

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaïd - Tlemcen

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du Diplôme de Master en Informatique

Option : Système d'information et des connaissances (SIC)

Thème :

**Plateforme Intelligente de Gestion Scolaire : Intégration de l'Analyse
Prédictive et de l'IA pour une Éducation Personnalisée**

Réalisé par :

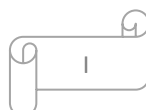
TRAORE Moussa

HADJ MAHAMMED Smail

Soutenu publiquement, le 03/06/2025, devant le jury composé de :

- Dr. BELABED Amine (Président).
- Dr. CHAOUICHE RAMDANE Fatima Houbaba Lamia (Examinatrice).
- Dr. CHOUITI Sidi Mohammed (Encadrant).

Année universitaire : 2024-2025



Dédicaces

Je dédie ce travail :

À mes parents, qui ont tout donné pour me créer les conditions idéales tout au long de mon parcours académique. Leur soutien sans faille a été le pilier de ma réussite.

À toute ma famille, proche comme lointaine, pour leur présence bienveillante et leurs encouragements constants qui ont illuminé mon chemin.

Traore Moussa

À ceux qui ont façonné la personne que je suis aujourd'hui...

À ma mère, source inépuisable de tendresse, de patience et de courage. Ton amour discret mais puissant a été le fil conducteur de mon parcours. Chaque étape franchie porte la trace de tes sacrifices silencieux et de ta foi inébranlable en moi. Ce travail est le reflet de ton soutien constant. Merci.

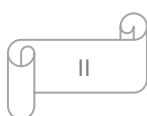
À mon père, exemple de rigueur, de persévérance et d'intégrité. Par tes conseils, tu m'as transmis le goût de l'effort, la force de continuer et la fierté d'atteindre mes objectifs. Ce chemin n'aurait pas été le même sans ta présence discrète mais essentielle.

À ma famille, ma force dans les moments de doute. Votre soutien, vos encouragements et votre affection ont toujours su me redonner confiance. Ce travail vous appartient autant qu'à moi.

À ceux qui m'ont accueilli comme un des leurs, m'offrant leur bienveillance, leur écoute et leur générosité. Vous avez été une lumière précieuse dans les périodes les plus sombres, et une motivation supplémentaire à persévérer.

Je dédie aussi ce travail à toutes les personnes qui, de près ou de loin, ont contribué à mon parcours. À ceux qui m'ont inspiré, soutenu, guidé ou simplement cru en moi, je vous exprime ma plus profonde gratitude. À mes amis, mes enseignants et à tous les visages croisés sur ce chemin : merci pour votre présence, vos mots, vos gestes et votre foi en ma réussite.

HADJ MAHAMMED Smail



Remerciements

Avant tout, nous remercions Allah, le Tout-Puissant, de nous avoir accordé la santé, la longévité, la force et le courage nécessaires pour mener à bien ce travail.

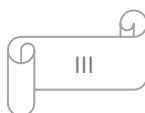
Nos remerciements les plus sincères vont particulièrement à notre encadrant, **Dr. CHOUITI Sidi Mohammed**, pour ses précieux conseils, sa disponibilité, son expertise, sa rigueur professionnelle et son accompagnement constant. Il nous a accordé sa confiance tout en nous guidant efficacement dans l'élaboration de ce projet.

Nous exprimons également notre gratitude aux membres du jury, pour avoir accepté d'évaluer ce mémoire et d'y consacrer de leur temps.

Nous tenons à remercier chaleureusement tous nos enseignants, qui, tout au long de notre parcours, ont fourni des efforts remarquables afin de nous offrir une formation de qualité.

Merci également à notre chère université, pour l'environnement de travail agréable et les conditions propices à l'apprentissage.

Enfin, un grand merci à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.



Résumé

Ce projet vise à répondre à l'insuffisance des systèmes traditionnels de gestion scolaire, qui peinent notamment à offrir des fonctionnalités avancées permettant d'accompagner efficacement les élèves.

L'analyse de cette problématique révèle que, malgré l'existence de nombreuses solutions de gestion scolaire, des aspects essentiels tels que la prédiction des performances des élèves et la recommandation personnalisée de ressources pédagogiques restent encore peu exploités.

Pour y remédier, nous avons conçu une plateforme intelligente permettant la prédiction des performances des élèves, la recommandation personnalisée de ressources éducatives ainsi que la simplification des tâches administratives.

Mots clés : systèmes de gestion scolaire, analyse prédictive, machine Learning, algorithmes de recommandation.

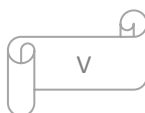
Abstract

This project aims to address the shortcomings of traditional school management systems, which often struggle to provide advanced features that effectively support students.

An analysis of this issue reveals that, despite the existence of many school management solutions, key aspects such as student performance prediction and personalized recommendation of educational resources remain largely underutilized.

To overcome this, we have designed an intelligent platform that enables prediction of student performance, personalized recommendation of educational resources, and simplification of administrative tasks.

Keywords: school management systems, predictive analysis, machine learning, recommendation algorithms.



الملخص

هدف هذا المشروع إلى معالجة أوجه القصور في أنظمة إدارة المدارس التقليدية، التي غالبًا ما تعجز عن توفير وظائف متقدمة تُمكن من دعم التلاميذ بفعالية.

تكشف دراسة هذه الإشكالية أنه، على الرغم من وجود العديد من حلول إدارة المدارس، فإن الجوانب الأساسية مثل التنبؤ بأداء التلاميذ والتوصية المخصصة بالموارد التعليمية لا تزال غير مستغلة بشكل كافٍ.

ولمعالجة ذلك، قمنا بتصميم منصة ذكية تُمكن من التنبؤ بأداء التلاميذ، وتقديم توصيات مخصصة بالموارد التعليمية، بالإضافة إلى تبسيط المهام الإدارية.

الكلمات المفتاحية: أنظمة إدارة المدارس، التحليل التنبؤي، التعلم الآلي، خوارزميات التوصية.

Tables des matières

Chapitre I : Fondements théoriques	4
1.1 Introduction	5
1.2 L'apprentissage automatique (machine Learning).....	5
1.2.1 Définition de l'apprentissage automatique	5
1.2.2 Les types d'apprentissage automatique	5
1.2.2.1 L'apprentissage supervisé.....	5
1.2.2.2 L'apprentissage non supervisé	7
1.2.2.3 L'apprentissage par renforcement	11
1.2.2.4 L'apprentissage profond.....	11
1.3 Les systèmes de recommandation	11
1.3.1 Typologie des systèmes de recommandation	11
1.3.1.1 Filtrage basé sur le contenu	11
1.3.1.2 Filtrage collaboratif	12
1.3.1.3 Approches hybrides.....	12
1.4 Conclusion	12
Chapitre II : Etat de l'art	13
2.1 Introduction :	14
2.2 Présentation de quelques plateformes éducatives intelligentes	14
2.2.1 Mindspark.....	14
2.2.2 Knewton Alta	15
2.2.3 Edmodo	18
2.2.4 DreamBox Learning	21
2.2.5 EduIA (notre solution)	22
2.3 Tableaux comparatifs	23
2.4 Synthèse des solutions éducatives analysées	24
2.5 Conclusion	25
Chapitre III : Conception.....	26
3.1 Introduction	27
3.2 Processus de modélisation avec UML	27
3.2.1 UML (Unified Modeling Language).....	27
3.2.2 Différents types de diagrammes d'UML.....	27
3.3 Délimitation du périmètre du système à modéliser	28
3.3.1 Les acteurs du système.....	28
3.3.2 Spécification des besoins.....	28
3.3.2.1 Besoins fonctionnels.....	29
3.3.2.2 Besoins non fonctionnels	31

3.4	Modélisation UML.....	33
3.4.1	Le diagramme de cas d'utilisation	33
3.4.2	Le diagramme de séquence.....	35
3.4.2.1	Description textuelle des cas d'utilisation.....	35
3.4.3	Conception de la base de données.....	49
3.4.3.1	Diagramme de classes :	49
3.4.3.2	Passage au modèle relationnel.....	52
3.5	Conception du système d'analyse prédictive.....	57
3.5.1	Choix du type d'apprentissage	57
3.5.2	Fonctionnement de notre choix	58
3.6	Conclusion	59
Chapitre IV : Implémentation		60
4.1	Introduction	61
4.2	Environnement de travail.....	61
4.2.1	Environnement matériel	61
4.2.2	Environnement logiciel.....	61
4.3	Choix des technologies utilisées.....	62
4.4	Mise en œuvre de l'architecture MVC	64
4.4.1	Modèle	64
4.4.2	Vue.....	64
4.4.3	Contrôleur	65
4.5	Rubix ML.....	66
4.6	Conception des interfaces utilisateur (IHM)	66
Conclusion générale et perspectives		72
Bibliographie		74

Liste des figures

Figure 1:Écran de Mindspark affichant un exemple de question de la classe de 6e sur le thème des POURCENTAGES [20].....	14
Figure 2 : TABLEAU DE BORD DES ENSEIGNANTS [21]	16
Figure 3: interface d'entrainement des élèves dans un MATIERE [21]	16
Figure 4:la performance de l'élève dans une MATIERE [21]	17
Figure 5: PAGE D'ACCUEIL D'EDMODO [22].....	19
Figure 6: PAGE D'ACCUEIL DE L'APPRENANT [22]	19
Figure 7: PAGE D'ACCUEIL DE L'ENSEIGNANT [22].....	20
Figure 8: PAGE D'ACCUEIL DE DREAMBOX [23]	21
Figure 9: CAPTURE D'ECRAN D'UN TABLEAU DE BORD D'ANALYSE DE CROISSANCE [23].....	22
Figure 10:Types de diagrammes d'UML.....	28
Figure 11:Diagramme de Cas d'utilisation du système	34
Figure 12:Diagramme de séquence du cas d'utilisation <<authentifier>>	38
Figure 13: Diagramme de séquence du cas d'utilisation <<inscrire l'élève>>	40
Figure 14:Diagramme de séquence du cas d'utilisation <<modifier un élève>>	42
Figure 15: Diagramme de séquence du cas d'utilisation <<supprimer un élève>>	44
Figure 16: Diagramme de séquence du cas d'utilisation << Prédire la performance des élèves >>	46
Figure 17: Diagramme de séquence du cas d'utilisation << générer les recommandations >>.....	48
Figure 18:Diagramme de classes de la solution	50
Figure 19: Exemple d'association « Un à plusieurs »	53
Figure 20: Exemple d'association « Plusieurs à plusieurs »	53
Figure 21:Exemple d'association « Un à un »	54
Figure 22:Décomposition par distinction d'une association d'héritage	55
Figure 23: Décomposition descendante d'une association d'héritage	55
Figure 24:Décomposition ascendante d'une association d'héritage	56
Figure 25: Fonctionnement du model.....	58
Figure 26:Architecture MVC.....	66
Figure 27: interface d'authentification	67
Figure 28:interface d'inscription d'un élève	67
Figure 29: liste des élèves	68
Figure 30: formulaire de modification	68
Figure 31:formulaire de notation.....	69
Figure 32:formulaire des recommandations.....	69
Figure 33:formulaire de notation des absences	70
Figure 34: formulaire de paramétrage.....	70
Figure 35:Tableau de bord d'administration	71

Liste des tableaux

Tableau 1:tableau comparatif entre notre solution et les autres existantes..... 24

Tableau 2:description textuelle de s’authentifier 37

Tableau 3: description textuelle de l’inscription d’un élève 39

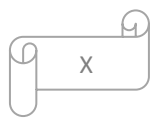
Tableau 4: description textuelle de modifier un élève 41

Tableau 5: description textuelle de supprimer un élève..... 43

Tableau 6: description textuelle de Prédire la performance des élèves 45

Tableau 7: description textuelle de générer les recommandations 47

Tableau 8: Description de la base de données..... 51



Introduction générale

Contexte

Dans un monde en perpétuelle évolution, la technologie s'impose comme un levier incontournable de développement, touchant de nombreux secteurs, dont celui de l'éducation. Pilier fondamental du progrès social, économique et politique, l'éducation se doit d'évoluer au rythme des avancées technologiques. Un retard dans ce domaine peut freiner significativement le développement global d'un pays.

Aujourd'hui, l'intégration des technologies dans le système éducatif ne constitue plus un choix, mais une nécessité. Elle permet de moderniser les méthodes de gestion et d'enseignement, de faciliter l'accès au savoir, d'améliorer l'engagement des apprenants et de suivre plus efficacement les résultats scolaires. À l'ère du numérique, l'éducation doit être repensée pour répondre aux attentes des apprenants et aux exigences d'une société en constante mutation.

Problématique

Malgré l'importance de ces avancées, de nombreuses institutions scolaires continuent à gérer les tâches administratives de manière traditionnelle : suivi des absences, gestion des notes, inscriptions, etc. Ces processus, souvent manuels, sont chronophages, peu fiables et sujets aux erreurs en raison de leur dépendance excessive à l'intervention humaine.

Par ailleurs, les systèmes d'évaluation actuels ne permettent pas d'identifier rapidement les difficultés des élèves, ni d'agir de façon préventive contre l'échec scolaire. L'absence d'outils technologiques capables de traiter et d'analyser efficacement les données scolaires limite l'évolution vers une gestion plus intelligente et proactive. Les établissements manquent de solutions intégrant des technologies avancées, telles que l'intelligence artificielle (IA) et l'analyse prédictive, pour accompagner les élèves de manière personnalisée et aider les enseignants dans leur prise de décision.

Objectifs du projet

Face à ces constats, notre projet (EduIA) vise à développer une **plateforme intelligente de gestion scolaire intégrant des analyses prédictives**. Les objectifs principaux sont :

- **Automatiser** les processus administratifs liés à la gestion scolaire (inscriptions, absences, notes, etc.) pour gagner en efficacité.

- **Analyser** les données académiques (notes, absences, participation) pour mieux comprendre la progression des élèves.
- **Prédire** les performances futures des apprenants à l'aide d'algorithmes de machine Learning.
- **Générer des recommandations pédagogiques personnalisées** pour accompagner chaque élève selon son profil.
- **Fournir un tableau de bord interactif** permettant une visualisation en temps réel des données et facilitant la prise de décision.

Ce mémoire est structuré en quatre chapitres principaux, chacun abordant un aspect essentiel du projet :

- **Chapitre 1 : Fondements théoriques**

Présentation des concepts clés liés à l'intelligence artificielle, à l'apprentissage automatique (ML) et aux algorithmes de recommandation.

- **Chapitre II : État de l'art**

Analyse critique des systèmes existants de gestion scolaire intelligente, mettant en évidence leurs forces et leurs limites.

- **Chapitre III : Conception de la solution**

Description détaillée de l'architecture structurelle et fonctionnelle du système, incluant :

- Diagrammes de cas d'utilisation, de classes et de séquence.
- Modèle relationnel de la base de données.

- **Chapitre IV : Implémentation et réalisation**

Présentation de :

- Les outils et technologies utilisés ;
- L'architecture technique de l'application ;
- Les interfaces utilisateur de la plateforme.

- **Conclusion générale et perspectives**

Chapitre I : Fondements théoriques

1.1 Introduction

Ce chapitre présente les concepts clés liés à l'intelligence artificielle (IA), à l'apprentissage automatique (ML), ainsi qu'aux techniques de recommandation. Ces éléments théoriques sont indispensables pour justifier les choix techniques effectués dans la suite du mémoire (ou dans le projet).

Nous commençons par définir l'apprentissage automatique (ML), ses types. Ensuite nous aborderons les systèmes de recommandations avant de conclure par le type d'apprentissage que nous avons retenu pour notre projet.

1.2 L'apprentissage automatique (machine Learning)

1.2.1 Définition de l'apprentissage automatique

L'apprentissage automatique (ML) est un sous domaine de l'intelligence artificielle (IA), qui utilise les données pour entraîner une machine à effectuer des tâches.

Contrairement à la programmation traditionnelle, dans laquelle toutes les règles sont programmées explicitement, l'apprentissage automatique (ML) utilise des algorithmes pour créer automatiquement des ensembles de règles. Il emprunte également un ensemble de technique de nombreuses disciplines tels que : la théorie des probabilités et les neurosciences, combinées à de nouvelles idées dans le but d'obtenir des informations grâce aux données et au calcul [1].

1.2.2 Les types d'apprentissage automatique

L'apprentissage automatique (ou machine Learning) se divise en plusieurs types en fonction de la manière dont l'algorithme apprend à partir des données. Voici les principaux types d'apprentissage automatique et quelques exemples d'algorithmes associés.

1.2.2.1 L'apprentissage supervisé

L'apprentissage supervisé est un type d'apprentissage automatique qui intègre un signal d'entraînement sous la forme d'étiquettes, qui sont souvent déterminées par un expert humain. Les étiquettes sont le résultat souhaité par un apprenant compte tenu de l'échantillon que nous lui montrons. Pour cette raison, vous pouvez considérer l'apprentissage supervisé comme un apprentissage par l'exemple [1].

Quelques algorithmes de l'apprentissage supervisés sont :

- **Régression linéaire** : la régression linéaire est utilisée pour identifier la relation entre une variable dépendante et une ou plusieurs variables indépendantes et est généralement utilisée pour faire des

prédictions sur les résultats futurs. Lorsqu'il n'y a qu'une seule variable indépendante et une seule variable dépendante, on parle de régression linéaire simple. Lorsque le nombre de variables indépendantes augmente, on parle de régression linéaire multiple. Pour chaque type de régression linéaire, on cherche à tracer une ligne de meilleur ajustement, qui est calculée par la méthode des moindres carrés. Cependant, contrairement aux autres modèles de régression, cette ligne est droite lorsqu'elle est tracée sur un graphique [1].

- **Régression logistique** : alors que la régression linéaire est utilisée lorsque les variables dépendantes sont continues, la régression logistique est choisie lorsque la variable dépendante est catégorique, c'est-à-dire qu'elle a des résultats binaires, tels que « true » et « false » ou « yes » et « no ». Cette méthode d'apprentissage automatique est très utilisée dans le domaine du marketing pour évaluer la vente ou non de produits à la suite d'une décision ou dans le domaine médical pour évaluer la guérison ou non d'un patient [1].

- **Machines à vecteurs de support (SVM)** : une machine à vecteurs de support est un modèle d'apprentissage supervisé populaire développé par Vladimir Vapnik, utilisé à la fois pour la classification et la régression des données. Cela dit, il est généralement utilisé pour les problèmes de classification, en construisant un hyperplan où la distance entre deux classes de points de données est maximale. Cet hyperplan est connu sous le nom de limite de décision, séparant les classes de points de données (par exemple, oranges et pommes) de chaque côté du plan [1].

- **K plus proche voisin** : le k plus proche voisin, également connu sous le nom d'algorithme kNN (**k-Nearest Neighbors**), appelé en français « voisin le plus proche », est un algorithme non paramétrique qui classe les points de données en fonction de leur proximité et de leur association avec d'autres données disponibles. Cet algorithme suppose que des points de données similaires peuvent être trouvés à proximité les uns des autres. En conséquence, il cherche à calculer la distance entre les points de données, généralement au moyen de la distance euclidienne, puis attribue une catégorie en fonction de la moyenne ou de la catégorie la plus fréquente. Sa facilité d'utilisation et son faible temps de calcul en font un algorithme privilégié par les « data scientists », mais à mesure que le jeu de données de test s'agrandit, le temps de traitement s'allonge, ce qui le rend moins attrayant pour les tâches de classification. Le kNN est généralement utilisé pour les moteurs de recommandation et la reconnaissance d'images [1].

- **Forêt d'arbres décisionnels** : la forêt d'arbres décisionnels est un autre algorithme d'apprentissage automatique supervisé flexible utilisé à la fois à des fins de classification et de régression. La « forêt » fait référence à un ensemble d'arbres de décision non corrélés, qui sont ensuite fusionnés pour réduire la variance et créer des prédictions de données plus précises [1].
- **Arbres de Décision** : Les arbres de classification (ou arbre de décision) et de régression sont des méthodes qui permettent de créer un modèle de décision. Cette structure est constituée de nœuds racines, constituant les entrées de l'arbre, de nœuds internes, réalisant des opérations intermédiaires et de feuilles (ou nœuds terminaux), représentant la valeur des variables de sorties. Cette structure hiérarchique se base sur un enchaînement de décisions créant le passage d'un nœud à l'autre, jusqu'aux feuilles. L'arbre est généralement construit récursivement en cherchant pour chaque nœud la décision permettant de réaliser le meilleur partage de l'ensemble de données. Après une décision, on dispose alors de plusieurs sous-ensembles de données sur lesquels on applique le même processus. Les arbres de décision ont l'avantage d'avoir un processus de décision clair, garantissant une bonne explicabilité de la solution. Ils ont longtemps été utilisés avant le développement des réseaux de neurones artificiels. Il est possible de décomposer les problèmes plus complexes en utilisant plusieurs arbres de classification, on parle alors de forêt aléatoire [1].

