

Page de garde

Rapport de stage

Développeur Web

effectué par El Arche Wassim

du 06/01/2025 au 07/02/2025

Tuteur de stage : M.Mesle

Superviseur académique : M.Mesle

Établissement : Lycée Charles De Foucauld

Entreprise d'accueil : Lycée Charles De Foucauld , Paris

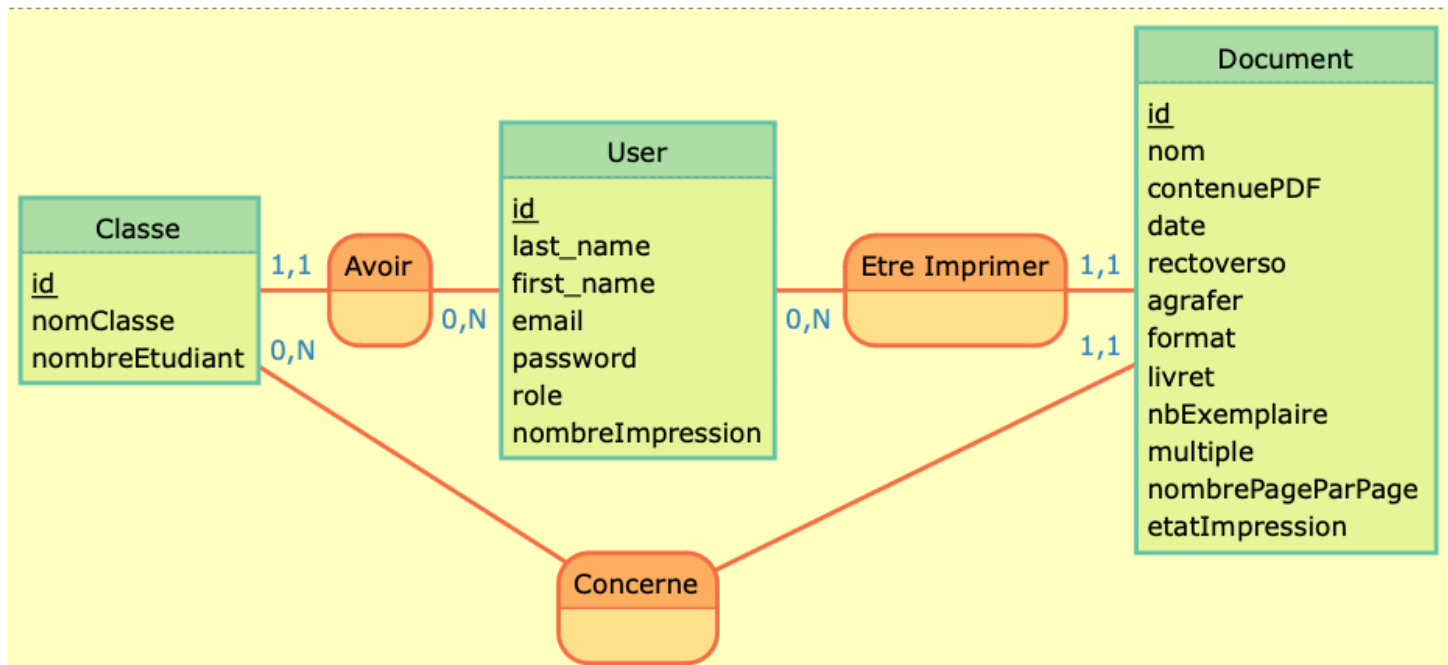
SOMMAIRE

I / Création du MCD et de la maquette	3
II / Création de la base de données.....	5
III / Controllers/Middlewares/Routes	9
IV / Les events , leurs listeners et notifications ..	15
V / Les policies et vues	16
VI / Docker	21

