



Unité de formation et de Recherche en Sciences  
Exactes et Appliquées (UFR/SEA)

Département : Informatique

Année Académique 2019/2020

UJKZ/SEA/Dept-Info/LSIR/2023



UNIPAY SAS

Startups exerçant dans le E-commerce ,  
la production de café burkinabé, et la  
conception de solutions écologiques.

## RAPPORT DE STAGE DE FIN DE CYCLE



### THEME : Refonte et conception du mode de paiement de UniPay SAS



Pour l'obtention de la **Licence Informatique**,  
option : *Systèmes d'Information et Réseaux*  
(SIR)

Stage effectué par :

**OUEDRAOGO Fernand**

**Période de stage : 01 mars 2023 – 02 juin 2023**

**Lieu de stage : UniPay SAS**

**Superviseur : Mr Gouayon KOALA** enseignant  
à l'université JOSEPH KI ZERBO

**Maitre de stage : Dr Faical CONGO**  
PDG de UniPay SAS

## **DEDICACE**

Ce document est dédié à :

- ✓ mes parents ;
- ✓ mes enseignants ;
- ✓ mes promotionnaires de l’UFR/SEA ;
- ✓ mes amis et amies qui m’ont soutenu et aidé.

# Refonte et conception du mode de paiement de UniPay SAS

---

## REMERCIEMENTS

Au terme de ce travail, nous saisissons cette occasion pour présenter nos remerciements à tous ceux qui ont participé à la réalisation de ce travail.

Tout d'abord, nous adressons nos remerciements à notre superviseur de stage, **Gouayon KOALA** qui nous a beaucoup aidé pour la rédaction de ce document.

Nous remercions vivement **Dr Congo Faical** mon maître de Stage, PDG de UniPay SAS, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Il fut d'une aide précieuse dans les moments les plus délicats.

Nous remercions également **Dr Didier BASSOLE** pour son aide et assistance précieuse dans l'obtention des ressources nécessaires pour la rédaction de ce document.

Enfin, nous remercions toute l'équipe pédagogique de l'UFR-SEA et intervenants professionnels responsables de la formation pour les enseignements reçus.

## AVANT-PROPOS

L'Université Joseph KI-ZERBO, la première Université publique au Burkina Faso, créée en 1974 est un établissement à caractère scientifique, culturel et technique. Elle se compose :

- ✓ de cinq (05) Unités de Formation et de Recherche (UFR) : Sciences Exactes et Appliquées (UFR/SEA), Sciences de la Vie et de la Terre (UFR/SVT), Sciences de la Santé (UFR/SDS), Lettres, Arts et Communication (UFR/LAC) et Sciences Humaines (UFR/SH) ;
- ✓ de cinq (05) instituts : l'Institut Burkinabè des Arts et Métiers (IBAM), l'Institut Supérieur des Sciences de la Population (ISSP), l'Institut Panafricain d'Etude et de Recherche sur les Médias, l'Information et la Communication (IPERMIC), ISSDH (Institut des Sciences du Sport et du développement Humain), IGEDD (Institut du Génie de L'Environnement et du Développement Durable) ;
- ✓ de deux (02) Centres Universitaires : le Centre Universitaire de Kaya et le Centre Universitaire de Ziniaré ;
- ✓ de trois écoles doctorales dans les domaines : Lettre, Sciences Humaines et Communication, Sciences de la Santé et Sciences et Technologies.

Chaque Unité de Formation et de Recherche est subdivisée en départements. Ainsi l'UFR/SEA comprend les départements Informatique, Mathématique, Physique et Chimie. Le Département Informatique a été créé en 2013 avec le passage au système Licence Master Doctorat (LMD). Ce département a pour objectif la formation dans les domaines de l'Informatique. Au département Informatique, l'obtention de la Licence à la fin du premier cycle est soumise à un stage d'une durée minimale de trois mois. L'objectif de ce stage est de permettre aux étudiants(es) en fin de cycle de s'imprégner des réalités de la vie professionnelle et de parfaire leurs connaissances théoriques et pratiques acquises au cours des trois années de formation.

# Refonte et conception du mode de paiement de UniPay SAS

---

## SIGLE ET ABBREVIATIONS

|         |  |
|---------|--|
| API     | Application Programming Interface                                    |
| CSS     | Cascading Style Sheet  |
| DOM     | Document Object Model  |
| HTML    | HyperText Markup Language  |
| JS      | JavaScript   |
| MVVM    | Model-View-ViewModel   |
| MVC     | Model View Controller  |
| NoSQL   | Not Only SQL   |
| ORM     | Object-Relational Mapping  |
| PI      | Raspberry PI   |
| SQL     | Structured Query Language  |
| UFR/SEA | Unité de Formation et de Recherche en Sciences Exactes et Appliquées |
| UJKZ    | Université Joseph KI ZERBO   |
| UML     | Unified Modeling Language  |
| USSD    | Unstructured Supplementary Service Data                              |

## INDEX DES FIGURES

|   |    |
|---|----|
| Figure 1 : Description de la méthode Agile SCRUM.....                     | 7  |
| Figure 2 : Planning.....  | 9  |
| Figure 3 : Diagramme de cas d'utilisation.....                            | 15 |
| Figure 4 : Diagramme de séquence ajouter de l'argent dans son wallet..... | 26 |
| Figure 5 : Diagramme de séquence de choisir moyen de paiement.....        | 27 |
| Figure 6 : Diagramme de séquence s'authentifier.....                      | 28 |
| Figure 7 : Diagramme de séquence transférer de l'argent.....              | 28 |
| Figure 8 : Diagramme de séquence générer une clé d'API.....               | 29 |
| Figure 9 : Diagramme de séquence initier un paiement.....                 | 29 |
| Figure 10 : Diagramme de séquence valider un paiement.....                | 30 |
| Figure 11: Diagramme d'activité mettre l'argent dans son wallet.....      | 32 |
| Figure 12 : Diagramme d'activité transférer de l'argent.....              | 33 |
| Figure 13 : Diagramme d'activité initier un paiement.....                 | 34 |
| Figure 14 : Diagramme d'activité valider un paiement.....                 | 34 |
| Figure 15 : Diagramme de classe.....                                      | 36 |
| Figure 16 : Architecture MMVC.....  | 38 |
| Figure 17 : Logo HTML5.....   | 39 |
| Figure 18 : Logo CSS3.....  | 39 |
| Figure 19 : logo javascript.....  | 40 |
| Figure 20 : Logo python.....  | 41 |
| Figure 21 : Logo Insomnia.....  | 48 |
| Figure 22 : Logo Docker.....  | 49 |
| Figure 23 : Logo Diagrams.io.....   | 50 |
| Figure 24 : Logo mongoDB.....   | 50 |
| Figure 27 : modifier une passerelle.....                                  | 52 |
| Figure 28 : Tableau de bord.....  | 52 |
| Figure 29 : Générer API.....  | 53 |
| Figure 30 : Mise a jour et info de l'API entreprise.....                  | 53 |
| Figure 31: Interface du mobile.....                                       | 54 |

# Refonte et conception du mode de paiement de UniPay SAS

---

## INDEX DES TABLEAUX

|  |    |
|--|----|
| Tableau 1 : Ouvrir un compte ohSwap.....             | 17 |
| Tableau 2 : S'authentifier.....                      | 18 |
| Tableau 3 : Initier paiement.....                    | 20 |
| Tableau 4 : Valider un paiement.....                 | 20 |
| Tableau 5 : Transférer de l'argent.....              | 21 |
| Tableau 6 : Ajouter de l'argent dans son wallet..... | 23 |
| Tableau 7 : Générer une clé d'API.....               | 23 |
| Tableau 9 : Coût du matériel de développement.....   | 46 |
| Tableau 10 : Coût du matériel de déploiement.....    | 46 |

## Table des matières

|   |     |
|---|-----|
| DEDICACE.....   | ii  |
| REMERCIEMENTS.....  | iii |
| AVANT-PROPOS.....   | iv  |
| SIGLE ET ABREVIATIONS.....                                  | iv  |
| INDEX DES FIGURES.....                                      | v   |
| INDEX DES TABLEAUX.....                                     | vi  |
| INTRODUCTION.....   | 1   |
| CHAPITRE I : GÉNÉRALITÉS.....                               | 3   |
| I. Historique et Présentation de UniPay SAS.....            | 3   |
| 1. Présentation.....  | 3   |
| 2. Services.....  | 3   |
| II. PRÉSENTATION DU CADRE DU PROJET.....                    | 4   |
| 1. Contexte du stage.....                                   | 4   |
| 2. Problématique.....                                       | 4   |
| 3. Objectif du stage.....                                   | 5   |
| 4. Les membres de l'équipe du projet.....                   | 5   |
| CHAPITRE II : SPECIFICATIONS ET ANALYSE DES BESOINS.....    | 6   |
| I. Démarche méthodologique.....                             | 6   |
| 1. Méthodologie.....  | 6   |
| 2. Langage de modélisation.....                             | 8   |
| 3. Planning.....  | 8   |
| II. Étude de l'existant.....                                | 10  |
| CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME.....        | 11  |
| I. Domaine d'étude et fonctionnalités.....                  | 11  |
| 1. Domaine d'étude.....                                     | 11  |
| 2. Fonctionnalités.....                                     | 11  |
| II. Modélisation.....                                       | 12  |
| 1. Diagramme des cas d'utilisation.....                     | 12  |
| 2. Description textuelle de quelque cas d'utilisation.....  | 16  |
| 3. Diagramme de séquence de quelques cas d'utilisation..... | 25  |



# Refonte et conception du mode de paiement de UniPay SAS

---

|  |    |
|--|----|
| 4. Diagramme d'activité.....   | 30 |
| 5. Diagramme de classe.....  | 35 |
| CHAPITRE IV : MISE EN ŒUVRE DE L'APPLICATION.....  | 37 |
| I. Principe : Model-View-ViewModel (MVVM).....   | 37 |
| II. Outils technologiques.....   | 38 |
| 1. Langage informatique.....   | 38 |
| 2. Outils technologiques utilisés.....   | 38 |
| III. Frameworks.....   | 42 |
| 1. Qu'est-ce qu'un Framework.....  | 42 |
| 2. Frameworks utilisés.....  | 43 |
| IV. Estimation des couts.....  | 45 |
| 1. Méthode de l'estimation par points de complexité et coût de prestation de l'équipe..... | 45 |
| 2. Coût du matériel de développement.....  | 46 |
| 3. Coût du matériel de déploiement.....  | 47 |
| V. Mise en œuvre.....  | 47 |
| 1. Environnement matériel de développement.....  | 47 |
| 2. Environnement logiciel.....   | 48 |
| 4. Politique de sécurisation.....  | 50 |
| 5. Présentation de quelques interfaces.....  | 51 |
| VI. Problèmes rencontrés.....  | 54 |
| CONCLUSION ET PERSPECTIVES.....  | 56 |
| BIBLIOGRAPHIE.....   | 58 |
| WEBOGRAPHIE.....   | 58 |

# INTRODUCTION

Dans un contexte où les solutions de portefeuille électronique et d'intégration d'API de paiement sont peu répandues en Afrique de l'Ouest, plus précisément au Burkina Faso, notre objectif était de développer un système répondant à cette demande croissante.

UniPay SAS, une entreprise bien établie sur le marché a souhaité offrir une solution complète à la fois aux utilisateurs de portefeuilles électroniques et aux entreprises désireuses de recevoir leurs paiements via des applications mobiles ou des sites web. Ce projet de stage a donc consisté à concevoir et réaliser une application web dédiée à l'administration et à l'interface entreprise, ainsi qu'une application mobile destinée aux utilisateurs des portefeuilles électroniques.

Notre solution ohSwap, intégrée à l'application web et mobile, offre aux entreprises la possibilité d'intégrer facilement un système de paiement en ligne sur leurs propres sites web. Grâce à cette intégration, les entreprises peuvent recevoir des paiements de manière sécurisée et efficace, tout en offrant une expérience utilisateur fluide à leurs clients.

Ce rapport vise à mettre en lumière les efforts déployés pour répondre à un besoin spécifique en Afrique de l'Ouest, en fournissant une solution complète de portefeuille électronique et d'intégration d'API de paiement. Il démontre l'importance de telles solutions dans un contexte où les transactions électroniques sont en pleine expansion et où les entreprises recherchent des moyens efficaces pour recevoir leurs paiements.

Dans la suite de ce document, nous détaillerons les différentes étapes de conception, de développement et d'intégration de notre solution. Nous présenterons également les technologies utilisées, les choix architecturaux effectués, ainsi que les résultats obtenus. Enfin, nous discuterons des perspectives d'amélioration et des potentielles évolutions futures de notre système.

# CHAPITRE I : GÉNÉRALITÉS

## I. Historique et Présentation de UniPay SAS

### 1. Présentation

**UniPay SAS** est une entreprise spécialisée dans les solutions de paiement électronique en Afrique de l'Ouest. Elle propose un mode de paiement en ligne nommé OhSwap, qui permet aux utilisateurs de réaliser des transactions électroniques en toute sécurité et de manière rapide.

OhSwap (anciennement unipay) permet aux utilisateurs de créer et gérer leur portefeuille électronique pour effectuer des paiements en ligne, des achats sur des sites marchands et des transferts d'argent en utilisant différentes méthodes de paiement telles que les cartes bancaires, les mobile money, les portefeuilles électroniques et les comptes bancaires.

**UniPay SAS** développe également d'autres solutions telles que Spots pour la gestion des campagnes publicitaires, IdVault pour la gestion des identités, Ecolo pour la gestion des ordures ménagères et des grands bacs communaux, ainsi que pour la production de café made in Burkina. Ces solutions visent à répondre aux besoins des utilisateurs et des entreprises de la région en offrant des solutions technologiques innovantes et adaptées.

### 2. Services

**UniPay SAS** est constituée de plusieurs services que sont :

- développement de solution digitale
- solutions de urbanisation et d'assainissement
- production de café made in Burkina

Pour notre stage nous avons été accueillis dans le service Développement de solution digitale.

## II. PRÉSENTATION DU CADRE DU PROJET

### 1. Contexte du stage

Le contexte du stage est la refonte du mode de paiement de OhSwap proposé par UniPay. Avant la refonte, OhSwap (anciennement unipay) avait des problèmes tels que l'absence de portefeuille électronique pour les utilisateurs, l'utilisation des commandes USSD pour les transactions, l'utilisation de Raspberry Pi pour l'exécution des transactions et l'absence d'API pour les entreprises.

Ainsi, le stage consiste à développer une application web utilisant Flask pour le backend, du JavaScript brute pour le front-end et Cordova pour le mobile, offrant une solution de portefeuille électronique aux utilisateurs pour leur permettre de déposer et de retirer de l'argent, ainsi qu'une API pour les entreprises pour leur permettre d'intégrer facilement le mode de paiement OhSwap sur leur site web ou application mobile.

Le stage vise également à améliorer l'expérience utilisateur en éliminant les problèmes liés à l'utilisation des commandes USSD pour les transactions et l'utilisation de Raspberry Pi pour l'exécution des transactions.

En somme, le contexte du stage est la refonte complète du mode de paiement OhSwap proposé par UniPay SAS pour offrir une solution de paiement électronique plus efficace, sécurisée et conviviale pour les utilisateurs et les entreprises en dans le monde.

### 2. Problématique

La problématique qui se pose dans le contexte des achats en ligne au Burkina Faso est le manque de solutions fiables permettant aux entreprises d'intégrer des fonctionnalités de paiement électronique à leurs applications mobiles ou sites web. Bien que des solutions de portefeuille électronique existent, elles ne proposent pas d'API adaptées aux besoins des entreprises locales. D'autre part, les rares solutions offrant des API ne sont pas considérées comme fiables et ne proposent pas de portefeuille électronique intégré.

