# QUESTIONS ET RÉFLEXIONS

# Évaluation en Cours de Formation

Jérôme OLLIVIER | Décembre 2022

# **Projet Green Fit**

TOUS LES DOCUMENTS SONT EN VERSION PDF SUR MON GOOGLE DRIVE:

https://drive.google.com/drive/folders/1uDTV3T hc-4TPkW tNblif8zWtyavXRa?usp=sharing

### Sommaire

- 1. Spécifications techniques
- 2. Documentation source utilisée
- 3. Organigramme du projet
- 4. Diagramme de cas d'utilisation (UML case)
- 5. Diagramme de séquence : sélection des partenaires actifs
- 6. Diagramme de séquence : modification des droits
- 7. Modèle Conceptuel de Données
- 8. Modèle Logique de Données
- 9. Modèle Physique de Données
- 10. Schéma base de données
- 11. Routes
- 12. Charte graphique
- 13. Personas
- 14. Maquettage
- 15. Choix d'architecture logicielle et de configuration
- 16. Tests

### 1. Spécifications techniques

Voici les différentes technologies que j'ai utilisé pour ce projet :

- Pour la partie front-end :
  - o HTML 5 / Twig
  - o CSS 3
  - Bootstrap 5.1.3 (notamment pour le responsive et les cards) : bibliothèques de composants front-end
  - Javascript
  - o JQuery 3.5.1 : bibliothèque JavaScript
  - TypewriterJS V2 (<a href="https://github.com/tameemsafi/typewriterjs">https://github.com/tameemsafi/typewriterjs</a>): librairie JS pour l'animation de textes
  - o Tippy.js (<a href="https://atomiks.github.io/tippyjs/">https://atomiks.github.io/tippyjs/</a>): librairie JS de Tooltips.
- Pour la partie back-end :
  - o PHP 8.1.6
  - o Extension PHP: PDO
  - Symfony 6.0.10
  - Symfony CLI version 5.4.11

- Base de données : mariaDB 10.4.22
- o IDE utilisé : Visual Studio Code
- Mailtrap : permet de récupérer sur une interface web, le contenu de l'email que vous avez expédié depuis votre site (pour le développement)

### - Pour le serveur de développement :

- Windows 11 (Windows 64)
- o Apache/2.4.53
- o OpenSSL/1.1.1n
- o PHP 8.1.6
- o Extension PHP: PDO
- o Composer 2.3.5
- XAMPP avec MariaDB 10.4.24 et Apache/2.4.53 (Win 64)

### - Pour le serveur de déploiement :

- o PAAS Heroku : plateforme en tant que service basée sur le cloud (PaaS) permettant de déployer ses applications sans se préoccuper de la gestion d'un serveur.
- o Apache/2.4.54
- o PHP 8.1.9
- o Composer 2.3.10
- o Extension PHP: PDO
- o Nginx 1.22.0
- o Doctrine ORM
- WorkBench d'Oracle pour le transfert vers la base de données en ligne grâce au dump de la base locale de XAMPP
- ClearDB MySQL : add-on de Heroku, base de données MySQL et permet également des backups de celle-ci avec une rétention de 5 jours

### - Outils de conception du maquettage :

- o Excalidraw.com: création du zoning pour la version Smartphone et la version Desktop
- o Mogups: wireframes de l'application en versions smartphones et desktop
- o Adobe XD : mockups de l'application en versions smartphones et desktop

### Outils de design pour le front-end :

- Flow Mapp : création des personas
- o Images gratuites et libres de droits : Unsplash.com
- o Freelogodesign.org: image du logo
- Affinity Designer: personnalisation du logo
- o TinyPNG: compression des images
- o Font Awesome : kit d'icônes
- o Realfavicongenerator.net : création du favicon
- Responsively : Tests web responsive :

### - Versionning: Git (local) et GitHub (distant)

- <u>Planification</u>: Trello

(https://trello.com/invite/b/nqCLMppj/2bf93bee36b02bd4789cbfddc24e8649/ecf

- Outils de conception et de développement :
  - o VSCode 1.71 : éditeur de code
  - Windows PowerShell 5.1.22000.832 : pour sa partie interface en ligne de commandes

### 2. Documentation source utilisée

https://www.php.net/docs.php

https://getbootstrap.com/docs/5.1/getting-started/introduction/

https://sql.sh/sgbd/mysql

https://sql.sh/sgbd/mariadb

https://symfony.com/doc/current/index.html

https://symfony.com/doc/current/mailer.html

https://developers.sendinblue.com/docs

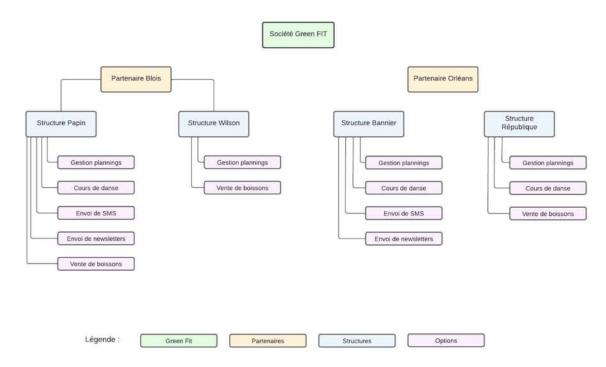
https://www.bcrypt.fr/

https://stackoverflow.com/

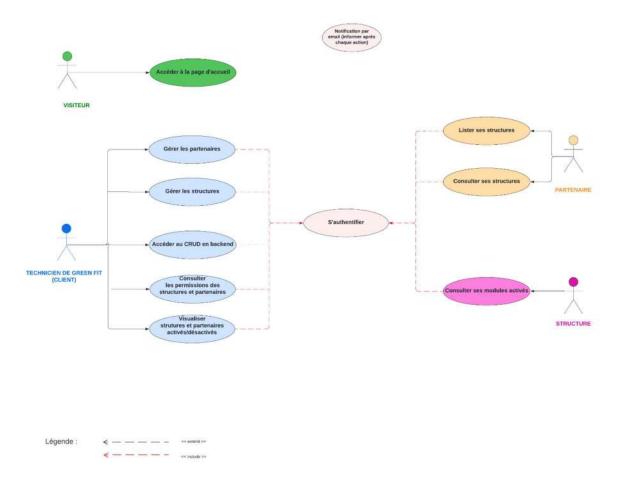
# 3. Organigramme du projet Green Fit

### DIAGRAMME DE LA SOCIÉTÉ GREEN FIT





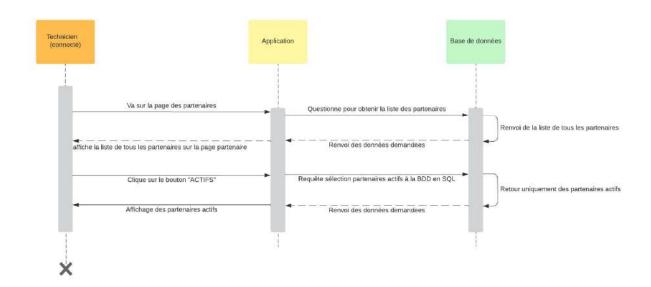
# 4. Diagramme de cas d'utilisation



5. Diagramme de séquence : sélection des partenaires actifs

#### Diagramme de séquence: sélection des partenaires actifs

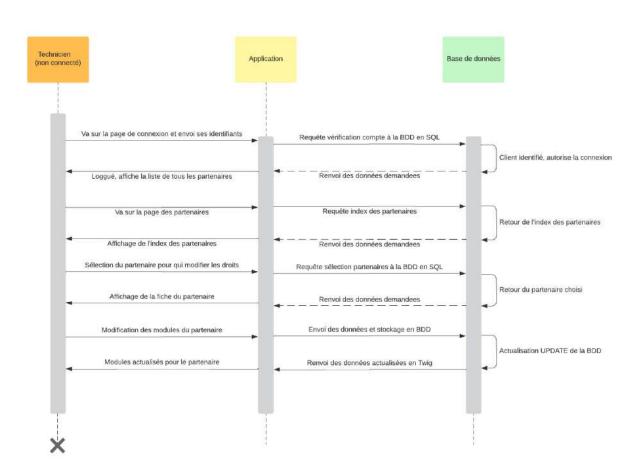
(depuis le Dashboard du technicien)



