

DOSSIER DE SITUATION N°1

SOLUTION APPLICATIVE WEB

BIBLIOTHEQUE MUSICALE

NEMO TORRES

Table des matières :

• Contexte.....	3
• Présentation du client.....	4
• Environnement existant.....	5
• Base de données.....	6
• fonctionnalités (besoin du client).....	7
• Explication du code.....	8
• Technologies utilisées.....	6
• Tests.....	11
• Mise en production.....	12
• Conclusion.....	13

Le contexte



la M2L décide en 2000 d'acquérir et réhabiliter la Maison Régionale des Sports de Lorraine.

Aujourd'hui la M2L représente : 6 500 clubs, plus de 525 000 licenciés et près de 50 000 bénévoles.

Un tel établissement est à la fois un facteur fort de cohésion et de qualité du sport régional. Il a pour vocation d'héberger les structures sportives régionales, de leur fournir des services administratifs, comptables et juridiques .

Le site où se trouve la M2L comprend plusieurs bâtiments offrant des bureaux aux ligues, ainsi que des espaces mutualisés tels qu'un amphithéâtre de 200 places, des salles de réunion, une salle de formation multimédia équipée et des locaux pour les différents organismes. Cette infrastructure permet aux dirigeants, bénévoles et salariés des associations sportives de se réunir, d'échanger et de se former dans des conditions optimales pour l'administration de leur propre ligue.

Le client

Dans cette situation nous projetons que nous travaillons pour la Maison des Ligues de Lorraine aussi appelée M2L située dans la banlieue de Nancy.

C'est un établissement dédié à l'accueil et au soutien des ligues sportives de Lorraine. inancée par le Conseil Régional de Lorraine et administrée par le Comité Régional Olympique et Sportif de Lorraine (CROSL), elle a pour mission principale de fournir des espaces fonctionnels et des services adaptés aux besoins des structures sportives hébergées.

Cela peut se présenter de plusieurs manières :

- Hébergement et soutien aux ligues sportives :
Cela signifie mettre à disposition des espaces pour les ligues et comités sportifs régionaux mais aussi faciliter la collaboration entre les différents comités au sein de la M2L.
- Formation et accompagnement :
la M2L organise la formations de bénévoles, dirigeants et salariés des différentes associations sportives. Elle propose aussi des ateliers et modules sur la gestion associative, la communication et le financement.
- Appui au développement du sport en Lorraine :
Participe à l'élaboration de politiques sportives régionales en lien avec les collectivités locales ainsi que le développement du sport notamment du handisport.
- Mise à disposition d'équipements et de services :
La M2L propose des salles de réunion, des amphithéâtres pour des discours ou remises de prix et des centres de formation multimédias. Mais la M2L peut fournir d'autres ressources comme des ressources immatérielles tels que des ressources logicielles ou du support informatique et technique pour tout les organismes sportifs de Lorraine.
C'est dans le cadre de cette mission que nous mettrons en places des solutions applicatives informatique.



Environnement Existant

Voici un audit de la M2L :

- Aucun système de gestion de base de données
- Aucun CMS, framework ou technologie imposée
- Aucun système de sécurité ou procédure de sauvegarde
- Aucun réseau ou parc informatique identifié
- Aucun serveur ou hébergement configuré

