

RAPPORT DE STAGE DE FIN D'ETUDES

5ème Année en Ingénierie Informatique et Réseaux

Conception et Développement d'une Plateforme e-Banking Intelligente et Sécurisée pour la Gestion Avancée des Cartes Visa, Basée sur l'IA et une Architecture Microservices, Intégrée à PowerCARD

Réalisé par :

Rhalim Sami
Senhaji Rhandour Nada

Tuteurs :

Encadrant Professionnel : Mr . Abdelilah LAMHAZET

Mr. Omar BENJELLOUN TOUIMI

Encadrant Pédagogique : Mme. Sofia BELKHALA

Mr. Abdellah OUAGUID

Mr. Abdesllam OUARDI

Au sein de HPS :



DEDICACE

Nous dédions ce modeste travail à :

Nous dédions ce travail, fruit de plusieurs semaines d'efforts, de recherches, de réflexions et de collaboration, à toutes les personnes qui ont joué un rôle essentiel dans notre parcours.

À nos familles, pour leur amour inconditionnel, leur soutien moral et matériel, et leur patience dans les moments de doute. Leur présence silencieuse mais constante a été une source de motivation précieuse. Leur confiance en nos capacités nous a poussés à donner le meilleur de nous-mêmes.

À nos professeurs et encadrants, pour leur accompagnement bienveillant, leur rigueur intellectuelle et leur générosité dans le partage du savoir. Grâce à leur encadrement, nous avons pu transformer des idées en un travail structuré et enrichissant. Leur passion pour l'enseignement et leur exigence académique nous ont permis de grandir, non seulement en compétences, mais aussi en maturité professionnelle.

À nos camarades et amis, pour les discussions stimulantes, l'entraide spontanée, les échanges d'idées constructifs et le climat de solidarité qui a rendu ce travail collectif plus riche et plus agréable. Chaque interaction a contribué, d'une manière ou d'une autre, à faire émerger une version plus aboutie de ce rapport.

Enfin, nous dédions ce travail à tous ceux qui croient en l'effort, la persévérance, le travail d'équipe et l'envie d'apprendre. Ce projet n'est pas seulement une production académique, c'est le reflet d'un engagement commun, d'un apprentissage continu et d'un pas de plus vers notre avenir professionnel.

REMERCIEMENTS

Avant tout, nous remercions Dieu le Tout-Puissant de nous avoir donné la force, la patience et le courage nécessaires pour mener à bien ce travail.

Nous exprimons notre profonde gratitude à nos parents pour leur soutien inconditionnel, leurs encouragements constants et leurs sacrifices, sans lesquels nous n'aurions pas pu arriver jusqu'ici.

Nous tenons à adresser en premier lieu nos sincères remerciements et notre reconnaissance à notre encadrant professionnel, **M. Abdelilah LAMHAZET**, pour son accompagnement tout au long de la période de stage, ainsi que pour ses directives, ses conseils et la confiance qu'il nous a accordée. Sans son encadrement, ce travail n'aurait pas été ce qu'il est.

Nos remerciements vont également à **M. Omar BENJELLOUN TOUIMI**, ingénieur, pour l'assistance précieuse qu'il nous a apportée tout au long du développement de notre application. Son expertise technique, sa disponibilité et son engagement ont été d'un grand soutien face aux défis rencontrés.

Nous exprimons également toute notre gratitude à **Mme Sofia BELKHALA**, notre encadrante pédagogique, pour son encadrement attentif, sa disponibilité et ses conseils pertinents tout au long de cette expérience.

Nous remercions les membres du jury pour avoir accepté d'évaluer notre travail avec bienveillance.

Nos remerciements s'adressent aussi à tous nos professeurs pour la qualité de leur enseignement et les efforts fournis durant notre formation, ainsi qu'à l'ensemble du corps administratif de l'**EMSI** pour leur appui tout au long de notre parcours.

Enfin, un grand merci à nos amis(es) et à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet.

RESUME

Le présent rapport présente les travaux réalisés dans le cadre de notre stage de fin d'études chez **HPS**, en vue de l'obtention du **Diplôme d'Ingénieur d'État en Ingénierie Informatique et Réseaux, option MIAGE**, à l'**EMSI**.

Notre mission a été de concevoir et développer une **plateforme e-Banking intelligente et sécurisée** pour la gestion avancée des cartes bancaires, intégrant une **architecture microservices**, des **APIs PowerCARD (PAPI/SAPI)**, et des fonctionnalités innovantes basées sur l'**intelligence artificielle**.

La solution propose :

- Des notifications en temps réel via **Kafka**
- Une sécurité renforcée avec **OAuth2, JWT, et MFA**
- Une classification intelligente des transactions par **IA**
- Une interface moderne et responsive en **React et Flutter**
- Un système de **caching avec Redis** et des **APIs REST/GraphQL**

Pour le volet IA, nous avons utilisé :

- **Pandas, NumPy, Scikit-learn, LightGBM**
- **DistilBERT** (via Hugging Face) pour l'analyse sémantique
- **PGVector** pour la recherche vectorielle
- **RAG** pour la génération de réponses enrichies dans le chatbot RH

Enfin, la plateforme a été testée, documentée, et validée selon des critères de performance, sécurité et qualité logicielle.

Mots-clés :

e-Banking – Microservices – Kafka – Sécurité – Flutter – Intelligence Artificielle – PowerCARD – PAPI/SAPI – JWT – OAuth2 – Redis – RAG – PGVector – LightGBM – DistilBERT – React – GraphQL

ABSTRACT

This report presents the work carried out as part of our final-year internship at **HPS**, in pursuit of the **State Engineering Degree in Computer Science and Networks, MIAGE option, at EMSI**.

Our mission was to design and develop a **smart and secure e-Banking platform** for advanced bank card management, based on a **microservices architecture**, integrated with **PowerCARD APIs (PAPI/SAPI)**, and enriched with **AI-powered features**.

The platform provides:

- Real-time notifications via **Kafka**
- Strong authentication using **OAuth2, JWT, and MFA**
- AI-based classification and analysis of transactions
- A modern and responsive interface using **React and Flutter**
- Efficient performance with **Redis caching and REST/GraphQL APIs**

For the AI components, we worked with:

- **Pandas, NumPy, Scikit-learn, LightGBM**
- **DistilBERT** (via Hugging Face) for semantic transaction label encoding
- **PGVector** for vector-based semantic search
- **RAG (Retrieval-Augmented Generation)** to power our intelligent HR chatbot

The entire platform was tested, documented, and validated through functional, performance, and security benchmarks.

Keywords:

e-Banking – Microservices – Kafka – Security – Flutter – Artificial Intelligence – PowerCARD – PAPI/SAPI – JWT – OAuth2 – Redis – RAG – PGVector – LightGBM – DistilBERT – React – GraphQL

LISTE DES ABREVIATIONS

Abréviation	Désignation
API	Application Programming Interface
JWT	JSON Web Token
HPS	Hightech Payment Systems
BD	Base de Données
CSS	Cascading Style Sheets
UML	Unified Modeling Language
EMSI	Ecole marocaine des sciences de l'ingénieur
RAG	Retrieval-Augmented Generation
OAuth2	Open Authorization 2.0
UX/UI	User Experience / User Interface
PGVector	PostgreSQL Vector Store
TQDM	taqaddum
LightGBM	Light Gradient Boosting Machine

liste des figures

Figure 1 : Emplacements de HPS.....	4
Figure 2 : Organigramme de HPS	6
Figure 3 : Architecture de PowerCard [HPS-worldwide]	7
Figure 4:Diagramme de Gant.....	13
Figure 5 : Chaine de valeur de paiement	16
Figure 6 : L'intervention des solutions PowerCARD dans la chaîne monétique.....	18
Figure 7 : Cycle de vie modèle spiralé	28
Figure 8 : Logo Uml.....	30
Figure 9 : Diagramme de cas d'utilisation Cardholder.....	31
Figure 10 : Diagramme de cas d'utilisation Agent.....	33
Figure 11 : Diagramme de class	35
Figure 12:diagramme de séquence d'authentification et de sécurité.....	37
Figure 13: diagramme de séquence Gestion des cartes.....	39
Figure 14 : Diagramme de séquence Travel Plan :Titulaire & Agent	41
Figure 15: Diagramme de séquence Transaction :Titulaire & Agent	43
Figure 16: Diagramme de séquence Chatbot.....	44
Figure 17 :Architec ture Générale	54
Figure 18: Architecture du Model de Transactions	55
Figure 19 : Landing Pages	59
Figure 20: Loging and Touch ID.....	60
Figure 21: Home	61
Figure 22 : Notifications.....	62
Figure 23 : Transactions.....	63
Figure 24 : Menu	64
Figure 25 : Profile	65
Figure 26: Cards	66
Figure 27: Add Card.....	67
Figure 28: Pysical Cards /CVV/PIN.....	68
Figure 29 :Card Details	69
Figure 30: Block Phisical Card.....	70
Figure 31:Request New Card	71
Figure 32: Cancel Card.....	72
Figure 33: Virtual Card Details	74
Figure 34:Request New Virtual Card	75
Figure 35 :Update CVV and PIN.....	76
Figure 36: Travel Plan	77
Figure 37: Contact Us	79
Figure 38 : Complain.....	80
Figure 39: Settings.....	80
Figure 40 :Policy	81
Figure 41 :Cards Pack.....	81
Figure 42 :Reset Password.....	82
Figure 43: ChatBot.....	83
Figure 44 :Loging Agent.....	84

Figure 45:All Cards	85
Figure 46 :Add New Card	86
Figure 47: Physical Card Details.....	87
Figure 48: Cancel and Uncancel Card	88
Figure 49 : Block / Unblock Cards.....	89
Figure 50:Generate PIN et CVV	90
Figure 51 : Virtual Card Details	90
Figure 52: Cancel /Uncancel Virtual Card	91
Figure 53 :Block / Unblock Virtual Card.....	92
Figure 54:Transactions	92
Figure 55 :Transaction Details.....	93
Figure 56: Analyse des Transactions Bancaires par Catégorie, Revenu et Précision du Modèle	94
Figure 57:Gestion des Travel Plan	95
Figure 58: Gestion des Utilisateurs.....	96
Figure 59 :Notifications – Suivi en temps réel.....	97

Table des matières

<i>DEDICACE</i>	<i>ii</i>
<i>REMERCIEMENTS.....</i>	<i>iii</i>
<i>RESUME.....</i>	<i>iv</i>
<i>ABSTRACT</i>	<i>v</i>
<i>LISTE DES ABREVIATIONS.....</i>	<i>vi</i>
<i>Chapitre I.....</i>	<i>2</i>
<i>I. Introduction</i>	<i>3</i>
<i>II. Présentation de l'organisme d'accueil.....</i>	<i>3</i>
1. Présentation de HPS	3
2. Historique	4
3. Organigramme de HPS.....	6
4. Présentation des secteurs d'activités.....	6
<i>III. Cadre général du projet.....</i>	<i>8</i>
1. Contexte du projet	8
2. Problématique.....	9
3. Objectifs.....	9
4. Solution Proposé	10
5. Méthodologie.....	10
6. Planification	11
Conclusion	13
<i>Chapitre II.....</i>	<i>14</i>
<i>I. Introduction</i>	<i>15</i>
<i>II. Introduction à la monétique.....</i>	<i>15</i>
1. Introduction à la monétique	15
2. Le déroulement d'une transaction électronique.....	15
3. La solution monétique PowerCard.....	16
<i>III. Contexte du Projet.....</i>	<i>18</i>
1. Contexte du Projet.....	18
2. Problématique.....	20
3. Objectifs.....	20
<i>IV. Besoins fonctionnels et non fonctionnels</i>	<i>22</i>
1. Besoins fonctionnels	22

2. Besoins non fonctionnels	23
V. Développement de la plateforme e-Banking.....	24
1. Introduction.....	24
2. Objectifs du développement	24
3. Bénéfices apportés par la solution	25
4. Défis Techniques	25
5. Critères de priorisation fonctionnelle	25
6. Bibliothèques et outils d'intelligence artificielle utilisés.....	26
7. Méthodologie Générale du projet.....	27
7.1. Choix de la méthodologie.....	27
7.2. Pourquoi le modèle spiralé ?.....	27
7.3. Schéma du cycle de vie adopté	28
Conclusion	28
Chapitre III	29
I. UML et son rôle dans la modélisation et la conception.....	30
1. Diagrammes de modélisation UML	31
1.1. Diagramme de Cas d'utilisation.....	31
a. Diagramme de Cas d'utilisation Cardholder	31
b. Diagramme de Cas d'utilisation Agent	33
1.2. Diagramme de class.....	34
1.4. Diagramme de séquence – Gestion des cartes : interactions Titulaire & Agent	38
1.5. Diagramme de séquence Travel Plan :Titulaire & Agent	41
1.6. Diagramme de séquence Transaction :Titulaire & Agent	42
1.7. Diagramme de séquence Chatbot	44
Conclusion	44
II. Outils et Technologie.....	45
1. Technologie et Outils	45
2. Environnement de développement	52
III. Architecture technique de la plateforme.....	53
1. Introduction.....	53
2. Architecture proposée.....	54
1.8. Architecture Générale	54
1.9. Architecture du Model de Transactions	55
Conclusion	57
Chapitre IV.....	58
I. Introduction	59
II. Réalisation de l'application mobile – Espace Cardholder	59
1. Landing page	59

2. Loging	60
3. Home Page	61
4. Notification.....	62
5. Transactions.....	63
6. Menu	64
7. Personal infos.....	65
8. Cards.....	66
9. Add Cards.....	67
10. Physical Cards	68
11. Virtual Cards	74
12. Travel Plan.....	77
13. Contact Us.....	79
14. Complain.....	80
15. Settings.....	80
16. Policy.....	81
17. Card Packs.....	81
18. Reset Password	82
19. Chatbot	83
III. Réalisation de l'application Web – Espace Agent.....	84
1. Loging	84
2. Gestion des Cartes	85
3. Physical Card Details.....	87
4. Cancel and Uncancel Card	88
5. Block et Unblock Card.....	89
6. Generate PIN and CVV	90
7. Virtual Card Details	90
8. Cancel and Uncancel Virtual Card.....	91
9. Block et Unblock Virtual Card.....	92
10. Transcation	92
11. Transaction Details	93
12. Analyse des Transactions Bancaires par Catégorie, Revenu et Précision du Modèle	93
13. Gestion des Travel Plan	94
14. Gestion des Utilisateurs – Interface Agent	96
15. Notification– Interface Agent	97
Conclusion	98

<i>Conclusion Générale</i>	99
<i>Perspectives.....</i>	100
<i>Webographie</i>	101

I. INTRODUCTION GENERALE

Le secteur de la monétique connaît une transformation rapide vers une société sans numéraire, où les cartes bancaires, physiques ou virtuelles, dominent désormais les moyens de paiement. Cette évolution s'accompagne d'une demande croissante en solutions digitales sécurisées, performantes et adaptées aux nouveaux usages des clients bancaires. Dans ce contexte, l'innovation technologique devient un levier essentiel pour renforcer la compétitivité des institutions financières.

HPS (Hightech Payment Systems), acteur de référence en Afrique et à l'échelle internationale, répond à ces enjeux à travers sa solution **POWERCARD**, utilisée par plus de 450 institutions financières dans le monde. Cette plateforme complète et modulaire couvre l'ensemble de la chaîne monétique, de l'émission à l'acquisition, tout en assurant un haut niveau de sécurité et de flexibilité.

C'est dans ce cadre que s'inscrit notre **projet de fin d'études**, réalisé au sein de **HPS**, et portant sur la conception et le développement d'une **plateforme e-Banking intelligente et sécurisée** pour la gestion avancée des cartes bancaires. Le projet intègre une architecture microservices, des API PowerCARD pour la synchronisation des données, ainsi que des fonctionnalités basées sur l'intelligence artificielle telles que les notifications intelligentes ou la détection de fraude.

Cette expérience nous a permis de mettre en pratique nos compétences techniques dans un environnement professionnel exigeant, tout en contribuant à la création d'un produit bancaire moderne et évolutif.

Ce rapport est structuré en cinq chapitres : le premier présente le contexte général du projet, le deuxième propose une analyse fonctionnelle, le troisième décrit la méthodologie suivie, le quatrième aborde les technologies et l'architecture utilisées, et enfin le cinquième détaille la mise en œuvre de la solution.

Chapitre I :

Contexte général

I. Introduction

Ce chapitre vise à fournir un aperçu général du contexte de notre projet de fin d'études. Tout d'abord, nous présenterons l'industrie de la monétique, en expliquant le processus d'une transaction électronique ainsi que les acteurs impliqués. Ensuite, nous ferons une présentation de l'organisme d'accueil, en mettant en évidence ses solutions et son secteur d'activité.

II. Présentation de l'organisme d'accueil

1. Présentation de HPS

Hightech Payment Systems (HPS) est une entreprise marocaine de dimension internationale, spécialisée dans le développement de solutions de paiement électronique à destination des banques, processeurs, switches nationaux et régionaux à travers le monde. Fondée en 1995 par un collectif d'experts et de consultants marocains en monétique, HPS avait pour ambition de concevoir une plateforme monétique ouverte, modulable et capable de répondre à l'ensemble des besoins liés aux transactions de paiement, quels que soient les marchés cibles.

Aujourd'hui, le groupe étend ses activités sur les cinq continents et intervient dans plus de 85 pays répartis en Europe, Asie, Océanie, Afrique et Amérique, en s'appuyant sur une équipe internationale de plus de 300 spécialistes. Sa suite logicielle PowerCARD est déployée auprès de plus de 400 clients parmi les émetteurs, acquéreurs et opérateurs de switchs, permettant le traitement de multiples types de cartes (crédit, débit, prépayées, cartes d'entreprise, de fidélité ou encore de carburant), et ce, sur l'ensemble des canaux de paiement (DAB, terminaux de paiement, commerce en ligne et mobile). Cette solution intégrée offre aux institutions financières une maîtrise complète et sécurisée de leurs transactions électroniques.

Présente dans plus de 90 pays et utilisée par des acteurs de premier plan figurant notamment parmi les classements « Fortune 500 » et « Forbes World's Biggest Companies », la solution PowerCARD couvre l'ensemble de la chaîne de valeur des paiements. Elle intègre sur une plateforme unique les fonctions liées à l'émission, l'acquisition, le switching ainsi que les différentes opérations monétiques. Grâce à sa flexibilité et à ses capacités de personnalisation avancées, PowerCARD permet à ses utilisateurs d'adapter rapidement leurs services à l'évolution du marché, tout en réduisant les coûts liés à

l'intégration, au développement et à la formation. Par ailleurs, la solution est certifiée PA-DSS et accréditée par l'ensemble des organismes de paiement internationaux tels que Visa, MasterCard, Diners Club, American Express et China Union Pay, garantissant ainsi son interopérabilité avec les principaux réseaux mondiaux.

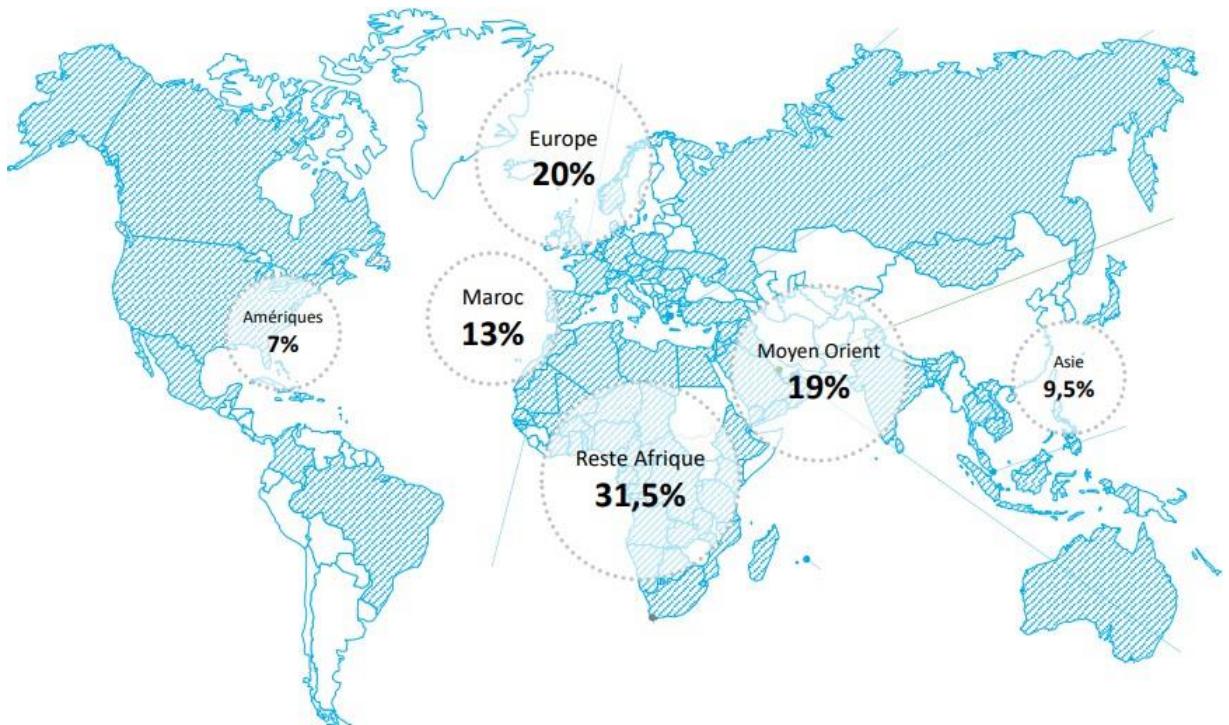


Figure 1 : Emplacements de HPS

2. Historique

L'histoire de HPS démarre en janvier 1995, lorsque quatre spécialistes marocains de la monétique, rejoints par quatre autres partenaires, décident de se lancer dans un secteur prometteur : celui des paiements électroniques. Leur premier projet consiste à réaliser une étude sur l'interopérabilité monétique au Maroc, qui servira de base au développement de PowerCARD, la solution phare de l'entreprise.

En 1996, HPS signe son premier contrat avec la Société Générale Marocaine de Banques, puis amorce son expansion internationale en équipant la BKME au Koweït. Cette ouverture précoce vers l'international illustre l'ambition globale de l'entreprise.

Entre 1997 et 2000, HPS diversifie sa clientèle en travaillant avec des banques et des sociétés de crédit. En 2000, l'entreprise décroche un contrat stratégique avec JCC à Chypre, renforçant sa crédibilité sur la scène internationale.

En 2001, Diner's Club, acteur majeur des paiements regroupant plusieurs millions de partenaires, choisit PowerCARD pour gérer huit centres monétiques en Europe. Dans la foulée, HPS élargit ses activités au secteur des télécommunications.

En 2002, le Maroc adopte PowerCARD comme solution nationale interbancaire de traitement des paiements. La même année, l'entreprise lève 40 millions de dirhams auprès de fonds d'investissement pour soutenir sa croissance et lancer la deuxième version de sa plateforme.

À partir de 2003, HPS accélère son développement international en ouvrant une filiale à Dubaï pour couvrir l'Asie et le Moyen-Orient. En 2004, l'entreprise obtient la certification ISO 9001, signe des contrats avec la BEAC et crée la coentreprise Global Payment Systems à Bahreïn avec Credimax, afin de proposer des services monétiques régionaux.

En 2005, la BCEAO confie à HPS la gestion de son centre monétique régional en Afrique de l'Ouest. L'entreprise débute également son activité sur le marché des banques islamiques avec des solutions conformes à la finance islamique.

En 2006, HPS remporte des contrats majeurs : en Libye pour la gestion du centre monétique national, et au Japon avec Acom, l'une des plus grandes sociétés de crédit. Cette même année, l'entreprise s'introduit en Bourse à Casablanca et reçoit le Trophée d'Excellence de l'OMPI en reconnaissance de son innovation.

En 2007, HPS renforce sa présence au Japon avec Shinsei Bank, se développe en Amérique du Nord grâce à un partenariat avec Carta Solutions au Canada, et crée ICPS dans l'océan Indien avec la Mauritius Commercial Bank.

Entre 2008 et 2010, l'entreprise poursuit sa croissance avec une filiale à Paris pour piloter l'activité européenne et signe un partenariat avec Embratec au Brésil, marquant son entrée en Amérique latine.

Aujourd'hui, PowerCARD est utilisé par plus de 450 institutions financières à travers le monde, positionnant HPS comme un leader mondial des solutions de paiement électronique.

3. Organigramme de HPS

L'organisation d'HPS se découvre au travers cet organigramme qui présente une vue d'ensemble de la répartition des postes et des fonctions au sein d'HPS.

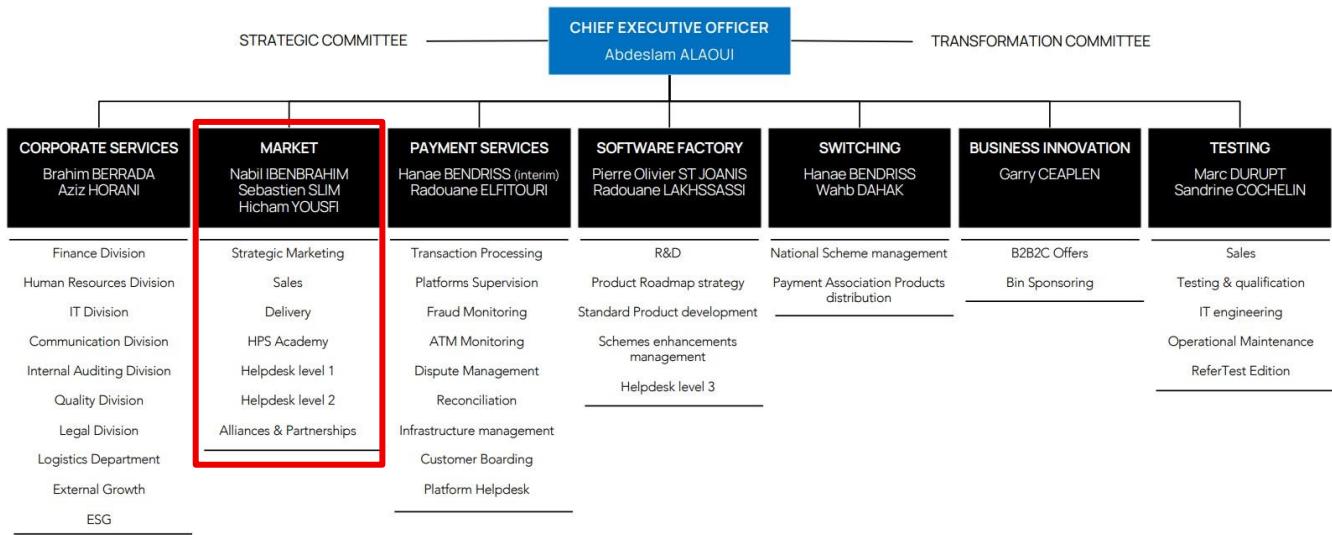


Figure 2 : Organigramme de HPS

4. Présentation des secteurs d'activités

Le progiciel Power CARD offert par HPS est une solution de paiement électronique évoluée et hautement sécurisée. Cette solution couvre l'intégralité de la chaîne de traitement les transactions financières liées au paiement électronique. Le produit est structuré en trois suites de solutions :

FrontOffice : Cette suite gère le pilotage des terminaux et guichets automatiques. Elle assure notamment les fonctions de sécurité et de vérification, la communication avec les moyens d'acquisition (TPE, GAB, Internet, etc.), la collecte et l'autorisation des transactions ainsi que la gestion des commerçants et des parcs de terminaux d'acquisition, etc.

Middle-Office : Cette deuxième suite assure le lien entre l'acquéreur et l'émetteur pour l'octroi d'autorisations, et le routage dans le cadre d'une interopérabilité nationale et/ou internationale, etc.

Back-Office : C'est la suite qui assure la gestion de l'émission de tout type de cartes (crédit, débit, prépayée, fidélité, etc.), la gestion de la personnalisation des cartes, la gestion de l'acquisition par les commerçants, la compensation (nationale et internationale), la gestion des litiges, le reporting, l'alimentation de la comptabilité des émetteurs et acquéreurs, etc.

HPS propose à ses clients des produits standards, néanmoins adaptables à leurs demandes spécifiques (langues, réglementations locales, intégration avec les systèmes d'information existants, méthodes de calcul des intérêts, etc.). Ce choix stratégique est mis en œuvre dans le cadre d'une politique cohérente en matière d'architecture des produits PowerCard et d'accompagnement, par des prestations efficaces et personnalisées. La figure 8 illustre l'architecture de la solution PowerCard d'HPS.



Figure 3 : Architecture de PowerCard [HPS-worldwide]

III. Cadre général du projet

Dans cette partie du chapitre on présentera le contexte du projet à travers la problématique, les objectifs attendus et la conduite du projet.

1. Contexte du projet

Notre mission principale lors de notre stage chez HPS était de concevoir et développer une plateforme e-Banking intelligente et sécurisée dédiée à la gestion avancée des cartes bancaires, intégrée à la solution PowerCARD.

Pour mener à bien cette mission, les étapes suivantes ont été suivies :

- Analyse des besoins fonctionnels et techniques : Identifier les fonctionnalités essentielles (activation, blocage, plafonds, notifications) et comprendre les exigences d'intégration avec PowerCARD via les APIs PAPI/SAPI.
- Conception de l'architecture microservices : Définir une architecture modulaire et évolutive, séparant les services (cartes, utilisateurs, sécurité, notifications) et prévoyant leur communication asynchrone.
- Mise en place de l'environnement de développement : Installer et configurer les outils nécessaires (Spring Boot, React, Flutter, Kafka, PostgreSQL) pour le développement des différentes composantes.
- Développement des interfaces web et mobiles : Créer des écrans modernes, ergonomiques et responsives, permettant la gestion complète des cartes bancaires depuis un navigateur ou une application mobile.
- Intégration avec les APIs PowerCARD : Implémenter la synchronisation des opérations et des données sensibles via PAPI/SAPI, tout en assurant la sécurité des échanges (OAuth2, JWT, HTTPS).
- Développement des fonctionnalités d'intelligence artificielle : Concevoir des modèles permettant la classification automatique des transactions bancaires et l'optimisation du suivi utilisateur.
- Mise en place du chatbot intelligent : Déployer un assistant conversationnel pour guider les utilisateurs et répondre à leurs questions en temps réel.
- Développement du système de notifications en temps réel : Intégrer Kafka pour générer et diffuser des notifications après chaque action sur une carte.
- Tests et validation de la plateforme : Vérifier la conformité, la stabilité, la sécurité et les performances dans différents scénarios utilisateurs.

Cette mission nous a permis de mobiliser nos compétences en développement full stack, architecture logicielle, sécurité des systèmes bancaires et intelligence artificielle, dans un contexte professionnel et exigeant.

2. Problématique

Dans un contexte de transformation digitale accélérée, les clients des établissements bancaires attendent des solutions e-Banking qui soient à la fois simples d'utilisation, sécurisées et accessibles à tout moment. Pourtant, de nombreuses plateformes existantes souffrent de limitations : les fonctionnalités de gestion des cartes bancaires (activation, blocage, modification des paramètres) restent souvent dispersées, peu intuitives et mal intégrées aux systèmes monétiques centraux.

Ces contraintes nuisent à l'expérience utilisateur, réduisent l'autonomie des clients et complexifient la gestion des opérations pour les banques. Par ailleurs, les nouvelles attentes en matière de personnalisation, intelligence artificielle (comme la classification automatique des transactions) et assistance instantanée via chatbot imposent d'importants défis technologiques et organisationnels.

Dès lors, comment concevoir et développer une plateforme e-Banking moderne et évolutive, capable d'unifier la gestion avancée des cartes bancaires, d'assurer une intégration fiable et sécurisée avec les systèmes existants comme PowerCARD, et de répondre aux standards actuels en matière de performance, de sécurité et d'expérience utilisateur ?

3. Objectifs

- Concevoir et développer une plateforme e-Banking intelligente et sécurisée permettant la gestion complète des cartes bancaires physiques et virtuelles.
- Offrir une expérience utilisateur moderne, fluide et accessible depuis le web et le mobile.
- Garantir une intégration fiable avec la solution PowerCARD via les APIs PAPI/SAPI pour assurer la synchronisation des opérations et des données en temps réel.
- Mettre en place des outils d'intelligence artificielle pour analyser et classifier automatiquement les transactions bancaires.
- Déployer un chatbot conversationnel intelligent pour accompagner les utilisateurs et renforcer leur autonomie.
- Assurer un haut niveau de sécurité et de performance, conformément aux standards bancaires internationaux.

4. Solution Proposé

Pour répondre à ces enjeux, la solution proposée consiste à développer une plateforme e-Banking moderne et sécurisée, reposant sur une architecture microservices qui facilite l'évolutivité et la maintenance des différents modules (gestion des cartes, utilisateurs, sécurité, notifications). La plateforme intègre les APIs PAPI/SAPI de PowerCARD afin de synchroniser en temps réel les données bancaires et garantir la fiabilité des opérations. Elle comprend des interfaces web et mobiles intuitives, réalisées avec React et Flutter, permettant aux utilisateurs de gérer facilement leurs cartes bancaires. Parallèlement, un système de notifications en temps réel basé sur Kafka informe les clients de toute opération effectuée. Des modèles d'intelligence artificielle sont intégrés pour classifier automatiquement les transactions et détecter les comportements inhabituels. Enfin, un chatbot conversationnel intelligent accompagne les utilisateurs en leur apportant une assistance instantanée et personnalisée. Cette combinaison de technologies vise à offrir une solution complète, performante et centrée sur l'expérience client, en phase avec les exigences actuelles du secteur bancaire.

5. Méthodologie

La méthodologie adoptée pour la réalisation de ce projet s'appuie sur une démarche **itérative et structurée**, inspirée des principes agiles afin de garantir une meilleure adaptation aux besoins et une validation progressive des livrables.

Elle repose sur les étapes suivantes :

1. Analyse des besoins et cadrage fonctionnel

- Étudier les attentes des utilisateurs et les exigences techniques.
- Identifier les contraintes d'intégration avec la solution PowerCARD.
- Définir les priorités et rédiger les spécifications fonctionnelles.

2. Conception de l'architecture technique

- Élaborer une architecture microservices modulaire et sécurisée.
- Définir les schémas d'interactions entre les différents modules (cartes, utilisateurs, sécurité, notifications).
- Choisir les technologies et outils adaptés.

3. Mise en place de l'environnement de développement

- Installer et configurer les environnements de travail.
- Mettre en place la base de données, les outils de versionnage, Kafka et les API.

4. Développement itératif des composants

- Implémenter les microservices backend avec Spring Boot.
- Développer les interfaces web et mobiles avec React et Flutter.
- Assurer l'intégration avec PowerCARD (PAPI/SAPI).

5. Déploiement des fonctionnalités d'IA et du chatbot

- Concevoir des modèles de classification des transactions.
- Développer le chatbot conversationnel.

6. Mise en place du système de notifications en temps réel

- Configurer Kafka pour diffuser les événements liés aux opérations sur les cartes.

