

Université Assane SECK de Ziguinchor



UFR : Sciences Economiques et Sociales

Département : Economie et Gestion

Filière : Informatique Appliquée à la Gestion des Entreprises

Mémoire de Fin d'Etudes en vue de l'obtention du diplôme de Master Méthodes

Informatiques Appliquées à la Gestion des Entreprises (MIAGE)

Conception et développement d'une solution mobile de gestion de la 'petite monnaie'

Réalisé par : Monsieur Abdoul MBALLO

Encadré par : Dr Guy MBATCHOU, Enseignant-Chercheur au Département
d'Informatique

Supervisé par : Pr Alassane DIEDHIOU, Enseignant-Chercheur au
Département de Mathématiques

Membres du jury :

Président du jury : Pr Mélyan MENDY

Superviseur du mémoire : Pr Alassane DIEDHIOU

Directeur du mémoire : Dr Guy MBATCHOU

Examineur : Dr Khadim DRAME

Examineur : Dr Papa Alioune CISSE

Année académique : 2018 - 2019

Dédicaces

Je dédie ce modeste travail

A mes parents en témoignage de mon affection et de ma profonde gratitude pour leur soutien moral et financier et leurs encouragements ;

A mes frères et sœurs ;

A toute ma famille, qu'elle soit proche ou lointaine ;

A tous mes amis et camarades de classes pour leur aide et conseils ;

A mon tuteur pour son accueil chaleureux tout au long de mon cursus universitaire

A mon oncle depuis les Etats-Unis pour son soutien et ses encouragements durant tout mon cursus universitaire

Remerciements

Avant tout, nous remercions **Allah** le tout puissant de nous avoir donné la santé, la longévité ; la force et le courage afin de terminer ce travail.

Mes remerciements s'adressent particulièrement à mon encadreur, **Mr Guy MBATCHOU** pour tous ses conseils, sa disponibilité, son savoir-faire, sa rigueur de travail et son soutien, il m'a inculqué une grande confiance tout en m'orientant dans le bon sens quant à l'élaboration de ce projet.

Merci également aux membres du jury qui ont accepté d'évaluer ce travail.

Sans oublier non plus nos enseignants, qui tout au long du cycle d'études n'ont ménagé aucun effort pour nous offrir la meilleur des formations.

Merci à notre chère université pour la convivialité et les bonnes conditions de travail.

Un grand merci également à tous ceux qui ont, de près ou de loin, ont participé à l'aboutissement de ce travail.

Résumé

Le but de ce travail est de répondre à une problématique qui tourne autour de la rareté de la 'petite monnaie' (transactions à faible valeur). L'étude de ce problème montre que malgré de nombreuses solutions de transfert d'agent en Afrique et particulièrement au Sénégal, les transactions dues à la « petite monnaie » restent encore une difficulté.

Nous avons donc conçu une solution dont le but est de gérer électroniquement la petite monnaie. Cela permettra aux utilisateurs de gagner du temps durant leurs transactions et diminuera les conflits liés à la rareté de la petite monnaie. Le développement de la solution s'est basé sur le Processus Unifié (UP) comme méthode de modélisation et UML (Unified Modelling Language) comme langage de modélisation.

La solution est implémentée en application mobile bénéficiant des outils Google pour la gestion des données (Firebase Realtime Database) ; la sécurité (Firebase Authentication) et l'envoi des messages (Firebase Cloud Messaging). L'application baptisée « Pimo » a été testée sur plusieurs smartphones avec le système Android tels que Samsung (Galaxy S7, A20), Alcatel. Des tests non concluants ont été faits également sur l'iPhone 6. Dans les perspectives, nous comptons rendre Pimo opérationnel sur la majorité des smartphones et développer une version web de Pimo.

Mots clés : Petite monnaie, Conception, Développement, Application mobile, UP, UML, React Native, Framework, Firebase.

Abstract

The aim of this work is to respond to a problem that revolves around the scarcity of 'small change' (low value transactions). The study of this problem shows that despite numerous agent transfer solutions in Africa and particularly in Senegal, transactions due to "small change" still remain a difficulty.

We have therefore designed a solution whose goal is to electronically manage small change. This will allow users to save time during their transactions and reduce conflicts linked to the scarcity of small change. The development of the solution was based on the Unified Process (PU) as the modeling method and UML (Unified Modeling Language) as the modeling language.

The solution is implemented as a mobile application benefiting from Google tools for data management (Firebase Realtime Database); security (Firebase Authentication) and sending messages (Firebase Cloud Messaging). The application called "Pimo" has been tested on several smartphones with the Android system such as Samsung (Galaxy S7, A20), Alcatel. Inconclusive tests were also made on the iPhone 6. In the future, we intend to make Pimo operational on most smartphones and develop a web version of Pimo.

Keywords: Small change, Design, Development, Mobile application, UP, UML, React Native, Framework, Firebase

Liste des figures

Figure 1 : Interface d'Orange et Moi	21
Figure 2 : Interface d'accueil de My Free	22
Figure 3 : Interface de myWari	23
Figure 4 : Interface de l'application Wave	24
Figure 5 : Interface de Tamweel Touch (BIS)	26
Figure 6 : Interface d'Ecobank MobileMoney.....	27
Figure 7: Interface d'Atlantique Mobile	28
Figure 8 : Interface de SG Connect	29
Figure 9 : Interface de CBAO Mobile.....	30
Figure 10 : Système d'information et système informatique.....	33
Figure 11: Phases et itérations d'UP	36
Figure 12: Historique d'UML.....	37
Figure 13: Types de diagrammes d'UML	38
Figure 14 : Architecture de J2ME.....	55
Figure 15 : Diagramme de Cas d'utilisation du système	60
Figure 16 : Diagramme de séquence "Ouverture compte et Authentification"	62
Figure 17 : Diagramme de séquence du cas " Effectuer transfert "	64
Figure 18 : Diagramme de séquence du cas d'utilisation " Recharger compte "	65
Figure 19 : Diagramme de séquence du cas d'utilisation " Effectuer retrait ".....	66
Figure 20 : Diagramme de séquence du cas d'utilisation " Convertir monnaie "	68
Figure 21 : Diagramme d'activité de l'authentification	70
Figure 22 : Diagramme d'activité de transfert d'argent	71
Figure 23 : Diagramme de classe de la solution.....	72
Figure 24: Exemple d'association " Un à plusieurs "	74
Figure 25 : Exemple d'association " Plusieurs à plusieurs "	74
Figure 26 : Exemple d'association " Un à un "	75
Figure 27 : Décomposition par distinction d'une association d'héritage.....	76
Figure 28 : Décomposition ascendante d'une association d'héritage	76
Figure 29 : Décomposition descendante d'une association d'héritage	77
Figure 30 : Diagramme de composants du système	79
Figure 31 : Diagramme de déploiement du système	80
Figure 32 : Architecture par couche de Node JS.....	81

Figure 33 : Architecture d'Ionic et Apache Cordova.....	91
Figure 34: Architecture de Xamarin.....	94
Figure 35 : Fonctionnement du framework React Native	96
Figure 36: Différence entre frameworks WebView et React Native	97
Figure 37 : Interface d'authentification	103
Figure 38 : Interface d'inscription.....	104
Figure 39 : Interface du compte principal	105
Figure 40 : Interface de transfert	106
Figure 41 : Interface de Transfert et Profil.....	108
Figure 42: Interface Login et Historique sur un Alcatel JOY 1	109

Liste des tableaux

Tableau 1 : Matrice SWOT du projet « Solution de gestion de petite monnaie »	13
Tableau 2 : Comparaison entre application native et Web App.....	46
Tableau 3 : Historique des versions d'iOS	49
Tableau 4: Historique des versions du système Android	51
Tableau 5 : Description du cas d'utilisation "Ouverture compte et Authentification "	63
Tableau 6 : Description textuelle du cas d'utilisation " Effectuer transfert "	65
Tableau 7 : Description textuelle du cas d'utilisation " Recharger compte "	66
Tableau 8 : Description textuelle du cas d'utilisation " Effectuer retrait "	67
Tableau 9 : Description textuelle du cas d'utilisation " Convertir monnaie"	69
Tableau 10 : Description de la base de données.....	73
Tableau 11 : Résumé des frameworks.....	99
Tableau 12: Tests et simulations	107

Table des matières

Introduction générale.....	1
Chapitre 1 :.....	3
La question de la petite monnaie : contexte, problématique et objectifs.....	3
Introduction	4
1.1 Contexte général du projet.....	4
1.1.1 Contexte géographique.....	4
1.1.2 Contexte socioéconomique	6
1.1.3 Rôle de la petite monnaie dans une économie	6
1.1.4 Les faits marquants de la rareté de la petite monnaie au Sénégal	7
1.1.5 Le comportement de spéculation des agents économiques sur la petite monnaie	7
1.1.6 Revue de quelques travaux sur la rareté de la petite monnaie.....	8
1.2 Problématique du sujet	9
1.3 Présentation du projet	10
1.3.1 Définition d'un projet.....	10
1.3.2 Cadre du projet	10
1.3.3 Objectifs du projet.....	10
1.3.4 Nature du projet.....	11
1.3.5 Etapes préalables de réalisation du projet	11
1.3.6 Analyse de l'environnement du projet	12
1.3.6.1 La matrice SWOT	12
1.3.6.2 La méthode PESTEL	15
1.3.7 Motivations.....	17
Conclusion.....	18
Chapitre 2 : L'état de l'art	19
Introduction	20

2.1	Description de l'existant	20
2.1.1	Les solutions Mobile Money	20
2.1.1.1	Orange Money	20
2.1.1.2	My Free (ex Tigo Cash)	21
2.1.1.3	myWari	23
2.1.1.4	Wave	24
2.1.2	Les solutions Mobile Banking (m-banking).....	25
2.1.2.1	Tamweel Touch	25
2.1.2.2	Ecobank MobileMoney.....	26
2.1.2.3	Atlantique Mobile	27
2.1.2.4	SG Connect	28
2.1.2.5	CBAO Mobile	29
2.1.3	Limites de l'existant	31
2.2	Orientation	31
2.3	Système d'information	32
2.3.1	Notion de système d'information.....	32
2.3.2	Système d'information et système informatique.....	32
2.3.3	Méthodologie de conception	34
2.3.3.1	Processus UP (Unified Process).....	34
2.3.4	Un processus de modélisation avec UML.....	37
2.3.4.1	UML (Unified Modeling Language)	37
2.3.4.2	Bases d'UML	37
2.3.4.3	Différents types de diagrammes d'UML	38
	Conclusion.....	39
	Chapitre 3 : Les applications mobiles : technologies et outils de développement	40
	Introduction	41
3.1	Les applications mobiles	41

3.1.1	Définition	41
3.1.2	Contraintes d'applications mobile.....	41
3.1.3	Domaines d'application mobile	42
3.1.4	Les différents types d'application mobile	44
3.1.4.1	Application native	44
3.1.4.2	Application mobile web ou Web App	44
3.1.4.3	Applications hybrides	44
3.1.5	Comparaison entre application native et Web App.....	45
3.2	Technologies et outils de développement.....	47
3.2.1	Les systèmes d'exploitation	47
3.2.1.1	Définition	47
3.2.1.2	Rôles d'un système d'exploitation mobile.....	47
3.2.1.3	Système d'exploitation iOS d'Apple	48
3.2.1.4	Android de Google.....	50
3.2.1.5	Windows phone	51
3.2.2	La plateforme Java 2 Micro Edition (J2ME)	53
3.2.2.1	Présentation de J2ME	53
3.2.2.2	Architecture de J2ME	54
	Conclusion.....	55
	Chapitre 4 : La conception	56
	Introduction	57
4.1	Cadrage du système à modéliser	57
4.1.1	Les acteurs du système	57
4.1.2	Spécification des besoins	58
4.1.2.1	Besoins fonctionnels	58
4.1.2.2	Besoins non fonctionnels	59
4.2	Les différents diagrammes.....	60

4.2.1	Le diagramme de cas d'utilisation	60
4.2.2	Diagramme de séquence.....	61
4.2.3	Diagramme d'activité.....	69
4.2.3.1	Présentation.....	69
4.2.3.2	Quelques exemples de diagramme d'activité	69
4.2.4	Conception de la base de données.....	71
4.2.4.1	Diagramme de classe	71
4.2.4.2	Passage au modèle relationnel	73
4.2.5	Architecture globale de la solution.....	78
4.2.5.1	Vue logique.....	78
4.2.5.2	Vue de déploiement (Vue physique).....	79
5	Conclusion.....	81
Chapitre 5 : L'implémentation.....		82
Introduction.....		83
5.1	Environnement de travail.....	83
5.1.1	Environnement matériel	83
5.1.2	Environnement logiciel	83
5.1.2.1	Visual Studio Code (VS Code).....	83
5.1.2.2	Git	84
5.1.2.3	Le SGBD Firebase	84
5.1.2.4	draw.io	87
5.2	Choix des outils de travail	88
5.2.1	Langages utilisés	88
5.2.1.1	JavaScript.....	88
5.2.1.2	JSX.....	88
5.2.2	Etude des frameworks existants	88
5.2.2.1	Définition	88

5.2.2.2	Ionic	90
5.2.2.3	Xamarin.....	92
5.2.2.4	React Native.....	95
5.2.2.5	Comparaison des frameworks.....	98
5.2.3	La sécurité	99
5.2.3.1	Définition	99
5.2.3.2	Quelques méthodes de défense	100
5.2.3.3	Politique sécuritaire minimale de notre système.....	100
5.2.3.4	Politique sécuritaire globale.....	101
5.3	Implémentation.....	101
5.3.1	Etape de réalisation	101
5.3.2	Interfaces Homme/Machine	102
5.3.3	Tests et optimisations	106
5.3.3.1	Tests et simulations.....	106
5.3.3.2	Essais sur divers modèles de smartphone	107
5.3.3.3	Vérification et validation	109
5.3.4	Les contraintes du système.....	110
	Conclusion.....	111
	Conclusion générale et perspectives.....	112
	Bibliographie.....	113

Introduction générale

Actuellement, à travers les progrès de la technologie, le smartphone est devenu un outil indispensable de travail qui peut apporter un plus dans la vie professionnelle et sociale de tout un chacun. D'ailleurs cela s'illustre parfaitement dans le domaine du Mobile Money qui est l'une des plus grandes évolutions dans le secteur consacré aux échanges de capitaux notamment en Afrique où une grande partie de la population n'a pas de compte bancaire.

Néanmoins, il existe encore beaucoup de progrès et d'innovation à effectuer dans ce secteur. En effet, nous faisons face à un problème accru de manque de petites monnaies dans nos échanges et transactions quotidiens entraînant des tensions entre acteurs et un ralentissement des activités économiques particulièrement du secteur informel. Malgré l'évolution des outils de transfert d'argent, cette question de petites monnaies fréquemment utilisées pour faciliter nos petits achats reste encore un problème non résolu qui mérite une étude approfondie dans le but d'en proposer une solution.

Par définition, le paiement mobile ou mobile money est toutes les transactions effectuées depuis un téléphone mobile et débitées soit sur une carte bancaire, soit sur un porte-monnaie électronique, qui peut être alimenté avec un dépôt de cash auprès d'un agent ou d'un commerçant [1]. De manière simple, le mobile money est une technologie qui permet aux individus de recevoir, conserver et dépenser de l'argent en utilisant leur téléphone portable [2].

Dans le même sens, M. Djibril Diallo, ancien directeur de Tigo Cash (actuel Free Money) s'est exprimé sur la question en disant que « le Mobile Money est l'ensemble des services qui tournent autour de la monnaie électronique et qui utilisent le téléphone mobile comme outil principal de transaction pour le client. A la place des outils traditionnels que sont les espèces, le chèque ou la carte bancaire, on utilise de la monnaie électronique et un téléphone portable pour effectuer des paiements, des transferts, ou aussi accéder à des transactions financières » [3].

Déjà très développés dans le continent africain avec 15% des transactions dans le monde, à en croire les statistiques de la GSMA (Association mondiale des opérateurs de téléphonie). Un article du magazine Jeune Afrique paru le 03 Avril 2020, établit qu'en 2019, 50 millions d'Africains subsahariens ont créé un compte pour transférer de l'argent via leur téléphone mobile, soit une croissance de 12% par rapport à 2018. En effet, Au Sénégal, les systèmes de transfert par téléphone sont présents dans toutes les entreprises de télécommunication (Orange,

Free, Espresso) et autre organismes (Wari, Wizall, Wave, etc.) évoluant dans ce même secteur assez large. Cependant, en se basant de leur limite nous nous engageons à travers ce travail de recherche de proposer notre solution motivée par un contexte assez particulier de notre quotidien, qui est le manque de petite monnaie.

Ainsi, ce présent document est structuré en cinq (05) chapitres établi comme suit :

- Le premier chapitre présente la question de la petite monnaie, le contexte du projet, la problématique et les objectifs préfixés.
- Le deuxième chapitre est dédié à l'état de l'art des travaux de recherche dans ce domaine.
- Le troisième chapitre sera consacré aux appareils et applications mobiles, nous citerons aussi quelques exemples d'applications et les principales plateformes de développement.
- Le quatrième chapitre comporte la conception du futur système à utiliser (structure interne, comportement des utilisateurs, etc.).
- Le cinquième chapitre est consacré à l'implémentation et la présentation du système obtenu.

Enfin, nous terminerons par une conclusion récapitulant le travail effectué jusqu'à l'aboutissement du résultat mais aussi nous allons dégager quelques perspectives de travail dans le but d'améliorer le système développé.

Chapitre 1 :

La question de la 'petite monnaie' :

contexte, problématique et objectifs

Introduction

Notre sujet d'étude s'inscrit dans un domaine purement technologique. Néanmoins, il est indispensable de le mettre au préalable dans son cadre théorique afin d'éclairer les concepts et le domaine dans lequel il s'inscrit.

La petite monnaie est souvent liée au domaine de l'économie car elle concerne en particulier les échanges d'argent et le règlement de petits achats. Il est donc important d'apporter quelques éclaircissements sur l'impact et le rôle de cette petite pièce sur l'économie du pays.

Dans ce chapitre, nous allons nous intéresser tout d'abord au contexte général de notre sujet ainsi qu'à la problématique. Ensuite, nous procédons à la présentation du cadre du projet et à son environnement de travail. Enfin, nous allons décrire clairement l'objectif général du projet en s'appuyant aux objectifs spécifiques et résultats spécifiques attendus.

1.1 Contexte général du projet

1.1.1 Contexte géographique

Le franc CFA (Communauté Financière Africaine), est présent dans plusieurs pays d'Afrique : le Sénégal, le Mali, le Niger, la Côte d'Ivoire, le Tchad, la Guinée-Bissau, le Togo, le Bénin, le Cameroun, la République Centrafricaine, le Gabon, le Congo et les Comores. C'est un avantage considérable pour tous ces pays pauvres d'avoir une monnaie commune leur permettant d'éviter les frais d'impression et de change et empêchant la dévaluation quotidienne dont souffrent l'ensemble des monnaies. Cependant, depuis la dévaluation du FF (Franc Français) en F CFA, l'Afrique subsaharienne est surtout frappée par un manque accru de la petite monnaie. Dans les pays de l'Afrique centrale, notamment au Cameroun, il est devenu plus difficile pour les commerçants et les usagers de disposer de pièces de monnaie, indispensables pour les courses de tous les jours et les petites transactions [4].

Au cours du 4^e Comité de politique monétaire de la Banque des Etats de l'Afrique Centrale (BEAC) tenu le 18 décembre à Yaoundé, Abbas Mahamat Tolli, le gouverneur de la Banque centrale, s'est exprimé sur le phénomène de l'exportation des pièces de monnaie au Cameroun. Le manège consiste pour des expatriés, notamment des Asiatiques, à faire sortir des pièces de 50 et 100 FCFA du Cameroun en grosse quantité, pour les recycler dans la fabrication de bijoux à l'étranger. Abbas Mahamat Tolli a alors déclaré : « je n'en ai pas connaissance de façon précise. Mais je sais que par voie de presse, nous avons appris qu'il y a des comportements de certaines entreprises qui, non seulement transfèrent des pièces de monnaie FCFA, mais

exportent ces pièces ». Il a ajouté : « Nos services mènent des enquêtes avec les autorités des pays membres de la Cemac (Communauté économique et monétaire de l'Afrique centrale) pour déterminer l'ampleur de ce phénomène afin de nous doter des moyens pour le circonscrire. Il y a déjà des mesures d'interdictions qui sont prises. C'est un sujet sur lequel nos équipes sont à pied d'œuvre pour déterminer l'ampleur, les circuits, et l'historicité » [5].

En novembre 2018, la chaîne de télévision privée camerounaise, Equinoxe TV, a diffusé un reportage montrant dans un quartier à Douala, la saisie par la gendarmerie, d'une cargaison de pièces d'argent d'une valeur de 2 millions FCFA chez un expatrié. Selon les enquêteurs du groupement de gendarmerie en charge de l'enquête, les pièces de monnaie étaient collectées dans des machines à sous et exfiltrer du Cameroun via des containers. Conséquence : les pièces de monnaie sont de plus en plus rares au Cameroun.

L'Afrique subsaharienne est l'une des rares régions en développement où la rareté de la petite monnaie pose encore problème. On doit se demander si dans la zone Franc, la rigueur actuelle de la gestion de la petite pièce ne constitue pas un frein à l'épanouissement des économies locales [6].

Ainsi, le Sénégal, la Côte d'Ivoire et le Burkina Faso sont également touchés par ce phénomène du fait que les usagers du franc CFA ont très souvent du mal à effectuer leurs transactions, faute de petite monnaie. Le commerce du détail est fréquemment perturbé dans les pays de l'UEMOA par la rareté de petites coupures (billets de 500 francs et de 1 000 francs) et de pièces de monnaie (5, 10, 25, 50, 100 francs). Des incompréhensions fréquentes opposent les vendeurs et leurs clients lors du règlement des opérations commerciales. Les commerçants détaillants (transporteurs, pharmacies, vendeurs ambulants, etc.) ont des difficultés à rendre la monnaie à leurs clients. Certaines ventes d'articles ne sont pas conclues lorsque l'acheteur tend un billet de grosse coupure [7].

En conférence de presse le 09 Juillet dernier à Dakar, le Directeur National de la Banque Centrale des Etats de l'Afrique de l'Ouest pour le Sénégal Monsieur Ahmadou Al Aminou LO a révélé que l'institution financière testera le paiement électronique à grande échelle en Août 2019, afin de résoudre une bonne fois pour toute la pénurie de petite monnaie que traverse le pays. Invitant les médias nationaux à sensibiliser davantage sur l'e-paiement, Ahmadou Al Aminou LO a indiqué que le système est disposé à revoir les conditions de tarification pour que les usagers puissent payer avec leur téléphone ou leur carte bancaire [8].

1.1.2 Contexte socioéconomique

Actuellement, trouver de la petite monnaie au Sénégal est devenu très difficile. Cette « situation anormale » pénalise de nombreux commerçants qui perdent des clients, ralentit le service dans de nombreuses administrations publiques (à la mairie pour un exemplaire d'acte de naissance, paiement de facture d'eau et d'électricité, achat de timbres administratifs pour une légalisation) et suscite même des activités illégales comme la perception de commissions pour faire la monnaie.

Une économie où les petites activités (constituées par le secteur informel) sont prépondérantes nécessite également que les petites coupures de monnaie soient facilement disponibles et en quantité suffisante pour couvrir librement les transactions désirées par les agents. Les imperfections des politiques économiques et les comportements de certains utilisateurs peuvent cependant nuire à la libre circulation de la monnaie entraînant des conséquences généralement néfastes sur l'activité économique et le bien-être collectif.

La petite monnaie de franc CFA est devenue une denrée rare au Sénégal, au point qu'elle est source de nombreux antagonismes, de désaccord et de différends dans les échanges commerciaux entre les populations, constate-t-on de plus en plus dans les principaux centres urbains et ruraux du pays. Comme exemple, une grande pénurie est observée sur toutes lignes de dessertes d'AFTU (Association de Financement des Transports Urbains) le plus grand réseau de bus au Sénégal et de la société de transport Sénégalaise Dakar Dem Dikk (DDD), entreprise étatique de transport urbain, occasionnant quelquefois des échauffourées entre employés et passagers.

1.1.3 Rôle de la petite monnaie dans une économie

Cette partie vient situer le rôle important que joue la petite monnaie dans une économie de marché, et en particulier le rôle que joue la petite monnaie dans une économie africaine, comme l'économie sénégalaise qui est caractérisée par une prédominance de l'économie informelle. Tout d'abord, il est important de définir la monnaie comme un intermédiaire aux échanges contrôlé par la banque centrale et qui remplit dans la vie quotidienne une triple fonction : celle de calcul économique, de paiement et de réserve de valeur.

La petite monnaie communément appelée monnaie divisionnaire est définie comme étant l'ensemble des pièces ou monnaie métalliques. En, Afrique sub-saharienne, elle est principalement utilisée dans les transactions de faibles montants [9], [10]. L'évolution de la part

de la petite monnaie dans la masse monétaire des pays de la zone franc est restée constante, sinon décroissante depuis la naissance de cette zone monétaire.

Pourtant dans une économie sénégalaise fortement influencée par l'économie souterraine ou informelle et vue la faiblesse du pouvoir d'achat de la plus part des personnes, il y'a une multiplicité des petites activités entraînant des petits échanges et par conséquent la forme de monnaie la plus liquide est la monnaie divisionnaire. Le premier fait marquant pour toute nouvelle personne débarquant dans ce pays c'est le problème de la rareté de la petite monnaie.

1.1.4 Les faits marquants de la rareté de la petite monnaie au Sénégal

Deux années après la dévaluation du franc CFA, la vie économique sénégalaise est marquée depuis 1996 par le phénomène de carence ou de rareté de la petite monnaie. Considérée au départ comme passagère et limitée à quelques types d'opérations, aujourd'hui la situation s'est généralisée, se prolonge et ses conséquences attirent aussi bien l'attention des institutions économiques à travers l'agitation dans les milieux les plus touchées. A presque tous les échelons de l'économie, et principalement ceux où l'on retrouve les petites transactions (en termes de masse monétaire ou somme d'argent) constituées d'achat de marchandises en détails ou de règlement des services élémentaires, les agents font face à la difficulté d'usage des pièces de monnaie. En effet la détention de monnaie par les différents agents pour le motif de transaction est plus forte en billet de banque qu'en pièces métalliques, et peu de vendeurs acceptent le règlement d'achats par billet dès que le remboursement nécessiterait d'être composé de pièces de monnaie. Ces contraintes freinent et limitent le déroulement des échanges et créent des rapports conflictuels entre les différents secteurs qui se rejettent la responsabilité de la disparition de la petite monnaie. La perception du phénomène de la rareté des petites pièces aux différents niveaux du circuit des échanges permet de mieux comprendre son étendue.

1.1.5 Le comportement de spéculation des agents économiques sur la petite monnaie

En principe, la petite monnaie doit être demandée juste pour assurer les petites transactions, comme acheter un bout de pain, prendre un bus, le taxi, etc. mais de nombreuses personnes détiennent des stocks de petites monnaies chez elle pour plusieurs raisons :

- La première est une conséquence même de la pénurie de la petite monnaie, c'est donc un comportement de précaution. On peut donc dire que la rareté de la petite monnaie entraîne une augmentation de la détention de ces petits jetons par les agents, ce qui entraîne une augmentation de la pénurie de la petite monnaie. Ça semble être un cycle.

- La deuxième raison est un fait non prouvé mais qui fait de témoignage de l'homme de la rue. Il paraît que la valeur intrinsèque de la pièce de monnaie soit plus élevée que sa valeur faciale : des personnes font état d'un circuit de fonte de pièce de monnaie en vue de la fabrication des bijoux.
- La troisième raison est le fait que la pénurie ait développé une activité parallèle qui aggrave encore le phénomène : c'est la vente de petites monnaies contre des coupures de valeurs beaucoup plus grande moyennant un intérêt. Les entreprises qui ont besoin de petites monnaies se ravitaillent dans ce marché.
- Le quatrième aspect est le comportement des banques commerciales qui crée des distorsions dans le circuit de distribution de la petite monnaie. Lors des retraits d'argent aux guichets des banques commerciales, chaque client devrait normalement en demander une partie en petite monnaie, malheureusement cette demande est rarement satisfaite.

1.1.6 Revue de quelques travaux sur la rareté de la petite monnaie

La rareté de la petite monnaie n'est pas un phénomène typique du Sénégal. Elle a été constatée dans plusieurs économies, des plus anciennes (Etats-Unis, France ...) aux plus jeunes (pays de la zone franc). C'est ainsi qu'entre 1917 et 1922, ce phénomène fût constaté à Nice, en France, paralysant ainsi le petit commerce [11]. Pour contourner cette difficulté, car clients et commerçants n'ayant plus de petite monnaie pour les transactions quotidiennes, la chambre de commerce de Nice dit se résoudre à fabriquer des espèces de substitution que l'on appelle "monnaies de nécessité". Cette opération fut aussi inhabituelle que remarquable dans la mesure où l'émission de monnaie n'entre pas dans les attributions traditionnelles des Chambres de commerce. La Chambre de Commerce de NICE suggérait la mise en place d'un système de carnets de tickets auprès d'entreprises grosses consommatrices de monnaie divisionnaire, telles que la Compagnie des Tramways ou les éditeurs de journaux. La création de jetons-monnaies de substitution était envisagée et des exemplaires étaient demandés à des fins d'étude à la ville de Nantes qui en avait fait fabriquer pour les besoins de ses entreprises de transport public.

Les Etats Unis ont été aussi victime de ce fléau en 2002, et c'est l'état de Philadelphie via sa banque centrale qui a entrepris de modéliser la demande de chaque coupure de petite monnaie, afin de prévoir dans le futur les demandes de monnaie. Une variété de modèles a été proposée, testée et utilisée pour faire des prévisions [12].

Par rapport à la rareté de la petite monnaie en Afrique, plusieurs journaux ont écrit des articles, relatant les manifestations concrètes dans le vécu quotidien des ivoiriens, des sénégalais, des burkinabés, des camerounais, etc. portant sur la rareté de la petite monnaie.

1.2 Problématique du sujet

Pour illustrer la couverture contre le risque d'échange, nous faisons référence à quelques exemples parmi les comportements les plus couramment constatés. Dans les kiosques, les alimentations, les taxis, les bus et les clandos, on observe des pancartes ou des autocollants qui délivrent des messages tels que « Préparez la monnaie SVP » afin de prévenir la clientèle sur les difficultés de remboursements.

Le travail est orienté particulièrement sur la gestion de la petite monnaie et son utilisation. Les travaux antérieurs se focalisent seulement sur les services de transfert d'argent, de règlement de factures et d'éventuels achats (crédit téléphonique e connexion internet). Avec moins de 500 F dans un compte Orange Money ou Wari, il est impossible de les retirer, de les transférer à quelqu'un d'autres ou d'en acheter quelques choses à la boutique voire même payer son transport dans un taxi ou un bus urbain. Les utilisateurs rencontrent également des problèmes à localiser facilement un point de vente de ses services. La faiblesse de la quasi-totalité des architectures réside dans le fait qu'elles se basent toutes sur des serveurs centralisés. De ce fait, l'utilisation d'un seul central dans un système de transfert d'argent entrave la disponibilité des informations des utilisateurs et ne permet pas une fluidité de transfert pouvant même entraîner une panne ou un ralentissement du système. Toutes ces observations suscitent un certain nombre d'interrogations :

- Quelle solution pour faire face à ce problème de manque de petite monnaie ?
- Comment définir et élaborer un système informatique répondant aux attentes des utilisateurs pour pallier ce phénomène ?

Ainsi nous constatons que malgré le nombre fulgurant des systèmes de transfert d'argent qui existent déjà au Sénégal, aucun d'eux n'a encore abordé cette situation. Pourtant ces mêmes systèmes ont apporté leur expertise dans plusieurs secteurs de la vie courante. Cependant ce besoin quotidien de la petite monnaie demeure toujours sans solution.

Le but de ce travail est d'apporter des réponses à toutes ces questions. La rareté de la petite monnaie n'est pas un phénomène nouveau et des études théoriques ont déjà été menées auparavant. L'idée n'est pas de reprendre ces même études et analyses mais plutôt de se baser

sur celles-ci afin d'apporter une solution techniques pouvant endiguer ou à la limite réduire ce manque permanent de petites monnaies. Il serait donc intéressant de proposer une solution de transfert de la petite monnaie adaptée aux nouvelles tendances et aux attentes des utilisateurs. Dans ce cadre, il est indispensable d'étudier au préalable les solutions existantes sur le marché.

1.3 Présentation du projet

1.3.1 Définition d'un projet

Plusieurs définitions ont été données pour expliquer le terme « projet » par les différents auteurs.

Le projet est « un ensemble complet d'activités et d'opérations qui consomment des ressources limitées (main d'œuvre, devises, etc....) et dont les cibles (clients, bénéficiaires, etc....) attendent des revenus ou autres avantages monétaires ou non monétaires » [13].

Mieux encore le projet est « un processus unique de transformation de ressources ayant pour but de réaliser d'une façon ponctuelle un extrant spécifique répondant à un ou des objectifs précis, à l'intérieur des contraintes budgétaires, matérielles, humaines et temporelles ». Enfin selon Hubert Wallot, le projet « est un processus ayant pour but de réaliser un produit spécifique, une œuvre concrète qui suit un cycle de vie de durée limitée et est constitué de phases et d'étapes typiques avec ses composantes variant selon le stade du cycle » [14].

Un projet est un ensemble finalisé d'activités et d'actions entreprises par une « équipe projet » sous la responsabilité d'un chef de projet dans le but de répondre à un besoin défini par un contrat dans des délais fixés et dans la limite d'une enveloppe budgétaire allouée [15].

1.3.2 Cadre du projet

Ce projet intitulé « Conception et développement d'un système de gestion de petites monnaies » est réalisé au sein de la filière Informatique Appliquée appartenant au département d'Economie et Gestion de l'Université Assane SECK de Ziguinchor dans le cadre de mon mémoire de fin de cycle de master en vue de l'obtention du diplôme de MIAGE (Méthodes Informatiques Appliquées à la Gestion des Entreprises).

