



eGate

Université Joseph KI-ZERBO (U.J.K.Z)

Unité de Formation et de Recherche en Sciences Exactes et Appliquées (U.F.R./S.E.A)

Département : Informatique

Année Académique 2019/2020 UJKZ/SEA/Dept-Info/LSIR/2023

RAPPORT DE STAGE DE FIN DE CYCLE

8003

THÈME: Conception et développement d'un système de gestion des transactions mobiles

8003

Pour l'obtention de la *Licence Informatique* Option : *Système d'Information et Réseau* (SIR)

> <u>Stage effectué par :</u> **OUEDRAOGO Issaka**

Période de stage: du 03 Mars au 03 Juin 2023

Lieu de Stage: eGate [1200 lgts, Ouagadougou – Burkina]

Superviseur:

M. Ousmane BARRA, Enseignant en informatique à Université Joseph Ki Zerbo

Maitre de stage:

M. Sylvain ZONGO, Responsable de eGate

DEDICACE

\circ							
(nie c	0	travall.	temo	1Ane	nos	respects	•
Qui	·	uuvuu	venio	igni	1103	respects	•

A nos parents: Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de nôtres études.

Aucune dédicace ne pourrait exprimer notre respect, notre considération et nos profonds sentiments envers eux.

On prit le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de nous.

A nos sœurs et nos frères

A tous nos professeurs: Leur générosité et leur soutien nous oblige de leurs témoigner notre profond respect et notre loyale considération.

A tous nos amis: Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

REMERCIEMENTS

Au terme de la rédaction de rapport de projet de fin d'études, il nous est particulièrement agréable d'exprimer notre gratitude à Dr. Francis BASSIONO directeur de l'UFR-SEA ainsi que Dr. Didier BASSOLE chef de département et le corps professoral, doctoral et administratif pour tous efforts déployés en faveur de la renommée de notre université.

Nos vifs remerciements vont également à notre encadrant M. Ousmane BARRA enseignant à l'UJKZ, pour son encadrement qui nous a accompagné avec patience durant la réalisation de ce travail.

Ainsi, le devoir de la gratitude et de la reconnaissance nous impose à exprimer nos vifs remerciements à M. Sylvain ZONGO le responsable de l'entreprise eGate, notre maître de projet pour leur aide et collaboration.

Finalement, nous remercions toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

AVANT-PROPOS

L'Université Joseph KI-ZERBO, la première Université publique au Burkina Faso, créée en 1974 est un établissement à caractère scientifique, culturel et technique. Elle porte le nom de l'intellectuel et historien Joseph KI-ZERBO aujourd'hui. Elle est composée des éléments suivants afin de mener à bien sa tâche :

- quatre écoles doctorales :
 - ✓ École Doctorale Sciences de la Santé (ED-2S) ;
 - ✓ École Doctorale Lettres Sciences Humaines et Communication (ED-LESHCO);
 - ✓ École Doctorale Sciences et Technologies (ED-ST) ;
 - ✓ École Doctorale Informatique et changements climatiques (ED-ICC) .
- > six instituts:
 - ✓ Institut Burkinabè des Arts et Métiers (IBAM) ;
 - ✓ Institut Supérieur des Sciences de la Population (ISSP) ;
 - ✓ Institut Panafricain d'Étude et de Recherche sur les Médias, l'Information et la Communication (IPERMIC) ;
 - ✓ Institut Sciences du Sport et Développement Humain (ISSDH);
 - ✓ Institut Formation Ouverte et à Distance (IFOAD) ;
 - ✓ Institut Environnement et Développement Durable (IGEDD).
- > cinq Unités de Formation et de Recherche (U.F.R.) :
 - ✓ Unité de Formation et de Recherche en Lettres, Arts et Communication (U.F.R./LAC);
 - ✓ Unité de Formation et de Recherche en Sciences Humaines (U.F.R./S.H.);
 - ✓ Unité de Formation et de Recherche en Sciences de la Santé (U.F.R./S.D.S.) :
 - ✓ Unité de Formation et de Recherche en Sciences de la Vie et de la Terre (U.F.R./S.V.T.) ;
 - ✓ Unité de Formation et de Recherche en Sciences Exactes et Appliquées (U.F.R./S.E.A.).
- deux Centres Universitaires (CU) :
 - ✓ Centre Universitaire de Ziniaré (CU-Z) ;
 - ✓ Centre Universitaire Polytechnique de Kaya (CU-PK).

L'U.F.R./S.E.A., U.F.R. de prestige de par la qualité des formations en sciences, est comme toutes les autres U.F.R., subdivisées en plusieurs départements à savoir, le département de :

- Mathématiques ;
- Physique;
- Chimie;
- Informatique.

Tout comme aux autres départements, l'accès au département d'informatique se passe par le tronc commun en Sciences et Technologies (ST) en première année. Il a été créé en 2009 et est devenu indépendant en 2015.

L'obtention du diplôme de la licence en informatique nécessite un stage pratique d'une durée minimum de trois (03) mois. C'est dans cette optique que, nous avons été accueillis dans les locaux de l'entreprise eGate pour notre stage.

SIGLES ET ABREVIATIONS

Acronyme	Signification	
API	Application Programming Interface	
СОСОМО	COnstructive COst MOdel	
CPU	Central Processing Unit	
CU	Cas d'Utilisation	
DOM	Document Object Model	
Go	Giga Octet	
MERISE	Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise	
Мо	Méga Octet	
MVC	Model View Controler	
MVVP	Model View View Controler	
MVP	Model View Pesenter	
OS	Operating System	
PHP	Hypertext Preprocessor	
RAM	Random Access Memory	
SEA	Sciences Exactes et Appliquées	
SARL	Société A Responsabilité Limité	
SGBD	Système de Gestion de Base de Donnée	
Sys.M.L	Systems Modeling Language	
UFR	Unité de Formation et de Recherche	
UML	Unified Modeling Language	
TIC	Technologies de l'Information et de la Communication	
XP	eXtreme Programming	

TABLE DES TABLEAUX

Tableau 1 : Etude comparative des méthodologies	7
Tableau 2 : Etude comparative des langages de modélisation	13
Tableau 3 : CU « Enregistrer transaction »	17
Tableau 4 : CU « S'authentifier »	18
Tableau 5 : CU « Modifier paiement »	19
Tableau 6 : CU « Ajouter approvisionnement »	20
Tableau 7 : Etude comparée de quelques langages de programmation	31
Tableau 8 : Etude comparée de quelques architectures logicielles	36
Tableau 9 : Etude comparée de quelques Framework Backend	40
Tableau 10 : Etude comparée de quelques Framework Frontend	44
Tableau 11: Tableau comparative de quelques SGBD	48
Tableau 12 : Etude comparée de quelques approches d'estimation de coûts	55
Tableau 13 : Modèle COCOMO	56
Tableau 14:Tableau récapitulatif	58

TABLE DES FIGURES

Figure 1 : Planning prévisionnel	4
Figure 2 : Description de la méthodologie XP	9
Figure 3 : Logo M.E.R.I.S.E	10
Figure 4 : Diagrammes U.M.L	11
Figure 5 : Diagrammes Sys.M.L	
Figure 6 : CU lié à acteur « Gestionnaire »	
Figure 7 : CU lié à acteur « Responsable »	16
Figure 8 : Diagramme de séquence « S'authentifier »	21
Figure 9 : Diagramme de séquence « Enregistrer transaction »	
Figure 10 : Diagramme de séquence « Modifier paiement »	
Figure 11 : Digramme d'activité « S'authentifier »	
Figure 12 : Digramme d'activité « Enregistrer transaction »	
Figure 13 : Diagramme de classe du projet	
Figure 14 : Caractéristique de l'ordinateur portable	
Figure 15 : Logo PHP	
Figure 16 : Logo Java	
Figure 17 : Logo JavaScript	
Figure 18 : Logo TypeScript	
Figure 19 : Model-View-Controler (MVC)	
Figure 20 : Model-View-Presenter (MVP)	
Figure 21 : Model-View-View Model (MVVM)	
Figure 22 : Logo Spring Boot	
Figure 23 : Logo Struts	
Figure 24 : Logo Play	
Figure 25 : Logo React.js	
Figure 26 : Logo AngularJS	
Figure 27 : Logo VueJS	
Figure 28 : Logo Oracle Database	
Figure 29 : Logo MariaDB	
Figure 30 : Logo PostgreSQL	
Figure 31 : Logo VS Code	
Figure 32 : Logo phpMyAdmin	
Figure 33 : Logo NodeJS	
Figure 34 : Logo Postman	
Figure 35 : Logo GitHub	
Figure 36 : Logo Enterprise Architect	
Figure 37 : Logo Figma	
Figure 38 : Logo GanttProject	
Figure 39 : Logo Zotero	
Figure 40 : Page de connexion	
Figure 41: Page d'accueil de l'administrateur	
Figure 42: Page d'accueil du gestionnaire	
Figure 43 : Page « Liste de transaction »	
Figure 44 : Page « Ajouter une transaction »	
Figure 45 : Planning prévisionnel réel	61

Table des matières

DED	ICACE	i
AVA	NT-PROPOS	iii
SIGL	LES ET ABREVIATIONS	v
TAB	LE DES TABLEAUX	vi
TAB	LE DES FIGURES	vii
INTF	RODUCTION	1
CHA	PITRE I : Généralités	2
I. P	résentation du cadre du stage	2
II. P	Présentation du thème	2
1.	Présentation du projet	2
2.	Problématique	2
3.	Objectifs	3
4.	Livrable	3
III.	Gestion de projet	3
1.	Acteurs du projet	3
2.	Planning prévisionnel	4
CHA	PITRE II : Analyse et Modélisation du système	5
I. A	Analyse du projet	5
1.	Domaine d'étude	5
2.	Fonctionnalités	5
3.	Méthodologie	6
II. N	Modélisation	9
1.	Langage de modélisation	9
2.	Modélisation	14
СНА	PITRE III : Réalisation	27
I. C	Outils de Développement	27
1.	Ressources matériels utilisées	27
2.	Langage de développement	28
3.	Architecture Logicielle	33
4.	Framework	37
5.	Système de gestion de bases de données (SGBD)	45
6.	Autres technologies utilisées	49
II. E	Estimation du coût	54
1.	Quelques approches d'estimation des coûts	54

2.	Etude comparative de quelques approches d'estimation des coûts	54
3.	Choix de notre approche	56
4.	Estimation du coût	56
III.	Description d'application	58
1.	Page de connexion	58
2.	Page d'accueil	59
3.	Page gestion des transactions	60
IV.	Planning réel de réalisation et analyse des écarts	61
CON	NCLUSION	62
BIB	LIOGRAPHIE	63
WEI	BOGRAPHIE	64

INTRODUCTION

Au cours des dernières décennies, l'évolution rapide des technologies de l'information et de la communication a révolutionné notre manière de gérer les transactions financières. Avec l'avènement des moyens de paiements électroniques tels que les cartes bancaires, les portefeuilles numériques et les services de mobile money, nous assistons à une transformation significative du paysage financier. Ces nouvelles méthodes de paiement offrent des avantages considérables en termes de rapidité, de sécurité et de praticité pour les consommateurs et les commerçants.

L'avènement des moyens de paiements électroniques a entrainé une prolifération des kiosques mobile money. Au niveau de ces kiosques, pour plus de sécurité, les transactions financières sont enregistrées sur des formulaires papiers pré-imprimés ou au sein d'un registre. Cette pratique engendre quelques problèmes à savoir des erreurs de saisie et de bilans financiers, des insécurités des données. Afin de faire face ces limites, eGate une société spécialisée dans le développement web et mobile, a décidé de mettre en place un outil afin de garantir une gestion sécurisée, fiable et efficace tout en optimisant l'expérience utilisateur. Ainsi, notre stage vise à mettre en place une plateforme informatique web qui répondra aux besoins de la société eGate à travers le développement d'une solution web. Ce projet gravite autour du thème suivant : « Conception et développement d'un système de gestion des transactions mobiles ».

Ce document décrit les étapes de développement de notre projet. Il contient trois (3) chapitres dont le premier est une généralité qui consiste à présenter le cadre du travail, le thème et la gestion du projet, le deuxième est une analyse et une modélisation de notre système qui consiste à faire une analyse et une spécification des besoins, ainsi que la conception U.M.L du système et le dernier qui consiste à faire une réalisation et une présentation de notre système.

CHAPITRE I : Généralités

Au sein de ce chapitre, nous présentons d'abord notre structure d'accueil, ensuite notre thème de projet et enfin les acteurs et le planning prévisionnel.

