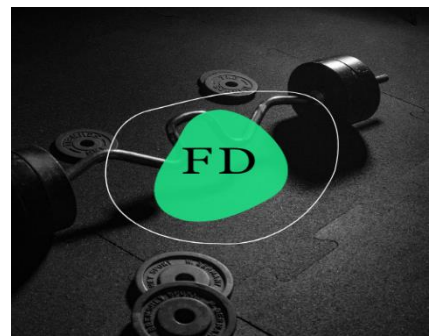


# DOSSIER PROJET

Virginie GIACOMETTI

Développeur Web et Web Mobile  
Studi

Application FitnessDrive



## Remerciements

Je tiens à remercier tous les formateurs de chez Studi. Grâce à leurs lives hebdomadaires, leurs bonnes humeurs et leurs explications, ils m'ont permis de bien réaliser ce projet. Ils ont su m'apporter également un réconfort dans les moments difficiles.

Je souhaite également remercier une ancienne apprenante, Nathalie, qui m'a aidé au déploiement de mon application.

Je remercie aussi ma famille qui a su me soutenir tout au long de ma formation.

# Sommaire

Remerciements .....	3
Sommaire .....	2
Introduction	
1. Compétences du référentiel couverte par le projet .....	4
a. Développer la partie frontend d'une application web ou web mobile en intégrant les recommandations de sécurité .....	4
b. Développer la partie backend d'une application web ou web mobile en intégrant les recommandations de sécurité .....	
2. Résumé du projet .....	5
Le projet	
1. Cahier des charges .....	6
2. Expressions des besoins .....	7
3. Spécifications fonctionnelles .....	8
4. Charte Graphique .....	12
5. Spécifications techniques .....	13
La réalisation du projet	
1. Diagrammes.....	15
2. Méthode Merise.....	18
3. Développement du projet	
a. Installation .....	20
b. Initialisation .....	20
c. Base de données .....	21
d. Routes et controllers .....	23
e. Panel d'admin.....	23
f. Service Email .....	24
g. Formulaire .....	23
4. Développement partie Front-End	
a. Mise en place .....	27
b. Interface .....	27
c. Navigation .....	28
d. JavaScript .....	29
e. Organisation .....	30
Fonctionnalité la plus représentative : L'authentification .....	31
Conclusion .....	36
Annexes .....	37

# Introduction

## 1. Compétences du référentiel couvertes par le projet

### a. Développer la partie frontend d'une application web ou web mobile en intégrant les recommandations de sécurité

- Réaliser la maquette d'une application web responsive
- Créer un site web avec HTML, CSS et le Framework Bootstrap
- Utiliser Git et Github pour gérer son code source
- S'initier à la programmation avec Javascript
- Dynamiser vos sites web avec Javascript
- Déployer son application web avec Heroku

### b. Développer la partie backend d'une application web ou web mobile en intégrant les recommandations de sécurité

- Créer et administrer une base de données
- Développer des sites web dynamiques avec PHP
- Créer une application web avec le framework Symfony
- Créer un panel d'administration

## 2. Résumé du projet

Dans le cadre de ma formation Graduate Développeur Web – Web Mobile chez Studi, j’ai eu à concevoir une application web complète.

Le projet est un site de gestion de salle de sport. L’équipe technique de la marque doit pouvoir ajouter des nouveaux franchisés et leurs structures mais aussi de gérer leurs différentes permissions. Lorsqu’une salle de sport ouvre et prend la franchise de la marque, elle a un accès à un outils de gestion en ligne qui liste les différentes permissions qu’elle peut utiliser en fonction de son contrat. Les franchisés et leurs structures auront un accès est en lecture uniquement.

L’objectif du projet est d’analyser les besoins et de développer l’application. J’ai utilisé le pattern MVC qui m’a permis de bien structurer mon code source. J’ai pu découper mon projet en trois parties, avec les technologies comme doctrine pour ORM (**M**odel), le moteur de template Twig (**V**iew) et PHP et le framework Symfony (**C**ontroller).

Cela m’a permis de mettre en pratique les compétences acquises lors de ma formation mais également d’en gagner de nouvelles en me référant aux différentes documentations, forums.

# Le projet

## 1. Cahier des charges

Pour réaliser la partie, je me suis inspiré de la méthode de conception, Le **Design Thinking**. C'est une méthode ou approche centrée sur l'utilisateur / humain et tournée vers l'innovation, quel que soit le domaine d'application, l'empathie, la définition, la créativité, l'itération et le droit à l'erreur sont au cœur de cette méthodologie d'innovation. Elle permet, via une démarche structurée, d'orienter les projets vers d'avantage d'innovation. Elle repose sur 5 étapes. Et pour chacune de ces étapes, des outils sont utilisés.

### a. Contexte

Notre client est une grande marque de salle de sport et souhaite la création d'une interface simple à destination de ses équipes qui gèrent les droits d'accès de ses franchisés et leurs structures. Ainsi, lorsqu'une salle de sport ouvre et prend la franchise de cette marque, on lui donne accès à un outil de gestion en ligne.

Le projet a donc pour but la création et la construction d'une interface cohérente et ergonomique destiné aux employés de l'entreprise. Ainsi depuis leurs espaces ils peuvent, gérer les accès aux différentes options des franchisés de la marque, proposées par la société. Depuis cette interface les franchisés ne peuvent que voir leurs options activées et les salles qu'ils possèdent. Seuls les employés de la marque peuvent modifier les droits.

### b. Cible – Personas, Définition

Il y aura trois d'utilisateurs du produit :

- Client => équipe technique qui ont un accès au panel d'administrateur
- Le franchisé => accès son compte et sa/ses structures en lecture
- La structure => accès à son compte en lecture

Pour mieux comprendre les attentes du client, j'ai créé des personnes fictives qui représentent les catégories d'utilisateur de l'application.

Administrateur :

Administrateur de Fitness Drive	
	Nom : FitCat Prénom : Julie Age: 28 ans Situation social : Salariée en CDI Adresse : Montpellier
Personnalité : Sociale, rigoureuse	Profil : Elle travaille dans la Société depuis 8 ans. Apprécie la relation direct avec les utilisateurs ainsi que sa capacité à résoudre leurs problèmes
Utilisation du produit : Elle se charge de créer ou de modifier les nouvelles franchises avec leurs structures et leurs permissions	
Objectifs/Problèmes résolus par le produits : Elle apprécie la simplicité de la saisie de données et valorise une ergonomie efficace lui permettant de faire son travail sans perte de temps	

Franchise :

### Franchise de Fitness Drive



Nom : Gaillard  
Prénom : Rémy  
Age: 47 ans  
Situation social : Gérant de la Franchise Fitness Drive Montpellier  
Adresse : Montpellier

**Personnalité :** Extraverti, sportif extrême

**Profil :** Sportif de haut niveau, il aime relever des défis extrêmes.  
Digital addict : Il possède de chaîne YouTube où il poste ses vidéos de ses exploits.

**Utilisation du produit :** Il utilisera cette application pour visualiser ses structures et permissions

**Objectifs/Problèmes résolus par le produits :** Toujours en mouvement, il n'utilisera l'application que depuis son mobile.

Structure :

### Structure de la franchise de Montpellier



Nom : Inshape  
Prénom : Thibo  
Age: 30 ans  
Situation social : Gérant de la structure - 3 av de la mer à Montpellier  
Adresse : Montpellier

**Personnalité :** Bodybuilder amateur, sportif extrême

**Profil :** Sportif de haut niveau, il a participé à des concours de musculation.  
Digital addict : Il possède de chaîne YouTube où il donne des conseils sur la musculation et autre.

**Utilisation du produit :** Il utilisera cette application pour visualiser sa structure et ses permissions

**Objectifs/Problèmes résolus par le produits :** Il apprécie le côté pratique et simple d'utilisation de l'application

## 2. Expressions des besoins, *Recherche*

Les besoins du client sont :

- Gérer les droits d'accès à une application Web
- Un panel d'administration
- Interface simple et facile à prendre en main

Pour gérer les droits d'accès, j'ai dû mettre en place un système de login pour que chaque utilisateur est accès uniquement à leur compte (pour les franchises et les structures) et développer un panel d'admin pour les employés de la marque.

Pour l'interface, je me suis inspiré du design Dark mode pour la navigation au sein de l'application avec des icones pour plus de faciliter.

Pour les clients, des images en fond, des cartes pour le contenu avec des couleurs vives.

Pour les admins, le fond est blanc, des cartes pour le contenu avec des couleurs vives.

### 3. Spécifications fonctionnelles, *Idéation*

Les fonctionnalités du projet sont :

- Général :
  - Se connecter
- Pour le panel d'admin :
  - Ajouter un Utilisateur
  - Ajouter une Franchise
  - Ajouter une Structure
  - Ajouter une permission
  - Ajouter une permission à une structure
  - Désactiver/Activer une Franchise, une structure ou une permission
  - Liste de toutes les Franchises/Structures/Utilisateurs/Permissions
  - Recherche dynamique
- Pour une Franchise :
  - Visualiser le détail de sa franchise
  - Visualiser le détail de sa/ses structures
  - Contacter l'équipe avec une page contact
- Pour une Structure :
  - Visualiser le détail de sa structure
  - Contacter l'équipe avec une page contact

Pour la rédaction d'une User story, je me suis assurée

- Qu'elle soit compréhensible de tous en utilisant un langage naturel,
- Qu'elle raconte une histoire liée à un utilisateur ou un type d'utilisateur,
- Qu'elle aille à l'essentiel,
- Qu'elle soit suffisamment claire pour permettre à l'équipe de développement de l'estimer,
- Qu'elle soit réalisable complètement en un seul sprint.

Afin de mieux me représenter les fonctionnalités, j'ai intégré les personas dans les Users Stories.

J'ai également divisé l'ensemble des Users Stories par thème :

- A. Gestion des Utilisateurs  
Description des fonctionnalités qu'un administrateur pourra effectuer.
- B. Gestion des Franchises  
Description des droits d'accès à application par une franchise
- C. Gestion des Structures  
Description des droits d'accès à application par une structure



## A. Gestion des Utilisateurs

- Scénario de test d'une fonctionnalité d'authentification d'un Administrateur :

<b>Titre de l'us :</b> US1 – Authentification d'un utilisateur / Administrateur
<p><b>En tant que,</b> Julie</p> <p><b>Je souhaite</b> me connecter au site</p> <p><b>Afin d'</b>accéder à mon compte</p>
<p><b>Règles métier</b></p> <p>1. Le login doit être un adresse e-mail unique</p> <p>2. Le mot de passe doit faire plus de 8 caractères</p>
<p><b>Critères d'acceptation</b></p> <p><b>Scénario 1 :</b> Tentative de connexion avec un compte valide</p> <p><b>Étant donné que,</b> je possède un compte utilisateur pour Julie</p> <p><b>Lorsque,</b> j'accède à la page d'authentification /login</p> <p><b>Et que</b> je saisis mon login <a href="mailto:fitnessdrive.ad01@outlook.com">fitnessdrive.ad01@outlook.com</a> dans le champ « login »</p> <p><b>Et que</b> je saisis mon mot de passe fitnessdriveadmin dans le champ « Mot de passe »</p> <p><b>Et que</b> je clique sur le bouton « Connexion »</p> <p><b>Alors</b> je suis connecté en tant que Admin01</p> <p><b>Et</b> je suis redirigé sur la page d'accueil /admin</p>

## B. Gestion des Franchises

- Scénario de test d'une fonctionnalité d'authentification d'un Franchisé :

<b>Titre de l'us :</b> US10 – Authentification d'un utilisateur / Franchise
<p><b>En tant que,</b> Rémy - Gérant de la franchise</p> <p><b>Je souhaite</b> me connecter au site</p> <p><b>Afin d'</b>accéder à mon compte</p>
<p><b>Règles métier</b></p> <p>1. Le login doit être un adresse e-mail unique</p> <p>2. Le mot de passe doit faire plus de 8 caractères</p>
<p><b>Critères d'acceptation</b></p> <p><b>Scénario 10 :</b> Tentative de connexion avec un compte valide</p>

