

E-learning plateforme

Année universitaire : 2020 – 2021

Table des matières

Remerciements	Erreur ! Signet non défini.
Résumé	3
E-ducation	4
I. Contexte général du projet	4
1. Les besoins fonctionnels :	4
1. Les besoins non-fonctionnels :	5
II. Analyse et conception de projet	6
1. Langage de modélisation.....	6
2. Diagramme de cas d'utilisation	6
3. Diagramme de classes	9
4. Diagramme de séquence	10
5. Diagramme d'activités.....	12
6. Modélisation conceptuelle des données	14
7. Modélisation logique des données	16
III. Réalisation, interfaces, tests	17
1. Introduction.....	17
2. Outils de développement	17
3. Etude d'enchaînement de programmes.....	19
4. Production des programmes	20
5. Présentation des interfaces.....	21

Le travail consiste à réaliser une application d'enseignement à distance. Cette application a été entièrement codée en PHP, en utilisant le Framework LARAVEL et le langage SQL, ce qui permettra sa portabilité et qui va faciliter l'accès aux données n'importe quand et n'importe où. Cette application se divise en trois parties majeures :

- Espaces enseignants
- Espace élève
- Espace administration

Un enseignant, après qu'il ait été ajouté par un administrateur, pourra gérer à volonté des cours, exercices dont il pourra contrôler la difficulté pour évaluer différents élèves avec différents niveaux. De plus, il dispose d'un espace didactique où il peut créer ses fiches pédagogiques et mettre à jour son cahier de texte. Quant aux élèves inscrits, une fois connectés via le panel d'authentification, ils pourront voir les différentes leçons et exercices disponibles dans les cours où ils se sont inscrits.

Finalement, l'administrateur aura accès à un panel qui lui servira de tableau de bord dans lequel il peut gérer les élèves, les enseignants, les annonces et bien d'autres fonctionnalités utiles pour le bon fonctionnement de l'application et la limitation des fraudes et des anomalies.

I. Contexte général du projet

Durant notre stage pratique au sein du lycée Abdellah Ben Yassine, il nous a été demandé de réaliser une plateforme d'enseignement à distance.

Cette application doit tenir en compte des exigences des différents acteurs. Dans cette partie, nous allons citer les fonctionnalités du système.

1. Les besoins fonctionnels :

Ce sont l'ensemble des actions que doit effectuer un système en réponse à une demande. Dans le cas de notre application, nous distinguons les besoins par acteur :

Administrateur :

- S'authentifier.
- Changer le mot de passe
- Ajouter un élève.
- Ajouter un enseignant.
- Créer un cours.
- Gérer une classe
- Affecter un enseignant à un cours.
- Créer des annonces.
- Désactiver un compte.

Enseignant :

- S'authentifier en tant qu'enseignant, en utilisant les informations communiquées par l'administrateur.
- Changer le mot de passe fourni par l'administrateur pour assurer la confidentialité.
- Consulter les cours.
- Créer des unités.
- Créer une leçon ou un devoir.
- Accéder à un espace pédagogique/didactique qui lui facilitera la mise à jour de son cahier de texte et la conception de ses fiches pédagogiques.
- Consulter les annonces.
- Noter les devoirs rendus par les élèves en plus des remarques.
- Déposer un document (leçon ou un devoir).

Elève :

- S'authentifier en tant qu'élève, en utilisant les informations communiquées par l'administrateur.
- Changer le mot de passe fourni par l'administrateur pour assurer la confidentialité.
- Consulter les annonces.
- Consulter les cours.
- Rendre un devoir

1. Les besoins non-fonctionnels :

Les besoins non fonctionnels précisent l'environnement de l'application.

Connectivité	<ul style="list-style-type: none">• Communication avec une base de données• Hachage de mot de passe• Déconnexion
Performance	<ul style="list-style-type: none">• Temps de réponse très court
Disponibilité	<ul style="list-style-type: none">• Le site doit être disponible aux utilisateurs à tout moment
Intégrité	<ul style="list-style-type: none">• Bonne gestion des erreurs
Compatibilité	<ul style="list-style-type: none">• Compatible avec les différents navigateurs web

Tableau 1 : Besoins non-fonctionnels

II. Analyse et conception de projet

Après avoir présenté le contexte général du projet ainsi que la spécification des besoins, cette partie sera dédiée à la spécification des exigences logicielles.

Dans cette section, nous allons commencer par justifier le choix du langage de modélisation. Ensuite, nous allons présenter le diagramme de cas d'utilisation.

1. Langage de modélisation

Dans le cadre de notre projet, nous avons opté pour le langage UML comme un langage de modélisation parce qu'il est le plus adapté aux projets à caractère orienté objet. Cette modélisation va nous permettre de bien définir les besoins fonctionnels de l'école et faciliter leur traduction en solution logicielle. Ainsi, ce langage de modélisation offre plusieurs diagrammes qui nous seront utiles pour produire un système cohérent.

2. Diagramme de cas d'utilisation

Les cas d'utilisation décrivent l'ensemble des fonctionnalités réalisées par un système, en réponse aux services d'un acteur.

2.1 Identification des acteurs

Dans le cas de notre application, les principaux acteurs sont :

L'administrateur : c'est le responsable de l'administration du site, il gère tout.

L'enseignant : c'est un utilisateur du site, il ajoute des leçons et des exercices et dispose d'un espace didactique où il ajoutera ses fiches pédagogiques et remplira son cahier de texte.

L'élève : c'est aussi un utilisateur du site, il télécharge ses cours et dépose ses devoirs.

2.2 Diagramme de cas d'utilisation : Enseignant – Admin



Figure 1 : Diagramme de cas d'utilisation Enseignant - Admin

2.3 Diagramme de cas d'utilisation : Elève– Admin

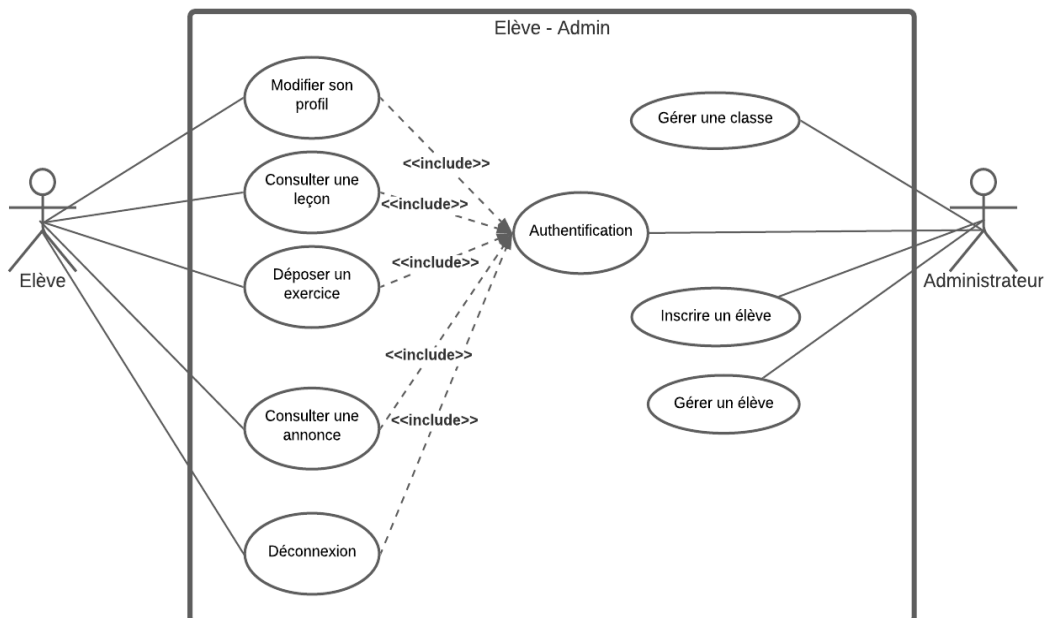


Figure 2 : Diagramme de cas d'utilisation Elève – Admin

2.4 Raffinement des cas d'utilisation :

- Raffinement du cas d'utilisation « S'authentifier » :