I. Présentation du cadre du stage

eGate est une société privée à responsabilité limitée (S.A.R.L) créée en novembre 2021 offrant des services dans les domaines suivants :

- Développement d'application ;
- > Test d'application, audit et consultation ;
- Maintenance et fourniture de matériel informatique et imprimerie ;
- Formation en collecte, traitement de données et virtualisation ;
- Vente de logiciels ;
- Formation en bureautique et en développement d'applications web et mobile.

eGate a mis sur le marché de nombreuses applications au Burkina Faso mais également dans plusieurs pays. Nous pouvons citer la Guinée Conakry, La République Démocratique Du Congo, Le Mali et bien d'autres.

II. Présentation du thème

1. Présentation du projet

L'entreprise eGate a constaté depuis l'événement de moyens des paiements électroniques où il a eu une prolifération des kiosques mobile money que la plupart de ces kiosques enregistrent leurs transactions financières sur des formulaires papiers pré-imprimés ou au sein d'un registres. Cette pratique engendre quelques limites. Face à ces limites, eGate nous a confié de mettre en place une application web qui permettra de faire face à ces limites.

2. Problématique

Un kiosque mobile money permet d'effectuer des transactions financières d'où une gestion impérative de ses transactions financières. Cette gestion étant manuelle, les kiosques mobiles money rencontrent des difficultés telles que :

- ➤ Erreurs humaines : lorsque les transactions sont gérées manuellement, il y a un risque d'erreurs de saisie ou de calcul de la part de l'opérateur. Cela peut entraîner des déséquilibres dans les comptes, des erreurs de transaction ou des problèmes de suivi des paiements ;
- Manque d'efficacité : la gestion manuscrite nécessite souvent beaucoup de temps et de ressources. Les gestionnaires doivent prendre des notes, effectuer des calculs manuels

- et suivre les transactions de manière manuelle. Cela peut ralentir le processus de gestion et entraîner des retards dans le traitement des transactions ;
- Manque de sécurité des données : Les enregistrements papier peuvent être perdus, volés ou endommagés, ce qui pourrait entraîner la perte de données précieuses ;
- Manque de stockage : la gestion manuelle occupe assez d'espace au sein du bureau du gestionnaire à cause des registres.

3. Objectifs

L'objectif de notre projet est de mettre en place une application web qui permettra de gérer des mouvements Mobile Money. Ainsi, notre application permettra de :

- > Gérer les enregistrements des transactions : le système doit pouvoir permettre aux gestionnaires d'enregistrer les dépôts, les retraits et les transferts d'argent ;
- ➤ Gérer les enregistrements des paiements : le système doit pouvoir permettre aux gestionnaires d'enregistrer les paiements de factures, de abonnements de toutes sortes ;
- > Gérer les enregistrements des ventes d'unités ;
- Consulter l'historique des transactions : les gestionnaires doivent pouvoir consulter l'historique à une date donnée ;
- Gérer les dépenses de l'entreprise ;
- Gérer les clients et les dealers.

4. Livrable

Un livrable est le résultat attendu d'un projet, les différents livrables de notre projet sont comme suit :

- ➤ Une application web de gestion de transactions mobiles ;
- ➤ La documentation technique.

III. Gestion de projet

1. Acteurs du projet

Le groupe de pilotages est composé de l'ensemble des personnes indispensables dans la prise de décisions relatives à l'atteinte des différents objectifs fixés. Il est chargé de la supervision du projet dans son ensemble et met en place les techniques et moyens pour sa réalisation en respectant les contraintes de délais et de fiabilité. Dans le cadre de notre réflexion, le groupe composé :

M. Ousmane BARRA, enseignant du département d'Informatique (UJKZ) (Superviseur);

M. Sylvain ZONGO, le responsable de l'entreprise eGate (Maitre de stage).

Le groupe de projet est chargé de l'exécution du projet, à savoir l'étude, la conception et éventuellement la réalisation de l'application. Il est composé de OUEDRAOGO Issaka étudiant au département Informatique (**U.J.K.Z**) (Stagiaire) sous l'encadrement de M. Sylvain ZONGO (Maitre de stage).

2. Planning prévisionnel

Un planning prévisionnel est un document qui établit une chronologie des tâches et des activités prévues pour un projet. Il fournit une vue d'ensemble des différentes étapes du projet, de leur durée estimée et de leur séquencement. Le planning prévisionnel permet de planifier les ressources nécessaires, de définir les délais et de suivre l'avancement du projet. Ainsi, notre planning prévisionnel se présente comme suit :

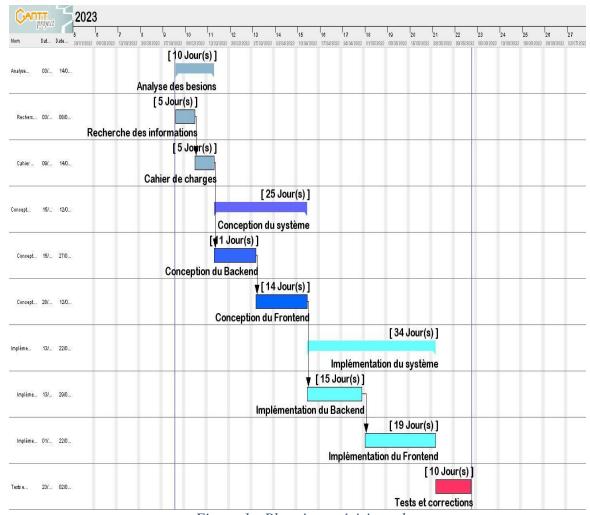


Figure 1 : Planning prévisionnel

CHAPITRE II: Analyse et Modélisation du système

I. Analyse du projet

1. Domaine d'étude

Le domaine d'étude de notre projet se situe dans le secteur des technologies de l'information et de la communication (TIC). Il s'agit d'utiliser ces technologies afin de faciliter et optimiser la gestion des transactions mobiles au sein des kiosques mobile money.

C'est dans ce cadre, que lors de l'entretien avec le client, nous avons pu conserver les tâches principales suivantes :

- Proposition de la gestion des transactions mobiles : le système que nous allons mettre en place doit permettre aux gestionnaires du kiosque mobile money d'enregistrer les dépôts, les retraits et des transferts ;
- ➤ Proposition de la gestion des paiements des factures : le système que nous allons mettre en place doit permettre aux gestionnaires d'un kiosque mobile money d'enregistrer les paiements de factures d'eau et d'électricité, les paiements d'abonnements Canal +, de cash power et autres ;
- Proposition de la gestion des ventes d'unités : le système que nous allons mettre en place doit permettre aux gestionnaires kiosque mobile money d'enregistrer les ventes d'unités ;
- Proposition de la gestion des dépenses de son kiosque mobile money : Le système que nous allons mettre en place doit permettre aux gestionnaires d'un kiosque mobile money d'enregistrer les dépenses d'un kiosque mobile money ;
- Proposition de l'approvisionnement des comptes des gestionnaires : Le système doit permettre au responsable kiosque mobile money d'approvisionner les caisses mobile money et opérateurs téléphoniques de chaque gestionnaire.

2. Fonctionnalités

L'analyse des besoins nous a permis de modéliser les différentes fonctionnalités suivantes en fonction de différents utilisateurs du système.

Un gestionnaire peut :

- Enregistrer les transactions mobiles : cela permet aux gestionnaires d'effectuer des enregistrements lorsqu'un client veut faire une ou plusieurs transaction(s) mobile(s) ;
- Enregistrer les paiements : cela permet aux gestionnaires d'effectuer des enregistrements lorsqu'un client veut payer ces factures ;

- Enregistrer les ventes d'unités : cela permet aux gestionnaires d'effectuer des enregistrements lorsqu'un client veut faire une ou plusieurs achat(s) unité(s) ;
- Enregistrer les dépenses : permet aux gestionnaires d'enregistrer les dépenses d'un point Mobile Money ;
- Consulter les historiques et les statistiques des transactions : permet aux utilisateurs de voir les enregistrements effectués et les statistiques.

Le responsable peut :

- Enregistrer les approvisionnements des caisses : cela permet au responsable de faire les enregistrements des approvisionnements aux caisses lié à chaque compte du gestionnaire en cas de rupture ;
- ➤ Gérer les gestionnaires : cela permet au responsable d'ajouter, modifier et supprimer les gestionnaires.

3. Méthodologie

Le choix d'une méthodologie de développement de logiciel appropriée est essentiel pour assurer le succès d'un projet. En effet, une méthodologie bien définie offre une structure et des lignes directrices claires pour l'équipe de développement, permettant ainsi une meilleure organisation et une gestion efficace des ressources. Elle favorise également la collaboration entre les membres de l'équipe et garantit une communication fluide avec les parties concernées. De plus, une méthodologie adaptée permet de gérer les risques, d'assurer la qualité du code et de répondre aux besoins changeants du projet. En somme, le choix d'une méthodologie appropriée est essentiel pour maximiser la productivité, réduire les erreurs et mener à bien le projet avec succès.

Pour toutes ces raisons, nous allons faire une comparaison afin de choisir celle adaptée à nos réalités.

3.1. Etude comparative

Voici une comparaison de trois (3) méthodologies de développement logiciel :

Tableau 1 : Etude comparative des méthodologies

Métho	odologie	Caractéristique	Avantages	Inconvénient
		- Approche	 Planification claire, 	 Difficulté à gérer
		séquentielle	 Facilité de suivi des 	les changements,
		Linéaire	progrès,	- Retours
Cascad	e	 Planifiée 	 Dépendances 	d'expérience des
			claires entre les	utilisateurs tardifs,
			phases.	 Délais rallongés
				en cas de
				modifications.
		Approche	 Visualisation du 	 Manque de
		visuelle	flux de travail	structure pour les
		- Flexible	 Gestion des tâches 	projets complexes
	Kanban	 Centrée sur 	en temps réel,	Risque de
		l'amélioration	 Réduction des 	surcharge de
		continue pour la	délais	travail
		gestion des	 Amélioration 	 Dépendance à une
		tâches dans le	continue	communication
		développement	– Flexibilité	fluide
		logiciel		
Agiles		- Approche	 Adaptabilité 	Apprentissage
		itérative et	Collaboration	initial
		incrémentale	 Livraison fréquente 	 Complexité de
	Scrum	- Utilisation de	- Transparence	gestion du backlog
		cycles de		 Nécessité d'une
		développement		culture d'équipe et
		- Approche		une coordination
		collaborative		étroite
		- Livraison		
		fréquente de		
		fonctionnalités		
		utilisables		

	XP	 Approche collaborative Communication constante Simplicité 	 Qualité du code Communication améliorée Livraison fréquente Réactivité aux 	 Nécessité d'une culture d'équipe et une coordination étroite Adaptabilité de
		Communication continueTests continusAdaptabilité	changements – Productivité accrue	l'équipe – Potentiel de dérive du projet
Spirale		 Approche itérative Gestion des risques Planification adaptative 	 Gestion des risques Evaluation itérative Test par cycles 	 Nécessité plus de temps et de ressources en raison des cycles itératifs. Nécessité d'une expertise Coûts de développement peuvent augmenter

3.2. Choix de la méthodologie

Après cette étude comparative, il ressort que la méthodologie la mieux adaptée à notre projet est la méthodologie agile XP (eXtreme Programming) car elle est facilement adaptable à l'évolution des besoins des clients, a une approche itérative, est rapide à mettre en œuvre, convient mieux aux petites équipes, et est principalement axée sur les besoins du client.

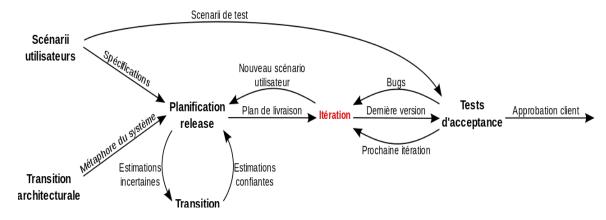


Figure 2 : Description de la méthodologie XP

Source: https://fr.wikipedia.org/wiki/Fichier:Extreme_programming.svg

Il est important de noter que les inconvénients potentiels de la méthodologie agile XP peuvent être atténués par une bonne planification, une formation adéquate de l'équipe et un soutien continu de l'organisation.

II. Modélisation

1. Langage de modélisation

Un langage de modélisation est un système de représentation symbolique permettant de décrire et de communiquer des concepts, des structures et des relations dans un domaine donné, facilitant ainsi la conception, l'analyse et la communication de modèles et de systèmes complexes. En effet, un langage de modélisation clair et précis facilite la compréhension et la collaboration entre les membres de l'équipe, réduit les risques d'erreurs et facilite la maintenance du logiciel. Il permet également d'aligner les parties prenantes sur une vision commune du système.

1.1. Quelques langages de modélisation

❖ M.E.R.I.S.E

M.E.R.I.S.E « Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise » est une méthode de conception, de développement et de réalisation de projets informatiques.

La méthode M.E.R.I.S.E est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. M.E.R.I.S.E définit trois niveaux de description du système d'information :

- ➤ Le niveau conceptuel ;
- ➤ Le niveau organisationnel ;
- ➤ Le niveau physique.

Cette approche permet de structurer et de représenter les différentes composantes d'un système d'information de manière claire et cohérente.



