A thick dark blue vertical bar is positioned on the left side of the page. Below it, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left towards the center of the page.

11 Juin-22 Août 2019

# Mise en place d'un site web

De Fabian TSCHIRHART pour la  
Licence 3ème année d'informatique  
à l'Université de Haute-Alsace

Année 2018/2019

Sous la tutelle de Dominique SCHMITT, professeur  
référant et Régis GHODBANE, gérant des Ecuries  
du Sundgau

## Remerciements

Je remercie Stéphane SCHWENKER pour nous avoir mis en contact avec Régis GHODBANE que je remercie également pour nous avoir accueillis dans son établissement et de nous avoir offert cette opportunité.

J'adresse mes remerciements à la communauté Symfony qui travaille gratuitement sur ce Framework et nous fournis donc un outil adapté.

Sans oublier ceux qui travaillent sur Doctrine et Twig.

Je remercie également Lior Chamla qui fournis des vidéos YouTube de qualité qui nous permettent de comprendre Symfony et de pouvoir partir sur une base de connaissances saine.

Ainsi que les discord communautaire d'entraide dans le domaine informatique « Not A Name », « Lior Chamla » et celui de « Grafikart » ainsi que leurs communautés pour m'avoir aidé sur certaines réflexions et erreurs.

Mr Dominique SCHMITT pour nous avoir accompagnée tout au long du projet en tant que professeur référent.

Pour finir je remercie l'intégralité de l'équipe de travaux, eux même stagiaire pour avoir travaillé sur ce projet avec moi, d'avoir tout donner même quand ce n'était pas facile, pour les petites prises de tête, les fou rire et le travail effectué ensemble et aussi pour m'avoir supporté tout au long du projet je remercie :

Hugo FREYBURGER notre chef de projet pour avoir endossé se rôle qui lui donnait du travail supplémentaire, pour avoir fait des rapports journaliers et rédiger le dossier de conception.

Laurine HUG et Julien Weissenberger pour m'avoir accompagnée sur le projet Symfony.

## Sommaire

<b>INTRODUCTION</b>	<b>3</b>
<b>PRESENTATION DE L'ENTREPRISE</b>	<b>4</b>
<b>STAGE</b>	<b>5</b>
<b>Dossier de conception</b>	<b>5</b>
Présentation Générale	5
<b>Analyse et conception</b>	<b>6</b>
Cas et Scénario d'utilisation	6
Spécificités Techniques et Technologies	6
<b>Mise en contexte</b>	<b>6</b>
<b>Présentation du Framework Symfony</b>	<b>7</b>
Architecture	7
Fonctionnement	9
Le cmd	12
Les bundles supplémentaires	12
Les Rôles	13
<b>Mes formations externes</b>	<b>13</b>
<b>Les problèmes durant le stage</b>	<b>14</b>
<b>Modèle conceptuelle des données</b>	<b>14</b>
Les entités	16
Les relations(liens) :	16
<b>Mon travail</b>	<b>17</b>
Utilisateur	17
Administrateur	18
Mise en ligne du site	20
Interaction	20
<b>Spécificités Techniques et Technologies qui me concernent</b>	<b>21</b>
Editeur de texte	21
Serveur virtuel	21
Partage d'information	21
Echange de données	21
Mise en ligne des fichiers	21
<b>CONCLUSION</b>	<b>22</b>
<b>BIBLIOGRAPHIE</b>	<b>23</b>
<b>RESUME</b>	<b>24</b>

## Introduction

L'entreprise les Ecuries du Sundgau dans un objectif de promouvoir son activité a eu comme idée de faire un site internet alors qu'elle avait à la base seulement un groupe Facebook pour attirer de la clientèle, c'est pourquoi l'entreprise a fait appel à des stagiaires.

Par l'intermédiaire de Stéphane nous avons été mis en contact avec le client, l'idée de faire un site internet nous paraissait sympathique donc nous avons accepté cette offre.

## Présentation de l'entreprise

Attention cette présentation est commune aux 4 rapports.

L'entreprise est un centre équestre dont le propriétaire est le seul employé. Lors des événements, il fait appel à des bénévoles en provenance d'une association toutefois, la principale activité du centre reste l'apprentissage de l'équitation à de jeunes cavaliers. Le centre regroupe des familles de toute l'Alsace et même de l'étranger.

L'entreprise ne dispose d'aucun service informatique à proprement parler puisque le gérant ne dispose pas des compétences nécessaires pour un tel service. De fait, il a été nécessaire d'assurer la facilité d'utilisation de l'ensemble du projet par n'importe qui. Le projet donné a été, à notre connaissance, le projet le plus conséquent des Ecuries du Sundgau pendant la durée du stage bien qu'il n'ait pas été en rapport direct avec l'activité du centre équestre.

Puisque nous étions les seuls à avoir suivi une formation informatique, nous représentions l'ensemble du service informatique. Celui-ci était donc constitué de quatre membres, Laurine HUG, Hugo FREYBURGER, Julien WEISSENBERGER et moi-même, Fabian TSCHIRHART. Il sera aussi dissout une fois le stage terminé d'où la nécessité d'assurer la simplicité d'utilisation du site et de l'application. Nous resterons néanmoins à disposition de Mr. GHODBANE pour toute question ou correction nécessaire qui nous aurait échappée.

## Stage

Le stage est décrit par le dossier de conception ci-dessous écrit par notre chef de projet Hugo FREYBURGER.

### Dossier de conception Présentation Générale

#### Projet

- Description : Réalisation d'une application Android et d'un site web relié via Symfony ayant comme objectif une meilleure mise en avant des écuries du Sundgau et de ses événements. Le site devra être scindé en deux parties, une pour les clients et une pour l'administrateur (potentiellement une troisième partie concernant les bénévoles)
- Utilisateurs : Les clients des écuries d'une part et le gérant d'autre part (bénévole en troisième part)
- Type :  

Site Web	Application Android
----------	---------------------
- Date de livraison souhaité : 23 août 2019

#### Equipe

- Chef de projet : Hugo FREYBURGER
- Graphiste et Développeur Général : Julien WEISSENBERGER
- Développeur Android : Hugo FREYBURGER
- Développeur Web Admin : Laurine HUG
- Développeur Web Client : Fabian TSCHIRHART
- Testeur : Tout le monde

