|  |  |  |
| --- | --- | --- |
| IsetFinal_mai_2005%20copier2 | **Ministère de l'Enseignement Supérieur, de la Recherche Scientifique**  **Direction Générale des Études Technologiques** | WhatsApp Image 2022-06-12 at 20.10.58.jpegLogo  Entreprise |
| **Institut Supérieur des Études Technologiques de Djerba**  **Département Technologies de l’informatique** | | |

|  |  |
| --- | --- |
| **Code projet** | **…………** |

**Rapport de**

**STAGE DE FIN D’ETUDES**

**Présenté en vue de l’obtention du diplôme de**

**Licence Appliquée en Technologies de**

**L’Informatique**

**Parcours: MDW**

**Développement d’une application de gestion d’école privée**

**Elaboré par :**

**Maamech Ayoub**

**Encadré par :**

**Mr. Suissi Oussema**

**Effectué à :**

**Entreprise : PowerMedia**

**Encadreur : Mr. Debouba Mekki**

**Année universitaire : 2021/2022**

***Dédicaces***

**À ma chère mère,**

**À mon cher père**

Pour tous les sacrifices qu’ils ont faits et pour tout le soutien qu’ils ont offert tout au long de mes études.

J’espère qu’ils peuvent trouver dans ce modeste travail un témoignage d’amour et d’affection envers eux.

À mes frères

Pour leur soutien et leurs conseils précieux tout au long de mes études.

À mes amis et mes collègues

Pour leur aide et support dans les moments difficiles.

À ma famille, mes proches et à ceux qui me donnent de l’amour et de la vivacité

À tous ceux que j’aime

***Remerciements***

Louanges tout d’abord à ALLAH qui est l’origine de toute réussite dans notre vie.

J’adresse mes sincères remerciements à Monsieur Suissi Oussema pour leur encadrement l’intérêt qu’il a porté à mon travail, sa bienveillance, sa rigueur scientifique, ses discussions fructueuses, ces conseils judicieux lors de la conception et la réalisation de ce projet et son aide à la rédaction et la bonne organisation de ce rapport.

Par la même occasion, j’adresse mes remerciements à Monsieur Mekki Debouba pour ses efforts, ses hautes qualités humaines et sa bonne coordination durant les quatre mois de mon stage.

*Je tiens également à exprimer toute mes gratitudes aux membres du jury pour avoir accepté d’assister à ce modeste travail.*

*Enfin, je ne veux pas oublier tous ceux qui ne cessent pas de nous encourager de près ou de loin.*

Introduction générale

De nos jours, la gestion des données d'une façon automatisée occupe une place privilégiée dans le monde de l'informatique. Grâce à cette technologie, le temps de recherche des données a chuté de manière considérable, ce qui laisse croire qu'elle a encore de beaux jours devant elle.

C’est dans ce contexte que plusieurs sociétés essayent de profiter au maximum possible de ces technologies afin d’améliorer leurs productivités et de faire face à quelques problèmes pénibles qui peuvent constituer un obstacle de progression.

Dans ce cadre, la société **Powermedia** m'a confié la responsabilité de développer une application web permettant de gérer une école privée, installé à Djerba.

Ce concept est né pour répondre à un ensemble de besoins dont : la gestion des élèves, la gestion des enseignants, la gestion des notes et des classes…

L’implémentation n’est pas la seule difficulté à surmonter, il faut savoir que la gestion d’un établissement scolaire privé est beaucoup plus complexe et différente que celle dans un établissement public, en effet les établissements privés ont un besoin de gérer les paiements, les activités parascolaires…etc.

Ce rapport est structuré comme suit :

Dans le premier chapitre, nous présentons également l'organisme d’accueil, le cadre général de notre stage à savoir la société « Powermedia » et les services et les domaines d’activité de la société..

Le second chapitre intitulé « Etude Préalable » présente, en détail, le concept et les objectifs de notre plateforme, ainsi que la spécification des besoins fonctionnels et non fonctionnels de notre projet.

Le troisième chapitre « Etude conceptuelle », présente en détail, les étapes de conception de et la méthodologie de travail de notre application. Un dernier chapitre « Réalisation et tests», présente l’environnement de travail et quelques exemples des interfaces de l’application développée.

Chapitre 1 : Présentation de l’organisme d’accueil

Introduction

Ce chapitre introductif est consacré à la présentation de l’organisme d’accueil. Ensuit l’organisation générale au sein de la société avec l’organigramme, les services et les domaines d’activité de la société.

1. Présentation de Powermedia

Powermedia est une agence web, située à Djerba Tunisie, spécialisée dans la création de sites web, application mobile, le référencement des sites internet et la création des logiciels sur mesure. Les services offerts sont la conception, le design et la création des sites internet compatible avec n’importe quelle plateforme mobile (ordinateur, mobile, Smartphone, tablette, …) ainsi que le référencement et l’optimisation des sites web.



**Figure 1** : Logo de la société PowerMedia

2. Services et domaines d’activité de Powermedia

Powermedia possède plusieurs domaines de compétence, spécialisée dans le développement informatique Offshore-Offshore.

Les services proposés par Powermedia sont les suivants :

* Le design graphique, application mobile, le référencement des sites internet et la création des logiciels sur mesure.
* Développement Web et d’e-marketing ainsi que développements spécifiques.
* Support technique via une hotline spécialisée.
* Consulting et Formation professionnelle.

3. Organigramme de l’organisme

## 

Conclusion

Ce chapitre nous a servi à mettre le projet dans son cadre. Après avoir présenté l’entreprise d’accueil.

Le chapitre suivant permettra d’introduire le contexte du projet ainsi que l’étude de l’existant avec les spécifications de besoins de notre projet.

Chapitre 2 : Etude Préalable

Introduction

L’étude d’un projet est une démarche stratégique qui va nous permettre d’avoir une vision globale sur ce dernier visant ainsi à bien organiser le bon déroulement du projet. Cette étude fera donc l’objet du deuxième chapitre qui sera consacré à la présentation du projet, l’objectif ciblé et l’étude de l’existant ainsi que les spécifications des besoins fonctionnels et non fonctionnels.

1. Présentation du projet :

Le Projet est une application de gestion d'école  privée qui doit gérer tout ce qui concerne les étudiants, les professeurs, le staff d'administration, le paiement des frais d’inscription et de scolarisation, le matériel, les salles et les emplois du temps.

1.1. Problématique

La gestion des données est un critère essentiel pour toute entreprise ou établissement, il se peut que ça soit une gestion de stock, gestion des ressources humaines…Pour ce projet on a opté pour les cas d’une école qui aura besoin de gérer l’ensemble de ces étudiants à travers une interface simple et pratique.

Le problème se pose au niveau de l’administration, surtout lorsque le nombre des étudiants, des professeurs et des matières augment et ça devient de plus en plus délicat de traiter tous ces données manuellement, et stocké par la suite dans l’archive de l’école.

1.2. Solution proposée

Afin de résoudre ces problèmes, l’informatisation des données devient un élément nécessaire, et le traitement automatique améliore énormément le rendement de l’établissement que ça soit au niveau du gain du temps ou au niveau de diminuer l'erreur humaine.

2. Spécifications des besoins

Nous nous intéressons aux besoins des utilisateurs traités dans notre projet à travers les spécifications fonctionnelles et non fonctionnelles pour aboutir à une solution de qualité qui répond aux besoins des différents utilisateurs.

En effet, la plateforme à réaliser doit répondre aux besoins fonctionnels suivants :

2.1. Spécifications des besoins fonctionnels :

L’application web à réaliser doit permettre à l’utilisateur de gérer un ensemble de modules, dont on trouve :

* Gestion des élèves :

Ce module contient quatre volets qui sont : l’inscription, la réinscription, les absences et les retards ainsi que l’exploitation des notes obtenues.

**a - L’inscription** : Pour une première inscription, les élèves qui veulent s'inscrire en première année, devront obligatoirement avoir l'âge réglementaire de scolarisation qui est fixé par l’administrateur.

Les parents des intéressés doivent présenter à l’école un dossier contient tous les documents nécessaires dont les photos d’identité de l’élève, la photocopie de la CIN des parents et un certificat médical. Après avoir rempli un formulaire qui comporte les informations de l’étudiant tel que son nom, tel, date naissance, son adresse et les cordonnées des parents.

Il est à noter que les parents ou tuteurs doivent régler les frais d’inscription et les frais de scolarisation.

Les élèves qui ont un niveau supérieur à la première année peuvent bénéficier aussi d’une inscription à condition d’avoir une place vacante dans le niveau souhaité et remplir le formulaire de l’inscription.

**b - La réinscription** : L’élève est en situation réglementaire, c’est-à-dire admis en classe supérieure ou ayant le droit au doublage. La réinscription se fait après la présentation des pièces demandées et après le règlement des frais d’inscription et de scolarisation.

**c - Les retards et les absences :** Le traitement des absences et retards des élèves est réglementé par le code interne de l’établissement.

Toutes les absences injustifiées font l’objet d’un courrier adressé aux parents ou par un SMS indiquant les jours d’absence de l’étudiant.

**d - Notes et matières :** L’exploitation et le traitement des notes obtenues par les élèves dans les contrôles continus visent à suivre et à mieux contrôler son évolution ou son retard scolaire est l’une des préoccupations de l’administration.

Le régime d’étude est composé de plusieurs modules, et un module contient plusieurs matières caractérisées par un nom et un volume horaire. La moyenne générale d’un module est composée de contrôle continu (30%) et un examen de fin module (70%).

