, StarUML, PowerDesigner, Umbrello, etc. Vu que les diagrammes UML que nous voulons réaliser sont disponibles dans tous ces logiciels, il n'y avait pas en effet un choix à argumenter car tous les choix étaient satisfaisants. Mais, de façon subjective, nous pouvons préciser que l'avantage de Modelio dans notre contexte est le fait que je m'y suis déjà habitués. Les diagrammes que nous avons réalisés avec Modelio sont ceux mentionnés ci-après.

### 3.1.1.7 Motifs d'architecture logicielle

#### 3.1.1.7.1 MVC

Le patron de conception MVC est l'une des bases du Framework laravel, c'est-à-dire modèle-vue- contrôleur. L'interaction avec la base de données est assurée par le Model, les regroupe, traite et gère les données. Pour faire afficher ce que le modèle renvoie on fait appelle à la vue . elle s'occupe d'autre part de la réception de toute interaction de l'utilisateur. Ce sont ces actions-là que le contrôleur gère. Ainsi que l'échange entre le modèle et la vue. Il intercepte toutes les activités de utilisateur et, en fonction de ces activités, il actionne les changements à effectuer sur l'application. Les composants sont séparés en ces trois catégories précédentes permet une clarté de architecture des dossiers et simplifie grandement la tâche aux développeurs.

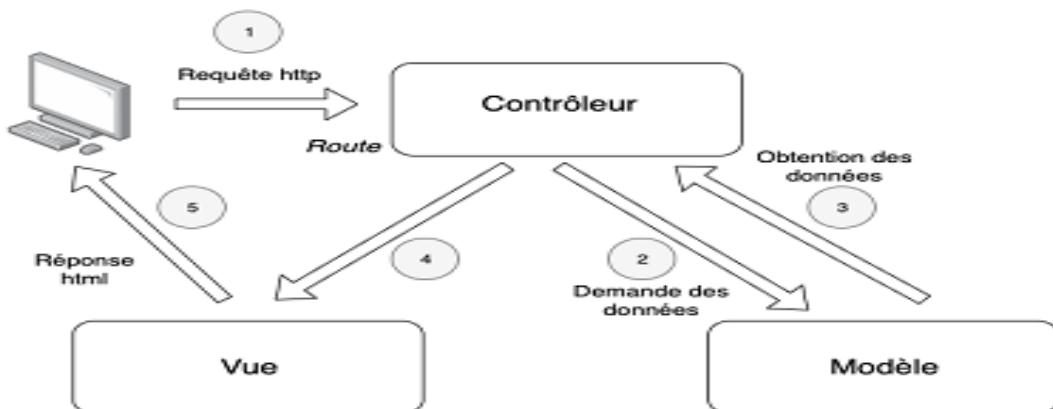


FIGURE 3.7 – MVC

**Les fichiers sont organisés comme suit :**

- **Route (Dispatcher)** : il contient les définitions des chemins d'entrées pour l'utilisateur, autrement dit les URI possibles et les dirige sur la classe définie dans le contrôleur qui doit traiter l'information.
- **Modèle** : pour chaque table de notre base de données que l'on veut utiliser pour notre application, il faut créer un modèle pour chacun. Ainsi nous avons ici un modèle de notre application. Il permet de décrire la méthode d'accès aux données de la base, tous cela à travers un objet défini par ORM Eloquent(Object-Relational Mapping).
- **Contrôleur** : il permet de récupérer les informations du modèle et de l'envoyer vers la vue pour la mise en forme.
- **Vue** : la vue réceptionne la réponse qui est envoyée par le contrôleur .

### 3.1.1.8 Serveur Web : Amazon Web Services (AWS)



FIGURE 3.8 – Amazon Web Services (AWS)

Amazon Web Services (AWS) est la plateforme cloud la plus complète et la plus largement adoptée au monde. Elle propose plus de 200 services complets issus de centres de données du monde entier. Des millions de clients (dont certaines des startups les plus dynamiques au monde, de très grandes entreprises et des agences fédérales de premier plan) utilisent AWS pour réduire leurs coûts, gagner en agilité et innover plus rapidement.

### 3.1.2 Conception

#### 3.1.2.1 Langage de modélisation : UML



FIGURE 3.9 – UML

On a utilisé UML comme langage de modélisation. Langage de modélisation unifié UML (Unified modeling Language) un consiste à modéliser une application logicielle d'une façon standard dans le cadre de conception orientée objet. UML consiste à couvrir le cycle de vie d'un logiciel depuis la spécification des besoins jusqu'au codage en offrant plusieurs moyens de description et de modélisation des acteurs.

