

RÉPUBLIQUE DU BÉNIN

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

ECOLE NATIONALE D'ECONOMIE
APPLIQUEE ET DE MANAGEMENT



DEPARTEMENT D'INFORMATIQUE DE GESTION

Option: Analyse Informatique et Programmation

MEMOIRE DE FIN DE FORMATION
POUR L'OBTENTION DE LA LICENCE

Thème :

Réalisation d'un agrégateur de paiement : KKiaPay

Mélaine KIOSSOU

melainekiossou@gmail.com

Sous la supervision de :

Dr Ing. Vinasétan Ratheil HOUNDJI

M.Shadai ALI

Année Académique : 2017-2018

Table des matières

Dédicace	iii
Remerciements	iv
Liste des figures	v
Liste des sigles et abréviations	vi
Résumé	1
Abstract	2
Introduction	3
1 Revue de littérature	6
1.1 Les agrégateurs de paiement	6
1.2 Acteurs clés du marché	9
2 Matériel et méthodes	12
2.1 Analyse	12
2.2 Langage de modélisation	14
2.2.1 MERISE	14
2.2.2 UML	15
2.3 Conception	15
2.3.1 Modélisation fonctionnelle	16
2.3.2 Modélisation statique	19
2.3.3 Modélisation dynamique	22
2.4 Choix techniques	24
2.4.1 Moteur de base de données	24
2.4.2 Technologies utilisées	26
2.4.3 Architectures logicielles	27
2.4.4 Architecture du système	28
2.4.5 Sécurité de l'application	28

3 Résultats et discussions	31
3.1 Application web KKiaPay	31
3.2 Discussion	36
Conclusion et perspectives	37
Bibliographie	38
Webographie	39

Dédicace

Toutes les lettres ne sauraient trouver les mots qu'il faut. Tous les mots ne sauraient exprimer la gratitude, le respect et la reconnaissance. Aussi, c'est tout simplement que je dédie ce mémoire à ceux que j'ai choisi d'appeler ma famille.

Remerciements

Nous tenons à exprimer notre gratitude et nos sincères remerciements à tous ceux qui ont participé de manière directe ou indirecte à la réalisation de ce projet. Nos remerciements vont particulièrement à l'endroit de :

- Mme Rosaline D. Worou-Houndékon, Professeur Agrégée des Sciences de Gestion, Directrice de l'ENEAM;
- M.Théophile K.Dagba, PhD., Maître de Conférences, Directeur-adjoint, Chargé des affaires académiques de l'ENEAM;
- M.Ratheil HOUNDJL, pour son regard éclairant et son accompagnement tout au long de la rédaction de ce document
- M.Gilles KOUNOU, PhD., qui nous a accordé la possibilité d'effectuer un stage de trois mois au sein de son cabinet;
- M. Shadai ALI, notre maître de stage et ses collaborateurs pour leur sollicitude et leurs conseils tout au long de notre stage;
- Mesdames Sandrine BOCCO et Doria REY pour leur accompagnement et assistance tout au long de notre stage;
- Messieurs Léonel ZEGUE, Latif BUSSARI et Merveil EZIN pour leur aide et accompagnement tout au long de notre stage;
- Toute l'équipe de la société Open SI pour leur accueil chaleureux, pour le soutien et l'appui qu'ils nous ont témoignés tout au long de ce stage;
- Tout le personnel enseignant et administratif de l'ENEAM pour leur accompagnement durant notre formation;
- Tous ceux qui de près ou de loin nous ont apporté leur soutien sous quelque forme que ce soit.

Table des figures

2.1	Diagramme de cas d'utilisation	17
2.2	Diagramme de classe	20
2.3	Diagramme d'objet	22
2.4	Diagramme de séquence - Cas d'utilisation " <i>Rembourser une transaction</i> "	23
2.5	Diagramme d'activité - Cas d'utilisation " <i>Ajouter un compte mobile money</i> "	24
3.1	Interface de paiement unique de KKiaPay - <i>KKiaPay Widget</i>	33
3.2	Capture d'écran du menu <i>Transactions</i> de KKiaPay	34
3.3	Point de lumière sur le possible téléchargement des données KKiaPay	35

Liste des sigles et abréviations

IP : Internet Protocol

GSMA : GSM Association

KYC : Know Your Customer

PCI : Payment Card Industry

PHP : Hypertext Preprocessor

MPOS : Mobile Point Of Sale

IFU : Identifiant Fiscal Unique

MVC : Modèle-Vue-Contrôleur

RIB : Relevé d'Identité Bancaire

GSM : Global System for Mobile

JSON : JavaScript Object Notation

URI : Uniform Resource Identifier

SQL : Structured Query Language

UML : Unified Modeling Language

W3C : World Wide Web Consortium

CMS : Content Management System

TCP : Transmission Control Protocol

SOA : Service Oriented Architecture

BSD : Berkeley Software Distribution

PME : Petites et Moyennes Entreprises

MID : Manufacturer Identification Code

API : Application Programming Interface

BSON : Binary JavaScript Object Notation

CRM : Customer Relationship Management

ANSI : American National Standards Institute

SGBD : Système de Gestion de Base de Données

RCCM : Registre du Commerce et du Crédit Immobilier

MERISE : Méthode d'Etude et de Réalisation Informatique

Résumé. Face au potentiel de croissance très prometteur des nouveaux services de paiement mobile en ligne, de nouvelles sociétés bourgeonnent un peu partout en Afrique francophone. En effet, surfant sur l'ascension fulgurante du mobile money, elles ont pour but de fédérer les différentes solutions de paiement mobile afin de permettre aux consommateurs de payer leurs biens et services, en ligne ou en physique. Leur présence témoigne d'un besoin fort, d'une population majoritairement dépourvue de comptes bancaires, usagère de téléphones mobiles et cliente de l'une ou l'autre des solutions de paiement mobile offertes par les opérateurs de téléphonie mobile, à l'instar de Mobile Money de MTN, Flooz Money de Moov.

Ce travail se veut offrir aux populations et aux entreprises une plateforme leur permettant de recevoir des paiements par Mobile Money, carte de crédit et compte bancaire en toute sécurité. L'infrastructure solide et flexible ainsi indexée se veut fournisseuse d'une expérience de traitement de paiement unique et simplifiée. **Mots clés :** *agrégateur de paiement, KKiaPay, mobile money, transaction.*

Abstract. Given the very promising growth potential of new online mobile payment services, new companies are sprouting up everywhere in French-speaking Africa. Surfing on the meteoric rise of mobile money, they aim to federate different mobile payment solutions to allow consumers to pay for their goods and services, online or in physics. Their presence testifies to a strong need, a population largely without bank accounts, mobile phone user and customer of any mobile payment solutions, offered by mobile operators, like MTN Mobile Money, Flooz Money from Moov.

This work wants to offer the populations and the companies a platform allowing them to receive payments by Mobile Money, credit card and bank account in all security. The robust and flexible indexed infrastructure is intended to provide a single, simplified payment processing experience.

Keywords: *payment aggregator, KKiaPay, mobile money, transaction.*

Introduction

En Afrique, et au Bénin de façon spécifique, il est donné l'impression qu'en raison de l'inaccessibilité des divers services de paiement en ligne internationaux par la grande majorité beaucoup d'opportunités d'affaires ne sont pas à la portée des autochtones. En effet, ces derniers sont incapables d'accepter la contrepartie que constituent les paiements des biens et services qu'ils offrent sur un même pied d'égalité que le reste du monde. Ceci en ayant recours aux solutions de paiement propres à nos réalités. Selon la rubrique économie du web media sLe Point Afrique, le montant des transactions par mobile money ces dernières années dans la zone ouest-africaine s'élèverait à 5000 milliards de francs CFA avec un marché de 250 millions d'utilisateurs des services de paiements mobiles d'ici 2019 en Afrique subsaharienne [15]. Dans un tel contexte où les paiements mobiles constituent une part aussi importante du marché des transactions financières, l'absence de solution de paiement en ligne prenant en compte le mobile money, est un handicap pour l'essor du e-business en Afrique.

Problématique

Le rapport 2018 de la GSMA sur le mobile money, pousse l'Afrique sur le devant de la scène comme étant le plus grand marché du monde en matière de mobile money avec 48% d'abonnés. Il met en lumière la sécurité que procure cette monnaie et la simplicité liée à son usage comme étant des facteurs justifiant son adoption massive [6]. À hauteur de 97,44% de la valeur totale des transactions, le transfert de fonds (de particulier à particulier, opérations de dépôt et de retrait) constitue le principal usage qui en est fait par les ménages devant les opérations de paiement de biens et services estimées à 2,66% de la valeur totale des transactions [24]. Il ressort donc de ce constat que le développement d'un ensemble d'offres facilitant les paiements contribuera de façon positive au développement de l'écosystème et à l'inclusion générale du mobile money.

Cet état de fait laisse à penser qu'offrir un moyen centralisé de collecte de paiements pour les marchands de bien et services constitue une étape importante dans la construction d'un écosystème de paiement viable. Ces solutions de paiement se doivent d'être simples, moins contraignantes, fiables, sécurisées et transparentes aussi bien pour les services de collecte que

pour le contribuable payeur. En effet, l'une des contraintes majeures liées à leur mise en place est de s'assurer qu'ils soient non seulement inclusifs mais également adaptés du point de vue économique, aux différentes couches de la population. La quasi ubiquité des solutions de paiement via Mobile Money et le fort taux de pénétration du Mobile représente à bien des égards, une voie à explorer pour répondre aux préoccupations ci-dessus énoncées.

Motivation

Disposer d'une solution de paiement en ligne intégrant le mobile money afin d'effectuer des transactions, s'avère en Afrique subsaharienne un essentiel. A l'heure actuelle, nombreuses sont les solutions de paiement en ligne à l'international, qui intègrent peu ou pas souvent les modes de paiement en vigueur dans la sous-région. C'est le cas de PayPal qui permet certes à un client d'effectuer des paiements mais qui toutefois interdit à un marchand localisé en Afrique de recevoir de l'argent ou d'en retirer. Afin de bénéficier pleinement de ces systèmes internationaux, ce dernier est donc obligé de créer des comptes bancaires dans les pays d'Europe ou d'Amérique. De plus, les solutions de paiement existantes sur le continent, conçues pour la plupart à l'image des modèles de paiement en ligne ayant fait leur preuve dans les pays développés où le système d'adresses physiques est bien organisé, les rues bien identifiées et les adresses existantes, ne tiennent guère compte des réalités de l'Afrique car demandant en effet de fournir avant toute transaction des informations (code postal, adresse ou numéro de rue) presque inexistantes en Afrique.