1.2.2.2 L'apprentissage non supervisé

Est une forme d'apprentissage qui ne nécessite pas de données étiquetées [1].

Selon [2] L'apprentissage non supervisé regroupe des techniques qui visent à découvrir des structures cachées dans des données non étiquetées. Les approches les plus répandues sont la classification, les règles d'association et la réduction de la dimensionnalité. Chacune de ces méthodes repose sur des principes et des algorithmes spécifiques que nous détaillons ci-après.

1. Classification non supervisée

La classification non supervisée, également appelée clustering, consiste à regrouper des données selon leur similarité, sans connaissance préalable de catégories ou d'étiquettes. Elle est utilisée pour révéler des structures sous-

jaçentes dans les données et peut prendre plusieurs formes : exclusive, chevauchante, hiérarchique et probabiliste [2].

✓ **Classification exclusive**

La classification exclusive est une forme de groupement qui stipule qu'un point de données ne peut exister que dans un seul groupe. On parle également de classification « dure ». L'algorithme de classification K-means ou (K-moyennes) est un exemple de classification exclusive [2].

◆ **La classification K-means :**

Est un exemple courant de méthode de classification exclusive où les points de données sont répartis en K groupes, où K représente le nombre de groupes selon la distance par rapport au centroïde de chaque groupe. Les points de données les plus proches d'un centroïde donné seront regroupés dans la même catégorie. Une valeur K plus élevée indiquera des groupements plus petits avec plus de granularité alors qu'une valeur K plus petite aura des groupements plus grands et moins de granularité. La classification K-means ou K-moyennes est communément utilisée dans la segmentation du marché, la classification de documents, la segmentation d'images et la compression d'images [2].

✓ **Les classifications chevauchantes**

Diffèrent de la classification exclusive en ce qu'elles permettent aux points de données d'appartenir à plusieurs groupes avec des degrés d'appartenance différents. La classification K-means « douce » ou « floue » est un exemple de classification chevauchante [2].

✓ **Classification hiérarchique**

La classification hiérarchique, également appelée classification ascendante hiérarchique (CAH), est une méthode non supervisée pouvant suivre une approche agglomérative ou divisive. Dans l'approche agglomérative, dite « ascendante », chaque donnée débute comme un groupe indépendant, puis les groupes sont fusionnés progressivement en fonction de leur similarité, jusqu'à former un ensemble unique. Quatre méthodes principales sont généralement utilisées pour évaluer cette similarité [2] :

1. **Liaison de Ward** : Cette méthode indique que la distance entre deux clusters est définie par l'augmentation de la somme des carrés après la fusion des groupes [2].
2. **Liaison moyenne** : Cette méthode est définie par la distance moyenne entre deux points dans chaque groupe [2].
3. **Liaison complète (ou maximale)** : Cette méthode est définie par la distance maximale entre deux points dans chaque groupe [2].
4. **Liaison simple (ou minimale)** : Cette méthode est définie par la distance minimale entre deux points dans chaque groupe [2].

La distance euclidienne est l'unité de mesure la plus couramment utilisée pour calculer ces distances, cependant, d'autres unités de mesure, telles que la distance de Manhattan, sont également citées dans les publications spécialisées [2].

✓ **Classification probabiliste**

Un modèle probabiliste est une approche non supervisée utilisée pour traiter des problèmes d'estimation de densité ou de classification « floue ». Dans ce type de classification, les données sont regroupées selon que la probabilité qu'un point appartient à une distribution spécifique. Le modèle de mélange gaussien (GMM) est l'une des méthodes de classement probabiliste les plus couramment utilisées [2].

◆ **Les modèles de mélange gaussien :**

C'est un modèle statistique qui permet d'estimer la distribution de variables aléatoires en la modélisant comme une somme de distributions gaussiennes de paramètres différents. Durant la phase d'apprentissage, l'objectif est d'estimer les moyennes et variances de ces distributions en maximisant la vraisemblance. Ce type de modèle est couramment utilisé dans des problèmes de classification comme la reconnaissance de l'écriture manuscrite et la segmentation d'images par leur histogramme. Le modèle de mélange de gaussien permet, contrairement à l'algorithme des K moyennes, de réaliser des groupes de tailles différentes [1].

2. Règles d'association

Les règles d'association visent à découvrir des relations fréquentes entre les éléments d'un ensemble de données. Elles sont particulièrement utiles dans l'analyse des paniers d'achat ou pour les systèmes de recommandation.

L'algorithme Apriori, très populaire, identifie des ensembles d'éléments fréquents, puis en déduit des règles de type : « Si un client achète A, alors il est probable qu'il achète B ». Il utilise des structures telles que des arbres de hachage pour optimiser l'exploration des combinaisons d'items. Des applications concrètes incluent les recommandations d'articles sur Amazon ou de playlists sur Spotify. D'autres algorithmes notables incluent FP-Growth et Eclat [2].

3. Réduction de la dimensionnalité

Lorsque les données comportent un grand nombre de variables, leur traitement peut devenir complexe (surapprentissage, lenteur d'exécution, etc.). La réduction de la dimensionnalité permet de simplifier les jeux de données tout en conservant leur structure essentielle.

Voici l'une des méthodes utilisées :

- **Analyse en Composantes Principales (ACP) :**

« Cet algorithme vise à réduire les dimensions d'un jeu de données, ce qui accélère et facilite son traitement. Cette méthode crée des variables décorrélées en réalisant un changement de base. Les vecteurs de cette nouvelle base, appelés composantes principales, sont ensuite sélectionnés pour ne garder que celles portant le plus d'informations. Il est alors possible de minimiser la perte d'information tout en réduisant efficacement la dimensionnalité de l'ensemble. Cet algorithme est largement utilisé en neuroscience ou en analyse des mouvements où de nombreux signaux de grande dimensions (activation de neurones ou mouvements des membres) sont utilisés pour traduire des informations plus simples (réponse à un stimuli ou déplacement dans une direction). D'autres algorithmes de réduction de dimensionnalité existent, comme la décomposition en valeurs singulières ou l'analyse en composantes indépendantes (ICP) » [1].

1.2.2.3 L'apprentissage par renforcement

L'apprentissage par renforcement est une méthode d'apprentissage automatique qui permet de faire interagir un modèle avec son environnement en l'entraînant. Au sein de son environnement (généralement simulé), l'agent dispose d'un modèle initial optimisé. En faisant certaines actions, basée sur des observations provenant généralement de capteurs, il doit alors atteindre son objectif [1].

1.2.2.4 L'apprentissage profond

L'apprentissage profond est une sous-discipline de l'apprentissage automatique qui intègre des couches de calcul qui forment des représentations de caractéristiques de plus en plus complexes. Il s'agit d'un changement de manière qui passe de l'ingénierie des fonctionnalités à la possibilité pour l'apprenant de construire ses propres caractéristiques à partir des données brutes. Il est utilisé dans la reconnaissance d'images, le traitement du langage naturel (NLP) et pour d'autres tâches impliquant des entrées brutes de très haute dimension [2].

1.3 Les systèmes de recommandation

Un système de recommandation a pour principal objectif de proposer à un internaute des éléments pertinents en fonction de ses préférences, des informations ou des produits qu'il recherche. L'utilisateur bénéficie ainsi d'un gain de temps mais profite aussi de conseils ou de propositions émanant du système auquel il n'aurait pas spontanément pensé [3].

1.3.1 Typologie des systèmes de recommandation

Les systèmes de recommandation peuvent être classés en trois grandes catégories :

1.3.1.1 Filtrage basé sur le contenu

Cette méthode se base sur les attributs des items à recommander et sur le profil de l'utilisateur. Il suggère donc des items proches de ceux que l'utilisateur a aimé auparavant [3]. Par exemple, dans un contexte scolaire, si un élève a bien réussi des exercices sur un thème, le système peut lui recommander des ressources similaires.

1.3.1.2 Filtrage collaboratif

Cette approche est basée sur les goûts et les comportements d'une population d'utilisateurs. Elle part du principe que si deux utilisateurs ont aimé les mêmes choses dans le passé, ils aimeront également les mêmes choses à l'avenir. Les systèmes de recommandation suggèrent donc des objets qu'ont apprécié des utilisateurs aux profils similaires à celui de l'utilisateur ciblé par le système [3].

1.3.1.3 Approches hybrides

Ces systèmes combinent les deux méthodes précédentes pour pallier leurs limites respectives. En intégrant à la fois les préférences individuelles et les tendances communautaires, ils offrent des recommandations plus robustes et pertinentes [3].

1.4 Conclusion

Ce chapitre nous a permis d'établir les fondements théoriques de l'apprentissage automatique ainsi que des systèmes de recommandation. Le chapitre suivant s'attachera à examiner l'état de l'art des solutions éducatives existantes, en mettant en lumière leurs limites actuelles et les opportunités qu'offre l'intégration d'une dimension prédictive.

Chapitre II : Etat de l'art

2.1 Introduction :

Dans cette section, nous allons nous intéresser à quelques plateformes et systèmes existants qui partagent des fonctionnalités communes avec notre plateforme (EduIA). Nous souhaitons ainsi identifier les points forts et les limites de ces solutions dans le but de bâtir notre propre plateforme. Grâce à cette analyse comparative, nous pourrions identifier les manques dans les solutions actuelles et mettre en avant les éléments innovants et différenciant de notre projet (EduIA) comme l'utilisation de l'intelligence artificielle pour prédire les notes des élèves et suggérer des ressources pédagogiques adaptés.

Nous débutons par la présentation de quelque plateforme avec leurs fonctionnalités respectives. Après nous établissons un tableau comparatif entre notre plateforme (EduIA) et celles-ci avant de conclure avec une synthèse mettant évidence les similitudes et les différences.

2.2 Présentation de quelques plateformes éducatives intelligentes

2.2.1 Mindspark

Est une plateforme d'apprentissage adaptatif qui permet aux élèves de progresser efficacement à leur rythme [4].

La figure 1 ci-dessous présente le nom de l'élève, la classe, le sujet sélectionné, une question liée à ce sujet, un indicateur de progression dans le thème, le nombre de "Sparkies" collectés pendant la session, des options pour changer de sujet ou terminer la session en cours, ainsi qu'une option pour fournir un retour (feedback).

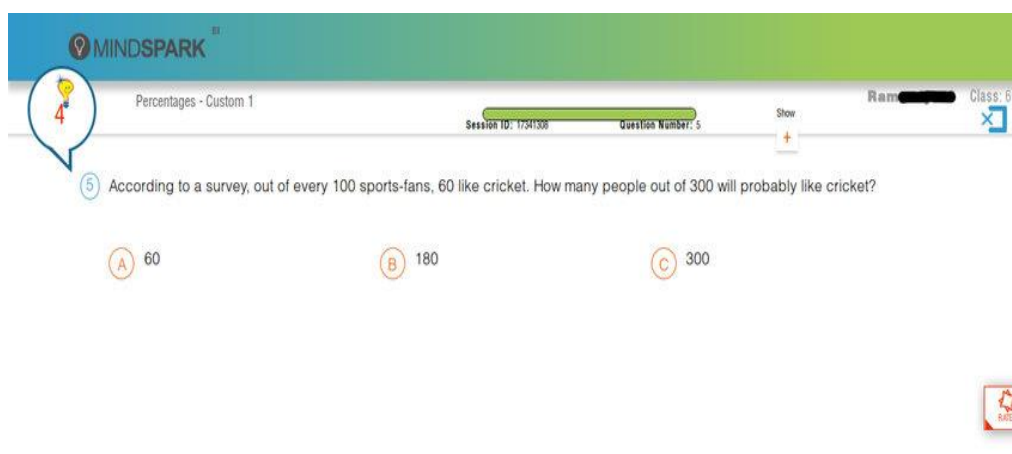


FIGURE 1: ÉCRAN DE MINDSPARK AFFICHANT UN EXEMPLE DE QUESTION DE LA CLASSE DE 6E SUR LE THEME DES POURCENTAGES [20].

Les Fonctionnalités principales de Mindspark

Détecter :

- Les élèves effectuent un test diagnostique afin d'évaluer leurs connaissances et leurs idées fausses sur un sujet
- Les élèves répondent à des questions conçues pour déterminer le type d'incompréhension détecté sur un sujet (des lacunes dans les connaissances) [4].

Diagnostiquer :

- Les schémas d'erreurs dans les réponses des élèves sur un sujet
- Un enseignement correctif ciblé est assigné en fonction du blocage conceptuel détecté [4].

Agir :

- **Automatisation conditionnelle :**
 - Fournit divers supports pédagogiques (questions, jeux interactifs, activités) conçus spécialement pour répondre aux incompréhensions et aux lacunes qui ont été diagnostiquées (niveau de la tâche)
 - Fournit des commentaires spécifiques et ponctuels lorsqu'une mauvaise réponse est donnée (niveau de l'étape)
 - Détermine si l'élève peut passer au niveau de difficulté suivant (niveau de la tâche ou du programme de cours) [4].

2.2.2 Knewton Alta

Après avoir présenté Mindspark, nous nous intéressons maintenant à Knewton Alta, une autre plateforme d'apprentissage adaptatif développé pour accompagner les élèves à combler leurs problèmes en matière de connaissances et à fournir plus d'effort dans les apprentissages [5].

La figure 2 ci-dessous montre le tableau de bord de l'enseignant. Sur ce tableau de bord, les enseignants peuvent consulter une banque de questions répertoriant les problèmes rencontrés par les élèves et analyser en détail leurs réponses à chaque problème, ainsi que les commentaires reçus. L'outil signale également les problèmes fréquemment oubliés afin que les enseignants puissent les examiner en classe.

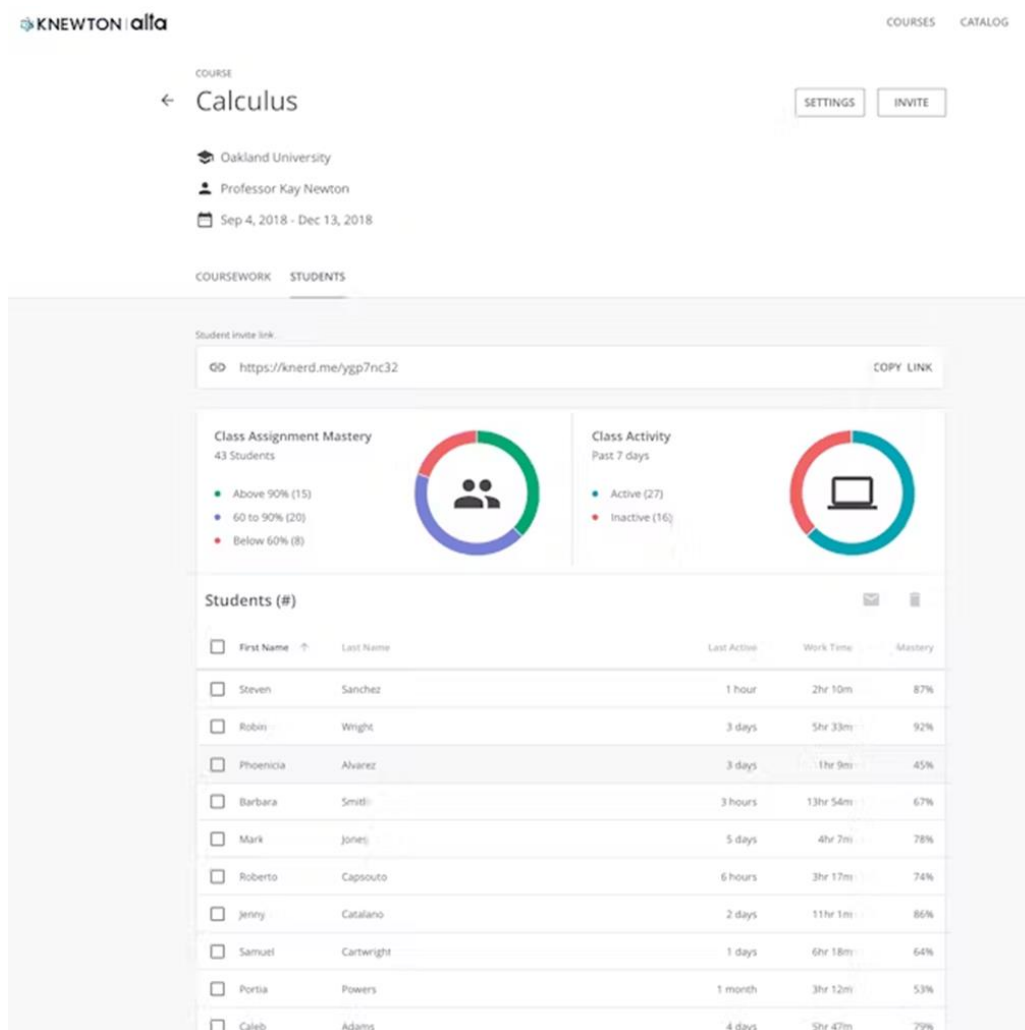


FIGURE 2 : TABLEAU DE BORD DES ENSEIGNANTS [21]

La figure 3 ci-dessous montre l'interface d'entraînement des élèves dans une matière donnée. Sur cette même interface, l'élève peut voir sa progression.

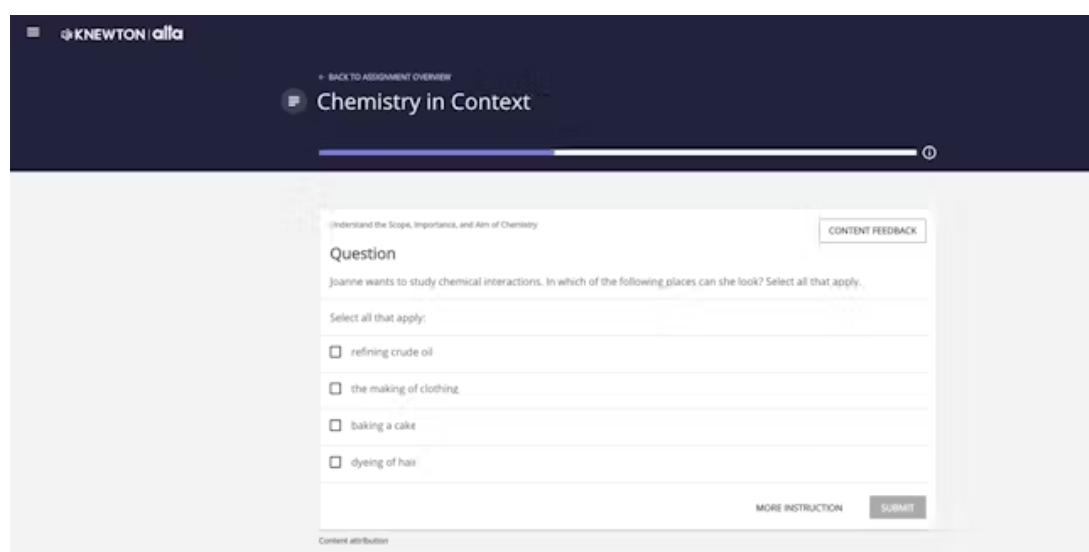


FIGURE 3: INTERFACE D'ENTRAÎNEMENT DES ELEVES DANS UN MATIERE [21]

Quant à cette figure 4, elle permet aux élèves de visualiser leur performance.