**Étant donné que**, je possède un compte utilisateur pour Rémy  
**Lorsque**, j'accède à la page d'authentification /login  
**Et que** je saisis mon login [remygaillard@gmail.com](mailto:remygaillard@gmail.com) dans le champ « login »  
**Et que** je saisis mon mot de passe « \*\*\*\*\* » dans le champ « Mot de passe »  
**Et que** je clique sur le bouton « Connexion »  
**Alors** je suis connecté en tant que « Rémy Gaillard »  
**Et** je suis redirigé sur la page d'accueil /home

### C. Gestion des Structures

- Scénario de test d'une fonctionnalité d'authentification d'une structure :

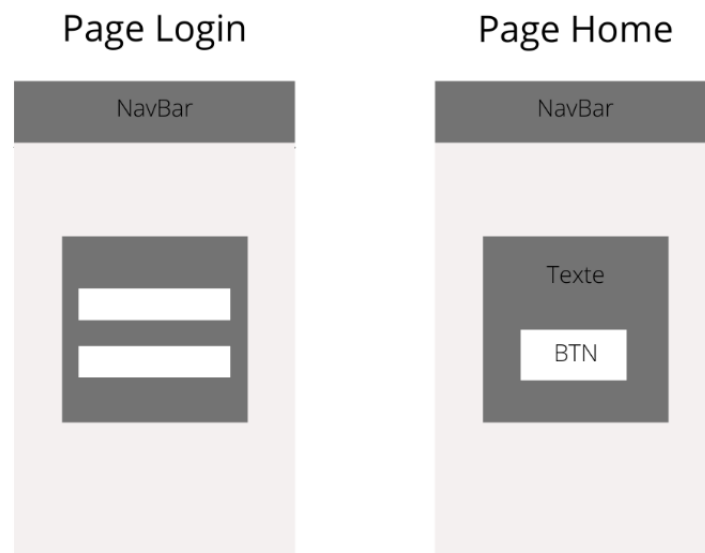
<b><u>Titre de l'us :</u></b> US14 – Authentification d'un utilisateur /Structure
<b>En tant que</b> , Thibo – Gérant de la structure <b>Je souhaite</b> me connecter au site <b>Afin d'</b> accéder à mon compte
<b><u>Règles métier</u></b> 1. Le login doit être un adresse e-mail unique 2. Le mot de passe doit faire plus de 8 caractères
<b><u>Critères d'acceptation</u></b> <b>Scénario 14 :</b> Tentative de connexion avec un compte valide <b>Étant donné que</b> , je possède un compte utilisateur pour Thibo <b>Lorsque</b> , j'accède à la page d'authentification /login <b>Et que</b> je saisis mon login <a href="mailto:tiboInshape@gmail.com">tiboInshape@gmail.com</a> dans le champ « login » <b>Et que</b> je saisis mon mot de passe « ***** » dans le champ « Mot de passe » <b>Et que</b> je clique sur le bouton « Connexion » <b>Alors</b> je suis connecté en tant que Thibo <b>Et</b> je suis redirigé sur la page d'accueil /home

(Toutes les Users Stories sont en *Annexe p :37*)

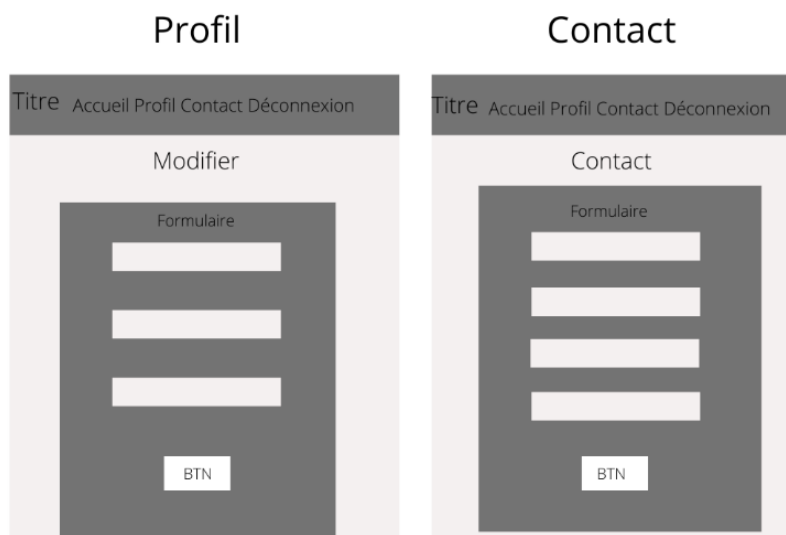
## 4. Charte Graphique, *Conception*

Une fois les besoins identifiés, les personas créés et les scénarios rédigés, j'ai commencé à élaborer le squelette de l'application sur papier. J'ai pu définir l'organisation général des blocs, matérialisé la navigation entre les pages et orienter ma réflexion en reprenant en compte les critères ergonomiques et les exigences du client.

### Wireframe Mobile

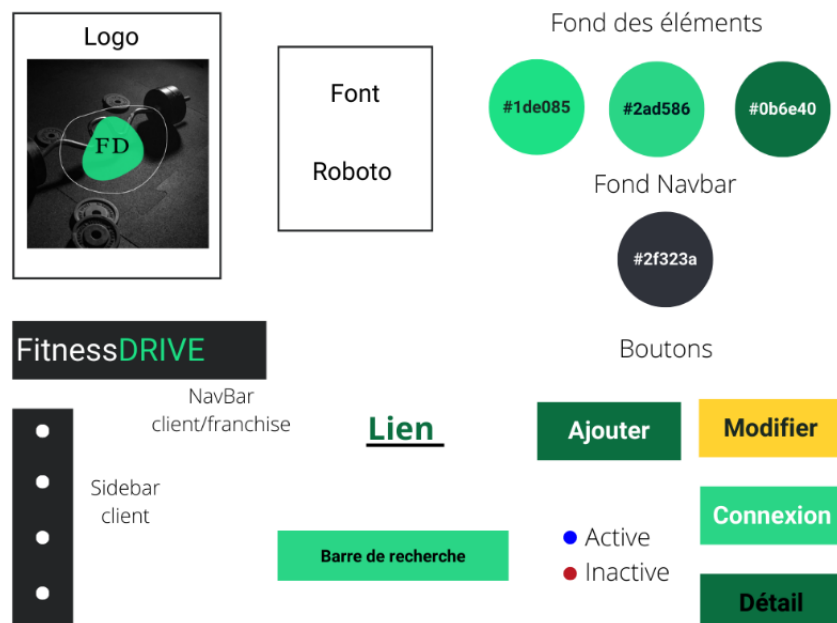


### Wireframe Desktop



(Toutes les Wireframes sont en **Annexe p :47.**)

A partir des Wireframes, j'ai élaboré un modèle d'interface d'utilisateur réaliste, abouti et de valider tous les aspects visuels du produit grâce à la planche de style. *Le Design Interface*



Enfin j'ai pu transformer les wireframes avec la planche de style en maquette graphique, les Mockups.

### Mockups Mobile

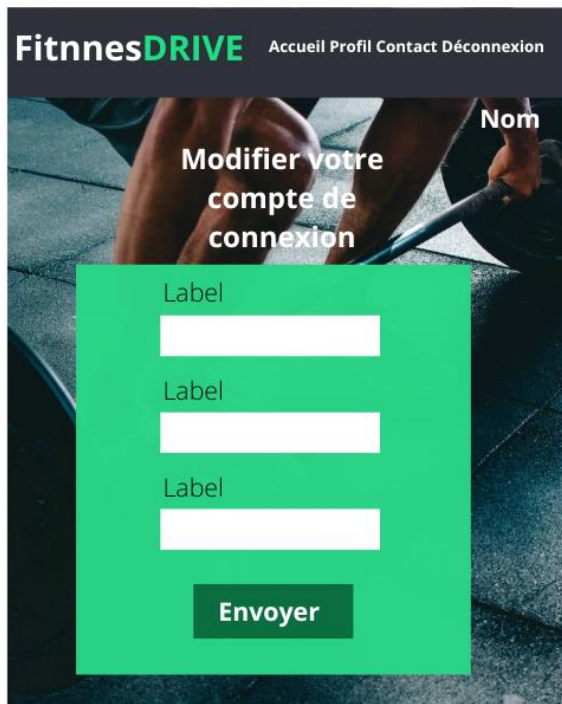
#### Page Login



#### Page Home



## Profil



FitnesDRIVE Accueil Profil Contact Déconnexion

Nom

Modifier votre compte de connexion

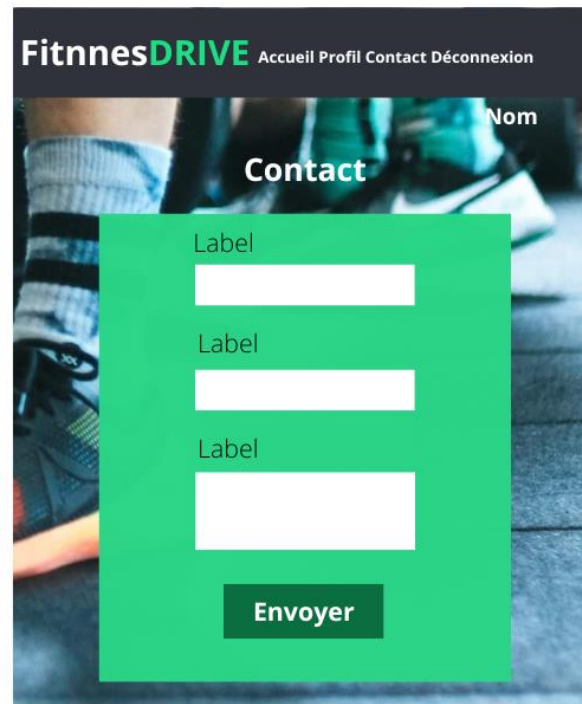
Label

Label

Label

Envoyer

## Contact



FitnesDRIVE Accueil Profil Contact Déconnexion

Nom

Contact

Label

Label

Label

Envoyer

(Toutes les Mockups sont en Annexe p :49.)

## 5. Spécifications techniques

Pour la partie maquettage, j'ai utilisé **Canva** pour la création des personas, de la planche graphique, des wireframes, des Mockups et du logo. Les User Stories je les ai réalisés avec **Word**. J'ai également fourni un mode d'emploi à destination du personnel de la marque sous forme de slides sur google, qui est dans mon dépôt **GitHub**.

Pour la partie Frontend, les technologies que j'ai utilisées sont **HTML**, **CSS (sass)**, **JavaScript** mais aussi **Bootstrap** et **Npm** pour installer les dépendances javascript et pour charger les modifications CSS ou le JS dans mon projet.

Pour la partie Backend, les diagrammes et les MCD, MLD ont été réalisés avec **iodraw.com**. J'ai choisi de développer ce projet avec **PHP** et du framework **Symfony**. **Composer** m'a permis d'installer les dépendances dont j'avais besoin pour ce projet.

Pour Ma Base de Données, grâce à **Composer**, j'ai pu installer **ORM** qui m'a permis de créer ma base de données, les entités avec **doctrine** et de créer ma variable. Avec tous ses éléments, j'ai pu me connecter à ma base de données. J'ai utilisé **PHPMyAdmin** au début

et une fois mon projet mis en ligne j'ai pu l'exporter sur **Workbench**. Les deux utilisent le langage **SQL**.

Pour les recherches, j'ai utilisé **Google**, les **sites officiels** de **PHP**, **Symfony** et **MySQL**. Ainsi que beaucoup de **tutos** disponibles sur le web et les cours **de Studi**.

Mais aussi, j'ai créé un service pour l'envoi de mail automatisé lors de la création ou de modification d'un compte avec **mailtrap.io**. J'ai déployé mon application sur **Heroku** avec l'aide du bundle [nat/deploy](#).

Mon éditeur de code est **VisualStudio Code**

Et enfin le plus important sans quoi rien ne serait possible mon PC, **un LAPTOP-6SOHJ3UE 64bits**.

Pour la partie responsive, j'ai utilisé le framework Bootstrap et CSS (sass) avec des médiasQuéry.

