

DEDICACE

A MES TRÈS CHERS PARENTS

Je dédie ce mémoire à mes parents, pour l'amour qu'ils m'ont toujours donné, leurs encouragements et toute l'aide qu'ils m'ont apportée durant mes études.

Aucun mot, aucune dédicace ne pourrait exprimer mon respect, ma considération, et mon amour pour les sacrifices qu'ils ont consentis pour mon instruction et mon bien-être.

Trouvez ici, chère mère et cher père, dans ce modeste travail, le fruit de tant de dévouement et de sacrifices ainsi que l'expression de ma gratitude et de mon profond amour.

Puisse Dieu leur accorder santé, bonheur, prospérité et longue vie afin que je puisse un jour combler de joie leurs vieux jours.

A MES FRÈRES, SŒURS ET AMIS

Je leur dédie ce travail pour tous les sacrifices qu'ils n'ont cessé de m'apporter tout au long de mes années d'études.

Que Dieu leur apporte le bonheur, les aide à réaliser tous leurs vœux et leur offre un avenir rempli de succès. Particulièrement à Ndeye Mareme DIOP, Maimouna DIOP, Ndeye Binta DIOP, Die Astou DIOP, Omar Sy et Tonton Ousseynou DIOP, Adama DANFA, Fatou Binetou NIANG pour leurs soutiens inconditionnels et intarissables qui m'ont permis de poursuivre mes ambitions et d'affronter les réalités de la vie sans crainte.

Et enfin mes pensées se tournent vers mes teams THREE-MONTH et WAKILYTECH qui m'ont beaucoup soutenu moralement. Donc un grand merci pour elles.

REMERCIEMENTS

Tout d'abord, je rends grâce à ALLAH le tout puissant, et pries au nom du prophète Mohamed (PSL), nous tenons à exprimer également mes vifs remerciements à tous ceux qui, par leurs travaux, leurs amours, leurs collaborations, leurs financements, ont contribué de près ou de loin à la réalisation de cet ouvrage en particulier :

J'exprime mes profonds remerciements à mon encadreur Mr. Baye NIASS, votre enseignement a renforcé en nous le sens de la rigueur et le goût du travail bien fait. Ces remerciements ne seraient terminés sans une pensée à l'ensemble du corps professoral de l'université Amadou Hampaté Bâ de Dakar, particulièrement à Mr. Moustapha DER, Mr. Kara SAMB, Mr Aliou BOLY et Mr. Lamine DIOP, qui n'ont ménagé aucun effort pour la réussite de nos études et ont inculqué en nous des valeurs qui, certainement, faciliteront notre intégration dans la vie active.

Je souhaite remercier également mes amis particulièrement à Djibril Diop, Baye Mor NDAO, Djibril MBAO, Omar Ka et Pape Sidy CISSE et les différentes personnes que j'ai pu croiser durant ces trois dernières années et qui ont su par leurs questions et leurs conseils me permettre de toujours pousser un peu plus notre raisonnement.

AVANT PROPOS

L'Université Amadou Hampaté BA de Dakar (UAHB), est un nouveau modèle d'université au Sénégal. Autorisée par l'Etat (0040/AG/ME/DES). Il forme des professionnels dans plusieurs domaines, notamment l'informatique, pour des niveaux d'études différents.

Ces formations sont sanctionnées par des diplômes, à savoir : Licence Professionnelle et Master.

En vue de l'obtention de leur diplôme de fin de formation, Les étudiants de l'UAHB se doivent de présenter un mémoire sur un thème de recherche choisi d'un commun accord avec le Directeur de Mémoire.

C'est dans ce cadre que nous réalisons ce document qui souhaite offrir des pistes de réflexion et de nouvelles perspectives sur les portefeuilles électroniques en Afrique plus particulièrement au Sénégal. À travers l'étude de la conception et de la mise en place d'une plateforme de gestion de portefeuille électroniques. Notre souhait est de pouvoir mettre en évidence les freins aux développements des flux de transactions au Sénégal et de profiter de ce contexte pour pouvoir proposer certaines solutions.

LISTE DES FIGURES :

Figure 1 Interface de l'application Orange Money Sur Android.....	17
Figure 2:Interface de l'application Wave sur Android.....	18
Figure 3: Diagramme de cas d'utilisation général du système.....	27
Figure 4: Diagramme de cas d'utilisation création compte Épargne	30
Figure 5: Diagramme de cas d'utilisation Gestion demande de crédit.....	33
Figure 6: Diagramme d'activité du cas Gestion dépôt (épargne).....	36
Figure 7: Diagramme d'activité du cas Gestion credit (d'emprunt))	37
Figure 8: Diagramme de classe	39
Figure 9: Représentation architecture 1-tier	47
Figure 10: Représentation architecture 2-tiers	48
Figure 11: Représentation architecture 3-tiers	49
Figure 12: Représentation architecture n-tiers	50
Figure 13: Diagramme de déploiement.....	51
Figure 14: Interface Administration	53
Figure 15: Interface Agence	54
Figure 16: Interfaces Ecran de Présentation(Connexion et Inscription)	54
Figure 17: Ecran d'accueil	55
Figure 18: Ecran menu de transfert	55
Figure 19: Ecran transfert en cash.....	55

LISTE DES TABLEAUX :

Tableau 1: avantages et inconvénients des wallets (orange money et wave).....	20
Tableau 2: Description textuelle création compte	28
Tableau 3: Scénario nominal création compte	28
Tableau 4: Scénario alternatifs création compte	29
Tableau 5: Description textuelle création compte épargne	30
Tableau 6: Scénario nominal création compte épargne.....	31
Tableau 7: Scénario alternatifs création compte épargne.....	32
Tableau 8: Description textuelle gestion dépôt	32
Tableau 9: Scénario nominal gestion dépôt	32
Tableau 10: Scénario alternatifs gestion dépôt.....	33
Tableau 11: Description textuelle Gestion de crédit	34
Tableau 12: Scénario nominal gestion crédit	34
Tableau 13: Scénario alternatifs gestion crédit	35
Tableau 14: Fiche descriptive des classes	38

TABLE DES MATIERES :

DEDICACE	I
REMERCIEMENTS	II
AVANT PROPOS	III
LISTE DES FIGURES :	IV
LISTE DES TABLEAUX :	V
TABLE DES MATIERES :	VI
LISTE DES ABREVIATIONS :	VIII
RESUME :	IX
ABSTRACT	X
Chapitre 1 : INTRODUCTION GENERALE.....	11
1.1. Contexte.....	11
1.2. Problématique	12
1.3. Motivation	13
1.4. Objectifs	14
1.4.1. Objectif général	14
1.4.2. Objectifs spécifiques.....	14
1.5. Intérêts du Mémoire	14
Chapitre 2 : ETUDE PREALABLE	16
2.1. Introduction	16
2.2. Définition.....	16
2.3. Champ de l'étude	16
2.4. Etude de l'existant.....	17
2.4.1. Analyse de l'existant	17
2.4.2. Critique de l'existant	19
2.5. Proposition des différentes solutions	20
2.5.1. Description détaillée des processus de traitement (fonctionnement).....	21
2.5.2. Description détaillée des processus de traitement des nouvelles fonctionnalités	22
2.6. Conclusion	23
Chapitre 3 : MODELISATION CONCEPTUELLE & ORGANISATIONNELLE	24
3.1. Introduction	24
3.2. Modélisation Conceptuelle	24
3.3. Etude comparative des approches Merise et UML	24
3.3.1. Merise	24
3.3.2. UML	25

3.3.3. Choix et justification.....	26
3.4. Diagramme des cas d'utilisations.....	26
3.4.1. Identification des acteurs.....	26
3.4.2. Les diagrammes des cas d'utilisation	27
3.5. Diagrammes d'activités par scénario des cas d'utilisation.....	36
3.5.1. Cas gestion dépôt(épargne)	36
3.5.2. Cas gestion crédit (d'emprunt).....	37
3.6. Modélisation Conceptuelle des données :	38
3.7. Diagramme de classes.....	39
Chapitre 4 : ARCHITECTURE ET REALISATION DE LA SOLUTION.....	40
4.1. Introduction	40
4.2. Choix des langages de développement et de SGBD	40
4.2.1. Frontend	40
4.2.2. Backend	44
4.2.3. Le Système de gestion des bases de données (SGBD)	45
4.2.4. Architecture applicative :	46
4.2.5. Architecture physique du système : diagramme de déploiement :	51
4.2.6. Gestion de la sécurité.....	52
4.3. Capture d'Ecran de la solution	53
4.3.1. Capture d'écran partie Administration	53
4.3.2. Capture d'écran partie Agence	54
4.3.3. Capture d'écran Application Mobile	54
Conclusion et perspectives	57
TABLEAU FINANCIER :	59
BIBLIOGRAPHIE :	61
WEBOGRAPHIE :	61

LISTE DES ABREVIATIONS :

API : Application Programming Interface

CSS : Cascading Style Sheet = Feuille de Style en cascade

EDI : Environnement de Développement Intégré

GPL : General Public License

H-M: Homme-Machine

HTML: HyperText Markup Language

HTTP: HyperText Transfert Protocol

J2RE : Java 2 Runtime Environnement

JS : JavaScript

JVM : Java Virtuelle Machine

MERISE : Méthode d'Etude de Réalisation Informatique par Sous-Ensemble

MVC : Modèle-Vue-Contrôleur

PHP : HyperText Preprocessor

SGBD : Système de Gestion de Base de Données

SGBDR : Système de Gestion de Base de Données Relationnel

SGBDRO : Système de Gestion de Base de Données Relationnel et Objet

SQL: Structured Query Language

UML: Unified Modeling Language

UC : Use Case (Cas d'utilisation)

RESUME :

Face à une faible bancarisation au Sénégal, l'utilisation des portefeuilles électroniques ou wallet est une solution à cette situation. Elles font ainsi intégralement partie de la vie économique des sénégalais.

Elles offrent plusieurs possibilités comme envoyer, recevoir de l'argent, payer ses factures, faire des achats etc.

Cependant leurs usages coûtent cher à long terme avec les frais de transfert, les frais de retrait qui augmentent par rapport à la somme de la transaction effectuée.

Ainsi le portefeuille électronique devient cher à entretenir avec le temps.

C'est dans ce contexte que notre mémoire est centré sur l'étude et la conception de la vision qu'on a du portefeuille électronique, celle d'un wallet permettant d'effectuer les tâches classiques d'un portefeuille électronique tout en offrant la possibilité à l'utilisateur d'auto épargner et de faire des demandes de crédits s'il le souhaite.

Bon nombre des fonctionnalités énumérées dans ce mémoire ont été développer on peut citer entre autres :

- Une plateforme web côté administration ;
- Une plateforme web côté Agence ;
- L'application mobile développer en Android pour l'instant.

ABSTRACT

Given the low level of banking in Senegal, the use of electronic wallets is a solution to this situation.

They are thus an integral part of the economic life of Senegalese people.

They offer several possibilities such as sending and receiving money, paying bills, making purchases etc.

However, their use is expensive in the long term with transfer fees and withdrawal fees that increase in relation to the amount of the transaction carried out.

This makes the e-wallet expensive to maintain over time.

It is in this context that our study focuses on the study and design of the vision we have of the e-wallet, that of a wallet that allows the user to carry out the classic tasks of an e-wallet while at the same time offering the possibility of self-saving and making credits if he so wishes.

The majority of the functionalities listed in this brief have been developed: the web platform on the administration side, on the agency side, the mobile application developed in Android for the moment.

Mots clés : Portefeuille électronique, transferts, retrait, argent, auto épargne, crédit.

Keywords: e-wallet, transfers, withdrawal, money, self-savings, credit.

Chapitre 1 : INTRODUCTION GENERALE

1.1. Contexte

Depuis l'émergence de l'informatique, de nombreuses activités de la vie courante ont été simplifiées. Actuellement, les individus peuvent facilement traiter des informations en se servant des logiciels intégrés au système informatique. Compte tenu de son évolution, ce système caractérise la majorité des grandes entreprises quel que soit le secteur d'activité.

Les ordinateurs remplacent les papiers, les calculatrices, les radios, les boîtes aux lettres, etc. De ce fait, l'objectif des informaticiens est de développer cette technologie dans le but d'améliorer les conditions de vie des populations. L'informatique se sert de nombreux outils pour pouvoir transmettre les informations. Il s'agit en effet d'un moyen de recherche à la fois simple et rapide permettant aux professionnels et aux étudiants d'obtenir des informations sur divers sujets notamment dans le domaine de l'économie.

C'est ainsi avec la forte présence des smartphones et l'internet en Afrique, les paiements par téléphone mobile sont de plus en plus répandus au Sénégal. Pour beaucoup d'habitants, cela apparaît comme une alternative pratique à la bancarisation.

Aujourd'hui, seul un Sénégalais sur cinq possède un compte bancaire¹, selon la Banque centrale de l'Afrique de l'Ouest. Pour pallier à cela, les opérateurs de téléphone ont investi dans le paiement mobile. Ce système permet de retirer, déposer ou transférer de l'argent depuis son téléphone, en un seul clic.

Cependant dans une société où le secteur informel représente une partie significative dans l'économie du pays, les frais occasionnés par l'utilisation quotidienne de porte-monnaie électronique peut être un frein économique à long terme.

En effet, on se rend compte que le coût de la circulation de l'argent est de plus en plus important au niveau des opérateurs monétiques comme Orange money, E-money entre autres. Grâce à leurs techniques de marketing (les promotions), ils soutirent plus d'argent pour

¹ Note d'Information 4e Trimestre 2019 N° 60 - BCEAO

augmenter leurs profits créant ainsi le gaspillage intelligent (c'est à dire créer des promotions par SMS qui motivent les utilisateurs à consommer sans pour autant s'en rendre compte).