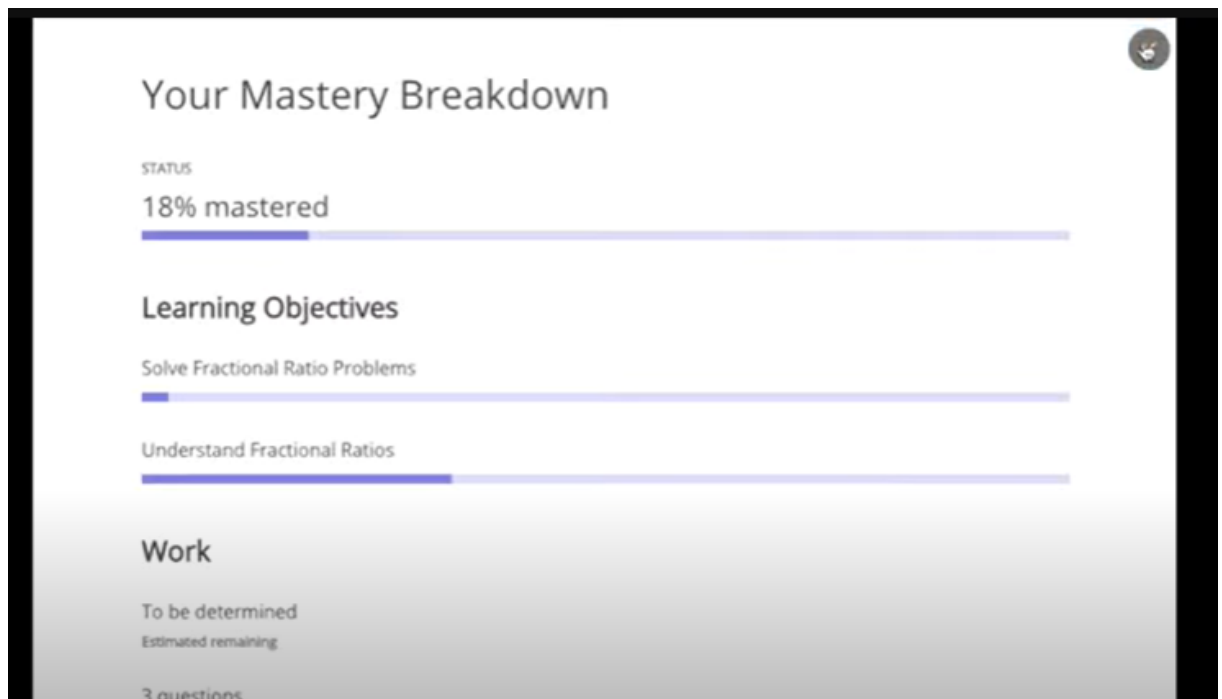


FIGURE 4: LA PERFORMANCE DE L'ELEVE DANS UNE MATIERE [21]

Les Fonctionnalités principales de Knewton Alta :

- **Apprentissage personnalisé basé sur la maîtrise**

Knewton Alta adapte le parcours d'apprentissage de chaque élève en fonction de ses performances, offrant ainsi une expérience individualisée qui vise la maîtrise des compétences [5].

- **Explications détaillées des réponses**

Pour chaque question, Alta fournit des explications immédiates et approfondies, qu'elles soient correctes ou incorrectes, afin de renforcer la compréhension des concepts et corriger les erreurs de raisonnement [5].

- **Remédiation dynamique**

Alta ajuste en temps réel les contenus en fonction du niveau de compétence détecté. En cas de difficulté, il identifie instantanément les lacunes et propose une remédiation ciblée, y compris sur les prérequis nécessaires [5].

- **Analyse prédictive de la préparation aux tests**

Des tableaux de bord prédictifs permettent aux enseignants d'évaluer la préparation des élèves avant les examens, à partir de leurs performances dans les activités d'apprentissage [5].

- **Analyses détaillées des élèves et des cours**

Knewton Alta offre aux enseignants des outils d'analyse pour observer les schémas d'apprentissage, repérer les élèves à risque et prendre des décisions pédagogiques éclairées [5].

- **Création facile d'expériences d'apprentissage ciblées**

Les enseignants peuvent concevoir et personnaliser rapidement des devoirs adaptatifs, avec une grande flexibilité pour s'adapter aux objectifs spécifiques de chaque cours [5].

Ces fonctionnalités font de Knewton Alta une solution d'apprentissage adaptatif complète et puissante, orientée vers l'amélioration continue des résultats des élèves grâce à une pédagogie centrée sur la maîtrise et l'adaptabilité.

2.2.3 Edmodo

Après avoir présenté Knewton Atla, nous nous intéressons maintenant à Edmodo, une autre plateforme technologique éducative qui connecte enseignants, élèves et parents. Elle offre un environnement d'apprentissage virtuel où les enseignants peuvent gérer les classes, partager des ressources, donner des devoirs et animer des discussions. Considérez-le comme un réseau social conçu pour l'éducation, similaire à Facebook mais avec des fonctionnalités adaptées à l'apprentissage. Grâce à l'intégration de l'intelligence artificielle (IA), Edmodo améliore les expériences d'apprentissage en automatisant les tâches, en fournissant des recommandations personnalisées et en améliorant l'engagement [6].

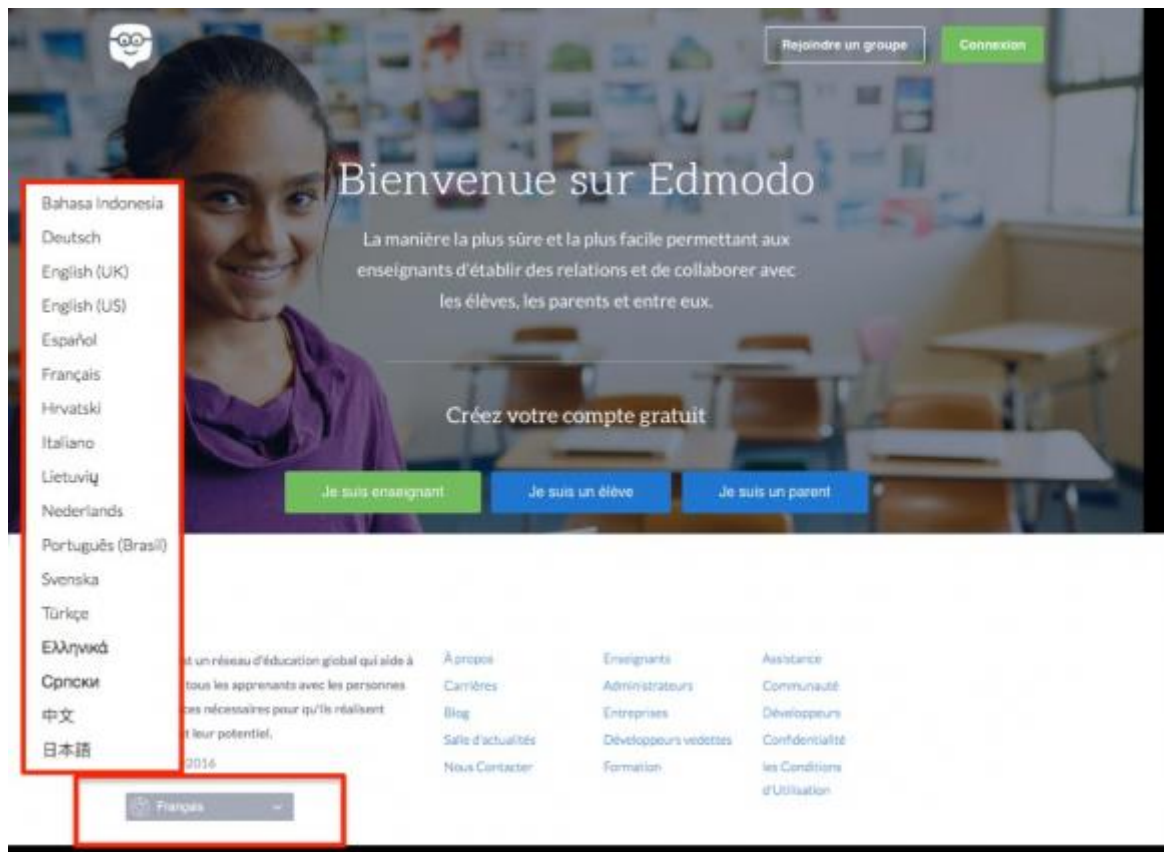


FIGURE 5: PAGE D'ACCUEIL D'EDMODO [22]

La figure 6 ci-dessous montre la page d'accueil de l'élève lorsqu'il se connecte.

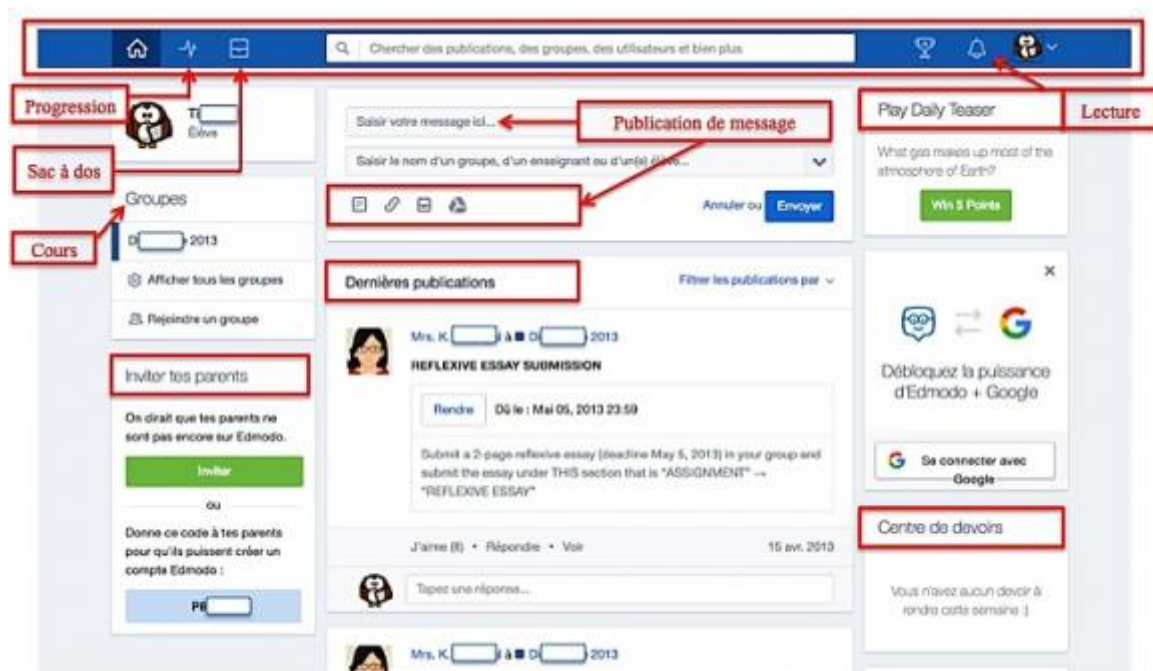


FIGURE 6: PAGE D'ACCUEIL DE L'APPRENANT [22]

La figure 7 ci-dessous montre la page d'accueil de l'enseignant lorsqu'il se connecte.

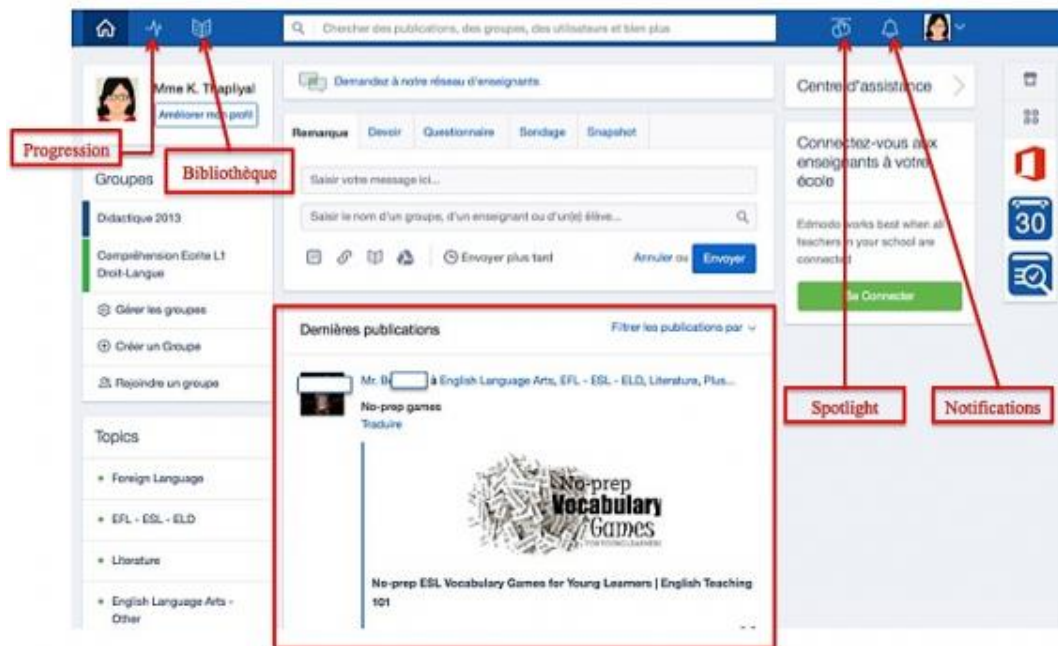


FIGURE 7: PAGE D'ACCUEIL DE L'ENSEIGNANT [22]

Les Fonctionnalités principales de Edmodo :

- **Gestion des classes**
 - Les enseignants peuvent créer des classes virtuelles pour différentes matières ou groupes d'élèves.
 - Ils peuvent partager des devoirs, des questionnaires et du matériel d'étude.
 - Les élèves peuvent soumettre leurs devoirs numériquement.
 - Les parents peuvent suivre les progrès de leur enfant [6]
- **Communication et collaboration**
 - Les forums de discussion permettent les interactions entre élèves et enseignants.
 - Un système de messagerie privée permet une communication directe entre les élèves et les enseignants.
 - Les sondages et les enquêtes aident les enseignants à recueillir des commentaires [6].

- **Partage des ressources et gestion du contenu**

- Téléchargez et organisez des fichiers (PDF, vidéos, présentations).
- Les enseignants peuvent partager des liens et des références pédagogiques.
- Bibliothèque numérique permettant aux enseignants de trouver et d'utiliser du matériel partagé [6].

- **Évaluation et analyse**

- Quiz et tests en ligne avec notation automatisée.
- L'analyse des performances aide les enseignants à suivre les progrès des élèves.
- Les informations basées sur l'IA recommandent des supports d'étude personnalisés.[6]

2.2.4 DreamBox Learning

Après avoir présenté Edmodo, nous nous intéressons maintenant à DreamBox Learning, une autre plateforme d'apprentissage adaptatif en ligne dédiée aux mathématiques, principalement conçue pour les élèves du primaire et du secondaire. Elle propose des leçons interactives qui s'ajustent en temps réel aux réponses et au rythme de chaque élève, offrant ainsi un parcours personnalisé visant à renforcer la compréhension conceptuelle et la maîtrise des compétences mathématiques.[7]

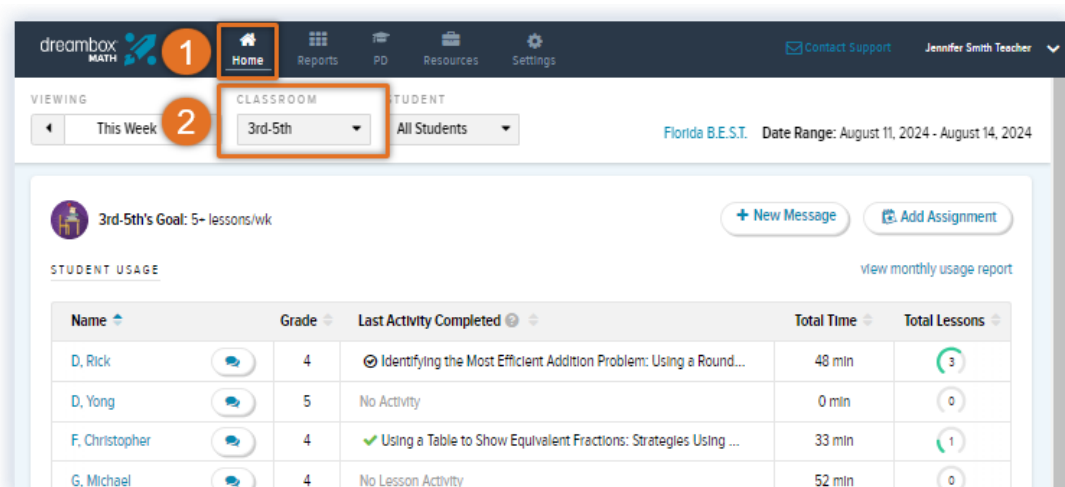


FIGURE 8: PAGE D'ACCUEIL DE DREAMBOX [23]

La figure 9 ci-dessous est une interface qui montre l'évolution des élèves.



FIGURE 9: CAPTURE D'ECRAN D'UN TABLEAU DE BORD D'ANALYSE DE CROISSANCE [23]

Les Fonctionnalités principales de DreamBox :

- **Apprentissage personnalisé et adaptatif**

DreamBox Learning propose des programmes mathématiques en ligne personnalisés et adaptatifs conçus pour aider les élèves de tous âges à améliorer leurs compétences mathématiques. En utilisant des algorithmes sophistiqués, la plateforme adapte l'expérience d'apprentissage à chaque élève, fournissant des leçons et des commentaires personnalisés pour assurer un engagement et une compréhension maximaux [7].

- **Suivi des progrès en temps réel**

La plateforme fournit des commentaires instantanés sur les performances des élèves, permettant aux enseignants et aux parents de suivre les progrès en temps réel et d'identifier rapidement les domaines nécessitant une attention particulière [7].

2.2.5 EduIA (notre solution)

Est une plateforme web intelligente de gestion scolaire, conçue pour moderniser la gestion administrative et pédagogique des établissements

éducatifs. Elle intègre une composante d'intelligence artificielle permettant de prédire les performances des élèves à partir de leurs données notes (classe, domicile), participation) et de générer des recommandations personnalisées. Elle offre également un tableau de bord interactif pour visualiser les statistiques scolaires en temps réel, et prend en charge la gestion des utilisateurs (élèves, enseignants, administrateurs), des cours et des absences. Son objectif principal est d'apporter une aide à la décision aux enseignants et aux responsables pédagogiques.

2.3 Tableaux comparatifs

Ci-dessous un tableau comparatif entre notre solution (EduIA) et les solutions existantes déjà mentionnées dans notre rapport :

Critères	EduIA	Knewton Alta	Edmodo	DreamBox Learning	Mindspark
Public ciblé	Primaire/sec ondaire	Universi taire	Primaire/sec ondaire	Primaire/sec ondaire	Primaire/sec ondaire
Gestion administrati ve	Complète (notes, absences, emploi du temps, etc.)	Non	Partiellemen t	Non	Non (focalisé uniquement sur l'apprentissa ge)
Recomman dations personnalis ées	Oui (en fonction du profil de l'élève : performance s, assiduité, etc.)	Oui (remédi ation ciblée)	Oui (contenus recommandé s via IA)	Oui (leçons et supports adaptatifs selon les progrès)	Oui (corrections ciblées + contenus personnalisé s selon les erreurs)
IA (Intelligence Artificielle)	Oui	Non	Partiellemen t	Partiellemen t	Partiellemen t
Multi- disciplines	Oui	Oui	Oui	Mathémati ques uniquement	Principaleme nt mathémati ques
Suivi des performanc es	Oui	Oui	Oui	Oui	Oui (indicateurs de progression

					et analyse des erreurs en temps réel)
--	--	--	--	--	---------------------------------------

TABLEAU 1: TABLEAU COMPARATIF ENTRE NOTRE SOLUTION ET LES AUTRES EXISTANTES

2.4 Synthèse des solutions éducatives analysées

Parmi les nombreuses solutions digitales que nous avons présentées dans ce rapport, la nôtre (EduIA) est unique en son genre parce qu'elle permet de gérer l'ensemble des aspects de la vie scolaire tout en étant intelligente.

Tandis que la plupart des outils interviennent soit au niveau de l'apprentissage personnalisé (Knewton Alta, DreamBox, Mindspark), ou de la communication dans la classe (Edmodo).

Notre solution (EduIA) va plus loin parce qu'elle :

- **Automatise** la gestion administrative de l'établissement (inscriptions, absences, notes, etc.) pour améliorer son efficacité.
- **Analyse** les informations pédagogiques (notes, absences, participation) pour mieux les comprendre.
- **Prédit** le comportement futur des élèves via du machine Learning.
- **Elabore des** recommandations pédagogiques personnalisées pour accompagner chaque élève en fonction de son profil.
- **Met à disposition** d'un tableau de bord interactif pour visualiser les données en temps réel et faciliter la prise de décision.

