

[www.isl.be](http://www.isl.be)



**INSTITUT SAINT-LAURENT**  
**enseignement de promotion sociale**

rue Saint-Laurent 33 - 4000 LIEGE  
04 223 11 31

# Plateforme de vente en ligne

Liège, le 8/11/2019

Travail de fin d'études présenté par

**Iracanye Seleus-Kévin**

En vue de l'obtention du brevet de  
l'enseignement supérieur de

**Web Developer**

Enseignement supérieur économique de  
promotion sociale et de type court

## Table des matières

1. Introduction générales.....	4
2. Description du site.....	5
2.1 Thème et brève description.....	5
2.2 Objectifs.....	5
2.3 Public cible.....	5
3. Exigences fonctionnelles.....	7
3.1 Fonctionnalités.....	7
3.1.1 API et librairie tiers .....	7
3.2 Règles métier.....	13
Définition.....	13
Liste des règles métier identifiées.....	13
3.3 Interface.....	14
4. Exigences non-fonctionnelles.....	15
4.1 Conception.....	15
4.1.1. Ergonomie.....	15
4.1.2. Comportement.....	15
4.1.3 Sécurité.....	16
4.2 Acteurs.....	17
4.2.1 Acteurs humains.....	17
4.2.2 Acteurs systèmes.....	18
5. Cas d'utilisation.....	19
5.1 API.....	19
5.1.1. Utilisateurs anonymes.....	19
5.1.2. Utilisateurs authentifiés.....	19
5.1.3 Administrateurs.....	19
5.2. Interface client.....	20
5.2.1. Utilisateurs anonymes.....	20
5.2.2. Utilisateurs authentifiés.....	20
5.2.3 Administrateurs.....	20
5.3 Interface administrateur.....	21
5.3.1 Administrateurs.....	21
5.3.2 Administrateur système.....	21
6. Description de la base de données.....	22
6.1 Modélisation conceptuelle de la base de données.....	23

6.1.1 Inscription (Utilisateurs de la plateforme).....	23
6.1.2 Publication.....	24
6.1.3 Commande.....	25
6.1.4 Paiement.....	26
6.1.5 Livraison.....	27
6.1.6 Relation clients - vendeurs.....	28
6.2 Modélisation physique des entités.....	31
6.2.1 Address.....	31
6.2.2 Admin.....	32
6.2.3 Bank Account.....	33
6.2.4 Bill.....	34
6.2.5 Bill Customer.....	35
6.2.6 Bill Refund.....	36
6.2.7 Bill Supplier.....	37
6.2.8 Category.....	38
6.2.9 Customer.....	39
6.2.10 Comment.....	40
6.2.11 Delivery Detail.....	41
6.2.12 Delivery set.....	42
6.2.13 Favorite.....	43
6.2.14 Forum.....	44
6.2.15 Image.....	45
6.2.16 Message.....	46
6.2.17 Order Detail.....	47
6.2.18 Order Set.....	48
6.2.19 Order Returned.....	49
6.2.20 Payment.....	50
6.2.21 Product.....	51
6.2.22 Shipper.....	52
6.2.23 Supplier.....	53
6.2.24 Supplier Product.....	54
6.2.25 User.....	55
6.2.26 Withdrawal.....	56
7. Réalisation.....	57
7.1 Choix de développement.....	57
7.2 Environnement de développement et logicielle.....	57
7.3 Présentation des problèmes rencontrés et des solutions envisagées.....	57

8. Conclusion.....	58
9. Bibliographie.....	59
10. Annexes.....	60

## Lexique des abbréviations et définitions

API	Application Programming Interface Une interface de programmation d'application ou interface de programmation applicative
Argon2i	Symfony recommande d'utiliser l'encodeur de mot de passe Bcrypt pour encoder les mots de passe de vos applications. Bcrypt a été spécialement conçu pour le stockage de mots de passe à long terme et est supporté nativement par PHP. Cependant, la sécurité est un domaine en évolution rapide où de nouvelles meilleures pratiques et recommandations apparaissent continuellement. Au vue de la popularité de l'algorithme de hachage Argon2 qui a explosé, notamment depuis sa victoire au concours de hachage de mots de passe 2015. En fait, PHP 7.2 (à paraître fin 2017) inclut un support intégré pour Argon2.
DELETE (HTTP method)	« HTTP définit un ensemble de <b>méthodes de requête</b> qui indiquent l'action que l'on souhaite réaliser sur la ressource indiquée. » La méthode DELETE supprime la ressource indiquée.
GET (HTTP method)	« HTTP définit un ensemble de <b>méthodes de requête</b> qui indiquent l'action que l'on souhaite réaliser sur la ressource indiquée. » « La méthode GET demande une représentation de la ressource spécifiée. Les requêtes GET doivent uniquement être utilisées afin de récupérer des données. »
HTTP	HyperText Transfer Protocol Protocole de communication entre un client (ex : navigateur) et un serveur via le World Wide Web (réseau de serveurs distants interconnectés)
HTTPS	HyperText Transfer Protocol Secured Protocole de communication entre un client (ex : navigateur) et un serveur via le World Wide Web (réseau de serveurs distants interconnectés). Ce protocole utilise le protocole SSL et TLS pour le chiffrement des données durant la communication.
OpenAPI	Anciennement connu comme Swagger « La spécification OpenAPI définit un standard, une interface de description de langage de programmation agnostique pour les API REST, qui permet à la fois aux humains et aux machines de découvrir et comprendre les capacités offertes par un service sans avoir besoin d'accéder à son code, consulter une documentation supplémentaire, ou analyser le trafic réseau. »
POST (HTTP method)	« HTTP définit un ensemble de <b>méthodes de requête</b> qui indiquent l'action que l'on souhaite réaliser sur la ressource indiquée. »

	La méthode POST est utilisée pour envoyer une entité vers la ressource indiquée. Cela entraîne généralement un changement d'état ou des effets de bord sur le serveur.
PUT (HTTP method)	« HTTP définit un ensemble de <b>méthodes de requête</b> qui indiquent l'action que l'on souhaite réaliser sur la ressource indiquée. » La méthode PUT remplace toutes les représentations actuelles de la ressource visée par le contenu de la requête.
Swagger UI	Swagger User Interface Une interface graphique permettant à chacun (développeur ou utilisateur ) d'interagir avec des ressources d'une API sans en connaître tout l'implémentation logique nécessaire.

## 1. Introduction générale

Dans le cadre de mon projet de fin d'études en vue de l'obtention du diplôme supérieur de Web développeur, j'ai été amené à réaliser un projet de développement d'application web.

Ce projet de développement doit nous permettre de mettre en œuvre l'ensemble de connaissances acquis au cours de notre formation tout en nous préparant aux exigences du monde professionnel.

De plus, dans un secteur en perpétuelle évolution, il est nécessaire d'effectuer une veille technologique constante pour veiller à fournir des applications à la pointe de la technologie et respectant les standards du marché.

De ce fait, il est primordiale de se mettre à jour sur les nouveaux standards et technologies utilisés dans l'industrie afin de pouvoir les implémenter et offrir des applications.

Dans cet objectif, il nous est demandé après une étude préalable d'un projet personnel ou sur demande réel, de réaliser une application ou site web. Ce rapport présente l'ensemble de travaux accomplis afin de matérialiser ce projet de développement.

Dans la suite du rapport, vous trouverez une présentation de mon projet, l'étude des exigences réalisées afin de le rendre opérationnel, une description de la base de données, une présentation des cas d'utilisation de l'application, ainsi qu'une description du processus de réalisation et des problèmes rencontrés au cours de celui-ci. Afin vous trouverez à la fin ce rapport, la conclusion ainsi qu'une bibliographie et les annexes.

Ce rapport est organisé selon les chapitres suivantes :

- Introduction
- Description du site
- Exigences
- Cas d'utilisation
- Description de la base de données
- Réalisation
- Conclusion

## 2. Description du site

Dans la suite de ce chapitre, vous trouverez une présentation du projet de application web.

Vous trouverez dans ce chapitre la présentation du thème du projet ainsi qu'une brève description de l'application, les objectifs de l'application et son public cible.

### 2.1 Thème et brève description

Mon choix s'est porté vers la réalisation d'un projet personnelle afin de me familiariser avec les composants les plus répandues dans les site internet :

- Un composant de paiement de paiement en ligne afin de réaliser tout type de transactions (abonnement, achat, remboursement)
- Un composant de chat afin de permettre aux utilisateurs de la plateforme de communiquer entre eux et avec les administrateurs.

### 2.2 Objectifs

Ce projet porte sur la réalisation d'une plateforme de vente en ligne avec des outils à la pointe de la technologie.

Cette plateforme permettra la rencontre de vendeurs de produits et d'acheteurs qui serait intéressés par leurs offres.

Cette plateforme doit permettre à des fournisseurs de produits de s'inscrire et proposer leurs produits à la vente.

Ces fournisseurs peuvent par la suite :

- suivre l'ensemble des commandes pour leurs produits
- voir les paiements perçus via la plateforme pour leurs produits.
- Permettre aux utilisateurs de communiquer avec les clients de la plateforme

Les clients potentiels doivent s'identifier sur la plateforme pour effectuer leurs achats

Les clients pourront par la suite :

- suivre l'ensemble de leurs commandes, du paiement à la livraison.
- communiquer avec les fournisseurs de produits ainsi que les administrateurs de la plateforme



## 2.3 Public cible

Ce chapitre présente le public cible de la plateforme.

Par définition, le terme « public cible » en marketing renvoie aux caractéristiques principales des consommateurs d'un produit donné.

### Fournisseurs de produits

Tout personne physique ou morale reconnu comme étant qualifié pour la vente voulant fournir des produits à des clients via la plateforme.

Chaque fournisseurs de la plateforme devra fournir une preuve d'inscription au registre de commerce du pays d'origine dans notre cas le Burundi.

Après vérification externe de l'authenticité des informations fournit par les administrateurs de la plateforme, ils seront autorisé à publier des produits à la vente sur la plateforme.

### Clients

Tout visiteur de la plateforme qui pourrait être intéressé par l'achat de produits proposés par l'application.

Cependant, l'achat de produits sera conditionné par la possibilité pour le transporteur choisi de livrer à l'adresse de destination.

Chaque client devra s'identifier sur la plateforme afin de réaliser ces achats, il devra fournir une adresse pour chaque commande effectué afin de permettre la livraison à destination.

Le processus de paiement se fera via bibliothèque externe afin de garantir la sécurité des données bancaires des clients de la plateforme. Les cartes bancaires autorisées par la bibliothèque externe, sont les cartes de type crédit comme VISA, MASTERCARD et AMERICAN EXPRESS



## 3. Exigences fonctionnelles

Ce chapitre présente de façon superficielle l'ensemble des exigences recueillies dans le cahier des charges du projet.

L'analyse des exigences sera divisée selon le type de celles-ci :

- Exigences fonctionnelles : L'analyse portera sur les fonctions attendues du système qui sera mis en place. Cette analyse contient une liste exhaustive des fonctionnalités de la plateforme, l'ensemble des règles métiers à respecter dans sa conception et le type d'interface désiré.
- Exigences non-fonctionnelles : L'analyse portera sur les besoins non-fonctionnels de l'application. Ces besoins comprennent l'ergonomie de la plateforme, le comportement de la plateforme, la sécurité et les contraintes spécifiques à la conception.

### 3.1 Fonctionnalités

#### 3.1.1 API<sup>1</sup> et librairie tiers

##### API tiers

- Google Sign-in : Inscription des particuliers et professionnels via leur compte Google
- Stripe : Système de paiement sécurisé en ligne
- UPS Rating : Calcul du coût du fret pour les services UPS

##### API interne

- Accès aux articles (produits) disponibles à la vente
- Calcul du coût du fret. Utilisation de l'API UPS rating pour le calcul du coût de transport associé à leurs services UPS
- Facture de commande sous format PDF : au choix après test jsPDF, RelaxedJS ou autre.