7. Tests et validation

- Effectuer des tests unitaires et fonctionnels.
- Vérifier la sécurité et la performance globale.

8. Documentation et préparation au déploiement

- Rédiger la documentation technique et utilisateur.
- Préparer les supports de présentation du projet.

6. Planification

Ce diagramme de Gantt illustre le planning détaillé du projet de développement de la plateforme e-Banking intelligente et sécurisée, mené au sein de l'entreprise HPS. Le diagramme couvre la période de mars à septembre et est structuré autour des différentes étapes clés du projet. Chaque tâche est représentée par une couleur distincte afin de faciliter la lecture.

○ Cadrage du besoin

- **Durée :** Semaine 1 à Semaine 4 (Mars)
- **Description :** Phase initiale visant à analyser le besoin du client, les fonctionnalités attendues et les contraintes techniques et sécuritaires.

○ Architecture microservices

- **Durée :** Semaine 3 à Semaine 6 (Mars - Avril)

- **Description** : Conception de l'architecture microservices de la plateforme (user-service, card-service, ai-service, notification-service) ainsi que la définition des communications interservices via Kafka.
- **Formation Visa / Mastercard**
 - **Durée** : Semaine 7 à Semaine 8 (Avril)
 - **Description** : Session de formation approfondie sur les standards et APIs Visa/Mastercard nécessaires pour l'intégration des modules de gestion de cartes et de paiement.
- **Développement frontend (React / Flutter)**
 - **Durée** : Semaine 6 à Semaine 13(Avril - Juin)
 - **Description** : Implémentation des interfaces utilisateurs web et mobiles avec un design iOS-like moderne pour les profils client (Cardholder) et agent (Agent).
- **Développement backend (Spring / Kafka / AI)**
 - **Durée** : Semaine 7 à Semaine 17 (Avril - Juin)
 - **Description** : Développement des services backend en Java Spring Boot incluant la gestion des utilisateurs, des cartes bancaires, des transactions et des règles métiers.
- **Intégration des modules IA**
 - **Durée** : Semaine 14 à Semaine 19 (Juin - Juillet)
 - **Description** : Intégration des modules d'intelligence artificielle pour l'analyse des CV, le matching intelligent entre candidats et offres, et le scoring de compatibilité.
- **Intégration PowerCARD & APIs**
 - **Durée** : Semaine 15 à Semaine 18 (Juin - Juillet)
 - **Description** : Intégration des APIs PowerCARD avec les modules internes via le bus Kafka et mise en place des flux sécurisés de synchronisation.
- **Notifications (Kafka)**
 - **Durée** : Semaine 16 à Semaine 20 (Juin - Juillet)
 - **Description** : Implémentation des notifications temps réel (Kafka Topics), incluant les emails, SMS et alertes de sécurité en cas d'actions sensibles.
- **Tests et validation**

- **Durée** : Semaine 17 à Semaine 22 (Juin - Juillet)
- **Description** : Phase de test fonctionnel, de validation des cas limites, de test de montée en charge et de validation des règles de sécurité et de conformité.

○ **Rédaction du rapport**

- **Durée** : Semaine 19 à Semaine 23 (Juin - Juillet)
- **Description** : Rédaction du rapport de fin de projet détaillant la méthodologie, les réalisations techniques, les obstacles rencontrés ainsi que les perspectives d'évolution du projet.

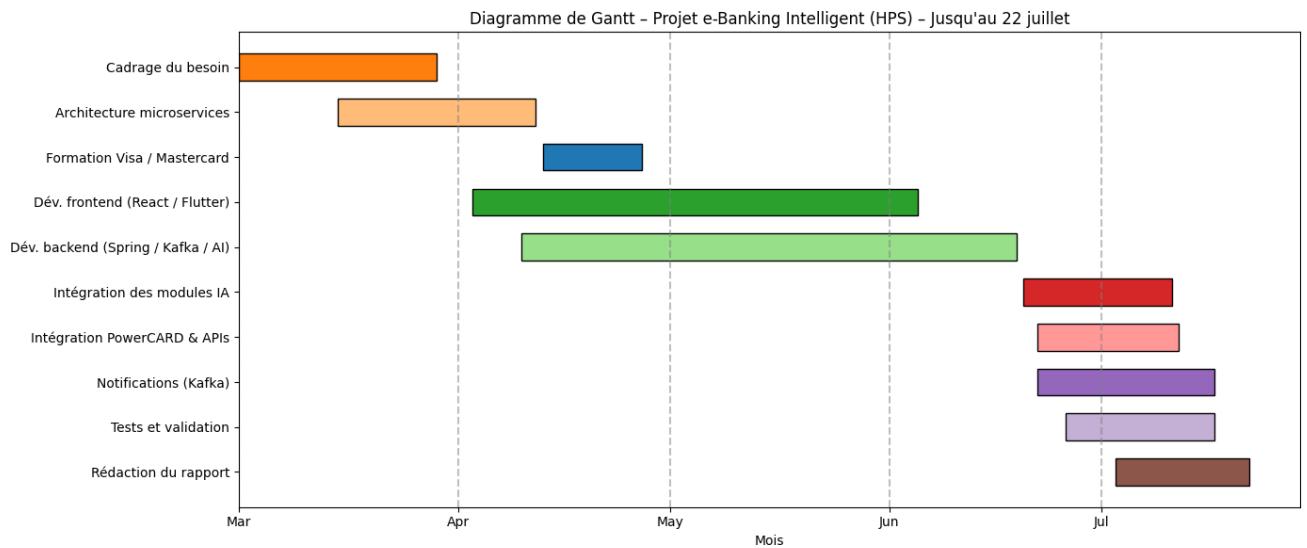


Figure 4:Diagramme de Gant

Conclusion

Ce chapitre a présenté le contexte du projet, l'environnement d'HPS, la problématique, les objectifs et la solution retenue pour développer une plateforme e-Banking moderne, sécurisée et intégrée à PowerCARD. La méthodologie agile et la planification détaillée jusqu'au 22 juillet posent les bases de la réalisation, qui sera développée dans les chapitres suivants.

Chapitre II :

Étude Préliminaire

I. Introduction

Dans cette partie du chapitre, nous présenterons d'abord le domaine de la monétique afin de mieux situer le contexte général, puis nous exposerons le contexte spécifique du projet à travers la problématique, les objectifs attendus et la conduite du projet.

II. Introduction à la monétique

1. Introduction à la monétique

La monétique désigne l'ensemble des technologies et des procédés informatiques, électroniques et télématiques permettant de gérer les paiements par carte et les différentes opérations associées. Elle couvre tout le cycle de vie des transactions, depuis la fabrication et la personnalisation des cartes jusqu'à la lecture des données et au traitement des paiements. En tant que composante essentielle du paiement électronique, la monétique englobe diverses modalités de transactions : cartes à piste magnétique, cartes à puce, paiements en ligne et solutions combinées. Ce secteur fait intervenir plusieurs acteurs majeurs tels que les établissements bancaires, les organismes émetteurs de cartes (comme Visa ou MasterCard), les prestataires de services de paiement et les entreprises spécialisées dans les infrastructures technologiques. Sur le plan technique, la monétique s'appuie sur des domaines clés tels que la conception de solutions de commerce électronique, la sécurisation des flux financiers, la gestion des réseaux de paiement et la cryptographie. D'un point de vue industriel, elle englobe également la production des équipements de paiement, notamment les terminaux, les distributeurs automatiques et les cartes elles-mêmes.

2. Le déroulement d'une transaction électronique

Une transaction électronique implique cinq acteurs principaux : le porteur de la carte (client), l'émetteur de la carte (banque du client), l'accepteur (commerçant équipé d'un terminal de paiement électronique), l'acquéreur (banque du commerçant) et l'organisme d'interchange (garantissant l'interopérabilité entre les acteurs). Le processus de transaction se déroule en trois phases : shopping (sélection des biens/services et accord sur le montant à payer), paiement (authentification et autorisation de paiement) et livraison (remise des biens/services).

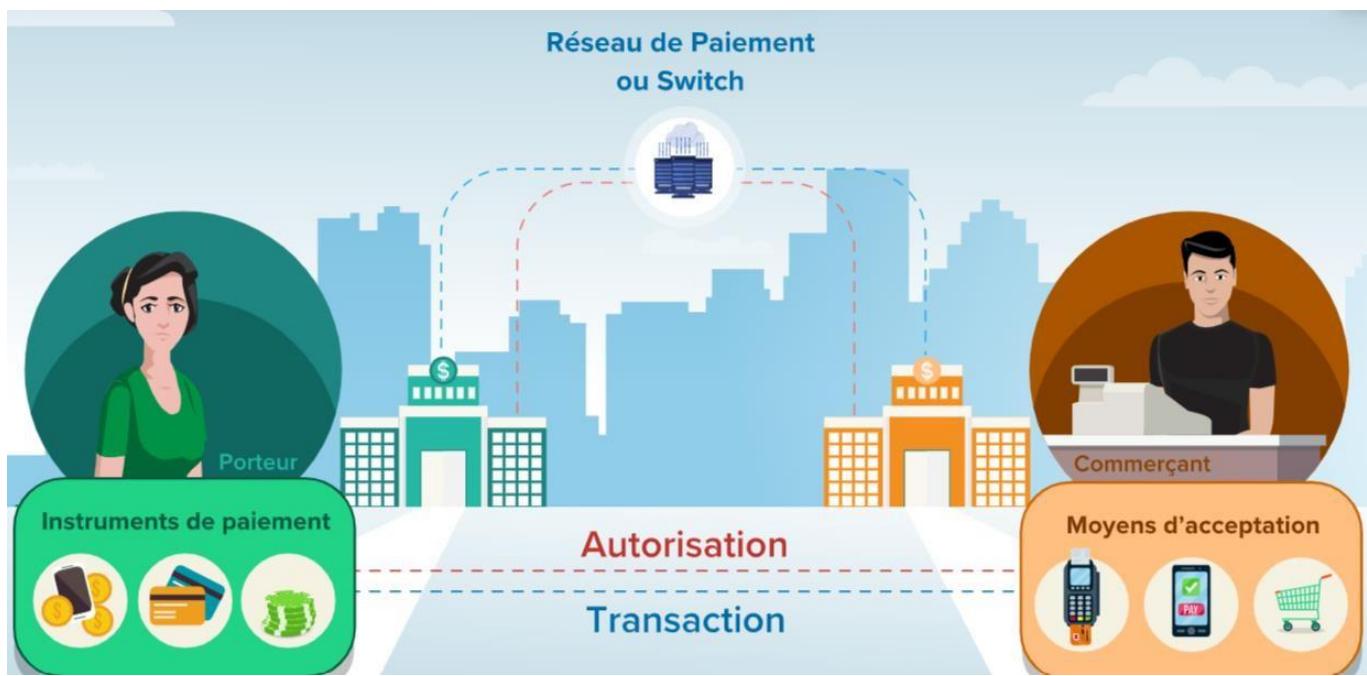


Figure 5 : Chaine de valeur de paiement

3. La solution monétique PowerCard

La solution PowerCARD, évoluée et hautement sécurisée, se compose de plusieurs modules chacun couvre une partie de la chaîne de valeur de paiement :

- **PowerCARD-eSecure** : Etape supplémentaire pour les paiements en ligne afin de lier une autorisation financière avec une authentification en ligne (ACS - 3D Secure)
- **PowerCARD-ACH** : Solution logicielle modulaire qui automatise le traitement et le règlement des opérations de paiement électronique.
- **PowerCARD-Issuer** : Services complets pour l'émission et la gestion de toutes les cartes sous tous les formats.
- **PowerCARD-xPOS** : Système de gestion des demandes d'autorisation et de gestion des transactions pour tout type de terminal de paiement électronique.
- **PowerCARD-Tokenisation** : Solution complète qui supporte l'émission, le provisionnement et le stockage de jetons de la part d'un demandeur de jeton.

- **PowerCARD-Fraud** : Solution de gestion de fraude autonome, qui contrôle en temps réel ou différé les autorisations venant d'un quelconque système
- **PowerCARD-eCommerce** : Solution multi-commerçants et multi-acquéreurs pour une gestion de paiement e-commerce.
- **PowerCARD-Switch** : Routing, stand-in et autorisation dans un environnement à haute disponibilité.
- **PowerCARD-Dashboard** : Ensemble de tableaux de bord qui fournissent une vision claire aux utilisateurs en se basant sur leurs indicateurs clés de performance.
- **PowerCARD-Acquirer** : Plateforme globale de gestion des commerçants permettant aux acquéreurs d'adapter des solutions spécifiques.
- **PowerCARD-WebPublisher** : Solution portail web avec des fonctions administratives efficaces, qui permettent de gérer et de personnaliser de nombreux portails web.
- **Retailer Open Payment Platform** : Plateforme offrant un parcours client sans faille et sécurisé, acceptant les transactions de tout canal de paiement.
- **PowerCARD-ATM** : Solution self-service globale qui permet aux institutions financières et aux détaillants une gestion et une optimisation des GAB.

La figure montre l'intervention des différentes solutions de PowerCARD dans la chaîne de paiement :

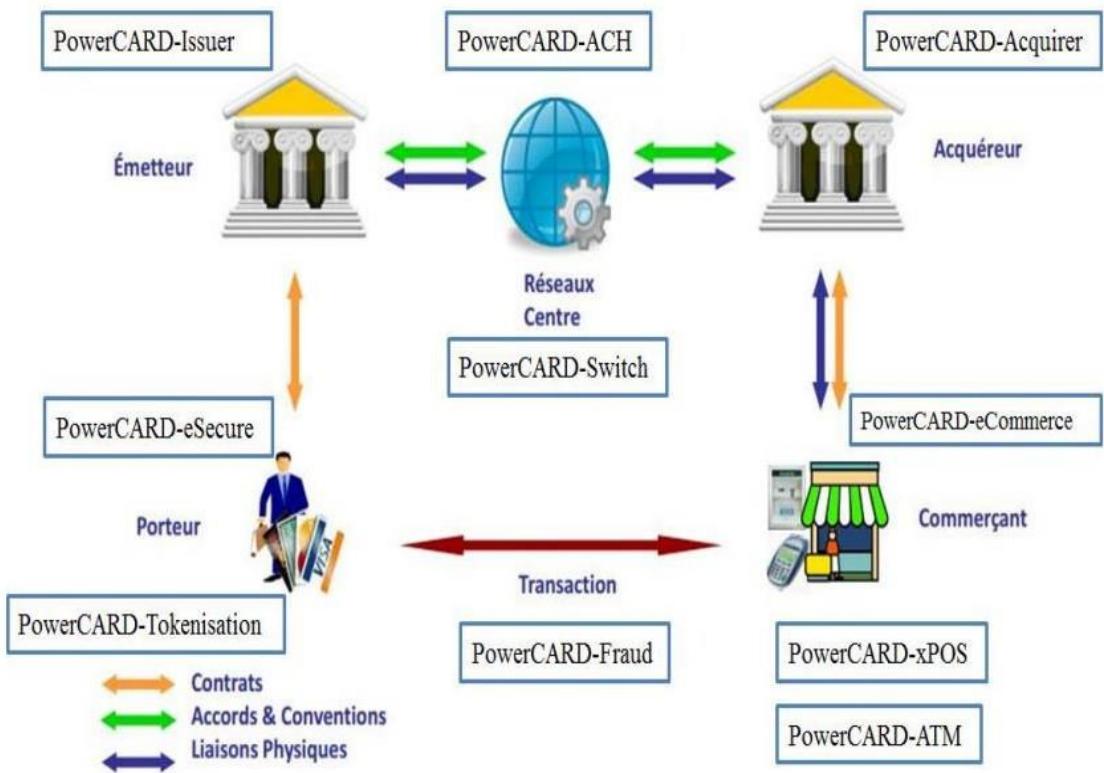


Figure 6 : L'intervention des solutions PowerCARD dans la chaîne monétique

III. Contexte du Projet

1. Contexte du Projet

Notre mission principale lors de notre stage chez HPS a été de concevoir et développer une plateforme e-Banking intelligente et sécurisée dédiée à la gestion avancée des cartes bancaires (physiques et virtuelles). Ce projet s'inscrit dans la stratégie d'innovation continue de HPS, visant à proposer des solutions modernes, flexibles et sécurisées, répondant aux nouvelles attentes des institutions financières et de leurs clients.

La plateforme repose sur une architecture microservices et s'intègre avec la solution POWERCARD, développée par HPS et adoptée par plus de 450 institutions dans le monde. L'objectif est de fournir un espace client moderne, accessible via le web et le mobile, permettant la gestion complète des cartes bancaires, incluant des fonctionnalités telles que

l’activation ou le blocage de carte, la configuration des plafonds, la consultation des opérations, ou encore la gestion des préférences utilisateur (notifications, services e-commerce, paiement sans contact...).

Parmi les innovations apportées, la plateforme exploite les APIs PowerCARD via PAPI/SAPI afin d’assurer une intégration fluide et sécurisée avec le système monétique. Elle intègre également des modèles d’intelligence artificielle pour analyser et classifier les transactions bancaires selon leur comportement, leur fréquence ou leur nature, ainsi qu’un chatbot RH intelligent permettant aux utilisateurs d’obtenir rapidement des réponses et de l’assistance sur les fonctionnalités liées à leurs cartes.

- Pour mener à bien cette mission, les étapes suivantes ont été réalisées : Analyse des besoins fonctionnels : Identification des modules à intégrer et définition des cas d’usage principaux liés à la gestion des cartes.
- Conception de l’architecture technique : Élaboration d’une architecture basée sur les microservices avec une séparation claire entre les services (cartes, utilisateurs, notifications, sécurité...).
- Développement des interfaces web et mobile : Création d’une expérience utilisateur fluide, moderne et responsive grâce à Flutter pour le mobile et React pour le web.
- Intégration avec les APIs PowerCARD (via PAPI/SAPI) : Connexion de la plateforme aux services monétiques de HPS pour assurer la cohérence et la mise à jour des données entre les systèmes.
- Implémentation de l’IA pour l’analyse des transactions : Utilisation de modèles intelligents permettant la détection de comportements anormaux, la segmentation des opérations, et l’optimisation du suivi utilisateur.
- Mise en place d’un chatbot RH assisté par IA : Déploiement d’un assistant conversationnel pour guider l’utilisateur dans la gestion de sa carte et lui fournir des réponses rapides aux questions fréquentes.
- Mise en place d’un système de notifications intelligentes : Génération et affichage des notifications en temps réel après chaque opération sur carte via Kafka.
- Tests et validation : Vérification de la stabilité, de la performance et de la sécurité de la plateforme dans différents scénarios utilisateurs.

Ce projet nous a permis d’allier nos compétences en architecture logicielle, développement frontend/backend, et intégration de solutions monétiques dans un cadre professionnel et structuré.

2. Problématique

Aujourd’hui, dans un contexte de transformation digitale rapide et de montée en puissance des usages mobiles, les clients des établissements financiers attendent de pouvoir gérer leurs cartes bancaires de manière autonome, instantanée et sécurisée, où qu’ils se trouvent. Pourtant, malgré les progrès réalisés, de nombreuses plateformes e-Banking restent limitées par une expérience utilisateur fragmentée : les opérations comme l’activation, le blocage, la modification des plafonds ou la gestion des options (sans contact, e-commerce, etc.) sont souvent réparties sur plusieurs canaux ou interfaces, ce qui nuit à la clarté et à la réactivité du service.

Par ailleurs, la montée des cybermenaces et le renforcement des exigences réglementaires en matière de protection des données imposent aux banques de repenser leurs architectures techniques pour garantir un haut niveau de sécurité, tout en assurant une disponibilité constante et une évolutivité adaptée aux nouveaux besoins.

Les clients attendent également des fonctionnalités innovantes, telles que la classification intelligente des transactions pour mieux comprendre et maîtriser leurs dépenses, ainsi qu’un accompagnement personnalisé via des chatbots conversationnels capables de répondre rapidement aux questions et de guider les démarches en toute autonomie.

Dans ce contexte exigeant, il devient essentiel de concevoir une plateforme e-Banking moderne, unifiée et évolutive, qui centralise toutes les opérations liées à la gestion avancée des cartes bancaires et qui assure une intégration fiable et sécurisée avec des systèmes monétiques existants tels que PowerCARD. La question centrale est donc : comment développer une solution digitale capable de concilier simplicité d’utilisation, sécurité renforcée, intelligence fonctionnelle et interopérabilité, tout en offrant une expérience utilisateur fluide et conforme aux standards du secteur bancaire ?

3. Objectifs

- Concevoir une plateforme e-Banking complète permettant aux clients de gérer de manière centralisée l’ensemble de leurs cartes bancaires (physiques et virtuelles) via des interfaces web et mobiles intuitives et modernes.
- Mettre en place une architecture microservices modulaire et sécurisée, garantissant la scalabilité, la maintenabilité et l’indépendance des différents modules (gestion des cartes, gestion des utilisateurs, sécurité, notifications en temps réel, etc.).

- Assurer l'intégration transparente avec la solution PowerCARD de HPS grâce aux APIs PAPI/SAPI, afin de synchroniser les données et les opérations de manière fiable et en temps réel.
- Renforcer la sécurité des transactions et des échanges de données sensibles, en appliquant les standards bancaires et les bonnes pratiques (OAuth2, JWT, protocoles HTTPS) afin de protéger l'information des clients et de se conformer aux réglementations.
- Développer des fonctionnalités d'intelligence artificielle dédiées à l'analyse et à la classification automatique des transactions bancaires, permettant de détecter les comportements inhabituels et d'améliorer la lisibilité des dépenses pour l'utilisateur final.
- Déployer un chatbot conversationnel intelligent, capable d'accompagner les clients dans leurs démarches courantes, de répondre à leurs questions fréquentes et de les assister de manière autonome et interactive.
- Concevoir un système de notifications en temps réel, basé sur l'utilisation de Kafka, pour informer instantanément les clients de toute opération ou modification effectuée sur leurs cartes.
- Garantir une expérience utilisateur homogène, fluide et responsive, adaptée aux différents terminaux (ordinateurs, smartphones, tablettes), avec une attention particulière portée à l'ergonomie, la simplicité de navigation et la rapidité d'exécution.
- Valider la solution par une phase de tests rigoureux, incluant des scénarios de performance, de sécurité et de non-régression, afin d'assurer la fiabilité et la qualité des livrables avant leur mise en production.

IV. Besoins fonctionnels et non fonctionnels

1. Besoins fonctionnels

Les besoins fonctionnels décrivent **les fonctionnalités concrètes que la plateforme doit offrir aux utilisateurs finaux (clients, agents) et aux administrateurs.** Ils se déclinent en plusieurs catégories :

a) Gestion des cartes bancaires

- Création, activation et désactivation des cartes bancaires physiques et virtuelles.
- Blocage temporaire ou définitif d'une carte.
- Modification des plafonds de paiement et de retrait.
- Gestion des préférences d'utilisation :
 - e-commerce activé/désactivé
 - paiement sans contact
 - autorisation internationale
- Consultation des informations détaillées d'une carte :
 - historique des transactions
 - statut de la carte
 - limites en vigueur
 - date d'expiration.

b) Gestion des utilisateurs

- Authentification sécurisée des utilisateurs avec gestion des rôles (cardholder / agent).
- Création, modification et suppression des comptes utilisateurs.
- Gestion des données personnelles (informations de contact, préférences de communication).

c) Notifications en temps réel

- Génération et diffusion de notifications après chaque action sur une carte (blocage, activation, paiement...).
- Support multicanal : notifications push, emails, SMS.

d) Intelligence artificielle

- Classification automatique des transactions bancaires par catégorie.
- Détection des comportements inhabituels et alertes proactives.

e) Chatbot conversationnel

- Assistance aux utilisateurs via un chatbot intelligent intégré (questions fréquentes, guidage pas à pas).
- Disponibilité du chatbot sur le web et sur mobile.

f) Intégration avec PowerCARD

- Synchronisation bidirectionnelle des données via les APIs PAPI/SAPI.
- Gestion des autorisations financières, de l'authentification forte (3D Secure) et des opérations en temps réel.

g) Interfaces utilisateurs

- Interface web ergonomique et moderne (React).
- Application mobile responsive et fluide (Flutter).

2. Besoins non fonctionnels

Les besoins non fonctionnels concernent **la qualité, la sécurité et la performance globale** du système. Ils couvrent :

a) Sécurité

- Chiffrement des données sensibles en transit (HTTPS) et au repos.
- Authentification via OAuth2 et génération de JWT.
- Conformité aux standards bancaires internationaux (PCI DSS, RGPD).
- Gestion des autorisations et des droits d'accès selon les rôles.

b) Performance

- Temps de réponse moyen inférieur à 500 ms sur les opérations principales.
- Capacité de montée en charge horizontale grâce à l'architecture microservices.

c) Disponibilité

- Disponibilité cible supérieure à 99,5 %.
- Tolérance aux pannes et redondance des services critiques (Kafka, PowerCARD).

d) Scalabilité

- Capacité d'ajouter de nouveaux modules (ex. : nouvelles fonctionnalités IA, intégration de wallets numériques) sans impact sur les modules existants.

e) Maintenabilité

- Code source structuré et documenté.
- Automatisation des tests et déploiements via des pipelines CI/CD.
- Journalisation centralisée des événements et suivi des erreurs.

f) Expérience utilisateur

- Interfaces homogènes, intuitives et accessibles sur tous les terminaux (desktop, mobile).
- Temps de chargement optimisé.
- Conception conforme aux standards UX/UI modernes (iOS-like design).

V. Développement de la plateforme e-Banking

1. Introduction

Le développement de la plateforme e-Banking intelligente et sécurisée a nécessité la mise en œuvre d'une méthodologie rigoureuse, tenant compte à la fois de la complexité technique du projet, des spécificités métier du domaine bancaire et des exigences de sécurité, de performance et de disponibilité. Cette méthodologie s'est appuyée sur une approche itérative et incrémentale afin de garantir un avancement progressif et un contrôle régulier de la qualité à chaque phase.

2. Objectifs du développement

Les principaux objectifs poursuivis durant le développement ont été :

- **Concevoir une interface fluide et intuitive pour la gestion des cartes bancaires (activation, blocage, limites, préférences, etc.).**
- **Assurer l'intégration sécurisée avec la solution PowerCARD via les APIs PAPI/SAPI.**
- **Mettre en œuvre une architecture microservices modulaire, évolutive et scalable.**
- **Intégrer les tests dans des cycles courts de développement via des processus CI/CD.**

3. Bénéfices apportés par la solution

La solution développée offre plusieurs avantages :

- **Centralisation** de toutes les opérations carte dans un seul espace sécurisé
- **Gain d'autonomie** pour les clients grâce à des services personnalisés
- **Performance et évolutivité** avec une architecture microservices scalable
- **Accessibilité multi-plateforme** (mobile et web) avec design cohérent
- **Réactivité temps réel** grâce à l'intégration de Kafka pour les notifications
- **Analyse intelligente** des opérations grâce au module de classification IA
- **Assistance automatisée** via le chatbot RH intégré à l'application

4. Défis Techniques

Le projet a nécessité de surmonter plusieurs défis techniques majeurs :

- Intégration complexe avec les APIs PAPI/SAPI.
- Gestion des dépendances et synchronisation des microservices.
- Traitement sécurisé de données sensibles.
- Adaptation des UI/UX aux spécificités des plateformes Flutter et React.
- Préparation et entraînement de jeux de données qualifiés pour l'IA.
- Gestion des cas limites métier dans les communications asynchrones Kafka.

5. Critères de priorisation fonctionnelle

- **Criticité fonctionnelle** de la fonctionnalité pour l'utilisateur final
- **Fréquence d'utilisation** estimée
- **Complexité de mise en œuvre technique**
- **Lien direct avec des composants sensibles (PowerCARD, IA, sécurité)**
- **Impact sur l'expérience utilisateur**
- **Temps de développement estimé**
- **Capacité à être réutilisé ou étendu dans le futur**

6. Bibliothèques et outils d'intelligence artificielle utilisés

Dans le cadre de la mise en place du module de classification intelligente des transactions bancaires, plusieurs bibliothèques et technologies avancées ont été utilisées afin de garantir un traitement performant, précis et évolutif des données.

Les principales bibliothèques et outils intégrés dans le pipeline d'intelligence artificielle sont :

- **Pandas** et **NumPy** : pour le traitement, la manipulation et l'analyse statistique des données transactionnelles (montants, dates, catégories, etc.).
- **Scikit-learn** : pour l'encodage des labels, la normalisation des variables, la création des ensembles d'entraînement/test, ainsi que pour l'utilisation de techniques de machine learning classiques.
- **LightGBM** : un algorithme de gradient boosting efficace et rapide, utilisé pour classer les transactions bancaires selon leur nature (paiement, retrait, virement, etc.) en tenant compte de leurs caractéristiques numériques.
- **Transformers (Hugging Face)** et **PyTorch** : utilisés pour encoder les libellés textuels des transactions grâce au modèle pré-entraîné **DistilBERT**, permettant d'extraire des représentations sémantiques fines des descriptions textuelles.
- **TQDM** : pour le suivi de la progression lors du prétraitement et de l'inférence sur de grands ensembles de données.

En parallèle de ces outils orientés classification, la plateforme intègre également des technologies avancées de recherche et génération augmentée par les données :

- **RAG (Retrieval-Augmented Generation)** : une technique hybride combinant recherche documentaire et génération de texte, utilisée pour enrichir les réponses du chatbot RH intégré à la plateforme.
- **PGVector (PostgreSQL Vector Store)** : un moteur de stockage vectoriel permettant d'indexer et de retrouver efficacement des embeddings générés par les modèles de langage, facilitant la recherche sémantique et les réponses personnalisées en temps réel.

L'ensemble de ces bibliothèques a permis d'élaborer un module IA robuste, capable de classifier les opérations, détecter les comportements inhabituels, et améliorer l'expérience utilisateur grâce à une assistance intelligente et contextuelle.

7. Méthodologie Générale du projet

7.1. Choix de la méthodologie

Dans le cadre de la réalisation de ce projet de développement d'une plateforme e-Banking intelligente et sécurisée, il était essentiel d'adopter une méthodologie de gestion de projet adaptée à la complexité technique, à la diversité des modules fonctionnels et à l'intégration progressive des différents composants.

Après une analyse des différentes approches existantes (cycle en cascade, cycle en V, méthodes agiles), le choix s'est porté sur un **cycle de vie itératif et incrémental, sous forme spiralée**.

Ce modèle permet d'introduire progressivement les fonctionnalités tout en intégrant des validations régulières et des ajustements continus au fil de l'avancement du projet.

7.2. Pourquoi le modèle spiralé ?

Le choix du modèle spiralé a été motivé par plusieurs facteurs propres aux caractéristiques du projet :

- **Complexité fonctionnelle élevée** : gestion des cartes bancaires physiques et virtuelles, sécurité, gestion des transactions, gestion des utilisateurs, intégration IA, etc.
- **Nécessité d'intégration progressive des microservices** : architecture basée sur des services indépendants (user-service, card-service, ai-service, notification-service...).
- **Gestion des risques maîtrisée** : le modèle spiralé permet une évaluation régulière des risques à chaque itération.
- **Flexibilité d'adaptation** : les spécifications fonctionnelles ont pu évoluer en fonction des tests, des validations et des retours intermédiaires.
- **Validation continue** : chaque boucle du spiral permet de livrer des versions intermédiaires, testées et validées avant de passer à l'incrément suivant.

7.3. Schéma du cycle de vie adopté

Le schéma ci-dessous illustre le modèle spiralé appliqué tout au long du projet :

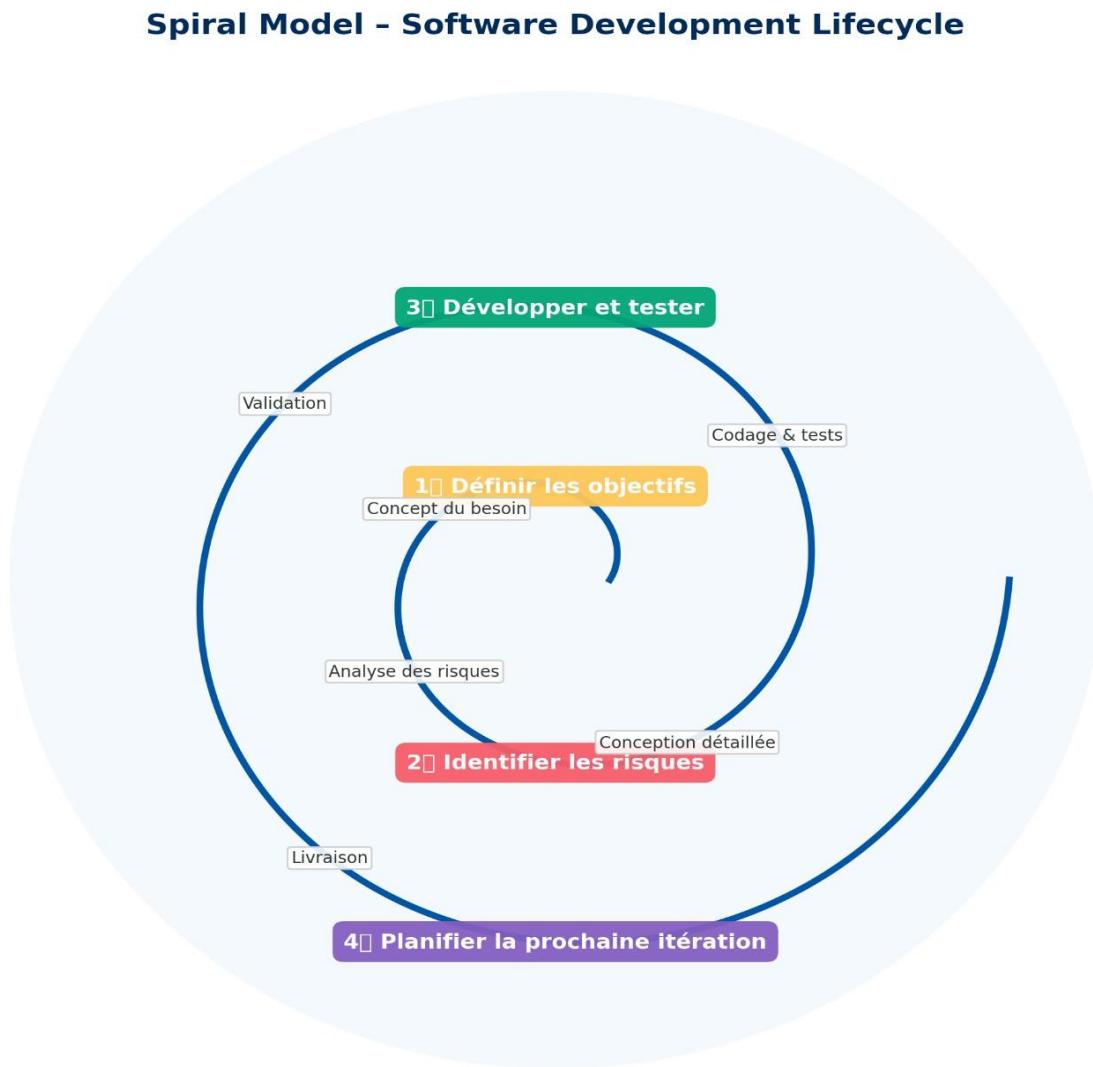


Figure 7 : Cycle de vie modèle spiralé

Conclusion

Ce chapitre a présenté le contexte du projet, la monétique, la solution PowerCARD, ainsi que les objectifs, enjeux et défis techniques. La méthodologie spiralée adoptée a permis une intégration progressive et une validation continue. Ces bases structurent le développement de la plateforme, qui sera détaillé dans les chapitres suivants.