1.3.3 Objectifs du projet

L'objectif principal du projet est de concevoir et développer une solution mobile dont le but est de faciliter aux futurs utilisateurs la gestion (envoi, réception, achat de produits, etc.) de leur petite monnaie. Le résultat de ce travail doit répondre aux objectifs fixés.

Notre objectif général se décompose en différents objectifs spécifiques à savoir :

- **Faciliter l'échange de petites monnaies** : Avec son téléphone, l'utilisateur pourra envoyer et recevoir sa petite monnaie à tout moment sans perdre du temps ni se faire escroquer (certaines personnes font exprès de refuser de rendre la monnaie en disant qu'ils n'en ont pas afin qu'on leur laisse avec).
- **Réduire ou négliger les coûts de transfert** : Avec une telle solution, l'idée est de pouvoir envoyer de petites sommes d'argent donc il n'y aura pas lieu de payer des frais de transfert.
- **Participer à la diversification des services de transfert d'argent** : Vu le nombre de système de transfert d'argent déjà existant, l'idée n'est pas ici d'être leur concurrent mais plutôt un complément des services qui sont sur le marché pour une bonne satisfaction des clients.

1.3.4 Nature du projet

A la base de tout projet, il y'a une activité principale qui peut être commerciale, artisanale, industrielle et agricole. Elle est très importante car elle a notamment des incidences au niveau des formalités et de la fiscalité. Néanmoins, notre projet est un cas particuliers car il s'agit d'une activité de prestation de services. Ce type d'activités peut être exercé dans des domaines variés et être de nature commerciale, artisanale ou libérale. Par suite, la vente de services de transfert d'argent nous range directement dans la nature commerciale.

En effet, ce projet se veut être une solution au manque accru de la petite monnaie dans les échanges récurrents entre individus. A travers cela, nous envisageons de réaliser un système digital pour smartphone afin de faciliter les transactions de petits avoirs de façon efficace et en toute sécurité.

1.3.5 Etapes préalables de réalisation du projet

Dans tout projet, il nécessaire de vérifier d'abord la réalisabilité avant sa réalisation. Ainsi pour une bonne entame d'un projet, il faudra impérativement respecter ces différentes phases suivantes :

- **Validation de l'idée retenue** : C'est l'étape initiale du projet qui consiste tout simplement à faire une étude de marché. Il s'agit surtout d'un travail d'exploration marketing destiner à analyser, mesurer et à comprendre le fonctionnement réel des forces à l'œuvre. Cela nous permettra de s'assurer que le marché dans lequel l'on souhaite se lancer répond à nos attentes.

- **Etude de faisabilité technologique** : Une fois que l'idée a été validée, il est également indispensable de procéder à une analyse technique des différentes solutions technologiques envisagées avant d'en choisir celle qui répondra le plus à nos exigences.
- **Etude de faisabilité économique** : Il faut s'assurer que les coûts et les délais sont acceptables et respectables. Dans notre cas, le processus de développement de la solution requiert une connexion à internet pour l'installation des outils ainsi que les différents tests à faire tout au long. Cependant, le déploiement du système demande un certain investissement à effectuer et des partenariats à nouer avec des organismes d'accompagnement tels que les banques, ONG et structures étatique (ANPEJ, FONGIP, DER). Etant donné qu'il s'agit d'un système de transfert d'argent, on aura obligatoirement besoin de ressources humaines pour assurer le bon fonctionnement et la disponibilité du service.

1.3.6 Analyse de l'environnement du projet

1.3.6.1 La matrice SWOT

Tout projet se déroule dans un environnement qui peut être complexe et varié d'un projet à un autre. Avant de commencer le projet, le gestionnaire analysera de manière exhaustive son environnement et celui du projet, afin d'en comprendre les enjeux et les contraintes potentielles.

La Matrice SWOT (Strengths - Weaknesses - Opportunities – Threats) ou FFOM (Forces – Faiblesses – Opportunités – Menaces), met l'accent sur l'environnement interne et externe d'une entreprise, d'une organisation ou même d'un projet [16]. L'environnement interne fait référence d'une part aux forces à la disposition de l'équipe projet afin d'atteindre leur objectif, et d'autres part les faiblesses à encadrer pour ne pas nuire la réalisation du projet. L'environnement externe touche aux variables qui sont indépendantes de la volonté de l'entreprise ou de l'équipe projet et qui peuvent influencer la bonne marche du projet. Il s'agit des opportunités et menaces qui entourent la réalisation de l'idée.

L'analyse SWOT s'inscrit dans la phase d'avant-projet. Elle permet de faire le point sur les compétences et les capacités du porteur de projet. Analyser un projet à travers cette matrice doit permettre de :

- Maximiser les forces et les opportunités du projet,
- Encadrer les faiblesses et les menaces pour réduire les risques d'échec dans la réalisation du projet

Le tableau 1 suivant résume la pensée de cet auteur sur la matrice SWOT

<i>FORCES</i>	<i>FAIBLESSES</i>
<ul style="list-style-type: none"> - Innovation - Clientèle ciblée - Ressources matérielles disponibles - Compatibilité au système Android et iOS 	<ul style="list-style-type: none"> - Marché ciblé limité - exigence de la connexion internet - Manque d'assistance financière - Manque de soutien technique
<i>OPORTUNITES</i>	<i>MENACES</i>
<ul style="list-style-type: none"> - Croissance des services de transfert d'argent - Demande forte dans le secteur - Possibilités de partenariats avec des financeurs 	<ul style="list-style-type: none"> - Instabilité du réseau téléphonique - Concurrence intense - Risque de piratage du système - Obligation de se signaler aux services étatiques

Tableau 1 : Matrice SWOT du projet « Solution de gestion de petite monnaie »

1.3.6.1.1 Forces

- ❖ ***L'innovation*** : Le fait que le projet soit nouveau au Sénégal malgré les multiples systèmes d'échanges d'argent, lui donne une certaine force sur le marché.
- ❖ ***Clientèle ciblée*** : La solution est accessible à tous, même les plus démunis qui se souciait parfois des commissions de transactions à supporter. Les marchands et les conducteurs des transports urbains en sont les plus concernés par cet outil.
- ❖ ***Ressources matérielles disponible*** : Techniquement, le développement de la solution se fait normalement dans le respect des outils exigés par la technologie utilisée.
- ❖ ***Compatibilité sur Android et iOS*** : Beaucoup de solution mobile sont développées pour un système d'exploitation spécifique. Cependant, notre projet fournit une solution disponible sur toutes les deux plateformes les plus utilisées. Ce qui peut diversifier les utilisateurs et toucher plus de clients.

1.3.6.1.2 Faiblesses

- ❖ ***Marché ciblé limité*** : Même si la solution sera déployée partout au Sénégal, elle sera absente pour l'instant dans les autres pays. Pendant que tous les autres systèmes évoluent à l'échelle internationale, le nôtre se limite qu'au Sénégal. C'est un désavantage par rapport à la concurrence.
- ❖ ***Exigence de la connexion internet*** : L'une des faiblesses remarquables de la solution est son utilisation avec une connexion internet. En effet, basée sur Firebase (base de

données en ligne de Google), il est impossible d'accéder aux données des utilisateurs sans se connecter à internet.

- ❖ **Manque d'assistance financière:** Même si la réalisation technique de la solution ne nécessite pas de ressources financières, son déploiement en exige. La mise en circulation effective du système demande un certain investissement. Mais il est tout à fait difficile d'avoir des partenaires prêt à assister financièrement le projet afin qu'il puisse décoller.
- ❖ **Manque de soutien technique :** Tout au long du développement du projet, le constat était que l'on ne pouvait pas bénéficier d'un soutien quand on avait un souci sur la technologie de développement utilisée. C'est l'une des conséquences des nouveaux framework car il est compliqué de trouver un développeur au cas où il sera nécessaire d'ajouter des fonctionnalités.

1.3.6.1.3 Opportunités

- ❖ **Croissance des services de transfert d'argent :** Les services de transfert d'argent sont en hausse au Sénégal. Avec l'augmentation de la couverture nationale des services de la télécommunication, de l'internet ainsi que celle des achats des smartphones, le contexte est très favorable pour le projet.
- ❖ **Demande forte dans le secteur :** La demande concernant les services de transfert d'argent est forte au Sénégal. Vu le marché, il n'existe pas un service de transfert de petite monnaie, malgré le besoin exprimé quotidiennement par les clients.
- ❖ **Possibilités de partenariats avec des bailleurs :** Le secteur du transfert d'argent est un marché propice aux partenariats avec des investisseurs particuliers et même des banques. Ce qui permettra au projet de grandir et de créer de la richesse et de l'emploi.

1.3.6.1.4 Menaces

- ❖ **Concurrence intense :** La concurrence fait rage dans le domaine des transferts d'argent au Sénégal. Il existe déjà des services très bien placés sur le marché, ce qui pourrait ralentir l'intégration de notre solution.
- ❖ **Instabilité du réseau téléphonique :** Il existe plusieurs zones du pays où il y'a encore un problème de couverture du réseau téléphonique. De ce fait, il sera difficile d'avoir une connexion internet stable. Ce qui pourrait entraîner un dysfonctionnement du service dans ces zones.

❖ **Le piratage** : Les systèmes de transfert d'argent font toujours face à des personnes malveillantes dont le but est d'accéder au compte des clients et de dérober leur argent. Une telle pratique amènera à certains de se méfier de la solution.

Obligation de se signaler aux services étatiques : Un projet appelé à grandir et se faire connaître de tous doit de se signaler auprès de l'Etat. Dans notre cas, nous devons nous enregistrer au niveau du RCCM (Registre de Commerce et crédit Mobilier) et à l'APIX (Agence nationale pour la Promotion des Investissements et des grands travaux). Il faudra également un NINEA (Numéro d'Identification Nationale des Entreprises et Associations) délivré par l'agence des impôts et domaines afin de contrôler si le service s'acquitte à ses droits d'impôts et taxes. A défaut de cela, on peut risquer une lourde amende à payer.

1.3.6.2 La méthode PESTEL

1.3.6.2.1 Politique

L'environnement politique d'un pays a une influence sur l'économie. Aussi une entreprise, un projet ou un service quel que soit sa nature doit en tenir compte. En effet on ne s'implante pas et on ne se développe pas de manière similaire au Sénégal qu'en Côte d'Ivoire ou au Maroc. Une entreprise doit donc se soucier de la stabilité politique du régime mis en place, des relations qu'il entretient avec les autres pays, de la liberté qu'il octroie ou pas à sa population et aux entreprises. Le droit intellectuel est-il respecté et protégé, tout comme les brevets ? La concurrence peut-elle librement se développer ? La législation mise en place est-elle favorable à l'entreprise ? Le gouvernement contrôle-t-il les taux d'intérêts ? Quelles sont les politiques de taxation ? Etc. Notre projet doit répondre à ces questions avant de s'implanter. La communication et le marketing que nous adoptons doivent s'adapter à la réalité politique du territoire Sénégalais.

1.3.6.2.2 Economie

Notre projet œuvre à priori dans les services de transfert d'argent, mais son expansion dans d'autres volets n'est pas exclue. Ainsi, nous comptons s'implanter dans tout le territoire Sénégalais afin de pouvoir toucher le maximum de clients. A travers cela, nous allons créer de la richesse et de l'emploi des jeunes mais aussi participer au développement de l'économie du pays. Dans le cadre du développement sur le territoire, nous allons prendre en compte certains facteurs tels que :

- **L'inflation** : Elle facilite pour notre projet une adaptation de la structure des prix et des salaires. Elle constitue, si elle est forte un soutien à la demande globale, ce qui stimule la croissance. Par définition, L'inflation est une baisse durable de la valeur de la monnaie. Il s'agit d'un phénomène persistant qui fait monter l'ensemble des prix, et auquel se superposent des variations sectorielles des prix [17], [18].
- **Les progrès économiques** : Quels sont les revenus réels de la population ciblée ? Y'a-t-il une augmentation généralisée ?
- **Le marché du travail** : Les offres et les demandes d'emplois s'équilibrent-elles ? Quels sont les salaires moyens dans le secteur ? Y a-t-il un salaire minimum obligatoire imposé ?
- **Le taux d'intérêt** : Nous devons nouer un partenariat avec les banques et discuter avec eux sur ce qu'ils doivent percevoir sur notre revenu. Plus le taux est faible, plus nous avons la possibilité de s'engager dans d'autres partenariats. Ce qui pourrait favoriser notre environnement économique et la notoriété du service.

1.3.6.2.3 Socio-culturel

Les facteurs socio-culturels ont une grande importance pour une entreprise, un projet ou un service donné souhaitant se développer dans un pays. Elle doit connaître les caractéristiques des consommateurs. Quel est l'âge moyen de la population ? Quels sont ses loisirs, ses croyances ? Est-elle familière aux nouvelles technologies ? Un projet comme le nôtre doit également connaître le niveau d'étude de la population pour mieux cibler ses offres mais également pour savoir si elle trouvera du personnel assez qualifié pour répondre à ses besoins d'employeurs. Cela conditionnera la gestion de ses ressources humaines locales et en particulier au niveau de son plan de formation allant dans le cadre du service et de l'utilisation du système.

1.3.6.2.4 Technologie

La technologie est primordiale pour nous car elle caractérise même notre projet. La moitié de nos investissements à effectuer pour accroître et améliorer le service sera dans le domaine de la technologie. L'achat d'équipements et le recrutement de personnel qualifié reste une priorité dans le but d'innover et d'offrir un meilleur service à la clientèle.

Ainsi, nous avons des perspectives de partenariats avec les sociétés de télécommunications du pays et les organismes évoluant dans le secteur de la technologie tels que l'ADIE (Agence De l'Informatique de l'Etat) et l'ARTP (Autorité de Régulation des Télécommunications et des Postes).

1.3.6.2.5 Environnement

Les initiatives en faveur de la durabilité de l'environnement sont très importantes pour l'image d'une entreprise ou d'un projet comme le nôtre ayant une politique communicationnelle très axée sur le consommateur. A travers notre service, nous comptons lancer des campagnes de sensibilisation sur l'hygiène, l'assainissement et le respect de l'environnement. Dans le même sens, nous allons mettre en avant un engagement fort concernant le recyclage des téléphones usagés et en offrant des réductions de frais pour l'achat d'un téléphone. Ce qui réduirait les déchets et la pollution dans nos localités.

On peut faire un appel aux collectivités territoriales et aux associations pour collecter des mobiles usagés afin de créer « des ateliers de récupération de déchets de mobile au Sénégal »

1.3.6.2.6 Légal

Une entreprise, un projet de développement ou un service donné doit se conformer aux législations d'un pays où elle souhaite s'implanter. Ces législations sont diverses et peuvent concerner le droit du travail, le respect de l'environnement, la protection des données des consommateurs, le droit à la propriété intellectuelle, le droit relatif à la concurrence, etc. En cas de non-respect de cette législation, l'entreprise peut être lourdement sanctionnée. Notre service est condamné à respecter ces lois et règles établies, y compris ces obligations de taxes et d'impôt auprès de la DGID (Direction Générale des Impôts et Domaines) du pays.

1.3.7 Motivations

La réalisation de cet outil me permettra non seulement de mettre en valeur mes connaissances tout au long de ma formation mais aussi cela pourra nous aider à réduire les conflits souvent causés par la rareté de la petite monnaie dans nos grandes villes. Issu d'une formation d'Informatique de Gestion de la licence au master 2, il nous incombe d'apporter des solutions technologiques adaptées aux entreprises et organisations ou même à toute une population dans le but de leur faciliter leur quotidien en automatisant certain de leur tâches. Par conséquent, la réussite de cet outil offrira davantage une bonne notoriété à mon institution et à ma filière. En effet, suite à un constat sur la problématique qui tourne autour de la petite monnaie, on s'est engagé d'apporter notre participation dans le combat de la résolution de ce phénomène à travers des connaissances techniques et technologiques acquises tout au long de notre formation.

L'autre aspect est la nouveauté (5 ans d'existence) et la rareté (pas encore très utilisé dans les entreprises et dans les programmes scolaires) de la technologie que j'utilise pour développer mon système. En effet, durant notre cursus universitaire, l'utilisation d'un tel environnement

de travail ne figurait pas dans le programme. Alors l'apprentissage de ce nouvel outil est un avantage qui pourra me distinguer des autres lors d'un recrutement dans le développement mobile.

Et enfin, à travers cette solution, je pourrai participer au développement de mon pays en créant ma propre entreprise mais également de l'emploi aux jeunes diplômés de ma génération. Véritablement, une fois le projet réalisé, il sera indispensable de trouver du personnel capable de continuer le développement, l'amélioration et la maintenance du système. L'ajout de nouvelles fonctionnalités, la diversification des services et l'innovation nous obligerons de recruter des stagiaires et des développeurs expérimentés capable d'assurer l'exécution de ces tâches. Puisqu'il s'agit d'une solution de transfert d'argent, il est également indispensable d'employer des distributeurs à déployer dans différents points de vente afin d'assurer la disponibilité du service. Ce qui participera à la réduction du chômage qui sévit dans notre pays.

La création de ce système combinée aux expertises des différents associés et partenaires permettront de trouver des solutions efficaces aux attentes et aux besoins de la clientèle. Ceci signifie concrètement : une réduction du déficit de petites monnaies, la facilitation de l'utilisation de petits montants en toute sécurité et l'apaisement des tensions et problèmes existants entre individus lors des transactions financières.

Conclusion

Ce chapitre constitue une partie introductive dans laquelle, le problème de la petite monnaie, le contexte du sujet et sa problématique ont été élaborés juste au début. Ensuite, une présentation du cadre du projet et les différentes grandes lignes à travers lesquelles repose tout le travail à faire.

Cette partie nous a permis de définir les notions liées au projet, de situer notre problème et de l'analyser dans son environnement autour duquel il doit évoluer.

Compte tenu de cette présentation, une étude de l'art du sujet est indispensable afin de pouvoir orienter le développement et d'extraire les principales fonctionnalités à réaliser.

Chapitre 2 :

L'état de l'art

Introduction

Faire une étude des principaux concepts existants en rapport avec la problématique étudiée est un préalable incontournable à la réalisation d'un travail scientifique. Dans ce chapitre, nous allons présenter l'état de l'art qui entoure notre sujet d'étude afin de bien cerner notre champ d'exécution.

Les solutions de transfert d'argent sont multiples et diversifiées dans tout le pays. Cependant, réaliser un projet répondant aux besoins et aux attentes de tous est quasiment impossible, ceci est d'ailleurs notre constat sur le secteur du mobile money.

Dans la suite de ce chapitre, nous allons d'abord faire une description détaillée des différentes solutions existantes sur le marché. Ensuite, nous allons faire une analyse critique de ces différentes solutions suite à laquelle nous dégagerons des améliorations à apporter.

2.1 Description de l'existant

L'étude de l'existant est une phase importante pour comprendre les solutions existantes et leur limite afin d'en apporter une amélioration à ces solutions ou même proposer un nouveau système pour une meilleure diversification des services.

De nos jours, les techniques de transfert d'argent sont multiples. Une bonne maîtrise et connaissance de ces divers systèmes est nécessaire pour pouvoir dimensionner judicieusement notre propre solution. Dans cette section, nous discutons les spécificités ainsi que les principaux enjeux auxquels est soumis le marché du mobile money et nous énumérons les différentes solutions proposées. Nous présentons également une étude comparative des performances attendues des différentes technologies employées en termes de réactivité, d'efficacité, de souplesse et de coût de l'application.

2.1.1 Les solutions Mobile Money

2.1.1.1 *Orange Money*¹

Il s'agit d'une solution de transfert d'argent et de paiement mobile du groupe Orange, proposé dans la majorité des pays africains où l'opérateur est présent. Cette application permet à ses utilisateurs de déposer de l'argent sur un compte associé à leur numéro de mobile, pour ensuite accéder à une gamme de services moyennant quelques frais notamment le transfert d'argent national et international, le paiement des factures et l'achat du crédit téléphonique. Pour

¹ : <https://www.orange.sn/orange-money-app>

alimenter son compte, l'utilisateur peut se rendre dans un point de vente Orange Money pour y déposer des espèces, recevoir de l'argent d'un autre utilisateur par transfert ou recevoir son salaire directement sur son compte. La Société Orange Sénégal a développé l'application mobile **Orange Money** pour faciliter l'utilisation de ses services et de ne pas être obligé de retenir des codes d'accès au service. Elle est disponible par téléchargement et son utilisation est assez pratique et facile. Ci-dessous nous vous présentons une interface de l'application Orange Money :

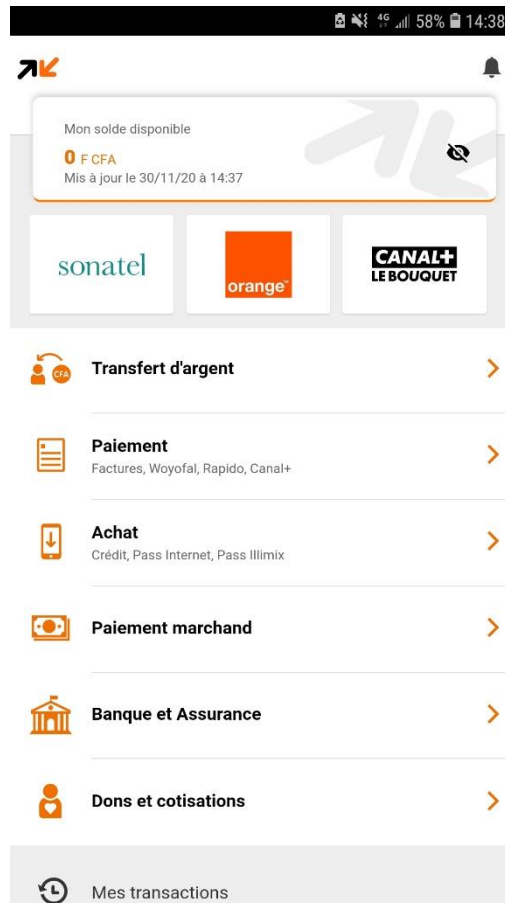


Figure 1 : Interface d'Orange Money

2.1.1.2 My Free² (ex Tigo Cash)

C'est un service de monnaie électronique lancé par l'opérateur téléphonique Free, propriété du groupe **SAGA Africa Holdings Limited** contrôlée par Yerim Sow, Xavier Niel et Hassanein Hiridjee.

Le service Free Money est accessible en composant #150# sur son téléphone mobile équipé d'une carte SIM de l'opérateur téléphonique Free³. Tout client peut utiliser ce service à

² : <https://free.sn/free-money>

³ : <https://www.free.sn/propos-de-free-senegal>

condition de s'enregistrer avec une pièce d'identité valide et de fournir les informations nécessaires.

Free Money permet d'effectuer les opérations suivantes :

- Envoi d'argent
- Dépôt d'argent sur son compte auprès d'un agent commercial
- Achat de crédit d'appel pour son numéro ou un autre numéro Free
- Retrait d'argent auprès d'un agent commercial
- Gestion de son compte (vérification du solde, changement de code pin, ...)

La solution mobile *My Free*⁴ est disponible en téléchargement sur Google Play Store et App Store. Voici l'interface d'accueil de la solution mobile My Free :



Figure 2 : Interface d'accueil de My Free

⁴ : <https://www.free.sn/my-free-appli-mobile>

2.1.1.3 myWari⁵

Wari est une entreprise de transfert d'argent et de services, créée en 2008 à Dakar. L'entreprise propose à ses clients une solution mobile pour faciliter l'utilisation du service. Leur offre s'est développée, et permet de réaliser de nombreux paiements auprès d'entreprises et d'institutions, parmi lesquelles des fournisseurs d'abonnements (chaines de télévision, téléphone, internet). La particularité de Wari est l'élargissement de ses services vers l'international en signant des partenariats avec quelques entreprises tel que MasterCard, WorldRemit et également le réseau social WhatsApp.

Ainsi pour faciliter l'utilisation de leurs services, l'entreprise a développé l'application **myWari**⁶ disponible sur Google Play Store et App Store. La figure suivante présente l'interface de l'application myWari.

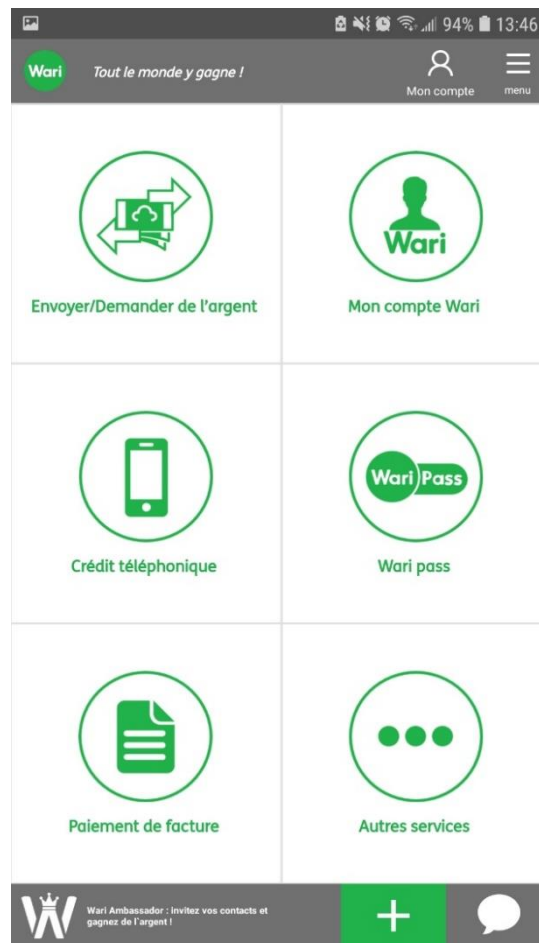


Figure 3 : Interface de myWari

⁵ : <https://www.wari.com/fr/>

⁶ : <https://play.google.com/store/apps/details?id=com.wari.mywariapp&hl=fr>

2.1.1.4 Wave⁷

Wave Money est un moyen innovant de transférer de l'argent. Vous pouvez envoyer et recevoir de l'argent en toute sécurité via votre téléphone portable ou depuis l'un des agents Wave Shop de votre quartier. Il est possible d'acheter du crédit pour free, orange ou expresso. Avec Wave Sénégal tu peux recevoir de l'argent venant de la France ou des USA vers le Sénégal grâce à l'application **SENDWAVE** disponible sur Google Play store et App store.

- ◆ **Si le destinataire a un compte Wave:** Il reçoit l'argent directement sur son compte et peut le retirer chez l'agent Wave de son choix. Une notification t'avertit par sms ou directement sur l'application.
- ◆ **Si le destinataire n'a pas de compte Wave :** Il reçoit un sms de notification avec un code. Il pourra donc retirer son argent sur un point Wave muni de sa pièce d'identité et du code.

L'application **Wave**⁸ est disponible gratuitement en téléchargement sur Play Store et App Store. Ci-dessous est représentée l'interface de l'application mobile de Wave.

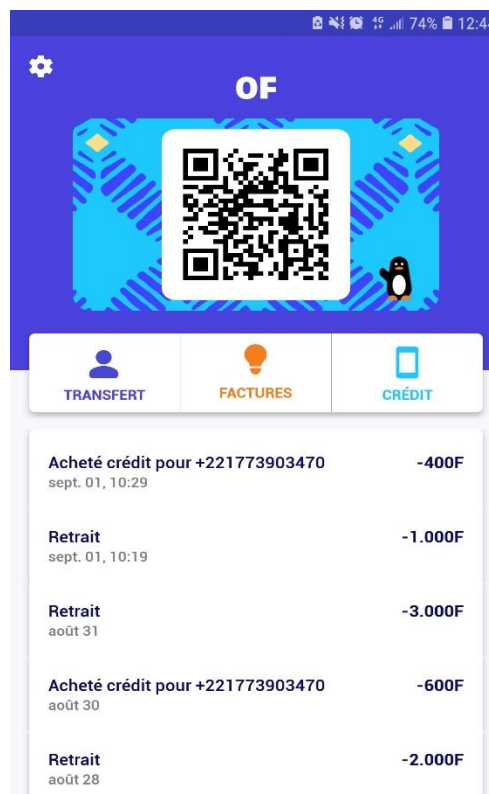


Figure 4 : Interface de l'application Wave

⁷ : <https://senegalndiaye.com/wave-la-nouvelle-solution-de-transfert-dargent-au-senegal/>

⁸ : <https://play.google.com/store/apps/details?id=com.wave.personal&hl=fr>

2.1.2 Les solutions Mobile Banking (m-banking)

Le Mobile Banking est un service bancaire mobile, parfois simplifié en banque mobile ou banque sans fil, correspond aux services financiers accessibles depuis un téléphone portable ou un autre appareil portable connecté. C'est en quelque sorte une adaptation du canal de banque à distance classique de « e-banking » sur terminal mobile. D'autre part l'e-banking est l'ensemble des services bancaires assurés par voie électronique « electronic banking » et donc par Internet tels que demander des informations, commander chéquier, service bancaire, consultation de solde, virements et achats de produits financiers. Le mobile banking joue un grand rôle dans le développement des banques au Sénégal et en Afrique. Toutes les structures se lancent dans ce domaine pour faciliter les transferts d'argent et également être plus proche du client.

Nous énumérons quelques solutions de m-banking qui existent au sein de certaines banques du pays. Toutes ces applications disponibles sur Google Play Store et App Store permet aux clients de :

- Consulter leurs comptes ;
- Effectuer des virements dans les banques qui acceptent le service;
- Commander et de suivre leurs moyens de paiements ;
- Identifier l'agence ou le GAB le plus proche via un système de géolocalisation intégré ;
- Prendre un rendez-vous ou communiquer avec leurs gestionnaires de compte ;
- Payer leurs factures SENELEC, SDE, ... ;
- Acheter du crédit téléphonique chez tous les opérateurs au Sénégal ;
- Transférer de l'argent à une personne non bancarisée

2.1.2.1 Tamweel Touch⁹

Ne voulant pas rater le train de la digitalisation et de l'offre de produits et de services, la Banque Islamique du Sénégal a lancé une nouvelle application dénommée « Tamweel Touch¹⁰ ».

L'application va faciliter l'accès et la gestion de compte bancaire via le smartphone. En plus des services classiques de banque à distance, l'outil Tamweel Touch donne la possibilité aux clients d'être en contact avec leurs gestionnaires et de disposer d'un portefeuille électronique.

⁹: <https://www.socialnetlink.org/2019/07/16/mobile-banking-tamweel-touch-la-nouvelle-application-de-la-banque-islamique-du-senegal/>

¹⁰ : <https://play.google.com/store/apps/details?id=com.taholding.banqueislamique&hl=fr>

« Ce nouveau produit est en parfaite corrélation avec les ambitions de la BIS (Banque Islamique du Sénégal), à savoir renforcer la bancarisation avec une qualité de service optimale en digitalisant le parcours client pour une meilleure célérité, une meilleure sécurité et une meilleure traçabilité de leurs transactions » [19] a souligné Monsieur Oumar MBODJ, Directeur Général de la BIS lors du lancement à Dakar.

L'application disponible en téléchargement sur Google Play Store et App Store est représentée par l'interface ci-dessous :



Figure 5 : Interface de Tamweel Touch (BIS)

2.1.2.2 Ecobank MobileMoney¹¹

Avec Ecobank MobileMoney, vous envoyez et retirez de l'argent, effectuez des paiements et achetez du crédit téléphonique. Il s'agit d'un portefeuille électronique sur votre téléphone, qui associe celui-ci rapidement et en toute sécurité à toutes vos transactions bancaires. Disponible dans 33 pays africains, on peut effectuer des virements à l'international pour un client de la

¹¹ : <https://www.ecobank.com/personal-banking/ways-to-bank/mobilemoney?hl=fr>

banque. Il est également possible de lier l'application à la carte GAB afin d'en effectuer des transactions.

L'application **Ecobank MobileMoney** est disponible en téléchargement sur Google Play Store et App Store. Voici l'interface de l'application :



Figure 6 : Interface d'Ecobank MobileMoney

2.1.2.3 Atlantique Mobile¹²

Atlantique Mobile est un service innovant, performant et intuitif déployé par la banque atlantique, permettant de gérer les comptes et profiter du meilleur de la banque depuis votre smartphone ou tablette.

L'application mobile de cette banque vous offre une multitude de fonctionnalités à portée de main pour accéder à nos produits et services et optimiser vos transactions bancaires 24h/24 et 7j/7. La particularité majeure de cette solution est la gestion de cartes bancaire des utilisateurs: activation /désactivation, personnalisation de plafonds de vos cartes de débit. Possibilité de

¹² : <https://www.banqueatlantique.net/atlantique-mobile/>

transférer de l'argent sans carte bancaire et vers n'importe quel bénéficiaire qui pourra le retirer depuis un GAB du réseau Banque Atlantique dans toute la zone UEMOA.

En plus d'être rapide et efficace, l'Application Atlantique Mobile vous garantit une :

- ✚ Sécurité : Cryptage des données et respect des plus hauts standards bancaires ;
- ✚ Mobilité : Service disponible à tout moment et n'importe où ;
- ✚ Fiabilité : Solution bancaire à la pointe de la technologie ;
- ✚ Simplicité : Utilisation facile et conviviale des fonctionnalités.