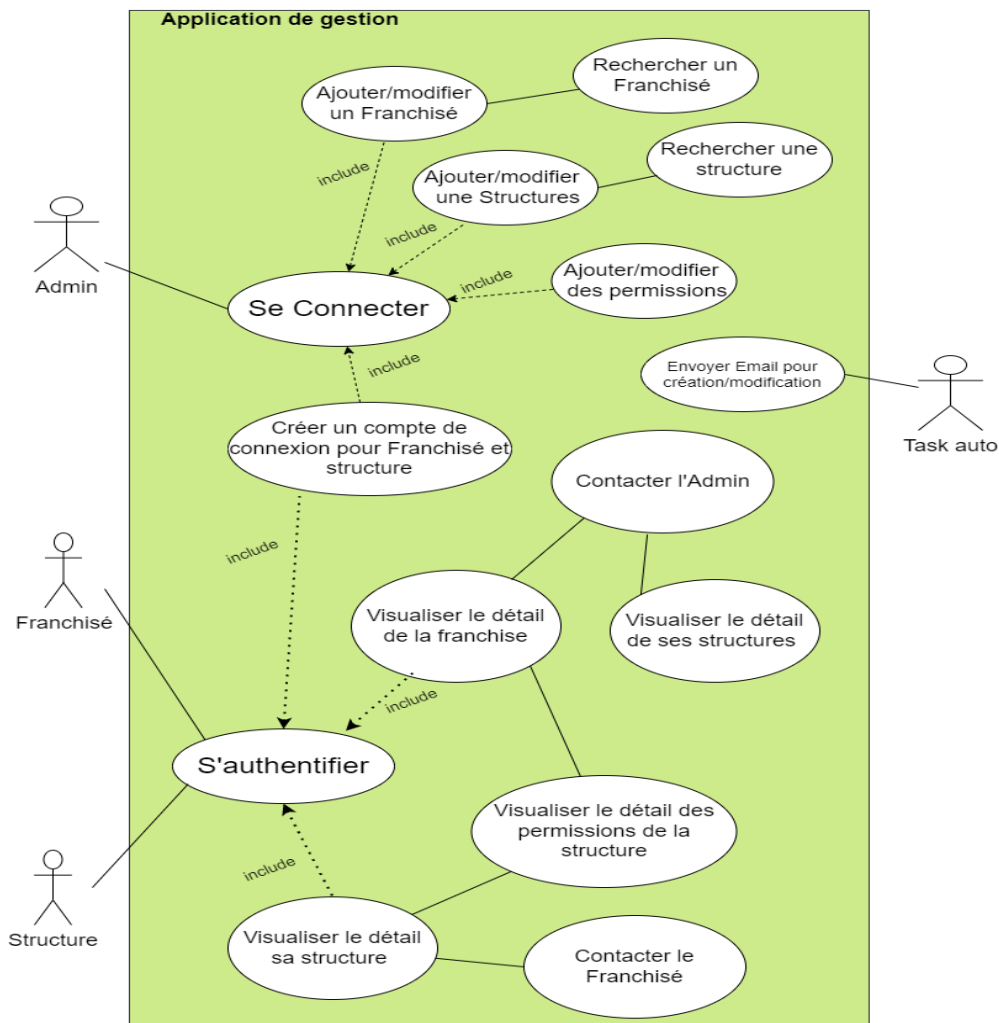
# La réalisation du projet

## 1. Diagrammes

J'ai commencé par composer des diagrammes qui permettent d'expliquer et de visualiser simplement de nombreux aspects du projet. Cela représente une retranscription des fonctionnalités issues du cahier des charges. Ils permettent de s'assurer que les besoins sont bien identifiés et fournissent aux développeurs le contexte du projet.

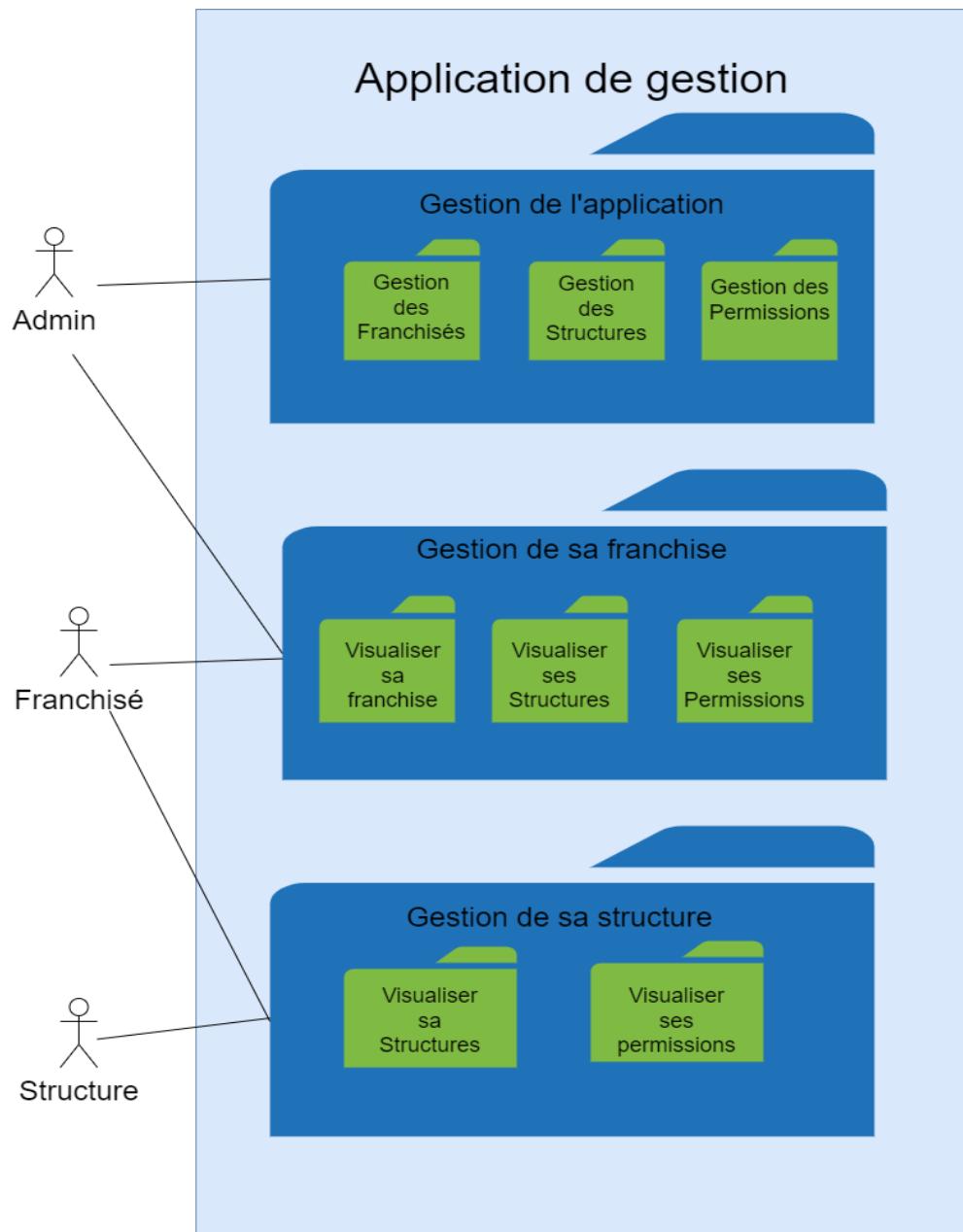
- Cas d'utilisation

Il définit les différents types d'utilisateurs et les fonctionnalités auxquelles ceux-ci ont accès. Il représente toutes les interactions possibles de l'application et met en valeur deux éléments, les acteurs et leurs fonctionnalités.



- Diagramme de Package

Ce diagramme représente une vision d'ensemble du projet et regroupe les cas d'utilisations en différentes parties structurées

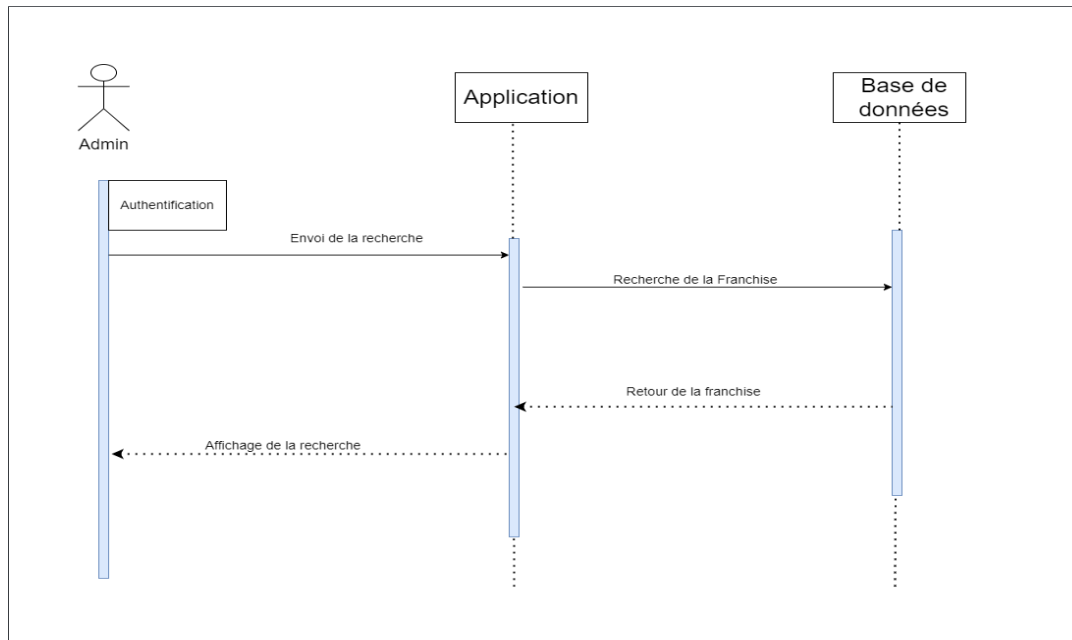




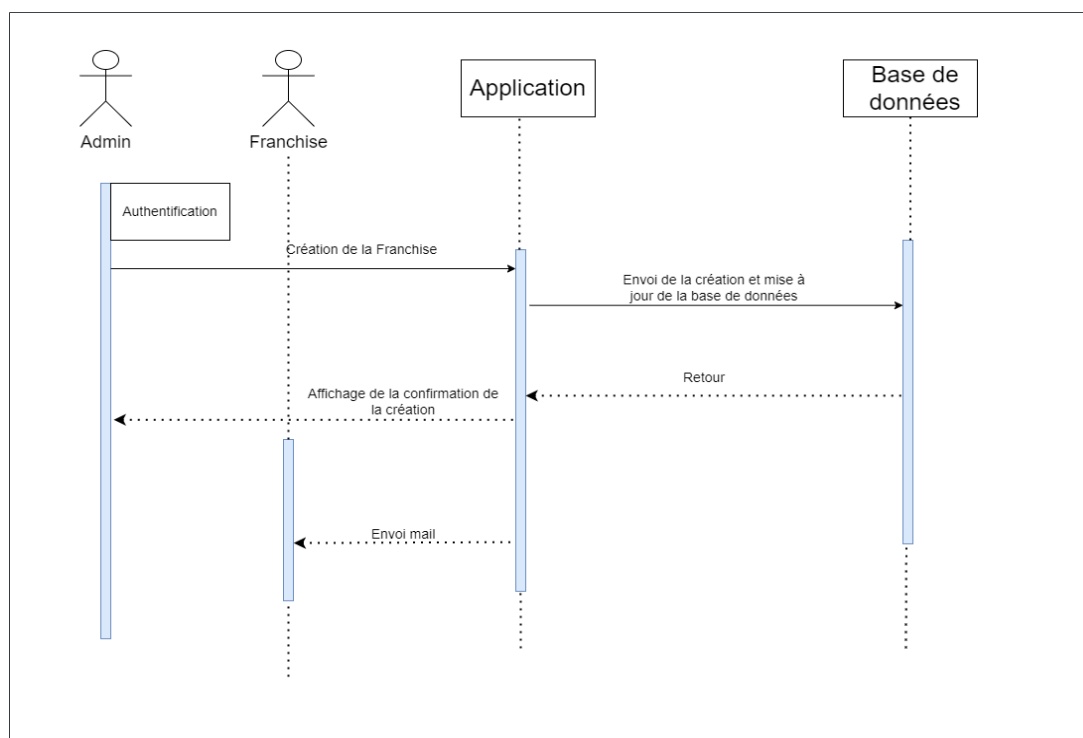
- Diagrammes de séquence

Il met en valeur les communications entre les différents éléments constitutif de l'application. Il illustre pour un cas d'utilisation, toutes les interactions entre un acteur, le produit lui-même et tous les systèmes externes.