Le portefeuille électronique ou encore appelé wallet est un logiciel client, d'application qui permet de stocker, d'envoyer, de recevoir ou d'effectuer des paiements. Ainsi se présentant comme la solution face à la faible bancarisation de la société sénégalaise, il est ainsi nécessaire que le portefeuille électronique évolue vers une solution qui présente plus de services comme l'épargne et le crédit. Ces améliorations des services de transfert d'argent permettront une meilleure gestion de nos économies.

C'est dans ce cadre que nous proposons la conception d'une application de gestion de portefeuille électronique.

1.2. Problématique

Malgré la vitesse de l'évolution des nouvelles technologies numérique, il est indispensable de noter que les porte-monnaie ont des particularités qui lui sont propres.

Du fait de ses caractéristiques, la question du mode de transfert, de coût du matériel et de la sécurité des parties est capitale. Pour assurer la sécurité par exemple, des mesures d'élimination de tous les risques d'interceptions des informations lors des transactions, encore accrues par le caractère ouvert et international du réseau et en cas d'interception, pouvoir rendre les informations inutilisables par le fraudeur. Nous notons une forte mobilité d'argent ainsi qu'une sécurisation basique des transactions des utilisateurs, des frais de transactions énormes et une indisponibilité des agents dans les systèmes présents.

Ce qui est une pièce incontournable du système d'information de la gestion porte-monnaie, indispensable pour une prise en charge optimale et de qualité de l'utilisateur.

Le phénomène de digitalisation a un impact sur tous les secteurs : une forte quantité d'éléments matériels ont été migrée et représentée en données numériques (dématérialisation). Ainsi dans le système bancaire, les transferts inter-comptes se font déjà depuis un bon moment en 100% digital. Par ailleurs, il existe toujours plusieurs opérations qui ne sont pas automatisées (numérisées indirectement).

En Afrique, particulièrement au Sénégal, des sujets comme la mise en place de banques 100% digitales sont de plus en plus présents. La digitalisation représente un enjeu majeur pour les institutions financières afin de garder leur compétitivité et de continuer à exister.

Cependant, la commission nationale de la monétique du Sénégal fait face à un problème de taille : les opérateurs qui possèdent leurs propres plateformes monétiques sont nombreux. De cela découle une évidente difficulté à mettre en place une logique de régulation pouvant être étendue à tous ces acteurs. Rares sont les banques qui ont aujourd'hui atteint une maturité dans le processus de digitalisation au Sénégal (en comparaison au marché mondial, nous sommes en croissance).

Notre positionnement est de contribuer à la phase de migration vers les nouvelles technologies de l'information en proposant des services complètement digitalisés et répondant aux besoins des acteurs de la monétique à tous les niveaux (auto-épargne et crédit).

Partant de la problématique formulée si haut, les points(questions) suivants font l'objet principal de notre investigation :

- Réduction de la masse d'argent physique qui circule, rendre accessible l'option d'auto-épargne et de de crédit et la sécurisation des transactions des utilisateurs ?
- Ces préoccupations constituent le point de départ de notre recherche sur laquelle il nécessite de suggérer une réponse anticipative.

1.3. Motivation

Les services de transfert d'argent occupent une place importante dans le flux monétaire au Sénégal.

Ils sont ainsi utilisés dans les villes et aux coins les plus reculés du pays.

Ainsi notre motivation vient du fait que le portefeuille électronique pourrait être plus qu'un simple wallet, de par l'ajout d'une nouvelle fonctionnalité : l'auto-épargne et du crédit (auto tontine) pour contribuer au confort économique du peuple sénégalais.

1.4. Objectifs

1.4.1. Objectif général

De manière générale, nous allons faire l'étude et la mise en place d'une application mobile qui pourrait bien aider la population dans la gestion des transactions de leurs biens mais aussi leur suivi et d'obtenir la traçabilité des traitements de chaque opération et de donner la possibilité aux utilisateurs d'épargner et de créditer.

1.4.2. Objectifs spécifiques

Nos objectifs spécifiques sont les suivants :

- La mise en place des fonctionnalités de base d'un portefeuille électronique (envoi, retrait d'argent, paiement de factures...),
- Garder la traçabilité d'un utilisateur lorsque celui-ci fait des transactions (envoie où recevoir),
- D'intégrer l'option d'épargne et d'emprunt dans de l'application.
- Assurez la sécurité des transactions au sein de l'application.

1.5. Intérêts du Mémoire

À La fin de notre formation en Licence, l'obligation nous incombe de rédiger un mémoire conciliant toutes les notions acquises durant notre apprentissage de faire une étude qui sera par la suite défendue en vue d'obtention du titre de licencié en Informatique, option génie logiciel, remplissant ainsi notre devoir académique. « **ETUDE POUR LA MISE EN PLACE D'UN SYSTÈME DE PORTEFEUILLE ÉLECTRONIQUE** » est pour nous un sujet à propos car il nous a permis non seulement d'avoir une vue globalisante de Réseaux informatiques mais aussi nous a familiarisé avec cette technologie émergente et en perpétuelle évolution à travers la planète.

Le reste de notre document s'organise comme suit :

- Dans le **premier chapitre**, nous mettrons l'accent sur le champ d'étude de notre application qui est une application de transfert d'argent. Nous présentons une synthèse des solutions existantes sur le marché en discutant les avantages et les

inconvenients de chacune d'elles. Puis nous proposons les différentes solutions aux problèmes soulevés.

- Dans le **deuxième chapitre**, la modélisation conceptuelle de notre solution sera détaillée. Ensuite nous exposons le modèle logique des données.
- Dans le **troisième chapitre**, une étude technique sera présentée où nous décrivons l'environnement de développement matériel et logiciel et nous présentons les différentes fonctionnalités de notre application à travers des captures d'écran.

On terminera par une conclusion générale présentant la synthèse du sujet et les perspectives futures pour améliorer notre application.

Chapitre 2 : ETUDE PREALABLE

2.1. Introduction

L'étude préalable constitue une étape préliminaire pour la réalisation d'une application.

En effet, elle permet d'analyser, d'évaluer et de critiquer le fonctionnement habituel, tout en élaborant la liste des solutions possibles.

Ce chapitre sera réservé pour présenter l'étude préalable de notre projet. Nous commençons par la définition du champ de l'étude et les objectifs à atteindre. Ensuite nous analysons quelques solutions existantes sur le marché en discutant leurs avantages et leurs inconvénients.

L'analyse et la critique de l'existant nous ont permis de cerner nos objectifs afin de développer un système de qualité dans le futur. Enfin, nous proposons les différentes solutions aux problèmes soulevés.

2.2. Définition

Wallet est une nouvelle approche du portefeuille électronique. Il permet ainsi d'effectuer les tâches basiques des portefeuilles électroniques traditionnelles comme déposer ou transférer de l'argent depuis son téléphone, de payer des factures et apporte un lot de nouvelles fonctionnalités : l'épargne et le crédit.

2.3. Champ de l'étude

L'utilisation des **Technologies d'Information et de Communication « TIC »** dans les processus de transmission de biens financiers fait désormais partie de l'environnement fonctionnel et intellectuel de la société sénégalaise.

Avec les **TIC**, nous intéresserons plus particulièrement aux dynamiques portées par les technologies numériques des **transferts d'argent**. Dans une approche générique, on peut considérer les **TIC** comme toute application informatique, participant au fonctionnement d'une solution répondant au besoin de la population. Cela inclut les services et applications informatiques utilisant la technologie du réseau internet à des fins de circulation de l'information de plus en plus rapide et sécurisé.

2.4. Etude de l'existant

Cette section a pour objectif d'étudier le tour sur les solutions de transfert d'argent les plus connues sur le marché. Cette étude permet de dégager les points forts et les points faibles de chacune ces solutions.

Dans ce qui suit, nous présentons une analyse de l'existant, puis nous détaillons la critique de l'existant.

2.4.1. Analyse de l'existant

Au Sénégal, il existe différent type de portefeuille électronique ou mobile money au Sénégal comme : Orange money, Wave, free money, Wizzal, PayDunya etc...

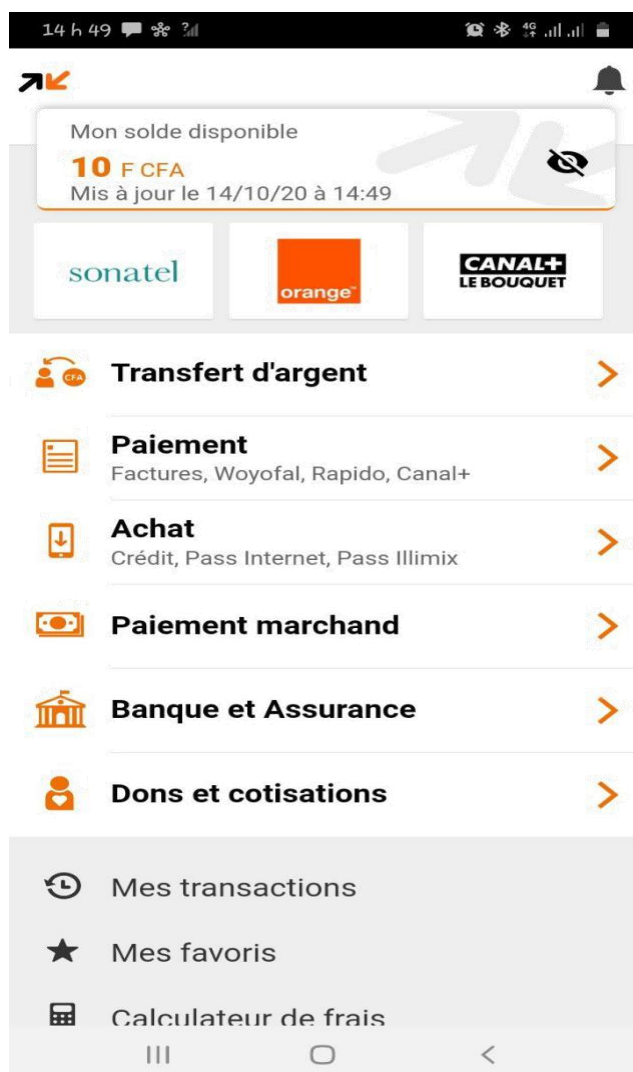


Figure 1 Interface de l'application Orange Money Sur Android

➤ Orange Money :

Orange Money créée en 2008 est le service de transfert d'argent et de paiement mobile du groupe Orange, proposé dans la majorité des pays d'Afrique de l'Ouest où l'opérateur est présent.

➤ Fonctionnement :

Orange Money permet à ses utilisateurs de déposer de l'argent sur un compte associé à leur numéro de mobile orange uniquement, pour ensuite accéder à une gamme de services, notamment transfert d'argent domestique et international, paiement de factures et achat de crédit téléphonique. L'inscription est gratuite et peut être réalisée dans tout point de vente équipée de la signalétique Orange Money.

Le client doit remplir le formulaire d'inscription et présenter une pièce d'identité. Pour alimenter son compte, l'utilisateur peut se rendre dans un point de vente Orange Money pour y déposer d'espèces, retirer de l'argent d'un autre utilisateur par transfert ou recevoir son salaire directement sur son compte (selon pays).

➤ WAVE

Wave est un service de transfert d'argent depuis 2014 dans le marché sénégalais.

Fonctionnement :

Ayant les fonctionnalités de bases d'un portefeuille électronique (dépôt, retrait, transfert d'argent, paiement de facture, achat...), Wave est venue avec une nouvelle approche tel que :

- Pour ouvrir un compte on n'a pas besoin de se déplacer dans une agence pour remplir des formulaires, on a juste besoins d'un smartphone, d'un numéro de téléphone et de la connexion internet
- Faire des dépôt et retrait d'argent sans frais
- Des frais de transfert d'argent qui s'élèvent à 1%
- Faire des achats des crédit pour tous les réseaux mobiles

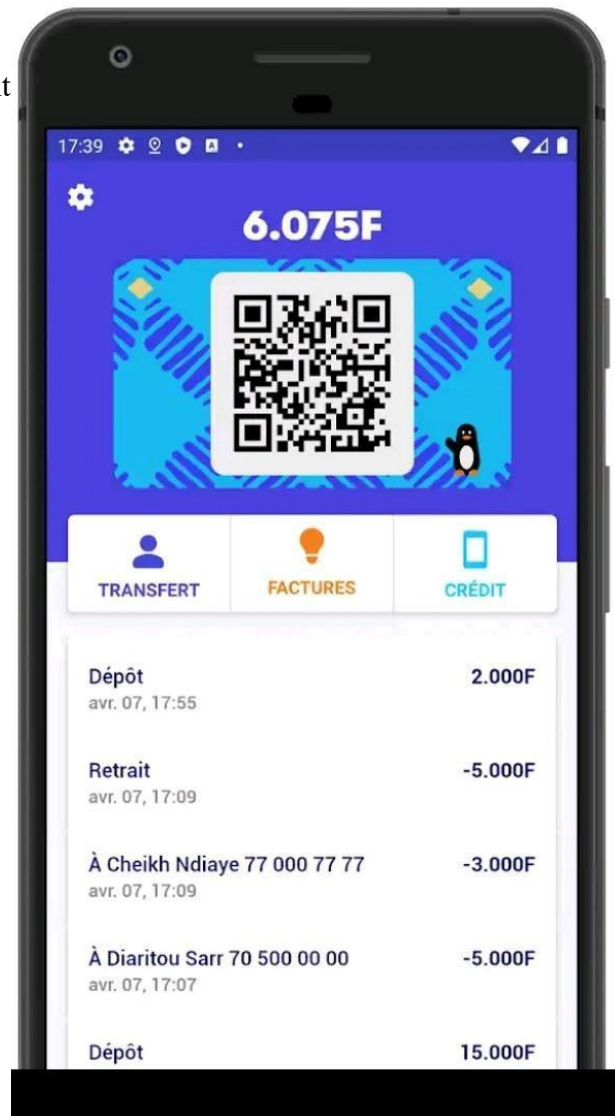


Figure 2: Interface de l'application Wave sur Android

2.4.2. Critique de l'existant

WALLET	AVANTAGES	INCONVIENTS
Orange money	<ul style="list-style-type: none"> ➤ Téléchargement gratuit (non payant) ➤ Fiable et sécurisé ➤ Traçable ➤ Beaucoup de points de distributions ➤ Compatible sur tout sorte de téléphone portable ➤ Application disponible sur Android et Ios 	<ul style="list-style-type: none"> ➤ De rares lenteurs sur le système (des notifications de confirmations qui tardent à venir) ➤ Un réseau qui est parfois temporairement indisponible (lors des événements dont il y'a une forte activité sur le réseau) ➤ Pas de possibilité d'ouvrir par lui-même son propre compte ➤ Des frais de transfert d'argent et de paiement de facture élevé ➤ Pas de possibilité d'annuler une transaction déjà validée sur place (on doit aller jusqu'à l'agence ou remplir un formulaire et l'envoyer dans un espace dédié) ➤ Des difficultés de joindre le service clientèle ➤ Se déplacer dans une agence pour débloquer son compte ➤ Pas de possibilité de

		s'auto- créditer ➤ Pas d'interopérabilité
Wave	➤ Téléchargement gratuit (non payant) ➤ Ouverture de compte ne nécessite pas l'intervention d'un agent ➤ Aucun frais de retrait jusqu'à 10 000 F CFA ➤ Frais de transfert de 1% ➤ Aucun frais lors des paiements de factures ➤ Achat de crédit vers tous les operateurs ➤ Utilisation de réseau pair à pair	➤ Pas option d'acheter de la passe internet ➤ Se déplacer dans une agence pour déplaçonner son compte ➤ Pas de possibilité de s'auto-créditer ➤ Pas d'interopérabilité ➤ Application non securisée

Tableau 1: avantages et inconvénients des wallets (orange money et wave)

2.5. Proposition des différentes solutions

L'étude de l'existant nous a permis de dégager plusieurs anomalies que nous avons détaillées dans la section précédente. Pour faire recours à ces anomalies, nous proposons de concevoir une plateforme de transfert d'argent qui englobe à la fois les fonctionnalités des portefeuilles disponibles dans le marché et des services d'épargne et de crédit.

