



UNIVERSITÉ  
DE MONTPELLIER



**RAPPORT FINAL DE STAGE EFFECTUÉ DU 06/04/2021 AU  
28/05/2021**

Synerg-In - Agence digitale sur Montpellier



**Réalisé par :**

Adèle LAURENS

**Sous la direction de :**

Agnès Mazars-Champelon

**Tuteur de stage:**

Pierre CHAMPEMONT

Année Universitaire 2020 - 2021

## Remerciements:

Je tiens à remercier toutes les personnes ayant participé au succès de mon stage et à l'élaboration de mon rapport de stage.

Tout d'abord, je voudrais remercier mon tuteur de stage Pierre Champemont qui a accepté de me prendre dans son entreprise et m'a soutenu durant tout mon stage. Il m'a permis d'apprendre et d'étendre mes connaissances.

Je souhaite aussi remercier toute l'équipe Synerg-in pour m'avoir aussi bien accueillis, intégrée et aidée afin de travailler dans une bonne ambiance: Sabrina Bastid, Vanessa Lesdel, Ines Bastid, Yann Morin, Lucas Jalabert, Raphaël Hauteville, Filipe Elias et Rezlhane Nsar Bianchi.

Enfin, je remercie mon tuteur M.Champemont qui a relu ce rapport et m'a conseillé quant à sa rédaction.

# Sommaire

<b>Présentation de l'entreprise</b>	<b>6</b>
L'agence	6
L'équipe	6
Les pôles	8
Le logo	9
L'organisation	9
<b>Cahier des charges</b>	<b>10</b>
Analyse du sujet et son contexte	10
Analyse de l'environnement	10
Product Backlog	11
Spécifications fonctionnelles	13
<b>Rapport technique</b>	<b>18</b>
Conception	18
Présentation des choix technologiques	18
Description des algorithmes	18
Réalisation	31
Description de l'architecture du programme	31
Organisation des fonctions dans le projet	32
Arborescence des fichiers du programme	32
<b>Résultats</b>	<b>34</b>
Installation	34
Manuel d'utilisation	34
Test/Validation	35
Améliorations possibles	36
<b>Rapport d'activité</b>	<b>37</b>
GitLab	37
Travail effectué en plus du projet principal	37
Discussion sur le travail effectué	39
<b>Conclusion</b>	<b>41</b>
<b>Bibliographie</b>	<b>42</b>
<b>Annexes</b>	<b>44</b>
Pages des maisons prises sur la maquette	44
Base de donnée pour les maisons	46
<b>4eme de couverture</b>	<b>48</b>

# Table des figures

Figure 1: Organigramme de l'entreprise	7
Figure 2: Logo du pôle Synerg-In Marketplace	8
Figure 3: Logo du pôle Synerg-In Marketing & Digital	8
Figure 4: Logo du pôle Synerg-In Stratégie Commerciale	8
Figure 5: Logo du pôle Synerg-In Communication et Événement Pro	8
Figure 6: Tableau Asana recensant les tâches à faire	12
Figure 7: Tableau Asana recensant les tâches et sous-tâches à faire	12
Figure 8: Tableau Asana avec toutes les tâches à faire	13
Figure 9: Capture d'écran du diagramme de cas d'utilisation	14
Figure 10: Capture d'écran de la page d'ajout d'une maison	15
Figure 11: Code de la page twig pour la création d'une cloche de notification	19
Figure 12: Capture d'écran de la cloche de notification sur le site	19
Figure 13: Capture d'écran de la classe "Notification"	19
Figure 14: Code de la fonction add dans la classe TicketController	20
Figure 15: Script pour les notifications	23
Figure 16: Fonction "notification" dans le HomeController	24
Figure 17: Capture d'écran de la page permettant de réinitialiser son mot de passe	25
Figure 18: Script pour la sécurisation de mot de passe	27
Figure 19: Fonction "register" dans le SecurityController	30
Figure 20: Capture d'écran de la page de réinitialisation de mot de passe après avoir rentré deux mots de passe différents	30
Figure 21: Capture d'écran du site Mailtrap	31
Figure 22: Capture d'écran d'une partie de l'arborescence du projet 'lucy'	31
Figure 23 : arborescence des fichiers du projet avec les explications pour chaque fichier	33
Figure 24: Capture d'écran de la page d'accueil en tant que super Admin	34
Figure 25: Schéma montrant les étapes faites pour résoudre un problème	35
Figure 26: Capture d'écran de la page "Evenementiel Pro" sur le site Synerg-in	38
Figure 27: Capture d'écran de la page de visualisation d'une maison	52
Figure 28: Capture d'écran de la table "panier"	52
Figure 29: Capture d'écran de la table "produit"	53
Figure 30: Capture d'écran de la table "house"	53

# Glossaire

Les termes définis dans ce glossaire sont identifiables dans le corps du texte au moyen d'un astérisque (\*).

- **SEO (Search Engine Optimization)**: Ensemble de techniques permettant de positionner un site web dans les moteurs de recherche afin de le rendre visible auprès des internautes.
- **ORM (object-relational mapping)**: Brique logicielle qui se place entre notre application et une BDD. Elle va faire le lien entre notre application et notre BDD en faisant la correspondance entre les enregistrements (tables + enregistrements qu'elle contient dans la BDD) et le monde des objets dans l'application. Donc dans notre application, nous aurons des classes qui vont représenter les différents enregistrements de notre BDD.
- **Conflit (Git)** : se produit lorsqu'un merge ne se passe pas correctement. Des modifications sont donc à apporter sur la branche qui ne peut pas être merge.
- **Merge (Git)** : Fonctionnalité permettant de fusionner les branches de chaque membre de l'équipe et de mettre en commun le code produit par chacun.
- **Product Backlog** : Liste des diverses fonctionnalités à développer, chaque fonctionnalité ayant un ordre d'importance définissant sa priorité de développement.
- **Branches (Git)** : Fonctionnalité permettant à chaque membre du projet de travailler de manière indépendante de la branche principale sur laquelle est située le code global du projet.
- **Framework (cadre de travail)**: Ensemble d'outils, un socle de fonctionnalités qu'on nous offre. Mais pour les utiliser, nous devons suivre certaines règles. Échanger contre la puissance, rapidité de mise en œuvre, un peu de notre liberté.
- **Bundle**: c'est un package/librairie qui existe sur composer et qu'on peut télécharger. À partir du moment où elle sera déclarée dans le fichier "bundles.php", dans la liste des bundles, symfony a aller voir tous les bundle et va enrichir le container de service.

## Introduction

Ma recherche de stage s'est faite de différentes façons, que ce soit par recherche personnelle, les rencontres organisées par l'IUT, via mes contacts, par les mails de l'école, ou encore par le lien git vers différentes entreprises proposant chaque année des stages pour les étudiants suivant le parcours informatique.

Lors des rencontres organisées par l'établissement, j'ai pu passer un entretien avec l'entreprise Radio Aviva. La mission était de créer un site qui recensait toutes les émissions de radio. À partir de là, un mail pouvait être envoyé à toutes les personnes de cette émission via un clic sur ces dernières.

N'ayant pas eu de nouvelles de cet entretien, j'ai commencé à envoyer plusieurs mails à différentes entreprises que j'avais trouvés sur le site ProPages et orientés dans le développement web. L'agence Synerg-in Digital m'a proposé un entretien et suite à celui-ci, une réponse positive de cette dernière m'est parvenue quelques jours plus tard. La mission au sein de cette entreprise était de développer un site web en utilisant le framework\* Symfony. J'ai accepté en me disant qu'en apprendre plus pendant mon stage ne pouvais être que bénéfique.

Synerg-In Digital est une entreprise de communication numérique concevant des sites internet et/ou des applications mobiles pour des clients. Elle s'occupe aussi de tout ce qui est dans l'événementiel au Marché et Halles du Lez. L'agence est composée de 4 pôles: Synerg-In MarketPlace, Synerg-In Marketing & Digital, Synerg-In Stratégie Commerciale et Synerg-In Communication et Evènements pro et comporte 9 employé.e.s réparties dans les différents pôles.

Mon stage s'est déroulé parmi d'autres développeurs:

- 4 stagiaires dont 2 arrivants 2 semaines après moi: Carla Canada et Esteban Dufour et 2 autres 2 semaine avant la fin de mon stage: Lucas Soulairol et Sofiane Amar
- un employé de l'entreprise, Raphaël Hauteville
- un chef de projet IT, Filipe Elias
- et mon tuteur M. Champemont

Tout d'abord je vais présenter l'entreprise, son organisation, son équipe et ses différents pôles. Dans un second temps, j'aborderais l'analyse du projet et le cahier des charges. Ensuite, je vais présenter le rapport technique, les résultats et enfin le rapport d'activité.

# 1. Présentation de l'entreprise

## 1.1. L'agence

Synerg-In Digital est une entreprise de communication digitale et événementielle lancée officiellement le 1<sup>er</sup> septembre. Officieusement au mois de mai, elle a pour but de donner de la visibilité à ses clients au travers de différents outils et moyens : la gestion événementielle, la conception de solutions numériques et digitales et l'exploitation d'une plateforme marketplace, des évènements, ou bien encore de les conseiller de manière bienveillante. Elle a été créée par Sabrina Bastid, ancienne cadre commerciale chez Darty Europe, très fortement expérimentée dans le pilotage et la gestion de projet et Pierre Champemont, freelance depuis plusieurs années avec un portefeuille clients existants, sur la base d'une fusion de compétences: Pierre Champemont avec l'entreprise Inkweb, et Sabrina Bastid avec son profil commercial.

## 1.2. L'équipe

L'agence est composée de 9 employés: Sabrina Bastid, Pierre Champemont, Vanessa Lesdel, Ines Bastid, Yann Morin, Lucas Jalabert, Raphaël Hauteville, Filipe Elias et Rezlhane Nsar Bianchi.

Pendant ma période de stage, quatre nouveaux stagiaires sont arrivés, Carla Canada, Esteban Dufour, Lucas Soulairol et Sofiane Amar. Carla a travaillé sur le SEO\* de la marketplace, la mise en place d'audits de trafic et sur les réseaux sociaux. Esteban était plus intéressé par le front-end et s'est occupé de faire des maquettes pour un projet Moodle nommé Sentein. Lucas Soulairol a fait la formation Symfony ainsi qu'un formulaire sur la marketPlace et avant de partir, je lui ai montré les fichiers concernant la maison afin qu'il puisse continuer ce que j'avais commencé. Enfin, Sofiane Amar fait partie du projet Sentein.

L'équipe est très jeune et l'ambiance est très conviviale même si les caractères de chacun sont différents. J'ai pu travailler en présentiel 2 fois par semaine ce qui était très plaisant car ça m'a permis de mieux connaître les membres de l'agence. De plus, c'était plus facile pour s'entraider dès qu'on avait un souci.

Ci-dessous, un organigramme de l'entreprise fait à l'aide de mon tuteur de stage:

