

Province de Hainaut – IETC – Mercure  
Enseignement Supérieur Economique de Promotion Sociale

# ayamarket

application web d'e-commerce de vente de vêtements

Bachelier en informatique de gestion  
Epreuve intégrée

Candidat : Rachid AHGGOUNE  
Chargé de cours : Alain MBAYO BOGA ASANI KALUNGA  
2021/2022



# Remerciement

---

Tout d'abord, je voudrai remercier des personnes très spéciales à mes yeux, qui m'ont beaucoup inspiré et soutenu pendant ces quatre années à l'IETCps, il s'agit de mes enfants, ma compagne ainsi que mes professeurs.

En effet, j'ai pu compter sur le soutien sans faille de ma compagne. Elle m'a toujours accordé du temps, l'attention et la confiance, ressenti accentué lors de la réalisation du projet. Elle a été mon mentor et je souhaite lui exprimer toute ma gratitude à travers ce travail de fin d'études.

Je voudrais aussi remercier Prajwal pour son aide extraordinaire et sans qui ce projet n'aurait certainement pas était aussi parfait. merci à toi !

Le plus grand remerciement est réservé aussi à ma maman qui a toujours insisté pour que je reprenne mes études et m'encourage à persévérer afin que je puisse réaliser mes rêves, elle a toujours cru en moi.

Je tiens à exprimer mes sincères remerciements à notre chargé du cours Mr Alain d'avoir accepté de nous encadrer, pour son aide, ses conseils, pour la pertinence de ses remarques et la justesse de ses corrections.

Tous les enseignants de l'IETCps plus particulièrement ceux du département informatique, pour leurs apports inestimables durant notre cursus à savoir :

- Paul Ficherouille ;
- Gaëtan Bury ;
- Roberto Bruno ;
- Vincent Lothaire ;
- Sidi Yasser El Jasouli ;
- Vincent Paulis ;
- Rudy Druine .

Je remercie les membres du jury d'avoir bien voulu examiner ce travail et tous les membres de secrétariat et la direction de l'IETCps pour leur disponibilité.

# Table des matières

---

1	Introduction.....	7
2	Cahier des charges .....	8
2.1	Introduction.....	8
2.2	Présentation du projet .....	8
2.2.1	Qui est AyaMarket ? .....	8
2.2.2	Contexte général .....	8
2.2.3	Problématique.....	9
2.2.4	Solution.....	9
2.2.5	Organisation .....	10
2.2.6	Méthodologie de travail et modélisation.....	10
2.2.6.1	Cycle de vie .....	10
2.2.6.2	méthodologie .....	11
2.2.7	Échéance et livrables.....	13
2.2.1	Coût du projet .....	13
2.3	Analyse et besoin (TTUP – Branche fonctionnelle) .....	14
2.3.1	Besoins fonctionnels.....	14
2.3.2	Besoins non fonctionnels .....	14
2.3.3	Besoins techniques.....	15
2.3.3.1	Études des outils de développement .....	15
2.3.3.2	Études de Framework LARAVEL .....	15
2.4	Analyse fonctionnelle de la plateforme (TTUP – Branche fonctionnelle) .....	16
2.4.1	Back-office .....	16
2.4.2	Le Shop .....	16
2.4.3	Les fiches produit .....	17
2.4.4	La navigation .....	17
2.4.5	Les promotions .....	17
2.4.6	L'identification.....	17
2.4.7	Les données clients .....	18
2.4.8	Panier.....	18
2.4.9	Achat.....	18
2.4.10	Mode de livraison .....	18
2.4.11	Le paiement.....	19
2.4.12	Confirmation.....	19

2.4.13	Expédition : .....	19
2.4.14	La facturation .....	19
2.4.15	Modalités et traitement .....	19
2.5	Conclusion .....	19
3	Conception .....	20
3.1	Introduction.....	20
3.2	Conception générique (TTUP – Branche technique) .....	20
3.2.1	Le langage UML (Unified Modeling Language).....	20
3.2.2	Les diagrammes UML .....	20
3.2.3	Modèles d'UML utilisé.....	21
3.3	Conception préliminaire (TTUP - Branche conception & dev) .....	21
3.3.1	Architecture de site Web dynamique (MVC) .....	21
3.3.2	Laravel et le modèle MVC .....	22
3.3.3	Avantage de Laravel .....	22
3.3.4	Conception préliminaire des interfaces – maquettes .....	22
3.4	Conception détaillée (TTUP - Branche conception & dev).....	25
3.4.1	Identification des acteurs.....	25
3.4.2	Identification des cas d'utilisations .....	25
3.4.2.1	Modèle global de cas d'utilisation.....	25
3.4.2.2	Description détaillée des cas d'utilisation.....	26
3.4.3	Diagramme de séquence système .....	32
3.4.4	Diagramme de composant .....	34
3.4.5	Diagramme de classe.....	35
3.4.6	Base de données (MySQL).....	37
3.5	Schéma relationnel.....	45
3.6	Les Contrôleurs.....	46
3.7	Conclusion .....	48
4	Développement.....	49
4.1	Introduction.....	49
4.2	Environnement de développement .....	49
4.2.1	Environnement matériel .....	49
4.2.2	Environnement logiciel.....	49
4.3	Développement application Web (TTUP - Branche conception & dev) .....	50
4.3.1	Création du projet Laravel.....	50
4.4	La solution Web.....	50

4.4.1	Interface d'authentification Admin .....	50
4.4.2	Interface backend.....	52
4.4.3	Interface gestion des produits.....	52
4.4.4	Interface de gestion des utilisateurs .....	53
4.4.5	Interface de la liste des commandes.....	54
4.4.6	Interface d'accueil .....	54
4.4.7	Interface contact .....	57
4.4.8	Interface Shop .....	58
4.4.9	Interface commande .....	58
4.4.10	Interface Login / Register .....	59
4.5	Obligations légales .....	59
4.6	Validation w3c .....	59
4.7	Conclusion .....	60
5	Conclusion générale .....	61
6	Références.....	62
7	Bibliographie .....	64
8	Le lexique.....	66
9	Annexes .....	67

# Liste des figures

---

Figure 1 Solution proposée .....	10
Figure 2: Cycle en Y (2TUP) (https://www.e-bancel.com/, 2022) .....	12
Figure 3 Processus de réalisation d'application Web.....	13
Figure 4 Les langages de programmation côté serveur le plus utilisé .....	15
Figure 5 Architecture de laravel MVC : .....	22
Figure 6 Maquette Home - produit .....	23
Figure 7 Maquette Product single.....	23
Figure 8 Maquette Admin Page Edite .....	24
Figure 9 Maquette Admin page Liste .....	24
Figure 10 UseCase .....	26
Figure 11 Diagramme de séquence (Authentifier).....	33
Figure 12 Diagramme de séquence (s'inscrire) .....	33
Figure 13 Diagramme de séquence (payer) .....	34
Figure 14 Diagramme de composant .....	35
Figure 15 Diagramme de classe (1) .....	36
Figure 16 Diagramme de classe (2) .....	37
Figure 17 Base de données relationnelle .....	45
Figure 18 Contrôleurs Frontend .....	46
Figure 19 Contrôleurs Backend .....	47
Figure 20 Composants ui, --auth .....	51
Figure 21 Interface Login Admin .....	51
Figure 22 Interface accueil Backend.....	52
Figure 23 Interface de gestion de produits .....	53
Figure 24 interface de gestion des utilisateurs .....	53
Figure 25 interface de la liste des commandes.....	54
Figure 26 Interface d'accueil .....	54
Figure 27 Function Search .....	55
Figure 28 Résumé du code page HOME .....	56
Figure 29 Interface de contact .....	57
Figure 30 code du formulaire du contact.....	57
Figure 31 Interface Shop .....	58
Figure 32 Interface commande .....	58
Figure 33 Interface Login / Register .....	59
Figure 34 Diagramme de navigation Client.....	67
Figure 35 Diagramme de navigation Admin.....	67
Figure 36 Diagramme d'utilisation (Admin) .....	68
Figure 37 Page de commande .....	68
Figure 38 Page confirmation de commande .....	69
Figure 39 Page d'information sur l'entreprise.....	69
Figure 40 Algorithme d' Incription .....	70

# 1 Introduction

---

Le monde actuel vit baigné dans la technologie informatique, aucun secteur n'est épargné ! Surtout depuis la crise sanitaire du « COVID19 » survenue en 2020 et dont nous subissons encore les effets aujourd'hui. Cette crise a notamment, amplifié de manière exponentielle l'e-commerce dans tous les secteurs actifs.

C'est tout naturellement qu'à l'issue de mes années d'études, ainsi que par la rencontre de personnes exceptionnelles, dont les professeurs, les étudiants de mon cursus et les collègues lors de stage, que je me suis penché sur la réalisation d'un projet personnel longuement et mûrement réfléchi, cette réalisation se concrétise par la création d'une application Web d'e-commerce qui représentera ma société de vente de vêtements « AYAMARKET ».

Néanmoins, avant de me lancer, plusieurs questions se posent pour la création de cette application Web :

- Quels sont les besoins de cette application ?
- Comment gérer les différentes données et leurs confidentialités ?
- Quels développements et outils seront nécessaires ?
- Comment rendre son ergonomie ~~peçayw\$fsyxni\$swwnfpi\$~~

Dans le but de bien m'organiser pour répondre à ces questions, J'ai décidé de séparer mon travail en 3 chapitres :

- Le premier chapitre traitera des besoins et des objectifs à atteindre pour le bon fonctionnement de l'application ;
- Le second chapitre décrira la conception UML relative à la solution proposée ;
- Enfin, le troisième chapitre sera consacré à la présentation des outils utilisées pour la mise en œuvre de la solution et quelques interfaces de l'application réalisé.

Bonne lecture !

## 2 Cahier des charges

---

### 2.1 Introduction

Au cours de ce chapitre, nous allons étudier le cadre général de notre projet et la problématique qui a poussé AyaMarket à réaliser cette application. Nous allons définir les besoins fonctionnels et non-fonctionnels de l'application et proposer les orientations de notre solution future.

Avant d'élaborer notre solution, nous avons étudié les différentes méthodologies de travail existantes afin de choisir la plus adaptée à notre projet.

### 2.2 Présentation du projet

#### 2.2.1 Qui est AyaMarket ?

AyaMarket est une entreprise de prêt-à-porter qui sera présente uniquement sur Internet et plus particulièrement via les réseaux sociaux, et qui traitera, dans un premier temps, uniquement avec les clients belges, ensuite, il est prévu d'élargir notre clientèle à travers les pays limitrophes de la Belgique.

La société compte démarrer ces activités début 2023, après la mise en place du réseau d'approvisionnement et d'acheminement jusqu'au client final.

AyaMarket emploie aujourd'hui 3 collaborateurs qui sont eux-mêmes fondateurs de l'enseigne. Leurs objectifs sont de créer au moins 200 postes de travail et à terme atteindre 3 millions d'euros de chiffre d'affaires annuel pour passer à l'ensemble de l'Europe.

Les cibles sont les particuliers (homme et femme) plutôt tendance, le site s'adresse surtout aux adultes qui sont souvent décideurs ou prescriptifs dans les achats de vêtements. AyaMarket mettra également en vente les grandes marques de vêtements avec des prix au rabais (Outlet) visant une large clientèle.

Les objectifs marketing de AyaMarket sont de :

- Faire de AyaMarket une enseigne qui définit la marque pour tous ;
- Donner à la marque les moyens de s'adapter aux différentes situations comme en période de crise sanitaire ;

#### 2.2.2 Contexte général

L'e-commerce est en constante évolution, nous sommes témoins d'une transformation digitale sans précédent. Cette évolution s'est bien implantée dans le quotidien du consommateur, son impact se fait sentir sur la façon dont le consommateur s'approprie l'information.

Nous constatons donc que nous ne pouvons pas négliger la vente en ligne, ce qui nous oblige à nous intéresser au commerce électronique. Le processus de l'e-commerce facilite l'accès aux produits et services. Le défi consiste à attirer les internautes qui deviendront nos futurs clients via une interface agréable qui facilitera la procédure d'achat de nos produits.

Dans cette perspective, la mode reste le **premier marché sur Internet** en termes de nombre d'acheteurs. En 2020, selon la Fevad (Fédération du e-commerce et de la vente à distance), 51 % des

produits et services achetés sur le Web font partie du secteur de l'habillement, d'où vient l'importance du domaine de vente de vêtements.

### 2.2.3 Problématique

Dans le cadre de cette vision, AyaMarket a pensé à promouvoir ses produits via des « Live : présentation en direct » diffusés sur les réseaux sociaux afin de présenter les produits quotidiennement avec de nouvelles collections hebdomadaires, le tout, agrémenté de promotions alléchantes.

Ma mission consiste à réaliser une application web pour AyaMarket. Cette dernière facilitera la gestion des ventes et permettra aux clients d'acheter nos produits sur notre plateforme. Les clients pourront acheter les produits directement sur le site, Tout les produits d'AyaMarket seront accessibles, qu'ils aient eu une démonstration lors d'un live ou non, ou qu'ils soient d'une collection précédente.

### 2.2.4 Solution

Face aux différentes problématiques, une question se pose :

Existe-t-il une application qui correspond à nos besoins ?

Non, les applications disponibles sont globales et souvent payantes. AyaMarket a besoin d'une application spécifique, qui répond à ses besoins spécifiques tout en respectant son autonomie.

C'est pourquoi nous avons pensé à analyser et développer l'application qui offrira LA bonne solution qui nous correspond.

Les objectifs de l'application Web e-commerce sont :

- Donner envie au client d'acheter sur le site et d'y revenir souvent ;
- Donner envie au client de suivre le site ;
- Donner envie au client d'en parler à son entourage ;
- D'augmenter le panier moyen.

L'application doit assurer la fiabilité des informations fournies par les utilisateurs à l'aide d'un système de pré-validation. Elle doit aussi assurer une meilleure sécurité en termes de gestion des droits d'accès ainsi qu'une interface ergonomique qui permettra aux utilisateurs de se guider sans avoir besoin d'aide, aussi agréable et simple d'utilisation qui ne perdra pas les utilisateurs.

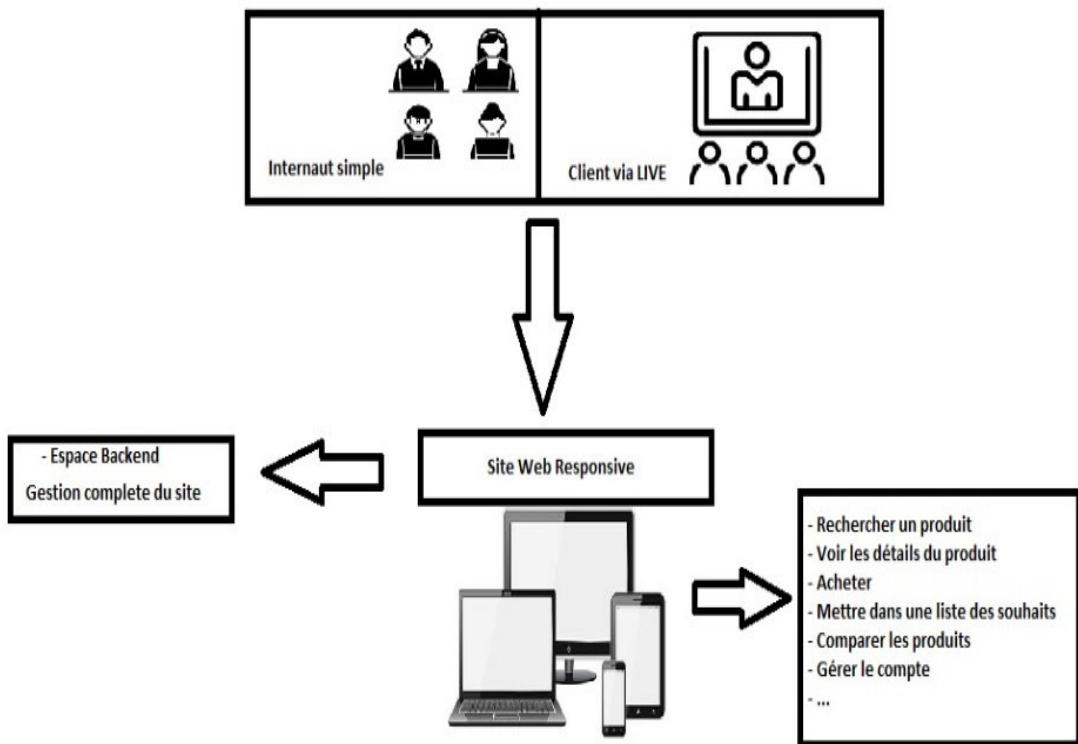


Figure 1 Solution proposée

## 2.2.5 Organisation

Le projet est mené conjointement par Rachid AHGGOUNE et Samira ZAMZAM qui sont Chefs de projets E-commerce.

Leurs responsabilités respectives sont :

- Gestion du site e-commerce et logistiques ;
- Gestion marketing, juridique et problématique ;

Dans le futur proche, l'équipe s'agrandira et pourra être constituée de chefs de projets, de rédacteurs, de support IT, d'infographistes, ...

## 2.2.6 Méthodologie de travail et modélisation

### 2.2.6.1 Cycle de vie

Le cycle de vie d'une application est constitué de plusieurs étapes importantes jalonnant son parcours dont au moins les étapes suivantes :

- Analyse des besoins et étude de faisabilité: vise à identifier les exigences du projet et à faire le point sur les éléments attendus, puis l'estimation de la faisabilité de ces besoins.
- Conception générale : élaboration des spécifications de l'architecture générale.
- Conception détaillée : détail chaque sous-ensemble.

- Réalisation : Traduire dans un langage de programmation des fonctionnalités définies lors de la conception.
- Tests unitaires : Vérification du bon fonctionnement par partie du code.
- Intégration : S'assurer de l'interfaçage des différents éléments.
- Qualification : Vérification de la conformité de l'application aux spécifications initiales.
- Mise en production : l'hébergement de l'application
- Maintenance : les interventions sur l'application après utilisation.

#### *2.2.6.2 méthodologie*

Il est indispensable d'utiliser une méthodologie de développement pour structurer, planifier et contrôler le développement de notre application. Avant d'entamer la réalisation de l'application, nous avons modélisé un système pour la compréhension de son fonctionnement et assuré sa cohérence.

Vu que le projet est basé sur un processus de développement bien défini qui commence par la définition des besoins fonctionnelle jusqu'au codage final, le cycle de développement le plus approprié est celui qui dissocie les aspects techniques des aspects fonctionnels : **Tow Tracks Unified Process** est la méthodologie que nous allons adopter pour la suite de notre travail.

Le processus s'articule autour de trois phases essentielles :

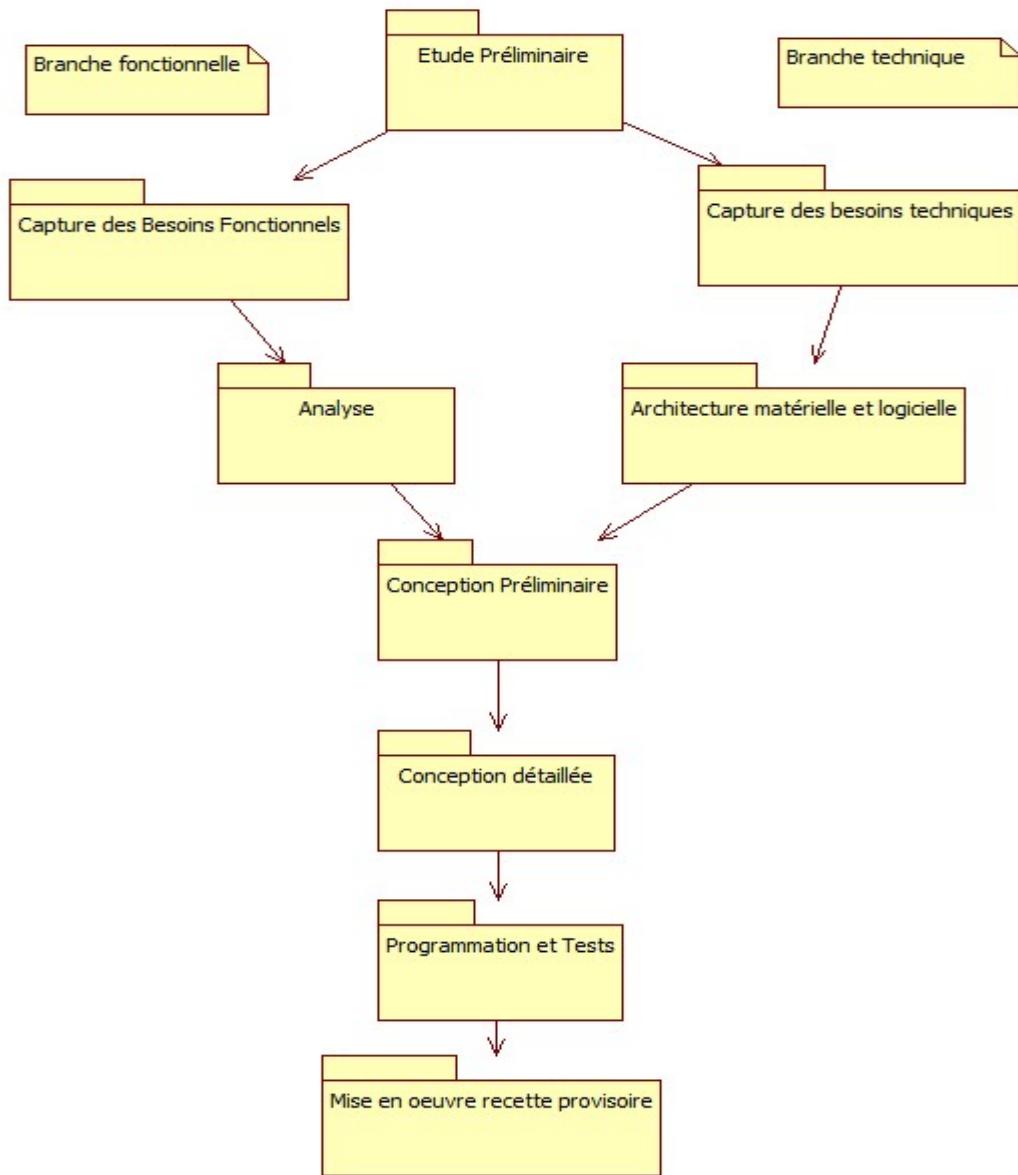
- Une branche technique ;
- Une branche fonctionnelle ;
- Une phase de réalisation.

Ce processus de développement est caractérisé par :

- Un processus itératif et incrémental, poussé par les circonstances d'utilisation ;
- Centré sur l'architecture et les modèles UML.