Cette situation présente un véritable défi pour les entreprises, en particulier celles opérant dans le domaine de l'e-commerce comme UniPay SAS. Elles se retrouvent confrontées à la nécessité de développer leur propre solution de paiement, faute d'alternatives adéquates sur le marché local. Le développement d'une telle solution représente donc un enjeu majeur pour

UniPay SAS, qui souhaite offrir à ses clients une expérience d'achat en ligne sécurisée, fluide et pratique.

En prenant en compte cette problématique, notre objectif lors de ce stage était de concevoir et développer un système de portefeuille électronique doté d'une API fiable, permettant aux entreprises d'intégrer facilement des fonctionnalités de paiement électronique à leurs plateformes. Nous avons ainsi cherché à combler ce manque en proposant une solution innovante répondant aux besoins spécifiques du marché burkinabé, tout en offrant une expérience utilisateur optimale aussi bien pour les entreprises que pour les utilisateurs des portefeuilles électroniques.

### **3. Objectif du stage**

L'objectif principal de notre stage est de participer à la refonte du mode de paiement de ohSwap pour améliorer la qualité du service proposé aux utilisateurs et aux entreprises. Plus précisément, notre objectif sera de développer une application web avec Flask pour le backend, du JavaScript brute pour front-end et Cordova pour le mobile, qui permettra aux utilisateurs de créer et gérer leur portefeuille électronique, de réaliser des transactions en ligne et de déposer/recevoir de l'argent via différentes méthodes de paiement.

En outre, nous devons également développer des API pour permettre aux entreprises d'intégrer facilement le service de paiement dans leur site web ou application mobile et de proposer une expérience de paiement en ligne facile et sécurisée à leurs clients.

Le but du stage est donc de contribuer à la création d'un mode de paiement électronique performant, fiable et sécurisé, qui répondra aux besoins des utilisateurs et des entreprises et qui sera en mesure de rivaliser avec les autres solutions de paiement en ligne présentes sur le marché.

### **4. Les membres de l'équipe du projet**

Ce projet a été réalisé avec les acteurs suivant :

- Dr Faical Congo, PDG de UniPay SAS (Maître de stage)

## REFONTE ET CONCEPTION DU MODE DE PAIEMENT DE UniPay SAS

- Monsieur Gouayon KOALA, enseignant du département d'Informatique (UJKZ) (Superviseur)
- Fernand OUEDRAOGO étudiant au département Informatique (UJKZ) (Stagiaire).

## CHAPITRE II : SPECIFICATIONS ET ANALYSE DES BESOINS

### I. Démarche méthodologique

#### 1. Méthodologie

La méthodologie que nous avons suivie pour concevoir ohSwap repose sur plusieurs étapes clés, en intégrant les principes de la méthode Agile pour assurer un développement efficace et flexible.

Tout d'abord, nous avons effectué une analyse approfondie du marché des services financiers, en nous intéressant notamment aux besoins et aux attentes des utilisateurs. Grâce à des techniques agiles telles que les entretiens avec les clients et les entreprises, ainsi que la réalisation d'études de marché, nous avons pu recueillir des informations précieuses pour comprendre les exigences spécifiques du marché burkinabé en matière de paiement électronique.

Ensuite, nous avons utilisé les principes de l'approche Agile notamment la méthode SCRUM pour la phase de conception détaillée. En collaboration étroite avec nos partenaires et nos utilisateurs, nous avons élaboré des maquettes et des modèles fonctionnels pour ohSwap. Cette démarche itérative nous a permis de recueillir régulièrement les commentaires et les suggestions des parties prenantes, ce qui nous a permis d'ajuster et d'optimiser continuellement l'expérience utilisateur.

Le développement de la plateforme ohSwap a été réalisé en suivant une approche Agile SCRUM. Nous avons découpé le projet en itérations courtes, appelées "sprints", au cours desquels nous avons développé, testé et validé les fonctionnalités de manière progressive. Cette approche nous a permis de nous adapter rapidement aux changements et de répondre de manière flexible aux nouvelles exigences qui émergeaient tout au long du processus.

De plus, en utilisant des pratiques Agile telles que les revues de sprint et les rétrospectives, nous avons régulièrement évalué notre progression, identifié les points d'amélioration et ajusté notre plan de développement en conséquence. Cela nous a permis de garantir la qualité et la pertinence de la solution tout au long du projet.

Enfin, la phase de déploiement et de mise en production de ohSwap a également suivi une approche Agile. Nous avons adopté une approche incrémentale pour déployer progressivement les fonctionnalités sur la plateforme, tout en garantissant la stabilité et la disponibilité pour les utilisateurs. Les retours continus des utilisateurs ont été pris en compte pour les mises à jour et les améliorations ultérieures.

Dans l'ensemble, en utilisant la méthodologie Agile SCRUM, nous avons pu bénéficier d'une approche itérative et adaptative, en nous concentrant sur la collaboration, la flexibilité et la satisfaction des besoins des utilisateurs. Cela nous a permis de concevoir et de développer ohSwap de manière efficace, en répondant aux exigences spécifiques du marché burkinabé en matière de paiement électronique, tout en offrant une expérience utilisateur optimale.

### Scrum Process

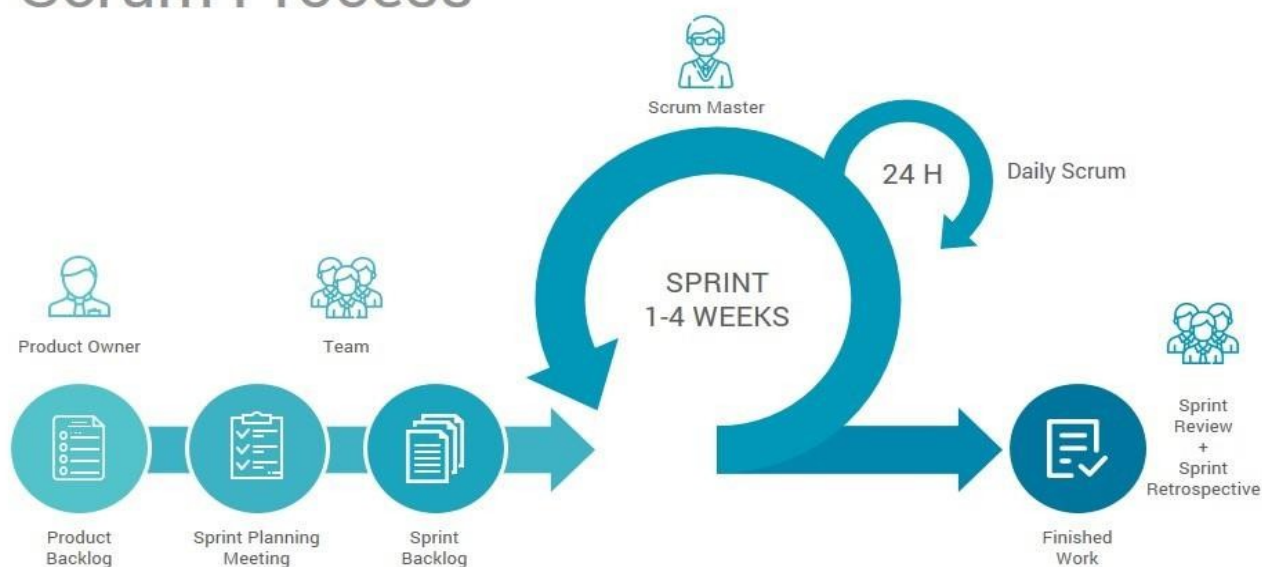


Figure 1 : Description de la méthode Agile SCRUM

Source : <https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/>

## 2. Langage de modélisation

Pour la réalisation de ce projet, nous avons utilisé le langage UML qui signifie « Unified Modeling Language » en français « Langage de Modélisation Unifié » et est défini comme un



outil de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. C'est une représentation abstraite et simplifiée d'une entité qui pourrait être un processus, un phénomène, un objet, etc., du monde réel en vue de le décrire, de l'expliquer ou de le prévoir.

De nos jours, UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet en offrant un standard logiciel pour la représentation de l'architecture logicielle. Ainsi, en accumulant UML et la méthode Agile, nous avons pu réaliser différents diagrammes de modélisation du projet.

### **3. Planning**

Pour ce projet nous avons suivi plusieurs étapes que sont :

- prise en main de l'écosystème de UniPay : UniPay étant une entreprise qui existait et qui proposait des solutions digitales elle avait son écosystème à elle qu'il fallait prendre en main avant toute chose ;
- l'analyse des besoins : Cela correspond à l'étape de recueil et la formalisation des besoins des futurs utilisateurs ;
- conception : Ici, nous définissons le fonctionnement du système qui sera mis en place et sa faisabilité. Cette phase est ensuite représentée sous plusieurs formes de diagramme afin de faciliter sa compréhension au plus large public et en particulier les utilisateurs ;
- Implémentation : Dans cette étape intervient le choix du langage de programmation que nous allons choisir afin de traduire les fonctionnalités définies en un programme fonctionnel qui répond aux besoins des utilisateurs ;
- Test : Cette étape nous permet de nous assurer du fonctionnement et de la conformité du système vis-à-vis des besoins exprimés.
- Déploiement : Ici nous faisons plusieurs déploiements y compris de test pour s'assurer que tout se passera bien en production ;

## REFONTE ET CONCEPTION DU MODE DE PAIEMENT DE UniPay SAS

- Maintenance : C'est la phase de correction de bug et d'incohérence après le déploiement ;
- Documentation : Documenter le travail permet de s'assurer que une autre personne pourra prendre la main plus tard et facilite surtout l'intégration de la solution ;

Le planning du projet a été établi en tenant compte des principes de la méthode Agile, ce qui implique que certaines étapes ont été menées simultanément plutôt que de manière séquentielle. Cette approche flexible nous a permis de maximiser l'efficacité en traitant différents aspects du projet en parallèle, tout en maintenant une communication régulière et une adaptation continue aux besoins changeants.

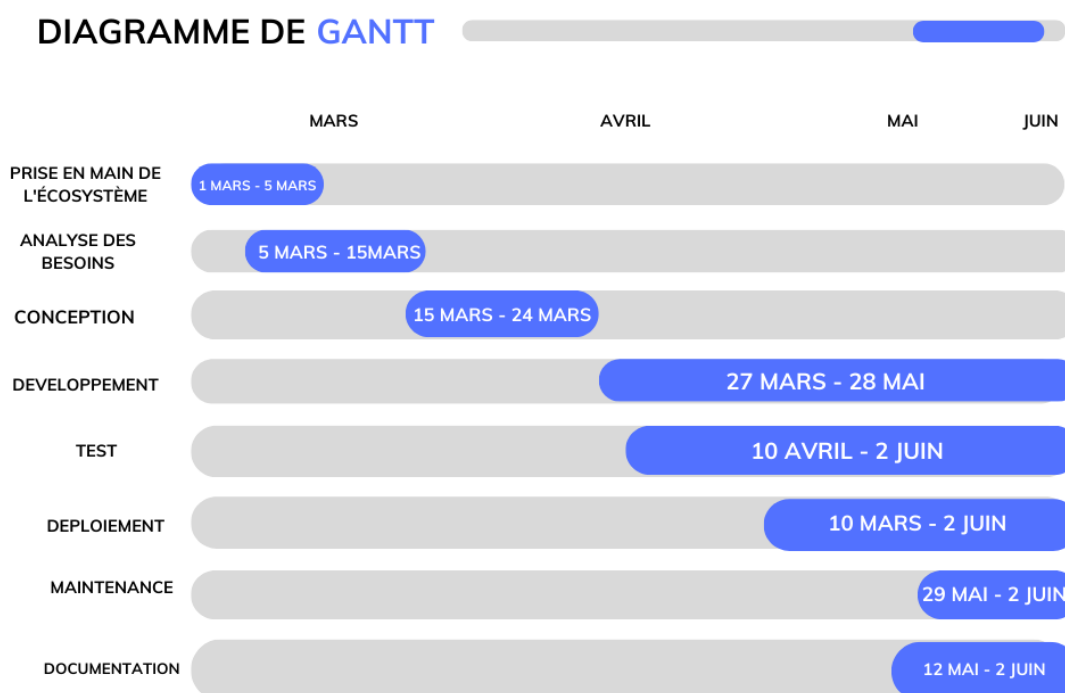


Figure 2 : Planning

## II. Étude de l'existant

Lors de notre arrivée chez UniPay, nous avons pu constater l'état initial du projet et identifier les différentes composantes qui nécessitaient des améliorations. Sur le plan technique, le backend de l'application n'était pas encore conçu, ce qui représentait un défi majeur à relever. De plus, le front-end, notamment le site d'administration, utilisait un template qui ne répondait pas aux besoins spécifiques de UniPay.

Un autre aspect important du projet était la mise en place des fonctionnalités permettant d'effectuer des transactions par le biais des commandes USSD, en utilisant les services d'Orange Money et de Moov Money. Heureusement, une partie du code pour exécuter ces commandes USSD avait déjà été développée, ce qui nous a donné une base solide à partir de laquelle travailler.

En ce qui concerne l'hébergement des applications, UniPay avait opté pour Microsoft Azure, une plateforme cloud réputée pour sa fiabilité et sa scalabilité. De plus, la base de données utilisée était MongoDB, qui était hébergée sur Atlas, la solution de base de données cloud de MongoDB. Ces choix technologiques ont offert à UniPay une infrastructure solide et évolutive pour soutenir ses opérations.

Dans cette phase initiale du projet, mon rôle a été d'évaluer l'existant, d'identifier les lacunes et les points d'amélioration, et de proposer des solutions pour les résoudre. Cela comprenait la conception et le développement du backend, l'adaptation du frontend aux besoins spécifiques d'UniPay, ainsi que l'optimisation et l'intégration des fonctionnalités liées aux commandes USSD et le développement de l'application mobile.

## CHAPITRE III : ANALYSE ET CONCEPTION DU SYSTEME

### I. Domaine d'étude et fonctionnalités

#### 1. Domaine d'étude

Ce stage s'inscrit dans le domaine des paiements en ligne, nos principales tâches seront de :

- proposer un API aux entreprises pour recevoir leurs paiements sur leurs applications mobile ou web ;
- mettre a la disposition des utilisateurs un portefeuille électronique pour transférer et recevoir de l'argent, faire des achats en ligne, transférer l'argent vers leurs compte bancaire et renflouer le portefeuille avec leurs cartes bancaires ;
- transférer l'argent de son wallet vers un compte ou une carte bancaire

#### 2. Fonctionnalités

Pour une meilleure gestion de la solution nous avons quatre (4) statuts d'utilisateurs.

Un wallet (utilisateur possédant un portefeuille électronique) peut:

- transférer de l'argent a un autre wallet
- consulter la liste de ses transactions
- valider une transaction initié sur le site web ou application mobile d'une entreprise
- Choisir son mode de paiement.

Une entreprise (utilisateur ayant ouvert un compte entreprise) peut:

- initier une transaction
- voir la liste de ses transactions
- voir la liste de ses clients
- générer une clé API
- faire une demande de virement de fond
- voir ses demandes de virement

Un administrateur peut :

- voir toutes les transactions réaliser sur la plateforme
- ajouter une transaction
- modifier les informations d'une transaction
- voir toutes les passerelles de la plateforme
- ajouter une passerelle
- modifier les informations d'une passerelle
- voir toutes les commandes USSD
- ajouter une commande USSD
- modifier une commande USSD
- voir toutes les entreprises
- voir tout les utilisateurs
- ajouter un utilisateur
- Voir toutes les clés d'API des entreprises

Un root peut

- faire tout ce que l'administrateur fait
- supprimer toute information disponible sur la plateforme

## II. Modélisation

### 1. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation est un modèle UML qui représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui ou s'assure la relation entre l'utilisateur et les objets que le système met en œuvre. Le diagramme des cas d'utilisateur est basé sur certains concepts tels que : acteur (l'utilisateur), cas d'utilisation (fonctionnalité) et les liens qui relient les acteurs, les cas d'utilisation ou les acteurs et les cas d'utilisation.