L'application Atlantique Mobile est disponible gratuitement en téléchargement sur Google Play Store et App Store. Voici l'application mobile de la banque :

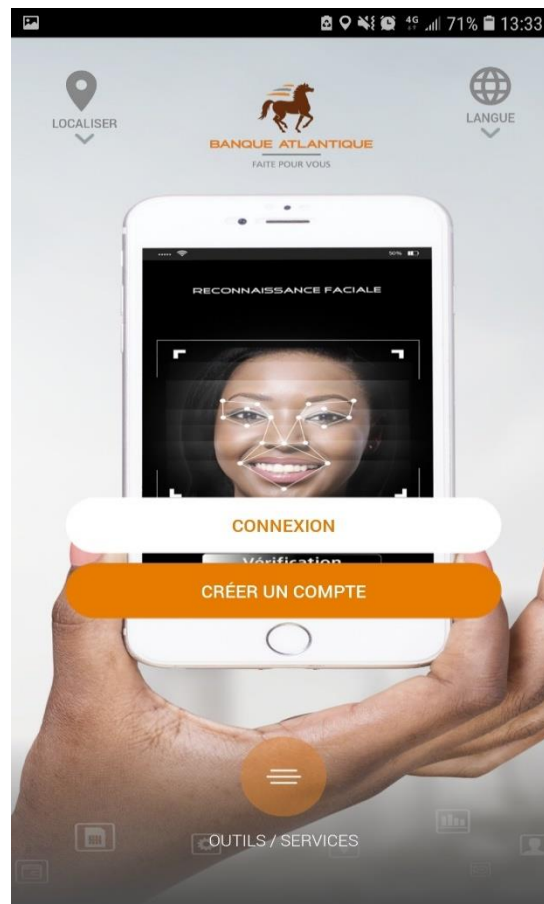


Figure 7: Interface d'Atlantique Mobile

2.1.2.4 SG Connect¹³

SG Connect est un outil de banque à distance déployé par la SGBS, disponible en version mobile ou depuis votre ordinateur. SG Connect offre la possibilité de consulter le solde de vos

¹³ : <https://societegenerale.sn/fr/particuliers/banque-distance/connect/>

comptes bancaires, d'accéder à un ensemble de services à forte valeur ajoutée (virements, achat de crédit de communication, alertes, paiement de factures) et bien d'autres avantages...

Les opérations sont traitées en temps réels. En termes de fiabilité, l'application assure la confidentialité et la sécurité des opérations grâce à un mot de passe personnalisable mais également l'unicité du terminal de connexion (impossible de se connecter à l'application à partir d'un autre smartphone en cas de vol des identifiants). Ci-dessous est présentée de l'interface de SG Connect.



Figure 8 : Interface de SG Connect

2.1.2.5 CBAO Mobile¹⁴

Pour faciliter la gestion de vos comptes, CBAO Mobile permet à ses utilisateurs de consulter et de gérer vos comptes en toute confidentialité et en toute sécurité à distance 24h/24 et 7j/7. Le téléchargement et l'accès à l'application sont gratuits pour les clients disposant du Web Banking (gratuit) et requiert les même codes d'authentification.

¹⁴ : <https://apps.apple.com/sn/app/cbao-mobile/id1151116813>

Avec CBAO Mobile, vous pouvez désormais :

- ◆ Consulter les soldes en temps réel de tous vos comptes (chèque, épargne, ...) ;
- ◆ Consulter l'extrait de vos opérations ;
- ◆ Effectuer et suivre vos virements vers les comptes CBAO Groupe Attijariwafa Bank et confrères ;
- ◆ Commander votre chéquier.

Aussi, et sans saisie des codes d'accès, l'application permet d'accéder à :

- L'offre produit et service de CBAO Groupe Attijariwafa Bank¹⁵
- La démonstration de l'application CBAO Mobile
- L'annuaire des contacts utiles

L'application mobile est disponible sur Google Play Store et App Store et voici l'une de ses interfaces :

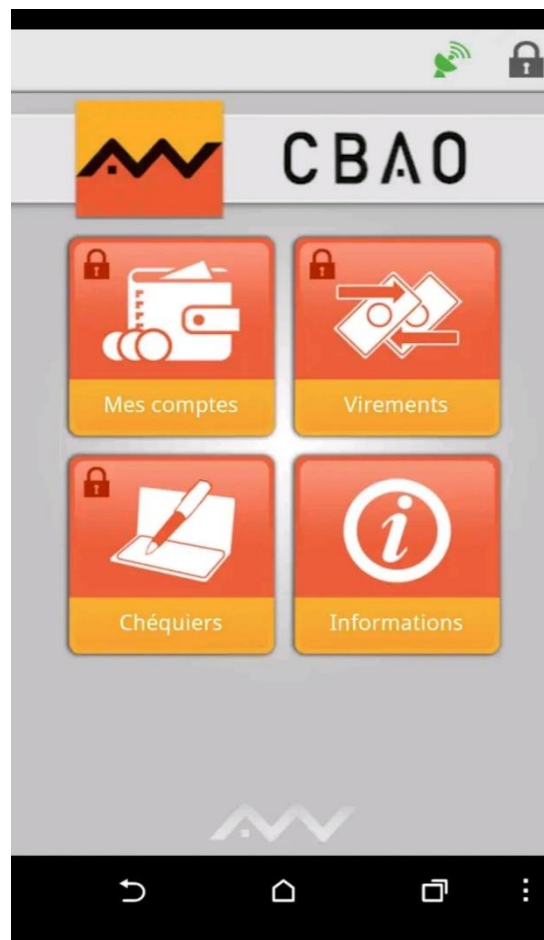


Figure 9 : Interface de CBAO Mobile

¹⁵ : <https://www.cbao.sn/fr/cbao/presentation>

2.1.3 Limites de l'existant

Il existe un bon nombre d'applications mobile de transfert d'argent au Sénégal, mais elles ne réussissent pas toutes à proposer à la fois un grand nombre de fonctionnalités et une interface moderne conviviale développée avec les dernières technologies innovantes. C'est tout à fait normal car aucun système d'information n'est pas réalisé pour combler tout un marché et satisfaire tous les besoins des clients en même temps. Néanmoins, il est important de signaler que ces systèmes de transfert ont convaincu un grand nombre d'utilisateurs et sont bien positionnés sur le marché malgré un certain manque qu'on les reproche parmi lesquels on peut citer :

- L'absence d'information sur les positions des points de vente Orange les plus proches,
- La gestion de la petite monnaie n'est pas prise en compte
- Les frais de transaction élevés à subir à chaque envoi d'argent

A travers ce projet, nous comptons apporter une solution face à ces limites afin de combler le manque que subi les autres applications du marché. Par conséquent, au lieu de se considérer comme étant un concurrent de ces systèmes déjà en fonctionnement, notre solution veut se comporter juste comme un complément de ces services. Pour cela, nous espérons pouvoir nouer des partenariats dans ce sens afin d'offrir aux clients un large éventail de possibilités dans le secteur du transfert d'argent.

2.2 Orientation

Après une présentation des différentes solutions existantes, il est primordial au regard de leurs limites de proposer une solution répondant aux besoins des utilisateurs concernant la petite monnaie. Il sera question de concevoir et développer un système d'information mobile ayant pour but principal d'accompagner les solutions en cours sur le marché dans un angle assez particulier. Ainsi notre système est sensé offrir aux utilisateurs un moyen facile et rapide de récupérer et d'utiliser leur petite monnaie. L'idée aussi c'est d'aider aux gens de bien faire circuler leur monnaie car il est constaté que certains ont du mal à porter leur pièce de monnaie à chaque fois et d'autres leur perdent facilement. D'où une nécessité d'en trouver une solution de garde (stockage) et de transfert de petite monnaie.

La particularité de notre future solution demeure sur le fait que les transactions concernent que les petites sommes d'argent et sans commission sur les transactions ce qui facilitera la fluidité d'échanges entre utilisateurs du système. Cependant, il existe des frais de tenue de compte à la

charge du client à la fin de chaque mois. Il sera ouvert à toute personne intéressée, les seuls éléments nécessaires pour l'utiliser sont un smartphone et une connexion à internet.

2.3 Système d'information

2.3.1 Notion de système d'information

La définition d'un système d'information varie selon les auteurs, de même que son appellation. Certains parlent de « système d'information pour le management », d'autres encore de « système de traitement d'informations ». Néanmoins ces définitions ont évolué en fonction de l'influence croissante de l'informatique dans les activités de l'organisation.

Ainsi, certains auteurs se sont plutôt focalisés sur le rôle de l'information dans la décision. Les plus célèbres sont R. Mason et I. Mitroff [20] qui ont considéré qu' « un système d'information comprend au moins une personne (avec son profil psychologique) dans un contexte organisationnel, confronté à un problème pour la résolution duquel (c'est-à-dire le choix d'une action) elle a besoin d'éléments qui lui sont fournis à travers un certain mode de présentation ». Dans cette perspective, le système d'information s'inscrit dans un processus de décision sous-jacent et apporte une aide au décideur.

D'autres auteurs ont proposé des définitions prenant en compte l'ensemble des activités organisées. Par exemple, G. Davis et M. Olson [21] ont défini un système d'information comme un « système utilisateur-machine intégré qui produit de l'information pour assister les êtres humains dans les fonctions d'exécution, de gestion et de prise de décision ». On observe, comme dans la définition précédente, un rôle du système d'information limité à la fourniture d'informations.

Les définitions vont progressivement s'élargir, traduisant le fait qu'un système d'information a dépassé le stade d'outil pour devenir l'élément structurant d'une organisation. Ainsi, quand R. Reix [22] le définit comme « un ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures permettant d'acquérir, traiter, stocker, communiquer des informations (sous forme de données, textes, images, sons, etc.) dans les organisations », on voit apparaître une notion essentielle : la procédure. Celle-ci décrit comment, quand et où le personnel est supposé utiliser matériel, logiciels et données pour que l'organisation soit informée.

2.3.2 Système d'information et système informatique

Un système d'information (figure 2) peut être défini comme « la partie du réel constituée d'informations organisées et d'acteurs qui agissent sur ces informations ou à partir de ces

informations, selon des processus visant une finalité de gestion et utilisant les technologies de l'information » [23]. Dans un projet, le maître d'ouvrage est responsable de la définition et la mise en œuvre du système d'information. La finalité de gestion oriente la définition du système d'information, c'est son objectif de production. Ainsi chaque système d'information est régi par l'interaction de ces trois entités ci-dessus :

- Les informations organisées sont celles que l'entreprise choisit de gérer, en général prédéfinies et structurées.
- Un acteur est un être humain ou une machine, qui crée, manipule, transforme les informations ou qui est sollicité par la présence ou la valeur de certaines informations.
- Le processus est un plan d'ensemble indiquant comment les acteurs collaborent au moyen des informations gérées pour accomplir l'objectif de production.

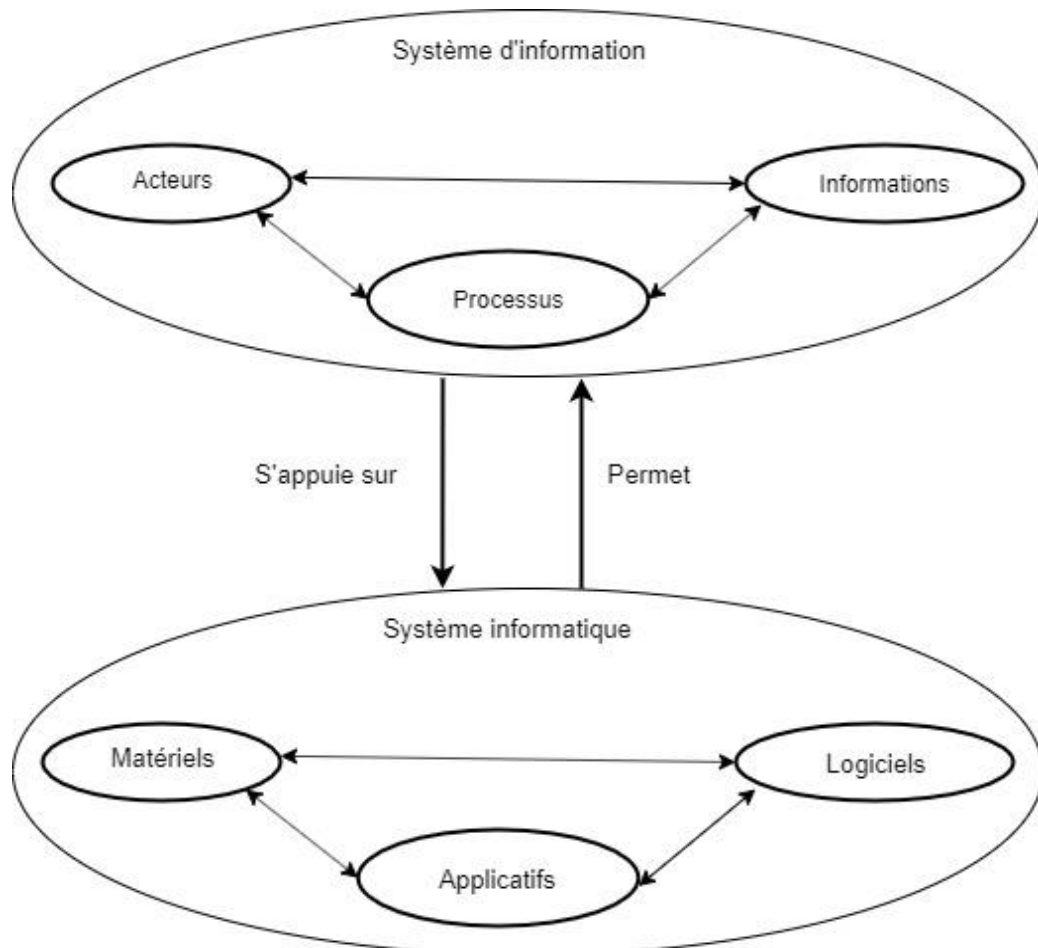


Figure 10 : Système d'information et système informatique

Un système informatique est « un ensemble organisé d'objets techniques, matériels, logiciels, applicatifs qui représentent l'infrastructure d'un système d'information » [23]. Dans un projet, le maître d'œuvre est responsable de la conception et la construction du système informatique.

Nous allons alors préciser les différentes étapes et processus indispensable à franchir pour la mise en place un système d'information mobile. D'abord, faire une étude approfondie des méthodes d'analyse guidant la reconfiguration des processus et leur informatisation. Ensuite, choisir la meilleure méthode qui correspond à l'architecture du système à modéliser.

2.3.3 Méthodologie de conception

Les méthodes d'analyse sont assez nombreuses et elles sont toutes spécifiques à un type de système d'information à réaliser. Le choix du processus de développement à utiliser se fait selon plusieurs critères tels que la complexité et le type de projet, le délai de livraison, le coût de développement et les compétences de l'équipe projet ou du maitre d'œuvre.

Un processus définit une séquence d'étapes, en parties ordonnées, qui concourent à l'obtention d'un système ou à l'évaluation d'un système existant [24].

L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles [24]. Généralement, il existe plusieurs processus de développement d'applications, après avoir parcouru ces processus et leurs caractéristiques, et en ayant connaissance de notre projet et ses besoins, nous avons opté pour le processus UP (Unified Process) qui semble adéquat pour ce genre d'applications.

2.3.3.1 Processus UP (Unified Process)

Le processus Unifié est un outil de réalisation ou d'évolution de logiciel entièrement basé sur UML, d'où son intérêt dans notre projet. Ce processus est constitué d'un ensemble de directives afin de produire un logiciel à partir des exigences. Chaque directive définit qui fait quoi et à quel moment. Il permet donc de structurer les différentes étapes d'un projet informatique [24], [25].

2.3.3.1.1 Principes fondamentaux du Processus Unifié(UP)

Le processus Unifié (UP) est un processus de développement logiciel itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques [25]. Ces principes sont les suivants :

- **Itératif et incrémental** : Le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. A la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale (validation et passage à une fonctionnalité à réaliser) [25].

- ***Centré sur l'architecture*** : Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facile. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et documentée [25].
- ***Piloté par les risques*** : Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations [25].
- ***Conduit par les cas d'utilisation*** : Le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés [25].

2.3.3.1.2 Les différentes phases du Processus Unifié

La gestion d'un tel processus est organisée à travers les quatre phases suivantes : initialisation, élaboration, construction et transition [25].

- ❖ ***La phase d'initialisation*** : Conduit à définir la vision du projet, sa portée et sa faisabilité afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.
- ❖ ***La phase d'élaboration*** : Poursuit trois objectifs principaux en parallèle : (1) Identifier et décrire la majeure partie des besoins des utilisateurs, (2) construire l'architecture de base du système et (3) enlever les risques majeurs du projet.
- ❖ ***La phase de construction*** : Consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels. C'est la phase la plus consommatrice en ressources (matérielles, financière et humaines) et en effort.
- ❖ ***La phase de transition*** : Permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux. Les mots clés sont : conversion des données, formation des utilisateurs, déploiement, bêta-tests.

La figure suivant montre les différentes phases et itérations du processus UP.

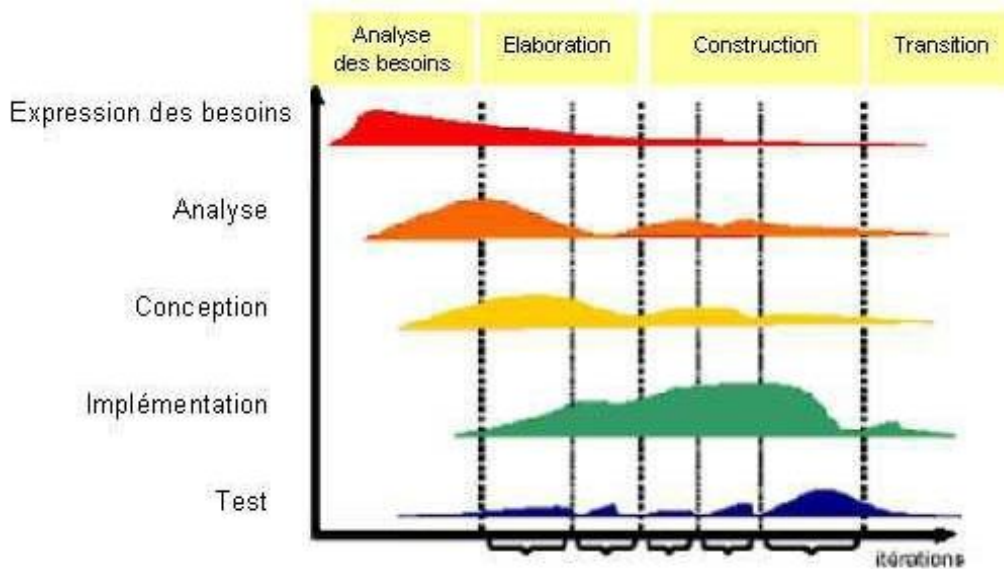


Figure 11: Phases et itérations d'UP

Les phases sont elles-mêmes divisées en itérations et produisent des versions du produit. Chaque itération produit une version du système (un jalon mineur) tandis que les phases définissent les grandes étapes du développement (les jalons majeurs), qui contrôlent ainsi le nombre d'itérations [25]. Chaque itération couvre les activités traditionnelles d'analyse des besoins, de conception, de réalisation et de test. En fait, les parts respectives de ces activités évoluent selon la phase considérée, les activités en amont sont plus poussées dans les itérations des premières phases, les activités en aval sont plus approfondies dans les itérations des phases ultérieures. Chaque itération donne lieu à un produit résultat. Plus le projet est grand, plus il y a d'itérations dans une phase [25]. À ces activités correspondent des modèles (cas d'utilisation, analyse, conception, déploiement, implantation et test) décrits avec la notation unifiée, une vue architecturale de ces modèles et des descriptions complémentaires. Nous retrouvons alors la notation UML (Unified Modeling Language), qui sert à décrire ces modèles. Nous traçons maintenant les grandes lignes de la corrélation entre le processus et la notation unifiée [25].

En plus des activités de développement, on trouve des activités support tel que la gestion de configuration et versions, la gestion de projet (organisation, risques, planification) et l'environnement du projet (support et méthode).

2.3.4 Un processus de modélisation avec UML

2.3.4.1 UML (Unified Modeling Language)

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [26]. La figure suivante présente l'historique d'UML qui est issu des travaux de trois auteurs.

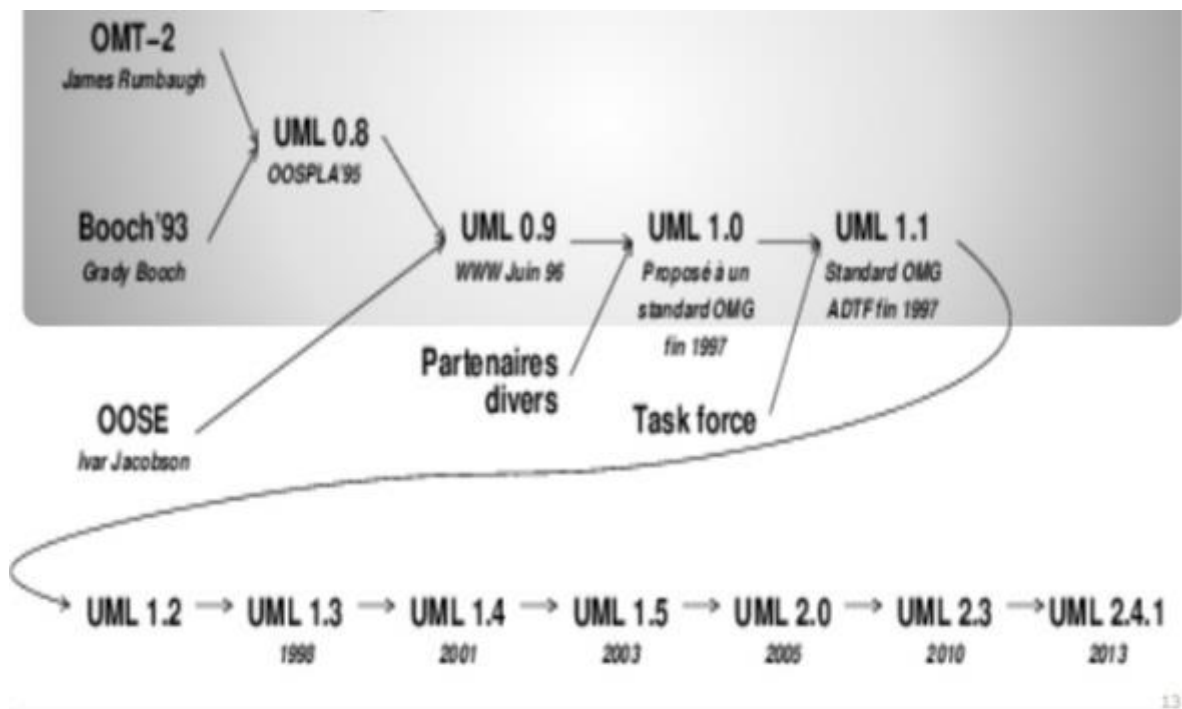


Figure 12: Historique d'UML

2.3.4.2 Bases d'UML

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. Ce langage est certes issu du développement de logiciel mais pourrait être appliqué à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système [27].

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la spécification jusqu'au codage. Dans ce cadre, un concept appartenant aux besoins des utilisateurs projette sa réalité dans le modèle de conception et dans le codage [27].

2.3.4.3 Différents types de diagrammes d' UML

UML s'articule maintenant autour de 13 diagrammes différents. Chacun d'eux est dédié à la représentation des diverses rubriques d'un logiciel en terme d'actions, d'acteurs, de fonctionnalités et données manipulées, suivant un point de vue particulier. Ainsi, UML modélise son système à travers deux modes de représentation : l'un concerne la structure du système et l'autre concerne sa dynamique de fonctionnement. Les deux représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composants fonctionnent entre elles [26].

La figure suivante présente les différents types de diagramme d'UML.

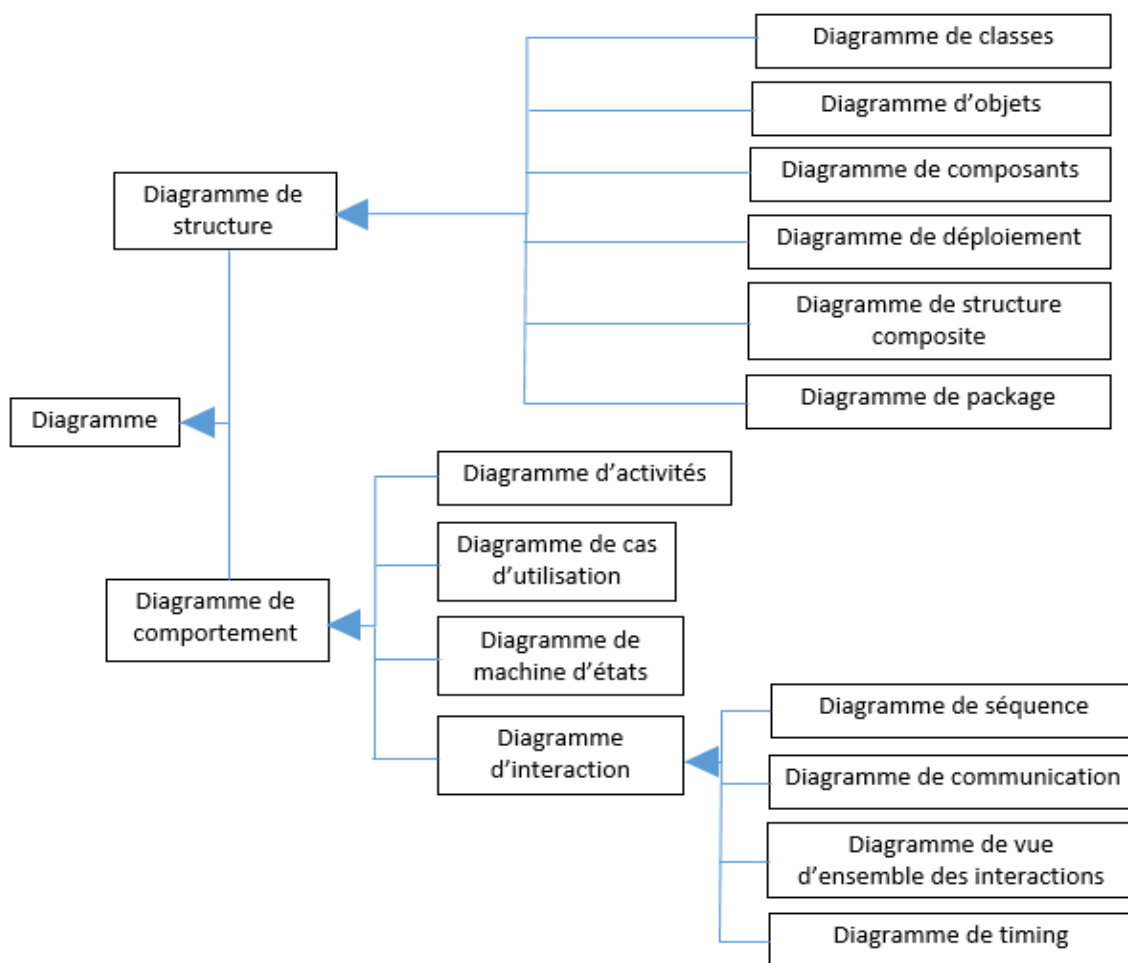


Figure 13: Types de diagrammes d'UML

Dans notre projet, nous allons utiliser seulement les diagrammes suivants :

- **Le diagramme de cas d'utilisation** représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est utilisé dans les deux étapes de capture des besoins fonctionnels et techniques [26].
- **Le diagramme de séquence** est un diagramme d'interaction, il représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système. Il sert ensuite à développer davantage les scénarios d'utilisation du système [26].
- **Le diagramme de classe** a toujours été le plus important dans toutes les méthodes orientées objet. C'est également celui qui contient la plus grande gamme de notations et de variantes, centralise l'organisation des classes de conception, c'est lui qui se transforme le plus aisément en code [26].
- **Le diagramme d'activité** permet de mettre l'accent sur les traitements. Il est donc particulièrement adapté à la modélisation du cheminement de flots de contrôle et de flots de données [26].
- **Le diagramme de composants** sert à représenter une vue logique d'une architecture en définissant les principaux composants sans se soucier des détails physiques de déploiement [26].
- **Le diagramme de déploiement** décrit la répartition physique des fonctions-métiers du système [26].

Conclusion

Nous avons présenté, tout au long de ce chapitre, les principaux concepts liés à notre projet. Nous avons également mené une étude sur l'existant avant de montrer leurs limites. Ce qui nous a permis de déterminer les méthodes et technologies adéquates à adopter dans notre solution et dégager les améliorations à apporter lors de la réalisation de notre projet.

L'intérêt de mener une étude sur les solutions déjà déployées sur le marché est de voir dans quel sens orienter notre travail. Cette étude permet également de savoir qu'il s'agit d'une amélioration d'une solution en cours d'utilisation, d'une innovation ou même d'un nouveau projet qui n'a pas encore été réalisé.

Dans la suite du travail, nous envisageons d'élaborer un chapitre intitulé "les applications mobiles" afin d'enrichir nos connaissances et de pouvoir utiliser l'ensemble de ces acquis pour le développement de notre application.

Chapitre 3 :

Les applications mobiles : technologies et outils de développement

Introduction

Aujourd'hui, les applications mobiles couvrent une grande variété de domaines tels que la communication, l'échange de fonds, le divertissement et le traitement de données. Ainsi, comme l'ordinateur, chaque téléphone fonctionne grâce à un système d'exploitation, et les applications leurs sont spécifiques.

Actuellement, il existe plusieurs systèmes d'exploitation; grâce à la concurrence les entreprises qui produisent ces systèmes se battent entre eux pour offrir le meilleur système. Ils améliorent leurs produits à chaque fois, c'est pour ça que nous trouvons plusieurs versions appartenant à un système d'exploitation. Il est alors important de cibler un ensemble de technologies à adopter et sur lequel orienter son projet de développement d'une application.

Dans ce chapitre nous allons décrire la notion d'application mobile, ses caractéristiques et types. Nous allons également présenter les systèmes exploitation mobiles puis les comparer.

3.1 Les applications mobiles

3.1.1 Définition

Une application mobile est un logiciel applicatif développé pour être installé sur un appareil électronique mobile, tel qu'un téléphone portable ou « smartphone » [29]. Ainsi, se basant sur le moyen d'acquisition de ces outils, d'autres le définissent comme étant « un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation du téléphone » [30].

Pour disposer d'une application sur un téléphone mobile, il existe différentes possibilités :

- Transfert depuis un ordinateur via un câble de connexion,
- A partir d'un service mobile (envoi de l'application ou de l'APK)
- Téléchargement des dépôts d'applications en ligne tels que :
 - Apple Store pour les smartphones compatibles avec le système IOS
 - Windows Market Place dédié au système Windows Mobile
 - AndroidMarket (Actuel Google Play Store) pour le système Android
 - etc.

3.1.2 Contraintes d'applications mobile

Il existe des contraintes techniques qu'il est nécessaire de prendre en compte lors de la conception d'une application mobile :

- ✓ Tailles d'écrans variables, pouvant dans certains cas être assez réduite
- ✓ Possibilité limitée de saisie de données
- ✓ Puissance du processeur, pouvant être limité sur les premiers smartphones
- ✓ Tailles de la mémoire pouvant varier
- ✓ Débits variables de la bande passante lors de la connexion à Internet.

Outre ces aspects techniques, il est également important de prendre en compte l'ergonomie de l'application mobile à réaliser, et c'est un point crucial à ne pas négliger. La conception d'une application mobile doit respecter certaines règles :

- Utiliser des images légères (taille de l'image en octet) ;
- Utiliser des éléments facilement accessibles (via la priorisation des éléments natifs du smartphone) ;
- Utiliser le langage JavaScript pour économiser la batterie (car léger et faible en consommation de ressources) ;
- Adapter le mode de saisie (langue, chiffres et caractères) des informations.

Les systèmes mobiles reposent sur cinq domaines de compétences :

- ✗ Techniques : Les systèmes d'exploitation (iOS, Android, Windows Phone) et les langages de développement (Objective-C, Java, C#, Kotlin, ...) ;
- ✗ Architecture : Performance, fiabilité, Intégration, sécurité, Evolutivité, etc. ;
- ✗ Design & Ergonomie ;
- ✗ Fonctionnelles spécifiques à la mobilité ;

3.1.3 Domaines d'application mobile

Avec les possibilités matérielles incorporées aux terminaux (caméra, GPS, gyroscope, ...), les applications sur smartphones et tablettes peuvent intégrer des fonctionnalités spécifiques et dédiées pour les utilisateurs, permettant ainsi d'enrichir le spectre fonctionnel et imaginer des usages non couverts jusqu'à présent par les systèmes d'information tels que :

- ★ **Géolocalisation et itinéraires** : En utilisant différents procédés de géolocalisation, ces fonctionnalités permettent :
 - d'améliorer l'expérience utilisateur en apportant une information localisée ;
 - d'afficher la localisation courante d'un utilisateur et de procéder à des recherches autour de soi ;

- d'accompagner l'utilisateur à travers une fonctionnalité de navigation ou d'itinéraire ;
- de proposer un service, un produit, suivant une information de géolocalisation.
- ★ **Scan de Code barre et QR Code** : Les Tags de type QR codes, Code Barre, Flash Code et autres peuvent contenir de l'information pour de nombreux usages :
 - Identification (inventaire, positionnement, jeux, ...)
 - Apport d'information complémentaire (texte, image, vidéo, ...)
- ★ **Réalité augmentée** : Proposer un nouveau moyen de visualisation informationnelle via des interfaces utilisateurs à la fois simples et enrichies et par la superposition, en temps réel, d'un modèle virtuel 3D ou 2D à la perception naturelle de la réalité. L'intérêt consiste notamment à :
 - Actionner un levier à la prise de décision (par exemple l'acte d'achat)
 - Aide à la décision pour l'utilisateur (orientation, indicateur...)
 - Apporter des éléments de jeux, d'attraction dans son espace réel.
- ★ **M-commerce, Paiement mobile** : Apporter une puissante relation commerciale à l'aide d'appareils mobiles comme :
 - Personnalisation et fidélisation (carte de fidélité numérique, ...)
 - Marketing géo-localisé (push marketing, diffusion de produits, ...)
 - Aide à la vente (tags, QR code, réalité augmentée)
 - Comparaison et suggestion de produits
 - Achat à distance, solutions de paiement
 - Recommandation sociale (commentaires, avis, évaluations de consommateurs, ...)
- ★ **Push et notification** : Le Push ou Notification est un envoi de messages, de sons à propos d'un événement interne à une application même si celle-ci n'est pas ouverte. C'est donc le serveur qui pousse des informations vers les smartphones et tablettes, éventuellement en fonction de critères liés au terminal, à l'utilisateur, la localisation, ... Ce mécanisme de notification est une alternative et/ou complément aux possibilités du principe des SMS.
- ★ **Analyse d'Audience** : Des informations et des analyses, en temps réel, apportent des données très utiles pour l'approche marketing, ou l'étude des usages des utilisateurs. Identifier les fonctions les plus utilisées, savoir exactement comment, où, quand et par qui l'application est utilisée. Identifier rapidement les éventuels problèmes.
- ★ **Gestion et Sécurisation de parc et de déploiement de terminaux mobiles** : Les plateformes de références permettent la diffusion gratuite ou payante d'applications

publiques. Afin de répondre au contexte d'applications internes, conçoit des stores privés, sécurisés et personnalisés pour restreindre la diffusion des applications d'une entreprise à ses employés, partenaires, ...

