

# Inscription à un cours



Laetitia Guidetti

Chef de projet : Cindy Hardegger

ETML mai-juin 2021



# Table des matières

<b>1</b>	<b>Analyse préliminaire</b>	<b>5</b>
1.1	Introduction	5
1.2	Fonctionnalités	5
1.3	Méthodologie de gestion de projet	6
<b>2</b>	<b>Analyse et Conception</b>	<b>7</b>
2.1	Outils	7
2.2	Stratégie de test	7
2.3	Planification initiale	8
2.4	GitHub	9
2.5	Symfony	9
2.6	Bootstrap	9
2.7	Twig	10
2.8	Mockup	10
2.9	MCD et MLD	11
2.10	Doctrine ORM	12
2.11	Vue smartphone	13
2.12	Politique de mots de passe	13
<b>3</b>	<b>Réalisation</b>	<b>15</b>
3.1	Fonctionnement de Symfony	15
3.1.1	Démarrage d'un projet	15
3.1.2	Routes	15
3.1.3	Affichage	16
3.2	Entité	16
3.2.1	Environnement	16
3.2.2	Création d'une entité	16
3.2.3	Migration	19
3.2.4	Les requêtes	20
3.3	Le paquet « Security »	21
3.3.1	L'entité « User »	21
3.3.2	Formulaire de connexion	22
3.3.3	Gestion des accès	22
3.3.4	Le fichier « security.yaml »	23
3.4	Les formulaires	26
3.4.1	Création d'un formulaire	26
3.4.2	Configuration du formulaire	27

3.4.3	Création et envoi des données	28
3.4.4	Affichage du formulaire	29
3.4.5	Validation des champs	30
<b>3.5</b>	<b>Le mot de passe</b>	<b>31</b>
3.5.1	Génération d'un nouveau mot de passe	31
3.5.2	Envoi d'un mail	31
<b>3.6</b>	<b>Mise en production</b>	<b>33</b>
3.6.1	Les variables d'environnement	33
3.6.2	Transfert des fichiers	33
3.6.3	.htaccess	34
<b>4</b>	<b>Tests</b>	<b>35</b>
4.1	Éléments modifiés	35
4.2	Page d'accueil	35
4.3	Connexion d'un administrateur	36
4.4	Gestion des cours	37
4.5	Inscription à un cours	39
4.6	Gestion du profil	41
4.7	Gestion des administrateurs	43
4.8	Mot de passe perdu	45
4.9	Tests de production	46
<b>5</b>	<b>Conclusion</b>	<b>48</b>
5.1	Problèmes rencontrés	48
5.2	Résultat des fonctionnalités	48
5.3	Bilan de la planification	49
5.4	Améliorations possibles	50
5.5	Bilan Personnel	50
<b>6</b>	<b>Glossaire</b>	<b>51</b>
<b>7</b>	<b>Sources et références</b>	<b>52</b>
<b>8</b>	<b>Annexe</b>	<b>53</b>
8.1	Cahier des charges	53
8.2	Résumé du rapport du TPI	57
8.3	Planification initiale détaillée	58
8.4	Journal de travail	68

# 1 Analyse préliminaire

## 1.1 Introduction

Ce projet a pour but de fournir une application permettant de gérer les inscriptions des membres aux différents cours proposés par un club pour l'éducation de leur chien. Par conséquent, il doit permettre de gérer les différentes étapes de l'inscription et de la gestion des cours par un administrateur.

Il est réalisé dans le cadre d'un TPI à l'ETML. Il comprend la réalisation d'un site du début à la fin. Il permet de passer par la conception de la base de données, des mockups ainsi que la réalisation du code. Il faudra également procéder à des tests pour vérifier les fonctionnalités et finir par une mise en production.

## 1.2 Fonctionnalités

Pour la réalisation de ce projet, plusieurs fonctionnalités sont demandées :

- Affichage d'une liste de cours selon des critères
- Inscription à un cours
- Authentification des administrateurs
- Gestion des accès
- Gestion des cours (listes de participants et ajout de cours)
- Gérer son profil
- Gérer les administrateurs
- Gestion d'un mot de passe perdu
- Mise en production

Chaque fonctionnalité sera analysée, réalisée puis testée. La réussite ou non de l'implémentation des fonctionnalités sera indiquée lors de la conclusion.

### 1.3 Méthodologie de gestion de projet

Pour ce projet le choix de la méthode de gestion de projet s'est porté sur le modèle en cascade. Il est généralement séparé en 5 phases qui doivent se réaliser les unes après les autres.

1. Analyse
2. Conception
3. Implémentation
4. Test
5. Exploitation

Ce modèle permet d'avoir une structure simple et claire. Comme chaque étape se déroule l'une après l'autre, le moindre retard est directement remarquable. Le défaut principal de cette méthodologie est qu'elle ne supporte pas le changement. Si le client change d'avis, il sera difficile d'apporter ces modifications à un projet utilisant la méthodologie en cascade. Dans le cadre de ce TPI, cela n'est pas un problème, car le CDC et les objectifs ne peuvent pas être modifiés au cours de la réalisation du projet.

Chaque fonctionnalité sera analysée puis conceptualisée. Elle sera ensuite implémentée dans le code, après cela des tests permettront de vérifier que la réalisation est conforme aux attentes. L'exploitation se fera à la toute fin du TPI avec la mise en service du site sur les serveurs de l'ETML.

## 2 Analyse et Conception

### 2.1 Outils

Liste des outils utilisés pour réaliser ce projet.

Outils	Version
PHP	7.4.16
Symfony CLI	4.24.1
Bootstrap	4.6
Composer	2.0.12
XAMPP Control Panel	3.2.4
Apache	2.4.46
MariaDB	10.4.18
phpMyAdmin	5.1.0
FileZilla Client	3.54.1

Il n'est pas obligatoire d'utiliser les mêmes outils pour réaliser ce projet. Par exemple, comme gestionnaire de base de données, il est possible d'utiliser MySQL à la place de MariaDB ou encore d'utiliser Adminer à la place de phpMyAdmin.

### 2.2 Stratégie de test

Des tests seront réalisés pour chaque fonctionnalité. Plusieurs tests permettront de vérifier si la fonctionnalité marche avec des valeurs acceptables. Pour les cas où il y a une donnée modifiable par un utilisateur, des tests seront réalisés afin de s'assurer qu'il n'est pas possible d'insérer des données illogiques.

Chaque test sera présenté sous la forme d'un tableau :

Numéro	Nom du test
But	Description de l'objectif du test.
Action	Détails des actions à effectuer pour réaliser ce test.
Résultat attendu	Résultat attendu avant la réalisation du test.
Résultat	Résultat obtenu après la réalisation du test.
Image indiquant le résultat du test	

Chaque groupe de tests est daté, car des modifications peuvent avoir eu lieu entre un test et la finalisation du projet.

## 2.3 Planification initiale

Les détails de la planification sont disponibles en annexe (cf. 8.3 Planification initiale détaillée). Elle est composée de plusieurs séquences qui correspondent chacune à une demi-journée. Les jours fériés n'ont pas été pris en compte. Répartition des heures sur une semaine :

Jour de la semaine	Matin	Après-midi
Lundi	3h10	3h10
Mardi	3h10	3h10
Mercredi		4h
Jeudi	4h	3h10
Vendredi	3h10	3h10

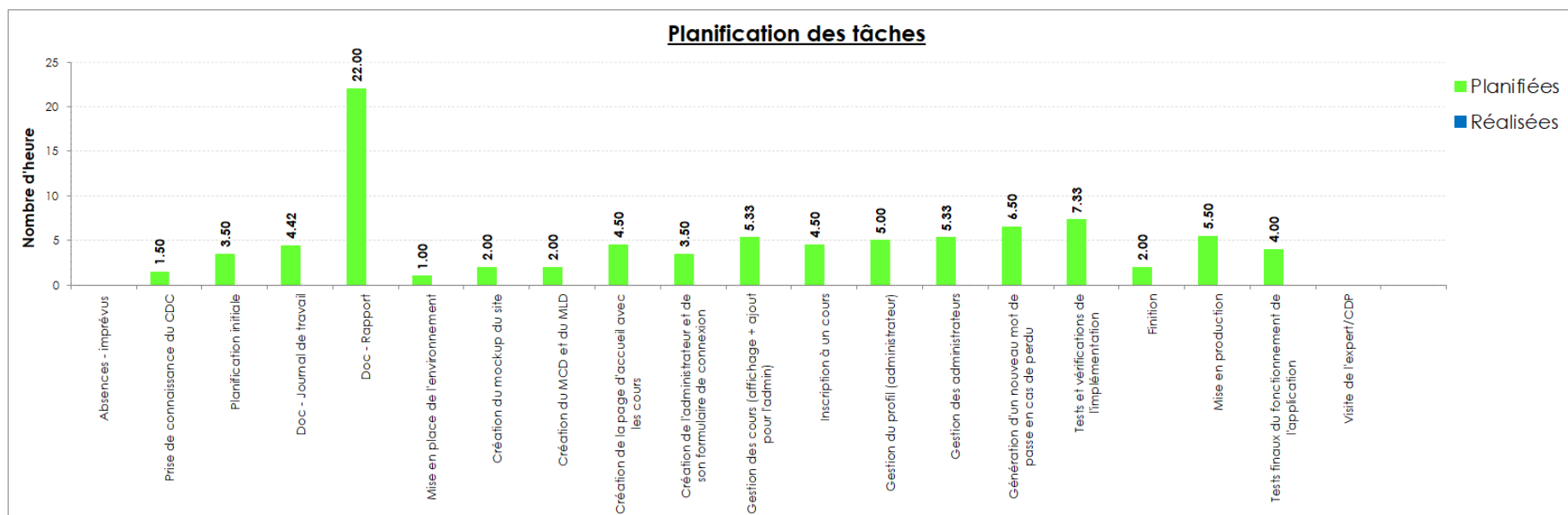


Figure 1 Planification initiale



## 2.4 GitHub

Pour gérer les versions et éviter de perdre une partie du projet en cas de problème un répertoire git est mis en place. Des commit réguliers permettront également de renseigner le journal de travail. La référence du commit sera alors utilisée pour permettre de retrouver la partie de code modifiée à chaque séquence.

Tous les mots de passe ou identifiants de connexion présentent sur le git ne sont pas ceux de l'environnement de production pour des raisons évidentes de sécurité.

Répertoire GitHub : [https://github.com/Chywou/TPI\\_InscriptionCours/tree/dev](https://github.com/Chywou/TPI_InscriptionCours/tree/dev)

## 2.5 Symfony

Dans ce projet, utiliser un framework PHP n'est pas imposé. Cependant, je souhaite quand même utiliser Symfony pour des raisons pratiques.

Lors de la création d'un site Web certaines fonctionnalités reviennent très souvent : la gestion de l'authentification des utilisateurs, un formulaire de contact ou encore la validation des données. Il s'agit de tâches redondantes, ce serait alors une perte de temps de les recréer à chaque nouveau site. Un framework PHP permet alors de fournir différents composants permettant d'implémenter directement ces fonctionnalités sans avoir besoin de les recréer. Ces composants ont chacun déjà été utilisés par des milliers de sites, par conséquent un grand nombre de bugs et failles de sécurité ont déjà été trouvés et corrigés.

Un framework permet une fois un minimum maîtrisé de gagner beaucoup de temps tout en améliorant la sécurité globale du site. Le défaut principal est qu'au début, il est assez dur d'apprendre un framework PHP, car il nécessite de comprendre son fonctionnement global.

Parmi tous les frameworks PHP, le choix s'est porté sur Symfony. Il s'agit de l'un des plus utilisés à travers le monde et il intègre l'architecture MVC nécessaire dans ce projet.

## 2.6 Bootstrap

Le choix du framework CSS s'est porté sur Bootstrap. Il est OpenSource et a été développé pour le mobile-first. Cela implique qu'il est facile à adapter pour les différentes tailles d'écran. J'ai également déjà eu l'occasion de l'utiliser durant des projets de développement Web à l'ETML, ce qui me permet de l'utiliser avec plus d'aisance qu'un autre framework CSS.

Avant de commencer le projet, j'avais également pensé au framework Foundation qui est souvent utilisé avec Symfony. Cependant, il m'a paru plus complexe que Bootstrap et pour des questions de temps je ne voulais pas réapprendre un framework CSS.

Son utilisation ne demande aucun téléchargement, il suffit d'insérer les liens vers le CSS et les fichiers JavaScript correspondant pour le faire fonctionner.

La version de Bootstrap utilisée durant ce projet est la 4.6. La version 5.0 est sortie le 5 mai dernier. Elle n'est cependant pas utilisée, car elle n'a pas encore été implémentée dans les thèmes de Symfony. Plusieurs fonctionnalités de ce dernier donnent la possibilité d'utiliser un thème précis pour modifier l'affichage. Par exemple, lorsqu'un formulaire est créé, il est possible de lui choisir le thème Bootstrap 4. Par conséquent, si l'application utilise la version 5, l'apparence du formulaire sera mal gérée, car certaines classes possèdent des noms différents d'une version à l'autre.

## 2.7 Twig

Twig est un moteur de modèle PHP et il est par défaut utilisé par Symfony. Il est utilisé pour toute la partie graphique d'un site. Il est recommandé d'utiliser la norme de nommage snake case pour les noms des fichiers et des dossiers. Cette convention indique que chaque espace est remplacé par un tiret bas.

Il possède 3 syntaxes de base :

`{{ ... }}` permet d'appeler une expression (variable, fonction, etc.)

`{% ... %}` exécute une commande

`{# ... #}` n'est pas exécuté, il s'agit des commentaires

Il est assez facile d'utiliser Twig, toute la partie HTML/CSS est identique à un projet classique. La seule modification est pour la partie PHP où des syntaxes spécifiques à Twig sont à utiliser. Cependant, les mots-clés sont similaires, il est donc facile de s'adapter.

## 2.8 Mockup

Une mockup est une maquette permettant de représenter le site web de manière simpliste.

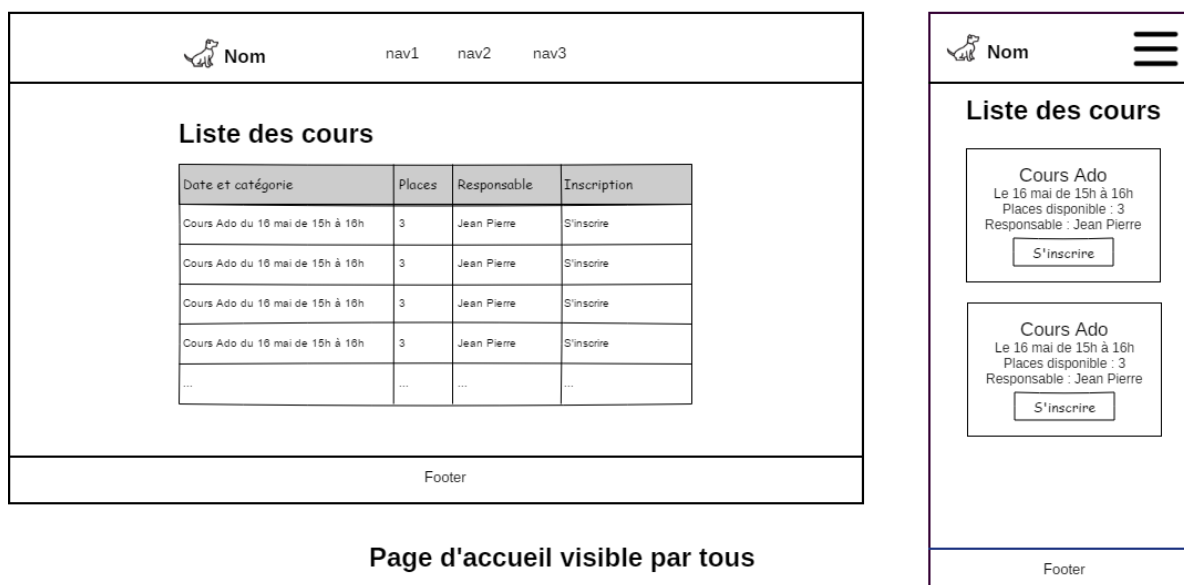



Figure 2 Mockup de la page d'accueil

La page d'accueil du site sert à afficher les différents cours et permet aux utilisateurs de s'y inscrire. Par conséquent, l'aspect graphique doit rester assez simple afin de ne pas rendre la tâche difficile. L'affichage sur téléphone est modifié, car la largeur de ces appareils ne permet pas d'afficher les grands tableaux.



**Formulaire d'ajout de cours pour l'administrateur**

*Figure 3 Mockup du formulaire d'ajout de cours*

Le formulaire d'ajout d'un cours ne contient que les informations utiles pour l'administrateur tout en restant intuitif. Ce mockup sert de base à toutes les pages du site utilisant des formulaires comme l'inscription d'un utilisateur à un cours.

