

Table des matières

[Avant-Propos 3](#_heading=h.gjdgxs)

[Liste des compétences couvertes par le titre 4](#_heading=h.30j0zll)

[Résumé du projet 5](#_heading=h.1fob9te)

[Outils Utilisés 5](#_heading=h.3znysh7)

[Maquettage et diagrammes 5](#_heading=h.2et92p0)

[Back-End 6](#_heading=h.tyjcwt)

[Front-End 6](#_heading=h.3dy6vkm)

[Autres 6](#_heading=h.1t3h5sf)

[Conception du site 7](#_heading=h.4d34og8)

[Questionnaire de définition de projet et Cahier des charges 7](#_heading=h.2s8eyo1)

[Veille concurrentielle 7](#_heading=h.17dp8vu)

[Diagrammes et Modélisation 9](#_heading=h.3rdcrjn)

[Maquettage 13](#_heading=h.26in1rg)

[Arborescence du site 15](#_heading=h.lnxbz9)

[Thème graphique et Design général du site 18](#_heading=h.35nkun2)

[Back-End 21](#_heading=h.1ksv4uv)

[Étape de création du projet 21](#_heading=h.44sinio)

[Création de la Base De Données 21](#_heading=h.2jxsxqh)

[Entités 22](#_heading=h.z337ya)

[Relations entre les entités 26](#_heading=h.3j2qqm3)

[Sécurité 27](#_heading=h.1y810tw)

[Front-End 32](#_heading=h.4i7ojhp)

[Materialize 32](#_heading=h.2xcytpi)

[Modèle de cartes 33](#_heading=h.1ci93xb)

[Header 36](#_heading=h.3whwml4)

[Menus de navigation 36](#_heading=h.2bn6wsx)

[Création d’un Calendrier 41](#_heading=h.qsh70q)

[Résolution d’un problème avec une documentation anglophone 45](#_heading=h.3as4poj)

[Mise en production 50](#_heading=h.1pxezwc)

[Hébergement 50](#_heading=h.49x2ik5)

[Optimisation des performances et Référencement 51](#_heading=h.2p2csry)

[Cookies/RGPD 52](#_heading=h.147n2zr)

[Analyse des données 52](#_heading=h.3o7alnk)

[Conclusion 53](#_heading=h.23ckvvd)

[Annexes 54](#_heading=h.ihv636)

[Annexe 1 : Questionnaire de Définition de Projet 54](#_heading=h.32hioqz)

## Avant-Propos

Ce dossier de projet a été réalisé dans le cadre de ma formation afin de devenir **Développeur Web** au sein de l’organisme de formation **ARINFO de Saint Nazaire**. Cette formation a duré du 28/09/2020 au 26/03/2021. Elle a eu lieu en pleine crise du ‘Covid-19’, de confinements en déconfinements, de couvre-feux en télétravail, mais l’organisme de formation et ses différents interlocuteurs (Formateurs, Responsable de centre, etc.) ont toujours cherché des solutions pour nous permettre de continuer notre formation dans les meilleures conditions.

Je suis titulaire d’un diplôme d’Ingénieur en Matériaux, mais pour diverses raisons, je n’ai jamais eu l’occasion de le mettre en pratique. Après une période d’inactivité, j’ai cherché à **changer de voie**, à **réacquérir des compétences** afin de prendre un nouveau départ. L’informatique m’avait toujours attiré, et j’avais régulièrement utilisé cet outil lors de mes études, aussi était-ce l’une des pistes privilégiées pour ma reconversion.

Mais c’est lorsque Gwendal Bescont a produit le site internet d’une association d’escrime médiévale dont je suis le secrétaire, dans le cadre de la même formation que je fais aujourd’hui, que j’ai véritablement pu estimer ce qu’était le métier de développeur Web.

J’ai été séduit par la possibilité de créer qu’offre la programmation, par la **quantité de ressources disponibles librement en ligne**, permettant de toujours découvrir et apprendre, et assouvir la curiosité qui me caractérise. J’ai également été attiré **par la variété des emplois** **et des thèmes** que regroupe le métier de développeur, et par la **possibilité d’évolution** qu’il propose dans une carrière.

Je tiens à remercier **Nicolas Gicquel**, mon formateur, pour sa patience et sa pédagogie. Il a su créer un environnement de travail sympathique et chaleureux. Je remercie aussi **Anaïs Limousin**, formatrice infographistes et designers Web au sein d’ARINFO Saint Nazaire pour tous ses conseils tout au long de ma formation, et **Élodie Boinet**, responsable du centre, pour sa disponibilité et sa bonne humeur, et plus généralement tout le personnel d’ARINFO. Je salue tous mes camarades de promotion, pour leur sympathie, et l’ambiance d’entraide que nous avons pu créer.   
Je tiens également à remercier **Gwendal Bescont**, qui m’a fait découvrir le métier de développeur et a su me conseiller, et **tous mes amis** pour leur soutien durant ces longs mois durant lesquels je n’ai pas été très disponible.  
Je remercie bien sûr **ma mère**, qui, par son soutien, et par son aide, m’a permis d’aller au bout de cette formation et l’a rendu plus aisée.

Enfin, même s’ils ne liront sans doute jamais ces quelques lignes, je remercie les créateurs de tutoriels que sont ‘Nouvelles Technos’, Lior Chamla, Open Classroom, Grafikart, qui m’ont permis d’approfondir mes connaissances ou mieux comprendre certaines notions.

## Liste des compétences couvertes par le titre

1 - Maquetter une application

2 - Réaliser une interface utilisateur web statique et adaptable

3 - Développer une interface utilisateur web dynamique

4 - Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

5 - Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité

6 - Créer une base de données

7 - Développer les composants d’accès aux données

8 - Développer la partie back-end d’une application web ou web mobile

9 - Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

## Résumé du projet

L’Association « **La Compagnie des Jeux** » est une association de joueurs de jeux de société, jeux de cartes, wargames, etc. Elle compte des membres dans tout le bassin nazairien et jusqu’à Nantes, Pontchateau ou Guérande.

Le bureau administratif de l’association ayant émis le souhait de **développer la communication** afin d’attirer de nouveaux membres, étant moi-même membre et m’apprêtant à commencer une formation afin de devenir développeur Web, j’ai proposé lors de l’assemblée générale de moderniser le site internet, qui avait été conçu bénévolement par un membre de l’association en 2016 (fig.1). 

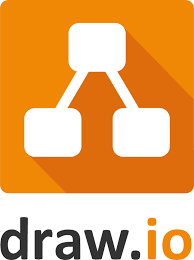
Afin de pouvoir échanger sur les différents aspects du projet, une section dédiée a été créée sur le Discord de l’association, un logiciel de discussion libre.

Suite à l’analyse d’un **questionnaire de définition de projet** (Annexe 1) soumis aux clients, j’ai pu leur proposer un **cahier des charges**, comprenant une veille concurrentielle, fixant les fonctionnalités et les limites du projet, le design général du site, ainsi que des wireframes décrivant le rendu des différents types de pages constituant le site dans son aspect final, pour différentes résolutions d’écran.

## Outils Utilisés

Au cours de son développement, le projet a nécessité l’utilisation de nombreux outils.

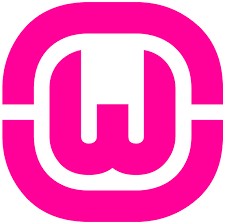
### Maquettage et diagrammes

* L’application google **draw.io**, pour construire les diagrammes d’activité, UML, ainsi que les wireframes du site.



* **Looping**, un logiciel gratuit et libre d’utilisation, qui a été conçu par l’Université Toulouse III, et qui permet notamment de construire des MCD (Modèles Conceptuels de Données) et des MLD (Modèles Logiques de Données), conçus à partir de la méthode Merise.

### Back-End

* Le framework MVC (Modèle Vue Contrôleur) en PHP **Symfony**. 
* La plate-forme de développement web **Wampserver**, qui m’a permis de simuler localement le fonctionnement de mon site internet.
* **MySQL**, un système de gestion de bases de données intégré notamment à Wampserver, avec lequel j’ai conçu et géré mes bases de données.



* Pour la mise en ligne du site, j’ai utilisé **WinSCP**, un client SFTP (SSH File Transfer Protocol).
* **cPanel,** un panneau de configuration permettant de gérer l’hébergement d’un site web.

### Front-End

* Si j’ai utilisé le framework HTML **Bootstrap** lors de la conception de la maquette du site en HTML/CSS, pour le site final j’ai utilisé **Materialize**, lui aussi un framework HTML, par préférence personnelle, mais également car je voulais inclure certaines fonctionnalités qu’il propose.
* Le préprocesseur CSS **Sass**, afin d’organiser et de construire mon CSS plus aisément. 

### Autres

* Tout au long de ce projet, et plus généralement tout au long de mon formation j’ai développé sur l’IDE (Integrated Development Environment ou Environnement de Développement) **Visual Studio Code**, qui permet de personnaliser facilement son espace de travail, et les nombreuses extensions que l’on peut lui adjoindre de rendre plus aisé le travail de codage.
* Afin de mettre en avant les différents réseaux de l’association, j’ai utilisé les solutions d’intégrations que proposent les plateformes **Facebook**, **Youtube** et **Twitch**.
* J’ai utilisé la plateforme d’hébergement **Github**, afin de faire des sauvegardes en cas de mauvaise manipulation ou de problèmes techniques.
* Afin de créer certains designs, de redimensionner certains contenus multimédias, ou de produire les différentes icones, j’ai utilisé le logiciel libre et gratuit de retouche d’images **Gimp**.

## Conception du site

### Questionnaire de définition de projet et Cahier des charges

Le questionnaire (Annexe 1) et les rencontres avec les clients ont mis en évidence plusieurs points importants que le projet final devait respecter, ils ont été reportés dans un cahier des charges, validé par le client, dont voilà le résumé :

* Le site internet devait être **évolutif**, et pouvoir intégrer les différentes **réseaux sociaux** et médias au fur et à mesure que l’association développait sa communication.
* Le contenu serait ajouté et modifié par les **membres du bureau exécutif de l’association**, et la partie back-end se devait donc d’être **accessible** et adaptée à des néophytes en développement et gestion de sites web.
* D’un point de vue général, le site devait être **moderne**, pour atteindre une cible de tous âges, mais plus principalement les jeunes actifs, de toutes catégories socio-professionnelles.
* Une intégration de fonctions de e-commerce ne serait pas nécessaire.
* L’association a émis l’idée d’intégrer un système de QR codes à sa communication pour renvoyer les gens potentiellement intéressés vers son site internet.
* Le **logo** existant allait servir de guide pour concevoir le design, qui se devait d’être **accueillant et épuré**, et rappeler les **activités ludiques du site.**

### Veille concurrentielle

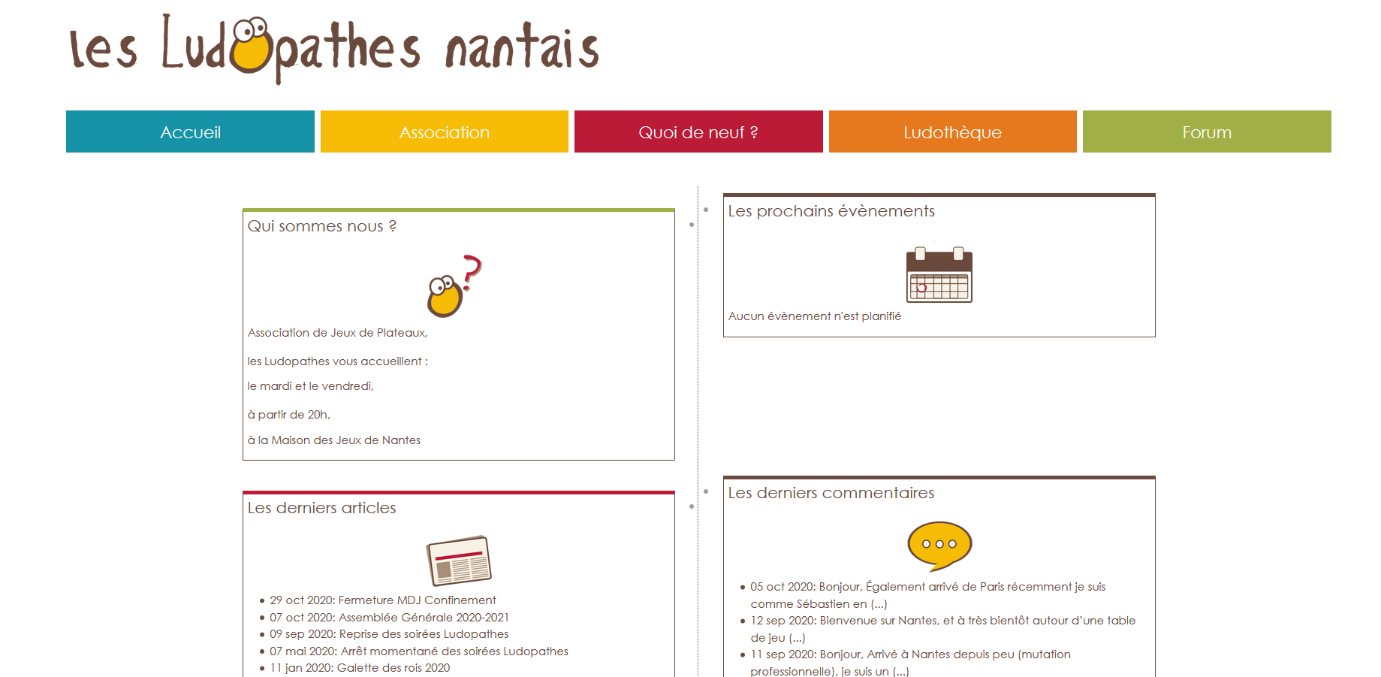
Dans l’idée de proposer à mes clients les solutions les mieux adaptées, de me renseigner sur les concurrents potentiels, ou de m’inspirer des bonnes idées d’acteurs équivalents, j’ai effectué une **veille concurrentielle**.

J’ai ciblé les sites des associations similaires, à proximité géographique mais pas uniquement, mais également les sites d’associations sportives, plus nombreuses, et bénéficiant souvent de moyens plus importants afin de développer leur communication.