3.1.4 Les différents types d'application mobile

On peut distinguer plusieurs types d'applications mobiles : les applications natives, les applications web et les applications hybrides [31].

3.1.4.1 Application native

Une application mobile native est une application mobile spécifique à un système d'exploitation mobile, développée avec le langage et les outils associés fournis par l'éditeur du système d'exploitation mobile, et installée directement sur le mobile. Cette installation de l'application native se faisant soit au travers d'un téléchargement via Internet soit par déploiement depuis un ordinateur connecté au mobile [31].

L'aspect dominant de ces types d'application est l'accès à toutes les fonctionnalités du mobile, la rapidité d'exécution, la possibilité de notifier l'utilisateur, l'écosystème associé aux magasins (store) d'applications permettant de générer des revenus.

3.1.4.2 Application mobile web ou Web App

L'application mobile web est un site Internet qui a été adapté pour une restitution sur un navigateur de mobile. De la même manière que les sites web, une application mobile web est généralement placée sur un serveur HTTP et se manipule en actionnant des éléments d'interface XHTML à l'aide d'un navigateur web [32].

Leur particularité est qu'une application mobile web s'adresse à l'ensemble des utilisateurs de mobile disposant d'une connexion internet, et non une population spécifique utilisant une marque précise de mobile.

3.1.4.3 Applications hybrides

Une application hybride est un mélange de code natif et d'affichage de vues HTML/JavaScript. Concrètement toutes les plateformes mobiles proposent un composant de type WebView, permettant d'afficher du contenu web soit sur une partie de l'écran, soit en plein écran, et en utilisant le moteur HTML du navigateur intégré au système [29].

Ces applications hybrides peuvent être distribuées sur les stores des systèmes mobiles. Plusieurs stratégies sont alors possibles, selon que l'on place le curseur plus du côté natif ou plus du côté web :

- Ne réaliser que certains écrans voir même que certains composants d'IHM en HTML ;
- Réaliser tous les écrans en HTML mais garder la logique applicative en code natif ; notamment les effets de transitions entre écrans et la gestion du scrolling (faire défiler verticalement ou horizontalement une page afin de voir un contenu qui est en dehors du cadre de l'écran);
- Réaliser les écrans en HTML, et les transitions en JavaScript. Le code natif peut alors se cantonner à quelques composants techniques très ciblés. De la même manière, selon les applications, le logique métier peut être codé en JavaScript ou bien en code natif.

3.1.5 Comparaison entre application native et Web App

Caractéristiques	Application mobile native	Application mobile web
Portabilité	Développement spécifique à chaque plateforme	Navigateur web, mais une intégration distincte selon la plateforme
Référencement	Arriver à se positionner dans une boutique d'applications accessible par la recherche	Accessible par les moteurs de recherche classiques et liens externes éventuels
Accessibilité technique	Dépendre de la plateforme et de l'éventuelle validation par une boutique d'application offline possible	Eventuelle dépendance aux navigateurs. Mode offline (quasi) impossible, support HTML5 nécessaire
Exploitation du mobile	Utilise toutes les possibilités du mobile (GPS, contacts, caméra, voix, calendrier, etc.)	Se limite aux possibilités du navigateur
Développement / Coût	Plus long, plus fastidieux Nécessite un SDK (Software Development Kit) + la connaissance d'un langage spécifique	Généralement moins onéreux HTML / JavaScript / CSS
Effet immersif	Plus de possibilités, richesse fonctionnelle et multimédia ; Logique marketing forte	Des possibilités plus importantes avec l'arrivée de HTML5
Expérience utilisateur	Maximale. Possibilité de notifier l'utilisateur (Push) même quand l'application n'est pas utilisée	Limitée (peu de fonctionnalités) mais conforme à l'utilisation classique du Web
Mises à jour	Processus de soumission à un magasin d'applications.	Mise à jour instantanée sur le serveur Web

	Mises à jour en mode par action (A la volonté de l'utilisateur)	
Développement / Courbe d'apprentissage	Dépend des compétences existantes sur le SDK du mobile visé	Compétences HTML / CSS / JavaScript plus classiques
Potentiel performance	Maximum (traitement en local plus nombreux)	Dépend du développement du site, de la connexion, ...

Tableau 2 : Comparaison entre application native et Web App

3.1.5.1.1 Coût de mise en œuvre

Développer une application native pour plusieurs plateformes mobiles peut coûter cher, de par la multitude de langages et technologies mises en œuvre.

Selon le nombre de plateformes cibles, une technologie web ou même hybride sera souvent moins coûteuse. De plus il sera souvent plus simple de disposer de développeurs maîtrisant les technologies web que les diverses plateformes mobiles.

3.1.5.1.2 Qualité et rapidité des applications

Difficile de rivaliser avec les applications natives, Celles-ci seront presque toujours plus rapides. Mais cela dépend fortement du type d'applications (messagerie, échange de fonds, jeux) en jeu et de ses fonctionnalités.

3.1.5.1.3 Publication et mise à jour

Une importante contrainte des applications natives est que celles-ci doivent être approuvées avant diffusion sur leur store respectif, ce qui peut s'avérer long et contraignant. Une application hybride permet de limiter ce désagrément, et une webApp de s'en affranchir complètement.

Le même problème se pose pour les mises à jour, il n'est souvent pas possible de diffuser un patch correctif en urgence ou même rapidement sur une application native. Cela est dû au fait que le code est utilisé par deux plateformes en même temps. Il faudra alors apporter des corrections similaires, et de procéder aux mises à jour sur les deux plateformes.

3.1.5.1.4 Monétisation

Les magasins d'applications permettent très facilement de vendre les applications, mêmes si Apple, Google et consorts prélèvent leur part sur les prix de vente, généralement autour de 30%.

Les stores servent aussi de moteur de recherche et de vitrines pour les applications, et permettent ainsi de les mettre en avant et de les faire découvrir.

3.2 Technologies et outils de développement

3.2.1 Les systèmes d'exploitation

3.2.1.1 Définition

Le système d'exploitation de téléphone mobile est un ensemble de programmes responsables de la liaison entre les ressources matérielles de l'appareil et ses applications logicielles. Il assure le démarrage et le fonctionnement de téléphone mobile [33].

Ainsi de manière général, sur un ordinateur ou un smartphone, le **système d'exploitation** (noté *SE* ou *OS*, abréviation du terme anglais *Operating System*) est chargé d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications (traitement de texte, jeu vidéo, ...). Ainsi lorsqu'un programme désire accéder à une ressource matérielle, il ne lui est pas nécessaire d'envoyer des informations spécifiques au périphérique, il lui suffit d'envoyer les informations au système d'exploitation, qui se charge de les transmettre au périphérique concerné via son pilote. En l'absence de pilotes il faudrait que chaque programme reconnaisse et prenne en compte la communication avec chaque type de périphérique. [33]

Le système d'exploitation permet ainsi d'associer les programmes et le matériel, afin notamment de simplifier la gestion des ressources et offrir à l'utilisateur une interface homme-machine (notée «IHM») simplifiée et de lui permettre de s'affranchir de la complexité de la machine physique [33].

3.2.1.2 Rôles d'un système d'exploitation mobile

Les rôles du système d'exploitation des smartphones sont divers :

- ✚ **Gestion des entrées/sorties** : Le système d'exploitation permet d'unifier et de contrôler l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes (appelés également gestionnaires de périphériques ou gestionnaires d'entrée/sortie).
- ✚ **Gestion du processeur** : Le système d'exploitation est chargé de gérer l'allocation du processeur entre les différents programmes. .
- ✚ **Gestion de la mémoire vive** : Le système d'exploitation est chargé de gérer l'espace mémoire alloué à chaque application. La mémoire virtuelle permet de faire fonctionner des applications nécessitant plus de mémoire qu'il n'y a de mémoire vive disponible sur le système.

- ✚ **Gestion de l'exécution des applications** : Le système d'exploitation est chargé de la bonne exécution des applications en leur affectant les ressources nécessaires à leur bon fonctionnement.
- ✚ **Gestion des fichiers** : Le système d'exploitation gère la lecture et l'écriture dans le système de fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications.

3.2.1.3 Système d'exploitation iOS d'Apple

iOS (anciennement **iPhone OS**) est le système d'exploitation mobile développé par Apple pour l'iPhone. Il est dérivé de Mac OS dont il partage les fondations (le kernel hybride XNU basé sur le micro-noyau Mach, les services Unix et Cocoa, etc.). iOS comporte quatre couches d'abstraction, une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ». Le système d'exploitation occupe moins de 512 Mo (mégaoctets) de la capacité mémoire de l'appareil [34].

L'App Store est la plateforme de téléchargement des applications mobiles iPhone. Selon les cas, les applications obtenues sur l'App Store sont gratuites ou payantes.

3.2.1.3.1 Caractéristiques

Le logiciel de l'iPhone est caractérisée par :

Le BaseBand : C'est donc un micro logiciel autonome qui s'occupe en temps réel de toutes les interactions avec les périphériques de communication de l'appareil : Bluetooth, Wi-Fi et GSM. [34]

Le BootLoader : C'est une partie du BaseBand, dont le rôle principal est d'assurer le démarrage de l'iPhone, de contrôler son activation, et sa compatibilité avec la carte SIM insérée [34].

Le firmware : Il s'agit d'un logiciel interne de l'appareil, cette fois responsable de la gestion de sa partie systémique (l'écran, le clavier tactile, etc.) [34].

Le SeckPack : C'est une partie de la mémoire flash de l'appareil contenant entre autres des informations sur le verrouillage de celui-ci [34].

3.2.1.3.2 Versions

Le tableau suivant contient l'ensemble des versions d'iOS depuis la première jusqu'à nos jours

Nom	Numéro version	Date
iOS 1	1.0	29 Juin 2007
	1.1.5	15 Juillet 2008
iOS 2	2.0	11 Juillet 2008
	2.2.1	27 Juin 2009
iOS 3	3.0	17 Juin 2009
	3.2.2	11 Août 2010
iOS 4	4.0	21 Juin 2010
	4.2.10	25 Juillet 2011
iOS 5	5.0	6 Juin 2011
	5.1.1	25 Mai 2012
iOS 6	6.0	11 Juin 2012
	6.1.6	21 Février 2014
iOS 7	7.0	10 juin 2013
	7.1.2	30 Juin 2014
iOS 8	8.0	2 Juin 2014
	8.4.1	13 Août 2015
iOS 9	9.0	8 Juin 2015
	9.3.6	28 Août 2016
iOS 10	10.0	13 Juin 2016
	10.3.3	19 Juillet 2017
iOS 11	11.0	5 Juin 2017
	11.1	31 Octobre 2017
iOS 12	12.0	17 Septembre 2018
	12.4.8	15 Juillet 2020
iOS 13	13.0	19 Septembre 2019
	13.7	1er septembre 2020
iOS 14	14.0	10 juillet 2020

Tableau 3 : Historique des versions d'iOS

3.2.1.4 *Android de Google*

Android est un système d'exploitation open source utilisant le noyau Linux, pour terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 5 novembre 2007. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des radio-réveils, des autoradios et même des voitures [30].

Google Play, anciennement dénommé AndroidMarket, est le magasin en ligne de Google. Celui-ci permet de télécharger des logiciels, des livres, des films ou de la musique, payants ou non.

3.2.1.4.1 Caractéristiques

- ◆ Android est un système d'exploitation fondé sur un noyau Linux.
- ◆ Il comporte une interface spécifique, développée en Java, toutefois il est possible de passer outre cette interface, en programmant ses applications en C, mais le travail de portabilité en sera plus important.
- ◆ La majorité des périphériques Android sont basées sur l'architecture ARM.
- ◆ Android a été conçu pour intégrer au mieux des applications existantes de Google comme le service de courrier Gmail, celui de cartographie, Google Maps, ou encore Google Agenda, Google Talk, YouTube. Un accent particulier est mis sur la géolocalisation avec Google Latitude et la météo correspondant à la ville la plus proche disponible sur le menu principal.

3.2.1.4.2 Versions

L'histoire d'Android commence en octobre 2003, lorsqu'Andy Rubin, Rich Miner, Nick Sears et Chris White créent la société Android à Palo Alto (Californie). Google a racheté la société en août 2005. Deux ans plus tard, l'Open Handset Alliance est annoncée et Android devient officiellement open source.

La première version du SDK Android 1.0 sort en 2008 avec le premier téléphone sous Android (HTC Dream). En avril 2019, la version 1.5 (API 3) d'Android sort. Cette version baptisée Cupcake (petit gâteau) inaugure les nouveaux noms des versions d'Android ce qui donnera pour les futures versions comme le montre la figure dessous.

Nom	Numéro version	Date	Niveau API
-----	----------------	------	------------

Première version	1.0	Septembre 2008	1
	1.1	Février 2009	2
Cupcake	1.5	Avril 2009	3
Donut	1.6	Septembre 2009	4
Eclair	2.0 – 2.1	Octobre 2009	5 - 7
Froyo	2.2 – 2.2.3	Mai 2010	8
Gingerbread	2.3 – 2.3.7	Décembre 2010	9 - 10
Honeycomb	3.0 – 3.2.6	Février 2011	11 - 13
Ice Cream Sandwich	4.0 – 4.0.4	Octobre 2011	14 - 15
Jelly Bean	4.1 – 4.3.1	Juin 2012	16 - 18
Kitkat	4.4.x	Octobre 2013	19 - 20
Lollipop	5.0 – 5.1.1	Octobre 2014	21 - 22
Marshmallow	6.0-6.0.1	Mai 2015	23
Nougat	7.0-7.1.1	Septembre 2016	24 - 25
Oreo	8.0	août 2017	26 - 27
Pie	9.0	Août 2018	28
Android 10	10.0	Septembre 2019	29

Tableau 4: Historique des versions du système Android

3.2.1.5 Windows phone

Windows Phone est un des systèmes d'exploitation pour smartphone développé par Microsoft [35].

3.2.1.5.1 Caractéristiques

- ◆ Windows Phone est pratiquement léger et prévu pour fonctionner sur des périphériques à faible empreinte mémoire. La taille d'une image système sans support graphique peut être de l'ordre de 4 Go.
- ◆ Windows Phone peut fonctionner sans support de stockage, exclusivement en RAM/ROM.
- ◆ Windows Phone est un système conçu pour fonctionner sur architecture ARM, x86, MIPS, SHx,
- ◆ Il répond aux contraintes d'un système temps-réel avec un ordonnanceur préemptif à 256 niveaux de priorités fixes et un temps de latence des interruptions déterministe. Sous certaines conditions, un mécanisme

d'inversion de priorité peut être pris en charge par l'OS. L'unité d'exécution est le thread,

- ◆ Windows Phone prend en charge un environnement graphique basé sur une mécanique événementielle (GWES - GraphicWindowing and Events Subsystem) et intégrant une couche GDI (GraphicDevice Interface) allégée par rapport à celle disponible sur XP,
- ◆ Communication: Bluetooth, IrDA, Ethernet, Wifi, NDIS 5.1, TCP/IP,

3.2.1.5.2 Versions

- ◆ Windows Phone 7 est la première version du système d'exploitation, elle est publiée le 21 octobre 2010. Déjà en 2011, Microsoft travaillait sur une unification entre Windows et Windows Phone. Irrémédiablement, un changement de noyau aura donc lieu prochainement dans Windows Phone. Le but est d'avoir le même que sur Windows 8, c'est-à-dire le noyau NT. Dès 2011, les développeurs savaient donc déjà qu'ils ne pourraient pas faire tourner Windows Phone 8 sur les mobiles disposant de Windows Phone 7.

Depuis le 9 septembre 2014, le support pour Windows Phone 7 n'est plus assuré. Seul des mises à jour de sécurité seront encore proposées jusqu'au mois de septembre 2019.

- ◆ Un an après, au Mobile World Congress de 2011, Steve Ballmer annonce une mise à jour majeure pour Windows Phone 7 : Windows Phone 7.5, nom de code Mango. La nouvelle version de l'OS aurait pour principales exigences de combler les lacunes de jeunesse de la plateforme. Techniquement le système est toujours en version 7.1 mais il sera toujours dénommé par son nom commercial Windows Phone 7.5.

Publiée à partir du 27 septembre 2011, cette nouvelle version apporte de très nombreuses nouveautés et améliorations de performances.

- ◆ Les développeurs le savaient, les téléphones sous Windows Phone 7 ne pourront pas migrer vers Windows Phone 8. C'est pourquoi, lorsque Microsoft annonce Windows Phone 8, il assure que les dispositifs sous Windows Phone 7.x recevront une mise à jour gratuite vers Windows Phone 7.8.
- ◆ Windows Phone 7.8 apporte donc la plupart des nouvelles fonctionnalités esthétiques de Windows Phone 8 mais les avancées techniques ne pourront jamais être ajoutées et l'OS n'évoluera plus après cette mise à jour.
- ◆ Le changement de noyau entre Windows Phone 7 et Windows Phone 8 est donc un mal pour un bien, en effet Windows Phone 8 dispose désormais du potentiel de la structure

NT. Cela permet évidemment des nouveautés esthétiques mais surtout de nombreuses avancées techniques avec par exemple l'intégration de Windows Update permettant les mises à jour Over the air.

- ◆ Les mises à jour proposées sous Windows Phone 8 sont appelées General Distribution Release (GDR), elles se présentent sous la forme d'un pack de mises à jour cumulatives. Le déploiement s'étale sur plusieurs semaines. Il commence généralement sur les mobiles achetés nus et dépend du constructeur. Il se poursuit ensuite sur les mobiles achetés chez les différents opérateurs et dépend alors de la rapidité de ceux-ci. En effet, les opérateurs ajoutent leurs applications spécifiques et des paramétrages particuliers nécessitant une phase de test supplémentaire. Ces modifications sont le plus souvent des ajouts mais peuvent aussi aboutir à la désactivation de certaines fonctionnalités. SFR est par exemple le dernier opérateur français avoir fourni la messagerie vocale visuelle en novembre 2014, alors que la fonction était proposée depuis le lancement de Windows Phone 8.
- ◆ Windows Phone 8.1 est une Version publiée avec le programme Preview For Developers, déployée sur certains modèles,
- ◆ Windows Phone ne recevra plus de mise à jour à proprement parler, les nouveaux téléphones fonctionnent désormais sous l'OS unifié de Microsoft Windows 10 qui est en constante évolution.

3.2.2 La plateforme Java 2 Micro Edition (J2ME)

3.2.2.1 Présentation de J2ME

Une plateforme est une base de travail à partir de laquelle on peut écrire, lire, utiliser, développer un ensemble de logiciels. Pour le développement des applications sur le système mobile, la technologie généralement utilisée est la technologie J2ME de Sun utilisant le langage Java.

Java 2 Micro Edition est une architecture technique dont le but est de fournir un socle de développement aux applications embarquées. L'intérêt est de proposer toute la puissance d'un langage tel que Java associé aux services proposés par une version bridée du framework J2SE : J2ME. Les terminaux n'ayant pas les mêmes capacités en terme de ressources que les ordinateurs de bureau classiques (mémoire, disque et puissance de calcul), la solution passe par la fourniture d'un environnement allégé afin de s'adapter aux différentes contraintes

d'exécution. Cependant, comment faire en sorte d'intégrer la diversité de l'offre à un socle technique dont la cible n'est pas définie à priori ? La solution proposée par J2ME consiste à regrouper par catégories certaines familles de produits tout en proposant la possibilité d'implémenter des routines spécifiques à un terminal donné. L'architecture J2ME se découpe donc en plusieurs couches :

- ❖ **Les profile** : Ils permettent à une certaine catégorie de terminaux d'utiliser des caractéristiques communes telles que la gestion de l'affichage, des événements d'entrées/sorties (pointage, clavier, ...) ou des mécanismes de persistance (Base de données légère intégrée). Ces profile sont soumis à des spécifications suivant le principe (coordonner l'évolution du langage Java et des technologies qui lui sont associées) du JCP (Java Community Process).
- ❖ **Les configurations** : Elles définissent une plateforme minimale en terme de services concernant un ou plusieurs profiles donnés.
- ❖ **Les machines virtuelles** : En fonction de la cible, la machine virtuelle pourra être allégée afin de consommer plus ou moins de ressources.
- ❖ **Le système d'exploitation** L'environnement doit s'adapter au système d'exploitation existant (Windows CE, Palm Os, SavaJe, ...).

3.2.2.2 Architecture de J2ME

Cette architecture en couches a pour but de factoriser pour des familles de produits données un ensemble d'API (Application Programming Interface) permettant à une application de s'exécuter sur plusieurs terminaux sans modification de code.

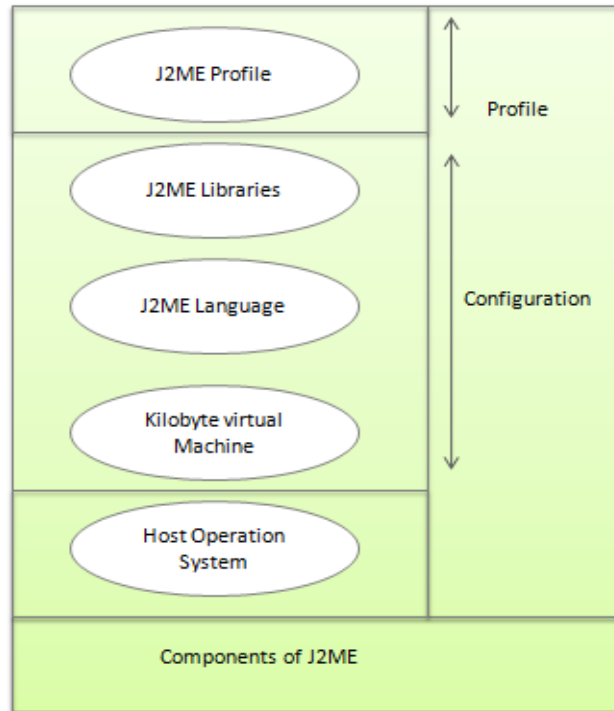


Figure 14 : Architecture de J2ME

Conclusion

Une application mobile réalisée dans une technologie n'est pas nécessairement compatible avec toutes les plateformes. Aussi une application est réalisée à partir de différents langages de programmation car il existe des environnements hétérogènes au niveau des plateformes installées sur les téléphones portables.

Ainsi, il existe pleins d'outils et de technologies développés qui permettant d'orienter la réalisation de solutions mobiles, mais le choix de nous outils de développement est fondé sur leur performance, leur adéquation au projet et aux objectifs préfixés qui sont de concevoir une application multi-plateforme, robuste et utilisant les récentes innovations.

Dans le chapitre suivant, nous allons aborder la partie conception décrivant la modélisation des besoins exprimés dans cette section.

Chapitre 4 :

La conception

Introduction

Après avoir mis le projet dans son cadre théorique, nous allons maintenant présenter sa spécification. Ainsi, le monde informatique a toujours souhaité disposer d'un processus qui puisse garantir le processus de développement efficace d'applications de qualité, quel que soit la grandeur et la complexité du projet, surtout que, de nos jours, les applications demandées sont de plus en plus imposantes et exigeantes qu'au paravent.

Le processus unifié déjà énoncé plus haut semble être la solution idéale pour remédier à l'éternel problème des développeurs. En effet, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en un système logiciel quel que soit la classe, la taille et le domaine d'application de ce système. Le processus unifié utilise le langage UML (Unified Modeling Language). Ce langage de modélisation est une partie intégrante du processus unifié

En effet, la première partie de ce chapitre consiste à énoncer les différents besoins fonctionnels auxquels devrait répondre l'application à réaliser, ainsi que les besoins non fonctionnels que l'application à développer devrait respecter. La seconde partie est consacrée à l'élaboration des différents diagrammes afin de détailler chacun des modules du projet conformément à la norme UML.

4.1 Cadrage du système à modéliser

4.1.1 Les acteurs du système

Notre système a comme fonction principale l'échange de petites monnaies entre utilisateurs. Cette fonction met en interaction quatre acteurs :

- **L'administrateur** : Il a pour rôle la gestion des utilisateurs, de régulariser l'accès, l'utilisation et le bon fonctionnement du système.
- **L'abonné** : C'est l'acteur principal du système qui consulte les différents services de l'application et en fait usage.
- **Les distributeurs** : Il s'agit également des abonnés du système mais ayant des privilèges plus avancés tels que l'activation de compte d'abonné, l'annulation d'une transaction ainsi que convertisseur de monnaie au sein des partenaires. Les utilisateurs se rendent auprès de ces distributeurs pour y déposer des espèces ou retirer de l'argent.
- **Les partenaires** : Ce sont des acteurs externes du système. Il s'agit des banques et autres services de transfert d'argent déjà existants qui n'agissent que lorsqu'on a besoin de

convertir notre monnaie électronique en espèces afin de faciliter aux distributeurs leur échange avec les abonnés.

4.1.2 **Spécification des besoins**

Dans cette partie, nous allons définir les différents besoins fonctionnels et non fonctionnels de notre système de transfert d'argent consacré aux petites monnaies.

4.1.2.1 *Besoins fonctionnels*

Les besoins fonctionnels doivent répondre aux exigences du futur système en termes de fonctionnalité. Ils constituent une sorte de promesse ou de contrat au comportement du système généré. De ce fait, la solution doit répondre aux besoins fonctionnels suivants :

- ❖ Gérer les utilisateurs : Mettre à jour les abonnées ainsi que l'attribution des droits d'accès.
- ❖ Permettre à tout utilisateur de pouvoir créer son compte. Cette fonctionnalité est obligatoire pour avoir l'accès à certaines fonctionnalités (échange de la petite monnaie) de l'application.
- ❖ Autoriser l'utilisateur d'accéder à son compte à travers son numéro de téléphone et son mot de passe pour assurer une certaine sécurité du système.
- ❖ Permettre l'abonné à alimenter son compte afin d'y effectuer un service au choix.
- ❖ Permettre aux utilisateurs de transférer et de recevoir de l'argent en toute sécurité et à tout moment.
- ❖ Permettre à l'utilisateur de consulter son solde.
- ❖ Permettre à l'utilisateur de faire une recherche sur un individu avec lequel il a déjà reçu ou transféré de la monnaie.
- ❖ Permettre à l'utilisateur de visualiser ces différentes transactions effectuées durant une période donnée.
- ❖ Permettre la modification et la réinitialisation de mot de passe d'un utilisateur.
- ❖ Permettre la modification des informations de compte en cas d'erreur de saisie pendant l'inscription.
- ❖ Permettre l'affichage des informations d'un utilisateur.
- ❖ Permettre à l'administrateur de maintenir, de consulter les différents mouvements des utilisateurs et ses avis sur l'application.
- ❖ Permettre au distributeur d'annuler un transfert à la demande d'un utilisateur

4.1.2.2 Besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ils peuvent être considérés comme des besoins fonctionnels spéciaux. Parfois ils ne sont pas rattachés à un cas d'utilisation particulier mais déterminent tout le système. Ce sont des besoins en matière de performance, de type de matériel ou de type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type de SGBD, de système d'exploitation, etc.), de sécurité, de temps de réponse, etc.

Dans le cadre de ce travail, le système devra être extensible c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier des fonctionnalités. Le système doit garantir les besoins opérationnels suivants :

- **Besoins matériels** : Le système doit s'exécuter de la même façon sur les différentes catégories de smartphones Android et iOS disponibles sur le marché.
- **Besoins de sécurité** : L'application doit être hautement sécurisée car les informations ne devront pas être accessibles à tout le monde.
- **Besoins de déploiement** : Le système doit s'adapter à tout moyen de connexion à internet (Wifi, données mobiles, etc.), assurer la facilité de la mise en place et du déploiement de l'application.
- **Besoins de performance** : Le système devra répondre rapidement au besoin de l'utilisateur :
 - ★ Délai d'acquisition de la première page au lancement de l'application inférieur à 5 secondes
 - ★ Un taux de rafraîchissement rapide : Pas de latence de l'affichage d'une recherche
- **Besoins de disponibilité et de fiabilité** : Le système doit :
 - ✖ Être disponible pour les utilisateurs lorsqu'ils le demandent, donc 24h/24, 7j/7 sauf période de maintenance et de mise à jour,
 - ✖ Fonctionner correctement lors de la demande de l'utilisateur
- **Besoins de robustesse** : Le système doit être robuste face à l'instabilité du signal Wifi ou des données mobiles du téléphone
- **Besoins de maintenance** : Le système doit être facile à installer et à maintenir
- **L'ergonomie des interfaces** : L'application doit présenter une interface claire, conviviale et simple à utiliser.

- **Le contrôle des champs** : L'application doit avoir un contrôleur des champs de saisie, pour éviter l'introduction des informations qui ne correspondent pas aux types des champs.

4.2 Les différents diagrammes

Dans ce qui suit, nous allons approfondir la description des besoins fonctionnels précédemment spécifiés par construction des diagrammes de cas d'utilisation correspondants avec une description de scénarios associés. Par la suite, nous proposons quelques diagrammes (séquences, d'activités, de composants, de déploiement et de classe) pour une meilleure visibilité de la solution à concevoir.

4.2.1 Le diagramme de cas d'utilisation

Les cas d'utilisation permettent de définir de manière normalisée les relations fonctionnelles entre les acteurs et le système étudié. Ils sont une représentation orientée « fonction » et permettent de modéliser les attentes des utilisateurs [26], [28]. La figure suivante présente le diagramme de cas d'utilisation global de notre solution.

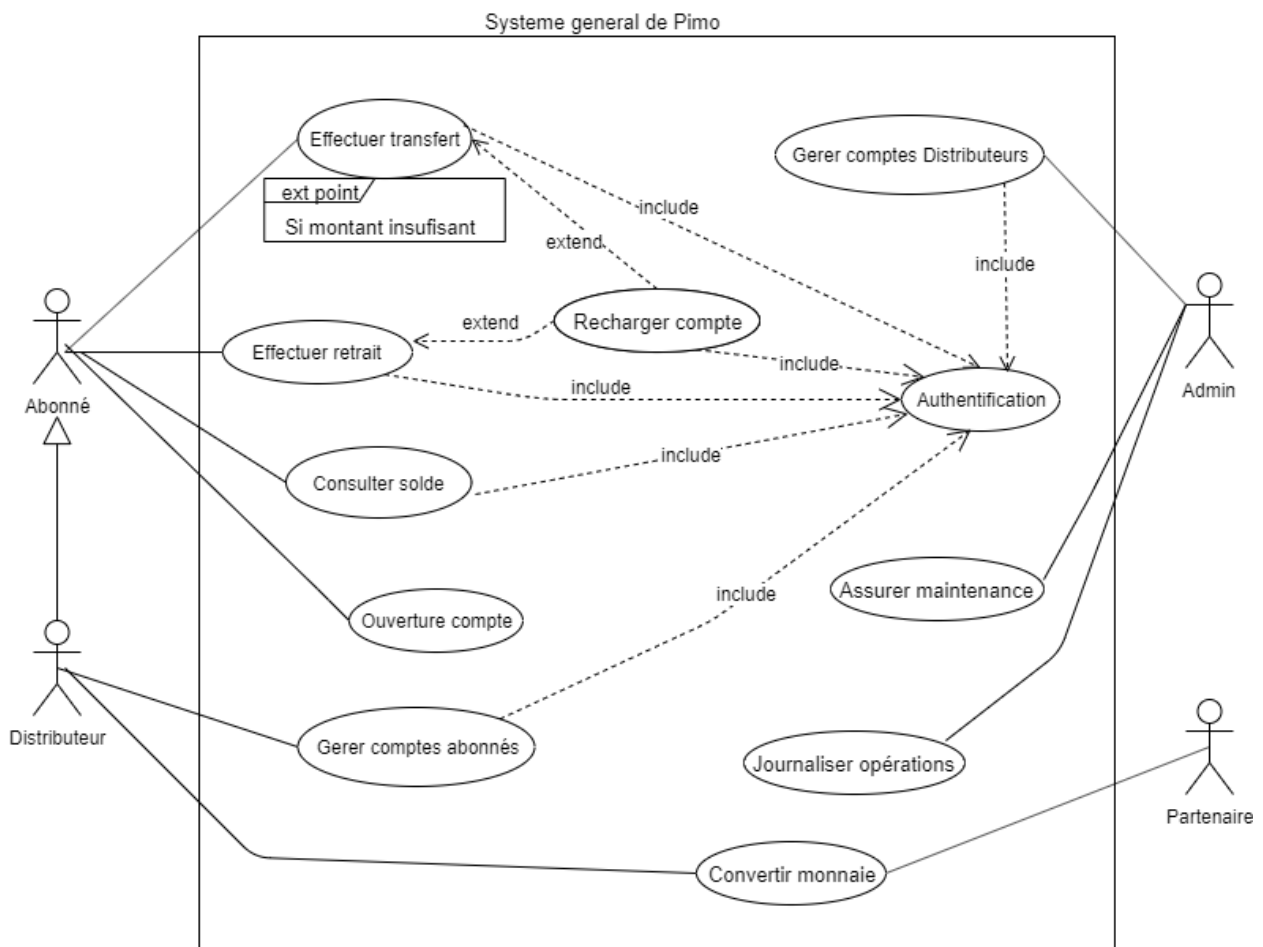


Figure 15 : Diagramme de Cas d'utilisation du système

La figure 15 illustre les différents cas d'utilisation de la solution à développer et les acteurs qui l'utilisent. Ce diagramme décrit une vue générale de l'ensemble des cas d'utilisation relatifs à notre système et éventuellement les différents scénarios susceptibles d'avoir lieu.