Les autres plateformes ne proposent pas, ou peu, de fonctionnalités de gestion administrative :

- **Knewton Alta** se destine à l'enseignement supérieur avec des fonctionnalités centrées sur l'apprentissage adaptatif, sans gestion d'établissement.
- **Edmodo** se présente comme un réseau social éducatif facilitant la communication entre les différents acteurs de l'éducation mais n'offre ni fonctionnalités prédictives ni IA adaptative avancée.
- **DreamBox Learning** propose un bon système d'apprentissage adapté, mais uniquement pour les mathématiques en élève de primaire, sans fonctionnalités de gestion des établissements scolaires.

- **Mindspark** se spécialise dans l'apprentissage adaptatif en mathématiques pour les élèves du primaire et du secondaire, en s'appuyant sur l'intelligence artificielle pour détecter, diagnostiquer et corriger les erreurs conceptuelles, mais il ne propose aucune gestion administrative ni prise en charge multidisciplinaire.

2.5 Conclusion

Les études comparatives nous ont permis d'identifier les solutions existantes dans le domaine de la gestion scolaire et de l'accompagnement pédagogique.

Grâce à cette analyse, nous avons pu positionner notre propre plateforme (EduIA) par rapport aux outils actuels. Elle nous offre une vision claire du paysage des innovations éducatives, en mettant en avant les spécificités de notre solution : automatisation des tâches administratives, analyses prédictives des performances, et recommandations pédagogiques personnalisées.

Ainsi, notre solution (EduIA) se distingue par sa capacité à combiner technologie, intelligence artificielle et besoins pédagogiques concrets, apportant une valeur ajoutée significative face aux solutions déjà disponibles.

Chapitre III : Conception

3.1 Introduction

Nous abordons à présent les phases de spécification et de conception. En informatique, le développement d'applications performantes et robustes nécessite une approche structurée, quelle que soit l'envergure ou la complexité du projet. Cette exigence est d'autant plus cruciale aujourd'hui, où les applications doivent répondre à des besoins de plus en plus exigeants.

Dans ce chapitre, nous adoptons une démarche intuitive de spécification et de conception basée sur le langage UML (Unified Modeling Language), un standard largement utilisé pour la modélisation des systèmes logiciels.

Ce Chapitre sera organisé en quatre axes principaux :

- Processus de modélisation avec UML
- L'analyse des besoins : Nous définirons les exigences fonctionnelles et non fonctionnelles de l'application.
- La modélisation UML : Nous élaborerons les diagrammes nécessaires (cas d'utilisation, classes, séquences, etc.) pour décrire de manière détaillée l'architecture et le fonctionnement du système.
- La conception de la partie prédictive

Cette approche nous permettra d'assurer une conception cohérente, modulaire et conforme aux bonnes pratiques du génie logiciel.

3.2 Processus de modélisation avec UML

3.2.1 UML (Unified Modeling Language)

UML (Unified Modeling Language), se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML modélise l'ensemble des données et des traitements en élaborant des différents diagrammes. En clair, il ne faut pas designer UML en tant que méthode (Il y manque la démarche) mais plutôt comme une boîte d'outils qui sert à améliorer les méthodes de travail.[9]

3.2.2 Différents types de diagrammes d'UML

Les treize types de diagrammes UML peuvent ainsi être résumés sur la figure ci- dessous :

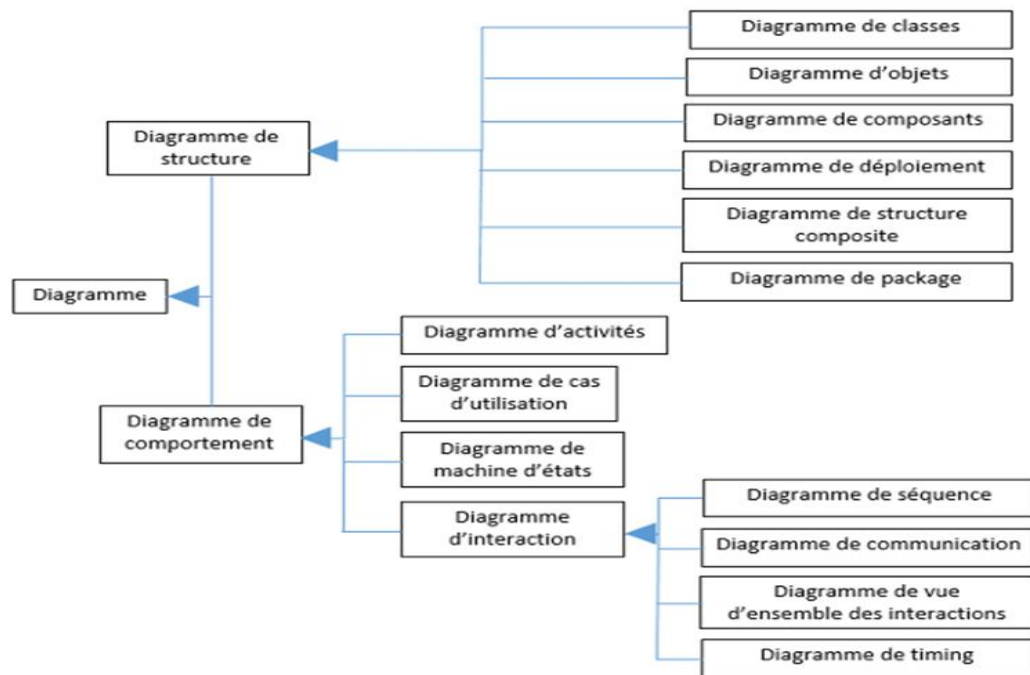


FIGURE 10: TYPES DE DIAGRAMMES D'UML

Dans cette phase, nous adoptons une démarche de modélisation orientée objet, articulée autour de trois diagrammes clés du langage UML : les cas d'utilisation pour la spécification fonctionnelle, les classes pour la structure du système, et les séquences pour le comportement dynamique :

3.3 Délimitation du périmètre du système à modéliser

Ici on aborde les acteurs de notre plateforme, la spécification des besoins ainsi que les différents diagrammes.

3.3.1 Les acteurs du système

Notre plateforme (EduIA) fait intervenir 5 acteurs principaux qui sont :

- L'administrateur
- L'enseignant
- L'élève
- Parent

3.3.2 Spécification des besoins

Dans cette partie, nous allons définir les différents besoins fonctionnels et non fonctionnels de notre plateforme de gestion scolaire avec analyse prédictive.

3.3.2.1 Besoins fonctionnels

Les besoins fonctionnels décrivent les services que le futur système devra fournir. Ils définissent les fonctionnalités essentielles attendues et représentent un engagement quant au comportement du système. Ainsi, la solution proposée devra satisfaire les besoins fonctionnels suivants [8] :

1. Gestion des utilisateurs

- Le système doit permettre l'enregistrement de nouveaux utilisateurs (élèves, enseignants, administrateurs, parents).
- Le système doit permettre à chaque utilisateur de s'authentifier via un identifiant et un mot de passe.
- Le système doit gérer les droits d'accès selon le rôle (enseignant, élève, administrateur, parent).
- Le système doit permettre la modification ou suppression d'un compte utilisateur.

2. Gestion des classes et des matières

- Le système doit permettre à l'administrateur de créer, modifier ou supprimer des classes.
- Le système doit permettre d'ajouter et gérer les matières enseignées.
- Le système doit permettre l'affectation des enseignants aux classes et aux matières.

3. Gestion des élèves

- Le système doit permettre la consultation des informations d'un élève (notes, absences, recommandations).
- Le système doit permettre aux élèves de consulter leurs notes, leurs recommandations, leur emploi du temps et imprimer leur résultat.

3. Gestion des enseignants

- Le système doit permettre d'affecter un enseignant à une ou plusieurs classe(s).

- Le système doit permettre la consultation du dossier d'un enseignant.
- Le système doit permettre aux enseignant de consulter leurs classes et leur emploi du temps.
- Le système doit permettre aux enseignant d'imprimer les informations.

4. Saisie et consultation des notes

- Le système doit permettre aux enseignants de saisir les notes des élèves pour chaque matière.
- Le système doit calculer automatiquement la moyenne par matière, par élève, et par période.

5. Gestion des absences

- Le système doit permettre aux enseignants de noter les absences ou présences des élèves.
- Le système doit générer des rapports d'assiduité pour chaque séance.

6. Analyse prédictive des performances

- Le système doit prédire la performance des élèves en se basant sur les données scolaires (notes (classe, domicile), absences, participation) pour chaque élève.

7. Système de recommandations personnalisées

- Le système doit générer des recommandations adaptées à chaque élève selon ses résultats et son profil.
- Le système doit proposer des ressources pédagogiques ciblées (exercices, documents)

8. Tableau de bord et visualisation des données

- Le système doit proposer un tableau de bord interactif aux enseignants et à l'administration.

9. gestion des emplois du temps

- Le système doit permettre aux admins de programmer les emplois du temps des classes et les examens.

10. gestion des parents

- Le système doit leur permettre de consulter les informations de ses enfants (notes, recommandations, emploi du temps).

11. communiquer

- Le système doit permettre aux utilisateurs de communiquer entre eux

12. gestion des ressources :

- Le système doit permettre aux enseignants de gérer les ressources pédagogiques qui seront recommandées aux élèves.

13. gestion des paramètres

- Le système doit permettre aux admins de paramétrer l'application

3.3.2.2 Besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins spéciaux du système. Parfois ils ne sont pas rattachés à un cas d'utilisation particulier mais déterminent tout le système. Ce sont des besoins en matière de performance, de type de matériel ou de type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type de SGBD, de système d'exploitation, etc.), de sécurité, de temps de réponse, etc. Le système doit garantir les besoins opérationnels suivants [8] :

1. Sécurité

- ❖ Le système doit garantir la confidentialité des données personnelles et scolaires des utilisateurs.
- ❖ L'accès aux fonctionnalités sensibles doit être protégé par un système d'authentification sécurisé (login/mot de passe).

- ❖ Les utilisateurs doivent uniquement accéder aux données correspondant à leur rôle.

2. Performance

- ❖ Le système doit permettre une navigation fluide, même avec un grand volume de données (élèves, notes, prédictions...).

- ❖ Le temps de réponse aux requêtes (connexion, affichage des notes, génération des graphiques...) ne doit pas dépasser quelques secondes.

3. Fiabilité

- ❖ Le système doit être capable de fonctionner sans interruption majeure.

4. Scalabilité

- ❖ Le système doit pouvoir supporter l'augmentation progressive du nombre d'utilisateurs, d'élèves, ou de données scolaires, sans perte de performance.

- ❖ L'ajout de nouvelles fonctionnalités (nouveaux modules, nouvelles analyses) doit être possible sans refonte complète.

5. Accessibilité et compatibilité

- ❖ Le système doit être accessible depuis différents navigateurs web modernes (Chrome, Firefox, Edge...).

- ❖ L'interface doit être responsive et facile à utiliser.

6. Ergonomie

- ❖ L'interface utilisateur doit être intuitive, claire et agréable à utiliser.

- ❖ Les utilisateurs doivent pouvoir naviguer dans l'application sans formation technique approfondie.

7. Maintenabilité

- ❖ Le système doit être développé avec une architecture claire et bien documentée afin de faciliter sa maintenance et ses évolutions futures.

3.4 Modélisation UML

Dans la suite de ce travail, nous allons détailler les besoins fonctionnels définis précédemment à travers l'élaboration des diagrammes de cas d'utilisation accompagnés de leurs scénarios respectifs. Ensuite, nous présenterons différents diagrammes (de séquence et de classes) afin d'offrir une vision plus claire et structurée de la solution envisagée.

3.4.1 Le diagramme de cas d'utilisation

En langage UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système. Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne [9].

La figure 11 suivante présente le diagramme de cas d'utilisation global de notre projet. Il permet d'identifier les différentes interactions entre acteurs et le système.

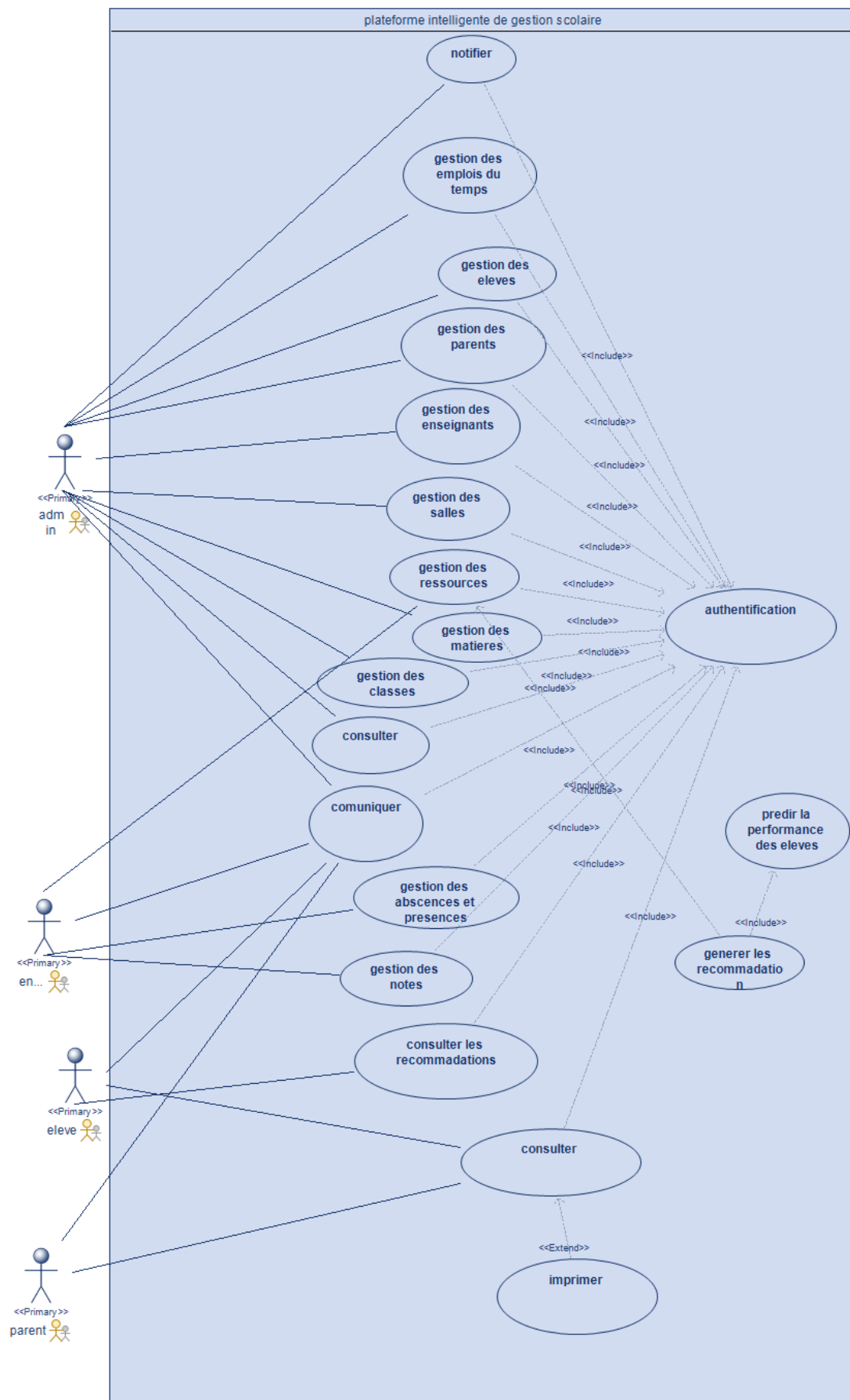


FIGURE 11:DIAGRAMME DE CAS D'UTILISATION DU SYSTEME

On remarque que l'administrateur peut gérer les utilisateurs, les classes etc., tandis que l'élève peut uniquement consulter et recevoir des recommandations. Cela illustre la séparation des rôles dans le système.

3.4.2 Le diagramme de séquence

Un diagramme de séquence est un diagramme UML (Unified Modeling Language) qui représente la séquence de messages entre les objets au cours interaction. Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction [10].

3.4.2.1 Description textuelle des cas d'utilisation

Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une description textuelle, car c'est une forme souple qui convient dans bien des situations [10].

Une description textuelle couramment utilisée se compose de deux parties :

✓ Première partie : Identification du cas

Elle doit contenir :

- Le nom du cas.
- Un résumé de son objectif.
- Les acteurs impliqués (principaux et secondaires) [10].

✓ Deuxième partie : Description du fonctionnement

Elle se présente sous la forme d'une séquence de messages échangés entre les acteurs et le système, et comprend une séquence nominale, correspondant au déroulement normal du cas [10].

Cette séquence se développe en trois points :

- **Les pré-conditions** : elles indiquent dans quel état est le système avant que se déroule la séquence.
- **L'enchaînement des messages.**
- **Les post-conditions** : elles indiquent dans quel état se trouve le système après le déroulement de la séquence nominale [10].

En complément, on peut également avoir :

- Une **séquence alternative** : elle diverge de la séquence nominale (c'est un embranchement dans une séquence nominale) [10].
- Une **séquence d'exception** : elle intervient lorsqu'une erreur se produit (l'enchaînement nominal s'interrompt, sans retour à la séquence nominale) [10].

➤ **Diagramme de séquence du cas « Authentification » :**

○ **Description du cas d'utilisation « s'authentifier »**

Le tableau suivant présente la description textuelle du cas d'utilisation **authentifier**, qui constitue une fonctionnalité essentielle du système.

Élément	Description
Cas d'utilisation	Authentification de l'utilisateur
But	Permettre à l'utilisateur d'accéder à l'application via un formulaire sécurisé.
Résumé	L'utilisateur remplit un formulaire d'authentification. Le système vérifie la non-vacuité et la validité des données, puis interroge la base pour vérifier l'existence de l'utilisateur afin de valider son accès à la plateforme. Selon les cas, il affiche l'accueil ou retourne une erreur.
Acteurs	Administrateur, Enseignant, Élève
Précondition	L'utilisateur doit avoir lancé l'application
Postcondition	L'utilisateur est authentifié et accède à l'interface correspondante
Scénario nominal	1. L'utilisateur lance l'application. 2. Le système retourne le formulaire d'authentification. 3. L'utilisateur remplit le formulaire avec ses identifiants. 4. Le système vérifie la complétude, validité et l'existence des données. 5. Le système affiche l'interface d'accueil.

Enchaînement alternatif	6. Si les données sont incorrectes ou incomplètes, le système retourne un message d'erreur et on retourne à l'étape 2. 7. Si les données sont valides mais inexistantes en base, le système retourne un message d'erreur et on retourne à l'étape 2.
--------------------------------	--

TABLEAU 2:DESCRIPTION TEXTUELLE DE S'AUTHTENTIFIER

Ce cas d'utilisation montre comment les acteurs du système s'authentifient pour accéder aux l'interfaces qui leur sont dédiées. Le scénario principal illustre clairement les étapes suivies pour effectuer les différentes opérations.