Figure 3: Logo M.E.R.I.S.E

Source: http://infofacile.over-blog.com/2017/11/introduction-a-merise.html

* U.M.L

Le langage de modélisation U.M.L « Unified Modeling Language ou Langage de Modélisation Unifié » est un langage visuel utilisé pour représenter, spécifier, construire et documenter les artefacts d'un système logiciel. Il consiste à savoir porter le bon effort au bon moment, à savoir se positionner suivant le bon angle de vision au bon moment et à savoir s'attacher au bon détail au bon moment.

La dernière de U.M.L est la version 2.5 et elle offre quatorze (14) diagrammes standardisés pour représenter différentes perspectives d'un système que sont :

- > Sept (7) diagrammes comportementaux : un diagramme de cas d'utilisation, un diagramme d'activité, un diagramme d'états-transitions et quatre (4) diagrammes d'interactions ;
 - NB : Les quatre (4) diagrammes d'interactions sont : un diagramme de séquences, un diagramme de temps, un diagramme de communication et un diagramme global d'interaction.
- ➤ Sept (7) diagrammes structurels : un diagramme de classe, un diagramme d'objets, un diagramme de composants, un diagramme de déploiement, un diagramme des paquets

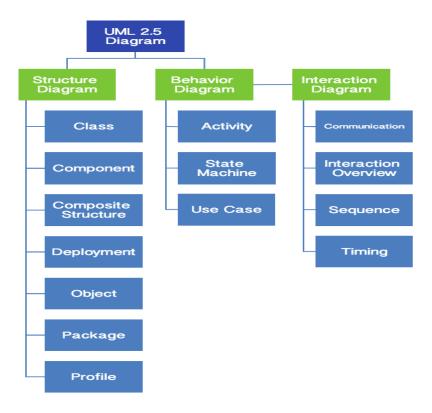


Figure 4: Diagrammes U.M.L

Source: https://cdn.nulab.com/learn-wp/app/uploads/2022/09/14212331/UML-2.5-
Diagrams-1.png

U.M.L facilite la communication entre les membres de l'équipe de développement, permet une compréhension claire des concepts et des relations du système, et peut être utilisé tout au long du cycle de vie du développement logiciel.

❖ Sys.M.L

Le Sys.M.L « Systems Modeling Language ou Language de Modélisation de Systèmes », est un language de modélisation graphique basé sur U.M.L utilisé dans le domaine de l'ingénierie des systèmes qui permet la description, l'analyse, la conception et la validation de systèmes.

Il permet de représenter et de communiquer les aspects structurels, comportementaux et fonctionnels d'un système. Sys.M.L est à la version 1.6 et il comporte huit (8) diagrammes :

- * Quatre (4) diagrammes comportementaux : un diagramme d'activité, un diagramme d'états, un diagramme de séquence et un diagramme cas d'utilisation ;
- * Un (1) diagramme des exigences ;

* Quatre (3) diagrammes structurels : un diagramme de définition de blocs, un diagramme paramétrique et un diagramme des paquets.

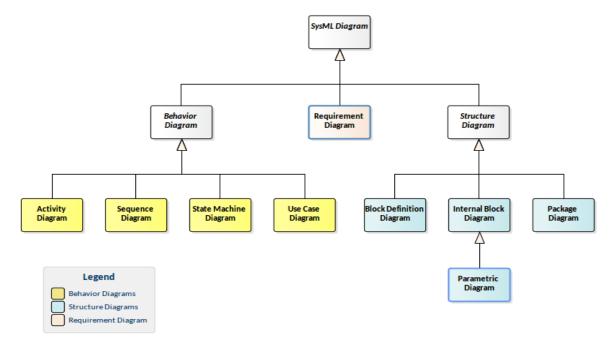


Figure 5: Diagrammes Sys.M.L

Source: https://www.sparxsystems.fr/enterprise architect user guide/16.0/images/sysmldiagram-taxonomy.png

Sys.M.L est largement utilisé dans les domaines de l'ingénierie des systèmes et de la conception de systèmes complexes, offrant une notation standardisée pour la modélisation, l'analyse et la documentation des systèmes.

1.2. Etude comparative

Un langage de modélisation adapté, nous favorisera une qualité du développement et nous maximiserons les chances de succès du projet. Une étude comparée nous permettra de choisir le mieux adapté à notre contexte.

Tableau 2 : Etude comparative des langages de modélisation

Langages de	Caractéristiques	Avantages	Inconvénients
modélisation			
M.E.R.I.S E	- Approche	- Structuration du processus	- Dépendance des
	descendante	de développement	schémas
	- Approche	- Compréhension des besoins	- Adaptabilité aux
	centrée sur les	- Séparation des	nouvelles
	données	préoccupations	technologies difficiles
		- Réutilisation des modèles	- Complexité
			- Rigidité
U.M.L	- Modélisation	- Communication efficace	- Nécessité d'un
	orientée objet	- Abstraction et	apprentissage
	- Variété de	conceptualisation	
	diagrammes	- Analyse et conception	
		efficaces	
		- Réutilisabilité	
		- Support pour le	
		développement logiciel	
Sys.M.L	- Modélisation	- Facilite la gestion de	- Outils de
	orientée objet	projets complexes	modélisation
	- Variété de	- Maintenabilité des	nécessaires
	diagrammes	systèmes complexes	- Portée limitée
		- Communication efficace	- Nécessité d'un
		- Abstraction et	apprentissage
		conceptualisation	
		- Analyse et conception	
		efficaces	
		- Réutilisabilité	

1.3. Choix du langage de modélisation

Le langage de modélisation le mieux adapté à notre projet est la modélisation UML car cette modélisation offre une représentation graphique polyvalente et standardisée pour la conception logicielle, systèmes d'information et systèmes orientés objet. UML est largement utilisé dans l'industrie et permet une modélisation plus complète et flexible, favorisant la communication et la compréhension entre les parties prenantes.

2. Modélisation

2.1. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation est un diagramme UML (Unified Modeling Language) qui permet une représentation graphique qui permet de décrire les interactions entre les acteurs (utilisateurs ou systèmes externes) et un système. Il permet d'identifier les fonctionnalités offertes par le système du point de vue des utilisateurs, en mettant l'accent sur les actions et les objectifs. Les acteurs sont représentés par des entités externes, tandis que les cas d'utilisation sont des scénarios spécifiques qui décrivent les actions effectuées par les acteurs. Ce diagramme facilite la communication entre les parties prenantes et aide à définir les besoins et les exigences fonctionnelles du système.

Ainsi, les différents acteurs identifiés pour notre système sont :

- Gestionnaire : il s'agit de la personne qui est chargée de faire des enregistrements ;
- Responsable : il s'agit du responsable du kiosque Mobile Money.

Les différents cas d'utilisations sont :

- > S'authentifier;
- Gérer enregistrement ventes d'unités ;
- Enregistrer vente d'unités ;
- Modifier vente d'unités ;
- > Supprimer vente d'unités ;
- Gérer enregistrement de transactions ;
- > Enregistrer transaction;
- ➤ Modifier transaction ;
- > Supprimer transaction :
- Gérer enregistrement dépenses ;
- Enregistrer dépense ;
- ➤ Modifier dépense ;

- Supprimer dépense ;
- Consulter solde ;
- Gérer gestionnaire ;
- > Ajouter gestionnaire;
- ➤ Modifier gestionnaire ;
- > Supprimer gestionnaire;
- > Gérer enregistrement approvisionnement ;
- > Enregistrer approvisionnement;
- ➤ Modifier approvisionnement ;
- > Supprimer approvisionnement.

Ci-dessous, nous présentons le diagramme de cas d'utilisation pour la compréhension du fonctionnement du système.

♣ Diagramme de Cas d'utilisation lié « Gestionnaire »

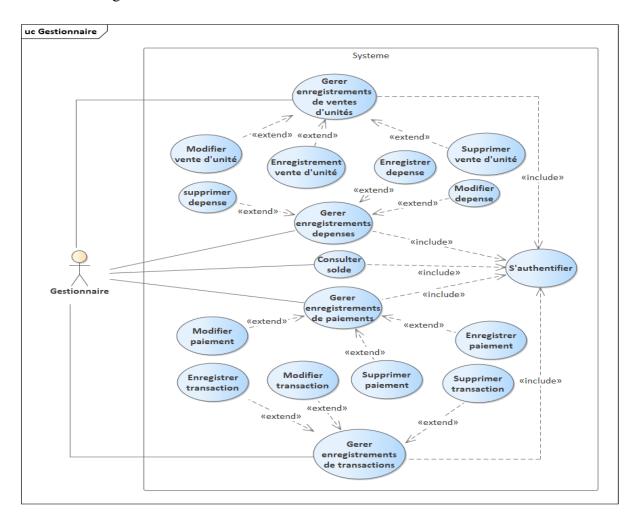


Figure 6 : CU lié à acteur « Gestionnaire »

Diagramme de Cas d'utilisation lié « Responsable »

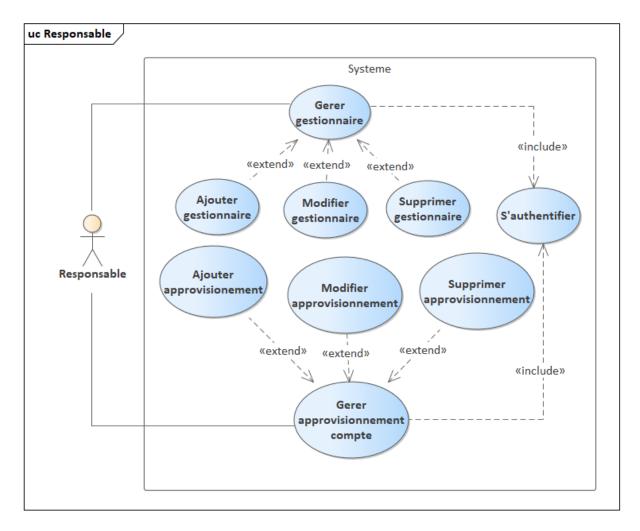


Figure 7 : CU lié à acteur « Responsable »

2.2. Description textuelle de quelques cas d'utilisation

Une description textuelle d'un cas d'utilisation (CU) décrit de manière détaillée et claire les différentes étapes, actions et interactions entre les acteurs et le système dans un scénario d'utilisation particulier.

Trois (3) étapes composent la description textuelle, les deux premières étant obligatoires et la troisième facultative : identifier le cas d'utilisation, le séquencement et les contraintes.

Cas d'utilisation « Enregistrer transaction »

Tableau 3 : CU « Enregistrer transaction »

Identification	* Nom du cas d'utilisation	on: Enregistrer transaction	
	* Objectif	: Permettre au gestionnaire d'enregistrer	
		une transaction	
	* Acteurs principaux	: Gestionnaires	
	* Date	: 29/06/2023	
	* Responsable	: OUEDRAOGO Issaka	
	* Version	: 1.0	
D (11/1	Le système est opération		
Précondition	Le gestionnaire doit êt	re authentifié.	
	1) Le gestionnaire clique su	ir le bouton nouveau	
Scénario		un pop-up qui contient le formulaire	
nominal	d'enregistrement.	on pop up qui contien is isimoimic	
	3) Le gestionnaire remplit et envoie le formulaire.		
	4) Le système vérifie les informations remplies.		
	5) Le système vérifie la disponibilité du montant.		
	6) Le système enregistre la transaction et notifie un message de succès.		
	A1: Les informations saisies sont incorrectes.		
Scénario	1) Le système affiche un message d'erreur.		
d'exception	2) Le Scenario reprend au point 2.		
	A2: Le montant est indisponible.		
	1) Le système affiche un message d'indisponibilité du montant.		
	2) Le Scenario reprend	l au point 2.	
Postcondition	Transaction enregistrée	en cas d'informations correctes et de montant	
	disponible.	and a morning converse of the montaint	
	r		

♣ Cas d'utilisation « S'authentifier »

Tableau 4 : CU « S'authentifier »

Identification	* Nom du cas d'utilisation * Objectif * Acteurs principaux * Date * Responsable * Version	 : S'authentifier : Permettre la connexion au système : Gestionnaires et Responsables : 24/06/2023 : OUEDRAOGO Issaka : 1.0
Précondition	 Le système est opérationnel. Les utilisateurs doivent être enregistrés dans la base de données et avoir un identifiant et un mot de passe. L'utilisateur se trouve sur la page de connexion. 	
Scénario nominal Scénario d'exception	 L'utilisateur saisit son identifiant et son mot de passe. Le système vérifie les informations saisies. Le système affiche la page d'Accueil de l'application. Al: Les informations sont incorrectes. Le système affiche un message d'erreur. Le Scenario reprend au point 1. 	
Postcondition	Authentification réussie en cas d'informations correctes.	

