

**REPUBLIQUE TOGOLAISE**

-----  
**Travail-Liberté-Patrie**

**MINISTRE DE LA PLANIFICATION  
DU DEVELOPPEMENT  
ET DE LA COOPERATION**



Institut Africain d'informatique –  
Représentation du TOGO (IAI-TOGO)

**Tel** : 22 20 47 00

**E-mail** : [iaitogo@iai-togo.tg](mailto:iaitogo@iai-togo.tg)

**Site Web** : [www.iai-togo.tg](http://www.iai-togo.tg)

07 BP 12456 Lomé 07, TOGO

## **RAPPORT DE STAGE PRATIQUE**

**Type de stage : Génie Logiciel**

**THEME DE STAGE**

**GESTION INFORMATISEE DES COMMANDES ET DES  
LIVRAISONS DE SACHETS D'EAU : CAS DE <<SHINE  
WATER>>**

**Période : 27 juillet 2020 au 26 septembre 2020**

Rédigé et présenté par : **WOLF Kodjo Adelphe Daniel**

**Etudiant en deuxième année Tronc Commun**

**Année Académique : 2019 – 2020**

**SUPERVISEUR**

**M. ALADJI Kodjo Michel**

**Enseignant à l'IAI-TOGO**

## Sommaire

<b>INTRODUCTION</b> .....	1
<b>PARTIE I : RAPPORT D'INSERTION</b> .....	2
<b>I-PRESENTATIONS</b> .....	3
<b>II-THEME DU STAGE</b> .....	4
<b>PARTIE II : RAPPORT DE PRE-PROGRAMMATION</b> .....	6
<b>I-ETUDE DE L'EXISTANT</b> .....	7
<b>II-CRITIQUE DE L'EXISTANT</b> .....	7
<b>III-PROPOSITION DE SOLUTIONS</b> .....	8
<b>IV-EVALUATION FINANCIERE DES SOLUTIONS</b> .....	9
<b>V-CHOIX DE SOLUTION</b> .....	11
<b>VI- PLANNING PREVISIONNEL DE REALISATION</b> .....	13
<b>VII-PRESENTATION DE LA METHODE D'ANALYSE</b> .....	15
<b>VIII-PRESENTATION DE L'OUTIL DE MODELISATION</b> .....	17
<b>IX-ETUDE DETAILLEE DE LA SOLUTION</b> .....	19
<b>D-VUE LOGIQUE</b> .....	37
<b>E-VUE DE DEPLOIEMENT</b> .....	39
<b>PARTIE III : REALISATION ET MISE EN OEUVRE</b> .....	40
<b>I-MISE EN OEUVRE</b> .....	41
<b>II. PRESENTATION DE L'APPLICATION</b> .....	46
<b>CONCLUSION</b> .....	63
<b>Table des matières</b> .....	68

## INTRODUCTION

La découverte de l'informatique a facilité plusieurs activités de la vie courante. Actuellement, les individus peuvent facilement traiter des informations en se servant des logiciels et des réseaux informatiques. Compte tenu de son évolution, ce système caractérise la majorité des grandes entreprises quel que soit le secteur d'activité.

Nous sommes dans un monde de plus en plus connecté, de plus en plus reliés aux outils informatiques. La technologie est aujourd'hui un élément fondamental pour toute entreprise. L'évolution dans le domaine de l'informatique offre des opportunités pour améliorer les processus de flux de travaux, favoriser l'innovation et acquérir un avantage concurrentiel.

Dans le cadre de la formation des Ingénieurs des Travaux Informatiques, l'Institut Africain d'Informatique représentation du TOGO (IAI-TOGO) a mis en place, pour les étudiants de deuxième (2<sup>ème</sup>) année, un stage pratique de deux (2) mois en entreprise. Son objectif est la mise en pratique des connaissances en analyse et programmation acquises pendant les deux premières années de formation, et la réalisation d'un projet concret, qui devra fournir une application viable et prête à l'emploi.

A cet effet, nous avons effectué un stage du 27 Juillet 2020 au 26 Septembre 2020 à IAI-TOGO. Notre travail consistait à développer une application de gestion des commandes et livraisons, d'où le thème de notre stage : Gestion informatisée des commandes et des livraisons de sachets d'eau : cas de <<SHINE WATER>>.

Notre document sera réparti en trois (03) grandes parties : le rapport d'insertion, le rapport de pré-programmation et la réalisation et mise en œuvre.

---

## *PARTIE I : RAPPORT D'INSERTION*

---

## I-PRESENTATIONS

### 1-PRESENTATION DE L'IAI-TOGO

L'institut Africain d'Informatique (IAI) est une école supérieure en informatique. La convention portant création de l'institut et les statuts y afférent ont été signés le 29 janvier 1972 à Fort Lamy (actuel N'Djamena) en république du TCHAD.

L'accord de siège entre l'IAI et le GABON (siège de l'Institut) a été signé en janvier 1975. Il est par conséquent un établissement Inter-États d'Enseignement Supérieur.

En application de la décision du Conseil d'Administration de délocaliser l'IAI, la Représentation du TOGO (IAI-TOGO) a ouvert ses portes le 24 Octobre 2002. IAI-TOGO est situé sur la rue de la Kozah à Nyékonakpoè, logé dans le bâtiment du Centre National d'Études et de Traitements Informatique (CENETI) et placé sous la tutelle du Ministère de la

Planification du Développement. L'accord d'établissement entre la République Togolaise et l'Institut Africain d'Informatique a été signé le 12 mai 2006 à Lomé.

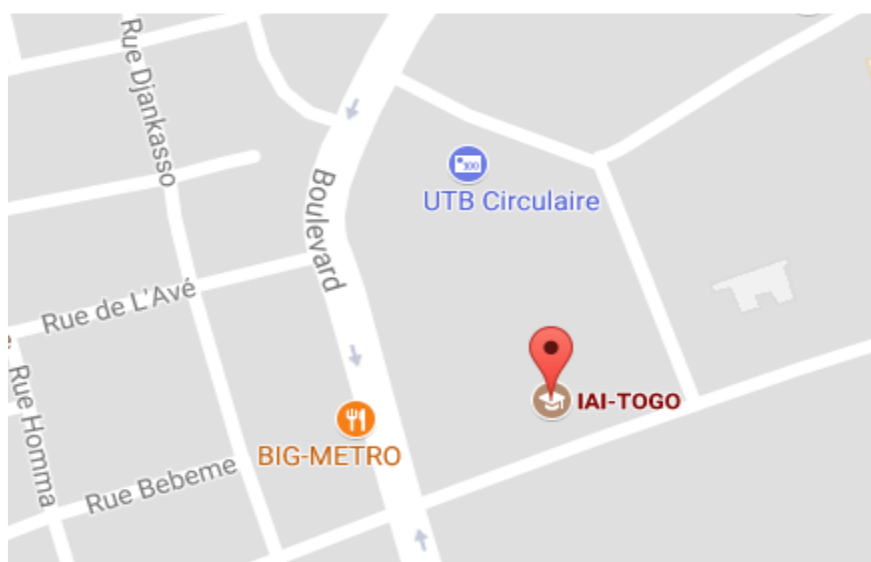
L'Institut Africain d'informatique, représentation du TOGO (IAI-TOGO) propose actuellement les cycles de formation d'Ingénieurs de Travaux Informatiques (Licence professionnelle en informatique), en Génie Logiciel (GL), Système & Réseaux (SR) et Multimédia Technologies Web et Infographie (M-TWI). La durée de ce cycle est de trois (03) ans. Au terme des trois années de formation, les diplômés peuvent poursuivre leurs études supérieures au siège, au GABON ou dans les universités occidentales ou asiatiques (UTBM en FRANCE, Université Laval du Québec au CANADA, etc.), bref dans le monde entier.

L'IAI-TOGO est joignable à l'adresse suivante :

BP : 07 BP 12456 Lomé 07, TOGO

Tél : (+228) 22 21 27 06 / 22 20 47 00

E-mail : [iaitogo@iai-togo.com](mailto:iaitogo@iai-togo.com)



**Figure 1: Localisation géographique de l'IAI-TOGO**

## II-THEME DU STAGE

### 1-Présentation du sujet

De nos jours, le besoin d'eau en sachet devient très régulier pour la plupart des consommateurs. Ainsi donc, les clients potentiels de ces eaux, pour s'en procurer, procèdent

Par commande auprès des sociétés de vente d'eau. Le problème qui se pose est que certains de ces sociétés n'arrivent pas à bien gérer les commandes ainsi que leurs livraisons. Ainsi, pour faciliter la gestion des commandes et des livraisons de ces sociétés, un système de gestion informatique s'avère indispensable, d'où l'intérêt de notre thème : **GESTION INFORMATISEE DES COMMANDES ET DES LIVRAISONS DE SACHETS D'EAU :CAS DE <<SHINE WATER>> .**

### 2-Problématique

Dans toute l'organisation, le temps et la qualité des informations sont des facteurs importants pour un bon rendement. L'entreprise SHINE WATER pour gérer les commandes et les livraisons, procède manuellement, ce qui complique l'exécution de ses tâches lui fait perdre plus de temps.

En vue d'apporter une solution satisfaisante à l'entreprise, la question << Comment améliorer la gestion des commandes et des livraisons des sachets d'eau ? >> retient notre attention et sera un repère tout au long de notre étude.

### **3-Interêt du sujet**

#### **a - Objectifs**

L'objectif principal de ce projet est : la gestion des commandes et des livraisons des sachets d'eau.

#### **b – Résultats attendus**

L'application mise en place doit permettre de :

- Gérer les clients, les produits, les commandes, les livraisons et les livreurs
- Sortir la liste des clients, des commandes, des produits et des livreurs
- Signaler des commandes dont la date de livraison est arrivée
- Voir les statistiques des ventes effectuées au cours d'une période donnée
- Editer une facture aux clients

---

## *PARTIE II : RAPPORT DE PRE-PROGRAMMATION*

---



## **I-ETUDE DE L'EXISTANT**

Elle est consacrée à l'étude de l'état actuel du système de gestion de SHINE WATER. La société SHINE WATER gère ses différentes tâches manuellement. A travers notre collecte d'informations, nous avons compris le fonctionnement du système de gestion des commandes et des livraisons mise en place par la société de production d'eau.

### **1-Processus des commandes**

L'enregistrement d'une commande se fait sur une fiche sur laquelle il est écrit le nom et le prénom du client, la zone dans laquelle il sera livré ainsi que la quantité de sachet qu'il a commandé.

### **2-Processus des livraisons**

Pour les livraisons, chaque jour les livreurs transportent dans des véhicules plusieurs sachets d'eau qu'ils livrent aux clients. Ces livreurs tiennent la fiche sur laquelle ont été enregistrées les commandes. A Chaque livraison, les livreurs cochent devant le nom du client et les produits qui lui ont été livrés.

## **II-CRITIQUE DE L'EXISTANT**

Les moyens d'actions mises en place par la société SHINE WATER sont limités.

- **Commande**

L'historique des commandes est inexistant et donc impossible de faire ressortir les états et les statistiques des commandes passées. La perte des fiches des commandes non livrées entraîne l'oubli de certaines commandes.

- **Client**

Les clients ne sont pas enregistrés, donc impossible d'avoir leur liste et leurs contacts. Il est alors difficile de contacter les clients en cas de changement de programme des livraisons.

- **Livraison**

Il est impossible de retracer toutes les livraisons effectuées par la société, impossible de faire ressortir les états des livraisons et de connaître le livreur qui a livré une commande donnée.

- **Facturation**

- On note l'absence d'un système de facturation prouvant que le client a reçu un certain nombre de sachets d'eau donnés. Ce qui fait que la réclamation concernant certaines livraisons pose problème.

### III-PROPOSITION DE SOLUTIONS

Après analyse et étude du système de gestion de la société SHINE WATER, nous avons opté pour deux (02) solutions spécifiques face aux problèmes précédemment recensés.

#### A-Première solution : ACHAT D'UN LOGICIEL DE GESTION

Cette solution consiste à proposer à SHINE WATER un logiciel déjà fonctionnel qui pourra gérer la clientèle, les commandes et les livraisons. Parmi les différents logiciels, nous avons un logiciel du nom de **Koban** qui répond aux besoins de la société en question.

##### Avantages :

- L'application est déjà développée ce qui inclut une rapidité de la mise en place.
- Le logiciel permet d'effectuer même les tâches que les acteurs du système n'ont pas l'habitude de faire.

##### Inconvénients :

- Ce logiciel ne répond pas à toutes les règles de gestions de SHINE WATER ;
- La personnalisation et le paramétrage sont limités.
- La maintenance deviendra coûteuse et fastidieuse compte tenu de la non proximité de l'éditeur.

#### B-Deuxième solution : LA CONCEPTION D'UNE APPLICATION WEB

Il sera question dans ce cas d'analyser précisément la situation de SHINE WATER ainsi que les problèmes présents et de concevoir une application appropriée.

##### Avantages :

- L'application étant fait sur mesure, nous pourrons ajouter des fonctionnalités au fur et à mesure que les besoins de la société évoluent.
- Elle comportera des fonctionnalités parfaitement adaptées pour répondre aux besoins de la société SHINE WATER.
- Elle sera une application WEB, elle pourra donc être hébergée en interne ou en externe.

