



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

Les mots ne sauront traduire ce qui est dans le cœur, 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 je suis arrivé là où je suis maintenant. Tous les remerciements et gratitudes ne remplissent pas leur droit.

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 but de ce travail est la reimagination de l'application CADORIM pour être à jour à l'évolution technologie dans le monde. La start-up Cadorim qui exerce dans le domaine du transfert d'argent de l'Europe vers la Mauritanie a proposé de créer un system d'obtention des informations de client pour la bonne connaissance au client (KYC) et de créer un system de gestion de relation client (CRM).Ce qui nécessite la reconstruction de l'application CADORIM, d'une part pour faciliter mon adaptation d'un l'environnement de travail dans la société et d'autre part assure la bonne intégration de nouveaux changements.

La nouvelle version de l'application permettra la validation automatique des comptes clients d'une manière automatique.

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

1	Introduction	9
1.1	Motivations	10
1.2	Problématiques	10
1.3	Objectifs	11
2	Présentation de l'entreprise	12
2.1	Introduction	12
2.2	Missions	13
2.3	Planification du projet	13
3	Analyse et Spécification des Besoins	14
3.1	Introduction	14
3.2	User expérience	14
3.3	Différence entre UX et UI	14
3.4	Mise en place du processus	14
3.4.1	L'atelier (workshop)	14
4	Analyse fonctionnelle et conceptuelle	15
4.1	Analyse fonctionnelle	15
4.1.1	Diagramme de cas d'utilisation	16
4.1.2	Diagramme d'activité	17
4.1.2.1	Création de compte	17
4.1.2.2	Transfert d'argent	18
4.1.2.3	Authentification	19
4.2	Modélisation de la base de données	19
4.2.1	Diagramme de classe	19

5	Les technologies utilisées	22
5.0.1	Front-end	22
5.0.1.1	Editeur de texte : VS Code	22
5.0.1.2	Language : Dart	23
5.0.1.3	Framework : Flutter	23
5.0.1.4	IDE : Android Studio	24
5.0.2	Back-End	24
5.0.2.1	Language : PHP	24
5.0.2.2	Serveur Web : Amazon Web Services (AWS)	25
5.0.2.3	Framework :Laravel	25
5.0.2.4	SGBD MariaDB	25
5.0.3	Motifs d'architecture logicielle	26
5.0.3.1	MVC	26
6	Implémentation	28
6.1	Architecture de l'application	28
6.2	Interfaces graphiques	29
6.2.1	Interfaces d'accueil	29
7	Interface et utilisation de l'application	31

Table des figures

2.1	CADOROM	12
4.1	Diagramme de cas d'utilisation.	16
4.2	Diagramme d'activité : Création de compte	17
4.3	Diagramme d'activité : Transfert d'argent	18
4.4	Diagramme d'activité : Authentification	20
4.5	Diagramme d'activité : Transfert d'argent	21
5.1	Architecture d'une application Flutter.	24
5.2	MVC	26
5.3	PHP MySQL REST API pour Android	27
6.1	Architecture logicielle de l'application.	29
6.2	Interfaces d'accueil	30

Chapitre 1

Introduction

Savoir qui est votre client et adopter des protocoles pour prévenir la criminalité financière sont des défis permanents pour les institutions financières. De manière significative, les institutions financières (y compris les banques, les coopératives de crédit et les sociétés financières du Fortune 50) doivent se conformer à un ensemble des réglementations de plus en plus complexes pour la vérification de l'identité des clients appelée KYC.

KYC, également connu sous le nom de "Know Your Customer" ou "Know Your Client", est un ensemble de procédures permettant de vérifier l'identité d'un client avant ou pendant les transactions avec les banques et autres institutions financières. Le respect des réglementations KYC peut aider à tenir à distance le blanchiment d'argent, le financement du terrorisme et d'autres stratagèmes de fraude courants. En vérifiant d'abord l'identité et les intentions d'un client au moment de l'ouverture du compte, puis en comprenant ses habitudes de transaction, les institutions financières sont en mesure d'identifier plus précisément les activités suspectes.

Les institutions financières sont soumises à des normes de plus en plus strictes en matière de lois KYC. Ils doivent dépenser plus d'argent pour se conformer à KYC ou être passibles de lourdes amendes. Ces réglementations signifient que presque toutes les entreprises, plateformes ou organisations qui interagissent avec une institution financière pour ouvrir un compte ou effectuer des transactions devront se conformer à ces obligations.

La gestion de la relation client (CRM) est la combinaison de pratiques, de stratégies et de technologies que les entreprises utilisent pour gérer et analyser les interactions et les données client tout au long du cycle de vie du client. L'objectif est d'améliorer les relations de service client, de contribuer à la fidélisation de la clientèle et de stimuler la croissance

des ventes. Les systèmes CRM compilent les données client à travers différents canaux, ou points de contact, entre le client et l'entreprise, qui peuvent inclure le site Web de l'entreprise, le téléphone, le chat en direct, le publipostage, les supports marketing et les réseaux sociaux. Les systèmes CRM peuvent également donner aux membres du personnel en contact avec les clients des informations détaillées sur les informations personnelles des clients, l'historique des achats, les préférences et les préoccupations d'achat.