Chapitre III :

Etude Conceptuelle



Figure 8 : Logo Uml

Le langage UML (Unified Modeling Language) est un langage de modélisation graphique standardisé permettant de représenter visuellement les différents aspects d'un système informatique. Il facilite la compréhension, la conception et la communication entre les différentes parties prenantes d'un projet logiciel en modélisant à la fois les structures statiques (classes, objets, composants) et les comportements dynamiques (interactions, flux de données, scénarios). Grâce à ses multiples diagrammes, UML offre une vision complète et normalisée du système, depuis l'analyse des besoins jusqu'à l'implémentation technique.

I. UML et son rôle dans la modélisation et la conception

En informatique, différentes approches de modélisation permettent de concevoir et structurer des solutions adaptées aux problématiques à résoudre. Parmi les principaux paradigmes de programmation, on retrouve les langages impératifs, fonctionnels, déclaratifs et orientés objet (OO).

Le paradigme orienté objet repose sur la représentation du système à travers des objets qui modélisent des entités du monde réel ou abstrait. Chaque objet possède ses propres caractéristiques et comportements, et interagit avec d'autres objets pour réaliser les fonctionnalités globales du système. Ces objets peuvent représenter, par exemple, des utilisateurs, des interfaces, des transactions ou des services métier.

1. Diagrammes de modélisation UML

1.1. Diagramme de Cas d'utilisation

Le diagramme de cas d'utilisation est un outil de modélisation permettant de représenter les interactions entre les différents acteurs et le système. Il met en évidence les fonctionnalités attendues du système du point de vue de l'utilisateur, en identifiant les services que celui-ci pourra utiliser. Ce type de diagramme est couramment utilisé lors des phases d'analyse et de conception afin de définir clairement les besoins fonctionnels du système.

a. Diagramme de Cas d'utilisation Cardholder

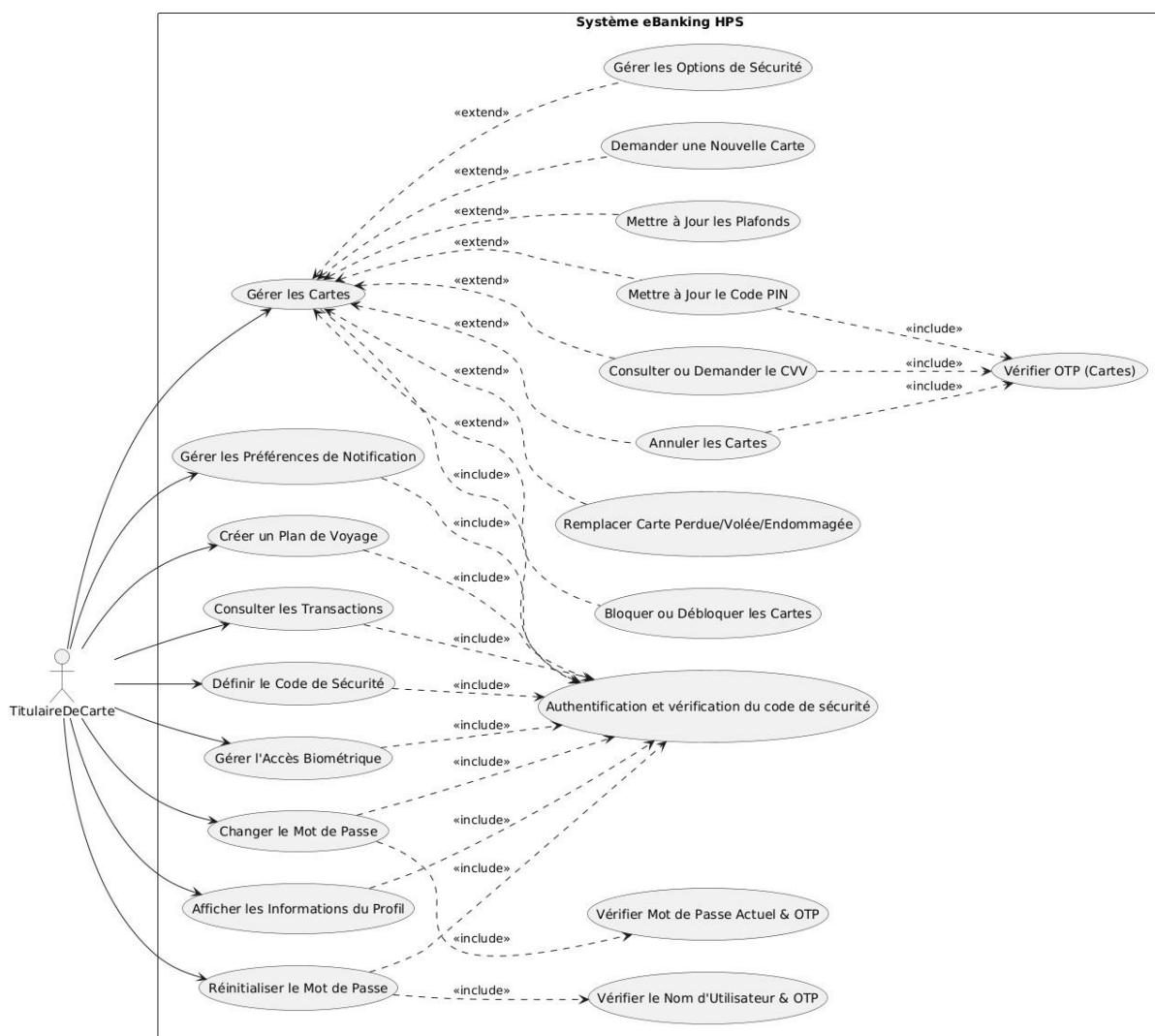


Figure 9 : Diagramme de cas d'utilisation Cardholder

Ce diagramme est un **diagramme de cas d'utilisation UML** pour le système eBanking développé chez HPS. Il illustre ce qu'un **titulaire de carte** peut faire sur la plateforme. Voici une explication synthétique :

Le **TitulaireDeCarte** (acteur principal) peut :

- **Gérer ses cartes**, ce qui inclut plusieurs sous-actions :
 - Gérer les options de sécurité
 - Demander une nouvelle carte
 - Mettre à jour les plafonds
 - Mettre à jour le code PIN
 - Consulter ou demander le CVV (avec vérification OTP)
 - Annuler une carte
 - Remplacer une carte perdue, volée ou endommagée
 - Bloquer ou débloquer la carte (nécessite une authentification)
- **Gérer ses préférences de notification**
- **Créer un plan de voyage**
- **Consulter ses transactions**
- **Définir un code de sécurité**
- **Gérer l'accès biométrique**
- **Changer le mot de passe** (nécessite vérification du mot de passe actuel et OTP)
- **Réinitialiser le mot de passe** (avec vérification du nom d'utilisateur et OTP)
- **Afficher les informations du profil**

Toutes les opérations sensibles (modification de carte, CVV, mot de passe...) incluent un mécanisme d'**authentification et de vérification** du code de sécurité ou OTP.

b. Diagramme de Cas d'utilisation Agent

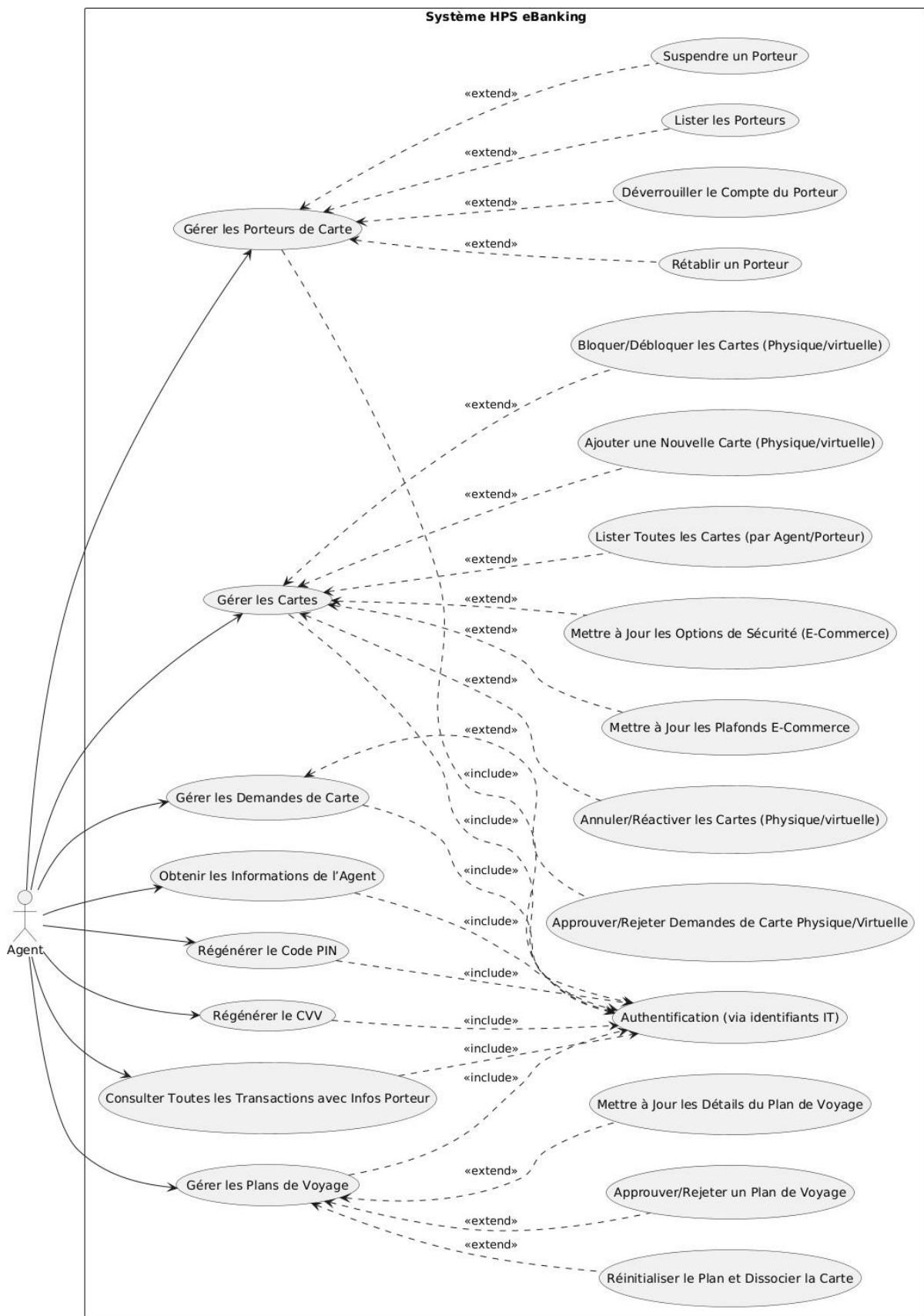


Figure 10 : Diagramme de cas d'utilisation Agent

Ce diagramme de cas d'utilisation montre les fonctionnalités offertes à un **agent** dans le système eBanking HPS.

L'**agent** peut :

- **Gérer les porteurs de carte**, ce qui inclut :
 - Suspendre, rétablir ou déverrouiller un porteur
 - Lister les porteurs
- **Gérer les cartes**, avec des sous-actions comme :
 - Ajouter, bloquer/débloquer, ou annuler/réactiver une carte (physique ou virtuelle)
 - Régénérer le code PIN ou le CVV
 - Mettre à jour les options de sécurité et les plafonds e-commerce
 - Lister toutes les cartes par agent ou porteur
- **Gérer les demandes de carte**, en approuvant ou rejetant les demandes
- **Gérer les plans de voyage**, incluant :
 - Approuver/rejeter un plan
 - Mettre à jour ses détails
 - Réinitialiser le plan et dissocier la carte
- **Consulter toutes les transactions**, avec les informations du porteur
- **Accéder à ses propres informations** (agent)

Toutes ces actions sensibles nécessitent une **authentification via identifiants IT**.

1.2. Diagramme de class

Un diagramme de classes UML est un diagramme structurel qui représente les entités (classes) d'un système, leurs attributs, leurs méthodes (facultativement) ainsi que les relations (associations, héritages, compositions) entre elles. Il permet de modéliser la structure statique du système et de comprendre comment les données sont organisées et interconnectées.

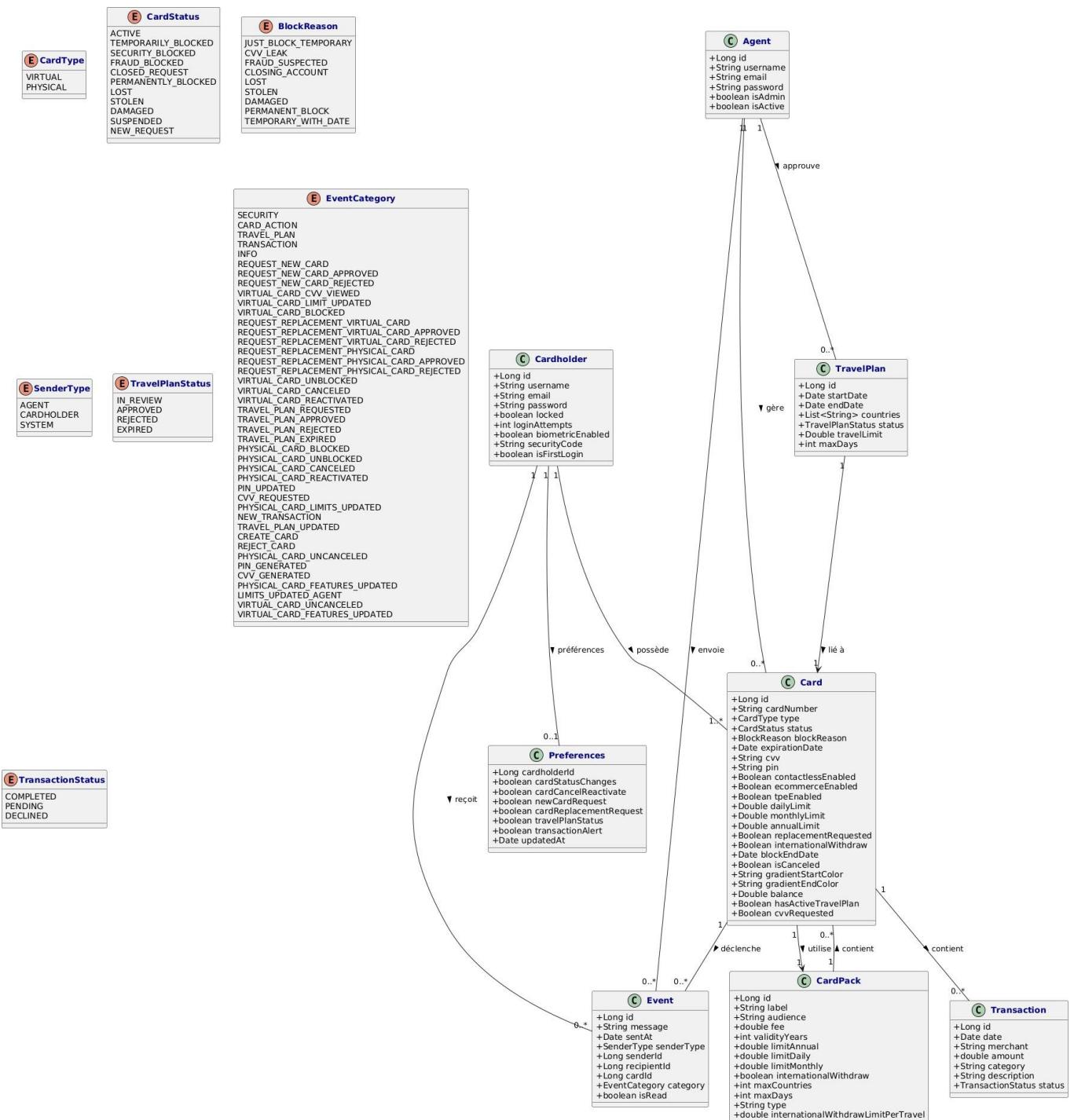


Figure 11 : Diagramme de classe

Ce diagramme représente les principales entités d'une plateforme e-Banking et leurs relations.

- Un **cardholder** (utilisateur) peut posséder plusieurs **cartes**, configurer ses **préférences**, créer des **plans de voyage**, et recevoir des **événements** liés à ses actions.
- Une **carte** est liée à un cardholder, peut appartenir à un **pack de cartes**, être associée à un **plan de voyage**, et dispose d'un **type**, d'un **statut**, et d'une **raison de blocage**.
- Un **agent** gère des cartes, approuve des plans de voyage, et génère des événements dans le système.
- Un **événement** enregistre une action (ex. : blocage, création), avec un type, une source (agent, système ou cardholder) et une date.
- Un **plan de voyage** est lié à une carte, contient une période, des pays, des limites, et un statut.
- Une **transaction** représente une opération effectuée avec une carte (montant, commerçant, devise, statut).
- Les **types énumérés** comme CardStatus, BlockReason, CardType, EventCategory, SenderType, etc., permettent de classifier les statuts et types de chaque entité.

1.3. Diagramme de séquence – Scénarios d'authentification et de sécurité du titulaire de carte

Un diagramme de séquence UML est un type de diagramme d'interaction qui modélise le comportement dynamique d'un système. Il décrit, de manière chronologique, l'échange de messages entre les différents acteurs, objets ou composants d'un système dans le cadre d'un scénario fonctionnel précis (ex. : authentification, soumission de formulaire, traitement de transaction, etc.).

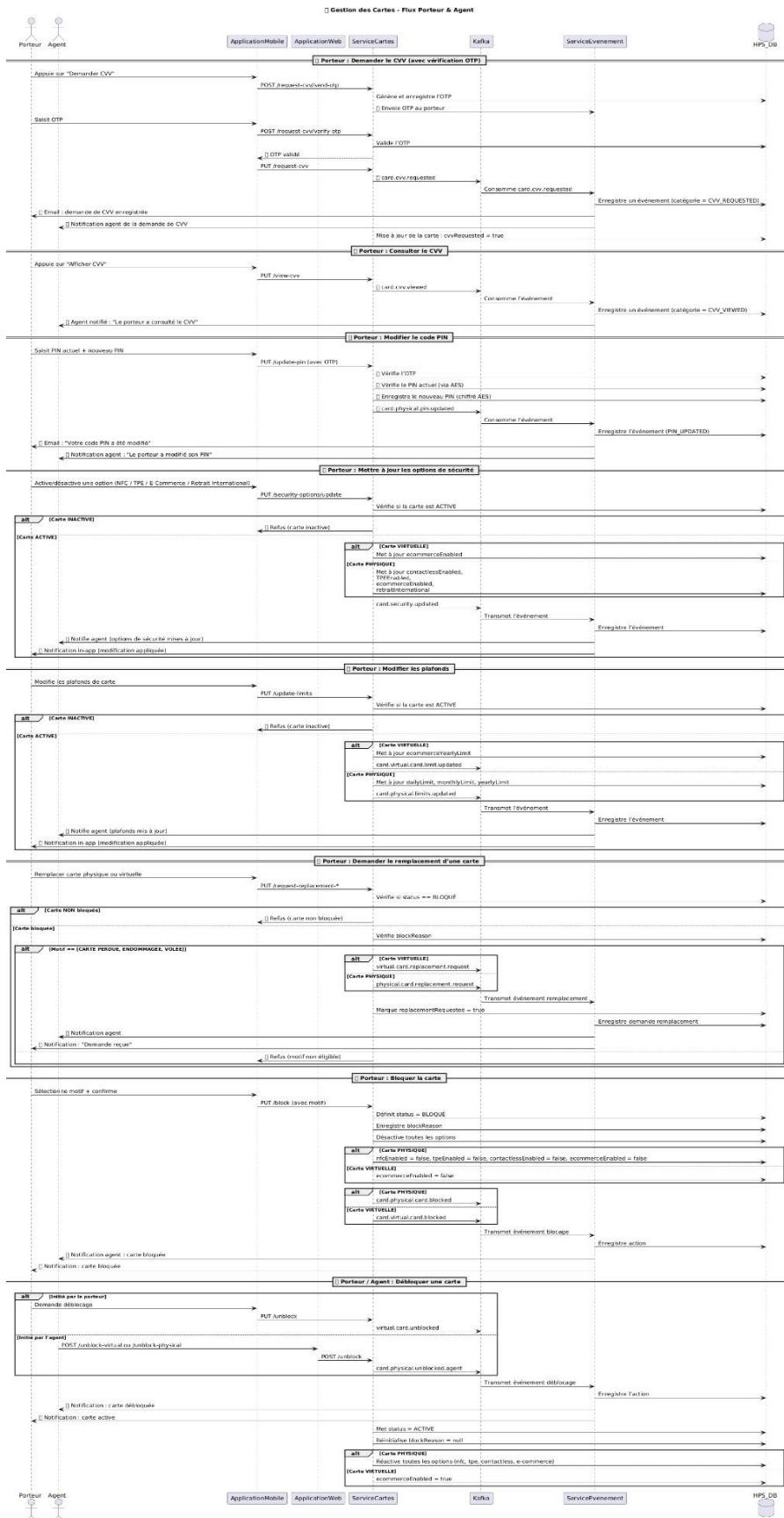


Figure 12:diagramme de séquence d'authentification et de sécurité

Ce diagramme de séquence illustre les principaux cas d'usage liés à la gestion des cartes bancaires, mettant en interaction le titulaire de carte et l'agent à travers leurs interfaces respectives.

- Le titulaire peut initier une demande de consultation du CVV. Cette opération nécessite une vérification préalable par OTP, envoyé automatiquement par le système. Une fois l'OTP validé, la demande est enregistrée, traçable, et une notification est envoyée à l'agent.
- Il peut ensuite consulter le CVV. Cette action déclenche un événement spécifique afin de garantir la traçabilité de l'accès aux données sensibles.
- Le titulaire a également la possibilité de modifier son code PIN. Cela implique la validation du code actuel, chiffré avec AES, ainsi que la vérification OTP. Une fois validée, le nouveau code PIN est stocké de manière sécurisée.
- Il peut activer ou désactiver certaines options de sécurité telles que le paiement sans contact, les paiements en TPE, l'e-commerce ou encore les retraits à l'étranger, selon le type de carte (physique ou virtuelle) et son statut (active ou non).
- Il est aussi possible de modifier les plafonds de paiement (quotidiens, mensuels ou annuels), à condition que la carte soit active. Chaque mise à jour est enregistrée, transmise via Kafka et notifiée aux parties concernées.
- Lorsqu'une carte est bloquée suite à une perte, un vol ou une détérioration, le titulaire peut en demander le remplacement. Le système vérifie que le motif est éligible, puis envoie l'événement au service de notification.
- Le titulaire peut également décider de bloquer lui-même une carte en sélectionnant un motif. Cette action désactive automatiquement les options sensibles et est historisée comme un événement de sécurité.
- Enfin, le déblocage d'une carte peut être effectué soit par le titulaire, soit par un agent, en fonction du scénario. La carte est alors réactivée, les options restaurées, et le changement est notifié et enregistré.

Toutes ces opérations entraînent systématiquement une mise à jour en base de données, une publication d'événement via Kafka et l'envoi d'une notification au titulaire et/ou à l'agent. Ce fonctionnement garantit une gestion fluide, centralisée, conforme et hautement traçable de l'ensemble des opérations liées aux cartes bancaires.

1.4. Diagramme de séquence – Gestion des cartes : interactions Titulaire & Agent

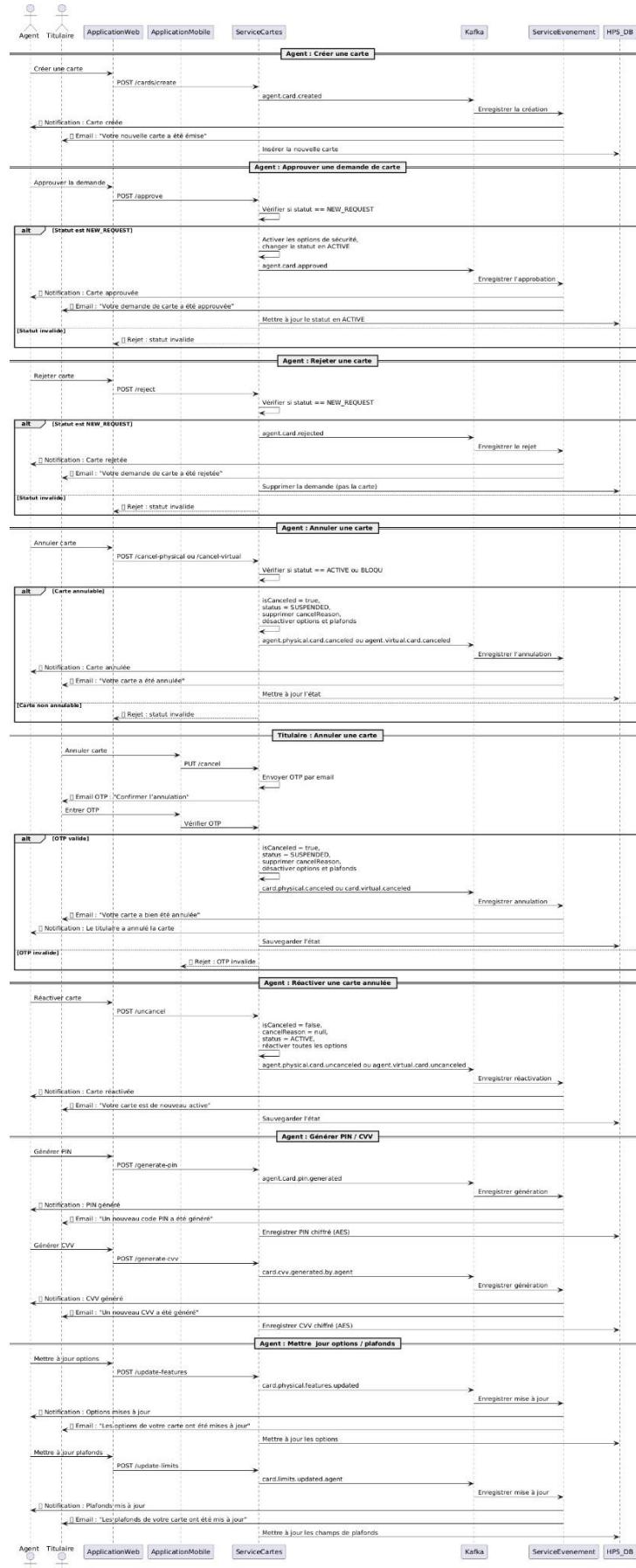


Figure 13: diagramme de séquence Gestion des cartes

Ce diagramme de séquence détaille les principaux cas d'usage liés à la gestion administrative des cartes bancaires, réalisés par l'agent via l'interface web, ainsi que certaines actions possibles par le titulaire via l'application mobile.

- L'agent peut créer une nouvelle carte bancaire pour un client. Une fois la requête traitée, la carte est enregistrée dans la base de données, et une notification est envoyée au titulaire l'informant de l'émission de sa carte.
- Il peut également approuver une demande de carte émise par un titulaire. Cette validation déclenche l'activation de la carte, la mise à jour de son statut, et l'envoi d'une notification à toutes les parties concernées.
- Si nécessaire, l'agent peut rejeter une demande de carte encore en attente. L'événement est enregistré, le titulaire est informé du rejet, et la demande est supprimée du système.
- L'agent a aussi la possibilité d'annuler une carte, qu'elle soit physique ou virtuelle, à condition que son statut soit valide (active ou bloquée).
- L'annulation implique la suspension de la carte, la désactivation de ses options, et l'enregistrement complet de l'opération.
- Le titulaire, de son côté, peut également annuler sa propre carte via l'application mobile. Cette opération nécessite une vérification OTP envoyée par email. En cas de saisie correcte, la carte est suspendue et l'agent est notifié.
- L'agent a la possibilité de réactiver une carte annulée. Cela réinitialise son statut, réactive ses fonctionnalités, et enregistre l'événement dans le système.
- En cas de besoin, l'agent peut générer un nouveau code PIN ou un nouveau CVV. Ces éléments sont chiffrés (AES), enregistrés, et communiqués au titulaire de façon sécurisée.
- L'agent peut également modifier les options de la carte (comme le NFC, le TPE, l'e-commerce, etc.) ou ajuster les plafonds (quotidien, mensuel, annuel). Chaque changement est propagé via Kafka, enregistré en base et notifié à l'utilisateur.

Toutes ces actions sont encadrées par une logique rigoureuse : chaque opération déclenche une mise à jour persistante, un événement Kafka pour traçabilité, et une notification systématique au titulaire et/ou à l'agent. Cette approche garantit une gestion fiable, conforme et entièrement auditable du cycle de vie des cartes bancaires.

1.5. Diagramme de séquence Travel Plan :Titulaire & Agent

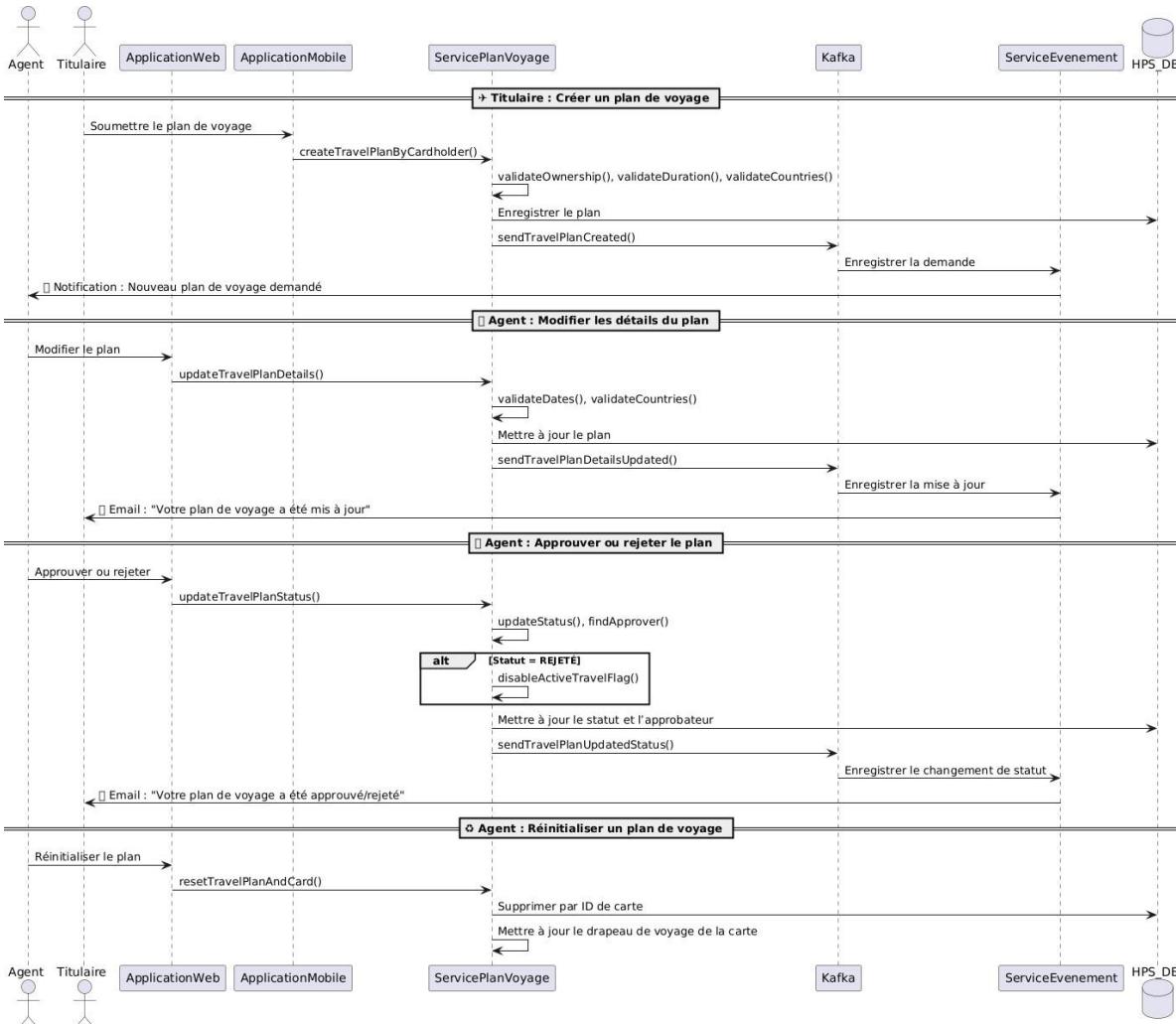


Figure 14 : Diagramme de séquence Travel Plan :Titulaire & Agent

Ce diagramme de séquence illustre les principaux cas d'usage liés à la gestion des plans de voyage, impliquant le titulaire de carte et l'agent, chacun interagissant à travers son interface dédiée : application mobile pour le titulaire, interface web pour l'agent.

- Le titulaire peut soumettre un plan de voyage via l'application mobile. Avant l'enregistrement, le système vérifie que la carte lui appartient, que la durée du voyage est valide et que les pays sélectionnés respectent les règles définies. Une fois validé, le plan est sauvegardé, un événement est envoyé via Kafka, et l'agent est notifié de la demande.

- L'agent peut ensuite modifier les détails du plan, notamment les dates ou les destinations. Le service effectue à nouveau des vérifications avant de mettre à jour les données en base.
- Un événement Kafka est alors émis, et une notification est envoyée au titulaire pour l'informer des modifications.
- Une fois le plan revu, l'agent peut approuver ou rejeter la demande. L'action entraîne la mise à jour du statut du plan ainsi que l'enregistrement de l'agent responsable. En cas de rejet, le système désactive également l'indicateur de plan actif associé à la carte. Une notification est alors transmise au titulaire.
- L'agent peut également réinitialiser un plan de voyage à tout moment. Cela entraîne la suppression du plan existant à partir de l'identifiant de la carte concernée, et la mise à jour de l'indicateur associé dans la base de données pour signaler qu'aucun plan n'est actif.

Chaque action suit un processus structuré : validation métier, persistance en base de données, publication d'un événement via Kafka, et notification aux utilisateurs concernés. Ce fonctionnement assure une gestion fluide, traçable et conforme des plans de voyage dans le cadre de l'utilisation des cartes bancaires.

