



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)

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

Les mots ne sauront traduire ce qui est dans le coeur, mais je vais rassembler mon vocabulaire et dire : je dédie cet humble travail à mes chers parents pour leur soutien tout au long de mon parcours universitaire, qui ont été les encouragements dans les moments de détresse, l'exhortation en période de prospérité, le soutien lors de la chute et le guide lors de la montée, qui n'a pas hésité un instant à m'aider jusqu'à ce que j'ai arrivé là où je suis maintenant. Tous les remerciements et gratitude ne remplissent pas leur droit.

REMERCIEMENTS

*Nos premiers remerciements sont adressés à nos professeurs à L’Institut Supérieur de Comptabilité et d’Administration des Entreprises pour leur forte implication à notre formation de master et on remercie particulièrement **Dr. Emani Mohamed Sidi**, coordinateur du master informatique. Ensuite, bien sûr, on tient à remercier notre encadrant **Dr. Cheikh Dhib**, l’ancien coordinateur de MPIAG, pour son encadrement, ses recommandations précieuses et sa disponibilité, ainsi que **Mr Mohamed Mboirick** qui nous a encadrer au sein de l’entreprise CADORIM il était une grande référence pour nous et on a beaucoup appris avec lui et son aide pour réaliser ce stage. Enfin nos remerciements tout ceux qui ont contribué de près ou de loin à l’élaboration de ce travail.*

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 mobile. L'un 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 et 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 menagment 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

Introduction générale	10
1 Contexte général du projet	11
1.1 Présentation de l'organisme d'accueil	11
1.1.1 Missions	12
1.1.2 Organigramme	12
1.2 Cadre général du projet	13
1.2.1 Problématique	13
1.2.2 Objectifs du projet	14
1.2.3 Planification du projet	14
2 Analyse fonctionnelle et conceptuelle	17
2.1 Analyse fonctionnelle	17
2.1.1 Diagramme de cas d'utilisation	17
2.1.2 Diagramme d'activité	22
2.1.2.1 Création de compte	23
2.1.2.2 Transfert d'argent	24
2.1.2.3 Authentification	25
2.1.3 Diagramme de classe	26
3 L'implémentation	28
3.1 Environnement de travail	28
3.1.1 Editeur de texte : VS Code	28
3.1.2 IDE : Android Studio	28
3.1.3 Git	29
3.1.4 Le SGBD Firebase	29

3.1.4.1	Firebase Realtime Database :	30
3.1.4.2	Firebase Authentication :	30
3.1.4.3	Firebase Cloud Messaging (FCM)	31
3.1.4.4	Cloud Firestore	31
3.1.5	SGBD MariaDB	32
3.1.6	Logiciel de modélisation : Modelio	32
3.1.7	Motifs d'architecture logicielle : MVC	33
3.1.8	Serveur Web : Amazon Web Services (AWS)	34
3.2	Choix des outils de travail	34
3.2.1	Langages utilisés	34
3.2.1.1	Dart	34
3.2.1.2	PHP	34
3.2.1.3	Langage de modélisation : UML	35
3.2.2	Frameworks utilisés	35
3.2.2.1	Flutter	35
3.2.2.2	Laravel	38
3.3	Implémentation	39
3.3.1	Etape de réalisation	39
3.3.2	Interfaces Homme/Machine	39
3.3.2.1	Application CADORIM	39
3.3.2.2	Application ServiceClient	47
Conclusion		51
Bibliographie		52

Table des figures

1.1 CADOROM	11
1.2 Organigramme du CADORIM	12
2.1 Diagramme de cas d'utilisation.	17
2.2 Diagramme d'activité : Création de compte	23
2.3 Diagramme d'activité : Transfert d'argent	24
2.4 Diagramme d'activité : Authentification	25
2.5 Diagramme de classe	26
3.1 MVC	33
3.2 Architecture d'une application Flutter.	36
3.3 Architecture de Laravel	38
3.4 Première interface	40
3.5 Interface d'authentification	41
3.6 Interface d'inscription	42
3.7 Processus de transactions	43
3.8 Transferts	44
3.9 Interface de discussion	45
3.10 Profil de l'utilisateur	46
3.11 Interface de menu	47
3.12 Interface d'authentification	48
3.13 Interface du compte administrateur	49
3.14 Interface du compte service client	50

Liste des tableaux

1.1	Planification du projet	15
2.1	Description détaillée des cas d'utilisation : Créer un compte et S'authentifier	18
2.2	Description détaillée des cas d'utilisation : Effectuer transfert	19
2.3	Description détaillée des cas d'utilisation : Consulter l'historiques(des transactions ou des bénéficiaires)	20
2.4	Description détaillée des cas d'utilisation : Consulter boite messagerie . . .	21
2.5	Description détaillée des cas d'utilisation : Envoyer un message	22
2.6	Description de la base de données	27

Introduction générale

Étant donné que nous sommes dans un monde le smartphone deviant de plus en plus présent dans la vie de l'être humain, il est impératif que toutes les transactions financières soient disponibles via celui-ci. Sur la base de cette règle, fournir une application à travers laquelle les fonds peuvent être transférés de vers l'intérieur du pays , est une nécessité. Les applications existant , on constate qu'elles manquent de rapidité pour déterminer l'identité de l'expéditeur et la validité des comptes, en plus, la possibilité que ces pièce justificative soit invalide, ou qu'il y ait des fautes d'orthographe au cours de saisir. Le remplacement de l'extraction manuelle par l'extraction automatique des données réduit considérablement le risque d'erreurs humaines. Il en résulte donc une amélioration globale de la précision.

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 en place et la planification de projet.

1.1 Présentation de l'organisme d'accueil



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 à transférer de l'argent depuis n'importe quel pays dans le monde vers ses proches en Mauritanie. Notre objectif et de fournir 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. 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 Marché Capital , Nouakchott, Mauritanie, immatriculée

au registre du commerce.

1.1.1 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

1.1.2 Organigramme

La structure organisationnelle de CADORIM comprend :

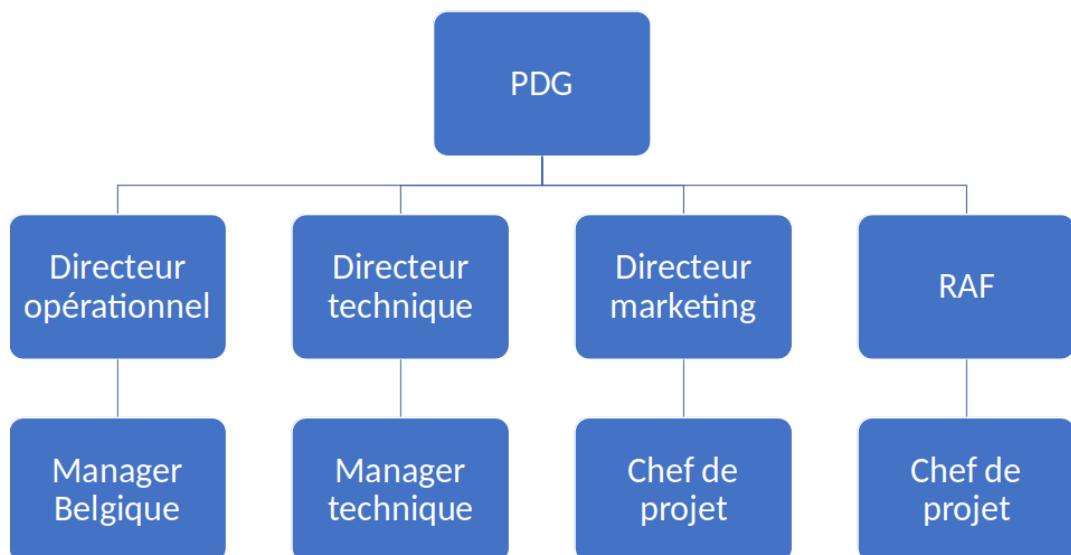


FIGURE 1.2 – Organigramme du CADORIM

1. **PDG** : Le président directeur général qui coiffe toutes les unités de la société et qui définit la vision stratégique à adopter en vue de pleinement remplir leurs missions.
2. **Directeur opérationnel** : Il joue un rôle d'intermédiaire et d'interface entre les différents services de CADORIM. Il assure un réel lien entre les équipes de Service et les clients.
3. **Directeur technique** : Il joue le rôle de l'encadrant des équipes techniques (recruter, animer, motiver et faire adhérer) et suivre la veille technologique.

4. **Directeur marketing** : Occupe un rôle stratégique et de management. Il s'occupe de la conception des produits, mais aussi de la présentation des offres qui ont pour but de satisfaire les consommateurs.
5. **RAF** : Le responsable administratif, comptable et financier est l'un des hommes-clés .le RAF doit assurer la gestion financière, administrative et même comptable.
6. **Manager** :Le rôle du manager et des cadres (leadership) est d'assurer un lien cohérent entre les équipes.
7. **Chef de projet** : Le chef de projet organise et conduit le projet. Il assume la responsabilité des différentes phases, depuis la traduction des besoins utilisateurs en spécifications fonctionnelles et techniques, jusqu'à la recette utilisateur, voire la mise en production.