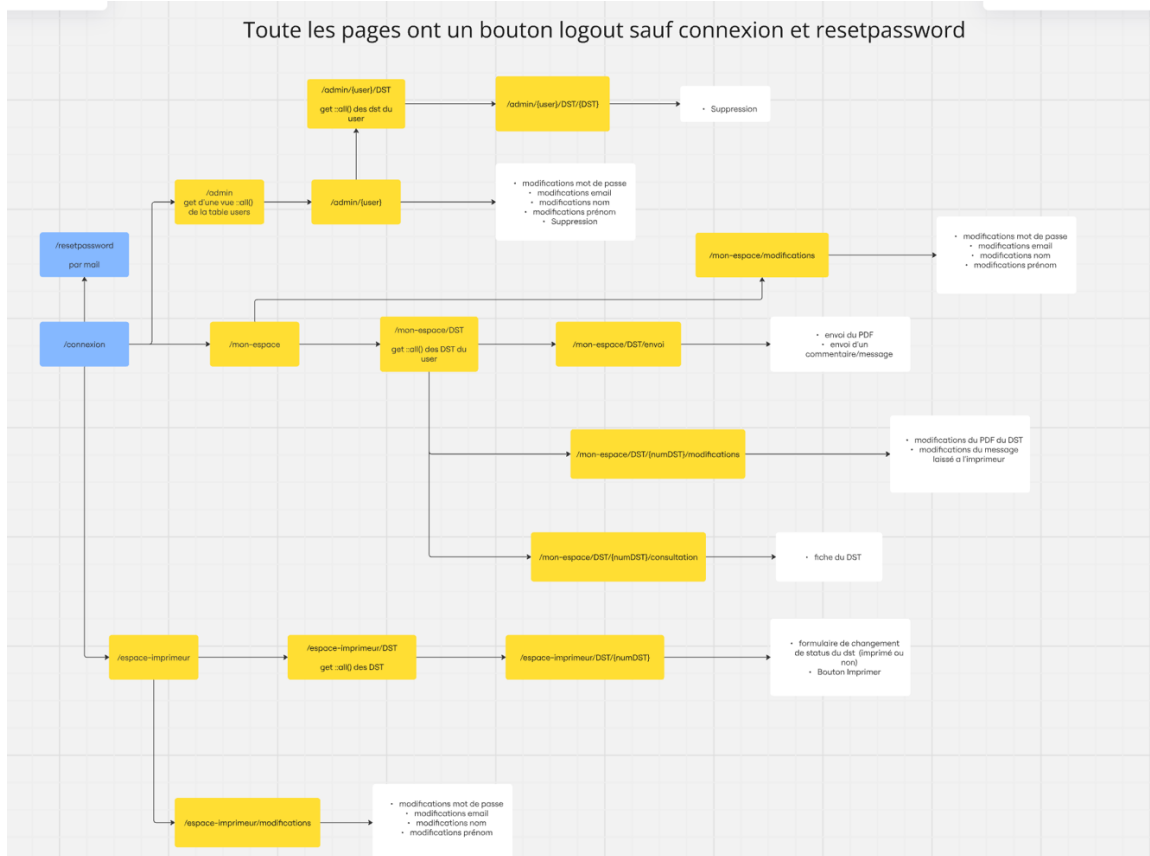
I / Création du MCD et de la maquette

MCD :

- Une classe peut avoir aucun ou plusieurs documents
- Un document est attribuer a une seul et unique classe
- Un utilisateur peut avoir aucune ou plusieurs classe
- Une classe est prise par un et un seul utilisateur
- Un utilisateur peut imprimer aucun ou plusieurs documents
- Un document est imprimé par un seul et unique utilisateur



Maquette :



II / Création de la base de données

1) Création de la table users

```
Schema::create('users', function (Blueprint $table) {  
    $table->id();  
    $table->string('last_name');  
    $table->string('first_name');  
    $table->string('email')->unique();  
    $table->timestamp('email_verified_at')->nullable();  
    $table->string('password');  
    $table->integer('role');  
    $table->integer('nombreImpression');  
    $table->rememberToken();  
    $table->timestamps();  
});
```

Rôle :

- 1 = Administrateur
- 2 = Professeur
- 3 = Imprimeur

```
public function isAdmin(){  
    return $this->role == 1;  
}  
  
public function isTeacher(){  
    return $this->role == 2;  
}  
  
public function isPrinter(){  
    return $this->role == 3;  
}
```

2) Création de la table classes

```
public function up(): void
{
    Schema::create('classes', function (Blueprint $table) {
        $table->id();
        $table->integer('user_id');
        $table->String('nomClasse');
        $table->integer('nombreEtudiant');
        $table->timestamps();
    });
}
```