Il définit un processus d'intégration de toutes les activités de conception et de réalisation dans les cycles de développement des composés

- D'une phase de création ;
- D'une phase de développement ;
- D'Une phase de construction ;
- D'Une phase de transition, comprenant chacun un certain nombre d'itérations.



1

Figure 2: Cycle en Y (2TUP) (<https://www.e-bancel.com/>, 2022)

Nous suivons scrupuleusement ces étapes, présentées dans la figure ci-dessus.

Le processus illustré par la figure suivante s'arrête lorsque l'application Web a été réalisée, testée et peut être livrée. D'autres activités liées à ce type de projets, comme le Web marketing et l'hébergement, ne sont pas dans la portée de ce processus.

La maintenance de l'application ne sera pas évoquée comme une étape du processus. Elle peut être vue comme des interactions supplémentaires sur les dernières étapes. Les évolutions et ajouts de fonctionnalités peuvent être modélisés par un nouveau projet et donc un reparcourt du processus depuis le début.

---

<sup>1</sup> <https://www.e-bancel.com/.> (2022, 01 01). [https://www.e-bancel.com/\).](https://www.e-bancel.com/>.) Consulté le 03 01, 2022, sur E-bancel: [https://www.e-bancel.com/Processus\\_2TUP.php](https://www.e-bancel.com/Processus_2TUP.php)

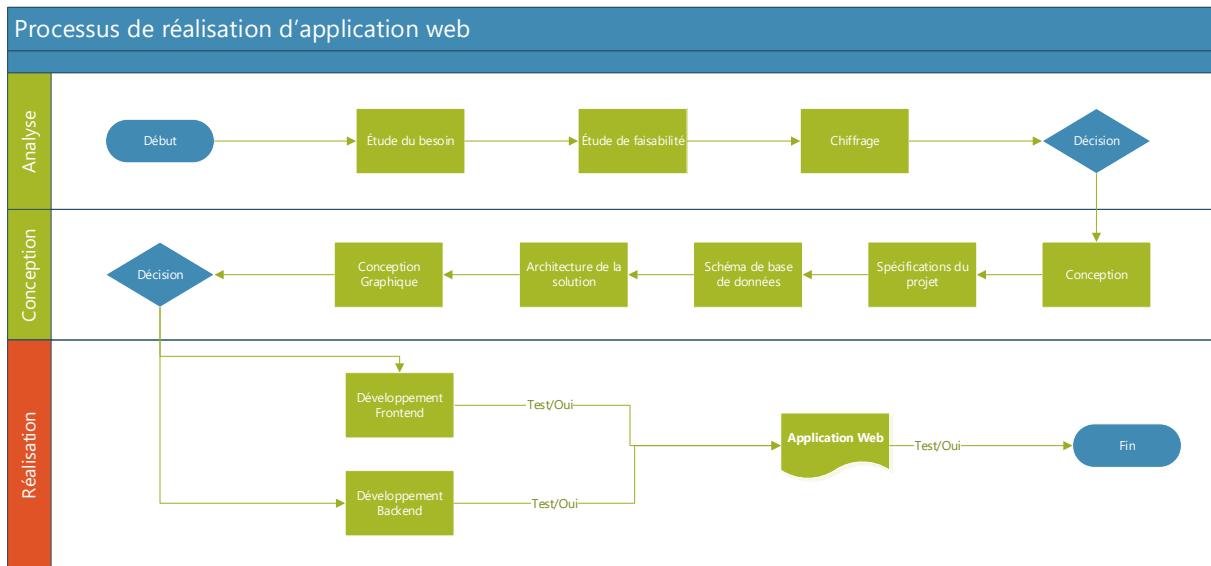


Figure 3 Processus de réalisation d'application Web

#### 2.2.7 Échéance et livrables

Afin d'ordonnancer et déterminer les tâches du projet, il est primordial de tout planifier, cette étape compte parmi les phases « avant-projet » qui estime les charges respectives des tâches.

AyaMarket se lancera début 2023, pour les premières soldes de l'année 2023.

Avant cette date, le projet devrait respecter les échéances suivantes :

- Conception graphique et ergonomique du site 17/11/2021 ;
- L'ensemble des contenus hors visuels 01/12/2021 ;
- Livraison du site e-commerce en maquette 15/01/2022 ;
- Livraison du site e-commerce test 1 25/02/2022 ;
- Livraison du site e-commerce pour validation 15/03/2022 ;
- Livraison de site e-commerce pour finition 15/04/2022 ;
- Formation à la plateforme (technique et fonctionnelle) 02/05/2022 ;
- Mise en ligne du site 01/12/2022 ;

La proposition devra être constituée de :

- Une présentation PowerPoint qui représente les maquettes de back-office ;
- Une présentation des fonctionnalités fournies par défaut dans la plateforme e-commerce ;
- Une présentation des fonctionnalités qui devront être développées ;

#### 2.2.1 Coût du projet

Le projet ne sera pas financièrement évalué, car il n'a pas été facturé. En effet, je suis co-fondateur de l'entreprise et j'apporte mon savoir-faire à l'entreprise.

## 2.3 Analyse et besoin (TTUP – Branche fonctionnelle)

### 2.3.1 Besoins fonctionnels

Le système devrait répondre à des besoins, à des acteurs susceptibles d'interagir avec le système, nous commençons par définir ce qui est un acteur.

Un acteur joue le rôle des entités externes (que ce soit humain, dispositif matériel ou autre système) qui interagit directement avec le système. Dans notre application je liste les acteurs suivants :

- Internaute pourra :
  - Créer un compte dans le système.
  - Consulter les produits : chaque produit sera présenté en détail sur une page qui contiendra des images, une description, un prix et sa disponibilité ;
  - Chercher un produit : la recherche peut se faire à l'aide de mots clés ou références, ce filtre sera actif sur la page en cours.
  - Contacter le service AyaMarket : via un formulaire de contact ;
- Client pourra :
  - En plus des accès de l'internaute ;
  - Gérer le panier : après l'ajout d'un produit, le client peut effacer ou consulter la disponibilité du produit qui sera confirmé au moment du paiement.
  - Gérer ses commandes : une page liste les commandes déjà effectuées ou en cours afin de visualiser le statut de chacune.
  - Gérer son compte : le client aura la possibilité de modifier ses informations personnelles sauf l'e-mail.
- L'administrateur pourra :
  - Gérer le site : l'administrateur peut gérer la partie back-end du site (produit, catégorie et toutes autres informations liées au site ou aux produits) ;
  - Gérer les utilisateurs : bloquer, supprimer ou modifier le compte des membres.
  - Consulter les commandes : consulter, manipuler et modifier le statut des commandes.

### 2.3.2 Besoins non fonctionnels

Les besoins non fonctionnels spécifient les propriétés du système, contraintes d'implémentation, d'environnement et de performance.

- Qualité :
  - Ergonomie : effectuer un achat et passer une commande doit être agréable, rapide et le plus simple possible, sans contraintes ;
  - Commande simple : une seule page, formulaire, pour fournir les renseignements du bon de commande. (Adresse, mode de livraison et mode de paiement) ;
- Performance :
  - Sécurité : l'authentification des clients par un login et mot de passe crypté et le paiement par PayPal garantit aux clients la confidentialité de ses données ;
  - L'intégrité : l'application doit traiter les erreurs des interfaces, effectuer les traitements des mauvaises données sans oublier l'intégrité des données dans la base de données et des interfaces.
  - Portabilité : le système doit être compatible avec différents systèmes d'exploitation ainsi que les navigateurs ;
- Contraintes :

- Langue : l'application Web doit être en anglais pour atteindre plus de clients en Belgique avec une perspective internationale.
- Base de données : les informations sur les produits devront être saisies depuis un espace administrateur.
- Les informations saisies par le client via les formulaires du site Web doivent être enregistrées dans la base de données et protégées afin de garantir la confidentialité
- Les commandes seront enregistrées dans la base de données, puis traitées par l'administrateur, et le client peut consulter l'historique de toutes ses commandes.

### 2.3.3 Besoins techniques

#### 2.3.3.1 Études des outils de développement

Notre choix est orienté vers le langage de programmation PHP pour développer notre site Web, tout d'abord, PHP est gratuit et ne nécessite aucune licence d'utilisation. Une communauté de développeurs très actifs qui met à disposition des librairies PHP de grande qualité.

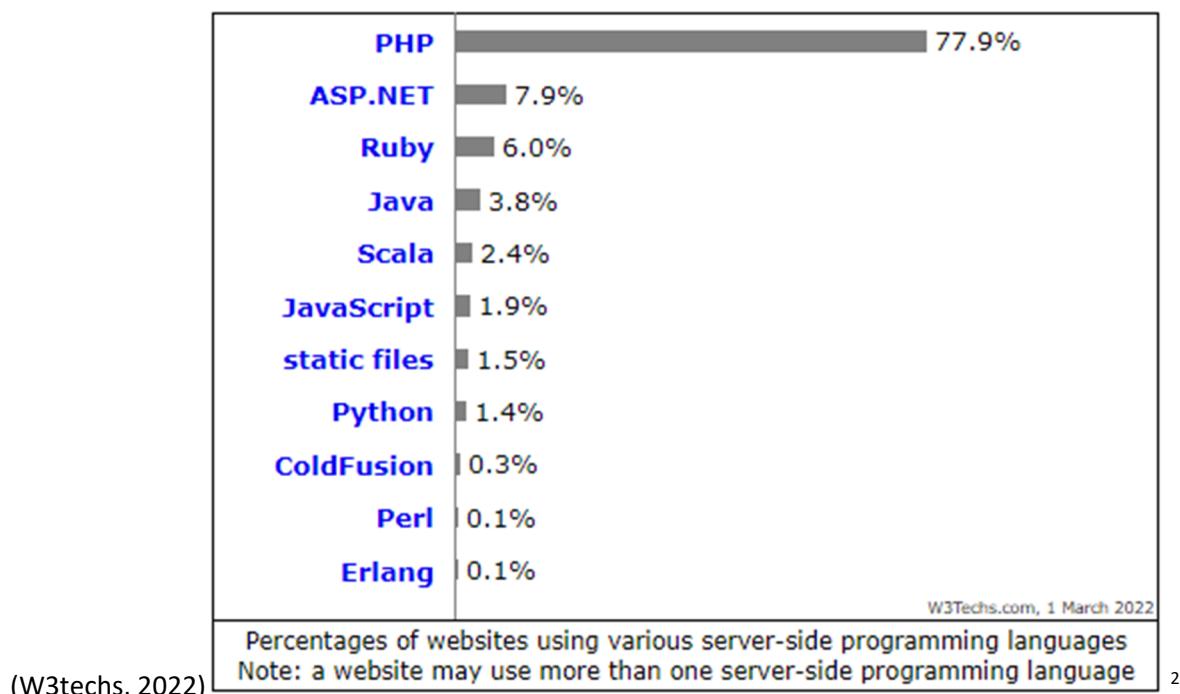


Figure 4 Les langages de programmation côté serveur le plus utilisé

#### 2.3.3.2 Études de Framework LARAVEL

Laravel est un **Framework PHP** open source conçu pour rendre le développement d'applications Web plus facile et plus rapide grâce à des fonctionnalités intégrées. Ces fonctionnalités font partie de ce qui rend **Laravel** si largement utilisé par les développeurs Web.

Entre autres, on cite :

- Un système modulaire avec gestion des dépendances. Cela signifie que vous pouvez facilement ajouter des fonctionnalités à votre **application Laravel** sans les écrire from scratch. Vous

<sup>2</sup> W3techs. (2022, 01 01). *Programming language*. Consulté le 03 01, 2022, sur <https://w3techs.com/>: [https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language)

pouvez soit créer vos propres packages pour le code que vous utilisez régulièrement, soit installer des packages prêts à l'emploi via le **Composer**.

- Un système d'authentification complet
- Une cartographie **objet-relationnelle**. L'**Eloquent ORM** inclus dans **Laravel** présente les tables de base de données sous forme de classes pour faciliter l'accès et la manipulation des données.
- Une interface de ligne de commande (**CLI**) fournie avec des dizaines de commandes prédéfinies.
- Les Tests automatiques. Les tests automatisés font partie intégrante de Laravel.
- Un environnement de développement portable et virtuel. **Laravel Homestead** fournit aux développeurs tous les outils nécessaires pour développer rapidement et facilement Out Of The Box. (CORRE, 2020)<sup>3</sup>

Nous avons porté notre choix sur LARAVEL pour réaliser l'application Web, car ce Framework possède de meilleurs avantages par rapport aux autres, comme le support d'Ajax et des extensions. LARAVEL est parmi les Frameworks le plus utilisé dans le monde, alors je n'aurais pas de difficulté à trouver de la documentation ou de l'aide si besoin, car il y a une communauté très active et disponible au niveau de l'entraide.

## 2.4 Analyse fonctionnelle de la plateforme (TTUP – Branche fonctionnelle)

Dans le cadre de l'analyse fonctionnelle, nous décrivons les besoins d'AyaMarket par rapport à la plateforme e-commerce. En effet, cette dernière doit répondre à des exigences, d'une part, d'ordre interne liées à l'utilisation de la solution, et d'autre part, d'ordre commercial car le site e-commerce représente l'image de notre marque.

### 2.4.1 Back-office

La disponibilité de back-office 24/24 7/7.

Cette partie de la plateforme doit être ergonomique, simple et ne doit pas nécessiter la mise en place d'une équipe informatique.

Le back-office doit exister en version anglaise.

L'accès doit être sécurisé avec des logins pour chaque membre de l'équipe afin de consulter ou gérer le contenu de l'application.

Les modifications portées aux données à partir du back-office doivent être immédiatement prises en compte et appliquer sur la Base de données.

### 2.4.2 Le Shop

La plateforme affichera tous les produits et réalisera la gestion du stock en adaptant les quantités en temps réel.

L'affichage des produits pourra être aléatoire ou non et permettre aux clients de voir différents produits à chaque rafraîchissement de la page ou pendant la navigation entre pages.

---

<sup>3</sup> CORRE, F. L. (2020, 12 17). <https://www.alphorm.com/>. Récupéré sur <https://www.alphorm.com/tutoriel/formation-en-ligne-laravel-le-guide-complet-du-debutant/tuto-video-presentation-de-laravel-et-pourquoi-l-utiliser>

#### 2.4.3 Les fiches produit

La plateforme doit pouvoir gérer un nombre illimité de produits avec leurs descriptions complètes.

Attention, Certains champs ne sont pas affichés autre part que dans le back office.

Tableau 1 Fiche produit

Affichés pour tout le monde	Affichés au back-office
<ul style="list-style-type: none"><li>- <b>Le nom commercial du produit ;</b></li><li>- <b>La référence ;</b></li><li>- <b>La description ;</b></li><li>- <b>Les Couleurs disponibles ;</b></li><li>- <b>Les tailles disponibles ;</b></li><li>- <b>La photo principale ;</b></li><li>- <b>Des photos secondaires ;</b></li><li>- <b>La mention (NEW) et autres ;</b></li><li>- <b>La mention (promo - ...) ;</b></li><li>- <b>Le prix (produit, livraison.).</b></li></ul>	<ul style="list-style-type: none"><li>- Différents prix pratiqués ;</li><li>- Quantités disponibles ;</li><li>- Taille ;</li><li>- Photos ;</li><li>- Description ;</li><li>- Mention new promo.</li></ul>

Lors de l'affichage, l'internaute peut ajouter le produit au panier, si disponible en stock, ou à une sélection pour un achat ultérieur.

Le back-office permettra de modifier des éléments des fiches une par une si besoin ;

La plateforme met automatiquement à l'échelle des photos.

Back-office doit permettre la gestion des photos de chaque produit ou catégorie.

#### 2.4.4 La navigation

L'internaute ne doit pas avoir de difficulté à naviguer entre les produits ou de revenir en arrière sans perdre d'information.

Le shop doit avoir une architecture arborescente et ce, afin de faciliter la navigation par catégorie. L'internaute peut filtrer aussi sur base de ces catégories, sous-catégories ou par prix.

Le back-office doit permettre la gestion des catégories et des sous-catégories, de mettre un lien entre produit et catégorie, ajouter, modifier et supprimer facilement.

#### 2.4.5 Les promotions

Il y'aura deux types de promotions : Promotion rabais en pourcentage et Promotion rabais fixe. Les promotions doivent être gérées à partir du back-office en %, ou avec des codes de réductions. La plateforme doit afficher les produits en promotion.

#### 2.4.6 L'identification

Le site met en place dès l'arrivée de l'internaute la possibilité de s'identifier ou de s'inscrire avec un message lisible. Sinon laisser la possibilité de le faire à n'importe quel moment, de n'importe quel page. Après l'identification, afficher le nom du client.

Pour passer une commande, l'identification est obligatoire.

Le client qui oublie son mot de passe peut demander à en recevoir un nouveau sur sa boîte mail. Seulement s'il a fourni un email valide à son inscription et que ce dernier est toujours bien présent dans notre base de données.

La possibilité de consulter et modifier les données des clients pour un usage SAV par l'admin, sans avoir accès au mot de passe du client.

#### 2.4.7 Les données clients

Les données du client seront accessibles par ce dernier à tout moment afin de pouvoir les modifier. Le client peut demander de se désinscrire du site à tout moment via le formulaire de contact.

Le client peut consulter aussi l'historique de ses achats.

Les données sont sécurisées par un mot de passe tout en respectant la durée du stockage de tes données suivant le GDPR. Et le traitement sera dans le cadre de notre application.

#### 2.4.8 Panier

Le panier doit être accessible et visible sur la page en cours, avec le nombre d'articles qu'il contient. La page panier doit afficher en détail les produits qu'il contient.

La possibilité d'ajouter un code promo à chaque instant. Mais à vérifier de sa validité au moment de passer la commande.

Avant de passer commande, le client peut ajouter ou supprimer des éléments du panier.

La plateforme doit mettre à jour le panier si le stock d'un produit qui s'y trouve change.

Un bouton de passer commande dans la page panier renvoie vers l'identification si ce n'est pas déjà fait.

#### 2.4.9 Achat

Pour chaque achat le client doit suivre un processus bien défini.

(Identification) ! -> sélection du produit -> ajout dans le panier -> validation -> (identification) ! -> confirmer les coordonnées de livraisons et facturation, choix du mode de livraison et choix du mode de paiement -> confirmation de la commande -> réception de la confirmation de commande sur la page et par mail.

La plateforme doit permettre de garder la liste du panier visible à chaque étape, signaler aux clients le moment d'attente du serveur. L'informer de la moindre des choses pendant le processus.

#### 2.4.10 Mode de livraison

Pour l'instant nous travaillerons qu'avec des livraisons 24h à un point de relai en Belgique, ce qui nous permettra d'offrir une livraison rapide, avec des frais de port réduits.

La plateforme devrait :

- Mettre en place un choix de mode de livraison si notre politique d'expédition évoluait.
- Mémoriser les adresses de livraison ;

- Gérer les frais de livraison ;

#### 2.4.11 Le paiement

Le seul moyen de paiement est PAYPAL, par la suite il y aura d'autres modes de paiement comme carte de banque. Back office doit prévoir un paramétrage du mode de paiement et gérer les modes de paiement.

#### 2.4.12 Confirmation

Après le paiement, une confirmation est affichée à l'écran ainsi un e-mail de confirmation est envoyé avec l'ensemble des informations lié à la commande.

Dans le back-office, on devrait pouvoir paramétrier le serveur SMTP pour l'emailing.

#### 2.4.13 Expédition :

Quand la commande est remise aux prestataires de livraison, le client est informé dans son espace client,

Le back-office permettra de consulter et modifier ces informations.

#### 2.4.14 La facturation

Après paiement et expédition de la commande du client, la société doit informer le client par mail et joindre les détails de la commande.

L'application ne doit pas gérer pas la facturation.

#### 2.4.15 Modalités et traitement

Le retour sera traité indépendamment de la plateforme, via une demande par mail au service après-vente, toute la communication se déroulera par mail.

### 2.5 Conclusion

Nous avons mis en pratique le processus 2TUP et nous avons, aussi, capturé les différents besoins de notre projet. Nous avons défini ces besoins en besoins fonctionnels, non fonctionnels et techniques. Après l'analyse des exigences et le choix des techniques à utiliser, il est temps de passer à la conception de notre application.

### 3 Conception

---

#### 3.1 Introduction

Maintenant que le côté analyse est terminé, l'aspect conceptuel de l'application peut commencer. Pour la conception et la réalisation de cette dernière, je vais m'appuyer sur le formalisme UML basé sur les diagrammes.

#### 3.2 Conception générique (TTUP – Branche technique)

Cette phase permet de décrire de manière non ambiguë, grâce à UML, le fonctionnement futur du système, afin d'en faciliter la réalisation.

##### 3.2.1 Le langage UML (Unified Modeling Language)

UML va me faciliter la tâche, ça me permettra de modéliser le problème de façon standard. Ce langage est devenu une référence en termes de modélisation objet.

Ce langage est caractérisé par :

- Formel & normalisé ;
- Gain de précision ;
- Gage de stabilité ;
- Support de communication performant ;
- Cadre l'analyse ;
- Facilite la compréhension de représentation abstraite et complexes ;
- Langage polyvalent et universel.

##### 3.2.2 Les diagrammes UML

UML 2.5 tourne sur trois points de vue classiques de modélisation : fonctionnel, statique et dynamique,

- Fonctionnel :
  - o Diagramme de cas d'utilisation : représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier ;
  - o Diagramme de séquence : représente la séquence verticale des messages passés entre objets au sein d'une interaction ;
  - o Diagramme d'activité : l'enchaînement des actions et décisions au sein d'une activité ;
- Statique :
  - o Diagramme de classes : est le point central dans un développement POO. Son objectif est de décrire la structure des entités utilisée par les utilisateurs, en analyse, et en conception, il représente la structure d'un code orienté objet ou les modules du langage de développement ;
  - o Diagramme de packages : il montre l'organisation logique du modèle et les relations entre packages ;
  - o Diagramme d'objet : met en évidence les instances des éléments structurels et leurs liens à l'exécution ;

- Diagramme de structure composite : il montre des structures complexes avec leurs interfaces fournies et requises ;
- Diagramme de déploiement : décris le déploiement physique des informations générées par le logiciel sur des composants matériels ;
- Dynamique :
  - Diagramme d'états : montre les états et transactions possibles des objets d'une classe ;
  - Diagramme d'activité : décris l'enchaînement des actions et décisions au sein d'une activité ;
  - Diagramme de séquence : affiche les événements par ordre chronologique ;
  - Diagramme de communication : fournit une représentation graphique des interactions entre les objets d'un scénario de cas d'utilisation, l'exécution d'une opération, ou une interaction entre des classes, en mettant l'accent sur la structure du système.