## 6. Diagramme de séquence : modification des droits

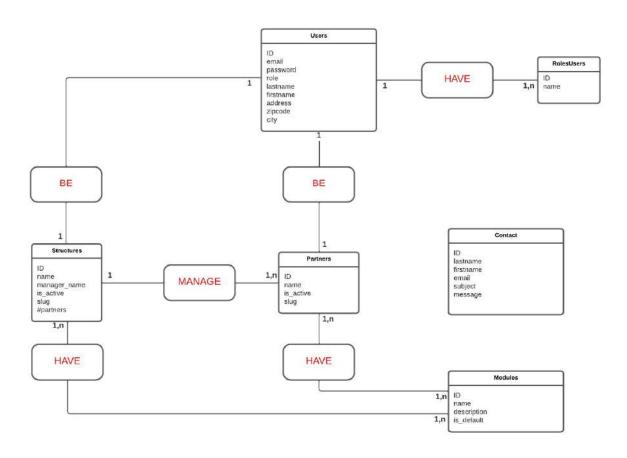
#### Diagramme de séquence: modification des droits

(depuis la page de connexion)



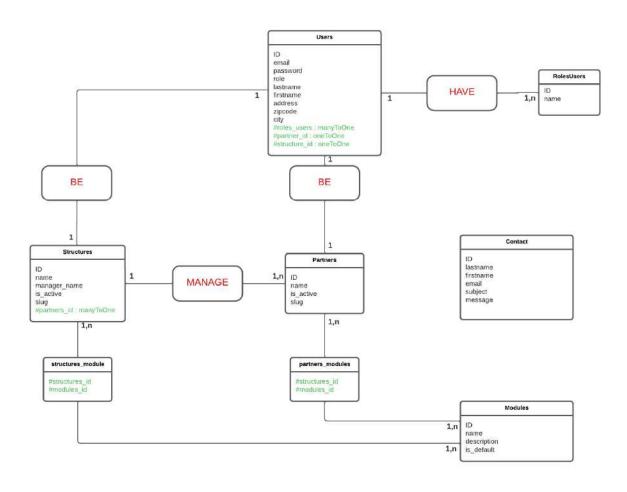
# 7. Modèle Conceptuel de Données

### 1 - MCD - GREENFIT



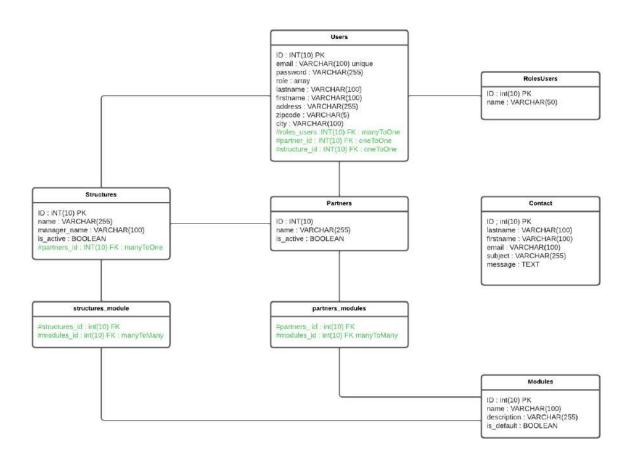
# 8. Modèle Logique de Données

### 2 - MLD - GREENFIT

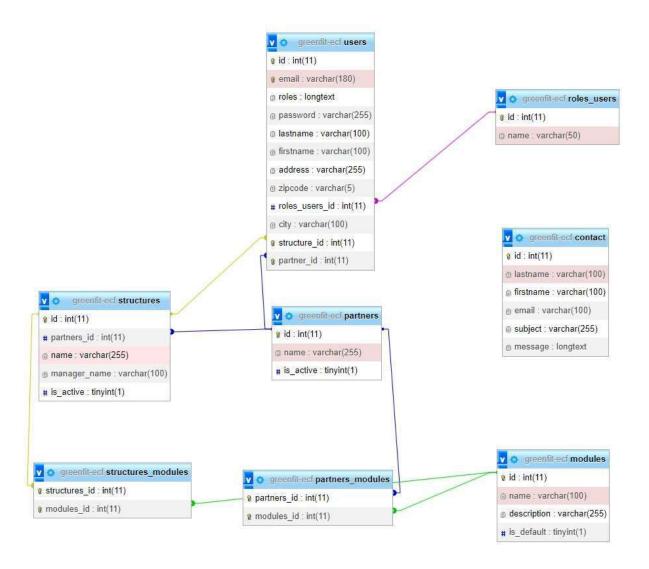


# 9. Modèle Physique de Données

#### 3 - MPD - GREENFIT



### 10. Schéma base de données



### 11. Routes

Page	Public	ÉQUIPETECHNIQUE	PARTENAIRE	STRUCTURE	Path	liom	Várif
		ROU	TES DE L'APPUCATION	GREENFIT			
ccueil	OUI	oul	OUI	OU1	1	app_admin	ak
ngin .	OUI	OUI	OUI	OUT	/login	app_(ogin	qk
ontact	OUI	OUI	OUI	OUT	/contact	app_contact	o
lentions légales	OUI	OUI	OUI	OUT	/mentions-legalex	app_legal	d
Ficher la liste de tous les techniciens	NON	OUI	NON	NON	/users/	app users index	a
réer un nouveau technicien	NON	OUI	NON	NON	/users/new	app_users_new	
étails d'un technicien	NON	OUI	NON	NON	/users/(id)	app_users_new	0
lodifier un technicien	NON	OUI	NON	NON	/users/(idi/edit	app users edit	
facer un technicien	NON	OUI	NON	NON	/users/(id)/delete	app_users_eur	
MANAGEMENT AND	1100000	Territor .	1100	3000000	alimitical description		18
fficher la liste de tous les partenaires	NON	OUI	NON	NON	/partners/	app_partners_index	9
réer un nouveau partenaire	NON	OUI	NON	NON	/partners/new	app_partners_new	
étalis d'un partenaire	NON	OUI	NON	NON	/partners/{id}	app_partners_show	
todifier un partenaire	NON	OUI	NON	NON	/partners/{id}/edit	app_partners_edit	0
ffacer un partenaire	NON	OUI	NON	NON	/partners/{id}/delete	app_partners_delete	
fficher toutes les structures d'un partenaire	NON	Oui	OUI	NON	/partners/structures/{id}	app_partners_structures	c
fficher la liste de toutes l'es structures	NON	OUI	NON	NON	/structures/	app_structures_index	
réer une nouvelle structure	NON	OUI	NON	NON	/structures/new	app structures new	
étails d'une structure	NON	OUI	NON	NON	/structures/fid1	app structures show	0
todifier une structure	NON	OUI	NON	NON	/structures/{id}/edit	app_structures_edit	
ffacer une structure	NON	OUI	NON	NON	/structures/{id}/delete	app structures delete	
fficher les modules d'une structure	NON	OUI	OUI	OUT	/structures/modules/{id}	app_structures_modules	
			100.000	100000			
fficher la liste de tous les modules	NON	OUI	NON	NON	/modules/	app_modules_index	9
réer un nouveau module	NON	OUI	NON	NON	/modules/new	app_modules_new	
étails d'un module	NON	OUI	NON	NON	/modules/{id}	app_modules_show	
1odifier un module	NON	OUI	NON	NON	/madules/(id}/edit	app_modules_edit	3
ffacer un module	NON	OUI	NON	NON	/madules/fid3/delete	app modules delete	