1.1 Motivations

KYC est un moyen de rendre la vérification de l'identité des clients plus précise et moins vulnérable à la fraude.

KYC doivent être effectuées lors de l'intégration d'un nouveau client, mais il est préférable de répéter ces vérifications de temps en temps, pour s'assurer que tout est comme il se doit. En surveillant les comptes clients de cette manière, les comportements suspects peuvent être signalés plus rapidement.

Un système CRM fournit des flux de travail automatisés qui permettent à votre équipe marketing de consacrer plus de temps à des tâches stratégiques, telles que la création de campagnes marketing qui résonnent, l'analyse des données de ces campagnes et le test de différentes approches basées sur ces analyses. Les agents du service client peuvent passer leur temps à travailler avec des clients qui ont des questions, des problèmes ou des besoins plus complexes. En bref, avec des processus de service client plus efficaces, les entreprises peuvent établir de meilleures relations avec leurs clients.

1.2 Problématiques

En réalité, la réalisation d'une application, qui applique le principe de KYC et intègre un système CRM, nécessite de faire face à des problématiques diverses et complexes. Ainsi, la société a décidé de se contenter, dans un premier temps, de la mise en place d'un système d'extraction des données à partir des images (carte d'identité ou passeport) et traitement de ces données. Ce sujet soulève de nombreuses questions aux implications différentes. Comment peut-on extraire le texte à partir de l'image ? Comment sera-t-il traité ? Comment peut-il être utilisé dans le principe KYC ? Comment pouvons-nous obtenir un système CRM intégré ?

1.3 Objectifs

La mise en place d'une application pour appliquer l'idée de KYC en basant sur les différentes technologies disponibles. En basant sur l'extraction du texte à partir d'une image OCR on peut extraire le code MRZ à partir d'une image d'une pièce d'identité ou d'un passeport. Le code est passé à un algorithme qui permet de détecter les informations personnelles.

Chapitre 2

Présentation de l'entreprise

2.1 Introduction



FIGURE 2.1 – CADORIM

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 est 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 marche capital , Nouakchott, Mauritanie, immatriculée au registre du commerce.

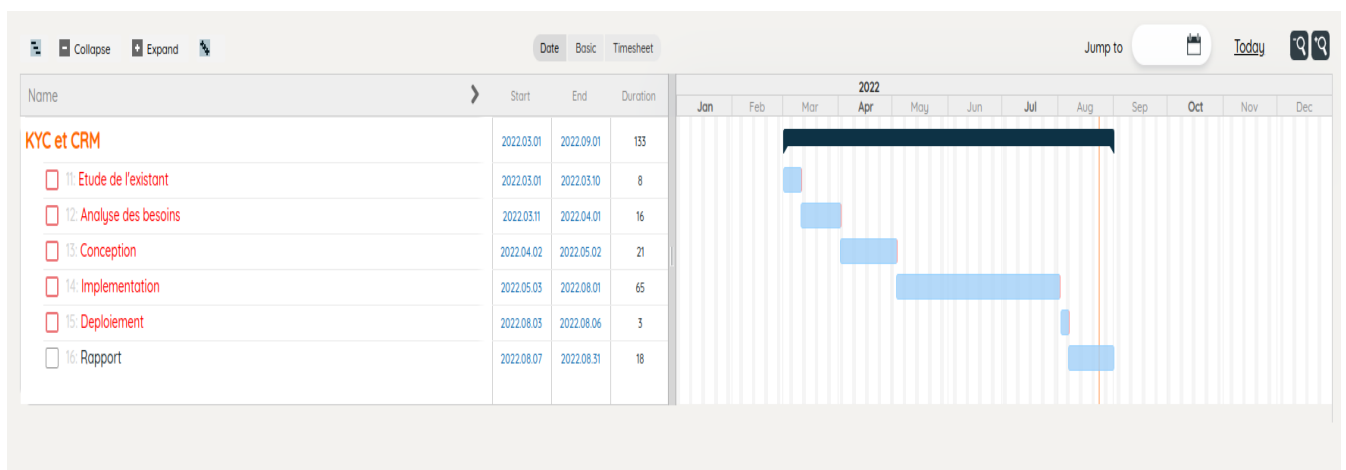
2.2 Missions

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

2.3 Planification du projet

J'effectuais le diagramme de Gantt, pour avoir une meilleure compréhension de la chronologie des étapes de mon 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 3