#### Planning

	10-juin	17-juin	24-juin
Hugo FREYBURGER	Maquettage Android	Prototypage Appli Android	
Laurine HUG	Maquettage Admin	Prototypage Site Admin	
Fabian TSCHIRHART	Maquettage Client	Prototypage Site Client	
Julien WEISSENBERGER	Maquettage Général	Prototypage Site Général et Visuel	

	01-juil	08-juil	15-juil	22-juil
Hugo FREYBURGER	Mise en ligne et mise en production	Debug		Congé
Laurine HUG				
Fabian TSCHIRHART				
Julien WEISSENBERGER				

	29-juil	05-août	12-août	19-août
Hugo FREYBURGER	Ajout de fonctionnalités		Debug	
Laurine HUG				
Fabian TSCHIRHART				
Julien WEISSENBERGER				

## Analyse et conception

### Cas et Scénario d'utilisation

- Utilisation du site par des clients lambda :  
Le client doit pouvoir accéder à la liste des événements passés, en cours et à venir pour s'y inscrire en tant que participants. Il doit aussi voir les tarifs d'inscription aux écuries, les informations de contact et la localisation du centre.
- Utilisation du site par l'administrateur :  
L'administrateur doit pouvoir organiser des événements, retirer des clients qui se désistent et doit pouvoir changer ses tarifs aisément (importation d'un document Word par exemple) ainsi qu'éditer aisément les informations de contact et les textes présent sur le site.
- (Facultatif) Utilisation du site par des bénévoles :  
Un bénévole actif peut s'inscrire à un événement pour apporter sa contribution. De ce fait, il doit pouvoir indiquer sa participation depuis la page d'un événement dans un créneau horaire disponible, ainsi que voir la répartition des bénévoles inscrits à un tel événement dans ses créneaux horaires correspondants.
- (Facultatif) Utilisation du site par l'administrateur des bénévoles :  
L'administrateur des bénévoles peut accéder à la page d'un événement pour déterminer le nombre de bénévoles nécessaires à un événement donné, ainsi que les dates et créneaux horaires qui seront aménagés pour eux. Il est aussi capable de retirer d'un événement un bénévole qui souhaiterait se désister.
- Utilisation de l'application :  
Purement informative, l'utilisateur doit se connecter pour ensuite voir un flux d'actualité personnaliser (en fonction de son galop). Une redirection vers le site web doit être aisément visible pour permettre à la personne de s'inscrire aux événements. L'application doit aussi envoyer des notifications lorsqu'un nouvel événement qui concerne l'utilisateur est créé.

### Spécificités Techniques et Technologies

- Framework Symfony 4.3, éditeur de texte Visual Studio Code, GitHub, Android Studio, Notepad++, WampServer, Paint.net
- 

## Mise en contexte

Le cahier des charges à évoluer au fur et à mesure du projet, le client pouvait être disponible de temps à autre pour regarder l'évolution du travail émettre des idées, des suggestions, des choses à changer et ainsi il y a eu des légères modifications visuelle et fonctionnelles au cours du projet sans trop toucher au cas et scénario d'utilisations

Nous étions 3 sur la partie site internet, une personne sur le visuel du site et une personne sur le code ainsi que moi-même. La 4<sup>ème</sup> personne s'occupait de l'application mobile.

Nous étions tous au même endroit pour travailler, rendant la communication plus simple

## Présentation du Framework Symfony

Nous avons utilisé le Framework Symfony qui nous permet de travailler plus rapidement, il permet un gain de temps considérable car il fait pleins de choses à notre place notamment grâce au lignes de commandes et à sa structure préfète, de plus l'installation de librairie appelé « bundles » est très simple dans la version que nous utilisons qui est la version 4.3.

Il permet de travailler de façon structurer et est très bien maintenue.

Symfony utilise Twig et Doctrine ce sont deux choses qui ne sont pas créé et maintenue par eux.

Twig est une surcouche HTML, Doctrine est une surcouche du côté Base de Données.

Ses 2 composantes sont du coup utilisé par d'autre Framework et ont leur documentation propre a eu même.

## Architecture

Par ailleurs Symfony inclue le schéma « Modèle-vue-contrôleur »

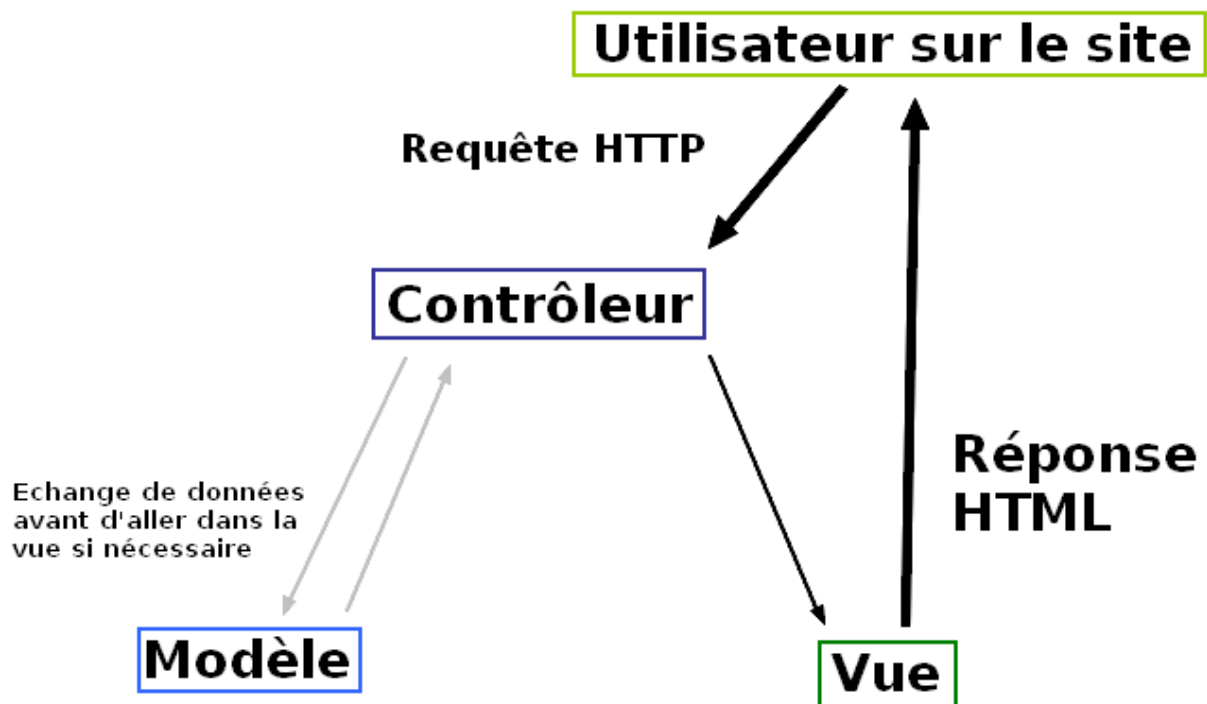


Figure 0-1

Ce schéma montre le fonctionnement global du site à partir de l'architecture du site Symfony qui permet de séparer les données, de la vue et du contrôleur.

