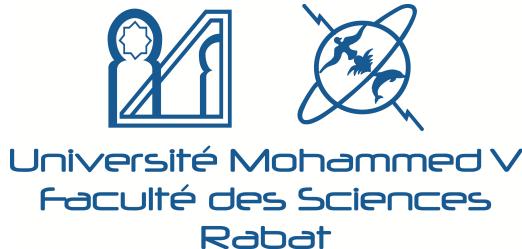


UNIVERSITÉ MOHAMMED V de Rabat  
Faculté des Sciences



Département d’Informatique

Filière Licence fondamentale  
en Sciences Mathématiques et Informatique

---

PROJET DE FIN D’ÉTUDES

---

intitulé :

**SMIBANQUE :Application E-banking Sécurisée En Java EE**

Présenté par :  
MOHAMMED KIHEL  
MERYEM LAASIRI

soutenu le 20 Juin 2022 devant le Jury

M. Abderrahim AIT WAKRIME Professeur à la Faculté des Sciences - Rabat Président  
M. Ahmed EL-YAHYAOUI Professeur à la Faculté des Sciences - Rabat Encadrant

Année universitaire 2021-2022

# Remerciements

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à notre encadrant Professeur Ahmed EL YAHYAOUI, pour tout le temps qu'il nous a consacré, ses conseils précieux, et pour la qualité de son suivi durant toute la période de notre projet.

Nous tenons également à remercier et exprimer nos profonds respects aux membres de jury d'avoir accepté de juger ce modeste travail.

De plus nous adressons nos fidèles remerciements à tout le corps professoral de La Faculté des Sciences de Rabat pour leurs qualités d'enseignement.

Nous n'oubliions pas de remercier nos familles , nos amis et toutes les personnes qui ont contribué de près ou de loin à l'élaboration de cette mémoire pour .

# Résumé

L'objectif visé à travers ce projet de fin d'études est de développer une application bancaire sécurisée en JAVA EE, cette application va proposer de nombreux services aux clients. D'un simple coup d'œil l'application permet d'avoir un aperçu de ses dépenses et de ses rentrées d'argent ainsi que sur le solde restant, une façon pratique et intuitive de gérer son budget à n'importe quel moment.

Pour aboutir à cette fin, nous allons effectuer une analyse des besoins fonctionnels ainsi qu'aux besoins non fonctionnels, ensuite on va procéder à l'analyse conceptuelle de l'application ce qui va nous permettre une meilleur structuration.  
Et dans le dernier chapitre nous allons faire une description détaillée des outils utilisés pour développer l'application . Et nous allons décrire les technologies utilisées pour la partie sécurité (MD5,Authentification forte ,HTTPS) .

Et enfin nous allons décrire l'application réalisée via des interfaces graphiques.

**Mots clés :** JAVA EE, Servlet, SQL, JSP, MD5, Firebase,Https,Apache

# Abstract

The objective of this graduation project is to develop a secure banking application in JAVA EE, this application will offer many services to customers. At a glance, the application provides an overview of your expenses and cash receipts as well as the remaining balance, a practical and intuitive way to manage your budget at any time. To achieve this end, we will carry out an analysis of the functional needs as well as the non-functional needs, then we will proceed to the conceptual analysis of the application which will allow us a better structuring.

And in the last chapter we will make a detailed description of the tools used to develop the application. And we will describe the technologies used for the security part (MD5, strong authentication, HTTPS) .

And finally we will describe the application carried out via graphical interfaces.

**Keywords :** JAVA EE, Servlet, SQL, JSP, MD5, Firebase,Https,Apache

# Table des matières

<b>Remerciements</b>	i
<b>Résumé</b>	ii
<b>Abstract</b>	iii
<b>Introduction</b>	ix
<b>1 Contexte et spécification des besoins</b>	1
1.1 Planification du projet . . . . .	5
1.2 Conclusion . . . . .	5
<b>2 Analyse des besoins et conception</b>	6
2.1 Analyse des besoins . . . . .	6
2.2 Conception . . . . .	7
2.2.1 Diagramme de cas d'utilisation . . . . .	7
2.2.2 Diagramme de classe . . . . .	8
2.2.3 Diagramme de séquence . . . . .	9
2.2.4 Diagramme d'activités . . . . .	12
2.3 Conclusion . . . . .	13
<b>3 Implémentation et réalisation</b>	14
3.1 Environnement du travail . . . . .	14
3.1.1 Modelio . . . . .	14
3.1.2 GanttProject . . . . .	15
3.1.3 Eclipse . . . . .	15
3.1.4 MySQL . . . . .	16
3.1.5 ApacheTomcat . . . . .	16
3.2 Les technologies utilisées . . . . .	17
3.2.1 Java EE . . . . .	17
3.2.2 Html . . . . .	18
3.2.3 CSS . . . . .	18
3.2.4 JavaScript . . . . .	19
3.2.5 SQL . . . . .	19
3.3 Outils utilisés pour la partie sécurité . . . . .	20
3.3.1 Fonction hachage : MD5 . . . . .	20
3.3.2 FireBase : Authentification forte . . . . .	21

3.3.3	Le protocole Https . . . . .	23
3.3.4	Conclusion . . . . .	23
3.4	Les interfaces graphiques . . . . .	24
3.4.1	Conclusion . . . . .	34
<b>Conclusion</b>		<b>35</b>
<b>Bibliographie</b>		<b>37</b>

# Table des figures

1.0.1 Le modèle MVC. . . . .	1
1.0.2 Modèle. . . . .	2
1.0.3 Vue. . . . .	2
1.0.4 Contrôleur. . . . .	2
1.0.5 Authentification forte. . . . .	3
1.0.6 MD5. . . . .	3
1.0.7 Le protocole HTTPS. . . . .	4
1.1.1 diagramme de Gantt sur la planification du projet. . . . .	5
2.2.1 Diagramme de cas d'utilisation . . . . .	8
2.2.2 Diagramme de classe . . . . .	9
2.2.3 Diagramme de séquence du procédure "authentification d'un client" . . . . .	10
2.2.4 Diagramme de séquence du procédure "Modification des coordonnées d'un client par l'administrateur" . . . . .	11
2.2.5 Diagramme d'activités du procédure "Désactiver un compte par l'administrateur" . . . . .	12
3.1.1 Modelio . . . . .	14
3.1.2 GanttProject . . . . .	15
3.1.3 eclipse. . . . .	15
3.1.4 MySQL . . . . .	16
3.1.5 ApacheTomcat . . . . .	17
3.2.1 JAVA EE . . . . .	17
3.2.2 HTML . . . . .	18
3.2.3 CSS . . . . .	18
3.2.4 JavaScript . . . . .	19
3.2.5 SQL . . . . .	20
3.3.1 Message Digits 5 . . . . .	20
3.3.2 Authentification forte . . . . .	21
3.3.3 Authentification forte :partie1 . . . . .	21
3.3.4 Authentification forte :partie2 . . . . .	22
3.3.5 Authentification forte :partie3 . . . . .	22
3.3.6 Authentification forte :partie4 . . . . .	22
3.3.7 Activation du protocole https . . . . .	23
3.4.1 Interface d'accueil . . . . .	24
3.4.2 1 ère interface d'inscription . . . . .	24
3.4.3 2 ème interface d'inscription . . . . .	25

3.4.4 Interface d'authentification . . . . .	25
3.4.5 Menu et profile de l'application . . . . .	26
3.4.6 Interface de virement normal . . . . .	26
3.4.7 Interface de virement vers un bénéficiaire . . . . .	27
3.4.8 Interface pour déposer une réclamation . . . . .	27
3.4.9 Interface qui affiche l'historique des notifications . . . . .	28
3.4.10 Interface qui affiche l'historique des opérations . . . . .	28
3.4.11 Interface de sécurité . . . . .	29
3.4.12 Interface qui affiche les informations concernant la carte bancaire . . . . .	29
3.4.13 Interface de l'accueil . . . . .	30
3.4.14 Interface pour déposer de l'argent . . . . .	30
3.4.15 Interface pour afficher les messages . . . . .	31
3.4.16 Interface pour afficher les invitations . . . . .	31
3.4.17 Interface pour créer la carte bancaire . . . . .	32
3.4.18 Interface pour ouvrir un compte . . . . .	32
3.4.19 Interface pour activer un compte . . . . .	33
3.4.20 Interface pour désactiver un compte . . . . .	33

## Table des abréviations

J2EE	Java Platform Enterprise Edition
JSP	Java Server Pages
API	Application Programming Interface
MVC	Model View Controller
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
MD5	Message Digest 5
UML	Unified Modeling Language
JDBC	Java™ Database Connectivity
SQL	Structured Query Language
MYSQL	My Structured Query Language
HTTPS	HyperText Transfer Protocol Secure

# Introduction générale

Personne ne peut douter que l'informatique est une technologie fondamentale et innovante qui a eu un impact majeur sur la vie au cours du siècle dernier. En fait, loin d'être un phénomène bouillonnant ou une tendance dépassée, l'informatique vient d'être exploitée dans tous les aspects de la vie. Aucun domaine n'échappe à cette politique qui facilite la vie de l'entreprise et de ses salariés. Grâce à Internet, la communication et l'échange d'informations sont devenus plus faciles.

Le marché de la technologie a connu une évolution considérable ces dernières années ,et l'intégration d'Internet a changé la nature et la vitesse d'accès à l'information. Plusieurs secteurs se sont adaptés à ce changement et d'autres secteurs ont été créés pour accueillir ces nouveaux clients. La banque fait partie de ces domaines qui comptent un grand nombre de clients et qui souffre pour les satisfaire. Ils doivent alors fournir des services de manière plus rapide et plus sécurisée mais aussi à distance pour éviter l'engorgement des agences.

Avec l'avènement des appareils mobiles tels que les smartphones et les tablettes qui ont subi une révolution technologique, il est devenu plus facile pour une banque de gérer ce conflit en proposant des applications mobiles capables de fournir rapidement les services dont elles ont besoin.

Au Maroc, comme ailleurs dans le monde, avoir un compte bancaire est une nécessité, et avec l'essor d'internet, la relation banque/client connaît de nouvelles normes. L'application web va au-delà de la simple banque, les clients n'ont plus à attendre des heures pour les virements ou les dépôts, chaque client peut facilement consulter son solde, effectuer des virements, percevoir un salaire, imprimer ses coordonnées bancaires, commander des chéquiers et consulter l'historique de fonctionnement de son compte sans avoir à se déplacer en agence. En effet, avec un simple accès à distance, les opérations courantes sont possibles en plus de pouvoir accéder à tous les services bancaires 24h/24 et 7j/7, via un smartphone ou un ordinateur.

Pour cela, nous avons été amenées à réaliser dans le cadre de notre projet de fin d'étude une application e-banking sécurisée en J2EE qui offre de nombreuses fonctionnalités à l'utilisateur,ainsi que la partie administrative qui consiste à gérer les services clientèle.

Pour réaliser notre projet on va suivre la démarche suivante :

- Chapitre 1 : Le premier chapitre a pour but de mettre une vue globale sur le cadre général du projet.
- Chapitre 2 : Analyse des besoins et conception, cette partie consiste à collecter, analyser , définir les besoin et à modéliser l'application.
- chapitre est divisé en deux parties : la 1ère partie concerne la conception technique, une description détaillée des outils utilisés pour développer l'application . Et dans la 2ème partie nous allons décrire les technologies utilisées pour la partie sécurité . Et dans la 3ème partie nous allons décrire l'application réalisée.

# Chapitre 1

## Contexte et spécification des besoins

Avec l'avènement d'Internet, le développement Web fait référence au processus d'écriture d'un site ou d'une page Web dans un langage technique. Il s'agit d'une étape importante pour mettre votre contenu en ligne et atteindre les lecteurs. Le développement web repose sur l'écriture de programmes utilisant des langages (HTML/CSS, JavaScript, PHP,J2EE, etc...), qui sont ensuite exécutés par un ordinateur. Ces instructions sont définies sur Internet et exécutées sur le serveur.

Dans le cadre de notre projet ,nous avons travailler avec le langage J2EE , et pour une organisation rigoureuse et logique du code nous avons choisi Le MVC (design pattern) qui propose une solution au problème de la structuration d'une application.L'objectif général de MVC est de séparer les parties de traitement(contrôleur), de données(modèle) et de présentation (vue) et de définir l'interaction entre ces trois deniers.