Analyse et Spécification des Besoins

3.1 Introduction

3.2 User expérience

3.3 Différence entre UX et UI

3.4 Mise en place du processus

3.4.1 L'atelier (workshop)

Chapitre 4

Analyse fonctionnelle et conceptuelle

4.1 Analyse fonctionnelle

4.1.1 Diagramme de cas d'utilisation

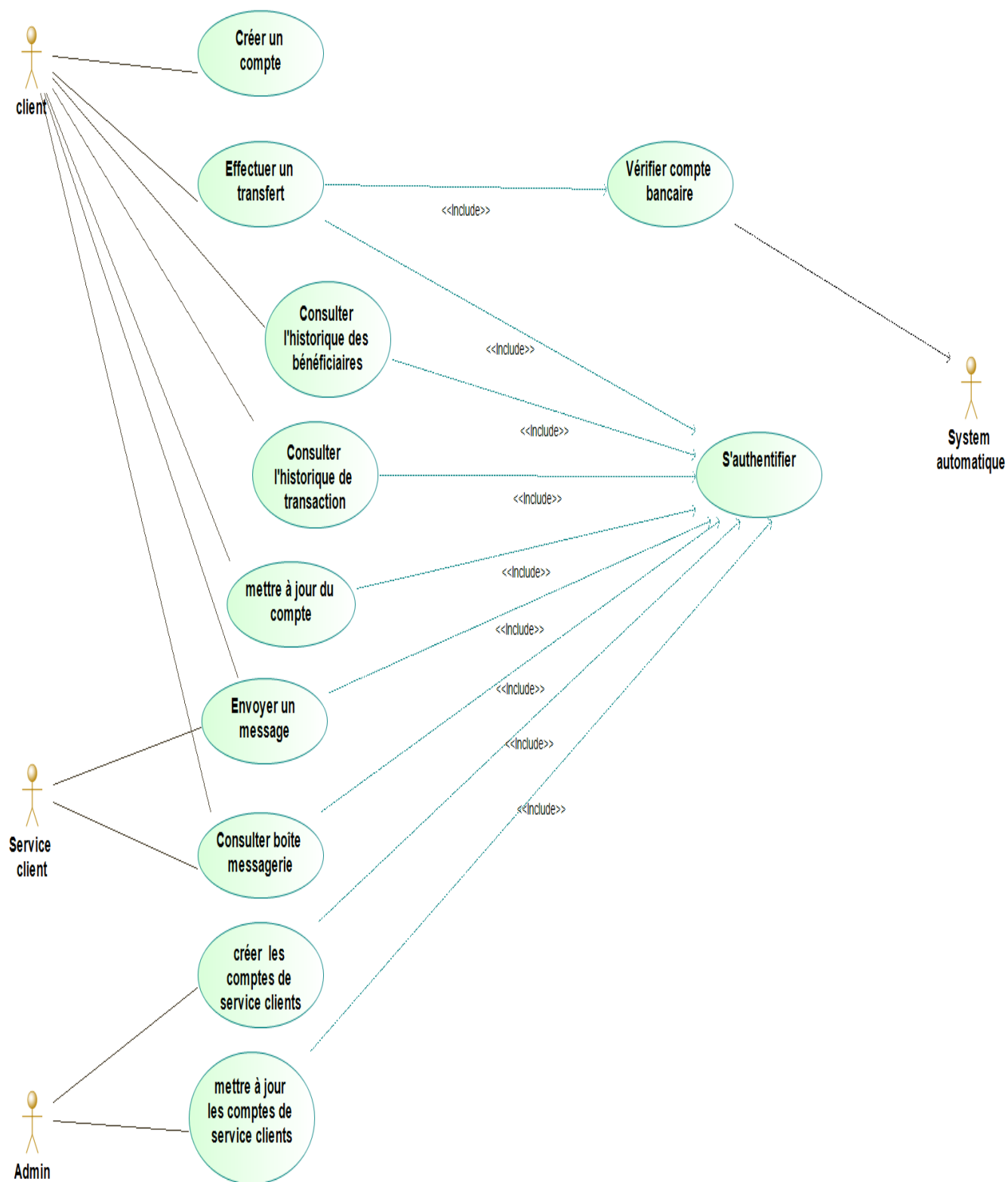


FIGURE 4.1 – Diagramme de cas d'utilisation.

4.1.2 Diagramme d'activité

4.1.2.1 Création de compte

Le diagramme d'activité qu'illustre la figure 4.2 décrit le cas d'utilisation « Créer un compte ».