Une description détaillée des différents cas d'utilisation que nous avons développés (Ouverture de compte, Effectuer retrait, Effectuer transfert, Recharger compte, Gestion des comptes, Conversion de monnaie, etc.) sera alors faite dans la suite. Nous allons maintenant décrire de façon détaillée les cas d'utilisation afin d'obtenir une expression précise des besoins avant d'attaquer la conception. Pour détailler la dynamique du cas d'utilisation, la façon de procéder consiste à recenser textuellement toutes les interactions entre les acteurs et le système. Nous compléterons la description textuelle des cas par un diagramme de séquence de toutes les activités des acteurs du système.

4.2.2 Diagramme de séquence

IL permet d'étudier les interactions entre les objets et constitue une partie dynamique du système d'information. Il montre d'une façon séquentielle les envois des messages qui interviennent entre les objets, il peut également montrer les flux des données [26], [28].

Ainsi, pour une meilleure visibilité des interactions effectuées au sein du système, nous avons divisé notre diagramme de séquence en fonction des modules représentés par les cas d'utilisation.

❖ *Diagramme de séquence du cas « Ouverture compte et Authentification »*

Le diagramme de séquence qu'illustre la figure 16 suivante décrit le cas d'utilisation « Ouverture compte et Authentification »

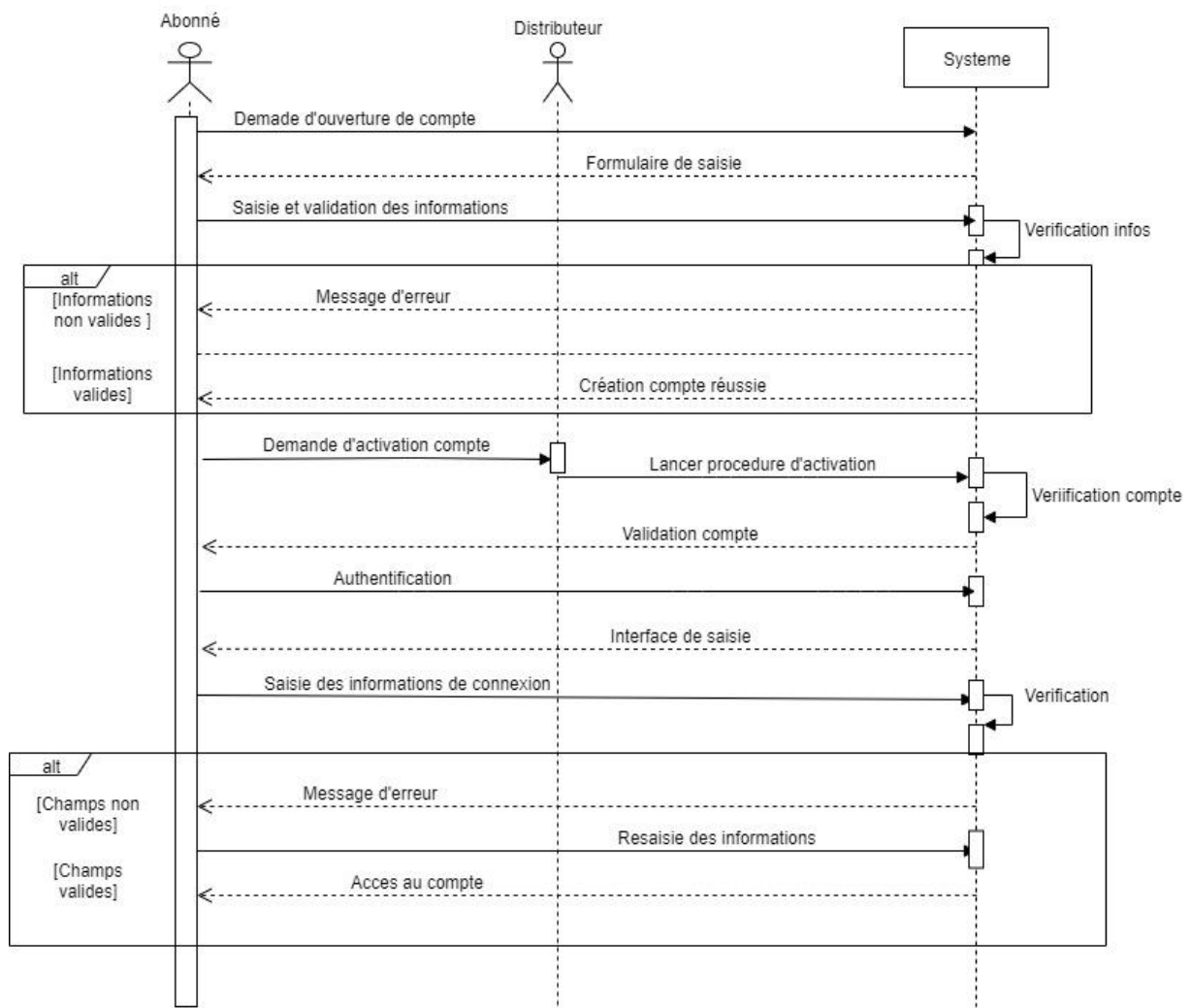


Figure 16 : Diagramme de séquence "Ouverture compte et Authentification"

❖ **Description du cas d'utilisation « Créer un compte et s'authentifier »**

Les diagrammes réalisés jusqu'à maintenant nous ont permis de découvrir petit à petit les fonctionnalités (appelés aussi des cas d'utilisation) que l'on devrait avoir dans le futur logiciel.

La description d'un cas d'utilisation permet de :

- Clarifier le déroulement de la fonctionnalité ;
- Décrire la chronologie des actions qui devront être réalisées ;
- Identifier les parties redondantes pour en déduire des cas d'utilisation plus précises qui seront utilisées par inclusion, extension ou généralisation/spécialisation. Et oui, dans ce cas nous réaliserons des itérations sur les diagrammes de cas d'utilisation ;
- Indiquer d'éventuelles contraintes déjà connues et dont les développeurs vont devoir tenir compte lors de la réalisation du logiciel.

Ainsi ci-dessous, nous commençons par la description des cas d'utilisation « Ouverture compte » et « Authentification » qui sont pratiquement liés.

Cas d'utilisation : Ouverture compte et Authentification	
Titre	Ouverture compte, Authentification
But	Créer un compte pour accéder aux fonctionnalités de l'application
Résumé	L'abonné doit remplir un formulaire d'inscription puis valide son action. Le système effectue une vérification puis une mise à jour de la base de données
Acteurs	Abonné
Descriptions des enchaînements	
Pré condition	Post condition
L'abonné doit accéder au système	Abonné inscrit
Scenario nominal	
<ol style="list-style-type: none"> 1. L'utilisateur demande le formulaire d'inscription en cliquant sur « <i>Créer votre compte</i> » 1. Le système lui envoie le formulaire d'inscription à remplir 2. L'utilisateur remplit le formulaire et appuie sur valider 3. Le système effectue les validations et l'enregistrement dans la base de données 4. L'utilisateur est redirigé vers la page d'authentification et renseigne ses identifiants 5. Le système lui présente son compte et son solde à l'état initial 	
Enchaînement d'échec	
<ol style="list-style-type: none"> 6. Le compte existe déjà ou les données saisies sont incorrectes 7. Il n'y a pas de connexion internet 	

Tableau 5 : Description du cas d'utilisation "Ouverture compte et Authentification "

❖ **Diagramme de séquence du cas « Effectuer transfert »**

Le diagramme de séquence système qu'illustre la figure 17 décrit le cas d'utilisation « Effectuer transfert »

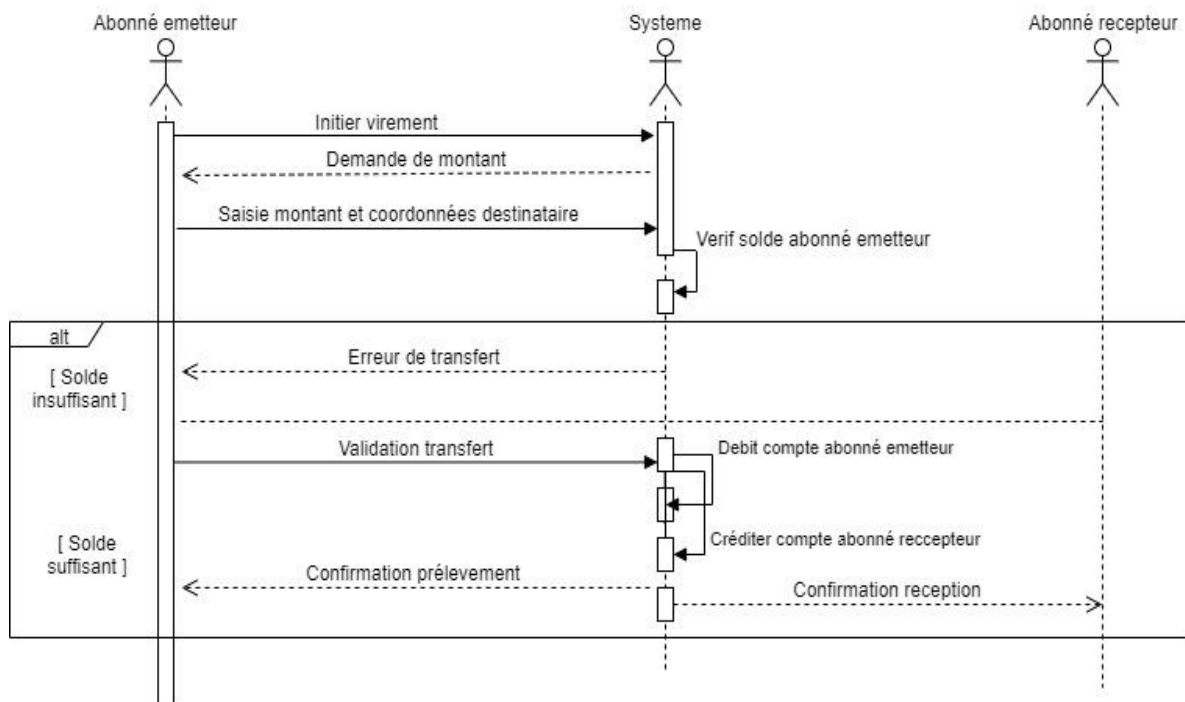


Figure 17 : Diagramme de séquence du cas " Effectuer transfert "

❖ *Description du cas d'utilisation « Effectuer transfert »*

Cas d'utilisation : Effectuer transfert	
Titre	Effectuer transfert
But	Envoyer de l'argent à un utilisateur du système
Résumé	L'utilisateur saisit le numéro du destinataire et le montant de la transaction et valide. Le système vérifie le solde avant de finaliser l'opération.
Acteurs	Abonné
Descriptions des enchainements	
Pré condition	Post condition
<ul style="list-style-type: none"> - L'utilisateur doit se connecter - L'utilisateur doit avoir un montant suffisant 	<ul style="list-style-type: none"> - Le transfert est effectué - Afficher solde courant
Scenario nominal	
<ol style="list-style-type: none"> 1. L'utilisateur accède au formulaire de transfert en cliquant sur « Envoyer de l'argent » 2. Le système lui demande de saisir les informations de transfert 3. L'utilisateur remplit le formulaire appuie sur le bouton « Valider » 4. Le système vérifie de solde de l'abonné émetteur avant de débiter son compte 5. L'utilisateur reçoit une notification de confirmation du transfert 6. Le système lui affiche la page du compte avec le nouveau solde 	

Enchaînement d'échec
7. Le solde est insuffisant
8. La saisie de données n'est pas correcte
9. La connexion n'est pas bonne pour effectuer et suivre un transfert

Tableau 6 : Description textuelle du cas d'utilisation " Effectuer transfert "

❖ *Diagramme de séquence du « Recharger compte »*

Le diagramme de séquence qu'illustre la figure 18 décrit le cas d'utilisation « Recharger compte ». C'est une représentation graphique des différents échanges entre l'utilisateur et le système lors de l'approvisionnement du compte.

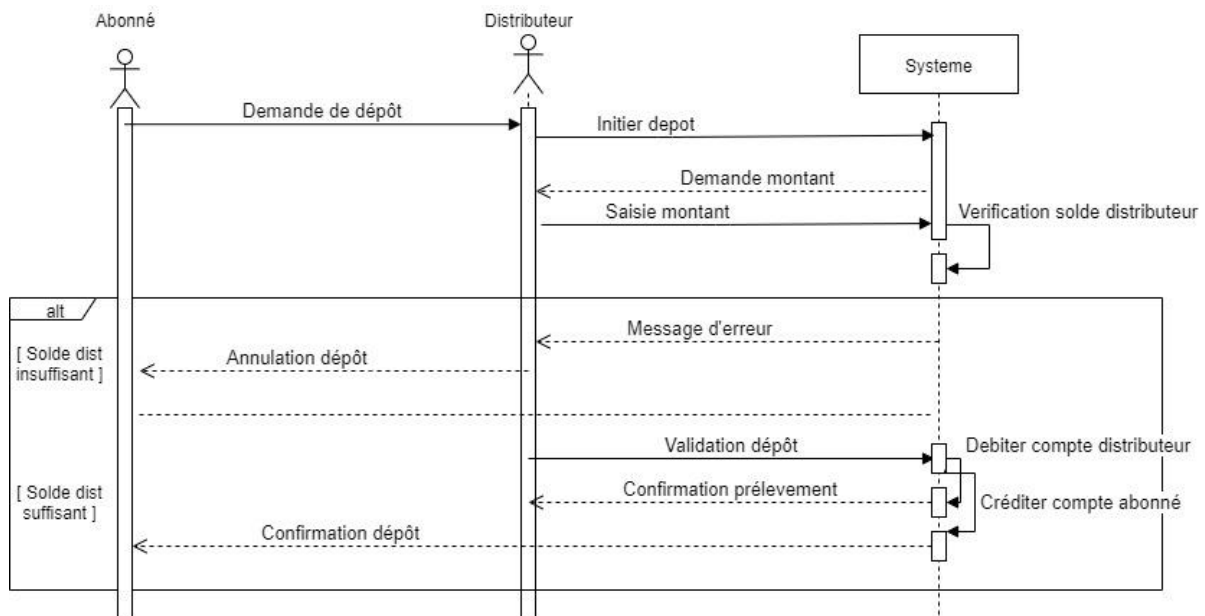


Figure 18 : Diagramme de séquence du cas d'utilisation " Recharger compte "

❖ *Description du cas d'utilisation « Recharger compte »*

Cas d'utilisation : Recharger compte	
Titre	Recharger compte
But	Alimenter son compte afin de pouvoir faire d'éventuelles transactions
Résumé	L'utilisateur doit se rendre chez un distributeur ou un abonné pour recharger son compte en monnaie électronique.
Acteurs	Abonné, Distributeur
Descriptions des enchaînements	
Pré condition	Post condition

- L'utilisateur doit se connecter	- Afficher solde courant
- Le compte est insuffisant pour faire un transfert	- Le compte est rechargé
Scenario nominal	
1. L'utilisateur fait une demande de dépôt dans son compte électronique auprès d'un distributeur ou un autre abonné	
2. Le distributeur demande le formulaire de recharge en cliquant sur « Recharger compte »	
3. Le système lui affiche le formulaire de recharge à remplir	
4. Le distributeur remplit les données du destinataire et valide	
5. Le système vérifie le solde du distributeur avant de débiter son compte	
6. L'utilisateur reçoit une notification de recharge de son compte avec le montant	
Enchaînement d'échec	
7. Le solde maximal autorisé à recharger est atteint	
8. Le distributeur n'a pas suffisamment de solde permettant d'effectuer une transaction	

Tableau 7 : Description textuelle du cas d'utilisation " Recharger compte "

❖ *Diagramme de séquence du cas d'utilisation « Effectuer retrait »*

Le diagramme de séquence système qu'illustre la figure 19 décrit le cas d'utilisation 'Effectuer retrait ». Il représente les différents enchaînements à suivre lors d'une opération de retrait d'argent depuis son compte chez un distributeur.

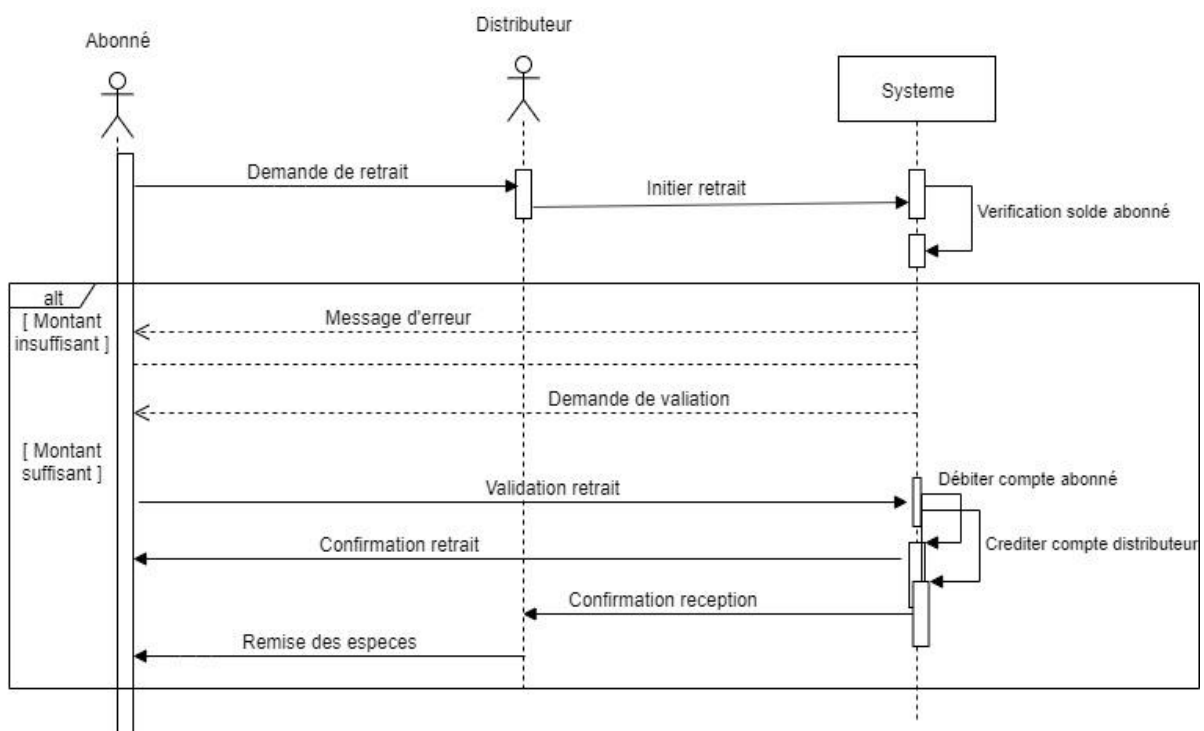


Figure 19 : Diagramme de séquence du cas d'utilisation " Effectuer retrait "

❖ *Description textuelle du cas d'utilisation « Effectuer retrait »*

Cas d'utilisation : Effectuer retrait	
Titre	Effectuer retrait
But	Faire un retrait d'argent depuis son compte
Résumé	L'utilisateur doit avoir un montant suffisant dans son compte et effectuer un retrait. Il peut le faire chez un distributeur ou un autre utilisateur.
Acteurs	Abonné, Distributeur
Description des enchainements	
Pré condition	Post condition
- L'utilisateur s'authentifier, - Le compte est suffisant pour faire un retrait	Le retrait est effectué
Scenario nominal	
1. L'utilisateur fait une demande de retrait chez un distributeur ou un autre utilisateur simple 2. Le distributeur ouvre le formulaire de retrait d'argent, le rempli puis valide 3. Le système vérifie le solde de l'abonné avant de débiter son compte 5. Le système débite (solde suffisant) le compte de l'abonné et crédite le celui du distributeur 6. L'utilisateur reçoit une notification que le retrait est effectué 7. Le système lui affiche son compte avec le nouveau solde	
Enchainement d'échec	
8. Le solde est insuffisant pour permettre un retrait 9. Absence ou faiblesse de connexion à internet 10. Erreur de saisie des informations de retrait	

Tableau 8 : Description textuelle du cas d'utilisation " Effectuer retrait "

❖ *Diagramme de séquence du cas d'utilisation « Convertir monnaie »*

Le diagramme de séquence système de la figure 20 décrit le cas d'utilisation « Convertir monnaie ». Il fait intervenir le distributeur et un prestataire (banque).

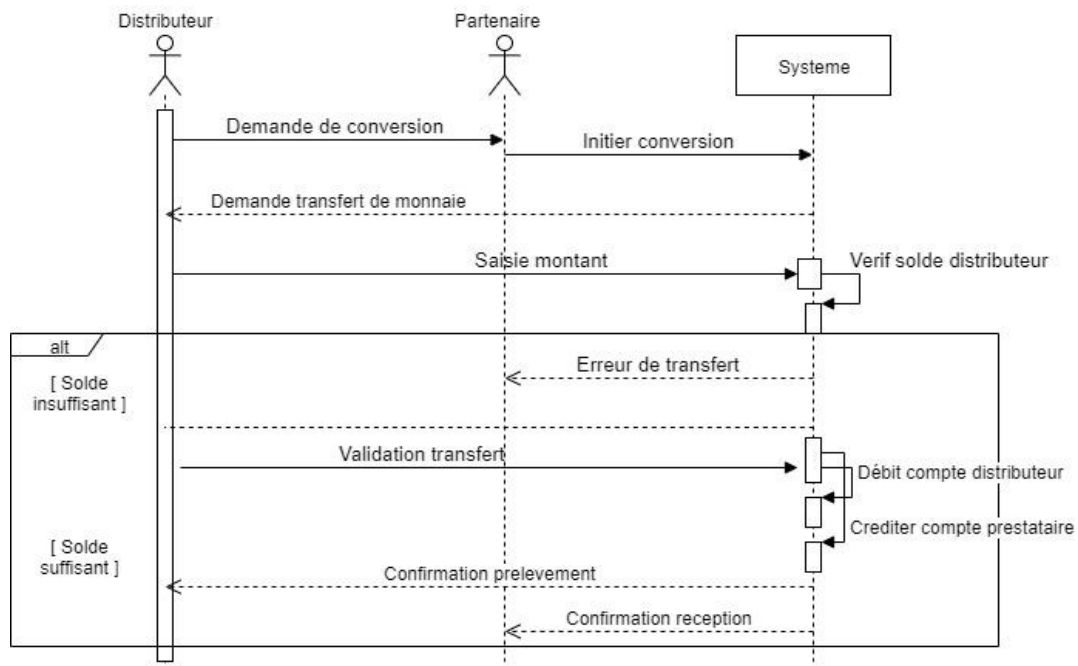


Figure 20 : Diagramme de séquence du cas d'utilisation " Convertir monnaie "

❖ *Description textuelle du cas d'utilisation « Convertir monnaie »*

Cas d'utilisation : Convertir monnaie	
Titre	Convertir monnaie
But	Echanger la monnaie électronique en espèces
Résumé	Le distributeur doit se rendre chez un partenaire (banque) pour convertir la monnaie électronique en espèces afin de pouvoir réaliser des échanges avec les utilisateurs.
Acteurs	Distributeur, prestataire
Descriptions des enchainements	
Pré condition	Post condition
Le distributeur doit avoir un montant suffisant	Conversion effectuée
Scenario nominal	

<ol style="list-style-type: none"> 1. Le distributeur fait une demande de conversion de monnaie auprès du prestataire 2. Le prestataire lui demande de transférer le montant à convertir sur son compte 3. Le distributeur demande le formulaire de transfert d'argent 4. Le système lui envoie le formulaire qui permet d'effectuer un transfert 5. Le distributeur saisit le montant et les coordonnées bancaires du prestataire 6. Le système vérifie le solde distributeur avant de débiter son compte 7. Le distributeur et le prestataire reçoivent une notification que le transfert est effectué 8. Le distributeur est redirigé vers la page de son compte avec son nouveau solde 9. Le prestataire lui remet de l'argent en espèces
Enchaînement d'échec
<ol style="list-style-type: none"> 10. La connexion n'est pas bonne 11. Erreur de saisie des données pour la conversion

Tableau 9 : Description textuelle du cas d'utilisation " Convertir monnaie"

4.2.3 Diagramme d'activité

4.2.3.1 Présentation

Le diagramme d'activités permet de mettre l'accent sur les traitements. Il est donc particulièrement adapté à la modélisation du cheminement de flots de contrôle et de flots de données. Il permet ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation. De plus, leur représentation sous forme d'organigrammes les rend facilement intelligibles et beaucoup plus accessibles que les diagrammes d'états-transitions.

4.2.3.2 Quelques exemples de diagramme d'activité

Diagramme d'activité du cas d'utilisation « Authentification »

Le diagramme d'activité qu'illustre la figure 21 décrit le cas d'utilisation « Authentification ».

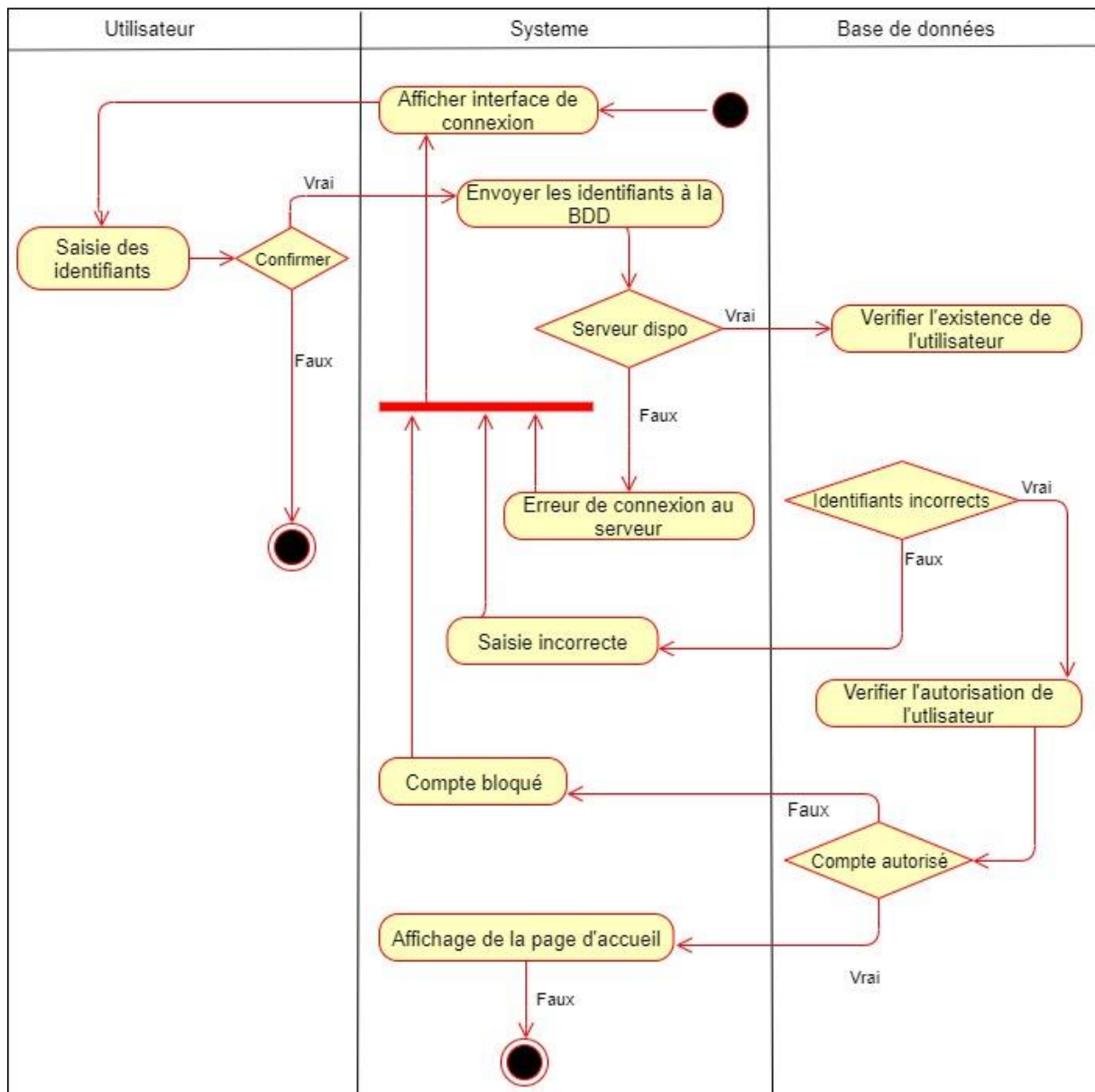


Figure 21 : Diagramme d'activité de l'authentification

✚ Diagramme d'activité « Effectuer transfert »

Le diagramme d'activité qu'il illustre la figure 22 décrit les différentes actions ou enchainements effectués lors d'une opération de transfert d'argent.

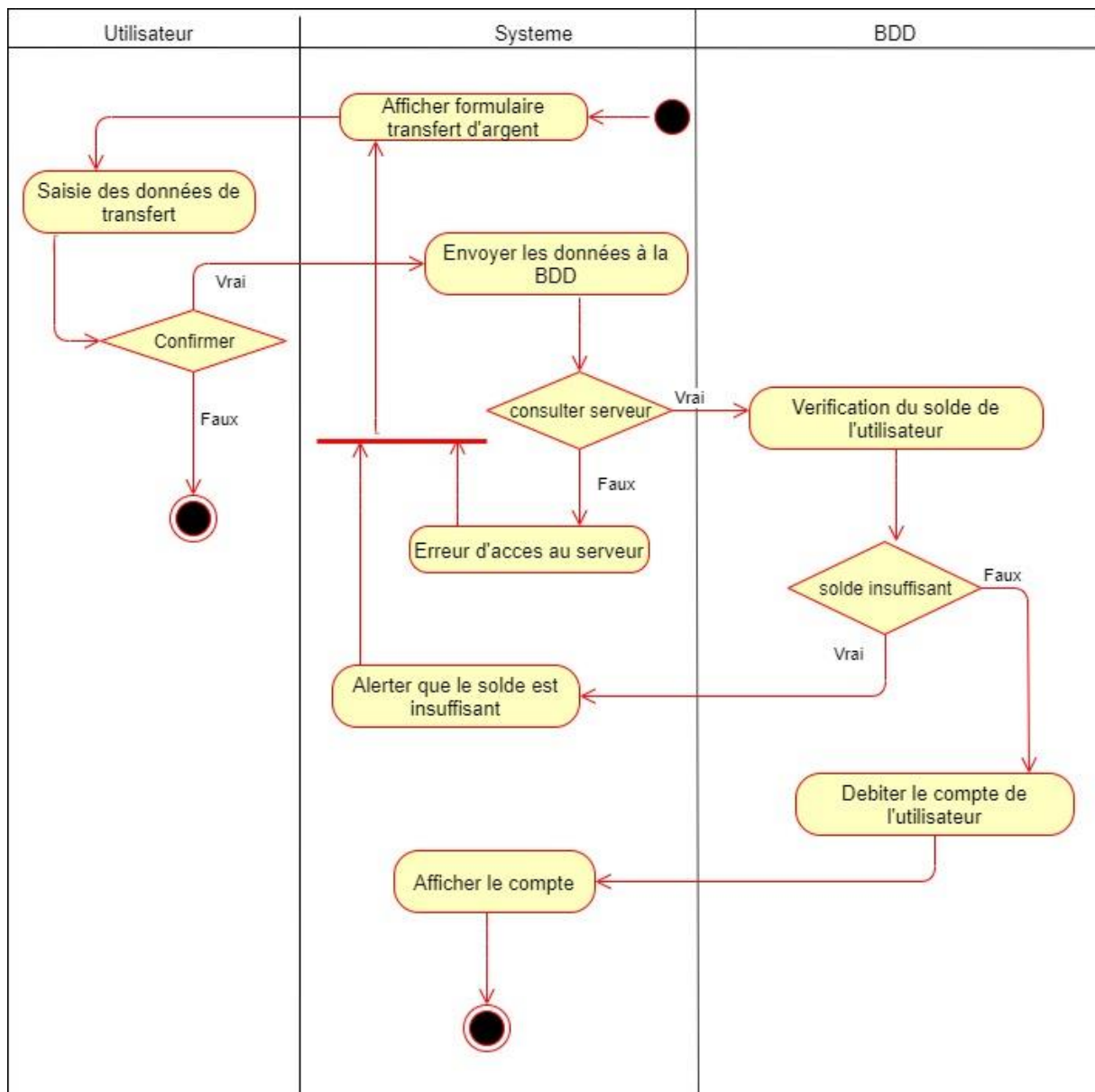


Figure 22 : Diagramme d'activité de transfert d'argent

4.2.4 Conception de la base de données

4.2.4.1 Diagramme de classe

La principale question concernant la mise en œuvre de la base de données est : Comment organiser et stocker toutes les informations utilisées dans notre système de transfert d'argent à savoir les informations des abonnés, la gestion de leur compte, la manipulation du temps pour servir les principales fonctionnalités de l'application à savoir les opérations de transaction de monnaie électronique.

L'objectif du système consiste à transférer et à recevoir de la petite monnaie, ainsi le modèle de données doit fournir une structure efficace pour les requêtes d'échange de fonds entre

abonnés et de calcul de solde. La base de données doit contenir des données concernant les utilisateurs du système, des données concernant les différents types d'opérations qui s'effectuent sur les comptes des abonnés ainsi que des données concernant les partenaires externes.

Afin de bien détailler l'architecture de la base de données, nous avons conçu le diagramme de classe représenté dans la figure suivante.

