|  |
| --- |
| AFPA |
| Rapport de projet |
| Projet School |

|  |
| --- |
| Mekki Ilyes  Du 22/12/2020 au 10/09/2021 |

Table des matières

[I. Introduction générale 3](#_Toc81917457)

[1. Présentation personnelle 3](#_Toc81917458)

[2. Présentation du projet 4](#_Toc81917459)

[a. Contexte 4](#_Toc81917460)

[b. Objectifs 4](#_Toc81917461)

[II. CONCEPTION DU SYSTEME D’INFORMATION 5](#_Toc81917462)

[3. Diagramme de cas d’utilisation : 5](#_Toc81917463)

[4. Maquettage applicative 7](#_Toc81917464)

[5. Diagrammes 8](#_Toc81917465)

[c. Diagramme de séquence connexion à son espace 8](#_Toc81917466)

[d. Diagramme de séquence création d’un utilisateur 9](#_Toc81917467)

[e. Diagramme de séquence modification d’un utilisateur 10](#_Toc81917468)

[f. Diagramme de séquence suppression d’un utilisateur 11](#_Toc81917469)

[g. Modèle Conceptuel de Données (MCD) 12](#_Toc81917470)

[h. Modèle Logique de Données (MLD) 12](#_Toc81917471)

[i. Modèle Physique de Données (MPD) 13](#_Toc81917472)

[III. ENVIRONNEMENT TECHNIQUES ET OUTILS DE DÉVELOPPEMENT 14](#_Toc81917473)

[IV. DÉVELOPPEMENT DE L’APPLICATION 16](#_Toc81917474)

[V. Conclusion : 36](#_Toc81917475)

# Introduction générale

## Présentation personnelle

Je me présente, Ilyes MEKKI, 26 ans, j’ai intégré l’AFPA en formation du Concepteur Développeur d’application dans le cadre d’une reconversion professionnelle. Suite à l’obtention de mon baccalauréat professionnel comptabilité gestion en 2013,j’ai décidé d’intégrer un BTS assistant de gestion en alternance au sein du CFA IGS mon alternance s’est déroulé au sein de l’ETSUP (l’école du travail social) au poste d’assistant de formation et d’aide comptable, les 2 ans de BTS se sont bien passé ayant validé mon BTS et ne voulant vraiment pas continuer dans la gestion j’ai tenté ma chance dans le domaine de l’informatique qui m’intéresse beaucoup, je me suis inscrit dans plusieurs écoles pour intégrer une formation en alternance mais n’ayant aucune base en informatique les entretiens d’embauche ne se sont pas bien déroulé je me suis donc inscrit à d’autres formation courtes pour apprendre les bases de l’informatique, j’ai donc intégrer une formation de découverte des métiers du numérique au CNAM, par la suite j’ai intégré un bac +1 dans le développement web au CNAM qui n’a pas pu être validé à cause du COVID, par la suite le pôle emploi m’a proposé la formation CDA une formation qui me permettra d’acquérir de nouvelles compétences dans le domaine du développement.

## Présentation du projet

### Contexte

N’ayant pas trouvé de stage pour validé la formation j’ai dû travaillé sur un projet personnel sur le quel travaillé, après réflexion j’ai décidé de faire une plateforme web de **gestion d’un collège**.

### Objectifs

Mettre en place une application web permettant de gérer les utilisateurs, nouveaux élèves, contrôles et un CDI.

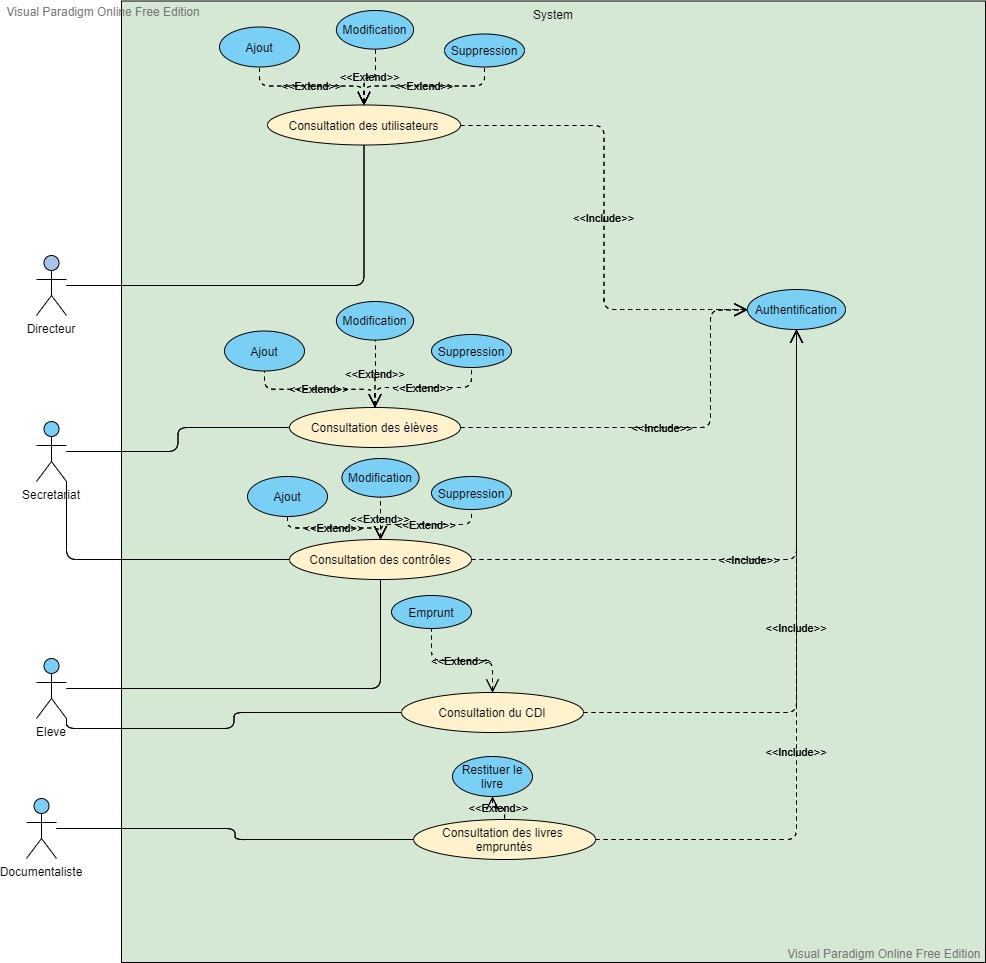
L’application permettra d’assurer les objectifs suivants :

* Création et modification des utilisateurs
* Création et modification et affectation des élèves.
* Ajout de contrôles par classe et attribution des notes par élèves
* Emprunt de livre au CDI et gestion des livres empruntés

# CONCEPTION DU SYSTEME D’INFORMATION

## Diagramme de cas d’utilisation :

La figure suivante désigne le diagramme de cas d'utilisation global de l’application dans lequel nous allons mettre en évidence les services offerts par celle-ci.



**Explication de diagramme de cas d’utilisation :**

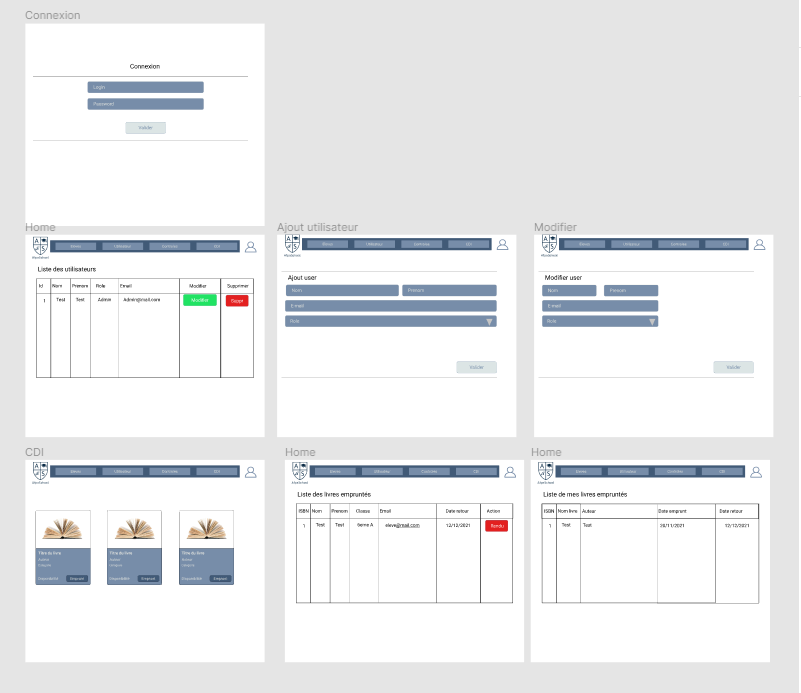
|  |  |
| --- | --- |
| **Diagrammes** | **Explications** |
| Diagramme de cas d’utilisation | Ce diagramme de cas d’utilisation présente 2 types d’acteurs : visiteur et administrateur :  **Administrateur :**  S’authentifier : Saisir ses identifiants.  Gérer son compte.  Ajout/modification/suppression des utilisateurs/ élèves/contrôle.  **Secrétariat :**  S’authentifier : Saisir ses identifiants.  Gérer son compte.  Ajout/modification/suppression des élèves/contrôle.  **Elève :**  S’authentifier : Saisir ses identifiants.  Gérer son compte.  Consultation des élèves/contrôle.  **Documentaliste :**  S’authentifier : Saisir ses identifiants.  Gérer son compte.  Consultation/modification des livres empruntés. |

***Explication de diagramme de cas d’utilisation global.***

## Maquettage applicative

La maquette de l’application a été réalisé grâce au wireframe Figma

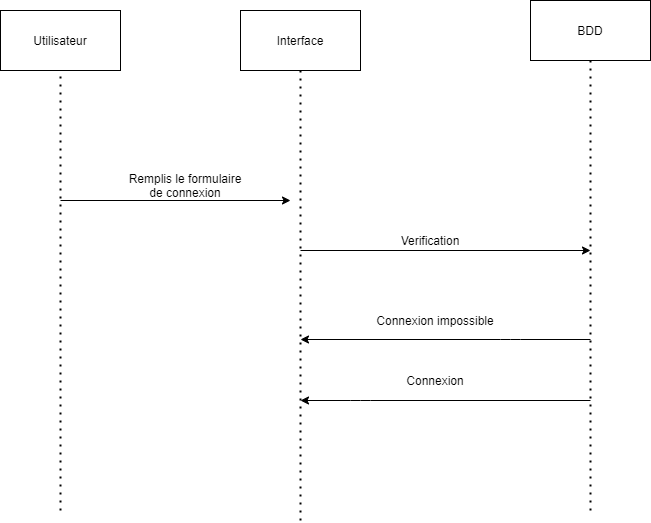
**Maquette de l’application**



## Diagrammes

Les diagrammes de séquence permettent d’exprimer la séquence des interactions entre l’utilisateur et l’application

### Diagramme de séquence connexion à son espace

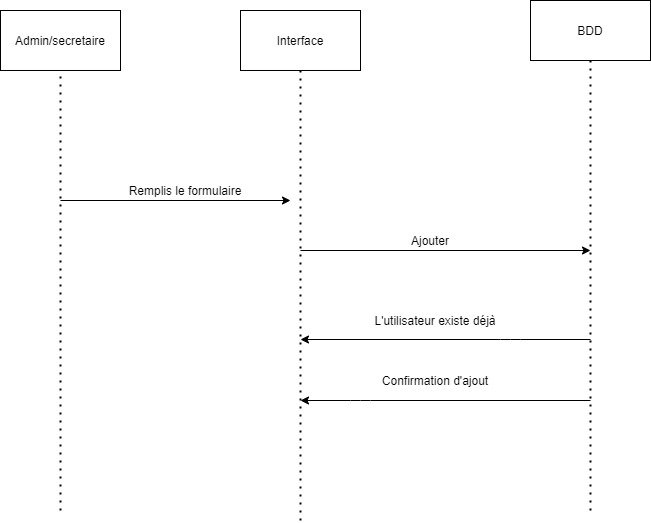


***Diagramme de séquence connexion***

**Explication du diagramme de séquence :**

|  |  |
| --- | --- |
| **Diagramme** | **Explications** |
| Diagramme de séquence pour tous les utilisateur | **L’Administrateur :**  L’utilisateur entre son login et son mot de passe.  Envoi des informations à la BDD : compare les données envoyées par l’utilisateur et les compare aux données existante.  Si l’authentification réussie l’utilisateur est redirigé vers la page d’accueil.  Si l’authentification échoue l’utilisateur doit réessayer. |