## Fonctionnalités disponibles pour les utilisateurs anonymes

- Effectuer la recherche d'un produit par nom et/ou catégorie sur la plateforme
- Visualiser les produits disponibles à la vente sous forme de liste par catégorie
- Détailler les informations de chaque produit disponible à la vente
- Détailler les informations de chaque professionnel
- Panier de commande contenant les produits sélectionnés par les clients
- Détailler les informations de livraison par zone géographique
- Inscription à la plateforme en tant que particulier ou professionnel
- La validation des inscriptions se fait par e-mail tant pour les professionnels que pour les particuliers
- Connexion à la plateforme d'achat via un formulaire simple ou compte Google

## Fonctionnalités disponibles pour les professionnels authentifiés non-admis à publier des articles destinés à la vente

- Visualiser et modifier leurs informations de profil
- Visualiser les modalités de publications d'articles destinés à la vente

## Fonctionnalités disponibles pour les utilisateurs authentifiés (particuliers)

- Afficher le Fil d'Ariane pour illustrer les étapes du processus de commande
- Ajouter les articles que les clients ont aimé dans une liste de souhaits (favoris)
- Visualiser et valider le panier de commande
- Enregistrer les informations de livraison
- Effectuer l'achat via le système de paiement sécurisé en ligne : Stripe
- Fournir une facture pour chaque commande sous format PDF sur demande du client ou au-delà de 25€.
- Fournir une confirmation du paiement et présenter les modalités de réclamation, rétractation ou de retour de marchandise.
- Visualiser les commandes en cours
- Permettre aux clients d'annuler une commande en cours si la livraison n'est pas confirmée ou que le délai de rétractation n'est pas dépassé
- Fournir un formulaire de réclamation sur une commande dont la livraison n'a pas été confirmée
- Fournir un formulaire de réclamation sur une livraison dont le colis n'est pas arrivé
- Fournir un formulaire de retour de marchandise : le client doit fournir une preuve de renvoi de l'article. Ex : Copie d'un recippé d'envoi du colis.
- Commenter et évaluer les produits achetés par les particuliers. Après confirmation de l'envoi du colis.
- Visualiser et modifier leurs informations de profil.
- Contacter les administrateurs de la plateforme par e-mail pour une demande d'informations ou pour tout autre litige.

## Fonctionnalités disponibles pour les professionnels authentifiés et admis à publier des articles destinés à la vente

- Articles
  - Visualiser la liste de tous les articles publiés par ceux-ci
  - Ajouter/modifier/supprimer des articles
  - Ajouter des images illustrant les articles mis en vente
- Commandes
  - Voir l'ensemble des commandes (ceux en cours et ceux qui sont complétées)
  - Voir le détail de chaque commande
  - Fournir une possibilité d'annuler une commande en cours (14 jours)
  - Confirmer la livraison d'une commande
  - Marquer une commande comme étant en litige
  - Contacter le client ou les administrateurs du site
- Livraisons
  - Voir les livraisons en cours.
  - Voir l'ensemble des livraisons (en cours, en litige, complétées)
  - Contacter le client ou les administrateurs du site
- Facturation
  - Visualiser l'ensemble des transactions en cours
  - Visualiser l'ensemble des paiements reçus
  - Mettre à jour les informations bancaires
- Clients
  - Visualiser les commentaires et évaluations des clients qui ont acheté leurs produits
  - Voir l'ensemble des messages reçus via la plateforme
- Litiges
  - Commandes en litige
    - Voir l'ensemble des commandes en litige
    - Visualiser le détail de chaque commande en litige
    - Possibilité de contacter le client concerné par un litige
  - Livraisons en litige
    - Voir l'ensemble des livraisons en litige
    - Visualiser le détail de chaque livraison en litige
    - Possibilité de contacter le client concerné par un litige
- Profil personnel
  - Visualiser et modifier les informations personnelles
  - Valider les termes et conditions de vente sur la plateforme ainsi que les garanties légales de vente à l'étranger
  - Permettre de se désinscrire de la plateforme
  - Permettre de se déconnecter de la plateforme

## Fonctionnalités disponibles pour les administrateurs de la plateforme

- Articles
  - Visualiser la liste de tous les articles publiés sur la plateforme
  - Ajouter/modifier/supprimer des articles
- Commandes
  - Commandes en cours
    - Visualiser l'ensemble des commandes en cours ainsi que le détail de chacune
    - Permettre aux administrateurs d'annuler une commande en cours
    - Permettre aux administrateurs de contacter un client ou un professionnel concerné par une commande.
  - Commandes effectuées
    - Visualiser l'ensemble des commandes effectuées sur la plateforme
  - Commandes en litige
    - Visualiser l'ensemble des commandes en litige
    - Permettre de marquer une commande en litige comme étant réglée
    - Permettre à l'administrateur d'effectuer le remboursement du client
    - Contacter le client ou le professionnel concerné par une commande en litige
- Livraisons
  - Livraisons en cours : livraison effectuée par le professionnel au transporteur mais dont le délai de rétractation n'est pas dépassé
    - Visualiser l'ensemble des livraisons en cours et leurs détails
  - Livraisons effectuées : livraison confirmée ou délai de rétractation dépassé
    - Visualiser l'ensemble des livraisons effectuées via la plateforme ainsi que leurs détails
  - Livraisons en litige
    - Visualiser l'ensemble des livraisons en litige ainsi que leur détails
    - Permettre aux administrateurs de marquer une livraison en litige comme étant réglée
    - Contacter le client ou le professionnel concerné par une livraison en litige
- Facturation
  - Visualiser l'ensemble des transactions en cours : Transactions validées sur la plateforme dont les sommes n'ont pas encore été versées sur le compte des professionnels
  - Valider le paiement du professionnel après dépassement du délai de rétractation ou après réception de la preuve d'envoi du produit commandé
  - Permettre aux administrateurs de procéder au remboursement d'un client dans le cadre du règlement d'un litige. Garder une trace du décaissement.
    - En cas de livraison non-effectuée, un remboursement automatique sera prévu
    - En cas de livraison effectuée, une preuve de retour de marchandise doit être associée. Une preuve d'envoi doit être fournie par le professionnel également.
  - Voir la balance des paiements sur la plateforme

- **Clients**
  - Visualiser les informations de profil de l'ensemble des membres de la plateforme
  - Permettre aux administrateurs de bloquer l'accès à la plateforme à un client pour comportement inadéquat
  - Contacter par e-mail un client
- **Professionnels**
  - Visualiser les informations de profil de l'ensemble des membres de la plateforme
  - Permettre aux administrateurs de bloquer l'accès à la plateforme à un professionnel pour comportement inadéquat
  - Contacter par e-mail un professionnel
- **Messages**
  - Visualiser l'ensemble des messages reçus sur la plateforme par catégorie d'utilisateur
  - Visualiser l'ensemble des messages concernant les litiges
  - Lire et répondre aux messages envoyés aux administrateurs de la plateforme
- **Profils**
  - Voir et modifier l'ensemble de ses informations de profil
  - Envoyer un e-mail contenant un lien sécurisé pour inscrire un autre administrateur
  - Changer de mot de passe
  - Se déconnecter



## 3.2 Règles métier

### Définition

Les règles métier<sup>1</sup> (ou règles de gestion, ou « *business rules* » en anglais) sont des déclarations de haut niveau structurées qui permettent de contraindre, contrôler et influencer un aspect du métier. Ces règles sont des conditions qui doivent être satisfaites lorsqu'une activité métier est exécutée.

Une règle métier permet d'influencer une prise de décision. Elle peut appliquer une stratégie métier, prendre une décision ou déduire les nouvelles données des données existantes.

La méthode utilisée sera de séparer la logique métier de la logique système ou applicatif dans une application.

Ainsi la logique métier pourra évoluer et être maintenue séparément du code de l'application. L'objectif étant de disposer de règles claires, c'est-à-dire non ambiguës et compréhensibles par tous, notamment par les non informaticiens.

### Liste des règles métier identifiées

- Les particuliers et les professionnels possédant un compte Google peuvent se connecter via l'API Google Sign-in
- Un compte est bloqué après 4 essais infructueux
- Site multilingue : tous les contenus textuels de la plateforme peuvent être traduits
- Modifier les mentions légales pour une plateforme d'e-commerce
- Choix de la devise lors de l'ajout d'article
- Affichage du prix de l'article selon les devises suivantes : BFI, USD, EUR.  
API Devise Zone : <https://devises.zone/api>

### 3.3 Interface

Cette section reprend les contraintes que l'interface graphique et le système doivent respecter. Ci-dessous vous trouverez les contraintes liées à l'interface graphique :

- La plateforme est multilingue : Français, Anglais, Kirundi
- Les mentions légales pour un site d'e-commerce doivent être visibles sur toutes les pages de la plateforme.
- La page d'accueil doit contenir les produits favoris des particuliers et les meilleurs professionnels.
- Une zone de recherche doit être accessible sur toutes les pages du site.
- Le résultat de la recherche doit être paginé

## 4. Exigences non-fonctionnelles

Ce chapitre présente les exigences non-fonctionnelles qui ne se rapportent pas au système.

### 4.1 Conception

Cette section présente les contraintes de conception à respecter

- Le design doit être mobile-first
  - Voir et modifier l'ensemble des informations de profil de l'administrateur
- Le layout sera adapté aux smartphones, tablettes, ordinateurs portables et écrans de bureau
- La plateforme respectera les normes de référencement
  - Permettre aux administrateurs et aux professionnels de ré-écrire les URL
  - Les mots-clés du site
  - Descriptif du site
  - Indexation des textes
- Les statistiques techniques doivent être accessibles aux administrateurs du site.

#### 4.1.1. Ergonomie

Cette section présente les contraintes d'ergonomie à respecter dans le cadre du projet

- La plateforme de vente doit être responsive
- Le design doit être mobile first
- Le client doit pouvoir effectuer l'achat en quelques clics
- aucune exigence au niveau de charte graphique

#### 4.1.2. Comportement

Cette section comprends les contraintes liées au comportement du système.

- Les articles publiés seront accessibles via une API interne
- Le calcul du coût du fret se fera via une API interne

#### 4.1.3 Sécurité

Cette section comprends les contraintes relatives à la sécurité du système.

- L'interface client sera accessible aux particuliers et professionnels
- L'interface administrateur sera accessible aux administrateurs uniquement.
- Les données personnelles des particuliers (adresse)
- L'authentification sur la plateforme se fera via :
  - formulaire simple : identifiant et mot de passe
  - compte Google : Google Sign-in
- Les mots de passe des utilisateurs authentifiés seront cryptés avant leur sauvegarde en base de données
- Les données confidentielles des particuliers et des professionnels seront cryptées avant leur sauvegarde en base de données. (adresse et données bancaires)
- Un Backlund de la structure de la base de données sera fournit

## 4.2 Acteurs

Ce chapitre présente les parties prenantes au système. Ces acteurs vont interagir avec le système en lui fournissant des données.

### 4.2.1 Acteurs humains

#### 4.2.1.1 Client

Le client représente une personne quelconque qui visite la plateforme de vente, sélectionne des produits puis se connecte à la plateforme d'achat pour effectuer les opérations de commande et paiement.

Ce client, dès lors qu'il est authentifié, peut suivre le processus des commandes qu'il a effectué ainsi que leur livraison.

Il peut ensuite faire part de sa satisfaction à l'ensemble des utilisateurs de la plateforme.