1.2 Cadre général du projet

1.2.1 Problématique

Afin de résoudre le problème de fraude financière et réaliser plus de ventes, 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. ce system quand il est bien implémenté, permet non seulement de prévenir les fraudes mais il va également améliorer l'expérience client en rendant la vérification des informations plus simple et fluide, ce 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 :

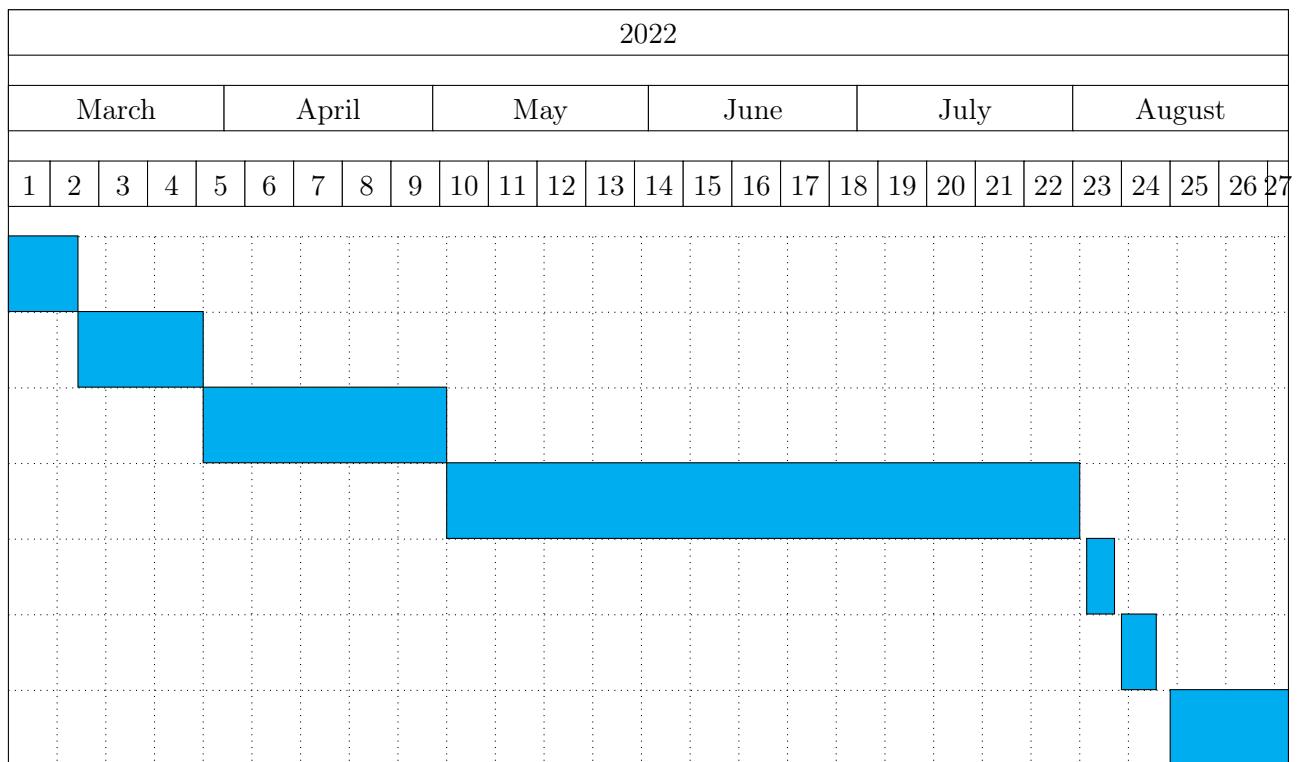
- * **Faciliter le parcours d'inscription :** Le client renseigne un certain nombre d'informations via un formulaire. Ensuite, le client scan un document d'identité, via la caméra de son smartphone . Les informations sont ensuite analysées automatiquement.
- * **Faciliter le transfert d'argent :** Avec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

Dans le cadre de la conduite du projet, la réalisation d'un planning à suivre tout au long du stage de fin d'études est indispensable. Ainsi, le stage a débuté le mardi 1 mars 2022. En conséquence, une réunion s'est tenue afin de définir le calendrier du projet.

N°	Phase	Date début	Durée	Date de fin
1	Découverte de produit et stratégie	2022/03/01	8 jours	2022/03/10
2	Planification du processus de conception	2022/03/11	12 jours	2022/03/28
3	Idéation	2022/03/29	25 jours	2022/05/02
4	Conception et développement	2022/05/03	65 jour	2022/08/01
5	Test de l'application	2022/08/03	3 jour	2022/08/06
6	Activités pré-lancement	2022/08/08	4 jour	2022/08/12
7	Rapport	2022/08/15	13 jour	2022/08/31

TABLE 1.1 – Planification du projet



1. L'objectif de la phase de découverte du produit est de connaître nos utilisateurs finaux afin de comprendre leurs besoins et être en mesure de leur proposer la solution adaptée, en priorisant les fonctionnalités essentielles. La définition d'une stratégie de produit est une partie essentielle du processus, car elle nous aidera à garder nos objectifs commerciaux à l'esprit à chaque étape du développement de notre produit final.
2. La deuxième phase consiste à rédiger une feuille de route produit qui nous aidera à

définir les éléments essentiels de la conception de la note produit et à les regrouper en différentes étapes de livraison. de cette manière, nous obtiendrons un plan précis des différents composants de notre produit et des délais dans lesquels chacun doit être complété.

3. La phase d'idéation consiste à explorer différentes idées créatives pour atteindre les objectifs du projet. C'est là qu'une ligne directrice doit être établie, qui sera suivie à toutes les étapes du processus de développement.
4. Le succès d'une application dépend de la réaction du public ciblé . Le design est une étape clé dans le processus de développement d'une application, car il définit non seulement le design de votre application, mais aussi et surtout son fonctionnement pour les utilisateurs finaux. Une application qui offre une excellente expérience utilisateur a de meilleures chances de succès.
5. La phase de test permet de s'assurer que notre produit fonctionne correctement et qu'il est fiable, utilisable et agréable. Les tests peuvent être divisés en deux groupes principaux : les tests fonctionnels et non fonctionnels.
 - Tests fonctionnels** : Pour optimiser les tests fonctionnels, il convient de mettre en place un processus d'intégration continue. L'intégration continue automatise les tests à chaque fois que le code source change.
 - Tests non fonctionnels** : Les tests non fonctionnels vérifient les performances ou l'utilisabilité d'un produit. L'idée de base est de vérifier si le produit offre un design d'interaction efficace pour ses utilisateurs. Les tests d'utilisabilité peuvent être effectués de manière formelle, avec un chercheur créant une série de tâches, recrutant des participants, menant des sessions de tests et les enregistrant, etc., ou de manière plus informelle sous la forme de tests de guérilla. .
6. Le développement d'une application est un processus itératif, qui ne s'arrête pas une fois l'application lancée. Après le développement, nous devons obtenir les commentaires de l'équipe technique et consacrer du temps et des ressources à l'amélioration de l'expérience utilisateur (UX). Cela vous aidera également à évaluer le succès à long terme de notre application.
7. Préparation d'un rapport pour la documentation de travail.