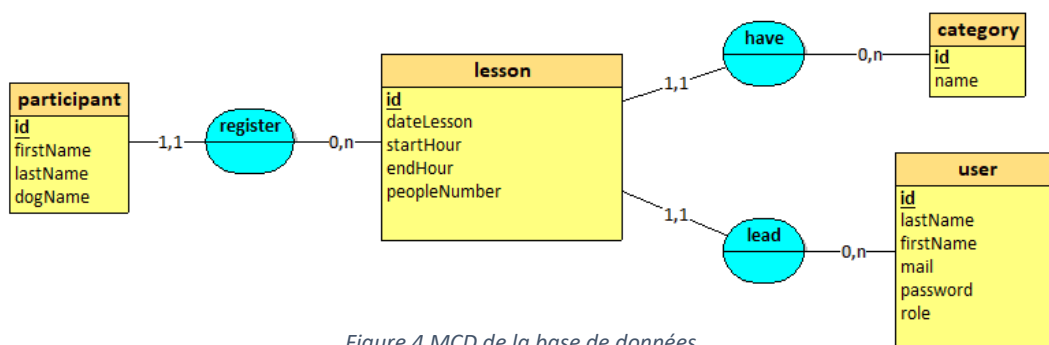
## 2.9 MCD et MLD

La modélisation d'une base de données se fait en 3 étapes.

1. Le modèle conceptuel des données (MCD)
2. Le modèle logique de données (MLD)
3. Le modèle physique de données (MPD)

Pour réaliser une base de données plusieurs règles sont à respecter.

1. Non-redondance : chaque information n'apparaît qu'une fois
2. Atomicité : une seule information par champ
3. Unicité : chaque champ est identifiable de manière unique



*Figure 4 MCD de la base de données*

Dans le CDC, une liste des caractéristiques d'un cours est donnée. Cependant, créer un type d'entité directement d'après cette liste ne respecterait pas les règles d'élaboration d'une BD. L'une des caractéristiques est qu'un cours possède « une catégorie de cours (ex : Ado) ». Ajouter, sur le type d'entité « lesson », l'attribut « category » serait contraire au principe de la non-redondance. En effet, si plusieurs cours avaient comme catégorie « Ado », il y aurait alors une redondance. Un nouveau type d'entité est donc utilisé afin de respecter cette règle. On remarque également que pour le nom des participants, 2 attributs sont prévus l'un pour le nom de famille et l'autre pour le prénom, afin de respecter la règle d'atomicité.

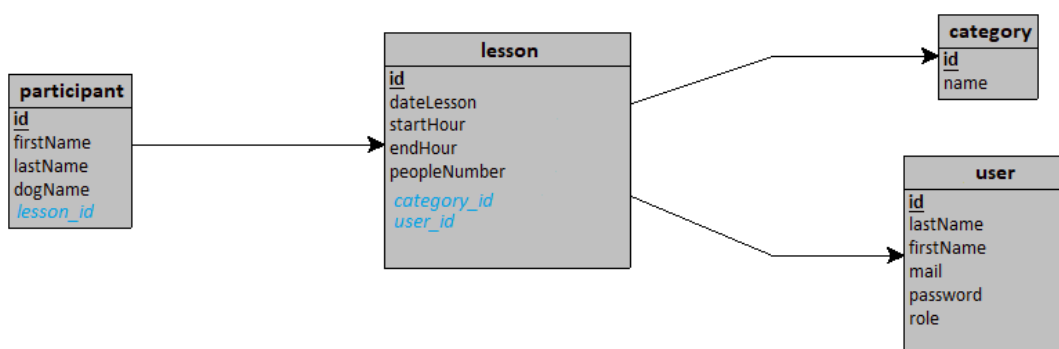


Figure 5 MLD de la base de données

Le MLD est basé sur le MCD et il représente les futures tables et colonnes de la base de données. Les normes de nommages ne sont pas celles de l'ETML, il s'agit des normes de Symfony.

Pour les champs qui concernent les noms et prénom, une taille de 50 caractères a été choisie. Il s'agit d'une taille générique qu'on retrouve régulièrement pour ces champs. Pour le mail, la taille de 254 caractères est utilisée, il s'agit de la taille recommandée par l'[erratum 1690](https://www.rfc-editor.org/errata_search.php?rfc=3696&eid=1690)<sup>1</sup> qui corrige la RFC 3696.

## 2.10 Doctrine ORM

Doctrine est l'ORM (object-relational mapping) utilisé par Symfony. Un ORM est une couche d'abstraction qui synchronise les objets PHP avec une base de données relationnelles. Son but est de créer un ensemble de classes permettant de manipuler les tables et les données de la BD. Il permet de transformer les objets PHP en écriture SQL, ce qui permet notamment de générer des requêtes SQL.

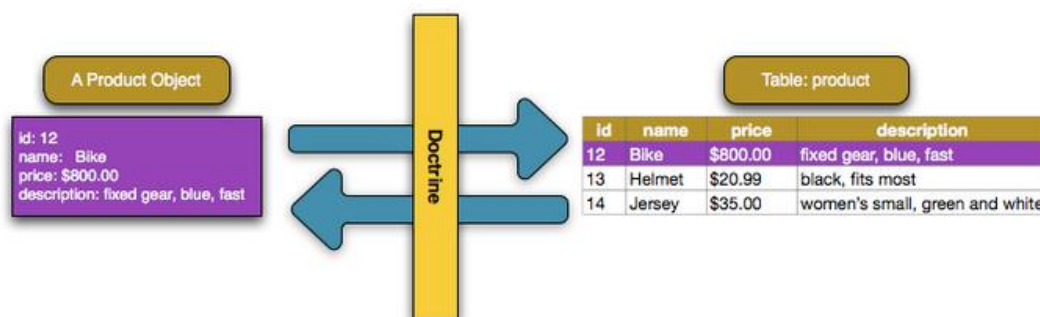


Figure 6 Fonctionnement de Doctrine, source : <https://symfony.com/doc/current/doctrine.html>

<sup>1</sup> [https://www.rfc-editor.org/errata\\_search.php?rfc=3696&eid=1690](https://www.rfc-editor.org/errata_search.php?rfc=3696&eid=1690)

C'est donc via Doctrine que la base de données va être créée ainsi que ces différentes tables en se basant sur le MLD. Il va permettre de créer des classes appelées entités. À partir de ces dernières, une base de données correspondante pourra être générée automatiquement.

## 2.11 Vue smartphone

Bootstrap est un framework mobile-first. 5 points de ruptures sont définis pour les différentes tailles d'écrans existantes.

Extra small	Small	Medium	Large	Extra large
<576px	≥576px	≥768px	≥992px	≥1200px
	xs	md	lg	xl

Si aucun point de rupture n'est utilisé, une classe s'applique par défaut pour toutes les tailles d'écran. Par exemple la classe « text-left » placée sur une balise place le texte à gauche pour toutes les tailles. Si on y ajoute la classe « text-md-center », pour tous les écrans de taille supérieure ou égale à 768px, le texte sera centré. Ceux de taille inférieure au point de rupture ne seront pas affectés par cette classe.

Dans ce projet, c'est la taille « medium » qui est utilisée. L'affiche sous forme de tableau comme visible sur les mockups n'est plus vraiment adapté en dessous de cette taille.

Il est également utile de préciser que la page d'accueil n'utilise pas la balise « table ». Elle serait pratique à utiliser, car de nombreux styles existent pour les paramétrer avec Bootstrap. Cependant, il est presque impossible de rendre un tableau responsive comme visible sur les mockup. Par conséquent, ce sont des listes qui seront utilisées, car elles permettent de facilement implémenter une responsivité.

## 2.12 Politique de mots de passe

Un mot de passe permet de protéger l'accès à des informations et à leur modification. Dans ce projet, si une personne malintentionnée récupérerait les identifiants de connexion d'un administrateur, il pourrait nuire au club d'éducation en récupérant les informations des participants ou en créant des faux cours. Posséder un mot de passe difficile à trouver permet de limiter au mieux cette possibilité. Pour cela une politique de mot de passe est mise en place afin que seuls des mots de passe difficiles soient utilisés. Cette dernière est notamment recommandée par le [NCSC<sup>2</sup>](https://www.ncsc.admin.ch/ncsc/fr/home/infos-fuer/infos-private/aktuelle-themen/schuetzen-sie-ihre-konten.html) (centre national pour la cyber sécurité).

Politique de mots de passe :

- Longueur minimale de 12 caractères
- Minimum une lettre minuscule
- Minimum une lettre majuscule
- Minimum un chiffre
- Minimum un caractère spécial

<sup>2</sup> <https://www.ncsc.admin.ch/ncsc/fr/home/infos-fuer/infos-private/aktuelle-themen/schuetzen-sie-ihre-konten.html>

Elle est plus stricte que celle de nombreux sites web. Généralement, le nombre minimal de caractères demandé est de 8 et aucun caractère spécial n'est exigé. Cependant, dans le cadre de ce projet, les personnes touchées par cette politique sont les administrateurs. Ils ont un accès direct aux données personnelles des utilisateurs. Pour cette raison, une politique très stricte est utilisée. Si à l'avenir, des utilisateurs non-administrateurs pouvaient s'inscrire sur le site, la politique de mots de passe de ces derniers pourrait être différentes de celle des administrateurs.

## 3 Réalisation

### 3.1 Fonctionnement de Symfony

#### 3.1.1 Démarrage d'un projet

Avant de créer un projet, il faut que les outils présents au point 2.1 soient installés dans l'environnement. Ils ne sont pas nécessaires à la création, mais ils pourraient bloquer lors du reste de la réalisation. La création d'un projet Symfony se fait de manière simple via une commande de Composer.

```
C:\Users\eleve\Projet> composer create-project symfony/website-skeleton TPI_inscriptionCours
Creating a "symfony/website-skeleton" project at "./TPI_inscriptionCours"
```

Figure 7 Création d'un projet Symfony

Elle peut durer un certain temps, car de nombreux fichiers sont téléchargés.

```
Some files may have been created or updated to configure your new packages.
Please review, edit and commit them: these files are yours.
```

Figure 8 Fin de l'installation d'un projet Symfony

Une fois l'installation terminée un message invite l'utilisateur à commencer l'édition des fichiers.

#### 3.1.2 Routes

Quand un utilisateur clique sur un lien ou écrit une URL, la page correspondante doit s'afficher. Le routage permet de définir quelle action doit être effectuée en fonction de la requête reçue par l'application Web. L'action renverra ensuite les différentes informations à afficher.

Avec Symfony, il est possible de définir les routes de plusieurs façons, : via des attributs ou des annotations, dans des fichiers YAML, XML ou PHP. Selon les [bonnes pratiques](https://symfony.com/doc/current/best_practices.html#use-attributes-or-annotations-to-configure-routing-caching-and-security)<sup>3</sup> de Symfony, il est recommandé de les définir via des attributs ou des annotations.

```
// src/Controller/HomeController.php

/**
 * @Route("/", name="home")
 */
public function index(Request $request, PaginatorInterface $paginator): Response
```

Figure 9 Route définie par une annotation

Dans ce cas, quand l'utilisateur va sur l'adresse « / », la méthode index est appelée. Elle se charge de répondre à la demande en affichant la page correspondante.

<sup>3</sup> [https://symfony.com/doc/current/best\\_practices.html#use-attributes-or-annotations-to-configure-routing-caching-and-security](https://symfony.com/doc/current/best_practices.html#use-attributes-or-annotations-to-configure-routing-caching-and-security)

### 3.1.3 Affichage

Avec ce framework, tout l’affichage, donc le code HTML et CSS, est situé dans des fichiers Twig. Le contrôleur est chargé d’afficher le bon modèle Twig et de lui fournir toutes les données dont il a besoin.

```
// src/Controller/HomeController.php  
  
return $this->render('pages/home.html.twig', ['allLessons' => $allLessons]);
```

Figure 10 méthode permettant l’affichage d’une page

La méthode « render » permet de retourner le contenu créé par le modèle spécifié. La variable « allLessons » permet d’envoyer des données précises que le modèle est chargé d’utiliser et d’afficher.

## 3.2 Entité

### 3.2.1 Environnement

Pour créer la structure de la base de données, il faut utiliser les entités provenant de Doctrine. L’environnement doit alors être correctement configuré. Cela demande la création d’une base de données vide et des informations pour s’y connecter. Tout cela est défini dans le fichier .env situé à la racine du projet sous la variable d’environnement « DATABASE\_URL ». Cette ligne contient en premier le nom et le mot de passe de l’utilisateur qui accède à la BD. Puis, l’adresse de l’hôte et le nom de la base de données sont renseignés.

```
DATABASE_URL="mysql://user:root@127.0.0.1:3306/db_registerLesson?serverVersion=mariadb-10.4.18"
```

Figure 11 Information de la base de données dans le fichier .env

Une base de données vides correspondant à ces informations peut être alors générée via une commande. Une erreur sera soulevée si une BD portant son nom existe déjà.

```
C:\Users\eleve\Projet\TPI_InscriptionCours> php bin/console doctrine:database:create  
Created database `db_registerLesson` for connection named default
```

Figure 12 Commande créant la base de données

Il est maintenant possible de créer les entités qui permettront de générer les tables dans la base de données.

### 3.2.2 Création d’une entité

La création d’une entité se fait via une ligne de commande.

```
C:\Users\eleve\Projet\TPI_InscriptionCours>php bin/console make:entity  
  
Class name of the entity to create or update (e.g. AgreeablePopsicle):  
> Lesson  
  
created: src/Entity/Lesson.php  
created: src/Repository/LessonRepository.php  
  
Entity generated! Now let's add some fields!  
You can always add more fields later manually or by re-running this command.
```

Figure 13 Création d’une entité



Après avoir donné le nom de l'entité, 2 fichiers sont automatiquement créés. Le premier « Lesson.php » contient la classe « Lesson ». Le deuxième est le fichier « LessonRepository.php », il permet d'exécuter des requêtes de base et de créer des requêtes précises.

L'invite de commande proposera ensuite d'ajouter des propriétés à l'entité créée.

```
New property name (press <return> to stop adding fields):
> dateLesson

Field type (enter ? to see all types) [string]:
> ?

Main types
* string
* text
* boolean
* integer (or smallint, bigint)
* float

Relationships / Associations
* relation (a wizard will help you build the relation)
* ManyToOne
* OneToMany
* ManyToMany
* OneToOne

Array/Object Types
* array (or simple_array)
* json
* object
* binary
* blob

Date/Time Types
* datetime (or datetime_immutable)
* datetimetz (or datetimetz_immutable)
* date (or date_immutable)
* time (or time_immutable)
* dateinterval

Other Types
* ascii_string
* decimal
* guid
* json_array

Field type (enter ? to see all types) [string]:
> date

Can this field be null in the database (nullable) (yes/no) [no]:
> no

updated: src/Entity/Lesson.php
```

Figure 14 Ajouter une propriété à une entité

Plusieurs questions seront alors posées pour configurer la propriété correctement. Le fichier « Lesson.php » est alors mis à jour en conséquence. Il est possible dans ce dernier de configurer de manière plus précisée les propriétés comme le fait d'ajouter une valeur par défaut. Ce fichier génère également des getter et setter pour chaque propriété de manière à faciliter leurs utilisations.

Pour chaque entité, une propriété « id » est automatiquement créée.

Après avoir créé toutes les entités, il manque les relations entre elles.

```
C:\Users\eleve\Projet\TPI_InscriptionCours>php bin/console make:entity

Class name of the entity to create or update (e.g. GrumpyPopsicle):
> Lesson

Your entity already exists! So let's add some new fields!

New property name (press <return> to stop adding fields):
> category

Field type (enter ? to see all types) [string]:
> relation

What class should this entity be related to?:
> Category

What type of relationship is this?
-----
Type      Description
-----
ManyToOne Each Lesson relates to (has) one Category.
          Each Category can relate to (can have) many Lesson objects

OneToMany Each Lesson can relate to (can have) many Category objects.
          Each Category relates to (has) one Lesson

ManyToMany Each Lesson can relate to (can have) many Category objects.
          Each Category can also relate to (can also have) many Lesson objects

OneToOne   Each Lesson relates to (has) exactly one Category.
          Each Category also relates to (has) exactly one Lesson.
-----

Relation type? [ManyToOne, OneToMany, ManyToMany, OneToOne]:
> ManyToOne

Is the Lesson.category property allowed to be null (nullable)? (yes/no) [yes]:
> no

Do you want to add a new property to Category so that you can access/update Lesson objects from it - e.g. $category->
> yes

A new property will also be added to the Category class so that you can access the related Lesson objects from it.

New field name inside Category [lessons]:
>

Do you want to activate orphanRemoval on your relationship?
A Lesson is "orphaned" when it is removed from its related Category.
e.g. $category->removeLesson($lesson)

NOTE: If a Lesson may *change* from one Category to another, answer "no".

Do you want to automatically delete orphaned App\Entity\Lesson objects (orphanRemoval)? (yes/no) [no]:
> no

updated: src/Entity/Lesson.php
updated: src/Entity/Category.php
```

Figure 15 Création d'une relation

Pour modifier une entité déjà créée, il suffit lors d'un « make:entity » d'indiquer son nom. La console reconnaît automatiquement que l'entité existe déjà et permet de lui ajouter de nouvelles propriétés. Comme son nom l'indique le type « relation » permet de définir les relations. Il faut alors préciser avec quelle entité la relation se fait et son type. Cela mettra à jour les 2 entités avec ces nouvelles informations.