Ci -dessous le diagramme de séquence qu'illustre la figure 12 suivante décrit le cas d'utilisation « Authentification »

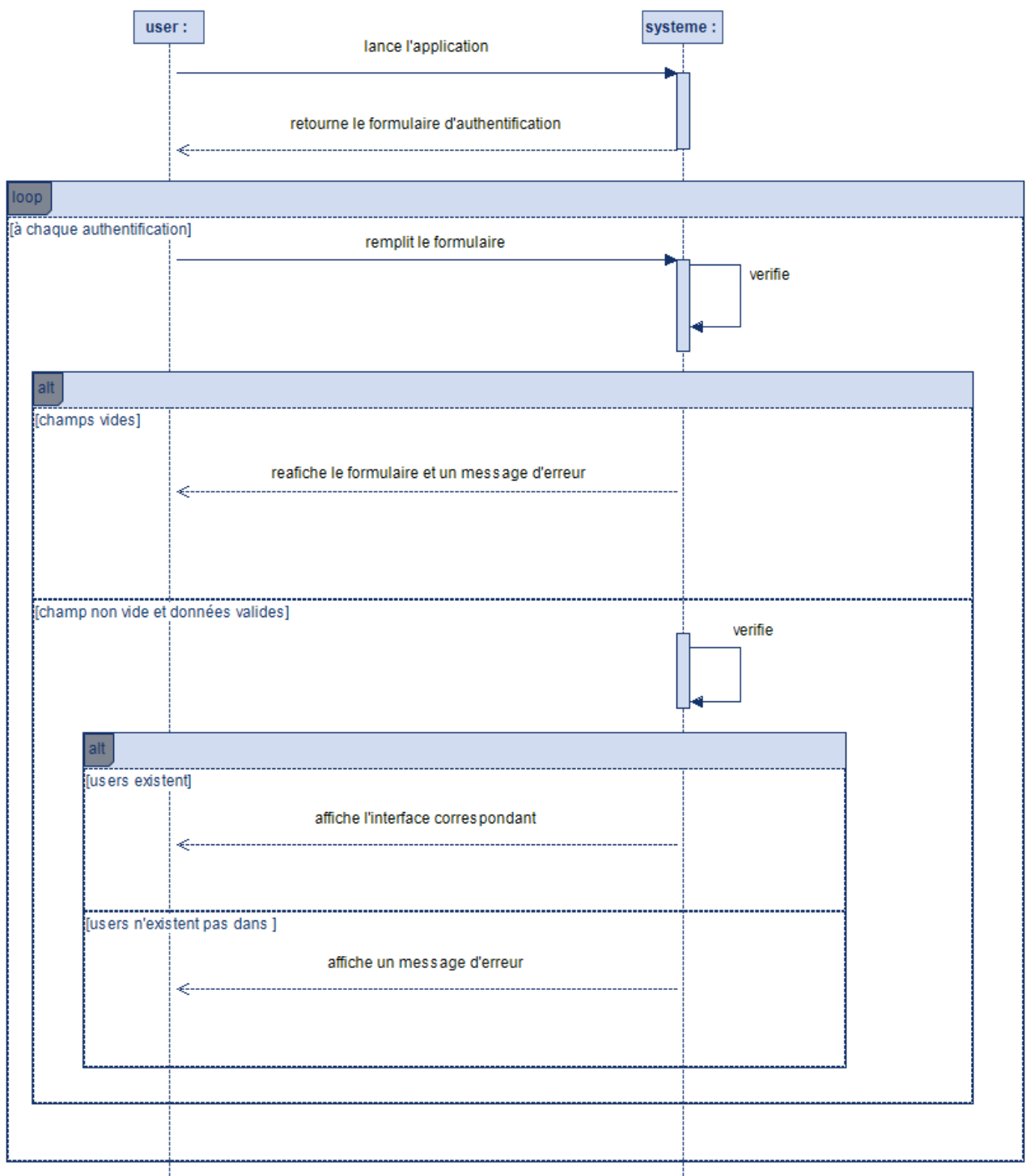


FIGURE 12:DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION <<AUTHENTIFIER>>

- Diagramme de séquence du cas « gestion d'élèves » :
 - Description du cas d'utilisation « gestion d'élève »

Le tableau 3 suivant présente la description textuelle du cas d'utilisation **inscrire l'élève**, qui constitue une fonctionnalité essentielle du système.

Élément	Description
Cas d'utilisation	Inscription d'un élève
But	Permettre à l'administrateur d'ajouter un nouvel élève dans le système scolaire.
Résumé	L'administrateur initie l'inscription en cliquant sur un bouton. Le système affiche un formulaire. Une fois rempli, le système vérifie les données : - Si l'élève n'existe pas, l'inscription est validée. - Sinon, un message d'erreur est affiché.
Acteurs	Administrateur
Précondition	L'administrateur est connecté au système
Postcondition	L'élève est inscrit dans la base de données et accessible au système
Scénario nominal	1. L'administrateur clique sur « Ajouter un élève ». 2. Le système retourne le formulaire d'inscription. 3. L'administrateur saisit les informations. 4. Le système vérifie la complétude et la validité. 5. Le système enregistre l'élève et actualise la liste. 6. Le système affiche : « Élève inscrit avec succès ».
Enchaînements alternatifs	7. Si les données saisies sont incomplètes ou invalides, le système retourne un message d'erreur et redemande à l'administrateur de corriger le formulaire. 8. Si les données sont valides mais correspondent à un élève déjà inscrit, la BDD retourne un message d'erreur : « Cet élève existe déjà », et le système revient à l'étape du formulaire.

TABLEAU 3: DESCRIPTION TEXTUELLE DE L'INSCRIPTION D'UN ELEVE

Ce cas d'utilisation montre que seul l'administrateur, après authentification, peut accéder à l'interface dédiée à l'inscription de l'élève. Le scénario principal illustre clairement les étapes suivies pour effectuer les différentes opérations.

Ci-dessous le diagramme de séquence qu'illustre la figure 13 suivante décrit le cas d'utilisation « gestion des élèves »

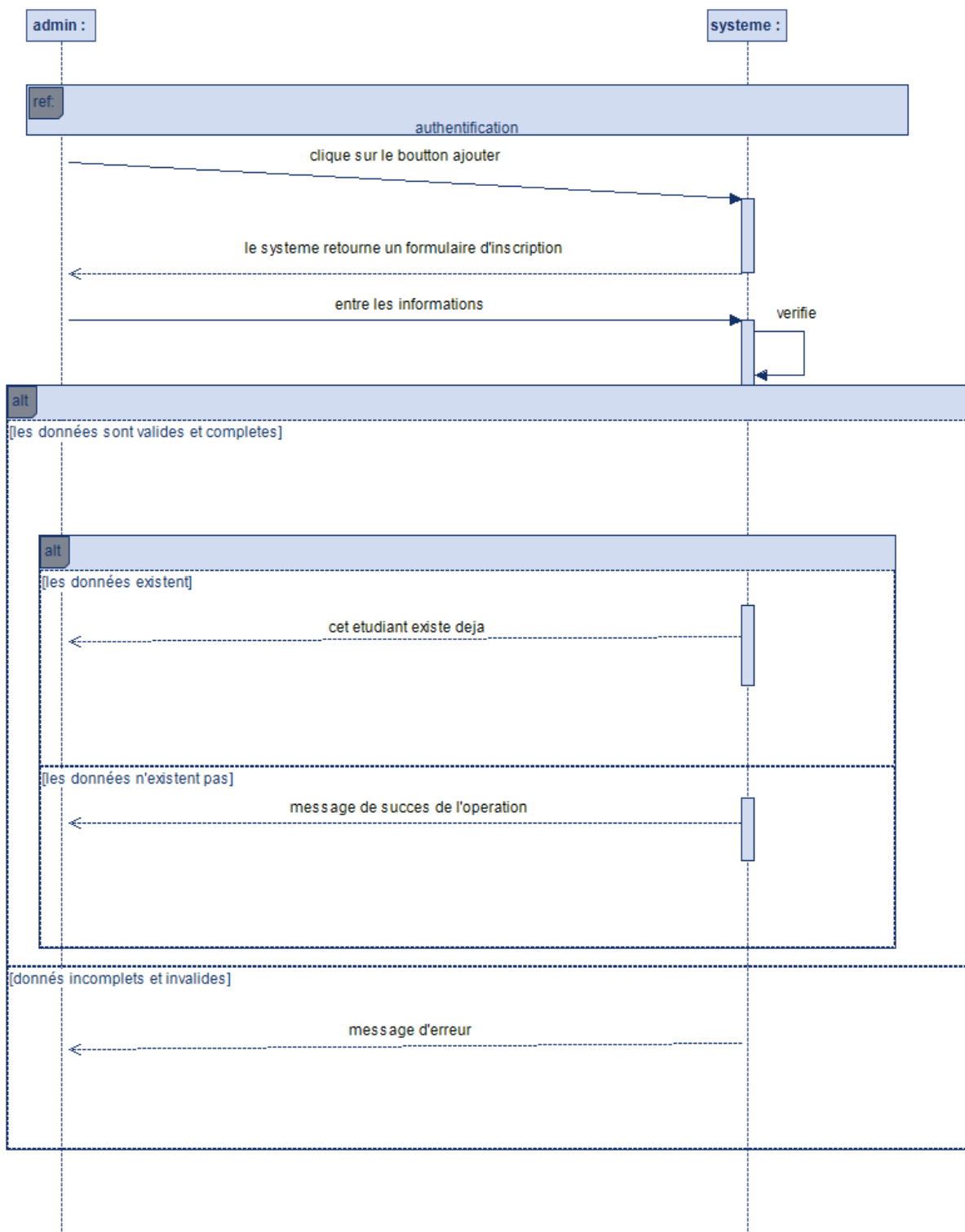


FIGURE 13: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION <<INSCRIRE L'ÉLÈVE>>

Cas d'utilisation 2 : modifier un élève

Le tableau 4 suivant présente la description textuelle du cas d'utilisation **modifier un élève**, qui constitue une fonctionnalité essentielle du système.

Élément	Description
Cas d'utilisation	Modification d'un élève
But	Permettre à l'administrateur de modifier un élève dans le système scolaire.
Résumé	L'administrateur, après avoir cherché l'élève en question et le trouve, clique sur le bouton modifier. Le système lui retourne un formulaire de modification avec toutes les informations concernant l'élève. Une fois les modifications apportées, le système vérifie la complétude des données et la validité des données après met à jour les informations de l'élève.
Acteurs	Administrateur
Précondition	L'administrateur est connecté au système
Post condition	L'élève est mis à jour dans la base de données
Scénario nominal	1. l'administrateur cherche l'élève en question. 2. L'administrateur clique sur le bouton « modifier un élève ». 3. Le système retourne le formulaire de modification contenant les informations de l'élève. 4. L'administrateur apporte les modifications. 5. Le système vérifie la complétude et la validité des données modifiées. 6. Le système met à jour les informations de l'élève. 7. Le système retourne un message de succès : « Élève modifié avec succès ».
Enchaînement alternatif	1. Si l'élève, après recherche n'existe pas, le système affiche un message << cet élève n'existe pas>> 2. Si les données saisies sont incomplètes ou invalides, le système retourne un message d'erreur et redemande à l'administrateur de corriger le formulaire.

TABLEAU 4: DESCRIPTION TEXTUELLE DE MODIFIER UN ELEVE

Ce cas d'utilisation montre que seul l'administrateur, après authentification, peut accéder à l'interface dédiée à la modification des informations d'un élève. Le scénario principal illustre clairement les étapes suivies pour effectuer les différentes opérations.

Ci-dessous le diagramme de séquence qu'illustre la figure 14 suivante décrit le cas d'utilisation « modifier un élève »

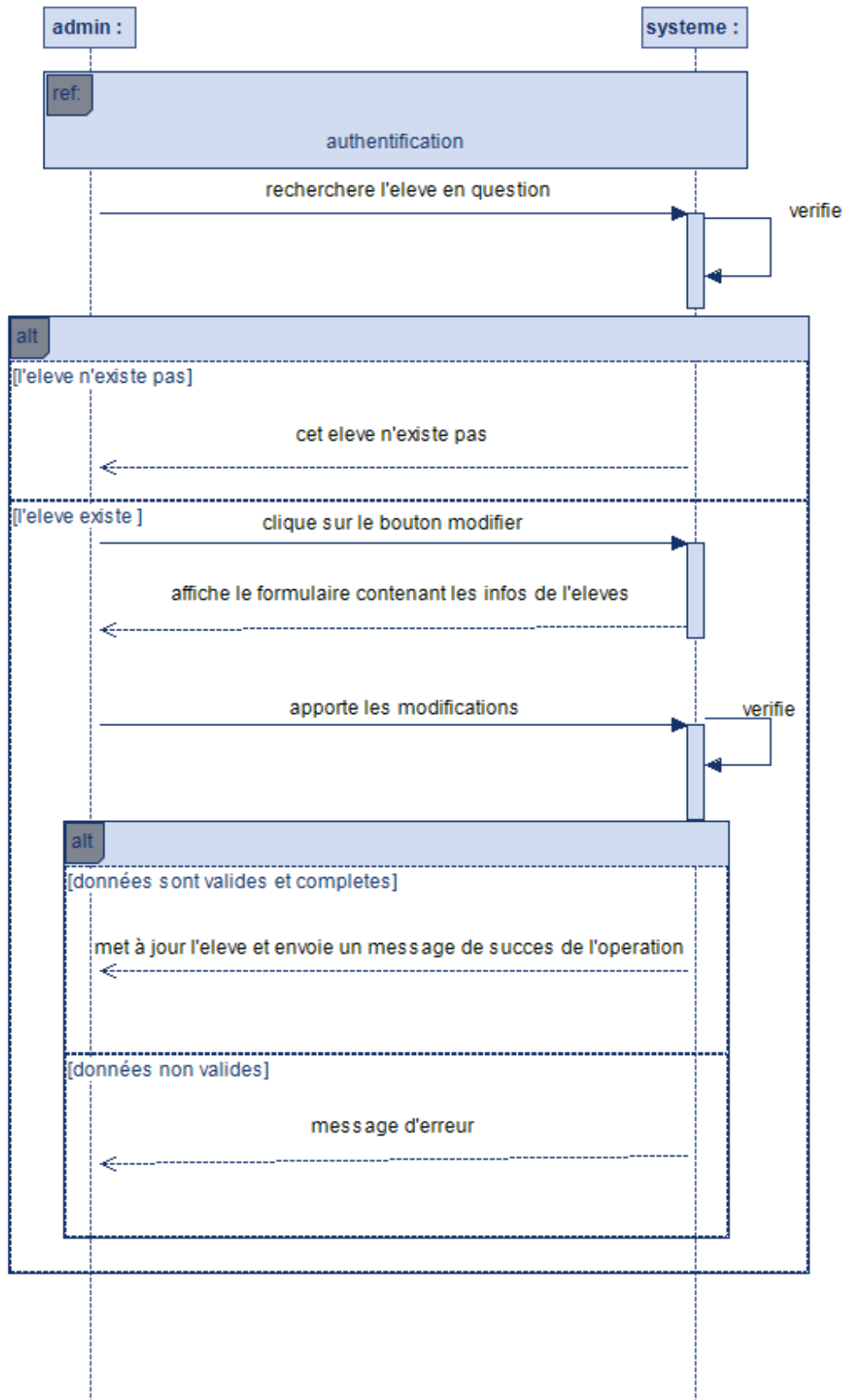


FIGURE 14:DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION <<MODIFIER UN ELEVE>>

Cas d'utilisation 3 : supprimer un élève

Le tableau 5 suivant présente la description textuelle du cas d'utilisation supprimer **un élève**, qui constitue une fonctionnalité essentielle du système.

Élément	Description
Cas d'utilisation	Suppression d'un élève
But	Permettre à l'administrateur de supprimer un élève dans le système scolaire.
Résumé	L'administrateur, après avoir cherché l'élève en question et le trouve, clique sur le bouton supprimer. Le système lui demande s'il veut réellement le supprimer. Une fois la suppression faite, le système met à jour la liste des élèves.
Acteurs	Administrateur
Précondition	L'administrateur est connecté au système
Post condition	L'élève est supprimé dans la base de données
Scénario nominal	1. l'administrateur cherche l'élève en question. 2. L'administrateur clique sur le bouton « supprimer un élève ». 3. Le système lui demande s'il veut réellement le supprimer. 4. Le système met à jour la liste des élèves. 7. Le système retourne un message de succès : « supprimé avec succès ».
Enchaînement d'échec	8. Si l'élève, après recherche n'existe pas, le système affiche un message <<cet élève n'existe pas>>

TABLEAU 5: DESCRIPTION TEXTUELLE DE SUPPRIMER UN ELEVE

Ce cas d'utilisation montre que seul l'administrateur, après authentification, peut accéder à l'interface dédiée à la suppression d'un élève. Le scénario principal illustre clairement les étapes suivies pour effectuer les différentes opérations.

Ci -dessous le diagramme de séquence qu'illustre la figure 15 suivante décrit le cas d'utilisation « supprimer un élève »

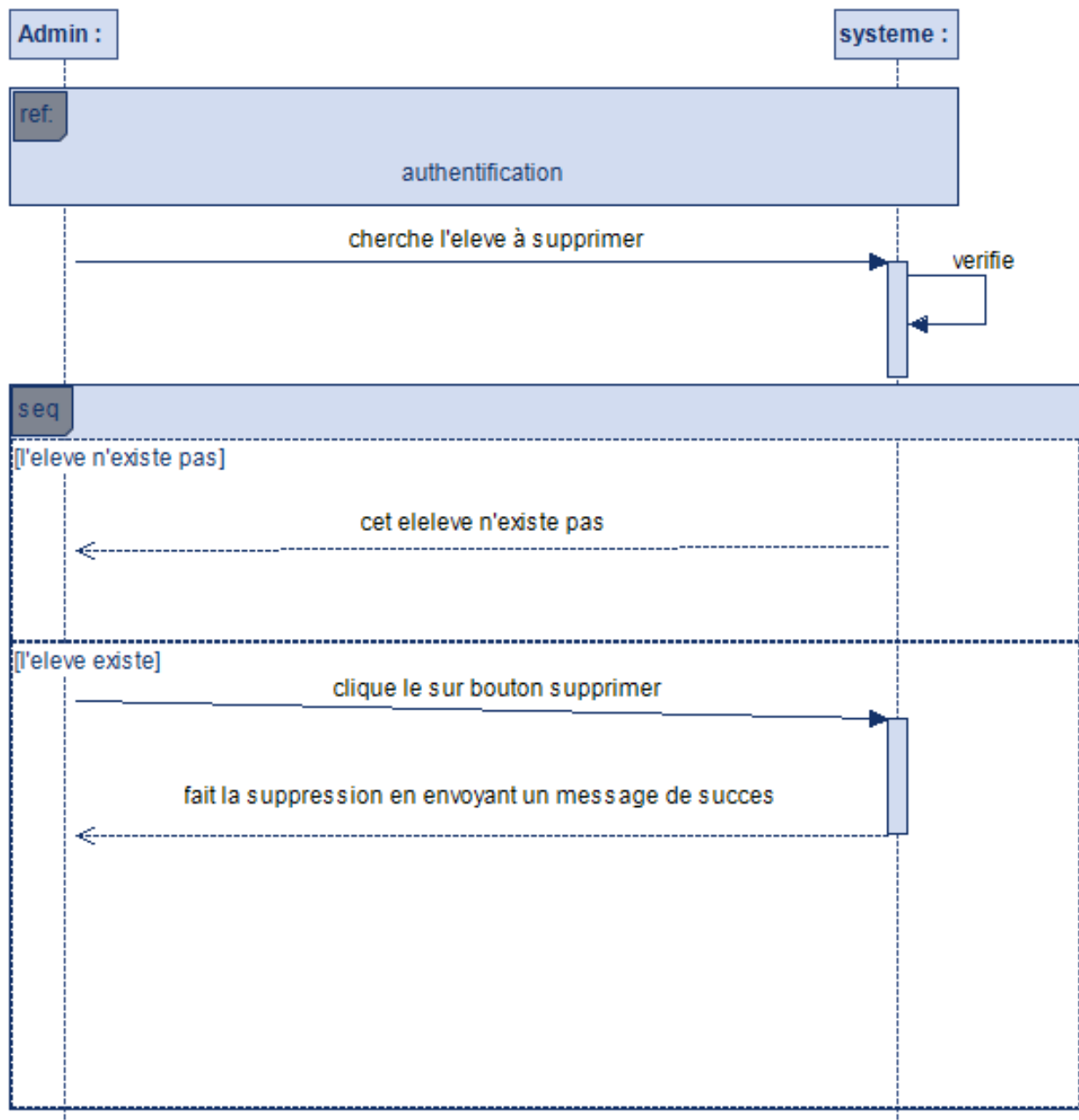


FIGURE 15: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION <<SUPPRIMER UN ELEVE>>

- **Diagramme de séquence du cas « prédire la performance de l'élève »**
 - **Description du cas d'utilisation « prédire la performance de l'élève »**

Le tableau 6 suivant présente la description textuelle du cas d'utilisation **prédire la performance d'un élève**, qui constitue une fonctionnalité essentielle du système.