Les différents acteurs du système :

- un wallet (utilisateur possédant un portefeuille électronique)
- un entreprise (utilisateur ayant ouvert un compte entreprise)
- un administrateur

- un root

Les différents cas d'utilisation du système :

- s'enregistrer
- s'authentifier
- valider un paiement
- transférer de l'argent
- voir ses transactions
- choisir un moyen de paiement
- initier un paiement
- générer une clé d'API
- consulter son solde
- voir la liste des clients
- consulter l'historique des transactions
- modifier une transaction
- supprimer une transaction
- voir les commandes USSD
- ajouter une commande USSD
- modifier une commande USSD
- supprimer une commande USSD
- voir les passerelles
- ajouter une passerelle
- modifier une passerelle
- supprimer une passerelle

- voir la listes des utilisateurs
- ajouter un utilisateur
- modifier un utilisateur
- supprimer un utilisateur

Figure 3 : Diagramme de cas d'utilisation





## 2. Description textuelle de quelque cas d'utilisation

Le diagramme de cas d'utilisation sert à décrire les grandes fonctions du système du point de vue des acteurs, mais ne donne pas d'explication sur le dialogue entre les acteurs et les cas d'utilisation.

La description textuelle est une forme plus légère pour l'illustration d'un cas d'utilisation. La description textuelle comporte trois (3) étapes dont les deux premières sont obligatoires et la dernière facultative : l'identification du cas d'utilisation, le séquençement et les contraintes opérationnelles. Nous présenterons la description des différents cas d'utilisation sous forme de tableau.

Description textuelle du cas d'utilisation « Ouvrir un compte ohSwap »

Tableau 1 : Ouvrir un compte ohSwap

|                  |   |
|------------------|---|
| Préconditions    | <ul style="list-style-type: none"> <li>L'utilisateur accède à la page d'inscription ohSwap.</li> </ul>  |
| Identification   | <ul style="list-style-type: none"> <li>Nom du cas d'utilisation : Ouvrir un compte ohSwap</li> <li>Objectif : Ouvrir un compte ohSwap</li> <li>Acteurs principaux : Utilisateur non connecté</li> <li>Date : 10/06/2023</li> <li>Responsable : OUEDRAOGO Fernand</li> <li>Version : 1.0</li> </ul>                      |
| Scénario nominal | <p>L'utilisateur saisit ses informations personnelles (nom, prénom, adresse e-mail ou numéro de téléphone, mot de passe).</p> <p>1</p> <p>2 Le système envoie un code de validation par e-mail ou par numéro de téléphone à l'adresse fournie.</p> <p>3 L'utilisateur saisit le code de validation reçu par e-mail.</p> |

|                      |     |   |
|----------------------|-----|---|
| Scénario alternatifs | 4   | Le système vérifie que le code de validation est correct.   |
|                      | 5   | Le système crée un compte ohSwap avec le type en question associé à l'adresse e-mail fournie                        |
|                      | 2.a | L'adresse e-mail saisie par l'utilisateur n'est pas unique et valide  |
|                      |     | Le système affiche un message d'erreur demandant à l'utilisateur de saisir une adresse e-mail valide.               |
|                      |     | L'utilisateur saisit une nouvelle adresse e-mail.   |
|                      | 3.a | Le système n'arrive pas à envoyer le code de validation   |
|                      |     | Le système affiche un message d'erreur demandant à l'utilisateur de vérifier son adresse e-mail et de réessayer     |
|                      |     | L'utilisateur peut demander à renvoyer le code de validation.   |
|                      | 5.a | Le code de validation saisi par l'utilisateur n'est pas correct   |
|                      |     | Le système affiche un message d'erreur demandant à l'utilisateur de vérifier le code de validation et de réessayer. |
| Postcondition        | —   | L'utilisateur a un compte ohSwap et peut s'authentifier pour accéder aux fonctionnalités de la plateforme.          |
| Scénario d'exception |     | Aucun scénario d'exception  |

Description textuelle du cas d'utilisation « S'authentifier »

Tableau 2 : S'authentifier

|               |  |
|---------------|--|
| Préconditions | <ul style="list-style-type: none"> <li>• L'utilisateur doit avoir un compte enregistré sur le système</li> <li>• L'utilisateur doit être en possession d'un API token</li> </ul> |
|---------------|--|

|                      |  |
|----------------------|--|
|                      | <ul style="list-style-type: none"> <li>L'utilisateur doit fournir une adresse e-mail et un mot de passe valides pour se connecter</li> </ul>   |
| Identification       | <ul style="list-style-type: none"> <li>Nom du cas d'utilisation : S'authentifier</li> <li>Objectif : Etre connecter a l'application mobile ou a l'application web</li> <li>Acteurs principaux : Wallet, entreprise, admin ou root</li> <li>Date : 10/06/2023</li> <li>Responsable : OUEDRAOGO Fernand</li> <li>Version : 1.0</li> </ul>  |
| Scénario nominal     | <p>L'utilisateur saisit son adresse e-mail et mot de passe ou numéro</p> <ol style="list-style-type: none"> <li>de téléphone.</li> <li>Le système envoie un code de validation par e-mail ou par numéro de téléphone à l'adresse fournie.</li> <li>L'utilisateur saisit le code de validation reçu par e-mail.</li> <li>Le système vérifie que le code de validation est correct.</li> <li>Le système authentifie l'utilisateur et le redirige vers la page principale de l'application mobile ou web</li> </ol> |
| Scénario alternatifs | 2.a Le système détecte que l'utilisateur n'a pas fourni des informations de connexion valides, il affiche un message d'erreur à l'utilisateur et lui demande de saisir à nouveau ses informations de connexion   |
|                      | 3.a Le système détecte que l'utilisateur n'a pas reçu de code de connexion, il affiche un message d'erreur à l'utilisateur et lui demande de vérifier son e-mail ou son application mobile pour recevoir le code de connexion  |
|                      | 4.a Le système détecte que le code de connexion fourni par   |

|                      |   |
|----------------------|---|
|                      | l'utilisateur n'est pas valide, il affiche un message d'erreur à l'utilisateur et lui demande de saisir à nouveau le code de connexion  |
| Scénario d'exception | Si le système ne parvient pas à valider le token d'authentification, il affiche un message d'erreur à l'utilisateur et l'empêche de poursuivre le processus de connexion  |
| Postcondition        | <ul style="list-style-type: none"> <li>– L'utilisateur est authentifié et connecté au système</li> <li>– L'utilisateur peut accéder aux fonctionnalités réservées aux utilisateurs connectés</li> <li>– Génère un token pour l'utilisateur</li> </ul> |

Description textuelle du cas d'utilisation « Initier paiement »

Tableau 3 : Initier paiement

|                     |  |
|---------------------|--|
| Préconditions       | <ul style="list-style-type: none"> <li>• L'entreprise doit avoir fourni une clé d'API valide à ohSwap</li> <li>• Le client doit entrer son numéro de téléphone sur le site web de l'entreprise</li> </ul>  |
| Identification      | <ul style="list-style-type: none"> <li>• Nom du cas d'utilisation : Initier paiement</li> <li>• Objectif : Pour initier une réception d'argent sur son site web ou application mobile</li> <li>• Acteurs principaux : Entreprise</li> <li>• Acteurs secondaires : Wallet</li> <li>• Date : 10/06/2023</li> <li>• Responsable : OUEDRAOGO Fernand</li> <li>• Version : 1.0</li> </ul>   |
| Scénario nominal    | <p>Le client entre son numéro de téléphone sur le site web de</p> <ol style="list-style-type: none"> <li>1 l'entreprise pour initier le paiement</li> <li>2 L'entreprise envoie une demande de paiement à ohSwap en utilisant sa clé d'API.</li> <li>3 ohSwap vérifie que le client a suffisamment d'argent dans son wallet pour effectuer le paiement.</li> <li>4 Le système vérifie que le code de validation est correct.ohSwap crée une transaction pour le paiement demandé.</li> <li>5 ohSwap envoie une notification à l'application mobile du client pour lui demander de valider la transaction.</li> </ol> |
| Scénario Alternatif | Aucun  |

|                      |  |
|----------------------|--|
| Scénario d'exception | Si le client n'a pas suffisamment d'argent dans son wallet, la transaction est annulée et un message d'erreur est envoyé à l'entreprise.                       |
| Postcondition        | <ul style="list-style-type: none"><li>• L'entreprise a initié un paiement via son site web ou son application mobile avec les coordonnées du wallet.</li></ul> |

## Description textuelle du cas Description « Valider un paiement »

Tableau 4 : Valider un paiement

|                      |   |
|----------------------|---|
| Préconditions        | <ul style="list-style-type: none"> <li>Le client doit être enregistré et authentifié sur l'application mobile ohSwap.</li> <li>Un paiement a été initié dans le site web de l'entreprise avec ses infos.</li> </ul>   |
| Identification       | <ul style="list-style-type: none"> <li>Nom du cas d'utilisation : Valider un paiement</li> <li>Objectif : Valider un paiement initié sur le site web ou l'application mobile de l'entreprise</li> <li>Acteurs principaux : Wallet</li> <li>Acteurs secondaires : Entreprise</li> <li>Date : 10/06/2023</li> <li>Responsable : OUEDRAOGO Fernand</li> <li>Version : 1.0</li> </ul>       |
| Scénario nominal     | <p>Le client reçoit une notification de paiement en attente sur son</p> <ol style="list-style-type: none"> <li>1 application mobile.</li> <li>2 Le client ouvre l'application mobile et sélectionne la notification pour valider le paiement.</li> <li>3 ohSwap vérifie que la transaction est valide.</li> <li>4 ohSwap envoie une confirmation de paiement à l'entreprise.</li> </ol> |
| Scénario Alternatif  | Si le client ne sélectionne pas la notification de paiement en attente, la transaction est annulée.   |
| Scénario d'exception | Si la transaction n'est pas valide, la transaction est annulée et un message d'erreur est envoyé à l'entreprise.  |
| Postcondition        | Le paiement est validé et confirmé par ohSwap et l'entreprise est   |

|  |                              |
|--|------------------------------|
|  | informée de la confirmation. |
|--|------------------------------|

## Description textuelle du cas d'utilisation « Transférer de l'argent »

Tableau 5 : Transférer de l'argent

|                     |  |
|---------------------|--|
| Préconditions       | <ul style="list-style-type: none"> <li>L'utilisateur est connecté à son compte OhSwap et a suffisamment de fonds dans son wallet.</li> </ul>   |
| Identification      | <ul style="list-style-type: none"> <li>Nom du cas d'utilisation : Transférer de l'argent</li> <li>Objectif : Transférer de l'argent à un autre wallet</li> <li>Acteurs principaux : Wallet</li> <li>Date : 10/06/2023</li> <li>Responsable : OUEDRAOGO Fernand</li> <li>Version : 1.0</li> </ul>   |
| Scénario nominal    | <ol style="list-style-type: none"> <li>1 L'utilisateur sélectionne l'option "Transférer de l'argent".</li> <li>2 L'utilisateur entre le montant qu'il souhaite transférer et l'adresse de wallet de l'utilisateur destinataire.</li> <li>3 Le système vérifie que l'adresse de wallet du destinataire est valide et que l'utilisateur a suffisamment de fonds pour effectuer la transaction.</li> <li>4 Le système crée une transaction à partir du wallet de l'utilisateur et débite le montant transféré.</li> <li>5 Le système envoie une notification à l'utilisateur pour l'informer que la transaction a été effectuée.</li> </ol> |
| Scénario Alternatif | Si l'adresse de wallet du destinataire est invalide ou si l'utilisateur n'a pas suffisamment de fonds pour effectuer la transaction, le  |



|                      |  |
|----------------------|--|
|                      | système affiche un message d'erreur à l'utilisateur et la transaction est annulée.   |
| Scénario d'exception | Aucun scénario d'exception identifié   |
| Postcondition        | <ul style="list-style-type: none"> <li>Le montant transféré a été débité du wallet de l'utilisateur et crédité sur le wallet du destinataire.</li> </ul> |

Description textuelle du cas d'utilisation « Ajouter de l'argent dans son wallet »

Tableau 7 : Ajouter de l'argent dans son wallet

|                  |  |
|------------------|--|
| Préconditions    | <ul style="list-style-type: none"> <li>L'utilisateur est connecté à son compte OhSwap.</li> <li>L'utilisateur doit avoir accès à un moyen de paiement valide.</li> </ul>   |
| Identification   | <ul style="list-style-type: none"> <li>Nom du cas d'utilisation : Ajouter de l'argent dans son wallet</li> <li>Objectif : Ajouter de l'argent dans son wallet</li> <li>Acteurs principaux : Wallet</li> <li>Date : 10/06/2023</li> <li>Responsable : OUEDRAOGO Fernand</li> <li>Version : 1.0</li> </ul>   |
| Scénario nominal | <p>L'utilisateur sélectionne l'option "Ajouter de l'argent" dans le</p> <ol style="list-style-type: none"> <li>1 menu de l'application.</li> <li>2 Le système affiche les différents moyens de paiement disponibles pour l'utilisateur.</li> <li>3 L'utilisateur sélectionne le moyen de paiement qu'il souhaite utiliser pour ajouter de l'argent à son wallet.</li> <li>4 Le système demande à l'utilisateur de fournir les informations nécessaires pour effectuer le transfert de fonds depuis le moyen</li> </ol> |

|                      |   |
|----------------------|---|
|                      | <p>de paiement sélectionné.</p> <p>5 L'utilisateur fournit les informations requises pour effectuer le transfert de fonds.</p> <p>6 Le système vérifie les informations fournies par l'utilisateur et effectue le transfert de fonds depuis le moyen de paiement sélectionné vers le wallet de l'utilisateur.</p> <p>7 Le système met à jour le solde du wallet de l'utilisateur pour refléter le montant ajouté.</p> |
| Scénario Alternatif  | Si l'adresse de wallet du destinataire est invalide ou si l'utilisateur n'a pas suffisamment de fonds pour effectuer la transaction, le système affiche un message d'erreur à l'utilisateur et la transaction est annulée.  |
| Scénario d'exception | Aucun scénario d'exception identifié  |
| Postcondition        | <ul style="list-style-type: none"> <li>Le wallet de l'utilisateur a été crédité du montant ajouté.</li> <li>Le solde du moyen de paiement utilisé pour le transfert de fonds a été débité du montant transféré.</li> </ul>  |

Description textuelle du cas d'utilisation « Générer une clé d'API »

Tableau 7 : Générer une clé d'API

|                |   |
|----------------|---|
| Préconditions  | <ul style="list-style-type: none"> <li>L'entreprise est connecté à son compte OhSwap.</li> </ul>  |
| Identification | <ul style="list-style-type: none"> <li>Nom du cas d'utilisation : Générer une clé d'API</li> <li>Objectif : Obtenir un API pour ses transactions</li> <li>Acteurs principaux : Entreprise</li> <li>Date : 10/06/2023</li> </ul> |

|                      |  |
|----------------------|--|
|                      | <ul style="list-style-type: none"> <li>• Responsable : OUEDRAOGO Fernand</li> <li>• Version : 1.0</li> </ul>   |
| Scénario nominal     | <ol style="list-style-type: none"> <li>1 L'entreprise accède à la page de gestion des clés d'API.</li> <li>2 L'entreprise choisit de créer une nouvelle clé d'API.</li> <li>3 OhSwap vérifie que l'entreprise n'a pas déjà un API</li> <li>4 OhSwap génère une nouvelle clé d'API pour l'entreprise.</li> <li>5 L'entreprise utilise la nouvelle clé d'API pour intégrer OhSwap comme moyen de paiement sur son site web ou application mobile.</li> </ol> |
| Scénario Alternatif  | 3.a L'entreprise possède déjà une clé d'API, et le système lui refuse la création de la nouvelle.  |
| Scénario d'exception | Aucun scénario d'exception identifié   |
| Postcondition        | <ul style="list-style-type: none"> <li>• L'entreprise a une nouvelle clé d'API valide pour intégrer OhSwap comme moyen de paiement sur son site web ou application mobile.</li> </ul>  |

### 3. Diagramme de séquence de quelques cas d'utilisation

Le diagramme de séquence est un graphe montrant les interactions entre les objets en mettant l'accent sur l'aspect temporel, c'est-à-dire la chronologie des envois de messages. Le diagramme de séquence permet de modéliser :

- Les scénarios d'utilisation : une façon d'utiliser le système ; tout ou partiel d'un Cas d'Utilisation, ou une combinaison de plusieurs Cas d'Utilisation
- La logique des méthodes : en parti dans le cas des opérations complexes

- La logique des services : Méthodes de haut niveau invocable par plusieurs variétés de clients.