Toutes ces opérations sont également réalisables en modifiant à la main les divers fichiers d'entités.

### 3.2.3 Migration

Toutes les entités et leurs relations sont créées, cependant la base de données ne contient toujours pas de tables. Il est alors nécessaire de migrer toutes les informations présentes dans les classes entités dans cette dernière. Une commande permet de générer automatiquement un fichier contenant des requêtes SQL correspondant aux entités.

```
C:\Users\eleve\Projet\TPI_InscriptionCours>php bin/console make:migration
```

```
Success!

Next: Review the new migration "migrations/Version20210517084130.php"
Then: Run the migration with php bin/console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html

C:\Users\eleve\Projet\TPI_InscriptionCours>php bin/console doctrine:migrations:migrate

WARNING! You are about to execute a migration in database "db_registerLesson" that could result in schema
changes and data loss. Are you sure you wish to continue? (yes/no) [yes]:
> yes

[notice] Migrating up to DoctrineMigrations\Version20210517084130
[notice] finished in 817.8ms, used 20M memory, 1 migrations executed, 7 sql queries
```

Figure 16 Migration

Une deuxième commande permet d'exécuter tous les fichiers de migrations qui n'ont pas déjà été opérés. La base de données est alors modifiée en conséquence. Si plus tard des modifications sur une entité surviennent, il suffit de recréer un fichier de migration et de l'exécuter.

Il existe d'autres commandes liées aux migrations. Il est possible de supprimer la dernière migration, d'obtenir la liste de toutes les migrations ou encore la version actuelle. Ces cas spécifiques ne sont pas abordés ici, mais ils sont présents dans la [documentation](https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html)<sup>4</sup> de Symfony.

Le MPD est alors créé automatiquement via ces commandes.

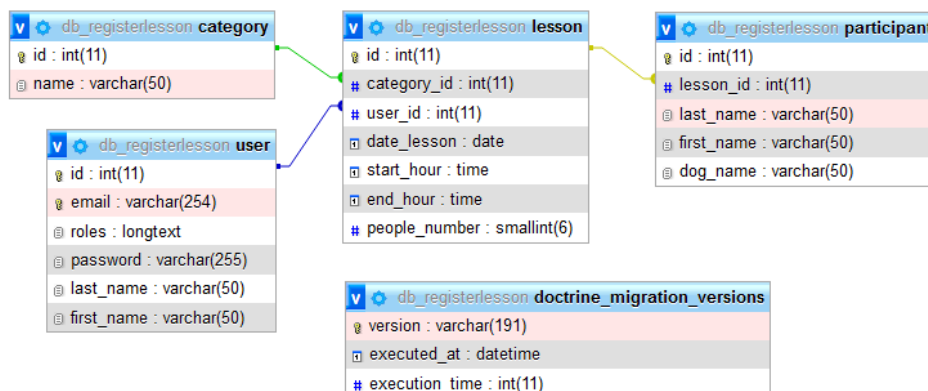


Figure 17 MPD de la base de données après toutes les migrations

La table « doctrine\_migration\_version » est générée automatiquement par Doctrine. Elle sert à gérer automatiquement les versions des différentes migrations.

<sup>4</sup> <https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html>

### 3.2.4 Les requêtes

Certaines informations présentes dans la base de données doivent être affichées aux utilisateurs. Dans le cadre de ce projet, il s'agirait d'afficher les différents cours. C'est à ce moment que le fichier « LessonRepository.php » généré plutôt devient utile. Doctrine permet via ce dernier de générer des requêtes SQL à partir d'objets PHP. Certaines requêtes basiques sont même déjà utilisables. Dans un contrôleur via une instance de la classe « LessonRepository », il est possible d'appeler la méthode « findAll() ». Cette dernière va alors retourner toutes les entrées présentes dans la table « Lesson ». D'autres méthodes pré-écrites du même genre existent comme « find(\$id) » qui retournent l'entrée qui possède l'id spécifié.

Cependant, tout cela n'est pas suffisant pour ce projet. Les utilisateurs ont besoin de voir tous les cours ne s'étant pas encore déroulés et pour des raisons de lisibilité dans un ordre précis. Doctrine permet via le « QueryBuilder » d'écrire des requêtes en utilisant du PHP.

```
// src/Repository/LessonRepository.php
public function findOrderByDate()
{
    return $this->createQueryBuilder('l')
        ->where('l.dateLesson >= CURRENT_DATE()')
        ->addorderBy('l.dateLesson', 'ASC')
        ->addorderBy('l.startHour', 'ASC')
        ->getQuery()
        ->getResult();
}
```

Figure 18 méthode retournant les cours qui ne se sont pas encore déroulés

La méthode « createQueryBuilder » définit automatiquement le SELECT et le FROM, l'argument fourni est généralement la première lettre de l'entité du répertoire. Dans ce cas, l'entité est « Lesson », l'argument utilisé est donc un « l ».

La classe « [QueryBuilder](https://www.doctrine-project.org/projects/doctrine-orm/en/2.8/reference/query-builder.html)<sup>5</sup> » possède de nombreuses méthodes permettant de construire des requêtes complexes. Les noms de ces dernières sont souvent très similaires au SQL ce qui facilite leurs utilisations. La requête est alors construite étape par étape, il est donc facile d'y ajouter diverses conditions.

Doctrine se charge ensuite de générer le SQL correspondant et de l'exécuter.

<sup>5</sup> <https://www.doctrine-project.org/projects/doctrine-orm/en/2.8/reference/query-builder.html>

### 3.3 Le paquet « Security »

Security est un paquet utilisé pour l'authentification et la gestion des accès de manière sécurisée.

#### 3.3.1 L'entité « User »

Les informations des utilisateurs seront stockées dans l'entité « User ». Elle doit obligatoirement implémenter « `Symfony\Component\Security\Core\User\UserInterface` » pour être utilisée. Afin de générer facilement cette entité spéciale, une commande est disponible.

```
C:\Users\eleve\Projet\TPI_InscriptionCours>php bin/console make:user

The name of the security user class (e.g. User) [User]:
> User

Do you want to store user data in the database (via Doctrine)? (yes/no) [yes]:
> yes

Enter a property name that will be the unique "display" name for the user (e.g. email, username, uuid) [email]:
> email

Will this app need to hash/check user passwords? Choose No if passwords are not needed or will be checked/hashed by some other system
gle sign-on server).
Does this app need to hash/check user passwords? (yes/no) [yes]:
> yes

created: src/Entity/User.php
created: src/Repository/UserRepository.php
updated: src/Entity/User.php
updated: config/packages/security.yaml

Success!

Next Steps:
- Review your new App\Entity\User class.
- Use make:entity to add more fields to your User entity and then run make:migration.
- Create a way to authenticate! See https://symfony.com/doc/current/security.html
```

Figure 19 Création de l'entité « User »

Comme durant la création des entités classiques, 2 fichiers sont générés. Ils sont similaires aux précédents à quelques exceptions près. Le classe « User » implémente « `UserInterface` », elle possède la propriété « `roles` » qui permettra la gestion des accès ainsi que la propriété « `password` » qui permettra de stocker le mot de passe haché. Le fichier de configuration du paquet « `security.yaml` » est également mis à jour, il est détaillé au point 3.3.4 Le fichier « `security.yaml` ».

### 3.3.2 Formulaire de connexion

Des administrateurs peuvent être créés dans la table « user ». Il faut maintenant qu'ils puissent se connecter pour administrer le site. Le paquet « Security » permet de générer un formulaire sécurisé via une commande.

```
C:\Users\eleve\Projet\TPI_InscriptionCours>php bin/console make:auth

What style of authentication do you want? [Empty authenticator]:
[0] Empty authenticator
[1] Login form authenticator
> 1

The class name of the authenticator to create (e.g. AppCustomAuthenticator):
> LoginFormAuthenticator

Choose a name for the controller class (e.g. SecurityController) [SecurityController]:
>

Do you want to generate a '/logout' URL? (yes/no) [yes]:
> yes

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

Success!

Next:
- Customize your new authenticator.
- Finish the redirect "TODO" in the App\Security\LoginFormAuthenticator::onAuthenticationSuccess() method.
- Review & adapt the login template: templates/security/login.html.twig.
```

Il génère le fichier « templates/security/login.html.twig », il contient le formulaire de connexion. Il est à modifier, car tous les termes y sont en anglais et son aspect graphique n'est pas le plus agréable. Le fichier « src/Controller/SecurityController.php » est également créé. Il contient un contrôleur chargé de gérer les routes de connexion et de déconnexion. Le dernier fichier généré « src/security/LoginFormAuthenticator.php » est le plus complexe. Il se charge de toute la procédure d'authentification.

### 3.3.3 Gestion des accès

Les utilisateurs peuvent maintenant se connecter et se déconnecter. Cependant, cela ne donne aucun accès aux administrateurs et n'en restreint aucun, il faut donc les configurer. Par défaut, la class User possède une propriété utilisée pour la gestion des accès. Il s'agit de « roles » qui est de type JSON. C'est la méthode getRoles() qui est ensuite utilisée pour déterminer quels rôles sont possédés par un utilisateur. Dans tous les cas, pour les utilisateurs connectés, la méthode renverra au minimum le rôle « ROLE\_USER ».

```
// src/Entity/User.php

/**
 * @see UserInterface
 */
public function getRoles(): array
{
    $roles = $this->roles;
    // garanti à tous les utilisateurs d'avoir au minimum ROLE_USER
    $roles[] = 'ROLE_USER';

    return array_unique($roles);
}
```

Figure 20 La méthode getRoles()

Il est possible d'attribuer d'autres rôles aux utilisateurs, mais ils doivent respecter une règle : commencer par « ROLE\_ » comme par exemple « ROLE\_ADMIN ». Il est possible de configurer les différentes restrictions d'accès dans le fichier `security.yaml` ou dans le contrôleur (ou dans d'autres parties du code comme les fichiers Twig). Cependant, il n'est pas recommandé par les [bonnes pratiques](https://symfony.com/doc/current/best_practices.html#use-attributes-or-annotations-to-configure-routing-caching-and-security)<sup>6</sup> de Symfony de le faire dans le fichier `security.yaml`.

```
// src/Controller/AdminController.php

/**
 * Demande le rôle admin pour l'accès de ce contrôleur
 *
 * @IsGranted("ROLE_ADMIN")
 */
class AdminController extends AbstractController
```

Figure 21 Annotation empêchant l'accès à tous les utilisateurs ne possédant pas le rôle "ROLE\_ADMIN"

Avant une méthode d'un contrôleur, l'annotation « `IsGranted` » permet de restreindre la méthode aux utilisateurs possédant le rôle indiqué. Si aucun utilisateur n'est connecté, il sera redirigé vers la page de connexion. Si l'utilisateur est connecté, mais ne possède pas le rôle, il aura le message d'erreur 403 accès refusé. L'annotation « `IsGranted` » peut également être utilisée avant la déclaration du contrôleur pour restreindre l'entièreté des méthodes de ce dernier. Cette annotation demande de faire référence à l'espace de nom `Sensio\Bundle\FrameworkExtraBundle\Configuration\IsGranted` pour fonctionner.

Il est également possible de faire une restriction sur un modèle Twig. Par exemple, si l'on souhaite que les boutons « Connexion » et « Déconnexion » s'affichent en fonction de si l'utilisateur est connecté ou non. Il est possible d'utiliser la fonction « `is_granted(role)` » pour appliquer des restrictions. À noter qu'il est possible d'utiliser certains attributs à la place des rôles. Par exemple l'attribut « `IS_ANONYMOUS` » est valide uniquement avec les utilisateurs non connectés.

### 3.3.4 Le fichier « `security.yaml` »

Plusieurs fois lors des étapes précédentes, l'invite de commande indiquait que le fichier « `security.yaml` » avait été mis à jour. Il contient la configuration du paquet « Security » et permet son fonctionnement. Il est séparé en plusieurs options.

```
# config/packages/security.yaml

encoders:
    App\Entity\User:
        algorithm: auto
```

Figure 22 L'option « `encoders` »

<sup>6</sup> [https://symfony.com/doc/current/best\\_practices.html#use-attributes-or-annotations-to-configure-routing-caching-and-security](https://symfony.com/doc/current/best_practices.html#use-attributes-or-annotations-to-configure-routing-caching-and-security)

La première option est « encoders », elle permet de déterminer l'algorithme utilisé pour le hachage. Le nom fait plus penser à de l'encodage que du hachage, mais il s'agit bien de hachage ce nom est présent pour des raisons historiques. La classe sur laquelle s'applique cet algorithme est à préciser, car il est possible d'avoir plusieurs classes avec des algorithmes différents. Choisir « auto » permet de laisser Symfony choisir le meilleur algorithme de hachage disponible, selon lui. Actuellement, il s'agit de l'Argon2 qui hache sur 96 caractères. S'il n'est pas disponible, c'est l'algorithme bcrypt qui est utilisé. Symfony se charge de réaliser un salage différent à chaque nouveau mot de passe haché. L'algorithme de hachage utilisé est voué à être changé dans le futur lorsqu'un meilleur algorithme sera trouvé, par conséquent il est recommandé de prévoir assez d'espace pour sauvegarder les mots de passe hachés.

```
# config/packages/security.yaml

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

Figure 23 L'option « providers »

L'option suivante est « providers ». Il permet d'indiquer quelle entité représente les utilisateurs et quelle propriété est à utiliser pour l'authentification. Ces paramètres ont été générés automatiquement à la création de la classe « User ». La classe « User Providers » se charge également de l'actualisation de l'utilisateur courant dans les variables de session et de rendre possible l'ajout de plusieurs fonctionnalités comme le « se souvenir de moi ».

```
# config/packages/security.yaml

firewalls:
    dev:
        pattern: ^/(_(profiler|wdt)|css|images|js)/
        security: false
    main:
        anonymous: true
        lazy: true
        provider: app_user_provider
        guard:
            authenticators:
                - App\Security\LoginFormAuthenticator
        logout:
            path: logout
            target: home
```

Figure 24 L'option « firewalls »

Une des autres options est « firewalls », elle est considérée comme la plus importante. Elle gère comment les utilisateurs vont être authentifiés. Il peut y avoir plusieurs firewalls, chacun opère comme un système de sécurité différent. Dans ce cas, il y en a deux « dev » et « main ». Le premier comme



son nom l'indique est présent pour le développement. Il gère les accès au « profiler » un outil de développement permettant d'obtenir des informations détaillées sur chaque requête. La deuxième « main » permet de gérer l'authentification des utilisateurs du site. Chaque firewall peut ensuite être configuré via plusieurs options. L'option « provider » permet de déterminer quelle entité définie plus tôt comprend les utilisateurs. La classe chargée de l'authentification est indiquée grâce à l'option « guard ». Il est également possible de définir la procédure de déconnexion. Sous « logout », « path » indique quelle route gère la déconnexion et « target » indique la route sur laquelle l'utilisateur doit être redirigé après cette opération.

Les informations présentes dans ce fichier sont générées automatiquement lors de la création d'un système d'authentification via « Security ». Si ce système est modifié, il faut impérativement modifier ce fichier. Par exemple, si à la place du mail, il est souhaité d'utiliser un pseudo lors de la connexion. La valeur de l'option « property » du « provider » doit être modifiée pour indiquer la nouvelle propriété.

## 3.4 Les formulaires

Sur un site web, il est commun pour les utilisateurs de remplir des formulaires. Cela permet de renseigner une base de données dans les cas d'inscriptions ou cela peut permettre d'envoyer un mail dans le cas du formulaire de contact.

Cependant, la création de ces derniers est relativement répétitive et longue à réaliser. Il faut renseigner tous les champs, faire une validation de données, s'assurer de la conformité de tous les champs et de leurs bonnes intégrations dans la BD. Mais, Symfony possède une fonctionnalité qui simplifie grandement toutes ses opérations, il s'agit de l'outil « form ».

Dans ce projet, les formulaires sont utilisés à de nombreuses reprises. Notamment, car les administrateurs ont accès à plusieurs ajouts ou modifications d'informations.

### 3.4.1 Création d'un formulaire

Une commande permet de générer automatiquement un formulaire.

```
C:\Users\eleve\Projet\TPI_InscriptionCours>php bin/console make:form

The name of the form class (e.g. AgreeableKangarooType):
> AddLessonType

The name of Entity or fully qualified model class name that the new form will be bound to (empty for none):
> Lesson

created: src/Form/AddLessonType.php

Success!