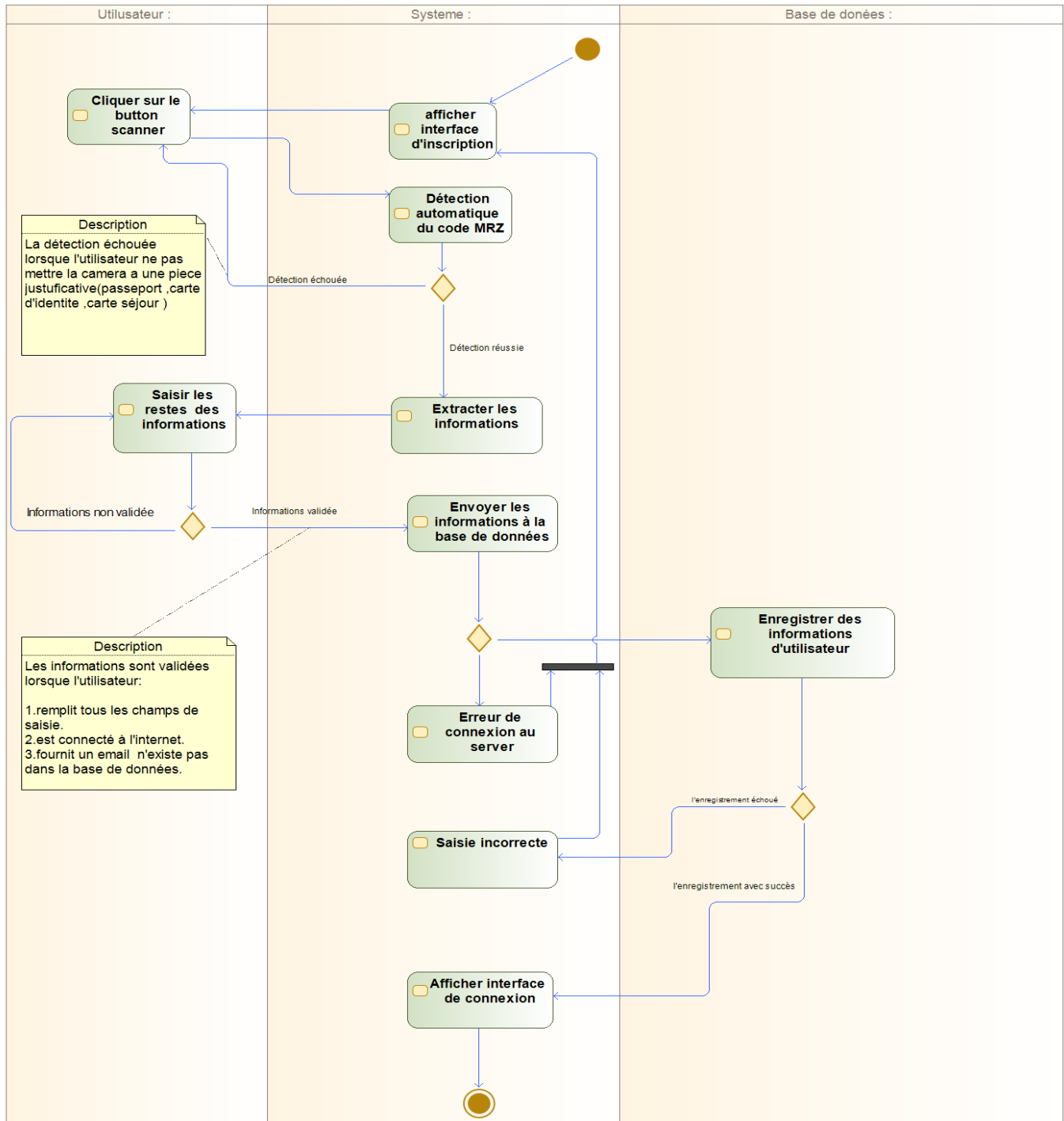


FIGURE 4.2 – Diagramme d'activité : Création de compte

4.1.2.2 Transfert d'argent

Le diagramme d'activité qu'illustre la figure 4.3 décrit les différentes actions ou enchainements effectués lors d'une opération de transfert d'argent.