Chapitre 2

Analyse fonctionnelle et conceptuelle

2.1 Analyse fonctionnelle

2.1.1 Diagramme de cas d'utilisation

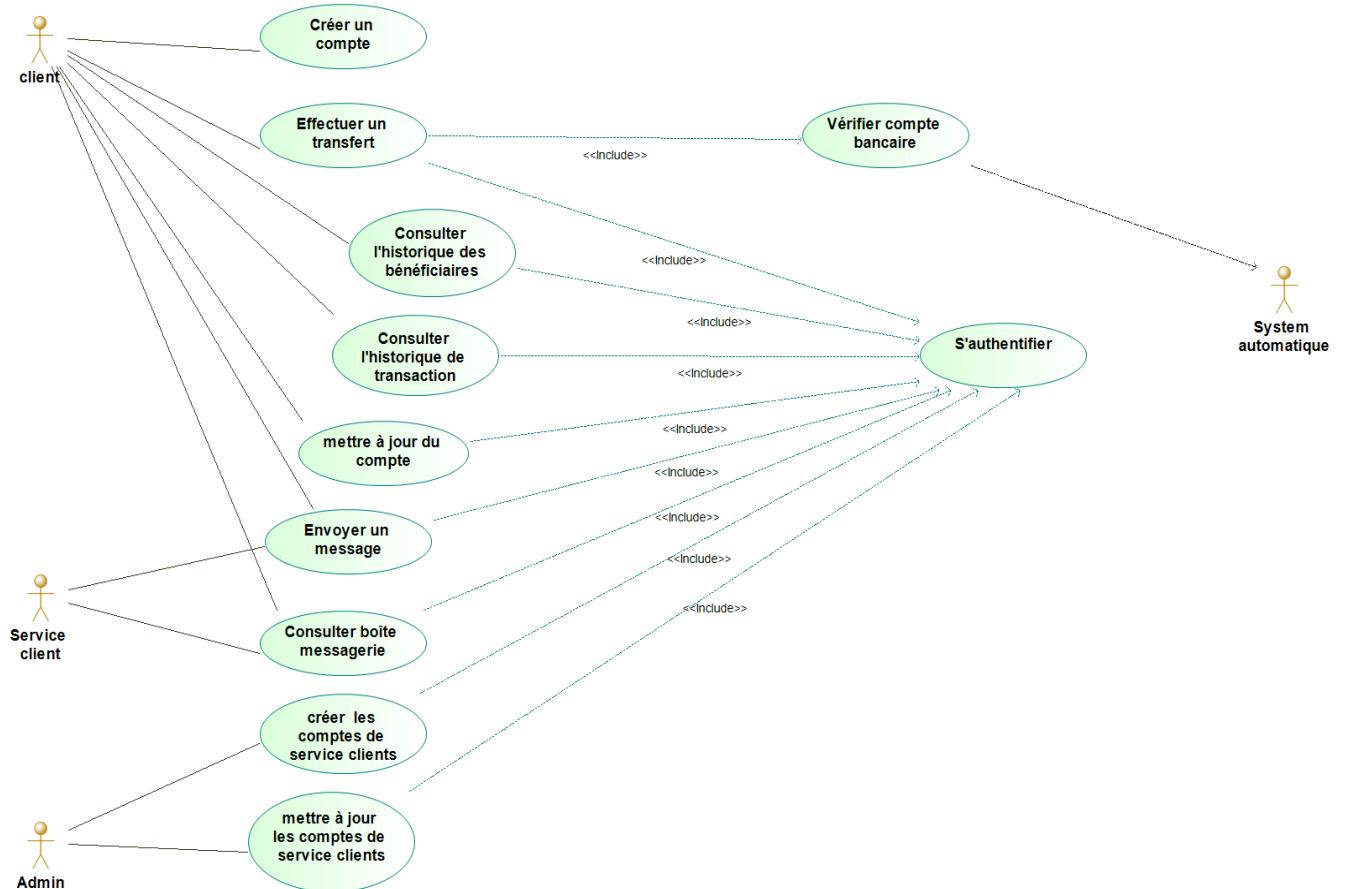


FIGURE 2.1 – Diagramme de cas d'utilisation.

- Description détaillée des cas d'utilisation :

Cas d'utilisation : Créer un compte, S'authentifier et Créer les comptes de service clients	
Titre	Créer un compte , S'authentifier , Créer les comptes de service clients
But	Créer un compte pour accéder aux fonctionnalités de l'application
Résumé	L'utilisateur doit remplir un formulaire d'inscription et identifier électroniquement leur document comme des pièces d'identité (carte d'identité, passeport) à partir de leur caméra du téléphone puis valide son action. Le système effectue une vérification puis une mise à jour de la base de données.
Acteurs	Client,Admin
Descriptions des enchainements	
Pré condition	Post condition
L'utilisateur doit accéder au système	L'utilisateur inscrit
Scenario nominal	
<ol style="list-style-type: none"> 1. L'utilisateur demande la page d'inscription en cliquant sur «S'INSCRIRE» 2. Le système lui envoie la page d'inscription 3. L'utilisateur remplit le formulaire et scanne leur document puis appuie sur S'INSCRIRE 4. Le système effectue les validations et l'enregistrement dans la base de données 5. L'utilisateur est redirigé vers la page d'authentification et renseigne ses identifiants en cas d'acteur client 	
Enchainement d'échec	
<ol style="list-style-type: none"> 6. Le compte existe déjà ou les données saisies sont incorrectes 7. Il n'y a pas de connexion internet 	

TABLE 2.1 – Description détaillée des cas d'utilisation : Créer un compte et S'authentifier

Cas d'utilisation : Effectuer transfert

Titre	Effectuer transfert
But	Envoyer de l'argent à un bénéficiaire
Résumé	L'utilisateur saisit les informations du bénéficiaire et le montant de la transaction et valide. Le système envoyait les informations de la carte bancaire au fournisseur de paiement avant de finaliser l'opération.
Acteurs	Client

Descriptions des enchainements

Pré condition	Post condition
<ul style="list-style-type: none"> - L'utilisateur doit se connecter - L'utilisateur doit utiliser une carte bancaire valide 	<ul style="list-style-type: none"> - Le transfert est effectué - Afficher l'historique des transactions

Scenario nominal

1. L'utilisateur accède au formulaire de transfert en saisit le montant puis cliquant sur « Envoyer »
2. Le système lui demande de saisir les informations de transfert
3. L'utilisateur rempli le formulaire appuie sur le bouton « Valider »
4. Le système débiter le compte de cadorim et crédité le compte de bénéficiaire
5. Le système lui affiche la page de l'historique des transactions

Enchainement d'échec

6. La carte bancaire invalide ou le solde est insuffisant
7. La saisie de données n'est pas correcte
8. La connexion n'est pas bonne pour effectuer et suivre un transfert

TABLE 2.2 – Description détaillée des cas d'utilisation : Effectuer transfert

Cas d'utilisation : Consulter l'historique des bénéficiaires et l'historique de transaction

Titre	Consulter l'historique des bénéficiaires,Consulter l'historique de transaction
But	Consultations de l'historique
Résumé	L'utilisateur voit les renseignements sur le bénéficiaire et l'opération de transaction.
Acteurs	Client

Descriptions des enchainements

Pré condition	Post condition
- L'utilisateur doit se connecter	- Afficher l'historique des transactions ou des bénéficiaires

Scenario nominal

1. L'utilisateur accède à l'historique des transactions ou des bénéficiaires en cliquant sur « Transferts » ou « Benef »
2. Le système lui affiche la page de l'historique

Enchainement d'échec

3. Il n'y a pas de connexion internet

TABLE 2.3 – Description détaillée des cas d'utilisation : Consulter l'historiques(des transactions ou des bénéficiaires)

Cas d'utilisation : Consulter boite messagerie

Titre	Consulter boite messagerie
But	Consulter boite messagerie
Résumé	L'utilisateur voit l'historique des messages et les messages reçus.
Acteurs	Client, Service client

Descriptions des enchainements

Pré condition	Post condition
- L'utilisateur doit se connecter	- Afficher l'historique des messages et les messages reçus

Scenario nominal

1. L'utilisateur accède à la boîte messagerie
2. Le système lui affiche la page des messageries

Enchainement d'échec

3. Il n'y a pas de connexion internet

TABLE 2.4 – Description détaillée des cas d'utilisation : Consulter boite messagerie

Cas d'utilisation : Envoyer un message	
Titre	Envoyer un message
But	Envoyer un message
Résumé	L'utilisateur peut envoyer un message où répond à un message
Acteurs	Client, Service client
Descriptions des enchainements	
Pré condition	Post condition
- L'utilisateur doit se connecter	- Le message est envoyé
Scenario nominal	
1. L'utilisateur demande le formulaire d'envoi de message	
2. Le système lui affiche la page des discussions	
Enchainement d'échec	
3. Il n'y a pas de connexion internet	

TABLE 2.5 – Description détaillée des cas d'utilisation : Envoyer un message

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 ».

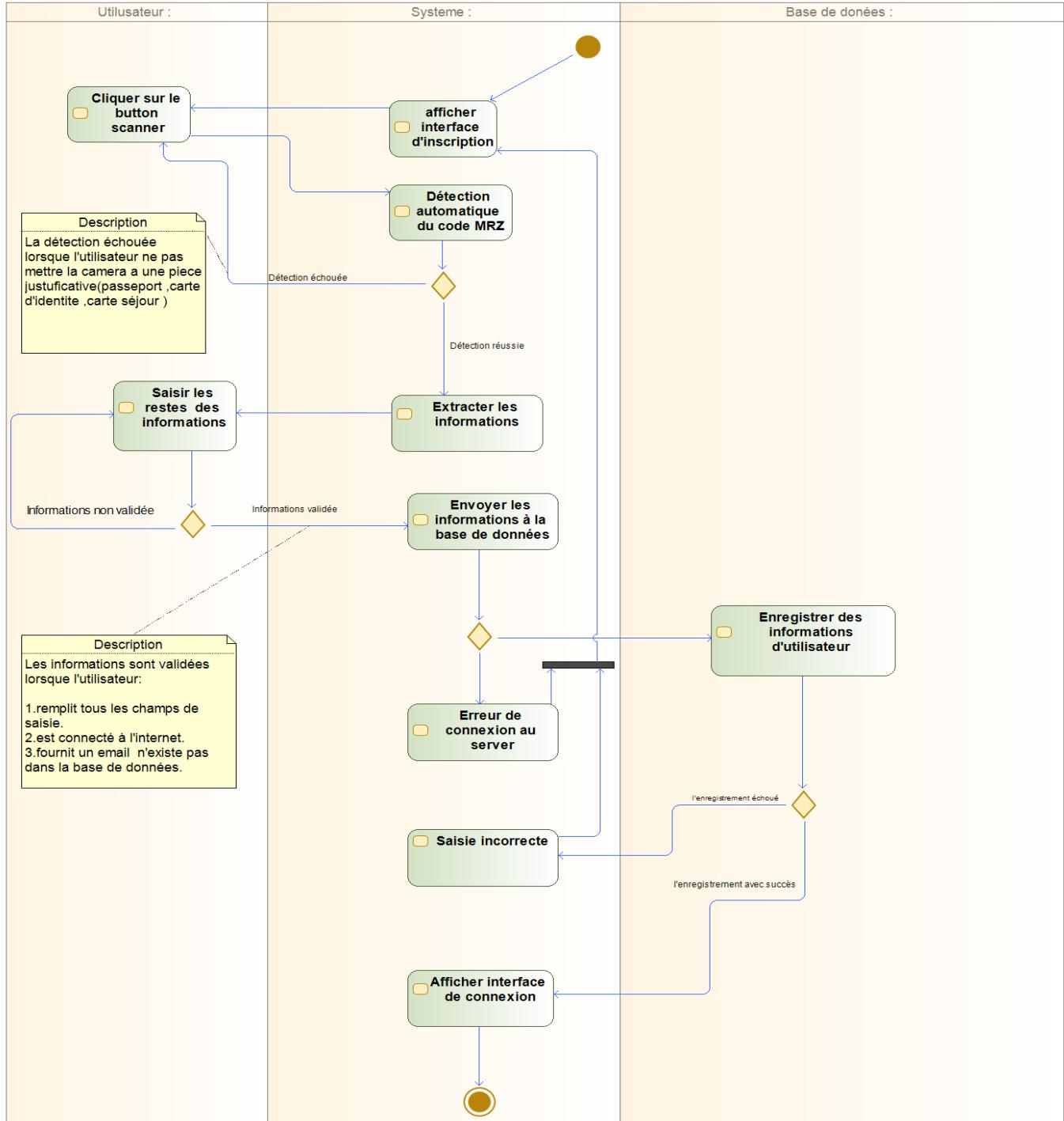


FIGURE 2.2 – Diagramme d'activit  : Cr ation de compte

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.