### Diagramme de séquence création d’un utilisateur



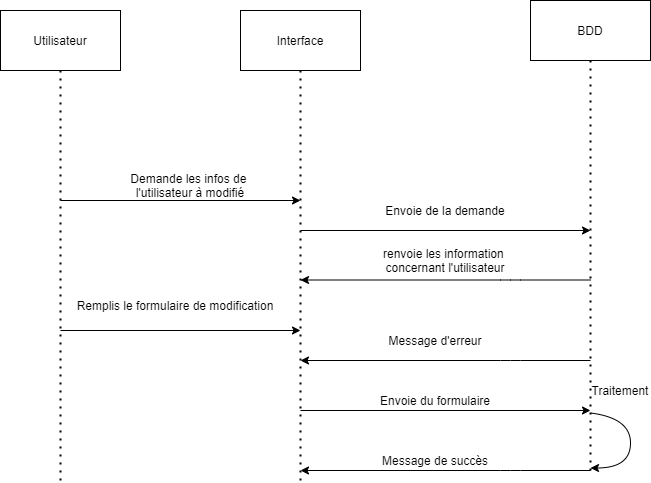
***Diagramme de* séquence création d’un utilisateur**

**Explication du diagramme de séquence :**

|  |  |
| --- | --- |
| **Diagramme** | **Explications** |
| Diagramme de séquence | - Vérification du formulaire : si le formulaire est vide une erreur est renvoyée.  - Envoi des informations du formulaire à la BDD.  - Requête de vérification si l’adresse mail existe déjà dans la BDD.  - Afficher un message d'erreur si l’adresse mail existe déjà.  - Requête de création d'un utilisateur.  - Afficher un message de succès. |

**Remarque : le système fonctionne de la même manière pour ajouter une note ou un élève.**

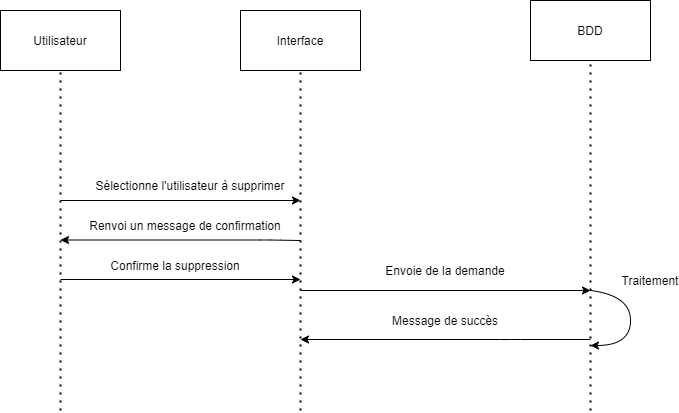
### Diagramme de séquence modification d’un utilisateur



**Explication de diagramme de séquence :**

|  |  |
| --- | --- |
| **Diagramme** | **Explications** |
| Diagramme de séquence | -Lorsque l’on clique sur le bouton ”Modifier” on récupère les informations de l’utilisateur dans un formulaire pré-remplis.  - Modification des informations dans le formulaire.  - Vérification du formulaire : si le formulaire est vide une erreur est renvoyée.  - Envoi des informations du formulaire à la BDD.  - Requête de vérification si l’adresse mail existe déjà dans la BDD.  - Afficher un message d'erreur si l’adresse mail existe déjà.  - Requête de modification de l’utilisateur.  - Afficher un message de succès. |

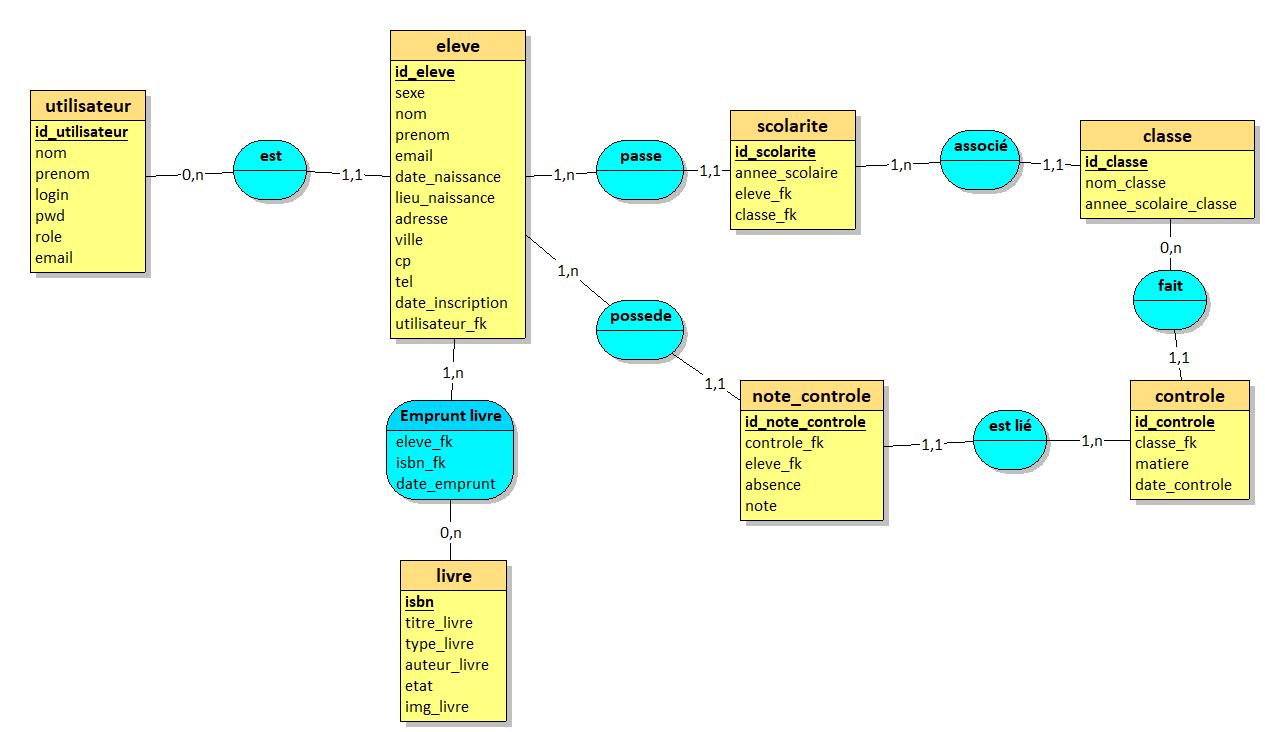
### Diagramme de séquence suppression d’un utilisateur



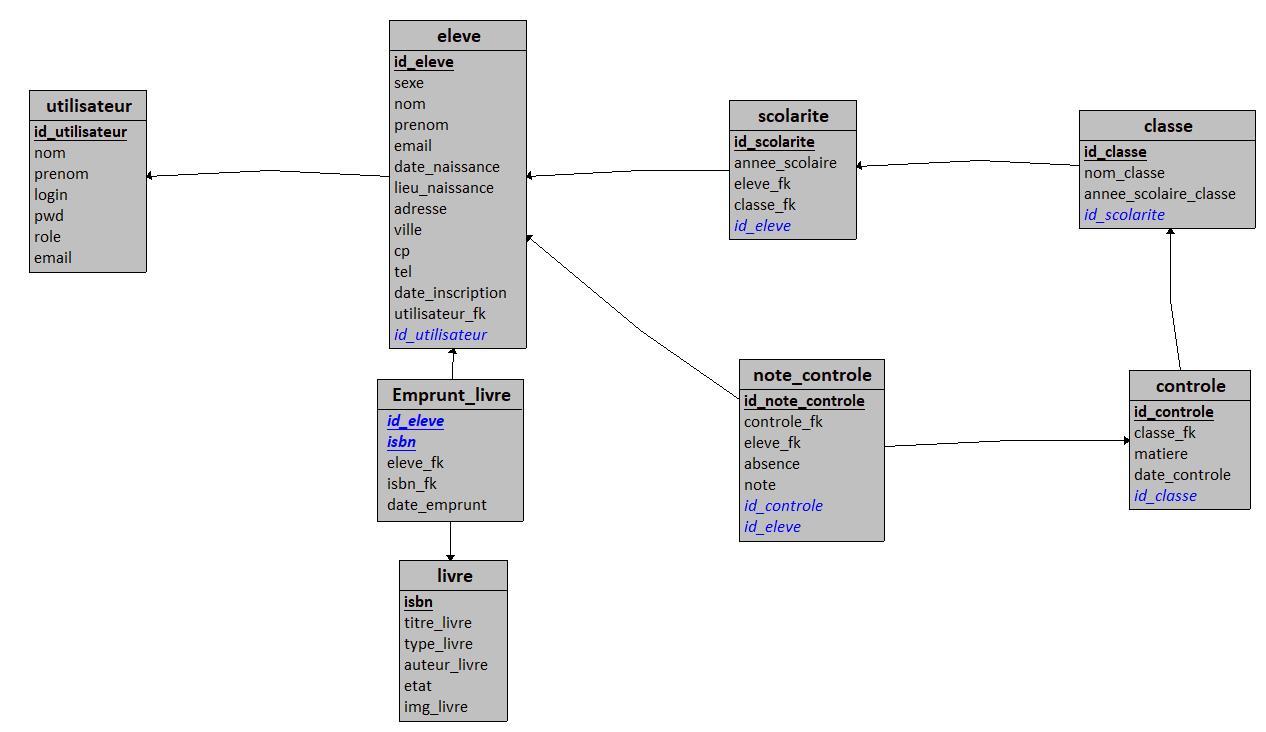
**Explication de diagramme de séquence :**

|  |  |
| --- | --- |
| **Diagramme** | **Explications** |
| Diagramme de séquence | -Lorsque l’on clique sur le bouton ”Supprimer” on reçoit un message de validation pour la suppression de l’utilisateur.  - Requête de suppression de l’utilisateur.  - Afficher un message de succès. |

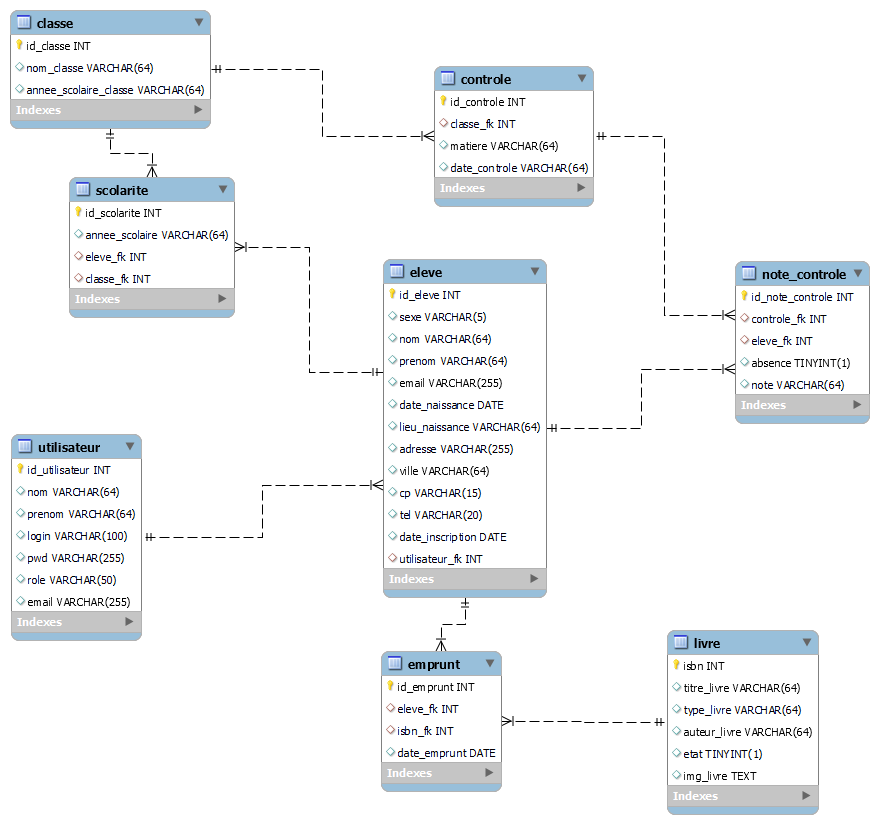
### Modèle Conceptuel de Données (MCD)



### Modèle Logique de Données (MLD)



### Modèle Physique de Données (MPD)



# ENVIRONNEMENT TECHNIQUES ET OUTILS DE DÉVELOPPEMENT

#### Environnement de développement :

##### 1. Environnement matériel :

* Ordinateur dont les caractéristiques sont comme suite :
* Marque : Lenovo
* Mémoire : 16gb
* Système d’exploitation : Windows 10

##### 2. Environnement logiciel :

Les logiciels utilisés pour cette application sont :

* Looping : Création des diagrammes MCD/MLD
* Git : Versionning du code
* Figma pour la maquette.
* Draw.io pour les différents diagrammes.
* Google chrome : permet d’afficher les pages web de l’application
* MySQL Workbench : Création du script SQL et MPD.
* Visual Studio Code : Environnement de développement
* Microsoft Word : pour la rédaction du rapport.
* Xampp est une plateforme de développement Web, comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases de données MySQL. Permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP.

