

REPUBLIQUE TOGOLAISE

Travail – Liberté – Patrie

MINISTERE DE LA PLANIFICATION
ET DU DEVELOPPEMENT



Institut Africain d'Informatique
Représentation du TOGO
(IAI-TOGO)

07 BP 12456 Lomé 07

Tel : (+228) 22 20 47 00

E-mail : iaitogo@gmail.com

Site-Web : www.iai-togo.com

**Ingénieurs
& Experts**

INGENIEURS & EXPERTS.

Bureau de conseils-Etude et contrôle en Ingénierie.

Tel: (+228) 98 93 89 18 / 91 54 85 49

/ 97 37 53 80

E-mail: groupe_ie@hotmail.com

RAPPORT DE STAGE PRATIQUE EN ENTREPRISE

Type de stage : Génie Logiciel

**Application web de gestion de stocks de marchandises : cas de
IE Boutique**

Période : Du 06 Juin au 05 Août 2022

Rédigé par :

HONYIGLO Wilfried

Etudiant en deuxième année Tronc Commun

Année Universitaire : 2021 – 2022

Maître de stage :

M. KOSSI-TITRIKOU E.M-O

Responsable adjoint du département informatique

Superviseur :

M. AZOTI Hodabalo

Enseignant à IAI-Togo

REMERCIEMENTS

A DIEU TOUT PUISSANT

Qui m'a donné la santé et la force nécessaire pour mener à bien cette année scolaire.

A ma mère

Sans qui je ne serai pas ce que je suis aujourd'hui et qui n'a jamais cessé de faire son maximum pour m'offrir tout ce dont j'ai besoin et pour toutes les forces qu'elle m'a donné et me donne tout au long de mon parcours.

A MON PERE

Qui a toujours été là pour moi et qui m'a encouragé dans les moments difficiles.

A MA TANTE

Sans qui je n'aurai jamais eu la possibilité de poursuivre mes études universitaires.

A tout le soutien financier, moral qu'elle m'a accordé tout au long de mon parcours.

SOMMAIRE

REMERCIEMENTS	I
A DIEU TOUT PUISSANT.....	ERREUR ! SIGNET NON DEFINI.
A MA MERE	ERREUR ! SIGNET NON DEFINI.
A MON PERE	ERREUR ! SIGNET NON DEFINI.
A MA TANTE.....	ERREUR ! SIGNET NON DEFINI.
AVANT-PROPOS	ERREUR ! SIGNET NON DEFINI.
RESUME	ERREUR ! SIGNET NON DEFINI.
LISTE DES PARTICIPANTS AU PROJET	ERREUR ! SIGNET NON DEFINI.
SOMMAIRE	II
LISTE DES FIGURES	III
LISTE DES TABLEAUX	III
INTRODUCTION.....	1
PARTIE 1 : CAHIER DES CHARGES.....	ERREUR ! SIGNET NON DEFINI.
1 CAHIER DES CHARGES	3
1.1 PRESENTATION DU SUJET.....	3
1.2 PROBLEMATIQUE DU SUJET	3
1.3 UTILISATEURS DU SYSTEME	3
1.4 INTERET DU SUJET	3
1.5 LES CIBLES	4
1.6 TYPE D'APPLICATION	4
1.7 EQUIPEMENTS DES CIBLES.....	4
PARTIE 2 : PRE-PROGRAMMATION	ERREUR ! SIGNET NON DEFINI.
2 PREPROGRAMMATION.....	6
2.1 ETUDE DE L'EXISTANT	6
2.2 CRITIQUE DE L'EXISTANT	6
2.3 PLANNING PREVISIONNEL DE REALISATION	7
2.4 ETUDE DETAILLEE DE LA SOLUTION	7
PARTIE 3 : REALISATION ET MISE EN ŒUVRE.....	ERREUR ! SIGNET NON DEFINI.
3 REALISATION ET MISE EN ŒUVRE.....	21
3.1 MATERIELS ET LOGICIELS UTILISES.	21
3.2 SECURITE DE L'APPLICATION	26
3.3 EVALUATION FINANCIERE DE LA SOLUTION	26
3.4 PRESENTATION DE L'APPLICATION	26
CONCLUSION.....	37
BIBLIOGRAPHIE INDICATIVE	VI
TABLES DES MATIERES	VI

LISTE DES FIGURES

FIGURE 1 : DIAGRAMME DE CAS D'UTILISATION ADMINISTRATEUR	8
FIGURE 2 : DIAGRAMME DE CAS D'UTILISATION AFFILIE	9
FIGURE 3 : DIAGRAMME DE CAS D'UTILISATION DU SYSTEME	10
FIGURE 4 : DIAGRAMME D'ACTIVITE DU CAS D'UTILISATION S'AUTHTENTIFIER	13
FIGURE 5 : DIAGRAMME D'ACTIVITE DU CAS D'UTILISATION S'ABONNER	14
FIGURE 6 : DIAGRAMME D'ACTIVITE DU CAS D'UTILISATION EFFECTUER UNE VENTE	15
FIGURE 7 : DIAGRAMME DE SEQUENCE S'AUTHTENTIFIER	16
FIGURE 8 : DIAGRAMME DE SEQUENCE S'ABONNER	17
FIGURE 9 : DIAGRAMME DE SEQUENCE DU CAS EFFECTUER UNE VENTE	18
FIGURE 10 : DIAGRAMME DE CLASSE DU SYSTEME	19
FIGURE 11: LOGO DE PHP	22
FIGURE 12: LOGO DE JAVASCRIPT	22
FIGURE 13 : LOGO DE LARAGON	23
FIGURE 14 : LOGO DE VISUAL STUDIO CODE	23
FIGURE 15 : LOGO DE LARAVEL	24
FIGURE 16 : LOGO DE GIT	24
FIGURE 17: LOGO DE GITHUB	25
FIGURE 18 : LOGO DE BOOTSTRAP	25
FIGURE 19 : LOGO DE JQUERY	26
FIGURE 20 : PLAN DE NAVIGATION ADMINISTRATEUR	32
FIGURE 21 : PLAN DE NAVIGATION AFFILIE	33
FIGURE 22 : PLAN DE NAVIGATION GLOBAL	34
FIGURE 23 : DETAILS D'UN ARTICLE	35
FIGURE 24 : LISTE DES VENTES.....	35
FIGURE 25 : LISTE DES ARTICLES.....	36

LISTE DES TABLEAUX

TABEAU 1 : PLANNING PREVISIONNEL.	7
TABEAU 2 : CAPTURE DES BESOINS FONCTIONNELS LES DIFFERENTS CAS D'UTILISATION.	8
TABEAU 3 : DESCRIPTION DU CAS S'AUTHTENTIFIER	10
TABEAU 4 : DESCRIPTION DU CAS EFFECTUER UNE VENTE	11
TABEAU 5 : DESCRIPTION DU CAS S'ABONNER	12
TABEAU 6 : INFORMATION SUR LE MATERIEL UTILISE.	21

INTRODUCTION

L'informatique apparaît de nos jours comme un élément indispensable dans le processus de développement des entreprises, son utilisation apporte un gain de temps considérable en matière de traitement de tâches, garantissant en plus la sécurité, la fiabilité, la disponibilité ainsi que la réutilisabilité des données déjà enregistrées. Compte tenu des nombreux avantages qu'il offre, il se trouve de nos jours au carrefour de tous les domaines. Ainsi de nombreux entreprises et écoles de renommées optent pour son utilisation. Par conséquent dans le cadre du cycle d'Ingénieurs des travaux Informatiques, l'institut Africain d'Informatique du TOGO (IAI-TOGO) a mis en place, pour les étudiants de deuxième (2^{ème}) année, un projet de stage pratique. Son objectif est la mise en pratique des connaissances en analyse et programmation acquises tout au long des deux (2) années d'études à l'institut et la réalisation d'un projet concret qui devra fournir une application viable et prête à l'emploi. A cet effet, nous avons eu un projet de deux mois. Notre mission consistait à créer une application web permettant la gestion de stock et d'activités d'une boutique.

PARTIE 1 :
CAHIER DES CHARGES

1 Cahier des charges

1.1 Présentation du sujet

La gestion de stock des produits est un facteur important dans l'évolution de toute entreprise spécialisée dans le domaine commercial. Une mauvaise gestion entraînerait une diminution du chiffre d'affaires des entreprises spécialisées dans ce domaine. La gestion de stock ne se limite pas en une demande d'approvisionnement quand une limite de produit est atteinte, elle va encore plus loin. C'est dans ce cadre que la société Ingénieurs et Experts veut mettre à la disposition de ces entreprises un système automatisé qui leur permettra de bien gérer leur stock, que ce soit pour les approvisionnements, les commandes, la traçabilité des différentes opérations qui ont eu lieu.

1.2 Problématique du sujet

Dans la plupart des entreprises de vente d'articles, la gestion de stock se fait manuellement. Ainsi des manquements peuvent être faites comme des erreurs de calculs ou bien d'autres, ou encore il peut arriver que des données soient perdues. Le suivi des emballages chez le client peut s'avérer parfois très contraignant. Ainsi, l'on assiste souvent à des pertes d'emballages et donc des pertes financières. Or dans le domaine commercial, il est primordial de dépenser moins pour gagner plus et ainsi minimiser ses dépenses, alors :

- Comment faciliter la traçabilité des opérations ?
- Comment établir rapidement les états tout en minimisant les erreurs ?
- Comment avoir une vue d'ensemble sur l'état des marchandises ?
- Comment mieux gérer les coûts des achats des marchandises ?

1.3 Utilisateurs du système

Le système disposera de deux utilisateurs :

- Les utilisateurs internes qui sont les administrateurs et les gérants.
- Les utilisateurs externes qui sont les clients ou encore des utilisateurs affiliés à une boutique ou à une entreprise.

1.4 Intérêt du sujet

1.4.1 Objectifs

Ils sont divisés en deux parties :

1.4.1.1 *Objectif générale*

Il consiste à améliorer la qualité des services, une meilleure gestion de la boutique.

1.4.1.2 *Objectifs spécifiques*

Le système doit permettre :

- Aux administrateurs de :
 - Ajouter et gérer les formules d'abonnement ;
 - Gérer le compte d'un abonné ;
 - Gérer les historiques des activités ;

- Gérer les demandes ou requêtes des affiliés.
- Aux affiliés de :
 - Enregistrer les informations sur sa boutique ;
 - Consulter l'état de la boutique et de chaque entrepôt ;
 - Effectuer une opération sur sa boutique ;
 - Retrouver une opération effectuée à une date donnée ainsi que ses détails ;
 - Contrôler son stock de marchandise ;
 - Faire un récapitulatif d'une date donnée.

1.4.2 Les Résultats

- Du côté de l'administrateur :
 - Les abonnements sont ajoutés et mieux gérés ;
 - Les comptes des abonnés sont gérés ;
 - Les historiques des activités sont établis et consultables.
- Du côté des clients ou des affiliés :
 - Les informations sur les boutiques sont enregistrées ;
 - L'état de la boutique et des entrepôts est facilement accessible et consultable ;
 - Diverses opérations sont effectuées sur la boutique ;
 - L'historique de chaque opération est enregistrée et est consultable par l'utilisateur y ayant droit ;
 - Les stocks de marchandise sont contrôlés ;
 - Les récapitulatifs des opérations suivant une date donnée sont réalisés.

