



# Projet personnel encadré 1 :



# *DragN'Receive*



**Dev'iliser**

CLEMENT Pierre de Dev'iliser

04/10/2021

Pour PROMEO - BTS Services informatiques aux organisations - Session 2022

---



<b>1 - Définition de la situation professionnelle</b>	<b>3</b>
1.1 - Présentation de l'entreprise cliente	3
1.1.1 - Généralités	3
1.1.2 - Environnement technique	4
1.2 - Présentation du prestataire de service	4
<b>2 - Conception graphique</b>	<b>4</b>
2.1 - Brief créatif	4
2.2 - Aperçu de la charte graphique	4
2.2.1 - Logos	4
2.2.2 - Palette de couleurs	5
2.2.3 - Typographies	6
<b>3 - Cahier des charges fonctionnel</b>	<b>6</b>
3.1 - Présentation générale	6
3.2 - Analyse du besoin	6
3.3 - Solutions envisagées	7
3.4 - Solution retenue	7
3.4.1 - Présentation des choix	7
3.4.2 - Benchmark	8
3.4.2.1 - Langage de programmation	8
3.4.2.2 - Framework	9
3.4.3 - Explication des choix	10
<b>4 - Analyses et estimations</b>	<b>11</b>
4.1 - Planification	11
4.2 - Budgétisation	11
<b>5 - Conception fonctionnelle</b>	<b>12</b>
5.1 - Dictionnaire de données	12
5.1.1 - Utilisateur	12
5.1.2 - Rôle	13
5.1.3 - Boutique	13
5.1.4 - Produit	14
5.1.5 - Commande	15
5.1.6 - Statut Commande	16
5.1.7 - Ticket	16
5.1.8 - Statut Ticket	17



5.1.9 - Message	17
5.1.10 - Web Token	18
5.2 - Exigences	19
5.3 - Cas d'utilisations	20
5.4 - Maquettes	21
5.4.1 - Page de connexion	21
5.4.2 - Profil	21
5.4.2 - Boutique	22
5.4.3 - Commande	23
5.4.4 - Produit	24
5.4.5 - Ticket	25
5.4.6 - Utilisateur	26
5.4.6 - Récupération de mot de passe	27
5.4.6 - Changement de mot de passe	27
5.5 - Droits d'Administration des fonctionnalités	28
<b>6 - Cahier des charges technique</b>	<b>30</b>
6.1 - Conception technique	30
6.1.1 - Séquence	30
6.1.2 - Spécificités des droits d'Administration des fonctionnalités	31
6.2 - Environnement technologique	34
6.2.1 - Hardware	34
6.2.2 - Software	34
6.3 - Modélisation de données	35
<b>7 - Réalisation technique</b>	<b>36</b>
7.1 - Les points forts du projet	36
7.2 - Les problèmes rencontrés	36
<b>8 - Tests</b>	<b>37</b>
8.1 - Tests unitaires	37
8.2 - Tests de validation	38
8.2.1 - Connexion	38
8.2.2 - Réinitialisation de mot de passe	40
<b>9 - Indicateurs de performance</b>	<b>43</b>
9.1 - Analyse de la planification mise à jour	43
9.2 - Analyse du coût global	44
<b>10 - Conclusion</b>	<b>45</b>



# 1 - Définition de la situation professionnelle

Le contexte défini ci-après est une situation fictive mais réaliste d'un projet d'entreprise visant à mettre en place une solution logicielle.

## 1.1 - Présentation de l'entreprise cliente

L'entreprise cliente, nommée DragN'Receive, est une société fictive de service Click & Collect implanté à Compiègne.

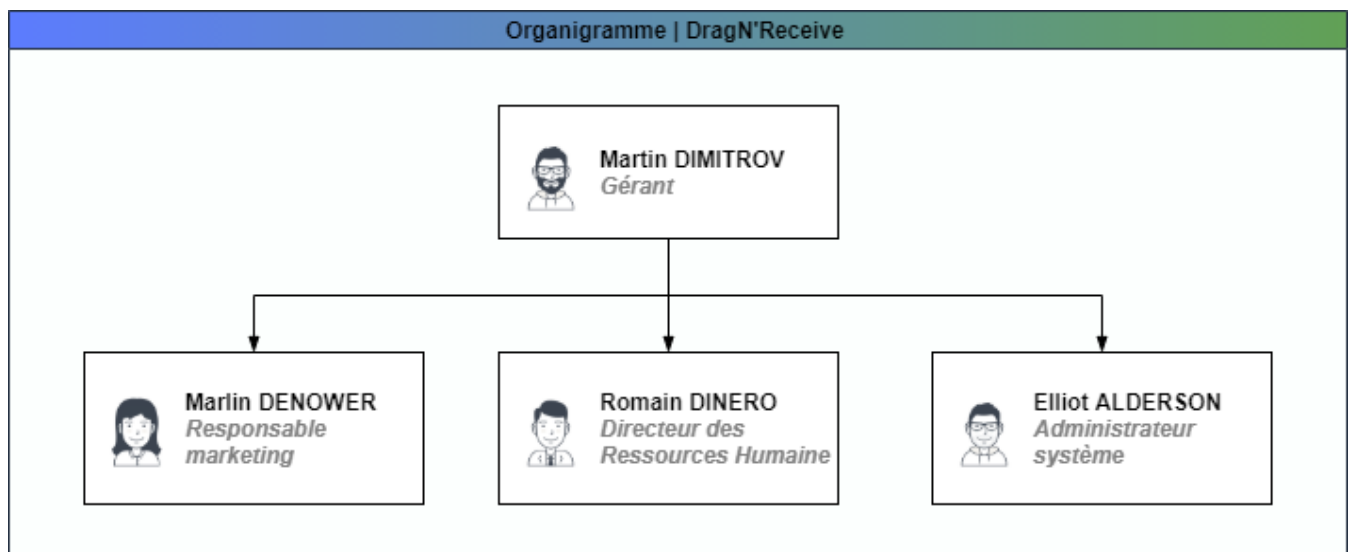
### 1.1.1 - Généralités

L'entreprise DragN'Receive est une SARL composée de 15 à 30 salariés, cette entreprise possède plusieurs contacts et partenaires dans les domaines de la restauration, de la distribution ainsi que des petits commerces.

La création de l'entreprise était d'abord à des fins opportunistes lors de la période de confinement puis s'est vu devenir un avantage primordial pour les personnes avec peu de disponibilités, les personnes à handicap et à mobilités réduites. La société se tourne également vers le développement durable avec des partenaires primant des produits possédant des impacts environnementaux faibles ainsi qu'un service de logistique verte, afin de réduire son impact sur l'environnement.

Leur capital social durant la période de 2019-2020 s'élève à 13 000 euros, pour un chiffre d'affaires de 210 000 euros.

Voici l'organigramme de l'entreprise DragN'Receive :





### 1.1.2 - Environnement technique

L'environnement technique de DragN'Receive est formé de plusieurs composants :

- Systèmes d'exploitation des appareils : Windows 10
- Serveurs : Dédiés
- Base de données : MySQL hébergé sur un serveur dédié
- Nombre d'utilisateurs simultanés potentiels : 100

## 1.2 - Présentation du prestataire de service

Le prestataire pour ce projet sera l'entreprise Dev'iliser, étant une société spécialisée dans le développement de solution logicielle et applicative depuis plus de 15 années. Dans ce cadre d'activité, elle s'occupera du développement, suivis de la mise en service ainsi que du service technique de l'application web demandée par DragN'Receive. La principale personne chargée de la réalisation de ce projet ainsi que la communication avec la société DragN'Receive sera Clément Pierre.

## 2 - Conception graphique

### 2.1 - Brief créatif

Le graphisme du site se doit de retranscrire l'identité de DragN'Receive tout modernisant leur image et design. Pour ce faire nous conserverons les touches de couleurs représentative des logos de la société que nous poserons sur un style graphique moderne et épuré (flat design).

