

UNIVERSITÉ MOHAMMED V de Rabat

# Faculté des Sciences



## Département d'informatique

Filière Licence fondamentale  
En Sciences Mathématiques et Informatique

# PROJET DE FIN D'ETUDES

Intitulé :

Développement d'une application web de gestion d'une  
agence bancaire

Présenté par:

OUDER Samir

OUBRAHIM Hamza

Soutenu le 26 Juin 2019 devant le Jury :

Mr.SLAOUI Said Professeur à la Faculté des Sciences Encadrant

Mr.BENKHALIFA Mohammed Professeur à la Faculté des Sciences Examineur

Mr.REDA Oussama Professeur à la Faculté des Sciences Examineur

# Remerciements

C'est avec un grand plaisir qu'on réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à la réalisation et l'aboutissement de ce travail.

Tout d'abord on remercie notre encadrant Monsieur SLAOUI Said, pour son soutien, son sérieux, sa disponibilité, ses précieux conseils et son aide tout au long de l'élaboration de ce travail. Nous remercions les membres de jury qui ont bien voulu examiner et évaluer notre projet de fin d'étude.

Nous nous acquittons, enfin, volontiers d'un devoir de gratitude et de remerciements à tous nos enseignants pour la qualité de l'enseignement qu'ils ont bien voulu nous prodiguer durant nos études afin de nous fournir une formation efficiente.

# Résumé

Ce document présente la mission qui nous a été confiée pour notre projet de fin d'études effectué au sein de la Faculté des Sciences Rabat. Ce travail obligatoire pour l'obtention du Diplôme de la licence fondamentale en SMI.

L'objectif de ce projet de fin d'études porte sur la gestion des comptes, la gestion des opérations bancaires (virement, versement, retrait), généralement la gestion d'une agence bancaire.

Pour la réalisation de ce projet, nous avons utilisé plusieurs technologies telles que LARAVEL, PHP, BOOSTRAP, JAVASCRIPT. Par ailleurs, nous avons travaillé avec le système de gestion de base de données MySQL.

Mots Clés: LARAVEL, PHP, BOOSTRAP, JAVASCRIPT.

# Sommaire

Remerciements	2
Résumé	3
Introduction	9
1 Contexte Générale	10
1.1 Contexte et intérêt du projet	10
1.2 Structure et organisation administrative de l'établissement	10
1.2.1 La Faculté des Sciences de Rabat	10
1.2.2 Organisation générale	10
1.2.3 Départements	11
1.3 Licence Sciences Mathématiques et Informatique (SMI)	11
2 Présentation du Projet	12
2.1 Gestion Application Bancaire	12
2.2 Avantage d' AgenceBanq	13
2.3 Objectifs à atteindre	14
2.4 Cahier des charges	15
2.4.1 Fonctionnalités	15
2.4.2 Règles de gestion	15
2.5 Planning	16
2.6 Conclusion	16
3 Etude Conceptuelle	17
3.1 Introduction	17
3.2 Patron de conception	17
3.2.1 Définition	17
3.2.2 MVC (Modèle ~ Vue ~Contrôleur)	18
3.3 Modèle	19
3.4 La modélisation	19
3.4.1 Les avantages de la modélisation	20
3.4.2 L'outil	20
3.5 Diagramme de cas d'utilisation	21
3.5.1 scénario cas d'utilisation	21
3.6 Diagramme d'activités	24
3.7 Diagramme de classe	27
3.8 Conclusion	28
4 Environnement technique	29
4.1 Architecture du système et environnement de test	29
4.1.1 Pourquoi utiliser un Framework ?	29
4.2 Le Framework Laravel	29

4.2.1 Définition .....	29
4.2.2 Structure du répertoire .....	29
4.3 Les technologies utilisées .....	29
4.3.1 Laravel .....	30
4.3.2 Bootstrap .....	30
4.4 Les outils utilisés .....	31
4.4.1 MySQL .....	31
4.4.2 PhpMyAdmin .....	31
4.4.3 Modelio .....	31
4.4.4 XAMPP .....	31
4.4.5 Conclusion .....	31
<b>5 Réalisation</b> .....	<b>32</b>
5.1 L'architecture de l'application .....	32
5.2 L'implémentation de <<agenceBanq>> .....	32
5.2.1 Installation et organisation .....	33
5.2.2 Les routes .....	34
5.2.3 Les migrations .....	35
5.2.4 Classe Model .....	35
5.2.5 Contrôleur .....	35
5.2.6 Les vues. ....	36
5.3 Présentation des interfaces .....	37
5.3.1 Interface d'authentification .....	38
5.3.2 Interface d'accueil .....	38
5.3.3 Partie clients .....	39
5.3.3.3 Ajouter un client .....	40
5.3.4 Modifier un client .....	41
5.3.5 Supprimer un client .....	41
5.3.6 Gestion des comptes .....	42
5.3.7 L'ajout d'un nouveau compte .....	42
5.3.8 Choix de type de compte a créé .....	43
5.3.9 Création de compte courant .....	43
5.3.10 Création de compte courant avec succès .....	44
5.3.11 Création de compte épargne .....	44
5.3.12 Création de compte courant avec succès .....	45
5.3.13 Suppression de Compte.....	45
5.3.14 Recherche par client et supprimer .....	46
5.3.15 Opération sur compte .....	46
5.3.15.1 Versement .....	47
5.3.15.2 Relevé de compte .....	48
5.3.16.1 Retrait de compte .....	48
5.3.16.2 Vérification du retrait .....	50
5.3.17 Virement .....	50
5.3.17.1 Effectuer Virement .....	50
5.3.17.2 Relevé de compte X .....	51
5.3.17.3 Relevé de compte Y .....	51
5.4 Conclusion .....	53

# List of Figures

3.1 Architecture MVC .....	9
3.2 Diagramme de cas d'utilisation .....	10
3.3 Diagramme d'activités: Création Compte .....	13
3.4 Diagramme d'activités : Retrait .....	14
3.5 Diagramme d'activités : Versement .....	14
3.6 Diagramme d'activités : Virement .....	15
4.1 Laravel .....	19
5.1 MVC .....	21
5.2 Dossier d'installation Laravel .....	22
5.3 Création du Framework Laravel .....	22
5.4 Les routes .....	22
5.5 Les migrations .....	23
5.6 Model .....	24
5.7 Controller .....	24
5.8 Vues .....	25
5.9 Interface d'authentification. ....	26
5.10 Interface d'accueil. ....	26
5.11 Accéder aux opérations sur clients. ....	27
5.12 Nouveau Client.....	27
5.13 Ajout d'un client. ....	28
5.14 Redirection Liste des clients.....	28
5.15 Pour modifier un client. ....	29
5.16 Pour supprimer un client. ....	29
5.17 accéder aux Guide des comptes.....	30
5.18 ajouter un nouveau compte. ....	30
5.19 Choix de type de compte à créer. ....	31
5.20 Création du compte courant.....	31
5.21 Après création compte courant. ....	32
5.22 Création du compte épargne.....	32
5.23 Après création compte épargne. ....	33
5.24 Supprimer un compte. ....	33
5.25 Supprimer un compte par recherche .....	34
5.26 Opérations sur les comptes .....	34
5.27 Versement.....	35
5.28 Relevé. ....	36
5.29 Consulter relevé .....	36
5.30 Consulter relevé .....	36
5.31 Retrait .....	37
5.32 Interface retrait .....	37
5.33 Vérification retrait. ....	38
5.34 Virement. ....	38

5.35 Effectuer virement. ....	38
5.36 Relevé d compte. ....	39
5.37 Relevé d compte. ....	39
5.38 Virement compte courant vers compte épargne.....	40
5.39 Relevé d compte. ....	40
5.40 relevé de compte. ....	41

# Liste des tables

2.1 le planning prévu pour l'exécution du projet .....	7
--	---



# Introduction

Dans un monde de plus en plus moderne, disposer d'un compte bancaire est nécessaire pour sécuriser son argent, percevoir un salaire, une allocation ou faire des achats, payer son loyer ou ses factures, dans ce contexte, notre application Gestion Agence Bancaire permet de gérer l'ensemble des clients et leurs comptes bancaires.

L'application est destinée à l'employé d'une agence bancaire. Un employé d'une agence peut accéder à l'ensemble des clients, comptes et transactions et effectuer toutes les opérations de l'application.

En effet, avec un simple clic l'employé peut faire toutes les opérations d'une manière sécurisée. C'est pourquoi dans le cadre des objectifs assignés au PFE encadré par Mr SLAOUI Said nous avons opté pour le développement d'une application Web bancaire offrant les fonctionnalités suivantes :

1. Création de compte
2. Modification de compte
3. Retrait / dépôt
4. Virement

En effet, un étudiant d'informatique est concerné par la connaissance des notions de base qui lui permettent d'enrichir son vocabulaire et par la suite d'être capable d'exploiter ses acquis dans la vie professionnelle. Les points forts de ce projet, c'est qu'il combine entre le côté théorique et sa projection en réalité (côté pratique).

Le but auquel nous aspirons dans ce projet c'est de bien comprendre le côté théorique via une recherche, poser un modèle qui sera capable de faire la structuration d'un site web dès le départ, faire une interface graphique pour simuler la gestion, et enfin faire la liaison entre le Modèle (model), Interface graphique (vue) et le Traitement (contrôler).

La suite du document est organisée en 5 chapitres : Le premier chapitre présente le contexte général du projet, le deuxième chapitre présente les besoins et les spécifications techniques et fonctionnelles dont une présentation des différents diagrammes, la troisième partie est consacrée à l'étude conceptuelle de l'application aux perspectives des travaux réalisés, et le quatrième chapitre présente l'architecture et l'environnement technique, et finalement la partie démonstration.

# Chapitre 1

## Contexte Générale

### 1.1 Contexte et intérêt du projet

Le projet de fin d'études (PFE) est un travail à la charnière entre la formation académique et l'entrée dans la vie active. Il constitue un complément de formation pratique obligatoire aux apprenants (étudiant, élève ingénieur, élève technicien etc...). C'est un travail à caractère professionnel, sous forme d'un travail d'ingénierie encadré par au moins un enseignant universitaire ou un enseignant professionnel. Le PFE a pour principaux objectifs de finaliser la formation scientifique et technique par la mise en pratique des compétences acquises, de se confronter à des situations professionnelles proches de celles que rencontre un jeune ingénieur diplômé lors de l'exercice de ses fonctions. Enfin le PFE lui permet d'approfondir ses connaissances en la matière et d'acquérir une première expérience professionnelle et sociale pour faciliter sa future insertion professionnelle.

### 1.2 Structure et organisation administrative de L'établissement

#### 1.2.1 La Faculté des Sciences de Rabat

Créée au sein de l'Université Mohammed V par le dahir numéro 1-58-390 du 15 Muharram 1379 (21 Juillet 1959), la faculté des Sciences de Rabat est un établissement d'enseignement supérieur et de recherche qui a longtemps assuré, dans les spécialités de mathématique, informatique, Physique, Chimie, Biologie, les formations de Licence, Certificat d'études Approfondies (CEA), Diplôme d'Etudes Supérieures de 3ème cycle (DES) et Doctorat d'état en sciences. L'effort déployé dans le domaine de la formation et de la recherche a permis de marocaniser rapidement le corps enseignant de cet établissement et de pourvoir en enseignants chercheurs toutes les institutions scientifiques créées au Maroc, au cours des dernières années.

#### 1.2.2 Organisation générale

L'équipe de la faculté des Sciences de Rabat est constituée de :

- Doyen;
- Vice doyen;
- Services ;
- Annexes.

### 1.2.3 Départements

En vertu des fonctions confiées aux professeurs chacun selon ses attributions, et dans l'objectif de subvenir aux multiples besoins des étudiants dans les meilleures conditions possibles, le corps professoral de la Faculté des sciences de Rabat est réparti sur les départements suivants:

- Département de chimie ;
- Département de Mathématiques;
- Département de Physique;
- Département des Sciences de la Terre et de l'Univers ;
- Département Informatique;

## 1.3 Licence Sciences Mathématiques et Informatique (SMI)

La Licence Sciences Mathématiques et Informatique offre une formation de base Bac +3 en Mathématiques fondamentales, Mathématiques appliquées, ainsi qu'en Informatique. Elle a pour but de donner aux étudiants à la fois:

- Une formation de base, en mathématiques et en informatique,
- Une maîtrise des principaux outils de calcul et de modélisation, ainsi que des outils informatiques leur permettant de mettre en œuvre cette formation théorique dans diverses applications.
- Un esprit de synthèse, de rigueur et d'initiative afin de préparer le lauréat à intégrer avec succès sa vie professionnelle.

# Chapitre 2

## Présentation du Projet

Dans ce chapitre, nous spécifions en détail les besoins du cahier de charge. D'abord, nous présentons toutes les fonctionnalités de l'application Gestion Agence Bancaire, ensuite nous introduisons les règles de gestion, puis nous présentons le planning adopté dans le déroulement de notre projet.

### 2.1 Application de Gestion Agence Bancaire

“ AgenceBanq ” est une application web qui fait la gestion d'une agence bancaire, qui facilite la tâche aux employés d'une agence bancaire d'une part, et facilite les opérations pour le client d'une autre part.

“ AgenceBanq ” est un système qui supporte la gestion automatique des clients et leurs comptes. Les différentes fonctionnalités de l'application sont données ci-après.

Les services disponibles de notre Application sont:

1. Les opérations sur le comptes bancaire (consulter, créditer, débiter, supprimer).
2. Virement d'un compte à un autre compte.
3. La sécurité totale des comptes.
4. Les employés sont responsables de l'inscription des clients (d'ouvrir les comptes courants et épargnes)

Nous serons amenés plus tard à développer l'application web avec une interface graphique (objectif principal du projet) en se basant sur une le Framework du PHP LARAVEL.

## 2.2 Les avantages d' « AgenceBanq »

Nous sommes convaincus que la gestion bancaire doit être simple, accessible. Nous nous chargeons de la partie fastidieuse du travail pour que l'employé se concentre sur son propre métier.