1.5 Les cibles

Le système qui sera mise en place est destiné à des particuliers se trouvant dans le domaine commercial.

1.6 Type d'application

Le système qui sera mise en place est destiné à des particuliers se trouvant dans le domaine commercial.

1.7 Equipements des cibles

Afin de pouvoir utiliser le système, chaque utilisateur doit être en possession de certains éléments ou équipements, à savoir

- Un support électronique comme un téléphone ou un ordinateur, ou une tablette ;
- Une connexion internet.

PARTIE 2 :
PRE-PROGRAMMATION

2 Préprogrammation

2.1 Etude de l'existant

La boutique IE-Boutique ne dispose pas de programmes informatiques. Les opérations sont toutes faites de façon manuelle. Les documents souvent utilisés sont :

- Les cahiers d'inventaires
- Les cahiers de ventes
- Les documents de facturation et de vérification des produits en stock.

Description des modes de fonctionnement de la société.

- Les commandes se font généralement suite à une diminution du stock constatée par le magasinier. Selon les fonds disponibles dans les caisses de la boutique, la commande est passée sur le champ ou non.
- Les ventes se réalisent lorsque les articles sont disponibles et sont enregistrer dans un cahier de vente.
- La quantité de produit en stock et la facturation des différentes opérations sont sauvegardées dans un document.

En fin de journée, le vendeur ou le gérant comptabilise les quantités écoulées pour chaque article vendu et ainsi remplir les cahiers correspondants.

2.2 Critique de l'existant

Pour pouvoir assurer une bonne gestion et faire accroître les revenus de la boutique, il est primordial d'assurer la cohérence de la sauvegarde des informations comme la quantité en stock, le nombre d'articles vendus et bien d'autres. IE-Boutique présente des insuffisances dans ces domaines. Les insuffisances rencontrées sont en outre :

- L'impossibilité de faire un inventaire correct : les calculs qui interviennent dans les inventaires sont faits manuellement et sont donc exposés aux erreurs.
- L'impossibilité d'élaborer des statistiques : les informations sur les ventes, les dépenses, ... sont traitées manuellement ce qui rend impossible l'élaboration des statistiques sur une longue période.
- La perte de temps : La mise en place des états sur les dépenses, les ventes nécessitent un temps considérable et beaucoup d'attentions.
- L'insécurité des informations : En effet il est tout à fait possible que les documents où sont sauvegardés les informations liées aux dépenses, aux stocks soit perdus suite à un incident ou bien égarés suite à une négligence.
- L'absence de professionnalisme : Aujourd'hui, une structure qui fonctionne sans une gestion informatisée si l'on peut le dire mal préparer pour affronter la concurrence grandissante, et peut vite faire faillite.

2.3 Planning prévisionnel de réalisation

Le planning prévisionnel représente la manière dont le projet va s'organiser dans le temps et les différentes étapes à suivre pour atteindre les objectifs du projet.

Tableau 1 : Planning prévisionnel.

Prévisions			
Tâches	Date de début	Date de fin	Durée (En Jours)
Prise de contact avec le centre d'accueil	06 Juin 2022.	06 Juin 2022.	1
Prise de connaissance du thème	07 Juin 2022.	07 Juin 2022.	1
Compréhension du thème.	08 Juin 2022.	09 Juin 2022	2
Rédaction et validation du cahier des charges.	10 Juin 2022	13 Juin 2022	4
Apprentissage des outils, logiciels et langages de programmation.	14 Juin 2022	24 Juin 2022	10
Analyse, conception et modélisation.	25 Juin 2022	26 Juin 2022	2
Programmation.	26 juin 2022	26 juillet 2022	30
Tests fonctionnels et corrections	26 juillet 2022	01 Juillet 2022	7
Déploiement de la solution.	02 Août 2022	03 Août 2022	2
Dépôt du rapport de stage.	09 Août 2022	09 Août 2022	1
Préparation pour la validation.	10 Août 2022	11 Août 2022	3
Validation.	12 Août 2022	12 Août 2022	1

2.4 Etude détaillée de la solution

Dans cette partie, il sera question de représenter les différents diagrammes d'UML selon la structure préconisée par le 2TPU.

2.4.1 Etape 1 : Etude préliminaire.

Dans cette rebrique, nous allons aborder la signification d'un acteur et citer ceux qui interviennent dans notre système.

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie. Le diagramme de cas d'utilisation d'UML distingue deux acteurs à savoir : les acteurs principaux et les acteurs secondaires.

Dans notre cas de figure, l'on dispose de 2 acteurs principaux qui sont :

- L'affilié ou l'abonné à la plateforme ;
- L'administrateur.

2.4.2 Etape 2 : Capture des besoins fonctionnels

A. Le diagramme de cas d'utilisation.

Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisations reliés aux acteurs par des associations.

- Les différents cas d'utilisations.

Un cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système.

Tableau 2 : Capture des besoins fonctionnels les différents cas d'utilisation.

Acteurs	Cas d'utilisation
Administrateur	<ul style="list-style-type: none">➤ Créer un compte➤ S'authentifier➤ Gérer les comptes➤ Consulter toutes les activités
Affilié	<ul style="list-style-type: none">➤ Créer un compte➤ S'abonner➤ Ajouter un utilisateur➤ Effectuer une vente➤ Gérer son stock➤ Consulter les activités de sa boutique

- Représentation des différents cas d'utilisation par acteur

- Administrateur.

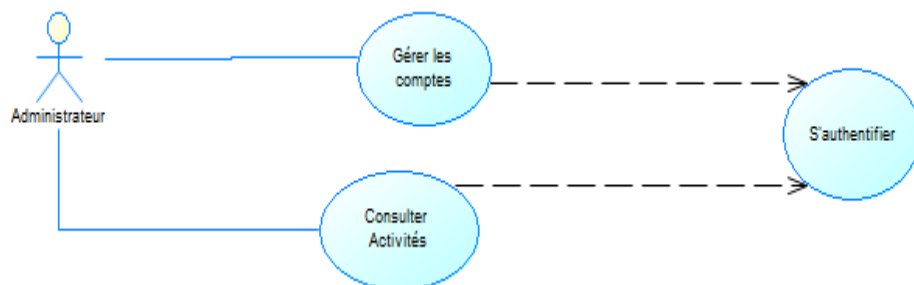


Figure 1 : Diagramme de cas d'utilisation **Administrateur**.

- Affilié.

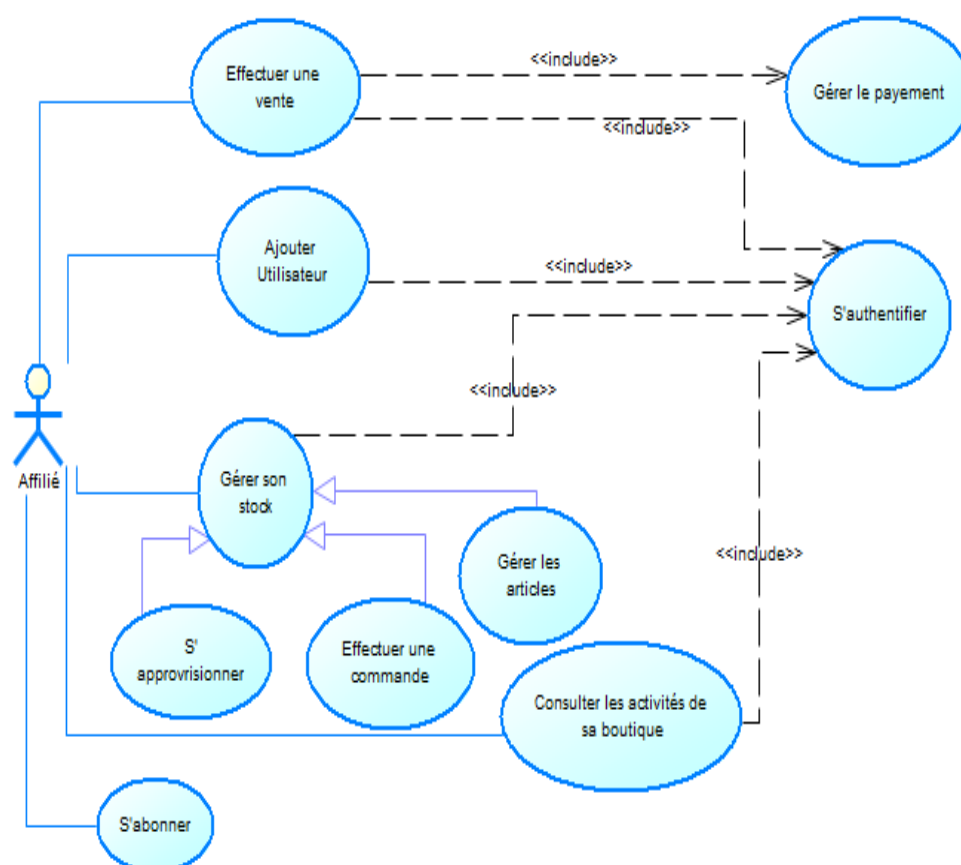


Figure 2 : Diagramme de cas d'utilisation **Affilié**.

- Représentation des cas d'utilisation du projet.