### 2.2 - Aperçu de la charte graphique

#### 2.2.1 - Logos

Comme exprimé ci-dessus nous avons décidé de reprendre comme logo principal le logo initial de DragN'Receive :





Mais nous avons également, expressément pris la peine de réaliser par nos graphistes divers logos dans le thème de votre entreprise pour éviter la monotonie et les faire apparaître dans divers pages de votre futur site web, voici notre sélection :



NB: Les logos que nous vous fournissons seront disponibles : avec les couleurs de la marque, en bleu et blanc mais aussi en blanc intégral, permettant de les utiliser sur tout type de tonalités.

### 2.2.2 - Palette de couleurs

Voici une palette à l'image de vos logos, nous utiliserons bien entendu d'autres tonalités de couleurs mais, celle-ci seront les principales :



#91a16a

#afb854

#afb891

#014d5a

#24646f

#4a7f88

### 2.2.3 - Typographies

Pour ce qui est de la typographie, nous avons choisis Roboto, nos critères de recherche étant une police d'écriture sobre et simple à lire.

## 3 - Cahier des charges fonctionnel

### 3.1 - Présentation générale

La société DragN'Receive souhaite créer un système de Click & Collect à l'échelle de toute l'agglomération de Compiègne, en ayant pour partenaires des marchés et commerces centrés principalement sur la vente de denrées alimentaires et de produits de grande consommation.

Ils envisagent donc la réalisation d'un site web permettant la gestion des boutiques partenaires, permettant ainsi à ces dernières de proposer leurs produits et d'ajouter de nouveaux produits et leurs stocks tout en gérant les commandes clients.

### 3.2 - Analyse du besoin

Pour la réalisation du logiciel web, il sera nécessaire d'utiliser un serveur, permettant l'hébergement du client mais aussi une éventuelle base de données et la future application mobile de vente.

L'utilisation d'un langage de programmation web sera également nécessaire, il faudra également mettre en place une interface graphique, se basant sur la charte graphique définie avec le client.



Le but serait donc de réaliser un site web focalisé sur la gestion des partenaires et de leurs produits et ainsi leurs permettre d'aménager par la suite une application mobile permettant la mise en vente des produits de leur partenaires.

### 3.3 - Solutions envisagées

Les solutions envisagées permettant la réponses des divers besoin du projet sont :

- Utiliser un serveur dédié de l'entreprise ou bien utiliser un serveur OVH loué par cette dernière.
- L'utilisation d'un de ces langages web suivant : JavaScript, PHP, Python, Ruby étant les langages les plus utilisés pour la réalisation de site web.
- La réalisation d'une Interface de programme (API) communiquant avec (et entre) le site web et la base de données. Les langages les plus répandus pour sa réalisation sont le Python, le JavaScript ainsi que le Java.
- Utilisation d'un outil de design web pour une interface utilisateur fidèle et aux couleurs de l'entreprise tel que : BootStrap, Materialize, Ionicons, Skeleton, Quasar, Bulma, Vuetify.

### 3.4 - Solution retenue

#### 3.4.1 - Présentation des choix

La solution optimale permettant de répondre à chacune des problématiques serait donc, selon notre expertise :

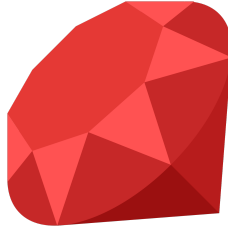
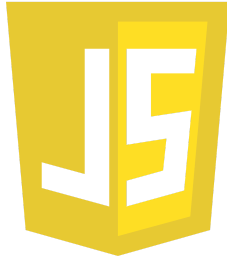
- D'utiliser le serveur dédié, permettant la mise en place du site web ainsi que la base de données hébergée par le serveur dédié, tout en conservant la liberté totale du serveur. Nous nous servons de leur offre mutualisée OVH pour utiliser leur nom de domaine une fois la mise en production réalisée.
- Nous avons retenu comme langage de programmation le JavaScript avec l'utilisation du framework Vue.
- Pour ce qui est de l'API, nous avons sélectionné JavaScript avec l'utilisation de l'environnement d'exécution NodeJS accompagné du framework ExpressJS.
- Pour ce qui est de l'interface graphique nous avons choisi d'intégrer le framework Vuetify au projet, étant très simple d'utilisation, dédié à Vue, et permettant l'utilisation de composants travaillés et complets, ce dernier permettra une belle interface utilisateur ainsi qu'une rapidité de développement non négligeable pour la réalisation de ce projet.





### 3.4.2 - Benchmark

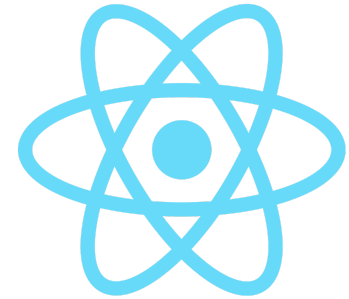
#### 3.4.2.1 - Langage de programmation



	JavaScript	Ruby	Php	Python
Utilisation	Généraliste	Généraliste	Web (Omniprésent)	Généraliste
Typage	Fort	Fort	Fort (depuis la 7.1)	Fort
Avis	Langage très simple de compréhension, en plein essor ces dernières années dans de multiples domaines par ses performances de dynamisation et de polyvalence.	Lisible, léger, rapide et simple à écrire. Permet de gagner du temps dans la programmation par ses performances. Langage sous-estimé en pleine résurgence.	Simple à comprendre, grosse marge d'erreur dans le code et nécessite un apprentissage supplémentaire pour empêcher les failles de sécurité.	Langage possédant des code plus verbeux (plus explicite et plus long), la compréhension est très rapide. Sa polyvalence permet de réaliser de petits comme de gros projets.
Date de lancement	1996	1995	1994	1991
Frameworks	Vue, Angular, React, ExpressJS	Ruby on rails	Symfony, Laravel	Django



### 3.4.2.2 - Framework



	Vue	Angular	React
Réactivité	Très réactif, une seule page permet le changement d'affichage sans rechargement	Bien réactif, permet de changer le contenu sans rafraîchissement, possibilité de faire des formulaires réactifs	Si un composant modifié est rendu, tous ses sous-composants également même ceux non modifiés le seront (à configurer manuellement)
UI	HTML / CSS / Composants	HTML / CSS	Gestion basé en JSX (même le CSS) nécessite l'apprentissage
Routage	Router Vue officiel	Router Angular officiel	Pas de router officiel
Ecosystème	Important mais moins que React	Structure très encadrée	Énorme, gros avantage
Apprentissage	Documentation irréprochable, apprentissage facile et rapide.	Conséquent pour réaliser des applications complexes	Conséquent pour réaliser des applications complexes
Mobile	Vue Native / Ionic	Angular 2 et NativeScript	React Native, testé et stable
Performances	Pas de différences significatives	Pas de différences significatives	Pas de différences significatives
Taille	Très léger	2x plus lourd que Vue	Un petit peu plus lourd que Vue
Date de lancement	2014	2010	2013
Utilisé par	Alibaba, GitLab	Google, Wix	Facebook, Uber



### 3.4.3 - Explication des choix

Notre équipe a donc choisie d'utiliser le langage JavaScript qui, pour la réalisation du projet, est le plus approprié, cela se justifiant par ses performances mais également car il va nous permettre de réaliser aussi bien le site web que le côté permettant les interactions avec la base de données.

Veuillez noter que JavaScript permettra la potentielle réalisation d'une application mobile pour la vente des produits partenaires aux clients.

Nous avons donc choisie trois frameworks, l'un pour la réalisation du **Back-End\***, l'autre pour le **Front-End\*\*** et le dernier pour embellir l'**UI\*\*\*** du site.

Nous avons choisie en Back-End le framework ExpressJS basé sur du NodeJS.