Le modèle ici représente la base de données.



Le contrôleur représente l'accès au chemin du site et la vue représente ce que voit l'utilisateur, il doit y avoir le moins de traitement possible dans une vue hors formulaire (car les données viennent de l'utilisateur) qui sont protégées avec Symfony, tout ce qui est traitement doit surtout être fait dans le contrôleur et autres codes en PHP (orientée objet).

Voici ci-dessous l'arborescence d'un projet Symfony lorsque nous l'avons créé en ligne de commande à l'aide de composer excepté les fichiers GitHub et dossier Translations.

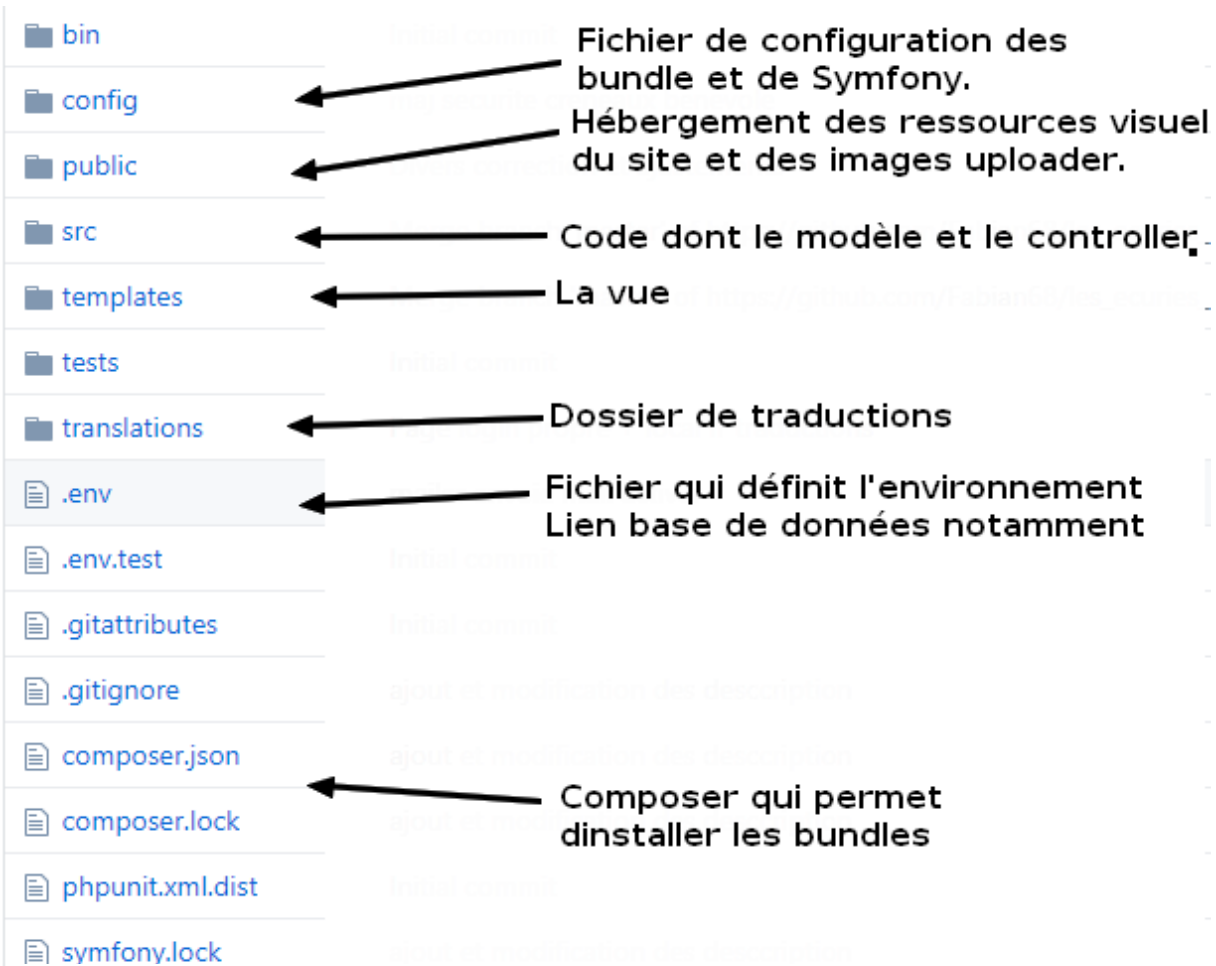


Figure 0-2

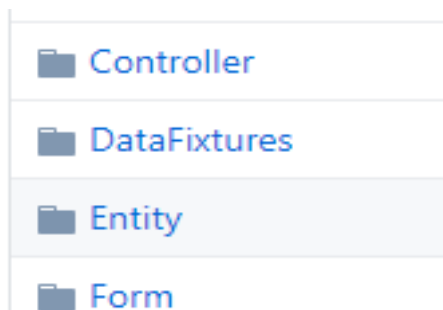


Figure 0-3

Voici une partie du dossier **src**, il contient un sous dossier avec le contrôleur, le modèle (Entity) et les formulaires, il contient d'autres dossier dont je ne parlerai pas.



Et voici le dossier des vues, ils sont séparés pour gagner en visibilité et éviter d'avoir une longue liste de vue, le fichier base.html.twig donne un visuel que l'on peut ensuite hériter sur tous les pages, comme le menu du haut.

Figure 0-4

Rentrons un peu plus dans les détails.

### Fonctionnement

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

{% block body %}
/* ici du code d'affichage */
{% endblock %}
```

Figure 0-5

Voici à quoi ressemble un fichier .html.twig la première ligne montre que l'on hérite du visuel de base.html.twig . Ensuite on redéfinit un affichage dans des blocs.