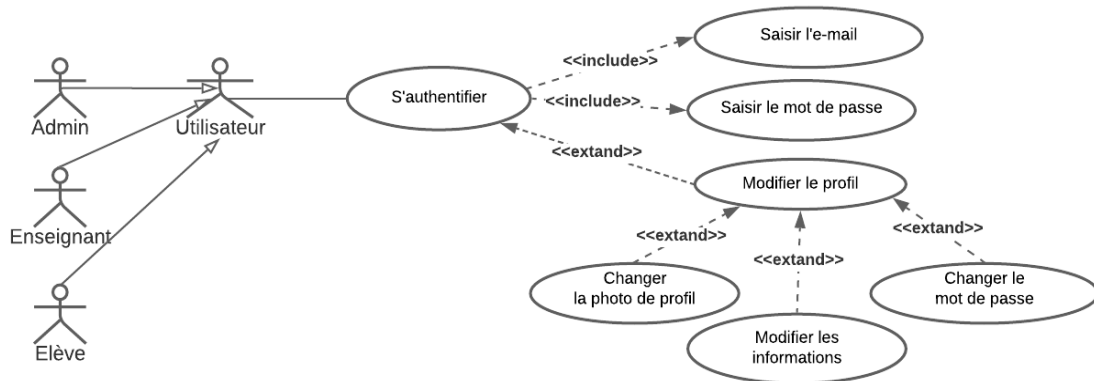


Figure 3 : Raffinement du cas d'utilisation - S'authentifier

- Raffinement des cas d'utilisation « Gérer (enseignant/étudiant/ classe/ annonce) »

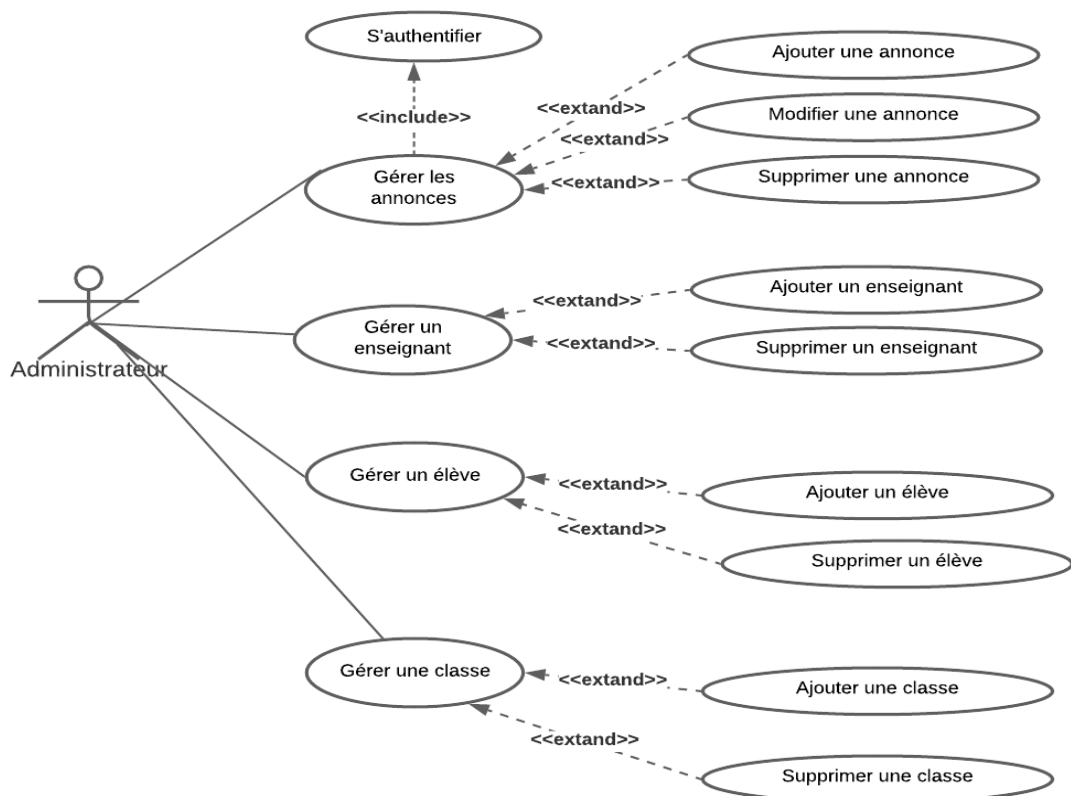


Figure 4 : Raffinement du cas d'utilisation - Gérer Annonce/ Enseignant/ Etudiant

3. Diagramme de classes

Le diagramme ci-dessous modélise les classes, les attributs et décrit clairement la structure du système.

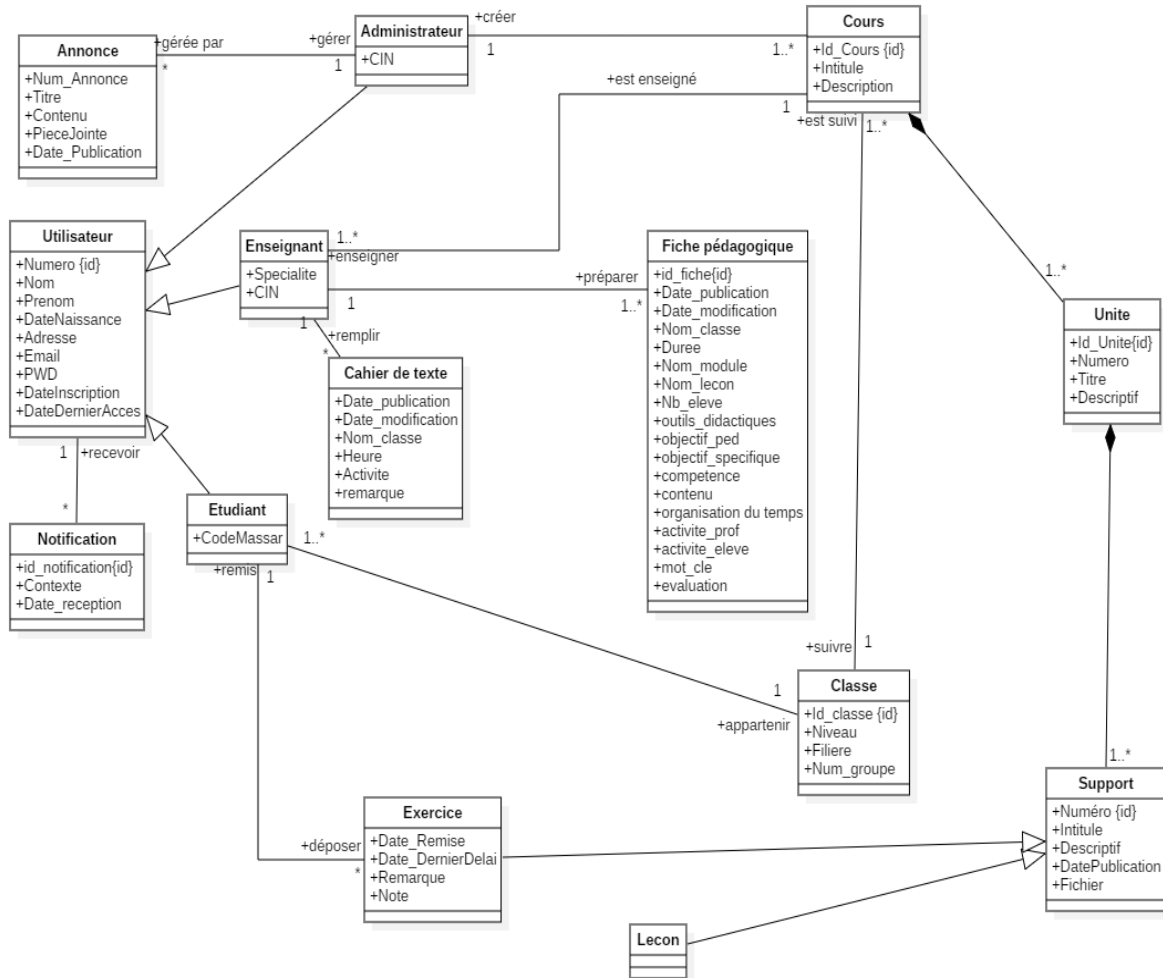


Figure 5 : Diagramme de classes

4. Diagramme de séquence

Le diagramme de séquence permet de décrire les interactions des objets par ordre chronologique.