* Gestion des parents :

Consiste à noter les informations relatives aux parents des élèves inscrits, tels que : l’adresse, le GSM ou le N° du fixe Afin que l'administration maintienne un contact permanent avec les parents d'élèves.

* Gestion des professeurs :

**Les vacataires** : ceux qui travaillent par heure.

**Les permanents** : sont ceux qui sont engagés mensuellement par des contrats.

La gestion des professeurs consiste à gérer leurs dossiers professionnels (diplômes, ancienneté, cv …..), ainsi que leurs absences et retards (justifiés ou non) durant l’année scolaire.

* Gestion des emplois du temps :

Ce module se présente en deux parties : emplois du temps pour les professeurs et un autre emploi du temps pour les élèves.

Celui des professeurs : l’administration établit un emploi de temps en prenant en considération la disponibilité des salles et celles des professeurs et en respectant le nombre d’heures pour chaque niveau et matière.

Celui des élèves : l’emploi du temps ne doit être chargé (maximum 6 heures/jours), il doit comporter les matières autorisées à être enseignée dans chaque niveau.

* Gestion de messagerie :

La messagerie SMS reste le moyen de communication le plus efficace. Un module de communication par SMS pour communiquer facilement avec les parents d'élèves afin de les notifier pour :

* Des absences des élèves.
* Des rappels de paiement.
* …

2.2. Spécifications des besoins non fonctionnels :

Les besoins non fonctionnels représentent un ensemble de lignes directrices et de normes liés aux performances du l’application et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d’objectifs spécifiques que doit atteindre le système.

**Maintenance** : Le code doit être facile à maintenir pour des raisons de réutilisation et de modification.

**Compatibilité et interopérabilité** : l’application doit être compatible avec tous les navigateurs Web (Google chrome, Firefox, Opéra ...) et interopérable pour toutes les plates-formes et tous les systèmes d’exploitation.

**Fiabilité** : Le système doit prendre en charge le traitement des erreurs qui peuvent survenir et informer l’utilisateur au type d’erreur et lui demander de recommencer l’opération.

**Disponibilité** : Le produit devra être disponible pour l’internaute 24/24h.

**Temps de réponse** : Le temps de réponse doit être le plus rapide possible.

**Sécurité**: Le système doit être sécurisé au niveau des données : authentification et contrôle des droits d’accès.

Conclusion

Dans ce chapitre, nous avons présenté le cadre général de notre projet et l’étude de l’existant en précisant nos solutions ainsi que nous avons exposé l’analyse et la spécification des besoins permettant de concevoir et de développer la plateforme. Après avoir fixé nos objectifs, l’étape suivante sera consacrée à une conception détaillée.

Chapitre 3 : Etude conceptuelle

Introduction

La phase de conception détermine généralement les structures et les modèles à suivre lors de l’étape d’implémentation de l’application et les transcrire d’une façon commune.

C’est dans ce stade que nous commençons par une présentation du langage de modélisation utilisé. Ensuite, nous détaillons le diagramme du contexte, les diagrammes des cas d’utilisation, les diagrammes des classes et les diagrammes de séquences.

1. Méthodes de modélisation

1.1. Langage de modélisation

Pour la conception de notre site web, nous avons adopté une méthode orientée objet. En effet cette dernière est une approche incontournable dans le cadre du développement des applications.

1. **Choix d’UML**

Pour mieux présenter l’architecture de notre site, nous avons choisi le langage de modélisation le plus adopté à savoir UML : C'est un langage de modélisation, défini comme une norme de modélisation objet qui sert à décrire de manière visuelle et graphique les besoins, les solutions fonctionnelles et techniques du projet.

1. **Les avantages de la modélisation**

* Faciliter la communication entre les différents intervenants au sein d’un projet informatique.
* Faire des simulations avant de construire un système.
* Coordonner les activités entre les différents intervenants.
* Gérer l’évolution d’un projet informatique.

1.2. Analyse du contexte

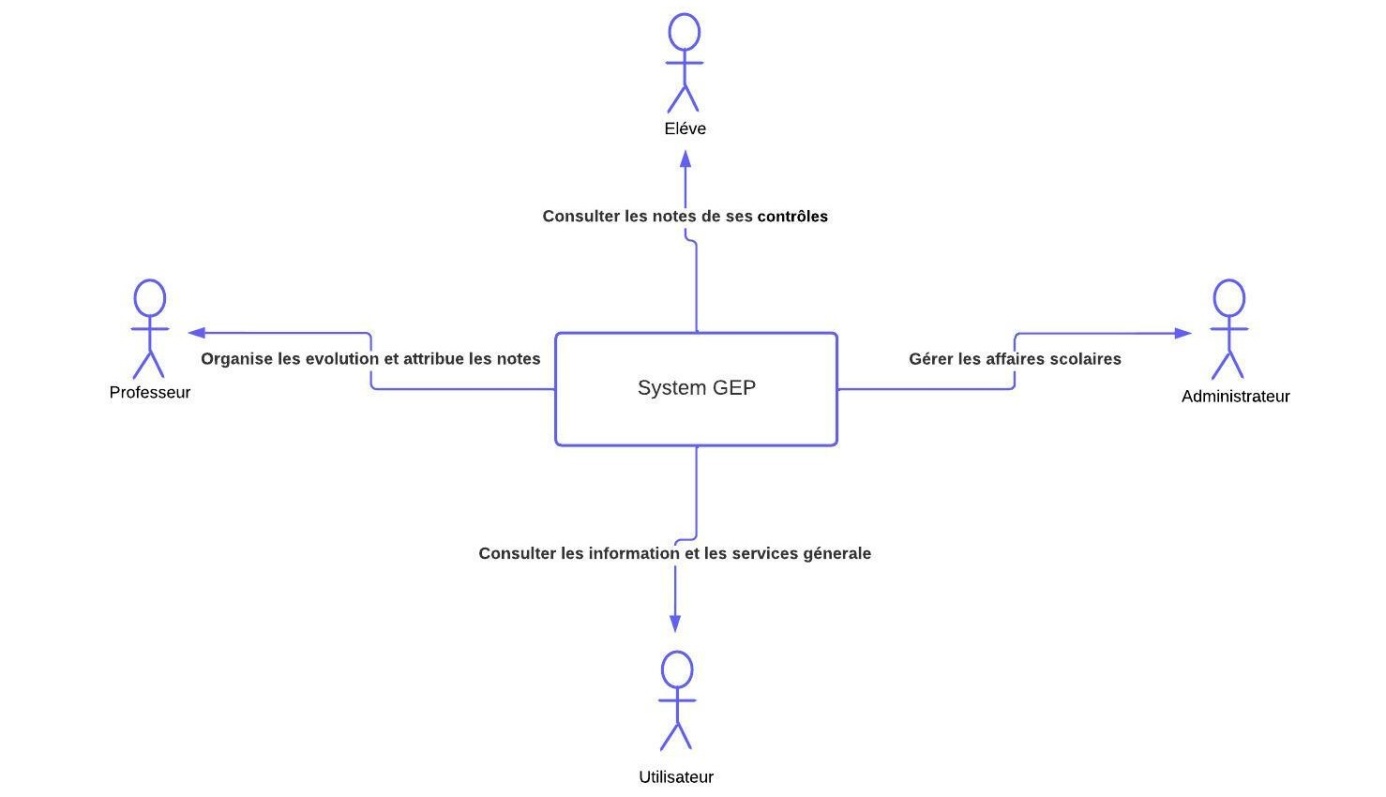
L’analyse du contexte a pour objectif l’identification des différents acteurs qui interagiront avec le système. Ces acteurs seront regroupés par la suite sous forme de rôles dont chacun englobe un ensemble d’entités. Le regroupement sera effectué en se basant sur les interactions.

1.3. Diagramme de contexte

Plutôt que de répertorier simplement les acteurs textuellement, on peut réaliser un premier diagramme que nous appelons diagramme de contexte statique [PAR 06]. Il suffit pour cela d’utiliser un diagramme de classes dans lequel chaque acteur est relié par une association à une classe centrale unique représentant le système.

Le diagramme de contexte a pour but de représenter les flux d’informations entre le système et les acteurs externes.

GEP pour gestion d’école privée.

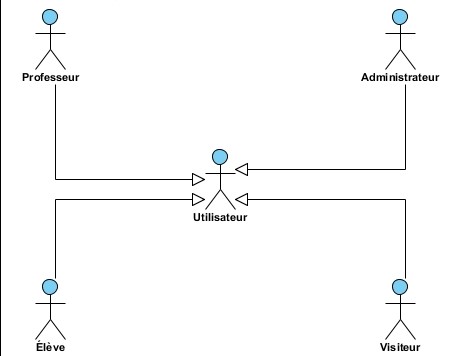


**Figure 2:** Diagramme de contexte.

3. Identification des acteurs et cas d’utilisation

3.1. Diagramme des acteurs

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Le diagramme d’acteurs permet de déterminer les acteurs du système.



**Figure 3:** Diagramme des acteurs

3.2. Cas d’utilisation

Un cas d'utilisation décrit une fonction qu'un système exécute pour atteindre l'objectif de l'utilisateur. Un cas d'utilisation doit renvoyer un résultat observable qui est utile pour l'utilisateur du système.