##### 3. Langages :

1. Front end

* HTML5 (HyperText Markup Language 5) permet l’affichage du contenu du site sur le navigateur.
* CSS3 (*Cascading Style Sheets*) permet de donner du style à nos page HTML
* Java Script permet l’interactivité des pages web

1. Back end

* PHP (HyperText Preprocessor) permet de créer des page web dynamique
* SQL (Structured Query Language) permet de manipuler la base de données

1. Framework

* Bootstrap 4 : Framework CSS qui met à disposition des outils de design pour aider au développement.
* jQuery : Framework Java script fourni avec Bootstrap.

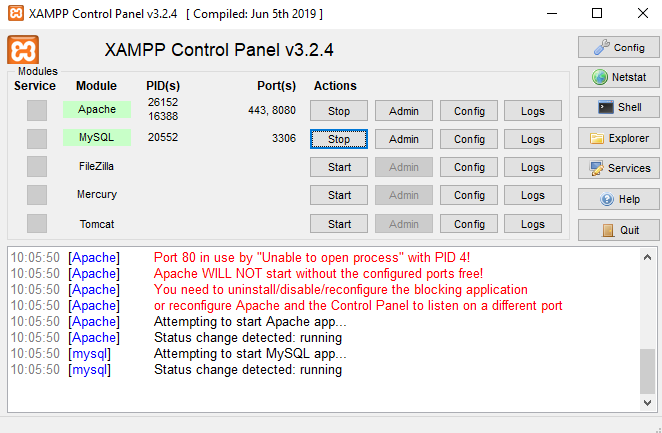
# DÉVELOPPEMENT DE L’APPLICATION

.

#### 1. Importation de la base de données :

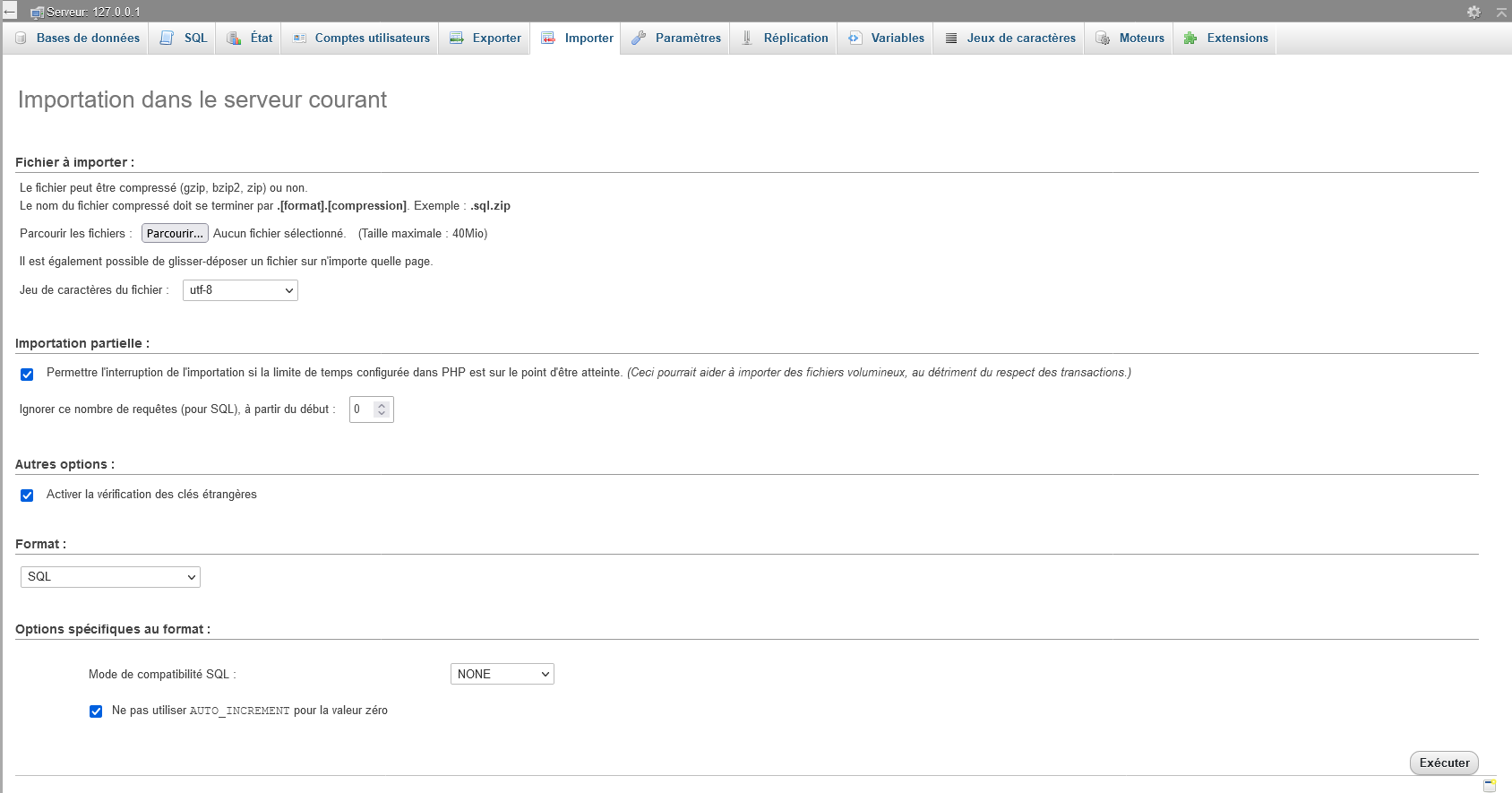
On commence par lancer Xampp et depuis l’interface on lance les services Apache et MySQL

**Panel de contrôle Xampp**

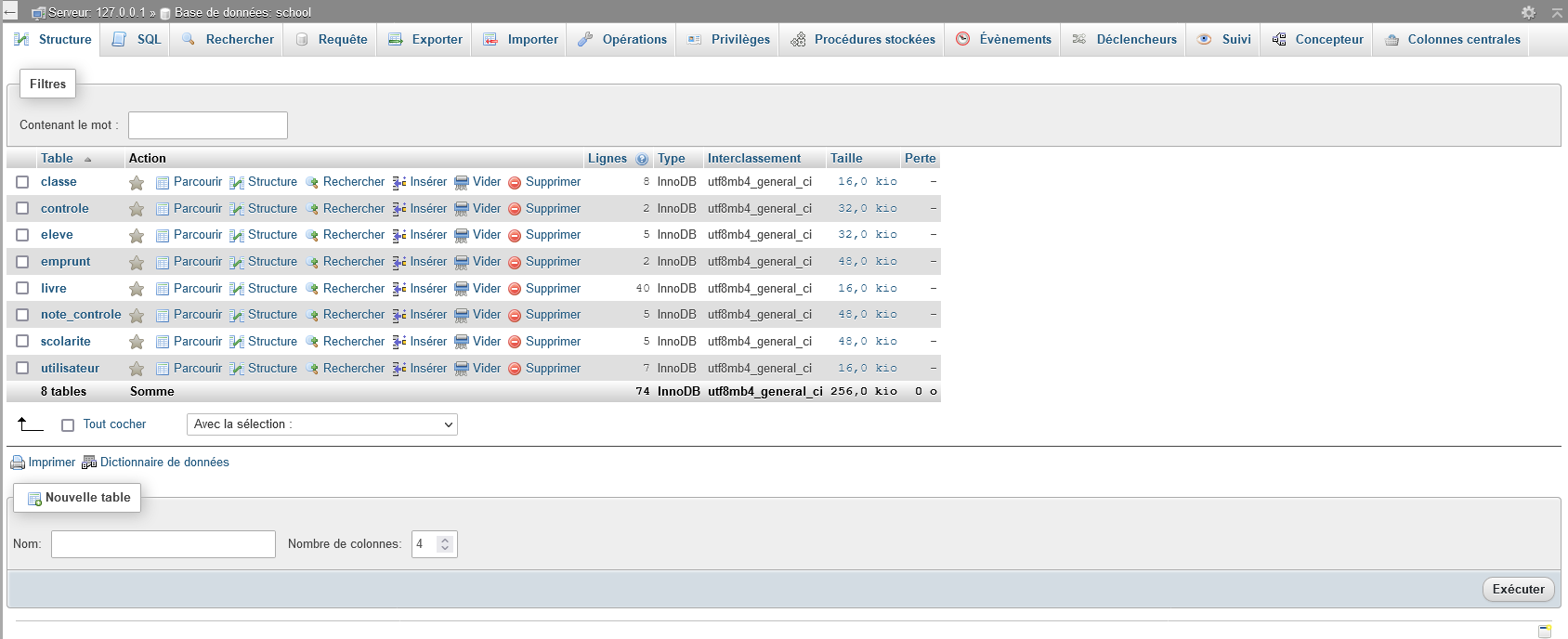


Depuis notre navigateur on tape dans la barre de recherche : <http://localhost:8080/phpmyadmin>, on retrouve une interface de gestion de base de données, on clique sur l’onglet importer et on importe notre script SQL.

**Page d’import phpmyadmin**



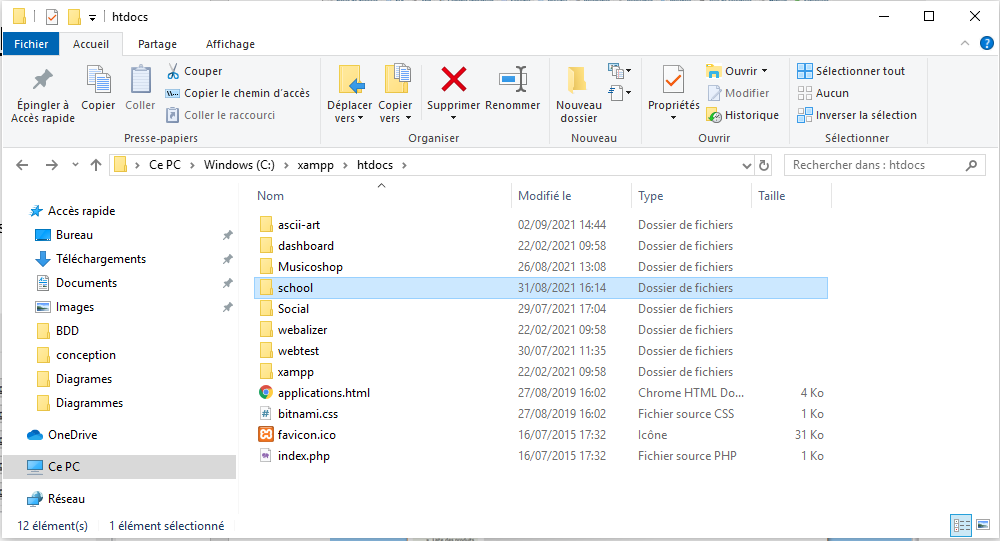
**Vue d’ensemble de la BDD**



#### 2. Préparation de l’environnement de travail :

Il faut placer l’ensemble du projet dans le dossier htdocs de xampp.

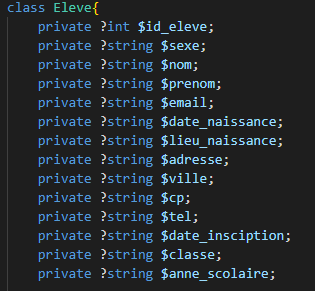
**Contenu du dossier htdocs**



#### 3. Création des model :

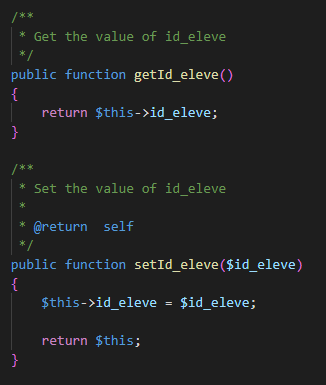
Ayant choisi le pattern MVC Model Vue Contrôleur, il faut commencer par la création des classes qui ont des attributs et des méthodes.

**Attributs de la classe élève**



Pour chaque attribut il faut créer les méthodes Getter (Récupérer la valeur de l’attribut) et Setter (Attribuer une valeur a un attribut)

**Méthodes getter/setter de l’id de l’élève**



#### Description des étapes :

1. Connexion

Chaque utilisateur doit se connecter pour accéder à l’application, les comptes sont créés par le directeur ou le secrétariat.