## 3.2 Choix des outils de travail

### 3.2.1 Langages utilisés

#### 3.2.1.1 Dart



FIGURE 3.10 – Dart

Dart est un langage de programmation open source à usage général. Il est initialement développé par Google. Dart est un langage orienté objet avec une syntaxe de C-style. Il prend en charge les concepts de programmation tels que les interfaces, les classes, contrairement aux autres langages de programmation, Dart ne prend pas en charge les tableaux. Les collections Dart peuvent être utilisées pour répliquer des structures de données telles que des tableaux, des génériques et un typage facultatif.

Pour en savoir plus, veillez visiter le lien : <https://www.tutorialspoint.com/flutter/dart>

### 3.2.1.2 PHP



FIGURE 3.11 – PHP

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur web. PHP peut également générer d'autres formats comme le WML, le SVG et le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP tels que IIS ou nginx. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

Pour en savoir plus, veillez visiter le lien : <https://fr.wikipedia.org/wiki/PHP>

## 3.2.2 Frameworks utilisés

### 3.2.2.1 Flutter



FIGURE 3.12 – Flutter

Flutter est un framework de développement d'applications mobiles open source de Google. La principale raison de sa popularité est qu'il prend en charge la création d'applications multiplateformes. Flutter est également utilisé pour créer des apps interactives

qui s'exécutent sur des pages web ou sur le bureau.

- Les caractéristiques de Flutter :

1. Base de code unique pour Android et iOS
2. Fonction de rechargement à chaud (hot reload)
3. Open-source et par Google
4. Programmation Dart

- Architecture d'une application Flutter :

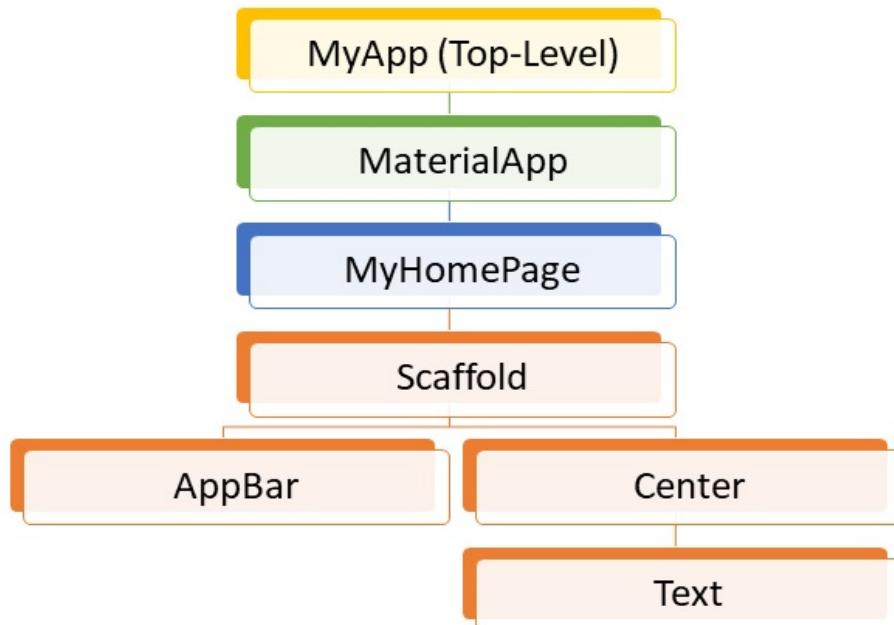


FIGURE 3.13 – Architecture d'une application Flutter.

### 3.2.2.2 Laravel

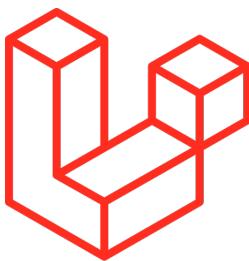


FIGURE 3.14 – Laravel

Laravel est un framework web open-source écrit en [PHP](#) respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous [licence MIT](#), avec ses sources hébergées sur [GitHub](#).