4.1 Diagramme de séquence « Authentification »

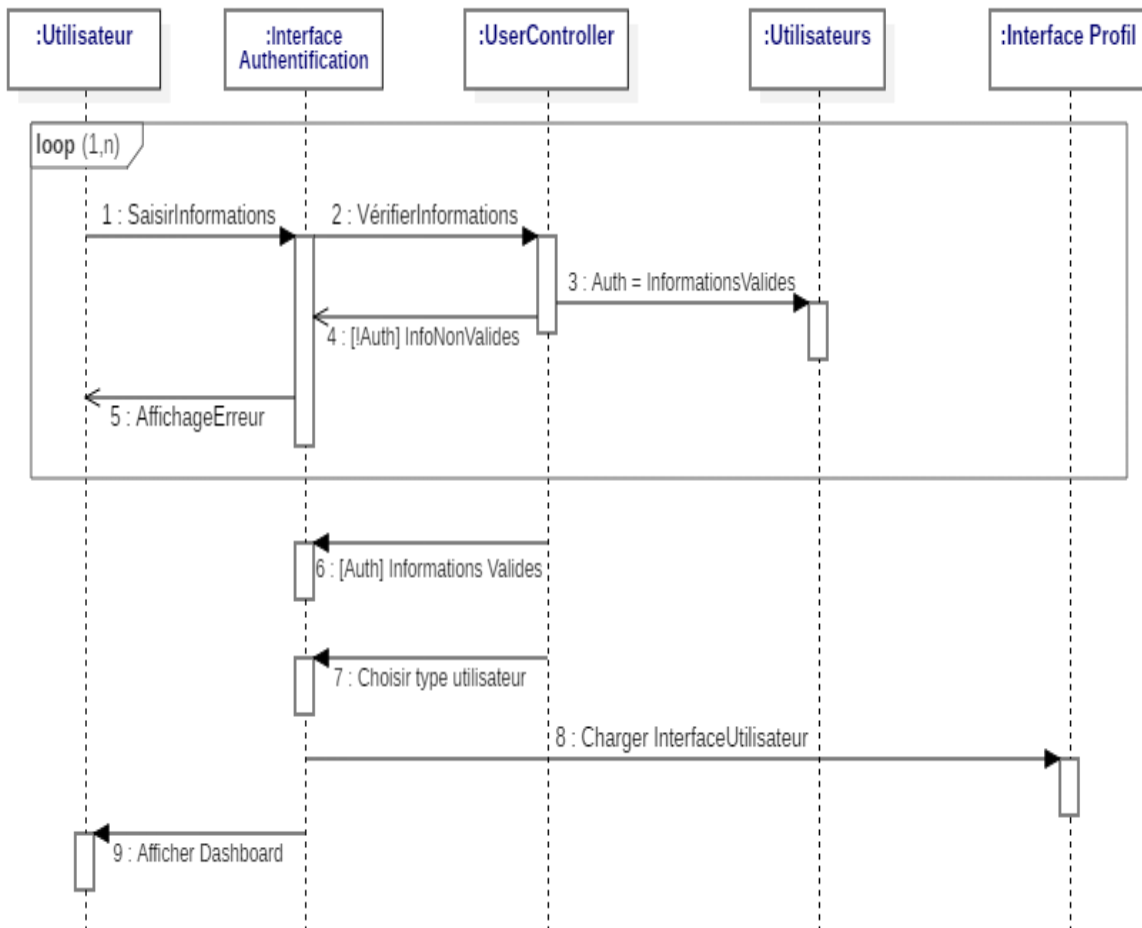


Figure 6 : Diagramme de séquence -Authentification Utilisateur-

4.2 Diagramme de séquence « Ajout d'un cours »

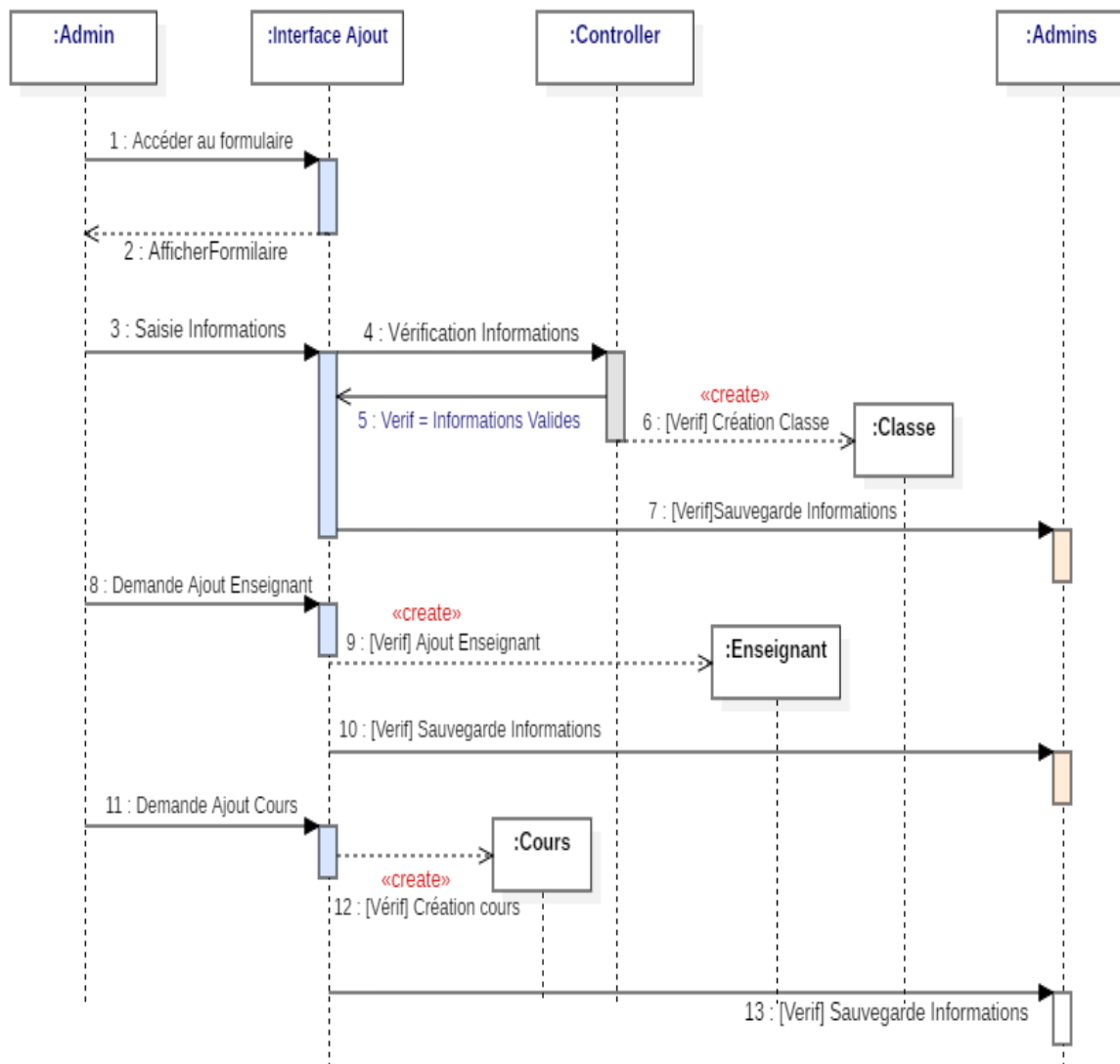


Figure 7 : Diagramme de séquence - Ajout d'un cours -

5. Diagramme d'activités

Les diagrammes d'activités sont conçus pour décrire les activités du système, à partir d'un point de départ jusqu'au point d'arrivée. Dans le cas de notre projet, nous avons réalisé trois diagrammes d'activités :