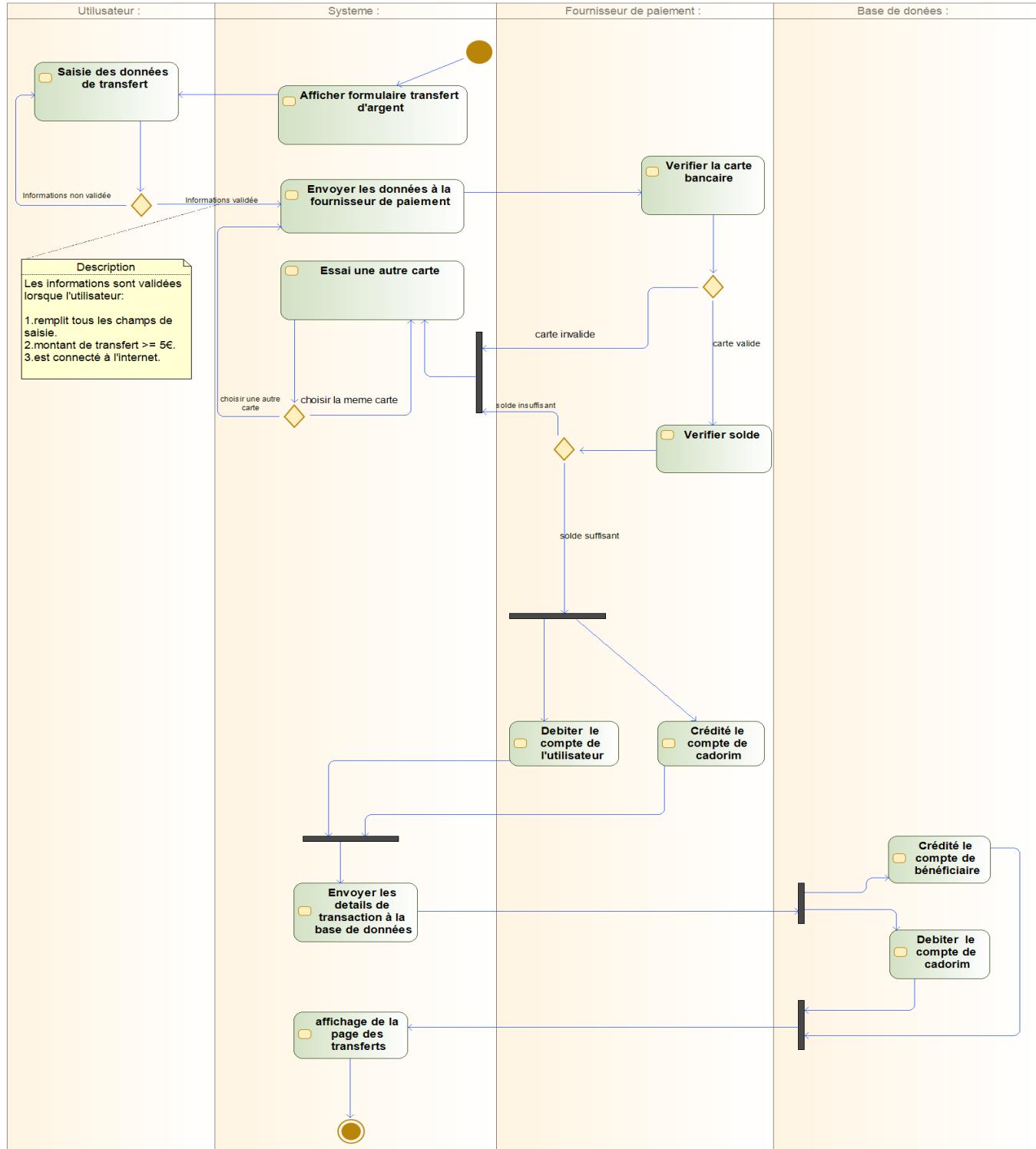


FIGURE 2.3 – Diagramme d'activité : Transfert d'argent

2.1.2.3 Authentification

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

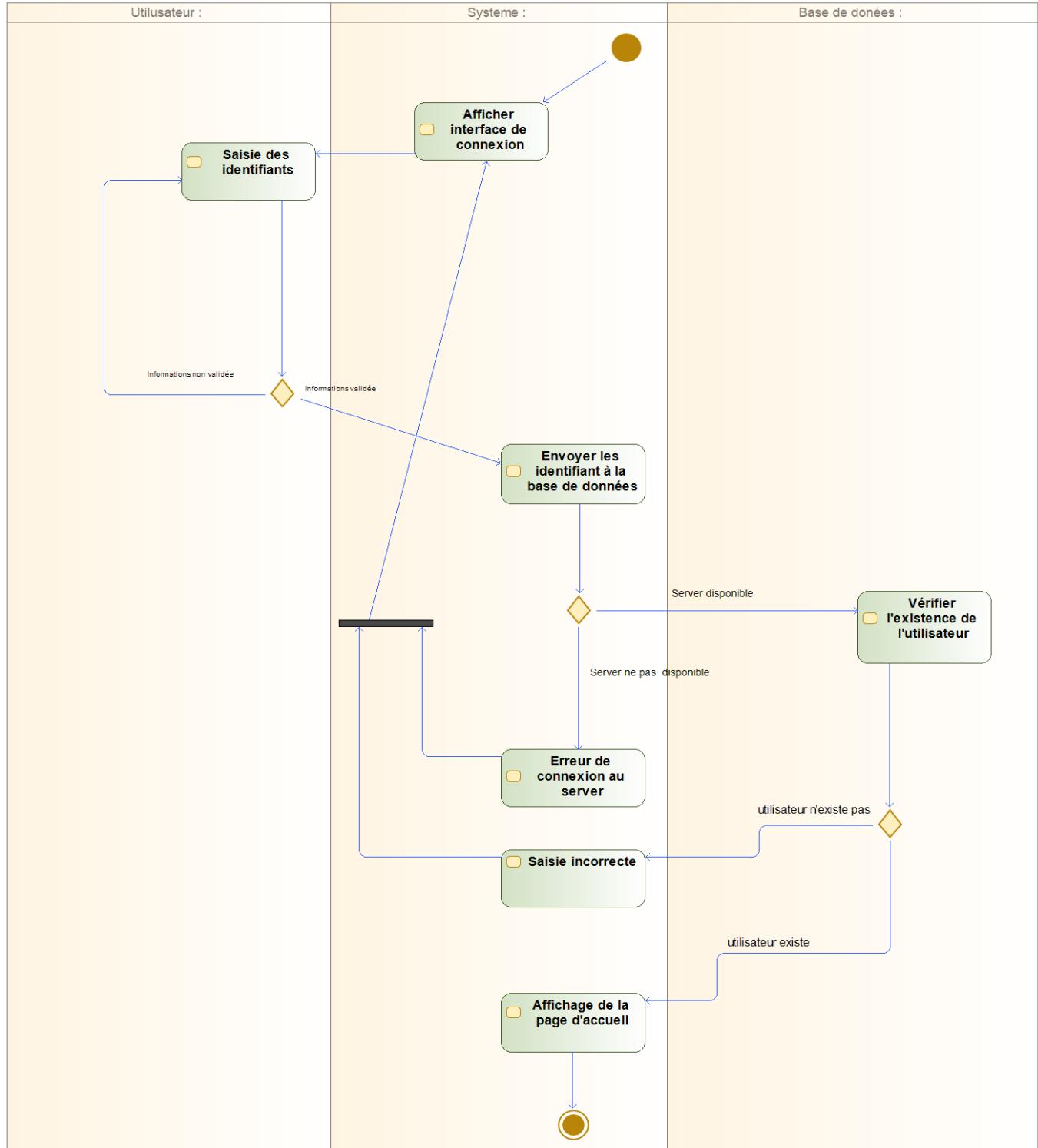


FIGURE 2.4 – Diagramme d'activité : Authentification

2.1.3 Diagramme de classe

Afin de bien détailler l'architecture de la base de données, nous avons conçu le diagramme de classe représenté dans la figure 2.5.