#### - Les caractéristiques de Laravel :

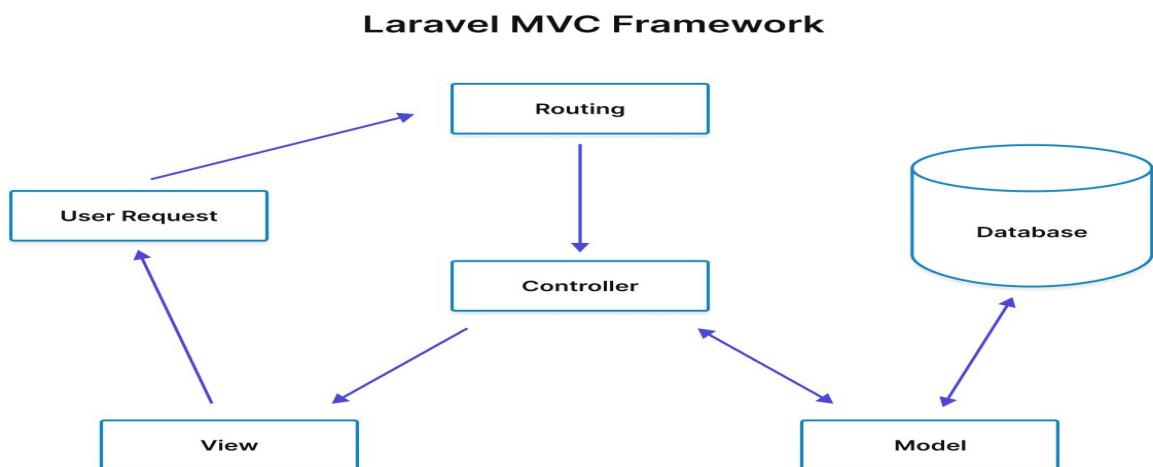


FIGURE 3.15 – Architecture de Laravel

## 3.3 Implémentation

### 3.3.1 Etape de réalisation

Pour réaliser notre système, il y a des étapes à élaborer dans l'ordre suivant :

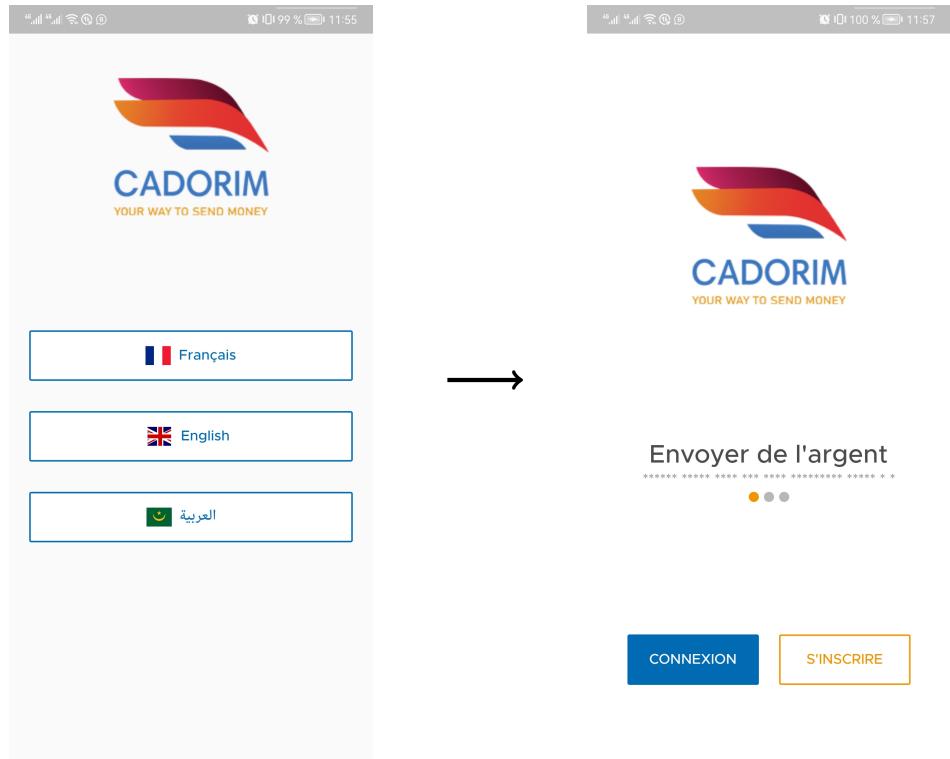
- \* **Conception de la base de données** : La conception d'une base de données est la première étape. Le choix des algorithmes et de l'approche de travail exige l'utilisation d'une base de données spécifique (en fonction du système à développer).
- \* **Extraction des données** : On va utiliser notre SGBD Firebase ainsi que notre base de données en interaction avec l'application pour extraire des informations. Firebase est conçu avec des fonctionnalités et des requêtes de sélection de données assez spécifique et facile à utiliser.
- \* **Conception et développement du front-end et du back-end** : Cette étape consiste à détailler la conception coté client et coté serveur. Il s'agit de mettre en place un design ergonomique, simple et attractif répondant aux exigences du système. Le choix d'un serveur d'application adéquat aux fonctionnalités et aux données est une étape fondamentale pour le bon fonctionnement de l'application.
- \* **Développement de back-end** : On commence par le développement de l'application coté serveur, dans notre cas avec **Laravel**. C'est la partie du code exécuté sur le serveur afin de vérifier le comportement des fonctionnalités de base du système.
- \* **Développement front-end** : On développe la partie client en interaction avec le serveur. C'est la conception de l'interface graphique utilisateur. En effet, il s'agit de la partie visible de l'application, destinée à être manipulée par un tiers.

### 3.3.2 Interfaces Homme/Machine

Dans ce qui suit, nous présentons quelques écrans de notre produit final des applications CADORIM et Cadorim service.