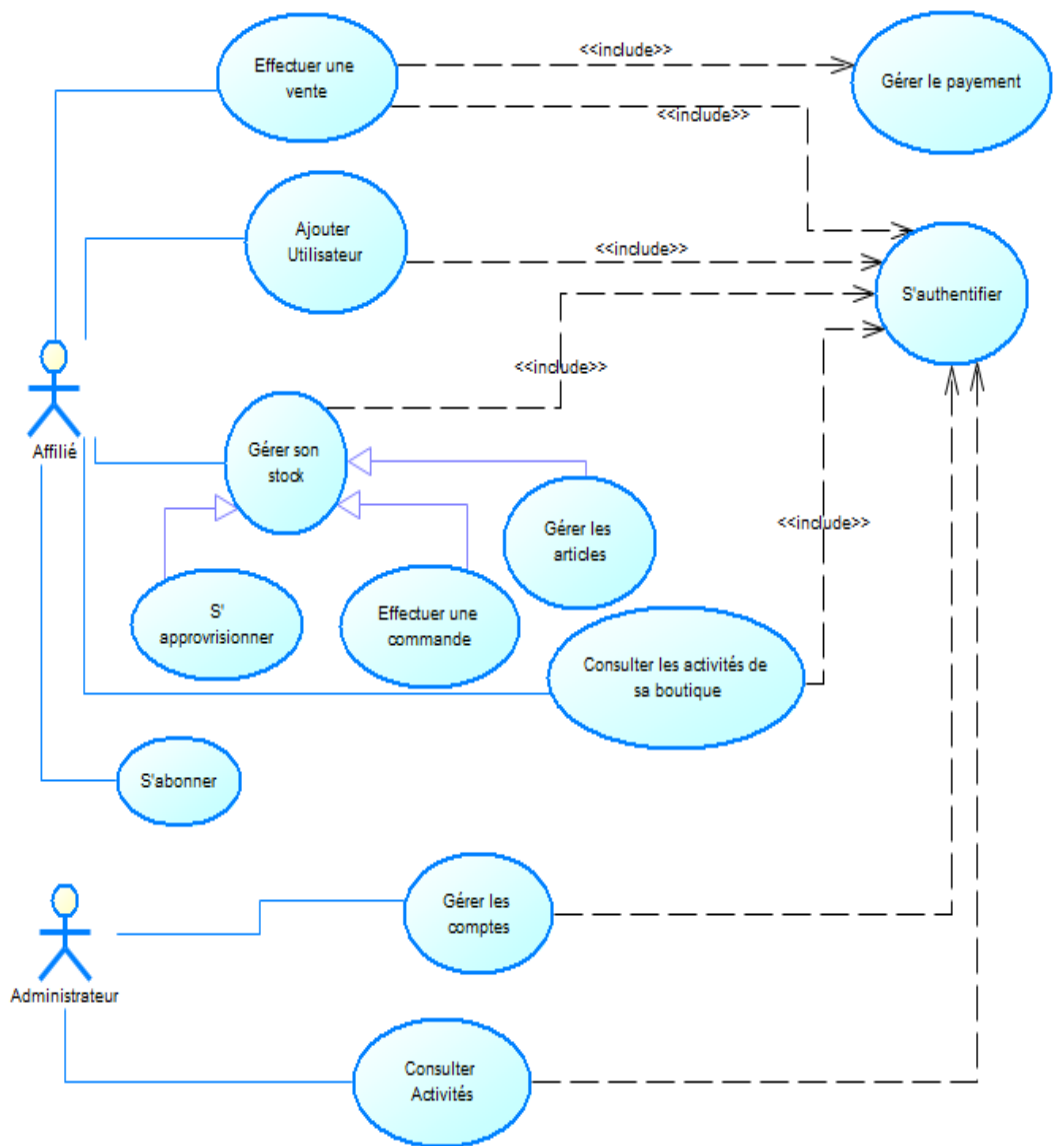


Figure 3 : Diagramme de cas d'utilisation du **Système**.

- Description textuelle de quelques cas d'utilisation.
 - o Cas d'utilisation s'authentifier.

Tableau 3 : Description du cas **S'authentifier**.

Sommaire d'identification.
Titre : S'authentifier.
Résumé : Ce cas permet à l'affilié ou l'administrateur d'avoir accès à la plateforme.
Acteur principal : Affilié.
Date de création : 26 Juillet 2022.
Version : 1.0.
Responsable : HONYIGLO Wilfried
Description des scénarios.
Préconditions :

<ul style="list-style-type: none"> • L'acteur doit avoir un compte ou être inscrit sur la plateforme. • L'acteur doit disposer d'une connexion internet.
Scénario nominal :
<ol style="list-style-type: none"> 1. Le système demande à l'acteur de remplir le formulaire d'authentification. 2. L'acteur remplit le formulaire. 3. Le système contrôle les entrées saisies par l'acteur et renvoie un retour. 4. L'acteur à accès à la plateforme.
Scénarios alternatifs :
<p>SA1 : L'acteur entre une ou des entrées incorrectes. Le scénario commence au point 3.</p> <ol style="list-style-type: none"> 4. Le système informe l'acteur sur la plateforme que les entrées qu'ils a entrées sont incorrectes. 5. L'acteur réessaie le cas d'utilisation en saisissant de nouvelles entrées. 6. Le système vérifie et renvoie un retour. 7. Le système donne accès de la plateforme à l'acteur.
Scénarios d'exceptions :
<p>SE1 : L'acteur annule le cas d'utilisation. Le scénario nominal est interrompu.</p>
Exigences non fonctionnelles.
Le mot de passe est caché lors de la saisie.

- Cas d'utilisation effectuer une vente.

Tableau 4 : Description du cas *Effectuer une vente*.

Sommaire d'identification.
<p>Titre : Effectuer une vente.</p> <p>Résumé : Ce cas permet à l'affilié de vendre un article sur la plateforme.</p> <p>Acteur principal : Affilié.</p> <p>Acteur secondaire : Le client.</p> <p>Date de création :26 Juillet 2022.</p> <p>Version : 1.0.</p> <p>Responsable : HONYIGLO Wilfried</p>
Description des scénarios.
Préconditions :
<ul style="list-style-type: none"> • L'acteur principale doit être authentifié. • La boutique doit avoir un abonnement. • Une demande de vente doit être émise par un client.
Scénario nominal :
<ol style="list-style-type: none"> 1. Le client liste les articles qu'il veut acheter. 2. Le système vérifie que les articles demander sont disponibles en local. 3. L'affilié confirme que les articles sont disponibles en local. 4. L'affilié offre une réduction au client. 5. L'affilié communique le montant total des achats au client. 6. Le client confirme le montant des achats et paie la somme confirmée. 7. L'affilié enregistre la vente. 8. Le système valide la vente et informe que la vente a bien été effectuer.
Scénarios alternatifs :

<p>SA1 : Les articles ne sont pas disponibles en stock. Le scénario commence au point 2.</p> <p>3. L'affilié informe le client que les articles ne sont pas disponibles en local, et lui indique qu'il peut attendre qu'il soit livré.</p> <p>SA2 : Le client modifie la liste des articles qu'il veut acheter. SA2 commence au point 3 du scénario nominal.</p>
<p>Scénarios d'exceptions :</p> <p>SE1 : Les articles sont tous indisponibles. SE1 commence au point 2 du scénario nominal. Le vendeur vérifie si les articles sont disponibles dans un entrepôt distant et communique la réponse au client et met fin au cas.</p> <p>SE2 : Le client n'a pas la somme totale du montant à payer. SE2 commence au point 5 du scénario nominal. Le vendeur met fin au cas.</p>
<p>Exigences non fonctionnelles.</p>

- Cas d'utilisation s'abonner.

Tableau 5 : Description du cas **S'abonner**.

<p>Sommaire d'identification.</p> <p>Titre : S'abonner. Résumé : Ce cas permet à un affilié de faire un abonnement pour avoir accès à la plateforme. Acteur principal : Affilié. Acteur secondaire : Api de Paygate. Date de création : 26 Juillet 2022. Version : 1.0. Responsable : HONYIGLO Wilfried</p>
<p>Description des scénarios.</p>
<p>Préconditions :</p> <ul style="list-style-type: none"> • L'acteur doit avoir un compte ou être inscrit sur la plateforme. • L'acteur doit disposer d'une connexion internet. • Disponibilité de l'api de Paygate. • L'acteur doit avoir de l'argent sur son compte de paiement.
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1. Le système demande à l'acteur de remplir le formulaire d'abonnement tout en choisissant son forfait d'abonnement. 2. L'acteur remplit le formulaire. 3. Le système contrôle les entrées saisies par l'acteur. 4. Le système affiche le formulaire de paiement. 5. L'acteur remplit le formulaire de paiement. 6. Le système contrôle les entrées saisies par l'acteur. 7. Le système envoie les informations à l'api de Paygate. 8. L'api de Paygate renvoie un retour. 9. L'abonnement de l'affilié est validé.
<p>Scénarios alternatifs :</p> <p>SA1 : L'acteur entre une ou des entrées incorrectes. Le scénario commence au point 3 du scénario nominal.</p> <ol style="list-style-type: none"> 4. Le système informe l'acteur sur la plateforme que les entrées qu'ils a entrées sont incorrectes. 5. L'acteur réessaie le cas d'utilisation en saisissant de nouvelles entrées. 6. Le système vérifie et renvoie un retour.

<p>7. Le système donne accès de la plateforme à l'acteur.</p> <p>SA2 : L'api de Paygate renvoie une erreur.</p> <p>Le scénario commence au point 8 du scénario nominal.</p> <p>9. Le système informe l'affilié que la transaction n'a pas abouti et lui demande vérifier ses informations.</p> <p>10. Le système affiche une erreur et renvoie au scénario nominal 1.</p>
<p>Scénarios d'exceptions :</p> <p>SE1 : L'acteur annule le cas d'utilisation.</p> <p>Le scénario nominal est interrompu.</p>
<p>Exigences non fonctionnelles.</p>

B. Le diagramme d'activité

Le diagramme d'activité est un diagramme qui permet de représenter le déclenchement d'évènement en fonction des états du système et de modéliser les comportements du système.

- Cas d'utilisation s'authentifier.

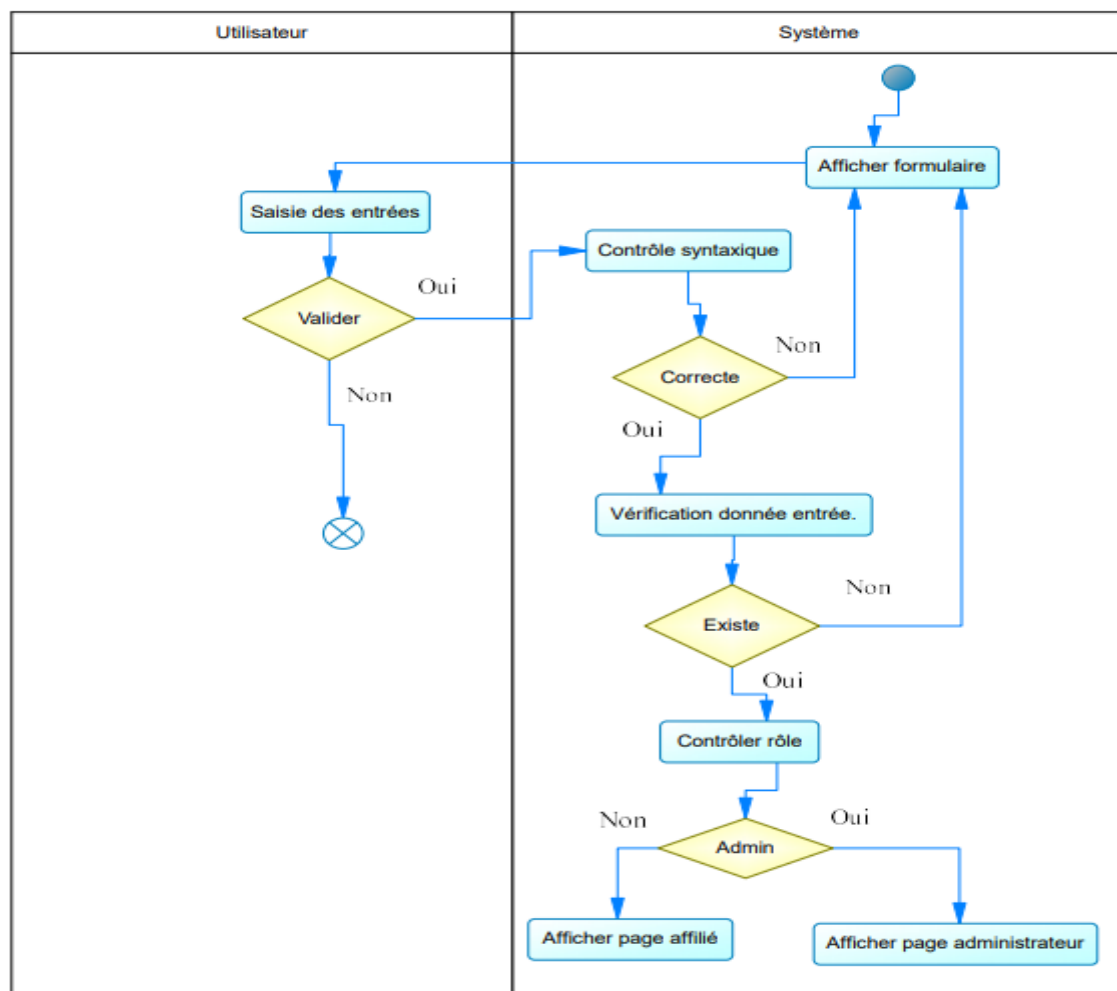


Figure 4 : Diagramme d'activité du cas d'utilisation S'authentifier.