#### 4.2.1.2 Professionnel

Le professionnel représente une personne physique ou morale exerçant une activité de commerce qui souhaite présenter ces produits à la vente sur la plateforme.

Un professionnel doit s'identifier par les biais d'informations vérifiables (nom complet, nom de l'organisation, adresse du siège social, numéro de registre de commerce, numéro de téléphone, e-mail, informations bancaires).

Par la suite, un professionnel authentifié pourra proposer ses produits à la vente sur la plateforme. Suite à l'achat par un client d'un produit manufacturé par le professionnel, un paiement sera effectué par la plateforme sur le compte de celui-ci après confirmation de livraison des produits achetés au transporteur.

De plus, les professionnels pourront communiquer avec leurs clients via la plateforme.

#### 4.2.1.3 Administrateurs

L'administrateur représente une ou plusieurs personnes physiques qui seront responsables de la gestion de la plateforme au quotidien.

Ils auront accès à toutes les entités en base de données.

De plus, ils auront pour attribution de gérer les litiges entre clients et professionnels.

## **4.2.2 Acteurs systèmes**

### **4.2.2.1 Google Sign-in**

Google Sign-In est un système d'authentification sécurisé qui permet aux utilisateurs de s'inscrire sur un autre site internet avec les informations de leur compte Google.

L'API sert de passerelle pour se connecter avec les utilisateurs et services Google de façon sécurisée. Il donne la possibilité aux utilisateurs de Google d'effectuer leurs paiements avec Google Pay, de partager avec leurs contacts Google, d'ajouter des événements à leur calendrier Google, etc.

Dans le cadre de ce projet, nous l'utiliserons pour inscrire directement les utilisateurs ayant un compte e-mail de Google.

### **4.2.2.2 Stripe**

Stripe est un système de paiement sécurisé en ligne. Il fournit des outils permettant d'effectuer tous types de transactions bancaires sur la même plateforme.

Il permet aux développeurs de centraliser les opérations financières de leur activité au sein d'une seule plateforme.

Nous l'utiliserons afin de recevoir les paiements des clients et effectuer les décaissements vers les professionnels pour les produits achetés.

### **4.2.2.3 UPS Rating**

Le service d'envoi et livraison de colis d'UPS propose une API UPS Rating qui permet aux professionnels d'accéder aux calculs du coût du fret via leur service.

Dans le cadre de ce projet, il sera utilisé afin de calculer le coût du fret pour la livraison d'un produit acheté via la plateforme.

### **4.2.2.4 Devise Zone**

Devise Zone est un service de conversion de devises permettant un accès gratuit aux cours actualisés des devises étrangères.

L'API sera utilisé dans le but de convertir le prix des marchandises selon la situation géographique ou le besoin des utilisateurs et de façon actualisée.

## 5. Cas d'utilisation

Dans la suite de ce chapitre, vous trouverez une présentation des différents cas d'utilisation de la plateforme selon l'interface d'accès à celle-ci.

La plateforme de vente en ligne a été développée avec le framework API Platform qui offre

Ci-dessous vous trouverez comment le découplage de l'application a été effectué :

- Interface client : Interface graphique accessible par tous les visiteurs de la plateforme (clients potentiels et fournisseurs)
- Interface administrateur : Interface graphique accessible uniquement par les administrateurs de la plateforme
- Interface API : Service web permettant l'accès aux données de la plateforme par requête HTTP[S]

En effet, l'accès à la plateforme de vente peut s'effectuer de plusieurs manières selon l'information recherchée ou le type d'utilisateur connecté.

L'application permet l'accès aux ressources selon 2 schémas :

- Par requête HTTP vers l'API de la plateforme de vente
- Par requête HTTP vers l'interface graphique qui se charge ensuite d'acquiescer les ressources

En cas d'accès via l'interface graphique de la plateforme, une différenciation sera effectuée entre un accès utilisateurs authentifiés et un accès administrateur afin de servir l'interface graphique répondant aux besoins de chacun.

Ce chapitre est organisé de la façon suivante pour présenter les cas d'utilisation :

- Interface API
  - ◆ Présentation de toutes les routes et schémas de requête disponibles pour communiquer avec l'API
- Interface Client
  - ◆ Présentation des différentes pages et composants d'interface graphique disponibles selon le type d'utilisateur
- Interface Administrateurs
  - ◆ Présentations du composant d'administration de la plateforme ainsi que les pages disponibles pour les administrateurs de la plateforme et l'administrateur système

## 5.1 API – Application Programming Interface

Par définition, « une API<sup>2</sup> ou Interface de programmation d'application (en français) est Une interface de programmation est une façade clairement délimitée par laquelle un logiciel offre des services à d'autres logiciels[5]. L'objectif est de fournir une porte d'accès à une fonctionnalité en cachant les détails de la mise en œuvre. Une interface de programmation peut comporter un ensemble normalisé de [classes](#), de [méthodes](#), de [fonctions](#) et de [constantes](#) qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une [bibliothèque logicielle](#) ou un [service web](#), le plus souvent accompagnée d'une description qui spécifie comment des programmes *consommateurs* peuvent se servir des fonctionnalités du programme *fournisseur*. »

Dans le cadre de ce projet, une implémentation d'une API d'accès complet aux données de l'application a été favorisé. Dans cet objectif, le framework API Platform a été choisi pour les possibilités qu'il offre. Une explication de ce choix est exposé dans le chapitre « Réalisation – Choix de développement ».

Dans la suite de ce chapitre, vous trouverez l'ensemble des routes d'accès à l'API classées selon de type d'utilisateur autorisé d'accès ainsi qu'une description de la manière d'interagir avec chaque celle-ci.



### 5.1.0 Interface d'accueil de l'API

Le framework [API Platform](#) fournit beaucoup de fonctionnalités donc une particulièrement intéressante, la description de l'API dans le format de spécification [OpenAPI](#) ainsi qu'une interface graphique [Swagger UI](#) pour l'affichage de la documentation de l'API et l'interaction avec celle-ci.

Il permet d'envoyer les requêtes directement à l'API à partir d'une interface graphique conviviale en construisant les requêtes à partir des données fournies par l'utilisateur. Il retourne ensuite la réponse du serveur sous une de ces formats JSON/JSON+LD/HTML.

Ci-dessus, vous trouverez une représentation de la page d'accueil de l'API.

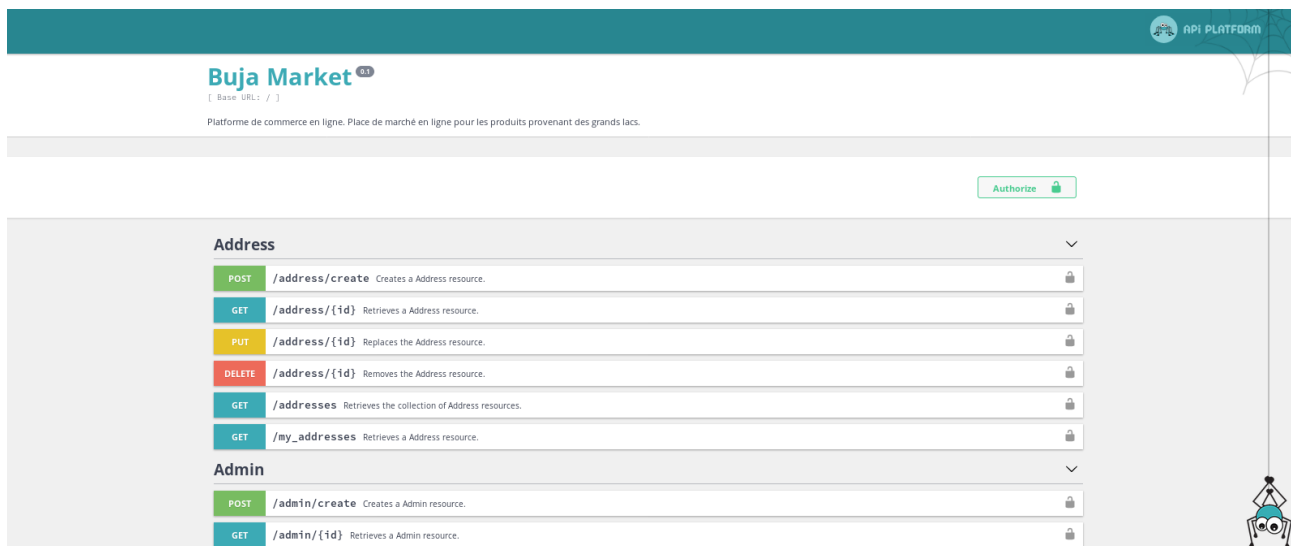


Figure 1: API Platform - API Homepage

La page d'accueil affiche pour chaque route d'accès à l'API disponible :

- les paramètres de requêtes requis
- Un exemple de contenu de réponse sous format JSON à une requête contenant :
  - le code de réponse HTTP
  - Un modèle de contenu de réponse qui peut être un message, une ressource ou une collection de ressources

Address

POST /address/create Creates a Address resource.

GET /address/{id} Retrieves a Address resource.

Parameters

Name

Description

id ★ required

string

(path)

Responses

Response content type application/json

Code	Description
200	<div>Address resource response</div> <div>Example Value   Model</div> <pre>{   "id": 0,   "locationName": "string",   "street": "string",   "number": "string",   "state": "string",   "town": "string",   "zipCode": "string",   "country": "string",   "user": "string",   "shipper": "string",   "orderSets": [     "string"   ] }</pre>
404	<div>Resource not found</div>

Figure 2: API Platform - Page d'accueil de l'API

### **5.1.1 Utilisateurs anonymes**

Ce chapitre présente les routes d'accès de l'API dite public car accessible à tous.

La partie publique non-sécurisée de l'API fournit les informations sur les produits proposés et leurs fournisseurs. Les informations sur les produits concerne la catégorie de classification, les informations des fournisseurs du produit, les illustrations

Dans le cas des utilisateurs anonymes, seuls les requêtes HTTP[S] de type GET seront admis pour la récupération d'informations sur les produits et fournisseurs, puisqu'aucun utilisateur non-identifié ne peut créer, mettre à jour ou supprimer une ressource sur le serveur.

## Ressource : Category

Ci-dessous les routes d'accès à la ressource catégorie de produit.

### Route : GET - /categories



Figure 3: Buja Market API - Route : GET /categories

**Cette route d'accès à l'API retourne les categories de produits disponibles sur la plateforme.**

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

**Le format de réponse par défaut est 'JSON', le corps de la réponse contiendra la propriété « hydra:member » les ressources demandées sous forme de table d'objets :**

```
[
  {
    "id": 0,
    "name": "string",
    "description": "string",
    "dateCreated": "2019-11-09T07:53:59.781Z",
    "platformFee": 0,
    "products": [
      {
        "id": 0,
        "title": "string",
        "resume": "string",
        "countryOrigin": "string",
        "minimumPrice": 0,
        "images": [
          {
            "title": "string",
            "alt": "string",
            "url": "string",
            "size": 0
          }
        ]
      }
    ]
  },
  {
    "image": {
      "title": "string",
      "alt": "string",
      "url": "string",
      "size": 0
    }
  }
]
```



### Route : GET - /category/{id}

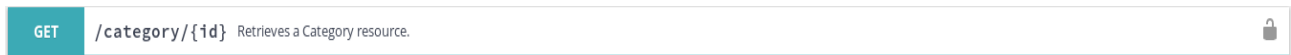


Figure 4: Buja Market API - Route : GET /category/{id}

Cette route retourne une catégorie de produits. Cette route nécessite un paramètre qui est l'identifiant unique de la catégorie de produit en base de donnée.

Cette route retourne un réponse par défaut de type JSON donc le corps contient à la propriété « hydra:member » la catégorie de produits demandé.