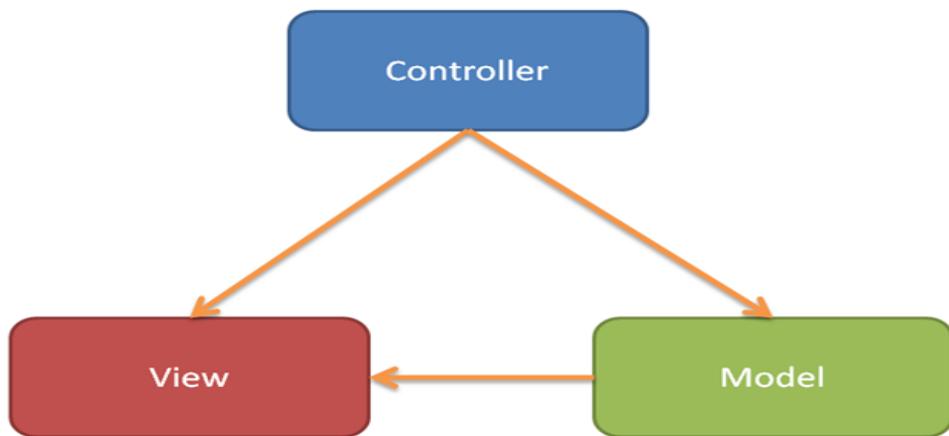


FIGURE 1.0.1 – Le modèle MVC.

Un **modèle** contient les données utilisées par un programme. Il peut s'agir d'une base de données, d'un fichier ou d'un simple objet. Par exemple, un objet Client récupérera les informations de la base de données, les manipulera et mettra à jour ses données dans la base de données.

Une **vue** est une moyenne d'afficher des objets dans une application. Par exemple, l'affichage d'une fenêtre ou des boutons ou d'un texte dans une fenêtre. Il comprend tout

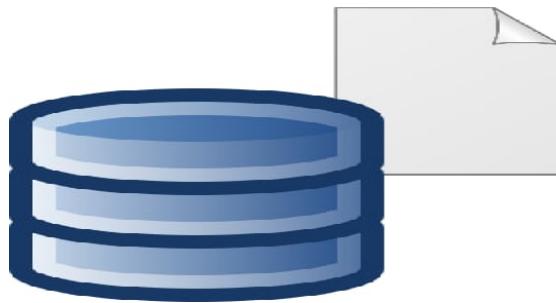


FIGURE 1.0.2 – Modèle.

ce que l'utilisateur peut voir.

La vue est l'interface utilisateur. La vue permet à l'utilisateur d'afficher les données à l'aide d'un modèle et lui permet également de modifier les données.

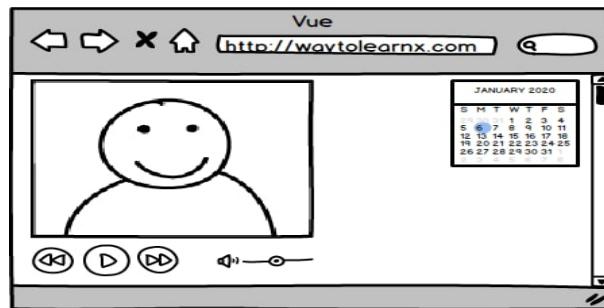


FIGURE 1.0.3 – Vue.

Le **contrôleur** agit comme une interface entre le modèle et la vue, pour traiter toute la logique métier et les requêtes entrantes, manipuler les données à l'aide du composant Modèle et interagir avec les Vues pour rendre le résultat final. Par exemple, le contrôleur 'Client' va traiter toutes les interactions et les entrées de la Vue 'Client' et mettre à jour la base de données en utilisant le Modèle 'Client'. Le même contrôleur sera utilisé pour visualiser les données du client.

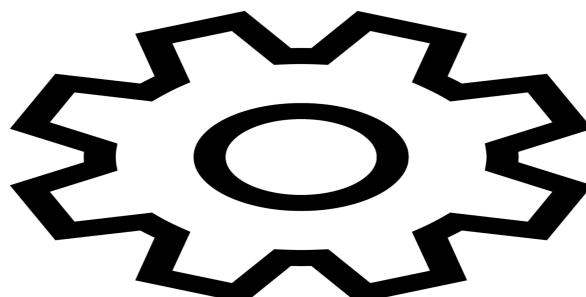


FIGURE 1.0.4 – Contrôleur.

Pour la vue nous avons utilisé les JSP qui permettent aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web ,mais pour la partie de traitement nous avons utilisé les servlets qui sont des programmes qui s'exécutent côté serveur,elle reçoit une requête du client, elle effectue des traitements et interagit avec la base de données via l'API JDBC est l'acronyme de Java DataBase Connectivity .

La sécurité des applications Web est essentielle pour protéger les données contre le vol , pour La sécurisation de notre application nous avons fait :

- L'authentification forte ou L'authentification à deux facteurs qui est largement utilisée dans les entreprises et les administrations pour protéger l'accès aux données sensibles et aux réseaux.



FIGURE 1.0.5 – Authentification forte.

- La fonction de hachage MD5 pour que les clients soient sûr que leurs mots de passe sont stockés de manière sécurisée dans la base de données.



FIGURE 1.0.6 – MD5.

- Le protocole HTTPS qui a pour but de chiffrer les données échangées entre le navigateur et le site web.

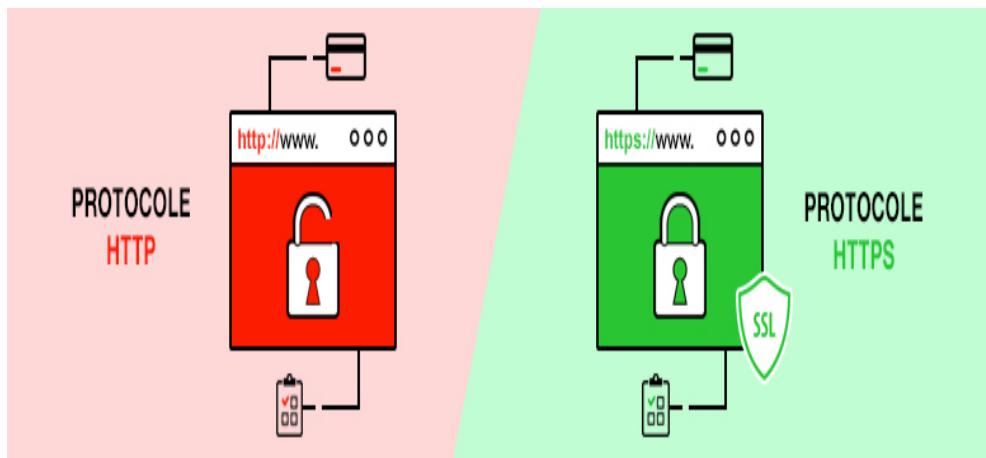


FIGURE 1.0.7 – Le protocole HTTPS.

- Assurer l'intégrité des données consiste à :toute entrée doit être vérifiée et validée et être cohérente , contrôler les permissions et les droits d'accès et de modification. De manière générale, l'intégrité des données désigne l'état de données lors de leur traitement, de leur conservation ou de leur transmission, ne subissent aucune destruction accidentelle.
- La confidentialité des données est la protection des communications ou des données stockées contre l'interception et la lecture par des personnes non autorisées.

## 1.1 Planification du projet

Nous avons utiliser le diagramme de Gantt (un outil de gestion de projet informatique) pour représenter les différentes activités et tâches que nous devons mettre en œuvre pour le développement de notre projet. Cet outil est important pour nous situer dans le temps et Prévoir le temps requis pour diverses activités.  
La colonne de gauche du diagramme énumère les tâches que nous devons accomplir,tandis que la ligne d'en-tête à droite définisse le temps nécessaire pour réaliser le projet.

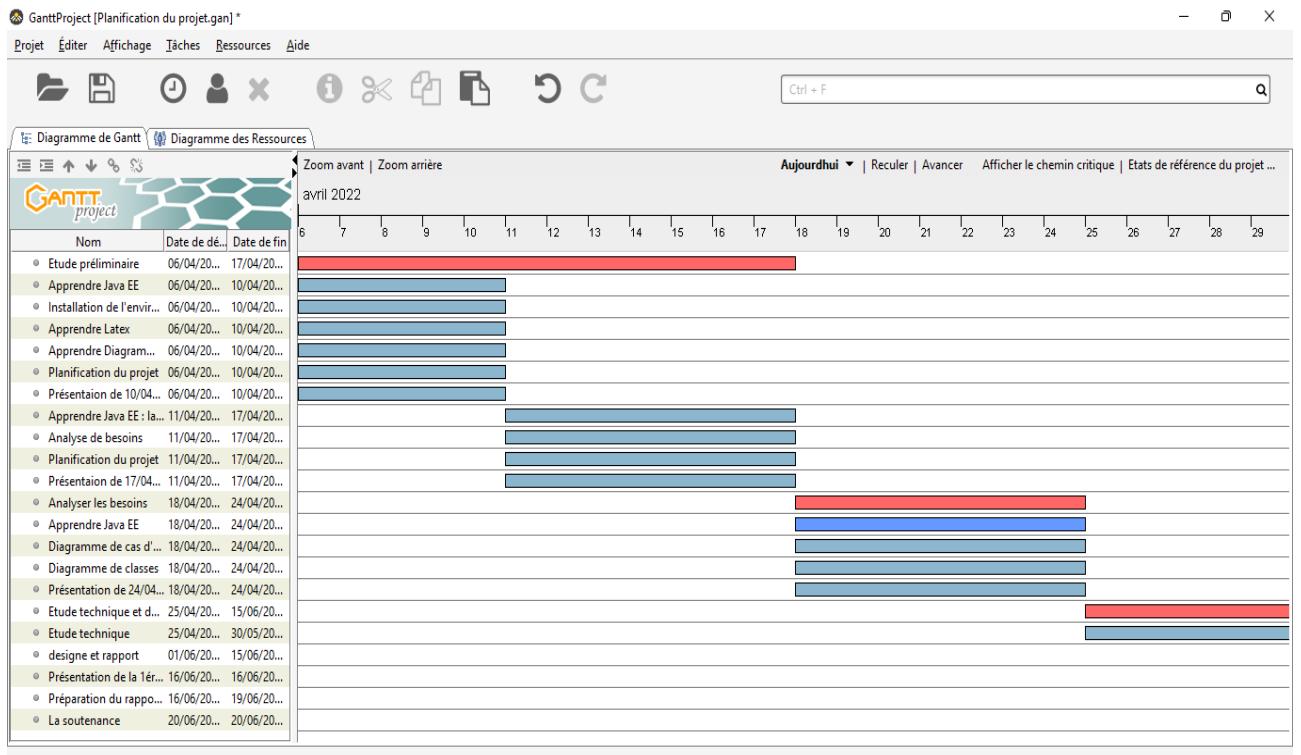


FIGURE 1.1.1 – diagramme de Gantt sur la planification du projet.

## 1.2 Conclusion

Nous avons présenté dans ce chapitre l'objectif principal de notre projet et aussi nous allons fait un diagramme de Gantt pour présenter le temps d'étude de chaque phase.

# Chapitre 2

## Analyse des besoins et conception

La phase d'analyse des besoins, est la première étape du processus de développement que nous avons adopté. En effet, elle permet de dégager l'étude fonctionnelle du système. Elle permet ainsi d'obtenir une idée sur ce que va réaliser le système en termes de métier (comportement du système). La phase suivante est la phase de conception. En fait, elle représente une étape importante de réflexion dans le cycle de développement logiciel après la phase de l'analyse et de spécification.

Tout au long de ce chapitre, on va définir les besoins de la solution que nous allons proposer, ensuite nous allons présenter en détails la conception du projet à travers les diagrammes UML.

### 2.1 Analyse des besoins

#### **Les besoins fonctionnels :**

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement du Système. Le système à concevoir doit permettre :

#### **A l'administrateur :**

- Ouvrir un compte.
- Editer un compte.
- Activer ou désactiver un compte.
- Traiter les demandes des clients(accepter ou refuser).
- Traiter les messages.
- Préparer la carte.
- Consulter les messages.
- Traiter les invitations.
- Ajouter un message.
- Effectuer des opérations (déposer ou retirer )
- Déconnexion

### Aux Clients :

- Créer un compte.
- Consulter profile.
- Déposer une réclamation.
- Ajouter codeSMI .
- Changer le codeSMI ou le mot de passe.
- Effectuer un virement.
- Effectuer des recharges.
- Demande d'ouvrir un compte épargne.
- Consulter l'historique des notifications et opérations.
- Consulter l'existance de la carte.
- Ajouter un bénéficiaire.
- Déconnexion

### Les besoins non fonctionnels :

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate aux attentes des concepteurs des architectures dynamiques.

Notre application doit nécessairement assurer ces besoins :

- **L'extensibilité** : dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- **La sécurité** : l'application devra être hautement sécurisée.
- **La performance** : l'application devra être performante et aussi simple à manipuler même par des non experts.

## 2.2 Conception

Avant de pouvoir programmer une application et commencer à écrire du code, vous devez d'abord organiser vos pensées puis organiser votre implémentation en définissant les phases de votre implémentation. Cette approche est appelée modélisation ou conception. La modélisation consiste à créer des représentations virtuelles de la réalité pour faire ressortir des points d'intérêt. Dans le cadre de notre projet, nous modélisons différents diagrammes en utilisant la méthodologie [UML].