## 12. Charte graphique



# Fichier charte graphique

### Mise en page

Les dessins ont été créés dans les largeurs suivantes :

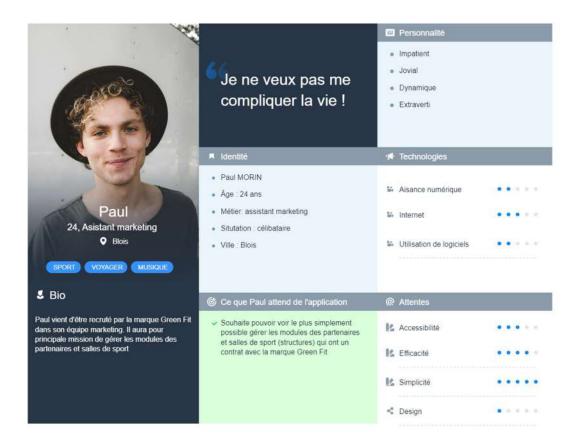
Mobile : 386 px Desktop : 1440 px

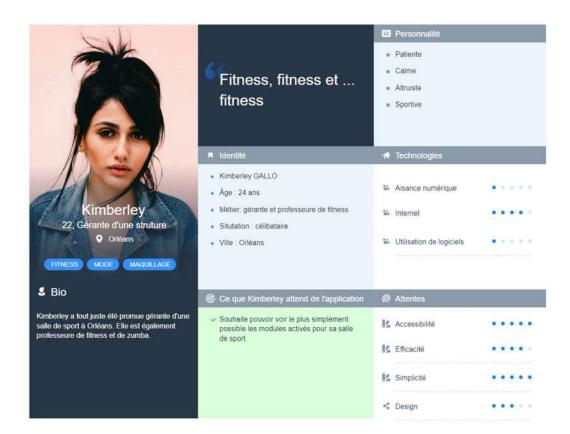


### Couleurs



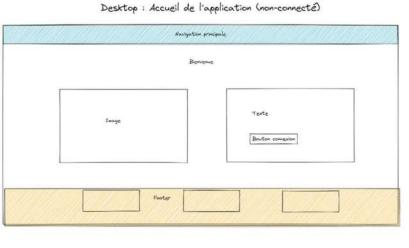
### 13. Personas

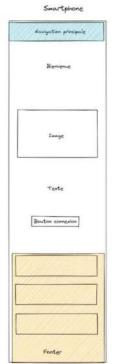




## 14. Maquettage

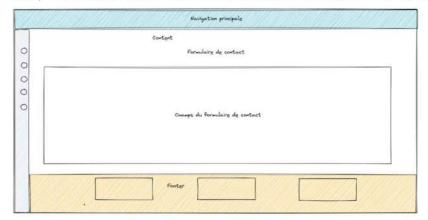
### 14.1. Zonings





### Zoning page d'accueil de l'application(public)

Desktop: Formulaire de contact (accessible aux utilisateurs connectées et non-connectés

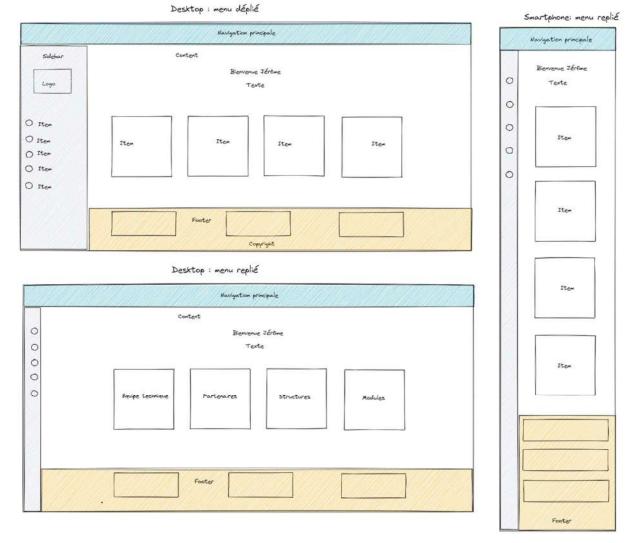






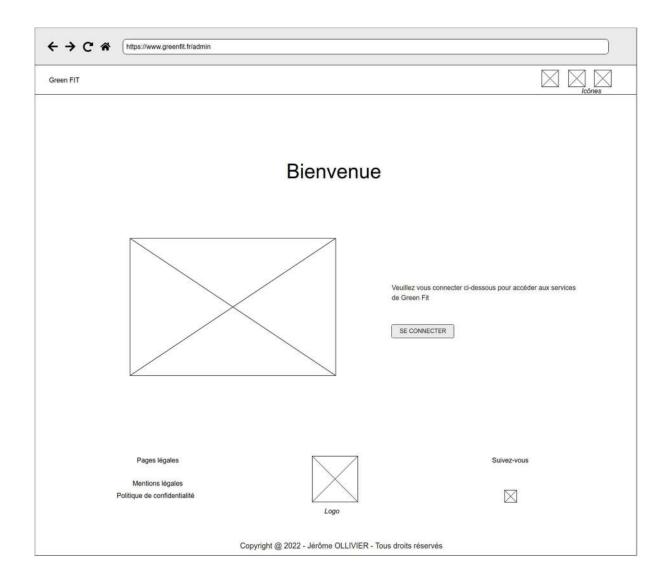
Zoning page de contact(public)

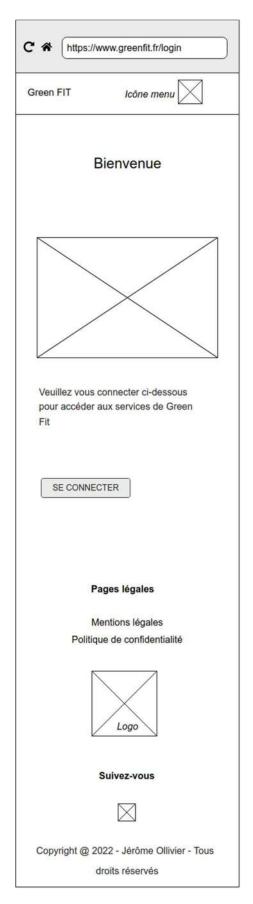
# Page d'acceuil d'un technicien de Greeenfit



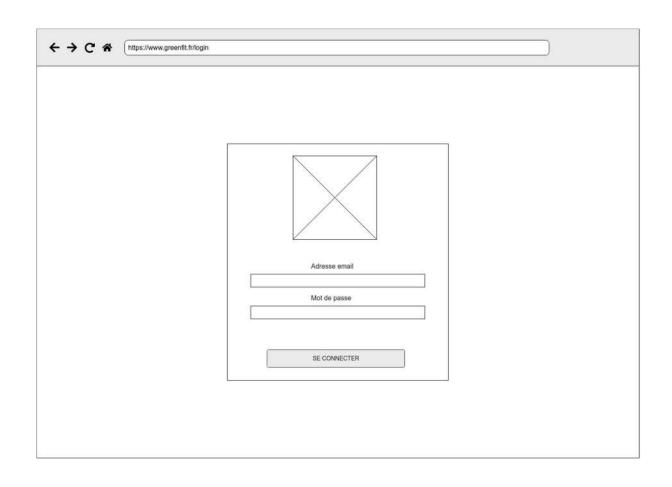
Zoning page d'accueil des techniciens (connectés)

