



RAPPORT DE PROJET DE FIN D'ÉTUDES

Pour L'Obtention du Diplôme

LICENCE SCIENCES ET TECHNIQUES

SPÉCIALITÉ : GÉNIE INFORMATIQUE

CONCEPTION ET REALISATION D'UNE PLATEFORME NUMERIQUE INTEGREE POUR LA GESTION GLOBALE DE L'ACADEMIE NOBOUGH

Réalisé par :

LAMRISSI Bahaa-eddine

Sous la direction de :

Pr. Loubna HAMAMI, professeur à la FST de Settat

Mr. Mehdi ELJAMAI, responsable du bloc informatique à l'académie Nobough

Soutenu le : 12 juin 2025

JURY

Pr. Loubna HAMAMI, professeur à la FST de Settat

Pr. Nassereddine BOUCHAIB, professeur à la FST de Settat

Pr. Imane LMATI, professeur à la FST de Settat

Année Universitaire : 2024-2025

Dédicace

Avant toute chose, je rends grâce à Dieu, source de toute lumière, de sagesse et de bienveillance. C'est en Lui que résident l'origine de tout savoir et la force qui m'a soutenu à chaque étape de ce parcours.

À mon cher père, Chaque pas dans ma vie porte l'empreinte de ton amour, de ta rigueur et de ta sagesse. Tu m'as transmis les valeurs fondamentales du travail, de la persévérance et de l'intégrité. Ton soutien indéfectible est le socle sur lequel j'ai bâti mes réussites. Merci d'être cet exemple inspirant.

À ma douce mère, Comme une étoile polaire, tu as guidé mes pas avec tendresse et fermeté. Ta lumière, ton amour inconditionnel et ta force silencieuse m'ont porté dans les moments d'incertitude. Tu as nourri mes rêves, apaisé mes peines et célébré chacun de mes accomplissements.

À Madame Loubna Hamami, mon encadrante, Je vous adresse ma profonde reconnaissance pour votre bienveillance, votre patience et votre professionnalisme. Votre accompagnement constant, vos conseils avisés et votre confiance ont été d'un immense soutien tout au long de ce projet. Merci pour votre présence précieuse.

Enfin, à moi-même, Merci pour ta résilience face aux tempêtes, pour avoir tenu bon quand tout semblait incertain. Tu as puisé au fond de toi la force d'avancer, la foi en tes rêves, et le courage de ne jamais abandonner. Tu mérites aussi ta part de gratitude.

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet. Leur soutien, leurs conseils et leur encouragement ont été essentiels pour mener à bien ce travail.

Je remercie tout particulièrement mon encadrante, Madame Loubna Hamami, pour son accompagnement précieux, sa patience et ses conseils avisés tout au long de ce projet. Son expertise et son professionnalisme ont été d'une grande aide pour surmonter les défis rencontrés.

Je remercie également tous les membres de l'Académie Nobough pour leur collaboration et leur soutien. Leur accueil chaleureux et leur disponibilité ont grandement facilité la réalisation de ce projet.

Un grand merci à ma famille et à mes amis pour leur soutien indéfectible, leurs encouragements et leur compréhension durant cette période intense de travail. Leur présence à mes côtés a été une source inépuisable de motivation.

Enfin, je remercie tous ceux qui, de près ou de loin, ont contribué à la réussite de ce projet. Leur aide et leur soutien ont été précieux et je leur en suis profondément reconnaissant.

Ce projet n'aurait pas été possible sans l'aide et le soutien de toutes ces personnes, et je leur en suis éternellement reconnaissant.

Résumé

Ce projet constitue l'aboutissement de mon stage de fin d'études, réalisé dans le cadre de l'obtention de la licence en Génie Informatique. Il porte sur la conception et le développement d'une plateforme web moderne dédiée à la gestion de l'académie Nobough, une structure proposant diverses formations académiques et techniques à un large public. L'objectif principal de ce projet est de mettre en place un système complet permettant à l'académie d'informatiser entièrement le processus de gestion de ses formations, de ses étudiants, de ses inscriptions et de ses ressources pédagogiques, tout en offrant une expérience utilisateur fluide et intuitive.

Afin d'atteindre cet objectif, j'ai dans un premier temps mené une étude approfondie afin d'analyser les besoins spécifiques de l'académie, d'identifier les utilisateurs cibles (administrateurs, formateurs, étudiants...), et de définir les fonctionnalités essentielles du système. J'ai ensuite conçu l'interface utilisateur en m'appuyant sur des outils de prototypage visuel tels que Canva, avant de modéliser le système à l'aide de diagrammes UML (cas d'utilisation, classes, séquences...).

Pour la mise en œuvre du projet, nous avons opté pour des technologies web modernes, en particulier le framework Next.js, qui combine performance, modularité et rendu côté serveur. Pour le design, nous avons utilisé Tailwind CSS afin de garantir une personnalisation rapide et un rendu visuel professionnel. Le site développé repose sur une architecture modulaire respectant les bonnes pratiques du développement web, notamment en matière de séparation des responsabilités et de maintenabilité du code.

Ce projet m'a permis non seulement de mettre en pratique les compétences acquises tout au long de ma formation, mais également de me familiariser avec des outils récents et puissants, au cœur du développement web professionnel actuel.

Abstract

This project represents the culmination of my final-year internship, carried out as part of obtaining my Bachelor's degree in Computer Engineering. It focuses on the design and development of a modern web platform dedicated to the management of Nobough Academy, an institution offering various academic and technical training programs to a wide audience.

The main objective of this project is to implement a comprehensive system that allows the academy to fully digitize the management process of its training programs, students, registrations, and educational resources, while providing a smooth and intuitive user experience.

To achieve this objective, I first conducted an in-depth study to analyze the specific needs of the academy, identify the target users (administrators, instructors, students...), and define the system's essential features. I then designed the user interface using visual prototyping tools such as Canva, before modeling the system with UML diagrams (use cases, classes, sequences...).

For the project implementation, we chose modern web technologies, particularly the Next.js framework, which combines performance, modularity, and server-side rendering. For the design, we used Tailwind CSS to ensure quick customization and a professional visual appearance. The developed site is based on a modular architecture that adheres to best practices in web development, especially regarding separation of concerns and code maintainability.

This project not only allowed me to apply the skills acquired throughout my studies but also enabled me to become familiar with recent and powerful tools that are at the heart of modern professional web development.

ملخص

Mulakhas

يمثل هذا المشروع تنويجاً لفترة تدريبي النهائي، الذي أجزته في إطار الحصول على شهادة الإجازة في هندسة المعلومات. يتمحور المشروع حول تصميم وتطوير منصة إلكترونية حديثة (موقع إلكتروني) مخصصة لإدارة أكاديمية نبوغ، وهي مؤسسة تقدم مجموعة متنوعة من الدورات التعليمية والتقنية لفئات واسعة من المتعلمين.

الهدف الرئيسي من هذا المشروع هو بناء نظام معلوماتي متكامل يسمح للأكاديمية برقمنة جميع عملياتها المتعلقة بإدارة البرامج الدراسية، والطلاب، وعمليات التسجيل، والموارد التعليمية، مع ضمان توفير تجربة استخدام سلسة وسهلة.

لتحقيق هذا الهدف، قمت في البداية بإجراء دراسة معمقة لتحليل الاحتياجات الخاصة بالأكاديمية، وتحديد الفئات المستهدفة من المستخدمين (مدربون، معلمون، طلاب...)، وتحديد الوظائف الأساسية التي يجب أن يقدمها النظام.

بعد ذلك، قمت بتصميم واجهة المستخدم باستخدام أدوات التصميم الرسومي عبر الإنترنت مثل منصة "كانفا" (وهي أداة إلكترونية تسمح بتصميم واجهات الاستخدام والمواد البصرية بشكل مرئي دون الحاجة إلى مهارات برمجية).

التي تشمل مخططات حالات UML ثم قمت بنمذجة النظام باستخدام مخططات النمذجة الموحدة الاستخدام (التحديد كيفية استخدام النظام من قبل مختلف المستخدمين)، ومخططات الكائنات والفئات (تمثيل مكونات النظام البرمجية)، ومخططات تسلسل العمليات (تمثيل تسلسل خطوات التنفيذ داخل النظام).

في مرحلة تنفيذ المشروع البرمجي، تم اختيار استخدام إطار العمل "نكست جي إس" (وهو إطار عمل برمجي متقدم مبني على لغة جافاسكريبت، يُستخدم لتطوير موقع إلكترونية عالية الأداء، ويتميز بامكانية تنفيذ جزء كبير من المعالجة على الخادم لتحسين سرعة الموقع).

أما بالنسبة للتصميم الجمالي للواجهة، فقد تم اعتماد نظام "تايلوند سي إس إس" (وهو إطار عمل خاص بتصميم واجهات الواقع الإلكتروني، يعتمد على استخدام أصناف جاهزة لتنسيق الصفحات، مما يسهل عملية بناء تصاميم عصرية ومهنية بشكل سريع).

الموقع الإلكتروني الذي تم تطويره يعتمد على بنية معمارية معيارية (أي أن مكونات النظام مقسمة إلى وحدات برمجية منفصلة يسهل تطويرها وصيانتها بشكل مستقل)، مع الالتزام بأفضل الممارسات في تطوير البرمجيات، لا سيما فيما يخص مبدأ فصل المسؤوليات بين مكونات النظام المختلفة، لضمان جودة الشيفرة البرمجية وسهولة صيانتها.

لقد أتاح لي هذا المشروع فرصة تطبيق مختلف المهارات التي اكتسبتها خلال مسيرتي الدراسية، بالإضافة إلى تعميق معرفتي بالأدوات والتقنيات الحديثة التي تشكل حالياً عماد تطوير الواقع الإلكتروني المحترفة.

Table des matières

Dédicace	2
Remerciements	3
Résumé	4
Abstract	5
Mulakhas	6
Liste des figures	12
Liste des tableaux	13
Liste des abréviations	14
Introduction générale	15
1 Contexte général du projet	18
I Introduction	18
II Présentation de l'organisme d'accueil	18
1 L'Académie Nobough	18
2 Mission de l'académie	19
3 Historique de l'académie	19
III Cadre général du projet	19
1 Analyse de l'existant	19
2 Problématique	20
3 Objectif du projet	20
IV Planification et conduite du projet	21
1 Planning et méthodologie de travail	21
2 Diagramme de GANTT	21
V Conclusion	22
2 Analyse et Conception	24
I Introduction	24
II Besoins fonctionnels et non fonctionnels	24
1 Besoins fonctionnels pour l'Administrateur	24
2 Besoins fonctionnels pour le Comptable	25
3 Besoins fonctionnels pour la Secrétaire	26
4 Besoins fonctionnels pour le Professeur	27

5	Besoins fonctionnels pour l'Étudiant	27
6	Besoins fonctionnels pour l'Utilisateur	27
7	Besoins non fonctionnels	28
III	Conception adoptée	28
1	Langage UML	28
2	Outil de modélisation UML	29
3	Identification des acteurs	29
4	Exigences du Système	30
5	Diagramme des cas d'utilisation	32
6	Diagrammes de séquences	36
7	Diagrammes d'Etat-transition	41
8	Diagramme des Classes	43
9	Diagramme des Packages	45
IV	Structure de la base de données	46
1	Modèle physique de données	46
V	Conclusion	47
3	Etude technique du projet	49
I	Introduction	49
II	Architecture du projet	49
1	Modular Folder-Based Architecture (MFBA)	49
2	Dossiers principales	50
3	API REST	50
4	Middlewares	50
III	Outils utilisés	51
1	Introduction	51
2	Environnement logiciel	51
3	Environnement logiciel	52
4	Langages	53
5	Frameworks	54
6	Bibliothèques	55
7	Outils AI	57
IV	Intégration de l'intelligence artificielle	58
1	Intégration d'un moteur IA pour la gestion des dettes	58
2	Utilisation de l'API Gemini pour les Recommandations Financières	60
V	Communication en temps réel avec WebRTC	61
1	Principe de WebRTC	61
2	Rôle de Socket.io comme serveur de signalisation	61
3	Avantages de cette architecture	61
VI	Automatisation et optimisation de la gestion des paiements mensuels	62
1	Problématique	62
2	Solution proposée	62
3	Automatisation via Cron Job	63
4	Optimisation par indexation	64
5	Avantages de la solution	64
VII	Conclusion	64

4 Réalisation de l'application	66
I Introduction	66
II Logo de notre application	66
III Présentation des interfaces de l'application	67
1 Page d'accueil	67
2 Première visite du site	68
3 Processus de paiement et de validation	71
4 Gestion de la session et récupération de mot de passe	75
5 Rejoindre une formation après login	77
6 Validation ou refus par la secrétaire	78
7 Consultation du paiement	78
8 Gestion des paiements	79
9 Annonces	80
10 Protestations	81
11 Consultation du programme d'une formation	82
12 Séance en ligne	85
13 Génération des documents	87
14 Pages de profil pour l'étudiant et le professeur	89
15 Partie Comptable - Admin	90
IV Conclusion	95
Conclusion générale et perspectives	96
Bibliographie / Webographie	97

Table des figures

1.1	Bâtiment de l'académie	18
1.2	Méthodologie de travail	21
1.3	Diagramme de Gannt	22
2.1	Visual paradigm	29
2.2	Diagramme des acteurs	30
2.3	Diagramme de cas d'utilisation de l'utilisateur	32
2.4	Diagramme de cas d'utilisation de l'étudiant	33
2.5	Diagramme de cas d'utilisation du professeur	33
2.6	Diagramme de cas d'utilisation du comptable	34
2.7	Diagramme de cas d'utilisation de la secrétaire	35
2.8	Diagramme de cas d'utilisation de l'admin	35
2.9	Diagramme de séquence "Inscription au site"	36
2.10	Diagramme de séquence "Inscription à une formation"	37
2.11	Diagramme de séquence "Paiement mensuel"	38
2.12	Diagramme de séquence "Déposer une protestation"	38
2.13	Diagramme de séquence "Rejoindre une séance en ligne"	39
2.14	Diagramme de séquence "Récupérer mot de passe"	40
2.15	Diagramme d'état-transition "Inscription au site"	41
2.16	Diagramme d'état-transition "Protestation"	42
2.17	Diagramme d'état-transition "Room"	42
2.18	Diagramme de classes	45
2.19	Diagramme des packages	46
2.20	Modèle physique de données	47
3.1	L'architecture MFBA	49
3.2	VS Code	51
3.3	VS Code	52
3.4	XAMPP	52
3.5	Node.js	52
3.6	SQL	53
3.7	JavaScript	53
3.8	CSS	53
3.9	Next.js	54
3.10	React	54
3.11	Express	54
3.12	Tailwind CSS	55
3.13	GSAP	55
3.14	Chartjs	55

3.15	JWT	56
3.16	SendGrid	56
3.17	Socket.IO	56
3.18	Nodemailer	57
3.19	Puppeteer	57
3.20	TensorFlow	57
3.21	Fonctionnement d'un réseau de neurones	58
3.22	Algorithme des paiements mensuels	62
3.23	Cron-job commande	63
4.1	Logo de l'académie	66
4.2	Page d'accueil	67
4.3	Accès restreint	68
4.4	Rejoindre une formation	68
4.5	Login : compte n'existe pas	69
4.6	Registration	69
4.7	Registration	70
4.8	Secrétaire refuse	70
4.9	Email de refus	71
4.10	Secrétaire valide	71
4.11	Lien de paiement	72
4.12	Secrétaire valide	72
4.13	Registration	73
4.14	Lien de paiement	73
4.15	Lien de paiement	74
4.16	Login réussi	74
4.17	Logout	75
4.18	Mot de passe oublié	75
4.19	Mot de passe oublié	76
4.20	Mot de passe oublié	76
4.21	Mise à jour du mot de passe	77
4.22	Processus de demande pour rejoindre une formation	77
4.23	Processus de demande pour rejoindre une formation	78
4.24	Consultation de paiement par l'étudiant	78
4.25	Gestion des paiements	79
4.26	Gestion des paiements en détails	79
4.27	Ajouter annonce	80
4.28	Consulter les annonces	80
4.29	Déposer une protestation	81
4.30	Répondre à une protestation	81
4.31	les FAQs	82
4.32	Programme d'une formation	83
4.33	Chat en temps réel	83
4.34	Bibliothèque numérique	84
4.35	Autres participants à la formation	84
4.36	Générer un certificat pour un étudiant	85
4.37	Informations du professeur	85
4.38	Création d'une séance en ligne	86

4.39 rejoindre la séance en ligne	86
4.40 Déroulement de la séance en ligne	87
4.41 Documents pour l'étudiant	88
4.42 Attestation d'inscription	88
4.43 Emploi du temps 1	88
4.44 Reçu de paiement	88
4.45 Certificat	88
4.46 Documents pour le professeur	89
4.47 Attestation de travail	89
4.48 Emploi du temps 2	89
4.49 Pages de profil pour l'étudiant et le professeur	90
4.50 Tableau de bord du le comptable	91
4.51 Consultation des revenus, dépenses et dettes	92
4.52 Ajout d'un revenu	92
4.53 Rembourser les dettes intelligeament	93
4.54 Tableau de bord de l'admin	94
4.55 Création de comptes secrétaire et comptable	94

Liste des tableaux

2.1 Exigences du système	30
2.2 Description des classes	43

Liste des abréviations

API : Application Programming Interface

ARIA : Accessible Rich Internet Applications

BDD : Base de Données

CSS : Cascading Style Sheets

CRUD : Create, Read, Update, Delete

DBMS : Database Management System

GSAP : GreenSock Animation Platform

HTTP : HyperText Transfer Protocol

ID : Identifier

IP : Internet Protocol

JS : JavaScript

JSON : JavaScript Object Notation

JWT : JSON Web Token

MFBA : Modular Folder-Based Architechture

NPM : Node Package Manager

REST : Representational State Transfer

SMTP : Simple Mail Transfer Protocol

URI : Uniform Resource Identifier

URL : Uniform Resource Locator

UI : User Interface

UX : User Experience

Introduction générale

Dans un monde où la transformation numérique ne cesse de s'accélérer, la digitalisation des structures éducatives est devenue une nécessité pour répondre aux attentes des apprenants modernes et pour améliorer l'efficacité des processus administratifs. Les académies de formation, en particulier, doivent s'adapter à cette évolution afin de rester accessibles, performantes et compétitives dans un environnement en constante mutation. Dans ce contexte, et dans le cadre de ma formation en licence Génie Informatique à la Faculté des Sciences et Techniques de Settat (FSTS), j'ai effectué un stage de fin d'études au sein de l'académie Nobough, durant la période allant du 10/04/2025 au 10/06/2025. Ce stage m'a permis de mettre en pratique les acquis théoriques de ma formation, de me confronter aux réalités du monde professionnel, et de faciliter ainsi ma transition vers la vie active.