Ci-dessous le modèle de réponse retourné par l'API :

```
{
  "name": "string",
  "description": "string",
  "dateCreated": "2019-11-09T07:58:31.060Z",
  "isValid": true,
  "platformFee": 0,
  "products": [
    {}
  ],
  "image": {
    "place": 0,
    "title": "string",
    "alt": "string",
    "url": "string",
    "size": 0,
    "user": {}
  }
}
```

## Route : GET - /categories\_names

**GET** /categories\_names Retrieves the collection of Category resources.



Cette route retourne l'ensemble des noms de catégorie de produits.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Les noms de catégorie de produits sont souvent requis par les composants d'affichage de page et les formulaires.

Ci-dessous vous trouverez le modèle de réponse JSON retourné par l'API :

```
[  
  {  
    "o": {},  
    "id": 0,  
    "name": "string"  
  }  
]
```

## Route : GET /category/{id}/products

GET

/category/{id}/products Retrieves the collection of Product resources.



Cette route retourne l'ensemble des produits de la plateforme associé à une catégorie.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Ci-dessous le modèle de réponse JSON retourné par l'API :

```
[
  {
    "id": 0,
    "title": "string",
    "resume": "string",
    "description": "string",
    "countryOrigin": "string",
    "weight": 0,
    "length": 0,
    "width": 0,
    "height": 0,
    "minimumPrice": 0,
    "category": {
      "id": 0,
      "name": "string",
      "description": "string",
      "dateCreated": "2019-11-09T12:29:03.110Z",
      "platformFee": 0
    },
    "images": [
      {
        "id": 0,
        "place": 0,
        "title": "string",
        "alt": "string",
        "url": "string",
        "size": 0
      }
    ]
  }
]
```



```
}  
]
```

## Route : Produits

### Route : GET /products

GET

/products Retrieves the collection of Product resources.



Cette route permet d'accéder aux produits proposés sur la plateforme.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Ci-dessous vous trouverez un modèle du contenu de la réponse JSON pour cette route :

```
[  
  {  
    "id": 0,  
    "title": "string",  
    "resume": "string",  
    "description": "string",  
    "countryOrigin": "string",  
    "weight": 0,  
    "length": 0,  
    "width": 0,  
    "height": 0,  
    "minimumPrice": 0,  
    "category": {  
      "id": 0,  
      "name": "string",  
      "description": "string",  
      "dateCreated": "2019-11-09T12:36:15.372Z",  
      "platformFee": 0  
    },  
    "images": [  
      {  
        "id": 0,  
        "place": 0,  
        "title": "string",  
        "alt": "string",  
        "url": "string",  
        "size": 0  
      }  
    ]  
  }  
]
```

```
]
}
]
```

### **Route : GET /product/{id}**

GET

/product/{id} Retrieves a Product resource.



Cette route d'accès permet de récupérer un produit par son identifiant unique {id}.

Comme pour toutes les requêtes de récupération d'une ressource sur l'API, cette route accepte un paramètre obligatoire qui est le suivant :

- id : correspond à l'identifiant unique du produit en base de donnée

Ci-dessous, vous trouverez un modèle de contenu de réponse JSON retourné l'API :

```
{
  "id": 0,
  "title": "string",
  "resume": "string",
  "description": "string",
  "countryOrigin": "string",
  "weight": 0,
  "length": 0,
  "width": 0,
  "height": 0,
  "minimumPrice": 0,
  "category": {
    "id": 0,
    "name": "string",
    "description": "string",
    "dateCreated": "2019-11-09T13:05:20.758Z",
    "platformFee": 0
  },
  "images": [
    {
      "id": 0,
      "place": 0,
      "title": "string",
      "alt": "string",
      "url": "string",
      "size": 0
    }
  ]
}
```



### Route : GET /category/{id}/products/{products}/images

**GET** /category/{id}/products/{products}/images Retrieves the collection of Image resources.



Cette route permet pour une categorie donnée et un ensemble de produits donné, de récupérer toutes les images des produits de cet ensemble.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Ci-dessous vous trouverez un modèle de contenu de réponse JSON retourné par l'API :

```
[  
  {  
    "id": 0,  
    "place": 0,  
    "title": "string",  
    "alt": "string",  
    "url": "string",  
    "size": 0  
  }  
]
```

### **Route : GET /category/{id}/products/{products}/product\_suppliers**

**GET**`/category/{id}/products/{products}/product_suppliers` Retrieves the collection of SupplierProduct resources.

Cette route d'accès permet, pour une catégorie donnée (fournir l'ID de la catégorie) et un ensemble de produits donnée, de récupérer l'ensemble des informations de vente des fournisseurs de cet ensemble de produit.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- `pagination` : (booléen) permet d'activer la pagination
- `page` : permet d'indiquer la page de la collection que l'on veut voir
- `itemPerPage` : permet d'indiquer le nombre d'éléments par page

Vous trouverez un modèle de contenu de réponse JSON retourné par l'API à la page suivante.

```
[
{
  "id": 0,
  "initialPrice": 0,
  "quantity": 0,
  "additionalFee": 0,
  "additionalInformation": "string",
  "isAvailable": true,
  "isLimited": true,
  "supplier": {
    "id": 0,
    "socialReason": "string",
    "brandName": "string"
  },
  "product": {
    "id": 0,
    "title": "string",
    "resume": "string",
    "description": "string",
    "countryOrigin": "string",
    "minimumPrice": 0,
    "category": {
      "name": "string",
      "description": "string",
      "isValid": true
    },
  },
  "images": [
    {
      "place": 0,
      "title": "string",
      "alt": "string",
      "url": "string"
    }
  ]
},
  "rating": 0,
  "finalPrice": 0
}
```

### Route : GET /product/{id}/images

**GET** /product/{id}/images Retrieves the collection of Image resources.



Cette route d'accès permet de récupérer toutes les images associées à un produit donnée (en fournissant son identifiant unique).

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Ci-dessous vous trouverez le modèle de contenu de réponse JSON retourné par l'API :

```
[
  {
    "id": 0,
    "place": 0,
    "title": "string",
    "alt": "string",
    "url": "string",
    "size": 0
  }
]
```

**Route : GET /products/{id}/product\_suppliers**

Cette route permet de récupérer les informations de vente des différents fournisseurs pour un produit donnée.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Vous trouverez le modèle de contenu de réponse JSON retourné par l'API à la page suivante:



```
[
{
  "id": 0,
  "initialPrice": 0,
  "quantity": 0,
  "additionalFee": 0,
  "additionalInformation": "string",
  "isAvailable": true,
  "isLimited": true,
  "supplier": {
    "id": 0,
    "socialReason": "string",
    "brandName": "string"
  },
  "product": {
    "id": 0,
    "title": "string",
    "resume": "string",
    "description": "string",
    "countryOrigin": "string",
    "minimumPrice": 0,
    "category": {
      "name": "string",
      "description": "string",
      "isValid": true
    },
    "images": [
      {
        "place": 0,
        "title": "string",
        "alt": "string",
        "url": "string"
      }
    ]
  },
  "rating": 0,
  "finalPrice": 0
}
]
```

## Route : GET /products\_names

**GET****/products\_names** Retrieves the collection of Product resources.

Cette route permet de récupérer l'ensemble des noms de produits proposés sur la plateforme.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Ci-dessous vous trouverez le modèle de contenu de la réponse JSON retourné par l'API :

```
{
  "names": [
    {
      "o": {},
      "id": 0,
      "title": "string"
    },
  ]
}
```

## Route : Image

### Route : GET /product/{id}/images

GET

/product/{id}/images Retrieves the collection of Image resources.



Cette route permet de récupérer toutes les images associé à un produit.

Cette route nécessite un paramètre d'URL obligatoire:

- id : identifiant unique du produit dont on veut récupérer les images

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Vous trouverez le modèle de contenu de réponse JSON retourné par l'API à la page suivante:

```
[
  {
    "id": 0,
    "place": 0,
    "title": "string",
    "alt": "string",
    "url": "string",
    "size": 0
  }
]
```

## Route : GET /image/{id}

GET

/image/{id} Retrieves a Image resource.



Cette route d'accès permet de récupérer un image grâce à son identifiant unique.

Elle nécessite un paramètre d'URL :

- id : identifiant unique de l'image

Ci-dessous vous trouverez un modèle de contenu de réponse JSON retourné par l'API :

```
{  
  "id": 0,  
  "place": 0,  
  "title": "string",  
  "alt": "string",  
  "url": "string",  
  "size": 0  
}
```

## **Route : GET /category/{id}/products/{products}/images**

GET

/category/{id}/products/{products}/images Retrieves the collection of Image resources.



Cette route permet de récupérer les images pour un ensemble de produits donné et une catégorie donnée.

Cette route nécessite des paramètres GET obligatoires :

- {id } : identifiant unique de la catégorie
- {products} : ensemble d'identifiant unique sous format de chaîne de caractère. Ex : /1,2,3/

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Ci dessous vous trouverez un modèle de contenu de la réponse JSON retourné par l'API :

```
[
  {
    "id": 0,
    "place": 0,
    "title": "string",
    "alt": "string",
    "url": "string",
    "size": 0
  }
]
```

### **Route : Authentification sur l'API - POST /authentication\_token**

Cette route d'accès permet de s'authentifier sur la plateforme API.

Elle nécessite deux paramètres dans le corps (body) de la requête POST :

- email : l'email servant d'identifiant unique d'un utilisateur sur la plateforme
- password : le mot de l'utilisateur encodé sous le format Argon2i

La réponse retourne sera un clé de sécurité (token) contenant les données de l'utilisateur connectée.

Cette clé peut être décodé pour utiliser les informations contenues, ci-dessous, une liste des propriétés de l'objet contenues dans la clé :

- iat (issued at) : Date de création du token
- exp (expiration time) : Date d'expiration du token de sécurité
- username : l'identifiant unique de l'utilisateur en base de donnée (email dans notre cas)
- roles : rôles de sécurité admis pour l'utilisateur

Ci-dessous vous trouverez le modèle de contenu de réponse JSON retourné par l'API:

```
{  
  token : 'string'  
}
```

**Route : Inscription sur la plateforme - POST /register**

Cette route d'accès permet de s'inscrire sur la plateforme en fournissant des informations sur l'utilisateur connecté.

Elle nécessite plusieurs paramètres d'identification :

- lastname : Nom de l'utilisateur
- firstname : Prénom de l'utilisateur
- email : E-mail de l'utilisateur
- password : Mot de passe
- userType : Le type d'utilisation de la plateforme. Choix : Client et Fournisseur

### 5.1.2 Utilisateurs authentifiés

Dans ce chapitre, une présentation des routes d'accès à l'API accessible après identification sur la plateforme API.

Chaque utilisateur aura différents rôles de sécurité en fonction des actions que le type d'utilisateur est admis à accomplir sur la plateforme API.

Les utilisateurs seront subdivisés en différentes catégories selon ces mêmes rôles de sécurité. Ci-dessous, les catégories d'utilisateur et leurs rôles de sécurité associé.

- Membre : ROLE\_MEMBER
- Client : ROLE\_CUSTOMER, ROLE\_MEMBER, ROLE\_ALLOWED\_TO\_SWITCH
- Fournisseur : ROLE\_SUPPLIER, ROLE\_MEMBER, ROLE\_ALLOWED\_TO\_SWITCH
- Administrateur : ROLE\_ADMIN, ROLE\_SUPPLIER, ROLE\_CUSTOMER, ROLE\_MEMBER, ROLE\_ALLOWED\_TO\_SWITCH
- Administrateur du système : ROLE\_SUPER\_ADMIN, ROLE\_ADMIN, ROLE\_SUPPLIER, ROLE\_CUSTOMER, ROLE\_MEMBER, ROLE\_ALLOWED\_TO\_SWITCH

Chaque rôle de sécurité permet d'accomplir des actions spécifiques sur la plateforme. Le rôle de sécurité « ROLE\_ALLOWED\_TO\_SWITCH » permet de changer de rôle de sécurité en fonction de l'action à accomplir.