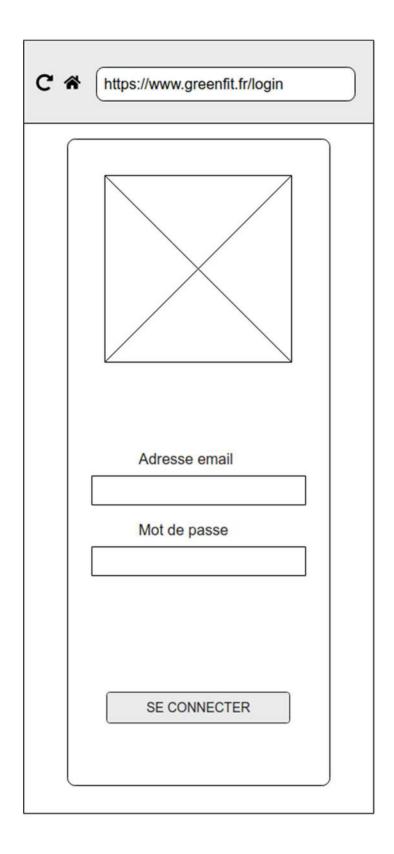
### 14.2. Wireframes





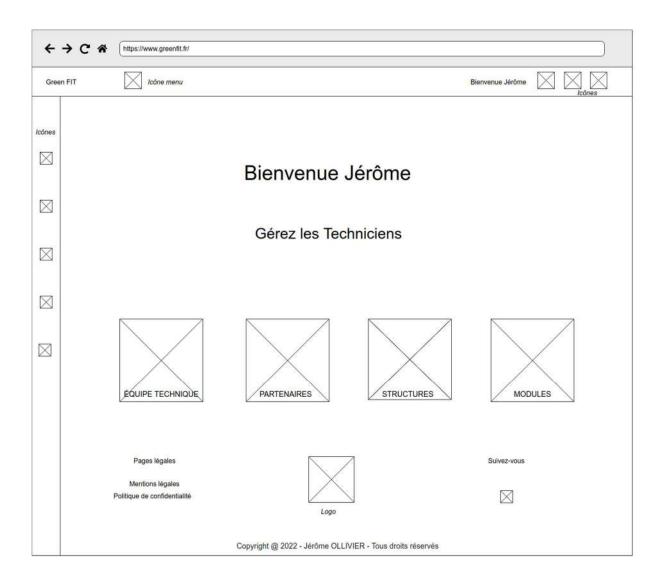
Wireframe Page de login, version Desktop (public)

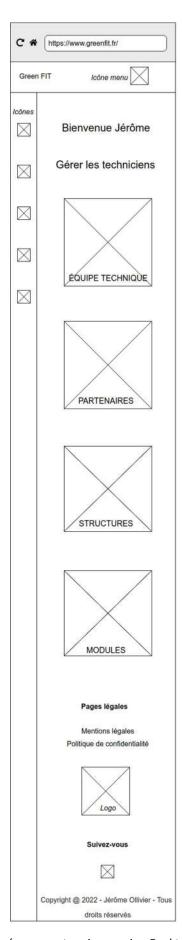




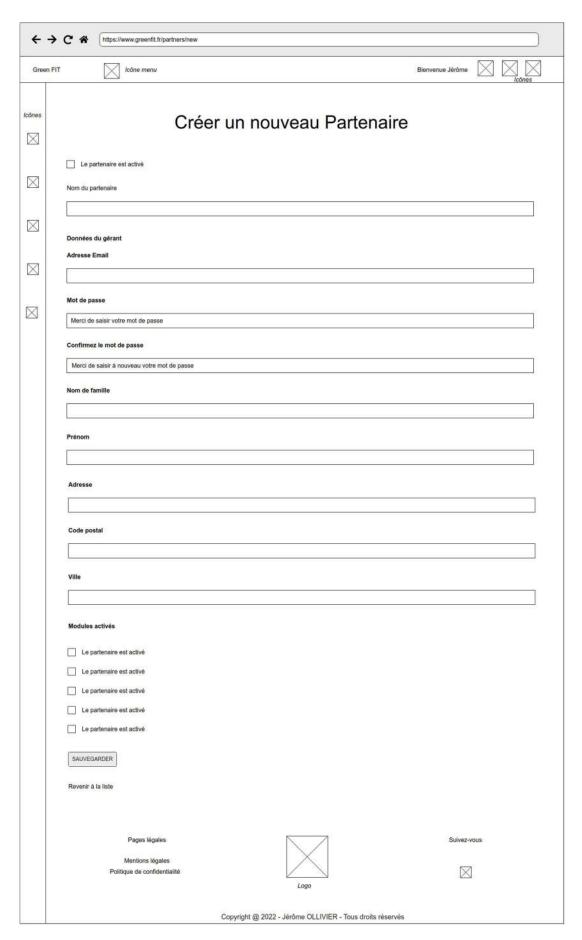


C ♣ (https://www.greenfit.fr/contact						
Green FIT						
Formulaire de contact						
Prénom						
Nom de famille						
Non de familie						
Adresse e-mail						
Objet de la demande						
Message						
Envoyer mon formulaire						
Pages légales						
Mentions légales Politique de confidentialité						
Logo						
Suivez-vous						
$\boxtimes$						
Copyright @ 2022 - Jérôme Ollivier - Tous droits réservés						



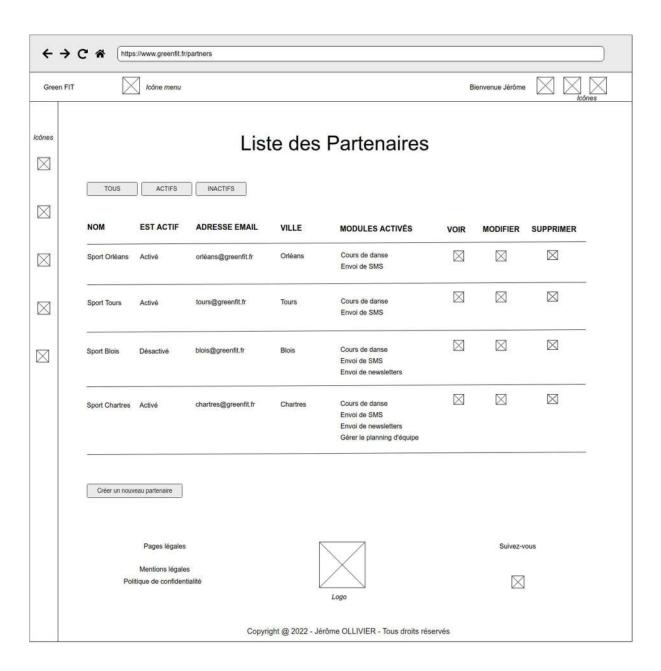


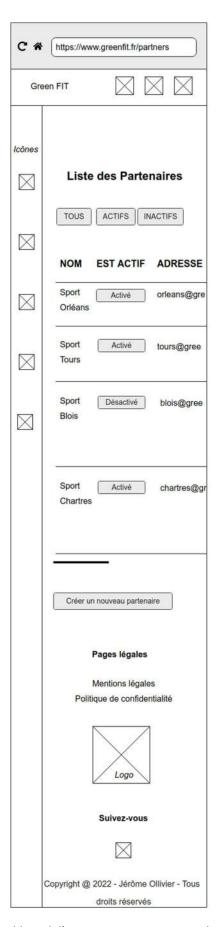
Wireframe créer un partenaire, version Desktop (connecté)