5.1 Diagramme d'activités « Gestion du compte d'un élève » :

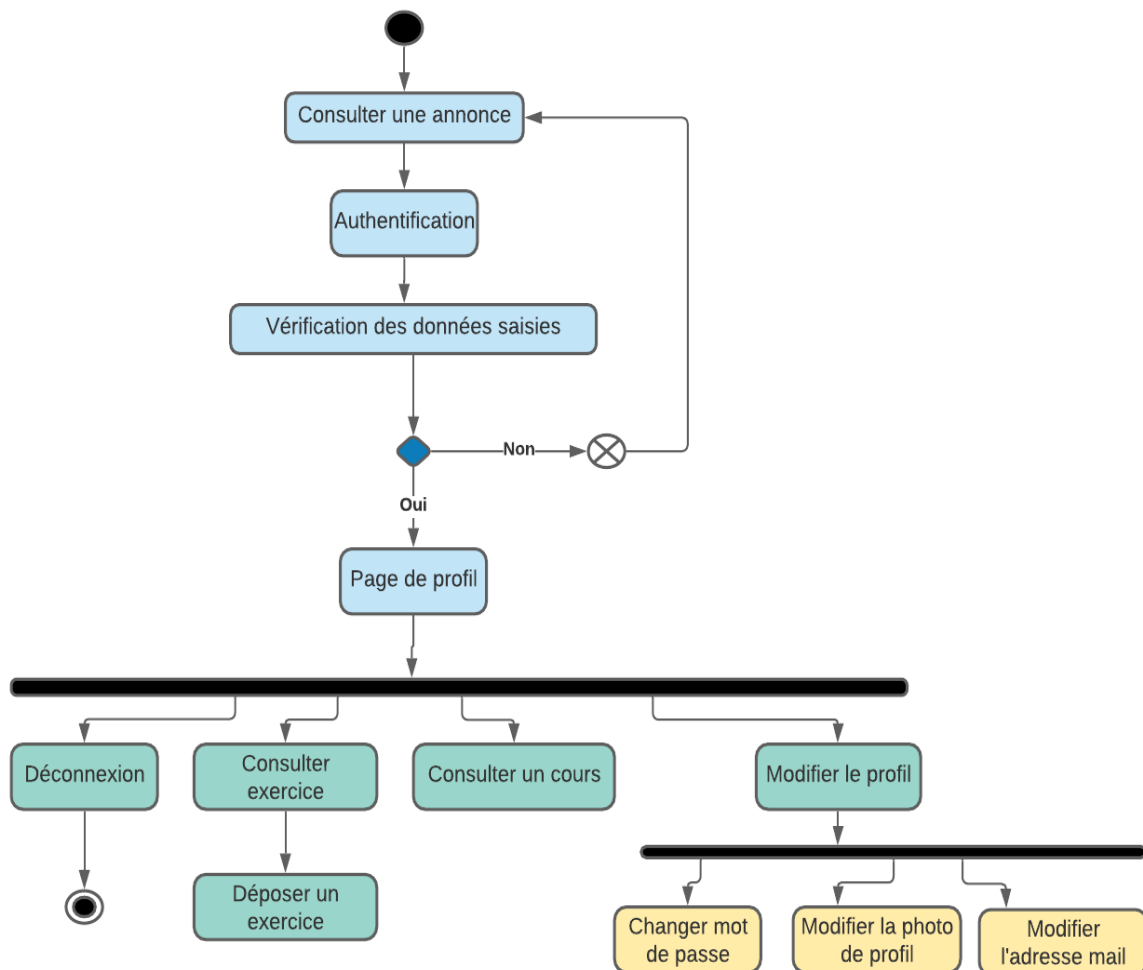


Figure 8 : Diagramme d'activités - Espace élève

5.2 Diagramme d'activités « Gestion du compte d'un professeur » :

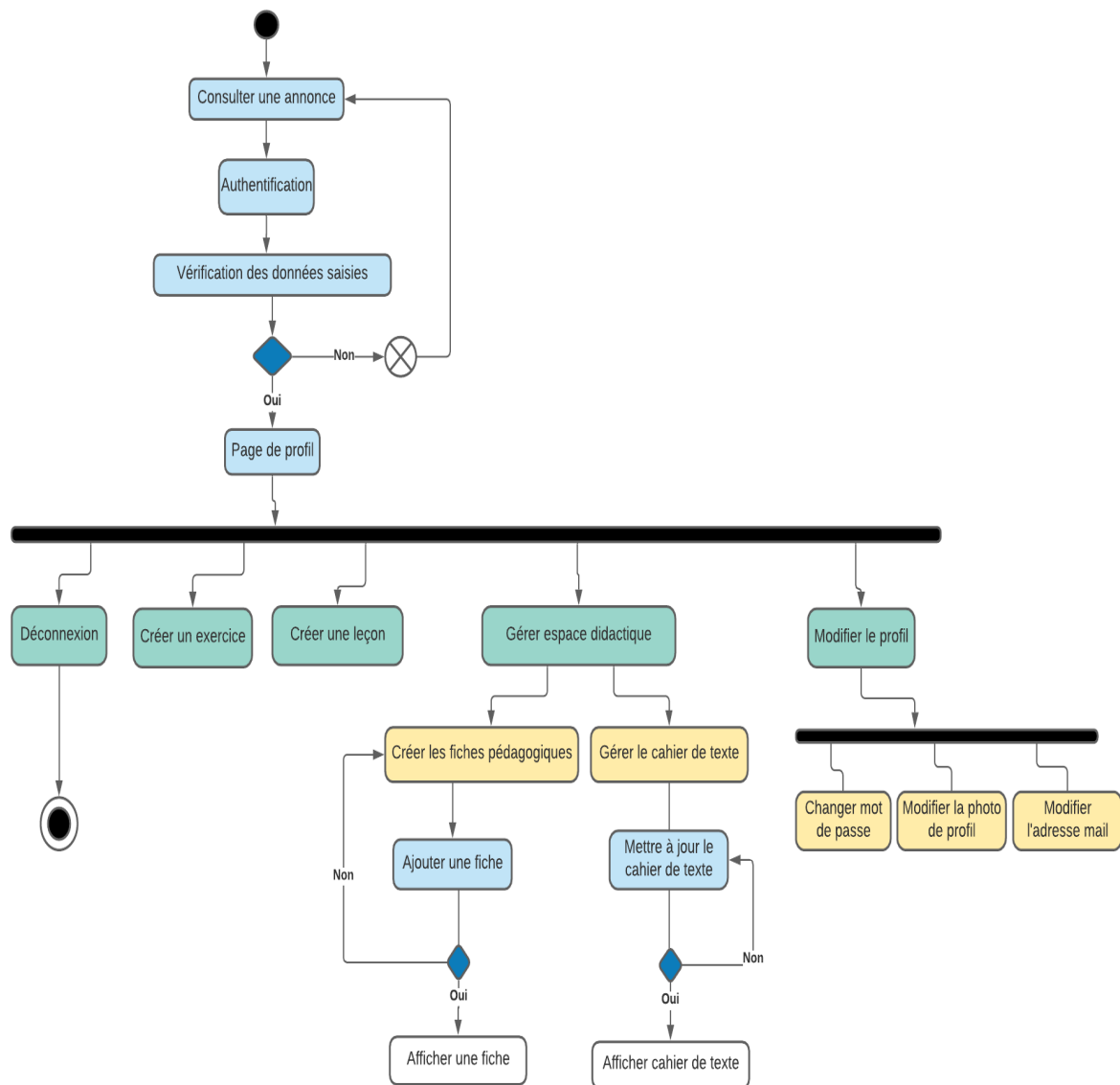


Figure 9 : Diagramme d'activités - Enseignant

5.3 Diagramme d'activité « Espace administrateur »

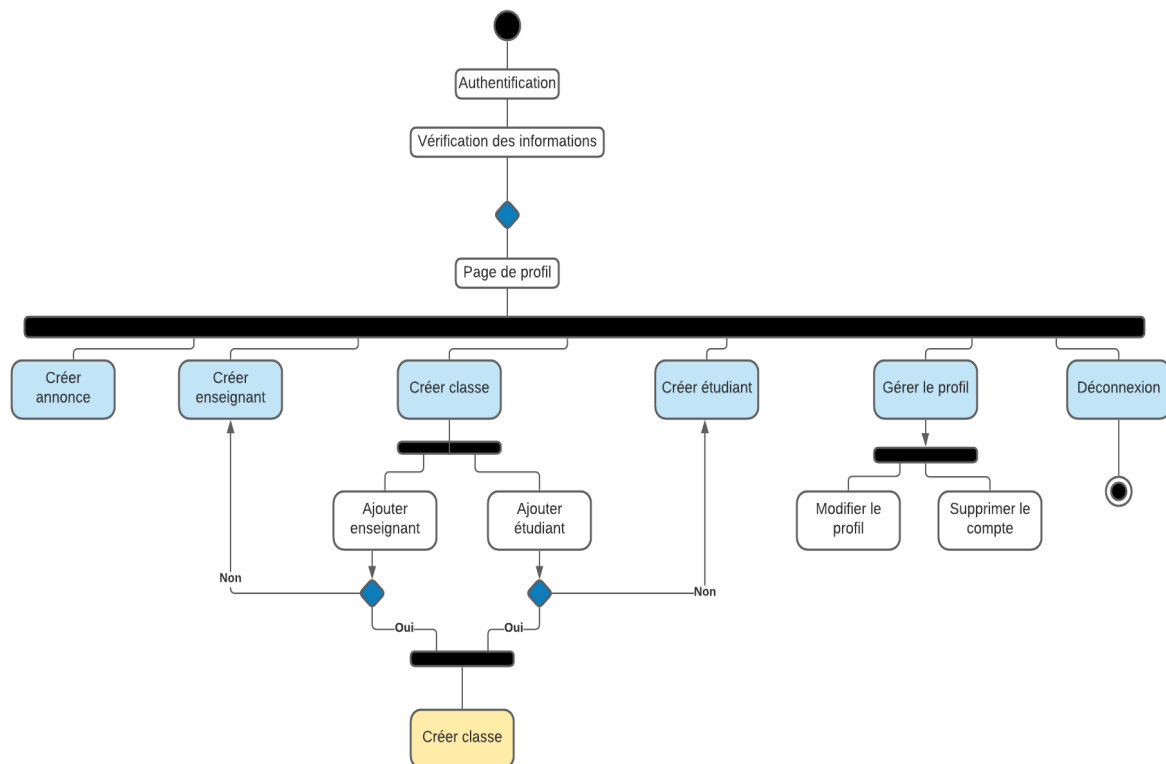


Figure 10 : Diagramme d'activités – Administrateur

6. Modélisation conceptuelle des données

Dans le but de concevoir un système d'information clair et cohérent, nous avons opté pour la méthode MERISE. Cette méthode se base sur la séparation des données et des traitements. Elle permet de présenter facilement les données et de comprendre le fonctionnement du système. Nous avons conçu un modèle conceptuel de données, qui, nous a permis de passer au modèle logique (MLD).

➤ Modèle Conceptuel de Données – MCD :