```

/**
 * @ApiResponse(normalizationContext={"groups"={"read"}})
 * @ORM\Entity(repositoryClass="App\Repository\GalopsRepository")
 */
class Galops
{
    /**
     * @Groups("read")
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @Groups("read")
     * @ORM\Column(type="integer")
     */
    private $niveau;

    /**
     * @ORM\ManyToMany(targetEntity="App\Entity\Event", mappedBy="galops")
     */
    private $evenements;

    /**
     * @ORM\OneToMany(targetEntity="App\Entity\Utilisateur", mappedBy="galop")
     */
    private $utilisateurs;

    public function __construct()
    {

```

Figure 0-6

Voici ici la structure d'une entité, ici galops, l'id et le niveau sont les propriétés de l'entité ensuite évènement et utilisateurs sont des relations, plus bas on a accès à des méthodes pour ajouter des utilisateurs et évènement qui sont créés automatiquement, les getters et setters sont aussi créés lors de la création d'une entité en ligne de commande.

Grâce à l'annotation en commentaire on forme notre modèle, en utilisant une ligne de commande on peut créer un fichier qui fait les requêtes de création de table et de relation juste à partir de nos entités, cela ne prend en compte que les champs annotés ORM.

La plupart du temps on veut pouvoir stocker ses données, dans Symfony on crée une base de données en se basant sur le modèle conceptuel des données au lieu du modèle logique, ainsi on est parfois limité, mais cela facilite la fabrication de la base de données ainsi malheureusement avec Symfony on perd en efficacité pour gagner en facilité et en gain de temps c'est pourquoi quand les bases de données sont trop grandes les développeurs ont tendance à utiliser d'autres moyens tels que l'utilisation de fichiers Json au lieu d'une base de données.

```

class DatesEvenementsType extends AbstractType
{
    /**
     * @param FormBuilderInterface $builder
     * @param array $option
     */
    public function buildForm(FormBuilderInterface $builder, array $option)
    {
        $builder
            ->add('dateDebut', DateTimeType::class, [
                'data' => new \DateTime("now"),
                'label' => 'Jour et horaire de début',
            ])
            ->add('dateFin', DateTimeType::class, [
                'data' => new \DateTime("now"),
                'label' => 'Horaire de fin',
            ]);
    }
}

```



Figure 0-8

Voici la structure d'un formulaire celui-ci permet de rajouter les deux dates correspondant à notre entité date et heure. Ce formulaire est ensuite traité dans les contrôleurs pour traiter ce que l'on reçoit en données et aussi dans la vue pour déterminer sont affichage sur la page. Et voici sont visuel ce dessous :

- Jour et horaire de début

14 ▾	août ▾	2019 ▾
18 ▾	:	17 ▾

Horaire de fin

14 ▾	août ▾	2019 ▾
18 ▾	:	17 ▾

Figure 0-7

Comme dit précédemment on envoie parfois un formulaire, un des formulaires les plus connue est celui de l'inscription, un formulaire a pour but de recevoir des données dans des champs qui peuvent de différents types, allant du type texte, entier, date etc.

```

/**
 * @Route("/lecentre", name="facilities")
 */
public function facilities()
{
    return $this->render('/general/facilities.html.twig', [
        'controller_name' => 'GeneralController',
    ]);
}

```

Figure 0-9

Voici une fonction du contrôleur, en commentaire en vert on le chemin d'accès en français et juste à droite son nom utiliser dans le code.

Ici on ne fait aucun traitement et on affiche directement la page avec le nom du fichier donné en argument. C'est le traitement le plus basique qu'il soit.