Objectifs

L'objectif principal de ce travail est de mettre en place une solution d'agrégation de divers services de paiement mobile disponibles en République du Bénin qui, une fois intégrée dans une plateforme/application de collecte d'argent ou de e-commerce, offre nativement la possibilité d'utiliser aussi bien MTN Mobile Money, Moov Money, les cartes bancaires Visa ou MasterCard ou ECOBANK Xpress pour régler des paiements de manière électronique. L'agrégation effectuée permet également au service de collecte de procéder à des versements au contribuable (remboursement, rollback d'une transaction, paiement d'une prestation à un contribuable). Il s'agit d'offrir :

- au client final (ici le contributeur payeur) une expérience de paiement homogène et simple ;
- aux développeurs une intégration facilitée ;
- au service de collecte, une gestion aisée de la relation avec le contributeur au moyen d'outils de suivi de transactions ;
- une sécurité et une gestion de la fraude accrues, venant compléter et renforcer les mesures préalablement mises en place par les solutions de paiement agréées.

Organisation du travail

Le présent travail sera structuré en trois chapitres. Dans le premier chapitre, nous effectuerons une revue de littérature présentant une rétrospective des différentes productions portant sur le thème. Le deuxième chapitre sera l'occasion pour nous de faire le point sur le matériel utilisé dans notre système ainsi que sur la méthodologie suivie. Dans le dernier chapitre, nous procéderons à la présentation et à l'analyse des résultats obtenus après implémentation de nos solutions.

Revue de littérature

Résumé. Pour mener à bien notre projet, il est important de faire le tour des concepts clés qui y apparaissent et de présenter l'état actuel des recherches sur le sujet. Ce chapitre présente de ce fait la généralité et un inventaire des diverses solutions existantes.

Introduction

Pour la bonne mise en oeuvre d'un projet, une revue de littérature permettant de faire un point sur le sujet du projet est requis. Ainsi, ce chapitre présente un état des lieux des moyens de paiement que représentent les agrégateurs de paiement en général. Cette revue de littérature n'est pas exploratoire et n'est donc qu'un récapitulatif sur la thématique abordée. Dans un premier temps nous parlerons des généralités sur les agrégateurs de paiement puis nous finirons, critiques à l'appui, par une présentation des solutions existantes dans ce domaine.

1.1 Les agrégateurs de paiement

Faire des affaires en ligne ou lancer une boutique en ligne à des fins de commerce électronique nécessite des solutions financières dédiées pour l'acceptation et le traitement des paiements effectués par les clients. Les passerelles de paiement, les fournisseurs de services de paiement et les banques acquéreuses ne sont que quelques exemples des catégories à choisir lors du démarrage d'une entreprise de commerce électronique. S'ajoute à ces derniers, les agrégateurs de paiement vers lesquelles commerçants, microfinances, banques et assurances se tournent en raison des avantages qu'ils offrent dans le contexte de la structure des frais et du système de fourniture de services. Leur modèle d'entreprise simplifie grandement la longueur des dé-

lais de demande et d'approbation et le risque élevé de fraude en ligne. Les questions ci-après faciliteront la compréhension de la notion d'agrégateur de paiement :

- Qu'est-ce que l'agrégation de paiement ?
- Quels sont les avantages liés à un agrégateur de paiement ?
- Quels sont les inconvénients des agrégateurs de paiement ?
- Frais d'agrégation de paiement : à quoi s'attendre ?

Qu'est-ce que l'agrégation de paiement ?

L'agrégation de paiements, également appelée agrégation de marchands, est un modèle commercial dans lequel un fournisseur de paiement tiers inscrit des commerçants directement sous son propre numéro d'identification de commerçant (MID) afin de traiter des transactions via un compte principal unique. Les marchands qui traitent des transactions sous un agrégateur sont appelés sous-marchands. En d'autres termes, un agrégateur de paiements permet aux commerçants sans compte marchand individuel auprès d'une banque ou d'un fournisseur de services financiers de commencer à accepter les paiements par carte de crédit et les transferts d'argent en ligne. Ces fournisseurs de services sont appelés «agrégateurs» car plusieurs commerçants sont regroupés pour former un compte global, contrôlé par le système de paiement en leur nom, alors que les comptes des marchands sont détenus et contrôlés directement par les marchands.

Quels sont les avantages liés à un agrégateur de paiement ?

En tant que commerçant, si votre priorité est de commencer rapidement à accepter les paiements en ligne et par carte de crédit en toute simplicité, un agrégateur de paiements est le meilleur choix. Les agrégateurs de paiement offrent une configuration sans tracas afin que vous puissiez commencer à traiter les paiements presque instantanément. L'on peut citer comme avantages liés à leur utilisation :

- La facilité d'application

La demande d'un compte marchand implique un long processus de demande et de souscription, qui inclut également des contrôles de solvabilité, une garantie personnelle, un contrôle de conformité PCI et un examen minutieux de votre modèle commercial. En revanche, utiliser un agrégateur de paiements nécessite un minimum de paperasse et de contrôles de conformité pour le sous-commerçant.

- La rapidité d'approbation

Les approbations dans un agrégateur de paiement ne prennent que quelques jours, ce qui convient parfaitement aux petites entreprises pour qui le temps est compté. L'acceptation

des paiements par carte peut commencer pratiquement immédiatement après la demande du commerçant pour le service d'un agrégateur de paiements.

- La structure de frais est simple, compréhensible et généralement sans contrat. Il est donc plus facile pour les commerçants de comprendre combien ils dépenseront en frais de traitement.

Quels sont les inconvénients des agrégateurs de paiement ?

Alors que le modèle commercial d'agrégation de paiements est adapté aux petites et moyennes entreprises ayant un faible volume de transactions, le coût de fonctionnement sous l'égide d'un agrégateur de marchand peut augmenter considérablement lorsque le volume de transactions augmente. Le modèle d'agrégation fonctionne parfaitement s'il n'est traité qu'une poignée de transactions. Les sous-commerçants ne paient généralement que lorsqu'ils traitent des paiements en ligne, plutôt que de déboursier des frais mensuels. En fonction des objectifs commerciaux visés et de la croissance projetée, il convient de déterminer si un compte marchand répond mieux aux besoins identifiés. En effet, étant donné que les agrégateurs de paiement assument des risques considérables pour le compte des sous-marchands, y compris les rétrofacturations liées à la fraude, toute activité de paiement suspectée d'être frauduleuse pourrait entraîner la mise en attente d'un compte marchand. Bien que les temps d'attente soient généralement aussi courts que 24 à 48 heures, ils peuvent exceptionnellement durer un mois. Avec les comptes marchands individuels, il y a moins de chances que les entreprises rencontrent de telles interruptions.

Frais d'agrégation de paiement : à quoi s'attendre ?

La plupart des agrégateurs de paiement facturent aux sous-marchands :

- Des frais nominaux ou de création de compte

Dans certains cas, aucun frais n'est prélevé auprès du sous-commerçant pour la création de son compte.

- Des frais de transaction

Lorsque les commerçants acceptent les paiements en ligne, des frais leurs sont facturés. Ces frais diffèrent en fonction de la nature de la transaction. Ainsi, par exemple, les transactions avec des détails de paiement saisis peuvent être facturées différemment des transactions par carte ou de paiements sans contact.

Avant de présélectionner un agrégateur de paiement, il convient donc de ces faits, de s'informer sur les services qu'ils offrent (l'intégration d'API, l'analyse de données, etc.). Ceci peut aider le marchand à optimiser ses investissements.

1.2 Acteurs clés du marché

Il existe de nombreux systèmes de paiement numériques concurrentiels et leur nombre ne cesse d'augmenter. Certains sont populaires dans le monde entier tandis que d'autres ne le sont que dans leur propre pays. Pour se démarquer, ils proposent aux consommateurs et aux commerçants de commencer à accepter et à envoyer des paiements dans le confort de leur foyer, au lieu de faire la queue. Leurs propositions commerciales tentantes leur assurent une place sur les divers marchés financiers et font d'eux des références. Sont précurseurs sur le plan international :

- **Stripe** [17]

Stripe est une plate-forme de paiement en ligne créée par des experts du secteur ayant noué des partenariats avec l'ensemble du secteur des paiements à savoir : des partenaires attendus (banques locales, grands réseaux de cartes, Conseil PCI) et imprévu (le W3C, prestataires de services et associations professionnelles). Il est fortement axé sur les développeurs, leur permettant de créer rapidement des intégrations prêtes à la production avec des outils modernes, des composants React aux Webhooks en temps réel. Son utilisation signifie moins de maintenance pour les systèmes existants et davantage de concentration sur les expériences client et produit essentielles. De même, il traite les frais et affiche les prix dans la devise préférée du client afin d'améliorer les ventes et d'aider les clients à éviter les coûts de conversion. L'inscription à la passerelle de paiement se fait sans problème et rapidement. Toutefois, il est à noter que cette solution n'est à ce jour guère disponible en Afrique.

- **Paielements PayPal Pro** [17]

Payments Pro est la tentative ultime de PayPal d'introduire un service de traitement de paiement pour les utilisateurs de commerce électronique n'ayant pas nécessairement de compte PayPal. La plate-forme de bout en bout prend en charge les cartes de crédit (Visa, Master, Discover, American Express) ainsi que les paiements par PayPal. Elle s'appuie sur une infrastructure API ouverte permettant de se connecter facilement à toute solution panier / place de marché. Elle est un modèle de paiement sécurisé, que l'argent soit transmis depuis un appareil local ou mobile. Elle offre diverses options aux utilisateurs afin de simplifier la conformité PCI, et s'associe à toutes les principales normes de cryptage afin de protéger les informations de paiement. Il s'agit également de la passerelle de paiement la plus internationale en ce moment, puisqu'il accepte 26 des principales devises et qu'il est disponible sur plus de 200 marchés. Il est alors dommage de constater, qu'à ce jour, la création d'un compte marchand PayPal Pro pour une entreprise localisée en Afrique demeure impossible.

Sont leaders sur le marché africain :

- **Rave Flutterwave**

Rave Flutterwave est une passerelle de paiement qui permet des transactions multi-devises. Elle favorise les paiements à partir de cartes, de portefeuilles d'argent mobile et de comptes bancaires. Son implémentation facile et gratuite ne nécessite point l'intervention d'un développeur. L'utilisation de la page de paiement qu'elle aide à fournir demande juste l'ajout de l'extrait de code pour la clé de page et les clés d'API.

- **Paystack**

Paystack constitue une passerelle de paiement en pleine croissance en Afrique. Son utilisation ne demande point à l'utilisateur d'écrire de code et son intégration aucun frais. Sa documentation exhaustive et son interface utilisateur exceptionnelle rendent facile le «Go Live» avec Paystack. Sachant que le Go Live désigne le passage d'un compte test vers un compte Live. Il s'agit d'un processus qualifiable d'intimidant.