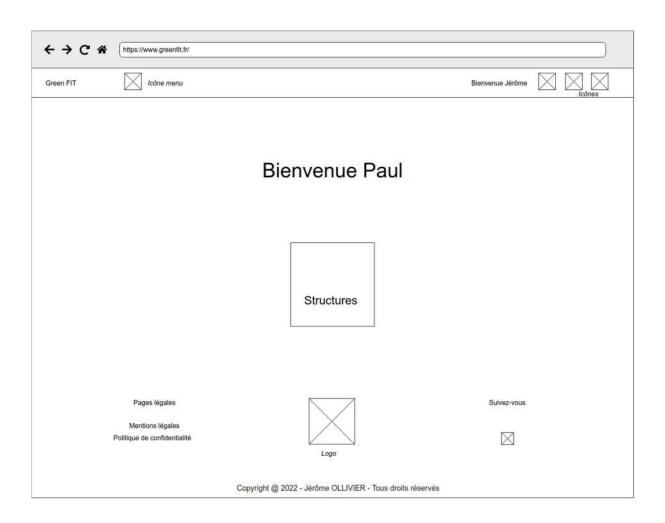


Wireframe lister les partenaires, version Desktop (connecté)



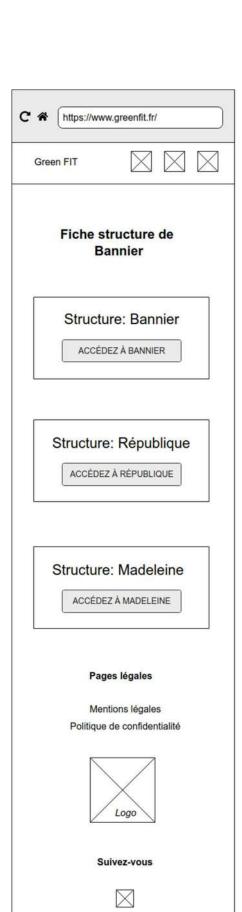


Wireframe Dashboard d'un partenaire, version Desktop (connecté)



C ♠ (https://www.greenfit.fr/					
Green FIT					
Bienvenue Paul					
Structures					
Pages légales					
Mentions légales					
Politique de confidentialité  Logo					
Suivez-vous					
Copyright @ 2022 - Jérôme Ollivier - Tous droits réservés					



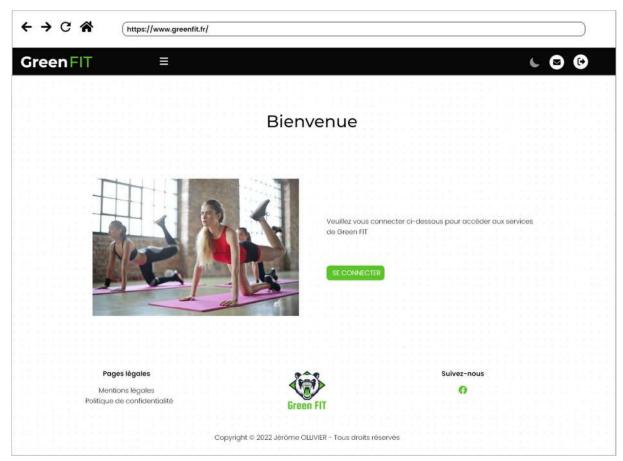


Copyright @ 2022 - Jérôme Ollivier - Tous droits réservés

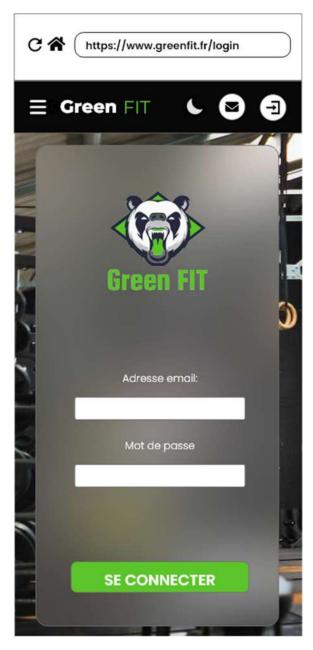
← → C ☆ https://www.greenfit.fr/structure/1						
Green FiT Icône menu		Bienvenue Jérôme loônes				
Ficl	he Structure de BANNIER	, ,				
EST ACTIVÉE	Oui					
NOM DU GÉRANT	Dubet					
PRÉNOM DU GÉRANT	Chantale					
EMAIL	orleans-bannier@greenfit.fr					
ADRESSE	52, rue Bannier					
CODE POSTAL	45000					
VILLE	Orléans					
MODULES ACTIVÉS	Cours de danse Envoi de SMS Envoi de newwsletters					
Pages légales Mentions légales Politique de confidentialité	Logo	Suivez-vous				
Copyright @ 2022 - Jérôme OLLIVIER - Tous droits réservés						



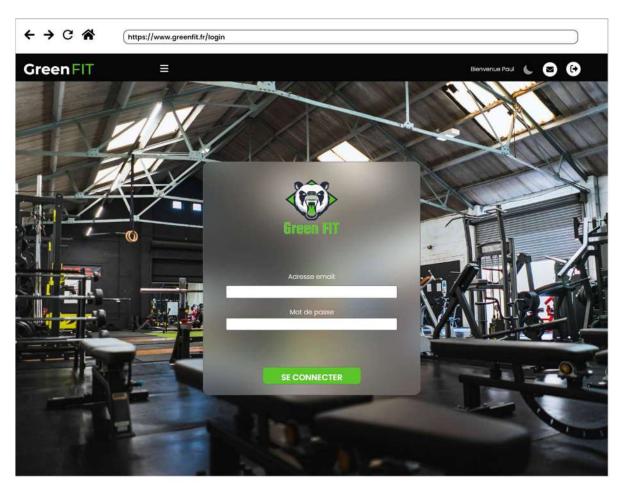
### 14.3. Mockups



Annexe XXX : Mockup version Desktop de la page d'accueil de l'application (Public)



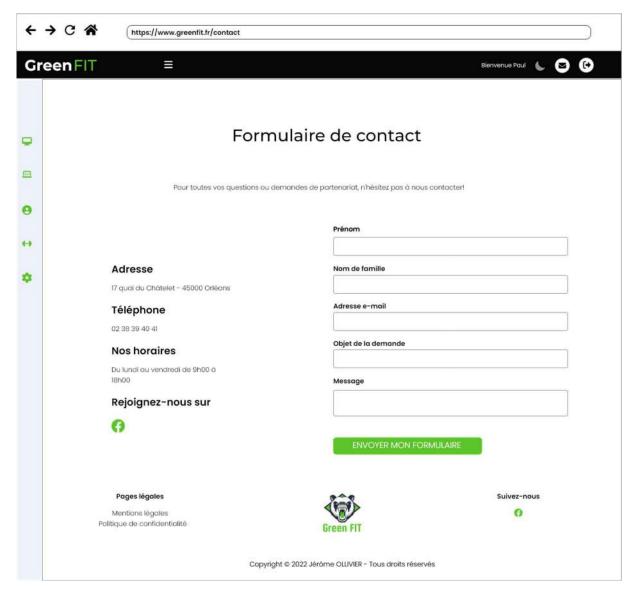
Annexe XXX : Mockup version Smartphone de la page de login (Public)