- Cas d'utilisation s'abonner.

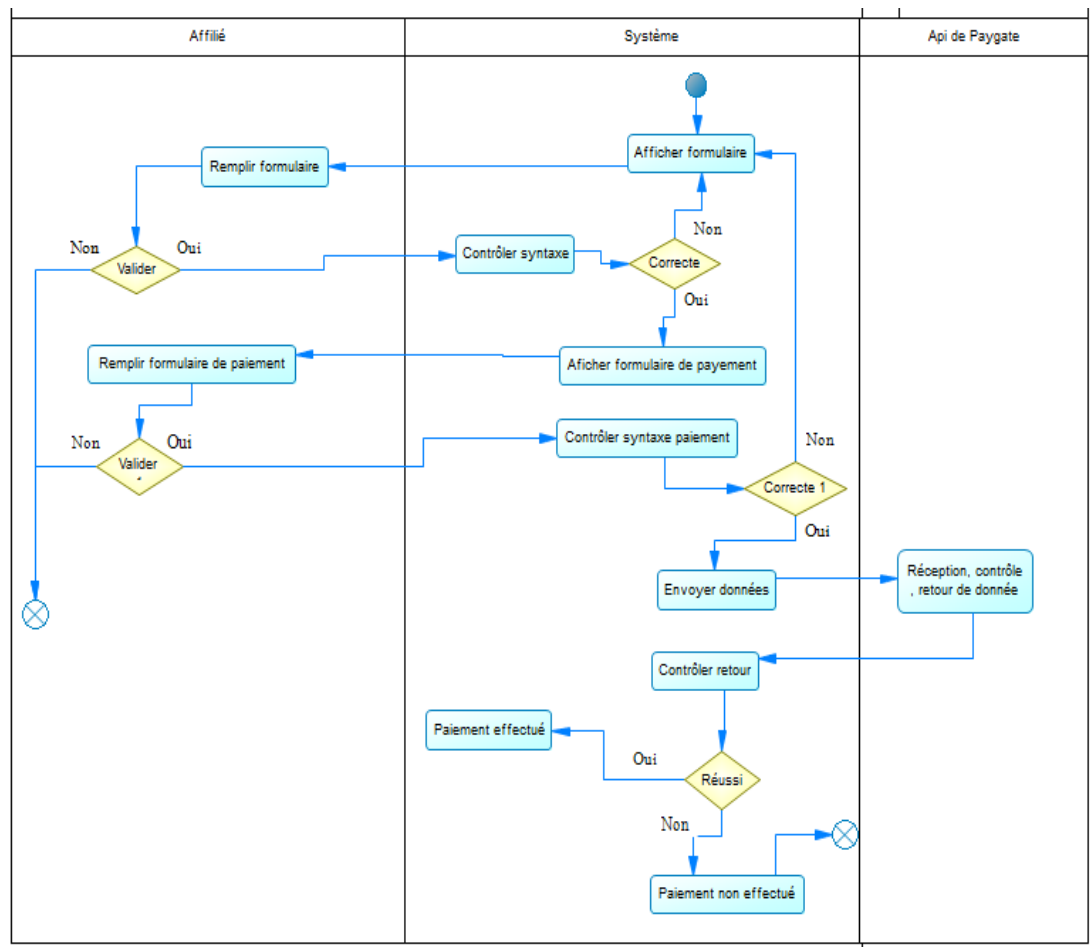


Figure 5 : Diagramme d'activité du cas d'utilisation **S'abonner**.

- Cas d'utilisation effectuer une vente.

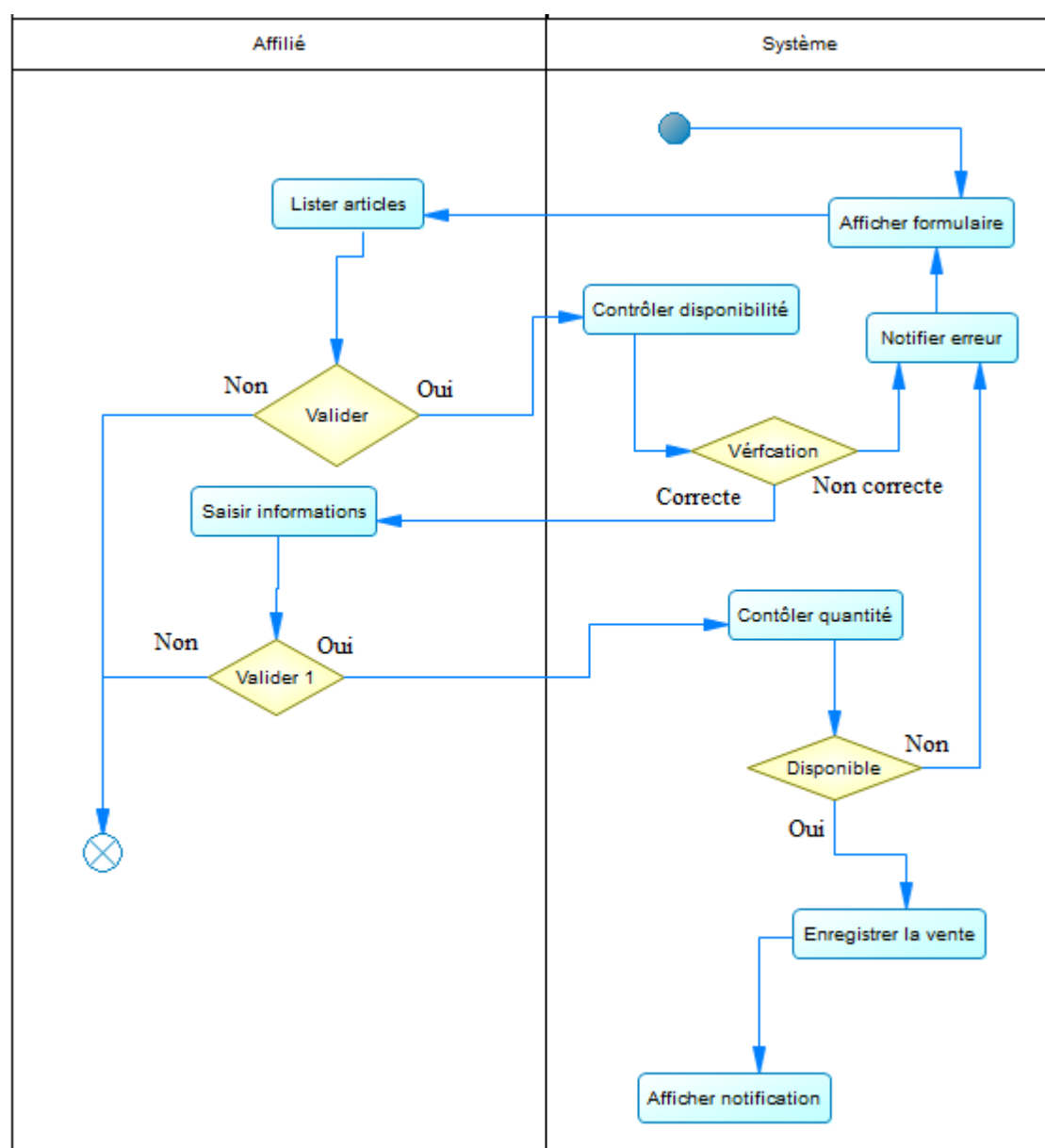


Figure 6 : Diagramme d'activité du cas d'utilisation *Effectuer une vente*.

C. Le diagramme de séquence.

- Cas d'utilisation s'authentifier.