***Interface de connexion***

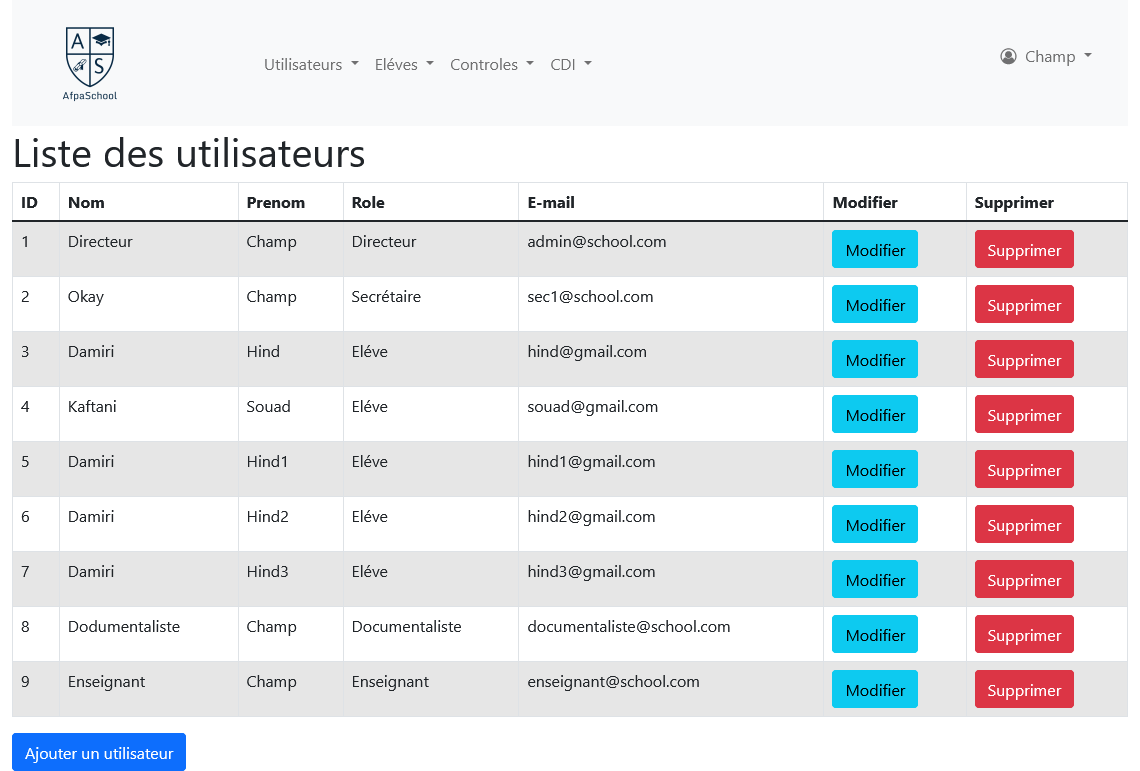


Ici lors de l’exécution du script de connexion nous allons tester plusieurs étapes :   
1- Vérifier si tous les champs sont remplis sinon renvoie une erreur   
2- Hacher le mot de passe qui se trouve dans le formulaire  
3- Vérifier si le mail et le mot de passe match avec les données existante via une requête SQL sinon renvoie une erreur  
4- Attribuer les données reçu dans la requête en session qui nous serons utiles durant la navigation dans l’application : comme l’id de l’utilisateur, son rôle, son nom et son id d’élève si c’est un élève qui se connecte.  
5- On redirige l’utilisateur vers une page d’accueil.

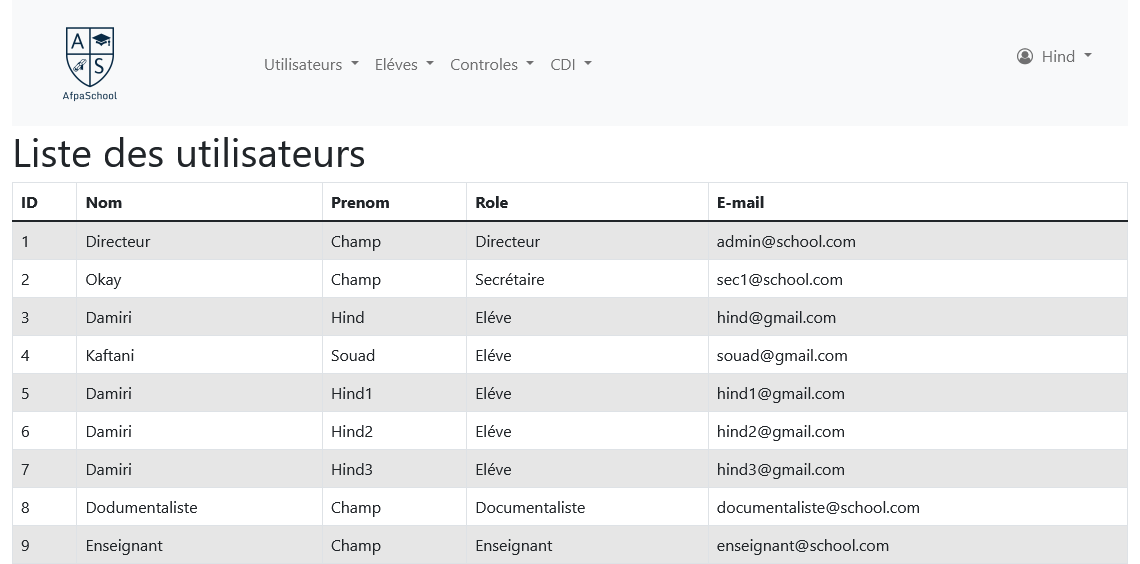
1. Utilisateurs

**b.1. Liste des utilisateurs**

**Interface de la liste des utilisateurs(direction/secrétariat)**



**Interface de la liste des utilisateurs(Elèves/Enseignant/Documentaliste)**

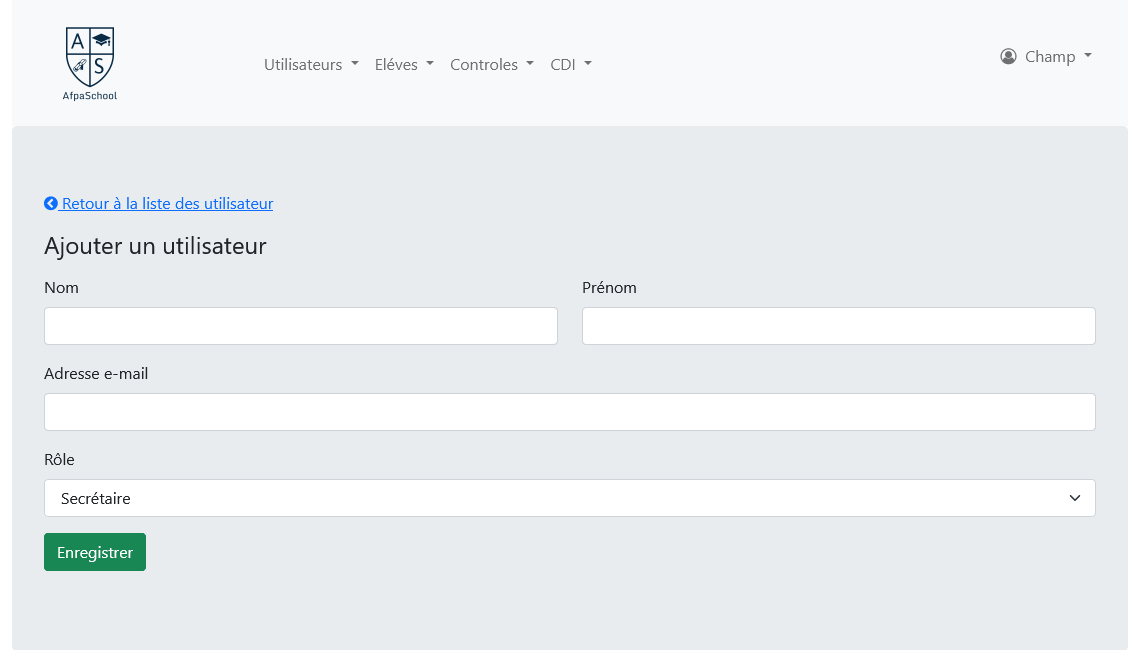


Pour sécuriser l’application certaines fonctionnalité ne sont pas visibles pour tous les utilisateurs. Si un utilisateur essaye d’accéder à une page à laquelle il n’est pas autoriser il est automatiquement redirigé sur une autre page.

**b.2. Ajout d’utilisateur**

Pour ajouter un utilisateur, l’administrateur/secrétariat doit remplir le formulaire d'ajout.

**Interface du formulaire d’ajout d’utilisateur**



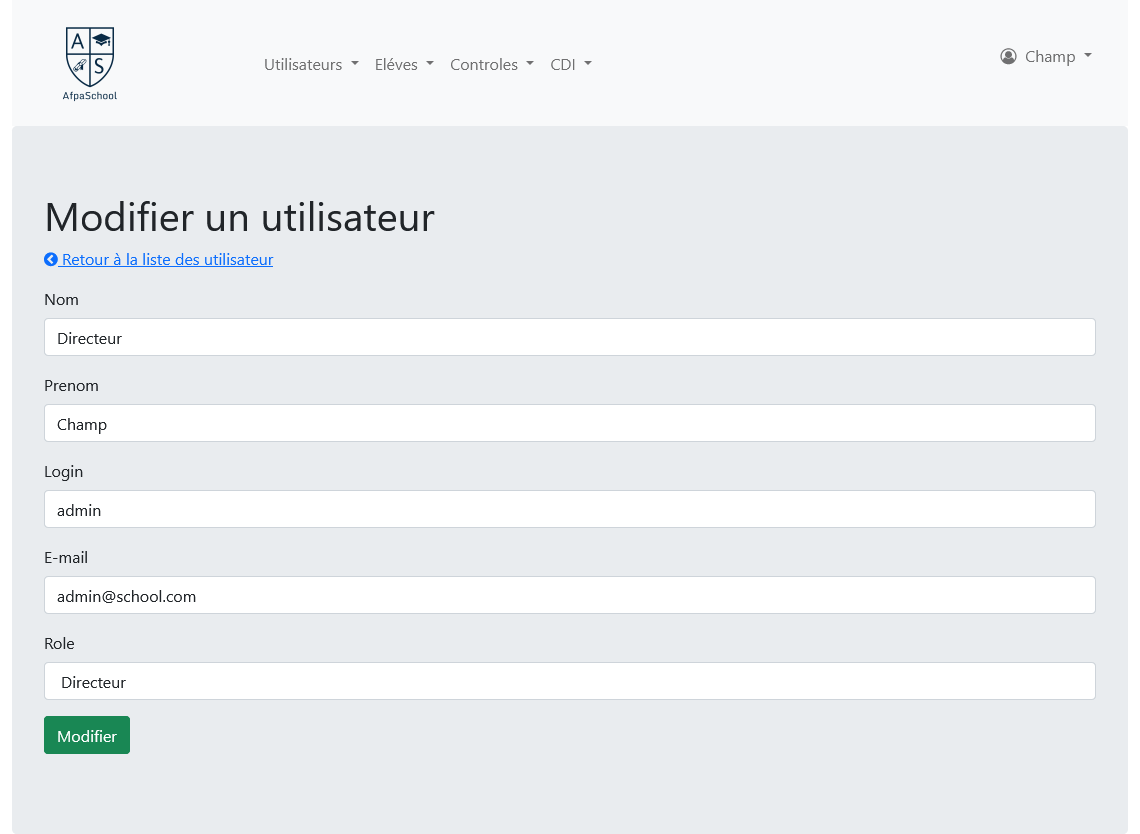
Lors de la soumission du formulaire le contrôleur vérifie :

1. Que tous les champs sont bien remplis (renvoie un message d’erreur)
2. Que l’adresse e-mail ne soit pas dupliqué (renvoie un message d’erreur)

Le programme va générer le login à partir de la 1ere partie de l’adresse e-mail et un mot de passe qui fusionne le nom et prénom de l’utilisateur.

**b.3. Modification d’un utilisateur**

Pour modifier un utilisateur le script remplis le formulaire avec les données récupérer en base.