« AgenceBanq » est développée afin que les agences bancaires puissent disposer d'une solution flexible, ouverte, automatisée, sécurisée, complète et qui répond à leurs exigences. C'est pour ça nous avons choisi le Framework LARAVEL parmi les avantages de ce dernier :

- un développement plus rapide

Aujourd'hui, toutes les entreprises cherchent à accélérer leur transformation numérique et c'est la raison pour laquelle il est nécessaire de choisir le bon Framework pour votre application Web. Le choix du Framework Laravel accélère votre développement d'applications Web et vous permet donc de le faire vivre beaucoup plus rapidement que si vous utilisiez d'autres Framework.

- Sécurité top-notch

Chaque année, de nombreuses cyber-attaques sont perpétrées. Dans de telles situations, vous avez besoin de Framework qui protègent vos applications Web. L'un des avantages les plus importants du choix de Laravel pour le développement de vos applications Web réside dans ses capacités à fournir une sécurité de haut niveau. Laravel est lui-même un Framework sécurisé qui ne permet à aucune application de logiciel malveillant ni de menace à la sécurité d'entrer dans l'application Web. Cela signifie que le code de développement de vos applications Web est sécurisé. Si vous choisissez Laravel, votre application Web ne présente aucun risque d'injections SQL involontaires et cachées.

- Performances améliorées

Une autre raison forte de choisir Laravel est sa capacité à fournir une excellente performance des applications Web. Il arrive que certaines caractéristiques et fonctionnalités affectent les performances du site. Mais Laravel propose divers outils qui aident les développeurs à améliorer les performances des applications Web. Les outils tels que Memcached et Redis doivent être intégrés au Framework de Laravel lors du développement des applications Web. Ils facilitent la tâche des développeurs pour améliorer les performances des applications Web. Cela signifie qu'avec Laravel, vous disposez de l'application Web la plus performante.

- Grande communauté et beaucoup de documentation

Un autre avantage pour Laravel est qu'il est open-source. Laravel possède une puissante communauté de développeurs et de sociétés de développement qui fournit en permanence une assistance pour la rendre plus flexible et évolutive. Ainsi, si votre développeur souhaite apporter des fonctionnalités complexes, il/elle est toujours ouvert (e) à suivre les conseils avisés de la communauté et à adapter votre application à vos besoins. En bref, le développement de votre application ne s'arrête pas malgré de telles fonctionnalités complexes et vous obtenez ce que vous désirez.

- Tests unitaires

Une autre raison pour laquelle vous devriez envisager de choisir le Framework Laravel pour votre développement Web est qu'il offre des tests unitaires faciles. Qu'est-ce qu'un test unitaire ? Le test unitaire est un type de test dans lequel chaque module ou composant de votre application Web est testé afin qu'aucune partie de votre site Web ne soit brisée. Les tests unitaires garantissent qu'il n'y a pas de bogues ou d'exceptions dans votre application Web avant sa mise en production. Cela signifie que vous disposez d'une application performante et sans bugs pour vos utilisateurs finaux.

## 2.3 Objectifs à atteindre

La banque doit pouvoir atteindre plusieurs objectifs dans le cadre de la gestion quotidienne des comptes bancaires. Ces objectifs sont complémentaires. Il s'agit de :

1. Maîtriser les risques associés aux opérations bancaires réalisées sur le compte.
2. Améliorer la qualité des services rendus aux clients.
3. Rendre les opérations sur les comptes plus rapides.
4. Un accès simple et rapide aux données.

## 2.4 Cahier des charges

Afin d'expliciter les besoins de cette application et pour répondre au cahier de charges, nous allons détailler les différentes fonctionnalités nécessaires. Ainsi que les règles de gestion de certaines fonctionnalités.

### 2.4.1 Fonctionnalités

Les fonctionnalités de l'application permettent à un employé de :

1. S'authentifier.
2. Créer un compte courant (sans intérêt). L'ouverture d'un compte courant donne lieu à la livraison d'un carnet de chèques au client. Un chèque peut être assimilé de la manière suivante : il s'agit d'un ordre donné par une personne X (Appelée tireur) à sa banque (Appelé tirée) de verser une somme d'argent à une seconde personne Y (appelée bénéficiaire)
3. Crée un compte épargne (sans intérêt). L'ouverture d'un compte sur carnet donne lieu à la livraison d'un carnet sur lequel sont portées toutes les opérations de versement et de retrait. Dans le cadre de cette application on fera la distinction entre les deux comptes à la base de l'attribut "type de compte". Un client peut ouvrir plusieurs comptes sur carnet.
4. Gérer les opérations quotidiennes (Retrait, Versement, Virement).

### 2.4.2 Règles de gestion

Pour rendre notre application plus logique, différentes règles de gestion sont nécessaires :

1. L'utilisateur doit savoir les références du tireur bénéficiaire ou un tiers
2. L'utilisateur doit savoir les références d'une banque
3. L'utilisateur doit savoir les références de cheque hors place
4. Un client de l'agence peut ouvrir un compte épargne (carnet) pour lui-même et un compte épargne (carnet) pour chacun de ces enfants
5. Un client ne peut ouvrir qu'un seul compte courant

## 2.5 Planning

Notre travail a débuté par une étude théorique afin de mieux comprendre les différents aspects fonctionnels du projet. Cette phase a été suivie d'une analyse et spécification des besoins fonctionnels de l'application avant d'aborder la phase de conception pour passer finalement à la réalisation. La planification est un outil indispensable de management de projet. Planifier optimise les chances de réussite du projet grâce à une meilleure maîtrise du plan de travail. Le tableau suivant présente le planning prévu pour l'exécution du projet :

N.de tâche	Tâche	Début	Fin	Durée
1	Etude préliminaire	01/03/2019	10/03/2019	10 jours
2	Analyse et Conception (Laravel)	11/03/2019	23/03/2019	13 jours
3	Etude Technique (Laravel)	24/03/2019	02/04/2019	10 jours
4	Analyse et Conception (Laravel)	03/04/2019	22/04/2019	20 jours
5	Etude Tech	22/04/2019	21/05/2019	30 jours
6	Tests finaux	22/05/2019	06/05/2019	16 jours

Table 2.1: le planning prévu pour l'exécution du projet

## 2.6 Conclusion

Dans ce chapitre, nous avons détaillé les fonctionnalités et les règles de gestion de notre projet, puis nous avons présenté le planning prévisionnel du déroulement du projet. Dans le chapitre suivant, nous allons aborder en détail l'étude conceptuelle du projet.



# Chapitre 3

## Étude Conceptuelle

La phase de la conception est une phase qui initialise la création et la mise en œuvre de notre projet. En fait, elle représente une étape importante de la réflexion dans le cycle de développement logiciel de l'application après la phase de l'analyse et de spécification. Dans ce chapitre, nous allons présenter en détails la conception du notre projet à travers une modélisation des diagrammes UML (diagramme de cas d'utilisation, diagramme d'activité, diagramme de classe, diagramme de séquence).

### 3.1 Introduction

Dans ce chapitre nous avons traité principalement tous ce qui concerne la modélisation, comprendre c'est quoi la modélisation, son utilité, ses avantages et ses inconvénients.

### 3.2 Patron de conception

#### 3.2.1 Définition

Un patron de conception est une solution au problème récurrent dans la conception d'applications orientées objets. Il décrit des solutions standards pour répondre à des problèmes d'architecture et de conception des logiciels. On peut considérer un patron de conception comme une formalisation de bonnes pratiques. Pour notre application, nous avons un patron de conception <<MVC >>.

#### 3.2.2 MVC (Modèle – Vue –Contrôleur)

Le modèle-Vue-Contrôleur permet d'organiser l'interface homme-machine d'une application logicielle. Ce patron de conception impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur. Le modèle représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité, tout en offrant des méthodes permettant de les mettre à jour et de les récupérer en cas de besoin. La vue correspond à l'interface permettant d'agir avec les utilisateurs de l'application, d'où son rôle d'assurer la communication avec l'utilisateur tout en recevant ses action (clic de souris, sélection d'une entrée...) et en présentant les résultats envoyés par les traitements effectués par le modèle. Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle. Il reçoit tous les événements de l'utilisateur, interprété les requêtes correspondantes, et enclenche les actions à effectuer.

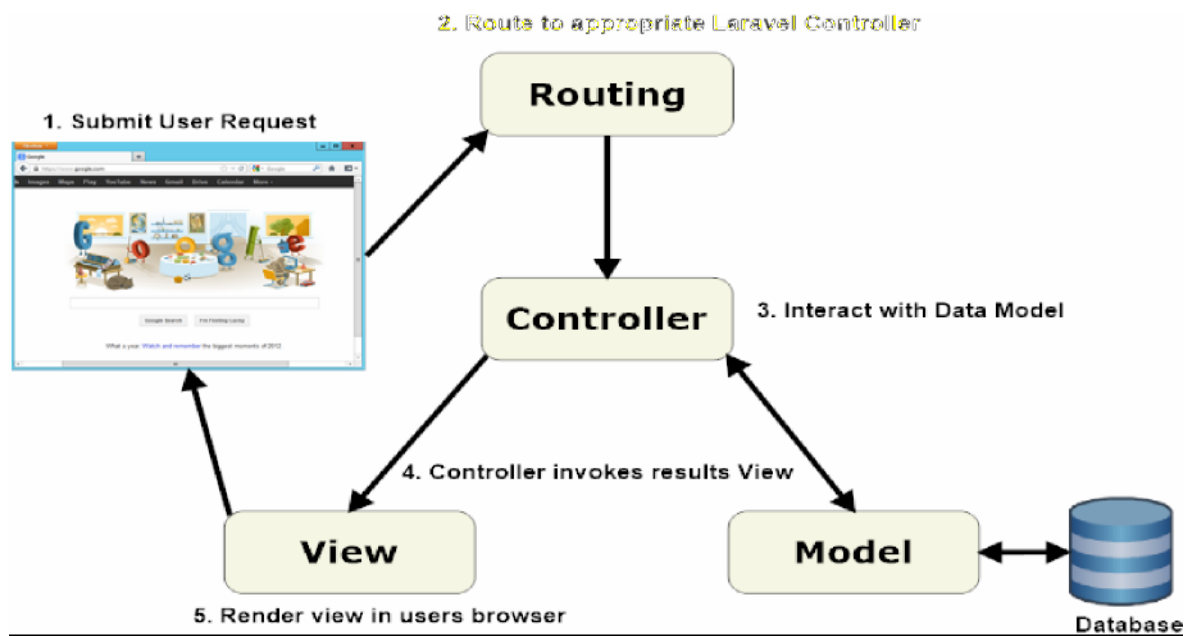


Figure 3.1: Architecture MVC

Une requête utilisateur est interprétée par le contrôleur, qui détermine quelles portions du modèle et de la vue doivent être appelées. Le modèle gère les interactions avec les données et applique les règles métiers, puis renvoie les données. Enfin, le contrôleur sélectionne une vue et lui passe les données. L'avantage majeur apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. L'exemple classique est représenté par un changement du type de la base de données sans affecter les vues, il suffira de mettre à jour les traitements relatifs d'interaction avec la base. Le MVC très pratique, peut se révéler lourd à mettre en place. Ceci à cause de la multitude de contrôleurs à implémenter. Afin de simplifier la réalisation d'un tel modèle, une nouvelle version a été introduite le MVC2. C'est exactement le même modèle de conception à la différence qu'il n'y a plus qu'un seul contrôleur qui se charge de rediriger, par le biais d'un fichier XML, la requête vers le bon traitement.

### 3.3 Modèle

Un modèle est une abstraction d'objets de la réalité. C'est donc une simplification du monde réel. La problématique consiste alors à trouver le bon niveau d'abstraction et à exprimer les concepts du monde réel dans un langage assez abstrait mais aussi précis qu'un langage de programmation, pour que ce langage soit interprétable par un programme informatique. En d'autres termes : << En informatique, un modèle a pour objectif de structurer les données, les traitements, et les flux d'informations entre entités. >>

### 3.4 La modélisation

Modéliser consiste à décrire dans un langage graphique formel, des besoins, les solutions fonctionnelles et techniques qui leur correspondent afin de bénéficier des atouts évidents qu'apporte un bon modèle, il est nécessaire que les personnes qui les réalisent et que celles qui les utilisent soient convenablement formées et accompagnées et travaillent selon le même cadre méthodologique.

### 3.4.1 Les avantages de la modélisation

Ceci nous ramène à poser la question suivante quels sont les avantages de la modélisation ?

Un bon modèle permet :

- Lors du développement : de simplifier la communication entre les intervenants d'un projet.
- Lors de la maintenance : de bénéficier d'une source d'information garantissant une analyse d'impact rapide et une meilleure qualité de réalisation.
- Abstrait : Il fait ressortir les points importants tout en enlevant les détails non nécessaires.
- Compréhensible : Il permet d'exprimer une chose complexe dans une forme plus facile et plus compréhensible par l'observateur.
- Précis : Il représente fidèlement le système modélisé.
- Prédicatif : Il permet de faire des prévisions correctes sur le système modélisé.

### 3.4.2 L'outil

On trouve des différents outils utilisés dans L'ingénierie diriger par model on va utiliser dans notre projet UML.

#### 3.4.2.1 Langage UML

L'UML est un langage de modélisation fondé sur les concepts objet: l'objet d'UML est de fournir une notation standard utilisable dans le développement de systèmes informatiques basés sur l'objet. Cependant, UML n'est pas une méthode car il n'inclut pas la manière d'utiliser les concepts qu'il se propose c'est à dire l'enchaînement des étapes qui mènent à la résolution des problèmes posés.

## 3.5 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les services et les fonctionnalités les plus importants de l'application. Ce diagramme clarifie seulement les cas d'utilisation principale

De l'application web. C'est vrai que le diagramme du Cas d'utilisation nous a servi de bien comprendre les grandes fonctionnalités de l'application, mais il manque d'avoir la force de construire une vision générale sur l'application (Interface graphique), l'ordre temporel du fonctionnement, ceci va être expliqué avec un diagramme d'activités.