- L'application peut être utilisée sans connexion internet et est facilement accessible par un navigateur.
- Il n'y a pas de coût de licence récurrent.

**Inconvénients :**

- L'application n'a pas encore été conçue.
- Le temps de mise en œuvre de l'application est long.

**IV-EVALUATION FINANCIERE DES SOLUTIONS****A-ACHAT D'UN LOGICIEL DE GESTION**

L'achat d'une application de gestion de commandes et des livraisons est un peu coûteux. Ce prix varie en fonction de l'application demandée, des difficultés à réaliser l'application et des et des complexités autour. Cependant l'achat de Koban s'élève à vingt mille (20000) francs par mois soit un montant annuel de (240000) francs.

**B-LA CONCEPTION D'UNE APPLICATION WEB**

Avant de débiter un projet, il est important de disposer des moyens financiers suffisants pour couvrir toutes les activités de ce projet. Pour notre projet, le budget sera scindé en deux (02) parties : le coût matériel c'est-à-dire le coût des équipements nécessaires pour la mise en place en place du nouveau système et le coût de l'application.

**a- Le coût matériel**

Pour la mise en place de notre application, nous aurons besoins de certains éléments qui sont estimés à des coûts suivants :

**Tableau 1:Coût matériel**

<b>LIBELLE</b>	<b>PRIX UNITAIRE</b>	<b>QUANTITE</b>	<b>MONTANT</b>
Serveur de base de données	800.000	01	300.000
Ordinateur	300.000	01	300.000
<b>LIBELLE</b>	<b>PRIX UNITAIRE</b>	<b>QUANTITE</b>	<b>MONTANT</b>
Onduleur	58.385	02	116.770
Imprimante	80.000	01	80.000
<b>TOTAL</b>			<b>796.770</b>

**b- Le coût de la conception**

C'est en fonction du nombre d'heures journaliers passés pour le développement de l'application, du coût horaire et du nombre de développeurs sur le projet que nous avons calculé ce coût. Ainsi donc :

**Tableau 2:Coût de la conception**

<b>FONCTION</b>	<b>PRIX/HEURE</b>	<b>NOMBRE DE DEVELOPPEURS</b>	<b>NOMBRE D'HEURES</b>	<b>COÛT TOTAL (FCFA)</b>
Développeur	4.000	01	360	1.440.400
<b>TOTAL</b>				<b>1.440.000</b>

## C-LE COÛT TOTAL

***Tableau 3:Tableau du coût total***

Critères	Logiciel développé sur mesure
Coût d'acquisition/conception	1.440.000
Coût d'hébergement	18.500
Coût de formation	25.000
Coût de maintenance	20.000
Coût du matériel obligatoire	796.770
Coût total	2.300.270

Après évaluation des différents coûts ; le coût total s'élève à : **2.300.270 FCFA**

## V-CHOIX DE SOLUTION

À la suite de l'étude et l'analyse des contraintes, le choix retenu est le développement et le déploiement d'une application web. Nous étudierons donc dans l'ordre les spécification fonctionnalités suivantes.

**A-SPECIFICATION FONCTIONNELLE**

L'application à développer présente les fonctionnalités suivantes :

**Tableau 4:Spécification fonctionnelle**

<b>PROFILS</b>	<b>DROITS</b>
<b>RESPONSABLE</b>	Gérer les profils Gérer les comptes utilisateurs Gérer les attributions de menus Gérer les attributions de droits Consulter les droits Consulter les menus Consulter le journal des activités Gérer les clients Consulter la liste des livreurs Gérer les livraisons Gérer les commandes Gérer les produits
<b>VENDEUR</b>	Gérer les clients Consulter la liste des livreurs Gérer les livraisons Gérer les commandes Consulter la liste des produits

**B-SPECIFICATION TECHNIQUE****Tableau 5:Spécification technique**

<b>Langage de modélisation</b>	<b>UML (Unified Modeling Language)</b>
<b>Méthode d'analyse</b>	<b>2 Track Unified Process (2TUP)</b>
<b>Outil de modélisation</b>	<b>Sybase PowerAMC 15.1</b>
<b>Système de gestion de base de données(SGBD)</b>	<b>MySQL 5.6.17</b>
<b>Langage de programmation</b>	<b>PHP avec le framework Laravel</b>
<b>Environnement de Développement Intégré (EDI)</b>	<b>Visual Studio Code</b>

**VI- PLANNING PREVISIONNEL DE REALISATION**

Le planning prévisionnel permet de répartir les tâches à accomplir sur la période totale de réalisation afin d'atteindre une meilleure productivité et aussi dans l'optique de respecter le cahier des charges.

Période de réalisation du projet : Du 10/08/2020 au 26/09/2020

Durée totale de réalisation du projet : 7 semaines soit 49 jours

**Tableau 6:Planning prévisionnel de réalisation**

N°	Tâches	Date début	Date fin	Durée en jour
1	-Discussion du thème avec le superviseur -Etude détaillée -Mise en place d'un document provisoire	10/08/2020	14/08/2020	05
2	-Familiarisation avec le Framework laravel -Création de la base de donnée	15/08/2020	20/08/2020	06
3	Phase de programmation et de test	21/08/2020	10/09/2020	21
4	-Tester l'application -Correction, amélioration et finalisation de l'outil	11/09/2020	20/09/2020	10
5	Finalisation du document	21/09/2020	23/09/2020	03
6	Pré-présentation	24/09/2020	25/09/2020	02



## VII-PRESENTATION DE LA METHODE D'ANALYSE

### A-LE LANGAGE DE PROGRAMMATION

#### a-Présentation



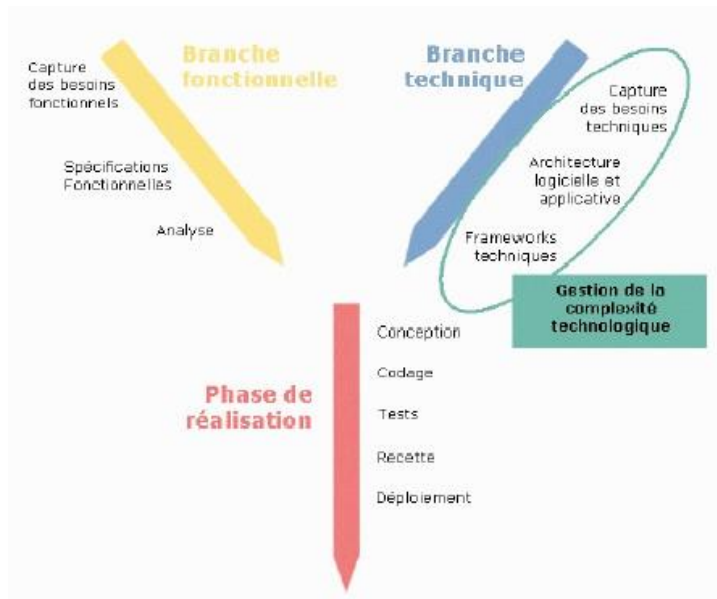
***Figure 1:Logo d'UML***

Le langage de modélisation unifié, en anglais Unified Modeling Language (UML) est un langage de modélisation graphique à base de pictogrammes (dessins figuratifs stylisés ayant fonction de signe) 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. 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).

#### **b-UML est un formalisme**

UML est un langage de conception objet qui permet de modéliser les solutions informatiques par des diagrammes (cités ci-dessus). Pourtant, le langage ne propose pas d'ordre qui doit régir ces diagrammes. C'est ainsi que les méthodes viennent appuyer le langage pour rendre totale et parfaite l'approche objet. Les méthodes imposent au langage l'ordre et la démarche. Il existe une multitude de méthodes de conception objet telles que : 2TUP (Two Track Unified Process), XP (Extreme Programming), Scrum, RUP (Rational Unified Process). La méthode retenue pour notre projet est le 2TUP.

## **B-LE PROCESSUS DE MODELISATION**



**Figure 2: Illustration du 2TUP**

2 Track Unified Process (2TUP), à prononcer "toutiyoupi" est un processus de développement logiciel qui implémente le Processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système.

Ensuite à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de 3 phases essentielles :

- Une branche technique ;
- Une branche fonctionnelle ; ✓ Une phase de réalisation.
- La branche technique (droite)
  - La capture des besoins techniques qui recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (prérequis d'architecture technique) ;

La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects

fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.

- La branche fonctionnelle (gauche)
  - Elle capture des besoins fonctionnels qui produisent le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs ;
  - L'analyse qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.
- La phase de réalisation (milieu)

Elle correspond à :

- Une conception préliminaire qui représente une étape délicate car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;
- La conception détaillée qui étudie comment réaliser chaque composant ;
- L'étape de codage qui produit ses composants et teste au fur et à mesure les unités de code réalisées ;

L'étape de recette, qui consiste à valider les fonctionnalités du système développé.

## VIII-PRESENTATION DE L'OUTIL DE MODELISATION

Dans la modélisation, on pouvait se contenter d'utiliser uniquement UML, mais sur quels supports ? De nos jours, on ne peut pas se contenter des dessins faits sur papier ni au tableau ; c'est là qu'intervient les logiciels de modélisation qui permettent la réutilisation des modèles, leur portabilité (réalisées une fois et utilisables partout), etc.

L'outil retenu pour la modélisation est : Sybase PowerAMC version 15.1.

## A-QU'EST-CE QUE PowerAMC ?



**Figure 3:Logo de PowerAMC**

PowerAMC est un environnement graphique de modélisation d'entreprise très simple d'emploi. Il a été créé par la société SDP sous le nom de AMC\*Designor, racheté par PowerSoft qui lui-même a été racheté par Sybase en 1995. PowerAMC prend en compte plusieurs outils de modélisation tels UML et MERISE.

PowerAMC permet d'effectuer les tâches suivantes :

- Modélisation intégrée via l'utilisation de méthodologies et de notation standards :
  - Données (E/R, Merise) ; - Métiers (BPMN, BPEL, ebXML) ; - Application (UML).
- Génération automatique de code via des Template personnalisables :
  - SQL (avec plus de 50 SGBD) ;
  - Java ;
  - .NET
- Fonctionnalités de reverse engineering pour documenter et mettre à jour des systèmes existants ;
- Présentation de l'outil de modélisation
- Une solution de référentiel d'entreprise avec des fonctionnalités de sécurité et de gestion des versions très complètes pour permettre un développement multiutilisateur
- Fonctionnalités de génération et de gestion de rapports automatisés et personnalisables
- Un environnement extensible, qui vous permet d'ajouter des règles, des commandes, des concepts et des attributs à vos méthodologies de modélisation et de codage.

## **B-MODELISATION AVEC PowerAMC**

PowerAMC fournit un jeu unique d'outils de modélisation professionnels qui associent les techniques et notations standard de la modélisation de processus métiers, de la modélisation des données et de la modélisation des diagrammes UML et d'autres fonctionnalités puissantes afin d'aider à analyser, concevoir, construire et maintenir des applications, en utilisant les techniques les plus élaborées d'ingénierie logicielle.

La solution de modélisation PowerAMC permet d'intégrer étroitement la conception et la maintenance des couches de données centrales de l'application et exigences de projet, processus métiers, code orienté objet, vocabulaires XML et informations de réplication de base de données.

## **IX-ETUDE DETAILLEE DE LA SOLUTION**

Dans cette partie, nous explorerons l'application d'un point de vue interne et externe. Les éléments des spécifications fonctionnelles y seront détaillés afin d'avoir une vision claire et sans ambiguë de l'application.

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système.

### **A-VUE DE MODELISATION**

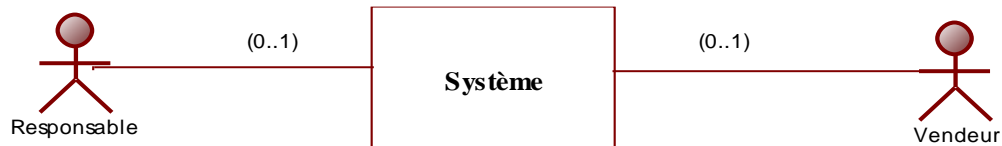
#### **Diagramme de contexte statique**

Le diagramme de contexte statique délimite le domaine d'étude en précisant ce qui est à la charge du système et en identifiant l'environnement extérieur au système étudié avec lequel ce dernier communique. Ces composant sont :

- Un processus unique symbolisant le Système Information étudié
- Les acteurs externes : entités externes au système étudié interagissant avec lui
- Echange entre le système étudié et son environnement

Ce diagramme n'est pas officiellement désigné comme diagramme UML. Il ne fait donc pas partie des diagrammes « officiels », mais il est utile pour la description du domaine d'étude avant d'aborder les aspects plus fins.

Le diagramme de contexte statique de notre système est représenté comme suit :



**Figure 4:Diagramme de contexte statique de notre système**

## B-VUE DE CAS D'UTILISATION

### 1. Le diagramme de cas d'utilisation

#### *a-Définition du cas d'utilisation*

Un diagramme de cas d'utilisation traduit tout ce que l'utilisateur exprime comme action sur le logiciel ou le système à modéliser. C'est une représentation faisant intervenir les acteurs et les cas d'utilisation. Il traduit les besoins des utilisateurs vis-à-vis du système développé.

#### *b-Eléments de modélisation des cas d'utilisation*

- Les acteurs

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie.