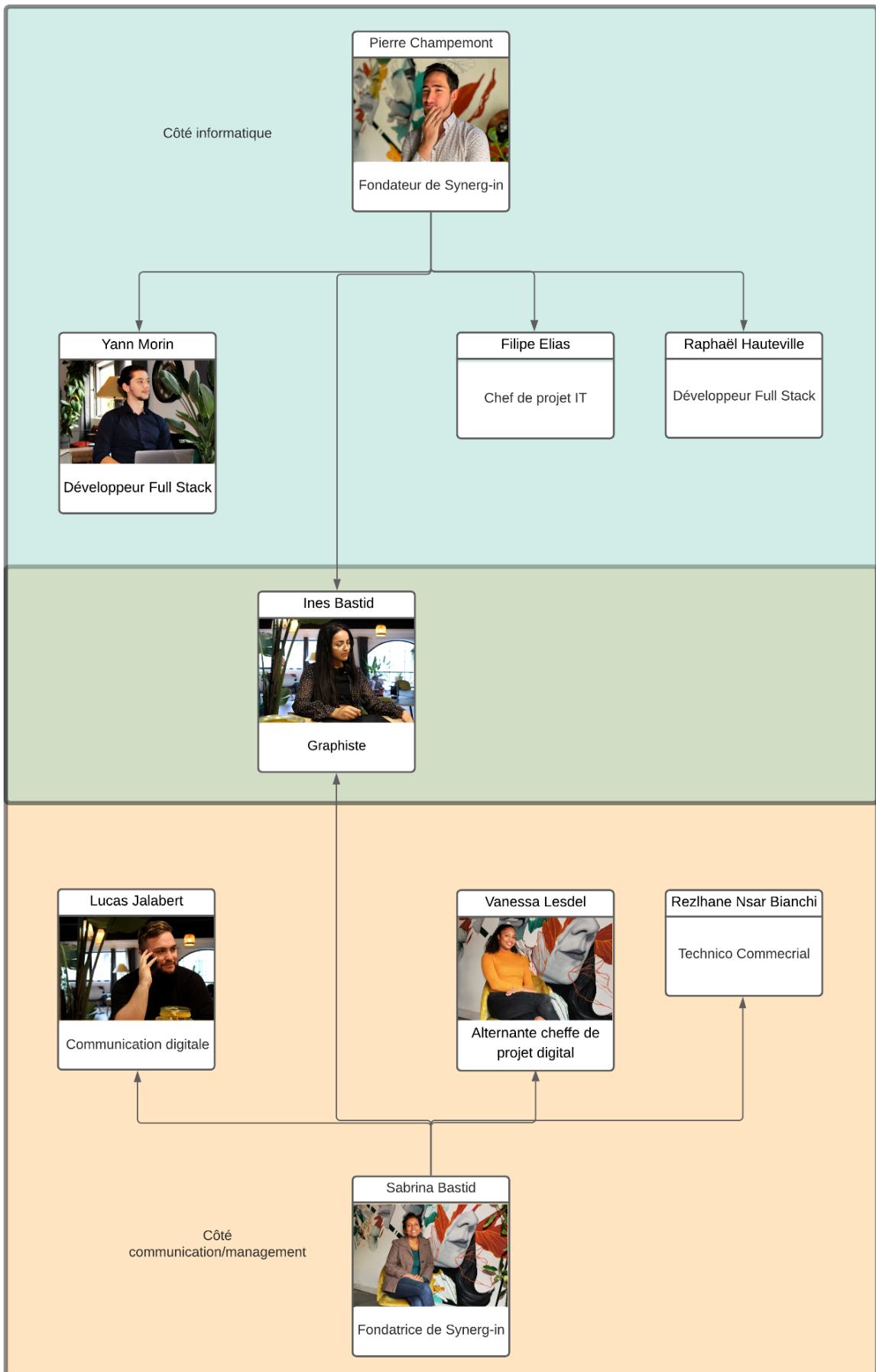


Figure 1: Organigramme de l'entreprise

### 1.3. Les pôles

Il y a 4 pôles au sein de l'entreprise:

Synerg-In MarketPlace :  Ce pôle met en place une Marketplace dans un tiers lieu, quartier ou ville. Par exemple la marketplace du Marché et Halles du Lez	Figure 2: Logo du pôle Synerg-In Marketplace  The logo features a circular arrangement of green leaves and branches. In the center is a stylized yellow and blue geometric shape resembling a mountain or a rising sun. Below the logo, the text "SYNERG-IN" is written in a bold, sans-serif font, with "MARKETPLACE" in smaller letters underneath.
Synerg-In Marketing & Digital :  Un pôle de développement technique de solutions digitales et numériques, comme des sites vitrines, des applications mobiles ou des progiciels sur mesure comme un CRM	Figure 3: Logo du pôle Synerg-In Marketing & Digital  The logo consists of a circular design with red and orange foliage or flame-like shapes on the left and green and blue shapes on the right. The text "SYNERG-IN" is at the bottom, with "MARKETING & DIGITAL" in smaller letters below it.
Synerg-In Stratégie Commerciale :  Ce pôle conseille et oriente les clients qui sont commerçants à avoir une meilleure approche commerciale. Il s'occupe aussi des affaires sur les pôles digitaux et événementiels.	Figure 4: Logo du pôle Synerg-In Stratégie Commerciale  The logo features a circular arrangement of green and yellow leaves and branches. In the center is a stylized red hand reaching out. The text "SYNERG-IN" is at the bottom, with "STRATÉGIE COMMERCIALE" in smaller letters below it.
Synerg-In Communication et Événements pro :  Comme son nom l'indique ce pôle gère la gestion et le déploiement d'un évènement ainsi que les campagnes de communication médias.	Figure 5: Logo du pôle Synerg-In Communication et Événement Pro  The logo features a circular arrangement of green leaves and branches. In the center is a stylized red hand holding a white hand. The text "SYNERG-IN" is at the bottom, with "COMMUNICATION & EVENTS PRO" in smaller letters below it.

## 1.4. Le logo

Le logo de Synerg-In est utilisé pour les 4 pôles. Seul le coloris change. Il représente deux personnes qui vont se serrer la main: l'agence et les clients. Cette poignée de main représente un contrat/un engagement entre les deux parties. Ainsi, l'agence a décidé d'utiliser ce logo pour donner un côté humain à l'entreprise.

Les couleurs et la végétation sur le logo donnent un effet nature et bio qui rappelle le lieu de l'agence, le marché et les Halles du lez à Montpellier. Pour reprendre les mots de mon tuteur il s'agit : «de l'association de compétences et une synergie de talents».

## 1.5. L'organisation

Dans l'agence, 2 grandes tables sont installées afin de diviser les personnes en deux groupes: un pour la communication et un autre pour l'informatique. Chaque groupe comporte un dirigeant.

Chaque semaine, nous avons 2 réunions avec le groupe informatique sur Teams: une le mardi et une le vendredi. Nous partageons notre avancée et nos difficultés et ce que nous allons/sommes en train de faire. Nous utilisons aussi Asana, qui nous permet de choisir des tâches et voir celles en cours afin de s'organiser pour faire avancer le projet. L'organisation de l'agence correspond donc aux méthodes agiles.

Tous les mois, une réunion est faite entre le groupe de communication et celui d'informatique. Cette réunion permet de voir l'avancement de chaque membre et de débattre sur un site afin de voir comment l'améliorer au niveau visuel mais aussi technique. Elle permet aussi de visualiser ce qu'il se passe sur l'autre pôle.

Je travaillais 2 jours par semaine en présentiel dans un worklab aux Marchés et Halles du Lez et le reste de la semaine en distanciel. Nous portions tous le masque et respections les distanciations. J'ai trouvé cela agréable et m'a permis de "changer d'air". J'ai travaillé ainsi durant tout mon stage sur mon ordinateur portable Windows 10 et ai pu communiquer avec mes collègues par Teams lorsque j'étais en distanciel. La plupart des employés sont des alternants et des stagiaires. Ce choix, de la part des deux fondateurs, est dû au fait de vouloir transmettre de nouvelles connaissances. Il y a aussi une part non négligeable au niveau des coûts pour cette startup. En effet, commencer avec des alternants est plus simple et facile. Bien sûr, cela nécessite plus d'investissement temporel qu'un contrat plus qualifié. Mais c'est aussi un des objectifs de l'agence afin d'apporter plus d'encadrement aux stagiaires.

## 2. Cahier des charges

### 2.1. Analyse du sujet et son contexte

La mission principale qui m'était attribuée durant cette période de stage était de concevoir un site web avec Symfony. J'ai accompagné d'autres développeurs. Ce site, destiné à un promoteur vendant des maisons standardisées, permet à différents intervenants, plombiers, électriciens etc., d'avoir accès aux informations nécessaires pour intervenir sur le projet. La création d'une maison représente un projet. Celui-ci est attribué à une entreprise et à partir de là, cette dernière peut attribuer des utilisateurs à ce projet. De plus, le promoteur peut avoir une estimation financière.

J'ai trouvé la mission principale très intéressante car j'ai pu apprendre à utiliser Symfony 4.24.1 via une formation vidéo que m'a envoyée mon tuteur de stage. De plus, je suis arrivée dans le début du projet et j'ai donc pu m'y raccrocher plus facilement.

Durant ce stage, j'ai eu peur de ne pas y arriver car je ne savais pas comment m'y prendre afin de résoudre les problèmes. De plus, je ne voulais pas poser trop de questions à mon tuteur de stage par crainte de le déranger. Mais lorsque je suis arrivée sur le projet "Lucy" (projet Symfony), je me suis sentie plus soulagée. Je commençais aussi à mieux connaître les autres membres du groupe et je me sentais plus à l'aise pour leur poser des questions.

J'ai fait beaucoup de recherches internet durant ce stage qui m'ont permis d'approfondir mes connaissances et j'ai aussi pu remarquer que j'aimais beaucoup travailler sur des projets web.

### 2.2. Analyse de l'environnement

Lors de mon stage, j'ai utilisé plusieurs logiciels en adéquation avec les besoins. Étant déjà présent dans l'entreprise, je me suis familiarisé avec ceux-ci. Ils étaient d'ailleurs plus que suffisant pour travailler correctement avec tous les membres de l'agence. J'avais des logiciels pour coder et d'autres pour m'organiser avec l'équipe. Ainsi, pour coder j'ai utilisé:

- Visual Studio Code : C'est un éditeur de code extensible développé par Microsoft pour Windows, Linux et mac OS. J'ai utilisé ce logiciel pour développer le site web Lucy.

Afin d'utiliser Symfony, j'ai dû télécharger d'autres logiciels:

- Laragon: C'est un environnement de développement universel portable, isolé, rapide et puissant pour PHP, Node.js, Python, Java, Go, Ruby. Cela m'a permis d'accéder et ainsi visualiser ma BDD en local.
- Composer: C'est un outil de gestion des dépendances en PHP. Il permet de déclarer les bibliothèques dont dépend le projet et il les gérera (installera / mettra à jour) pour vous.

Pour organiser et répertorier tous les projets de l'entreprise, nous utilisions:

- GitLab : C'est un logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et la livraison continue.
- Asana : C'est un gestionnaire de communication d'équipe. Le logiciel prend en charge de nombreuses fonctionnalités, notamment les espaces de travail, des projets, des tâches, des étiquettes, des notes, des commentaires.

Enfin, afin de pouvoir communiquer ensemble afin de s'entraider et faire des réunions, nous avions:

- Microsoft Teams : C'est une application de communication collaborative propriétaire officiellement lancée par Microsoft en novembre 2016. Le service s'intègre à la suite Microsoft Office 365 et Skype et propose des extensions pouvant être intégrées à des produits autres que Microsoft.

### 2.3. Product Backlog

Je n'avais pas un cahier des charges à proprement parler mais une liste de tâches à faire sur Asana. Mon tuteur m'en attribuait afin que je sache quoi faire et pour ne pas me perdre dans ce que je faisais. Nous pouvions aussi en choisir nous-mêmes en

fonction de l'avancement des autres. Le chef de projet s'entretenait avec le client et nous faisait ensuite part des changements que nous pouvions noter dans le product backlog\*.

Nom de la tâche	Échéance	Projets
Page ajout/edition/lecture utilisateur		Lucy
House		Lucy
Controller House		Lucy
Plugin Naturellement en famille		Général

Figure 6: Tableau Asana recensant les tâches à faire

Page ajout/edition/lecture utilisateur	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Réduire l'espace en haut de page</li> <li><input checked="" type="checkbox"/> Titre aligné à gauche</li> <li><input checked="" type="checkbox"/> Champ rôle utilisateur</li> <li><input checked="" type="checkbox"/> Champ avatar (ajout d'une image par défaut)</li> <li><input checked="" type="checkbox"/> Case pour contact en sous élément partie informations personnelles</li> <li><input checked="" type="checkbox"/> Champ téléphone fixe et mobile</li> <li><input checked="" type="checkbox"/> Bouton retour</li> <li><input checked="" type="checkbox"/> Upload image</li> </ul>
--	---

Figure 7: Tableau Asana recensant les tâches et sous-tâches à faire

Ci-dessus, des captures d'écrans qui montrent les tâches que je dois faire avec des sous-tâches pour certaines. Ces dernières me permettent de savoir quoi faire plus précisément. De plus, mon tuteur de stage nous les expliquait aussi lors de réunions avant de nous les attribuer.

The screenshot shows the Asana application interface. At the top, there's a navigation bar with 'Lucy' selected, along with options like 'Aperçu', 'Liste', 'Tableau' (which is currently active), 'Chronologie', and 'Plus...'. There are also buttons for 'Rejoindre', 'Rechercher', and 'Changer de formule'. Below the navigation is a search bar and a filter section.

The main area displays a 'Tableau' (kanban board) with three columns: 'À faire', 'En cours', and 'Terminé'. Each column has a '+' button to add new tasks. The 'À faire' column contains tasks such as 'Décalage titre entreprise à gauche', 'Icône alerte à coté du statut si date actuelle supérieure à date expiration', 'Ajout Google Api pour pré-rempissage de l'adresse', and 'Notification'. The 'En cours' column contains tasks like 'Templating' (status 'Moyen'), 'User Liaison prévisionnel' (status 'LA'), 'Page ajout/edition/lecture utilisateur' (status 'LA'), and 'Controller House' (status 'Élevée'). The 'Terminé' column contains tasks such as 'Réduire police navbar', 'Problème de compatibilité conf.php et script statut dynamique sur serveur plesk', 'Ajout logo via upload', and 'Page création d'entreprise'. Each task card includes a small profile picture and a due date indicator.