Parmi les sites m’ayant inspiré, j’ai sélectionné ceux-ci :

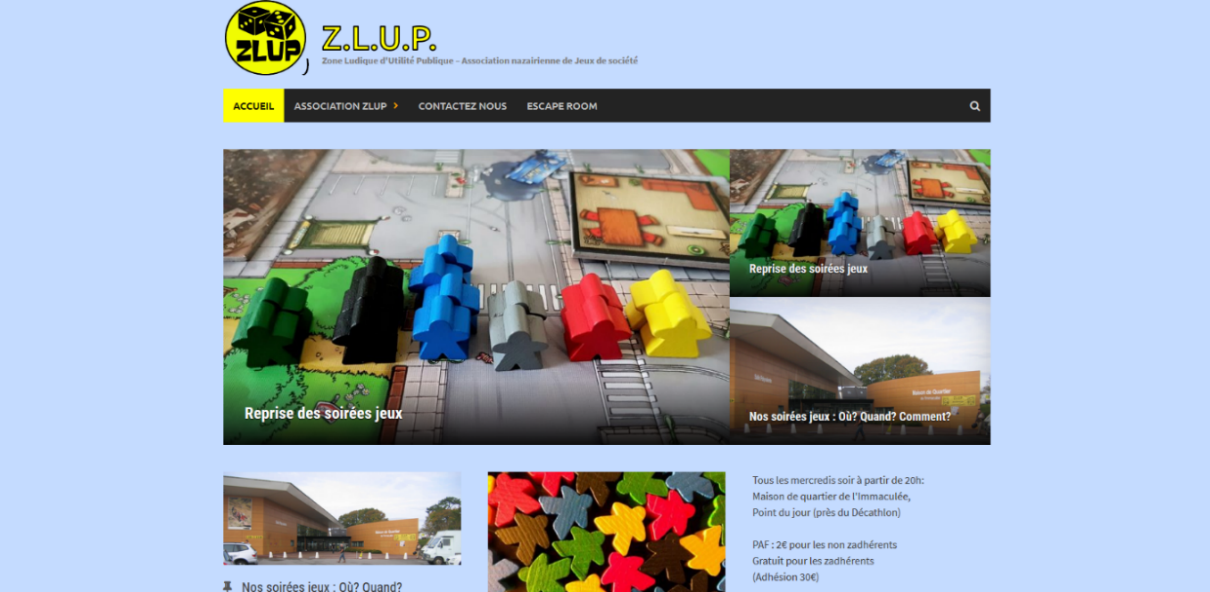
* [Les Ludopathes Nantais : http://www.lesludopathesnantais.fr/](http://www.lesludopathesnantais.fr/)

Notamment pour sa chronologie centrale, mais aussi pour son côté épuré, simple.



* [La Z.L.U.P. : https://www.zlup.fr/](https://www.zlup.fr/)

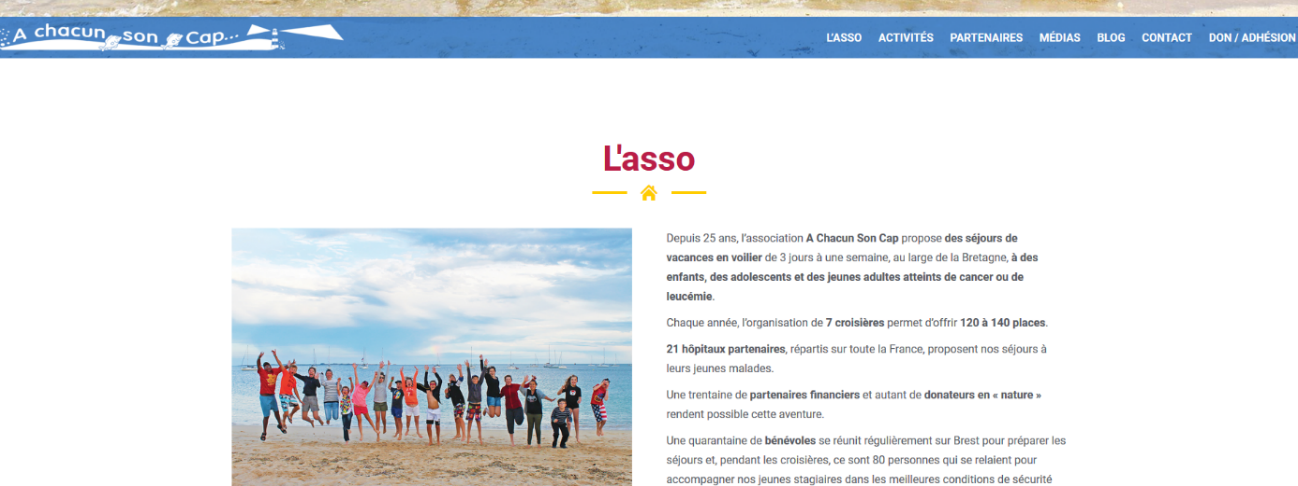
C’est une association assez similaire à celle de La Compagnie des Jeux, en termes d’activité et de localisation. Si le site possède quelques défauts de conception (Double menu principal, très grandes marges latérales, …), il est sobre et ne perd pas ses visiteurs avec trop d’informations inutiles.



***Figure 3 - Site de la ZLUP***

* [A Chacun Son Cap : https://www.achacunsoncap.com/](https://www.achacunsoncap.com/)

Si les activités de cette association diffèrent de celles de mon client, le design simple, élégant et intelligent de ce site m’a paru particulièrement adapté dans le cadre du site vitrine d’une association.



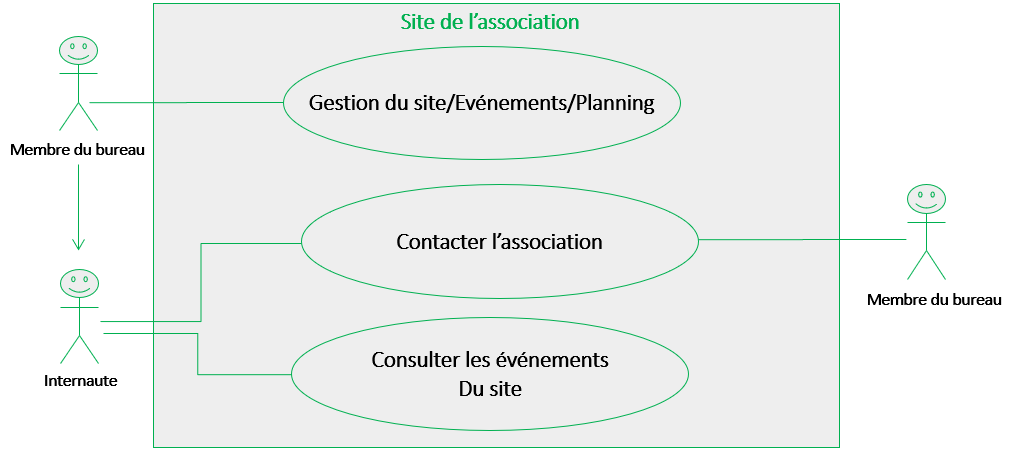
***Figure 4 - Site d'A Chacun Son Cap***

### Diagrammes et Modélisation

Le langage de modélisation UML (Unified Modelling Langage) est un ensemble de diagrammes qui permet de décrire de manière précise le projet que l’on veut développer.

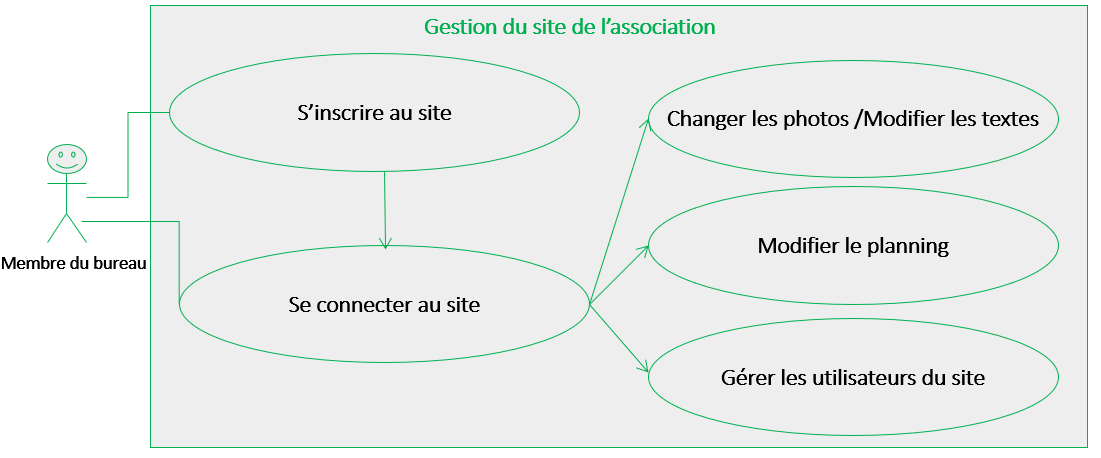
##### Diagrammes de cas d’utilisation

Les diagrammes de cas d’utilisation permettent de représenter le fonctionnement du site.



***Figure 5 - Diagramme de cas d'utilisation pour la partie publique***

A partir du site, un internaute lambda peut contacter l’association, le système envoie alors un message à un administrateur (Membre du bureau), ou consulter les événements.

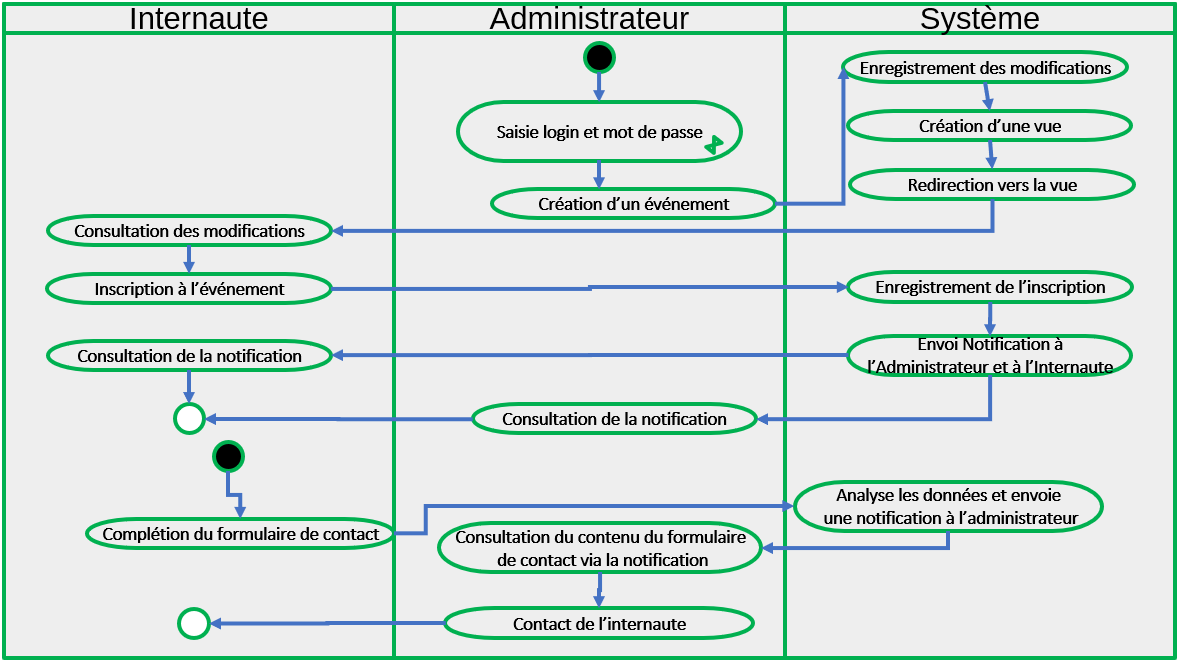


***Figure 6 - Diagramme de cas d'utilisation pour la gestion du site***

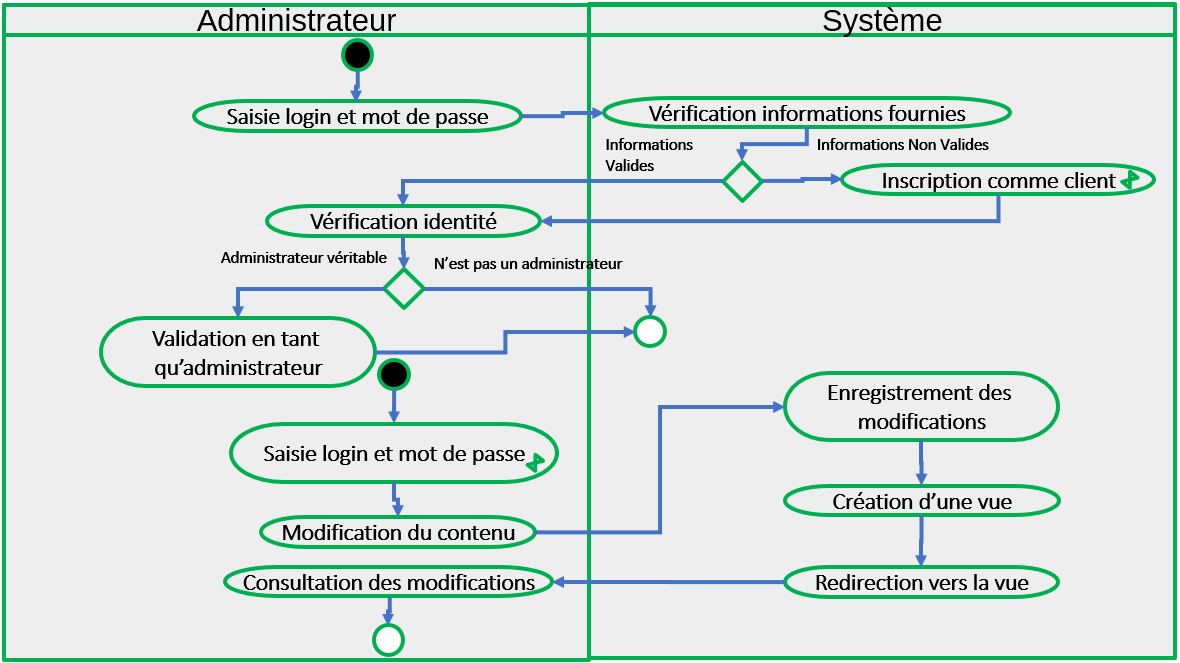
Un membre du bureau, doté du rôle d’administrateur, peut gérer l’affichage du site, le planning, et les utilisateurs.

##### Diagrammes d’activité

Les diagrammes d’activité représentent sous la forme d’un enchaînement d’activités le comportement du système.