Le diagramme des cas d'utilisation d'UML distingue deux acteurs :

- Les acteurs principaux (qui modifient l'état du système ou qui consultent cet état)
- Les acteurs secondaires (acteurs auxquels le système fait appel pour répondre aux sollicitations d'un acteur principal).

Dans notre projet, nous avons décelé uniquement des acteurs principaux qui sont :

- Le responsable
- Le vendeur
- **Le cas d'utilisation**

Un cas d'utilisation exprime le comportement du système en termes d'actions et réactions face à un besoin d'un utilisateur, c'est une fonctionnalité offerte par le système. Le cas d'utilisation (ou use case) correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs. L'ensemble des cas d'utilisation décrit les objectifs (le but) du système.

- **Les relations entre acteur et cas d'utilisation**

Elles expriment l'interaction existant entre un acteur et un cas d'utilisation. Elles lient l'acteur au cas d'utilisation et illustre l'utilisation de la fonctionnalité par l'utilisateur. Elle est représentée par un trait simple.

Le résumé des cas d'utilisations de notre projet est consigné dans le tableau suivant :

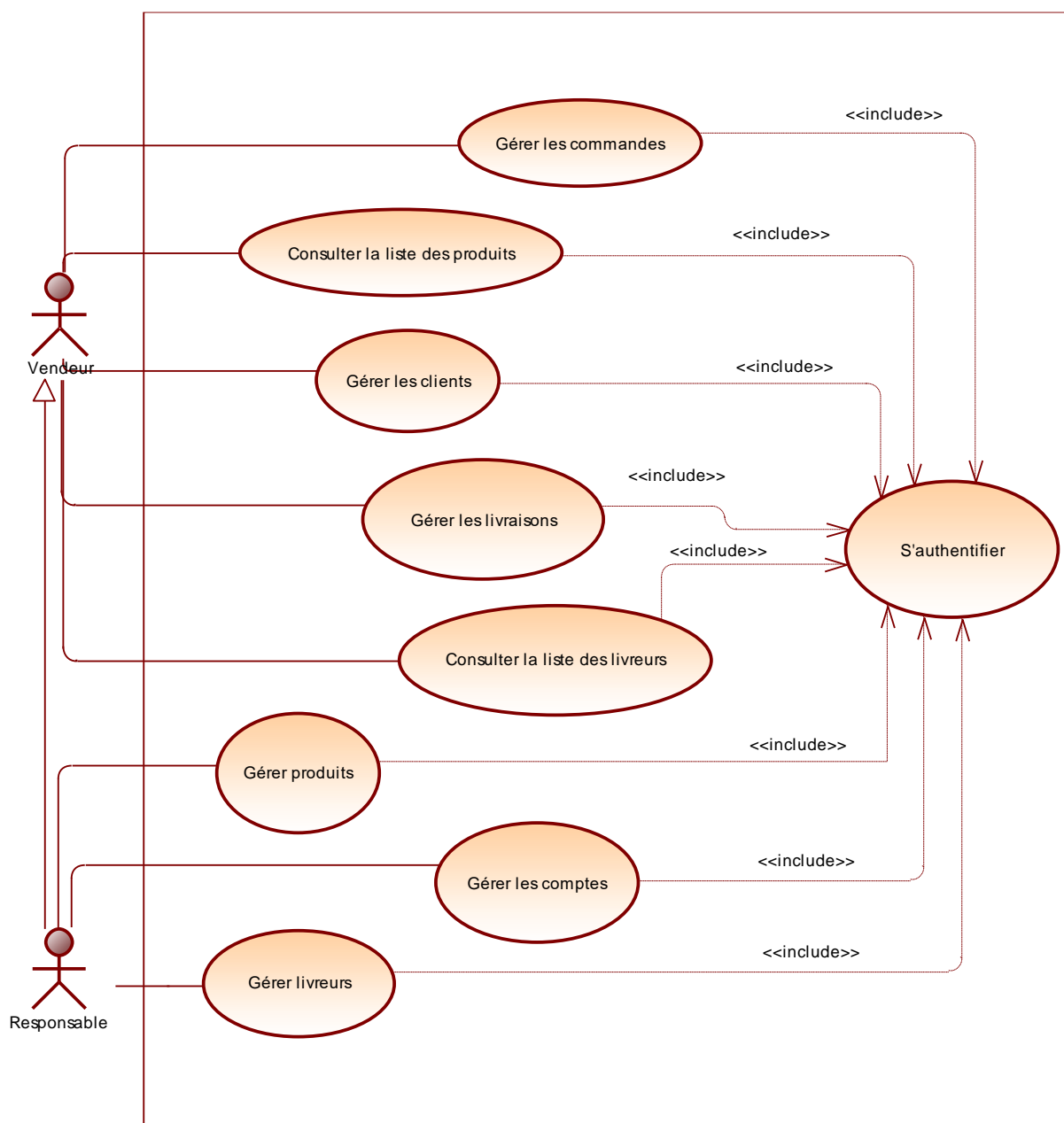
**Tableau 7:Les cas d'utilisations**

Cas d'utilisations		Acteurs
Gérer les comptes	Créer un compte Modifier un compte Désactiver un compte Consulter la liste des utilisateurs	Responsable
Gérer les produits	Enregistrer un produit Modifier un produit	Responsable
	Consulter la liste des produits	Responsable, Vendeur
Gérer les commandes	Enregistrer une commande Modifier une commande Supprimer une commande Consulter la liste des commandes	Responsable, Vendeur
Gérer les livraisons	Enregistrer une livraison Modifier une livraison Supprimer une livraison Consulter la liste des livraisons	Responsable, Vendeur

Cas d'utilisations		Acteurs
Gérer les clients	Enregistrer un client Modifier un client Supprimer un client Consulter la liste des clients	Responsable, Vendeur
Gérer les livreurs	Enregistrer un livreur Modifier un livreur Supprimer un livreur	Responsable
	Consulter la liste des livreurs	Responsable, Vendeur



## 2. Diagramme de cas d'utilisation du système



**Figure 5:Diagramme de cas d'utilisation du système**

## C-VUE DES PROCESSUS

### 1- Description textuelle de quelques cas d'utilisation

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

- Clarifier le déroulement de la fonctionnalité ;
- Décrire la chronologie des actions qui devront être réalisées ;

#### a- S'authentifier

##### ➤ Identification

**Nom du cas** : S'authentifier

**Résumé** : Permet à un utilisateur du système, d'être reconnu par ce dernier en saisissant les paramètres de connexion afin d'y effectuer les actions qui lui sont attribuées.

**Acteurs** : Responsable, Vendeur

**Date de création** : 20/08/2020

**Responsable** : WOLF Daniel

**Version** : 1.0

##### ➤ Séquencement

Le cas d'utilisation commence lorsqu'un utilisateur veut accéder au système.

#### • Pré conditions :

- Le système est fonctionnel
- La page d'authentification est accessible
- Disposer d'un compte

#### • Scénario nominal :

- 1.L'utilisateur lance l'application
- 2.L'utilisateur saisit son nom d'utilisateur et son mot de passe(E1)
- 3.L'utilisateur valide sa saisie
- 4.Le système vérifie le nom de l'utilisateur et le mot de passe (A1)
- 5.Le système affiche la page avec les menus correspondant aux droits de l'utilisateur

- **Scénario alternatif :**

**A1 : Login et/ou mot de passe erroné(s)**

Ce scénario se déclenche au point 4 du scénario nominal. Le système envoie un message d'erreur demandant à l'utilisateur de saisir des informations correctes. Le scénario nominal est repris au point 2.

- **Post conditions :**

- L'utilisateur est authentifié
- Le journal est mis à jour
- Le système est toujours fonctionnel

**b-Enregistrer une commande**
**➤ Identification**

**Nom du cas** : Enregistrer une commande

**Résumé** : Permet à l'utilisateur d'enregistrer une commande concernant un client à une date donnée

**Acteurs** : Responsable, Vendeur

**Date de création** : 20/08/2020

**Responsable** : WOLF

**Version** : 1.0

**➤ Séquencement**

Le cas d'utilisation commence lorsqu'un utilisateur veut passer une commande.

- **Pré conditions :**

- Le système est fonctionnel
- L'utilisateur s'est authentifié
- Le client de la commande doit être enregistré

- **Scénario nominal :**

- 1.L'utilisateur saisit les informations relatives à la commande du client (E1)
- 2.L'utilisateur valide la création
- 3.Le système vérifie les informations (A1)

4. Le système enregistre la commande dans la base

- **Scénario alternatif :**

**A1 : Informations incorrectes**

Ce scénario se déclenche au point 1 du scénario nominal. Le système envoie un message d'erreur spécifiant les informations incorrectes. Le scénario nominal est repris au point 1.

- **Post conditions :**

- La commande est enregistrée avec succès
- Le journal est mis à jour
- Le système est toujours fonctionnel

**c-Consulter la liste des commandes**

➤ **Identification**

**Nom du cas** : Consulter la liste des commandes

**Résumé** : Permet à l'utilisateur de consulter la liste des commandes du système

**Acteurs** : Responsable, Vendeur

**Date de création** : 20/08/2020

**Responsable** : WOLF Daniel

**Version** : 1.0

➤ **Séquencement**

Le cas d'utilisation commence lorsqu'un utilisateur veut consulter la liste des commandes.

- **Pré conditions :**

- Le système est fonctionnel
- L'utilisateur s'est authentifié

- **Scénario nominal :**

1. L'utilisateur demande la liste des commandes
2. Le système recherche toutes les commandes
3. Le système affiche le résultat

- **Post conditions :**

- Le système est toujours fonctionnel

### **f-Enregistrer une livraison**

#### ➤ **Identification**

**Nom du cas** : Enregistrer une livraison

**Résumé** : Permet à l'utilisateur d'enregistrer la date de livraison d'une commande donnée

**Acteurs** : Responsable, Vendeur

**Date de création** : 20/08/2020

**Responsable** : WOLF

**Version** : 1.0

#### ➤ **Séquencement**

Le cas d'utilisation commence lorsqu'un utilisateur veut enregistrer une livraison.

- **Pré conditions :**

- Le système est fonctionnel
  - L'utilisateur s'est authentifié
  - La commande existe déjà dans le système

- **Scénario nominal :**

- 1.L'utilisateur saisit les informations relatives à la livraison (E1)
- 2.L'utilisateur valide la création
- 3.Le système vérifie les informations (A1)
- 4.Le système enregistre la livraison dans la base
5. Le système renvoie le message <<**Livraison enregistré**>>

- **Scénario alternatif :**

#### **A1 : Informations incorrectes**

Ce scénario se déclenche au point 1 du scénario nominal. Le système envoie un message d'erreur spécifiant les informations incorrectes. Le scénario nominal est repris au point 1.

