

Département Mathématique et Informatique

# Mémoire de Projet de Fin d'Etudes

*Pour l'obtention du titre*

**D'Ingénieur d'Etat de l'ENSET de Mohammedia**

**Filière :**

**« Génie du Logiciel et des Systèmes Informatiques  
Distribués »**

**GLSID**

## Conception et mise en place des solutions E-Banking

**Lieu du stage : ADRIA BUSINESS & TECHNOLOGY**



**Soutenu le 28 / 06 / 2022**

Réalisé par :

Soufiane Boubella

Membres du jury :

M. Elhoussaine Illoussamen

M. Mohamed Youssfi

Mme. Nezha Benmoussa

Encadré par :

Encadrant(s) Entreprise :

- M. Anas BELLAGHZILIA
- Mlle. Chaima El Baz

Encadrant(s) Académique(s) :

- M. Mohamed Youssfi
- Mme. Nezha Benmoussa

**Année Universitaire : 2021-2022**

## *Dédicaces*

---

*Du profond de mon cœur, je dédie ce modeste travail à :*

### ***Mes chers parents***

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.*

*Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.*

*Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices. Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie.*

### ***Ma grande sœur***

*Que ce travail soit l'expression de mes vœux les plus ardents de bonheur et de réussite.*

### ***Mes amies***

*Pour une sincérité si merveilleuse jamais oubliable, en leurs souhaitant tous le succès et tout le bonheur.*

### ***Tous les membres de ma famille***

*Pour l'amour et le respect qu'ils m'ont toujours accordé.*

### ***Toute personne***

*Qui m'a aidé à franchir un horizon dans ma vie.*

# **Remerciement**

---

*Il m'est agréable de nous acquitter d'une dette de reconnaissance auprès de toutes les personnes, dont l'intervention au cours de ce projet, a favorisé son aboutissement.*

*Je remercie en premier lieu **M. BEKKAR Rachid** de nous avoir accordé l'occasion d'effectuer mon stage de fin d'études au sein d'Adria Business & Technology.*

*Ainsi, je tiens à remercier également tout le personnel d'Adria, pour son soutien et pour sa générosité considérable quant à l'offre de l'information.*

*Mes très chers remerciements vont à **Mlle Chaima el baz**, mon encadrant au sein d'Adria, qui m'a encadré et encouragé, et qui n'a pas manqué de me préparer les conditions favorables au bon déroulement du projet.*

*Mes remerciements les plus sincères vont aussi à **M. Mohamed Youssfi**, mon encadrant à l'ENSET MOHAMMEDIA, pour son judicieux encadrement, et pour avoir été toujours à l'écoute et pour ses conseils et coopérations, ainsi que ses conseils pertinents qui m'ont été d'un appui considérable dans ma démarche.*

*Je saisis aussi l'occasion pour remercier **Mme. Nezha Benmoussa** pour avoir été toujours à l'écoute et pour ses conseils et coopérations.*

*Mes vifs remerciements accompagnés de toute ma gratitude à l'ensemble du corps enseignant de l'ENSET MOHAMMEDIA, pour avoir porté un vif intérêt à ma formation et pour avoir accordé le plus clair de leur temps, leur attention et leur énergie et ce dans un cadre agréable de complicité et de respect.*

*Enfin, Que les membres du jury trouvent ici l'expression de ma reconnaissance pour avoir accepté d'évaluer ce travail. Que tous ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent ici l'expression de nos remerciements les plus sincères.*

## *Résumé*

---

Le présent document résume et synthétise le travail réalisé durant le stage effectué au sein de Adria Business & Technology dans le cadre du projet de fin d'études pour l'obtention du diplôme d'ingénieur d'état en Génie du Logiciel et des Systèmes Informatiques Distribués au sein de l'ENSET MOHAMMEDIA. L'objectif visé par ce projet est la conception et le développement et l'intégration d'une application web permettant l'émission des cartes pour un établissement de paiement et la création d'un wallet par reconnaissance optique de caractères.

Dans l'intention de réaliser une solution extensible, sécurisée et maintenable, une analyse des besoins est faite pour la réalisation d'un cahier de charge qui spécifie les détails du projet. En outre, la conception des fonctionnalités de la solution, l'établissement d'une étude détaillée concernant les technologies à utiliser pour la mise en œuvre de l'application, et enfin le développement des différentes fonctionnalités de l'application ont été élaborées.

La réalisation a fait appel à plusieurs technologies telles que Maven, SpringBoot, API Rest, Spring MVC, React JS, Redux, Webpack, ainsi que le langage de modélisation UML. Ce dernier est destiné à spécifier, à visualiser et à construire les diagrammes nécessaires au bon développement de notre application, mettant ainsi en pratique l'ensemble des connaissances techniques et théoriques apprises à l'Ecole Normale Supérieure de l'Enseignement Technique de Mohammedia.

**Mots clés :** SpringBoot, Rest, Maven, Spring MVC, React JS, Redux, Webpack.

# *Abstract*

---

This document summarizes and synthesizes the work done during the internship at Adria Business & Technology as part of the final year project for the diploma of state engineer in Software Engineering and Distributed Computing Systems within the ENSET of MOHAMMEDIA. The objectives of this project are the design, development, and integration of a web application allowing the issuance of cards for a payment institution and the creation of a wallet by optical character recognition.

With the intention of realizing an extensible, secure and maintainable solution, a specification and requirement analysis is made. In addition, the design of the functionalities of the solution, the establishment of a detailed study concerning the technologies used, and finally the development of the various functionalities of the application were developed.

Several technologies such as Maven, SpringBoot, API Rest, Spring MVC, React JS, Redux, Webpack were used for the implementation of the solution. The UML modeling language is intended to specify, visualize and construct the diagrams necessary for the proper development of our application, thus putting into practice all of the technical and theoretical knowledge learned at the Higher Normal School of Technical Education in Mohammedia.

Key words: SpringBoot, Rest, Maven, Spring MVC, React JS, Redux, Webpack

# *Table des matières*

---

Cadre général du projet .....	16
1. Organisme d'accueil.....	9
1.1. Présentation .....	9
1.2. Mission principale .....	10
1.3. Activités.....	10
1.4. Nos domaines d'expertise.....	10
1.5. Produits et service.....	10
1.6. Valeurs.....	11
1.7. Présence internationale .....	12
2. Présentation du projet.....	12
2.1. Contexte du projet .....	12
2.2. Etude de l'existant .....	14
2.3. Problématique.....	15
2.4. Conduite du projet.....	16
Conclusion.....	23
Etude fonctionnelle.....	24
1. Capture des besoins .....	24
1.1. Objectif du projet.....	24
1.2. Les acteurs du système .....	25
1.3. Besoins fonctionnels.....	25
2. Spécification des cas d'utilisation : .....	25
Conclusion.....	29
Etude Technique.....	30
1. Spécification des besoins techniques.....	32
2. Langages utilisés .....	33
2.1. Java.....	33
2.2. Java Script .....	33
2.3. CSS.....	33
3. Frameworks utilisés.....	34
3.1. Spring .....	34
3.2. Spring Boot.....	34

# Table des matières

---

3.3.	Spring MVC .....	34
3.4.	Maven.....	35
3.5.	Bootstrap .....	35
3.6.	React Js.....	36
3.7.	JSX .....	36
3.8.	Redux.....	37
3.9.	Web pack.....	37
4.	API web Service.....	38
4.1.	Casanet Api .....	38
4.2.	PowerCardWallet .....	38
4.3.	Magix API .....	38
5.	Outils utilisés.....	40
5.1.	IntelliJ Idea.....	40
5.2.	PostMan.....	40
5.3.	Base de données Oracle.....	41
5.4.	SQL Developer.....	41
6.	Architecture du projet.....	42
	Conclusion.....	44
	Implémentation de la solution .....	45
1.	Sprint 1 : Souscription par OCR.....	46
1.1.	Analyse .....	46
1.2.	Conception.....	48
1.3.	Réalisation .....	50
2.	Sprint 2 : Validation de la souscription par OCR.....	53
2.1.	Analyse .....	53
2.2.	Conception.....	57
2.3.	Réalisation .....	59
3.	Sprint 3 : Demande d'une carte.....	61
3.1.	Analyse .....	61
3.2.	Conception.....	63
3.3.	Réalisation .....	65
4.	Sprint 4 : Gestion des cartes .....	67
4.1.	Analyse .....	67
4.2.	Conception.....	68

## *Table des matières*

---

4.3.	Réalisation .....	70
5.	Sprint 5 : Paramétrage des cartes.....	71
5.1.	Analyse .....	71
5.2.	Conception.....	74
5.3.	Réalisation .....	77



## Liste des figures

---

Figure 1: Logo Adria B & T. ....	9
Figure 2: Les solutions Adria. ....	11
Figure 3: la présence d'ADRIA dans le monde. ....	12
Figure 4: Wallet/EDP. ....	14
Figure 5: L'enchainement des fonctionnalités du projet en général. ....	15
Figure 6: La méthodologie Agile (Scrum). ....	21
Figure 7: Diagramme de Gantt. ....	22
Figure 8: Diagramme de cas d'utilisation pour la gestion des cartes pour les agents BO. ....	26
Figure 9: Diagramme de cas d'utilisation pour l'agent (détaillant, principale). ....	27
Figure 10: Le Framework Spring. ....	34
Figure 11: architecture du Spring MVC. ....	35
Figure 12: Maven. ....	35
Figure 13: Bootstrap. ....	36
Figure 14: React Js. ....	36
Figure 15: Redux. ....	37
Figure 16: le cycle de vie des données Redux. ....	37
Figure 17: Webpack. ....	38
Figure 18: Exemple de requête pour le service Demande Service. ....	39
Figure 19: Exemple de requête pour le service CardServices. ....	39
Figure 20: Exemple de requête pour le service CardLimiteChange. ....	39
Figure 21: IntelliJ Idea. ....	40
Figure 22: PostMan. ....	40
Figure 23: Oracle SQL Developer. ....	41
Figure 24: Architecture physique du système. ....	42
Figure 25: architecture applicative du projet. ....	43
Figure 26: Diagramme de cas d'utilisation pour la souscription par OCR. ....	48
Figure 27: la classe « Demande Souscription ». ....	49
Figure 28: Diagramme de séquence de la vérification par OTP. ....	49
Figure 29: Diagramme de séquence de la création d'un compte M-Wallet. ....	50
Figure 30: la page de la souscription par OCR. ....	50
Figure 31: cas d'erreur de création du compte. ....	51
Figure 32: page de création de la demande. ....	51
Figure 33: page du récapitulatif de la souscription. ....	52
Figure 34: souscription par OCR (OTP vérification). ....	52
Figure 35: souscription par OCR cas de succès. ....	53
Figure 36: Diagramme de cas d'utilisation de la fonctionnalité Valider une souscription. ....	57
Figure 37: Diagramme de classe de validation d'une souscription. ....	57
Figure 38: Diagramme de séquence pour validation souscription. ....	58
Figure 39: cadran 1 de l'écran liste M-wallet à valider. ....	59
Figure 40: page de détails de la souscription. ....	59
Figure 41: page des détails de la demande(2). ....	60
Figure 42: Diagramme de cas d'utilisation de la fonctionnalité Demande d'une carte. ....	63
Figure 43: diagramme de classe de la demande de carte. ....	64
Figure 44: Diagramme de séquence du cas d'utilisation demande. ....	65
Figure 45: le premier cadran de la page information du compte client. ....	66
Figure 46: deuxième cadran de la page "récapitulatif des informations". ....	66
Figure 47: diagramme de cas d'utilisation de la fonctionnalité gestion des cartes. ....	68
Figure 48: Diagramme de séquence de gestion des cartes. ....	69
Figure 49: diagramme de classe de la gestion des cartes. ....	69
Figure 50: l'écran de gestion des cartes. ....	70
Figure 51: Diagramme de cas d'utilisation de la fonctionnalité Paramétrer une carte. ....	74
Figure 52: Diagramme de séquence pour le cas d'utilisation Activer service. ....	75
Figure 53: Diagramme de séquence de gestion d'état d'une carte. ....	76
Figure 54: Diagramme de séquence de recalcul du PIN. ....	77
Figure 55: cadran 1 de l'écran paramétrer la carte. ....	78
Figure 56: cadran 2 de l'écran paramétrer la carte. ....	78

## *Liste des figures*

---

Figure 57: opération d'activation du service paiement E-Com et Retrait Gabs BMCI. ....	79
Figure 58: opération d'activation du service (cas Succès).....	79
Figure 59: le cadran 3 de l'écran Paramétrer la carte. ....	79
Figure 60: le cadran 4 de l'écran Paramétrer une carte. ....	79
Figure 61: vérification par OTP pour le recalcule de PIN. ....	79
Figure 62: Recalcule de PIN (cas de succès). ....	79

## *Liste des tableaux*

---

Tableau 1: Comparaison entre les méthodologies de gestion du projet.....	17
Tableau 2: Le backlog du projet .....	28
Tableau 3: Description du cas d'utilisation de « Souscription par OCR » .....	47
Tableau 4 : Description de la fonctionnalité « Validation de la souscription par OCR » .....	53
Tableau 5: Description du cas d'utilisation « Demande d'une carte ».....	62
Tableau 6: Description de cas d'utilisation « Gestion des cartes ».....	67
Tableau 7: Description du cas d'utilisation « Paramétrer une carte ».....	73

## *Abréviations*

---

<b>Api</b>	Application Programming Interface
<b>EDP</b>	Etablissement de paiement
<b>CSS</b>	Cascading Style Sheets
<b>GAB</b>	Guichets Automatiques de Banque
<b>IDE</b>	integrated development environment
<b>JSON</b>	JavaScript Object Notation
<b>JSX</b>	Jakarta Stock Exchange
<b>JEE</b>	Java Enterprise Edition
<b>MVC</b>	Model View Controller
<b>SQL</b>	Structured Query Language
<b>SMS</b>	Short Message Service
<b>TPE</b>	Terminal de paiement électronique
<b>UML</b>	Unified Modeling Language
<b>OCR</b>	Reconnaissance optique de caractères

## *Introduction générale*

---

Ces dernières années, une foule d'intervenants sont apparus dans le paysage bancaire marocain et ont obligé les acteurs traditionnels à proposer des nouveaux services. Les dernières évolutions technologiques et les changements des modes de consommation ont bouleversé ce secteur qui doit s'adapter en permanence pour satisfaire les besoins de la clientèle.

Avec l'apparition des appareils mobiles, comme les smartphones et les tablettes qui ont connu une révolution technologique, il est devenu possible pour une banque de proposer des services sous formes des applications mobiles capables de procurer des services indispensables rapidement et à tout moment.

La solution nationale de paiement mobile en est encore au stade de démarrage. Bien que des offres aient été déjà lancées par certaines banques, il reste l'interopérabilité, l'offre dédiée aux commerçants, les applications des établissements de paiement et des autres banques...

L'établissement de paiement a fait son apparition depuis quelques années au Maroc, pour mettre fin au monopole des activités de paiement par les banques. Ce statut permet à un établissement d'offrir des services bancaires sans avoir le statut d'une banque.

Notre projet consiste à développer un module de gestion des cartes de paiement et souscription d'une demande de création d'un Wallet par reconnaissance optique de caractère pour les intégrer dans un projet de E-Banking existant, pour répondre aux besoins des établissements de paiement.

En vue de présenter notre travail effectué dans le cadre du projet de fin d'études au sein d'Adria B & T, nous vous exposons au sein de ce rapport qui est témoin de quatre mois de stage, un premier chapitre qui vous met dans le contexte du projet, en présentant en détails l'entreprise qui nous a accueillis, en discutant l'étude menée sur ce projet de fin d'étude, ainsi qu'en exposant la méthodologie utilisée et la planification du stage.

Le deuxième chapitre s'oriente vers la présentation de l'ensemble des apports du projet d'un point de vue fonctionnel des solutions E-Banking proposés.

## *Introduction générale*

---

Le troisième chapitre met le point sur les différentes technologies et outils confrontés tous au long du processus de réalisation du projet, en présentant les architectures physique et technique.

Le dernier chapitre fournit une présentation détaillée des différents sprints du processus de développement de ce projet.

Enfin, nous clôturons ce mémoire par une conclusion générale résumant les points essentiels de notre travail.

Nous espérons que vous prendrez autant de plaisir à lire ce rapport que nous en avons pris à le rédiger durant tout le déroulement de ce projet.

# *Chapitre I*

---

## **Cadre général du projet**

*Ce chapitre a pour but de présenter le cadre général dans lequel s'est déroulé le projet. Il s'agit d'introduire l'organisme d'accueil, le contexte de notre projet, la problématique qu'il traite et le planning suivant lequel a été réalisé.*

## 1. Organisme d'accueil

Adria Business & Technology est une société d'ingénierie informatique spécialisée dans l'édition et l'intégration de logiciels dans les domaines financiers que sa philosophie est celle des entreprises qui aspirent à la performance durable.



*Figure 1: Logo Adria B & T.*

### 1.1. Présentation

Le e-Banking est un terme générique qui désigne le process par lequel un client peut effectuer des transactions bancaires par voie électronique en utilisant un PC, mobile, etc... sans avoir besoin à rendre dans une institution physique.

Adria Business & Technology est une société de la nouvelle génération qui offre des services informatiques basée à Casablanca. Ces compétences s'expriment principalement dans les domaines de l'ingénierie et du conseil informatique, avec le souci permanent d'offrir à la fois les meilleures innovations technologiques et un accompagnement sur mesure. Il ambitionne de se positionner comme un partenaire de référence auprès des établissements financiers souhaitant procéder à l'externalisation complète ou partielle de leur système d'information. Son objectif est de générer de la valeur ajoutée pour ses clients en respectant leurs spécificités, contraintes, et exigences. Adria Business & Technology dispose d'une triple expertise à savoir une parfaite connaissance du secteur financier et une maîtrise technologique approfondie, notamment au niveau internet et mobile, ainsi qu'une fiabilité méthodologique intégrée dans l'ensemble des gammes des services y compris dans la gestion des projets, les normes de sécurité et les normes de développement.



## 1.2. Mission principale

Adria Business & Technology est un expert dans l'édition et l'intégration des logiciels destinés aux banques et institutions financières. Sa principale mission est d'accompagner les organisations financières dans leurs projets de Digital Banking en leur offrant des solutions innovantes, efficaces et sécurisées et des services à haute valeur ajoutée.

## 1.3. Activités

- ✓ Edition et intégration de logiciels dans le domaine de la Banque Digitale.
- ✓ Conseil et accompagnement des institutions financières dans leurs projets de Digital Banking.

## 1.4. Nos domaines d'expertise

- ✓ Expertise Métier dans le domaine de la banque ;
- ✓ Expertise Méthodologique dans la gestion des projets et services informatiques ;
- ✓ Expertise Technologique dans les nouvelles technologies Web & Mobile (JEE, Android, iOS).

## 1.5. Produits et service

Adria Business & Technology a pour principal objectif, assurer la qualité et l'aptitude des produits et services qu'elle offre à ses clients, afin de satisfaire amplement les besoins exprimés ou potentiels des consommateurs. Elle expose une variété de produits et services couvrant tous types de canaux de communication, à savoir le web et le mobile.