Figure 8: Tableau Asana avec toutes les tâches à faire

Sur cette capture d'écran, on peut donc voir mes tâches dans la section “Lucy”. Ce tableau représente le Product Backlog\* du projet. Il y a la colonne des tâches à faire, celles qui sont en cours et celles qui sont terminées. Cela me permettait de savoir ce que j'avais à faire et où j'en étais dans mon travail. Je pouvais aussi voir l'avancement des autres membres du projet et ajouter des commentaires à des tâches pour apporter des précisions sur celle-ci.

## 2.4. Spécifications fonctionnelles

Ainsi, afin d'analyser le projet à réaliser, j'ai créé avec l'aide de mon tuteur de stage, un diagramme de cas d'utilisation qui montre comment un projet va être ajouté, les différentes étapes pour le créer et donc quelles fonctionnalités nous devons développer.

[https://drive.google.com/file/d/1uSl2o4kuf8BEZGRkQYKM\\_GpZdzvjt4zU/view?usp=sharing](https://drive.google.com/file/d/1uSl2o4kuf8BEZGRkQYKM_GpZdzvjt4zU/view?usp=sharing)

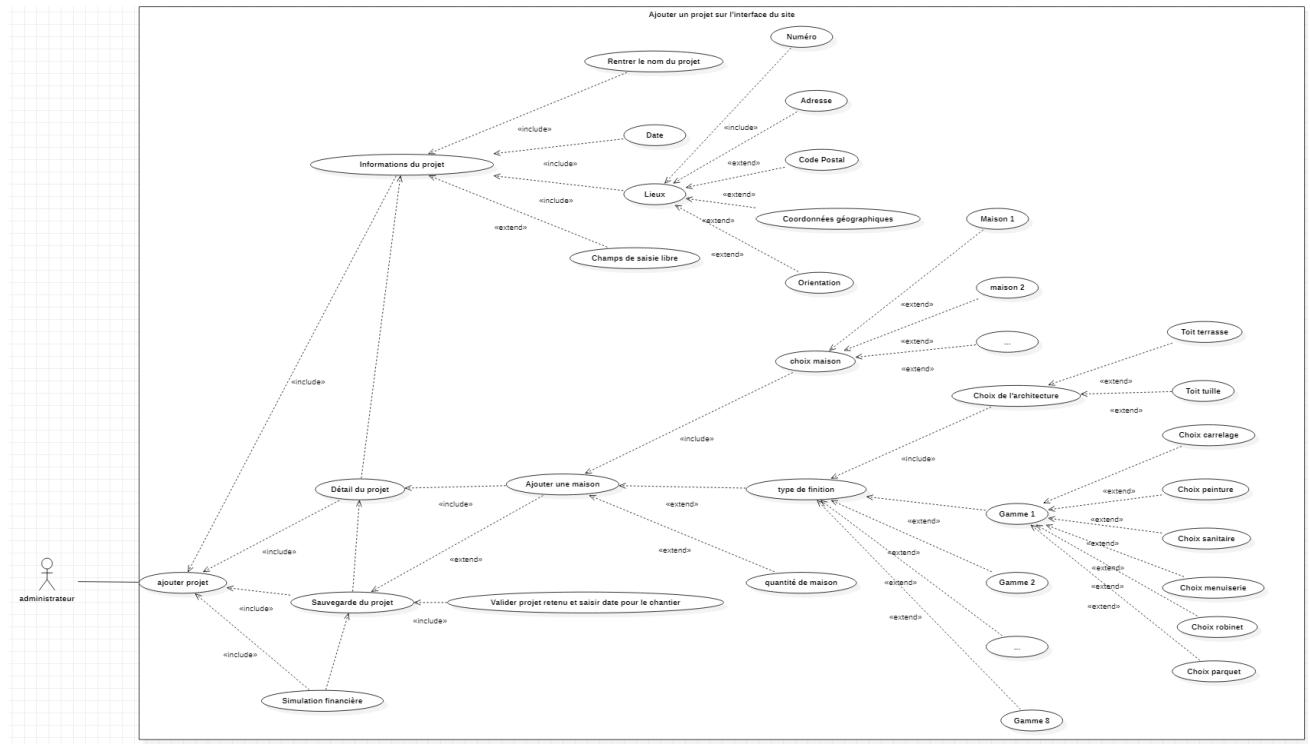


Figure 9: Capture d'écran du diagramme de cas d'utilisation

Lors du projet, j'ai notamment dû m'occuper des pages des maisons. Donc le point sur lequel je travaillais par rapport au diagramme ci-dessus correspond au cas d'utilisation "Ajouter une maison" ainsi que les cas suivants que ce dernier entraîne. Pour m'aider à créer les pages concernant l'ajout, la modification et la visualisation d'une maison, le chef de projet Filipe a fait des maquettes afin que les développeurs aient des modèles sur lesquels se reposer.

Voici par exemple celle pour ajouter une maison:

**Bouton flottant + sauvegarde auto + bouton**

clic sur la case pour éditer le contenu (éditeur)

Figure 10: Capture d'écran de la page d'ajout d'une maison

Ces pages ne sont pas définitives et sont susceptibles d'être modifiées si nous rencontrons des problèmes lors de leur conception. Par exemple, si une icône d'édition se trouve sur la page qui permet de visualiser les maisons, nous devons modifier cela car sur ce type de page , nous ne pouvons pas ajouter/modifier

d'informations. Par conséquent, nous interrogions le chef de projet sur la situation afin qu'il puisse orienter nos choix tout en respectant l'avis du clients.

Nous avons créé une base de données nommées 'lucy' qui pouvait répondre à ces attentes.

Pour cela, nous avons téléchargé le bundle de "Doctrine" (qui est une ORM\*), via la commande "composer require doctrine" dans le terminal de VS Code. La connexion à la base de données est possible grâce au fichier .env contenant des variables d'environnement. Nous avons une donnée où nous pouvons mettre notre identifiant, notre mot de passe et le nom de notre base de données afin de nous connecter à notre BDD locale.

Ensuite, nous avons créé notre BDD grâce à la commande:

```
php bin/console doctrine:database:create
```

À partir de là, nous pouvons créer une entité dans cette BDD en utilisant la commande:

```
php bin/console make:entity "nom de l'entité"
```

qui va créer une classe dans notre application représentant une table sur MyAdmin. Nous rentrons aussi autant d'attribut que nous désirons dans cette entité.

Enfin, nous créons une migration et envoyons les informations sur la BDD via les commandes:

```
php bin/console make:migration
```

qui "sauvegarde" les modifications/ajouts à l'instant où nous lançons la commande

```
php bin/console doctrine:migrations:migrate
```

qui envoie toutes les nouvelles migrations sur à la BDD depuis la dernière enregistrée dans la BDD

Ainsi, après avoir récupéré toutes les informations du client et analysé la maquette faite par le chef de projet, la base de données permettant de créer une maison était constituée de:

- une table 'house' qui comporte tous les principaux attributs qui caractérisent la maison
- une table 'panier' qui permet de relier les maisons et les produits. Elle permet aussi d'attribuer une valeur et un étage pour chaque produit
- une table 'produit' qui contient que des produits, tous différents les uns des autres via leur "référence" qui se trouve dans l'attribut json "propriete".

Cette BDD a évolué au fur et à mesure du projet car de nouvelles informations arrivent que ça soit sur les maquettes ou de par le chef de projet. Ainsi, nous devions être modulables.

### 3. Rapport technique

#### 3.1. Conception

##### 3.1.1. Présentation des choix technologiques

Le projet était déjà en cours lorsque je suis arrivée dessus. J'ai suivi une formation afin d'apprendre à utiliser Symfony. J'ai donc dû téléchargé Visual Studio Code ainsi que les extensions:

- Dart 3.22.0 est un langage de programmation orientée objet,
- Flutter 3.22.0 qui est un framework\* permettant de créer des applications,
- PHP Getter&Setter 1.2.3 qui permet de créer des getter et/ou setter en cliquant sur un attribut,
- PHP Intelephense 1.7.1 qui permet d'améliorer les fonctionnalités PHP de VSCode (autocomplétion, importations etc.),
- PHP Namespace Resolver 1.1.8 qui permet d'importer des classes via un clic droit
- et Twig Language 2 0.9.2 qui nous offre des fonctionnalités et raccourcis dans les fichiers .twig.

##### 3.1.2. Description des algorithmes

Lors du projet, j'ai créé des pages .twig et utilisé du js. Par exemple, j'ai été emmenée à mettre en place un système de notification. C'est-à-dire, lorsqu'une maison ou un ticket était créé, cela devait incrémenter le chiffre dans la cloche des notifications.

- Notification

Dans ma page twig, j'ai donc ajouté le visuel:

```
{# Icône de notification #}
<div id="clocheNotif" style="position : relative; z-index: 1;"></div>

{# Nombre de notifications (chiffre) #}
```

```
<div id="nbrNotification" style="color : white; position : relative;
z-index : 2; left : -27px;"></div>
```

Figure 11: Code de la page twig pour la création d'une cloche de notification

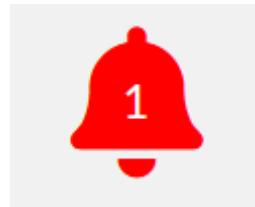


Figure 12: Capture d'écran de la cloche de notification sur le site

Ensuite, j'ai créé une Entité "Notification":

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b>	int(11)			No	None		AUTO_INCREMENT
2	<b>id_user_id</b>	int(11)			Yes	NULL		
3	<b>view</b>	tinyint(1)			Yes	NULL		
4	<b>name</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
5	<b>attribut</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		

Figure 13: Capture d'écran de la classe "Notification"

Ainsi, dès qu'une maison ou un ticket est créé, j'ajoute une ligne dans la table de notification via mon controller en faisant par exemple:

```
//création de la route accessible sur le site via l'url
https://localhost:8000/ticket/ajouter
/**
 * @Route("/ticket/ajouter", name="ticket_add")
 */
public function add(Request $request, SluggerInterface $slugger,
EntityManagerInterface $em)
{
    // Création d'un ticket
    $ticket = new Ticket;

    // Création du formulaire "ticketType" pour pouvoir ajouter un ticket
    $form = $this->createForm(TicketType::class, $ticket);
    $form->handleRequest($request);
```

```

    // Si on clique sur le bouton de type "submit" et que les champs
    "required" sont remplis
    if ($form->isSubmitted() && $form->isValid($ticket)) {

        //On remplit certains attributs importants mais pas accessibles
        via le formulaire
        $ticket
            ->setidUser($this->getUser())
            ->setView("0");

        //On persist le ticket (on "gère" l'objet $ticket mais aucune
        requête n'est faite)
        $em->persist($ticket);

        //On crée une notification et on remplit les champs importants
        $notification = new Notification;
        $notification
            ->setView(0)
            ->setName("ticket_show")
            ->setidUser($this->getUser());

        //On enregistre la notification
        $em->persist($notification);

        //Le gestionnaire d'entité exécute une requête INSERT afin de
        créer une
        //nouvelle ligne dans la table notification et ticket
        $em->flush();

        //On va chercher la route ayant comme nom "home_index"
        return $this->redirectToRoute('home_index');
    }

    //Tant qu'on ne clique pas sur le bouton "submit" on retourne la page
    add dans le dossier
    //ticket auquel on ajoute le formulaire
    return $this->render('ticket/add.html.twig', [
        'formView' => $form->createView(),
    ]);
}

```

Figure 14: Code de la fonction add dans la classe TicketController

Enfin, j'ai créé un script avec l'aide de Yann afin de pouvoir récupérer le nombre de lignes dans la table "notification".

```
{# On ouvre une balise script dans la page twig #}
<script>

// Lorsqu'on charge la page, on lance la fonction requeteAjax()
window.onload = function () {
requeteAjax();
}

// Cette fonction ajoute l'icône de notification au niveau du div ayant comme
id clocheNotif
function remplir() {
$("#clocheNotif").append("<i class='fas fa-bell' style='color : red;'></i>");
}

// Cette fonction fait disparaître la cloche de notification
function vider() {
document.getElementById("clocheNotif").innerHTML = "";
}

// On fait une requête ajax
function requeteAjax() {
request = $.ajax({
    // On va chercher la fonction qui contient le code permettant d'accéder au
nombre de
    // lignes dans la table notification (ici la route ayant comme nom
'notification')

url: "{{ path('notification') }}",

// la requête se fait en POST
type: "POST",

// Si on arrive à récupérer ce nombre, alors une fonction se lance avec comme
paramètre
// data qui est ce qu'a renvoyé le code
success: function (data) {
```

```

    // On transforme le résultat récupéré en nombre pour que ça soit plus
simple et on le
    // met dans une variable
let nombre = Number(data.replace(/\[^d]/g, ""));

// si ce nombre est égal à 0 alors on n'affiche pas la cloche
if (nombre == "0") {
vider();
}

// sinon on l'affiche
else {
vider();
remplir();
}

//on ajoute le nombre de notification à la cloche
document.getElementById('nbrNotification').innerHTML = nombre;
}

});

//si la requete ajax fonctionne on exécute la fonction suivante
request.done(function () {

// on récupère le nombre de notification
let nbrNotif = document.getElementById('nbrNotification');

// si le nombre de notification est égal à 0, on ne l'affiche pas
if (nbrNotif.innerHTML == "0") {
nbrNotif.style.display = "none";
document.getElementById('clocheNotif').style.display = "none";
}

// sinon s'il est supérieur à 9, on lui donne une certaine position sur la
page
else if (parseInt(nbrNotif.innerHTML) > 9) {
nbrNotif.style.left = "-30px";
}
})