**UML** (Unified Modeling Language) est un langage de modélisation unifié permettant de modéliser une application d'une façon standard dans le cadre de conception orienté objet.

### 2.2.1 Diagramme de cas d'utilisation

L'objectif de ce diagramme est la modélisation de la vue des utilisateurs et définir les fonctionnalités du système autrement dit la structuration des besoins.

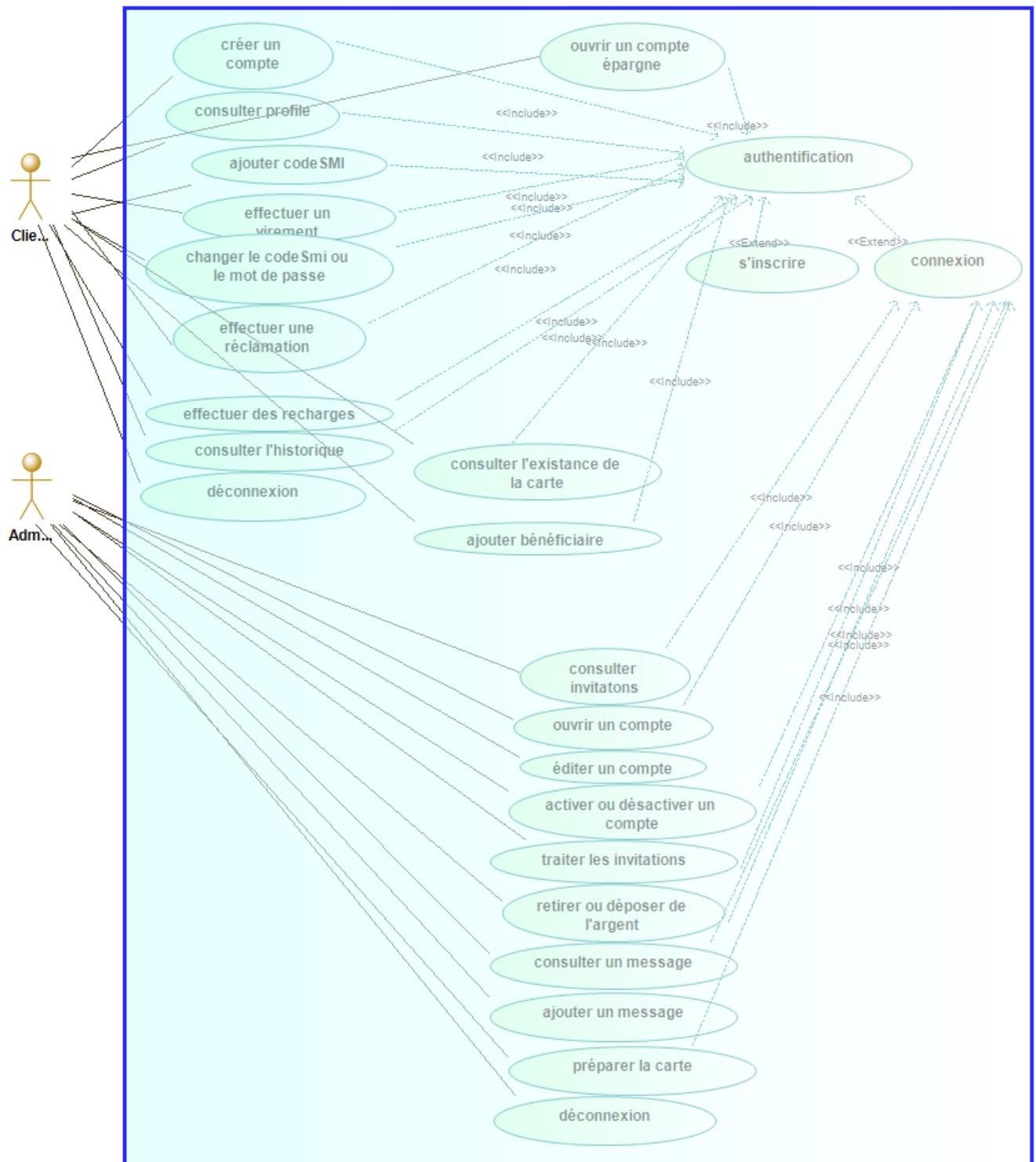


FIGURE 2.2.1 – Diagramme de cas d'utilisation

## 2.2.2 Diagramme de classe

Diagramme de Classe : Représentation abstraite des objets du système qui vont interagir ensemble pour la réalisation des cas d'utilisation.

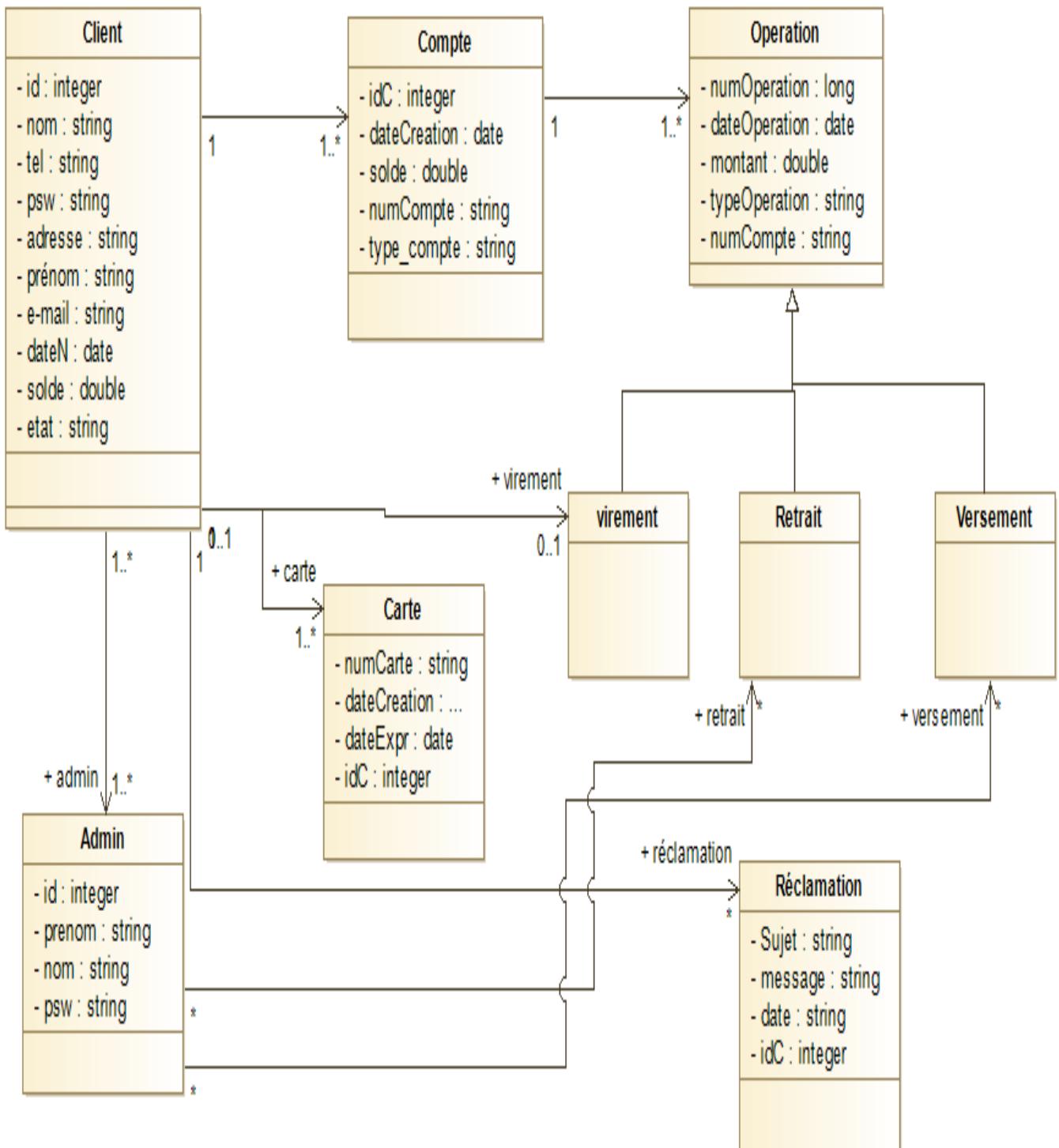


FIGURE 2.2.2 – Diagramme de classe

### 2.2.3 Diagramme de séquence

Les diagrammes de séquences décrivent la façon dont des groupes d'objets interagissent pour réaliser un comportement donné, et représentent un certain nombre d'objets

et les messages transmis entre ces objets pour la réalisation du cas d'utilisation.

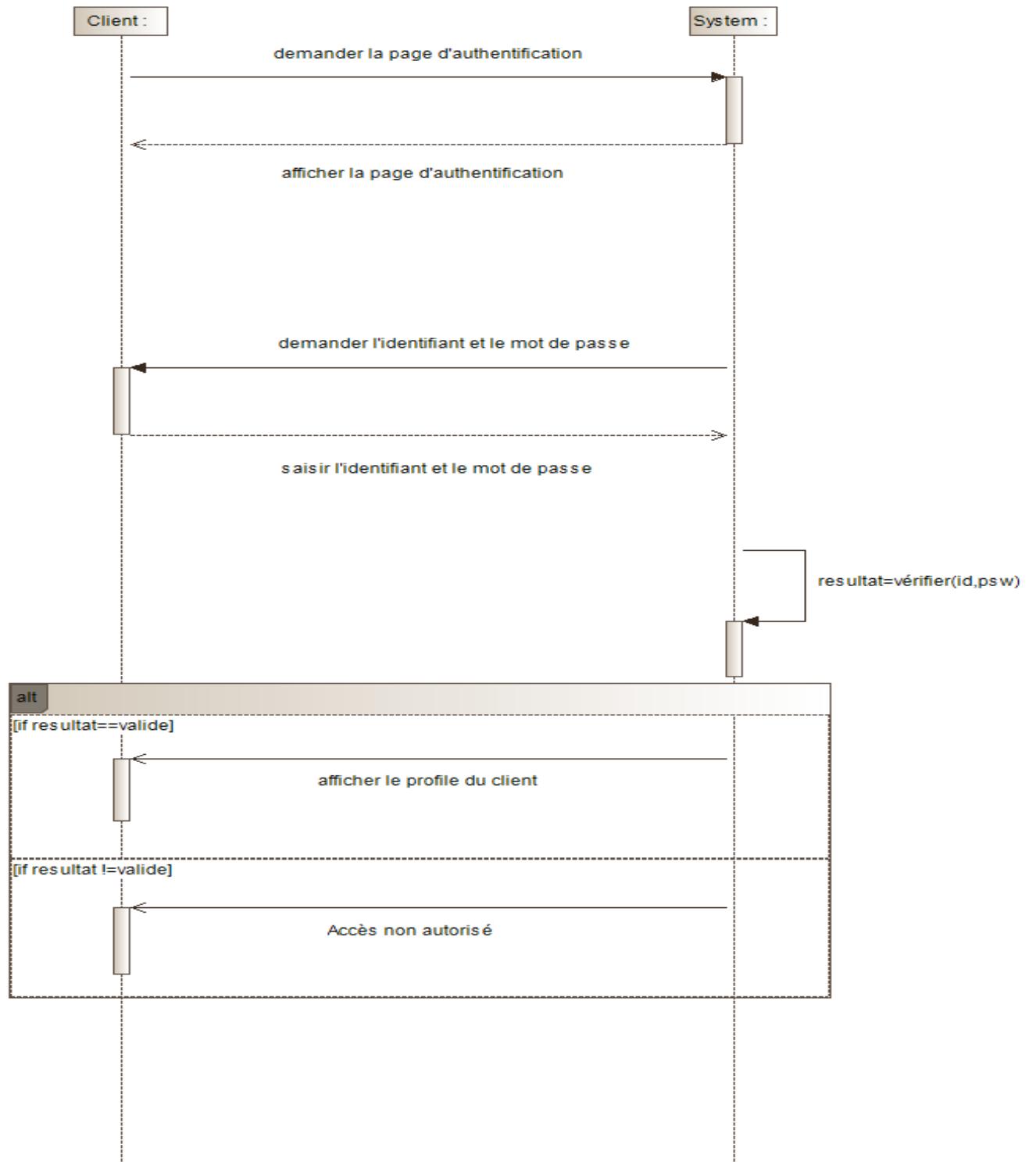


FIGURE 2.2.3 – Diagramme de séquence du procédure "authentification d'un client"