- ✓ **Adria banque directe cross canal** : Il s'agit d'un produit modulaire qui se positionne sur l'ensemble de la trajectoire commerciale et présente un espace en ligne intégré de la relation avec le client disposant ainsi de fonctionnalités de marketing, de vente dématérialisée et complètement en ligne, de services transactionnels, et de communication avec le client.
- ✓ **Adria mobile branch Banking** : Elle permet d'équiper les agences mobiles et les agents nomades d'une solution qui couvre toutes les fonctionnalités d'entrée en relation, de souscription de produits, de gestion de comptes et des services transactionnels...
- ✓ **Adria analytics** : Il permet de collecter, d'analyser et visualiser l'ensemble des interactions du client avec la banque quel que soit le canal utilisé.



Figure 2: Les solutions Adria.

## 1.6. Valeurs

« Au-delà des expertises, la pérennité d'une entreprise vient de ses valeurs » Adria a toujours véhiculé des valeurs simples mais fortes :

- ✓ **Satisfaction client** : La satisfaction client est le miroir de sa compétence et son engagement. La qualité du service rendu à ses clients est un préalable au développement et à la pérennité de son entreprise. Pour Adria, satisfaire le client, c'est le faire gagner au-delà de son attente et faire gagner son entreprise au-delà de ses objectifs.
- ✓ **Transparence** : Adria privilégie la transparence dans ses relations internes et externes. En véritable famille, ils considèrent que la transparence est la clé pour une communication saine et des relations sans conflits. Ils ont tous opté pour le partage d'informations claires et complètes à tous les niveaux de l'organisation, afin que tous puissent accomplir adéquatement leurs fonctions et réaliser leurs missions.
- ✓ **Esprit d'équipe** : Son principe est : « Tous pour un et un pour tous ».
- ✓ **Esprit de challenge** : L'ADN d'Adria est marqué par un grand amour pour le challenge. La digitalisation bancaire étant une véritable révolution dans la vie de la banque, la famille Adria accepte de se lancer dans ce monde nouveau pour accompagner les banques et marquer ainsi la réussite de ce grand changement dans la vie de son client.
- ✓ **Excellence opérationnelle** : La famille Adria adopte l'excellence dans tout ce qu'elle entreprend, ce qui cultive un esprit de challenge et de dépassement de soi pour viser la perfection au quotidien.

## 1.7. Présence internationale

Adria est présente par ses projets dans 16 pays :



Figure 3: la présence d'ADRIA dans le monde

## 2. Présentation du projet

### 2.1. Contexte du projet

Les clients des banques ont actuellement des exigences et attentes toutes nouvelles. C'est pour cette raison que le digital Banking devient indispensable pour les banques, afin de garder une bonne relation avec leurs clients. De plus, cette relation doit évoluer rapidement afin de satisfaire ces derniers.

De nos jours, la relation d'une banque avec ses clients est en permanence sous la pression des acteurs de l'Internet ainsi que des évolutions technologiques, modifiant ainsi les stratégies, les processus et l'organisation des banques. Tous les établissements financiers sont concernés par le digital Banking aussi bien les classiques que les néo-banques 100% en ligne.

Le digital Banking est une innovation consistant à fournir des services bancaires en ligne; ceux-ci peuvent être complémentaires aux prestations fournies en agence ou les supplanter. Les banques traditionnelles ont déjà amorcé cette transformation avec la mise en place des services numériques comme la consultation du compte, les virements, la gestion d'épargne ou la commande des chèquiers ... Cette digitalisation répondait aux besoins des consommateurs, d'où son succès. Par ailleurs, cette démarche réduit de façon appréciable les coûts de gestion au sein des banques en diminuant les charges liées aux agences physiques.

Adria est l'un des plus grands acteurs dans le domaine du digital Banking, qui a une suite de produits parmi lesquelles on trouve l'établissement de paiement (EDP).

L'établissement de paiement a fait son apparition depuis quelques années seulement dans les pays européens. C'est un opérateur qui fournit les services de paiement à l'instar des banques. Il n'a pas toutefois le statut de banque, la loi bancaire le dote d'un statut propre, mais dans l'exercice de ses activités, il est soumis au même cadre réglementaire que les banques avec quelques adaptations fixées par la banque centrale.

N'étant pas des banques, les établissements de paiement ne peuvent offrir que quelques services limitativement énumérés par la loi. Il s'agit des :

- Dépôts et retraits sur les comptes de la clientèle (comptes de paiement) ;
- Opérations de transfert de fonds ;
- Opérations de paiement par tout moyen de communication à distance ;
- Prélèvements de paiement ;
- Des opérations de paiement par carte et de l'exécution des virements.

En conséquence, les établissements de paiement ne sont pas autorisés à effectuer les opérations de paiement par chèque, par lettre de change ou par mandat postal. Ces opérations sont réservées exclusivement aux banques. Ainsi, par dérogation à l'obligation de vigilance incombant aux établissements de paiement, la réglementation, définie par Bank Al-Maghrib, prévoit une modulation des exigences en matière d'identification du titulaire d'un compte de paiement, selon 3 niveaux de solde créditeur du compte de paiement :

- **Solde créditeur ne devant pas dépasser 200 dirhams** : la présentation d'un numéro national de téléphonie mobile suffit.
- **Solde créditeur ne devant pas dépasser 5 000 dirhams** : il exige le renseignement d'une fiche d'ouverture de compte au nom du titulaire justifié par la copie d'un document d'identité officiel en cours de validité, portant la photographie du titulaire.
- **Solde créditeur ne devant pas dépasser 20 000 dirhams** : à la fiche d'ouverture de compte dont l'identité est justifiée par un document d'identité officiel, vient s'ajouter un justificatif de domicile.

Dans ce contexte Adria Business & Technology a prêté une attention particulière à ce sujet, elle s'est positionnée dans le but d'offrir des services financiers aux établissements de paiement en développant une solution, Wallet-EDP.

## 2.2. Etude de l'existant

Wallet-EDP étant un grand projet, il est réparti en plusieurs applications, une application mobile, un portail Agent et une application d'administration. Le Portail Adria Agent est une solution mise en place pour répondre aux besoins des établissements de paiement. C'est l'interface entre les agents et les clients qui permet d'effectuer un ensemble d'opérations financières tout en respectant le plafond défini par type de profil connecté, on peut distinguer entre trois profils : Agent BO, Agent principal et Agent détaillant. Les opérations offertes par le portail sont :

- Enrôlement d'un client ;
- Opérations Cash In ;
- Opérations Cash Out ;
- Gestion des clients ...



Figure 4: Wallet/EDP

Le projet Wallet-EDP se répartie en quatre applications :

- ✓ Une application mobile pour les clients
- ✓ Une application web Portail Adria, c'est l'interface entre l'agent et les clients qui permet d'effectuer toute opération financière
- ✓ Un middleware qui permet de communiquer les transactions avec les systèmes externes y compris Powercard Wallet.

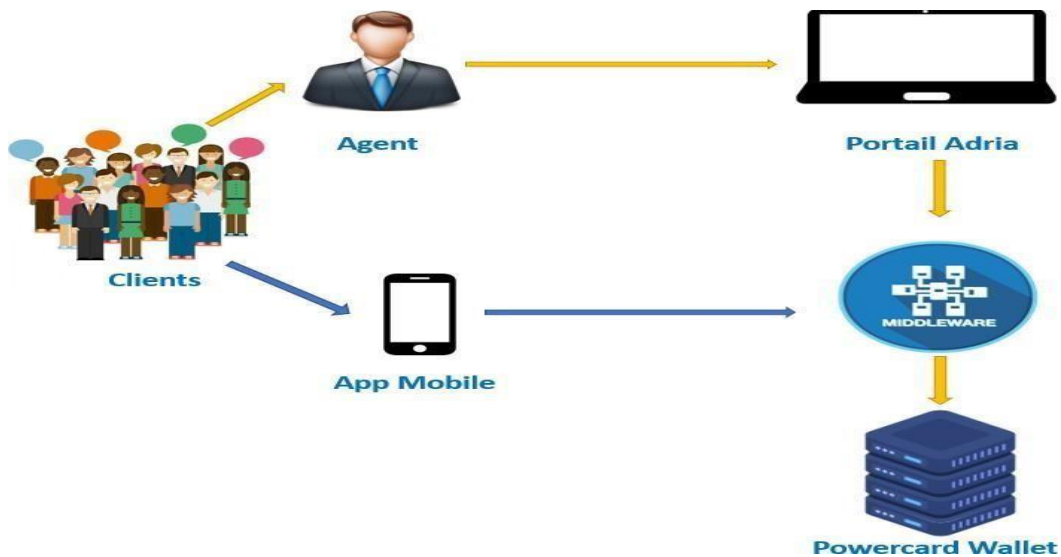


Figure 5: L'enchainement des fonctionnalités du projet en général.

### 2.3. Problématique

Dans l'ancienne loi, seules les banques avaient la possibilité d'opérer et mettre sur le marché des moyens de paiement (cartes bancaires, chèques, espèce, etc.). La nouvelle loi, en introduisant un nouveau statut accessible à tous, celui d'établissement de paiement, met fin au monopole des Banques sur les services et moyens de paiement.

La carte bancaire est un produit qu'un établissement de paiement souhaite commercialiser au travers de son réseau d'agence. Pour ce faire, l'établissement de paiement en question repose sur l'offre monétique de son partenaire qui est basée sur le socle monétique MAGIX (API).

Les cartes en question doivent être valables au Maroc et à l'international, et doivent permettre le retrait sur les GAB Nationaux, les paiements sur TPE Nationaux et le paiement e-commerce national et international.

Ainsi la souscription manuelle des demandes de création de compte était devenue fastidieux pour les agents, c'est à ce stade qu'Adria a eu l'idée d'adopter une nouvelle fonctionnalité de souscription par reconnaissance optique de caractère.

Mon projet de fin d'étude a comme objectif d'introduire un module d'émission des cartes et de souscription par reconnaissance optique de caractère dans le projet Wallet-EDP afin d'offrir aux agents des nouvelles fonctionnalités.

## **2.4. Conduite du projet**

Le bon choix du processus de développement logiciel conduit à la bonne réalisation du projet, c'est pour cette raison nous avons fait référence à une comparaison entre les principaux processus de développement, pour pouvoir choisir le meilleur processus correspondant à notre cas.

### 2.4.1. Méthodologie du projet

#### 2.4.1.1. Comparaison entre les processus de développement :

Le choix entre un modèle de processus de développement et un autre, dépend de la nature du projet et de sa taille. Le tableau suivant représente une comparaison entre les différents processus de développement.

Cycle de vie	Forces	Faiblesses	Cas d'utilisation
<b>En Cascade</b>	<ul style="list-style-type: none"> <li>• Facile à comprendre et à utiliser.</li> <li>• Adapté pour une équipe inexpérimentée</li> <li>• Les limites de chaque étape sont visibles</li> <li>• Facilite un management du Projet</li> </ul>	<ul style="list-style-type: none"> <li>• Donne une fausse impression sur l'avancement des travaux</li> <li>• Pas d'interaction entre les phases du développement</li> <li>• N'accepte pas les changements et les imprévus</li> <li>• Le client peut se retrouver non satisfait</li> </ul>	<ul style="list-style-type: none"> <li>• La phase de spécification bien faite</li> <li>• La définition du produit est stable</li> <li>• Il s'agit d'une nouvelle version d'un produit existant</li> <li>• L'implémentation d'un produit existant dans une nouvelle plate-forme.</li> </ul>
<b>En V</b>	<ul style="list-style-type: none"> <li>• Facile à utiliser</li> <li>• Les tests sont effectués en parallèle avec le développement.</li> <li>• Le contrôle se fait progressivement à chaque étape.</li> <li>• Les phases de validation sont prises en main très tôt dans le processus de développement</li> </ul>	<ul style="list-style-type: none"> <li>• Une mauvaise prise en compte des événements concurrents</li> <li>• Le processus n'est pas itératif</li> <li>• Ne prévoit pas de nouvelles décisions lors d'un cycle</li> <li>• N'est pas adaptée aux changements.</li> </ul>	<ul style="list-style-type: none"> <li>• Les spécifications de besoins doivent être bien faites</li> <li>• La solution à développer et la technologie à utiliser, doivent être parfaitement connues</li> <li>• Les changements doivent être faits avant l'analyse</li> </ul>
<b>En spirale</b>	<ul style="list-style-type: none"> <li>• Sans coût élevé, donne des indications sur les risques</li> </ul>	<ul style="list-style-type: none"> <li>• facilite la gestion des risques.</li> <li>• adaptabilité aux</li> </ul>	<ul style="list-style-type: none"> <li>• les coûts et l'évaluation des risques est important</li> </ul>



	<p>Majeurs</p> <ul style="list-style-type: none"> <li>• Les fonctions critiques à haut risque sont développées en premier lieu</li> <li>• La conception ne doit pas forcément être terminée</li> </ul>	<p>changements.</p> <ul style="list-style-type: none"> <li>• Ce modèle est complexe</li> </ul>	<ul style="list-style-type: none"> <li>• pour des projets à risque au moins moyennement élevé</li> <li>• pour des projets à long terme dont les financements peuvent varier</li> </ul>
<b>Agile</b>	<ul style="list-style-type: none"> <li>• Centrée sur le produit</li> <li>• Progrès incrémental, facilement mesurable et clairement visible pour les acteurs économiques.</li> <li>• Les développeurs fixeront le rythme, ne sont pas surchargés de travail, et bénéficient d'un rôle accru</li> <li>• La charge du travail est réglable, basé sur la capacité de l'équipe et la priorité des tâches à réaliser.</li> </ul>	<ul style="list-style-type: none"> <li>• peu ou pas de documentation écrite, ce qui pose un problème majeur lors du changement de l'équipe du développement.</li> <li>• Cette méthode demandera un client toujours disponible et prêt à s'impliquer à tout moment.</li> <li>• Si l'équipe n'est pas homogène le projet ne finira jamais.</li> </ul>	<ul style="list-style-type: none"> <li>• Pour les organisations orientées produit, ✓ pour augmenter la productivité.</li> <li>• Scrum dispose du progrès tangible réalisés par le biais fréquents, les versions de production incrémentales (Sprints).</li> </ul>

Tableau 1: Comparaison entre les méthodologies de gestion du projet.

Lorsqu'il s'agit d'un projet où les données ne sont pas réunies dès le départ, où les besoins sont incomplets, il est recommandé de s'orienter vers une méthode itérative ou orientée prototypes.

En se basant sur ce tableau comparatif, et en faisant liaison avec le projet du stage, nous avons adopté la méthode agile comme méthodologie de gestion du projet pour :

- ✓ Plus de souplesse et de réactivité.
- ✓ La grande capacité d'adaptation au changement grâce à des itérations.
- ✓ Les besoins du projet sont bien définis dès le départ.

- ✓ S'adapter à la méthodologie du travail de l'organisme d'accueil.
- ✓ La nature de cette méthode qui est bien organisée et permet de tester et valider chaque partie du projet avant de passer à une autre.

#### **2.4.1.2. La méthode SCRUM :**

Une méthode AGILE est menée dans un esprit collaboratif et s'adapte aux approches incrémentales. Elle engendre des produits de haute qualité tout en tenant compte l'évolution des besoins du client. Elle permet aussi de gérer la qualité en continu et de détecter des problèmes le plus tôt au fur et à mesure, permettant ainsi d'entreprendre des actions correctrices sans trop de pénalités dans les coûts et les délais.

Sous cette bannière se regroupent plusieurs méthodes basées sur un développement itératif et incrémental. Pour notre projet, nous sommes orientés vers la méthode Scrum, tout simplement parce que Scrum est, de très loin, la méthodologie la plus utilisée parmi les méthodes Agile existantes. Elle est donc la plus éprouvée, documentée et supportée.

Livres, blogs, formations, vidéos, associations, conférences traitant de Scrum ne manquent pas, et la majorité de ces ressources sont accessibles gratuitement. On pourrait pratiquement parler d'un standard Agile. De plus, Scrum est simple à comprendre. Or, Sa maîtrise est en revanche difficile. Les experts de Scrum, même ses fondateurs, le décrivent comme un « cadre de travail permettant de répondre à des problèmes complexes et changeants tout en livrant de manière productive et créative des produits de la plus grande valeur possible ».

##### **➤ Le principe :**

Selon Claude Aubry, auteur du plus célèbre guide sur Scrum, une pratique est « une approche concrète et éprouvée qui permet de résoudre un ou plusieurs problèmes courants ou d'améliorer la façon de travailler lors d'un développement. [...] La notion de pratique prend toute son importance avec l'utilisation des méthodes agiles. À côté des valeurs et des principes qui sont universels, les pratiques sont le reflet de la mise en œuvre sur le terrain ».

Le principe de base de la méthodologie Scrum est le suivant :

- ✓ Dégager dans un premier lieu le maximum des fonctionnalités à réaliser pour former le backlog du produit.
- ✓ En second lieu, définir les priorités des fonctionnalités et choisir lesquelles seront réalisées dans chaque itération.

- ✓ Par la suite, focaliser l'équipe, de façon itérative, sur l'ensemble de fonctionnalités à réaliser dans des itérations appelées Sprints.
- ✓ Un Sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel appelé incrément.

➤ **Les rôles dans la méthodologie Scrum :**

La méthodologie SCRUM fait intervenir trois rôles principaux qui sont :

Product Owner : Dans la majorité des projets, le responsable produit (Product Owner) est le responsable de l'équipe projet client. C'est lui qui va définir et prioriser la liste des fonctionnalités du produit et choisir la date et le contenu de chaque sprint sur la base des valeurs (charges) qui lui sont communiquées par l'équipe.

Scrum master : Véritable facilitateur sur le projet, il veille à ce que chacun puisse travailler au maximum de ses capacités en éliminant les obstacles et en protégeant l'équipe des perturbations extérieures.

Equipe : L'équipe s'organise elle-même et elle reste inchangée pendant toute la durée d'un sprint. Elle doit tout faire pour délivrer le produit.

➤ **Sprint :**

Le sprint agile représente le cœur de la méthode Scrum. Cette qualification lui correspond plutôt bien, puisque tous les développements incrémentaux menant petit à petit au produit final du projet sont réalisés au sein des sprints. Un périmètre de développement est défini au début d'un sprint et doit être entièrement réalisé lorsqu'il se termine. Chaque sprint doit apporter des fonctionnalités supplémentaires à l'application en cours de développement qui doivent être livrées lorsqu'il se termine.

Le Product Owner est le responsable de définir les sprints et d'organiser le « Product Backlog » afin de faciliter la construction du produit.

La figure suivante explique le fonctionnement de cette méthodologie :

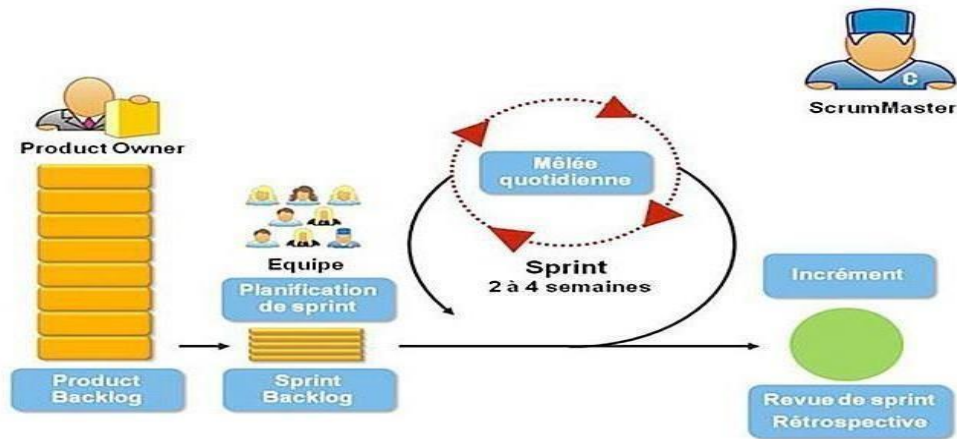


Figure 6: La méthodologie Agile (Scrum).