### US9 – Recherche dynamique de la franchise :



### US3 – Création de la Franchise



## 2. Méthode merise

C'est une méthode de conception, de développement et de réalisation de projets informatiques. Elle améliore la communication entre tous les acteurs du processus de développement du projet. Elle encadre le projet grâce à des modèles et protège un développement hors sujet potentiels. Suivre les étapes de la méthodologie Merise, permet une analyse fine pour la création de la base de données. Cela se fait par étapes pour obtenir un modèle de données prêt pour la base de données de notre application. Ça permet d'avoir une base de données bien construite et qui ne sera pas ou peu modifier.

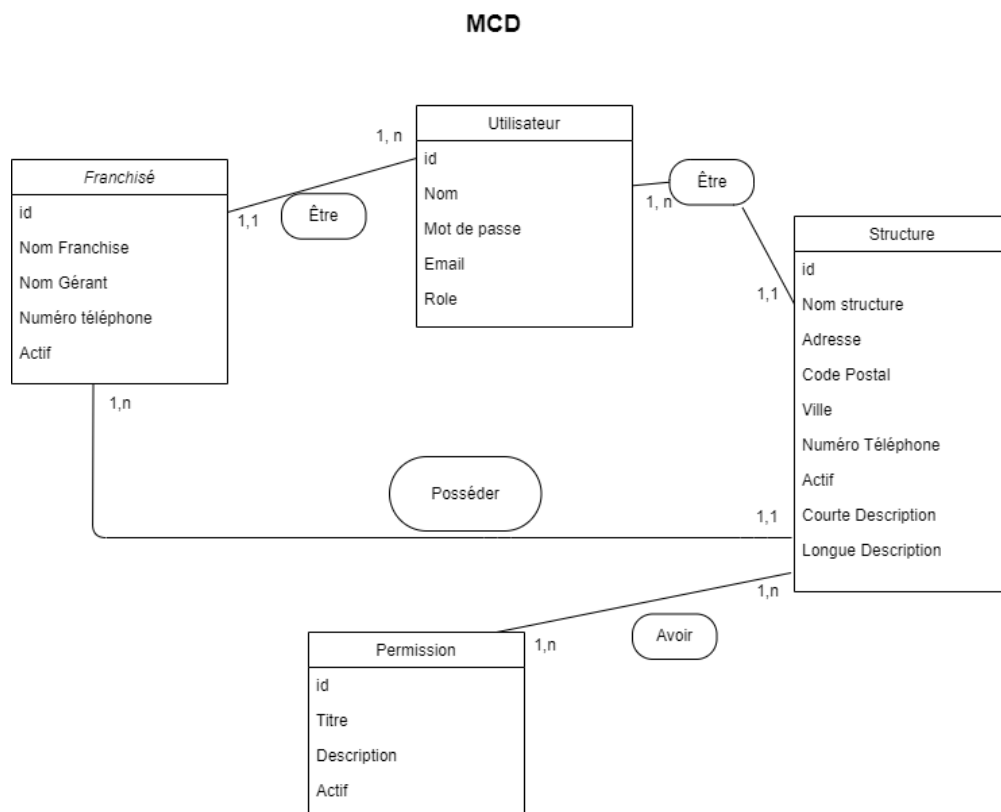
Il y a trois niveaux d'analyses :

### a. MCD - Modèle Conceptuel de données

Décrit les acteurs du système (soit des personnes, soit des objets), leurs caractéristiques, et leurs interactions. Il est concrétisé par un schéma **Entités-Association** constitué d'un ensemble **d'entités, d'associations, de propriétés, d'identifiants, de cardinalités.**

Pour produire un MCD les étapes à suivre sont :

- Déterminer les entités,
- Déterminer les attributs des entités,
- Déterminer les identifiants des entités,
- Déterminer les associations des entités,
- Déterminer les éventuels attributs des associations,
- Déterminer les cardinalités des associations,
- Faire la représentation graphique.



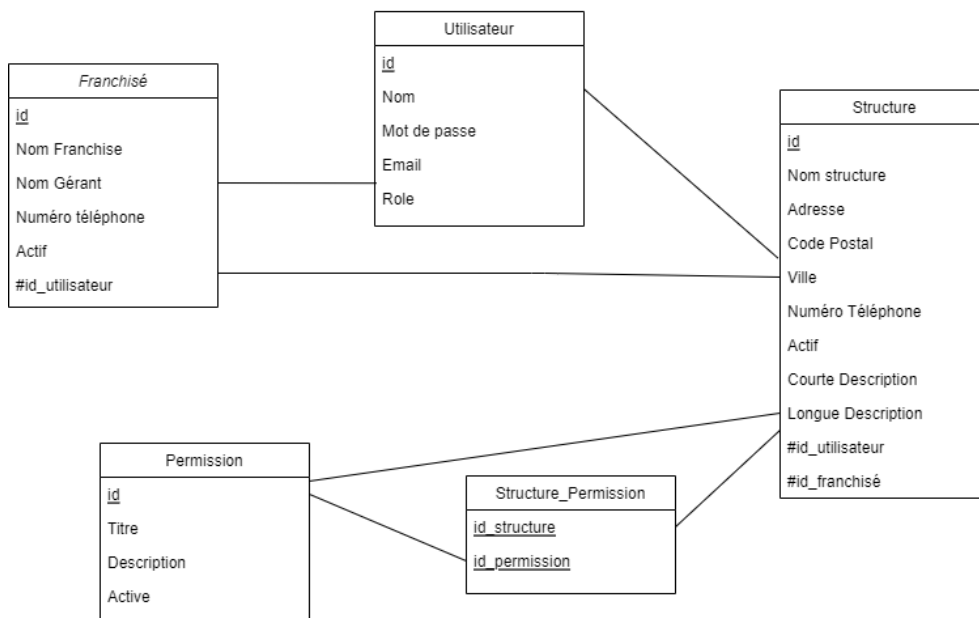
## b. MLD - Modèle Logique de données

Une fois le schéma conceptuel de votre base de données effectué, il faudra établir une logique entre les différentes tables. Le Modèle Logique de Données (MLD) est un modèle extrait à partir du MCD.

On définit :

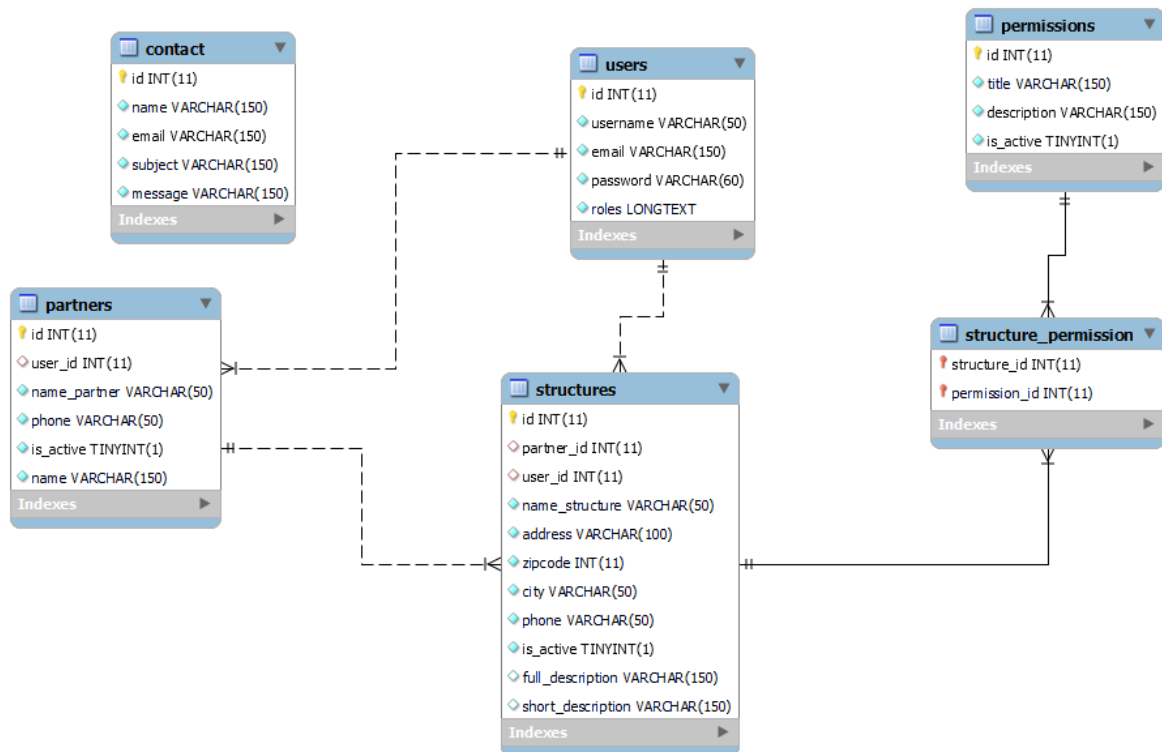
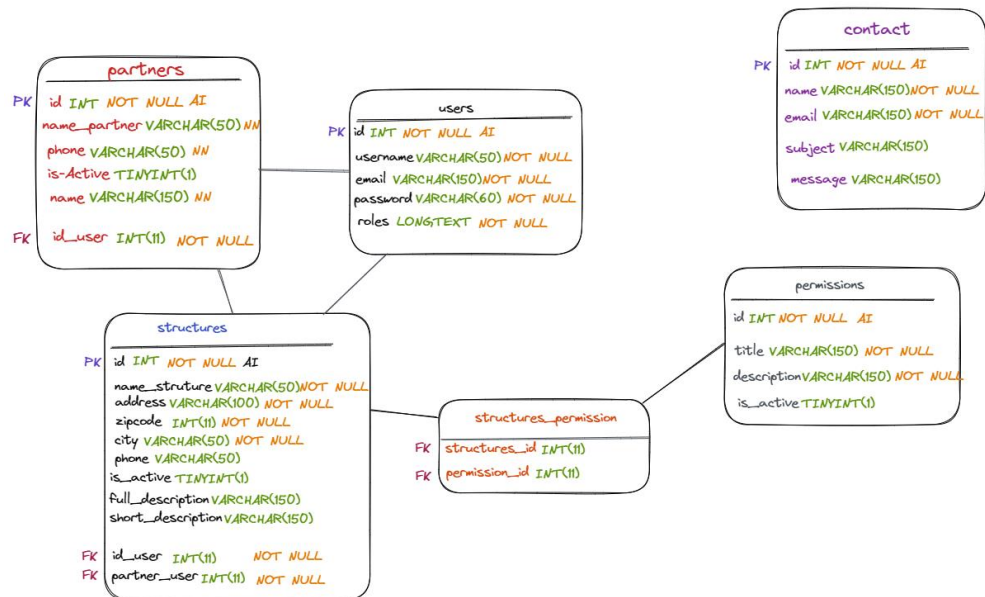
- Les tables
- Les clés primaires
- Les clés étrangères

### MLD



## c. MPD - Modèle Physique de données

Dernière étape avant l'implémentation réelle de la base de données. Le MPD permet d'établir la manière dont le système sera mis en place et permet de préciser les systèmes de stockage employés. Il s'agit donc à cette étape de préparer l'implémentation d'un SGBD.



### 3. Développement du projet

Ce projet a été développé avec le pattern MVC, qui permet de bien structurer le code source et de le découper en trois parties avec les technologies comme doctrine pour ORM (**M**odel), le moteur de template Twig (**V**iew) et le framework Symfony et PHP (**C**ontroller).

#### a. Installation

J'ai ouvert un terminal, je me suis rendu dans ...xampp/htdocs/mes-sites, j'ai lancé ces commandes :

***Symfony check:requirements*** , pour vérifier si les prérequis sont bien installés (PHP, Symfony, Composer, Symfony CLI, nodejs et npm)

***symfony new fitness-drive --webapp***, qui m'a créé un dossier fitness-drive.

***symfony server:start***, pour vérifier si l'installation a bien marcher

J'ai installé à l'aide de composer certaines dépendances dont j'ai eu besoin au début de mon projet, ***Composer require annotations***,

```
#config/routes.yaml
controllers:
    resource: ../src/Controller/
    type: annotation
```

Et initialiser un dépôt sur Github, pour la sauvegarde de mon projet.