Élément	Description
Cas d'utilisation	Prédire la performance des élèves
But	Analyser les données scolaires des élèves pour prédire leur niveau de performance académique à l'aide d'un modèle d'intelligence artificielle.
Résumé	Le système d'IA accède aux données des élèves (notes, absences, participation, etc.), applique un algorithme de prédiction, et attribue un niveau de performance à chaque élève (par exemple : médiocre, passable, bien, très bien). Ces résultats peuvent ensuite être utilisés pour la génération de recommandations personnalisées.
Acteurs	IA (acteur secondaire)
Précondition	Les données des élèves sont disponibles en base de données, et le modèle prédictif est déjà entraîné.
Postcondition	Le système affiche ou stocke les performances prédites des élèves pour un usage ultérieur (consultation, recommandations, alertes pédagogiques, etc.).
Scénario nominal	1. L'IA récupère les données des élèves. 2. L'IA applique le modèle de prédiction. 3. Le système enregistre et affiche les résultats des prédictions.
Enchaînements alternatifs	4. Si les données sont absentes, le système affiche un message d'erreur et interrompt la prédiction.

TABLEAU 6: DESCRIPTION TEXTUELLE DE PREDIRE LA PERFORMANCE DES ELEVES

Ce cas d'utilisation montre comment le modèle IA fait la prédiction des performances d'un élève en se basant sur certaines de ces données. Le scénario principal illustre clairement les étapes suivies pour effectuer les différentes opérations.

Ci-dessous le diagramme de séquence qu'illustre la figure 16 suivante décrit le cas d'utilisation « Prédire la performance des élèves »

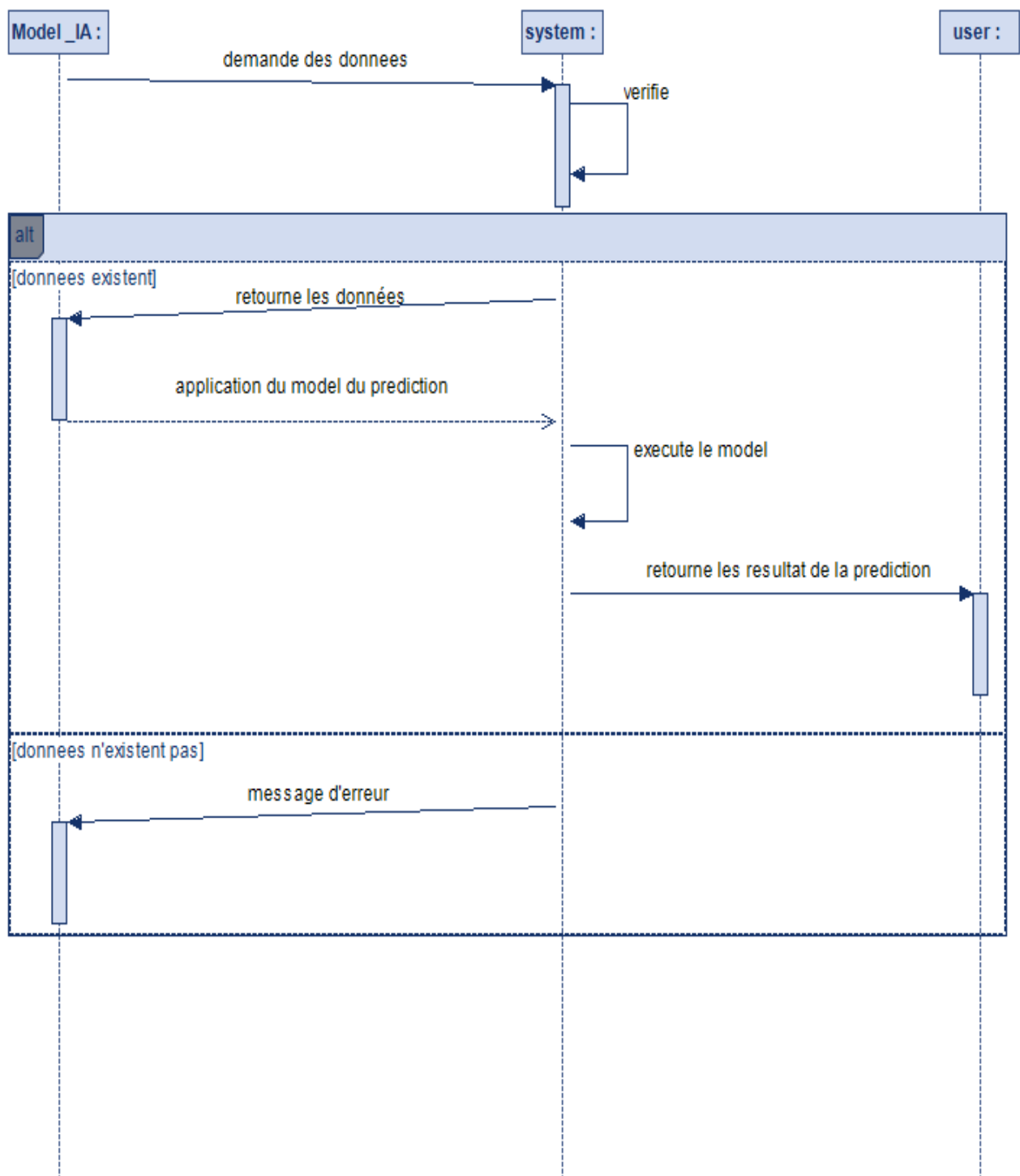


FIGURE 16: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION << PREDIRE LA PERFORMANCE DES ELEVES >>

- Diagramme de séquence du cas « générer les recommandations »
 - Description du cas d'utilisation « générer les recommandations »

Le tableau 7 suivant présente la description textuelle du cas d'utilisation **Générer les recommandations personnalisées**, qui constitue une fonctionnalité essentielle du système.

Élément	Description
Cas d'utilisation	Recommandation de ressources pédagogiques
But	Permettre au système de proposer des ressources adaptées à chaque élève en fonction de ses performances scolaires
Résumé	L'algorithme de recommandation demande les performances des élèves au modèle IA. Pour chaque élève, s'il dispose d'un historique de performances, l'algorithme génère des recommandations personnalisées. Pour les nouveaux élèves sans historique, le système n'affiche rien comme ressources.
Acteurs	Algorithme de recommandation, Système, Élève
Précondition	Les performances des élèves sont disponibles ou le système dispose de ressources
Post condition	Les recommandations sont générées, enregistrées dans le système et affichées à chaque élève
Scénario nominal	1. L'algorithme de recommandation demande les performances des élèves au modèle IA. 2. Le modèle IA retourne les performances. 3. Pour chaque élève : 3.1. Si l'élève est nouveau, le système n'affiche rien. 3.2. Sinon, l'algorithme associe des recommandations aux performances. 4. L'algorithme envoie les recommandations au système. 5. Le système enregistre les recommandations. 6. Le système affiche les recommandations à l'élève.

TABLEAU 7: DESCRIPTION TEXTUELLE DE GENERER LES RECOMMANDATIONS

Ce cas d'utilisation montre comment l'algorithme de recommandation recommande les ressources pédagogiques en se basant sur les le niveau de difficulté de chaque élève.

Ci -dessous le diagramme de séquence qu'illustre la figure 17 suivante décrit le cas d'utilisation « générer les recommandations »

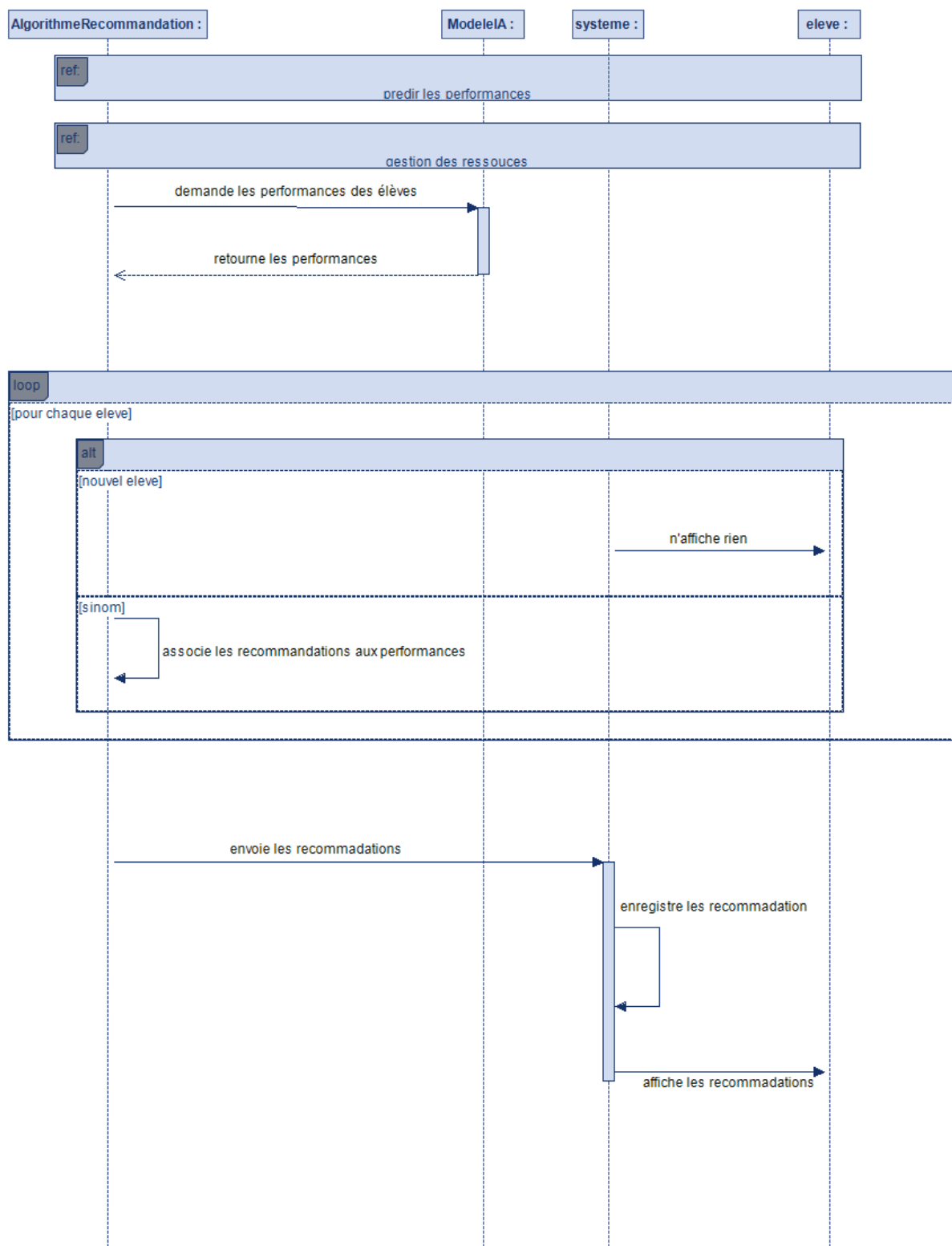


FIGURE 17: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION << GENERER LES RECOMMANDATIONS >>

3.4.3 Conception de la base de données

3.4.3.1 Diagramme de classes :

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet ou, à un niveau de détail plus important, les modules du langage de développement [9].

Le diagramme de classes est utilisé pour représenter la structure statique du système, en mettant en évidence les classes principales, leurs attributs, leurs méthodes ainsi que les relations entre elles. La figure suivante montre les classes essentielles pour la gestion des étudiants, des enseignants et des matières dans la plateforme.

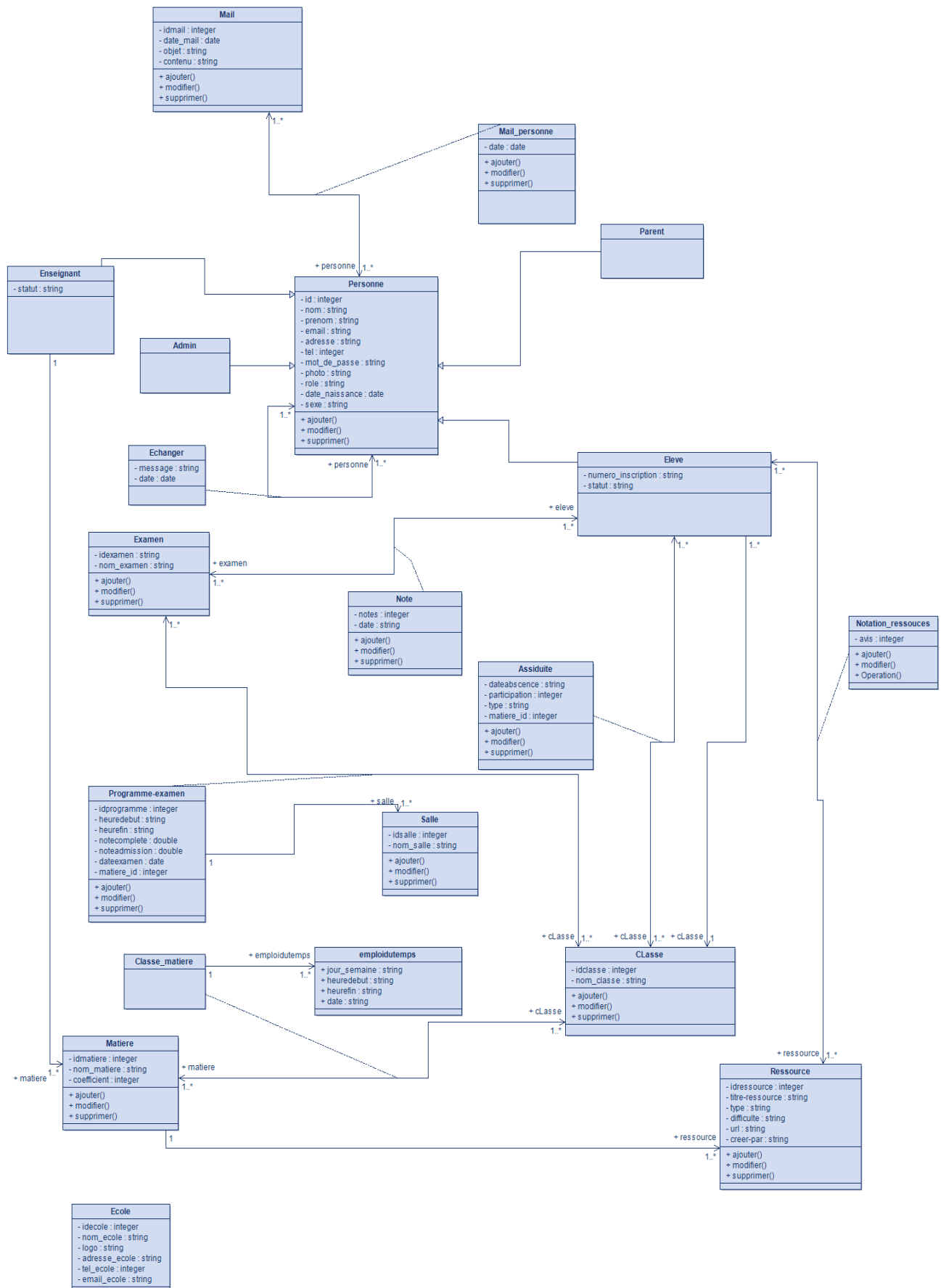


FIGURE 18:DIAGRAMME DE CLASSES DE LA SOLUTION

Le tableau suivant explique les différentes classes de la base de données.

Classe	Description
Élève	Représente un élève inscrit dans l'établissement. Contient les informations personnelles, le niveau scolaire, la classe associée, ainsi que les performances académiques (notes, absences).
Enseignant	Représente un membre du corps professoral
Admin	Représente un utilisateur ayant des droits d'administration sur la plateforme. Il gère les utilisateurs, les données scolaires et les paramètres du système.
Parent	Est la classe qui enregistre les parents des élèves
Mail	Représente un mail qui doit être envoyé à un ou plusieurs utilisateurs par administrateur.
Salle	Représente une salle physique de l'établissement où se tiennent les séances.
Matière	Représente une discipline enseignée dans l'école. Elle est liée à une ou plusieurs classes et enseignants.
École	Contient les informations globales sur l'établissement scolaire (nom, adresse, contact, logo).
Classe	Représente un groupe d'élèves regroupés par niveau (ex. : 2nde A, 1ère S...).
Assiduité	Est une classe d'association qui enregistre les absences des élèves aux différentes séances.
Ressources	Contient les ressources pédagogiques qui doivent être recommandées aux élèves.
Note	Représente une évaluation d'un élève dans une matière donnée. Contient la note chiffrée, la date, le type d'évaluation (examen, devoir, test, etc.).
Programme_examen	Elle représente la table qui permet de planifier les examens, de la même manière que pour les emplois du temps.
notation_ressouces	Est une classe d'association qui enregistre les ressources appréciées par les élèves
Echanger	Est une classe d'association qui enregistre les échanges entre les utilisateurs
Emploitudutemp	Est une classe qui décrit les emplois du temps de notre plateforme

TABLEAU 8: DESCRIPTION DE LA BASE DE DONNEES

3.4.3.2 Passage au modèle relationnel

Le modèle relationnel est le modèle logique de données qui correspond à l'organisation des données dans la base de données relationnelle. Un modèle est composé de relations encore appelées tables. Ces dernières sont décrites par des attributs aux champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses.

L'identifiant d'une relation est composé d'un ou de plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée [11].

❖ Règles de passage au modèle relationnel

Cette partie présente les règles qui permettent de créer un schéma relationnel à partir d'un modèle conceptuel exprimé par un diagramme de classes UML. Ces règles sont surtout basées sur l'existence des associations et la valeur des multiplicités de chaque association [11].

Nous donnons ci-après quatre règles (R1 à R4) pour traduire UML à un modèle relationnel équivalent. Il existe d'autres solutions de transformation, mais ces règles que nous présentons ci-dessous sont les plus simples et les plus opérationnelles.

- **Transformation de classe (R1)** : Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter de telle sorte que la relation dispose d'une clé primaire [8].

- **Transformations des associations** : Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations :

- ♦ **Association un à plusieurs (R2)** : Ce sont les plus nombreuses en général dans un modèle conceptuel ; chaque association un à plusieurs déclenche la génération d'une clé étrangère. Toute association un-à-plusieurs connecte une classe parent (multiplicité maximale 1) à une classe enfant (multiplicité maximale *). L'attribut identifiant de la relation parent migre dans la relation enfant et devient une clé étrangère dans la table enfant issue de l'association [11].