Et existe déjà sur le marché béninois :

- **FedaPay [20]**

FedaPay est une passerelle de transactions qui regroupe la plupart des moyens de paiement existants sur le continent africain. Elle intègre aussi bien les solutions de paiement mobile que celles par cartes de crédit et se présente sous forme d'une plateforme web assurant la gestion des comptes marchands en usant d'une API flexible et facile permettant aux entrepreneurs et développeurs d'intégrer le paiement en ligne dans des sites e-commerce et applications mobiles. FedaPay propose trois produits pour satisfaire ses utilisateurs quel que soit leur domaine d'activité :

- *Feda Direct* qui permet aux particuliers de recevoir juste avec un lien, des paiements partout dans le monde depuis leur smartphone, ordinateur, tablette en toute sécurité ;
- *Feda Commerce* qui offre aux e-commerces, un moyen d'intégration facile et flexible dans tout type d'application et sur toute plateforme ;
- Et *Feda Connect* conçu pour offrir aux grandes entreprises une intégration complète dans leurs outils de tous les jours (Par exemple les systèmes de gestion de relation clients communément appelé Customer Relationship Management ou CRM).

Conclusion

L'agrégation de paiements, également appelée agrégation de marchands, est un modèle commercial dans lequel un fournisseur de paiement tiers est également appelé «agrégateur de paiements». Il inscrit les commerçants directement sous son propre numéro d'identification de commerçant (MID) pour traiter les transactions par l'intermédiaire d'un seul maître compte. Par

exemple, Stripe, PayPay, Rave Flutterwave, etc. Un compte marchand est utilisé pour représenter un certain nombre de marchands opposés au modèle traditionnel qui décaisse un compte marchand à chaque commerçant. Les marchands traitant des transactions sous un agrégateur sont appelés sous-marchands.

En termes simples, un agrégateur de paiements habilite les commerçants en leur fournissant les moyens d'accepter les paiements par carte de crédit et les transferts d'argent en ligne sans compte marchand individuel auprès d'une banque ou d'un fournisseur de services financiers.

Matériel et méthodes

Résumé. Les phases d’analyse, de modélisation ainsi que celle des choix techniques de tout projet informatique sont d’une importance capitale pour le bon déroulement du projet. Elles permettent de s’assurer d’avoir bien traduit les besoins en vue d’une préparation optimale de la phase de codage proprement dite.

Introduction

La réalisation de ce projet a nécessité l’utilisation d’un ensemble d’outils et de services qui ont été déterminés après la réalisation du cahier de charges du thème et l’analyse des besoins. Dans le présent chapitre, nous faisons un inventaire des idées et outils liés à la mise en place de l’architecture sécurisée de notre système et nous abordons la mise en place proprement dite du système.

2.1 Analyse

Pour conduire ce projet, nous avons opté pour la Méthode Scrum qui peut être définie comme un schéma d’organisation de développement de produits complexes et considérée comme une pratique agile ; Laquelle qui implique au maximum le demandeur (client) et permet une grande réactivité à ses demandes. Scrum est défini comme un cadre de travail suivant un cycle de développement itératif, incrémental et adaptatif. Il s’appuie sur le découpage d’un projet en boîtes de temps, nommées « sprints » pouvant durer de quelques heures à un mois (avec une préférence pour deux semaines). Chaque sprint commence par une estimation suivie d’une planification opérationnelle et se termine par une démonstration de ce qui a été achevé. [19]

Analyse des besoins

L'analyse consiste à l'aboutissement de l'élaboration d'une solution technique à partir de l'étude des besoins. C'est la première phase du cycle de développement d'un logiciel. Elle sert à identifier les acteurs du système et leur associer chacun l'ensemble des actions avec lesquelles il intervient dans l'objectif de donner un résultat optimal et satisfaisant au client. On distingue deux types de besoins : les besoins fonctionnels et non fonctionnels.

2.1.0.1 Les besoins fonctionnels

Les besoins fonctionnels répondent aux points précis du cahier de charges et sont donc requis par le client. Ils constituent le besoin primaire du client et définissent une fois résolus l'opérationnalité du système. Ainsi l'agrégateur de paiements KKiaPay doit permettre :

- **au financier de :**
 - consulter la liste des clients ;
 - consulter la liste des reversements ;
 - consulter la liste des transactions ;
 - consulter le tableau de bord du marchand ;
- **au développeur de :**
 - consulter les logs ou rapports du système applicatif ;
 - ajouter un webhook ou une URL de rappel de données ;
 - consulter la liste des webhooks ;
 - consulter les diverses clés API mises à disposition ;
 - copier une clé API ;
- **au marchand ou à l'administrateur de :**
 - initier une transaction mobile money ;
 - rembourser une transaction ;
 - configurer les reversements ;
les reversements d'avoirs peuvent être périodique, spontané ou par palier.
 - ajouter un compte dans le cadre des reversements ;
il est à noter que le compte ajouter par ce dernier dans le cadre des reversements peut aussi bien être un compte mobile money qu'un compte bancaire.

- supprimer un compte ajouté dans le cadre du reversement d’avoirs ;
cette action ne peut toutefois être exécutée que s’il existe rattaché au compte marchand un autre compte ajouté dans le cadre des reversements.
- demander l’activation de son compte ;
l’activation d’un compte marchand implique le téléversement par ce dernier des pièces que sont : le Relevé d’Identité Bancaire (RIB), l’Identité Fiscale Unique (IFU), le Régistre du Commerce et du Crédit Immobilier (RCCM) et le numéro d’une quelconque pièce d’identité (passeport ou carte d’identité).
- ajouter un business ;
- supprimer un business ;
- inviter un utilisateur sur la plateforme ;
le marchand ou l’administrateur du compte peut choisir d’inviter sur ce dernier un utilisateur. A ce dernier, peuvent être attribués les rôles qui suivent : financier, développeur ou administrateur.
- supprimer un utilisateur ;
- générer un lien de paiement ;
- supprimer un lien de paiement.

2.1.0.2 Les besoins non fonctionnels

Ces besoins sont soit des besoins optionnels, soit des besoins/contraintes liés à l’implémentation. Ainsi, il faudra que l’application web KKiaPay soit sécurisée, performante, dotée d’une bonne expérience utilisateur et compatible à la plupart des navigateurs.

2.2 Langage de modélisation

Un langage de modélisation est un langage artificiel utilisable afin d’exprimer de l’information ou de la connaissance dans une structure qui est définie par un ensemble cohérent de règles. Leaders du marché, MERISE et UML sont deux grands principes de modélisation d’un système d’informations. Le choix d’user de l’un ou de l’autre afin de procéder à la description future du fonctionnement du système se fera suivant les trois (03) axes que sont : l’accessibilité, la précision et l’exploitabilité.

2.2.1 MERISE

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques poursuivant comme but principal la conception facile d’un système d’informations. La méthode MERISE basée sur la séparation des données et des traitements à effectuer en plusieurs

modèles conceptuels et physiques assure une longévité au modèle. En effet, l'agencement des données n'a guère besoin d'être remanié aussi fréquemment que les traitements. Utilisée dans les années 1970 et 1980 pour l'informatisation massive des organisations, cette méthode reste adaptée pour la gestion des projets internes aux organisations limités à un domaine précis. Elle est, en revanche, moins adaptée aux projets transversaux aux organisations qui gèrent le plus souvent des informations à caractère sociétal (environnemental et social) avec des parties prenantes. [5]

2.2.2 UML

UML est un langage qui permet de modéliser des classes et leurs interactions. C'est un ensemble d'annotations graphiques qui s'appuient sur des diagrammes et permettent de spécifier, de visualiser et de documenter les systèmes logiciels orientés-objets [4]. UML s'impose aujourd'hui comme langage de modélisation objet standardisé pour la conception de logiciels. La version finalisée, UML2 largement enrichie et corrigée est celle que nous utiliserons.

Choix : Notre choix s'est porté sur le langage UML pour ses nombreux avantages. En effet, avec ce dernier nous pouvons utiliser le du concept des objets auxquels sont associables les principes d'héritages de composition et transition d'état.

2.3 Conception

Cette phase permettra de définir de manière non ambiguë, en utilisant un langage de modélisation, le fonctionnement futur du système, afin de faciliter sa réalisation. La modélisation de notre système s'articulera autour des diagrammes qui suivent :

- le diagramme de cas d'utilisation ;
- le diagramme de classe ;
- le diagramme d'objet ;
- le diagramme de séquence ;
- le diagramme d'activité.

L'on distingue les différents types de modélisation que sont : la modélisation fonctionnelle, la modélisation statique et la modélisation dynamique.

2.3.1 Modélisation fonctionnelle

Seront présentés dans cette partie les différents acteurs du système ainsi que les différentes interactions réalisables avec ce dernier. En effet les diagrammes fonctionnels rassemblent : le diagramme des cas d'utilisation (use-cases ou Use Case Diagram), le diagramme de contexte statique, le diagramme de contexte dynamique. Nous présenterons ici que le diagramme de cas d'utilisation.

2.3.1.1 Identification des acteurs du système

En UML, on entend par acteur toute idéalisation d'un rôle joué par une personne externe, un processus ou une entité qui interagit avec le système. Il existe deux types d'acteurs, à savoir : les acteurs principaux qui utilisent directement les fonctionnalités du système et les acteurs secondaires qui contribuent à la réalisation d'un cas d'utilisation. Dans le cadre de notre système nous avons retenu :

- comme acteurs principaux :
 - le financier ;
 - le développeur ;
 - le marchand ou l'administrateur ;
- comme acteur secondaire :
 - l'administrateur du système.

2.3.1.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation présente le comportement de notre système, d'un package ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde donc la fonctionnalité du système en unité cohérente : cas d'utilisation, qui permet d'exprimer les besoins des utilisateurs de notre système.