***Figure 7 - Diagramme d'activité Internaute/Administrateur/Système***

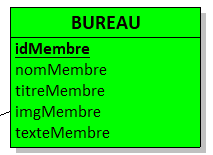


***Figure 8 - Diagramme de cas d'utilisation Administrateur/Système***

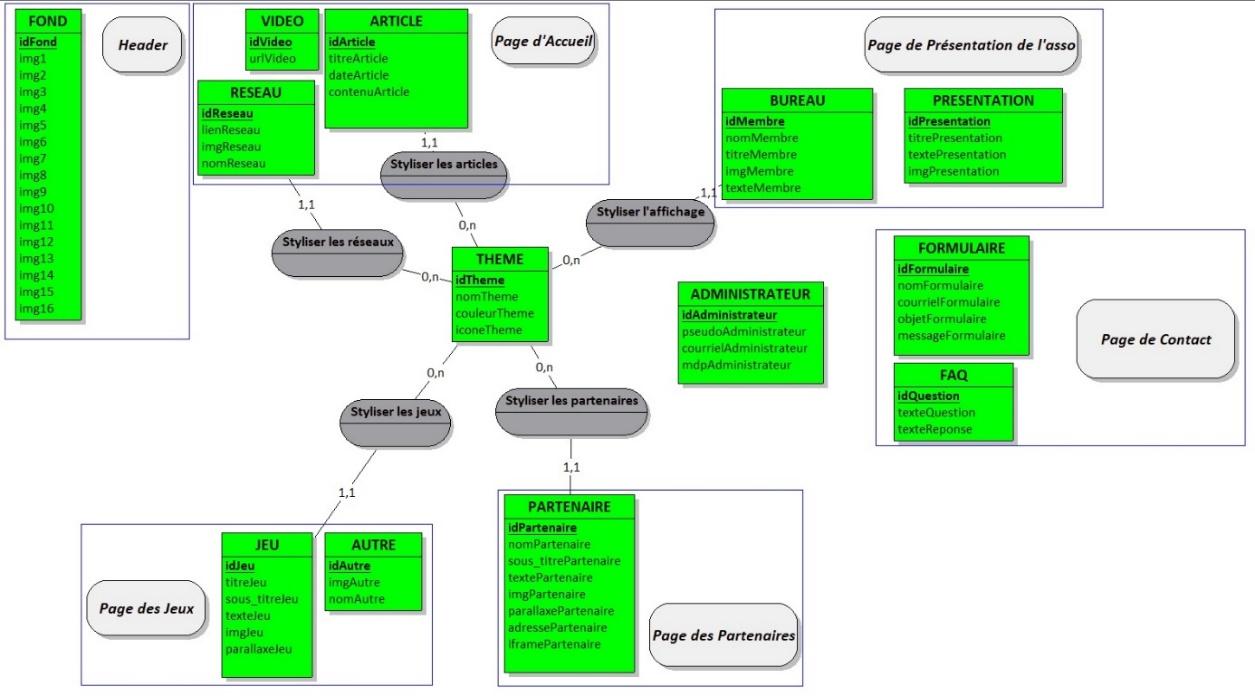
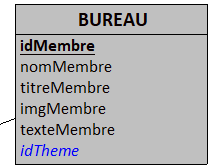
Ce diagramme décrit le système d’inscription et de connexion au site, cette dernière est permise uniquement aux administrateurs. Un nouvel utilisateur ne pourra pas s’enregistrer lui-même, mais devra le faire via un administrateur.

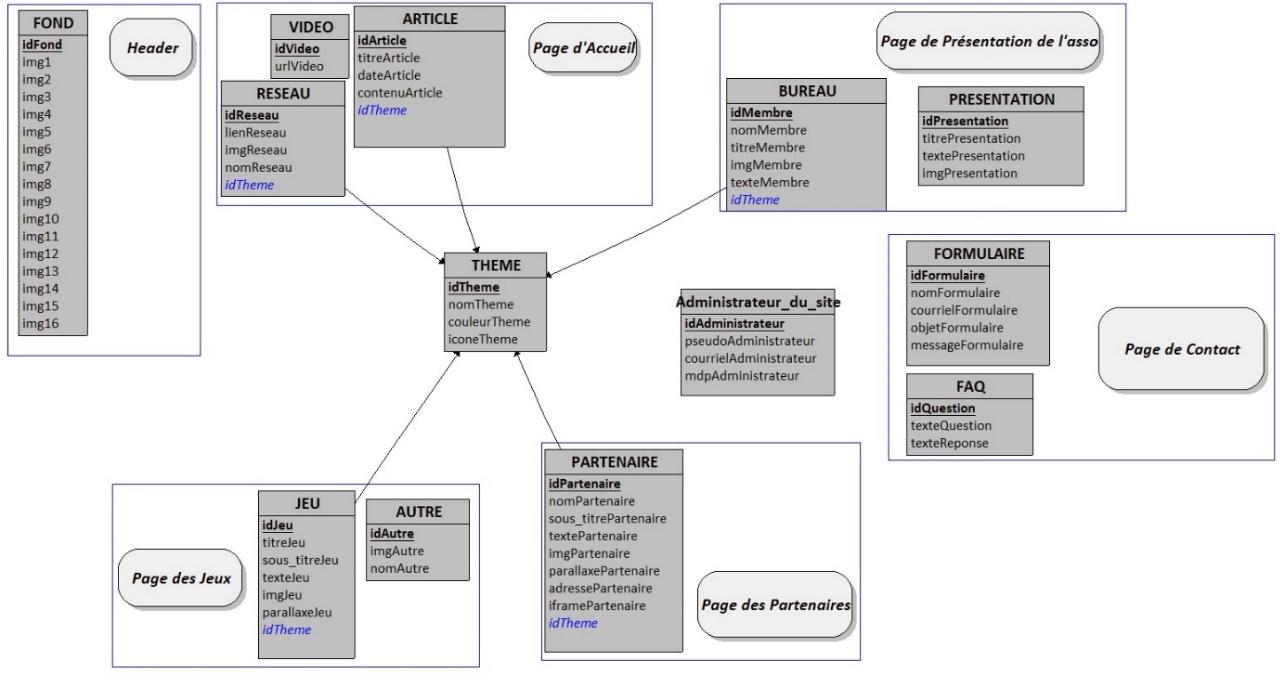
##### MCD et MLD

Afin de construire mes **MCD (Modèles Conceptuels de Données)** et mes **MLD (Modèles Logiques de Données**, j’ai utilisé le logiciel **Looping**. C’est un logiciel gratuit et libre d’utilisation qui a été conçu par l’Université Toulouse III.



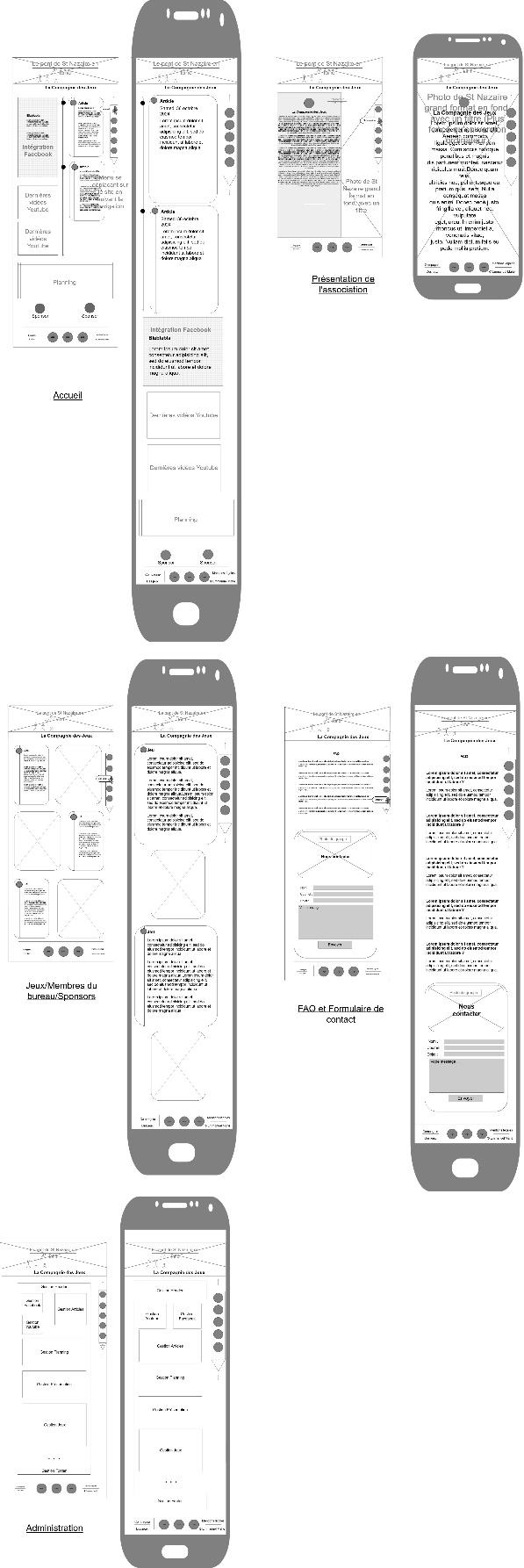
Les **MCD** m’ont permis de construire plus facilement ma **base de données** en la schématisant (fig.10).  
Les différentes ‘**entités**’, en majuscules (fig.9), seront autant de ‘**classes**’ dans Symfony, et les **propriétés** qui les caractérisent autant de ‘**fonctions**’ dans ces ‘classes’.

Une fois le MCD construit, le logiciel génère automatiquement le **MLD** (fig.12). Celui-ci, par rapport au MCD, met en évidence les **relations entre les entités**. Pour le même exemple de l’entité Bureau (fig.11), on peut appliquer un thème différent pour chaque membre du bureau, aussi y retrouve-t-on une propriété ‘idTheme’.



### Maquettage

##### Wireframes



Après avoir réalisé quelques croquis à main levée, je me suis attelé à la création de mes wireframes.   
Pour cela j’ai utilisé l’application de Google **draw.io**, qui a l’avantage d’être simple à prendre en main et permet de stocker ses projets via le drive.

Pour chaque modèle de page, j’ai produit des wireframes à la fois pour la **version écran large** et pour la **version mobile**.

Ce sont ces modèles qui ont été proposés au client dans le cadre du cahier des charges.

Ces wireframes diffèrent évidemment du rendu final, de par leur nature, mais aussi de par l’évolution de mes compétences et de ma vision d’un site internet.   
Cependant, dans la structure générale, et dans l’organisation des différentes fonctionnalités du site, ils restent très similaires.



##### Maquette en HTML/CSS

Dans la volonté de proposer un rendu plus représentatif au client, j’ai réalisé **une maquette en HTML/CSS**.

Elle a été faite en utilisant l’un des frameworks HTML les plus utilisés, **Bootstrap.** Il m’a notamment permis de construire mon front-end plus rapidement et plus efficacement grâce à son système de classes.

Le préprocesseur CSS **Sass** a également été utilisé, afin de rendre le code plus élégant et plus facile à lire grâce à son système d’indentation. 

### Arborescence du site

##### Partie publique du site







##### Partie Administration



### Thème graphique et Design général du site

Par rapport aux souhaits du client, le site se devait d’être ‘**Accueillant, épuré et intéressant**’, mais également de rappeler les **activités ludiques** de l’association.

##### Couleurs

Dans le questionnaire de définition de Projet soumis aux membres du bureau de La Compagnie des Jeux, il est apparu que l’association souhaitait un design simple, fonctionnel et moderne. Le thème graphique du site devait s’axer autour des 3 couleurs du logo de l’association, et ce même logo pouvait servir d’inspiration au design général du site (fig.11).



Dans le code du site, j’ai privilégié le **codage hexadécimal**, qui a l’avantage de se charger légèrement plus rapidement sur les navigateurs.



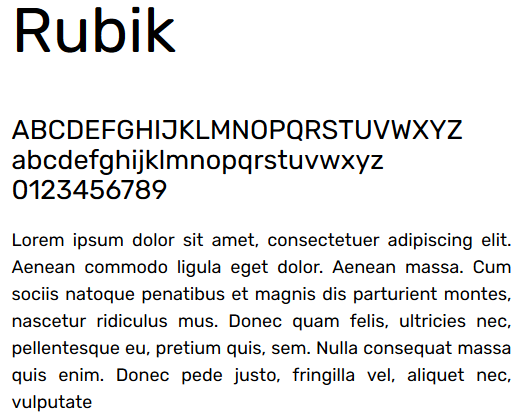
##### Police du site

Pour les différentes polices de caractères employées dans le projet, j’ai utilisé le site **Google Font** qui propose de nombreuses polices de caractères libres d’usage.



Toujours dans une volonté de cohérence, la même police que celle du logo, ‘**Anton**’ (fig.22), a été utilisée pour le titre principal, et s’avère bien adaptée pour un tel usage avec ses lettres larges.





Pour les autres titres, notamment les titres des pages ou des différentes sections, j’ai utilisé la police de caractère ‘**Rubik**’ (fig.23), qui offre de nombreuses possibilités de styles différents, ce qui est avantageux pour une police principale.



En me basant sur les recommandations de Google Font, qui propose pour chaque police de caractère d’autres polices qui la complètent bien, j’ai choisi d’utiliser la police ‘**Poppins**’ (fig.24). Pour une police qui sera présente partout sur le site, elle offre elle aussi de nombreuses options de style.

##### Construction des Cartes

Afin de rappeler l’univers ludique de l’association, et de renforcer l’homogénéité du site, j’ai choisi de présenter une grande partie du contenu sous la forme de **cartes à jouer** (Fig.17).



Une des difficultés était de créer un modèle qui rappelle au premier coup d’œil l’inspiration, mais suffisamment neutre pour qu’il ne soit pas rattaché à un jeu en particulier, tout en proposant un outil pratique, simple et moderne. On peut remarquer sur les exemples que les cartes à jouer ont généralement des bordures épaisses, et leur titre est contenu dans un cadre dans leur partie supérieure.



J’en suis arrivé à ces 2 designs (fig.18), l’un proposant plus de place pour du contenu écrit, l’autre permettant d’illustrer le contenu via une image. Comme visible sur la figure 18, un **thème** peut être appliqué à chaque carte, ce qui va définir la couleur du liseré, ainsi que son icone.

## Back-End

### Étape de création du projet

Sous **Symfony**, il existe une commande qui va créer la structure du site. Elle nécessite **‘Composer’**, un logiciel gestionnaire de dépendances libre, écrit sous PHP. Après l’avoir téléchargé et installé, il est préférable de vérifier si le processus s’est bien déroulé et si PHP est à jour sur l’ordinateur grâce à ces commandes :

composer -v

et

php -v

Si tout est en ordre, on peut alors écrire :

composer create project symfony/website-skeleton CompagnieDesJeux

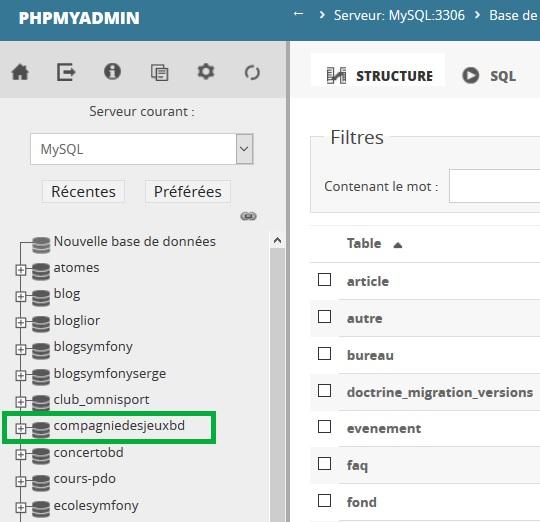
Il est également possible d’omettre le website-, pour une structure plus épurée, par exemple pour un projet d’API ou d’application de type console, dans le cadre d’un site internet la formulation complète est plus adaptée.