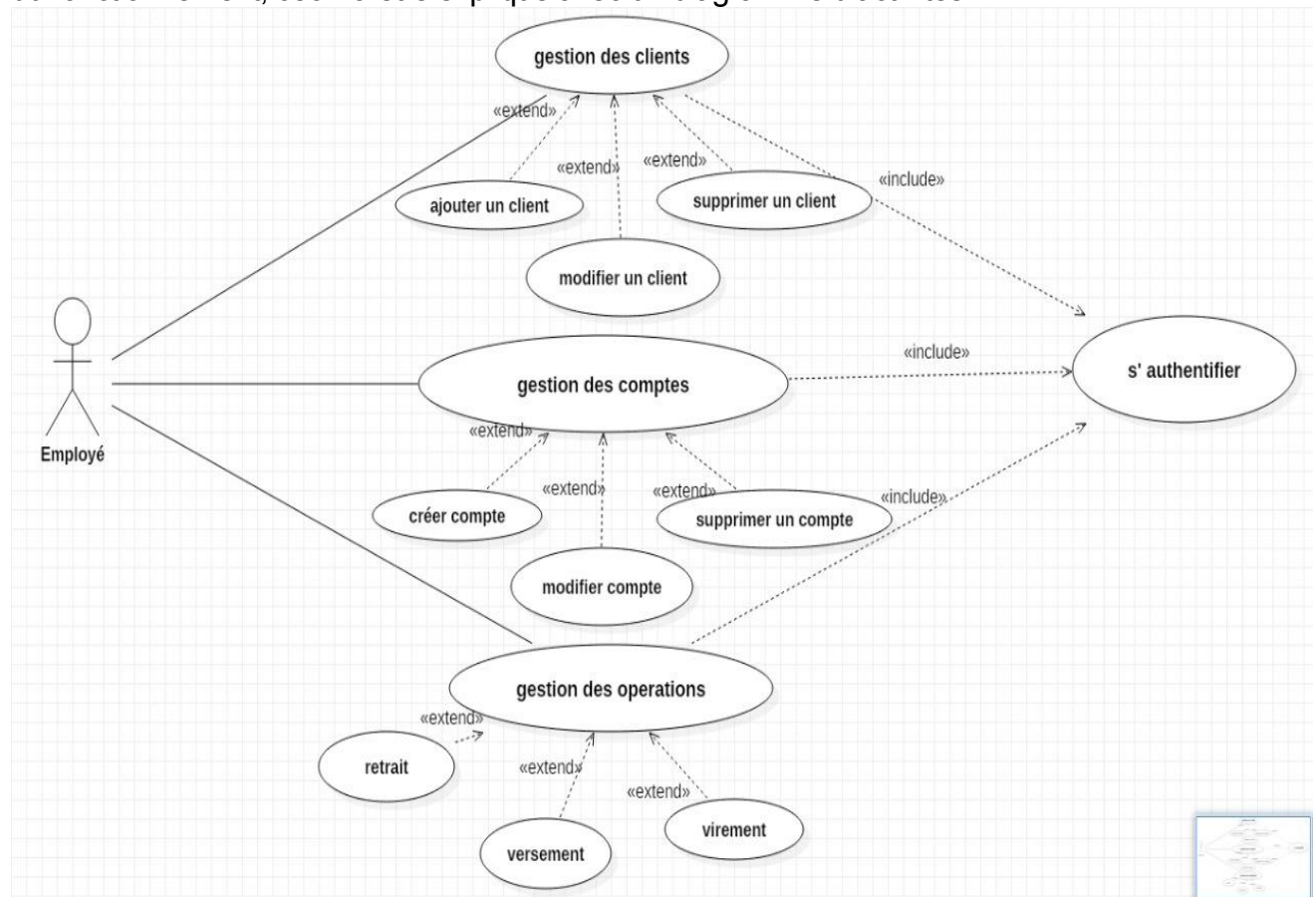


Figure 3.2: Diagramme de cas d'utilisation

### 3.5.1 Les scénarios des cas d'utilisation

#### 3.5.1.1 Scenario du cas d'utilisation <authentification>

1. L'utilisateur s'authentifier
2. Le système vérifie son identité
3. Le système affiche la page d'accueil

#### 3.5.1.2 Scenario du cas d'utilisation < gestion de comptes>.

1. L'employé s'authentifie
  - (a) Le système vérifie l'identité de L'employé
  - (b) Le système affiche la page d'accueil
2. L'employé demande la page de création de compte

- (a) Le système affiche la page de création de compte
- 3. L'employé choisie le type de compte qui veut créer et fait une recherche concernant le client, puis l'employé remplit les informations du nouveau compte.
  - (a) Le système vérifie des données saisies

#### . La Modification ?

- 1. L'employé s'authentifie
  - (a) Le système vérifie l'identité de L'employé
  - (b) Le système affiche la page d'accueil
- 2. L'employé recherche le compte à modifier
  - (a) Le système affiche la page de modification concernant le compte
- 3. L'employé modifie les champs souhaités

#### 3.5.1.3 Scenario du cas d'utilisation < versement >

- 1. L'employé s'authentifie
  - (a) Le système vérifie l'identité de L'employé
  - (b) Le système affiche la page d'accueil
- 2. L'employé identifie le compte pour effectuer un versement
  - (a) Le système vérifie l'identité du compte
  - (b) Le système affiche le compte
- 3. L'employé renseigne le montant a déposer
  - (a) Le système confirme le dépôt

#### 3.5.1.4 Scenario du cas d'utilisation < retrait >

- 1. L'employé s'authentifie
  - (a) Le système vérifie l'identité de L'employé
  - (b) Le système affiche la page d'accueil
- 2. L'employé identifie le compte pour effectuer un retrait
  - (a) Le système vérifie l'identité du compte
  - (b) Le système affiche l'état du compte
- 3. L'employé renseigne le montant à retirer
  - (a) Le système vérifie si le montant est disponible

#### 3.5.1.5 Scenario du cas d'utilisation < virement >

1. L'employé s'authentifie
  - (a) Le système vérifie l'identité de L'employé
  - (b) Le système affiche la page d'accueil
2. L'utilisateur identifie le compte émetteur
  - (a) Le système vérifie l'identité du compte émetteur
3. L'employé identifie le compte destinataire
  - (a) Le système affiche le compte émetteur
  - (b) Le système vérifie l'identité du compte destinataire
4. L'utilisateur renseigne le montant à virer
  - (a) Le système Affiche le compte destinataire
  - (b) Le système confirme le virement

#### 3.5.1.6 Scenario du cas d'utilisation< encaissement >

1. L'employé s'authentifie
    - (a) Le système vérifie l'identité de L'employé
    - (b) Le système affiche la page d'accueil
  2. L'employé vérifie le compte bénéficiaire s il existe sinon il crée un nouveau 3.
- L'employé identifie le compte tireur et vérifie si le montant est disponible
- (a) Le system affiche le compte
  - (b) L'employé mis le chèque en mis encaissement pour la vérification
  - (c) Le système confirme l'encaissement

## 3.6 Diagramme d'activités

Les diagrammes d'activités présentent plusieurs avantages pour n'importe quel type d'organisation. En effet modéliser des éléments de l'architecture des logiciels, tels que la méthode, la fonction et l'utilisation. Aussi former une idée préliminaire sur l'interface graphique de l'application.