Pour ce qui est du Front-End nous avons décidé d'utiliser Vue. Le choix de ce dernier, comme nous le montre le Benchmark réalisé ci-dessus, n'aura aucun impact sur ses performances mais aura un impact positif sur la rapidité, qualité et compréhension du code de notre équipe.

Pour ce qui est de l'UI, nous avons choisie Vuetify car ce dernier est compatible et réalisé pour Vue, possédant des environnements de configuration qui permettent donc des libertés visuelles. Vuetify possède également une documentation complète et bien rédigée mais aussi des performances rapides.

**\*Back-End** : Partie cachée de l'utilisateur, permettant la gestion de la base de données et les interactions de cette dernière permettant la communication avec le Front-End

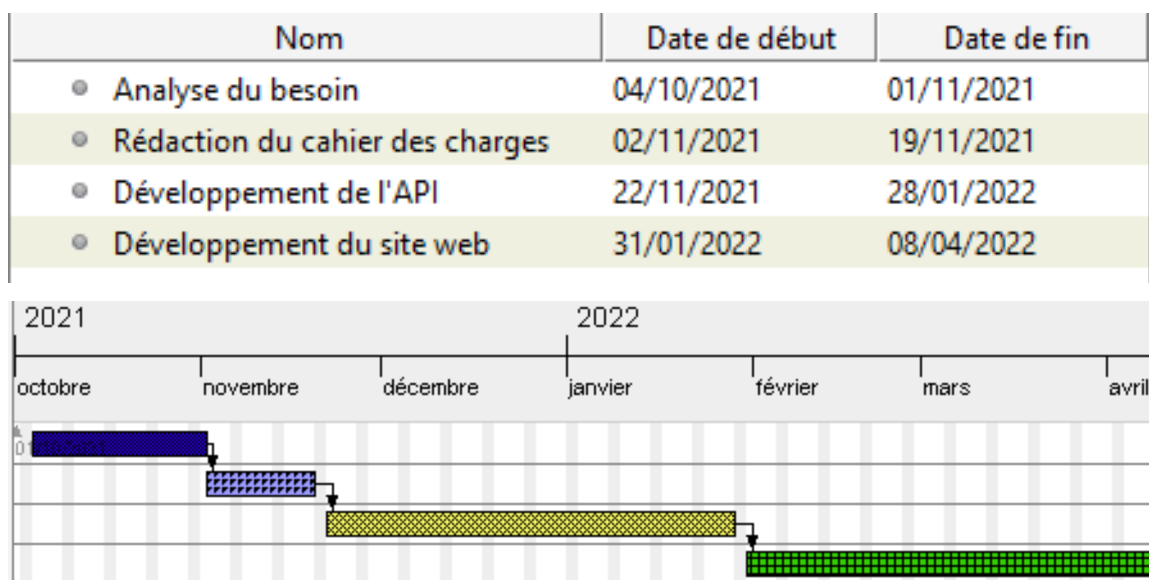
**\*\*Front-End** : Partie d'interfaces graphiques visible par l'utilisateur qui vient interagir avec le Back-End pour retourner/envoyer des éléments à l'utilisateur/la base de données.

**\*\*\*UI (User interface)** : L'interface utilisateur, environnement graphique de l'utilisateur permettant d'exprimer visuellement les besoins et interactions du site.



## 4 - Analyses et estimations

### 4.1 - Planification



### 4.2 - Budgétisation

Quantité	Description	Réduction	Coût**
1	Création d'un site web permettant la gestion de partenaire et permettant à ces derniers la mise en vente de produits et sa gestion pour le Front-End.	-	7200 €
1	Conception d'une API pour le Back-End		7200 €
1	Adaptation de la charte graphique pour le Web	-	300 €
1	Création de logos	-	235€
1	Formation à l'utilisation du site web	-	450€
1	Référencement	-	650€
1	Service technique*	100%	250€

**Total :**

**16 035 €**

\* Seuls les 2 premiers mois seront offerts, par la suite le tarif du service sera de 250 euros par mois.

\*\* Les coûts horaires sont de 18 euros.



## 5 - Conception fonctionnelle

### 5.1 - Dictionnaire de données

#### 5.1.1 - Utilisateur

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	36	PK, Obligatoire	Chaîne de caractère unique générée aléatoirement.
lastname	varchar	50	Obligatoire	nom de famille
firstname	varchar	50	Obligatoire	prénom
email	varchar	100	Obligatoire, format d'email	adresse mail
password	varchar	60	Obligatoire, comporte une majuscule, une minuscule, un nombre et un caractère spécial au minimum	mot de passe
path	varchar	255	Non-Obligatoire	chemin vers la photo
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification
roleId	varchar	1	FK: Rôle, Obligatoire	permet de séparer, admins, clients et partenaires
shopId	varchar	36	FK: Boutique, Non-Obligatoire	relie la boutique du partenaire



### 5.1.2 - Rôle

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	1	PK, Obligatoire	Chaîne de caractère entre 1 et 4
label	varchar	20	Obligatoire	nom du rôle
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification

### 5.1.3 - Boutique

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	36	PK, Obligatoire	Chaîne de caractère unique générée aléatoirement.
name	varchar	50	Obligatoire	nom de la boutique
email	varchar	100	Obligatoire	adresse mail
phone	varchar	10	Non-Obligatoire	numéro de téléphone
city	varchar	60	Non-Obligatoire	ville
street	varchar	100	Non-Obligatoire	rue
postal	int	5	Non-Obligatoire	code postal
path	varchar	255	Non-Obligatoire	chemin vers le logo
deleted	tinyint (bool)	1	Obligatoire	Booléen pour savoir si la boutique est supprimée
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification



### 5.1.4 - Produit

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	36	PK, Obligatoire	Chaîne de caractère unique générée aléatoirement.
name	varchar	50	Obligatoire	nom du rôle
description	varchar	255	Obligatoire	description du produit
price	float	9,2	Obligatoire	prix
stock	int	11	Obligatoire	stock restant
path	varchar	255	Obligatoire	chemin vers l'image du produit
deleted	tinyint (bool)	1	Obligatoire	entier 1 ou 0 pour savoir si le produit est supprimé
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification
shopId	varchar	36	FK: Boutique, Obligatoire	relie le produit à sa boutique



### 5.1.5 - Commande

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	36	PK, Obligatoire	Chaîne de caractère unique générée aléatoirement.
quantities	int	11	Obligatoire	quantité de produit
price	float	9,2	Obligatoire	prix total de la commande
number	varchar	100	Obligatoire	numéro de commande
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification
userId	varchar	36	Obligatoire	relie l'utilisateur à la commande
productId	varchar	36	Obligatoire	relie le produit à la commande
shopId	varchar	36	Obligatoire	relie la boutique à la commande
orderId	varchar	1	FK: Statut Commande, Obligatoire	relie un statut à la commande





### 5.1.6 - Statut Commande

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	1	PK, Obligatoire	Chaîne de caractère entre 1 et 5
label	varchar	20	Obligatoire	nom du statut
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification

### 5.1.7 - Ticket

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	36	PK, Obligatoire	Chaîne de caractère unique générée aléatoirement.
title	varchar	60	Obligatoire	titre du ticket
content	varchar	255	Obligatoire	contenu
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification
userId	varchar	36	Obligatoire	relie l'utilisateur au ticket
ticketStatusId	varchar	1	FK: Statut Ticket, Obligatoire	relie un statut au ticket



### 5.1.8 - Statut Ticket

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	1	PK, Obligatoire	Chaîne de caractère entre 1 et 2
label	varchar	20	Obligatoire	nom du statut
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification

### 5.1.9 - Message

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	36	PK, Obligatoire	Chaîne de caractère unique générée aléatoirement.
content	varchar	255	Obligatoire	contenu
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification
userId	varchar	36	Obligatoire	relie l'utilisateur au message
ticketId	varchar	36	FK: Statut Ticket, Obligatoire	relie un message au ticket