1.6. Diagramme de séquence Transaction :Titulaire & Agent

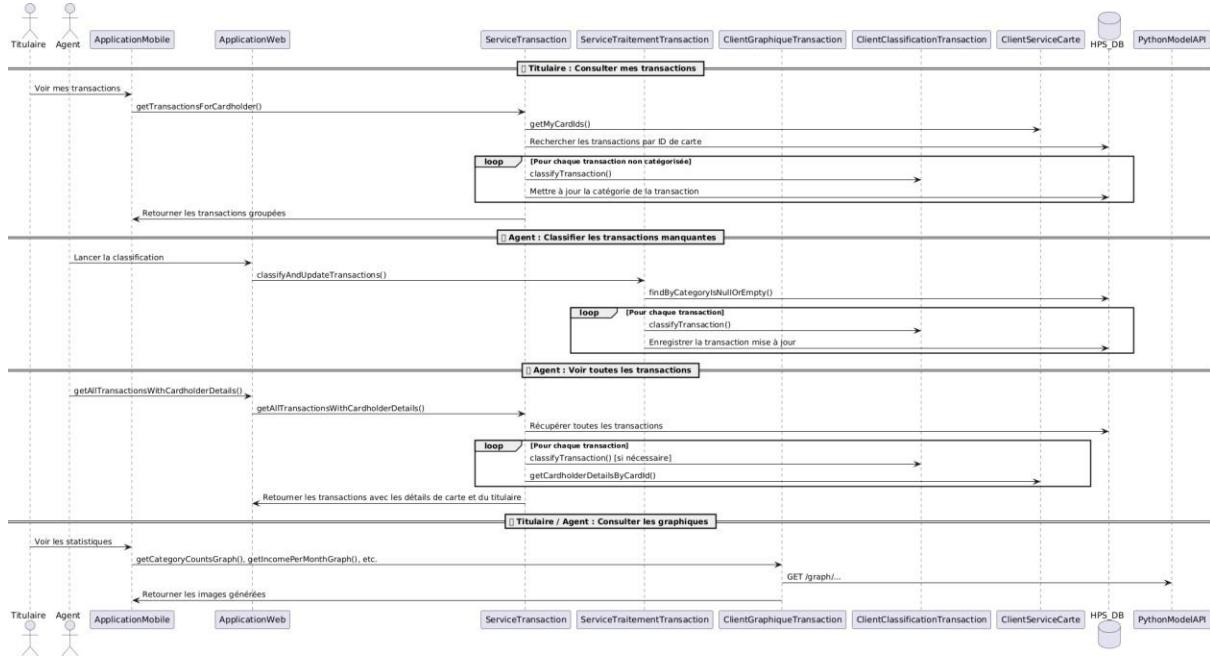


Figure 15: Diagramme de séquence Transaction :Titulaire & Agent

Ce diagramme de séquence décrit les principales opérations liées à la gestion des transactions bancaires, réalisées par le titulaire via l'application mobile et par l'agent via l'interface web.

- Le titulaire peut consulter ses transactions. Le système récupère les cartes associées, extrait les transactions, puis classe automatiquement celles qui ne le sont pas encore. Les résultats sont regroupés et affichés.
- L'agent peut lancer une classification des transactions non catégorisées. Chaque transaction est traitée puis mise à jour en base.
- Il peut également visualiser toutes les transactions avec les détails des titulaires. Si nécessaire, les transactions sont classées à la volée, et les informations liées aux cartes sont jointes.
- Enfin, le titulaire et l'agent peuvent consulter des graphiques statistiques générés via une API Python, permettant une visualisation claire des dépenses et revenus.

Ces actions garantissent une gestion automatisée, enrichie et traçable des transactions dans l'écosystème bancaire.

1.7. Diagramme de séquence Chatbot

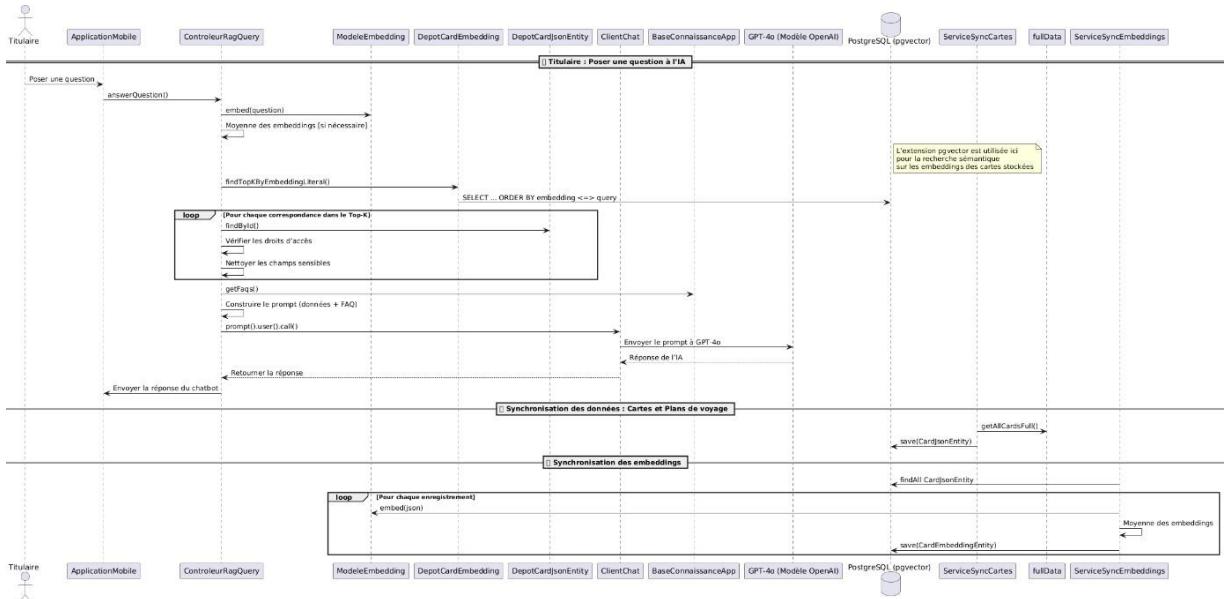


Figure 16: Diagramme de séquence Chatbot

Ce diagramme décrit le fonctionnement du **chatbot intelligent basé sur RAG**, utilisé par le titulaire via l'application mobile.

- Lorsqu'une question est posée, le système génère un embedding vectoriel, puis effectue une recherche sémantique dans PostgreSQL (via pgvector) pour trouver les cartes pertinentes.
- Les résultats sont filtrés (accès + anonymisation), enrichis avec des FAQs, puis envoyés sous forme de prompt à GPT-4o. La réponse de l'IA est ensuite retournée au titulaire.
- Deux processus de synchronisation maintiennent la base à jour :
 - L'un pour les cartes et plans de voyage,
 - L'autre pour générer et stocker les embeddings des cartes.

Ce flux assure des réponses personnalisées, pertinentes et sécurisées.

Conclusion :

La phase de conception a permis de modéliser l'architecture globale de la plateforme e-Banking en identifiant les principales entités, cas d'utilisation et interactions entre les composants. À travers les diagrammes UML (cas d'utilisation, classes, séquence), nous avons posé les bases d'un système structuré, évolutif et aligné avec les besoins métier. Cette étape a été essentielle pour garantir la cohérence entre l'interface utilisateur, les services backend et les flux d'événements au sein de l'architecture microservices.

II. Outils et Technologie

Ce chapitre présente les outils et les technologies mobilisés pour le développement de la plateforme e-Banking intelligente et sécurisée. Il décrit les environnements de développement, les frameworks, les langages, ainsi que les solutions utilisées pour assurer la sécurité, l'intégration avec PowerCARD, l'interface utilisateur web et mobile, et l'implémentation des fonctionnalités d'intelligence artificielle..

1. Technologie et Outils

Spring Boot est un sous-projet du Framework Spring qui offre une approche orientée opinion pour la construction d'applications autonomes, prêtes pour la production et basées sur Spring. Il simplifie la configuration et la mise en œuvre des applications Spring, ce qui permet aux développeurs de se concentrer sur le développement de fonctionnalités métier plutôt que sur la configuration de l'infrastructure



Spring Data est un module de Spring qui facilite l'accès et la manipulation des données en fournissant une couche d'abstraction pour travailler avec des bases de données relationnelles et non relationnelles. Il prend en charge plusieurs technologies de stockage de données, notamment JDBC, JPA, MongoDB, Redis, etc.



Spring Security Module de sécurité de Spring utilisé pour gérer l'authentification (connexion) et l'autorisation (droits d'accès) des utilisateurs, que ce soit via des rôles ou des tokens. Il permet de sécuriser les APIs exposées par les services.



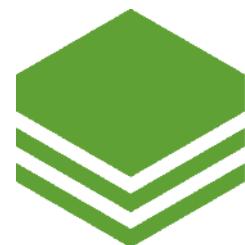
Spring AI Extension du framework Spring permettant d'intégrer facilement des modèles d'intelligence artificielle (NLP, embeddings, classification, etc.) dans les microservices. Elle a été utilisée pour automatiser certaines tâches comme l'analyse de CV ou la catégorisation des transactions.



Spring Cloud Ensemble de bibliothèques facilitant le développement de systèmes distribués. Il permet la gestion centralisée de la configuration, le routage dynamique, la tolérance aux pannes et la découverte de services, essentiels dans une architecture microservices.



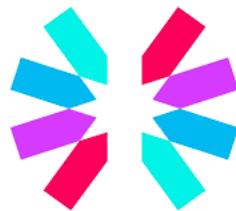
Spring Batch est un framework robuste conçu pour exécuter des traitements de données en masse, de manière planifiée ou déclenchée à la demande. Il est particulièrement adapté aux processus répétitifs comme l'envoi groupé de notifications ou la génération de rapports. Dans notre plateforme e-Banking, il a été utilisé pour automatiser l'envoi d'OTP et la gestion de traitements RH périodiques, tout en assurant fiabilité, traçabilité et reprise sur erreur en cas d'échec.



OAuth2 est un protocole d'autorisation standardisé qui permet à une application d'accéder à des ressources protégées sans exposer les identifiants de l'utilisateur. Il repose sur un système de jetons d'accès temporaires, garantissant sécurité et contrôle d'accès. Dans notre plateforme e-Banking, OAuth2 a été utilisé pour sécuriser les appels aux APIs, en autorisant uniquement les utilisateurs authentifiés à accéder aux services correspondant à leur rôle.



JWT (JSON Web Token) est un jeton signé qui contient les informations d'authentification d'un utilisateur, échangé de manière sécurisée entre le client et le serveur. Il permet une gestion d'authentification sans session (stateless), ce qui allège la charge côté serveur. Dans notre plateforme e-Banking, un JWT est généré à chaque connexion réussie, puis utilisé pour authentifier automatiquement toutes les requêtes ultérieures de l'utilisateur, en fonction de son rôle et de ses droits.



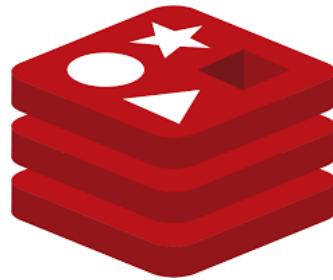
Kafka est une plateforme de messagerie distribuée, conçue pour gérer des flux de données en temps réel selon un modèle orienté événements. Elle permet une communication asynchrone, fiable et hautement scalable entre microservices grâce à un système de topics (sujets) auxquels les services peuvent publier ou s'abonner. Dans notre plateforme e-Banking, Kafka est utilisé pour transmettre des événements métier tels que la création de compte, l'envoi d'alertes de sécurité ou la validation de transactions, assurant ainsi une orchestration fluide et découpée des services.



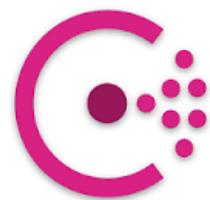
Zookeeper est un composant essentiel de Kafka, chargé de coordonner et de gérer la configuration des brokers d'un cluster. Il assure le suivi des partitions, l'élection des leaders et la tolérance aux pannes, garantissant la cohérence et la haute disponibilité. Dans notre infrastructure e-Banking, il contribue à la stabilité et à la synchronisation des services Kafka.



Redis Cache est une base de données clé-valeur en mémoire, réputée pour sa rapidité et ses faibles temps de latence. Elle est idéale pour stocker temporairement des données fréquemment consultées, permettant ainsi de soulager la base de données principale et d'améliorer les performances globales du système. Dans notre plateforme e-Banking, Redis est utilisé pour stocker les OTP lors de la vérification d'identité, les sessions utilisateurs actives ainsi que certaines préférences applicatives mises en cache.



Consul est un système de découverte de services et de gestion de configuration distribué. Il permet aux microservices de s'enregistrer automatiquement et de découvrir dynamiquement les autres services disponibles, sans configuration manuelle d'adresses ou de ports. Dans notre architecture e-Banking, Consul facilite le déploiement, le scaling et l'interconnexion des microservices, tout en assurant une meilleure résilience et adaptabilité du système.



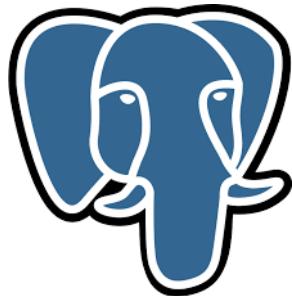
API Gateway est un composant central qui agit comme point d'entrée unique entre les clients (applications web ou mobiles) et les différents microservices du backend. Il prend en charge des fonctions essentielles telles que la sécurité, le routage intelligent des requêtes, la limitation de débit (rate limiting), la journalisation et parfois l'agrégation des réponses provenant de plusieurs services. Dans notre projet e-Banking, l'API Gateway joue un rôle clé en filtrant les accès selon les rôles (agent, cardholder, admin) et en simplifiant considérablement les appels effectués depuis le frontend.



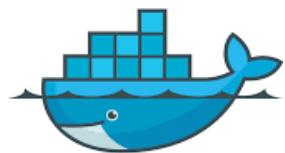
REST API est une architecture web basée sur les méthodes HTTP (GET, POST, PUT, DELETE) permettant une communication simple et uniforme entre applications et services backend. Dans notre plateforme e-Banking, chaque microservice expose des APIs REST sécurisées pour gérer les utilisateurs, les cartes et les plans de voyage.



PostgreSQL représente la couche d'accès à la base de données relationnelle utilisée dans notre architecture, basée sur le système PostgreSQL. Elle permet d'exécuter des requêtes métier complexes, de gérer des relations entre tables via des jointures, et d'assurer la cohérence des données grâce à la gestion transactionnelle. Dans notre plateforme e-Banking, cette couche est essentielle pour stocker et interroger de manière fiable les données liées aux utilisateurs, aux cartes bancaires, aux transactions financières ainsi qu'aux événements métier générés par les différents services.

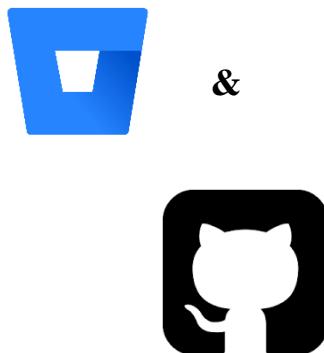


Docker est une plateforme de virtualisation légère qui permet d'encapsuler une application, ses dépendances, sa configuration et son environnement d'exécution dans une unité appelée *conteneur*. Cette approche garantit que s'exécute de manière identique quel que soit l'environnement hôte (développement, test ou production). Dans notre projet e-Banking, Docker a été utilisé pour isoler chaque microservice, faciliter les tests, accélérer les déploiements, et assurer la cohérence entre les différentes phases du cycle de vie logiciel.



Bitbucket et **GitHub** sont des plateformes d'hébergement de code source basées sur Git, utilisées pour le versionnement, la collaboration en équipe et la gestion du cycle de vie du développement logiciel. Elles permettent de suivre les modifications, de gérer les branches, de valider les contributions via pull requests, et d'intégrer des workflows CI/CD.

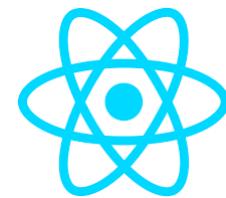
Dans notre projet, elles ont facilité la coordination entre les développeurs, la gestion des versions des microservices, et le déploiement automatisé via des pipelines



LightGBM (Light Gradient Boosting Machine) est un algorithme de machine learning basé sur les arbres de décision, optimisé pour les performances et la rapidité d'exécution, même sur de grands volumes de données. Il est particulièrement adapté aux tâches de classification, de régression et de ranking. Dans notre projet, LightGBM a été utilisé pour la classification intelligente des transactions bancaires, en s'appuyant sur des données historiques enrichies, afin d'optimiser la détection de comportements ou d'anomalies et améliorer l'analyse financière automatisée.



React JS est une bibliothèque JavaScript développée par Meta qui permet de créer des interfaces utilisateurs dynamiques et modulaires grâce à une architecture orientée composants. Dans notre plateforme e-Banking, React a été utilisé pour concevoir le portail web des agents et des cardholders, offrant une interface fluide et adaptée à chaque profil.



Redux JS est une bibliothèque de gestion d'état centralisé utilisée avec React. Elle permet de stocker l'état global dans un store et de partager facilement les données entre composants. Dans notre projet, Redux a servi à gérer l'état des utilisateurs, les réponses des APIs et le suivi en temps réel des actions sur les cartes.



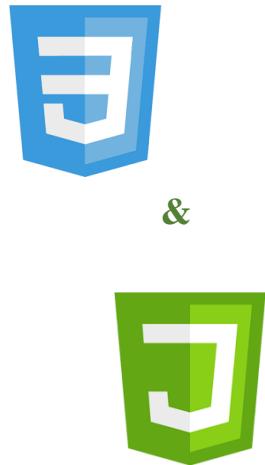
TypeScript est un surensemble de JavaScript avec typage statique, développé par Microsoft. Il aide à détecter les erreurs à la compilation et à renforcer la robustesse du code. Dans la plateforme e-Banking, il a permis de fiabiliser le frontend, notamment pour les appels API, la gestion des états et l'intégration avec Redux.



Tailwind CSS est un framework CSS utilitaire qui propose une large palette de classes prédéfinies pour styliser rapidement des composants. Il permet de construire des interfaces modernes, responsives et cohérentes sans écrire de feuilles de style personnalisées. Dans notre projet, il a été utilisé pour concevoir des vues épurées, mobiles-first, tout en garantissant un design aligné avec les standards actuels (iOS-like notamment).



CSS (Cascading Style Sheets) et **JavaScript** sont deux technologies fondamentales du développement frontend. CSS est utilisé pour définir le style visuel des pages web : couleurs, typographies, mise en page, animations et adaptation responsive. JavaScript, quant à lui, permet de rendre ces pages interactives et dynamiques en manipulant les éléments HTML, en répondant aux actions de l'utilisateur, et en orchestrant les comportements côté client. Dans notre application e-Banking, CSS complète Tailwind pour des ajustements précis de style, tandis que JavaScript constitue le moteur des interactions utilisateur, assurant une navigation fluide et une expérience moderne.



Maven et **Gradle** sont des outils de gestion de projet et d'automatisation des builds dans l'écosystème Java. **Maven** utilise un fichier *pom.xml* pour gérer les dépendances, compiler le code et produire les artefacts de manière standardisée. **Gradle**, plus moderne, s'appuie sur des scripts Groovy ou Kotlin et offre des builds plus rapides grâce à la compilation incrémentale. Dans notre projet e-Banking, **Maven** a été préféré pour sa simplicité et sa compatibilité directe avec Spring Boot, tandis que **Gradle** reste une alternative puissante pour des besoins de configuration avancés.



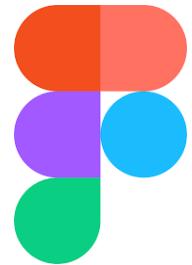
Insomnia est un outil intuitif pour tester les APIs REST et GraphQL. Il permet d'envoyer des requêtes HTTP, de gérer les en-têtes (JWT) et d'analyser les réponses. Dans notre projet e-Banking, il a servi à tester les endpoints, valider l'authentification OAuth2 et simuler des scénarios complexes.



MongoDB est une base de données NoSQL orientée documents, qui stocke les données sous forme de collections de documents JSON. Elle offre une grande flexibilité de schéma et une scalabilité horizontale. Dans notre projet e-Banking, MongoDB a été utilisée pour enregistrer certaines données non structurées et assurer des requêtes rapides sur de gros volumes d'informations.

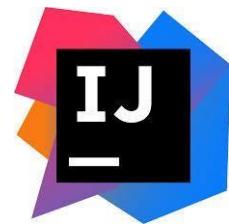


Figma est un outil de design collaboratif en ligne qui permet de créer des maquettes d'interfaces web et mobiles. Il facilite le prototypage, le partage et la coédition en réel entre les équipes. Dans notre projet e-Banking, Figma a été utilisé pour concevoir les écrans et définir les parcours utilisateurs avant le développement.



2. Environnement de développement

IntelliJ IDEA est un environnement de développement intégré (IDE) performant, utilisé dans ce projet pour le développement backend en Spring Boot. Il a facilité la gestion des dépendances avec Maven, l'intégration de Kafka et PostgreSQL, ainsi que le débogage. Grâce à son auto-complétion intelligente et son support natif des outils Spring, il a permis un développement rapide et structuré des microservices.



WebStorm est un IDE moderne dédié au développement web, utilisé ici pour créer l'interface React/TypeScript du portail e-Banking. Il offre un excellent support pour JSX, Redux, Tailwind CSS, ainsi qu'un débogueur intégré, ce qui a permis une construction fluide, maintenable et rapide du frontend. Son système d'analyse de code a permis de détecter rapidement les erreurs potentielles. WebStorm a également facilité l'intégration avec Git pour suivre les évolutions du code.



Android Studio est l'IDE officiel pour Android, utilisé dans ce projet pour tester et déployer l'application mobile Flutter. Grâce à ses simulateurs intégrés et à son support du SDK Flutter, il a permis de valider rapidement l'ergonomie et la compatibilité mobile de l'application.

Il a été essentiel pour tester l'app sur plusieurs résolutions et versions Android. L'intégration des outils de performance a permis d'optimiser le comportement de l'app mobile.



Jupyter Notebook est un environnement interactif utilisé pour analyser, visualiser et prototyper les modèles IA du projet. Il a permis de traiter les données transactionnelles, tester les algorithmes et documenter les résultats de manière claire et dynamique.

Son format hybride (code + visualisation + explication) a facilité la compréhension des tests.

Il a été un support précieux pour itérer rapidement sur les choix de modèles et de paramètres.



III. Architecture technique de la plateforme

1. Introduction

Cette section décrit l'architecture technique retenue pour la réalisation de la plateforme e-Banking. L'objectif est de présenter la structuration globale du système, l'organisation des différents composants logiciels et leur mode d'interaction. L'architecture a été conçue pour garantir la modularité, la scalabilité, la sécurité et la facilité de maintenance, tout en permettant une intégration transparente avec les services externes, notamment la solution PowerCARD. Elle définit également les choix techniques majeurs concernant les technologies de développement, les protocoles de communication et les mécanismes de traitement des données.

2. Architecture proposée

2.1. Architecture Générale

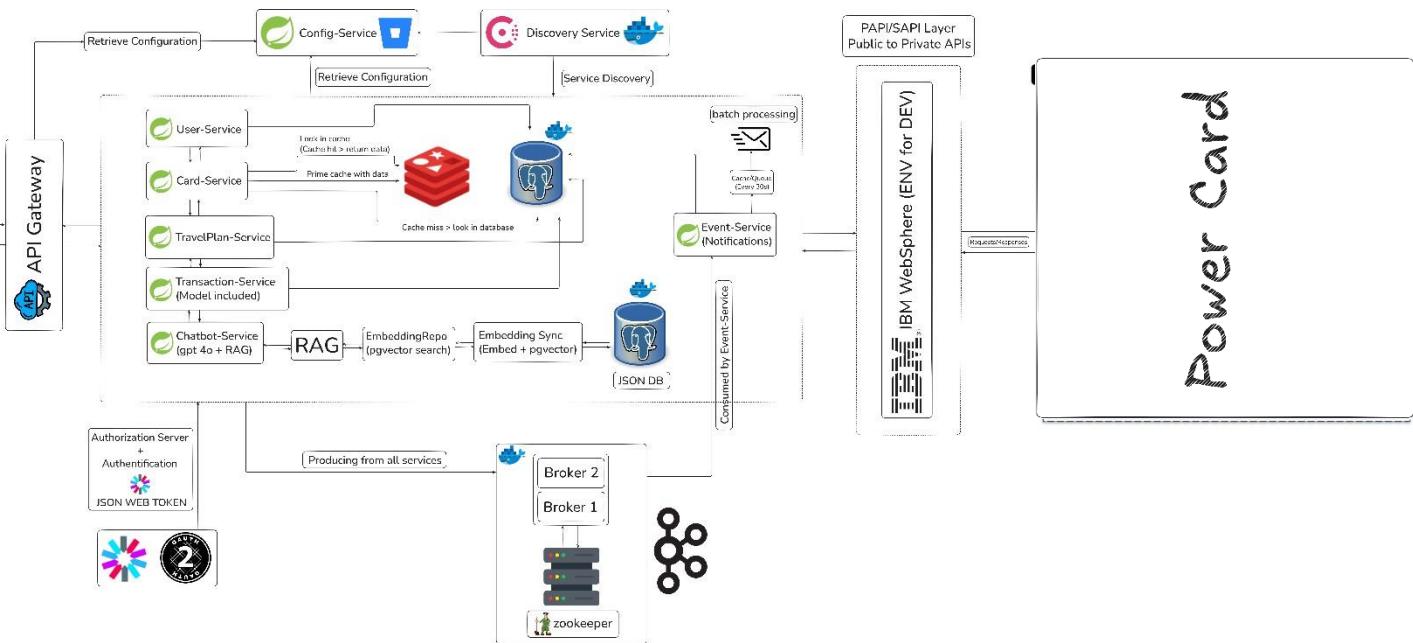


Figure 17 :Architecture Générale

Cette architecture repose sur une approche microservices intégrant l'intelligence artificielle (GPT-4o + RAG), une API Gateway sécurisée, un système d'événements Kafka, un moteur de notifications batchées, et une intégration avec la plateforme bancaire PowerCARD via IBM WebSphere.

1. Accès Utilisateur

Les utilisateurs accèdent au système via une application web (React.js) ou mobile (Flutter). Toutes les requêtes passent par une API Gateway, responsable de la sécurité, de l'authentification (OAuth2 + JWT) et du routage vers les microservices adéquats.

2. Microservices Métier (Spring Boot)

Le backend repose sur une architecture modulaire, avec des microservices indépendants :

- User-Service : gestion des utilisateurs (clients et agents)
- Card-Service : gestion des cartes bancaires (blocage, CVV, limites...)
- TravelPlan-Service : gestion des plans de voyage liés aux cartes
- Transaction-Service : enregistrement et classification intelligente des transactions à l'aide d'un modèle ML
- Chatbot-Service : assistant conversationnel intelligent basé sur GPT-4o et RAG

3. Intelligence Artificielle

Le Chatbot-Service combine :

- GPT-4o pour générer des réponses naturelles en langage humain
- RAG (Retrieval-Augmented Generation) avec pgvector (PostgreSQL vectoriel) pour rechercher dynamiquement les documents internes les plus pertinents (ex. FAQ, CGU, notices...)

4. Infrastructure Technique

- L'écosystème est soutenu par plusieurs composants techniques robustes :
- JWT + OAuth2 : sécurisation des accès
- Redis : cache haute performance pour réduire les accès à la base
- PostgreSQL : base de données principale des microservices
- Kafka : bus d'événements pour la communication asynchrone entre services
- Config-Service & Discovery-Service : gestion centralisée des configurations et découverte des services

Event-Service :

- Consomme les événements Kafka (ex. user.account.created)
- Gère l'envoi d'e-mails de notification
- Fonctionne par batch processing toutes les 30 secondes

5. Intégration avec PowerCARD

Le système communique avec le système bancaire PowerCARD V3.5 via IBM WebSphere à travers une couche PAPI/SAPI permettant :

- L'accès aux services métiers existants, batchs bancaires et procédures stockées
- Une intégration fluide avec l'environnement Oracle (PL/SQL, DataGuard, GoldenGate)

2.2. Architecture du Model de Transactions

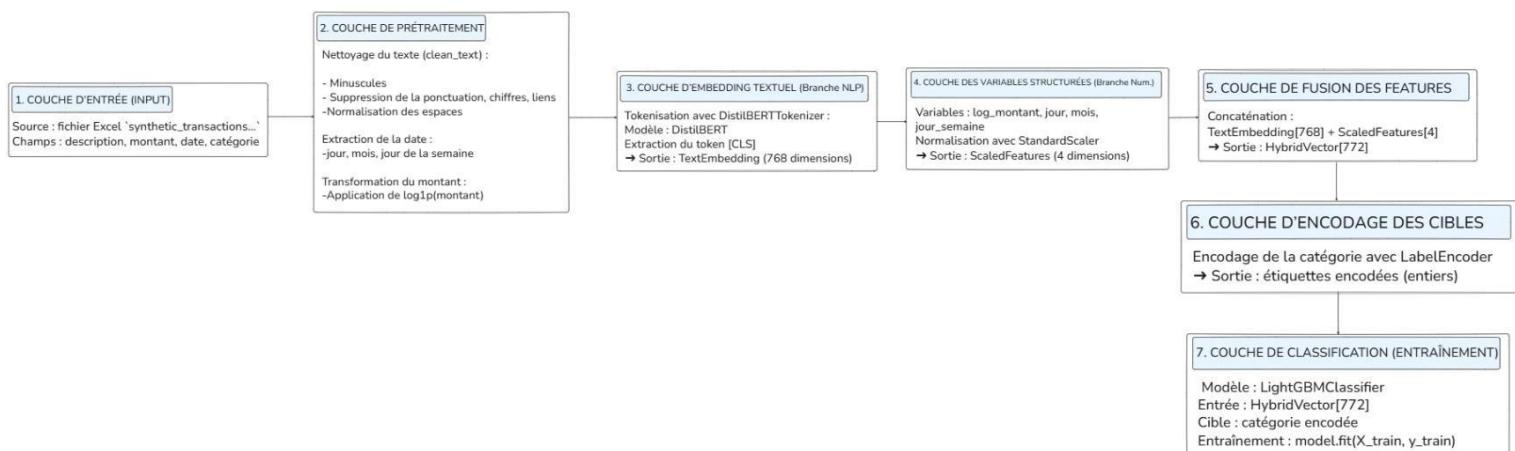


Figure 18: Architecture du Model de Transactions

Dans le cadre du projet, nous avons mis en place un pipeline intelligent permettant de classer automatiquement les transactions bancaires selon leur nature (alimentation, transport, loisirs, etc.). Ce modèle repose sur l'intelligence artificielle et combine à la fois l'analyse du texte (description de la transaction) et des informations chiffrées (montant, date...).

Voici les différentes étapes du processus d'apprentissage du modèle :

1. Chargement des Données

- Les données sont extraites d'un fichier Excel contenant :
- La description de la transaction
- Le montant
- La date
- Et la catégorie (utilisée comme référence pour l'apprentissage)

2. Préparation des Données

- Avant de construire le modèle, les données sont nettoyées :
- Le texte est simplifié (mise en minuscules, suppression des caractères inutiles...)
- La date est découpée (jour, mois, jour de la semaine)
- Le montant est transformé pour améliorer la stabilité des calculs

3. Analyse du Texte

Le cœur du modèle repose sur une technologie d'intelligence artificielle capable de comprendre le sens de la description d'une transaction. Pour cela, un modèle pré-entraîné appelé DistilBERT est utilisé pour transformer le texte en un vecteur numérique.

4. Analyse des Données Chiffrées

En parallèle, les informations comme le montant et la date sont converties en un format standardisé, afin d'être interprétées plus efficacement par le modèle.

5. Fusion des Informations

Les données textuelles et numériques sont ensuite fusionnées dans un seul vecteur, qui servira d'entrée pour l'apprentissage.

6. Apprentissage Automatique

Le modèle d'apprentissage, basé sur LightGBM, apprend à reconnaître les catégories à partir des exemples fournis. Il s'entraîne en associant chaque transaction à sa catégorie réelle.

7. Sauvegarde du Modèle

Une fois l'apprentissage terminé, tous les composants nécessaires sont sauvegardés pour une utilisation ultérieure :

- Le modèle lui-même
- Les outils de traitement des données
- Les configurations d'encodage

Conclusion

En résumé, l'architecture technique de la plateforme e-Banking allie modularité, intelligence artificielle et interopérabilité avec les systèmes bancaires existants. Grâce à une approche microservices sécurisée, un traitement intelligent des transactions, et l'intégration d'un chatbot basé sur GPT-4o + RAG, la solution garantit performance, évolutivité et personnalisation. L'ensemble du système repose sur des choix technologiques modernes assurant robustesse, traçabilité et expérience utilisateur enrichie.

Chapitre IV :

Réalisation

I. Introduction

Dans ce chapitre, nous allons présenter la réalisation de notre application e-Banking à travers une série de captures d'écran illustrant les différentes interfaces développées. Chaque écran reflète une fonctionnalité précise mise en place durant le projet, aussi bien côté titulaire de carte que côté agent. L'objectif est de donner un aperçu visuel du travail accompli et de montrer concrètement le résultat final de notre développement.

II. Réalisation de l'application mobile – Espace Cardholder

Dans cette section, nous allons découvrir la partie mobile de notre application dédiée aux titulaires de cartes (Cardholders). À travers plusieurs captures d'écran, nous illustrerons les différentes interfaces et fonctionnalités accessibles depuis l'application mobile, telles que la gestion des cartes, les transactions, les préférences, ou encore les plans de voyage.

1. Landing page

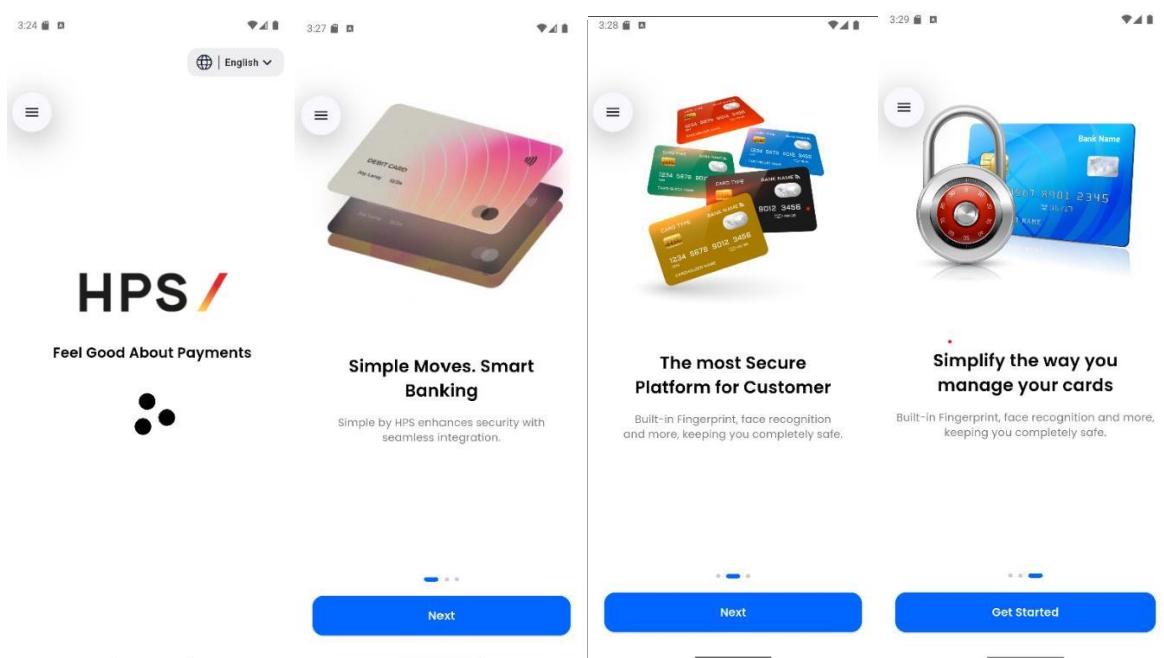


Figure 19 : Landing Pages

- Écran d'introduction de l'app e-Banking HPS.
- Présente les atouts principaux :
 - Carte bancaire intelligente et moderne.