```

```

// Sinon (s'il est inférieur ou égal à 9), on lui donne une autre disposition
sur la page
else {
nbrNotif.style.display = "block";
document.getElementById('clocheNotif').style.display = "block";
nbrNotif.style.left = "-27px";
}

});

}

// on lance la fonction requeteAjax toutes les 6 secondes afin de pouvoir
// actualiser le chiffre souvent
let chb = setInterval(requeteAjax, 6000);
</script>

```

Figure 15: Script pour les notifications

Voici le code qu'il y a dans la route ayant le nom “notification”:

```

// Route ayant le nom notification
/**
 * @Route("/notification", name="notification")
 */
public function notification(NotificationRepository
$notificationRepository)
{
    /* On regarde si l'utilisateur est connecté
    s'il ne l'est pas, on retourne à la page d'authentification */
    if (!$this->getUser()) {
        return;
    }

    // Sinon, on fait un echo du nombre de lignes qu'il y a dans la table
notification
    // dont l'idUser est égal à celui de l'utilisateur connecté et dont
l'attribut "view"
    // est égal à 0 (cela signifie ici qu'il n'a pas encore ouvert la
notification)
    else {
        /* CHECK NUMBER OF NOTIFICATION NOT VIEW */
    }
}

```

```
    echo count($notificationRepository->findBy(['idUser' =>
$this->getUser(), 'view' => '0']));
        exit();
    }
}
```

Figure 16: Fonction “notification” dans le HomeController

Ainsi, j'ai pu afficher le nombre de notifications en fonction de l'utilisateur.

Tout ce travail s'est fait en beaucoup d'étapes. Au début, j'ai voulu faire une requête ajax comme vu en cours mais ça n'était pas adapté à Symfony. Avec l'aide de Yann, j'ai donc conçu un fichier PHP dans lequel j'ai créé une requête PDO pour me connecter à ma BDD, une requête SQL que j'ai préparée puis envoyée dans une requête Ajax d'un fichier JS. Lorsque ceci marchait, Yann m'a ensuite aidé pour que le code soit plus optimal. J'ai donc pu raccourcir le code du fichier php en quelques lignes comme vu ci-dessus.

Je me suis aussi occupée de mettre en place une sécurité sur les mots de passe créés par les utilisateurs. Si ces derniers n'en créent pas un comportant une minuscule, une majuscule, un caractère spécial et un chiffre, alors le mot de passe n'est pas enregistré.

- Sécurité du mot de passe

Tout d'abord, voici à quoi ressemble la page pour modifier son mot de passe. J'ai commencé à rentrer un mot de passe contenant une majuscule et une minuscule. Nous pouvons voir qu'un paragraphe s'est mis en gras et que l'icône de validation s'est cochée.

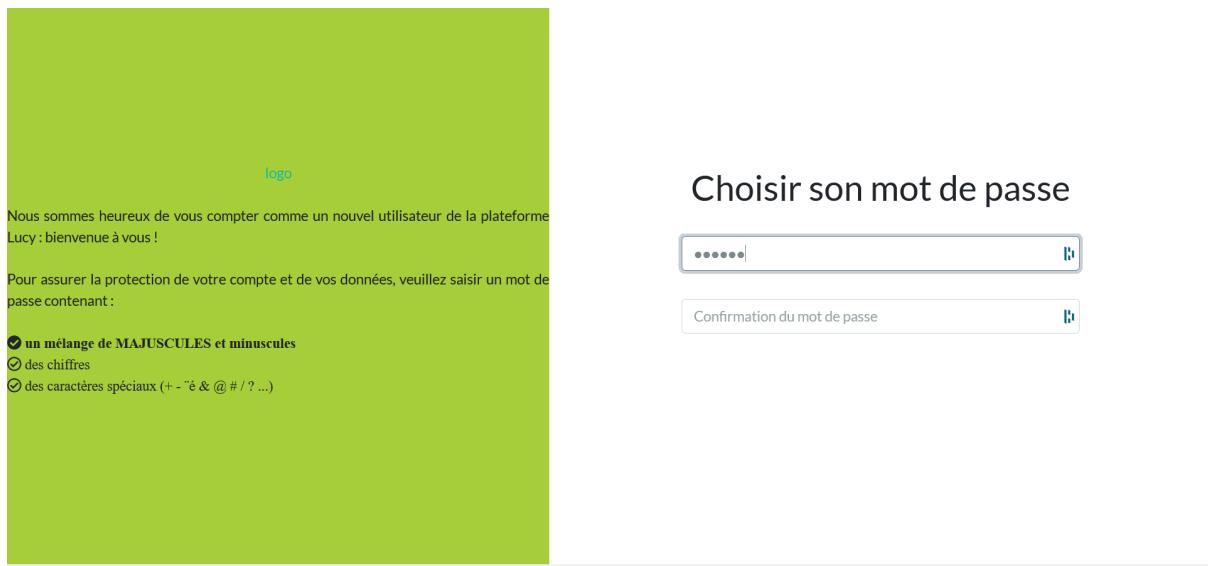


Figure 17: Capture d'écran de la page permettant de réinitialiser son mot de passe

Pour que nous puissions changer le style des paragraphes afin de signaler le client que son code respecte bien les normes, j'ai créé un script:

```
{# ouverture de la balise pour le script #}
<script>

    // On cache le bouton de validation (invisible sur la page)
document.getElementById("boutonSubmitRegister").style.visibility = "hidden";

// dès qu'on rentre un caractère dans le champ de mot de passe
document.getElementById("register_password_password").addEventListener('input'
, function () {

    // la variable mdp correspond au mot de passe rentré
    let mdp = document.getElementById("register_password_password").value;

    // on regarde si le mot de passe contient des majuscules
    let maj = /[A-Z]/g;
    let foundMaj = mdp.match(maj);

    // on regarde si le mot de passe contient des minuscules
    let min = /[a-z]/g;
    let foundMin = mdp.match(min);

    // on regarde si le mot de passe contient des chiffres
    let chiffre = /[0-9]/g;
```

```

let foundChiffre = mdp.match(chiffre);

//on regarde si le mot de passe contient des caractères spéciaux
let modele =
/[ààääääçéèëëïïñóòôööùûüýÿæœÀÀÄÄÅÇÉÈËËÍÌÎÑÓÒÔÖÖÙÛÜÝÝÆŒ.#^&é\#\{\_}\|è`ç
^à\[\\]°=+/:;,?!§?*²$#\_\s\.,\,\-\-]/;
let foundCarSpe = mdp.match(modele);

// si le mdp contient des majuscules et des minuscules, alors on change
l'icône de validation
// et on la met en check
if (foundMaj && foundMin) {
    document.getElementById("majMin").className = "fas fa-check-circle";
}

//sinon on ne met pas l'icône du paragraphe correspondant en check
else {
    document.getElementById("majMin").className = "far fa-check-circle";
}

// on fait la même chose pour le paragraphe spécifiant que le mdp contient
des chiffres
if (foundChiffre) {
    document.getElementById("chiffres").className = "fas fa-check-circle";
}
else {
    document.getElementById("chiffres").className = "far fa-check-circle";
}

// on fait la même chose pour le paragraphe spécifiant que le mdp contient
des caractères spéciaux
if (foundCarSpe) {
    document.getElementById("carSpe").className = "fas fa-check-circle";
}
else {
    document.getElementById("carSpe").className = "far fa-check-circle";
}

// si le mot de passe contient au moins une minuscule, une majuscule, un
caractère
// spécial et un chiffre alors le bouton de validation devient visible
if(foundMaj && foundMin && foundChiffre && foundCarSpe){

```

```

        document.getElementById("boutonSubmitRegister").style.visibility =
"initial";
    }

    //sinon le bouton reste caché
    else{
        document.getElementById("boutonSubmitRegister").style.visibility =
"hidden";
    }
}

</script>

```

Figure 18: Script pour la sécurisation de mot de passe

Une fois le visuel mis en place, j'ai fait en sorte que cela marche aussi quand on envoie le mot de passe donc qu'on clique sur le bouton valider. Voici donc le controller correspondant à la page:

```

/**
 * @Route("/register/{token}", name="security_register")
 */
public function register(Request $request, $token, UserRepository
$userRepository, EntityManagerInterface $em)
{
    /* Check if user log */
    $userLoggin = $this->getUser();
    if ($userLoggin) {
        return $this->redirectToRoute('home_index');
    }

    /* Check if user exist */
    $user = $userRepository->findOneBy(array('activationToken' =>
$token));
    if (!$user) {
        return $this->redirectToRoute('security_login');
    }

    /* Check if user exist and userToken in time */
    if ($this->isRequestInTime($user->getDelayTokenAt()) ||
$user->getDelayTokenAt() === null) {

```

```

    //On crée le formulaire
    $form = $this->createForm(RegisterPasswordType::class);
    $form->handleRequest($request);

    //Quand on clique sur le bouton valider et que tout est respecté et rempli
    if ($form->isSubmitted() && $form->isValid()) {

        // Si les deux mots de passe sont identiques
        if ($form->get('password')->getData() ===
$form->get('conf_password')->getData()) {

            // On met dans la variable $password le mot de passe
            $password = $form->get('password')->getData();

            // On regarde que le mot de passe contienne bien au moins
une majuscule, une
            //minuscule, un chiffre et un caractère spécial
            if
(preg_match('/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$%^&]).*$/', $password)) {

                // On hash le code afin qu'il soit plus sécurisé dans
la BDD
                $hash = $this->encoder->encodePassword($user,
$form->get('password')->getData());

                // On ajoute ce mot de passe dans la table user à
l'utilisateur concerné
                $user->setPassword($hash);

                // On met des variables nulles car elles ne sont plus
utiles
                //le mot de passe a été modifié donc nous n'en avons
plus besoin
                $user->setActivationToken(null, null);
                $user->setDelayTokenAt(null);

                // On persist $user et on l'envoie dans la BDD
                $em->persist($user);
                $em->flush();

                // On revient sur la page de connexion
                return $this->redirectToRoute('security_login');
            }
        }
    }
}

```

```

        }

        //Si le code ne contient pas au moins 1 des 3 conditions
(majuscule,
 // minuscule, chiffre, caractère spécial)
else {

    // On ajoute un message d'erreur sur la page comme
quoi les
    // champs de mot de passe n'ont pas été remplis
correctement
    echo "
<script>
window.onload = function () {
let a = document.createElement('p');
a.innerHTML = 'Vous n\'avez pas remplis les champs
correctement';

document.getElementById('ajoutChampsErreur').appendChild(a);
};

</script>";
}

}

// si les deux mots de passe ne sont pas identiques
else {

    // On ajoute un message d'erreur sur la page comme quoi
les
    // mots de passe ne sont pas identiques
    echo "
<script>
window.onload = function () {
let a = document.createElement('p');
a.innerHTML = 'Les mots de passes doivent être
identiques';

document.getElementById('ajoutChampsErreur').appendChild(a);
};

</script>";
}

}

```

```

        // Tant qu'on ne clique pas sur le bouton submit, la page de
réinitialisation
        // est affichée
        return $this->render('security/register.html.twig', [
            'formView' => $form->createView(),
        ]);
    }

    // si l'utilisateur n'existe pas on revient sur la page de connexion
    return $this->redirectToRoute('security_login');
}

```

Figure 19: Fonction "register" dans le SecurityController

Donc si les deux mots de passe ne sont pas identiques, voici à quoi ressemble la page:

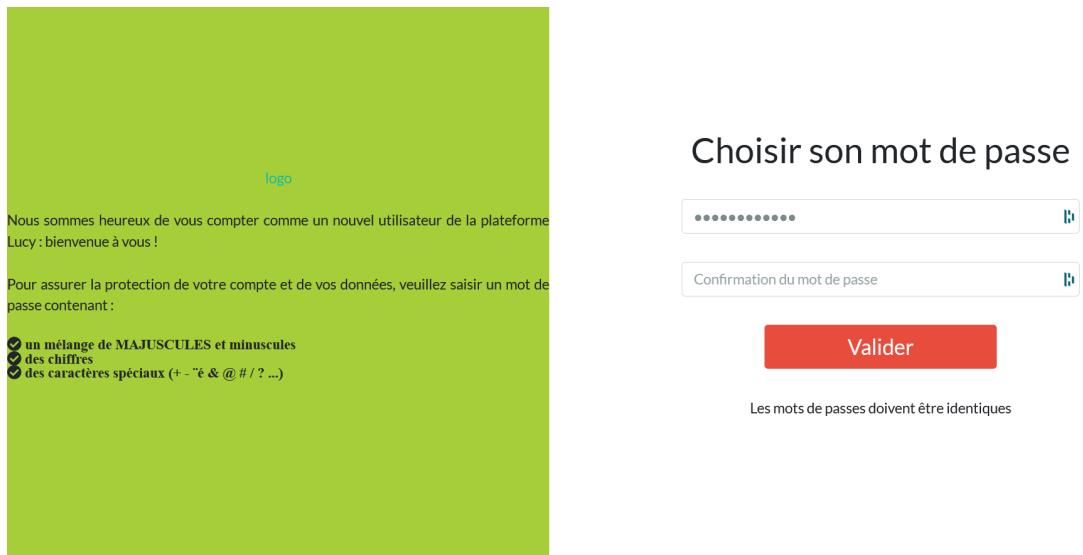


Figure 20: Capture d'écran de la page de réinitialisation de mot de passe après avoir rentré deux mots de passe différents

Pour faire tout cela, Yann m'a aidé afin de m'expliquer comment se passait la réinitialisation d'un mot de passe au niveau de la BDD. Il m'a montré le logiciel "mail trap" qui nous permet de récupérer un lien nous ramenant vers la page ci-dessus.