♣ Cas d'utilisation « Modifier paiement »

Tableau 5 : CU « Modifier paiement »

Identification	* Nom du cas d'utilisation	: Modifier paiement	
	* Objectif	: Permettre au gestionnaire de modifier	
		un paiement	
	* Acteurs principaux	: Gestionnaires	
	* Date	: 29/06/2023	
	* Responsable	: OUEDRAOGO Issaka	
	* Version	: 1.0	
	Le système est opérationn	nel.	
Précondition	• Le gestionnaire doit être a	authentifié.	
	1) Le gestionnaire clique sur le bouton modifier.		
Scénario	2) Le système récupère les informations dans la base de données affiche un		
nominal	pop-up qui contient le form	ulaire.	
	3) Le gestionnaire modifie et envoie le formulaire.		
	4) Le système vérifie les informations modifiées.		
	5) Le système vérifie la disponibilité du montant.		
	6) Le système enregistre la transaction et notifie un message de succès.		
	A1: Les informations saisies sont incorrectes.		
Scénario	1) Le système affiche un message d'erreur.		
d'exception	2) Le Scenario reprend au point 2.		
	A2: Le montant est indisponible.		
	1) Le système affiche un	message d'indisponibilité du montant.	
	2) Le Scenario reprend au point 2.		
	• Paiement modifié en	cas d'informations correctes et de montant	
Postcondition	disponible		

♣ Cas d'utilisation « Ajouter approvisionnement »

Tableau 6 : CU « Ajouter approvisionnement »

Identification	* Nom du cas d'utilisation : Ajouter approvisionnement	
	* Objectif	: Permettre au responsable d'approvisionner
		un compte
	* Acteurs principaux	: Respensable
	* Date	: 29/06/2023
	* Responsable	: OUEDRAOGO Issaka
	* Version	: 1.0
	 Le système est opérationnel. Le responsable doit être authentifié. 	
Précondition		
	1) Le responsable clique sur le bouton ajouter.	
Scénario	2) Le système affiche un pop-up qui contient le formulaire	
nominal	d'approvisionnement.	
	3) Le responsable remplit et envoie le formulaire.4) Le système vérifie les informations modifiées.	
	5) Le système enregistre la transaction et notifie un message de succès.	
	A1: Les informations saisies sont incorrectes	
Scénario	Le système affiche un message d'erreur	
d'exception	2) Le Scenario reprend au point 2	
Postcondition	 Approvisionnement effectué en cas d'informations correctes 	
1 OSCORULUM	Typrovisionnement effectue on eas a informations correctes	

2.3. Diagramme de séquence

Les diagrammes de séquence peuvent servir à illustrer les cas d'utilisation. Ils permettent de représenter la succession chronologique des opérations réalisées par un acteur et qui font passer d'un objet à un autre pour représenter un scénario. Nous allons représenter quelques diagrammes de séquence.

♣ Diagramme de séquence « S'authentifier »

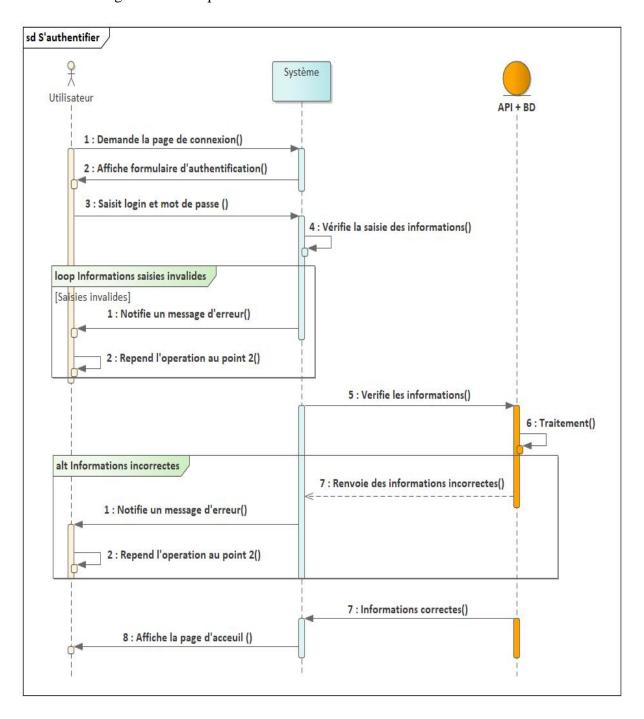


Figure 8 : Diagramme de séquence « S'authentifier »

Diagramme de séquence « Enregistrer transaction »

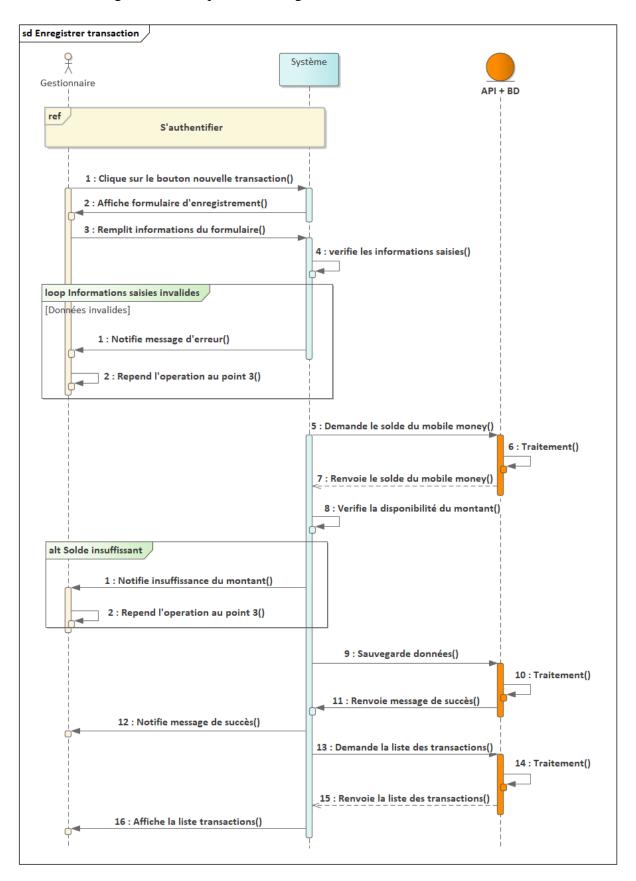


Figure 9 : Diagramme de séquence « Enregistrer transaction »

Diagramme de séquence « Modifier paiement »

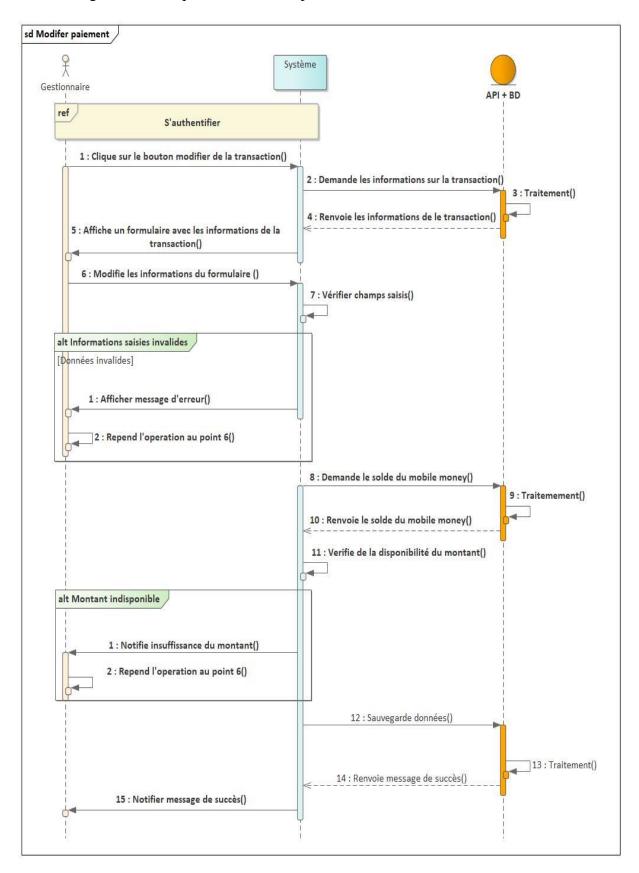


Figure 10 : Diagramme de séquence « Modifier paiement »

2.4. Diagramme d'activité

Le diagramme d'activité permet de représenter le déclenchement d'évènements en fonction des états du système et de modéliser des comportements du système. Il donne une vision des activités propres à une opération ou à un cas d'utilisation.

Une activité est une opération d'une certaine durée qui peut être interrompue. Dans ce cas, nous allons représenter ci-après quelques diagrammes d'activités.

♣ Diagramme d'activité « S'authentifier »

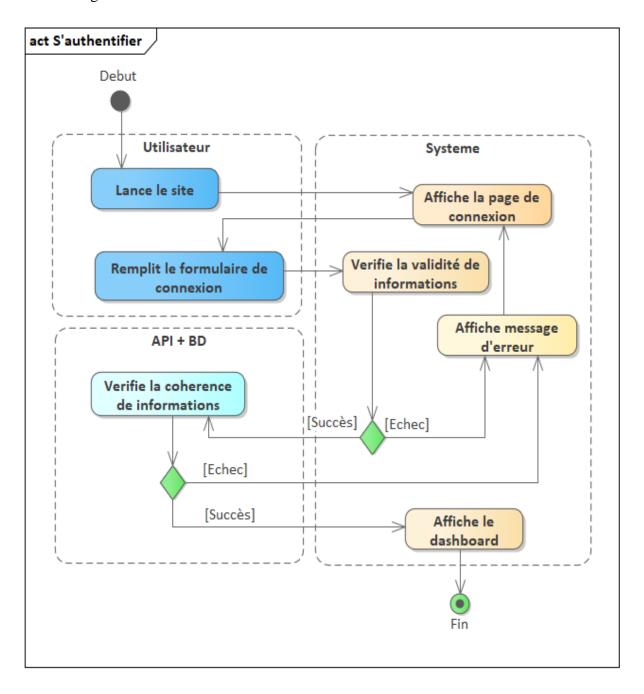


Figure 11 : Digramme d'activité « S'authentifier »

Diagramme d'activité « Enregistrer transaction »

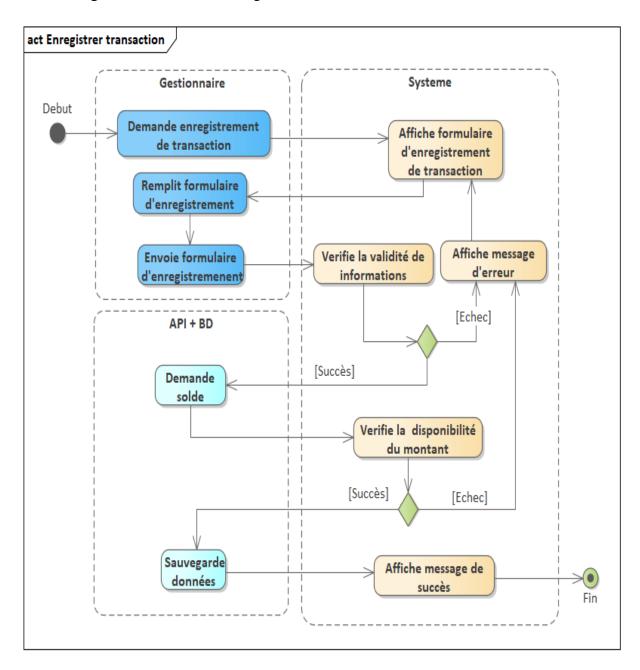


Figure 12 : Digramme d'activité « Enregistrer transaction »

2.5. Diagramme de classe

Un diagramme de classe est une représentation graphique statique qui décrit la structure d'un système logiciel ou d'une application en identifiant les classes, les attributs, les méthodes et les relations entre elles. Il permet de visualiser les entités du système, leurs caractéristiques et leurs interactions, facilitant ainsi la conception et la compréhension du système. Ainsi, notre diagramme se présente comme suit :