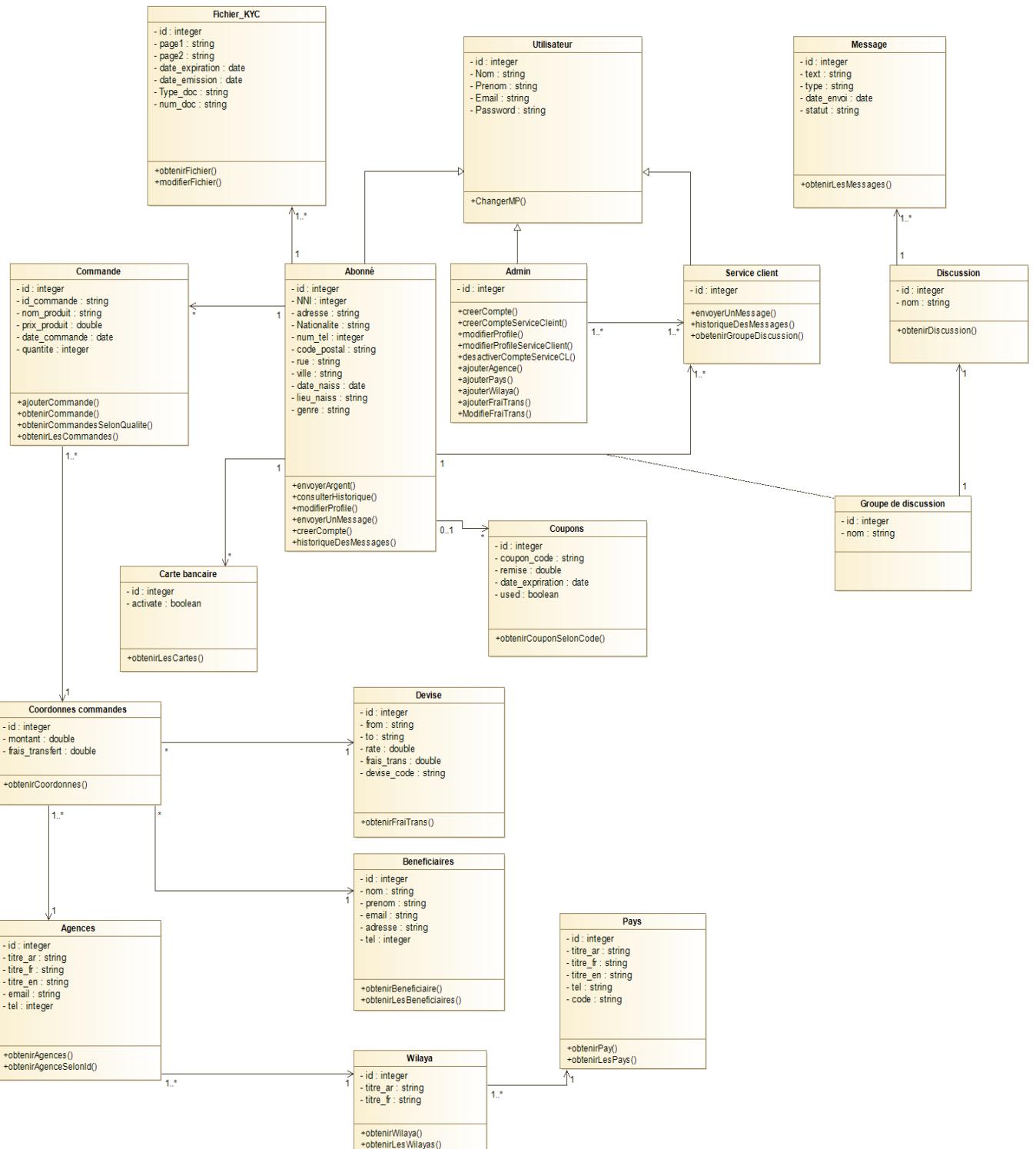


FIGURE 2.5 – Diagramme de classe

Le tableau 2.6 explique les différentes classes de la base de données.

Classe	Description
Abonné	Informations de l'utilisateur
Fichier KYC	Contient les informations du fichier (carte d'identité, passeport, titre de séjour, carte bancaire) de client
Commande	Informations concernant une commande
Coordonnes commandes	Détails de chaque commande
Agences	Informations sur les agences de cadorim
Carte bancaire	Gestion de carte bancaire
Devise	Informations sur la devise selon le pays d'envoi et le pays reçoivent
Beneficiaires	Informations sur les Beneficiaires
Wilaya	Informations sur les Wilayas
Admin	Informations de l'Admin et ses privilèges
Service client	Service client avec certains privilèges
Coupons	Informations d'un coupon
Groupe de discussion	Informations sur les groupes
Discussion	Informations de discussion de chaque groupe
Message	Contient les messages de chaque discussion
Pays	Informations sur les pays

TABLE 2.6 – Description de la base de données

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 Editeur de texte : 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>

3.1.2 IDE : 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>.

3.1.3 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/>

3.1.4 Le SGBD 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 :

3.1.4.1 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.

3.1.4.2 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.

3.1.4.3 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.

3.1.4.4 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.

3.1.5 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.

3.1.6 Logiciel de modélisation : 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.

3.1.7 Motifs d'architecture logicielle : 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.

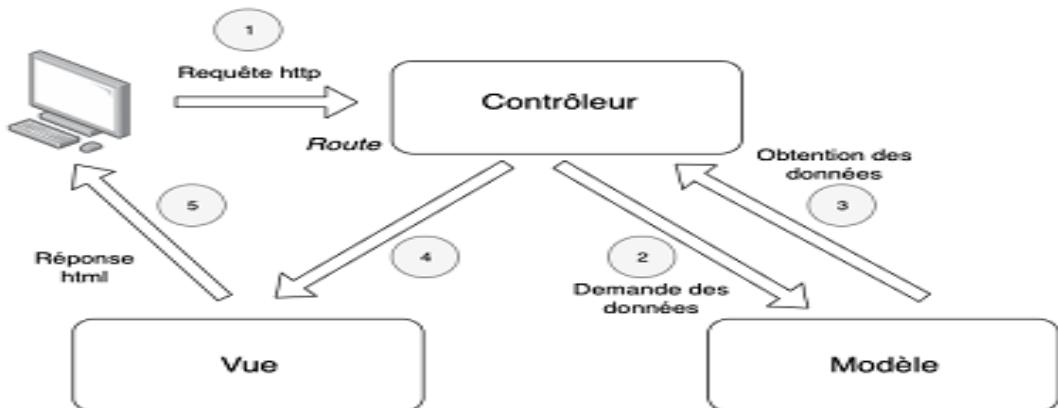


FIGURE 3.1 – MVC

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 .

3.1.8 Serveur Web : 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.

3.2 Choix des outils de travail

3.2.1 Langages utilisés

3.2.1.1 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

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>

3.2.1.3 Langage de modélisation : 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.

3.2.2 Frameworks utilisés

3.2.2.1 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 recharge à chaud (hot reload)
3. Open-source et par Google
4. Programmation Dart

- Architecture d'une application Flutter : Flutter a une architecture modulaire qui effectivement permet d'écrire le code une seule fois et de l'utiliser sur plusieurs plateformes

(Android, ios, ou web) même si le moteur est différent de plateforme à plateforme, pour le développeur c'est transparent. voici en image cette architecture :

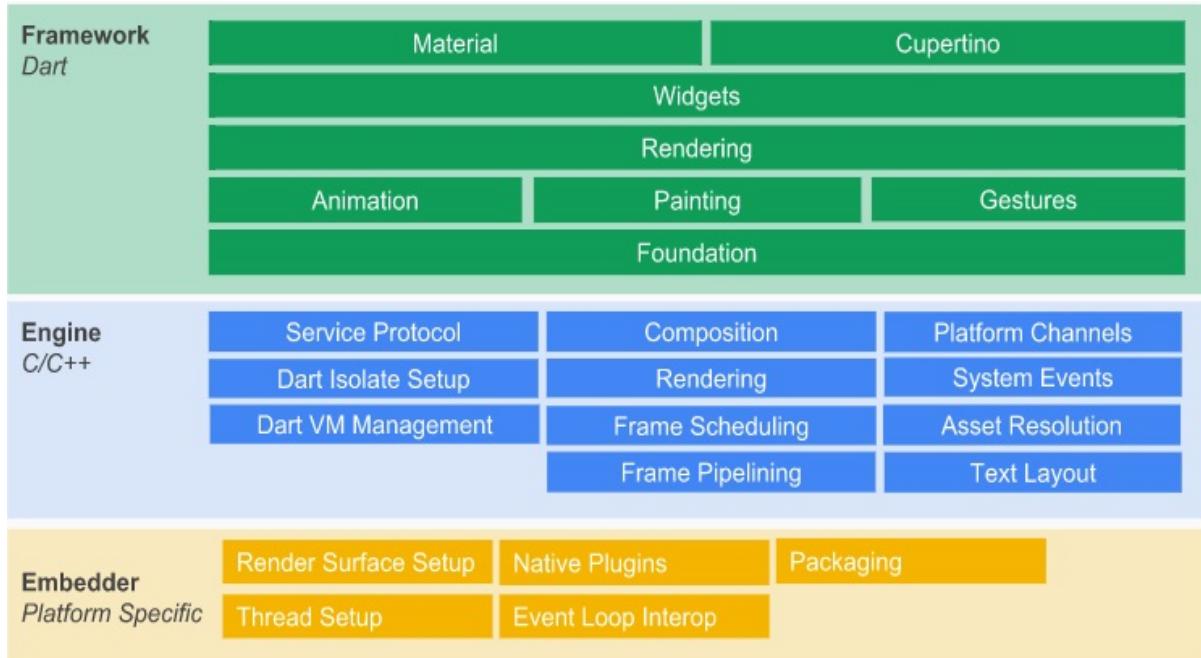


FIGURE 3.2 – Architecture d'une application Flutter.