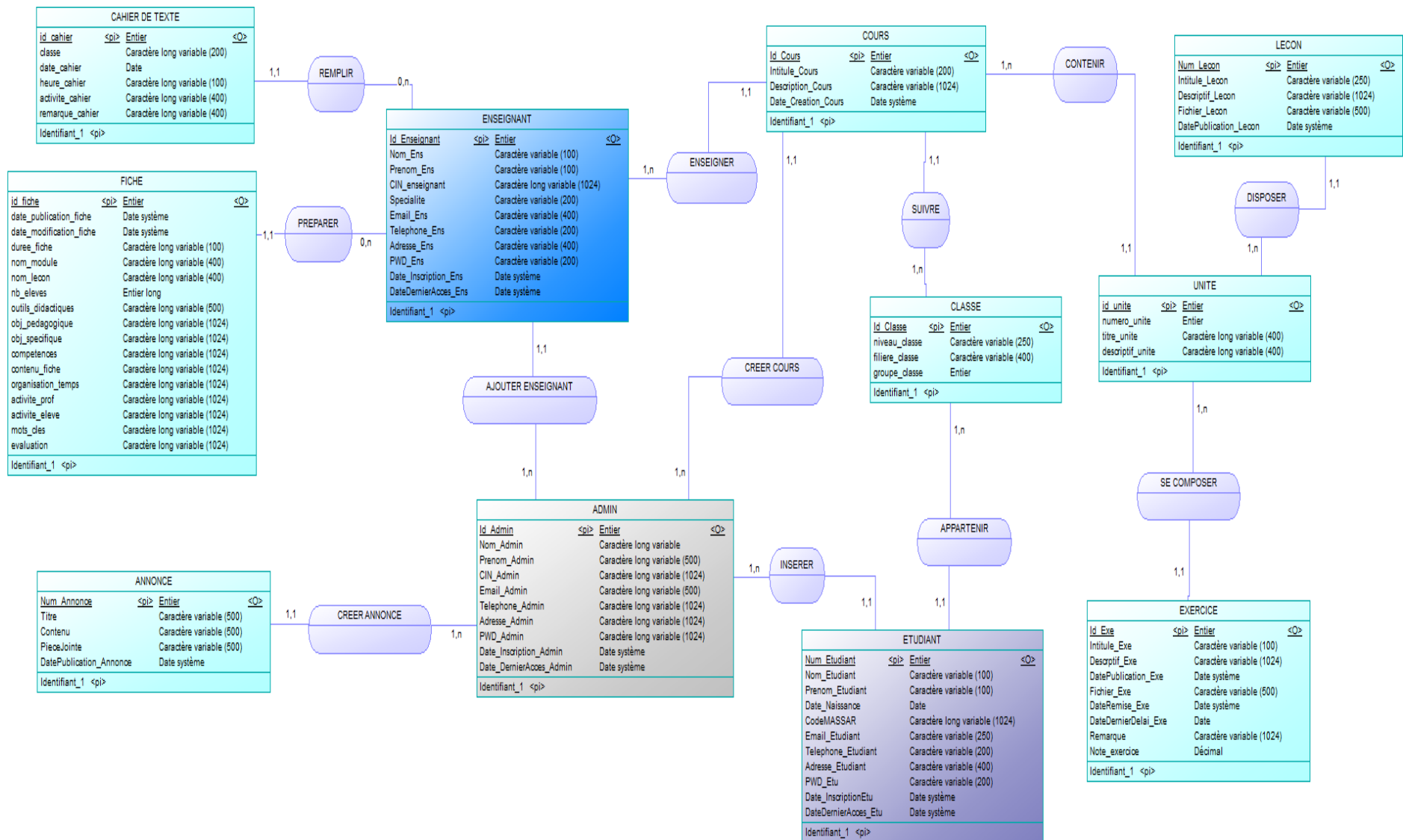


Figure 11 : Modèle conceptuel de données - MCD

7. Modélisation logique des données

➤ Modèle logique des données - MLD

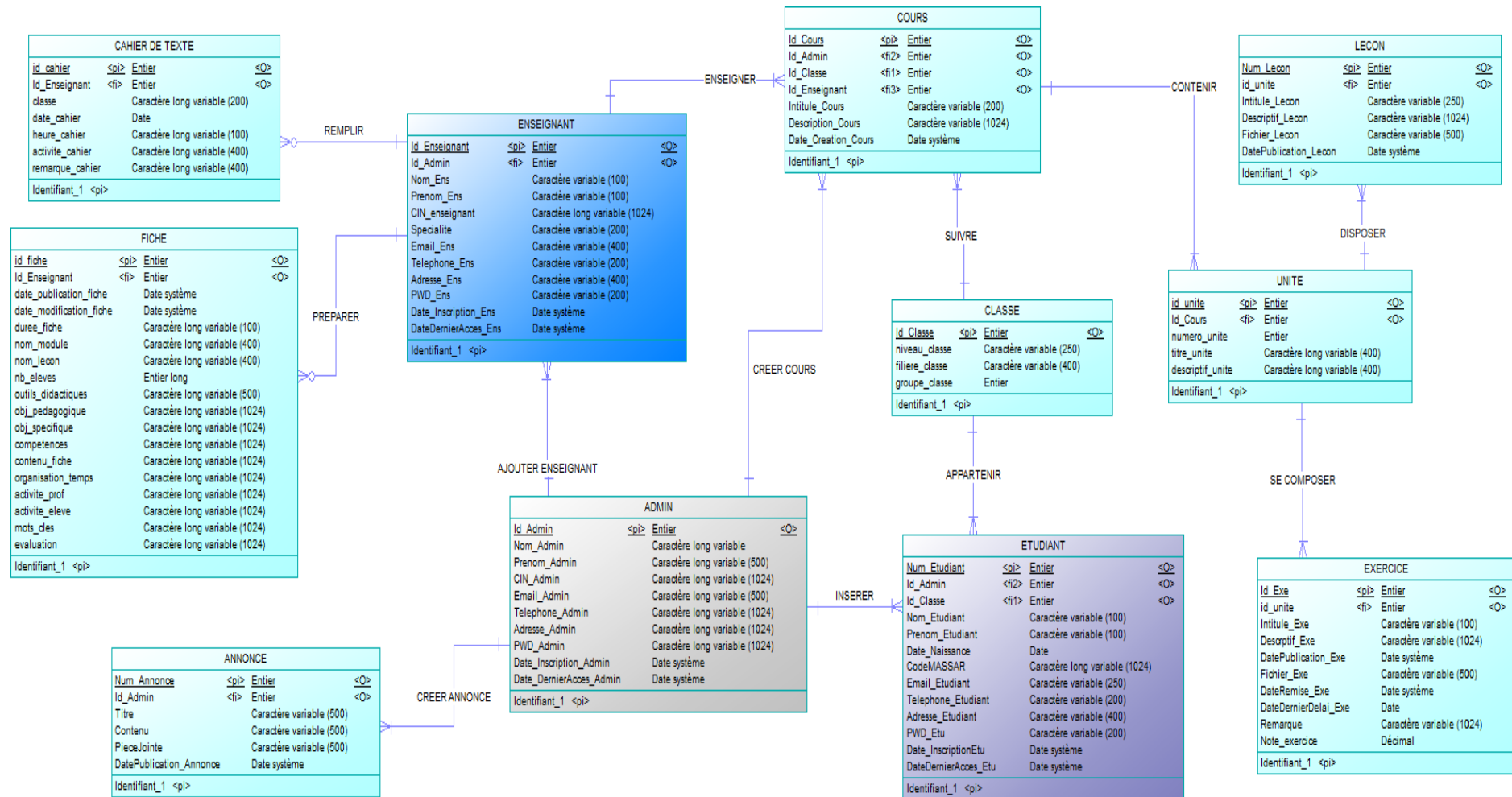


Figure 12 : Modèle logique de données - MLD

III. Réalisation, interfaces, tests

1. Introduction

Après avoir mené des phases d'analyse, de spécification de besoins et de conception, nous avons passé à la phase de réalisation. Durant la réalisation de notre application, nous avons essayé de choisir des technologies récentes d'une part pour développer ce que nous avons déjà appris durant notre formation initiale, d'autre part pour que notre projet soit conforme à l'essor technologique et que les interfaces soient claires et faciles à utiliser.

2. Outils de développement

Pour le développement de notre application, nous avons choisi les outils suivants :

Laravel 8		C'est un Framework web open-source, complet, écrit en PHP. Il adopte une architecture MVC (Model-View-Controller) et il est totalement orienté objet.
AJAX		C'est une méthode qui utilise des différentes technologies ajoutées au navigateur web, qui permet d'effectuer des requêtes au serveur web et modifier la page actuelle sans avoir besoin d'afficher une nouvelle page.
HTML5		C'est un langage de balisage utilisé pour créer et concevoir des pages WEB. Il permet de présenter des documents hypertextes, d'où son nom, destinés à être affichés sur le navigateur.
CSS3		Les feuilles de styles en cascade, sont un langage qui permet de gérer la mise en forme d'une page WEB (titres, alignement, bordures, polices, etc...). Il permet aussi de définir des règles et contrôler l'apparence et la disposition des pages HTML.