3) Création de la table Documents

```
public function up(): void
{
    Schema::create('documents', function (Blueprint $table) {
        $table->id();
        $table->integer('user_id');
        $table->integer('classe_id')->nullable();
        $table->string('nom')->nullable();
        $table->string('contenuePDF')->nullable();
        $table->date('date')->nullable();
        $table->boolean('rectoverso')->nullable();
        $table->string('message')->nullable();
        $table->boolean('agrafer')->nullable();
        $table->String('format')->nullable();
        $table->boolean('livret')->nullable();
        $table->boolean('nbExemplaire')->nullable();
        $table->boolean('multiple')->nullable();
        $table->string('documentNom');
        $table->integer('nombrePageParPage')->nullable();
        $table->boolean('etatImpression')->nullable();
        $table->timestamps();
    });
}
```

4) Création de la table Imprimante

```
public function up(): void
{
    Schema::create('imprimantes', function (Blueprint $table) {
        $table->id();
        $table->boolean('Disponible');
        $table->integer('nombreFeuille');
        $table->integer('nombreImpression');
        $table->timestamps();
    });
}
```

5) Création d'un seeder

```
User::create([
    'first_name' => 'printer',
    'last_name' => 'printer',
    'email' => 'test@printer.com',
    'role'=> 3,
    'nombreImpression' => 0,
    'password' => bcrypt('a'),
]);

Classe::create([
    'user_id' => '2',
    'nomClasse' => 'SI02',
    'nombreEtudiant' => 28,
]);

Document::create([
    'user_id' => '2',
    'classe_id' => '1',
    'nom' => 'Test document',
    'contenuePDF' => "" ,
    'date' => '2024-12-12' ,
    'rectoverso' => 0 ,
    'agrafer' => 0 ,
    'format' => 'A4',
    'livret' => 0 ,
    'multiple' => 0 ,
    'documentNom' => "",
    'nombrePageParPage' => 0,
    'etatImpression' => 0,
]);

Imprimante::create([
    'Disponible' => 1,
    'nombreFeuille' => 100000,
```


III / Controllers/Middlewares/Routes

Pour le système d'authentification nous avons utilisé une bibliothèque nommée Laravel Breeze qui se chargera de cela

Document Controller :

Routes	Methode HTTP	Description
document/create	GET	Permet d'afficher le formulaire pour créer un document (document.create)
document	POST	Permet d'enregistrer un document en base de données (document.store)
document/{i}/edit	GET	Permet d'afficher le formulaire de modifications d'un document (document.edit)
document/{i}	PUT/PATCH	Permet de modifier le document i (document.update)
document/{i}	DELETE	Permet de supprimer un document i (document.destroy)

Administrateur Controller :

Routes	Methode HTTP	Description
admin/	GET	Affiche tous les utilisateurs (admin.index)
admin/create	GET	Permet d'afficher le formulaire pour créer des utilisateurs (admin.create)
admin	POST	Permet d'enregistrer un ou plusieurs utilisateurs en base de données (admin.store)
admin/{i}/edit	GET	Permet d'afficher le formulaire de modifications d'un utilisateur (admin.edit)
admin/{i}	PUT/PATCH	Permet de modifier le utilisateur i (admin.update)
admin/{i}	DELETE	Permet de supprimer un utilisateur i (admin.destroy)

La route **admin/create** permet de créer plusieurs utilisateurs en spécifiant uniquement leurs adresses email, séparées par une virgule ou un espace.

Imprimeur Controller :

Routes	Methode HTTP	Description
professeur/	GET	Affiche tous les professeurs (imprimeur.index)
professeur/	POST	Permet de trier les professeur par nombre d'impression decroissant ou croissant (professeur.choixTri)

Imprimante Controller

Routes	Methode HTTP	Description
imprimante/{i}/edit	GET	Affiche le formulaire de modification des données de l'imprimante (imprimante.edit)
imprimante/{i}	PUT/PATCH	Effectue les modifications de l'imprimante (imprimante.update)

Classe Controller :

Routes	Methode HTTP	Description
classes/	GET	Affiche tous les classes (classes.index)
classes /create	GET	Permet d'afficher le formulaire pour créer des classes (classes.create)
Classes/	POST	Permet d'enregistrer une classe en base de données (classes.store)
classes /{i}/edit	GET	Permet d'afficher le formulaire de modifications d'une classes (classes.edit)
classes /{i}	PUT/PATCH	Permet de modifier la classes i (classes.update)
classes /{i}	DELETE	Permet de supprimer une classes i (classes.destroy)

Middlewares :

Concernant les middlewares j'ai mis en place 2 middlewares .