Diagramme de séquence du cas d'utilisation « Ajouter de l'argent dans son wallet »

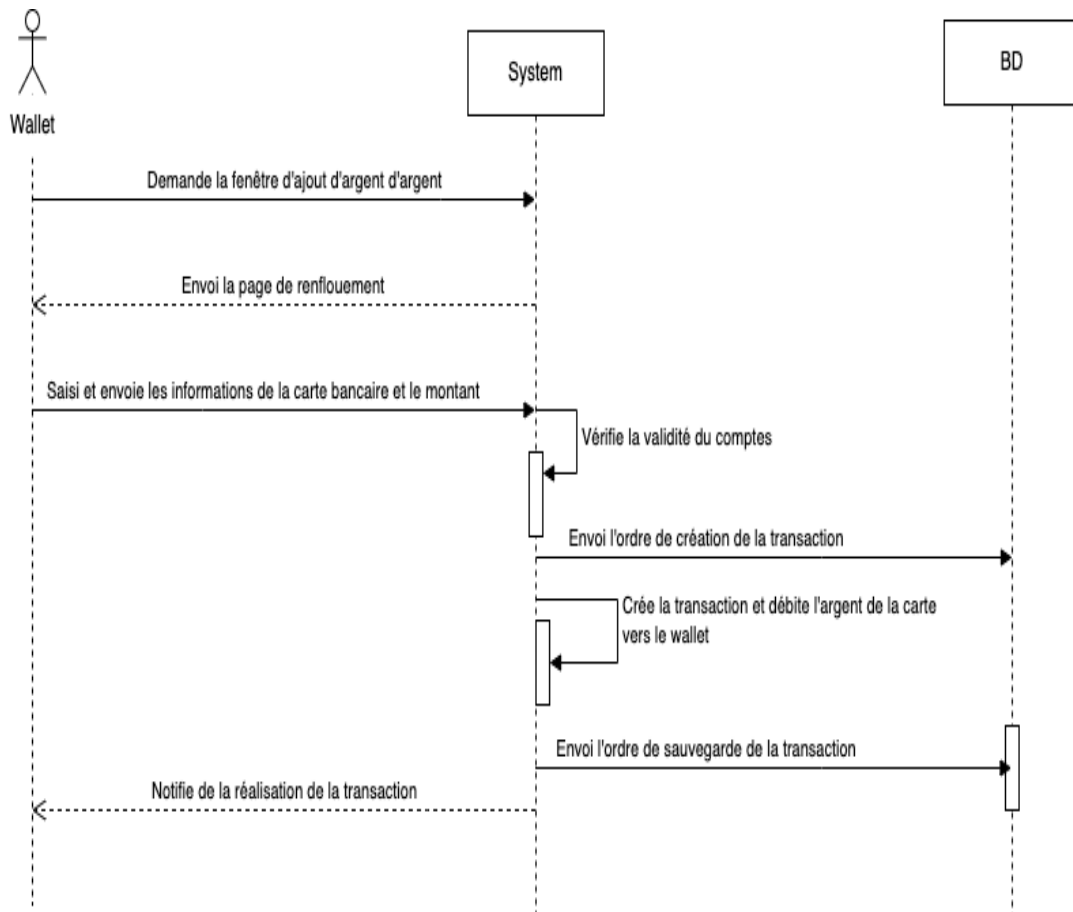


Figure 4 : Diagramme de séquence ajouter de l'argent dans son wallet

Diagramme de séquence du cas d'utilisation « Choisir moyen de paiement »

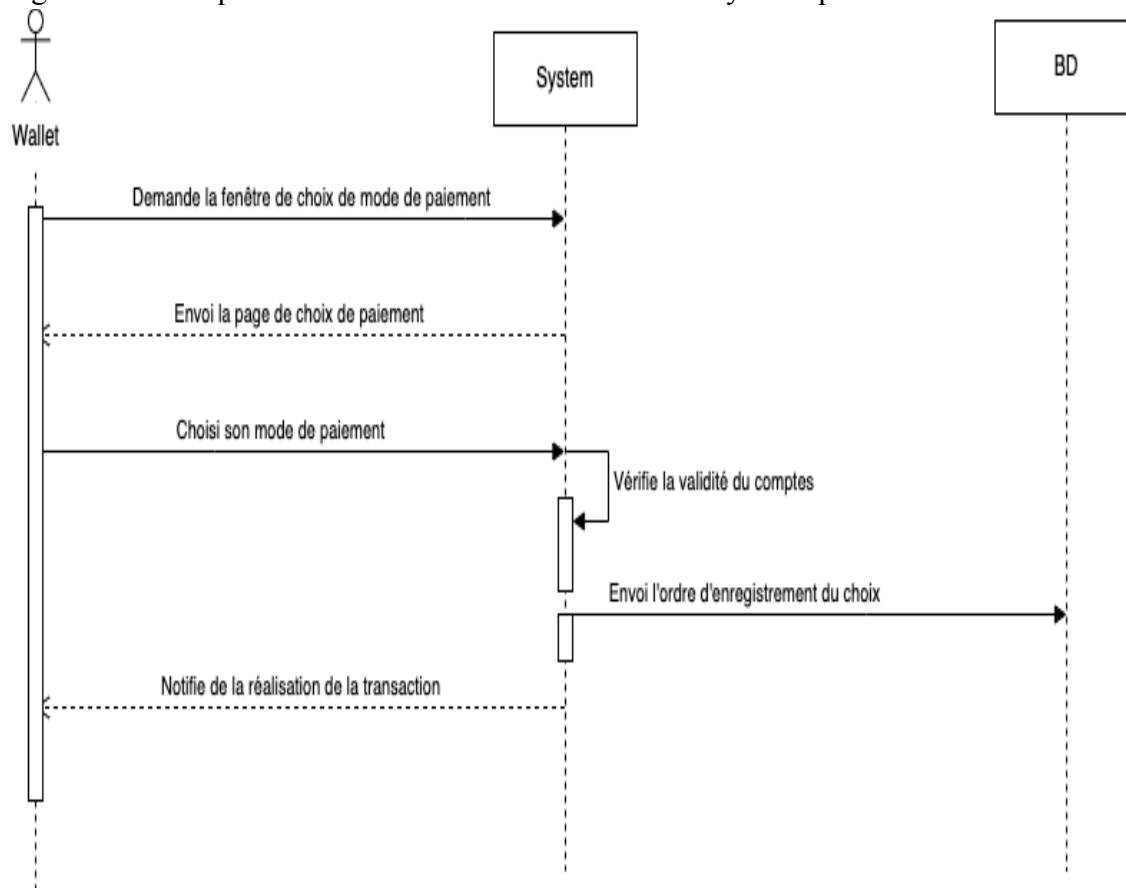


Figure 5 : Diagramme de séquence de choisir moyen de paiement

Diagramme de séquence du cas d'utilisation « S'authentifier »

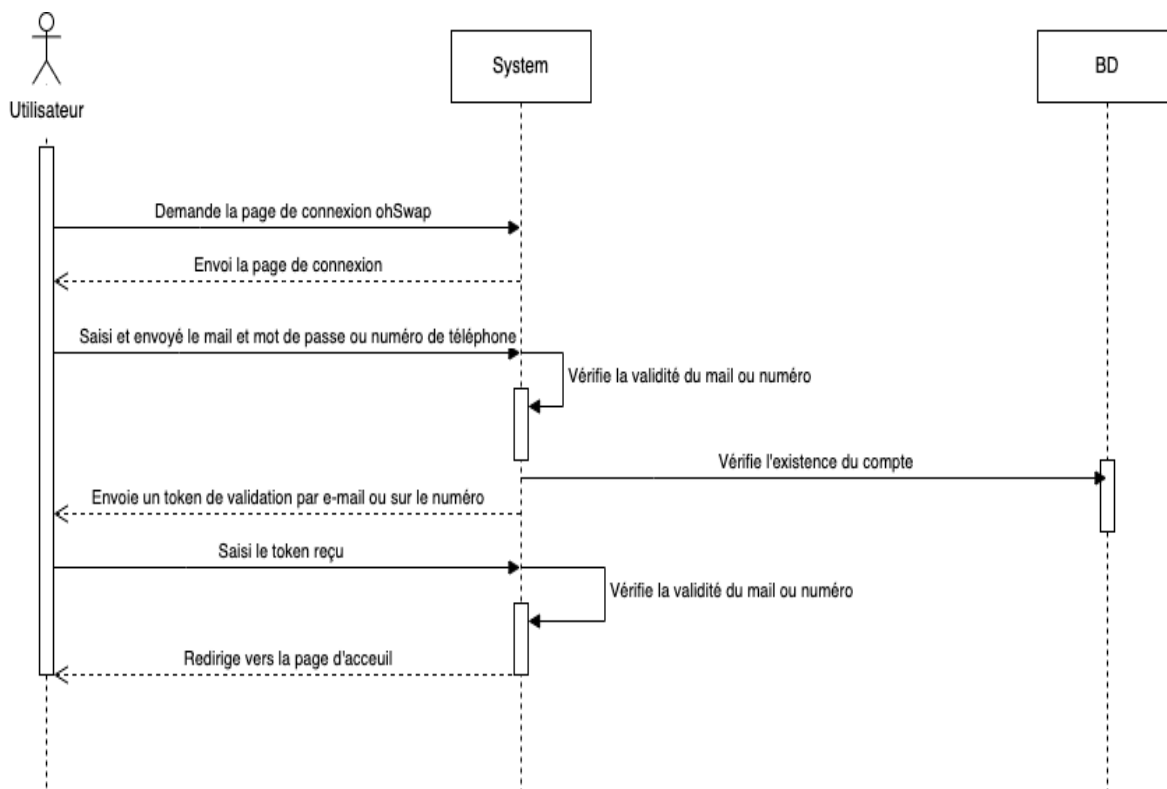


Figure 6 : Diagramme de séquence s'authentifier

## Diagramme de séquence du cas d'utilisation « Transférer de l'argent »

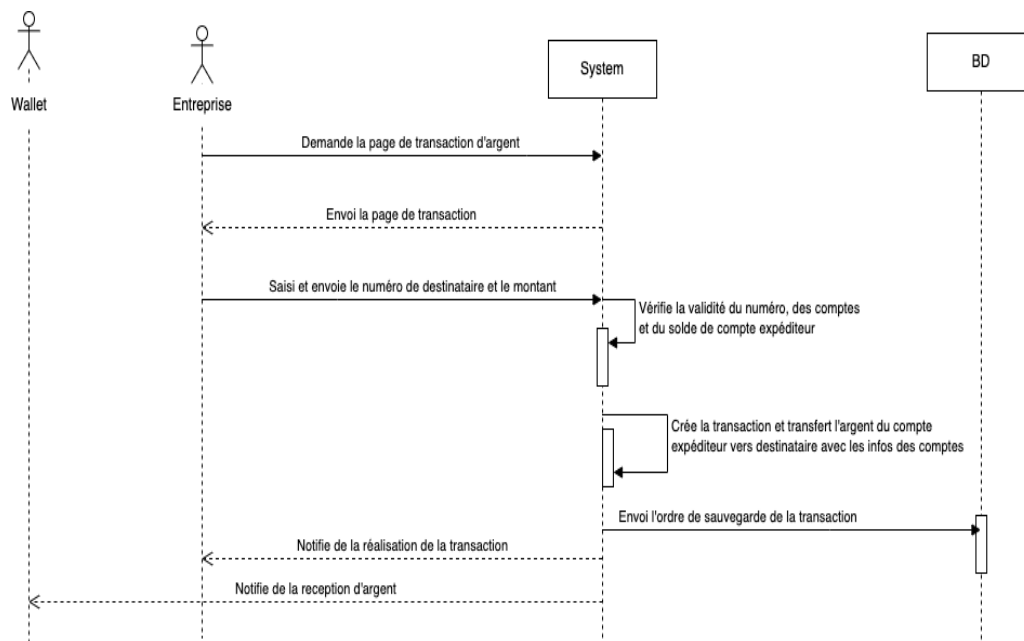


Figure 7 : Diagramme de séquence transférer de l'argent

Diagramme de séquence du cas d'utilisation « Générer une clé d'API »

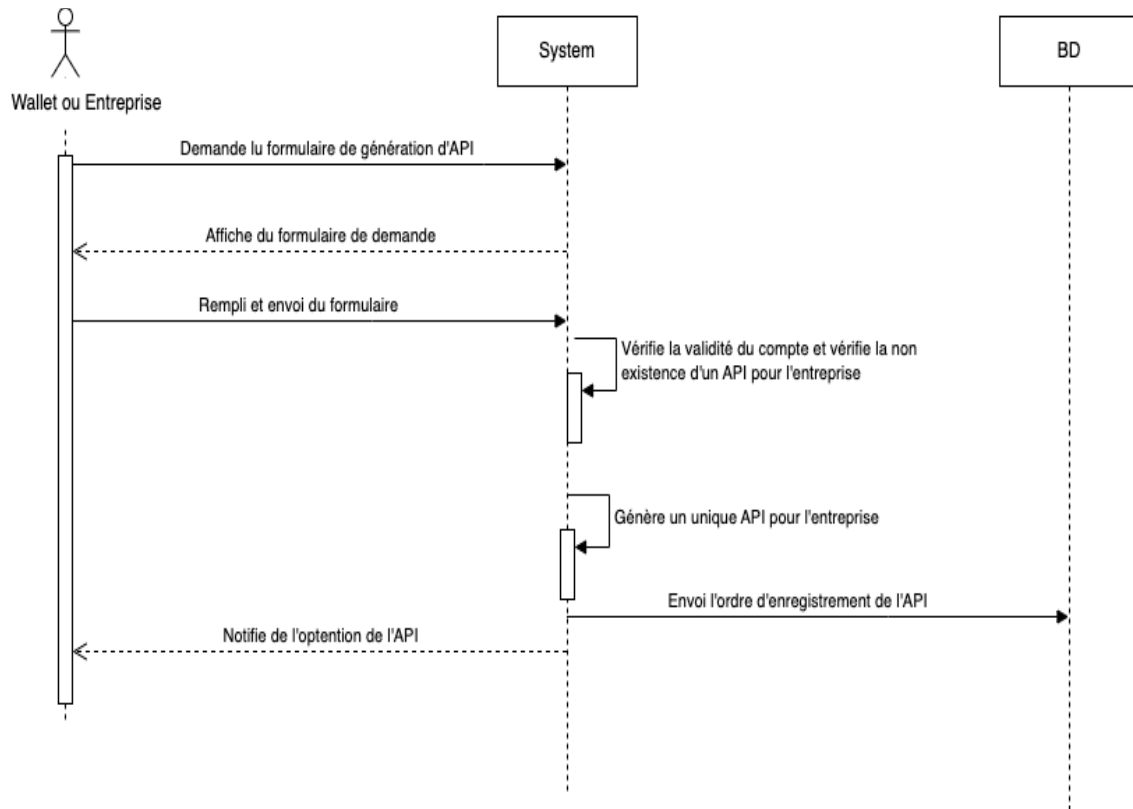


Figure 8 : Diagramme de séquence générer une clé d'API

Diagramme de séquence du cas d'utilisation « Initier un paiement »