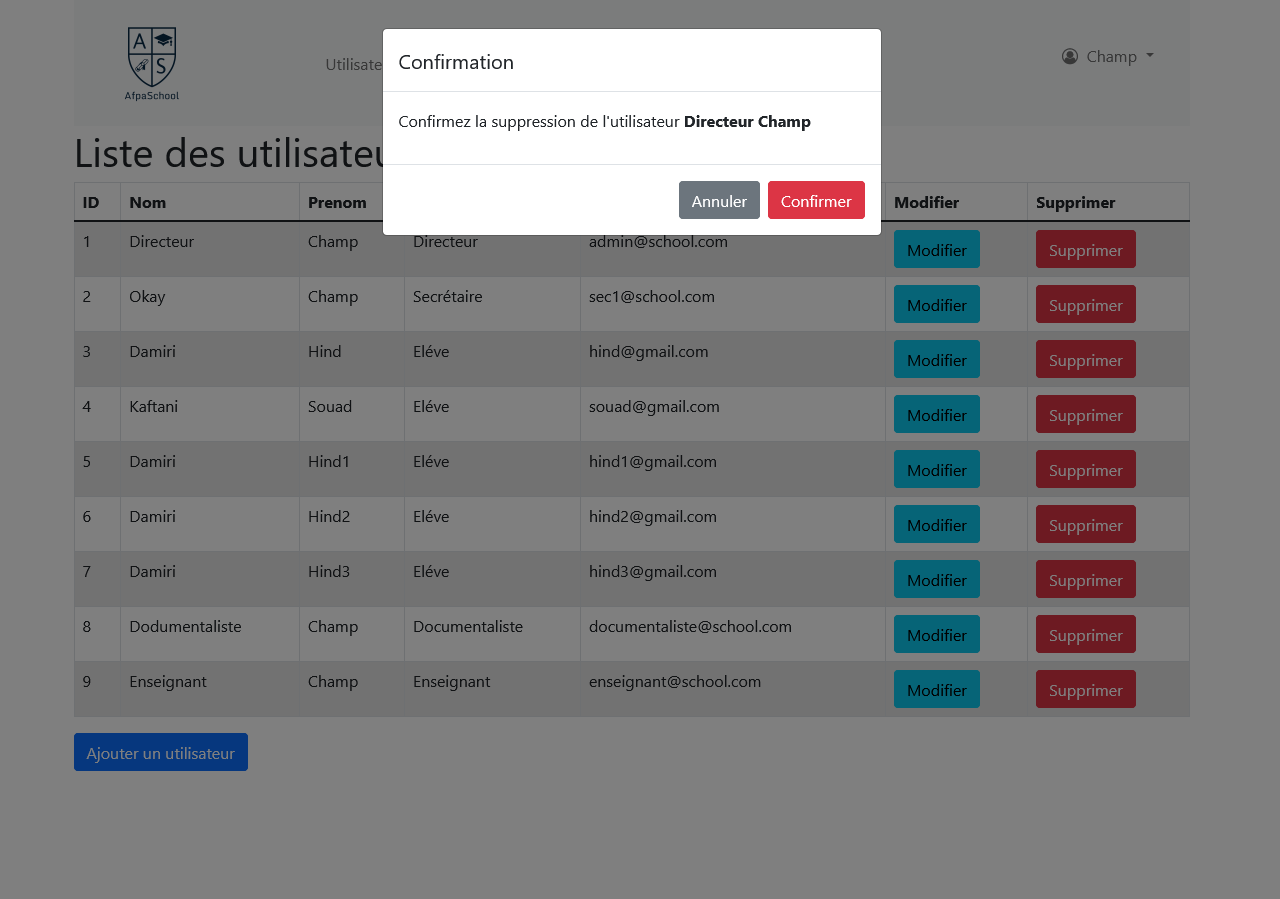
**Interface du formulaire modification d’un utilisateur**

Lors de la soumission du formulaire le contrôleur vérifie :

1. Que tous les champs sont bien remplis (renvoie un message d’erreur)
2. Que l’adresse e-mail ne soit pas dupliqué (renvoie un message d’erreur)

**b.3. Suppression d’un utilisateur**

Pour supprimer un utilisateur il faut confirmer la suppression.

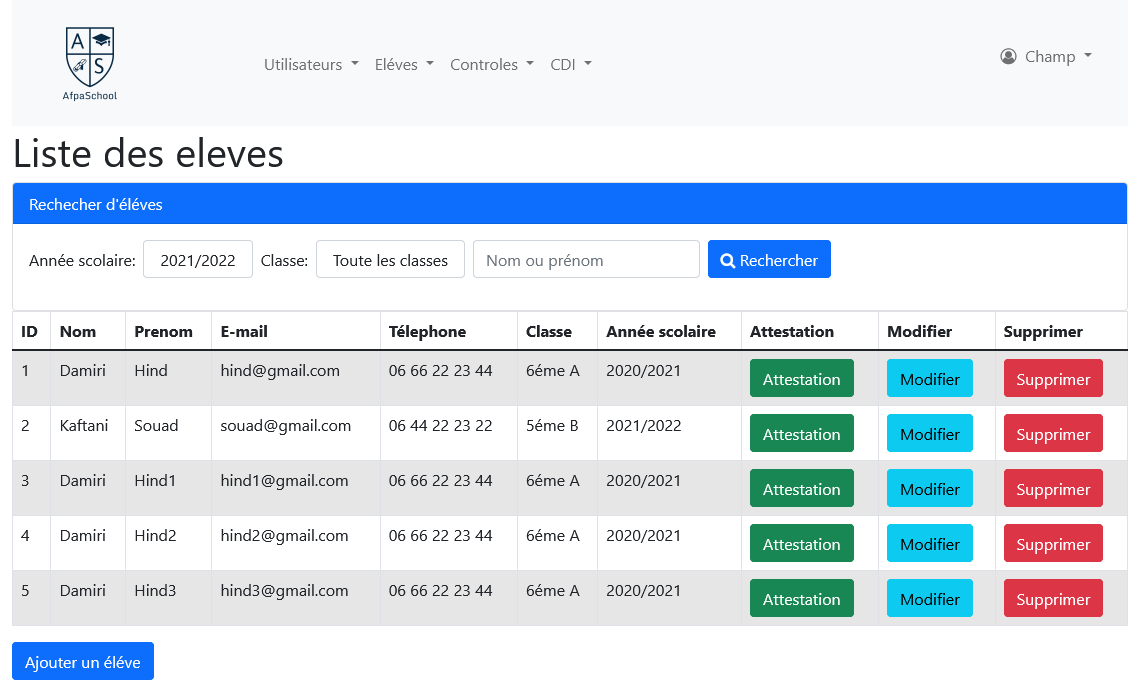


**Interface de la suppression d’un utilisateur**

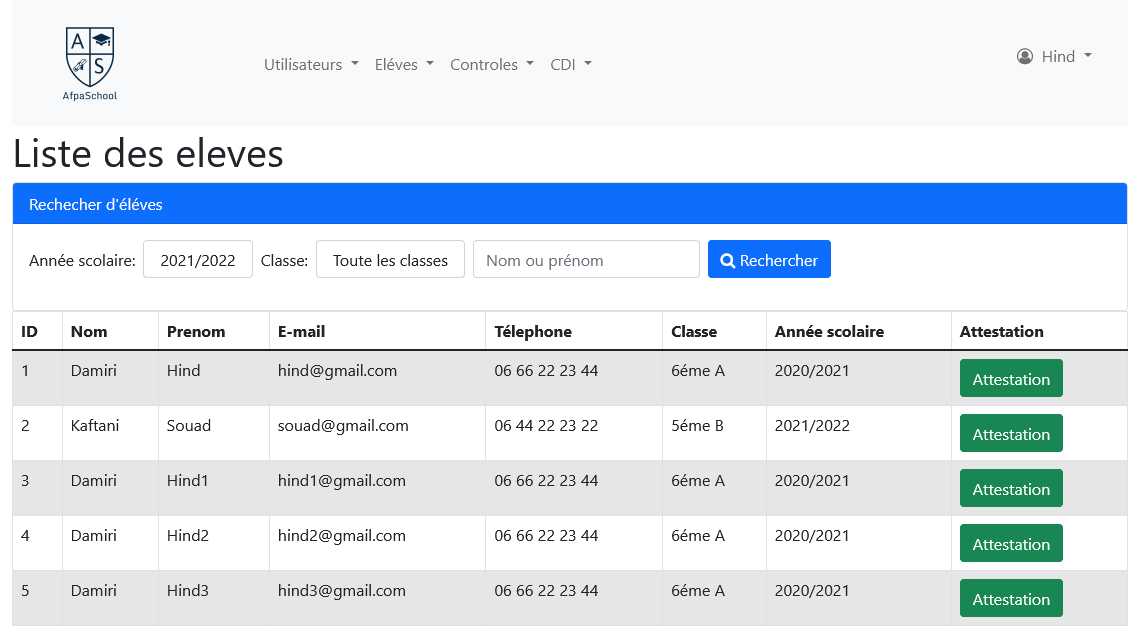
Le contrôleur va vérifier si l’utilisateur à supprimer est l’administrateur et renvoyer un message d’erreur.