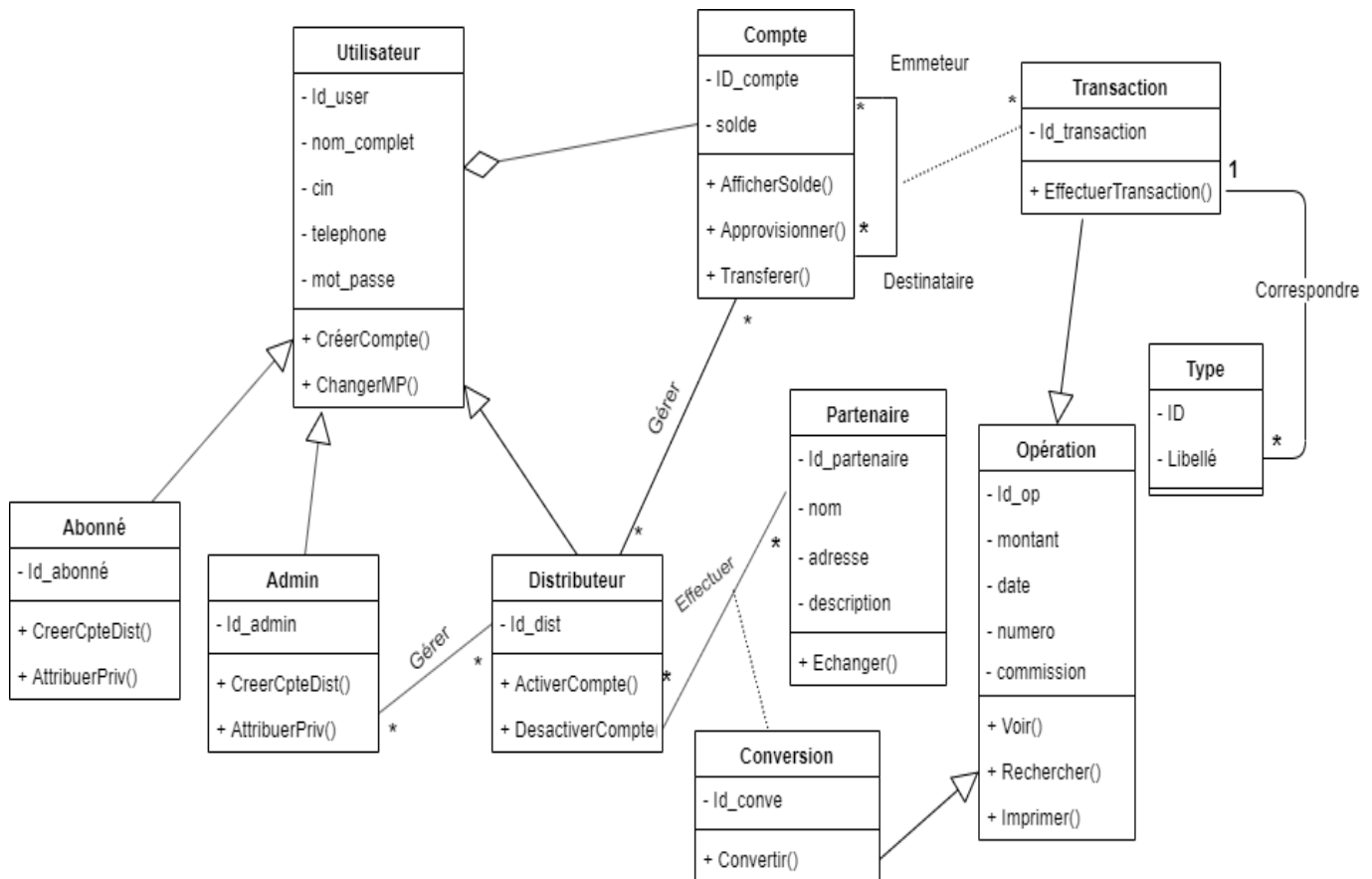


Figure 23 : Diagramme de classe de la solution

Le tableau suivant explique les différentes classes de la base de données.

Classe	Description
Abonné	Informations concernant un utilisateur
Distributeur	Abonné avec certains privilèges
Administrateur	Informations de l'Admin et ses privilèges

Compte	Informations sur les comptes de chaque abonné
Transaction	Détails de chaque transaction effectuée
Type	Informations du type de transaction faite
Opération	Informations d'une opération
Prestataire	Informations concernant le prestataire (partenaire)
Conversion	Détails de la conversion de monnaie en espèces

Tableau 10 : Description de la base de données

4.2.4.2 Passage au modèle relationnel

Le modèle relationnel est le modèle logique de données qui correspond à l'organisation des données dans la base de données relationnelle. Un modèle est composé de relations encore appelées tables. Ces dernières sont décrites par des attributs aux champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses. L'identifiant d'une relation est composé d'un ou de plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée.

4.2.4.2.1 Règles de passage au modèle relationnel

Cette section présente les règles permettant de décrire un schéma logique dans les modèles relationnels et objet-relationnels à partir d'un diagramme de classe UML. Nous donnons ci-après quatre règles (R1 à R4) pour traduire UML à un modèle relationnel équivalent. Il existe d'autres solutions de transformation, mais ces règles que nous présentons ci-dessous sont les plus simples et les plus opérationnelles.

❖ Transformation de classe (R1)

Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter de telle sorte que la relation dispose d'une clé primaire.

❖ Transformations des associations

Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations :

- **Association un à plusieurs (R2)** : Il faut ajouter un attribut du type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de

la relation père de l'association. La clé de la relation père migre dans la relation fils.

Exemple :

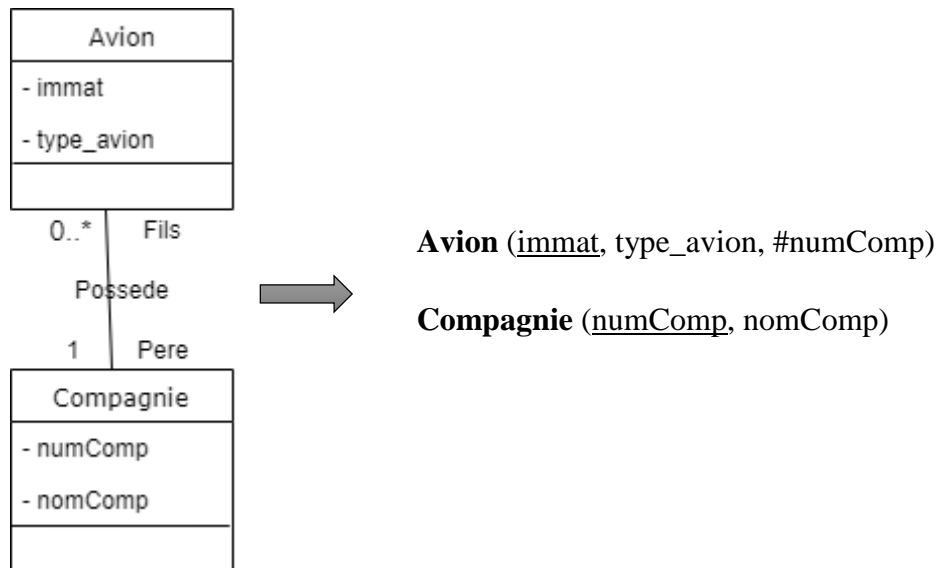


Figure 24: Exemple d'association " Un à plusieurs "

- **Associations plusieurs à plusieurs (R3)** : L'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectées à l'association. Chaque attribut devient clé étrangère et leur combinaison forme la clé primaire. Les attributs de l'association (classe-association) doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

Exemple :

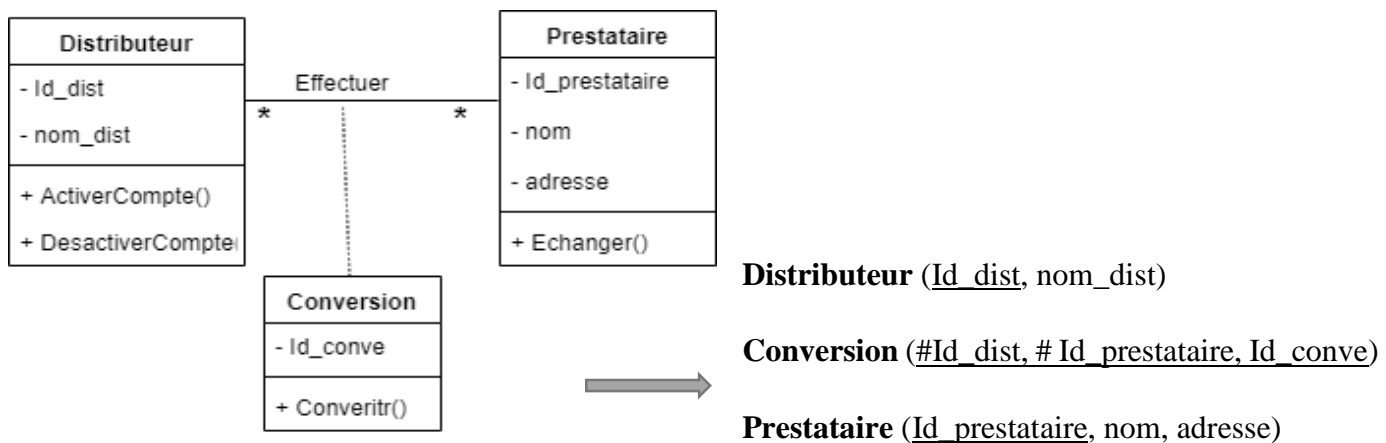


Figure 25 : Exemple d'association " Plusieurs à plusieurs "

- **Association un à un (R4)** : Il faut ajouter un attribut clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro (exemple : 0..1 et 1). Dans le cas d'UML, il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de l'entité (classe) connectée à l'association. Si les deux cardinalités (multiplicités) minimales sont à zéro, le choix est donné entre les deux relations dérivées de la R1. Si les deux cardinalités minimales sont à un, il est sans doute préférable de fusionner les deux classes en une seule relation.

Exemple :

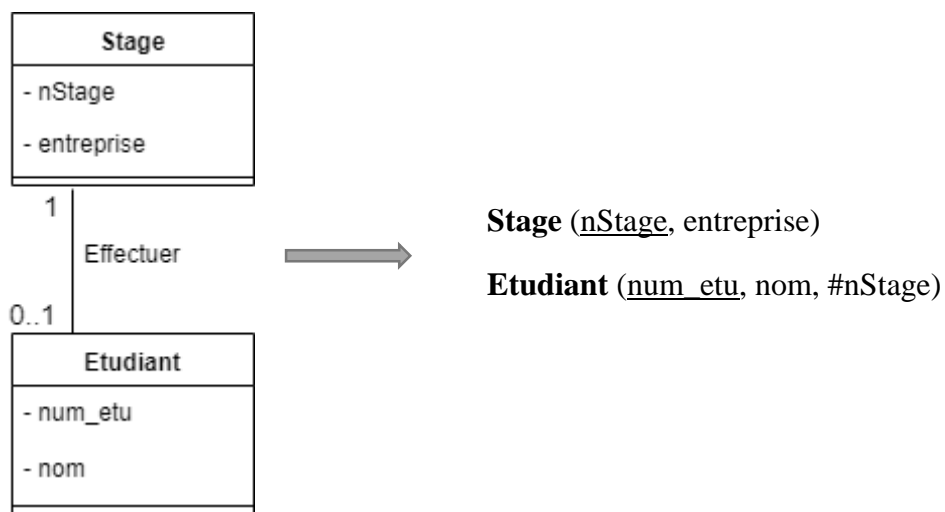


Figure 26 : Exemple d'association " Un à un "

❖ Transformation de l'héritage

Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes dont la décomposition par distinction, la décomposition descendante et celle ascendante décrites ci-dessous :

- **Décomposition par distinction** : Il faut transformer chaque sous-classe en une relation. La clé primaire de la super-classe migre dans les relations issues des sous-classes et devient à la fois clé primaire et clé étrangère.

Exemple : L'association d'héritage est traduite en faisant migrer l'identifiant de la super-classe (Personnel) dans les deux relations déduites des sous-classes. Cet attribut devient aussi clé étrangère.

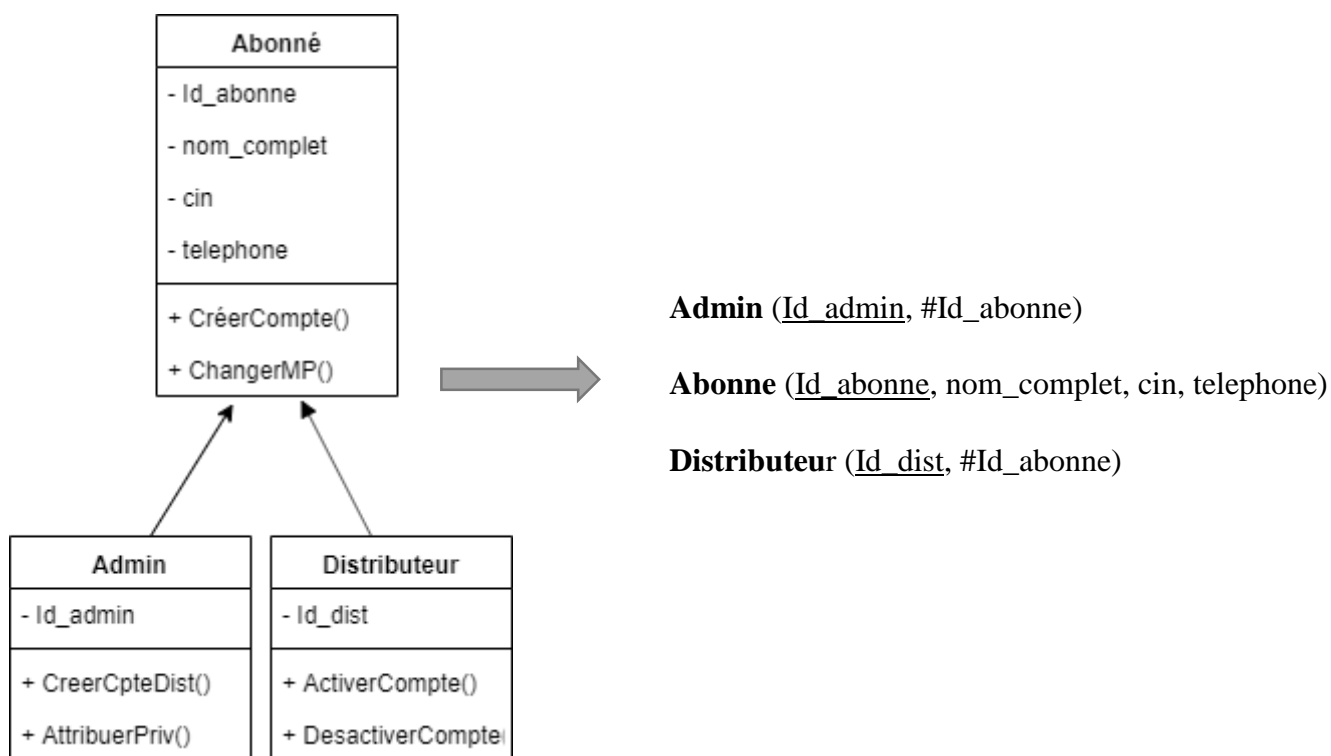


Figure 27 : Décomposition par distinction d’une association d’héritage

- **Décomposition ascendante** : Il faut supprimer les relations issues des sous-classes et faire migrer les attributs dans la relation issue de la super-classe.

Exemple : En appliquant le principe de décomposition ascendante dans la figure ci-dessous, nous obtenons une relation issue de la super-classe dans laquelle se trouve le contenu des relations issues des sous-classes.

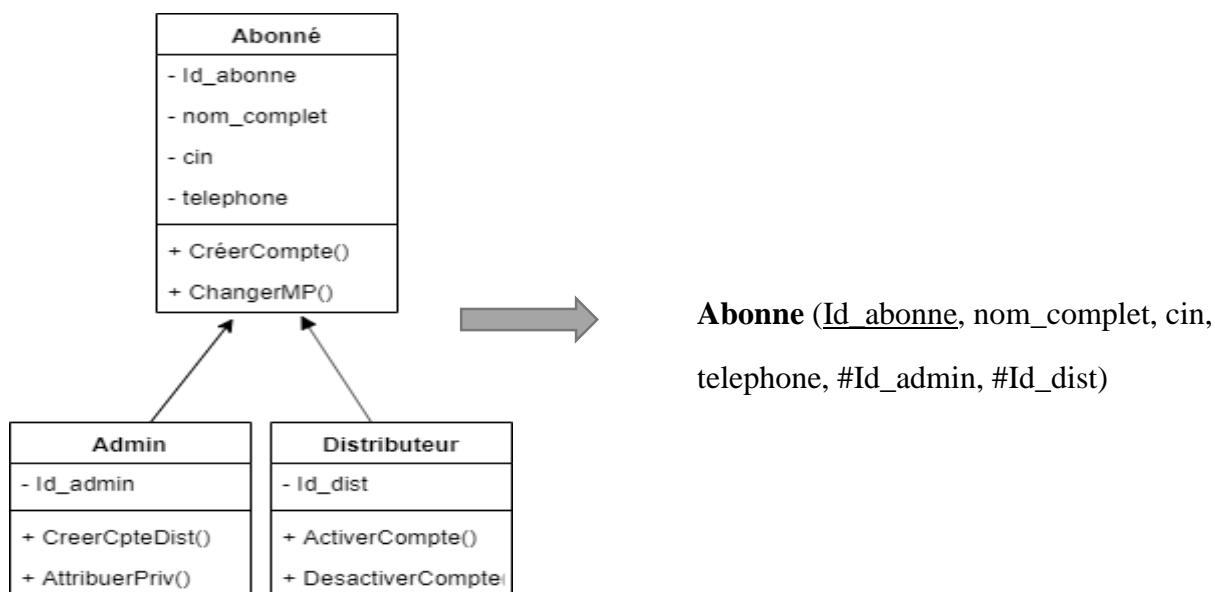


Figure 28 : Décomposition ascendante d’une association d’héritage

- **Décomposition descendante** : S'il existe une contrainte de totalité (toutes les instances d'une classe correspondent au moins à une des instances des classes liées) ou de partition (indique que toutes les instances d'une classe correspondent à une et une seule instance des classes liées) sur l'association, il est possible de ne pas traduire les relations issues de la super-classe. Il faut alors faire migrer tous ses attributs dans les relations issues des sous-classes.

Exemple : La composition du personnel de l'Université Assane SECK est une bonne illustration de la contrainte de partition. Le personnel est composé de PER et de PATS et il n'existe pas un personnel n'étant ni PER ni PATS. La figure suivante représente une décomposition d'héritage descendante.

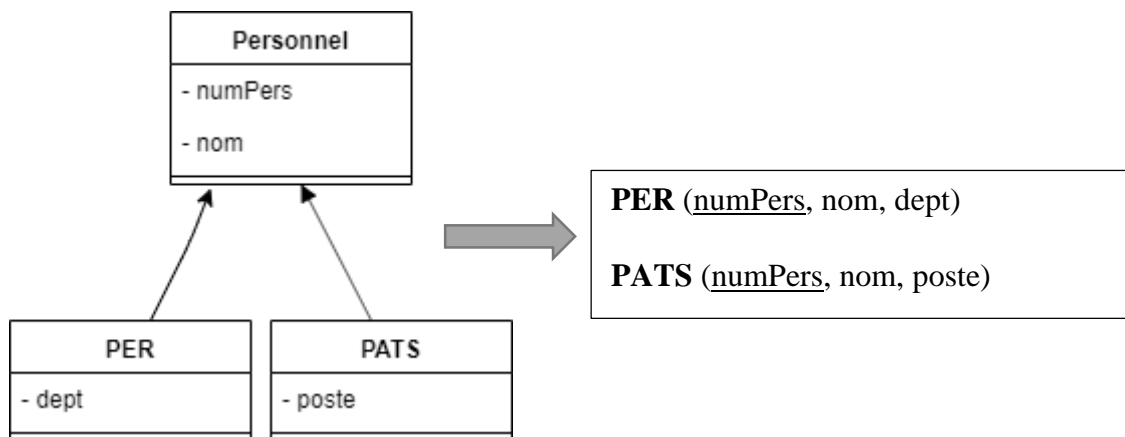


Figure 29 : Décomposition descendante d'une association d'héritage

Ainsi, dans notre cas nous avons choisi la décomposition par distinction qui correspond le plus à nos attentes. L'héritage composée de la super-classe *Abonné*, des sous-classes *Admin* et *Distributeur* doit être transformé en relation issue de chaque sous-classe. En effet, l'administrateur et le distributeur sont tous des abonnés mais jouent des rôles différents dans le fonctionnement du système. Dans ce cas l'utilisation de la décomposition ascendante ou descendante entrainerait une incohérence des données au niveau de l'application.

4.2.4.2.2 Modèle logique de données

- **Abonne** (Id_abonne, nom_complet, cin, téléphone, mot_passe)
- **Admin** (Id_admin, #Id_abonne)
- **Distributeur** (Id_dist, #Id_abonne)
- **Compte** (Id_compte, solde, #Id_abonne)

- **Opération** (Id, montant, date, numéro, commission, #Id_transaction)
- **Type** (Id, #Id_transaction, libellé)
- **Prestataire** (Id_prestataire, nom, adresse, description)
- **Conversion** (Id_conversion, #Id_prestataire, #Id_op)

4.2.5 Architecture globale de la solution

L'architecture informatique définit la structuration d'un système informatique, matériel, et logiciel, en termes de composants et d'organisation de ses fonctions. Les vues structurelles d'une architecture logicielle sont :

- ❖ **La vue logique** qui définit les principaux composants d'une architecture sans se soucier des détails physiques,
- ❖ **La vue physique** qui est une description de l'intégration et de la distribution de la partie logicielle sur la partie matérielle.

4.2.5.1 Vue logique

La vue logique permet de décrire, de décomposer et de représenter un système en différents composants appelés sous-système. Un sous-système est une définition cohérente qui traite une partie du problème défini en termes de services qu'il fournit. C'est un ensemble de classes, d'associations, d'opérations, d'événements, de contraintes ayant des interfaces bien définies et restreintes avec les autres sous-systèmes. La division du système en sous-systèmes cohérents et liés chacun fournissant un service permet de réduire la complexité de conception et de développement de l'application. Le diagramme de composants permet de représenter une vue logique d'une architecture en définissant les principaux composants sans se soucier des détails physiques de déploiement. Notre système est composé par les sous-systèmes suivants :

- ✓ Un sous-système d'identification d'un utilisateur,
- ✓ Un sous-système d'opérations de transactions de monnaie entre utilisateurs,
- ✓ Un sous-système de l'historique des transactions déjà effectuées,
- ✓ Un sous-système consacré au profil des utilisateurs (modification, gestion de mot de passe, etc.) comme le montre la figure suivante :

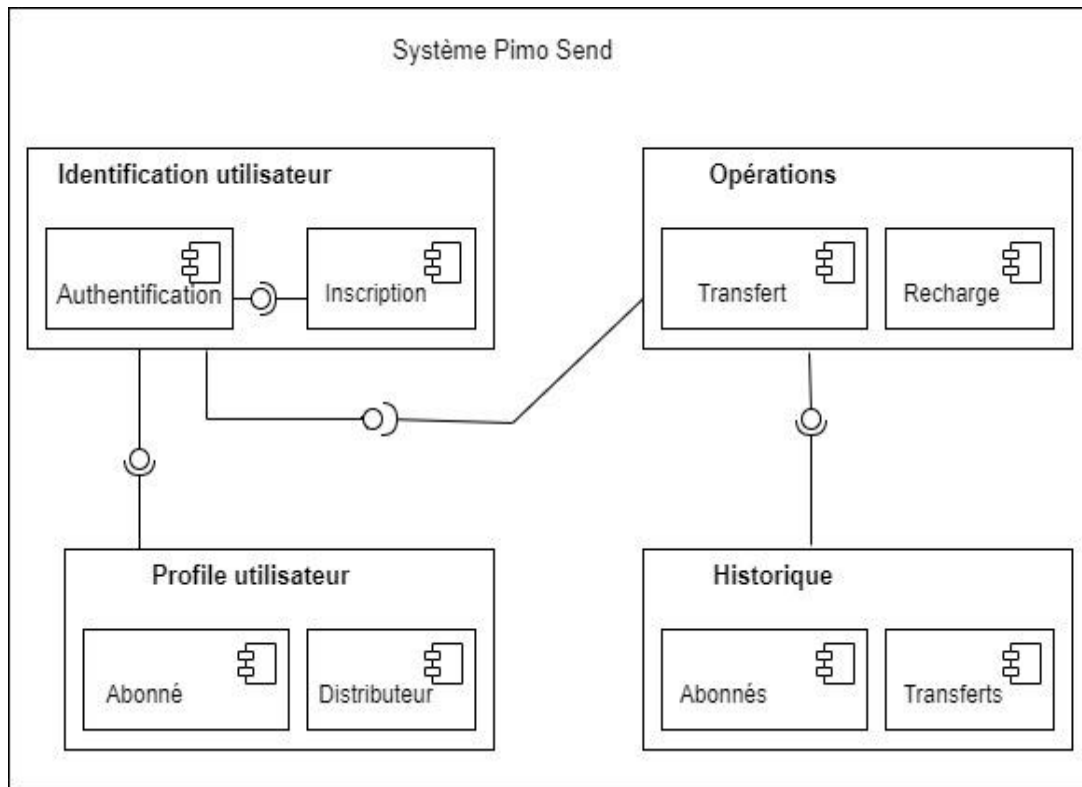


Figure 30 : Diagramme de composants du système

4.2.5.2 Vue de déploiement (Vue physique)

Le modèle de configuration matérielle a pour but d'exprimer les contraintes de mise en œuvre au niveau physique. On y trouve les nœuds de connexions physiques du système, qui sont les différents types de machines connectées par des moyens divers. Les modèles de déploiement et de configuration matérielle s'expriment tous deux à l'aide d'un diagramme de déploiement. Un système de transfert d'argent exige de supporter les déplacements de ses utilisateurs en temps réel. La figure 33 représentant le diagramme de déploiement de notre système décrit la répartition physique des fonctions-métiers du système. On a décidé d'utiliser un serveur en ligne accessible partout une fois connecté à internet.

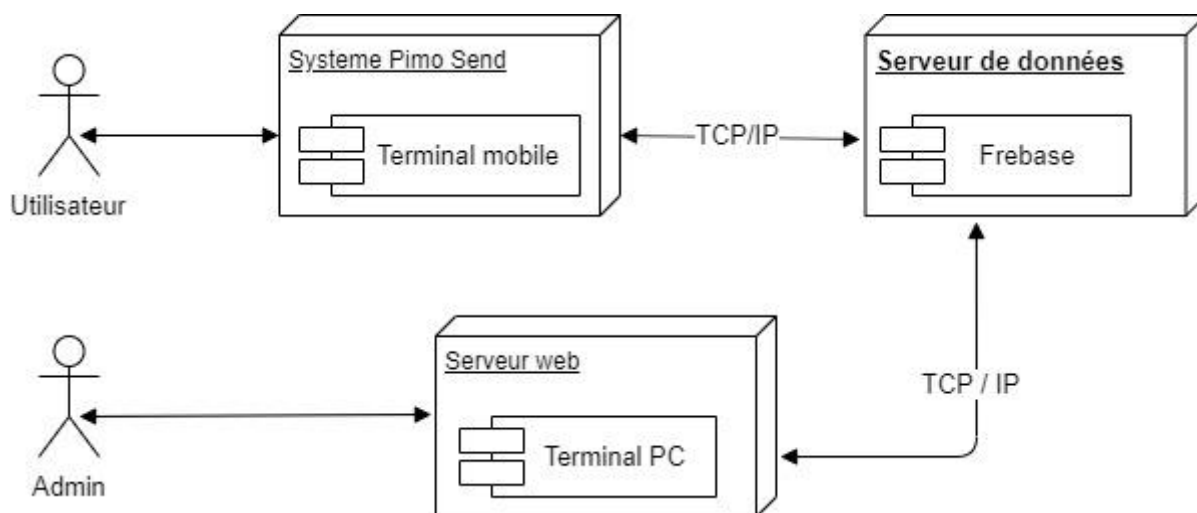


Figure 31 : Diagramme de déploiement du système

4.2.5.2.1 Le serveur d'application

Pour le serveur d'application, on a utilisé **Node.js** qui est une plateforme de logiciel libre en JavaScript orientée réseau avec une architecture en couches, il est utilisé notamment comme serveur web. Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur, ce qui nous permet donc de faire du JavaScript en dehors du navigateur [36].

4.2.5.2.2 Architecture par couche

Parmi les différentes façons de structurer une architecture, nous avons opté l'approche par couche qui résume et correspond parfaitement à la solution envisagée. Une couche (layer en anglais) est une division logique horizontale d'un système qui fournit une abstraction particulière du système aux couches supérieures. Chaque couche possède des rôles spécifiques. Dans une structuration par couches, les couches inférieures prennent en charge des rôles qui offrent un fonctionnement de base pour les couches supérieures, permettant par la suite d'abstraire l'implémentation de ces services basiques. Une telle architecture permet également d'obtenir un bon niveau de réutilisation, à travers l'utilisation des solutions aux problèmes ayant un fonctionnement similaire. Voici les différentes couches utilisées pour la réalisation de notre solution :

- **La couche de données** : Cette couche est liée au serveur de base de données (SGBD). Elle est responsable du stockage physique et l'accès des données. Le système de stockage des données a pour but de conserver une quantité plus ou moins importante de données de façon structurée. Nous pouvons utiliser pour cette partie des systèmes très

variés qui peuvent être des systèmes de fichiers, des mainframes, des systèmes de bases de données relationnelles des systèmes de base de données objets sur lequel se base notre système[36], [37].

- **La logique applicative** : Elle se compose généralement d'un script ou d'un programme qui constitue les traitements métiers nécessaires sur l'information afin de le rendre exploitable par chaque utilisateur. C'est ici que se trouvent les fonctionnalités de base qui permettent de créer, de rechercher, de modifier et de supprimer les entités métier dans le respect des propriétés transactionnelles [36], [37].
- **La couche présentation (Affichage)** : Elle est associée à l'utilisateur et n'assume aucune fonction de traitement. C'est la partie la plus immédiatement visible pour l'utilisateur, d'où son importance primordiale pour rendre l'information lisible, compréhensible et accessible [36], [37].

La figure suivante représente l'architecture de notre serveur d'application Node.js.

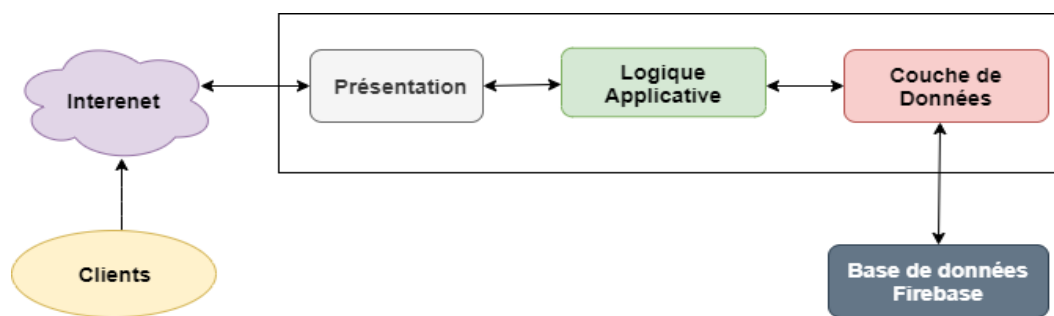


Figure 32 : Architecture par couche de Node JS

Conclusion

Nous avons présenté dans cette partie les différentes phases à mener pour aboutir à la réalisation de notre application. Ensuite, à travers celles-ci, nous avons élaboré tous les diagrammes nécessaires lors de la conception d'une application, et une interprétation textuelle de ces graphiques a été effectuée dans le but de les rendre plus compréhensible.

Dans cette partie, l'application du processus unifié (UP) a été indispensable pour mieux structurer notre démarche de travail, pour une meilleure visibilité des acteurs, des fonctionnalités et des données à manipulées.

Dans le prochain chapitre, nous passerons à une description de l'état de l'implémentation et les résultats de celles-ci.

Chapitre 5 :

L'implémentation

Introduction

La phase de réalisation est la plus importante étape dans le cycle de vie de notre application car à la fin de cette phase nous allons avoir le fruit de toutes les étapes précédentes. A ce stade les cas d'utilisations sont terminés, le problème a été analysé en profondeur ; nous avons défini une conception mieux appropriées aux besoins de l'application.

Maintenant, nous pouvons alors entreprendre la dernière phase du processus unifié qui est d'ailleurs composé de deux parties (implémentation et test) ayant comme objectif d'aboutir à un produit final, exploitable par les utilisateurs.

Dans cette partie, nous allons présenter l'environnement de développement, les outils et langages de programmation, les framework utilisés, l'architecture matérielle mise en place, implémenter tous les cas d'utilisations, et enfin les tester.

5.1 Environnement de travail

5.1.1 Environnement matériel

Pour le développement de cette application, nous utilisons un ordinateur portable de marque ACER avec les caractéristiques suivantes :

- ✦ Système d'exploitation : Windows 10, 64 bits,
- ✦ Processeur : Intel (R) Core i5-4200U CPU @ 1.60GHz 2.30GHz,
- ✦ Mémoire RAM : 6 Giga Octet,
- ✦ Pour le test : un smartphone Samsung avec une version d'Android 8.0.0

5.1.2 Environnement logiciel

5.1.2.1 Visual Studio Code (VS Code)¹⁶

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Ses fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, la refactorisation du code et **Git** intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

¹⁶ : <https://code.visualstudio.com/docs>

Le code source de Visual Studio Code provient du projet logiciel libre et open source VS Code de Microsoft publié sous la licence MIT permissive, mais les binaires compilés sont des logiciels gratuits pour toute utilisation.