- Plateforme sécurisée pour les clients.
- Gestion simplifiée des cartes.
- Navigation par étapes avec boutons “Next” et “Get Started”.
- Option multilingue visible en haut de l’écran.

3. Logging

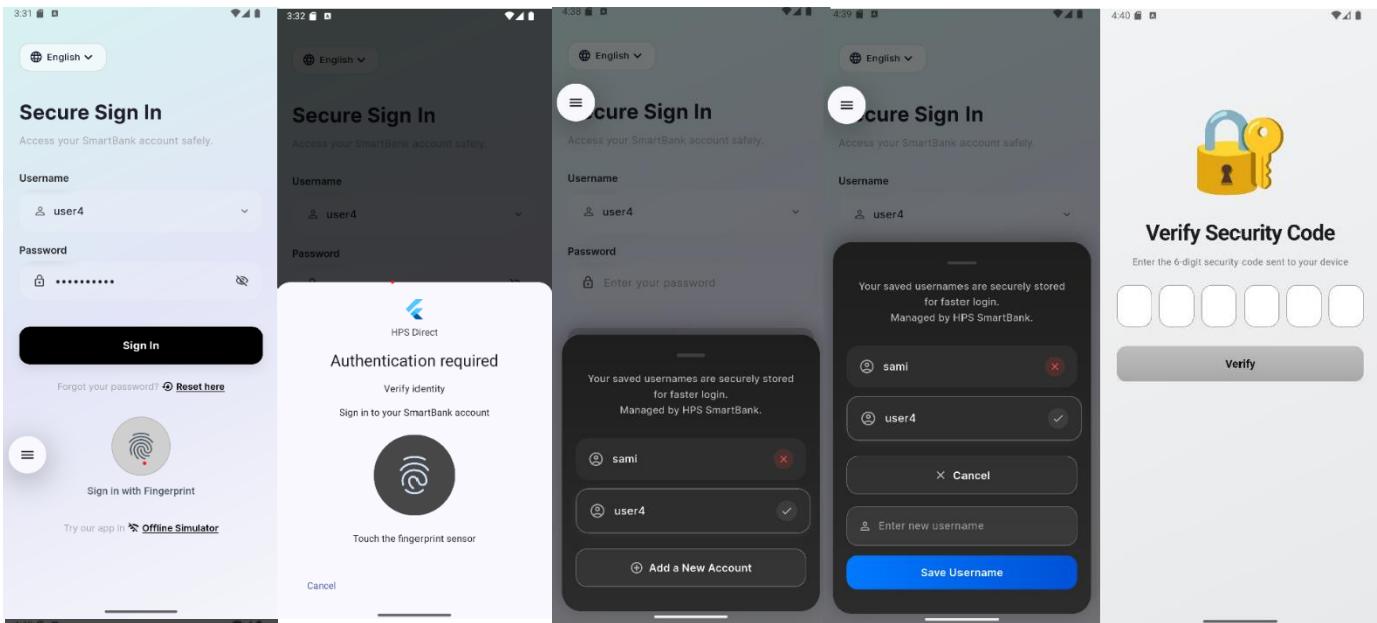
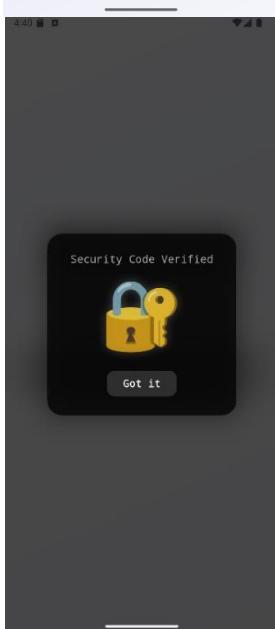


Figure 20: Logging and Touch ID



- Ce processus illustre la connexion sécurisée à l’application mobile.
- L’utilisateur peut s’authentifier via mot de passe ou empreinte digitale.
- La fonctionnalité « Touch ID » est également disponible.
- Lors de la première connexion, un code de sécurité à 6 chiffres est défini.
- Ce code est ensuite saisi pour vérification et sécurisation du compte.
- L’interface est conçue pour offrir une expérience fluide et moderne.

4. Home Page

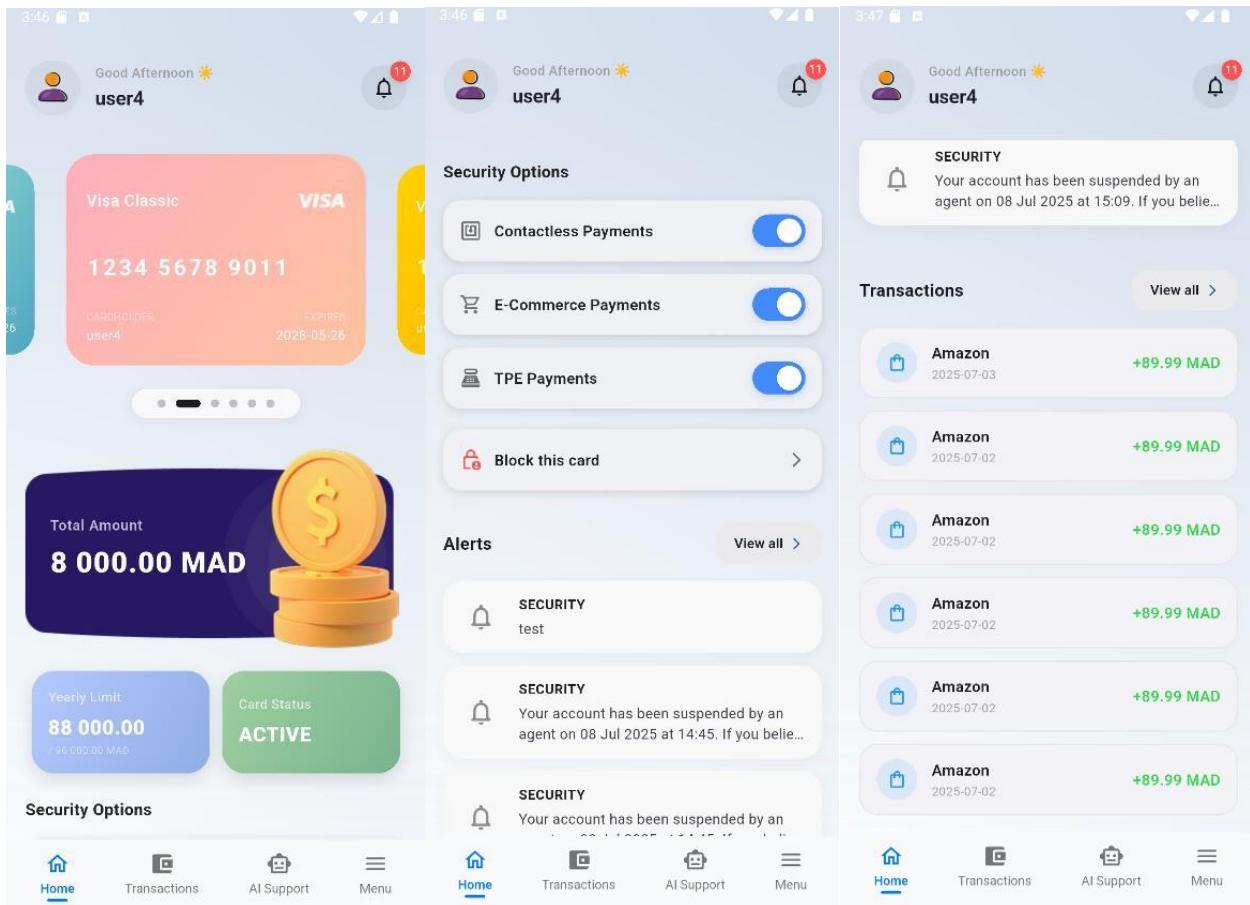


Figure 21: Home

Il s'agit de la page **Home** de l'application mobile dédiée au **Cardholder**, offrant une vue d'ensemble sur sa carte et ses opérations :

- Affichage de la carte bancaire avec type, numéro et statut (ACTIVE).
- Visualisation du solde total disponible et de la limite mensuelle.
- Accès direct aux **options de sécurité** : paiements sans contact, en ligne, TPE.
- Bouton de blocage temporaire de la carte.
- Section **Alerts** pour consulter les messages de sécurité importants.
- Liste des **transactions récentes** avec nom du commerçant et montant affiché.

5. Notification

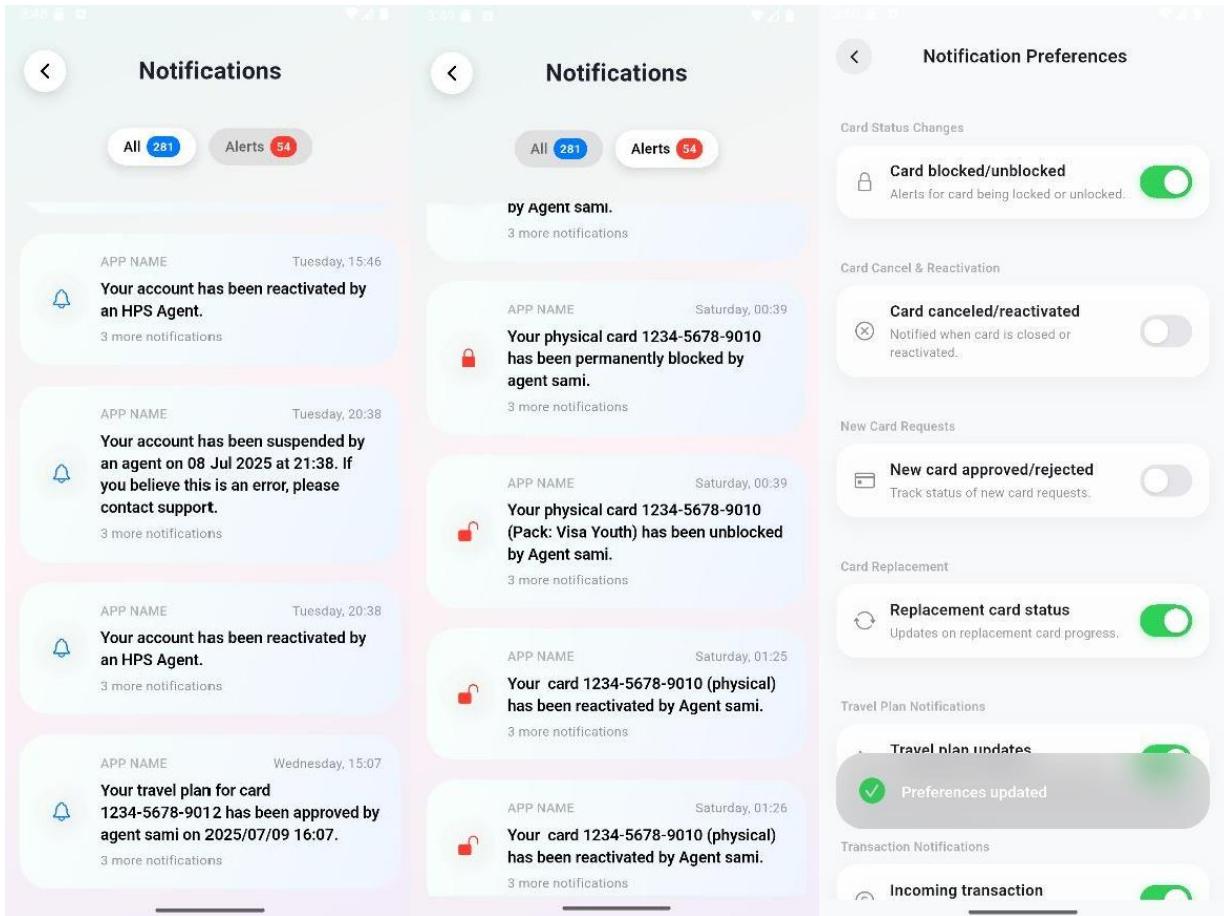


Figure 22 : Notifications

Il s'agit de la page **Notifications** du cardholder permettant de suivre les événements liés à sa carte bancaire :

- Affichage des notifications triées par type : **Toutes** ou **Alertes**.
- Détail de chaque événement : suspension, réactivation, approbation ou blocage de carte.
- Présence du nom de l'agent ayant réalisé l'action.
- Section **Notification Preferences** pour activer/désactiver les alertes selon le type :
 - Blocage / déblocage.
 - Activation / annulation.
 - Requêtes de carte ou remplacements.
 - Transactions entrantes ou plans de voyage.

6. Transactions

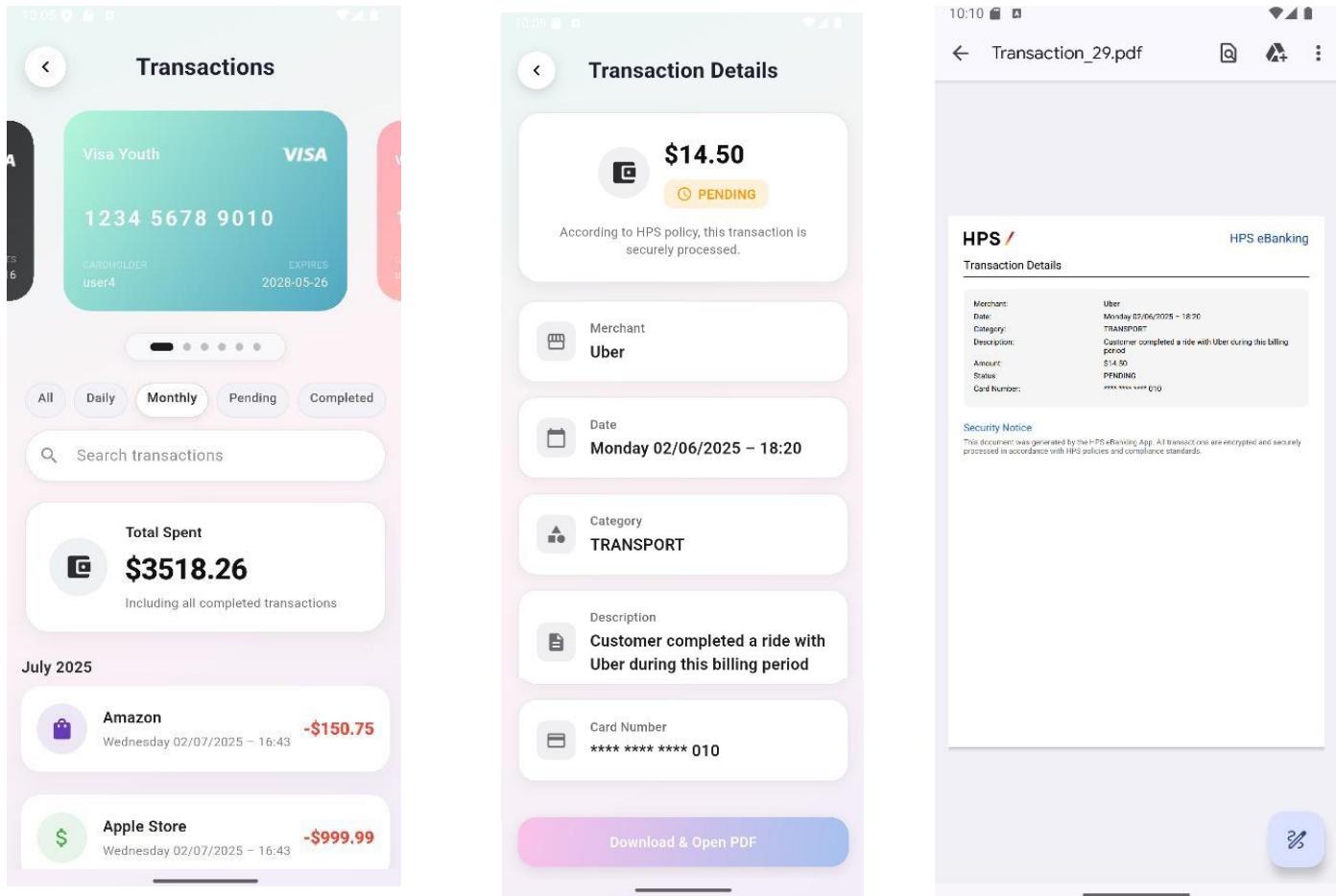


Figure 23 : Transactions

L'utilisateur accède à l'écran des transactions liées à sa carte.

– Il peut **filtrer les transactions** par période (All, Daily, Monthly...) et rechercher une transaction spécifique.

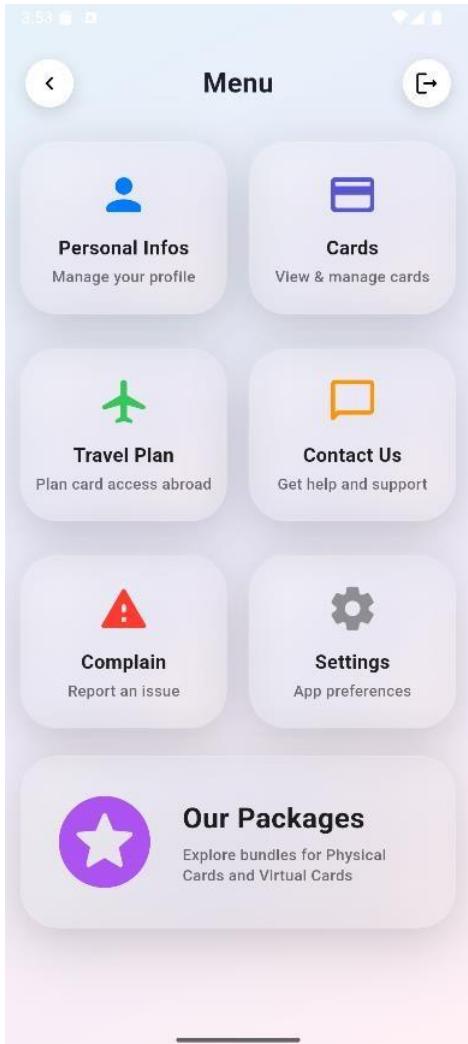
– Chaque ligne montre : Le **marchand** (ex. Amazon, Apple Store), La **date**, Le **montant** (positif ou négatif) et un visuel clair avec une carte et le total dépensé.

– En cliquant sur une transaction, l'utilisateur accède à ses **détails** :

- Montant, statut (ex : Pending)
- Marchand, date, catégorie,
- Derniers chiffres de la carte utilisée

– Il peut **générer et visualiser un reçu PDF** officiel avec toutes les informations pour archivage ou usage administratif.

7. Menu



Il s'agit du **menu principal** de l'application mobile du cardholder :

- **Personal Infos** : gérer les informations personnelles du profil.
- **Cards** : consulter et gérer les cartes bancaires.
- **Travel Plan** : déclarer des déplacements à l'étranger pour activer les cartes.
- **Contact Us** : contacter le support HPS.
- **Complain** : signaler un problème ou déposer une plainte.
- **Settings** : modifier les préférences de l'application.
- **Our Packages** : découvrir les offres de cartes physiques et virtuelles.

Figure 24 : Menu

8. Personal infos

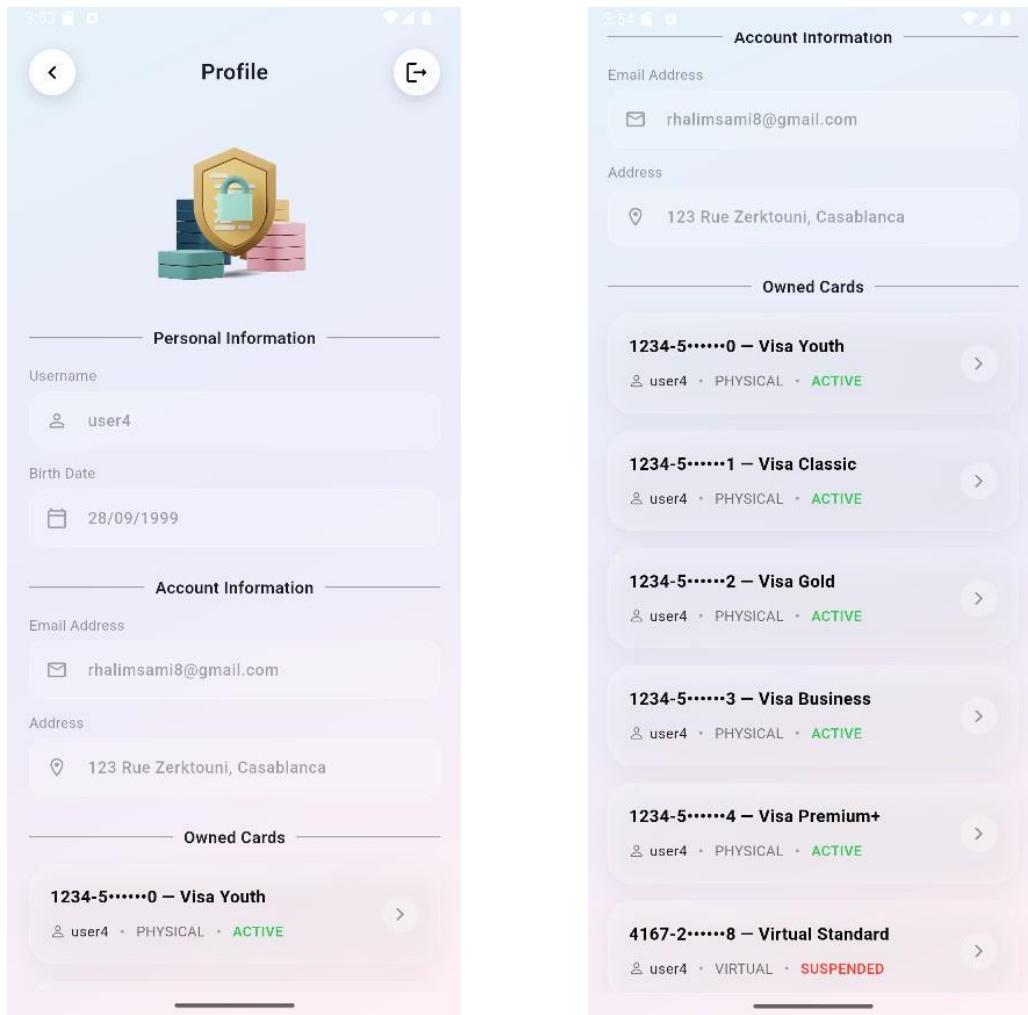


Figure 25 : Profile

Il s'agit de l'écran **Profil** du cardholder :

- Affiche les informations personnelles : nom d'utilisateur, date de naissance, email, adresse.
- Liste toutes les cartes bancaires possédées (physiques ou virtuelles).
- Chaque carte indique son type (Youth, Classic, Gold...), son format (PHYSICAL ou VIRTUAL) et son statut (ACTIVE ou SUSPENDED).
- Permet d'accéder aux détails de chaque carte en cliquant dessus.

9. Cards

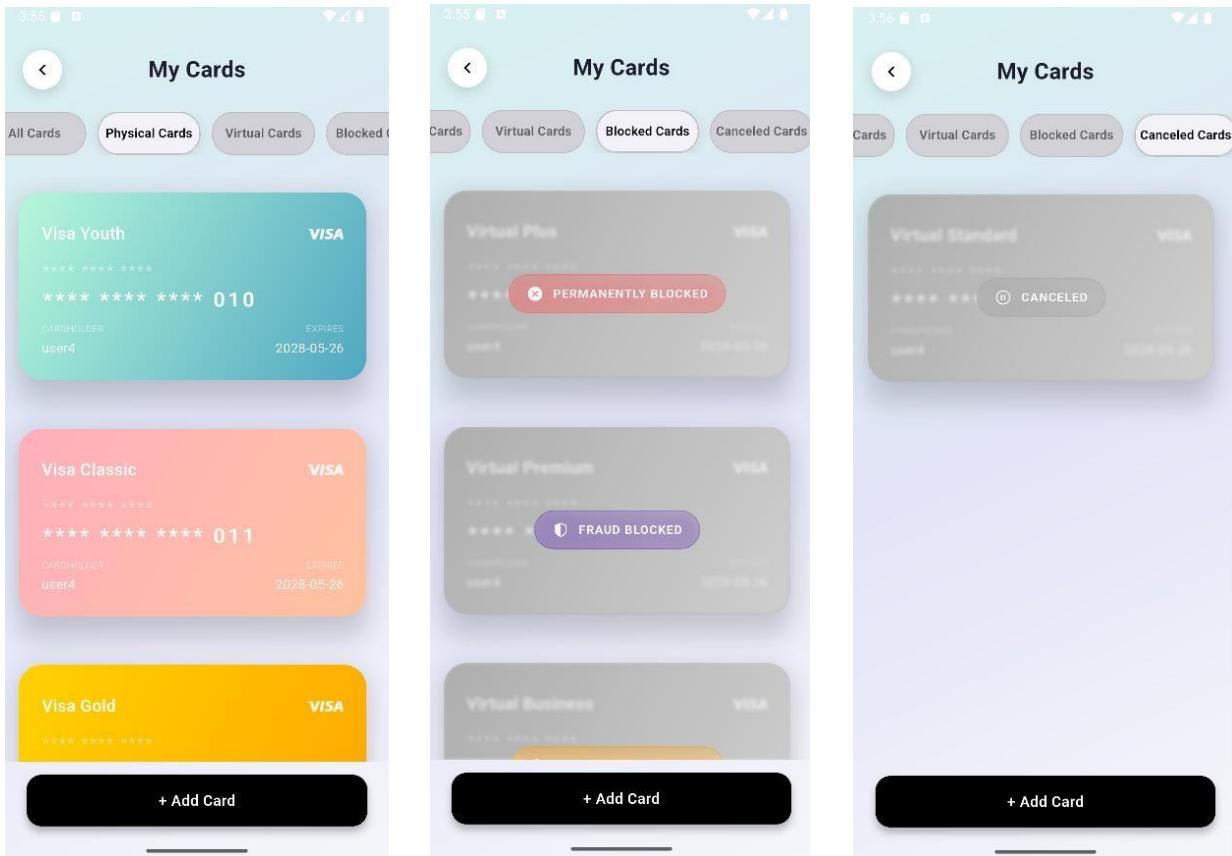
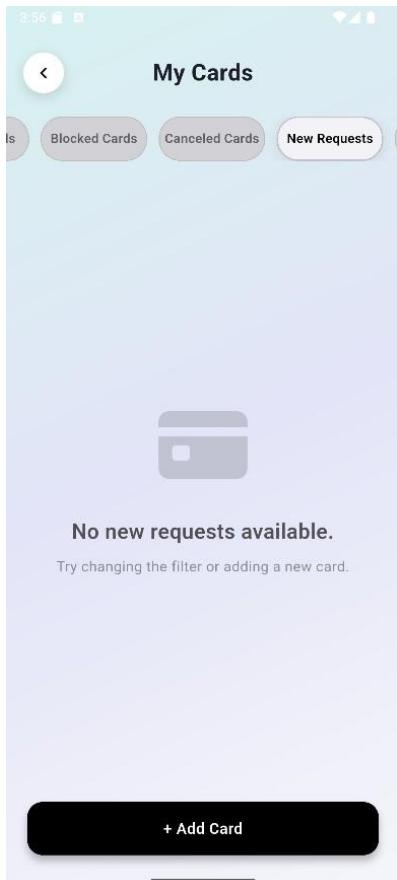


Figure 26: Cards



Cette page permet au détenteur de carte de visualiser et gérer toutes ses cartes selon leur statut ou type.

- Affichage catégorisé : cartes physiques, virtuelles, bloquées, annulées et demandes en cours.
- Présentation claire avec design moderne et filtres en haut.
- Possibilité d'ajouter une nouvelle carte via le bouton “+ Add Card”.
- Statuts visibles sur chaque carte : active, bloquée (fraude ou permanente), annulée.
- Informations essentielles visibles : nom, type, numéro partiel et date d'expiration.

10.Add Cards



Figure 27: Add Card

11. Physical Cards

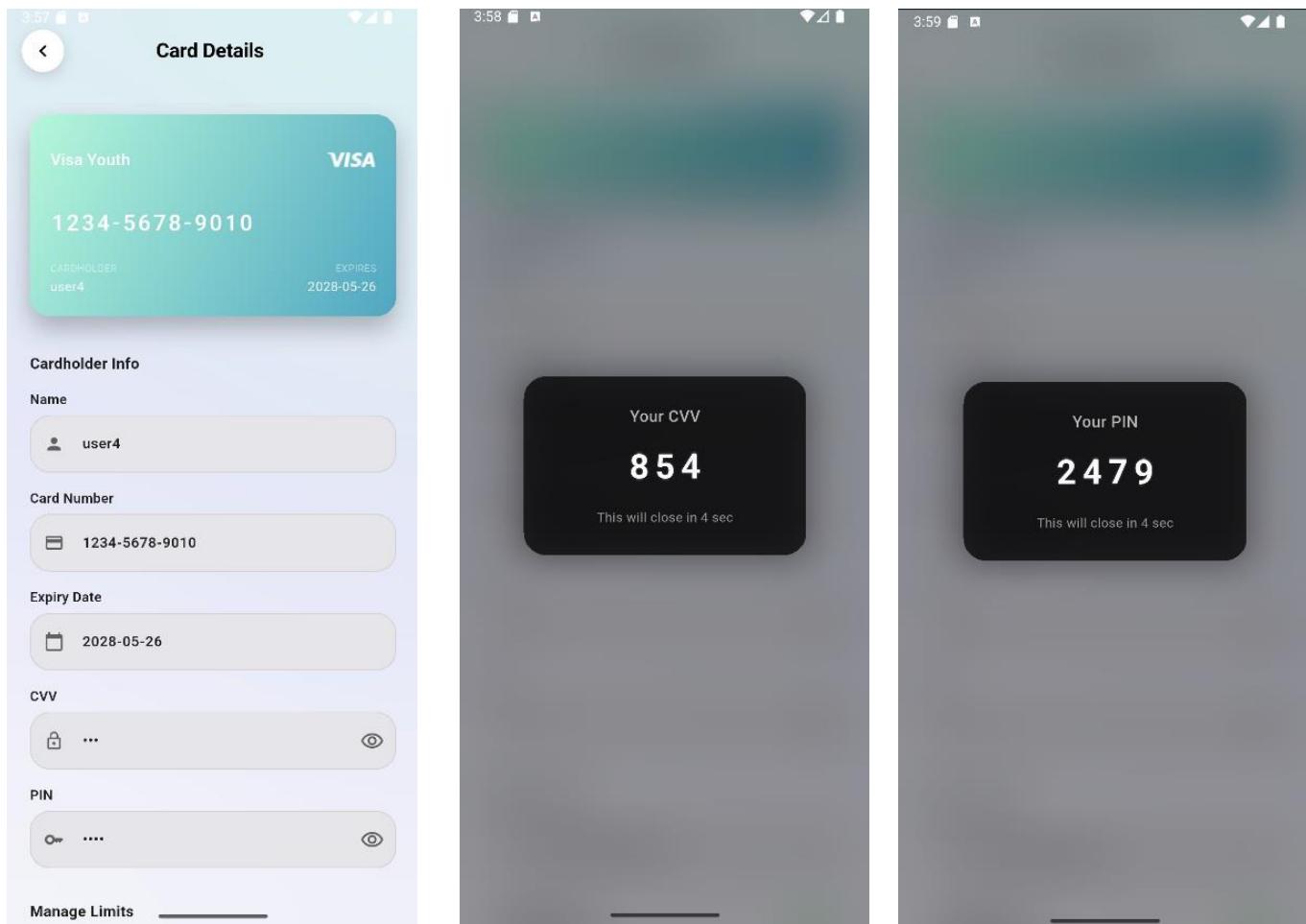


Figure 28: Pysical Cards /CVV/PIN

- L'utilisateur accède aux détails d'une carte physique (ex. Visa Youth).
- Les informations principales sont affichées : nom du titulaire, numéro de carte, date d'expiration.
- Il peut consulter le CVV et le code PIN via des écrans sécurisés, affichés temporairement pendant quelques secondes.
- Cette visualisation permet une consultation rapide et sécurisée des informations sensibles.