En effet, comme le représente ce schéma, avec SCRUM le projet est décomposé en incréments qui traitent les fonctionnalités relevées dans le Backlog une à une d'une manière itérative jusqu'à l'aboutissement du produit final. Grâce à la mêlée quotidienne nous pouvons évaluer le progrès du travail accompli et nous focaliser sur les tâches qui restent à accomplir.

Le choix de Scrum comme une méthodologie de pilotage pour notre projet s'est basé sur les atouts de ce dernier. Il se résume comme suit :

- ✓ Plus de souplesse et de réactivité.
- ✓ La grande capacité d'adaptation au changement grâce à des itérations courtes.
- ✓ La chose la plus importante, c'est que Scrum rassemble les deux cotés théorique et pratique et se rapproche beaucoup de la réalité.

#### 2.4.2. Planification du projet :

Avant de se lancer dans la réalisation du projet, il est nécessaire de prendre le temps de découper celui en sprint, afin de planifier l'exécution de ces sprints et le temps alloué pour chacun.

Le diagramme de Gantt est un outil permettant de modéliser la planification des tâches nécessaires à la réalisation de notre projet.

La figure suivante représente le planning prévisionnel de notre projet détaillé :

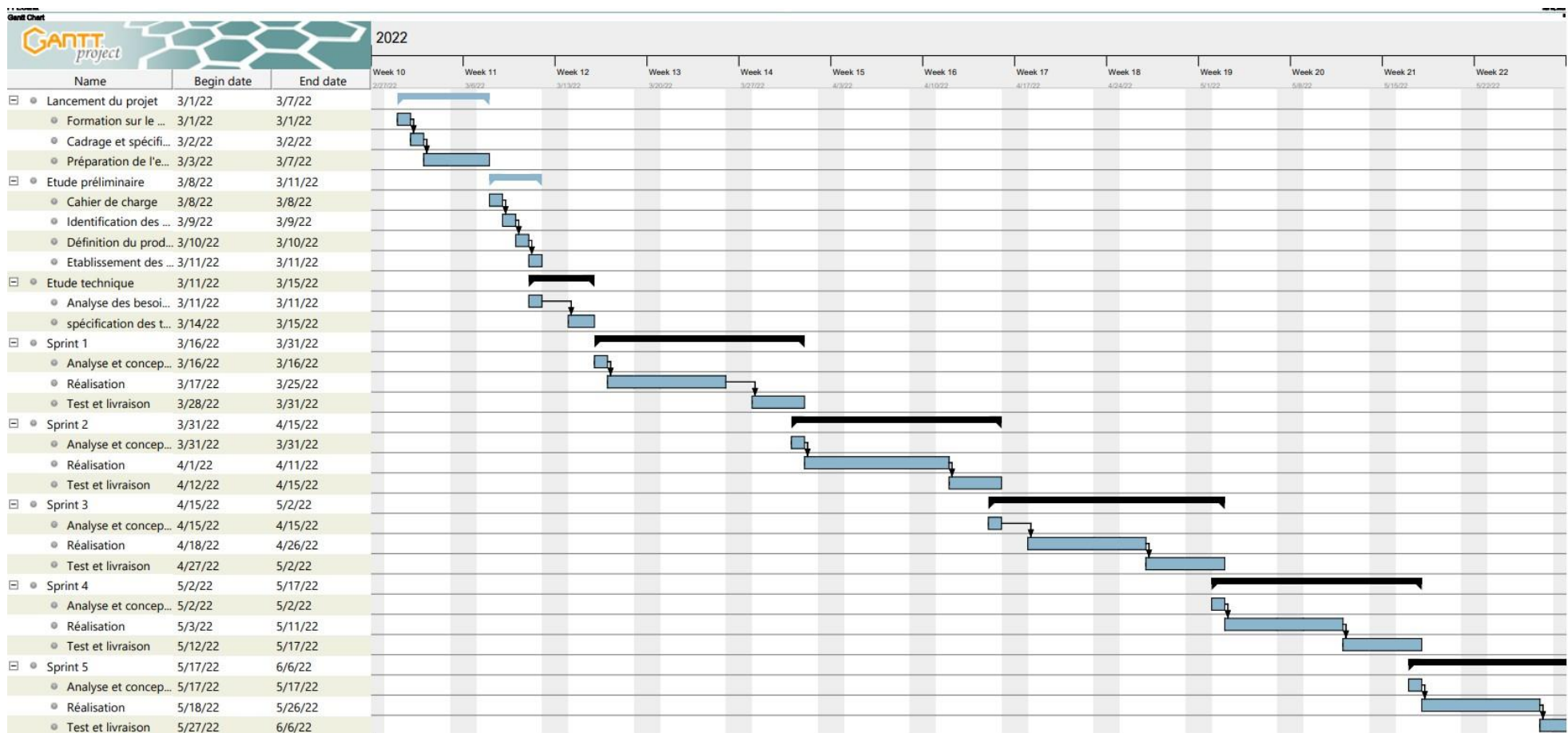


Figure 7: Diagramme de Gantt.

## **Conclusion**

Nous avons abordé au cours de ce premier chapitre le contexte général du projet, en introduisant l'organisme d'accueil, et en décrivant la problématique du projet ainsi que la méthodologie suivie pour piloter notre projet. Dans le chapitre suivant nous traiterons la partie d'analyses et spécifications du projet.

# *Chapitre II*

---

## **Etude fonctionnelle**

*Afin d'atteindre les objectifs de notre projet, l'étude fonctionnelle est une étape primordiale. Dans ce chapitre, nous nous intéresserons à l'étude et la spécification des besoins. Il a pour principal objectif de présenter un repérage initial des besoins et exigences fonctionnelles que doit satisfaire la solution.*

# 1. Capture des besoins

## 1.1. Objectif du projet

Le grand objectif du projet d'Adria Wallet-EDP, est de permettre aux clients de déposer et retirer leurs argents à l'aide d'un établissement de paiement sans avoir à passer par une banque.

L'EDP offre trois types de compte de paiement :

- Compte Niveau 1 : Plafonné à 200DHs, c'est un niveau de compte qui ne nécessite pas l'identification du client, seul un numéro de téléphone mobile est requis. Mais, c'est aussi le niveau le plus limité en termes de plafonds. Il ne peut, à aucun moment, contenir plus de 200 dirhams de fonds.
- Compte Niveau 2 : Plafonné à 5.000DHs, le niveau 2 apporte plus de flexibilité et un plafond bien plus élevé que celui du niveau 1, mais nécessite une identification « basique » du client.
- Compte Niveau 3 : Ici nous rentrons plutôt dans le domaine bancaire. Les comptes de niveaux 3 sont plafonnés à 20.000 dirhams, plafond assez élevé pour une utilisation quasi- similaire à celle d'un compte en banque classique. Les exigences d'identifications sont bien plus contraignantes que celles des autres niveaux.

Mon projet de fin d'étude porte sur le développement et l'intégration d'un module de gestion des cartes dans le projet Wallet-EDP ainsi que la création d'un Wallet par reconnaissance optique de caractère. Ces cartes seront associées aux différents types des comptes.

Les clients peuvent utiliser ses cartes pour profiter des services tel que le retrait au près des GAB ou bien des paiements TPE, lorsqu'elles sont **activées**.



Dans ce qui suit, nous détaillerons l'ensemble des fonctionnalités qui seront implémentées dans le cadre de notre projet.

## 1.2. Les acteurs du système

Les principaux acteurs de ce projet sont :

- ✓ **Les agents** : qui se chargent de répondre à toutes les demandes du client (Agent Principal, Agent Détaillant).
- ✓ **L'administrateur** (Agent BO) qui a de plus la possibilité de gestion des agents et les paramètres de l'application.
- ✓ **Le client** : utilise les fonctionnalités de la solution mise en place.

## 1.3. Besoins fonctionnels

- ✓ **Fonctionnalités d'accès** : notre solution doit comporter un menu qui facilite l'accès aux différentes fonctionnalités offertes.
- ✓ **Fonctionnalités de création d'un compte par OCR** : l'utilisateur doit avoir à sa disposition une fonctionnalité qui lui permet de créer un compte sans besoin de déplacer vers une agence précise.
- ✓ **Fonctionnalités de recherche** : l'utilisateur doit avoir à sa disposition une méthode de recherche qui lui permet de rechercher une demande de carte précise.
- ✓ **Fonctionnalités de personnalisation** : la recherche doit être personnalisée en utilisant des critères qui permettent de faciliter la recherche et d'obtenir des résultats intégrés.
- ✓ **Fonctionnalités de gestion** : les agents doivent avoir à leur disposition des moyens pour gérer les différentes demandes de cartes selon les droits dont ils possèdent, tel que la possibilité de consulter la liste des demandes, ainsi que les informations relatives à leurs propriétaires.
- ✓ **Fonctionnalité de paramétrage** : Cette fonctionnalité consiste à paramétrer les infos relatives aux cartes des clients, tel que l'activation des services, Recalcule du code PIN...

## 2. Spécification des cas d'utilisation :

Le diagramme de cas d'utilisation est un diagramme UML fournissant une vision statique et globale des comportements fonctionnels d'un système logiciel. Il est composé des acteurs externes et des cas d'utilisation.

- ✓ **Les acteurs** : toute entité externe qui interagissent avec le système.

- ✓ **Le cas d'utilisation** : représente un ensemble de séquences d'action qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Les cas d'utilisation se diffèrent selon les utilisateurs du système. Pour les agents BO, le diagramme d'utilisation sera comme suit :

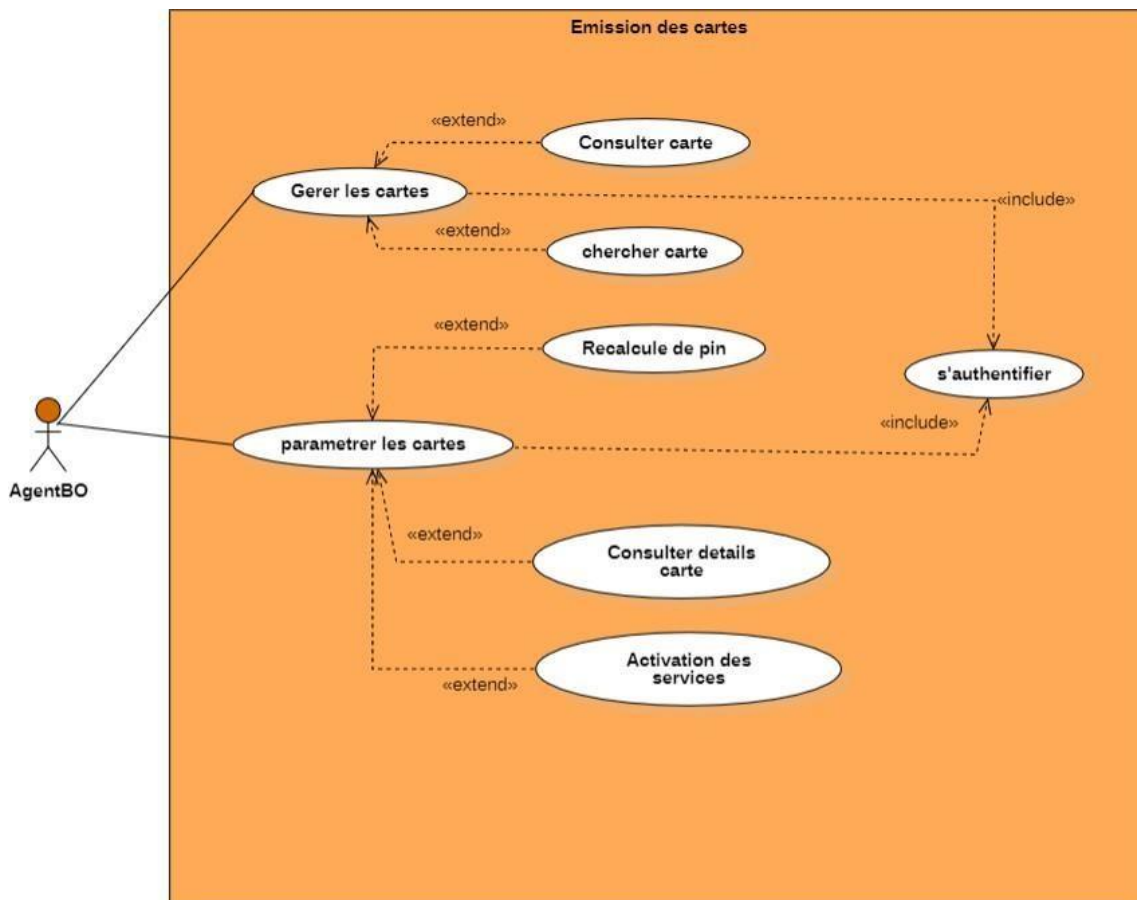


Figure 8: Diagramme de cas d'utilisation pour la gestion des cartes pour les agents BO.

Pour les agents (détaillant et principal), le diagramme de cas d'utilisation sera comme suit :

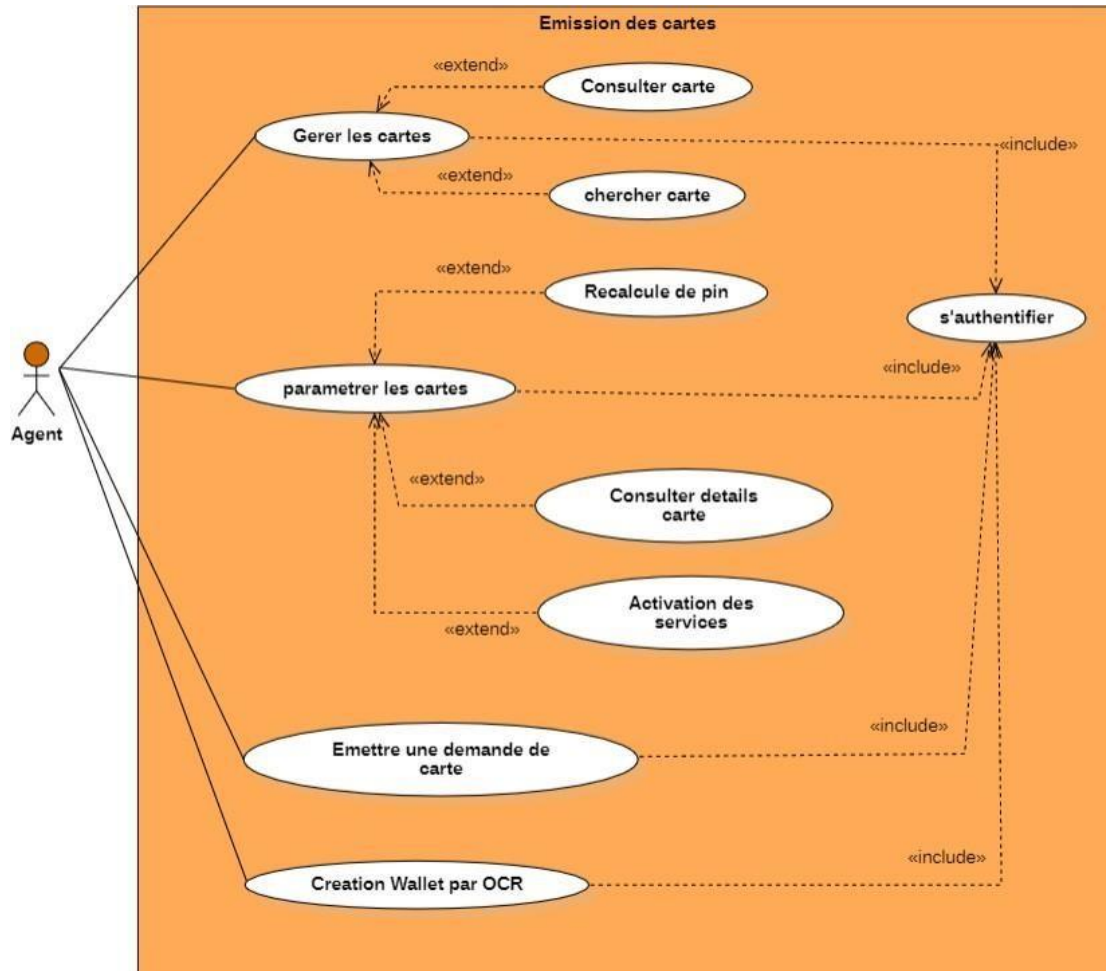


Figure 9: Diagramme de cas d'utilisation pour l'agent (détaillant, principale).

### 3. Product Backlog

L'approche Scrum propose de commencer par lister les exigences du client afin de produire le Product Backlog sous forme d'une liste d'item ou User Story, cette liste contient tout ce qui pourrait être requis dans le produit, et est l'unique source des besoins pour tous les changements à effectuer sur le produit. Le tableau suivant résume les User Stories de notre projet :

	En tant que	Epic	Je voudrais...
1	Agent	Création d'un wallet par OCR	Chercher un wallet par numéro de téléphone
2			Afficher les données récupérées.
3			Emettre la demande.
4	Agent / Agent BO	Gestion de la carte	Consulter la carte
5			Chercher la carte
6	Agent	Demande des cartes	Chercher un wallet par numéro de téléphone
7			Afficher les données récupérées.
8			Emettre la demande.
9	Agent/ Agent BO	Paramétrage des cartes	Visionner la liste des demandes.
10			Consulter les informations de la carte et du client.
11			Activer/désactiver des services.
12			Changer l'état de la carte
13			Modifier le code PIN de ma carte.

Tableau 2: Le backlog du projet.

Chaque ligne du Product backlog représente une User Story ou histoire utilisateur. Les éléments du backlog sont classés par priorité ce qui permet de définir l'ordre de réalisation. Souvent les User Stories peuvent avoir également des numéros ID uniques permettant de les identifier.

## **Conclusion**

Tout au long de ce chapitre, nous avons spécifié, tout d'abord, nos besoins afin d'avoir une vision plus claire et plus profonde sur le projet réalisé. Ensuite, nous avons présenté le backlog du projet. Dans le chapitre suivant, une étude technique sera élaborée.

# *Chapitre III*

---

## **Etude Technique**

*La capture des besoins techniques complète celle des besoins fonctionnels en spécifiant les différentes technologies et les principaux outils utilisés pour aboutir aux objectifs du projet.*

## 1. Spécification des besoins techniques

La capture des besoins techniques complète celle des besoins fonctionnels en couvrant toutes des contraintes qui ne traitent ni de la description du métier de l'utilisateur, ni de la description applicative. Cette phase nécessite un minimum de prérequis techniques.