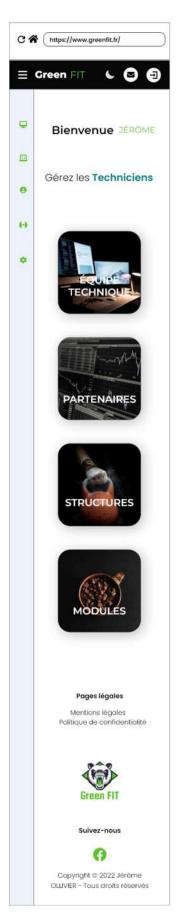
Annexe XXX : Mockup version Desktop de la page de login (Public)



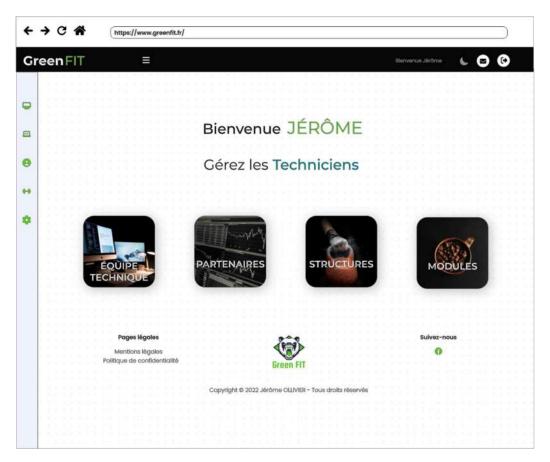
Annexe XXX : Mockup version Smartphone de la page de Contact (Public



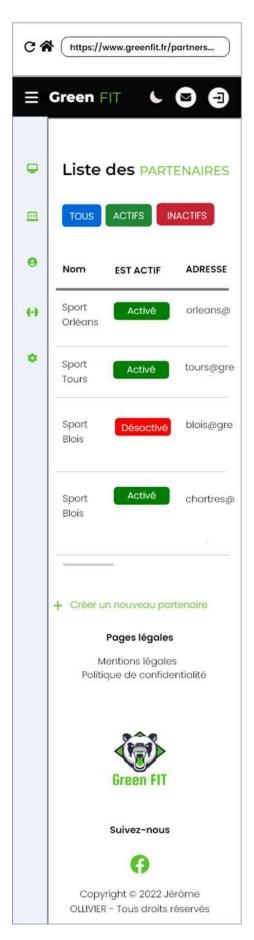
Annexe XXX: Mockup version Desktop de la page de Contact (Public)



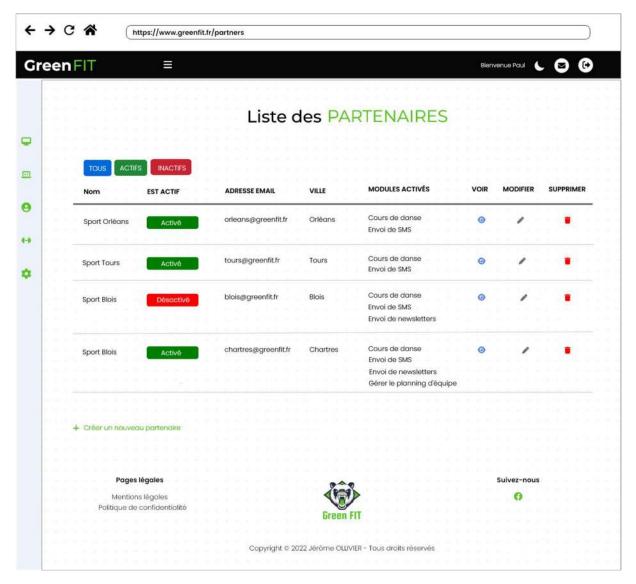
 $Annexe~\it XXX: Mockup~version~Smartphone~du~Dashboard~d'un~technicien~(technicien,~menu~repli$ 



Annexe XXX : Mockup version Desktop du Dashboard d'un technicien (technicien, menu replié)



Annexe XXX : Mockup version Smartphone du Dashboard de l'index des partenaires (technicien



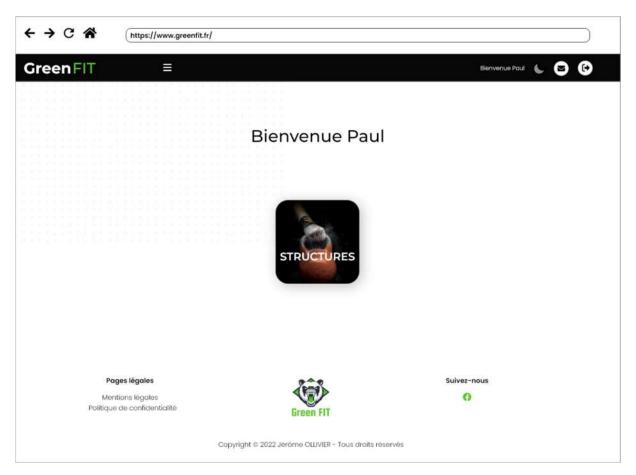
Annexe XXX : Mockup version Desktop du Dashboard de l'index des partenaires (technicien)



Annexe XXX : Mockup Smartphone de la page de création d'un nouveau partenaire (technicien)

≡				Bienvenue Paul	. (
	Créer u	un nouveau F	Partenaire		
Le partenaire est activé					
Nom du partenaire					
DONNÉES DU GÉRANT			2 - 3	1 3 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	
Adresse Email					
Mot de passe					
Merci de saisir votre mot de pas	ise	13/11/11/11/12	1311221	1211	
Mot de passe				31 14 30 30 H 10 10 10 10	
Merci de soisir à nouveau votre	mot de passe				-
Nom de famille	19231111			111111111111111111111111111111111111111	1 7
×× (					
Prénom					
Adresse					
N = 4000					
Code postal					
/ille					
Modules activés pour le partenai	ire				
Cours de danse					
☐ Envoi de SMS					
☐ Envoi de newsletters					
Gérer le planning d'équipe					
☐ Vendre des boissons					
Contractored					
Sauvegarder					
tevenir à la liste					
Pages légales		200		Suivez-n	ous
Mentions légales				0	

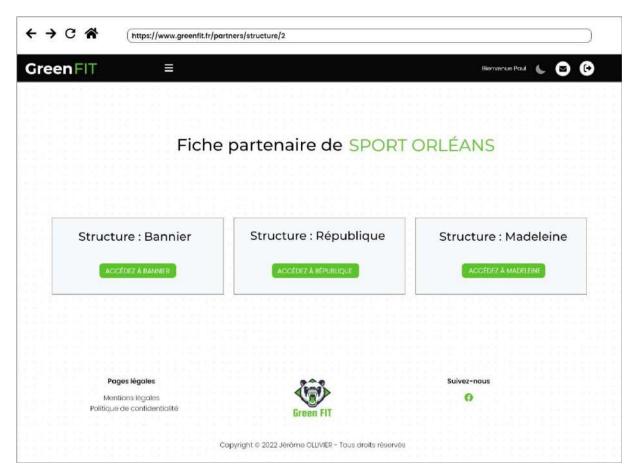




Annexe XXX : Mockup version Desktop du Dashboard d'un partenaire (partenaire)