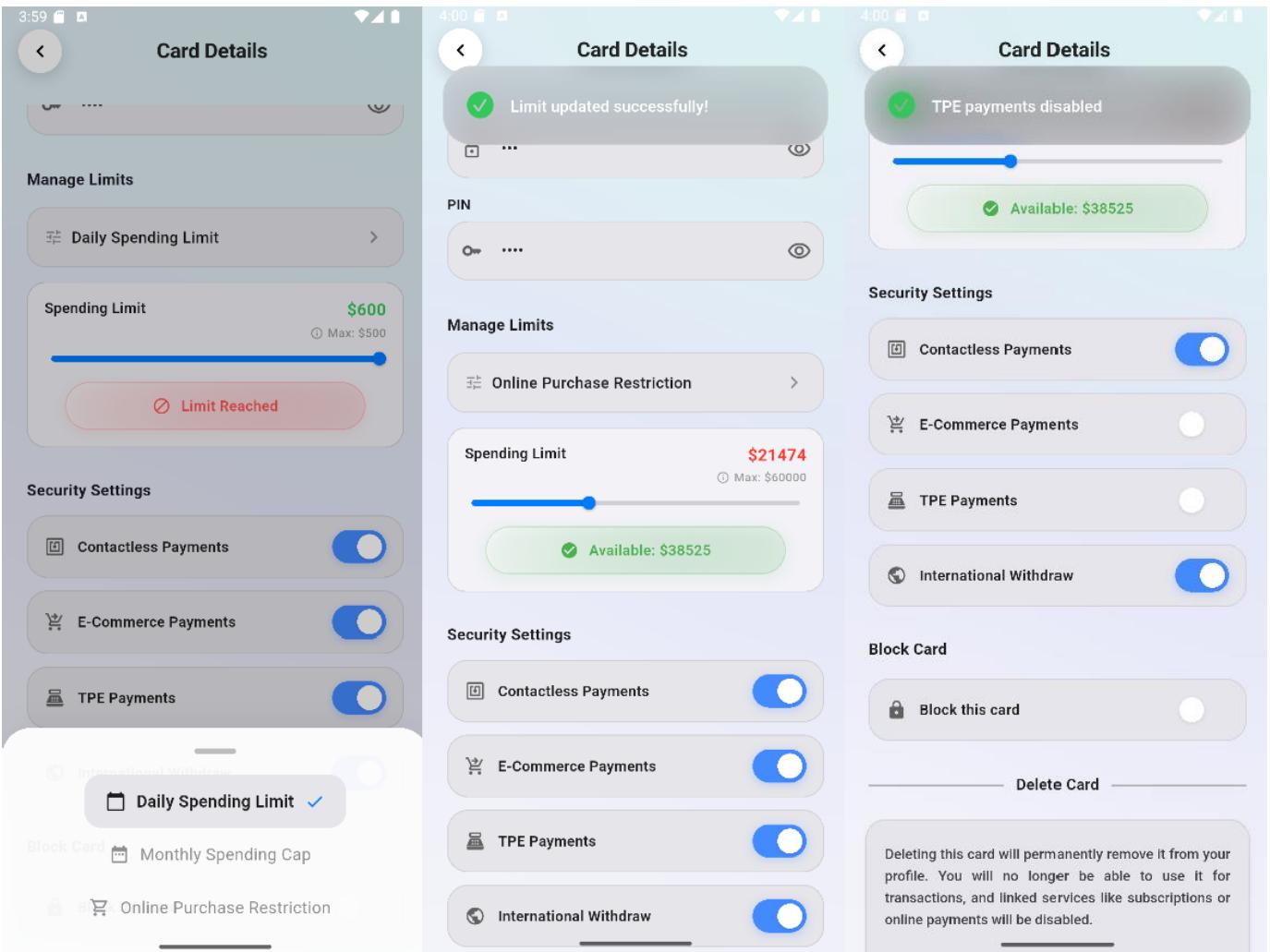


Figure 29 :Card Details

L'utilisateur accède à l'écran de gestion de carte avec une vue claire des paramètres de la carte. Il peut modifier les limites de dépenses, comme :

- **Plafond quotidien** : avec indication visuelle si le plafond est atteint ("Limit Reached").
- **Restriction d'achat en ligne** : avec affichage du montant actuel utilisé et du solde restant.

Les options de sécurité sont présentées sous forme de switches interactifs :

- **Paiements sans contact**

- **Paiements e-commerce**

- **Paiements via TPE**

- **Retraits internationaux**

Chacune peut être activée ou désactivée individuellement selon les préférences du titulaire.

À chaque modification, un **message de confirmation dynamique** s'affiche en haut de l'écran (ex : « TPE payments disabled » ou « Limit updated successfully ») pour informer l'utilisateur du succès de son action.

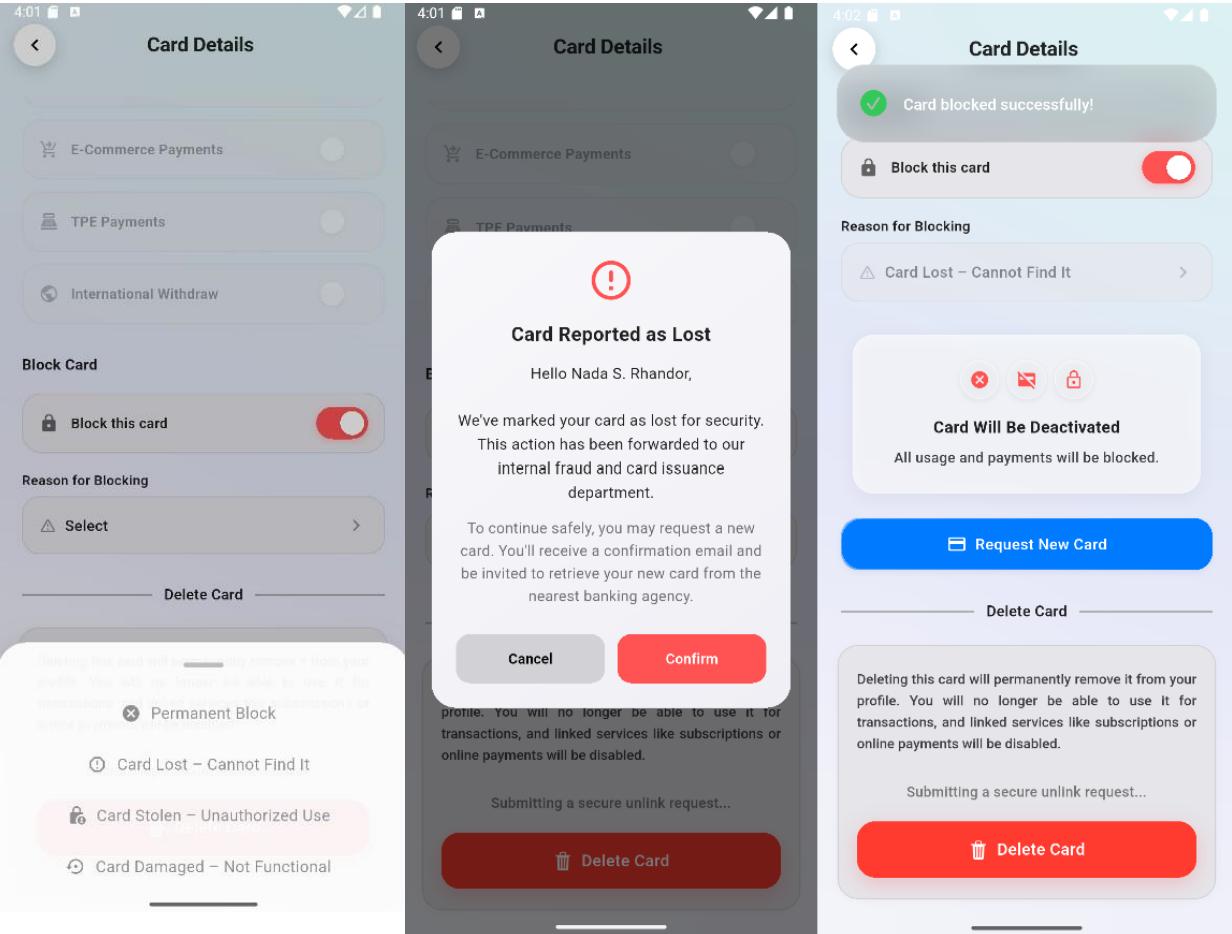


Figure 30: Block Physical Card

- L'utilisateur peut **bloquer manuellement une carte** en activant le bouton **Block this card**.
- Il doit ensuite **sélectionner un motif** parmi plusieurs options :
 - Permanent Block
 - Card Lost – Cannot Find It
 - Card Stolen – Unauthorized Use
 - Card Damaged – Not Functional
- Une fois le motif sélectionné, un **message de confirmation contextuel** s'affiche (ex : carte déclarée perdue).
- Si le blocage est confirmé, un message "Card blocked successfully" s'affiche.

- L'option “Request New Card” devient alors disponible.
- L'utilisateur peut aussi supprimer définitivement la carte via le bouton “Delete Card”.

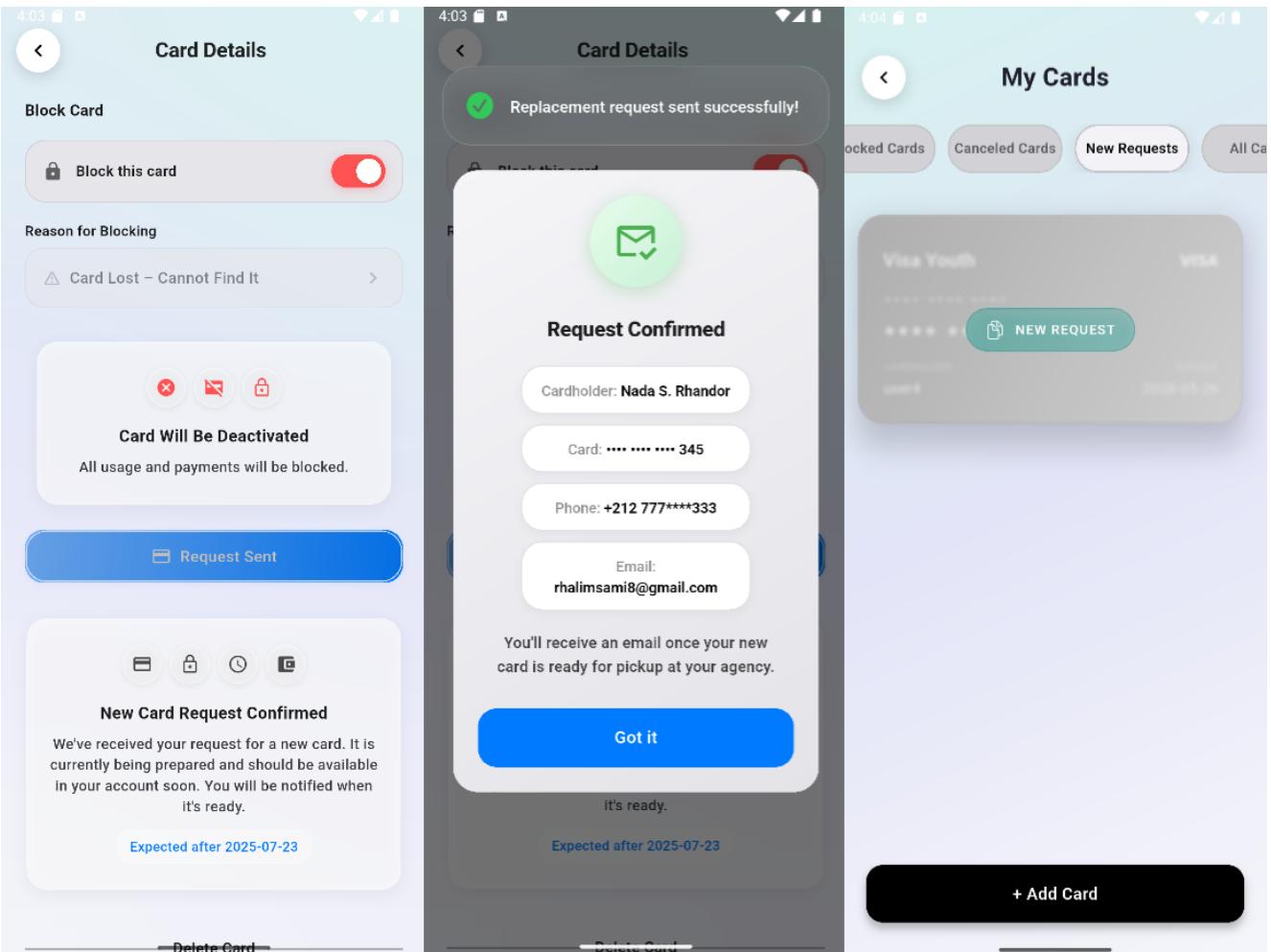


Figure 31:Request New Card

- Après avoir bloqué une carte, l'utilisateur peut envoyer une demande de remplacement via le bouton “Request Sent”.
- Un message “New Card Request Confirmed” s'affiche avec une date estimée de disponibilité.
- Une fenêtre de confirmation récapitule les informations du titulaire : nom, carte, téléphone, email.
- Une fois la demande envoyée, la nouvelle carte apparaît dans l'onglet “New Requests” avec l'état In Progress.
- L'utilisateur sera informé par email lorsque la carte sera prête à être récupérée en agence.

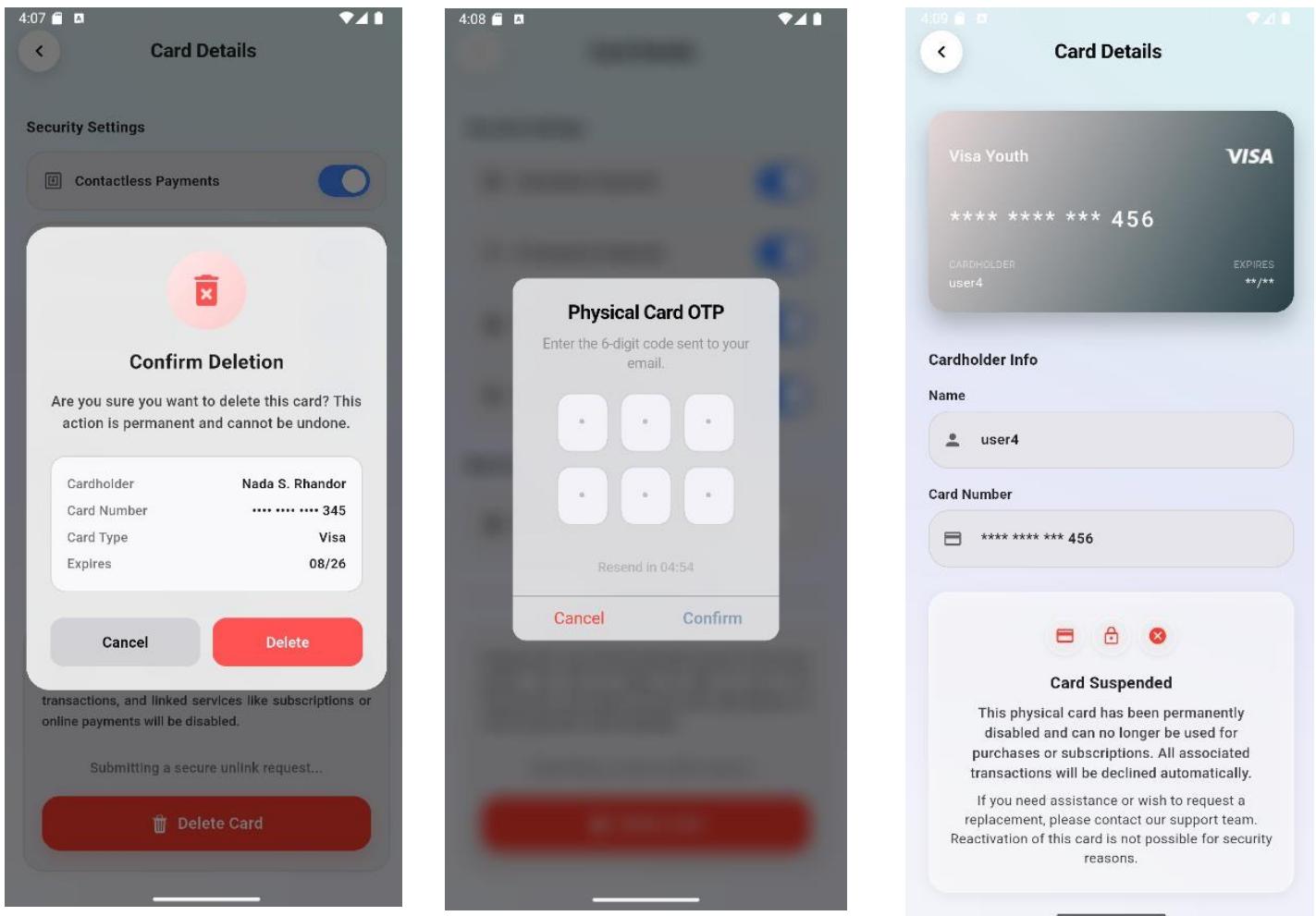


Figure 32: Cancel Card

Ces captures d'écran illustrent le processus complet de suppression d'une carte physique dans l'application mobile du cardholder :

- La première capture affiche une fenêtre de confirmation avant la suppression, contenant les informations principales de la carte (nom, numéro, type, date d'expiration). L'utilisateur peut choisir d'annuler ou de confirmer l'action.
- La deuxième capture montre l'écran de saisie du code OTP à usage unique (6 chiffres), envoyé par email afin de sécuriser l'opération.
- La troisième capture indique que la carte a bien été suspendue définitivement. Elle ne pourra plus être utilisée pour des paiements ou abonnements. Un message informe l'utilisateur qu'il doit contacter le support pour toute assistance.
- La dernière image montre l'email reçu contenant le code OTP de vérification, valable pendant 5 minutes.

Ce processus assure une suppression sécurisée et irréversible de la carte, avec validation par code de sécurité.

12. Virtual Cards

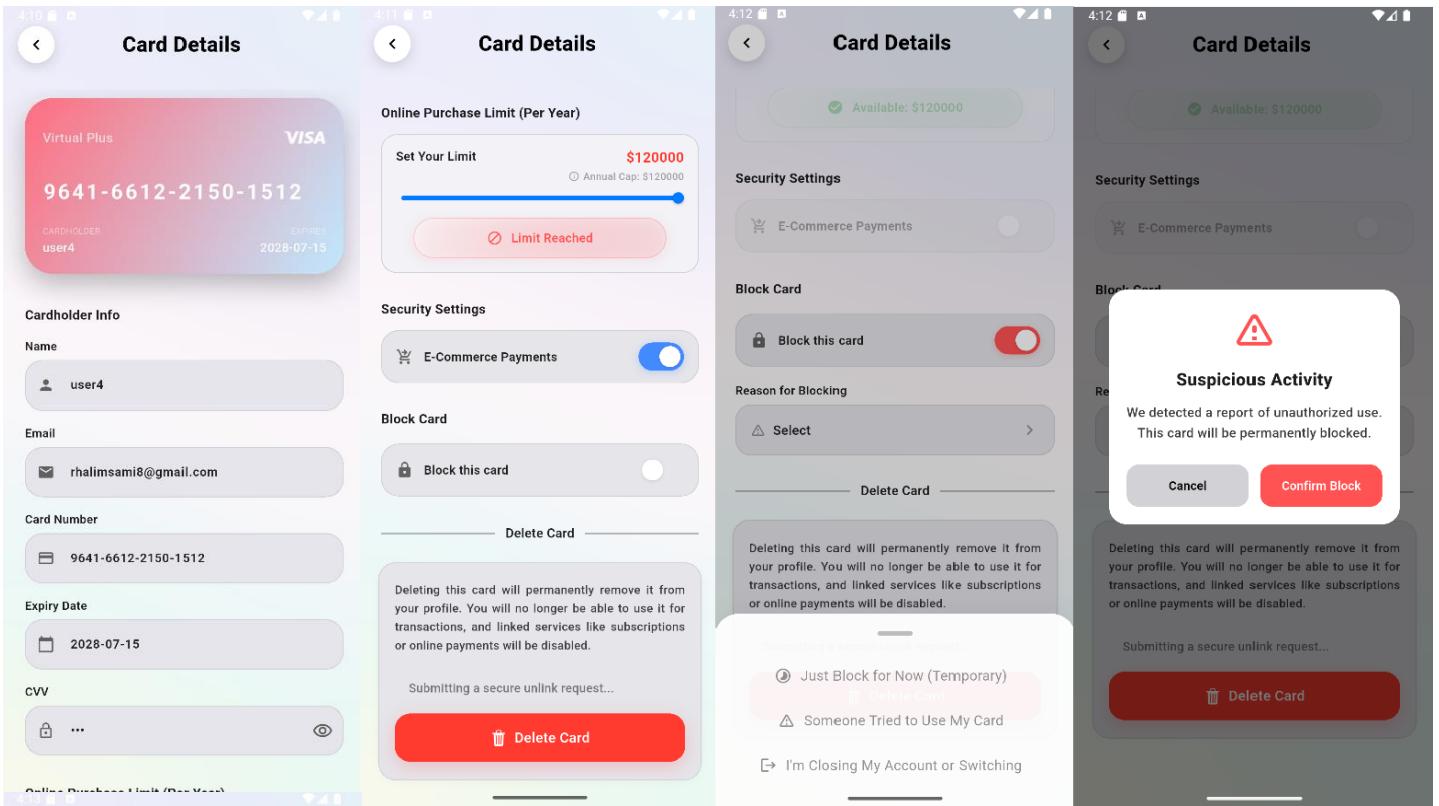


Figure 33: Virtual Card Details

Ces captures d'écran illustrent le scénario de blocage immédiat d'une carte physique suite à une activité suspecte signalée par l'utilisateur :

- L'utilisateur accède aux détails de sa carte (nom, numéro, email, etc.) et remarque une anomalie.
- Un blocage rapide est effectué via l'option “**Block this card**” après sélection du motif “**Suspicious Activity**” (ex : tentative d'utilisation non autorisée).
- Une alerte contextuelle informe que la carte est immédiatement désactivée. Un message explique que toutes les opérations futures seront bloquées pour garantir la sécurité du titulaire.
- L'application propose ensuite à l'utilisateur de demander une **nouvelle carte**. Un message d'avertissement s'affiche si une demande de remplacement est déjà en cours.

Ce processus permet de réagir rapidement à toute activité frauduleuse en bloquant la carte concernée et en lançant le processus de remplacement de

façon sécurisée.

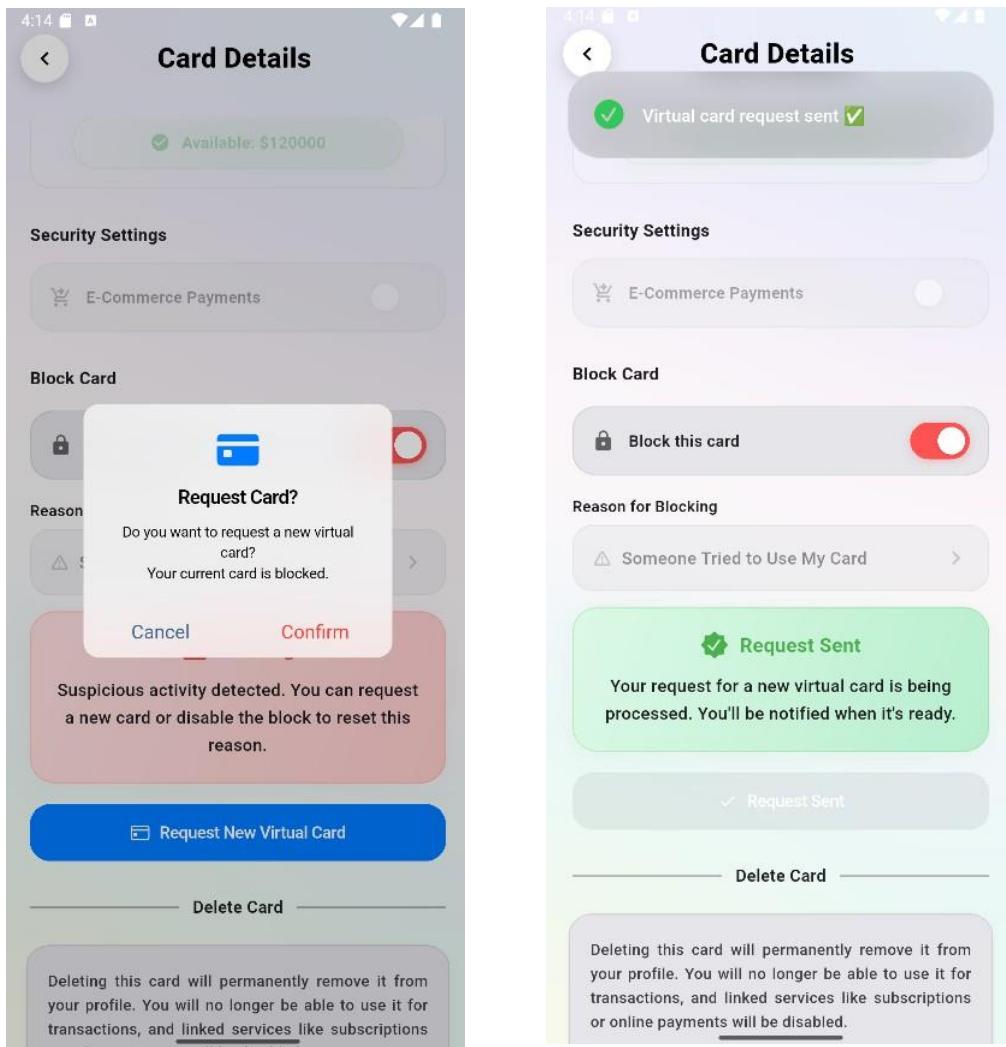


Figure 34: Request New Virtual Card

Ces deux écrans illustrent la procédure de **demande d'une nouvelle carte virtuelle** après signalement d'une activité suspecte :

- **Blocage de la carte** : l'utilisateur sélectionne le motif “*Someone Tried to Use My Card*”. Un message d'alerte rouge informe qu'une activité suspecte a été détectée et qu'une nouvelle carte peut être demandée.
- **Demande de nouvelle carte** :
 - Une fenêtre de confirmation s'affiche : “*Do you want to request a new virtual card?*”. L'utilisateur doit valider l'action en cliquant sur **Confirm**.
 - Une fois la demande envoyée, un message vert “*Request Sent*” confirme que la nouvelle carte est en cours de traitement.

Ce processus assure une réaction rapide et sécurisée face à une tentative de fraude en permettant au titulaire de remplacer immédiatement sa carte compromise.

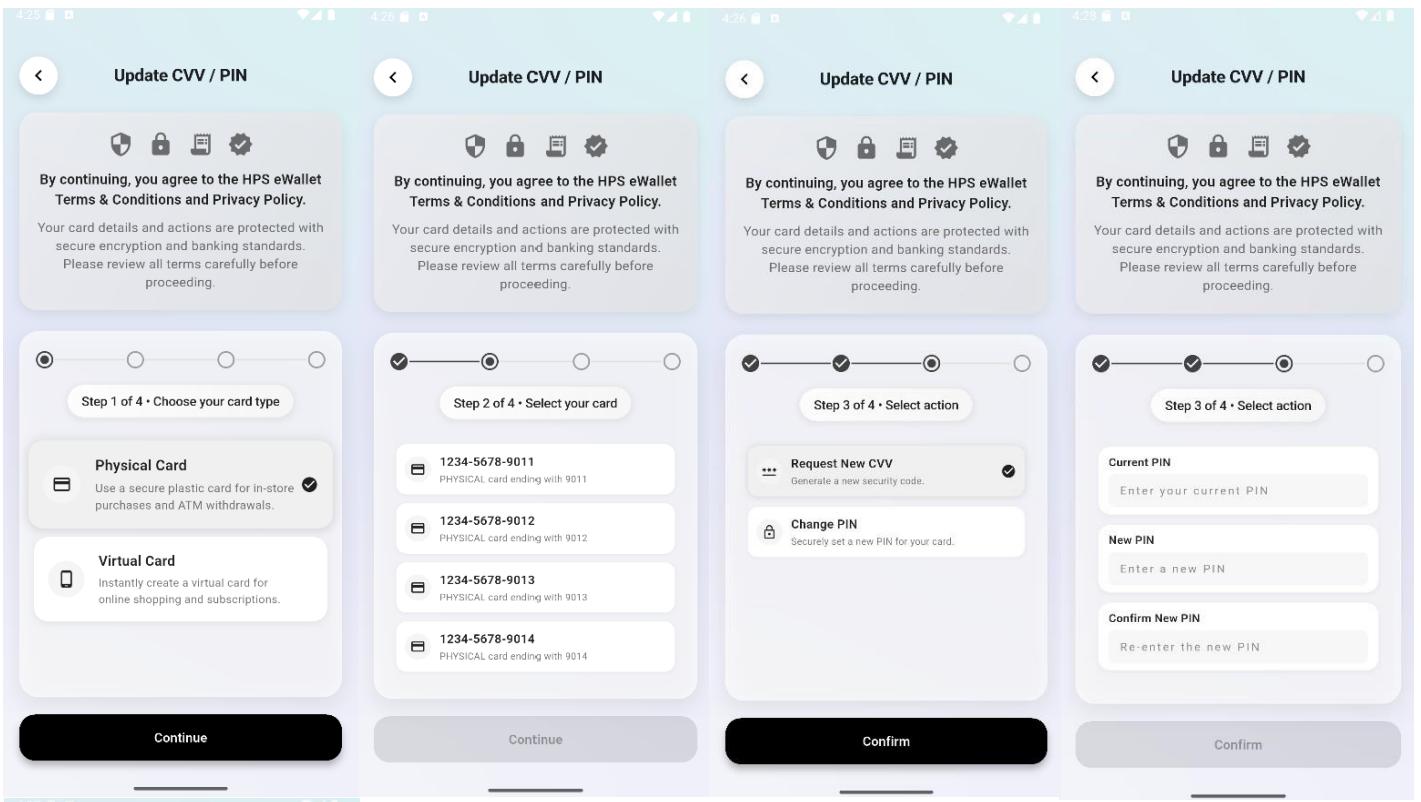
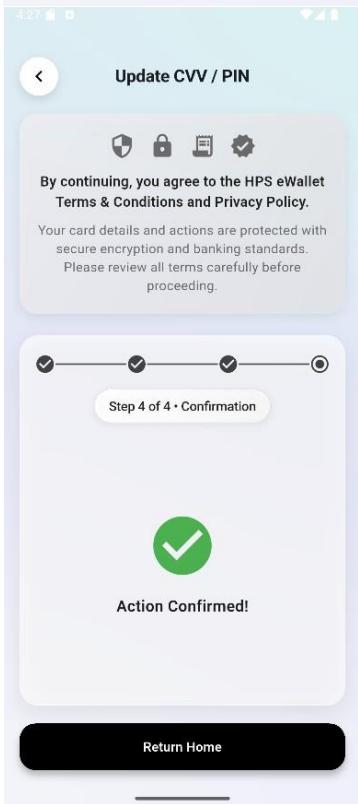


Figure 35 :Update CVV and PIN



Ces écrans montrent une procédure simple et sécurisée pour mettre à jour le CVV ou le code PIN d'une carte :

- L'utilisateur choisit d'abord s'il souhaite modifier une carte physique ou virtuelle.
- Il sélectionne ensuite la carte concernée parmi celles actives.
- Il décide de l'action à effectuer : régénérer un CVV ou changer un code PIN.
- Si l'action concerne le code PIN, l'utilisateur saisit un nouveau code sécurisé.
- Une fois les informations fournies, une confirmation finale est affichée indiquant que l'action a bien été prise en compte.

13. Travel Plan

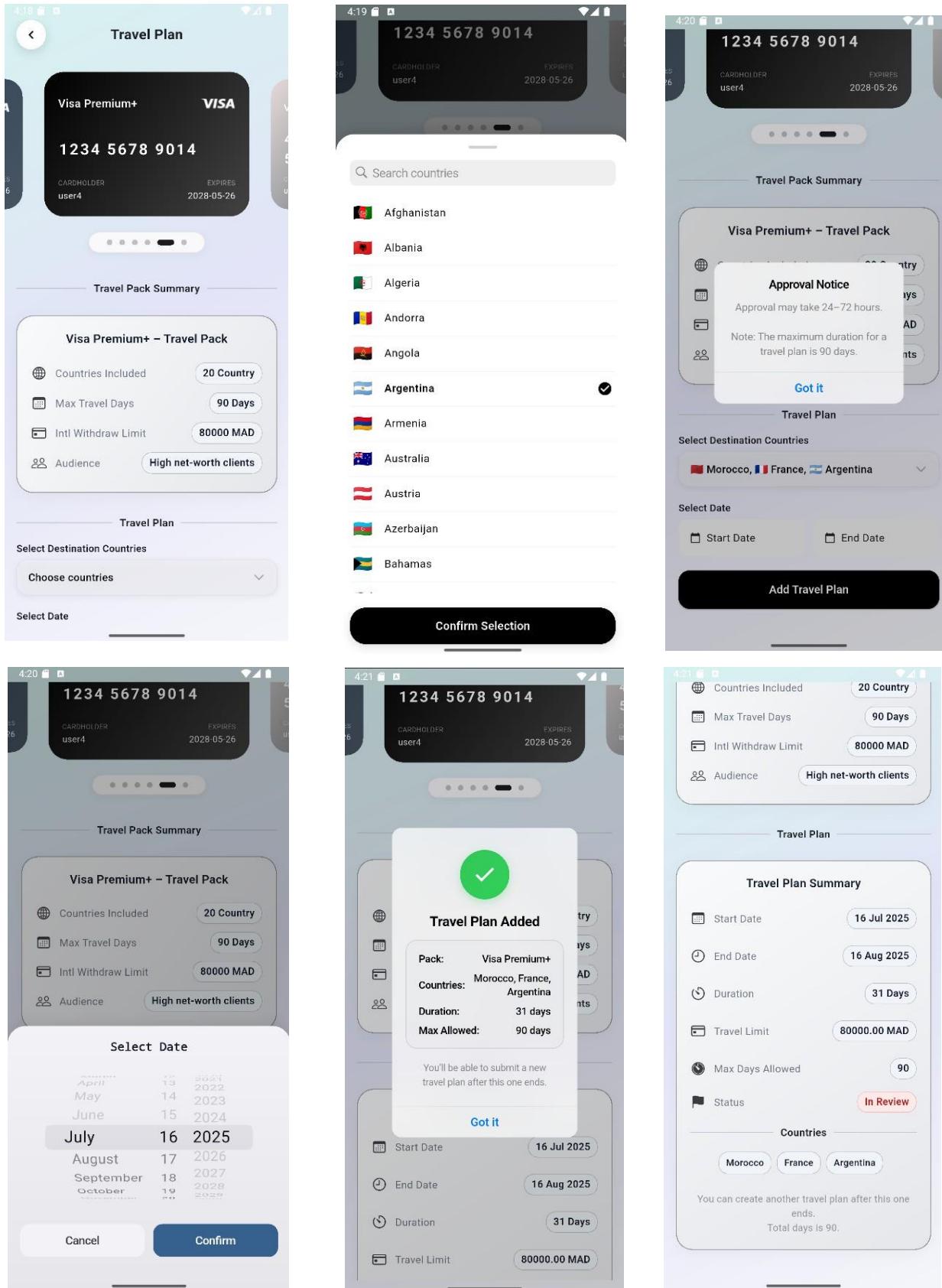
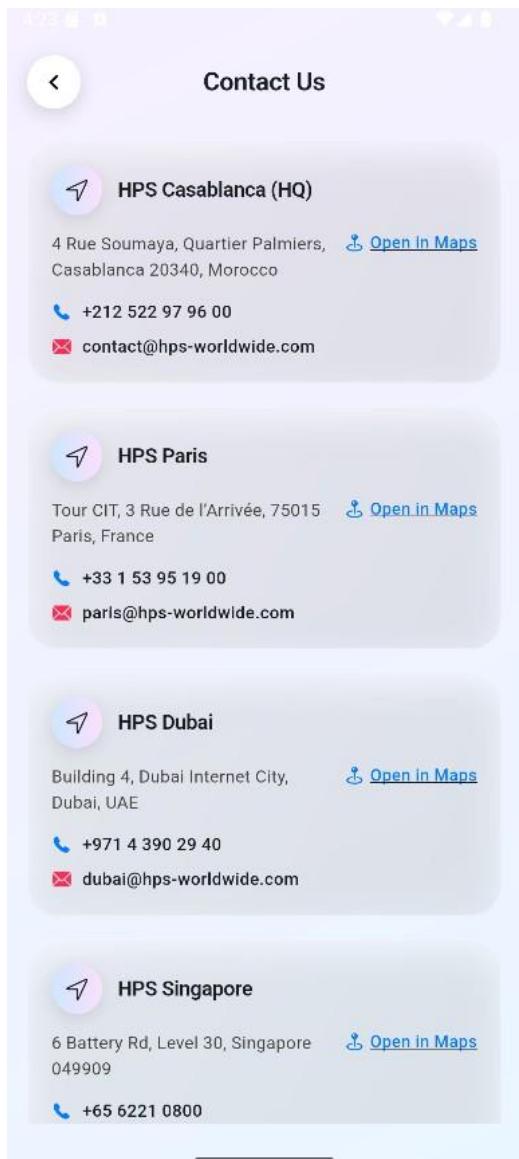


Figure 36: Travel Plan

Ces écrans illustrent le processus complet d'ajout d'un plan de voyage :

- L'utilisateur sélectionne une carte active et un pack de voyage (ex. : Visa Premium – Travel Pack).
- Il choisit ensuite les destinations (plusieurs pays possibles) dans une liste déroulante avec recherche.
- Une fois les pays sélectionnés, un message de confirmation l'informe que le plan doit être validé.
- L'utilisateur choisit les dates de début et de fin du voyage via un calendrier intégré.
- Un message confirme l'ajout du plan de voyage avec un récapitulatif : carte utilisée, destinations, nombre de pays et période.
- Enfin, un résumé complet s'affiche avec toutes les informations du plan enregistré.