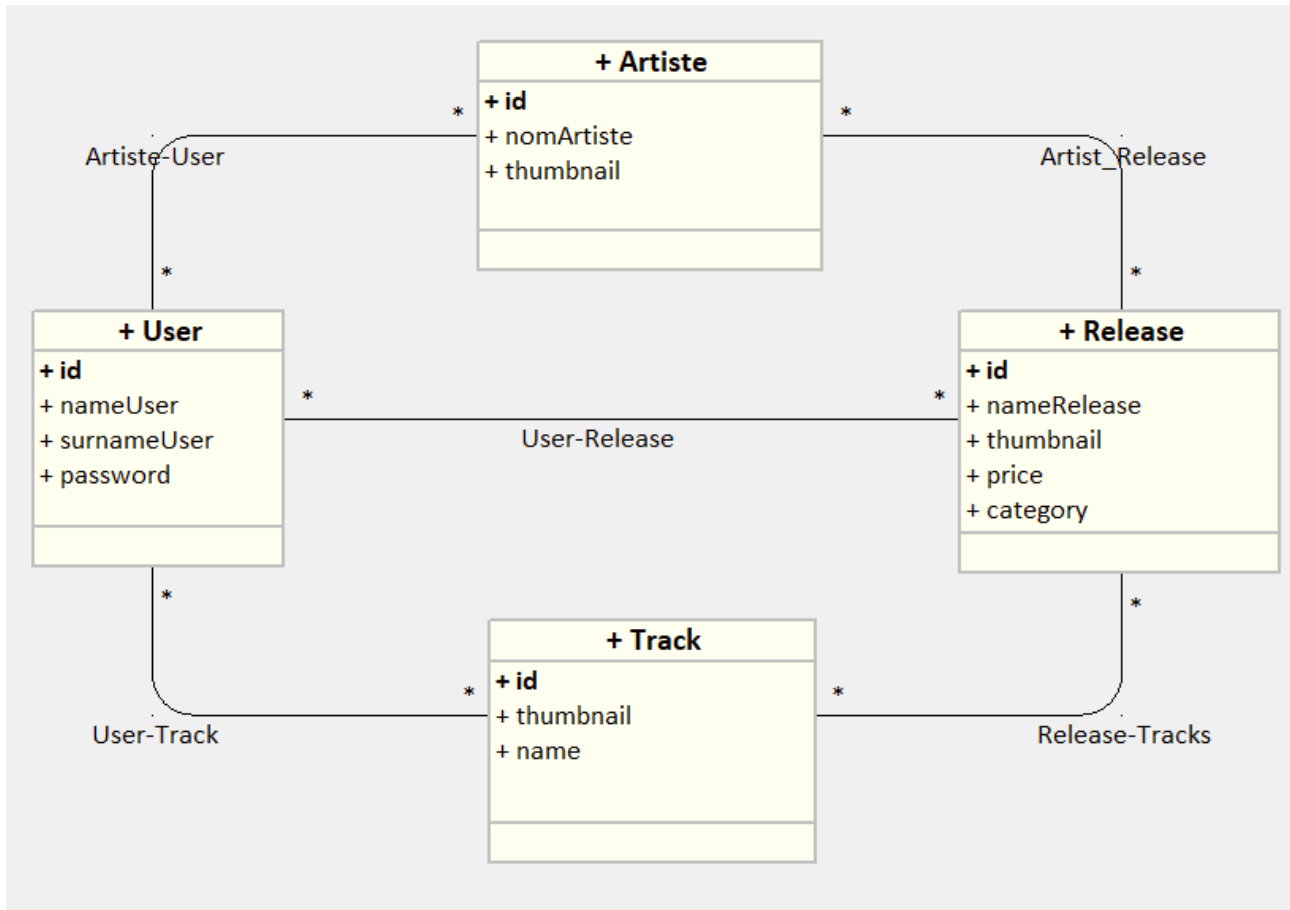
Cela nous a permis de partir d'un projet vierge et de proposer une architecture entièrement adaptée aux besoins du projet.

Nous sommes partis d'une simple liste d'énumération de titres musicaux que souhaitait M2L et une liste d'utilisateurs à rajouter dans une éventuelle base de données.

Nous avons accès à un logo et le code couleur rgb des couleurs qui forment l'identité de M2L.

Base de Données

Pour la réalisation de la base de donnée nous avons tout d'abord réalisé un UML pour bien visualiser les attentes de la solution.



Un user peut rajouter dans sa bibliothèque un artiste ou des albums (release) ou des pistes de musique (track) à sa bibliothèque personnelle pour gérer sa propre bibliothèque.

Pour réaliser cette base de données nous sommes passés par l'ORM doctrine.