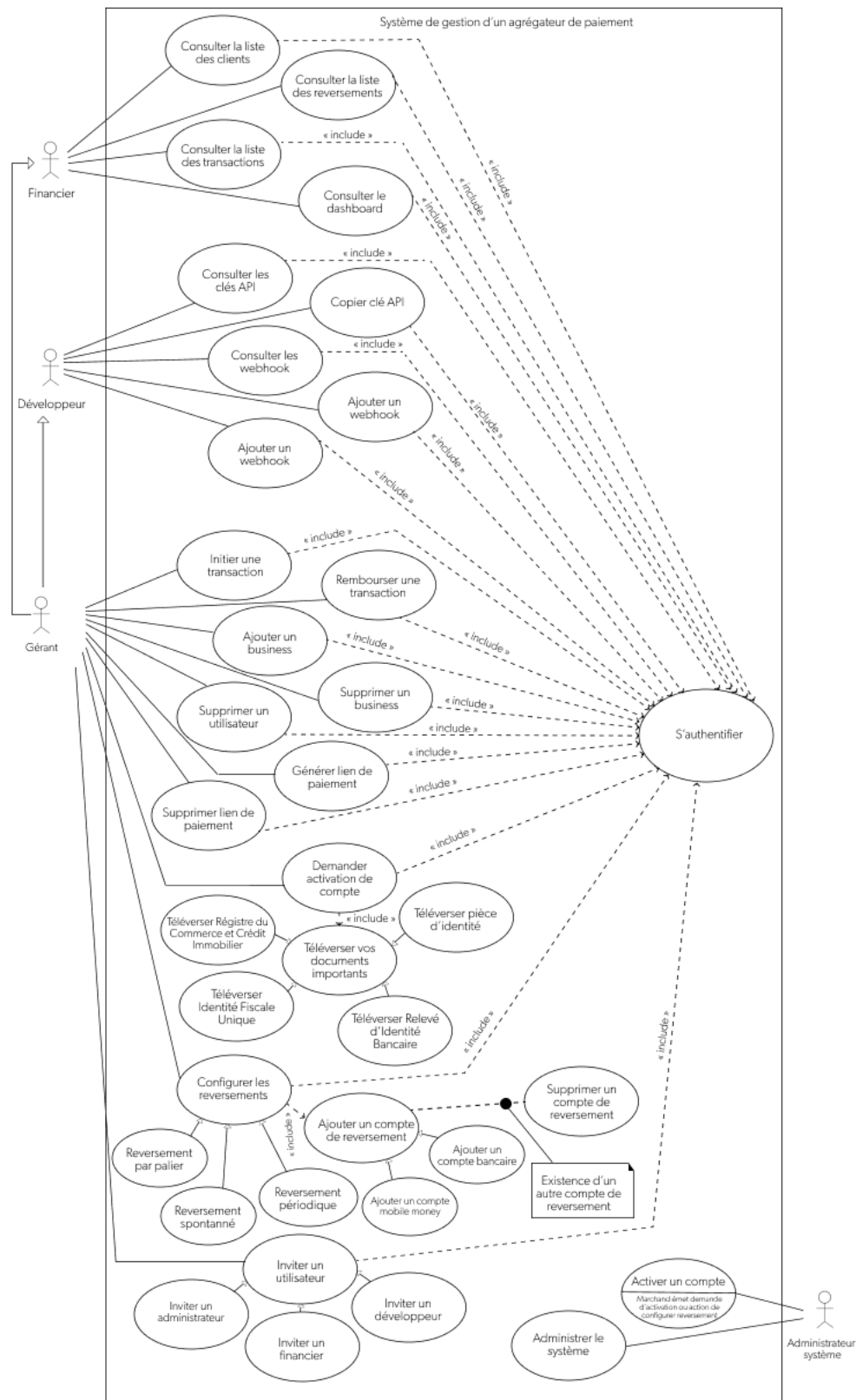


FIGURE 2.1 – Diagramme de cas d'utilisation

2.3.1.3 Description des cas d'utilisation

Cas d'utilisation 1 : Configurer les reversements (package « *Gestion des reversements* »)

Acteur(s) : Marchand

Description : La configuration des reversements doit être possible pour le marchand ou l'administrateur.

Auteur : Mélaine G. KIOSSOU

Date : 02 Mai 2019

Pré-conditions : L'utilisateur doit être authentifié en tant que marchand ou administrateur (Cas d'utilisation « *Se connecter* »)

Démarrage : L'utilisateur a demandé la configuration de ses reversements

Scénario nominal

1. Le système affiche une boîte de dialogue contenant le formulaire de configuration desdits reversements.
2. L'utilisateur sélectionne le type de reversement souhaité.
3. L'utilisateur spécifie les modalités liées au type de reversement choisi.
 - (a) Dans le cas d'un reversement par palier, l'utilisateur spécifie le montant-palier qui doit impérativement être supérieur à 50 000 CFA.
 - (b) Dans le cas d'un reversement périodique, l'utilisateur sélectionne la période de reversement : chaque mois, chaque semaine, tous les 03 jours.
4. L'utilisateur renseigne s'il veut recevoir un mail à chaque opération de reversement.
5. L'utilisateur choisit le moyen de paiement. Lui sont offerts comme option : le mobile money et le compte bancaire.
6. L'utilisateur choisit le compte mobile money ou bancaire de reversement.
 - (a) Si le moyen de paiement choisi est le mobile money : l'utilisateur choisit le numéro de téléphone destinataire du reversement.
 - (b) Si le moyen de paiement choisi est le compte bancaire : l'utilisateur choisit le compte, destinataire du reversement.

7. L'utilisateur valide le formulaire de configuration des reversements.

Scénario alternatif

A1 : Le scénario démarre au point 6. L'utilisateur choisit comme mode de reversement le mobile money.

6-1 L'utilisateur clique sur le bouton " *Ajouter un compte mobile money*".

6-2 Appel du cas d'utilisation «*Ajouter un compte mobile money*».

6-3 L'utilisateur valide le formulaire de configuration des reversements.

A2 : Le scénario démarre au point 6. L'utilisateur choisit comme mode de reversement le compte bancaire.

6-1 L'utilisateur clique sur le bouton " *Ajouter un compte bancaire*".

6-2 Appel du cas d'utilisation «*Ajouter un compte bancaire*».

6-3 L'utilisateur valide le formulaire de configuration des reversements.

Scénario d'exception

E1 : Le scénario démarre au point 3.a. L'utilisateur spécifie un montant-palier inférieur à 50 000 CFA.

3-1 Le système affiche un message d'erreur. Retour à l'étape 2.

Post-condition : Affichage de la notification portant sur le succès de l'opération de configuration des reversements.

2.3.2 Modélisation statique

Elle consiste en l'identification des objets, des classes, de leurs attributs et opérations ainsi qu'en la présentation des différents diagrammes à savoir celui des classes, des paquetages et des objets. Elle offre une vue statique des éléments qui composent le système.

2.3.2.1 Diagramme de classe

Ce diagramme permet de représenter une vue statique du système d'informations. Statique, car l'on n'évoquera point ici les stimuli qui font réagir le système mais plutôt des relations entre les classes, des services rendus et utilisés par chacune d'elles et de l'articulation de l'ensemble.

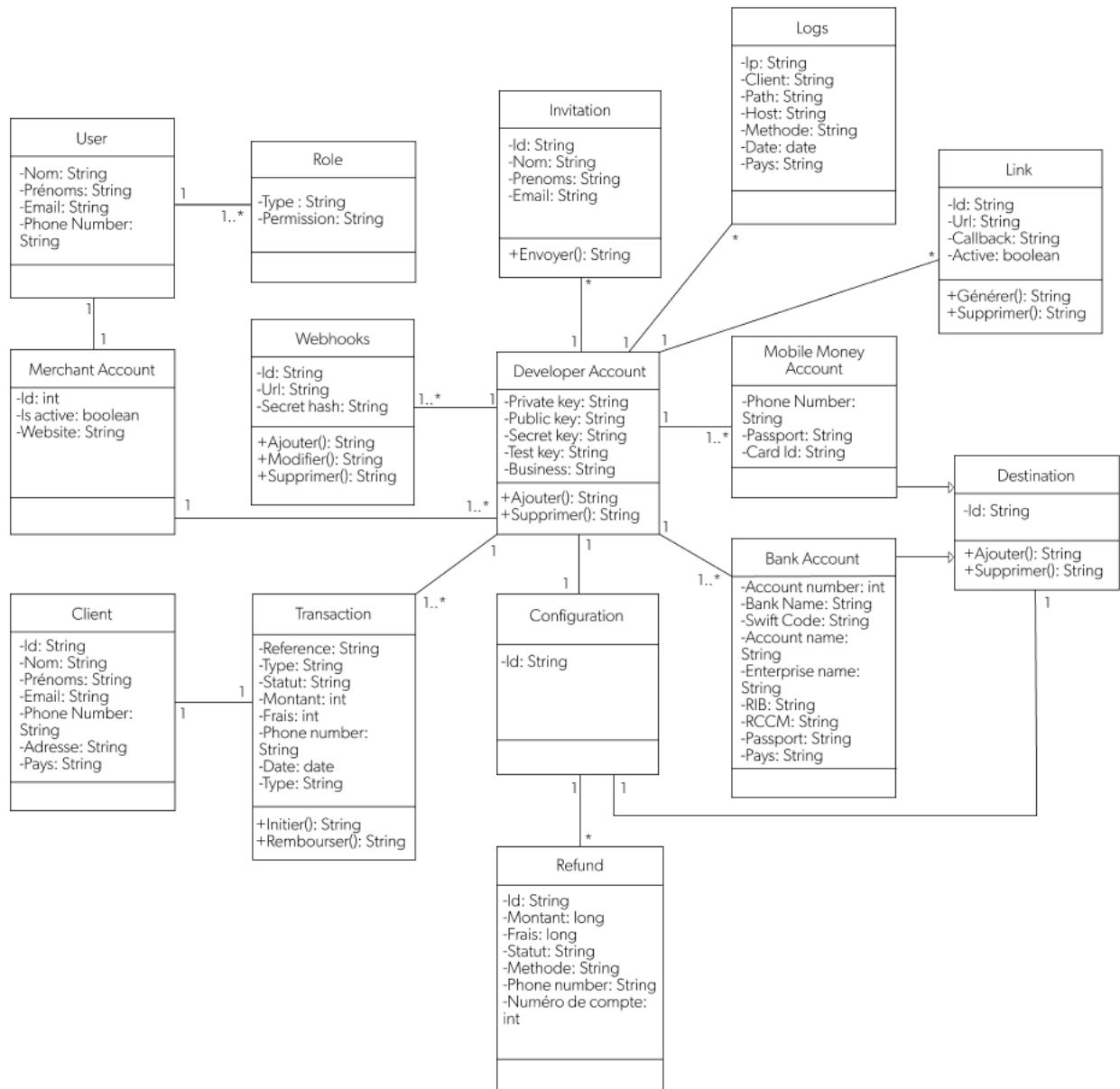


FIGURE 2.2 – Diagramme de classe

Son distinguables trois (03) types d'utilisateurs : le financier, le développeur et le marchand. À un utilisateur est associable un et un seul compte marchand et à un compte marchand peut être lié un ou plusieurs comptes développeurs. Sur un compte développeur peut-être ajouté aussi bien un ou plusieurs comptes mobile money qu'un ou plusieurs comptes bancaires. L'ajout d'un compte mobile money nécessite du marchand les informations suivantes : le numéro de téléphone marchand, une pièce d'identité ainsi que le numéro d'identification de ladite pièce. Celui d'un compte bancaire quant à lui nécessite de fournir les informations ci-après : le numéro

du compte, son intitulé et son code Swift, le nom de la banque et son pays d'accueil, le nom de l'entreprise, une pièce d'identité du marchand, son RIB et son RCCM. Est liée, non seulement à un client mais également à un compte développeur toute opération de transaction. En effet, une transaction ne peut concerner qu'un et un seul client mais à un compte développeur peut être associée une ou plusieurs transactions. De plus, la restitution des divers avoirs d'un marchand ne peut se faire que si et seulement si les reversements ont au préalable fait l'objet d'une configuration. Il doit être spécifié le mode de reversement (spontané, par palier et périodiquement) et le compte marchand (mobile money ou bancaire) de destination. Peut-être ajoutés sur un compte développeur les urls de rappel de données ou webhooks et les rapports de système applicatif ou logs. Il est offert la possibilité au marchand de générer des liens de paiement et d'envoyer des invitations utilisateurs en usant du compte développeur.