Dans notre solution nous envisageons que :

- Notre plateforme de transfert d'argent soit basée sur une application mobile, ce qui nous permettra de toucher un large public.
- La mise en place des fonctionnalités de base d'un portefeuille électronique (envoi, retrait d'argent, paiement de facture...).
- L'ajout de l'auto-épargne et du crédit.

- Les concepts d'utilisation de l'application doivent être clairs pour une meilleure expérience d'utilisateur.
- Notre application doit être rapide et responsive. Étant donné le nombre de fonctionnalités importantes, elle devra aussi offrir une simplicité d'utilisation et surtout une ergonomie d'interface.
- Aussi, notre application doit être hautement sécurisée.
- Pour la partie administration, une application web sécurisée est envisagée pour la gestion et une vue d'ensemble des informations des utilisateurs.
- Une application web sera mise à disposition pour les agences de terrain pour effectuer les transactions avec le client (dépôt, retrait, demande d'approvisionnement caisse etc...).

2.5.1. Description détaillée des processus de traitement (fonctionnement)

Dans chaque portefeuille électronique nous avons des fonctionnalités comme suit :

➤ Inscription :

C'est l'ouverture de compte pour avoir accès aux différents services que proposent le portefeuille. Il est nécessaire d'avoir un numéro de téléphone mobile, un email pour recevoir le code d'activation lors de l'inscription. Après l'activation de votre wallet un numéro de compte est généré automatiquement.

➤ Faire un dépôt :

Cette opération consiste à solder son compte pour effectuer les différentes transactions et pour se faire, le numéro de téléphone est obligatoire.

➤ Transférer de l'argent :

Comme son nom l'indique il s'agit d'envoyer de l'argent vers un compte Mobile Money ou vers n'importe quel numéro (un code de retrait est envoyé)

➤ Retirer de l'argent :

Il s'agit de récupérer de l'argent déjà déposé par le client au niveau du compte.

➤ Effectuer des paiements marchands ou paiement de factures :

Il donne une possibilité aux clients de payer leurs achats ou payer leur facture sans se déplacer.

➤ Faire des achats de crédits téléphoniques :

Il permet aux clients de pouvoir acheter du crédit téléphonique avec les fonds du compte client.

2.5.2. Description détaillée des processus de traitement des nouvelles fonctionnalités

➤ **Epargne :**

L'**épargne** correspond à la partie du revenu qui n'est pas encore consommée. Au niveau d'une économie dans son ensemble, l'**épargne** peut être l'<<**Auto-tontine**>> des ménagers, mais aussi des entreprises ou des administrations publiques.

Les motifs de l'**épargne** :

- ✓ Disposer de liquidités : c'est pour une dépense de consommation plus importante dans un futur proche ;
- ✓ Disposer d'une réserve : permet aux ménages d'avoir une marge de sécurité face aux imprévus ;
- ✓ Constituer un patrimoine.

Ainsi lors de l'inscription, le compte d'épargne n'est pas créé, on leur suggère dans créer. Lors de la création, l'utilisateur doit remplir un formulaire dont on lui proposera une fourchette de pourcentages qu'il choisira lors de la transaction de dépôt (Caisse-Client). A chaque dépôt vers le compte courant, une somme est déduite du montant pour solder en même temps le compte épargne.

➤ **Crédit :**

Il peut être défini comme un **crédit** remboursable mensuellement, servant à financer une activité économique, durable et rémunératrice pour son porteur.

Cependant pour être éligible à cette fonctionnalité il faudra respecter certaines normes :

- ✓ Avoir un compte épargne datant au moins de 6 mois
- ✓ Le montant du prêt doit être inférieur ou égal au montant minimal lors de ses dernières transactions effectuées (ou légèrement supérieur pour les utilisateurs très actifs dans le système) au solde compte épargne.
- ✓ Le respect strict des délais de remboursement

2.6. Conclusion

Dans ce premier chapitre, nous avons défini le champ de notre étude suivi d'une étude de l'existant afin de préciser nos objectifs à atteindre.

En effet, l'étude de l'existant nous a permis de préparer une bonne conception pour une amélioration de gestion de porte monnaies électroniques que nous allons ajouter dans les solutions proposées.

Dans le chapitre qui suit nous présenterons les démarches de développement et de conception de notre solution.

Chapitre 3 : MODELISATION CONCEPTUELLE & ORGANISATIONNELLE

3.1. Introduction

La modélisation conceptuelle et organisationnelle constitue une étape importante dans la convergence des notations utilisées dans le domaine de l'analyse de conception objet puisqu'elle représente une synthèse pour notre système.

Dans ce chapitre, nous commençons par le modèle conceptuel détaillé de notre application, ensuite nous exposons le modèle logique des données. Enfin nous clôturons ce chapitre par une conclusion.

3.2. Modélisation Conceptuelle

Le Modèle conceptuel de données est une représentation statique du système d'information. Il a comme objectif de constituer une représentation claire et cohérente des données manipulées dans le système d'information.

Cette partie, sera présentée comme suit : nous commençons par le choix de la méthodologie de conception et justification. Ensuite nous identifions les acteurs et les diagrammes des cas d'utilisation, puis nous présentons le diagramme de classe et enfin les diagrammes de séquence.

3.3. Etude comparative des approches Merise et UML

3.3.1. Merise

Issue de l'analyse systémique, la méthode MERISE (Méthode d'Etude de Réalisation Informatique par Sous-Ensemble) est née dans les années 1970. La méthode Merise présente un ensemble de formalismes et de règles destinées à modéliser de manière indépendante les données et les traitements du système d'information. Ces modèles ne sont qu'une base de réflexion pour le concepteur et un moyen de communication entre les différents acteurs du système d'information. MERISE est une méthode de conception, de développement et de réalisation d'applications informatique. Son but est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements

à effectuer en plusieurs modèles conceptuels, organisationnels et physiques. La méthode Merise comprend 3 niveaux :

- Le niveau conceptuel ;
- Le niveau organisationnel ;
- Le niveau opérationnel.

Mais, malgré ces avantages, on reproche à MERISE d'utiliser un formalisme jugé complexe. En plus de cela, cette méthode est essentiellement franco-française et n'a pas eu beaucoup de succès à l'étranger où les méthodes anglo-saxonnes sont plus présentes.

3.3.2. UML

UML (Unified Modeling Language) est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines de l'application et aux processus, son caractère polyvalent et sa souplesse ont fait lui un langage universel. En plus UML est essentiellement un support de communication, qui facilite la représentation et la compréhension de solution objet. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation des solutions. L'aspect de sa notation, limite l'ambiguïté et les incompréhensions.

UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues.

Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues :

- La vue statique, permettant de représenter le système physiquement. Elle est essentiellement de :
 - ✓ Diagrammes de classes : représentent des collections d'éléments de modélisation statiques (classes, paquets...), qui montrent la structure d'un modèle.
 - ✓ Diagrammes d'objets : ces diagrammes montrent des objets (instances classes dans un état particulier) et des liens (relations sémantiques) entre objets.
 - ✓ Diagrammes de cas d'utilisation : identifient les utilisateurs du système (acteurs) et leurs interactions avec le système.
- La vue dynamique, montrant le fonctionnement du système. Elle est essentiellement de :

- ✓ Diagrammes de séquence : permettent de représenter des collaborations eu objets selon un point de vue temporel, on y met l'accent sur la chronologie (envois de messages).

La conception de notre interface a été élaborée en suivant la démarche suivante :

- ✓ L'élaboration des diagrammes de cas d'utilisation. Cette étape a été réalisée suite à la spécification fonctionnelle de l'application.
- ✓ Recensement des classes clients et élaboration du diagramme des classes.
- ✓ Dresser le diagramme de séquences pour mettre en évidence interactions entre les différents objets du système.

3.3.3. Choix et justification

Pour mener à bien notre projet, nous devons choisir une méthode qui nous permet de modéliser notre système en le structurant en objets. Ainsi nous allons opter pour la modélisation avec UML. Comme nous l'avons dit UML est un langage de modélisation unifié qui est beaucoup plus orienté objet que les autres méthodes énoncées précédemment. La maintenance est également beaucoup plus facile avec UML car ici lors de la modélisation on ne sépare pas les données des traitements. Avec la modélisation UML, au lieu d'avoir une structure de données manipulée par des fonctions, nous avons une entité autonome composée de données et des traitements associés à ces données.

3.4. Diagramme des cas d'utilisations

Les cas d'utilisation décrivent un ensemble d'actions réalisées par le système, en réponse à une action d'un acteur.

3.4.1. Identification des acteurs

L'agence, le client et l'administrateur sont les acteurs qui interagissent avec notre système.

- Responsable agence : elle est une filiale qui relie le client et le système.
- Caissier : il est l'acteur qui utilise le système pour le client.
- Le client : il est l'acteur qui paie pour les services rendus

- Administrateur : c'est le responsable de l'administration de l'application.

3.4.2. Les diagrammes des cas d'utilisation

3.4.2.1. Cas General du système

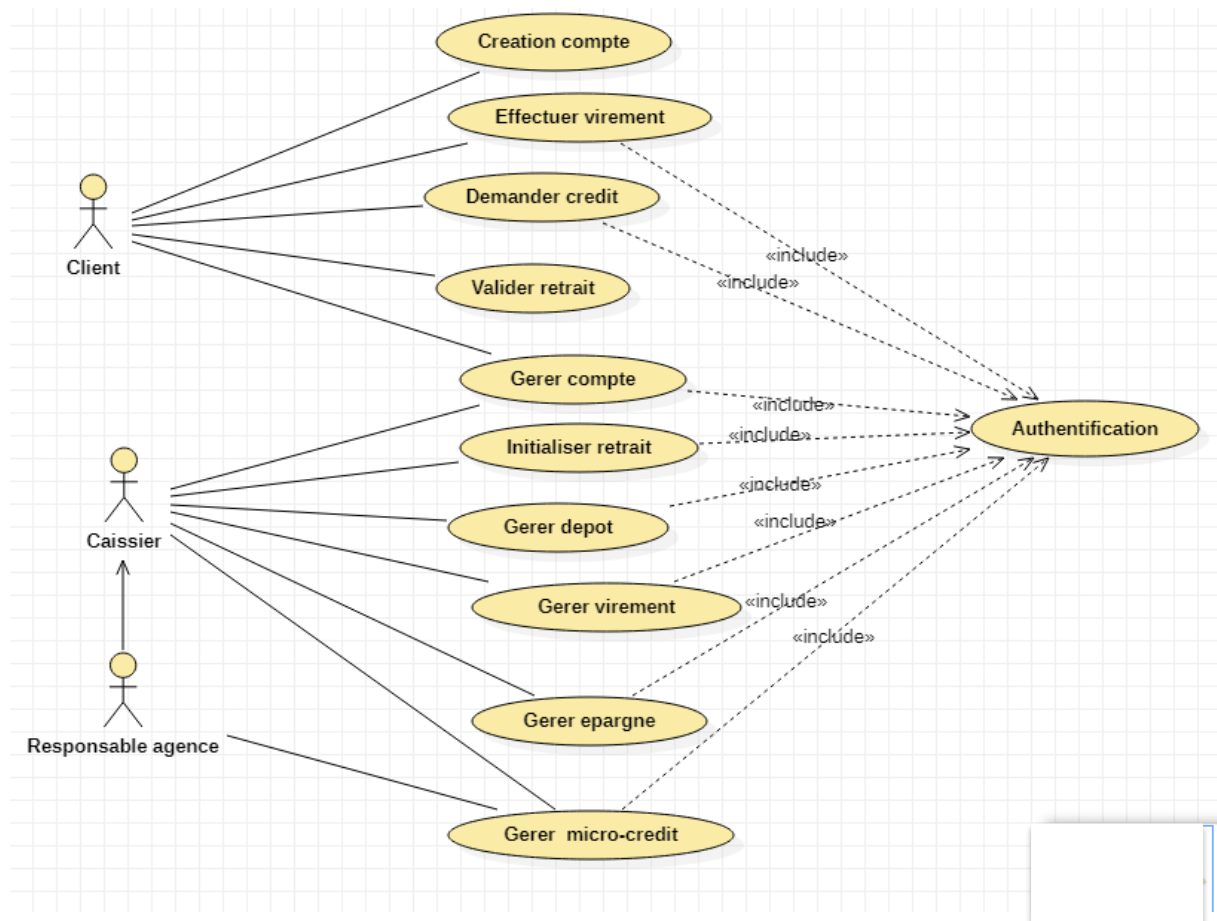


Figure 3: Diagramme de cas d'utilisation général du système

3.4.2.2. Cas création de compte

Description textuelle création compte

Création Compte	
Résumé : Le client crée son compte wallet	
Acteurs : Clients	
Événement déclencheur : Le client veut avoir un compte wallet	
Précondition : Client a déjà installer l'application	
Post-Condition : Le Compte a été enregistrée en base de données, de même que les informations du client sont bien à jour (wallet).	