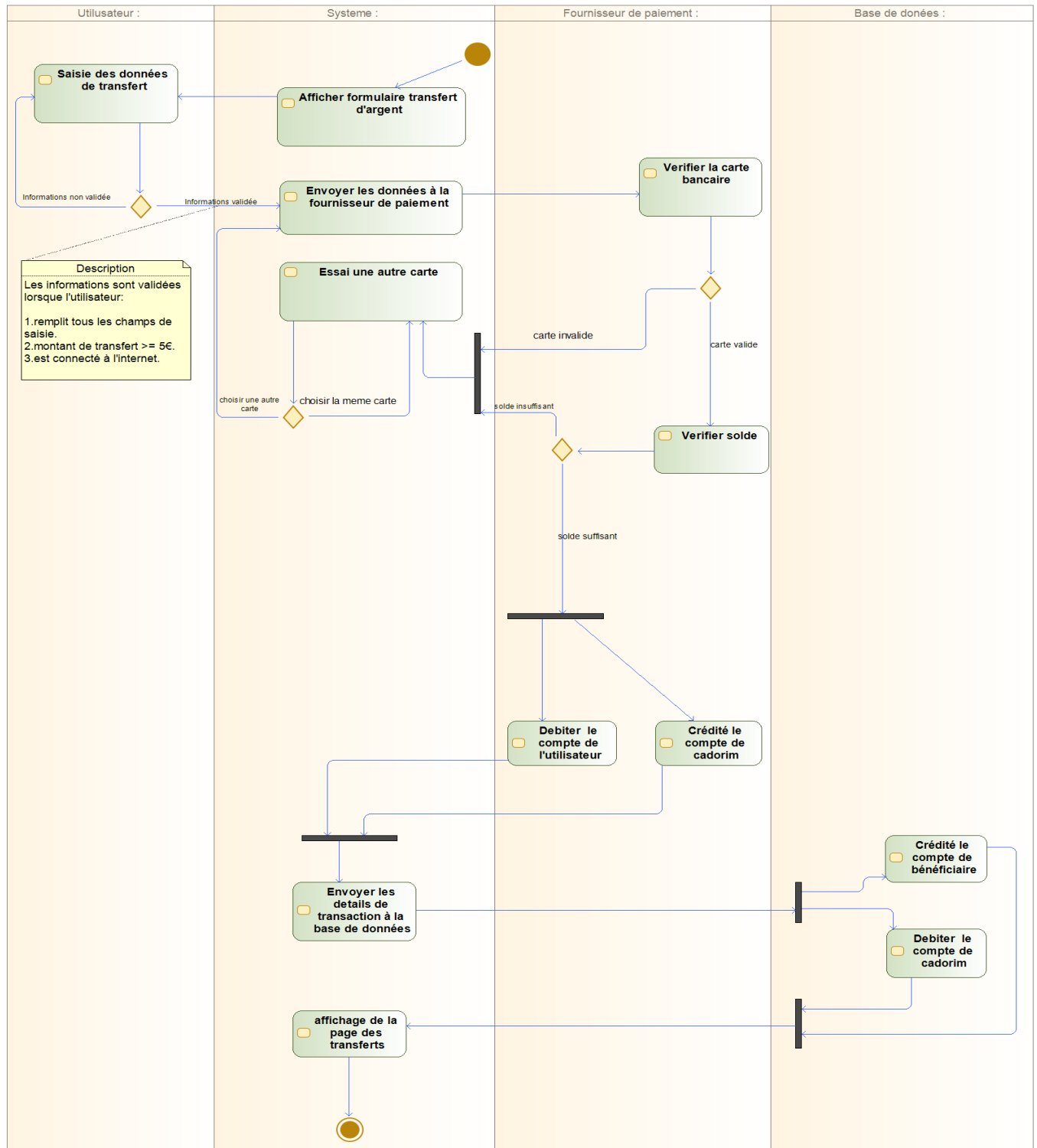


FIGURE 4.3 – Diagramme d'activité : Transfert d'argent

4.1.2.3 Authentification

Le diagramme d'activité qu'illustre la figure 4.4 décrit le cas d'utilisation « Authentification ».