### 3.2.3 Modelés d'UML utilisé

Les treize diagrammes du UML 2.5 sont des moyens de description des objets ainsi que les liens qui les relient. Nous n'allons pas utiliser l'ensemble de ces diagrammes, mais seulement les diagrammes suivants :

- Diagramme de cas d'utilisation ;
- Diagramme de séquence ;
- Diagramme de composants ;
- Diagramme de classes.

Ces quatre diagrammes sont suffisants pour la plupart des projets et particulièrement pour mon projet.

## 3.3 Conception préliminaire (TTUP - Branche conception & dev)

### 3.3.1 Architecture de site Web dynamique (MVC)

Pour chaque lancement de développement d'un site Web dynamique, le choix du DESIGN-PATTERN est primordial, souvent présenté par MVC (Modèle-Vue-Contrôleur), qui dans une architecture en couche se retrouve implémentée dans la couche présentation. Le but c'est d'avoir une organisation rigoureuse et logique du code.

- Modèle : modèle de données ;
- Vue : présentation, interface utilisateur ;
- Contrôleur : logique de contrôle, gestion des événements et synchronisation.

Cette architecture a pas mal d'avantages pour qu'elle reste toujours la plus utilisée dans le monde de développement Web étant donné qu'elle se caractérise par l'allégement du poste de travail, la prise en compte de l'hétérogénéité des plateformes, l'introduction de clients légers et une meilleure répartition de la charge entre les différentes entités clients et serveurs.

### 3.3.2 Laravel et le modèle MVC

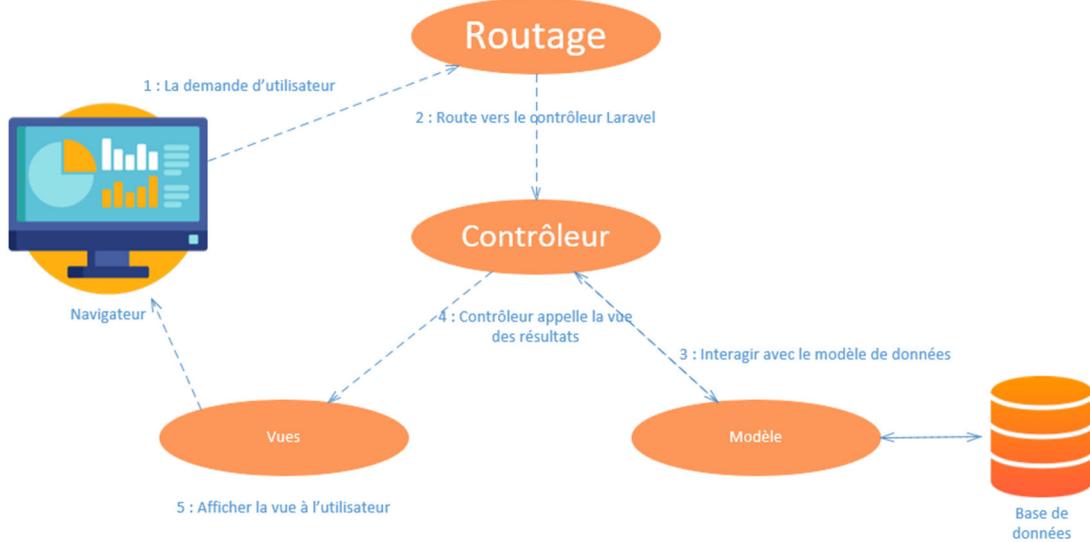


Figure 5 Architecture de laravel MVC :

### 3.3.3 Avantage de Laravel

Laravel offre de nombreux avantages pour concevoir une application Web dessus :

- L'application Web devient plus évolutive, grâce au Framework Laravel ;
- Gaine de temps importants dans la conception de l'application Web ;
- Laravel réutilise les composants d'autre Framework dans le développement d'applications Web ;
- Il comprend des espaces de noms et des interfaces, ce qui permet d'organiser et de gérer les ressources.

#### Composer

C'est un outil en PHP qui inclut des dépendances et des bibliothèques.

- Composer permet à l'utilisateur de créer des projets en respectant le cadre mentionné.
- Nous pouvons installer des bibliothèques tierces avec l'aide de composer.
- Toutes les dépendances de composer sont écrites dans le fichier *composer.json* placé dans le dossier source.

#### Artisan

- Artisan est une interface de ligne de commande utilisée dans laravel.
- Ensemble de commandes incluses dans artisan pour aider à créer une application Web.

### 3.3.4 Conception préliminaire des interfaces – maquettes

Une maquette est une présentation préliminaire du projet, qui reste évolutif, elle est développée rapidement afin d'améliorer la relation développeur client.

Les maquettes réalisées avec VISIO de Microsoft pour notre application sont les suivantes :

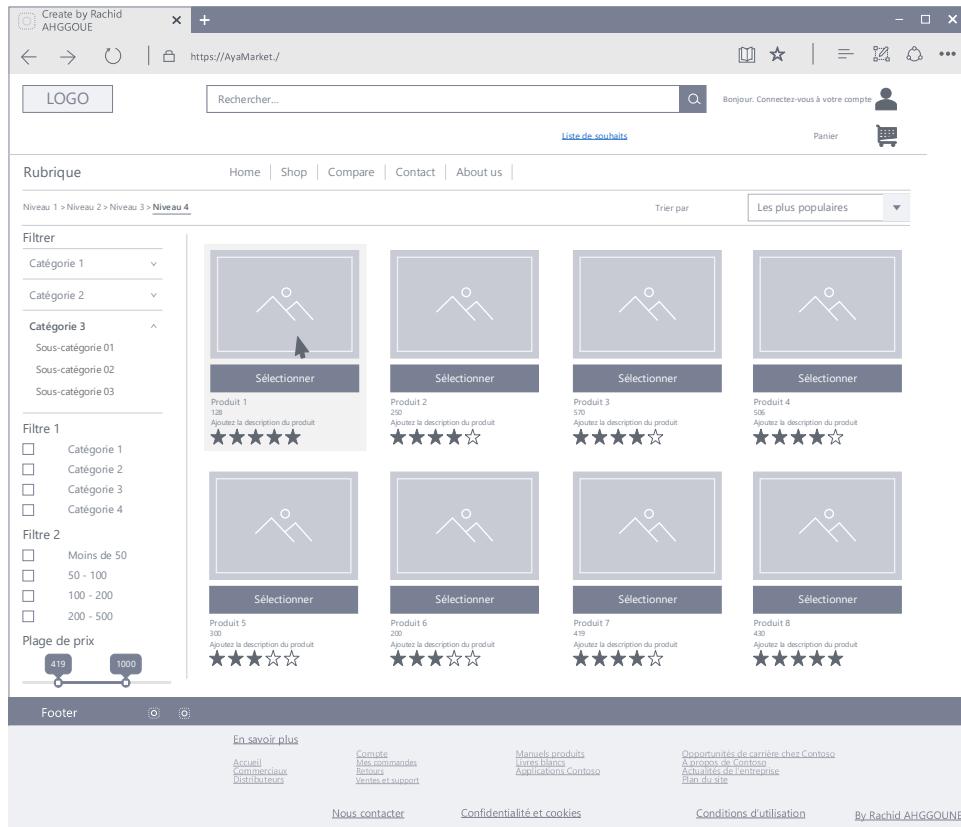


Figure 6 Maquette Home - produit

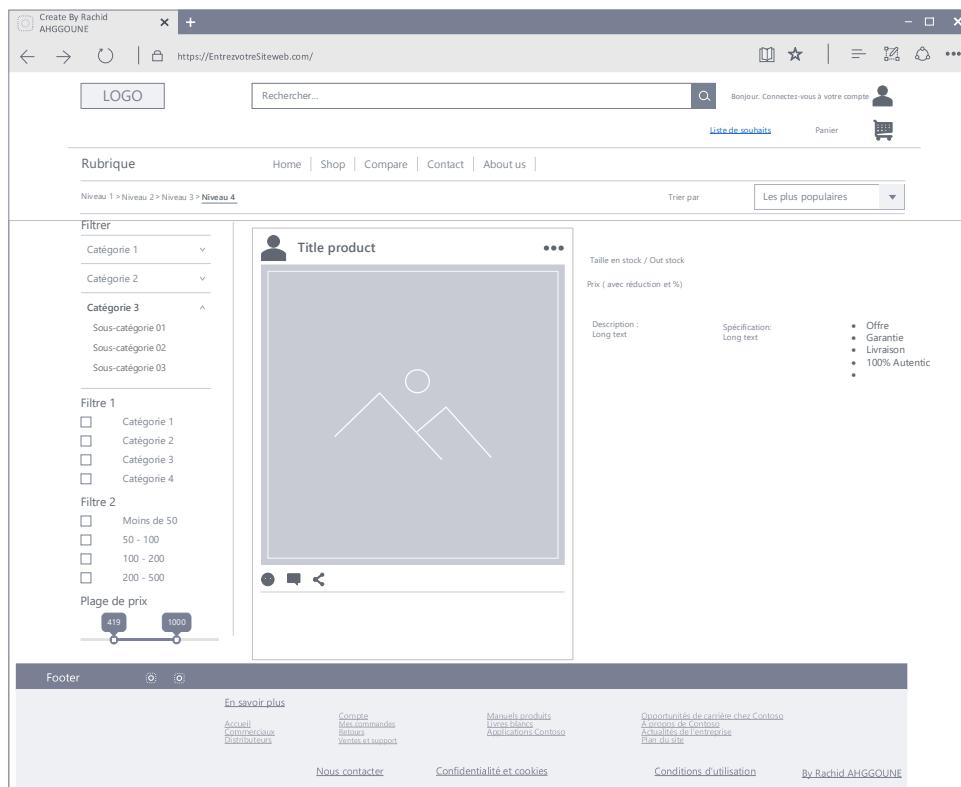


Figure 7 Maquette Product single

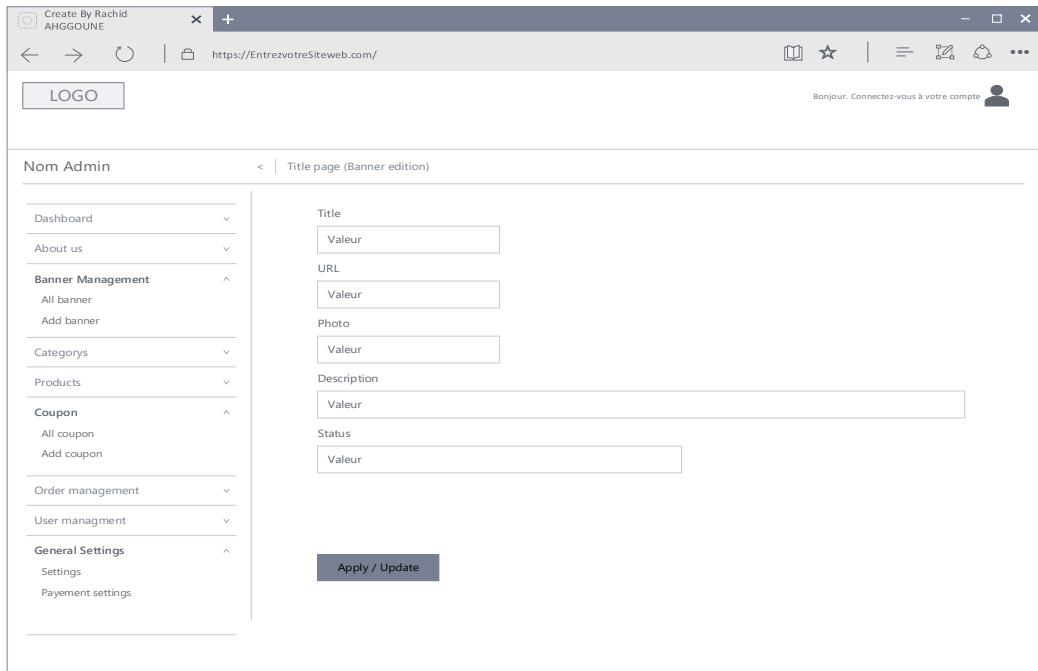


Figure 8 Maquette Admin Page Edit

S.N	Title	Description	Photo	Condition	Status	Action
1	Category 1	Women		PROMO	Active	<a href="#">Edit</a> <a href="#">Delete</a>
2	Category 2	Men		BANNER	Inactive	<a href="#">Edit</a> <a href="#">Delete</a>
3	Category 3	Child		BANNER	Active	<a href="#">Edit</a> <a href="#">Delete</a>
4	Category 4	All		PROMO	Inactive	<a href="#">Edit</a> <a href="#">Delete</a>

Figure 9 Maquette Admin page Liste

## 3.4 Conception détaillée (TTUP - Branche conception & dev)

### 3.4.1 Identification des acteurs

Un acteur représente un élément externe qui interagit avec le système, il peut s'agir d'un utilisateur ou un système tiers (autre programme, autre ordinateur, base de données).

Les acteurs qui interagissent avec le système sont :

Tableau 2 Identification des acteurs

Acteur	Description
Internaute	L'acteur peut consulter le contenu de l'application et contacter la société ;
Futur client	L'acteur hérite de l'internaute et veut créer un compte client ;
Client	L'acteur hérite de l'internaute et il est inscrit dans l'application, il peut passer des commandes et gérer les commandes et son compte ;
Administrateur	L'acteur responsable de la gestion du contenu de l'application ;
Paypal	L'acteur secondaire sollicité par le système pour les paiements.

### 3.4.2 Identification des cas d'utilisations

#### 3.4.2.1 Modèle global de cas d'utilisation

La figure suivante illustre le diagramme de cas d'utilisation globale (Back office & front), il nous permet d'obtenir une vision globale du comportement fonctionnel de l'application Web.

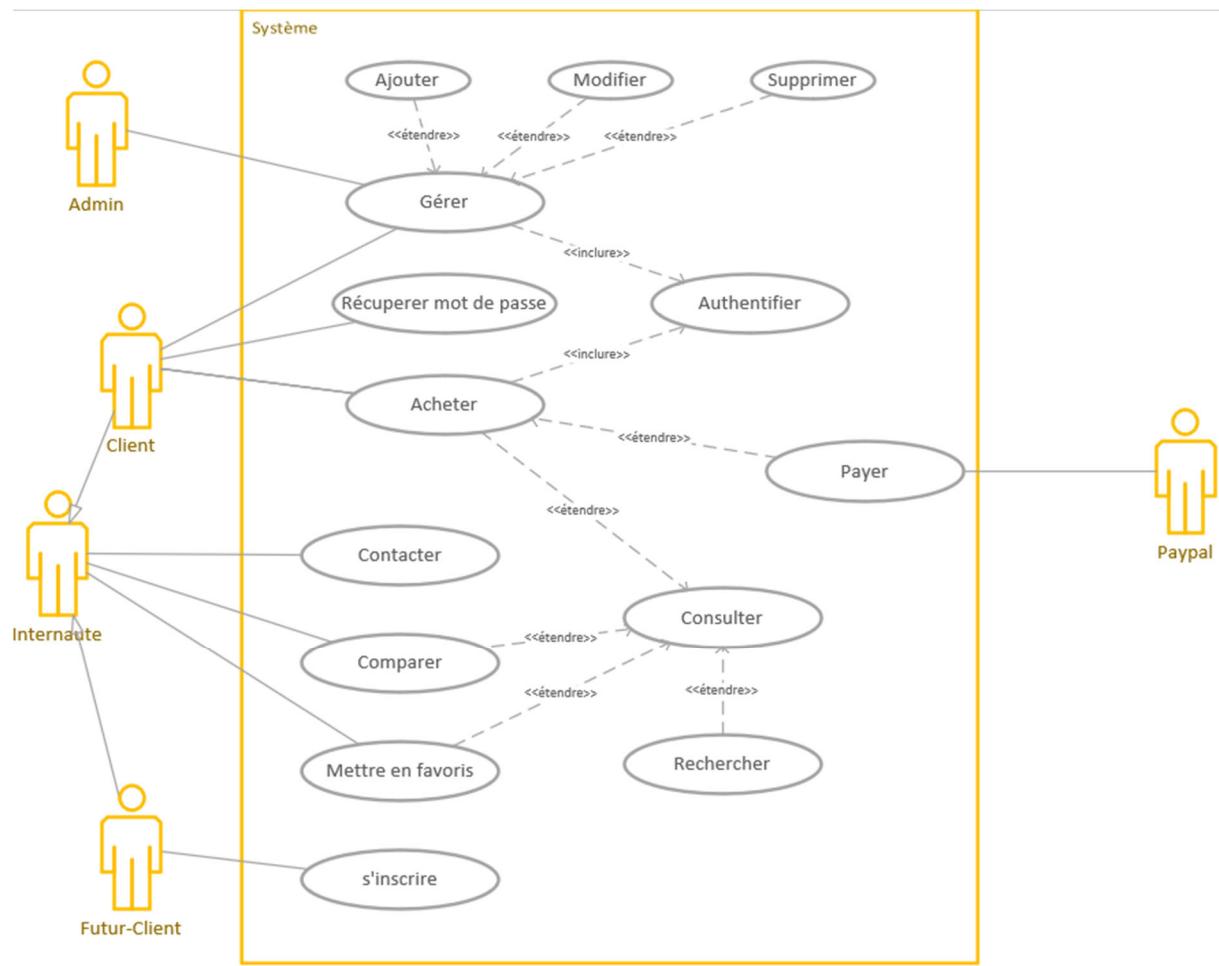


Figure 10 UseCase

#### 3.4.2.2 Description détaillée des cas d'utilisation

Pour mieux comprendre le fonctionnement des tâches faites par les acteurs, j'utilise les tableaux suivants.

Tableau 3 Use case (consulter)

Cas d'utilisation	Consulter
Acteurs	Client / admin
Objectifs	Il permet à l'acteur de parcourir le catalogue.
Préconditions	L'admin doit être authentifié (back-end)
Postcondition	Acteur consulte le catalogue
Scénario nominal	L'acteur ouvre l'application, Le système affiche le catalogue
Scénario alternatif	

Tableau 4 Use case (s'inscrire)

Cas d'utilisation	S'inscrire
Acteurs	Futur Client
Objectifs	Il permet au Futur Client de s'inscrire en remplissant le formulaire d'inscription
Préconditions	Consulter le site Web
Postcondition	- Futur Client inscrit
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le Futur Client ouvre l'application,</li> <li>2. Le système affiche la page d'authentification/inscription,</li> <li>3. Le Futur Client saisit les données dans le formulaire,</li> <li>4. Le système vérifie l'intégrité et validation les données,</li> <li>5. Le système enregistre les données dans la BD.</li> <li>6. Le système affiche la page d'accueil.</li> </ol>
Scénario alternatif	<p>A. Erreur d'inscription : un champ du formulaire non valide. Cet enchaînement démarre au point 4.</p> <p>5. Le système affiche un message d'erreur. Le scénario reprend au point 2.</p> <p>B. Champs obligatoires vides. Cet enchaînement démarre au point 4.</p> <p>5. Le système affiche un message d'erreur. Le scénario reprend au point 2.</p> <p>C. Conditions ne sont pas acceptés : bouton d'inscription invisible. D. E-mail existant. Cet enchaînement démarre au point 4.</p> <p>5. le système affiche message d'erreur. Le scénario reprend au point 2.</p>

Tableau 5 Use case (se connecter)

Cas d'utilisation	Authentifier
Acteurs	Client / admin
Objectifs	Il permet à l'acteur de s'identifier en saisissant son login et mot de passe.
Préconditions	L'acteur possède un compte
Postcondition	-Acteur authentifié.
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'acteur accède à l'application,</li> <li>2. Le système affiche la page d'authentification/inscription,</li> <li>3. L'acteur saisit le login et le mot de passe,</li> <li>4. Le système vérifie l'intégrité et la validation des données,</li> <li>5. Le système affiche la page d'accueil.</li> </ol>
Scénario alternatif	<p>A. Erreur d'authentification : login ou mot de passe non valide. Cet enchaînement démarre au point 4.</p> <p>5. Le système affiche un message d'erreur. Le scénario reprend au point 2.</p> <p>B. Champs obligatoires vides. Cet enchaînement démarre au point 4.</p> <p>Le scénario reprend au point 2.</p>

Tableau 6 Use case (gérer panier)

Cas d'utilisation	Gérer (panier)
Acteurs	Client
Objectifs	Il permet à l'acteur de contrôler son panier
Préconditions	L'acteur doit être authentifié. Le panier doit contenir au moins un produit.
Postcondition	Panier modifier
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'acteur clique sur le panier.</li> <li>2. Le système affiche la liste des produits dans le panier.</li> <li>3. Client choisi de supprimer le produit ou d'ajouter en quantité.</li> <li>4. Le système exécute la suppression, ou vérifie la disponibilité pour l'ajoute de la quantité.</li> </ol>
Scénario alternatif	A : Quantité en stock ne permet pas d'ajouter. <ul style="list-style-type: none"> <li>- Message d'erreur informe l'acteur.</li> </ul>

Tableau 7 Use case (Gérer compte)

Cas d'utilisation	Gérer (compte)
Acteurs	Client / admin
Objectifs	Il permet à l'acteur de gérer son compte : <ul style="list-style-type: none"> <li>- Modifier les adresses,</li> <li>- Modifier le mot de passe ;</li> <li>- Modifier les données personnelles ;</li> <li>- Consulter la liste des commandes ;</li> </ul>
Préconditions	L'acteur doit s'authentifier.
Postcondition	Gestion du compte
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'acteur ouvre l'application,</li> <li>2. Le système affiche la page de compte,</li> <li>3. L'acteur peut consulter ou modifier,</li> <li>4. Si l'acteur modifie les données alors Le système vérifie les données,</li> <li>5. Le système modifie sur la BD et affiche le compte avec les nouvelles données.</li> </ol>
Scénario alternatif	A. données entrées erronées : information non valide. Cet enchaînement démarre au point 4. <ol style="list-style-type: none"> <li>5. Le système affiche un message d'erreur.</li> </ol> Le scénario reprend au point 2.