C'est dans ce cadre que j'ai conçu et développé une plateforme web appelée "Nobough" dédiée pour la gestion de l'académie, visant à centraliser la gestion des étudiants, des inscriptions, des formateurs, des ressources pédagogiques et des formations proposées.

La plateforme s'articule autour de plusieurs axes essentiels de la gestion éducative moderne : l'organisation des modules et des sessions de formation, la gestion des comptes utilisateurs (étudiants, secrétaires, formateurs...), le suivi des inscriptions, la communication interne via un système de messagerie, ainsi que la diffusion de contenus pédagogiques. Des fonctionnalités avancées telles que l'envoi de notifications personnalisées, un tableau de bord statistique, ainsi que la gestion des certificats de fin de formation, ont également été intégrées pour enrichir l'expérience utilisateur.

La mise en place d'un tel système présente de nombreux avantages pour l'académie. Elle permet l'automatisation des processus administratifs, la réduction des erreurs humaines, un accès simplifié à l'information, une amélioration de la communication interne et externe, ainsi qu'un gain de temps considérable pour les responsables de la gestion. Les étudiants, quant à eux, bénéficient d'une interface moderne, claire et accessible, leur permettant de suivre leur parcours de formation de manière autonome et organisée.

Afin de présenter ce travail de la manière la plus exhaustive qui soit, le présent mémoire est scindé en quatre parties :

- Contexte général du projet.
- Analyse et conception.
- Etude technique du projet.
- Réalisation de l'application. A la fin de ce manuscrit une conclusion sera présentée ainsi que des perspectives.

Ce mémoire présente les différentes étapes de réalisation du projet : analyse des besoins de l'académie Nobough, conception du système, choix techniques, et mise en œuvre des principales fonctionnalités. À travers ce projet, nous visons à contribuer à la modernisation de la gestion académique et à faciliter l'accès à la formation dans un contexte numérique en pleine évolution.

Chapitre 1 : Contexte général du projet

Chapitre 1

Contexte général du projet

I Introduction

L'analyse du projet constitue une étape stratégique cruciale pour assurer sa réussite et en faciliter la gestion à chaque phase. Ce premier chapitre est donc consacré à cette étude, incluant la présentation générale du projet, de l'organisme d'accueil, l'académie Nobough, ainsi que la méthodologie adoptée pour le développement. Les sections suivantes exposeront les objectifs visés et les étapes essentielles ayant guidé la mise en œuvre du projet.

II Présentation de l'organisme d'accueil

1 L'Académie Nobough

L'académie Nobough est une structure de formation privée qui propose des programmes académiques et techniques dans divers domaines, tels que l'informatique, la religion, la gestion et les langues. Elle met l'accent sur la qualité, la pratique et l'innovation pédagogique, tout en offrant un accompagnement personnalisé. Grâce à une équipe de formateurs expérimentés et à un environnement moderne, l'académie prépare des profils compétents et autonomes. Elle s'engage également dans la transformation numérique pour améliorer l'expérience de formation et la gestion administrative.



FIGURE 1.1 – Bâtiment de l'académie

2 Mission de l'académie

L'Académie Nobough propose un large éventail de formations de haute qualité, conçues pour garantir à ses étudiants une expérience d'apprentissage exceptionnelle. L'excellence pédagogique est au cœur de sa mission : seuls les enseignants les plus compétents et les plus expérimentés sont sélectionnés afin d'assurer un encadrement optimal et un accompagnement personnalisé.

Le programme de l'académie couvre de nombreux domaines variés, permettant aux apprenants de développer des compétences à la fois intellectuelles, créatives et physiques. Parmi les filières proposées figurent :

- La religion, pour approfondir les connaissances spirituelles et éthiques.
- La programmation et les technologies numériques, pour maîtriser les outils du monde moderne.
- Les arts, pour encourager l'expression artistique et la créativité sous toutes ses formes.
- Les langues, pour favoriser l'ouverture culturelle et renforcer la communication internationale.
- Les sports, afin de promouvoir la santé, le bien-être et l'esprit d'équipe.

En complément de ses cursus, l'Académie Nobough offre un environnement d'apprentissage moderne et inclusif, propice à l'épanouissement personnel et professionnel des étudiants, les préparant aux défis du monde actuel.

3 Historique de l'académie

L'Académie Nobough a vu le jour le 4 février 2016. À ses débuts, elle était une modeste école dédiée à l'enseignement du Coran, accueillant un petit groupe d'élèves passionnés. Fondée avec l'ambition de transmettre les valeurs spirituelles et morales, l'académie s'est rapidement forgée une réputation de sérieux et de qualité.

Au fil des années, face à l'enthousiasme croissant des étudiants et à la demande grandissante des familles, l'académie a progressivement élargi son offre éducative. De nouvelles disciplines ont été intégrées, telles que la programmation informatique, les langues étrangères, les arts, et les activités sportives, tout en conservant une solide base d'enseignement religieux.

Aujourd'hui, l'Académie Nobough est devenue un véritable pôle d'excellence pluridisciplinaire, accueillant des apprenants de tous horizons. Elle continue de s'appuyer sur ses valeurs fondatrices — qualité, engagement, respect et innovation — pour offrir une éducation moderne et équilibrée, adaptée aux besoins du monde actuel.

III Cadre général du projet

1 Analyse de l'existant

Après une analyse complète, il apparaît que l'académie ne dispose pas actuellement d'un système informatisé de gestion. La gestion actuelle repose uniquement sur l'enregistrement manuel des transactions sur papier ou dans des feuilles de calcul, ce qui ralentit les opérations et augmente le risque d'erreurs.

Cette méthode empêche le suivi en temps réel des données (inscriptions, paiements...), ce qui peut entraîner des retards, des incohérences et des pertes d'informations.

2 Problématique

Ces dernières années, plusieurs institutions éducatives au Maroc ont entrepris une transition vers la digitalisation pour moderniser leurs processus administratifs et pédagogiques. Cependant, certaines académies, notamment l'académie **Nobough** située à **Berrechid**, continuent de gérer leurs opérations de manière traditionnelle à l'aide de documents papier.

Cette méthode manuelle génère de nombreux problèmes : désorganisation dans la gestion des paiements, mauvaise visibilité sur les budgets, retards dans le suivi des dettes et dépenses, et perte de temps considérable dans les tâches administratives. Ce manque de modernisation freine considérablement le bon fonctionnement de l'établissement et nuit à son image professionnelle.

3 Objectif du projet

Dans ce contexte, l'objectif principal de ce projet est de concevoir et de développer une **application web moderne et professionnelle**, pour l'académie **Nobough**, permettant **l'informatisation complète des processus administratifs, financiers et pédagogiques** de l'établissement.

Cette application garantira de nombreux avantages, notamment :

— **Pour la secrétaire :**

- Gestion centralisée des inscriptions et protestations.
- Suivi des paiements et génération de rapports financiers.
- Gestion des profils étudiants et professeurs.
- Gestion des annonces et des formations.

— **Pour le comptable :**

- Suivi précis des dépenses, revenus, et dettes.
- Gestion du budget et prévisions financières.
- Génération de rapports financiers détaillés.

— **Pour l'admin :**

- Accès complet à toutes les fonctionnalités de l'application.
- Gestion des secrétaires et des comptables.
- Surveillance de la sécurité et maintenance du système.

— **Pour les étudiants :**

- Consultation des annonces et des formations disponibles.
- Paiement en ligne pour les inscriptions et les cours.
- Accès à un tableau de bord personnalisé pour suivre les progrès.

— **Pour les utilisateurs :**

- Interface intuitive pour naviguer et accéder aux informations.
- Support client intégré pour résoudre les problèmes rapidement.

— **Pour les professeurs :**

- Gestion des classes et des étudiants inscrits.
- Publication des documents et des ressources pédagogiques.
- Suivi des performances des étudiants et génération de rapports.

IV Planification et conduite du projet

1 Planning et méthodologie de travail

La planification de l'application Nobough a suivi une approche méthodique et itérative, garantissant un déroulement structuré tout au long du projet.

On a commencé par analyser les besoins spécifiques de l'académie, en étudiant le fonctionnement interne et les attentes des utilisateurs pour mieux comprendre les enjeux. Cette phase m'a permis de déterminer les principales fonctionnalités et priorités de l'application. Ensuite, on a défini les objectifs du projet, en mettant l'accent sur la conception d'une interface intuitive et d'une architecture évolutive qui permettrait une gestion efficace des étudiants, des formateurs et des ressources pédagogiques.

Le développement de l'application a été réalisé selon une méthode agile, avec des itérations régulières. Cela a permis d'intégrer progressivement les fonctionnalités principales, tout en tenant compte des retours des utilisateurs et des besoins émergents.

Enfin, la rédaction du rapport a été effectuée parallèlement à l'avancement du projet, ce qui m'a permis de documenter chaque étape, les décisions prises, les défis rencontrés et les solutions apportées, assurant ainsi une vision claire et complète du projet de développement de l'application pour l'académie Nobough.

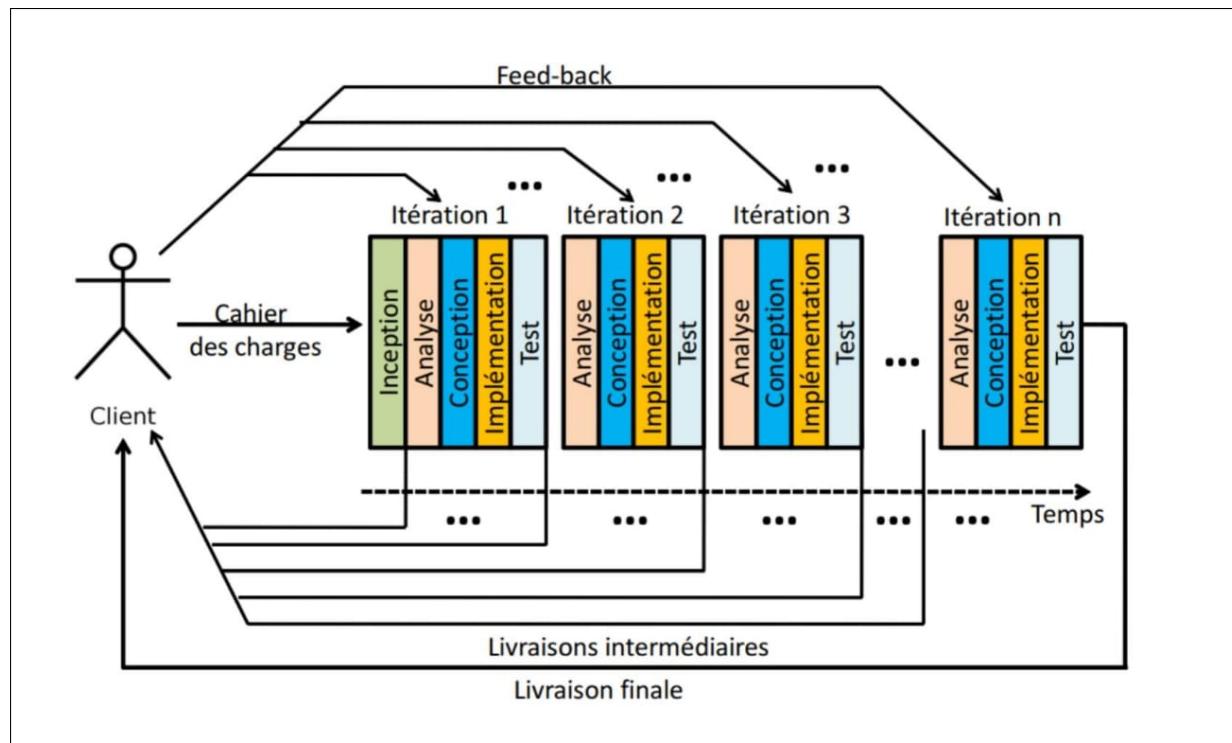


FIGURE 1.2 – Méthodologie de travail

2 Diagramme de GANTT

Le diagramme de Gantt est un outil de gestion de projet essentiel pour visualiser l'évolution des tâches dans le temps. Il représente de manière graphique les différentes étapes d'un projet, en indiquant leur durée et leur interdépendance.

Ce diagramme est une représentation d'un graphe connexe, valué et orienté, qui permet

de suivre l'avancement des différentes tâches. Le diagramme de Gantt ci-dessous illustre la gestion et le déroulement de notre projet, en montrant clairement les étapes clés et leur planification dans le temps.

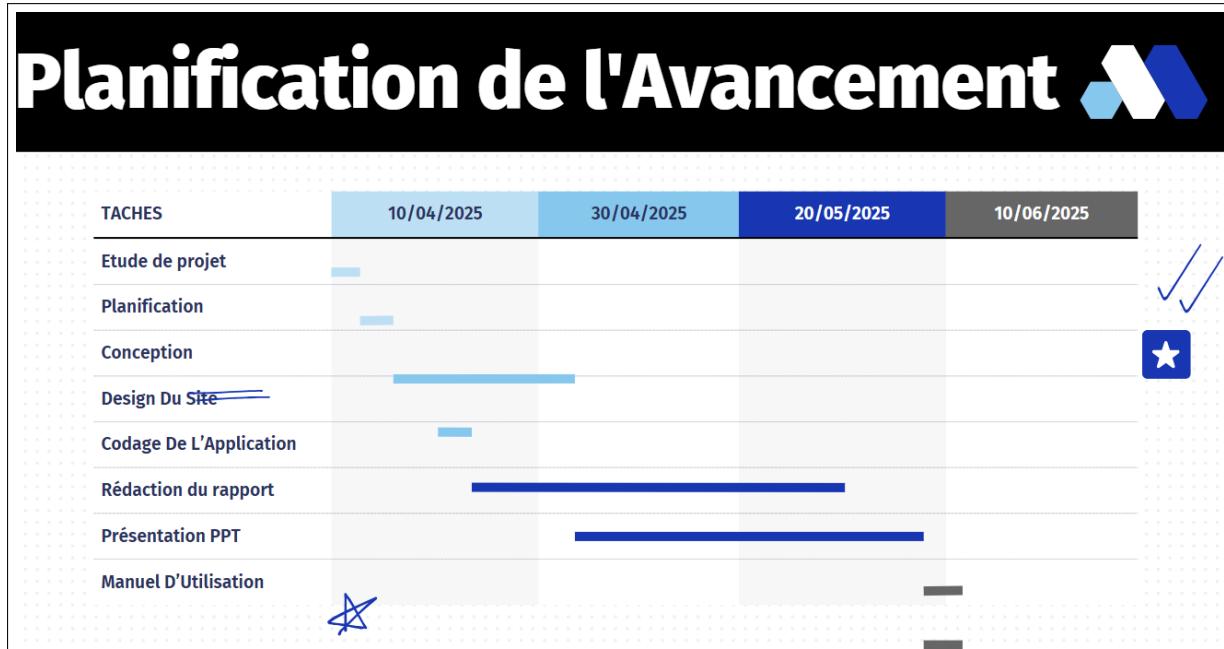


FIGURE 1.3 – Diagramme de Gannt

V Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil ainsi qu'une vue d'ensemble de notre projet de fin d'études.