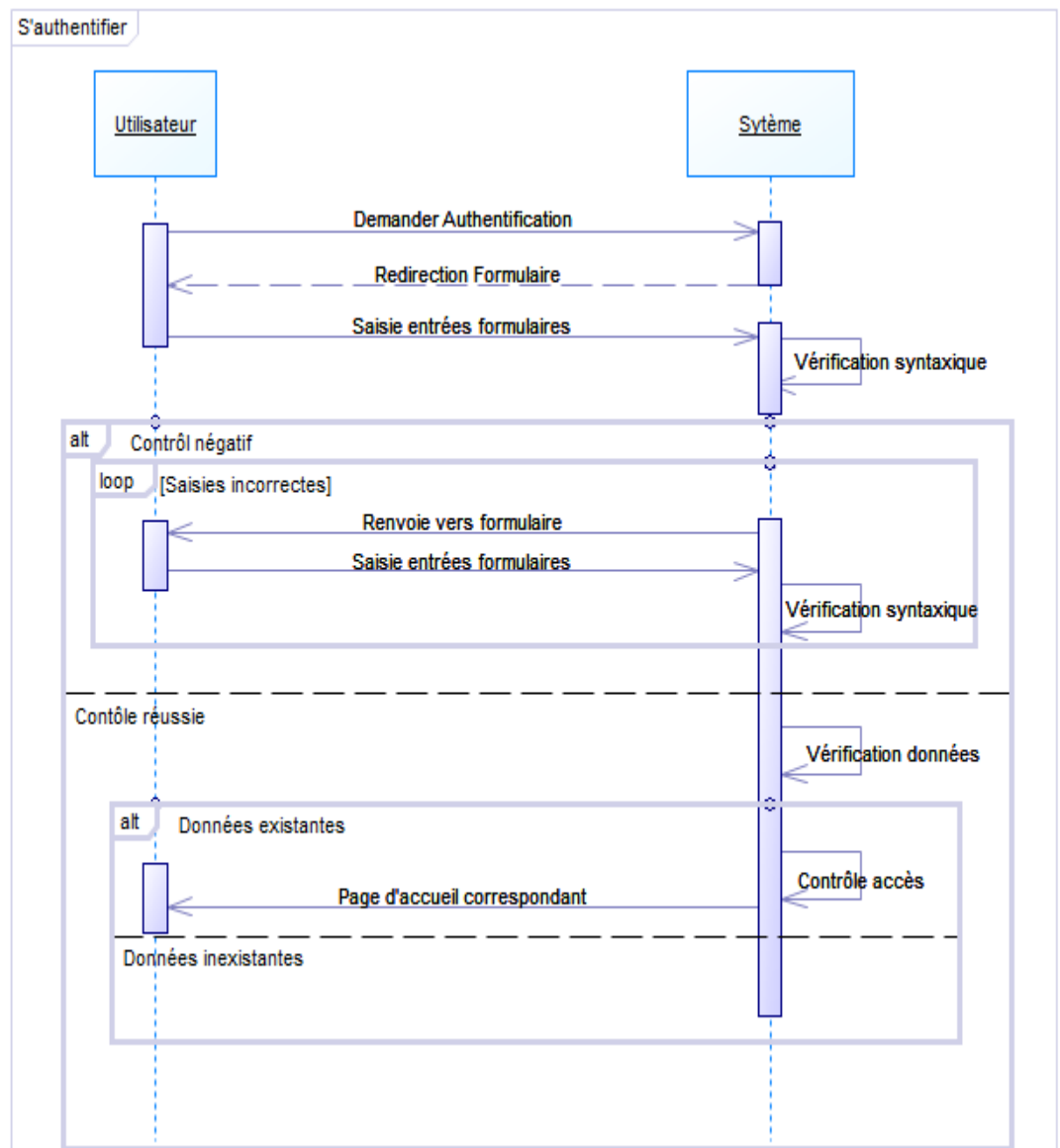


Figure 7 : Diagramme de séquence *S'authentifier*.

- Cas d'utilisation s'abonner.

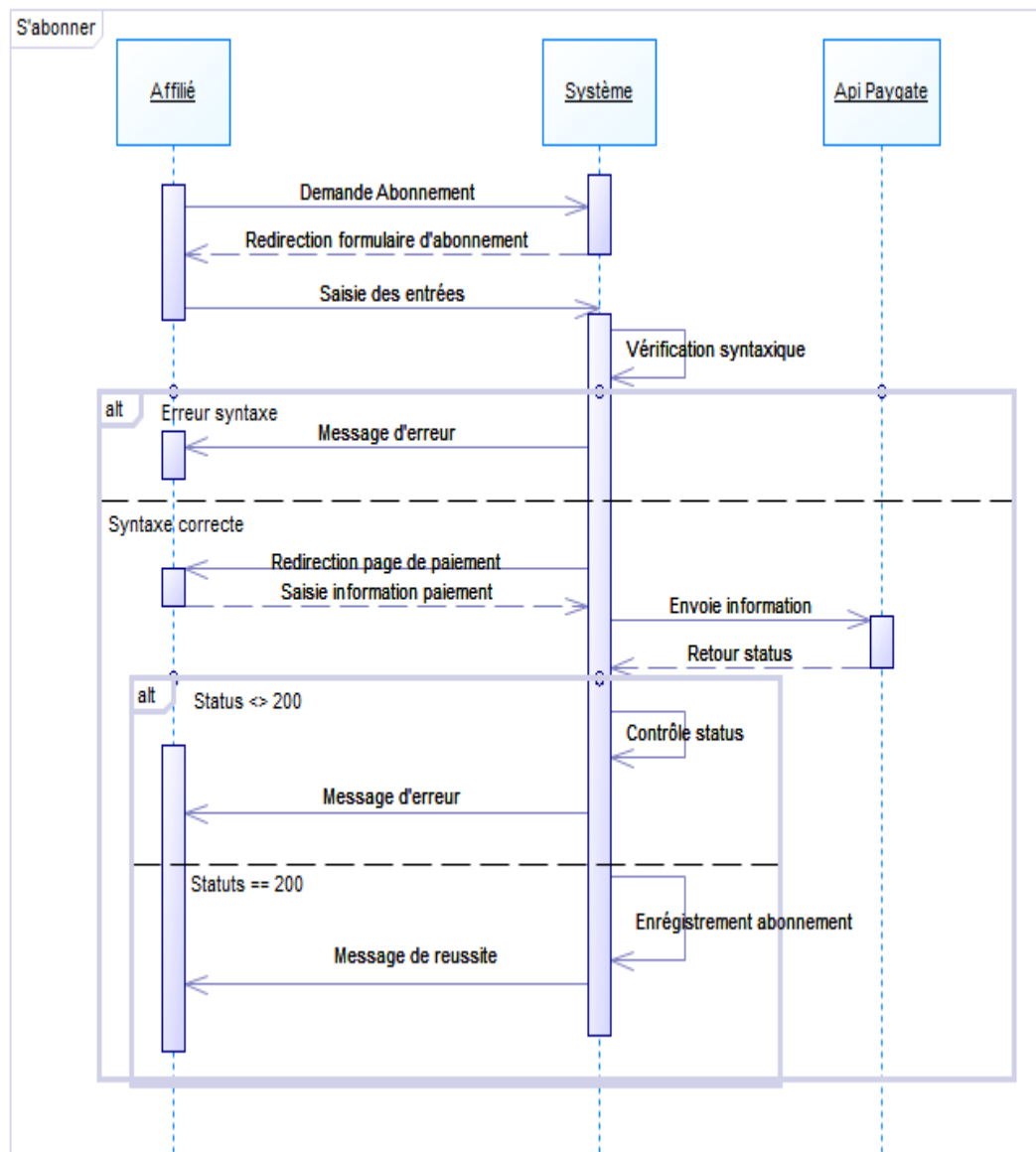


Figure 8 : Diagramme de séquence *S'abonner*.

- Cas d'utilisation effectuer une vente.

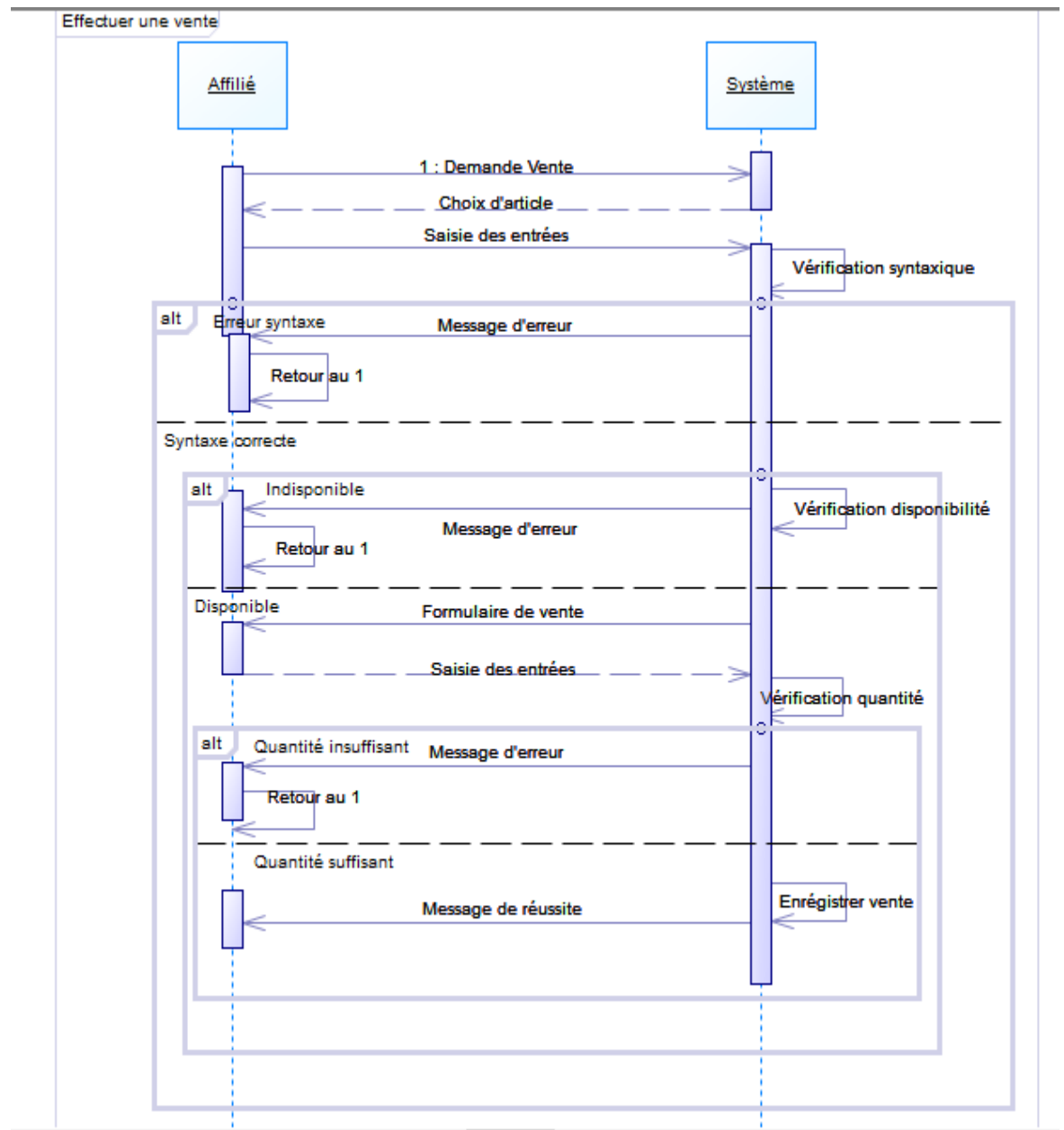


Figure 9 : Diagramme de séquence du cas *Effectuer une vente*.

D. Le diagramme de classes.

Le diagramme est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Une classe décrit les responsabilités et le type d'un ensemble d'objets. Les éléments de cet ensemble les instances de la classe.