Dans la suite de ce chapitre, vous trouverez une présentation détaillée des routes d'accès à l'API pour chaque catégorie d'utilisateur.



[Membres \(Utilisateurs authentifiés\)](#)

Route : Adresse

Route : Compte bancaire

Route : Ensemble de commande

Route : Ensemble de livraison

Route : Fournisseurs

Route: Image

Route : Produits

Route : Produits fournisseurs

Route : Transporteur

## Clients

Route : Comment

Route : Commande retourné

Route : Facture client

Route : Facture remboursement

Route : Favoris

### **Route : Paiement**

Route : Panier de commande

## Professionnels

Route : Facture fournisseur

Route : Ensemble de livraison

Route : Détail de livraison

### **Route : Paiement**

### **5.1.3 Administrateurs**

Route : Admin

## Route : Image

### Route : GET /images

**GET****/images** Retrieves the collection of Image resources.

Cette route permet de récupérer toutes les images de l'application. Cette route ne nécessite pas de paramètre d'URL.

Comme tous les requêtes de récupérations de collections, les paramètres suivants sont optionnels pour paramétrer la pagination de la collection :

- pagination : (booléen) permet d'activer la pagination
- page : permet d'indiquer la page de la collection que l'on veut voir
- itemPerPage : permet d'indiquer le nombre d'éléments par page

Vous trouverez le modèle de contenu de réponse JSON retourné par l'API à la page suivante:

Route : User

## **5.2. Interface client**

### **5.2.1 Utilisateurs anonymes**

[Interface d'accueil](#)

[Interface d'authentification sur la plateforme](#)

[Interface d'inscription en tant que membre ou fournisseur de produits](#)



[Interface de présentation des produits disponibles à la vente](#)

[Interface de classification des produits de la plateforme](#)

## **5.2.2 Utilisateurs authentifiés**

### **5.2.2.1 Clients**

[Interface d'ajout au panier de commande de produits](#)

[Interface d'ajout de produits aux favoris](#)

[Interface de validation de commande](#)

[Interface de validation d'adresse de livraison](#)

[Interface de validation de la commande définitive](#)

[Interface de paiement en ligne](#)



[Interface de facturation de la commande](#)

[Interface de l'ensemble de commande](#)

[Interface des commandes réalisés d'un client](#)

[Interface des livraisons à destination des clients](#)

[Interface de communication avec le fournisseur d'un produit acheté](#)

[Interface de réclamation par rapport à un produit commandé](#)

[Interface de rétractation sur une commande](#)

#### **5.2.2.2 Professionnels**

[Interface d'ajout de produits proposés à la vente](#)



[Interface des produits commandés proposés par un fournisseur](#)

[Interface des livraisons de produits commandés proposés par un fournisseur](#)

## **5.3 Interface administrateur**

### **5.3.1 Administrateurs**

[Interface des inscriptions en cours sur la plateforme](#)

[Interface des clients de la plateforme](#)

[Interface des fournisseurs](#)

[Interface des produits de la plateforme](#)

[Interface des informations fournisseurs pour les produits](#)

[Interface des images de la plateforme](#)



[Interface des catégories de produits](#)

[Interface des paniers de commande](#)

[Interface des produits favoris des clients](#)

## [Interface des ensembles de commandes](#)

[Interface des détails de commande](#)

[Interface des commandes retournées](#)

[Interface des commandes annulées](#)

[Interface des paiements de la plateforme](#)



[Interface des remboursements de la plateforme](#)

[Interface de l'ensemble des factures](#)

[Interface de la facturation client](#)

[Interface de la facturation fournisseur](#)

[Interface de la facturation de remboursement](#)

## [Interface des transporteurs](#)

## [Interface des ensembles de livraisons](#)

[Interface des détails de livraisons](#)



[Interface des adresses utilisateurs](#)

[Interface des comptes bancaires des utilisateurs](#)

[Interface des forums de communication de la plateforme](#)

[Interface des messages dans les forums](#)

### **5.3.2 Administrateur système**

Interface des utilisateurs de la plateforme

[Interface des adresses utilisateurs](#)

[Interface des comptes bancaires des utilisateurs](#)

[Interface des administrateurs](#)



## 6. Description de la base de données

Ce chapitre présente une description de la base de données qui sera implémentée pour les nécessités du projet.

La description sera subdivisée en 2 parties :

1. Modélisation conceptuelle de la base de données :  
Elle contient une description des étapes de construction de la modélisation. Le modèle conceptuel sera divisé selon le type d'actions utilisateurs qui sera accompli sur la plateforme. Voici les subdivisions choisies : inscription, publication, commande, paiement, livraison, évaluation, réclamation, remboursement.
2. Modélisation physique des entités : elle contient une description détaillée de l'ensemble des tables de la base de données.

## 6.1 Modélisation conceptuelle de la base de données

### 6.1.1 Inscription (Utilisateurs de la plateforme)

La première étape sera d'implémenter les utilisateurs de la plateforme dans le modèle de la base de données.

La table **User** contient tous les utilisateurs de la plateforme qui auront fourni leurs informations d'identification (nom complet, email, mot de passe, type d'utilisateur)

Ceux-ci seront subdivisés en 3 tables selon leur utilisation de la plateforme :

➔ **Customer**

Cette table contient les utilisateurs qui désirent acquérir un bien via la plateforme ainsi que les informations spécifiques aux clients de la plateforme

➔ **Supplier** (Vendeur)

Cette table contient les vendeurs qui désirent présenter leurs produits à la vente sur la plateforme ainsi que les informations relatives aux vendeurs de la plateforme.

➔ **Admin** (Administrateur)

Cette table contient les responsables de la gestion de la plateforme ainsi que les informations spécifiques aux administrateurs de la plateforme.

De plus, les tables suivantes seront ajoutées pour contenir des informations complémentaires sur les utilisateurs de la plateforme :

➔ **Address**

Cette table contiendra les adresses de livraison des utilisateurs (clients et vendeurs) ainsi que les adresses de livraison des entreprises de transport de fret.

Chaque client de la plateforme pourra ainsi préciser l'adresse de livraison de chaque commande dans les adresses enregistrées ou préciser une nouvelle adresse de livraison.

➔ **Bank Account**

Cette table contiendra les informations bancaires non-sécurisées des utilisateurs authentifiés. Ainsi on pourra procéder au paiement des fournisseurs et au remboursement des clients.

Voir l'annexe « MCD final »

### 6.1.2 Publication

Cette section intègre les tables qui seront utilisées lors de la publication d'articles par les vendeurs de la plateforme.

Les tables ajoutées sont les suivantes :

➔ **Product**

Cette table contient tous les produits publiés par les vendeurs.

Chaque vendeur pourra publier autant de produits qu'il désire mettre en vente.

Un produit pourra être commercialisé par plusieurs vendeurs.

Cette table contient les informations détaillées du produit.

➔ **Image**

Cette table contient toutes les images illustrant les produits disponibles à la vente ainsi que les images de profil des utilisateurs.

Chaque produit sera obligatoirement illustré par une ou plusieurs images pour être publié à la vente.

Les utilisateurs pourront choisir une image de profil qui sera affichée à côté de leur nom d'utilisateur.

➔ **Category** (catégorie)

Cette table contient les catégories de produits disponibles à la vente.

Chaque produit publié sera obligatoirement classé dans une catégorie de produits.

➔ **Supplier product**

Cette table contiendra les informations relatives à la vente des produits publiés par les vendeurs.

Étant donné que chaque produit pourra être vendu par plusieurs vendeurs et que chaque vendeur commercialise plusieurs produits, cette table fera la relation entre les vendeurs et les produits vendus par ceux-ci.

Ainsi cette table répertorie les produits vendus par chaque vendeur ainsi que les informations relatives à la vente.

Voir l'annexe « MCD final »

### 6.1.3 Commande

Cette étape intègre les tables qui seront utilisées lors de commandes de produits via la plateforme.

Les tables suivantes seront ajoutées :

➔ **Order detail**

Cette table contient les informations détaillées relatives à la commande de chaque produit. Chaque détail de commande concernera un seul produit proposé par un vendeur (***supplier product***).

➔ **Order set**

Cette table contient les ensembles de commande des clients.

Un ensemble de commande regroupera les détails des produits commandés pour chaque session d'achat.

Voir l'annexe « MCD final »

## 6.1.4 Paiement

Cette section intègre dans le modèle conceptuel les tables qui seront employées dans le processus de paiement de la plateforme.

Les tables suivantes seront intégrées :

### ➔ **Bill**

Cette table contient les informations générales liées aux factures. Les informations fournies par le système de paiement Stripe devront être intégrées pour chaque facture.

Les tables de facturation seront subdivisées selon les types suivants :

#### ➤ **Bill Customer**

Cette table contient les informations spécifiques aux factures à destination des clients.

#### ➤ **Bill Supplier**

La table « Supplier bill » contient les informations relatives aux factures à destination des vendeurs.

#### ➤ **Bill Refund**

Cette table contient uniquement les factures liées au remboursement de commandes annulées ou retournées.

(Voir la section « Remboursement »)

### ➔ **Payment**

Toutes les transactions de la plateforme devront être enregistrées dans cette table.

Ceci comprend les informations du destinataire, le montant total, la référence à une facture pour chaque transaction. Il faudra prendre en compte les informations utiles retournées par le système de paiement Stripe lors de la validation du paiement.

Voir l'annexe « MCD final »

### 6.1.5 Livraison

Cette section intègre les tables relatives au processus de livraison des commandes.

Les tables suivantes seront ajoutées :

➔ **Delivery set**

Cette table contient les informations spécifiques à l'ensemble des éléments constitutifs d'une commande à livrer. Les informations de cette table pourront indiquer le statut de la livraison d'une commande ainsi que le choix du transporteur responsable.

➔ **Delivery detail**

Cette table contient les informations spécifiques au détail de livraison de chaque élément constituant l'ensemble d'une commande. On y trouvera la référence de suivi attribuée au colis par l'exportateur responsable ainsi que le statut de la livraison de chaque élément de la commande et une preuve d'envoi fournie par les vendeurs.

➔ **Shipper**

Cette table contient les informations sur les sociétés de transport de fret et les options de transport offertes par celles-ci.

Chaque société de transport sera associée à une ou plusieurs adresses si l'adresse de livraison du fret aérien et maritime sont différentes.

Voir l'annexe « MCD final »

### 6.1.6 Relation clients - vendeurs

La section suivante intègre les tables relatives à la relation client-vendeur. Ceci comprend l'évaluation des produits proposés par des commentaires publics et l'ajout à une liste de souhaits de leurs produits favoris.

Les tables suivantes seront intégrées :

➔ **Comment**

Cette table contient les évaluations et commentaires des clients sur les produits commandés.

Chaque client pourra publier une évaluation et un commentaire du produit qu'il a acheté.

➔ **Favorite**

Cette table contient les produits proposés par les vendeurs que les clients ont adoré.

Chaque client pourra enregistrer les produits qu'il apprécie dans une liste de souhaits pour un éventuel achat futur.

Voir l'annexe « MCD final »

### 6.1.7 Relation utilisateurs - administrateurs

La section suivante intègre les tables relatives à la relation des utilisateurs authentifiés de la plateforme avec les administrateurs. Un forum de messagerie sera mis en place pour couvrir tous les types de messages qui pourraient être échangés entre les utilisateurs et les administrateurs.