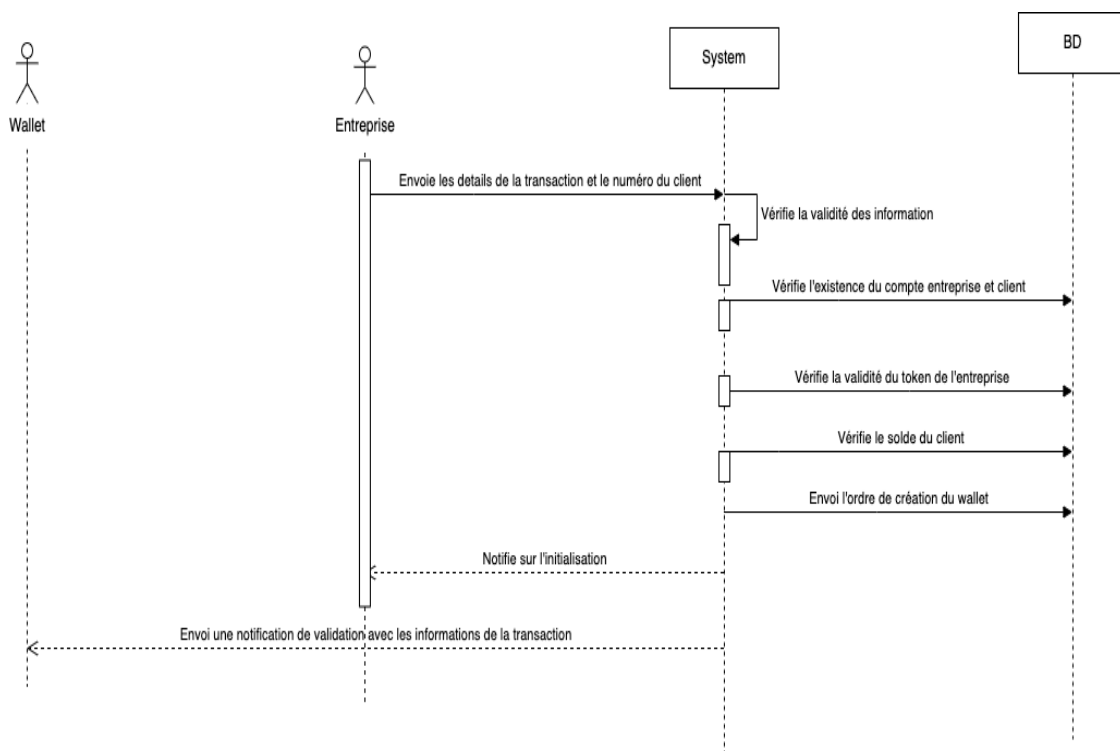


Figure 9 : Diagramme de séquence initier un paiement

Diagramme de séquence du cas d'utilisation « valider un paiement »

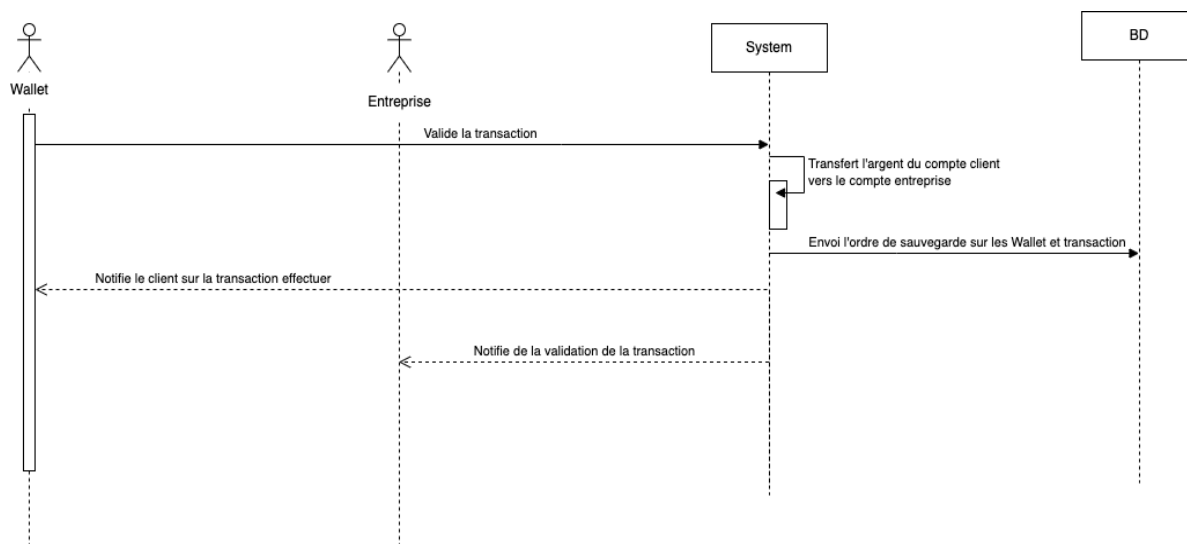


Figure 10 : Diagramme de séquence valider un paiement



#### **4. Diagramme d'activité**

Les diagrammes d'activités permettent de décrire les traitements. C'est le diagramme le plus adapté à la modélisation du cheminement de flots de contrôle et de flots de données, car il permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Les diagrammes d'activités viennent illustrer et consolider la description textuelle des cas d'utilisation en donnant une description plus précise des traitements. Pour notre étude, nous allons représenter les diagrammes d'activité de quelques cas d'utilisation vus plus hauts.

Diagramme d'activité mettre l'argent dans son wallet



Figure 11: Diagramme d'activité mettre l'argent dans son wallet

## Diagramme d'activité transférer de l'argent

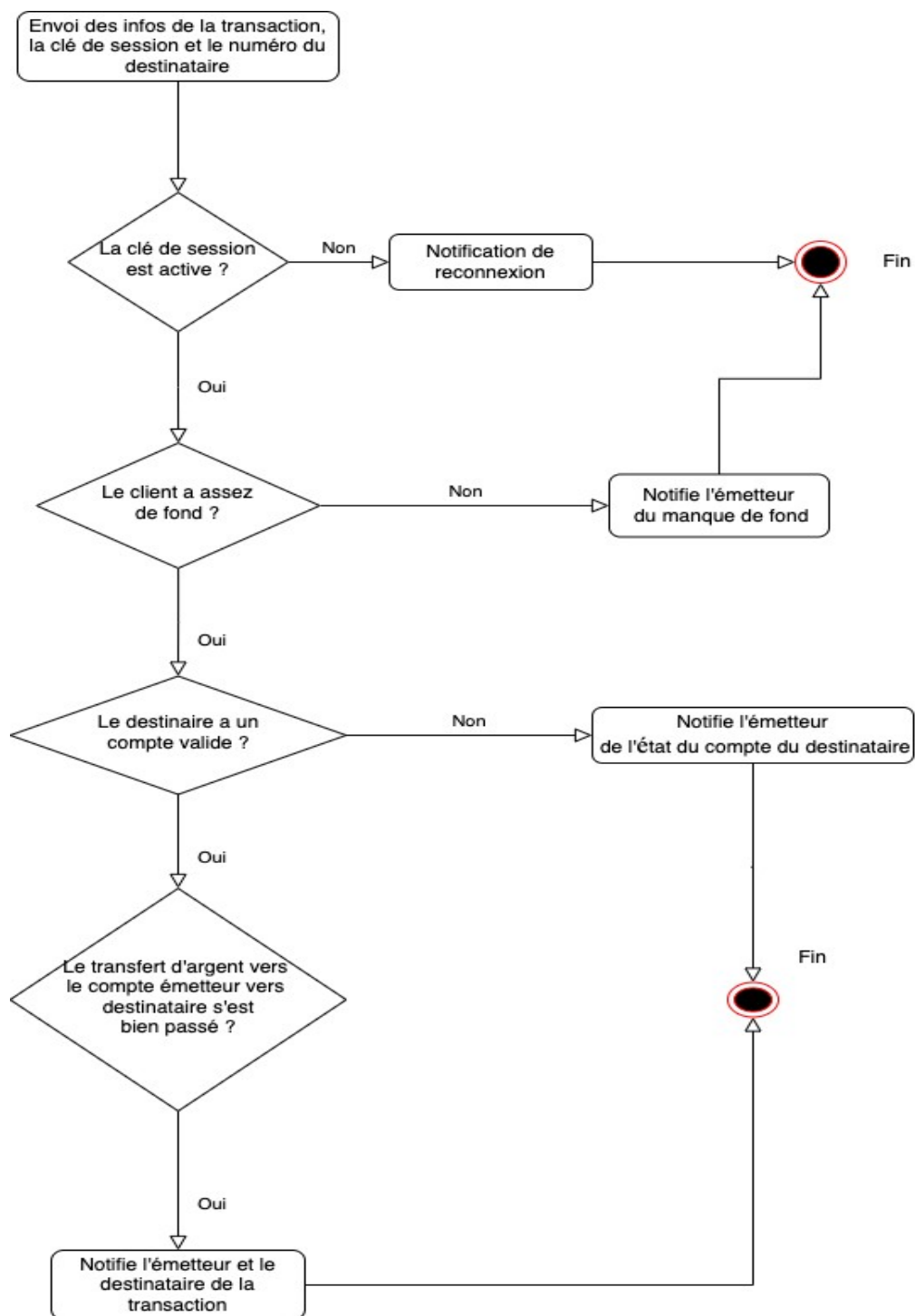


Figure 12 : Diagramme d'activité transférer de l'argent

## Diagramme d'activité initier un paiement

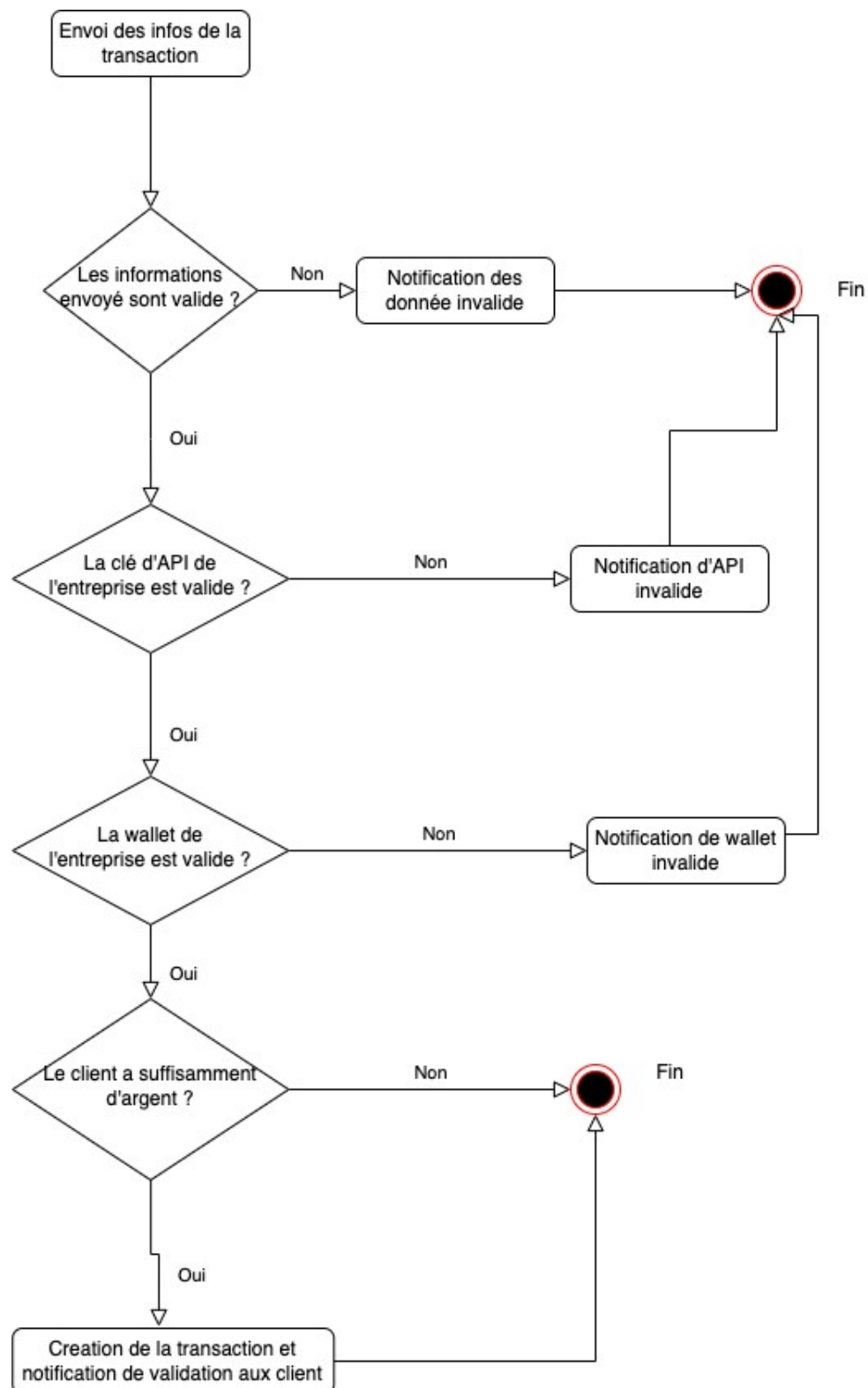


Figure 13 : Diagramme d'activité initier un paiement

## Diagramme d'activité valider un paiement

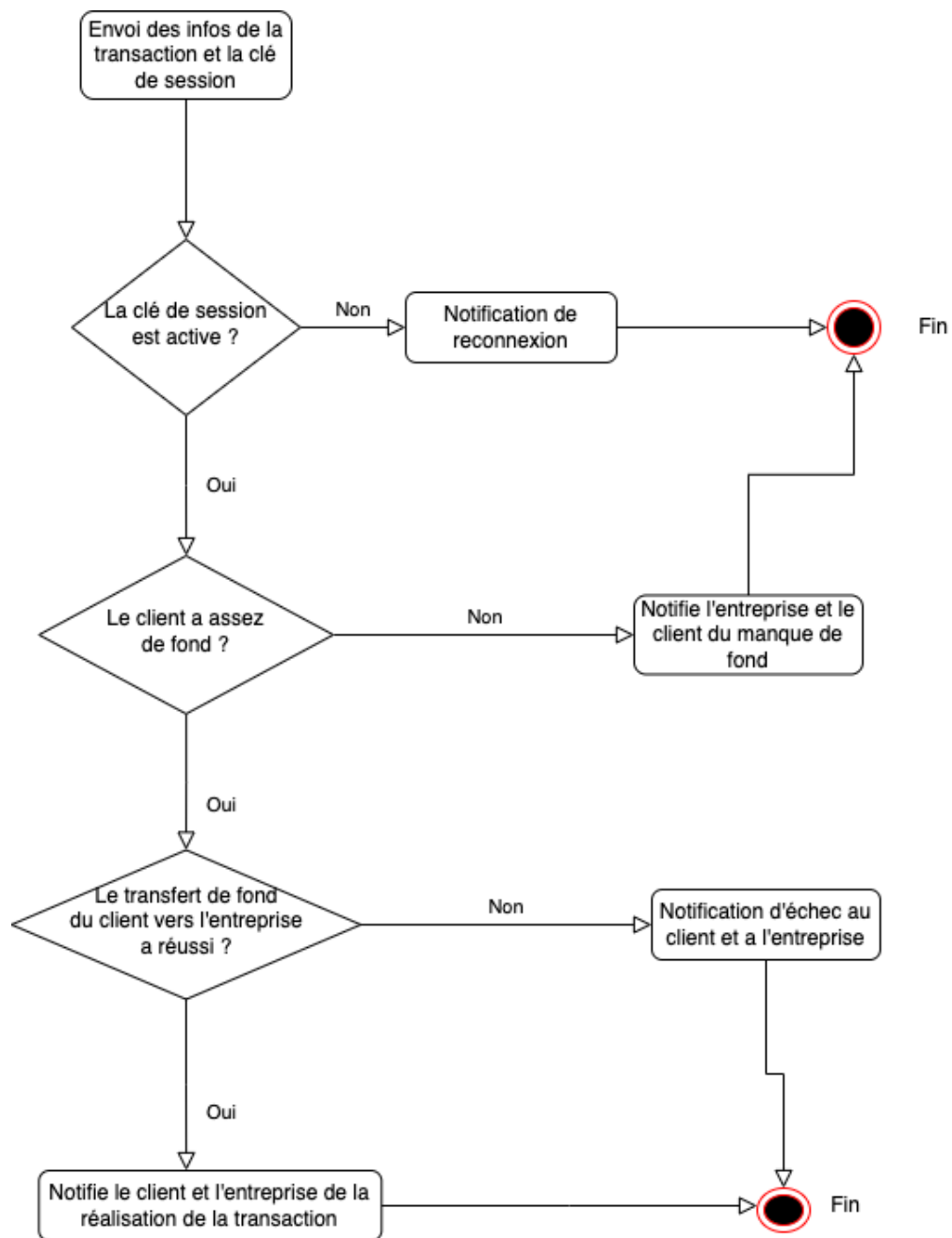


Figure 14 : Diagramme d'activité valider un paiement

## 5. Diagramme de classe

Le diagramme de classe est le diagramme le plus important de la modélisation orientée objet, il est le seul obligatoire lors de la modélisation d'un système. Il montre la structure interne d'un système en fournissant une représentation abstraite des objets du système qui vont interagir ensemble pour la réalisation des cas d'utilisation. C'est un graphe montrant les classes du système, les relations entre elles ainsi que les attributs et les opérations de ces classes.