L'architecture Flutter comprend principalement quatre composants :

- **Flutter Engine (Moteur de flutter)** : Le Flutter Engine est un environnement d'exécution portable pour les applications mobiles de haute qualité. Il implémente les bibliothèques principales de Flutter, y compris l'animation et les graphiques, les E/S de fichiers et de réseau, la prise en charge de l'accessibilité, l'architecture des plugins, ainsi qu'un environnement d'exécution et une chaîne d'outils Dart pour développer, compiler et exécuter des applications Flutter.
- **Foundation Library (Bibliothèque de la Fondation)** : Il contient tous les packages requis pour les éléments de base de l'écriture d'une application Flutter. Ces librairies sont écrites en langage Dart.
- **Widgets (Widget)** :
 - * Les widgets sont les éléments de base de l'interface utilisateur dans Flutter.
 - * La conception de l'interface utilisateur pour une application implique la composition et la modification de divers widgets simples tels que du texte, des formes et des animations pour en créer des plus complexes.

- * Les widgets Flutter ne sont que des éléments de votre interface utilisateur.
Si vous êtes familier avec le développement Android ou iOS, vous établirez la connexion immédiate aux vues (pour Android) et aux UIViews (pour iOS). C'est une bonne comparaison à faire et vous ferez bien de commencer votre voyage avec cet état d'esprit. Une façon plus précise de penser, cependant, est qu'un widget est un plan
- **Embedder Platform Specific (widgets spécifiques)** : Embbeder est différent pour chaque plateforme et son rôle est de créer l'exécutable ou les modules pour chaque plateforme.
 - * Les applications Flutter sont packagées de la même manière que toute autre application native pour les plates-formes cibles.
 - * Flutter fournit un certain nombre d'intégrateurs spécifiques à la plate-forme pour les plates-formes cibles qui coordonnent une application flottante avec le système d'exploitation sous-jacent pour accéder à des services tels que les surfaces de rendu, l'accessibilité, la saisie et la gestion de la boucle d'événements de message.
 - * L'intégrateur est généralement écrit dans le meilleur langage approprié pour la cible. plate-forme.
 - * L'embedder pour Android est écrit en C++ et Java, Objective C/Objective C++ pour iOS et macOS, et C++ pour Windows et Linux. Avec l'utilisation de l'embedder, le code Flutter peut également être intégré dans une application existante en tant que module.

Pour en savoir plus, veillez visiter le lien : <https://apcpedagogie.com/architecture-d'une-application-flutter/>

3.2.2.2 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](#).

- Architecture de Laravel :

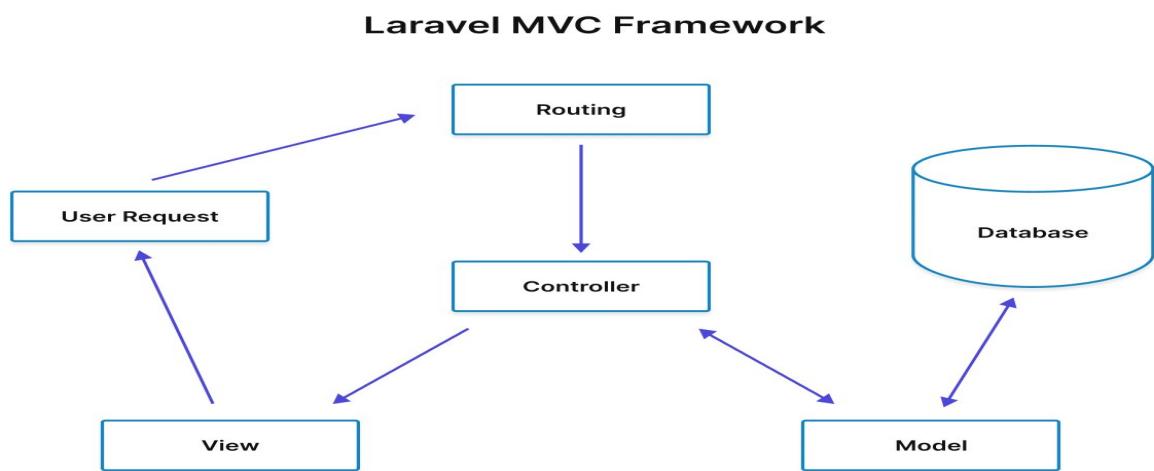


FIGURE 3.3 – Architecture de Laravel

3.3 Implémentation

3.3.1 Etape de réalisation

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

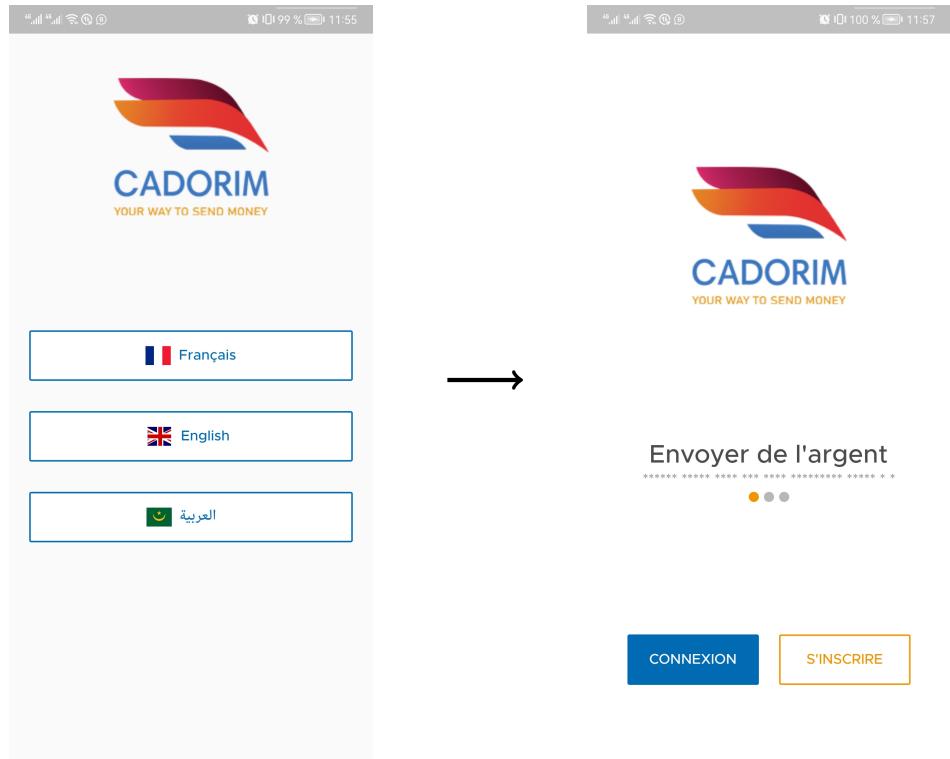
- * **Conception de la base de données** : La conception d'une base de données est la première étape. Le choix des algorithmes et de l'approche de travail exige l'utilisation d'une base de données spécifique (en fonction du système à développer).
- * **Extraction des données** : On va utiliser notre 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 fonctionnalités avec leur interfaces graphiques 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 3.4 – 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.