Les tables suivantes seront intégrées :

➔ **Forum**

Cette table contient les informations sur les échanges par messages entre les utilisateurs de la plateforme et les administrateurs.

Chaque utilisateur de la plateforme pourra contacter les administrateurs pour demander des informations, faire une réclamation et signaler un abus.

➔ **Message**

Cette table contiendra tous les messages écrits lors des échanges entre les utilisateurs de la plateforme et les administrateurs via le forum.

Voir l'annexe « MCD final »



### 6.1.8 Remboursement

Dans cette dernière section, les tables relatives au processus de remboursement des clients seront intégrées.

Le remboursement des clients se fera dans 2 cas :

➔ **Withdrawal**

Cette table contient toutes les données relatives aux rétractations sur des commandes déjà payées par les clients.

Les rétractations seront possibles seulement pendant 14 jours à partir de la date de paiement.

Chaque rétractation, dans les délais impartis par la loi, sera associée à une facture de remboursement avant paiement.

➔ **Order returned**

Cette table contient les données relatives aux commandes retournées par les clients.

Chaque retour de marchandise devra contenir une preuve d'envoi du produit par le client ainsi qu'une preuve de réception de la marchandise fournie par l'administrateur avant sa facturation et son paiement.

Voir l'annexe « MCD final »

## 6.2 Modélisation physique des entités

Ce chapitre présente de façon détaillée les attributs de l'ensemble des tables de la base de données. Ces tables sont listées par ordre alphabétique.

### 6.2.1 Address





La table **Address** sera en relation avec les tables suivantes :

- ➔ **User** et **Shipper** puisqu'elle contient les adresses des utilisateurs et des entreprises de fret nécessaires pour les livraisons.

Chaque utilisateur pourra enregistrer les adresses utilisées pour la livraison de ses produits.

Ci-dessous se trouve une liste des attributs de la table.

Attributs	Description
ID (identifiant)	Identifiant unique de l'adresse
Location name	Nom du lieu (résidence, siège social, centre logistique, adresse de facturation, adresse de livraison)
Street	Nom de la rue
Number	Numéro d'appartement / maison
State	Nom de l'état, de la région
Town	Nom de la ville de livraison
Zip code	Code postal de la ville
Country	Pays

address		
	<i>Pk_Primary</i>	
	<u>id</u>	AUTO_INCREMENT
	<i>Attributs</i>	
	location_name	VARCHAR (255)
	street	VARCHAR (255)
	number	VARCHAR (30)
	state	VARCHAR (255)
	town	VARCHAR (255)
	zip_code	VARCHAR (30)
	country	VARCHAR (255)
	<i>Fk_Foreign</i>	
	id_user	INT
	id_shipper	INT (11)




## 6.2.2 Admin

La table **Admin** aura les relations suivantes :

- ➔ Elle héritera des attributs de la table **User** pour les utilisateurs qui seront administrateurs de la plateforme.
- ➔ Une relation One-To-Many avec la table **Forum** car chaque administrateur pourra répondre aux questions posées dans le forum d'aide afin d'assister tous les utilisateurs de la plateforme
- ➔ Une relation One-To-Many avec la table **Order returned** car chaque commande retournée sera validée par un administrateur.

Ci-dessous se trouve une liste des attributs de la table.




Attributs	Description
ID (identifiant)	Identifiant unique de l'administrateur ( ID user)
Nb Refund validated	Nombre de remboursements validés
Nb Issue resolved	Nombre de problèmes résolus dans le forum

admin	
	<i>Pk_Primary</i> <u>id</u> INT
	<i>Attributs</i>
	nbRefundValidated INT
	nbIssueResolved INT
	password VARCHAR (255)
	firstname VARCHAR (255)
	lastname VARCHAR (255)
	nb_error_connection INT
	banned BOOL
	signin_confirmed BOOL
	date_registration DATE
	language VARCHAR (255)
	currency VARCHAR (5)
	<i>Idx_Index</i> <u>email</u> VARCHAR (150)
	<i>Fk_Foreign</i> <u>id_image</u> INT

### 6.2.3 Bank Account

La table **Bank account** sera en relation avec la table **User** car elle contiendra les informations non-sécurisées des comptes bancaires des utilisateurs de la plateforme.

Attributs	Description
ID (identifiant)	Identifiant unique du compte bancaire
ID card	ID de la carte bancaire dans Stripe
Brand	Nom de l'institution bancaire
Country code	Code du pays d'émission de la carte
Last 4	4 derniers chiffres du numéro de la carte bancaire
Expiry month	Mois d'expiration de la carte bancaire
Expiry year	Année d'expiration de la carte bancaire
Fingerprint	Empreinte digitale de la carte
Funding	Type de financement associé à la carte (crédit, débit)
Account balance	Solde du compte

bank_account		
	<i>Pk_Primary</i>	
	<u>id</u>	AUTO_INCREMENT
	<i>Attributs</i>	
	brand	VARCHAR (255)
	country_code	VARCHAR (3)
	last_4	INT (4)
	expiry_month	INT (2)
	expiry_year	INT (4)
	fingerprint	VARCHAR (50)
	funding	VARCHAR (50)
	account_balance	FLOAT
	<i>Idx_Index</i>	
	<u>id_card</u>	VARCHAR (255)
	<i>Fk_Foreign</i>	
	<u>id_user</u>	INT


## 6.2.4 Bill

La table **Bill** aura les relations suivantes :

- ➔ Les tables **Bill customer**, **Bill supplier** et **Bill refund** hériteront des attributs de la table **Bill**.
- ➔ Une relation One-To-One avec la table **Payment** car chaque facture sera associée à une transaction commerciale à destination d'un utilisateur ou de la plateforme.

Ci-dessous se trouve une description des attributs de la table.





Attributs	Description
ID (identifiant)	Identifiant unique de la facture
Status	Statut de la facture (void, uncollectible, paid)
Date created	Date de création de la facture
Date payment	Date de paiement de la facture
Currency used	Type de devise utilisée (BIF, USD, EUR)
VAT rate used	Taux TVA utilisé
Total exclu tax	Montant total hors TVA
Total incl tax	Montant total TVA comprise
Url	Url de la facture sous format PDF

bill	
	<i>Pk_Primary</i>
<u>id</u>	INT
<i>Attributs</i>	
status	VARCHAR (255)
date_created	DATETIME
date_payment	DATETIME
currency_used	VARCHAR (3)
vat_rate_used	FLOAT
total_excl_tax	DOUBLE
total_incl_tax	DOUBLE
url	VARCHAR (60)

## 6.2.5 Bill Customer

La table **Bill customer** hérite des attributs de la table **Bill**. De plus, **Bill customer** sera en relation avec la table **Order set** car chaque facture client sera associée à un ensemble de commande effectuée lors d'une session d'achat par un client.

Attributs	Description
ID (identifiant)	Identifiant unique de la facture client (ID user)
Additional cost	Charges additionnelles
Additional fee	Frais additionnels
Additional information	Informations sur les frais et charges additionnels
Total shipping cost	Montant total du coût du transport

bill_customer	
	<i>Pk_Primary</i>
	<u>id</u> INT
	<i>Attributs</i>
	additional_cost DOUBLE
	additional_fee INT
	total_shipping_cost DOUBLE
	status VARCHAR (255)
	date_created DATE
	date_payment DATE
	vat_rate_used INT
	total_excl_tax DOUBLE
	total_incl_tax DOUBLE
	url VARCHAR (60)
	<i>Fk_Foreign</i>
	id_order_global INT
	id_customer INT

### 6.2.6 Bill Refund

La table **Bill refund** hérite des attributs de la table **Bill**. De plus, la table **Bill refund** sera en relation avec les tables **Order returned** et **Withdrawal** (annulation).




En effet, chaque facture de remboursement sera créée dans le cas où une annulation de commande est requise ou lorsqu'un retour de marchandise est effectué par un client.

Attributs	Description
ID (identifiant)	Identifiant unique de la facture de remboursement (ID Facture)
Reason	Raison du remboursement (annulation, retour de marchandise)
Description	Description si nécessaire
Additional cost	Charges additionnelles
Additional fee	Frais additionnels
Additional_information	Informations sur les charges et frais additionnels

## 6.2.7 Bill Supplier

la table **Bill supplier** hérite des attributs de la table **Bill**. En outre, cette table sera en relation avec la table **Order detail**. En effet, chaque facture vendeur sera associée au détail de commande d'un produit par un client.

Attributs	Description
ID (identifiant)	Identifiant unique de la facture vendeur (ID facture)
Delivery cost	Coût de livraison du produit vers l'exportateur



bill_supplier	
	<i>Pk_Primary</i>
<u>id</u>	INT
	<i>Attributs</i>
delivery_cost	DOUBLE
status	VARCHAR (255)
date_created	DATE
date_payment	DATE
vat_rate_used	INT
total_excl_tax	DOUBLE
total_incl_tax	DOUBLE
url	VARCHAR (60)
	<i>Fk_Foreign</i>
	id_order_detail INT
	id_supplier INT



### 6.2.8 Category

La table **Category** aura une relation One-To-Many avec la table **Product** car chaque catégorie de produits pourra contenir autant de produits qu'il est nécessaire.

Attributs	Description
ID (identifiant)	Identifiant unique de la catégorie
Name	Nom de la catégorie (unique)
Description	Description de la catégorie
Date created	Date de création
Is valid	Validé par un administrateur

category		
	<i>Pk_Primary</i> <u>id</u>	INT
	<i>Attributs</i>	
	description	TEXT
	date_created	DATE
	is_valid	BOOL
	<i>Idx_Index</i>	
	name	VARCHAR (255)




## 6.2.9 Customer

La table **Customer** aura une relation d'héritage avec la table **User** car chaque client héritera des attributs d'un utilisateur authentifié.

La table **Customer** aura les relations suivantes :

- ➔ **Customer** aura une relation One-To-Many avec la table **Order set** car chaque client pourra effectuer des commandes composées de plusieurs produits.
- ➔ **Customer** aura une relation One-To-Many avec la table **Comment** car chaque client pourra commenter tous les produits qu'il a acheté.
- ➔ **Customer** aura une relation One-To-Many avec la table **Favorite** car chaque client pourra ajouter tous les produits qu'il a aimé dans une liste de favoris (ou souhaits)

Attributs	Description
ID (identifiant)	Identifiant unique du client (ID user)
Customer key	Clé client (unique)
Nb Abuse Identified	Nombre d'abus identifiés
Average Rating	Moyenne du score donné aux produits commandés
Nb Order Completed	Nombre de commandes complétées
Nb Order Withdrawn	Nombre de commandes annulées

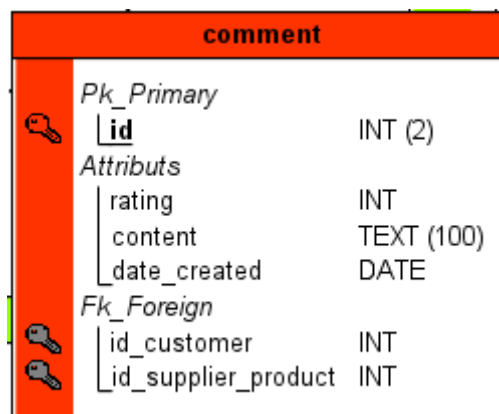
customer		
	<i>Pk_Primary</i>	
	<u>id</u>	INT
	<i>Attributs</i>	
	customerKey	VARCHAR (100)
	nbAbuseIdentified	INT
	averageRating	FLOAT
	nbOrderCompleted	INT
	nbOrderWithdrawn	INT
	password	VARCHAR (255)
	firstname	VARCHAR (255)
	lastname	VARCHAR (255)
	nb_error_connection	INT
	banned	BOOL
	signin_confirmed	BOOL
	date_registration	DATE
	language	VARCHAR (255)
	currency	VARCHAR (5)
	<i>Idx_Index</i>	
	<u>email</u>	VARCHAR (150)
	<i>Fk_Foreign</i>	
	<u>id_image</u>	INT

## 6.2.10 Comment

La table **Comment** aura les relations suivantes :

- ➔ **Comment** aura une relation Many-To-One avec la table **Client** car chaque commentaire sera associé à un seul client
- ➔ **Comment** aura une relation Many-To-One avec la table **Supplier products** car chaque produit d'un vendeur pourra être commenté à plusieurs reprises.