Les Fonctionnalités (besoin client) :

La M2L nous demande à présent de réaliser une bibliothèque musicale qui a pour but de permettre de constituer une playlist pour des événements sportifs tels que des matchs discours, etc.

Cette solution applicative sera une application web qui sera développée en HTML, CSS, JS pour le front-end et en PHP pour le back-end.

Nous avons donc proposé à la M2L d'utiliser un framework nommé Symfony qui permettra d'utiliser tout ses différents composants.

L'avantage de symfony c'est de pouvoir administrer facilement une base de données grâce à Doctrine et créer des entités qui vont jouer le rôle de table dans une base de données.

Ainsi nous pourrions facilement récupérer les infos des musiques juste en récupérant les différentes instances de l'entité musique mais aussi toutes ses propriétés tels que l'artiste qui a créé la musique, l'album dont elle provient, la date de sortie pour déterminer si on passe une musique récente ou non.

Explications de Code

```
#[ORM\Entity(repositoryClass: ReleaseRepository::class)]
25 references | 0 implementations
class Release
{
    1 reference
    const ALBUM = 0;
    1 reference
    const EP = 1;
    1 reference
    const SINGLE = 2;

    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    1 reference
    private ?int $id = null;

    //(type: Types::DATE_IMMUTABLE): Annotation qui ajoute des infos pour le php qui ne peut pas
    #[ORM\Column(type: Types::DATE_IMMUTABLE)]
    2 references
    private ?\DateTimeImmutable $releasedAt = null;

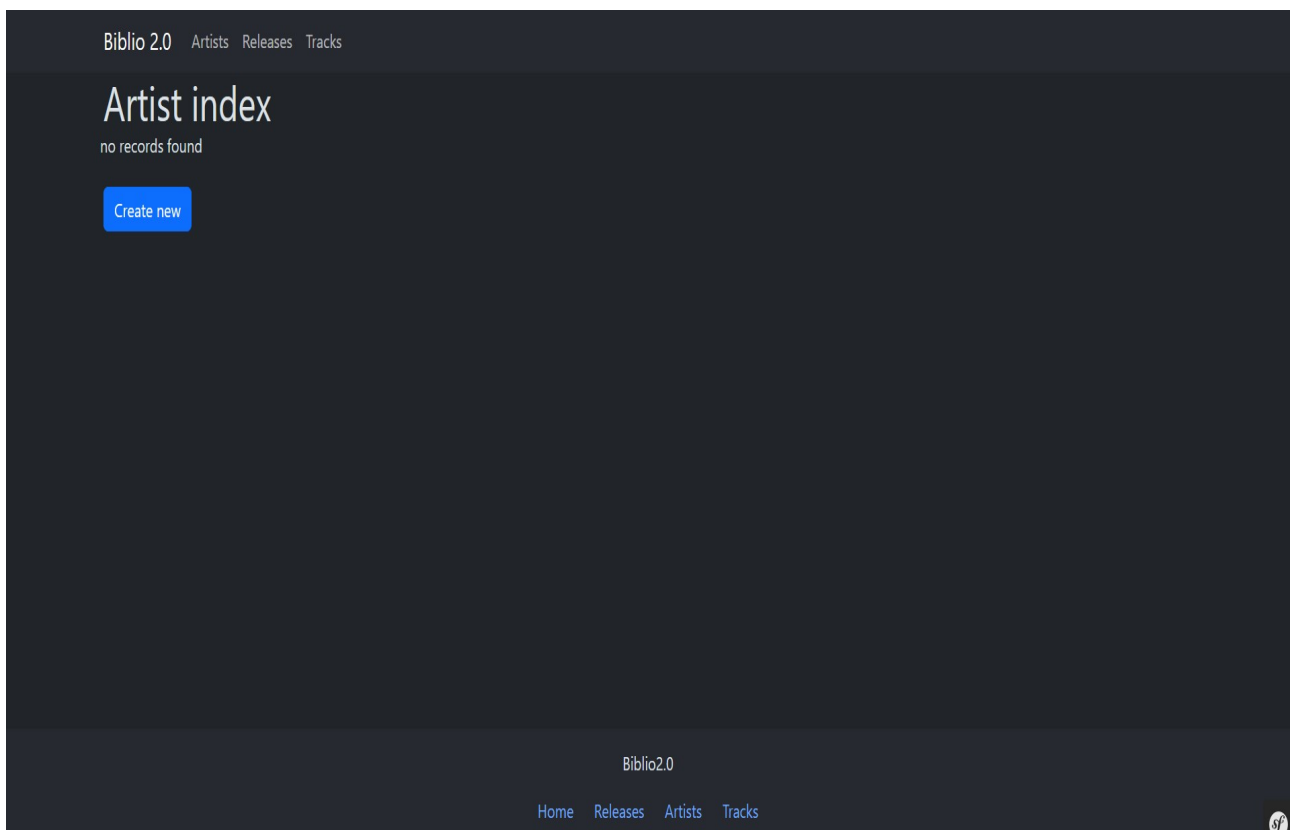
    #[ORM\Column(length: 255)]
    2 references
    private ?string $title = null;

    #[ORM\Column(length: 255)]
    2 references
    private ?string $thumbnailUrl = null;

    #[ORM\Column]
    2 references
    private ?int $type = null;

    #[ORM\ManyToOne(inversedBy: 'releases')]
    #[ORM\JoinColumn(nullable: false)]
```