```
/**  
 * @Route("/evenement/{id}", name="event")  
 */  
public function event($id, Request $request, ObjectManager $manager, \Swift_Mailer $mailer)  
{
```

Figure 0-10

Les fonctions peuvent avoir besoin d'un identifiant ici on veut accéder à un événement particulier donc on lui donne un identifiant, une fois l'identifiant donné on fait un traitement pour trouver l'événement qui a cet id ainsi que d'autres traitements pour traiter les formulaires et l'affichage se fait à la fin.

Le fonctionnement reste toujours le même, c'est dans des cas particuliers où l'on doit chercher comment faire tels traitements mais ce que j'ai cité juste au-dessus ne change globalement pas.

Le fonctionnement général, quand on crée un contrôleur à l'intérieur on va principalement y mettre des fonctions qui redirigent à une « route » c'est-à-dire un chemin d'accès au site comme par exemple « /ma\_page » et si une personne va sur l'adresse `monsite/ma_page` il exécute donc le code qui est dans cette fonction. La fonction s'interrompt avec une fonction d'affichage qui lui dit quel page afficher, avec parfois des données, une ou plusieurs formulaires.

## Le cmd

Le cmd est un terminal dans lequel on peut rentrer des lignes de commandes qui font appel à des petits programmes avec Symfony son rôle est crucial c'est pourquoi nous utilisons un éditeur de texte avec un terminal intégré.

Grâce à ce cmd et au fichier console qui est dans le dossier bin on a accès à plusieurs lignes de commande liées à Symfony si nous sommes bien dans la racine du projet dans l'invite de commande. Ces commandes permettent notamment de créer une entité avec seulement les informations essentielles ensuite Symfony se charge de mettre les méthodes, les annotations, quasiment tout en bref. On peut faire aussi des formulaires qui se basent sur une entité.

Avec une commande cette fois-ci très importante on crée la base de données qui se base sur les classes entités, dans phpMyAdmin qui est le site où l'on a accès à notre base de données.

Les bundles peuvent être installés de façon automatique grâce aux lignes de commandes.

## Les bundles supplémentaires

Au cours de notre projet nous avons installé plusieurs bundles grâce à Composer qui permet la création du site à la base avec les bundles de base pour pouvoir réaliser des choses plus simplement.

Voici la liste des bundles que j'ai installé gérer et utiliser :

VichUploader	Permet l'upload d'image et de toute sorte de fichier de façon simplifié.
Fixtures	Permet de crée de fausses données pour tester la base de données.
Swift Mailer	Permet l'envoi de mail.
Api Platform	Permet de crée et afficher des fichiers Json a des liens particuliers du site.
LiipImagineBundle	Permet d'optimiser l'affichage des images
Apache-Pack	Petit bundle qui permet le déploiement du site sur un serveur Apache.

LiipImagineBundle a été utiliser et gérer par Julien par la suite.

## Les Rôles

Rôle	
ROLE_USER	Utilisateur normal
ROLE_ADMIN	Administrateur
ROLE_ADMIN ASSO	Administrateur association
ROLE_BENEVOLE	Utilisateur bénévole
IS_AUTHENTICATED_ANONYMOUSLY	Utilisateur non connecté au site

Lors de l'inscription au site internet nous avons tous le rôle « ROLE\_USER » qui nous est attribué. Nous avons le choix entre rôle multiple ou rôle unique, le rôle unique a été choisis, plus précisément une personne ne peut avoir qu'un seul rôle et un rôle peut hériter des droits d'un autre rôle.

Tous les rôles héritent du rôle user de base ici sauf celui des personnes non connectées qui n'est par ailleurs jamais stocker dans la base de données en toute logique.

## Mes formations externes

Pour ma part je me suis formé au préalable notamment grâce à la chaîne YouTube Lior Chamla qui présente le Framework et la documentation avant de commencer le projet ce qui m'a permis de pouvoir commencer avec de bonnes notions de base.

Au cours du projet j'ai découvert la chaîne YouTube Grafikart qui m'a appris quelques petites choses en plus durant le stage, notamment l'utilisation d'images sur le site et un peu la mise en ligne du site.

Dès lors au cours du projet j'ai réussi à me débrouiller par moi-même grâce à la documentations Symfony, Twig et Doctrine ainsi qu'une aide en temps réel sur une application de tchat en direct du nom de Discord ou j'ai pu poser des questions, aux furs et a mesures j'ai beaucoup moins ressenti le besoin de poser de question.

## Découverte

Au cours du projet j'ai commencé à utiliser « PHP tester » qui me permettait de tester des courtes fonctions de PHP sans à avoir à modifier mon code, sans faire tourner la machine virtuelle pour pouvoir tester car PHP est un langage qui s'exécute coté serveur et du coup nécessite un serveur pour être exécuter c'est pourquoi nous avons à faire un serveur virtuel

pour tout tester et ça prenais parfois 3 minutes pour charger une page, donc cet découverte m'a fait gagner beaucoup de temps sur les tests de fonctions PHP.

### Les problèmes durant le stage

Durant le stage il y a eu divers problèmes dont pas mal auxquels on ne s'attend pas

#### Problèmes liés au lieu de travail

Le premier problème auquel je pense ce sont les mouches très nombreuses sur notre lieu de travail je pense que je n'ai pas besoin de préciser pourquoi.

Le froid ou la chaleur due au travail en extérieur étant donné que l'on travaillait en extérieur on pouvait vite se sentir assommer par la chaleur comme si le toit nous tombait dessus, ou refroidit par les courants d'air lorsqu'il ne faisait que maximum 19 degré dans la journée.

Au cours du stage j'ai commencé à avoir des douleurs au dos liée au fait que l'on n'avait pas siège mais un simple banc.

Les problèmes de connexion internet et les déconnexions intempestive de mon ordinateur du réseau était un problème qui fu le problème qui m'a le plus dérangé quant à côté les autres font de belles anecdotes ici cela m'empêchait de pouvoir envoyer mes données via GitHub et de faire des recherches sur internet.

Etant donné que nous étions dans pas dans une entreprise informatique nous devions ramener nos propres ordinateurs, or je n'ai qu'un ordinateur portable à faible performance qui n'est pas fait pour sa normalement du coup il était un peu lent c'est pourquoi j'ai toujours essayé de trouver des solutions simple et efficace pour éviter d'avoir à perdre du temps.

#### Problèmes liés au fait qu'il n'y est pas de service informatique à notre charge

Ne pas avoir de service informatique et de tuteur informatique est un problème pour la vérification du travail réalisé du suivis et de l'apprentissage néanmoins cela nous a permis de nous forger en autonomie et auto apprentissage.

Ajoutons à cela parfois le stress du fait que l'on choisit les directions nous-même alors que c'est une première pour nous ce projet, du coup on avance un peu dans l'inconnue on ne sait pas s'y on s'y prend de la bonne manière c'est pourquoi on peut regretter de ne pas avoir eu de tuteur.

### Modèle conceptuelle des données

Voici la base de données ci-dessous, Hugo m'aidait pour les réflexions sur le schéma, je me suis occupé de faire toutes les entités sur le site excepter Description qui a été fait par une autre personne.