Les acteurs et les cas d’utilisation sont résumés dans le tableau suivant :

|  |  |
| --- | --- |
| Acteurs | Description fonctionnelle |
| **Visiteur** | * Consulter les informations et les services généraux. * Préinscrire |
| **Élève** | * Consulter son emploi du temps. * Voir ses absences. * Consulter les notes de ses contrôles * Consulter les matières. * Consulter liste des classes. * Consulter liste des exams. |
| **Professeur** | * Consulter son emploi du temps * Consulter la liste de ses élèves. * Organiser et insérer les évaluations. * Attribuer les notes aux élèves * Marquer les présences et les absences |
| **Administrateur** | * Gestion des élèves. * Gestion des parents. * Gestion des professeurs. * Gestion du paiement. |

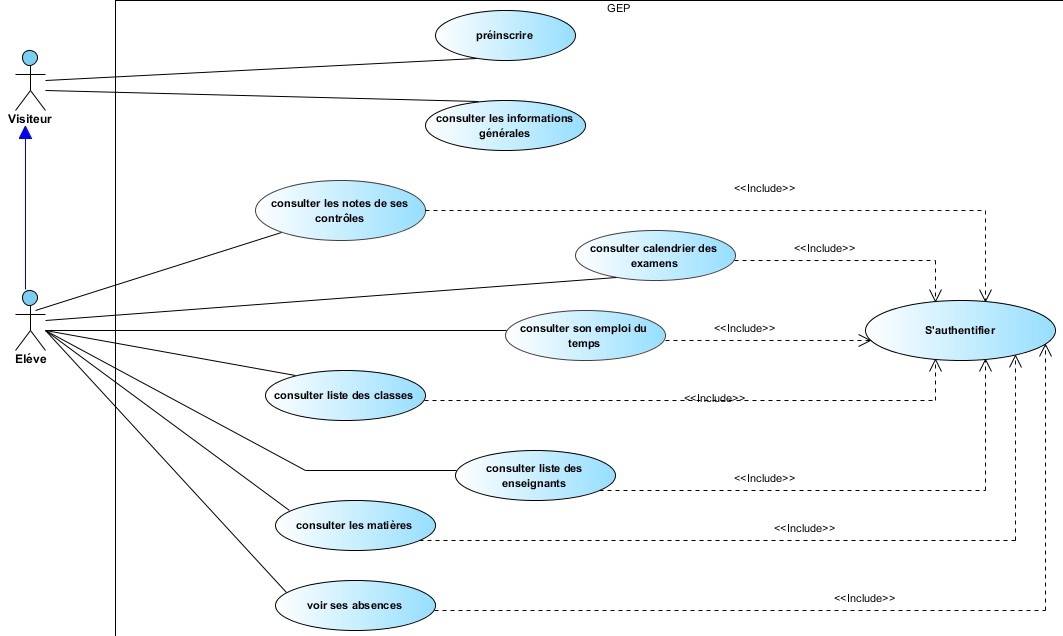
**Tableau 1: les acteurs et les cas d’utilisations**

3.3. Diagramme de cas d’utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

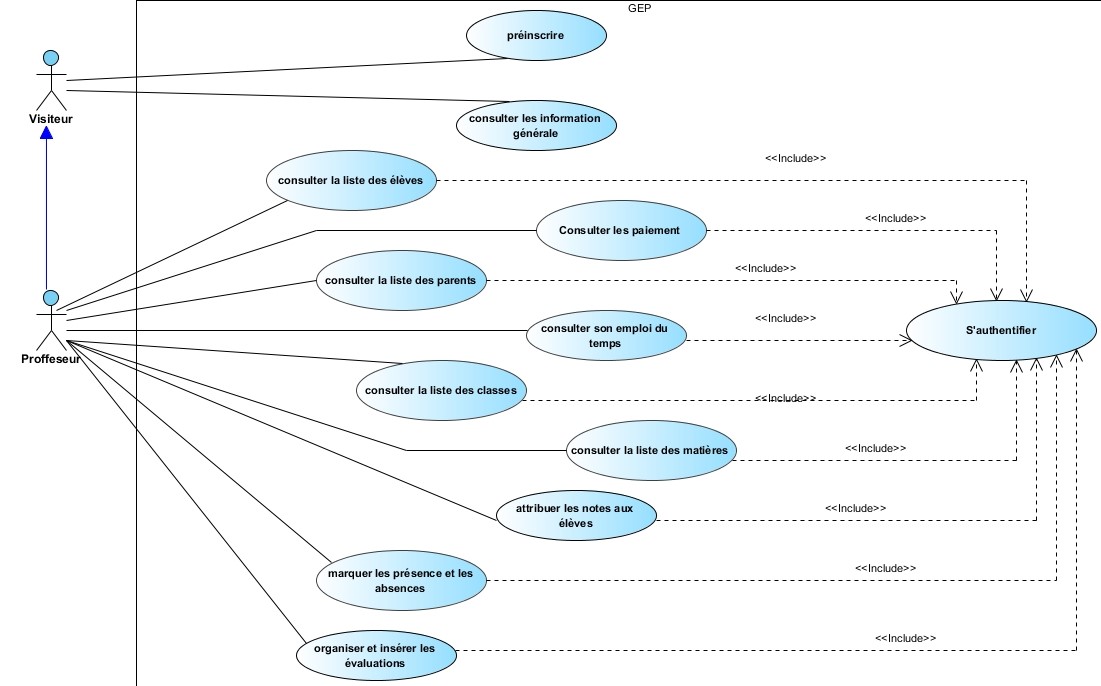
Dans les figures qui suivent, nous présenterons les cas d’utilisation qui mettent en évidence les fonctionnalités de chaque acteur dans le système.

1. **Diagramme de cas d’utilisation d’acteur «Élève»**



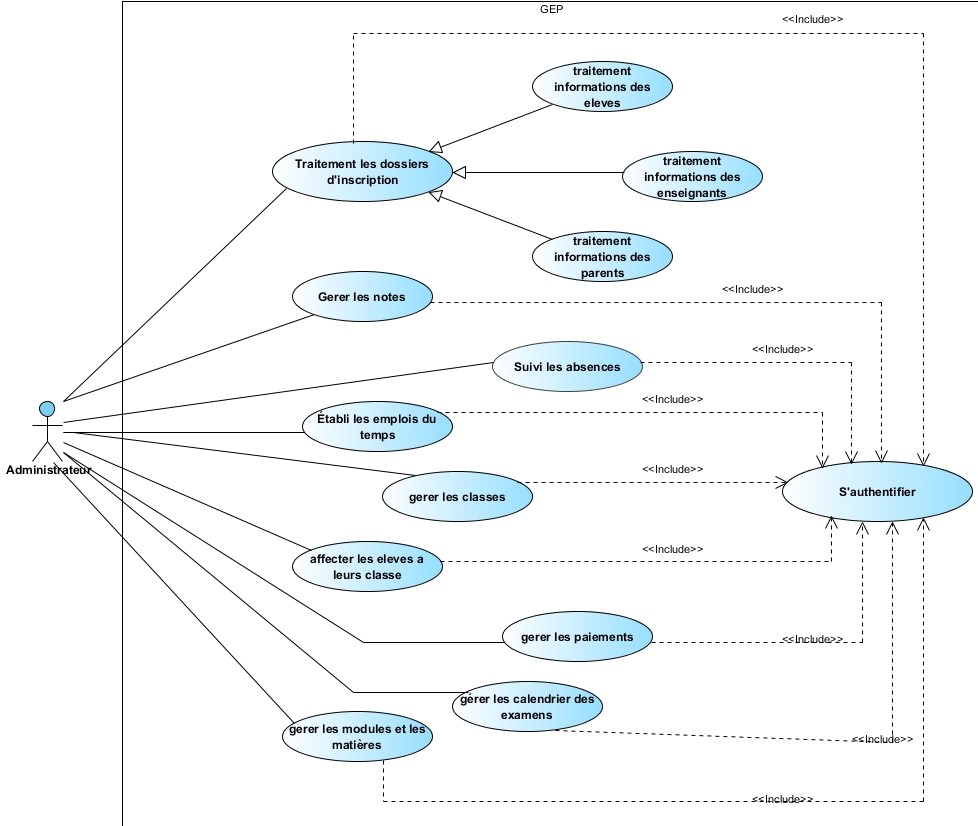
**Figure 4**: Diagramme du cas d’utilisation d'acteur élève

1. **Diagramme de cas d'utilisation d'acteur «Professeur»**



**Figure 5**: Diagramme du cas d'utilisation d'acteur professeur

1. **Diagramme de cas d'utilisation d'acteur «Administrateur»**

****

**Figure 6**: Diagramme du cas d'utilisation d'acteur Administrateur

En ce qui suit nous décrivons quelques cas d’utilisation par une fiche décrivant les divers scénarios de leur exécution.