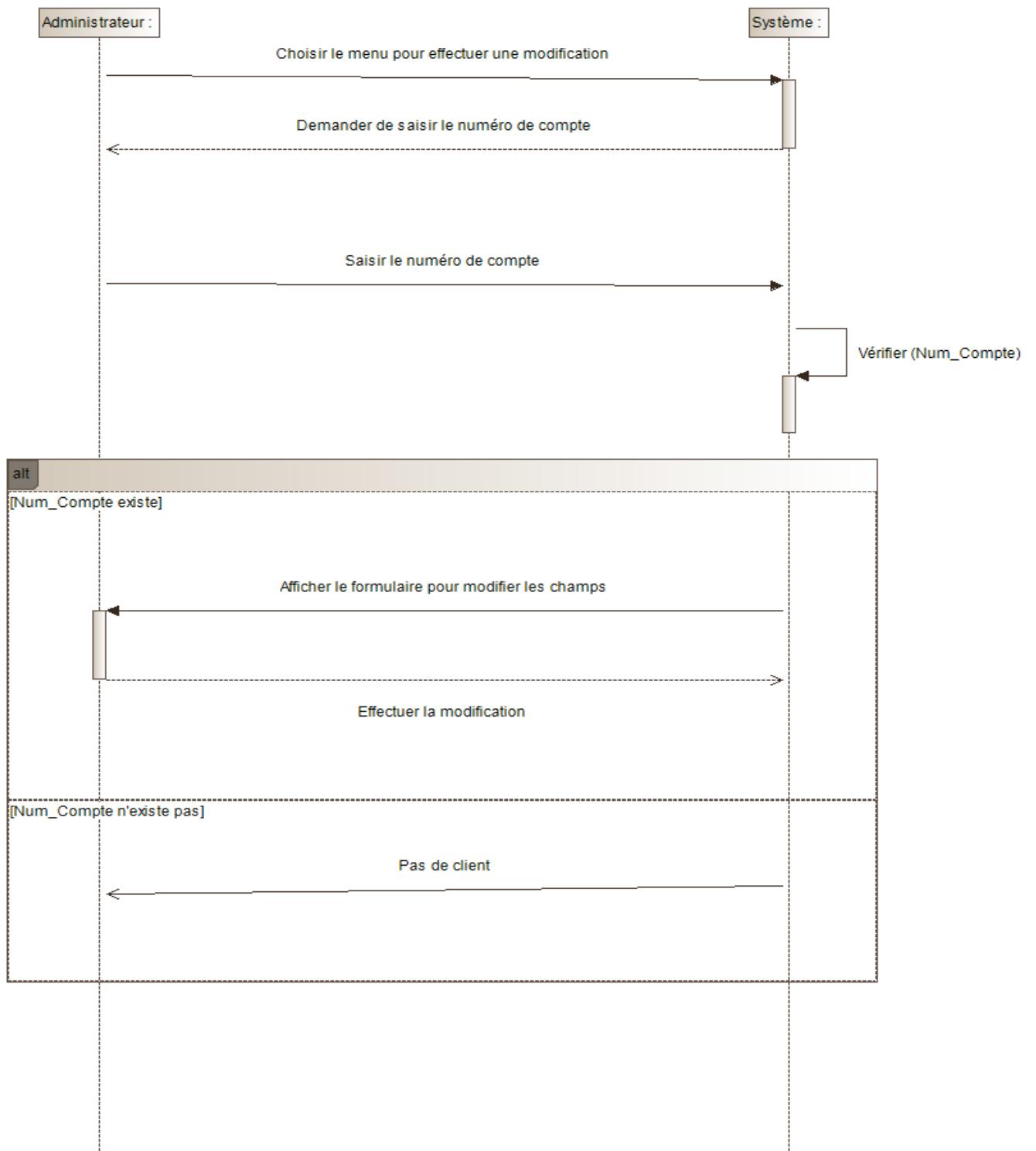


FIGURE 2.2.4 – Diagramme de séquence du procédé "Modification des coordonnées d'un client par l'administrateur"

## 2.2.4 Diagramme d'activités

Le diagramme d'activités permet de représenter le comportement interne d'un use case ou processus.

Représente le déroulement des traitements en les regroupant dans des étapes appelées « Activité » .

Activité : représentation d'un comportement paramétrable décrit par un séquencement d'actions.

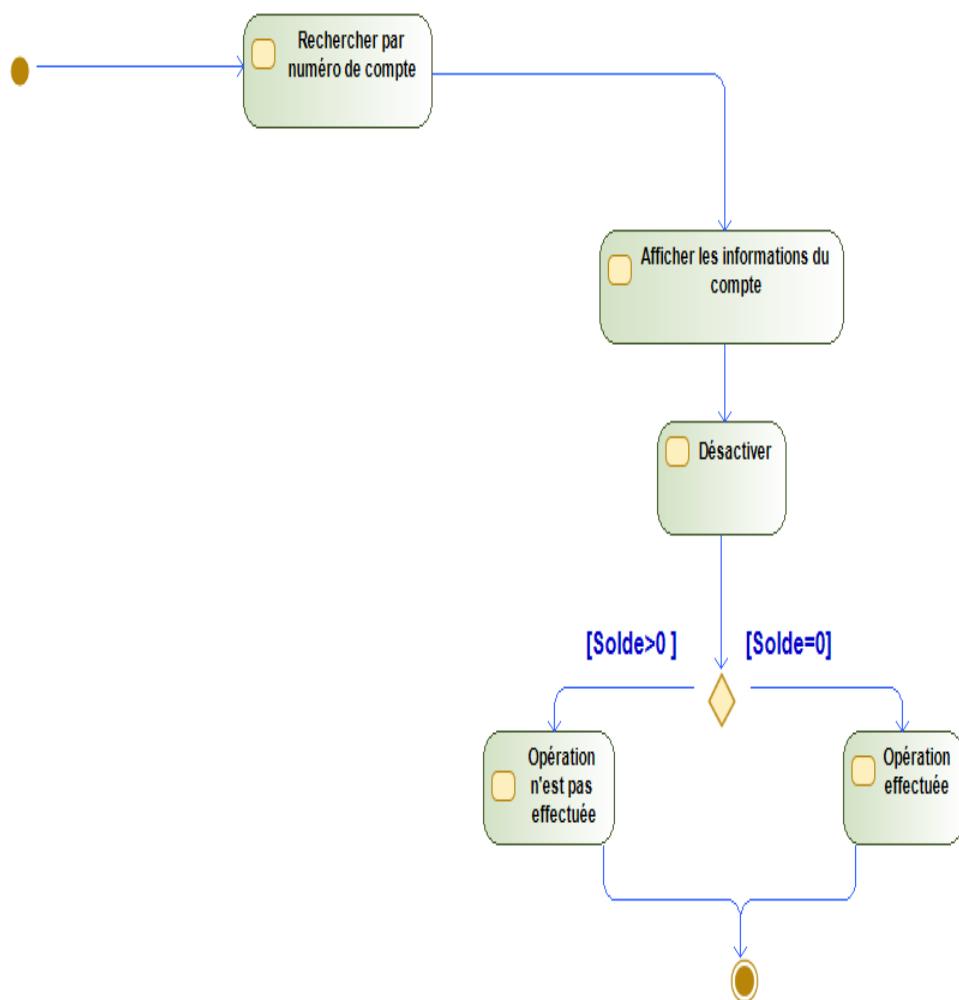


FIGURE 2.2.5 – Diagramme d'activités du procédure "Désactiver un compte par l'administrateur"

## 2.3 Conclusion

Ce chapitre présente une phase indispensable pour l'analyse et la conception de notre application. Nous avons défini les différents besoins fonctionnels et non fonctionnels, nous avons illustré la vue fonctionnelle de l'application par le diagramme de cas d'utilisation, ensuite nous avons réalisé le diagramme de classe et le diagramme de séquence afin de délimiter le cadre de notre travail et de préparer un terrain favorable pour la prochaine étape. Maintenant, notre application est prête à être codée. Dans le chapitre suivant, nous allons nous intéresser à l'implémentation de notre application en se basant sur la conception détaillée de ce chapitre.

# Chapitre 3

## Implémentation et réalisation

### 3.1 Environnement du travail

Nous allons énumérer au cours de cette partie les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

#### 3.1.1 Modelio

Lors de la phase de modélisation, nous avons utilisé Modelio comme outil, un outil de modélisation UML disponible sur les plateformes Windows, Linux et Mac, développé par SOFTEAM et distribué par Modeliosoft et Objecteering Software.

**Modelio** est un logiciel de modélisation UML très complet. À partir de la version gratuite, l'application prend en charge de nombreux diagrammes tels que les diagrammes de classe, de cas d'utilisation, d'état, de séquence ou d'objet. Son interface est claire et personnalisable grâce à des modules selon vos besoins. Par conséquent, cette application vous permettra de générer un grand nombre de diagrammes indispensables au développement d'applications.



FIGURE 3.1.1 – Modelio

### 3.1.2 GanttProject

**Ganttpoint** est un logiciel de gestion de projet gratuit disponible sur plusieurs systèmes d'exploitation : Windows, Mac et Linux. Il permet de planifier les projets dans le temps en créant des diagrammes de Gantt, qui sont des représentations visuelles de la répartition des tâches du projet sur des périodes longues, moyennes et courtes . Cela vous permet d'afficher les diverses tâches pour un projet au fil du temps, ainsi que d'afficher et de trier les dépendances des tâches. Le logiciel permet de répartir les activités par personne ou par "ressource". Peut également être utilisé pour gérer des projets de groupe.



FIGURE 3.1.2 – GanttProject

### 3.1.3 Eclipse



FIGURE 3.1.3 – eclipse.

**Eclipse** est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français).

Eclipse est un environnement de développement intégré pour les applications open source et multiplateformes. Il fonctionne principalement comme une plate-forme de programmation et peut compiler et déboguer de nombreux langages de programmation : bien qu'il soit surtout connu pour la programmation Java, sa modularité permet de l'utiliser

pour la programmation en C, Python, etc., et bien d'autres. Il est clairement encadré et organisé en plusieurs fenêtres différentes avec un grand degré de personnalisation. En plus de cela, bien que l'éditeur dispose de plusieurs outils intégrés pour le débogage et l'analyse des erreurs, eclipse compile automatiquement le code écrit, en soulignant en rouge les problèmes qu'il détecte. Il vous permet également d'utiliser des aperçus de projet afin que vous puissiez voir à quoi ressemblera votre application en Java sans compiler ni exécuter.

### 3.1.4 MySQL

**MySQL** est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle.

**MySQL** C'est l'un des logiciels de gestion de base de données les plus utilisés au monde. Il s'agit d'un serveur de base de données qui stocke les données dans des tables séparées . Cela améliore la vitesse et la flexibilité globales. Les tables sont reliées par des relations définies, qui permettent la combinaison de données entre plusieurs tables lors de la requête. MySQL peut être répliqué, de sorte que la charge peut être répartie sur plusieurs machines, Optimisez les performances ou effectuez facilement des sauvegardes de données.



FIGURE 3.1.4 – MySQL

### 3.1.5 ApacheTomcat

**Apache Tomcat** est un serveur web open source développé en Java qui vous offre un environnement 100% Java pour exécuter vos applications web. Contrairement aux serveurs HTTP qui affichent des pages Web aux utilisateurs , Apache Tomcat fournit une prise en charge des servlets et des JSP pour vos applications Web. Avec cet outil, vous pouvez créer et ajouter du contenu dynamique à votre serveur Web. Apache Tomcat utilise le

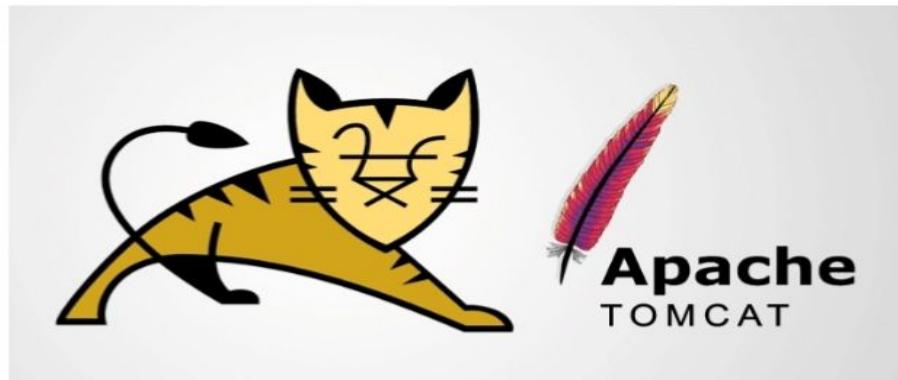


FIGURE 3.1.5 – ApacheTomcat

conteneur de servlet Catalina, qui interagit directement avec l’application Java et garantit que les droits d’accès des utilisateurs sont maintenus pendant la session.

## 3.2 Les technologies utilisées

Nous avons présenté dans ce chapitre les différentes technologies qu’on va utiliser pour réaliser notre application.