WAMP		C'est un ensemble de logiciels et une plateforme WEB permettant, sans se connecter à un serveur externe, de faire fonctionner des scripts PHP. Il est composé de deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour gérer les bases de données MySQL.

Bootstrap5		C'est un Framework, open-source, qui utilise les langages HTML, CSS et JavaScript. Il contient des formulaires, des boutons, des d'autres éléments interactifs. Il permet de créer facilement des sites WEB avec un design responsive, qui s'adapte à tous les écrans.
JavaScript		Le JavaScript est un langage informatique de programmation des scripts, employé pour produire des pages WEB dynamiques. C'est un langage orienté objet dont l'exécution est effectuée par les navigateurs.
Git		Git est un système de contrôle de version distribué gratuit et open source conçu pour tout gérer, des petits aux très grands projets avec rapidité et efficacité.
Github		GitHub est un fournisseur d'hébergement Internet pour le développement de logiciels et le contrôle de version à l'aide de Git.

Tableau 2 : Outils de développement

3. Etude d'enchaînement de programmes

Cette section consiste à détailler les principales fonctionnalités du système. Ceux-ci dépendent de la nature de l'utilisateur.

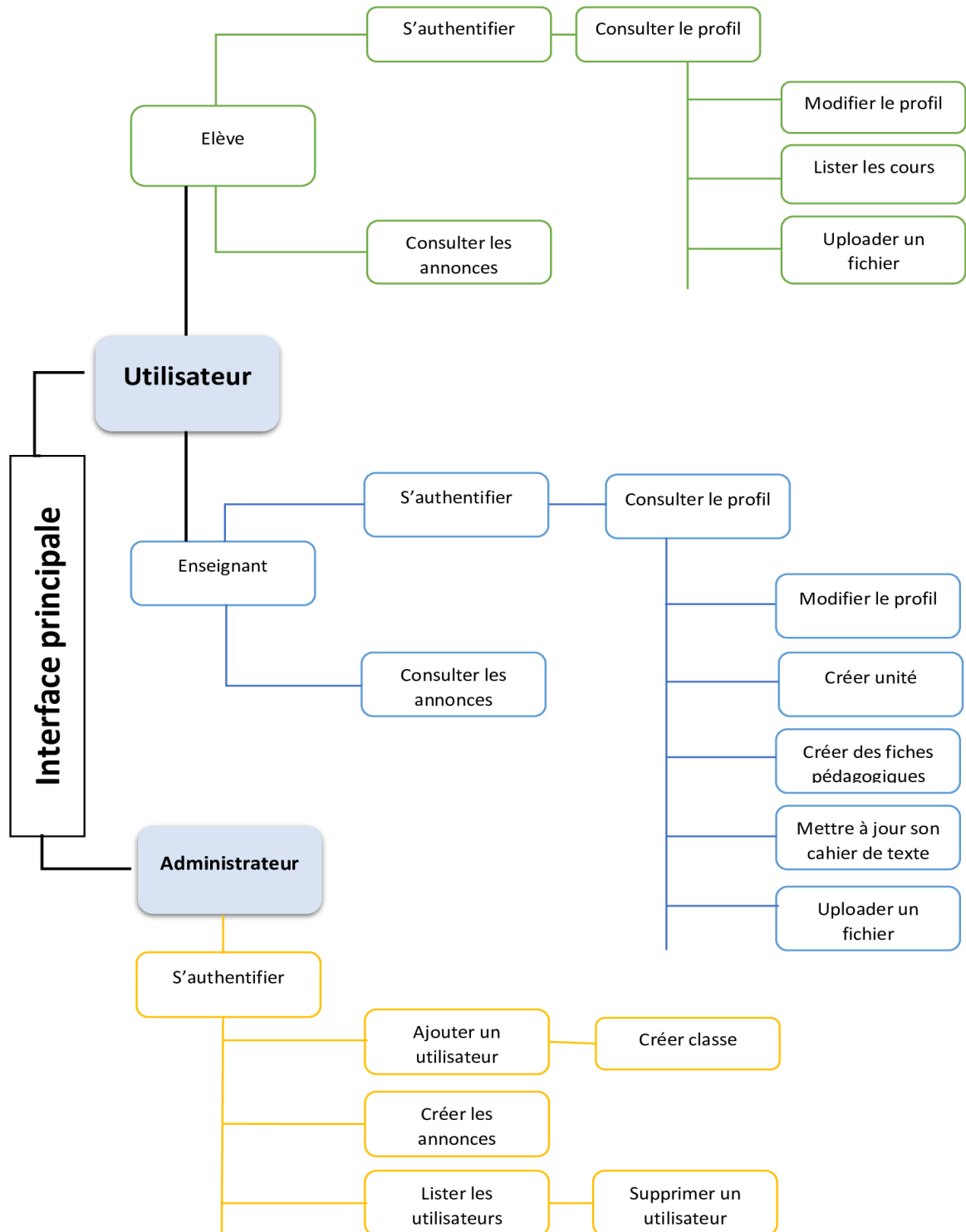


Figure 13 : Menu général de l'application

4. Production des programmes

Pour développer notre application nous avons choisi le Framework Laravel. Comme une grande partie des autres Framework PHP, Laravel a aussi une architecture dite MVC (Model – View – Controller). (Laravel, s.d.).