Ceci est un exemple de l'ORM Doctrine, il m'a permis de créer une entité « release » qui correspond aux différents albums d'un artiste et cette entité possède des attributs tels que une date de sortie et un titre. L'avantage de Doctrine est de permettre d'écrire du code indépendant du SGBD (MySQL, PostgreSQL, SQLite, etc...). On peut facilement changer de base sans modifier le code applicatif. De plus, Doctrine gère nativement les relations entre entités (OneToOne, OneToMany, ManyToMany) et même l'héritage dans les entités (Single Table Inheritance, Class Table Inheritance, etc...).



Ceci est un aperçu de la page administrateur, elle permet de créer des instances de l'entité artiste sans passer par du code. Idéal pour un administrateur néophyte en informatique.

Ceci est la version prototype de la page il n'y a pas encore de logo et c'est vraiment la base de la décoration du site. En cliquant sur create new je serai amené à rentrer les informations correspondantes à mon objet ce qui va créer une instance de l'entité artiste.

Biblio 2.0

☰

Create new Release

Released at

jj / mm / aaaa

📅

Release Title

Enter the release title

Thumbnail URL

Enter the thumbnail URL

release type

EP

▼

Artist

▼

Save

Go back

Biblio2.0

Voici un exemple de la page de création de Release, sachant que une release nécessite un artiste il est impossible de créer une release sans artiste, il faut une instance de l'entité artiste qu'on a déjà créé sur le site lui même ou par terminal.

Les technologies utilisées :

-PHP aussi appelé hypertext Preprocessor est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques par un serveur web, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook et Wikipédia. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

- Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour écrire les pages web.

- nous avons utilisé le framework Symfony, un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web tel que :

- Doctrine qui est un ORM de base de données qui est aussi un logiciel libre sous licence GNU LGPL. Cet ORM permet la persistance transparente des objets PHP. C'est l'interface qui permet de faire le lien ou "mapping" entre les objets et les éléments de la base de données.

- Son fonctionnement MVC qui permet de structurer le projet et le rendre plus sécurisé car ce modèle, aussi appelé : Model-View-Controller ou Modèle-Vue-Contrôleur, met l'accent sur la séparation entre la logique métier et l'affichage du logiciel. Cette «séparation des préoccupations» permet une meilleure répartition du travail et une maintenance améliorée.

- Composer qui est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin.