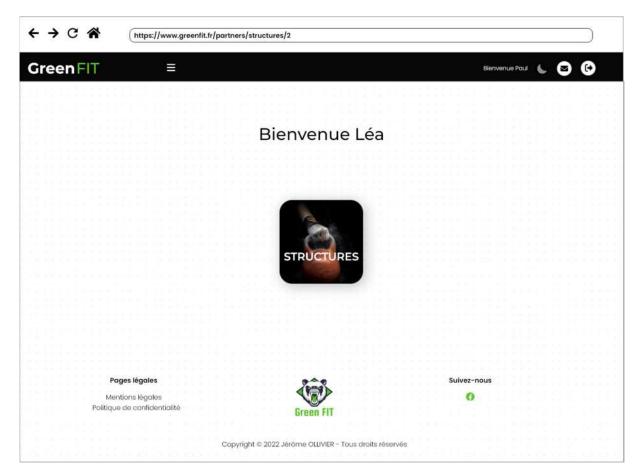


Annexe XXX : Mockup version Smartphone de la liste des structures d'un partenaire (partenaire)



Annexe XXX : Mockup version Desktop de la liste des structures d'un partenaire (partenaire)

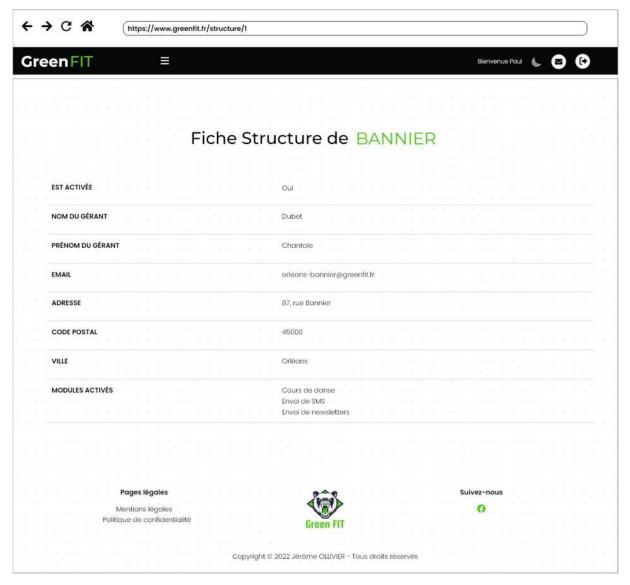




Annexe XXX : Mockup version Desktop du Dashboard d'une structure (structure)



Annexe XXX : Mockup version Smartphone d'une fiche de structure (structure, partenaire et technicien)



Annexe XXX : Mockup version Desktop d'une fiche de structure (structure, partenaire et technicien)

# 15. Choix d'architecture logicielle et de configuration

## 15.1. Programmation Orientée Objet

La programmation orientée objet, que j'ai utilisé tout au long de mon projet, est un paradigme de programmation qui permet de développer des applications complexes en maintenant un code plus clair et maniable. La POO regroupe certaines informations liées entre elles conceptuellement dans des « objets ». Un objet représente un concept, une idée, ou toute entité du mon physique comme dans notre application une structure ou un partenaire.

Prenons un exemple tiré de mon projet, l'entité contact :

the L'entité Contact est ici implémentée par une classe (d'objet). Elle possède six attributs : \$id, \$lastname, \$firstname, \$email, \$subject et \$message. Ces attributs sont des variables (de type "integer" et "string). Enfin, les "getters" et "setters" comme getId(), getLastname() et setLastname() sont des méthodes de classe (permettant d'invoquer l'attribut auquel ils sont associés : pour obtenir la valeur de l'attribut \$lastname d'un objet de la classe Contact, le code est : \$objet->getLastname()).

### → Annexe : la POO

(J'y ai volontairement supprimer des getters et setters pour la lisibilité du code).

### 1.1.1. L'architecture MVC

L'architecture "Modèle-Vue-Contrôleur", dite MVC, fait partie des briques de base sur lesquelles j'ai construit tout mon projet. Elle consiste en une factorisation du code bien particulière, reposant sur trois composants principaux :

- Le contrôleur, qui contient la logique des actions à exécuter suivant la requête de l'utilisateur (le code qui doit être exécuté),
- La vue, qui contient le code de la partie visible par l'utilisateur du projet (HTML, Twig dans ce projet)
- Et enfin le modèle, qui gère l'accès aux données (le plus souvent dans une base de données)

Nous ferons aussi appel de façon implicite à un quatrième élément parfois ajouté au modèle MVC : le routeur.

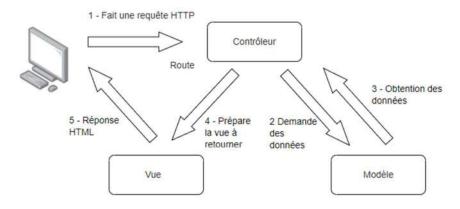


Schéma du modèle MVC de Symfony

### 15.2. Symfony

J'ai choisi d'utiliser le framework PHP Symfony afin de gagner du temps dans le développement.

Les intérêts d'un tel framework sont multiples : une architecture MVC, la sécurité est un des piliers de Symfony, les bonnes pratiques sont largement favorisées, l'organisation en services est naturelle, de nombreux bundles sont disponibles,

De plus, on peut de plus analyser finement les opérations se déroulant sur l'application, grâce notamment à la "debug bar" dans l'environnement de développement (qui permet par exemple d'avoir le détail sur les éventuels codes d'erreur HTTP, l'utilisateur connecté, les logs...).

### 15.3. Twig

J'ai utilisé le moteur de templates Twig tout au long de mon projet. C'est un outil qui permet de simplifier au maximum la partie associées aux vues, afin de séparer au maximum la partie interface graphique de la partie traitement.

Par exemple, le code suivant en HTM et PHP permet d'afficher le contenu échappé d'un attribut \$name d'un objet de la classe Module.

```
<?php echo htmlspecialchars($module ->getName(), ENT_QUOTES , 'UTF8') ?>
```

En Twig, cela se traduit par ce code:

### 15.4. Maria BD

Maria DB est le système de gestion de bases de données (SGBD) que j'ai utilisé pour ce projet, car il est open source et hautement compatible avec MySQL. J'ai principalement fait ce choix car c'est le SGBD avec lequel j'ai le plus d'expérience. En phase de développement, j'ai utilisé PHPMyAdmin 5.2.0 et la version de Maria DB 10.4.24.

Ma base de données de développement est uniquement accessible en local.

#### 15.5. Sécurité

Je reviens sur toutes les mesures mises en œuvre pour la sécurité dans le document intitulé « Sécurité et bonnes pratiques de l'application Green Fit » disponible sur mon Google Drive et sur GitHub dans le dossier « Annexes ». Voici néanmoins quelques éléments essentiels que nous pouvons citer dans cette partie :

- Le back office de mon application est protégée par l'authentification. Il y a trois types d'accès : pour les techniciens, pour les partenaires et enfin pour les structures.
- Dans les formulaires du site accessibles aux utilisateurs connectés, les champs sont vérifiés (adresse email valide, code postal ne contenant que des chiffres...).

# 16. Tests unitaires et fonctionnels

Voici quelques exemples de tests unitaires et fonctionnels exécuté durant le développement.