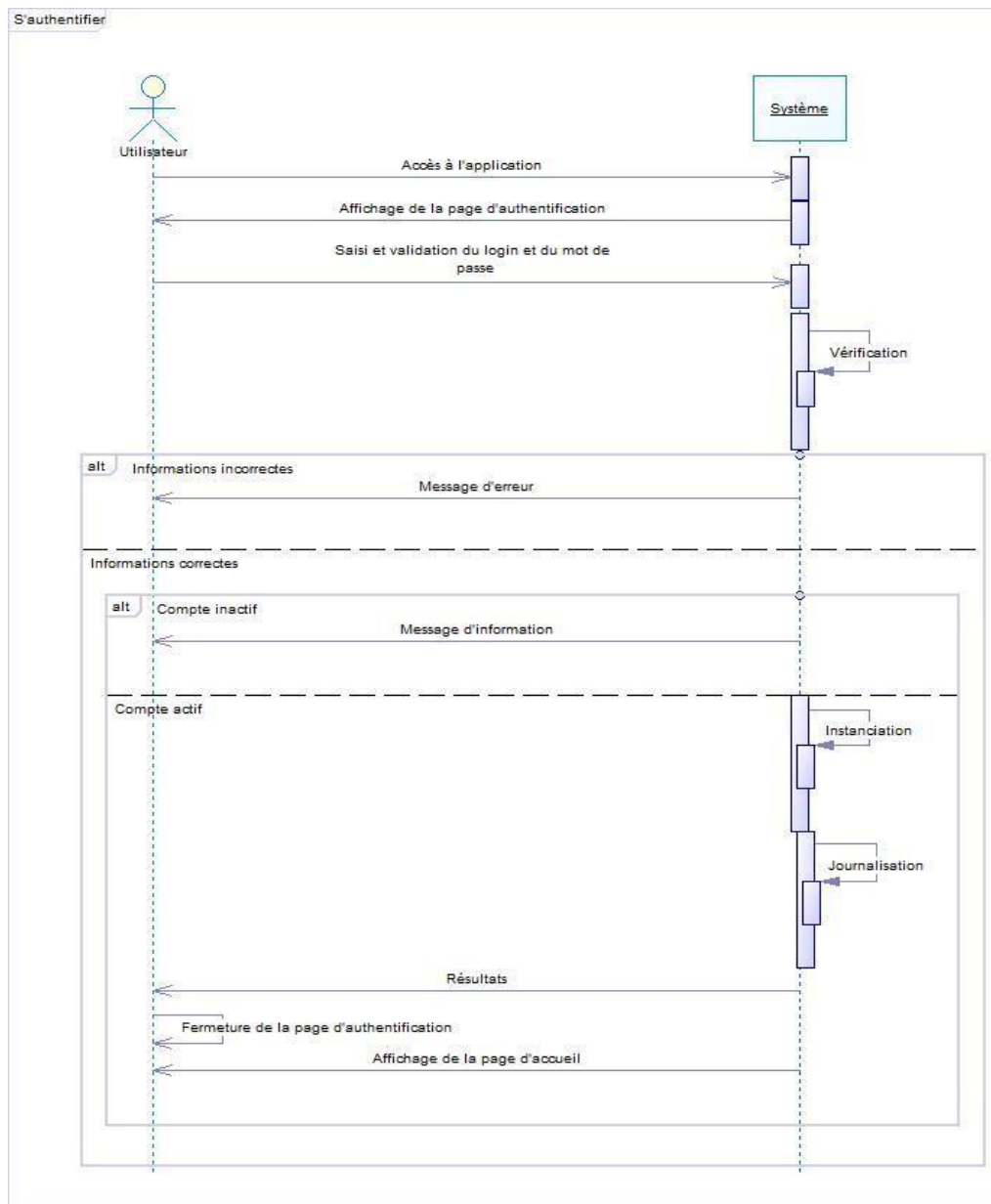
- **Post conditions :**

- La livraison est enregistrée avec succès

- Le journal est mis à jour
- Le système est toujours fonctionnel

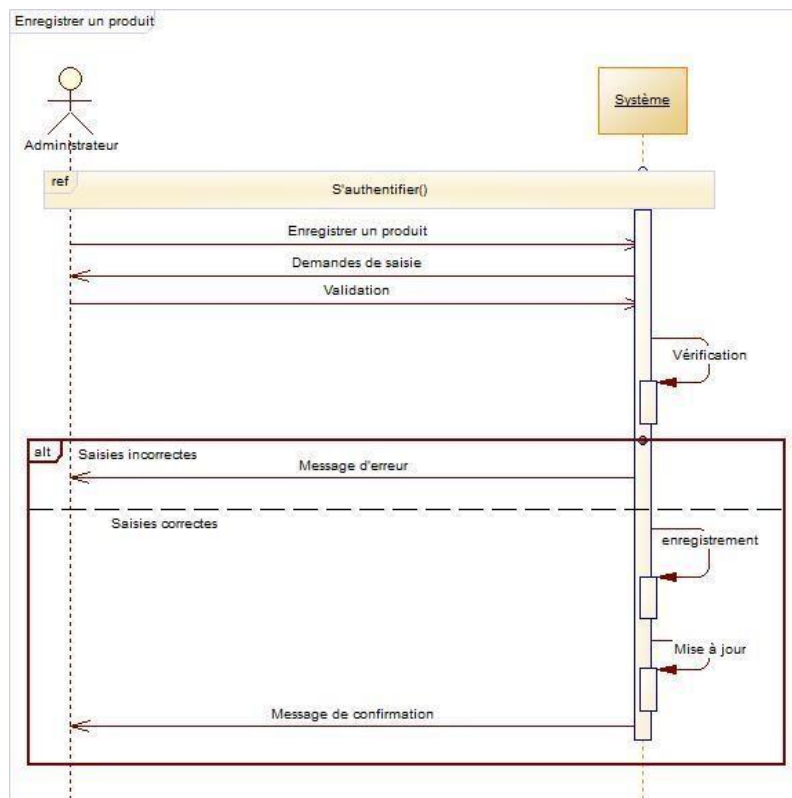
## 2- Quelques diagrammes de séquence

### ➤ S'authentifier



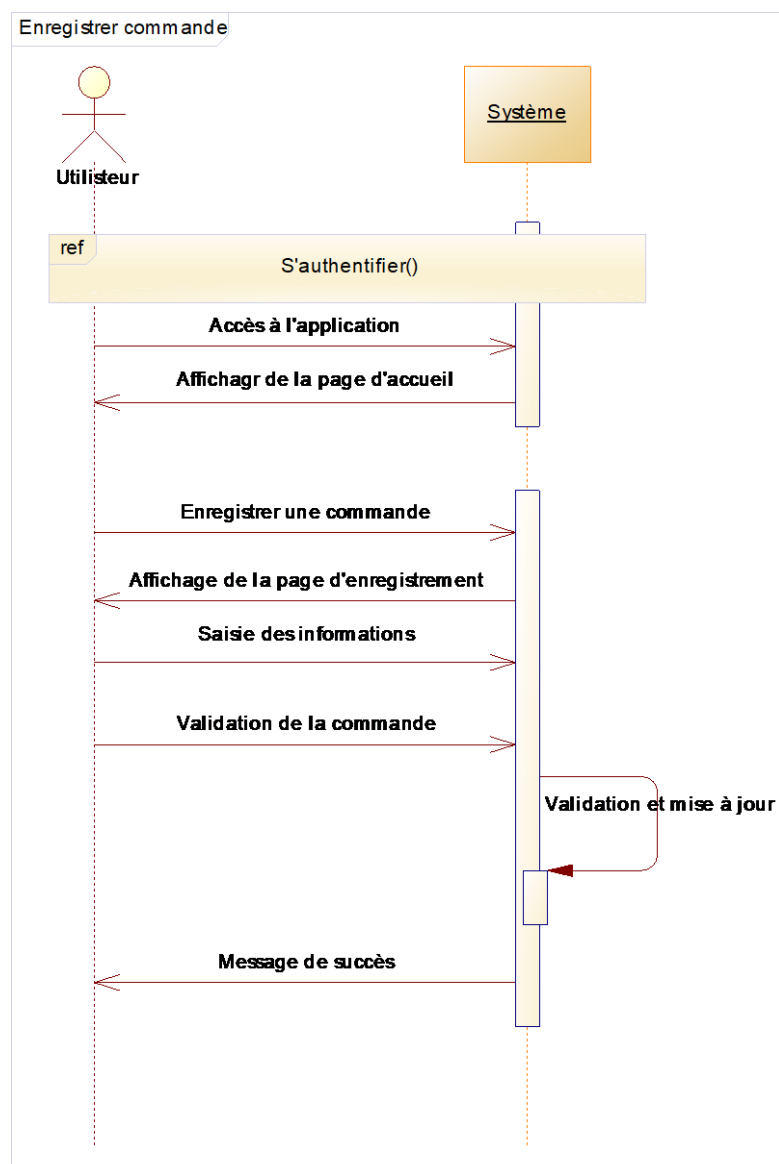
**Figure 6:Diagramme de séquence du cas d'utilisation « s'authentifier »**

## ➤ Enregistrer produit



***Figure 7: Diagramme de séquence du cas d'utilisation « enregistrer un produit »***

## ➤ Enregistrer commande



**Figure 8:Diagramme de séquence du cas d'utilisation « enregistrer une commande »**



## ➤ Lister commandes



**Figure 9:Diagramme de séquence du cas d'utilisation « Consulter la liste des commandes»**

## 3.Diagramme d'activité

### a. Généralités

Le diagramme d'activités se présente comme un organisme qui décrit les différents enchainements ou étapes du déroulement d'un cas d'utilisation ou groupe de cas d'utilisation.

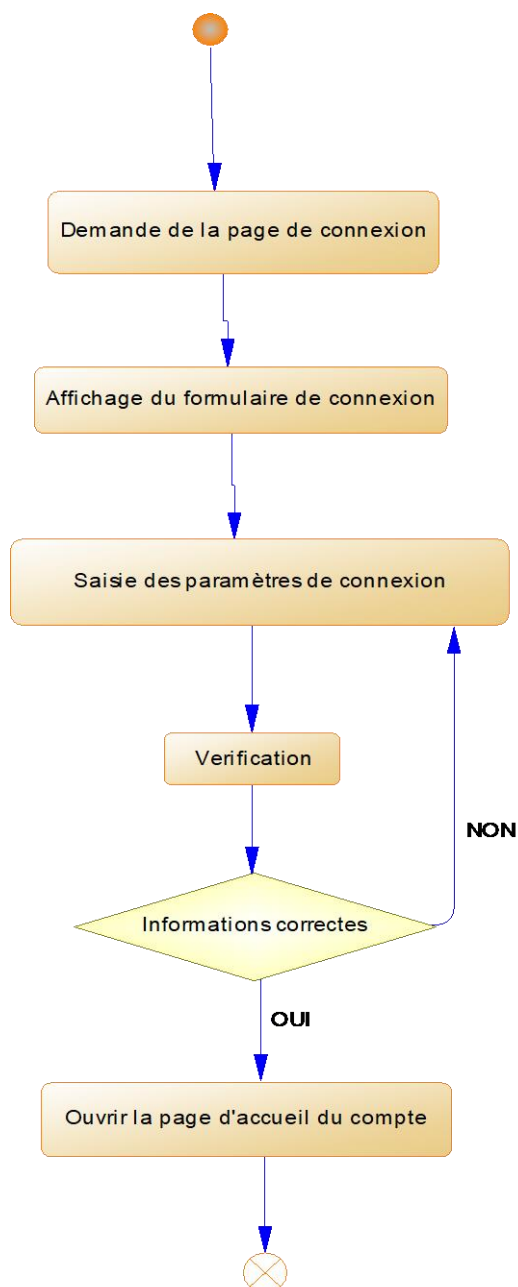
Le diagramme d'activité est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis. Certains des composants les plus courants d'un diagramme d'activités sont :

- **Actions** : étape dans l'activité où les utilisateurs ou le logiciel exécutent une tâche donnée.
- **Nœud de décision** : embranchement conditionnel dans le flux qui est représenté par un losange. Il comporte une seule entrée et deux sorties ou plus.
- **Flux de contrôle** : autre nom des connecteurs qui illustrent le flux entre les étapes du diagramme.
- **Nœud de départ** : symbolise le début de l'activité. Cet élément est représenté par un cercle noir.
- **Nœud de fin** : représente l'étape finale de l'activité. Il est représenté par un cercle noir avec un contour.

#### b. Quelques diagrammes d'activités de notre système

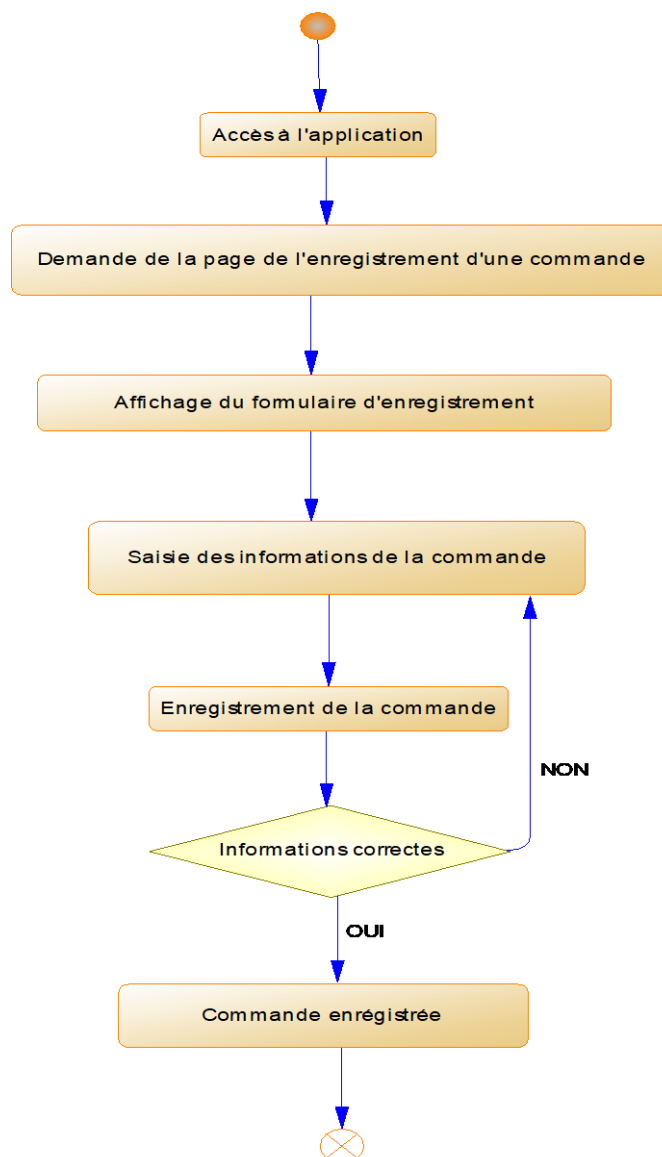
Le diagramme d'activités présente uniquement les actions du système. Voici quelques diagrammes d'activité de notre système :