1. Elèves

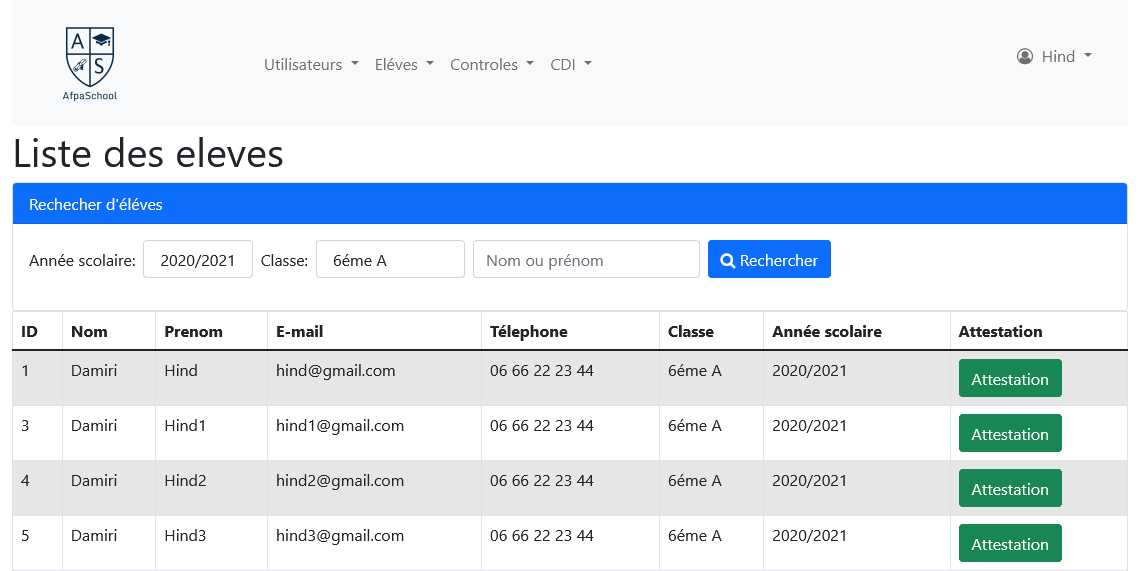
**c.1. Liste des élèves**



**Interface de la liste des élèves(direction/secrétariat)**



**Interface de la liste des élèves (Elèves/Enseignant/Documentaliste)**



**Interface de la recherche d’élèves**

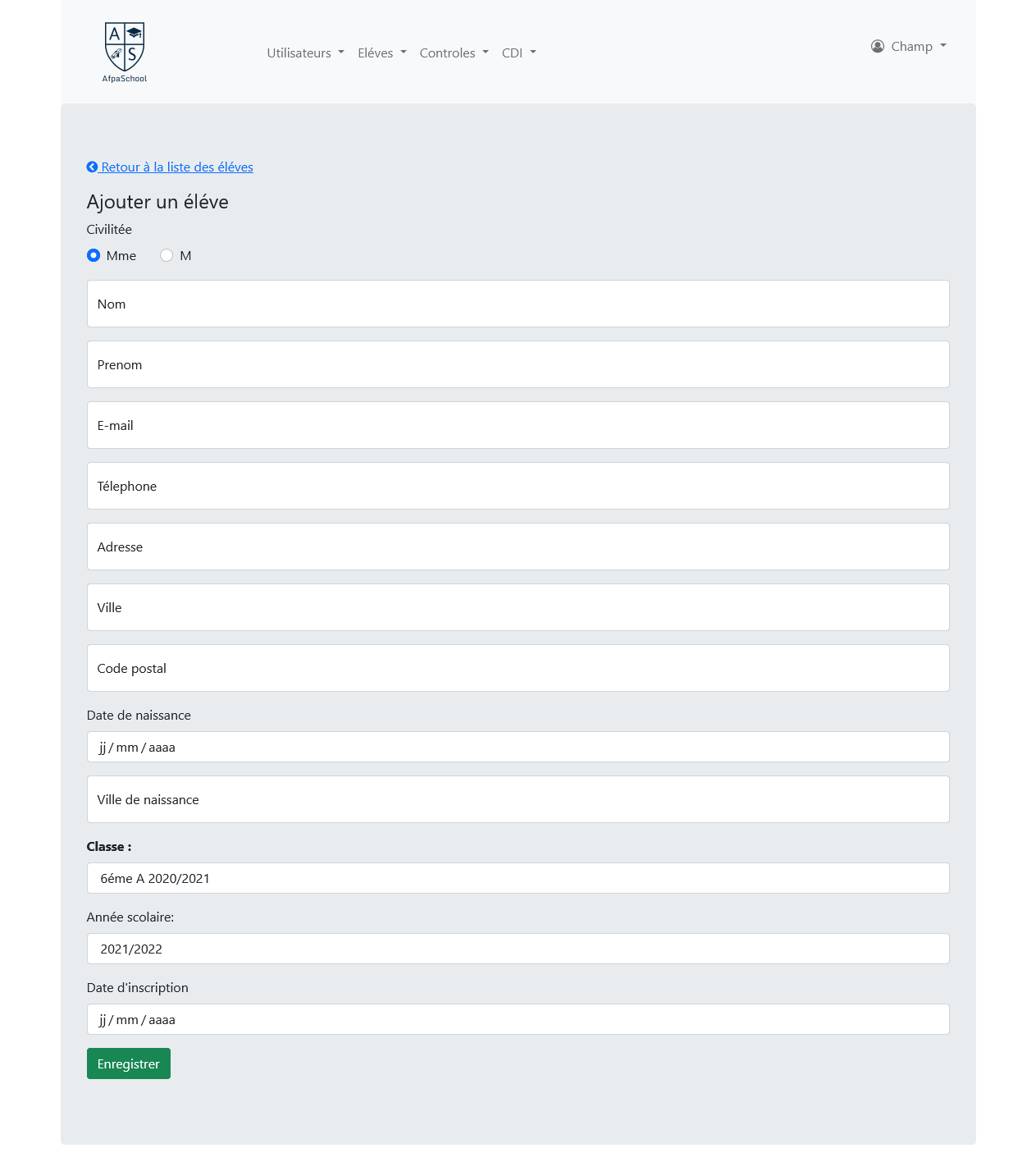
**c.2. Attestation de scolarité**



**Attestation de scolarité élève**

Les attestations sont générées automatiquement grâce à la librairie FPDF qui permet de créer des fichiers PDF en PHP.

**c.3. Ajout d’élève**



**Interface du formulaire d’ajout d’élève**

Lors de la soumission du formulaire le contrôleur vérifie :

1. Que tous les champs sont bien remplis (renvoie un message d’erreur)
2. Que l’adresse e-mail ne soit pas dupliqué (renvoie un message d’erreur)

Le programme va générer un utilisateur avec le nom, prénom, e-mail de l’élève et comme pour les utilisateurs le login et le mot de passe sont généré automatiquement.

**c.4. Modification de l’élève**

Pour modifier un élève le script remplis le formulaire avec les données récupérer en base.



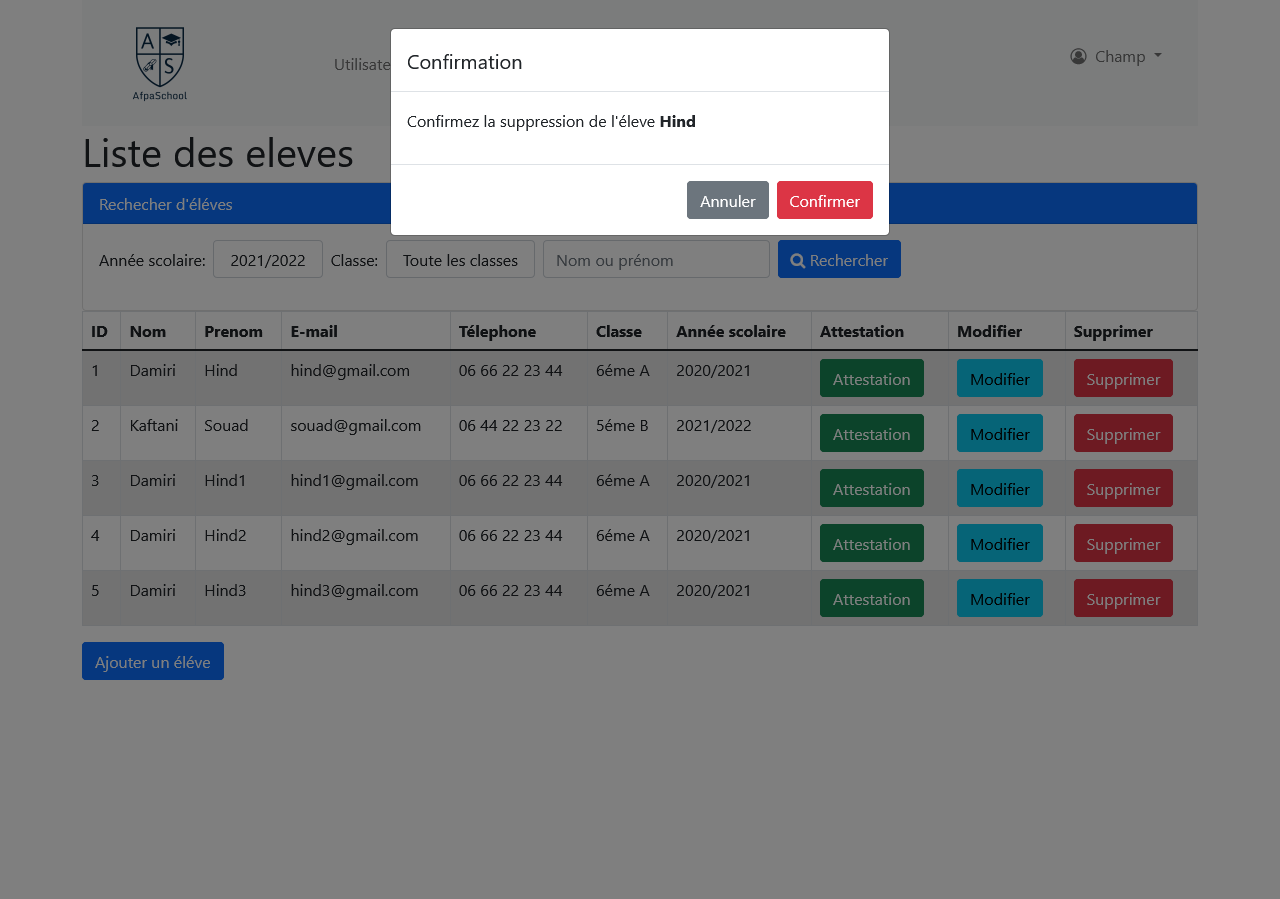
**Interface du formulaire modification de l’élève**

Lors de la soumission du formulaire le contrôleur vérifie :

1. Que tous les champs sont bien remplis (renvoie un message d’erreur)
2. Que l’adresse e-mail ne soit pas dupliqué (renvoie un message d’erreur)

**c.4. Suppression d’un élève**

Pour supprimer un utilisateur il faut confirmer la suppression.

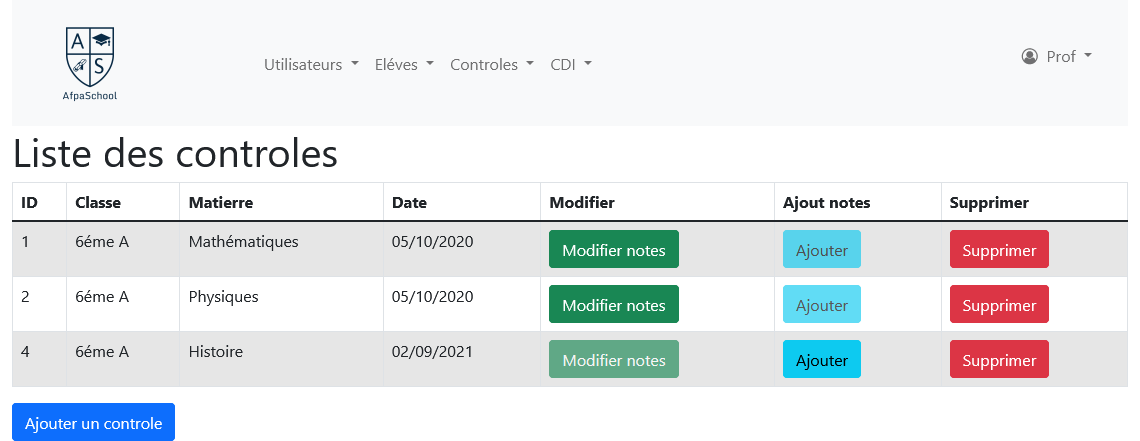


**Interface suppression d’élève**

La suppression de l’élève entraine la suppression de l’utilisateur.

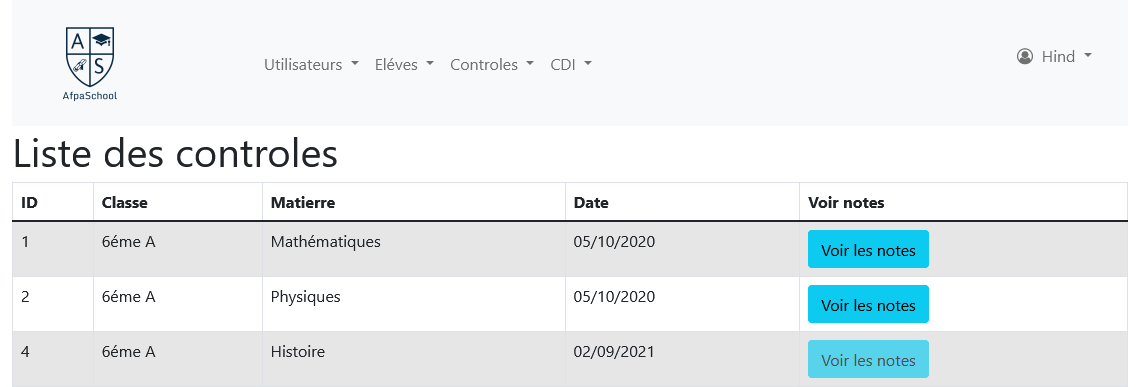
1. Contrôles

**d.1. Liste des contrôles**



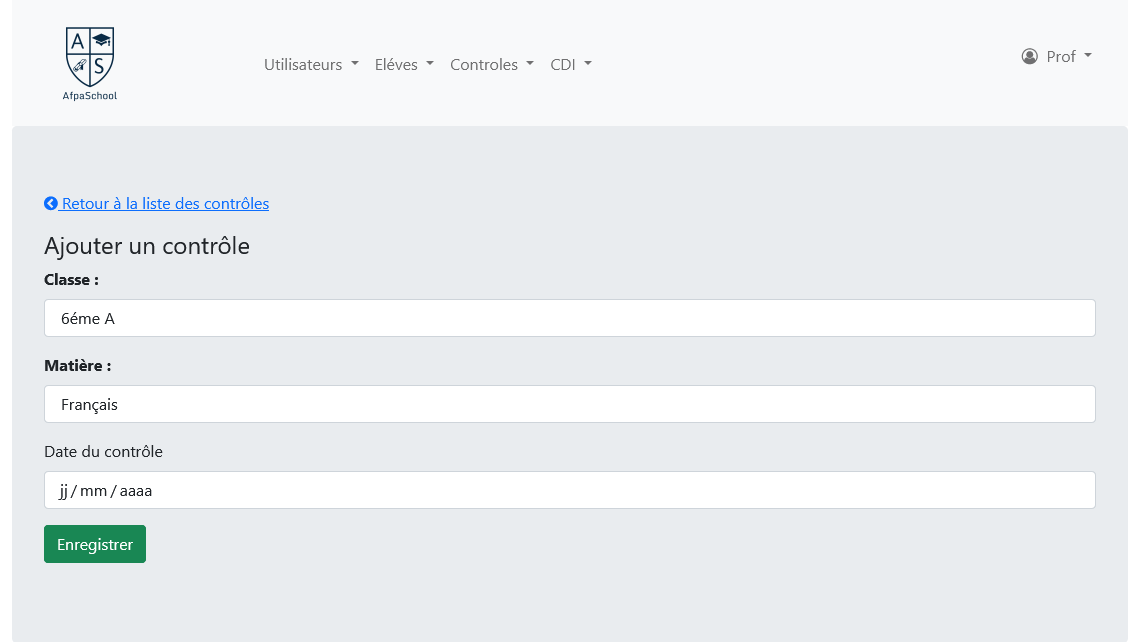
**Interface de la liste des contrôles (direction/secrétariat/enseignant)**

Quand un contrôle n’est pas noté on ne peut pas modifier les notes, il faut ajouter les notes avant de pouvoir modifier.