Next: Add fields to your form and start using it.
Find the documentation at https://symfony.com/doc/current/forms.html
```

*Figure 25 Création du formulaire « AddLessonType »*

L'invite de commande demande en premier le nom du formulaire qui doit finir par « type ». Puis, il demande s'il se réfère à une entité existante. Cela permet de générer automatiquement des champs correspondant aux différentes propriétés de l'entité. Un fichier contenant la classe du formulaire est ensuite généré automatiquement.

### 3.4.2 Configuration du formulaire

Une fois la classe du formulaire générée, certaines modifications peuvent être à faire pour le rendre plus pertinent.

```
// src/Form/AddCategoryType.php

class AddCategoryType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('name', Null, ['label' => 'Nom'])
    }

    public function configureOptions(OptionsResolver $resolver)
    {
        $resolver->setDefaults([
            'data_class' => Category::class,
        ]);
    }
}
```

Figure 26 Classe « AddCategoryType » permettant l'ajout d'une catégorie

La méthode « buildForm » permet d'ajouter les différents champs du formulaire via la méthode « add » provenant de l'interface « FormBuilderInterface ». Pour chaque champ, il est possible de configurer plusieurs paramètres. Par exemple, à l'affichage d'un champ, le nom de ce dernier est utilisé comme label. Cependant, lorsque le formulaire provient d'une entité, le nom de chaque champ se refait à la propriété correspondante de l'entité. Cependant, pour l'utilisateur il n'est pas pratique d'avoir le nom d'un paramètre comme label. Il est possible de régler ce type de problème via l'option « label » qui permet de redéfinir le label affiché. Il existe diverses autres options permettant de modifier l'affichage ou le fonctionnement d'un champ.

La deuxième méthode « configureOptions » permet d'indiquer la classe utilisée par le formulaire. Elle permet également de modifier certaines options comme la méthode http utilisée qui est par défaut du POST. Il est également possible de modifier l'action qui par défaut redirige sur la même URL que celle qui a affiché le formulaire.

```
// src/Form/AddLessonType.php

->add('category', EntityType::class, ['label' => 'Catégorie', 'class' => 'App:Category', 'choice_label' => 'name'])
```

Figure 27 Champ « category »

Cependant, certains champs doivent être plus détaillés pour ne pas générer d'erreur. C'est le cas des champs provenant d'une relation entre 2 tables. Un exemple est dans la classe « AddLessonType » qui permet d'ajouter un cours. Chaque cours doit appartenir à une catégorie renseignée dans la classe « Category ». Pour cela le champ « category » est utilisé. Cependant, le formulaire ne sait pas ce qu'il doit afficher à partir de cet objet. Il faut donc lui donner des informations détaillées : il s'agit d'une

entité « EntityType » et plus précisément de la classe « Category ». Il ne sait alors toujours pas quelle propriété il doit afficher. Le problème est réglé grâce à « choice\_label » qui permet d'indiquer laquelle est à afficher.

### 3.4.3 Création et envoi des données

Une fois la classe du formulaire configurée, il faut l'afficher dans la vue pour que l'utilisateur puisse le remplir et permettre alors de récupérer les données pour les enregistrer dans la base de données.

```
// src/Controller/AdminController.php

/**
 * @Route("/admin/addCategory", name="admin_add_category")
 */
public function addCategory(Request $request): Response
{
    $newCategory = new Category();
    $form = $this->createForm(AddCategoryType::class, $newCategory);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid())
    {
        $this->om->getManager()->persist($newCategory);
        $this->om->getManager()->flush();
        return $this->redirectToRoute('admin_home');
    }

    return $this->render('admin/add_category.html.twig', ['form' => $form->createView()]);
}
```

Figure 28 Gestion du formulaire

Dans le cas de l'ajout d'une catégorie, il faut tout d'abord créer un objet « Category » qui permettra de contenir les informations rentrées. Il faut ensuite créer le formulaire grâce à la méthode « createForm » héritée de la classe « AbstractController ». En paramètre, il faut lui donner le formulaire à utiliser ainsi que l'objet créé plutôt. La représentation visuelle du formulaire est ensuite envoyée à la vue via la méthode « createView ».

Cependant, il manque encore plusieurs points pour que ce formulaire soit utilisable. Le formulaire n'est pas traité, les informations fournies ne sont pas sauvegardées et l'affichage n'est pas géré.

Pour régler ses problèmes, il faut commencer par utiliser la méthode « handleRequest » qui permet de reconnaître si le formulaire est soumis et qui permet l'inscription des données dans l'objet « \$newCategory ». Ensuite, la méthode « isValid » détecte si les informations remplissent les critères donnés (cf 3.4.5 Validation des champs). Il reste alors à ajouter toutes ses informations dans la base de données. Pour cela, l'objet « entity manager » est utilisé, il a été au préalable instancié dans la variable « \$om » lors de la création de ce contrôleur. L'ajout des données dans la BD se fait en 2 méthodes « persist(\$newCategory) » et « flush() ». La première indique à Doctrine qu'il doit gérer l'objet « \$newCategory ». La deuxième demande à Doctrine de conserver dans la base de données tous les objets qu'il gère. Cela signifie qu'il va automatiquement générer et exécuter des requêtes correspondent aux informations indiquées pour les sauvegarder. Doctrine sait déterminer si les objets sont à ajouter ou à mettre à jour.

### 3.4.4 Affichage du formulaire

Au point précédent, le formulaire a été envoyé à la vue. Mais, il n'est toujours pas affiché et n'est donc pas utilisable.

```
{# templates/admin/add_category.html.twig #}

{{form_start(form)}}
    {{form_widget(form)}}
    <div class="text-center">
        <button class="btn btn-primary" type="submit">Ajouter</button>
    </div>
{{form_end(form)}}
```

Figure 29 Code permettant l'affichage du formulaire d'ajout de catégorie

Dans un fichier Twig, il est possible d'afficher l'entière du formulaire en écrivant : « {{ form(form) }} ». Cela affiche tous les champs et gère automatiquement les tags <form> de l'HTML. Mais cela ne permet aucune personnalisation. Twig possède plusieurs fonctions pour permettre cela. Il est possible d'utiliser les fonctions « form\_start() » et « form\_end() » pour gérer les tags du début et de la fin du formulaire.

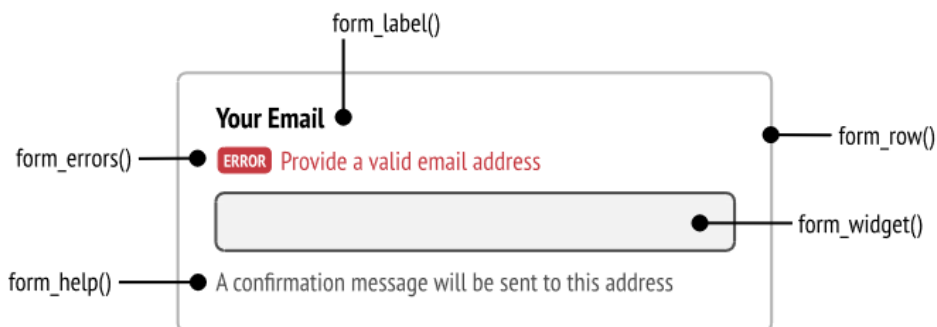


Figure 30 Les fonctions Twig permettant la personnalisation de chaque champ  
Source : [https://symfony.com/doc/current/form/form\\_customization.html](https://symfony.com/doc/current/form/form_customization.html)

D'autres fonctions permettent un affichage très précis et personnalisé pour chaque champ.

Twig permet également l'utilisation de thèmes gérant un affichage de base pour chaque type de champ.

```
# config/packages/twig.yaml

twig:
    default_path: '%kernel.project_dir%/templates'
    form_themes: ['bootstrap_4_layout.html.twig']
```

Figure 31 Configuration du thème pour les formulaires

Il est possible d'en créer un ou d'en utiliser un déjà existant. L'un des thèmes existants est celui de Bootstrap 4. Le choix du thème utilisé est à définir dans le fichier « twig.yaml », il sera alors automatiquement utilisé pour tous les formulaires. Si l'on souhaite un thème différent pour un formulaire particulier, il est possible de l'indiquer grâce au tag « form\_theme » de Twig.

### 3.4.5 Validation des champs

En fonction du contexte, chaque champ peut être soumis à des règles précises à respecter. Par exemple, un cours ne peut pas avoir un nombre négatif de participants ou il n'est pas supposé être possible de créer un cours à une date antérieure à la date du jour. Pour empêcher l'utilisateur de remplir des données incohérentes, une validation doit être effectuée sur chaque champ.

```
// src/Entity/Lesson.php

/**
 * @Assert\GreaterThan("today UTC", message = "La date doit être dans le futur.")
 * @ORM\Column(type="date")
 */
private $dateLesson;
```

Figure 32 Contrainte de la propriété « dateLesson »

Pour chaque propriété présente dans une entité, il est possible de lui définir une liste de règles qu'elle doit vérifier pour être considérée comme valide. Quand elles sont définies dans les annotations, chaque règle commence par « @Assert\ » suivi du nom de la contrainte et de ses propriétés. Dans le cas en exemple, la propriété « dateLesson » doit être plus grande que la date du jour, le cas contraire le message d'erreur « La date doit être dans le futur. » est retourné. Les contraintes de chaque propriété sont vérifiées lors de la soumission du formulaire et que la méthode « isValid » est appelée. Si au moins une des contraintes de n'importe quelle propriété testée n'est pas respectée, la méthode retournera « false » et les erreurs seront affichées.

La liste de toutes les contraintes utilisables est disponible dans la [documentation](https://symfony.com/doc/current/validation.html#constraints)<sup>7</sup> de Symfony.

```
// src/Form/ForgotPasswordType.php

$builder
    ->add('email', Null, [
        'label' => 'Mail',
        'constraints' => [
            new Email(['message' => 'Veuillez entrer une adresse mail.']),
        ]
    ])
;
```

Figure 33 Contrainte de la propriété « email »

Dans les cas où le formulaire n'est pas issu d'une entité, il est possible de directement déclarer les contraintes lors de la création de chaque champ dans le formulaire. Pour cela, il faut utiliser le paramètre « constraints » et définir toutes les contraintes dans un tableau.

<sup>7</sup> <https://symfony.com/doc/current/validation.html#constraints>

## 3.5 Le mot de passe

### 3.5.1 Génération d'un nouveau mot de passe

Lorsqu'un compte est créé ou un mot de passe est perdu, un nouveau mot de passe respectant la politique des mots de passe doit être généré automatiquement.

```
// src/Controller/MailerController.php

$random = substr(base64_encode(random_bytes(12)), 0, -2);
```

Figure 34 Génération d'un mot de passe

La méthode « random\_bytes » va permettre de générer une chaîne de caractères en octets. Cependant, les mails ne gèrent pas correctement les octets. Par conséquent, la méthode « base64\_encode » se charge d'encoder en base64. Dans cette base, une chaîne prend environ 33 % d'espace de plus que celle de base<sup>8</sup>.

À ce moment, la chaîne retournée ne respecte pas systématiquement la politique de mot de passe. Il faut alors la tester.

```
if(strlen($random) >= 12 and preg_match($patern, $random))
{
    $isValide = true;
}
```

Figure 35 Vérifie que le mot de passe respecte la politique de mot de passe

En premier la taille de la chaîne est testée avec la méthode « strlen ». Puis, un paterne Regex vérifie s'il correspond au reste de la politique de mot de passe.

```
$patern = "/(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[\W+_])/";
```

Figure 36 Paterne utilisé pour vérifier le mot de passe

Ce paterne signifie que la chaîne de caractères doit contenir au moins une lettre minuscule, une lettre majuscule, un nombre et un caractère spécial.

Si la chaîne de caractères n'est pas valide, une nouvelle chaîne est automatiquement générée. Ce processus se reproduit jusqu'à obtenir une chaîne valide. Cette opération ne consomme pas beaucoup de ressources, car la probabilité de tomber sur une chaîne valide est haute. Elle est souvent trouvée au premier essai et il faut rarement plus de 3 essais pour l'obtenir.

### 3.5.2 Envoi d'un mail

Après la génération du mot de passe, il faut envoyer le mail le contenant à l'utilisateur. Cela demande la configuration d'un serveur SMTP dans le fichier .env.

```
# .env

MAILER_DSN=smtp://9039ed8ae9cd17:be9ed92eab3595@smtp.mailtrap.io:2525?encryption=tls&auth_mode=login
```

Figure 37 Variable « MAILER\_DSN » permettant de renseigné le serveur SMTP

<sup>8</sup> <https://www.php.net/manual/fr/function.base64-encode.php>

Pour l'environnement de développement, un faux serveur SMTP est utilisé : [Mailtrap](https://mailtrap.io/)<sup>9</sup>. Il s'agit d'un outil de test pour le mail ne demandant aucun téléchargement. Il fournit les informations à rentrer pour la variable « MAILER\_DSN ».

Pour envoyer un mail via Symfony, il faut utiliser le composant « Mailer ».

```
// src/Controller/MailerController.php

public function sendEmail(User $user, $random, $template, $name)
{
    $email = (new TemplatedEmail())
        ->from('support@courscyno.com')
        ->to($user->getEmail())
        ->subject($name)
        ->htmlTemplate($template)
        ->context(['user' => $user, 'random' => $random])
        ;

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

Figure 38 Méthode gérant l'envoi d'un mail

Pour faire fonctionner cette méthode, il est obligatoire d'utiliser «Symfony\Component\Mailer\MailerInterface» et «Symfony\Bridge\Twig\Mime\TemplatedEmail». La création du mail se fait en instanciant un objet «TemplatedEmail» et en le configurant via des méthodes. La première «from» permet d'indiquer qui envoie le message, «to» permet d'indiquer le destinataire et «subject» l'objet du message. Pour le contenu, «htmlTemplate» indique le Template Twig à utiliser pour le mail et «context» les variables à transmettre à ce dernier. Il suffit ensuite d'envoyer le mail avec «send». Il est alors envoyé grâce à la configuration du serveur SMTP.

Cependant, l'aspect graphique d'un mail est géré de manière différente d'une page HTML normale. Un paquet de Twig permet de gérer automatiquement ses différences. Il est installable via la commande suivante.

```
C:\Users\eleve\TPI_InscriptionCours> composer require twig/extra-bundle twig/cssinliner-extra
```

Figure 39 installation d'un paquet de Twig

Il faut ensuite au début du Template d'envoi de mail préciser le style CSS à utiliser.

```
{# templates/emails/create_password.html.twig #}

{% apply inline_css(source('@styles/bootstrap.min.css')) %}
```

Figure 40 Ajout du style de Bootstrap

Le paquet se charge ensuite du reste, il suffit de créer le mail de manière similaire à une page HTML.

<sup>9</sup> <https://mailtrap.io/>



## 3.6 Mise en production

### 3.6.1 Les variables d'environnement

Pour mettre un site en production, il faut commencer par modifier les variables d'environnement afin de renseigner celles utilisées pour la production. Dans ce projet, il s'agit de la variable de connexion à la base de données et la variable renseignant le serveur SMTP. Les 2 sont situés dans le fichier `.env`.

Aucune capture d'écran n'est utilisée pour montrer ses modifications, car elle contient des informations privées de l'ETML.

Après ses modifications, il est possible de générer un fichier « `.env.local.php` » grâce à une commande.

```
C:\Users\eleve\TPI_InscriptionCours>composer dump-env prod
Successfully dumped .env files in .env.local.php
```

Figure 41 Création du fichier `.env.local.php`

Il contient les configurations présentes dans le fichier `.env` et il est plus optimisé.

Un problème peut être rencontré lors de cette phase concernant les informations du serveur de production. Il est possible que certaines informations contenant des caractères spéciaux soient utilisées. Comme il peut s'agir de caractère réservé, il faut les encoder grâce à la méthode « `urlencode` » pour pouvoir les utiliser sinon une erreur se produira. Il faut également dans le fichier « `config/packages/doctrine.yaml` » supprimer le préfixe « `resolve:` » pour « `url: '%env(resolve:DATABASE_URL)%'` ». Les variables sont alors configurées pour la mise en production.

### 3.6.2 Transfert des fichiers

Pour ce projet, je n'avais pas l'accès à la console du serveur. Normalement, le plus simple serait de cloner les fichiers via Git sur le serveur, puis d'y installer les dépendances via la commande « `composer install` » qui se charge de les mettre dans le dossier « `vendor` ».

Sans l'accès à la console, il faut transférer tous les fichiers y compris le contenu du dossier « `vendor` » via un client FTP. Ce transfert peut prendre assez longtemps, car le dossier « `vendor` » peut contenir plusieurs milliers de fichiers.