Tableau 2: Description textuelle création compte

Scénario nominal		
Étapes	Actions acteur	Réponses système
1	Le client entre son numéro de téléphone	
2		Le système vérifie l'existence des informations du client sur le wallet
3	Le client saisit les informations par rapport à la création de compte, et au besoin du système.	
4		Le système vérifie la conformité des informations
5		Le système enregistre la création du compte et met à jour les données sur la base de données
6		<i>Fin</i>

Tableau 3: Scénario nominal création compte

Scénario alternatifs		
Etapes	Actions alternatif acteur	Réponses alternatif système
2.a		Les informations existent
2.a.1		Le système renvoie les informations du client
2.b		Les informations n'existent pas
2.b.1		Le système permet une saisie des informations du client
2.b.2	Le client saisit les informations	
4.a		Informations conformes
4.a.1		Le système permet d'enregistrer la création de compte
4.a.2	Le système enregistre la création de compte	
4.b		Informations non conformes
4.b.1		Le système affiche un message des données non conformes

Tableau 4: Scénario alternatifs création compte

3.4.2.3. Création compte Epargne

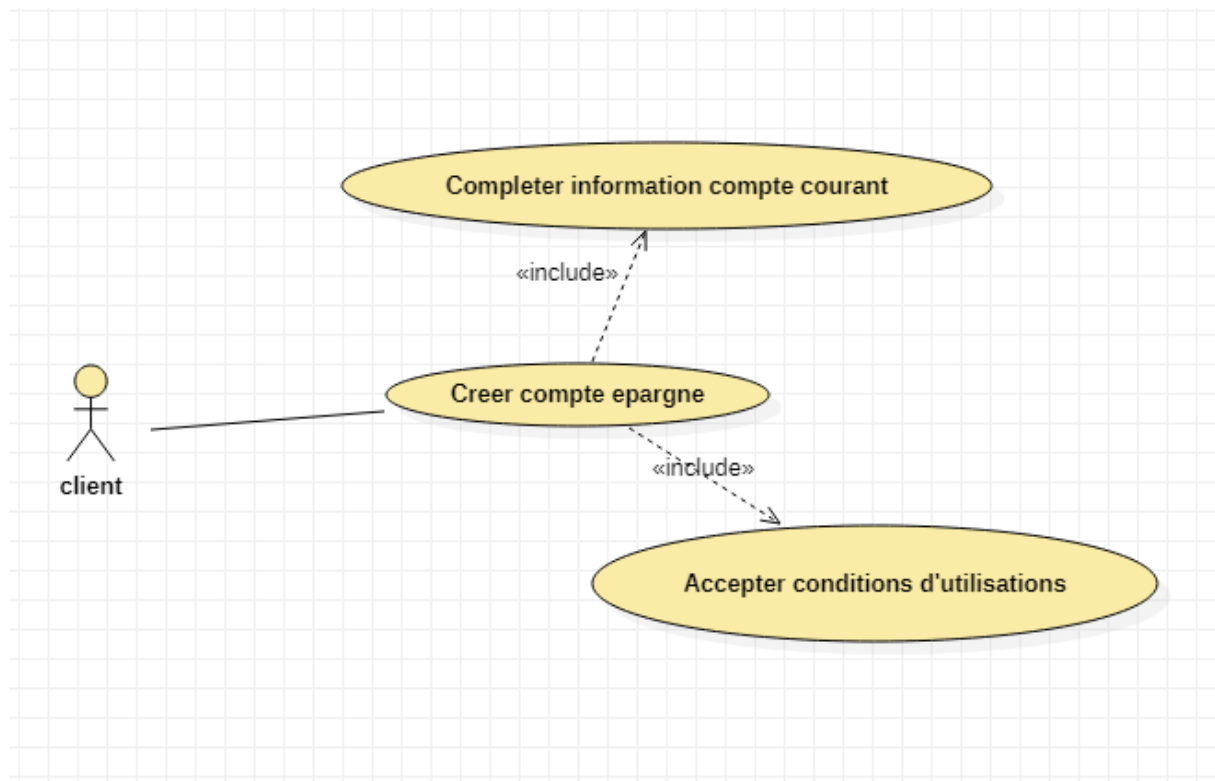


Figure 4: Diagramme de cas d'utilisation création compte Épargne

Description textuelle création compte épargne :

Création Compte Epargne
Résumé : Permet au client de créer un compte d'épargne, de verser de l'argent sur son compte et ce dernier est bloqué pour une durée déterminée
Acteurs : Clients
Événement déclencheur : Le client veut épargner
Pré-Condition : Client s'est déjà authentifié
Post-Condition : Le Compte épargne a été enregistrée en base de données, de même que les informations du client sont bien à jour (wallet).

Tableau 5: Description textuelle création compte épargne

Scénario nominal		
Etapes	Actions acteur	Réponses système
1	Le client entre son numéro de téléphone	
2		Le système vérifie l'existence des informations du client sur le wallet
3	Le client saisit les informations par rapport à la création de compte, et au besoin du système.	
4		Le système vérifie la conformité des informations
5		Le système enregistre la création du compte épargne et met à jour les données sur la base de données
6		<i>Fin</i>

Tableau 6: Scénario nominal création compte épargne

Scénario alternatifs		
Etapes	Actions alternatif acteur	Réponses alternatif système
2. a		Les informations existent
2. a.1		Le système renvoie les informations du client
2. b		Les informations n'existent pas
2. b.1		Le système permet une saisie des informations du client
2. b.2	Le client saisit les informations	
4. a		Informations conformes
4. a.1		Le système permet d'enregistrer la création de compte
4. a.2	Le système enregistre la création de compte	
4. b		Informations non conformes
4. b.1		Le système affiche un message des données non conformes

Tableau 7: Scénario alternatifs création compte épargne

3.4.2.4. Gestion dépôt

Gestion dépôt	
Résumé :	Permet au client de verser de l'argent sur son compte
Acteurs :	Client, Caissier, Responsable Agence, Administrateur
<u>Événement déclencheur :</u>	Le client ou Caissier veut verser de l'argent
<u>Pré-condition :</u>	Caissier s'est déjà authentifié
Post-condition :	Le Compte a été créditer ;

Tableau 8: Description textuelle gestion dépôt

Description textuelle gestion dépôt

Scénario nominal		
Etapes	Actions acteur	Réponses système
1	Le caissier renseigne les informations du client	
2		Le système vérifie l'existence des informations du client sur le wallet
3	Le caissier saisit le montant d déposer.	
4		Le système vérifie la conformité des informations
5		Le système enregistre le dépôt de l'argent dans le compte et met à jour les données sur la base de données
6		<i>Fin</i>

Tableau 9: Scénario nominal gestion dépôt

Scénario alternatifs		
Etapes	Actions alternatif acteur	Réponses alternatif système
2.a		Les informations existes
2.a.1		Le système renvoie les informations du client
2.b	Le caissier saisit le montant d déposer.	
4.a		Informations conformes
4.a.1		Le système permet de solder le compte
4.a.2	Le système enregistre le dépôt	
4.b		Informations non conformes
4.b.1		Le système affiche un message des données non conformes

Tableau 10: Scénario alternatifs gestion dépôt

3.4.2.5. Gestion demande de crédit

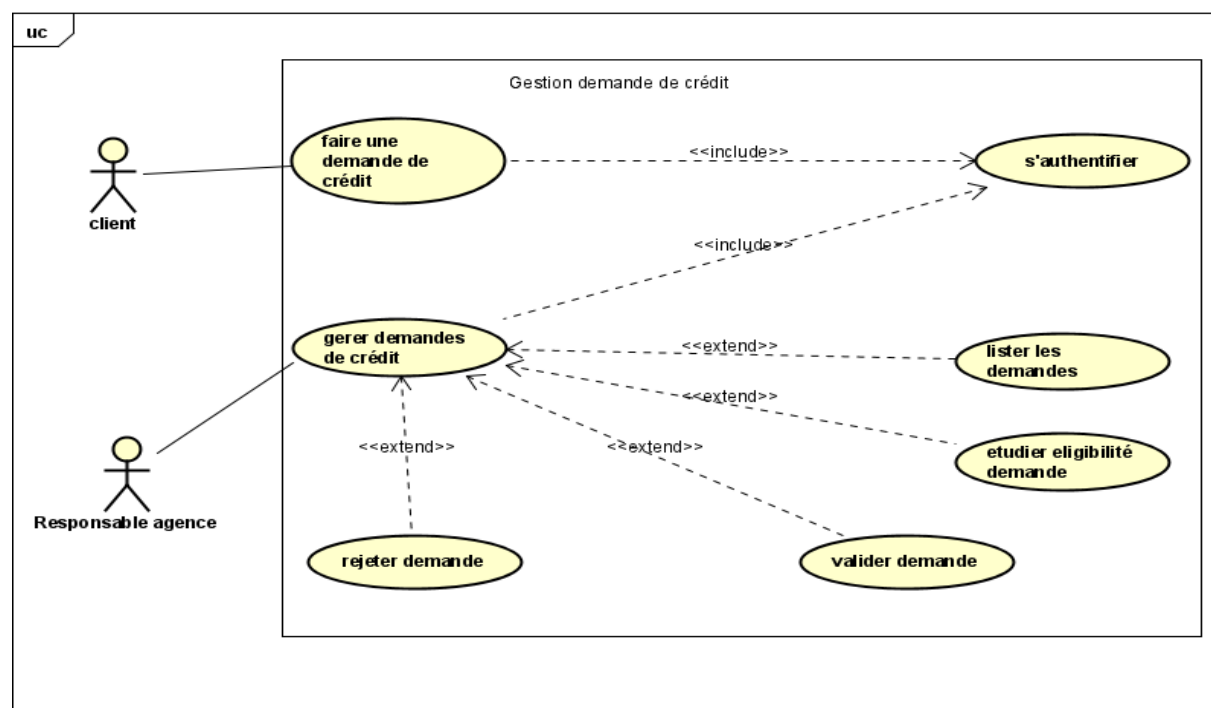


Figure 5: Diagramme de cas d'utilisation Gestion demande de crédit

Gestion demande de crédit
Résumé : Permet au client d’emprunter de l’argent au niveau du système.
Acteurs : Caissier et Responsable Agence
<u>Événement déclencheur :</u> Le client veut emprunter de l’argent
<u>Pré-condition :</u> Caissier ou Responsable Agence , s’est déjà authentifié
Post-condition : Le montant demandé a été enregistrée en base de données, de même que le compte client est bien à jour (solde).

Tableau 11: Description textuelle Gestion de crédit

Scénario nominal		
Etapes	Actions acteur	Réponses système
1	Le caissier entre les informations du client	
2		Le système vérifie l’existence de compte épargne du client sur le wallet
3	Le caissier saisit le montant à demander	
4		Le système vérifie la conformité des informations (date de création du compte épargne et le nombre de transaction effectuées)
5		Le système enregistre la demande.
6	Le responsable de l’agence valide la demande d’emprunt	Le système enregistre la demande et solde le compte et met à jour les données sur la base de données
7		Fin

Tableau 12: Scénario nominal gestion crédit

Scénario alternatifs		
Etapes	Actions alternatif acteur	Réponses alternatif système
2.a		Les numéros existent
2.a.1		Le système renvoie les informations du client
2.b	Le caissier saisit le montant à demander	
4.a		Informations conformes
4.a.1		Le système permet de solder le compte
4.a.2		Le système enregistre la demande d'emprunt
4.b		Informations non conformes
4.b.1		Le système affiche un message des données non conformes