5.1.2.2 *Git*¹⁷

Git est un système de contrôle de version distribué gratuit et Open Source conçu pour gérer des petits projets aux très grands projets avec rapidité et efficacité. 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 (GPLv2)¹⁸. Depuis 2016, il s'agit du logiciel de gestion de versions le plus populaire. Git possède deux structures de données : une base d'objets et un cache de répertoires. Il existe quatre types d'objets :

- l'objet **blob** (pour binary large object désignant un ensemble de données brutes) qui représente le contenu d'un fichier ;
- l'objet **tree** (mot anglais signifiant arbre) qui décrit une arborescence de fichiers. Il est constitué d'une liste d'objets de type blobs et des informations qui leur sont associées telles que le nom du fichier et les permissions. Il peut contenir récursivement d'autres objets trees pour représenter les sous-répertoires ;
- l'objet **commit** (résultat de l'opération du même nom signifiant « valider une transaction ») qui correspond à une arborescence de fichiers (tree) enrichie de métadonnées comme un message de description, le nom de l'auteur, etc. Il pointe également vers un ou plusieurs objets commit parents pour former un graphe d'historiques ;
- l'objet **tag** (étiquette) qui est une manière de nommer arbitrairement un commit spécifique pour l'identifier plus facilement. Il est en général utilisé pour marquer certains commits.

5.1.2.3 *Le SGBD Firebase*

Firebase est une plateforme d'hébergement pour n'importe quel type d'application (mobiles, Web, etc.) qui fournit aux développeurs une pléthore d'outils et de services pour les aider à développer des applications de haute qualité, à élargir leur base d'utilisateurs et à générer davantage de profits. Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github) et des

¹⁷ : <https://git-scm.com/s>

¹⁸ : <https://edu.casio.com/support/fr/gplv2.html>

notifications, ou encore des services tels que par exemple un serveur de communication en temps réel [38].

Lancé en 2011 sous le nom d'Envolv, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet (maison-mère de groupe qui regroupe une majorité des activités du groupe comme YouTube, Android, la recherche et les publicités). L'objectif premier de Firebase est de vous libérer de la complexité de création et de la maintenance d'une architecture serveur, tout en vous garantissant une scalabilité (capacité de l'application à s'adapter à un changement d'ordre de grandeur de la demande, en particulier sa capacité à maintenir ses fonctionnalités et ses performances en cas de forte demande) à toute épreuve (plusieurs milliards d'utilisateurs) et une simplicité dans l'utilisation. Pour cela, Firebase a été décomposée en plusieurs produits extrêmement riches et adaptés au monde du mobile, dont la liste suivante :

5.1.2.3.1 Firebase Realtime Database :

Firebase Realtime Database est une base de données NoSQL hébergée dans le cloud qui vous permet de stocker et de synchroniser des données entre vos utilisateurs en temps réel [38], [39].

Realtime Database est vraiment juste un gros objet JSON que les développeurs peuvent gérer en temps réel.

La synchronisation en temps réel permet à vos utilisateurs d'accéder facilement à leurs données depuis n'importe quel appareil, que ce soit sur le Web ou sur un appareil mobile. La base de données en temps réel permet également à vos utilisateurs de collaborer les uns avec les autres.

Un autre gros avantage de Realtime Database est qu'elle est livrée avec des SDK mobiles et Web, vous permettant de créer vos applications sans avoir besoin de serveurs. Lorsque vos utilisateurs sont hors ligne, les SDK de base de données en temps réel utilisent le cache local sur l'appareil pour servir et stocker les modifications. Lorsque l'appareil est en ligne, les données locales sont automatiquement synchronisées.

5.1.2.3.2 Firebase Authentication

Firebase Authentication fournit des services backend, des SDK faciles à utiliser et des bibliothèques d'interfaces utilisateur prêtes à l'emploi pour authentifier les utilisateurs de votre application [38], [39].

Vous pouvez authentifier les utilisateurs de votre application à l'aide des méthodes suivantes :

- ✓ Email & mot de passe
- ✓ Numéro de téléphone
- ✓ Compte Google
- ✓ Compte Facebook
- ✓ Compte Twitter
- ✓ Etc.

L'utilisation de Firebase Authentication facilite la création de systèmes d'authentification sécurisés, tout en améliorant l'expérience de connexion et d'intégration pour les utilisateurs finaux.

5.1.2.3.3 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) fournit une connexion fiable et à faible consommation de batterie entre votre serveur et vos périphériques, vous permettant d'envoyer et de recevoir gratuitement des messages et des notifications sur iOS, Android et sur le Web [38], [39]. Vous pouvez envoyer des messages de notification (limite de 2 Ko) et des messages de données (limite de 4 Ko).

En utilisant FCM, vous pouvez facilement cibler les messages en utilisant des segments prédéfinis ou créer les vôtres, en utilisant les données démographiques et comportementales. Vous pouvez envoyer des messages à un groupe d'appareils abonnés à des rubriques spécifiques, ou vous pouvez obtenir des informations aussi détaillées qu'un seul appareil.

FCM peut envoyer des messages instantanément, ou à un moment ultérieur dans le fuseau horaire local de l'utilisateur. Vous pouvez envoyer des données d'application personnalisées, telles que la définition des priorités, des sons et des dates d'expiration, ainsi que le suivi des événements de conversion.

5.1.2.3.4 Cloud Firestore

Cloud Firestore est une base de données de documents NoSQL qui vous permet de facilement stocker, synchroniser et interroger des données pour vos applications mobiles et Web à l'échelle mondiale [38], [39]. Bien que cela puisse ressembler à quelque chose de similaire à la base de données en temps réel, Firestore apporte beaucoup de nouvelles choses à la plateforme qui est en fait quelque chose de complètement différent de Realtime Database.

Là où Realtime Database stocke des données sous la forme d'un arbre JSON géant, Cloud Firestore adopte une approche beaucoup plus structurée. Firestore conserve ses données dans

des objets appelés documents. Ces documents sont constitués de paires clé-valeur et peuvent contenir n'importe quel type de données, depuis les chaînes jusqu'aux données binaires en passant par des objets qui ressemblent à des arbres JSON (Firestore l'appelle des maps). Les documents, à leur tour, sont regroupés en collections.

La base de données Firestore peut se composer de plusieurs collections qui peuvent contenir des documents pointant vers des sous-collections. Ces sous-collections peuvent à nouveau contenir des documents qui pointent vers d'autres sous-collections, et ainsi de suite.

Vous pouvez créer des hiérarchies pour stocker les données associées et récupérer facilement les données dont vous avez besoin à l'aide de requêtes. Toutes les requêtes peuvent évoluer en fonction de la taille de votre jeu de résultats. Votre application est donc prête à évoluer depuis le premier jour.

5.1.2.4 draw.io

draw.io est l'un des outils de création de diagrammes les plus riches en fonctionnalités du marché et un concurrent direct de solutions professionnelles de bureau comme Visio. Ses nombreuses barres d'outils, réunissant des centaines d'éléments visuels et de fonctions, permettent de créer des diagrammes pour toutes les applications imaginables. Positionnement libre des formes et objets sur la grille, déplacement, redimensionnement et modification de ceux-ci en fonction des besoins : vous y retrouverez les mêmes fonctionnalités que sur n'importe quel outil avancé de création de diagrammes, avec d'excellentes performances, même sur les ordinateurs et tablettes anciens ou peu puissants.

À tout moment, les utilisateurs peuvent personnaliser et organiser leur interface pour travailler le plus efficacement possible. Ils ont la possibilité d'afficher ou de masquer les barres d'outils à tout moment, en un seul clic. Les utilisateurs avancés tireront parti de l'ensemble des fonctionnalités, tandis que les utilisateurs occasionnels réduiront l'interface pour n'afficher que les fonctions de base.

L'interface est simple et facile d'utilisation, les objets sont classés à gauche de l'écran par thèmes, un moteur de recherche vous permet d'interroger la base de données de dessins. Une fois votre travail terminé, la sauvegarde est réalisée au format XML, l'exportation propose différents formats (png, jpg, gif, svg, html, intégration en pages web...). Il est possible d'importer un diagramme au format XML.

A noter que draw.io peut enregistrer directement votre travail sur votre compte Dropbox ou Google Drive (autorisation de connexion nécessaire).

5.2 Choix des outils de travail

5.2.1 Langages utilisés

5.2.1.1 JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation par exemple de Node.js. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet de créer des objets héritiers personnalisés [40]. Le langage a été créé en mai 1995 pour le compte de la Netscape Communications Corporation¹⁹ par Brendan Eich, qui s'est inspiré de nombreux langages, notamment de Java mais en simplifiant la syntaxe pour les débutants. A travers l'évolution du langage, beaucoup de frameworks web et mobile ont été développés afin de soutenir et de faciliter son utilisation et sa prise en main par les développeurs.

5.2.1.2 JSX²⁰

Créé par Facebook, le JSX est un langage qui ressemble à HTML (avec son système de balise) mais ne représente pas les mêmes objets. Chaque balise est une encapsulation d'un élément natif du téléphone. Le JSX est une syntaxe très proche du JavaScript qui est utilisée par la plupart des développeurs pour construire leurs applications web en React (bibliothèque JavaScript libre développée par Facebook depuis 2013) et qui donne accès à pas mal de fonctionnalités utiles [41].

5.2.2 Etude des frameworks existants

5.2.2.1 Définition

En programmation informatique, un framework (appelé aussi infrastructure logicielle, socle d'applications, infrastructure de développement, ou cadre d'applications) désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes

¹⁹ : <https://www.britannica.com/topic/Netscape-Communications-Corp>

²⁰ : <https://fr.reactjs.org/docs/introducing-jsx.html>

lignes de tout ou d'une partie d'un logiciel [41]. Un framework se distingue d'une simple bibliothèque logicielle principalement par :

- ◆ son caractère générique, faiblement spécialisé, contrairement à certaines bibliothèques; un framework peut à ce titre être constitué de plusieurs bibliothèques, chacune spécialisée dans un domaine. Il peut néanmoins être spécialisé, sur un langage particulier, une plateforme spécifique, un domaine particulier : communication de données, data mapping, etc. ;
- ◆ le cadre de travail qu'il impose de par sa construction même, guidant l'architecture logicielle voire conduisant le développeur à respecter certains patrons de conception ; les bibliothèques le constituant sont alors organisées selon le même paradigme.

Les frameworks sont donc conçus et utilisés pour modeler l'architecture des logiciels applicatifs, des applications web, des middlewares et des composants logiciels. Un framework est conçu en vue d'aider les programmeurs dans leur travail. L'organisation du framework vise la productivité maximale du programmeur qui va l'utiliser ; gage de baisse des coûts de construction et maintenance du programme [41].

Il existe un certain nombre d'outils permettant de créer des applications mobiles. La plupart sont « cross-platform », c'est-à-dire qu'ils offrent la possibilité de créer un seul projet qui peut ensuite être utilisé sur plusieurs systèmes d'exploitation mobiles (IOS et Android, le plus souvent). Pour permettre le système « cross-platform », ces frameworks n'utilisent pas le langage natif du téléphone (Java pour Android et Objective-C pour IOS) car il serait alors nécessaire de créer un projet pour chaque langage de programmation et donc pour chaque système d'exploitation mobile. Ils utilisent alors un langage différent des langages natifs (souvent JavaScript pour la partie logique et HTML pour la partie visuel). Ces langages ne sont pas compris par les téléphones mobiles et doivent être exécutés dans un environnement spécifique : le « Runtime Environnement ». Ces environnements encapsulent les projets « cross-platform » dans une « boîte » qui comprend le langage du framework. Cette boîte offre des entrées et des sorties permettant ensuite au framework de communiquer directement avec les composants du téléphone (RAM, CPU, GPU, etc.). L'environnement de runtime peut différer selon les mécanismes utilisés.

De nombreux frameworks mobiles ont été développés pour faciliter la création d'application plus efficacement. Mais seulement quatre sont les plus populaires auprès de la communauté des développeurs, ci-dessous triés par ordre de popularité :

- React (et sa version de développement mobile React Native)
- Angular (et sa version de développement mobile la plus connue Ionic)
- Xamarin
- Cordova

Ces quatre frameworks sont les plus populaires auprès de la communauté des développeurs. Je vais expliquer ci-dessous le fonctionnement de « Ionic », « Xamarin » et « React Native »

5.2.2.2 Ionic

Ionic est un framework utilisant les technologies WEB (Javascript et HTML) permettant la création de sites Web et d'applications mobiles cross-platform. Ce framework est une surcouche du Framework AngularJs, Angular2 ou Angular4 (dépendant de la version d'Ionic) permettant le binding de données (explication détaillée à la section Angular).

L'objectif de ce framework est d'offrir un développement court, efficace et ne nécessitant pas de grandes connaissances dans le domaine. Le développement WEB est la première chose privilégiée que tout informaticien apprend, il est très facile de se lancer dans la création d'une application utilisant le framework Ionic.

Ionic ne peut donc pas être utilisé seul. Il a besoin de fonctionner en corrélation avec Angular pour le binding de données ainsi qu'Apache Cordova pour transformer le projet en un package que le téléphone mobile peut comprendre, appelé APK.

5.2.2.2.1 Architecture

Comme décrit précédemment, Ionic est une surcouche du framework Angular qui, lui-même, utilise le framework Apache Cordova pour la création de l'APK. Le fonctionnement de ces trois frameworks va être expliqué ci-après. La création de l'APK et le test de l'application sur le téléphone mobile sont une étape essentielle du développement. Il existe plusieurs façons de créer un APK. Apache Cordova propose d'encapsuler facilement l'application dans un package lisible par les systèmes d'exploitation mobiles et offre des accès aux composants natifs du téléphone (appareil photo, accéléromètre, etc.) via une collection de plugins (qui font office d'interface) intégrés à Apache Cordova.

La figure ci-dessous montre l'architecture du framework Ionic combiné à Apache Cordova.

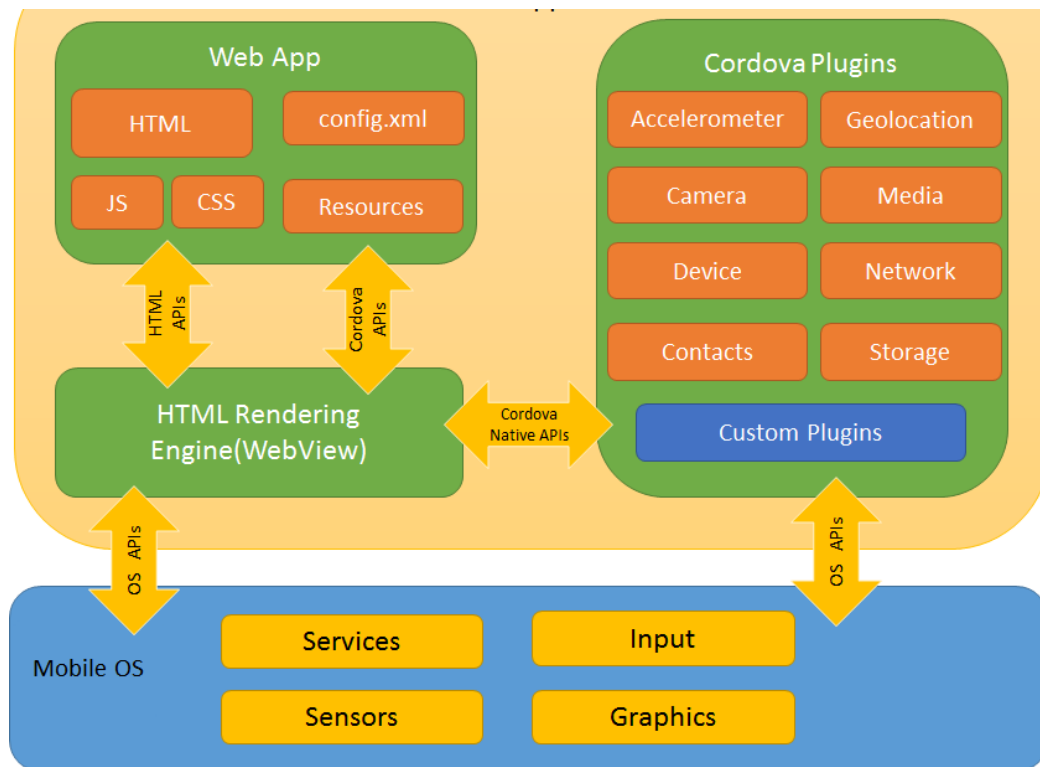


Figure 33 : Architecture d'Ionic et Apache Cordova

L'application, qui peut être créée avec n'importe quel Framework compatible, sera encapsulée dans la « boîte » Web App. Cette dernière discute avec la WebView grâce à des systèmes d'API IN / OUT pour afficher le rendu visuel. De son côté, la WebView à un accès aux composants natifs du téléphone via les plugins d'Apache Cordova. Il est donc possible de récupérer des informations natives et de les traiter. Ces informations seront récupérées / calculées sur les composants natifs, lesquels seront ensuite envoyés aux plugins d'Apache Cordova. Ces derniers vont alors transpiler les données en quelque chose de lisible par l'application et les envoyer à la WebView qui, elle-même, transmettra les informations à la partie logique de l'application.

Apache Cordova utilise l'environnement de runtime natif JavaScript de chaque système d'exploitation pour faire fonctionner l'application et créer les APKs.

5.2.2.2 Avantages

Ionic est un framework récent permettant le développement simple et efficace d'applications cross-platform. C'est dans sa structuration et son fonctionnement simplifiés que réside sa plus grande force.

Comme Ionic utilise le JavaScript ainsi que l'HTML pour la partie visuelle, cela permet à un grand nombre de développeurs d'utiliser ce framework. En effet, les langages WEB sont les

premiers langages que tout développeur apprend car ils sont très simples à comprendre. Il est très facile de trouver de l'aide ainsi que des tutoriels sur internet.

Un autre grand atout d'Ionic est sa portabilité. Il est effectivement possible de créer un site web responsive utilisant les composants donnés par Ionic et ensuite de transformer le site web en une application par quelques modifications seulement.

Ionic a, en effet, une portabilité tellement grande avec les composants offerts qu'il est même possible de créer un seul code pour une application Android, IOS et un site web (pour autant qu'il n'y ait pas de PHP). Seuls quelques pourcents du code doivent être adaptés aux plateformes pour permettre son fonctionnement (cela concerne surtout les plugins qui ne fonctionnent pas de la même manière sur toutes les plateformes).

Un point important d'Ionic est son système de débogage. Comme l'application peut être lancée dans un navigateur, il est possible d'afficher des informations dans la console et de modifier l'interface utilisateur en temps réel.

5.2.2.3 Faiblesses

Actuellement, l'expérience utilisateur est au centre de toutes les préoccupations dans l'informatique. Que ce soit pour des clients lourds, riches, légers, des applications de bureau ou des applications mobiles, il est important que l'utilisateur ait un ressenti positif de l'application.

Un point qu'Ionic ne parvient pas toujours à combler. En effet, comme Ionic est composé de 3 couches (Ionic → Angular → Apache Cordova), les performances de l'application sont grandement diminuées. En cas d'utilisation gourmande de l'application, le nombre d'images par seconde va grandement chuter et l'utilisateur aura l'impression que l'application fonctionne au ralenti et qu'elle n'est pas fluide.

Un autre gros désavantage de ce framework est la compatibilité entre les différents « sous-frameworks » qui le compose. Il est assez fréquent que des modifications soient faites au niveau d'Apache Cordova (surtout au niveau des plugins) et que celles-ci cassent le fonctionnement actuel de notre application.

5.2.2.3 Xamarin

Xamarin est un framework dédié au développement d'applications mobiles « crossplatform » (Android, IOS et pour ceux qui veulent, Windows Phone). Contrairement à la plupart des autres

F

frameworks de développement cross-platform, Xamarin est basé sur le framework .NET et utilise le langage C# et non JavaScript.

Contrairement aux idées reçues, Xamarin n'est pas un framework interprété mais compilé nativement, ce qui lui donne la possibilité de créer des applications très performantes avec un design proche d'un design natif. Ce framework offre la possibilité de créer un seul code source logique codé en C# et transférable sur plusieurs OS cibles. En ce qui concerne la partie visuelle de l'application, il existe deux options :

- Xamarin.Android et Xamarin.iOS permettent de créer une interface native pour chaque OS cible
- Xamarin.Forms permet de créer un seul design « cross-platform » pour Android, IOS et Windows Phone.

La première option est bien évidemment la plus performante, car chaque OS cible aura son propre design créé en langage natif. De plus, créer un design en langage natif nous permet de rester le plus fidèle possible aux normes des interfaces utilisateurs proposées par chaque plateforme. Le désavantage réside dans l'obligation d'écrire plusieurs fois le même code pour chaque OS ciblé.

La deuxième option, Xamarin.Forms, facilite grandement le développement et permet d'avoir un design unifié entre les différentes plateformes mais pose un problème de taille. Les performances sont grandement diminuées pour certaines opérations et la création de designs compliqués sera impossible.

5.2.2.3.1 Architecture

Comme expliqué précédemment, Xamarin offre la possibilité de créer un seul code logique et une ou plusieurs interfaces utilisateurs. Le schéma ci-dessous représente l'architecture complète d'une application Xamarin, incluant l'environnement de runtime, le code partagé et les interfaces utilisateurs.

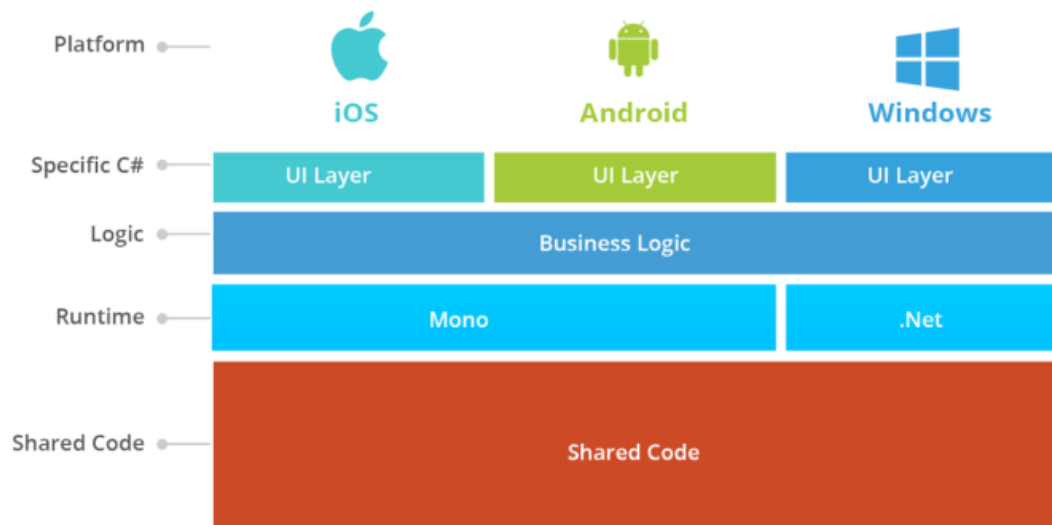


Figure 34: Architecture de Xamarin

Tout comme pour Ionic, les systèmes d'exploitation mobiles ne comprennent pas le langage C# et doivent donc avoir un environnement de runtime spécifique. Étant donné que ces plateformes n'ont pas cet environnement de manière native, Xamarin a pris la décision de l'ajouter dans le code source de l'application. De cette manière, quel que soit le système d'exploitation mobile sur lequel est lancée l'application, elle pourra fonctionner.

5.2.2.3.2 Avantages

Xamarin est un framework proposant une approche totalement différente des autres frameworks de développement mobile. L'utilisation du langage C#, fortement typé, y ajoute une couche de protection contre les éventuels bugs générés par un code approximatif.

Comme expliqué précédemment, les performances d'une application créée grâce à Xamarin est très proche de celles d'une application native. Cela en fait un atout majeur pour ce framework. En effet, il est très important d'avoir une application rapide et fluide car les utilisateurs aiment de moins en moins attendre le chargement d'une application.

La possibilité qu'offre Xamarin de créer l'interface utilisateur de deux manières différentes est aussi un atout pour le framework. Si l'application développée n'utilise pas beaucoup de ressources, il est possible de créer une seule interface utilisateur partagée sur chaque plateforme. Même s'il s'agit d'une application lourde, Xamarin donne aussi la possibilité de créer des interfaces utilisateur natives pour garder des performances proches du natif. Il est aussi possible d'utiliser Xamarin.Forms pour créer des prototypes de l'application et de les montrer aux

clients. Le client donne son avis et lorsque l'interface sera définitive, elle peut être réécrite pour chaque OS cible et ainsi augmenter les performances de l'application.

5.2.2.3.3 Faiblesses

L'utilisation de Xamarin oblige le développeur à se former à deux nouveaux langages : le C# et celui qu'utilise Xamarin pour créer ses interfaces Xamarin.Forms est le XAML (eXtensible Application Markup Language). La communauté utilisant le framework Xamarin est aussi plus faible que celle utilisant les frameworks utilisant les technologies WEB. Il sera donc plus difficile de trouver des réponses aux questions lorsque qu'il y aura un problème.

Comme expliqué précédemment, l'utilisation de Xamarin.Forms réduit grandement les performances d'une application créée avec Xamarin. Il n'est donc pas possible de créer des interfaces utilisateurs compliquées ou contenant des graphismes lourds.

La taille de l'application est aussi un inconvénient de Xamarin par rapport à ses concurrents. Comme elle n'est pas comprise par le système d'opération du téléphone mobile, elle emporte avec elle d'autres outils pour la faire fonctionner (runtime, assemblage de libraires de classes de base, etc.) qui l'alourdissent grandement.

Xamarin a été acheté par Microsoft en février 2016, permettant son utilisation gratuitement. Il a également intégré le framework à son IDE (environnement de développement) Visual Studio. Il est conseillé d'obtenir Visual Studio pour créer une application Xamarin. Une version gratuite de Visual Studio existe mais elle n'offre pas toutes les fonctionnalités qui permettent de travailler correctement sur le projet. Pour profiter pleinement de Visual Studio et de Xamarin, il est essentiel d'acquérir une licence Visual Studio.

5.2.2.4 *React Native*

React Native est un framework de développement mobile basé sur React pour créer et composer des interfaces riches à partir de composants déclaratifs codés en JavaScript. React Native se distingue des autres outils de développement mobile utilisant les technologies WEB car il ne crée par une interface tournant dans une WebView (comme Ionic) mais il propose un système semblable à Xamarin qui permet la création d'interfaces utilisateurs natives. React Native va, en quelque sorte, encapsuler les composants natifs dans des composants compréhensibles pour le framework [29].

Ce système impose à React Native de créer les interfaces dans un langage différent du HTML, le JSX. Cette méthode a été choisie car React part du principe que la logique du rendu est

fortement couplée à la logique d’affichage des interfaces utilisateur : comment les évènements sont traités, comment l’état de l’application change au fil du temps et comment les données sont préparées pour l’affichage [29].

5.2.2.4.1 Architecture

Chaque composant est directement branché au composant natif dont il fait référence. Aucune couche supplémentaire n’est nécessaire pour communiquer entre les composants natifs et ceux offerts par React Native.

Il est néanmoins important de comprendre que l’on n’interagit pas immédiatement avec les composants natifs du téléphone. React Native reste un framework utilisant le langage JavaScript et ne peut donc pas directement être lancé sur un téléphone Android ou IOS.

C’est pour cela que React Native utilise un environnement de runtime lié à JavaScript comme le montre la figure suivante.

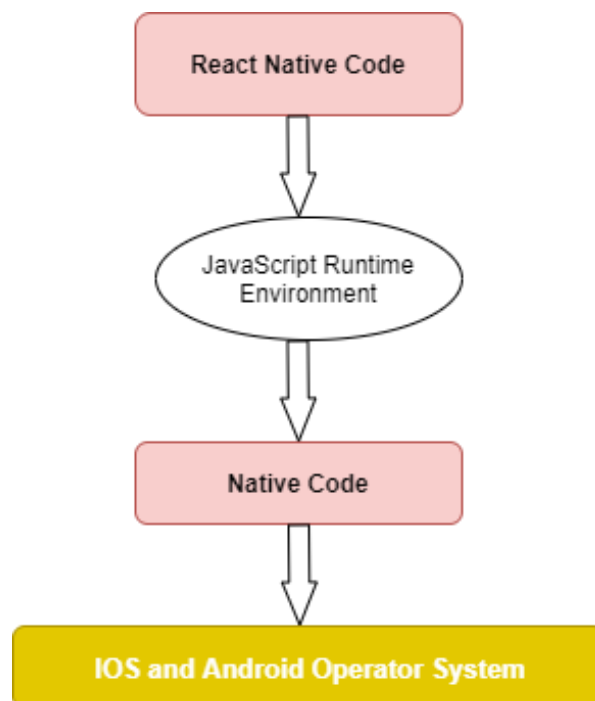


Figure 35 : Fonctionnement du framework React Native

L’application est alors lancée dans cet environnement, ce qui permet d’accéder au langage JavaScript et de la faire fonctionner. Des ponts sont ensuite créés entre les composants offerts par React Native et les composants natifs du téléphone. Le reste de l’application communique ensuite uniquement via les composants React Native qui sont eux-mêmes connectés aux composants natifs du téléphone.

L'interface créée via le framework React Native n'est donc pas une interface « HTML/CSS ». L'interface est semi-native et est lancée directement sur la couche native du téléphone. React Native n'a pas besoin d'une WebView pour afficher l'application. La disparition de la couche WebView ainsi que de la couche d'accès aux composants augmentent grandement les performances d'une application React Native. La figure suivante montre la différence qui existe entre les frameworks utilisant la WebView et React Native.

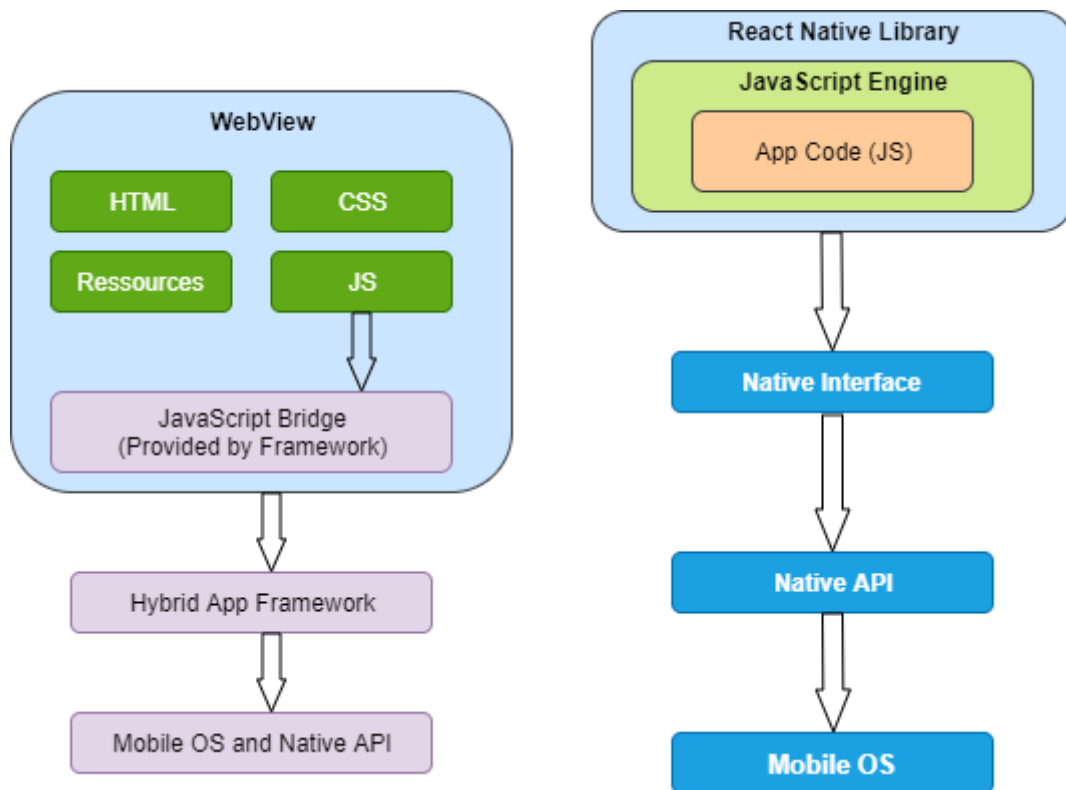


Figure 36: Différence entre frameworks WebView et React Native

On peut voir sur cette image que React Native ne dispose pas de WebView. L'interface utilisateur qui est dans la WebView pour une application JavaScript/HTML, est ici directement dans la boîte « Native Interface ».

5.2.2.4.2 Avantages

Le point important à retenir du framework React Native est la manière dont il lie le framework aux composants natifs du téléphone. Cela permet des performances accrues, très proches d'une application native tout en gardant un développement cross-plateforme (un seul langage pour chaque plateforme). Evidemment, chaque système d'exploitation a ses spécificités et il est nécessaire d'adapter légèrement son code pour chaque plateforme ciblée. De 90% à 100% du code crée pourra être partagé entre toutes les plateformes. Il est possible également de créer ses

propres composants natifs (en Objective-C ou Java) et de les implémenter directement dans le framework React Native. Comme le framework utilise des ponts pour communiquer, il est très simple de rajouter ses propres composants.

Toute la partie logique de l'application étant codée en JavaScript, aucun apprentissage spécifique ne doit normalement être fait pour apprendre ce langage très connu. Un point important de React Native est son système de débogage. Il est possible de lancer son application directement sur son téléphone et de pouvoir déboguer en temps réel les problèmes depuis son ordinateur. Ce système est très semblable à celui d'Ionic.

5.2.2.4.3 Faiblesses

Comme React Native est basé sur React, une connaissance de ce dernier est conseillée. React Native crée une encapsulation des composants natifs du téléphone et ne propose donc qu'une petite collection de composants. Pour créer des interfaces compliquées, il faudra empiler ces composants basiques les uns dans les autres, chose qui peut vite se révéler complexe.

Le plus gros désavantage réside dans l'ajout de composants externes / tierces à l'application. La communauté a conçu des composants et modules augmentant les possibilités de création de React Native qu'il faudra installer un par un. Malheureusement, ces composants ne sont pas toujours mis à jour et souvent, ils ne vont pas pouvoir être implémentés au projet.