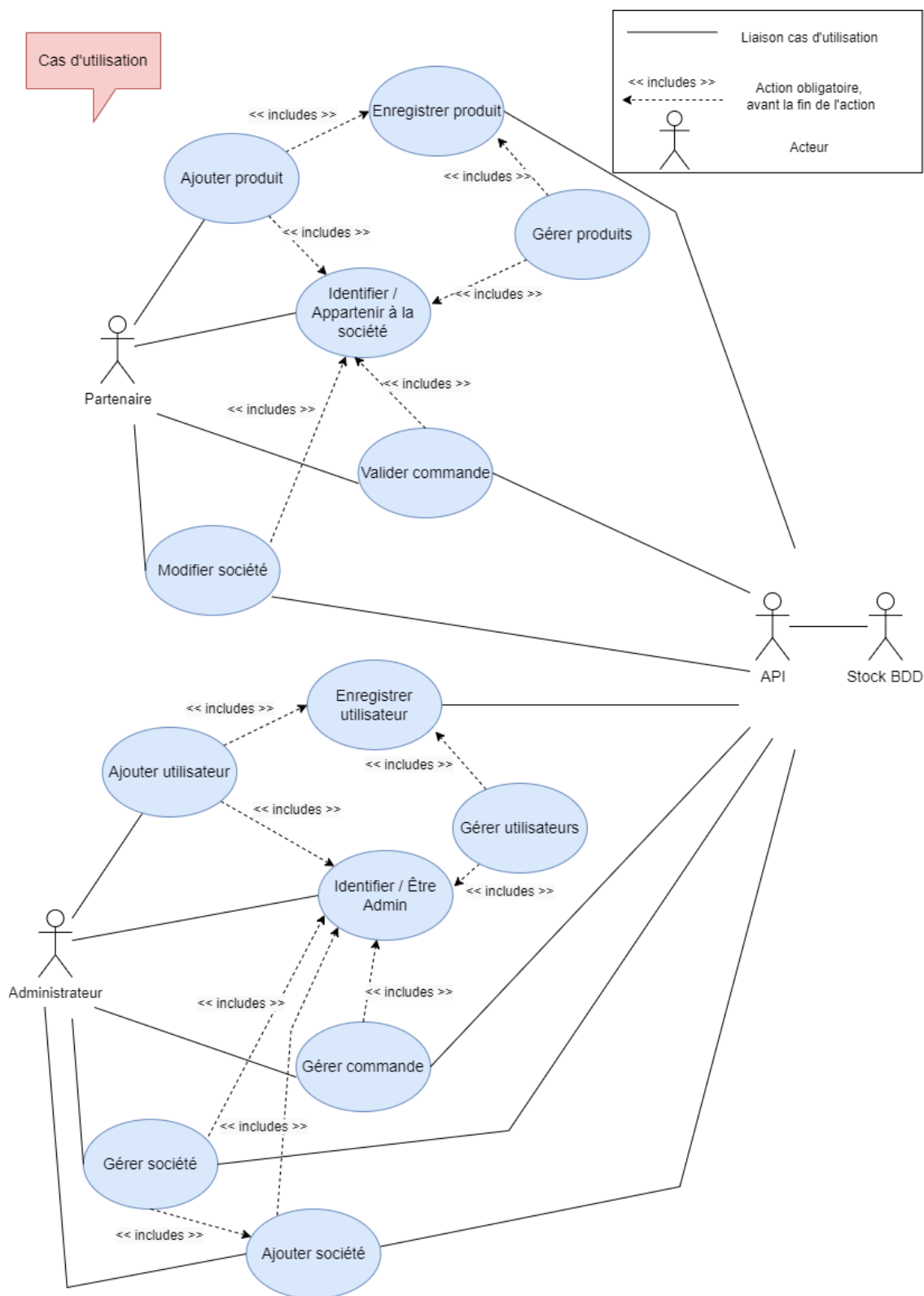
### 5.1.10 - Web Token

Champs	Types	Tailles	Contraintes	Descriptions
id	varchar	36	PK, Obligatoire	Chaîne de caractère unique générée aléatoirement.
token	varchar	255	Obligatoire	token généré aléatoirement
createdAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de création
updatedAt	datetime	yyyy-mm-ddThh:mm:ss.000Z	Obligatoire	date de la dernière modification
userId	varchar	36	Obligatoire	relie l'utilisateur au ticket





### 5.3 - Cas d'utilisations








## 5.4 - Maquettes

⚠ Les maquettes suivantes sont des aperçus de la basse fidélité du site web. Elles sont donc volontairement dénuées de toute forme de graphisme. Ces aperçus servent donc à lister et placer les éléments fonctionnels sur les différentes pages du site plus qu'à donner une idée de son esthétique, qui seront voués à changer en fonction des besoins exprimés.

### 5.4.1 - Page de connexion


Logo



Identifiant


Mot de passe

Connexion

### 5.4.2 - Profil

Logo
Gérer boutiques
Gérer tickets
Gérer utilisateurs


Nom  
Prénom  
Email  
Modifier

(boutique)  
Voir

Suppression du compte  
Supprimer



5.4.2 - Boutique

LogoGérer boutiquesGérer ticketsGérer utilisateurs

Nom

email

adresse

téléphone

Voir

Modifier

Supprimer

Nom

email

adresse

téléphone

Voir

Modifier

Supprimer

Nom

email

adresse

téléphone

Supprimé

Nom

email

adresse

téléphone

Voir

Modifier

Supprimer

LogoGérer boutiquesGérer ticketsGérer utilisateurs

Nom

Email

Téléphone

Adresse

Description

Produits...

22



5.4.3 - Commande

Logo

Gérer boutiques

Gérer tickets

Gérer utilisateurs

<div>  ...</div>	Préparer	Statut   Date   Numéro de commande	Voir
<div></div>	Terminé	Statut   Date   Numéro de commande	Voir
<div> </div>	Récupéré	Statut   Date   Numéro de commande	Voir

Logo

Gérer boutiques

Gérer tickets

Gérer utilisateurs

Numéro de commande

Statut de commande | Nom de la boutique

x Quantité  
Nom

x Quantité  
Nom

x Quantité  
Nom

x Quantité  
Nom

Récapitulatif

Adresse de facturation

Facture

Montant total

Contacteur





5.4.4 - Produit

Logo

Gérer boutiques

Gérer tickets

Gérer utilisateurs

Voir

Voir

Voir

Voir

Voir

Logo

Gérer boutiques

Gérer tickets

Gérer utilisateurs

Prix  
Stock

Description



5.4.5 - Ticket

LogoGérer boutiquesGérer ticketsGérer utilisateurs

Titre

nom de l'utilisateur

Voir

Clore

Titre

nom de l'utilisateur

Voir

Clos

Titre

nom de l'utilisateur

Voir

Clore

Titre

nom de l'utilisateur

Voir

Clore

LogoGérer boutiquesGérer ticketsGérer utilisateurs

Titre

Contenu

Auteur

Contenu

Date

Auteur

Contenu

Date

Auteur

Contenu

Date

Auteur

Contenu

Date

Rédigez une réponse

Contenu

Envoyer



5.4.6 - Utilisateur

Logo

Gérer boutiques

Gérer tickets


Gérer utilisateurs

Nom	email	Rôle	(Boutique)	<div>Modifier</div> <div>Supprimer</div>
Nom	email	Rôle	(Boutique)	<div>Modifier</div> <div>Supprimer</div>
Nom	email	Rôle	(Boutique)	<div>Modifier</div> <div>Supprimer</div>
Nom	email	Rôle	(Boutique)	<div>Modifier</div> <div>Supprimer</div>



