

DÉDICACE

*A la famille SIYAPZE...*

## REMERCIEMENTS

Avant de présenter ce travail effectué dans le cadre du projet de fin d'études Ingénieur Informatique modules : Java Professionnel et Technologie web et mobile, nous tenons à remercier tous ceux et celles qui ont participé à la réussite de notre projet, et ceux qui ont fait de notre formation une expérience enrichissante. Nos remerciements vont à l'endroit de :

- Monsieur GUIMEZAP Paul, Président Fondateur de l'IUC pour l'initiative qu'il a eu de contribuer à la formation de la jeunesse ;
- Madame NOUBANKA Manuella, Directrice de 3IAC pour son encadrement, ses conseils et l'attention qu'elle nous a accordé ;
- Docteur AZEUFACK ULRICH, le Superviseur, pour ses conseils et son suivi tant sur le plan académique que professionnel ;
- Monsieur TEKOU DJOU Xavier, l'encadreur Académique, qui de ses mains de maître, a su canaliser ce présent travail ;
- Monsieur TALOM GUY, Le CEO de IZIWAY CAMEROUN ,de nous avoir accordé cette confiance d'effectuer notre stage ;
- Madame SIMO CLARISSE EPSE SIYAPZE, La Directrice GENERALE de IZIWAY CAMEROUN, pour ses encouragements et ses conseils stratégiques sur le projet ;
- Monsieur TCHOUA EDMOND l'encadreur professionnel pour ses multiples conseils et l'édification de la vie en entreprise ;
- L'équipe pédagogique de notre école 3IL, pour nous avoir apporté les connaissances nécessaires nous permettant d'effectuer notre travail dans les meilleures conditions.

## SOMMAIRE

<b>Dédicace</b>	<b>i</b>
<b>Remerciements</b>	<b>ii</b>
<b>Sommaire</b>	<b>iv</b>
<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>Liste des sigles et abréviations</b>	<b>vii</b>
<b>Liste des tableaux</b>	<b>viii</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Introduction</b>	<b>1</b>
<b>Chapitre 1 NOTION DE BASE SUR L’OPTIMISATION.</b>	<b>2</b>
1.1 En préalable à l’optimisation des processus . . . . .	2
1.2 Les principes de base de l’optimisation des processus . . . . .	3
1.3 Une démarche possible pour optimiser un processus[4] . . . . .	5
<b>Chapitre 2 ETUDE DE L’EXISTANT.</b>	<b>11</b>
2.1 État de l’art d’une Marketplace . . . . .	11
2.2 Description de l’existant. . . . .	15

2.3	Cahier de charge . . . . .	18
2.4	Méthodologie de conduite de projet. . . . .	28
<b>Chapitre 3</b>	<b>Analyse et Conception.</b>	<b>34</b>
3.1	Présentationd de la méthodologie. . . . .	34
3.2	Modèle. . . . .	37
3.3	Concept d'UML . . . . .	39
3.4	Analyse et Modélisation . . . . .	42
3.5	Architecture générale de l'application . . . . .	52
<b>Chapitre 4</b>	<b>Résultats et Discussion</b>	<b>56</b>
4.1	Outils, langage et méthodes d'implémentations. . . . .	56
4.2	Présentation de la solution réalisée . . . . .	58
4.3	Discussion de la solution et bilan. . . . .	69
<b>Conclusion</b>		<b>75</b>
<b>Bibliographie</b>		<b>1</b>
<b>Chapitre A</b>	<b>Table des matières.</b>	<b>AA</b>
<b>Table des matières</b>		<b>AA</b>

## RÉSUMÉ

Le terme UX (acronyme de l'anglais : User Expérience), expérience utilisateur en français, désigne la qualité de l'expérience vécue par l'utilisateur dans toute situation d'interaction. L'objectif primordial de notre stage a été de développer une solution informatique permettant d'augmenter la qualité de service de l'application IZIWAY CAMEROUN. Cette solution vient pallier à des problèmes tels que le faible taux d'utilisateur, le faible taux de conversion ( nombre d'achat) sur la plateforme, le manque d'animation sur l'application. Au regard de ces différents problèmes, nous avons travaillé sur comment optimiser l'application mobile d'une entreprise qui fait dans le E-commerce. Dans cette optique, nous avons opté pour la méthode agile plus précisément la méthodologie SCRUM en raison de la complexité du projet et surtout parce que cette méthodologie de travail met le client au centre des activités. L'application IZIWAY CAMEROUN est basée sur l'architecture MVC, l'ensemble des interfaces ont été développés avec le Framework Flutter, les interactions avec la base de données stockées dans le système de gestion de base de données MYSQL SERVER se font à travers une api qui a été développé en Csharp. Au vu des éléments qui précèdent, nous pouvons dire qu'il y'a une nette amélioration sur l'ergonomie de l'application ce qui participe à l'augmentation du nombre d'utilisateur et augmente le taux de conversion (taux d'achat) sur notre plateforme. La solution offerte à la suite de notre travail est une réussite en raison de la satisfaction du top management de IZIWAY CAMEROUN et de celle des employés qui n'ont pas manqué de signifier une nette amélioration dans leur travail.

*Mots clés* : Marketplace, Expérience utilisateur, Processus d'achat.

## ABSTRACT

The term UX (User Experience) refers to the quality of the user's experience in any interaction situation. The primary objective of our internship was to develop a computer solution to increase the quality of service of the IZIWAY CAMEROON application. This solution addresses problems such as the low user rate, the low conversion rate (number of purchases) on the platform, the lack of animation on the application. With regard to these various problems, we worked on how to optimize the mobile application of a company that is in E-commerce. In this perspective, we opted for the agile method, more precisely the SCRUM methodology because of the complexity of the project and especially because this work methodology puts the customer at the center of the activities. The IZIWAY CAMEROUN application is based on the MVC architecture, all the interfaces have been developed with the Flutter Framework, the interactions with the database stored in the database management system MYSQL SERVER are done through an api which has been developed in Csharp. In view of the above, we can say that there is a clear improvement on the ergonomics of the application which contributes to the increase of the number of users and increases the conversion rate (purchase rate) on our platform. The solution offered as a result of our work is a success because of the satisfaction of the top management of IZIWAY CAMEROON and that of the employees who did not fail to signify a clear improvement in their work.

*Keywords* : Marketplace, User experience, Purchasing process.

## LISTE DES SIGLES ET ABRÉVIATIONS.

- **MOA** : Maître d'ouvrage
- **MOE** : Maître d'œuvre
- **SGBD** : Système de Gestion de Base de Données
- **UML** : Unified Modeling Language
- **SI** : Système d'information
- **COCOMO** : COConstructive COst MOdel
- **HM** : Homme - Mois
- **KLSL** : Nombre de milliers d'instructions de l'application
- **TDEV** : Temps de Développement
- **NP** : Nombre de personnes
- **ISO** : International Organization for Standardization

## LISTE DES TABLEAUX

2.1	Caractéristiques de la solution voulue . . . . .	21
2.2	Avantages et inconvénients de l'externalisation . . . . .	23
2.3	Avantages et inconvénients de l'internalisation . . . . .	24
2.4	coûts de réalisation du projet . . . . .	26
2.5	Caractéristiques de la solution voulue . . . . .	27
2.6	Méthodes : agiles vs classiques.[9] . . . . .	33
3.1	Comparaison entre Merise et UML . . . . .	38
3.2	Acteurs du système.[9] . . . . .	43
3.3	Description du cas d'utilisation "Gestion des blocs produits" . . . . .	45
3.4	Description du cas d'utilisation "Gestion des doubles bannières" . . . . .	45
3.5	Description du cas d'utilisation "Gestion des push notifications" . . . . .	46
3.6	Description du cas d'utilisation "Gestion des sliders" . . . . .	46
4.1	Etude financière du matériel. . . . .	68
4.2	Etude financière de la formation. . . . .	68
4.3	Total coût de mise en oeuvre. . . . .	68
4.4	Méthodologie. . . . .	70
4.5	Evaluation. . . . .	70



## LISTE DES FIGURES

1.1	Cartographie des processus . . . . .	6
1.2	Exemple d'indicateurs . . . . .	8
2.1	Acteurs de la marketplace . . . . .	12
2.2	Processus de passation d'une commande . . . . .	16
2.3	Amazon. . . . .	19
2.4	Alibaba. . . . .	20
2.5	Jumia. . . . .	20
2.6	Diagramme de Gantt. . . . .	28
2.7	Modèle de cycle de vie en cascade . . . . .	29
2.8	Modèle de cycle de vie en V . . . . .	30
2.9	Modèle de cycle de vie en spirale . . . . .	31
2.10	Méthode agile. . . . .	32
3.1	Comparatif des méthodes agiles. . . . .	35
3.2	Processus SCRUM. . . . .	36
3.3	Diagramme de cas d'utilisation du système. . . . .	44
3.4	Diagramme de séquence d'authentification. . . . .	47
3.5	Diagramme de séquence "Gestion des doubles bannières". . . . .	48
3.6	Diagramme de séquence "sliders". . . . .	49
3.7	Diagramme de séquence "Gestion des push notifications". . . . .	50
3.8	Diagramme de classe . . . . .	51
3.9	Diagramme de déploiement. . . . .	52
3.10	Architecture 2-tiers. . . . .	53
3.11	Architecture 3-tiers. . . . .	53

3.12	Architecture N-tiers. . . . .	54
4.1	Architecture MVC de l'application. . . . .	60
4.2	Page du tableau de bord. . . . .	61
4.3	Page de gestion des sliders. . . . .	61
4.4	Rendu des sliders sur l'application mobile. . . . .	62
4.5	Tableau de bord onesignal. . . . .	63
4.6	Liste des push notifications. . . . .	63
4.7	Création d'un push notification. . . . .	64
4.8	Statistiques d'envoi d'un message. . . . .	65
4.9	Exemple d'un push notification reçu. . . . .	66
4.10	Tableau de bord de gestion de bloc categorie. . . . .	67
4.11	Liste des produits d'une catégorie (Bloc Produit). . . . .	67
4.12	Formulaire d'ajout des produits dans un bloc produit. . . . .	68
4.13	Nombre d'utilisateurs avant le début du stage. . . . .	71
4.14	Nombre d'utilisateurs après mon arrivé. . . . .	72
4.15	Nombre de conversion avant mon arrivé. . . . .	72
4.16	Nombre de conversion après mon arrivé. . . . .	73

## INTRODUCTION

Le concept de e-commerce a fait son apparition au moment des premières transactions faites par des moyens électroniques. Dans les années 90, internet fait son apparition dans les familles Française lorsque les ordinateurs se commercialisent et se démocratisent. Les transactions entre entreprises et particuliers commencent alors apparaître. En 1997, le e-commerce se démocratise enfin et les grandes entreprises du secteur informatique comme Microsoft commencent à entrer dans le virtuelle. Les nouvelles technologies (smartphones, tablettes,...) dont nous disposons maintenant, nous permettent de passer de la vie réelle à la vie virtuelle en quelques instants, ainsi les consommateurs ont vu dans ce nouveau canal un moyen de trouver le meilleur rapport qualité prix. Pour pouvoir permettre une gestion optimale d'une entreprise de e-commerce, les différents services s'appuient sur les processus métier tel que les passations de commandes, les demandes de publications des articles. Ces processus exécutés de manière manuelle ou même sémi-manuelle ( via les outils développés), rencontrent plusieurs problèmes à savoir : le faible taux d'utilisateur sur l'application, le faible taux de conversion ( nombre d'achat ) des articles à travers l'application, le manque d'animation sur l'application. A travers ses problèmes cités, nous ressortons la problématique suivante : « Comment améliorer l'expérience utilisateur et augmenter notre chiffre d'affaire ». L'objectif majeur de notre étude est d'assurer la qualité de services de l'application. Les objectifs spécifiques sont : améliorer l'ergonomie de l'application, augmenter le nombre d'animation sur l'application, améliorer les indicateurs de performance ( taux de conversion, ). La première chapitre et deuxième chapitre porteront sur l'étude de l'art c'est-à-dire les notions d'une optimisation, ensuite nous ferons une étude de l'existant c'est-à-dire présenter la Marketplace les différentes typologie de Marketplace qui existe. Ensuite nos deux derniers, nous ferons une étude, une analyse, une conception et implémentation de la solution répondant à la problématique étudiée.

# CHAPITRE 1

## NOTION DE BASE SUR L'OPTIMISATION.

### 1.1 En préalable à l'optimisation des processus

#### 1.1.1 Qu'est-ce qu'un processus.

Un processus [1] est un ensemble d'opérations ou d'activités réalisées par des acteurs avec des moyens et selon des références en vue d'une finalité. Dans le cadre des démarches qualité, un processus doit toujours être tourné vers un bénéficiaire ou un système bénéficiaire, interne ou externe. Un processus peut comprendre des activités réalisées par différents services, différentes entités. Ce caractère transversal, supposant de nombreuses interfaces, est souvent un des points cruciaux de l'amélioration du service ou du produit fourni aux bénéficiaires.

#### 1.1.2 Pourquoi les optimiser ?[2]

Une action sur les processus vise différents objectifs :

- mieux prendre en compte les attentes des bénéficiaires pour améliorer les services fournis
- permettre aux différents acteurs de s'impliquer dans le fonctionnement du processus,
- clarifier les rôles et responsabilités des acteurs, définir les marges de manœuvre et les cohérences nécessaires, simplifier les interfaces entre entités ;
- transformer ou créer un nouveau processus pour répondre à de nouvelles attentes ;

- diminuer les coûts, les délais d'un processus, augmenter sa performance au regard d'indicateurs définis ;
- mieux réagir aux aléas ;
- viser une certification via la mise en place d'un système qualité ;
- accompagner la mise en place d'un progiciel de gestion. . .

### **1.1.3 A quel moment faut-il le faire ?**

Le travail sur les processus s'inscrit en général dans le cadre d'une démarche qualité. Cette démarche doit par conséquent être lancée : affichage de la politique qualité[3] et de ses axes, communication aux personnels, engagement de la direction, plan d'action, formation des acteurs clés. Par ailleurs, l'optimisation des processus est une méthode qui accompagne efficacement différents types de démarches :

- Des démarches qualité comme, par exemple, l'élaboration d'engagements de service, la certification selon les normes ISO, la réduction de dysfonctionnements
- Mais aussi d'autres démarches comme, par exemple, les réorganisations globales, les fusion de service, la gestion des compétences, l'émergence de nouvelles activités.

## **1.2 Les principes de base de l'optimisation des processus**

### **1.2.1 Bien caractériser le périmètre couvert par le processus**

L'essoufflement des démarches d'optimisation des processus s'explique bien souvent par un mauvais cadrage du périmètre des différents processus. Ces zones de flous provoquent vite des débats, des revendications des pilotes des différents processus. Il convient donc de définir avec précision les champs que couvre chaque processus, en terme d'activités, de productions mais aussi d'acteurs. Cette tâche se révèle parfois difficile lorsque les processus sont transverses à différentes entités.

### **1.2.2 Identifier les interfaces**

C'est souvent aux interfaces entre processus ou entre entités à l'intérieur d'un même processus que se situent les principales zones d'amélioration potentielle. Il convient donc de les identifier au mieux, d'un point de vue commun aux différents acteurs qui y interviennent. Il est également important d'étudier, à ces interfaces, les modalités de circulation de l'information liée au processus : y-a-t-il une bonne traçabilité ? Les informations importantes des étapes passées sont-elles bien prises en compte aux étapes suivantes ? N'y-a-t-il pas de jeux d'acteurs aux interfaces avec des objectifs de pouvoir par rétention d'informations ?

### **1.2.3 Ne travailler que des processus-clés ou processus critiques**

Le travail sur les processus doit être cadré d'un point de vue stratégique et ne viser qu'à améliorer des performances qui font sens au niveau du service et de ses bénéficiaires. Il ne s'agit donc pas de travailler sur l'ensemble des processus, mais seulement sur quelques-uns qui pourraient apparaître prioritaires au vu de différents critères :

- Forts dysfonctionnements,
- Insatisfaction des bénéficiaires ou émergence de nouvelles attentes,
- Évolution de la stratégie du service

### **1.2.4 Privilégier une approche participative**

Les démarches d'optimisation de processus les plus efficaces sont celles qui associent assez étroitement les acteurs des processus dans leur amélioration. A charge du pilote de fixer les modalités de ce travail participatif, en les échelonnant dans le temps. Une discussion sur la caractérisation du processus est, dans tous les cas, indispensable. Les travaux peuvent également associer des bénéficiaires. Enfin, il est important, sur ces processus-clés, d'anticiper les possibles résistances au changement des acteurs face aux évolutions : optimiser un processus signifie souvent modifier des pratiques routinières et davantage se tourner vers les bénéficiaires.

### **1.2.5 Garder de la souplesse dans la formalisation pour rester ouvert à l'urgence**

La formalisation des processus ne doit pas créer un système rigide dans lequel chaque acteur se limiterait à observer à la lettre la procédure. Au contraire, ce système doit être ouvert, pour permettre et même favoriser les initiatives des acteurs, et adaptable, pour pouvoir réagir aux aléas, aux urgences, et à plus long terme aux évolutions des attentes des bénéficiaires. Dans certains cas, il peut même être utile de configurer un processus spécifique, adapté, pour traiter au mieux l'urgence. Il faut enfin noter que le degré de formalisation d'un processus varie selon les compétences des acteurs qui le font fonctionner. De façon générale, plus les compétences sont élevées, moins la formalisation est stricte. Elle est alors remplacée par la maîtrise professionnelle des acteurs.

## **1.3 Une démarche possible pour optimiser un processus[4]**

Nous proposons une démarche en plusieurs étapes séquencées. Selon le contexte, la stratégie, les priorités de votre service, des poids différents peuvent être mis derrière ces étapes. Nous conseillons dans tous les cas de mettre au point la démarche de façon précise, par exemple en suivant les principes de la conduite de projet. Cela suppose notamment de bien cadrer la démarche ou le projet d'optimisation, de nommer un pilote pour chaque processus, de définir un dispositif de suivi, de formaliser un calendrier et d'allouer les moyens nécessaires.

### **1.3.1 Cartographier les processus**

Pour aborder une démarche globale de travail sur les processus, il est utile d'en avoir une vision d'ensemble. Un repérage de l'ensemble des grands processus d'une organisation peut ainsi être réalisé sous la forme d'un schéma. Les conventions suivantes peuvent être respectées :

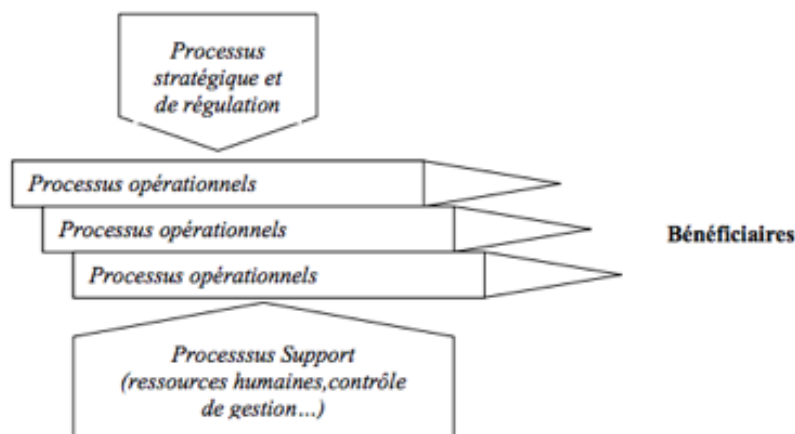


FIGURE 1.1 – Cartographie des processus

source : *Groupe de travail - Optimisation des processus - Etienne Jasmin*

### 1.3.2 Choisir les processus-clés

A l'intérieur de la cartographie précédente, il ne s'agit pas toujours de vouloir optimiser tous les processus, sauf si l'on souhaite se réorganiser entièrement. Différents critères peuvent ainsi aider aux choix des processus sur lesquels les travaux d'optimisation porteront en priorité :

- Constats des dysfonctionnements,
- Insatisfaction des bénéficiaires ou émergence de nouvelles attentes,
- Évolution de la stratégie ou émergence de nouvelles attentes,
- Développement de nouvelles démarches (ARTT, gestion des compétences) qui peuvent avoir un impact fort sur certains processus,
- Mise en place de nouveaux outils informatiques et notamment et progiciels de gestion intégrée,
- Lancement d'une démarche d'engagements de service (pour s'engager vis à vis d'un bénéficiaires, il est impératif de maîtriser les processus afférents)

### 1.3.3 Caractériser un processus

Il s'agit de répondre aux questions suivantes :



- Quelle est la finalité du processus ?
- Quel est le bénéficiaire ou le système bénéficiaire du processus ?
- Quel(s) est(sont) le(s) service(s) ou produit(s) fourni(s) ?
- Quelles sont les exigences des bénéficiaires par rapport à ce service / produit ?
- Quels sont les indicateurs qui permettent de mesurer le respect de ces exigences et plus globalement la performance du processus ?
- Quels sont les acteurs qui concourent directement au processus ?
- Quels sont les principaux moyens utilisés ?
- Quels sont les éléments d'entrée du processus ? (ce sont parfois les éléments déclencheurs) ?
- Quels sont les fournisseurs de ces éléments ?
- Quelles sont les exigences du processus par rapport à ces fournisseurs ?
- Quels sont les indicateurs qui permettent de mesurer le respect de ces exigences ?