Tableau 8 Use case (Acheter)

Cas d'utilisation	Acheter
Acteurs	Client
Objectifs	Il permet au Client d'ajouter des produits dans le panier et effectuer le paiement
Préconditions	Le stock du produit à acheter doit être > 0. Le stock du produit à acheter $\geq$ quantité commandée Le client doit s'authentifier.
Postcondition	- Article(s) acheté.
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le système affiche la page liste produits.</li> <li>2. Le client choisit un produit.</li> <li>3. Le système affiche la page du produit détaillé.</li> <li>4. Le client choisit la quantité et taille désirée.</li> <li>5. Le système vérifie la disponibilité et ajoute le produit au panier.</li> <li>6. Le client peut mettre une adresse de livraison.</li> <li>7. Le client confirme la commande.</li> </ol>
Scénario alternatif	<p>A : produit n'est pas disponible      Cet enchaînement démarre au point 4.      5. Le système affiche message d'erreur      Le scénario reprend au point 3.</p> <p>B : l'adresse n'est pas valide      Cet enchaînement démarre au point 6.      7. le système affiche message d'erreur.      Le scénario reprend au point 6.</p>

Tableau 9 Use case (Payer)

Cas d'utilisation	Payer
Acteurs	Client - Paypal
Objectifs	Il permet au client d'effectuer le paiement de la facture via le service de PayPal.
Préconditions	Le client doit s'authentifier ; Le client doit posséder un compte PayPal ; La commande doit être confirmée ;
Postcondition	- Paiement effectué.
Scénario nominal	<ol style="list-style-type: none"> <li>1) Le système affiche la page commande confirmée.</li> <li>2) le client clique sur le bouton payer.</li> <li>3) le système redirige vers l'application Paypal.</li> <li>4) Le client effectue le paiement sur le système PayPal.</li> <li>5) PayPal renvoie le statut de paiement</li> <li>6) le système affiche une page de fin de transaction avec le code de la commande.</li> </ol>
Scénario alternatif	<p>A : Paiement n'est pas effectué      Cet enchaînement démarre au point 2.</p> <p>3. le système affiche la page de fin de transaction avec message d'erreur      Le scénario reprend au point 1.</p>

Tableau 10 Use case (Contacter)

Cas d'utilisation	Contacter
Acteurs	Internaute
Objectifs	Il permet à l'Internaute de contacter les services AyaMarket
Préconditions	Accès au formulaire contact
Postcondition	Message envoyé
Scénario nominal	1. L'acteur ouvre l'application, 2. Le système affiche la page de contact, 3. L'acteur remplis le formulaire, 4. Le système vérifie les données introduites, 5. Le système envoie un mail au service AyaMarket
Scénario alternatif	A. données entrées erronées : information non valide. Cet enchaînement démarre au point 4. 5. Le système affiche un message d'erreur. Le scénario reprend au point 2.

Tableau 11 Use case (Ajouter)

Cas d'utilisation	Ajouter
Acteurs	Admin
Objectifs	Il permet à l'admin d'ajouter : informations boutique, produit, catégorie, banner, livraison, utilisateur et coupon
Préconditions	L'acteur doit être authentifié.
Postcondition	Élément ajouter
Scénario nominal	1. L'acteur ouvre l'application, 2. Le système affiche la page à de l'élément à ajouter, 3. L'acteur remplis le formulaire, 4. Le système vérifie les données introduites, 5. Le système ajoute dans la base de données.
Scénario alternatif	A. données entrées erronées : information non valide. Cet enchaînement démarre au point 4. 5. Le système affiche un message d'erreur. Le scénario reprend au point 2. B. Élément existe dans la BD (autre que celui à modifier) : Cet enchaînement démarre au point 5. 6. le système affiche message d'erreur Le scénario reprend au point 2.

Tableau 12 Use case (Modifier)

Cas d'utilisation	Modifier
Acteurs	Admin
Objectifs	Il permet à l'admin de modifier : informations boutique, produit, catégorie, banner, livraison, utilisateur et coupon
Préconditions	L'acteur doit être authentifié.
Postcondition	Élément modifier
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'acteur ouvre l'application,</li> <li>2. Le système affiche la page de l'élément à modifier,</li> <li>3. L'acteur remplis le formulaire,</li> <li>4. Le système vérifie les données introduites,</li> <li>5. Le système ajoute dans la base de données.</li> <li>6. Le système affiche un message de succès</li> </ol>
Scénario alternatif	<p>A. données entrées erronées : information non valide. Cet enchaînement démarre au point 4.</p> <p>5. Le système affiche un message d'erreur. Le scénario reprend au point 2.</p> <p>B. Élément existe dans la base de données : Cet enchaînement démarre au point 5.</p> <p>6. le système affiche message d'erreur Le scénario reprend au point 2.</p>

Tableau 13 Use case (Supprimer)

Cas d'utilisation	Supprimer
Acteurs	Admin
Objectifs	Il permet à l'admin de supprimer des éléments : produit, catégorie, banner, livraison, utilisateur et coupon
Préconditions	L'acteur doit être authentifié.
Postcondition	Élément supprimer
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'acteur ouvre l'application,</li> <li>2. Le système affiche la page de l'élément à supprimer (Produit, catégorie, utilisateur, coupon...)</li> <li>3. L'acteur supprimer l'élément,</li> <li>4. Le système supprime l'élément de la base de données.</li> <li>5. Le système affiche un message de succès</li> </ol>
Scénario alternatif	<p>A. l'élément n'existe plus : Cet enchaînement démarre au point 3.</p> <p>5. Le système affiche un message d'erreur. Le scénario reprend au point 2.</p>

Tableau 14 Use case Comparer

Cas d'utilisation	Comparer
Acteurs	Client
Objectifs	Il permet à l'acteur de comparer les produits
Préconditions	Ajouter des produits à la liste de comparaison.
Postcondition	Produits comparés.
Scénario nominal	1. L'acteur ouvre la page du produit, 2. Le système affiche la page du produit, 3. L'acteur clique sur le bouton ajouter à la liste comparaison, 4. Le système ajoute les produits sélectionnés à la liste comparaison.
Scénario alternatif	A. l'élément existe dans la liste à comparer : Cet enchaînement démarre au point 3. 4. Le système affiche un message d'erreur. Le scénario reprend au point 2.

Tableau 15 Use cas (Mettre en favoris)

Cas d'utilisation	Mettre en favoris
Acteurs	Client
Objectifs	Il permet à l'acteur de garder à l'œil certains produits.
Préconditions	Ajouter des produits à la liste de favorite.
Postcondition	
Scénario nominal	1. L'acteur ouvre l'application ; 2. Le système affiche la page d'accueil ; 3. L'acteur sélectionne les produits à ajouter à la liste favorite ; 4. Le système ajoute les produits sélectionnés.
Scénario alternatif	A. l'élément existe dans la liste de favoris: Cet enchaînement démarre au point 3. 4. Le système affiche un message d'erreur. Le scénario reprend au point 2.

### 3.4.3 Diagramme de séquence système

Les diagrammes de séquence sont le graphique qui représente l'interaction entre différents objets. Cette interaction est une séquence de messages entre les objets.

Je me limite à présenter trois diagrammes de séquences importantes dans mon projet. Le diagramme d'authentifier, de s'inscrire et de payer.

Authentifier : Un internaute demande l'authentification, le système renvoie le formulaire de connexion, l'acteur saisit les informations d'identification puis le système valide les informations et le client est renvoyé à la page d'accueil, sinon un message d'erreur dans la page de connexion.

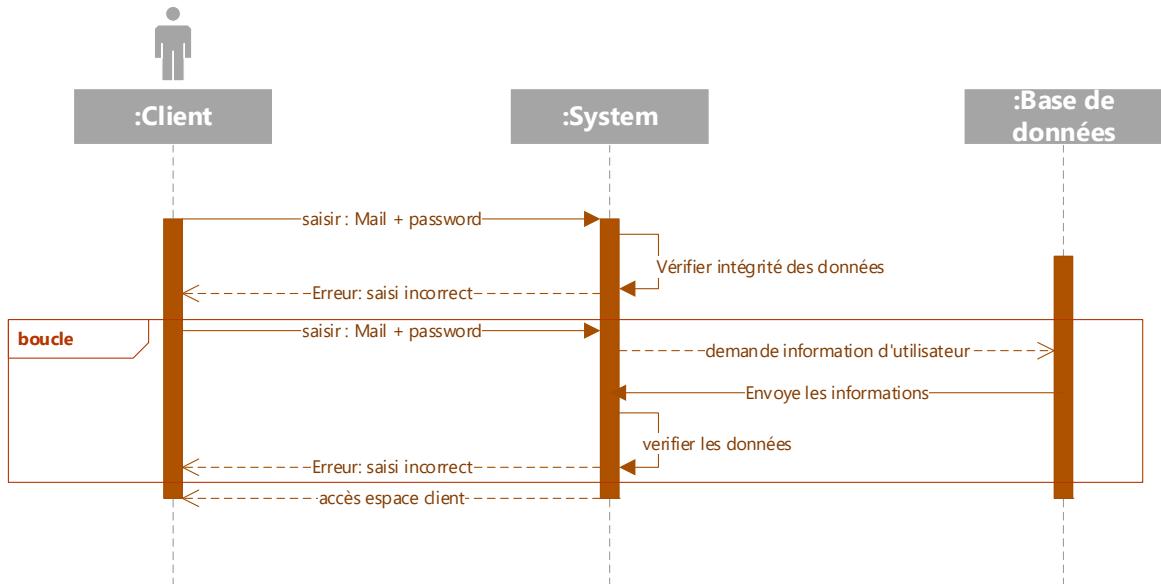


Figure 11 Diagramme de séquence (Authentifier)

**S'inscrire :** Le futur client rempli le formulaire d'inscription et renvoi au système qui vérifie les informations et valide, le système enregistre le client dans la base de données, puis renvoie le client dans la page d'accueil, si les informations ne sont pas valides, le système affiche les erreurs sur la page d'inscription.

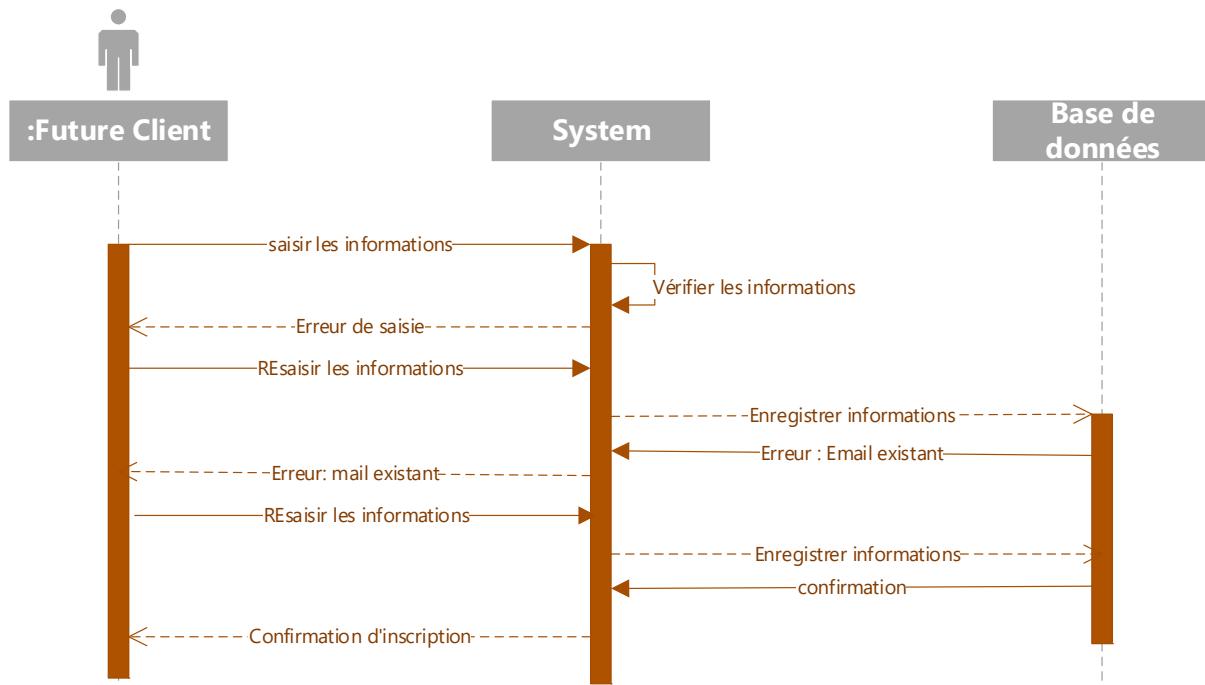


Figure 12 Diagramme de séquence (s'inscrire)

**Diagramme de séquence (payer) :** Le client passe à la page de paiement, il choisit le mode de paiement disponible, dans notre cas (PayPal uniquement), le système redirige le client sur la page PayPal qui demande l'authentification et la validation de sa commande, si tout se passe bien, PayPal redirige le client vers notre application avec une confirmation de paiement et le système enregistre la commande dans la base de données, affiche un message avec numéro de commande, le système vide

le panier et envoi un e-mail de confirmation. Sinon le système renvoi vers la page d'accueil sans vider le panier et aucune information supplémentaire afficher, car PayPal affiche déjà l'erreur au client.

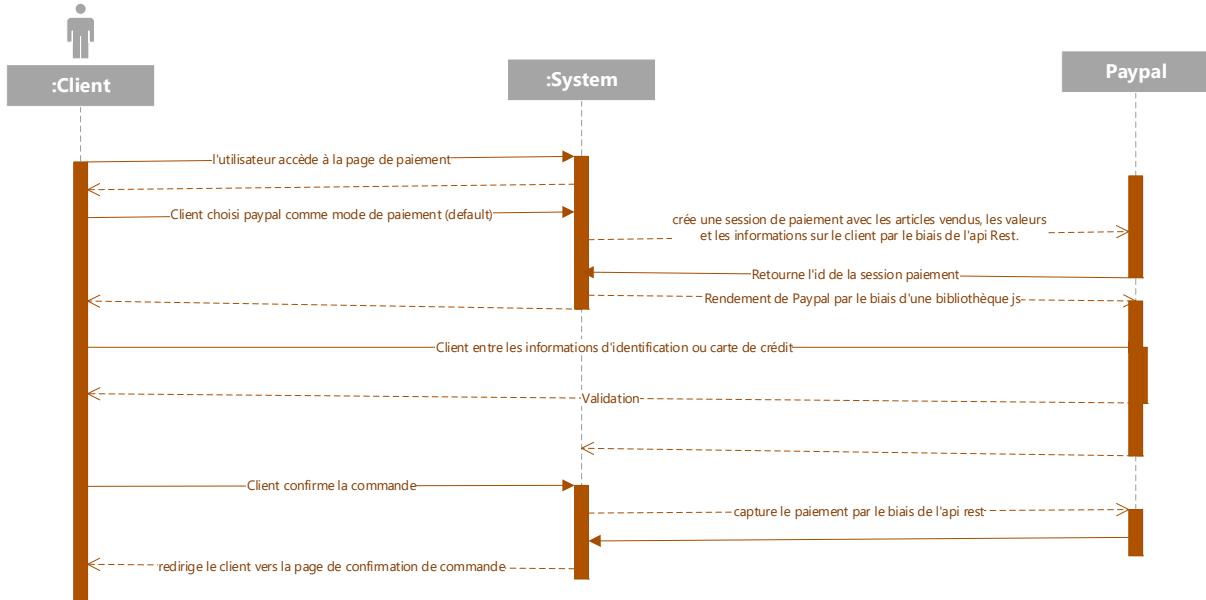


Figure 13 Diagramme de séquence (payer)

#### 3.4.4 Diagramme de composant

Le diagramme de composant est une présentation de l'architecture physique et statique de l'application Web, comme présenté dans le diagramme suivant :

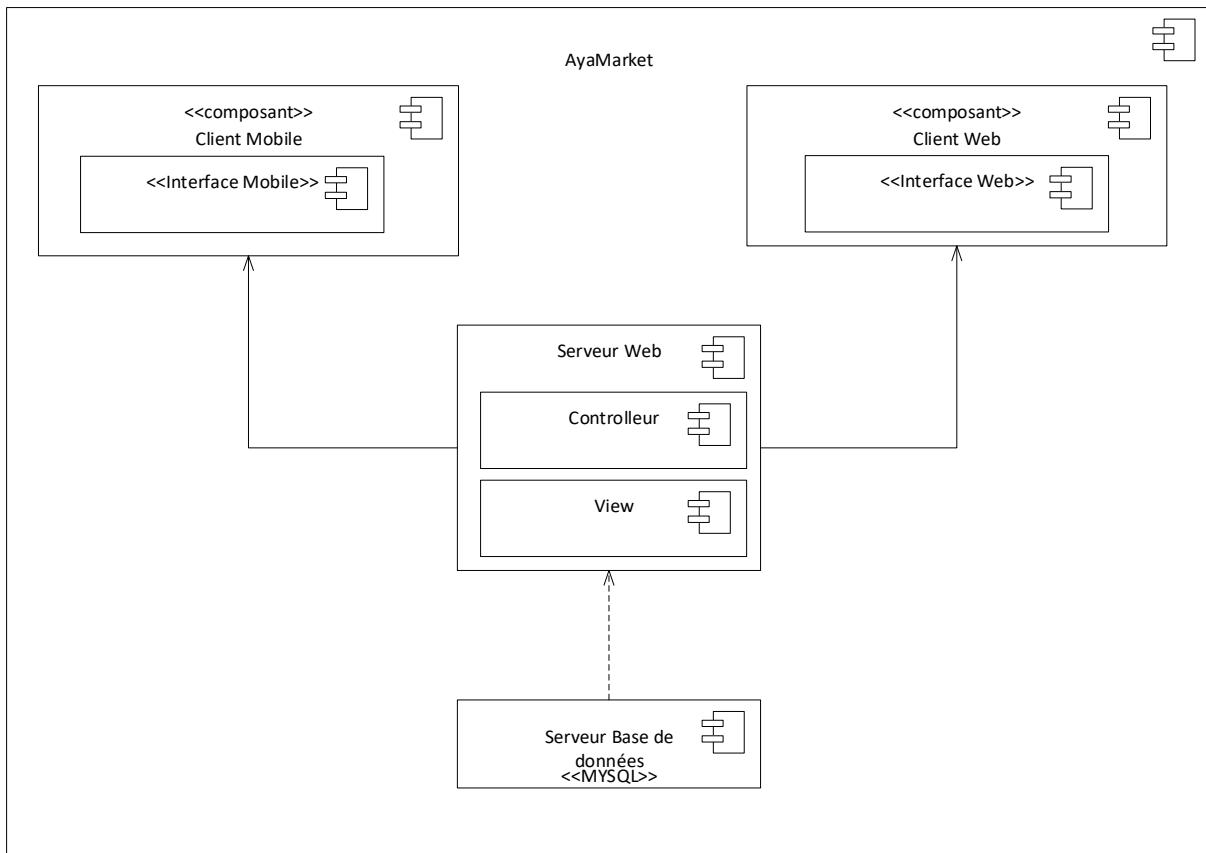


Figure 14 Diagramme de composant

Dans notre cas d'un model MVC On distingue trois couches:

- D'abord la couche présentation qui joue le rôle de VUE dans le model MVC avec l'interface mobile et l'interface web (coté responsive de l'application),
- Puis la couche logique métier qui joue le rôle Controller dans le model MVC qui décrit les objets, les classes du projet,
- et enfin, la couche persistance concerne la persistance des données ORM dans un serveur MySQL,

#### 3.4.5 Diagramme de classe

Le diagramme de classes est le plus important de la modélisation orientée objet, il 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 de classe est le seul obligatoire lors d'une telle modélisation. À partir de l'analyse et des diagrammes de séquence présente, nous avons extrait un ensemble d'entités et de dépendances, ça a été traduit par UML en un diagramme de classes. (diagramme-de-classes)<sup>4</sup>

<sup>4</sup> *diagramme-de-classes.* (s.d.). Récupéré sur ecole-en-ligne.blogspot: <http://ecole-en-ligne.blogspot.com/2016/02/diagramme-de-classe.html>

Pour décrire la structure d'un système en montrant le système de classes, leurs attributs, les opérations (méthodes) et les relations entre les classes, nous allons compter sur le diagramme de classes suivant :

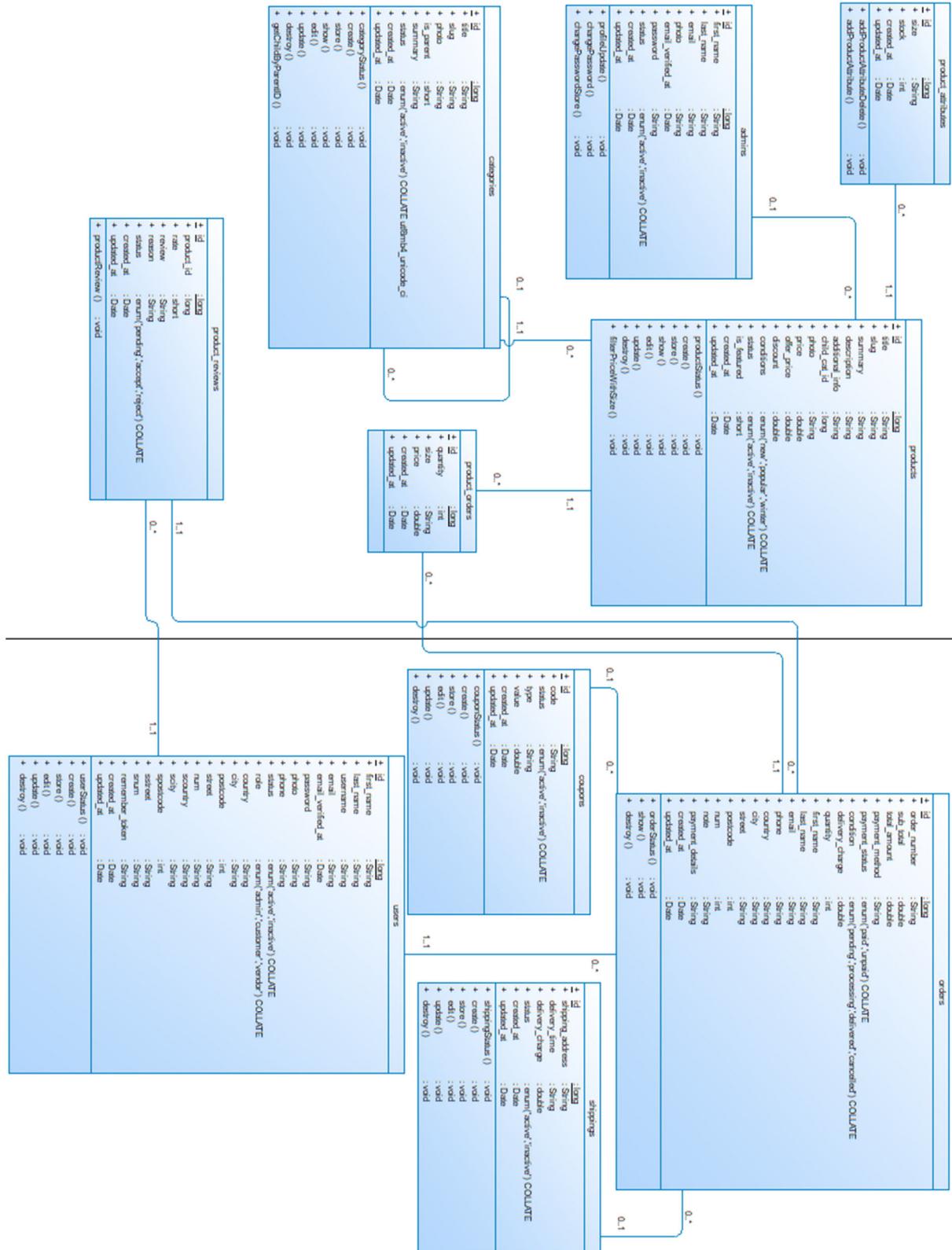


Figure 15 Diagramme de classe (1)