Ce schéma représente la modélisation des données dans notre site internet.





## Les entités

L'entité EVENEMENTS fut la première créée avec DATES\_HEURES et GALOPS.

L'entité GALOPS n'a jamais changer il représente la liste des Galops possible c'est-à-dire les niveaux d'équitation que l'on peut avoir.

L'entité DATES\_HEURES na pas changer non plus.

L'entité EVENEMENT comportait les nombres de bénévoles mais on a eu une meilleure idée de conception et on l'a déplacé autre part, on a rajouté Divers et Repas en cours de route.

Son titre, son texte et les tarifs ainsi que le nombre de participant n'ont pas changer depuis le début.

Par la suite on a ajouté l'entité UTILISATEUR avec son Nom prénom mail et mot de passe, après divers tests j'ai rajouté des informations en plus tels que son Adresse, sont numéro de téléphone et sa date de naissance. Puis encore plus tard on rajouter une image et un champ « mis à jour le ».

Pour finir il a fallu stocker des valeurs que l'on a besoin si on a oublié son mot de passe et pour vérifier le mail ce sont des jetons qui servent a pour accéder à des liens temporaires inaccessible sinon que l'on appelle « Token ».

Mail vérifier sert à savoir si le mail est vérifié.

La simulation ternaire permet de connaitre quel moyen de paiement a été utiliser à quel évènement par quel utilisateur et on a une liste des moyens de paiement dans la table MOYEN\_PAIEMENTS.

CRENEAUX\_BENEVOLES est presque identique à l'entité DATE\_HEURES, n'étant pas pour le même type d'utilisation on en a fait une nouvelle entité elle possède le nombre de bénévoles en plus.

Pour avoir des images et des vidéos dans un évènement on a rajouté deux entités qui y correspondent.

Deux entités n'ont aucun lien, Description et PDF.

## Les relations(liens) :

Un utilisateur a forcément un Galops et il ne peut en avoir qu'un seul/un galop ne possède pas forcément d'utilisateur mais peut en avoir plusieurs

Un utilisateur ne participe pas forcément à un évènement et donc ne paient pas forcément pour un évènement mais peut en participer a plusieurs et donc payer à plusieurs évènements.

Un évènement peut avoir 0 participant comme plusieurs participants.

Un utilisateur ne mange pas forcément durant un évènement mais peut manger à plusieurs évènement/ un évènement n'a pas forcément de personnes qui y mangent mais ils peuvent en avoir plusieurs.

Un utilisateur n'est pas forcément bénévole pour un créneau et il peut l'être pour plusieurs créneau/un créneau n'a pas forcément de bénévole et peut en avoir plusieurs

Un évènement peut avoir 0 créneau bénévole associer comme plusieurs créneaux bénévoles,

Un créneau bénévole est associé à un seul et unique évènement.

Un évènement peut avoir plusieurs images, vidéo et dates, mais ils ne sont associés à qu'un seul et unique évènement.

## Mon travail

### Utilisateur

Ici je vais parler de tout ce que les utilisateurs peuvent faire sur le site avec ce que j'ai fait.

#### **Utilisateur non connecté :**

Un utilisateur non connecté a accès à une page tarif avec un PDF à afficher, j'ai moi-même travaillé sur cette fonctionnalité, l'utilisateur voit le PDF comme s'il l'avait ouvert avec un logiciel tel qu'Adobe Reader et lui permet ainsi de pouvoir l'imprimer en un seul clic par exemple.

L'utilisateur a accès à une page connexion, sous cette page il y a un lien qui redirige vers la page inscription si l'utilisateur n'est pas inscrit, et juste en dessous il y a une fonctionnalité mot de passe oublié.

Pour commencer la page connexion est une page assez particulière, je me suis beaucoup inspirée de la documentation Symfony qui fournit un « Guard Authenticator » c'est-à-dire une classe qui donne une couche de sécurité et de vérification lors de la connexion d'un utilisateur, il se charge de faire du travail pour nous.

Il est considéré comme un service que l'on configure dans le fichier de config security.yaml. Cette partie correspond comme deux gouttes d'eau à la documentation car on se connecte de façon très classique sur le site, il n'y avait pas besoin de modifier ce qu'il existe déjà et qui est utilisé la majeure partie du temps.

Bien sûr au préalable il avait fallu faire une entité Utilisateur, celle-ci est particulière car elle hérite d'une interface qui s'appelle « User Interface » et grâce à cela on n'a pas besoin de gérer une grosse partie de l'aspect sécurité.

Au niveau de l'inscription ici on n'a pas juste à mettre le mail et le mot de passe donc pour le coup ici il n'existe rien de pré fait pour nous, de ce fait ici je me charge de faire un formulaire où on demande les informations pertinentes de l'utilisateur, pour l'adresse on aurait pu séparer en 3 champs tels que, Code Postal Adresse et Ville, à la place nous avons fait un champ Adresse complète.

Il y a le mot de passe qui est masqué et qui demande une confirmation nécessaire, il y a d'autres contraintes sur d'autres champs qui doivent être valides. Une fois ce formulaire rempli et valide la personne reçoit une notification comme quoi elle est inscrite et qu'elle reçoit un mail de validation, ce mail est envoyé grâce au bundle Swift Mailer et à l'aide du mail que j'ai créé sur l'hébergeur qui s'occupe d'envoyer le mail avec le texte et l'URL de validation de mail. Si l'utilisateur clique sur ce lien, son mail est validé et on sait que la personne n'utilise pas un faux mail ou celui d'une autre personne.

La connexion est dotée d'un bouton « se souvenir de moi » qui permet de stocker un cookie qui permet à l'utilisateur de rester connecté tant que le cookie existe, dans le cas contraire en fermant son navigateur il doit se connecter à nouveau.

#### **Utilisateur connecté :**

Un utilisateur connecté peut s'inscrire à un évènement, s'il s'inscrit avec le moyen de paiement sur place, il n'y a rien de plus à faire et il reçoit directement une notification comme quoi il est inscrit, le second traitement, s'il paie par prélèvement le ramène sur une autre

page , nous avons besoin pour cela de conserver des informations dans une variable de session c'est la seule fois du projet où j'utilise cela , la personne doit rentrer son nom et prénom ainsi qu'un des 3 tarifs qui doit absolument être valide , si le formulaire est valide la personne est inscrite et reçoit une notification comme quoi elle est inscrite à l'événement et Régis reçoit un mail comme quoi la personne est d'accord pour payer par prélèvement et voit sa signature ainsi que son tarif.

De plus l'inscription n'est plus possible si la date de la veille de l'événement à midi est dépassée ou si tout simplement la personne n'est pas connectée. Les autres contraintes liées à l'inscription ont été faites par d'autres personnes.

La page affiche le nombre de places disponibles.

Des images de l'événement sont affichées sur l'événement et l'accueil tout ceci est géré par mes entités images qui Julien réutilise pour les organiser à souhait ainsi toutes les images visibles sur le site sont travaillées postérieurement par moi et testées sur un affichage basique de l'image et Julien n'a plus qu'à les organiser selon ses besoins du client.