ImprimanteMiddleware :

```
class imprimanteHS
{
    /**
     * Handle an incoming request.
     *
     * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        if(!Imprimante::find(1)->Disponible)
            session()->put(
                'ImprimanteHS','L\'imprimante est hors service , donc aucune impression ne sera effectué pour le moment');
        else
            session()->forget('ImprimanteHS');
        return $next($request);
    }
}
```

Si l'imprimante est indisponible alors valider l'impression d'un document est impossible

changerNomNewUserMiddleware :

```
public function handle(Request $request, Closure $next): Response
{
    $user = Auth::user();
    if($user != null){
        if(($user->last_name === $user->email || $user->first_name === $user->email) && $request->path() !== 'profile' && $request->path() !== 'logout'){
            session()->put('changerNom', 'Veuillez modifier votre nom et prénom avant de pouvoir utilisé l\'application');
            return redirect('/profile');
        }
        else if(!($user->last_name === $user->email && $user->first_name === $user->email)){
            session()->forget('changerNom');
        }
    }

    return $next($request);
}
```

Lorsqu'un administrateur crée plusieurs utilisateurs simultanément, chaque utilisateur reçoit un email pour réinitialiser son mot de passe. Ce middleware a pour rôle de bloquer l'accès au reste de l'application tant que l'utilisateur n'a pas modifié son nom et son prénom, car, par défaut, ces champs correspondent à l'adresse email.

IV / Les events , leurs listeners et notifications

Event	Listener	Notifications	Description
desactiverEvent	DesactiverListener	DesactiverNotification	Sert à envoyer un mail à tous les utilisateurs dont le compte vient d'être désactivé.
documentPrinterEvent	documentPrinter Listener	documentPrinter Notification	Sert a envoyer un mail à l'utilisateurs si son document a été imprimé
documentUploadEvent	documentUpload Listener	documentUpload Notification	Sert a notifié les imprimeurs que un document a été envoyé.
ImprimanteFeuilleEvent	ImprimanteFeuille Listener	ImprimanteFeuille Notification	Sert a envoyé un mail a l'accueil si le nombre de feuille est inférieur a un nombre n
ImprimanteHSEvent	ImprimanteHS Listener	ImprimanteHS Notification	Sert a envoyer un mail quand l'imprimante est indisponible

Toutes ces notifications implémente ShouldQueue pour effectuer ces taches en arrière-plan

V / Les policies et vues.

Classe Policy :

Policy	Droit
viewAny	Seul un professeur
view	Seul un administrateur
Create	Seul un administrateur ou un professeur
Update	Seul un administrateur ou un professeur propriétaire de la classe
Delete	Seul un administrateur ou un professeur propriétaire de la classe

Document Policy :

Policy	Droit
viewAny	Seul un professeur ou un admin ou un imprimeur
« View » en fonction d'un document	Seul un admin ou imprimeur ou que le document soit encore en attente d'impression et que le professeur soit propriétaire du document
« ViewRefuser » en fonction d'un document	Seul un professeur et que le document soit en statuts : annulé
Create	Seul un administrateur ou un professeur ou un imprimeur
Update	Seul un administrateur ou professeur
Delete	Seul un administrateur
Print en fonction d'un document	Seul un administrateur ou un professeur et que le document soit en attente d'impression
recup	Seul un administrateur ou un professeur ou un imprimeur
Annuler en fonction d'un document	Seul un professeur propriétaire du document et que ce soit dernier soit non imprimé

Imprimante Policy :

Policy	Droit
view	Seul un admin ou imprimeur
update	Seul un admin ou imprimeur

User Policy :

Policy	Droit
View	Seul un administrateur
Create	Seul un administrateur
Update	Seul un administrateur
viewProf	Seul un administrateur ou imprimeur
Delete	Seul un administrateur

Admin view :

Vue	Description
adminAll	Afficher tous les utilisateurs et leurs rôles.
adminEdit	Afficher le formulaire de modification d'un utilisateur
createUsers	Afficher le formulaire de création de plusieurs utilisateurs.