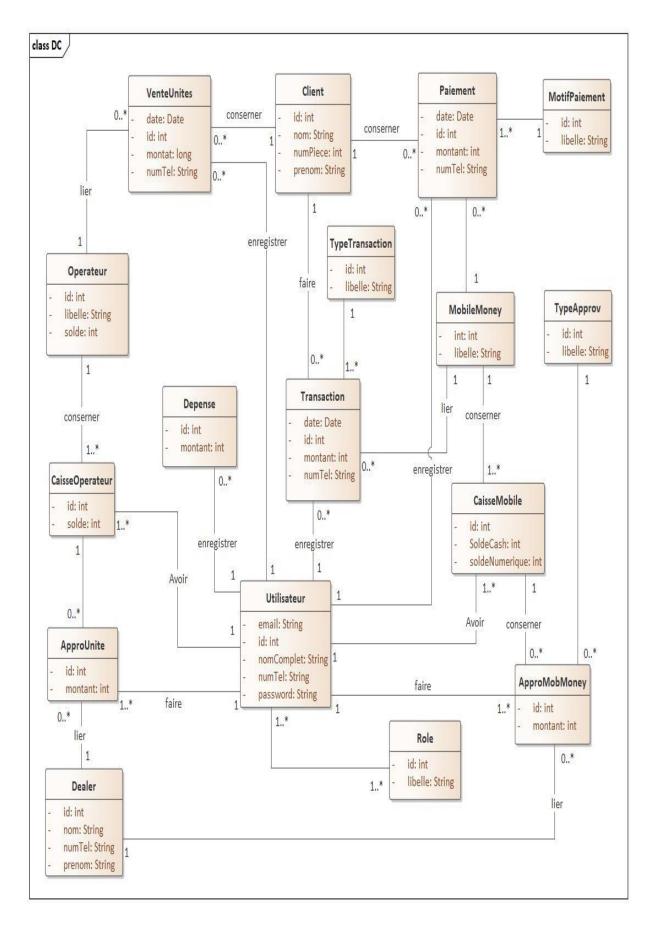


Figure 13 : Diagramme de classe du projet

CHAPITRE III: Réalisation

I. Outils de Développement

1. Ressources matériels utilisées

Pour la réalisation de notre application Web, Nous avons utilisé un ordinateur HP avec un processeur Intel(R) Core (TM) i5-6900U CPU @ 2.40GHz x 4, 8 Go de RAM et un système d'exploitation (OS) Ubuntu 22.04

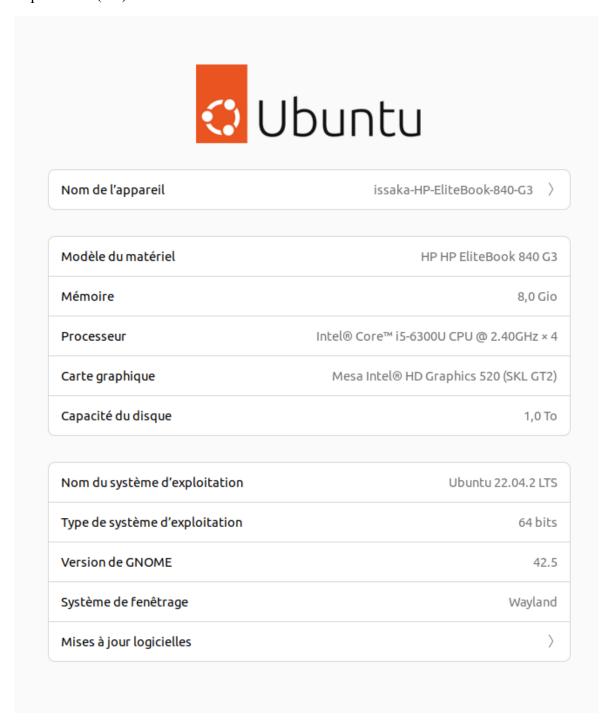


Figure 14 : Caractéristique de l'ordinateur portable

2. Langage de développement

Un langage de programmation est un ensemble de règles et de symboles utilisés pour écrire des instructions compréhensibles par un ordinateur. Il s'agit d'un moyen de communication entre un programmeur et la machine, permettant de définir les actions que l'ordinateur doit effectuer pour résoudre un problème donné.

2.1. Quelques langages de programmations

PHP

PHP « Hypertext Preprocessor », est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP. PHP est un langage impératif orienté objet. Il s'agit d'un langage de script interprété côté serveur. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.



Figure 15: Logo PHP

Source: https://chuidiang.org/images/thumb/f/f3/Logo_php.png/900px-Logo_php.png

Java

Java est un langage de programmation polyvalent et orienté objet. Il a été créé par Sun Microsystems (maintenant propriété d'Oracle) et est largement utilisé pour le développement d'applications web, d'applications mobiles, d'applications de bureau, de systèmes distribués et bien plus encore.



Figure 16: Logo Java

Source: https://upload.wikimedia.org/wikipedia/fr/thumb/2/2e/Java_Logo.svg/483px-Java_Logo.svg.png?20061227215918

❖ JavaScript

JavaScript est un langage de programmation de script du côté client utilisé par les développeurs pour concevoir des sites web interactifs. Les fonctions JavaScript peuvent permettre d'améliorer l'expérience utilisateur d'un site web, de mettre à jour des flux de médias sociaux à l'affichage d'animations et avoir cartes interactives.

Les fonctionnalités de JavaScript comprennent la manipulation de contenu HTML et CSS, la gestion des événements utilisateur (clics de souris, entrées clavier, etc.), la validation de formulaires, l'envoi de requêtes réseau (AJAX), la création de jeux, l'animation d'éléments, la création d'effets visuels, et bien plus encore.



Figure 17: Logo JavaScript

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/6/6a/JavaScript-logo.png?20120221235433

***** TypeScript

TypeScript est un langage de programmation open source développé par Microsoft. Il est un sur-ensemble de JavaScript qui ajoute un typage statique optionnel et des fonctionnalités avancées à JavaScript. Il offre des fonctionnalités avancées telles que l'héritage de classes, les interfaces, les modules, les génériques et les décorateurs. Il prend en charge les dernières spécifications ECMAScript (le standard sur lequel JavaScript est basé) et fournit des fonctionnalités supplémentaires qui ne sont pas encore présentes dans tous les navigateurs. Il permet de détecter certaines erreurs en amont et se compile en JavaScript pour une utilisation côté navigateur ou côté serveur à l'aide de NodeJS.



Figure 18: Logo TypeScript

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/4/4c/Typescript_logo_2020.
svg/768px-Typescript_logo_2020.svg.png?20221110153201

2.2. Etude comparative

Le choix d'un langage de programmation approprié est essentiel pour la création de code robuste, maintenable et évolutif. Il peut également offrir des bibliothèques et des Framework adaptés au projet, permettant ainsi de gagner du temps et d'optimiser les ressources. De plus, un langage populaire offre un large éventail de ressources communautaires, de forums et de documentation, ce qui facilite le support et la résolution des problèmes.

Tableau 7 : Etude comparée de quelques langages de programmation

Langage de	Caractéristiques	Avantages	Inconvénients
programmation			
Java	- Plateforme	- Portabilité	- Utilisation de la
	indépendante	- Vaste bibliothèque	mémoire
	- Orienté objet	standard	- Lourdeur syntaxique
	- Langage compilé	- Gestion	- Besoin de la Machine
	et interprété	automatique de la	Virtuelle Java (JVM)
		mémoire	- Problèmes de sécurité
		- Communauté et	
		écosystème	
		- Sécurité	
		- Evolutivité	
JavaScript	- Langage côté	- Exécution côté	- Non Sécurisé
	client	client	- Faibles performances
	- Syntaxe flexible	- Flexibilité	- Problème de gestion
	- Typage dynamique	- Intégration facile	des erreurs
		avec HTML et CSS	
		- Vaste écosystème et	
		bibliothèques	
TypeScript	- Type statique	- Détection précoce	- Complexité accrue
	- Orienté objet	des erreurs	- Configuration et
	- Trans-compilation	- Meilleure	outillage
	- Prise en charge des	évolutivité	- Compilation
	dernières	- Outil de	nécessaire
	fonctionnalités	développement	
	ECMAScript	solide	
PHP	- Syntaxe simple	- Gestion facile du	- Ne convient pas aux
	- Large	code	grandes applications
	compatibilité avec	d'apprentissage	- Mauvaise méthode de
	certains serveurs	- Large	gestion des erreurs
	- Modules de	communauté de	
	connexion aux	développeurs	

	bases de données	- Intégration facile	- Incapable de gérer un
	intégrés	avec le web	grand nombre
		- Large support de	d'applications
		bases de données	
		- Performance	
		- Large adoption	
		- Stable	
Python	- Multiplateforme	- Syntaxe claire et	- Utilisation intensive
	- Large bibliothèque	lisible	du processeur
	standard	- Grande	- Absence de typage
	- Langage interprété	communauté et	statique strict
	- Orienté objet	bibliothèques riches	- Gestion de la mémoire
		- Polyvalence	
		- Portabilité	
		- Productivité élevée	
		- Intégration facile	

2.3. Choix de langage de développement

A travers cette étude comparative, nous retenons Java pour développer le Backend et TypeScript pour la partie vue. En effet, Java est un langage de programmation puissant conçu pour être sûr, inter-plateformes et international et permet de réduire le temps de développement d'une application grâce à la réutilisation du code développé et TypeScript offre plusieurs avantages. Il ajoute une couche de typage statique à JavaScript, améliorant la qualité du code et facilitant la détection des erreurs. Il permet également une meilleure prise en charge des outils de développement et fournit une documentation plus claire et précise pour les développeurs.

3. Architecture Logicielle

L'architecture logicielle désigne la structure organisationnelle d'un système informatique, définissant les composants, les relations et les principes de conception. Elle guide le développement et facilite la compréhension, la maintenance et l'évolutivité du logiciel.

3.1. Quelques architectures logicielles

❖ Model-View-Controller (MVC)

Le Modèle-Vue-Contrôleur (en abrégé MVC, de l'anglais Model-View-Controller) est un modèle architectural utilisé dans le développement logiciel. Il permet de séparer les différentes responsabilités des composants de l'application et de les organiser de manière claire et cohérente. Il divise une application en trois composants principaux : le modèle (représentant les données et la logique métier), la vue (gérant l'interface utilisateur) et le contrôleur (facilitant les interactions entre le modèle et la vue).

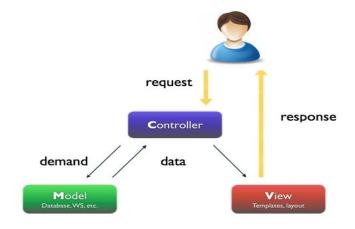


Figure 19 : Model-View-Controler (MVC)

Source: https://www.tresfacile.net/wp-content/uploads/2023/04/mvc-modele-vue-controleur.png

❖ Model-View-Presenter (MVP)

Le Modèle-Vue-Présentation (en abrégé MVP, de l'anglais Model-View-Presenter) est un modèle architectural qui facilite la conception de logiciels. Il divise une application en trois composants : le modèle (représentant les données et la logique métier), la vue (responsable de l'interface utilisateur) et le présentateur (gérant les interactions entre le modèle et la vue).

MVP est une dérivation du modèle MVC dans lequel le contrôleur est remplacé par le présentateur qui organise les données à afficher dans la vue.

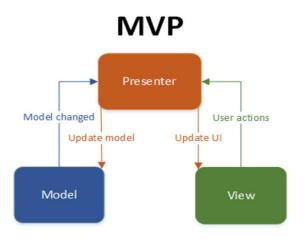


Figure 20: Model-View-Presenter (MVP)

 $\label{eq:source:https://miro.medium.com/v2/resize:fit:1100/format:webp/1*TuWeZzR14MmB-RBbjtZl-A.png$

❖ Model-View-View Model (MVVM)

Le Modèle-Vue-Vue Modèle (en abrégé MVVM, de l'anglais Model-View-View Model) est une approche de conception logicielle qui vise à séparer les données d'une application (le modèle) de sa présentation (la vue). Le View Model est une sorte de pont entre le modèle et la vue, il fournit une interface que la vue peut utiliser pour accéder aux données du modèle de manière appropriée. Cette architecture permet de rendre les vues plus indépendantes du modèle.

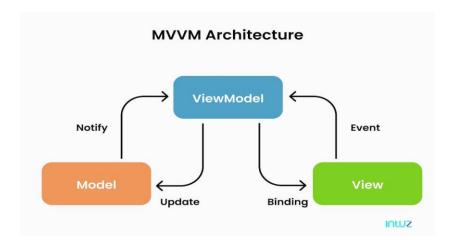


Figure 21: Model-View-View Model (MVVM)

Source: https://bachkhoa-npower.vn/wp-content/uploads/2022/10/so-sanh-mvvm-mvc-mvp-840x450.jpg

3.2. Etude comparative

Le choix d'une architecture logicielle appropriée lors du développement permet de favoriser la maintenabilité, la modularité, la réutilisabilité et la scalabilité du système, ce qui facilite les évolutions futures, réduit les coûts de maintenance et offre une meilleure expérience utilisateur.