Comme Symfony est en POO (Programmation Orientée Objet), il faut créer un ‘controller’ qui va permettre de retourner une réponse à une route ou une requête donnée. On rentre alors dans le terminal :

php bin/console make:controller

### Création de la Base De Données

Une fois la structure générale du site construite, je me suis intéressé à la création de la base de données.



Je suis donc allé modifier la ligne 31 dans le fichier nommé sobrement ‘.env’ à la racine du projet, afin de lier celui-ci avec ma base de données sous MySQL.

DATABASE\_URL=mysql://root:''@127.0.0.1:3306/CompagnieDesJeuxBD

On crée ensuite notre base de données avec la commande :

php bin/console doctrine:database:create

On la retrouve alors dans notre gestionnaire de bases de données (fig.27).

### Entités



***Figure 28 - MCD***

Le site a été pensé pour être géré par des néophytes, aussi beaucoup d’aspects devaient pouvoir être gérés sans avoir recours au code.

##### Création des entités

Pour créer une entité avec Symfony, il suffit d’écrire dans le terminal la commande :

php bin/console make:entity

Un questionnaire nous permettra alors de définir ses propriétés, et les caractéristiques de celles-ci.

Exemple détaillé pour l’entité Article :

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

created: src/Entity/Article

created: src/Repository/ArticleRepository.php

Entity generated! Now let's add some fields!

You can always add more fields later manually or by re-running this command.

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

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

>

Field length [255]:   
>

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

>

updated: src/Entity/Article.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> dateArticle

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

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

>

updated: src/Entity/Article.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> contenuArticle

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

> text

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

>

updated: src/Entity/Article.php

Add another property? Enter the property name (or press <return> to stop adding fields):

>

Success!

Next: When you're ready, create a migration with make:migration

Comme vous pouvez le constater avec la dernière phrase, Symfony nous explique ensuite la marche à suivre, à savoir, la commande

php bin/console make:migration

puis, une fois cela fait, la commande

php bin/console doctrine:migrations:migration

qui va nous permettre de mettre à jour notre base de données.

Ainsi pour chaque entité, nous définissons ses propriétés, puis nous synchronisons avec la base de données.

##### Manipulation des données – La fonction CRUD

Afin de gagner du temps, j’ai utilisé une fonction de Symfony qui se nomme CRUD (Created Read Update Delete). Celle-ci permet de générer automatiquement une structure spécifique à chaque entité, et notamment les formulaires permettant de créer, de modifier une entité ou la fonction de la supprimer.

La commande est assez simple :

Php bin/console make:crud

Évidemment la gestion de certaines entités ne nécessite pas une telle fonction, mais il est souvent plus facile de supprimer les pages inutiles créées automatiquement que de générer manuellement les pages souhaitées.

##### Cas de la propriété dateArticle :

Afin que la date des articles soit enregistrée automatiquement lors de la création d’un article, j’ai suivi un tutoriel du formateur indépendant Benoit de nouvelle-techno.fr intitulé *‘Automatiser les slugs et les dates avec Symfony 5.1’.*

J’ai commencé par installer les extensions Doctrine via la commande :

composer require antishov/doctrine-extensions-bundle

Puis, on modifie la fichier ‘config/packages/stof\_doctrine\_extensions.yaml’ ainsi créé comme suit :

*# See the official DoctrineExtensions documentation for more details: https://github.com/Atlantic18/DoctrineExtensions/tree/master/doc/*

stof\_doctrine\_extensions:

    default\_locale: fr\_FR

    orm:

        default:

            sluggable: true

            timestampable: true

On ajoute ensuite le ‘use’ suivant dans notre entité (src\Entity\Article.php) :

use Gedmo\Mapping\Annotation as Gedmo;

On modifie enfin les annotations de la propriété correspondante et ses ‘setters’ éventuels.

*/\*\**

*\* @var \DateTime $dateArticle*

*\**

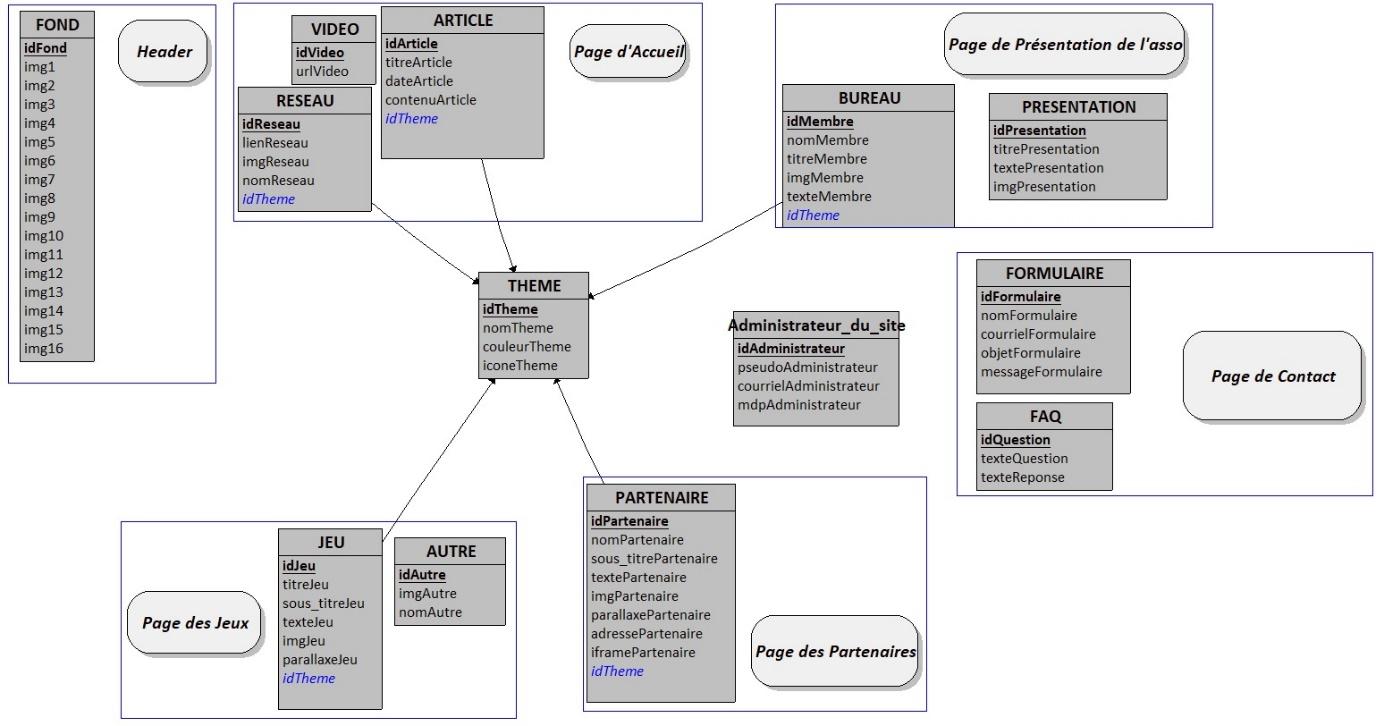
*\* @Gedmo\Timestampable(on="create")*

*\* @ORM\Column(type="datetime")*

*\*/*

    private $dateArticle;

### Relations entre les entités



***Figure 29 - MLD***

Après la création de mon entité ‘Thème’ et de ses outils de manipulation (via la fonction CRUD), j’ai entrepris de la relier aux entités dont elle influencerait le style.

Pour cela je refais un make:entity. Par exemple si je veux relier Article à Theme.

Class name of the entity to create or update (e.g. GrumpyPuppy):

> Article

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

Symfony repère que cette entité existe déjà et je choisis que cette propriété est de type « relation ».

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

> Theme

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

> relation

What class should this entity be related to?:

> Theme

Je vais définir que ces entités sont reliées par une relation ‘Many To One’, en effet chaque Article n’aura qu’un seul thème graphique, alors qu’un Theme pourra être utilisé pour plusieurs articles.

Ce processus a été reproduit pour mes différentes relations.

### Sécurité

##### Entité Administrateur

Pour créer l’administrateur de mon site, j’ai tout d’abord effectué un ‘make:user’ :

php bin/console make:user

Puis, afin que chaque nouvel utilisateur inscrit ait les droits administrateurs, j’ai modifié le fichier src\Entity\User.php  et rajouté une constante :

class User implements UserInterface

{

    public const ROLE\_USER = 'ROLE\_USER';

*/\*\**

*\* @ORM\Id*

*\* @ORM\GeneratedValue*

*\* @ORM\Column(type="integer")*

*\*/*

    private $id;

. . .

J’ai ensuite mis en place un formulaire d’authentification grâce à la commande

php bin/console make:auth

puis mis à jour la base de données avec une migration.

Je complète enfin mon système d’utilisateur en créant un formulaire d’enregistrement.

php bin/console make:registration-form

Afin que seuls les utilisateurs inscrits puissent avoir accès aux pages d’administration et aux différents formulaires de gestion des entités, je vais restreindre l’accès aux méthodes new, edit et delete en ajoutant l’annotation @isGranted dans mes controllers.

Par exemple pour la méthode ‘new’ dans l’entité Article :

. . .

*/\*\**

*\* @Route("/new", name="article\_new", methods={"GET","POST"})*

*\* @isGranted("ROLE\_USER")*

*\*/*

    public function new(Request $request, FondRepository $fondRepository): Response

    {

. . .

##### Sécurisation et tests

Pour effectuer ma veille de sécurité, j’ai consulté le site de l’[OWASP](https://owasp.org/) (Open Web Application Security Project), une communauté en ligne travaillant sur la sécurité des applications Web. Il publie régulièrement des recommandations de sécurité et explique aux internautes comment sécuriser leurs projets.  
J’ai également consulté le blog de [Vaadata](https://www.vaadata.com/fr/), une société spécialisée dans le ‘pentest’ ( = test d’intrusion). J’ai donc pris quelques mesures afin de rendre mon site plus sûr.

* **Tests unitaires**

Pour effectuer des tests unitaires j’ai installé le framework opensource PHPUnit.  
Pour cela on utilise Composer et on rentre les lignes de commande suivantes :

composer require --dev symfony/phpunit-bridge

Et

composer require --dev symfony/browser-kit symfony/css-selector



Pour tester les controlleurs et vérifier que toutes les routes que l’on a définies mènent bien à une page, je crée un dossier ‘Controllers’ dans le dossier ‘Test’ (présent dans l’architecture de base d’un projet Symfony), dans lequel j’ajoute un nouveau fichier pour chacune de mes entités (fig.34).



Exemple pour l’entité ‘Theme’ :

<?php

*//tests/Controller/ThemeControllerTest.php*

namespace App\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class ThemeControllerTest extends WebTestCase

{

    public function testNewTheme()

    {

        $client=static::createClient();

        $client->request('GET','/theme/new');

**$this**->assertEquals(200, $client->getResponse()->getStatusCode());

    }

}

Ici nous avons un test pour vérifier si la page d’ajout d’un nouveau thème existe et fonctionne correctement.

Je cherche également à tester mes formulaires. Pour cela, dans la même logique que pour les controlleurs, je crée dans ‘Tests’ un dossier ‘form’, et j’ajoute un nouveau fichier pour chaque formulaire que je souhaite tester. Par exemple pour l’entité ‘Formulaire’ (mon formulaire de contact) :

class FormulaireFormTest extends KernelTestCase{

    public function testNewFormulaire(){

        $formulaire=(new Formulaire())

        ->setNomFormulaire('1')

        ->setCourrielFormulaire('1')

        ->setObjetFormulaire('1')

        ->setMessageFormulaire('1');

        self::bootKernel();

        $error = self::$container->get('validator')->validate($formulaire);

**$this**->assertCount(0,$error);

    }

}

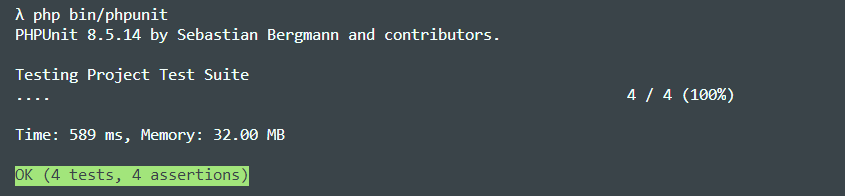
Vous pouvez remarquer que les données rentrées ne devraient pas être valides. C’est ce que je cherche à éviter. J’aurais pu également tester en laissant les champs vides, ou en rentrant un très grand nombre de caractères, etc.

Nous allons effectuer nos tests (controlleurs et formulaires) en rentrant la ligne de commande :

php bin/phpunit

Comme j’ai déjà mis en place les contraintes de validation dans mon projet (*voir section suivante*), les tests me renvoient des erreurs.

Si les controlleurs et les formulaires testés n’avaient pas été sécurisés, le test aurait ressemblé à cela :



***Figure 31 - Exemple de test "sans erreurs" avec PHPUnit***

* **Contraintes de validation**

Afin de me prémunir des injections, j’ai suivi les recommandations de la documentation de Symfony et j’ai appliqué des contraintes de validation sur mon formulaire de contact.  
Exemple :

*/\*\**

*\* @ORM\Column(type="text")*

*\* @Assert\Email(*

*\*     message = "Le Courriel '{{ value }}' n'est pas un courriel valide."*

*\* )*

*\*/*

    private $courrielFormulaire;

Ici, on vérifie que le courriel entré est valide et affiche à l’utilisateur un message si ce n’est pas le cas.

*/\*\**

*\* @ORM\Column(type="string", length=255)*

*\* @Assert\NotBlank(*

*\*      message = "Ce champs ne peut être vide"*

*\* )*

*\* @Assert\Length(*

*\*      min = 2,*

*\*      max = 100,*

*\*      minMessage = "Ce champs doit comporter au moins {{ limit }} caractères",*

*\*      maxMessage = "Ce champs doit comporter au maximum {{ limit }} caractères"*

*\* )*

*\*/*

    private $objetFormulaire;

Là on vérifie si le champs « objet » du message n’est pas vide, et si le nombre de caractère n’est pas trop faible (inférieur à 2) ou trop important (supérieur à 100), et un message correspondant est affiché.

* **Mise en place du « https » et sécurité en production**

Le **https** ( = protocole de transfert hypertextuel sécurisé) est variante du protocole **http** (HyperText Transfer Protocol) chiffrée grâce au protocole **TLS** (Transport Layer Security). Il garantit la confidentialité et l'intégrité des données envoyées par et reçues du serveur.  
Pour des raisons de **sécurité**, mais également pour optimiser **le référencement d’un site** (*Exemple : Dans ses résultats de recherche, Google favorise les sites en https*), il devient indispensable de mettre en place le https.

Pour cela, une fois le site **mis en production** à partir du panneau de contrôle **cPanel**, j’ai utilisé **Let’s Encrypt** qui permet de délivrer une certification de validation de domaine.

J’ai ensuite effectué des tests de sécurité grâce au site **SSL Labs** (fig.27), qui a donné à mon site un score tout à fait satisfaisant.