Tableau 13: Scénario alternatifs gestion crédit

3.5. Diagrammes d'activités par scénario des cas d'utilisation

3.5.1. Cas gestion dépôt(épargne)

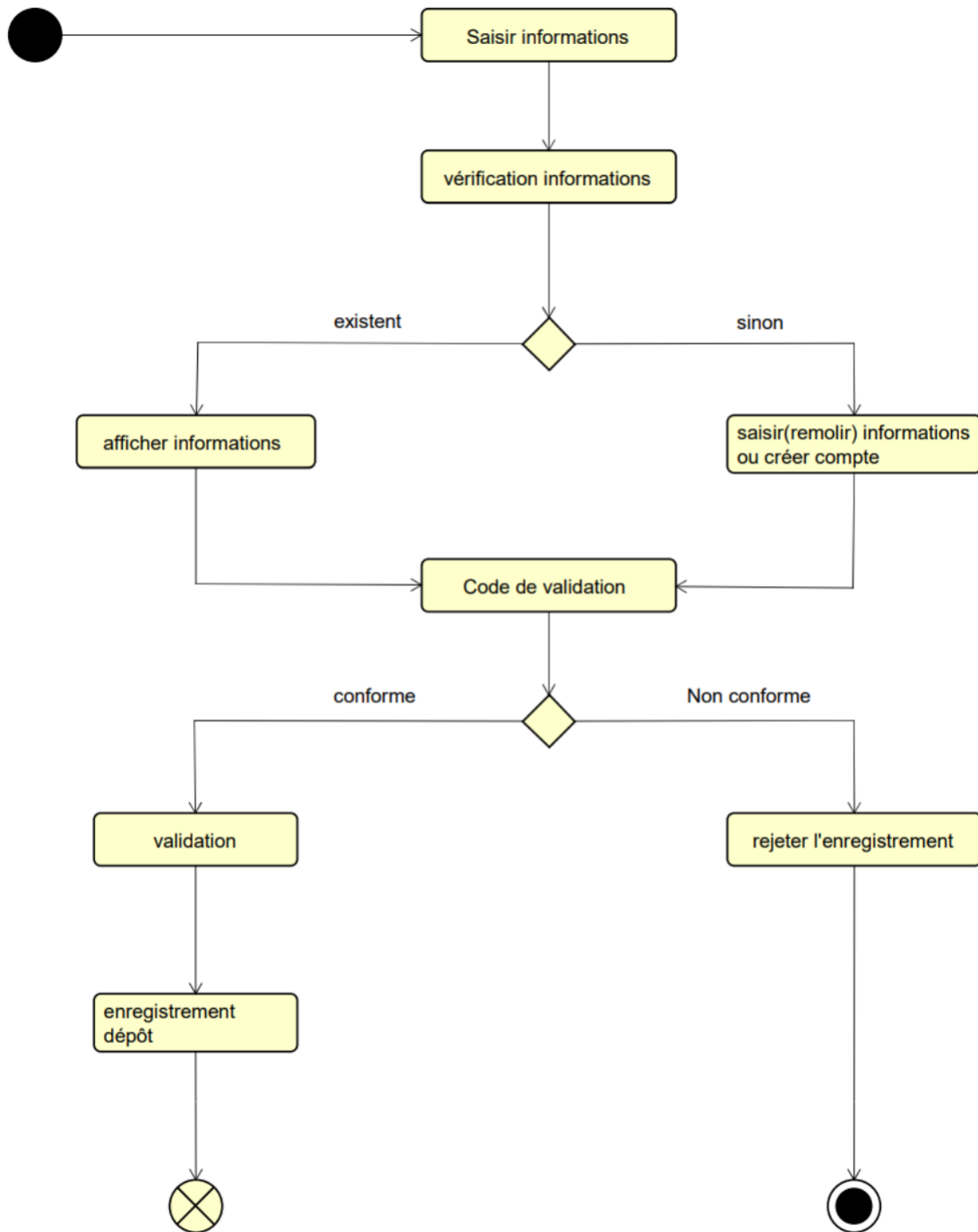


Figure 6: Diagramme d'activité du cas Gestion dépôt (épargne)

3.5.2. Cas gestion crédit (d'emprunt)

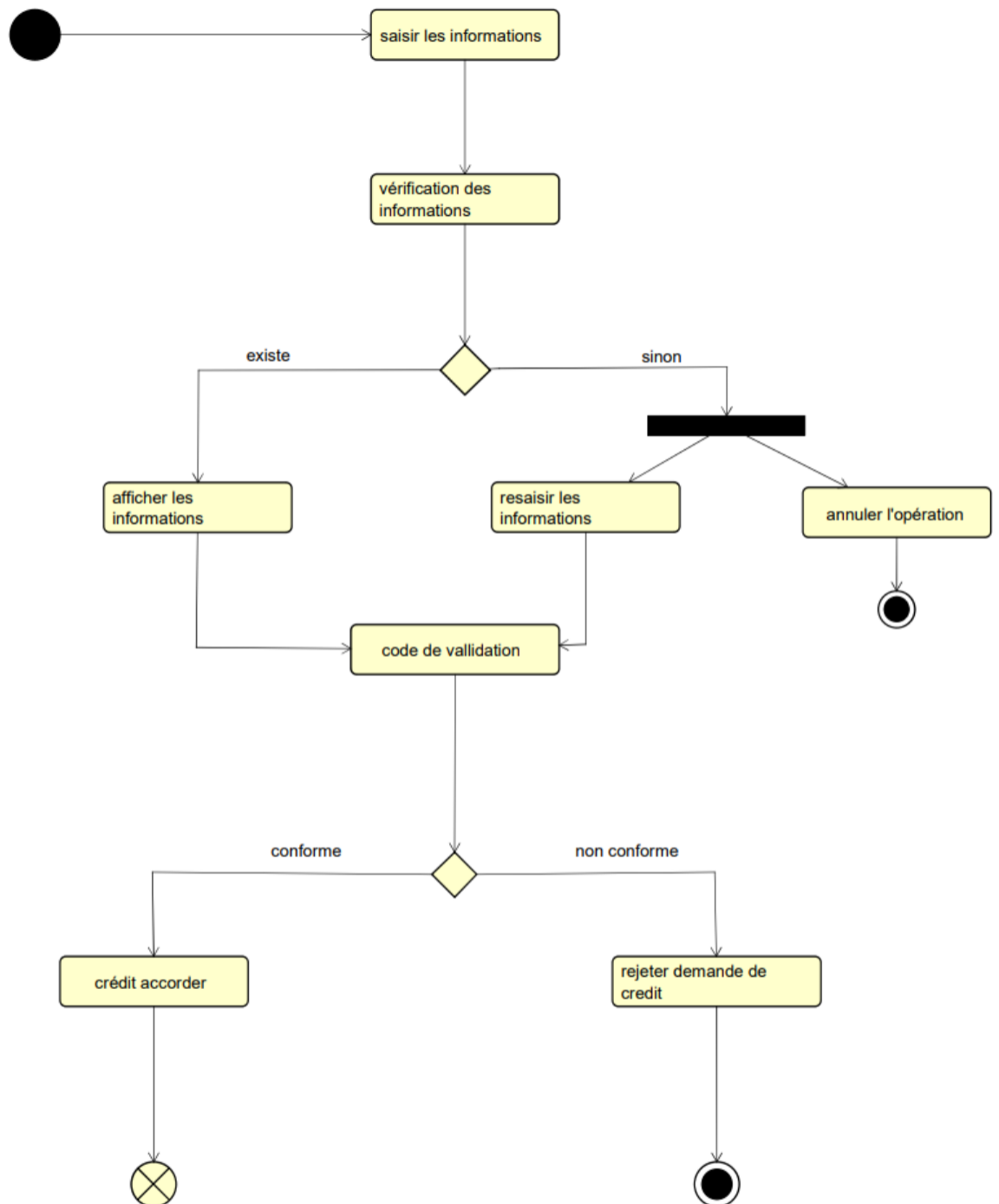


Figure 7: Diagramme d'activité du cas Gestion credit (d'emprunt))

3.6. Modélisation Conceptuelle des données :

La modélisation conceptuelle des données permet de dégager l'ensemble des données manipulées en vue d'élaborer le diagramme de classes. En effet, ce dernier donne une vue statique du système. Il décrit les types et les objets du système.

Il s'agit donc d'une représentation des données du champ de l'étude ainsi que le lien sémantique entre ces données, facilement compréhensible, permettant de décrire le système d'information à l'aide des concepts proposés par le modèle UML.

Fiche descriptive des classes :

Classes	Descriptions
User	Répertorie l'ensemble des utilisateurs
Adminitrateur	Répertorie l'ensemble des administrateurs
Responsable	Répertorie l'ensemble des responsables
Caissier	Répertorie l'ensemble des caissiers
Client	Répertorie l'ensemble des clients
Compte	Répertorie l'ensemble des comptes
Type de compte	Recense l'ensemble des types de comptes
Agence	Répertorie l'ensemble des comptes
Operation	Recense l'ensemble des operations
Type operation	Recense l'ensemble des types d'operations
Credit	Recense l'ensemble des types credit

Tableau 14: Fiche descriptive des classes

3.7. Diagramme de classes

La figure ci-dessous récapitule les tableaux précédents dans un diagramme de classes qui Contient toutes les informations telles que les classes, les méthodes, les associations et les propriétés.

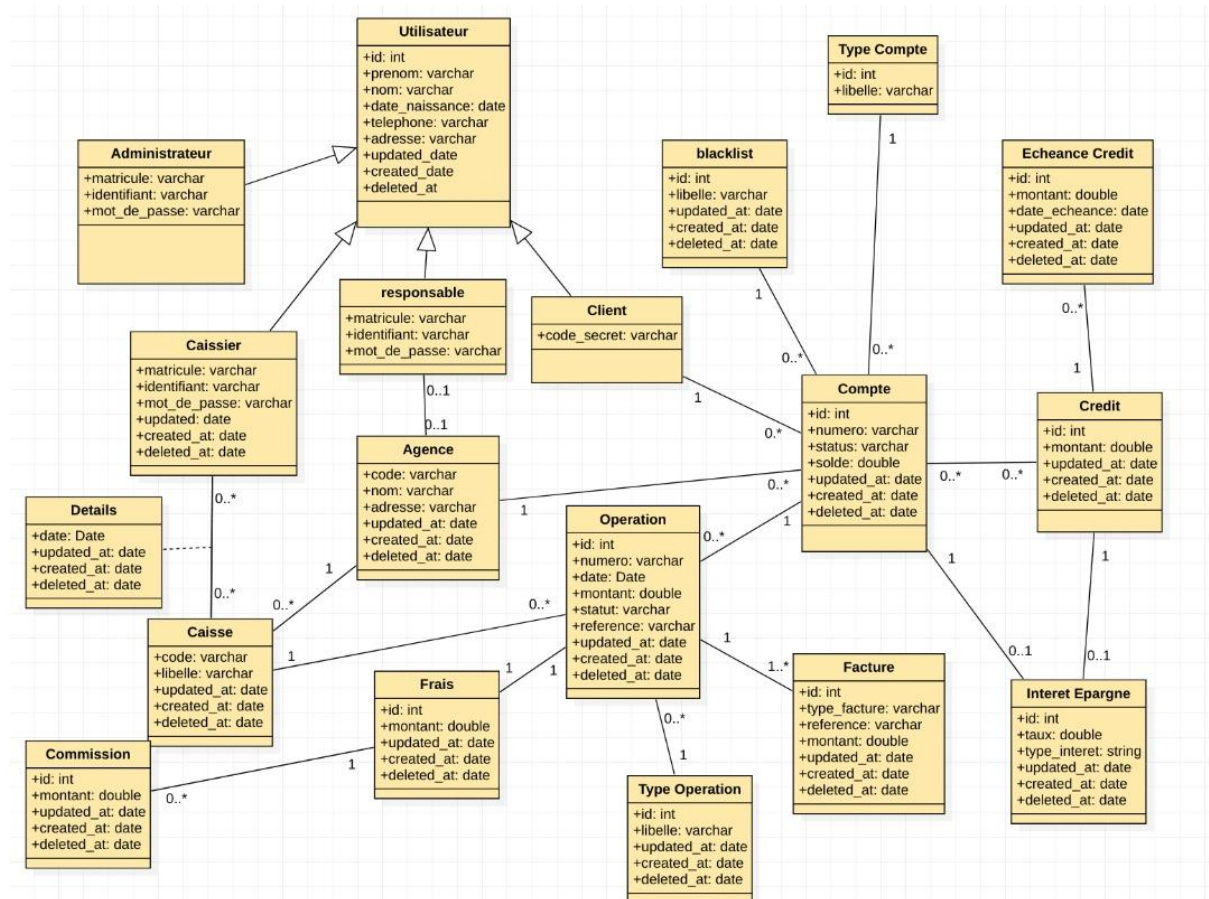


Figure 8: Diagramme de classe

Conclusion

Dans ce deuxième chapitre, nous avons réalisé la modélisation conceptuelle et organisationnelle de notre solution. En effet cette partie nous permet de mieux cerner notre projet avec l'aide des différents diagrammes.

Dans le chapitre qui suit nous présenterons l'architecture et la réalisation de la solution.

Chapitre 4 : ARCHITECTURE ET REALISATION DE LA SOLUTION

4.1. Introduction

Après avoir élaboré la conception de notre application, nous abordons dans ce chapitre le dernier volet de ce mémoire, qui a pour objectif d'exposer la phase de réalisation et l'architecture de la solution proposée.

La phase de réalisation est considérée comme étant la concrétisation finale de toute la méthode de conception.

Nous menons tout d'abord une étude technique où nous décrivons les ressources logicielles utilisées dans le développement de notre projet. Nous présentons en premier lieu l'architecture de notre projet, où nous spécifions l'environnement matériel et logiciel qu'on a utilisé pour réaliser notre application puis nous détaillons l'architecture, aussi nous présentons quelques interfaces réalisées pour illustrer le fonctionnement de quelques activités du système.

4.2. Choix des langages de développement et de SGBD

4.2.1. Frontend


4.2.1.1. Web




HTML : Le langage HTML (HyperText Markup Language) est le langage De

balisage standard pour les documents conçus pour être affichés dans un navigateur Web. Les navigateurs Web reçoivent des documents HTML à partir d'un serveur Web ou d'un stockage local et les convertissent en pages Web multimédias. HTML décrit la structure d'une page Web de manière sémantique et inclut à l'origine des indices pour l'apparence du document. Les éléments HTML sont les blocs de construction des pages HTML. Avec les constructions HTML, des images et d'autres objets tels que des formulaires interactifs peuvent être incorporés dans la page rendue. HTML fournit un moyen de créer des documents structurés en indiquant une sémantique structurelle pour un texte tel que des en-têtes, des paragraphes,

des listes, des liens, des guillemets et d'autres éléments. Les éléments HTML sont délimités par des balises, écrites à l'aide de chevrons. Des balises qui introduisent directement le contenu dans la page. Et d'autres balises qui entourent et fournissent des informations sur la page et peuvent inclure d'autres balises en tant que sous-éléments. Les navigateurs n'affichent pas les balises HTML, mais les utilisent pour interpréter le contenu de la page.