➤ **S'authentifier**



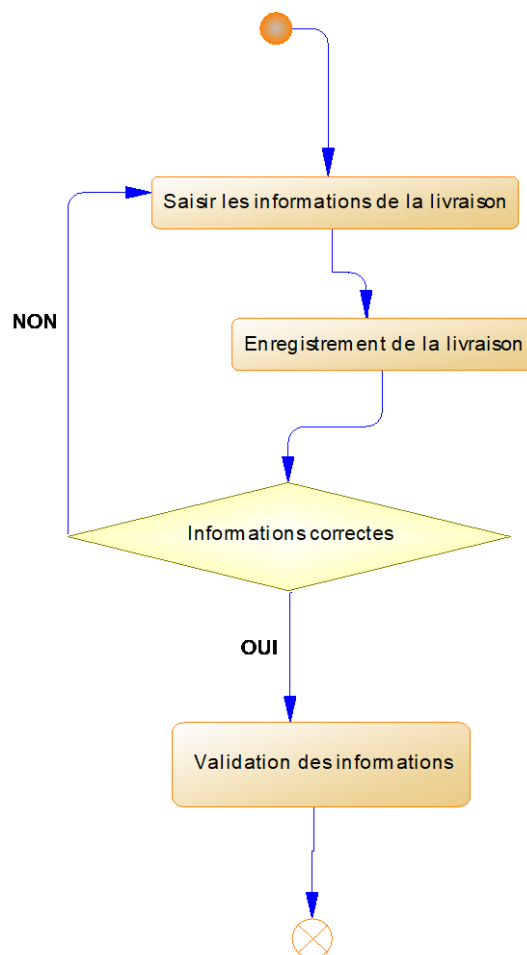
***Figure 10: Diagramme d'activité du cas d'utilisation « s'authentifier »***

➤ Enregistrer commande



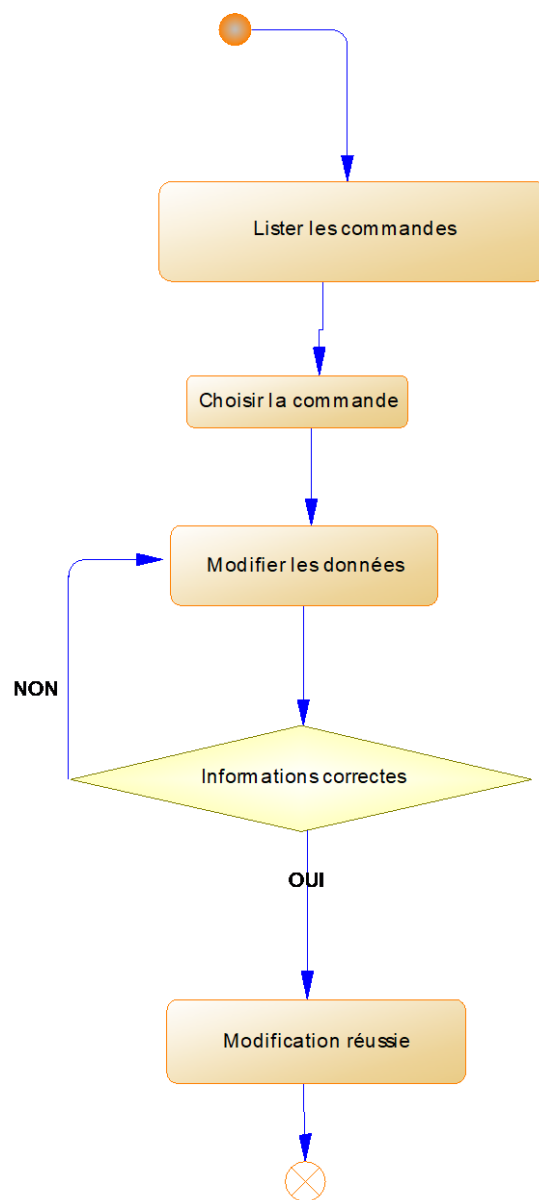
***Figure 11: Diagramme d'activité du cas d'utilisation « Enregistrer commande »***

➤ **Enregistrer livraison**



***Figure 12:Diagramme d'activité du cas d'utilisation « Enregistrer livraison»***

➤ **Modifier commande**



***Figure 13:Diagramme d'activité du cas d'utilisation « Modifier Commande »***

## D-VUE LOGIQUE

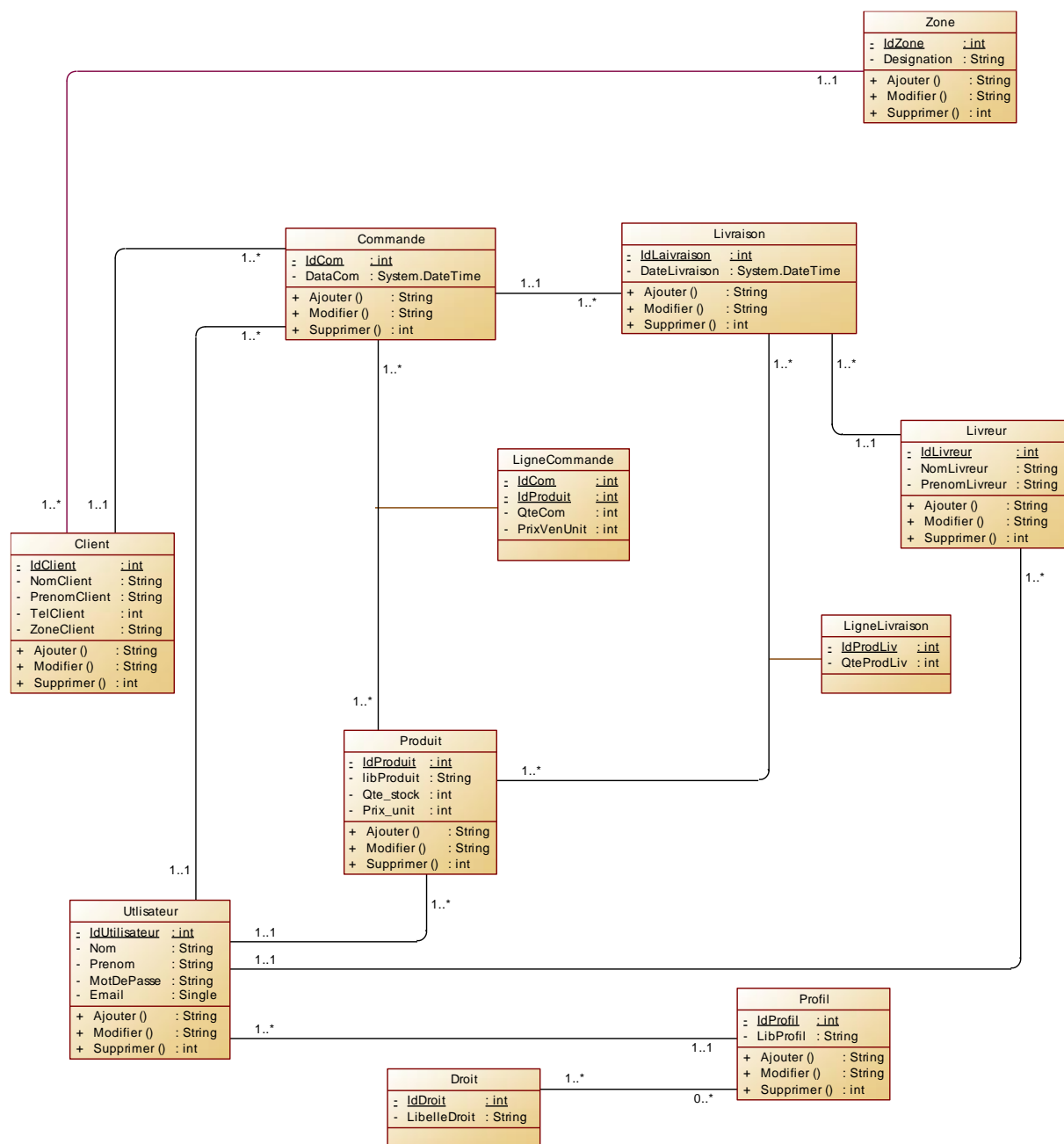
### Diagramme de classe-entité

#### *a. Généralités*

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Le diagramme des classes comporte les concepts suivants :

- Classe
- Attribut
- Identifiant
- Opération
- Relation
- Généralisation

*b. Diagramme des classes de notre système*



**Figure 14:Diagramme de classe du système**



## E-VUE DE DEPLOIEMENT

Cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le où). Le pourquoi, n'est pas défini dans UML.

### *Diagramme de déploiement*

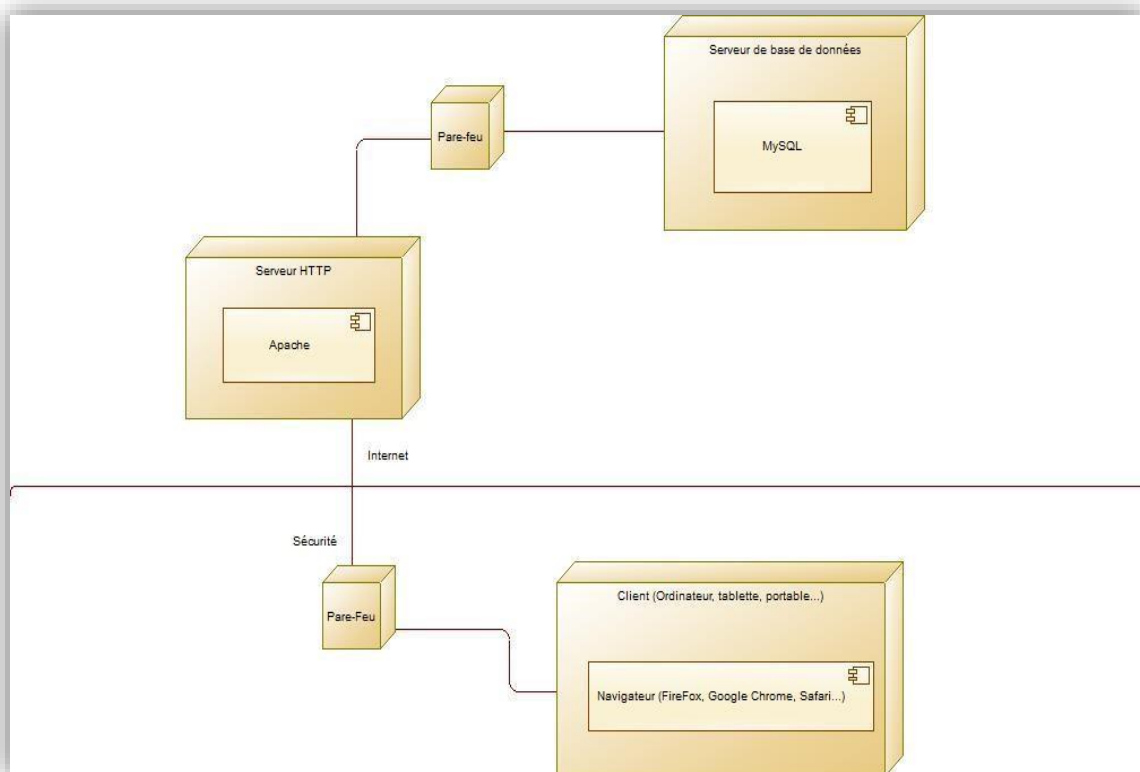
Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.

- Les ressources matérielles sont représentées sous forme de nœuds
- Les nœuds sont connectés entre eux, à l'aide d'un support de communication -  
La nature des lignes de communication et leurs caractéristiques peuvent être

Précisées

- Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.
- Les diagrammes de déploiement correspondent à la vue de déploiement d'une architecture logicielle.

Le diagramme de déploiement de notre système se présente comme suit :



**Figure 15:Diagramme de déploiement du système**

---

## *PARTIE III : REALISATION ET MISE EN OEUVRE*

---

## I-MISE EN OEUVRE

La mise en œuvre constitue la réalisation effective de l'application. Faisant intervenir divers outils, elle comprend donc le choix du matériel, des logiciels utilisés et la sécurité de l'application.

### 1-CHOIX MATERIELS

Un matériel nous a permis de développer notre solution informatique : il s'agit d'un ordinateur portable. En voici les caractéristiques :

***Tableau 8:Tableau du matériel utilisé***

Marque	Model	Ecran	Processeur	Mémoire RAM	Disque dur	Système D'exploitation
HP	EliteBook 4530s	15.6 "	Intel core-i5- 2330M	8Go	500Go	Windows 10 Professionnel

### 2-CHOIX LOGICIELS

Nous avons choisi Sublime Text comme IDE (Environnement de Développement Intégré) pour la réalisation de cette application, XAMP et Apache comme serveur Web et Chrome comme navigateur.

a-PHP



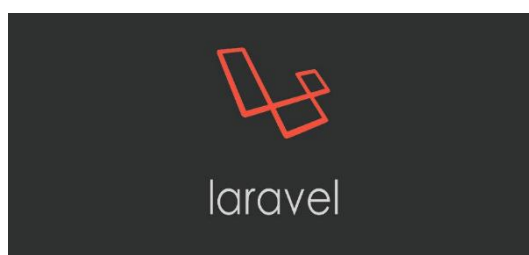
***Figure 16:Logo PHP***

PHP (HyperText Preprocessor), plus connu sous son sigle PHP (acronyme récursif), est un langage de génération de script Open Source, spécialement conçu pour le développement

d'applications web. Il peut être intégré facilement au HTML. Il est utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais fonctionne également comme n'importe quel langage interprété de façon locale.