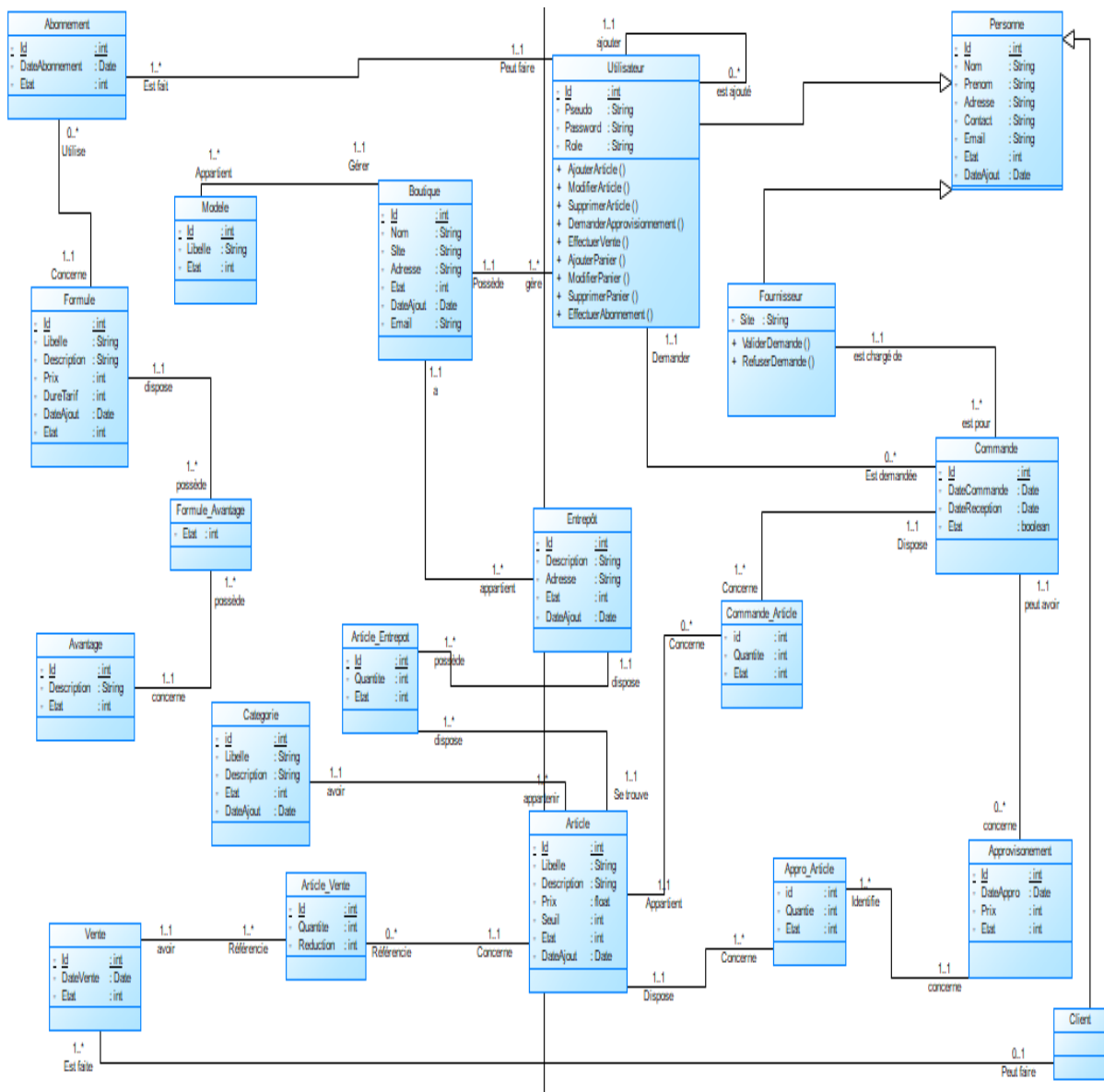


Figure 10 : Diagramme de classe du **Système**.

PARTIE 3 :
REALISATION ET MISE EN ŒUVRE

3 Réalisation et mise en œuvre

3.1 Matériels et logiciels utilisés.

3.1.1 Présentation du langage de modélisation : UML

Le Langage de Modélisation Unifié, de l'anglais *Unified Modeling Language (UML)*, est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). UML 1.0 a été normalisé en janvier 1997 ; UML 2.0 a été adopté par l'OMG en juillet 2005. La dernière version de la spécialisation validée par l'OMG est l'UML 2.5.1 (2017). Cette version propose 14 types de diagrammes dont 7 structurels et comportementaux. Les différents diagrammes réalisés dans la partie 2 tels que le diagramme de cas d'utilisation, le diagramme d'activité, le diagramme de séquence et le diagramme de classe font parties des diagrammes de l'UML. C'est donc à ce dernier que nous avons réalisé ces derniers.

3.1.2 Présentation de l'outil de modélisation : PowerAMC.

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom de AMC Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque PowerDesigner. Ce logiciel est celui qui nous a permis de réaliser les différents diagrammes du langage de modélisation UML.

3.1.3 Matériels

Les matériels qui nous ont permis de développer la solution sont entre autres :

- Un ordinateur portable ;
- Une connexion internet pour les tests et débogages.

Voici les caractéristiques de l'ordinateur portable.

Tableau 6 : Information sur le matériel utilisé.

Marque et modèle	Ecran	Processeur	Ram	Disque dur	Système d'exploitation
Asus.	14.6 pouces	Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz	6,00 Go (5,89 Go utilisable)	256Go	Windows 10 Professionnel

3.1.4 Logiciels

Lors de la réalisation du système nous avons dû utiliser certains logiciels et outils notamment :

- Le langage de programmation PHP ;

PHP est un langage de programmation libre, principalement utilisé pour réaliser des sites ou applications web dynamiques via un serveur http, mais peut également être utilisé comme n'importe quel langage de programmation interprété en local. C'est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code HTML par exemple plus les données demandées pouvant être interprété par le navigateur web. Le langage PHP peut également générer d'autres formats comme le WML, le SVG et le PDF. Il a été conçu pour permettre la création d'application web dynamiques le plus souvent développées pour le web.

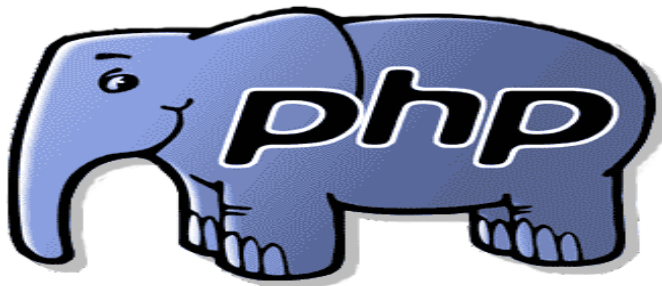


Figure 11: Logo de PHP.

- Le langage de programmation Javascript ;

Javascript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Couplé aux langages HTML et CSS, Javascript est parfois considéré comme l'une des technologies cœurs du World Wide Web. Une grande majorité de sites web l'utilise, et la majorité des navigateurs web disposent d'un moteur Javascript dédié pour l'interpréter pour pouvoir le faire interagir avec la page web.



Figure 12: Logo de Javascript.

➤ Laragon ;

Laragon est un environnement de développement universel portable, isolé, rapide et puissant pour PHP, Node.js, Python, Java, Go, Ruby. Il est rapide, léger, facile à utiliser et facile à étendre.

Laragon est idéal pour la création et la gestion des applications web modernes. Il est axé sur sa plateforme – conçu autour de la stabilité, de la simplicité, de la flexibilité et de la liberté.

Laragon est très léger et restera aussi maigre que possible. Le binaire de base lui-même est inférieur à 2 Mo et utilise moins de 4 Mo de RAM lors de son exécution.

Laragon n'utilise pas les services Windows. Il a son propre service d'orchestration qui gère les services de manière asynchrone et non bloquante afin d'assurer de meilleurs services à ses utilisateurs.



Figure 13 : Logo de *Laragon*.

➤ Environnement de développement intégré : Visual Studio Code ;

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux, MacOS.

Ces fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la compétition intelligente du code, les snippets, la refactorisation du code et bien d'autres fonctionnalités. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.



Figure 14 : Logo de *Visual Studio Code*.

➤ Le Framework Laravel ;

Laravel est un Framework dédié à PHP respectant le principe modèle-vue-contrôleur (MVC) et est orienté objet. Laravel a été créé en Juin 2011 par **Taylor Otwell**. En peu de temps, une communauté d'utilisateur du framework s'est constituée, et est devenu en 2016 le projet PHP le mieux noté. La version retenue pour ce projet est la version 9.

Il a permis de créer l'application web d'administration du web service car il est facile à prendre en main et permet un développement rapide d'application grâce à son vaste écosystème d'outils.



Figure 15 : Logo de *Laravel*.

➤ GIT ;

GIT est un outil de gestion de versions décentralisés. Il se distingue par sa rapidité et sa gestion des branches qui permettent de développer en parallèle des nouvelles fonctionnalités. C'est ainsi qu'avec GIT Bash du GIT, nous avons géré nos dépendances et mis à jour les différents modules du code.



Figure 16 : Logo de *Git*.

➤ GitHub

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de versions GIT. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de

fonctionnalités, la gestion des tâches et une description pour chaque projet. Il nous a permis d'effectuer le versioning et de faciliter le suivi et le développement du projet.



Figure 17: Logo de *GitHub*.

➤ Bootstrap

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation, interaction avec le navigateur, etc...) de sites et d'applications web. Il se base sur un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style aux différents composants d'une page web. Nous avons fait le choix de Bootstrap 5 car en plus de sa simplicité, c'est l'un des projets Open Source les plus populaires sur la plateforme de gestion de développement GitHub.



Figure 18 : Logo de *Bootstrap*.

➤ JQuery ;

JQuery est une bibliothèque Javascript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code Html des pages web. La première version est lancée en Janvier 2006 par **John Resig**, aujourd'hui JQuery est sa version 3.6.0. JQuery est facilite la manipulation du code Html, la gestion des évènements, l'animation ainsi que le fonction AJAX grâce à sa bibliothèque compatible avec la plupart des navigateurs. Avec aujourd'hui près de la moitié des sites internet intègre sa bibliothèque. JQuery est à l'heure actuelle le framework frontend le plus utilisé dans le monde. La version retenue pour le développement de notre projet est la 3.6.0.



Figure 19 : Logo de JQuery.

3.2 Sécurité de l'application

Pour assurer la sécurité de notre application, nous disposons d'un certains nombres de règles de sécurité qu'on doit assumer au cours du développement. Des règles concernant :

- L'authentification : L'accès à l'application est conditionné par la connexion de l'utilisateur. Ainsi chaque utilisateur est sûr que ses données ou informations sont accessibles que par lui ou bien par des utilisateurs de sa boutique.
- Les mots de passe : Les mots de passe sont cryptés pour chaque utilisateur. De cette manière même les administrateurs de l'application ne peuvent connaître les informations d'une boutique.
- Les comptes utilisateurs : Chaque utilisateur possède un seul compte utilisateur lui permettant d'accéder ou non aux différentes fonctionnalités de l'application.
- Les tâches pouvant engendrer des pertes d'information tel que la suppression sont conditionnées par une confirmation de ce fait aucune suppression malencontreuse ne pourra être observée.