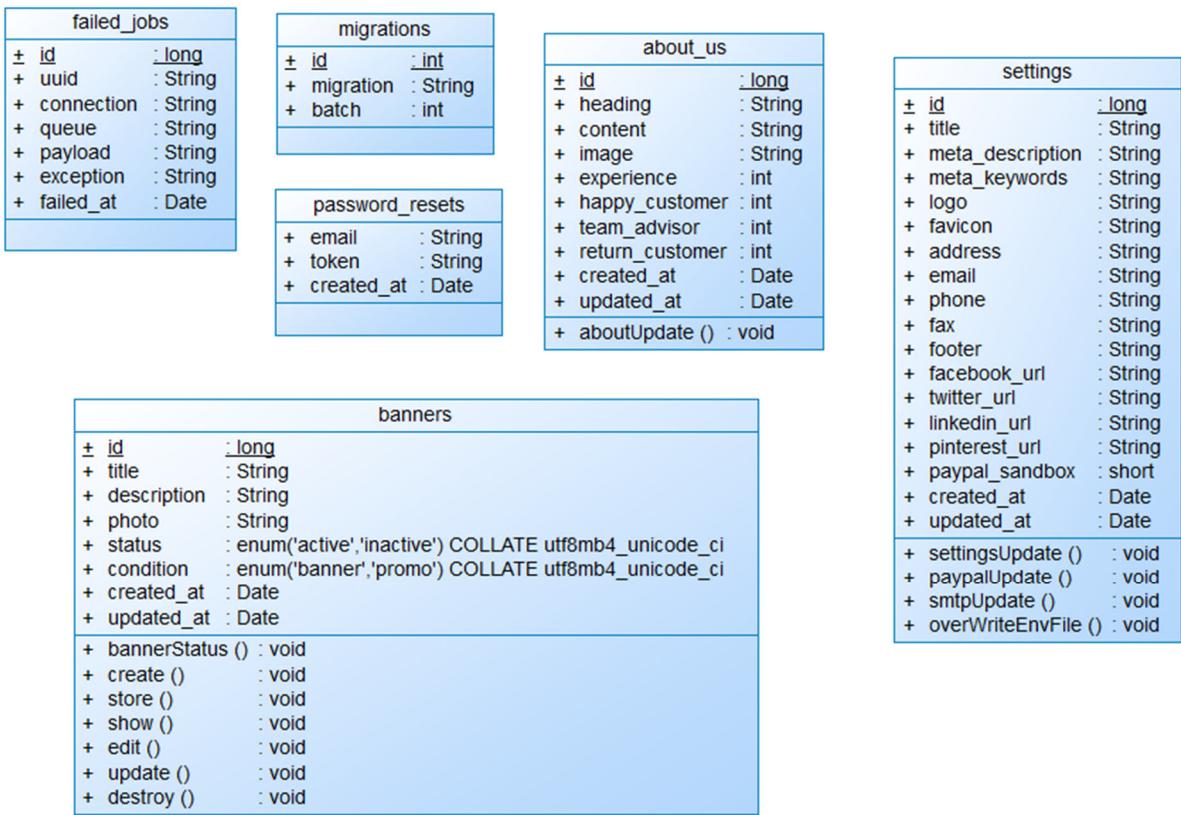


Figure 16 Diagramme de classe (2)

### 3.4.6 Base de données (MySQL)

La base de données de l'application a été créée en suivant les besoins fonctionnels et en respectant la troisième forme normale et les contraintes d'intégrité.

Notre base de données est constituée de tables suivantes :

1. **about\_us** : Informations spécifiques de l'entreprise ;
2. **admins** : Informations spécifiques des administrateurs de l'entreprise ;
3. **banners** : Informations spécifiques des bannières à publier ;
4. **categories** : Informations spécifiques des catégories de produit ;
5. **coupons** : Informations spécifiques des coupons de réduction ;
6. **orders** : Informations spécifiques des ventes (commande) passées ;
7. **password\_resets** : Informations spécifiques des PW à récupérer ;
8. **products** : Informations spécifiques des produits ;
9. **product\_attributes** : Informations spécifiques des détails de produit ;
10. **product\_orders** : Informations spécifiques des produits de la commande ;
11. **product\_reviews** : Informations spécifiques des commentaires de client ;
  - a. table pas utilisée
12. **settings** : Informations spécifiques des paramètres de l'application ;
13. **shippings** : Informations spécifiques des livraisons ;
14. **users** : Informations spécifiques des clients.

Tableau 16 Table About-us

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>heading</b>	text		Oui	NULL			
<b>content</b>	longtext		Oui	NULL			
<b>image</b>	varchar(255)		Oui	NULL			
<b>experience</b>	int(11)		Non	3			
<b>happy_customer</b>	int(11)		Non	500			
<b>team_advisor</b>	int(11)		Non	200			
<b>return_customer</b>	int(11)		Non	70			
<b>created_at</b>	timestamp		Oui	NULL			
<b>updated_at</b>	timestamp		Oui	NULL			

Tableau 17 Table Admins

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>first_name</b>	varchar(255)		Non				
<b>last_name</b>	varchar(255)		Non				
<b>E-mail</b>	varchar(255)		Non				
<b>Photo</b>	varchar(255)		Oui	NULL			
<b>email_verified_at</b>	Timestamp		Oui	NULL			
<b>password</b>	varchar (255)		Non				
<b>status</b>	enum('active', 'inactive')		Non	Active			
<b>created_at</b>	Timestamp		Oui	NULL			
<b>updated_at</b>	Timestamp		Oui	NULL			

Tableau 18 Table Banners

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
id	bigint(20)	UNSIGNED	Non		auto_increment		
title	varchar(255)		Non				
description	Mediumtext		Oui	NULL			
photo	varchar(255)		Non				
status	enum('active', 'inactive')		Non	inactive			
condition	enum('banner', 'promo')		Non	banner			
created_at	Timestamp		Oui	NULL			
updated_at	timestamp		Oui	NULL			

Tableau 19 Table Categories

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
id	bigint(20)	UNSIGNED	Non		auto_increment		
title	varchar(255)		Non				
slug	varchar(255)		Non				
photo	varchar(255)		Oui	NULL			
is_parent	tinyint(1)		Non	1			
summary	mediumtext		Oui	NULL			
parent_id	bigint(20)	UNSIGNED	Oui	NULL		-> categories.id ON UPDATE RESTRICT ON DELETE SET_NULL	
status	enum('active', 'inactive')		Non	active			
created_at	timestamp		Oui	NULL			
updated_at	timestamp		Oui	NULL			

Tableau 20 Table Password\_resets

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
e-mail	varchar(255)		Non				
token	varchar(255)		Non				
created_at	timestamp		Oui	NULL			

Tableau 21 Table Coupons

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>code</b>	varchar(255)		Non				
<b>status</b>	enum('active', 'inactive')		Non	active			
<b>type</b>	varchar(255)		Non	percent			
<b>value</b>	double(8,2)		Non	0.00			
<b>created_at</b>	timestamp		Oui	NULL			
<b>updated_at</b>	timestamp		Oui	NULL			

Tableau 22 Table Products

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>title</b>	varchar(255)		Non				
<b>slug</b>	varchar(255)		Non				
<b>summary</b>	Mediumtext		Non				
<b>description</b>	Longtext		Oui	NULL			
<b>additional_info</b>	Longtext		Oui	NULL			
<b>cat_id</b>	bigint(20)	UNSIGNED	Non			-> categories.id ON UPDATE RESTRICT ON DELETE CASCADE	
<b>child_cat_id</b>	bigint(20)	UNSIGNED	Oui	NULL		-> categories.id ON UPDATE RESTRICT ON DELETE SET_NULL	
<b>photo</b>	varchar(255)		Non				
<b>price</b>	double(8,2)		Non	0.00			
<b>offer_price</b>	double(8,2)		Non	0.00			
<b>discount</b>	double(8,2)		Non	0.00			
<b>conditions</b>	enum('new', 'popular', 'winter')		Non	new			
<b>status</b>	enum('active', 'inactive')		Non	active			
<b>added_by</b>	varchar(255)		Oui	NULL			
<b>is_featured</b>	tinyint(1)		Oui	0			
<b>created_at</b>	timestamp		Oui	NULL			
<b>updated_at</b>	timestamp		Oui	NULL			

Tableau 23 Table Orders

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>user_id</b>	bigint(20)	UNSIGNED	Non				
<b>order_number</b>	varchar(10)		Non				
<b>sub_total</b>	double(8,2)		Non	0.00			
<b>total_amount</b>	double(8,2)		Non	0.00			
<b>Coupon</b>	double(8,2)		Oui	0.00			
<b>payment_method</b>	varchar(255)		Non	cod			
<b>payment_status</b>	enum('paid', 'unpaid')		Non	unpaid			
<b>Condition</b>	enum('pending', 'processing', 'delivered', 'cancelled')		Non	pending			
<b>delivery_charge</b>	double(8,2)		Oui	0.00			
<b>Shipping</b>	varchar(255)		Non	Bpost			
<b>Quantity</b>	int(11)		Non	0			
<b>first_name</b>	varchar(255)		Non				
<b>last_name</b>	varchar(255)		Non				
<b>E-mail</b>	varchar(255)		Non				
<b>Phone</b>	varchar(255)		Non				
<b>Country</b>	varchar(255)		Non				
<b>City</b>	varchar(255)		Non				
<b>street</b>	varchar(255)		Non				
<b>postcode</b>	int(11)		Non				
<b>Num</b>	int(11)		Non				
<b>Note</b>	Mediumtext		Oui	NULL			
<b>payment_details</b>	Text		Oui	NULL			
<b>created_at</b>	Timestamp		Oui	NULL			
<b>updated_at</b>	Timestamp		Oui	NULL			

Tableau 24 Table Product\_attributes

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>product_id</b>	bigint(20)	UNSIGNED	Non			-> products.id ON UPDATE RESTRICT ON DELETE CASCADE	
<b>size</b>	varchar(255)		Non				
<b>stock</b>	int(11)		Non	0			
<b>created_at</b>	timestamp		Oui	NULL			
<b>updated_at</b>	timestamp		Oui	NULL			

Tableau 25 Table Product\_orders

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>product_id</b>	bigint(20)	UNSIGNED	Non			-> products.id ON UPDATE RESTRICT ON DELETE CASCADE	
<b>order_id</b>	bigint(20)	UNSIGNED	Non			-> orders.id ON UPDATE RESTRICT ON DELETE CASCADE	
<b>quantity</b>	int(11)		Non	0			
<b>size</b>	varchar(255)		Oui	NULL			
<b>price</b>	decimal(8,2)		Non	0.00			
<b>created_at</b>	timestamp		Oui	NULL			
<b>updated_at</b>	timestamp		Oui	NULL			

Tableau 26 Table Shippings

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>shipping_address</b>	varchar(255)		Non				
<b>delivery_time</b>	varchar(255)		Non				
<b>delivery_charge</b>	double(8,2)		Non	0.00			
<b>status</b>	enum('active', 'inactive')		Non	active			
<b>created_at</b>	Timestamp		Oui	NULL			
<b>updated_at</b>	Timestamp		Oui	NULL			

Tableau 27 table product\_reviews

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>user_id</b>	bigint(20)	UNSIGNED	Non			-> users.id ON UPDATE RESTRICT ON DELETE CASCADE	
<b>product_id</b>	bigint(20)	UNSIGNED	Non			-> products.id ON UPDATE RESTRICT ON DELETE CASCADE	
<b>rate</b>	tinyint(4)		Non	0			
<b>review</b>	text		Oui	NULL			
<b>reason</b>	varchar(255)		Oui	NULL			
<b>status</b>	enum('pending', 'accept', 'reject')		Non	accept			
<b>created_at</b>	timestamp		Oui	NULL			
<b>updated_at</b>	timestamp		Oui	NULL			

Tableau 28 Table SETTINGS

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>title</b>	varchar(255)		Non				
<b>meta_description</b>	Text		Oui	NULL			
<b>meta_keywords</b>	Text		Oui	NULL			
<b>logo</b>	varchar(255)		Oui	NULL			
<b>favicon</b>	varchar(255)		Oui	NULL			
<b>address</b>	varchar(255)		Oui	NULL			
<b>e-mail</b>	varchar(255)		Oui	NULL			
<b>phone</b>	varchar(255)		Oui	NULL			
<b>fax</b>	varchar(255)		Oui	NULL			
<b>footer</b>	varchar(255)		Oui	NULL			
<b>facebook_url</b>	varchar(255)		Oui	NULL			
<b>twitter_url</b>	varchar(255)		Oui	NULL			

<b>linkedin_url</b>	varchar(255)		Oui	NULL			
<b>pinterest_url</b>	varchar(255)		Oui	NULL			
<b>paypal_sand_box</b>	tinyint(1)		Non	1			
<b>created_at</b>	timestamp		Oui	NULL			
<b>updated_at</b>	timestamp		Oui	NULL			

Tableau 29 Table USERS

Colonne	Type	Attributs	Null	Valeur par défaut	Extra	Est relié à	MIME
<b>id</b>	bigint(20)	UNSIGNED	Non		auto_increment		
<b>first_name</b>	varchar(255)		Non				
<b>last_name</b>	varchar(255)		Non				
<b>username</b>	varchar(255)		Oui	NULL			
<b>email</b>	varchar(255)		Non				
<b>email_verified_at</b>	Timestamp		Oui	NULL			
<b>password</b>	varchar(255)		Non				
<b>photo</b>	varchar(255)		Oui	NULL			
<b>phone</b>	varchar(255)		Oui	NULL			
<b>status</b>	enum('active', 'inactive')		Non	active			
<b>role</b>	enum('admin', 'customer', 'vendor')		Non	customer			
<b>country</b>	varchar(255)		Oui	NULL			
<b>city</b>	varchar(255)		Oui	NULL			
<b>postcode</b>	int(11)		Oui	NULL			
<b>street</b>	varchar(255)		Oui	NULL			
<b>num</b>	varchar(255)		Oui	NULL			
<b>scountry</b>	varchar(255)		Oui	NULL			
<b>scity</b>	varchar(255)		Oui	NULL			
<b>spostcode</b>	int(11)		Oui	NULL			
<b>sstreet</b>	varchar(255)		Oui	NULL			
<b>snum</b>	varchar(255)		Oui	NULL			
<b>remember_token</b>	varchar(100)		Oui	NULL			
<b>created_at</b>	Timestamp		Oui	NULL			
<b>updated_at</b>	Timestamp		Oui	NULL			