L'application à concevoir doit être en mesure de répondre à un nombre d'exigences techniques, qu'on résume dans ce qui suit :

- **Portabilité** : L'outil doit être capable de fonctionner plus ou moins facilement dans différents environnements d'exécution. Les différences peuvent porter sur l'environnement matériel (processeur) comme sur l'environnement logiciel (système d'exploitation).
- **Ouverte à l'extensibilité** : la possibilité d'étendre simplement les fonctionnalités de l'application dans le futur sans compromettre son intégrité et sa fiabilité.
- **Disponibilité** : Etre en permanence à la disposition des utilisateurs.
- **Intégration** : Garantir la possibilité de s'interconnecter avec d'autres applications.
- **Sécurité** : Sécuriser les données confidentielles (Mots de passes, N° des cartes bancaires...);
- **Réutilisabilité** : Les composants de l'application doivent être réutilisables pour les intégrer dans différents projets ;
- **Ergonomie** : L'application doit avoir une interface conviviale, souple, ergonomique et exploitable par l'utilisateur ;
- **Temps de réponse** : L'application doit optimiser les traitements pour avoir un temps de réponse raisonnable ;
- **Maintenabilité** : Le code de l'application doit être compréhensible et lisible pour minimiser l'effort d'évolution du système et pour localiser et corriger les erreurs facilement ;
- **Tolérance aux pannes** : la partie client doit fonctionner même en cas de défaillance des microservices;

## **2. Langages utilisés**

### **2.1. Java**

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

Le langage Java reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)

### **2.2. Java Script**

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.

### **2.3. CSS**

Le CSS, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de style ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.



### 3. Frameworks utilisés

#### 3.1. Spring

Spring est un Framework de développement d'applications Java, qui apporte plusieurs composants tels que Spring Security, Spring MVC, Spring Batch, Spring Ioc, Spring Data, etc. Ces Frameworks ont pour objectif de faciliter la tâche aux développeurs. Malheureusement, leurs mises en œuvre deviennent très complexes à travers les fichiers de configuration XML qui ne cessent de grossir, et une gestion des dépendances fastidieuse.



*Figure 10: Le Framework Spring.*

#### 3.2. Spring Boot

Spring Boot est un sous projet de Spring qui facilite l'utilisation de ce dernier en éliminant plusieurs étapes de configuration. Il a la particularité d'être très léger et d'embarquer avec lui le strict minimum pour faire tourner le service, ainsi qu'il possède un serveur d'application Tomcat embarqué afin de faciliter le déploiement d'une application web et il est possible d'utiliser un serveur autre ou externe, grâce à une simple déclaration dans le fichier pom.xml.

Spring Boot permet essentiellement de réaliser des microservices. C'est une approche module particulièrement rapide et efficace pour créer une API de web services. En quelques lignes de code, avec un petit projet Maven ou Gradle, on obtient un .jar directement exécutable.

#### 3.3. Spring MVC

Spring MVC est un Framework qui permet d'implémenter des applications selon le design Pattern MVC. Donc, comme tout autre MVC Framework, Spring MVC se base sur le principe décrit par le schéma ci-dessous :

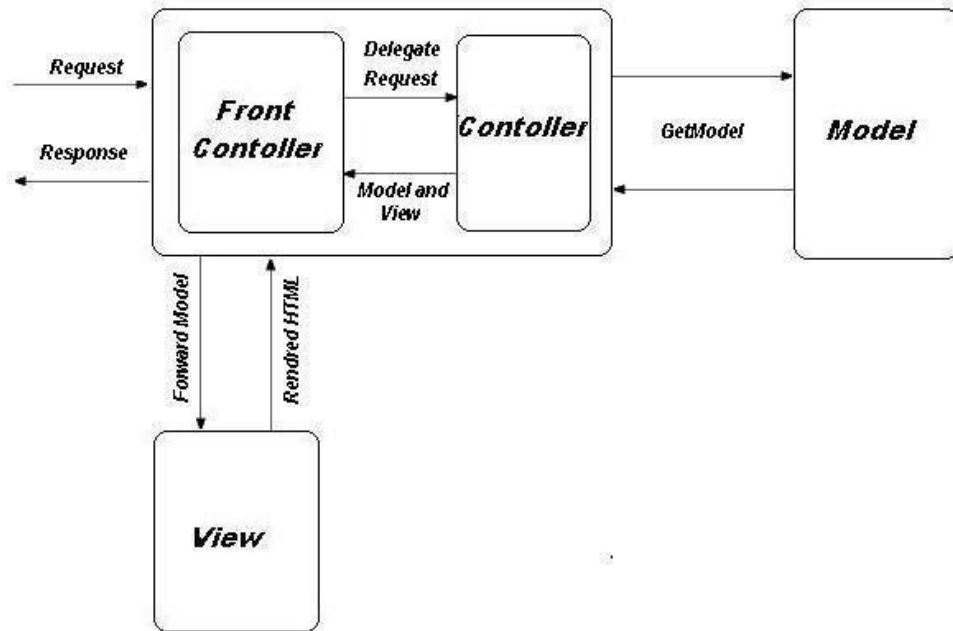


Figure 11: architecture du Spring MVC.

### 3.4. Maven

Maven est un outil de construction de projets (build) open source développée par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il permet de faciliter et d'automatiser certaines tâches de gestion d'un projet Java. Il permet notamment de :

- Automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet.
- Gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet.
- Générer des documentations concernant le projet.



Figure 12: Maven.

### 3.5. Bootstrap

C'est un Framework destiné aux applications web. Développé par Twitter et distribué sous licence Apache 2, c'est un outil à considérer lors du développement rapide d'applications web. L'utilisation combinée du HTML, du CSS, et du JavaScript propose au développeur des méthodes de développement très efficaces. Bootstrap propose :

- Une mise en page basée sur une grille de 12 colonnes ;
- Un reset CSS basé sur Normalize.css Du code fondé sur HTML 5 et CSS 3 ;
- Une bibliothèque totalement open source sous licence Apache ;
- Du code qui tient compte du format d'affichage des principaux outils de navigation (responsive design) : smartphones, tablettes...
- Des plugins JQuery de qualité ;



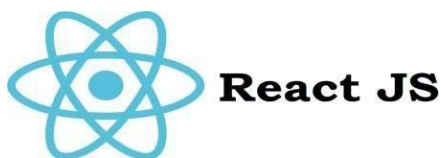
*Figure 13: Bootstrap.*

### 3.6. React Js

Une bibliothèque d'interface utilisateur développée sur Facebook pour faciliter la création des composants interactifs, statefull et réutilisables de l'interface utilisateur.

L'un de ses points de vente unique est que non seulement il fonctionne sur le côté client, mais il peut également être rendu côté serveur, et les deux peuvent travailler ensemble de manière inter-opérationnelle.

Il utilise également un concept appelé le DOM virtuel qui rend des sous-arbres de nœuds basés sur des changements d'état. Il effectue le moins de manipulation de DOM possible pour maintenir les composants à jour.



*Figure 14: React Js*

### 3.7. JSX

React peut s'écrire avec la véritable syntaxe JavaScript mais cela est facile à écrire et encore moins à lire. JSX apporte une véritable souplesse dans l'écriture, dans la lecture et la compréhension. Babel propose un preset pour JSX ce qui permet de transpiler le JSX en code JavaScript classique et exécutable par le navigateur.

### 3.8. Redux

Redux se présente comme étant un système de centralisation des données et des actions. React est particulièrement convaincant lorsqu'on commence mais on se retrouve rapidement confronter à certains problèmes de conceptions lorsque nos applications deviennent un peu plus élaborées qu'une simple liste.



Figure 15: Redux.

L'architecture Redux s'articule autour d'un flux de données unidirectionnel strict. Cela signifie que toutes les données d'une application suivent le même modèle du cycle de vie, ce qui rend la logique de votre application plus prévisible et plus facile à comprendre.

Le cycle de vie des données dans n'importe quelle application Redux suit ces 4 étapes :

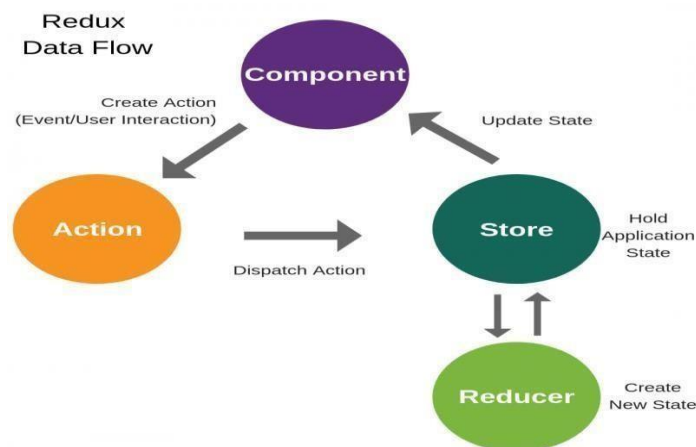


Figure 16: le cycle de vie des données Redux

### 3.9. Web pack

C'est un outil qui est devenu aujourd'hui incontournable pour les projets JavaScript complexes. Il permet de morceler le code sous forme de modules qui seront ensuite fusionnés en un seul fichier par Webpack. Il dispose, en plus, d'un système de "loaders" qui vont permettre d'inclure de nouveaux types de fichiers ou d'appliquer des transformations spécifiques.



Figure 17: Webpack.

## 4. API web Service

« API » et « Service web » désignent tous deux des moyens de communication. La seule différence entre ces deux termes est le fait qu'un service web facilite les interactions entre deux machines hétérogènes situées sur un réseau. Une API, elle, sert d'interface entre deux applications différentes, afin qu'elles puissent communiquer ensemble.

En général, pour communiquer avec un web service, on va utiliser une API (Application Programming Interface). Ce sont des interfaces permettant d'utiliser un outil sans connaître son fonctionnement.

### 4.1. Casanet Api

Elle fournit des services web permettant d'envoyer des SMS aux clients.

### 4.2. PowerCardWallet

PowerCARD-Wallet est une solution de paiement mobile, avec des capacités de paiement instantané et de notification instantanée, fournissant son propre compte prépayé ou se connectant à tout moyen de paiement externe.

PowerCARD-Wallet gère l'ensemble de l'écosystème, y compris les clients, les commerçants, les agents, les distributeurs, ... Une fois inscrits à la plateforme, les clients peuvent utiliser leur compte prépayé intégré ou l'une de leurs cartes bancaires et comptes bancaires existants comme source de fonds pour transférer de l'argent ou payer des marchandises.

### 4.3. Magix API

Cette API nous offre les services web en REST qu'on doit consommer pour de la réalisation du projet. Elle permet de communiquer avec le serveur tiers qui représente le système monétique.

Les web service utilisés dans ce projet sont :

- /DemandeService : Permet de faire une demande pour créer une carte :

```
{
  "crt_securtoken_24_e1": "7IVxybs7Rnv1JGvb",
  "crt_code_product": "P001",
  "crt_command_id": "020115000003",
  "crt_action": "1",
  "crt_command_nbr": "003",
  "mac_value": ""
}
```

Figure 18: Exemple de requête pour le service Demande Service.

- /CardServices : Permet de faire des actions sur la carte :  
Activation/Résiliation/Recalcul de PIN ;

```
{
  "crt_securtoken_24_e1": "666Yh8AQPEE53WIj",
  "crt_card_token": "02P0010169000151",
  "rib_client": "002258015588666666110033",
  "consumer_id": "00000000000000000054",
  "crt_action": "L",
  "cpt_reason": "04",
  "mac_value": ""
}
```

Figure 19: Exemple de requête pour le service CardServices

- /CardLimiteChange : Permet d'activer / désactiver des services de la carte.

```
{
  "crt_securtoken_24_e1": "PFS9iRdVgq5CwCsv",
  "crt_card_token": "P001020048000034",
  "rib_client": "002258014088550000000033",
  "consumer_id": "00000100000000000099",
  "limite_cartes_code": "DC-6K/W-60/M-3",
  "date_application": "20072020",
  "mac_value": "321654"
}
```

Figure 20: Exemple de requête pour le service CardLimiteChange.

## 5. Outils utilisés

### 5.1. IntelliJ Idea

IntelliJ IDEA est un IDE intégré de technologie Java destiné au développement de logiciels informatiques. Une étude de différents IDE (Integrated Development Environment) a été menée pour choisir le bon environnement. Les deux principales alternatives étaient Eclipse IDE et IntelliJ IDEA, mais après avoir évalué IntelliJ le choix nous a semblé évident. En considérant qu'on peut apprécier les qualités d'IntelliJ IDEA en termes de stabilité et de périmètre fonctionnel.



*Figure 21: IntelliJ Idea.*

### 5.2. PostMan

PostMan est une extension Google Chrome permettant d'interagir avec les API HTTP. Il présente une interface graphique conviviale pour la création de requêtes et la lecture des réponses.



*Figure 22: PostMan*

### 5.3. Base de données Oracle

Oracle est un SGBD (système de gestion de bases de données) édité par la société du même nom (Oracle Corporation - <http://www.oracle.com>), leader mondial des bases de données, créée en 1977. Il permet d'assurer :

- La définition et la manipulation des données ;
- La cohérence des données ;
- La confidentialité des données ;
- L'intégrité des données ;
- La sauvegarde et la restauration des données ;
- La gestion des accès concurrents ;

### 5.4. SQL Developer

Oracle SQL Developer est un environnement de développement intégré multiplateformes, fourni gratuitement par Oracle Corporation et utilisant la technologie Java. C'est un outil graphique permettant d'interroger des bases de données Oracle à l'aide du langage SQL.



*Figure 23: Oracle SQL Developer.*



## 6. Architecture du projet

Adria offre ses fonctionnalités en se basant sur un standard composé d'un serveur Backend (Noyau) qui traite des requêtes envoyées par le FrontEnd, une base de données Adria pour l'enregistrement des données et un middleware. Ce dernier est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques. Dans notre cas, les composants logiciels du middleware assurent la communication entre le BackEnd et les APIs consommées de l'extérieur tel que Magix...

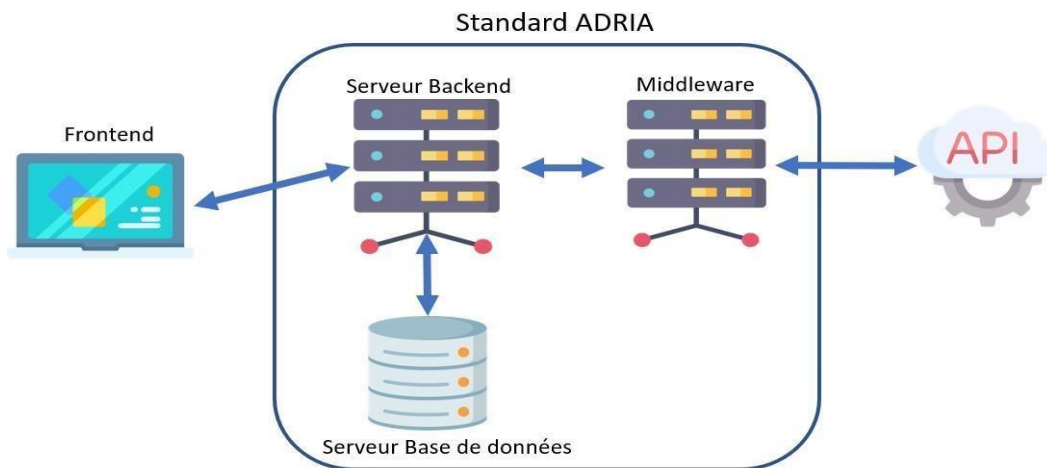


Figure 24: Architecture physique du système.

Notre application se compose de trois couches :

❖ **Partie FrontEnd** : composée de deux couches :

- Couche Vue : il s'agit des écrans et interfaces avec lesquels on peut interagir. |  
Couche Reducer : Le rôle des réducteurs (Reducer) est de gérer l'état global de notre système. C'est une fonction qui prend deux arguments (l'état actuel et une action) et renvoie un nouvel état en fonction des deux arguments. Ces réducteurs jouent le rôle des contrôleurs qui se chargent de consommer les services web du BackEnd.

❖ **Partie Backend** : composée de trois couches :

- Couche Contrôleur : comme son nom l'indique, a pour rôle essentiel de coordonner les séquences d'actions/réactions d'une application web. Il reçoit des requêtes, déclenche une logique à base d'objets métiers, et choisissent la vue qui constituent la réponse à la requête. Cette couche offre une suite de services web qui seront consommés par la partie FrontEnd et plus précisément par les réducteurs.
- Couche Service : Elle regroupe tous les services, ou réalisation de cas d'utilisation. C'est la couche dans laquelle se trouvent les fonctions permettant la communication avec la couche repository et qui se charge de consommer les services web exposés par le middleware.
- Couche Repository : c'est la couche dans laquelle se trouvent les fonctions permettant de persister dans la base de données.

❖ **Partie Middleware** : composée de deux couches :

- Couche Contrôleur : Son rôle est d'offrir une suite de services web qui seront consommés par la partie Backend (par les contrôleurs).
- Couche Service : Cette couche contient les fonctions permettant de consommer les services web exposés par les API (Magix...).

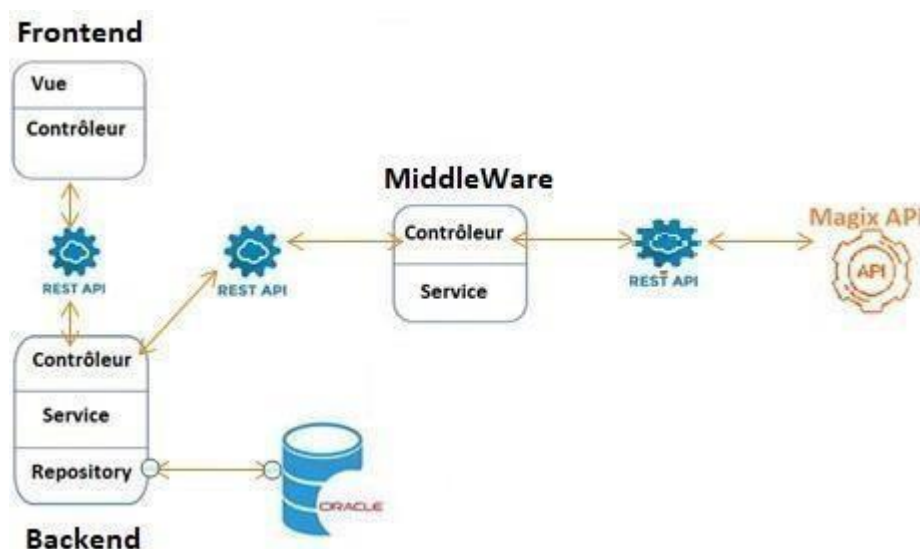


Figure 25: architecture applicative du projet.

## **Conclusion**

Dans cette section, nous nous sommes intéressés à introduire l'ensemble des outils et technologies utilisés pour réaliser ce travail, ainsi que l'architecture physique et technique du projet. Dans la partie suivante, nous allons présenter les différents sprints du projet en détaillant, pour chacun, les différentes étapes analyse, conception et réalisation.

# *Chapitre IV*

---

## **Implémentation de la solution**

*Dans ce chapitre nous allons détailler les différentes fonctionnalités fournies par notre système pour répondre au cahier des charges. Le processus de réalisation de notre application est composé de cinq Sprints.*

# 1. Sprint 1 : Souscription par OCR

## 1.1. Analyse

La création d'un M-Wallet par OCR est une fonctionnalité très importante dans les établissements de paiement. C'est elle qui permet au client de créer un compte facilement et sans avoir l'obligation de se déplacer vers une agence. Dans cette section nous allons analyser ce cas d'utilisation de notre solution en utilisant le tableau descriptif suivant.