-  **CSS** Les feuilles de style en cascade (CSS) sont un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel que HTML. CSS est conçu pour permettre la séparation de la présentation et du contenu, y compris la mise en page, les couleurs et les polices. Cette séparation peut améliorer l'accessibilité au contenu et offrir plus de flexibilité et de contrôle dans la spécification des caractéristiques de présentation, permettant à plusieurs pages Web de partager le formatage en spécifiant le fichier CSS pertinent dans un fichier.

Css distinct et de réduire la complexité et la répétition du contenu structurel. La séparation du formatage et du contenu permet également de présenter la même page de balisage dans différents styles pour différentes méthodes de rendu, telles que l'affichage à l'écran, l'impression, la voix (via un navigateur ou un lecteur d'écran à base vocale) et en braille. CSS a également des règles pour le formatage alternatif si le contenu est accédé sur un appareil mobile.

 **JavaScript**, souvent abrégé en JS, est un langage de programmation orienté objet interprété de haut niveau, conforme à la spécification ECMAScript. JavaScript possède une syntaxe pour les accolades, le typage dynamique, l'orientation objet et les fonctions de première classe. Outre HTML et CSS, JavaScript est l'une des technologies de base du World Wide Web. JavaScript permet de réaliser des pages Web interactives et constitue un des éléments essentiels des applications Web. La grande majorité pour ne pas dire tous les sites Web l'utilisent et les principaux navigateurs Web disposent d'un moteur JavaScript dédié pour l'exécuter. En tant que langage multiparadigme, JavaScript prend en charge les styles de programmation événementiels, fonctionnels et impératifs (y compris ceux orientés objet et basés sur des prototypes). Il possède des API pour travailler avec du texte, des tableaux, des dates, des expressions régulières et le DOM.

Nb : Les termes Vanilla JavaScript et Vanilla JS font référence à JavaScript non étendu par des

Framework ou des bibliothèques supplémentaires. Les scripts écrits en Vanilla JS sont du code

JavaScript simple.



Bootstrap est un Framework Web qui vise à simplifier le développement de pages Web informatives (par opposition aux applications Web). Son objectif principal consiste à appliquer les choix de couleur, de taille, de police et de présentation de Bootstrap à ce projet. En tant que tel, le principal facteur est de savoir si les développeurs en charge trouvent ces choix à leur goût. Une fois ajouté à un projet, Bootstrap fournit des définitions de style de base pour tous les éléments HTML. Le résultat est une apparence uniforme pour la prose, les tableaux et les éléments de formulaire dans les navigateurs Web. En outre, les développeurs peuvent tirer parti des classes CSS définies dans Bootstrap pour personnaliser davantage l'apparence de leur contenu.

Le projet aujourd'hui est de concevoir une plateforme dynamique et modulaire capable d'être affiché de manière optimale sur tous les appareils ce qui est primordial considérant le nombre d'utilisateurs sur le continent qui accèdent à internet via leur smartphone. C'est là où Bootstrap intervient grâce à sa popularité, aux nombres de Template disponible et à la foule de fonctionnalités offert par le Framework Bootstrap se présente comme étant la meilleure solution pour construire le squelette de notre site web et le rendre responsive avec le moins d'effort possible.



Développé par Google, **Angular** est un Framework open source écrit en JavaScript qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet

donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.

Il y a plusieurs Framework JavaScript très populaires aujourd'hui : Angular, React, Ember, Vue... Angular présente également un niveau de difficulté légèrement supérieur, car on utilise le TypeScript plutôt que JavaScript pur ou le mélange JS/HTML de React. Ainsi, quels sont donc les avantages d'Angular ?

- Angular est géré par Google — il y a donc peu de chances qu'il disparaisse, et l'équipe de développement du Framework est excellente.
- Le TypeScript : ce langage permet un développement beaucoup plus stable, rapide et facile.
- Le Framework Ionic : il permet de développer des applications mobiles multi-plateformes à partir d'une seule base de code et il est basé sur Angular.

4.2.1.2. Mobile



Java est le langage natif dédié à l'os Android. Dans le cas de notre étude pour la première version, nous avons choisi de développer notre application mobile uniquement sur Android avec les langages java/kotlin et l'outil développement android studio pour une application native.



Kotlin est le nouveau langage de programmation Open Source supporté par

Google pour les applications Android (mais aussi pour le back-end Java et les moteurs JavaScript des navigateurs web). Il a pour but d'améliorer la productivité des développeurs, tout en restant compatible avec le code existant. Il est également possible de le compiler en natif.

4.2.2. Backend



Notre choix du langage de programmation s'est porté sur le **langage JAVA** du fait qu'il soit un langage hybride, à la fois compilé et interprété. On dit qu'il est semi-compilé. Pour simplifier, disons qu'un programme Java est compilé dans un langage qui devra ensuite être interprété.

Mais aussi de ses avantages qu'il présente :

- C'est portable (donc exécutable sur n'importe quel système, à condition d'avoir installé une JVM).
- C'est populaire, donc on peut facilement trouver du monde pour coder/aider/reprendre du Java.
- Il y'a souvent des API qui correspondent à peu près à notre besoin. Pour le développement de cette application, nous optons l'environnement IntelliJ IDEA du fait qu'il ait une meilleure gestion des raccourcis pour gagner en temps.
- On dit que Java est plus sécurisé que tout autre langage de programmation. Sa sécurité est obtenue à partir d'un large ensemble d'API, d'outils et d'implémentations d'algorithmes, de mécanismes et de protocoles de sécurité couramment utilisés. Les API Java incluent la cryptographie, l'infrastructure à clé publique, la communication sécurisée, l'authentification et le contrôle d'accès. JDK a été développé à l'origine avec un fort accent sur la sécurité permettant de créer des applications sécurisées dès le début.

Pour une bonne mise en place de notre solution nous utiliserons le Grails qui est un framework basé sur la plateforme Java et le langage de programmation Groovy. Grails lui-même est basé en interne sur certains composants logiciels tels que Java Enterprise Edition (Java EE). Le Framework Spring pour la gestion interne des objets et Hibernate pour l'accès à la base de données. Le langage à typage dynamique Groovy est utilisé comme élément de liaison dans Grails-Framework.

Le Framework Grails est une version allégée du langage de programmation Java. Comme l'espace mémoire alloué dans la carte est de taille réduite, il y a certaines caractéristiques qui ne sont pas supportées par grails telles :

- Les types simples : Int, float, double... à l'exception du byte, short et Boolean ;
- Les chaînes de caractères ;
- Manipulation des tableaux multidimensionnels ;
- Multithreading ;

4.2.3. Le Système de gestion des bases de données (SGBD)

4.2.3.1. Définition et rôles

Une base de données est composée de données stockées dans des mémoires de masse sous une forme structurée, et accessibles par des applications différentes et des utilisateurs différents. Une base de données doit pouvoir être utilisée par plusieurs utilisateurs en même temps. Un SGBD (Système de Gestion de Base de Données) est un ensemble de logiciels chargés d'assurer les fonctions minimales suivantes :

- Le maintien de la cohérence des données entre elles
- Le contrôle d'intégrité des données
- Les opérations classiques sur les données (consultation, insertion, modification, suppression)
- Les autorisations d'accès aux données

A cette étape de l'analyse comparative des solutions, il est question de faire une étude sur les différents serveurs de base de données pour la gestion des données du système. Derrière toute application informatique appelée à manipuler des informations, il faut un système dédié à la gestion des différentes bases de données. Pour guider objectivement notre choix, nous orientons cette étude sur quelques SGBD qui sont : MySQL, SQL Server, PostgreSQL et Oracle.

4.2.3.2. Choix et justification

Pour la gestion de la base de données, nous retenons « MySQL » qui est un SGBDR (Système de Base de Données Relationnelles) car il est assez robuste et offre des performances techniques et intéressantes. Il est efficace dans la gestion des bases de données et offre un bon niveau de sécurisation des données. En effet, une bonne configuration du serveur de base de données garantit efficacement la sécurité des données du système.

SQL (Structured Query Language) est un langage de manipulation de bases de données il permet trois types de manipulations :

- La maintenance des tables : création, suppression, modification de la structure des tables

- La manipulation des données : sélection, modification, suppression d'enregistrements.
- La gestion des droits d'accès aux tables : contrôle des données, droits d'accès, validation des modifications.

4.2.4. Architecture applicative :

4.2.4.1. Présentation

Une architecture désigne la structure générale inhérente à un système informatique, l'organisation des différents éléments du système (logiciels et/ou matériels) et des relations entre les éléments. Cette structure fait suite à un ensemble de décisions stratégiques prises durant la conception de tout ou partie du système informatique. Une architecture applicative assure une meilleure gestion des données et des composants qui doivent interagir (serveur de base de données, poste client, serveur web, serveur d'application...).

Les différents types d'architectures sont : architecture 1-Tiers, architecture 2Tiers, architecture 3-Tiers, architecture n-Tiers.

➤ Architecture 1-tiers :

Dans cette architecture, les trois couches s'exécutent sur le même serveur. Elle constitue la première variante qui consiste en un client gérant uniquement la couche présentation et un serveur réalisant l'ensemble des traitements applicatifs. Malgré sa simplicité d'administration et sa centralisation des données, cette architecture présente quelques limites :

- ✓ Plusieurs utilisateurs se partagent des fichiers de données stockés sur un serveur commun ; la gestion des conflits d'accès aux données doit être prise en charge par chaque programme de façon indépendante, ce qui n'est pas toujours évident.
- ✓ Lors de l'exécution d'une requête, l'intégrité des données nécessaires doit transiter sur le réseau et on arrive à saturer ce dernier.
- ✓ La cohabitation de plusieurs moteurs de base de données indépendants manipulant les mêmes données peut devenir instable. Ces conflits peuvent affecter l'intégrité des données.
- ✓ Il est difficile d'assurer la confidentialité des données.

Architecture 1-tiers (client autonome)

- L'application est sur le client
- Les données sont sur un serveur (éventuellement distant)
- Exemple :

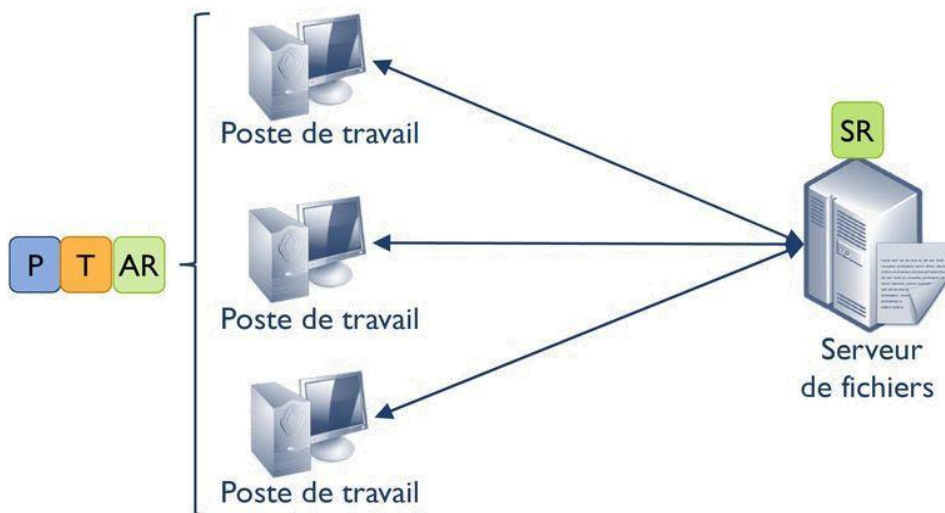


Figure 9: Représentation architecture 1-tier

➤ Architecture 2-tiers :

L'architecture à deux niveaux (aussi appelé architecture 2-tiers, tiers signifiant étages en anglais) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service. Elle permet l'utilisation d'une interface utilisateur riche et l'appropriation des applications par l'utilisateur mais présente également quelques limites :

- ✓ La relation étroite qui existe entre le programme client et l'organisation de la partie serveur complique les évolutions de cette dernière.
- ✓ On ne peut pas soulager la charge du poste client, qui supporte la grande majorité des traitements applicatifs, une modification de l'application ou de la structure de la base de données nécessite un redéploiement sur les postes clients. Cet aspect fait que le client dans cette variante est dit « lourd »
- ✓ La difficulté d'administrer les postes clients
- ✓ Le déploiement est couteux et très difficile à réaliser à grande échelle.

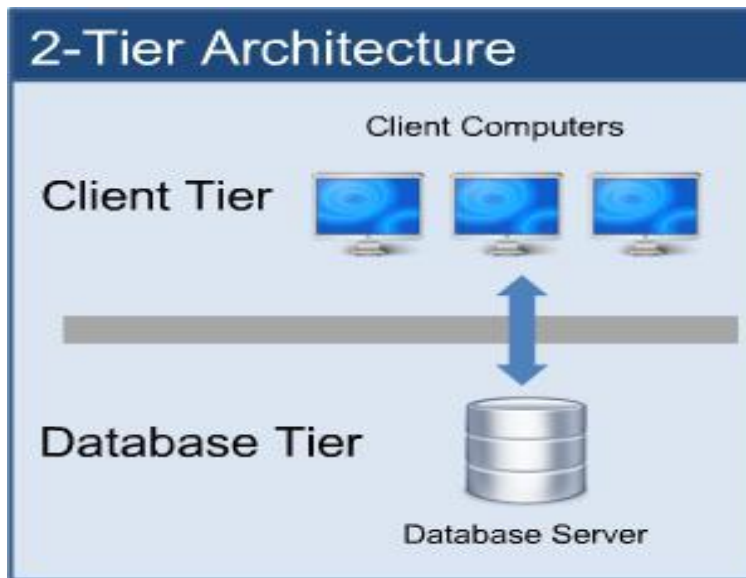


Figure 10: Représentation architecture 2-tiers

➤ Architecture 3-tiers :

Dans l'architecture à 3 niveaux (appelé architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

- ✓ Le client : le demandeur de ressources
- ✓ Le serveur d'application (appelé aussi middleware) : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur
- ✓ Le serveur secondaire (généralement un serveur de base de données), fournissant un service au premier serveur.