Voici à quoi ressemble ce logiciel:

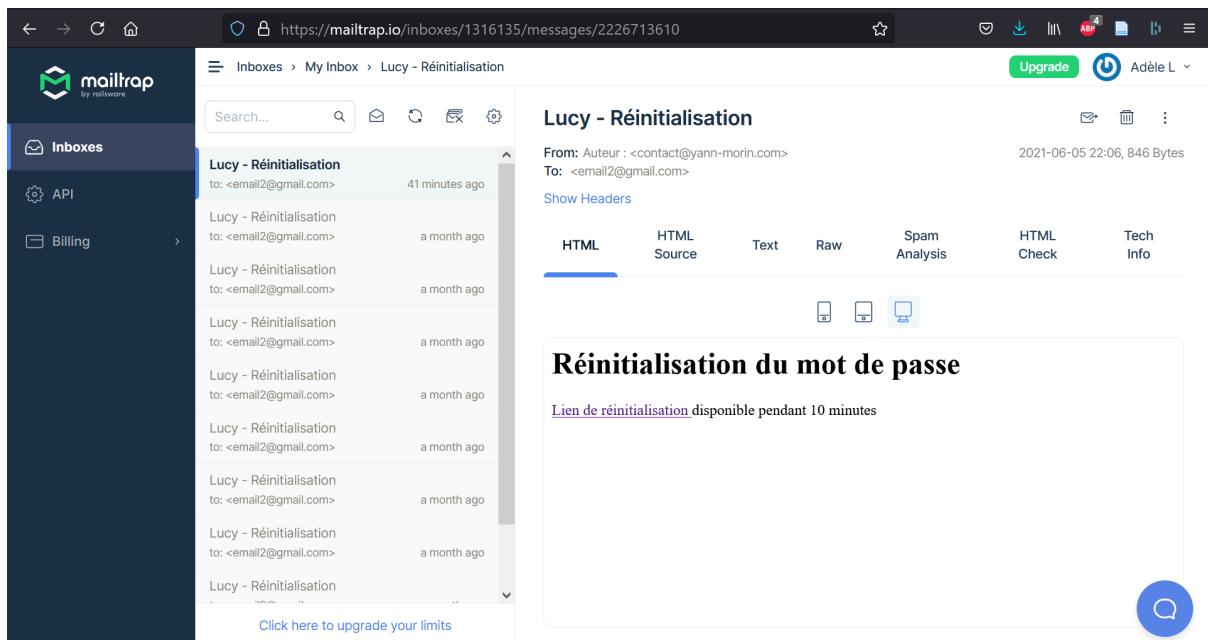
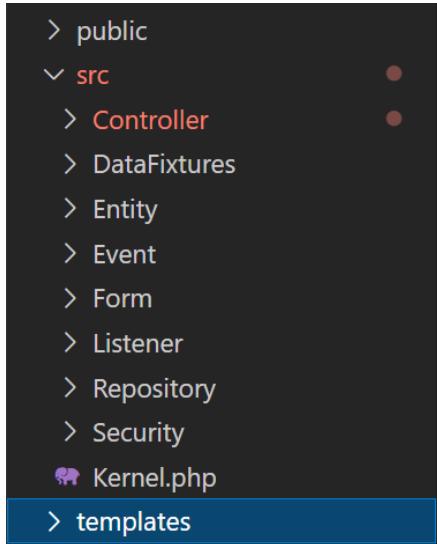


Figure 21: Capture d'écran du site Mailtrap

## 3.2. Réalisation

### 3.2.1. Description de l'architecture du programme

Figure 22: Capture d'écran d'une partie de l'arborescence du projet 'lucy'



Le projet est réalisé avec une architecture simple et bien organisée.

Il y a un dossier “public” où nous mettons tout le css, les images, le js et le php accessibles au public. Ensuite, nous avons le dossier “src” dans lequel nous mettons tout le code.

Dans ce dernier, nous avons un dossier Controller qui contient tous les controller que nous utilisons. Dans ces classes, nous créons des “Routes” afin de pouvoir accéder aux pages sur le site en rentrant la bonne url. Nous pouvons aussi appeler des

formulaires pour ajouter, modifier, supprimer des tables de l'entité concernée.

Nous avons aussi le fichier AppFixtures dans le dossier “DataFixtures” qui permet de remplir la BDD rapidement pour pouvoir faire des tests et rendre le site plus réaliste.

Nous avons le dossier “Entity” qui contient toutes les entités créées.

Il y a le dossier “Form” qui contient tous les formulaires.

Pour finir, le dossier “Repository” se crée automatiquement à la création d'une entité et nous permet d'aller chercher via différentes fonctions les données d'une entité.

Donc par exemple le nom d'une maison.

Enfin, nous avons un dossier “Template” qui contient toutes les pages twig

### 3.2.2. Organisation des fonctions dans le projet

Toutes les fonctions dans les balises <script> des pages .twig sont spécifiques à la page dans laquelle elle se trouve. Celles dans les dossiers js permettent d'être utilisées dans différentes pages .twigs.

Pour chaque fonction dans les Controller, nous avons une page correspondante et accessible sur le web via la route notée en dessus de la fonction. Elles servent à créer les bons formulaires pour chaque page ainsi qu'à remplir/modifier des lignes dans la BDD lorsque ce dernier est envoyé, ou seulement visualiser certaines données contenues dans une table.

Les fonctions dans les “Type” du dossier “Form” servent à créer des formulaires et ainsi avoir la possibilité de les appeler dans les controller.

Les fonctions dans les “Repository” sont créées automatiquement tout comme le fichier. Elles permettent de récupérer l'entité concernée dans un controller de différentes manières.

### 3.2.3. Arborescence des fichiers du programme

Voici maintenant l'arborescence des fichiers que nous utilisons pour mettre en place le site. Cette arborescence respecte celle de Symfony contenant l'architecture MVC. Nous avons utilisé Laragon où nous avons ajouté PHP 8.0.3 et PhpMyAdmin avec la version 10.3.2. Nous avons utilisé composer avec la version 2.0.13.

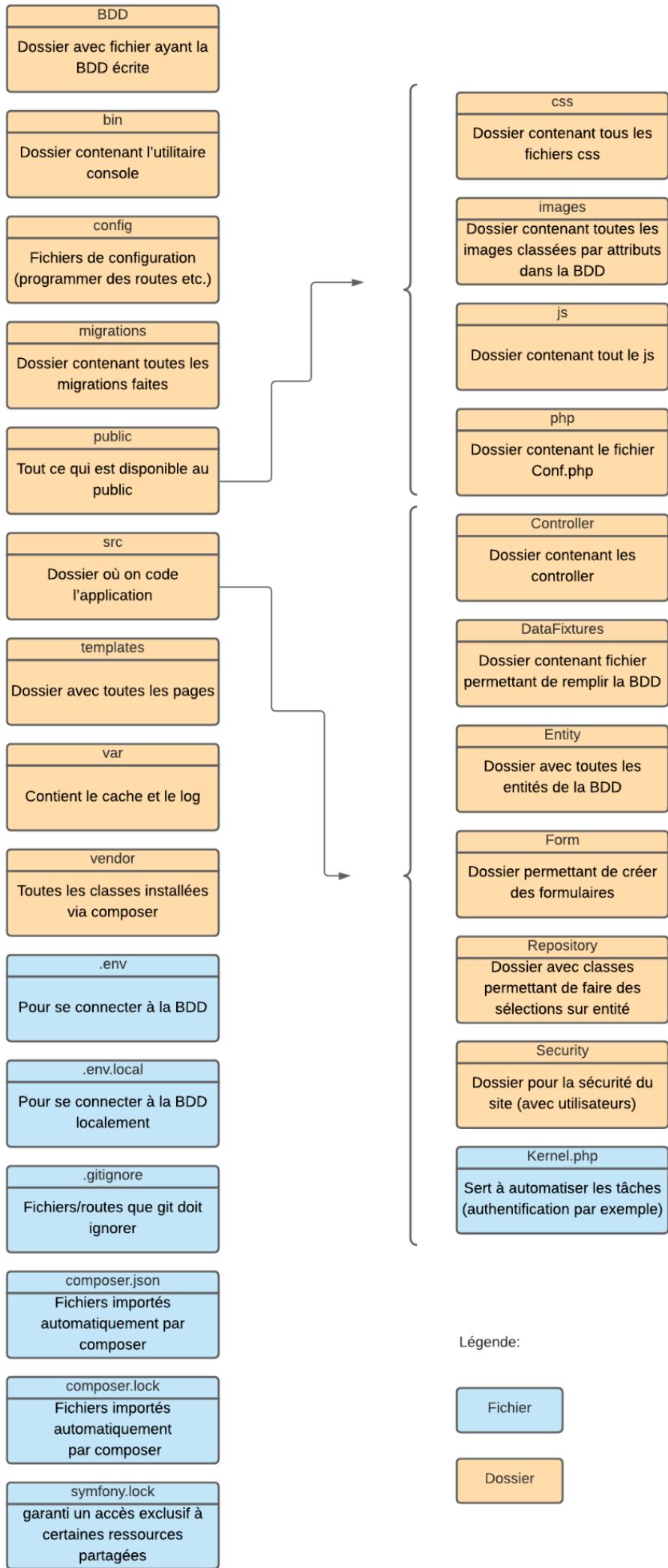


Figure 23 : arborescence des fichiers du projet avec les explications pour chaque fichier

Légende:



## 4. Résultats

### 4.1. Installation

Je vais maintenant présenter l'installation de tous les outils que j'ai utilisé afin de pouvoir faire un site web Symfony sous Windows.

- Pour installer Visual Studio Code: <https://code.visualstudio.com>
- Télécharger Laragon : <https://laragon.org>
- Il ne faut pas oublier de rajouter le chemin du dossier où a été installé PHP dans notre variable d'environnement PATH. Pour cela, nous pouvons faire la suite des commandes ci-dessous:  
Windows -> taper "env" dans la barre de recherche -> cliquer sur "Modifier les variables d'environnement système" -> "paramètre système avancés" -> "Variables d'environnement" -> cliquer sur "PATH" -> ajouter le chemin vers le dossier php
- Pour installer Composer : <https://getcomposer.org>

### 4.2. Manuel d'utilisation

En tant que super admin, nous avons accès à cette page qui permet de tout visualiser rapidement et efficacement:

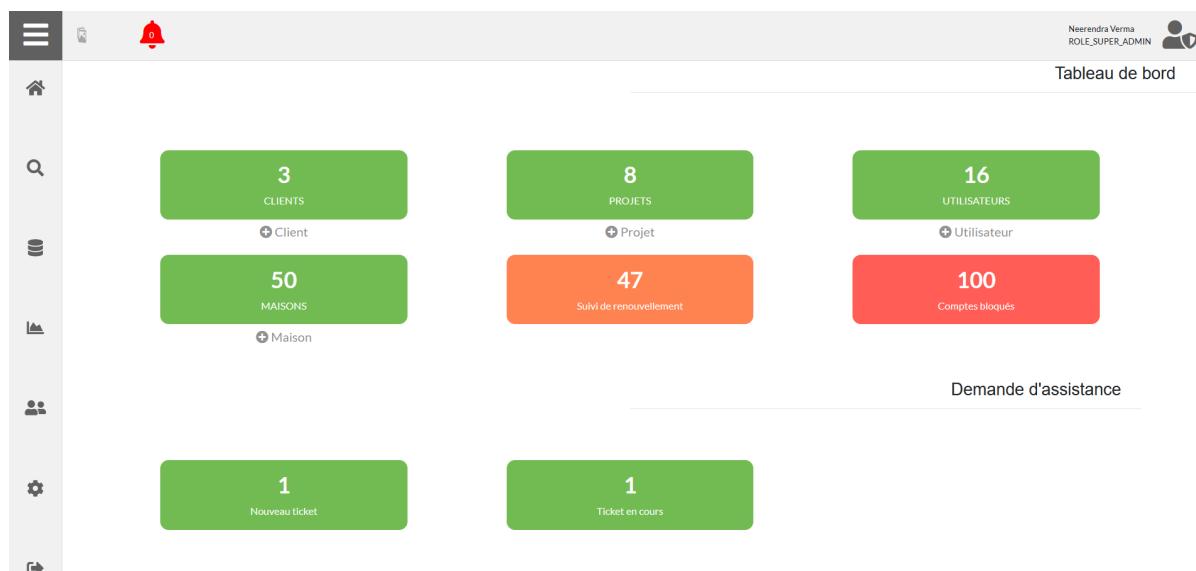


Figure 24: Capture d'écran de la page d'accueil en tant que super Admin

Nous pouvons donc voir quel client gère quelle entreprise et de quel(s) projet(s) elle(s) s'occupe(nt) ainsi que les utilisateurs travaillant dessus.