#### b. Initialisation

Une fois l'installation de toutes les dépendances effectuée, j'ai pu configurer mes variables d'environnement dans un fichier .env.local.php. Comme la variable nécessaire à la connexion à la base de données : le host, l'utilisateur, le mot de passe et le nom de la base de données

```
APP_ENV=dev
APP_SECRET=4955e7a99ffa42699ae908461df3d9e3
MESSENGER_TRANSPORT_DSN=doctrine://default?auto_setup=0
DATABASE_URL="mysql://root@127.0.0.1:3306/fitness_drive?serverVersion=mariaDB-10.4.24&charset=utf8mb4"
```

Ensuite l'installation de ORM qui me permettra de faire la connexion avec ma base de données, grâce à doctrine, avec cette commande :

**composer require symfony/orm-pack**

**composer require --dev symfony/maker-bundle**

Cette commande : **php bin/console doctrine:database:create** m'a permis de créer ma base de données.

### c. Développement de la base de données

La commande : **php bin/console make:entity** permet de créer des modèles qui correspondent à chacune de nos entités. On peut y définir les relations, les associations avec les autres Entités.

```
<?php

namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;
use App\Repository\PartnerRepository;
use Doctrine\Common\Collections\Collection;
use Doctrine\Common\Collections\ArrayCollection;
use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;

#[ORM\Entity(repositoryClass: PartnerRepository::class)]
#[ORM\Table(name: '`partners`')]
#[UniqueEntity(fields:'namePartner', message:'Le nom que vous avez
indiqué existe déjà')]
#[UniqueEntity(fields:'phone', message:'Le numéro de téléphone que
vous avez indiqué existe déjà ')]
#[UniqueEntity(fields:'name', message: 'Le nom que vous avez
indiqué existe déjà')]

class Partner
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 50, unique:true)]
    private ?string $namePartner = null;

    #[ORM\Column(length: 50,unique:true)]
    private ?string $phone = null;

    #[ORM\Column]
    private ?bool $is_active = null;

    #[ORM\OneToMany(mappedBy: 'partner', targetEntity:
Structure::class)]
    private Collection $structures;
```

```
#[ORM\ManyToOne(inversedBy: 'partner', cascade: ['persist',
'remove'])]
private ?User $user = null;
```

Extrait de l'entité Partner

La création et la mise à jour de la base de données se fait via la création de fichiers de migrations grâce aux commandes : **php bin/console make:migration , php bin/console doctrine:migrations:migrate.**

J'ai exécuté cette commande une fois que toutes mes entités ont été créées ainsi que leurs relations.

```
public function up(Schema $schema): void
{

$this->addSql('CREATE TABLE `partners` (id INT AUTO_INCREMENT NOT NULL,
user_id INT DEFAULT NULL, name_partner VARCHAR(50) NOT NULL, phone
VARCHAR(50) NOT NULL, is_active TINYINT(1) NOT NULL, name VARCHAR(150) NOT
NULL, UNIQUE INDEX UNIQ_EFEB516439407BA8 (name_partner), UNIQUE INDEX
UNIQ_EFEB5164444F97DD (phone), UNIQUE INDEX UNIQ_EFEB51645E237E06 (name),
INDEX IDX_EFEB5164A76ED395 (user_id), PRIMARY KEY(id)) DEFAULT CHARACTER SET
utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');

$this->addSql('ALTER TABLE `partners` ADD CONSTRAINT
FK_EFEB5164A76ED395 FOREIGN KEY (user_id) REFERENCES `users` (id)');
```

Extrait du fichier de migration pour Entité Partner

#### Insertion d'une donnée User avec le ROLE\_ADMIN sur phpMyAdmin

```
INSERT INTO `users` (`id`, `username`, `email`, `password`, `roles`) VALUES
```

```
(4, 'Admin01', 'fitnessdrive.ad01@outlook.com',
'$2y$10$Ae6uNGMhPOjzV2g6aDcymeDRTTxmHndBXEW15II56kJR7KIu8sRXy', '['ROLE_ADMIN']');
```

Pour hacher le mot de passe, je me suis rendu sur le site [www.bcrypt.fr](http://www.bcrypt.fr).

#### Insertion des données Permissions sur phpMyAdmin

```
INSERT INTO permissions (title, description, is_active) VALUES
```

```
('Distributeur de Boisson', 'Eau, soda, boisson énergisante...', true),
```

```
('Distributeur de Nourriture', 'Barre vitaminée, chocolatée, chips...', true),
```

```
('Vente de produit de la Marque', 'Sportif, vêtement, poster...', true),
```

```
('Gérer le planning', 'Application de gestion de planning des coachs sportif', true),
```

#### d. Développement des routers et controllers

La base de données mise à jour, j'ai pu commencer à développer les controllers qui vont me permettre d'accéder aux données stockées dans la base de données. Chaque entité possède son controller avec ses requêtes, sa route et son template twig associé.

```
namespace App\Controller;

#[Route('/partner')]

class PartnerController extends AbstractController
{
    #[Route('/show/{id<\d+>}', name: 'app_partner_show')]

    public function index(ManagerRegistry $doctrine, int $id, Partner
$partner, PartnerRepository $partnerRepo): Response
    {
        $repositoryPartner = $doctrine->getRepository(Partner::class);
        $partner = $repositoryPartner->find($id);
        $structures= $partner->getStructures();

        return $this->render('partner/index.html.twig',[
            'partner'=> $partner,
            'structures' => $structures,
        ]);
    }
}
```

Extrait du Controller pour Entité Partner

#### e. Développement du panel d'Admin

J'ai créé un controller Admin qui contient toutes les requêtes pour accéder aux données de chaque entité. Mais également quatre autres controllers (ex : NewPartnerController) qui permettent la création ou la modification d'un utilisateur, une franchise, une structure et une permission.

```
namespace App\Controller;

class NewPartnerController extends AbstractController
{
    #[Route('/newpartner', name: 'new_partner')]

    public function formNewPartner(Request $request, EntityManagerInterface
$entityManager, Partner $partner=null, MailService $mail): Response
    {
        if(!$partner){
```



```

        $partner = new Partner();
    }
    $formPartner = $this->createForm(PartnerType::class, $partner);
    $formPartner->handleRequest($request);

    if ($formPartner->isSubmitted() && $formPartner->isValid()){
        if(!$partner->getId()){
        }
        $structures = $partner->getStructures();
        foreach ($structures as $structure) {
            $partner->addStructure($structure);
        }
        $entityManager->persist($partner);
        $entityManager->flush();
        $this->addFlash('success', 'La Franchise a bien été créé');
    }

    return $this->render('admin/partner/newPartner.html.twig',[
        'formpartner'=>$formPartner->createView(),
    ]);
}

```

Extrait du fichier de migration pour Entité Partner

## f. Développement d'un service Email

Il est également demandé que pour chaque nouvelle création, un email soit envoyé automatiquement. Dans le dossier src, j'ai créé un dossier Service avec un fichier MailService.php qui contient toute la logique pour l'envoi de mail. J'ai rajouté également une variable d'environnement

MAILER\_DSN=smtp://8d4988e29245f2:61d74c2fc9b32d@smtp.mailtrap.io:2525?encryption=tls&auth\_mode=login

```

namespace App\Service;
class MailService
{
    public function __construct(private MailerInterface $mailer){}

    public function sendEmail(string $from, string $to, string $subject,
string $template, array $context ) :void
    {
        $email = (new TemplatedEmail())
            ->from($from)
            ->to($to)
            ->subject($subject)
            ->htmlTemplate("email/$template.html.twig")
            ->context($context);
    }
}

```

```

        $this->mailer->send($email);
    }
}

```

Extrait du fichier de MailService.php

Ensuite j'ai injecté ce service dans les différents controllers concernés avec les bonnes données.

```

//sendEmail
    $mail->sendEmail(
        'fitnessDrive@outlook.fr',
        $user->getEmail(),
        'Activation de votre compte',
        'registerUser',
        compact('user')

    );

    $this->addFlash('send', 'Email d\'Activation a bien été
envoyé');

```

Extrait du fichier de NewUserController

## g. Développement des Formulaires

Pour permettre aux employés de la marque de créer ou de modifier les différents acteurs, j'ai dû créer des formulaires de création de compte avec l'aide de la commande :

**composer require symfony/form**

J'ai modifié le fichier avec des labels, des requêtes, des constraints propre à chaque entité.

```

namespace App\Form;

class PartnerType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('user', EntityType::class,[
                'class'=> User::class,
                'label'=> 'Sélectionner un compte',
                'placeholder'=>'Choisissez ...',
                'choice_label'=> function (User $user){
                    return $user->getUsername();
                },
                'query_builder' => function (UserRepository $userRepo) {
                    $qb = $userRepo->createQueryBuilder('u');
                    return $qb
                        ->where('u.roles LIKE :role')
                        ->setParameter('role', '%" . 'ROLE_PARTNER' . '%"');
                }
            ]);
    }
}

```

```

        ->orderBy('u.username');}
    ])
    ->add('name', TextInputType::class,[
        'label'=> 'Nom du franchisé',
        'attr'=>[
            'placeholder'=>'Son nom',
        ],
        'required'=> true,
        'constraints'=> [
            new NotBlank(['message'=> 'Le champ ne peut pas être
vide !']),
            new Length([
                'min'=> 5, 'max'=> 100, 'minMessage' =>'Le champ
doit faire entre {{ limit }} et {{ limit }} caractères', 'maxMessage' =>'Le
champ doit faire entre {{ limit }} et {{ limit }} caractères']),
            ]
        ])
    ...
}

```

Extrait du fichier de PartnerType

## 4. Développement de la partie frontend

### a. Mise en place de l'environnement de développement

J'ai choisi d'utiliser le moteur de template Twig utilisé par défaut par le framework Symfony. Une fois twig installer sur mon projet, cela a généré un dossier template où j'ai pu créer un dossier pour chaque controller avec des fichiers qui seront soit mes pages, soit des morceaux de code que je peux inclure dans mes différentes vues.

#### Exemple de l'arborescence du dossier admin/

Partner/

- `_formPartner.html.twig` => morceau de code
- `_listpartner.html.twig` => morceau de code
- `index.html.twig`
- `newpartner.html.twig`
- `showpartner.html.twig`

Ces commandes **composer require symfony/webpack-encore-bundle** et `npm install`, me permettent d'installer les dépendances dont j'ai besoin pour travailler le front.

### b. Développement de la partie visuel de l'application

Les fichiers renvoient un contenu html mais on peut récupérer des données du controllers associés pour les afficher. On peut y insérer des conditions ou des boucles pour parcourir un tableau de données.

En règle général, on crée un fichier `base.html.twig`, et ensuite sur chaque page on rajoute

```
{% extends 'admin/index.html.twig' %}
```

qui permet d'hériter de `base.html.twig`

Chaque élément de la page se trouve dans des blocks personnalisés.

```
{% block body %} Ici Le contenu personnalisé pour la page {% endblock %}
```

Récupère le nom du partner selon la requête du controller :

```
{{partner.namePartner}}
```

Inclus un morceau de code :

```
{% include "_partials/_headerAdmin.html.twig" %} => inclus un morceau de code
```

Twig permet de réaliser des pages très rapidement avec des données de l'application

```
{% extends 'base.html.twig' %}

{% block title %}Admin | {{ parent() }} {% endblock %}
```

```
{% block body %}
...
<main class='main-container col-12'>
  <section>
    <div class="row">
      <div class='count col-12'>
        <span class='total'>Total :
          {{ countPartners }} Franchises |
          {{ countStructures }} Structures
        </span>
      </div>
    </div>
  </section>

  <h2 class="titleIndex titleListAdmin"
id="accueil">{{titlePage}}</h2>

  ...

</main>
</body>
{% endblock %}
```

Extrait d'un fichier admin/index.html.twig

### c. Navigation au sein de l'application

Afin de mieux sécuriser mes routes, j'ai choisi de faire deux interfaces :

- La partie Admin, qui a sa propre Barre de navigation et une Sidebar ainsi que les pages qui permettent la création d'utilisateur. L'interface est très épurée.
- La partie pour les franchises et les structures avec une interface plus remplie.