Le repérage des exigences des bénéficiaires est une étape cruciale de la caractérisation du processus. Elle peut mériter une enquête approfondie auprès des bénéficiaires.

Enfin, lorsque la démarche porte sur plusieurs processus, il est important, à partir de leurs caractérisations, de vérifier qu'ils ne se recouvrent pas ou alors que leurs interfaces sont clairement identifiées.

### **1.3.4 Décrire un processus**

Il s'agit de réaliser une description d'ensemble du processus avec la présentation synthétique des activités et de leurs responsables. Ce travail peut utilement se faire sous la forme d'un diagramme « qui fait quoi ». Les principales étapes du processus peuvent ensuite être identifiées, les principaux points de contrôle et les indicateurs actuels également. Cette représentation synthétique du processus permet d'en prendre connaissance rapidement et de susciter de façon participative des interrogations sur l'enchaînement des activités, les relations entre acteurs et avec les bénéficiaires.

### 1.3.5 Diagnostiquer un processus et son contexte pour définir les objectifs d'optimisation

A partir de la description précédente, il s'agit de mener un diagnostic approfondi du processus. Ce diagnostic repose sur l'identification précise des principaux faits marquant le fonctionnement du processus et son contexte. Ces faits peuvent par exemple porter sur :

- Les dysfonctionnements internes au processus,
- Les non-qualités constatées,
- La fréquence des anomalies,
- Les insatisfactions des bénéficiaires,
- Les évolutions des indicateurs (coût, délai...),
- Les temps passés à la réalisation de tout ou partie du processus,
- L'émergence de nouvelles attentes des bénéficiaires

Une fois le diagnostic réalisé, il s'agit de le traduire sous forme d'objectifs clairement formulés et visant l'optimisation du processus : "réduire de 15% le temps passé à l'accomplissement de cette partie de processus", "accélérer d'un jour les délais", "réduire le nombre d'anomalies de 20%", "augmenter de 2 points la satisfaction des bénéficiaires".

### 1.3.6 Choisir le degré d'optimisation : améliorer ou reconcevoir selon des objectifs et indicateurs de performance

En fonction des objectifs identifiés à partir du diagnostic précédent, il s'agit de décider des actions à mener. Celles-ci peuvent avoir une ampleur très variable, selon deux dimensions :

	Tout le processus	Une partie du processus
Amélioration	X	X
Reconception	X	X

FIGURE 1.2 – Exemple d'indicateurs

source : *Groupe de travail - Optimisation des processus - Etienne Jasmin*

Une action d'amélioration consiste à reprendre le processus existant pour agir sur certains de ses facteurs. Exemples : mise en parallèle de certaines étapes, suppression d'une étape, changement d'outil sur une étape, développement des compétences pour une tâche, mise en place d'indicateurs à un point stratégique du processus, modification de la procédure régissant une activité.

Une action de reconception ne part pas du processus existant. Elle ne se base que sur les niveaux de performance visés et les moyens et ressources disponibles pour concevoir un tout nouveau processus, avec des façons de faire nouvelles, des enchaînements non encore pratiqués.

### **1.3.7 Optimiser le processus**

En fonction des objectifs précédemment identifiés, il s'agit de décider des actions d'optimisation à mettre en œuvre, actions qui conduiront à modifier de façon plus ou moins forte le processus. Les modalités de ce travail varient en fonction des actions décidées. Quelques conseils restent néanmoins valables pour tout type d'optimisation :

- Procéder d'abord sur papier en représentant le processus cible,
- Faire réagir les acteurs concernés,
- Procéder ensuite par tests successifs des différentes phases du processus, rectifier ce qui s'avère difficile à mettre en œuvre,
- Concevoir les actions d'accompagnement à mettre en œuvre (formation, gestion des compétences, outillage...),
- Communiquer tout au long de ces phases,
- Installer enfin l'ensemble du processus, en permettant, sur un délai limité, les ajustements nécessaires à sa bonne mise en œuvre,
- Installer l'ensemble des indicateurs : indicateurs sur le processus (en cours de production) et indicateurs de résultats (sur le produit / service fourni et sur la satisfaction des bénéficiaires),
- Définir le pilote de processus (personne ou fonction)

### **1.3.8 Mettre en œuvre et piloter le nouveau processus**

Une fois installé, le processus doit vivre sur le long terme. Son pilotage en continu est indispensable :

- de façon opérationnelle : suivi des indicateurs de processus et de résultat, traitement des dysfonctionnement, relevé de fonctionnement, bouclages, traitement des suggestions des acteurs du processus, suivi des moyens mis en œuvre, suivi des compétences
- et de façon stratégique : réorientations du processus selon les évolutions du contexte, de l'environnement, des attentes, maintien de la cohérence entre le processus piloté et le système global.

## CHAPITRE 2

### ETUDE DE L'EXISTANT.

## 2.1 État de l'art d'une Marketplace .

### 2.1.1 Qu'est-ce qu'une Marketplace.

#### 1. Definition.

Le terme de Marketplace, ou place de marché en bon français, désigne toute plateforme qui met en relation des acheteurs et des vendeurs sur Internet. La plateforme récupère en échange une commission sur les ventes qu'elle a permis de réaliser. Certains sites marchands comme eBay ou Rakuten sont des pures Marketplace, c'est-à-dire qu'ils ne proposent que les produits de vendeurs tiers. D'autres, comme Amazon, Cdiscount ou Rueducommerce, accueillent les produits de ces vendeurs tiers, aux côtés de leur propre gamme de produits[5].

Ces vendeurs, qui sont parfois même des particuliers, profitent alors des fonctionnalités de la plateforme d'e-commerce et de son audience. Autrement dit et de manière schématique, une Marketplace, c'est comme un centre commercial, mais sur Internet. La place de marché propose en effet une sorte de galerie marchande digitale dédiée aux vendeurs. Ces derniers y installent leur propre magasin en ligne et y font leurs affaires. Le marchand qui gère la Marketplace peut, dans certains cas, leur proposer de prendre en charge le stockage et l'expédition des produits. Le vendeur doit lui faire appel à un gestionnaire de flux pour gérer la diffusion de son catalogue produit au sein de la Marketplace, suivre ses ventes et ajuster sa stratégie en conséquence.

A la base, l'intérêt de la Marketplace était de servir d'intermédiaire dans une relation interprofessionnelle (B2B). Quand le client peut payer en ligne sur la Marketplace. Aujourd'hui, cette technique s'est ouverte au B2C via des sites marchands gigantesques.

Au regard de tout ce qui précède, IZIWAY ambitionne de se doter de nombreux outils qui sera intégré à son application mobile et devra améliorer l'utilisation de son application et aussi de fidéliser ses clients.

## 2. Les acteurs de la marketplace

Dans un Marketplace on distingue trois acteurs principaux acteurs et d'autres secondaires, qui ont des rôles spécifiques et complémentaires afin de concrétiser une vente.



FIGURE 2.1 – Acteurs de la marketplace

source : <https://www.redsen-consulting.com/fr/inspired/marketplace/marketplace-les-acteurs-secondaires>

## 3. Typologie des relations commerciales

Les entreprises et les consommateurs sont les deux acteurs plus courts au commerce électronique. Il existe quatre types principaux[6] :

- La Marketplace B2C (Business to Consumer) : elle est destinée aux transactions ou mises en relations entre vendeurs professionnels et acheteurs particuliers . C'est aujourd'hui la forme la plus connue de la Marketplace avec des acteurs comme Amazon, Cdiscount. Un des facteurs clés de succès de la Marketplace B2C est basé sur les volumes que celle-ci a su drainer.

- La Marketplace B2B (Business to Business) : commerce électronique entre des entreprises, elle est plus ancienne que B2C
- La Marketplace C2B (Consumer to Business) : commerce électronique dont un consommateur est au service de l'entreprise, comme appels d'offres d'emplois d'un projet.
- La Marketplace C2C (Consumer to Consumer) : commerce électronique entre deux consommateurs sur des sites qui offrent des annonces sur internet, ou peut-être même deux particuliers peuvent vendre et acheter en ligne.

#### 4. Les cinq piliers de la Marketplace

Une Marketplace efficace repose sur cinq piliers nécessaires : [7]

- Le trafic : est le pilier le plus important dans une Marketplace, car cette dernière est en relation directe avec le trafic, plus il y'a des visiteurs plus les opportunités de transactions seront nombreux et la visibilité des articles des vendeurs augmente.
- Le mix produit : on peut distinguer deux types de Marketplace :
  - Des Marketplace avec une faible limite imposé aux vendeurs et la restriction des quasi-inexistant, le vendeur peut poser tous types d'articles
  - Des Marketplace ou les opérateurs définissent précisément quels produits mis en vente avec une maîtrise de catalogue, chaque article doit respecter le catalogue qui lui convient.
- Le cadre légale : les Marketplace n'imposent pas le même niveau d'exigence au niveau juridique. Pour certains Marketplace les contrats sont bien contrôler pour les échanges des articles sensibles. Et pour d'autres cette exigence est moindre, l'identification des intervenants n'est pas trop poussé.
- Les outils de communication et de la gestion de cycle de vente :
  - Les outils de communication entre les acheteurs et les vendeurs sont les facteurs les plus importants de la réussite de concrétisation d'une vente et de l'après-vente, sa peut être des messageries, chat, et beaucoup d'autres outils.
  - Pour la gestion du cycle de vente, les Marketplace affichent aux vendeurs la quantité de stocks de ses produits mise à jour au temps réel. Ainsi que les opé-

rateurs suivent les transactions d'achat avec tous les états d'avancement de la commande jusqu'à la réception de l'article par le client.

- Le paiement : cette partie est très sensible, elle doit être bien contrôlée, pour avoir la confiance du client. Il faut leur assurer que les opérateurs enregistrent la transaction dans un environnement sécurisé, et que leurs données sont bien protégées.

### **2.1.2 Avantages et inconvénients d'une Marketplace**

Il faut bien connaître les avantages et inconvénients de la Marketplace par rapport aux deux principaux acteurs, l'entreprise et le client :

#### **1. Avantages.**

Pour l'entreprise :

- Permet d'envisager la politique de la fidélisation du client à travers une offre des services.
- Nouveau canal de distribution qui permet de se développer sur le marché mieux qu'un magasin physique.
- Meilleure économie en évitant les coûts des locaux, des magasins.
- Améliorer la visibilité des produits, donc c'est un marketing très efficace.
- Permet de toucher des nouveaux clients et développer une relation entre eux et l'entreprise.

Pour le client :

- Plus de choix des produits et avec des prix moins chers
- Économie de temps puisque le client peut acheter sans se déplacer.
- Un renseignement sur la qualité de produit grâce aux descriptions bien détaillées.
- Des économies grâce à la possibilité de comparer les prix.
- Possibilité d'acheter n'importe où et à n'importe quelle heure (24h/24 et 7j/7)
- Pas de file d'attente
- Shopping de la maison



## 2. Inconvénients.

Pour l'entreprise :

- L'insécurité des transactions et le manque de confiance des moyens de paiements face aux pirates.
- Des frais supplémentaires ou la nécessité d'une bonne formation du personnel
- L'anonymat du client
- Coût de la logistique plus élevée que le commerce classique comme les frais de transports.

Pour le client :

- L'insécurité de paiement et la possibilité de tomber sur des sites malveillants.
- En cas de problème c'est difficile de contacter le service après-vente
- Le manque de contact avec le produit et par la suite le client ne peut pas bien se rendre compte sur ses qualités et ses défauts
- Absence de contact avec le vendeur et par la suite le manque de conseils.

## 2.2 Description de l'existant.

### 2.2.1 Type de marketplace

IZIWAY est une plateforme de e-commerce 100% camerounaise qui a été lancée en décembre 2019. Pour son lancement il a été question pour le promoteur de la plateforme de savoir sur quel business model [8]. Soucieux de faire du chiffre d'affaire et de satisfaire notre clientèle, IZIWAY a décidé d'utiliser le model B2C (Business to Consumer) qui est aujourd'hui le model le plus utilisé dans le monde.

De nos jours beaucoup d'entreprises effectuent l'achat de leurs matériels en ligne, il est donc question pour nous de les avoir dans notre portefeuille. C'est dans cette optique que IZIWAY est entrain de migrer vers un business model hybride c'est-à-dire qu'il va être du B2C (Business to Consumer) et du B2B (Business to Business). Ce modèle hybride va nous permettre d'avoir une clientèle large qui peut être à la fois des personnes physiques ou des entreprises.

### 2.2.2 Présentation de l'existant

Nous allons présenter la procédure allant de l'achat d'un article par un client jusqu'à sa livraison. Cette procédure s'applique à toute plateforme de e-commerce.

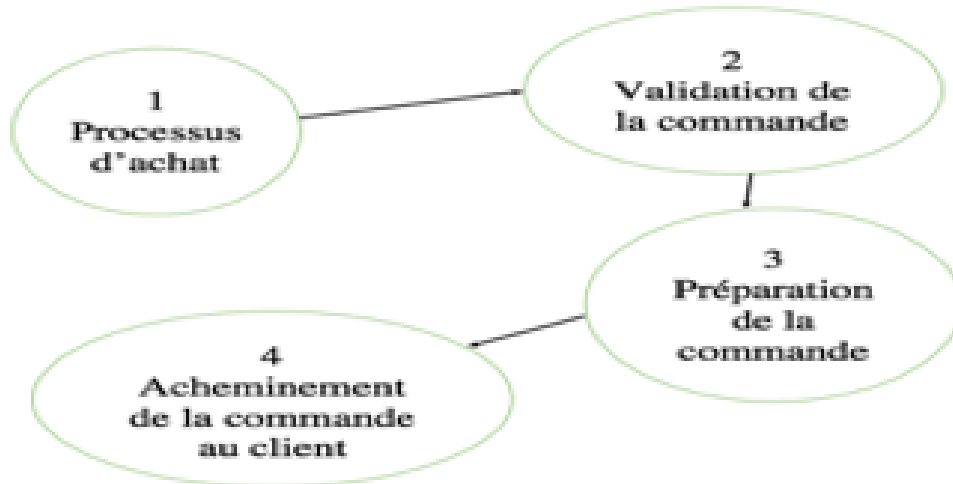


FIGURE 2.2 – Processus de passation d'une commande

source : <https://www.joelpro-educ.com/s/passation-de-la-commande-2nde-stt-cours>

La description de la figure ci-dessus est capitale pour une bonne compréhension de la situation.

1. **Proposition d'achat** : étape au cours de laquelle un client fait une recherche du produit qu'il veut, valide son produit et effectue son achat..

- La recherche : ici le client parcourt l'application et utilise le module de recherche afin de découvrir les produits disponibles,
- La validation sociale : appelée 'Social Proof' en anglais, elle fait référence à la consultation d'avis clients sur le produit par le client
- L'ajout au panier : l'internaute peut choisir de poursuivre ses recherches et ajouter d'avantages de produits ou bien de s'arrêter là,
- La création d'un compte
- Le renseignement des données de facturation et de livraison
- La sélection du mode livraison : à domicile, retrait en agence

- La sélection du moyen de paiement : par transaction bancaire, par Orange Money/ Mobile Money, par paiement à la livraison
  - La confirmation de l'achat : le client a terminé sa commande.
2. **La Validation de la commande** : étape au cours de laquelle le client est contacté par le service client qui lui remémore sa commande avec les différents prix et lui demande s'il veut valider la commande.
  3. **Préparation de la commande** : ici le service client appelle le vendeur pour qu'il vienne déposer l'article contre une décharge. Lorsque le vendeur dépose l'article le service de la logistique prépare l'article en emballant le produit.
  4. **Acheminement de la commande au client** : Ici le service de livraison appelle le client pour lui faire comprendre que son article est déjà prêt et qu'un livreur est en chemin pour le livrer.

### 2.2.3 Critique de l'existant.

D'entrée de jeu, nous avons créé un sondage que nous avons mis à la disposition de notre clientèle afin de collecter un maximum d'idées nouvelles sur notre hypothèse. Après notre investigation avec l'aide de Pareto[9], nous avons constaté de nombreux problèmes relevés par nos différents utilisateurs. Parmi les problèmes relevés lors de notre côtoiement du métier nous pouvons énumérer les plus pertinents :

- Ergonomie peu désirable : En effet la première chose qu'un utilisateur remarque sur une application c'est son côté artistique, la beauté de l'application.
- Le manque de notification sur l'application : Les notifications sur une application sont primordiales parce qu'elles permettent à l'opérateur de mettre les utilisateurs des nouveaux deals qui se préparent
- Le manque de changement des produits sur la page d'accueil de l'application
- La mise à jour récurrente de l'application
- Recherche peu conviviale

## **2.3 Cahier de charge .**

### **2.3.1 Définition du projet.**

Aujourd'hui, toutes les entreprises qui veulent progresser dans un environnement concurrentiel doivent se doter d'un système d'information fiable. Le système d'information est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information. Il s'agit d'un système sociotechnique composé de deux sous-systèmes, l'un social et l'autre technique :

- Le sous-système social est composé de la structure organisationnelle et des personnes liées au SI,
- Le sous-système technique est composé des technologies (hardware, software et équipements de télécommunication) et des processus métiers concernés par le SI.

Afin de bénéficier d'un SI fiable et compétitif, les entreprises utilisent l'informatique comme socle de leur SI afin de bénéficier d'une circulation et d'un traitement rapide des informations pour une meilleure exploitation.

C'est dans ce cadre que nous devons améliorer une infrastructure informatique fiable et compétitive dans leur SI pour le processus d'achat des articles et l'aisance de plus de 10000 utilisateurs, afin de disposer de fidèle client.

### **2.3.2 Présentation des concurrents .**

Le commerce sur Internet est dominé par les deux géants américains Amazon et eBay. Nombreux sont les internautes qui se déplacent directement sur ces deux plateformes de vente pour effectuer leurs achats. Néanmoins en Afrique le leader du e-commerce est Jumia. Il est conseillé de s'intéresser également aux autres Marketplace en ligne, car elles peuvent représenter des opportunités séduisantes (et ce en particulier pour les petites et moyennes entreprises). Nous pouvons citer entre autre :

- AMAZON : Le géant Amazon s'est lancé en 1994 en tant que libraire en ligne pour ensuite se diversifier et devenir un énorme centre logistique. L'entreprise américaine propose une palette assez large et spécialisée de ses propres produits, des services en ligne

mais donne surtout la possibilité aux entreprises du monde entier de commercialiser leurs produits sur sa Marketplace.



FIGURE 2.3 – Amazon.

source : *Google*

Nous pouvons avoir les avantages suivants :

- Fort trafic
- Nombreuses interfaces
- Logistique possible via Amazon
- Possibilité de faire de la publicité pour son offre (payant)

Comme inconvénients nous pouvons avoir :

- Forte concurrence
  - Frais élevés
  - Impossible de créer son propre design
- **ALIBABA** : Le groupe chinois Alibaba est représenté en ligne sur plusieurs places de marché : en plus de Alibaba.com (une plateforme B2B) et Taobao (une plateforme de vente similaire à eBay), l'entreprise a également un marché en ligne pour le B2C dans son portefeuille avec AliExpress. Sur AliExpress, les clients peuvent faire des achats dans le monde entier, mais les vendeurs eux-mêmes sont tous chinois. Les autres marchands en ligne ne sont pas autorisés à proposer leurs produits sur la plateforme de vente. Il en va différemment pour la filiale éponyme : les commerçants internationaux peuvent trouver des clients dans le monde entier.



FIGURE 2.4 – Alibaba.

source : *Google*

Nous pouvons avoir les avantages suivants :

- Fort trafic
- Pages d'articles personnalisables
- Marche international
- Option (payante) de promotion des annonces
- pas de commissions sur les ventes
- Logiciel spécifique pour faciliter les vente

Comme inconvénients nous pouvons avoir :

- Forte concurrence
  - Frais élevés
  - La traduction française laisse parfois à désirer
- JUMIA : est une entreprise de commerce électronique présente sur le marché africain et fondée en 2012. En 2019, plus de 80 000 vendeurs proposent une large gamme de produits et de services à la demande : appareils électroménagers et électroniques, mode, jouets pour enfants mais aussi des services tels des réservations d'hôtels ou d'avion, et la livraison de repas. Jumia est notamment qualifié d'« Alibaba africain » ou d'« Amazon africain ».