#### 3.3.2.1 Application CADORIM

- \* **Première interface** : L'utilisateur du Cadorim avant d'être invité à consulter les services de l'application, doit choisir leur langue avant qu'il passe à l'interface suivante.



(a) Choix de langue

(b) Interface suivante

FIGURE 3.16 – Première interface

- \* **L'interface d'authentification :** La figure suivante représente l'interface d'authentification de notre application. Elle permet aux utilisateurs de s'identifier en introduisant leurs identifiants afin d'accéder aux fonctionnalités de l'application.

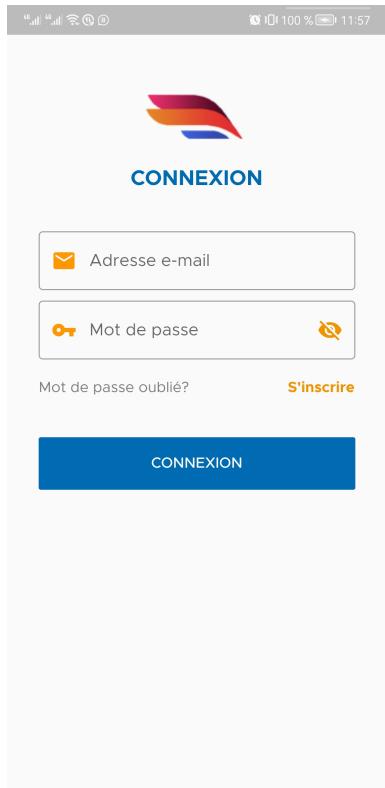
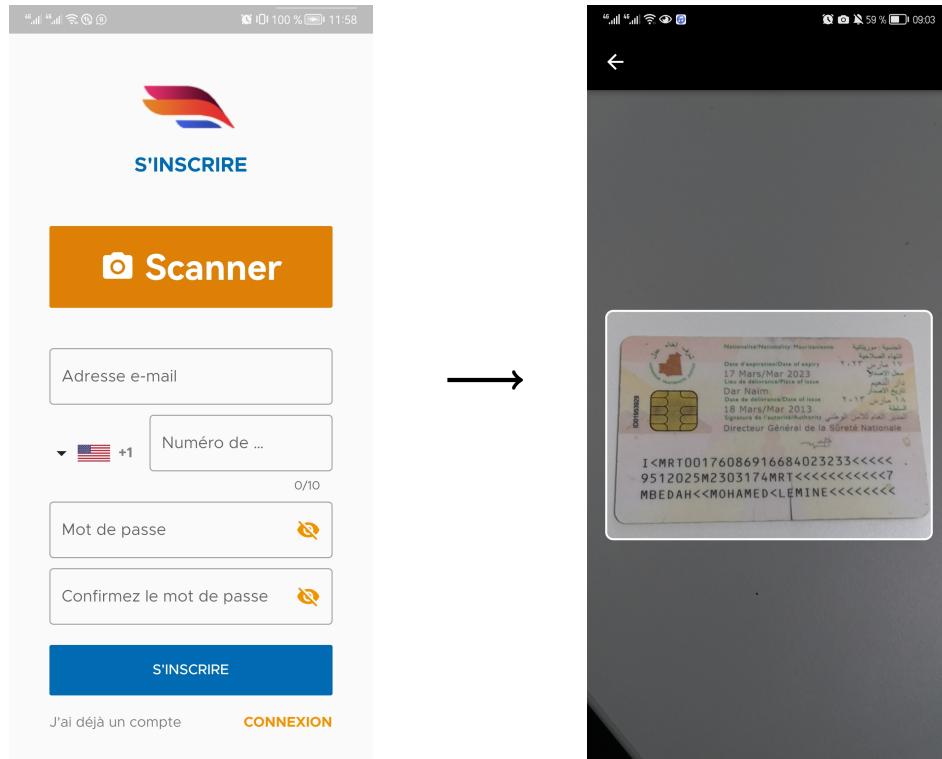


FIGURE 3.17 – Interface d’authentification

La demande d’identification du client est traitée pour vérifier ses paramètres dans la base de données. L’absence de l’utilisateur dans la base de données ou une erreur de saisie des informations entraîne une alerte d’erreur d’authentification.

- \* **L’interface d’inscription :** Avant de pouvoir s’authentifier, l’utilisateur doit impérativement s’enregistrer au préalable dans la base de données. La figure suivante représente l’interface de création de compte pour un client. Toutes les informations personnelles sont extraites partir de scan de code MRZ.



(a) Interface d'inscription

(b) Scanner le code MRZ

**S'INSCRIRE**

Photo du face avant

Photo du face arrière

Adresse e-mail  
med.lemine.lemine@cado.com

Numéro de téléphone  
+222 27666242

Mot de passe  
.....