5.2.2.5 Comparaison des frameworks

Chaque framework comporte des avantages et des inconvénients. Une application créée via le framework Ionic pourra être personnalisée mais sera vite limitée au niveau des performances. Xamarin se trouve être un juste milieu permettant d'obtenir de bonnes performances et un accès aux composants natifs simplifiés. Toutefois, Xamarin limite la création d'interface et oblige à connaître le langage C#. React Native propose un mélange des deux autres frameworks avec des performances proches du natif et un développement simplifié cross-platform mais une personnalisation moyenne. Le tableau suivant résume les performances, les aspects techniques et de développement de ces différents outils de travail.

Framework / Caractéristiques	Ionic	Xamarin	React Native
Performances	Mauvaises	Bonnes	Bonnes
Outils de développement moderne	Partiel	Non	Oui
Support des fonctionnalités natives	Partiel	Oui	Partiel
Connaissances techniques	Faibles	Hautes	Moyennes

Tableau 11 : Résumé des frameworks

On peut comprendre ici que chaque framework à ses avantages et ses inconvénients :

- Xamarin est destiné à des développeurs confirmés connaissant bien le langage C#. Il est également le framework idéal pour des projets nécessitant des performances accrues mais sans interface utilisateur compliqué.
- React Native est un bon compris proposant de bonnes performances ainsi que des aides au développement poussées tout en ne nécessitant pas trop de connaissances supplémentaires aux technologies WEB.
- Ionic est le framework le plus simple à prendre en main avec des possibilités de personnalisation énormes. Il est malheureusement aussi le moins performant des trois.

Les trois frameworks qui ont été présentés précédemment sont très intéressants. Chacun a ses avantages et ses inconvénients. Après avoir analysé les fonctionnalités et besoins de l'application, j'ai estimé que React Native était le moyen le plus adapté aux besoins de du système que je souhaite développer. Il offre des performances accrues ainsi que des accès aux fonctionnalités natives complètes. De plus, des libraires peuvent y être ajoutées afin d'offrir une expérience utilisateurs améliorée.

5.2.3 La sécurité

5.2.3.1 Définition

Les enjeux stratégiques et économiques liés à l'écoute de réseau de communication et l'impact médiatique des attaques informatiques sont tels que la définition et la mise en œuvre d'une

politique de sécurité sont devenues une activité de tout premier plan qu'il est aujourd'hui impossible d'ignorer.

Au cours du bon fonctionnement des entreprises, les systèmes d'information sont devenus des cibles d'attaques informatiques privilégiées (virus, intrusion, usurpation, etc.) dont l'impact est extrêmement préjudiciable à l'organisation. Toutes ces attaques, menacent quotidiennement la vie des entreprises. La sécurité dans un système d'information consiste à s'assurer que celui qui consulte ou modifie les informations en a l'autorisation et le fait dans le cadre établi.

Il s'agira concrètement dans cette section, de parler de : pourquoi la vulnérabilité des systèmes d'information, quelques méthodes pour se protéger contre les attaques, et nous finirons par la présentation de la sécurité minimale et globale de notre application.

5.2.3.2 Quelques méthodes de défense

5.2.3.2.1 Authentification

Vérifier la véracité des utilisateurs, du réseau et des documents. Celle-ci consiste à déterminer l'authenticité de ceux qui accèdent au système. L'authentification des utilisateurs permet la non-répudiation des données. Cette opération peut utiliser plusieurs techniques :

- PIN (mot de passe, login) ;
- Caractéristiques physiques (empreintes digitales, reconnaissance vocale, ou par visage)

L'accès à un système d'information par mot de passe et login est la solution la plus simple et la plus utilisée. Dans le cadre de l'application, le client utilise son numéro de téléphone et son mot de passe fournis lors de la création de son compte pour accéder aux services du système.

5.2.3.2.2 Contrôle d'accès aux ressources

L'accès aux ressources doit être contrôlé. Il s'agira même de l'accès physiques. La base de données étant accessible en ligne via une connexion internet et un navigateur, il n'est réservé qu'à un nombre restreint d'individu de pouvoir accéder, visualiser et gérer les données des utilisateurs. Seuls les administrateurs avec un compte d'accès ont la possibilité de se connecter aux ordinateurs liés à la base de données de l'application.

5.2.3.3 Politique sécuritaire minimale de notre système

Nous avons parlé de la sécurité de données à plusieurs endroits de ce travail. Cette attention à la sécurité informatique prouve à suffisance les dangers et les menaces permanents des attaques

informatiques ainsi que notre ferme décision à assurer l'intégrité et la permanence de données de notre système d'information.

Nous avons développé une politique à deux volets :

- Au niveau de la base de données, la sécurité va consister à la protection contre les destructions accidentelles, à l'intégrité et à la permanence de données ainsi qu'à la maintenance et à la reprise sur pannes.
- Au niveau du système d'information, la sécurité va consister à accorder des droits d'accès aux utilisateurs et à contrôler leurs accès au système d'information. Ce contrôle est dans le cadre de s'assurer que la personne qui accède au système en a la permission et le fait dans le cadre établi.

5.2.3.4 Politique sécuritaire globale

5.2.3.4.1 Au niveau de l'application

- Identification et authentification des utilisateurs ;
- Création d'une table journal recueillant toutes les informations sur toutes les sessions ouvertes sur l'application (heure et date de connexion) ;
- Accord de droit d'accès ;
- Désactivation et masquage d'interface selon la catégorie de l'utilisateur (simple ou marchand)

5.2.3.4.2 Au niveau du réseau informatique

Le travail en réseau accroît l'insécurité de données. Nous nous sommes alors demander, comment assurer l'intégrité et la permanence de données dans un environnement multiutilisateurs ?

Pour répondre à cette question, notre politique sécuritaire bien que minimale, a prévu à tous les niveaux : l'identification et l'authentification des utilisateurs ; la protection physique des équipements informatiques contenant des données sensibles, la configuration de firewall et de serveur proxy, l'installation d'anti-virus mis à jour, la cryptographie de données ainsi que la sensibilisation des utilisateurs.

5.3 Implémentation

5.3.1 Etape de réalisation

Pour réaliser notre système, il y a des étapes à élaborer dans l'ordre suivant :

- **Conception de la base de données** : La conception d'une base de données est la première étape. Le choix des algorithmes et de l'approche de travail exige l'utilisation d'une base de données spécifique (en fonction du système à développer).
- **Extraction des données** : On va utiliser notre SGBD Firebase ainsi que notre base de données en interaction avec l'application pour extraire des informations. Firebase est conçu avec des fonctionnalités et des requêtes de sélection de données assez spécifique et facile à utiliser.
- **Conception et développement du front-end et du back-end** : Cette étape consiste à détailler la conception coté client et coté serveur. Il s'agit de mettre en place un design ergonomique, simple et attractif répondant aux exigences du système. Le choix d'un serveur d'application adéquat aux fonctionnalités et aux données est une étape fondamentale pour le bon fonctionnement de l'application.
- **Développement de back-end** : On commence par le développement de l'application coté serveur, dans notre cas avec Node JS. C'est la partie du code exécuté sur le serveur afin de vérifier le comportement des fonctionnalités de base du système.
- **Développement front-end** : On développe la partie client en interaction avec le serveur. C'est la conception de l'interface graphique utilisateur. En effet, il s'agit de la partie visible de l'application, destinée à être manipulée par un tiers.

5.3.2 Interfaces Homme/Machine

Dans ce qui suit, nous présentons quelques écrans de notre produit final. Nous avons choisi d'appeler l'application PIMO (PI représente le terme « petit » et MO représente le terme « monnaie ») qui constitue le terme « Petite Monnaie ».

- ❖ **L'interface d'authentification** : La figure suivante représente l'interface d'authentification de notre application. Elle permet aux utilisateurs de s'identifier en introduisant leurs identifiants afin d'accéder aux fonctionnalités de l'application.

The image shows a mobile application login screen. At the top, there is a red header with the word 'Identification' in white. Below the header is a light green background. In the center, there is a logo consisting of a stylized 'P' with a red and black design. Below the logo, the text 'Veillez entrer vos données d'accès' is displayed. There are two input fields: the first is labeled 'Numéro téléphone' and has a phone icon on the left; the second is labeled 'Mot de passe' and has a lock icon on the left and an eye icon on the right. Below these fields is a blue button labeled 'Se connecter'. At the bottom of the screen, there are three links: 'Créer votre compte', 'Mot de passe oublié ?', and 'Conditions d'utilisation'.

Figure 37 : Interface d'authentification

La demande d'identification du client est traitée pour vérifier ses paramètres dans la base de données. L'absence de l'utilisateur dans la base de données ou une erreur de saisie des informations entraîne une alerte d'erreur d'authentification.

- ❖ ***L'interface d'inscription*** : Avant de pouvoir s'authentifier, l'utilisateur doit impérativement s'enregistrer au préalable dans la base de données. La figure suivante représente l'interface de création de compte pour un client.

The image shows a mobile application registration screen. At the top, there is a red header with the word "Inscription" in white. Below the header is a logo consisting of a stylized 'P' with a red and black design. Underneath the logo, the text "Veuillez entrer vos informations" is displayed. There are four input fields, each with a blue border and a corresponding icon on the left: a person icon for "Nom complet", a document icon for "CNI", a telephone icon for "Numéro téléphone", and a lock icon for "Mot de passe". To the right of the password field is a blue eye icon for toggling visibility. Below these fields is a prominent blue button with the text "Créer votre compte". At the bottom of the screen, there is a link that says "Connectez-vous".

Figure 38 : Interface d'inscription

- ❖ ***L'interface du compte utilisateur*** : Cette page principale de l'application affiche le solde courant de l'utilisateur. C'est à partir de cette fenêtre qu'est accessible la majorité des autres interfaces et fonctionnalités de base de l'application. La figure suivante représente l'interface du compte d'un client donné.

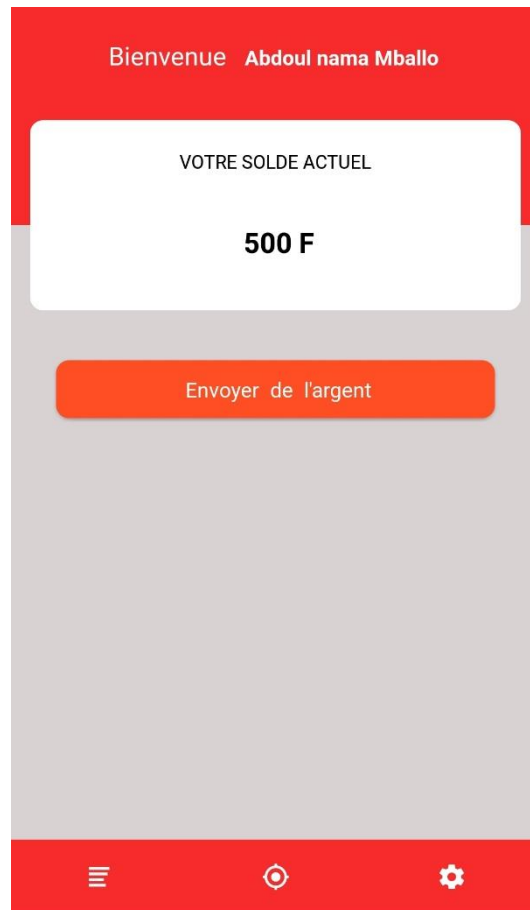


Figure 39 : Interface du compte principal

- ❖ ***L'interface de transfert*** : La figure suivante montre la fenêtre de transfert de petite monnaie vers un utilisateur en saisissant son numéro de téléphone et le montant à envoyer. Une liste des transferts les plus récents s'affiche en dessous du formulaire de saisie.

← Transfert

Veuillez saisir les données de transfert

Numero téléphone

Montant

Valider

Transferts récents

777212391	27-07-2020 11:07	250 F
777212391	27-07-2020 11:07	150 F
785426143	28-07-2020 20:19	500 F

Figure 40 : Interface de transfert

5.3.3 Tests et optimisations

Dans cette partie, nous présentons les différents tests réalisés pour vérifier le bon fonctionnement de notre application. Pour tester et optimiser l'application, il est indispensable de le compiler sur un smartphone ou une tablette Android. Durant cette phase, nous avons rencontré des problèmes d'intégration et nous avons eu recours de nouveau au code de l'application dans le but de l'améliorer. En plus, les différents tests de l'application ont conduit à plusieurs modifications surtout au niveau du métier.

5.3.3.1 Tests et simulations

Le cycle de vie « incrémental » adopté en adéquation aux principes du Processus Unifié (UP) pour la réalisation du projet, nous permet de découper le système en sous-systèmes et de valider le travail à chaque étape (chaque incrément est validé par des tests unitaires). Dans cette optique, nous avons établi une séquence de tests afin de valider les fonctionnalités envisagées et détecter les éventuels problèmes. Le tableau suivant illustre les principaux tests fonctionnels effectués.

Test 1	Inscription d'un nouvel utilisateur
	Contrôle de saisie (caractères et nombre) lors de l'inscription
	S'assurer du bon fonctionnement de l'authentification
	Manipulation des services de la connexion internet
Test 2	Affichage du solde du client après chaque transaction
	Affichage du nom et prénom lié au compte du client
Test 3	S'assurer du bon fonctionnement du transfert d'argent
	S'assurer du bon fonctionnement de la réception d'argent
	Vérifier l'existence du numéro du destinataire
	Vérifier la disponibilité du solde afin de valider le transfert
	S'assurer de l'affichage de la liste des transferts récents
Test 4	S'assurer du listing de l'historique des transferts effectués
	S'assurer de l'affichage des retraits en détails
	S'assurer de l'affichage des dépôts et réceptions d'argent
Test 5	Affichage de liste des points de distribution
	Afficher l'adresse et les coordonnées d'un distributeur
Test 6	Test de fonctionnement de la déconnexion du client
	Test sur le partage de l'application avec des amis ou parents
	Affichage du profil et possibilité de changer de mot de passe

Tableau 12: Tests et simulations

5.3.3.2 Essais sur divers modèles de smartphone

Les interfaces précédentes ont été prises lors d'un test de l'application sur un Samsung Galaxy S7 qui donne un bon affichage des éléments

- Sur un Samsung J2 : Ayant un écran étroit, ce smartphone nous présente un affichage assez spécifique par rapport aux précédents.

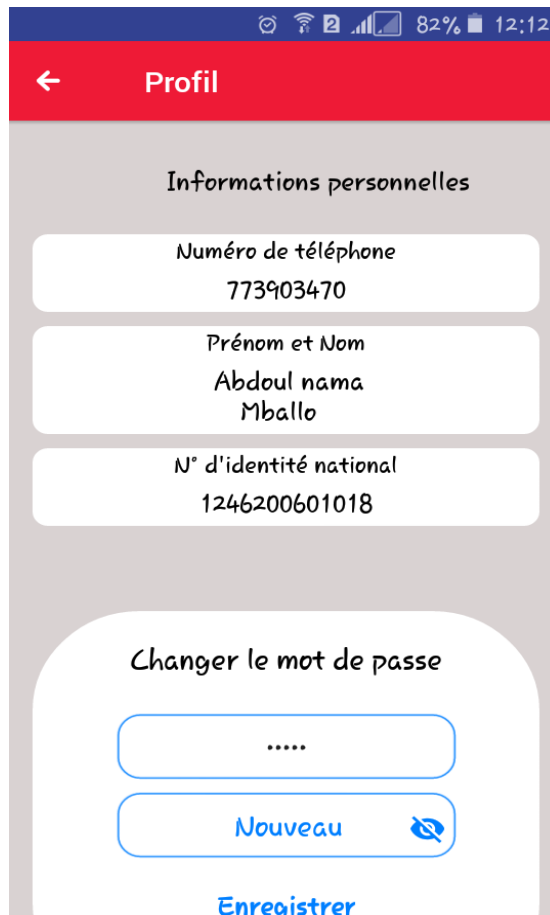


Figure 41 : Interface de Transfert et Profil

Sur cette interface, nous constatons que certains éléments de l'application dépassent la taille de l'écran du téléphone. C'est ainsi que nous avons effectué quelques ajustements et améliorations afin d'avoir une bonne ergonomie sur différents smartphones.

- Sur Alcatel JOY 1 : Ce smartphone est plus petit que le Samsung J2, alors l'affichage n'était aussi bonne et présentable sur les interfaces suivantes :

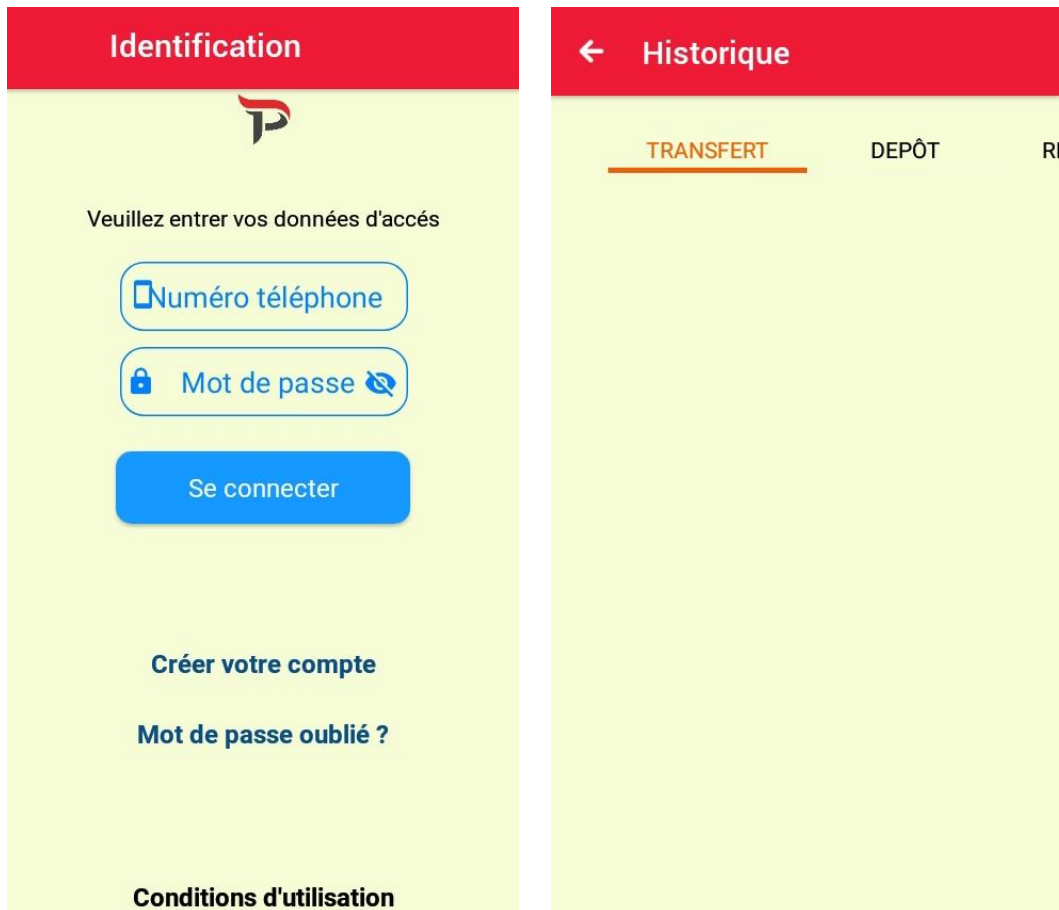


Figure 42: Interface Login et Historique sur un Alcatel JOY 1

La taille de l'écran étant petite, les éléments de l'interface sont compressés afin de pouvoir s'afficher. Nous remarquons même que l'onglet « Retrait » de l'historique a dépassé l'extrémité droite de l'écran, un aspect à régler. Ainsi pour remédier à cela, nous avons diminué la taille de police du smartphone qui était un peu grande par défaut. Ce qui nous a permis d'obtenir un meilleur affichage.

5.3.3.3 Vérification et validation

La publication de l'application est prévue avant la fin de l'année 2020. Ainsi, elle sera disponible dans les différentes plateformes de téléchargement et sera fonctionnelle pour tout utilisateur. Nous avons encore beaucoup de tests et de vérification de fonctionnalités à effectuer. Il faudra une équipe de validation du produit et un taux de 80% des fonctionnalités de base pour pouvoir procéder à sa publication. Dans le même sens, nous sommes en train de réaliser un système de gestion et de visualisation de toutes les activités des utilisateurs sur l'application. Ce système sera accessible et géré que par l'administrateur de la solution mobile.

Les clients sont rangés par catégorie (simple, marchand, et distributeurs). Seul l'administrateur peut attribuer des privilèges à ces clients pour aller d'un statut à un autre, et de pouvoir bénéficier de certains services et avantages.

5.3.4 Les contraintes du système

La réalisation du mémoire ne s'est pas fait sans contraintes à savoir l'ampleur du projet, la connaissance et la maîtrise de l'outil de travail, l'intervention de nouvelles méthodes de développement d'application, l'accès à certains documents de travail souvent écrit en Anglais, etc. Concernant les contraintes de notre système, nous pouvons évoquer celle d'ordre technique et logicielle.

Les contraintes techniques sont relatives aux matériels utilisés pour la réalisation de notre système d'information. L'une des principales difficultés est de disposer un ordinateur capable de lancer le projet et l'émulateur en même temps. Le démarrage de l'émulateur d'Android ralentissait considérablement l'exécution des autres tâches. Il s'y ajoute un dysfonctionnement de certains boutons du clavier de la machine de développement, dont l'alternative était de trouver un clavier externe. L'installation des modules du framework, l'accès à la base de données et le test de l'application exigent tous une disponibilité de la connexion internet. Toutes ces contraintes matérielles ont eu un impact considérable pour la réalisation du système entraînant un ralentissement des travaux d'où le retard que nous avons accusé.

Pour les contraintes logicielles, un certain nombre d'outils est nécessaire pour l'environnement de travail. Il s'agit des :

- ✦ **Node JS** : Indispensable pour utiliser npm (Node Package Manager) qui est le gestionnaire de paquets officiel de Node.js. npm fonctionne avec un terminal et gère les dépendances pour une application.
- ✦ **Git** : C'est un logiciel de gestion de versions décentralisé. Facilite l'utilisation de npm, l'exécution et le débogage de l'application.
- ✦ **Android studio** : Il n'est pas utilisé pour développer l'application. Cependant il est accompagné du SDK (Software Development Kit) qui est indispensable pour la réalisation du projet. Android Studio nous fournit un large éventail d'émulateur pour des besoins de test tout au long du développement.

Conclusion

Dans ce chapitre, avons décrit brièvement le processus de développement de notre application de transfert de petites monnaies en spécifiant l'environnement de développement, le choix des outils ainsi que la démarche suivie pour la réalisation.

En effet, nous avons achevé l'implémentation tout en respectant la conception, mais les tests des différents cas d'utilisations sont toujours en cours en vue d'une amélioration de l'application avant sa publication.

En d'autres termes, nous avons la version beta (test et amélioration avant sa publication) de l'application installée dans notre environnement de développement. Aussi, nous avons prévu la période pendant laquelle la solution finale sera déployée sur les plateformes de téléchargement.

Conclusion générale et perspectives

Face à un manque remarquable de la petite monnaie dans nos échanges, nous avons étudié, conçu et réalisé à travers ce mémoire un système (application) mobile qui permet aux utilisateurs d'effectuer leur échange dans le calme et faciliter la récupération et l'utilisation de leur petite monnaie.

Nous avons dans un premier temps exposé la question de la petite monnaie (contexte, problématique, objectifs, etc.) dans le but d'apporter quelques éclaircissements sur les notions en relation avec le projet. Nous avons présenté dans la deuxième partie l'état de l'art du sujet à travers l'étude de quelques systèmes de transfert déjà existants. Nous avons également évoqué l'univers des appareils et applications mobiles, puis expliqué les technologies et outils à utiliser pour la réalisation d'une solution mobile adaptée aux besoins des utilisateurs. Ensuite nous avons entamé les différentes étapes du processus de développement UP, afin de mettre en œuvre notre solution et d'apporter des réponses à la problématique.

Nous avons commencé par l'identification des besoins qu'on a ensuite modélisé sous forme de diagrammes de cas d'utilisation, diagrammes de séquences, diagrammes de classes et terminer par la réalisation de l'application. Lors de ce travail, nous avons pu mettre en pratique nos connaissances théoriques acquises durant notre formation.

Ce projet a fait l'objet d'une expérience intéressante. Il nous a permis de comprendre que le projet de réalisation d'une application mobile est un ensemble de plusieurs actions planifiées et dépendantes les unes des autres. Toutes les étapes de ce projet nous ont permis d'enrichir notre expérience notamment dans les différents outils et langages dédiés à la programmation mobile.

Nous avons retenu également que la réussite d'une application mobile demande une bonne organisation et une cohérence tout au long de la phase de réalisation. Bien que notre application n'est pas encore finie, nous avons l'intention de la compléter et l'améliorer en terme de design (ergonomie), et quelques fonctionnalités tel que les notifications, l'itinéraire (géolocalisation), la recherche avancée, la sécurité et la déployer au niveau des boutiques d'application afin de pouvoir l'exploiter.

Bibliographie

- [1] S. Madise, *The regulation of mobile money*. New York, NY: Springer Berlin Heidelberg, 2019.
- [2] E. M. Ndadoum et B. Kordjé, *Mobile money en Afrique - Son rôle pour l'inclusion financière au Tchad*. L'Harmattan, 2020.
- [3] O. Fédior, « Mobile money/Mobile banking: La guerre des transferts - OSIRIS: Observatoire sur les Systèmes d'Information, les Réseaux et les Inforoutes au Sénégal », mars 08, 2019. <http://www.osiris.sn/Mobile-money-Mobile-banking-La.html> (consulté le juill. 27, 2020).
- [4] B. A. Lassaad, « Cameroun: pénurie inédite de pièces de monnaie. », janv. 10, 2020. <https://www.aa.com.tr/fr/afrique/cameroun-pénurie-inédite-de-pièces-de-monnaie/1698715> (consulté le juin 13, 2020).
- [5] S. Andzongo, « La Banque centrale de la Cemac ouvre une enquête sur l'exportation des pièces de monnaie au Cameroun. », *Investir au Cameroun*, déc. 19, 2018. <https://www.investiraucameroun.com/finance/1912-11901-la-banque-centrale-de-la-cemac-ouvre-une-enquete-sur-lexportation-des-pieces-de-monnaie-au-cameroun> (consulté le juin 13, 2020).
- [6] OECD et S. and W. A. Club, *Écoloc, Gérer l'économie localement en Afrique: Evaluation et prospective Le financement de l'urbanisation et le développement local*. OECD Publishing, 2002.
- [7] CAVIE, « La crise de la « petite monnaie » en zone CFA | CAVIE | Centre Africain de Veille et d'Intelligence Economique ». .
- [8] S. MBODJI, « WANTED, Les Petites pieces de monnie ont disoaru de la circulation », *SenePlus*, avr. 12, 2019. <https://www.senepius.com/economie/wanted-les-petites-pieces-de-monnaie-ont-disparu-de-la> (consulté le juin 11, 2020).
- [9] J.-M. Albertini, *Les rouages de l'économie*. Editions de l'Atelier, 2008.
- [10] D. Benjamin, *La monnaie et les banques dans l'économie*. Educa Vision Inc., 2005.
- [11] Y. BRUGIERE, « Cercle numismatique de Nice, dans son article intitulé: « Au secours du petit commerce Niçois, Les actions monétaires de la Chambre de commerce de Nice (1917-1922): Difficultés et réalisations ») ». http://numismatiquenice.eu/page_3.html (consulté le juin 12, 2020).
- [12] D. D. Croushore et T. Stark, *Forecasting coin demand*. Federal Reserve Bank of Philadelphia, 2002.
- [13] M. Bridier et S. Michailof, « Guide pratique d'analyse de projets. Librairie Eyrolles », 1995. .
- [14] H. Wallot, *Développement entrepreneurial. Recueil de textes*. Sainte-Foy, QC: Télé-Université, 1996.
- [15] J.-P. Boutinet, *Psychologie des conduites à projet*. 2014.
- [16] C. Speth, « La Matrice SWOT : Élaborer un plan stratégique pour votre entreprise », 2015. .
- [17] P. Bezbakh, « II. Les grandes théories de l'inflation », *Reperes*, vol. 6e éd., p. 25-46, 2011.
- [18] J. Rivoire, *L'inflation*. Éditions France-Empire, 1973.
- [19] Y. SOW, « Tamweel Touch : La Nouvelle application mobile de la BIS », *La référence du Business au Sénégal*, juill. 16, 2019. <http://reussirbusiness.com/actualites/tamweel-touch-la-nouvelle-application-mobile-de-la-bis/>.
- [20] R. O. Mason et I. I. Mitroff, « A Program for Research on Management Information Systems », *Manag. Sci.*, vol. 19, n° 5, p. 475-487, 1973.
- [21] « Systèmes d'information pour le management Gordon B. Davis, Margrethe H. Olson, Jacques Ajenstat... [et al.] ».

http://bibliotheque.bordeaux.fr/in/details.xhtml?id=mgroup%3Ap+unimarcbu_107567
(consulté le juin 10, 2020).

- [22] *Systèmes d'information et management des organisations - Robert Reix... - Librairie Eyrolles.* .
- [23] J. Lonchamp, *Introduction aux systèmes informatiques: architectures, composants, mise en œuvre.* Malakoff: Dunod, 2017.
- [24] W. Ben Abdesslem Karaa, *Le Processus Unifié par la pratique Cours et étude de cas.* 2012.
- [25] I. Jacobson, G. Booch, et J. Rumbaugh, *Le processus unifié de développement logiciel.* Paris: Eyrolles, 2000.
- [26] P. Roques et F. Vallée, *UML 2 en action: de l'analyse des besoins à la conception.* Paris: Eyrolles, 2009.
- [27] P. André et A. Vailly, *Développement de logiciels avec UML 2 et OCL cours, études de cas et exercices corrigés.* Paris: Ellipses, 2013.
- [28] L. Debrauwer, *De UML à Java: conception et réalisation d'une application web dans le domaine de la sémantique.* St. Herblain: ENI, 2012.
- [29] B. Barré, *Concevez des applications mobiles avec React Native: développement, publication sur les stores et stratégie marketing.* 2020.
- [30] S. Hébuterne et S. Pérochon, *Android guide de développement d'applications Java pour smartphones et tablettes.* Saint-Herblain: Éd. ENI, 2014.
- [31] C. Gilbert, *Conception d'interfaces pour mobiles: graphisme et développement des applications natives, web et hybrides.* 2017.
- [32] J.-E. Pelet, *M-commerce: du design d'interface à l'optimisation des ventes.* Pearson Education France, 2014.
- [33] A. S. Tanenbaum, *Systèmes d'exploitation: [avec plus de 400 exercices.* Paris: Pearson Education, 2008.
- [34] M. Martin, *Créez des applications pour iPhone, iPad et iPod Touch: Le développement pour appareils mobiles Apple à la portée de tous.* 2012.
- [35] P. Antouly, *Smartphones sous Windows mobile 2003: découverte, prise en main et utilisation.* Nantes: Éd. ENI, 2006.
- [36] T. Parisot, *Node.js: apprendre par la pratique.* 2019.
- [37] E. Sarrion, *Programmation avec Node.js, Express.js et MongoDB: JavaScript côté serveur.* 2014.
- [38] N. Symth, *Firestore Essentials.* Cary: Payload Media, 2017.
- [39] J. Crowther, *Firestore.* London: Constable, 2015.
- [40] C. Vigouroux, *Apprendre à développer avec JavaScript.* Saint-Herblain: Éd. ENI, 2014.
- [41] E. Sarrion, *React.js: [le framework JavaScript de Facebook.* 2019.

Résumé : Le but de ce travail est de répondre à une problématique qui tourne autour de la rareté de la 'petite monnaie' (transactions à faible valeur). L'étude de ce problème montre que malgré de nombreuses solutions de transfert d'agent en Afrique et particulièrement au Sénégal, les transactions dues à la « petite monnaie » restent encore une difficulté.

Nous avons donc conçu une solution dont le but est de gérer électroniquement la petite monnaie. Cela permettra aux utilisateurs de gagner du temps durant leurs transactions et diminuera les conflits liés à la rareté de la petite monnaie. Le développement de la solution s'est basé sur le Processus Unifié (UP) comme méthode de modélisation et UML (Unified Modelling Language) comme langage de modélisation.

La solution est implémentée en application mobile bénéficiant des outils Google pour la gestion des données (Firebase Realtime Database) ; la sécurité (Firebase Authentication) et l'envoi des messages (Firebase Cloud Messaging). L'application baptisée « Pimo » a été testée sur plusieurs smartphones avec le système Android tels que Samsung (Galaxy S7, A20), Alcatel. Des tests non concluants ont été faits également sur l'iPhone 6. Dans les perspectives, nous comptons rendre Pimo opérationnel sur la majorité des smartphones et développer une version web de Pimo.

Mots clés : Petite monnaie, Conception, Développement, Application mobile, UP, UML, React Native, Framework, Firebase.

Abstract : The aim of this work is to respond to a problem that revolves around the scarcity of 'small change' (low value transactions). The study of this problem shows that despite numerous agent transfer solutions in Africa and particularly in Senegal, transactions due to "small change" still remain a difficulty.

We have therefore designed a solution whose goal is to electronically manage small change. This will allow users to save time during their transactions and reduce conflicts linked to the scarcity of small change. The development of the solution was based on the Unified Process (PU) as the modeling method and UML (Unified Modeling Language) as the modeling language.

The solution is implemented as a mobile application benefiting from Google tools for data management (Firebase Realtime Database); security (Firebase Authentication) and sending messages (Firebase Cloud Messaging). The application called "Pimo" has been tested on several smartphones with the Android system such as Samsung (Galaxy S7, A20), Alcatel. Inconclusive tests were also made on the iPhone 6. In the future, we intend to make Pimo operational on most smartphones and develop a web version of Pimo .

Keywords: Small change, Design, Development, Mobile application, UP, UML, React Native, Framework, Firebase