Le chapitre suivant sera consacré à la conception de notre projet, où nous détaillerons les diagrammes de classes, les cas d'utilisation, les diagrammes de séquence, les diagrammes d'état-transition, ainsi que le modèle physique des données.

Chapitre 2 : Analyse et Conception

Chapitre 2

Analyse et Conception

I Introduction

Dans ce chapitre, nous commencerons par analyser les besoins fonctionnels et non fonctionnels de notre application. Ensuite, nous réaliserons une étude conceptuelle de la solution proposée en présentant plusieurs diagrammes UML, tels que le diagramme de cas d'utilisation, le diagramme de classes, les diagrammes de séquences et les diagrammes d'état-transition, afin de clarifier et formaliser les objectifs du système. Enfin, nous présenterons la structure de la base de données.

II Besoins fonctionnels et non fonctionnels

Les besoins fonctionnels décrivent ce que le système doit faire pour satisfaire les utilisateurs, tandis que les besoins non fonctionnels définissent les critères de qualité comme la sécurité et la performance. Les deux sont essentiels pour un système fiable et efficace.

1 Besoins fonctionnels pour l'Administrateur

Les besoins fonctionnels pour l'Administrateur illustrent les différentes actions et priviléges qu'il a dans notre application.

- **Gérer les comptables et les secrétaires :**
 - Créer des comptes, avec la possibilité de les supprimer ou modifier.
- **Consulter les dépenses et les revenus :**
 - Types pour les dépenses : salaires (prof, caméramans, secrétaires), fournitures, dettes.
 - Types pour les revenus : inscriptions, paiements mensuels, dons.
 - Filtres : dernier mois, anciens mois.
- **Consulter les dettes :**
 - Informations : montant, titre, date, deadline.
 - Filtres : date, deadline, montant.
 - Ajouter une nouvelle dette (montant, titre, remarque, deadline).
- **Consulter le budget :**
 - Budget = revenus - dépenses.

- Alerte si risque de budget réel négatif.
- Alerte si possibilité de rembourser des dettes avec suggestions.
- Budget réel = budget - salaires non payés.
- **Ajouter une dépense :**
 - Champs : type, montant, titre, remarque.
- **Ajouter un événement :**
 - Champs : titre, lieu, date, description, remarques, dépenses.
 - Alerte si le budget réel est en risque.
 - Vérification de faisabilité après création.
- **Consulter les protestations des étudiants.**
- **Gérer les demandes d'inscription :**
 - Vérifier la première étape : documents (CIN, 2 photos) et conditions d'admission
 - Vérifier la deuxième étape : frais d'inscription (200 DH), paiement du premier mois.
 - Valider (génération fiche + reçue) ou annuler.
- **Consulter la liste des étudiants.**
 - Modifier les informations d'un étudiant.
 - Cocher le paiement manuellement (générer reçue).
 - Incrire un étudiant manuellement.
- **Ajouter une nouvelle formations, 3 types possibles.**
 - Formation à vie (pas de durée spécifiée, pas de certificats).
 - Formation à délai (durée spécifiée, obtenir un certificat à la fin).
 - Formation graduelle (chaque niveau a son certificat de grade).
- **Suivi des formations.**
 - Génération de certificats pour les étudiants ayant finis la période de la formation.
- **Gérer les professeurs :** ajouter, modifier, supprimer.
- **Gérer les annonces :** ajouter, modifier, supprimer.
- **Consulter les paiements en ligne.**

2 Besoins fonctionnels pour le Comptable

Les besoins fonctionnels correspondent aux services et fonctionnalités que le système doit fournir pour répondre aux attentes des utilisateurs.

- **Consulter les dépenses et les revenus :**
 - Types pour les dépenses : salaires (prof, caméramans, secrétaires), fournitures, dettes.
 - Types pour les revenus : inscriptions, paiements mensuels, dons.
 - Filtres : dernier mois, anciens mois.
- **Consulter les dettes :**
 - Informations : montant, titre, date, deadline.

- Filtres : date, deadline, montant.
- Ajouter une nouvelle dette (montant, titre, remarque, deadline).
- **Consulter le budget :**
 - Budget = revenus - dépenses.
 - Alerte si risque de budget réel négatif.
 - Alerte si possibilité de rembourser des dettes avec suggestions.
 - Budget réel = budget - salaires non payés.
- **Ajouter une dépense :**
 - Champs : type, montant, titre, remarque.
- **Ajouter un événement :**
 - Champs : titre, lieu, date, description, remarques, dépenses.
 - Alerte si le budget réel est en risque.
 - Vérification de faisabilité après création.
- **Consulter les protestations des étudiants.**

3 Besoins fonctionnels pour la Secrétaire

Les besoins fonctionnels de la Secrétaire détaillent les actions et priviléges dont elle dispose au sein de notre application.

- **Gérer les demandes d'inscription :**
 - Vérifier la première étape : documents (CIN, 2 photos) et conditions d'admission
 - Vérifier la deuxième étape : frais d'inscription (200 DH), paiement du premier mois.
 - Valider (génération fiche + reçue) ou annuler.
- **Consulter la liste des étudiants.**
 - Modifier les informations d'un étudiant.
 - Cocher le paiement manuellement (générer reçue).
 - Incrire un étudiant manuellement.
- **Ajouter une nouvelle formations, 3 types possibles.**
 - Formation à vie (pas de durée spécifiée, pas de certificats).
 - Formation à délai (durée spécifiée, obtenir un certificat à la fin).
 - Formation graduelle (chaque niveau a son certificat de grade).
- **Suivi des formations.**
 - Génération de certificats pour les étudiants ayant finis la période de la formation.
- **Gérer les professeurs :** ajouter, modifier, supprimer.
- **Gérer les annonces :** ajouter, modifier, supprimer.
- **Consulter les paiements en ligne.**

4 Besoins fonctionnels pour le Professeur

Les besoins fonctionnels du professeur définissent les tâches et droits spécifiques qu'il peut exercer dans notre application.

- **Consulter ses formations** et les étudiants inscrits.
- **Attribuer des notes aux étudiants.**
- **Ajouter des documents** dans la bibliothèque numérique.
- **Modifier son profil personnel.**
- **Recevoir des notifications :**
 - Nouvel étudiant inscrit.
 - Étudiant ayant quitté une formation.
- **Créer une séance en ligne.**

5 Besoins fonctionnels pour l'Étudiant

Les besoins fonctionnels de l'étudiant précisent les opérations qu'il peut effectuer et les accès dont il bénéficie dans notre application.

- **Consulter les formations** et demander une inscription, si il est accepté il va recevoir une notification avec lien de paiement du premier mois.
- **Consulter les annonces et les événements.**
- **Déposer une protestation :**
 - Suggestions automatiques.
 - Réponse automatique via un bot si suggestion choisie.
- **Consulter la bibliothèque numérique** contenant des documents ajoutés par le prof.
- **Espace pour le paiement en ligne.**
- **Consulter ses notifications :**
 - Rappel de mois non payés.
 - Paiement validé.
 - Rappel de séances...
- **Rejoindre une séance en ligne.**
- **Modifier son profil personnel.**

6 Besoins fonctionnels pour l'Utilisateur

Les besoins fonctionnels de l'utilisateur définissent les actions qu'il peut accomplir et les droits auxquels il a accès au sein de notre application.

- **Faire inscription sur le site** si il est accepté il va recevoir une notification avec lien de paiement des frais d'inscription.
- **Consulter les formations** mais sans le droit de s'inscrire.
- **Consulter les annonces et les événements.**
- **Consulter son tableau de bord pour une vision globale sur l'académie.**
- **Déposer une protestation :**
 - Suggestions automatiques.
 - Réponse automatique via un bot si suggestion choisie.

7 Besoins non fonctionnels

Après avoir identifié les différents besoins fonctionnels, qui sont les services attendus par les utilisateurs de l'application, nous allons maintenant explorer les besoins non fonctionnels qui définissent les contraintes et les critères de qualité du système à développer.

7.1 Interface Utilisateur

- 2 langues disponibles, avec sens de lecture RTL pour l'arabe.
- Thème : blanc, pistache, rosé-rouge (Identique au logo).
- Design moderne et esthétique.

7.2 Performance

- Chargement rapide.
- Optimisation via SSR et SSG.
- Mise en cache des données non sensibles.

7.3 Sécurité

- Authentification par rôle.
- Paiements en ligne sécurisés.
- Protection contre les failles (XSS, CSRF, etc.).
- Permissions d'accès selon rôle.

7.4 Accessibilité

- Responsive design (mobile, tablette, desktop).
- Compatible avec les navigateurs récents.

7.5 Maintenance et Évolutivité

- Architecture modulaire pour faciliter les évolutions.
- Suivi des bonnes pratiques de développement.
- Documentation technique pour administrateurs.

III Conception adoptée

1 Langage UML

L'UML est un langage visuel utilisé pour modéliser des systèmes, surtout dans le développement logiciel. Il sert à représenter la structure et le comportement d'un système complexe à travers différents types de diagrammes. Bien qu'il ne soit pas un langage de programmation, certains outils permettent de générer du code à partir de ses schémas. Il est étroitement lié à l'analyse et à la conception orientées objet, et peut aussi s'appliquer à d'autres domaines comme les processus industriels.

2 Outil de modélisation UML



FIGURE 2.1
– Visual paradigm

Visual Paradigm est un outil de modélisation visuelle complet qui permet de créer différents types de diagrammes comme les diagrammes UML, ERD (diagrammes de base de données), BPMN (processus métiers), ou encore les cartes de parcours utilisateur. Il est conçu pour accompagner toutes les étapes du cycle de vie logiciel, de l'analyse des besoins jusqu'à la génération de code et de documentation. Son interface intuitive, ses fonctionnalités avancées et ses options de travail collaboratif en font un choix solide pour les projets académiques ou professionnels.

Nous avons choisi Visual Paradigm pour sa richesse fonctionnelle, sa compatibilité avec les principaux standards de modélisation, et sa capacité à gérer des projets complexes. Contrairement à des outils comme Draw.io, qui est simple mais limité aux schémas basiques, ou Lucidchart, qui est plus orienté vers le business sans support technique poussé, Visual Paradigm offre des fonctionnalités avancées comme la génération de code, la traçabilité des exigences ou encore la gestion de projet. Par rapport à StarUML, qui est plus léger mais moins intégré, Visual Paradigm propose une solution plus complète et professionnelle, ce qui le rend idéal pour une modélisation rigoureuse, structurée et collaborative.

3 Identification des acteurs

Un acteur, en modélisation de systèmes, désigne toute personne ou entité externe qui interagit avec le système pour réaliser une action. Il peut s'agir d'un utilisateur, d'un autre système ou d'un service extérieur, et il est représenté séparément dans les diagrammes pour illustrer ces échanges.

L'analyse du système a permis de repérer les différents acteurs impliqués dans son fonctionnement.

- **Admin** : L'Admin a tous les priviléges de la secrétaire et du comptable, ainsi que la capacité de créer les comptes des secrétaires et comptables.
- **Comptable** : Il s'occupe de toute la gestion financière de l'académie : il suit les dépenses, les revenus, les dettes et le budget. Il reçoit aussi des alertes si le budget est en danger et peut proposer des événements en tenant compte des moyens disponibles.
- **Secrétaire** : Elle gère tout le côté administratif : inscriptions, dossiers des étudiants, formations, certificats, et même les annonces. Elle s'occupe aussi du suivi des paiements et de la gestion des professeurs.
- **Utilisateur** : L'utilisateur peut s'inscrire, consulter les formations, voir les annonces et déposer des protestations s'il a des remarques ou des soucis. Cependant, il ne peut pas demander une inscription à une formation tant qu'il n'a pas payé les frais d'inscription et que son paiement n'a pas été validé par la secrétaire, si oui il devient un étudiant.
- **Étudiant** : L'étudiant peut consulter les formations, payer en ligne, voir les annonces, participer aux séances en ligne, accéder à une bibliothèque de documents, et déposer des protestations s'il a des remarques ou soucis.
- **Professeur** : Le prof suit ses classes, note les étudiants, ajoute du contenu pédagogique, et peut organiser des séances en ligne. Il est aussi informé des inscriptions ou départs d'étudiants dans ses cours.

- **Système de paiements** : Le système suit les paiements, se charge de l'envoi automatique des emails de validation, annulation ou de rappel de mois non payés.
- **Bot de protestations** : Lorsqu'un utilisateur souhaite déposer une protestation, un bot peut lui proposer des suggestions. Si l'une d'elles est sélectionnée, le bot y répond automatiquement.
- **Bot de l'intelligence artificielle** : Aide le comptable à prédicter des dettes à rembourser pour gagner le temps de calcul.

Le diagramme ci-dessous présente les acteurs de système :

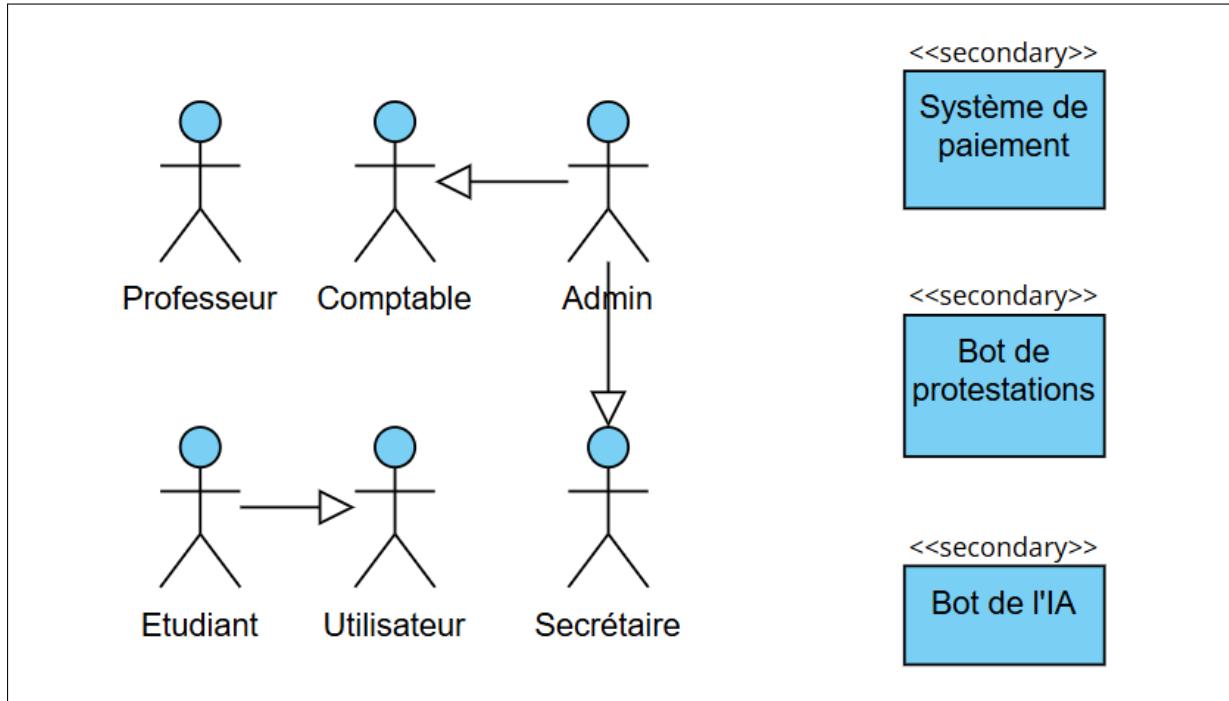


FIGURE 2.2 – Diagramme des acteurs

4 Exigences du Système

Ce tableau résume les exigences pour le développement de l'application de gestion de l'Académie Nobough, couvrant les fonctionnalités, la sécurité, la performance et l'ergonomie. Il prend en compte les besoins des différents utilisateurs (étudiants, professeurs, administrateurs...) et sert de référence pour garantir que toutes les attentes sont respectées tout au long du projet.