Les Tests :

Nous avons réalisé les tests manuellement, un des avantages de ce projet est qu'en cas de mauvais fonctionnement de l'application, Symfony affichera rapidement une erreur. Nous avons donc créé des titres, des releases et des artistes et nous avons testé par exemple ce qui se passerait si nous supprimions l'artiste, nous avons décidé que cela enlèverait l'artiste et les tracks et release en cascade (si la personne décide d'enlever l'artiste de sa bibliothèque c'est qu'elle ne souhaite plus garder ses titres)

Nous avons aussi testé la sécurité du système de connexion, que se passe-t-il si je tente une injection dans ma page, si je me trompe de mot de passe, si j'essaie d'accéder aux données sans me connecter. Dans le contexte de n'importe quelle solution applicative, un minimum de sécurité est primordial.

Mise en Production :

Nous avons réalisé l'UML de notre solution applicative pour en saisir tout les enjeux.

Nous avons ensuite utilisé doctrine pour générer les entités nécessaires dans notre base de données.

Symfony est un outil puissant qui nous a ensuite permis de générer un CRUD de nos entités, il génère des formulaires mais nous avons décidé de les personnaliser.

Nous avons pu faire la partie ajout d'utilisateurs facilement avec Symfony, nous avons réalisé notre propre page de login avec notre propre code.

Nous avons ensuite travaillé sur la partie liste d'artistes/releases/tracks dépendant de l'utilisateur connecté ce qui crée une vue pour chaque utilisateur, pour cela il faut qu'à la création de l'entité artiste il soit rattaché à l'utilisateur qui l'a créé. Nous avons donc rajouté cela dans nos différentes entités.

Conclusion :

Cette solution applicative nous a permis de réaliser la puissance de certains framework comme Symfony qui nous donnent accès à une palette d'outils hallucinante. Que ce soit l'ORM Doctrine mais encore le langage twig, le serveur intégré dans le framework, ou encore la facilité à mettre en place une architecture MVC qui représente une bonne pratique à prendre.

Cette solution applicative aurait pu être amélioré avec notamment un moyen de vraiment lancer la musique des différentes musiques en passant par l'api de Spotify par exemple.

Mettre en place une telle fonctionnalité aurait nécessité la gestion de l'authentification OAuth avec Spotify et l'utilisation d'un lecteur intégré. Cela aurait représenté un challenge technique stimulant, tout en donnant une vraie valeur ajoutée à l'application.

En somme, ce projet a été une excellente opportunité pour approfondir notre compréhension des frameworks modernes, et il ouvre des perspectives intéressantes d'évolutions futures en matière d'interactivité, d'ergonomie et d'intégration d'APIs externes.

Administration des systèmes et des réseaux (E6 – SISR)

Conception et développement d'applications (E6 – SLAM)

PAGE DE PRÉSENTATION DU DOSSIER

N° d'inscription ¹ : | 0 | 2 | 4 | 4 | 3 | 8 | 3 | 7 | 3 | 0 | 6 |

NOM : TORRES

PRÉNOM : Nemo

Date de passage ¹ : / / 2025	Heure de passage ¹ :h.....
--	---

CATÉGORIE CANDIDAT ² (UNE CASE À COCHER)	
<input type="checkbox"/> Scolaire	<input type="checkbox"/> Ex-scolaire
<input type="checkbox"/> Apprenti	<input type="checkbox"/> Ex-apprenti
<input type="checkbox"/> Formation professionnelle continue	<input type="checkbox"/> Ex-formation professionnelle continue
<input type="checkbox"/> Expérience professionnelle 3 ans	

¹ Informations communiquées sur votre convocation envoyée courant mars 2025 sur votre compte **Cyclades**

² Informations communiquées sur votre confirmation d'inscription.