<b>Acteur</b>	Agent détaillant, Agent principal
<b>Objectif</b>	L'agent peut créer un compte M-Wallet en profitant de cette fonctionnalité.
<b>Description</b>	<p>Cette fonctionnalité comporte deux étapes, la première est une page de saisie, pour la création du compte M-Wallet, constituée de trois champs :</p> <ul style="list-style-type: none"> <li>• Liste déroulante : Type de compte ;</li> <li>• Liste déroulante : Type de la carte d'identité ;</li> <li>• Champ : Numéro de téléphone.</li> <li>• Champ : Email (facultatif).</li> </ul> <p>Et d'autres champs qui sont récupérés automatiquement à partir de la carte d'identité grâce à l'algorithme de traitement d'image SDK api :</p> <ul style="list-style-type: none"> <li>• Nom</li> <li>• Prénom</li> <li>• CIN</li> <li>• Date de validité CIN</li> <li>• Date de naissance</li> <li>• Lieu de naissance :</li> <li>• Adresse</li> <li>• Sexe</li> </ul>

	<p>Elle contient aussi un bouton « Enregistrer » qui permet de créer la demande.</p> <p>La deuxième est un récapitulatif des informations de la demande :</p> <ul style="list-style-type: none"> <li>• Nom et Prénom ;</li> <li>• Téléphone ;</li> <li>• Email ;</li> <li>• Pièce d'identité et CIN ;</li> <li>• Date de validité CIN ;</li> <li>• Adresse et Lieu de naissance ;</li> <li>• Statut de la demande ;</li> </ul> <p>Elle contient de plus un bouton « Créer » pour créer le compte M-Wallet, et un autre « Annuler » pour l'annuler, et le bouton « Modifier ».</p>
<b>Scénario</b>	<ul style="list-style-type: none"> <li>• L'agent accède au menu et appuie sur « Souscription par OCR » ;</li> <li>• Le système affiche l'écran ;</li> <li>• Il remplit le formulaire, puis il appuie sur « Enregistrer » ;</li> <li>• Le récapitulatif s'affiche ;</li> <li>• Il sélectionne créer pour créer le compte ou bien annuler pour l'annuler et aussi il a la possibilité de modifier les informations saisies.</li> </ul>

Tableau 3: Description de cas d'utilisation « Souscription par OCR ».

## ❖ Diagramme de cas d'utilisation

La figure suivante illustre le diagramme des cas d'utilisation pour la souscription par OCR.

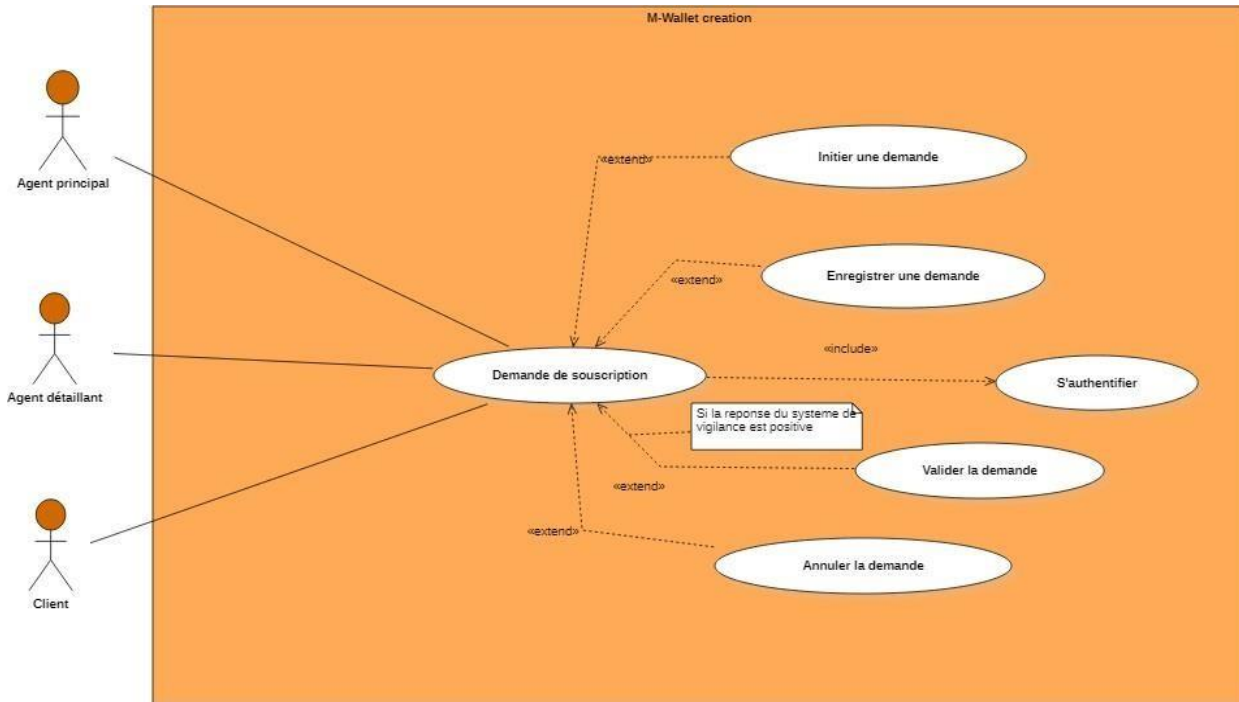


Figure 26: Diagramme de cas d'utilisation pour la souscription par OCR.

## 1.2. Conception

### 1.2.1. Diagramme de classes

Dans notre système la souscription par OCR est représentée par la classe «DemandeSouscription », composée des champs suivants :

- Cle\_rib\_compte: RIB du compte ;
- Numero\_compte: Numéro du compte;
- Statut\_vigilance : le statut du test de vigilance ;
- Date\_validation : date de validation de la demande ;
- Type\_demande : le type du compte
- Statut : le statut de la demande (enregistrée, signée, rejeté) ;

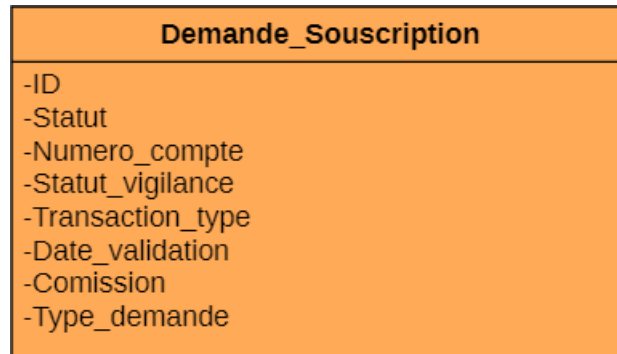


Figure 27: la classe « Demande Souscription ».

### 1.2.2 Diagramme de séquence

Après avoir remplis le formulaire de la page « Souscription par OCR », l'agent valide la demande et elle sera ensuite persistée dans la base de données avec le statut «enregistrée », et un récapitulatif des informations de la demande créée sera affiché, et par la suite l'agent doit créer le compte en cliquant sur le bouton «Créer », et après l'envoi de la demande de création du compte au système monétique, l'agent doit poursuivre un processus de vérification par OTP, et un champ de saisie d'OTP s'affichera. Le client renseigne le code qu'il a reçu à l'agent pour qu'il puisse finaliser l'opération. Dans le cas d'un OTP valide, une demande sera envoyée au système monétique PowerCardWallet via le middle. La figure qui suit illustre le diagramme de séquence de la vérification par OTP :

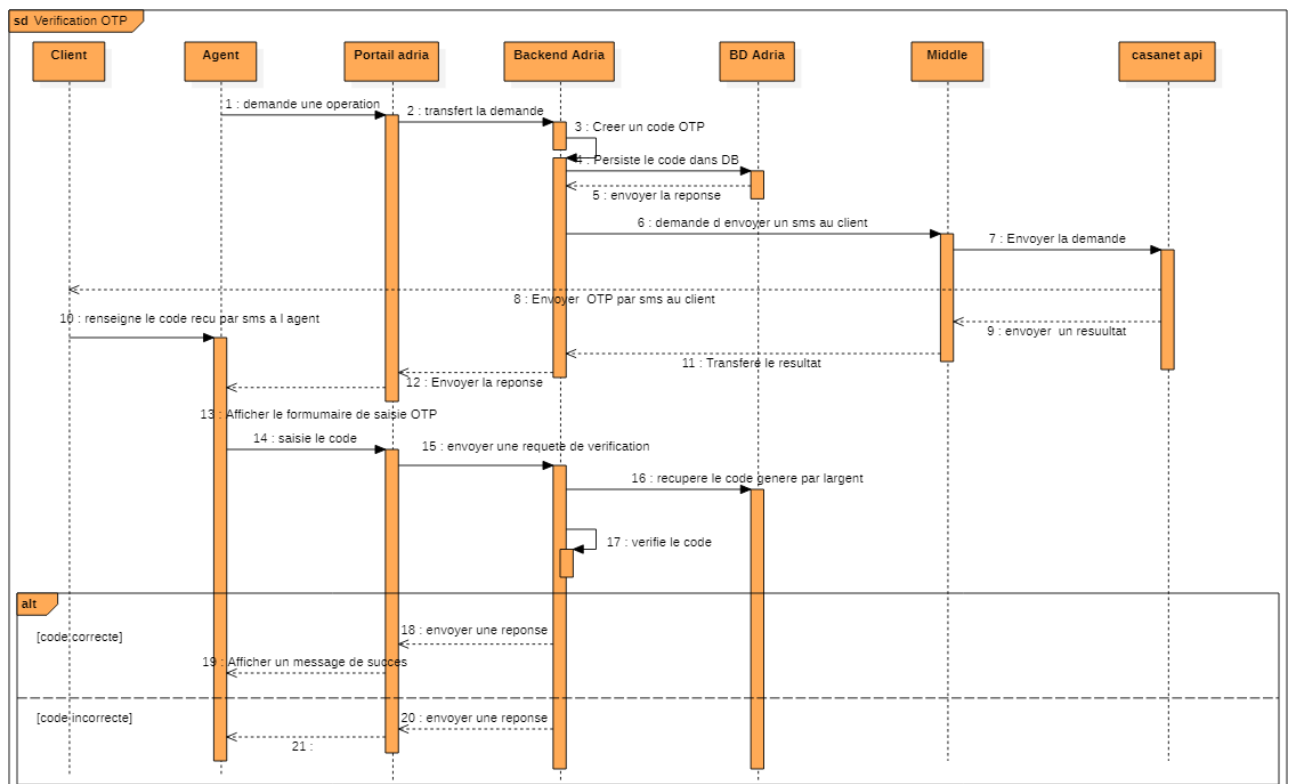


Figure 28:Diagramme de séquence de la verification par OTP



Pour la validation, la demande doit passer un test du système de vigilance. Dans le cas de succès, le système va transférer la demande au système PowerCardWallet, si la demande est valide le système change la valeur de statut de la demande pour qu'il soit « signée », si le teste de vigilance est négative, la demande doit être validé par l'agent BO.

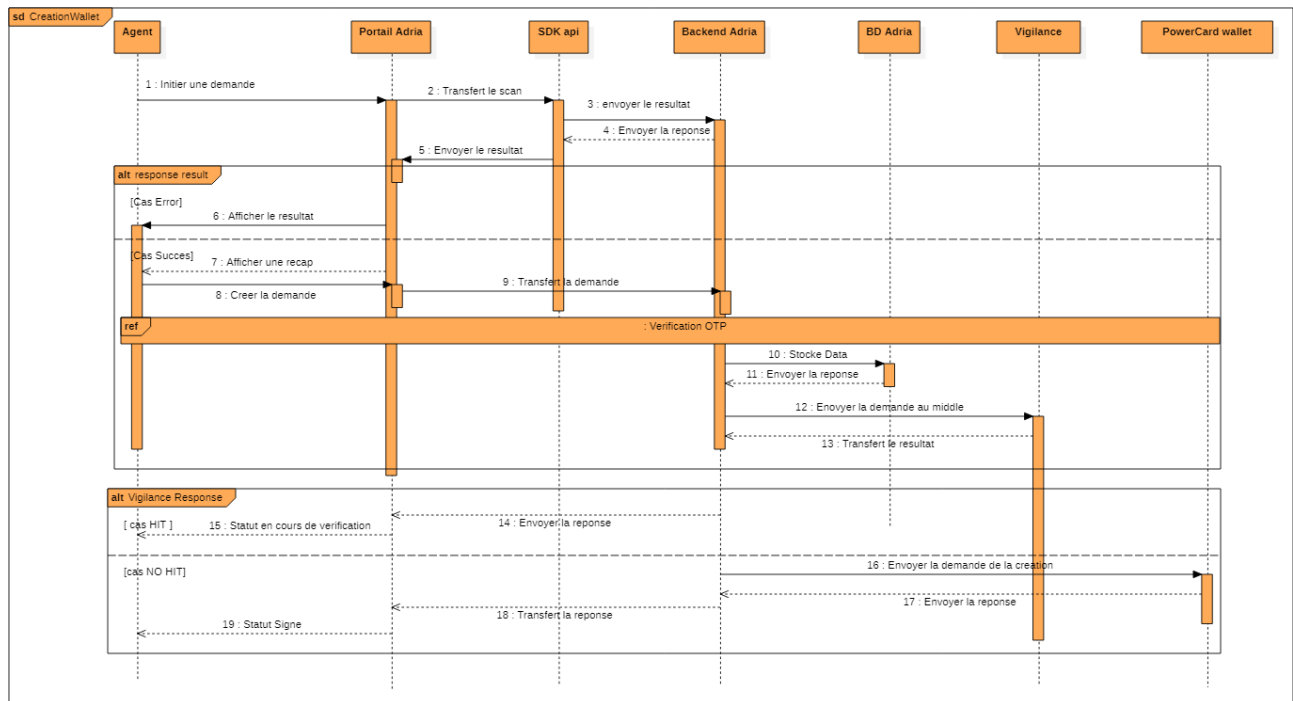


Figure 29: Diagramme de séquence de la création d'un compte M-Wallet.

### 1.3. Réalisation

La figure suivante représente le formulaire de la souscription par OCR.

Figure 30: la page de la souscription par OCR.

L'utilisateur va choisir le type du compte, et il va aussi importer la carte d'identité, Dans le cas d'échec, le système affiche un message d'erreur :

The screenshot shows the 'Informations clients' section of a web application. At the top, there is a red error message: 'Un problème technique est survenu ! Merci de réessayer.' Below this, there is a link 'Scanner la carte d'identité'. The main section is titled 'Nouvelle carte' and contains two steps: '1 Face avant' and '2 Face arrière'. Each step has a corresponding icon of a document. In the center, there is a large box with a cloud icon and the text 'Choisissez une image'. To the right, there are two preview images of a Moroccan National Identity Card (Carte Nationale d'Identité) with the text 'ROYAUME DU MAROC' and 'CARTE NATIONALE D'IDENTITE'. At the bottom right, there is a green 'Valider' button.

Figure 31: cas d'erreur de création du compte.

Dans le cas de succès, la page suivante s'affiche et par la suite l'agent va remplir le champ du numéro de téléphone et email :

The screenshot shows the 'Documents justificatifs' section of a web application. It contains a form with the following fields: 'Nom' (MEJJATI), 'Prénom' (ADIL), 'Téléphone' (+212679219164), 'Email (facultatif)' (boubellasoufiane@gmail.com), 'Pièce d'identité' (C.I.N.), 'CIN' (B3336145), 'Date de validité CIN' (02/03/2031), 'Date de naissance' (30/06/1985), 'Lieu de naissance' (HAY MOHAMMADI HAY MOHAMMADI AIN SEBAA), 'Adresse' (LOT JARDINS DE LA COLLINE VILLA 29 COMMUNE SIDI HAJJAJ TIT MELLIL CASA), and 'Sexe' (Masculin). Below the form, there is a section for 'Documents justificatifs' with two preview images of a Moroccan National Identity Card and a large box with a cloud icon and the text 'Choisissez une image'. At the bottom right, there are two buttons: 'Réinitialiser' and 'Enregistrer'.

Figure 32: page de création de la demande.

Une page récapitulative des informations s'affiche :

Figure 33: page du récapitulatif de la souscription.

Figure 34: souscription par OCR(OTP vérification).

Après la validation une vérification par OTP nécessaire pour compléter l'opération :  
Après la vérification par OTP un message de succès sera affiché :

Super ! votre demande d'ouverture de compte est en cours de validation

Type de compte  
Smart 5000

**Informations clients**

Sexe <b>Masculin</b>	Prénom <b>ADIL</b>
Nom <b>MEJJATI</b>	Email <b>boubellasoufiane@gmail.com</b>
Téléphone <b>+212679219164</b>	CIN <b>B3336145</b>
Pièce d'identité <b>C.I.N</b>	Date de naissance <b>30-06-1985</b>
Date de validité CIN <b>02-03-2031</b>	Adresse <b>LOT JARDINS DE LA COLLINE VILLA 29 COMMUNE SIDI HAJJAJ TIT MELLIL CASA</b>
Lieu de naissance <b>HAY MOHAMMADI HAY MOHAMMADI AIN SEBAA</b>	
Statut : <b>En cours de validation</b>	

Figure 35: souscription par OCR cas de succès.

## 2. Sprint 2 : Validation de la souscription par OCR

### 2.1. Analyse

Le tableau suivant représente une analyse détaillée du cas d'utilisation « Validation de la souscription par OCR » :

<b>Acteur</b>	Agent BO
<b>Objectif</b>	Cette fonctionnalité permet à l'agent back-office de gérer la validation d'une demande en cas de nécessité.

<b>Description</b>	<p>Cette fonction est une page d’affichage se portant sur deux cadrans.</p> <p>❖ <b><i>Cadran 1 : Titre : Liste des ouvertures des M-Wallet à valider</i></b></p> <p>Les informations à afficher dans ce cadran sont :</p> <ul style="list-style-type: none"><li>• Date ;</li><li>• Type compte ;</li><li>• Nom ;</li><li>• Prénom ;</li><li>• Numéro de téléphone ;</li><li>• Email ;</li><li>• Statut ;</li><li>• Action (une colonne pour voir en détails la demande à valider)</li></ul>
--------------------	--

### ❖ *Cadran 2 : Titre : Détails de la demande*

Le cadran est constitué de 4 parties :

#### ➤ Partie 1 : information du client

Cette partie contient une récapitulatifs des informations de la demande à valider :

- Nom
- Prénom
- Type de compte
- CIN
- Tribunal code
- Source de revenu ....

#### ➤ Partie 2 : Documents justificatifs

C'est une partie qui affiche l'ensemble des documents justificatifs lié à la demande de souscription. L'agent a la possibilité de les consulter.

#### ➤ Partie 3 : Scan rapport vigilance

Cette section est réservée en cas d'échec du scan de vigilance automatique. L'agent aura la possibilité d'ajouter une description et voir des images pour valider le scan du rapport de vigilance.

#### ➤ Partie 4 : Motif (en cas de rejet de la demande)

Si l'agent n'approuve pas la demande, il doit spécifier un motif de rejet. Ça peut être en cas de :

- CIN expiré
- Contrat non signé
- Document scanné illisible
- Document non valide
- Adresse CIN saisie différent du justificatif

<p><b>Scénario</b></p>	<ul style="list-style-type: none"> <li>• L'agent clique sur le bouton « Liste M-Wallet » se trouvant sur le menu ;</li> <li>• L'écran « Liste ouverture M-Wallet à valider » se charge, et le premier cadran sera affiché ;</li> <li>• Pour valider une demande l'agent doit cliquer sur le bouton « Visionner » dans la colonne action du tableau.</li> <li>• L'écran du deuxième cadran va s'afficher, celui du « détails de la demande ».</li> <li>• Ce dernier est composé de 4 sections.</li> <li>• L'agent peut donc consulter les informations récapitulatives de la demande dans la première section de la page.</li> <li>• Dans la deuxième section l'agent peut aussi consulter les documents justificatifs reliés à la demande.</li> <li>• Dans la troisième, il peut ajouter manuellement une description et un justificatif lié au Scan de rapport de vigilance.</li> <li>• Dans la dernière section l'agent peut spécifier un motif de rejet en cas de rejet de la demande.</li> <li>• Si l'agent n'approuve pas la validation de la création du compte, il va cliquer sur le bouton « rejet la demande »</li> <li>• Dans le cas contraire l'agent doit cliquer sur le bouton « valider la demande »</li> </ul>
------------------------	---

**Tableau 4 : Description de la fonctionnalité « Validation de la souscription par OCR ».**

### ❖ Diagramme de cas d'utilisation

Le diagramme suivant regroupe l'ensemble d'opération qu'un agent peut effectuer sur la validation de souscription d'un Wallet.