L'administrateur peut supprimer une image et changer l'image principale avec une autre.

Le bundle VichUploader nous a permis de faire la gestion d'image de façon simple.

L'utilisateur connecté a accès à son profil, sur son profil elle a accès à toutes ces informations personnelles ainsi que son image de profil, il y a plusieurs boutons disponibles sur son profil.

Il a accès à un bouton pour modifier les informations de son profil avec un formulaire qui reprend les informations de son profil.

Il a accès à un bouton pour modifier son mot de passe qui est bien sûr à part de la modification de profil.

Un autre bouton n'apparaît que tant que son compte n'est pas vérifié pour avoir la possibilité de renvoyer un mail de vérification celui-ci ne redirige vers aucune page (en réalité sur la même page).

Le dernier bouton auxquels il a accès c'est un bouton pour devenir bénévole, celui-ci permet de s'inscrire en tant que bénévole, dès lors on appuie sur ce bouton on reçoit le rôle bénévole, à tout moment l'utilisateur peut changer d'avis et choisir de ne plus être bénévole avec le même bouton, son fonctionnement est le même que le bouton précédent, il ne redirige sur aucune autre page.

Seulement un utilisateur normal peut avoir le rôle bénévole car on ne veut pas que les administrateurs ne puissent changer de rôle.

## Administrateur

Un administrateur sur son profil possède un bouton qui permet la création d'événement, je me suis occupée des listes des dates, images et vidéos sur ce formulaire, ceci demandait de réutiliser d'autres formulaires dans ce formulaire, car en même temps que l'on crée cet événement on crée des dates, images et vidéos qui lui seront associées, nous avons déjà parlé des images. Les vidéos quand elles ne sont que des liens YouTube, le lien est traité pour correspondre à un lien utilisable dans un objet.

Sur la page d'un événement en particulier j'ai rajouté l'option de désinscription d'un utilisateur que j'ai rajoutée dans la liste d'inscrits faite par une autre personne.

La même option est rendue possible pour les inscrits bénévoles.

Il y a un bouton pour supprimer l'évènement, je rajoute que pour toute création et suppression j'ai rajouter une optimisation en cours de route dans le code qui permet la suppression et la persistance des entités en cascade donc si je supprime un évènement ces dates associer sont directement supprimer sans que je n'aie à le faire.

Un autre bouton quant à lui sert à avoir une liste des participants sur un fond blanc dans un tableau, ce tableau fait en sorte que l'administrateur puissent l'imprimer sans mise en forme derrière et d'avoir un récapitulatif clair des personnes inscrit dans l'évènement.

Si on retourne sur le profil de l'administrateur il y a un bouton presque identique à la création d'évènement mais celui-ci c'est pour des évènement divers, ici il ne met qu'un titre, un texte, des images et des vidéos, et qu'une seule date, ce formulaire sert à faire des évènements ou les personnes ne peuvent pas s'inscrire mais juste regarder les photos et vidéo qui sont selon le client des photos d'anciens évènement.

Je vais passer les boutons sur lesquels je n'ai pas travailler, le prochain est le bouton pour modifier les PDF, a la base on pensait avoir plusieurs PDF mais au final pour le moment on n'a besoin que du PDF « Tarif.pdf » donc ce bouton redirige vers une page d'upload de PDF assez sommaire.

Le dernier bouton dont je me suis occupé est la liste des utilisateurs ou j'ai fourni une liste des utilisateurs avec quelques informations pertinentes. Dans cette liste il peut supprimer un utilisateur s'il le veut, je précise qu'il y a une confirmation nécessaire avec un popup qui apparait pour tout ce qui est des suppressions.

Je rappelle que sur tous les pages, je ne m'occupais jamais du visuel, je n'apportais que des informations mais sans mise en forme.

### **Interne :**

En interne j'ai fait la création de fausses données que l'on appelle « DataFixtures », j'ai fait 4 faux comptes comme sa à tout moment on pouvait utiliser des comptes sans avoir à en recrée.

Et j'ai créé 10 faux évènement, cette fois ci je ne les ai pas créés à la main j'ai utilisé un autre petit bundle qui s'appelle Faker, ce bundle servait à faire de fausses données facilement, et puis j'avais juste à boucler, assigner de fausses dates à l'évènement et des galops au hasard, puis à la fin j'ai attribué une image par évènement pour que le visuel soit testé. Généralement on utilise que les Fixtures dans la phase développement.

Puis pour que l'application mobile puissent connaître les évènements j'ai installé une api qui est un bundle particulier il s'agit d'Api Platform, j'ai fait des utilisations assez basiques de cet Api. J'envoie sur un fichier Json disponible a tous dans une certaine adresse particulière du site, accessible seulement si on connaît le lien, cette adresse fournit un fichier Json accessible en ligne avec les informations pertinente des évènements avec notamment les dates qui sont des Json aussi.

Je n'ai pas besoin de savoir comment traiter ce Json, celui qui a besoin de ses informations la s'en occupe lui-même.

## Mise en ligne du site

Pour la partie mise en ligne j'ai pris une semaine au moins à comprendre comment bien réussir à mettre le site en ligne.

Je ne savais pas comment m'y prendre du tout.

Après avoir vue une vidéo qui montrait comment faire j'étais toujours bloquer car la personne avait fait sa depuis un ordinateur Linux et du coup il y avait des différences.

Donc j'avais les fichiers sur le site et le nom de domaines lié au site mais je ne savais pas pourquoi le site ne fonctionnait pas.

J'ai appris en faisant des recherches que l'adresse devait pointer sur le dossier public du site internet au lieu du dossier source.

Malgré tout ça ne fonctionnait toujours pas.

Quelques temps après sa je découvre qu'il fallait juste supprimer le cache du développement en local que je ne devais pas envoyer sur le site internet.

Après sa le site internet fonctionnait mais je devais encore y rajouter la base de données.

Pour créer la base de données il fallait que je le fasse avec le protocole ssh, j'ai passé des jours à le faire en local alors que ce n'est pas possible mais grâce à mes connaissances discord j'ai appris qu'il fallait utiliser le protocole.

Donc j'utilise le protocole, je me connecte au site je me dirige sur le dossier source du site pour faire la ligne de commande qui fait la base de données avec les entités et les relations. Mais ma ligne de commande s'exécute avec une vieille version de PHP. J'essaie d'utiliser un alias sans succès.

Au bout de quelques jours je trouve le bon alias et la base de données est créée depuis lors mes cheveux ont repoussé et je n'ai plus de problème.

Finalement c'était très simple une fois que l'on a trouvé comment faire.