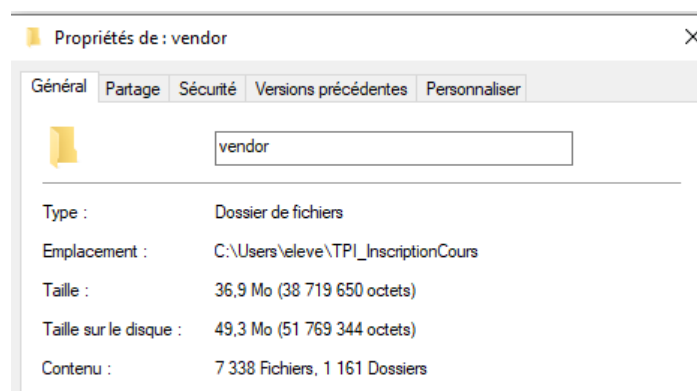


Figure 42 propriété du dossier `vendor` : 7338 fichiers

Après ce transfère, en tentant d'accéder au site web, l'erreur 500 qui indique une erreur provenant du serveur apparaîtra. Il faut obligatoirement nettoyer le cache du serveur pour résoudre ce problème.

```
C:\Users\eleve\TPI_InscriptionCours>php bin/console cache:clear --env=prod
```

```
// Clearing the cache for the prod environment with debug false
```

```
[OK] Cache for the "prod" environment (debug=false) was successfully cleared.
```

Figure 43 nettoyage du cache (capture d'écran réalisée sur le serveur de développement et non de production)

Pour réaliser cette commande, il faut avoir accès à la console du serveur de production. Dans mon cas, j'ai dû demander l'aide de ma cheffe de projet qui avait cet accès.

Cette commande doit être réalisée à chaque modification de fichiers sur le serveur de production.

### 3.6.3 .htaccess

Le site n'est pas encore fonctionnel, car seul le contenu de la page d'accueil est disponible. Une erreur se produit pour chaque tentative d'accès à une autre page.

Pour régler ce problème, il faut ajouter un fichier .htaccess dans le dossier public qui permettra de configurer la redirection vers les pages. Le fichier peut être automatiquement généré via une commande.

```
C:\Users\eleve\TPI_InscriptionCours> composer require symfony/apache-pack
Using version ^1.0 for symfony/apache-pack
```

Figure 44 Génération du fichier .htaccess

Le fichier sera automatiquement créé dans le dossier public. Si la commande n'a pas été faite sur le serveur de production, il suffit d'ajouter le fichier dans le dossier public du serveur.

Si tous les points sont correctement réalisés, le site web est en production et complètement fonctionnel. Lien du site : <https://registration.section-inf.ch>

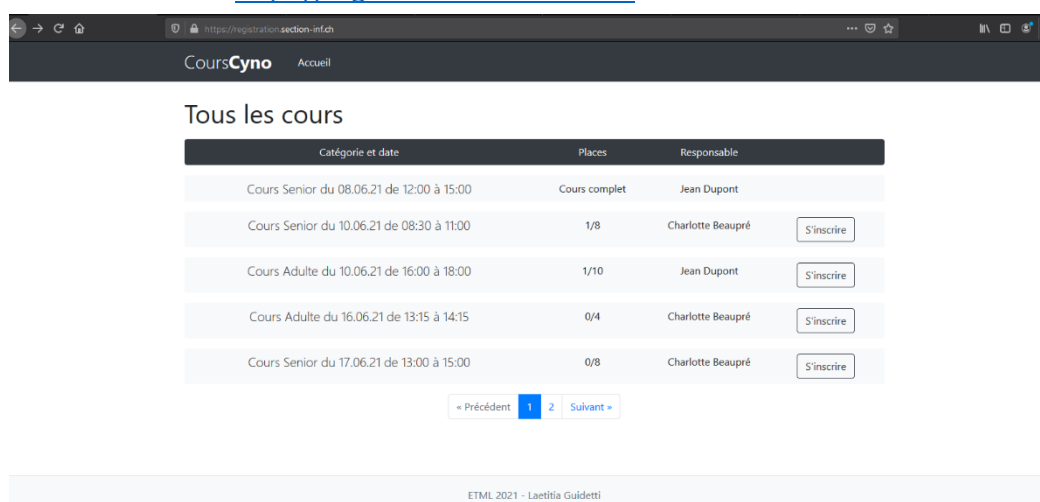


Figure 45 Site web du TPI en production

Des modifications d'affichage peuvent avoir été effectuées entre cette capture d'écran et le rendu du TPI.

## 4 Tests

### 4.1 Éléments modifiés

Dans les tests qui vont suivre, certains points des descriptions des tests peuvent ne plus être actuels, car une mise à jour a eu lieu. Ces derniers sont donc renseignés ici.

Avant	Actuellement	Raison de la modification
/admin/home	/admin/cours	Traduction des noms des URLs en français.
/admin/addLesson	/admin/ajoutCours	Traduction des noms des URLs en français.
/register:{id}	/inscription :{id}	Traduction des noms des URLs en français.
La modification d'informations personnelles contenait 3 champs	Elle contient 4 champs, le champ « Mot de passe actuel » a été ajouté.	Cette fonctionnalité permet de vérifier qu'il s'agit bien de la bonne personne qui souhaite faire la modification.
Le changement de mot de passe contenait 2 champs	Elle contient 3 champs, le champ « Ancien mot de passe »	Cette fonctionnalité permet de vérifier qu'il s'agit bien de la bonne personne qui souhaite faire la modification.

### 4.2 Page d'accueil

Tests réalisés le 19.05.2021.

1	Affichage des cours
<b>But</b>	Vérifier que tous les cours s'affichent sauf ceux passés sur la page d'accueil.
<b>Action</b>	Comparer les cours affichés sur la page d'accueil avec les entrées présentes dans la BD.
<b>Résultat attendu</b>	Tous les cours avec une date supérieure ou égale au 19.05.2021 présents dans la BD sont affichés.
<b>Résultat</b>	Les cours s'affichent triés par jour puis par heure. Toutes les entrées de la table « lesson » dont la date est égale ou supérieure au 19.05.2021 s'affichent.

2	Affichage du bouton d'inscription
<b>But</b>	Vérifier que le bouton d'inscription ne s'affiche pas pour le jour actuel.
<b>Action</b>	Vérifier que pour tous les cours du 19.05.2021 le bouton « s'inscrire » n'est pas affiché sur la page d'accueil.

<b>Résultat attendu</b>	Le bouton d'inscription ne doit pas être affiché pour les cours du 19.05.2021.
<b>Résultat</b>	Il n'est pas affiché pour les cours du 19.05.2021.

<b>3</b>	<b>Cours complet</b>
<b>But</b>	Affichage de « Cours complet » lorsqu'un cours est complet.
<b>Action</b>	Regarder dans la BD tous les cours ayant autant d'inscrits que de places disponibles. Puis, sur la page d'accueil vérifier que le texte « Cours complet » s'affiche à la place du nombre de places disponibles.
<b>Résultat attendu</b>	« Cours complet » doit être affiché pour les cours complets.
<b>Résultat</b>	Le texte s'affiche pour les cours complets et le bouton d'inscription disparaît.
<div> Cours Ado du 26.05.21 de 14:30 à 16:00 Cours complet Jean Dupont </div>	

<b>4</b>	<b>Pagination</b>
<b>But</b>	La pagination est automatique.
<b>Action</b>	Vérifier que la pagination est automatique en ajoutant des cours à la BD. Cela devrait automatiquement ajouter une page.
<b>Résultat attendu</b>	La pagination doit être automatique en fonction du nombre d'entrées dans la BD.
<b>Résultat</b>	Le nombre de pages est passé de 2 à 3. La pagination est toujours fonctionnelle.

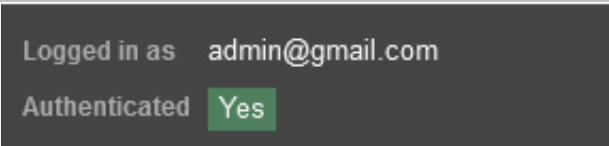
<b>5</b>	<b>Responsivité</b>
<b>But</b>	La page doit être visuellement correcte sur mobile.
<b>Action</b>	Grâce à la vue adaptive (Ctrl + shift + m sur Firefox), passer la largeur de l'écran à 300 pixels et vérifier comment les éléments s'affichent.
<b>Résultat attendu</b>	Un visuel agréable sur téléphone.
<b>Résultat</b>	La barre de navigation devient un menu hamburger. Les cours ne s'affichent plus sous la forme d'un tableau, mais sous forme de blocs les uns après les autres.

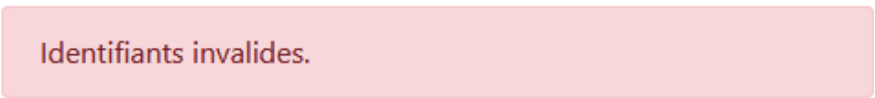
### 4.3 Connexion d'un administrateur

Tests réalisés le 19.05.2021.

<b>6</b>	<b>Accessibilité à la page de connexion de l'administrateur</b>
<b>But</b>	Accéder au formulaire de connexion de l'administrateur.
<b>Action</b>	Être sur la page d'accueil et ajouter « /admin » dans l'URL.
<b>Résultat attendu</b>	Arriver sur une page contenant un formulaire de connexion.

<b>Résultat</b>	Une page proposant de se connecter s'affiche.
-----------------	---

<b>7</b>	<b>Connexion</b>
<b>But</b>	Se connecter en tant qu'administrateur.
<b>Action</b>	Remplir le formulaire avec comme mail « admin@gmail.com » et comme mot de passe « 123456 ». Il s'agit d'un utilisateur de test qui doit être présent dans la table « user » avec le rôle « ROLE_ADMIN ».
<b>Résultat attendu</b>	La connexion se fait et l'utilisateur est redirigé vers une page d'administrateur.
<b>Résultat</b>	La connexion est réussie et l'utilisateur est redirigé vers la page /admin/home.
	

<b>8</b>	<b>Vérification des erreurs</b>
<b>But</b>	Impossible de se connecter avec des identifiant incorrecte.
<b>Action</b>	Remplir le formulaire avec des identifiants non présent dans la BD
<b>Résultat attendu</b>	La connexion échoue et un message d'erreur s'affiche.
<b>Résultat</b>	Le texte « Identifiants invalides » s'affichent si les informations ne sont pas présentes. Il n'est pas possible de soumettre le formulaire si un champ est vide.
	

## 4.4 Gestion des cours

Tests réalisés le 26.05.2021 sur la page possédant l'URL /admin/home.

<b>9</b>	<b>Accessibilité de la page</b>
<b>But</b>	Vérifier que seul un admin peut accéder à la page gestion des cours.
<b>Action</b>	Taper l'URL /admin/home sans être connecté.
<b>Résultat attendu</b>	Redirection de l'utilisateur vers une autre page.
<b>Résultat</b>	L'utilisateur est automatiquement redirigé vers la page /admin qui demande la connexion. Dans la barre de navigation l'onglet « Gestion des cours » n'est pas visible pour un utilisateur déconnecté.

<b>10</b>	<b>Affichage des participants</b>
<b>But</b>	Vérifier que l'administrateur peut voir les participants à un cours.

<b>Action</b>	Appuyer sur le bouton « Les participants » de l'un des cours.
<b>Résultat attendu</b>	Une liste de participants s'affiche contenant le prénom, le nom et le nom du chien de chaque inscrit.
<b>Résultat</b>	La liste de tous les participants du cours s'affiche en appuyant sur le bouton. Appuyer à nouveau dessus fait disparaître la liste. Le message « Aucun participant » est affiché à la place du bouton si le cours ne contient pas encore de participant.

Cours Ado du 26.05.21 de 14:30 à 16:00

Places : 4

Jean Dupont

Les participants

Prénom et nom	Nom du chien
Marie Boulanger	Medor
Dupont Jean	Chienchien
Hugo DuCommun	Milou
Jeremiah Steiner	Smoko

Cours Adulte du 28.05.21 de 14:00 à 15:00

Places : 7

Jean Dupont

Aucun participant

<b>11</b>	<b>Ajout d'un cours</b>
<b>But</b>	Il est possible d'ajouter un cours.
<b>Action</b>	Appuyer sur le bouton « Ajouter un cours » sur la page /admin/home et tenter d'ajouter un cours.
<b>Résultat attendu</b>	Le cours est correctement ajouté et un message de confirmation s'affiche.
<b>Résultat</b>	Le bouton redirige vers l'URL /admin/addLesson contenant un formulaire. Une fois rempli correctement et soumis. Le cours ajouté est visible sur la page /admin/home et un message confirmant l'ajout du cours s'affiche.

Vous avez correctement ajouté le cours Adulte du 03.06.2021 de 13:00 à 16:00	
--	--

<b>12</b>	<b>Vérification des erreurs</b>
<b>But</b>	Il n'est pas possible d'ajouter des cours contenant des options impossibles.
<b>Action</b>	Inscrire un jour passé, un nombre de participants négatif et une heure de fin avant l'heure de début du cours dans le formulaire de l'URL /admin/addLesson.
<b>Résultat attendu</b>	La tentative d'ajout échoue et des messages d'erreurs apparaissent.
<b>Résultat</b>	Des erreurs apparaissent pour chacun des cas.

Date du cours

ERREUR La date doit être dans le futur.

11.05.2021

Heure

ERREUR Le cours doit commencer avant l'heure de fin.

16:00

à

15:00

Résponsable

Jean Dupont

Catégorie

Ado

Nombre de participants

ERREUR Le nombre de participant doit être supérieur à 0.

-7

Ajouter

## 4.5 Inscription à un cours

Tests réalisés le 26.05.2021.

13	Inscription à un cours
<b>But</b>	Un utilisateur peut s'inscrire à un cours disponible.
<b>Action</b>	Appuyer sur le bouton « S'inscrire » d'un des cours disponibles sur la page d'accueil et tenter de s'inscrire au cours. Le test se déroule sur le cours possédant l'id 18.
<b>Résultat attendu</b>	L'inscription est réussie et un message de confirmation s'affiche.
<b>Résultat</b>	L'utilisateur est redirigé vers l'URL /register:18 contenant un formulaire. Les informations concernant le cours sont également présentes. Une fois le formulaire envoyé en appuyant sur « S'inscrire » un message confirmant l'inscription s'affiche.

## Inscription

Cours Adulte du 27.05.21 de 14:00 à 15:00

Prénom

Nom

Nom du chien

[S'inscrire](#)


Après avoir appuyé sur « S'inscrire »

Vous êtes inscrit au cours Adulte du 27.05.2021 de 14:00 à 15:00 ✕

14	Tenter de s'inscrire à un cours indisponible
<b>But</b>	Vérifier qu'il n'est pas possible de s'inscrire à un cours indisponible (cours complet ou passé).
<b>Action</b>	Lors de l'inscription à un cours, dans l'URL l'id du cours est écrit. Rentrer les URLs /register:7 et /register:11 pour accéder aux cours qui possèdent l'id 7 et l'id 11. Le cours possédant l'id 7 s'est déroulé le 20 mai 2021 et le cours possédant l'id 11 est complet.
<b>Résultat attendu</b>	L'utilisateur est informé qu'il n'est pas possible de s'inscrire.
<b>Résultat</b>	Dans les 2 cas, la page affiche le message « Il n'est pas possible de s'inscrire à ce cours ».
	<div style="text-align: center;"> <p>Il n'est pas possible de s'inscrire à ce cours</p> <p><a href="#">Retour</a></p> </div>


15	Vérification des erreurs
<b>But</b>	Vérifier que seul un nom et un prénom valide peuvent être entrés.
<b>Action</b>	Sur le formulaire d'inscription un prénom invalide est rentré (contenant des chiffres ou caractères spéciaux) et un nom valide est rentré (contenant des lettres, espaces, apostrophe ou tiret). Prénom invalide : Jéré45mia435h: Nom valide : Steiner Du'Commun-Dit-Veron.



<b>Résultat attendu</b>	L'inscription échoue et un message d'erreur apparaît.
<b>Résultat</b>	Pour le prénom, le message d'erreur « Veuillez entrer un prénom valide » apparaît. Pour le nom, aucun message d'erreur n'apparaît.
<div> <div>Prénom</div> <div>ERREUR Veuillez entrer un prénom valide</div> <div>Jéré45mia435h: </div> </div> <div> <div>Nom</div> <div>Steiner Du'Commun-Dit-Veron</div> </div>	

## 4.6 Gestion du profil

Tests réalisés le 26.05.2021 sur la page possédant l'URL /admin/profil.