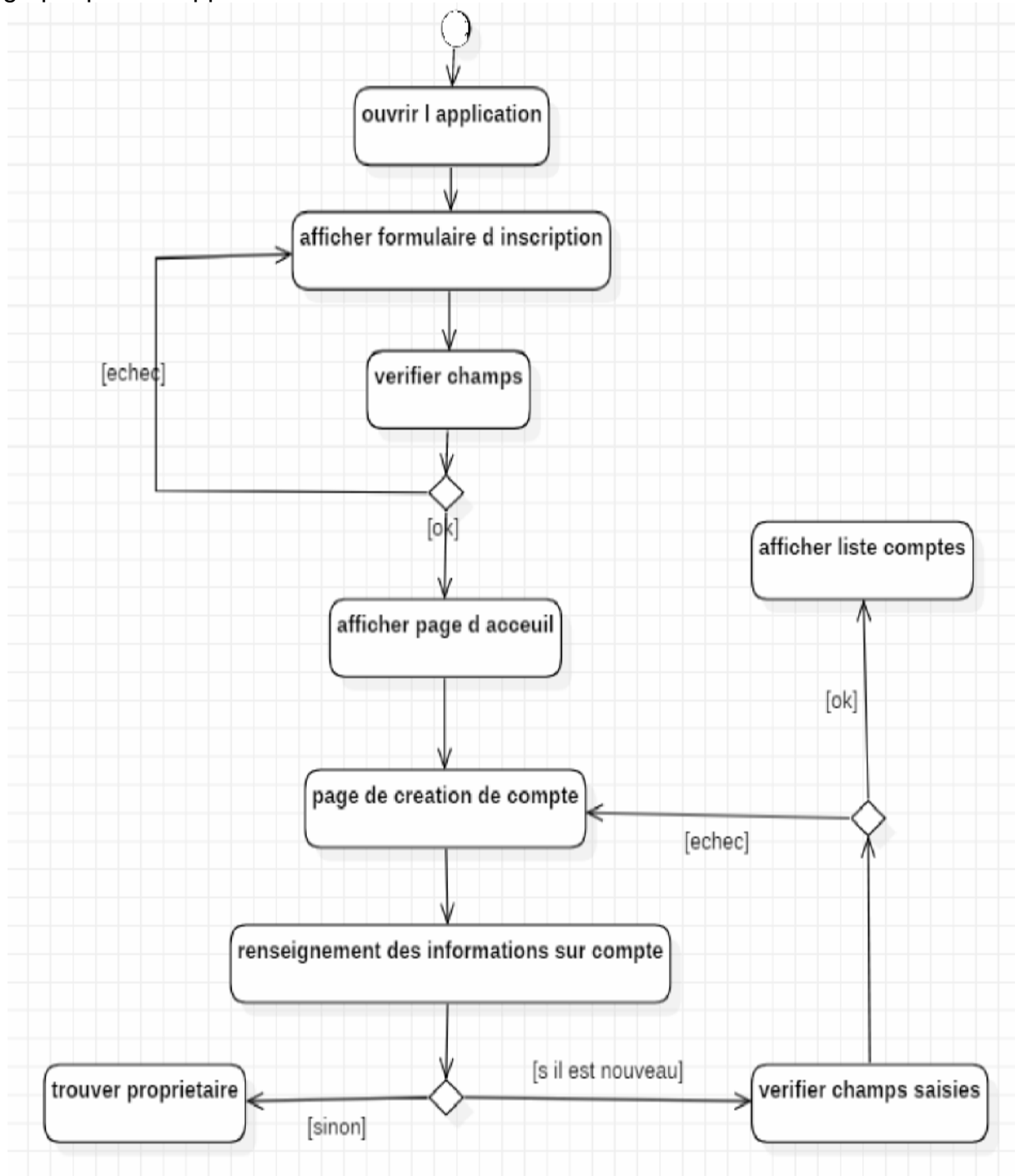


Figure 3.3: Diagramme d'activités : Création Compte



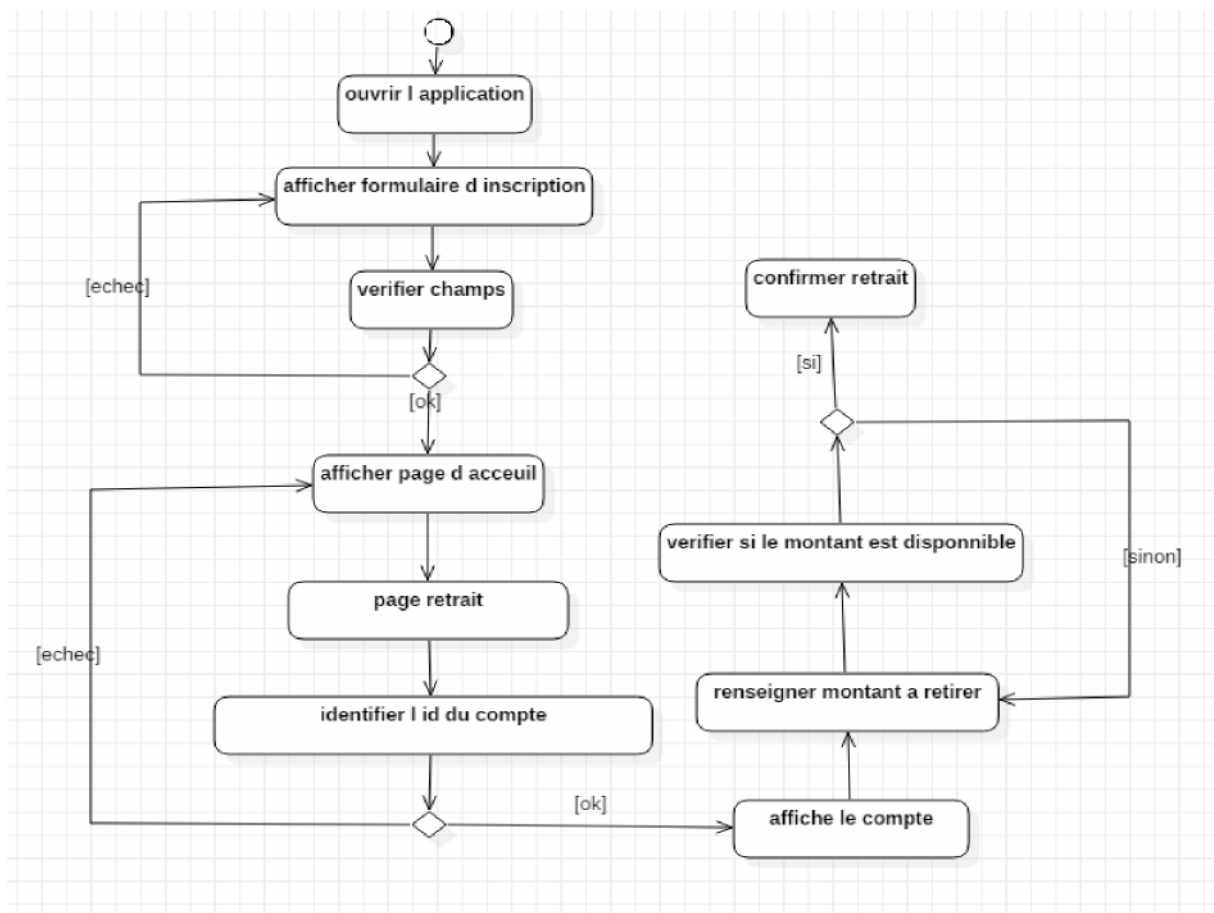


Figure 3.4: Diagramme d'activités : Retrait

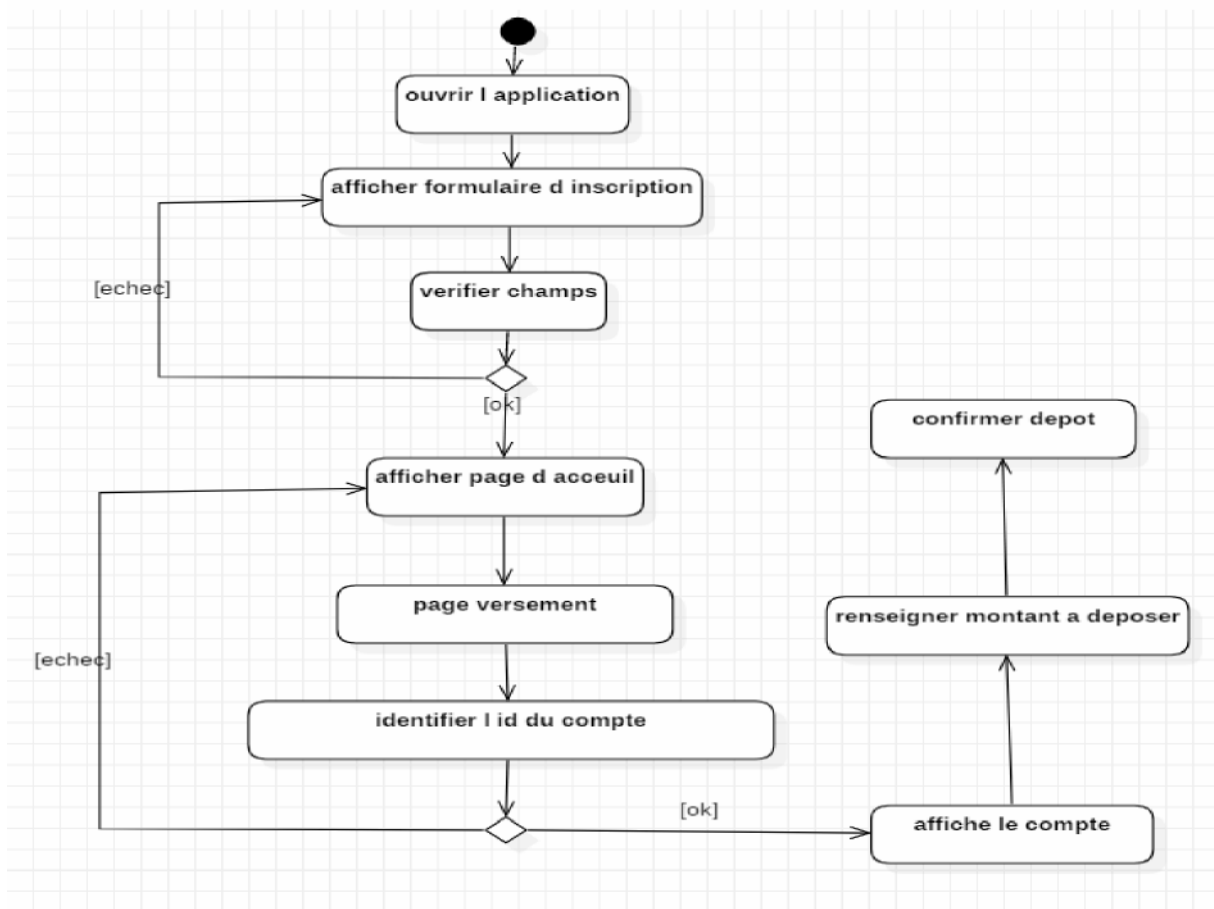


Figure 3.5: Diagramme d'activités : Versement

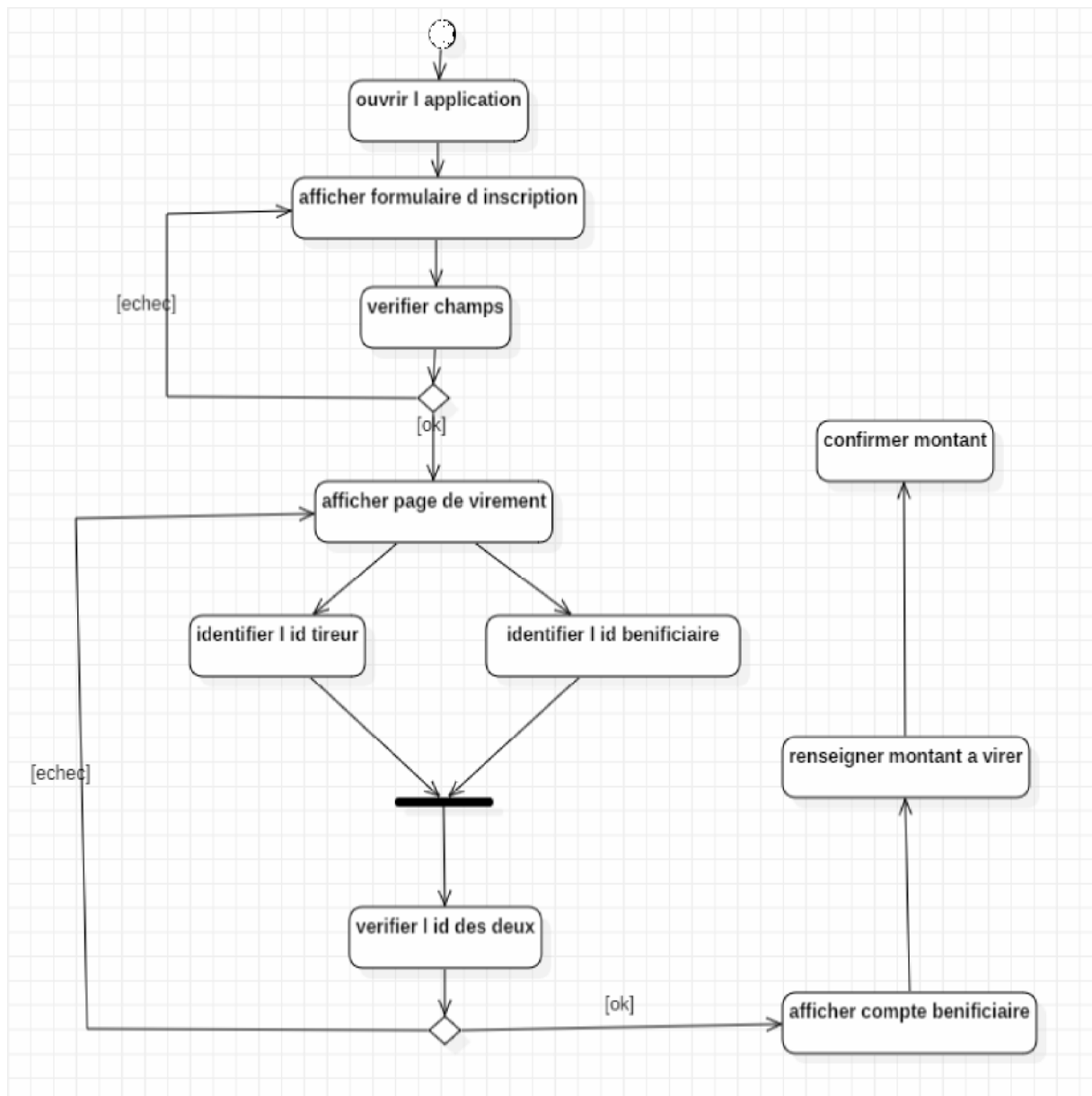


Figure 3.6: Diagramme d'activités : Virement

On peut dire que ce diagramme fait le même travail qu'un chronogramme, avec une séparation des activités faites par l'utilisateur et les autres par le système.

## 3.7 Diagramme de classes

### 3.7.0.1 Représentation des classes

La modélisation objet est utilisée dans le langage UML pour définir les objets-métiers et l'architecture de l'application. Ces objets sont créés en tant qu'instances de classes et interagissent dynamiquement pour offrir le comportement décrit par les cas d'utilisation. La modélisation objet définit le comportement requis par les différentes classes pour assurer la bonne mise en place des cas d'utilisation et des règles de gestion. Les objets constituent la base de l'architecture des applications, ils peuvent être réutilisés à travers des domaines d'application ou encore être identifiés et dérivés directement des cas d'utilisation ou des domaines d'application. Une classe est composée de :

- Attributs : représentant des données dont les valeurs représentent l'état de l'objet.
- La méthode : il s'agit des opérations applicables aux objets.

Après avoir dégagé le dictionnaire de données épuré, nous pouvons dégager les classes ainsi leurs méthodes et leurs attributs qui sont présentés dans le tableau suivant :

N°	Nom Classe	Liste des Attributs	Méthodes : constructeur Getters and setters
1	Client	ID	client () ;getID() ;
		nom	getNom() ; setNom(Sting a) ;
		prenom	getPrenom() ; setPrenom(Sting a) ) ;
2	Compte	num_comptes	Compte() ;getNum_comptes() ; set-
3	CompteCourant	solde	getSolde() ; setSolde (Double a) ;
4	CompteEpargne	solde	getSolde() ; setSolde (Double a) ;
		prenom	getPrenom() ; setPrenom(Sting a) ) ;
5	Virement	ID_operaton	getlOperation() ;setldOperation(int a) ;
6		montant	getMontant() ; setMontant(Double a) ;
		date_Virement	getDateVirement; setDateVirement(Date a)
6	Versement	montant	getMontant() ; setMontant(Double a) ;
		Date_Versement	getDateversement () ; setDateversement (Date a) ;
7	Retrait	ID_Retrait	getlDretrait() ;setlDretrait(Date a) ;
8		Date_Retrait	getDateretrait() ;setDateretrait(Date a) ;
		montant	getMontant() ; setMontant(Double a) ;
	tracessoldes	ID_Operation	getldoperation() ;setildOperation(Int a) ;
		date_operation	getDateoperation() ;setDateoperation (Date
		solde	getSolde() ;setSolde(Double a) ;

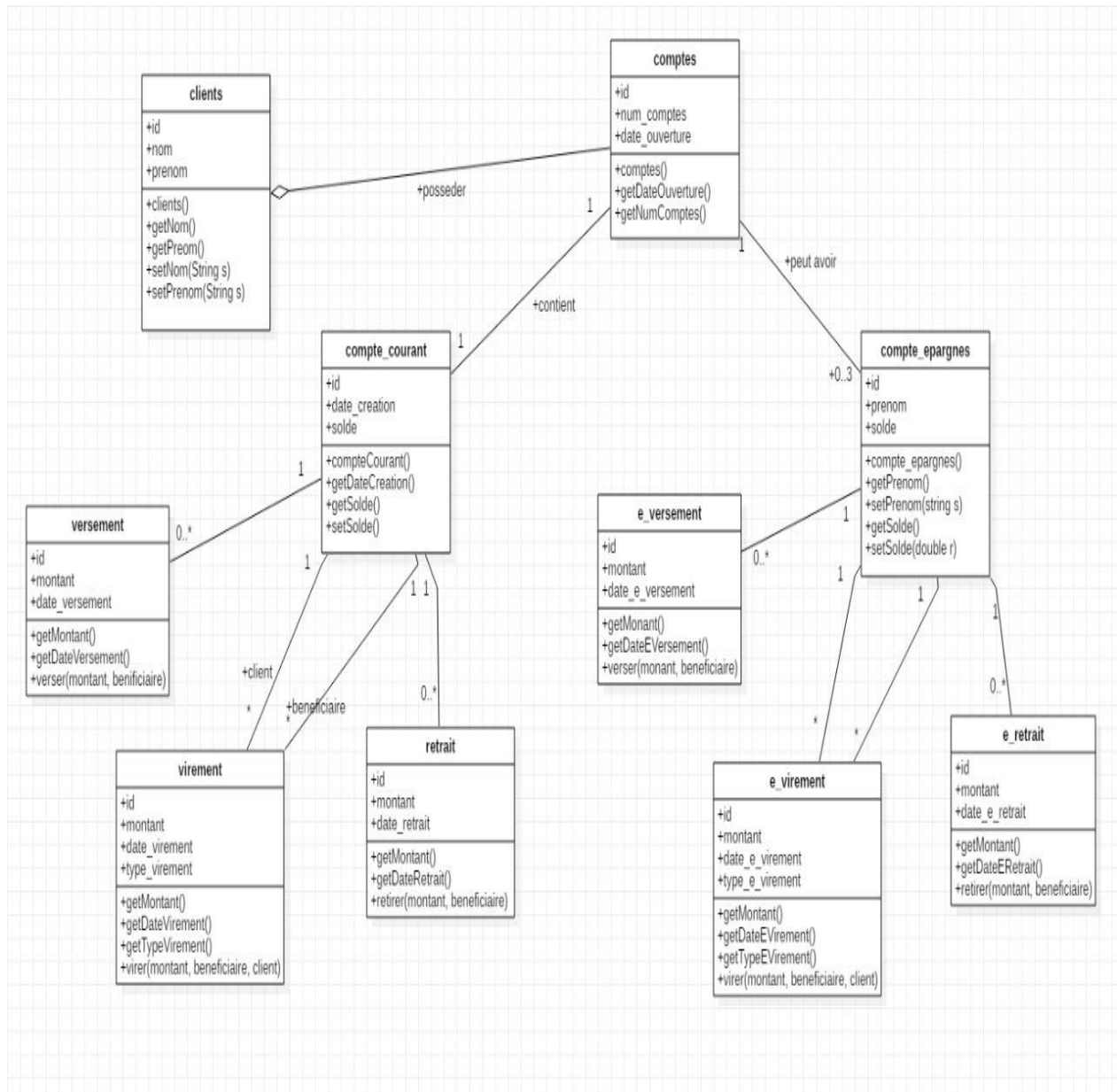


Figure 3.6: Diagramme d'activités : Virement

## 3.8 Conclusion

Dans ce chapitre, nous avons présenté le langage de modélisation UML, nous avons décrit le diagramme des cas d'utilisation, d'activités, de classes, et le diagramme de séquences. Dans le chapitre suivant nous allons spécifier le langage de programmation et les différents outils utilisés.