FIGURE 2.5 – Jumia.

source : *Google*

Nous pouvons avoir les avantages suivants :

- Fort trafic
- Propre boutique jumia et conception de la mise en page possible
- Nombreuses interfaces
- Possibilité de faire la publicité pour son offre (payant)

Comme inconvénients nous pouvons avoir :

- Forte concurrence
- Connexion à paypal pas très sécurisé

### 2.3.3 Caractéristiques de la solution voulue :

TABLE 2.1 – Caractéristiques de la solution voulue

Objectif	Définition	Comment ?
Fiabilité	Assurer l'authenticité des informations	En connectant l'application à une base de données sql server qui contient au préalable des informations
Intégrité	Assurer l'authenticité de l'application c'est à dire la non modification possible du contenu et la forme	En donnant les droits de modifications a l'administrateur juste
Sécurité	Assurer la sécurité des informations	Le personnel n'a pas accès a l'application sauf les responsables des différentes équipes
Evolutivité	Permettre l'actualisation des informations par rapport personnel	Après le test de l'application, nous sommes à l'écoute des remarques de la Direction des Ressources Humaines

### 2.3.4 Spécification des besoins :

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques.

#### Besoins fonctionnels

L'amélioration qui doit être effectuée sur l'application se veut d'être opérationnel, évolutif, convivial et offrant les informations nécessaires à temps réel.

Pour ceci, le système à réaliser doit satisfaire les exigences de la totalité des utilisateurs. Nous présenterons dans ce qui suit tous les besoins fonctionnels classés par acteurs ainsi que les besoins non fonctionnels communs à tous ces acteurs.

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une plus-value.

Les principaux acteurs en interactions avec notre système sont :

- L'informaticien
- Le responsable marketing
- Le responsable content-writer
- Le client
- Le responsable de la planification commercial

Les besoins fonctionnels classés par acteur :

- Le responsable marketing
  - Créer les push notifications
- Le responsable content-writer
  - Créer et modifier les sliders
  - Créer et modifier les doubles bannières
- L'informaticien
  - Faire la maintenance de l'application
  - Faire le paramétrage de l'application
  - Faire le suivi
- Le client
  - Effectué un achat
  - Consulter les différents produits disponibles, les différents sliders et les différentes bannières



- Le responsable de la planification commercial
  - Ajouter et modifier le bloc produits et le bloc catégori

### **Besoins non fonctionnels**

A part les besoins fondamentaux, notre système doit répondre aux critères suivants :

- La rapidité de traitement : En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel,
- La performance : une application doit être avant tout performant c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale,
- La convivialité : l'application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simple, ergonomiques et adaptées à l'utilisateur.

### **2.3.5 Solutions envisageables.**

L'informatique en entreprise peut être gérée de deux façons :

- **L'externalisation** : c'est le principe selon lequel une entreprise confie à un prestataire extérieur,
- **L'internalisation** l'entreprise s'occupe elle-même de la gestion de son système d'information en interne.

Chacune des deux méthodes présente des avantages et inconvénients que nous allons présenter par la suite.

#### **L'externalisation.**

TABLE 2.2 – Avantages et inconvénients de l'externalisation

Avantages	Inconvénients
Ouverture de l'entreprise sur l'extérieur : la coopération est une démarche enrichissante	Choc culturel engendré par la venue de nouveaux collaborateurs salariés du prestataire
Concentration sur le coeur métier dans une logique de spécialisation	Crainte d'un backsourcing impossible ou d'un changement de prestataire
Concentration sur le coeur de métier dans une logique de spécialisation	perte de confidentialité
Gain de réactivité de l'entreprise	risque de mésentente entre le prestataire et le client

source : <https://www.petite-entreprise.net/P-2555-81-G1-avantages-et-inconvenients-de-l-externalisation.html>

### **L'internalisation.**

TABLE 2.3 – Avantages et inconvénients de l'internalisation

Avantages	Inconvénients
Le contrôle complet : garantissant une certaine confidentialité des données	investissement de départ lourd.
La gestion de la réputation : certains clients seront rassurés que leurs données financières soient gérées en interne par l'entreprise	Coûts liés à la maintenance
Maîtrise de réduction du budget relié au service sous-traité	perte de confidentialité
Acquisition d'une nouvelle base de compétence pour l'entreprise	Délai de livraison

source : <https://www.petite-entreprise.net/P-2555-81-G1-avantages-et-inconvenients-de-l-internalisation.html>

### **Solution envisageable pour le projet.**

Dans notre cas, l'internalisation serait plus avantageuse en termes de coût et d'appropriation de la solution qu'une externalisation. Aussi le critère principal entrant dans la stratégie de l'entreprise qui nous a donné le projet est le caractère confidentiel des données. De plus le risque d'une externalisation étant la dépendance au prestataire, cela peut être très contraignant pour l'entreprise qui gère des données sensibles dans un environnement où l'activité est très variée et évolue constamment. En prenant compte l'objectif de IZIWAY, et l'ensemble des besoins fonctionnels et non fonctionnels mentionné plus haut, il serait idéal d'améliorer l'application existante.

### **2.3.6 Coûts de réalisation par la méthode COCOMO.**

COCOMO (COConstructive COst MOdel) est un modèle qui permet d'estimer le coût, l'effort et le temps nécessaire au développement d'un logiciel. Le modèle original de COCOMO a été édité la première fois par le Dr. Barry Boehm en 1981 et a reflété les pratiques en matière de développement de logiciel de cette époque. Durant les 15 années suivantes les techniques de développement de logiciel ont changé.

COCOMO est le modèle le mieux documenté dont les paramètres sont adaptables à l'environnement. A l'origine, il a été construit sur une étude de 63 projets logiciels de 2000 à 100.000

lignes de codes dans l'entreprise TRW Inc. (Société Américaine spécialisée dans l'Automobile et le Transport). Ce modèle existe en trois versions : simple, intermédiaire et détaillé.

Nous allons présenter les grandes lignes du modèle simples toutefois c'est ce dernier qui est utilisé pour notre cas précis, afin d'introduire la modélisation comme outil d'estimation des coûts et d'illustrer ses avantages en matière de gestion de projet. Le modèle COCOMO simple est destiné à donner des estimations approximatives de coûts. Il s'appuie uniquement sur la taille estimée du logiciel et sur le type de logiciel à développer. Trois modes de projets peuvent donc être distingués :

- **mode semi détaché** : c'est le pont entre le mode organique et le mode embarqué décrit ci-dessous. Dans cette catégorie, l'équipe de projet peut être composée de programmeurs de divers niveaux d'expérience. Les membres de l'équipe ont une expérience limitée dans ce type de système. Ils peuvent être totalement inexpérimentés en ce qui concerne quelques-uns des aspects du système à développer, mais pas tous.
- **mode embarqué** : dans cette catégorie, Le projet à développer est une partie d'un système complexe et fortement connecté de matériels et de logiciels, de normes et de procédures opérationnelles. Sa principale caractéristique c'est que le système doit fonctionner sous des contraintes particulièrement fortes.
- **mode organique** : concerne des petites équipes travaillant dans un environnement qui leur est familier et un domaine d'application bien connu. Ceci étant, le coût de communication est négligeable, rapidité d'exécution du travail.

Les formules permettant de calculer le coût, ou plus exactement l'effort requis pour le développement du logiciel sont les suivantes :

- **Pour le mode organique** :  $HM = 2.4(KLSL)^{1.05}$
- **Pour le mode semi-détaché** :  $HM = 3(KLSL)^{1.12}$
- **Pour le mode semi-détaché** :  $HM = 3.6(KLSL)^{1.20}$

HM étant le nombre d'homme-mois nécessaire pour l'accomplissement d'un projet et KLSL le nombre de milliers d'instructions de l'application. Présupposons alors que notre logiciel est de mode organique, tout en sachant que notre application contient environ 15000 lignes de

code. Le KLSL correspond a 15, ce qui implique,  $HM = 2.4(15)^{1.05} = 41,220$  homme-mois. Ceci implique que le nombre d'hommes-mois nécessaire pour notre projet est de 41,220.

Le modèle COCOMO simple permet également d'estimer le temps de développement nécessaire du projet (TDEV), il est le temps essentiel pour achever le projet sans doute en admettant que le ressources de personnel essentielles sont accessibles, pour ces différents modes, les équations suivantes peuvent être utilisées :

- **Pour le mode organique :**  $TDEV = 2.5(HM)^{0.38}$
- **Pour le mode semi-détaché :**  $TDEV = 2.5(HM)^{0.35}$
- **Pour le mode semi-détaché :**  $TDEV = 2.5(HM)^{0.32}$

Le choix du mode pour notre projet a été porté sur le mode organique, le temps de développement nécessaire sera défini comme suit :  $TDEV = 2.5(41,220)^{0.38} = 9,86 = 10$  mois.

Le nombre de personnes nécessaires pour la réalisation de ce projet dans cet intervalle de temps est :  $N_p = HM/TDEV = 41,220/10 = 4,122 = 4$  personnes. Connaissant le nombre de personnes et le nombre de temps avec un coût que nous suggérons de 1000 FCFA par heure pour chaque personne, nous déduisons ce que suit :

- **Coût journalier :**  $= 1000fcfa * 8 \text{ heures} * 4 \text{ personnes} = 32000 \text{ FCFA},$
- **Coût mensuel :**  $= 32000FCFA * 26 \text{ jours} = 832000 \text{ FCFA},$
- **Coût annuel pour dix mois :**  $= 832000FCFA * 10 \text{ mois} = 8320000 \text{ FCFA}$

TABLE 2.4 – couts de realisation du projet

Designation	Total
Effort de développement (HM)	41,220.
Temps de développement (TDEV)	10 mois
Nombres de développeurs	4
Coûts de réalisation	8 320 000 FCFA

### 2.3.7 Planification

TABLE 2.5 – Caractéristiques de la solution voulue

N°	Libellé	Durée(jours)	Tâches antérieures
1	Définition du projet	2	/
2	Etude du projet	7	1
3	Analyse	7	2
4	Elaboration du plan de travail	2	3
5	Conception	40	4
6	Déploiement du logiciel	100	5
7	Contrôle et test	52	6
8	Rédaction des documents	5	7
9	Techniques	25	8
10	Chargement des dossiers	20	9

Suite à une première réunion avec le tuteur de stage en entreprise **M. Edmond TCHOUA**, des spécifications techniques ont été décidées pour ensuite servir à faire un planning de l'organisation du travail. Lors du déroulement du stage, le planning a été adapté en raison de divers changements :

- L'ajout de nouvelles fonctionnalités imprévues,
- Divers bugs durant le développement du projet.

Ces deux paramètres modifiant singulièrement le projet, il fallait souvent revoir l'architecture logicielle pour convenir aux nouvelles spécifications techniques, avant d'aboutir à une architecture bien plus ouverte qui permettrait d'ajouter des fonctionnalités sans tout changer .

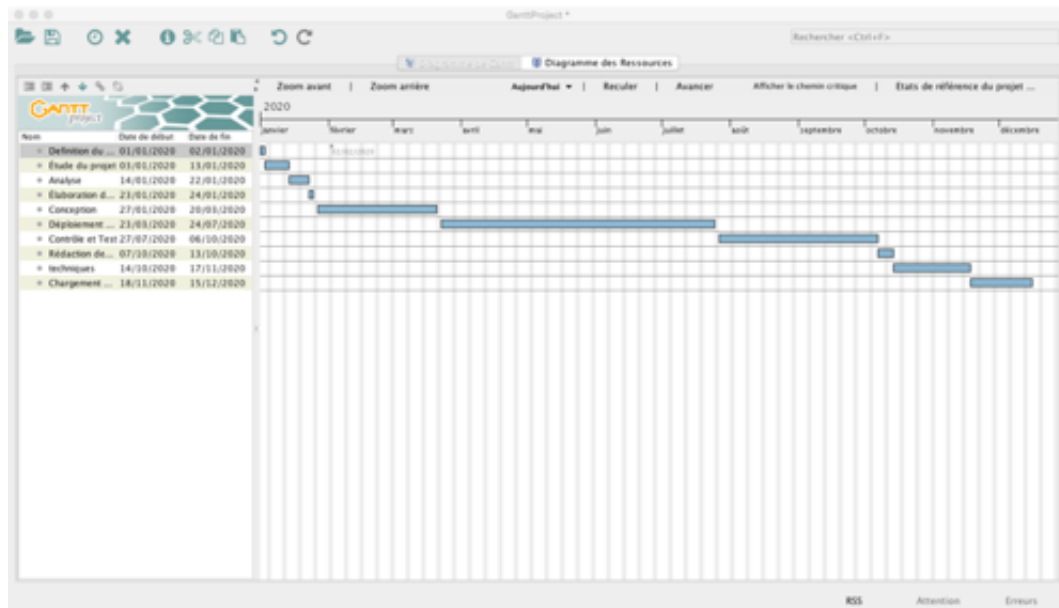


FIGURE 2.6 – Diagramme de Gantt.

### 2.3.8 Contraintes du projet

#### Contraintes de sécurité.

La gestion de la sécurité est la principale contrainte de notre système. L'application doit posséder une gestion de privilèges et niveaux d'accès pour les différents types d'utilisateurs. Selon leur statut, le contenu des pages varie et l'accès aux informations avec un statut supérieur est interdit.

#### Autres contraintes techniques

Pour le développement de notre système, nous disposons d'une architecture existante sur laquelle nous devrons baser notre application. La structure de notre système doit être extensible. De plus, le développement devra suivre toutes les normes techniques pour une meilleure performance, maintenance et facilité de mise à jour.

## 2.4 Méthodologie de conduite de projet.

### 2.4.1 Modèle de cycle de vie en cascade[10]

Mis au point dès 1966, puis formalisé vers 1970, le cycle de vie de projet en cascade est un type de cycle de vie, simple à comprendre et à implémenter, convient aux projets où la qualité a plus d'importance que les coûts ou les délais, et dont les besoins sont clairement

définis et stables. Dans le cas contraire, la prise en compte de nouveaux besoins nécessite de dérouler toute la cascade depuis le début. De plus, le client n'est impliqué qu'au début du projet et il ne peut tester le produit qu'à la fin du processus. Dans le cadre d'un projet de gestion des identités et des accès, les besoins peuvent évoluer. En effet, le déploiement de nouveaux services implique notamment la définition de nouveaux profils ainsi que de nouveaux rôles applicatifs qui doivent être pris en compte, même après la phase de spécification. De ce fait, ce modèle de gestion de projet ne convient pas à notre projet.

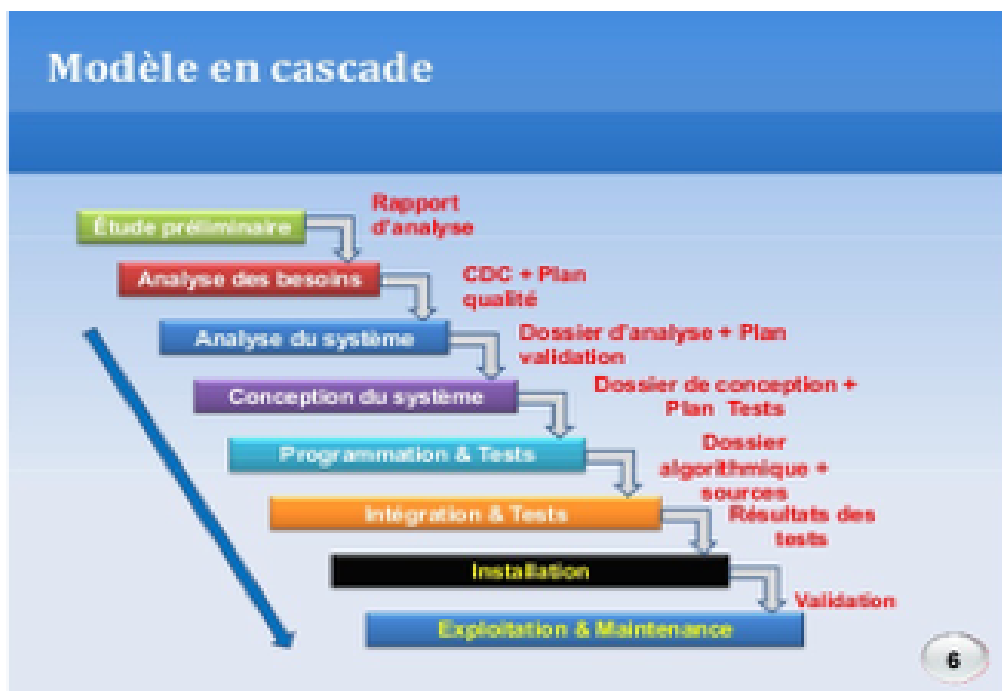


FIGURE 2.7 – Modèle de cycle de vie en cascade

source :Google

## 2.4.2 Modèle de cycle de vie en V[10]

A l'instar du modèle en cascade, celui en V prend difficilement en charge de nouveaux besoins ou la modification des spécifications. En effet, l'effet tunnel induit par les modèles séquentiels montre qu'une erreur dans la formulation ou l'interprétation des spécifications ne peut être détectée qu'à la fin du cycle. La maîtrise d'ouvrage n'est impliquée qu'en début et fin de cycle, ce qui peut représenter plusieurs mois d'intervalle pour un gros projet. Bien plus nombreuses que dans un cycle en V, les possibilités de prise en compte de nouveaux besoins restent faibles. En effet, dans le cadre de notre projet, après la phase de spécification, la mise à dispo-

sition d'un nouveau service, ne peut être prise en compte qu'au moment des tests d'intégration. De plus, ces changements impliqueraient la remise en cause du travail effectué jusqu'à la phase des tests unitaires.

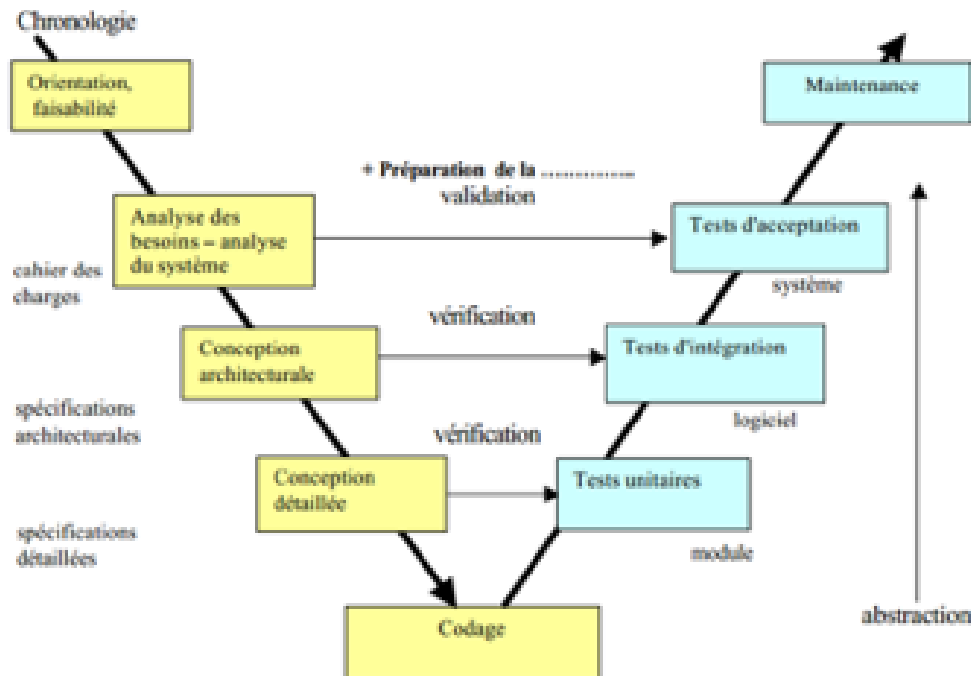


FIGURE 2.8 – Modèle de cycle de vie en V

source : *Google*

### 2.4.3 Modèle de cycle de vie en spirale [10]

Représenté à l'aide d'une spirale et proposé par Boehm en 1988, ce modèle est beaucoup plus général que le précédent. Chaque boucle de la spire représente une phase du développement celle la plus interne traite des premières phases. La plus externe traite de la livraison, chaque boucle traverse quatre sections :

- Définition des objectifs de la phase
- Évaluation des risques et plan de gestion
- Développement et validation
- Planification de la phase suivante



Le fait qu'il soit un méta-modèle entraine l'obligation de l'instanciation de chaque boucle, création d'une boucle de faisabilité, d'une boucle de prototypage, des boucles de développement itératif, etc.

De ce fait, il faut alors trouver le bon modèle de processus pour chaque boucle.