Cette architecture présente de nombreux avantages :

- ✓ Les applications serveurs sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans sa tâche.
- ✓ Le poste client ne supporte plus l'ensemble des traitements, il est moins sollicité et peut être moins évolué donc moins coûteux.
- ✓ La fiabilité et les performances de certains traitements se trouvent améliorées par leur centralisation, il est relativement simple de faire face à une forte montée en charge en renforçant le service applicatif.
- ✓ Une grande sécurité compte tenu que l'on peut définir la sécurité pour chaque service.

- ✓ De bonnes performances dues au fait que les tâches sont partagées entre plusieurs serveurs.

Mais également des limites :

- ✓ Le serveur se trouve souvent fortement sollicité et il est difficile de répartir la charge entre le client et le serveur.
- ✓ Les solutions mises en œuvre sont relativement complexes à maintenir et la gestion des sessions compliquée
- ✓ Les contraintes semblent inversées par rapport à celles rencontrées avec les architectures deux tiers : le client est soulagé, mais le serveur est fortement sollicité.

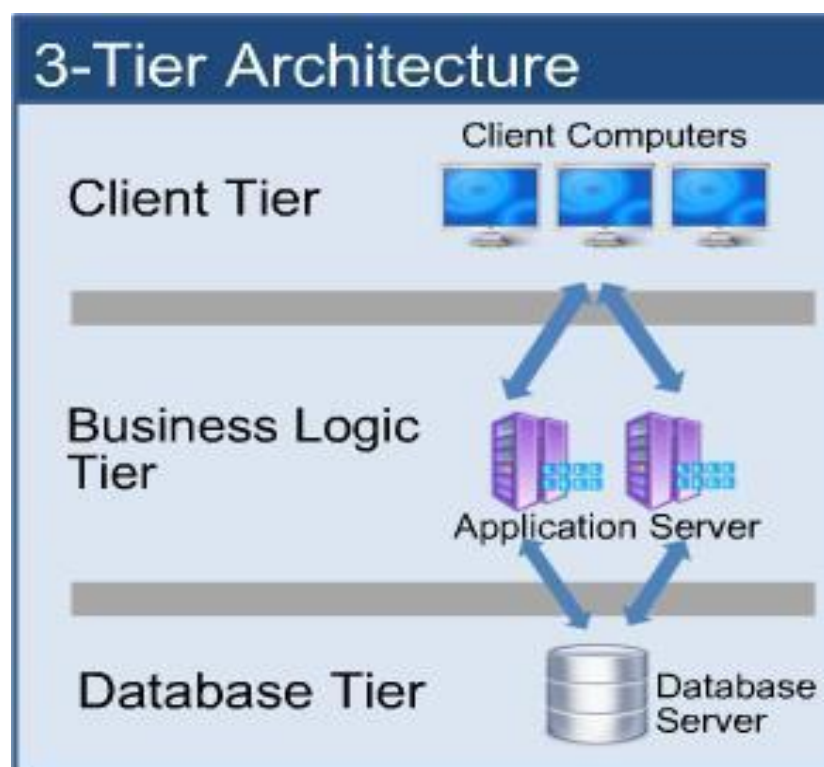


Figure 11: Représentation architecture 3-tiers

➤ L'architecture n-tiers

En génie logiciel, architecture multi-tiers (souvent appelé architecture n-tiers) est une architecture client-serveur dans lequel la présentation, le traitement de la demande et la gestion des données sont logiquement des processus séparés. Par exemple, une application qui utilise le middleware aux demandes de données de service entre un utilisateur et une base de données, emploie architecture multi-tiers. L'utilisation la plus répandue de l'architecture n-

tiers est l'architecture à trois niveaux. L'architecture d'une application N-tiers fournit un modèle pour les développeurs pour créer une application flexible et réutilisable. En décomposant l'application dans les niveaux, les développeurs peuvent modifier ou ajouter une couche spécifique, plutôt que d'avoir à réécrire toute l'application. Les concepts de couche et de niveau sont souvent utilisés de façon interchangeable. Cependant, un point de vue assez commun, c'est qu'il y'a effectivement une différence, et qu'une couche est un mécanisme de structuration logique pour les éléments qui composent la solution logicielle, tandis qu'un niveau est un mécanisme de structuration physique pour l'infrastructure du système.

Les couches de l'arch. n-tiers

6

- Une architecture n-tiers comprend généralement une **couche de présentation**, une **couche applicative**, une **couche objets métier** et une **couche d'accès aux données**

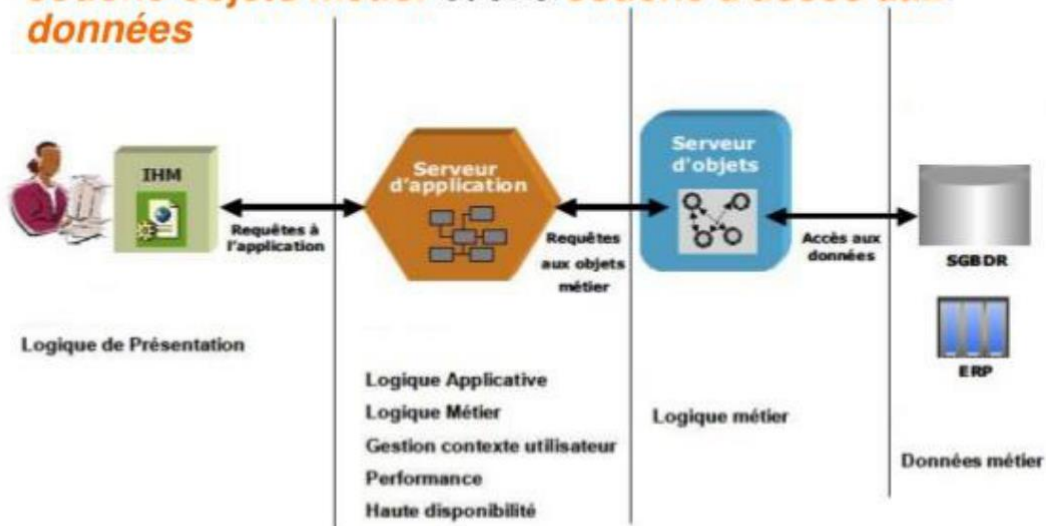


Figure 12: Représentation architecture n-tiers

4.2.4.2. Choix et Justification

Notre choix porte sur l'architecture 3-tiers car elle est une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client. Ses principaux avantages sont :

- Des ressources centralisées : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- Une meilleure sécurité : car le nombre de points d'entrée permettant l'accès aux données est moins important
- Un réseau évolutif : grâce à cette architecture il est possible de supprimer ou de rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

4.2.5. Architecture physique du système : diagramme de déploiement :

4.2.5.1. Définition et rôle

Le diagramme de déploiement montre la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels. Les ressources matérielles sont représentées sous forme de nœuds. Les nœuds sont connectés entre eux à l'aide d'un support de communication.

4.2.5.2. Diagramme de déploiement

Le diagramme suivant montre la disposition physique de notre application.

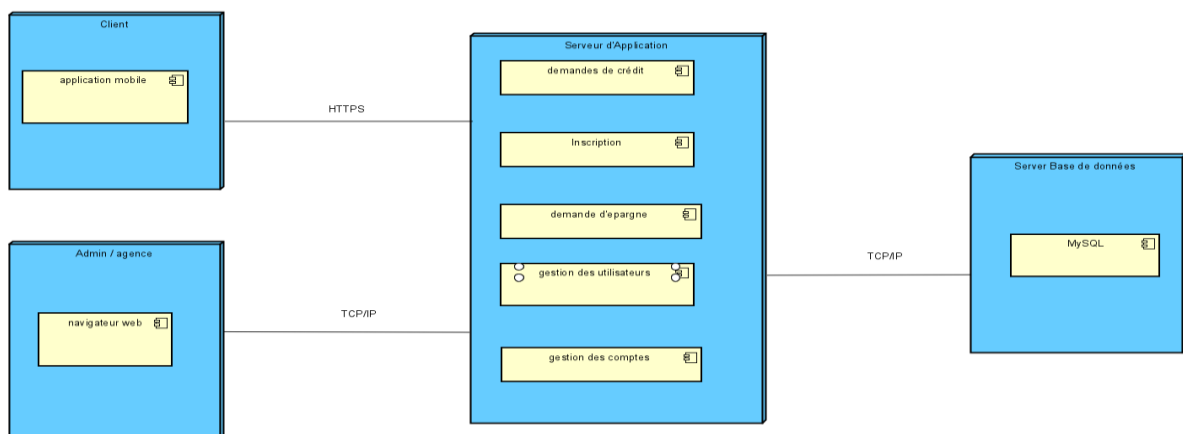


Figure 13: Diagramme de déploiement

4.2.6. Gestion de la sécurité

4.2.6.1. Définition et rôle

C'est une phase très importante lors de la création d'une application. Les moyens de sécurités utilisées renforcent la qualité de l'application et lui permet de durer sur le marché. C'est dans ce contexte que nous allons mettre des normes de sécurité comme l'authentification et les privilèges aux utilisateurs pour accéder à certaines fonctionnalités de l'application.

4.2.6.2. Gestion des accès

La gestion des accès informatique est un élément clé de la sécurité qui repose sur la cohérence d'attribution des droits relatifs aux ressources du système d'information. La gestion des accès informatiques consiste à administrer le cycle de vie des utilisateurs dans le système d'information et leurs accès aux données. Il s'agit de définir des privilèges dans la consultation et l'utilisation des données. Ces privilèges doivent être déterminés en fonction du besoin réel de chaque collaborateur dans la réalisation des tâches qui lui incombent au sein de la société.

4.2.6.3 Politique de sauvegarde

Quel que soit la qualité des moyens de défense mis en œuvre (physique ou logiques) les données peuvent être altérées sciemment ou accidentellement. Les données et les applications informatiques doivent être disponibles « à tout moment » lorsqu'on en a besoin, et doivent être conservées (sauvegardées) afin de pouvoir être récupérées (restauration) le moment voulu. Il convient par conséquent de définir une politique de sauvegarde. Il n'y a pas de politique de sauvegarde universelle. Elle doit être définie en fonction du volume de données, de la quantité d'information que l'on accepte de perdre, et éventuellement de la durée « légale » de conservation de l'information.

➤ Sauvegarde complète :

C'est une méthode de type « annule » et « remplace ». On écrase le contenu de sauvegarde par la nouvelle information. Méthode très sûre mais longue si le volume est important (par ex : la sauvegarde de gros volumes peut être supérieure à la durée de la nuit et empêcher le travail des utilisateurs le lendemain matin).

➤ **Sauvegarde différentielle :**

C'est une méthode qui sauvegarde toutes les informations qui ont été modifiées depuis la dernière sauvegarde complète.

➤ **Sauvegarde mixte :**

- ✓ Une sauvegarde journalière différentielle
- ✓ Une sauvegarde complète
- ✓ Une sauvegarde mensuelle
- ✓ À chaque intervention technique (mise à jour, ...) sur un poste de travail ou un serveur, une sauvegarde complète (image de la machine) du poste ou du serveur réalisé par le prestataire.

➤ **Sauvegarde incrémentale :**

C'est une méthode qui ne sauvegarde que les informations qui ont été modifiées depuis la dernière sauvegarde enregistrée sur le support.

4.3. Capture d'Ecran de la solution

4.3.1. Capture d'écran partie Administration

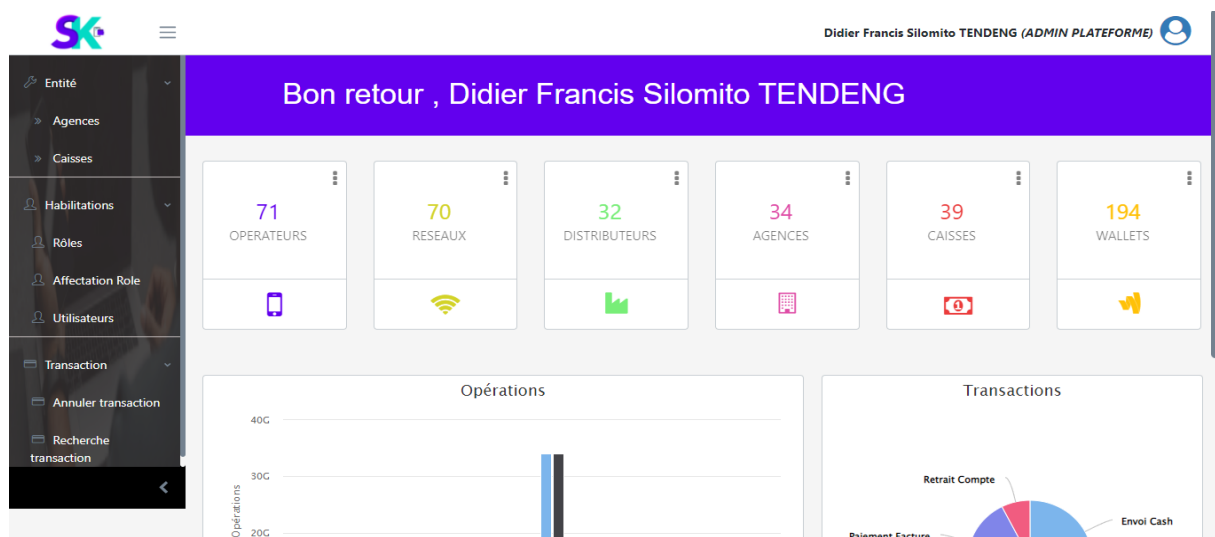
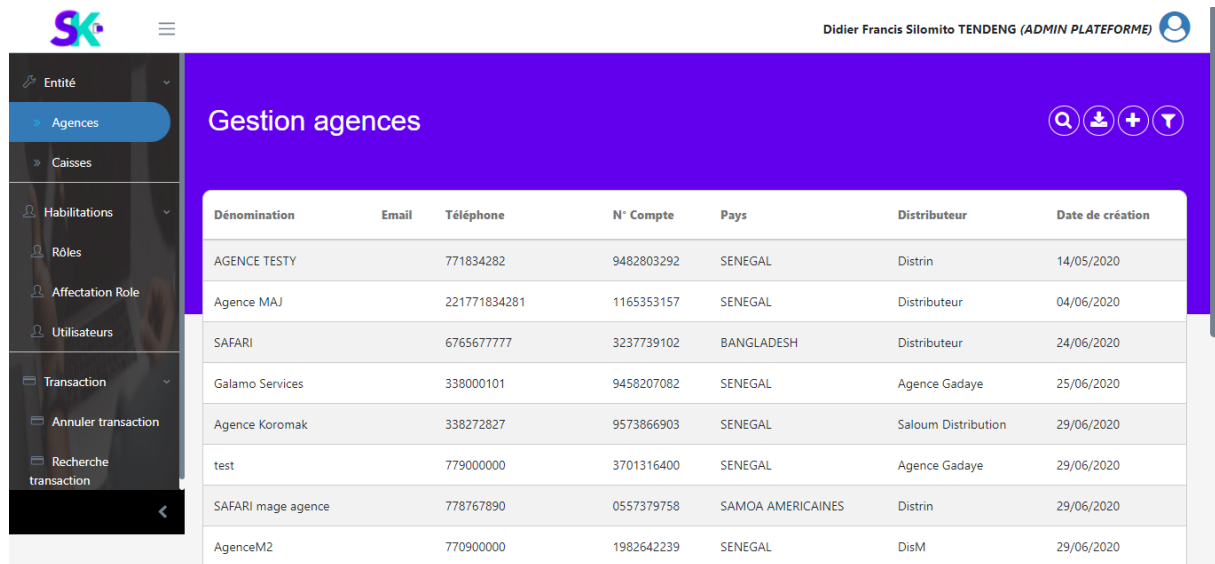