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation** : Authentification | | |
| **Objectif** | Permettre à chaque utilisateur de l’application, de s’authentifier vers ses rôles, à travers un login et mot de passe. | |
| **Acteurs** | Utilisateur – système | |
| **Enchainements** : | **Scénario nominal** | 1. Saisie le login et mot de passe.  2. Envoyer login et mot de passe.  3. Traitement des informations envoyées.  4. Redirection d’utilisateur vers son rôles. |
| **Scénario d’erreur** | Si le nom d’utilisateur ou le mot de passe est erroné, la plateforme affiche un message d’erreur. |
| **Pré-condition** | Compte existant. | |
| **Post-condition** | Autorisation d’accès accordée. | |

**Tableau 2: Fiche de description « Authentification »**

**Fiche de description «Inscription d’un élève » :**

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation** : Inscription d’un élève. | | |
| **Objectif** | Permettre à l’administration d’ajouter toutes les informations concernant un élève, y compris l’état civil, les informations des parents, les informations complémentaires, les informations sur la santé et enfin l’affectation d’élève dans une classe. | |
| **Acteurs** | Utilisateur, administrateur. | |
| **Enchainements** : | **Scénario nominal** | 1. Saisie les informations de l’élève.  2. Validation de la saisie.  3. Traitement des informations envoyées.  4. l’inscription est effectuée avec succès avec redirection d’utilisateur vers la liste d’élèves. |
| **Scénario d’erreur** | En cas d’une erreur, l’inscription est rejetée on précisant l’erreur effectuée. |
| **Pré-condition** | 1. Authentification.  2. L’inscription doit répond aux conditions d’inscription établie par l’administration. | |
| **Post-condition** | 1. Redirection d’utilisateur vers la liste d’élèves.  2. Mise à jour de base de données. | |

**Tableau 3: Fiche de description « inscription éleve »**

**Fiche de description «inscription d’un professeur » :**

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation :** Ajouter professeur. | | |
| **Objectif** | Permettre à l’administration d’ajouter toutes les informations concernant le professeur, y compris les informations personnelles, et la possibilité d’ajouter les diplômes obtenus par le professeur. | |
| **Acteurs** | Utilisateur, administrateur. | |
| **Enchainements** : | **Scénario nominal** | 1. Saisie les informations du professeur.  2. Validation de la saisie.  3. Traitement des informations envoyées.  4. L’inscription est effectuée avec succès avec redirection d’utilisateur vers la liste des professeurs. |
| **Scénario d’erreur** | En cas d’une erreur, l’inscription est rejetée on précisant l’erreur effectuée.. |
| **Pré-condition** | 1. Authentification.  2. L’inscription doit répond aux conditions d’inscription établie par l’administration. | |
| **Post-condition** | 1. Redirection d’utilisateur vers la liste des professeurs.  2. Mise à jour de base de données. | |

**Tableau 4: Fiche de description « inscription professeur »**

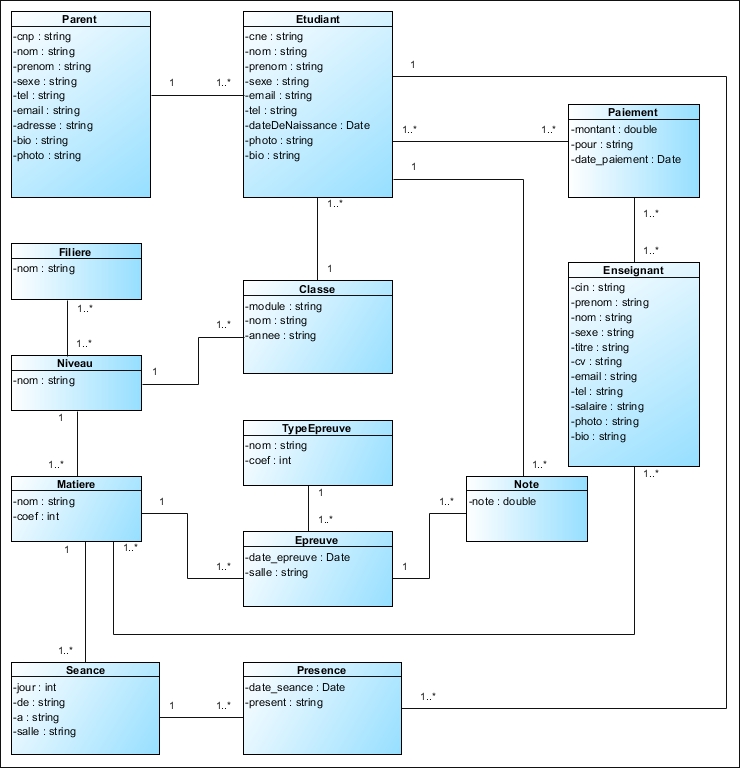
4. Diagramme de classes

Le diagramme de classes [LAA 08] est généralement considéré comme le plus important dans un développement orienté objet. Il représente l’architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtage conceptuel (héritage) ou une relation organique (agrégation).

La page suivante présente la figure diagramme de classes de notre application. Elle regroupe les classes parent, étudiant, enseignant, classes, niveaux, séances...

Nous décrivons quelques classes que le système utilise ainsi que les liaisons entre eux.

Une filière comporte plusieurs niveaux, et chaque niveau comporte un ou plusieurs classes. Un enseignant donne cours en plusieurs matières et une matière comporte un ou plusieurs épreuves et chaque épreuve comporte un ou plusieurs notes.

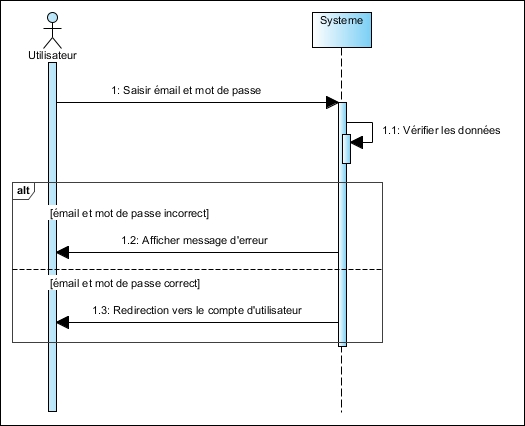


**Figure 7** : Diagramme de classes

5. Diagramme de séquence

Pour mieux concrétiser les interactions entre les acteurs du système et la base de données, nous sommes amenés à traduire nos scénarios en diagrammes de séquence. Les diagrammes de séquence servent à illustrer les cas d’utilisation. Ils permettent de représenter des collaborations entre les objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois des messages.

5.1. Diagramme de séquence « authentification »

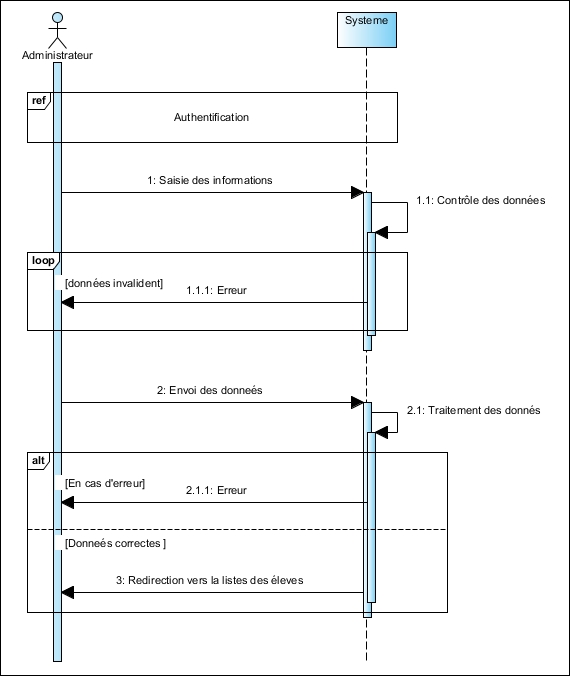


**Figure 8 :** Diagramme de séquence « s’authentifier »

* **Fiche de description**  « authentification» :
* **Titre :** Authentification
* **Objectif :** Permettre à chaque utilisateur de l’application, de s’authentifier vers son compte, à travers un login et mot de passe.
* **Scenario :**

1. L'utilisateur saisit son adresse email et son mot de passe.
2. Le système vérifie les données saisies et envoie une requête d'authentification à la base de données.
3. Si les données correspondent à un compte valide la connexion sera établie.
4. Sinon un message d'erreur sera affiché.

5.2. Diagramme de séquence « inscription élève »

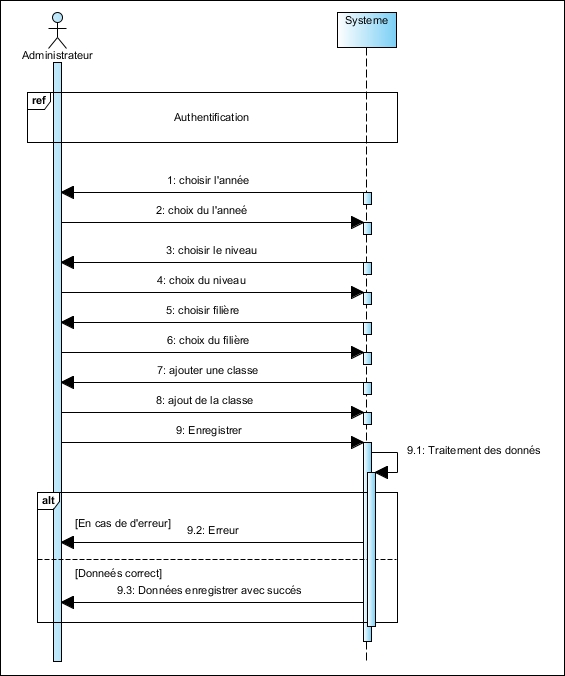


**Figure 9** : Diagramme de séquence « inscription élève »

* **Fiche de description** « Inscription élève»
* **Objectif :** Permettre à l’administration d’ajouter toutes les informations concernant un élève
* **Scénario :**

1. Saisie les informations de l’élève.
2. Validation de la saisie.
3. Traitement des informations envoyé.
4. En cas d’une anomalie, l’inscription est rejetée on précisant l’erreur effectuée.
5. Si non, l’inscription est effectuée avec succès.

5.3. Diagramme de séquence « Ajouter classe »



**Figure 10** : Diagramme de séquence« Ajouter classe »

* **Fiche de description** « Ajouter classe »
* **Objectif** : Permettre à l’administration d’affecter des classes à chaque niveau.
* **Scenario**:

1. Choisir le niveau.
2. Choisir la filière.
3. Ajouter une classe et valider les choix.
4. En cas d’une erreur, l’ajout est rejeté en précisant l’erreur effectuée.
5. Si non, l’ajout est effectué avec succès.