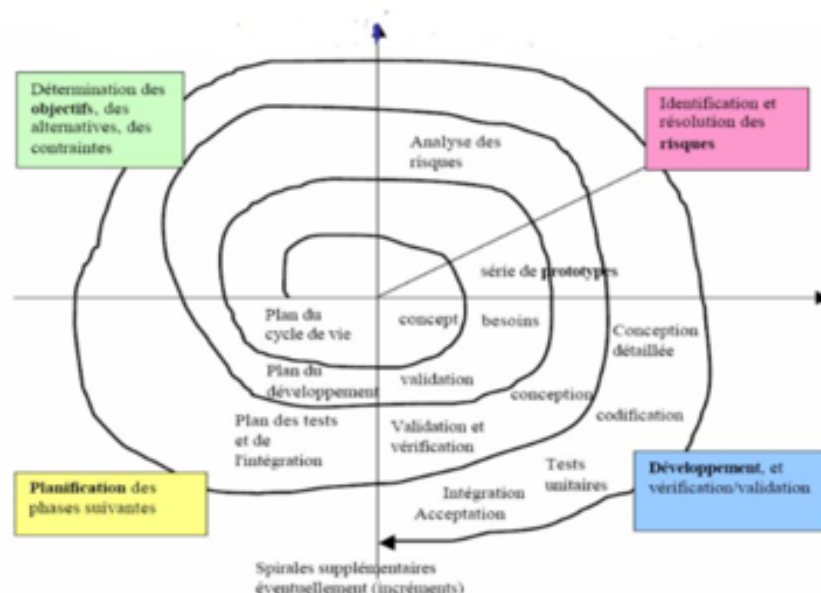


FIGURE 2.9 – Modèle de cycle de vie en spirale

source :*Google*

#### 2.4.4 La méthodologie agile [11]

Adopter les méthodes agiles constitue un acte courageux car il s'agit de quitter un système organisationnel, culturel et économique connu (étudié en cours et utilisé tout au long de la formation) pour s'orienter vers l'inconnu avant d'atteindre à nouveau un équilibre stable. Les raisons qui nous poussent à adopter une méthode agile sont nombreuses et variées. La capacité à s'adapter au changement, à livrer plus fréquemment et à accroître la qualité des logiciels ainsi développés figurent parmi les motivations les plus fréquentes. La bonne surprise réside dans le fait que la motivation des équipes est également une raison très souvent citée. Preuve est ainsi faite que les méthodes agiles représentent un système de valeurs et sont perçues par un grand nombre comme un changement culturel motivant, bénéfique pour tous.

L'agilité ou plutôt les méthodes agiles sont un groupe de processus et de pratiques pour le pilotage et la réalisation de projets. Toutes ces pratiques sont basés sur le manifeste agile, qui a

été mis en place en 2001 et qui a pour but d'impliquer au maximum le client, ou bénéficiaire du projet, dans le développement, pour permettre une réactivité dans la réalisation de ses demandes.

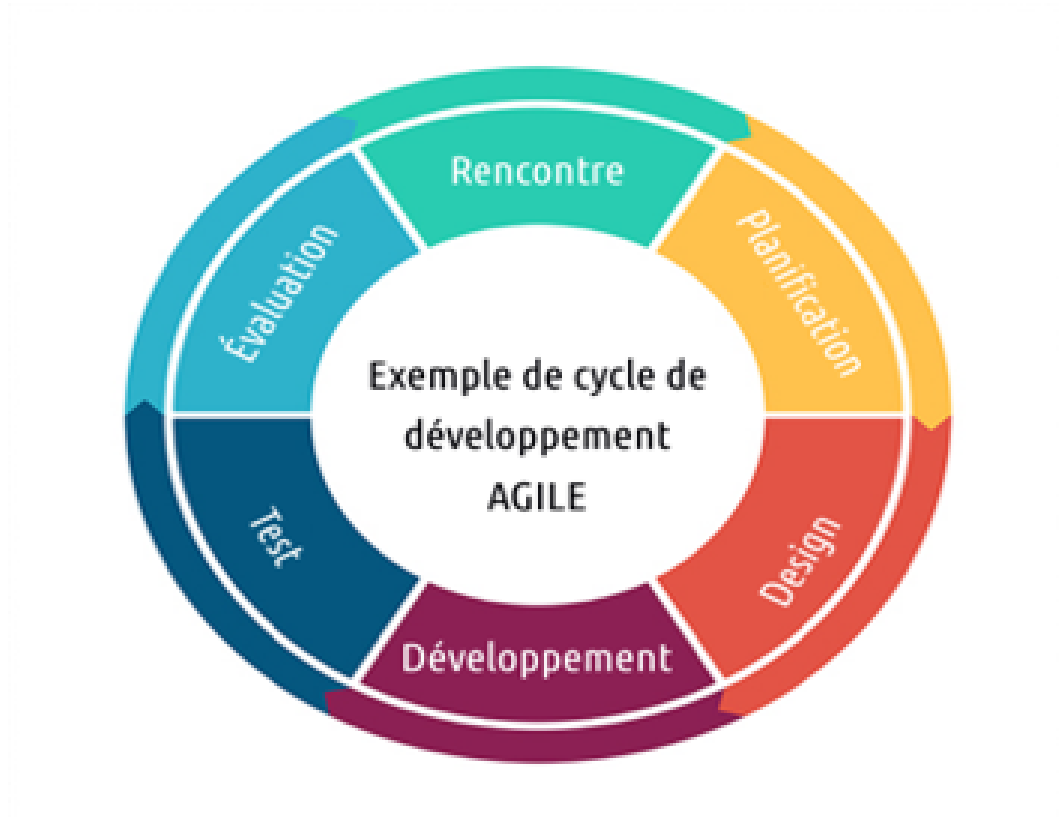


FIGURE 2.10 – Méthode agile.

source :*Google*

### 2.4.5 Pourquoi la méthodologie agile

La motivation de notre choix sur la méthode agile est contenue dans le tableau ci-dessous. Qui présente les principales différences entre les approches classique et les approches traditionnelles. Ces différences sont organisées par thèmes, lesquels correspondent aux principes du manifeste.

TABLE 2.6 – Méthodes : agiles vs classiques.[9]

Thème	MÉTHODES AGILES	MÉTHODES CLASSIQUES
Objectif	Satisfaire l'utilisateur	Respecter le besoin initial et les engagements
Changement	Accepter le changement	Opposé au changement ou, en tout cas, moins enclin à l'accepter compte tenu des livraisons tardives et des processus de gestion lourds
Livraison	Livrer fréquemment	Livrer en une seule fois une application « finalisée »
Équipe	Travailler en synergie	Travailler de façon segmentée (chacun voit sa partie du travail)
Moteur	Stimuler la motivation	Stimuler la productivité
Communication	Communiquer en direct avec les opérationnels	Communiquer de façon verticale en passant par des relais hiérarchiques (par exemple, le chef de projet MOE est relais entre la MOA et ses développeurs)
Indicateurs	Un seul indicateur : les fonctionnalités implémentées	Justifier par les indicateurs (sans livraison intermédiaire, les indicateurs sont les seuls justificatifs de l'avancement, des écarts, etc.)
Rythme	Bannir les rushs de production	Adapter la production aux contraintes projets
Livrables	Rester concentré sur l'essentiel (la production)	Une documentation précise est essentielle pour assurer les échanges et la validation autour des besoins client
Autonomie	Favoriser une certaine autonomie des équipes	Encadrer scrupuleusement le travail des équipes
Rythme	Intégrer la notion d'amélioration continue tout au long du projet	Introspection possible mais uniquement en fin de projet

source : <https://www.consultrade.info/gestion-de-projet/la-gestion-de-projet-methode-classique-vs-agiles/>

## CHAPITRE 3

## ANALYSE ET CONCEPTION.

### 3.1 Présentation de la méthodologie.

#### 3.1.1 Méthodologie adoptée Scrum

Suite à une étude comparative, nous avons remarqué que les approches agiles sont plus adaptés à notre projet que toutes les approches traditionnelle. Bien que agile soit l'approche choisi, il faut noter que cet approche contient plusieurs méthodes parmi lesquelles :

- Scrum/XP Hybrid
- Agile Unified Process (AgileUp)
- Scrumban
- Iterative Development
- Lean Development
- Scrum

Cependant, la méthodologie Scrum de l'approche agile est celle qui a retenu notre attention en raison des points suivants :

- Scrum convient aux équipes ayant un nombre de développeurs réduits. Ceci est notre cas
- Le client est impliqué dans le développement de l'application : La consultation du client est nécessaire dès l'achèvement d'une tâche

- La progression des tâches s'effectue pendant une durée de développement courte

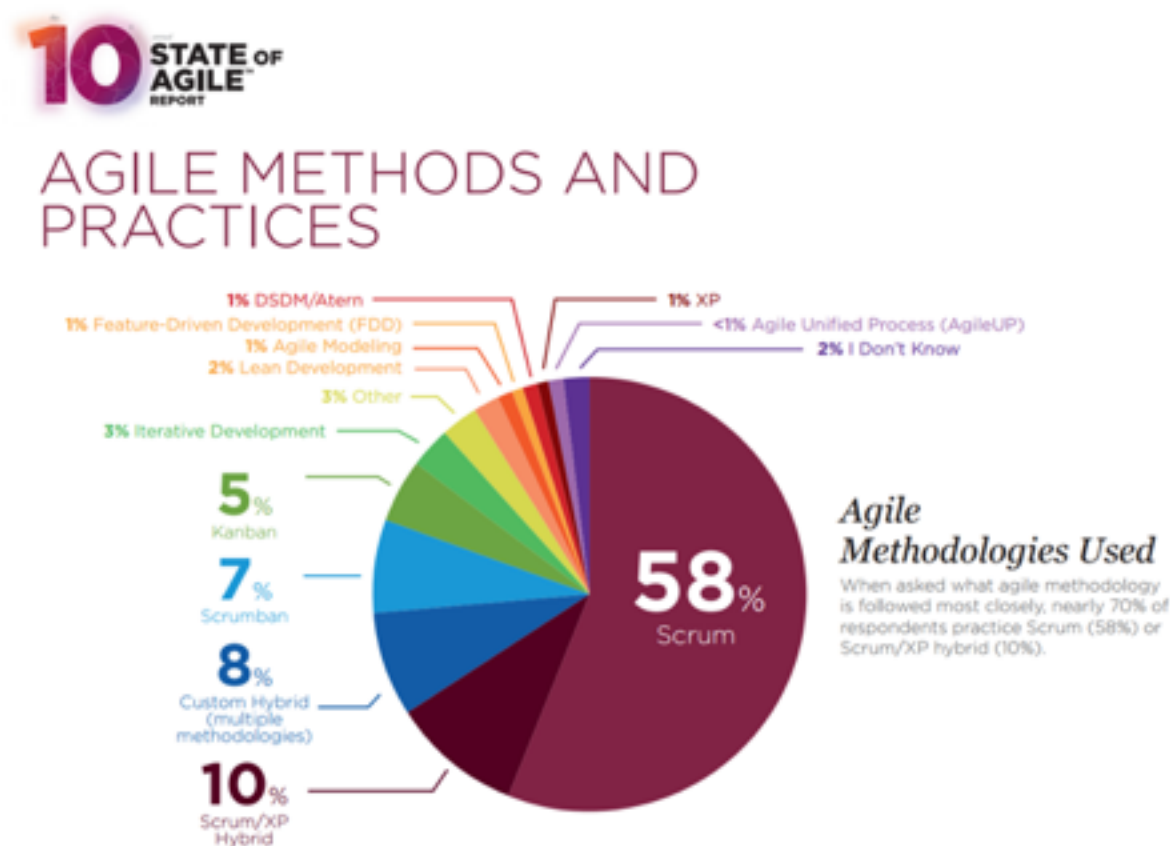


FIGURE 3.1 – Comparatif des méthodes agiles.

D'après la figure ci-dessus, Scrum est la méthodologie se basant sur les préceptes agiles la plus utilisés dans le monde. Il s'agit d'un recueil de bonne pratique, de définition de rôle et de cérémonies récurrentes rythmant le projet.

1. **Présentation de Scrum.** La méthodologie Scrum a été conçue pour améliorer grandement la productivité dans les équipes auparavant paralysés par des méthodologies plus lourdes. Le principe de base de Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe allant d'une à quatre semaines, appelées SPRINT.

Chaque sprint possède un but à atteindre, défini par le directeur de produit, à partir duquel sont choisies les fonctionnalités à implémenter dans le sprint. Un sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel. Pendant ce temps, le Scrum Master a la charge de réduire au maximum les perturbations extérieurs et de résoudre les problèmes non techniques de l'équipe. Ce processus est illustré par la figure suivante

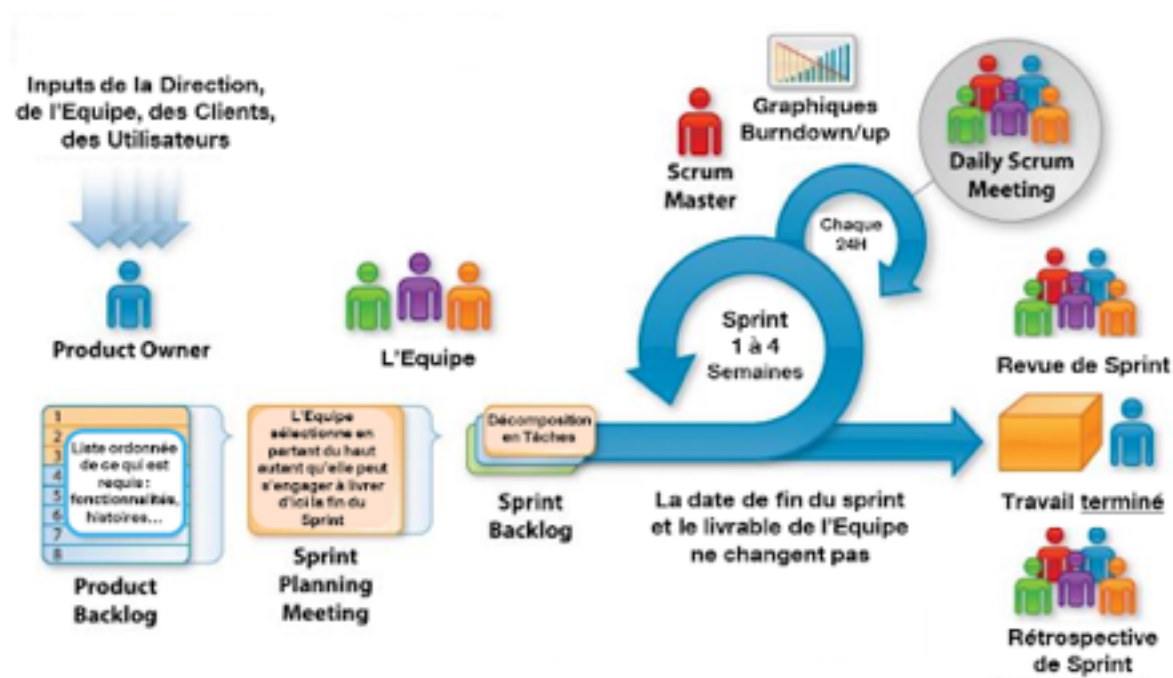


FIGURE 3.2 – Processus SCRUM.

### Les acteurs.

On distingue plusieurs acteurs, dont : Le directeur de produit product owner est le représentant des clients et des utilisateurs, et fait également parti de l'équipe.

Le ScrumMaster veille à l'application de la méthodologie Scrum au sein de l'équipe.

L'équipe qui contribue à la réalisation des fonctionnalités du projet (planification, développement, test et documentation). A noter que les membres de l'équipe travaillent tous ensemble : Chaque membre peut faire ainsi des propositions, exprimer idées et écoute les autres.

### Le processus.

Tous les critères ou exigences du produits sont regroupées dans les journaux ou backlogs dont on distingue 2 types :

- Le backlog de produit ou « Product backlog » qui regroupe la liste de fonctionnalités du produit.
- Le backlog de sprint ou « Sprint backlog » en fonction des fonctionnalités du produit, regroupe la liste des tâches qui devra être réalisées à l'itération en cours. Chaque tâche aurait fait l'objet d'une estimation préalable de charge par l'ensemble des membres de l'équipe afin d'estimer au mieux les taches qui peuvent réaliser un sprint.

### La planification.

Le sprint : Dès le début d'un projet, la première planification permet de définir le périmètre de chaque itération appelé sprint. Chaque sprint dure quelques semaines et regroupe une liste de tâches (défini dans le backlog).

La mêlée quotidienne : De plus, elle est rythmée par ce qu'on appelle une mêlée quotidienne d'un quart d'heure qui consiste chaque jour avec les membres de l'équipe ainsi que le directeur de produit de se tenir au courant de l'avancement du projet, notamment en :

- Faisant le point sur le travail effectué la veille par chacun
- Définissant les tâches qui sont réalisées durant la journée
- Résolvant les éventuels problèmes qui avaient ou qui pourraient être rencontré par chacun
- Le développement suit un processus itératif et incrémental : de nouvelles fonctionnalités sont rajoutées au produit
- Lean Development
- Scrum

#### **La revue du sprint.**

La fin d'un sprint aboutit à la réalisation d'un produit avec des fonctionnalités partielles avec la documentation associée. Dans la plupart des cas, cela conduit à une revue de sprint consistant à faire une démonstration de la réalisation du sprint devant le client afin de valider le travail réalisé et d'avoir un retour pour éventuellement ajuster le backlog du produit.

## **3.2 Modèle.**

Modéliser consiste à créer une représentation virtuelle d'une réalité de manière à faire ressortir les points auxquelles on s'intéresse, dans notre domaine, deux approches se démarquent : MERISE et UML

TABLE 3.1 – Comparaison entre Merise et UML

MERISE	UML
Méthode d'analyse et de conception d'un système d'information	Langage de représentation d'un SI.
Méthode de modélisation de données et traitements orienté bases de données relationnelles	Système de notation orienté objet
relationnel	Objet
Franco-Français	Internationale
Schéma directeur, étude préalable, étude détaillée et la réalisation	Langage de modélisation des systèmes standards, qui utilise des diagrammes pour représenter chaque aspect d'un système : statique, dynamique. En s'appuyant sur la notion d'orienté objet.

### 3.2.1 Choix du langage de modélisation.[12]

UML est un langage de modélisation graphique à base de pictogrammes, conçu pour représenter, spécifier les artefacts de systèmes logiciels. Il est destiné à comprendre et décrire des besoins spécifiés et documentés des systèmes, esquissé des architectures logicielles. Ce langage de modélisation comble une lacune importante des technologies objet, il permet d'exprimer, d'élaborer et de modéliser au sens de la théorie des langages, de ce fait il contient les éléments constitutifs de ce derniers : concepts, syntaxe et sémantique. La puissance et l'intérêt d'UML est qu'il normalise la sémantique des concepts qu'il véhicule, il repose sur un méta modèle pour permettre à n'importe qui de déchiffrer son intention de manière non équivoque, il est donc primordiale de s'accorder sur la sémantique des éléments de modélisation, bien avant de s'intéresser à la manière de les présenter.

Les points forts suivant sont ceux ayant motiver notre choix :

- Gain de précision
- Gage de stabilité
- Utilisation d'outils
- Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes.

Son caractère polyvalent et sa souplesse en font un langage universel.



## 3.3 Concept d'UML

### 3.3.1 Définition

Langage de modélisation unifié « Unified Modeling Language » est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

### 3.3.2 Unité

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture de logicielle. Les différents éléments représentables sont « Activité d'un objet/logiciel, Acteurs, Processus, Schéma de base de données, Composants logiciels et Réutilisation de composants ». Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple en langage Java, à partir des divers documents réalisés.

### 3.3.3 Présentation de quelques diagrammes UML

#### Diagramme de cas d'utilisation

Les cas d'utilisation constituent un moyen de recueillir et de décrire les besoins des acteurs du système. Ils peuvent être aussi utilisés ensuite comme moyen d'organisation du développement du logiciel, notamment pour la structuration et le déroulement des tests du logiciel. Un cas d'utilisation permet de décrire l'interaction entre les acteurs (utilisateurs du cas) et le système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur. La représentation d'un cas d'utilisation met en jeu trois concepts : l'acteur, le cas d'utilisation et l'interaction entre l'acteur et le cas d'utilisation.

- **Acteur** : Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation. Ainsi les utilisateurs d'un système appartiennent à une ou plusieurs classes d'acteurs selon les rôles qu'ils tiennent par rapport au système. Un acteur peut

aussi être un système externe avec lequel le cas d'utilisation va interagir. Un acteur peut se représenter symboliquement par un bonhomme et être identifié par son nom.

- **Cas d'utilisation et interaction :**

- Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur. Un cas d'utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système.
- Une interaction permet de décrire les échanges entre un acteur et un cas d'utilisation.

- **Relations entre cas d'utilisation :** Afin d'optimiser la formalisation des besoins en ayant recours notamment à la réutilisation de cas d'utilisation, trois relations peuvent être décrites entre cas d'utilisation : une relation d'inclusion (include), une relation d'extension (extend) et une relation de généralisation.

### **Diagramme de séquences**

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.

- **Ligne de vie :**

Une ligne de vie représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour.

- **Message synchrone et asynchrone :**

- Message synchrone – dans ce cas, l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.
- Message asynchrone – dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

UML possède plus de diagramme que celles présenté ci-haut. Nous ne pouvons pas les présenter tous dans ce mémoire, de peur que cela soit long et que beaucoup de livre explique

déjà toutes ses notions de façon claire avec des exemples à l'appui. Pour plus d'information, nous conseillons la consultation des livres UML, Analyse et Conception et UML 2 pour les développeurs qui ont été notre principale source d'inspiration dans cette partie.