# Chapitre 4

## Environnement technique

### 4.1 Architecture du système et environnement de test

#### 4.1.1 Pourquoi utiliser un Framework ?

Un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle, et accompagné du plan de l'architecture cible du Framework. Les avantages des Framework sont nombreux. En effet, un Framework est portable, de la part de son abstraction de la base de données et de la gestion générique du cache. Les temps de développement avec un Framework sont réellement plus courts. Tous les outils essentiels sont déjà écrits. Le développement des applications sécurisées est facile. Grâce aux systèmes d'authentification, à la gestion des injections SQL ainsi qu'à la protection CSRF (Cross-Site Request Forgery) qui est gérée par la plupart des Framework. Les Framework sont des outils communautaires et ont, par conséquent, des forums, des listes de diffusion et des canaux IRC pour les soutenir. De plus vu que les Framework sont largement déployés, la chance de trouver les correctifs des problèmes rencontrés est plus grande. Utiliser le langage PHP et passer par le modèle MVC signifie l'utilisation des outils convenables pour bien respecter ce modèle et bien mener nos travaux de développement. Les solutions proposées sur ce domaine sont les Framework et les CMS.

### 4.2 Le Framework Laravel

#### 4.2.1 Définition

Laravel est un Framework web gratuit, open-source, créé par Taylor Otwell et destiné au développement d'applications Web suivant le modèle-vue-contrôleur (MVC). Certaines des caractéristiques de Laravel sont un système modulaire système d'emballage avec un gestionnaire de dépendances dédié, pour accéder à différentes façons de bases de données relationnelles, les services publiques que l'aide au déploiement d'applications et de maintenance, et son orientation vers le sucre syntaxique. En Mars 2015, Laravel est considéré comme l'une des plus populaires Framework PHP. Le code source de Laravel est hébergé sur Git Hub sous licence conformément aux termes de licence MIT. [1\*]

#### 4.2.2 Structure du répertoire

- Répertoire App

Contient le code de base de l'application. Le répertoire App contient une variété



Figure 4.1: Laravel

De répertoires supplémentaires tels que Console, http qui contient les contrôleurs et Providers et les modèles d'application.

- Répertoire Bootstrap :  
Contient des fichiers d'auto loading.
- Répertoire Config : Comme son nom l'indique, contient tous les fichiers de configuration de l'application.
- Répertoire Data base : Contient la migration de la base de données.
- Répertoire public  
Contient le fichier, ce qui est le point d'entrée pour toutes les demandes entrant dans l'application. Ce répertoire contient également les images, JavaScript et CSS.
- Répertoire Routes  
Contient toutes les définitions de route pour l'application. Par défaut, plusieurs fichiers d'itinéraire sont inclus avec Laravel:
  - web.php
  - api.php
  - console.php

## 4.3 Les technologies utilisées

### 4.3.1 Laravel

Laravel, initie une nouvelle façon de concevoir un Framework en utilisant ce qui existe de mieux pour chaque fonctionnalité. Laravel n'est pas seulement le regroupement de bibliothèques existantes ; c'est aussi un ensemble de nombreux composants originaux et surtout une orchestration de tout cela.

### 4.3.2 Bootstrap

Bootstrap est une boîte à outils open source pour le développement avec HTML, CSS et JS, propose un prototypage rapide des idées, permet de créer une application entière,

un système de grille réactif, de nombreux composants prédéfinis et des plugins puissants construits sur jQuery . [2\*]

## 4.4 Les outils utilisés

### 4.4.1 MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server.

### 4.4.2 PhpMyAdmin

PhpMyAdmin est un outil logiciel gratuit écrit en PHP, destiné à gérer l'administration de MySQL sur le Web. phpMyAdmin supporte un large éventail d'opérations sur MySQL et MariaDB.[3\*]

### 4.4.3 Modelio

Modelio est un environnement de modélisation open source UML, il offre une large gamme de fonctionnalités basées sur des standards pour les développeurs de logiciels, les analystes, les concepteurs, les architectes d'entreprise et les architectes système. [4\*]

### 4.4.4 XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

### 4.4.5 Conclusion

Ce chapitre a été consacré à l'environnement technique de notre projet. Nous y avons cité le langage de programmation et les différents outils choisis en présentant une description de chacun de ces outils. Dans le chapitre qui suit, nous présenterons la réalisation de notre projet.

# Chapitre 5

## Réalisation

Dans ce chapitre, nous présentons la réalisation du projet, nous traiterons la phase qui a pour objectif l'implémentation de notre application, puis nous décrirons les différentes fonctionnalités de ces dernières. Une fois l'analyse et la conception de l'application terminées, nous passons à l'étape suivante, celle de la réalisation. La phase de réalisation est considérée comme étant la concrétisation finale de toute la méthode de conception.

### 5.1 L'architecture de l'application

Le Framework Laravel se base sur une architecture MVC comme nous indique la figure suivante. Le contrôleur est responsable de la logique de contrôle de l'application, il sert à gérer les demandes des utilisateurs et à récupérer des données, en tirant parti des modèles, les modèles servent à interagir avec la base de données et récupérer les informations des objets, les vues pour afficher des pages.

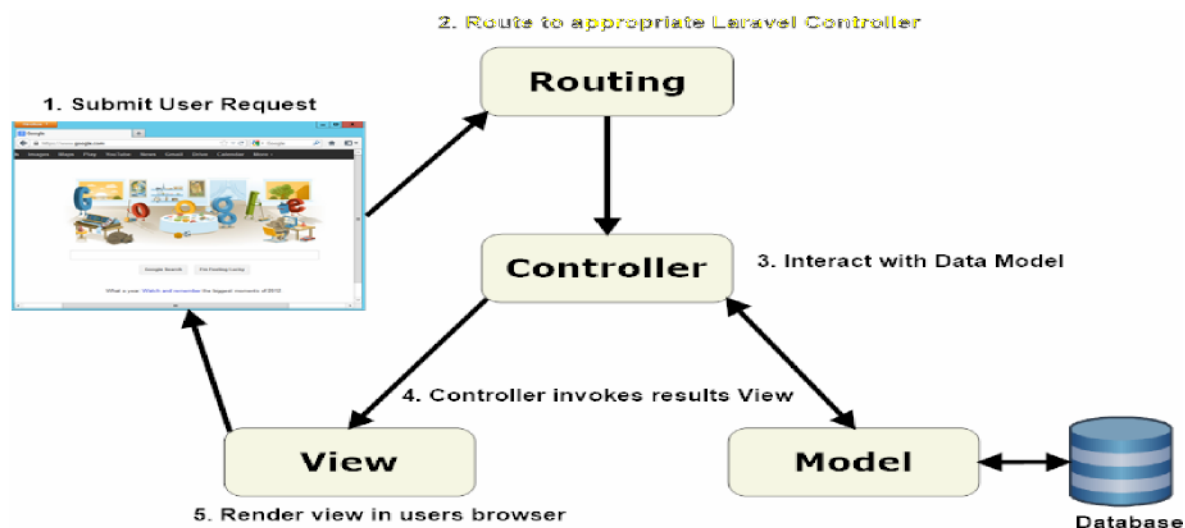


Figure 5.1: MVC

### 5.2 L'implémentation de <<agenceBanq>>

#### 5.2.1 Installation et organisation

Avant de commencer l'installation du Framework Laravel, nous devons installer <<composer >> qui est un gestionnaire de dépendances libre écrit en PHP et permet d'installer les bibliothèques, la commande est <<composer global require laravel/installer>> Nous nous plaçons dans le répertoire <<www>> de wamp afin de pouvoir travailler en local.



```
MINGW64:/c/xampp/htdocs
HP@DESKTOP-CAS8NAD MINGW64 ~
$ cd ..

HP@DESKTOP-CAS8NAD MINGW64 /c/Users
$ cd ..

HP@DESKTOP-CAS8NAD MINGW64 /c
$ cd xampp/

HP@DESKTOP-CAS8NAD MINGW64 /c/xampp
$ cd htdocs/

HP@DESKTOP-CAS8NAD MINGW64 /c/xampp/htdocs
$ |
```

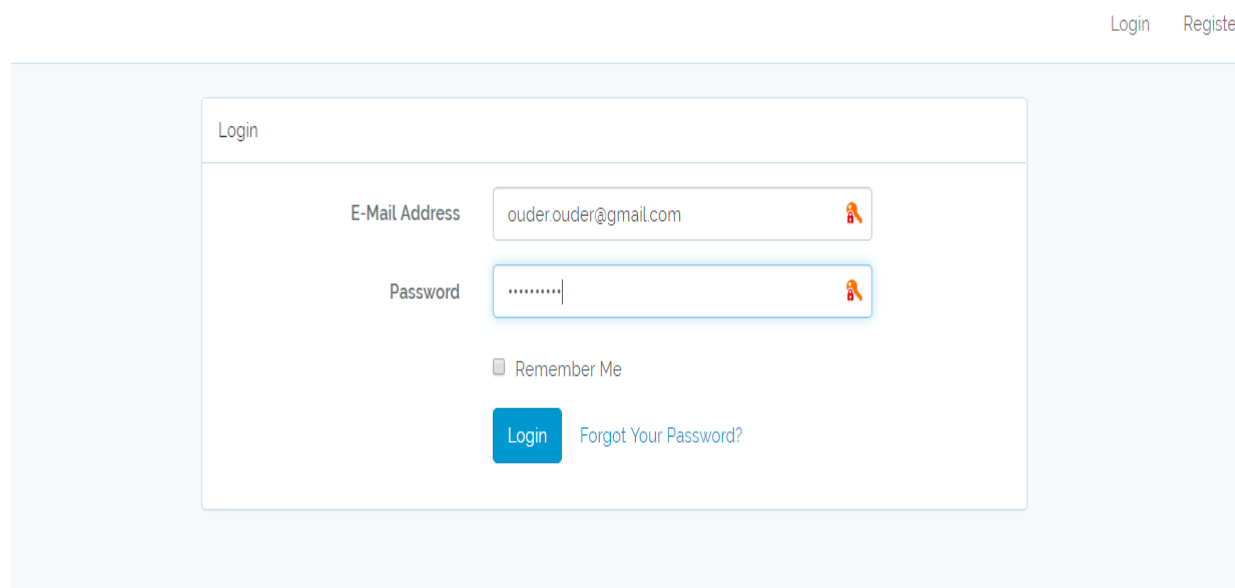
Figure 5.2: dossier d'installation Laravel

Pour installer le projet Laravel avec des composants de base la commande suivante composer << composer create-project laravel/laravel agenceBanq ~prefer-dist>>

```
HP@DESKTOP-CAS8NAD MINGW64 /c/xampp/htdocs
$ composer create-project laravel/laravel agenceBanq --prefer-dist
```

Figure 5.3: création du Framework Laravel

Le dossier <<AgenceBanq>> est créée par la suite contenant les différents dossiers utiles à l'utilisation du Framework Laravel. Pour lancer localement notre application Laravel, nous utilisons la commande <<php artisan serve >>, nous aurons accès sur notre navigateur web à une page d'accueil Laravel qui ressemblera à la suivante



## 5.2.2 Les routes

Quand la requête atteint le fichier public/index.php l'application Laravel est créée et configurée et l'environnement est détecté. Ensuite le fichier web.php est chargé, c'est avec ce fichier que la requête va être analysée et dirigée. Regardons ce que contient notre fichier de routage sur la figure suivant

```
//// compte courant
Route::delete('Compte_courant/{id}','CompteCourantController@destroy');
Route::delete('Compte_epargnes/{id}','CompteEpargneController@destroy');
Route::post('Store_Compte_Courant','CompteCourantController@store');
// Opération
Route::get('Operations','OperationController@index');
Route::get('Effectuer_Virement','OperationController@index_virement');
Route::get('Effectuer_Virement_Epargne','OperationController@index_retrait_virement');

Route::get('Effectuer_Retrait','OperationController@index_retrait');
Route::get('Effectuer_Retrait_Epargne','OperationController@index_retrait_epargne');

Route::get('Effectuer_Virement','OperationController@index_virement');
Route::get('Effectuer_Virement_epargne','OperationController@index_virement_epargne');
Route::post('Virement_Compte_Epargne','CompteEpargneController@virement');
```

Figure 5.4: les routes

### 5.2.3 Les migrations

Les migrations représentent la partie base de données de notre application, elles permettent de mettre à jour le schéma des bases de données. Pour cela nous devons configurer en premier lieu sur le fichier << .env >> les champs nécessaires à la connexion à notre base de données créée auparavant. La création de la migration se fait à l'aide de la commande << php artisan make:migration create\_comptes\_table >>. Nous définissons l'ensemble des champs utilisés dans notre table << comptes >>

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateComptesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('comptes', function (Blueprint $table) {
            $table->increments('id_comptes');
            $table->dateTimeTz('num_comptes');
            $table->dateTime('date_ouverture');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('comptes');
    }
}
```

Figure 5.5: les migrations

### 5.2.4 La classe Model

Cette classe nous permet de donner une représentation sous forme d'objets manipulables à tous les éléments de la base de données et qui permet de simplifier grandement les opérations en utilisant Eloquent qui représente un ORM. La création du model se fait à l'aide de la commande <<php artisan make:model Compte>>

```

<?php

namespace App;
use Illuminate\Database\Eloquent\SoftDeletes;

use Illuminate\Database\Eloquent\Model;
use App\Client;

class Compte extends Model
{
    use SoftDeletes;
    protected $dates = ['deleted_at'];
    protected $primaryKey = 'id_comptes';
    public function client(){
        return $this->belongsTo('App\Client');
    }
    public function compte_courant(){
        return $this->hasOne('App\Compte_courant');
    }
    public function compte_epargne(){
        return $this->hasMany('App\Compte_epargne');
    }
}

```

Figure 5.6: model

### 5.2.5 Le contrôleur

La tâche d'un contrôleur est de réceptionner une requête et de définir la réponse appropriée, à l'aide de la commande << php artisan make:controller CompteController >> nous créons le contrôleur CompteController. Nous avons les méthodes qui sont chargées de générer la vue, parmi elles la méthode index qui renvoie vers la vue Comptes.