Conclusion

Dans ce chapitre, nous avons présenté les différents aspects conceptuels de notre projet. Nous avons commencé par introduire les différents diagrammes de cas d’utilisation de notre plateforme en expliquant le fonctionnement de chacun des cas d’utilisation, par la suite nous présentons le diagramme de classes et les diagrammes de séquences les plus importants.

Dans le chapitre suivant, nous détaillons la phase de réalisation de notre projet.

Chapitre 4 : Réalisation

Introduction

Après avoir achevé l’étape d’analyse et conception de l’application, on va entamer dans ce chapitre la partie réalisation

1. Environnement de travail

Dans cette partie, nous présentons les différents outils logiciels utilisés pour développer notre solution.

1.1. Environnement de développement

1. **Définition d’un Framework**

Un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle, et accompagné du plan de l'architecture cible du Framework.

1. **Le Framework Laravel**

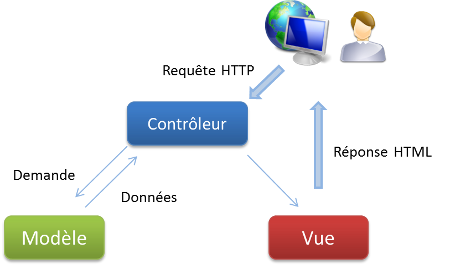
Laravel est un Framework web gratuit, open-source, créé par Taylor Otwell et destiné au développement d'applications Web suivant le modèle-vue-contrôleur (MVC). Laravel est un système modulaire ou système d'emballage avec un gestionnaire de dépendances dédié, pour accéder à différentes façons de bases de données relationnelles, les services publics que l'aide au déploiement d'applications et de maintenance.



**Figure 11** : Logo de Framework Laravel

1. **Le modèle MVC**

Le pattern modèle-vue-contrôleur (en abrégé MVC, de l'anglais model-view-controller), est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.



**Figure 12** : Pattern MVC

Une application conforme au motif MVC comporte trois types de modules : les modèles, les vues et les contrôleurs.

* Modèle : Élément qui contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement. Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application. Pour notre application, le modèle représente des étudiants, des enseignants, des classes, des matières, des séances…
* Vue : La Vue retourne une présentation des données venant du model. Etant séparée par les Objets Model, elle est responsable de l’utilisation des informations dont elle dispose pour produire une interface de présentation de votre application.
* Contrôleur : Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante.

Il effectue la synchronisation entre le modèle et les vues. Il est la couche qui se charge d'analyser et de traiter la requête de l'utilisateur. Le contrôleur contient la logique de notre application et va se contenter « d'utiliser » les autres composants : les modèles et les vues. Concrètement, un contrôleur va récupérer, par exemple, les informations sur l'utilisateur courant, vérifier qu'il a le droit de modifier un tel outil, récupérer les données de cet outil et demander la page du formulaire d'édition de l’outil [3].

* La raison pour laquelle on a choisi le MVC : Nous avons choisi cette méthode de conception parce qu’elle répond au besoin de notre projet. L’avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Nous avons intérêt à maintenir une grande indépendance entre L'interface utilisateur, La logique applicative et La source de données. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet si l'une des entités change, les deux autres n'aient pas à changer.

1.2. Technologies utilisées

Dans cette partie nous allons présenter les technologies choisies pour la réalisation de notre projet.

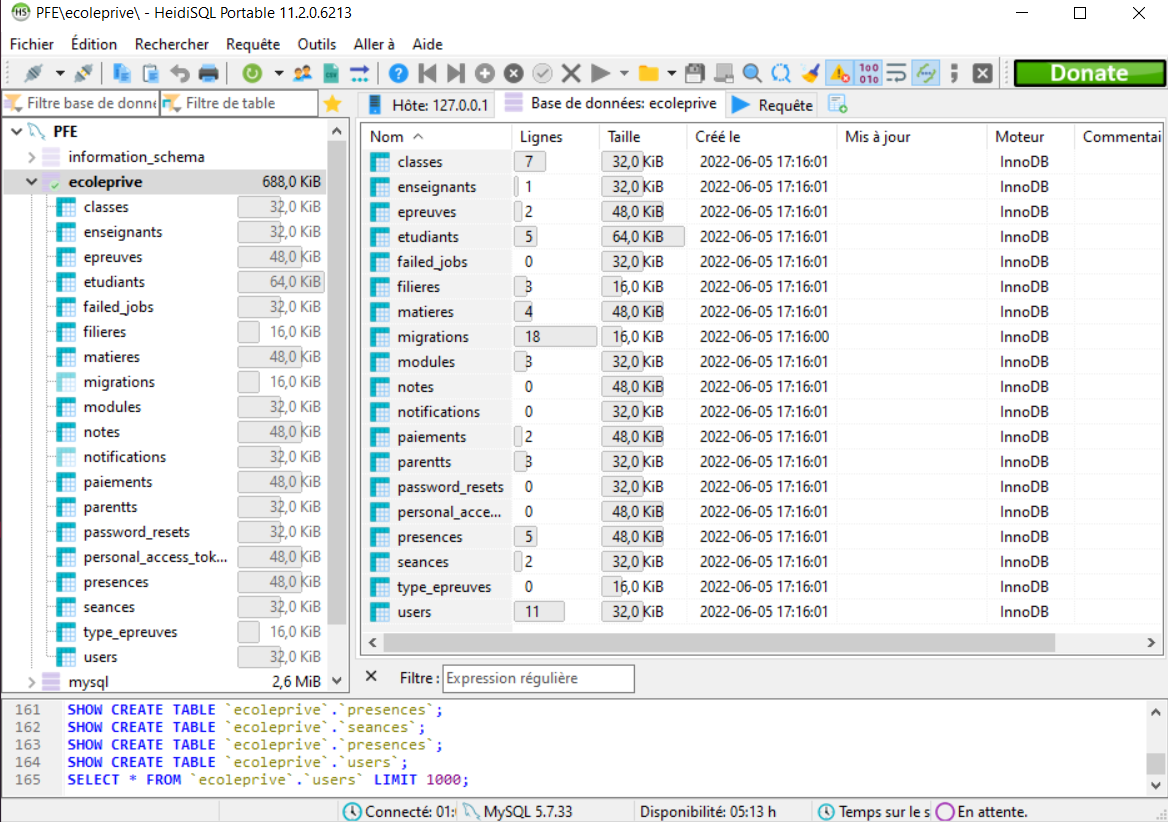
**a - Visual Studio Code :** Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C++, C#, Java, Python, PHP, Go) et des runtimes (tels que .NET et Unity) [4].

**b - Visuel Paradigm :** Visuel Paradigm est un outil de conception et de gestion puissant, multiplateforme et pourtant facile à utiliser pour les systèmes informatiques. Visual Paradigm fournit aux développeurs de logiciels la plate-forme de développement de pointe pour créer des applications de qualité plus rapidement, mieux et moins cher ! Il facilite une excellente interopérabilité avec les autres outils CASE et la plupart des principaux IDE, ce qui excelle dans l'ensemble de votre processus de développement Modèle-Code-Déploiement dans cette solution à guichet unique [5].

**c - Laragon :** Laragon est un environnement de développement universel portable, isolé, rapide et puissant pour PHP, Node.js, Python, Java, Go, Ruby. Il est rapide, léger, facile à utiliser et facile à étendre. Laragon est idéal pour créer et gérer des applications Web modernes. Il est axé sur la performance - conçu autour de la stabilité, de la simplicité, de la flexibilité et de la liberté.

**d - SGBD :** Un système de gestion de [base de données](https://www.oracle.com/fr/database/definition-base-de-donnees.html) (SGBD) est le logiciel qui permet à un ordinateur de stocker, récupérer, ajouter, supprimer et modifier des données. Un SGBD gère tous les aspects primaires d'une base de données, y compris la gestion de la manipulation des données, comme l'authentification des utilisateurs, ainsi que l'insertion ou l'extraction des données. Un SGBD définit ce qu'on appelle le schéma de données ou la structure dans laquelle les données sont stockées [6].

Le SGBD que j’ai utilisé est **HeidiSQL.** HeidiSQLest un logiciel libre, et a pour objectif d’être facile à apprendre. Heidi vous permet de voir et de modifier des données et des structures a partir d’ordinateurs exécutent l’un des systèmes de base de données MaraiDB, MySQL… HeidiSql fait partie des outils les plus populaires pour MariDB et MySQL ans le monde [7]. La figure ci-dessous représente l’interface utilisateur de HeidiSQL.