**Diagramme de classes** Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations.

La description du diagramme de classe est fondée sur :

- Le concept d'objet

Un objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer à priori les valeurs de ses propriétés. Un objet est caractérisé par les valeurs de ses propriétés qui lui confèrent des états significatifs suivant les instants considérés.

- Le concept de classe comprenant les attributs et les opération

Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition). Un objet est une instance d'une classe. La classe représente l'abstraction de ses objets. Au niveau de l'implémentation, c'est-à-dire au cours de l'exécution d'un programme, l'identificateur d'un objet correspond à une adresse mémoire.

- Les différents types d'association entre classes.

La généralisation est la relation entre une classe et deux autres classes ou plus partageant un sous-ensemble commun d'attributs et/ou d'opérations. La classe qui est affinée s'appelle superclasse, les classes affinées s'appellent sous-classes. L'opération qui consiste à créer une superclasse à partir de classes s'appelle généralisation. Inversement la spécialisation consiste à créer des sous-classes à partir d'une classe.

## 3.4 Analyse et Modélisation

### 3.4.1 Les acteurs système

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système. Les acteurs qui interagissent avec l'application à concevoir sont :

- Le responsable du content-writer : consultation de l'application, modification des doubles bannières, modification des sliders de l'application
- Le responsable marketing : consultation de l'application, la création, modification des notifications push, la création/modification des pop-ups
- Le client : consultation de l'application
- L'informaticien : paramétrage de l'application et création des différents utilisateurs
- Le responsable de la planification commerciale : consultation de l'application, la mise en place des bloc produits et catégories

### 3.4.2 Les cas d'utilisation

#### Identification des cas d'utilisation :

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent comment le système devrait interagir avec les utilisateurs (appelés acteurs) pour atteindre un but ou une fonction spécifique d'un travail. Un acteur d'un cas d'utilisation peut être un humain ou un autre système externe à celui que l'on tente de définir.

Lors de notre analyse des besoins nous avons pu identifier les actions importantes que nous présenterons ci-dessous et nous les modélisons par la suite avec les diagrammes cas d'utilisations d'UML.

TABLE 3.2 – Acteurs du système.[9]

Cas d'utilisation	Acteurs	Opérations
Authentification	Tous les utilisateurs	Connexion, Déconnexion
Gestion des sliders	Responsable content-writer	Ajouter, Modifier, Supprimer, Consulter
Gestion des doubles bannières	Responsable content-writer	Ajouter, Modifier, Supprimer, Consulter
Gestion des blocs produits	Responsable de la planification commercial	Ajouter, Modifier, Supprimer, Consulter
Gestion des push notification	Responsable marketing	Ajouter, Modifier, Supprimer, Consulter
Gestions des paramètres	Informaticien	Ajouter, Modifier, Supprimer, Consulter
Gestion des utilisateurs	Informaticien	Ajouter, Modifier, Supprimer, Consulter

**Identification des cas d'utilisation :**

Le diagramme de cas d'utilisation est le diagramme UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

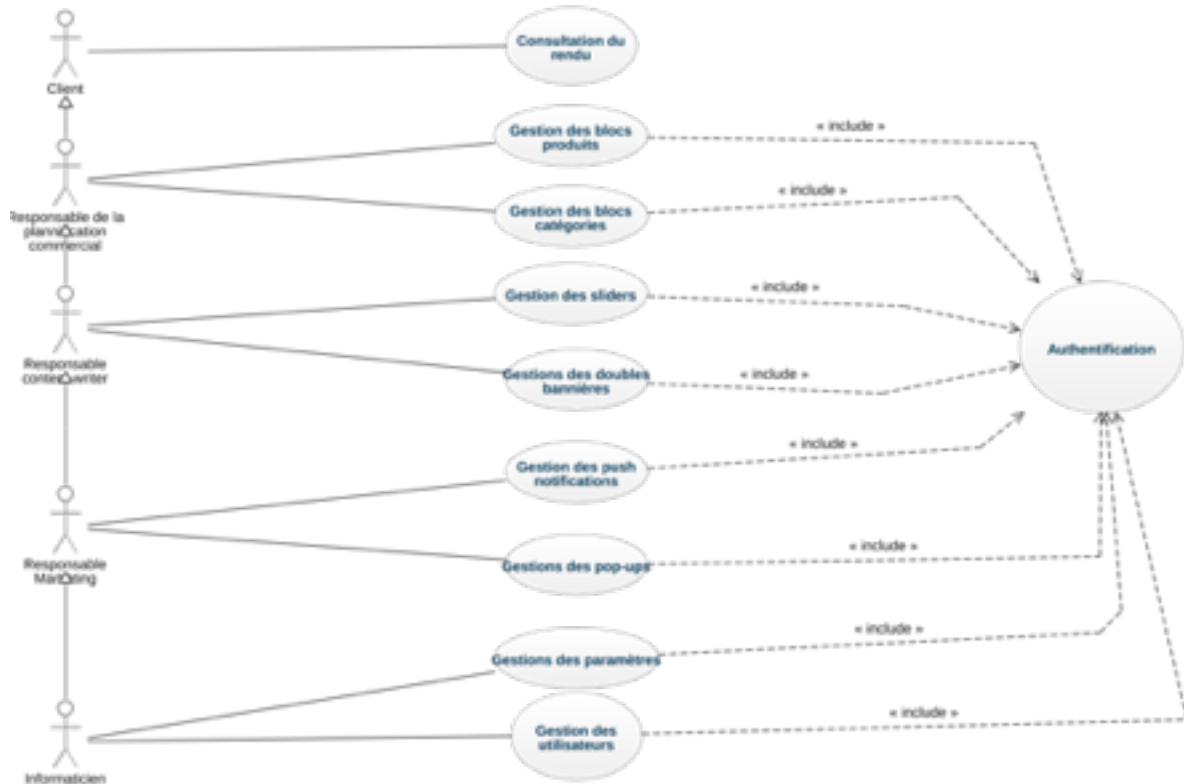


FIGURE 3.3 – Diagramme de cas d'utilisation du système.

**Description de quelques cas d'utilisation :**

Nous allons désormais parler de l'interaction entre les acteurs et le système : il s'agit de décrire la chronologie des actions qui devront être réalisées par les acteurs et par le système lui-même. On parle d'ailleurs de scénarios.

**1. Description du cas d'utilisation "Gestion des blocs produits".**

TABLE 3.3 – Description du cas d'utilisation "Gestion des blocs produits"

	Description
Titre	Gestion des blocs produits
But	Il permet de mettre sur la page d'accueil les produits voulu afin de présenter une variété de produits à un utilisateur
Acteur	Responsable de la planification commercial
Pré conditions	L'utilisateur doit s'authentifier
Enchainement	L'utilisateur s'authentifie, L'utilisateur demande d'ajouter, modifier ou supprimer les produits mis en avant, le système affiche le formulaire ou la liste des produits, le système verifie la validité des données, enregistre et confirme
Alternative	le système affiche un message d'erreur lorsque l'utilisateur fournit des données incomplètes ou erronées ( ajout et modification)

## 2. Description du cas d'utilisation "Gestion des doubles bannieres".

TABLE 3.4 – Description du cas d'utilisation "Gestion des doubles bannieres"

	Description
Titre	Gestion des doubles bannieres
But	Il permet de mettre sur la page d'accueil les doubles bannieres voulu afin de présenter une variété de catégories et ou boutiques officielles à un utilisateur
Acteur	Responsable content-writer
Pré conditions	L'utilisateur doit s'authentifier
Enchainement	L'utilisateur s'authentifie, L'utilisateur demande d'ajouter, modifier ou supprimer les produits mis en avant, le système affiche le formulaire ou la liste des doubles bannieres, le système verifie la validité des données, enregistre et confirme
Alternative	le système affiche un message d'erreur lorsque l'utilisateur fournit des données incomplètes ou erronées ( ajout et modification)

## 3. Description du cas d'utilisation "Gestion des push notifications".

TABLE 3.5 – Description du cas d'utilisation "Gestion des push notifications"

	Description
Titre	Gestion des push notifications
But	Il permet d'effectuer les notifications sur l'application des utilisateurs afin de les annoncer nos meilleurs deals ou nos promotions
Acteur	Responsable Responsable marketing
Pré conditions	L'utilisateur doit s'authentifier
Enchaînement	L'utilisateur s'authentifie, L'utilisateur demande d'ajouter, modifier ou supprimer les produits mis en avant, le système affiche le formulaire ou la liste des push notifications, le système vérifie la validité des données, enregistre et confirme
Alternative	le système affiche un message d'erreur lorsque l'utilisateur fournit des données incomplètes ou erronées ( ajout et modification)

#### 4. Description du cas d'utilisation "Gestion des sliders".

TABLE 3.6 – Description du cas d'utilisation "Gestion des sliders"

	Description
Titre	Gestion des sliders
But	Il permet de mettre sur la page d'accueil les sliders pour présenter nos multiples catégories et ou boutiques officielles
Acteur	Responsable Responsable content-writer
Pré conditions	L'utilisateur doit s'authentifier
Enchaînement	L'utilisateur s'authentifie, L'utilisateur demande d'ajouter, modifier ou supprimer les produits mis en avant, le système affiche le formulaire ou la liste des sliders, le système vérifie la validité des données, enregistre et confirme
Alternative	le système affiche un message d'erreur lorsque l'utilisateur fournit des données incomplètes ou erronées ( ajout et modification)

### 3.4.3 Modélisations Conceptuelles

#### Diagramme de séquence..



Les diagrammes de séquences permettent décrire comment les éléments du système interagissent entre eux et avec les acteurs. Les objets au cœur d'un système interagissent en s'échangeant des messages. Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

Quelques diagrammes de séquence de l'application.

1. Diagramme de séquence "S'authentifier".

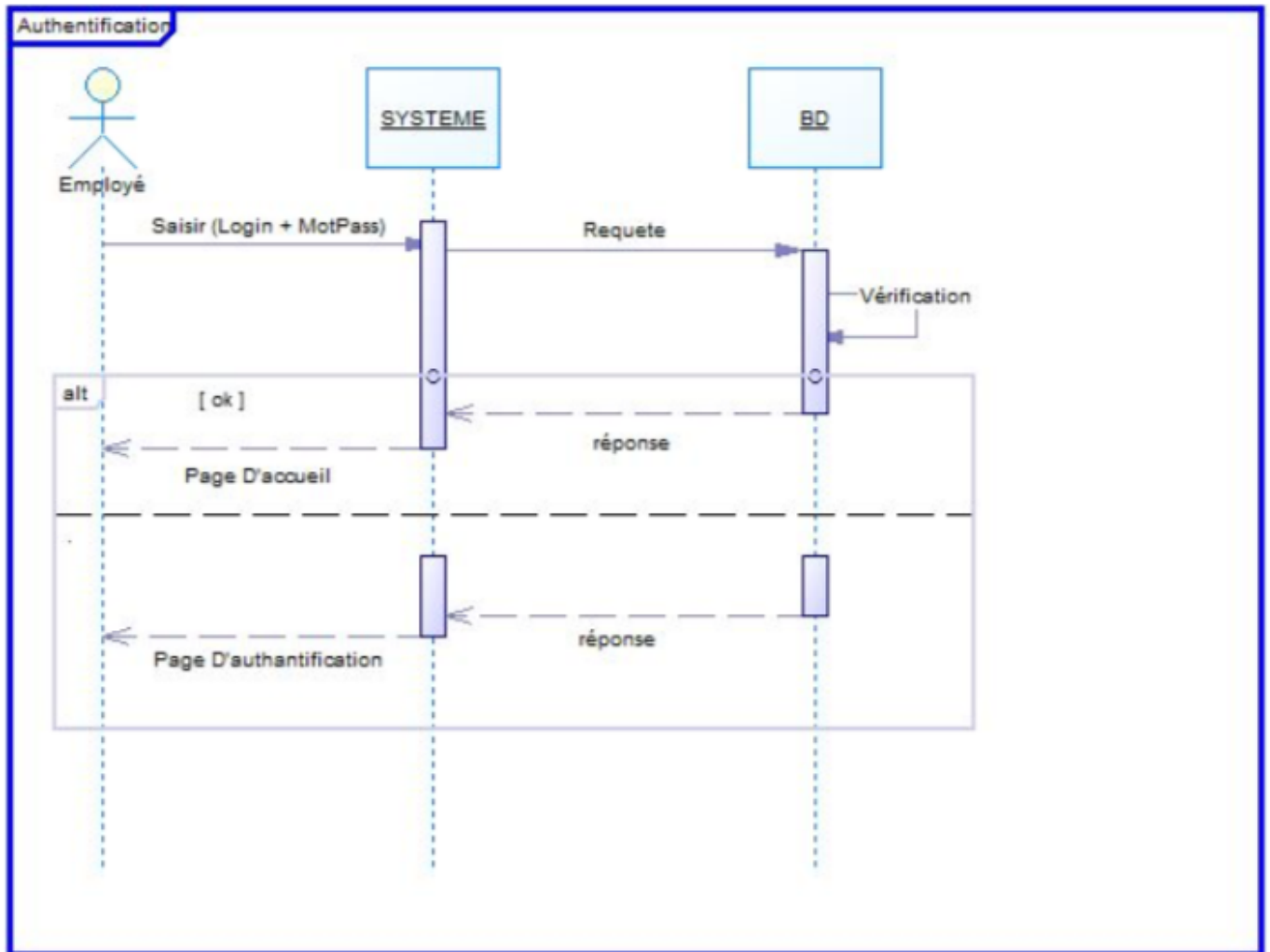


FIGURE 3.4 – Diagramme de séquence d'authentification.

2. Diagramme de séquence "Gestion des doubles bannières".

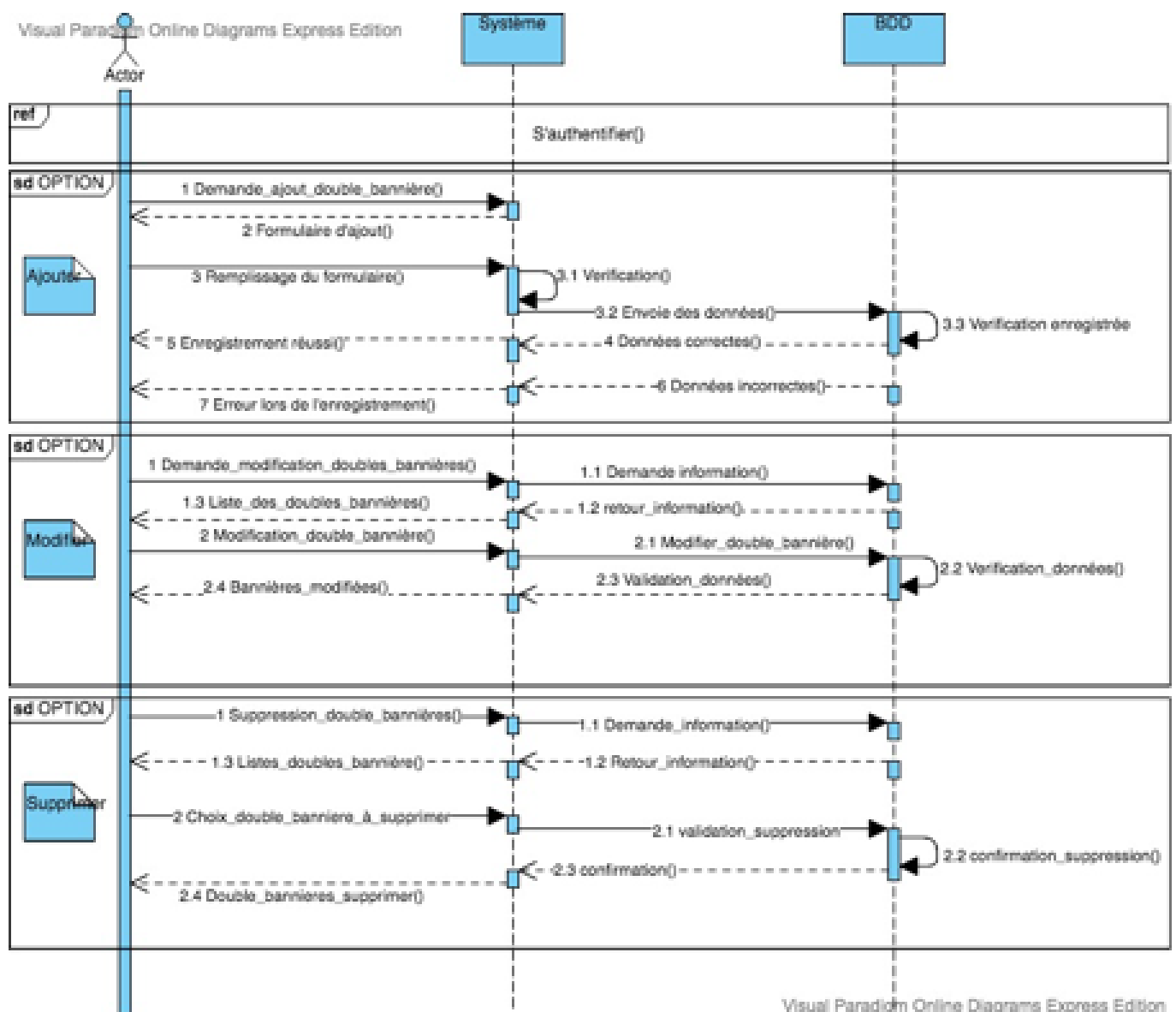


FIGURE 3.5 – Diagramme de séquence "Gestion des doubles bannières".

### 3. Diagramme de séquence "sliders".

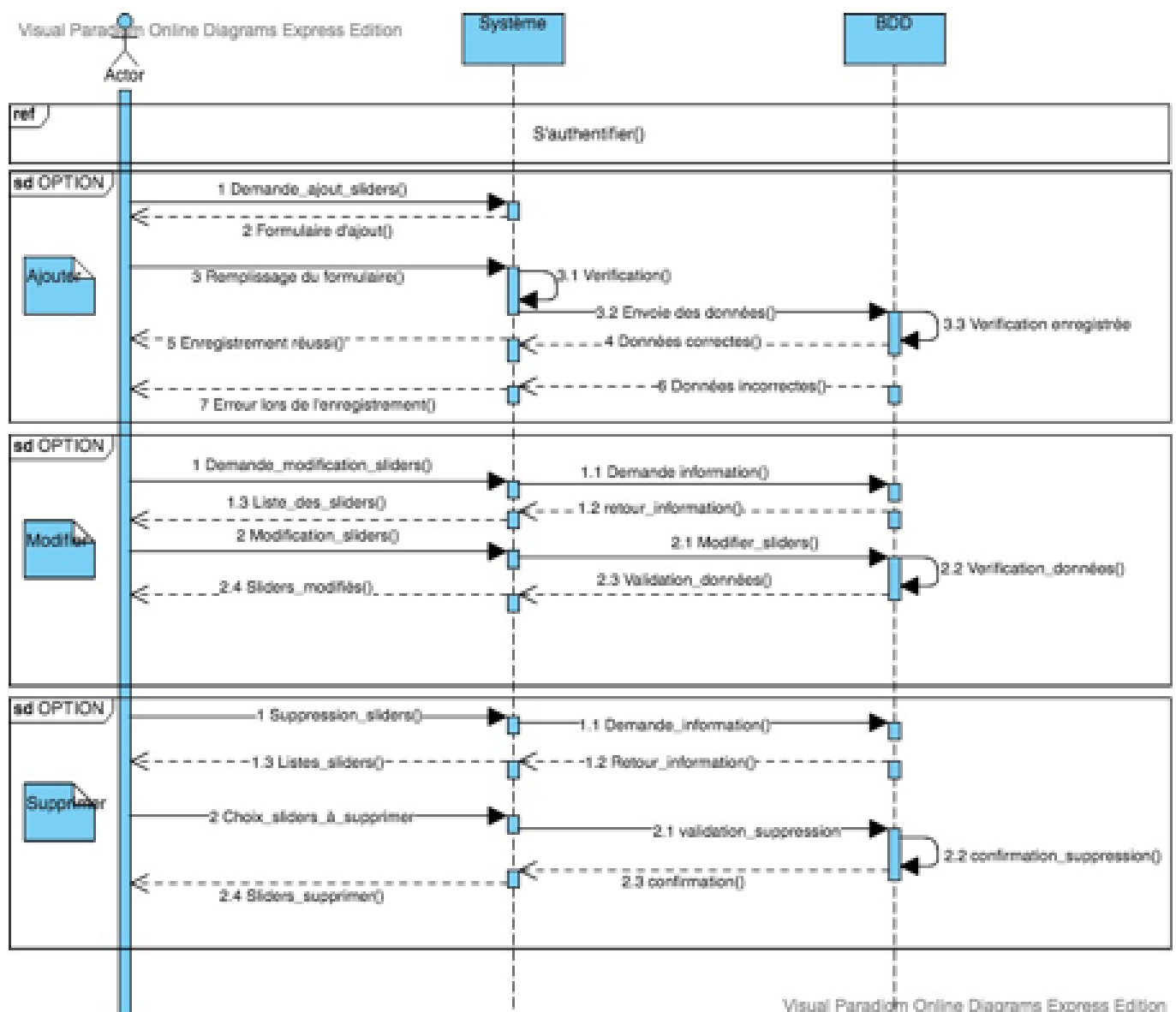


FIGURE 3.6 – Diagramme de séquence "sliders".