## Test n°1:

J'en profite à nouveau pour faire un test unitaire pour le contrôleur Login pour vérifier que la page de Login (/login) renvoie bien une réponse HTTP 200. La page est bien accessible.

```
1 <?php
 2
 3
    namespace App\Tests\Controller;
 4
 5  use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
 6
 7 class UrlLoginTest extends WebTestCase
 8 {
9
        public function testLogin()
10
11
        $client = static::createClient();
12
         $client->request('GET', '/login');
13
14
        $this->assertResponseIsSuccessful();
15
16
```

Fichier de test unitaire tests/UrlLoginTest.php

Résultat : le test fonctionne.

```
PS C:\Users\spire\Desktop\Versionning ECF\160922-08h45\xampp\apps\greenfit-ecf> symfony php bin/phpunit --testdox PHPUnit 9.5.23 #StandWithUkraine

Testing Url Login (App\Tests\Controller\UrlLogin) ✓ Login

Time: 00:00.207, Memory: 26.00 MB

OK (1 test, 1 assertion)
```

### Test n°2:

Exemple d'un test fonctionnel : tester la bonne authentification à l'application avec un compte de technicien.

```
1 <?php
2
3 namespace App\Tests\Entity;
4
 5 use App\Entity\Users;
6 use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;
 7 use Symfony\Component\Validator\ConstraintViolationList;
8 use Symfony\Component\Validator\Validator\ValidatorInterface;
9
10 class UserEntityTest extends KernelTestCase
11 ▼ {
12
13
       private const VALID_EMAIL_VALUE = 'admin@greenfit.fr';
14
        private const VALID_PASSWORD_VALUE = 'paswwordAdmin22';
15
        public ValidatorInterface $validator;
16
17
        protected function setUp(): void
18 ▼
        {
19
            self::bootKernel();
20
            $container = static::getContainer();
21
           $this->validator = $container->get('validator');
22
        }
23
24
        public function testUserEntityIsValid(): void
25 ▼
26
            $user = new Users();
27
            $user
28
               ->setEmail(self::VALID_EMAIL_VALUE)
29
               ->setPassword(self::VALID_PASSWORD_VALUE);
30
            $this->getValidatorErrors($user, 0);
31
```

### Résultat : le test fonctionne

```
PS C:\Users\spire\Desktop\Versionning ECF\160922-08h45\xampp\apps\greenfit-ecf> symfony php bin/phpunit --testdox PHPUnit 9.5.23 #StandWithUkraine

Testing User Entity (App\Tests\Entity\UserEntity)

✓ User entity is valid

✓ User entity is not valid

Time: 00:00.204, Memory: 26.00 MB

OK (2 tests, 2 assertions)
```

# Test n°3:

Exemple de test unitaire pour voir si on peut se connecter à la page /users sans être connecté :

```
1
     <?php
2
3
     namespace App\Tests\Controller;
4
5
     use App\Repository\UsersRepository;
     use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
6
7
     use Symfony\Component\HttpFoundation\Response;
8
9
     class UsersPageTest extends WebTestCase
10
11
         public function testReservationPageIsRestricted()
12
13
            $client = static::createClient();
14
            $client->request('GET', '/users');
15
            $this->assertResponseRedirects();
16
17
```

Code de tests/UsersPageTest

Résultat du test : impossible de se connecter à cette page, sans avoir les autorisations pour s'y connecter. Ici il faudrait avoir le rôle ROLE ADMIN pour cela.

## Test n°4:

J'ai ajouté dans les fichiers templates/partners/index.html.twig et templates/partners/index.html.twig, trois boutons avec des id différents.

Puis dans ces fichiers j'ai ajouté une condition ternaire. Dans le fichier templates/partners/index.html.twig, si le partenaire est actif, il aura la classe « actif » et s'il n'est pas activé, il aura la classe « inactif ». J'ai effectué la même chose dans le fichier templates/structures/index.html.twig

Enfin, j'ai ajouté un script dans mon fichier js/main.js. Ce script sera le même pour les pages d'index des partenaires et des structures.

Je souhaite qu'en cliquant sur le bouton :

- « TOUS » : tous les partenaires soient affichés
- « ACTIFS » : seuls les partenaires actifs soient affichés
- « INACTIFS » : seuls les partenaires inactifs soient affichés.

Je ferai la même chose pour les structures.

→ Annexe XXX : Filtre actifs/désactivés des partenaires et des structures

```
const elementsInactifs = document.getElementsByClassName('inactif');
const elementsActifs = document.getElementsByClassName('actif');
const elementsAll = [...elementsInactifs,...elementsActifs];
```

Puis j'ai créé 3 méthodes addEventListener(), un par bouton qui appelleront les fonctions showActifs(), showInactifs() et showAll() au click sur le bouton.

```
53 // sur le bouton actif cela nous montre les actifs
     const boutonActif = document.getElementById('actif');
    if(boutonActif){
55
    boutonActif.addEventListener("click", showActifs);
56
57 }
58
59 // sur le bouton inactif cela nous montre les inactifs
    const boutonInactif = document.getElementById('inactif');
60
61
    if(boutonInactif) {
     boutonInactif.addEventListener("click", showInactifs);
62
63
64
65
66 // sur le bouton tout cela nous montre tous les partenaires
67 const boutonAll = document.getElementById('all');
68 if(boutonAll) {
69
     boutonAll.addEventListener("click", showAll);
70
```

# Enfin, j'ai créé 3 fonctions :

- showActifs(): parcourt le tableau des éléments avec deux boucles for, et ajoutera la classe d-none (display none de Bootstrap) aux éléments inactifs, ils disparaîtront de la liste, et supprimera la classe d-none pour les éléments actifs, ils seront affichés dans la liste
- showInactifs(): parcourt le tableau des éléments avec deux boucles for, et supprimera la classe d-none (display none de Bootstrap) aux éléments inactifs, ils seront affichés dans la liste, et ajoutera la classe d-none pour les éléments actifs, ils disparaîtront de la liste
- showAll(): parcourt le tableau des éléments avec deux boucles for, et supprimera la classe dnone, pour afficher tous les éléments actifs et inactifs.

```
17 v const showActifs = ()=> {
18
19
         // Iterate through the retrieved elements and add the necessary class names.
20
          for(var i = 0; i < elementsInactifs.length; i++)</pre>
21
22,
           elementsInactifs[i].classList.add('d-none');
23
24
         for(var i = 0; i < elementsActifs.length; i++)</pre>
25
         {
26
           elementsActifs[i].classList.remove('d-none');
27
28
29
30 v const showInactifs= ()=> {
31
32
         // Iterate through the retrieved elements and add the necessary class names.
33 ~
         for(var i = 0; i < elementsInactifs.length; i++)</pre>
34
35
           elementsInactifs[i].classList.remove('d-none');
36
          for(var i = 0; i < elementsActifs.length; i++)</pre>
37
38
39
40
           elementsActifs[i].classList.add('d-none');
41
42
     const showAll= ()=> {
43 ~
44
         // Iterate through the retrieved elements and add the necessary class names.
45
46
47
         for(var i = 0; i < elementsAll.length; i++)</pre>
48
           elementsAll[i].classList.remove('d-none');
49
50
51
```

TOUS ACTIFS INACTIFS			
NOM	EST ACTIF	ADRESSE EMAIL	VILLE
Sport Orléans	Activé	orleans@greenfit.fr	Orléans
Sport Tours	Activé	tours@greenfit.fr	Tours
Sport Blois	Désactivé	blois@greenfit.fr	Blois
Sport Chartres	Activé	chartres@greenfit.fr	Chartre

En cliquant sur le bouton « TOUS »

TOUS	ACTIFS	INACTIFS		
NOM		EST ACTIF	ADRESSE EMAIL	VILLE
Sport O	.556	Activé	orleans@greenfit.fr	Orléans
Sport To	ours	Activé	tours@greenfit.fr	Tours
Sport C	Sport Chartres		chartres@greenfit.fr	Chartres

# En cliquant sur le bouton « TOUS »



En cliquant sur le bouton « INACTIFS »