#### 5.4.6 - Récupération de mot de passe


Logo 




Demander

#### 5.4.6 - Changement de mot de passe

Logo 





Changer



## 5.5 - Droits d'Administration des fonctionnalités

Fonctionnalités / Rôles	Super Admin	Admin	Partenaire	Client
Connexion au site web	Oui	Oui	Oui	Non
Génération token changement de mot de passe Web	Oui	Oui	Oui	Non
Réinitialisation de mot de passe Web	Oui	Oui	Oui	Non
Créer un utilisateur	Oui	Partenaire uniquement	Non	Non
Voir un utilisateur	Oui	Soi-même, Partenaire et Client uniquement	Soi-même et ses partenaires de boutique	Soi-même
Modifier un utilisateur	Oui	Soi-même, Partenaire et Client uniquement	Soi-même	Soi-même
Supprimer un utilisateur	Oui	Non	Non	Non
Créer une boutique	Oui	Oui	Non	Non
Voir une boutique	Oui	Oui	Les non-supprimé es	Les non-supprimé es
Modifier une boutique	Non	Non	La sienne si non-supprimé es	Non
Supprimer une boutique	Oui	Non	Non	Non
Créer un produit	Non	Non	Oui dans sa boutique	Non



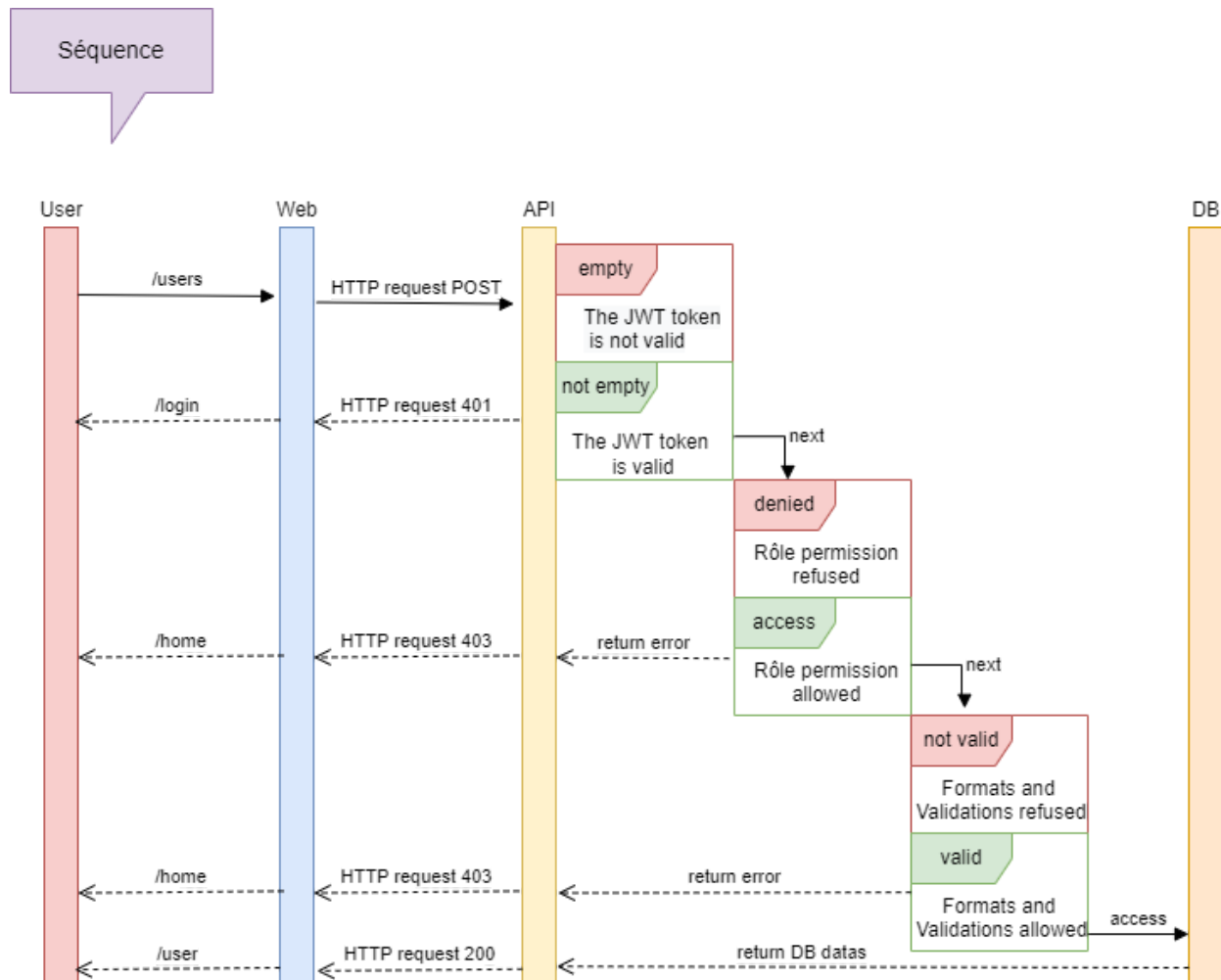
Voir un produit	Oui	Oui	Si non-supprimé et boutique non-supprimé e	Si non-supprimé et boutique non-supprimé e
Modifier un produit	Non	Non	Oui ceux de sa boutique et non-supprimés	Non
Supprimer un produit	Non	Non	Oui ceux de sa boutique	Non
Créer un ticket	Non	Non	Oui	Oui
Voir un ticket	Oui	Oui	Les siens	Les siens
Clore un ticket	Oui	Oui	Les siens	Les siens
Répondre à un ticket	Oui	Oui	Les siens	Les siens
Voir une commande	Oui	Oui	Celles de sa boutique	Les siennes
Modifier le statut d'une commande	Non	Non	Avancer les statuts d'expédition	Annuler une commande



## 6 - Cahier des charges technique

### 6.1 - Conception technique

#### 6.1.1 - Séquence





### 6.1.2 - Spécificités des droits d'Administration des fonctionnalités

**Connexion au site web :** Vérifie que l'utilisateur existe via son email, Vérifie les **permissions de l'utilisateur\***. Comparer le mot de passe avec celui en base de données. Si tout est valide, retourne le Token JWT de l'utilisateur ainsi que ses informations personnelles.

**Génération de token changement de mot de passe Web :** Vérifie la validité de l'email, vérifie que l'utilisateur existe et les permissions de l'utilisateur. Vérifie que l'utilisateur n'ai aucun Token valide durant 1h pour en générer un nouveau. Si valide, envoie par email du lien de réinitialisation.

**Réinitialisation de mot de passe Web :** Vérifie la validité du mot de passe, celle du Token (existe et non expiré), que l'utilisateur existe et les permissions de l'utilisateur. Si valide, supprime le Token et change le mot de passe de l'utilisateur et envoie par email pour confirmer le changement à l'utilisateur.

**Créer un utilisateur :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que les formats du nom, prénom et email soient valides, vérification que l'email n'existe pas, génération d'un mot de passe aléatoire. Si valide, envoie par email à l'utilisateur son mot de passe temporaire et création de son dossier sur le dossier **"Store"**.

**Voir un utilisateur :** Vérification du Token, vérifie les permissions de l'utilisateur, retourne le ou les utilisateur(s) ainsi que sa boutique si Partenaire.

**Modifier un utilisateur :** Vérification du Token, vérifie les permissions de l'utilisateur.

- Sur soi-même, un utilisateur peut modifier son mot de passe et sa photo profil, vérifier le nouveau mot de passe et comparer le mot de passe actuel avec celui en base de donnée. Pour la photo de profil, vérifie la validité de son format et stock l'image dans le dossier de l'utilisateur dans le **"Store"**. Si valide, modifie les informations de l'utilisateur.

- Si admin, peut modifier le nom, le prénom et l'email. vérifie que les formats soient valides puis vérifie que l'email n'existe pas. Si valide, modifie les informations de l'utilisateur.

**Supprimer un utilisateur :** Vérification du Token, vérifie les permissions de l'utilisateur. Si valide, supprime l'utilisateur en base de données, ainsi que son dossier dans le **"Store"** puis envoie un email à ce dernier pour l'informer.