PHP est un langage impératif orienté objet et dit server-side c'est-à-dire que le script est exécuté sur le serveur qui envoie le résultat html au navigateur client. Il dispose d'une bibliothèque riche et d'un modèle objet de plus en plus complet. Sa syntaxe est empruntée aux langages C, Java et Perl, et est facile à apprendre.

b-Framework Laravel



**Figure 17:Logo Laravel**

Nous utiliserons du PHP couplé à un Framework. Un Framework est un ensemble de composants éprouvés et réutilisables. Utiliser un Framework permet non seulement l'évolution rapide dans le développement d'une application mais garantie aussi la sécurité du code. CakePHP, CodeIgniter, Symfony, Zend Framework sont des framework PHP que nous pouvons utiliser mais notre choix s'est porté sur Laravel car :

- Laravel est un framework bien documenté ;
- Il existe une énorme communauté derrière ce framework ;
- Il est un framework simple d'utilisation ;
- Facilité de trouver des extensions ;
- Laravel offre un système d'authentification pour les connexions ;
- Il intègre un créateur de requêtes SQL et un ORM performants (éloquent) ;
- Il propose un système de routage performant ainsi qu'un système d'envoi de mail ;
- Laravel intègre aussi un système de migration pour les bases de données, un système de cache, une gestion des sessions.

## c-Sublime Text



**Figure 18:Logo de Sublime Text**

Sublime Text est un éditeur de texte générique conçu pour prendre en charge plusieurs langages de programmation allant du langage C à l'action du script en passant par les langages PHP, objective-Cou encore Ocalm voire même le SQL et est disponible sur Windows, MAC et Linux. Il fut créé en 2007 par Jon Skinner. Il intègre la plupart des fonctionnalités de base d'un éditeur de texte et propose un support des macros et des plugins Python.

## b-XAMPP



**Figure 19:Logo de XAMPP**

**XAMPP** est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels

libres (**X** (cross) **A**pache **M**ariaDB **P**erl **P**HP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

L'**utilisation** de **XAMPP** sert à tester les sites de vos clients ou votre propre site Web avant de le télécharger sur le serveur Web distant. Ce logiciel serveur **XAMPP** vous donne l'environnement approprié pour tester des projets MySQL, PHP, Apache et Perl sur l'ordinateur local.

c-APACHE



**Figure 20:Logo d'Apache**

Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows. Apache (prononcer à la française ou bien pour les puristes à l'anglophone << Apatchy >>) tire son nom de la façon dont il a été mis au point (<<A patchy server>> traduisez <<un serveur rafistolé>>) car il est le fruit d'une multitude de correctifs logiciels afin d'en faire une solution très sûre.

En effet Apache est considéré comme sûr dans la mesure où peu de vulnérabilités le concernant sont publiées.

Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, Common

Gateway Interface, Server Side Includes, réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc. Néanmoins, il est à noter que l'existence de nombreux modules Apache complexifie la configuration du serveur Web.

## d-CHROME



**Figure 21:Logo de Chrome**

Chrome a été conçu à partir de 25 bibliothèques de code différentes. Ces dernières proviennent de Google ou de tiers, telles les bibliothèques « Netscape Portable Runtime (en) », « Network Security Services (en) », « NPAPI (en) » de Mozilla, ainsi que la base de données SQLite et de nombreux autres projets *open source*<sup>51</sup>. La machine virtuelle JavaScript a été considérée comme un projet suffisamment important pour que son développement et sa gestion soient assurés par une équipe distincte coordonnée par Lars Bak à Aarhus au Danemark.

### 3-Sécurité de l'application

#### a. Sécurité au niveau de l'hébergeur de l'application web

Depuis plus de quinze ans, OVH héberge les données et porte une attention particulière à leur protection. Pour garantir les plus hauts niveaux de sécurité, OVH a pris soin de mettre en place des dispositifs efficaces ainsi qu'un arsenal de bonnes pratiques à tous les niveaux de leur organisation et de leurs infrastructures :

- Des datacenters placés sous haute protection
- Gestion des risques d'incendie
- Sécurité côté serveur
- Des centres de données géographiquement distants

- Protection anti-D DOS
- Double authentification
- Confidentialité, disponibilité, intégrité des informations

#### b. Sécurité au niveau de l'application

Vu les expositions des applications Web aux risques, un certain nombre de règles ont été instaurées dans le processus de développement pour sécuriser la plateforme :

- **Une page d'authentification** : première page de l'application permettant aux utilisateurs de se connecter à partir des paramètres de connexions (login et mot de passe) ;
- **Compte utilisateur** : chaque client possède un seul compte utilisateur. Un compte peut être à l'état « actif » ou « inactif » (précaution prise pour bloquer les comptes en cas de piratage par exemple).
- **Journalisation des opérations** : toute action ou opération effectuée par un utilisateur est journalisé (enregistrées dans un journal portant les informations suivantes : date de l'événement, type d'événement, libellé de l'action, etc.), histoire de les retracer.

## II. PRESENTATION DE L'APPLICATION

Cette partie présente l'application d'un point de vue graphique (les Interfaces Hommes Machines (IHM)) et logique (le script de création de la base de données).

### 1-Présentation

L'application que nous avons développée est nommée « **Shine Water** ». Cette application est un module web écrit dans le langage PHP avec le Framework Laravel, permettant d'effectuer les commandes et livraisons des sachets d'eau.

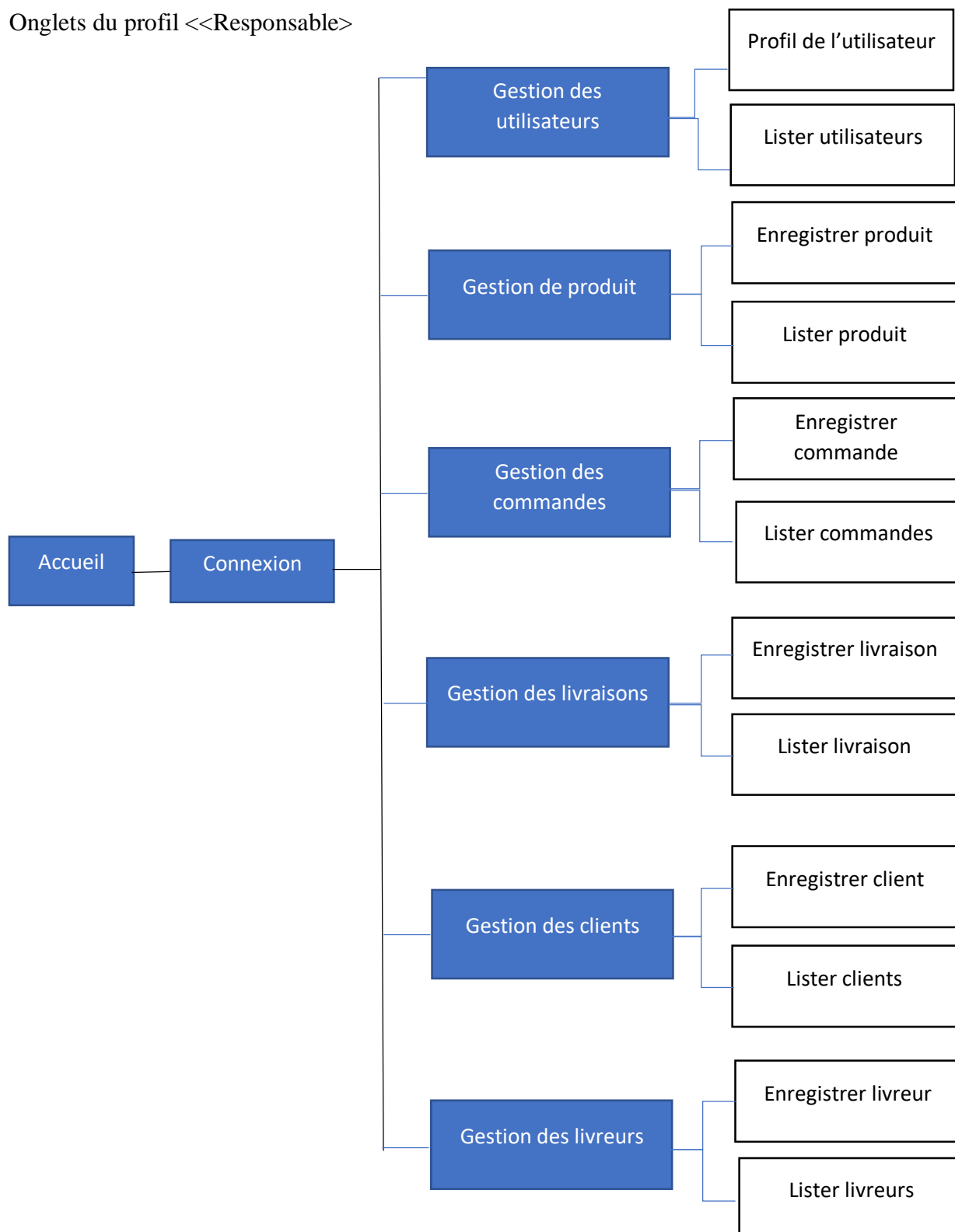
Elle offre des interfaces simples et facilement manipulables pour les commandes et les livraisons. Elle utilise une architecture Client-Serveur deux-tiers et sera donc installée sur un serveur distant et les postes client y accéderont avec une URL dans un navigateur.

### 2-Architecture de l'application

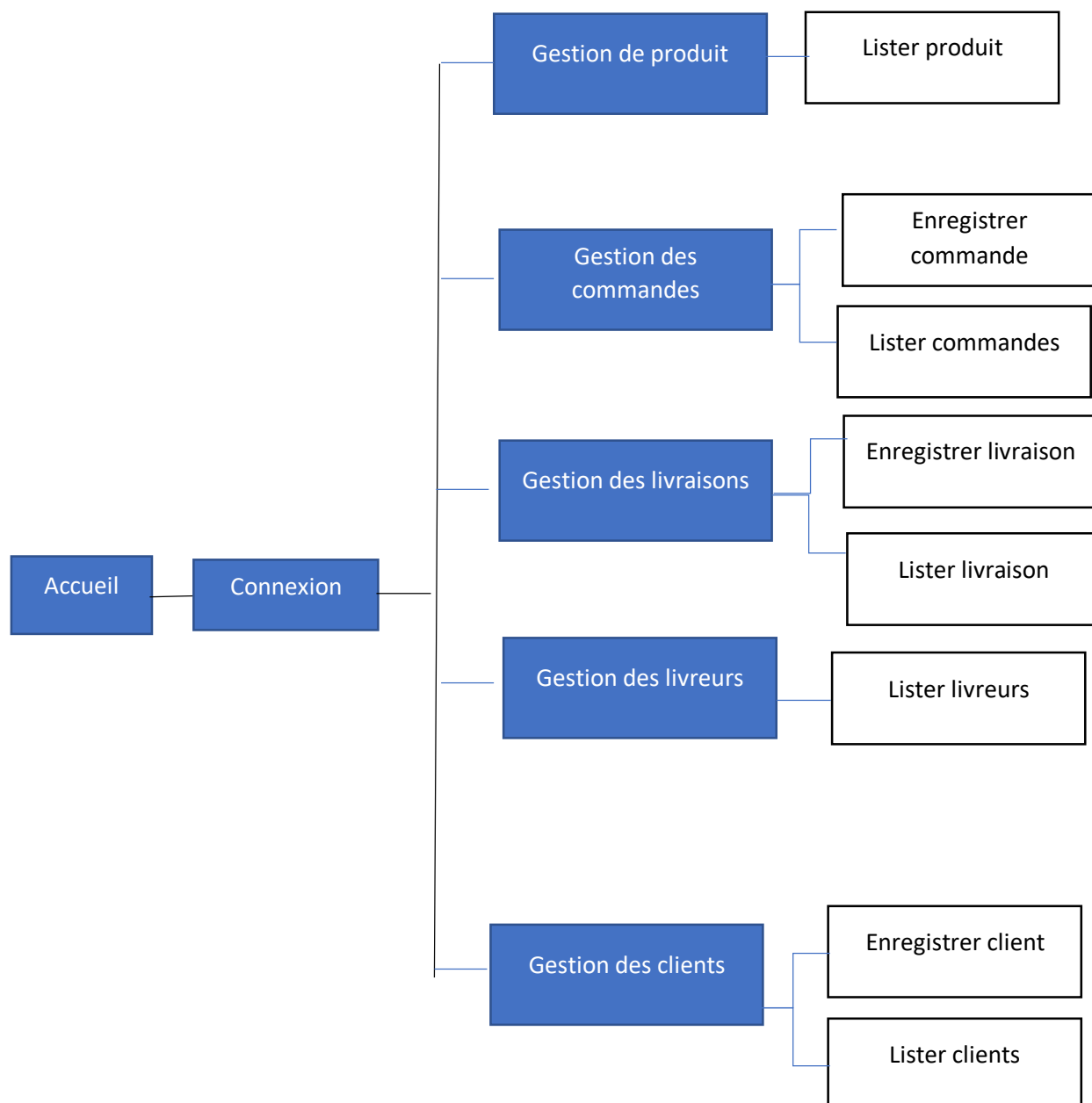
Les menus de l'application sont accessibles en fonction du profil utilisateur. Nous présentons la structure de quelques onglets en fonction des profils.