Confirmez le mot de passe  
.....

**S'INSCRIRE**

J'ai déjà un compte **CONNEXION**

(c) Saisir le rest des données

FIGURE 3.18 – Interface d'inscription

\* **Processus de la transaction** : La figure suivante montre les processus de la transaction depuis la sélection de montant jusqu'à la validation (une notification de succès va Apparence si le traitement se fait avec succès)

**ENVOYER DE L'ARGENT**

FR 5

MR 193

Min 5€ / Max 2500€

Frais de transfert 0.99 EUR

Total à payer : 5.99 EUR

ENVOYEZ

(a) Interfaces d'accueil

**PASSER VOTRE COMMANDE**

Selectionner un bénéficiaire

Ou bien saisir un bénéficiaire

Prenom bénéficiaire

Nom bénéficiaire

Numéro du bénéficiaire

Wilaya

Agence

Choisir votre carte bancaire

Transfert de France Vers Mauritanie  
Taux de change 1 EUR = 38.5 MRU  
Total à payer : 5.99 EUR  
Montant total du bénéficiaire : 193 MRU

(b) Interface de transfert

Ou bien saisir un bénéficiaire

Prenom bénéficiaire: med

Nom bénéficiaire: ali

Wilaya: Adrar - أدرار

Agence: Assaba - العصابة

Brakna - البراكنة

Dakhlet Nouadhibou - داخلة - نواذيبو

Gorgol - كوركول

Guidimakha - كيدى ماغا

Hodh El Chargui - الحوض الشرقي

Montant total du bénéficiaire : 193 MRU

Ajouter le code promo APPLIQUER

(c) Choix de wilaya

Ou bien saisir un bénéficiaire

Prenom bénéficiaire: med

Nom bénéficiaire: ali

Numéro du bénéficiaire: 47851588

Wilaya: Brakna - البراكنة

Agence: Aleg - Bouhdida - ألاك - بوجديد

master 9900

Transfert de France Vers Mauritanie  
Taux de change 1 EUR = 38.5 MRU  
Total à payer : 5.99 EUR  
Montant total du bénéficiaire : 193 MRU

Ajouter le code promo APPLIQUER

(d) Choix d'agence

Ou bien saisir un bénéficiaire

Prenom bénéficiaire: ali

Numéro du bénéficiaire: 47851588

Wilaya: Brakna - البراكنة

Agence: Aleg - Bouhdida - ألاك - بوجديد

master 9900

Transfert de France Vers Mauritanie  
Taux de change 1 EUR = 38.5 MRU  
Total à payer : 5.99 EUR  
Montant total du bénéficiaire : 193 MRU

Ajouter le code promo APPLIQUER

J'ai lu et j'accepte les conditions générales

VALIDER ANNULER

(e) Choix de catre bancaire

Ou bien saisir un bénéficiaire

votre transaction est en cours de traitement

47851588

Wilaya: Brakna - البراكنة

Agence: Aleg - Bouhdida - ألاك - بوجديد

master 9900

Transfert de France Vers Mauritanie  
Taux de change 1 EUR = 38.5 MRU  
Total à payer : 5.99 EUR  
Montant total du bénéficiaire : 193 MRU

Ajouter le code promo APPLIQUER

J'ai lu et j'accepte les conditions générales

VALIDER ANNULER

(f) État de transfert

FIGURE 3.19 – Processus de transactions



(a) Transfere avec succès

Date	Bénéficiaire	Montant	Détail
19-09-2022 – 15:37	ali	5	<a href="#">i</a>

(b) Historique transactions

ID_commande	668363810
Nom exp	Mohamed lemene
Num exp	27666242
Email exp	med.lemine.lemine@cado.com
Nom benef	med ali
Num benef	47851588
Adresse benef	Brakna - البراكنة-Aleg - Bouhdida
Montant	5 EUR
Frais de transfert	0.99
Date	19-09-2022 – 15:37
Statut	Completed

(c) Détails de la transaction

FIGURE 3.20 – Transferts

- \* **L'interface de transfert :** La figure 3.20 montre l'historique des transactions de l'utilisateur vers les bénéficiaires avec détails pour chaque transaction.
- \* **L'interface de discussion :** La figure suivante montre que le client peut envoyer un message (photo ou un texte) au service client et bien sur voir l'historique des messages.