J'ai utilisé Bootstrap pour les barres de navigations avec du sass et des médiaQuery

```
/******MEDIA QUERY POUR ECRAN 576PX *****/
/******min-width 576px*****/
@include media($sm)
  .fas-fa
    margin-right: 20px

  .textConnected
    font-size: 0.6em
    padding: 0
    margin: 0

/******MEDIA QUERY POUR ECRAN 768PX *****/
/******min-width 768px*****/
@include media($md)
```

Extrait d'un fichier \_sidebar.sass

## d. Intégration du Javascript

Il est également demandé de développer une recherche dynamique. J'ai créé mon input qui permettra de récupérer la saisie de l'utilisateur.

```
<!-- Search all/active/Inactive Start-->
{% block searchLink %}
  <section class="searchLink">
    <div class="row">
      <div class="col-12">
        <h2 class='titleIndex' id='backL'>{{ titleIndex
}}</h2>

        <div class="items_check">
          <a class='active item_check check' id="all"
alt="Affichez toutes les franchises" title="Affichez toutes les franchises"
href='#>Toutes</a>

          <a class='item_check check' id="active"
alt="Affichez uniquement les franchises Active" title="Affichez uniquement
les franchises Active" href='#>Active</a>

          <a class='item_check check' id="inactive"
alt="Affichez uniquement les franchises Inactive" title="Affichez uniquement
les franchises Inactive" href='#>Inactive</a>
        </div>
      </div>
    </div>
  </section>
{% endblock %}
<!-- Search all/active/Inactive end-->
```

Extrait du fichier admin/partner/index.html.twig

Je récupère la saisie :

```
//BarSearch
const inputsearch = document.getElementById("searchbar");
const partners = document.getElementsByClassName('partner');
```

Extrait du fichier partner.js

J'affiche la recherche et je cache les autres :

```
// function barsearch
const mySearch = () => {
  let valueInput = inputsearch.value;
  for( i=0; i< partners.length; i++) {
    if (partners[i].innerHTML.toLowerCase().indexOf(valueInput)
!= -1) {
      partners[i].classList.remove("d-none");
    }
  }
}
```

```

        partners[i].classList.add("d-flex");
    } else {
        partners[i].classList.remove("d-flex");
        partners[i].classList.add("d-none");
    }
}
};
inputsearch.addEventListener("change", mySearch);

```

Extrait du fichier partner.js



Extrait du site

Une fois la fonction créer, il faut taper la commande : `npm run watch` ou `build`, ce qui permet de charger les langages front.

### e. Organisation de la partie Front

Pour les styles, dans mon dossier styles, j'ai créé plusieurs sous-dossiers qui représentent les parties de mon site. Par exemple :

Styles/components/ => j'y est inclus des fichiers pour la barre de navigation, le breadcrumb...

Pour le Javascript, chaque entité a sa page. Exemple partner.js

## Fonctionnalité la plus représentative : L'authentification

L'utilisateur a besoin d'être connecté pour avoir pleinement accès à certaines fonctionnalités. Il fallait donc constituer une passerelle entre l'application et le front pour authentifier l'utilisateur, avec une attention toute particulière sur la sécurisation du processus.

L'authentification se fait avec l'entité User, donc à l'aide de la commande **php bin/console make:user**, j'ai créé mon entité User.

Ensuite j'ai créé un formulaire de connexion grâce à cette commande **symfony console make:auth**. Cela m'a généré ces fichiers

```
created: src/Security/LoginFormAuthenticator.php
updated: config/packages/security.yaml
created: src/Controller/SecurityController.php
created: templates/security/login.html.twig
```

J'ai modifié ce fichier pour qu'il représente mon entité User avec ses constraints... :

```
class LoginUserType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('email', EmailType::class,[
                'required'=> true,
                'constraints'=> [
                    new NotBlank(['message'=> 'Veuillez saisir un Email
valide !']),
                    new Email(['mode'=>'html5', 'message'=>'L\'adresse {{
value }} n\'est pas valide']),
                ]
            ])
            ->add('password', PasswordType::class,[
                'required'=> true,
                'constraints'=> [
                    new NotBlank(['message'=> 'Veuillez saisir un mot de
passe!']),
                    new Length([
                        'min'=> 8, 'max'=> 100, 'minMessage' =>'Votre
mot de passe doit contenir au moins {{ limit }} caractères',]),
                ]
            ])
        ;
    }
}
```

Extrait du fichier LoginUserType



J'ai vérifié si le provider est bien paramétré

```
providers:
  app_user_provider:
    entity:
      class: App\Entity\User
      property: email
```

Ainsi que les routes de login et logout

```
main:
  lazy: true
  provider: app_user_provider
  custom_authenticator: App\Security\LoginFormAuthenticator
  logout:
    path: app_logout
    target: /
```

Mais j'ai également protégé mes routes grâce au moteur de template twig. J'ai inséré des conditions

```
{% block body %}

{% if app.user %}
    <h3 class=''>Bienvenue {{ app.user.username }}</h3>

    {% if is_granted('ROLE_ADMIN') %}
        <a class="btn btnHome btn-sm " role="button"
title="Connectez-vous " alt="Connectez-vous " href="{{ path('app_admin',
{'id':id}) }}">Connexion</a>

    {% elseif is_granted('ROLE_PARTNER') %}

        <a class="btn btn-sm btnHome" role="button"
title="Connectez-vous " alt="Connectez-vous " href="{{
path('app_partner_show', {'id': id}) }}">Connexion </a>

    {% elseif is_granted('ROLE_STRUCTURE') %}

        <a class="btn btn-sm btnHome" role="button"
title="Connectez-vous " alt="Connectez-vous " href="{{
path('app_structure_show', {'id': id}) }}">Connexion</a>
    {% endif %}

{% endblock %}
```

```

    {% endif %}
    </div>
  </div>
</main>
</body>
{% endblock %}

```

Extrait du fichier [home.html.twig](#)

Lors de l'authentification sur le site, si l'administrateur se connecte, il arrive directement sur sa page d'accueil et si c'est un autre utilisateur, il est redirigé vers une page de transition et en cliquant sur le bouton, il pourra accéder à sa page.

Sur cette page il n'y a aucune autre action possible.

Dans le fichier config/security.yaml, j'ai rajouté de la sécurité :

Avec les rôles pour l'entité User :

```

security:
  role_hierarchy:
    ROLE_STRUCTURE: [ROLE_STRUCTURE]
    ROLE_PARTNER : [ROLE_PARTNER]
    ROLE_ADMIN : [ROLE_ADMIN, ROLE_PARTNER, ROLE_STRUCTURE]

```

Extrait du fichier sécurité.yaml

Mais aussi j'ai sécurisé les routes de l'application

```

access_control:
  - { path: ^/admin, roles: ROLE_ADMIN }
  - { path: ^/partner, roles: ROLE_PARTNER }
  - { path: ^/structure, roles: ROLE_STRUCTURE }
  - { path: ^/home, roles: [ROLE_ADMIN, ROLE_PARTNER,
ROLE_STRUCTURE] }
  - { path: ^/contact, roles: [ROLE_ADMIN, ROLE_PARTNER,
ROLE_STRUCTURE] }

```

Extrait du fichier sécurité.yaml

Une fois toute la configuration de la sécurité, j'ai pu tester et tout fonctionne.

## Recherche anglophone

Ayant très peu abordé la notion d'authentification et de sécurité d'une application lors de ma formation, je me suis documentée sur le site officiel de symfony. Voici un extrait, ainsi que sa traduction :

## **Create your User Class**

No matter *how* you will authenticate (e.g. login form or API tokens) or *where* your user data will be stored (database, single sign-on), the next step is always the same: create a "User" class. The easiest way is to use the [MakerBundle](#).

Let's assume that you want to store your user data in the database with Doctrine.

That's it! The command asks several questions so that it can generate exactly what you need. The most important is the file itself. The *only* rule about your class is that it *must* implement [UserInterface](#).

## **The "User Provider"**

In addition to your class, you also need a "User provider": a class that helps with a few things, like reloading the User data from the session and some optional features, like [remember me](#) and [impersonation](#).

## **Encoding Passwords**

Not all applications have "users" that need passwords. *If* your users have passwords, you can control how those passwords are encoded in . The command will pre-configure this for you:

Now that Symfony knows *how* you want to encode the passwords, you can use the service to do this before saving your users to the database. `UserPasswordEncoderInterface`

### Traduction :

Quelle que soit la *façon dont* vous allez vous authentifier (par exemple, formulaire de connexion ou jetons API) ou *où* vos données utilisateur seront stockées (base de données, authentification unique), l'étape suivante est toujours la même : créer une classe « User ». Le moyen le plus simple est d'utiliser le [MakerBundle](#).

Supposons que vous souhaitiez stocker vos données utilisateur dans la base de données avec Doctrine. Voilà ! La commande pose plusieurs questions afin qu'elle puisse générer exactement ce que tu as besoin. Le plus important est le fichier lui-même. La *seule* règle concernant ta classe est qu'elle *doit* implémenter [UserInterface](#).

En plus de ta classe, tu as également besoin d'un « User provider »: une classe qui aide, comme le rechargement des données utilisateur de la session et certaines fonctionnalités facultatives, telles que [Se souvenir de moi](#) et [Usurpation d'identité](#)

Toutes les applications n'ont pas d'« utilisateurs » qui ont besoin de mots de passe.

*Si* vos utilisateurs ont des mots de passe, Vous pouvez contrôler la façon dont ces mots de passe sont encodés.

Maintenant que Symfony sait *comment* je veux encoder les mots de passe, je peux utiliser le service pour le faire avant d'enregistrer mes utilisateurs dans la base de données.

# Conclusion

Concernant le projet, les fonctionnalités de base sont développées. Il a été le premier projet développer avec le langage PHP et le framework Symfony. J'ai beaucoup appris lors de son développement. Mon projet est mis en ligne sur heroku et il est fonctionnel.

Le développement est un défi de tous les jours, car il me faut toujours apprendre, à l'aide de la documentation officielle, à l'aide des forums dédiés, de tutoriels vidéo, et reproduire ce que je viens d'apprendre. Je pense que c'est la meilleure méthode de travail afin de progresser rapidement, et que c'est une bonne façon d'entrevoir ce qu'est le métier de développeur.

Lien pour accéder à mon Application :

<https://virginie-giacometti.netlify.app/>

Lien GitHub :

[lana-12/ecf-fitnessDrive: ECF \(github.com\)](https://github.com/lana-12/ecf-fitnessDrive)

Code pour tester l'application :

[fitnessdrive.ad01@outlook.com](mailto:fitnessdrive.ad01@outlook.com) => Ad01\_FitnessDrive

[ibanez@exemple.fr](mailto:ibanez@exemple.fr) => franchise

[trinhduc@exemple.fr](mailto:trinhduc@exemple.fr) => structure

# ANNEXE

## A. Gestion des Utilisateurs

- Scénarios de test d'une fonctionnalité de création d'un nouvel utilisateur :

<b>Titre de l'us :</b> US2 – Création de Compte de Connexion
<b>En tant que,</b> Julie <b>Je souhaite</b> créer un nouvel utilisateur <b>Afin que</b> le nouvel utilisateur accède à son compte
<b>Règles métier</b>  1. L'E-mail de la franchise ou de la structure doit être unique  2. Le mot de passe doit faire plus de 8 caractères
<b>Critères d'acceptation</b>  <b>Scénario 2 :</b> Création d'un compte de connexion  <b>Étant donné que,</b> je possède un compte utilisateur pour Julie  <b>Lorsque,</b> j'accède à la page « Admin » /admin  <b>Et que</b> je clique sur « Créer un compte utilisateur »  <b>Et que</b> je saisis son nom dans le champ « Nom du Gérant »  <b>Et que</b> je saisis son E-mail dans le champ « E-mail »  <b>Et que</b> je saisis son Mot de Passe dans le champ « Mot de Passe »  <b>Et que</b> je sélectionne son rôle « Franchise » ou « Structure »  <b>Et que</b> je clique sur le bouton « Ajouter »  <b>Alors</b> j'ai créé un nouvel utilisateur  <b>Et</b> un E-mail est envoyé automatiquement à ce nouvel utilisateur avec ses identifiant de connexion