<b>16</b>	<b>Modification d'informations personnelles</b>
<b>But</b>	Modifier ses informations personnelles.
<b>Action</b>	Rentrer un mail, prénom et nom dans le formulaire « Modifier mes informations personnelles » et le soumettre. Les informations doivent être valides.
<b>Résultat attendu</b>	Les modifications s'appliquent et un message de confirmation s'affiche
<b>Résultat</b>	Les informations sont correctement modifiées et le message « Modification d'informations effectuées » apparaît.
<div>Modification d'informations effectuées </div>	

<b>17</b>	<b>Informations invalides</b>
<b>But</b>	Vérifier que seules des informations valides peuvent être entrées.
<b>Action</b>	Rentrer un texte qui n'est pas un mail et rentrer des caractères spéciaux ou des nombres pour le prénom et le nom. Mail : pas un mail, Prénom : test42, Nom : ._wqe
	La modification échoue et des messages d'erreurs apparaissent.
<b>Résultat</b>	Pour chaque champ, une erreur indique que l'entrée n'est pas valide.

Email

ERREUR Cette valeur n'est pas une adresse email valide.

Prénom

ERREUR Veuillez entrer un prénom valide

Nom

ERREUR Veuillez entrer un nom valide

18	Modifier son mot de passe
<b>But</b>	Modifier son mot de passe.
<b>Action</b>	Entrer 2 fois le mot de passe « 3c2F.erUZ.5+ » dans le formulaire « Modifier le mot de passe » et le soumettre. Puis se déconnecter et tenter de se reconnecter.
<b>Résultat attendu</b>	La modification de mot de passe s'effectue correctement et un message de confirmation s'affiche.
<b>Résultat</b>	Le message « Mot de passe modifié » apparaît. Une fois déconnecté, il est possible de se connecter avec le nouveau mot de passe, mais pas l'ancien.
<div>Mot de passe modifié</div>	

19	Mot de passe invalide
<b>But</b>	Inscrire un mot de passe invalide.
<b>Action</b>	Entrer les mots de passe suivants : <ol style="list-style-type: none"> <li>2 mots de passe différents : hE5T3 k8?Ns( et +^ah&amp;Y4r22YJ</li> <li>Moins de 12 caractères : t5Uk[7</li> <li>Pas de minuscules : ;Y7:5U8_DW7</li> <li>Pas de majuscules : h=&lt;)29@r7d3d</li> <li>Pas de chiffres : MQuXb^^pfC {</li> <li>Pas de caractères spéciaux : B27hv3KShhF2</li> </ol>
<b>Résultat attendu</b>	Le changement de mot de passe échoue et des messages d'erreurs apparaissent.
<b>Résultat</b>	Dans tous les cas un message d'erreur apparaît.
<b>Cas 1</b>	<div>ERREUR Les 2 mots de passe ne sont pas identiques</div>
<b>Cas 2</b>	<div>ERREUR Le mot de passe doit faire au minimum 12 caractères.</div>

<b>Cas 3 à 6</b>	<b>ERREUR</b> Le mot de passe doit contenir au moins un chiffre, une lettre minuscule, une lettre majuscule et un caractère spécial.
------------------	--

## 4.7 Gestion des administrateurs

Tests réalisés le 31.05.20201.

<b>20</b>	<b>Affichage des administrateurs</b>
<b>But</b>	Vérifier que seulement les administrateurs sont affichés sur la page.
<b>Action</b>	Aller sur la page /admin/adminGestion, comparer la liste affichée avec la liste des utilisateurs présentent dans la BD, seul ceux possédant le rôle « ROLE_ADMIN » doivent être affichés.
<b>Résultat attendu</b>	Uniquement les utilisateurs possédant le rôle « ROLE_ADMIN » sont affichés.
<b>Résultat</b>	Le responsable « Jean Dupont » qui ne possède pas le rôle « ROLE_ADMIN » n'est pas affiché. Tous les autres possèdent ce rôle sont affichés.

<b>21</b>	<b>Ajout d'administrateur</b>
<b>But</b>	Possibilité d'ajouter un administrateur.
<b>Action</b>	Sur la page /admin/adminGestion, appuyer sur le bouton « Ajouter un administrateur » et remplir le formulaire avec des informations valides. Mail : <a href="mailto:test.test@gmail.com">test.test@gmail.com</a> , prénom : test, nom : test
<b>Résultat attendu</b>	L'ajout se fait correctement et un message de confirmation s'affiche.
<b>Résultat</b>	Le formulaire est soumis et un message confirmant la réussite de l'inscription est affiché.
	<div>Vous avez créé l'administrateur test test</div>

<b>22</b>	<b>Informations invalides</b>
<b>But</b>	Les informations invalides ne doivent pas être acceptées.
<b>Action</b>	Sur la page /admin/adminGestion, appuyer sur le bouton « Ajouter un administrateur » et remplir le formulaire avec des informations invalides. Mail : test@gma, prénom : t232t, nom : .Test23
<b>Résultat attendu</b>	L'ajout d'administrateur échoue et des messages d'erreurs apparaissent.
<b>Résultat</b>	Le formulaire affiche des messages d'erreurs.

E-mail	ERREUR Cette valeur n'est pas une adresse email valide.
<input type="text" value="test@gma"/>	!
Prénom	ERREUR Veuillez entrer un prénom valide.
<input type="text" value="t232t"/>	!
Nom	ERREUR Veuillez entrer un nom valide.
<input type="text" value=".Test23"/>	!

23	Administrateur existant
But	Vérifier qu'il est impossible de créer un utilisateur dont l'adresse mail existe déjà.
Action	Sur la page /admin/adminGestion, appuyer sur le bouton « Ajouter un administrateur » et remplir le formulaire avec le mail d'un utilisateur déjà inscrit. Mail : test.test@gmail.com.
Résultat attendu	L'ajout d'administrateur échoue et un message d'erreur apparaît.
Résultat	Un message d'erreur s'affiche.
<div>Un administrateur avec l'adresse mail test.test@gmail.com existe déjà.</div>	

24	L'administrateur créé reçoit son mot de passe
But	L'administrateur doit avoir reçu son mot de passe pour pouvoir se connecter par mail.
Action	Via Mailtrap, vérifier qu'un mail a été envoyé au bon utilisateur lors de la création d'un compte et qu'il contient un mot de passe. La connexion est ensuite testée avec les informations reçues.
Résultat attendu	Un mail contenant un mot de passe fonctionnel est reçu à l'adresse mail de l'administrateur créé.
Résultat	Un mail provenant de « support@courscyno.com » est présent et il contient le mot de passe « hj1WBq+xcK5LLA ». La connexion est réussie et l'utilisateur a accès aux fonctionnalités d'un administrateur.



## 4.8 Mot de passe perdu

Tests réalisés le 31.05.20201.

25	Mot de passe perdu
<b>But</b>	Envoi d'un mail contenant un nouveau mot de passe à l'administrateur qui a perdu le sien.
<b>Action</b>	Sur la page /admin, appuyer sur « mot de passe oublié ? » et remplir le formulaire avec le mail d'un utilisateur présent dans la BD. Vérifier que le mot de passe reçu par mail est fonctionnel. Mail : test.test@gmail.com
<b>Résultat attendu</b>	Un mail contenant un mot de passe fonctionnel est reçu à l'adresse mail de l'administrateur qui a oublié son mot de passe.
<b>Résultat</b>	Un mail provenant de « support@courscyno.com » est présent et il contient un nouveau mot de passe « gXH3HaxJ0/YgH7 ». La connexion est ensuite réussie.





26	Informations invalides
<b>But</b>	Un mail invalide ne doit pas être accepté.
<b>Action</b>	Sur la page /admin, appuyer sur « mot de passer oublié ? » et remplir le formulaire avec le mail invalide. Mail : testagmail.com.
<b>Résultat attendu</b>	La soumission du formulaire échoue et un message d'erreur apparait.
<b>Résultat</b>	Le formulaire affiche un message d'erreur.
	<p>Mail</p> <p><b>ERREUR</b> Veuillez entrer une adresse mail.</p> <p>testagmail.com</p>

## 4.9 Tests de production

Pour tester la production, tous les tests réalisés précédemment sont refaits sur le serveur de production. Les données de la BD de production n'étaient pas identiques à celles de la BD de développement, les tests demandant des données se feront sur des données similaires.

Seules les différences avec les tests précédents sont renseignées. Le numéro de chaque test permet d'identifier à quel test d'implémentation il se réfère.

Numéro	Différences	Commentaires
1	Aucune	
2	Aucune	
3	Aucune	

4	Aucune	
5	Aucune	Le test n'a pas été réalisé via la vue adaptative mais via des téléphones (un Sony Xperia 10 et un iPhone 7).
6	Aucune	
7	La page de redirection lors d'une connexion n'est plus /admin/home, mais /admin/cours.	
8	Aucune	
9	Aucune	L'URL testée est /admin/cours et pas /admin/home.
10	Aucune	
11	Aucune	L'URL d'ajout de cours est /admin/ajoutCours.
12	Aucune	
13	Aucune	L'URL d'inscription est /inscription:{id}.
14	Aucune	
15	Aucune	
16	Un champ « mot de passe actuel » est ajouté. Il doit être équivalent au mot de passe actuel pour permettre de valider le formulaire.	Fonctionnalité ajoutée pour des raisons de sécurité.
17	Une erreur de plus apparaît si le mot de passe n'est pas équivalent au mot de passe actuel. 	
18	Un champ « ancien mot de passe » est ajouté. Il doit être équivalent au mot de passe actuel pour permettre de valider le formulaire.	
19	Une erreur de plus apparaît si le mot de passe n'est pas équivalent au mot de passe actuel. 	
20	Aucune	
21	Aucune	
22	Aucune	
23	Aucune	
24	Aucune	Pas besoin d'utiliser Mailtrap, le mail est réellement envoyé à l'adresse mail inscrite.
25	Aucune	
26	Aucune	

## 5 Conclusion

### 5.1 Problèmes rencontrés

Durant la partie réalisation du projet, j'ai rencontré assez peu de difficultés. Je trouvais la solution à la majorité des problèmes après un peu de temps de recherche. Cela est dû à mon pré-TPI où j'avais réalisé une application Symfony. J'y avais rencontré de nombreuses difficultés que j'avais pris beaucoup de temps à résoudre. Par conséquent, lors de ce TPI, je connaissais à l'avance la résolution de nombreux problèmes. J'ai également appris lors du pré-TPI les possibilités et limites de Symfony ainsi que la méthode à utiliser pour trouver des informations dans la documentation. Tout cela m'a permis d'éviter un grand nombre de problèmes.

Cependant, j'ai rencontré une grosse difficulté durant ce projet : la mise en production. Je n'avais alors jamais mis en ligne un site utilisant un framework PHP. Ce dernier complique cette mise en ligne, il ajoute des variables d'environnement, le dossier public et le cache. Comme je n'avais pas l'accès aux logs, il était compliqué de comprendre d'où provenaient les erreurs. Il y avait également beaucoup moins de documentation que pour la partie de développement.

Finalement, la mise en ligne a réussi notamment grâce à l'aide de ma cheffe de projet. Mais, d'un point de vue technique faire une mise en production d'un projet Symfony prend environ 10 minutes, il y a très peu de modifications à faire si elles sont connues. Cependant, comme je ne les connaissais pas, cela m'a pris plusieurs heures.

### 5.2 Résultat des fonctionnalités

Résumé des fonctionnalités :

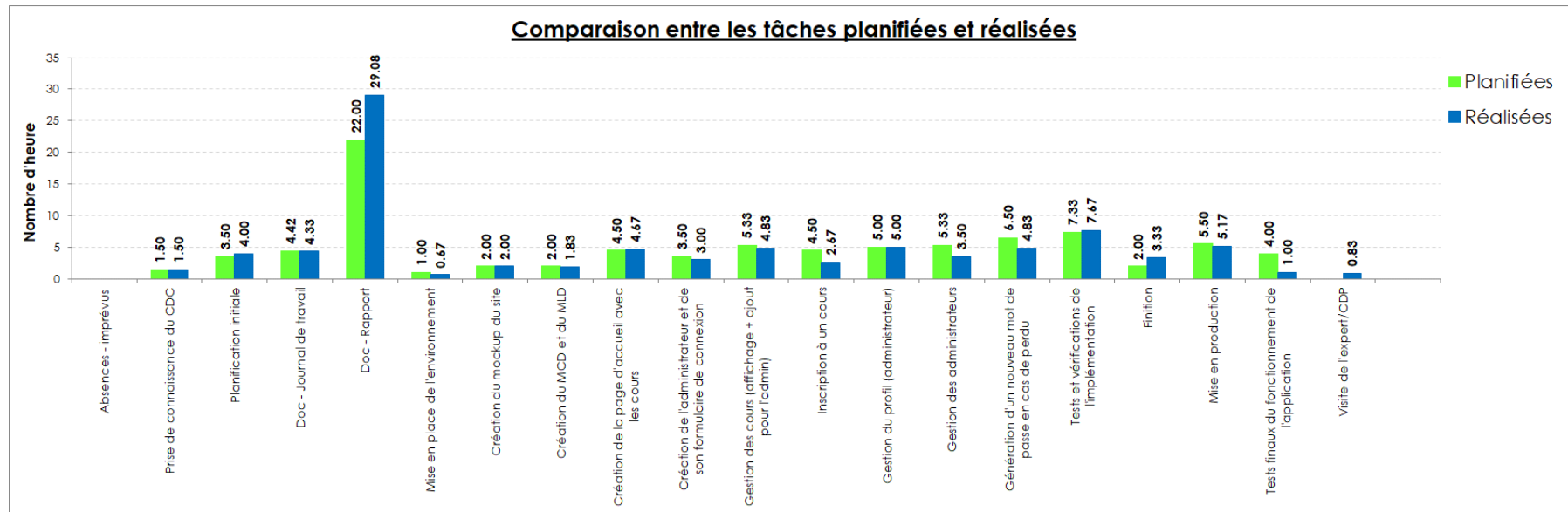
Fonctionnalité	Résultat
Affichage d'une liste de cours selon des critères	Valide
Inscription à un cours	Valide
Authentification des administrateurs	Valide
Gestion des accès	Valide
Gestion des cours (listes de participants et ajout de cours)	Valide
Gérer son profil	Valide
Gérer les administrateurs	Valide
Gestion d'un mot de passe perdu	Valide
Mise en production	Valide

Les fonctionnalités prévues ont toutes pu être implémentées.

La gestion des administrateurs est même un peu plus poussée que prévu avec la possibilité d'ajouter des administrateurs et l'envoi automatique d'un mail à ces derniers.



### 5.3 Bilan de la planification



La planification initiale a globalement été respectée. Cela est probablement dû à ma connaissance de Symfony qui m'a permis de déterminer une durée des tâches cohérentes. On remarque tout de même que la réalisation a pris moins de temps que prévu, ce qui a permis d'augmenter le temps passé sur le rapport. Lors de la planification, le passage des experts ou du CDP avait cependant été oublié, mais cela n'a pas eu d'impact sur la réalisation du projet.

## 5.4 Améliorations possibles

Ce projet remplit l'objectif qui lui était donné « fournir une application permettant de gérer les inscriptions des membres aux différents cours proposés par un club pour l'éducation de leur chien ». Cependant, certains points peuvent être améliorés.

En premier, il s'agirait de donner la possibilité aux utilisateurs de se créer un compte. Quand un utilisateur s'inscrit à un cours un message de confirmation apparaît. Mais, si jamais il oublie l'heure d'un cours auquel il s'est inscrit, il n'a aucun moyen de le retrouver. La création d'un compte pour les utilisateurs permettrait de résoudre ce problème. Cela donnerait la possibilité de créer un onglet « mes cours » dans lequel chaque utilisateur pourrait voir les cours auxquels il est inscrit. Ceci permettrait également d'ajouter une fonctionnalité pour se désinscrire d'un cours au cas où un empêchement se produirait.

L'ajout de cette fonctionnalité ne poserait pas de problème pour la gestion des accès. L'accès aux pages d'administration demande le rôle « ROLE\_ADMIN » qui n'est pas un rôle par défaut. Il suffirait donc de donner le rôle « ROLE\_USER » aux nouveaux utilisateurs pour gérer les accès. Les responsables non-administrateurs possèdent également un rôle spécifique « ROLE\_MANAGER ». Il pourrait être utilisé pour donner des accès spéciaux comme pouvoir voir les participants inscrits à leur cours.

Le deuxième point concerne la fonctionnalité du mot de passe oublié. Actuellement, un nouveau mot de passe est généré, puis envoyé. Il serait plus logique et pratique pour les utilisateurs d'envoyer un mail contenant un lien permettant de définir un nouveau mot de passe. Symfony possède notamment un paquet « [SymfonyCastsResetPasswordBundle](https://symfony.com/doc/current/security/reset_password.html)<sup>10</sup> » qui génère automatiquement une bonne partie du code de cette fonctionnalité. Pour l'utiliser, il faut que l'authentification utilise « Security » ce qui est le cas dans ce projet.

## 5.5 Bilan Personnel

Ce projet m'a permis d'améliorer ma connaissance de Symfony. J'avais de manière générale une connaissance de base suffisante pour réaliser ce TPI.