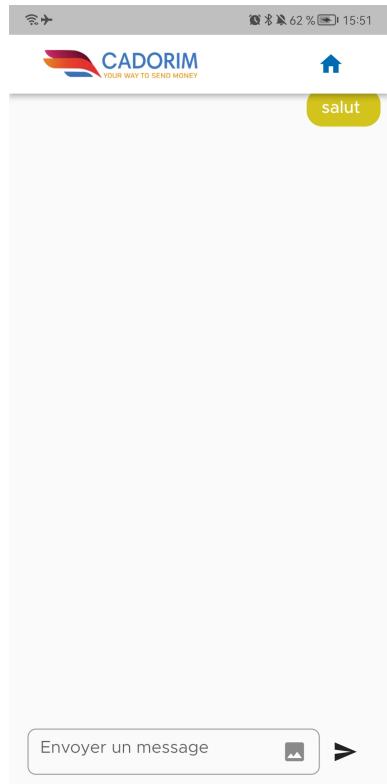
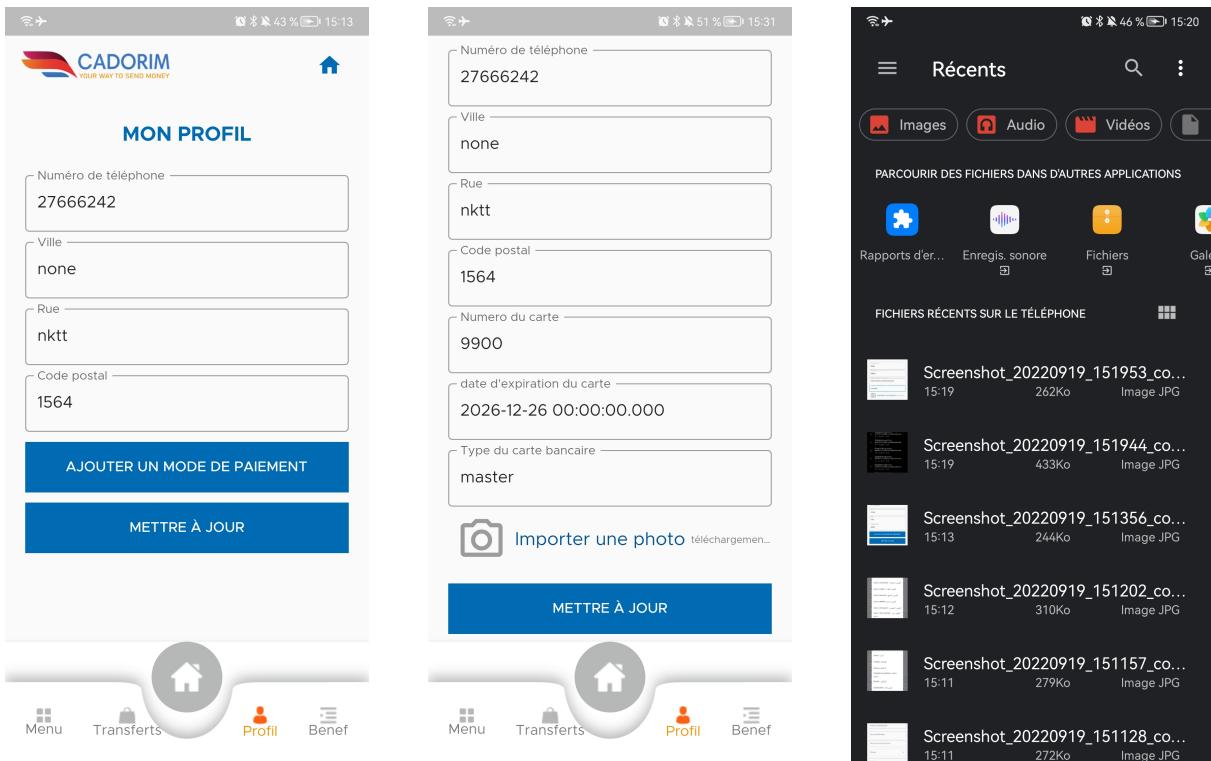


FIGURE 3.21 – Interface de discussion

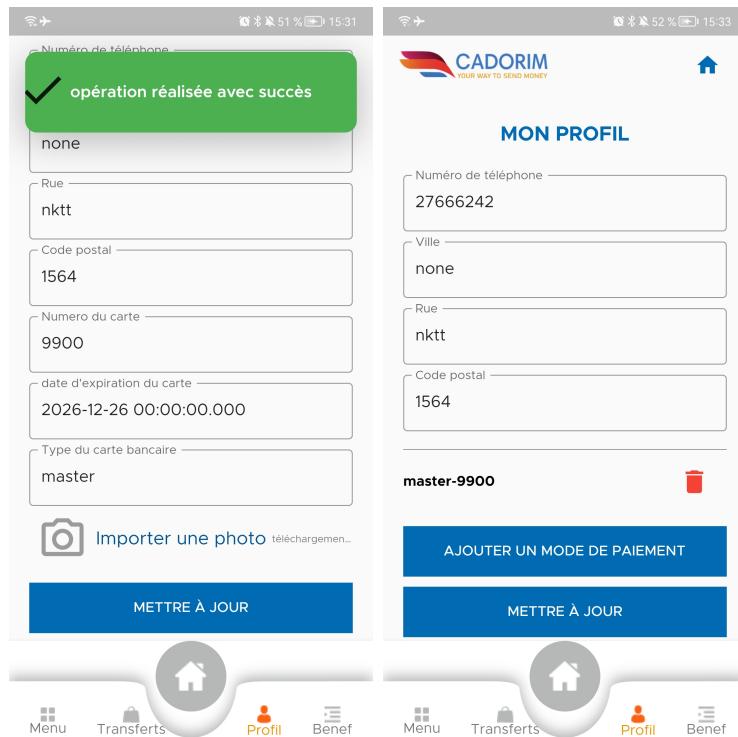
- \* **Profil de l'utilisateur :** L'utilisateur peut mettre à jour leur profil ou ajouter un mode paiement comme il montre la figure suivante