### 3.2.1 Java EE



FIGURE 3.2.1 – JAVA EE

Aujourd’hui, **Java EE** est la plate-forme de choix pour les grandes entreprises pour développer des sites web solide, stable et bien construit, Il est notamment très utilisée dans la finance. Par exemple, il y a de fortes chances que votre banque l’utilise, ou encore même l’Etat, parce qu’on a confiance dans cette plateforme, et elle a acquis une très grande maturité professionnelle. Concrètement, qu’est ce que Java EE ?

Java EE est basée sur un langage de développement qui est aujourd’hui bien connu et

bien établi, notamment dans le monde des entreprises. C'est Java.

Java permet de développer des applications .Par dessus Java, On a rajouté un ensemble de bibliothèques qui ajoutent en réalité des fonctionnalités à Java et c'est Java EE.

### 3.2.2 Htm



FIGURE 3.2.2 – HTML

L'Hypertext Markup Language généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

### 3.2.3 CSS



FIGURE 3.2.3 – CSS

**CSS** (Cascading Style Sheet ou Feuilles de style en cascade) est une norme et une recommandation de W3C. La première version CSS1 a été créée en 1996 et la deuxième

CSS2 en 1998, CSS2.1 en 2004 (avec environ 70 nouvelles propriétés) . CSS3 en cours de finalisation. Le principe d'une feuille de style CSS est de séparer la mise en forme du contenu de document. Les instructions CSS de mise en forme sont rassemblées au début de document HTML dans la balise ... ou carrément dans un fichier à part. Dans une feuille de style CSS nous décrivons la mise en forme des balises HTML..

### 3.2.4 JavaScript

**JavaScript** est un langage de script principalement utilisé pour les pages Web interactives et constitue donc une partie essentielle des applications Web. Avec HTML et CSS, JavaScript est au cœur du langage utilisé par les développeurs web. La grande majorité des sites Web l'utilisent, et la plupart des navigateurs Web disposent d'un moteur JavaScript pour l'interpréter. JavaScript est également utilisé dans les serveurs Web, par exemple avec Node.js ou Deno. JavaScript a été créé par Brendan Eich en 1995 et intégré au navigateur Web Netscape Navigator 2.0. L'implémentation JavaScript concurrente de Microsoft dans Internet Explorer s'appelle JScript, tandis que l'implémentation d'Adobe Systems s'appelle ActionScript. En juin 1997, Ecma International a normalisé JavaScript en tant qu'ECMAScript dans la norme ECMA-262. La version de cette norme mise en place depuis juin 2020 est la 11ème édition.



FIGURE 3.2.4 – JavaScript

### 3.2.5 SQL

**SQL** (Structured Query Language) est un langage utilisé pour communiquer avec des bases de données. Ce langage informatique est particulièrement utilisé par les développeurs web pour communiquer avec les données des sites web. Il peut notamment stocker, manipuler et récupérer ces données. Vous pouvez également effectuer des requêtes, mettre à jour des données, réorganiser des données ou créer et modifier le schéma et la structure d'un système de base de données et contrôler l'accès à ses données.



FIGURE 3.2.5 – SQL

### 3.3 Outils utilisés pour la partie sécurité

Nous avons présenté dans cette partie les différentes technologies qu'on va utiliser pour sécurisé notre application.

#### 3.3.1 Fonction hachage : MD5

Pour que le mot de passe utilisé par chaque client serait crypté dans la base de données ,même si l'administrateur lui-même ne peut pas déchiffré les mots de passe de ces clients , nous avons exprimé la méthode MD5.

**MD5** pour Message Digits 5 est une fonction de hachage cryptographique qui vous permet de "hacher" (chiffrer) une séquence de nombres dans un hachage md5 de 128 bits ou 32 caractères, il est impossible d'obtenir la séquence originale (déchiffrement) en utilisant uniquement le hachage md5. Ainsi, la seule façon de déchiffrer le hachage est de le comparer à une base de données contenant des hachages md5 en ligne et leurs séquences correspondantes.

la figure suivante montre le code utilisé pour implémenter la fonction qui fait le cryptage du code.

```

eclipse-workspace - SMIBANQUE/src/main/java/com/bd/conBD.java - Eclipse IDE
File Edit Refactor Source Navigator Search Project Run Window Help
[admin.java] [conBD.java] [rechercherE...]
1771}
1772}
1773
1774 public void ajouteClientA(Client Client) {
1775     loadDatabase();
1776
1777     try {
1778         // Create MessageDigest instance for MD5
1779         MessageDigest md = MessageDigest.getInstance("MD5");
1780
1781         // Add password bytes to digest
1782         md.update(passwordToHash.getBytes());
1783
1784         // Get the hash's bytes
1785         byte[] bytes = md.digest();
1786
1787         // This bytes[] has bytes in decimal format. Convert it to hexadecimal format
1788         StringBuilder sb = new StringBuilder();
1789         for (int i = 0; i < bytes.length; i++) {
1790             sb.append(Integer.toString((bytes[i] & 0xFF) + 0x100, 16).substring(1));
1791         }
1792
1793         // Get complete hashed password in hex format
1794         generatedPassword = sb.toString();
1795     } catch (NoSuchAlgorithmException e) {
1796         e.printStackTrace();
1797     }
1798
1799     try {
1800

```

FIGURE 3.3.1 – Message Digit 5

### 3.3.2 FireBase : Authentification forte

Si un utilisateur qui utilise n'importe quelle application oublie le mot de passe pour se connecter à son propre compte , l'application doit forcément contenir une solution qui gère ce problème , c'est pour ça on a choisi d' utiliser **Firebase** Authentication pour connecter un client en lui envoyant un SMS sur son téléphone. Le client se connecte à l'aide d'un code contenu dans le message SMS.

**Firebase** Authentication est conçu pour faciliter la création de systèmes d'authentification sécurisés tout en améliorant les expériences de connexion et d'intégration des utilisateurs . Il fournit une solution d'identité de bout en bout qui prend en charge les comptes de messagerie et de mot de passe, l'authentification du téléphone et les connexions Google, Twitter, Facebook et GitHub, et plus encore.

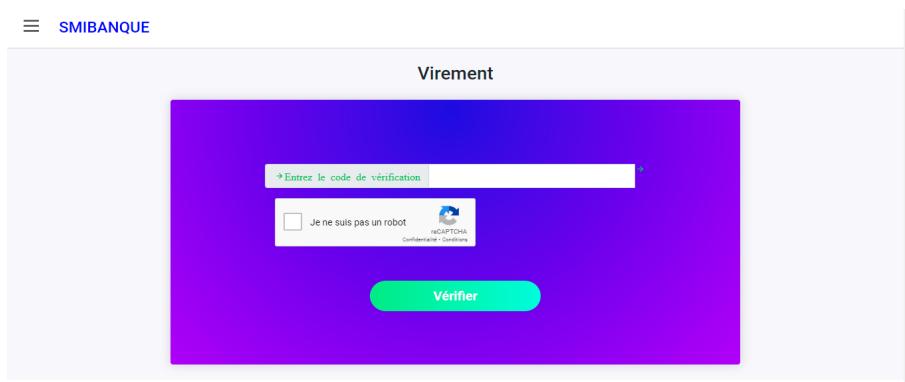


FIGURE 3.3.2 – Authentification forte

Vous devez suivre les étapes suivantes pour avoir faire une authentification forte par numéro de téléphone .

```
eclipse-workspace : SMIBANQUE/src/main/webapp/WEB-INF/connection.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
[1] admin.jsp [2] connection.jsp [3] recherche.jsp [4] admin.java [5] versement.jsp [6] versement.java [7] index.java [8] firebase.jsp [9] connection.jsp
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title></title>
5
6 </head>
7 <body>
8 <form>
9<div class="formcontainer">
10<div class="container">
11<div id="recaptcha-container"></div>
12
13<script>
14 window.onload=function(){
15     render();
16 }
17 }
18 function render(){
19
20     window.recaptchaVerifier=new firebase.auth.RecaptchaVerifier('recaptcha-container');
21     recaptchaVerifier.render();
22     phoneAuth();
23 }
24
25 function phoneAuth(){
26     var phone_numbers=$("#num");
27     firebase.auth().signInWithPhoneNumber(phone_number,window.recaptchaVerifier).then(function(confirmationResult) {
28         // SMS sent. Prompt user to type the code from the message, then sign
29         // them in with confirmationResult.confirm(code).
30         window.confirmationResult = confirmationResult;
31         coderesult=confirmationResult;
32     })
33 }
34
35 
```

FIGURE 3.3.3 – Authentification forte :partie1

```

eclipse-workspace : SMIBANQUE/src/main/webapp/WEB-INF/connection.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
admin.java conDJava rechercher... admin.jsp versement.jsp versement.java index.java client.java firebase.jsp connection.jsp
25 function phoneAuth(){
26     var phone_number=$("#num");
27     firebase.auth().signInWithPhoneNumber(phone_number,window.recaptchaVerifier).then(function(confirmationTokenResult) {
28         // Ask the user to type the code from the message, then sign the user in with confirmationResult.confirm(code).
29         window.confirmationTokenResult = confirmationTokenResult;
30         window.confirmationResult = confirmationResult;
31         coderesult=confirmationResult;
32         coderesult.confirmationResult;
33         alert('message sent');
34         // ...
35     }).catch(function(error){
36         // Error; SMS not sent
37         alert(error.message);
38     });
39     //alert(phone_number);
40 }
41 </script>
42 <script src="https://www.gstatic.com/firebasejs/8.3.2.firebaseio.js"></script>
43 <script>
44 var firebaseConfig = {
45   apiKey: "AIzaSyBtWpP4Tl1nUWwYmo2NH5L1uscP0C3JLSY",
46   authDomain: "login-jzee.firebaseioapp.com",
47   projectId: "login-jzee",
48   storageBucket: "login-jzee.appspot.com",
49   messagingSenderId: "574452488450",
50   appId: "1:574452488450:web:0a7d8d75293a8ba1680a1b",
51   measurementId: "G-MHINC9H93T"
52 };
53
54 };
55

```

FIGURE 3.3.4 – Authentification forte :partie2

```

eclipse-workspace : SMIBANQUE/src/main/webapp/WEB-INF/connection.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
admin.java conDJava rechercher... admin.jsp versement.jsp versement.java index.java client.java firebase.jsp connection.jsp
55
56 // Initialize Firebase
57 firebase.initializeApp(firebaseConfig);
58 firebase.analytics = getAnalytics();
59 </script>
60 </div>
61 </div>
62 </form>
63
64
65
66
67</form >
68 <input type="text" id="verifycode" name="uname" required><td>
69 <button type="button" onclick="verify();>submit</button>
70 </td>
71 </form>
72
73 <form action="SMT_CODE1">
74 <input type="submit" name="submit">
75 </form>
76
77
78<script>
79 function verify(){
80     var codedocument.getElementById('verifycode').value;
81     coderesult.confirm(code).then(function(result){
82         // User signed in successfully.
83         const user = result.user;
84         console.log(user);
85         alert('successfully registered');
86     var r=confirm("Pour finir votre operation cliquez sur OK");
87     if(r==true){
88         var s=<br><p><h1 style='color:red'><ul><li>Important !</ul><h2 style='color:white'><ul>les étapes à suivre pour modifier
89         document.write(<html><head><style> body{background-color:purple;}input {background-color: black; /* Green */border: none;color
90         >
91     }
92
93
94
95
96     }).catch(function(error){
97         alert(error.message);
98         // User couldn't sign in (bad verification code)
99     });
100 });
101
102 </script>
103
104
105 </body>
106 </html>
107

```

FIGURE 3.3.5 – Authentification forte :partie3

```

eclipse-workspace : SMIBANQUE/src/main/webapp/WEB-INF/connection.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
admin.java conDJava rechercher... admin.jsp versement.jsp versement.java index.java client.java firebase.jsp connection.jsp
77
78<script>
79 function verify(){
80     var codedocument.getElementById('verifycode').value;
81     coderesult.confirm(code).then(function(result){
82         // User signed in successfully.
83         const user = result.user;
84         console.log(user);
85         alert('successfully registered');
86     var r=confirm("Pour finir votre operation cliquez sur OK");
87     if(r==true){
88         var s=<br><p><h1 style='color:red'><ul><li>Important !</ul><h2 style='color:white'><ul>les étapes à suivre pour modifier
89         document.write(<html><head><style> body{background-color:purple;}input {background-color: black; /* Green */border: none;color
90         >
91     }
92
93
94
95
96     }).catch(function(error){
97         alert(error.message);
98         // User couldn't sign in (bad verification code)
99     });
100 });
101
102 </script>
103
104
105 </body>
106 </html>
107

```

FIGURE 3.3.6 – Authentification forte :partie4

### 3.3.3 Le protocole Https

#### Pourquoi utiliser un protocole HTTPS ?

Le protocole **HTTPS** « Hypertext Transfer Protocol Secure » est une extension sécurisée du protocole http, ce protocole a été créé pour sécuriser davantage les sites Web qui souhaitent protéger et crypter (confidentialité) les données échangées entre le navigateur de l'internaute et le site web.