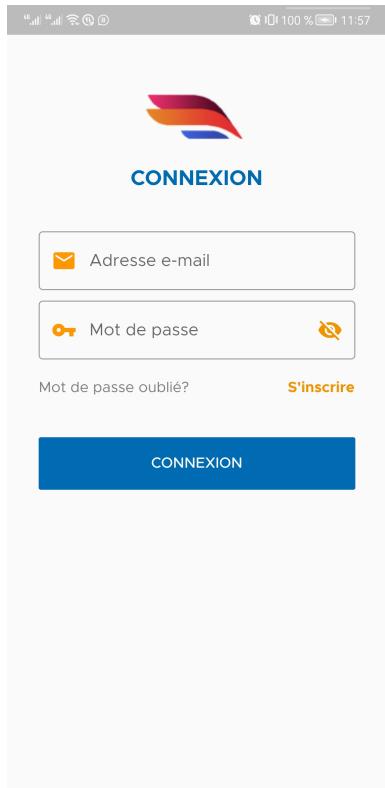
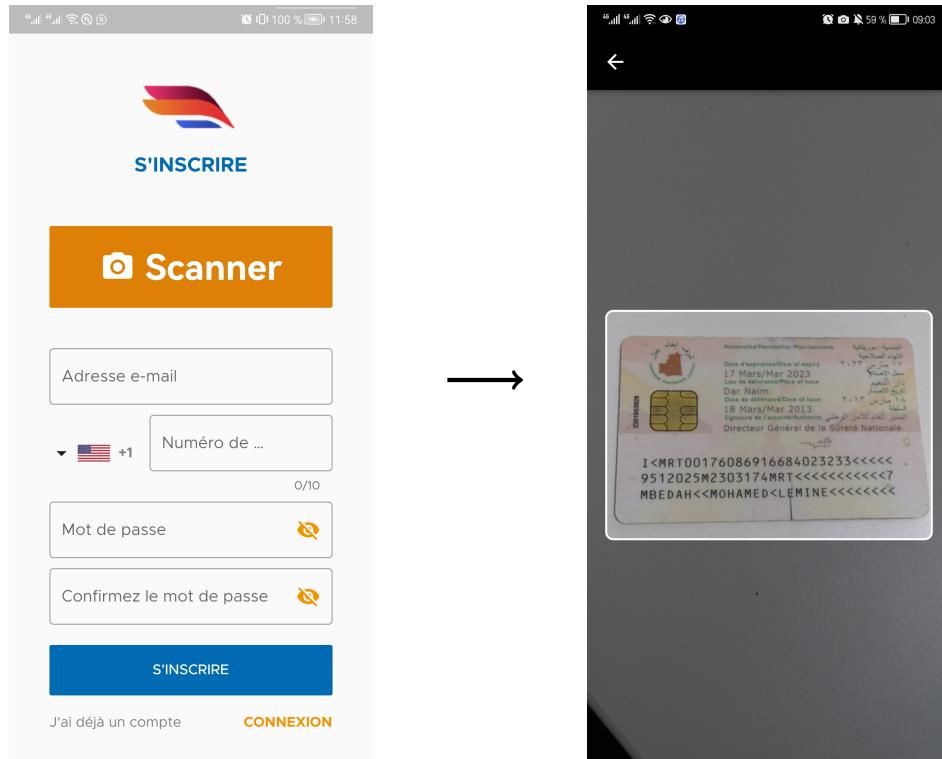


FIGURE 3.5 – Interface d’authentification

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.
- * **Processus de la transaction :** La figure 3.7 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)



(a) Interface d'inscription

(b) Scanner le code MRZ

The form includes fields for:

- Photo du face avant** (Front face photo)
- Photo du face arrière** (Back face photo)
- Adresse e-mail**: med.lemine.lemine@cado.com
- Numéro de téléphone**: +222 27666242
- Mot de passe** and **Confirmez le mot de passe** (Password and Confirm password fields)

(c) Saisir le rest des données

FIGURE 3.6 – Interface d'inscription

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 -

Brakna - البراكنة -

Dakhlet Nouadhibou - دakhlet Nouadhibou - 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

Wilaya: Aioun -Doueirare - العيون - دويرارة -

Agence: Aioun -Egjert - العيون- اكجرة -

Agence: Aioun -Marche - العيون- السوق -

Agence: Aioun -Météo - العيون- متىو -

Agence: Aioun -N'Savenni - العيون- انصفني -

Agence: Aioun -Ten Hamadi - العيون- تن حمادي

Agence: Aleg - Aghchorguitt - ألاك -

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

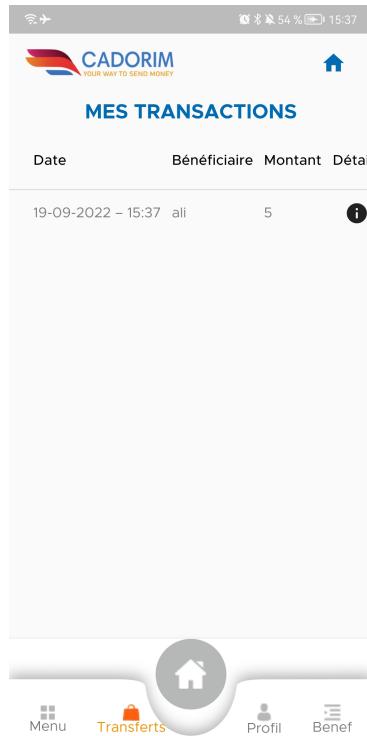
VALIDER ANNULER

(f) État de transfert

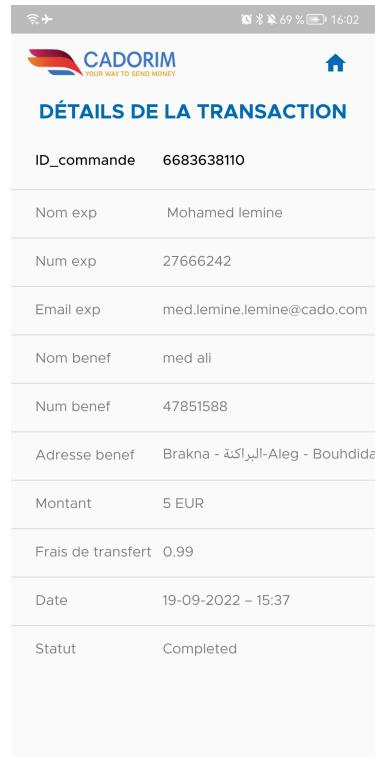
FIGURE 3.7 – Processus de transactions



(a) Transfere avec succès



(b) Historique transactions



(c) Détails de la transaction

FIGURE 3.8 – Transferts

- * **L'interface de transfert :** La figure 3.8 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.

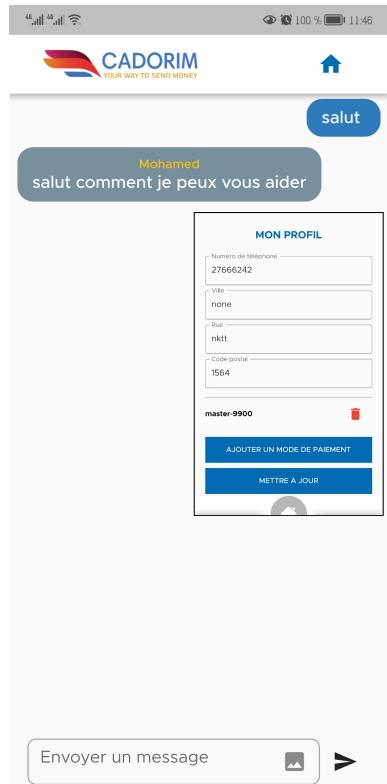
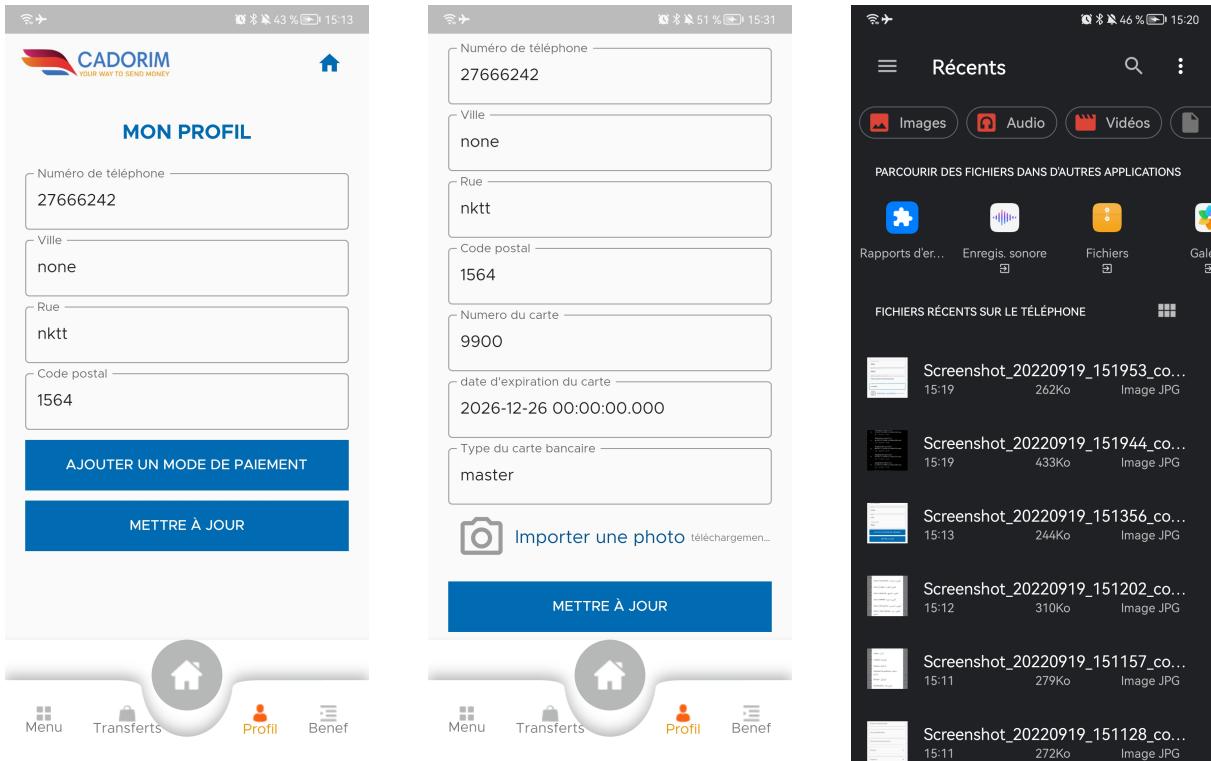