Classe view :

Vue	Description
classeAll	Afficher tous les classes.
classeEdit	Afficher le formulaire de modification d'une classe.
classeCreate	Afficher le formulaire de création de classe.

Document view :

Vue	Description
documentAll	Afficher tous les document.
documentCreate	Afficher le formulaire de création d'un document.
changerDocRefuser	Afficher le formulaire de modification d'un document en cas de refus par l'imprimeur de ce document.
documentShow	Afficher un document {i}
validerImpression	Afficher le formulaire permettant de confirmer l'impression par l'imprimeur

Imprimante view :

Vue	Description
imprimanteEdit	Afficher le formulaire permettant de modifier l'état de l'imprimante ou son nombre de feuilles.

VI / Docker

Création du docker-compose avec plusieurs service (queue , mysql , phpmyadmin , mailcatcher ,php/apache) :

```
version: '3.1'

services:
  app:|
    build:
      context: .
    container_name: ${PHP_CONTAINER_NAME}
    restart: always
    environment:
      USER_ID : ${USER_ID}
      USER_NAME : ${USER_NAME}
    volumes:
      - ./log/apache:/var/log/apache2
      - ../:/var/www/html
    networks:
      - traefik_network
    ports:
      - "8080:80"
    labels:
      - "traefik.enable=true"
      - "traefik.frontend.rule=Host:${HTTP_HOSTNAME}"
      - "traefik.port=80"
      - "traefik.backend=${TRAEFIK_HTTP_BACK_END}"
      - "traefik.frontend.entryPoints=http,https"
```

Création du dockerFile :

```
FROM ubuntu:latest

ARG USER_ID=1001
ARG USER_NAME=user

ENV USER_ID=${USER_ID}
ENV USER_NAME=${USER_NAME}

RUN useradd -m -u ${USER_ID} -s /bin/bash ${USER_NAME}

RUN apt-get update && \
    apt-get install -y software-properties-common && \
    add-apt-repository ppa:ondrej/php && \
    apt-get update && \
    apt-get install -y \
    curl \
    php8.3 \
    php8.3-cli \
    php8.3-fpm \
    php8.3-mbstring \
    php8.3-xml \
    php8.3-mysql \
    php8.3-zip \
    php8.3-curl \
    php8.3-bcmath \
    php8.3-redis \
    apache2 \
    npm \
    git \
    libapache2-mod-php8.3

RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer

RUN a2enmod php8.3
RUN a2enmod rewrite

WORKDIR /var/www/html
```

```

EXPOSE 80

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_PID_FILE /var/run/apache2/apache2.pid

ENV APACHE_DOCUMENT_ROOT /var/www/html/public

#RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
RUN sed -i "s|/var/www/html|${APACHE_DOCUMENT_ROOT}|g" /etc/apache2/sites-available/*.conf
RUN sed -ri -e 's|/var/www|${APACHE_DOCUMENT_ROOT}|g' /etc/apache2/apache2.conf /etc/apache2/conf-available/*.conf
RUN sed -i 's/AllowOverride None/AllowOverride All/' /etc/apache2/apache2.conf

ENV APACHE_RUN_DIR /var/lib/apache/runtime
RUN mkdir -p ${APACHE_RUN_DIR}

CMD apache2 -D FOREGROUND

```

Création d'un script shell permettant la mise en place du container php :

```

#!/bin/bash

#echo "test"

#git pull origin

CHEMIN="../storage"

chmod -R 775 $CHEMIN
chown -R www-data:www-data $CHEMIN

php ../artisan key:generate

php ../artisan migrate:refresh --seed

composer install --no-dev --optimize-autoloader

npm install

npm run production

```

Conclusion :

Mon stage m'a permis d'acquérir une maîtrise approfondie du framework Laravel, tout en découvrant de nouvelles technologies comme Docker. Cette expérience m'a également permis d'explorer le monde professionnel, en apprenant à développer une application conforme aux exigences d'un client. Grâce à ce stage, j'ai renforcé mes compétences techniques tout en développant une meilleure compréhension des besoins clients et des processus de développement en entreprise.