2.3.2.2 Diagramme d'objet

Le diagramme d'objets permet d'éclairer un diagramme de classes en l'illustrant par des exemples. Il peut être utilisé afin de vérifier l'adéquation d'un diagramme de classes à différents cas possibles.

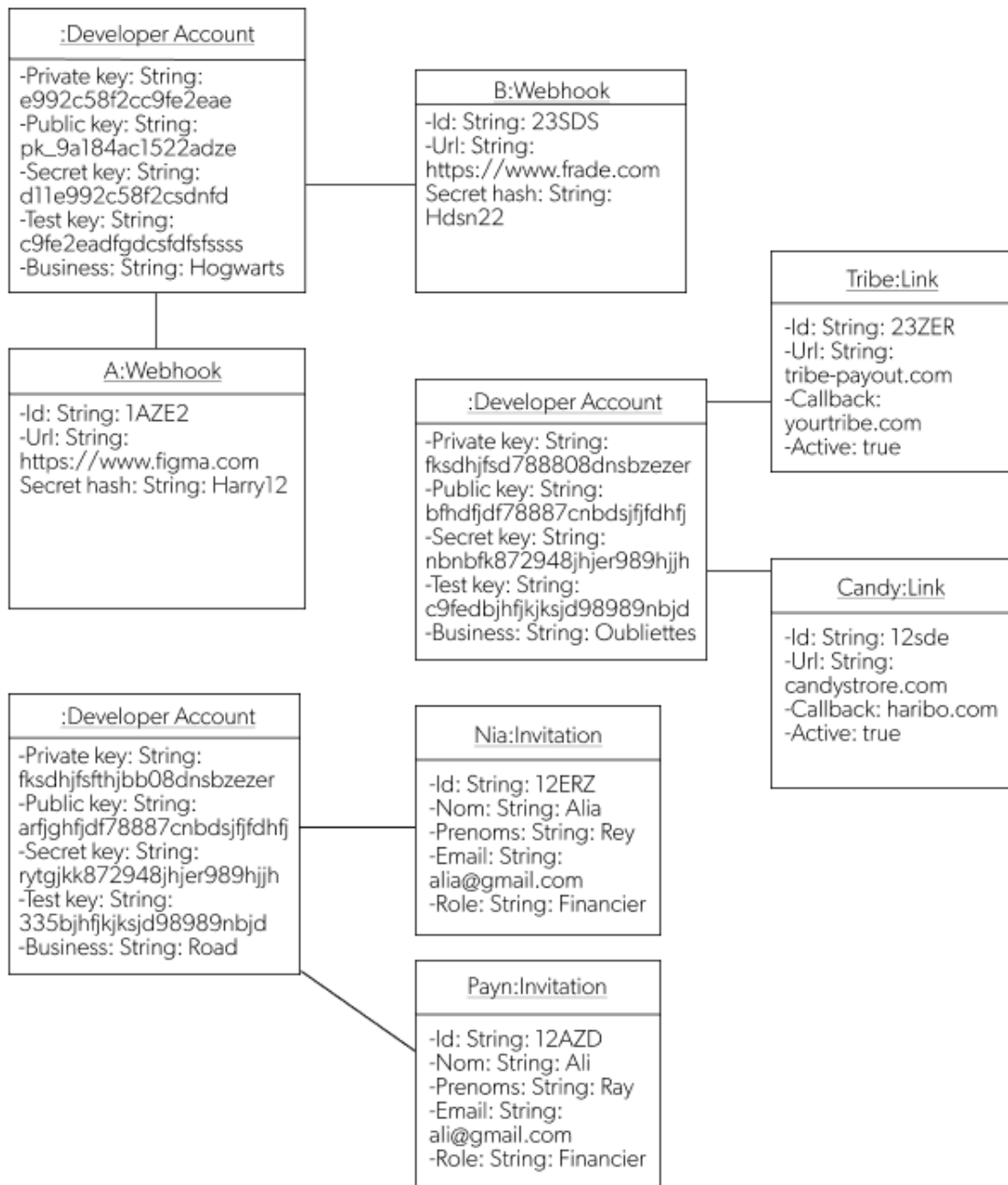


FIGURE 2.3 – Diagramme d’objet

2.3.3 Modélisation dynamique

Les diagrammes dynamiques permettent de modéliser le comportement dynamique du système et d’indiquer comment ses objets interagissent au moment de l’exécution.

2.3.3.1 Diagramme de séquence

Le diagramme de séquence qui représente la séquence met à l'honneur la succession de messages entre objets au cours d'une interaction. Il comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction et représente la succession chronologique des opérations réalisées par un acteur.

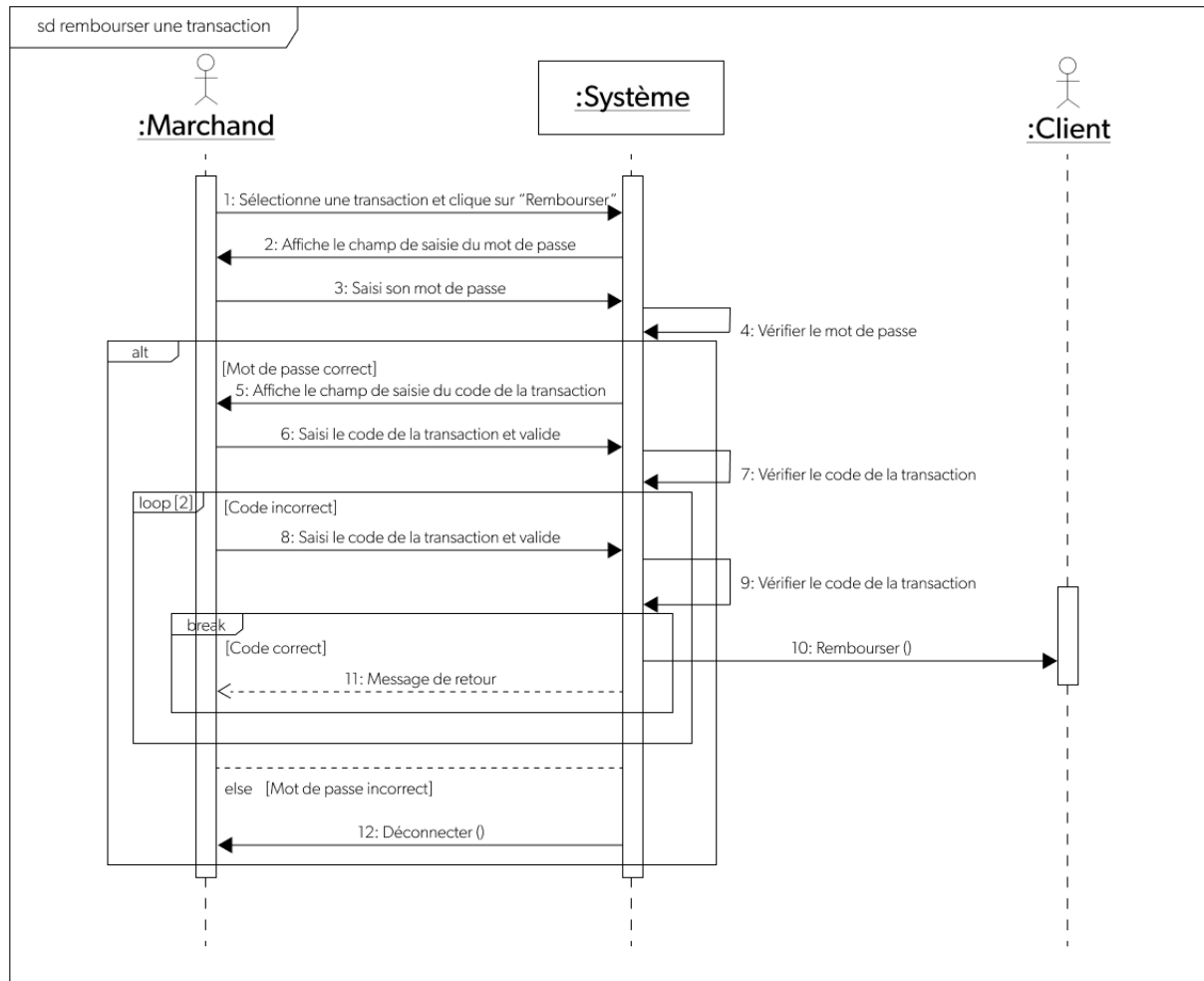


FIGURE 2.4 – Diagramme de séquence - Cas d'utilisation "Rembourser une transaction"

2.3.3.2 Diagramme d'activité

Le diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. Les actions sont les unités de base du comportement du système. Un diagramme d'activité permet de grouper et de dissocier des actions. Si une

action peut être divisée en plusieurs actions en séquence, vous pouvez créer une activité les représentant.