TABLEAU 2.1 : Exigences du système

Num	Exigence	Acteur
Ex1	Créer compte secrétaire	Admin
Ex2	Créer compte comptable	Admin
Ex3	Consulter protestations	Comptable, Admin
Ex4	Consulter dépenses	Comptable, Admin
Ex5	Ajouter dépense	Comptable, Admin
Ex6	Modifier dépense	Comptable, Admin
Ex7	Supprimer dépense	Comptable, Admin
Ex8	Consulter revenus	Comptable, Admin

Num	Exigence	Acteur
Ex9	Ajouter don	Comptable, Admin
Ex10	Modifier revenu	Comptable, Admin
Ex11	Supprimer revenu	Comptable, Admin
Ex12	Gérer dettes	Comptable, Admin
Ex13	Consulter budget	Comptable, Admin
Ex14	Ajouter événement	Comptable, Admin
Ex15	Modifier événement	Comptable, Admin
Ex16	Supprimer événement	Comptable, Admin
Ex17	Cocher dette	Comptable, Admin
Ex18	Vérifier infos d'inscription	Secrétaire, Admin
Ex19	Valider inscription	Secrétaire, Admin
Ex20	Ajouter annonce	Secrétaire, Admin
Ex21	Modifier annonce	Secrétaire, Admin
Ex22	Supprimer annonce	Secrétaire, Admin
Ex23	Ajouter étudiant	Secrétaire, Admin
Ex24	Modifier étudiant	Secrétaire, Admin
Ex25	Supprimer étudiant	Secrétaire, Admin
Ex26	Cocher paiement	Secrétaire, Admin
Ex27	Ajouter professeur	Secrétaire, Admin
Ex28	Modifier professeur	Secrétaire, Admin
Ex29	Supprimer professeur	Secrétaire, Admin
Ex30	Ajouter formation	Secrétaire, Admin
Ex31	Modifier formation	Secrétaire, Admin
Ex32	Supprimer formation	Secrétaire, Admin
Ex33	Générer certificats	Secrétaire, Admin
Ex34	S'inscrire	Utilisateur
Ex35	Payer frais d'inscription	Utilisateur
Ex36	Payer en ligne	Étudiant
Ex37	Rejoindre séance en ligne	Étudiant
Ex38	Consulter formations	Étudiant, Utilisateur
Ex39	Demander inscription formation	Étudiant
Ex40	Consulter événements	Étudiant, Utilisateur
Ex41	Consulter annonces	Étudiant, Utilisateur
Ex42	Consulter notifications	Étudiant, Professeur
Ex43	Consulter profil	Étudiant, Professeur, Secrétaire, Comptable, Admin
Ex44	Modifier infos personnelles	Étudiant, Professeur, Secrétaire, Comptable, Admin
Ex45	Consulter ses formations	Étudiant, Professeur
Ex46	Participer au chat	Étudiant, Professeur
Ex47	Demander sortie d'une formation	Étudiant
Ex48	Demander documents justificatifs	Étudiant, Professeur

Num	Exigence	Acteur
Ex49	Attribuer notes aux étudiants	Professeur
Ex50	Ajouter documents	Professeur
Ex51	Créer une séance en ligne	Professeur
Ex52	Envoyer lien de paiement	Système de paiement
Ex53	Ajouter paiement aux revenus	Système de paiement
Ex54	Proposer suggestions de budget	Bot de l'IA
Ex55	Répondre automatiquement	Bot de protestations

5 Diagramme des cas d'utilisation

Un diagramme de cas d'utilisation montre, de façon visuelle, comment des utilisateurs ou systèmes externes interagissent avec une application. Il met en avant ce que le système permet de faire, sans entrer dans les détails techniques, et aide à comprendre les fonctionnalités offertes du point de vue de l'utilisateur.

5.1 Diagramme de cas d'utilisation de l'utilisateur

Ce diagramme illustre les actions accessibles à un utilisateur non inscrit, comme naviguer sur le site, consulter les formations, voir les annonces et déposer des protestations. Cependant, pour bénéficier des priviléges d'un étudiant, il doit d'abord s'inscrire.

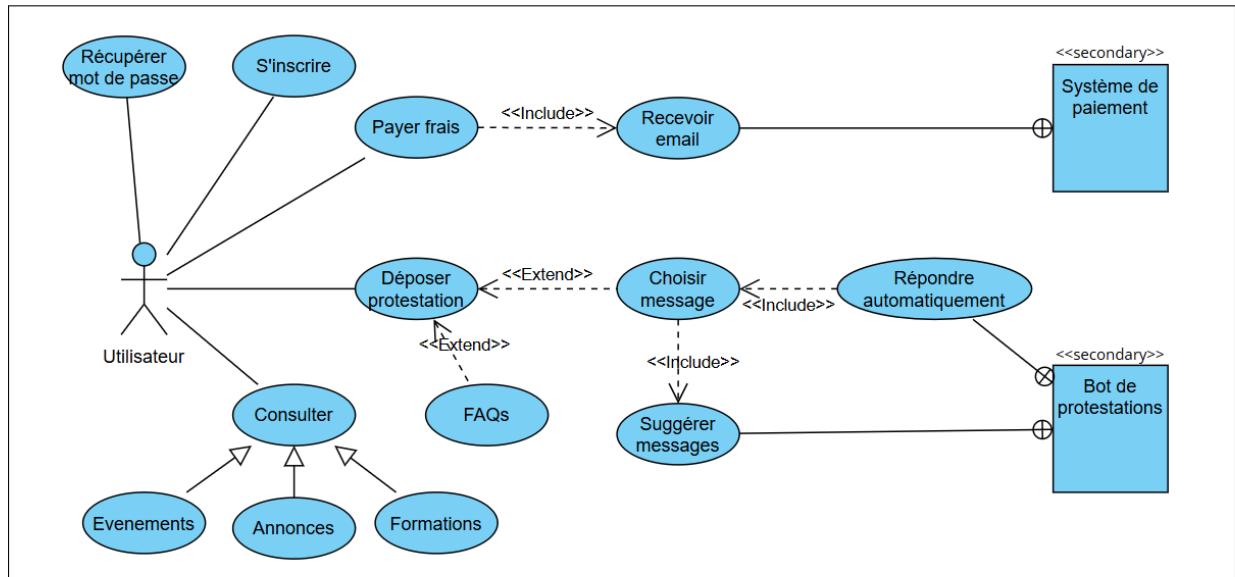


FIGURE 2.3 – Diagramme de cas d'utilisation de l'utilisateur

5.2 Diagramme de cas d'utilisation de l'étudiant

Ce diagramme illustre les actions disponibles pour un étudiant ayant complété son inscription et payé les frais. Une fois ces étapes franchies, l'étudiant peut accéder aux formations, consulter les événements et annonces, gérer son profil et suivre ses inscriptions aux formations. Il dispose ainsi de toutes les fonctionnalités essentielles pour interagir pleinement avec la plateforme.

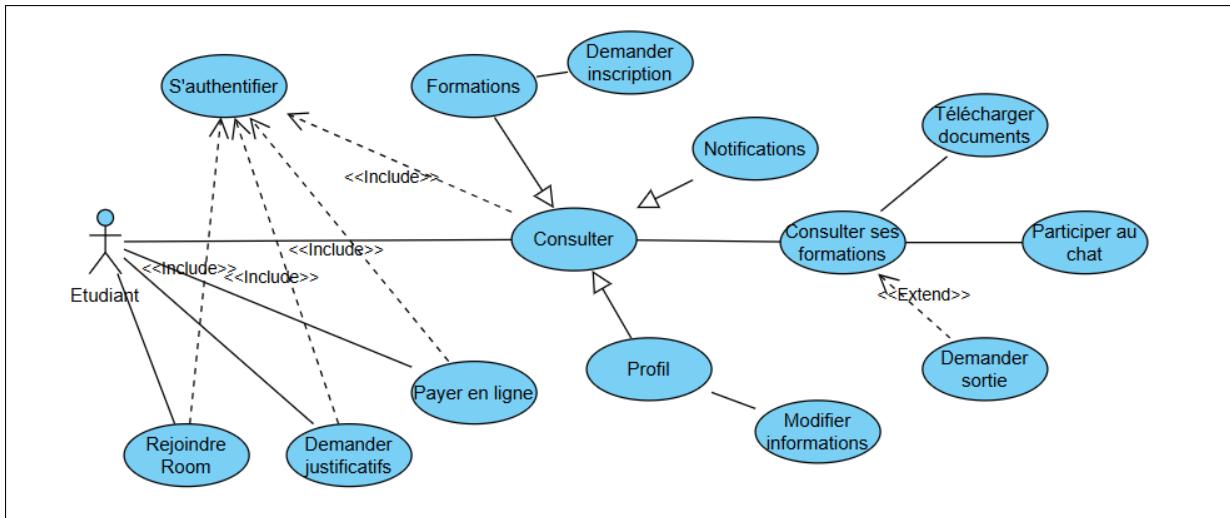


FIGURE 2.4 – Diagramme de cas d'utilisation de l'étudiant

5.3 Diagramme de cas d'utilisation du professeur

Ce diagramme montre les actions disponibles pour un professeur. Il peut consulter ses formations, attribuer des notes aux étudiants, ajouter des documents pédagogiques dans la bibliothèque numérique et créer des séances en ligne pour ses cours. Ces fonctionnalités lui permettent de gérer efficacement ses interactions avec les étudiants et de suivre leur progression.

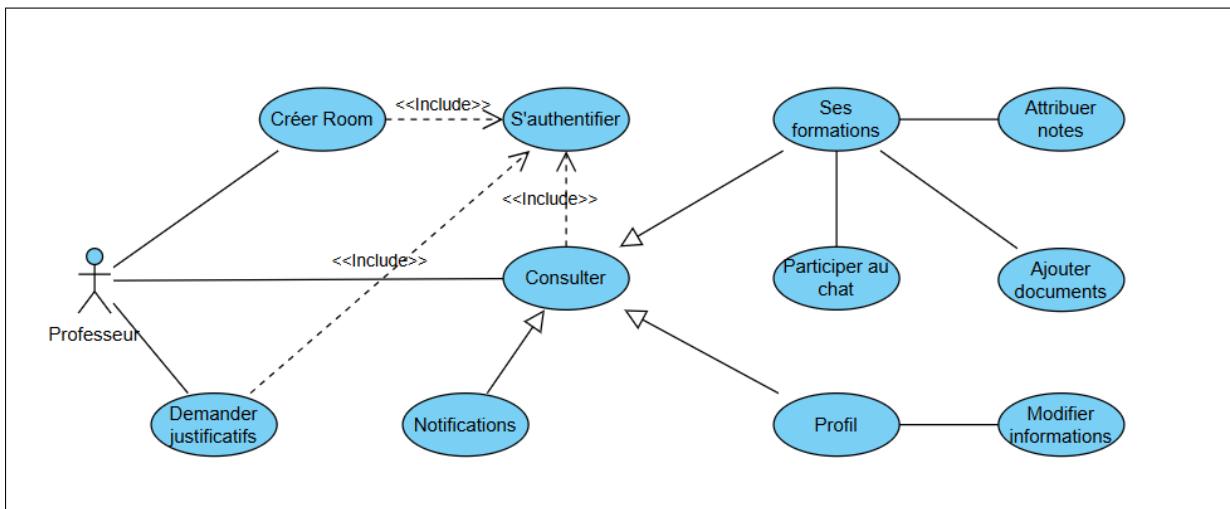


FIGURE 2.5 – Diagramme de cas d'utilisation du professeur

5.4 Diagramme de cas d'utilisation du comptable

Ce diagramme illustre les actions disponibles pour le comptable. Il peut consulter les protestations, les dépenses, les revenus, les dettes, et le budget de la plateforme.

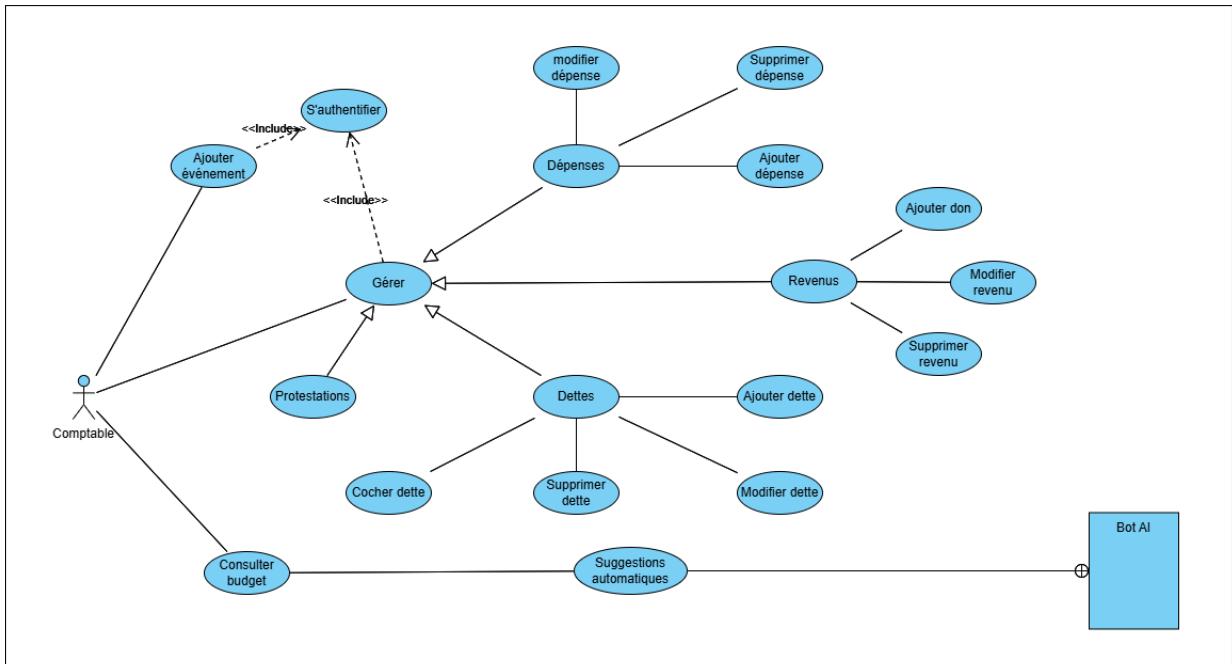


FIGURE 2.6 – Diagramme de cas d'utilisation du comptable

5.5 Diagramme de cas d'utilisation de la secrétaire

Ce diagramme illustre les actions de la secrétaire : gestion des inscriptions, des annonces, des étudiants, des professeurs, des formations, des paiements et génération de certificats, assurant le suivi administratif de la plateforme.

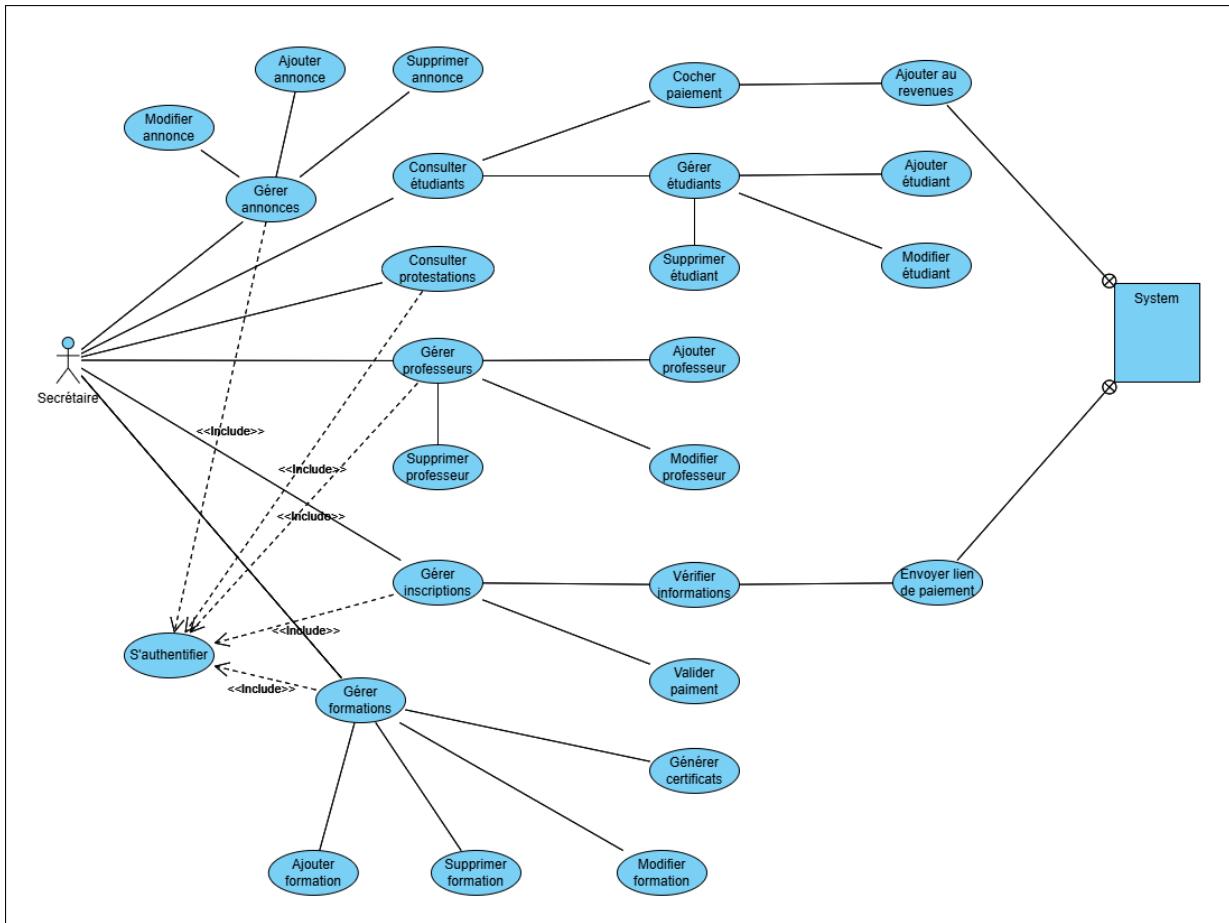


FIGURE 2.7 – Diagramme de cas d'utilisation de la secrétaire

5.6 Diagramme de cas d'utilisation de l'admin

Ce diagramme présente les actions accessibles à l'administrateur. En plus de disposer de tous les priviléges des secrétaires et des comptables, il peut également gérer leurs comptes.