Tableau 8 : Etude comparée de quelques architectures logicielles

Architecture Logicielle	Avantages	Inconvénients
MVC	- Séparation des	- Complexité initiale
	responsabilités	- Surcharge de
	- Facilité de maintenance	communication
	- Tests unitaires facilités	
	- Facilité de collaboration	
	- Réutilisabilité du code	
MVP	- Séparation des	- Complexité accrue
	responsabilités	- Surcharge du Présentateur
	- Tests unitaires facilités	
	- Réactivité de l'interface	
	utilisateur	
MVVM	- Liaison de données	- Complexité initiale
	bidirectionnelles	- Surcharge de la Vue
	- Testabilité améliorée	Modèle
	- Réutilisabilité des Vues et	
	des Vue Modèles	

3.3. Choix de l'architecture logicielle

Apres cette étude, nous optons l'architecture Model View Controler (MVC) car un choix populaire pour le développement d'applications web. De plus il offre une séparation claire des préoccupations, facilitant la gestion du code, la maintenance et l'évolutivité. Le modèle permet d'organiser l'application en trois composants distincts : le modèle (pour les données), la vue (pour l'interface utilisateur) et le contrôleur (pour la logique de gestion). Cette approche modulaire facilite la collaboration entre les développeurs, accélère le développement et permet des mises à jour plus faciles en minimisant les interactions directes entre les composants.

4. Framework

4.1. Backend

Un Framework Backend est un outil logiciel qui peut automatiser certains aspects du processus de développement, le rendant plus facile et plus rapide. Plus précisément, un Framework backend est une bibliothèque de modèles et d'outils qui peuvent aider le développeur à construire l'architecture de son site Web, de son application Web, de son application mobile ou de son logiciel.

4.1.1. Quelques Framework Java

Spring Boot

Spring Boot est un Framework Java open source utilisé pour créer rapidement et avec une configuration minimale des applications autonomes de niveau production. Il est construit sur le Framework *Spring* et offre une expérience de développement simplifié pour les développeurs Java.

Spring Framework est l'un des Framework backend les plus robustes et les plus polyvalents. Il contient plusieurs modules différents qui fournissent aux programmeurs une variété d'outils d'application. Ce cadre utilise les concepts d'injection de dépendance et d'inversion de contrôle pour configurer les composants de l'application dans des couplages lâches, aidant à donner à l'application plus de flexibilité dans ses réponses.

Spring fournit également un cadre d'accès aux données qui atténuent de nombreuses difficultés liées à l'utilisation d'une base de données d'application, notamment :

- La gestion des ressources et déballage ;
- La gestion des exceptions ;

➤ La participation aux transactions.



Figure 22: Logo Spring Boot

Source: https://www.nicepng.com/png/detail/31-314820_logo-spring-spring-framework-logo-svg.png

Apache Struts

Le Framework Struts est un Framework d'application Web open source utilisé pour créer des applications Web Java Enterprise Edition rapidement et efficacement. Il utilise et étend encore l'API Java Servlet pour promouvoir l'architecture modèle-vue-contrôleur (MVC).

Struts excelle en tant qu'outil complémentaire à d'autres Framework en raison de sa flexibilité et de son extensibilité. Cela fait de Struts un ajout productif à une variété de scénarios de développement Web. Sa polyvalence permet aux développeurs de l'utiliser pour tout, des sites Web de portefeuille simples aux applications riches en fonctionnalités nécessitant une validation utilisateur hautement sécurisée et des mesures de sécurité des informations.



Figure 23: Logo Struts

Source: https://seeklogo.com/images/S/struts-logo-8759EBE252-seeklogo.com.png

❖ Play

Play est un Framework Java réactif basé sur le Web et les mobiles pour le développement d'applications Java hautement évolutives. Il est open-source et suit le célèbre modèle architectural MVC. Play permet aux développeurs de créer des applications Java légères et

adaptées au Web pour les ordinateurs de bureau et les appareils mobiles. Il est utilisé par de grandes entreprises telles que LinkedIn, Verizon, Samsung, entre autres.



Figure 24: Logo Play

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Play_Framework_logo.svg.png

4.1.2. Etude comparative de quelques Framework Java

Un Framework Web backend robuste et efficace permet d'assurer la sécurité des données, la scalabilité, la performance et la fiabilité de l'application. Il constitue le socle sur lequel repose l'expérience utilisateur et garantit une gestion fluide des transactions et des interactions avec l'application. Ainsi, nous ferons une étude comparative, sur les Framework backend *Java* afin de choisir le mieux adapté dans notre contexte.

Tableau 9 : Etude comparée de quelques Framework Backend

Framework	Avantages	Inconvénients
Backend		
Sprint Boot	- Gestion des dépendances plus simple	- Complexité excessive
	- Autoconfiguration par défaut	- Surcharge de dépendances
	- Serveur web intégré	- Perte de contrôle fine
	- Des métriques et contrôles de santé de	- Courbe d'apprentissage initiale
	l'application	
	- Une configuration avancée externalisée	
	- Facilité de configuration	
	- Dépendances simplifiées	
	- Intégration avec Spring	
Struts	- Validation des données	- Apprentissage difficile
	- Intégration avec d'autres technologies	- Problème de Maintenance
	- Communauté active	- Intégration avec les
	- Validation des données	technologies modernes
	- Gestion des formulaires	
Play	- Communauté active	- Apprentissage difficile
	- Système de routage puissant	- Taille de l'application
	- Facilité de test	
	- Évolutivité	
	- Architecture réactive	
	- Programmation basée sur les routes	

4.1.3. Choix Backend

Après cette étude comparative, il ressort que le backend le mieux adapter à notre projet est Spring Boot. En effet, Spring Boot est un Framework de développement populaire car il simplifie la création d'applications robustes et évolutives. Il offre une configuration automatique, une gestion des dépendances simplifiée et des fonctionnalités prêtes à l'emploi telles que la sécurité, l'accès aux bases de données et les API REST. Spring Boot permet de gagner du temps et de se concentrer sur le développement des fonctionnalités métier.

4.2. Frontend

Le Frontend est la partie visible et interactive de l'application, responsable de l'interface utilisateur et de l'interaction avec les utilisateurs. Il facilite le travail des développeurs Web en leur fournissant des modules de code réutilisables, des technologies frontales standardisées et des blocs d'interface prêts à l'emploi qui facilitent le développement d'applications et d'interfaces utilisateur sans qu'il soit nécessaire de coder chaque fonction ou objet à partir de zéro.

Ainsi, un frontend bien conçu offre une navigation fluide, une présentation attrayante des données et une interactivité réactive. Il permet de créer des expériences utilisateur modernes et engageantes.

4.2.1. Quelques Frontend

ReactJS

React.JS, communément appelé simplement React, est une bibliothèque JavaScript open source utilisée pour construire des interfaces utilisateurs. Il permet aux développeurs de créer de grandes applications qui peuvent modifier les données, sans avoir à recharger la page. Créé par Facebook en 2013, React contient une collection d'extraits de code JavaScript réutilisables utilisés pour la construction d'interface utilisateur (UI) appelés composants. Chaque application web React est composée de composants réutilisables qui constituent des parties de l'interface utilisateur.

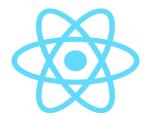


Figure 25: Logo React.js

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/a/a7/React-icon.svg.png

AngularJS

AngularJS est un Framework d'applications Web Frontend open source basé sur JavaScript. Il a été développé par Google et publié en 2010. AngularJS est conçu pour simplifier le développement d'applications Web dynamiques en fournissant un cadre structuré pour la création d'interfaces utilisateur interactives et basées sur les données.



Figure 26 : Logo AngularJS

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/c/cf/Angular_full_color_log
o.svg/2048px-Angular_full_color_logo.svg.png

Vue.IS

VueJS est un Framework JavaScript open source progressif et évolutif pour la construction d'interfaces utilisateur. Il propose une approche simple et flexible pour la création d'applications web interactives et réactives. Vue.js est souvent considéré comme facile à apprendre et à intégrer avec d'autres bibliothèques ou projets existants.



Figure 27: Logo VueJS

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/9/95/Vue.js_Logo_2.svg/23
67px-Vue.js_Logo_2.svg.png

4.2.2. Etude comparative

Un Frontend soigneusement choisi, on s'assure de répondre aux besoins des utilisateurs et d'offrir une application web performante et conviviale.

Tableau 10 : Etude comparée de quelques Framework Frontend

Frontend	Caractéristiques	Avantages	Inconvénients
ReactJS	 Composants réutilisables Concept Virtual DOM Syntaxe JSX Approche de liaison de données unidirectionnel 	 Composants réutilisables Performance élevée Réactivité Compatibilité avec d'autres technologies 	 Apprentissage difficile Configuration complexe Mises à jour fréquentes
AngularJS	 Approche de liaison de données bidirectionnelle Modèle MVC Injection de dépendances Testabilité Routages Services 	 Large écosystème et communauté active Prise en charge de TypeScript Performance Réutilisabilité des composants Gestion avancée des formulaires Structure et organisation 	 Taille de l'application Maintenance difficile Apprentissage un peu compliquée
VueJS	 Approche de liaison de données bidirectionnelle Directives Gestion des animations et transitions Écosystème extensible 	 Composants réutilisables Flexibilité Réactivité Performances optimisées 	 Apprentissage un peu compliquée Documentations moins complètes Écosystème moins étendu Complexité croissante avec des applications plus grandes

4.2.3. Choix du frontend

Suite à l'étude comparative, nous choisissons AngularJS car il utilise du TypeScript. De plus, il est un Framework JavaScript populaire pour le développement frontend en raison de sa structure Model-View-Controler (MVC), de sa gestion simplifiée des données et de sa capacité à créer des applications interactives. Il offre des fonctionnalités telles que la liaison de données bidirectionnelle, la manipulation du DOM « Document Object Mode », la modularité et la réutilisabilité du code, facilitant ainsi le développement rapide d'applications web robustes et évolutives.

5. Système de gestion de bases de données (SGBD)

Un système de gestion de base de données (SGBD) est le logiciel qui permet à un ordinateur de stocker, récupérer, ajouter, supprimer et modifier des données. Un SGBD gère tous les aspects primaires d'une base de données, y compris la gestion de la manipulation des données, comme l'authentification des utilisateurs, ainsi que l'insertion ou l'extraction des données. Un SGBD définit ce qu'on appelle le schéma de données ou la structure dans laquelle les données sont stockées.

Le SGBD gère trois choses importantes : les données, le moteur de base de données qui permet d'accéder aux données, de les verrouiller et de les modifier, et le schéma de base de données, qui définit la structure logique de la base de données. Ces trois éléments fondamentaux contribuent à assurer la concomitance, la sécurité, l'intégrité des données et l'uniformité des procédures administratives.

5.1.1. Quelques SGBD

❖ Oracle Database

Oracle Database est un système de gestion de base de données relationnelle. Il a été créé par Oracle Corporation et publié pour la première fois en 1980. Il est implémenté en C et C++ et est disponible sur la majorité des systèmes d'exploitation (Linux, Solaris, OSX, Windows). Il est également compatible avec un large éventail de langages de programmation, dont PHP. Il adhère au principe transactionnel ACID (Atomicité, Cohérence, Isolation et Durabilité).

L'approche ACID (Atomicité, Cohérence, Isolation, Durabilité) permet d'assurer l'intégrité des données au sein d'une base de données.



Figure 28 : Logo Oracle Database

Source: https://allvectorlogo.com/img/2017/02/oracle-database-logo.png

MariaDB

MariaDB est un système de gestion de base de données relationnelle (SGDBR) open source largement utilisé pour stocker et gérer des données structurées. C'est l'une des bases de données les plus populaires au monde et elle est connue pour son évolutivité, sa fiabilité et ses performances. MariaDB est un programme C et C++ qui s'exécute sur FreeBSD, Linux, Solaris, OS X et Windows. De plus, il prend en charge un large éventail de langages, y compris PHP, et adhère au principe ACID des transactions. Elle a également été développée par une société suédoise MySQLAB et a été achetée en 2008 par Sun Microsystems et rachetée en 2009 par Oracle Corporation.



Figure 29 : Logo MariaDB

Source: https://mariadb.com/wp-content/uploads/2019/11/mariadb-logo-vert_blue-transparent.png

❖ PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle (SGBDR) opensource, puissant et extensible, prenant en charge les fonctionnalités avancées telles que les transactions ACID, la réplication, la gestion des index et les requêtes complexes. Il est largement utilisé pour les applications web et d'entreprise.

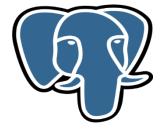


Figure 30 : Logo PostgreSQL

Source: https://w7.pngwing.com/pngs/358/849/png-transparent-postgresql-database-logo-database-symbol-blue-text-logo-thumbnail.png

5.1.2. Etude comparative

Un SGBD bien choisi permet de bénéficier d'une gestion efficace, sécurisée et performante des données, tout en offrant des fonctionnalités avancées telles que la cohérence, l'intégrité, la scalabilité et la compatibilité avec le langage SQL. Cela permet aux développeurs de se concentrer sur la logique métier de l'application sans avoir à gérer directement les détails de stockage et de manipulation des données.