```

namespace App\Http\Controllers;
use App\Compte;
use Illuminate\Http\Request;
use DB;

class CompteController extends Controller
{
    public function __construct(){
        $this->middleware('auth');
    }
    public function index(){
        $compte= Compte::all();
        return view('Comptes',['listcompte'=>$compte]);
    }
    public function suppression(){
        $compte= Compte::all();
        return view('Comptes.Supprimer_Compte',['listcompte'=>$compte]);
    }
    //affiche le formulaire de creation cvs
    public function create(){
        return view('Comptes.Create_Compte');
    }
    //enregister un cv
    public function store(Request $request){
        $compte=new Compte();
        session()->flash('success','le Compte a ete bien enregistrer');
        return redirect('Comptes');
    }
}

```

Figure 5.7: Contrôleur

### 5.2.6 Les vues :

Le processus de visualisation de pages se fait a travers des vues ainsi que les routes et a une extension de blade.php.

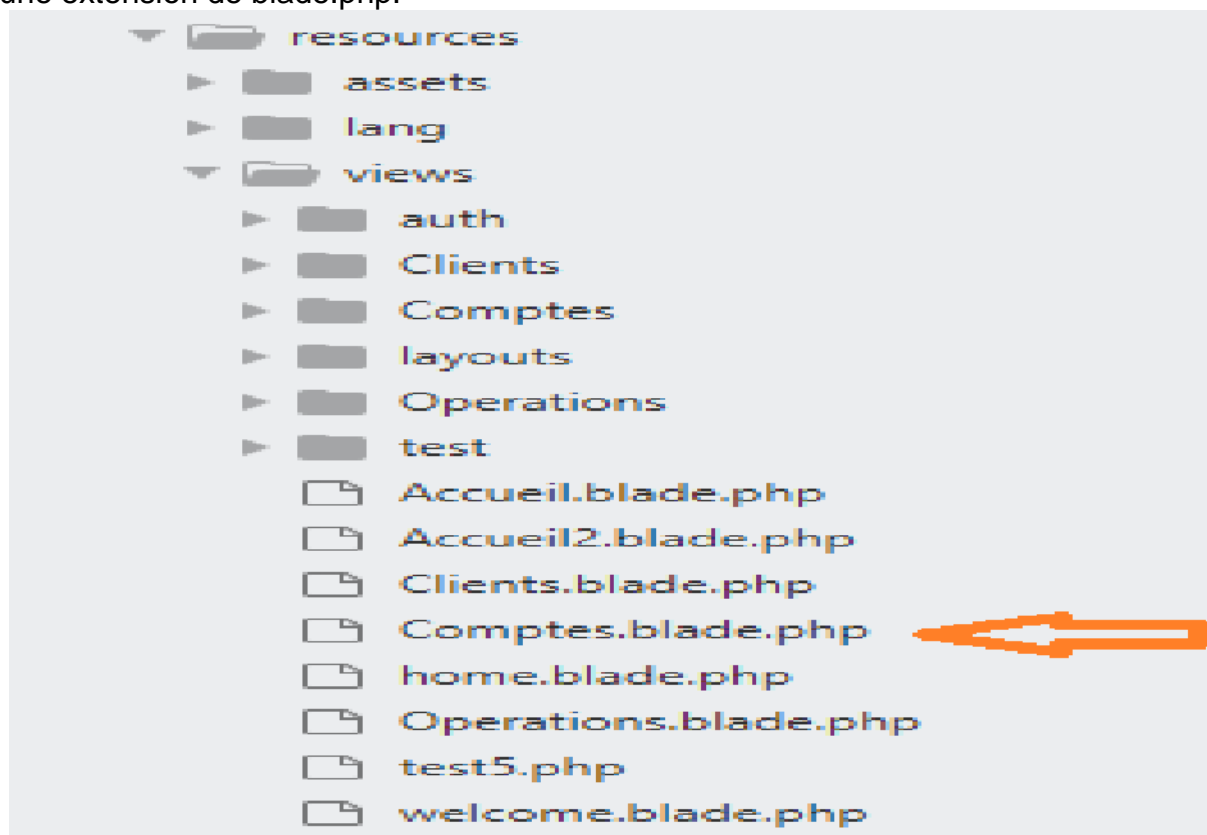


Figure 5.8: vues

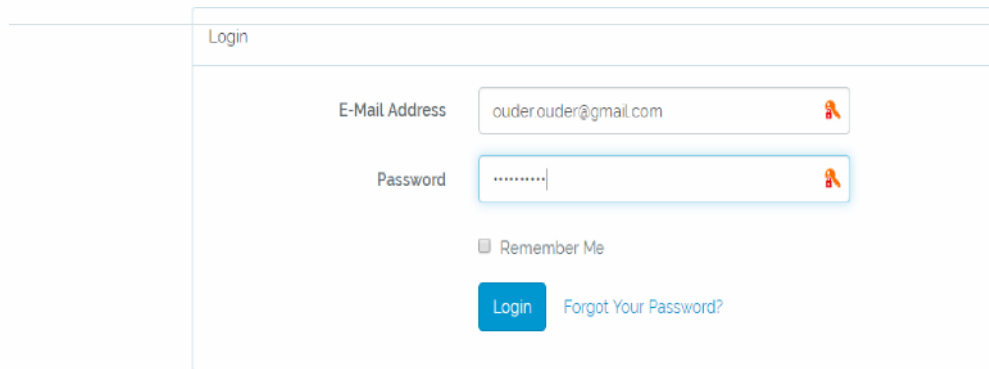
## 5.3 Présentation des interfaces

Après les phases d'étude de l'existant, la conception et la modélisation fonctionnelle et organisationnelle nous avons pu développer les interfaces de notre application, voici quelques captures de nos principales interfaces :

### 5.3.1 Interface d'authentification

Cette interface permet à l'employé de s'authentifier

[Login](#) [Register](#)



The image shows a login form titled "Login". It contains two input fields: "E-Mail Address" with the value "ouderouder@gmail.com" and "Password" with masked characters ".....". Both fields have a small red icon with a person silhouette to their right. Below the password field is a checkbox labeled "Remember Me". At the bottom, there is a blue "Login" button and a link "Forgot Your Password?".

Figure 5.9: Interface d'authentification.

### 5.3.2 Interface d'accueil

C'est l'interface d'accueil de l'application "AganceBanq"

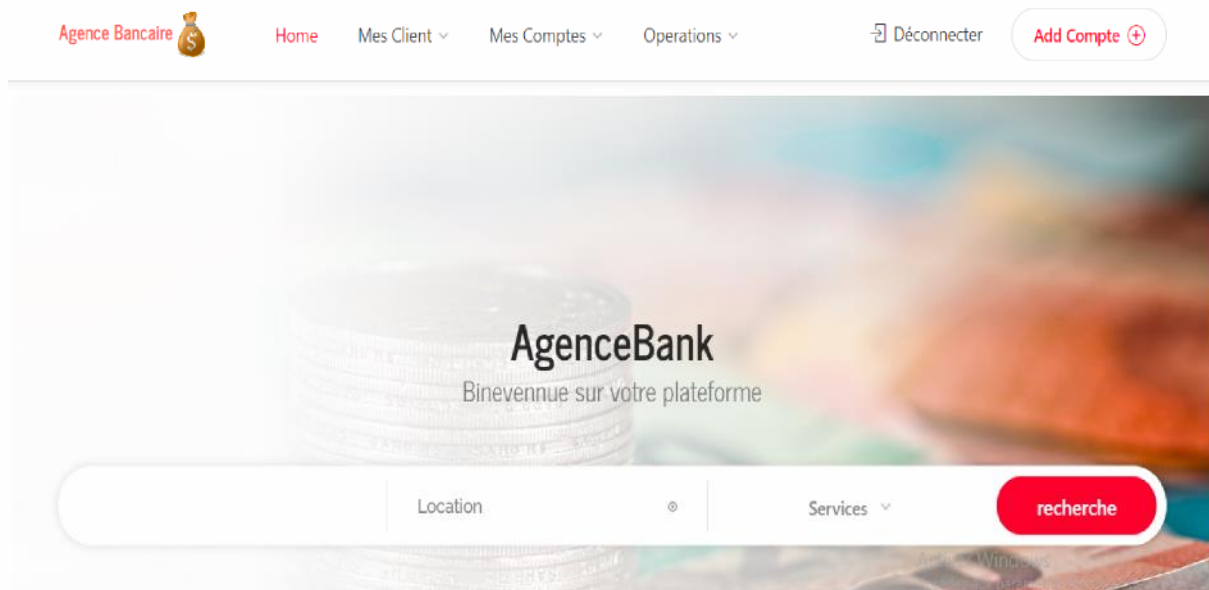


Figure 5.10: Interface d'accueil.

## 5.3.3 Partie clients

### 5.3.3.1 Guide client

Pour accéder à la liste des clients ou ajouter un nouveau client

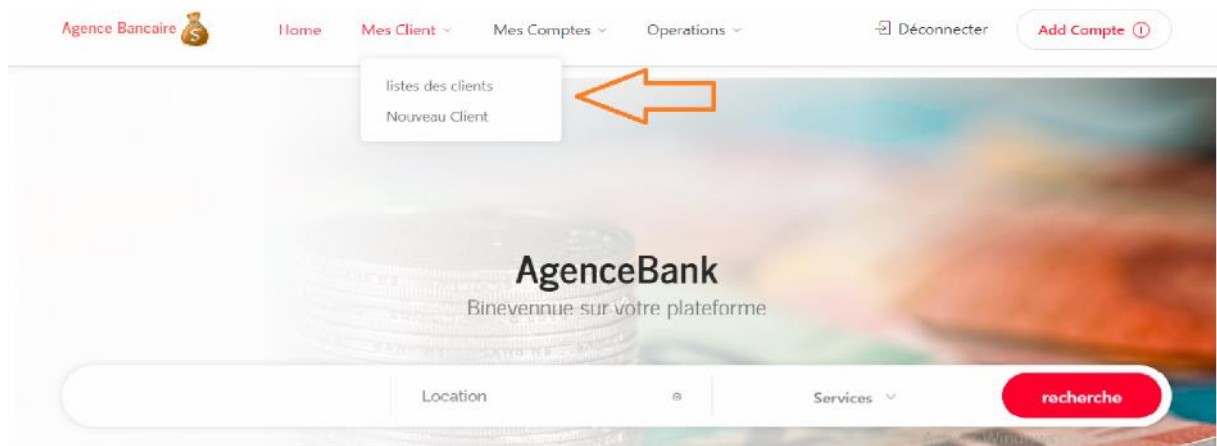


Figure 5.11: accéder aux opérations sur clients.

### 5.3.3.2 Accès à la page d'ajout d'un client

ajouter un nouveau client

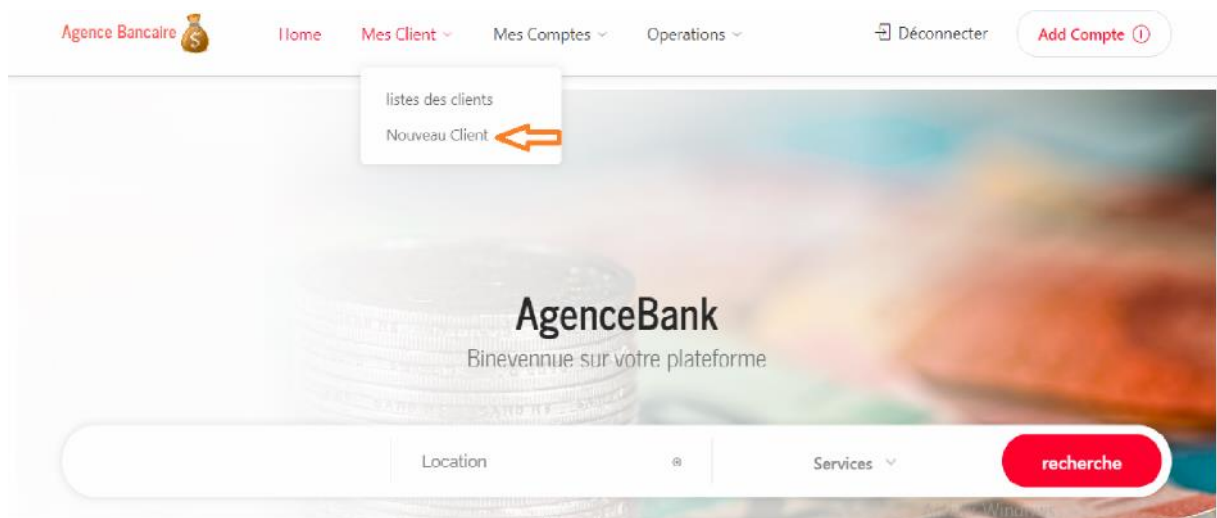


Figure 5.12: Nouveau Client.

### 5.3.3.3 Interface d'ajout d'un nouveau client

Après le remplissage du formulaire et après un clic sur le bouton enregistrer, une validation des données au niveau du contrôleur est faite, et se termine par l'ajout d'un client

Agence Bancaire Home Mes Client Mes Comptes Operations Déconnecter Add Compte

Nom: ouder Prenom: samir Cin: AA48661

Email: ouder.samir@gmail.com Telephone: 0619454828

Address: 57 av mokhtar jazoulite apt 6 rabat ocean

Address 2: Apartment, studio, or floor

Enregistrer

Options: Liste des Clients, Nouveau Client

Figure 5.13: ajout d'un client.

### 5.3.3.4 Interface de liste des clients

Cette interface affiche la liste des clients après l'insertion d'un client et affiche une alerte pour informer que le client a été bien ajouté ou non

le client a été bien enregistré

Nom	Prenom	Telephone	Cin	Adresse	Email	Numero compte	Action
ouder	ahmed	0619454020	AA39661	57 av mokhtar jazoulite apt 6 rabat ocean	ouder.samir@gmail.com	200619154657	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>
eloufir	chakib	0619454565	1212121212	57 av mokhtar jazoulite apt 6 rabat ocean	chakibEloufir@gmail.com	170619175535	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>
oubrahim	hamza	6632385404	AA46548	9 Holland Gardens Waterford Place	hamza.oubrahim@gmail.com	170619161717	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>

Figure 5.14: redirection liste clients.



### 5.3.4 Modifier un client

Pour modifier un client il faut cliquer sur buttons vert Editer

la liste des Clients							
Client							
Nom	Prenom	Telephone	Cin	Adresse	Email	Numero compte	Action
eloufir	chakib	0619454565	1212121212	57 av mokhtar jazoulite apt 6 rabat ocean	chakibEloufir@gmail.com	170619175535	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>
oubrahim	hamza	6632385404	AA46548	9 Holland Gardens Waterford Place	hamza.oubrahim@gmail.com	170619161717	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>
ouder	samir	0619454828	AA48661	57 av mokhtar jazoulite apt 6 rabat ocean	ouder.samir@gmail.com	170619155218	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>

Figure 5.15: pour modifier un client.

### 5.3.5 Supprimer un client

Pour supprimer un client

la liste des Clients							
Client							
Nom	Prenom	Telephone	Cin	Adresse	Email	Numero compte	Action
eloufir	chakib	0619454565	1212121212	57 av mokhtar jazoulite apt 6 rabat ocean	chakibEloufir@gmail.com	170619175535	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>
oubrahim	hamza	6632385404	AA46548	9 Holland Gardens Waterford Place	hamza.oubrahim@gmail.com	170619161717	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>
ouder	samir	0619454828	AA48661	57 av mokhtar jazoulite apt 6 rabat ocean	ouder.samir@gmail.com	170619155218	<a href="#">Detail</a> <a href="#">Editer</a> <a href="#">Supprimer</a>

Figure 5.16: pour supprimer un client.

### 5.3.6 Gestion des comptes

Pour accéder à la liste des comptes et faire des opérations (consulter solde ou relevé de compte, supprimer ou créer un compte courant ou épargne)

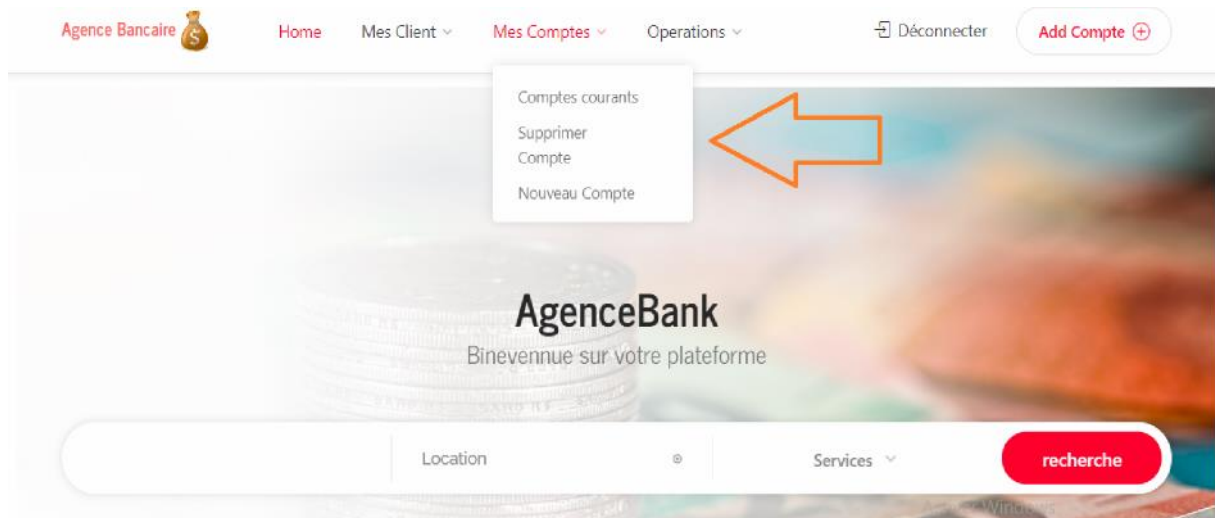


Figure 5.17: accéder aux Guide des comptes.

### 5.3.7 L'ajout d'un nouveau compte

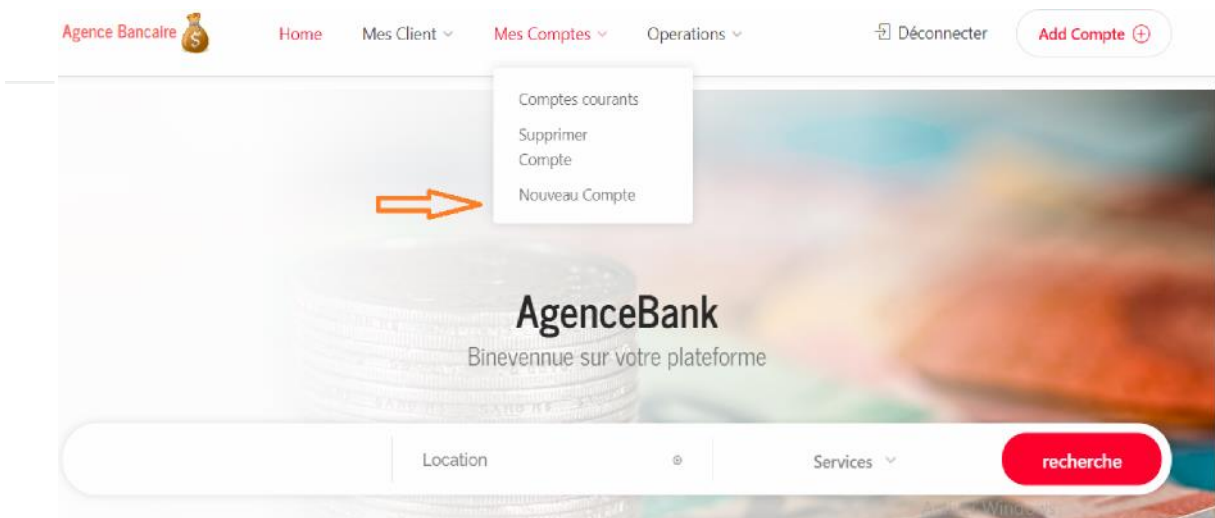


Figure 5.18: ajouter un nouveau compte.

### 5.3.8 Choix de type de compte à créer

Après pouvoir créer un compte on choisit le type de compte