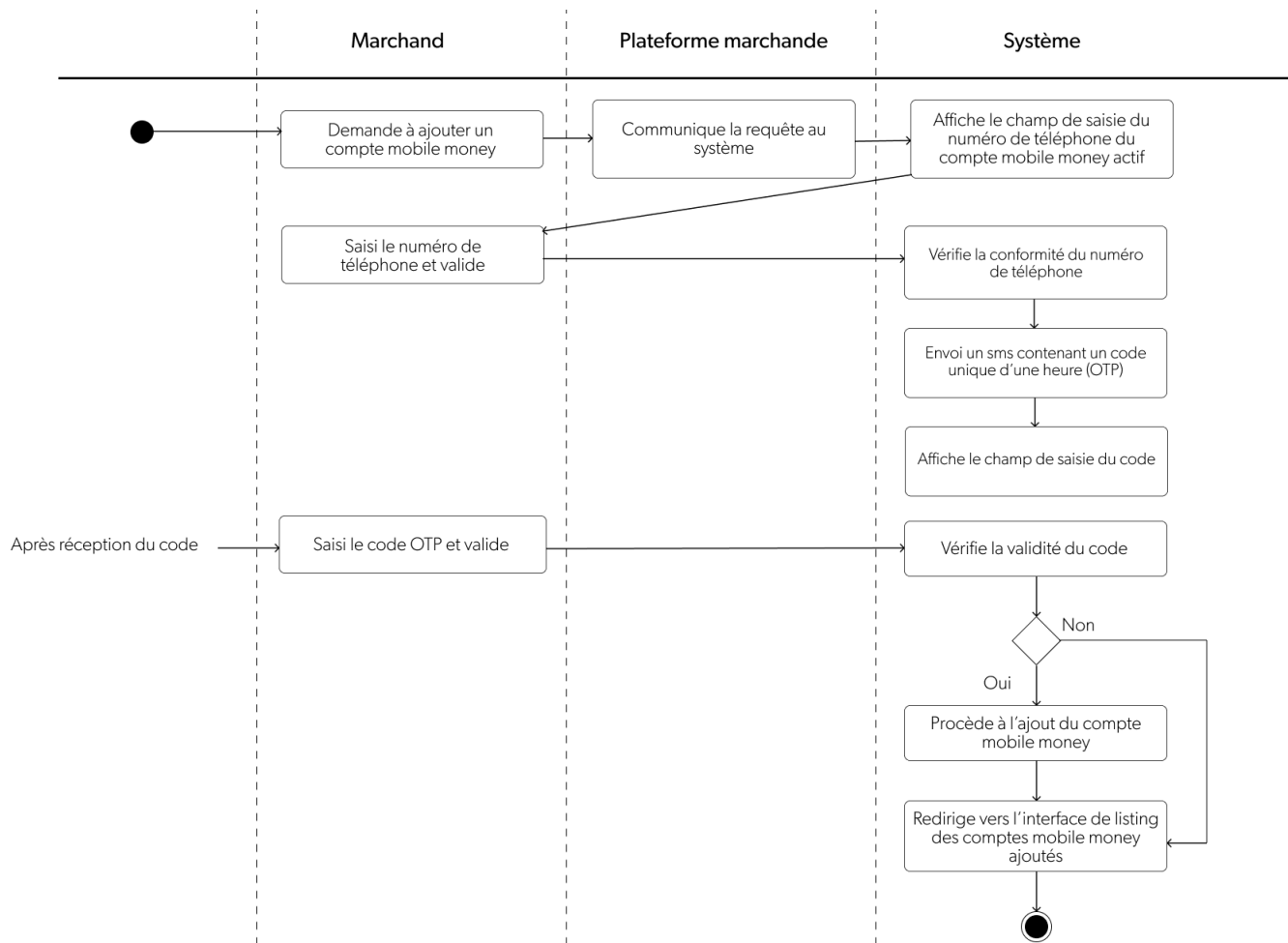


FIGURE 2.5 – Diagramme d'activité - Cas d'utilisation "Ajouter un compte mobile money"

2.4 Choix techniques

La réalisation d'un système demande une hiérarchisation des idées et outils liés. Nous procéderons à un inventaire des outils liés à la mise en place de l'architecture sécurisée de notre système et nous aborderons la mise en place proprement dite du système.

2.4.1 Moteur de base de données

Un système de gestion de base de données (SGBD) est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations. Il permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données. Il existe différentes natures de bases de données. Plus précisément, il

existe différents types de systèmes de gestion de base de données (SGBD ou DBMS en anglais pour « DataBase Management System »). Chaque type de SGBD présente des caractéristiques différentes et chacun est adapté à des contextes particuliers. Afin de procéder à la structuration et à la gestion de nos données, notre réflexion s'est tournée vers ces divers acteurs du marché.

2.4.1.1 PostgreSQL

PostgreSQL est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, même dans des situations critiques. Il est de type SQL et extrêmement respectueux des standards, se conformant au plus près à la norme ANSI-SQL 2008. [22]

2.4.1.2 MongoDB

MongoDB (de l'anglais humongous qui peut être traduit par « énorme ») est un système de gestion de base de données orientée documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant guère de schéma prédéfini des données. Il est écrit en C++. Développé en 2007 par la structure MongoDB, ce moteur de base de données, partie intégrante de la mouvance NoSQL permet de manipuler des objets structurés au format BSON (JSON binaire), sans schéma prédéterminé. En d'autres termes, des clés peuvent être ajoutées à tout moment « à la volée », sans reconfiguration de la base. [21]

2.4.1.3 Redis

Redis est un système de gestion de base de données écrit en C ANSI et distribué sous licence BSD. Il fait partie parti intégrante de la mouvance NoSQL et vise à fournir les performances les plus élevées possibles [23]. Il prend en charge les structures de données telles que les chaînes, les hachages, les listes et permet la persistance des données et une haute disponibilité grâce à la réplication et aux sauvegardes. Il combine le meilleur de la conception en mémoire, sans schéma, avec des structures de données optimisées et offre de ce fait, une base de données polyvalente et évolutive.

Choix : Notre choix s'est porté sur les moteurs de base de données qui suivent : MongoDB et Redis. Notre orientation est motivée par notre choix de stocker nos diverses données de traitement au sein d'une base de données NoSQL "Not only SQL" permettant non seulement la gestion aisée d'importants volumes de données mais également pour certains types d'applications des traitements plus rapides.

2.4.2 Technologies utilisées

La réalisation de l'interface de programmation liée à ce projet a nécessité l'utilisation de l'ensemble d'outils et de services qui suivent :

- **Vue.js**

Vue.js est une librairie Javascript qui permet de créer des interfaces web interactives. Il existe de très nombreuses librairies qui proposent la même chose dont les plus célèbres sont AngularJS et ReactJS. Vue.js est également un framework qui gagne très vite en popularité grâce à son alliance de simplicité, portée à l'extrême, et d'une richesse digne des meilleurs frameworks modernes avec une documentation très complète, mettant en avant sa simplicité. Il dispose également d'une bonne vitesse d'exécution facilitée par sa façon de rendre les vues de façon asynchrone afin d'éviter des rendus inutiles au milieu de longues opérations.

- **Sails.js**

Sails.js est une infrastructure d'application Web MVC développée sur l'environnement Node.js publiée en tant que logiciel libre et à code source ouvert sous licence MIT. Il est conçu pour faciliter la création d'applications Web et d'API Node.js personnalisées pour les entreprises. Emulation de l'architecture MVC d'autres frameworks, comme Ruby on Rails, il offre un modèle et une familiarité similaires, réduisant le fardeau cognitif lors du basculement entre autres cadres/langages.

- **UIKit**

UIKit est un framework css aussi léger que facile d'utilisation qui propose une collection de composants graphiques qualifiables d'intéressants et de minimalistes. Ces composants sont facilement customisables.

- **Notion**

Notion est une application de prise de notes avec prise en charge du démarquage qui intègre également des tâches, des wikis et des bases de données. L'objectif est d'être un espace de travail tout-en-un pour la prise de notes, la gestion de projet et la gestion de tâches. En plus des applications multiplates-formes, il est accessible via la plupart des navigateurs Web et prend en charge la syntaxe de démarquage.

- **Git**

Nous terminons cette partie par la présentation de Git un outil qui a été au cœur de notre travail de développement. Pour une bonne continuité dans l'implémentation et pour prévenir le risque que survienne un problème dans l'implémentation ou une perte des données du disque dur, nous avons besoin d'un gestionnaire de version. Notre choix s'est

donc porté sur Git qui est un logiciel de gestion de version décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux et distribué selon les termes de la licence publique générale GNU version 2. En 2017, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes (d'après Github). Comme Dépôt en ligne, nous avons utilisé la version communautaire de Gitlab avec des dépôts privés.

2.4.3 Architectures logicielles

Au cours de la réalisation de notre projet nous avons utilisé des standards d'architecture logicielle que sont :

2.4.3.1 Architecture RESTFUL

Créée par Roy Fielding, l'architecture RESTFUL est une architecture basée sur le protocole HTTP permettant de construire des applications (Web, Intranet, Web Service). Il s'agit d'un ensemble de conventions et de bonnes pratiques à respecter. Elle utilise les méthodes GET/-POST/PUT/DELETE et les URIs comme représentant des ressources. Dans cette architecture REST, nous avons choisi le JSON comme format d'échange de données pour sa syntaxe souple qui favorise la rapidité des échanges.

2.4.3.2 Patron de conception MVC

Le Modèle-Vue-Contrôleur (en abrégé MVC, de l'anglais model-view-controller) est un patron d'architecture facilement maintenable. Le Modèle représente une information enregistrée quelque part, le plus souvent dans une base de données. Il permet d'accéder à l'information, de la modifier, d'ajouter une nouvelle, de vérifier que celle-ci correspond bien aux critères (on parle d'intégrité de l'information), de la mettre à jour, etc. Il s'agit d'une interface supplémentaire entre notre code et la base de données. La Vue est comme son nom l'indique, la visualisation de l'information. C'est la seule chose que l'utilisateur peut voir. Non seulement elle sert à présenter une donnée, mais elle permet aussi de recueillir une éventuelle action de l'utilisateur (un clic sur un lien, ou la soumission d'un formulaire par exemple). Typiquement, un exemple de vue est une page web, ni plus, ni moins. Enfin, le contrôleur prend en charge la gestion des événements de synchronisation pour la mise à jour de la vue ou du modèle. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour. Chacun de ses composants a un rôle précis et ceux-ci sont tous complémentaires. Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue.

2.4.4 Architecture du système

Au vue des besoins fonctionnels et non-fonctionnels de notre application, il nous faudra procéder à la récupération d'un certain nombre d'informations du système informatique de KKiaPay de façon sécurisée afin de les distribuer de façon contrôlée à la plateforme marchande. Pour cela, il s'avère indispensable d'avoir un point d'entrée afin d'accéder au système d'informations de KKiaPay. Ce besoin implique la conception d'une interface de programmation au-dessus du système d'information.

Se dessine donc comme solution architecturale, une architecture client-serveur n-tiers orientée service (SOA). Au cœur de cette architecture un composant dénommé API pour Application Programming Interface soit Interface de Programmation Applicative devra fournir sous forme de services les fonctionnalités qui ont trait à la logique du métier et au système d'informations de KKiaPay pour l'application cliente. Ainsi toutes les fonctionnalités métiers citées plus haut doivent être développées au niveau de cette API. L'application ne sera donc qu'une simple consommatrice de ces services.