* **Installation du reCAPTCHA**

Le **reCAPTCHA** protège les sites internet contre les fraudes et les abus. Il bloque les faux utilisateurs, et analyse les risques pour empêcher d’éventuels logiciels malveillant de se livrer à des activités abusives.

Afin de l’installer, je me suis inscrit sur le site <https://www.google.com/recaptcha/about/> et j’ai ensuite intégré le reCAPTCHA sur mon site via le code :

. . .

*<!--reCAPTCHA Google (dans mon ‘head’)-->*

    <script *src*="https://www.google.com/recaptcha/api.js?render=MaCléreCAPTCHA"></script>

. . .

. . .

<script>

    function onClick(*e*) {

*e*.preventDefault();

        grecaptcha.ready(function () {

            grecaptcha.execute(' MaCléreCAPTCHA', { action: 'submit' }).then(function (*token*) {

                    console.log('captacha');

            });

       });

   }

</script>

. . .



Un symbole bleu apparaît alors en bas à droite de la page Web (fig.33), signalant que le reCAPTCHA est actif.

*Remarque : Sur la figure 33, il est effectivement partiellement caché par le symbole de ‘Tarte au citron’, le gestionnaire de cookies mis en place sur le site.*

## Front-End

### Materialize

Afin de développer plus rapidement la partie Front-End de mon site, ainsi que de profiter des fonctionnalités qu’il propose, j’ai choisi d’utiliser le framework HTML **Materialize**.

Symfony est plutôt conçu pour utiliser un framework bien connu, **Bootstrap**. Aussi si son intégration aurait été plus aisée dans ce projet, c’est pour des raisons esthétiques, et aussi pour m’exercer sur un autre framework que j’ai effectué ce choix.

##### Mise en place

Pour mettre en place ce framework, j’ai choisi l’intégration par CDN (Content Delivery Network), j’ai donc dû ajouter dans le ‘head’ de ma page de base cette ligne :

*<!-- Compiled and minified CSS -->*

<link *rel*="stylesheet" *href*="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/css/materialize.min.css">

Et j’ai ajouté cette ligne juste avant la fermeture de la balise ‘body’ :

*<!-- Compiled and minified JavaScript -->*

<script *src*="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/js/materialize.min.js"></script>

##### Système de grille

De manière similaire à de nombreux frameworks, **Materialize** utilise un système de **grilles**, basé sur la séparation de l’écran en 12 colonnes. Ce système s’applique grâce à l’ajout de ‘classes’ (fig.26) sur les différents éléments du code HTML.

| **Points de rupture** | **<= 600 px** | **> 600 px** | **> 992 px** | **> 1200 px** |
| --- | --- | --- | --- | --- |
| **Préfixe de la classe** | .s | .m | .l | .xl |
| **Largeur du conteneur** | 90% | 85% | 70% | 70% |

***Figure 34 - Adaptabilité du site en fonction de la largeur de l'écran***

Outre la possibilité d’organiser et de placer les éléments sur la page, les grilles facilitent l’adaptation du site aux différents formats d’écrans

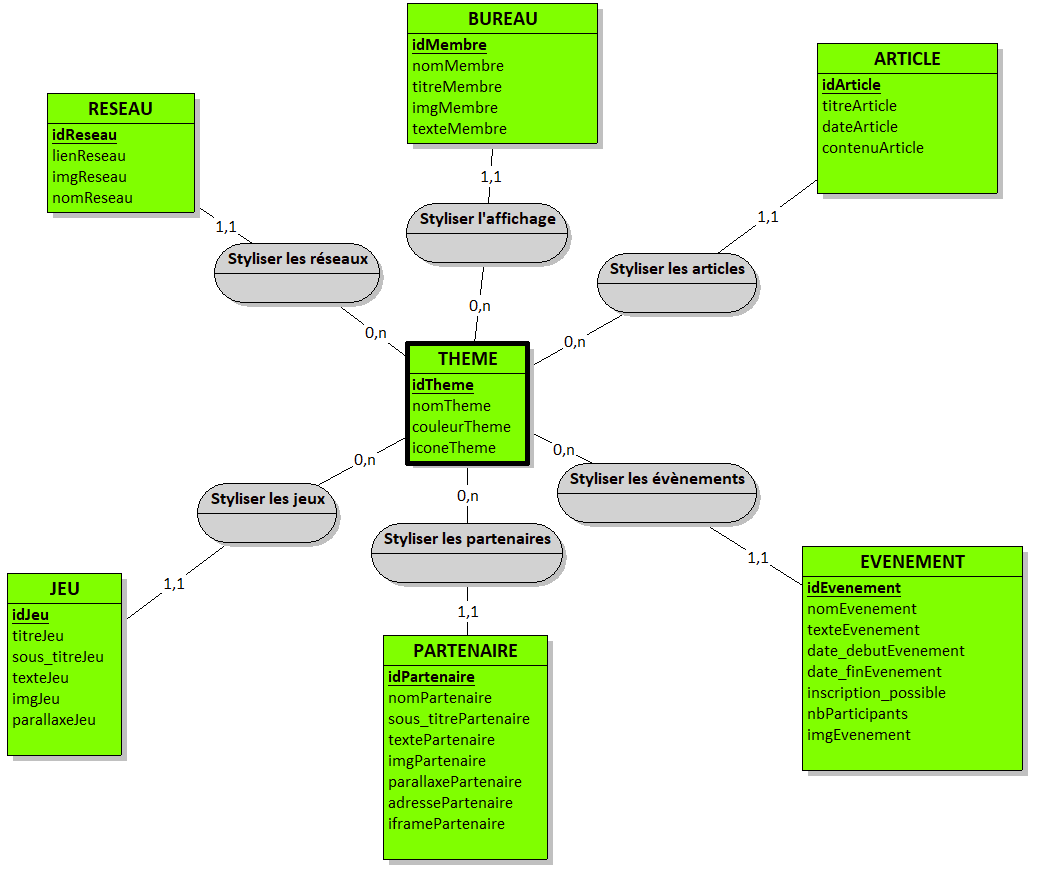
Exemple d’utilisation du système de grille :

### Modèle de cartes

##### Thème

Si l’affichage des cartes (cf*. Thème graphique et Design général du site/Construction des cartes*) allaient dépendre de leur contenu, un **thème dépendant du sujet** influence aussi celui-ci.

Pour cela, j’ai créé une entité ‘Theme’ (fig.27), à laquelle on peut adjoindre un nom, une couleur, et une image.



Une section dédiée dans la partie administration a été ajoutée afin de pouvoir ajouter de nouveaux thèmes, et modifier ceux existants (Figure 28). 

##### Exemple de code

Ceci est un exemple de la structure HTML type d’une « carte », ici pour afficher les articles de page d’accueil.

<div *class*="conteneur">

    {% for article in articles %}

    <div *class*="actu">

        <div *class*="point"></div>

        <div *class*="carte">

*<!-- Vous remarquerez ici l'utilisation de la classe "z-depth-" de Materialize*

*qui permet d'ajouter des ombres préconçues sur nos éléments -->*

*<!-- La couleur du thème défini, "couleurTheme", est amenée via l'utilisation du langage Twig -->*

            <div *class*="ligneArticle z-depth-1"

*style*="background-color: {{ article.theme.couleurTheme }}">

*<!-- De même en ce qui est de l'icone du thème, "iconeTheme" -->*

                <img *class*="z-depth-1"

*src*="{{ asset('uploads/photos/' ~ article.theme.iconeTheme) }}"

*style*="background-color: {{ article.theme.couleurTheme }}">

                    <h4>{{ article.titreArticle }}</h4>

            </div>

            <h5 *class*="date">

                Article du {{ article.dateArticle ? article.dateArticle|date('d/m/Y') : '' }}

            </h5>

            <div *class*="paragrapheArt z-depth-1">{{ article.contenuArticle|raw }}</div>

            <div *class*="ligne z-depth-1"

*style*="background-color: {{ article.theme.couleurTheme }}">

            </div>

        </div>

    </div>

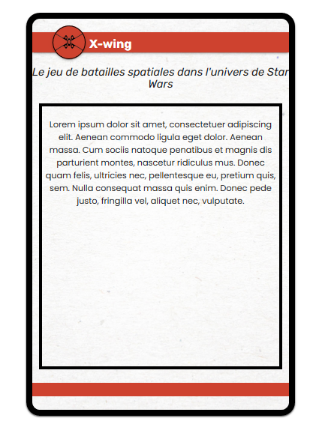
    {% endfor %}

</div>

##### Exemple d’application des thèmes

* **Thème X-Wing**





X-wing est un jeu de figurines basé sur l’univers de Star Wars et simulant des combats spatiaux. En tant que jeu phare de l’association, il est très présent dans son actualité.

La figure illustre un exemple d’une carte de présentation de jeu, telle qu’ils sont présentés sur la page ‘**JEUX**’.

* **Thème Multilud**

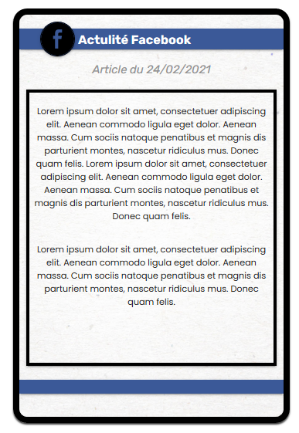


Multilud est un magasin de jeux de cartes, de jeux de plateaux et de jeux de figurines, et de tout ce qui en découle (accessoires, modélisme, peinture, univers, etc.) et un partenaire historique de l’association.

L’exemple correspond ici à la présentation d’un des partenaires comme dans la page ‘**NOS PARTENAIRES’**.

* **Thème Facebook**

****



La Compagnie des Jeux est présente sur les réseaux sociaux et souhaite développer sa communication sur ses médias. Facebook est celui où ses membres sont les plus actifs.

La figure correspond à la forme que prendrons les articles sur la page **‘ACCUEIL’** du site.

### Header

**L’En-tête** du site, en plus de reprendre les couleurs du logo, est une façon ludique et modifiable de rappeler les différentes activités de l’association et son dynamisme via un **ensemble des tuiles (fig.29).**

Les différentes illustrations peuvent être aisément remplacées dans la partie Administration du site, et **un effet au survol** leur est adjoint afin de leur ajouter du volume, comme si elles étaient un ensemble d’affiches ou les pages d’une revue illustrée.

### Menus de navigation

Afin de faciliter la navigation sur les différentes pages, sans que les menus correspondants n’accaparent trop de place sur l’écran, j’ai cherché à mettre en place deux modèles de menus différents pour la version ordinateur du site.

Ces deux modèles différents ont chacun deux versions différentes, l’une pour la partie publique du site, et l’une pour la partie administration.

Pour chaque menu, la page active est mise en évidence par des couleurs différentes, ainsi que la partie du menu que la souris survole. 

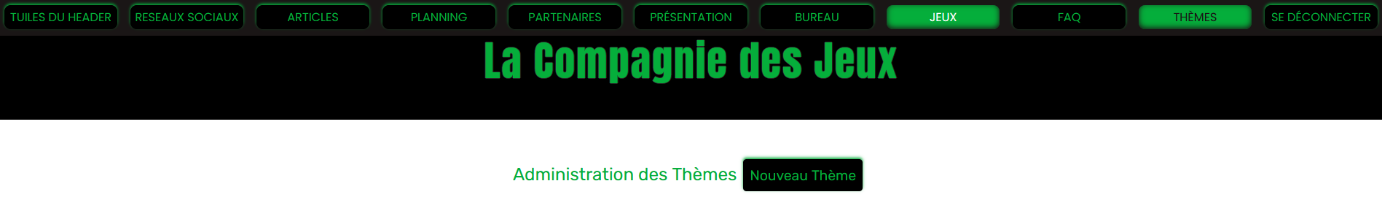
##### Menu Principal

Le menu principal (fig.31) se situe au niveau de l’en-tête et couvre toute la largeur de l’écran. Il a été pensé pour être lisible le plus facilement possible, en listant les différentes pages du site et à quoi elles renvoient.



***Figure 39 - Menu Principal pour la partie publique***

Si le menu principal de la partie administration est légèrement différent en termes de couleurs (fig.32), il est similaire en design et en objectifs.



***Figure 40 - Menu Principal pour la partie Administration***

##### Menu Latéral

Lorsque l’usager fait défiler la page, le menu principal va disparaitre, et un menu latéral va apparaitre (fig.33).

Afin d’obtenir ce résultat, j’ai utilisé le script JavaScript suivant :

*//Navbar*

  var hauteur = 300; *// Nombre de pixels de scroll où il est sensé apparaitre*

  var hauteur2 = 450;

  $(function () {

    $(window).scroll(function () {

*//Au scroll dans la fenetre on déclenche la fonction*

      if ($(**this**).scrollTop() > hauteur) {

*//si on a défile de plus de XXX (variable "hauteur) pixels du haut vers le bas*

        $('.preNav').animate().removeClass(" cache");

      } else {

        $('.preNav').animate().addClass(" cache");

      }

    });

  });

  $(function () {

    $(window).scroll(function () {

*//Au scroll dans la fenetre on déclenche la fonction*

      if ($(**this**).scrollTop() < hauteur) {

*//si on a défile de plus de XXX (variable "hauteur) pixels du haut vers le bas*

        $('.navHeader').fadeIn(500);

      } else {

        $('.navHeader').fadeOut(500);

      }

    });

  });

  $(function () {

    $(window).scroll(function () {

*//Au scroll dans la fenetre on déclenche la fonction*

      if ($(**this**).scrollTop() < hauteur2) {

*//si on a défile de plus de XXX (variable "hauteur2) pixels du haut vers le bas*

        $('.navHeaderBase').fadeIn(500);

        $('.preNavBase').animate().addClass(" cache");

      } else {

        $('.navHeaderBase').fadeOut(500);

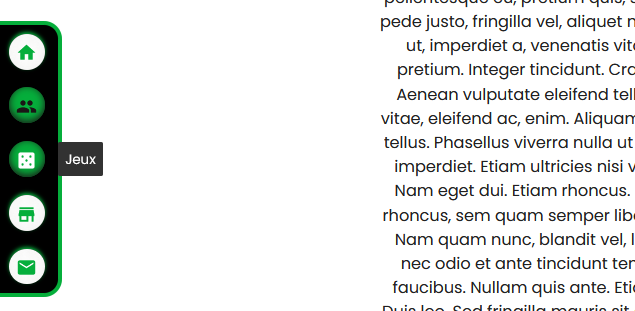
        $('.preNavBase').animate().removeClass(" cache");

      }

    });

  });

Ce menu latéral est pensé pour être plus léger, pour ne pas empiéter sur la lisibilité. Son orientation est verticale, afin d’être le plus fin possible par rapport à la largeur de l’écran.





Pour ce menu latéral, l’intitulé de chaque page est remplacé par une icône représentative. Mais l’intitulé de la page apparait au survol du bouton.