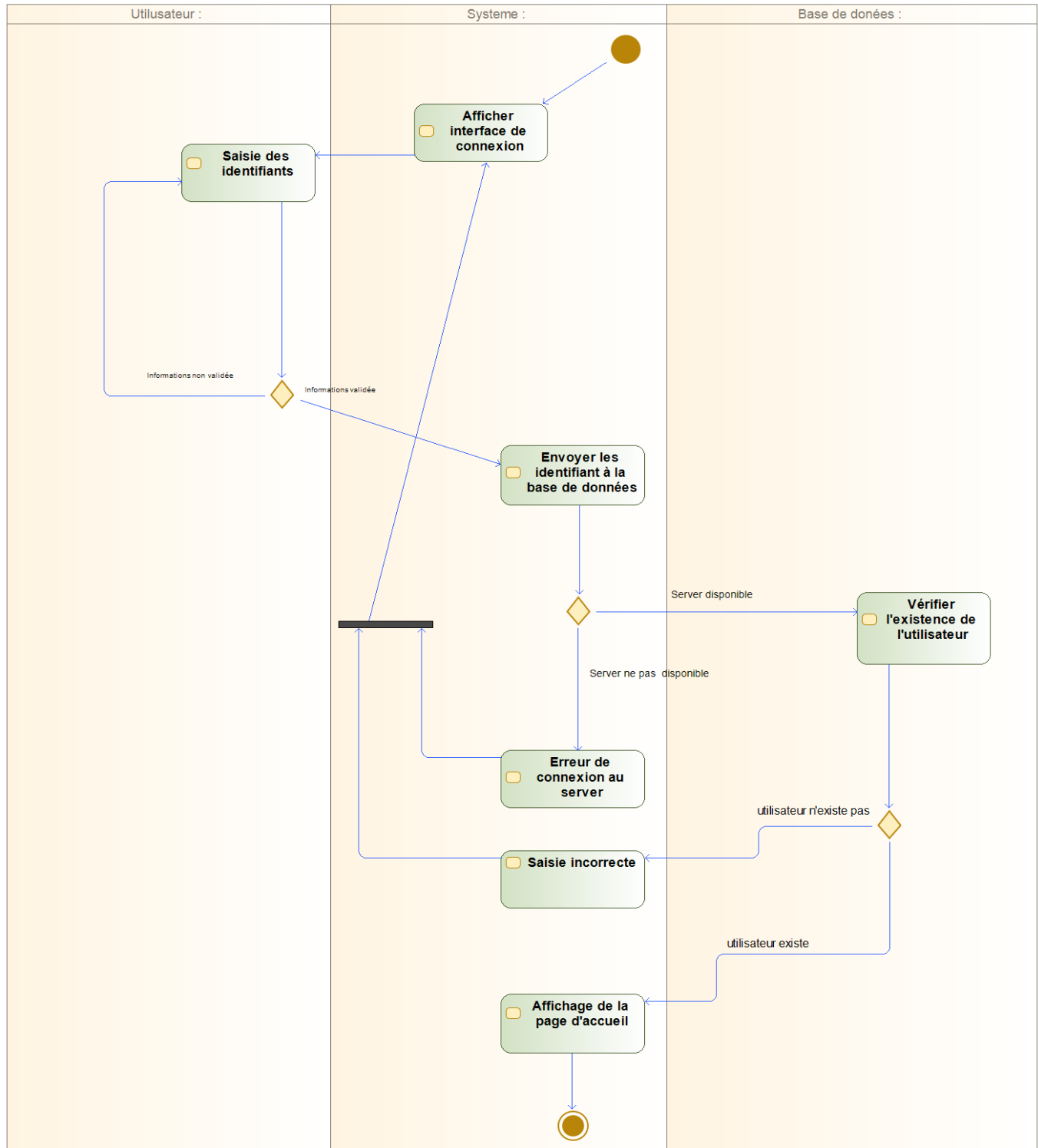


FIGURE 4.4 – Diagramme d'activité : Authentification

4.2 Modélisation de la base de données

4.2.1 Diagramme de classe

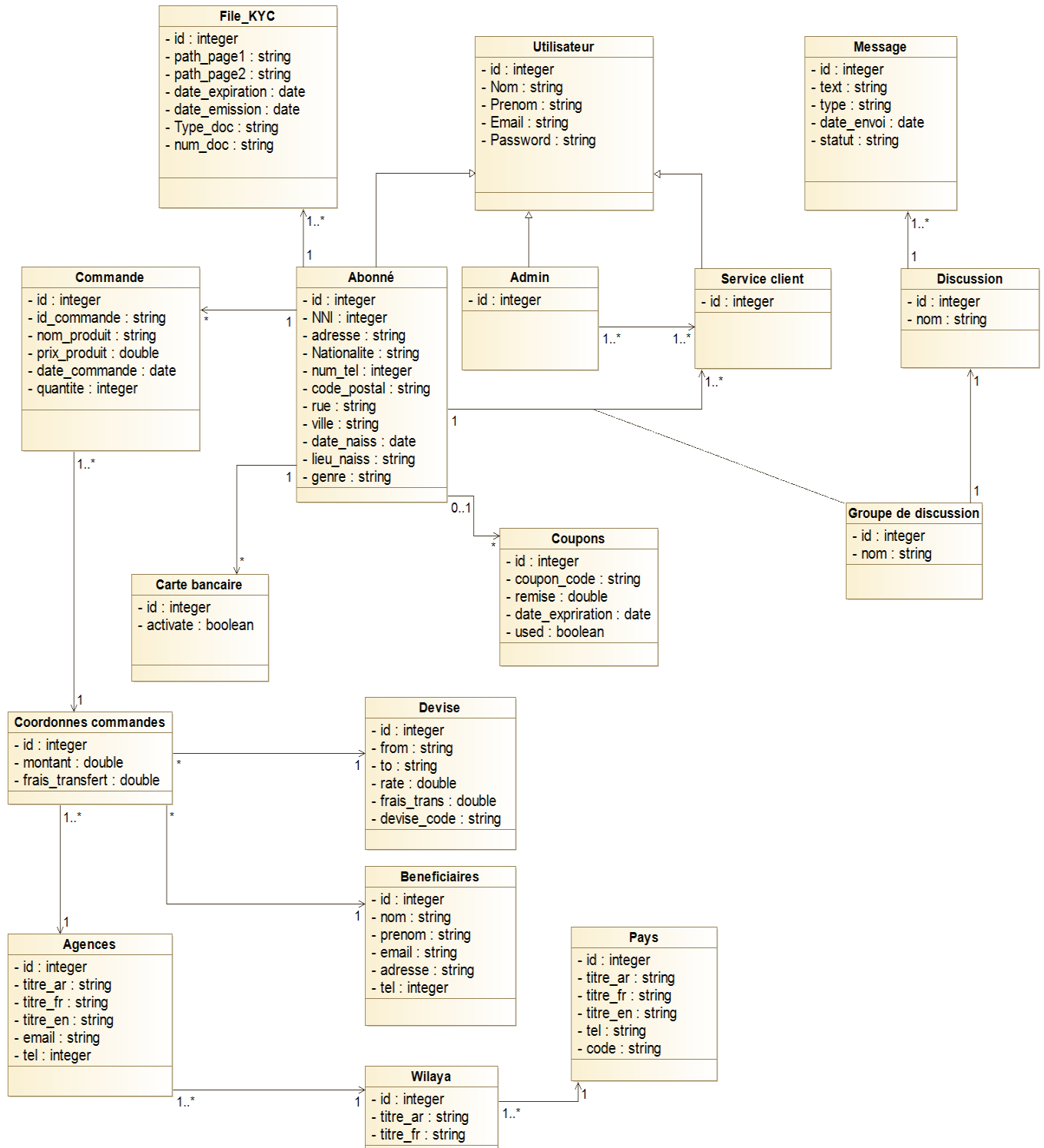


FIGURE 4.5 – Diagramme d'activité : Transfert d'argent

Chapitre 5

Les technologies utilisées

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 a nos besoins , et que nos interfaces soient claires et faciles à utiliser.

5.0.1 Front-end

C'est le développement coté client autrement dit la partie du code reçu par le client. On rappelle que le client désigne un navigateur web. On a choisi Flutter comme Framework pour développer la partie front de notre application mobile. 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.

5.0.1.1 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, veuillez visiter le lien : <https://en.wikipedia.org/wiki/VisualStudioCode>

5.0.1.2 Language : 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, veuillez visiter le lien : <https://www.tutorialspoint.com/flutter/dart>