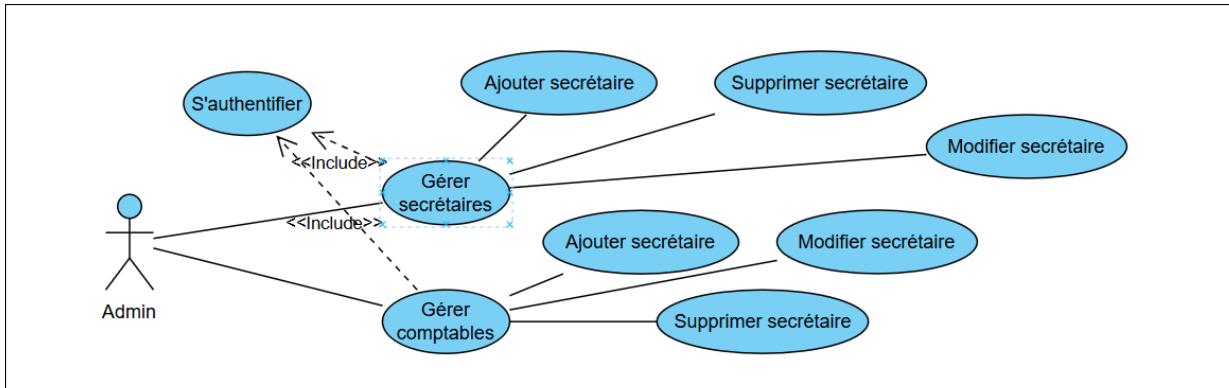


FIGURE 2.8 – Diagramme de cas d'utilisation de l'admin

6 Diagrammes de séquences

Les diagrammes de séquence permettent de visualiser comment les objets d'un système interagissent au fil du temps. Ils montrent l'enchaînement des messages échangés dans un scénario précis, ce qui aide à mieux comprendre le fonctionnement du système et à faciliter la communication entre les membres de l'équipe. Dans ce qui suit, on présente quelques exemples de ces diagrammes pour illustrer certains cas d'utilisation.

6.1 Diagramme de séquence pour le cas d'utilisation "Inscription au site"

Un utilisateur fait une inscription au site, si la secrétaire refuse cette demande un email d'annulation sera envoyé à lui, sinon on lui envoi un email avec lien de paiement. L'utilisateur effectue le paiement, si le paiement est non valide la secrétaire refuse et un email de refus sera envoyé, sinon un email de validation sera envoyé contenant le reçu de paiement, et le compte étudiant sera créé.

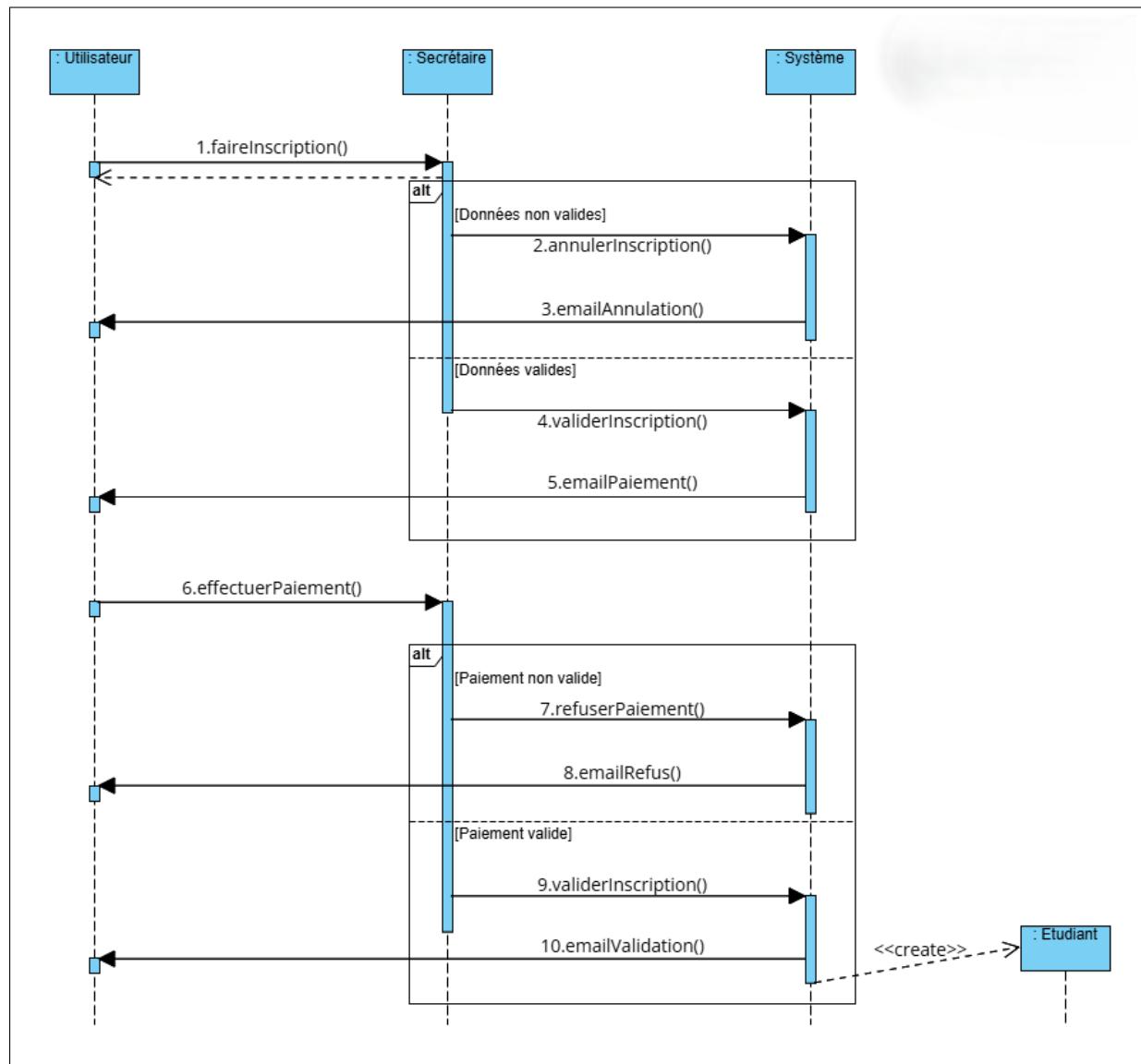


FIGURE 2.9 – Diagramme de séquence "Inscription au site"

6.2 Diagramme de séquence pour le cas d'utilisation "Inscription à une formation"

Un étudiant demande de rejoindre une formation, si la secrétaire refuse cette demande un email d'annulation sera envoyé à lui, sinon on lui envoi un email avec lien de paiement. L'utilisateur effectue le paiement, si le paiement est non valide la secrétaire refuse et un email de refus sera envoyé, sinon un email de validation sera envoyé contenant le reçu de paiement, et le compte étudiant sera créé.

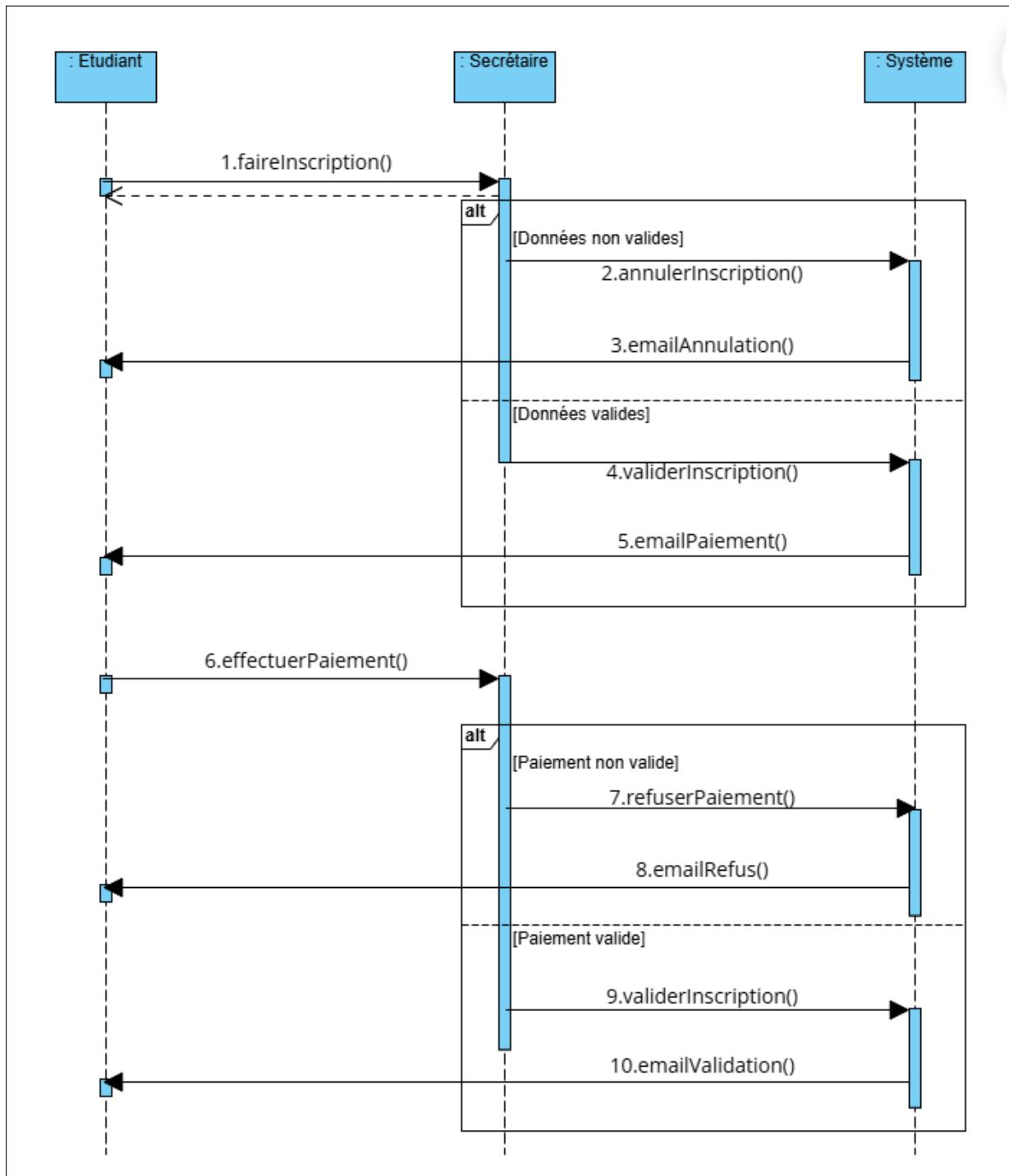


FIGURE 2.10 – Diagramme de séquence "Inscription à une formation"

6.3 Diagramme de séquence pour le cas d'utilisation "Paiement mensuel"

Un étudiant effectue son paiement mensuel, si la secrétaire refuse cette demande un email d'annulation sera envoyé à lui, sinon il reçoit un email avec son reçu de paiement.

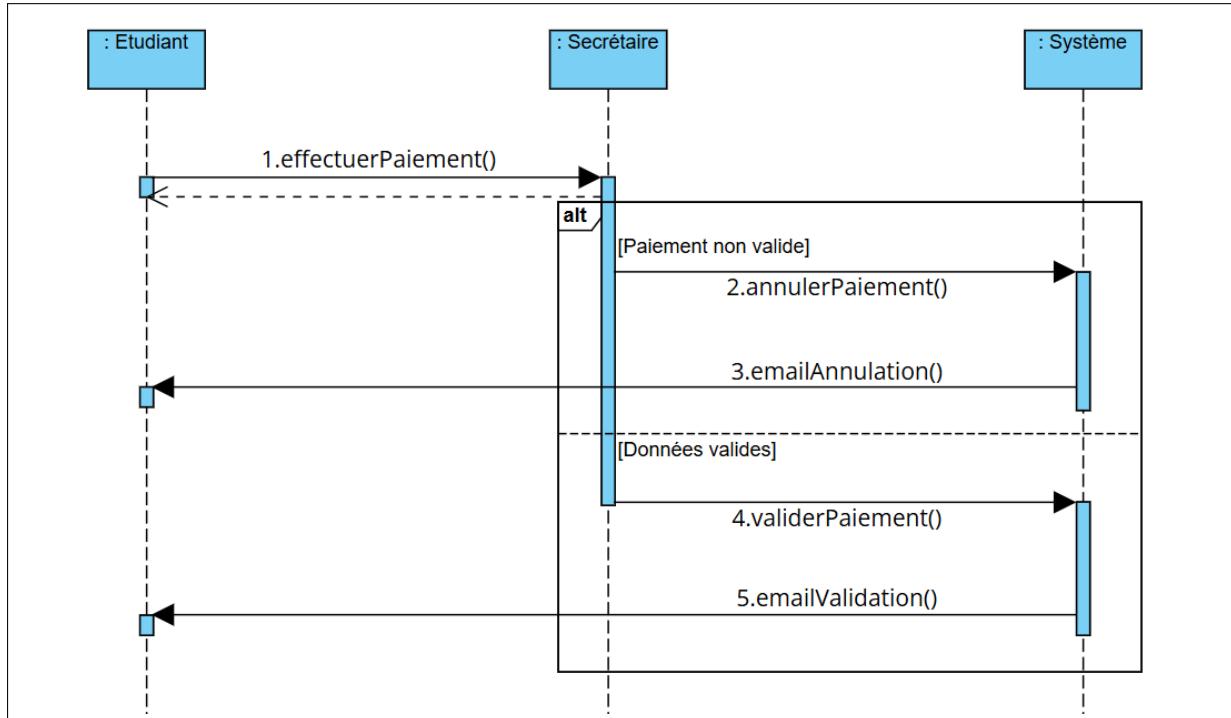


FIGURE 2.11 – Diagramme de séquence "Paiement mensuel"

6.4 Diagramme de séquence pour le cas d'utilisation "Déposer une protestation"

Un utilisateur ou étudiant dépose une protestation sur le site. Quand la secrétaire y répond, La réponse va être envoyée à lui par mail, et en plus, cette réponse et la question seront transmis au FAQs après avoir supprimé la protestation.