### 4.3. Test/Validation

Les tests que j'ai faits sont visuels. En effet, le projet n'étant pas fini, je n'ai pas eu le temps de faire des tests unitaires. De plus, je n'ai pas appris à le faire et cela n'était pas dans mes missions.

Voici donc ce que je fais lorsque j'ai un problème pour ajouter une maison. Ceci représente aussi ce que je fais pour la plupart des difficultés que je rencontre:

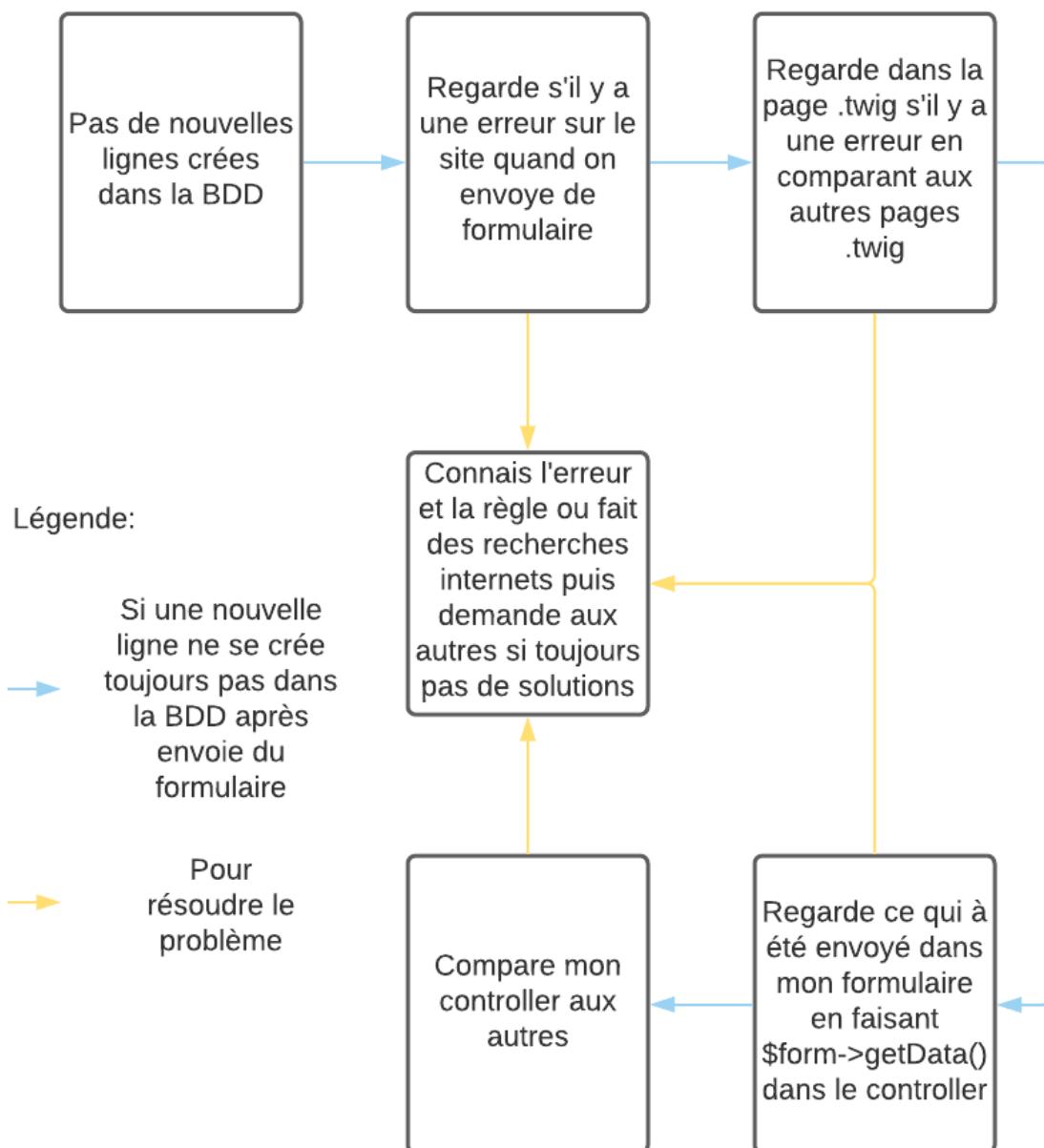


Figure 25: Schéma montrant les étapes faites pour résoudre un problème

Lorsque j'ai des erreurs dans mon js, je fais beaucoup de console.log pour chaque étape afin de pouvoir visualiser à quel endroit je me suis trompée et si ce que ça renvoie est bien ce que j'attends.

#### 4.4. Améliorations possibles

Il y a beaucoup d'améliorations possibles dans chaque tâche que j'ai faite mais aussi dans mon organisation.

- notifications: Je suis partie dans l'idée d'incrémenter le nombre de notifications à chaque fois qu'une ligne était rentrée dans la BDD mais je n'ai pas trouvé la documentation pour. Je me suis donc ravisée et ai fait un appel à une fonction toutes les 6 secondes.
- pages .twig des maisons: améliorer les fichiers dans le dossier "shared"
- optimiser/améliorer le code dans certains controller et mieux utiliser les outils Symfony. Je n'avais pas l'habitude de les utiliser et j'ai remarqué ça lorsque j'ai fait la tâche des notifications. J'avais juste à utiliser les Repository plutôt qu'aller faire la requête SQL moi-même.
- essayer de bien tout visualiser et de noter toutes les étapes nécessaires pour réaliser la tâche avant de la commencer. Par exemple, pour la tâche de sécurité pour les mots de passe, lorsque j'avais fini le javascript, je n'ai pas pensé à vérifier toutes les conditions dans le controller. Le même problème s'est posé pour la création de la page .twig sur les détails de la maison. J'ai pris beaucoup trop de temps à décider pour créer la base de données ce qui m'a beaucoup retardée. Il faut que je prenne l'habitude de réfléchir avant de coder plutôt que directement coder et tomber sur des problèmes au fur et à mesure. Cela ne me fait que perdre du temps.
- ne pas hésiter à demander de l'aide. En effet, notamment au début de la formation, je ne connaissais personne et j'avais aussi peur de les déranger dans leur travail en posant trop de questions. Je me suis aperçu durant mon stage que c'est justement à en posant que j'avançais plus rapidement. Effectivement, je pouvais ensuite travailler plus efficacement grâce aux bases et aux outils mis en main suite aux explications reçues (que je ne connaissais pas forcément).

## 5. Rapport d'activité

### 5.1. GitLab

Nous utilisons GitLab afin de sauvegarder le projet. Nous travaillons tous sur une branche\* que nous nommons en fonction de la tâche attribuée. Lorsque nous l'avons fini et que tout fonctionne, nous mergons\* nos modifications dans la branche\* "dev" et réglons tous les conflits\* s'il y en a. Ensuite, en fin de semaine, Yann ou Raphaël font un merge\* de la branche\* "dev" vers la branche\* "master". Ainsi, le chef de projet peut voir les modifications sur le lien: <https://preprod.inkweb.fr/login> qui correspond à la branche\* master et nous prévenir s'il devait y avoir une modification/amélioration. Mon tuteur de stage aussi nous rajoute des tâches dans Asana afin de rendre le site plus présentable pour le client.

### 5.2. Travail effectué en plus du projet principal

Au début du stage, j'ai fait plein de petites missions que m'ont confiées mon tuteur de stage et aussi un commercial: Lukas Jalabert.

- Site naturellement en famille

J'ai commencé sur le site <https://www.naturellement-en-famille.fr/> où il y avait un problème au niveau des transactions. Je devais créer un appel API vers Stripe afin d'automatiser les transactions car pour l'instant, il fallait aller sur Stripe pour valider la transaction.

Ainsi, je devais modifier une fonction Hook pour remédier au problème. Yann, Raphaël et mon tuteur de stage m'ont aidé et j'ai fait des recherches internet afin de mieux comprendre ce qu'étaient les Hook et comment les utiliser. J'ai pu faire des tests avec les Hooks d'action mais je ne suis pas arrivée à régler le problème. Il me fallait accéder à la BDD du site pour récupérer l'ID de la transaction et pour cela, je devais demander au client. Pierre Champemont a donc finalement pris le relais.

- Application mobile des Halles et Marchés du Lez

Ensuite, mon tuteur de stage m'a proposé de travailler sur l'application mobile du Marché du Lez. Je me suis dit que ça pouvait être intéressant car je ne m'y connaissais pas du tout. Ma mission était d'afficher des messages par rapport à l'horaire actuel. Ex: 30 minutes avant la fermeture de la boutique, écrire: "nous

fermons dans 30 minutes". Il m'a donc donné un lien pour que je puisse voir comment créer une petite application avec Flutter. Ensuite, il m'a montré le code qui posait problème et m'a demandé avec Yann de voir s'il y avait des erreurs et s'il pouvait être amélioré/optimisé. J'ai eu un problème pour pouvoir visualiser le site mobile avec mon téléphone à cause d'une erreur ADB. J'ai essayé d'utiliser un émulateur mais ça n'a pas non plus fonctionné. Finalement, je n'ai pas vraiment eu le temps de me pencher sur le problème car mon tuteur de stage m'a lancé sur la formation Symfony peu de temps après.

- Site Synerg-in

Yann Morin m'a contacté pour que je mette en place des carrousels sur le site Synerg-in au niveau de la page des évènements. Pour chaque type d'évènements, il y avait des photos que j'ai dû mettre dans un carroussel, faire en sorte qu'elles soient toutes de la même taille, qu'elles ne soient pas floues, mettre les flèches sur le côté afin de pouvoir faire défiler les photos et faire en sorte que la page soit responsive (téléphone, tablette, ordinateur). Cela m'a pris du temps car il y avait beaucoup de photos à mettre, les modifications en utilisant le Visual Builder sur Wordpress faisaient ramer mon ordinateur et ayant un pc portable, je n'avais pas le rendu d'un ordinateur fixe. C'était aussi compliqué pour les flèches car sur téléphone, elles n'étaient pas visibles. J'ai finalement dû modifier beaucoup de css pour faire tout cela. Dès que je finissais, j'allais voir le commercial Lukas Jalabert afin de lui demander si tout était bon.

The screenshot shows the Synerg-in website's "Evenementiel Pro" section. At the top, there is a navigation bar with links: Accueil, Qui sommes-nous, Nos solutions, Marketplace, Évènementiel (which is highlighted in orange), Presse, and Contact. Below the navigation, there is a section titled "LANCEMENT DE PRODUITS" (Product Launch) featuring four images of event setups:

- A KIA car with a charging cable connected to it, labeled "KIA".
- A SOWOOD - ICADE booth with a mural and a sign for "KALITHEOS".
- A LIQUERIOUS BAR with a bartender and a counter.
- A HONDA motorcycle parked in front of a building.

Below these images, there is another section titled "POP-UP ÉPHÉMÈRES ET CRÉATEURS" (Temporary Pop-ups and Creators) which is currently empty.