Le protocole HTTPS utilise un protocole de transport particulier(SSL/TLS).

SSL signifie "Secure Sockets Layer" et TLS signifie "Transport Layer Security" sont des protocoles de cryptage au niveau de la couche de transport Internet, les deux protocoles s'occupent de chiffrer le flux de données entre le client et le serveur. Lorsqu'un site Web souhaite communiquer en https ,il doit avoir une certificat SSL.

Les certificats SSL sont des fichiers de données qui lient les clés de chiffrement aux informations organisationnelles. Le certificat installé sur le serveur et active le protocole "https" pour assurer une connexion sécurisée entre le serveur web et le navigateur. SSL est couramment utilisé pour protéger les transactions bancaires, les transferts de données et les informations de connexion. Il est récemment devenu la norme pour une navigation sécurisée sur les sites de réseaux sociaux.

Pour résumer on peut dire que le HTTPS remplit deux fonctions :

- La communication entre le client Web et le serveur Web est cryptée. Cela permet d'empêcher des tiers non autorisés d'écouter les communications.
- Le serveur web est authentifié par le fait qu'au début de la communication, un certificat est envoyé au client web pour prouver la fiabilité du domaine. Cette mesure aide à lutter contre la tromperie causée par de faux sites Web.

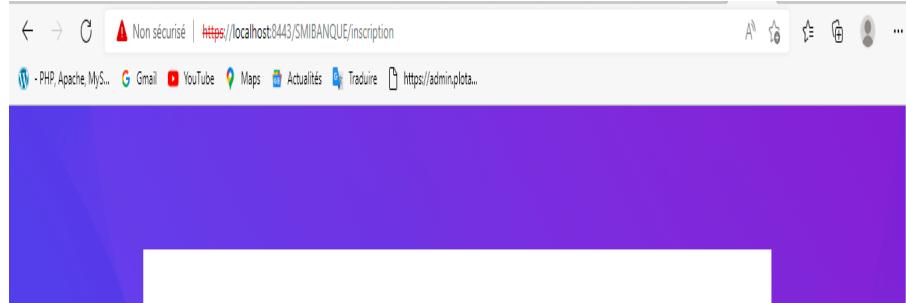


FIGURE 3.3.7 – Activation du protocole https

### 3.3.4 Conclusion

Dans cette partie, nous avons présenté les outils d'implémentations et les langages que nous avons utilisés , nous avons également traité les différents technologies que nous avons appliqué dans notre projet pour bien sécurisé l'application.

## 3.4 Les interfaces graphiques

Dans ce dernier chapitre nous allons présenter notre travail avec des captures d'écran de l'application réalisée.

Côté Client :



FIGURE 3.4.1 – Interface d'accueil

L'interface ci-dessus représente la page d'accueil de l'application, lors du clic sur le bouton démarrer, une interface d'inscription est affichée. L'utilisateur doit introduire ses coordonnées dans les champs correspondants .

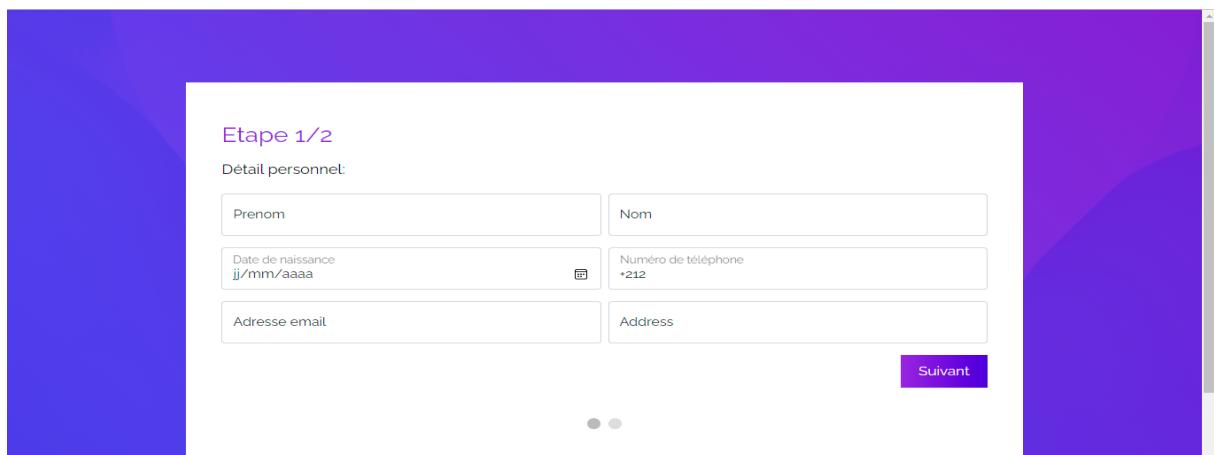


FIGURE 3.4.2 – 1 ère interface d'inscription

Etape 2/2

Créer un mot de passe

Entrez le mot de passe

Confirmer le mot de passe

Soumettre

FIGURE 3.4.3 – 2 ème interface d’inscription

lors du clic sur le bouton 'connectez vous', une interface d'authentification est affichée. L'utilisateur doit introduire son identifiant et son mot de passe dans les champs correspondants pour pouvoir accéder aux différentes fonctionnalités de l'application.

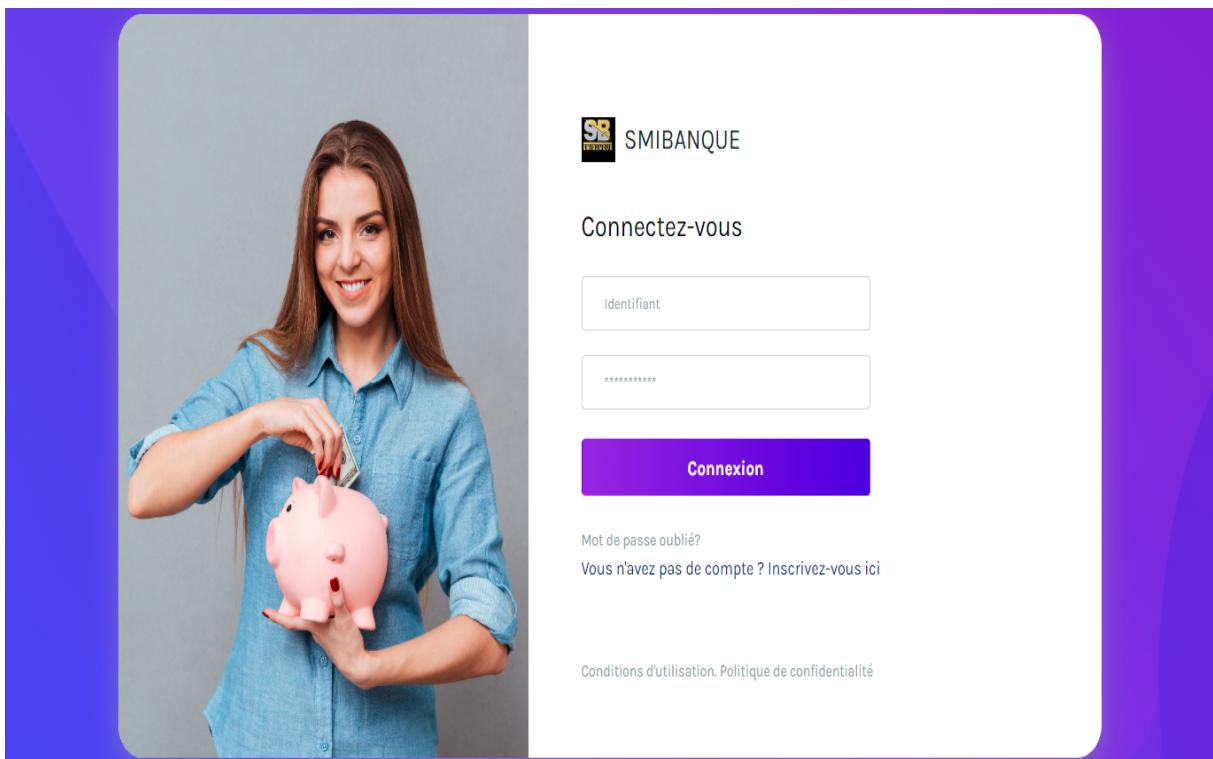


FIGURE 3.4.4 – Interface d’authentification

Une fois l'utilisateur a cliqué sur le bouton 'connexion', le système vérifie les données entrées. En cas d'échec, il affiche un message d'erreur. Si l'identifiant et le mot de passe

sont valides, le système redirige l'utilisateur vers son espace, où il va trouver l'interface ci-dessous.

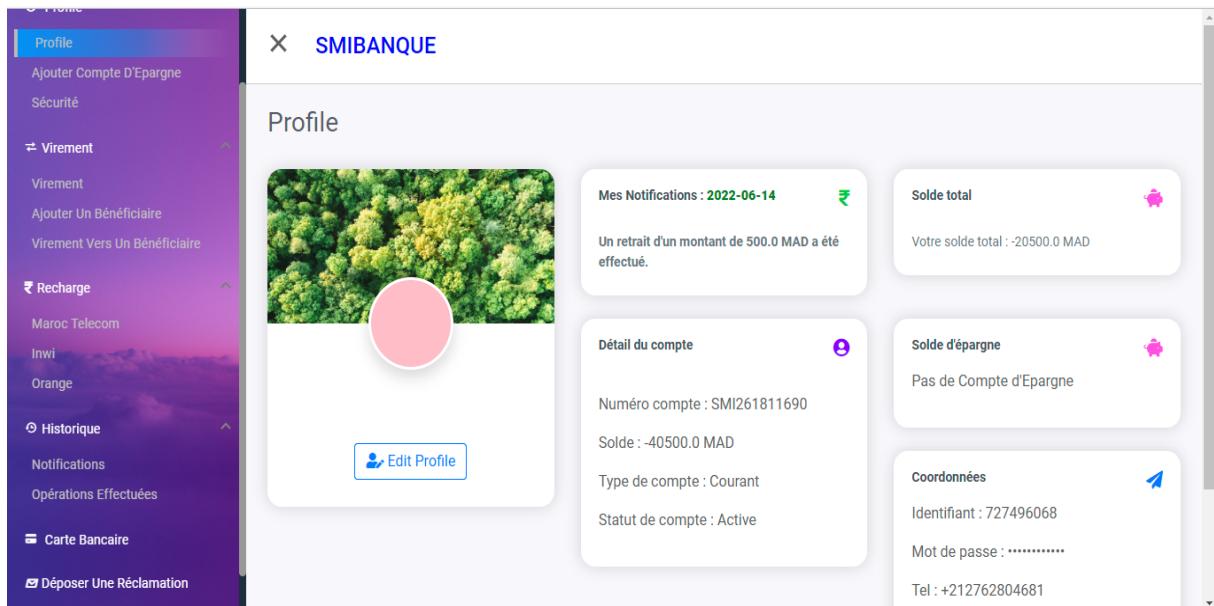


FIGURE 3.4.5 – Menu et profile de l'application

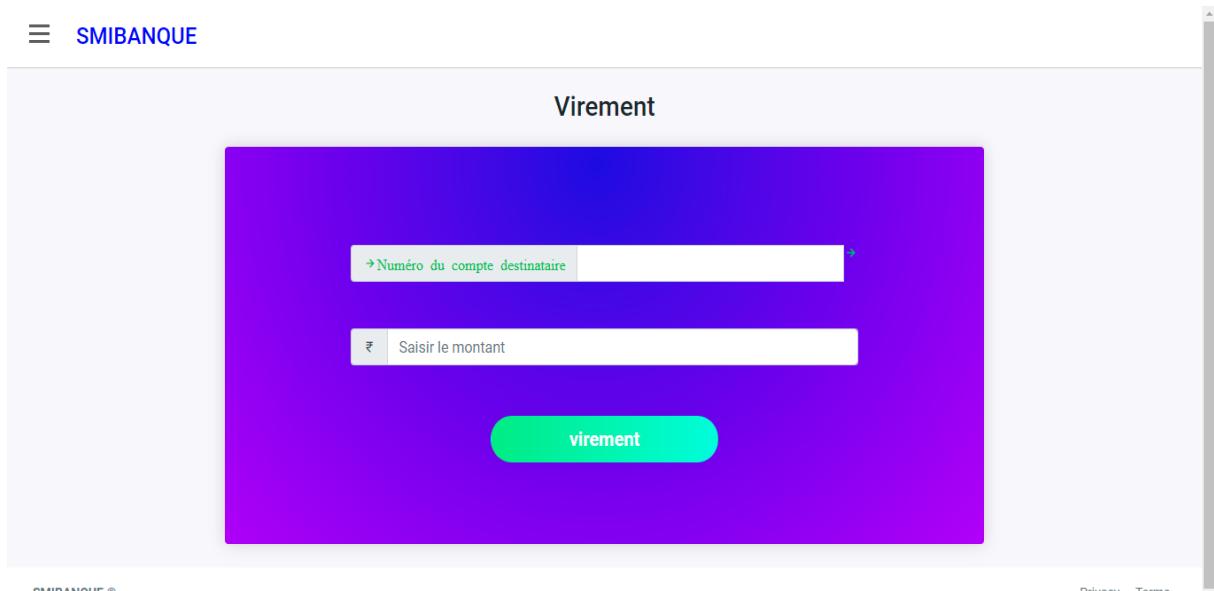


FIGURE 3.4.6 – Interface de virement normal

L'utilisateur peut faire un virement normal via l'interface ci-dessus, ou effectuer un virement vers un bénéficiaire via l'interface ci-dessous.