Après sa je me suis chargé de maintenir le site en l'état et gérer qui peut aller sur le site à l'aide du fichier htaccess, les IP changent à chaque fois que l'on quitte le réseau donc chaque matin je remettais mon IP pour pouvoir accéder au site je testais les performances d'affichages, notamment des images qui ont eu besoin de remaniement.

Après avoir mis le site en ligne, les autres journées de travail je mettais en ligne les derniers changements du site tout en continuant à travailler sur des bugs et fonctionnalités mineurs.

## Interaction

Au cours du stage j'ai dû interagir avec Laurine qui s'occupait d'autre fonctionnalité pour le site que je n'ai pas forcément cité du coup, il a fallu se mettre d'accord sur qui fait quoi.

Julien quant à lui s'occupait de remanier les pages visuellement il fallait aussi parfois se mettre d'accord mais moins souvent.

Hugo quant à lui s'occupait de son application, à partir du moment où le site était en ligne je l'aidais à faire en sorte que le site soit apte à la création d'événement pour ses tests personnels.

## Spécificités Techniques et Technologies qui me concernent

### Editeur de texte

Comme éditeur de texte nous avons spécifié Visual Studio Code, par la suite on y a intégré git ce qui a permis de l'utiliser de façon simplifiée, cet éditeur nous permet de facilement naviguer dans le dossier et sous dossier à partir du dossier source du projet, avec son intégration de terminal cmd on a facilement pu utiliser les lignes de commandes.

Il affiche le code façon à ce que l'on voie bien où sont les erreurs, les variables, les fonctions etc.

Pour finir on peut installer des extensions comme par exemple une extension qui inclut automatiquement les « use » nécessaires c'est-à-dire les classes que l'on a besoin dans notre fichier.

### Serveur virtuel

Pour avoir notre serveur virtuel nous avons utilisé le logiciel WampServer, grâce à cela on peut faire tourner notre site et base de données sur un serveur local apache, même si au début j'utilisais le serveur de Symfony je me suis rabattue sur le serveur de Wamp au milieu du stage.

Ce logiciel inclut PhpMyAdmin qui est notre outil pour avoir accès à notre base de données MySQL et pouvoir vérifier que ce que l'on fait est correct.

### Partage d'information

Quand nous n'étions pas sur le lieu de travail c'est-à-dire durant les journées on ne travaille pas mais que nous vouliez discuter de certains points nous avons utilisé Discord qui a finalement beaucoup servi. Même durant les heures de travail nous avons envoyé des fichiers, épinglé des liens ou texte qui servirait pour plus tard.

### Echange de données

Pour l'échange de données nous avons utilisé GitHub et l'extension Git sur Visual Studio Code

J'ai fait des changements sur GitHub le plus souvent possible pour que l'on puisse plus facilement voir les changements, et quand on veut on peut envoyer cette liste de changement avec la fonction « push » de git et ainsi faire les changements sur le projet principal sur GitHub, avec la fonction « pull » on récupérait les changements que les autres personnes avaient faites. Malgré quelques erreurs l'outil nous a fait gagner beaucoup de temps.

### Mise en ligne des fichiers

Pour mettre en ligne les fichiers sur notre site internet j'ai utilisé le fichier Filezilla qui envoie les fichiers grâce à un compte ftp-ssh sur le site fournisseur. Après le premier envoi je n'envoie que les fichiers qui ont été modifiés.

### **Retard :**

Par rapport au planning, au 7 Juillet le site devait être en ligne, au final il était seulement en ligne et entièrement opérationnel le 29 Juillet, on a décidé qu'il ne serait pas accessible par le grand public étant donné son visuel encore limité et des fonctionnalités non finies.

C'est en fin Juillet que j'attaque vraiment beaucoup le debug alors que normalement c'est un moment pour les ajouts de fonctionnalités étant donné que nous ajoutons déjà des fonctionnalités plus tôt cela pourrais expliquer le retard et puis cette avance au second debug.

### **Constat :**

Je pense que j'ai réalisé les choses tels que le client le voulait, en essayant de traiter des cas particuliers, des erreurs, mettre des assertions aux utilisateurs sur les formulaires et en essayant de mettre des mini fonctionnalités tels que mot de passe oublier qui ne sont pas demandés par le client mais semblaient incontournables malgré tout. Je ne peux qu'espérer qu'il y est le moins d'oubli et de problème possible.

Je me chargerai de la maintenance du site et je suis à l'écoute du client pour des ajustements mineurs jusqu'à que le flambeau soit passé à d'autre futur stagiaire dans l'entreprise, en tout cas je l'espère pour le futur du site qui ne demande que de nouvelles améliorations ajustements et fonctionnalités.

## Conclusion

Ce stage a pour moi été rempli d'émotion positive, c'est pour moi une expérience qui restera pour moi inoubliable, remplie d'anecdotes, je pense que personne ne s'attend à un stage informatique au milieu de la campagne, à l'extérieur au milieu des chevaux.

Et pourtant nous avons réussi là où personne ne s'y attend à développer un projet jusqu'à son terme, avec des nouvelles technologies qu'il a fallu aborder, de nouvelles notions, mais tout cela a pu bien se dérouler grâce au projet universitaire qui visait à nous mettre en situation au préalable ainsi qu'à certains cours qui avaient aussi pour objectif de nous préparer au monde du travail.

Tout cela fait que l'entreprise possède désormais une réalisation qui leur permet d'avoir une meilleure visibilité sur internet et de gérer plus facilement leurs événements.

Malgré tout cela je ne sais pas encore si je vais continuer mes études dans le domaine informatique.

## Bibliographie

Lior Chamla :

[https://www.youtube.com/watch?v=UTusmVpwJXo&list=PLpUhHhXoxrjdQLodxIHfY09\\_9XzqdPBW8](https://www.youtube.com/watch?v=UTusmVpwJXo&list=PLpUhHhXoxrjdQLodxIHfY09_9XzqdPBW8)

<https://www.grafikart.fr/tutoriels/symfony>

<http://phptester.net/>

<https://symfony.com/doc/current/controller.html>

<https://symfony.com/doc/current/forms.html>

<https://twig.symfony.com/doc/2.x/>

<https://symfony.com/doc/current/doctrine.html>

<https://www.doctrine-project.org/>

<https://symfony.com/doc/current/security.html>

[https://symfony.com/doc/current/security/form\\_login\\_setup.html](https://symfony.com/doc/current/security/form_login_setup.html)

<https://swiftmailer.symfony.com/docs/introduction.html>

<https://symfony.com/doc/master/bundles/EasyAdminBundle/integration/vichuploaderbundle.html>

<https://symfony.com/doc/current/bundles/DoctrineFixturesBundle/index.html>

<https://github.com/fzaninotto/Faker>

<https://api-platform.com/docs>



## Résumé

Dans ce rapport j'explique tout d'abord le fonctionnement de notre projet en présentant Symfony et son architecture ainsi que tous les autres briques qui forment un ensemble d'outil de travaux pour faire un site internet.

Ensuite je parle de ce que j'ai fait dans ce groupe de 4 stagiaires dont je faisais partie sans rentrer dans les détails, ainsi que les problèmes existants et la façon de travailler dans une entreprise sans service informatique.

### **Symfony-Equipe**