2.4.5 Sécurité de l'application

Le système que nous avons développé traite des informations sensibles propres aux divers utilisateurs. Du fait de la transition des données et de l'interconnexion des différentes applications un certain nombre de mesures sécuritaires doivent être prises afin de réduire autant que possible les failles de sécurité. Nous classons ci-dessous les mesures que nous avons adoptées en fonction des principes de sécurité informatique :

- **La confidentialité**

La confidentialité est le fait de s'assurer qu'une information est accessible uniquement par les entités qui ont le droit d'accéder à celle-ci. Le système que nous avons développé intègre cet aspect de la sécurité à deux niveaux :

- dans un premier temps nous avons installé un certificat ssl de 2048 bits qui nous permet de sécuriser toutes les communications utilisant le protocole tcp/ip ;
- dans un second temps un système de gestion de rôle et de droit intégré nous permet de respecter la confidentialité des données dans chaque sous système. En effet, les utilisateurs appartiennent à des groupes d'utilisateurs donnés et pour chaque groupe seulement un ensemble d'actions bien définies sont possibles.

- **L'intégrité**

L'intégrité s'assure que la donnée reste toujours intègre c'est-à-dire qu'elle n'a pas été modifiée par un tiers non autorisé. Ce principe de sécurité est pris en compte par la mise en place d'un certificat ssl sur le serveur d'applications.

- **La disponibilité**

La disponibilité est le fait de s'assurer que l'information soit toujours disponible peu importe le moment choisi. Elle a été assurée en partie par l'infrastructure mise en place comme des serveurs de redondance mais aussi par la virtualisation des ressources et des applications qui ont été déployées. Aussi une gestion de la montée en charge a-t-elle été prévue mais n'a pas encore été implémentée.

- **L'authenticité**

Au sein d'un système d'information, il est important de vérifier l'authenticité de chaque ressource. Cela est possible grâce au mécanisme d'authentification, qui permet de prouver l'identité d'une personne via le processus d'identification. A notre niveau nous avons mis cela en place par une forte authentification à deux niveaux. Dans un premier temps avant de faire appel à une quelconque URI, il faudrait être authentifié avec un couple d'identifiant comme service tiers de confiance. Donc nos applications clientes s'authentifient avant de demander un quelconque service au backend.

Après être autorisées, les applications clientes ont le devoir d'authentifier leurs utilisateurs avant de pouvoir faire une opération (lecture ou écriture) mettant en relation un utilisateur du système. Bien qu'ayant installé un certificat sur le serveur nous évitons de faire circuler les paramètres de connexion de l'utilisateur. En lieu et place nous avons utilisé un jeton qui est une longue chaîne de caractères générés et encryptés automatiquement lors de l'authentification de l'utilisateur. C'est désormais cette clé renouvelable et ayant une durée de vie limitée qui est désormais envoyée conjointement avec les requêtes pour authentifier l'utilisateur. Toute requête ne contenant pas de jeton ou contenant des jetons invalides est automatiquement rejetée.

- **Non-répudiation**

La non-répudiation se base sur un principe simple : une entité ne peut nier son implication dans une action à laquelle il a participé. Dans notre cas la non-répudiation peut être atteinte seulement en utilisant la technologie du certificat numérique avec un chiffrement asymétrique.

- **Imputabilité**

Il s'agit des techniques mises en œuvre pour assurer la traçabilité des actions d'un individu sur un système. Nous avons mis en place cette traçabilité en journalisant toutes les actions menées sur les objets du système.

Conclusion

Ce chapitre a présenté le fonctionnement de notre système et précisé la méthodologie de gestion de projet dont nous avons utilisé. L'analyse et la réalisation de nos divers diagrammes UML nous ont permis de définir l'architecture de notre système et de choisir convenablement les outils techniques.

Résultats et discussions

Résumé. Ce dernier chapitre, présente la réalisation et l’interprétation des résultats obtenus. Il nous permet de procéder à leur évaluation en tenant compte des objectifs précédemment énumérés.

Introduction

Les travaux menés tout au long de ce projet nous ont permis de concevoir l’agrégateur de paiement KKiaPay : une solution d’encaissement rapide facilement intégrable à tout site web ou à toute application mobile. La présentation des résultats obtenus ainsi qu’une phase de discussion sont nécessaires pour mieux jauger la qualité de ce qui a été fait. Nous procéderons, de ce fait, de manière détaillée notre système, puis nous discuterons de ses insuffisances et limites.

3.1 Application web KKiaPay

Agrégateur de paiement, KKiaPay est un outil qui offre à tout marchand, la possibilité de recevoir des paiements de partout dans le monde. Il est, une solution conviviale, dont l’intégration permet d’accepter aussi bien les paiements effectués par Mobile Money (tous opérateurs confondus) que par Cartes de Crédit (locales et internationales). Son utilisation requiert de disposer d’un compte marchand valide et son intégration de disposer des SDKs ou Plugins à votre technologie liée :

- **avec une plateforme Native (Web ou Mobile) :** Un SDK Javascript et un SDK Android sont disponibles et intégrables au code de votre application.

- **avec un CMS :** Un plugin WooCommerce et un plugin Prestashop sont disponibles et intégrables à votre plateforme lorsque celle-ci est développée avec Wordpress ou Prestashop.

Son infrastructure solide et flexible permet au marchand de fournir à ses clients la meilleure expérience de traitement de paiement possible. La création d'un compte KKiaPay est gratuite et offre à toute institution financière (microfinance, assurance), commerce en ligne, fournisseur de services en ligne ou service de collecte de fonds, la possibilité de recevoir leurs paiements via une interface unique. De plus, il est mis à disposition du marchand un SDK Admin orienté serveur riche en fonctionnalités lui permettant de gérer et de configurer votre compte KKiaPay, de surveiller son processus d'intégration, de garder un oeil sur ses opérations de paiements et de remboursements et de suivre l'état de ses transactions. A ce jour, ce SDK est uniquement disponible pour les technologies PHP et Node.js.

subsectionPrincipales fonctionnalités

Les principales fonctionnalités de KKiaPay s'articulent autour de :

3.1.0.1 L'encaissement à distance ou en point de vente

Si vous êtes un marchand, KKiaPay vous permet, via une interface unique, de vous faire payer grâce à un large éventail de solutions de paiement (MTN Mobile Money, Moov Money, Cartes VISA - MasterCard - American Express). En intégrant KKiaPay à votre plateforme web ou mobile, vous offrez à votre client final une expérience de paiement homogène et simple, quelque soit le moyen de paiement utilisé. L'intégration de KKiaPay dans votre plateforme permet à votre client d'utiliser un des moyens de paiement digital disponibles pour régler ses achats ou effectuer un paiement, sans quitter votre site web ou votre application mobile. L'intégration de KKiaPay passe par la récupération de clés API et l'intégration du kit de développement logiciel.

1. Récupération de clés API

Au nombre de trois (03), les clés API KKiaPay nécessaires à l'intégration de l'agrégateur de paiement, facilitent l'identification de la personne en tant qu'utilisateur approuvé KKiaPay. Orientée client, *la clé publique KKiaPay* utilisable aussi bien sous Javascript qu'Android est destinée à être renseignée sur le tableau WordPress dans la rubrique Paramètres du plugin WordPress. *La clé privée KKiaPay*, uniquement connue de l'utilisateur et de KKiaPay est destinée à un usage côté serveur et permet la vérification des transactions effectuées sur le compte marchand dudit utilisateur par l'API KKiaPay. La dernière clé API KKiaPay, *la clé secrète* est une clé destinée à un usage côté serveur afin d'effectuer des mutations sur le compte marchand KKiaPay de l'utilisateur.

2. Intégration du kit de développement logiciel

Sont mis à disposition des potentiels utilisateurs un set de kits de développement logiciel développés dans les langages de programmation et technologies courantes afin de faciliter l'intégration de KKiaPay dans leur solution, une fois leurs informations renseignées et

leurs documents personnels chargés. En effet, le choix est offert à l'utilisateur lors de l'intégration de KKiaPay, en fonction de la technologie qu'il utilise, d'user des plugins ou des SDKs mis à disposition. Dans le cas, d'un site e-commerce créé en utilisant Wordpress, il lui faudra procéder à l'intégration du plugin Wordpress. Il servira alors de passerelle entre les services qu'il met à disposition et ses clients. Dans le cas, d'une plateforme développée sous Prestashop, il lui faudra procéder à l'intégration du plugin Prestashop. Dans le cas d'une application web, il devra user du SDK Javascript et du SDK Android dans celui d'une application mobile.

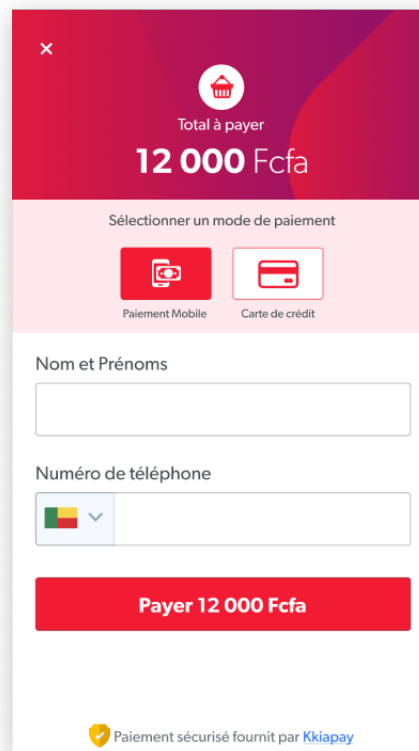
The image shows a mobile payment interface for KKiaPay. At the top, there's a red header with a close button (X) and a shopping cart icon. Below the icon, it says "Total à payer" followed by "12 000 Fcfa" in large white text. Underneath, it says "Sélectionner un mode de paiement". There are two buttons: " Paiement Mobile " with a mobile payment icon and " Carte de crédit " with a credit card icon. Below these are two input fields: " Nom et Prénoms " and " Numéro de téléphone ". The phone number field has a dropdown menu showing the flag of the Central African Republic. At the bottom, there's a large red button that says " Payer 12 000 Fcfa ". At the very bottom, there's a small logo and text: " Paiement sécurisé fournit par Kkiapay ".