Attributs	Description
ID (identifiant)	Identifiant unique du commentaire
Rating	Évaluation du produit commenté de 1-10
Content	Contenu du commentaire
Date created	Date de création







## 6.2.11 Delivery Detail

La table **Delivery detail** aura les relations suivantes :

- ➔ Une relation Many-To-One avec la table **Delivery set** puisque plusieurs livraisons d'un ou plusieurs vendeurs à l'expéditeur peuvent être effectuées pour une commande client.
- ➔ Une relation One-To-One avec la table **Order detail** car chaque élément d'une commande sera associé à un détail de livraison contenant les informations de sa livraison au client.

Attributs	Description
ID (identifiant)	Identifiant unique du détail de livraison
Reference	Référence attribuée au colis par l'expéditeur
Description	Description du contenu du colis
Date created	Date d'enregistrement du détail de livraison
Is shipped	Est expédié ( booléen )
Is received	Est réceptionné ( booléen )
Attachment file	Preuve d'envoi du colis




delivery_detail		
	<i>Pk_Primary</i>	
	<u>id</u>	INT
	<i>Attributs</i>	
	description	TEXT
	date_created	DATE
	is_shipped	BOOL
	is_received	BOOL
	attachment_file	VARCHAR (255)
	<i>Idx_Index</i>	
	<u>reference</u>	VARCHAR (50)
	<i>Fk_Foreign</i>	
	id_order_detail	INT
	id_delivery_global	INT

## 6.2.12 Delivery set

La table **Delivery set** aura les relations suivantes :

- ➔ Une relation One-To-Many avec la table **Delivery detail** puisqu'une commande peut contenir plusieurs produits qui seront livrés séparément ou pas. Ainsi pour un ensemble de commande, il pourrait y avoir plusieurs livraisons effectuées par un ou plusieurs vendeurs à l'expéditeur.
- ➔ Une relation One-To-One avec la table **Order set** puisque chaque commande d'un ensemble de produits sera associée à une livraison de cet ensemble de produits par l'expéditeur choisi par le client.
- ➔ Une relation One-To-One avec la table **Shipper** car chaque livraison d'un ensemble de produits d'une commande sera réalisé par un expéditeur choisi par le client.

Attributs	Description
ID (identifiant)	Identifiant unique de la livraison client
Shipping cost	Coût de l'expédition de tous les éléments de la commande
Date created	Date de création de la livraison
All shipped	Tous les éléments de la commande ont été expédiés (booléen)
All received	Tous les éléments de la commande ont été réceptionnés (booléen)

delivery_global		
	Pk_Primary	
	<u>id</u>	INT
	Attributs	
	shipping_cost	INT
	date_created	DATE
	all_shipped	BOOL
	all_received	BOOL
	Fk_Foreign	
	id_order_global	INT
	id_shipper	INT (11)

### 6.2.13 Favorite

La table ***Favorite*** fera la relation entre la table ***Client*** et ***Supplier products*** en conservant les identifiants des produits des vendeurs que les clients ont adoré.

Attributs	Description
ID (identifiant)	Identifiant unique du produit ajouté aux favoris

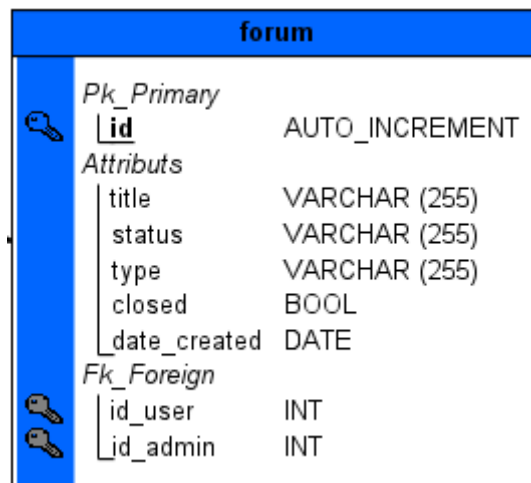
Favorite		
	<i>Pk_Primary</i> <u>id</u>	INT
	<i>Fk_Foreign</i> id_supplier_product	INT
	id_customer	INT

## 6.2.14 Forum

La table **Forum** aura les relations suivantes :

- ➔ Une relation One-To-One avec la table **User** car chaque session de discussion du forum sera ouverte par un utilisateur de la plateforme.
- ➔ Une relation One-To-One avec la table **Admin** car un administrateur pourra répondre à une session de discussion du forum
- ➔ Une relation One-To-Many avec la table **Message** car plusieurs messages seront échangés lors d'une session de discussion.



Attributs	Description
ID (identifiant)	Identifiant unique du produit ajouté aux favoris
Title	Titre du sujet de la discussion
Status	Statut de la discussion ( pas de réponse, en cours, résolu )
Type	Type de discussion ( information, litige, signalement d'abus )
Closed	Indique si la discussion est close (booléen)
Date created	Date de création de la discussion



### 6.2.15 Image

La table **Image** aura une relation Many-To-One avec la table **Product** car un produit pourra être illustré par plusieurs images.

Attributs	Description
ID (identifiant)	Identifiant unique de l'image
Place	Ordre d'affichage des images (1-10)
Title	Titre de l'image
Alt	Titre alternatif de l'image
Url	Url de la ressource image
Size	Taille de l'image

image	
	<i>Pk_Primary</i> <u>id</u> INT
	<i>Attributs</i>
	place INT
	title VARCHAR (255)
	alt VARCHAR (255)
	url VARCHAR (255)
	size INT
	<i>Fk_Foreign</i> <u>id_product</u> INT (11)






## 6.2.16 Message

La table **Message** sera en relation avec les tables suivantes :

- ➔ une relation Many-To-One avec la table **Forum** car plusieurs messages seront échangés dans une session de discussion du forum par les utilisateurs.
- ➔ Une relation One-To-One avec la table **User** car un message sera écrit par un utilisateur.

Attributs	Description
ID (identifiant)	Identifiant unique du message
Content	Contenu du message
Date created	Date de création
Attachment File	Fichier joint au message
Attachment URL	URL joint au message




message	
	<i>Pk_Primary</i> <u>id</u> INT
	<i>Attributs</i>
	content TEXT
	date_created DATE
	attachment_url VARCHAR (255)
	attachment_file VARCHAR (255)
	<i>Fk_Foreign</i>
	id_forum INT (11)
	id_user INT

## 6.2.17 Order Detail

La table **Order Detail** aura les relations suivantes :

- ➔ Une relation One-To-One avec la table **Supplier Product** car chaque détail de commande concernera un seul produit.
- ➔ Une relation One-To-One avec la table **Delivery detail** car chaque détail de commande pourra être livré par un vendeur à l'expéditeur séparément du reste de la commande.
- ➔ Une relation One-To-One avec la table **Order returned** car chaque détail de commande pourra être retourné par le client s'il n'est pas satisfait.
- ➔ Une relation One-To-One avec la table **Withdrawal** car chaque détail de commande pourra être annulé dans les 14 jours suivant son paiement.

Attributs	Description
ID (identifiant)	Identifiant unique du détail de commande
Status	Statut de l'élément d'une commande (en attente, en cours, livrée, terminée, annulée, gelée)
Quantity	Quantité du produit
Unit cost	Prix unitaire
Total cost	Coût total



order_detail		
	<i>Pk_Primary</i>	
	<u>id</u>	INT
	<i>Attributs</i>	
	status	VARCHAR (255)
	quantity	INT (60)
	unit_cost	DOUBLE
	total_cost	DOUBLE
	<i>Fk_Foreign</i>	
	id_order_global	INT
	id_supplier_product	INT

## 6.2.18 Order Set

La table **Order set** sera en relation avec les tables suivantes :

- ➔ Une relation One-To-Many avec la table **Order detail** car un ensemble de commande effectuée par un client contiendra plusieurs détails de commande de produits.
- ➔ Une relation One-To-One avec la table **Delivery set** car chaque ensemble de commande sera associée à une livraison vers un client.
- ➔ Une relation Many-To-One avec la table **Customer** car un client pourra effectuer plusieurs session d'achat sur la plateforme.

Attributs	Description
ID (identifiant)	Identifiant unique de l'ensemble de commande
Total weight	Poids total
NbPackage	Nombre de paquets
Total cost	Coût total

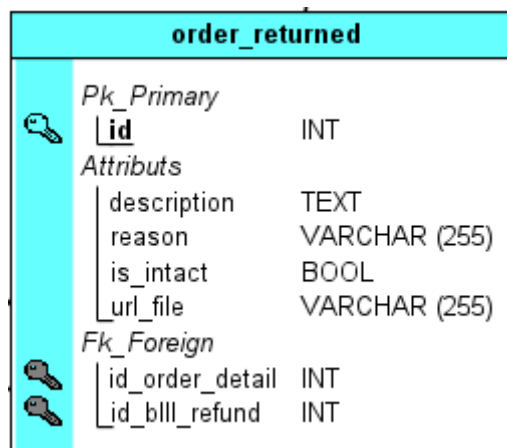
order_global	
	<i>Pk_Primary</i> <u>id</u> INT
	<i>Attributs</i>
	date_created DATETIME
	total_weight FLOAT
	nb_package INT
	total_cost FLOAT
	<i>Fk_Foreign</i> <u>id_customer</u> INT

## 6.2.19 Order Returned

La table **Order returned** aura les relations suivantes :

- ➔ Une relation One-To-One avec la table **Order detail** puisque chaque retour de commande correspondra au détail d'une commande existante en base de données.
- ➔ Une relation One-To-One avec la table **Bill refund** car chaque retour de marchandise pourra être remboursé après la production d'une facture de remboursement.




Attributs	Description
ID (identifiant)	Identifiant unique du produit
Reason	Raison du retour de marchandise
Description	Description de la raison du retour de marchandise
Is intact	Est retourné intact (booléen)
URL file	Preuve de retour de la commande



## 6.2.20 Payment

La table **Payment** aura une relation One-To-One avec la table **Bill** car chaque paiement enregistré sera associé à une facture.

Attributs	Description
ID (identifiant)	Identifiant unique du paiement
Amount	Montant perçu
Amount refund	Montant remboursé
Currency	Devise utilisée (USD,EUR)
Balance transaction	Attribut Stripe API : ID of the balance transaction that describes the impact of this charge on your account balance (not including refunds or disputes)
Status	Statut du paiement
Description	Description du paiement (destinataire)
Reference	Référence du paiement (ex : id user/id commande/id facture)
Source	Source de financement de la carte (débit, crédit)
Date created	Date de création
Email receipt	Email du destinataire


payment		
	<i>Pk_Primary</i> <u>id</u>	INT
	<i>Attributs</i>	
	date_created	DATETIME
	currency	VARCHAR (5)
	description	TEXT
	status	VARCHAR (255)
	amount	DOUBLE
	amount_refund	DOUBLE
	balance_transaction	VARCHAR (255)
	email_receipt	VARCHAR (255)
	source	VARCHAR (30)
	<i>Idx_Index</i>	
	reference	VARCHAR (3)
	<i>Fk_Foreign</i>	
	id_bill	INT

## 6.2.21 Product

La table **Product** aura les relations suivantes :

- ➔ Une relation One-To-Many avec la table **Image** car chaque produit sera illustré par une ou plusieurs images
- ➔ Une relation Many-To-One avec la table **Category** car chaque produit sera classé dans une catégorie de produits.
- ➔ Une relation One-To-Many avec la table **Supplier product** car chaque produit publié pourra être vendu par plusieurs vendeurs.