**Figure 13** : HeidiSQL

Le modèle physique de notre base de données de l’application est le suivant :

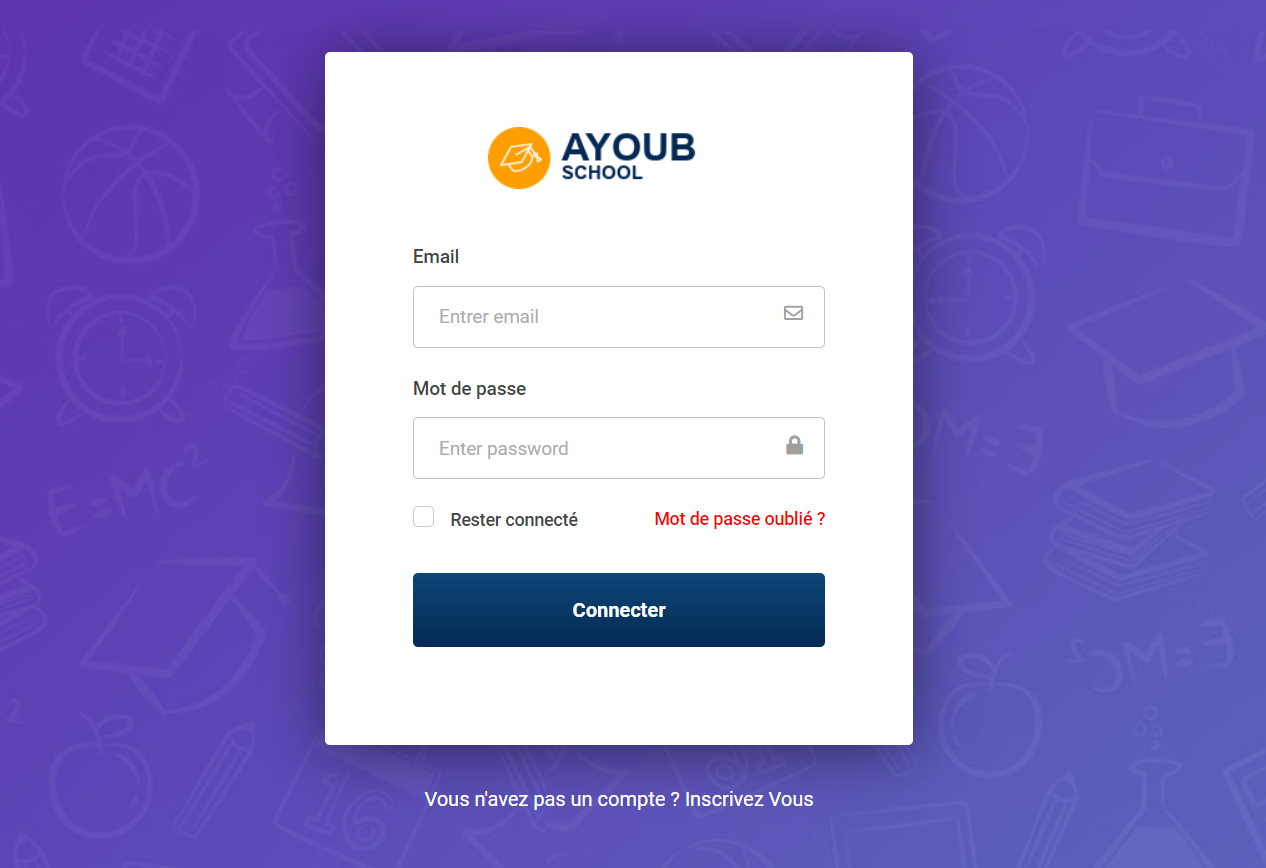
* Elève (id, cne, nom, prenom, sexe, date, email, tel, adresse, bio, image, #parent\_id, #classe\_id)
* Enseignants (id, cin, nom, prenom, sexe, titre, email, tel, adresse, salaire, bio, cv, image, #user\_id)
* Parents (id, cnp, nom, prenom, sexe, occupation, email, tel, adresse, salaire, bio, cv, image, #user\_id)
* Filiere (id, nom)
* Classes (id, annee, nom, #filiere\_id)
* Modules (id, #classe\_id)
* Matieres (nom, coef, #modules\_id, #enseignants\_id)
* Note (id, note, #epreuve\_id, #etudiant\_id)
* Seances (id, jour, de, a, salle, #matiere\_id)
* Presences (id, date\_seance, present, #seance\_id, #etudiant\_id)

2. Modules réalisés

Dans cette partie, nous allons présenter quelques interfaces de l’application, répondant aux recommandations de compatibilité, de guidage, de clarté, d’homogénéité et de souplesse. Nous avons choisi l’administrateur comme utilisateur vu qu’il présente la majeure partie des principales fonctionnalités de l’application.

2.1. Page authentification

Cette page est la première de notre application, elle permet aux différents utilisateurs de l'application d'accéder à leur espace. Pour s'authentifier, l'utilisateur donne son émail et son mot de passe et si les deux informations sont correctes alors il accède à son espace si non le système lui demande de saisir les informations correctes.

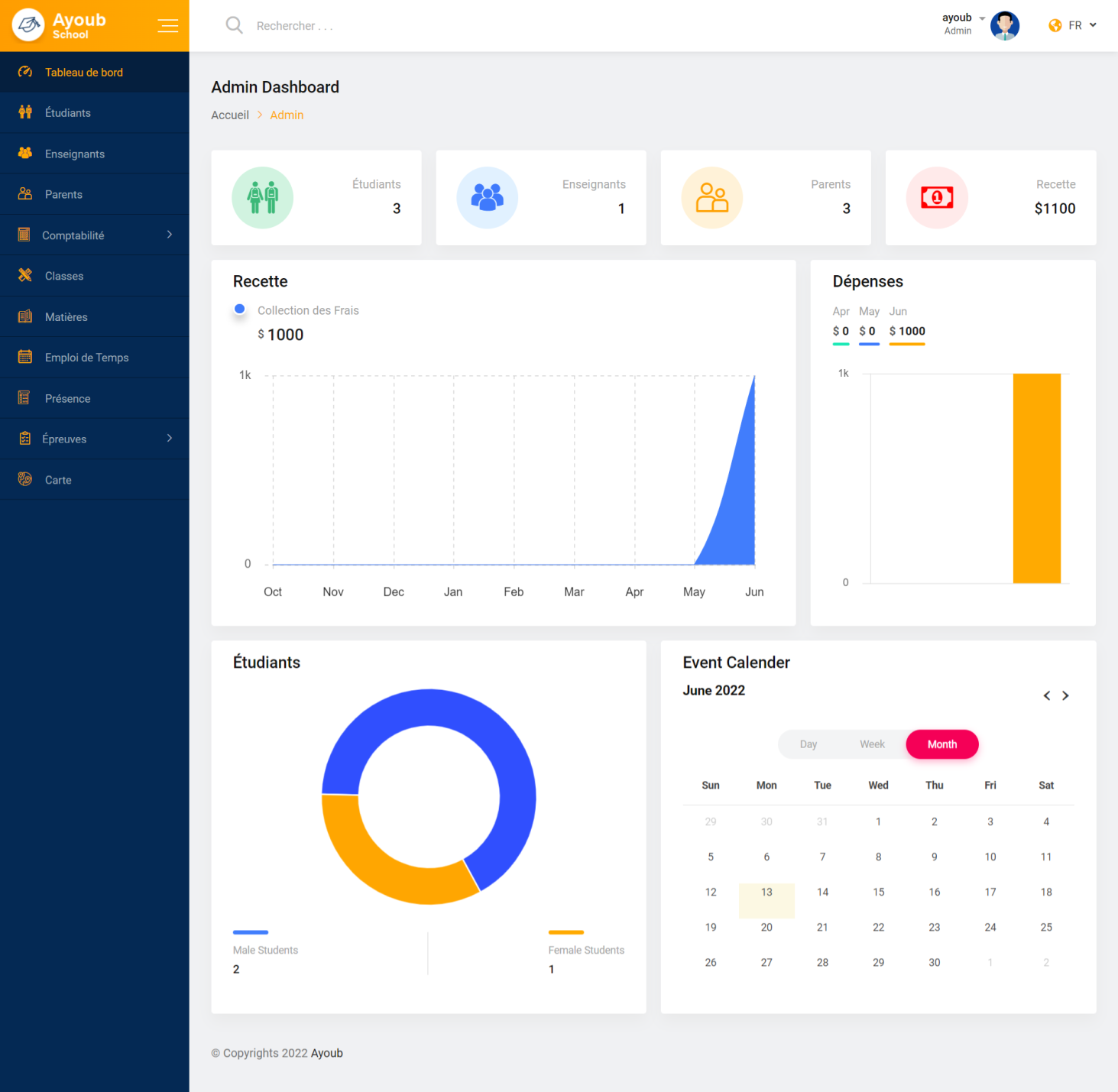


**Figure 14** : Page authentification

2.2. Page tableau de bord

Après l'authentification, l’utilisateur est rédigé vers son tableau de bord, dans lequel on trouve tout les informations et les fonctionnalités importantes sous forme des courbes et du diagramme circulaire dynamique.