14. Contact Us



Cet écran présente la section **Contactez-nous** de l'application, permettant aux utilisateurs de retrouver les coordonnées des différentes agences HPS à travers le monde :

- **HPS Casablanca (HQ)** : adresse principale au Maroc avec numéro de téléphone et email.
- **HPS Paris** : bureau en France avec lien direct pour ouvrir dans Maps.
- **HPS Dubaï** : bureau aux Émirats avec coordonnées complètes.
- **HPS Singapour** : bureau en Asie du Sud-Est avec numéro d'appel.

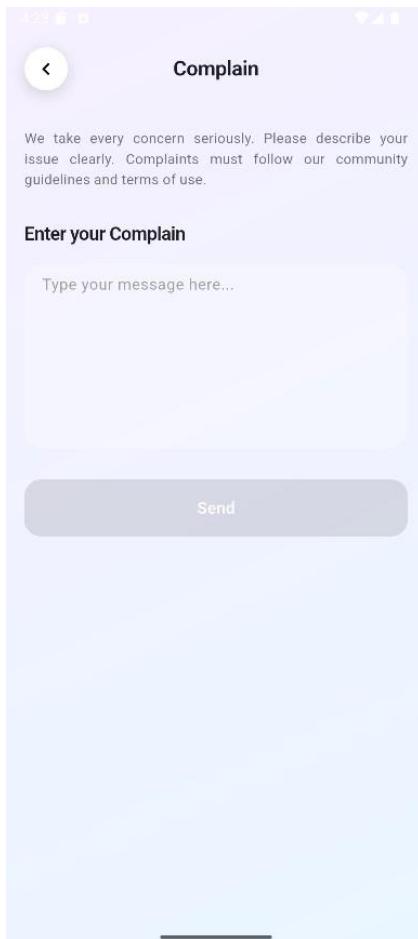
Chaque agence dispose :

- D'une adresse physique,
- D'un numéro de téléphone,
- D'une adresse email (si disponible),
- Et d'un lien pour l'ouverture dans une application de cartographie.

Cela facilite la prise de contact ou la localisation de l'agence la plus proche.

Figure 37: Contact Us

15. Complain



Cet écran permet à l'utilisateur de soumettre une **réclamation** ou un **retour** concernant un problème rencontré avec sa carte, son compte ou un service de l'application.

Composants visibles :

- **Titre** : *Complain* (Réclamation) – indique que l'utilisateur peut formuler une plainte.
- **Texte explicatif** : invite l'utilisateur à **décrire clairement son souci**, en précisant que les réclamations doivent respecter les **règles de la communauté**.
- **Zone de saisie** : champ de texte dans lequel l'utilisateur peut **taper son message** librement.
- **Bouton "Send" désactivé** : le bouton d'envoi est grisé tant que le champ de texte est vide

Figure 38 : Complain

16. Settings

Cet écran regroupe toutes les options de configuration liées à l'application e-Banking :

- **Notifications** : Permet de gérer les alertes et préférences de notification envoyées par l'application.
- **Politique de confidentialité** : Donne accès aux conditions d'utilisation et à la politique de protection des données personnelles.
- **Changer la langue** : Permet de sélectionner la langue d'affichage de l'application selon les préférences de l'utilisateur.
- **Mettre à jour le CVV / PIN** : Donne la possibilité de modifier les codes de sécurité associés aux cartes bancaires.
- **Activer l'empreinte digitale** : Active la connexion par empreinte digitale pour renforcer la sécurité de l'application.

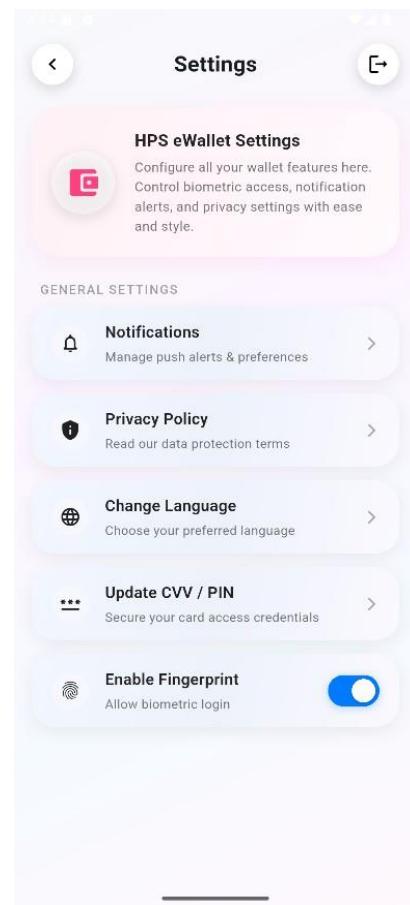
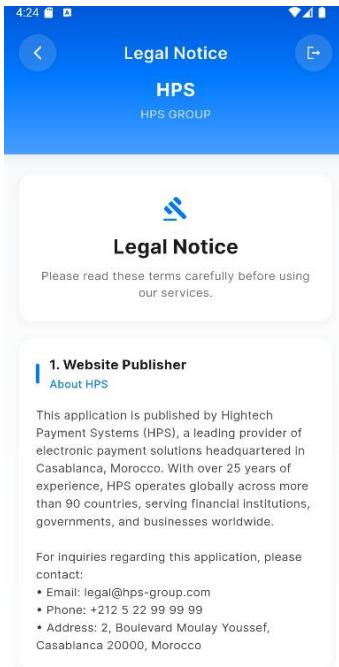


Figure 39: Settings

En haut de l'écran, une section informative présente brièvement l'objectif de ces paramètres : configurer facilement et avec style les fonctions du portefeuille numérique HPS, telles que la sécurité biométrique, les alertes ou la confidentialité.

17. Policy



Cet écran affiche les informations légales de l'application, publiée par **HPS**, entreprise marocaine spécialisée dans les solutions de paiement.

On y trouve :

- Les coordonnées de HPS (email, téléphone, adresse)
- Un rappel sur la **protection des données personnelles**
- Des liens vers plus de détails (ex. : "About HPS", "Your Privacy Matters")

Figure 40 :Policy

18. Card Packs

Ce screen présente les **packs de cartes** disponibles, répartis en deux catégories : **virtuelles** et **physiques**.

Ici, l'utilisateur a sélectionné un **pack virtuel** (*Virtual Standard*), avec un aperçu clair de ses caractéristiques : type de carte, durée de validité, limite annuelle, activation e-commerce, et frais associés. Cela permet de comparer facilement les options selon les besoins.

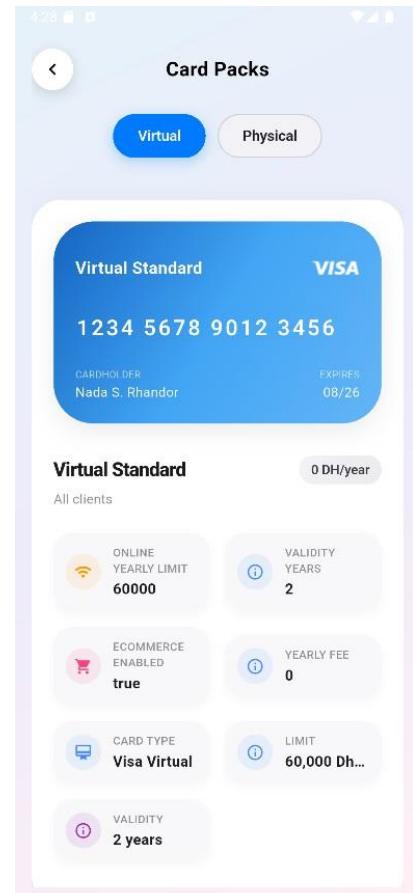


Figure 41 :Cards Pack

19. Reset Password

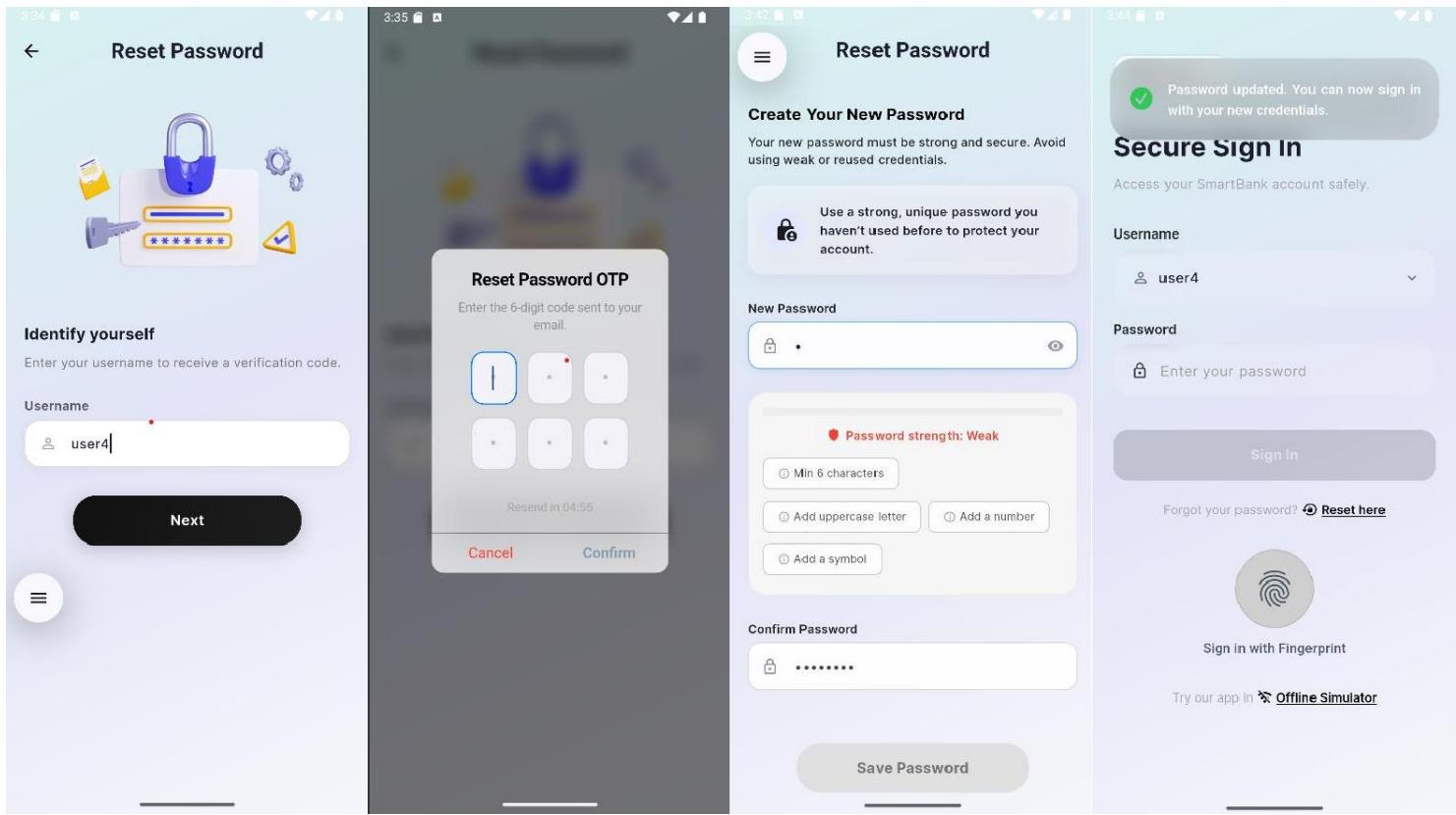


Figure 42 :Reset Password

Ce screen illustre le **processus complet de réinitialisation du mot de passe** en 4 étapes :

- **Identification** : l'utilisateur saisit son nom d'utilisateur pour lancer la vérification.
- **Code OTP** : un code à 6 chiffres est demandé pour confirmer l'identité.
- **Création du nouveau mot de passe** : l'utilisateur entre un nouveau mot de passe avec un indicateur de sécurité.
- **Connexion sécurisée** : l'utilisateur peut ensuite se connecter avec son mot de passe mis à jour ou via l'empreinte digitale.

20. Chatbot

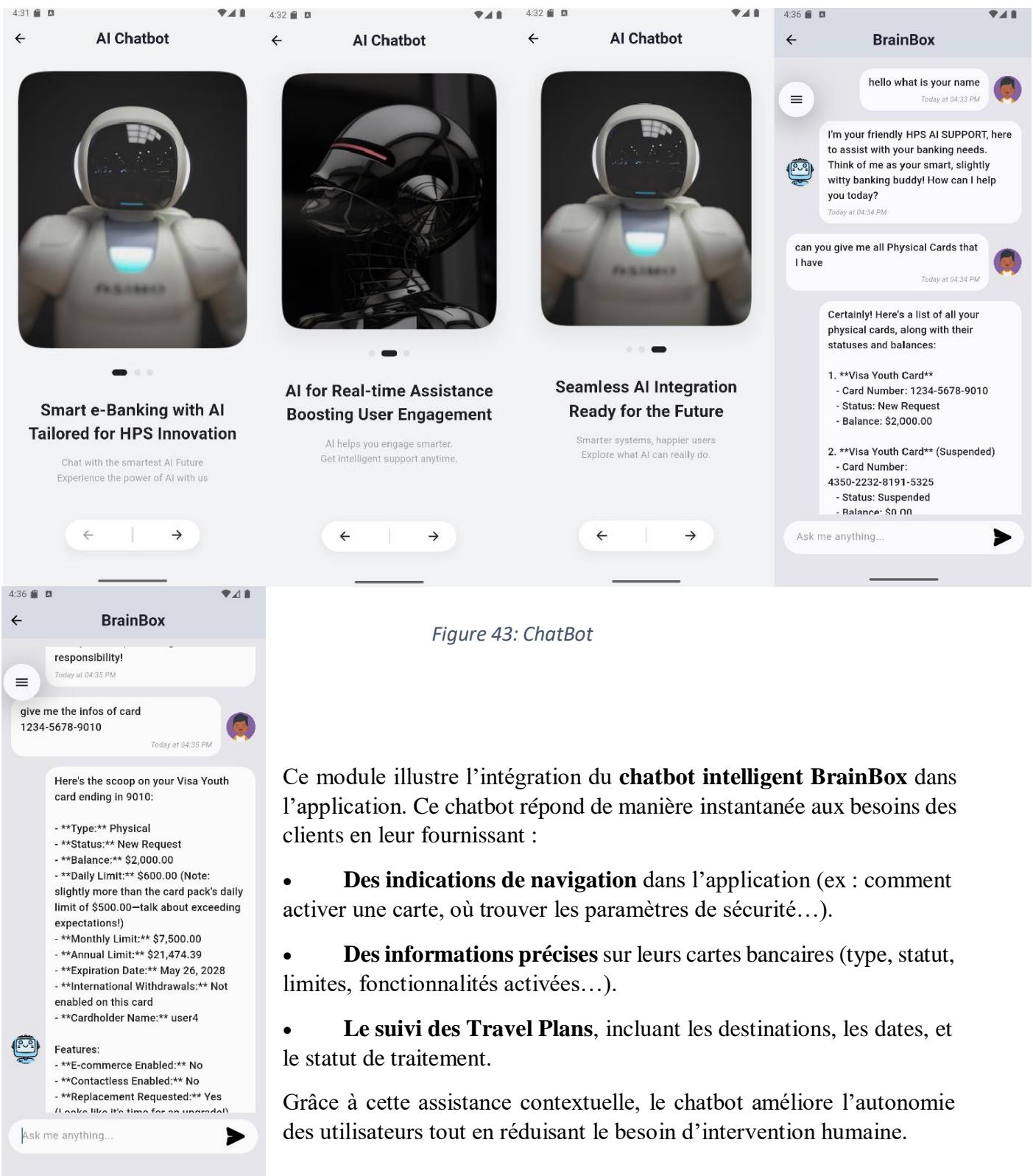


Figure 43: ChatBot

Ce module illustre l'intégration du **chatbot intelligent BrainBox** dans l'application. Ce chatbot répond de manière instantanée aux besoins des clients en leur fournissant :

- Des indications de navigation** dans l'application (ex : comment activer une carte, où trouver les paramètres de sécurité...).
- Des informations précises** sur leurs cartes bancaires (type, statut, limites, fonctionnalités activées...).
- Le suivi des Travel Plans**, incluant les destinations, les dates, et le statut de traitement.

Grâce à cette assistance contextuelle, le chatbot améliore l'autonomie des utilisateurs tout en réduisant le besoin d'intervention humaine.

III. Réalisation de l'application Web – Espace Agent

1. Logging

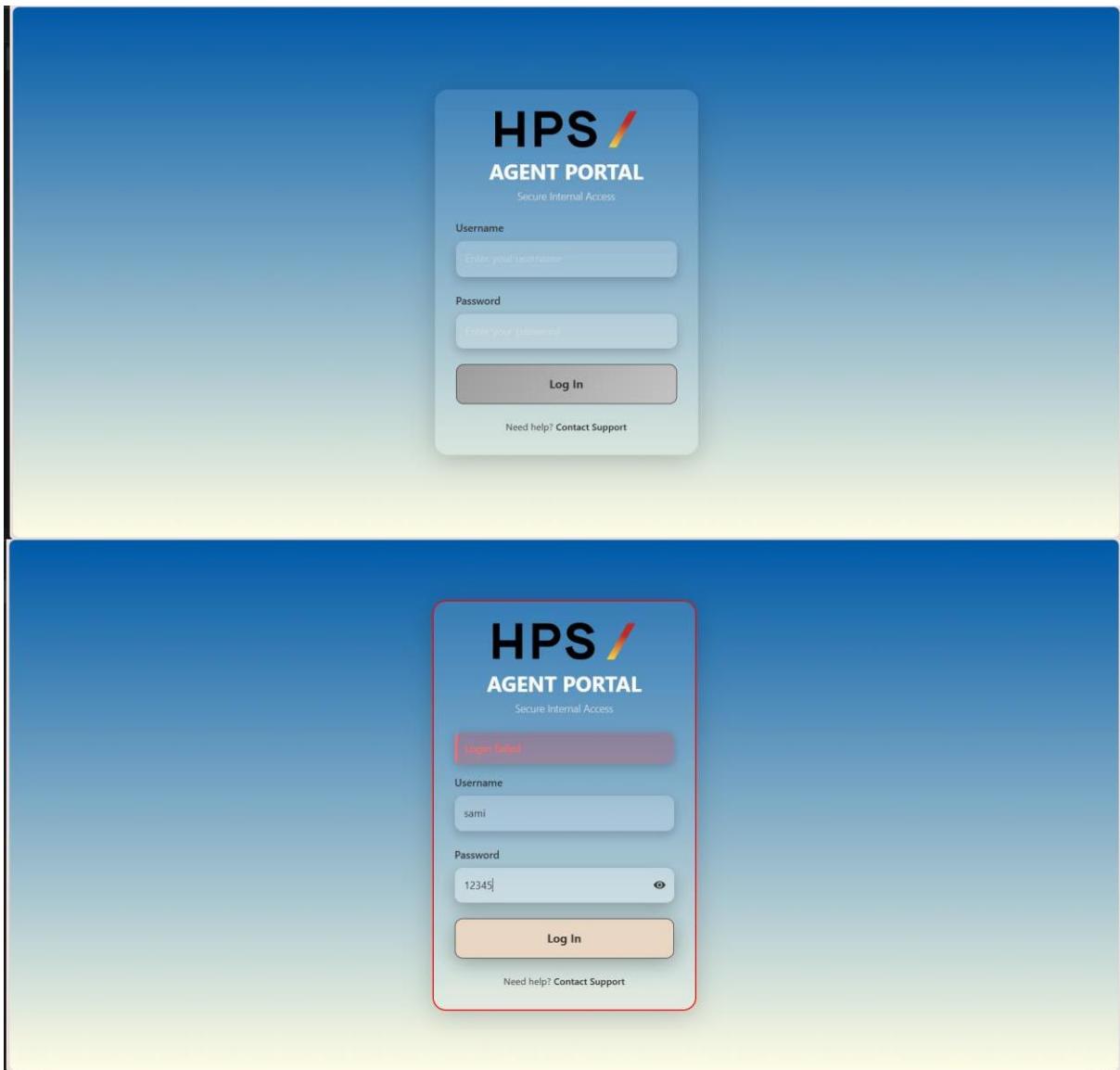


Figure 44 :Logging Agent

Cette interface permet aux **agents HPS** de se connecter à leur portail sécurisé à l'aide de leur **nom d'utilisateur** et **mot de passe**. En cas d'identifiants incorrects, un message "**Login failed**" s'affiche pour signaler l'échec de la connexion.

2. Gestion des Cartes

CARD NUMBER	CARDHOLDER	TYPE	STATUS	EXPIRATION DATE	REPLACEMENT CARD	IS CANCELED	PACK	ACTIONS
1234-5678-9011	user4	PHYSICAL	ACTIVE	2028-05-26	NO	NO	Visa Classic	<button>View</button>
1234-5678-9012	user4	PHYSICAL	ACTIVE	2028-05-26	NO	NO	Visa Gold	<button>View</button>
1234-5678-9013	user4	PHYSICAL	ACTIVE	2028-05-26	NO	NO	Visa Business	<button>View</button>
1234-5678-9014	user4	PHYSICAL	ACTIVE	2028-05-26	NO	NO	Visa Premium+	<button>View</button>
4167-2408-2677-1598	user4	VIRTUAL	SUSPENDED	2028-05-26	NO	YES	Virtual Standard	<button>View</button>

Figure 45:All Cards

Cette interface représente l'**espace de gestion des cartes** destiné aux agents HPS. Elle offre une **vue d'ensemble sur l'état des cartes** (actives, bloquées, suspendues...), tout en permettant :

- La **consultation des détails** de chaque carte (type, statut, date d'expiration, pack associé...)
- L'usage de **filtres dynamiques** pour rechercher par type, statut ou pack
- L'ajout d'une nouvelle carte via le bouton "**Add New Card**"
- Le suivi des demandes de remplacement et des cartes annulées

L'interface est conçue pour une **navigation fluide et rapide**, avec des **indicateurs visuels** clairs pour le statut de chaque carte.

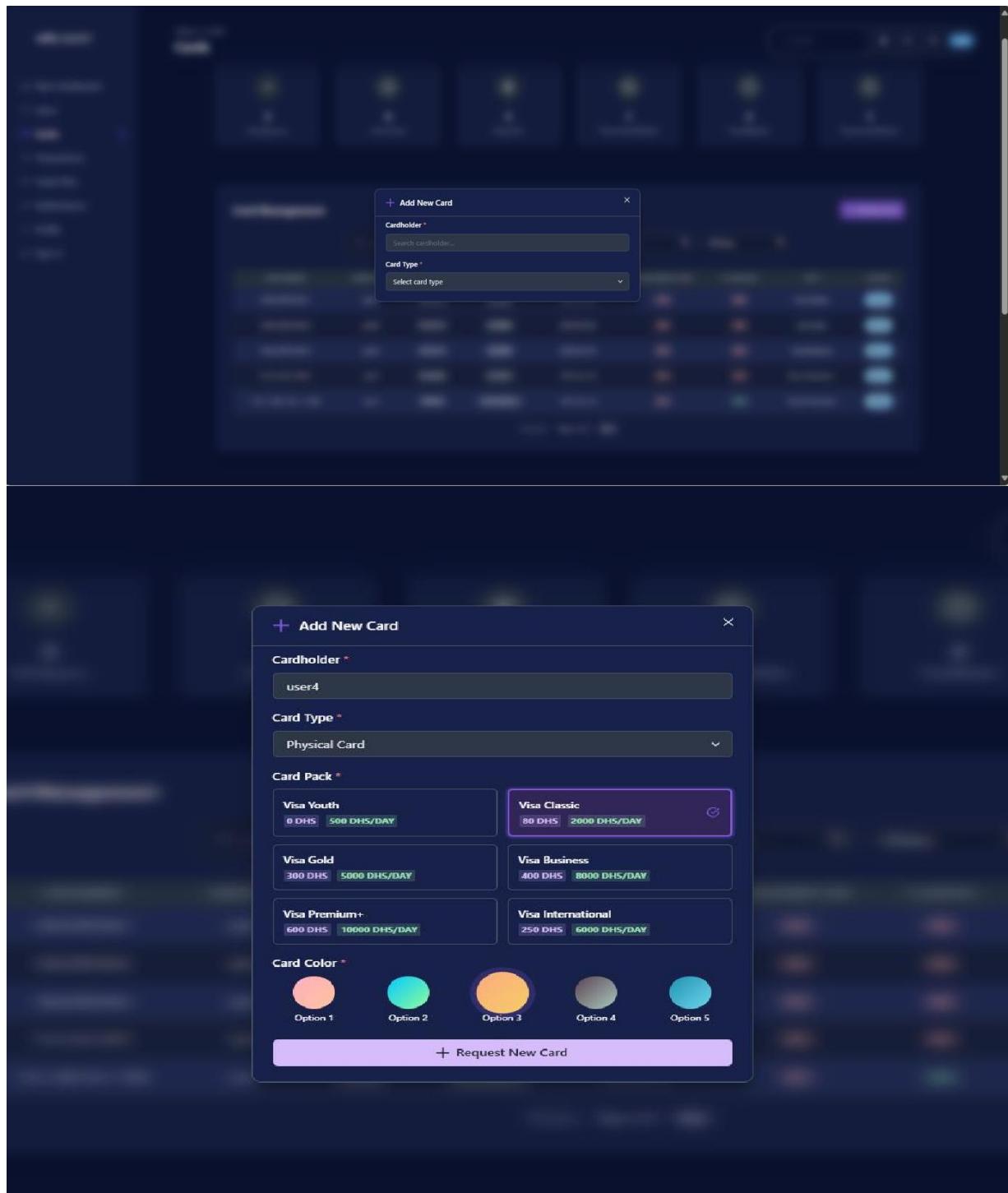


Figure 46 :Add New Card

L'interface d'ajout de carte permet à l'agent de sélectionner un **porteur de carte**, le **type de carte** (physique ou virtuelle), puis de choisir un **pack** parmi plusieurs options (Visa Youth, Classic, Gold, Business, Premium+, International). Chaque pack affiche les frais et les plafonds journaliers. L'agent doit également sélectionner une **couleur de carte** avant de finaliser la demande via le bouton "**Request New Card**". Cette interface garantit une création claire, rapide et personnalisée selon les besoins du client.

3. Physical Card Details

The screenshots illustrate the 'Physical Card Details' section of the HPS Agent interface. The top screenshot provides a high-level overview of the card's status, expiration, and associated pack details. The bottom screenshot delves deeper into specific features like contactless payment and e-commerce, along with real-time usage progress bars for daily, monthly, and annual limits.

Card Details:

- CARD NUMBER: 1234-5678-9011
- CARDHOLDER: user4
- CARD TYPE: PHYSICAL
- STATUS: ACTIVE
- BLOCK REASON: (empty)
- EXPIRATION: 2028-05-26
- INT'L WITHDRAW: YES
- CANCELED: NO
- REPLACEMENT REQUESTED: NO
- BALANCE: 8000 DH
- ACTIVE TRAVEL PLAN: YES
- CVV REQUESTED: YES

Pack Details:

- LABEL: Visa Classic
- AUDIENCE: All adult clients
- FEE: 80 DH
- VALIDITY YEARS: 3
- LIMIT ANNUAL: 96000 DH
- LIMIT DAILY: 2000 DH
- LIMIT MONTHLY: 8000 DH
- INTERNATIONAL WITHDRAW: true
- MAX COUNTRIES: 5
- MAX DAYS: 90
- TYPE: Physical
- INTERNATIONAL WITHDRAW LIMIT PER TRAVEL: 15000 DH

Card Features:

- Contactless: Enabled
- E Commerce: Enabled
- TPE Enabled: Enabled
- International Withdraw: Enabled

Card Limits:

- Daily Limit: 2000.00 DH of 1000.00 DH (Over Limit!)
- Monthly Limit: 8000.00 DH of 10000.00 DH
- Annual Limit: 96000.00 DH of 50000.00 DH (Over Limit!)

Figure 47: Physical Card Details

L'écran **Physical Card Details** offre une vue complète et intuitive sur une carte physique donnée. Il regroupe toutes les informations essentielles : numéro, statut, type, titulaire, solde, plan de voyage actif, et motif de blocage s'il y en a. On y retrouve également les détails du **pack associé** (nom, frais, durée de validité, plafonds journalier/mensuel/annuel, limites de retrait international, etc.).

Les agents peuvent effectuer des actions comme **bloquer ou annuler la carte**, **générer un CVV ou un code PIN**, ou encore **activer/désactiver des fonctionnalités** comme le paiement sans contact, le e-commerce, ou le retrait à l'étranger.

Un encadré à droite affiche les **plafonds de la carte**, avec des barres de progression illustrant leur état d'utilisation en temps réel.

4. Cancel and Uncancel Card

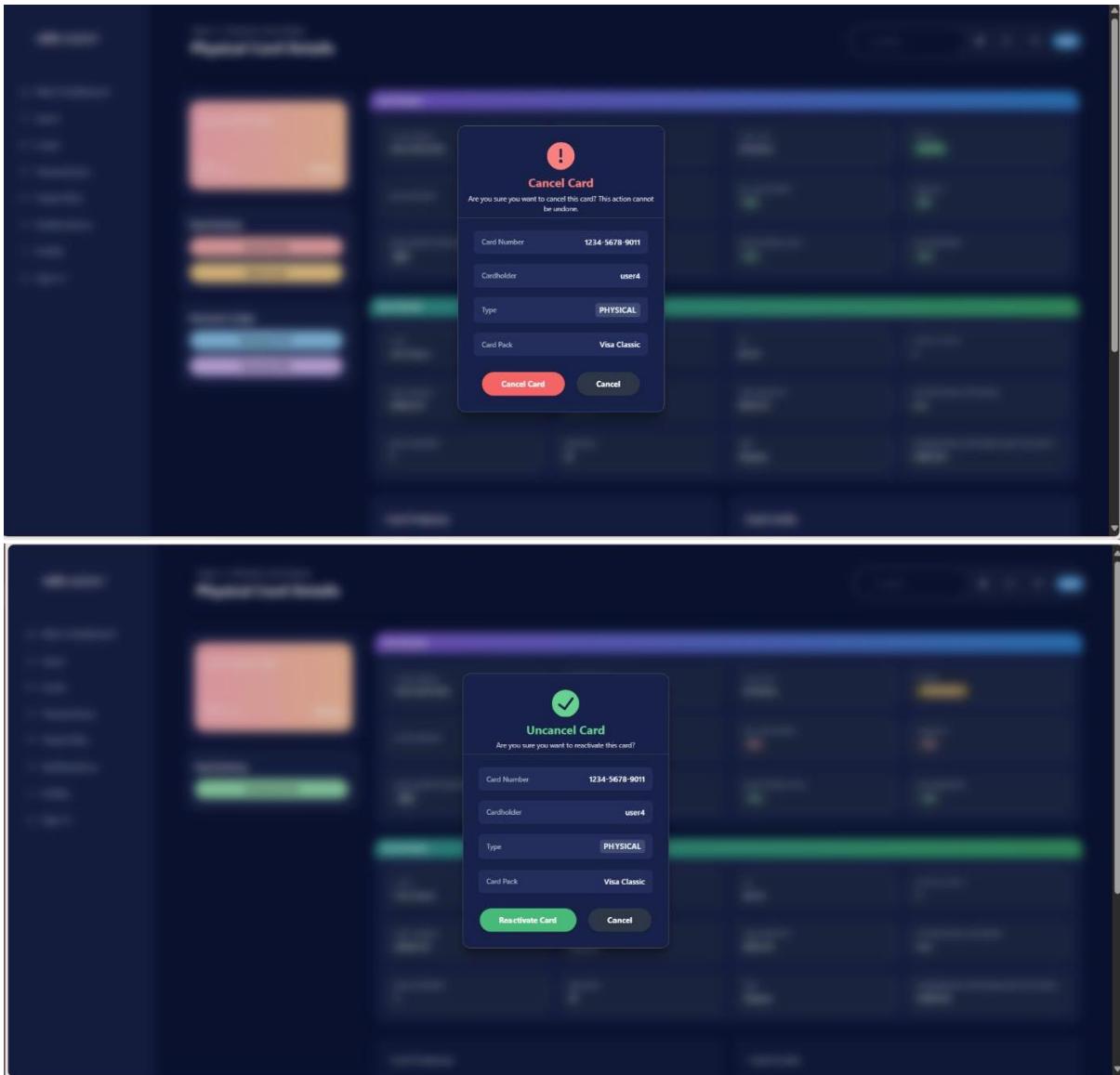


Figure 48: Cancel and Uncancel Card

La plateforme permet à l'agent d'annuler une carte via une interface de confirmation claire. Lors de l'annulation (Cancel Card), une fenêtre s'affiche avec un message d'alerte avertissant que cette action est irréversible. L'agent peut consulter les informations clés de la carte (numéro, type, pack...) avant de valider.