Le diagramme de classe est utilisé généralement pour explorer les concepts du domaine, analyser les besoins et pour définir la conception détaillée du système.

Le graphique suivant représente le diagramme de classes.

## REFONTE ET CONCEPTION DU MODE DE PAIEMENT DE UniPay SAS

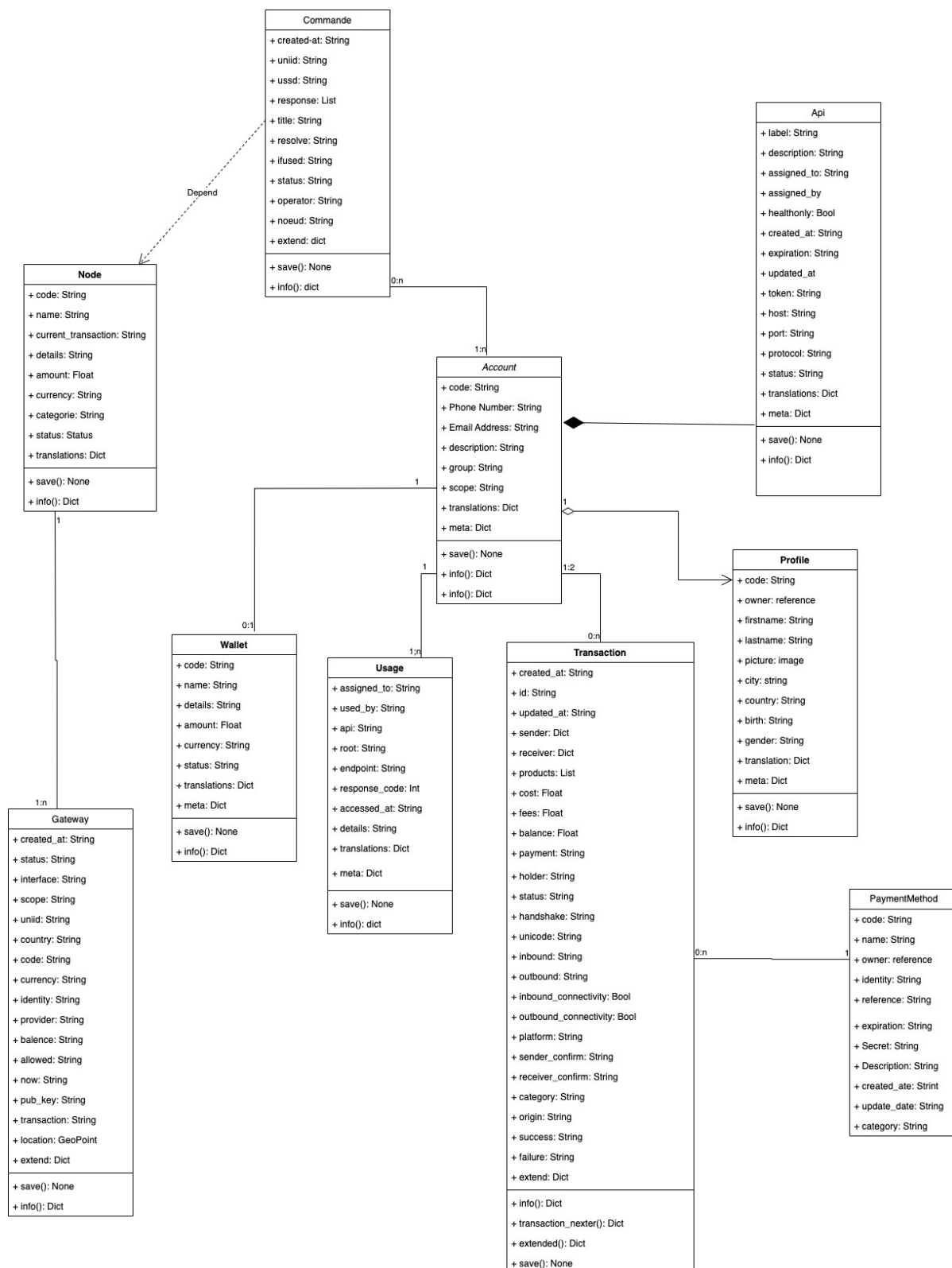


Figure 15 : Diagramme de classe

## CHAPITRE IV : MISE EN ŒUVRE DE L'APPLICATION

### I. Principe : Model-View-ViewModel (MVVM)

Dans le cadre de la conception de notre application web, nous avons adopté une approche basée sur le modèle MVVM (Model-View-ViewModel) pour structurer notre architecture. Cette approche nous a permis de séparer clairement la logique métier de la couche backend développée avec Flask, de l'interface utilisateur développée en JavaScript pour le front-end.

Le modèle MVVM a joué un rôle essentiel dans la conception de notre application, en nous offrant une structure claire et modulaire. La couche backend, représentée par le modèle, a été responsable de la gestion des données, des opérations de base de données et de la logique métier. Nous avons utilisé Flask, un framework Python, pour développer cette partie de l'application, en exploitant sa souplesse et sa richesse en fonctionnalités.

D'un autre côté, la couche front-end a été conçue en utilisant JavaScript pour créer l'interface utilisateur interactive. Nous avons utilisé JavaScript pour mettre en œuvre le modèle de vue (ViewModel) et faciliter la liaison des données entre la vue et le modèle. Cette couche s'est concentrée sur la présentation des données aux utilisateurs, les interactions utilisateur et la gestion des événements.

L'utilisation du modèle MVVM a apporté plusieurs avantages à notre projet. Tout d'abord, il a permis une séparation claire entre la logique métier et l'interface utilisateur, facilitant ainsi la maintenance et l'évolutivité de l'application. De plus, cela nous a permis de travailler de manière collaborative, car les développeurs backend et frontend pouvaient travailler simultanément sur leurs parties respectives de l'application, en se concentrant sur leurs responsabilités spécifiques.

En adoptant le modèle MVVM, nous avons également bénéficié d'une gestion plus efficace des données et des interactions, offrant ainsi une expérience utilisateur plus fluide et réactive. La liaison des données entre la vue et le modèle de vue a simplifié la gestion des mises à jour et a permis d'actualiser dynamiquement l'interface utilisateur en fonction des changements de données.

En conclusion, l'utilisation du modèle MVVM dans notre application web a permis une conception claire et modulaire, en séparant la logique métier de l'interface utilisateur. Cette approche a facilité la collaboration, amélioré la maintenance et offert une expérience utilisateur plus fluide.



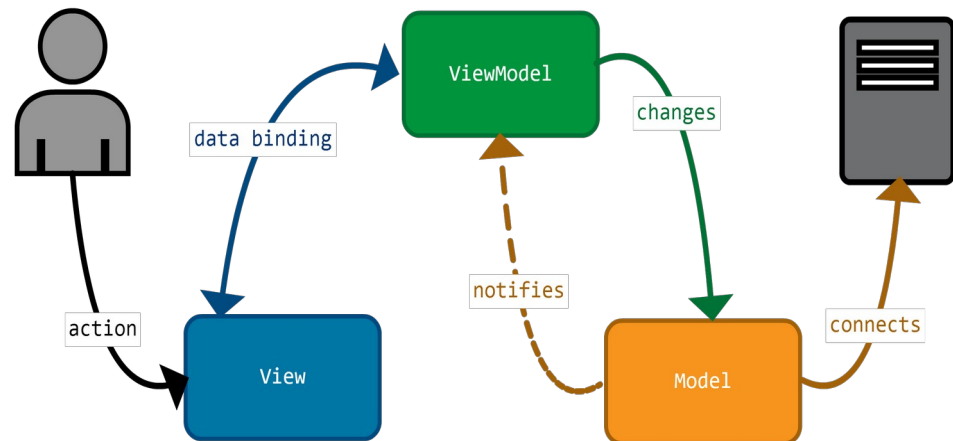


Figure 16 : Architecture MMVC

Source : <https://www.docdoku.com/blog/2015/02/17/architecturer-ses-applications-js-pattern-mvvm/>

## II. Outils technologiques

### 1. Langage informatique

Le langage informatique utilisé dans certains contextes dans le sens plus restrictif de langage de programmation, est un langage formel non nécessairement Turing-complet utilisé lors de la conception, la mise en œuvre, ou l'exploitation d'un système d'information.

### 2. Outils technologiques utilisés

Langages HTML, CSS, JavaScript et Python

## Langage HTML



Figure 17 : Logo HTML5

HTML, de son vrai nom HyperText Markup Language est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte. Il est l'une des trois inventions à la base du WWW, avec le Hypertext Transfert Protocol (HTTP) et les adresses web. HTML a été inventé pour permettre d'écrire des documents hypertextuels liant les différentes ressources d'Internet avec les hyperliens.

## ✓ Langage CSS



Figure 18 : Logo CSS3

Les feuilles de styles en cascade (en anglais Cascading Style Sheet) communément appelées CSS sont un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel le HTML.

## Langage JavaScript

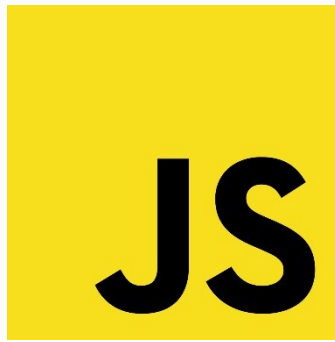


Figure 19 : logo javascript

Le JavaScript (en abrégé JS) est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs ; c'est un langage orienté objet à prototype. Créé en 1995 par Brendan Eich puis standardisé sous le nom d'ECMAScript en juin 1997 par Ecma International. JavaScript est un langage de programmation populaire utilisé principalement pour le développement web. Il est largement utilisé côté client pour rendre les sites web interactifs et dynamiques. JavaScript est un langage interprété, ce qui signifie qu'il est exécuté directement dans le navigateur sans nécessiter de compilation préalable.

Quelques-uns des plus grands projets réalisés avec JavaScript sont les suivants :

- ✓ React : React est une bibliothèque JavaScript développée par Facebook. Elle est utilisée pour créer des interfaces utilisateur réactives et interactives. React a été adopté par de nombreuses grandes entreprises, telles que Facebook, Instagram, Airbnb et Netflix, pour développer des applications web à grande échelle.
- ✓ Angular : Angular est un framework JavaScript open source développé par Google. Il est utilisé pour la création d'applications web dynamiques et évolutives. Angular a été largement utilisé dans le développement d'applications complexes, notamment par Google, Microsoft et IBM.
- ✓ Node.js : Node.js est un environnement d'exécution JavaScript côté serveur. Il permet d'exécuter du code JavaScript en dehors du navigateur, ce qui en fait un choix populaire pour la création de serveurs web et d'applications réseau. Node.js est utilisé par de grandes entreprises comme PayPal, Netflix et LinkedIn.

✓ Langage Python



Figure 20 : Logo python

Le langage Python est une interprétation de haut niveau à usage général langage de programmation. Créé par Guido van Rossum et publiée pour la première fois en 1991. Avec une philosophie de conception mettant l'accent sur la lisibilité du code grâce à son utilisation remarquable d'espaces significatifs, Python est dynamiquement typé et récupéré, prenant en charge la programmation procédurale, orienté objet et fonctionnelle.

Voici quelques-uns des plus grands projets réalisés avec Python :

- ✓ Django : Django est un framework web Python open source qui permet de développer rapidement des applications web robustes et sécurisées. Il est utilisé par de nombreux sites web et entreprises populaires, notamment Instagram, Pinterest et Disqus.
- ✓ TensorFlow : TensorFlow est une bibliothèque open source d'apprentissage automatique (machine learning) développée par Google. Elle est utilisée pour la création de modèles d'apprentissage automatique et de réseaux de neurones, et a été largement adoptée par la communauté de l'intelligence artificielle.
- ✓ Pandas : Pandas est une bibliothèque Python utilisée pour la manipulation et l'analyse de données. Elle fournit des structures de données et des fonctions puissantes pour le traitement des données, ce qui en fait un outil essentiel pour les scientifiques des données et les analystes.

✓ Langage NoSQL (MongoDB, Mongo Atlas)

Dans le cadre de ce projet, nous avons utilisé MongoDB, une base de données NoSQL, et nous avons choisi de stocker nos données sur MongoDB Atlas, une plateforme cloud de gestion de bases de données MongoDB.

Les bases de données NoSQL, telles que MongoDB, diffèrent des bases de données relationnelles traditionnelles en adoptant une approche plus flexible et évolutive pour le stockage et la gestion des données. Contrairement aux bases de données relationnelles qui utilisent des tables, des schémas prédéfinis et des requêtes SQL, les bases de données NoSQL utilisent un modèle de données plus souple et peuvent stocker des données non structurées, semi-structurées ou structurées de manière flexible.

MongoDB offre plusieurs avantages dans le contexte de notre projet. Tout d'abord, il permet une évolutivité horizontale, ce qui signifie que nous pouvons facilement augmenter la capacité de stockage en ajoutant de nouveaux serveurs à notre cluster MongoDB. Cela garantit que notre système peut gérer une charge croissante de données et de requêtes sans compromettre les performances.

De plus, MongoDB offre une grande flexibilité en termes de schéma de données. Contrairement aux bases de données relationnelles, où le schéma doit être défini à l'avance, MongoDB permet de stocker des données sans schéma fixe. Cela facilite l'ajout, la suppression ou la modification des champs de données sans avoir à effectuer de modifications coûteuses dans l'ensemble de la base de données.

En utilisant MongoDB Atlas, nous avons bénéficié des avantages de la gestion basée sur le cloud. MongoDB Atlas offre une infrastructure hautement disponible et gérée, ce qui signifie que nous n'avons pas à nous soucier de la maintenance de l'infrastructure sous-jacente. De plus, Atlas fournit des fonctionnalités de sauvegarde automatique, de récupération en cas de sinistre et de surveillance, ce qui garantit la sécurité et la disponibilité de nos données.