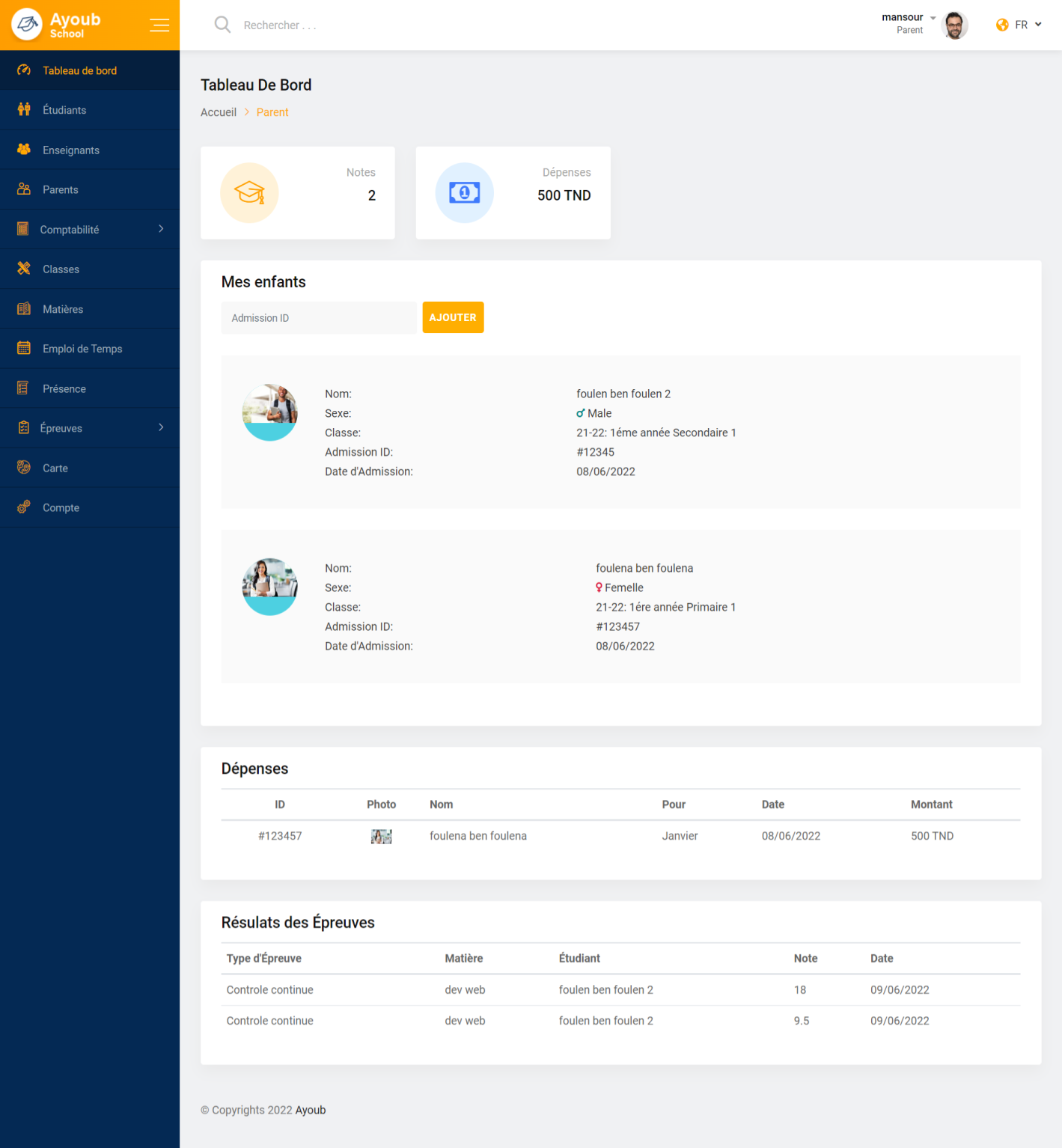
1. Tableau de bord « Administrateur »



**Figure 15** : Page tableau de bord « Administrateur »

1. Tableau de bord « Parent »

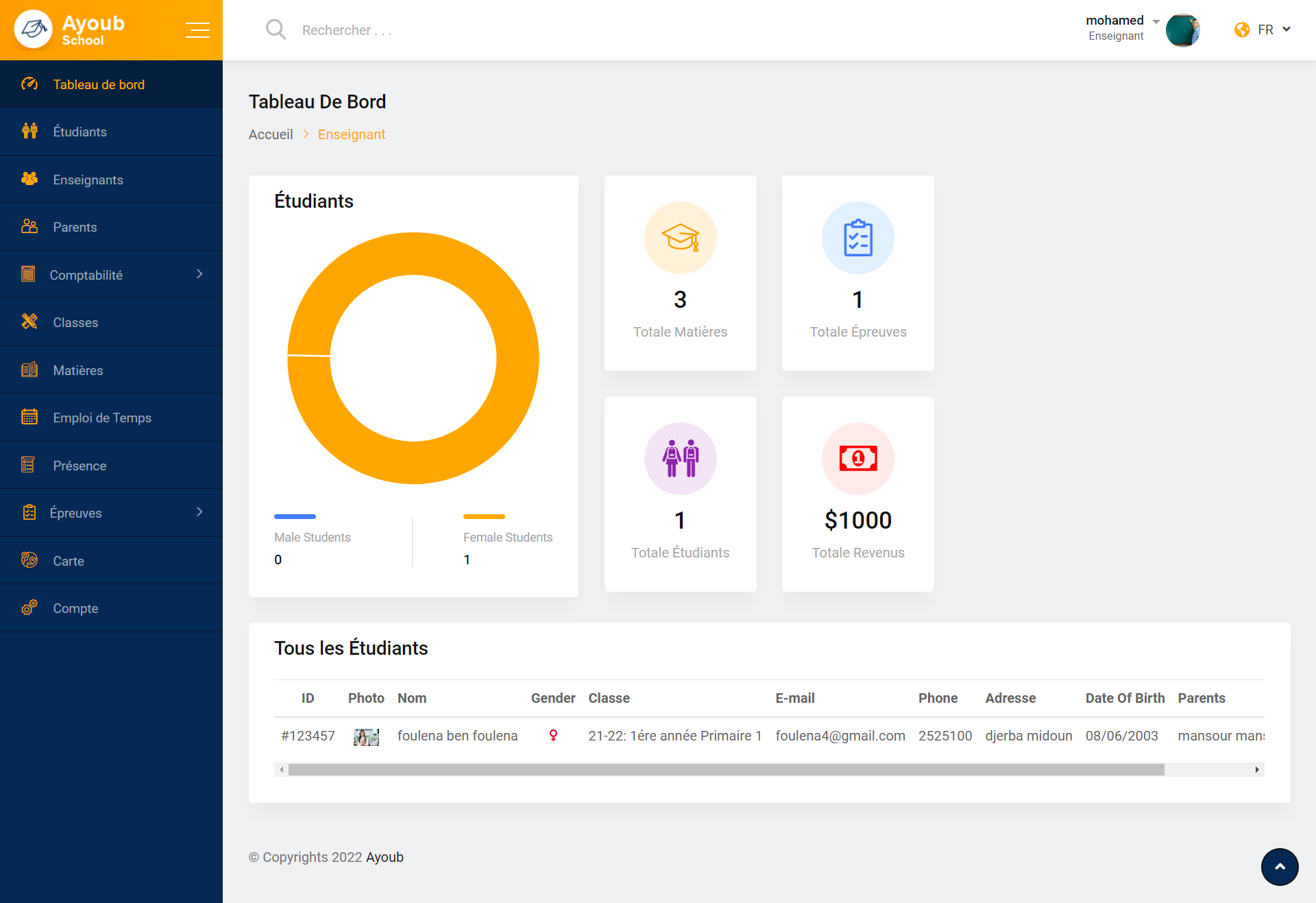
Après que le parent ajouter ses enfants a travers l’interface  (Admission), il peut voir tout les informations relatives a ses enfants, les notes, le résultat des épreuves, l’absence et le paiement.



**Figure 16** : Page tableau de bord « Parent»

1. Tableau de bord Enseignant « Enseignant »

Dans le tableau de bord de l’enseignant on trouve tous les informations relatives à ses étudiants, les matières qu'il enseigne, les épreuves et son salaire.