**Créer une boutique :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie ensuite que l'email et le nom de boutique n'existe pas. Vérifie ensuite que les formats d'email et de nom soient valides. Si valide, crée la boutique ainsi que le dossier de la boutique dans le **"Store"**, puis envoie par la suite un email à l'adresse de la boutique.

**Voir une boutique :** Vérification du Token, vérifie les permissions de l'utilisateur, retourne la ou les boutique(s) avec les produits ainsi que ses partenaires.





**Modifier une boutique :** Vérification du Token, vérifie les permissions de l'utilisateur, possibilité de modifier, l'email, le nom, le téléphone, la ville, la rue, le code postal ainsi que le logo de la boutique. Les différentes valeurs se feront vérifier leurs formats et leurs validités et disponibilités pour le nom, l'email, et le téléphone. Si valide, les modifications de la boutique seront réalisées, si le logo a été changé, l'image sera stockée dans le dossier de la boutique dans le "Store".

**Supprimer une boutique :** Vérification du Token, vérifie les permissions de l'utilisateur. Si valide, passage du statut de la boutique en "Supprimée". Changement également du statut de tous les produits de la boutique en "Supprimée". Mais supprime également définitivement les partenaires affiliés à cette dernière, supprimant leurs dossier dans le "Store". Envoie également par email la confirmation aux partenaires et à la boutique.

**Créer un produit :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que la boutique existe, vérifie que les formats du nom, de la description, du prix, du stock ainsi que de l'image du produit soient valides. Si valide, crée un sous-dossier du produit dans le dossier de la boutique dans le "Store".

**Voir un produit :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que le produit existe bien ainsi que la boutique du produit, si valide, retourne le ou les produit(s) ainsi que sa boutique.

**Modifier un produit :** Vérification du Token, vérifie les permissions de l'utilisateur, possibilité de modifier le nom, la description, le prix, le stock ainsi que l'image du produit. Les différentes valeurs se feront vérifier leurs formats et leurs validités. Si valide, les modifications du produit seront réalisées, si l'image a été changée, l'image sera stockée dans le sous-dossier du produit lui-même dans le dossier de la boutique dans le "Store".

**Supprimer un produit :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que le produit existe. Si valide, modifie le statut du produit en "Supprimé".

**Créer un ticket :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que les formats du titre et du contenu soient valides. Si valide, crée un ticket avec pour statut "Ouvvert".

**Voir un ticket :** Vérification du Token, vérifie les permissions de l'utilisateur, retourne ensuite le ou les ticket(s) contenant les messages qui eux mêmes contiennent un utilisateur.

**Clore un ticket :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que le ticket existe et que son utilisateur existe également. Vérifie que le ticket ne soit pas de statut "Clos". Si valide, passe le statut en "Clos" et avertit l'utilisateur possédant le ticket par email.



**Répondre à un ticket :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que le ticket existe et que l'utilisateur existe également. Vérifie le format du contenu. Si valide, crée un nouveau message au ticket et envoie par email à la personne possédant le ticket si c'est une réponse venant d'un Admin ou Super-Admin.

**Voir une commande :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que la commande existe, retournant ainsi la ou les commande(s) contenant les produits en question ainsi que l'utilisateur ayant réalisé la commande.

**Modifier le statut d'une commande :** Vérification du Token, vérifie les permissions de l'utilisateur, vérifie que la commande existe. Si valide, modifie en fonction des permissions de l'utilisateur le statut de la commande et envoie ainsi par email les modifications concernant le statut de la commande du client.

**\* Pour les permissions se référer aux droits d'Administration fonctionnels.**

**\*\* Le dossier du nom de "Store" est le lieu stockant les fichiers des différents acteurs de l'application.**



## 6.2 - Environnement technologique

### 6.2.1 - Hardware

Pour la réalisation de ce projet l'environnement technologique utilisé a été :

- Ordinateur fixe et portable
- Clavier, souris, écran
- Box Wifi

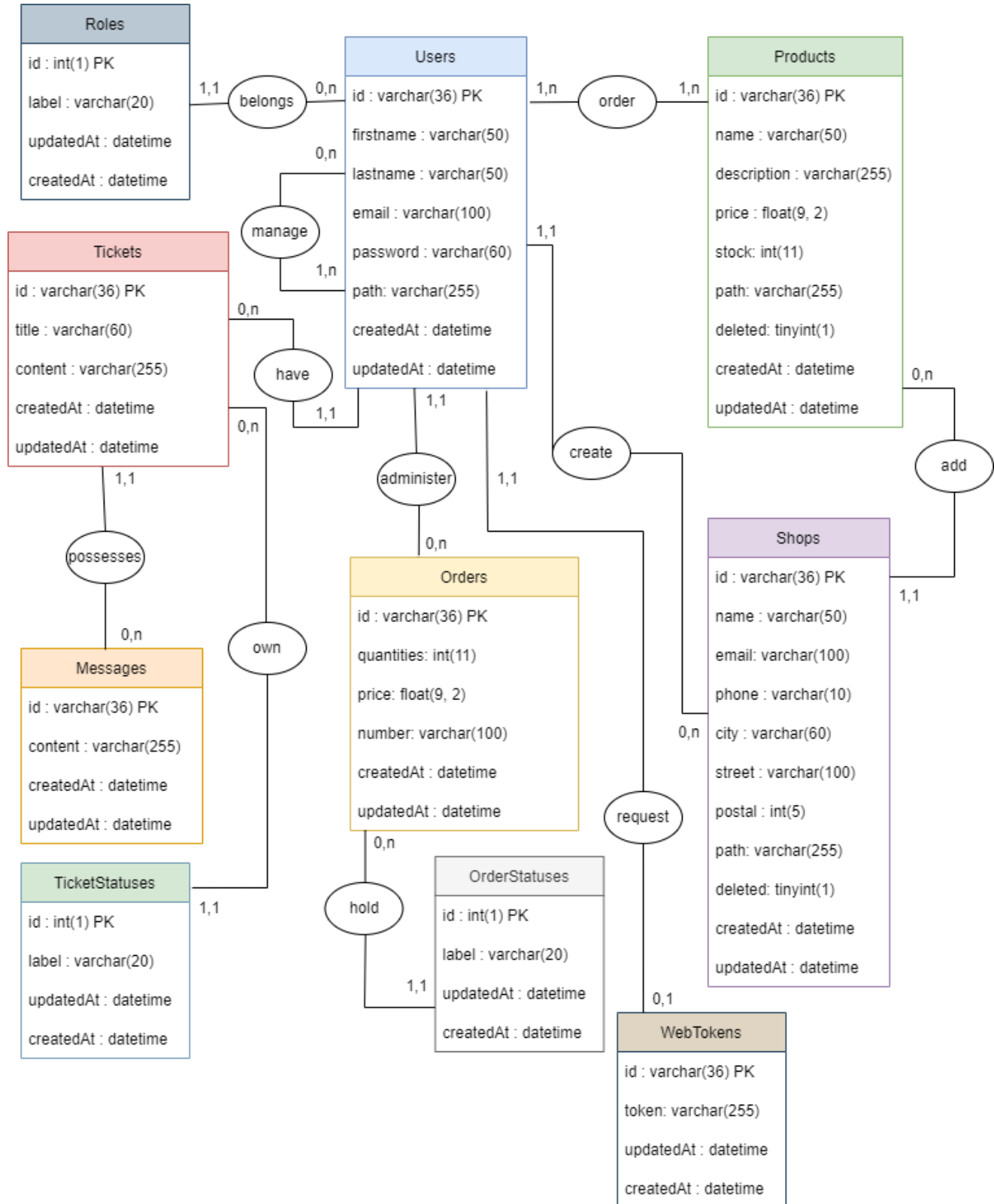
### 6.2.2 - Software

La conception de ce projet nous a fait utiliser les outils suivants :

- Windows comme système d'exploitation
- Visual Studio Code comme éditeur de texte et invite de commandes
- GitHub pour gestion de code et de version mais aussi son stockage
- Postman pour faire des requêtes HTTP sur l'API
- NPM, gestionnaire de paquets / librairies JavaScript
- NodeJS pour JavaScript
- Vue comme framework d'interface web
- Google Chrome pour tester le site web
- Draw.io pour la réalisation des diagrammes
- Jimdo pour la conception des logos
- Canva pour la conception des maquettes
- Gantt pour la planification des travaux
- Google doc pour la rédaction du cahier des charges, son stockage et sa gestion de versions



### 6.3 - Modélisation de données





## 7 - Réalisation technique

### 7.1 - Les points forts du projet

Le plus grand avantage de ce projet est qu'il m'a permis de réaliser une approche de développement et de gestion de projet autant au back-end qu'au front-end, la gestion des plusieurs rôles à également permis de diversifier le projet par l'intégration de diverses fonctionnalités.