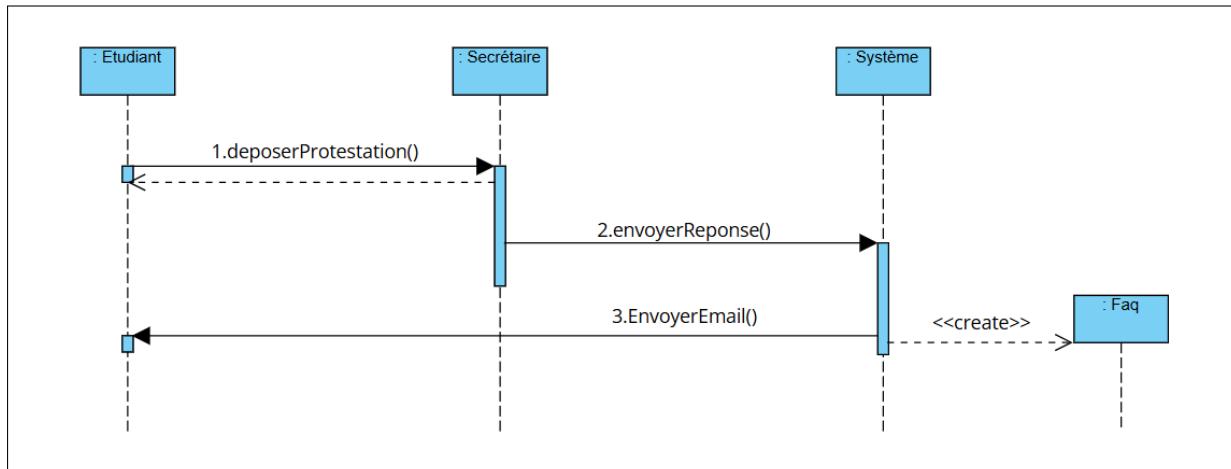


FIGURE 2.12 – Diagramme de séquence "Déposer une protestation"

6.5 Diagramme de séquence pour le cas d'utilisation "Rejoindre une séance en ligne"

Un professeur crée une salle et obtient son code et le partage avec ces étudiants. Un étudiant peut utiliser ce code pour rejoindre la séance en ligne.

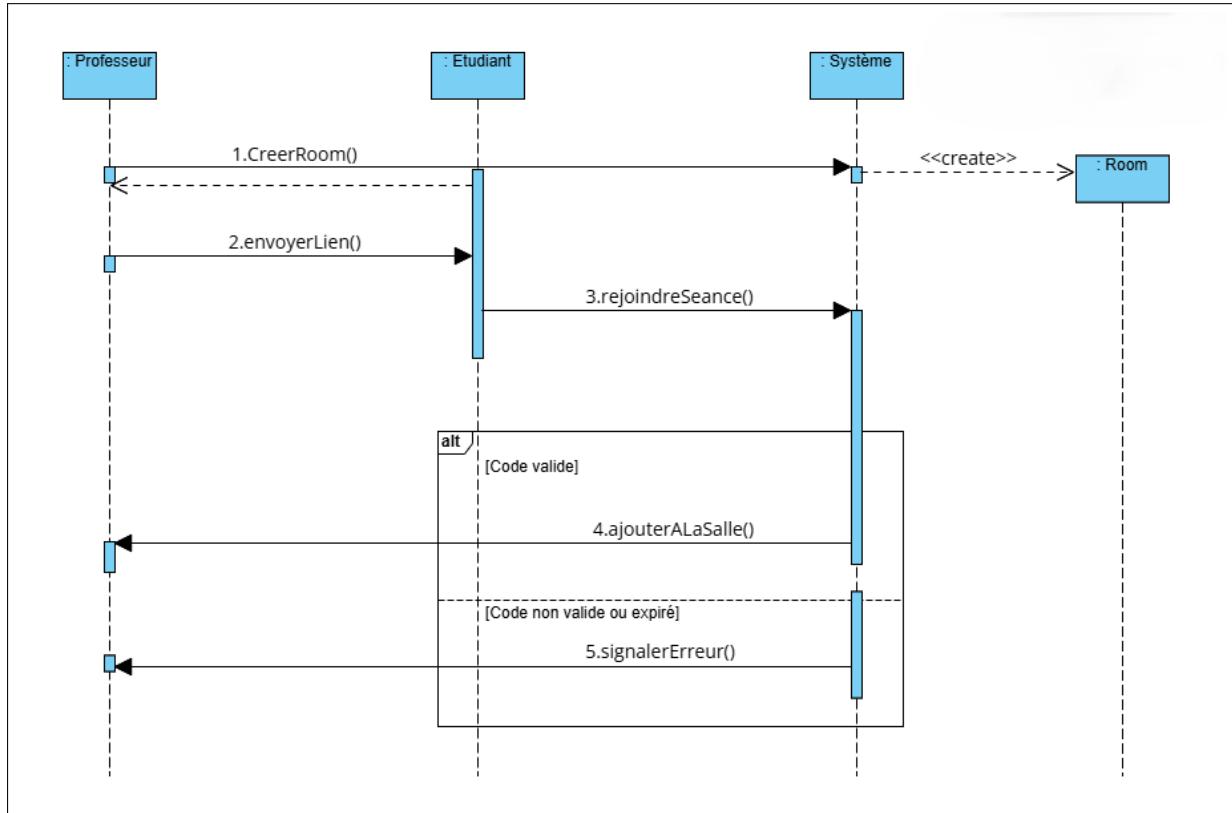


FIGURE 2.13 – Diagramme de séquence "Rejoindre une séance en ligne"

6.6 Diagramme de séquence pour le cas d'utilisation "Récupérer mot de passe"

En cas d'oubli du mot de passe, un utilisateur peut cliquer sur le bouton « Mot de passe oublié » pour lancer la procédure de récupération.

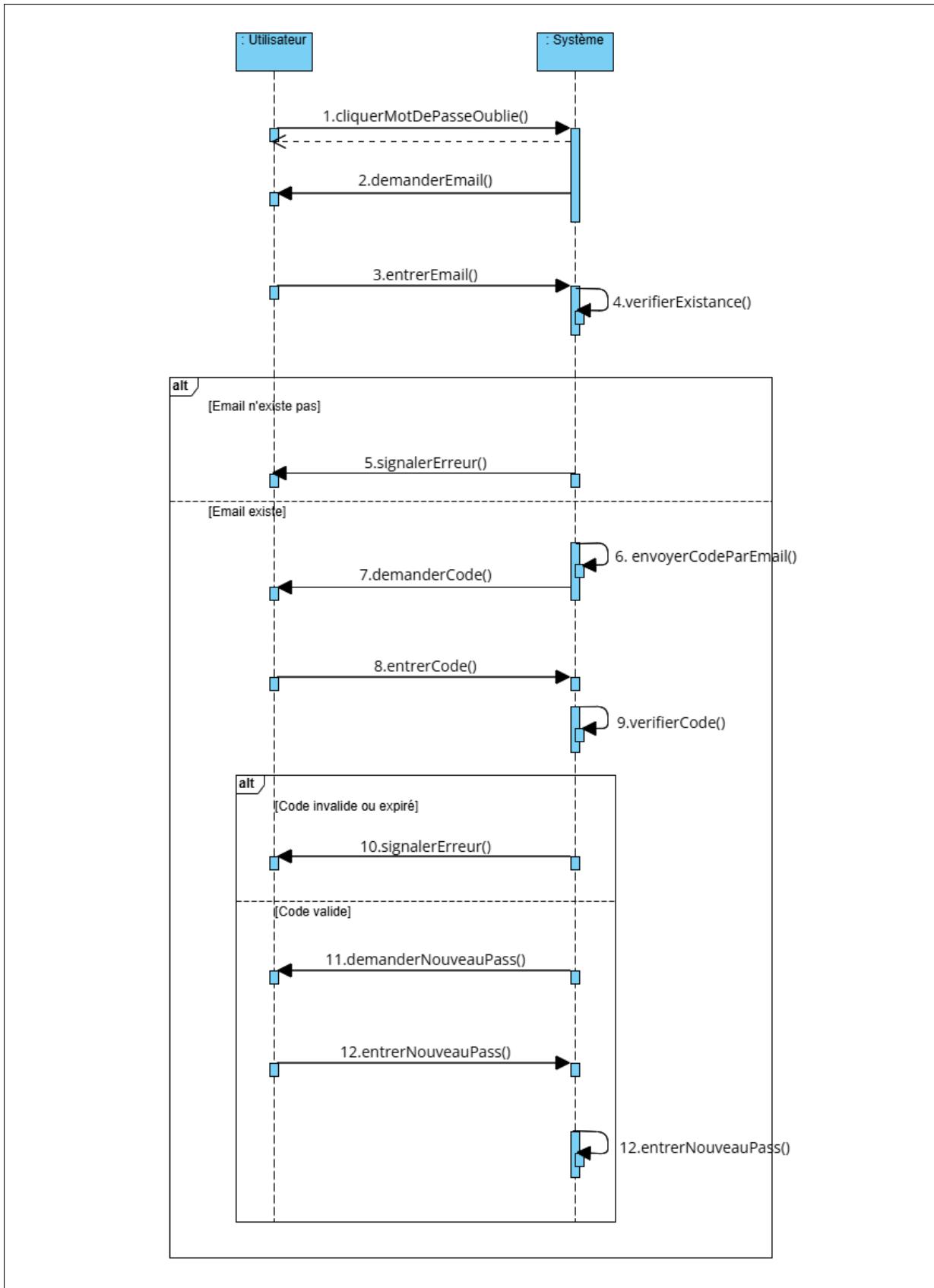


FIGURE 2.14 – Diagramme de séquence "Récupérer mot de passe"

7 Diagrammes d'Etat-transition

Le diagramme d'états-transitions (ou diagramme d'états) est un diagramme comportemental utilisé en génie logiciel et en modélisation UML pour représenter les différents états possibles d'un objet au cours de son cycle de vie, ainsi que les transitions qui permettent de passer d'un état à un autre en réponse à des événements ou conditions spécifiques.

Ce type de diagramme montre :

- Les états : les situations stables dans lesquelles l'objet peut se trouver (ex. : « En attente », « Validé », « Payé »).
- Les transitions : les changements d'état déclenchés par des événements ou des actions (ex. : « Payer », « Valider inscription »).
- L'état initial (souvent représenté par un point noir) : point de départ du cycle de vie de l'objet.
- L'état final (souvent un cercle entouré) : point d'achèvement du processus.

Il est particulièrement utile pour décrire le comportement dynamique d'un système, en illustrant comment un objet réagit aux événements selon l'état dans lequel il se trouve, ce qui aide à concevoir des systèmes robustes et bien définis.

7.1 Diagramme d'état-transition pour une inscription au site

Lorsqu'une inscription est envoyée, elle peut être soit refusée soit acceptée, si elle est acceptée c'est à dire qu'elle devient en attente de paiement, puis elle passe à l'état "compte créé" seulement si le paiement est accepté, sinon elle se marque comme non validée et revient à l'état en attente de paiement.

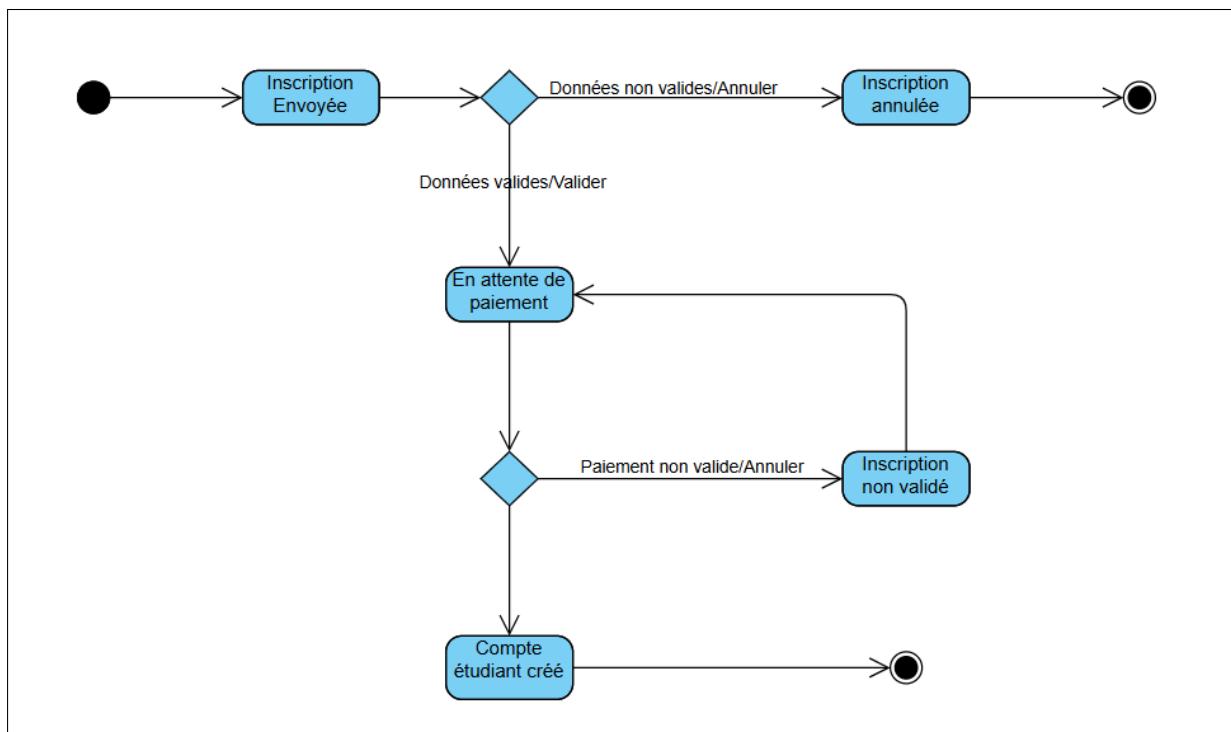


FIGURE 2.15 – Diagramme d'état-transition "Inscription au site"

7.2 Diagramme d'état-transition pour la transition protestation-faq

Une protestation posée peut être soit répondue soit supprimée, si elle est répondue et va devenir un FAQ.

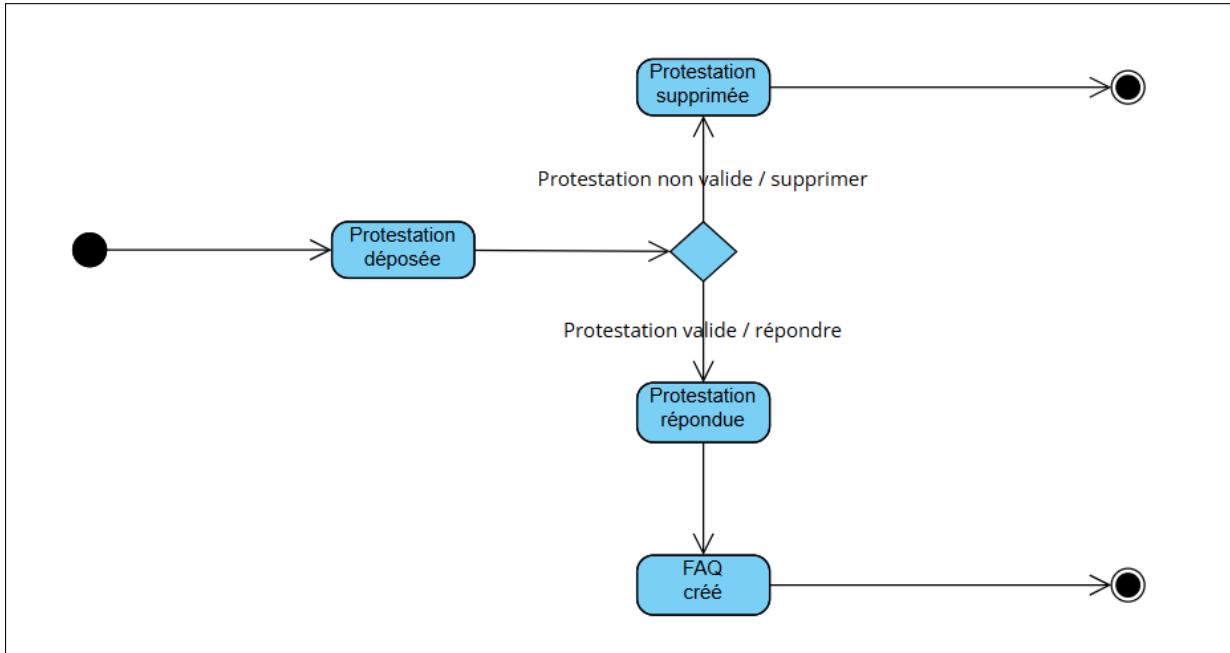


FIGURE 2.16 – Diagramme d'état-transition "Protestation"

7.3 Diagramme d'état-transition pour une séance en ligne (Room)

Lorsqu'une séance est démarrée elle passe à l'état en cours, une fois l'heure de fin est atteinte elle se termine puis elle se supprime quand elle devient vide à nouveau.