Attributs	Description
ID (identifiant)	Identifiant unique du produit
Title	Titre du produit (unique)
Resume	Résumé de la description du produit
Description	Description détaillée du produit
Country of origin	Pays d'origine du produit
Weight	Poids du produit
Length	Longueur du produit
Width	Largeur du produit
Height	Hauteur du produit




product		
	<i>Pk_Primary</i> <u>id</u>	AUTO_INCREMENT
	<i>Attributs</i>	
	resume	VARCHAR (255)
	description	TEXT
	country_of_origin	VARCHAR (255)
	weight	FLOAT
	length	FLOAT
	width	FLOAT
	height	FLOAT
	<i>Idx_Index</i> <u>title</u>	VARCHAR (255)
	<i>Fk_Foreign</i> <u>id_category</u>	INT

## 6.2.22 Shipper

La table **Shipper** aura les relations suivantes :

- ➔ Une relation One-To-Many avec la table **Address** car pour chaque service de livraison il faudra fournir une adresse de collecte des colis.
- ➔ Une relation One-To-One avec la table **Delivery** puisque chaque livraison sera effectuée par un service de l'expéditeur

Attributs	Description
ID (identifiant)	Identifiant unique de l'expéditeur (transporteur)
Name	Nom du transporteur
Social reason	Raison sociale de l'exportateur
Trade register number	Numéro de registre de commerce
VAT number	Numéro TVA
Service	Service du transporteur
Description	Description du service offert
Delivery commitment	Engagement de livraison
Contact number	Numéro de téléphone de la personne de contact




shipper		
	<i>Pk_Primary</i>	
	<u>id</u>	AUTO_INCREMENT
	<i>Attributs</i>	
	name	VARCHAR (255)
	trade_register_number	VARCHAR (255)
	vat_number	VARCHAR (255)
	description	TEXT
	delivery_commitment	TEXT
	contact_number	VARCHAR (60)
	<i>Idx_Index</i>	
	social_reason	VARCHAR (255)
	service	VARCHAR (255)

## 6.2.23 Supplier

La table **Supplier** aura les relations suivantes :

- ➔ Une relation d'héritage avec la table **User** car les vendeurs héritent des attributs des utilisateurs authentifiés.
- ➔ Une relation One-To-Many avec la table **Supplier product** car chaque vendeur proposera plusieurs produits publiés sur la plateforme à la vente.

Attributs	Description
ID (identifiant)	Identifiant unique du vendeur
Social reason	Raison sociale
Trade register number	Numéro de registre de commerce
VAT number	Numéro TVA du vendeur (VAT = Value Added Tax)
Contact fullname	Nom complet de la personne de contact
Contact phone number	Numéro de téléphone de la personne de contact
Contact e-mail	E-mail de contact
Site web	Site internet

supplier		
	<i>Pk_Primary</i> <u>id</u>	INT
	<i>Attributs</i>	
	social_reason	VARCHAR (255)
	trade_register_number	VARCHAR (255)
	vat_number	VARCHAR (3)
	contact_fullname	VARCHAR (255)
	contact_phone_number	VARCHAR (40)
	contact_email	VARCHAR (255)
	website	VARCHAR (255)
	password	VARCHAR (255)
	firstname	VARCHAR (255)
	lastname	VARCHAR (255)
	nb_error_connection	INT
	banned	BOOL
	signin_confirmed	BOOL
	date_registration	DATE
	language	VARCHAR (255)
	currency	VARCHAR (5)
	<i>Idx_Index</i>	
	email	VARCHAR (150)
	<i>Fk_Foreign</i>	
	id_image	INT






## 6.2.24 Supplier Product

La table **Supplier product** aura les relations suivantes :

- ➔ Une relation Many-To-One avec la table **Supplier** car un vendeur proposera plusieurs produits à la vente
- ➔ Une relation Many-To-One avec la table **Product** car un produit sera proposé par plusieurs vendeurs
- ➔ Une relation One-To-Many avec la table **Comment** car plusieurs commentaires pourront concerner un produit proposé par un vendeur
- ➔ Une relation One-To-Many avec la table **Favorite** car plusieurs listes de favoris pourront mentionner un produit proposé par un vendeur

Attributs	Description
ID (identifiant)	Identifiant unique du produit de ce vendeur
Quantity	Quantité disponible du produit
Initial price	Prix unitaire de l'article
Additional fee	Frais additionnels
Additional information	Informations additionnelles
Is available	Disponibilité du produit (Booléen)
Is limited	Quantité limitée du produit (Booléen)

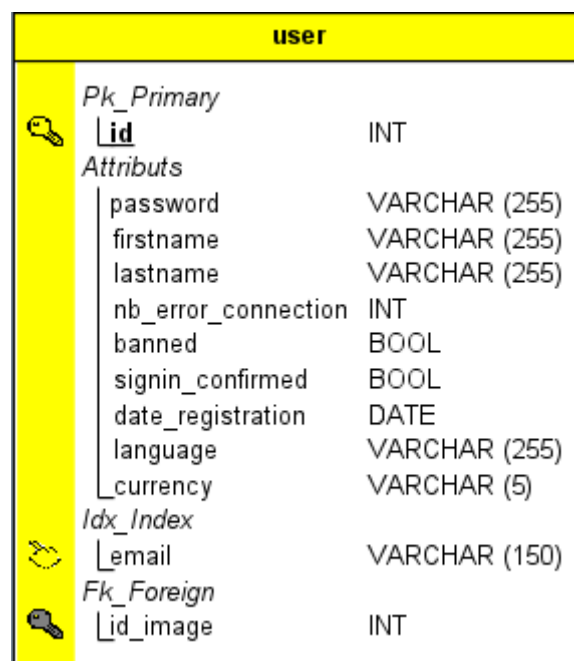
supplier_product		
	<i>Pk_Primary</i> <u>id</u>	INT
	<i>Attributs</i>	
	initial price	FLOAT
	quantity	INT
	additional_fee	FLOAT
	additional_information	TEXT
	is_available	BOOL
	is_limited	BOOL
	<i>Fk_Foreign</i>	
	id_supplier	INT
	id_product	INT (11)

## 6.2.25 User

La table **User** aura les relations suivantes :

- ➔ Les tables **Admin**, **Client** et **Supplier** hériteront des attributs de la classe **User**.
- ➔ Une relation One-To-Many avec la table **Address** car chaque utilisateur optera pour une ou plusieurs adresses de livraison.
- ➔ Une relation One-To-One avec la table **Bank account** puisque les données bancaires non-sécurisés de chaque utilisateur seront sauvegardées afin de pouvoir effectuer des paiements ou remboursements.
- ➔ Une relation One-To-Many avec la table **Forum** car chaque utilisateur pourra ouvrir autant de sessions de discussion qu'il souhaite avec les administrateurs de la plateforme.

Attributs	Description
ID (identifiant)	Identifiant unique de l'utilisateur
Email	Adresse e-mail de l'utilisateur (unique)
Password	Mot de passe
Nb Error Connection	Nombre d'erreurs de connexion
Banned	L'utilisateur est banni
Signin Confirmed	Inscription confirmée
Date Registration	Date d'inscription
Language	Langue utilisée
Currency	Devise préférentielle (ISO code pour les devises)

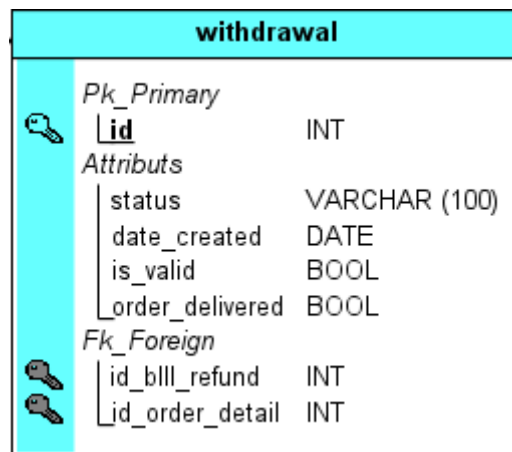


## 6.2.26 Withdrawal

La table **Withdrawal** aura les relations suivantes :

- ➔ Une relation One-To-One avec la table **Order detail** puisque chaque annulation sera obligatoirement associée à un détail de commande d'un client.
- ➔ Une relation One-To-One avec la table **Bill refund** car chaque annulation de commande valide entraînera la création d'une facture de remboursement

Attributs	Description
ID (identifiant)	Identifiant unique de l'article
Status	Statut de l'annulation de la commande (pending, solved)
Date created	Date de création
Is valid	Est valide (si date annulation - date commande < 14 jours )
Order delivered	Livraison entamée (colis )



## 7. Réalisation

### 7.1 Choix de développement

Dans le cadre de ce projet, il nous était demandé de créer une API[CITATION "ARJ00"]<sup>2</sup> permettant d'accéder à une partie ou l'ensemble des données de la plateforme.

De plus, afin de réaliser une single page application, j'ai choisi de me former à l'utilisation de frameworks moderne basé sur les langages (JavaScript et PHP) et framework (Symfony) appris au cours de notre formation.

Afin de me familiariser avec un framework conseillé par les développeurs du framework PHP [Symfony](#), mon choix s'est porté vers le framework [API Platform](#) qui permet le développement d'un projet accessible entièrement via une API afin de découpler l'application selon le type d'accès et utilisation que l'on en fait.

### 7.2 Environnement de développement et logicielle

### 7.3 Présentation des problèmes rencontrés et des solutions envisagées



## 8. Conclusion

## 9. Bibliographie

### 1. Définition « règles métier » :

- ➔ IBM Knowledge Center - Business rules :  
<https://www.ibm.com/support/knowledgecenter/fr/SSV2LR/com.ibm.wbpm.wid.bpel.doc/busrules/topics/cundbus.html>
- ➔ Wikipedia - Règles métier :  
[https://fr.wikipedia.org/wiki/Règles\\_métier](https://fr.wikipedia.org/wiki/Règles_métier)

### 2. Définition « API » :

- ➔ Wikipedia - Interface de programmation :  
[https://fr.m.wikipedia.org/wiki/Interface\\_de\\_programmation](https://fr.m.wikipedia.org/wiki/Interface_de_programmation)

### 3. Définir et comprendre OpenAPI Specification de Open API Initiative:

- ➔ GitHub – OAI / OpenAPI Specification :  
<https://github.com/OAI/OpenAPI-Specification>
- ➔ Developpez.com - Comprendre la spécification OpenAPI :  
<https://www.developpez.com/actu/178434/Comprendre-la-specification-OpenAPI-Swagger-et-apprendre-a-utiliser-Swagger-Editor-par-Hinault-Romarc/>

### 4 Définition des requêtes HTTP et de leurs méthodes de requêtes GET,DELETE, POST, PUT :

- ➔ MDN – Mozilla Developer Networks :  
<https://developer.mozilla.org/fr/docs/Web/HTTP/Méthode>

### 5. Définition de l'algorithme d'encodage de mot de passe Argon2i :

- ➔ Symfony – New Features  
<https://symfony.com/blog/new-in-symfony-3-4-argon2i-password-hasher>





## **10. Annexes**