(a) Mon profil

(b) Ajouter une carte

(c) Importer une photo



(d) État

(e) Interface suivante

FIGURE 3.22 – Profil de l'utilisateur

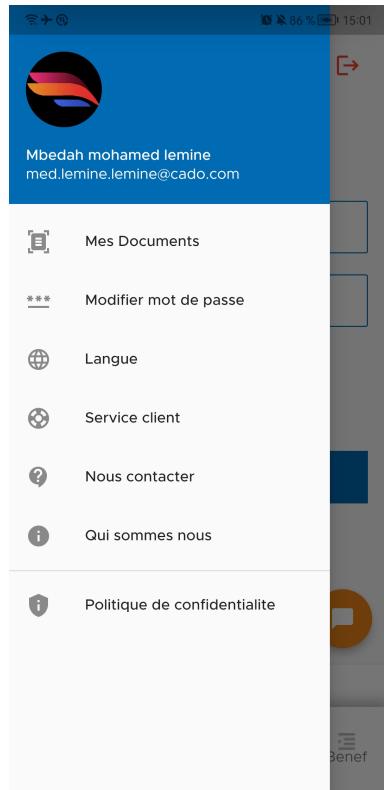


FIGURE 3.23 – Interface de menu

- \* **L'interface de menu :** À partir de cette fenêtre le client a l'accès aux autres fonctionnalités de l'application (Modifier mot de passe, changer la langue ....)

### 3.3.2.2 Application ServiceClient

- \* **L'interface d'authentification :** La figure suivante représente l'interface d'authentification de notre application. Elle permet aux utilisateurs de s'identifier en introduisant leurs identifiants afin d'accéder aux fonctionnalités de l'application.

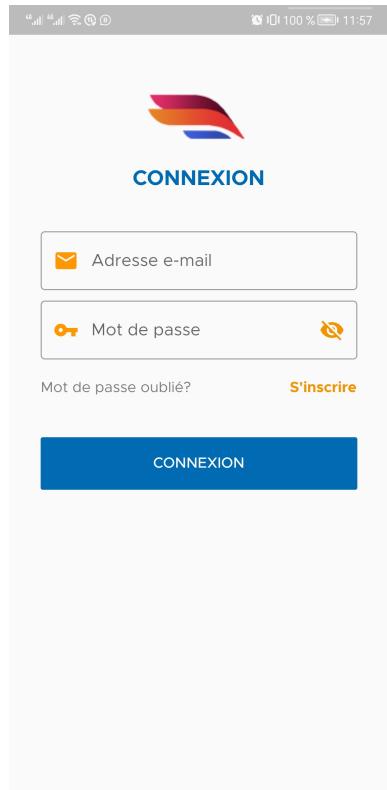


FIGURE 3.24 – Interface d’authentification

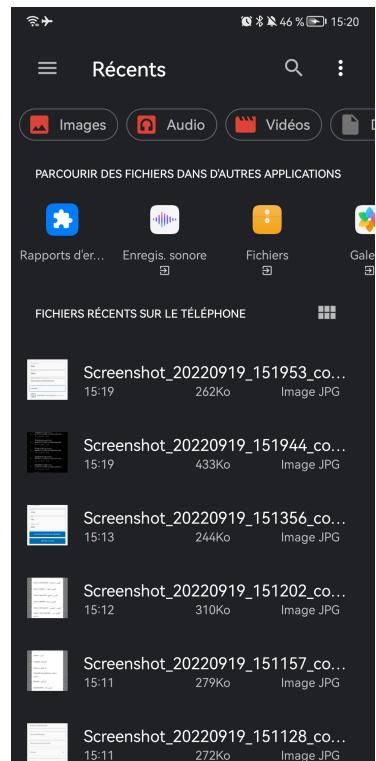
- \* **L’interface du compte administrateur :** La figure 3.25 représente la page principale de l’application pour l’administrateur. C’est à partir de cette fenêtre qu’est accessible la majorité des fonctionnalités de base de l’application(Gérer les comptes de service client).