Onglets du profil <<Responsable>



Onglets du profil <<Vendeur>>



**3-Script de création de la base de données**

```
CREATE DATABASE IF NOT EXISTS `laravel` DEFAULT CHARACTER SET latin1
COLLATE latin1_swedish_ci;
```

```
USE `laravel`;
```

```
-- Structure de la table `clients`
```

```
CREATE TABLE `clients` (
```

```
  `id` bigint(20) UNSIGNED NOT NULL,
```

```
  `NomClient` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `PrenomClient` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `TelClient` int(11) NOT NULL,
```

```
  `AdresseClient` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `created_at` timestamp NULL DEFAULT NULL,
```

```
  `updated_at` timestamp NULL DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
--
```

```
-- Structure de la table `commandes`
```

```
CREATE TABLE `commandes` (
```

```
  `id` bigint(20) UNSIGNED NOT NULL,
```

```
  `NumeroCommande` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `DateCommande` date NOT NULL,
```

```
  `DatePourLivrer` date NOT NULL,
```

```
  `user_id` bigint(20) UNSIGNED NOT NULL,
```

```

`client_id` bigint(20) UNSIGNED NOT NULL,

`created_at` timestamp NULL DEFAULT NULL,

`updated_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--

-- Structure de la table `commande_produits`

CREATE TABLE `commande_produits` (

  `id` bigint(20) UNSIGNED NOT NULL,

  `commande_id` bigint(20) UNSIGNED NOT NULL,

  `produit_id` bigint(20) UNSIGNED NOT NULL,

  `QteCom` int(11) NOT NULL,

  `PrixVenUnit` int(11) NOT NULL,

  `created_at` timestamp NULL DEFAULT NULL,

  `updated_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--

-- Structure de la table `failed_jobs`

CREATE TABLE `failed_jobs` (

  `id` bigint(20) UNSIGNED NOT NULL,

  `connection` text COLLATE utf8_unicode_ci NOT NULL,

  `queue` text COLLATE utf8_unicode_ci NOT NULL,

  `payload` longtext COLLATE utf8_unicode_ci NOT NULL,

  `exception` longtext COLLATE utf8_unicode_ci NOT NULL,

  `failed_at` timestamp NOT NULL DEFAULT current_timestamp()

```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
--
```

```
-- Structure de la table `livraisons`
```

```
CREATE TABLE `livraisons` (
```

```
  `id` bigint(20) UNSIGNED NOT NULL,
```

```
  `NumeroLivraison` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```
  `DateLivraison` date NOT NULL,
```

```
  `user_id` bigint(20) UNSIGNED NOT NULL,
```

```
  `livreur_id` bigint(20) UNSIGNED NOT NULL,
```

```
  `commande_id` bigint(20) UNSIGNED NOT NULL,
```

```
  `created_at` timestamp NULL DEFAULT NULL,
```

```
  `updated_at` timestamp NULL DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
--
```

```
-- Structure de la table `livraison_produits`
```

```
CREATE TABLE `livraison_produits` (
```

```
  `id` bigint(20) UNSIGNED NOT NULL,
```

```
  `livraison_id` bigint(20) UNSIGNED NOT NULL,
```

```
  `produit_id` bigint(20) UNSIGNED NOT NULL,
```

```
  `QteProLiv` int(11) NOT NULL,
```

```
  `created_at` timestamp NULL DEFAULT NULL,
```

```
  `updated_at` timestamp NULL DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
--
```

-- Structure de la table `livreurs`

```
CREATE TABLE `livreurs` (  
  `id` bigint(20) UNSIGNED NOT NULL,  
  `NomLivreur` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `PrenomLivreur` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `TelLivreur` int(11) NOT NULL,  
  `AdresseLivreur` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `user_id` bigint(20) UNSIGNED NOT NULL,  
  `created_at` timestamp NULL DEFAULT NULL,  
  `updated_at` timestamp NULL DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

--

-- Structure de la table `migrations`

```
CREATE TABLE `migrations` (  
  `id` int(10) UNSIGNED NOT NULL,  
  `migration` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `batch` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

--

-- Structure de la table `password\_resets`

--

```
CREATE TABLE `password_resets` (  
  `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `token` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
```

```

`created_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--

-- Structure de la table `produits`

CREATE TABLE `produits` (

  `id` bigint(20) UNSIGNED NOT NULL,

  `LibelleProduit` varchar(255) COLLATE utf8_unicode_ci NOT NULL,

  `Quantite` int(11) NOT NULL,

  `PrixUnit` int(11) NOT NULL,

  `user_id` bigint(20) UNSIGNED NOT NULL,

  `created_at` timestamp NULL DEFAULT NULL,

  `updated_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--

-- Structure de la table `profils`

CREATE TABLE `profils` (

  `IdProfil` int(10) UNSIGNED NOT NULL,

  `LibelleProfil` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,

  `created_at` timestamp NULL DEFAULT NULL,

  `updated_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

--

-- Structure de la table `users`

CREATE TABLE `users` (

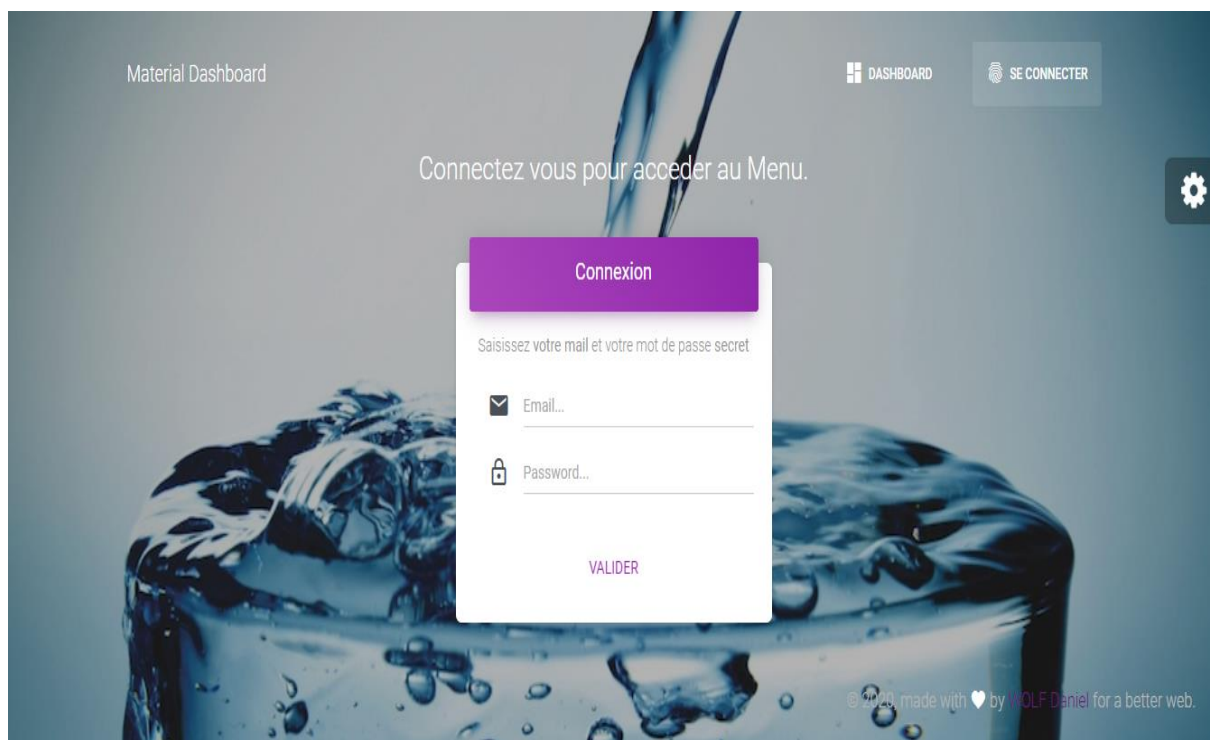
```

```
`id` bigint(20) UNSIGNED NOT NULL,  
`name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
`email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
`email_verified_at` timestamp NULL DEFAULT NULL,  
`password` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
`remember_token` varchar(100) COLLATE utf8_unicode_ci DEFAULT NULL,  
`created_at` timestamp NULL DEFAULT NULL,  
`updated_at` timestamp NULL DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```



## 4-Quelques masques de saisies et de codes sources de l'application

### a. Page de connexion



**Figure 22:page d'authentification**

#### ➤ Code source de la page

```
@extends('layouts.app', ['class' => 'off-canvas-
sidebar', 'activePage' => 'login', 'title' => __('Material Dashboard')])

@section('content')
<div class="container" style="height: auto;">
  <div class="row align-items-center">
    <div class="col-md-9 ml-auto mr-auto mb-3 text-center">
      <h3>{{ __('Connectez vous pour acceder au Menu.') }} </h3>
    </div>
    <div class="col-lg-4 col-md-6 col-sm-8 ml-auto mr-auto">
```

```

<form class="form" method="POST" action="{{ route('login') }}">
    @csrf

    <div class="card card-login card-hidden mb-3">
        <div class="card-header card-header-primary text-center">
            <h4 class="card-title"><strong>{{ __('Connexion') }}</strong></h4>

        </div>
        <div class="card-body">
            <p class="card-description text-center">{{ __('Saisissez ') }} <strong>votre mail</strong> {{ __(' et votre mot de passe ') }}<strong>secret</strong> </p>
            <div class="bmd-form-group{{ $errors->has('email') ? ' has-danger' : '' }}">
                <div class="input-group">
                    <div class="input-group-prepend">
                        <span class="input-group-text">
                            <i class="material-icons">email</i>
                        </span>
                    </div>
                    <input type="email" name="email" class="form-control" placeholder="{{ __('Email...') }}" value="{{ old('email', '') }}" required>
                </div>
                @if ($errors->has('email'))
                    <div id="email-error" class="error text-danger pl-3" for="email" style="display: block;">
                        <strong>{{ $errors->first('email') }}</strong>
                    </div>
                @endif
            </div>
            <div class="bmd-form-group{{ $errors->has('password') ? ' has-danger' : '' }} mt-3">
                <div class="input-group">
                    <div class="input-group-prepend">
                        <span class="input-group-text">

```

```

        <i class="material-icons">lock_outline</i>
    </span>
</div>

    <input type="password" name="password" id="password" class="form-
control" placeholder="{{ __('Password...') }}" value="{{ !$errors-
>has('password') ? "secret" : "" }}" required>

    </div>

    @if ($errors->has('password'))
        <div id="password-error" class="error text-danger pl-
3" for="password" style="display: block;">
            <strong>{{ $errors->first('password') }}</strong>
        </div>
    @endif
</div>

</div>

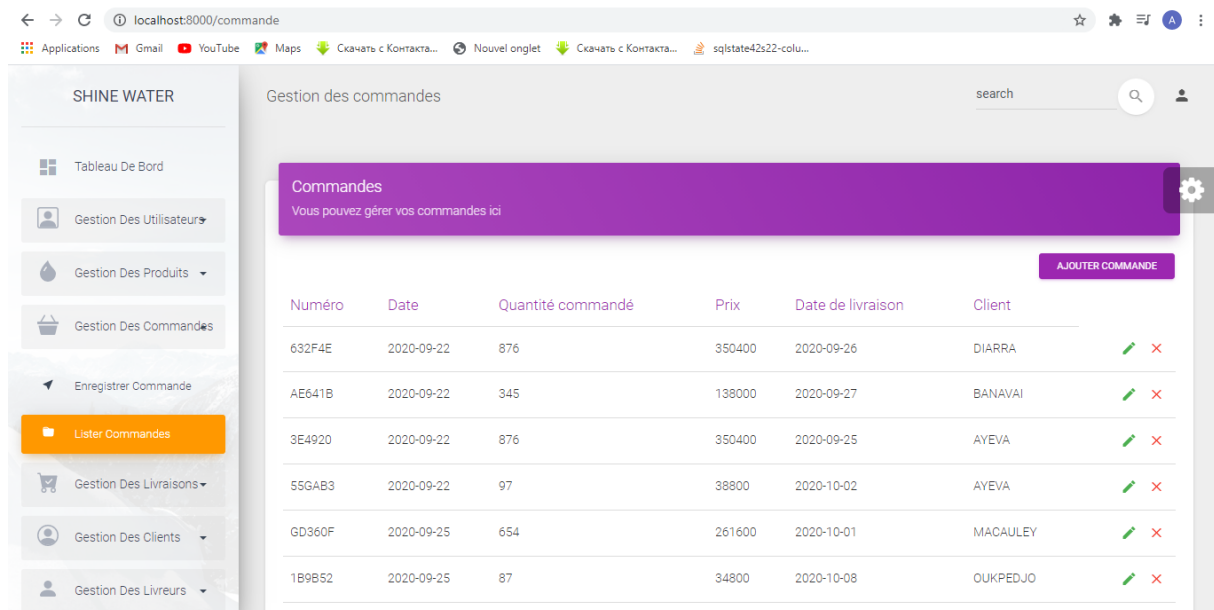
<div class="card-footer justify-content-center">
    <button type="submit" class="btn btn-primary btn-link btn-
lg">{{ __('Valider') }}</button>
</div>
</div>
</form>

</div>
</div>
</div>
@endsection