**Interface de la liste des contrôles (élèves)**

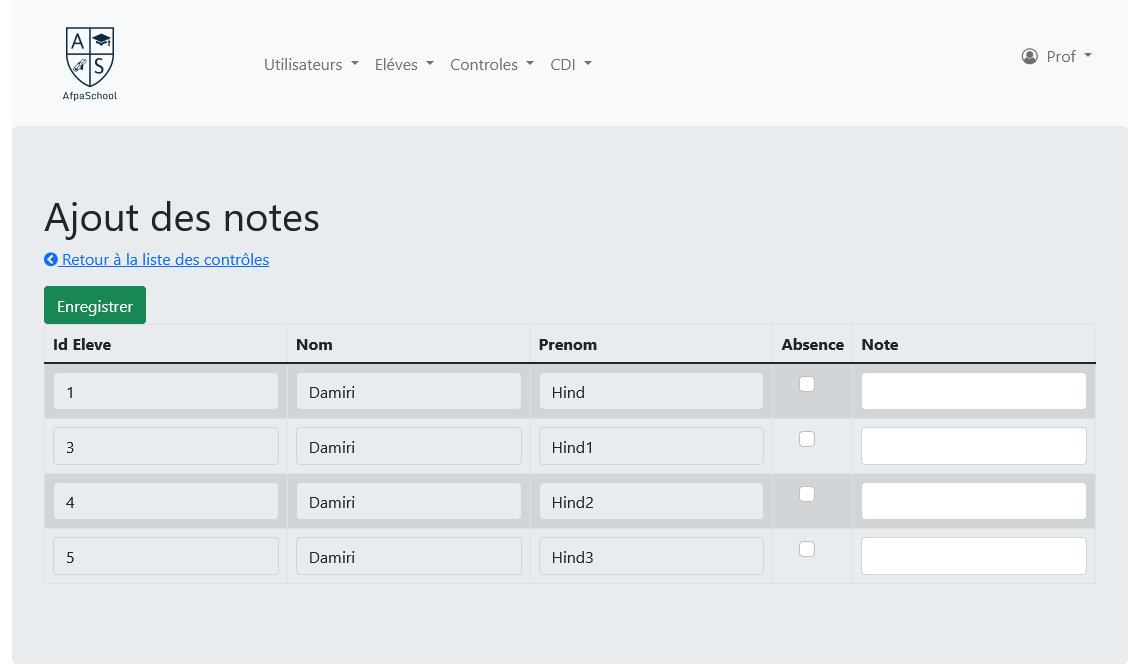
**d.2. Ajout de contrôles**



**Interface ajout contrôles**

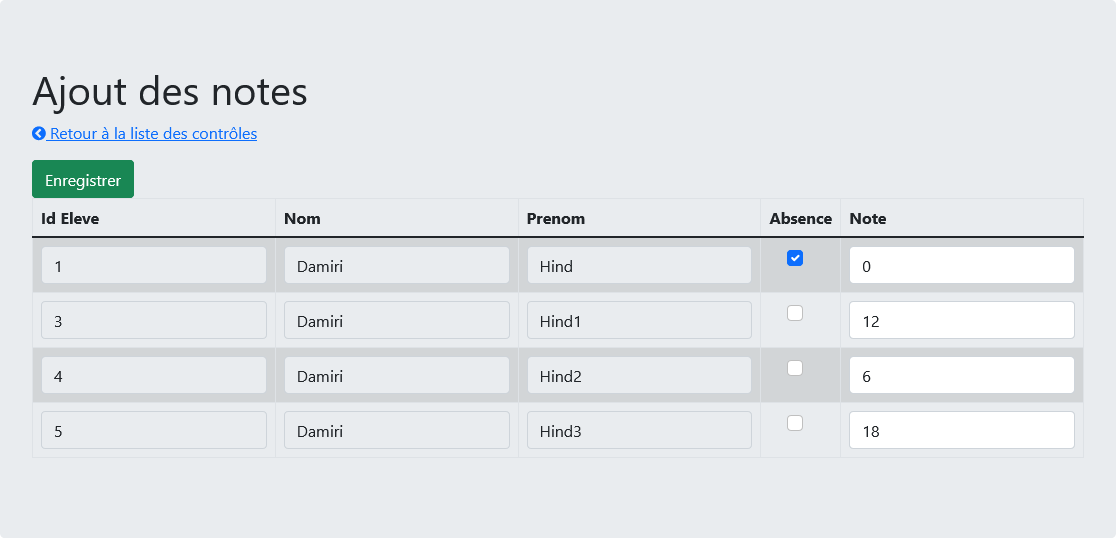
Pour ajouter un contrôle le professeur remplis le formulaire pour créer le contrôle.

**d.3. Ajout notes contrôle**



**Interface ajout notes contrôles**

Pour ajouter les notes le programme génère la liste des élèves

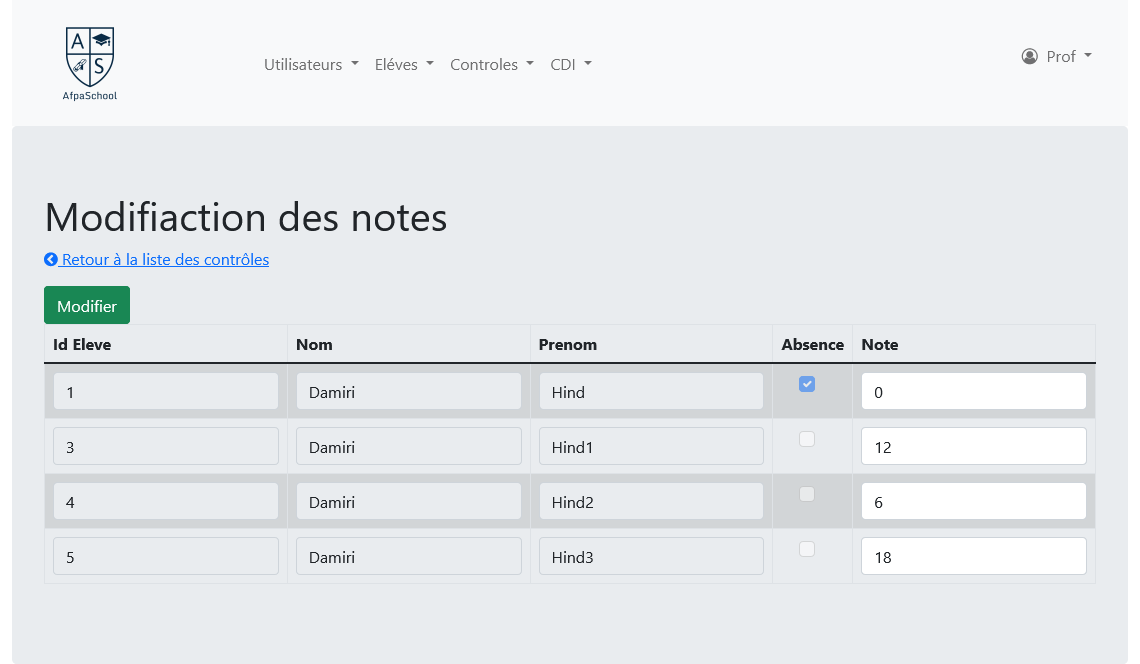


**Interface ajout notes contrôles remplis**L’enseignant remplis les cases et envoi le formulaire.

Le programme renvoi une erreur :

* Si le formulaire est vide
* Si une note est supérieure à 20

**d.4. Modification de notes**

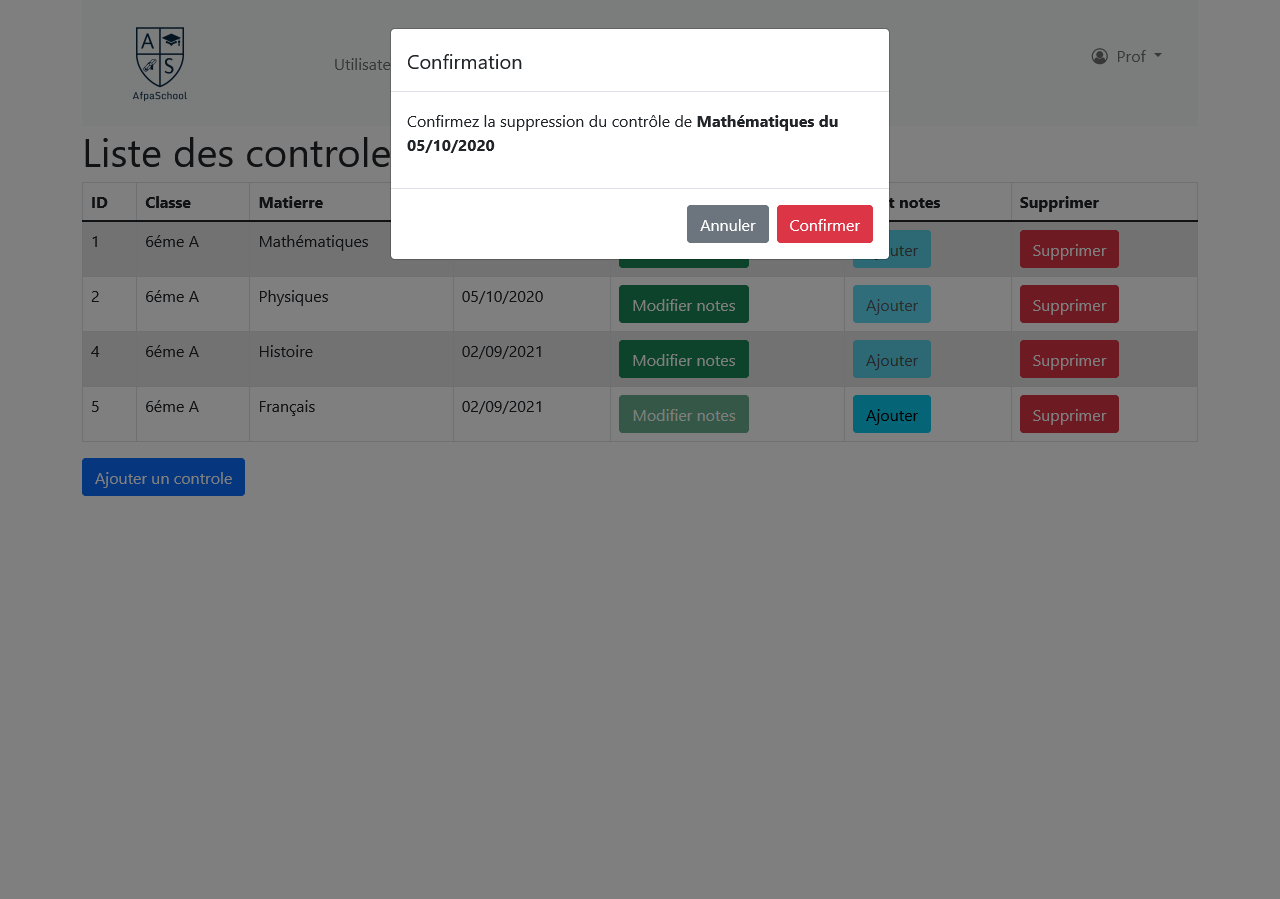


**Interface modification des notes**

Le programme renvoi une erreur :