### **III. Frameworks**

#### **1. Qu'est-ce qu'un Framework**

En programmation informatique, un Framework (appelé aussi infrastructure logicielle ou cadre d'applications) désigne un ensemble cohérent de composants logiciels structurels, qui sert à

créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture). Un Framework vise à aider les développeurs dans leur travail. Son organisation vise la productivité maximale du développeur qui l'utilise quitte à baisser les coûts de production et de maintenance du programme développé.

Les principaux avantages des Framework sont la réutilisation de leur code dans d'autres projets, la standardisation du cycle de vie d'un logiciel et formalisation d'une architecture adaptée aux besoins de l'entreprise.

### **2. Frameworks utilisés**

#### **✓ Framework Flask**

Flask est un framework web léger et flexible qui a été créé en 2010 par Armin Ronacher. Contrairement à certains frameworks qui imposent une architecture spécifique, Flask adopte une approche "sans opinions" en offrant une grande liberté aux développeurs pour organiser leur code et concevoir leurs applications selon leurs préférences.

Cette architecture sans opinions de Flask permet aux développeurs de choisir les outils et les bibliothèques qui correspondent le mieux à leurs besoins, sans imposer une structure rigide. Flask fournit les fonctionnalités de base nécessaires au développement web, telles que la gestion des routes, des requêtes HTTP et des templates, tout en permettant aux développeurs de définir leur propre structure et d'adopter différents modèles de conception, tels que MVC (Modèle-Vue-Contrôleur) ou d'autres approches.

L'objectif principal de Flask est de rester simple, facile à comprendre et à utiliser. Sa conception minimaliste et son faible couplage font de Flask un choix populaire pour les petites et moyennes applications web, les prototypes ou les API. Il offre également une documentation complète, une communauté active et un écosystème d'extensions qui permettent d'ajouter facilement des fonctionnalités supplémentaires à l'application.

#### **✓ Framework Bootstrap.**

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. Bootstrap est compatible avec les versions récentes de la majorité des navigateur web existant. Mais peut fonctionner de manière dégradée sur des versions plus anciennes. De plus Bootstrap adopte le concept de site web adaptatif, permettant aux développeurs l'utilisant d'adapter dynamiquement leur projet au format des supports depuis lesquels ils sont accédés (smartphone, tablette, TV, PC, ...). Bootstrap a été inventé en 2010 par deux développeurs employés de Twitter Mark Otto et Jacob Thornton sous le nom de Twitter Blueprint. Depuis 2012, Twitter a placé Bootstrap sous licence open source. Il est l'un des Framework CSS les plus utilisés.

### ✓ Framework Apache Cordova

Apache Cordova, anciennement connu sous le nom de PhoneGap, est un framework open-source pour le développement d'applications mobiles multiplateformes. Il a été créé par Nitobi Software, qui a ensuite été racheté par Adobe Systems. Cordova permet aux développeurs d'utiliser des technologies web courantes telles que HTML5, CSS et JavaScript pour créer des applications mobiles pouvant être déployées sur plusieurs plateformes, telles qu'Android, iOS et Windows Phone.

L'objectif principal de Cordova est de simplifier le processus de développement d'applications mobiles en permettant aux développeurs de créer une application avec des technologies web familières, puis de l'emballer dans une coquille native pour accéder aux fonctionnalités natives du dispositif, telles que la caméra, les contacts, l'accéléromètre, etc. Ainsi, les développeurs peuvent créer une seule base de code pour plusieurs plateformes, réduisant ainsi le temps et les efforts nécessaires pour développer et maintenir des applications mobiles multiplateformes.

L'un des avantages majeurs d'Apache Cordova est sa compatibilité avec de nombreux frameworks JavaScript populaires tels que AngularJS, React, et jQuery Mobile, ce qui permet aux développeurs de tirer parti des outils et des bibliothèques existants. Cordova fournit également un ensemble d'API pour accéder aux fonctionnalités natives des appareils, ainsi que des plugins personnalisés pour étendre les fonctionnalités de base.

Grâce à sa nature open-source et à sa large communauté de développeurs, Apache Cordova bénéficie d'une documentation complète, de forums de discussion et de nombreuses ressources en ligne pour aider les développeurs à résoudre les problèmes et à améliorer leurs compétences. Il est largement utilisé pour le développement d'applications mobiles hybrides, offrant un équilibre entre la réutilisabilité du code et l'accès aux fonctionnalités natives des appareils.

### IV. Estimation des couts

#### 1. Méthode de l'estimation par points de complexité et coût de prestation de l'équipe

Pour l'évaluation du coût de prestation de l'équipe, nous utiliserons la méthode de l'estimation par points de complexité. Cette approche se base sur l'évaluation de la complexité relative des fonctionnalités à développer, plutôt que sur une estimation de temps fixe. Voici comment cela fonctionne :

1. Établir une échelle de complexité : Créez une échelle de référence pour évaluer la complexité des fonctionnalités, par exemple de 1 à 10, où 1 représente une fonctionnalité simple et 10 une fonctionnalité très complexe.
2. Organiser des sessions d'estimation : Réunissez l'équipe de développement, les responsables du projet et les parties prenantes pour évaluer chaque fonctionnalité. L'équipe discute de la complexité de chaque fonctionnalité en se basant sur l'échelle établie.
3. Attribuer des points de complexité : Après discussion, attribuez à chaque fonctionnalité un nombre de points correspondant à sa complexité. Par exemple, une fonctionnalité simple pourrait être évaluée à 1 ou 2 points, tandis qu'une fonctionnalité complexe pourrait être évaluée à 8 ou 10 points.
4. Calculer la vélocité de l'équipe : Sur la base des itérations précédentes, calculez la vélocité de l'équipe, c'est-à-dire le nombre de points qu'elle est capable de traiter par itération.
5. Estimer le coût : Multipliez le nombre total de points de complexité par l'estimation de la vélocité de l'équipe pour obtenir une estimation du nombre d'itérations nécessaires pour terminer le projet. Ensuite, multipliez ce nombre par la durée d'une itération pour estimer la durée totale du projet.



Voici la formule pour estimer le coût d'un projet Agile avec cette méthode :

**Coût du projet = (Nombre total de points de complexité \* Estimation de la vitesse) \*  
Durée d'une itération**

En utilisant cette méthode, vous pouvez obtenir une estimation du coût du projet basée sur la complexité des fonctionnalités et la vitesse de l'équipe de développement. Cela permet d'obtenir une estimation plus réaliste et adaptable, en prenant en compte les incertitudes et en permettant des ajustements tout au long du projet.

D'après notre étude nous avons répertorié 26 fonctionnalités majeures pour ce projet et estimons qu'il faut en moyenne 3 jours pour chaque fonctionnalité ce qui donne la durée d'une itération (nombre de temps nécessaire pour réaliser une fonctionnalité) à 3 (en nombre de jours)

Dans ce projet nous estimons le nombre total de complexité à 106 et coût d'une unité à 5.000 FCFA.

La vitesse quand elle est nombre total de points de complexité divisé par le nombre total d'itération qui est de  $106/8.66 = 12.2$

Finalement **Coût du projet =  $106 * 12.2 * 3 * 5000$  FCFA = 19.398.000 FCFA**

## **2. Coût du matériel de développement**

La conception de notre application a nécessité un coût en matériels et en logiciels énumérés dans le tableau suivant :

| Description                               | Quantité | Coût (F cfa)         |
|---|----------|----------------------|
| Flask, Apache Cordova                     |          | 00                   |
| Python, Javascript                        |          | 00                   |
| Bootstrap                                 |          | 00                   |
| Ordinateur (256 SSD, RAM 8GB, CPU 2.8GHz) | 01       | 800 000              |
| Raspberry PI 4                            | 04       | 152 000              |
| Forfait internet                          |          | 10 000/mois          |
| <b>Total</b>                              |          | 952.000 +10 000/mois |

Tableau 9 : Coût du matériel de développement

### 3. Coût du matériel de déploiement

| Description      | Quantité | Coût (\$)            |
|------------------|----------|----------------------|
| Miscrosoft Azure | 01       | 34/mois              |
| Mongo Atlas      | 01       | 25/mois              |
| Google PlayStore | 01       | 25 a vie             |
| Apple Store      | 01       | 99/an soit 8.25/mois |
| <b>Total</b>     |          | 25 + 67.25/mois      |

Tableau 10 : Coût du matériel de déploiement

## V. Mise en œuvre

### 1. Environnement matériel de développement

Pour la réalisation de ce projet, j'ai travaillé avec un iMac 2019. Cette machine offre des caractéristiques matérielles puissantes qui ont facilité le développement et le déploiement de l'application. L'iMac 2019 est équipé d'un processeur Intel Core i5 ou i7, offrant des performances rapides et efficaces pour exécuter les tâches de développement. De plus, il dispose d'une mémoire RAM généralement de 8 Go à 32 Go, ce qui a permis d'exécuter les différents outils et logiciels nécessaires à la création de l'application sans ralentissements.

Au niveau logiciel, l'iMac 2019 est compatible avec le système d'exploitation macOS, offrant une interface utilisateur conviviale et une grande compatibilité avec les outils de développement. J'ai utilisé plusieurs logiciels essentiels tels que Insomnia, qui m'a permis de déboguer et tester l'application avec efficacité. J'ai également utilisé des éditeurs de texte tels que Visual Studio Code pour travailler sur le front-end en JavaScript et HTML sachant que tout était conteneuriser avec docker pour rendre le déploiement beaucoup plus facile.

L'écran Retina de l'iMac 2019 avec sa résolution élevée et ses couleurs précises a été un atout majeur pour la conception de l'interface utilisateur. La taille de l'écran, généralement entre 21,5 pouces et 27 pouces, a fourni un espace de travail confortable pour visualiser et organiser les différents composants de l'application.

### 2. Environnement logiciel

#### ✓ Insomnia

Insomnia est un outil populaire et puissant pour le développement d'API. C'est une application de bureau multiplateforme qui permet aux développeurs de tester, de déboguer et de gérer les appels d'API de manière efficace. Grâce à son interface conviviale et intuitive, Insomnia simplifie le processus de création et de modification de requêtes HTTP, de gestion des en-têtes et des paramètres, et de visualisation des réponses. Il offre également des fonctionnalités avancées telles que l'enregistrement et la reproduction de requêtes, la gestion des environnements, les collections de requêtes et la synchronisation avec des services cloud. En utilisant Insomnia, j'ai pu tester facilement les fonctionnalités de notre API, vérifier les réponses et effectuer des ajustements en temps réel pour assurer le bon fonctionnement de notre application.



Figure 21 : Logo Insomnia

### ✓ Docker

Docker est une plateforme open-source qui facilite le déploiement d'applications dans des environnements conteneurisés. Il permet d'emballer une application avec toutes ses dépendances et configurations nécessaires dans un conteneur léger et isolé. En utilisant Docker, j'ai pu créer des conteneurs pour notre application, ce qui garantit une portabilité et une reproductibilité élevées, quel que soit l'environnement d'exécution. Docker fournit également une gestion simplifiée des ressources et des réseaux, ce qui permet d'orchestrer facilement plusieurs conteneurs pour créer des environnements de développement, de test et de production cohérents. L'utilisation de Docker a considérablement simplifié le déploiement de notre application, en éliminant les problèmes liés aux différences d'environnement et en assurant une isolation entre les différents composants. De plus, Docker offre une grande flexibilité, en permettant l'intégration avec d'autres outils et services tels que Kubernetes pour la gestion des conteneurs à grande échelle. En résumé, l'utilisation de Docker a grandement simplifié le processus de déploiement de notre application, améliorant ainsi l'efficacité et la portabilité de notre développement.

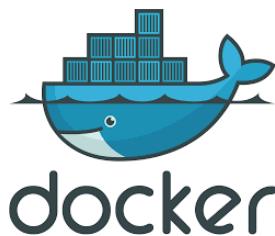


Figure 22 : Logo Docker

### ✓ Diagram.io

Diagrams.io est un outil en ligne gratuit qui permet de créer des diagrammes UML de manière intuitive et visuelle. Il offre une interface conviviale avec une large gamme d'éléments de diagramme prédéfinis, tels que les classes, les relations, les cas d'utilisation, les séquences, et bien d'autres encore. Grâce à ses fonctionnalités avancées, comme le glisser-déposer, l'alignement automatique, les connexions intelligentes et la personnalisation des

styles, Diagrams.io facilite la création de diagrammes UML professionnels et esthétiquement plaisants. En utilisant Diagrams.io, j'ai pu représenter visuellement la structure, les relations et le flux de notre système, ce qui a permis une meilleure compréhension et communication des concepts clés du projet. De plus, Diagrams.io permet de sauvegarder les diagrammes dans le cloud ou de les exporter sous différents formats, facilitant ainsi le partage et l'intégration dans la documentation et les présentations. L'utilisation de Diagrams.io a grandement simplifié la création de nos diagrammes UML, nous permettant de visualiser et de modéliser efficacement notre architecture logicielle.



Figure 23 : Logo Diagrams.io

### ✓ MongoDB

MongoDB est un système de gestion de base de données NoSQL orienté documents. Il stock les données au format BSON qui est le format binaire de JSON. Il ne nécessite aucun schéma prédéfini. MongoDB peut être distribué sur plusieurs serveurs (partitionnement horizontal ou sharding). Il existe en version community edition et en version enterprise edition. Il est écrit en C++, Go, Javascript et python.



Figure 24 : Logo mongoDB

## 4. Politique de sécurisation

La politique de sécurisation de notre application repose sur des mesures solides visant à protéger les données sensibles et à prévenir les menaces potentielles. La sécurité des applications revêt une importance cruciale dans un environnement où les transactions

électroniques et les échanges de données sont monnaie courante. Afin de garantir un niveau élevé de sécurité, nous avons mis en place plusieurs statuts d'accès avec des autorisations spécifiques.

Tout d'abord, nous avons le statut "Wallet" qui correspond au portefeuille électronique, permettant aux utilisateurs d'effectuer des transactions financières sécurisées. Les utilisateurs disposant de ce statut ont un accès limité aux fonctionnalités liées aux transactions financières, garantissant ainsi la confidentialité et l'intégrité des données financières.

Ensuite, nous avons le statut "Entreprise" qui représente les entreprises utilisant notre application pour leurs transactions. Les entreprises bénéficient d'une API sécurisée pour effectuer des transactions, avec des autorisations spécifiques pour accéder aux fonctionnalités de l'application en lien avec leurs activités commerciales.

Le statut "Administrateur" est attribué aux utilisateurs qui ont des privilèges étendus. Ils ont la possibilité de consulter, modifier et ajouter des données, mais ils n'ont pas la permission de supprimer des informations critiques. Cela garantit la cohérence des données et évite les suppressions accidentelles ou malveillantes.

Enfin, le statut "Root" est réservé aux utilisateurs ayant les droits les plus élevés. En plus des privilèges de l'administrateur, ils ont la capacité de supprimer toutes les données de l'application. Cependant, l'accès à ce statut est restreint et réservé à un nombre limité de personnes de confiance, afin d'éviter tout abus ou erreurs potentielles.

En adoptant cette approche de sécurisation par statut, nous avons pu mettre en place des contrôles d'accès et des autorisations granulaires, garantissant la confidentialité, l'intégrité et la disponibilité des données. Ces mesures de sécurité renforcées permettent d'éviter les accès non autorisés, les modifications indésirables et les suppressions accidentelles, assurant ainsi un environnement sécurisé et fiable pour nos utilisateurs.

### **5. Présentation de quelques interfaces**

L'interface que nous avons conçu est multi-langue (en français, anglais, espagnole, italien, portugais et arabe).

Une ou deux partie des interfaces seront présenter en 2 langues pour illustrer cela.