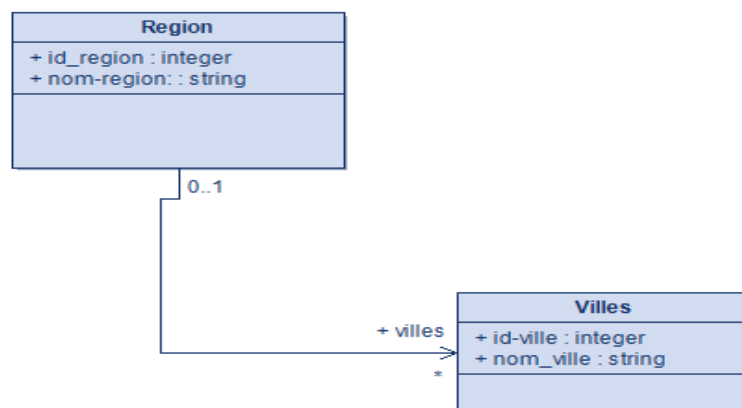
Exemple :

FIGURE 19: EXEMPLE D'ASSOCIATION « UN A PLUSIEURS »

Region(id_region,nom-region), **Ville**(id_ville,nom_ville,#id_region)

♦ **Associations plusieurs à plusieurs (R3)** : Toute association plusieurs-à-plusieurs devient une relation qui porte le nom de l'association. La clé primaire de cette relation est composée par le couple des identifiants des deux relations initiales qui sont déduites des classes connectées à l'association initiale. Chaque attribut clé primaire de la nouvelle relation est une clé étrangère vers la relation initiale associée.[11]

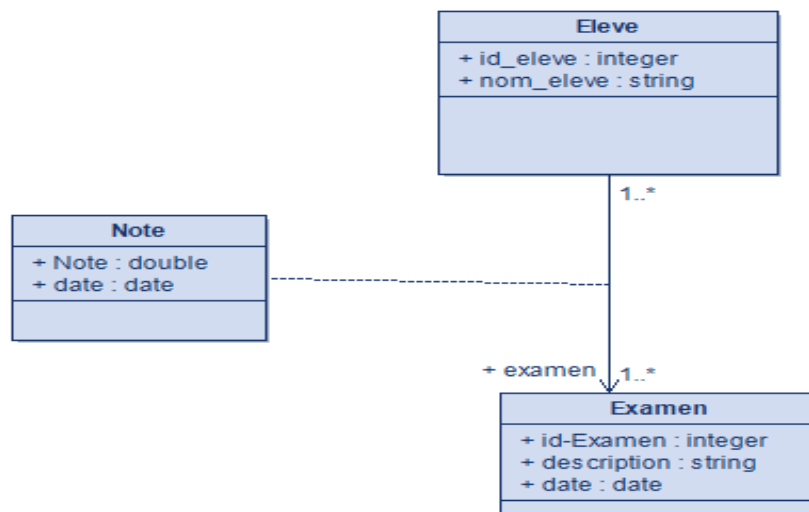
Exemple :

FIGURE 20: EXEMPLE D'ASSOCIATION « PLUSIEURS A PLUSIEURS »

Eleve(id_eleve,nom_eleve),**Examen**(id_examen,description,date),

Note(#id_eleve,#id_examen,note,date)

♦ **Association un à un (R4)** : Pour toute association un-à-un, considérez la relation du côté de la multiplicité 1..1 en tant que parent et appliquez la transformation d'une association un-à-plusieurs. Si aucune multiplicité n'a cette valeur (les deux valant 0..1), appliquez la transformation d'une association un-à-plusieurs sans considérer une relation en tant que parent en particulier. Si les deux multiplicités valent 1..1, vous devriez visiblement modifier votre schéma en fusionnant les deux classes en une seule.[11]

Exemple :

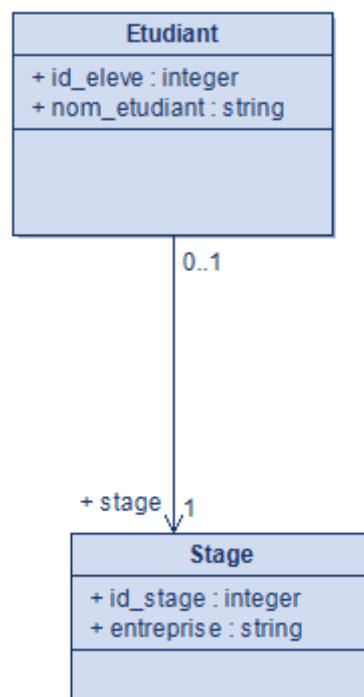


FIGURE 21:EXEMPLE D'ASSOCIATION « UN A UN »

Stage (id_stage, entreprise) **Etudiant** (id_etudiant, nom, #nStage)

- **Transformation de l'héritage** : Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes dont la décomposition par distinction, la décomposition descendante et celle ascendante décrites ci-dessous :

- **Décomposition par distinction** : Il faut transformer chaque sous-classe en une relation. La clé primaire de la super-

classe migre dans les relations issues des sous-classes et devient à la fois clé primaire et clé étrangère.[8]

Exemple :

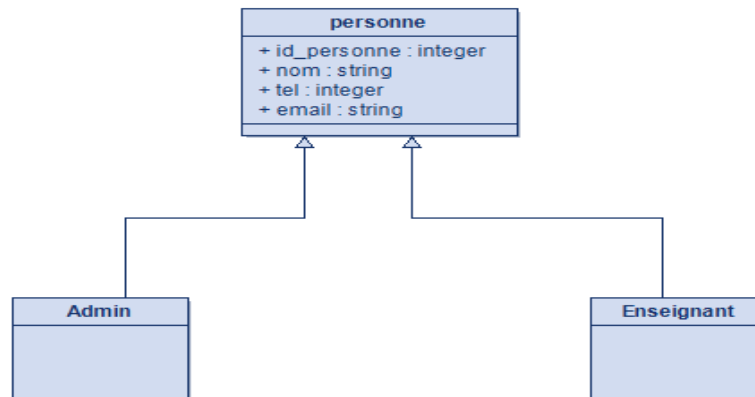


FIGURE 22: DECOMPOSITION PAR DISTINCTION D'UNE ASSOCIATION D'HERITAGE

Personne(id_personne, nom, tel, email), **Admin**(#id_personne),
Enseignant(#id_personne)

- **Décomposition descendante** : S'il existe une contrainte de totalité (toutes les instances d'une classe correspondent au moins à une des instances des classes liées) ou de partition (indique que toutes les instances d'une classe correspondent à une et une seule instance des classes liées) sur l'association, il est possible de ne pas traduire les relations issues de la super-classe. Il faut alors faire migrer tous ses attributs dans les relations issues des sous-classes [8].

Exemple :

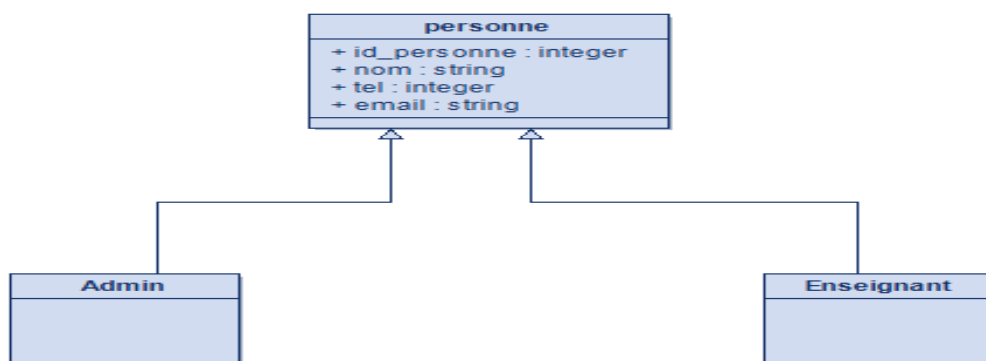


FIGURE 23: DECOMPOSITION DESCENDANTE D'UNE ASSOCIATION D'HERITAGE

Admin(id_personne, nom, tel, email) , **Enseignant**(id_personne, nom, tel, email)

- **Décomposition ascendante** : Il faut supprimer les relations issues des sous-classes et faire migrer les attributs dans la relation issue de la super-classe [8].

Exemple :

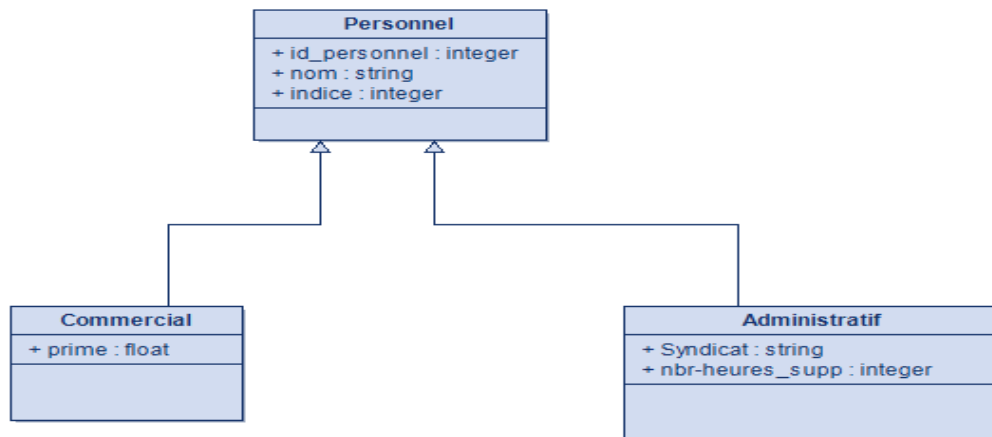


FIGURE 24: DECOMPOSITION ASCENDANTE D'UNE ASSOCIATION D'HERITAGE

Personnel(id_personnel, nom, indice, prime, syndicat, nbr_heures_supp)

Ainsi, dans le cadre de notre projet, nous avons choisi d'appliquer une décomposition ascendante d'une association d'héritage, afin de centraliser les attributs communs et ainsi faciliter l'évolution du système, en permettant leur modification à un seul endroit.

❖ **Modèle logique de données**

Le modèle logique décrit les tables, colonnes, types de données, et relations entre entités, sans encore tenir compte des contraintes physiques du SGBD (comme l'indexation, les performances, etc.) [11].

Ci-dessous la liste des tables de notre base de donnée.

Personne (idpersonne, nom, prenom, email, tel, rôle, password, adresse, photo, numero_inscription, statut)

Classe (idclasse, nom_classe, idecole)

Ecole (idecole, nom_ecole, logo, tel, adresse)

Matiere (idmatiere, nom_matiere, coefficient, # idenseignant)

Salle (idsalle, nom_salle)

Ressources (idressource, titreressource, titre, difficulte, url, #cree_par)

Assuidite (#ideleve, #idseance, date, absence, participation)

Examen (idexamen, description)

Programme_examen(idprogramme, heuredebut, heurefin, date_examen, note_admission, note_complete, #idsalle, #idexamen, #idmatiere, #idclasse)

Note (#idexamen, # ideleve, note, date)

Classe_matiere(#idclasse, #idmatiere)

Echanger(#emetteur_id, #recepteur_id, message, date)

Mail(idmail, datmail, objet, contenu)

Mail_personne(#personne_id, #idmail_id, date)

3.5 Conception du système d'analyse prédictive

Après avoir abordé les fondements théoriques de notre solution (EduIA) dans le chapitre I, nous allons à présent choisir un algorithme et expliquer un peu son fonctionnement.

3.5.1 Choix du type d'apprentissage

Pour notre projet (EduIA), nous avons choisi un modèle de machine Learning supervisé en particulier l'algorithme de forêt de décision. En effet, il nous semble approprié de prédire la réussite scolaire de l'élève à partir de données historiques, disponibles et étiquetées telles que les notes obtenues (classe, domicile), les absences, le degré de participation en classe. Ce que l'on appelle un apprentissage supervisé est d'autant plus pertinent qu'on dispose de jeux de données pour lesquelles on connaît à l'avance les résultats (ex : bas, moyen, haut), ce qui est souvent le cas en milieu scolaire.

En ce qui concerne le système de recommandation, nous avons opté pour une approche hybride selon deux cas :

- Les élèves qui ont déjà un historique, la machine leur recommande des ressources pédagogiques adaptées, en fonction des résultats scolaires de l'élève.
- Les élèves qui sont nouveaux et qui n'ont pas d'historique, la machine ne leur recommande rien.

Ces choix ont pour ambition de proposer des prédictions fiables et des recommandations adaptées pour améliorer le suivi des élèves dans leurs apprentissages et les aider à réussir à l'école.

3.5.2 Fonctionnement de notre choix

On explique ici comment l'algorithme que nous avons choisi fonctionne.

- ❖ **Collecte et préparation des données** : les données utilisées proviennent d'un dataset incluant :
 - Les **facteurs** qui sont :
 - ✓ **Note_classe** : la note de classe obtenue par chaque élève ;
 - ✓ **Note_domicile** : est la note que les élèves obtiennent à la suite d'exercices qu'ils font à la maison ;
 - ✓ **Note_participation** : est la note que chaque élève obtient suite à son interaction avec l'enseignant pendant chaque séance de cours ;
 - ✓ **Note_absence** : est le nombre d'absence que chaque élève obtient pendant tout un trimestre dans une matière donnée ;
 - La **sortie** : indique le niveau de risque de chaque élève. Elle est divisée en trois grandes classes qui sont :
 - Bas, moyen, élevé.
- ❖ **Prédiction en temps réel** : À chaque fois qu'un élève est mis à jour ou qu'un nouvel élève est ajouté, ses données sont soumises au modèle entraîné avec le dataset afin de **prédire sa mention**.
- ❖ **Génération de recommandations personnalisées** :
Selon la mention prédite, des recommandations éducatives sont générées automatiquement par le système.

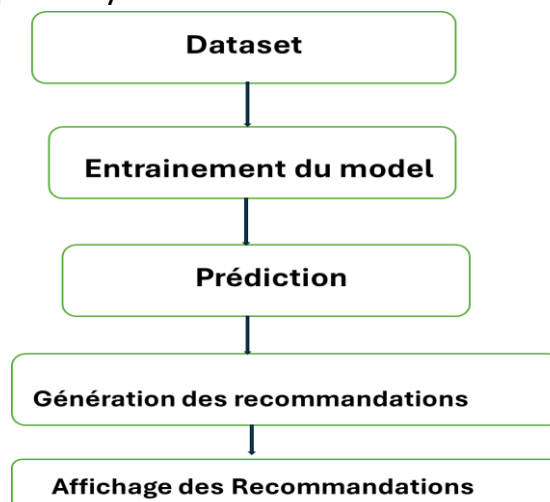


FIGURE 25: FONCTIONNEMENT DU MODEL

3.6 Conclusion

Dans ce chapitre, nous avons franchi une étape essentielle de notre projet en traduisant les besoins identifiés en une architecture claire et structurée. Grâce à l'utilisation du langage UML, nous avons pu concevoir les fondations techniques et fonctionnelles de notre plateforme intelligente de gestion scolaire.

La phase de spécification des besoins nous a permis de cerner précisément les attentes des différents acteurs du système (administrateurs, enseignants, élèves, parents) et de poser les contraintes clés liées à la sécurité, à la performance et à la pérennité du système. À travers les différents diagrammes UML (cas d'utilisation, séquence, classes), nous avons modélisé les interactions, les processus métiers ainsi que la structure logique des données.

Cette démarche a abouti à la construction d'un modèle relationnel cohérent et normalisé, garantissant une base de données bien structurée, prête à accueillir les développements à venir.

En somme, cette phase de conception constitue un socle solide sur lequel repose toute la suite du projet. Elle guide non seulement le développement technique, mais assure également que la solution réponde aux besoins réels des établissements scolaires de manière évolutive et durable.

Chapitre IV : Implémentation

4.1 Introduction

La phase d'implémentation constitue l'étape la plus déterminante du cycle de développement de notre plateforme, car c'est à partir de cette phase que nous obtenons le produit final issu de toutes les étapes précédentes.

Nous entamons ainsi la dernière partie du cycle, qui comprend l'implémentation et les tests, en vue de produire une solution fonctionnelle et exploitable par les utilisateurs finaux.

Dans ce chapitre, nous présenterons l'environnement de développement, les outils et langages de programmation utilisés, ainsi que l'architecture matérielle mise en place. Nous procéderons ensuite à l'implémentation des cas d'utilisation définis et terminerons par leur phase de test et de validation.

4.2 Environnement de travail

4.2.1 Environnement matériel

Pour le développement de notre application, nous avons utilisé un ordinateur portable de marque **LENOVO**, ayant les caractéristiques techniques suivantes :

- **Nom de l'appareil** : DESKTOP-H3CSBGH
- **Processeur** : Intel® Core™ i5-8250U CPU @ 1.60 GHz (jusqu'à 1.80 GHz)
- **Mémoire RAM installée** : 8 Go
- **Architecture** : Système d'exploitation 64 bits, processeur x64
- **Écran** : Non tactile

Cet environnement matériel a permis d'assurer un développement fluide et efficace de la plateforme.

4.2.2 Environnement logiciel

➤ **Visual Studio Code (VS Code)**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Ses fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, la refactorisation du code et Git intégré [12].

C'est un outil que nous avons utilisé comme notre environnement principal de développement.

➤ XAMPP

Est un environnement de développement libre, facile à installer, qui regroupe Apache (serveur web), MariaDB (système de gestion de base de données), ainsi que les interprètes pour les langages PHP et Perl [13].

Nous l'avons utilisé comme serveur local de test, incluant Apache pour servir nos pages web et MySQL pour héberger la base de données de la plateforme.

➤ MySQL

Est un système de gestion de bases de données relationnelles (SGBDR) open source développée initialement par la société suédoise MySQL AB, aujourd'hui propriété d'Oracle Corporation [14].

C'est un outil que nous avons utilisé pour stocker nos données.

➤ Modelio

Est un outil open source de modélisation qui prend en charge la création de diagrammes UML, BPMN¹ et d'autres types de modèles. Il permet aux développeurs et aux analystes de concevoir des architectures logicielles, des processus métiers et de générer du code à partir des modèles [15].

Nous l'avons utilisé dans notre projet pour dessiner tous les diagrammes.

4.3 Choix des technologies utilisées

Cette section concerne les technologies que nous avons utilisées pour développer notre plateforme.

➤ JavaScript

Est un langage de programmation principalement utilisé pour rendre les pages web interactives. Il permet d'ajouter des fonctionnalités dynamiques aux sites web, comme des animations, des formulaires interactifs, ou encore la mise à jour automatique de contenus sans recharger la page [17].

Nous l'avons utilisé principalement pour rendre l'interface plus interactive: validation des formulaires côté client, requêtes AJAX pour mettre à

¹ Selon [16] **BPMN (Business Process Model and Notation)** est une méthode de modélisation graphique qui permet de représenter clairement les processus métiers d'une organisation sous forme de diagrammes. Son but est de rendre les processus facilement compréhensibles aussi bien par les analystes métiers, les développeurs que par les décideurs.

jour certaines parties de la page sans rechargement (comme les tableaux de notes), et gestion dynamique du tableau de bord.

➤ **HTML (Hypertext Markup Language)**

Est un langage de balisage simple utilisé pour mettre en page le contenu d'une page web et la rendre disponible en ligne [17].

Nous l'avons utilisé pour structurer nos pages.

➤ **CSS (Cascading Style Sheets)**

Est un langage informatique utilisé pour mettre en forme et styliser des pages web écrites en HTML ou XML. Il définit l'apparence visuelle des éléments (couleurs, polices, espacements, dispositions...) à l'aide de « feuilles de style en cascade » (.css), permettant ainsi de séparer la structure du contenu de sa présentation [17].

CSS nous a permis de les styliser de manière claire et responsive, assurant une meilleure expérience utilisateur.

➤ **PHP (Hypertext Preprocessor)**

Est un langage de programmation côté serveur principalement utilisé pour développer des applications web dynamiques. Il permet de générer du contenu HTML, interagir avec des bases de données, gérer des sessions, et construire des sites interactifs. PHP est largement utilisé en combinaison avec MySQL et est intégré dans de nombreux systèmes de gestion de contenu comme WordPress [18].

Il nous a permis de développer toute la logique métier du projet : gestion des utilisateurs, traitement des données, communication avec la base de données, ainsi que les appels au moteur RubixML pour la prédiction.

➤ **Laravel**

Est un Framework PHP open source destiné au développement d'applications web. Il suit le modèle MVC (Modèle-Vue-Contrôleur) et offre une syntaxe élégante et expressive. Laravel simplifie de nombreuses tâches courantes comme l'authentification, la gestion des sessions, le routage, et facilite l'accès aux bases de données, tout en permettant un développement rapide et sécurisé [18].

C'est un Framework PHP puissant que nous avons utilisé pour développer notre projet.

4.4 Mise en œuvre de l'architecture MVC

L'architecture **MVC (Modèle-Vue-Contrôleur)** a été adoptée afin de structurer le projet de manière modulaire, maintenable et évolutive. Elle permet de séparer la logique métier (Modèle), la logique de présentation (Vue), et la gestion des actions utilisateur (Contrôleur) [18].

4.4.1 Modèle

Le modèle est chargé de gérer la logique métier de l'application. Il s'occupe principalement de la gestion des données (accès, stockage, modification) ainsi que du traitement des règles qui encadrent ces données. Il a pour but d'offrir au contrôleur une interface d'utilisation simple. On y retrouve notamment des algorithmes complexes et des requêtes SQL [18].

Quelques modèles utilisés :

- **User**
 - Représente les utilisateurs (administrateurs, enseignants, étudiants, parents). Gère les relations avec les rôles, les classes, les notes et les présences.
- **ClassModel**
 - Représente les classes. Gère la création, modification, suppression et l'affectation des étudiants.
- **StudentAttendance**
 - Enregistre et gère les données de présence des étudiants.
- **ExamScheduleModel**
 - Gère la planification des examens, les horaires et les conflits d'emplois du temps.
- **SchoolSetting**
 - Modèle des paramètres globaux de l'établissement (nom, logo, année scolaire, etc.).
- **MarksRegisterModel**
 - Gère l'enregistrement et la consultation des notes des étudiants pour les différents examens.