Tampon de L'établissement

SIEC – maison des examens

7 rue Ernest Renan
94749 ARCUEIL
CEDEX
Tél : 01 49 12 23 00
www.siec.education.fr



DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 01				
Nom, prénom : TORRES Nemo		N° candidat : 02443837306				
Épreuve ponctuelle	Contrôle en cours de formation	Date :				
Organisation support de la réalisation professionnelle La Maison des Ligues de la Lorraine, établissement du Conseil Régional de Lorraine, est responsable de la gestion du service des sports et en particulier des ligues sportives ainsi que d'autres structures hébergées. La M2L doit fournir les infrastructures matérielles, logistiques et des services à l'ensemble des ligues sportives installées. Elle assure l'offre de services et de support technique aux différentes ligues déjà implantées (ou à venir) dans la région. M2L souhaite avoir à disposition une bibliothèque musicale pour l'organisation d'événements.						
Intitulé de la réalisation professionnelle Création d'une solution SaaS pour la gestion d'une bibliothèque musicale						
Période de réalisation : 01/10/2024 - 03/12/2024		Lieu : EPSI MONTPELLIER				
Modalité : Seul		X En équipe				
Compétences travaillées Concevoir et développer une solution applicative Assurer la maintenance corrective ou évolutive d'une solution applicative Gérer les données						
Conditions de réalisation ¹ (ressources fournies, résultats attendus) <table border="0"> <tr> <td>Ressources fournies :</td> <td>Résultats attendus :</td> </tr> <tr> <td> <ul style="list-style-type: none"> • Cahier des charges M2L • SQLite • PHP/Symfony • Doctrine </td> <td> <ul style="list-style-type: none"> • Application web fonctionnelle permettant l'ajout de nouveaux titres et gestion des utilisateurs • Base de données administrant les titres et les utilisateurs </td> </tr> </table>			Ressources fournies :	Résultats attendus :	<ul style="list-style-type: none"> • Cahier des charges M2L • SQLite • PHP/Symfony • Doctrine 	<ul style="list-style-type: none"> • Application web fonctionnelle permettant l'ajout de nouveaux titres et gestion des utilisateurs • Base de données administrant les titres et les utilisateurs
Ressources fournies :	Résultats attendus :					
<ul style="list-style-type: none"> • Cahier des charges M2L • SQLite • PHP/Symfony • Doctrine 	<ul style="list-style-type: none"> • Application web fonctionnelle permettant l'ajout de nouveaux titres et gestion des utilisateurs • Base de données administrant les titres et les utilisateurs 					
Description des ressources documentaires, matérielles et logicielles utilisées <ul style="list-style-type: none"> • Cahier des charges M2L • Documentation de Symfony • Documentation de SQLite 						
Modalités d'accès aux productions ³ et à leur documentation Lien de production : naimau-del.github.io/myportfolio/#portfolio Lien repo Git : https://github.com/Naimau-del/projet1 Lien de documentations : <ul style="list-style-type: none"> • Documentation technique : nsh.xyz/8fd4b6 • Documentation utilisateur : nsh.xyz/8fd4b6 • Documentation administrateur : nsh.xyz/8fd4b6. 						

⁵ En référence aux conditions de réalisation et ressources nécessaires du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

⁶ Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

⁷ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁸ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle

(verso, éventuellement pages suivantes)

Épreuve E6 - Conception et développement d'applications (option SLAM)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**Contexte :**

La Maison des Ligues de la Lorraine, établissement du Conseil Régional de Lorraine, est responsable de la gestion du service des sports et en particulier des ligues sportives ainsi que d'autres structures hébergées. La M2L doit fournir les infrastructures matérielles, logistiques et des services à l'ensemble des ligues sportives installées. Elle assure l'offre de services et de support technique aux différentes ligues déjà implantées (ou à venir) dans la région.

M2L souhaite avoir à disposition une bibliothèque musicale pour l'organisation d'événements.

Afin de répondre à ce besoin, nous avons choisi d'utiliser Symfony afin de créer une bibliothèque musicale qui permet de se connecter et d'avoir sa propre bibliothèque.

Procédure de développement :

- Développement du front via Twig
- développement du back en PHP
- Implémentation d'une base de donnée SQLite
- Gestion d'authentification

Test du bon fonctionnement :

- ajout/retrait de titres
- ajout d'utilisateurs
- test des pages administrateurs

Schéma :