☰ SMIBANQUE

### Virement

Nom du bénéficiaire: Omar Kihel

Saisir le montant

virement

SMIBANQUE ©

Privacy Terms

FIGURE 3.4.7 – Interface de virement vers un bénéficiaire

☰ SMIBANQUE

### Déposer Une Réclamation

Sujet

Message

Envoyer message

FIGURE 3.4.8 – Interface pour déposer une réclamation

Lors du clic sur 'l'historique' dans le menu,nous avons trouvé deux choix soit on consulte les notifications ou bien l'historique des transactions .

Historique des Notifications	
Aperçu les Notifications de tous les temps	
Notification/th>	Date
Un versement d'un montant de 1000.0 MAD a été effectué.	2022-06-14
Un retrait d'un montant de 500.0 MAD a été effectué.	2022-06-14
Un virement d'un montant de 10000.0 MAD a été effectué.	2022-06-13
Un virement d'un montant de 1000.0 MAD a été effectué.	2022-06-13
Un virement d'un montant de 400.0 MAD a été effectué.	2022-06-12
Un virement d'un montant de 100.0 MAD a été effectué.	2022-06-11

FIGURE 3.4.9 – Interface qui affiche l'hisorique des notifications

Historique des transactions			
Aperçu de la transaction de tous les temps			
Numéro d'opération	Date d'opération	Montant	Type d'opération
17	2022-06-14	500.0 MAD	Retrait
16	2022-06-14	20000.0 MAD	Retrait
15	2022-06-14	1000.0 MAD	Versement

FIGURE 3.4.10 – .Interface qui affiche l'hisorique des opérations

Chaque utilisateur peut changer leur mot de passe ,ajouter ou modifier le codeSMI via cette interface.

## ☰ SMIBANQUE

### Sécurité

Ajoutez votre SmiCode

ajoutez votre SmiCode

Ajouter

Changez votre mot de passe

Entrez votre mot de passe actuel

Entrez votre nouveau mot de passe

Confirmez votre mot de passe

Modifier

Changez votre SmiCode

entrez votre code actuel

entrez votre nouveau code

Confirmez votre code

Modifier

FIGURE 3.4.11 – .Interface de sécurité

**Carte Bancaire**

Bonjour **Mohammed Kihel**,  
Votre Carte bancaire est disponible,Veuillez aller au votre agence **SMIBANQUE** pour la récupérer.

Merci,

**SMIBANQUE**

**Votre carte bancaire**

SMIBANQUE

Visa

2931 1192 1439 3522

Mohammed Kihel 06/32

**Conseils**

**1 - Assurer la sécurité de votre carte bancaire et de son code confidentiel :**

- Apprenez par cœur votre code secret.
- N'écrivez jamais votre numéro de code confidentiel, même "camouflé" sous un faux numéro de téléphone. Ne le confiez à personne : ni à votre banque, ni à votre assurance, ni aux services de police.
- Conservez votre carte bancaire dans un lieu sûr. Ne la laissez jamais dans votre voiture. Ne la laissez pas traîner chez vous ou sur votre lieu de travail afin

**2 - Assurez le suivi des utilisations de votre carte bancaire :**

- Conservez soigneusement vos tickets de retrait aux DAB/GAB ou facturettes jusqu'au reçu du relevé de compte débitant le solde des montants correspondants.
- Conservez les relevés de compte pendant 13 mois ou

FIGURE 3.4.12 – .Interface qui affiche les informations concernant la carte bancaire

### Côté Administrateur :

- Cette interface est l'accueil de l'admin où nous avons trouvé le nombre d'invitations, de clients et de messages,et également des boutons pour déposer , retirer de l'argent.

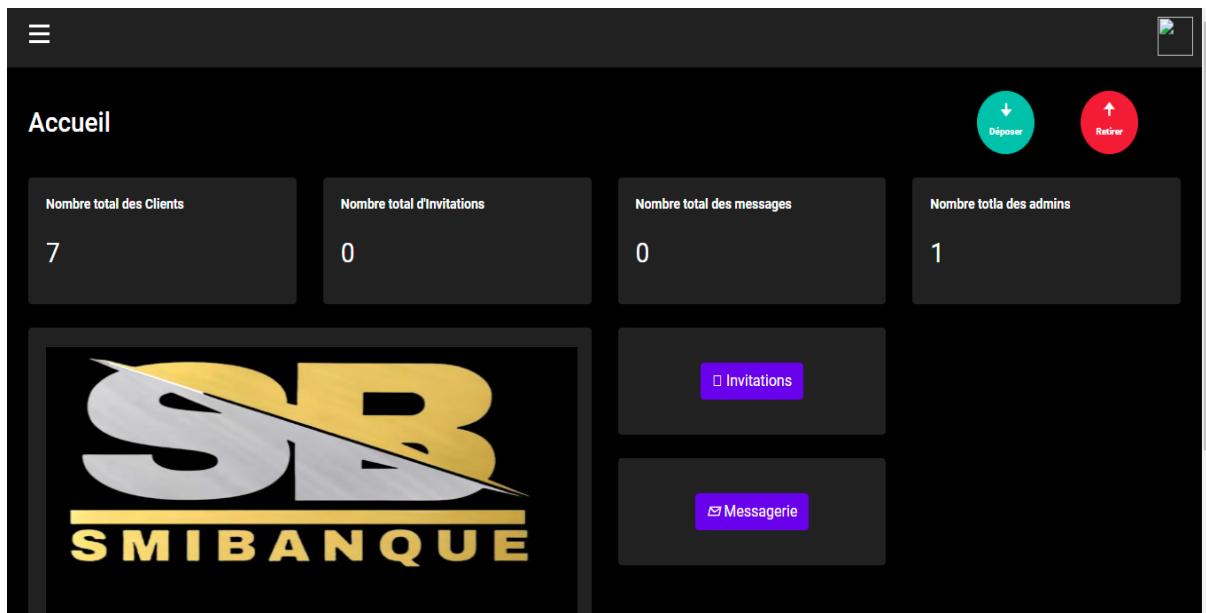


FIGURE 3.4.13 – .Interface de l'accueil

Parmi les tâches que l'administrateur peut faire :

- Effectuer un versement via l'interface suivante

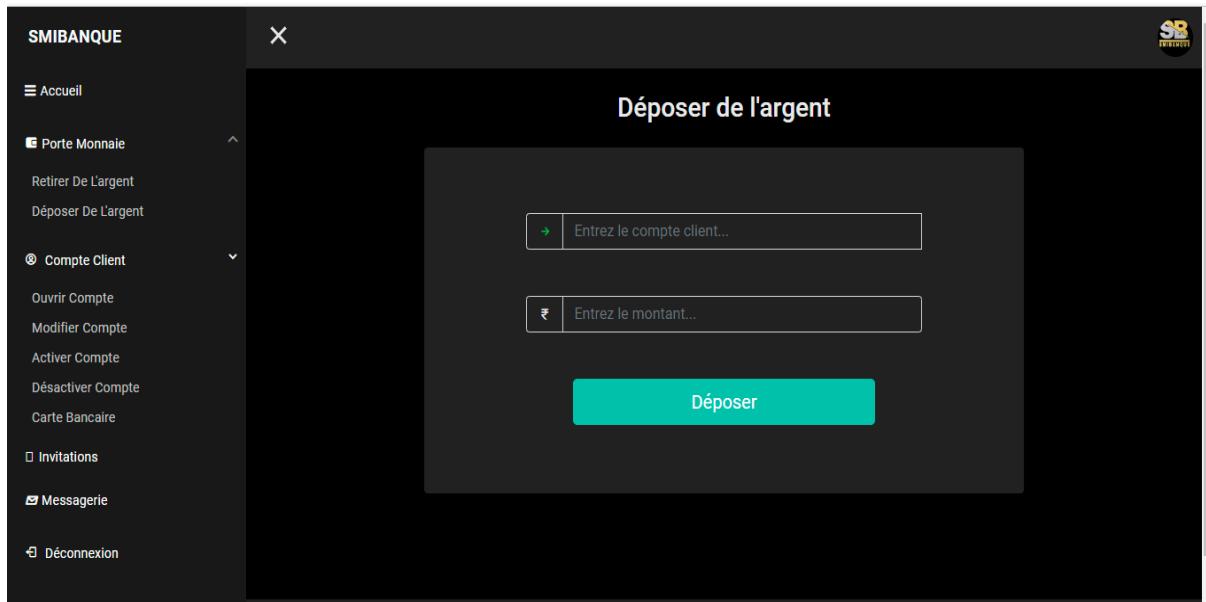


FIGURE 3.4.14 – .Interface pour déposer de l'argent

— Visualiser les messages

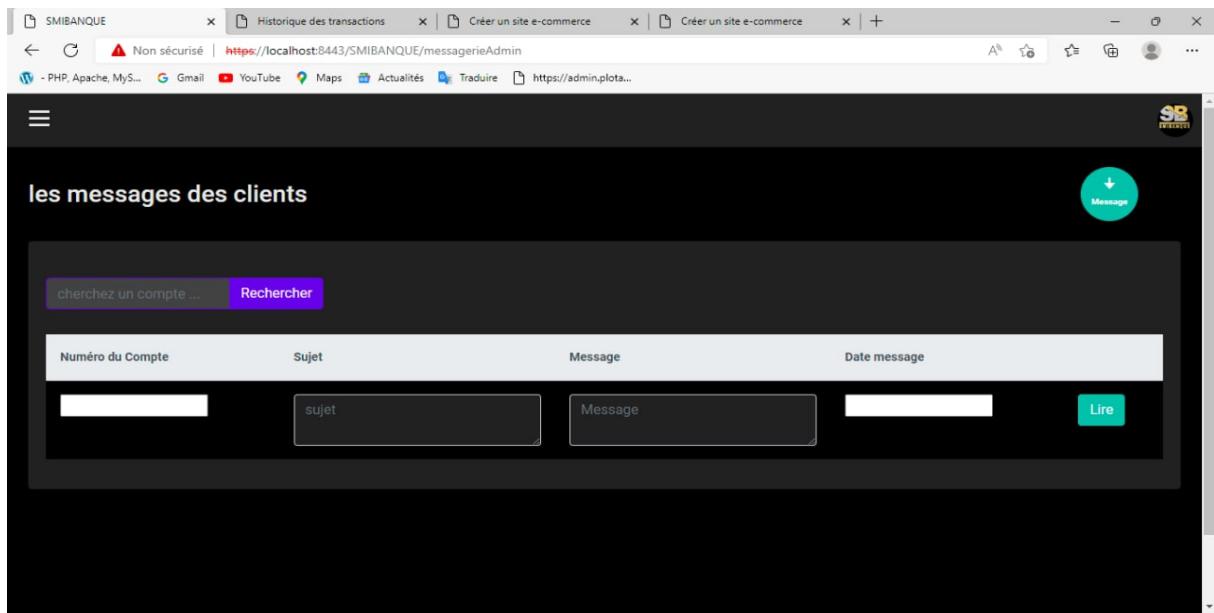


FIGURE 3.4.15 – .Interface pour afficher les messages

— Parcourir les invitations et les analyser afin d'avoir accepter ou refuser l'invitation.