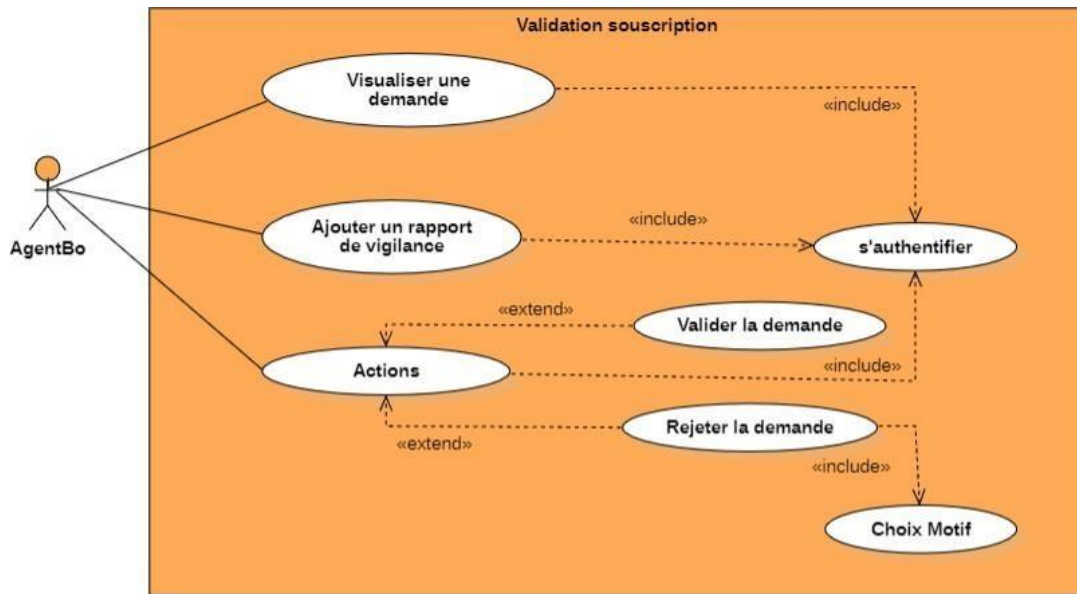


Figure 36: Diagramme de cas d'utilisation de la fonctionnalité Valider une souscription.

## 2.2. Conception

### 2.2.1. Diagramme de classe :

La figure suivante représente le diagramme de classe correspondant à la validation d'une demande de souscription par reconnaissance optique de caractère.

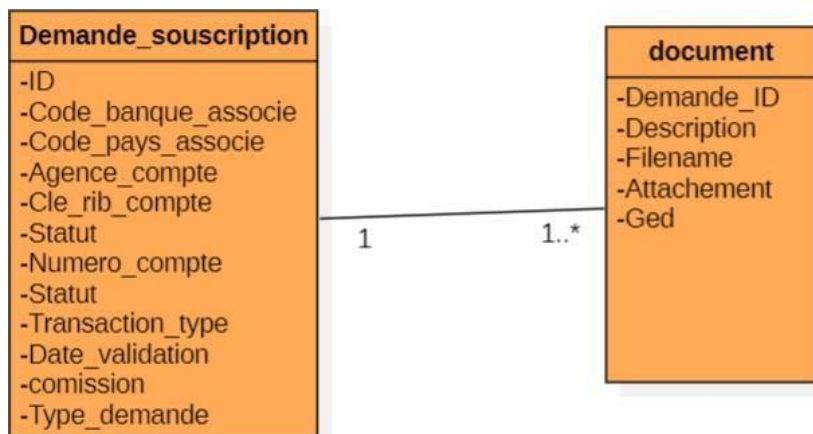


Figure 37: Diagramme de classe de validation d'une souscription



### 2.2.2. Diagramme de séquence

La validation d'une demande de souscription est l'une des fonctionnalités les plus importantes dédiées aux agents back-office. Il s'agit de valider les demandes après les avoir traités.

Une validation chez l'agent back-office est nécessaire selon 2 cas :

- Si le compte est de type niveau 3 (plafond 20 000 dirhams) : Dans ce cas la demande de souscription a besoin d'une validation par l'agent back-office
- Si le test de vigilance n'a pas abouti pour le compte de types 1 et 2.

La figure suivante représente le diagramme de séquence correspondant à la validation d'une demande de souscription.

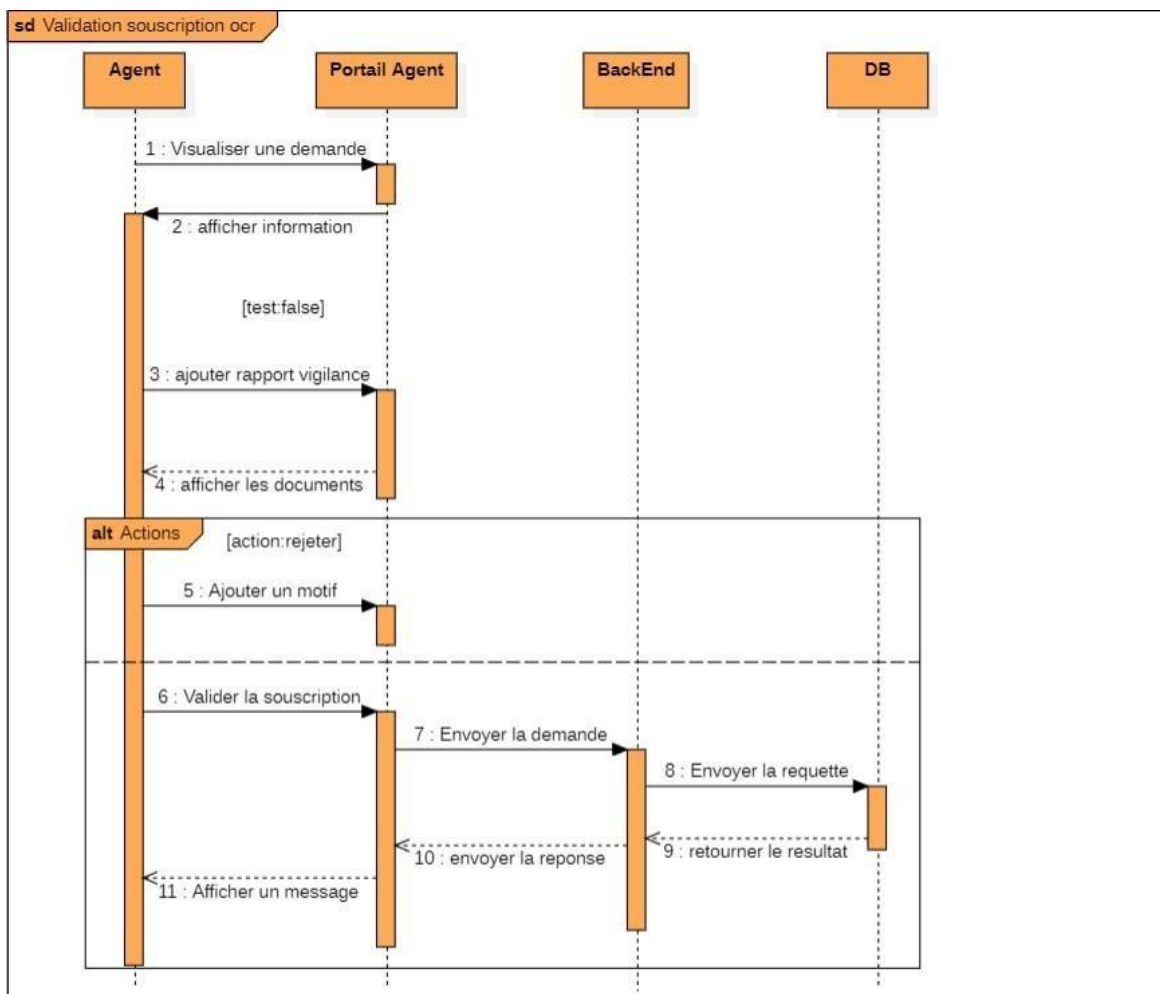


Figure 38: Diagramme de séquence pour validation souscription

## 2.3. Réalisation

La figure suivante représente le premier cadran de l'écran liste des ouvertures des M-wallet à valider.

Liste des ouvertures du M-Wallet à valider

DATE	TYPE DE COMPTE	NOM	PRENOM	N° TEL	EMAIL	STATUT	ACTIONS
01-06-2022 12:11:24	Compte commerçant - Plafond : 20 000 DH	test	test	+21265638666	test@gmail.com	En cours de validation	
01-06-2022 11:59:48	Compte commerçant - Plafond : 20 000 DH	test	test	+21265658566	test@gmail.com	En cours de validation	
31-05-2022 19:17:24	Compte commerçant - Plafond : 20 000 DH	achoukh	sara	+212627743777	sara.achoukh@gmail.com	En cours de validation	
31-05-2022 18:51:26	Compte commerçant - Plafond : 20 000 DH	achoukh	sara	+212627743499	sara.achoukh@gmail.com	En cours de validation	
31-05-2022 18:37:37	Compte commerçant - Plafond : 20 000 DH	achoukh	sara	+212627743444	sara.achoukh@gmail.com	En cours de validation	
27-05-2021 11:49:30	Smart 200	BENABDES	SOUHAÏL	+21266356997		En cours de validation	
27-05-2021 11:49:30	Smart 200	test	test	+212679239209		En cours de validation	
27-05-2021 11:49:30	Smart 20000	TLEMCAÏ	MERYEM	+212662167770		En cours de validation	
27-05-2021 11:49:30	Smart 20000	pedrosi	pedrosi	+212618532231		En cours de validation	
27-05-2021 11:49:30	Smart 200	LYOUZBACH	IBRAHIM	+212671585678		En cours de validation	

1 2 3 4 5 6 7 8 9 10

Figure 39: cadran 1 de l'écran liste M-wallet à valider.

Une fois que l'agent Back-office clique sur le bouton « visualiser », la page des détails de la demande s'affiche.

Type de compte  
Compte commerçant - Plafond : 20 000 DH

Informations clients

Date de création: 31-05-2022 18:51:26	Statut: En cours de validation
Nom: achoukh	Prénom: sara
Téléphone: +212627743499	Email: sara.achoukh@gmail.com
CIN: 3526899	Date de validité CIN: 05-06-2022
Date de naissance: 19-05-2006	
Adresse: Bloc 10 numéro 04 Bensergao, Agadir	Activité: Source de revenu: FO2
Type de justificatif de résidence: CDR	Code de la raison sociale: mcc: 5697
Nom de la raison sociale:	
Numéro patente:	
Tribunal code: 123456	

Figure 40: page de détails de la souscription

The screenshot shows a web application interface for 'Scan Rapport Vigilance'. On the left is a dark blue sidebar with icons for home, search, user profile, and document management. The main content area has a header section with a file upload bar showing 'FicheProfil\_achouikh\_sara.xls | Fiche Profil' and icons for user and delete. Below this is the 'Scan Rapport Vigilance' section. It contains a 'Description' field, a 'Fichier' section with a 'Choisir un fichier' button, a text input 'Aucun ...r choisi', and a '+' button. The 'Motif' section has a label 'Entrer le motif de rejet :' and a dropdown menu 'Choisir Motif :'. At the bottom right, there are two buttons: '✓ Valider la demande' and '✓ Rejeter la demande'.

Figure 41: page des détails de la demande (2).

### 3. Sprint 3 : Demande d'une carte

#### 3.1. Analyse

Dans cette section on va traiter le cas d'utilisation « Demande d'une carte » :

<b>Acteur</b>	Agent (détaillant et principale)
<b>Objectif</b>	Cette fonctionnalité permet d'émettre une demande de carte d'un wallet existant.
<b>Description</b>	<p>Cette fonction est une page d'affichage à deux cadrans :</p> <p>❖ <i>Cadran 1 - Titre : informations du compte client</i></p> <p>Constitué des Champs :</p> <ul style="list-style-type: none"> <li>• Nom client ;</li> <li>• Prénom client ;</li> <li>• Numéro de compte ;</li> <li>• Numéro de téléphone ;</li> <li>• Adresse ;</li> <li>• Ville de naissance</li> <li>• Pays de naissance ;</li> <li>• Date de naissance ;</li> <li>• Pièce d'identité ;</li> </ul> <p>❖ <i>Cadran 2 - Titre : Récapitulatif des informations</i></p> <p>Ce cadran est une page constituée d'un récapitulatif des informations du client choisi :</p> <ul style="list-style-type: none"> <li>• Nom ;</li> <li>• Prénom ;</li> <li>• Téléphone :</li> <li>• Pièce d'identité</li> <li>• RIB</li> </ul>

	<ul style="list-style-type: none"> <li>• Adresse</li> <li>• Date de naissance</li> <li>• Ville de naissance</li> <li>• Pays de naissance</li> <li>• Frais de création</li> <li>• Statut</li> </ul>
<b>Scénario</b>	<ul style="list-style-type: none"> <li>• L'agent accède à l'écran de demande des cartes en appuyant sur le bouton « Nouvelle Carte » dans la partie d'émission des cartes ;</li> <li>• Après avoir saisi le numéro de téléphone et cliquer sur le bouton de recherche, le système charge les informations liées au wallet du client ;</li> <li>• L'agent clique sur le bouton « enregistré » correspondant à la carte en question ;</li> <li>• Il accède au cadran « récapitulatif des informations »</li> <li>• Il appuie sur le bouton « Valider », il sera amené à saisir le code de validation reçu par le client via Sms.</li> <li>• Il clique en fin sur le bouton « Valider la commande » pour finaliser la demande ;</li> </ul>

Tableau 5: Description du cas d'utilisation « Demande d'une carte ».

### ❖ Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation suivant illustre les différentes opérations que les agents peuvent faire en utilisant cette fonctionnalité.

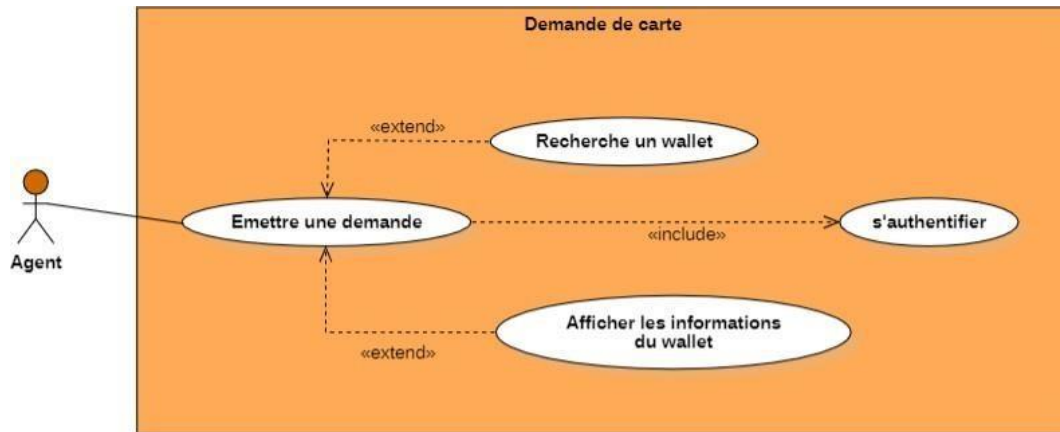


Figure 42: Diagramme de cas d'utilisation de la fonctionnalité Demande d'une carte.

## 3.2. Conception

### 3.2.1. Diagramme de classes

La demande des cartes est représentée dans notre système par la classe « DemandeCarte », composée des attributs suivants :

- Token : unique est associé à toutes les cartes depuis la création jusqu'à la fin de sa vie fournie par le système monétique ;
- Nom : le nom du client
- Prénom : le prénom du client ;
- Date de naissance : la date de naissance du client ;
- Numéro de téléphone : le numéro de téléphone du client ;
- Adresse : l'adresse du client ;
- Ville de naissance : la ville de naissance du client ;
- Pays de naissance : le pays de naissance du client ;
- CIN : le code identité nationale du client ;
- RIB : le numéro de compte du client ;
- Statut : le statut de la demande ;
- Frais création : les frais de création de la carte ;
- Service : les services que possède la carte ;
- Tentative signature : le nombre de tentative de saisi de sms effectué au moment de la demande ;

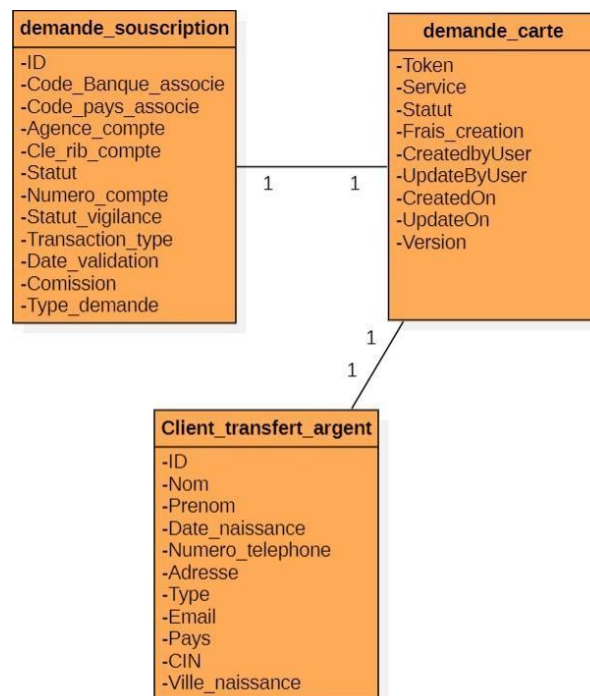


Figure 43: diagramme de la classe de la demande de carte

### 3.2.2. Diagramme de séquence

La figure suivante représente le diagramme de séquence correspondant à ce cas d'utilisation :