J'ai surtout pu apprendre la mise en production d'un projet utilisant un Framework PHP, ce que je n'avais jamais eu l'occasion de faire. Cela a été ma principale source de difficultés et de stress. Dès la planification, j'avais prévu de nombreuses heures pour le réaliser, car je savais que c'était le point où je pouvais avoir le plus de difficultés. Mais finalement, j'ai été très satisfaite lorsque la mise en production a fini par être fonctionnelle.

J'ai plutôt apprécié ce projet, car j'ai la satisfaction d'avoir réalisé un site fonctionnel du début à la fin.

Je remercie ma cheffe de projet, Cindy Hardegger, pour avoir toujours répondu à mes questions et m'avoir soutenu. Je remercie également mes experts pour leurs suivis de mon projet.

---

<sup>10</sup> [https://symfony.com/doc/current/security/reset\\_password.html](https://symfony.com/doc/current/security/reset_password.html)

## 6 Glossaire

Mot	Définition
<b>BD</b>	Abréviation de base de données.
<b>CSS</b>	Langage permettant de gérer le style des pages HTML.
<b>Framework</b>	Ensemble de composants qui servent de base à une partie ou à tout un logiciel. (exemple de Framework : Bootstrap, Symfony)
<b>MVC</b>	Abréviation de Modèle-Vue-Contrôleur. Il s'agit d'une architecture logicielle utilisée par Symfony qui sépare dans des fichiers distincts la logique.
<b>ORM</b>	C'est une couche d'abstraction qui synchronise les objets PHP avec une base de données relationnelles.
<b>PHP</b>	Langage utilisée pour les pages Web dynamiques. Symfony a été écrit en PHP.
<b>Route</b>	Le routage permet de définir quelle action doit être effectuée en fonction de la requête reçue par l'application Web. L'action renverra ensuite les différentes informations à afficher.
<b>SQL</b>	Langage permettant de gérer une base de données relationnelles

## 7 Sources et références

Information	Lien
<b>Mockup</b>	<a href="https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203293-mockup-definition-traduction/">https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203293-mockup-definition-traduction/</a>
<b>Doctrine ORM</b>	<a href="https://symfony.com/doc/current/doctrine.html">https://symfony.com/doc/current/doctrine.html</a>
<b>Information sur Symfony</b>	<a href="https://symfony.com/why-use-a-framework">https://symfony.com/why-use-a-framework</a>
<b>Bonnes pratiques Symfony</b>	<a href="https://symfony.com/doc/current/best_practices.html">https://symfony.com/doc/current/best_practices.html</a>
<b>Migration Symfony</b>	<a href="https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html">https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html</a>
<b>Informations sur Twig</b>	<a href="https://fr.wikipedia.org/wiki/Twig">https://fr.wikipedia.org/wiki/Twig</a>
<b>Information sur le modèle en cascade</b>	<a href="https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/">https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/</a>
<b>Gestion des accès</b>	<a href="https://symfony.com/doc/current/security.html#denying-access-roles-and-other-authorization">https://symfony.com/doc/current/security.html#denying-access-roles-and-other-authorization</a>
<b>Security.yaml</b>	<a href="https://symfony.com/doc/current/reference/configuration/security.html">https://symfony.com/doc/current/reference/configuration/security.html</a>
<b>Formulaire Symfony</b>	<a href="https://symfony.com/doc/current/forms.html">https://symfony.com/doc/current/forms.html</a>
<b>Validation Symfony</b>	<a href="https://symfony.com/doc/current/validation.html">https://symfony.com/doc/current/validation.html</a>
<b>Aléatoire PHP</b>	<a href="https://www.php.net/manual/fr/function.random-bytes.php">https://www.php.net/manual/fr/function.random-bytes.php</a>
<b>Composant Mailer</b>	<a href="https://symfony.com/doc/current/mailer.html">https://symfony.com/doc/current/mailer.html</a>
<b>Définition Framework</b>	<a href="https://fr.wikipedia.org/wiki/Framework">https://fr.wikipedia.org/wiki/Framework</a>
<b>Déploiement d'une application Symfony</b>	<a href="https://symfony.com/doc/current/deployment.html">https://symfony.com/doc/current/deployment.html</a>

## 8 Annexe

### 8.1 Cahier des charges

#### INFORMATIONS GENERALES

Candidat :	Nom : <b>GUIDETTI</b>	Prénom : <b>LAETITIA</b>
	✉ : laetitia.guidetti@eduvaud.ch	☎ : 079 560 52 57
Lieu de travail :	ETML, Sébeillon 12 1004 Lausanne	
Orientation :	<input type="checkbox"/> 88601 Développement d'application <input checked="" type="checkbox"/> 88602 Informatique d'entreprise <input type="checkbox"/> 88603 Technique des systèmes	
Chef de projet :	Nom : Hardegger	Prénom : Cindy
	✉ : cindy.hardegger@eduvaud.ch	☎ : 021 316 78 29
Expert 1 :	Nom : Oberson	Prénom : Bernard
	✉ : oberson.bernard@gmail.com	☎ : 078 708 02 96
Expert 2 :	Nom : Andolfatto	Prénom : Frédérique
	✉ : frederique.andolfatto@cpnv.ch	☎ : 077 206 66 45
Période de réalisation :	Du <b>mardi 11 mai à 8h00</b> au <b>vendredi 04 juin à 10h00</b>	
Horaire de travail : (Basé sur l'horaire officiel)	Lundi	08h00-11h25 13h10-16h35
	Mardi	08h00-11h25 12h20-15h45
	Mercredi	- 12h20-16h35
	Jeudi	08h00-12h15 13h10-16h35
	Vendredi	08h00-11h25 13h10-16h35
	<i>Attention : Toutes les demi-journées ont une pause obligatoire de 15 minutes.</i>	
Nombre d'heures :	89.9 heures	
Planning (en H ou %)	Analyse : 20%, Implémentation 40%, Tests 15%, Doc. 25%	

#### PROCÉDURE

Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour.

Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.

Le candidat a connaissance de la feuille d'appréciation avant de débiter le travail.

Le candidat est entièrement responsable de la sécurité de ses données.

En cas de problèmes graves, le candidat averti au plus vite les deux experts et son CdP.

Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.

A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

#### TITRE

Inscription à un cours

---

## MATÉRIEL ET LOGICIEL À DISPOSITION

1 PC standard de l'ETML  
Serveur web local  
Suite Microsoft Office  
Logiciel de développement PHP

---

## PRÉREQUIS

Avoir suivi les modules : 101, 104, 105, 133 et 151

Langages ou technologies à utiliser : HTML, CSS, PHP et MySQL

Utilisation d'un Framework CSS  
Utilisation du MVC  
Utilisation d'un dépôt Git

---

## DESCRIPTIF DU PROJET

L'objectif est de fournir une application permettant de gérer les inscriptions des membres aux différents cours proposés par un club pour l'éducation de leur chien. Il est important que l'application soit visualisable et fonctionnelle sur smartphone et doit respecter les normes W3C.

L'utilisation d'un Framework CSS est indispensable, il sera commenté dans le rapport.

L'utilisation d'un dépôt Git est indispensable, il sera disponible aux experts et au chef de projet. Il sera utilisé au moins 1x par jour.

### Liste des fonctionnalités de l'application

#### Authentification

L'application avec la liste des cours disponible est accessible sans connexion.

Une page de connexion est disponible en tapant « /admin » dans l'url

Seulement un administrateur peut se connecter pour :

- Gérer les administrateurs (à terme d'autres administrateurs pourront être ajoutés)
- Gérer son profil
- Gérer la liste des cours

#### Gestion des administrateurs

Pour l'instant, l'application aura un seul administrateur, mais il doit être possible à l'avenir de pouvoir en ajouter. Ce dernier se caractérise par un nom, un prénom, une adresse électronique, un droit admin, un mot de passe et le contrôle total sur l'application.

Un administrateur peut gérer l'application depuis une page « d'administration » qui sera accessible uniquement s'il est connecté. La page pour se connecter est accessible en tapant dans l'url « /admin ». Cette dernière comprend un formulaire de connexion.

Une fois connecté, la gestion de l'application comprend l'ajout de cours (voir « Gestion de la liste des cours ») et une page concernant son profil (voir « Gestion du profil »).

Si un administrateur perd son mot de passe, il peut le récupérer via un lien, « mon mot de passe perdu » qui sera affiché en dessous du formulaire de connexion. En cliquant dessus, il devra entrer des informations, puis il recevra un mail avec un nouveau mot de passe généré aléatoirement. L'apprenti expliquera la politique de mot de passe mis en place dans le

rapport, ainsi que les informations qu'un administrateur devra fournir afin de récupérer son mot de passe.

De plus, un administrateur connecté peut visualiser la liste des inscrits pour un cours.

### Gestion du profil

Une page pour gérer le profil d'un administrateur doit être mise en place. Un administrateur peut changer de nom, prénom, de mot de passe ou d'adresse électronique depuis son profil. Il n'est pas possible de supprimer son compte pour l'instant.

### Gestion de la liste des cours

Un administrateur peut ajouter des cours et voir les participants (nom du participant + nom du chien) pour chacun des cours. Pour connaître les caractéristiques d'un cours, voir les points suivants.

### Affichage des cours à disposition

En arrivant sur l'application web, tous les cours du jour même ou qui suivent sont affichées (les cours passés ne sont pas affichés).

Un cours est caractérisé par :

- Une date (ex : 27 mai 2020)
- Un horaire (ex : 17h45 à 18h45)
- Une catégorie de cours (ex : Ado)
- Un responsable de cours (ex : Jean Dupont)
- Le nombre de participant (ex : 10)
- Le nombre de places encore disponible (ex : 3)
- Un bouton d'inscription

A chaque inscription, le nombre de places disponible est modifié à la baisse (-1).

Précision sur le bouton d'inscription : Le bouton est disponible si le cours n'est pas complet (nombre de places disponible pas encore atteint) et si le cours n'a pas lieu le jour même. Si le nombre de places disponible est atteint, le bouton d'inscription disparaît et dans le nombre de place disponible, le texte « cours complet » est affiché. Si le cours a lieu le même jour, le bouton disparaît.

### Inscription pour un participant

Un participant peut s'inscrire à un cours (par exemple pour le cours du 27 mai 2020 de 17h45 à 18h45) en cliquant sur le bouton d'inscription. Il doit valider son inscription en entrant son nom (composé uniquement de lettres) et le nom de son chien.

En cas de succès, un message du type « vous êtes inscrit au cours du 27 mai 2020 de 17h45 à 18h45 » s'affiche.

En cas d'erreur, un message apparaît. Exemple de message : « Merci d'entrer un nom valide » ou « le nom existe déjà dans les inscriptions » ou « Merci d'entrer un nom pour le chien ».

Il n'est pour l'instant pas possible de savoir si un participant s'inscrit 2 fois (aucune création de compte pour les participants).

Il n'est pour l'instant pas possible de se désinscrire d'un cours.

### Mise en production

L'application doit être installée sur le serveur de la section. Les accès seront fournis ultérieurement.



## LIVRABLES

Durant le TPI, le candidat est tenu d'envoyer à son chef de projet et aux deux experts :

- La planification initiale à la fin du 1er jour de travail
- Le rapport de projet 1 fois par semaine
- Le journal de travail 2 fois par semaine
- Un accès à son dépôt Git dès que disponible.

A la fin du TPI, le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale
- Un rapport de projet
- Un journal de travail
- La modélisation de la base de données (MCD, MLD, MPD)
- L'application Web au complet
- Mise en place de l'application sur le FTP (informations de connexion fournies à part)

## POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20):

1. *Modélisation de la base de données selon le modèle Merise (MCD, MLD et MPD)*
2. *Mise en place de la vue smartphone d'un point de vue visuelle et fonctionnelle. (Définir la taille pour une vue smartphone, définir les éléments importants sur un smartphone)*
3. *Choix du Framework CSS (lequel et comparaison avec au moins un autre)*
4. *Mise en production de l'application sur le serveur de la section*
5. *Respect des normes W3C*
6. *Mise en place et utilisation d'un repository Git avec commit réguliers (min.1x par jour)*
7. *Explication de la politique de mot de passe (gestion des administrateurs)*

## VALIDATION

	Lu et approuvé le :	Signature :
Candidat :		
Expert n°1 :		
Expert n° 2 :		
Chef de projet :		



## 8.2 Résumé du rapport du TPI

### Situation de départ

L'objectif de ce TPI était de fournir une application permettant de gérer les inscriptions des membres aux différents cours proposés par un club pour l'éducation de leur chien. Cela comprend la réalisation des fonctionnalités suivantes :

- Affichage d'une liste de cours selon des critères
- Inscription à un cours
- Authentification des administrateurs
- Gestion des accès
- Gestion des cours (listes de participants et ajout de cours)
- Gérer son profil
- Gérer les administrateurs
- Gestion d'un mot de passe perdu
- Mise en production

Le projet doit utiliser git. Le site web doit être adapté pour la vue smartphone et être MVC.

### Mise en œuvre

Pour réaliser ce projet, j'ai utilisé le Framework PHP Symfony et le Framework CSS Bootstrap. Une base de données correspond à la problématique a été créée via les entités de l'ORM Doctrine. Puis les administrateurs ont été créés et configurés via le paquet « Security ». Il permet alors de chiffrer les mots de passe et réaliser une gestion des accès. Le formulaire d'inscription aux cours a ensuite été créé grâce à la fonctionnalité « form » de Symfony. Cette méthode a été réutilisée pour tous les formulaires contenus dans le projet. Pour permettre la récupération de mot de passe, une fonctionnalité a été ajoutée pour permettre l'envoi automatique d'un mail contenant un nouveau mot de passe généré aléatoirement en cas d'oubli. Pour finir, le projet a été mis en ligne sur les serveurs de l'ETML à l'adresse : <https://registration.section-inf.ch>.

### Résultats

Toutes les fonctionnalités ont pu être réalisées et testées, malgré des difficultés rencontrées lors de la mise en ligne du projet. Il a été créé de manière à pouvoir facilement ajouter de nouvelles fonctionnalités dans le futur tel que l'inscription pour les non-administrateurs.

### 8.3 Planification initiale détaillée

Séquence 1			Date: mardi, 11 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Prise de connaissance du CDC	18	1h30min			
Planification initiale	18	1h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			
Séquence 2			Date: mardi, 11 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Planification initiale	24	2h			
Mise en place de l'environnement	12	1h	Création du dépôt git et création du projet symfony		
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 3			Date: mercredi, 12 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Création du mockup du site	24	2h	Création et réflexion sur l'aspect graphique du site		
Création du MCD et du MLD	12	1h	Analyse des besoins de la base de données		
Doc - Rapport	10	0h50min			
Doc - Journal de travail	2	0h10min			
Total tranche	48	4h			
Séquence 4			Date: lundi, 17 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Création du MCD et du MLD	12	1h	Création du MCD et MLD en fonction des besoins du TPI		
Création de la page d'accueil avec les cours	24	2h	Analyse des besoins de la page et création des entités correspondantes au MLD		
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			
Séquence 5			Date: lundi, 17 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Création de la page d'accueil avec les cours	18	1h30min	Affichage dynamique des cours et aspect de la page		
Doc - Rapport	18	1h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 6			Date: mardi, 18 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Création de la page d'accueil avec les cours	12	1h	Aspect graphique de la page		
Création de l'administrateur et de son formulaire de connexion	24	2h	Analyse des besoins de l'administrateur et création des entités correspondantes		
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 7			Date: mardi, 18 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Création de l'administrateur et de son formulaire de connexion	18	1h30min	Formulaire de connexion		
Doc - Rapport	18	1h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 8			Date: mercredi, 19 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests et vérifications de l'implémentation	30	2h30min	Test du bon fonctionnement de l'affichage de la page d'accueil et de la connexion de l'administrateur		
Doc - Rapport	16	1h20min			
Doc - Journal de travail	2	0h10min			
Total tranche	48	4h			

Séquence 9			Date: jeudi, 20 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Gestion des cours (affichage + ajout pour l'admin)	40	3h20min	Création du formulaire d'ajout de cours pour l'administrateur		
Doc - Rapport	6	0h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	48	4h			

Séquence 10			Date: jeudi, 20 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Gestion des cours (affichage + ajout pour l'admin)	24	2h	Affichage des cours et des inscrits pour l'administrateur		
Doc - Rapport	12	1h			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 11			Date: vendredi, 21 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Inscription à un cours	36	3h	Analyse de l'inscription à un cours et réalisation du formulaire		
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 12			Date: vendredi, 21 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Inscription à un cours	18	1h30min	Réalisation du formulaire et de l'application de l'inscription dans la base de		
Doc - Rapport	18	1h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 13			Date: mardi, 25 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Gestion du profil (administrateur)	36	3h	Analyse de la politique du mot de passe et création du formulaire permettant de modifier les informations personnelles		
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 14			Date: mardi, 25 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Gestion du profil (administrateur)	24	2h	Formulaire permettant de modifier les informations personnelles		
Doc - Rapport	12	1h			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			
Séquence 15			Date: mercredi, 26 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests et vérifications de l'implémentation	34	2h50min	Test de la modification des informations personnelles de l'administrateur, l'inscription à un cours et de l'ajout de cours		
Doc - Rapport	12	1h			
Doc - Journal de travail	2	0h10min			
Total tranche	48	4h			
Séquence 16			Date: jeudi, 27 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Gestion des administrateurs	46	3h50min	Analyse de la gestion des administrateurs et ajout d'administrateur		
Doc - Journal de travail	2	0h10min			
Total tranche	48	4h			