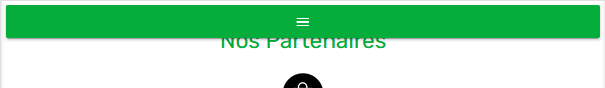
Le menu latéral de la partie administration reprend les couleurs du menu principal (fig.34).



##### Menu Version Petits Écrans

Dans le cas d’écrans à la résolution inférieure, les deux modèles de menus disparaissent, et un menu spécialement conçu apparait (fig.35).

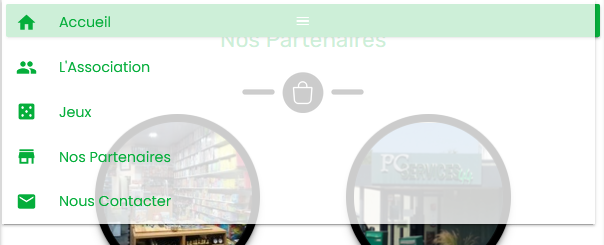
Il est basé sur la fonction « dropdown » du framework Materialize que j’ai modifié afin qu’il n’apparaisse que pour les écrans d’une résolution inférieure à 809px, et qu’il occupe toute la largeur de l’écran.



***Figure 43 - Menu Petits Écrans***



***Figure 44 - Menu Petits Écrans au survol***



***Figure 45 - Menu Petits Écrans déployé***

### Création d’un Calendrier

Dans l’optique de renseigner les visiteurs sur les séances hebdomadaires lors desquelles les membres se réunissent, ou pour annoncer les différentes manifestations ou compétitions auxquelles l’association participe ou qu’elle organise, j’ai mis en place un calendrier qui apparait sur la page d’accueil du site.

##### Cas de la gestion via la Base de Données

Dans un premier temps, ma volonté était de gérer toutes les informations de ce calendrier à partir de la base de données. C’est pourquoi j’ai créé différentes entités (fig.30) :

* Une entité « **Séance** », qui regroupait toutes les activités régulières de l’association.
* Une entité « **Événement** », pour les activités exceptionnelles ou ponctuelles.
* Une entité « **Participant** », pour permettre aux visiteurs de s’inscrire aux événements ou au compétitions organisées par la Compagnie Des Jeux.

Pour mettre en place ce système dans le site, j’ai choisi d’utiliser **FullCalendar, une bibliothèque JavaScript** populaire, utilisée par des particuliers ainsi que par de grandes entreprises et organismes (Amazon, Netflix, Paypal, la NASA, etc.).

Exemple du code que j’ai intégré :

*<!-- FullCalendar CSS-->*

<link *rel*="stylesheet" *href*="https://cdn.jsdelivr.net/npm/fullcalendar@5.5.1/main.min.css">

*<!--- FullCalendar JS-->*

    <script *src*="https://cdn.jsdelivr.net/npm/fullcalendar@5.5.1/main.min.js"></script>

    <script *src*="https://cdn.jsdelivr.net/npm/fullcalendar@5.5.1/locales-all.min.js"></script>

    <script>

        document.addEventListener('DOMContentLoaded', function() {

            var calendarEl = document.getElementById('calendar');

            var calendar = new FullCalendar.Calendar(calendarEl, {

*//plugins: [ 'dayGrid','timeGrid','list' ],*

                initialView: 'dayGridMonth',

                locale: 'fr',

                textColor: 'black',

                header: {

                    left: 'prev,next today',

                    center: 'title',

                    right: 'dayGridMonth,dayGridWeek,listMonth'

                },

                events: [{

                    title:"Session du Mercredi",

                    start: '18:00', *// commence à 18h*

                    end: '22:00', *// fini à 22h*

                    dow: [ 3 ], *// Tous les mercredis*

                    display: 'background'

                },

                {

                    title:"Session du Samedi",

                    start: '14:00', *// commence à 14h*

                    end: '22:00', *// fini à 22h*

                    dow: [ 6 ], *// Tous les samedis*

                    display: 'background'

                },

                {

                    title:"Weekend",

                    start:"2021-02-05 17:00",

                    end:"2021-02-08 09:00"

                }

                ],

            });

**Si FullCalendar est un outil puissant, doté de nombreuses fonctionnalités**, en tant qu’utilisateur de la version gratuite j’étais limité dans mes possibilités. D’autre part, l’utiliser en conjonction avec ma base de données s’avéraient une opération certes faisable à terme, mais **complexe sans un investissement en temps et en argent.**

##### Google Calendar

Intégrer une solution aussi puissante n’était pas une demande de mes clients, et au fur et à mesure de mes recherches il m’est apparu que **Google Calendar** pouvait satisfaire à mes besoins plus simplement, et même s’avérer plus pratique sur certains aspects, notamment par le fait de pouvoir le modifier à tout moment via son compte Google.

Aussi **la figure 30** n’est-elle présente que dans un souci d’illustration et n’est plus valable.

L’intégration de Google Calendar se fait via une iframe.

<div *class*="calendrier">

    <h2>Calendrier</h2>

    <div *class*="separation">

        <img *src*="{{ asset('ressources/images/barre.png') }}"><img

*src*="{{ asset('ressources/images/iconePlanning.png') }}"><img

*src*="{{ asset('ressources/images/barre.png') }}">

    </div>

*{# Calendrier pour les écrans larges, avec un affichage par mois #}*

    <iframe *class*="calendrierXL" *src*="https://calendar.google.com/calendar/embed?height=800&amp;wkst=1&amp;bgcolor=%2305ae3b&amp;ctz=Europe%2FParis&amp;src=Y2ZqMGJob2diZWRoMmFiZXZzMG9jNDkxNzRAZ3JvdXAuY2FsZW5kYXIuZ29vZ2xlLmNvbQ&amp;color=%23795548&amp;mode=MONTH&amp;title=La%20Compagnie%20des%20Jeux" *style*="border-width:0" *width*="1400" *height*="800" *frameborder*="0" *scrolling*="no"></iframe>

*{# Calendrier pour une résolution moyenne de l'écran avec un affichage par semaine #}*

    <iframe *class*="calendrierM" *src*="https://calendar.google.com/calendar/embed?height=400&amp;wkst=1&amp;bgcolor=%2305ae3b&amp;ctz=Europe%2FParis&amp;src=Y2ZqMGJob2diZWRoMmFiZXZzMG9jNDkxNzRAZ3JvdXAuY2FsZW5kYXIuZ29vZ2xlLmNvbQ&amp;color=%23795548&amp;mode=WEEK&amp;title=La%20Compagnie%20des%20Jeux" *style*="border-width:0" *width*="800" *height*="400" *frameborder*="0" *scrolling*="no"></iframe>

*{# Calendrier pour une résolution pour petits écrans, avec un affichage par jour. #}*

    <iframe *class*="calendrierS" *src*="https://calendar.google.com/calendar/embed?height=600&amp;wkst=1&amp;bgcolor=%2305ae3b&amp;ctz=Europe%2FParis&amp;src=Y2ZqMGJob2diZWRoMmFiZXZzMG9jNDkxNzRAZ3JvdXAuY2FsZW5kYXIuZ29vZ2xlLmNvbQ&amp;color=%23795548&amp;mode=AGENDA&amp;title=La%20Compagnie%20des%20Jeux" *style*="border-width:0" *width*="400" *height*="600" *frameborder*="0" *scrolling*="no"></iframe>

</div>

Pour que le calendrier s’affiche efficacement sur toutes les tailles d’écrans, j’ai intégré à la page 3 iframes du calendrier avec des réglages différents (figs.23 24 et 25). Pour ne pas alourdir le chargement de la page, une seule se charge à la fois, pour la résolution d’écran pour laquelle elle est conçue, grâce à l’utilisation de la propriété CSS ‘**display : none ;’ et des ‘medias queries’**.

.calendrierM, .calendrierS {

  display: none;

}

@media screen and (max-width: 1410px) and (min-width: 810px) {

  .calendrierXL {

    display: none;

  }

  .calendrierM {

    display: block;

  }

}

@media screen and (max-width: 809px)

{

  .calendrierXL, .calendrierM {

    display: none;

  }

  .calendrierS {

    display: block;

  }

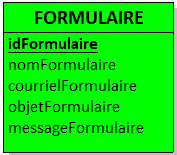
}







## Résolution d’un problème avec une documentation anglophone

Afin de pouvoir récupérer les données et pouvoir les classer, j’ai conçu le **formulaire de contact** de la page « Nous contacter » afin que les données soient stockées dans la base de données, via l’entité ‘**Formulaire**’ (fig.49). 

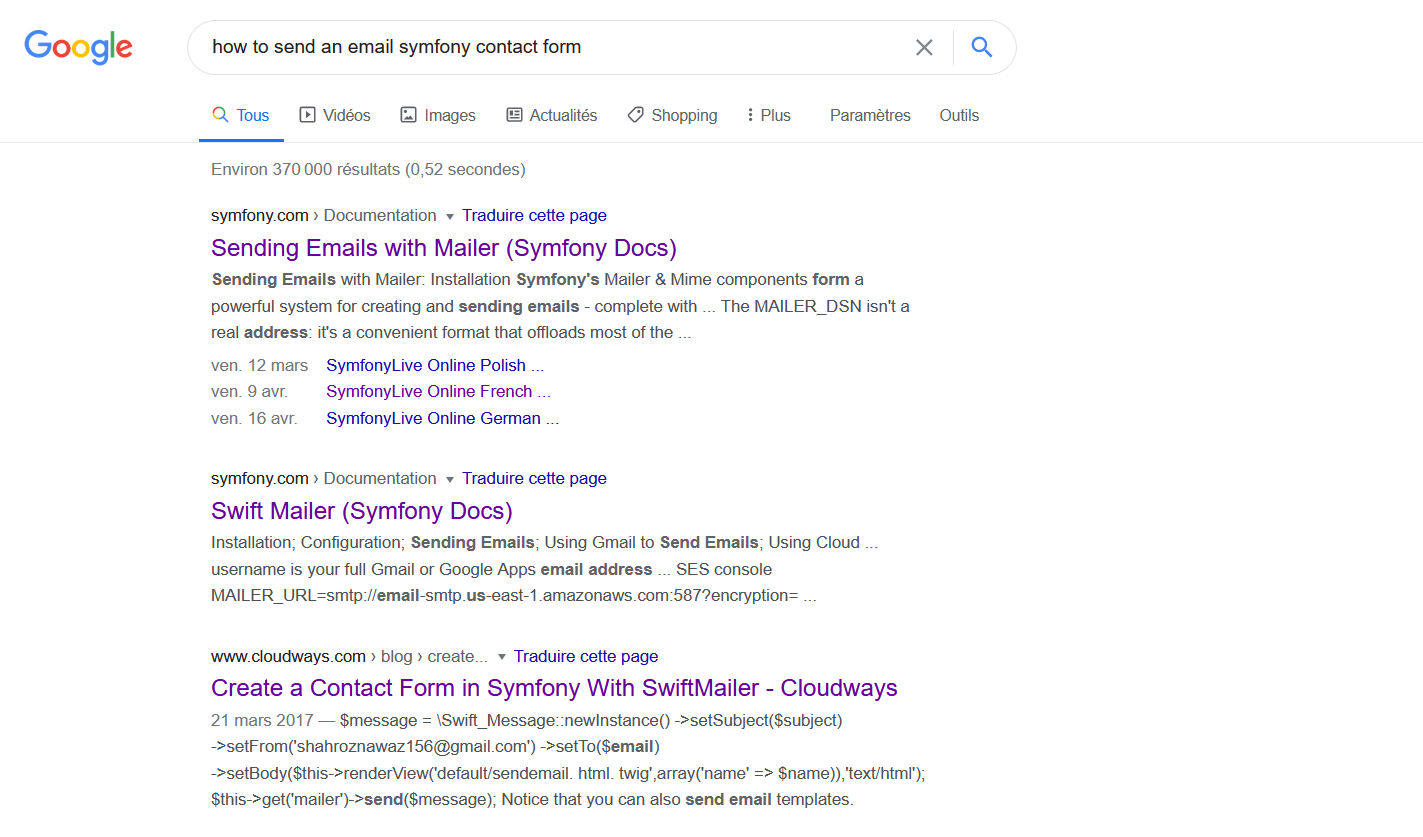
Seulement, afin d’améliorer l’expérience utilisateur, j’ai cherché à faire en sorte qu’à chaque message rédigé à partir du formulaire, un **courriel soit automatiquement envoyé** à la boite de réception lié au compte administrateur.

Tout au long de ma formation, j’ai eu l’occasion de consulter de nombreux sites anglophones, Stackoverflow, le site du W3C, les documentations des différents langages, rarement traduites en français, aussi n’ai-je pas d’appréhension à effectuer mes recherches en anglais lorsque je suis confronté à un problème.

La documentation de Symfony étant quasiment exclusivement disponible en anglais, j’ai effectué ma recherche à partir des mots clés :

***« How to send an email Symfony contact form »***

Un grand nombre de résultats renvoie vers un outil de Symfony appelé **« Swift Mailer »**(fig.50) :



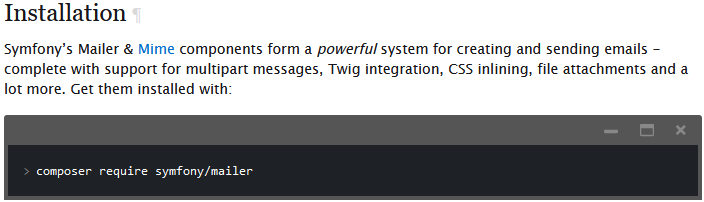
***Figure 51 - Recherche Google***

Une remarque est mise en évidence sur la documentation de Symfony le concernant (fig.51) : 

*« Dans Symfony 4.3, le composant* ***Mailer*** *a été ajouté et peut être utilisé à la place de Swift Mailer. »*

Assez logiquement, c’est aussi le premier résultat que renvoie la précédente recherche effectuée sur google (fig.50) : 

L’installation est assez simple (fig.52) :



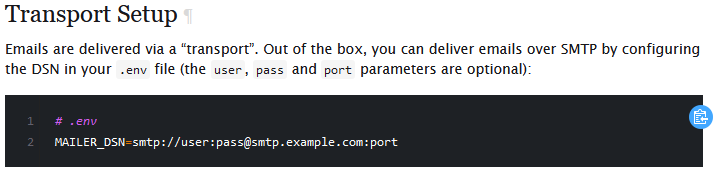
***Figure 53 - Installation Mailer***

*« Les composants Mailer et Mime de Symfony constitue un puissant système pour créer et envoyer des courriels – complétés par la prise en charge des messages en plusieurs parties, l’intégration de Twig, du CSS, le transfert de fichiers et bien plus. Installez-les avec : »*

composer require symfony/mailer

J’ai donc rentré cette ligne de commande dans mon terminal.

Il me fallait ensuite **configurer le transport du courriel** (fig.53).