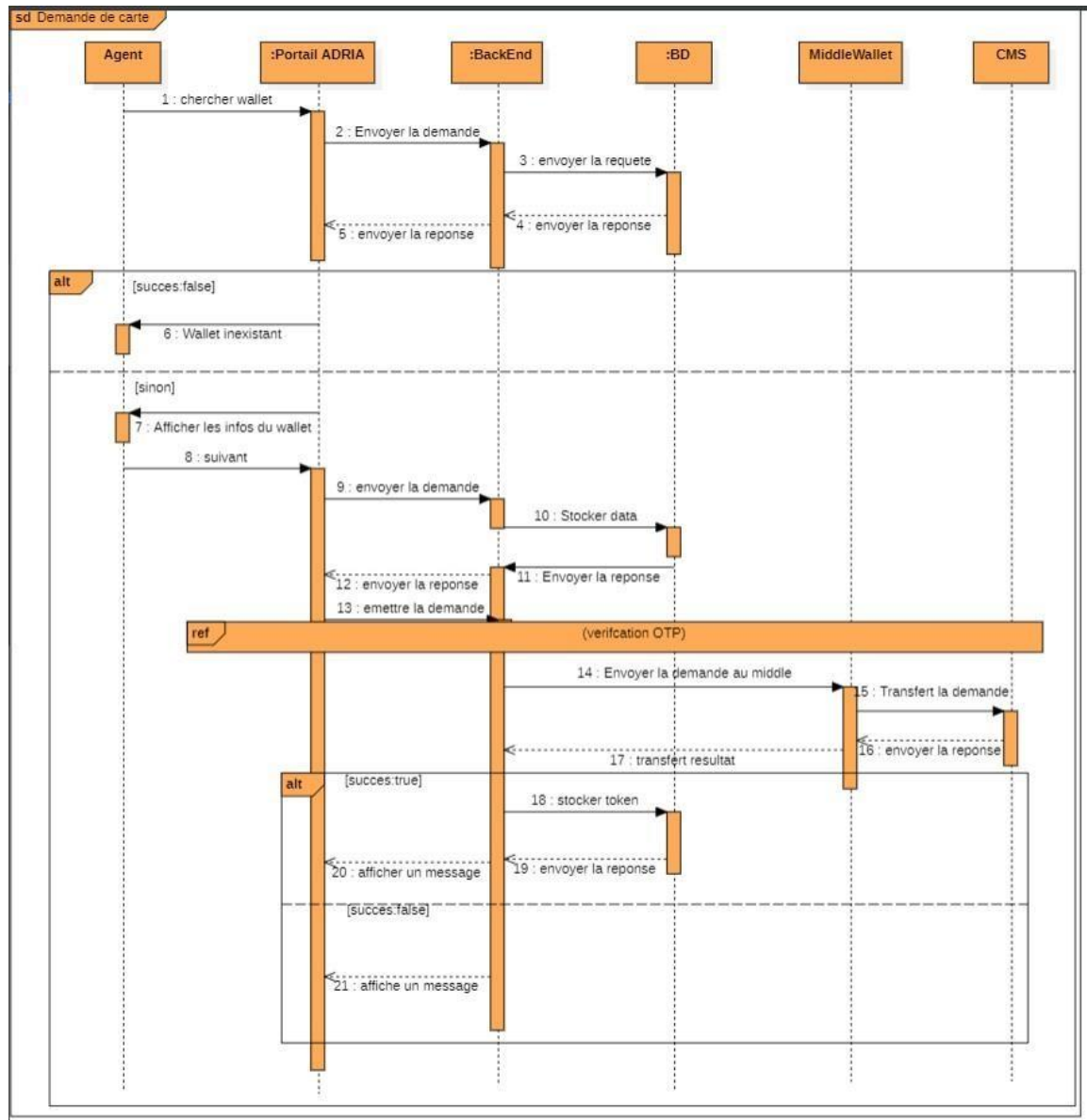


Figure 44: Diagramme de séquence du cas d'utilisation demande

### 3.3. Réalisation

Sachant que le système persiste toutes les opérations effectuées dans la base de données, l'agent peut récupérer l'ensemble des informations lié à un wallet en cliquant



sur le bouton de recherche existant dans le cadran «information du compte client» une fois saisi le numéro de téléphone. La figure suivante représente le premier cadran de la page information du compte client.

**Demande de Carte**

**Informations du compte client**

Téléphone: +212698501380 [Rechercher]

Téléphone: +212698501380

Nom: benabbes Prénom: souhail

RIB: 21269850 Adresse: CASABLANCA

Ville de naissance: CASA Pays de naissance: CASA

Date de naissance: 27-05-2021 Pièce d'identité: IDS89

[Retour] [Enregistrer]

Figure 45: le premier cadran de la page information du compte client.

Le deuxième cadran fournit un récapitulatif des informations du compte client afin que l'agent puisse valider la demande une fois la vérification de sms validée.

**Emission des cartes**

**Récapitulatif des informations**

Nom	Prénom	Téléphone	Pièce d'identité	RIB	Adresse
benabbes	souhail	+212698501380	IDS89	21269850	CASABLANCA
Date de naissance	Ville de naissance	Pays de naissance	Frais de Création	Statut	
27-05-2021	CASA	CASA	23 DH	Enregistre	

Saisir le code SMS reçu par le client: [ ] [Renvoyer le code]

[Valider la demande]

Figure 46: deuxième cadran de la page "récapitulatif des informations".

## 4. Sprint 4 : Gestion des cartes

### 4.1. Analyse

Afin de bien assimiler cette fonctionnalité, et pour comprendre son fonctionnement, le tableau suivant représente une analyse détaillée.

<b>Acteur</b>	Agent BO, Agent (détaillant, principales)
<b>Objectif</b>	Cette fonctionnalité permet aux acteurs d'accéder à la liste des cartes des clients et d'effectuer des recherches des cartes selon des critères.
<b>Description</b>	<p>Cette fonction servie via un écran de recherche via les critères suivants :</p> <p>(1 critère est obligatoire)</p> <ul style="list-style-type: none"> <li>• Numéro Téléphone ;</li> <li>• Numéro de Token ;</li> <li>• Statut ;</li> </ul> <p>Le bouton « Rechercher » renvoi la liste d'information suivante sous forme de tableau comportant les informations suivantes :</p> <ul style="list-style-type: none"> <li>• Numéro téléphone ;</li> <li>• Nom ;</li> <li>• Prénom ;</li> <li>• Pièce d'identité (Type de pièce d'identité + Numéro de pièce d'identité) ;</li> <li>• Numéro de compte RIB ;</li> <li>• Numéro de Token carte (unique est associé à toutes les cartes depuis la création jusqu'à la fin de sa vie fournie par le système monétique) ;</li> </ul> <p>Chaque ligne offre un Bouton « Action » : En cliquant ce dernier, l'écran « Fonction paramétrer une carte » est affiché.</p>
<b>Scénario</b>	<ul style="list-style-type: none"> <li>• L'agent s'authentifie en utilisant un login et mot de passe ;</li> <li>• Il accède à l'écran de gestion des cartes en appuyant sur le bouton « Gestion des cartes » dans le menu ;</li> </ul>

	<ul style="list-style-type: none"> <li>Le système charge la liste des cartes et les affiche l'écran dédié ; cette fonctionnalité.</li> <li>L'agent saisie un critère de recherche afin de personnaliser la recherche ;</li> <li>Il clique sur le bouton « rechercher » ;</li> <li>Le système affiche la liste des cartes qui répondent aux critères de l'agent.</li> </ul>
<b>Cas alternatif</b>	Si l'un des identifiants est incorrect, la page login se charge en affichant un message d'erreurs à l'utilisateur.

Tableau 6: Description du cas d'utilisation de « Gestion des cartes ».

### ❖ Diagramme de cas d'utilisation

Le diagramme suivant est dédié à la fonctionnalité de gestion des cartes qu'offre notre solution aux agents (BO, principal et détaillant) authentifiés, permettant de traiter la liste des cartes, ainsi qu'effectuer des recherches selon des critères.

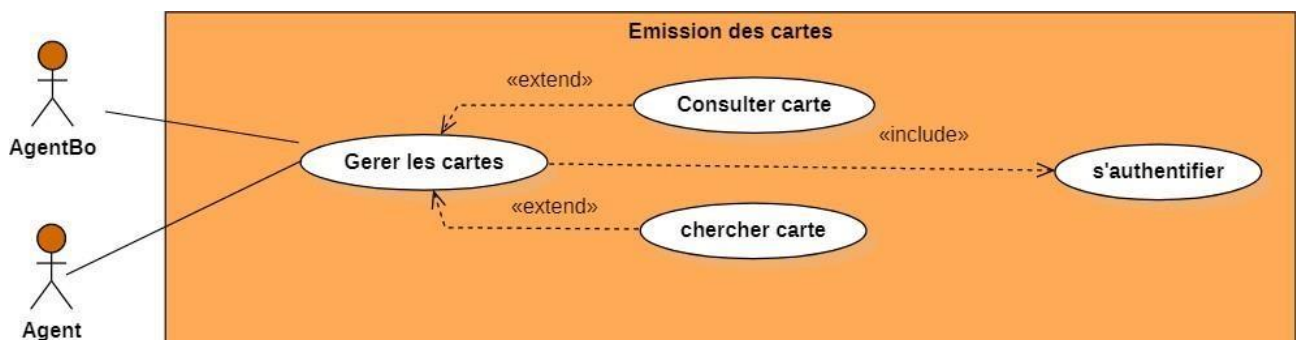


Figure 47:diagramme de cas d'utilisation de la fonctionnalité gestion des cartes.

## 4.2. Conception

### 4.2.1. Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entrées acteurs et le système selon un ordre chronologique dans la formulation UML. Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation.

La figure ci-dessous représente le diagramme de séquence du cas d'utilisation « gérer les cartes ».

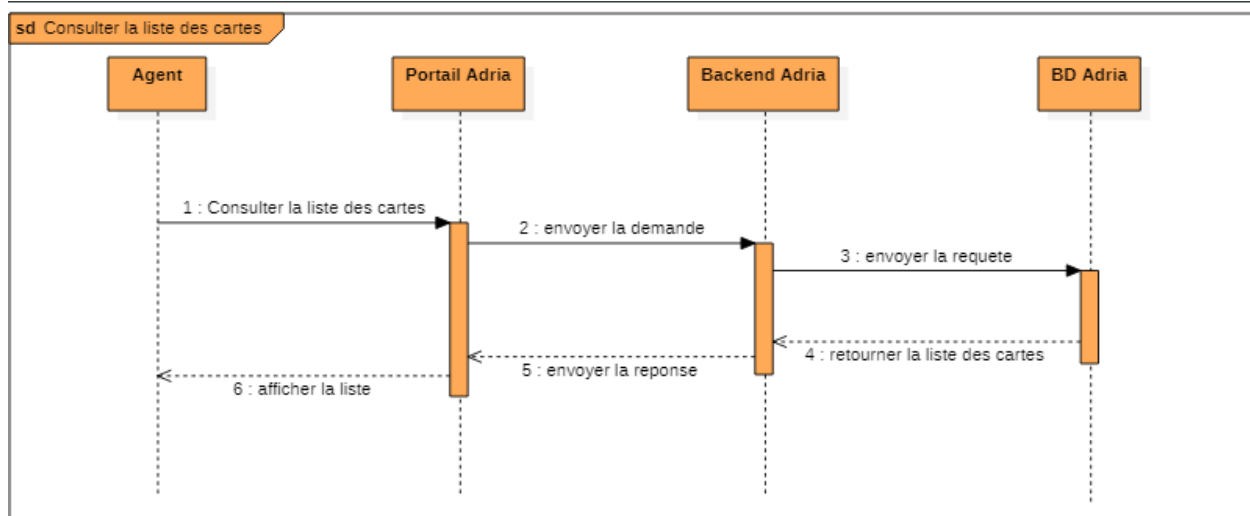


Figure 48: Diagramme de séquence de gestion des cartes.

#### 4.2.2. Diagramme de classe

La figure ci-dessous représente le diagramme de classe du cas d'utilisation « gérer les cartes » représentant les deux entités

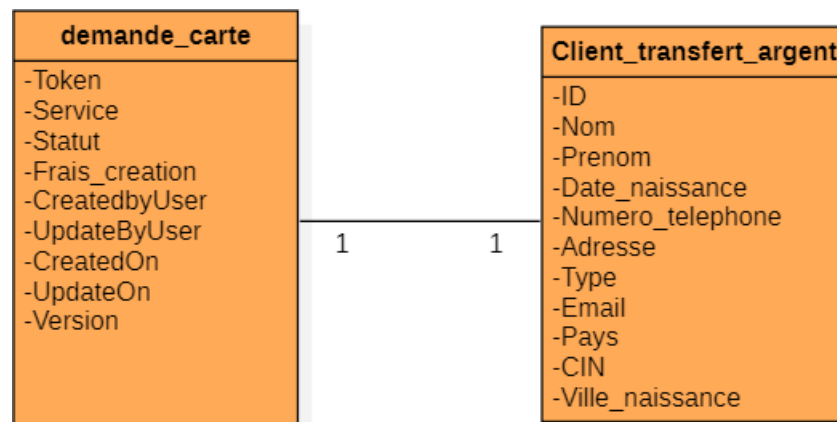


Figure 49: diagramme de classe de la gestion des cartes

### 4.3. Réalisation

La figure ci-dessous représente l'écran de gestion des cartes où l'agent peut consulter la liste des cartes personnalisée selon des critères qu'il peut renseigner.

The screenshot shows a web application interface titled "Emission de Carte". On the left is a vertical sidebar with icons for home, refresh, print, edit, and navigation. The main content area has a header "Emission de Carte" and a sub-header "CARTES". Below this is a search form with three input fields: "Téléphone", "Token de la carte", and "Statut" (with a dropdown menu labeled "Choisir un statut"). There are three buttons: "Rechercher" (with a magnifying glass icon), "Nouvelle carte" (with a plus icon), and "Réinitialiser" (with a reset icon). Below the form is a table with the following columns: TÉLÉPHONE, NOM, PRÉNOM, PIÈCE D'IDENTITÉ, DATE DE CRÉATION, TOKEN DE LA CARTE, ETAT DE LA CARTE, and ACTION. The table contains five rows of data.

TÉLÉPHONE	NOM	PRÉNOM	PIÈCE D'IDENTITÉ	DATE DE CRÉATION	TOKEN DE LA CARTE	ETAT DE LA CARTE	ACTION
+212698501380	benabbes	souhail	IDS89	09-06-2022	C4242342V4544	Commandee	
+212698501380	benabbes	souhail	IDS89	26-05-2022	C4242342V4544	Commandee	
+212772125560	pedrito smirss	pedrito	AD65225121	26-05-2022	C4242342V4544	Commandee	
+212660403950	TEST	TEST	1711677	26-05-2022	C4242342V4544	Commandee	
+212666603603	TANTAOUJ	ABDELALI	BL33477	26-05-2022	C4242342V4544	Commandee	

Figure 50: l'écran de gestion des cartes.

## 5. Sprint 5 : Paramétrage des cartes

### 5.1. Analyse

Le tableau suivant représente une analyse détaillée du cas d'utilisation « Paramétrer une carte » :

<b>Acteur</b>	Agent BO, Agent (détaillant, principales)
<b>Objectif</b>	Cette fonctionnalité permet aux acteurs de contrôler les différents paramètres liés au service, état et code PIN des cartes.
<b>Description</b>	<p>Cette fonction est une page d'affichage et de modification constituée de 4 cadrans (blocs) :</p> <p>❖ <i>Cadran 1 : Titre : Informations cartes et client</i></p> <p>Les informations à afficher dans ce cadran sont :</p> <ul style="list-style-type: none"> <li>• Numéro token carte ;</li> <li>• Etat de la carte ;</li> <li>• Nom du client ;</li> <li>• Prénom du client ;</li> </ul>

- Pièce d'identité (Type de pièce d'identité + Numéro de pièce d'identité)
- Numéro du compte ;
- Numéro de téléphone
- Adresse

❖ **Cadran 2 : Titre : Activation des services**

Le cadran est constitué des champs :

- Check box 1: Paiement E-Com
- Check box 2: Paiement TPE
- Check box 3: Retrait auprès des GABs BMCI
- Check box 4: Retrait auprès des GABs Confrère

L'agent pourra cocher ou décocher tous les check box.

- Un Bouton « Valider » : Lorsque ce bouton est pressé, un Appel de Web Service de mise à jour des services de la carte est envoyé au système monétique, une validation par OTP (un message envoyé au client contenant un code de validation) est recommandée pour finaliser la transaction.

❖ **Cadran 3 : Titre : Statut de la carte**

Ce cadran est constitué des champs suivants :

- Check box 1: Active ;
  - Check box 2: Bloque ;
- Bouton « Valider » : Lorsque ce bouton est pressé, un Appel de Web Service de mise à jour des statuts de la carte est envoyé au système monétique avec enregistrement des nouveaux paramètres.

❖ **Cadran 4 : Titre : Recalcule PIN**

Ce cadran présente un champ :

- Bouton « Recalcule de PIN »

<b>Scénario</b>	<ul style="list-style-type: none"> <li>• L'agent clique sur le bouton « visionner » correspondant à la carte qu'il veut paramétrer ;</li> <li>• L'écran « Paramétrer une carte » se charge, et le premier cadran sera affiché ;</li> <li>• Pour modifier les services de cette carte, L'agent clique sur le deuxième cadran, et coche le check box correspondant au service à activer, ou bien décocher le check box correspondant au service à désactiver ; le bouton « valider ».</li> <li>• Pour consulter ou changer l'état de la carte en question, l'agent accède au troisième cadran, et coche le check box correspondant au nouvel état souhaité ; il clique ensuite sur le bouton « valider », une validation par OTP (un message envoyé au client contenant un code de validation) est recommandée pour finaliser la transaction. ;</li> <li>• Pour le recalcule de PIN :                         <ul style="list-style-type: none"> <li>○ L'agent sélectionne le bouton « recalcule de PIN » ;</li> <li>○ Un sms est envoyé au client et le champ de saisi de l'OTP est affiché à l'agent.</li> <li>○ L'agent renseigne l'OTP et clique sur « Valider » ;</li> <li>○ En cas de OTP valide, l'écran appelle le web service de recalcule de pin du système monétique. En cas de retour positif, le client est débité du montant de frais de recalcule de pin.</li> </ul> </li> </ul>
<b>Cas alternatif</b>	<ul style="list-style-type: none"> <li>• Au cas où la carte est bloqué, cet écran sera grisé en entier en affichant un message indiquant que la carte est échoué ou résilié</li> </ul>

Tableau 7 : Description du cas d'utilisation « Paramétrer une carte ».



## ❖ Diagramme de cas d'utilisation

Le diagramme suivant regroupe l'ensemble d'opération qu'un agent peut effectuer sur les cartes grâce à cette fonctionnalité.

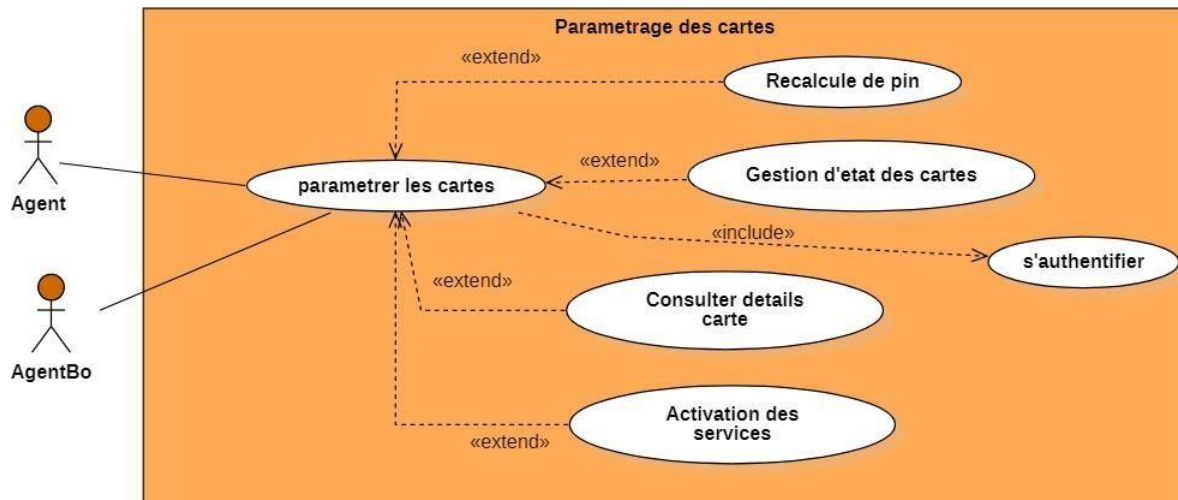


Figure 51: Diagramme de cas d'utilisation de la fonctionnalité Paramétrer une carte.