➤ Architecture MVC

Comme déjà évoqué, Laravel se base sur le patron de conception MVC : Modèle, Vue, Contrôleur. Cette architecture impose la séparation entre les données, la présentation et les traitements.

Couche Modèle :

Le modèle interagit avec la base de données. Il permet d'effectuer des opérations de sauvegarde de données, de récupération, d'affichage, de suppression et de mise à jour.

Couche Vue :

La vue est l'interface avec laquelle un utilisateur interagit. Elle est chargée d'afficher ce que le modèle renvoie. Par la suite, elle a comme deuxième tâche de recevoir les actions réalisées par l'utilisateur, qui seront envoyés au contrôleur.

Couche contrôleur :

Le contrôleur traite les actions d'un utilisateur, ensuite il met à jour le modèle et il passe les données la vue. Quand il reçoit les événements d'un utilisateur, il enclenche des actions à réaliser.

Couche Middleware :

C'est une nouvelle couche dans le Framework Laravel. Elle est située entre la requête et la réponse. Elle sert à agir sur des requêtes afin d'adapter la configuration de l'application. Pour notre cas, nous nous sommes servis de cette couche pour gérer les droits d'accès.

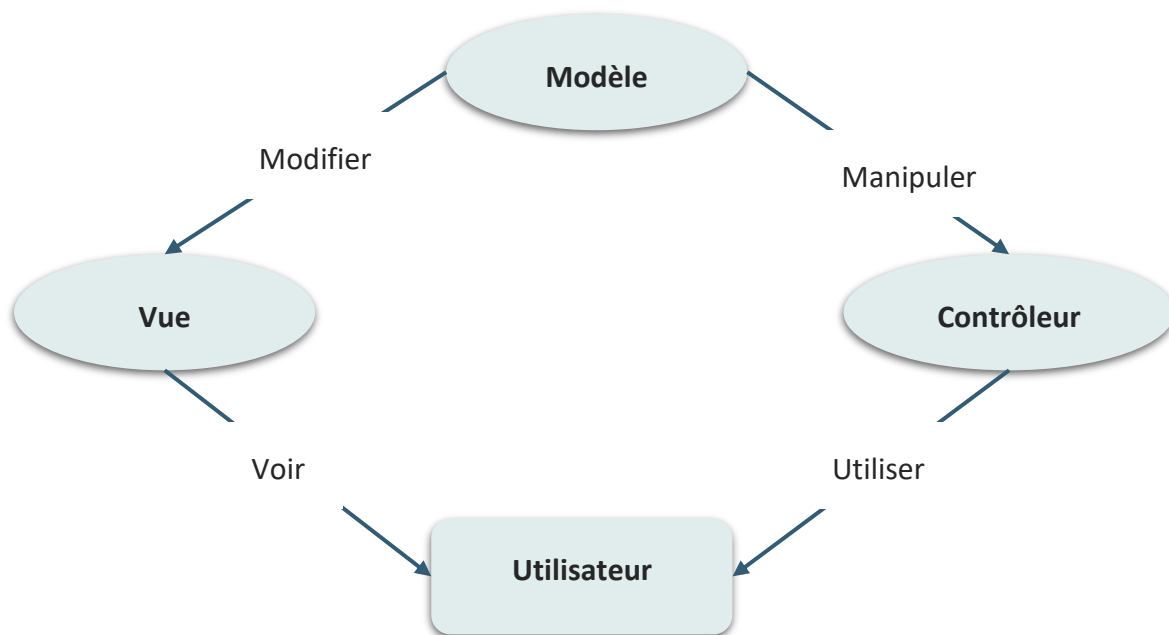


Figure 14 : Architecture MVC

5. Présentation des interfaces



Figure 15 : Page d'accueil - Les annonces

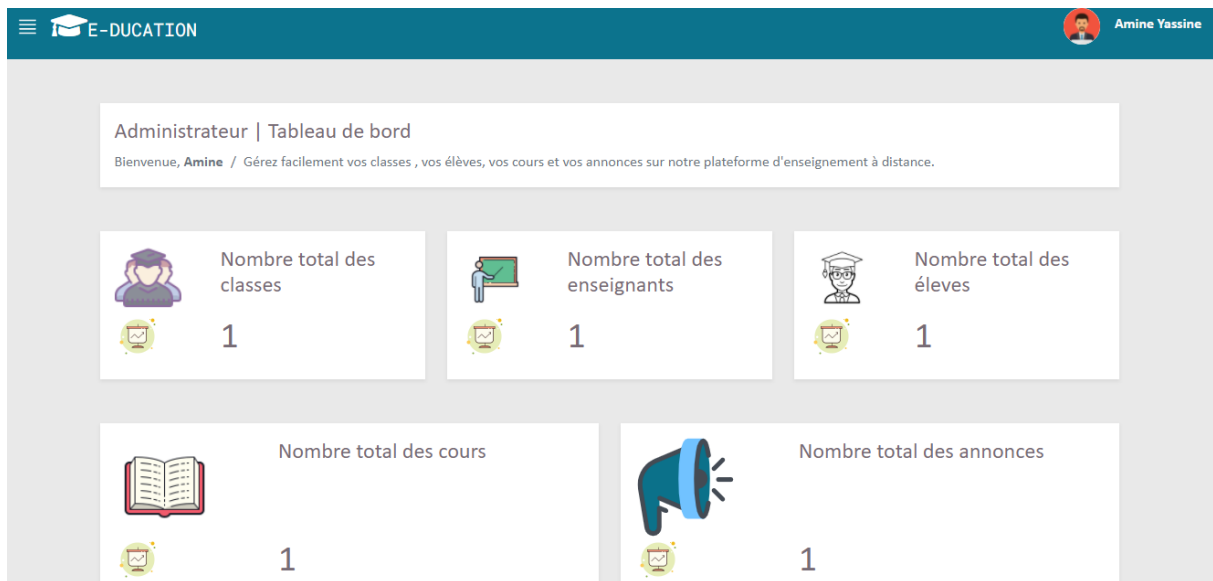


Figure 16 : Tableau de bord - Administrateur

Classe
 Créer une classe

Intitulé :

Filière :

Niveau :

Groupe :

[Ajouter](#)

Figure 17 : Espace administrateur - Formulaire d'ajout d'une classe

Liste des classes [Créer une classe +](#)

Afficher éléments

Rechercher :

Intitulé	Niveau	Filière	Groupe	Dernière modification	Actions
BS-1	BAC	Scientifique	1	2021-06-22 01:21:36	Modifier Supprimer

Affichage de l'élément 1 à 1 sur 1 éléments

Précédent **1** Suivant

Figure 18 : Espace Administrateur - Liste des classes


Enseignant

Ajouter un enseignant

Nom :

Prénom :

Email :

Téléphone :

CIN :



Spécialité :


Date de naissance :

Adresse :

[Ajouter](#)


Figure 19 : Espace Administrateur - Formulaire d'ajout d'un enseignant



E-EDUCATION


Laila Hamza

Bienvenue, **Laila** .


Consultez vos :



Classes

Gérez facilement vos classes, ajoutez les cours et les exercices à communiquer à vos élèves.


[Voir plus](#)



Cahier de texte

A la fin de chaque séance, inscrivez les activités et les remarques de chaque classe.

[Voir plus](#)



Outils pédagogiques

Remplissez vos fiches pédagogiques et gardez les dans cet espace.