FIGURE 3.1 – Interface de paiement unique de KKiaPay - *KKiaPay Widget*

3.1.0.2 Suivi des transactions

Peut être définie comme une transaction toute opération financière engageant un marchand et un acheteur (le client). Elle peut être initiée par le marchand en utilisant le tableau de bord de KKiaPay ou par le client en utilisant l'application web ou mobile dudit marchand. Est offerte, à chaque marchand utilisant KKiaPay, une visibilité en temps réel sur chacune des transactions effectuées (débit, crédit, reversement). Elles peuvent être consultées à tout moment via le tableau de bord KKiaPay. Il est également possible au marchand d'accéder au statut de chacune des transactions, de superviser ses encaissements de fonds et d'effectuer d'éventuels remboursements en direction de ses clients. Avec KKiaPay les frais de transactions sont à la charge du client :

- 3 % du montant de la transaction pour un paiement par Mobile Money (MTN ou Moov);
- 3,5 % du montant de la transaction pour un paiement par Carte bancaire (VISA, MASTERCARD ou AMERICAN EXPRESS).

The screenshot shows the KKiaPay web application interface. On the left is a sidebar with the user's name 'Saham Assurances' and contact info '(229) 66 24 60 31'. The sidebar menu includes 'Tableau de bord', 'Transactions' (highlighted), 'Clients', 'Reversements', 'Développeurs', 'Paramètres', and 'Déconnexion'. The main content area is titled 'Transactions' and features filters for date, period, and type, along with a search bar and a 'Télécharger' button. A green notification box at the top right says '2 transactions remboursées avec succès'. Below this is a table with transaction details. The table has columns: Type, Statut, Référence, Montant(Fcfa), Compte client, and Date. The first row shows a 'Débit' transaction with 'Succès' status. The second row shows a 'Crédit' transaction with 'Echec' status. The third row shows a 'Débit' transaction with 'En attente' status. The remaining rows show 'Débit' transactions with 'Succès' status. A 'Rembourser' button is located above the table. The table footer shows pagination: '1-50 sur 250'.

Type	Statut	Référence	Montant(Fcfa)	Compte client	Date
✓ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
✓ Crédit	Echec	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	En attente	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Remboursée	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019
□ Débit	Succès	rylZLk8z4	5 000 F	66246031	23 Janv 2019

FIGURE 3.2 – Capture d’écran du menu *Transactions* de KKiaPay

3.1.03 Suivi de la trésorerie

KKiaPay garantit le suivi de la trésorerie en facilitant l’exportation des données d’encaissement et de décaissement sous forme de rapports détaillés au format excel. En un seul coup d’oeil, peut être le montant collecté depuis chacun des moyens de paiement disponibles.

The screenshot shows the KKiaPay web application interface. The sidebar on the left contains navigation links: 'Tableau de bord', 'Transactions', 'Clients', 'Reversements', 'Développeurs', 'Paramètres', and 'Déconnexion'. The main content area is titled 'Transactions' and features a table of transaction records. Above the table, there are filters for 'Filtrer par date', 'Filtrer par période', and 'Filtrer par type', along with a search bar labeled 'Rechercher'. A red dashed box highlights a 'Télécharger' button in the top right corner of the transaction list. The table has columns for 'Type', 'Statut', 'Référence', 'Montant(Fcfa)', 'Objet', 'Initiateur', 'Téléphone', and 'Date'. The first row shows a 'Débit' transaction with a 'Succès' status, a reference of 'ryIZlk8z4', a amount of '5 000 F', and an object of 'processing_error'. The table also includes a 'Rembourser' button and a 'Télécharger' button.

FIGURE 3.3 – Point de lumière sur le possible téléchargement des données KKiaPay

3.1.0.4 Visualisation des informations clients et gestion de la relation client

La solution KKiaPay donne un accès aisé aux informations clients, et facilite de facto la gestion de la relation client. Elle offre ainsi au marchand, la possibilité d'intégrer les statistiques sur les paiements reçus directement dans son système d'informations afin d'améliorer son business plan.

3.1.0.5 Récupération des avoirs

Les fonds collectés par le marchand apparaissent sur son compte KKiaPay et peuvent être récupérés à tout moment via la plateforme. En paramétrant la fréquence et le mode de ses reversements, il lui est possible de recevoir ses avoirs en un clic, sur son compte bancaire ou sur son compte marchand mobile money. Les reversements peuvent être instantanés, périodiques (tous les 3 jours, à la semaine ou au mois) ou par palier. Ils sont possibles moyennant le paiement d'un montant fixe de :

- 0 FCFA dans le cas d'un transfert sur son compte marchand Mobile Money ;
- 7000 FCFA dans le cas d'un virement vers son compte bancaire.

Pour des volumes de transactions exceptionnellement élevés, une tarification adaptée peut être discutée. Un mail de confirmation est envoyé après chaque opération de reversement de vos fonds.

3.2 Discussion

Conformément au cahier de charges, notre projet est constitué d'une application Web offrant à tout marchand sous forme de statistiques un aperçu de son solde, de ses transactions et de toute activité effectuée par sa personne en utilisant son compte. Nous avons conçu des interfaces utilisateurs non seulement simples et conviviales mais également ergonomiques afin de faciliter leur utilisation. Les modules de l'agrégateur de paiement ainsi présentés constituent un bloc dans la réalisation d'autres modules plus complexes.

Conclusion

Ce chapitre a fait l'objet d'une présentation des résultats obtenus après l'implémentation de notre système. Il a été question de comparer les attentes formulées aux résultats obtenus. Au vu des résultats obtenus, nous pouvons dire que les objectifs visés par ce projet ont tous été atteints. En effet, nous disposons à ce jour d'un agrégateur de paiement permettant à tout marchand de :

- accepter plusieurs solutions de paiement ;
- faciliter les transactions financières et leur suivi en toute transparence ;
- collecter les données pour une meilleure connaissance du client.

Conclusion et perspectives

Le présent travail a donné à un outil permettant à tout marchand d'accepter des paiements effectués aussi bien par mobile money(tous opérateurs confondus) que par cartes de crédit (locales et internationales). Il nous a vu entamé notre étude par l'analyse qui est une étape cruciale et nécessaire permettant la définition des principaux intervenants et l'identification des besoins. Cette dernière nous a, en effet, permis de concevoir une architecture de base stable et de procéder au choix justifié des technologies avant de passer à l'implémentation dudit outil, ceci en tenant compte de l'architecture matérielle, de l'environnement logiciel et des contraintes techniques de l'entreprise.

Plusieurs perspectives sont donc envisageables afin d'élargir le système ou de l'améliorer. Ainsi, notre système peut être rendu un peu plus complet grâce à l'ajout de nouveaux modules notamment le KKiaPay Direct qui permettra au marchand de rediriger ses clients vers une page hébergée par KKiaPay pour effectuer leurs paiements. Elle n'obligera pas ce dernier à coder et à héberger sa propre page de paiement.

Bibliographie

- [1] Claire Pénicaut, *Le point sur le secteur : Résultats de l'Étude Mondiale 2012 sur l'Adoption des Services d'Argent Mobile*, GSMA — Mobile Money for the Unbanked, 2012.
- [2] Syrtals, *L'AGRÉGATION DE COMPTES : l'accélérateur de la digitalisation des services financiers*, Sytrals 2016.
- [3] Olivier Schmitt, *AGRÉGATEURS DE COMPTES : LE VIRAGE DU NUMÉRIQUE EN ACTION!*, Bank Insight : La newsletter banque des consultants Wavestone, Wavestone, 2017.
- [4] Pascal Roques, *UML 2 par la pratique - Etudes de cas et exercices corrigés*, Eyrolles, 2015.
- [5] Jean-Patrick Matheron, *Comprendre merise : Outils conceptuels et organisationnels*, Eyrolles, 1994.

Webographie

- [6] Cio Mag - Souleyman Tobias <https://cio-mag.com/mobile-money-lafrique-subsaaharienne-le-plus-gros-marche-dutilisateurs-rapport-gsma/> consulté le 13 mai 2019.
- [7] Jeune Afrique - Economie - Finance <https://www.jeuneafrique.com/460397/economie/total-intouch-lagregateur-de-paiement-senegalais/> consulté le 30 avril 2019.
- [8] Medium - Mathias Léopoldie <https://medium.com/banking-on-mobile-in-africa/paiement-mobile-et-e-commerce-en-afrique-francophone-la-ruée-vers-lor-4f29577bca81> consulté le 10 mai 2019.
- [9] Jeune Afrique - Saïd Aït-Hatrit <https://www.jeuneafrique.com/mag/589451/economie/le-mobile-money-en-chiffres/> consulté le 22 février 2019.
- [10] Agence eConfin - Tic <https://www.agenceecofin.com/mobile/1510-41696-digital-afrique-telecom-lance-une-nouvelle-plateforme-de-paiement-mobile> consulté le 10 avril 2019.
- [11] Financial Afrik - Amadjiguéne Ndoye <https://www.financialafrik.com/2018/03/15/lafrique-subsaaharienne-domine-le-marche-mondial-du-mobile-money/> consulté le 22 février 2019.
- [12] Benin Web TV - Benn Michodigni <https://beninwebtv.com/2018/11/entretien-avec-boris-koumondji-fedapay-la-solution-de-paiement-par-carte-de-credit-et-mobile-money/> consulté le 31 mars 2019.
- [13] MindGate <https://www.mindgate.in/what-is-a-payment-aggregator-and-why-would-you-need-one/> consulté le 01 mai 2019.
- [14] WorldCore, <https://worldcore.eu/blog/10-payment-aggregators-offering-access-accounts-right-home/>, consulté le 04 avril 2019.

-
- [15] Le point Afrique- Viviane Forson, https://www.lepoint.fr/economie/mobile-money-afrique-de-l-ouest-plus-de-5-000-milliards-de-francs-cfa-echanges-09-02-2016-2016500_28.php, consulté le 06 avril 2019.
- [16] Jeune Afrique - Télécoms, <https://www.jeuneafrique.com/225068/economie/mobile-money-un-marche-de-250-millions-de-clients-en-afrique-subsaaharienne-d-ici-2019/>, consulté le 06 avril 2019.
- [17] Finances Online <https://financesonline.com/top-20-payment-gateway-providers/>, consulté le 01 mai 2019.
- [18] Fedapay <https://www.fedapay.com/>, consulté le 10 avril 2019.
- [19] Thierry Pigot <https://www.thierry-pigot.fr/scrum-en-moins-de-10-minutes/>, consulté le 14 avril 2019.
- [20] Fedapay - Thierry Hode <https://medium.com/fedapay/paiements-en-ligne-en-afrique-casse-tete-et-mefiance-fedapay-change-la-donne-44b8cea0736>, consulté le 14 avril 2019.
- [21] Wikipedia <https://fr.wikipedia.org/wiki/MongoDB>, consulté le 05 avril 2019.
- [22] D-Booker Editions <https://www.d-booker.fr/content/72-postgresql>, consulté le 05 avril 2019.
- [23] Wikipedia <https://fr.wikipedia.org/wiki/Redis>, consulté le 05 avril 2019.
- [24] GSM Association <https://www.gsma.com/#mwc>, consulté le 25 Juin 2019.