4. Diagramme de séquence "Gestion des push notifications".

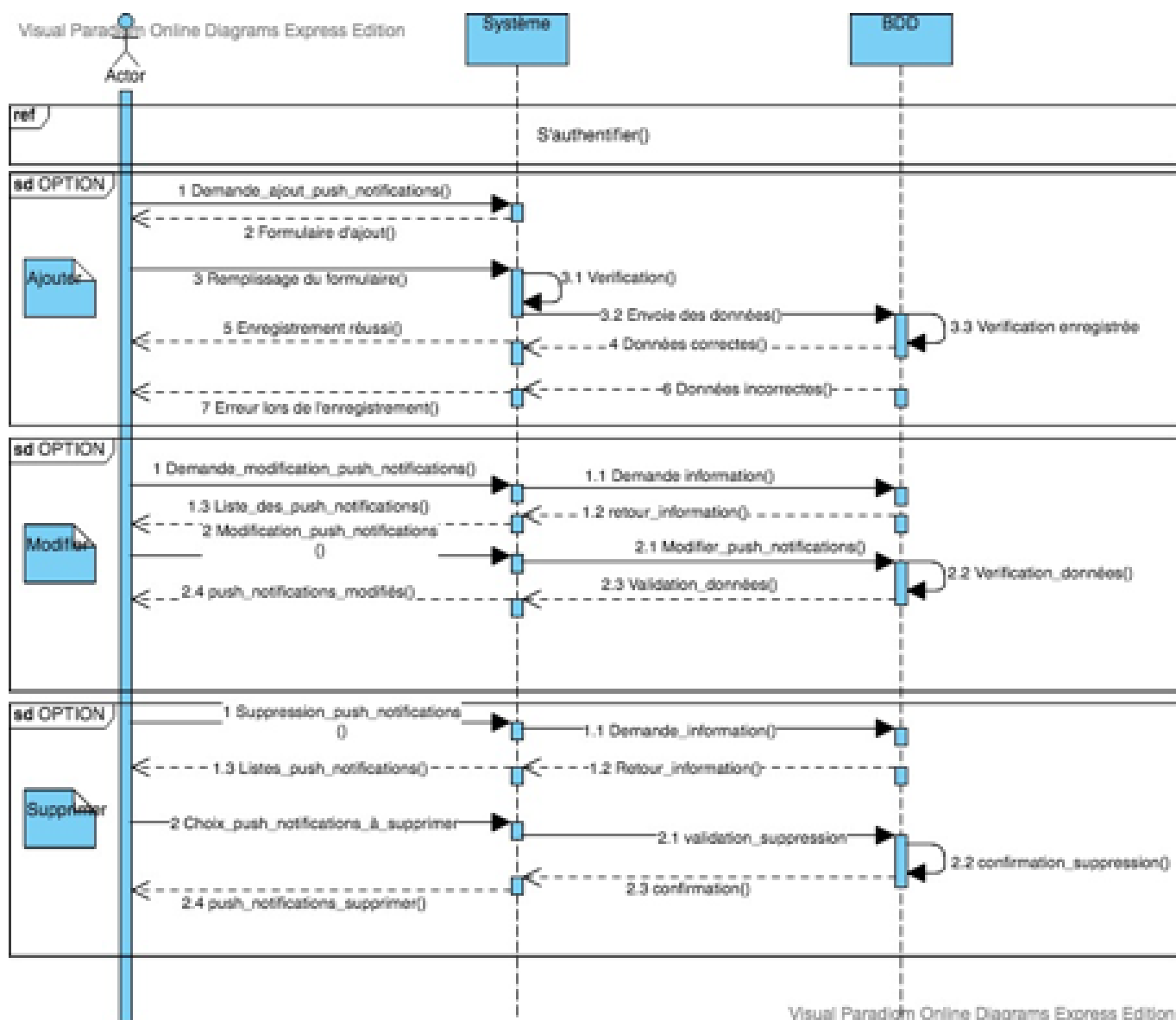


FIGURE 3.7 – Diagramme de séquence "Gestion des push notifications".

**Diagramme de classe..**

Un diagramme de classes UML décrit les structures d'objets et d'informations utilisées par votre application, à la fois en interne et dans la communication avec ses utilisateurs. Il décrit les informations sans faire référence à une implémentation particulière. Ses classes et relations peuvent être implémentées de nombreuses manières, comme les tables de bases de données, les nœuds XML ou encore les compositions d'objets logiciels.

Pour un soucis de confidentialité il m'a été demandé de ne présenter que les classes qui interviennent avec mon mémoire.

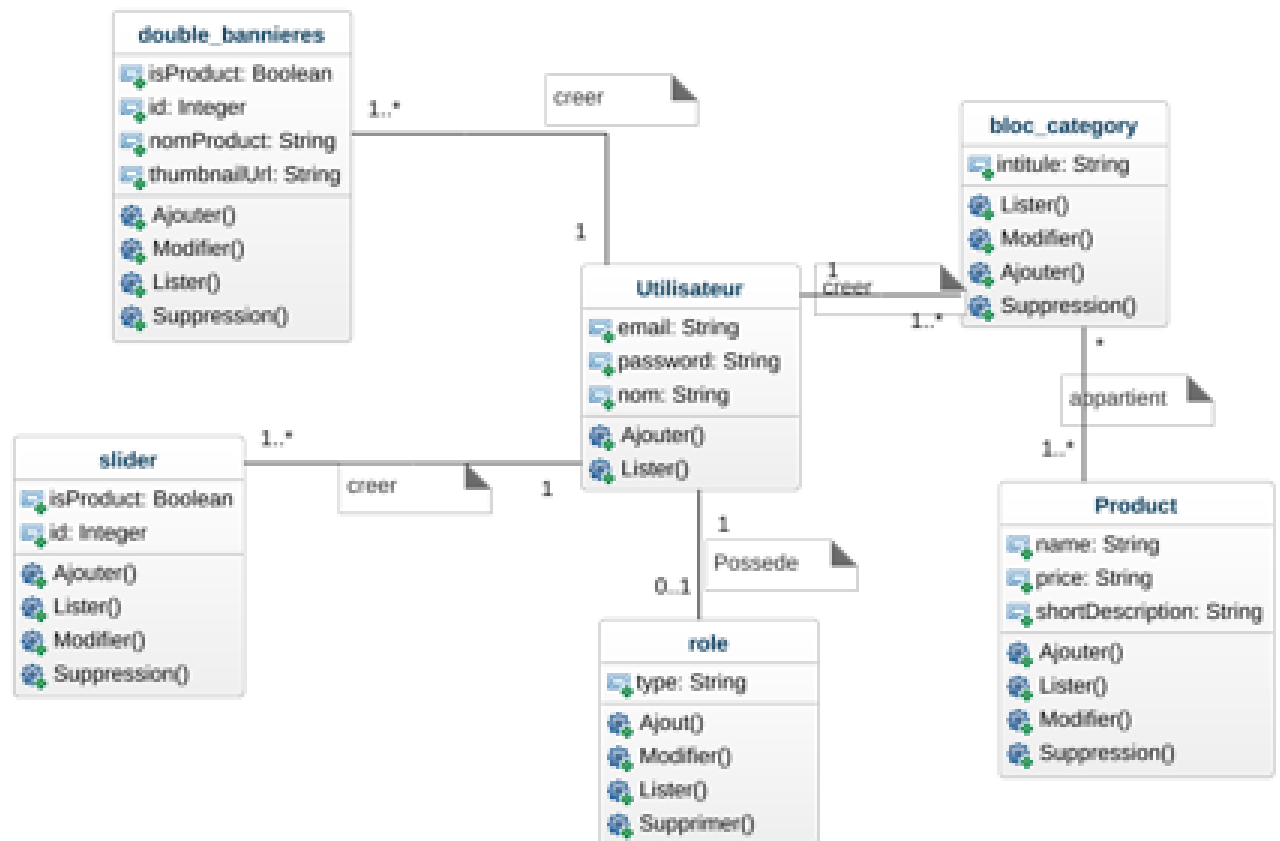


FIGURE 3.8 – Diagramme de classe .

**Diagramme de déploiement..**

Le diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que la relation entre eux.



FIGURE 3.9 – Diagramme de déploiement.

## 3.5 Architecture générale de l'application

### 3.5.1 Définition

Il existe plusieurs types d'architectures, parmi lesquelles :

1. L'architecture 2-tiers : L'architecture à deux niveaux (aussi appelée architecture 2-tier, tiers signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service

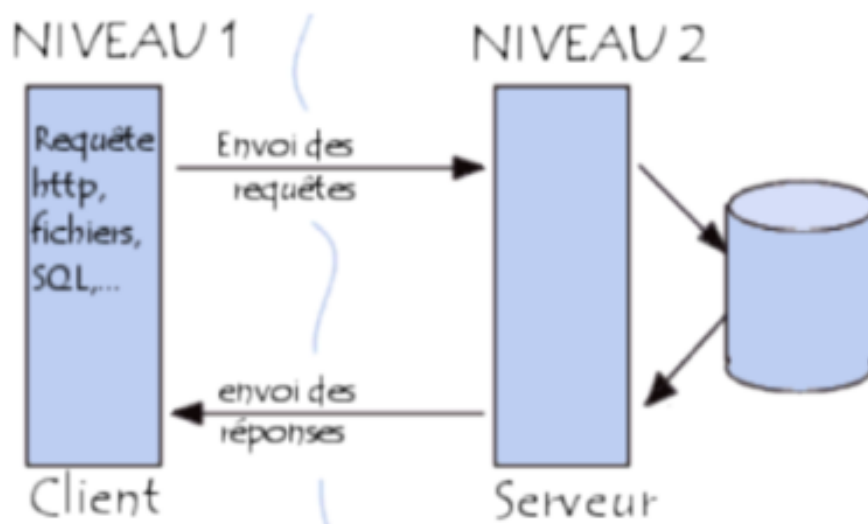


FIGURE 3.10 – Architecture 2-tiers.

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

- Client : c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation
- Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur
- Le serveur de données, fournissant au serveur d'application les données dont il a besoin

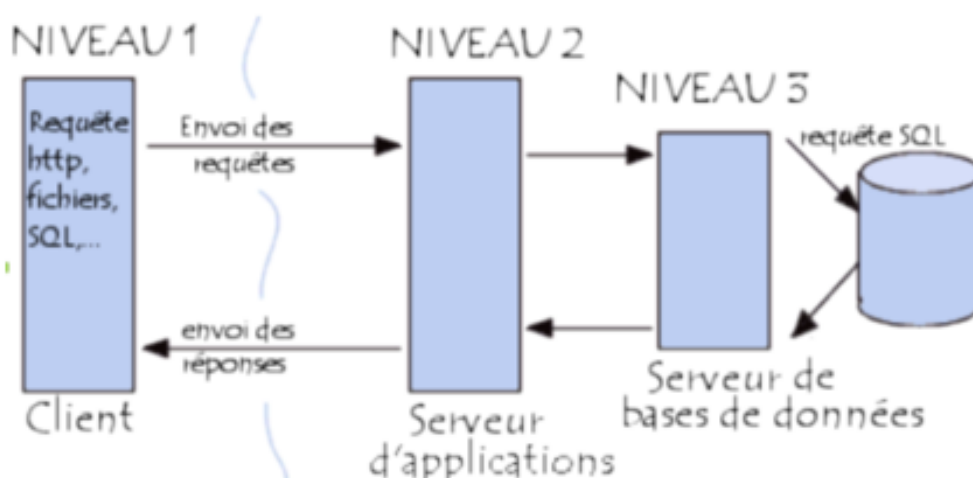


FIGURE 3.11 – Architecture 3-tiers.

3. L'architecture N-tiers : Dans l'architecture à 3 niveau, chaque serveur ( niveau 2 et 3 ) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux.

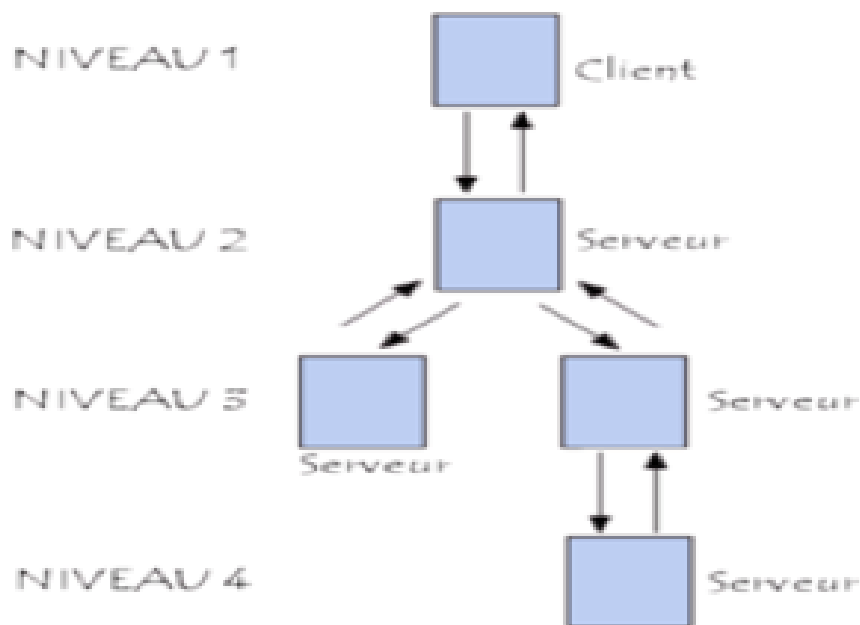


FIGURE 3.12 – Architecture N-tiers.

### 3.5.2 Architecture de notre système

Dans notre cas, il s'agit d'une architecture 3-tiers, c'est-à-dire à 3 niveaux :

- Le niveau 1 : il s'agit de l'interface utilisateur où sera lancée une requête.
- Le niveau 2 : il s'agit dans notre cas du serveur de l'établissement
- Le niveau 3 : il s'agit du serveur SQL Server

Tout système d'information nécessite la réalisation de trois groupes de fonctions : le stockage des données, la logique applicative et la présentation. Ces trois parties sont indépendantes les unes des autres : on peut ainsi vouloir modifier la présentation sans modifier la logique applicative. La conception de chaque partie doit également être indépendante, toutefois la conception de la couche la plus basse est utilisée dans la couche d'au-dessus.

Dans ce chapitre nous avons présenté d'une façon globale, les deux étapes essentielles du système élaboré pour l'analyse et la conception de notre application en suivant le processus de



normalisation UML et les différents diagrammes, afin de faciliter la phase de réalisation. Le chapitre suivant, quant à lui, sera consacré à la phase de développement de notre application.

## CHAPITRE 4

## RÉSULTATS ET DISCUSSION

Après avoir réalisé une conception qui répondait bien aux besoins de l'application, nous entamons la partie amélioration de l'application IZIWAY CAMEROUN, en exposant les différents outils de langages de développement utilisés lors de la réalisation et l'implémentation de la base de données ainsi qu'un aperçu sur les interfaces de notre application.

### 4.1 Outils, langage et méthodes d'implémentations.

#### 4.1.1 Outils d'implémentation.

Le développement d'un tel système nécessite l'utilisation de quelques outils. Dans ce qui suit nous citons les outils qui ont été utilisés.

##### **SQL Server :**

Microsoft SQL Server est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur Mac OS via Docker, car il en existe une version en téléchargement sur le site de Microsoft.

##### **Visual Studio :**

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement

de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages.

#### **4.1.2 Langages de programmation utilisés.**

##### **Langage SQL :**

SQL (Structured Query Language) en français « Langage d'interrogation Structuré », est un langage d'interrogation de base de données très populaire. Il constitue aujourd'hui une norme implémentée par de nombreux SGBDs (Systèmes de Gestion de Bases de Données), comprenez : des serveurs de bases de données. L'accès aux BDD (bases de données) se fait de façon standard à l'aide de requêtes du langage SQL. Il existe un outil d'administration, MySQL Management, qui nous offre une interface pour manipuler les tables. La connaissance de quelques requêtes permet de répondre à la majorité des besoins de programmation.

##### **C# :**

C# (prononcé voir dièse, comme la note de musique , mais écrit avec le signe du nombre) est un langage de programmation multi-paradigme à usage général, englobant des disciplines de programmation à typage fort, à portée lexicale, impérative, déclarative, fonctionnelle, générique, orientée objet (basée sur la classe) et orientée composant. Il a été développé vers 2000 par Microsoft dans le cadre de son initiative .NET et a ensuite été approuvé comme norme internationale par Ecma (ECMA-334) en 2002 et ISO (ISO/IEC 23270) en 2003. Mono est le nom du projet libre et open-source visant à développer un compilateur et un runtime pour le langage. C# est l'un des langages de programmation conçus pour l'infrastructure linguistique commune (CLI).

##### **Flutter :**

Flutter est un kit de développement de logiciels d'interface utilisateur à source ouverte créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code. La première version de Flutter était connue sous le nom de code "Sky" et fonctionnait sur le système d'exploitation Android. Elle a été dévoilée lors du sommet des développeurs Dart de 2015, avec l'intention déclarée de pouvoir effectuer un rendu cohérent à 120 images par seconde. Lors du discours d'ouverture des Google Developer Days à Shanghai, Google a annoncé la sortie de Flutter Release Preview 2 qui est la dernière grande version avant Flutter 1.0. Le 4 décembre 2018, Flutter 1.0 a été publié lors de l'événement Flutter Live, ce qui représente la première version

"stable" du Framework. Le 11 décembre 2019, Flutter 1.12 a été publié lors de l'événement Flutter Interactive.

## 4.2 Présentation de la solution réalisée

### 4.2.1 Architecture MVC de l'application.

Le patron de conception modèle-vue-contrôleur (en abrégé MVC, en anglais model-view-controller), tout comme les patrons modèle-vue-présentation ou présentation, abstraction, contrôle, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Ce paradigme regroupe les fonctions nécessaires en trois catégories :

1. **Un modèle :** Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation. Le modèle ne contient aucun lien direct vers le contrôleur ou la vue. Sa communication avec la vue s'effectue au travers du patron observateur. Le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir tous les cours d'une salle, tous les cours d'une personne ou tous les cours d'un groupe de TD.
2. **Une vue :** Ce avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, cochage d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Plusieurs vues peuvent afficher des informations partielles ou non d'un même modèle. Par exemple si une application de conversion de base a un entier comme unique donnée, ce

même entier peut être affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir à l'utilisateur la possibilité de changer de vue.

3. **Un Controller** : Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est lui « observable ». Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

Par exemple, dans le cas d'une base de données gérant les emplois du temps des professeurs d'une école, une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne. Quand un même objet contrôleur reçoit les événements de tous les composants, il lui faut déterminer l'origine de chaque événement. Ce tri des événements peut s'avérer fastidieux et peut conduire à un code peu élégant (un énorme switch). C'est pourquoi le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants.

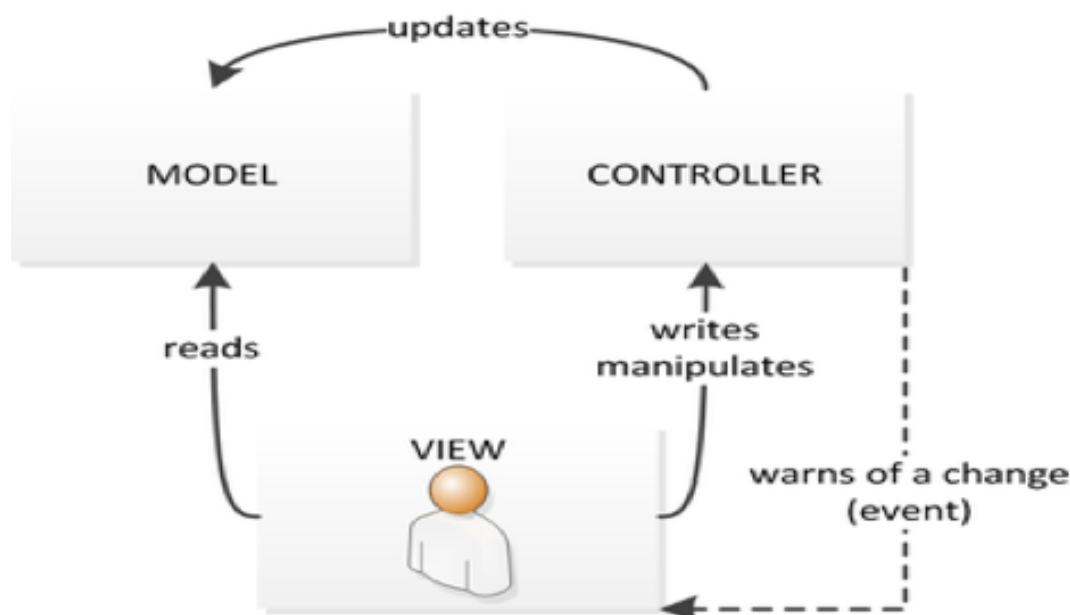


FIGURE 4.1 – Architecture MVC de l'application.