[Voir plus](#)

Figure 20 : Espace Enseignant - Tableau de bord

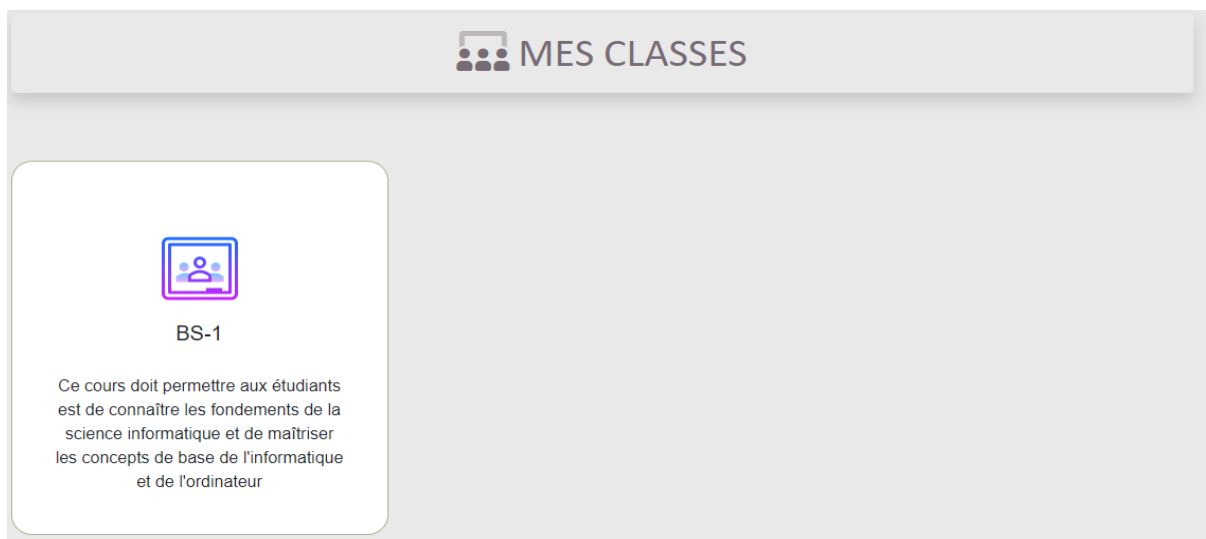


Figure 21 : Espace Enseignant - Les classes

Unités

AJOUTER UNE UNITÉ

Cours : Informatique - TCS1 ▼ Numéro de l'unité : Nombre d'unité ▼

Intitulé : Nom

Description : A propos de ce cours

AJOUTER ✓

Figure 22 : Espace Enseignant - Ajout d'une unité

CLASSE	COURS	DATE	ACTIVITÉS	REMARQUES
TCS1	ALgorithmes	12/12/2021	Ecrire l'algorithme de multiplication de deux nombres	Aucune

Figure 23 : Espace Enseignant - Cahier de texte

Date :	2021-06-24 20:59:53	Durée :	5h
Classe :	TCS1	Matière :	Informatique
Nom du module :	LEs logiciels	Nom de la leçon :	Les tableaux
Nombre d'élèves :	30	Outils didactiques :	ordinateur + projecteur
Objectifs pédagogiques :	Savoir utiliser les tableaux		
Objectifs spécifiques :	Savoir utiliser les tableaux numériques		
Compétences à construire :	Produire un document de calcul		
Pré-requis :	aucun		
Organisation du contenu	4 chapitres		

SCENARIO PEDAGOGIQUES:
Déroulement de la séquence pédagogique devant l'élève

Organisation du temps	Activité du professeur	Activité de l'élève	Supports utilisés Mots clés
30min	Faire les opérations de tri, ajustement, calcul	Poser et répondre aux questions	Tri, ajustement
20min	Montrer aux apprenants un tableur	Ecouter	Tableur

Evaluation :	Interrogative
--------------	---------------

Figure 24 : Espace Enseignant - Fiche pédagogique

☰ E-DUCATION
 Asma Mostare

Tableau de bord

Bienvenue, **Asma** / Téléchargez vos cours et les exercices à faire.

BS-1

Ce cours doit permettre aux étudiants est de connaître les fondements de la science informatique et de maîtriser les concepts de base de l'informatique et de l'ordinateur

Figure 25 : Espace Elève - Tableau de bord

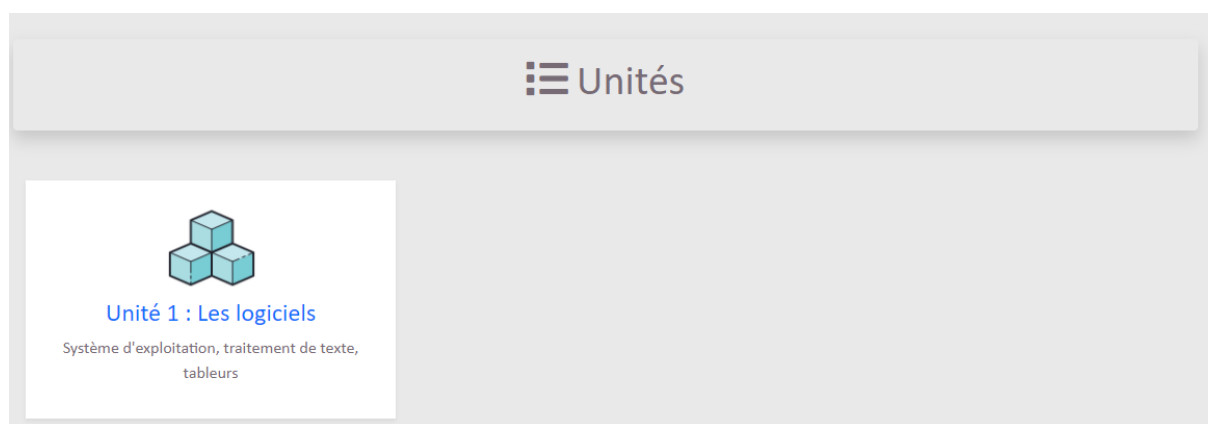


Figure 26 : Espace Elève - Liste des unités

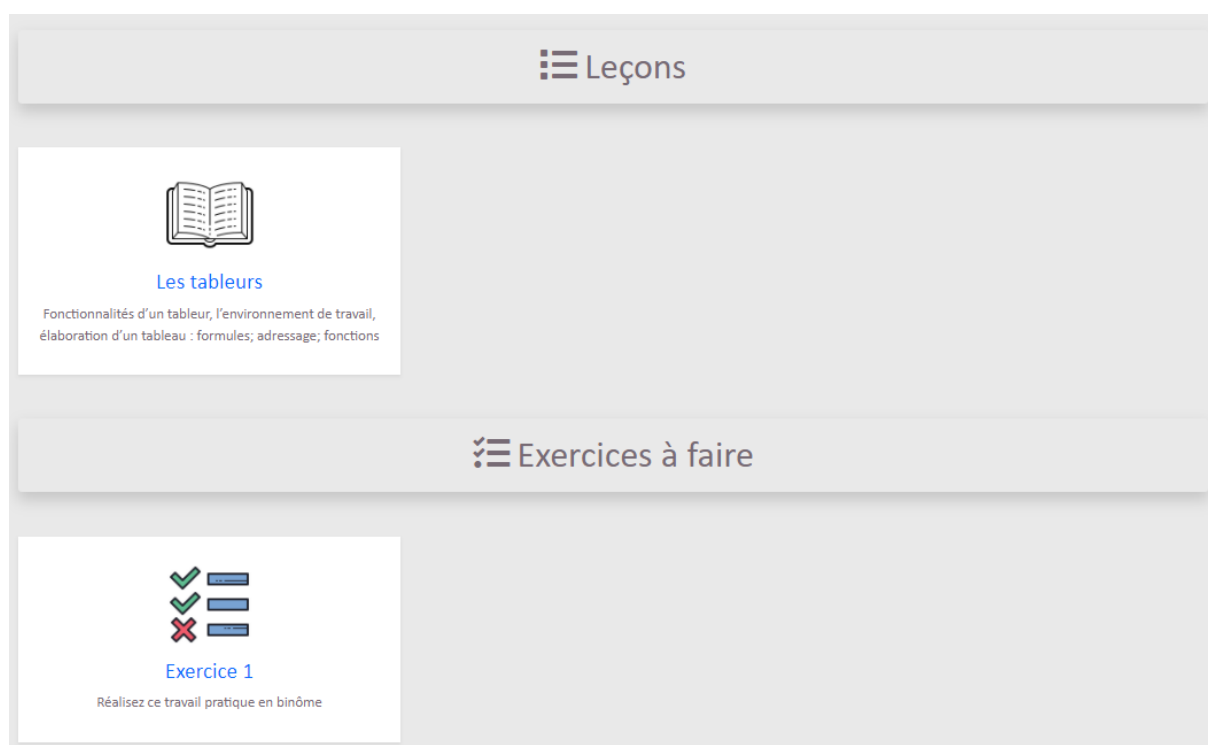


Figure 27 : Espace Elève - Leçons et exercices