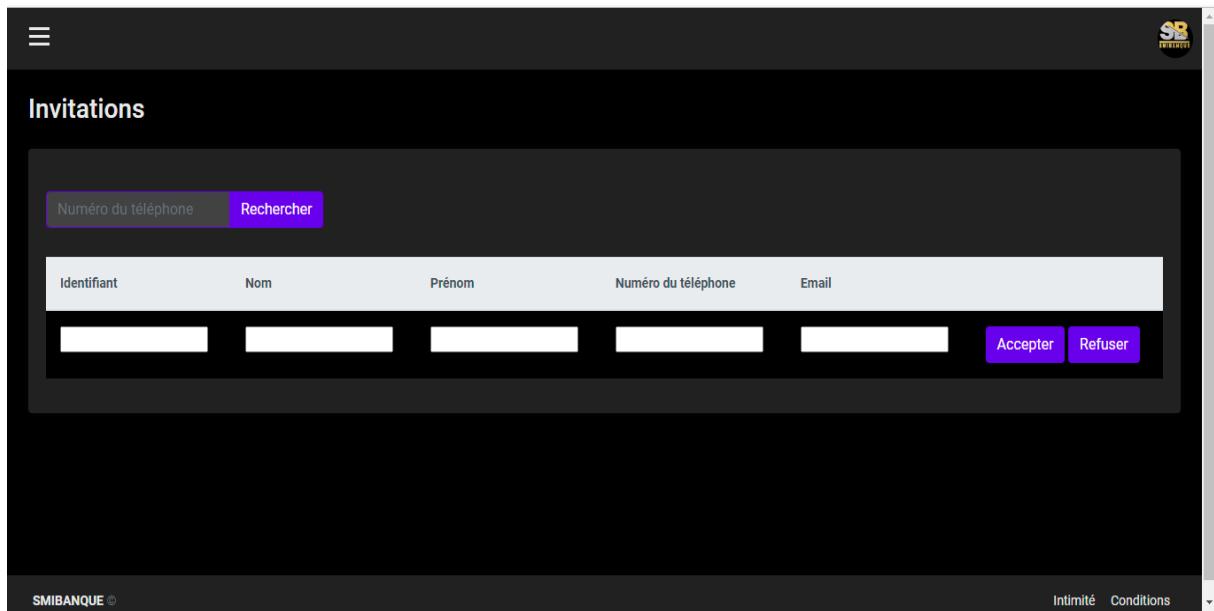


FIGURE 3.4.16 – .Interface pour afficher les invitations

— Créer une carte bancaire

FIGURE 3.4.17 – .Interface pour créer la carte bancaire

- Via cette interface l'administrateur peut ouvrir un compte en remplissant les champs avec des coordonnées correspondants au client concerné.

FIGURE 3.4.18 – .Interface pour ouvrir un compte

- Dans cette interface l'administrateur saisit le numéro de compte dans la barre de recherche pour l'activer .

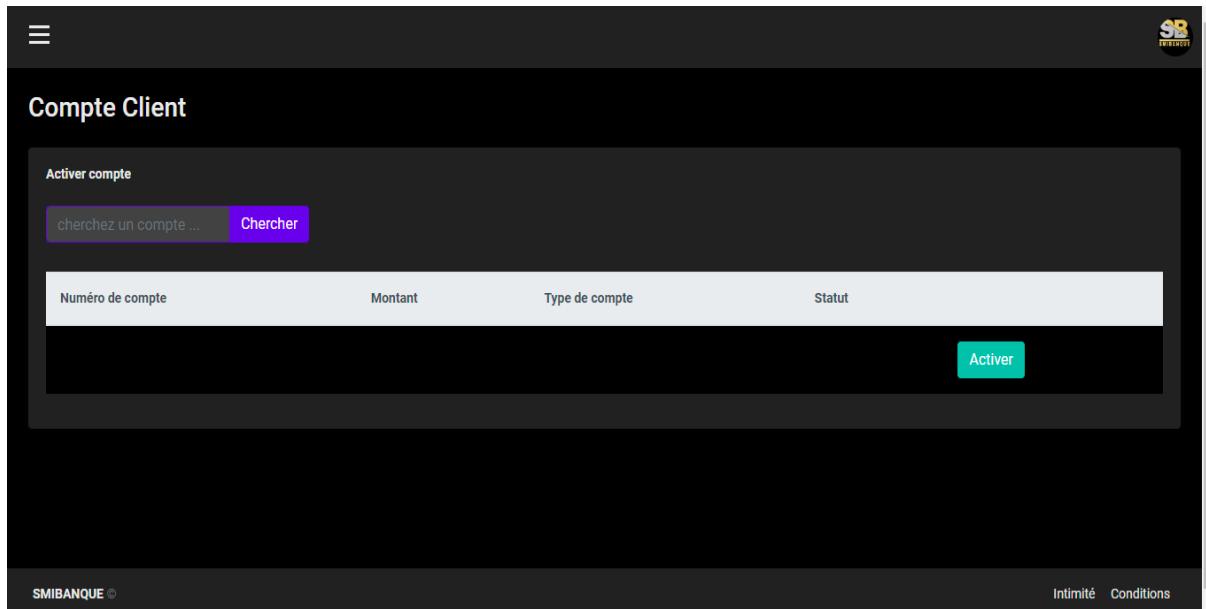


FIGURE 3.4.19 – .Interface pour activer un compte

- Dans cette interface l'administrateur saisit le numéro de compte dans la barre de recherche pour le désactiver .

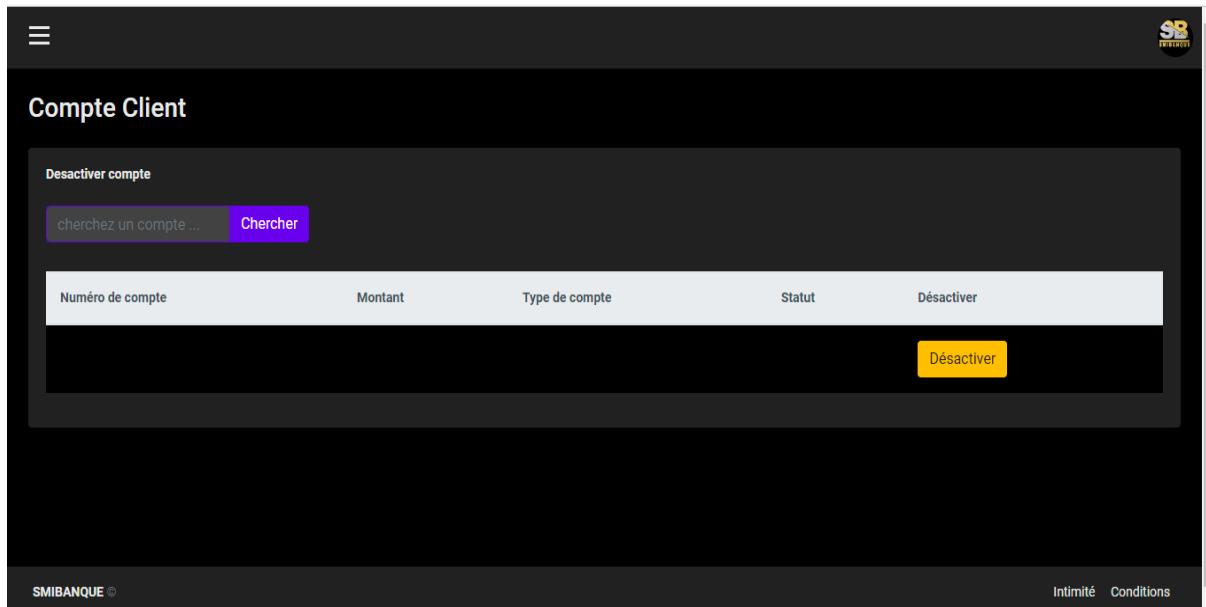


FIGURE 3.4.20 – .Interface pour désactiver un compte

### **3.4.1 Conclusion**

La dernière partie de ce projet était dédiée à la navigation dans notre application. Cette partie constitue le dernier volet de ce rapport, elle a pour objet d'exposer des captures d'écran décrivant les interfaces de l'application.

# Conclusion générale et perspective

L’élaboration de notre application nous a permis de nous confronter à une expérience professionnelle dans le domaine de développement des applications . En outre, elle nous a offert de l’opportunité de concrétiser nos connaissances théoriques à l'aide d'un cas réel. Ainsi, ce projet nous a apporté le plus concernant les langages de programmations et les plateformes que nous avons découvertes et pratiqués et qui seront certes utile dans nos futures vies professionnelles..

Sur le plan des nouvelles technologies, ce projet nous a permis de s'adapter avec l'environnement du développement informatique, de découvrir des nouvelles connaissances , d'acquérir une vision plus claire sur le développement d'applications web et d'enrichir notre formation théorique et pratique acquise tout au long de nos années de formation.

Le développement de l'application nous a permis d'approfondir nos connaissances théoriques, il nous a permis de maîtriser l'outil de UML . En plus d'avoir découvert Latex, maîtriser JAVA EE et bien comprendre la philosophie des servlets , JSP ,comment travailler avec JDBC , et aussi comment structurer notre code à l'aide de L'architecture MVC qui a pour objectif de diviser le code en 3 parties pour que ce dernier devient compréhensible , facile à le modifier...

Pendant le cycle de la réalisation de ce projet nous avons mis en pratique de nombreuses connaissances et compétences acquises durant les deux années d'études universitaires tant au niveau organisationnel technique que conceptionnel.

La réalisation de ce projet nous a permis aussi de comprendre la complexité d'un projet,comment travailler en binôme sans avoir des problèmes, comment faire des recherches cibles pour trouver les meilleures solutions aux problèmes que nous avons rencontré lors de la mise en place de ce projet.

Tout au long de ce rapport,nous avons présenté toutes les étapes nécessaires pour la conception et le développement d'une application web . Cette applications est développée spécialement pour l'utilisation de deux acteurs qui sont le client et l'admin.

- Dans un premier temps nous avons introduit le contexte général du projet à réaliser,nous avons ensuite procédé à l'analyse pour concevoir une architecture de base stable qui a été le point de départ de notre conception,et nous avons en plus défini les besoins de notre applications. Ces besoins ont été analysés dans la phase de l'étude conceptuelle à l'aide du langage de modélisation UML.
- L'étape suivante était la phase de réalisation, durant cette phase nous avons présenté l'environnement de travail et les technologies utilisées.

- Ensuite nous avons discuté les différentes technologies(MD5 , FireBase , https ) qu'on a utiliser pour réduisez les risques provenant de sources internes ou externes, et assurez la sécurité des données des clients et renforcez la confiance des utilisateurs.

La gestion de la sécurité des applications est le processus qui consiste à développer, ajouter et tester des fonctionnalités de sécurité au sein des applications, afin d'éviter les vulnérabilités face à des menaces telles que les accès et les modifications non autorisés.Pour assurer la sécurité, nous avons intégré des procédures d'authentification forte et d'autorisation ,afin de s'assurer que seuls les utilisateurs autorisés peuvent y accéder. Grâce aux procédures d'authentification, le système s'assure que l'utilisateur est bien le bon. Pour cela, nous avons demandé la saisie d'un identifiant et d'un mot de passe qui sera inséré dans la base de données d'une manière chiffré à l'aide de l'utilisation de la fonction de hachage MD5 lors de la connexion à l'application .Une fois authentifié, l'utilisateur peut être autorisé à accéder à l'application et à en tirer parti. Le système s'assure alors que l'utilisateur est capable d'accéder à l'application, en comparant son identité avec celles d'une liste d'utilisateurs autorisés qui ont déjà créé un compte bancaire.

- Et enfin nous avons présenté quelques interfaces graphiques concernant le côté client et administrateur avec leurs descriptions.

À titre général, ce projet a constitué une expérience professionnelle exceptionnelle. La vie universitaire ne permet que d'avoir un aperçu de son futur métier. On découvre rapidement que l'aspect pratique dépasse souvent les connaissances acquises et qu'il constitue une formation rapide.

Notre projet ne s'arrête pas à ce stade, en effet plusieurs fonctionnalités peuvent être ajoutées notamment un système qui permettrait de payer les factures à distance .

# Bibliographie

- [1] <https://openclassrooms.com/fr/courses/2434016-developpez-des-sites-web-avec-java-ee> **Visité le 17-06-2022**
- [2] <https://eclipse.org/downloads/> **Visité le 16-04-2022**
- [3] <https://tomcat.apache.org/download-80.cgi> **Visité le 18-04-2022**
- [4] <https://course.oc-static.com/ftp-tutos/cours/java-ee/jstl-1.2.jar> **Visité le 15-06-2022**
- [5] <https://stackoverflow.com/questions/4928271/how-to-install-jstl-the-absolute-uri-http-java-sun-com-jstl-core-cannot-be-r> **Visité le 11-05-2022**
- [6] <https://stackoverflow.com/> **Visité le 09-05-2022**
- [7] <https://www.mysql.com/fr/downloads/> **Visité le 04-05-2022**
- [8] <https://www.modelio.org/downloads/download-modelio.html> **Visité le 12-06-2022**
- [9] <https://www.01net.com/telecharger/windows/Bureautique/organiseurs/fiches/29021.html> **Visité le 20-05-2022**
- [10] <https://fr.overleaf.com> **Visité le 08-05-2022**
- [11] <https://howtodoinjava.com/java/java-security/how-to-generate-secure-password-hash-md5-sha-pbkdf2-bcrypt-examples/> **Visité le 20-05-2022**