En cas de besoin, une carte annulée peut également être réactivée (Uncancel Card). Une fenêtre de confirmation similaire s'affiche, récapitulant les mêmes informations, et permet à l'agent de restaurer l'état actif de la carte via l'action **Reactivate Card**

5. Block et Unblock Card

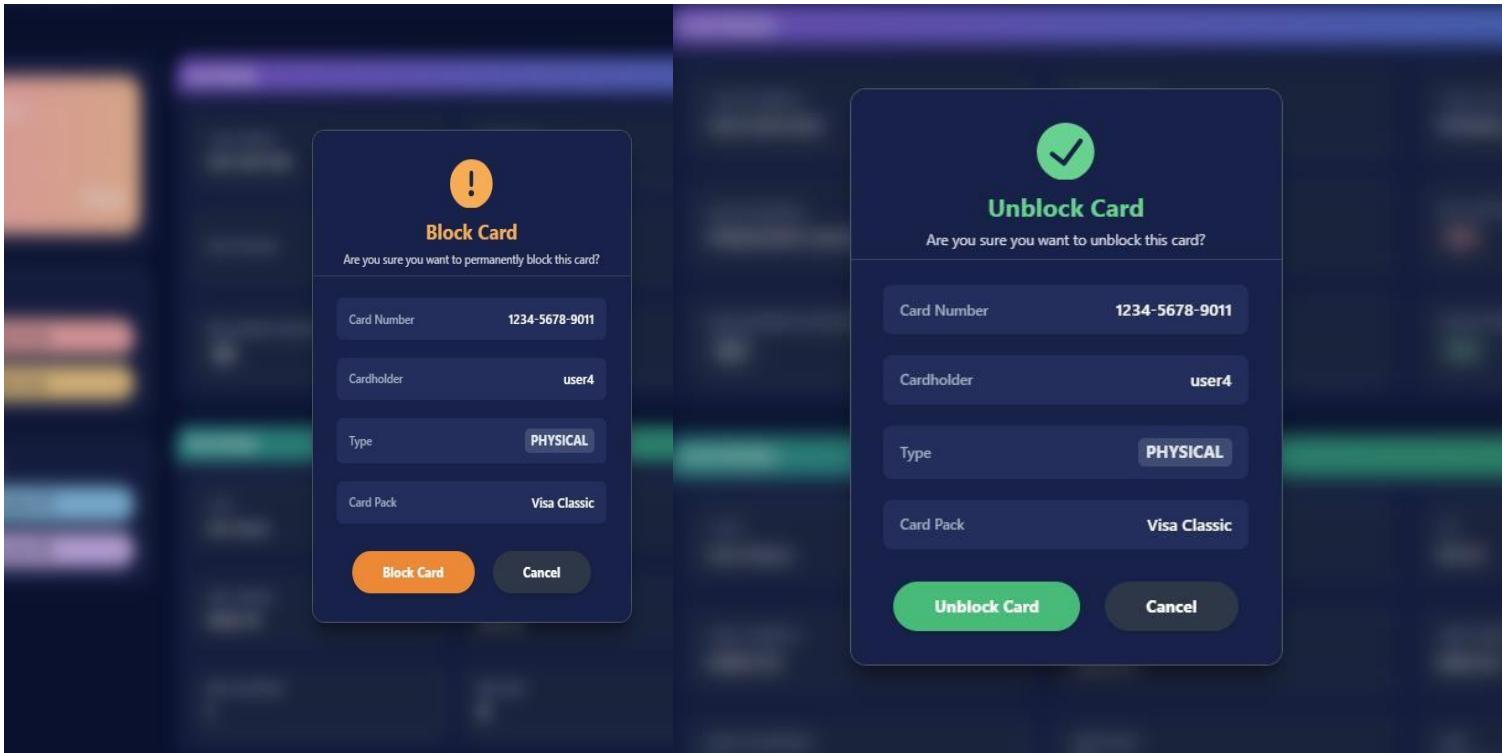


Figure 49 : Block / Unblock Cards

L’agent peut temporairement **bloquer une carte** via un bouton dédié. Cette action est utile en cas de suspicion de fraude, de perte provisoire ou pour prévenir une utilisation non autorisée. Une fois bloquée, la carte devient inactive, mais reste réversible.

La carte peut être **débloquée** si la situation est rétablie. L’interface affiche alors un bouton **Unblock Card** permettant à l’agent de restaurer instantanément les fonctionnalités de la carte.

Ces actions sont conçues pour être rapides, sécurisées et visibles dans l’écran de détails de la carte.

6. Generate PIN and CVV

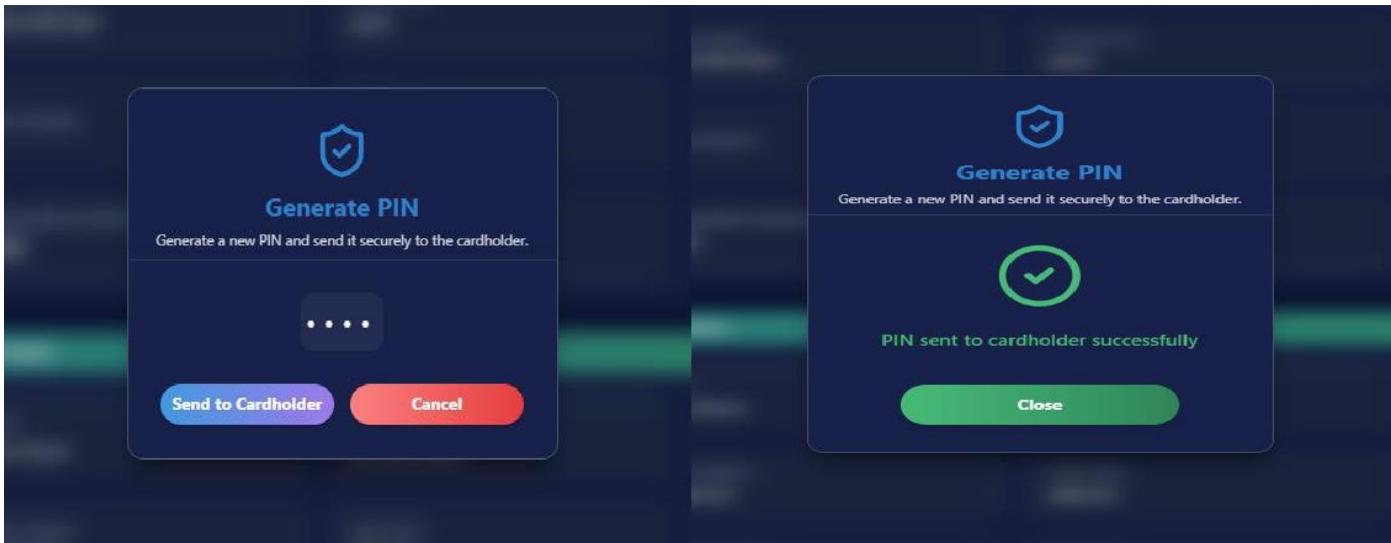


Figure 50: Generate PIN et CVV

Les agents autorisés peuvent générer de nouveaux codes **PIN** ou **CVV** via l'interface d'administration.

Une fois générés, ces codes sont automatiquement **envoyés de manière sécurisée au porteur de carte** via le canal défini (ex. : SMS ou e-mail).

Cette opération est strictement réservée aux agents habilités afin de garantir la **confidentialité et la sécurité des données sensibles** du client.

7. Virtual Card Details

The screenshot shows the 'Virtual Card Details' page within the HPS Agent application. On the left is a sidebar with navigation links: Main Dashboard, Users, Cards, Transactions, Travel Plan, Notifications, Profile, and Sign In. The main content area has a header 'Virtual Card Details'. It displays a large card thumbnail with the number '4002-6602-9201-3327', the expiration date '2028-05-26', and the brand 'VISA'. Below the thumbnail are sections for 'Card Details' (Card Number: 4002-6602-9201-3327, Cardholder: user4, Card Type: VIRTUAL, Status: ACTIVE), 'Pack Details' (Label: Virtual Plus, Audience: Active clients, Fee: 50 DH, Validity Years: 3, Annual: 120000 DH), 'Card Features' (E-Commerce: Enabled), and 'E-Commerce Limit per Year' (Annual Limit: 99578.30 DH of 120000.00 DH, 65% used). A search bar and a 'New' button are at the top right.

Figure 51 : Virtual Card Details

L'agent peut voir tous les détails de la carte (statut, pack, solde, etc.) et effectuer deux actions :

- **Cancel Card**
- **Block Card**

Il peut aussi gérer les **services activés** (comme E-Commerce) et consulter la **consommation du plafond annuel**.

8. Cancel and Uncancel Virtual Card

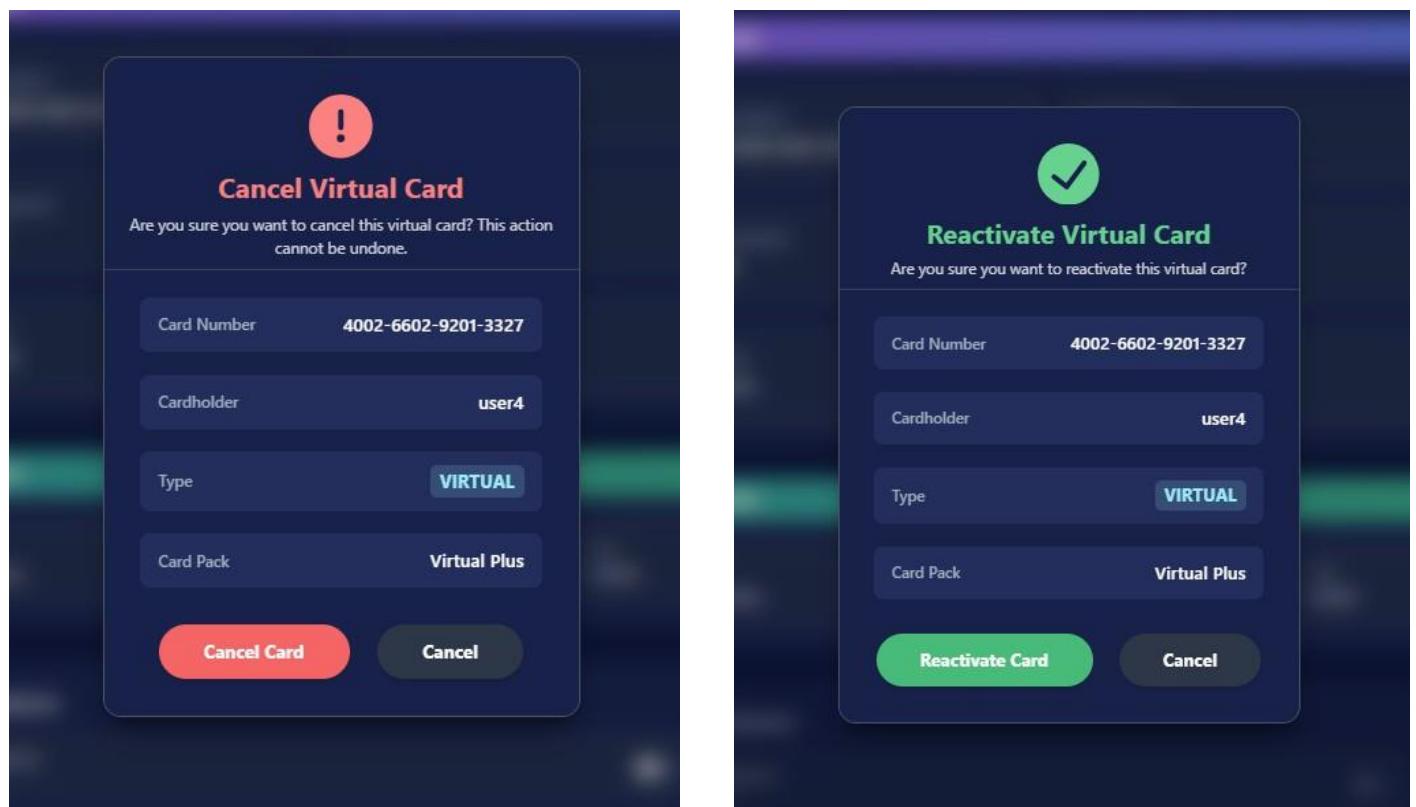


Figure 52: Cancel /Uncancel Virtual Card

L’agent peut annuler une carte virtuelle (action irréversible) et, si nécessaire, la réactiver via une interface claire affichant les détails essentiels : numéro, titulaire, type et pack associé.

9. Block et Unblock Virtual Card

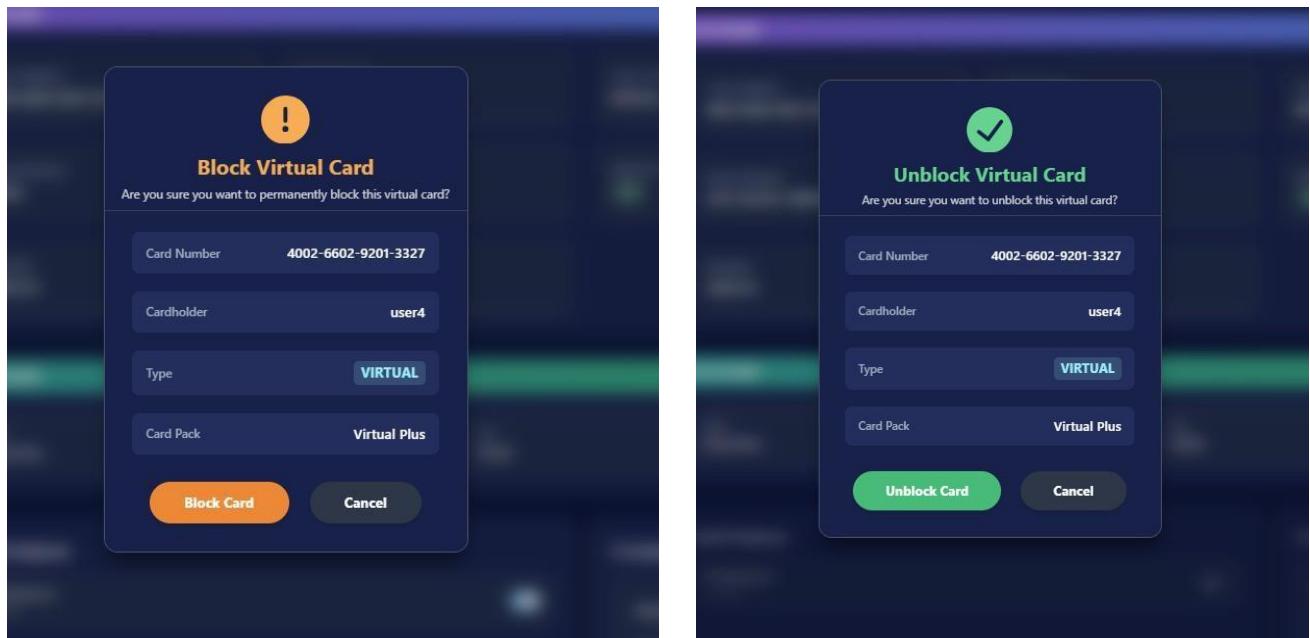


Figure 53 :Block / Unblock Virtual Card

Ces écrans illustrent le blocage et le déblocage d'une carte virtuelle.

10. Transaction

Transactions								Export to Excel
Search by merchant, description, category, card number, or								
DATE	MERCHANT	AMOUNT	CATEGORY	DESCRIPTION	STATUS	CARD NUMBER	CARDHOLDER USERNAME	ACTION
2025/06/02 19:20	Uber	\$14.50	TRANSPORT	Customer completed a ride with Uber during this billing period	PENDING	1234-5678-9010	user4	More Details
2025/06/03 14:05	Carrefour Market	\$67.30	GROCERIES	Weekly groceries successfully processed	COMPLETED	1234-5678-9010	user4	More Details
2025/06/04 22:40	Netflix	\$13.99	SUBSCRIPTION	Netflix subscription billed for the current month	PENDING	1234-5678-9010	user4	More Details
2025/06/05 10:30	ONSS	\$42.00	INSURANCE	Monthly health insurance contribution to CNSS	COMPLETED	1234-5678-9010	user4	More Details
2025/06/06 16:55	Airbnb	\$89.90	TRAVEL	Booking processed for Airbnb stay	PENDING	1234-5678-9010	user4	More Details

Figure 54:Transactions

Cet écran permet à l'agent de consulter l'historique des transactions par carte, avec des filtres par catégorie, statut et la possibilité d'exporter vers Excel.

11. Transaction Details

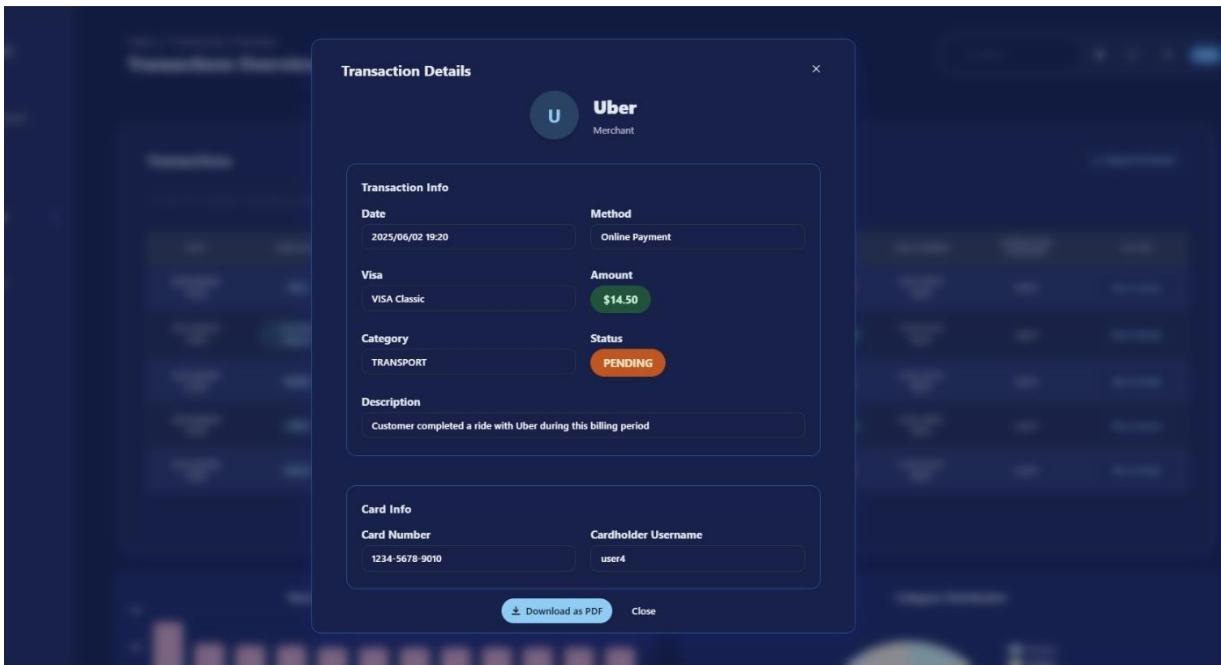


Figure 55 :Transaction Details

Cette interface affiche les détails complets d'une transaction : date, méthode de paiement, montant, statut, description, et informations carte. Un bouton permet de télécharger le reçu en PDF.

12. Analyse des Transactions Bancaires par Catégorie, Revenu et Précision du Modèle



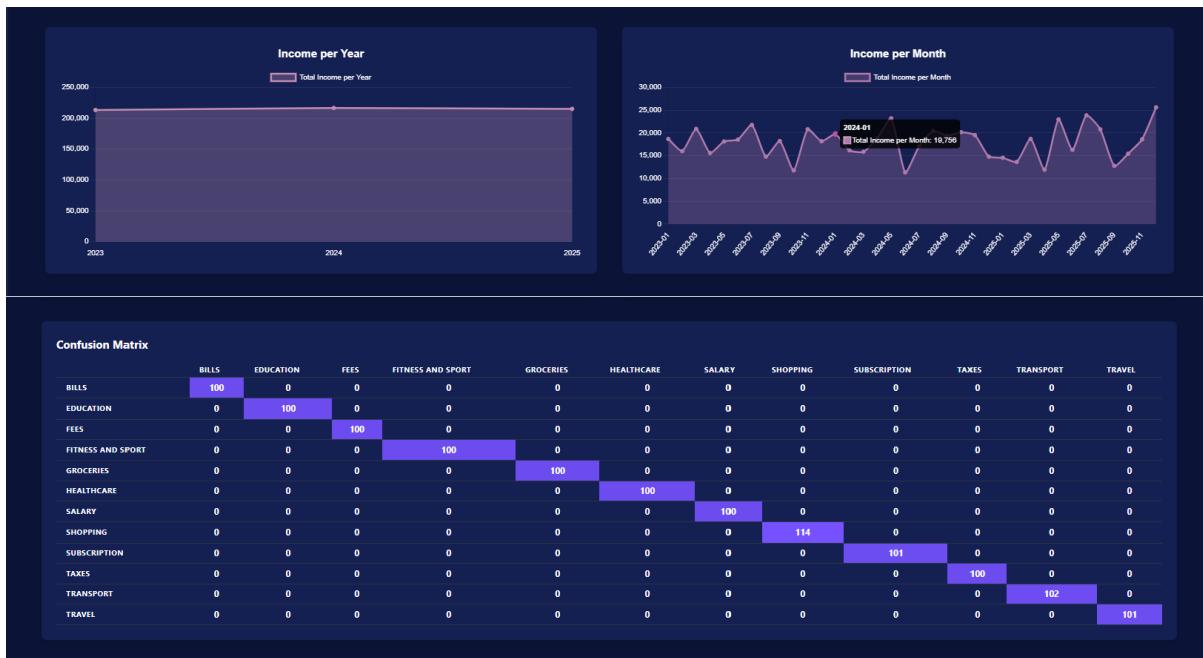
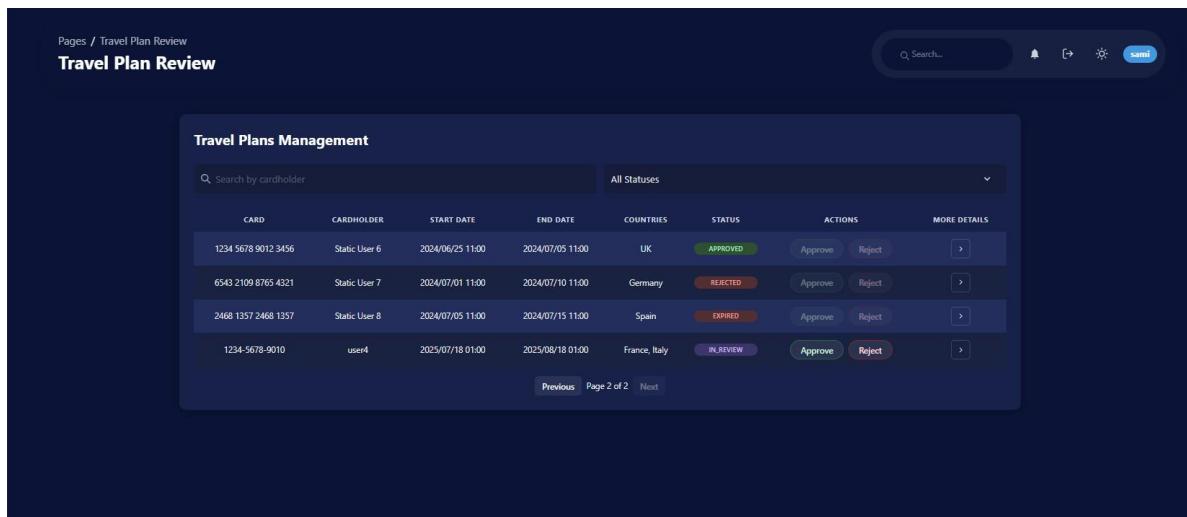


Figure 56: Analyse des Transactions Bancaires par Catégorie, Revenu et Précision du Modèle

Ces tableaux et graphiques offrent une vue d'ensemble complète des transactions effectuées, réparties par catégorie, mois et année. Le graphique en barres montre le volume par catégorie (ex. : shopping, transport), tandis que le graphique circulaire illustre leur répartition globale. L'évolution des revenus est suivie à l'année et au mois. Enfin, la matrice de confusion valide la précision du modèle de classification automatique des transactions par catégorie.

13. Gestion des Travel Plan



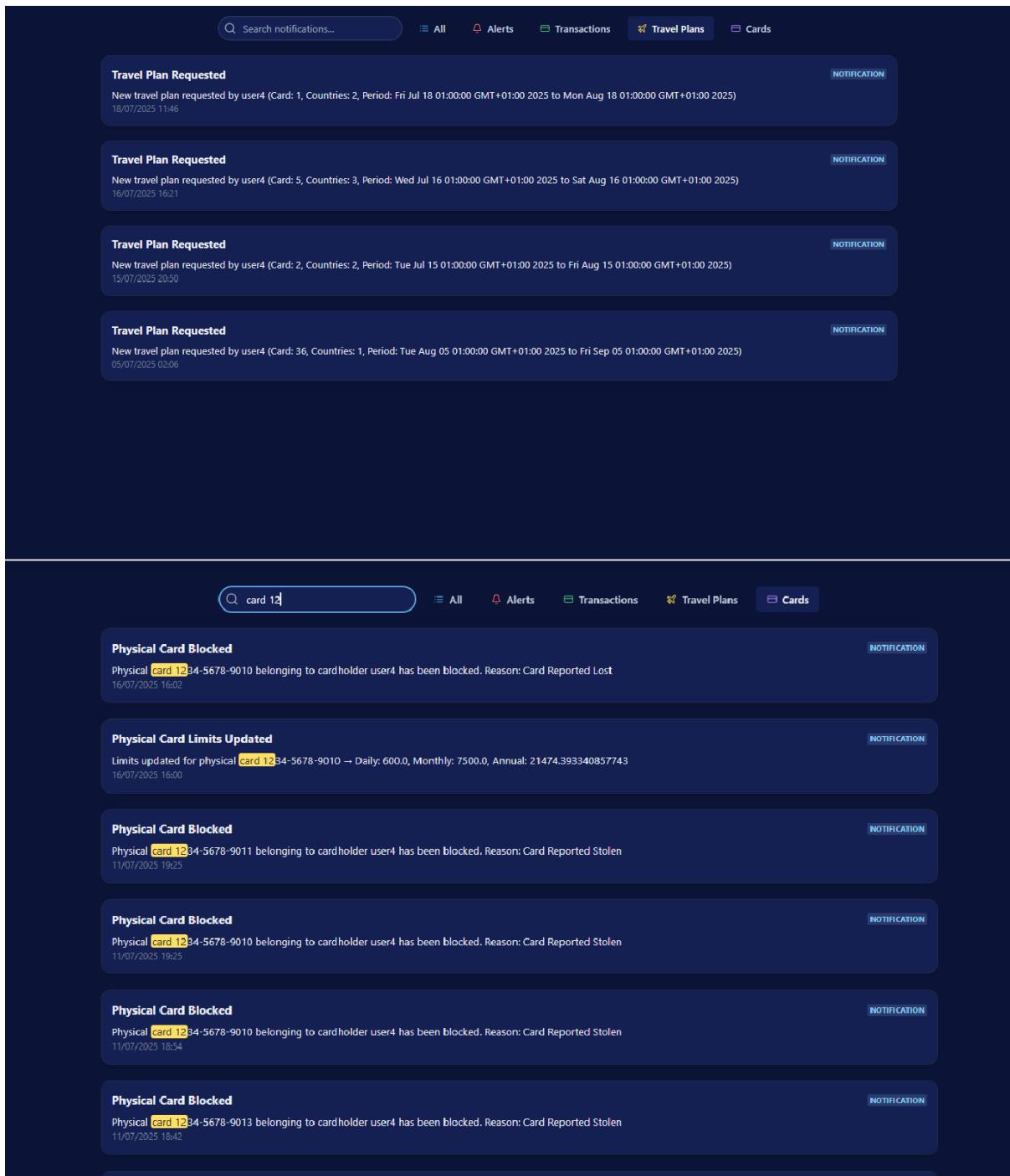


Figure 57: Gestion des Travel Plan

Lorsqu'un utilisateur bloque sa carte virtuelle, il choisit un motif précis comme « Carte volée » ou « Détails compromis ». Une fois l'action validée, la carte devient immédiatement inutilisable pour les paiements en ligne. En parallèle, une notification est générée automatiquement dans l'onglet dédié aux cartes, précisant le numéro de la carte, le nom du titulaire, la date et la raison du blocage. Si le blocage est temporaire, l'utilisateur a la possibilité de réactiver la carte. Dans ce cas, une nouvelle notification est émise pour confirmer la réactivation, et la carte redevient

instantanément fonctionnelle. Toutes les étapes sont donc traçables à travers le système de notifications, permettant à l'utilisateur comme à l'agent de suivre précisément l'historique des actions effectuées sur la carte.

14. Gestion des Utilisateurs – Interface Agent

The screenshot displays the HPS Agent interface, specifically the 'Users' section. The left sidebar includes links for Main Dashboard, Users (selected), Cards, Transactions, Travel Plan, Notifications, Profile, and Sign In. The main area shows key statistics: Total Users (250), New Users (15), Top Users (200), and Other Users (35). Below this is a table titled 'All Cardholders' with columns for Username, Email, Locked status, First Login, Attempts, Biometric, and Actions. The table lists several users including user4, nada, hamid, manal, static_user1, and static_user2. At the bottom of the page, there are three modals: one for user4 showing recent activities and card totals; another for 'Personal Information' with fields for Username, Email address, and Password; and a third for adding a new user.

USERNAME	EMAIL	LOCKED	FIRST LOGIN	ATTEMPTS	BIOMETRIC	ACTIONS
user4	rhalimsami8@gmail.com	ACTIVE	NO	0	ENABLED	View
nada	nadaechehaj003@gmail.com	ACTIVE	YES	0	ENABLED	View
hamid	hamid@gmail.com	ACTIVE	YES	0	ENABLED	View
manal	manal@gmail.com	ACTIVE	YES	0	ENABLED	View
static_user1	static1@example.com	LOCKED	NO	5	DISABLED	View
static_user2	static2@example.com	LOCKED	NO	3	DISABLED	View

Figure 58: Gestion des Utilisateurs

L'interface agent permet de visualiser et gérer les utilisateurs : statut, tentatives, biométrie, etc. Un bouton permet d'accéder aux détails de chaque utilisateur.

La fiche détaillée affiche les activités récentes et le nombre de cartes (physiques et virtuelles). L'agent peut aussi suspendre un utilisateur.

L'ajout d'un utilisateur se fait via un formulaire simple avec génération de mot de passe possible.

15. Notification– Interface Agent

The screenshot displays two side-by-side views of the HPS Agent application. On the left, a modal window titled 'New Card Issued Notification' shows a message to 'user4' confirming a new physical card has been issued. It includes a 'Card Details' section and 'Important Next Steps'. On the right, the main 'Notifications' page lists several recent events: 'Travel Plan Requested', 'Security Alert' (password change), 'Security Alert' (security code setup), 'CVV Requested', and another 'Travel Plan Requested'. Each notification includes a timestamp and a 'NOTIFICATION' link. The interface has a dark theme with blue and white text.

Figure 59 :Notifications – Suivi en temps réel

L'interface regroupe toutes les notifications importantes liées aux actions des porteurs de carte : demandes de CVV, changements de mot de passe, plans de voyage, alertes de sécurité... Les agents peuvent facilement filtrer les types d'alertes et consulter les détails pour chaque événement, avec date, utilisateur, et type d'action.

Conclusion

Ce chapitre a permis d'illustrer de manière concrète le travail accompli à travers les interfaces mobiles et web développées pour les titulaires de cartes et les agents. Chaque fonctionnalité a été pensée pour offrir une expérience fluide, sécurisée et intuitive, tout en respectant les standards de la banque digitale moderne. Les captures d'écran présentées témoignent de la richesse fonctionnelle de la plateforme et de sa capacité à répondre aux besoins des utilisateurs en matière de gestion des cartes, des transactions et des services associés.

Conclusion Générale

Notre plateforme e-Banking répond pleinement aux exigences d'une gestion moderne, intelligente et sécurisée des cartes bancaires. Grâce à une architecture microservices bien structurée, intégrant des technologies avancées telles que Spring Boot, Kafka, PostgreSQL vectoriel, Redis, et un chatbot basé sur GPT-4o et RAG, le système offre une expérience fluide aussi bien pour les agents que pour les titulaires de cartes.

Ce projet a été pour nous une expérience formatrice qui nous a permis de mettre en œuvre nos acquis académiques (UML, architecture logicielle, microservices, sécurité OAuth2/JWT...) dans un contexte professionnel réel. Nous avons également appris à travailler en équipe dans un environnement agile, à documenter notre travail, et à résoudre des problématiques complexes en collaboration avec les différents métiers.

L'intégration avec la solution PowerCARD via IBM WebSphere a constitué un véritable défi technique que nous avons su relever, nous rapprochant ainsi des standards des solutions bancaires internationales.

Enfin, bien que la version actuelle soit une première release fonctionnelle, nous envisageons d'enrichir la plateforme avec de nouvelles fonctionnalités telles que l'analyse prédictive des transactions, une gestion étendue des cartes virtuelles et des tableaux de bord analytiques en temps réel.

Perspectives

La plateforme développée durant ce projet repose sur une architecture microservices moderne, sécurisée et évolutive. Elle intègre des composants avancés tels que l'intelligence artificielle (GPT-4o combinée à une approche RAG), un système d'événements asynchrones basé sur Kafka, un assistant virtuel intelligent, ainsi qu'une interconnexion avec la plateforme bancaire PowerCARD via IBM WebSphere. Un premier niveau d'optimisation des performances a également été mis en place grâce à Redis Cache. À ce stade, Redis a été intégré spécifiquement au *Card-Service*, considéré comme l'un des microservices les plus sollicités en raison du volume important de requêtes liées à la gestion des cartes bancaires.

Dans cette continuité, plusieurs pistes d'amélioration ont été identifiées afin d'enrichir les fonctionnalités de la plateforme, renforcer sa robustesse et répondre à des cas d'usage plus avancés.

Tout d'abord, l'une des évolutions majeures envisagées concerne *l'extension de Redis Cache à l'ensemble des microservices*. Cette généralisation permettrait de réduire significativement les temps de réponse, limiter les accès directs aux bases de données relationnelles, et améliorer la fluidité de l'expérience utilisateur, notamment lors des pics de charge.

Par ailleurs, bien que le projet actuel ait déjà permis d'*intégrer les APIs Visa* à travers des simulations et des tests d'authentification et de transactions, l'objectif est désormais de *connecter également les APIs MasterCard* exposées par la plateforme PowerCARD. Cette double interconnexion offrirait une couverture plus large des réseaux de paiement et renforcerait l'interopérabilité avec les systèmes bancaires partenaires.

Enfin, une évolution fonctionnelle clé porte sur l'*élargissement du rôle du cardholder*. Actuellement limité à la visualisation et à la gestion de ses cartes (plafonds, blocages, préférences, plans de voyage), le client devrait prochainement être en mesure **d'effectuer lui-même des transactions bancaires* directement depuis son application mobile ou web, avec un support complet côté backend.

Ces perspectives s'inscrivent dans une logique d'évolution progressive vers un *écosystème bancaire intelligent, interactif et entièrement transactionnel*, tout en conservant les fondements d'une architecture ouverte, sécurisée et alignée avec les standards du secteur.

Webographie

<https://chat.openai.com/>

<https://stackoverflow.com/questions/>

<https://www.figma.com/files/team/1483190951552101191/recents-and-sharing?fuid=1483190949345011186/>

<https://redis.io/docs/latest/>

<https://www.fluidtopics.com/fr/hps-dita-technical-documentation-publishing-case-study/>

<https://redux.js.org/>

<https://start.spring.io/>