3.3 Evaluation financière de la solution

3.4 Présentation de l'application

3.4.1 Mise en place de la base de données

Pour éviter que le document ne soit trop volumineux, nous ne détaillerons pas toute la structure de la base de données.

```
-- Listage de la structure de la base pour ieboutique
CREATE DATABASE IF NOT EXISTS `ieboutique` /*!40100 DEFAULT CHARACTER SET
utf8mb4 */;
USE `ieboutique`;

-- Listage de la structure de la table ieboutique. abonnements
CREATE TABLE IF NOT EXISTS `abonnements` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Tarif_Id` bigint(20) unsigned NOT NULL,
  `DateAbonnement` date NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  `User_id` bigint(20) unsigned NOT NULL,
  PRIMARY KEY (`id`),
  KEY `abonnements_tarif_id_foreign` (`Tarif_Id`),
  CONSTRAINT `abonnements_tarif_id_foreign` FOREIGN KEY (`Tarif_Id`)
REFERENCES `tarifs` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. articles
CREATE TABLE IF NOT EXISTS `articles` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Libelle` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Prix` int(11) NOT NULL,
  `Seuil` int(11) NOT NULL,
  `Etat` int(11) NOT NULL,
  `DateAjout` date NOT NULL DEFAULT '2022-07-16',
  `CategorieId` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `articles_libelle_unique` (`Libelle`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. article_entrepot
CREATE TABLE IF NOT EXISTS `article_entrepot` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `ArticleId` bigint(20) unsigned NOT NULL,
  `EntrepotId` bigint(20) unsigned NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  `Quantite` int(11) NOT NULL DEFAULT '20',
  PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=44 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. avantages
CREATE TABLE IF NOT EXISTS `avantages` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. boutiques
CREATE TABLE IF NOT EXISTS `boutiques` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Nom` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Site` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Adresse` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Etat` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `DateAjout` date NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `boutiques_email_unique` (`Email`)
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. categories
CREATE TABLE IF NOT EXISTS `categories` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Libelle` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  `UserId` bigint(20) unsigned NOT NULL,
  `DateAjout` datetime NOT NULL DEFAULT '2022-07-15 14:08:01',
  PRIMARY KEY (`id`),
  KEY `categories_userid_foreign` (`UserId`),
  CONSTRAINT `categories_userid_foreign` FOREIGN KEY (`UserId`) REFERENCES
`utilisateurs` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=39 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```



```
-- Listage de la structure de la table ieboutique. entrepots
CREATE TABLE IF NOT EXISTS `entrepots` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Adresse` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `DateAjout` datetime NOT NULL DEFAULT '2022-07-16 01:55:42',
  `Etat` int(11) NOT NULL DEFAULT '0',
  `BoutiqueId` bigint(20) unsigned NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. historiques
CREATE TABLE IF NOT EXISTS `historiques` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `DateOperation` datetime NOT NULL,
  `Libelle` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=53 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. migrations
CREATE TABLE IF NOT EXISTS `migrations` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `migration` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `batch` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. modeles
CREATE TABLE IF NOT EXISTS `modeles` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `DateAjout` datetime NOT NULL DEFAULT '2022-07-23 17:12:19',
  `Etat` int(11) NOT NULL DEFAULT '0',
  `Quantite` int(11) NOT NULL,
  `BoutiqueId` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
```

```
-- Listage de la structure de la table ieboutique. personal_access_tokens
CREATE TABLE IF NOT EXISTS `personal_access_tokens` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `tokenable_type` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `tokenable_id` bigint(20) unsigned NOT NULL,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `token` varchar(64) COLLATE utf8mb4_unicode_ci NOT NULL,
  `abilities` text COLLATE utf8mb4_unicode_ci,
  `last_used_at` timestamp NULL DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `personal_access_tokens_token_unique` (`token`),
  KEY `personal_access_tokens_tokenable_type_tokenable_id_index`
  (`tokenable_type`,`tokenable_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. personnes
CREATE TABLE IF NOT EXISTS `personnes` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Nom` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Prenom` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Adresse` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  `Contact` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `DateAjout` datetime NOT NULL DEFAULT '2022-07-09 10:54:11',
  PRIMARY KEY (`id`),
  UNIQUE KEY `personnes_email_unique` (`Email`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. tarifs
CREATE TABLE IF NOT EXISTS `tarifs` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Libelle` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Prix` int(11) NOT NULL,
  `DureeTarif` int(11) NOT NULL,
  `DateAjout` date NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
```

```

    UNIQUE KEY `tarifs_libelle_unique` (`Libelle`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. tarif_avantage
CREATE TABLE IF NOT EXISTS `tarif_avantage` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `tarif_id` bigint(20) unsigned NOT NULL,
  `avantage_id` bigint(20) unsigned NOT NULL,
  `Etat` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  KEY `tarif_avantage_tarif_id_foreign` (`tarif_id`),
  KEY `tarif_avantage_avantage_id_foreign` (`avantage_id`),
  CONSTRAINT `tarif_avantage_avantage_id_foreign` FOREIGN KEY (`avantage_id`)
REFERENCES `avantages` (`id`),
  CONSTRAINT `tarif_avantage_tarif_id_foreign` FOREIGN KEY (`tarif_id`)
REFERENCES `tarifs` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=229 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