Tableau 11: Tableau comparative de quelques SGBD

SGBD	AVANTAGES	INCONVENIENTS
MariaDB	- Rapide	- Difficulté à gérer les
	- Facile à utiliser	données non structurées
	- Open Source	- Manque de fonctionnalités
	- Connexion et Sécurité	avancées
	- Portabilité	- Gestion des connexions
	- Distribution ouverte	concurrentes
Oracle	- Forte compatibilité avec toutes les	- Coût élevé
Database	plateformes et applications	- Complexité
	- Fiabilité et robustesse	- Consommation de
	- Évolutivité	ressources
	- Performance élevée	- Courbe d'apprentissage
	- Sécurité avancée	- Dépendance à un
	- Fonctionnalités avancées	fournisseur unique
	- Intégration avec d'autres produits	- Portabilité limitée
	Oracle	
	- Grande popularité	
	- Fonctions de sécurité et de protection	
	des données fiables	
PostgreSQL	- Open Source	- Complexité
	- Intégrité des données	- Limitations de plateforme
	- Conformité ACID	- Apprentissage difficile
	- Concurrence et évolutivité	- Frais généraux de
	- Réplication et haute disponibilité	maintenance

5.1.3. Choix du SGBD

Apres cette étude comparative, notre choix est porté sur MariaBD car il est open source et gratuit. De plus, il est un système de gestion de base de données populaire en raison de sa fiabilité, de sa performance et de sa simplicité d'utilisation. Il offre une large gamme de fonctionnalités, prend en charge le langage SQL, est compatible avec de nombreux langages de programmation et dispose d'une grande communauté de développeurs. MariaBD est un choix solide pour les applications web, les systèmes de gestion de contenu et les solutions d'entreprise.

6. Autres technologies utilisées

❖ Visual Studio Code (VS Code)

VS Code, abréviation de Visual Studio Code, est un éditeur de code source développé par Microsoft. Il est gratuit, open source et multiplateforme, compatible avec Windows, MacOs et Linux. VS Code offre une interface utilisateur intuitive et légère, ainsi que de nombreuses fonctionnalités avancées telles que la coloration syntaxique, l'auto-complétion, le débogage, la gestion de versions, les extensions et bien plus encore. Il est largement utilisé par les développeurs pour écrire du code dans différents langages de programmation.



Figure 31 : Logo VS Code

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/9/9a/Visual_Studio_Code_1
.35_icon.svg/2048px-Visual_Studio_Code_1.35_icon.svg.png

❖ PhpMyAdmin

PhpMyAdmin est une application web open-source écrite en PHP, conçue pour administrer facilement des bases de données MySQL. Il fournit une interface conviviale et graphique permettant aux utilisateurs de gérer leurs bases de données, tables, colonnes et enregistrements. Avec PhpMyAdmin, les utilisateurs peuvent exécuter des requêtes SQL, importer et exporter des données, créer des utilisateurs et gérer les privilèges d'accès à la base de données. C'est un outil couramment utilisé par les développeurs et les administrateurs de bases de données pour simplifier la gestion de MySQL.



Figure 32: Logo phpMyAdmin

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/4/4f/PhpMyAdmin_logo.sv g/2560px-PhpMyAdmin_logo.svg.png

❖ NodeJS

NodeJS est un environnement d'exécution open source basé sur le moteur JavaScript V8 de Google Chrome. Il permet d'exécuter des applications JavaScript côté serveur, ce qui facilite le développement d'applications web hautement évolutives et performantes. Grâce à son modèle asynchrone et non bloquant, Node.js peut gérer de manière efficace des connexions simultanées, offrant ainsi une scalabilité optimale pour les applications web en temps réel.



Figure 33: Logo NodeJS

Postman

Postman est un outil de développement d'API (Application Programming Interface) largement utilisé. Il permet aux développeurs de tester, de déboguer et de documenter facilement les API. Postman offre une interface conviviale pour envoyer des requêtes HTTP, explorer les réponses, gérer les autorisations, créer des collections d'API et automatiser les tests. Il est largement utilisé dans le développement d'applications web, mobiles et d'autres services qui communiquent via des API.



Figure 34 : Logo Postman

Source: https://img.uxwing.com/wp-content/themes/uxwing/download/brands-social-media/postman-icon.png

GitHub

GitHub est une plateforme de développement collaboratif basée sur Git. Elle permet aux développeurs de travailler ensemble sur des projets, de partager et de gérer le code source. GitHub offre des fonctionnalités telles que le contrôle de version, la gestion des problèmes, le suivi des modifications, les demandes de fusion, les actions automatisées et la collaboration avec d'autres développeurs. C'est un outil largement utilisé dans l'industrie du développement logiciel pour faciliter la collaboration et le partage de code.



Figure 35 : Logo GitHub

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/9/91/Octicons-mark-github.svg.png?20180806170715

***** Enterprise Architect

Enterprise Architect est un outil de modélisation et de gestion des architectures d'entreprise. Il permet aux organisations de créer des modèles visuels et conceptuels de leurs processus, systèmes, infrastructures et données. Cet outil offre des fonctionnalités telles que la modélisation UML, la cartographie des processus métier, la gestion des exigences, la simulation, la génération de documentation et la collaboration en équipe. Enterprise Architect

facilite la conception, la planification et la communication des architectures d'entreprise, favorisant ainsi une meilleure compréhension et une prise de décision éclairée.



Figure 36: Logo Enterprise Architect

Source: https://e7.pngegg.com/pngimages/703/439/png-clipart-enterprise-architecture-application-lifecycle-management-architecture-sparx-systems-others-miscellaneous-purple.png

Figma

Figma est un outil de conception d'interface utilisateur (UI) et d'expérience utilisateur (UX) basé sur le cloud. Il permet aux équipes de conception de collaborer en temps réel sur des projets, de créer des maquettes interactives, des prototypes et des designs d'interface utilisateur. Figma offre des fonctionnalités avancées telles que le partage de commentaires, le versioning, les composants réutilisables et la possibilité de créer des spécifications de design. Il est largement utilisé dans l'industrie pour créer des designs d'interface utilisateur professionnels et fonctionnels.



Figure 37: Logo Figma

Source: https://upload.wikimedia.org/wikipedia/commons/thumb/3/33/Figma-logo.svg.png

GanttProject

GanttProject est un logiciel de gestion de projet open-source qui permet de créer des diagrammes de Gantt pour planifier et suivre les tâches d'un projet. Il offre des fonctionnalités telles que la création de diagrammes de dépendance, la gestion des ressources, le suivi du calendrier, l'exportation de rapports, etc. GanttProject est largement utilisé pour la gestion de projets de petite à moyenne envergure.



Figure 38 : Logo GanttProject

Source: https://images.sftcdn.net/images/t_app-icon-m/p/3a64ed76-96d8-11e6-a242-00163ed833e7/3970617297/ganttproject-GanttProject-icn.png

Zotero

Zotero est un logiciel de gestion de références bibliographiques et de création de bibliothèques de recherche. Il permet aux chercheurs, étudiants et professionnels de collecter, organiser et citer facilement leurs sources bibliographiques. Zotero propose également des fonctionnalités de collaboration et d'intégration avec les traitements de texte, simplifiant ainsi le processus de rédaction et de citation dans les travaux académiques.



Figure 39 : Logo Zotero

Source: https://www.zotero.org/support/_media/logo/zotero_512x512x32.png

II. Estimation du coût

L'estimation du coût est essentielle pour planifier et contrôler efficacement un projet. En fournissant une estimation réaliste des dépenses requises, elle permet d'allouer les ressources de manière appropriée, de prendre des décisions éclairées et d'assurer la rentabilité du projet. Une estimation précise des coûts est donc cruciale pour assurer le succès et la viabilité d'un projet.

1. Quelques approches d'estimation des coûts

❖ Approche par le nombre de lignes de code

L'estimation basée sur le nombre de lignes de code (LOC) est une méthode d'estimation des coûts de projet qui utilise le nombre de lignes de code comme mesure pour prédire la charge de travail et les coûts associés. Cette approche suppose généralement qu'il existe une corrélation entre la quantité de code à développer et les efforts nécessaires pour le créer.

Approche par le nombre de points de fonctions

L'estimation basée sur les points de fonction est une méthode d'estimation des coûts de projet qui évalue la taille fonctionnelle du système en utilisant des points de fonction. Les points de fonction sont attribués en fonction des entrées, sorties, requêtes et fichiers de données du système. Cette approche permet une estimation objective et indépendante de la technologie utilisée, en se concentrant sur la valeur fonctionnelle plutôt que sur la quantité de code. Elle peut fournir une estimation plus précise en prenant en compte les besoins et la complexité fonctionnelle du projet.

Approche sur le nombre de fonctionnalités

L'estimation basée sur les fonctionnalités est une approche d'estimation des coûts de projet qui se concentre sur l'analyse et l'estimation des coûts en se basant sur les différentes fonctionnalités ou caractéristiques requises par le projet. Elle nécessite une compréhension détaillée des exigences fonctionnelles, permettant ainsi une estimation plus précise des coûts en prenant en compte la complexité et la valeur ajoutée de chaque fonctionnalité.

2. Etude comparative de quelques approches d'estimation des coûts

Une estimation du coût du projet est une pratique fondamentale pour assurer la viabilité financière du projet, prendre des décisions éclairées, gérer les attentes et contrôler les coûts tout au long de sa réalisation. Ainsi, nous allons faire une étude comparée afin de choisir l'approche le mieux adaptés pour notre projet.

Tableau 12 : Etude comparée de quelques approches d'estimation de coûts

Approche	Avantages	Inconvénients
Nombre de	- Simplicité	- Manque de précision
ligne code	- Rapidité	- Ignorance des autres facteurs
		- Dépendance de la taille du code
		- Variabilité de la productivité
Nombre de	- Adaptabilité aux	- Analyse détaillée requise
fonctionnalités	changements	- Complexité des fonctionnalités
	- Communication efficace	interdépendantes
	- Précision améliorée	- Sensibilité aux changements
		- Dépendance aux compétences
		techniques
Nombre de	- Objectivité sur la taille	- Complexité
points de	fonctionnelle du système	- Temps et effort
fonction	- Prédictibilité du résultat	- Dépendance des données
	- Comparabilité sur les projets	- Adaptabilité limitée
	de manière plus juste	-
	- Estimation de la productivité	

3. Choix de notre approche

Apres cette étude comparée, il ressort que, l'approche propice est l'approche par ligne de code appelé aussi méthode COCOMO car cette approche est facile à comprendre et à appliquer et peut donner une estimation rapide en phase préliminaire du projet.

4. Estimation du coût

4.1. Méthode COCOMO et coût de prestation de l'équipe

La méthode COCOMO (La méthode COnstructive COst MOdel) a été proposée par Barry W. Boehm en 1981 en fonction des hypothèses suivantes :

- Il est facile pour un informaticien d'estimer le nombre de ligne du code source ;
- La complexité d'écriture d'un programme est la même quel que soit le langage de programmation.

Ce modèle utilise des instructions, à savoir les milliers de codes d'instructions comme unité de temps et le mois-homme comme unité de coût. Il utilise le KLSI comme unité de mesure (Kilo Delivered Source Instruction). Les formules utilisées pour le calcul du coût sont les suivants :

- Charge en Mois/Homme = a [Kisl]b;
- ➤ Avec : Kisl= kilo instruction source livrée ;
- ➤ Délai = c [Charge] d;
- Taille moyenne d'équipe = Charge / Délai ;

Les paramètres a, b, c et d dépendent de type de projet auquel appartient le projet. Soit I la taille du projet, on a :

- ➤ Un projet est dit simple si I < 50 Kisl, spécifications stables, petite équipe ;
- ➤ Un projet est moyen si 50 Kisl <= I < 300 Kisl, spécifications stables, petite équipe ;
- ➤ Un projet est complexe si I >300 Kisl, grande équipe.

Tableau 13 : Modèle COCOMO

Type de projet	Charge en mois-personne	Délais en mois
Simple	Charge = $2.4(kis1)^{1.05}$	D=2.5(charge) ^{0.38}
Moyen	Charge = $3 \text{ (kis 1)}^{1,12}$	D=2,5 (Charge) ^{0,35}
Complexe	Charge = $3.6 \text{ (kis 1)}^{1.2}$	D=2,5 (Charge) ^{0,32}

Coût de développement

Nous estimons notre projet a 10 Kisl, ce qui le classe dans la catégorie de projet simple. Ainsi, on peut estimer le cout du développement comme suit :

- Arr Charge en Mois/Homme = 2.4(10)^{1,05} \approx 26,93;
- \triangleright Délai = 2,5(26,93)^{0,38} = 8.74 \approx 9;
- ➤ Taille moyenne de l'équipe = $26,93/8.74 = 3.08 \approx 3$.

On en déduit qu'il faudrait une équipe de 3 personnes travaillant pendant 9 mois pour réaliser le projet. En prenant 300 000 F CFA comme le salaire de base d'un développeur au Burkina nous pouvons donc calculer le cout de ressource humaine pour le développement du projet par : Coût des ressources humaines = $300.000 \times 3 \times 9 = 8.100.000 F$ CFA

- Coût de formation
 - Tarif = 6.000 F CFA / utilisateur / jour ;
 - Nombre d'utilisateur = 5 personnes ;
 - Durée de la formation = 2 jours.