```

## b. Gestion des commandes

## • Page de la liste des commandes

**Figure 23:page de la liste des commandes**

## ➤ Code source de la liste des commandes

```
@extends('layouts.app', ['activePage' => 'user-
management', 'titlePage' => __("Gestion des commandes")])

@section('content')
<div class="content">
  <div class="container-fluid">
    <div class="row">
      <div class="col-md-12">
        <div class="card">
          <div class="card-header card-header-primary">
            <h4 class="card-title ">{{ __('Commandes') }}</h4>
            <p class="card-
category"> {{ __('Vous pouvez gérer vos commandes ici') }}</p>
          </div>
          <div class="card-body">
```

```

@if (session('status'))
  <div class="row">
    <div class="col-sm-12">
      <div class="alert alert-success">
        <button type="button" class="close" data-dismiss="alert" aria-
label="Close">

          <i class="material-icons">close</i>
        </button>
        <span>{{ session('status') }}</span>
      </div>
    </div>
  </div>
@endif
<div class="row">
  <div class="col-12 text-right">

    <a href="{{ route('commande.create') }}"
      class="btn btn-sm btn-
primary">{{ __('Ajouter commande') }}</a>

  </div>
</div>
<div class="table-responsive">
  <table class="table">
    <thead class="text-primary">
      <th>
        {{ __('Numéro') }}
      </th>
      <th>
        {{ __('Date') }}
      </th>
      <th>
        {{ __('Quantité commandé') }}

```

```

</th>

<th>
    {{ __( 'Prix' ) }}
</th>

<th>
    {{ __( 'Date de livraison' ) }}
</th>

<th>
    {{ __( 'Client' ) }}
</th>
</thead>

<tbody>
@foreach($commandes as $commande)
    <tr>
        <td>
            {{ $commande->NumeroCommande }}
        </td>
        <td>
            {{ $commande->DateCommande }}
        </td>
        <td>
            {{ $commande->QteCom }}
        </td>
        <td>
            {{ $commande->PrixVenUnit }}
        </td>
        <td>
            {{ $commande->DatePourLivrer }}
        </td>
        <td>
            {{ $commande->nom }}
        </td>
    </tr>
</tbody>

```

```

        <td class="td-actions text-right">
            <form action="{{ route('commande.destroy', $commande-
>id) }}" method="post">
                @csrf
                @method('delete')

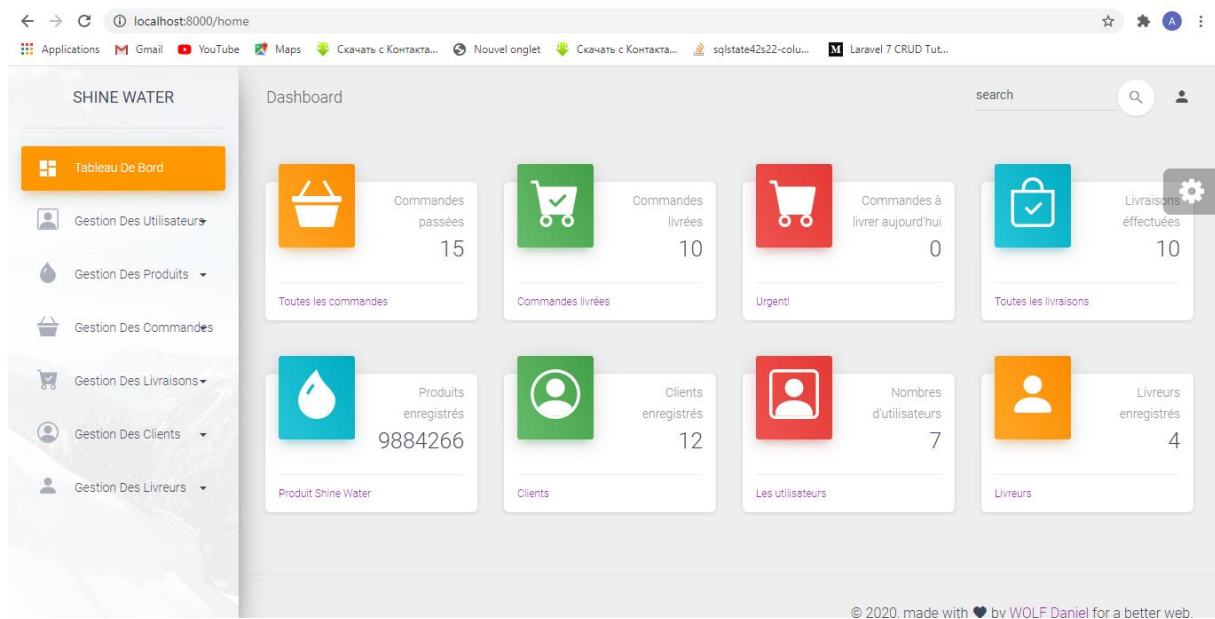
                <a rel="tooltip" class="btn btn-success btn-link"
                    href="{{ route('commande.edit', $commande->id) }}" data-
original-title=""
                    title="">
                    <i class="material-icons">edit</i>
                    <div class="ripple-container"></div>
                </a>
                <button type="button" class="btn btn-danger btn-link"
                    data-original-title="" title=""
                    onclick="confirm('{{ __("Etes vous sûr de supprimer cet
te commande?" ) }}') ? this.parentElement.submit() : "">
                    <i class="material-icons">close</i>
                    <div class="ripple-container"></div>
                </button>
            </form>

        </td>
    </tr>
    @endforeach
</tbody>
</table>
</div>
</div>
</div>

```

```
</div>
</div>
</div>
@endsection
```

## 5-Quelques statistiques



***Figure 24::Statistiques de Shine Water***



## CONCLUSION

Ce stage pratique de programmation de deux (02) mois nous a permis de faire face à certaines réalités du métier d'ingénieur des travaux informatiques et ainsi d'améliorer notre niveau d'expérience, d'approfondir et d'enrichir nos connaissances de base déjà acquises au cours de notre formation. Nous pouvons noter, pour ce qui nous a le plus marqué pendant ce stage, combien il est intéressant de trouver des solutions informatiques aux entreprises qui désirent informatiser leur Système d'Information (SI). En ce qui concerne notre application << SHINE WATER>>, elle rendra plusieurs tâches fiables et permettra de maîtriser plusieurs informations afin de faciliter plusieurs processus qui s'appliquent au sein d'une société de production d'eau.

La réalisation d'un tel projet, nous a permis d'apprendre et de toucher du doigt une partie de divers aspects du métier de développeur et de celui du concepteur.

### **Bibliographie**

- **Ouvrages**

- UML.2 par la pratique 5è Edition : Pascal Roques 4 Consulté le 24, 25, 26 juillet 2018.
- UML.2 Modéliser une application web 4è Edition : Pascal Roques Consulté le 24, 25, 26, 27, 28 juillet 2018.

- **Note de cours**

- Cours d'UML dispensé par M. AMEVOR, 2019-2020
- Cours de programmations web dispensé par M. ATTIOGBE, 2019-2020
- Cours de Programmation Orienté Objet dispensé par M. AMEVOR, 2019-2020
- Cours de Conception de base donnée dispensé par M. TCHANTCHO, 2020-2020
- Cours d'Implémentation de base donnée dispensé par M. KEOULA, 2020-2020

- **Site web consultés**

- [http : //www.google.com](http://www.google.com), tout au long du stage
- [http : //www.wikipedia.org](http://www.wikipedia.org), tout au long du stage
- [http : //www.openclassroom.com](http://www.openclassroom.com), tout au long du stage
- [http : //www.Gitub](http://www.Gitub), tout au long du stage
- [http : //www.youtube.com](http://www.youtube.com), tout au long du stage
- [http : //www.Laravel.com](http://www.Laravel.com), tout au long du stage
- [http : //www.systid.com](http://www.systid.com), tout au long du stage
- [http : //www.iaitogo.com](http://www.iaitogo.com), tout au long du stage

• **Anciens mémoires consultés**

- Mémoire de Mlle NIKABOU Siba Fidèle (2017-2018)
- Mémoire de M. KPAKPASSIM Esso-Wédéou (2018-2019)

**LISTES DES TABLEAUX**

Tableau 1:Coût matériel.....	10
Tableau 2:Coût de la conception .....	10
Tableau 3:Tableau du coût total.....	11
Tableau 4:Spécification fonctionnelle .....	12
Tableau 5:Spécification technique .....	13
Tableau 6:Planning prévisionnel de réalisation .....	14
Tableau 7:Les cas d'utilisations .....	21
Tableau 8:Tableau du matériel utilisé .....	41

## LISTES DES FIGURES

Figure 1:Logo d'UML .....	15
Figure 2:Illustration du 2TUP .....	16
Figure 3:Logo de PowerAMC .....	18
Figure 4:Diagramme de contexte statique de notre système .....	20
Figure 5:Diagramme de cas d'utilisation du système .....	23
Figure 6:Diagramme de séquence du cas d'utilisation « s'authentifier » .....	28
Figure 7:Diagramme de séquence du cas d'utilisation « enregistrer un produit » .....	29
Figure 8:Diagramme de séquence du cas d'utilisation « enregistrer une commande » .....	30
Figure 9:Diagramme de séquence du cas d'utilisation « Consulter la liste des commandes» .....	31
Figure 10: Diagramme d'activité du cas d'utilisation « s'authentifier » .....	33
Figure 11: Diagramme d'activité du cas d'utilisation « Enregistrer commande» .....	34
Figure 12:Diagramme d'activité du cas d'utilisation « Enregistrer livraison» .....	35
Figure 13:Diagramme d'activité du cas d'utilisation « Modifier Commande» .....	36
Figure 14:Diagramme de classe du système .....	38
Figure 15:Diagramme de déploiement du système .....	39
Figure 16:Logo PHP .....	41
Figure 17:Logo Laravel.....	42
Figure 18:Logo de Sublime Text.....	43
Figure 19:Logo de XAMPP .....	43
Figure 20:Logo d'Apache .....	44
Figure 21:Logo de Chrome.....	45
Figure 22:page d'authentification .....	55
Figure 23:page de la liste des commandes .....	58
Figure 24::Statistiques de Shine Water .....	62

## Table des matières

<b>INTRODUCTION .....</b>	<b>1</b>
<b>PARTIE I : RAPPORT D'INSERTION .....</b>	<b>2</b>
<b>I-PRESENTATIONS .....</b>	<b>3</b>
<b>1-PRESENTATION DE L'IAI-TOGO .....</b>	<b>3</b>
<b>II-THEME DU STAGE .....</b>	<b>4</b>
<b>1-Présentation du sujet .....</b>	<b>4</b>
<b>2-Problématique .....</b>	<b>4</b>
<b>3-Interêt du sujet.....</b>	<b>5</b>
a - Objectifs .....	5
b – Résultats attendus .....	5
<b>PARTIE II : RAPPORT DE PRE-PROGRAMMATION .....</b>	<b>6</b>
<b>I-ETUDE DE L'EXISTANT .....</b>	<b>7</b>
<b>1-Processus des commandes.....</b>	<b>7</b>
<b>2-Processus des livraisons .....</b>	<b>7</b>
<b>II-CRITIQUE DE L'EXISTANT .....</b>	<b>7</b>
<b>III-PROPOSITION DE SOLUTIONS .....</b>	<b>8</b>
<b>A-Première solution : ACHAT D'UN LOGICIEL DE GESTION.....</b>	<b>8</b>
<b>B-Deuxième solution : LA CONCEPTION D'UNE APPLICATION WEB .....</b>	<b>8</b>
<b>IV-EVALUATION FINANCIERE DES SOLUTIONS.....</b>	<b>9</b>
<b>A-ACHAT D'UN LOGICIEL DE GESTION .....</b>	<b>9</b>
<b>B-LA CONCEPTION D'UNE APPLICATION WEB.....</b>	<b>9</b>
a- Le coût matériel .....	9
<b>V-CHOIX DE SOLUTION.....</b>	<b>11</b>
<b>A-SPECIFICATION FONCTIONNELLE .....</b>	<b>12</b>
<b>B-SPECIFICATION TECHNIQUE.....</b>	<b>13</b>
<b>VI- PLANNING PREVISIONNEL DE REALISATION .....</b>	<b>13</b>
<b>VII-PRESENTATION DE LA METHODE D'ANALYSE .....</b>	<b>15</b>
<b>A-LE LANGAGE DE PROGRAMMATION .....</b>	<b>15</b>
<b>B-LE PROCESSUS DE MODELISATION.....</b>	<b>15</b>
<b>VIII-PRESENTATION DE L'OUTIL DE MODELISATION .....</b>	<b>17</b>
<b>A-QU'EST-CE QUE PowerAMC ?.....</b>	<b>18</b>
<b>B-MODELISATION AVEC PowerAMC.....</b>	<b>19</b>
<b>IX-ETUDE DETAILLEE DE LA SOLUTION .....</b>	<b>19</b>

<b>D-VUE LOGIQUE.....</b>	<b>37</b>
<b>E-VUE DE DEPLOIEMENT .....</b>	<b>39</b>
<b><i>PARTIE III : REALISATION ET MISE EN OEUVRE .....</i></b>	<b>40</b>
<b>I-MISE EN OEUVRE.....</b>	<b>41</b>
<b>II. PRESENTATION DE L'APPLICATION .....</b>	<b>46</b>
<b>CONCLUSION .....</b>	<b>63</b>
<b>Bibliographie .....</b>	<b>64</b>
<b>LISTES DES TABLEAUX .....</b>	<b>66</b>
<b>LISTES DES FIGURES .....</b>	<b>67</b>
<b>Table des matières .....</b>	<b>68</b>