### 3.5 Schéma relationnel

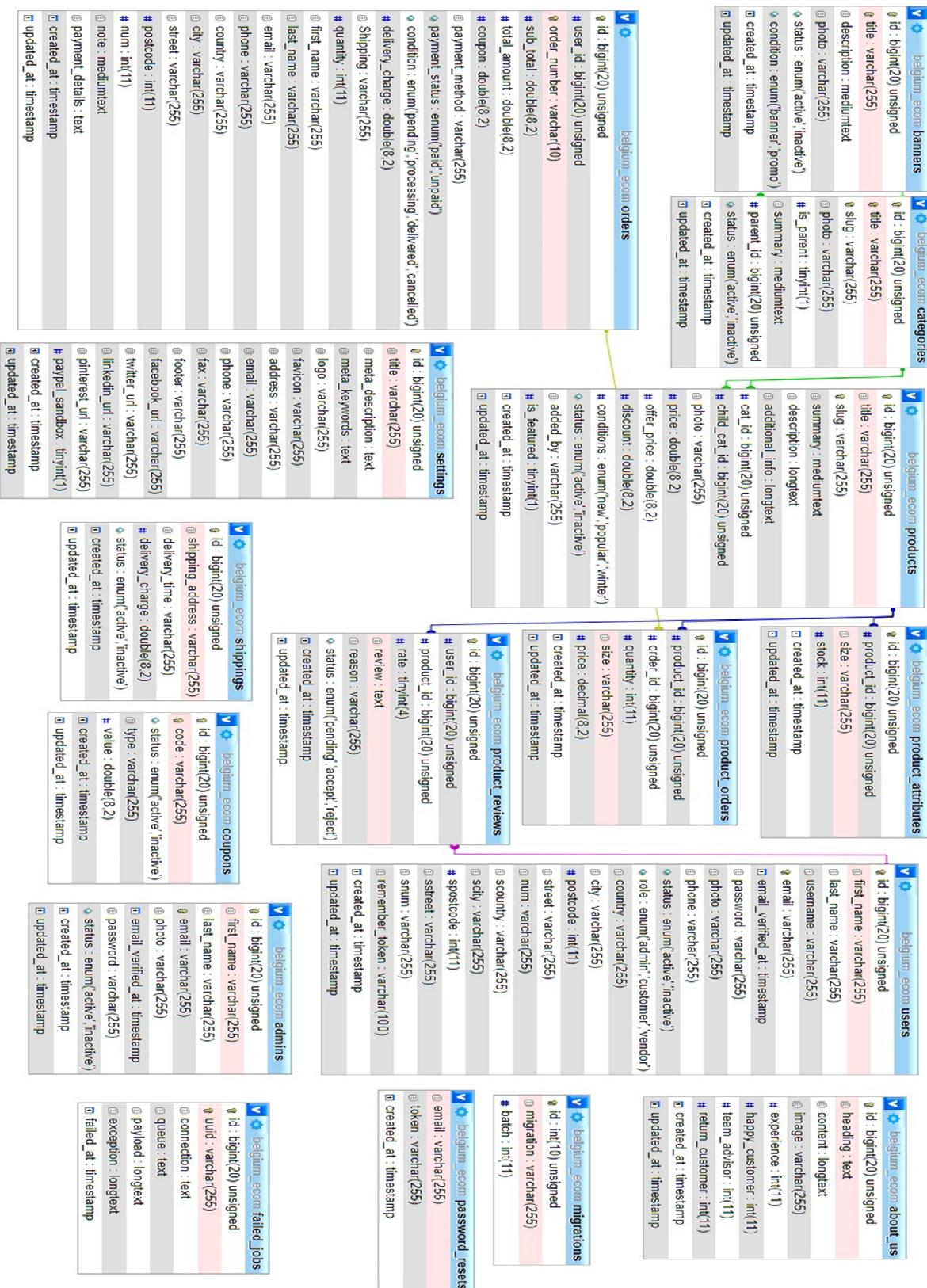


Figure 17 Base de données relationnelle

### 3.6 Les Contrôleurs

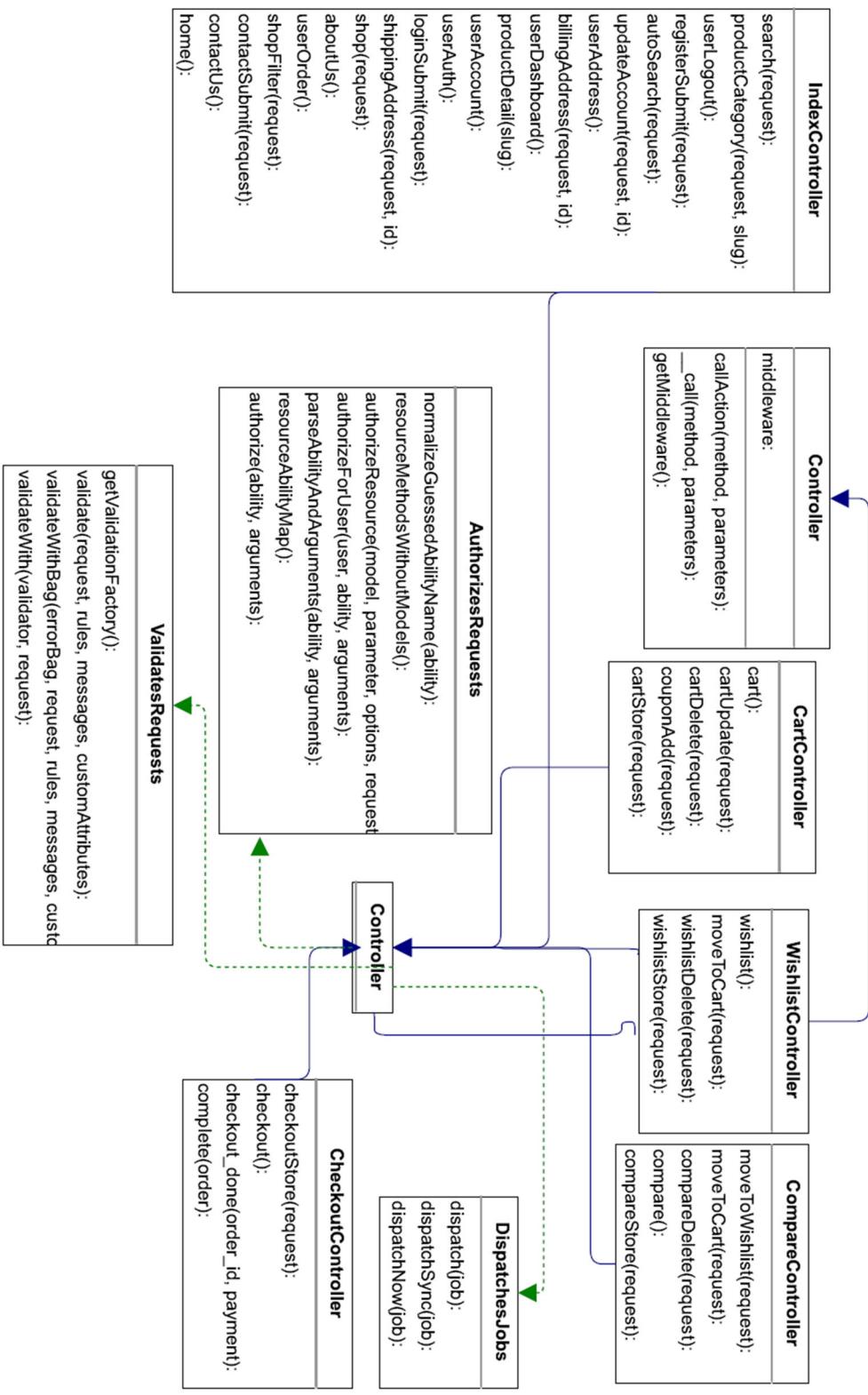


Figure 18 Contrôleurs Frontend

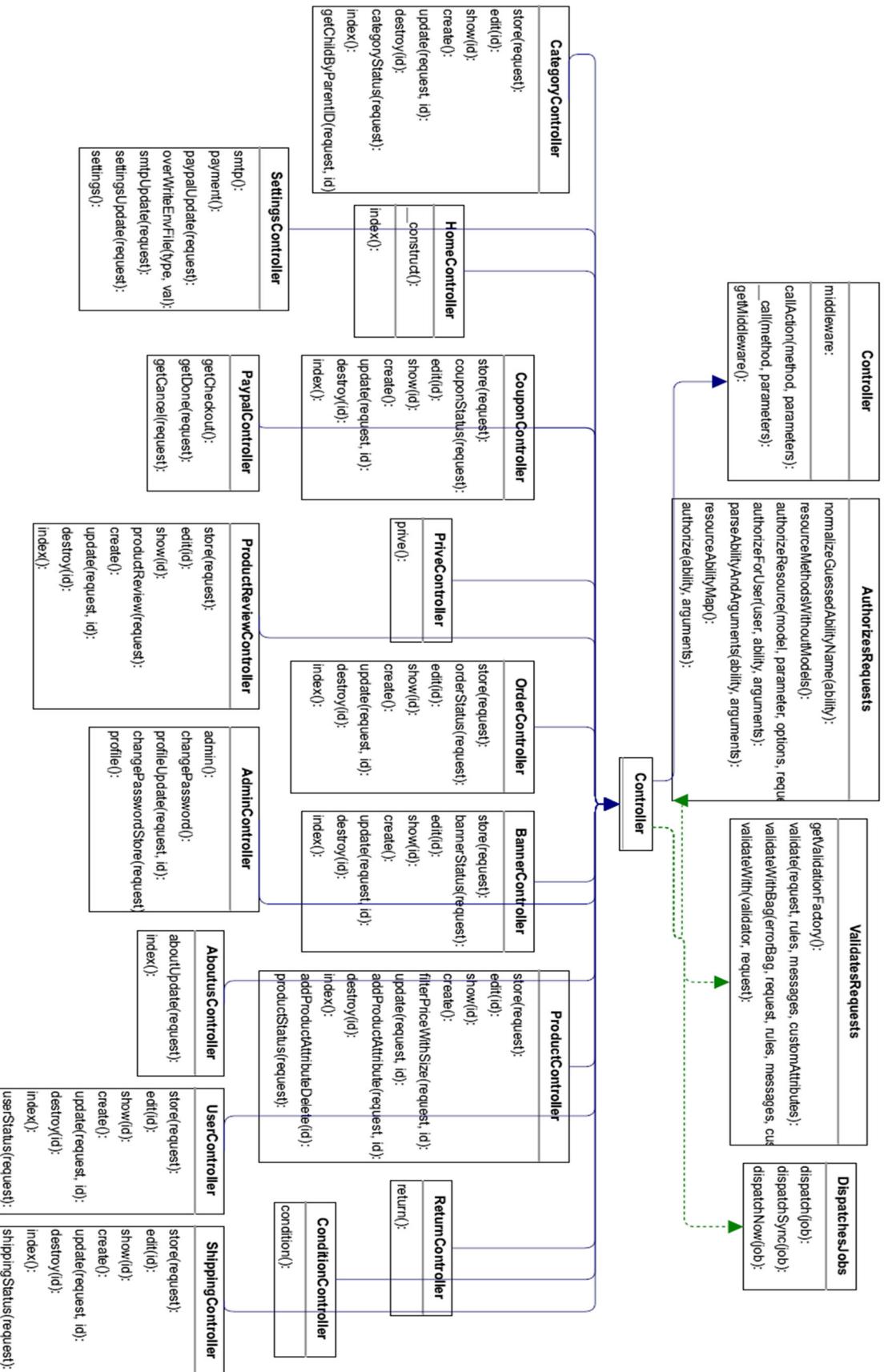


Figure 19 Contrôleurs Backend

### 3.7 Conclusion

À ce stade, j'ai abordé les différents diagrammes UML tels que les diagrammes de cas d'utilisation et les diagrammes de séquences permettant de cerner le fonctionnement de notre application Web et déterminer les interactions entre les acteurs et le système.

Ensuite, j'ai déterminé les différentes entités et associations qui ont généré le diagramme de classes. À la fin, j'ai présenté le modèle relationnel construit, structure logique et physique.

Le prochain Chapitre consistera donc en la réalisation des différents objets et des différentes fonctionnalités qui constituent le modèle de la solution que j'ai proposé.

## 4 Développement

---

### 4.1 Introduction

Le développement consistait à réaliser l'application en elle-même, c'est-à-dire en utilisant des technologies réelles et concrètes et non plus des modèles ou représentations abstraites qui sont non-exécutables. Ce travail s'appuyait massivement sur les étapes précédentes. le développement était complexe dans le sens où il fallait travailler avec toutes les contraintes de l'application et toutes les contraintes des environnements d'exécution. Il était donc important d'avoir préparé au maximum le travail en amont, de manière à perdre le moins de temps possible.

### 4.2 Environnement de développement

#### 4.2.1 Environnement matériel

Afin de réaliser mon projet, je vais présenter ici le matériel utilisé :

- Un ordinateur :
  - o Marque : Lenovo ;
  - o Processeur : Intel I7 ;
  - o RAM : 32 Go ;
  - o SSD : 256Go ;
  - o OS : Windows 11 ;
- Disque dur externe SSD : 512Go.

#### 4.2.2 Environnement logiciel

je présente dans cette section l'environnement logiciel performant mis à disposition.



- PHPStorm est un éditeur pour PHP, HTML, CSS et JavaScript, édités par JetBrains. C'est un IDE complet misant sur la productivité avec des systèmes d'auto-complétions intelligentes, d'analyse de code en temps réel. Avec sa gestion de versionning et la base de données, ça centralise tout le travail du développement. Je l'utilise en version étudiant qui est gratuite.



- 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 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. (XAMPP, 2022)<sup>5</sup>

---

<sup>5</sup> XAMPP. Consulté le 11 01, 2021, sur Wikipédia



- MailTrap est un faux serveur SMTP, le but d'un outil comme Mailtrap.io (Freemium) et de vous permettre de récupérer sur une interface Web, le contenu de l'e-mail que vous avez expédié depuis votre site. Ainsi vous ne risquez pas, par erreur, d'expédier un e-mail sur un client, abonné, réel. (@michaeldumontet, 2020)<sup>6</sup>.

### 4.3 Développement application Web (TTUP - Branche conception & dev)

Le développement de mon application a suivi un programme prédéfini, dont la découpe de l'ensemble du travail en plusieurs étapes, qui sera détaillé ci-après.

#### 4.3.1 Crédation du projet Laravel

La création du projet fut la première étape, pour cela je devais préparer l'environnement :

- Installation Xampp v 3.0 ;
- Installation de la bonne version PHP et configuration **php.ini** ;
- Installation de composer : gestionnaire de dépendances de PHP ;
- Création de la base donnée vierge ;
- Création de dossiers du projet (**c:\xampp\htdocs\tfe**) ;
- Création du projet LARAVEL avec la commande **Laravel new ayamarket** ;

Ici, nous avons un environnement de travail complet et parfait pour construire une application Web e-commerce.

### 4.4 La solution Web

#### 4.4.1 Interface d'authentification Admin

La phase d'authentification importante de mon projet, j'ai utilisé le « *Scaffolding front end* » proposé par laravel. Quelques étapes pour mettre le système en place :

- Étape 1 : **composer requires laravel/ui** : cette commande installe le laravel/ui qui est un package composé. Elle met en place le système complet de l'authentification (Login/registe) avec la migration pour la table users incluant les champs minimums requis (name, email, email\_verified-at, password, rememberToken et timestamps) et aussi une migration pour la table password.
- Étape 2 : L'exécution de la commande **ui artisan** générer un « *Scaffolding* » de base pour Bootstrap avec la commande artisan suivante :  
**PHP artisan ui bootstrap** cela créera un répertoire de composants dans le répertoire **resources/js**.

---

<sup>6</sup> @michaeldumontet. (2020, 12 3). *mailtrap cest quoi*. Consulté le 02 03, 2022, sur www.eewee.fr: <https://www.eewee.fr/mailtrap-cest-quoi/>

- Étape 3 : Pour générer les fichiers de vue pour l'authentification (login/register) il faut exécuter la commande suivante : `PHP artisan ui bootstrap --auth`, cela créera un répertoire de composants dans le répertoire `resources/js` et créera également un répertoire `auth` et `layouts` avec un fichier `home.blade.php` dans le répertoire `resources/views`.
- Étape 4 : nous devons par la suite installer le npm et exécuter la commande npm pour compiler « *Scaffolding* » avec la commande suivante : `npm install && npm run dev`.
- Étape 5 : pour finir l'installation il faut exécuter Mix pour surveiller et recompiler automatiquement les composants chaque fois qu'ils sont modifiés avec la commande : `npm run watch`

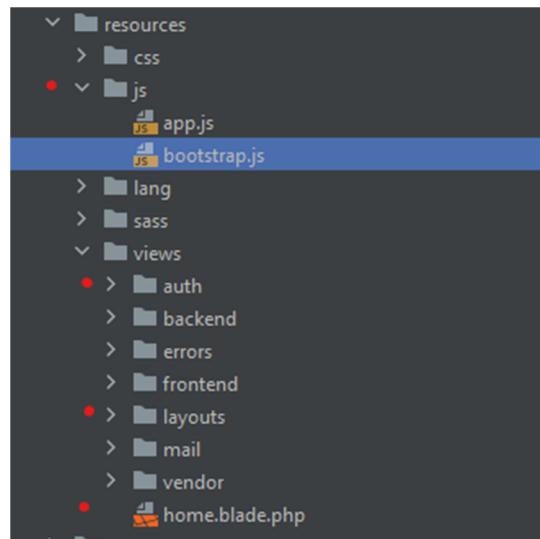


Figure 20 Composants ui, --auth

Maintenant que tout est mis en place, nous pouvons modifier les tables crée via les fichiers migrations pour répondre à nos besoins, puis crée le fichier seeder `UsersTableSeeder.php`, qui donne la possibilité d'introduire des données dans la base de données et le fichier modèle `Users.php`, qui crée le modèle de la table Users dans la base de données, avec la commande : `PHP artisan make :seeder UsersTableSeeder`. J'ai modifié préalablement les deux fichiers d'après le besoin, ensuite envoyé tout à la base de données avec la commande : `PHP artisan migrate`.

Avec ces étapes, j'ai créé le système d'authentification Admin dans l'application Web. La figure suivante présente la vue de l'interface login admin que j'ai réalisé.

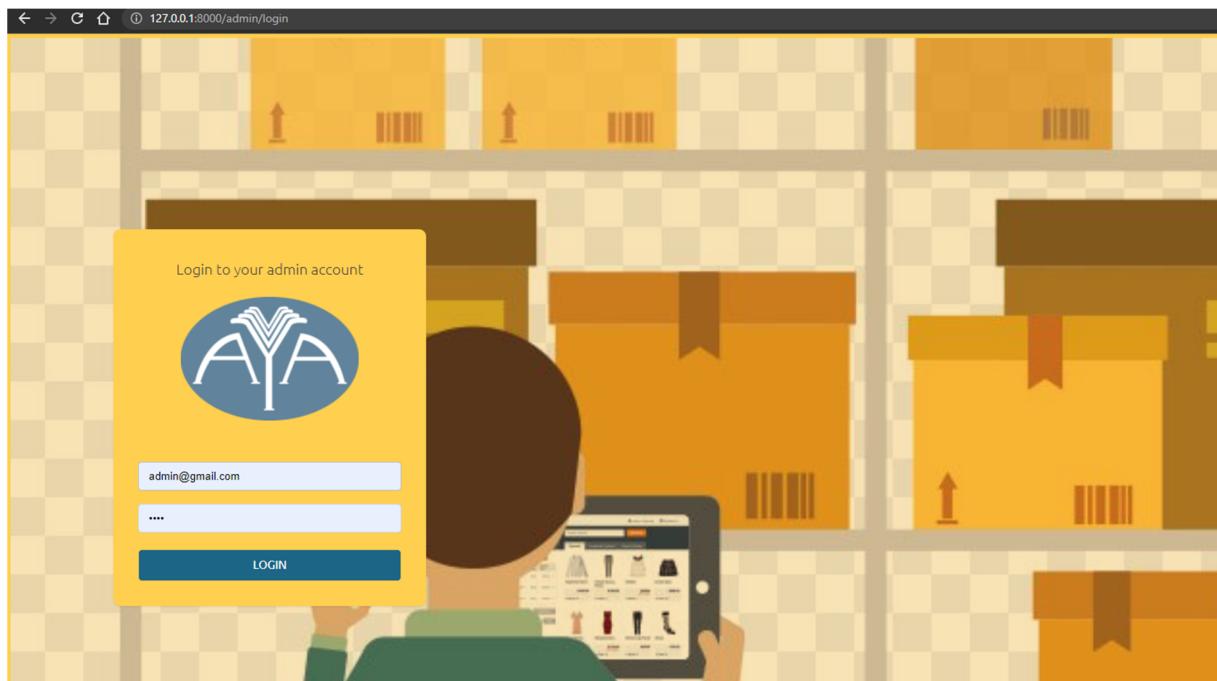


Figure 21 Interface Login Admin

#### 4.4.2 Interface backend

Sur cette interface l'administrateur de l'application pourra visualiser dans le Dashboard toutes les nouvelles du jour comme le nombre de nouvelles inscriptions, le total des ventes et garde un œil sur le nombre de catégories et produits disponible dans le catalogue.

La figure qui suit présente la page d'accueil de l'administration. Qui reprend un Dashboard résumant les chiffres importants à suivre pour le business et la liste des derniers commandes.

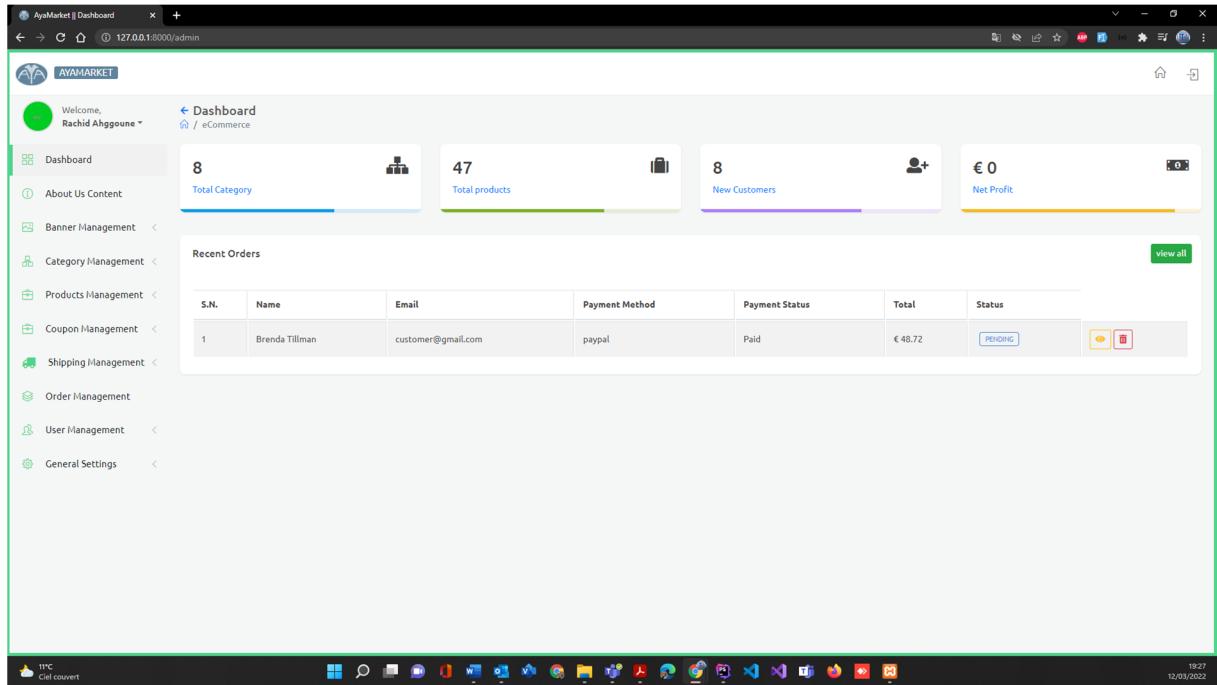


Figure 22 Interface accueil Backend

#### 4.4.3 Interface gestion des produits

L'interface de gestion du produit permet de gérer les produits ; l'administrateur seul peut ajouter, modifier ou supprimer un produit,

La figure suivante présente la vue (Products) qui affiche les produits présents dans la base de données ainsi que les options possibles pour l'administrateur.

The screenshot shows the AyAMarket product management interface. The left sidebar includes links for Dashboard, About Us Content, Banner Management, Category Management (with Products Management selected), Coupon Management, Shipping Management, Order Management, User Management, and General Settings. The main area displays a 'Product List' table with the following data:

S.N.	Title	Category	Photo	Price	Discount	Condition	Status	Actions
1	dolorem	Women Collection		€ 632.00	28%	<span>NEW</span>	<span>Inactive</span>	
2	dolor	nihil		€ 972.00	28%	<span>POPULAR</span>	<span>active</span>	
3	error	atque		€ 355.00	93%	<span>POPULAR</span>	<span>active</span>	
4	qui	maiores		€ 561.00	9%	<span>NEW</span>	<span>active</span>	
5	nam	aut		€ 194.00	39%	<span>WINTER</span>	<span>active</span>	

Figure 23 Interface de gestion de produits

#### 4.4.4 Interface de gestion des utilisateurs

Dans cette interface, l'administrateur de l'application pourra consulter certains détails des clients inscrit et également procéder à la modification des données à la demande du client depuis le formulaire de contact. L'administrateur peut aussi supprimer le client de la base de données.

La figure suivante présente la vue (Users) qui affiche à l'administrateur la liste des utilisateurs enregistrés avec les options possibles.

The screenshot shows the AyAMarket user management interface. The left sidebar includes links for Dashboard, About Us Content, Banner Management, Category Management (with User Management selected), Coupon Management, Shipping Management, Order Management, Products Management, and General Settings. The main area displays a 'User List' table with the following data:

S.N.	Photo	Full name	Email	Status	Actions
1		Rose Kuhn Patrick Hane	cornelius.windler@example.org	<span>active</span>	
2		Brooks Bailey I Ms. Dorothea Kunze IV	mrath@example.net	<span>Inactive</span>	
3		Abner Beier Jerod Funk	riley.weissnat@example.net	<span>active</span>	
4		Maritza Fadel Elda Quigley	justen10@example.org	<span>active</span>	
5		Darwin Considine Narciso Morar	wilkinson.rodrigo@example.com	<span>Inactive</span>	
6		Dr. Luciano Dicki V Hollis O'Reilly	kaylie55@example.org	<span>Inactive</span>	
7		Parker Bayer Angela Turner	goyette.frida@example.net	<span>active</span>	

Figure 24 interface de gestion des utilisateurs

#### 4.4.5 Interface de la liste des commandes

Une fois connecté, l'administrateur peut consulter les commandes des clients, il peut aussi chercher, modifier, annuler, faire le suivi de chaque commande.

The screenshot shows the AyaMarket admin interface. On the left is a sidebar with various management options: Dashboard, About Us Content, Banner Management, Category Management, Products Management, Coupon Management, Shipping Management, Order Management, User Management, and General Settings. The main area is titled 'Order List' and displays a table with one entry. The columns are S.N., Name, Email, Payment Method, Payment Status, Total, Status, and Action. The entry shows a customer named Brenda Tillman with an email of customer@gmail.com, payment method 'paypal', payment status 'Paid', total amount of € 48.72, and a status of 'PENDING'. There are also icons for viewing and deleting the order. At the bottom of the table, it says 'Showing 1 to 1 of 1 entries' and includes 'Previous' and 'Next' buttons. The top right corner shows 'Total Orders: 1'.

Figure 25 interface de la liste des commandes

#### 4.4.6 Interface d'accueil

L'interface d'accueil est accessible aux internautes, elle présente un aperçu des catégories et des produits disponibles, elle donne aussi la possibilité d'ajouter des produits au panier et effectuer une recherche par produit ou catégorie.

The screenshot shows the AyaMarket online shopping website. At the top, there is a header with the logo 'AYA AYAMARKET', a search bar containing 'Search Products here...', and social media links for Facebook, Twitter, and LinkedIn. To the right, there is a 'Hello, Sign In' button. Below the header, there are four large colored boxes (green, dark blue, green, and light blue) with placeholder text: 'sequi', 'neque', 'nesciunt', and 'sequi'. Below these boxes, there is a section titled 'New Arrivals' with a grid of six product cards. Each card shows a small image, the product name (e.g., 'neque', 'veniam', 'et', 'adipisci', 'sint', 'ullam'), and a price (e.g., € 273.00, € 201.00, € 373.00, € 345.00, € 600.00, € 252.00). At the bottom of the page, there is a footer with a 'Gel couvert' link and a date '12/03/2022'.

Figure 26 Interface d'accueil

La page d'accueil est organisée d'une façon à mettre en évidence les catégories disponibles, les nouveaux arrivages, les produits souvent consultés et les bannières.

La figure 28 présente l'organisation du code avec la balise « SECTION ».

Bien sûr que le visiteur pourra effectuer une recherche de produit grâce à la fonction que j'ai appelée Search (figure 27),

```
145     //search product
146     public function search(Request $request){
147         $query=$request->input('key: query');
148         $products=Product::where('title', 'LIKE', '%' . $query . '%')->orderBy('id', 'DESC')->paginate(12);
149
150         $cats=Category::where(['status'=>'active', 'is_parent'=>1])->with('products')->orderBy('title', 'ASC')->get();
151
152         return view('frontend.pages.product.shop', compact('var_names: products', 'var_names: cats'));
153     }
```

Figure 27 Function Search

L'internaute aura la possibilité de : Consulter le panier, Consulter la wishlist, Naviguer sur les autres pages et Contacter nos services. Pour le client peut en plus s'identifier et parcourir son panier ou gérer son compte. Le futur client pourra quant à lui de s'inscrire.

```
5      @if(count($banners)>0)
6          <!-- Hero Slider -->
7              <div id="main-slider" class="main-slider-wrapper col-md-9 centered">
8                  <div class="owl-carousel owl-theme hero-slider" ...>
9                  </div>
10             <!-- Hero Section End -->
11         @endif
12         <!-- Main Content Section Start -->
13         <div id="main">
14             @if(count($categories)>0)
15                 <!-- Discount Offers on Categories -->
16                     <section class="discounted-categories mt-3 mb-3">
17                         <div class="container-fluid" ...>
18                         </div>
19                     </section>
20                 <!-- Discount Offers on Categories End -->
21             @endif
22             <!-- Promotion Banner -->
23             @if(count($new_products)>0)
24                 <!--New Arrivals Start -->
25                     <section class="new-arrivals-section mt-3 mb-3 ml-3 mr-3 pt-3 pb-3">
26                         <div class="container-fluid" ...>
27                         </div>
28                     </section>
29             @endif
30             <!--New Arrivals End-->
31             <!-- Promo Start -->
32             @if(count($promo_banner)>0)
33                 <section id="main-slider" class="main-slider-wrapper">
34                     <div class="owl-carousel owl-theme hero-slider" ...>
35                     </div>
36                 </section>
37             @endif
38             <!-- Promo End -->
39             <!-- Featured Section -->
40             @if(count($featured_products)>0)
41                 <section class="featured-section">
42                     <div class="container-fluid" ...>
43                     </div>
44                 </section>
45             @endif
46             <!-- Featured Section End-->
47         </div>
48     
```

Figure 28 Résumé du code page HOME

#### 4.4.7 Interface contact

L'interface ContactUs met à disposition aux internautes un formulaire pour contacter la société pour de différents problèmes ou questions.