Coût de formation = $2 \times 5 \times 6.000 = 60.000$ F CFA

Coût du matériel utilisé

Un poste de travail de type et de marque HP Core i7- 6820HQ 16Go Ram SSD 512 Go 1To HDD à **575.000 FCFA**.

- Coût du matériel déploiement
 - Acquisition d'un serveur pour l'hébergement de l'application en local : 4.000.000
 FCFA ;
 - Un onduleur de type Tripp Lite Smart UPS LCD 1500VA Tower Line-Interactive 230V à **300.000 FCFA**.

Cout total du matériel = 3.000.000 + 300.000 = 3.300.000 F CFA

4.2. Tableau récapitulatif

La conception de notre application a nécessité d'autres coût énumérés dans le tableau suivant :

Tableau 14:Tableau récapitulatif

Désignation	Coût (F CFA)
Matériel utilisé	575.000
Matériel de	3.300.000
déploiement	
Développement	8.100.000
Formation	60. 000
Total	12.035.000

4.3. Caractéristiques d'utilisation

Pour utiliser cette application, nous avons besoin de :

- Connexion haute débit ;
- Ordinateur d'au moins un processeur Core (TM) i3 CPU @ 1.50GHz x 4, 4 Go de RAM.

III. Description d'application

1. Page de connexion

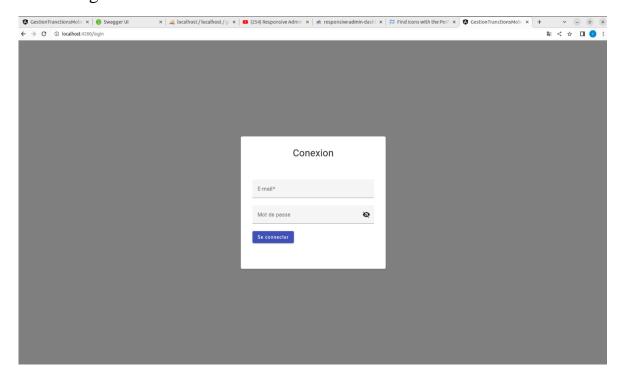


Figure 40 : Page de connexion

2. Page d'accueil

2.1.Administrateur

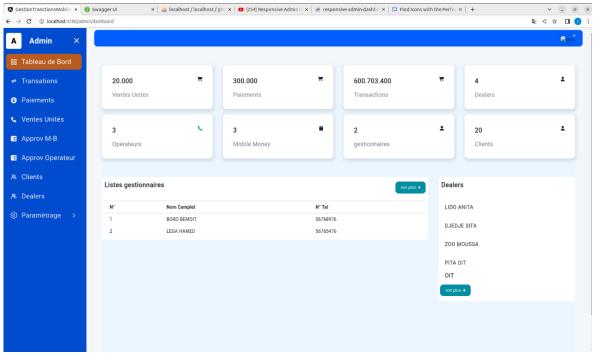


Figure 41: Page d'accueil de l'administrateur

2.2. Gestionnaire

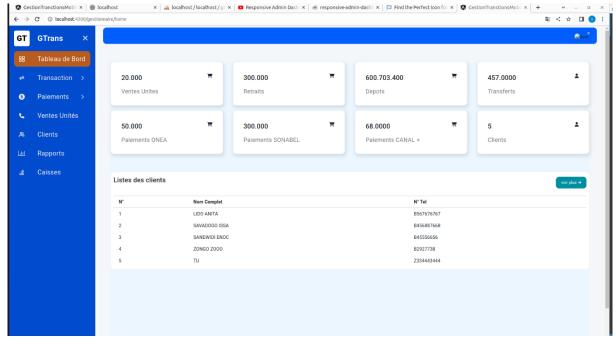


Figure 42: Page d'accueil du gestionnaire

3. Page gestion des transactions

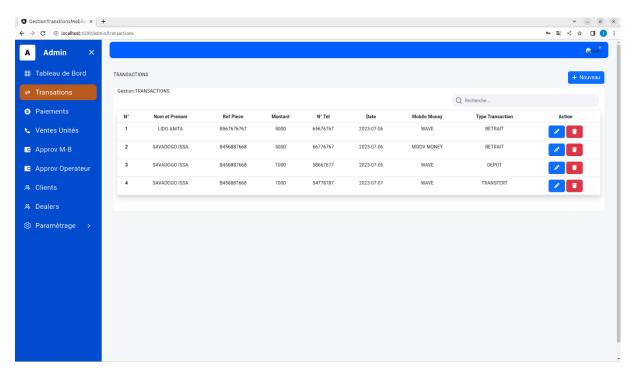


Figure 43: Page « Liste de transaction »

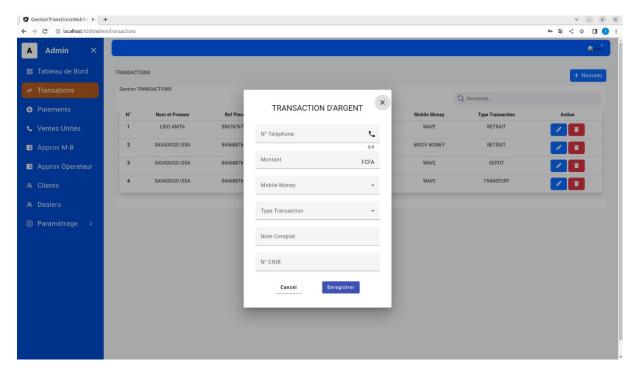


Figure 44: Page « Ajouter une transaction »

IV. Planning réel de réalisation et analyse des écarts

Au cours de l'exécution de notre projet, nous avons constaté plusieurs retards, notamment dans la mise en œuvre et la conceptualisation. Le projet dans son ensemble a accusé un retard de 23 jours causé principalement par l'apprentissage des Framework et la réalisation du Backend.

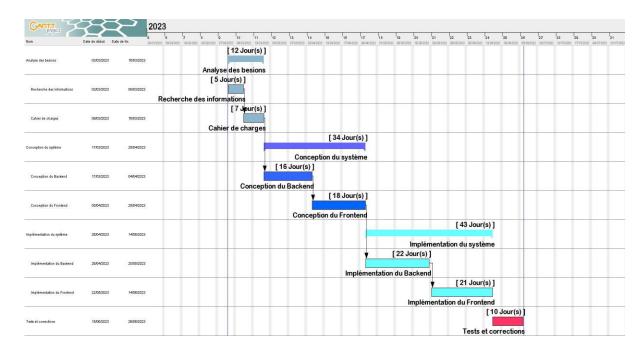


Figure 45 : Planning prévisionnel réel

CONCLUSION

Notre stage au sein de l'entreprise **eGate** durant lequel nous avons été chargé d'effectuer un travail sur le thème : « **Conception et développement d'un système de gestion des transactions mobiles** » nous a permis de mettre en application nos connaissances acquises durant ces trois (03) dernières années à l'Unité de Formation et de Recherche en Sciences Exactes et appliquées (UFR/SEA). Dans ce travail, nous nous sommes intéressés à la création d'un système qui permet aux gestionnaires de garder une trace sur tous les mouvements effectués (les transactions, les paiements et les ventes des unités). L'application permet aussi la gestion des clients, des dealers, ainsi que la gestion des dépenses de l'entreprise.

Ce projet a fait l'objet d'une expérience intéressante. Il nous a permis de comprendre que le projet de réalisation d'une application web 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 web. La réalisation de ce projet nous a permis de mieux nous organiser, d'apprendre à travailler en équipe et de faire progresser nos connaissances et nos compétences en programmation. Nous avons appris des méthodes plus efficaces pour travailler avec les langages de programmation suivants : Java, TypeScript et HTML.

Cependant, compte tenu du manque de temps tout au long des trois mois de notre stage, nous avons alors développé les fonctionnalités suivantes : gérer les enregistrements de paiements, gérer les enregistrements des ventes des unités, gérer les approvisionnements, gérer les clients, les gestionnaires et les dealers.

Comme perspective, nous allons implémenter les restes des fonctionnalités à savoir la gestion des dépenses du kiosque, gestion des bilans mensuels et manuels. Nous allons aussi implémenter le deuxième module de notre application qui est la gestion des produits du kiosque.

BIBLIOGRAPHIE

- [1] OUILY Hamed Joseph, 'Développement d'une application web d'extraction et de structuration de textes juridiques a partir de fichiers PDF et images ', Rapport de Licence, Département Informatique, UFR/SEA, UJKZ, Ouagadougou, 2019.
- [2] OUEDRAOGO Abdoulaye, 'Intégration d'une application de paiement mobile destinée à une solution de gestion commerciale ', Rapport de Licence, Département Informatique, UFR/SEA, UJKZ, Ouagadougou, 2019.
- [2] LOMPO Laurent, 'Implémentation d'une plateforme e-commerce au Burkina Faso', Rapport de Licence, Département Informatique, UFR/SEA, UJKZ, Ouagadougou, 2022.
- [2] NIKIEMA Marius, 'Développement d'une application web de souscription de kit solaire pour la société Giga Energie', Rapport de Licence, Département Informatique, UFR/SEA, UJKZ, Ouagadougou, 2022.

WEBOGRAPHIE

- [1] « Introduction à MERISE Art du numérique ».

 http://infofacile.overblog.com/2017/11/introduction-a-merise.html (20 juin 2023).
- [2] « Le SysML : cours 1re Sciences de l'ingénieur SchoolMouv ». https://www.schoolmouv.fr/cours/le-sysml/fiche-de-cours (20 juin 2023).
- [3] « Présentation de la méthode MERISE Apprendre en ligne ». https://www.clicours.com/presentation-de-la-methode-merise/ (20 juin 2023).
- [4] « Most Popular Java Backend Frameworks for 2023 ». *Rollbar*. https://rollbar.com/blog/most-popular-java-backend-frameworks/ (20 juin 2023).
- [5] « Présentation du langage de modélisation unifié (UML) ». Cybermédiane. https://www.cybermedian.com/fr/unified-modeling-language-uml-introduction/ (20 juin 2023).
- [6] « Top 10 des frameworks frontaux populaires à utiliser en 2023 | AppMaster ». https://appmaster.io/fr/blog/frameworks-frontaux-populaires (20 juin 2023).
- [7] « Top 10 des meilleurs frameworks Web Backend en 2023 pour le développement Web | AppMaster ».

 https://appmaster.io/fr/blog/10-meilleurs-frameworks-web-backend (20 juin 2023).
- [8] « 8 Most Popular Java Frameworks for Backend Development ». *BariTechSol*. https://baritechsol.com/blog/popular-java-frameworks/ (20 juin 2023).
- [9] « Qu'est-ce que React.js ? Un regard sur la bibliothèque JavaScript populaire ». 2022. *Kinsta*®. https://kinsta.com/fr/base-de-connaissances/qu-est-react-js/ (20 juin 2023).
- [10] « Qu'est-ce qu'un système de gestion de base de données ». *Oracle France*.

 //www.oracle.com/fr/database/systeme-gestion-base-de-donnees-sgbd-definition.html
 (20 juin 2023).
- [11] « Qu'est-ce que JavaScript ? JavaScript expliqué AWS ». *Amazon Web Services, Inc.* https://aws.amazon.com/fr/what-is/javascript/ (21 juin 2023).

- [12] « Qu'est-ce que TypeScript ? Un guide complet ». 2023. *Kinsta*®. https://kinsta.com/fr/base-de-connaissances/guide-complet-typescript/ (21 juin 2023).
- [13] « Qu'est-ce que Java Spring Boot? Easy Partner ». 2023. https://easypartner.fr/blog/quest-ce-que-java-spring-boot/ (21 juin 2023).
- [14] « Développement web : qu'est-ce que React.JS et pourquoi le choisir ? » 2022. *Bocasay*. https://www.bocasay.com/fr/avantages-reactjs-developpement-web/ (21 juin 2023).
- [15] « Oracle Database : définition et fonctionnement ». 2022. *IONOS Digital Guide*. https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/oracle-database/ (22 juin 2023).
- [16] « Le modèle MVC (Model-View-Controller) Très Facile ». https://www.tresfacile.net/le-modele-mvc-model-view-controller/ (4 juillet 2023).
- [17] TEKOMBO, Gabriel. 2021. « Comprendre l'architecture MVVM Sur Android ». AndroidMood. https://medium.com/androidmood/comprendre-larchitecture-mvvm-sur-android-aa285e4fe9dd (4 juillet 2023).
- [18] kexugit. 2015. « MSDN Magazine: Cutting Edge Better Web Forms with the MVP Pattern ». https://learn.microsoft.com/en-us/archive/msdn-magazine/2010/september/msdn-magazine-cutting-edge-better-web-forms-with-the-mvp-pattern (4 juillet 2023).