***Figure 54 - Configuration de l'envoi***

*« Les courriels sont envoyés via un « transport ». En gros, vous pouvez envoyer des courriels via SMTP en configurant le DSN dans votre fichier ‘.env’. (les paramètres ‘user’, ‘pass’ et ‘port’ sont optionnels).»*

**Mon controlleur** allait également devoir être modifié (fig.54) :

*« Pour envoyer un courriel, obtenez une instance ‘Symfony\Component\Mailer\Mailer’ en indiquant le type ‘Symfony\Component\MailerInterface’ et créez un objet ‘Symfony\Component\Mime\Email’.*

*C’est bon ! Le message sera envoyé via le transport que vous avez configuré. »*

Derrière ces explications un peu barbares, il me suffit en fait de compléter mon controlleur en ajoutant les ‘use’ correspondant, et de créer ou de compléter la fonction par rapport à **où je souhaite afficher mon formulaire**.

Pour mon controlleur, j’obtiens ainsi :

*/\*\**

*\* @Route("/home", name="faq\_home", methods={"GET","POST"})*

*\*/*

    public function home(FAQRepository $fAQRepository, Request $request, MailerInterface $mailer): Response

    {

        $formulaire = new Formulaire();

        $form = **$this**->createForm(FormulaireType::class, $formulaire);

        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {

            $courriel=$formulaire->getCourrielFormulaire();

            $objet=$formulaire->getObjetFormulaire();

            $nom=$formulaire->getNomFormulaire();

            $message=$formulaire->getMessageFormulaire();

            $email = (new Email())

                ->from($courriel)

                ->to('votremail@gmail.com')

                ->subject('Message depuis le site internet : '.$objet)

                ->text('Expéditeur : '.$nom.\PHP\_EOL.

                    $message,

                    'text/plain');

            $mailer->send($email);

**$this**->addFlash('Succès', 'Votre message a bien été envoyé');

            $entityManager = **$this**->getDoctrine()->getManager();

            $entityManager->persist($formulaire);

            $entityManager->flush();

*// return $this->redirectToRoute('faq\_home');*

        }

        return **$this**->render('faq/home.html.twig', [

            'faqs' => $fAQRepository->findAll(),

            'formulaire' => $form->createView(),

        ]);

    }









Voilà enfin la configuration à appliquer dans mon fichier ‘.env’, où « votrecourriel@mail.com » est l’adresse de la boîte courriel servant à envoyer les courriels, et « votremotdepasse » le mot de passe d’application de celle-ci (fig.56) :

###> symfony/mailer ###

MAILER\_DSN=smtp://votrecourriel@mail.com:votremotdepasse@smtp.gmail.com?verify\_peer=0

###< symfony/mailer ###

## Mise en production

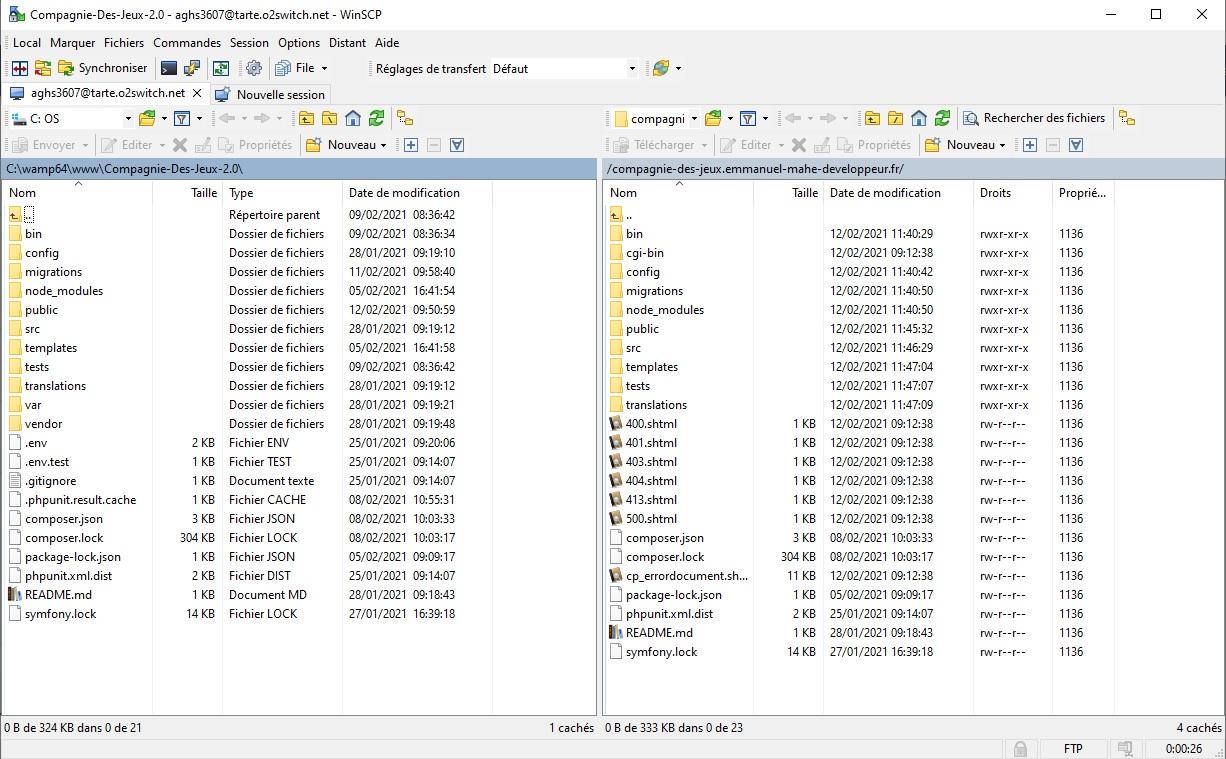
### Hébergement

Le projet étant toujours en développement, pour les besoins de l’examen, j’ai choisi d’héberger le site sur mon domaine personnel, sous le nom de domaine : **compagniedesjeux.emmanuel-mahe-developpeur.fr.**

J’utilise **O2Switch**, qui est un hébergeur français qui dispose d’un service de support technique particulièrement accessible.

A partir d…

Afin de transférer les données locales de mon ordinateur, je me suis servi du client SFTP (SSH File Transfer Protocol) **WinSCP.**



***Figure 22 - Interface de WinSCP***

Après avoir importé ma base données via cPanel, j’ai ensuite modifié mon fichier ‘.env’ :

DATABASE\_URL=mysql://MonIdentifiant:'MonMotDePasse'@MonIP:3306/LeNomDeMaBaseDeDonnées

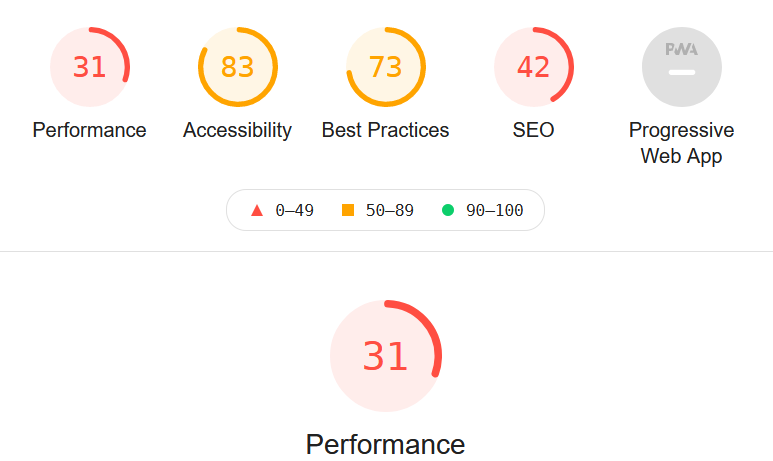
J’ai ajouté le fichier ‘.htaccess’ à la racine de mon dossier public.

…

### Optimisation des performances et Référencement

##### Lighthouse

Afin d’évaluer les performances du site internet, j’ai utilisé l’outil Google Lighthouse, qui permet de mesurer la qualité des pages web.



***Figure 58 - Analyse Google Lighhouse 1***

Les évaluations présentent beaucoup d’opportunités d’améliorations, proposées point par point :

* **SEO (Search Engine Optimisation) ou Optimisation pour les moteurs de recherche.**

Pour améliorer le référencement, j’ai complété les informations de metadonnées dans le **‘head’** de ma page HTML.

. . .

<meta *charset*="UTF-8">

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1">

    <meta *name*="description"

*content*="La Compagnie des Jeux, association de joueurs de jeux de société, de cartes, de rôles, de figurines et wargames sur Saint Nazaire (44)|Venez nous rencontrer !">

    <title>{% block title %}La Compagnie Des Jeux - Association de joueurs - Saint Nazaire{% endblock %}</title>

. . .

J’ai également pris soin de remplir l’attribut de texte alternatif ‘alt’ pour chacune de mes images.

Ex :

. . .

<div *class*="imgJeu">

     <img *loading*="lazy" *src*="{{ asset('uploads/photos/' ~ jeu.imgJeu) }}"

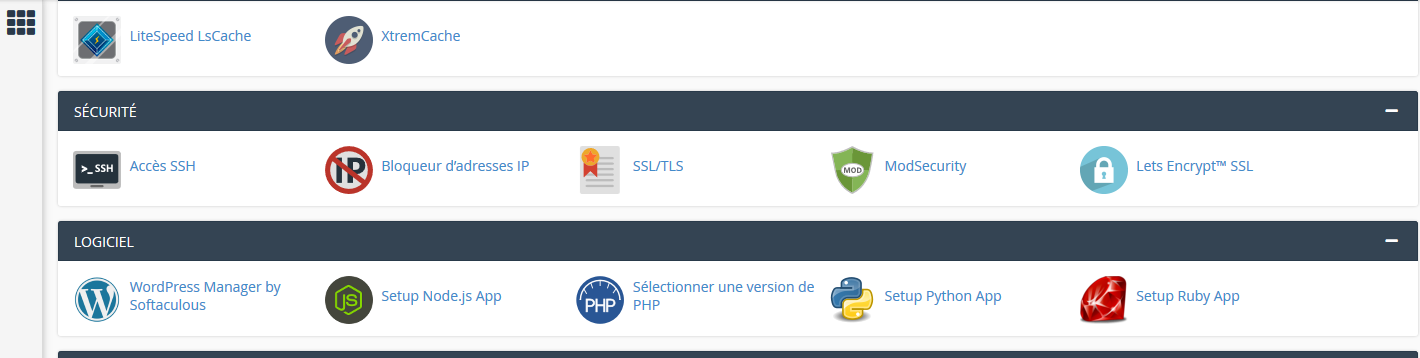
*alt*="image {{ jeu.soustitreJeu }}">

</div>

. . .

* **Bonnes Pratiques**

A partir du panneau de configuration de cPanel, pour augmenter la sécurité du site, et améliorer son référencement, j’ai mis en le protocole **‘https’.**



***Figure 59 - Let's Encrypt SSL***



***Figure 60 - Domaines certifiées***

D’autre part, j’ai changé le ratio des images afin qu’il soit plus adaptés à leurs différents usages.

* + ratio image /
  + twitch

Accessibilité

* + Contraste entre le texte et le fond de la page.
  + Alt pour les images /
  + Noms iframes /
  + Noms des liens pas assez discernables -> utilisation d’aria-label /

### Cookies/RGPD

##### Tarte au citron

Pour pouvoir gérer les cookies sur mon site, j’ai mis en place le système de gestion appelé ‘Tarte au citron’.

Après avoir téléchargé les fichiers nécessaires et les avoir inclus dans mon projet, j’ai copié ces scripts qui me permettent notamment de gérer les options de ce système :

<script *type*="text/javascript" *src*="{{asset('tarteaucitron/tarteaucitron.js')}}"></script>

<script>

    tarteaucitron.init({

        "privacyUrl": "", */\* URL de la page de la politique de vie privée \*/*

        "hashtag": "#tarteaucitron", */\* Ouvrir le panneau contenant ce hashtag \*/*

        "cookieName": "tarteaucitron", */\* Nom du Cookie \*/*

        "orientation": "middle", */\* Position de la banniére (top - bottom) \*/*

        "showAlertSmall": false, */\* Voir la bannière réduite en bas à droite \*/*

        "cookieslist": true, */\* Voir la liste des cookies \*/*

        "adblocker": false, */\* Voir une alerte si un bloqueur de publicités est détecté \*/*

        "AcceptAllCta": true, */\* Voir le bouton accepter tout (quand highPrivacy est à true) \*/*

        "highPrivacy": true, */\* Désactiver le consentement automatique \*/*

        "handleBrowserDNTRequest": false, */\* Si la protection du suivi du navigateur est activée  \*/*

        "removeCredit": false, */\* Retirer le lien vers tarteaucitron.js \*/*

        "moreInfoLink": true, */\* Afficher le lien 'voir plus d'infos' \*/*

        "useExternalCss": false, */\* Si false, tarteaucitron.css sera chargé\*/*

        "cookieDomain": '.my-multisite-domaine.fr', */\* Cookie multisite \*/*

        "readmoreLink": '/cookiespolicy' */\* Lien vers la page 'Lire plus' \*/*

    });

</script>

##### RGPD

Depuis le **bas de page du site (Footer**), l’utilisateur peut accéder à une page de **mentions légales.**





### Analyse des données

Afin de pouvoir analyser l’audience du site, **Google Analytics** a été intégré au site.