- Scénarios de test d'une fonctionnalité de création d'une nouvelle Franchise :

<b>Titre de l'us :</b> US3 – Création d'une Nouvelle Franchise
<b>En tant que,</b> Julie <b>Je souhaite</b> créer une nouvelle Franchise <b>Afin que</b> la Franchise puisse visualiser ses permissions et ses structures

### Règles métier

1. Le Gérant de la franchise doit posséder un compte de connexion
2. L'administrateur doit être connecter

### Critères d'acceptation

**Scénario 3 :** Création d'une Nouvelle Franchise

**Étant donné que,** je possède un compte utilisateur pour Julie

**Lorsque,** j'accède à la page « Admin » /admin

**Et que** je clique sur « Créer une Nouvelle Franchise »

**Et que** je sélectionne son compte de connexion « Compte de connexion »

**Et que** je saisis le nom de la franchise dans le champ « Nom de la Franchise »

**Et que** je saisis son numéro de téléphone dans le champ « Téléphone »

**Et que** je coche la case « Activer la franchise » ou pas

**Et que** je clique sur le bouton « Enregistrer »

**Alors** j'ai créé une nouvelle franchise

**Et** un E-mail est envoyé automatiquement au Gérant de la Franchise

- Scénarios de test d'une fonctionnalité de **Modification d'une nouvelle Franchise** :

**Titre de l'us :** US4 – Modification d'une Nouvelle Franchise

**En tant que,** Julie

**Je souhaite** Modifier une nouvelle Franchise

**Afin de** mettre à jour la Franchise

### Règles métier

1. La franchise doit être créée
2. L'administrateur doit être connecter

### Critères d'acceptation

**Scénario 4 :** Modification d'une Nouvelle Franchise

**Étant donné que,** je possède un compte utilisateur pour Julie

**Lorsque,** j'accède à la page « Admin » /admin

**Et que** je clique sur « Modifier une Nouvelle Franchise »

**Et que** je sélectionne son compte de connexion « Compte de connexion »

**Et que** je saisis les modifications de la franchise dans les champs concernés

**Et que** je clique sur le bouton « Enregistrer »

**Alors** j'ai modifié la franchise

**Et** un E-mail est envoyé automatiquement au Gérant de la Franchise

- Scénarios de test d'une fonctionnalité d'**Affichage des Franchises** :

<b><u>Titre de l'us</u> :</b> US5 – Affichage des Franchises
<p><b>En tant que</b>, Julie</p> <p><b>Je souhaite</b> Afficher les Franchises actives ou inactives</p> <p><b>Afin de</b> visualiser les Franchises qui sont actives ou inactives</p>
<p><b><u>Règles métier</u></b></p> <ol style="list-style-type: none"> <li>1. La franchise doit être créée</li> <li>2. L'administrateur doit être connecter</li> </ol>
<p><b><u>Critères d'acceptation</u></b></p> <p><b>Scénario 5</b> : Affichage des Franchises</p> <p><b>Étant donné que</b>, je possède un compte utilisateur pour Julie</p> <p><b>Lorsque</b>, j'accède à la page « Admin » /admin</p> <p><b>Et que</b> je coche la case « Afficher les Franchises « Active » ou « Inactive » »</p> <p><b>Alors</b> j'accède aux Franchises selon le choix sélectionné</p> <p><b>Et</b> une page contenant la liste des Franchises s'affiche</p>

- Scénarios de test d'une fonctionnalité de **création d'une nouvelle structure** :

<b><u>Titre de l'us</u> :</b> US6 – Création d'une Nouvelle Structure
<p><b>En tant que</b>, Julie</p> <p><b>Je souhaite</b> créer une nouvelle structure</p> <p><b>Afin que</b> La structure puisse visualiser ses permissions</p>
<p><b><u>Règles métier</u></b></p> <ol style="list-style-type: none"> <li>1. La Structure doit posséder un compte de connexion</li> </ol>

2. la franchise doit être créée

3. L'administrateur doit être connecter

#### **Critères d'acceptation**

**Scénario 6 : Création d'une Nouvelle Structure**

**Étant donné que**, je possède un compte utilisateur pour Julie

**Lorsque**, j'accède à la page « Admin » /admin

**Et que** je clique sur « Créer une Nouvelle Structure »

**Et que** je sélectionne son compte de connexion « Compte de connexion »

**Et que** je sélectionne la franchise « Sélectionner une franchise »

**Et que** je saisis le nom de la structure dans le champ « Nom de la Structure »

**Et que** je saisis l'adresse de la structure dans le champ « Adresse de la Structure »

**Et que** je saisis le code postal de la structure dans le champ « Code postal de la Structure »

**Et que** je saisis la ville de la structure dans le champ « Ville de la Structure »

**Et que** je saisis son numéro de téléphone dans le champ « Téléphone »

**Et que** je saisis une description dans le champ « Description »

**Et que** je coche la case « Activer la Structure » ou pas

**Et que** je sélectionne les permissions dans le champ « Permissions »

**Et que** je clique sur le bouton « Enregistrer »

**Alors** j'ai créé une nouvelle structure

**Et** un E-mail est envoyé automatiquement au Gérant de la Franchise et de la nouvelle structure

- **Scénarios de test d'une fonctionnalité de Modification d'une nouvelle structure :**

**Titre de l'us :** US7 – Modification d'une Nouvelle Structure

**En tant que**, Julie

**Je souhaite** Modifier une nouvelle Structure

**Afin de** mettre à jour la Structure

#### **Règles métier**

1. La Structure doit être créée



2. la franchise doit être créée
3. L'administrateur doit être connecter

**Critères d'acceptation**

**Scénario 7 :** Modification d'une Nouvelle Structure

**Étant donné que,** je possède un compte utilisateur pour Julie

**Lorsque,** j'accède à la page « Admin » /admin

**Et que** je clique sur « Modifier une Nouvelle Structure »

**Et que** je sélectionne son compte de connexion « Compte de connexion »

**Et que** je sélectionne la franchise « Sélectionner une franchise »

**Et que** je saisis les modifications de la Structure dans les champs concernés

**Et que** je clique sur le bouton « Enregistrer »

**Alors** j'ai modifié la franchise

**Et** un E-mail est envoyé automatiquement au Gérant de la Franchise et de la nouvelle structure

- Scénarios de test d'une fonctionnalité d'**Affichage des Structure** :

**Titre de l'us :** US8 – Affichage des Structure

**En tant que,** Julie

**Je souhaite** Afficher les Structure actives ou inactives

**Afin de** visualiser les Structure qui sont actives ou inactives

**Règles métier**

1. La Structure doit être créée
2. L'administrateur doit être connecter

**Critères d'acceptation**

**Scénario 8 :** Affichage des Structure

**Étant donné que,** je possède un compte utilisateur pour Julie

**Lorsque,** j'accède à la page « Admin » /admin

**Et que** je coche la case « Afficher les Structure « Active » ou « Inactive » »

**Alors** j'accède aux Structure selon le choix sélectionné

**Et** une page contenant la liste des Structure s'affiche

- Scénarios de test d'une fonctionnalité de recherche dynamique :

<b><u>Titre de l'us :</u></b> US9 – Recherche dynamique
<p><b>En tant que,</b> Julie</p> <p><b>Je souhaite</b> rechercher des franchises ou des structures</p> <p><b>Afin de</b> visualiser la Franchise ou la Structure que je veux voir</p>
<p><b><u>Règles métier</u></b></p> <ol style="list-style-type: none"> <li>1. La Franchise doit être créée</li> <li>2. La Structure doit être créée</li> <li>3. L'administrateur doit être connecter</li> </ol>
<p><b><u>Critères d'acceptation</u></b></p> <p><b>Scénario 9 :</b> Recherche dynamique</p> <p><b>Étant donné que,</b> je possède un compte utilisateur pour Julie</p> <p><b>Lorsque,</b> j'accède à la page « Admin » /admin</p> <p><b>Et que</b> je tape le nom de la franchise ou la structure dans la barre de recherche</p> <p><b>Alors</b> j'accède directement aux informations du compte sélectionner</p> <p><b>Et</b> une page contenant le détail s'affiche</p>

## B. Gestion des Franchises

- Scénarios de test d'une fonctionnalité de droit d'accès d'une franchise :

<b><u>Titre de l'us :</u></b> US11 – Droit d'Accès d'une franchise seulement en lecture
<p><b>En tant que,</b> Rémy - Gérant de la franchise</p> <p><b>Je souhaite</b> accéder à mon compte</p> <p><b>Afin que</b> visualiser mes structures et mes permissions</p>
<p><b><u>Règles métier</u></b></p> <ol style="list-style-type: none"> <li>1. Le Gérant de la Franchise doit posséder un compte de connexion</li> <li>2. La franchise doit être créée</li> </ol>

3. La structure doit être créée
4. Le Gérant doit être connecter

#### **Critères d'acceptation**

**Scénario 11** : Droit d'Accès d'une franchise seulement en lecture

**Étant donné que**, je possède un compte utilisateur pour Rémy

**Lorsque**, j'accède à la page « Se connecter » /login

**Et que** je clique sur « se Connecter »

**Et que**, j'accède à la page « home » /partner

**Alors** j'accède à mon compte

**Et** je peux visualiser le détail de mes structures et de mes permissions

- Scénarios de test d'une fonctionnalité d'une page contact :

**Titre de l'us** : US12 – Page contact

**En tant que**, Rémy - Gérant de la franchise

**Je souhaite** accéder à mon compte

**Afin de** contacter la maison mère

#### **Règles métier**

1. Le Gérant de la Franchise doit posséder un compte de connexion
2. Le Gérant doit être connecter
3. La franchise doit être créée

#### **Critères d'acceptation**

**Scénario 12** : Page de Contact

**Étant donné que**, je possède un compte utilisateur pour Rémy

**Lorsque**, j'accède à la page « Se connecter » /login

**Et que** je clique sur « se Connecter »

**Et que**, j'accède à la page « home » /partner

**Et que** je clique sur « Contactez-nous »

**Et que** je saisis mon Nom dans le champ « Nom »

**Et que** je saisis le sujet de mon message dans le champ « Objet »

**Et que** je saisis mon message dans le champ « Message »

**Et que** je clique sur « Envoyer »

**Alors** mon message est envoyé à l'administrateur  
**Et** la réponse à ce message me sera envoyé sur mon E-mail

- Scénarios de test d'une fonctionnalité de **modifier son compte de connexion** :

**Titre de l'us :** US13 – Modifier son Compte de Connexion

**En tant que**, Rémy - Gérant de la franchise  
**Je souhaite** accéder à mon compte  
**Afin que** changer mes identifiants de connexion

**Règles métier**

1. Le Gérant de la Franchise doit posséder un compte de connexion
2. Le Gérant doit être connecter
3. La franchise doit être créée

**Critères d'acceptation**

**Scénario 13** : Modifier son Compte de Connexion  
**Étant donné que**, je possède un compte utilisateur pour Rémy  
**Lorsque**, j'accède à la page « Se connecter » /login  
**Et que**, j'accède à la page « home » /partner  
**Et que** je clique sur « Modifier mon compte de connexion »  
**Et que** je saisis nom nouveau nom dans le champ « Nom du Gérant »  
**Et que** je saisis mon nouvel E-mail dans le champ « E-mail »  
**Et que** je saisis mon nouveau Mot de Passe dans le champ « Mot de Passe »  
**Et que** je clique sur le bouton « Modifier »  
**Alors** mon compte de connexion  
**Et** je reçois un mail avec mes nouveau identifiant

## C. Gestion des Structures

- Scénarios de test d'une fonctionnalité de **droit d'accès d'une structure** :