Un des autres avantages serait également que ce projet va pouvoir servir à mon autre projet, qui est l'application mobile, l'API sera donc vouée à évoluer avec le temps, ce qui permet de rendre plus concret cette mise en application de réalisation de projet.

### 7.2 - Les problèmes rencontrés

Les problèmes rencontrés se sont vus être des soucis de compétences, étant mon premier projet long terme avec peu d'interruptions, cependant la réalisation de ce dernier m'a, au fur et à mesure, permis de monter très rapidement en compétence et de prendre en main les outils à ma disposition.

La conception de l'API m'a permis d'apprendre à me creuser la tête sur l'algorithmique et les validations et conditions ainsi que la gestion de session / administration, me permettant de comprendre tout le cheminement d'une requête.

Le client Web en Vue lui m'a permis également de mieux comprendre l'affichage, j'ai eu beaucoup moins de problèmes dessus que sur l'API, le seul problème était le temps avant la deadline ainsi que l'utilisation de Vuex pour stocker les données.



## 8 - Tests

### 8.1 - Tests unitaires

Pour le passage des tests unitaires de mon API, j'ai choisi d'utiliser Mocha et Chai, des bibliothèques JavaScript permettant d'effectuer des tests sur différentes fonctionnalités et vérification de l'API, notamment via requêtes HTTP.

#### Exists

##### Email exist

- ✓ Email exist in Database (59ms)
- ✓ Email dont exist in Database

##### Shop exist

- ✓ Shop exist in Database
- ✓ Shop dont exist in Database

#### Formats

##### Email format

- ✓ Email format is valid
- ✓ Email should be a string
- ✓ Email should be "string@string.string" format
- ✓ Email length should be less or equal to 100

##### Password format

- ✓ Password format is valid
- ✓ Password should be a string
- ✓ Password should have one upper case character
- ✓ Password should have one lower case character
- ✓ Password should have one number character
- ✓ Password should less or equal to 60 characters
- ✓ Password should be more than 7 characters

#### Requests API

##### GET /api/currentUser

- ✓ Response is an object
- ✓ Response is null because no token in header

##### POST /api/loginUser

- ✓ Client cannot loggedin
- ✓ Wrong password cannot loggedin
- ✓ Wrong email cannot loggedin
- ✓ Valid Sadmin, Admin or Partner can be logged in

21 passing (214ms)



## 8.2 - Tests de validation


Il y aura également à disposition dans l'API des validations plus approfondies des différents rôles utilisateur, réalisées avec la librairie JavaScript "puppeteer", permettant de simuler un navigateur web ainsi que les actions d'un utilisateur.

### 8.2.1 - Connexion

The screenshot shows the 'Connexion' page of the DragN'Receive application. The header includes the logo and the title 'Connexion'. The main form is titled 'Connectez-vous' and contains two input fields: 'E-mail' with the value 'admin@admin.admin' and 'Mot de passe' with masked characters. A 'SE CONNECTER' button is visible at the bottom left, and a link for 'Mot de passe oublié ? Réinitialisez votre mot de passe.' is at the bottom right.

This screenshot shows the same login page but with validation errors. The 'E-mail' field now contains 'admin@admin' and has a message below it: 'Le format de l'e-mail doit être valide'. The 'Mot de passe' field contains masked characters and has a message below it: 'Le mot de passe doit contenir : une majuscule, une minuscule, un chiffre, un caractère spécial'. The 'SE CONNECTER' button and the password reset link remain visible.





Connexion

Connectez-vous


Email introuvable.

E-mail  
 admin@admin.admi

Mot de passe  
 .....

SE CONNECTER

Mot de passe oublié ? Réinitialisez votre mot de passe.



Connexion

Connectez-vous


Mot de passe incorrect.

E-mail  
 admin@admin.admin

Mot de passe  
 .....

SE CONNECTER

Mot de passe oublié ? Réinitialisez votre mot de passe.



Connexion

Connectez-vous

Vous ne pouvez pas vous connecter avec un compte client, utilisez notre application mobile !

E-mail  
 client@client.client


Mot de passe  
 .....


SE CONNECTER

Mot de passe oublié ? Réinitialisez votre mot de passe.







Profil Gestion des utilisateurs Gestion des boutiques

 Déconnexion

Que faire en tant qu'Administrateur ?



En tant qu'Administrateur, vous avez la possibilité de pouvoir surveiller et gérer les utilisateurs, les tickets ainsi que les boutiques des autres utilisateurs.


## 8.2.2 - Réinitialisation de mot de passe

Mot de passe oublié ?

E-mail

ENVOYER

 Retourner à la connexion.



Connexion

Mot de passe oublié ?

E-mail

ENVOYER

 Retourner à la connexion.

 L'utilisateur n'existe pas !



Connexion

Mot de passe oublié ?

E-mail  
admin@admin.admin

ENVOYER

Retourner à la [connexion](#).

Vous avez déjà reçu un email contenant un token valide !

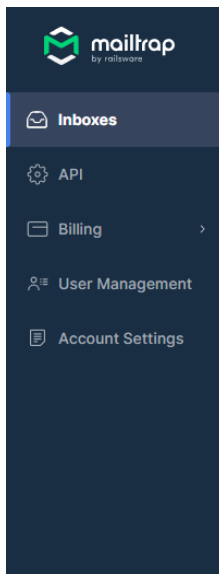


Connexion

Email envoyé !

L'email a bien été envoyé, dedans il y aura un lien permettant de réinitialiser votre mot de passe.

Le token a bien été envoyé.



Inboxes > My Inbox > DragN'Receive - Demande de réinitialisation de mot de passe !

Search...

DragN'Receive - Demande de réinitialisation de mot de passe !  
to: <sadmin@sadmin.sadmin> a few seconds ago

DragN'Receive - Réponse au ticket n°28221559-e3bd-49a0-94e6-96d26b728e07 !  
to: <Gaspar-Colin@multi-channelled-... 10 minutes ago

DragN'Receive - Ticket n°bd47ba0c-cf9e-4d6c-b0f4-de6663436017 clos !  
to: <partner@partner.partner> 15 minutes ago

DragN'Receive - Ticket n°fb5e65ac-2b42-4dc6-bf1d-bd5c629892cf clos !  
to: <partner@partner.partner> 15 minutes ago

DragN'Receive - Demande de réinitialisation de mot de passe !  
to: <sadmin@sadmin.sadmin> 24 minutes ago

DragN'Receive - Demande de réinitialisation de mot de passe !

From: Service DragN'Receive <service@dragnreceive.fr>  
To: <sadmin@sadmin.sadmin>

[Show Headers](#)

HTML

HTML Source

Text

Raw

Spam Analysis

HTML Check

Tech Info



Bonjour Sadmin Sadmin,  
Vous avez demandé une réinitialisation de mot de passe :  
Lien : <http://localhost:8080/reset/1/0xF4aCEE934b63bbfA83E0bCBB6ea7d134>  
Ce n'est pas vous ? Veuillez changer de mot de passe par sécurité.