Figure 14: Interface Administration

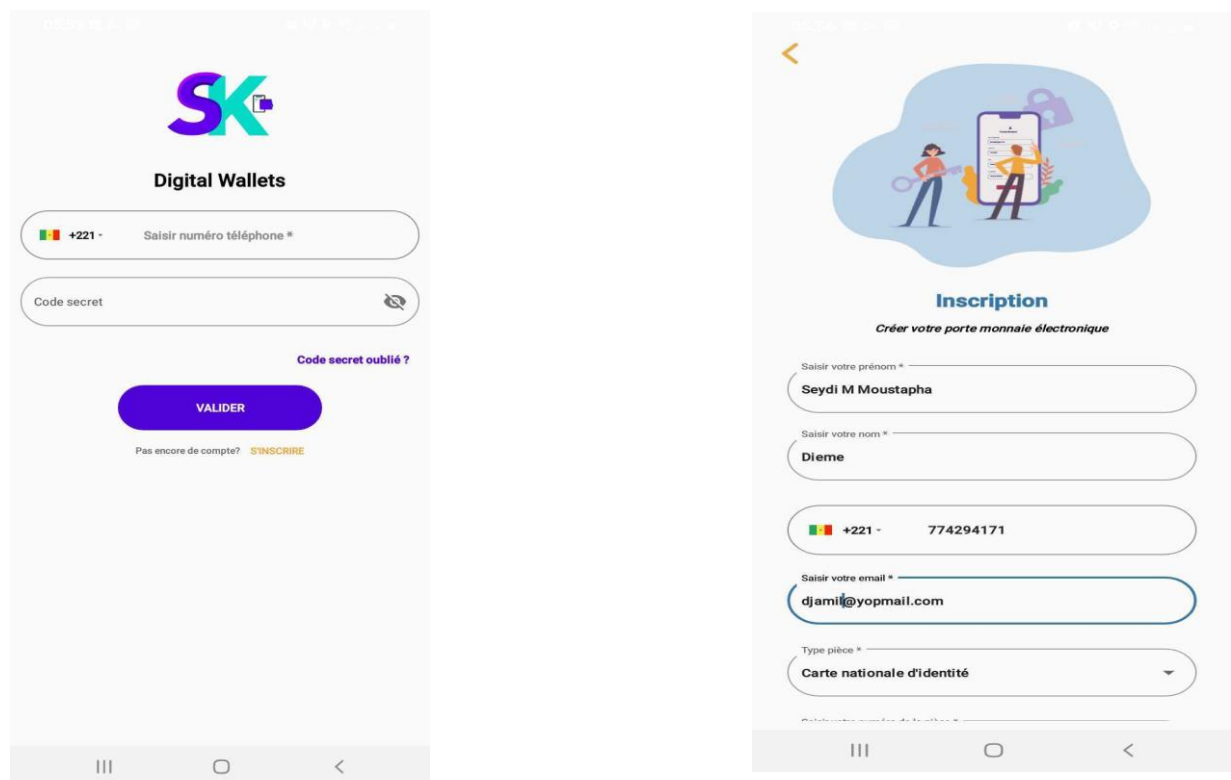
4.3.2. Capture d'écran partie Agence



Dénomination	Email	Téléphone	N° Compte	Pays	Distributeur	Date de création
AGENCE TESTY		771834282	9482803292	SENEGAL	Distrin	14/05/2020
Agence MAJ		221771834281	1165353157	SENEGAL	Distributeur	04/06/2020
SAFARI		6765677777	3237739102	BANGLADESH	Distributeur	24/06/2020
Galamo Services		338000101	9458207082	SENEGAL	Agence Gadaye	25/06/2020
Agence Koromak		338272827	9573866903	SENEGAL	Saloum Distribution	29/06/2020
test		779000000	3701316400	SENEGAL	Agence Gadaye	29/06/2020
SAFARI mage agence		778767890	0557379758	SAMOA AMERICAINES	Distrin	29/06/2020
AgenceM2		770900000	1982642239	SENEGAL	DisM	29/06/2020

Figure 15: Interface Agence

4.3.3. Capture d'écran Application Mobile



Digital Wallets

+221 - Saisir numéro téléphone *

Code secret

Code secret oublié ?

VALIDER

Pas encore de compte? **S'INSCRIRE**

Inscription

Créer votre porte monnaie électronique

Saisir votre prénom *

Seydi M Moustapha

Saisir votre nom *

Dieme

+221 - 774294171

Saisir votre email *

djamil@yopmail.com

Type pièce *

Carte nationale d'identité

Figure 16: Interfaces Ecran de Présentation(Connexion et Inscription)

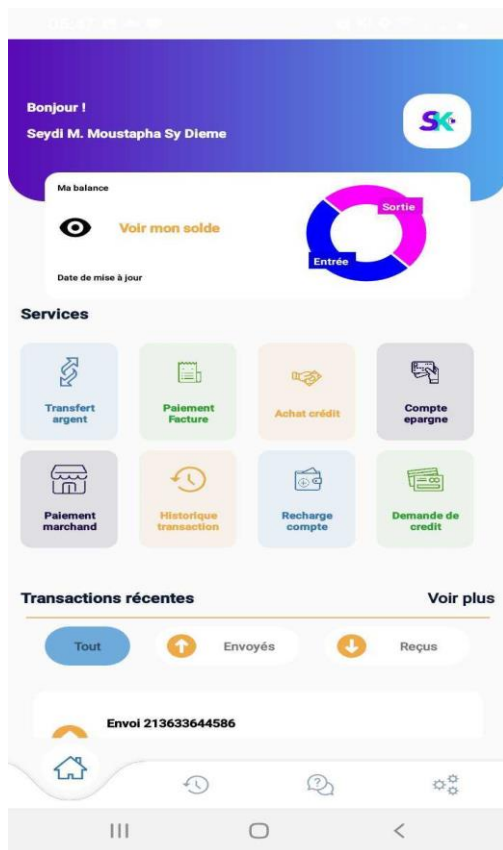


Figure 17: Ecran d'accueil

Figure 18: Ecran menu de transfert
Figure 19: Ecran d'accueil

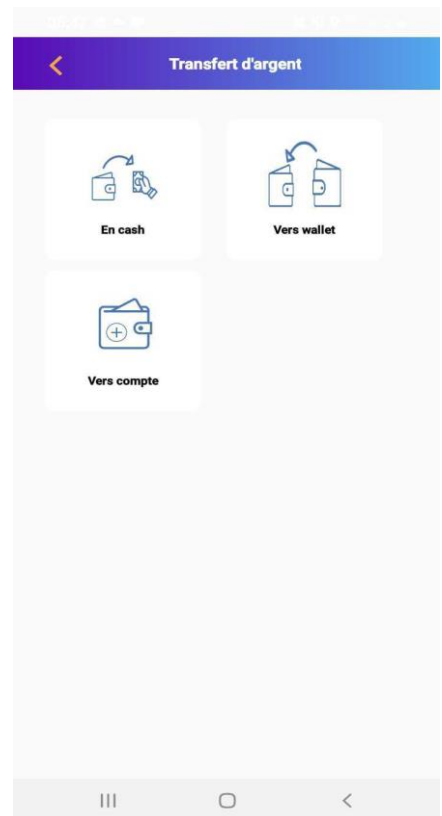


Figure 20: Ecran menu de transfert

Figure 21: Ecran transfert en cash
Figure 22: Ecran menu de transfert



Figure 23: Ecran transfert en cash

Figure 24: Ecran transfert en cash

Conclusion

Dans ce chapitre, nous avons pu présenter les langages de développement et de SGBD et une aperçue de l'évaluation financière. Nous avons exposé ainsi le résultat de développement à l'aide de capture d'écran.

Conclusion et perspectives

Notre projet intitulé « **MISE EN PLACE D'UNE APPLICATION DE GESTION DE PORTEFEUILLE ELECTRONIQUE** » **consiste** à la conception et la réalisation d'un portefeuille électronique.

En ce qui concerne la démarche, nous avons en premier lieu effectué une phase d'étude des différentes solutions de transferts d'argent existants et nous avons spécifié notre application pour discerner les fonctionnalités. En deuxième lieu nous sommes penchés sur Modélisation Conceptuelle & Organisationnelle de notre solution. En troisième lieu, nous avons procédé à sa conception ainsi qu'aux choix technologiques pour sa réalisation. Enfin, nous l'avons mise en œuvre.

Dans nos perspectives nous prévoyons de chercher des partenariats, l'ajout des fonctionnalités tels que la fonction de gestion des commerçants, la possibilité de lier son compte bancaire directement avec notre portefeuille électronique, le développement de l'application vers iOS(iPhone).

Dans le futur notre portefeuille électronique pourra même se décentraliser aux tours de famille et pourra même intégrer de nouvelle technologie comme le Machine Learning, la Blockchain.

Il est important à noter que la réalisation de ce projet nous a été bénéfique sur tous les plans. Sur le plan technique, ce projet nous a été une bonne occasion pour découvrir la technologie des apis, d'approfondir nos connaissances sur le développement web, mobile.

Sur le plan humain, ce projet a été une véritable occasion de vivre de près l'expérience du travail au sein d'une société, lors de nos enquêtes avec **Le Directeur Général de CHAKA MOBILE, Mr Mouhamadou El Mansour Sall** qui nous a gratifié de son accueil chaleureux.

Annexe

TABLEAU FINANCIER :

L'étude financière permet de faire le bilan de l'ensemble des besoins nécessaires pour mettre en place l'application. Dès lors, nous allons les répertorier dans un tableau.

Besoins	Articles	Descriptions	Quantité	Prix unitaire (F CFA)	Cout total par articles
Matériel informatique	Ordinateur	Processeur core i5, écran 15pouces, disque dur 500giga	7	275.000	1.925.000
	Imprimante	Imprimante Multifonction laser couleur Laserjet MFP m477fdw	3	320.000	960.000
	Photocopieur	Photocopieurs complet, impression en A3 et A4, numérisation	1	1.300.000	1.300.000
	Serveur Dédié	2xIntelXeon E52620 128GB RAM, 2To	1	5.000.000	5.000.000
	Smartphone		7	90.000	630.000
		Routers CISCO 400	1	4.721.835	4.721.835
	Matériel de connectivité	Switch (TP-Link TLSG1024D Switch 24 Ports Gigabit (Bureau/Rackable, Boîtier Métal))	2	60.000	120.000
		Abonnement Internet	1	150.0000	150.000

		Des cables (câbles de catégorie 6 pour le câblage réseau)	5	100.000	500.000
		TOTAL			5.491.835
Licence	Licence MySQL	MySQL Standard Edition Subscription (5+ socket server)	1	2.200.000	2.200.000
Matériel Electrique	Multiprise	4 prises avec fusible	2	5.000	10.000
	Régulateur de tension pour le serveur		1	40.000	40.000
Main d'œuvre	Conceptiond éploiement Formation – Maintenance sur 3mois			1.500.000	1.500.000
COUT TOTAL					18.946.835 ²

² Les montants sont exprimés en Francs CFA.

BIBLIOGRAPHIE :

- Pierre-Alain Muller – Modélisation objet avec UML, Eyrolles 1997
- Modélisation des bases de données de Christian Soutou et Frédéric Brouard, 15 mars 2012
- Programmer avec MySQL de Christian Soutou édition 2006, édité par EYROLLES
- Note d'information 4ème trimestre 2019 N° 60 - BCEAO

WEBOGRAPHIE :

<https://www.journaldunet.fr/patrimoine/guide-des-finances-personnelles/1208977-wallet/>

<https://www.memoireonline.com/04/13/7128/Etude-pour-la-mise-en-place-d-un-systeme-de-paiement-electronique-dans-une-institution-financiere.html>

<http://projet.eu.org/pedago/sin/term/3-UML.pdf>

<https://www.youtube.com/watch?v=dJd6azZr9Kg&list=PLRR7wjtXb1cBQCE8ddM0B1D9D Fj-WL3BX>

<https://scand.com/company/blog/why-use-java-for-back-end-development/>

<https://www.bceao.int/sites/default/files/2020-02/Note%20d%27information%20n%C2%B0%2060%2C%204e%20trimestre%202019.pdf>