Séquence 17			Date: jeudi, 27 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Gestion des administrateurs	18	1h30min	Affichage des administrateurs		
Doc - Rapport	18	1h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 18			Date: vendredi, 28 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Génération d'un nouveau mot de passe en cas de perdu	36	3h	Analyse et recherche sur les informations à demander avant une réinitialisation de mot de passe, l'envoi de mail en PHP et la génération aléatoire de mot de passe		
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 19			Date: vendredi, 28 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Génération d'un nouveau mot de passe en cas de perdu	18	1h30min	Envoi d'un mail en cas de mot de passe oublié		
Doc - Rapport	18	1h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			



Séquence 20			Date: lundi, 31 mai 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Génération d'un nouveau mot de passe en cas de perdu	24	2h	Génération aléatoire d'un mot de passe		
Doc - Rapport	12	1h			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			
Séquence 21			Date: lundi, 31 mai 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Finition	24	2h	Finition de l'application		
Doc - Rapport	12	1h			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			
Séquence 22			Date: mardi, 1 juin 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests et vérifications de l'implémentation	24	2h	Test de la gestion des administrateurs et de la perte de mots de passe		
Doc - Rapport	12	1h			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 23			Date: mardi, 1 juin 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Mise en production	36	3h	Recherche sur le fonctionnement d'une mise en production d'une application		
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			

Séquence 24			Date: mercredi, 2 juin 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Mise en production	30	2h30min	Placement du site web sur le serveur de l'ETML		
Doc - Rapport	16	1h20min			
Doc - Journal de travail	2	0h10min			
Total tranche	48	4h			

Séquence 25			Date: jeudi, 3 juin 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests finaux du fonctionnement de l'application	30	2h30min	Test de l'application en production		
Doc - Rapport	16	1h20min			
Doc - Journal de travail	2	0h10min			
Total tranche	48	4h			

Séquence 26			Date: jeudi, 3 juin 2021		après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests finaux du fonctionnement de l'application	18	1h30min	Test de l'application en production		
Doc - Rapport	18	1h30min			
Doc - Journal de travail	2	0h10min			
Total tranche	38	3h10min			
Séquence 27			Date: vendredi, 4 juin 2021		matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	20	1h40min			
Doc - Journal de travail	1	0h5min			
Total tranche	21	1h45min			

## 8.4 Journal de travail

Séquence 1			Date: mardi, 11 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Prise de connaissance du CDC	18	1h30min	Rencontre avec l'expert 1, prise de connaissance du CDC, analyse détaillée du CDC et questions précises sur le CDC au chef de projet.	
Planification initiale	20	1h40min	Création des différentes tâches	
Total tranche	38	3h10min		
Séquence 2			Date: mardi, 11 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Planification initiale	28	2h20min	Planification détaillée de chaque séquence	
Mise en place de l'environnement	8	0h40min	Initialisation du dépôt git et du projet symfony	Dépôt : <a href="https://github.com/Chywou/TPI_InscriptionCours/tree/dev">https://github.com/Chywou/TPI_InscriptionCours/tree/dev</a> commit : 4c6662d0c0ee964a223face6522db417fa4ef9ce
Doc - Journal de travail	2	0h10min	Mise à jour du JDV pour le mardi 11	
Total tranche	38	3h10min		
Séquence 3			Date: mercredi, 12 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Création du mockup du site	24	2h	Création du mockup de la page d'accueil, la page de gestion de la liste des cours et la page d'ajout d'un cours	Logiciel : Pencil
Création du MCD et du MLD	12	1h	Installation de looping pour créer le MCD et le MLD. Réflexion sur les informations à stocker dans la BD par rapport au CDC. Création d'un MCD provisoire.	<a href="https://www.looping-mcd.fr/">https://www.looping-mcd.fr/</a>
Doc - Rapport	10	0h50min	Rédaction d'information concernant les mockups	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail et envoi des avancées aux experts	
Total tranche	48	4h		

Séquence 4			Date: lundi, 17 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Création du MCD et du MLD	10	0h50min	Vérification que le MCD correspond aux attentes du CDC. Vérification du respect des règles d'élaboration d'une base de données (non-redondance, atomicité, unicité). Modification du MCD en fonction des indications du CdP	
Création de la page d'accueil avec les cours	26	2h10min	Création du Controller Home, création des Entity Cartegory, Lesson, Participant et User. Migration des Entity sur la base de données.	Commit : 2b7793678bccade0e3c44e401095b110c67e583a
Doc - Journal de travail	2	0h10min	Mise à jour du JDT	
Total tranche	38	3h10min		

Séquence 5			Date: lundi, 17 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Création de la page d'accueil avec les cours	12	1h	Affichage de manière dynamique les différents cours. Les cours sont triés par date, puis par heure. Les cours passés ne sont pas affichés.	Comparer avec la date du jour : <a href="https://stackoverflow.com/questions/10582108/how-can-i-compare-a-date-with-current-date-using-doctrine-2/11599622">https://stackoverflow.com/questions/10582108/how-can-i-compare-a-date-with-current-date-using-doctrine-2/11599622</a> , commit : d3e51cd0bfec2c8375027142270eb3b0a0e0b189
Doc - Rapport	24	2h	Informations concernant le MCD et le MLD. Text explicatif de Doctrine ORM. Début de l'explication sur les entités	
Doc - Journal de travail	2	0h10min	Mise à jour du JDT	
Total tranche	38	3h10min		

Séquence 6			Date: mardi, 18 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Création de la page d'accueil avec les cours	18	1h30min	Le contenu de la page d'accueil est responsive. Le paquet Knp_paginator permettant la pagination a été ajouté	Knp_paginator : <a href="https://github.com/KnpLabs/KnpPaginatorBundle">https://github.com/KnpLabs/KnpPaginatorBundle</a>
Création de l'administrateur et de son formulaire de connexion	18	1h30min	Création d'un administrateur, création du contrôleur "AdminController" et génération de base du formulaire de connexion.	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 7			Date: mardi, 18 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Création de l'administrateur et de son formulaire de connexion	18	1h30min	Configuration du formulaire de connexion de l'administrateur.	Commit : 8b1da63a7796fb3ac3f4ac2582efaf0c67560136
Doc - Rapport	18	1h30min	Documentation des routes, des entités et informations sur divers outils utilisés.	
Doc - Journal de travail	2	0h10min	Mise à jour du JDT	
Total tranche	38	3h10min		

Séquence 8			Date: mercredi, 19 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Tests et vérifications de l'implémentation	34	2h50min	Test de la page d'accueil et de la connexion d'un administrateur, corrections d'erreurs.	commit : 1f216419a21a573fc780ce373f78739f41cec8a8
Doc - Rapport	12	1h	Documentation des tests.	
Doc - Journal de travail	2	0h10min	Mise à jour du JDT	
Total tranche	48	4h		

Séquence 9			Date: jeudi, 20 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Gestion des cours (affichage + ajout pour l'admin)	34	2h50min	Ajout des cours pour l'administrateur	commit : 8bfee8804f5ab6acd3ddd50d4021f814a9a1f96a Formulaire: <a href="https://symfony.com/doc/current/forms.html">https://symfony.com/doc/current/forms.html</a> , Validation : <a href="https://symfony.com/doc/current/validation.html#validation-constraint-configuration">https://symfony.com/doc/current/validation.html#validation-constraint-configuration</a>
Gestion des cours (affichage + ajout pour l'admin)	12	1h	Affichage des cours et des participants de manière responsive.	Commit : 8bfee8804f5ab6acd3ddd50d4021f814a9a1f96a
Doc - Journal de travail	2	0h10min		
Total tranche	48	4h		

Séquence 10			Date: jeudi, 20 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Gestion des cours (affichage + ajout pour l'admin)	12	1h	Amélioration de l'affichage, amélioration de la responsivité des participants.	commit : ff60b6da0e11496349ddb3de669e611066c4939b
Doc - Rapport	24	2h	Documentation des requêtes et de "Security"	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 11			Date: vendredi, 21 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Inscription à un cours	8	0h40min	Création d'un formulaire d'inscription.	
Visite de l'expert/CDP	4	0h20min	Visite de l'experte Frederique Andolfatto afin de vérifier l'avancée du projet	
Inscription à un cours	24	2h	Le bouton "s'inscrire" redirige vers une page d'inscription. Il n'est pas possible de s'inscrire à un cours passé ou complet (même en modifiant l'URL). Affichage d'une alerte quand l'inscription s'est déroulée correctement. La validation du formulaire est cependant à revoir.	commit : 7e11cc6b0ad8b78ede1be801b97cb6b492b42c93 Format des dates : <a href="https://stackoverflow.com/questions/10209941/object-of-class-datetime-could-not-be-converted-to-string">https://stackoverflow.com/questions/10209941/object-of-class-datetime-could-not-be-converted-to-string</a>
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 12			Date: vendredi, 21 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Gestion du profil (administrateur)	12	1h	Génération du formulaire de base permettant à un administrateur de modifier ses données personnelles. Le formulaire est à revoir, il ne permet pas encore la modification du mot de passe.	commit : ef7502a12733492ebb1a8f0a680643b15c336256
Doc - Rapport	24	2h	Explication de la gestion des accès et du fichier security.yaml	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 13			Date: mardi, 25 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Gestion du profil (administrateur)	36	3h	Analyse de la politique de mot de passe à utiliser. Formulaire de modification des informations personnelles et du mot de passe. Création des regex permettant de vérifier que le mot de passe respecte la politique de mot de passe. Création d'un regex pour les noms et prénoms des utilisateurs. Problème : la politique de mot de passe est très stricte pour un utilisateur classique.	commit : 6bec8463d805c06b5f759699cd55dc64232d6225 Encodage du mot de passe : <a href="https://stackoverflow.com/questions/59320710/how-to-encode-a-password-in-symfony-5">https://stackoverflow.com/questions/59320710/how-to-encode-a-password-in-symfony-5</a> Regex nom : <a href="https://qastack.fr/programming/2385701/regular-expression-for-first-and-last-name">https://qastack.fr/programming/2385701/regular-expression-for-first-and-last-name</a> Politique de mot de passe : <a href="https://www.ncsc.admin.ch/ncsc/fr/home/infos-fuer/infos-private/aktuelle-themen/schuetzen-sie-ihre-konten.html">https://www.ncsc.admin.ch/ncsc/fr/home/infos-fuer/infos-private/aktuelle-themen/schuetzen-sie-ihre-konten.html</a>
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail.	
Total tranche	38	3h10min		



Séquence 14			Date: mardi, 25 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Gestion du profil (administrateur)	12	1h	Correction des regex et ajustement des messages d'erreurs	commit : 6bec8463d805c06b5f759699cd55dc64232d6225
Doc - Rapport	24	2h	Explication sur les formulaires et la politique du mot de passe (les 2 sont à compléter).	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 15			Date: mercredi, 26 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Tests et vérifications de l'implémentation	34	2h50min	Tests sur la gestion du profil, l'inscription à un cours et la connexion d'un administrateur. Correction d'un Regex, de la validation concernant l'heure du début et de fin de cours et ajout d'un message lors de l'ajout d'un cours. Il n'y a cependant pas de messages lors de l'ajout d'une catégorie ou d'un nouveau responsable. Erreur dans la planification : pour le mercredi 26 mai 2021, la tâche était "Tests finaux du fonctionnement de l'application" au lieu de "Tests et vérifications de l'implémentation". Une mauvaise sélection de case a dû être réalisée lors de la planification, car l'explication correspond à des tests d'implémentation.	commit : 045d228db23bd05310328e9e9aea1233861399ab
Doc - Rapport	10	0h50min	Documentation des tests et mise à jour de la forme des tests précédents.	
Doc - Journal de travail	4	0h20min	Mise à jour du journal de travail.	
Total tranche	48	4h		

Séquence 16			Date: jeudi, 27 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Gestion des administrateurs	28	2h20min	Analyse des besoins et questions au CDP sur les administrateurs. Recherche de tous les utilisateurs administrateur et ajout d'administrateurs. L'ajout n'est pas encore parfaitement opérationnel, car le mot de passe n'est pas généré automatiquement et envoyé par mail au nouvel admin.	commit : 7f0037a601124eda0b1c5e6f119eafad64c046c Recherche par rôle : <a href="https://openclassrooms.com/forum/sujet/symfony-4-entitytype-recuperer-user-par-role">https://openclassrooms.com/forum/sujet/symfony-4-entitytype-recuperer-user-par-role</a>
Génération d'un nouveau mot de passe en cas de perdu	18	1h30min	Analyse sur l'envoi par mail et la génération aléatoire de mot de passe. Création du formulaire et envoi d'un mail contenant un mot de passe généré de manière aléatoire. Cette fonctionnalité n'est pas terminée, le nouveau mot de passe n'est pas sauvegardé dans la BD, le mail ne possède aucune forme et il n'y a pas de vérification sur la chaîne de caractère générée (elle peut ne pas respecter la politique de mot de passe).	commit : 7f0037a601124eda0b1c5e6f119eafad64c046c Génération aléatoire : <a href="https://stackoverflow.com/questions/50877915/how-to-generate-a-token-in-symfony-3-4">https://stackoverflow.com/questions/50877915/how-to-generate-a-token-in-symfony-3-4</a>
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	48	4h		

Séquence 17			Date: jeudi, 27 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	36	3h	Explication sur les formulaires et mise à jour à de divers points.	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 18			Date: vendredi, 28 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Génération d'un nouveau mot de passe en cas de perdu	36	3h	Génération d'un mot de passe qui respecte la politique de mot de passe. Le contenu du mail est géré via un templates. Le mot de passe est mis à jour dans la BD. L'affichage des pages concernant la réinitialisation a été revu pour être responsive et plus agréable pour un utilisateur. Cependant, il n'y a pas de vérification lors de l'inscription du mail par l'utilisateur.	Commit : 9549711d29fbaa31eb3c790470feb2d06d469548 Mailler : <a href="https://symfony.com/doc/current/mailler.html">https://symfony.com/doc/current/mailler.html</a>
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 19			Date: vendredi, 28 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Génération d'un nouveau mot de passe en cas de perdu	4	0h20min	Vérification du mail et si un utilisateur correspond bien à l'utilisateur donné.	Commit : 23e0659577bf597c1c56de0717cf3de4d0a49362
Gestion des administrateurs	14	1h10min	Ajout d'un mail envoyé automatiquement lors de la création d'un administrateur pour lui fournir son mot de passe.	Commit : 23e0659577bf597c1c56de0717cf3de4d0a49362
Doc - Rapport	18	1h30min	Documentation sur la validation des données et sur l'affichage avec Symfony.	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 20			Date: lundi, 31 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Finition	30	2h30min	Traduction des noms des pages, ajout de commentaires, ajout d'une vérification lors de la pagination. Correction de la validation de donnée de l'ajout de cours et de l'ajout de responsable. Ajout d'une demande de l'ancien mot de passe pour modifier les informations personnelles ou modifier le mot de passe.	Commit : 8839e40c49335013686705c4a47c41ed6b89420d Ancien mot de passe : <a href="https://symfony.com/doc/current/reference/constraints/UserPassword.html">https://symfony.com/doc/current/reference/constraints/UserPassword.html</a>
Visite de l'expert/CDP	6	0h30min	Passage du CDP et vérification des critères d'évaluation	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

Séquence 21			Date: lundi, 31 mai 2021	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Tests et vérifications de l'implémentation	24	2h	Test sur la gestion des administrateurs et la fonctionnalité de mot de passe perdu. Vérification si le mail est déjà présent dans la BD lors de l'ajout d'un admin ou la modification de son profil.	Commit : eb5455ae34b311fccde2c04b0a6a1e756fed4222
Doc - Rapport	12	1h	Documentation des tests	
Doc - Journal de travail	2	0h10min	Mise à jour du journal de travail	
Total tranche	38	3h10min		

---

Auteur : Laetitia Guidetti		Création : 30.04.2021
Modifié par : Laetitia Guidetti	Page 76 sur 77	Impression : 04.06.2021 08:42:00
Version : 423 du 04.06.2021 08:42		R-TPI-Rapport-laeguidetti.docx

Auteur : Laetitia Guidetti  
Modifié par : Laetitia Guidetti  
Version : 423 du 04.06.2021 08:42