Connexion


### Mot de passe oublié ?

 Mot de passe

 Confirmez le mot de passe


ENVOYER

Retourner à la [connexion](#).




Connexion

### Mot de passe oublié ?

 E-mail

ENVOYER

Retourner à la [connexion](#).

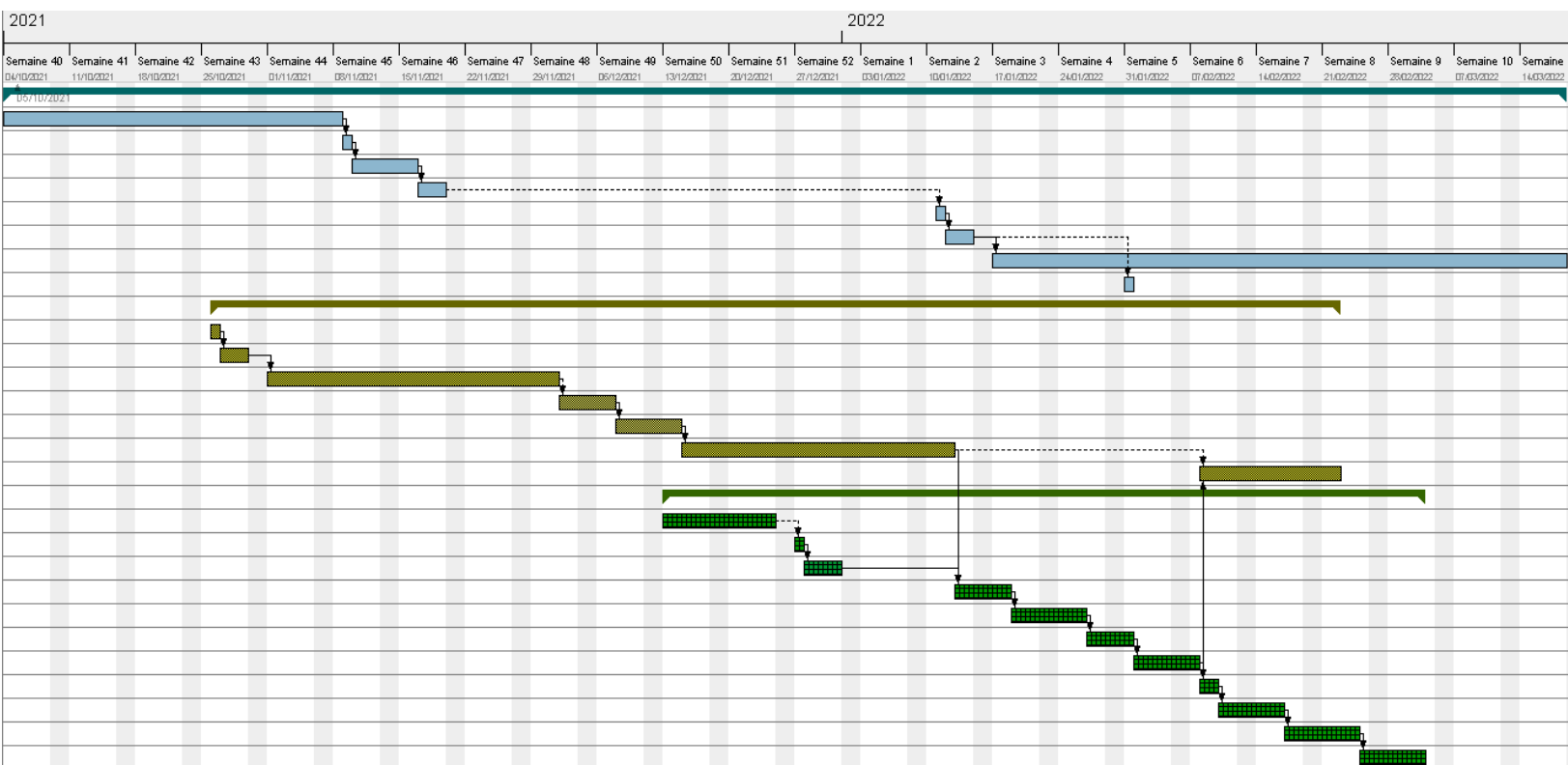
 Le lien est invalide ou a expiré !



## 9 - Indicateurs de performance

### 9.1 - Analyse de la planification mise à jour

Nom	▲ Date de début	Date de fin
☐ • AUTRES	04/10/2021	18/03/2022
• Analyse et rédaction du besoin	04/10/2021	08/11/2021
• Benchmarks / Logos	09/11/2021	09/11/2021
• Diagrammes	10/11/2021	16/11/2021
• MCD	17/11/2021	19/11/2021
• Dictionnaire de données	11/01/2022	11/01/2022
• Droits d'Administration	12/01/2022	14/01/2022
• Modifications / rédactions	17/01/2022	18/03/2022
• Maquettes	31/01/2022	31/01/2022
☐ • API	26/10/2021	22/02/2022
• Initialisation de l'API	26/10/2021	26/10/2021
• DB, routing et modèles	27/10/2021	29/10/2021
• Authentification	01/11/2021	01/12/2021
• Faker et Controllers	02/12/2021	07/12/2021
• Stockage fichier	08/12/2021	14/12/2021
• Controllers, permissions, Validat...	15/12/2021	12/01/2022
• Mailer / modifications / unit tests	08/02/2022	22/02/2022
☐ • WEB	13/12/2021	03/03/2022
• Initialisation du site web	13/12/2021	24/12/2021
• navbar / footer / auth pages	27/12/2021	27/12/2021
• Pages de présentation	28/12/2021	31/12/2021
• Requêtes HTTP et fonctions	13/01/2022	18/01/2022
• Vérifications / permissions / rou...	19/01/2022	26/01/2022
• Pages gestion utilisateurs	27/01/2022	31/01/2022
• Pages de gestion boutiques	01/02/2022	07/02/2022
• Page profil	08/02/2022	09/02/2022
• Réinitialisation de mots de passe	10/02/2022	16/02/2022
• Pages de gestions de commandes	17/02/2022	24/02/2022
• Tests unitaires web	25/02/2022	03/03/2022



## 9.2 - Analyse du coût global

Quantité	Description	Réduction	Coût
1	Création d'un site web permettant la gestion de partenaire et permettant à ces derniers la mise en vente de produits et sa gestion pour le Front-End.	-	9792 €
1	Conception d'une API pour le Back-End		7344 €
1	Adaptation de la charte graphique pour le Web	-	300 €
1	Création de logos	-	235€
1	Formation à l'utilisation du site web	-	450€
1	Référencement	-	650€
1	Service technique	100%	250€

**Total :**

**18 771 €**



## 10 - Conclusion

Le projet s'est trouvé être très enrichissant que ce soit dans la gestion du projet, la rédaction du cahier des charges mais aussi dans le développement back-end et front-end d'applications web.

D'un point de vue personnel il m'a également appris, grâce à l'autonomie réalisé sur le projet, à chercher les réponses par moi même et prendre conscience des compétences que j'ai.

Le projet m'a aussi appris à réaliser de la veille et me renseigner sur les différents langages et frameworks de programmation, me permettant de choisir un environnement de développement me permettant de m'épanouir dans le domaine du développement Web.

Au début du projet, en voyant sa taille et la quantité de fonctionnalités à développer, j'ai eu peur de ne pas avoir le temps de tout terminer, ce qui m'a permis d'apprendre à être organisé et avancer petit à petit avec la tête froide.

Pour conclure, la réalisation de ce projet m'a permis de réaliser quelque chose de concret sur une longue période ainsi qu'apprendre à me tenir des deadlines et définir des endpoints. Je suis satisfait de mon travail et ces heures passées à le réaliser ont confirmé ma détermination à poursuivre dans le développement.