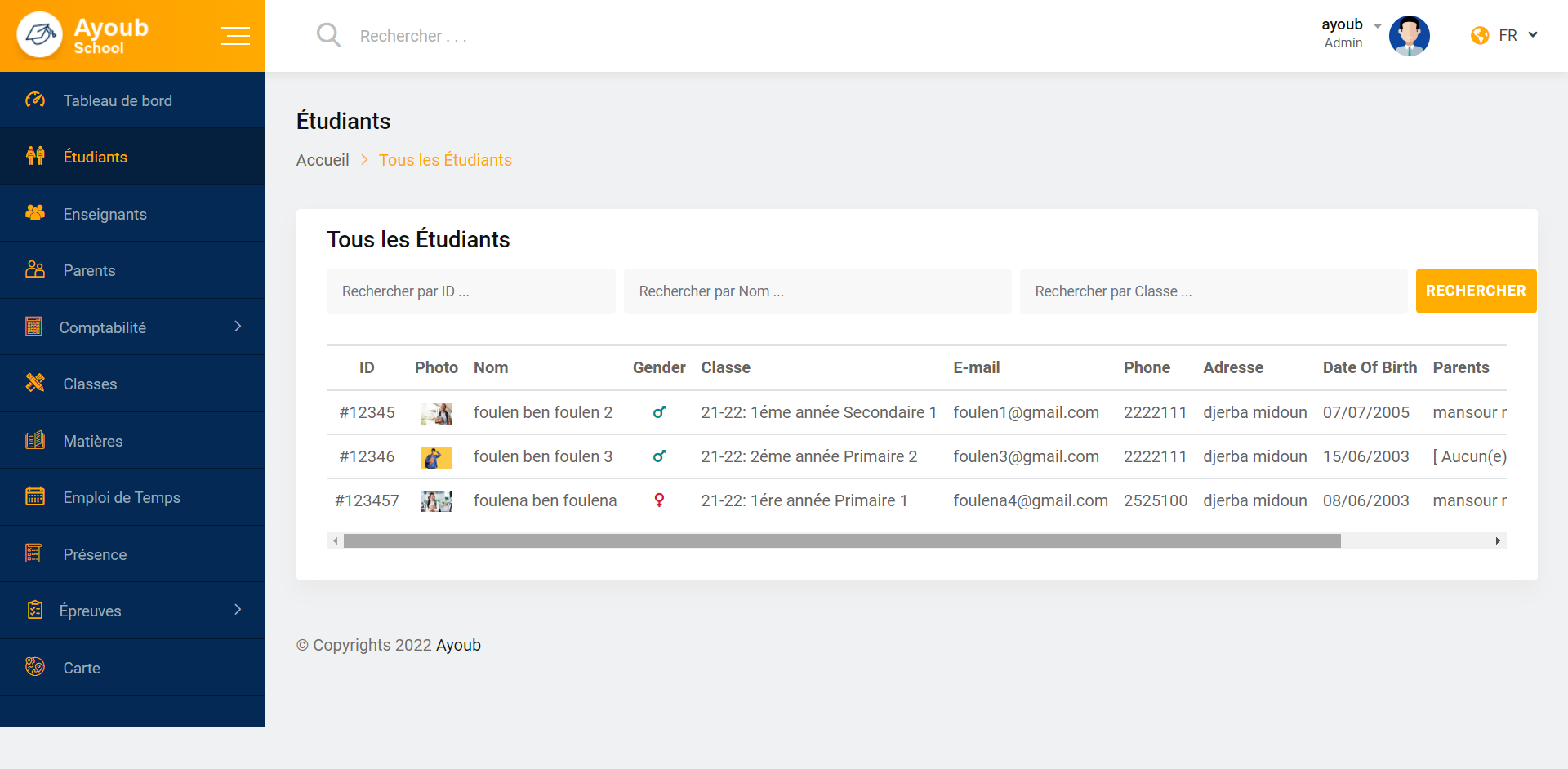


**Figure 17 :** Page tableau de bord « enseignants»

2.3. Page liste des étudiants

Cette interface contient l’ensemble des élèves, avec un certain nombre d’options :

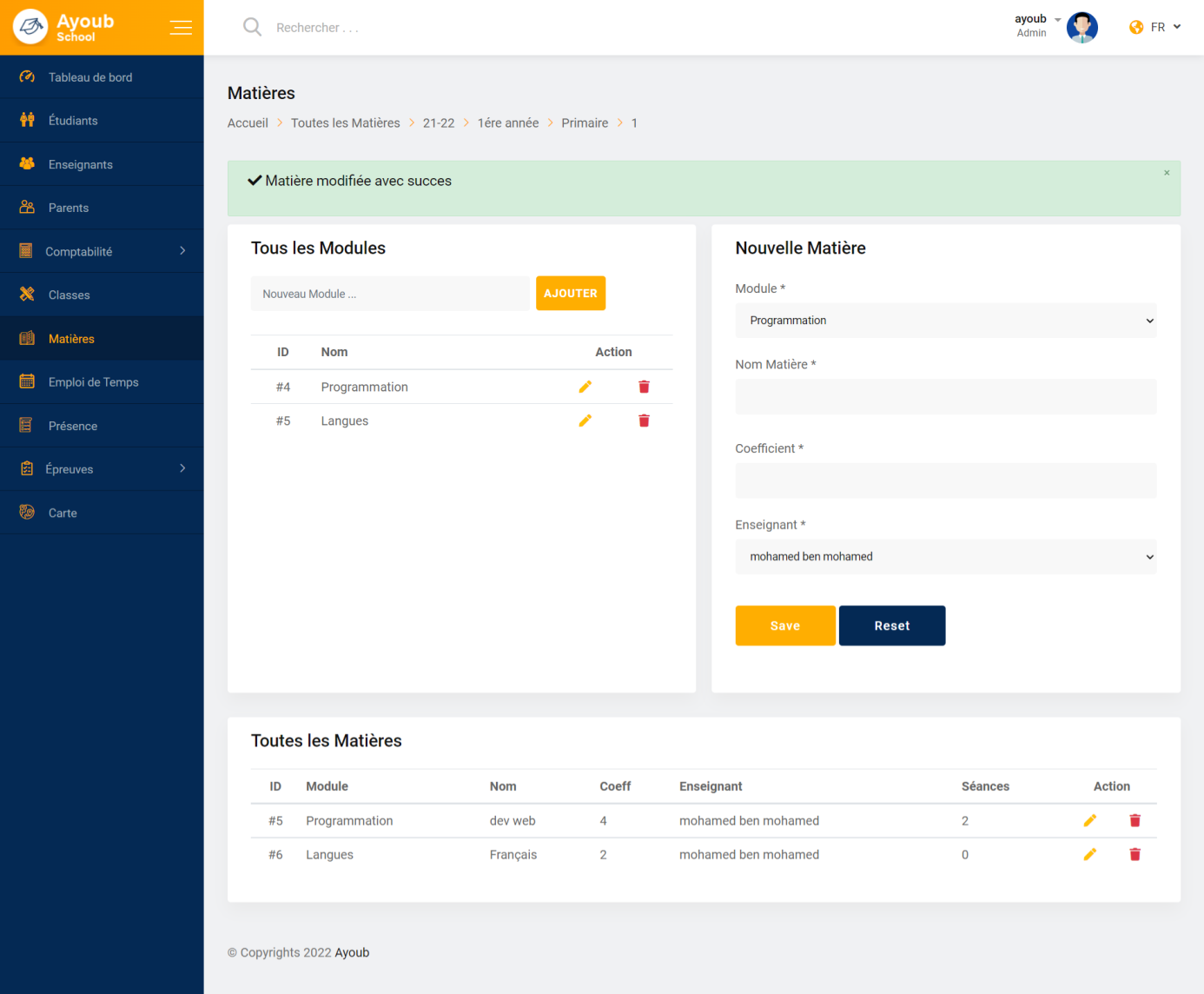
* Possibilité de filtrage par : id – classe – nom.
* La recherche rapide.
* La possibilité de voir toutes les informations concernant un élève.
* Modifier les données d’un élève.
* Supprimer un élève



**Figure 18 :** Page liste des étudiants

2.4. Page ajouter des matières

Dans cette interface on peut ajouter des matières, on précisant le module et le coefficient.



**Figure 19** : Page ajouter matières

Conclusion

Au cours de ce chapitre, nous avons décrit les outils matérielles et logicielles avec lesquelles nous avons développé notre plateforme. Nous avons ensuite présenté notre application à travers quelques interfaces que nous avons développées.

Conclusion générale

L’objectif de notre projet de fin d’études était de concevoir et implémenter une application de gestion d’école privée.

Le point de départ de la réalisation de ce projet était une récolte d’informations nécessaires pour dresser un état de l’existant, qui nous ont permis par la suite de répondre à notre problématique, tout en spécifiant les exigences fonctionnelles et non fonctionnelles. Par la suite, nous nous sommes intéressés à l’analyse et la spécification des besoins qui nous a permis de distinguer les différents acteurs interagissant avec l’application visée.

L’objectif de la partie suivante était la conception détaillée, dans laquelle nous avons fixé la structure globale de l’application. Le dernier volet de notre projet était la partie réalisation qui a été consacrée à la présentation des outils du travail et les interfaces les plus significatives de notre application.

L’apport de ce travail a été d’une importance très considérable. En effet, il nous a permis : de suivre une méthodologie de travail bien étudiée, d’approfondir nos connaissances dans le monde de développement des applications et de bien nous maitrisons sur le Framework Laravel.

Ce projet est un noyau sur lequel peuvent s’étendre plusieurs perspectives pour l’enrichir et le développer de projet.

Bibliographie

**[PAR 06]** 1 Uml 2 par la pratique, **Pa**scal **R**ogues, Eyrolles, 20**06**

**[LAA 08]** 2 Uml 2 **La**urent **A**udibert, Eyrolles, 20**08**

**[3]** <https://book.understanding-model-view-controller.html>

**[4]**  <https://code.visualstudio.com/docs>

**[5]**  <https://www.visualparadigm.com/support/documents/visualparadi.html>

**[6]** <https://www.oracle.co/systeme-gestion-base-de-donnees-sgbd-> definition.html

**[7]** https://www.heidisql.com/

**Sommaire**

[Introduction générale 1](#_Toc106064947)

[Chapitre 1 : Présentation de l’organisme d’accueil 2](#_Toc106064948)

[Introduction 3](#_Toc106064949)

[1. Présentation de Powermedia 3](#_Toc106064950)

[2. Services et domaines d’activité de Powermedia 3](#_Toc106064951)

[3. Organigramme de l’organisme 4](#_Toc106064952)

[Conclusion 4](#_Toc106064953)

[Chapitre 2 : Etude Préalable 5](#_Toc106064954)

[Introduction 6](#_Toc106064955)

[1. Présentation du projet : 6](#_Toc106064956)

[1.1. Problématique 6](#_Toc106064957)

[1.2. Solution proposée 6](#_Toc106064958)

[2. Spécifications des besoins 6](#_Toc106064959)

[2.1. Spécifications des besoins fonctionnels : 7](#_Toc106064960)

[2.2. Spécifications des besoins non fonctionnels : 9](#_Toc106064961)

[Conclusion 9](#_Toc106064962)

[Chapitre 3 : Etude conceptuelle 10](#_Toc106064963)

[Introduction 11](#_Toc106064964)

[1. Méthodes de modélisation 11](#_Toc106064965)

[3. Identification des acteurs et cas d’utilisation 13](#_Toc106064966)

[4. Diagramme de classes 19](#_Toc106064967)

[5. Diagramme de séquence 20](#_Toc106064968)

[Conclusion 24](#_Toc106064969)

[Chapitre 4 : Réalisation 25](#_Toc106064970)

[Introduction 26](#_Toc106064971)

[1. Environnement de travail 26](#_Toc106064972)

[2. Modules réalisés 30](#_Toc106064973)

[Conclusion 36](#_Toc106064974)

[Conclusion générale 37](#_Toc106064975)

[Bibliographie 38](#_Toc106064976)