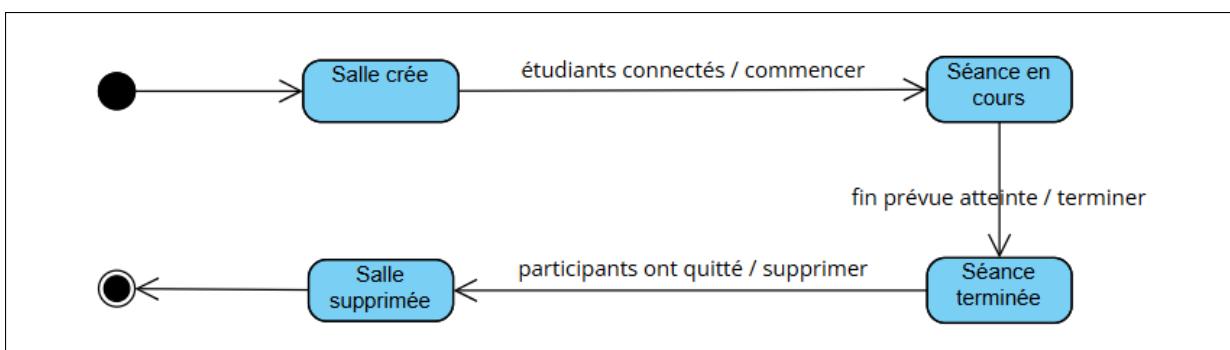


FIGURE 2.17 – Diagramme d'état-transition "Room"

8 Diagramme des Classes

Un diagramme de classes est un outil clé de la modélisation orientée objet, utilisé lors de la conception d'un système. Il représente visuellement les composants du système sous forme de classes, mettant en évidence :

- **Attributs** : propriétés des classes.
- **Méthodes** : fonctions des classes.
- **Relations** : héritage, association, agrégation, composition.

8.1 Description des Classes

Ce tableau décrit chaque classe du diagramme UML utilisé pour modéliser le système de gestion de l'académie. Il met en évidence le rôle principal des classes afin de faciliter la compréhension du modèle.

TABLEAU 2.2 : Description des classes

Classe	Description
Personne	Représente tout utilisateur du système, avec les attributs communs comme nom, email, mot de passe, etc. Base pour l'héritage.
Utilisateur	Représente tout utilisateur non inscrit.
Etudiant	Hérite de Utilisateur, représente les apprenants. Gère les inscriptions, paiements, protestations, et rooms.
Professeur	Hérite de Utilisateur, représente les enseignants. Peut créer des rooms, ajouter des documents et noter les étudiants.
Secrétaire	Hérite de Utilisateur, gère les inscriptions, étudiants, formations, certificats, annonces et professeurs.
Comptable	Hérite de Utilisateur, gère les dépenses, revenus, dettes et événements.
Formation	Représente une formation ou un cours avec un nom, prix, type, et un professeur assigné.
Tag	Ensemble de tags pour une formation.
Bot	Contient des couples questions-réponses prédéfinis, affichées à un étudiant ou utilisateur comme choix rapides lors de la déposition d'une protestation.
Inscription	Lien entre un étudiant et une formation, avec la date d'inscription.
Attente	Pour les étudiants demandants de rejoindre une formation.
Frais	Pour les frais d'inscriptions.
Paiement	Représente un paiement effectué par un étudiant avec montant, date et statut. Pour chaque étudiant, il sera mis à jour à chaque fois dès son inscription.
Document	Document pédagogique ajouté par un professeur et lié à une formation.
Room	Représente une classe virtuelle ou un espace de discussion créé par un professeur.

Classe	Description
NotificationProf	Notification envoyée à un professeur (contenu, type, date).
NotificationEtud	Notification envoyée à un étudiant (contenu, type, date).
VerifCode	Code de vérification pour la récupération du mot de passe en cas d'oublié, avec une date limite d'expiration pour plus de sécurité.
Certificat	Certificat délivré à un étudiant après avoir suivi une formation.
Evenement	Représente un événement organisé par l'académie (titre, description, date).
Dette	Représente une dette contractée par l'académie avec deadline et statut.
Depense	Dépense financière enregistrée dans le système par le comptable.
Revenu	Entrée financière (ex. paiements) enregistrée soit automatiquement soit par le comptable.
Protestation	Requête ou réclamation déposée par un étudiant ou un utilisateur concernant un problème.
Séance	Designe le programme hebdomadaire pour une formation (jour, horaire).
FAQ	Contient des couples questions-réponses déjà posés par d'autres utilisateurs, ils seront affichés à un étudiant ou utilisateurs lors de la déposition d'une protestation.
Salaire	Contient les salaires des professeurs payés.

8.2 Relations entre les classes

- **Personne** est une classe de base dont héritent **Utilisateur**, **Etudiant**, **Professeur**, **Secretaire**, **Admin**, et **Comptable**.
- **Etudiant** peut s'inscrire à une **Formation** et effectuer son **Paiement**.
- **Utilisateur** peut payer des **Frais** d'inscription.
- **Professeur** peut enseigner une **Formation**, ajouter un **Document**, et créer une **Room**.
- **NotificationProf** et **NotificationEtud** sont envoyées à **Professeur** et **Etudiant**, respectivement.
- **Document** est associé à **Formation** et peut être ajouté par **Professeur**.
- **Message** est envoyé par un **Etudiant** ou un **Professeur** et peut appartenir à une **Formation**.
- **Inscription** et **Attente** est liée à **Etudiant** et **Formation**.
- **Tag** est associé à **Formation** et peut être possédé par plusieurs **Formations**.
- **Seance** est associée à **Formation**.
- **Certificat** est généré pour **Etudiant** après la validation de **Formation**.

8.3 Présentation du diagramme

La figure si-dessous représente le diagramme de classes de notre projet :

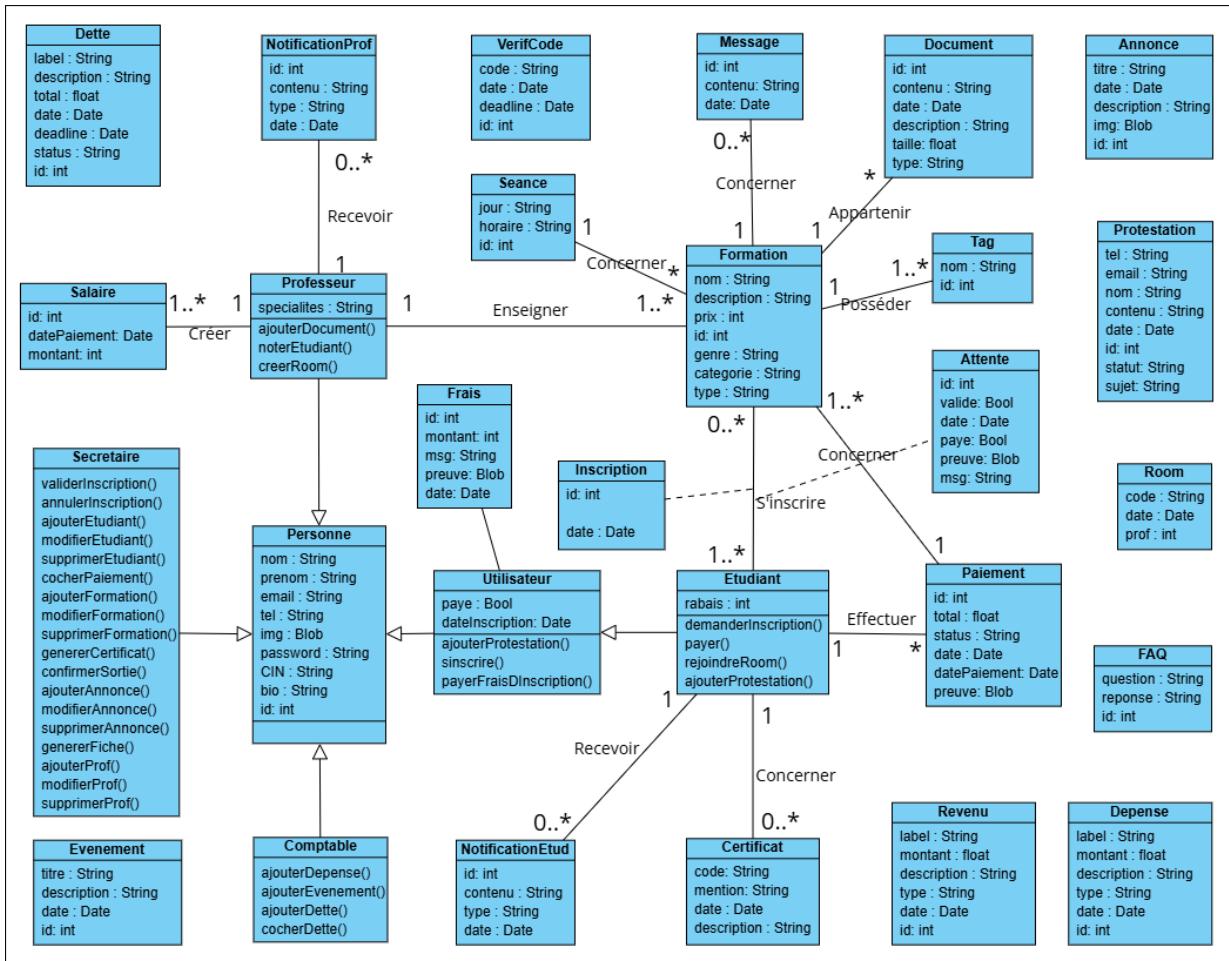


FIGURE 2.18 – Diagramme de classes

9 Diagramme des Packages

Le diagramme de packages est un diagramme structurel de l'UML qui permet d'organiser les différents éléments d'un système en regroupant les classes, interfaces, ou composants similaires au sein de packages. Il sert principalement à représenter l'architecture logique du projet en mettant en évidence les relations de dépendance entre ces groupes. Ce diagramme facilite la compréhension de la structure globale du système, la modularité du code, ainsi que la répartition des responsabilités. Il est particulièrement utile pour les grands projets où il est essentiel de segmenter l'application en sous-systèmes cohérents et indépendants.

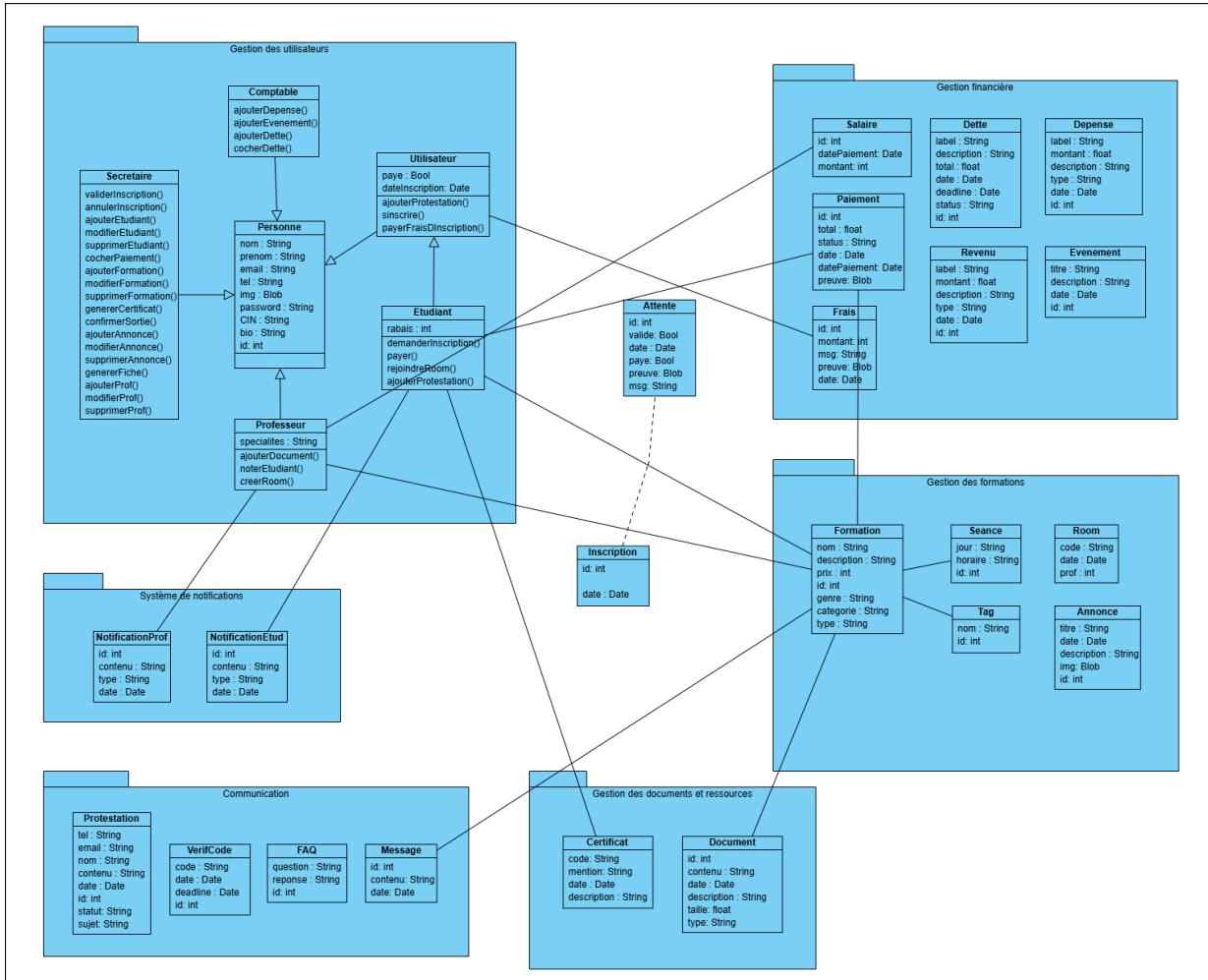


FIGURE 2.19 – Diagramme des packages

IV Structure de la base de données

1 Modèle physique de données

Le Modèle Physique des Données (MPD) est la dernière étape de conception d'une base, traduisant le MCD en tables prêtes à être implémentées dans un SGBDR via le langage SQL. Il définit concrètement la structure de la base.

Voici une présentation schématique du MPD à notre conception :

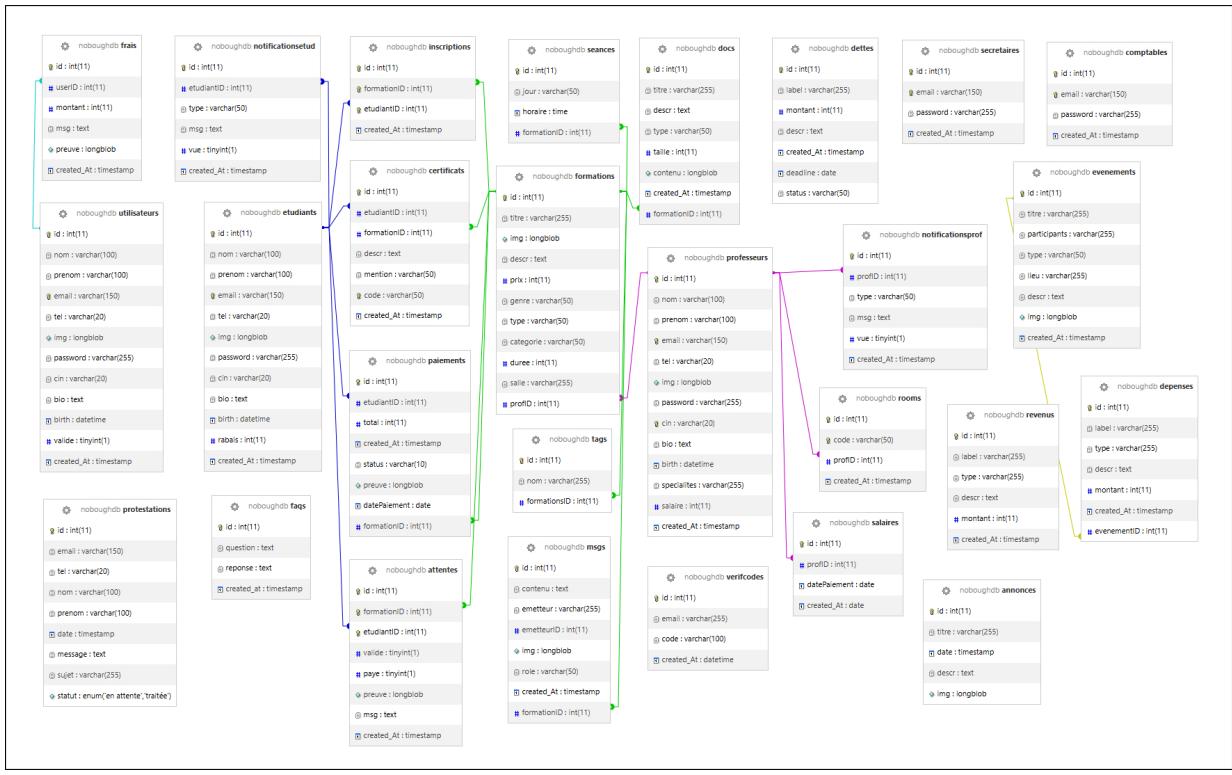


FIGURE 2.20 – Modèle physique de données

V Conclusion

Dans ce chapitre dédié à la conception, nous avons défini les différents acteurs du système, identifié les exigences fonctionnelles et non fonctionnelles, puis élaboré les cas d'utilisation correspondant à chaque acteur. Ces interactions ont été représentées à l'aide de diagrammes de séquence et de diagrammes d'état-transition. Nous avons également détaillé la structure de la base de données. Le prochain chapitre sera consacré à l'architecture logicielle du projet, en mettant en lumière les outils, technologies et bibliothèques choisis pour sa mise en œuvre.