# REFONTE ET CONCEPTION DU MODE DE PAIEMENT DE UniPay SAS

## ➤ Modifier une passerelle (en anglais)

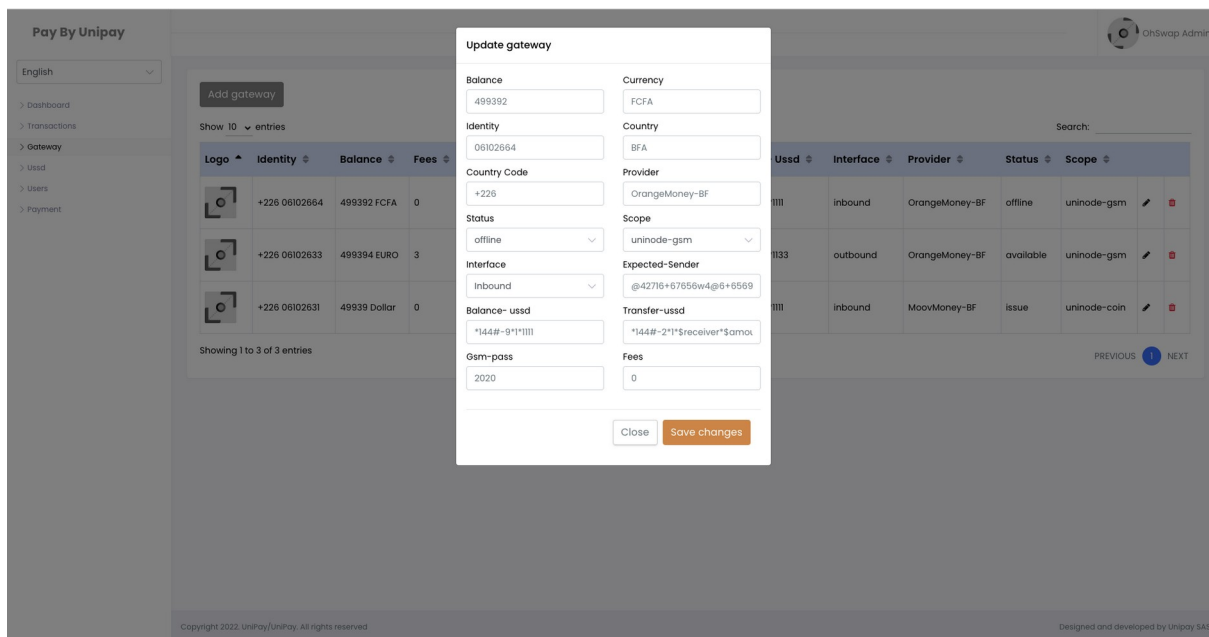


Figure 27 : modifier une passerelle

## ➤ Tableau de bord (Root)

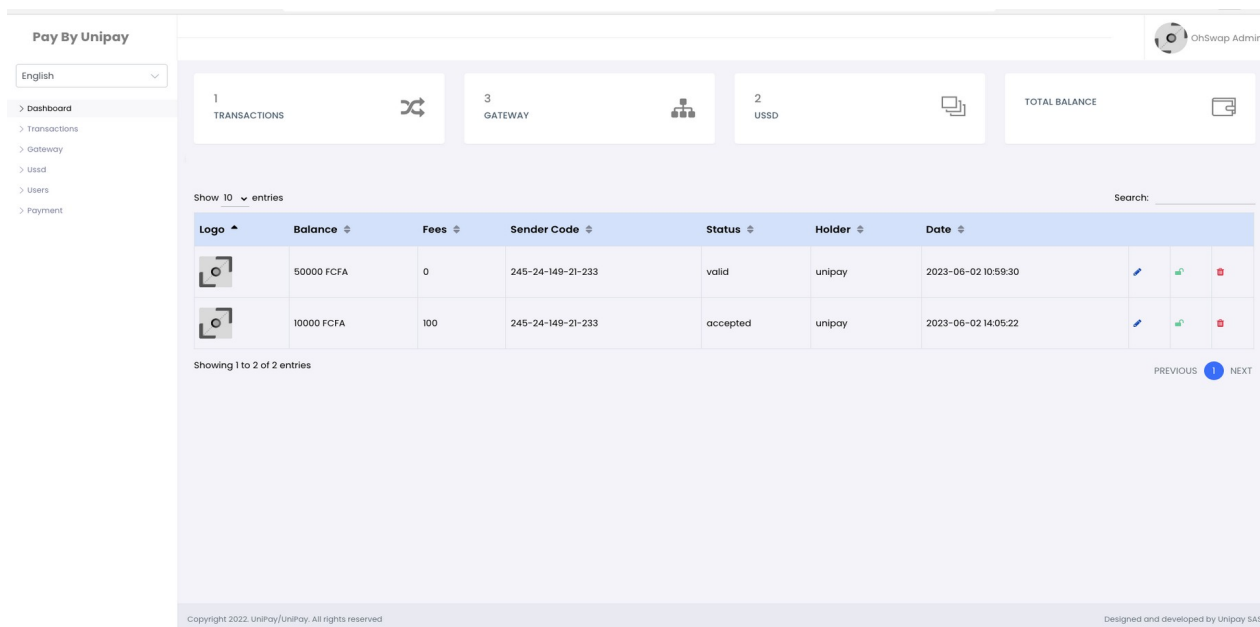


Figure 28 : Tableau de bord

# REFONTE ET CONCEPTION DU MODE DE PAIEMENT DE UniPay SAS

## ➤ Générer API (Entreprise)

The screenshot displays the UniPay Admin interface with a modal form titled "Mise à jour de l'API". The modal contains the following fields and controls:

- Nom de l'API \***: A text input field.
- Statut de l'API \***: A dropdown menu with "Online" selected.
- Pourquoi voulez-vous l'API \***: A large text area for justification.
- Entrez le lien de notification**: A text input field.
- Fermer**: A button to close the modal.
- Obtenir l'API**: An orange button to generate the API.

The background interface shows a sidebar with navigation links (Accueil, Transactions, Virement, API, Client), a language selector (Français), and a table of API entries with columns for Logo, Nom De L'API, Hôte, and Expiration. The footer includes copyright information for UniPay/Unipay and a note about development by UniPay SAS.

Figure 29 : Générer API

## ➤ Info de l'API (Enterprise)

This screenshot shows the same "Mise à jour de l'API" modal form, but with additional information displayed:

- Créé le**: A date field showing "2023-06-02 09:17:27 (Z)216".
- L'API sera active \***: A dropdown menu with "Oui" selected.
- Statut**: A dropdown menu with "Online" selected.
- Clé d'API**: A text field displaying a long alphanumeric string: "c3cld04ac30209728dd13418c3967b6287ceef87b852eeff0c4024".
- Enregistrer les modifications**: An orange button to save the changes.

The background interface remains the same, showing the API management table and sidebar navigation.

Figure 30 : Mise a jour et info de l'API entreprise



## ➤ Interface du mobile

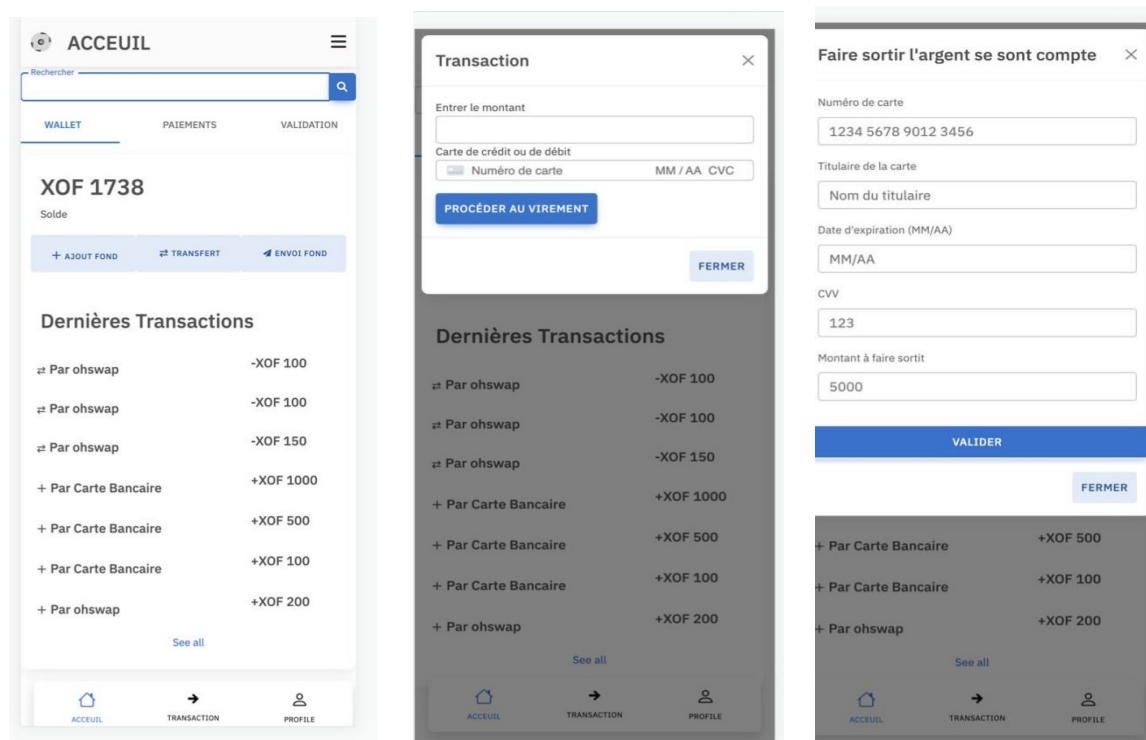


Figure 31: Interface du mobile

## VI. Problèmes rencontrés

Au cours du stage, j'ai été confronté à plusieurs difficultés qui ont nécessité des ajustements et des efforts supplémentaires. L'une des principales difficultés que j'ai rencontrées était liée au code JavaScript existant que j'ai trouvé dans le projet. Les indentations et la structure du code ne correspondaient pas à ma propre façon de coder, ce qui rendait la lecture et la maintenance du code plus difficile. De plus, les requêtes étaient effectuées à l'aide de l'ancienne API XMLHttpRequest de JavaScript, tandis que j'utilisais habituellement la méthode Fetch, plus récente et plus conviviale. Pour remédier à cette situation, j'ai dû réécrire une grande partie du code afin de l'adapter à mes préférences et d'améliorer sa lisibilité et sa compatibilité avec les normes actuelles.

Une autre difficulté majeure à laquelle j'ai été confronté était liée à l'obtention des API nécessaires pour les transactions axées sur les opérateurs téléphoniques, conformément aux fonctionnalités initiales de l'entreprise. Malheureusement, nous avons rencontré des problèmes

importants pour obtenir les API auprès des fournisseurs, notamment avec Orange. Cette situation nous a contraints à revoir notre approche et à orienter nos transactions vers les paiements par carte bancaire en collaborant avec Stripe. Bien que cette transition ait permis de résoudre les problèmes liés aux API des opérateurs téléphoniques, cela a également nécessité une adaptation et une refonte significatives de certaines parties de notre solution.

Ces difficultés rencontrées durant le stage m'ont permis d'acquérir une expérience précieuse dans la résolution de problèmes techniques et d'adaptation aux contraintes du projet. J'ai appris à faire face aux situations imprévues, à remettre en question et à améliorer le code existant, ainsi qu'à trouver des alternatives viables lorsque les solutions initiales se sont avérées difficiles à mettre en place. Ces défis ont renforcé mes compétences en développement et m'ont permis de développer ma capacité d'adaptation et de recherche de solutions créatives face à des problèmes complexes.

## CONCLUSION ET PERSPECTIVES

En conclusion, ce projet de refonte et de conception du mode de paiement pour UniPay SAS a été une expérience enrichissante et réussie. Nous avons relevé les défis liés aux achats en ligne en fournissant une solution complète, tout en mettant en avant des valeurs telles que l'agilité, l'innovation et la durabilité. Grâce à notre travail d'équipe, nous avons pu réaliser une application fonctionnelle, conviviale et fiable, qui répond aux besoins des utilisateurs et des entreprises dans le domaine du e-commerce. Ce projet a été une opportunité de mettre en pratique nos connaissances et compétences en informatique, et nous sommes fiers du résultat obtenu.

En termes de perspectives, UniPay SAS envisage d'élargir ses services en explorant le domaine des cryptomonnaies. La popularité croissante des cryptomonnaies offre de nouvelles opportunités dans le domaine des paiements en ligne, offrant des transactions rapides et sécurisées. UniPay souhaite explorer cette avenue et intégrer des fonctionnalités de paiement basées sur les cryptomonnaies, offrant ainsi une alternative aux modes de paiement traditionnels.

Par ailleurs, UniPay reconnaît les limitations actuelles liées à l'envoi d'argent par carte bancaire, notamment en raison des restrictions imposées par des fournisseurs de services tels que Stripe. Dans le but de garantir une plus grande flexibilité et une meilleure expérience utilisateur, UniPay envisage de développer son propre mode d'envoi d'argent ou de collaborer avec des partenaires offrant des solutions d'envoi d'argent plus complètes. Cela permettrait d'élargir les possibilités de paiement offertes aux utilisateurs et aux entreprises, tout en assurant une plus grande adaptabilité aux besoins du marché.

En poursuivant ces perspectives, UniPay SAS renforce son engagement à rester à l'avant-garde de l'innovation dans le domaine des services financiers en ligne. En élargissant ses fonctionnalités de paiement, en explorant les opportunités offertes par les cryptomonnaies et en recherchant des partenariats stratégiques, UniPay vise à rester compétitif sur le marché et à continuer de fournir des solutions de paiement fiables et adaptées aux besoins changeants des utilisateurs et des entreprises.

Cette volonté d'évolution et d'adaptation témoigne de la vision d'UniPay SAS en tant qu'entreprise qui vise à anticiper les tendances du marché, à offrir des solutions de paiement innovantes et à rester un acteur majeur dans le domaine du e-commerce et des services financiers en ligne en Afrique de l'Ouest, notamment au Burkina Faso.

## BIBLIOGRAPHIE

- [1] OUILY Hamed Joseph, “ Développement d’une application web d’extraction et de structuration de textes juridiques à partir de fichiers PDF et images. ”, Rapport de stage, Département Informatique, UFR/SEA, UJKZ, Ouagadougou, 2017.
- [2] KOALA Gouayon, “Mise en place d’une plateforme web/sms d’information, d’achat et de vente de produits agro-sylvo-pastoraux”, Rapport de stage, Département Informatique, UFR/SEA, UJKZ, Ouagadougou, 2015.

## WEBOGRAPHIE

- [3] Wikipédia. (10.03.2021). Framework --- Wikipédia [En ligne]. <https://fr.wikipedia.org/wiki/Framework>.
- [4] (10.03.2021). Introduction Bootstrap [En ligne]. <https://getbootstrap.com/docs/5.0/getting-started/introduction/>.
- [5] Wikipédia. (12.03.2021). Mongo Engine [En ligne]. <https://mongoengine.org/>.
- [6] jQuery. (12.03.2021). JQuery API | Documentation [En ligne]. <https://flask.palletsprojects.com/>.
- [7] Wikipédia. (13.03.2021). L’outils de conception de diagramme de tout genre. [En ligne]. <https://diagrams.io>.
- [8] Wikipédia. (28.02.2021). Mongo atlas [En ligne]. <https://www.mongodb.com/atlas/database>.
- [9] Wikipédia. (03.03.2021). Insomnia [En ligne]. <https://insomnia.rest/download>.
- [10] (08.03.2021). Documentation de apache cordova [En ligne]. <https://cordova.apache.org/docs/en/11.x/https://cordova.apache.org/docs/en/11.x/>

- [11] (09.02.2021). Architect Enterprise [En ligne]. <https://stripe.com/>.
- [12] Wikipédia. (23.02.2021) Mozilla doc for javascript [En ligne]. <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- [16] Microsoft azure [En ligne]. <https://azure.microsoft.com/fr-fr>
- [17] (04.02.2021) Docker [En ligne]. <https://www.docker.com/>.