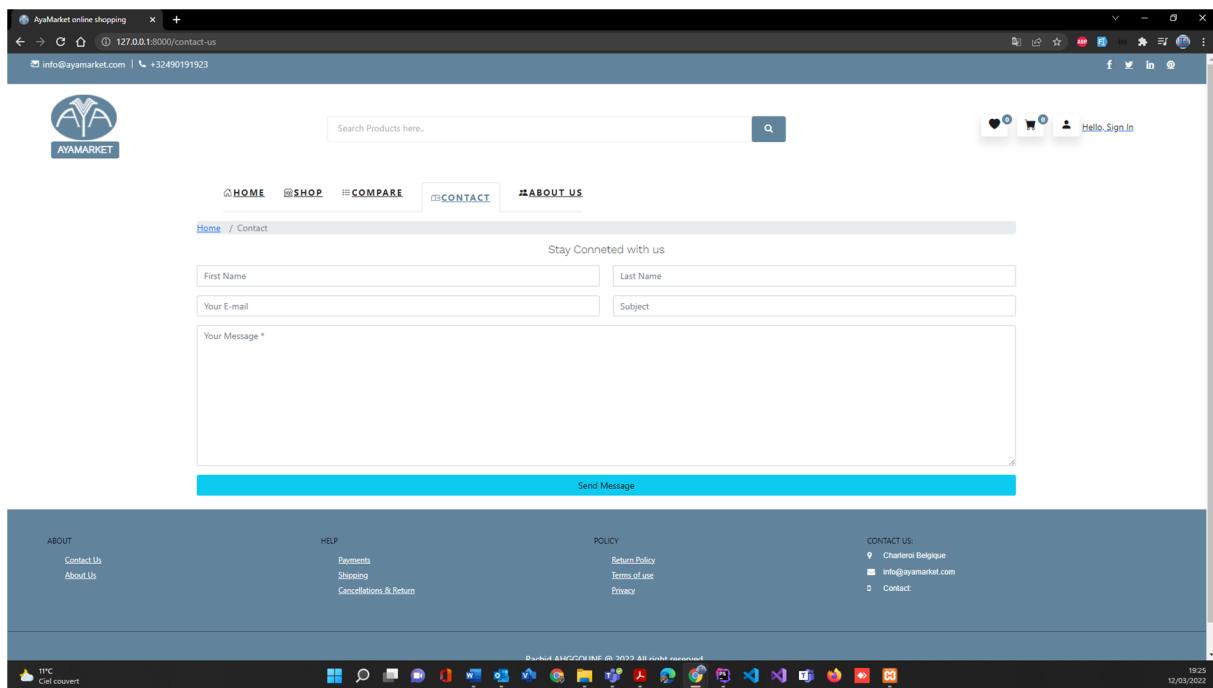


Figure 29 Interface de contact

```

46     public function contactSubmit(Request $request){
47         $this->validate($request,[
48             'f_name'=>'string|required',
49             'l_name'=>'string|required',
50             'email'=>'email|required',
51             'subject'=>'min:4|string|required',
52             'message'=>'string|nullable|max:200',
53         ]);
54         $data=$request->all();
55         Mail::to( users: 'admin@gmail.com')->send(new Contact($data));
56         return back()->with('success','Successfull send your enquiry');
57     }

```

Figure 30 code du formulaire du contact

La fonction `contactSubmit` reçoit les informations de la page contact-us et appelle la méthode (`to`) et (`send`) de la classe `Mail`. La méthode (`to`) prend une adresse mail en paramètre et la méthode (`send`) quant à elle, prend en paramètre un objet dont ces attributs contiennent les éléments nécessaires pour construire un e-mail.

#### 4.4.8 Interface Shop

Dans cette interface le visiteur peut consulter les produits disponibles dans notre catalogue et filtrer par catégories ou prix et aussi trier par prix ou ordre alphabétique.

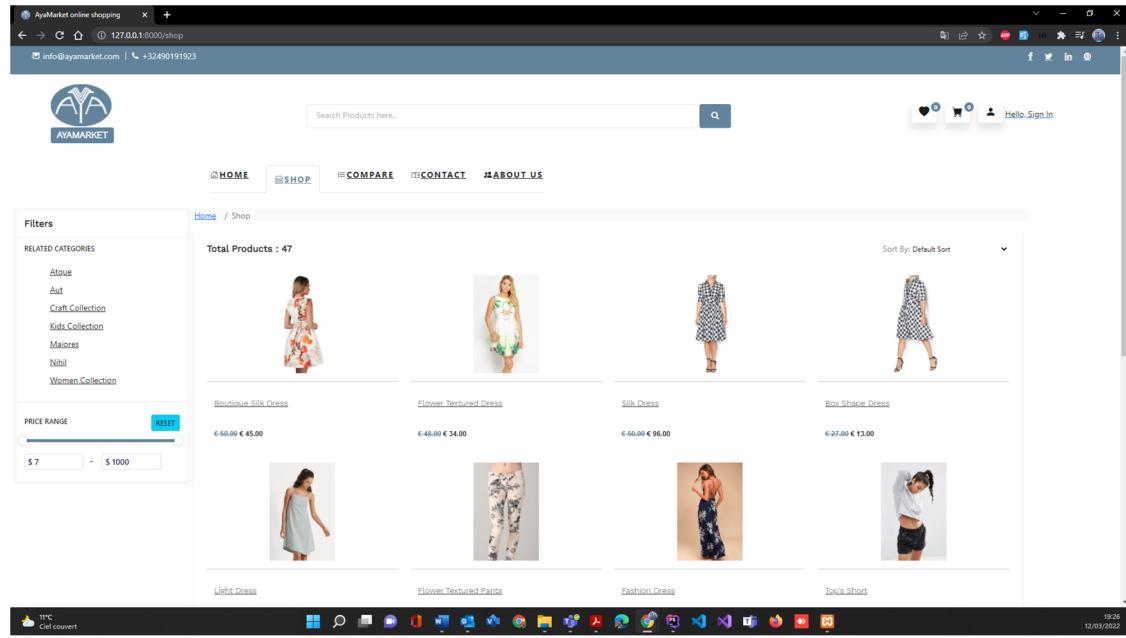


Figure 31 Interface Shop

#### 4.4.9 Interface commande

Lorsque le client ajoute au moins un produit dans le panier, cette interface du panier peut être consultée afin de prévisualiser la commande et éventuellement ajouter un coupon, ensuite, passer aux étapes suivantes pour finaliser la commande.

La figure suivante présente la réaction du système lorsque le client ajoute un produit au panier.

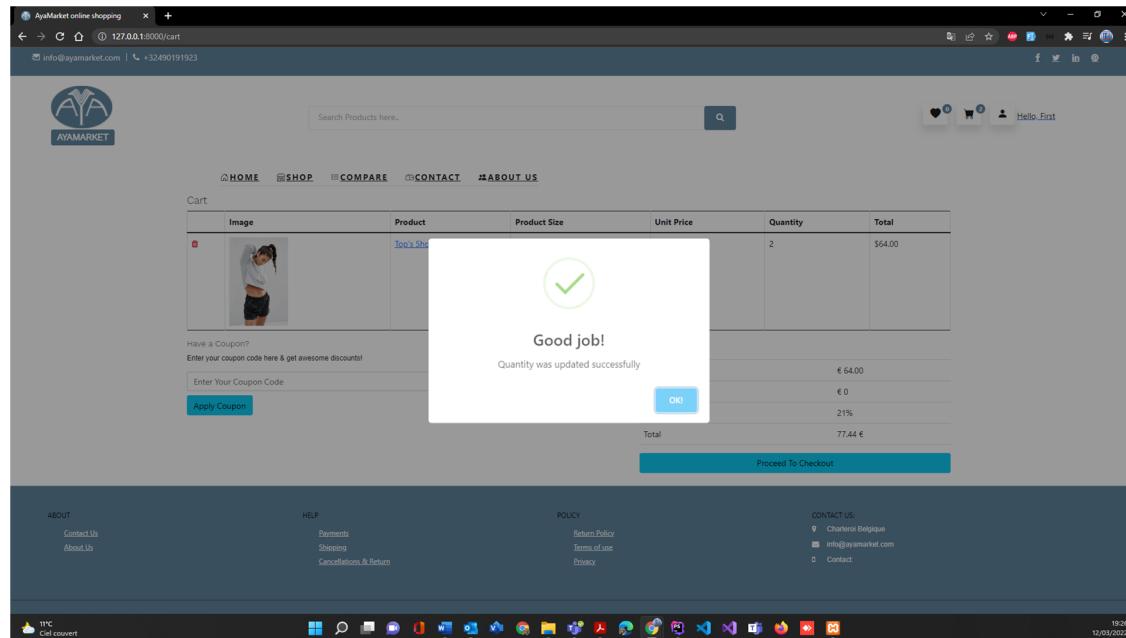


Figure 32 Interface commande

#### 4.4.10 Interface Login / Register

Il s'agit d'un simple formulaire login qui permet à l'internant de se connecter et pouvoir effectuer des achats ou consulter leur compte.

Le client peut aussi récupérer son mot de passe en suivant le liens (Forget password ?) présent sur cette page.

L'interface permet au futur client de pouvoir s'inscrire via formulaire register.

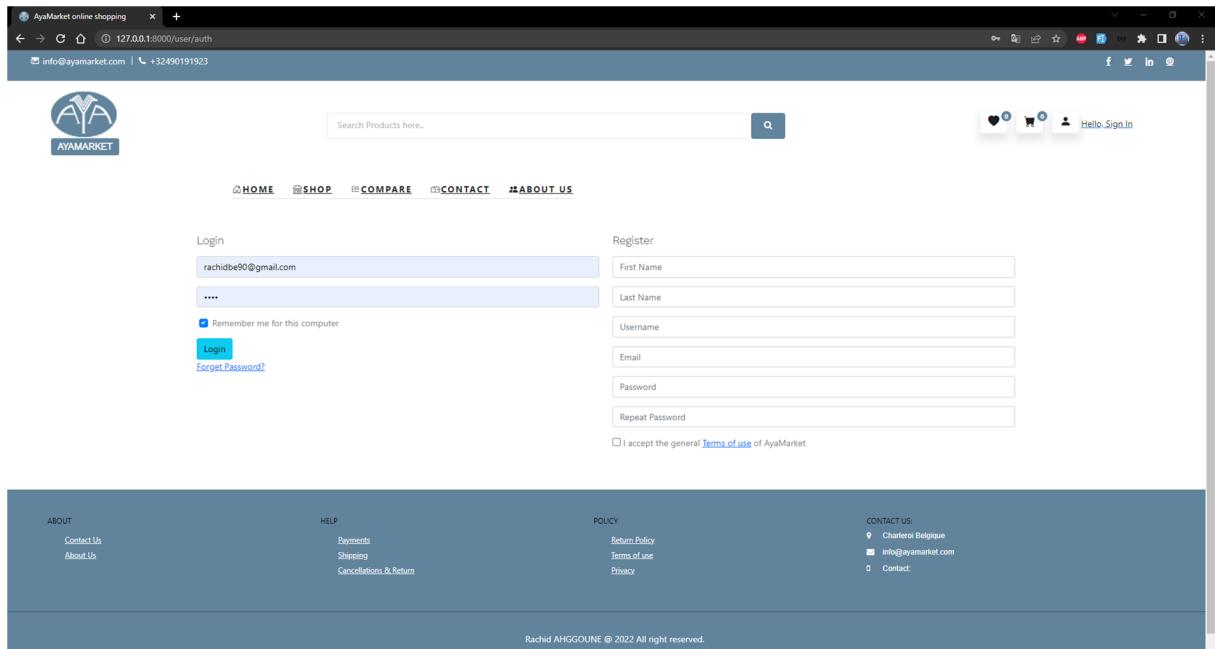


Figure 33 Interface Login / Register

#### 4.5 Obligations légales

L'application Web AyaMarket est dotée d'un système de cookies interne qui permet à notre application de stocker sur le pc client les informations comme l'identifiant et le contenu du panier, ces informations sont strictement nécessaires aux fonctionnalités expressément demandées par l'utilisateur. La première visite de l'internaute sur l'application Web lui permet de choisir d'accepter ou de refuser l'utilisation de ces cookies. Dans le cas de refus l'internaute sera redirigé vers la page d'accueil de Google. S'il accepte, son choix sera stocké sur les cookies du navigateur, aucune inscription ne sera acceptée si l'internaute n'accepte pas l'utilisation des cookies.

Pour plus d'information aux internautes, nous avons créés des pages de consultation des conditions générales d'utilisation de l'application Web, accessible depuis le Footer, la Politique de protection de la vie privée et d'autres informations utiles pour la bonne utilisation de l'application AyaMarket seront également disponible.

#### 4.6 Validation w3c



Tous les codes ont été bien sûr validés sur *W3C Validator*, aussi bien le CSS que HTML. Cela signifie que les pages de l'applications sont 100% conforme au standard ce qui aide pour le référencement sur les moteurs de recherche.

#### 4.7 Conclusion

Durant ce chapitre, j'ai présenté l'environnement matériel et logiciel utilisé lors du développement de notre application E-commerce. Ensuite, j'ai présenté les différentes interfaces et le bon fonctionnement de notre application AyaMarket.

Et voici le lien GitHub pour ceux qui désire consulter mon travail de programmation ([https://github.com/rachidbe90/Tfe\\_ecom/tree/Rachid-Final](https://github.com/rachidbe90/Tfe_ecom/tree/Rachid-Final)), vous pouvez y trouver les diagrammes et les schémas réalisé à l'aide de Vision, ainsi que ma présentation PowerPoint.

## 5 Conclusion générale

---

Cet ouvrage permet de visualiser les différentes étapes de la mise en place détaillée d'une application d'e-commerce, plus précisément dans le domaine de l'habillement.

Il s'agit d'un projet ambitieux de la vente de vêtements spécifiquement en ligne avec tous les avantages et inconvénients que cela comporte et dont nous avons tiré profit à chaque étape du développement.

Ce processus avait comme objectif principal, la création d'une application Web qui servira à notre société en ligne « AYAMARKET ».

Dans cette perspective, j'ai été amené à analyser les exigences de base pour une fonction optimale d'une application Web d'e-commerce, telles que les besoins, la gestion des données, la garantie de confidentialité, la gestion des stocks, la gestion des différents produits et la sécurité lors des paiements.

Pour cela, J'ai prédéfini les différentes étapes de la création de l'application servant de « pilote » dans le développement des fonctionnalités de cette conception, entre-autres, grâce à des tableaux et de diagrammes UML2.5.

Le but de cette application est avant tout, la satisfaction client, ensuite, lui permettre de passer et payer ses commandes en ligne, mais pas seulement ! elle permet tant au client qu'à tous les internautes d'explorer notre application, visualiser les produits présentés ainsi que la possibilité d'inscription et de mise à jour des données personnelles. En ce qui concerne l'administrateur, l'application l'aide à la gestion des produits, des commandes, des stocks et des clients.

Pour conclure, cette application aura encore des perspectives de développement afin de se positionner dans ce nouveau monde de technologie en perpétuelle évolution et dont nous devons trouver constamment des nouvelles stratégies afin d'être performant pour faire face à la concurrence féroce qui ne cesse de croître.

Aujourd'hui, je me sens fier du résultat et du fonctionnement de l'application Web « AYAMARKET » qui reflète mes ambitions.

## 6 Références

---

- @michaeldumontet. (2020, 12 3). *mailtrap cest quoi*. Consulté le 02 03, 2022, sur [www.eewee.fr:](http://www.eewee.fr/) <https://www.eewee.fr/mailtrap-cest-quoi/>
- Bootstrap. (2020, 01 01). *alerts*. Consulté le 12 12, 2021, sur [https://getbootstrap.com:](https://getbootstrap.com/) <https://getbootstrap.com/docs/4.0/components/alerts/>
- Communauté. (2020, 01 01). *All*. Consulté le 12 01, 2021, sur [https://stackoverflow.com:/](https://stackoverflow.com/) <https://stackoverflow.com/>
- CORRE, F. L. (2020, 12 17). <https://www.alphorm.com/>. Récupéré sur <https://www.alphorm.com/tutoriel/formation-en-ligne-laravel-le-guide-complet-du-debutant/tuto-video-presentation-de-laravel-et-pourquoi-l-utiliser-diagramme-de-classes>. (s.d.). Récupéré sur [ecole-en-ligne.blogspot.com/2016/02/diagramme-de-classe.html](http://ecole-en-ligne.blogspot.com/2016/02/diagramme-de-classe.html)
- Edwards, T. (2020, 01 01). *guides/*. Consulté le 01 22, 2022, sur [https://sweetalert.js.org:](https://sweetalert.js.org/) <https://sweetalert.js.org/guides/>
- forum, I. (2022, 01 01). *laravel-auth-password-reset-email-template*. Consulté le 03 17, 2022, sur <https://laracasts.com: https://laracasts.com/discuss/channels/laravel/laravel-auth-password-reset-email-template>
- gitbrent. (2019, 10 19). *switch-button-bootstrap*. Consulté le 02 03, 2022, sur <https://www.jqueryscript.net: https://www.jqueryscript.net/form/switch-button-bootstrap.html>
- HAJJI, R. (2020, 12 20). *le modèle MVC de Laravel*. Consulté le 12 15, 2021, sur <https://apcpedagogie.com/: https://apcpedagogie.com/le-modele-mvc-de-laravel/>
- <https://www.e-bancel.com/>. (2022, 01 01). <https://www.e-bancel.com/>. Consulté le 03 01, 2022, sur E-bancel: [https://www.e-bancel.com/Processus\\_2TUP.php](https://www.e-bancel.com/Processus_2TUP.php)
- jeremyguyot. (2010, 01 22). *affich-16268257-condition-a-accepter-d-un-formulaire*. Consulté le 11 09, 2021, sur <https://forums.commentcamarche.net: https://forums.commentcamarche.net/forum/affich-16268257-condition-a-accepter-d-un-formulaire>
- JetBrains. (2022, 03 17). *quick-start-guide-phpstorm*. Consulté le 10 01, 2021, sur <https://www.jetbrains.com/: https://www.jetbrains.com/help/phpstorm/quick-start-guide-phpstorm.html>
- Kinsta. (2022, 05 24). *qu'est ce que laravel*. Consulté le 03 20, 2022, sur <https://kinsta.com/: https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/>
- Laravel. (2020, 06 22). *laravel-email-verification*. Consulté le 03 10, 2022, sur <https://mailtrap.io/: https://mailtrap.io/blog/laravel-email-verification/>
- Laravel. (2022, 01 01). *Docs*. Consulté le 11 01, 2021, sur <https://laravel.com/: https://laravel.com/docs/8.x>

- ligne, E. e. (2022, 01 26). *diagramme de classe*. Récupéré sur Blog Ecole en ligne: <http://ecole-en-ligne.blogspot.com/2016/02/diagramme-de-classe.html>
- Old-fashioned. (2020, 10 06). *get+join+with+another+model+in+laravel*. Consulté le 12 16, 2021, sur <https://www.codegrepper.com/>: <https://www.codegrepper.com/code-examples/php/get+join+with+another+model+in+laravel>
- Rana, U. (2020, 05 31). *send-email-using-gmail-smtp-in-laravel-7*. Consulté le 03 24, 2022, sur <https://programmingfields.com/>: <https://programmingfields.com/send-email-using-gmail-smtp-in-laravel-7/>
- Roques, P. (2018). *UML 2.5 par la pratique* (éd. 8). Eyrolles. Consulté le 10 07, 2021
- Sillo. (2020, 09 30). *cours-laravel-8-la-securite-on-se-protege*. Consulté le 11 15, 2021, sur <https://laravel.sillo.org/>: <https://laravel.sillo.org/cours-laravel-8-la-securite-on-se-protege>
- Summernote. (2020, 01 01). *getting-started*. Consulté le 12 26, 2021, sur <https://summernote.org/>: <https://summernote.org/getting-started>
- W3techs. (2022, 01 01). *Programming language*. Consulté le 03 01, 2022, sur <https://w3techs.com/>: [https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language)
- XAMPP. (2022, 03 20). Consulté le 11 01, 2021, sur Wikipédia: <https://fr.wikipedia.org/wiki/XAMPP#:~:text=XAMPP%20est%20un%20ensemble%20de,son%20installation%20simple%20et%20rapide>.