FIGURE 3.9 – Interface de discussion

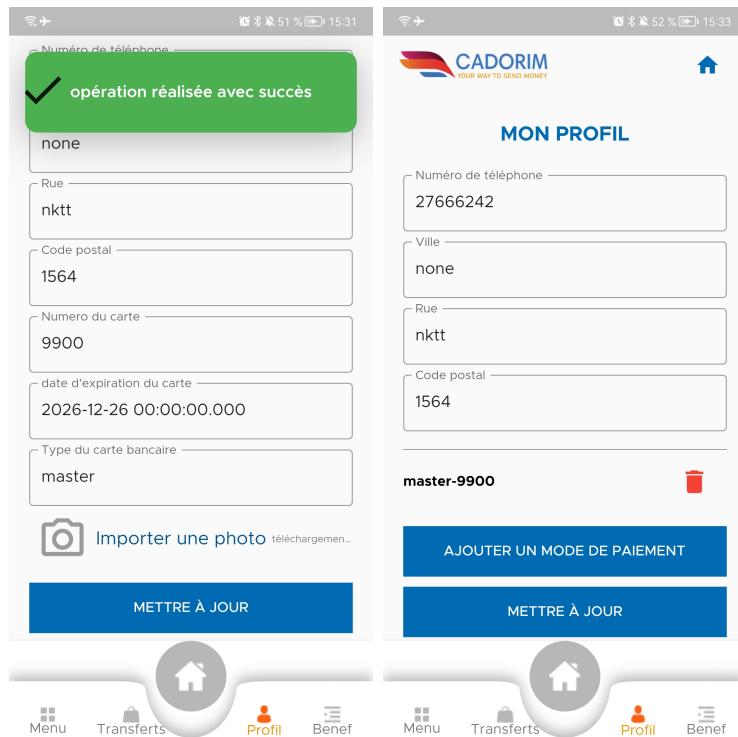
- * **Profil de l'utilisateur :** L'utilisateur peut mettre à jour leur profil ou ajouter un mode paiement comme il montre la figure suivante



(a) Mon profil

(b) Ajouter une carte

(c) Importer une photo



(d) État

(e) Interface suivante

FIGURE 3.10 – Profil de l'utilisateur

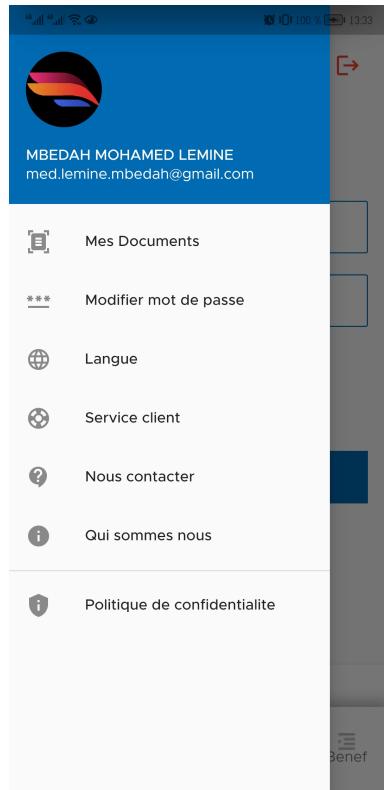


FIGURE 3.11 – Interface de menu

- * **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.

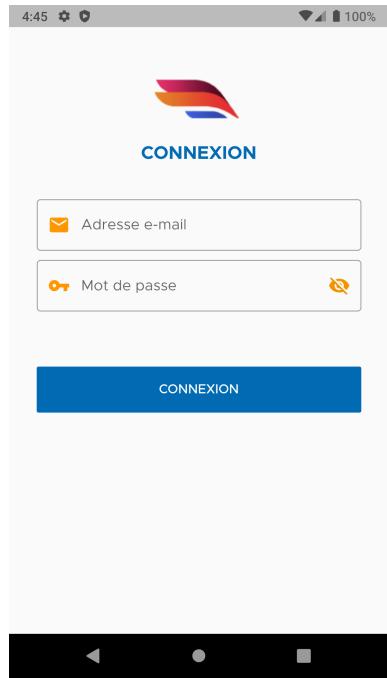
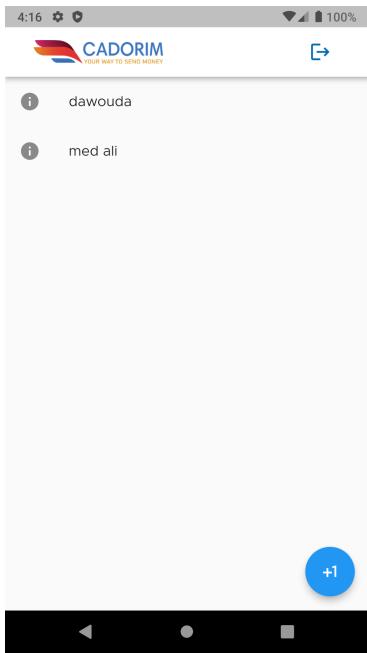
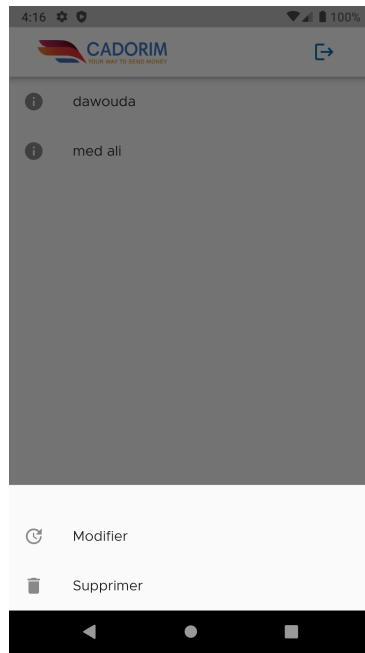


FIGURE 3.12 – Interface d'authentification

- * **L'interface du compte administrateur :** La figure 3.13 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).



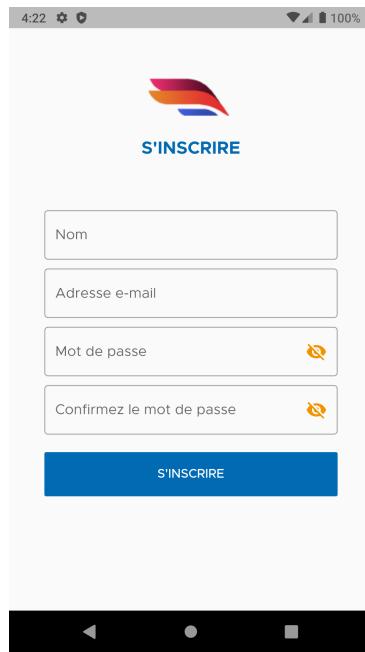
(a) Interfaces d'accueil



(b) Gérer un compte



(c) Mettre à jour un compte



(d) Ajouter un compte

FIGURE 3.13 – Interface du compte administrateur

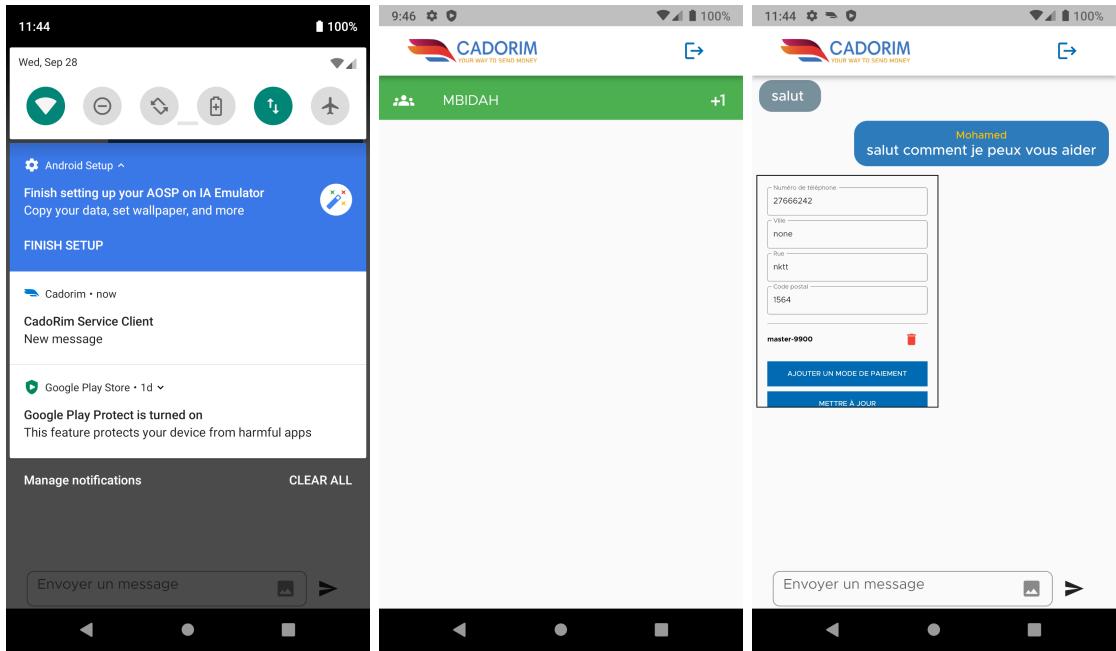


FIGURE 3.14 – Interface du compte service client

- * **L'interface du compte service client** : La figure 3.14 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, nous 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 plateformes 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.