5.0.1.3 Framework : 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 :

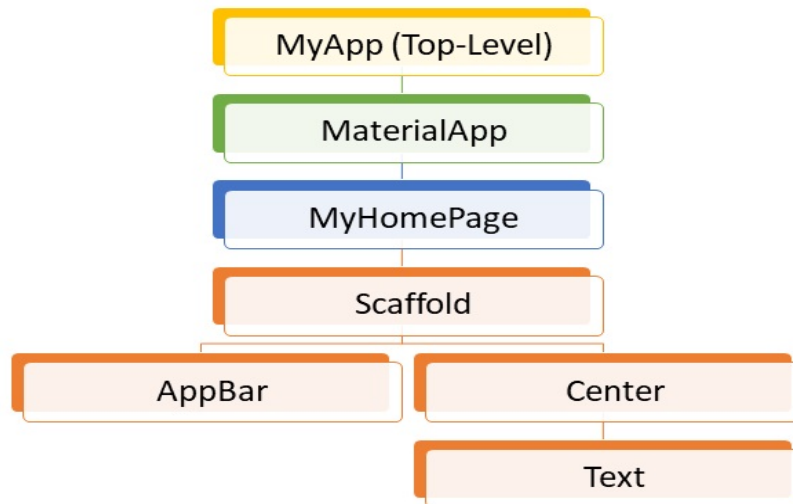


FIGURE 5.1 – Architecture d'une application Flutter.

5.0.1.4 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, veuillez visiter le lien : <https://en.wikipedia.org/wiki/AndroidStudio>.

5.0.2 Back-End

C'est le développement côté serveur c'est-à-dire la partie du code exécutée par le serveur.

5.0.2.1 Language : 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, veuillez visiter le lien : <https://fr.wikipedia.org/wiki/PHP>

5.0.2.2 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. Pour en savoir plus, veuillez visiter le lien : <https://aws.amazon.com/fr/what-is-aws>

5.0.2.3 Framework :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](#).

5.0.2.4 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.

5.0.3 Motifs d'architecture logicielle

5.0.3.1 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 a 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.