## 7 Bibliographie

---

- Pascal Roques – UML 2.5 par la pratique (8<sup>e</sup> édition) – EYROLLES 2018.
- Jean-Luc Hainaut – Base de données Conception, utilisation et développement (4<sup>e</sup> édition) – DUNOD 2018.
- Rodolphe Rimelé –HTML 5 Une référence pour le développeur Web (3<sup>e</sup> édition) –EYROLLES 2018.<sup>2</sup>
- S. Berger, C. Cassagne, C. Chaissac & R. Rampnoux – SQL par l'exemple –ELLIPSES 2017.
- Vincent Granet – Algorithmique et programmation 5<sup>e</sup> édition –DUNOD 2018.
- Jean Engels – PHP 7 Cours & exercices –EYROLLES 2017.
- C.Pierre de Geyer, G. Ponçon – Mémento PHP7 & SQL (2<sup>e</sup> édition) - EYROLLES 2020.
- Raphaël Goetter – CSS3 Grid Layout –EYROLLES 2019.
- Raphaël Huchet – LARAVEL –ENI 2018.
- Pascal Roques – Mémento UML 2.5 (4<sup>e</sup> édition) –EYROLLES 2020

# Liste des tableaux

---

Tableau 1 Fiche produit.....	17
Tableau 2 Identification des acteurs .....	25
Tableau 3 Use case (consulter).....	26
Tableau 4 Use case (s'inscrire) .....	27
Tableau 5 Use case (se connecter).....	27
Tableau 6 Use case (gérer panier).....	28
Tableau 7 Use case (Gérer compte) .....	28
Tableau 8 Use case (Acheter) .....	29
Tableau 9 Use case (Payer).....	29
Tableau 10 Use case (Contacter).....	30
Tableau 11 Use case (Ajouter).....	30
Tableau 12 Use case (Modifier).....	31
Tableau 13 Use case (Supprimer).....	31
Tableau 14 Use case Comparer .....	32
Tableau 15 Use cas (Mettre en favoris) .....	32
Tableau 16 Table About-us.....	38
Tableau 17 Table Admins .....	38
Tableau 18 Table Banners .....	39
Tableau 19 Table Categories .....	39
Tableau 20 Table Password_resets .....	39
Tableau 21 Table Coupons .....	40
Tableau 22 Table Products .....	40
Tableau 23 Table Orders .....	41
Tableau 24 Table Product_attributes.....	42
Tableau 25 Table Product_orders .....	42
Tableau 26 Table Shippings .....	42
Tableau 27 table product_reviews.....	43
Tableau 28 Table SETTINGS.....	43
Tableau 29 Table USERS .....	44

## 8 Le lexique

---

Mot	Définition
<b>Front-end</b>	Elément du site que l'on voit à l'écran, composé en HTML, CSS et Javascript. Cette notion est vague et peut être différente en fonction des entreprises.
<b>Back-end</b>	Tous les éléments d'une application qui sont invisibles pour l'utilisateur : modèles, contrôleurs, fonctions. Cette notion est vague et peut être différente en fonction des entreprises.
<b>GDPR / RGPD</b>	Réglementation européenne sur la protection des données applicable à partir du 25 mai 2018
<b>API</b>	Interface de programmation d'application
<b>PHP Artisan</b>	Interface CLI de Laravel
<b>Slug</b>	Texte court utilisable dans une URL
<b>Hash</b>	Empreinte servant à identifier une chaîne de caractère cryptée
<b>From Scratch</b>	À partir de rien
<b>ORM</b>	Est une technique de programmation qui permet de gérer une base de données orientée comme une base orientée objet.
<b>UML</b>	langage de modélisation objet unifié est une démarche orientée objet
<b>W3C</b>	Organisme de standardisation du code HTML & CSS
<b>MySQL</b>	Système de gestion de bases de données relationnelles (SGBDR) Open Source la plus Populaire au Monde

## 9 Annexes

Les deux figures suivantes présentent le diagramme de navigation de l'application web.

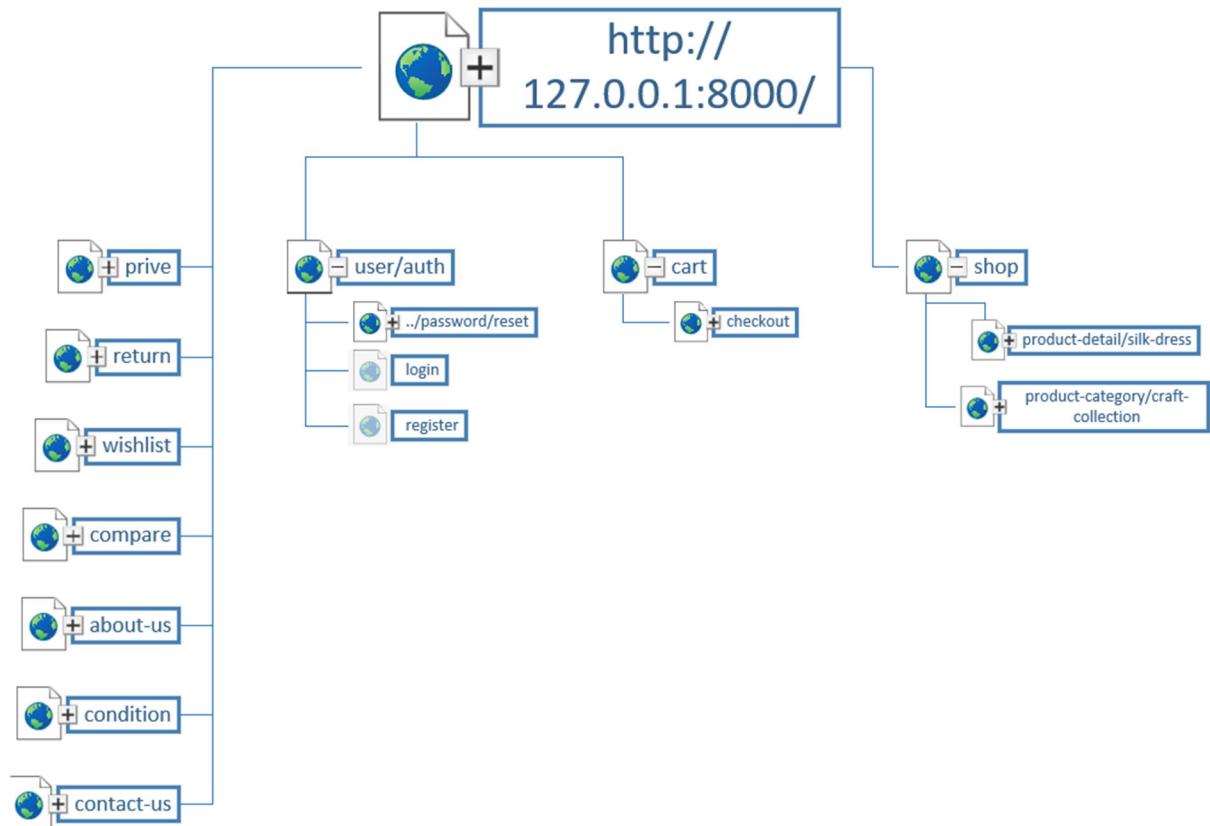


Figure 34 Diagramme de navigation Client

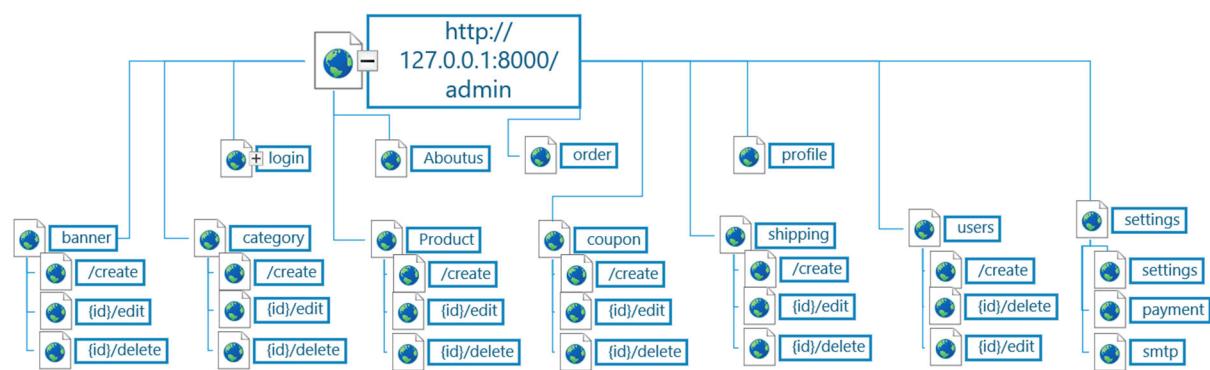


Figure 35 Diagramme de navigation Admin

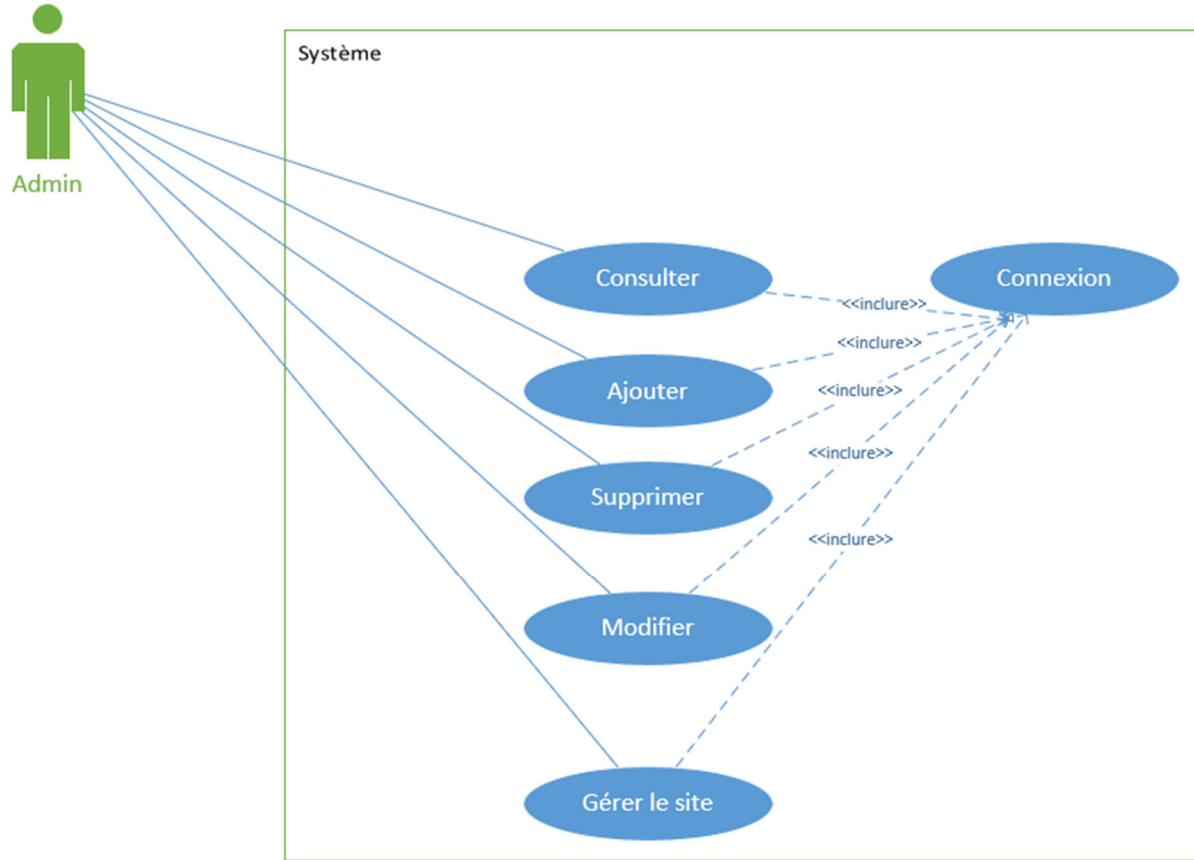


Figure 36 Diagramme d'utilisation (Admin)

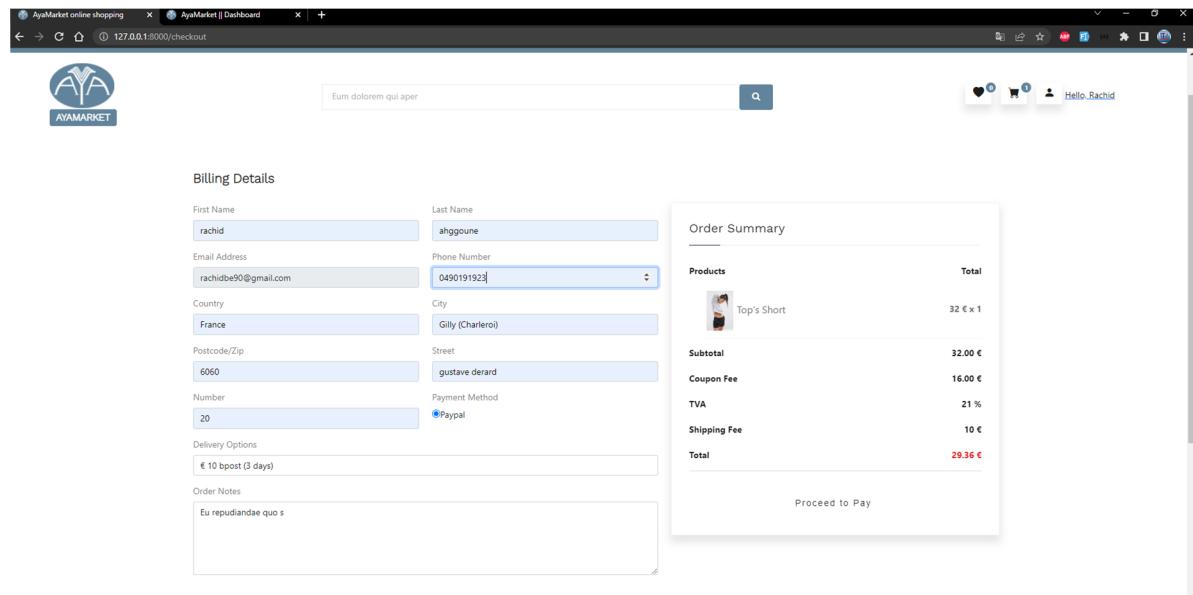


Figure 37 Page de commande

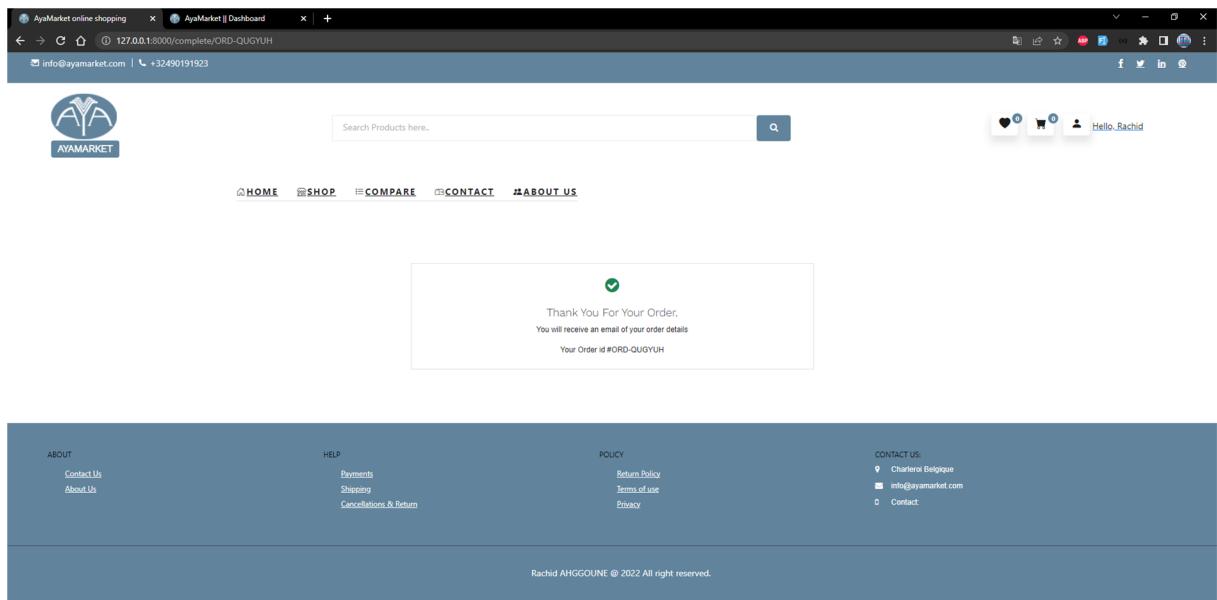


Figure 38 Page confirmation de commande

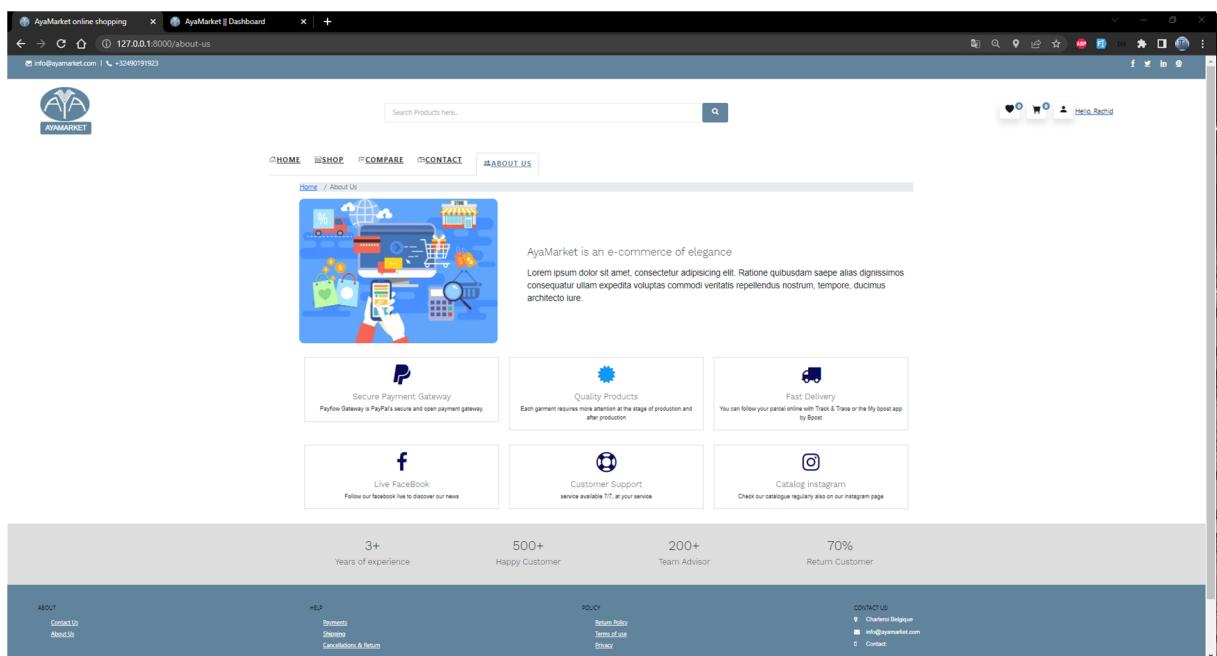


Figure 39 Page d'information sur l'entreprise.

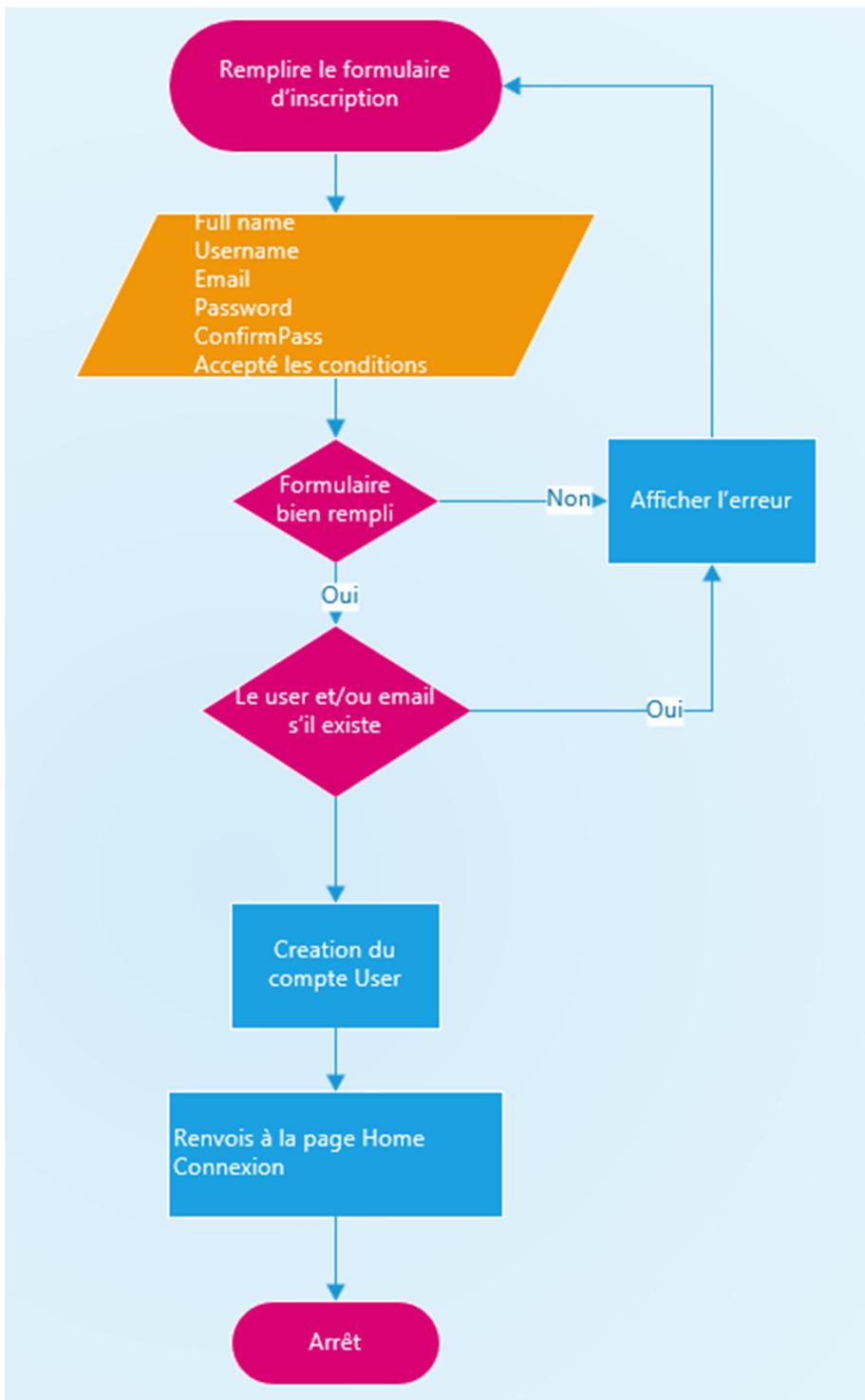


Figure 40 Algorithme d'Inscription

# Liste des Abréviations

---

Abréviation	Désignation
<b>UML</b>	Unified Modeling Language
<b>MVC</b>	Model View Controller
<b>IDE</b>	Integrated Development Environment
<b>HTML</b>	Hypertext Markup Lanuguage
<b>TTUP / 2TUP</b>	Tow Tracks Unified Process
<b>CLI</b>	Command Line Interface
<b>PHP</b>	PHP Hypertext Preprocessor
<b>ORM</b>	Mapping Objet-Relationnel
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>CSS</b>	Cascading Style Sheets
<b>Ajax</b>	Asynchronous JavaScript + XML
<b>SQL</b>	Structured Query Language
<b>API</b>	Application Programming Interface
<b>POO</b>	Programmation Orientée Objet
<b>ORM</b>	L'object RelationShip Mapping
<b>GDPR / RGPD</b>	General Data Protection Regulation
<b>W3C</b>	Le World Wide Web Consortium
<b>IETCps</b>	Institut d'Enseignement Technique Commercial de promotion social
<b>SGBDR</b>	Système de gestion de bases de données relationnelles
<b>BD</b>	Base de données