## 5.2. Conception

### 5.2.1. Activer/Désactiver des services :

#### 5.2.1.1 Diagramme de séquence

La figure suivante représente le diagramme de séquence du cas d'utilisation « Activer/désactiver un service ». Suite à la demande du client l'agent accède à l'écran « Paramétrer la carte » pour éditer la carte, une vérification de l'état de la carte se fait pour s'assurer que la carte n'est pas bloquée. Si c'est le cas, l'écran s'affiche grisée avec un message indiquant l'état de la carte dans le premier cadran. Si la carte est bien activée, l'écran s'affiche normalement et l'agent pourra accéder au deuxième cadran, pour qu'il coche/décoche la case du service afin de l'activer/désactiver. Les nouvelles informations de la carte seront envoyées au BackEnd-Adria pour qu'il envoie une demande de mise à jour de la carte au Middle. Ce dernier va la transmettre au service monétique, une validation par OTP nécessaire dans la dernière étape de l'opération.

La figure suivante indique le scénario d'activation d'un service.

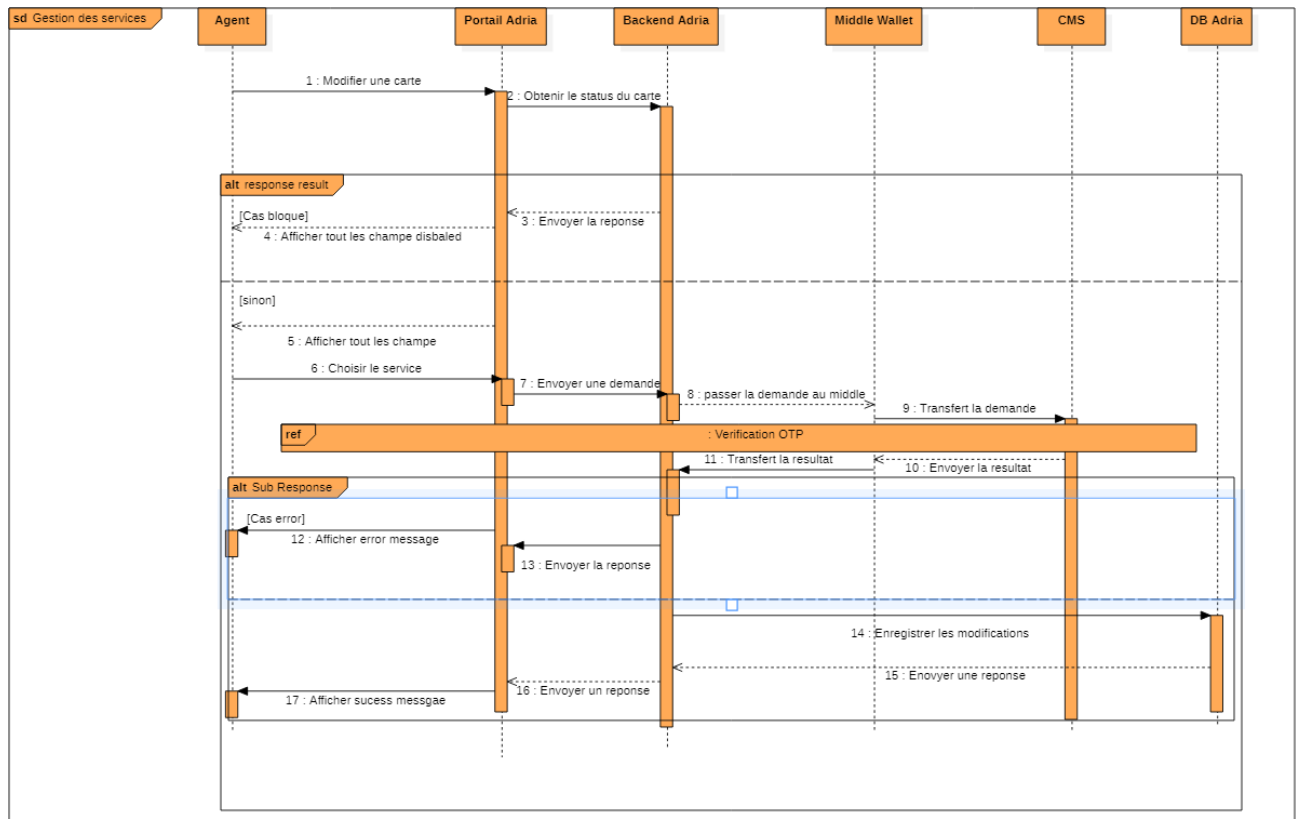


Figure 52: Diagramme de séquence pour le cas d'utilisation Activer service.

## 5.2.2. Etat de la carte

### 5.2.2.1. Diagramme de séquence

La modification de l'état des cartes est l'une des fonctionnalités les plus importantes dédiées aux agents. Il s'agit d'activer ou bloquer une carte suite à la demande d'un client.

Après avoir édité la carte en question, l'agent coche la case correspondante à l'état souhaité, et appuie sur le bouton valider. La demande de MAJ sera ensuite envoyée au Backend Adria qui va la transmettre au service monétique via le middle qui assure la communication entre le Backend et le service monétique.

La figure suivante représente le cas de gestion d'état d'une carte :

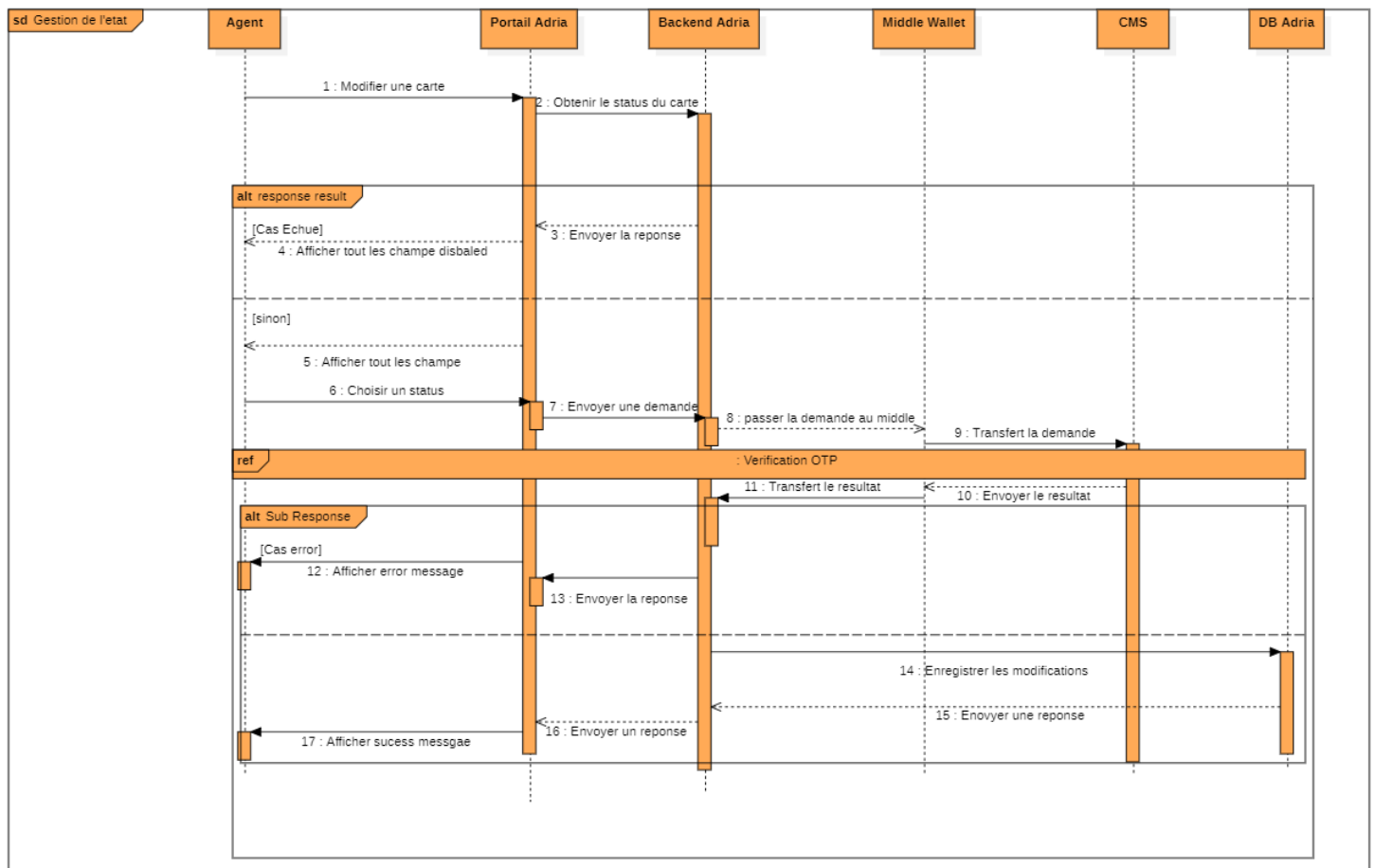


Figure 53: Diagramme de séquence de gestion d'état d'une carte.

Une vérification de l'OTP est obligatoire pour cette transaction.

### 5.2.3. Recalcul de PIN

#### 5.2.3.1. Diagramme de séquence

Le recalcul du pin est une fonctionnalité permettant à l'agent de modifier le code PIN d'une carte d'un client. Pour en profiter, il édite la carte en question, accède au cadran « Recalcul PIN », et sélectionne le bouton « recalculer ».

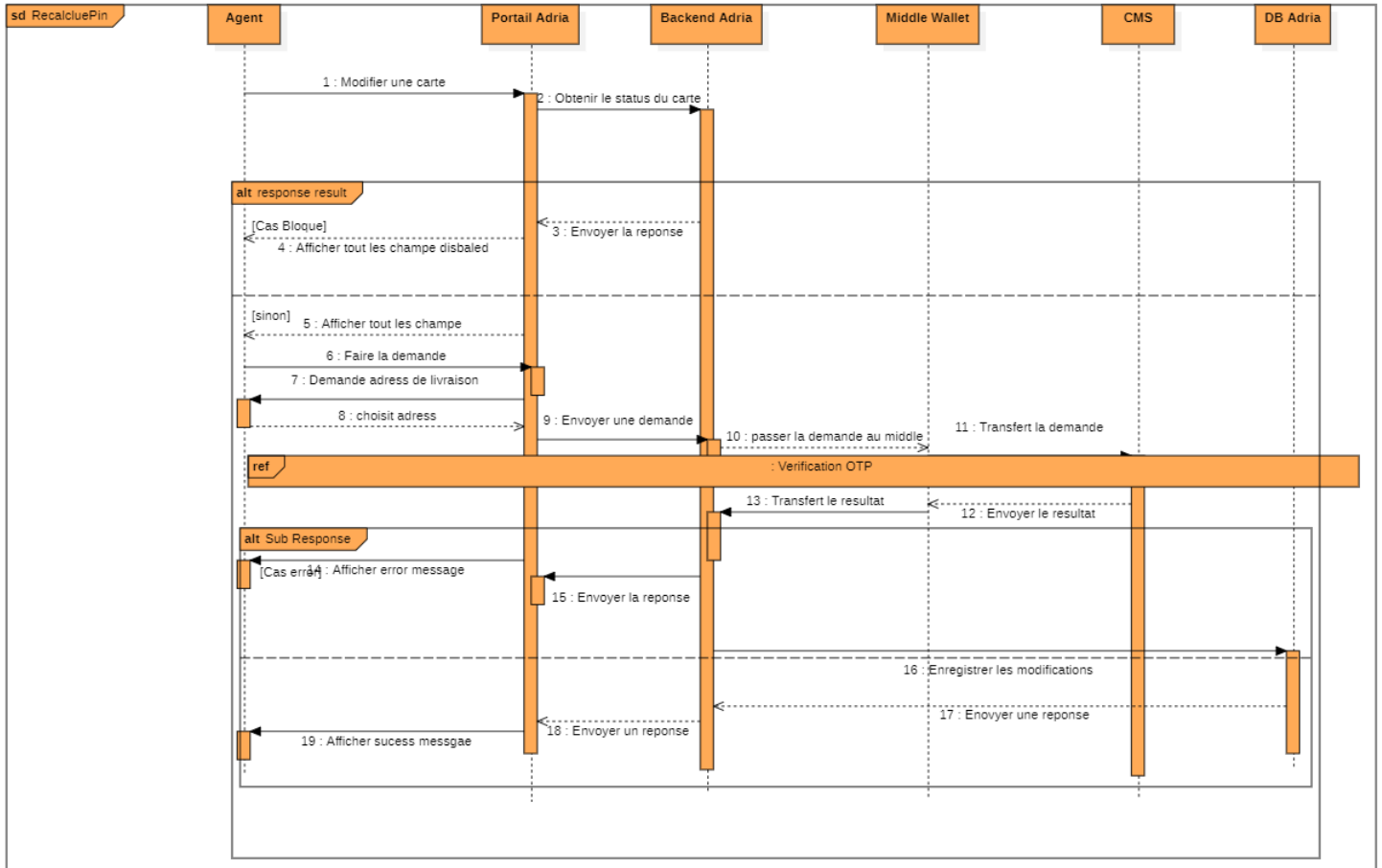


Figure 54: Diagramme de séquence de recalcul du PIN.

Une vérification de l'OTP est obligatoire pour cette transaction.

### 5.3. Réalisation

#### 5.3.1. Consulter les informations relatives aux cartes et aux clients.

La figure suivante représente le premier cadran de l'écran de paramétrage de la carte sélectionnée en cliquant sur le bouton d'affichage « Visionner » qui lui est correspondant. Il s'agit du cadran « Information Carte et client ».

Paramétrer la carte

INFORMATIONS CARTES ET CLIENT   ACTIVATION DES SERVICES   STATUT DE LA CARTE   RECALCULE PIN

Token de la carte: C4242342V4544	Nom: benabbes
Prénom: souhail	Etat de la carte: Commandée
Pièce d'identité: IDS89	Adresse: CASABLANCA
Numéro de compte: 21269850	Téléphone: +212698501380

Figure 55: cadran 1 de l'écran paramétrer la carte.

### 5.3.2. Activation des services

En cliquant sur « Activation services », le deuxième cadran s'affiche. Il contient une liste des services duquel le client peut profiter en utilisant cette carte : Paiement E-Com, Paiement TPE, Retrait auprès des GAB BMCI, Retrait auprès des GAB Confrère

Paramétrer la carte

INFORMATIONS CARTES ET CLIENT   ACTIVATION DES SERVICES   STATUT DE LA CARTE   RECALCULE PIN

Activation des services:

- ☒ Paiement E-Com
- ☐ Paiement TPE
- ☒ Retrait auprès des GABs BMCI
- ☐ Retrait auprès des GABs Confrère

✓ Enregistrer

Figure 56: cadran 2 de l'écran paramétrer la carte.

Pour le moment, le client ne peut utiliser cette carte que pour effectuer des paiements sélectionnés, or suite à sa demande, l'agent peut activer d'autre service pour cette carte en cochant la case correspondante, dans ce cas une validation par OTP est recommandée, ce qui est clair dans la figure suivante :

Figure 57: opération d'activation du service paiement E-Com et Retrait Gabs BMCI.

En cas de succès un message sera affiché a l'écran :

Figure 58: opération d'activation du service (cas Succès).

### 5.3.3. Changement d'état d'une carte

Le client peut utiliser sa carte, tant qu'elle est activée. La figure suivante représente le cadran « statut de la carte » d'une carte activée.

Figure 59: le cadran 3 de l'écran Paramétrer la carte.

Il existe des circonstances, où le client souhaite bloquer sa carte. Pour ce, il demande à son agent de changer l'état de carte.

### 5.3.4. Recalcule de PIN

Pour changer le code PIN d'une carte, notre solution lui offre la fonctionnalité « Recalcule PIN » utilisé par l'agent comme l'indique la figure suivante :

Figure 60: le cadran 4 de l'écran Paramétrer une carte.

Comme cette opération traite des données confidentielles, elle aussi recommande une validation par OTP.

The screenshot shows a web application interface titled 'Paramétrer la carte'. On the left is a vertical sidebar with icons for home, refresh, print, edit, and navigation. The main content area has four tabs: 'INFORMATIONS CARTES ET CLIENT', 'ACTIVATION DES SERVICES', 'STATUT DE LA CARTE', and 'RECALCULE PIN'. The 'RECALCULE PIN' tab is active. Below the tabs, there is a label 'Saisir le code SMS reçu par le client' above a text input field. To the right of the input is a blue button labeled 'Renvoyer le code'. Below the input field is a warning icon and the text 'Ce champ est obligatoire'. At the bottom right of the form is a blue button labeled 'Valider'.

Figure 61: vérification par OTP pour le recalcule de PIN.

Dans le cas du succès, le système affiche un message indiquant que la demande de recalcule du PIN a été bien enregistrée.

This screenshot shows the same 'Paramétrer la carte' interface as Figure 61, but with a success message displayed. The message 'La Demande de recalcule est bien été envoyée' is shown in a green box at the top of the main content area, below the tabs. The rest of the interface, including the sidebar and the 'RECALCULE PIN' tab, remains the same.

Figure 62: Recalcule de PIN (cas de succès).



## *Conclusion générale*

---

Ce travail s'inscrit dans le cadre de l'accomplissement de mon projet de fin d'études au cycle d'ingénieur à l'Ecole Normale Supérieure de l'Enseignement Technique de Mohammedia, effectué au sein de l'entreprise Adria Business & Technology. Nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de mettre en œuvre un module de gestion des cartes de paiement pour un établissement de paiement et de souscription par reconnaissance optique de caractère. Nous avons donc eu l'occasion de travailler sur une application dont le domaine est très répandu à l'intérieur et en dehors du Maroc, ce qui nous a permis d'enrichir nos connaissances et notre expérience sur le marché bancaire.

Durant le stage, nous étions amenés à effectuer plusieurs tâches au sein d'une équipe adoptant la méthodologie Agile pour la conduite de ces projets. Notre travail a été rapporté à couvrir plusieurs aspects compris la spécification fonctionnelle, la conception, le développement et l'intégration.

En tant qu'étudiant sur le point d'intégrer le monde du travail, la gestion du projet, le travail en groupe et la prise d'initiative ont été de puissants leviers du bénéfice de mon stage, et ainsi un impact direct sur mon développement personnel et professionnel.

## *Références*

---

### *Webographie*

- ✓ <https://fintech.ma/nouvelle-loi-bancaire-la-r%C3%A9volution-des-fintech-au-maroc-d9ec8af7f341>
- ✓ <https://lematin.ma/express/2018/etablissements-paiement-cinq-agrements-octroyes-bam/297350.html>
- ✓ <https://adria-bt.com/>
- ✓ <https://spring.io/>
- ✓ <https://getbootstrap.com/>
- ✓ <https://www.oracle.com/>
- ✓ <https://www.challenge.ma/place-aux-etablissements-de-paiement-76068/>