-- Listage de la structure de la table ieboutique. utilisateurs
CREATE TABLE IF NOT EXISTS `utilisateurs` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `Pseudo` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Role` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT 'User',
  `BoutiqueId` bigint(20) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `utilisateurs_pseudo_unique` (`Pseudo`),
  CONSTRAINT `utilisateurs_id_foreign` FOREIGN KEY (`id`) REFERENCES
`personnes` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

```

3.4.2 Plan de navigation

- Administrateur ;



Figure 20 : Plan de navigation **Administrateur**.

➤ Affilié ;

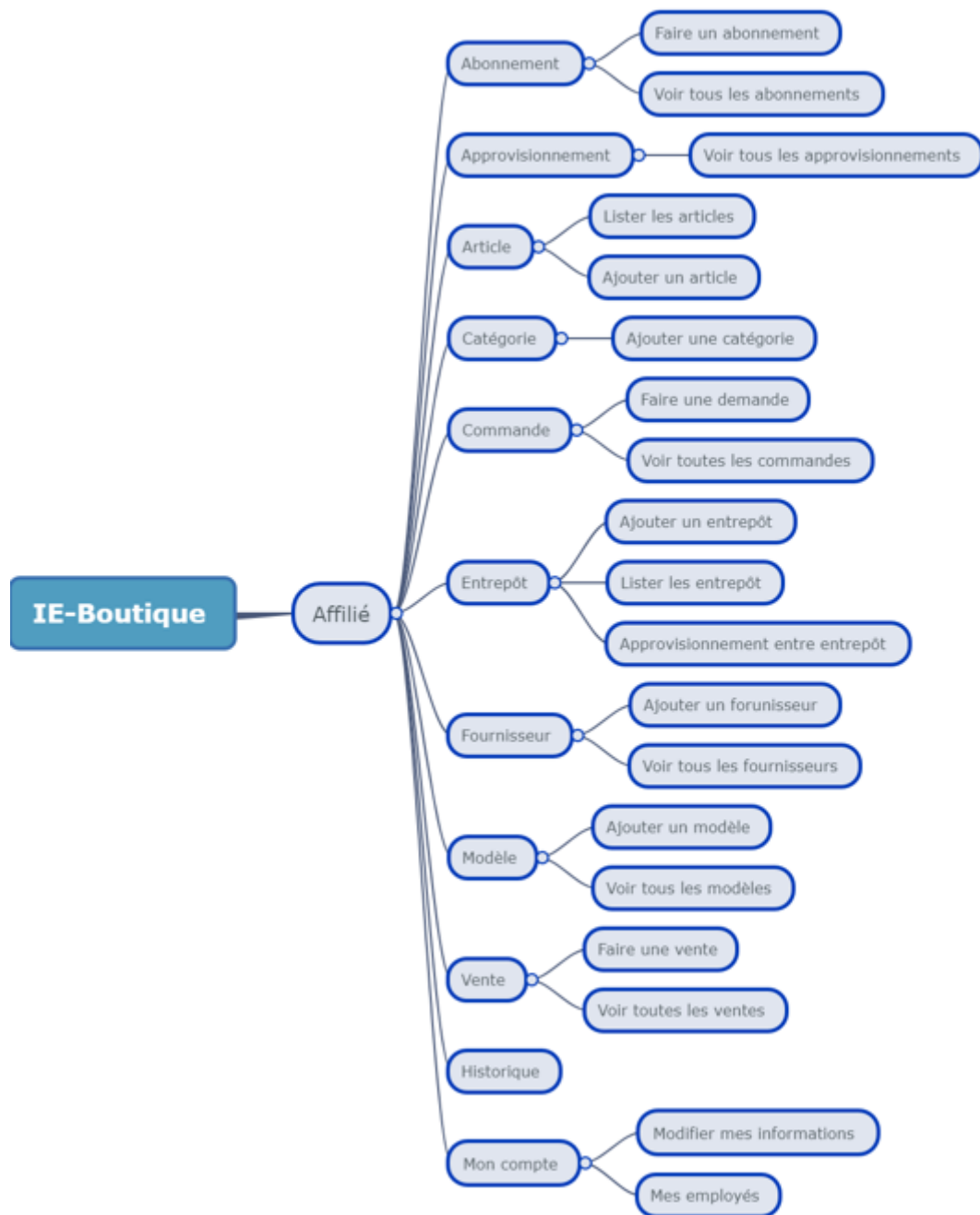


Figure 21 : Plan de navigation *Affilié*.

➤ Global.

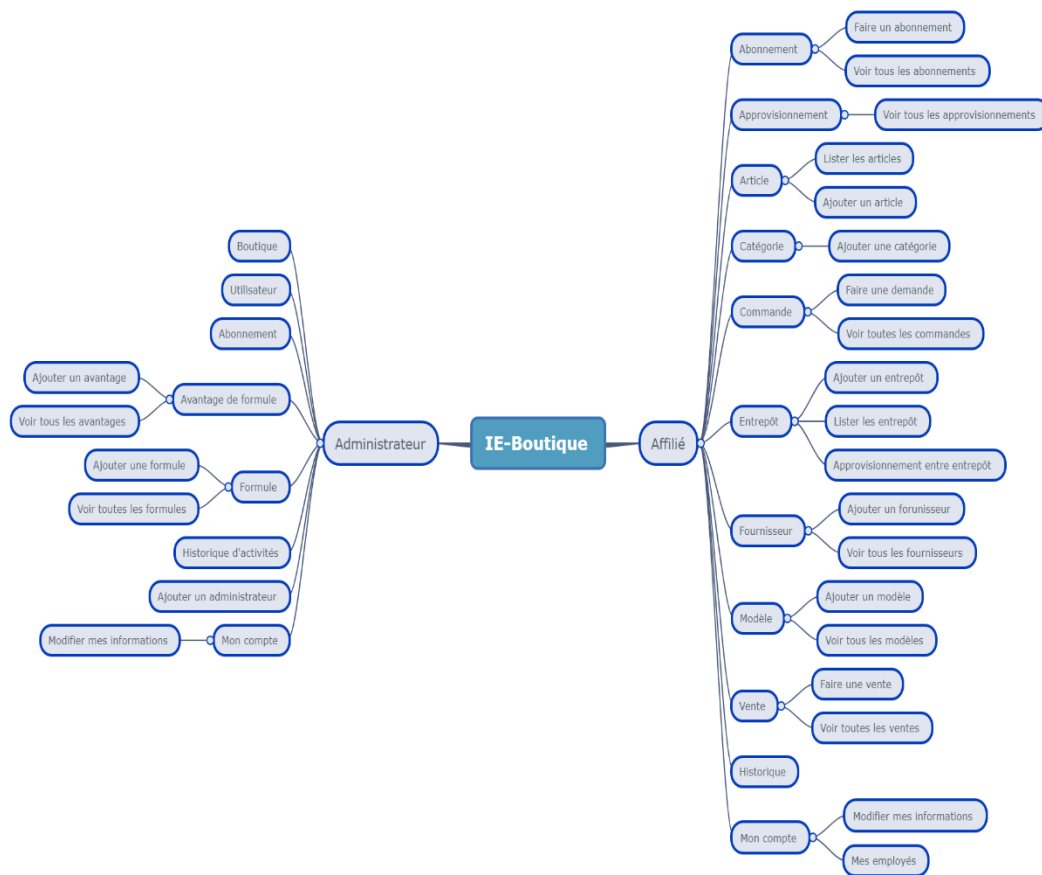


Figure 22 : Plan de navigation **Global**.

3.4.3 Quelques masques de saisie

3.4.4 Quelques états et statistiques

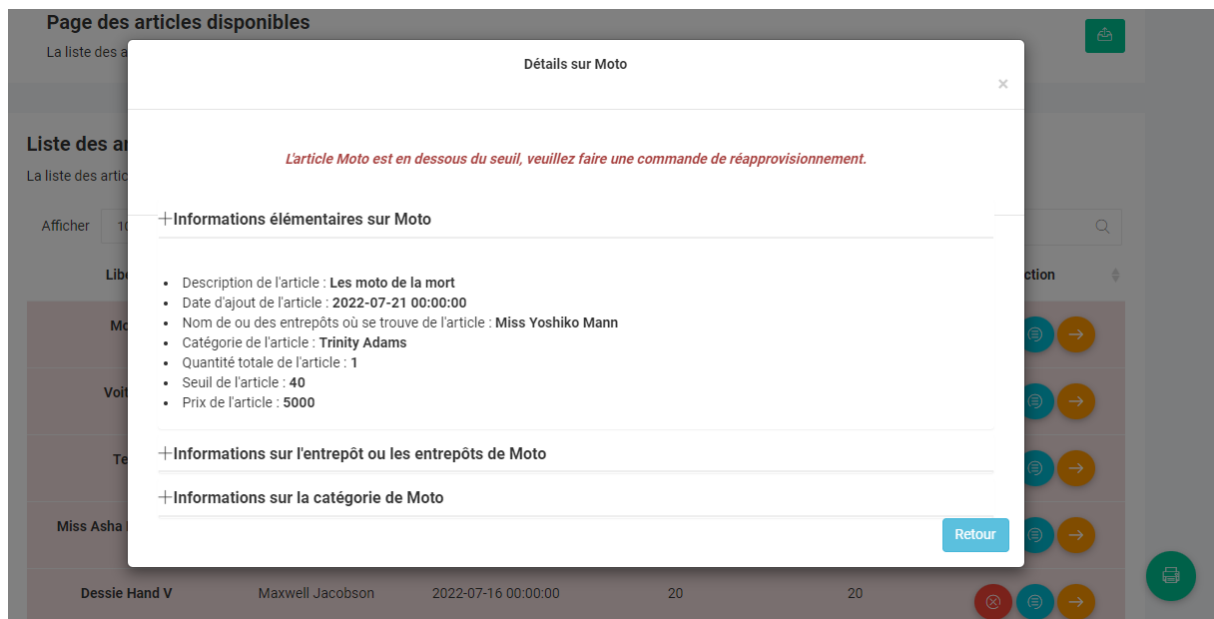


Figure 23 : Détails d'un article

Liste des ventes de votre boutique.

La liste des ventes s'affichera en dessous, vous pouvez rechercher une vente en particulier ou également trier la liste des ventes obtenues.

Afficher éléments

Date de la vente	Client	Nombre d'articles vendus	Quantité total d'articles vendus	Prix total	Action
2022-08-06 11:08:05	Pas de client pour cette vente	1	8	1940 fcfa	  
2022-08-06 11:06:00	Pas de client pour cette vente	1	5	1940 fcfa	  
2022-08-05 14:35:23	F F1	1	4	1980 fcfa	  
2022-08-05 14:34:36	Fangbémi Edith	2	7	3940 fcfa	  
2022-08-05 11:07:40	Fangbémi Edith	1	3	1920 fcfa	  
2022-08-04 00:00:00	Pas de client pour cette vente	2	6	3800 fcfa	  

Figure 24 : Liste des ventes.

Afficher
éléments
























Libelle	Description	Date d'ajout	Seuil	Quantité disponible	Action
Moto	Les moto de la mort	2022-07-21 00:00:00	40	1	  
Voiture	Les voitures qui filent	2022-07-21 00:00:00	40	1	  
Test	Les fruits de la vie	2022-07-21 00:00:00	40	3	  
Miss Asha Prosacco II	Abby Zemlak PhD	2022-07-16 00:00:00	20	12	  
Dessie Hand V	Maxwell Jacobson	2022-07-16 00:00:00	20	20	  
Layne Huel Sr.	Miss Ena Jenkins II	2022-07-16 00:00:00	20	20	  
Dr. Woodrow Feil Jr.	Opal Mills III	2022-07-16 00:00:00	20	20	  
Errolu O'Kon	Jude Dickinson I	2022-07-16 00:00:00	20	20	  

Figure 25 : Liste des articles.

CONCLUSION

Notre projet avait pour objectif de mettre en place une application web permettant la gestion des stocks tout en gardant une historique des activités d'une boutique. Cette application devrait être capable d'aider les affiliés à ajouter des articles, les modifier, les vendre tout en enregistrant les différentes informations de chaque opération, les tenir informer lorsqu'un article est en dessous du seuil d'approvisionnement, de notifier aux affiliés les détails de leurs actions posés sur l'application. Ce projet a été pour nous une occasion en or de sortir du cadre théorique et d'appliquer les connaissances acquises lors de notre formation à l'IAI-Togo dans un environnement réel de travail. Ce qui nous a également permis de nous initier dans le domaine professionnel et d'apprendre plusieurs attitudes et habitudes sociales.

Sur le plan technique, ce travail nous a permis de découvrir de nouvelles notions qui nous étaient inconnus, comme par l'utilisation du Framework Laravel qui est très reconnu dans le développement web et de ses modules complémentaires en plus de nous permettre de maîtriser le langage PHP. Dans la première partie, le cahier des charges, nous avons présenté notre centre de formation et le cadre général de notre stage, fait l'étude descriptive de notre projet. Dans la deuxième partie, celle de l'analyse et de la conception, nous avons effectué une étude détaillée de notre projet en utilisant la méthode UML. Ensuite, la troisième partie nous a permis de décrire l'environnement technique et les outils de développement de notre application et aussi de présenter la structure de la base de données, ainsi que quelques interfaces utilisateurs et codes sources de notre application. La première version de cette application ne saurait combler totalement les attentes des utilisateurs. Ainsi, lorsque de nouveaux besoins se feront sentir, de nouvelles fonctionnalités constitueront de nouvelles versions de l'application.

BIBLIOGRAPHIE INDICATIVE

- UML 2 en action 01/03/2007 Edition EYROLLES 4^{ème} édition de Pascal Roques et Franck Vallée
- UML 2 modéliser une application web. Edition EYROLLES 4^{ème} édition de Pascal Vallée.
- Documentation Laravel 9.

WEBOGRAPHIE INDICATIVE

- Le site web de GITHUB : **<https://github.com>** consulté tout au long du projet.
- Le site web Wikipédia : **<https://fr.wikipedia.com>** consulté tout au long du projet.
- Le géant Stackoverflow : **<https://stackoverflow.com>** consulté tout au long du projet.
- Le site YouTube : à l'adresse **<https://www.youtube.com>** consulté tout au long du projet.

DOCUMENTS ANNEXES

Anciens mémoires consultés :

- « Mise en place d'une plateforme mobile de recherche et de diffusion de stage » de **AMEYIKPO Sewa Jospin Beni (2019-2020)**.
- « Mise en place d'une plateforme pour la gestion informatisée du transport en commun au Togo » de **MABANTEY Siegfried Dametote (2019-2020)**.
- « Mise en place d'une plateforme web de publication d'offres d'emplois, de stages et de gestion de recrutement » de **SEGLA Ayawoavi Rebecca**.

TABLES DES MATIERES

REMERCIEMENTS	I
A DIEU TOUT PUISSANT.....	ERREUR ! SIGNET NON DEFINI.
A MA MERE	ERREUR ! SIGNET NON DEFINI.
A MON PERE	ERREUR ! SIGNET NON DEFINI.
A MA TANTE.....	ERREUR ! SIGNET NON DEFINI.
AVANT-PROPOS	ERREUR ! SIGNET NON DEFINI.
RESUME	ERREUR ! SIGNET NON DEFINI.
LISTE DES PARTICIPANTS AU PROJET	ERREUR ! SIGNET NON DEFINI.
SOMMAIRE	II
LISTE DES FIGURES	III
LISTE DES TABLEAUX	III
INTRODUCTION.....	1
PARTIE 1 : CAHIER DES CHARGES.....	ERREUR ! SIGNET NON DEFINI.
1 CAHIER DES CHARGES	3
1.1 PRESENTATION DU SUJET.....	3
1.2 PROBLEMATIQUE DU SUJET	3
1.3 UTILISATEURS DU SYSTEME	3
1.4 INTERET DU SUJET	3
1.4.1 Objectifs.....	3
1.4.2 Les Résultats.....	4
1.5 LES CIBLES	4
1.6 TYPE D'APPLICATION	4
1.7 EQUIPEMENTS DES CIBLES.....	4
PARTIE 2 : PRE-PROGRAMMATION	ERREUR ! SIGNET NON DEFINI.
2 PREPROGRAMMATION	6
2.1 ETUDE DE L'EXISTANT	6
2.2 CRITIQUE DE L'EXISTANT	6
2.3 PLANNING PREVISIONNEL DE REALISATION	7
2.4 ETUDE DETAILLEE DE LA SOLUTION	7
2.4.1 Etape 1 : Etude préliminaire.....	7
2.4.2 Etape 2 : Capture des besoins fonctionnels.....	8
PARTIE 3 : REALISATION ET MISE EN ŒUVRE.....	ERREUR ! SIGNET NON DEFINI.
3 REALISATION ET MISE EN ŒUVRE.....	21
3.1 MATERIELS ET LOGICIELS UTILISES.	21
3.1.1 Présentation du langage de modélisation : UML	21
3.1.2 Présentation de l'outil de modélisation : PowerAMC.....	21
3.1.3 Matériels.....	21
3.1.4 Logiciels.....	22

3.2	SECURITE DE L'APPLICATION	26
3.3	EVALUATION FINANCIERE DE LA SOLUTION	26
3.4	PRESENTATION DE L'APPLICATION	26
3.4.1	<i>Mise en place de la base de données</i>	26
3.4.2	<i>Plan de navigation</i>	31
3.4.3	<i>Quelques masques de saisie</i>	34
3.4.4	<i>Quelques états et statistiques</i>	35
CONCLUSION		37
BIBLIOGRAPHIE INDICATIVE		VI
TABLES DES MATIERES		VI