**Titre de l'us :** US15 – Droit d'Accès d'une structure seulement en lecture

<p><b>En tant que</b>, Thibo – Gérant de la structure</p> <p><b>Je souhaite</b> accéder à mon compte</p> <p><b>Afin que</b> visualiser les permissions de ma structure</p>
<p><b><u>Règles métier</u></b></p> <ol style="list-style-type: none"> <li>1. Le Gérant de la structure doit posséder un compte de connexion</li> <li>2. La franchise doit être créée</li> <li>3. Le Gérant doit être connecter</li> <li>4. La structure doit être créée</li> </ol>
<p><b><u>Critères d'acceptation</u></b></p> <p><b>Scénario 15</b> : Droit d'Accès d'une structure seulement en lecture</p> <p><b>Étant donné que</b>, je possède un compte utilisateur pour Thibo</p> <p><b>Lorsque</b>, j'accède à la page « Se connecter » /login</p> <p><b>Et que</b> je clique sur « se Connecter »</p> <p><b>Et que</b>, j'accède à la page « home » /structure</p> <p><b>Alors</b> j'accède à mon compte</p> <p><b>Et</b> je peux visualiser le détail de ma structure</p>

- Scénarios de test d'une fonctionnalité d'une page contact :

<p><b><u>Titre de l'us</u></b> : US16 – Page contact</p>
<p><b>En tant que</b>, Thibo – Gérant de la structure</p> <p><b>Je souhaite</b> accéder à mon compte</p> <p><b>Afin de</b> contacter le gérant de la Franchise</p>
<p><b><u>Règles métier</u></b></p> <ol style="list-style-type: none"> <li>1. Le Gérant de la structure doit posséder un compte de connexion</li> <li>2. La franchise doit être créée</li> <li>3. Le Gérant doit être connecter</li> <li>4. La structure doit être créée</li> </ol>
<p><b><u>Critères d'acceptation</u></b></p> <p><b>Scénario 16</b> : Page de Contact</p>

**Étant donné que**, je possède un compte utilisateur pour Thibo

**Lorsque**, j'accède à la page « Se connecter » /login

**Et que** je clique sur « se Connecter »

**Et que**, j'accède à la page « home » /structure

**Et que** je clique sur « Contactez-nous »

**Et que** je saisis mon Nom dans le champ « Nom »

**Et que** je saisis le sujet de mon message dans le champ « Objet »

**Et que** je saisis mon message dans le champ « Message »

**Et que** je clique sur « Envoyer »

**Alors** mon message est envoyé au gérant de la Franchise

**Et** la réponse à ce message me sera envoyé sur mon E-mail

- Scénarios de test d'une fonctionnalité de **modifier son compte de connexion** :

**Titre de l'us :** US17 – Modifier son Compte de Connexion

**En tant que**, Rémy - Gérant de la franchise

**Je souhaite** accéder à mon compte

**Afin que** changer mes identifiants de connexion

#### Règles métier

1. Le Gérant de la Franchise doit posséder un compte de connexion
2. La franchise doit être créée
3. Le Gérant doit être connecter
4. La structure doit être créée

#### Critères d'acceptation

**Scénario 17 :** Modifier son Compte de Connexion

**Étant donné que**, je possède un compte utilisateur pour Rémy

**Lorsque**, j'accède à la page « Se connecter » /login

**Et que** je clique sur « Modifier mon compte de connexion »

**Et que**, j'accède à la page « home » /structure

**Et que** je saisis mon nouveau nom dans le champ « Nom du Gérant »

**Et que** je saisis mon nouvel E-mail dans le champ « E-mail »

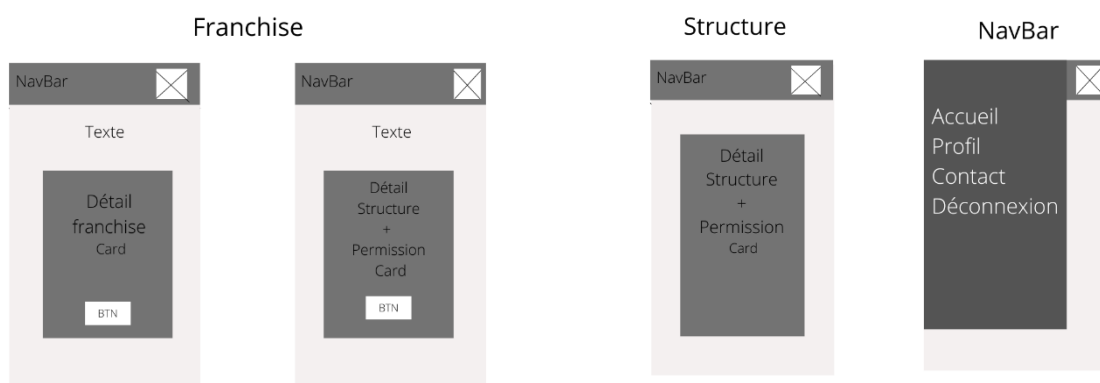
**Et que** je saisis mon nouveau Mot de Passe dans le champ « Mot de Passe »

**Et que** je clique sur le bouton « Modifier »

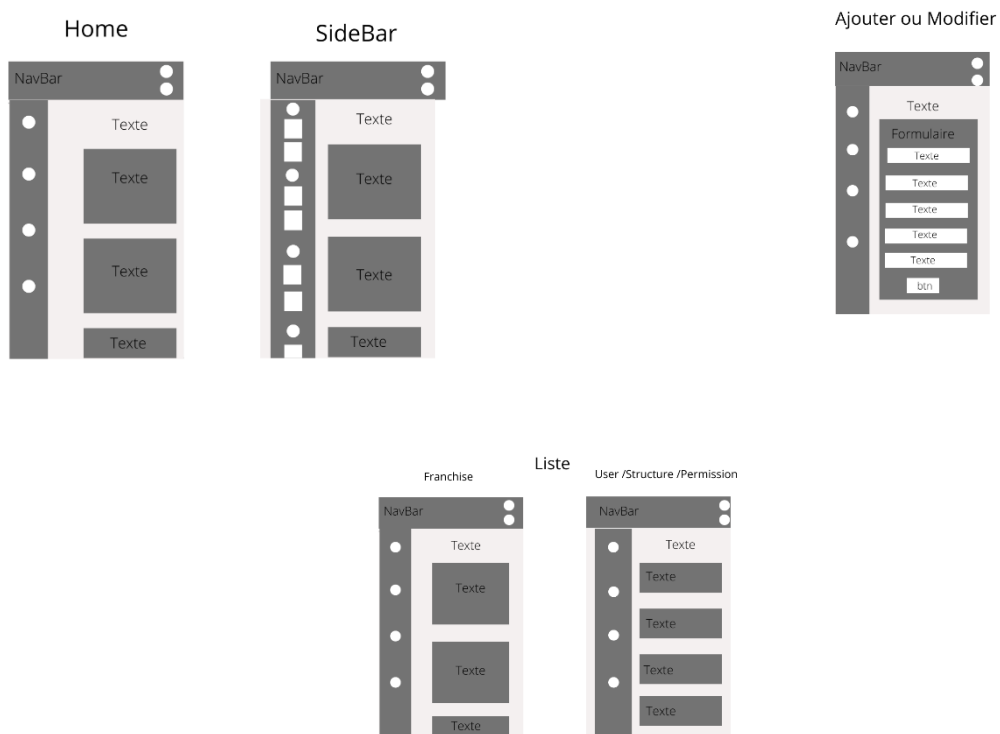
**Alors** mon compte de connexion est modifié

**Et** je reçois un mail avec mes nouveau identifiant

## Wireframes Mobile



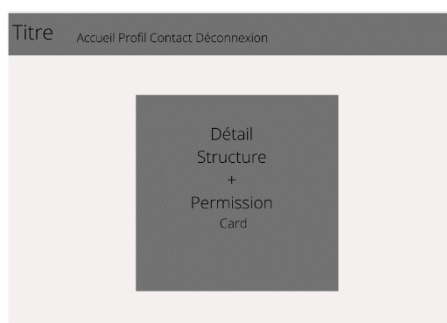
## Admin



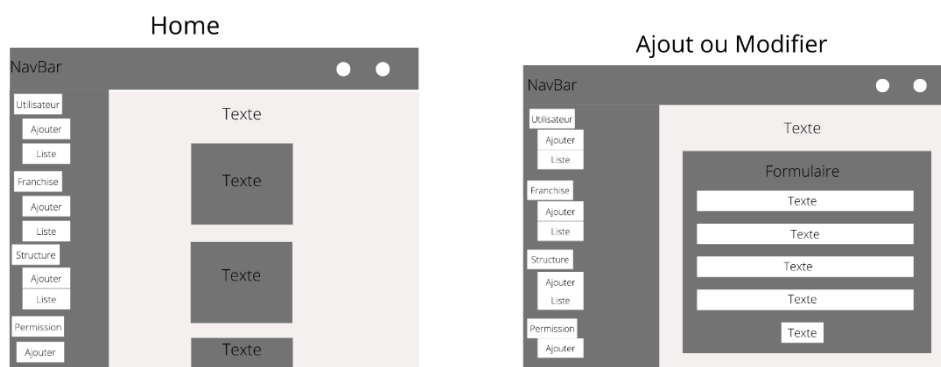
## Wireframe Desktop



## Structure

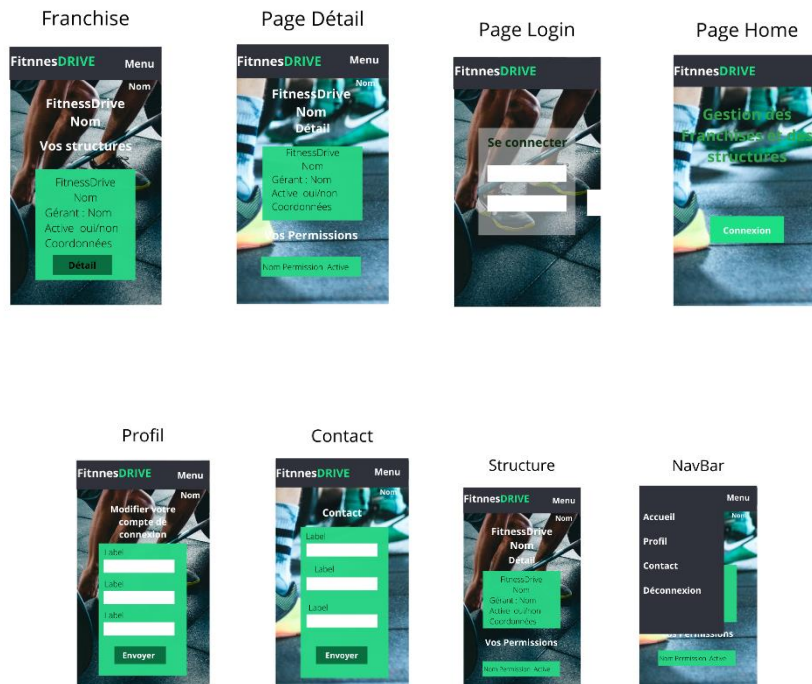


## Admin

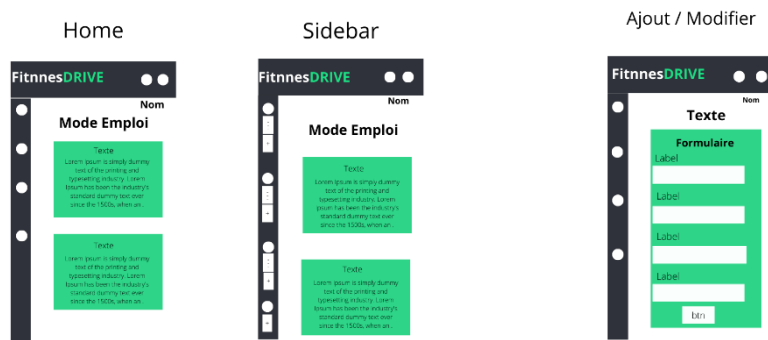




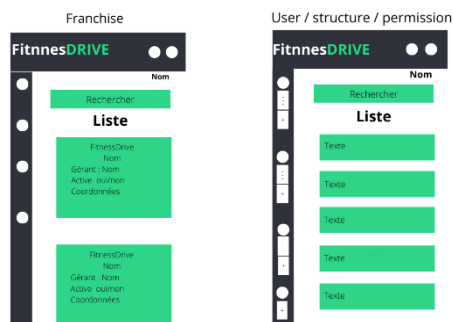
## Mockup Mobile



## Admin



## Liste



## Mockup Desktop

Franchise

Page Détail et Structure

Profil

Contact

Page Login

Page Home

## Admin

Liste

User / Structure/ Permission

Home

Ajout / Modifier