Figure 26: Capture d'écran de la page "Evenementiel Pro" sur le site Synerg-in

- Petites missions données par le commercial Lukas Jalabert
  - Sur le site Synerg-in
    - Sur la page Contact, la maps s'affichait mais la localisation n'était pas bonne. J'ai cherché comment modifier cela avec la clef API mais je n'y suis pas arrivée. J'ai finalement affiché la map grâce à google maps et sa fonctionnalité de partage que j'ai pu intégrer à la page de contact.
    - Sur la page des pôles de l'agence, l'image tout en haut n'était pas responsive. L'image s'étirait en largeur. Donc plus l'écran était grand, plus l'image était étirée. J'ai réglé cela en ajoutant du css.
  - Sur le site du Marché du lez
    - Sur la page des événements. Lorsqu'on veut participer à un événement et donc acheter un ticket, les boutons "+" et "-" ne se voient pas. Il fallait donc changer leur couleur pour les voir. Ici aussi, il fallait faire du css.
    - Sur la page Restaurant. Les icônes sur la nouvelle page ne sont plus cliquables. Il faut donc aller dans le code afin de changer cela pour chaque page. Je ne suis pas arrivée à faire cette tâche car dès que je modifiais le code, je faisais planter le site. De plus, le temps pour télécharger tous les dossiers utiles était très long.

### 5.3. Discussion sur le travail effectué

Durant ce stage, j'ai passé beaucoup de temps à créer des pages .twig pour les maisons. En effet, après ma formation Symfony, Raphaël m'a proposé de commencer par créer les pages add, edit et show de la maquette. Au départ, la page add correspondait à la figure 27 et les pages show et edit étaient très ressemblantes. J'ai découvert comment nous intégrons les formulaires dans les pages .twig et comment les créer. Lorsque j'ai eu fini de créer toutes les pages avec les données que j'avais, j'ai créé des fichiers dans un dossier shared car il y avait beaucoup de redondance de code et je me suis occupée du css de la page ainsi que du responsive.

Ensuite, 3 semaines avant la fin de mon stage, le chef de projet Filipe m'a attribué une nouvelle tâche: celle de faire la page concernant les détails de la maison (figure 10). Pour cela, il fallait que j'utilise un champ "caractéristiquesPrincipales" de la table House. Ce champ représentait les produits et était en JSON. J'ai donc fait beaucoup de recherches internet afin de comprendre le JSON mais je n'ai pas trouvé

beaucoup d'informations concernant son utilisation avec Symfony et les formulaires. Mon tuteur de stage et Yann m'ont aidé mais on a finalement changé de méthode et on a mis ce champ en array afin de pouvoir le récupérer plus facilement. Cette disposition n'était finalement pas la bonne.

J'ai créé une table Produit avec un champ en JSON pour les caractéristiques. Ainsi, pour sauvegarder les données dans la BDD, j'ai mis 2 inputs cachés sur la page: un pour le type de produit, et un autre pour sa valeur. J'ai récupéré les informations en POST dans le controller et ai pu remplir les lignes en base de données. Le problème a été que j'écraserais à chaque fois la valeur du produit dès que je construisais un produit déjà existant car si le produit existait déjà, je faisais en sorte de ne changer que la valeur. De plus, l'id de la maison aussi se faisait remplacer.

Une autre tâche dont je devais m'occuper a été l'autocomplétion des champs au niveau de l'ajout d'un produit. Donc dès qu'un cliquait sur le bouton "ajouter produit" et qu'on rentrait le nom du produit, nous devions avoir des propositions. Pour faire cela, j'ai créé un tableau dans lequel j'ai récupéré toutes les valeurs de l'attribut "type" (qui à ce moment-là correspondaient au nom des produits) dans la table "produit". Ainsi, j'ai pu récupérer le tableau dans la page .twig puis dans le fichier js pour pouvoir afficher dynamiquement tous les noms des produits à chaque modification dans le champ concerné sur le formulaire.

Une semaine avant la fin de mon stage, j'ai eu une réunion avec le chef de projet Filipe. Il a décidé de changer la maquette suite à mes questions. Il veut que la page sur les détails de la maison (figure 10) soit la page d'ajout d'une maison et aussi la page d'édition d'une maison. Ensuite, il veut que la page d'ajout d'une maison (figure 27) soit la page permettant de visualiser une maison. De plus, il a ajouté de nouvelles caractéristiques qui ont entraîné la "reconstruction" de la base de données. J'ai ainsi modifié les tables "house" et "produit" ainsi qu'ajouter une table "panier" (figures 28 à 30).

J'ai ensuite adapté mes fonctions pour l'autocomplétion et la sauvegarde du JSON via les inputs mais Raphaël m'a ensuite prévenu que je pouvais faire tout cela avec une imbrication de formulaire. J'ai donc essayé de le faire en m'aidant de son code car il l'avait déjà fait auparavant. Ceci a permis de remplacer l'autocomplétion mais je n'arrivais pas à ajouter le produit dans le tableau. En effet, l'ajout se faisait via une fonction récupérée sur la documentation de Symfony et il fallait pour cela que je modifie une chaîne de caractères afin de récupérer et mettre en place les données utiles. De plus, les données n'étaient pas envoyées et donc je ne pouvais pas les recevoir dans la base de données. Je ne suis finalement pas arrivée à finir cette tâche avant la fin de mon stage.

## Conclusion

Pour conclure, j'ai participé à un projet nommé 'lucy' dans l'agence Synerg-in. Ceci m'a permis d'apprendre Symfony et d'utiliser mes connaissances afin de valider certaines tâches.

J'ai appris beaucoup de choses dans ce stage. Via les premières petites missions sur différents sites au début de mon stage, j'ai téléchargé Filezilla et appris à l'utiliser. J'ai pu apprendre à utiliser les Hooks via Prestashop et ce que c'était. J'ai aussi pu voir à quoi ressemblait le code d'une application mobile et créer avec Flutter une petite application permettant d'afficher des messages en fonction du nombre de clics faits. J'ai appris à utiliser le Visual Builder sur WordPress pour mettre en place un carrousel d'images.

Au niveau du projet 'lucy', j'ai pu utiliser tous les outils de Symfony permettant par exemple de créer des entités avec Doctrine et son système de migration. J'ai aussi appris quelques bases de jQuery et les ai utilisées pour faire des requêtes Ajax. J'ai découvert comment créer des formulaires en m'aidant du terminal. Je sais maintenant créer des pages .twig et des dossiers shared en utilisant la puissance de Symfony. J'ai aussi appris à bien utiliser les Repository pour chercher un objet précis sans utiliser de requêtes SQL. J'ai aussi pu avoir un aperçu du JSON.

J'ai pu voir comment s'organisait une vraie entreprise, que ça soit au sein d'un groupe informatique mais aussi avec le groupe de communication/marketing. J'ai appris à utiliser leurs outils. J'ai aussi remarqué qu'internet était mon meilleur ami et j'ai pris l'habitude de faire des recherches internet dès que j'avais un problème ou que je cherchais de la documentation. En effet, je n'avais pas du tout ce réflexe avant de commencer ce stage.

Ainsi, ce stage m'aura apporté beaucoup de connaissances, des réflexes utiles en entreprise et aussi une expérience professionnelle très enrichissante. La manière de travailler m'a aussi beaucoup plu.

Vu le 02/06/2021

# Bibliographie

## Bibliographie pour rédiger le rapport de stage:

[1]

« Accueil - Naturellement en famille ».

<https://www.naturellement-en-famille.fr/> (consulté le juin 08, 2021).

[2]

« Ce que l'on va faire ensemble 😊 ».

<https://learn.web-develop.me/courses/symfony-5-le-guide-complet-debutants-et-intermediaires/450396-a-propos-de-cette-formation-15-minutes/1529081-ce-que-l-on-va-faire-ensemble> (consulté le juin 08, 2021).

[3]

« Composer ». <https://getcomposer.org/> (consulté le juin 08, 2021).

[4]

« Dart programming language | Dart ». <https://dart.dev/> (consulté le juin 08, 2021).

[5]

« Flutter - Beautiful native apps in record time ». <https://flutter.dev/> (consulté le juin 08, 2021).

[6]

« Laragon - portable, isolated, fast & powerful universal development environment for PHP, Node.js, Python, Java, Go, Ruby. » <https://laragon.org/> (consulté le juin 08, 2021).

[7]

« Lucy | Connexion ». <https://preprod.inkweb.fr/login> (consulté le juin 08, 2021).

[8]

« Mailtrap - Safe Email Testing ».

<https://mailtrap.io/inboxes/1316135/messages/2191730459> (consulté le juin 08, 2021).

[9]

« SEO : Qu'est ce que le SEO (Search Engine Optimisation) ? »

<https://www.seo.fr/definition/seo-definition> (consulté le juin 08, 2021).

[10]

« Synerg-In | Agence digitale sur Montpellier, Lausanne et leurs environs ».

<https://synerg-in.com/> (consulté le juin 08, 2021).

[11]

« Visual Studio Code - Code Editing. Redefined ».

<https://code.visualstudio.com/> (consulté le juin 08, 2021).

## Bibliographie des sites sur lesquels je suis allée lors du stage:

[1]

« ● Lucy - Asana ». <https://app.asana.com/0/1200029743354404/board>

(consulté le juin 08, 2021).

[2]

« 1H pour créer votre première app Flutter | Purple Giraffe ».

<https://www.purplegiraffe.fr/p/1h-pour-creer-app-flutter> (consulté le juin 08,

2021).

[3]

« Accueil – My Project – Google Cloud Platform ».

<https://console.cloud.google.com/home/dashboard?project=empirical-state-3>

[10710](#) (consulté le juin 08, 2021).

[4]

« Adobe Experience League ».

<https://experienceleague.adobe.com/?lang=fr#home> (consulté le juin 08,

2021).

[5]

« Alsacréations : Actualités et tutoriels web, HTML, CSS, JavaScript ».

<https://www.alsacreations.com/> (consulté le juin 08, 2021).

[6]

« Apache Lounge ». <https://www.apachelounge.com/> (consulté le juin 08,

2021).

[7]

« Baptiste Donaux - Blog ». <https://www.baptiste-donaux.fr/> (consulté le juin 08, 2021).

[8]

« Base de donnée - MindMeister Mind Map ».

<https://www.mindmeister.com/fr/1689006593?t=XOTAefHYCZ> (consulté le juin 08, 2021).

[9]

« Bootswatch: Free themes for Bootstrap ». <https://bootswatch.com/> (consulté le juin 08, 2021).

[10]

« Boxtal, le meilleur comparateur d'envoi de colis | Boxtal ».

<https://www.boxtal.com/fr/fr/accueil> (consulté le juin 08, 2021).

[11]

« Buttonizer Community ». <https://community.buttonizer.pro/> (consulté le juin 08, 2021).

[12]

« Buy Plugins & Code from CodeCanyon ». <https://codecanyon.net/> (consulté le juin 08, 2021).

[13]

« CCM - Comment Ça Marche - Communauté informatique ».

<https://www.commentcamarche.net/> (consulté le juin 08, 2021).

[14]

« ChocoBLAST - Accueil ». <https://www.chocoblast.fr/> (consulté le juin 08, 2021).

[15]

« Cloud Stack Ninja ». <https://cloudstack.ninja/> (consulté le juin 08, 2021).

[16]

« Créez et développez votre boutique en ligne avec PrestaShop ».

<https://www.prestashop.com/fr> (consulté le juin 08, 2021).

[17]

« CSS Débutant : cours et tutoriels sur les feuilles de style CSS 1,2 et 3 ».

<http://css.mammouthland.net/> (consulté le juin 08, 2021).

[18]

« CSS ICON -- project by Wenting Zhang ». <https://cssicon.space/#/> (consulté le juin 08, 2021).

[19]

« Des parcours diplômants et des cours gratuits 100% en ligne - OpenClassrooms ». <https://openclassrooms.com/fr/> (consulté le juin 08, 2021).

[20]

« Développement Informatique- Plateforme d'apprentissage en ligne - Développement Informatique ». <https://developpement-informatique.com/> (consulté le juin 08, 2021).

[21]

« Developpez.com, le club des développeurs et IT Pro ». <https://www.developpez.com/> (consulté le juin 08, 2021).

[22]

« Font Awesome ». <https://fontawesome.com/> (consulté le juin 08, 2021).

[23]

« Form field setRendered throwing exceptions · Issue #36448 · symfony/symfony · GitHub ».

<https://github.com/symfony/symfony/issues/36448> (consulté le juin 08, 2021).

[24]

« Forum d'entraide PHPFrance - Page d'index ». <https://forum.phpfrance.com/> (consulté le juin 08, 2021).

[25]

« Free vector icons - SVG, PSD, PNG, EPS & Icon Font - Thousands of free icons ». <https://www.flaticon.com/> (consulté le juin 08, 2021).

[26]

« Git ». <https://git-scm.com/> (consulté le juin 08, 2021).

[27]

« GitHub: Where the world builds software · GitHub ». <https://github.com/> (consulté le juin 08, 2021).

[28]

« Heroku Dev Center ». <https://devcenter.heroku.com/> (consulté le juin 08, 2021).