(a) Interfaces d'accueil



(b) Gérer un compte

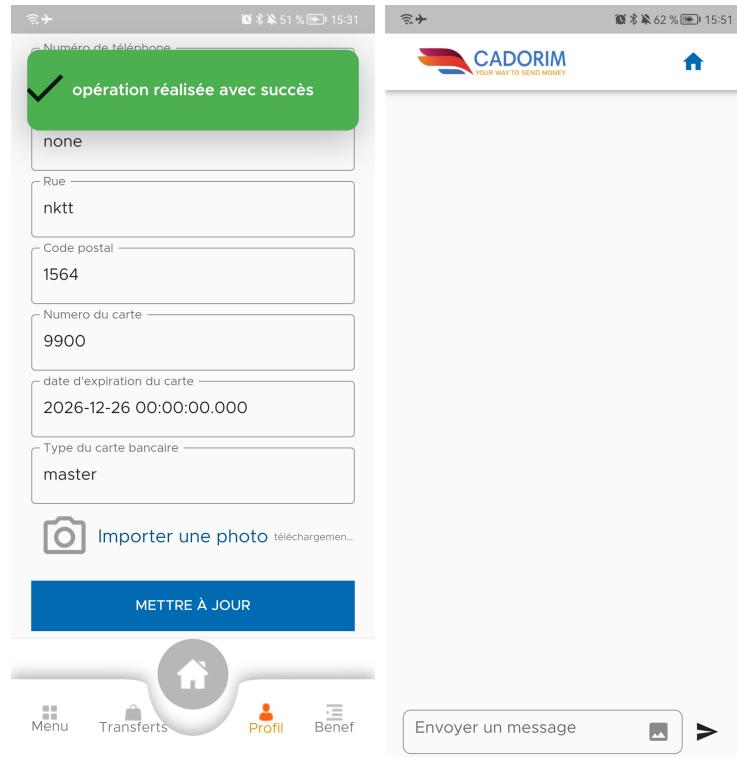


(c) Mettre à jour un compte



(d) Ajouter un compte

FIGURE 3.25 – Interface du compte administrateur



(a) Interfaces d'accueil

(b) Interface de discussion

FIGURE 3.26 – Interface du compte administrateur

- \* **L'interface du compte service client :** La figure 3.26 représente la page principale de l'application pour le compte service client. C'est à partir de cette fenêtre l'utilisateur peut répondre aux clients.

# Conclusion

Dans ce chapitre, avons décrit brièvement le processus de développement de notre application de transfert d'argent en spécifiant l'environnement de développement, le choix des outils ainsi que la démarche suivie pour la réalisation. En effet, nous avons achevé l'implémentation tout en respectant la conception, mais les tests des différents cas d'utilisations sont toujours en cours envie d'une amélioration de l'application avant sa publication. En d'autres termes, nous avons la version beta (test et amélioration avant sa publication) de l'application installée dans notre environnement de développement. Aussi, nous avons prévu la période pendant laquelle la solution finale sera déployée les plate-formes de téléchargement.

# Bibliographie

- [ 1 ] N. Symth, *Firebase Essentials*. Cary : Payload Media, 2017.
- [ 2 ] J. Crowther, *Firebase*. London : Constable, 2015.
- [ 3 ] S. Madise, *The regulation of mobile money*. New York, NY : Springer Berlin Heidelberg, 2019.
- [ 4 ] E. M. Ndadoum et B. Kordjé, *Mobile money en Afrique - Son rôle pour l'inclusion financière au Tchad*. L'Harmattan, 2020.
- [ 5 ] O. Fédior, « *Mobile money/Mobile banking : La guerre des transferts - OSIRIS : Observatoire sur les Systèmes d'Information, les Réseaux et les Inforoutes au Sénégal* », mars 08, 2019. <http://www.osiris.sn/Mobile-money-Mobile-banking-La.html> (consulté le juill. 27, 2020).
- [ 6 ] B. A. Lassaad, « *Cameroun : pénurie inédite de pièces de monnaie* », janv. 10, 2020. <https://www.aa.com.tr/fr/afrique/cameroun-pénurie-inédite-de-pièces-de-monnaie/1698715> (consulté le juin 13, 2020).
- [ 7 ] Glabb, Ryan et al. (2007), « *Multi-mode operator for SHA-2 hash functions* », in : *journal of systems architecture* 53.2-3, p. 127–138.
- [ 8 ] Dworkin, Morris J (2015), *SHA-3 standard : Permutation-based hash and extendable-output functions*, rapp. tech.
- [ 9 ] Dolmatov, Vasily et Alexey Degtyarev (2013), « *GOST R 34.11-2012 : hash function* », in : *Independent Submission*, Ed. Request for Comments : Updates 5831, p. 2070–1721.