The screenshot shows a web interface for creating an account. On the left, a dropdown menu labeled 'Type Compte' is open, showing three options: 'Type Compte' (highlighted in blue), 'Compte Courant', and 'Compte Epargne'. Below the dropdown is a search bar with the placeholder text 'rechercher un client'. On the right, there is a vertical sidebar with four buttons: 'Options', 'Liste des Comptes', 'Supprimer Comptes', and 'Nouveau Compte' (which is green and highlighted).

Figure 5.19: choix de type de compte à créer.

### 5.3.9 Création de compte courant

Il faut saisir le solde initial et le débit max

The screenshot shows the 'Creation de compte' form for a current account. At the top, there is a navigation bar with the logo 'Agence Bancaire', links for 'Home', 'Mes Client', 'Mes Comptes', and 'Operations', a 'Déconnecter' button, and an 'Add Compte' button. The main form has a dropdown menu set to 'Compte Courant'. Below it is a text field containing 'Mr ouder samir AA48661 : 170619155218'. There are three input fields: 'NumCompte' (empty), 'Solde' (containing '100000'), and 'Debit Max' (containing '50'). At the bottom, there is a red 'enregister' button. On the right, the same sidebar from Figure 5.19 is visible, with 'Nouveau Compte' highlighted.

Figure 5.20: création du compte courant.

### 5.3.10 Création de compte courant avec succès

Après une création d'un compte courant une redirection vers la liste des clients pour vérifier la création

la liste des Comptes							
le compte a été bien enregistrer							
Nom et Prenom	Telephone	Cin	Date D'ouverture	Numero Compte	Solde	operation	compte epargne
ouder samir	0619454828	AA48661	2019-06-17 15:52:18	170619155218	100000	Relevé de Compte	<div>samir 2000 DH</div> <div>hassan 20000 DH</div> <div>asmae 3000 DH</div>
oubrahim hamza	6632385404	AA46548	2019-06-17 16:17:17	170619161717	10000	Relevé de Compte	<div>rania 100 DH</div>
eloufir chakib	0619454565	1212121212	2019-06-17 17:55:35	170619175535	95000	Relevé de Compte	<div>hicham 18900 DH</div> <div>nezha 30000 DH</div>
ouder ahmed	0619454020	AA39661	2019-06-20 15:46:57	200619154657	50000	Relevé de Compte	

Figure 5.21: après création compte courant.

### 5.3.11 Création du compte épargne

Il faut saisir le solde initial et le prénom et on clique sur le bouton enregistrer

Agence Bancaire

Home

Mes Client

Mes Comptes

Operations

Déconnecter

Add Compte

Compte Epargne

Options

Creation de compte

Mr ouder samir AA48661 : 170619155218

Solde

2000

Prenom

samir

Taux interet

10

Liste des Comptes

Supprimer Comptes

Nouveau Compte

enregistrer

Figure 5.22: création du compte épargne.

### 3.12 Création de compte courant avec succès

Après une création d'un compte épargne une redirection vers la liste des clients pour vérifier la création est faite

la liste des Comptes							
le compte epargne a ete bien enregistrer							
Nom et Prenom	Telephone	Cin	Date D'ouverture	Numero Compte	Solde	operation	compte epargne
ouder samir	0619454828	AA48661	2019-06-17 15:52:18	170619155218	100000	Relevé de Compte	samir 2000 DH hassan 20000 DH asmae 3000 DH
oubrahim hamza	6632385404	AA46548	2019-06-17 16:17:17	170619161717	10000	Relevé de Compte	rania 100 DH
eloufir chakib	0619454565	1212121212	2019-06-17 17:55:35	170619175535	95000	Relevé de Compte	bicham 18900 DH nezha 30000 DH
ouder ahmed	0619454020	AA39661	2019-06-20 15:46:57	200619154657	50000	Relevé de Compte	ahmed 400 DH

Figure 5.23: après création compte épargne.

### 5.3.13 Suppression de Compte

Pour supprimer un compte ...

Agence Bancaire

Home

Mes Client

Mes Comptes

Operations

Déconnecter

Add Compte

Information sur Compte

Options

Liste des Comptes

Supprimer Comptes

Ajouter Compte

Figure 5.24: supprimer un compte.

### 5.3.14 Recherche par client et supprimer

Supprimer un compte après recherche par critère

### Compte

References : ouder samir (AA48661) / Num Compte : 170619155218

	Nom	Date Ouverture	Solde	Action
Compte courant	ouder	2019-06-17 16:37:41	100000 DH	<a href="#">Detail</a> <a href="#">Supprimer</a>

	Prenom	Date Ouverture	Solde	Action
Compte Epargne	1 samir	2019-06-17 17:25:06	2000 DH	<a href="#">Detail</a> <a href="#">Supprimer</a>
	2 hassan	2019-06-17 17:56:37	20000 DH	<a href="#">Detail</a> <a href="#">Supprimer</a>
	3 asmae	2019-06-17 17:58:10	3000 DH	<a href="#">Detail</a> <a href="#">Supprimer</a>

Figure 5.25: supprimer un compte après recherche.

### 5.3.15 Opération sur compte

Pour accéder au guide des opérations sur les comptes (courants/épargnes)

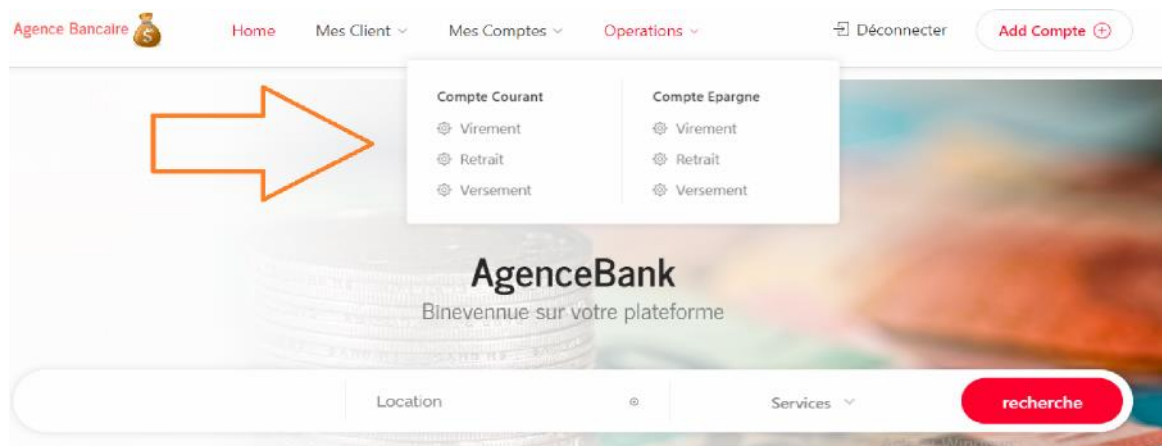



Figure 5.26: opérations sur les comptes.

### 5.3.15.1 Versement

Pour effectuer un versement sur un compte courant il faut faire une recherche sur le compte courant du client et saisir le montant

Agence Bancaire 

Home

Mes Client ▾

Mes Comptes ▾

Operations ▾

Déconnecter

Add Compte +

Virement compte courant

Mr ouder samir AA48661 170619155218

NumCompte	Nom	Prenom	Montant
170619155218	ouder	samir	9000

Effectuer un Virement

Effectuer un Versement

Options

Effectuer un Retrait

enregistrer

Figure 5.27: versement.

### 5.3.15.2 Le Relevé du compte

Après la réussite du versement une redirection vers la liste des clients est faite et pour la vérification du versement il faut cliquer sur le bouton de relevé du compte

la liste des Comptes							
Nom et Prenom	Telephone	Cin	Date D'ouverture	Numero Compte	Solde	operation	compte epargne
ouder samir	0619454828	AA48661	2019-06-17 15:52:18	170619155218	100000	Relevé de Compte	<div>samir 3000 DH</div> <div>hassan 15000 DH</div> <div>asmae 3000 DH</div>
oubrahim hamza	6632385404	AA46548	2019-06-17 16:17:17	170619161717	10000	Relevé de Compte	<div>hania 100 DH</div>
eloufir chakib	0619454565	1212121212	2019-06-17 17:55:35	170619175535	95000	Relevé de Compte	<div>hicham 15000 DH</div> <div>merha 30000 DH</div>
ouder ahmed	0619454020	AA39661	2019-06-20 15:46:57	200619154657	50000	Relevé de Compte	<div>ahmed 400 DH</div>

Figure 5.28: relevé.

### 5.3.15.3 Consultation du compte X

Releve de compte courant	170619155218	solde: 100000
jj/mm/aaaa		
jj/mm/aaaa		

Figure 5.29: consulter le relevé.