[29]

« Home - Twig - The flexible, fast, and secure PHP template engine ». <https://twig.symfony.com/> (consulté le juin 08, 2021).

[30]

« Home - WPMeta ». <http://www.wpmeta.org/> (consulté le juin 08, 2021).

[31]

« Home | Laragon ». <https://forum.laragon.org/> (consulté le juin 08, 2021).

[32]

« HTML Color Codes ». <https://htmlcolorcodes.com/> (consulté le juin 08, 2021).

[33]

« JDN : E-business, FinTech, Big Data, IoT, tendances média, décideurs... » <https://www.journaldunet.com/> (consulté le juin 08, 2021).

[34]

« JSON pour les débutants ». <https://la-cascade.io/json-pour-les-debutants/> (consulté le juin 08, 2021).

[35]

« Learn to Code — For Free — Coding Courses for Busy People ». <https://www.freecodecamp.org/> (consulté le juin 08, 2021).

[36]

« Les 71 meilleurs plugins WooCommerce à se procurer d'urgence en 2021 ». <https://wpmarmite.com/woocommerce-plugins/> (consulté le juin 08, 2021).

[37]

« Les Docs – Formation Web, rétro-informatique, blog ». <https://lesdocs.fr/> (consulté le juin 08, 2021).

[38]

« MDN Web Docs ». <https://developer.mozilla.org/fr/> (consulté le juin 08, 2021).

[39]

« Medium – Where good ideas find you. » <https://medium.com/> (consulté le juin 08, 2021).

[40]

« Mysql installation on Linux and Windows - mysql tutorials | w3resource ».  
<https://www.w3resource.com/mysql/mysql-installation-on-linux-and-windows.php> (consulté le juin 08, 2021).

[41]

« Nouvelle-Techno.fr - Formateur indépendant - Crédit de sites web ».  
<https://nouvelle-techno.fr/> (consulté le juin 08, 2021).

[42]

« Online Mockup, Wireframe & UI Prototyping Tool · Moqups ».  
<https://app.moqups.com/> (consulté le juin 08, 2021).

[43]

« Outil de blog, plateforme de publication et CMS. | WordPress.org Français ».  
<https://fr.wordpress.org/> (consulté le juin 08, 2021).

[44]

« Packagist ». <https://packagist.org/> (consulté le juin 08, 2021).

[45]

« PatSaTECH - Best Custom Payment Gateway Integration ».  
<https://www.patsatech.com/> (consulté le juin 08, 2021).

[46]

« PHP: Hypertext Preprocessor ». <https://www.php.net/> (consulté le juin 08, 2021).

[47]

« phpMyAdmin ». <https://www.phpmyadmin.net/> (consulté le juin 08, 2021).

[48]

« Portfolio de Numa BORD : Développeur web freelance à Montpellier / Vézénobres / Alès ». <https://numa-bord.com/> (consulté le juin 08, 2021).

[49]

« Programmer Sought ». <https://www.programmersought.com/> (consulté le juin 08, 2021).

[50]

« QA Stack ». <https://qastack.fr/> (consulté le juin 08, 2021).

[51]

« React – Une bibliothèque JavaScript pour créer des interfaces utilisateurs ».

<https://fr.reactjs.org/> (consulté le juin 08, 2021).

[52]

« SourceForge - Download, Develop and Publish Free Open Source Software ».

<https://sourceforge.net/> (consulté le juin 08, 2021).

[53]

« Stack Overflow - Where Developers Learn, Share, & Build Careers ».

<https://stackoverflow.com/> (consulté le juin 08, 2021).

[54]

« Symfony, High Performance PHP Framework for Web Development ».

<https://symfony.com/> (consulté le juin 08, 2021).

[55]

« SymfonyCasts - PHP and Symfony Video Tutorial Screencasts ».

<https://symfonycasts.com/> (consulté le juin 08, 2021).

[56]

« The Most Popular WordPress Themes In The World ».

<https://www.elegantthemes.com/> (consulté le juin 08, 2021).

[57]

« The Web framework for perfectionists with deadlines | Django ».

<https://www.djangoproject.com/> (consulté le juin 08, 2021).

[58]

« Toutes vos ressources sur le développement web Apprenez à coder en HTML, CSS, JavaScript, Python, PHP... - Pierre Giraud ».

<https://www.pierre-giraud.com/> (consulté le juin 08, 2021).

[59]

« Tutoriels et Formations vidéos sur le développement web | Grafikart ».

<https://grafikart.fr/> (consulté le juin 08, 2021).

[60]

« W3Schools Online Web Tutorials ». <https://www.w3schools.com/> (consulté le juin 08, 2021).

[61]

« WayToLearnX - Plateforme e-learning ». <https://waytolearnx.com/> (consulté le juin 08, 2021).

[62]

« We see people answers ... » <http://www.quabr.com/> (consulté le juin 08, 2021).

[63]

« Weenesta | Création de site Internet et visuels sur-mesure à Caen ». <https://weenesta.com/fr> (consulté le juin 08, 2021).

[64]

« Welcome to Our Code World | Our Code World ». <https://ourcodeworld.com/> (consulté le juin 08, 2021).

[65]

« Wikilivres ». <https://fr.wikibooks.org/wiki/Accueil> (consulté le juin 08, 2021).

[66]

« Wikipédia, l'encyclopédie libre ».

[https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil\\_principal](https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal) (consulté le juin 08, 2021).

[67]

« WooCommerce - Sell Online With The eCommerce Platform for WordPress ». <https://woocommerce.com/> (consulté le juin 08, 2021).

[68]

« World Wide Web Consortium (W3C) ». <https://www.w3.org/> (consulté le juin 08, 2021).

[69]

« Zendevs - Le dev en toute zenitude \_/\_/\_ ». <https://www.zendevs.xyz/> (consulté le juin 08, 2021).

## Annexes

### 1. Pages des maisons prises sur la maquette

Page permettant de visualiser la maison sans les précisions:

The screenshot shows a user interface for managing houses. At the top right, a user profile for "FILIPE DE" (Super admin) is visible. The main title is "Ajouter maison". On the left, a sidebar contains icons for search, home, statistics, users, and settings. The main form includes fields for "Nom de la maison" (House name), "Description courte de la maison" (Short description), "Type" (Type) with options T1, T2, T3, T4, T5, and "Statut" (Status) with options "En cours" (In progress), "Attente BE" (Waiting BE), and "Validé par le BE" (Approved by BE). Below these, there's a section for "Caractéristiques principales" (Main characteristics) with fields for "Surface habitable en m<sup>2</sup>" (Habitable surface in m<sup>2</sup>), "Surface de plancher en m<sup>2</sup> (SDP)" (Floor area in m<sup>2</sup> (SDP)), and a price of "220 000 €". There are also fields for "Largeur" (Width) and "Longueur" (Length). At the bottom, there are fields for "Nombre de chambre" (Number of bedrooms), "Nombre de SDB" (Number of SDB), "Nombre cuisine" (Number of kitchens), and "Nombre séjour" (Number of living rooms).

**ou sont les dimensions des maisons dans ce fichier et les détails**

**valider ce qu'on garde**

Nombre d'étage	
0	
1	
2	

RDC

Entrée / Cuisine / Séjour \_\_\_\_\_ m<sup>2</sup>  
 WC \_\_\_\_\_ m<sup>2</sup>  
 Cellier \_\_\_\_\_ m<sup>2</sup>

 Déposer plan RDC

Étage 1

Chambre 1 \_\_\_\_\_ m<sup>2</sup>  
 Chambre 2 \_\_\_\_\_ m<sup>2</sup>  
 Chambre 3 \_\_\_\_\_ m<sup>2</sup>  
 Dégt \_\_\_\_\_ m<sup>2</sup>  
 WC \_\_\_\_\_ m<sup>2</sup>  
 Salle de bain \_\_\_\_\_ m<sup>2</sup>

 Déposer plan R+1

Type de maison	Toiture	Gamme
<input type="checkbox"/> Maison de plain-pied	<input type="checkbox"/> Mono pente	<input type="checkbox"/> Contemporaine
<input type="checkbox"/> Maison à étage	<input type="checkbox"/> 2 pans	<input type="checkbox"/> Moderne
<input type="checkbox"/> Maison avec sous sol	<input type="checkbox"/> 4 pans	<input type="checkbox"/> Traditionnelle
<input type="checkbox"/> Terrasse	<input type="checkbox"/> Toit terrasse	

Garage	Cuisine
<input type="checkbox"/> Garage intérieur	<input type="checkbox"/> Séparée <input type="checkbox"/> avec séjour

Détail maison  
(données fichier fiche recapitulative.pdf)



Déposer autre plan



Déposer photos

Ajouter documents



Ajouter étude thermique



Ajouter étude acoustique



Ajouter étude ....



Ajouter étude ...

Détail maison (données fichier fiche recapitulative.pdf)



Intervenants clé

Architecte M. Léon	BE SOCOTEC	Assureur VERSPIEREN
Constructeur SELVEA	Notaire Mme Vernet	

Commentaire

Enregistrer en brouillon
Publier

Figure 27: Capture d'écran de la page de visualisation d'une maison

## 2. Base de donnée pour les maisons

- table “panier”

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	id_maison_id	int(11)			Yes	NULL		
3	id_produit_id	int(11)			Yes	NULL		
4	<b>ratio</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
5	<b>etage</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		

Figure 28: Capture d'écran de la table “panier”

- table ‘produit’:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 	int(11)			No	None		AUTO_INCREMENT
2	<b>type</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
3	<b>prix_achat</b>	double			Yes	NULL		
4	<b>prix_vente</b>	double			Yes	NULL		
5	<b>unite_mesure</b>	varchar(10)	utf8mb4_unicode_ci		Yes	NULL		
6	<b>propriete</b>	json			Yes	NULL		

Figure 29: Capture d'écran de la table "produit"

- table "house"

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b> 	int(11)			No	None		AUTO_INCREMENT
2	<b>id_company_id</b> 	int(11)			No	None		
3	<b>name</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
4	<b>description</b>	varchar(1500)	utf8mb4_unicode_ci		Yes	NULL		
5	<b>type</b>	varchar(20)	utf8mb4_unicode_ci		No	None		
6	<b>status</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
7	<b>living_area_m2</b>	int(11)			Yes	NULL		
8	<b>floor_area_m2</b>	int(11)			Yes	NULL		
9	<b>width</b>	int(11)			Yes	NULL		
10	<b>length</b>	int(11)			Yes	NULL		
11	<b>room</b>	int(11)			Yes	NULL		
12	<b>kitchen</b>	int(11)			Yes	NULL		
13	<b>stays</b>	int(11)			Yes	NULL		
14	<b>area_water_m2</b>	int(11)			Yes	NULL		
15	<b>kitchen_area_m2</b>	int(11)			Yes	NULL		
16	<b>kitchen_desc</b>	int(11)			Yes	NULL		
17	<b>house_type_info</b>	int(11)			Yes	NULL		
18	<b>range_info</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
19	<b>form_info</b>	int(11)			Yes	NULL		
20	<b>pictures_url</b>	longtext	utf8mb4_unicode_ci		Yes	NULL	(DC2Type:array)	
21	<b>ground_floor_plan_url</b>	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
22	<b>slug</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
23	<b>is_valide</b>	tinyint(1)			No	None		
24	<b>prix_promoteur</b>	double			Yes	NULL		
25	<b>prix_particulier</b>	double			Yes	NULL		
26	<b>commentaire</b>	longtext	utf8mb4_unicode_ci		Yes	NULL		
27	<b>nbr_etage</b>	int(11)			Yes	NULL		

Figure 30: Capture d'écran de la table "house"

## 4eme de couverture

To conclude, I participated to a project called 'lucy' in the agency Synerg-in. This allowed me to learn Symfony and use my knowledge to validate some tasks.

I learned a lot from this internship. Through the first small missions on different sites at the start of my internship, I downloaded Filezilla and learned how to use it. I was able to learn how to use Hooks through Prestashop and what it is. I was also able to see what the code of a mobile application looked like and create with Flutter a small application allowing me to display messages according to the number of clicks made. I learned how to use the Visual Builder on WordPress to set up an image carousel.

For the 'lucy' project, I was able to use all Symfony tools to create entities with Doctrine and its migration system. I also learned some basics of jQuery and used them to do Ajax queries. I also learned to create forms using the terminal. I can now create .twig pages and shared folders using the power of Symfony. I also learned how to use Repositories well to search for a specific object without using SQL queries. I also got an overview of JSON.

I was able to see how a real company was organized, both within an IT group and with the communication/marketing group. I learned to use their tools. I also noticed that the internet was my best friend and I used to search the internet whenever I had a problem or looked for documentation. Indeed, I did not have this reflex at all before starting this internship.

Thus, this internship brought me a lot of knowledge, useful reflexes in a company and also a very enriching professional experience. I also liked the way of working.