4.4.2 Vue

La vue est entièrement dédiée à l'affichage. Elle réalise très peu de traitements : elle se contente d'utiliser les données qui lui sont transmises pour produire une interface utilisateur. Principalement composée de code HTML, elle peut contenir quelques instructions PHP légères (comme des boucles ou des conditions simples) pour, par exemple, afficher une liste d'éléments [18].

Quelques vues :

- **resources\views\admin\dashboard.blade.php**
 - Tableau de bord de l'administrateur : statistiques générales, alertes et raccourcis.
- **resources\views\admin\risk_table.blade.php**
 - Affichage des étudiants classés par niveau de risque (faible, moyen, élevé), basé sur leurs performances.
- **resources\views\teacher\attendance\student.blade.php**
 - Interface permettant aux enseignants de consulter et saisir la présence des étudiants.
- **resources\views\student\dashboard.blade.php**
 - Tableau de bord étudiant : classe, ressources pédagogiques, notes, emploi du temps, etc.
- **resources\views\admin\settings\index.blade.php**
 - Interface de configuration des paramètres généraux de l'établissement.

4.4.3 Contrôleur

Le contrôleur agit comme intermédiaire entre l'utilisateur, le modèle et la vue. Il reçoit les requêtes de l'utilisateur, fait appel au modèle pour exécuter des actions (comme récupérer des articles ou supprimer un commentaire), puis transmet les résultats à la vue. Il prépare ainsi les données nécessaires à l'affichage et retourne la page HTML générée à l'utilisateur [18].

Quelques contrôleurs :

- **DashboardController**
 - Centralise les données du tableau de bord selon le rôle connecté (admin, élève, enseignant).
- **StudentController**
 - Gère les opérations sur les étudiants : ajout, mise à jour, suppression, listing, affectation aux classes.
- **TeacherController**
 - Gère les opérations sur les enseignants et la gestion des ressources pédagogiques associées.
- **ExaminationsController**
 - Contrôle la gestion des examens : création de sessions, saisie des notes, publication des résultats.

- **SettingsController**
 - Permet à l'administrateur de modifier les paramètres de l'école : logo, nom, année scolaire, etc.
- **AttendanceController**
 - Gère les présences des étudiants. Permet de les enregistrer, consulter ou générer des rapports d'assiduité.

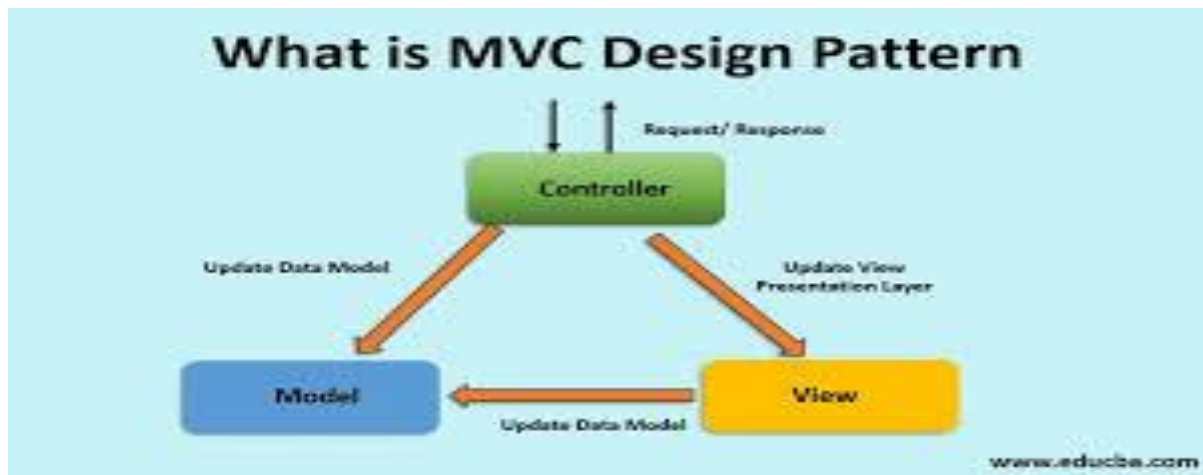


FIGURE 26:ARCHITECTURE MVC

4.5 Rubix ML

Rubix ML Est une bibliothèque gratuite d'apprentissage automatique open source (ML) [19].

Nous l'avons utilisé pour entraîner notre modèle prédictif à partir des données des élèves (notes, absences, participations) et générer automatiquement des mentions (faible, moyen, élevé) et des recommandations personnalisées.

4.6 Conception des interfaces utilisateur (IHM)

Dans cette partie nous allons faire une présentation illustrative des interfaces de la plateforme :

- **L'interface d'authentification** : la figure suivante représente l'interface d'authentification de notre plateforme. Elle permet aux utilisateurs finaux d'accéder aux fonctionnalités qui leur sont accordées.

FIGURE 27: INTERFACE D'AUTHENTIFICATION

- **Interfaces de gestion des élèves** : Ces interfaces nous permettent de gérer correctement un élève donné c'est-à-dire l'inscrire, modifier ses infos ou le supprimer.

La figure 28 ci-dessous montre l'interface qui permet à un admin d'inscrire un élève.

FIGURE 28:INTERFACE D'INSCRIPTION D'UN ELEVE

La figure 29 ci-dessous montre l'interface qui permet de visualiser la liste des élèves.

Liste des élèves (Total: 15) Ajouter un nouvel élève Exporter en PDF

Rechercher un élève

Nom: Nom de famille: E-mail: Numéro d'admission: Classe: Genre:

Groupe sanguin: Numéro de téléphone: Date d'admission: Date de création: Statut:

Rechercher Effacer

Liste des élèves

Photo de profil	Nom de l'élève	Nom du parent	E-mail	Numéro d'admission	Classe	Genre	Date de naissance	Notes	Date d'admission	Groupe sanguin	Statut	Date de création	Action
	Slade Duke	Ashton Rivera1 Burch	lewjyjf@mailinator.com	103	Fourth Year	Male	2017-12-17	Eos qui est dolores	2022-11-01	A-	Actif	06-05-2025	Modifier Supprimer Envoyer un message
	Solomon Cabrera	Nolan	qezymo@mailinator.com	799	Third Year	Male	2000-04-21	Non eveniet et enim	2012-04-30	AB-	Actif	05-05-2025	Modifier Supprimer Envoyer un message
	Destiny Yang	Hodges	redihyary@mailinator.com	931	First Year	Male	1971-07-11	Et consequat Nulla	1979-10-19	A-	Inactif	05-05-2025	Modifier Supprimer Envoyer un message

FIGURE 29: LISTE DES ELEVES

La figure 30 ci-dessous montre l'interface qui permet à un admin de modifier les infos d'un élève.

Modifier un élève

Prénom: Slade Duke
 Nom de famille: Acosta
 Numéro d'admission: 103
 Classe: Fourth Year
 Genre: Homme
 Date de naissance: 12/17/2017
 Notes: Eos qui est dolores
 Numéro de téléphone: 26653566664
 Date d'admission: 11/01/2022
 Photo de profil: Choose File No file chosen
 Groupe sanguin: A-
 Statut: Actif
 E-mail: lewjyjf@mailinator.com
 Mot de passe: Entrez le mot de passe

FIGURE 30: FORMULAIRE DE MODIFICATION

- **Interfaces de gestion des notes** : cette interface nous permet de noter chaque élève dans chaque examen.

Nom de l'élève	Science	Physique	Math	Arabic	Action
Destiny Yang Hodges	Travail en classe: 12 Devoirs: 12 Examen: 11 Moyenne: 11.5 Succès	Travail en classe: 15 Devoirs: 13 Examen: 12 Moyenne: 13 Succès	Travail en classe: 12 Devoirs: 13 Examen: 11 Moyenne: 11.75 Succès	Travail en classe: 17 Devoirs: 15 Examen: 14 Moyenne: 15 Succès	Enregistrer 12.35 Succès
Willa Carrillo Whitley	Travail en classe: 10 Devoirs: 10 Examen: 10 Moyenne: 10 Échec	Travail en classe: 10 Devoirs: 10 Examen: 10 Moyenne: 10 Échec	Travail en classe: 5 Devoirs: 5 Examen: 5 Moyenne: 5 Échec	Travail en classe: 5 Devoirs: 5 Examen: 5 Moyenne: 5 Échec	Enregistrer 4.13 Échec

FIGURE 31: FORMULAIRE DE NOTATION

▪ **Interfaces de la liste des recommandations :** cette interface nous affiche les recommandations adaptées aux besoins individuels de chaque élève.

FIGURE 32: FORMULAIRE DES RECOMMANDATIONS

▪ **Interfaces de gestion des absences :** cette interface nous permet de noter les présences et absences de chaque élève.

IA School

Fuller Burgess

Tableau de bord

Mes Classes

Mes Éléves

Calendrier des Examens

Mon Calendrier

Registre des Notes

Présence

Présence des élèves

Rapport de Présence

Tableau d’Affichage

Ressources

Mon compte

Changer Mot de Passe

Déconnexion

Présence des élèves

Rechercher

Classe

First Year

Matière

Science

Date

05/18/2025

Rechercher

Effacer

Liste des élèves

Id	Nom de l'élève	Présence	Participation (0-5) (0-5)
		☐ Présent ☐ En retard ☐ Absent	
45	Destiny Yang Hodges	☐ Présent ☐ En retard ☐ Absent	0
44	Willa Carrillo Whitley	☐ Présent ☐ En retard ☐ Absent	0
37	Neville Miller Cross	☐ Présent ☐ En retard ☐ Absent	0
36	Phillip Rich Ramos	☐ Présent ☐ En retard ☐ Absent	0
35	Casey Pate Rice	☐ Présent ☐ En retard ☐ Absent	0
34	Quin Santana Lowery	☐ Présent ☐ En retard ☐ Absent	0

FIGURE 33:FORMULAIRE DE NOTATION DES ABSENCES

- **Interface pour paramétrer la plateforme :** cette interface nous permet de paramétrer la plateforme.

IA School

Admin

Tableau de bord

Administrateur

Enseignant

Élève

Parent

Académique

Examens

Présence

Communication

Mon compte

Paramètres

Changer Mot de Passe

Déconnexion

Paramètres de l'école

Nom de l'école

IA School

Logo de l'école

Choose File

No file chosen

Taille recommandée : 50x50 pixels

Adresse

Tlemcen

Numéro de téléphone

+213552711669

Adresse e-mail

sms@gmail.com

Enregistrer les paramètres

Copyright © 2024-2025 **Small & Moussa**. All rights reserved.

Contact us: sms@gmail.com

FIGURE 34: FORMULAIRE DE PARAMETRAGE

La figure 35 ci-dessous montre le tableau de bord de notre plateforme.

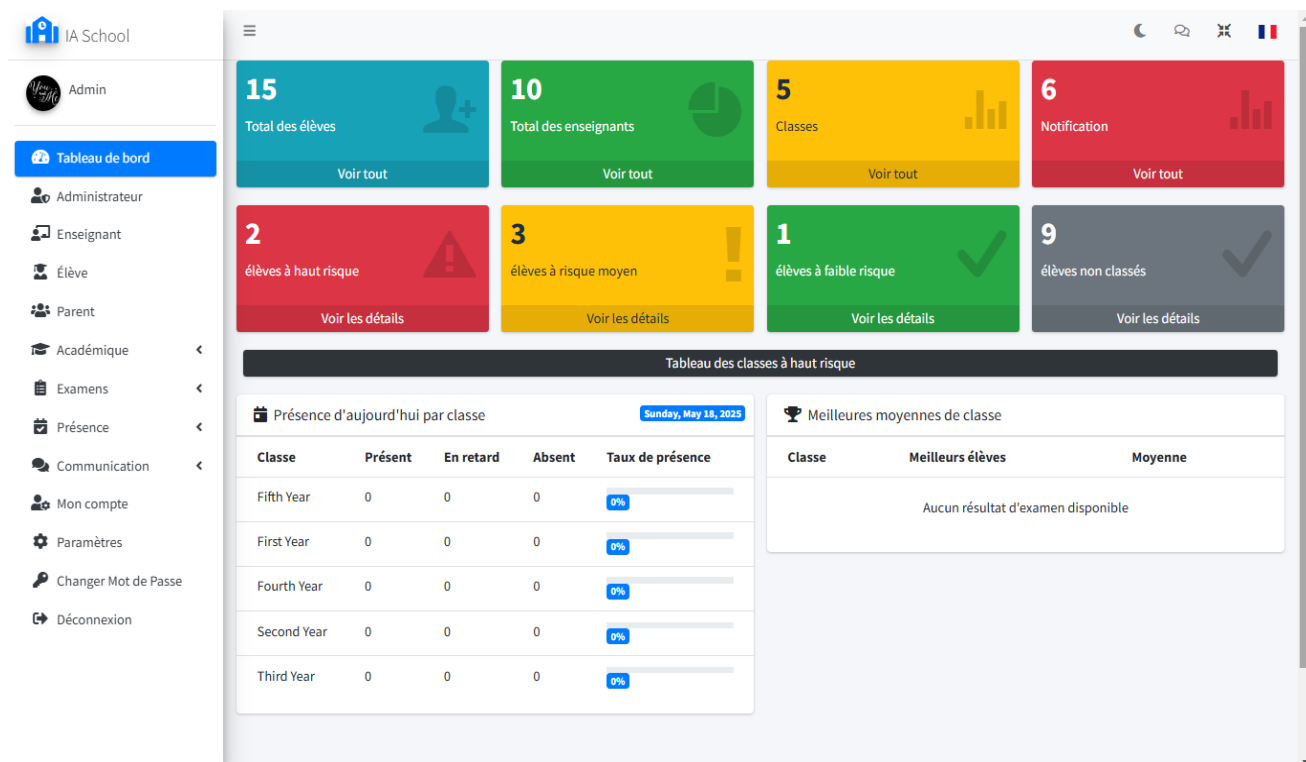


FIGURE 35:TABLEAU DE BORD D'ADMINISTRATION

Conclusion générale et perspectives

Ce projet de Plateforme intelligente de gestion scolaire avec analyse prédictive s'inscrit dans une logique d'innovation par rapport aux solutions existantes telle qu'Edmodo, Knewton Alta ou DreamBox Learning, en proposant une approche plus intégrée et mieux adaptée aux besoins actuels du système éducatif.

En combinant des fonctionnalités de gestion scolaire classique avec des technologies d'intelligence artificielle, notre plateforme se distingue par sa capacité à :

- Automatiser les principales fonctions de gestion scolaire (inscriptions, absences, notes, emplois du temps, paramètres généraux) à travers une architecture web robuste basée sur Laravel ;
- Analyser les données pédagogiques (notes, absences, participation) ;
- Prédire les performances des élèves ;
- Et générer des recommandations personnalisées ;

Notre contribution principale réside dans l'intégration, dans un même système, une dimension prédictive et recommandation personnalisée rarement exploitées dans les systèmes traditionnels existants.

Ce travail a mobilisé des méthodes de machine learning via la bibliothèque Rubix ML en PHP, dans une architecture web modulaire, intuitive et accessible.

✓ Perspectives

Afin d'enrichir et de pérenniser cette solution, plusieurs axes d'amélioration peuvent être envisagés :

- **Amélioration des modèles de prédiction** : tester d'autres algorithmes plus performants pour une meilleure précision des résultats.
- **Extension du périmètre des données** : intégrer d'autres paramètres influents comme le contexte socio-économique, l'implication parentale, ou les activités extrascolaires.
- **Application mobile** : développement d'une version mobile de la plateforme pour une accessibilité accrue et une interaction continue.
- **Amélioration de communication** : par exemple si un élève est marqué comme absent, un message SMS est envoyé automatiquement à ses parents.

Bibliographie

- [1] De Matteï, L., Janny, S., Nathan, S., & Shu-Quartier, W. (2022, avril). Introduction à l'apprentissage automatique. La Revue 3EI, (108), 1–12. <https://sti.eduscol.education.fr/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/14512/14512-introduction-lapprentissage-automatique-ensps.pdf>
- [2] IBM. (s.d.). Apprentissage supervisé. <https://www.ibm.com/fr-fr/topics/supervised-learning>
- [3] Béchet, N. (2012). État de l'art sur les systèmes de recommandation. Projet Axis de l'INRIA. <https://people.irisa.fr/Nicolas.Bechet/Publications/EtatArt.pdf>
- [4] OCDE. (2022). Perspectives de l'OCDE sur l'éducation numérique 2021 : Repousser les frontières avec l'IA, la blockchain et les robots. Éditions OCDE.
- [5] Johns Hopkins University. (2019, 7 janvier). Johns Hopkins University analysis draws link between Knewton Alta and improved student outcomes. Wiley. <https://www.wiley.com/en-us/education/alta/resources/blog---johns-hopkins-university-analysis-draws-link-between-knew>
- [6] Redress Compliance. (2023, 24 mars). What is Edmodo and how does it use AI? Redress Compliance. <https://redresscompliance.com/what-is-edmodo-and-how-does-it-use-ai/>
- [7] Harvard University Center for Education Policy Research. (s.d.). DreamBox Learning Achievement Growth. Consulté le 6 mai 2025, à partir de <https://cepr.harvard.edu/dreambox-learning-achievement-growthcepr.harvard.edu+1cepr.harvard.edu+1>
- [8] Mballo, A. (s.d.). Conception et développement d'une solution mobile de gestion de la 'petite monnaie' [Mémoire de fin d'études, Université Assane SECK de Ziguinchor].
- [9] Roques, P. (2006). UML 2 par la pratique : Études de cas et exercices corrigés (5e éd.). Éditions Eyrolles.
- [10] Benyoucef, L., & Bouamara, N. (2020). Conception et réalisation d'une application mobile multiplateforme pour la gestion des ventes [Mémoire de fin d'études, Université Mouloud Mammeri de Tizi-Ouzou]. Dspace UMMTO. <https://dspace.ummto.dz/items/408ebdcd-691d-4e17-b30c-a1a64e976646>
- [11] Soutou, C. (2012). UML 2 pour les bases de données : Modélisation, normalisation, génération SQL, outils : avec 30 exercices corrigés (2e éd.). Éditions Eyrolles.
- [12] Microsoft. (s.d.). Visual Studio Code documentation. Consulté le 27 avril 2025, à partir de <https://code.visualstudio.com/docs>

- [13] Apache Friends. (s.d.). XAMPP Installers and Downloads. Consulté le 27 avril 2025, à partir de <https://www.apachefriends.org/index.html>
- [14] Oracle. (s.d.). MySQL Community Edition. Consulté le 27 avril 2025, à partir de <https://www.mysql.com/products/community/>
- [15] Modeliosoft. (s.d.). Modelio – UML and BPMN free modeling tool. Consulté le 27 avril 2025, à partir de <https://www.modelio.org/>
- [16] Object Management Group. (s.d.). Business Process Model and Notation (BPMN). Consulté le 27 avril 2025, à partir de <https://www.bpmn.org/>
- [17] Templier, T., & Gougeon, A. (2007). JavaScript pour le Web 2.0 : Programmation objet, DOM, Ajax, Prototype, Dojo, Script.aculo.us, Rialto... Éditions Eyrolles.
- [18] Chavelli, M. (2016). Découvrez le framework PHP Laravel. Éditions Eyrolles.
- [19] Rubix ML. (s.d.). Documentation Rubix ML. Consulté le 27 avril 2025, à partir de <https://docs.rubixml.com/latest/>
- [20] Bagless, S., & Muralidharan, K. (2014). Mindspark screen showing a sample question from class 6 on topic Percentages [Figure 4]. In Improving education outcomes in developing countries: Evidence, knowledge gaps, and policy implications. ResearchGate. https://www.researchgate.net/figure/Mindspark-screen-showing-a-sample-question-from-class-6-on-topic-Percentages-The_fig4_265235341
- [21] Wan, T. (2018, August 21). Knewton's new business attracts new \$25M in funding, but some things don't change. EdSurge. <https://www.edsurge.com/news/2018-08-21-knewton-s-new-business-attracts-new-25m-in-funding-but-some-things-don-t-change>
- [22] **Kalyaniwala, C.** (2018). Analyse d'Edmodo : un site de réseautage éducatif. Alsic, 21, Article 3333. <https://doi.org/10.4000/alsic.3333>
- [23] DreamBox Learning. (2025, 10 mars). DreamBox Math Educator Dashboard: Printing Student Login Cards (Home Page). <https://dreamboxlearning.zendesk.com/hc/en-us/articles/27281427464083>