#### 4.2.2 Avantage du MVC.

Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur lors de la maintenance ou d'une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées. Le MVC montre ses limites dans le cadre des applications utilisant les technologies du web, bâties à partir de serveurs d'applications. Des couches supplémentaires sont alors introduites ainsi que les mécanismes d'inversion de contrôle et d'injection de dépendance.

#### 4.2.3 Page du tableau de bord.

Le tableau de bord présente de manière globale toutes les informations de l'application, sous forme de graphes et chiffres statistique. On présente le nombre d'informations enregistrées, ensuite l'alerte, qui est l'avancement automatique, on présente les commandes.

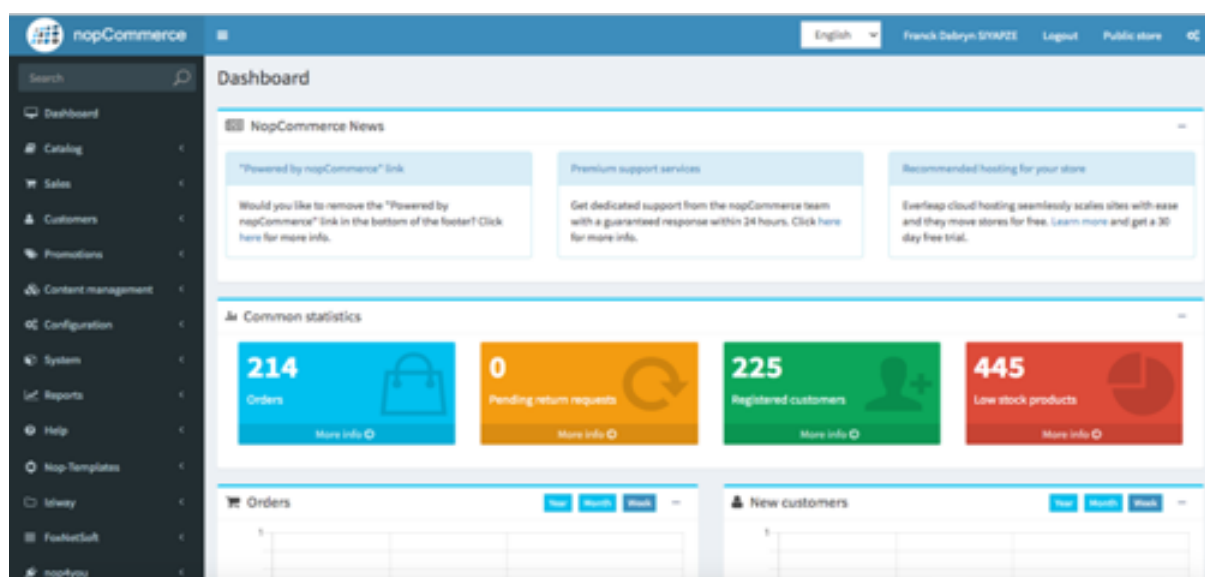


FIGURE 4.2 – Page du tableau de bord.

#### 4.2.4 Page de gestion des sliders.

Dans cette page on présente les informations concernant les sliders. Le premier bloc représente l'ajout des sliders et le deuxième bloc la liste des sliders déjà implémentés.

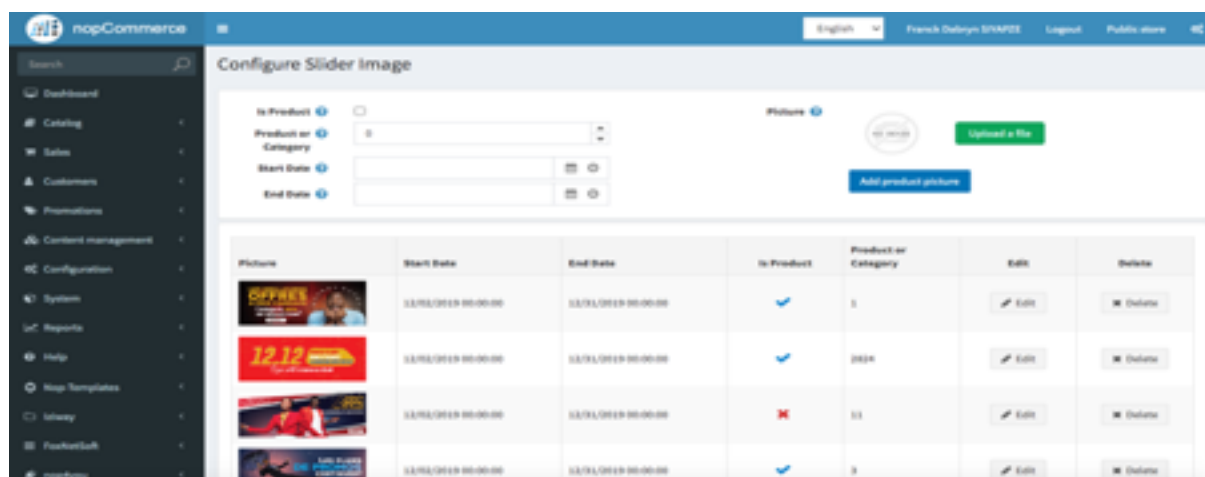


FIGURE 4.3 – Page de gestion des sliders.

Lorsque nous créons les sliders, l'utilisateur final peut voir ceci :



FIGURE 4.4 – Rendu des sliders sur l'application mobile.

#### 4.2.5 Page de gestion des push notifications.

Dans cette page il s'agit de montrer comment on crée les push notifications, on modifie et on le supprime.



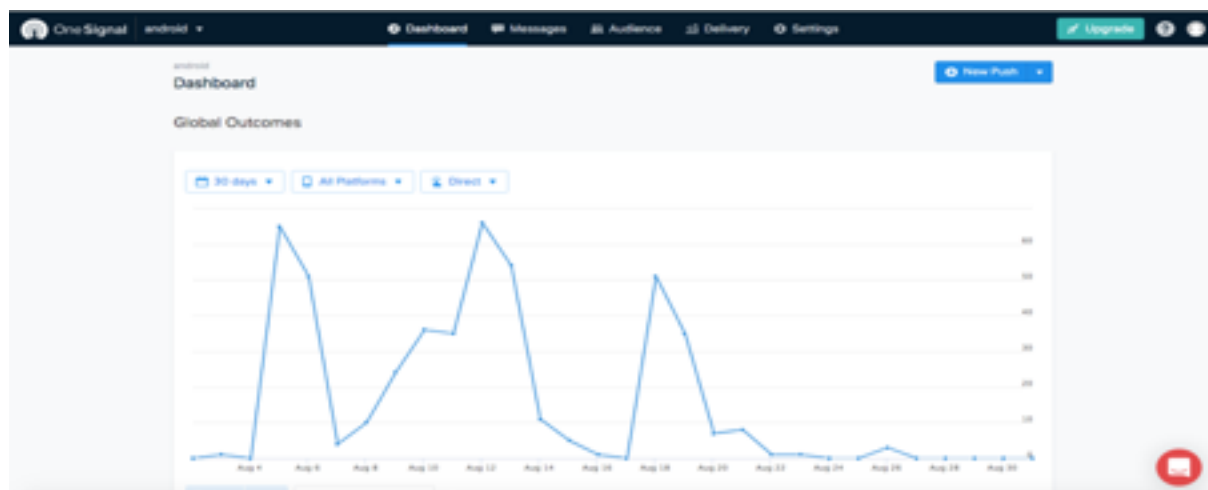


FIGURE 4.5 – Tableau de bord onesignal.

La figure ci-dessus représente la liste des push notifications envoyées.

Message	Status	Delivered	Open Rate	Action
Eduquez vos enfants avec les tablettes éducatives pour enfant pas chères de bonne qualité disponible sur iziway.cm	Sent	1.4k	1.52 %	Options X
Disque dur externe Toshiba 1000 Go neuf avec garantie disponible sur iziway.cm	Sent	1.4k	1.66 %	Options X
Achat du disque dur externe Toshiba 1000 Go neuf avec garantie disponible sur iziway.cm	Sent	1.4k	2.04 %	Options X
Ne battez plus vos pâtes avec la fourchette, voici le batteur sur socle - 2L	Sent	1.4k	2.35 %	Options X
GAGNEZ une ampoule LED le XIAOMI Redmi 8 de couleur rouge 32Go/ 3Go RAM 64 l'achat d'un téléphone XIAOMI REDMI 8	Sent	1.3k	1.58 %	Options X

FIGURE 4.6 – Liste des push notifications.

### Création d'un push notification

La création d'un nouveau push notification est assez simple, il suffit pour cela de cliquer sur le bouton « New Push » du menu droit, ensuite la page de création s'affiche comme l'image qui suit.

The screenshot shows the 'New Message' interface in the IZIWAY dashboard. The top navigation bar includes 'Dashboard', 'Messages' (active), 'Audience', 'Delivery', and 'Settings'. Below this, a sub-navigation bar lists 'Messages', 'In-App', 'A/B Tests', 'Templates', and 'Automated'. The main content area is titled 'New Message' and is divided into several sections:

- Audience:** Options to 'Send to Subscribed Users' (selected) or 'Send to Particular Segment(s)'.
- Message:** A form for creating the message. It includes:
  - ADD LANGUAGES:** A button to add more languages.
  - TITLE:** A text input field with the placeholder 'Title'.
  - MESSAGE:** A text input field with the placeholder 'Message'.
  - IMAGE:** A section with a 'Select or input URL, to a large >' button and a 'SUBMIT' button. Below it is a 'LARGER URL' input field with the placeholder 'http://url.com'.
  - Preview:** A large smartphone mockup showing the notification. The time is 11:17, the date is THURSDAY, OCTOBER 24, and the notification text is 'android Title Message Body'.
  - Buttons:** 'Send via Firebase' and 'Send via Apple Push'.
- Platform Settings:** A section with 'Platform Settings' and 'Advanced Settings' links.
- Schedule:** A section with two main options:
  - DELIVERY:**
    - ☒ Begin sending immediately
    - ☐ Begin sending at a particular time
  - DELIVERY OPTIMIZATION:**
    - ☒ Send immediately
    - ☐ Intelligent delivery (Recommended)
    - ☐ Optimize by user time zone
- DELIVERY:** A section with a 'DELIVERY' button and a 'SEND' button. It includes a 'READ OUR CONSUMER POLICY' link and a 'WHERE WILL IT SEND' link.

At the bottom, there are 'CONFIRM' and 'SAVE AS DRAFT' buttons.

FIGURE 4.7 – Création d'un push notification.

Après l'envoi du push, on peut avoir ces statistiques liés

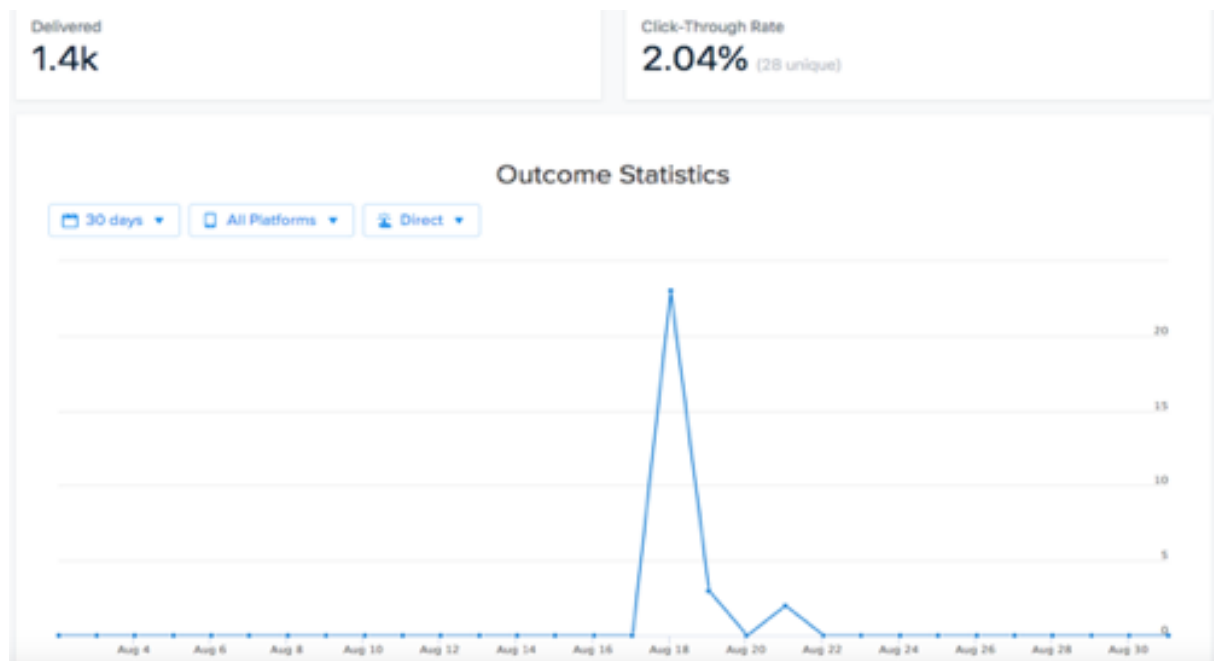


FIGURE 4.8 – Statistiques d'envoi d'un message.

Lorsqu'on envoie un push notification voilà comment il se présente sur l'application



FIGURE 4.9 – Exemple d'un push notification reçu.

#### 4.2.6 Page de gestion des bloc catégorie et bloc produit .

Dans cette page on renseigne les blocs catégories que nous voulons mettre en avant sur notre application et une fois les catégories mise en place nous allons pouvoir mettre en place les blocs produits correspondant aux différentes catégories.

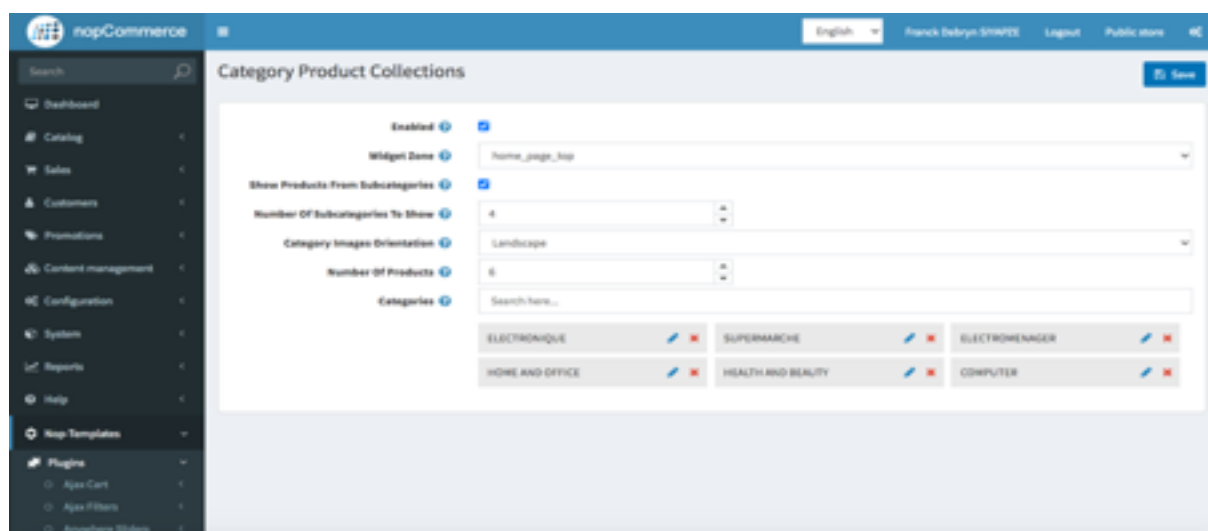


FIGURE 4.10 – Tableau de bord de gestion de bloc categorie.

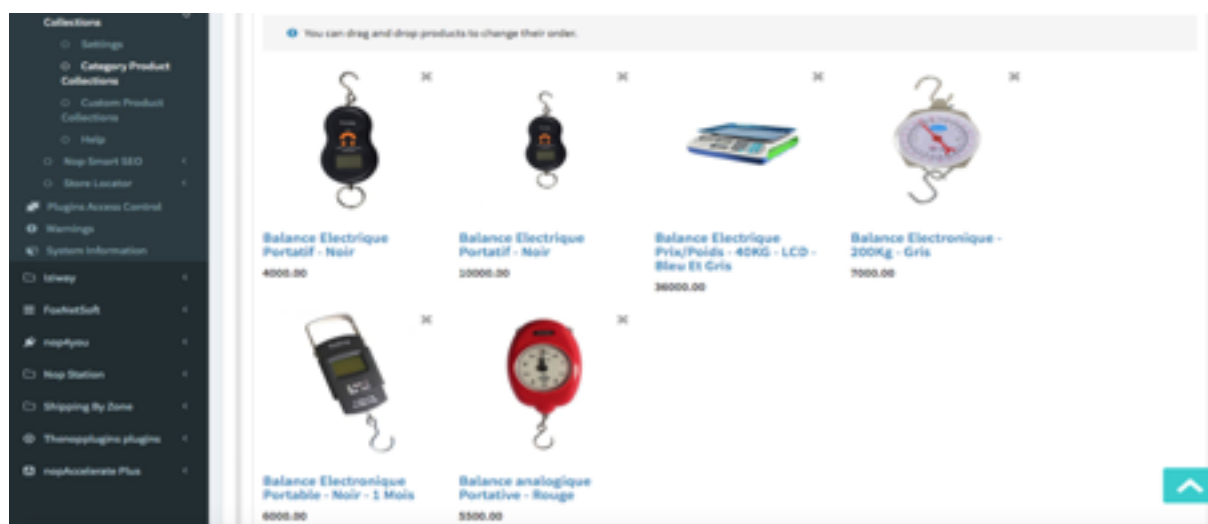


FIGURE 4.11 – Liste des produits d'une catégorie (Bloc Produit).

FIGURE 4.12 – Formulaire d'ajout des produits dans un bloc produit.

#### 4.2.7 Coût de mise en oeuvre de la solution .

TABLE 4.1 – Etude financière du matériel.

Nom du produit	Caractéristiques	Total
Serveur	Microsoft Azure offre Paas (Platform as Service) 3,20GHz, Disque Dur 250Go SSD, 3,5Go à 7Go de RAM	500 000 FCFA
Ordinateur	MacBook Pro Intel Core i7 Processor (2.2Ghz) 16Go / 255Go Disque SSD, MacOS High Sierra	250 000 FCFA
Google play console	Play Console	14 000 FCFA
<b>TOTAL</b>		<b>764 000 FCFA</b>

Les utilisateurs de l'application devront être formés.

TABLE 4.2 – Etude financière de la formation.

Prix de l'heure	Nombres d'heures par utilisateur	Nombre d'utilisateurs	Montant
0	10	4	0 FCFA

TABLE 4.3 – Total coût de mise en oeuvre.

	Prix
Matériels	764 000 FCFA
Formation	0 FCFA
Coût de réalisation	8 320 000 FCFA
<b>TOTAL</b>	<b>9 084 000 FCFA</b>

## **4.3 Discussion de la solution et bilan.**

La discussion de la solution est un point primordial, car elle permet de mesurer l'écart entre les prévisions et la réalisation. La mesure des écarts entre prévisions et réalisations, permettra d'apporter les correctifs nécessaires.

Ainsi la structure de l'application a été évaluée en fonction des objectifs et exigences du cahier des charges, pour déterminer si le produit fini est conforme aux besoins exprimés. Cette évaluation a porté sur les aspects techniques et organisationnels du projet. Une autre sera organisée après le lancement officiel de l'application et sera axée sur son usage réel.

### **4.3.1 Evalutation technique .**

Dans un premier temps nous avons testé toutes les pages et liens pour s'assurer de leur bon fonctionnement. Étant donné qu'il y a un grand nombre de téléphones et que des problèmes de compatibilité peuvent se poser nous avons testé l'application sur plusieurs téléphones, ensuite nous avons testé sa vitesse de téléchargement avant de faire des tests utilisateurs.

#### **Tests sur téléphones**

Nous avons déployé l'application iziway sur Google Play console. Le résultat s'est avéré très concluant, car nous n'avons pas remarqué une incompatibilité particulière avec les téléphones de différentes tailles.

#### **Tests utilisateurs**

Le test utilisateur est une étape indispensable pour savoir si l'application a répondu aux besoins et attentes des utilisateurs. La facilité d'utilisation d'une application est un critère de satisfaction des visiteurs, aussi cette expérience est importante pour pouvoir rectifier les erreurs.

Ainsi nous avons testé avec comme utilisateurs le responsable marketing, le CEO de iziway, le responsable informatique et le responsable content-writer. Ces derniers ont été mis dans des conditions réelles permettant de réaliser de véritables opérations durant les tests. Nous avons fait en sorte qu'ils se sentent en confiance et que ce n'est pas eux qui sont testé, mais l'application.