Pour cela, j’ai utilisé la documentation de mon **gestionnaire de cookies, Tarte au citron**, et j’ai inclus ce code dans mes pages :

*<!-- Global site tag (gtag.js) - Google Analytics -->*

    <script *type*="text/javascript">

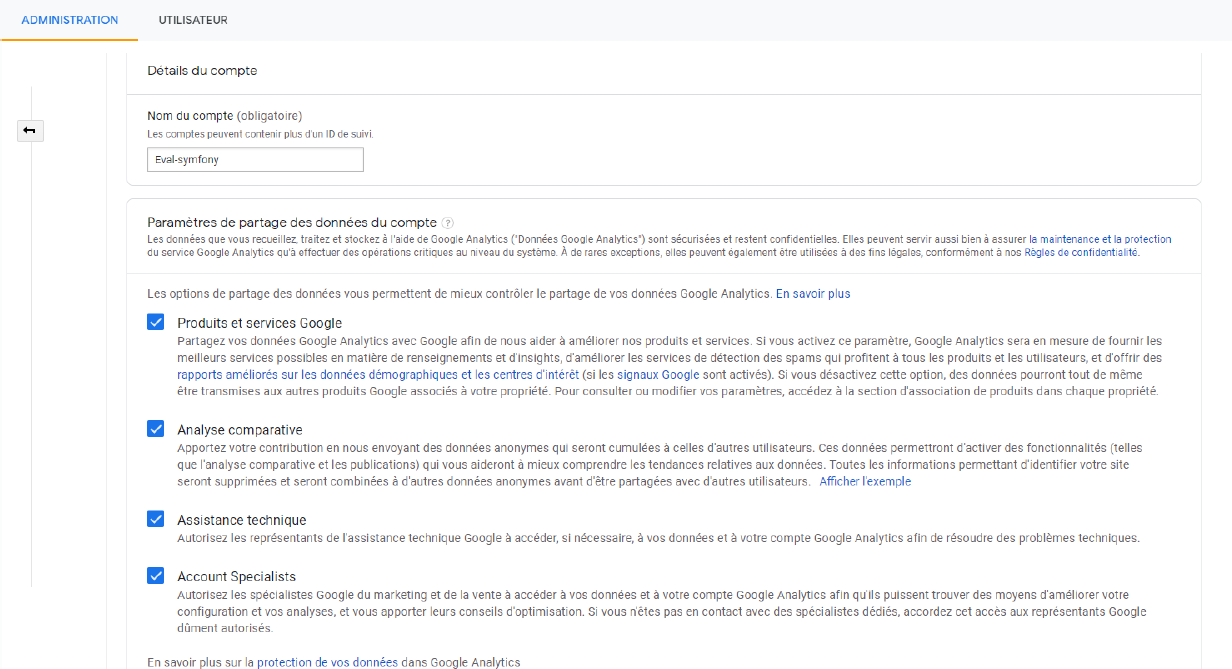
        tarteaucitron.user.gtagUa = 'G-XXXXXXXXX';

        tarteaucitron.user.gtagMore = function () { */\* add here your optionnal gtag() \*/* };

        (tarteaucitron.job = tarteaucitron.job || []).push('gtag');

    </script>

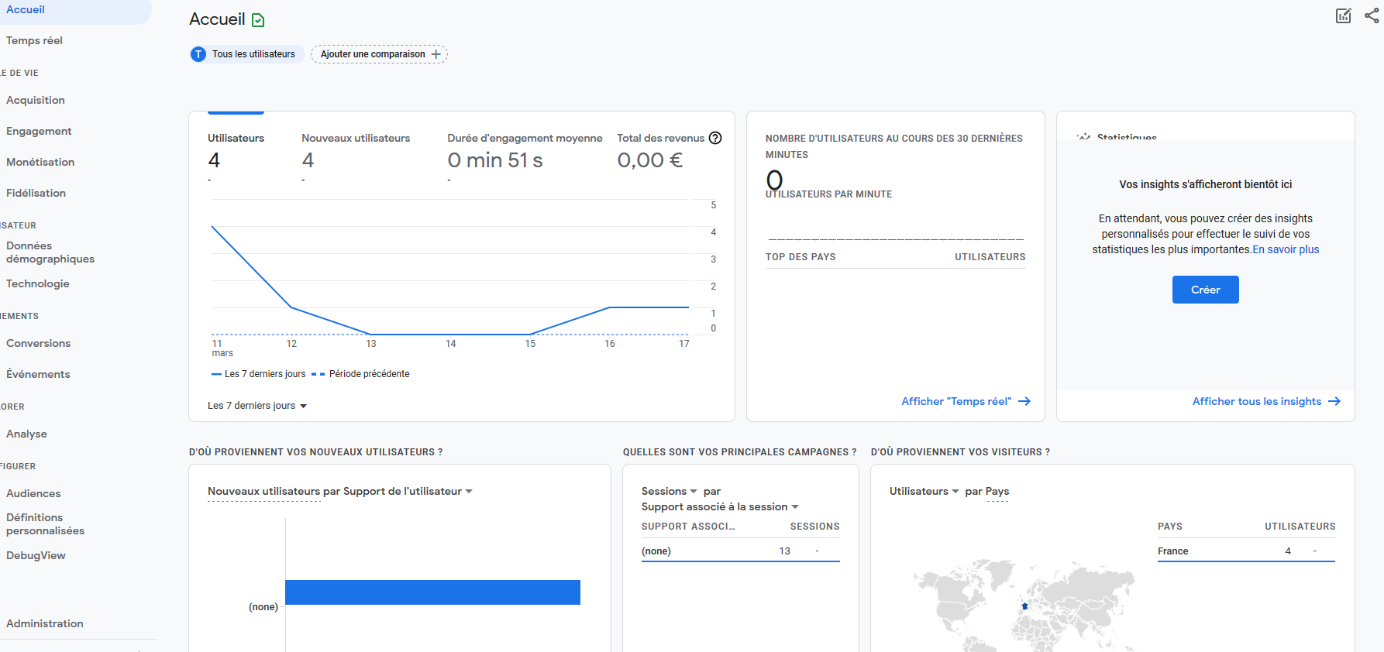
Pour profiter du service, il faut posséder un compte Google et s’inscrire (fig.58) :



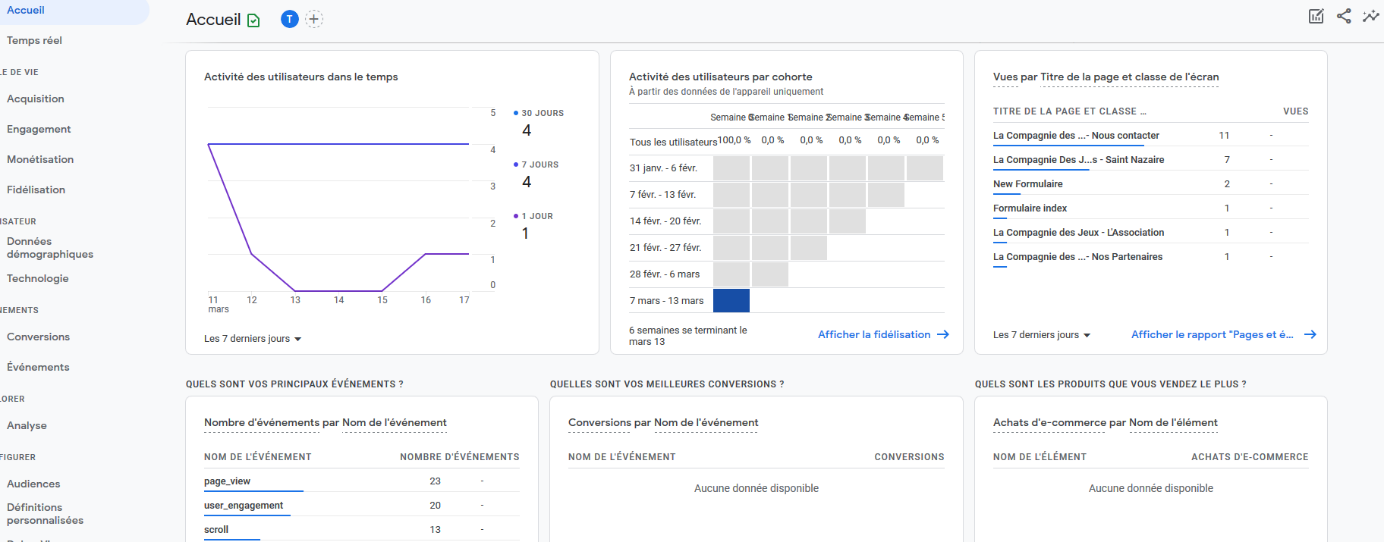
***Figure 62 - Inscription Google Analytics***

Cette inscription est suivie d’une phase de configuration, en fonction des objectifs du site internet et de l’identité du destinataire du site.  
Une fois toutes ces étapes franchies, il faut intégrer le code d’identification.

Google Analytics nous permet alors, via un tableau de bord personnalisable, de consulter et de pouvoir analyser l’activité de notre site.



***Figure 63 - Tableau de bord Google Analytics 1***



***Figure 64 - Tableau de bord Google Analytics 2***

*Remarque : Au vu de son affluence faible pour le moment, les données de mon site internet ne sont pour le moment pas très exploitables, et sont présentes à titre d’exemple.*

## Conclusion

Ce projet a été pour moi une entreprise enrichissante à plusieurs niveaux. Tout au long de son avancement, j’ai eu tour à tour l’occasion de me muer dans les différents rôles qui participent à la création d’un site internet. De **chef de projet** au tout début, je suis ensuite devenu **designer Web**, **intégrateur**, puis enfin **développeur**, à la fois **front-end et back-end**, pour finir par prendre le rôle de **data-analyst**.

Même si je suis en quelque sorte à l’origine du besoin (l’association a fait part de son envie de développer sa communication et je leur ai proposé mon aide), j’ai également beaucoup appris de la **relation client/entreprise**, de ses contraintes, surtout dans une période où il est compliqué de se réunir pour parler de vive-voix.

Enfin, après une période d’inactivité, réaliser ce projet et plus généralement cette formation m’a permis de reprendre un rythme de vie active, avec des **horaires à respecter, des dates limites à tenir, des contraintes à la fois techniques et matérielles** auxquelles réfléchir et prendre en compte.

Au jour où j’écris ces lignes, mon travail me laisse un goût d’inachevé. Une formation telle que celle que je viens de suivre est intense, et courte pour acquérir les connaissances pour concevoir et livrer un tel site avec si peu d’expérience. J’aurais aimé aller plus loin dans certaines fonctionnalités, travailler mieux le design, produire un code plus propre, plus épuré et plus efficace. Cette impression n’est sans doute que l’expression de toutes les choses et les méthodes que j’ai apprises au fur et à mesure de cette production, et qui m’aideront à faire de meilleurs choix dans mes projets futurs.

De même, si un site vitrine ne nécessite peut-être pas l’utilisation d’un outil aussi puissant que Symfony, qu’en entreprise un **CMS comme Wordpress ou Webflow** aurait été préféré afin de gagner en temps et en simplicité, je suis satisfait du choix de cette solution.  
Si ce site a une raison concrète à son existence, faire connaître une association que j’apprécie et l’aider à recruter de nouveaux membres, il était avant tout l’un des **supports principaux à ma formation**, et donc un prétexte pour pratiquer et acquérir des automatismes sur un framework aussi **populaire et polyvalent que Symfony**.

Le projet est encore en production, les activités de l’association sont à l’arrêt et il est difficile pour ses membres de s’organiser afin de produire le contenu qui agrémentera au final le site. Cependant, il est déjà fonctionnel et la Compagnie des Jeux semble satisfaite du rendu final.

Alors que j’arrive à l’issue de ma formation, je suis d’ores et déjà à la recherche d’un emploi, tout en continuant à me former de manière personnelle. Le métier de développeur Web présente l’avantage de laisser à portée de clic toutes les ressources et les aides que l’on peut avoir besoin si l’on souhaite se perfectionner ou apprendre de nouveaux langages et de nouveaux outils.   
J’ai notamment la volonté d’approfondir ma pratique de **JavaScript et de ses bibliothèques**, langage qui m’intéresse, que je n’ai pas pu assez mettre en pratique durant ma formation, et que j’aimerais maîtriser.

## Annexes

### Annexe 1 : Questionnaire de Définition de Projet

***1) Informations générales***

**▪ Nom du projet**

Compagnie des jeux 44

**▪ URL actuelle ou URL souhaitée**

Compagniedesjeux44.Fr compagniedesjeux44.org

▪ **Contacts pour le projet (nom, email, téléphone)**

Gombaud Xavier (x.gombaud@free.fr) 0674860989

**▪ Échéance de lancement du site**

Pas d’urgence

**▪ Les facteurs externes pouvant influer sur l’échéancier**

ND

***2) Pourquoi ?***

**▪ Quel est l’objectif que vous souhaitez atteindre avec ce projet (quelle problématique d’entreprise souhaitez-vous re soudre au travers de votre site web) ?**

Avoir une vitrine globale pour permettre d’axer notre communication sur un support évolutif et simple. Montrer que nous sommes actifs, notamment auprès des allergiques aux re seaux sociaux. Peut-être faire la promotion de notre chaîne YouTube si nous développons le contenu.

***3) Pour qui ?***

▪ **Qui est votre cible ?**

Les joueurs de jeux semi compétitif.

**▪ Décrivez votre cible (CSP, usage d’Internet...)**

Tous les âges, mais les 25+ particulièrement, toute CSP, usage d’internet de modéré a expert.

***4) Valeur***

**▪ Pourquoi votre cible aurait-elle besoin de vous ?**

Elle ne sait pas qu’il y a une asso pour partager ses passions.

**▪ Quel est son intérêt ?**

Trouver des joueurs passionne s, et pouvoir pratiquer son loisir avec 2 créneaux par semaine, et des moyens de communication h24 avec les autres membres.

**▪ Qu’offrez-vous de mieux que votre concurrence ?**

2 créneaux dans la semaine, inscription à l’année très accessible.

**▪ Comment sont perçus vos produits/services par votre cible ?**

Nous sommes perçus comme actif et accueillant.

**▪ Comment votre cible utilisera-t-elle votre site ?**

Pour apprendre à nous connaître.

**▪ Quels sont ses objectifs (ceux de la cible)**

Trouver des gens qui ont les me mes centres d’intérêt et s’amuser.

***5) Perception***

**▪ Listez des adjectifs sur la manie re dont vous souhaitez que votre cible perçoive votre nouveau site**

Accueillant, épuré, intéressant.

**▪ Qu’y a-t-il de différent par rapport a la perception actuelle de votre entreprise (ou ancien site web)**

Changement de jeux, trop chargé.

**▪ Listez des sites web que vous aimez. Qu’aimez-vous dans ces sites ?**

https://www.zlup.fr/ (pour le côté simple et accessible, mais esthétique bof).

***6) Le contenu***

**▪ Possédez-vous un contenu existant a fournir pour votre site ou est-ce un nouveau contenu**

**? Qui rédigera le nouveau contenu ?**

On a des photos supports. Les membres du bureau rédigeront le nouveau contenu, et hors bureau s’il y a des volontaires.

**▪ Quels sont les visuels dont nous devrons nous servir ? Existent-ils ou devrons-nous les créer (logos, photos, charte graphique...) ?**

Logo oui, chartes graphique raccord avec le logo (noir blanc vert).

***7) Technologie***

**▪ Aurez-vous besoin de transactions sécurisées (dans le cadre d’un e-commerce) ?**

Non

**▪ Aurez-vous besoin d’autres fonctionnalités de programmation, comme une fonction de recherche par exemple ?**

Selon le contenu oui.

***8) Marketing et communication***

**▪ Comment souhaitez-vous que votre cible vous trouve ?**

Via des liens sur nos support de communication physique, via des communications partenaires, publicité sur les forums de jeux.

**▪ Quels leviers initieront une visite (back links ou liens entrants, incentive (incitation au clic : concours, ...etc.), résultats organiques (référencement naturel, non payant), liens sponsorisés**

**(Achat de mots-clés) ?**

ND

**▪ Si vous avez déjà un site web, comment communiquez-vous votre URL online et offline ?**

Il n’est plus actif

**▪ Avez-vous déjà une ide e sur la stratégie marketing que vous souhaitez adopter pour promouvoir votre site ? Si oui, pouvez-vous nous la décrire ?**

Créer des QR-code pour les placer sur des supports de communication, le mettre sur nos flyers.

***9) Mises à jour***

**▪ Qui est responsable de la mise à jour et de la rédaction des contenus ?**

Le bureau

**▪ Notes et commentaires additionnels :**