Après le choix de date début et de date de fin (échéance) le résultat suivant montre que le versement est bien fait sur le compte X

Releve de compte courant : 170619155218      solde : 100000

ouder samir

0619454828

AA48661

2019-06-17

170619155218

100000

01/06/2019

23/06/2019

Opération	Type Opération	Date Opération	Montant	Solde	Num Compte Bénéficiaire
virement CC -> CC	debit	2019-06-18 00:09:58	8900 DH	100000 DH	eloufir 170619175535
versement	crédit	2019-06-17 18:21:42	9000 DH	109000 DH	
retrait	Débit	2019-06-17 23:33:18	100 DH	108900 DH	

Figure 5.30: consulter le relevé.



### 5.3.16 Retrait

Pour effectuer un retrait sur un compte courant il faut faire une recherche sur le compte courant du client et saisir le montant

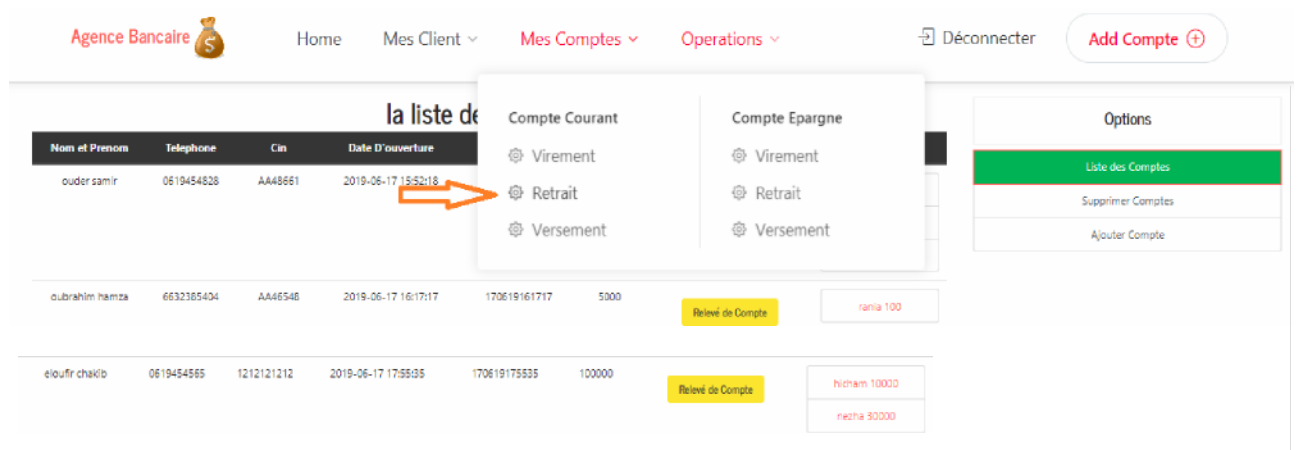


Figure 5.31: retrait.

#### 5.3.16.1 Interface de retrait

Pour effectuer un retrait sur un compte courant il faut faire une recherche sur le compte courant de client et saisir le montant.

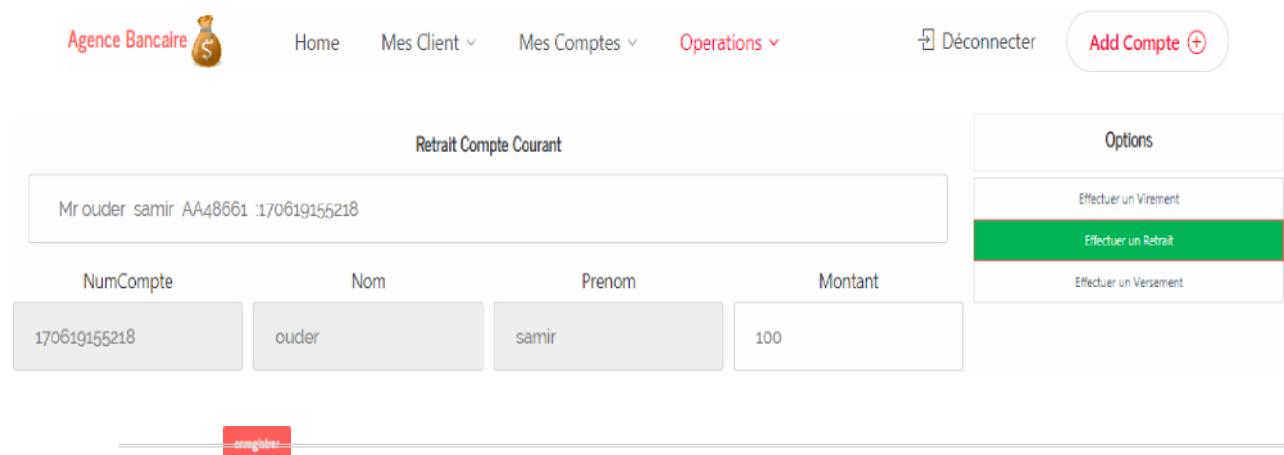


Figure 5.32: interface retrait.

### 5.3.16.2 Interface vérification retrait

#### vérification du retrait

Releve de compte courant : 170619155218    solde : 100000					
01/06/2019					
23/06/2019					
Opération	Type Opération	Date Opération	Montant	Solde	Num Compte Bénéficiaire
virement CC -> CC	debit	2019-06-18 00:09:58	8900 DH	100000 DH	eloufir 170619175535
versement	crédit	2019-06-17 18:21:42	9000 DH	109000 DH	
retrait	Débit	2019-06-17 23:33:18	100 DH	108900 DH	

Figure 5.33: vérification retrait.

### 5.3.17 virements

#### Pour effectuer un virement

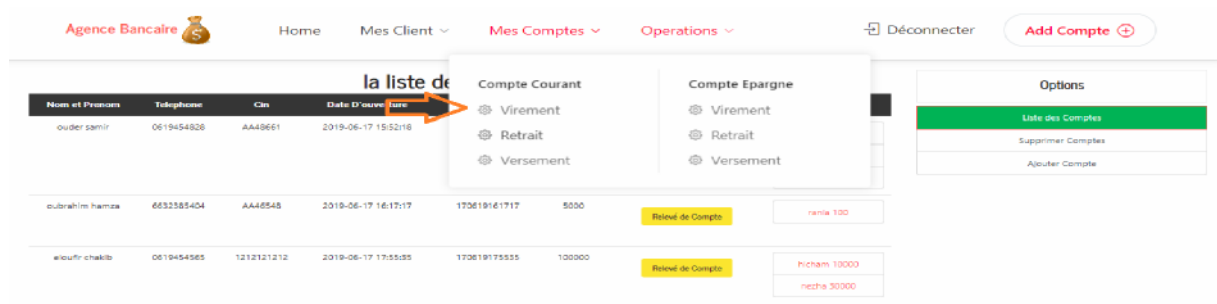


Figure 5.34: Virement.

#### 5.3.17.1 Effectuer virement

Effectuer un virement du compte courant x vers le compte courant y de montant 8900 DH

### Virement de Compte Courant

Client

ouder samir

Compte

Numero de compte : 170619155218 ---- compte courant --- solde --> 108900 DH

Bénéficiaire

eloufir chakib

Compte

Numero de compte : 170619175535 ---- compte courant --- solde --> 100000 DH

Montant

8900

Virement

Figure 5.35: effectuer virement.

### 5.3.17.2 Relevé de compte X

Le relevé de compte après le virement pour le compte courant x

Releve de compte courant : ouder samir 170619155218      solde : 100000					
la liste des Comptes					
01/06/2019					
30/06/2019					
Opération	Type Opération	Date Opération	Montant	Solde	Num Compte Bénéficiaire
virement CC -> CC	debit	2019-06-18 00:09:58	8900 DH	100000 DH	eloufir 170619175535
virement CC -> CC	credit	2019-06-21 11:37:35	8900 DH	108900 DH	ouder 200619154657
virement CC -> CC	debit	2019-06-21 11:43:15	8900 DH	100000 DH	eloufir 170619175535
versement	crédit	2019-06-17 18:21:42	9000 DH	109000 DH	
retrait	Débit	2019-06-17 23:33:18	100 DH	108900 DH	

Figure 5.36: relevé d compte.

### 5.3.17.3 Relevé de compte Y

Le relevé de compte après le virement pour le compte courant y

Releve de compte courant : eloufir chakib 170619175535      solde : 108900					
la liste des Comptes					
01/06/2019					
29/06/2019					
Opération	Type Opération	Date Opération	Montant	Solde	Num Compte Bénéficiaire
virement CC -> CC	credit	2019-06-18 00:09:58	8900 DH	108900 DH	ouder 170619155218
virement CC -> CE	debit	2019-06-18 00:18:52	8900 DH	100000 DH	hicham 170619175535
virement CC -> CC	debit	2019-06-20 14:56:31	5000 DH	95000 DH	oubrahim 170619161717
virement CC -> CC	credit	2019-06-21 11:38:17	5000 DH	100000 DH	ouder 200619154657
virement CC -> CC	credit	2019-06-21 11:43:15	8900 DH	108900 DH	ouder 170619155218

Figure 5.37: relevé d compte.

#### 5.3.17.4 Virement compte courant vers compte épargne

le virement entre un compte courant x et un compte épargne x1

**Virement de Compte Courant**

Client	eloufir chakib
Compte	Numero de compte : 170619175535 --- <b>compte courant</b> --- solde --> 108900 DH
Bénéficiaire	eloufir chakib
Compte	Numero de compte : 170619175535 --- <b>compte epargne</b> de hicham --- solde --> 10000 DH

Montant

8900

Virement

Figure 5.38: virement compte courant vers compte épargne.

#### 5.3.17.5 Relevé de compte courant

Le relevé du compte courant pour vérifier le virement

Releve de compte courant : eloufir chakib 170619175535 solde : 100000

Opération	Type Opération	Date Opération	Montant	Solde	Num Compte Bénéficiaire
virement CC -> CC	credit	2019-06-18 00:09:58	8900 DH	108900 DH	ouder 170619155218
virement CC -> CE	debit	2019-06-18 00:18:52	8900 DH	100000 DH	hicham 170619175535
virement CC -> CC	debit	2019-06-20 14:56:31	5000 DH	95000 DH	oubrahim 170619161717
virement CC -> CC	credit	2019-06-21 11:38:17	5000 DH	100000 DH	ouder 200619154657
virement CC -> CC	credit	2019-06-21 11:43:15	8900 DH	108900 DH	ouder 170619155218
virement CC -> CE	<b>debit</b>	2019-06-21 14:32:37	<b>8900 DH</b>	100000 DH	<b>hicham 170619175535</b>

Figure 5.39: relevé d compte.

### 5.3.17.6 Relevé de compte épargne

Le relevé de compte épargne pour vérifier le virement

Releve de Compte : 170619175535    Compte epargne : hicham    solde : 18900 DH					
01/06/2019					
30/06/2019					
Opération	Type Opération	Date Opération	Montant	Solde	Num Compte Bénéficiaire
virement CE -> CC	credit	2019-06-18 00:18:52	8900 DH	18900 DH	compte courant : eloufir
virement CE -> CE	debit	2019-06-21 14:24:23	9000 DH	10000 DH	compte epargne : nezha solde : 39000
virement CE -> CC	credit	2019-06-21 14:32:37	8900 DH	18900 DH	compte courant : eloufir

Figure 5.40: relevé de compte.

## 5.4 Conclusion

Ce chapitre a été consacré à la phase de réalisation. Nous y avons présenté l'ensemble des interfaces utilisées dans l'application et les différentes fonctionnalités de cette dernière.

# Bibliographie

- [1] <https://laravel.com/docs/5.8/>
- [2] <https://webilio.ca/pourquoi-utiliser-le-framework-laravel/>
- [3] <https://www.tristatetechnology.com/blog/13-advantages-laravel-development-services-enterprises/>
- [4] <https://openclassrooms.com/fr/courses/3613341-decouvrez-le-framework-php-laravel>
- [5] <https://openclassrooms.com/fr/courses/1811341-decouvrez-le-framework-php-laravel-ancienne-version/>