De ce fait nous avons été neutres, sans jamais intervenir durant leurs passages sur l'application. Ces tests ont été assez positifs du fait de la simplicité de l'application tant du point de vue de la charte graphique que de l'arborescence. Certes il a fallu expliquer d'une fois comment les choses fonctionnent, car la première visite est toujours difficile du fait de la nouveauté à laquelle on fait face. Il ressort de ces tests qu'il n'est pas difficile de se familiariser avec l'application.

### 4.3.2 Respect du cahier des charges.

Cette étape a nécessité la vérification du cahier des charges pour savoir si les différents points de ce dernier ont été bien réalisés. Aussi nous allons établir d'abord la méthodologie, ensuite nous allons procéder à l'évaluation proprement dite avant de faire une conclusion sur le résultats obtenus.

#### Méthodologie

TABLE 4.4 – Méthodologie.

Type d'indicateurs	Objet de vérification	Méthodologie
Conformité	Le contenu et les fonctionnalités sont conformes aux points définis dans le cahier des charges	Comparer le contenu et les fonctionnalités de l'application par rapport aux charges
Efficience	Respect du temps de réalisation demandé	Vérifier que tout a été réalisé dans le délai imparti
Performance	Temps de téléchargement, comptabilité avec les navigateurs	Vérifier erreurs techniques et mauvais paramétrages

#### Evaluation

TABLE 4.5 – Evaluation.

Prévisions / cahier des charges	Réalisation	Résultats
Gestion des blocs produits	Réalisé	OK
Gestion des sliders	Réalisé	OK
Gestion des push notifications	Réalisé	OK
Gestion des doubles bannières	Réalisé	OK
Gestion des pop-ups	Réalisé	OK

#### Évaluation de l'usage en temps réel de l'application

Cette évaluation de l'usage réel du portail est extrêmement importante, et sera effectué après le chargement des documents dans l'application et sa mise en service officielle. La question de l'usage revêt une importance particulière. Il est donc nécessaire d'évaluer l'impact des différentes mesures prises en la matière et l'intérêt du contenu de l'application au regard de l'utilisation réelle de celui-ci.



Il s'agit en fait d'analyser ce que l'on appelle l'appropriation de la technologie par les utilisateurs, c'est-à-dire la manière dont ceux-ci adoptent la technologie. Par rapport à l'application, cela implique notamment d'évaluer quelles parties du site sont plus ou moins visitées par les utilisateurs, quels sont les fonctionnalités et services les plus utilisés.

En définitive il s'agit de déterminer si l'application a rempli les objectifs de dépôts. En d'autres termes qu'est que le portail a apporté réellement à IZIWAY. Et tout ça en prenant en compte les commentaires spontanés fournis par les utilisateurs de l'application.

Nous pouvons donc ressortir les différentes clés de performances sur l'application

### **Concernant les nombres d'utilisateurs :**

Étant donné que l'application été déployé en 20 décembre 2019 nos statistiques vont aller du 20 décembre 2019 au 31 août 2020 (date de fin de mon stage).



FIGURE 4.13 – Nombre d'utilisateurs avant le début du stage.

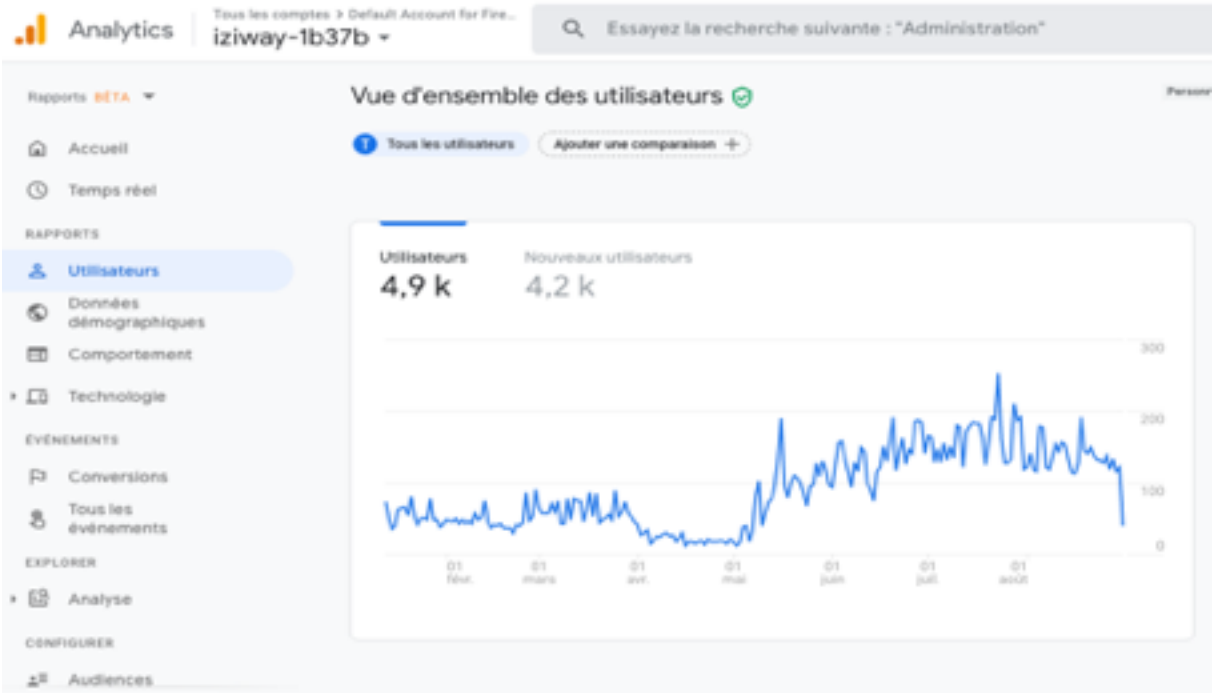


FIGURE 4.14 – Nombre d’utilisateurs après mon arrivé.

**Concernant les nombres de conversion :**

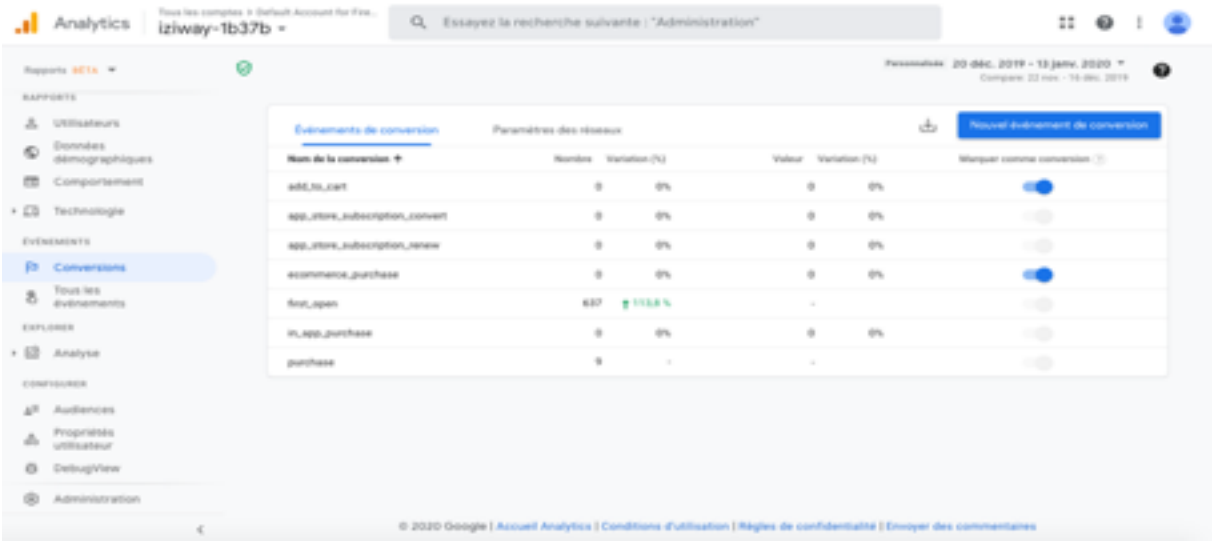


FIGURE 4.15 – Nombre de conversion avant mon arrivé.

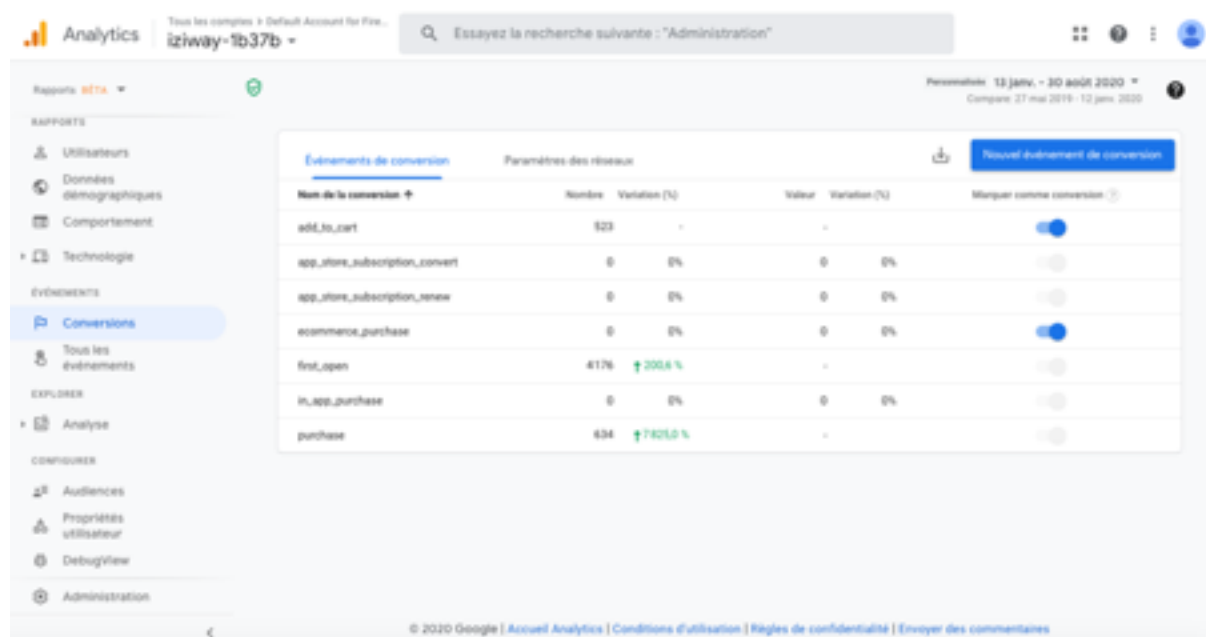


FIGURE 4.16 – Nombre de conversion après mon arrivé.

### 4.3.3 Bilan.

#### Technique

Après 8 mois de travail, nous avons finalement terminé la plus grande partie du projet, et ne reste que le paramétrage. Ayant comme base de travail la mission assignée et le cahier des charges, nous avons optimiser l'application Android de la Marketplace IZIWAY CAMEROUN. Cette application étant un cadre de référence souple, est une porte d'entrée vers des contenus, des outils, qui permettent aux utilisateurs d'acheter et consulter.

De ce fait, il est amené à être évolutif donc en perpétuelle amélioration afin de mieux s'adapter aux besoins des utilisateurs. Ainsi une évaluation permanente sera effectuée pour mesurer les insuffisances de l'application et les nouveaux besoins. Dans cet ordre d'idées, d'autres services plus pointus pourraient être développés et ajoutés à l'application.

#### Humain

Ce projet nous a permis de découvrir le cadre de travail, l'organisation, le personnel, bref la vie d'une entreprise. D'abord il y'a un cadre convivial avec des relations très fraternelles, et chaleureuses. Simplicité dans les relations car on ne sent pas les rapports hiérarchiques. Ensuite un esprit de groupe très développé, pour ne pas dire de travail d'équipe, on collabore et s'entraide pour atteindre les objectifs, même s'il y a des fois une certaine tendance au laxisme.

Avec des moyens modestes IZIWAY fait un travail extrêmement important pour la promo-

tion et le développement du pays. Aussi nous avons été honoré de pouvoir apporter une modeste contribution par la réalisation de cette application qui, nous espérons, leur donnera des opportunités pour être soutenu dans leur action.

## CONCLUSION

Le travail effectué dans ce mémoire a eu pour objectif l'optimisation de l'application de la Marketplace IZIWAY CAMEROUN, dans le but d'améliorer sa qualité de service . C'est ainsi que nous avons analysé la problématique et nous sommes arrivé à optimiser l'application qui est une solution efficace et bénéfique. Pour cela on a mené en premier lieu une présentation des concepts généraux et du contexte. Ensuite, nous avons entamé le second chapitre dans lequel nous avons étudié l'existant et présenté un cahier des charges de la solution. Dans le troisième chapitre, nous avons décrit le langage de modélisation pour la conception de notre application qui est le langage UML. On a également recensé les acteurs qui interagissent avec l'application, puis, on a décrit les besoins de chaque acteur sous forme de cas d'utilisation. Et aussi, pour chaque cas d'utilisation, on a établi le diagramme de séquence dont l'objectif est de représenter les interactions entre les objets du système en indiquant la chronologie des échanges. Après, la réalisation d'une modèle statique représenté par le diagramme de classe. Enfin, on a pris le temps de réaliser à bien notre application tout en spécifiant les outils de développement ainsi que les langages de programmation utilisés, suivi d'un aperçu des interfaces que comprend celle-ci. Ce travail nous a permis d'acquérir une expérience personnelle et professionnelle. Il nous a été très bénéfique car on a eu la chance d'améliorer nos connaissances dans le domaine de la conception et cela sur le plan théorique, mais aussi de découvrir et d'acquérir de nouvelles connaissances en matière de programmation et de développement des bases de données en ce qui concerne l'aspect pratique. Nous souhaitons que ce travail puisse servir comme un outil d'aide et de documentation pour les étudiants à l'avenir, et une base de travail pour les utilisateurs concernés.

## BIBLIOGRAPHIE

- [1] AFNOR, *"Fascicule de documentation ". Management de processus ( référence FD X 50-116).*
- [2] HERNIAUX Gérard, NOYE Didier , *"Organiser et améliorer les processus". Introduction à la méthode EFPRO, INSEP Editions ; (1996).*
- [3] GRUA Hervé, SEGONAC J.Michel ; DUNOD, *"Maîtriser les processus et piloter le progrès". (1999).*
- [4] NOYE Didier *"L'amélioration participative des processus". INSEP Editions, 1997 (1997).*
- [5] *url*[https : //prisync.com/blog/e-commerce-business-models-choosing-the-right-one-for-b2b-marketplace/](https://prisync.com/blog/e-commerce-business-models-choosing-the-right-one-for-b2b-marketplace/) 04/09/2020, 04h40
- [6] Henri Issac and Pierre Volle, *"E-commerce de la stratégie à la mise en œuvre opérationnelle". Pearson Education France.*
- [7] Philippe Carrot and Adrien Nussenbaum, *"Marketplace l'E-commerce de Demain". avec la collaboration de Clément Vouillon, (2012).*
- [8] *https : //prisync.com/blog/e-commerce-business-models-choosing-the-right-one-for-b2b-marketplace/ .* 04/08/2020, 06h49.
- [9] Meddour Mohamed Lamine & Barchen Mahdi *"L'application de la méthode de "Pareto" pour l'identification des factures critiques de la qualité, cas du complexe de kenchela". Université Larbi Ben M'Hidi -Oum El-Bouagh (2017/2018).*
- [10] *https : //prisync.com/blog/e-commerce/ .* 08/08/2020, 04h40.
- [11] *https : //prisync.com/blog/articles/single/ .* 08/08/2020, 06h49.

[12] Martin Fowler "*UML 2.0*". *Campus Press*, (23/02/2004).

# ANNEXE A

## TABLE DES MATIÈRES.

<b>Dédicace</b>	<b>i</b>
<b>Remerciements</b>	<b>ii</b>
<b>Sommaire</b>	<b>iv</b>
<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>Liste des sigles et abréviations</b>	<b>vii</b>
<b>Liste des tableaux</b>	<b>viii</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Introduction</b>	<b>1</b>
<b>Chapitre 1 NOTION DE BASE SUR L’OPTIMISATION.</b>	<b>2</b>
1.1 En préalable à l’optimisation des processus . . . . .	2
1.1.1 Qu’est-ce qu’un processus. . . . .	2
1.1.2 Pourquoi les optimiser ?[2] . . . . .	2
1.1.3 A quel moment faut-il le faire ? . . . . .	3
1.2 Les principes de base de l’optimisation des processus . . . . .	3
1.2.1 Bien caractériser le périmètre couvert par le processus . . . . .	3
1.2.2 Identifier les interfaces . . . . .	4
1.2.3 Ne travailler que des processus-clés ou processus critiques . . . . .	4



1.2.4	Privilégier une approche participative . . . . .	4
1.2.5	Garder de la souplesse dans la formalisation pour rester ouvert à l'urgence . . . . .	5
1.3	Une démarche possible pour optimiser un processus[4] . . . . .	5
1.3.1	Cartographier les processus . . . . .	5
1.3.2	Choisir les processus-clés . . . . .	6
1.3.3	Caractériser un processus . . . . .	6
1.3.4	Décrire un processus . . . . .	7
1.3.5	Diagnostiquer un processus et son contexte pour définir les objectifs d'optimisation . . . . .	8
1.3.6	Choisir le degré d'optimisation : améliorer ou reconcevoir selon des objectifs et indicateurs de performance . . . . .	8
1.3.7	Optimiser le processus . . . . .	9
1.3.8	Mettre en œuvre et piloter le nouveau processus . . . . .	10
<b>Chapitre 2 ETUDE DE L'EXISTANT.</b>		<b>11</b>
2.1	État de l'art d'une Marketplace . . . . .	11
2.1.1	Qu'est-ce qu'une Marketplace. . . . .	11
2.1.2	Avantages et inconvénients d'une Marketplace . . . . .	14
2.2	Description de l'existant. . . . .	15
2.2.1	Type de marketplace . . . . .	15
2.2.2	Présentation de l'existant . . . . .	16
2.2.3	Critique de l'existant. . . . .	17
2.3	Cahier de charge . . . . .	18
2.3.1	Définition du projet. . . . .	18
2.3.2	Présentation des concurrents . . . . .	18
2.3.3	Caractéristiques de la solution voulue : . . . . .	21
2.3.4	Spécification des besoins : . . . . .	21
2.3.5	Solutions envisageables. . . . .	23
2.3.6	Coûts de réalisation par la méthode COCOMO. . . . .	24
2.3.7	Planification . . . . .	27
2.3.8	Contraintes du projet . . . . .	28
2.4	Méthodologie de conduite de projet. . . . .	28

2.4.1	Modèle de cycle de vie en cascade[10] . . . . .	28
2.4.2	Modèle de cycle de vie en V[10] . . . . .	29
2.4.3	Modèle de cycle de vie en spirale [10] . . . . .	30
2.4.4	La méthodologie agile [11] . . . . .	31
2.4.5	Pourquoi la méthodologie agile . . . . .	32
<b>Chapitre 3 Analyse et Conception.</b>		<b>34</b>
3.1	Présentation de la méthodologie. . . . .	34
3.1.1	Méthodologie adoptée Scrum . . . . .	34
3.2	Modèle. . . . .	37
3.2.1	Choix du langage de modélisation.[12] . . . . .	38
3.3	Concept d'UML . . . . .	39
3.3.1	Définition . . . . .	39
3.3.2	Unité . . . . .	39
3.3.3	Présentation de quelques diagrammes UML . . . . .	39
3.4	Analyse et Modélisation . . . . .	42
3.4.1	Les acteurs système . . . . .	42
3.4.2	Les cas d'utilisation . . . . .	42
3.4.3	Modélisations Conceptuelles . . . . .	46
3.5	Architecture générale de l'application . . . . .	52
3.5.1	Définition . . . . .	52
3.5.2	Architecture de notre système . . . . .	54
<b>Chapitre 4 Résultats et Discussion</b>		<b>56</b>
4.1	Outils, langage et méthodes d'implémentations. . . . .	56
4.1.1	Outils d'implémentation. . . . .	56
4.1.2	Langages de programmation utilisés. . . . .	57
4.2	Présentation de la solution réalisée . . . . .	58
4.2.1	Architecture MVC de l'application. . . . .	58
4.2.2	Avantage du MVC. . . . .	60
4.2.3	Page du tableau de bord. . . . .	60
4.2.4	Page de gestion des sliders. . . . .	61
4.2.5	Page de gestion des push notifications. . . . .	62

4.2.6	Page de gestion des bloc catégorie et bloc produit . . . . .	66
4.2.7	Coût de mise en oeuvre de la solution . . . . .	68
4.3	Discussion de la solution et bilan. . . . .	69
4.3.1	Evalutation technique . . . . .	69
4.3.2	Respect du cahier des charges. . . . .	70
4.3.3	Bilan. . . . .	73
<b>Conclusion</b>		<b>75</b>
<b>Bibliographie</b>		<b>1</b>
<b>Chapitre A Table des matières.</b>		<b>AA</b>
<b>Table des matières</b>		<b>AA</b>