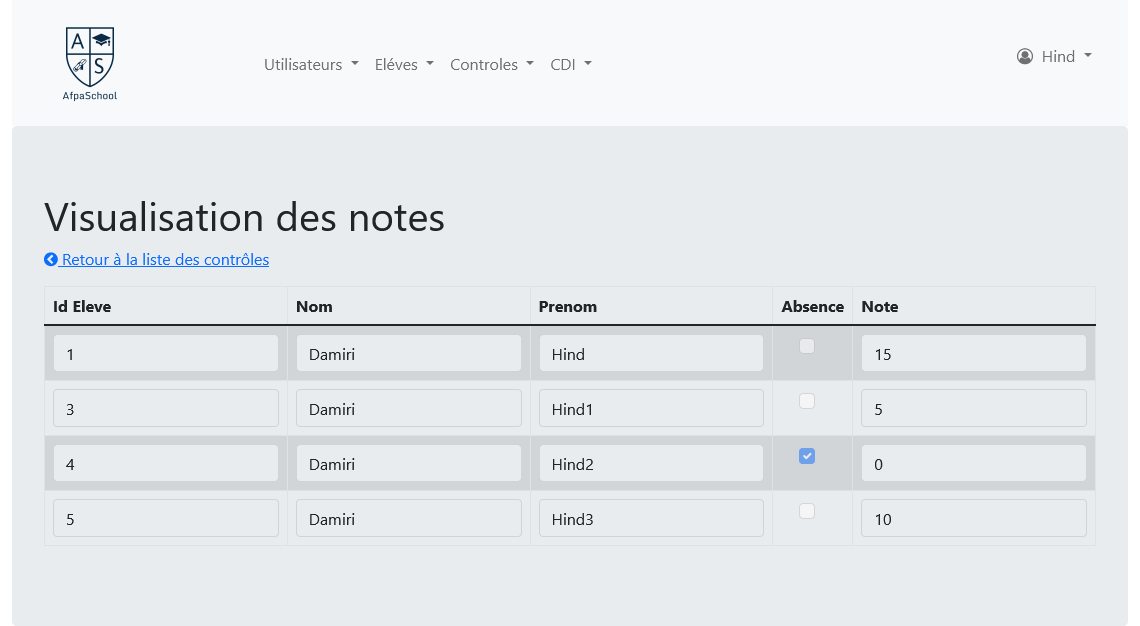
* Si une note est supérieure à 20

**d.5. Suppression de contrôle**



**Interface suppression contrôles**

**d.6. Voir les notes contrôles**

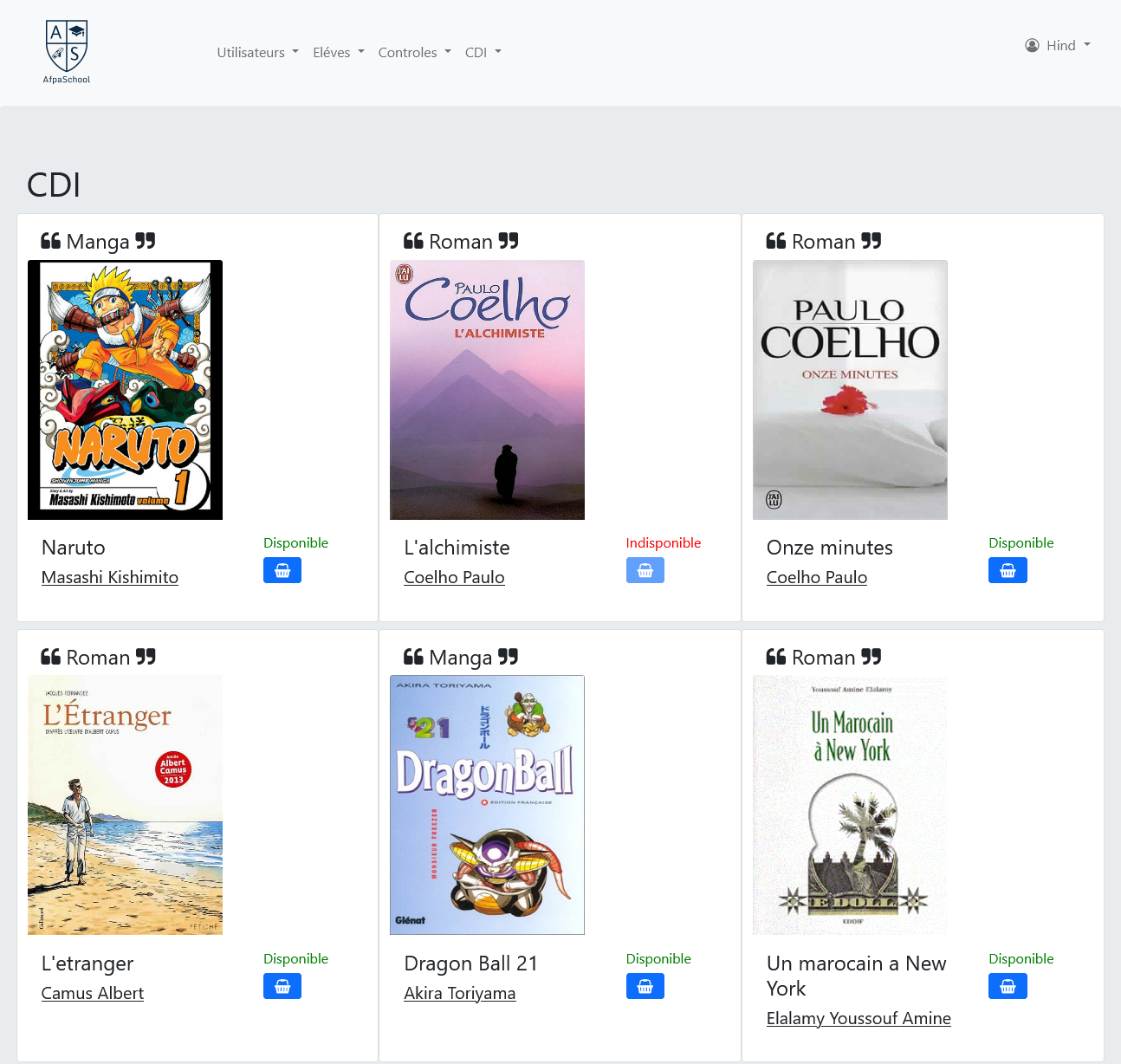


**Interface visualisation des notes (élèves)**

Les élèves ne peuvent pas interagir avec les cases du formulaire.

1. CDI

e.1. Affichage CDI

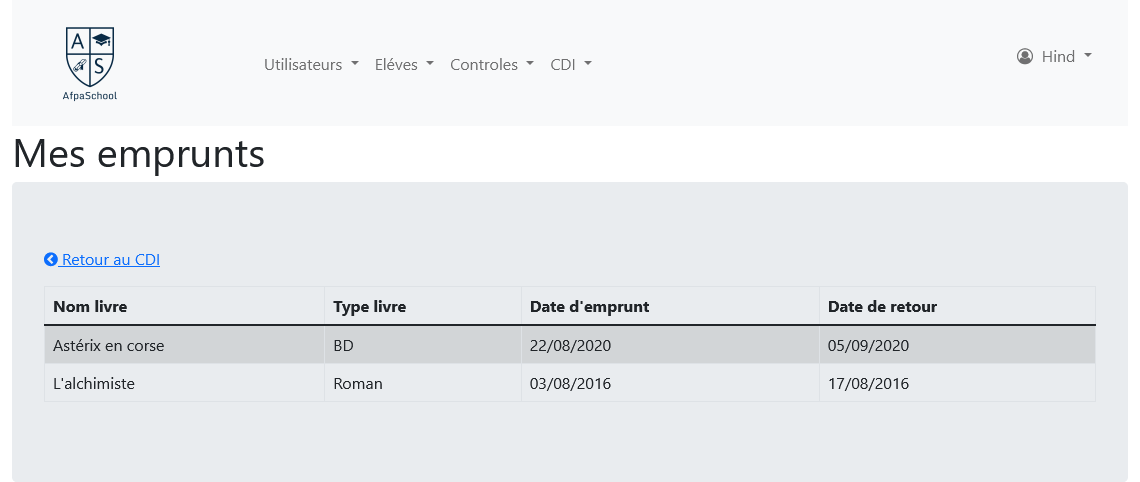


**Interface CDI**

Le CDI est accessible uniquement aux élèves, ils peuvent empruntés uniquement les livres disponibles.

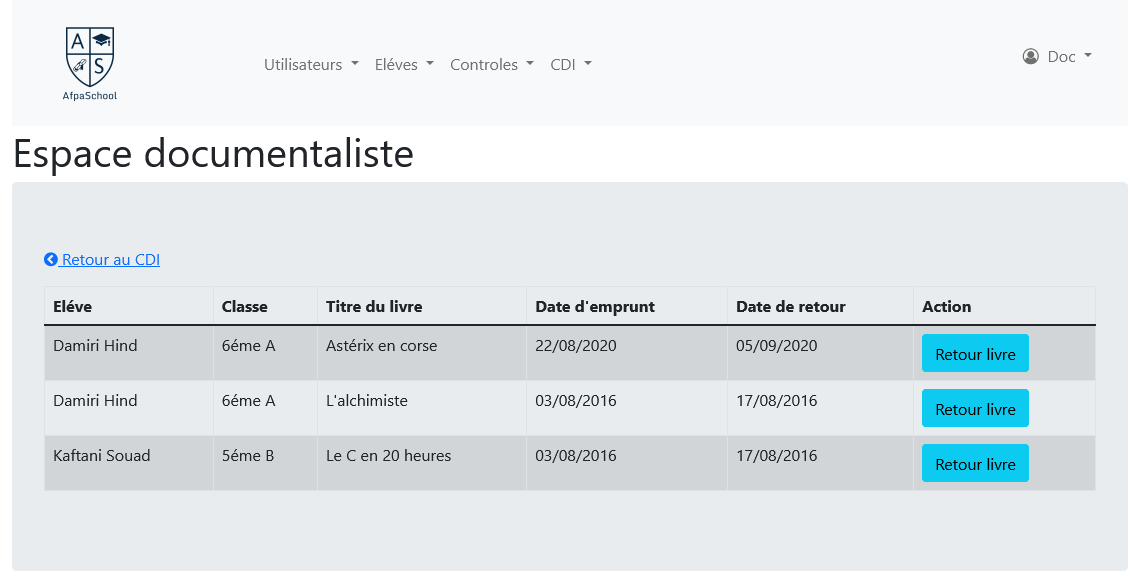
**e.2. Emprunts par élève**

**Interface emprunt par élève**



La liste des emprunts par élève la date de retour est calculé à partir de la date d’emprunt.

**e.3. Espace documentaliste**

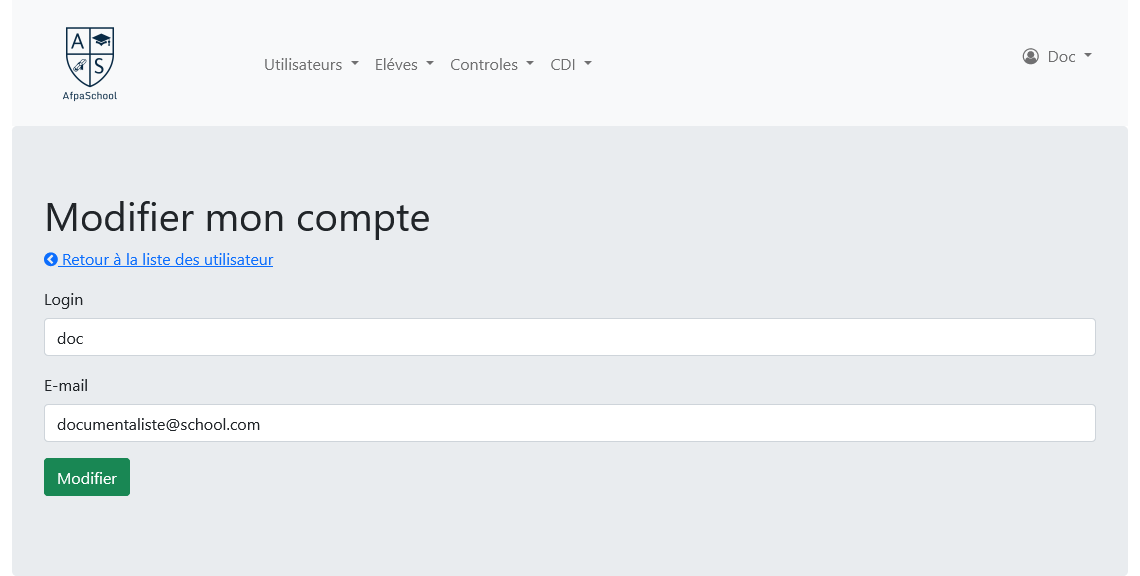


**Interface espace documentaliste**

Le documentaliste enregistre le retour des livres lorsque les élèves rendent le livre.

1. Compte

**f.1. Modification du compte**



**Interface modification du compte**

L’utilisateur modifie son compte via le formulaire le programme renvoie une erreur si :

* Que tous les champs sont bien remplis
* Que l’adresse e-mail ne soit pas dupliqué

**f.2. Modification du mot de passe**



L’utilisateur modifie son mot de passe via le formulaire, le programme renvoie une erreur si :

* Que tous les champs sont bien remplis
* Que les mots de passe sont identiques

# Conclusion :

1. **Résultat obtenu**

Ce projet a été pour moi une expérience intéressante et riche, qui m’a permis d'apprendre de nouvelles chose concernant le PHP, j’ai réussi à développer la totalité des modules qui était dans mes objectifs fixées, j’ai eu des problèmes avec la structure de mon projet que j’ai pu rattraper avec la bibliothèque PHP Altorooter qui permet de bien structurer la redirection des pages.

1. **Perspectives**

Je souhaiterai compléter cette application en changeant la gestion des classes, ajout suppression de livres au CDI, ajoutant une gestion des salles et d’emploi du temps.