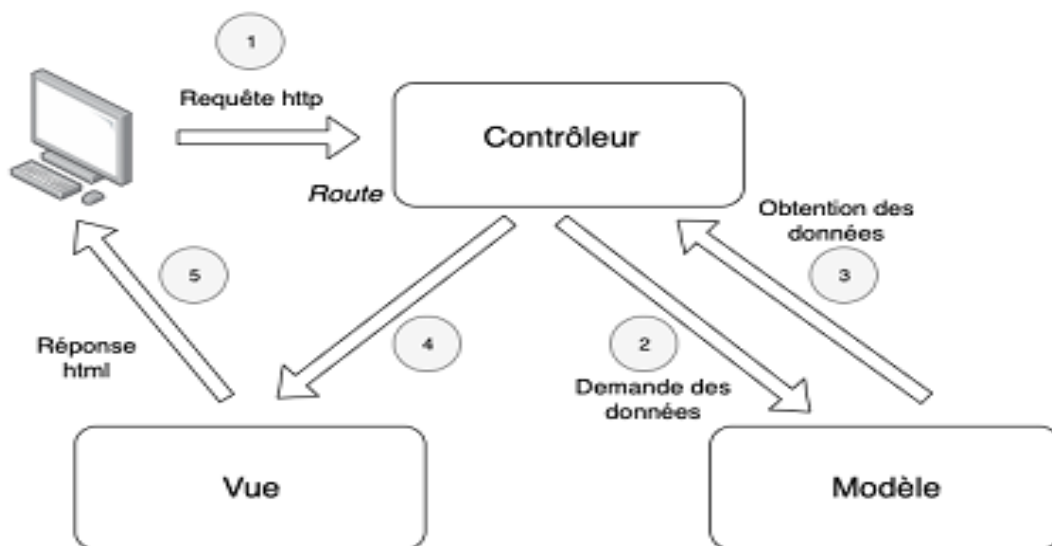


FIGURE 5.2 – MVC

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 définit 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 .

Ce figure ci-dessous résume en quelque sort le fonctionnement de notre application. L'api est considéré comme l'intermédiaire entre la partie données (base de données) et la partie présentation (Application mobile).

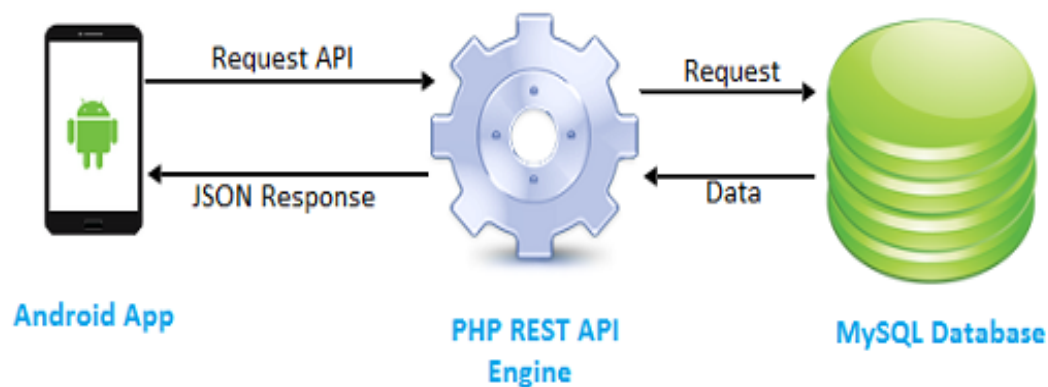


FIGURE 5.3 – PHP MySQL REST API pour Android

Chapitre 6

Implémentation

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 a nos besoins , et que nos interfaces soient claires et faciles à utiliser.

6.1 Architecture de l'application

Cadorim est une application embarquée qui se connect à un serveur de base de données distant, via Internet, afin de récupérer les données, Ce qui necessite aussi l'intégration d'un serveur web entre l'application client et le serveur de bases de données. D'où l'architecture de notre application est à 3 niveaux, elle est partagée entre ;

1. **L'application mobile (IOS ou Android) :** Ce le client qui demande les ressources.
2. **Le Serveur Web :** Vue que les données seront communiquées entre deux environnements hétérogènes, le rôle principale du serveur web est de gérer la communication entre le client (Android ou IOS) et le serveur de données.
3. **Le serveur de base de données :** fournis les données au serveur web.

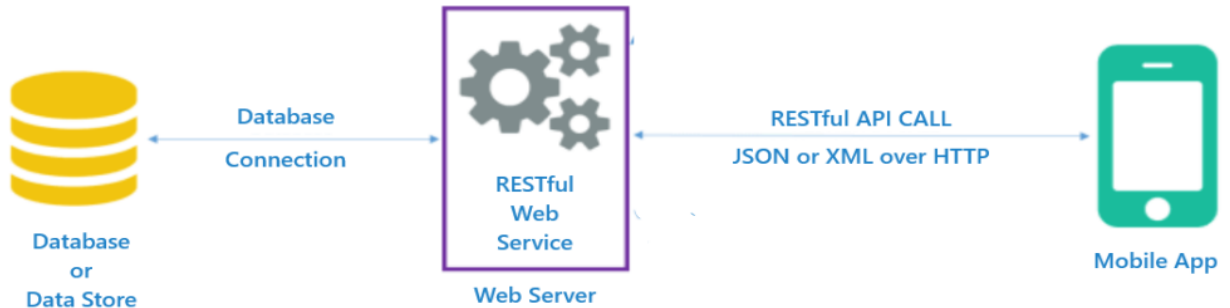


FIGURE 6.1 – Architecture logicielle de l’application.

6.2 Interfaces graphiques

Les interfaces utilisateur doivent respecter les heuristiques d’utilité pour permettre à l’utilisateur un accès facile à ces interfaces afin de garantir une bonne compréhension des fonctionnalités de l’application. Nous présentons ici les interfaces les plus significatives de l’application.

6.2.1 Interfaces d’accueil

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.



FIGURE 6.2 – Interfaces d'accueil

Chapitre 7

Interface et utilisation de l'application