Chapitre 3 : Etude technique du projet

Chapitre 3

Etude technique du projet

I Introduction

Dans cette partie, nous visons à présenter les outils, technologies et l'architecture technique de l'application utilisés pour mettre en œuvre notre solution.

II Architecture du projet

1 Modular Folder-Based Architecture (MFBA)

Le projet suit une architecture modulaire basée sur les dossiers. La Modular Folder-Based Architecture est une approche d'organisation du code où chaque fonctionnalité ou domaine de l'application est séparé dans des dossiers indépendants, chacun contenant tous les éléments nécessaires à son fonctionnement.

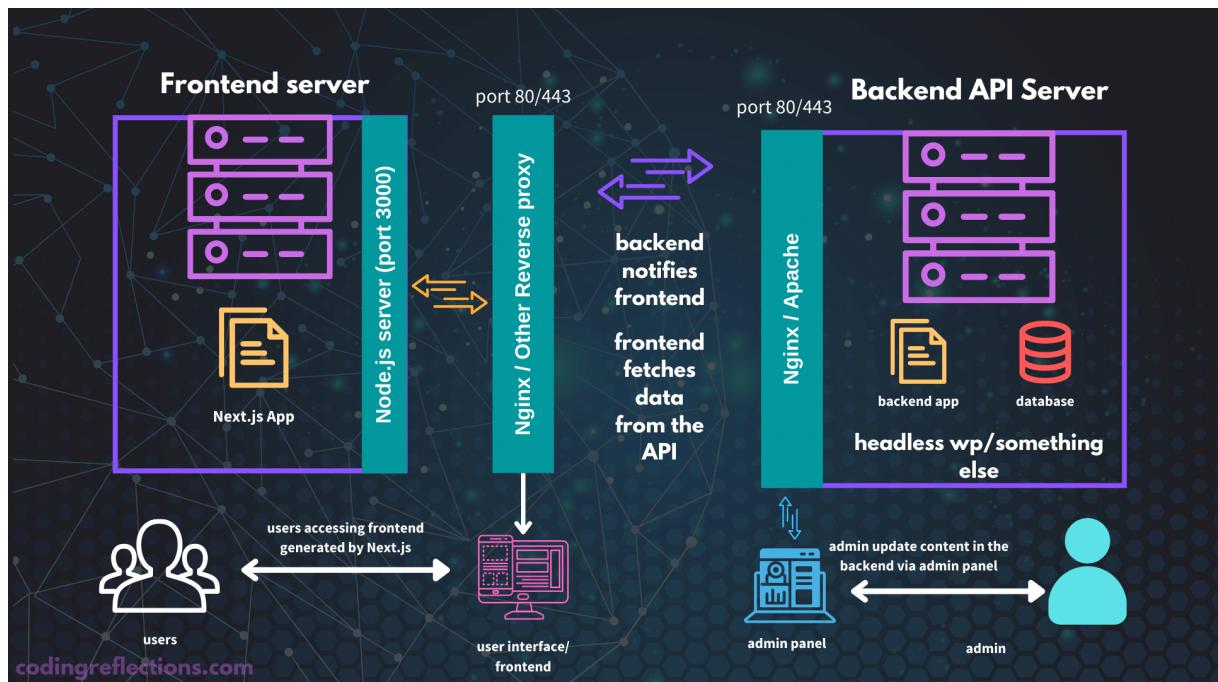


FIGURE 3.1 – L'architecture MFBA

2 Dossiers principales

Cette architecture favorise la clarté, la réutilisabilité, la scalabilité et la maintenance du projet, surtout dans les applications web modernes composées d'un frontend et d'un backend distincts.

- **components/** : contient les composants réutilisables de l'interface utilisateur.
- **layouts/** : comme les components mais plus général, peut contenir des pages, etc.
- **pages/** : gère la structure des pages et le système de navigation.
- **styles/** : contient les fichiers CSS ou utilitaires de style.
- **pages/api/** : logique métier, appels API, etc.
- **hooks/** : fonctions personnalisées (hooks) pour abstraire de la logique réutilisable.
- **middlewares/** : fonctions intermédiaire entre le frontend et les apis pour améliorer la sécurité et gérer les session selon les rôles.
- **utils/** : contient des utilitaires comme les notifications, les gestions des sockets, etc.
- **public/** : dossier spécial pour les medias (images, vidéos, etc.).
- **node-modules/** : contient toutes les bibliothèques et dépendances installées via npm pour le projet.
- **.next/** : dossier généré automatiquement par Next.js, contenant les fichiers compilés et optimisés pour le rendu côté serveur et côté client.

Le projet utilise un dossier **pages/** pour gérer le routage. Chaque fichier représente une route de l'application. Le fichier **_app.jsx** est utilisé pour personnaliser le comportement global de l'application (injection de contextes, styles globaux, etc.).

3 API REST

Le projet expose des **API REST** organisées en endpoints. Chaque route d'API correspond à une action spécifique (CRUD) sur des ressources. L'objectif est de séparer le frontend de la logique métier et des données.

- **GET** : récupération de données.
- **POST** : création de nouvelles données.
- **PUT / PATCH** : mise à jour de données existantes.
- **DELETE** : suppression de ressources.

4 Middlewares

Des **middlewares** sont utilisés pour intercepter les requêtes avant qu'elles n'atteignent les handlers. Ils permettent, par exemple, de :

- Vérifier l'authentification.
- Logger les requêtes entrantes.
- Appliquer des règles de sécurité ou de validation.

III Outils utilisés

1 Introduction

Pour développer des applications web modernes, performantes et interactives, il est essentiel de choisir les bons outils et technologies. Parmi les plus populaires et efficaces, Next.js, Tailwind CSS, Socket et SendGrid se démarquent par leur complémentarité et leurs fonctionnalités puissantes.

Next.js est un framework React qui facilite le rendu côté serveur et la génération statique, ce qui améliore la performance et le référencement naturel. Il offre un routage automatique et des optimisations intégrées, ce qui le rend idéal pour construire des applications web modernes, rapides et bien structurées. Sa simplicité d'utilisation et son écosystème riche en font un choix privilégié pour des projets variés.

Tailwind CSS est une bibliothèque utilitaire qui permet de concevoir rapidement des interfaces personnalisées en appliquant des classes directement dans le code HTML. Elle offre une grande flexibilité pour gérer les styles, le responsive design et le mode sombre, tout en permettant un contrôle précis et efficace du design sans être limité par des composants préconçus.

Socket.IO est utilisé pour gérer la communication en temps réel entre le client et le serveur. Cela permet de créer des applications interactives où les données sont échangées instantanément, comme les chats en direct, les notifications ou les jeux multijoueurs. Cette technologie est essentielle pour offrir une expérience utilisateur fluide et dynamique.

SendGrid est un service d'envoi d'emails transactionnels et marketing. Il simplifie la gestion de la messagerie dans une application web, permettant d'envoyer facilement des emails de confirmation, des notifications ou des newsletters avec un bon taux de délivrabilité et un suivi efficace.

2 Environnement logiciel

2.1 Système de gestion de base de données (SGBD)

MySQL est un système de gestion de base de données relationnelle open source très populaire, utilisé pour stocker, organiser et gérer efficacement des données structurées. Il permet de créer, lire, mettre à jour et supprimer des données via le langage SQL (Structured Query Language). MySQL est apprécié pour sa robustesse, sa rapidité, sa scalabilité et sa compatibilité avec de nombreux langages de programmation. Il est largement utilisé dans les applications web pour gérer les données des utilisateurs, les contenus dynamiques et les transactions. Son modèle relationnel facilite les relations entre les différentes tables, assurant ainsi l'intégrité et la cohérence des données.



FIGURE 3.2
– VS Code

3 Environnement logiciel

3.1 Visual Studio Code

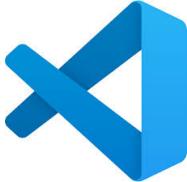


FIGURE 3.3
– VS Code

Visual Studio Code est un éditeur de code source gratuit et open source développé par Microsoft. Il est réputé pour sa légèreté, sa rapidité et ses nombreuses fonctionnalités avancées comme l'autocomplétion, la coloration syntaxique, la refactorisation, et l'intégration Git. Son grand avantage réside dans sa forte personnalisation grâce à un large éventail d'extensions disponibles sur son marketplace. VS Code inclut aussi un terminal intégré et des outils de débogage pour de nombreux langages, ainsi que le support des snippets, facilitant ainsi la productivité.

Par rapport à d'autres éditeurs populaires comme Sublime Text, Atom ou JetBrains IntelliJ, VS Code se distingue par son écosystème extensible, sa gratuité complète, et son support natif pour les conteneurs Docker et les frameworks mobiles comme Flutter ou React Native. Ce choix s'explique aussi par sa communauté active et ses mises à jour fréquentes, offrant un équilibre idéal entre performances, flexibilité et facilité d'utilisation pour des projets variés.

3.2 XAMPP



FIGURE 3.4
– XAMPP

XAMPP est une distribution Apache facile à installer qui regroupe MariaDB, PHP et Perl, spécialement conçue pour le développement local de sites et d'applications web.

Ce choix s'est imposé car XAMPP propose une solution complète et prête à l'emploi, permettant de mettre en place rapidement un environnement web fonctionnel sur une machine locale sans complications techniques. En intégrant Apache, MariaDB (une variante de MySQL) ainsi que PHP et Perl, il facilite grandement le travail des développeurs sur Windows, Linux ou macOS. Comparé à d'autres solutions comme WAMP, MAMP ou LAMP, XAMPP est privilégié pour sa simplicité d'utilisation, son interface intuitive pour gérer les services et bases de données, ainsi que sa compatibilité étendue. Il est particulièrement adapté pour tester et déboguer des projets en local avant leur déploiement, ce qui explique sa popularité auprès de nombreux développeurs.

3.3 Node.js



FIGURE 3.5
– Node.js

Node.js est un environnement d'exécution JavaScript côté serveur, conçu pour créer des applications réseau performantes grâce à son modèle d'E/S non bloquant. Il gère efficacement de nombreuses connexions simultanées, contrairement à des technologies comme PHP ou Java EE. Il permet l'exécution de JavaScript en dehors des navigateurs (côté serveur). Choisi pour sa rapidité et sa flexibilité, Node.js dispose d'un vaste écosystème via npm, facilitant l'ajout de bibliothèques. Par rapport à Django ou Ruby on Rails, il est plus polyvalent, utilisé pour serveurs web, scripts d'automatisation, et optimise les performances grâce aux modules natifs en C++.

4 Langages

4.1 SQL



SQL (Structured Query Language) est un langage standardisé utilisé pour gérer et manipuler des bases de données relationnelles. Il permet de créer, lire, mettre à jour et supprimer des données dans une base structurée en tables, chaque table représentant une entité avec ses attributs. SQL fonctionne avec des systèmes de gestion de bases de données relationnelles (SGBDR) tels que MySQL, PostgreSQL, Oracle ou SQLite. Grâce au support des transactions ACID (Atomicité, Cohérence, Isolation, Durabilité), il garantit la fiabilité et l'intégrité des données.

FIGURE 3.6 Le choix de SQL s'explique par sa maturité, sa robustesse et sa large adoption dans les applications web, où les données structurées sont prépondérantes. Comparé à des solutions NoSQL comme MongoDB ou Cassandra, SQL est préféré pour sa capacité à gérer des relations complexes et assurer une cohérence stricte, ce qui est essentiel dans de nombreux projets nécessitant des opérations transactionnelles précises.

4.2 JavaScript



FIGURE 3.7 JavaScript est un langage de programmation clé pour rendre les pages web interactives, en permettant des effets dynamiques, la validation de formulaires et la gestion d'événements utilisateur. Il est choisi pour sa compatibilité avec tous les navigateurs modernes, contrairement à Flash ou Java applets aujourd'hui moins supportés. Avec Node.js, il s'utilise aussi côté serveur, offrant un langage unique pour toute l'application, ce qui simplifie par rapport à PHP ou Ruby. JavaScript supporte plusieurs styles de programmation (objet, fonctionnel, impératif), le rendant très flexible. Il sert aussi au développement de jeux, d'applications mobiles via React Native, et à l'automatisation avec Puppeteer. Sa grande communauté et ses outils en constante évolution le différencient d'autres langages moins dynamiques. Il est aussi indispensable pour créer des Progressive Web Apps (PWAs), qui offrent une expérience proche des applis natives, un avantage sur les sites classiques.

4.3 CSS



CSS (Cascading Style Sheets) définit la présentation des documents HTML/XML, gérant couleurs, polices et mise en page. Il est essentiel pour des sites adaptatifs grâce aux media queries et aux outils modernes comme Flexbox et Grid, qui simplifient les mises en page complexes. CSS permet aussi animations et transitions pour améliorer l'expérience utilisateur. Préprocesseurs comme SASS facilitent l'écriture.

FIGURE 3.8 On choisit CSS pour sa simplicité, sa compatibilité et son intégration directe, contrairement à Bootstrap (bibliothèque prête à l'emploi) ou Tailwind CSS (style utilitaire), qui offrent des approches différentes avec leurs avantages et limites.

5 Frameworks

5.1 Next.js



FIGURE 3.9
– Next.js

Next.js [2] est un framework basé sur React qui facilite le rendu côté serveur (SSR) et la génération statique (SSG), ce qui améliore la vitesse et le SEO.

On l'a choisi pour sa simplicité, son routage automatique et ses optimisations intégrées (images, polices, scripts), offrant un bon équilibre entre performance et flexibilité. Contrairement à Create React App, qui ne fait que du rendu côté client, Next.js combine SSR et SSG pour une meilleure expérience utilisateur.

Par rapport à d'autres frameworks, Vue (avec Nuxt.js) et Angular offrent aussi des solutions pour SSR et SPA, mais Next.js reste plus simple à prendre en main pour React et bénéficie d'un écosystème riche, ce qui le rend très adapté aux sites e-commerce, blogs ou applications SaaS.

5.2 React

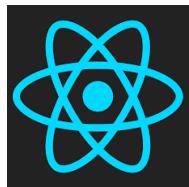


FIGURE 3.10
– React

React est une bibliothèque JavaScript qui facilite la création d'interfaces utilisateur grâce à des composants réutilisables. Son DOM virtuel permet de mettre à jour efficacement l'affichage, ce qui améliore les performances. Les hooks offrent une manière simple et moderne de gérer l'état et les effets dans les composants fonctionnels, tandis que le contexte sert à partager des données globales sans passer par des props à chaque niveau. React est choisi pour sa grande flexibilité et son écosystème riche (comme React Router pour la navigation ou React DevTools pour le débogage). Comparé à Vue.js, plus intuitif pour les débutants, ou Angular, un framework complet qui intègre beaucoup de fonctionnalités nativement (comme la gestion avancée des formulaires ou la programmation réactive), React laisse plus de liberté dans l'architecture et la sélection des outils. Enfin, React Native permet de développer facilement des applications mobiles avec la même base de code.

5.3 Express



FIGURE 3.11
– Express

Express [4] est un framework minimalist pour Node.js, idéal pour créer rapidement des API et des serveurs web grâce à sa simplicité et flexibilité. Il gère facilement les routes, requêtes et middlewares, ce qui le rend très populaire pour des applications web évolutives avec peu de configuration. On a utilisé sa documentation officielle (Voir la Webographie).

On choisit Express surtout pour sa légèreté et sa large adoption dans l'industrie, facilitant l'intégration avec d'autres outils et une communauté active. Par rapport à d'autres frameworks comme NestJS (qui apporte une structure modulaire et TypeScript avancé), FastAPI (en Python, très performant pour les API rapides avec typage automatique) ou Django (un framework Python complet avec ORM intégré et administration automatique), Express reste le choix simple et rapide quand on veut une base souple sans imposer une architecture stricte.

6 Bibliothèques

6.1 Tailwind CSS



FIGURE 3.12

– Tailwind CSS

Tailwind CSS est une bibliothèque utilitaire CSS qui facilite la création rapide de designs personnalisés en appliquant des classes directement dans le HTML. Elle est choisie pour sa grande flexibilité, car elle permet de contrôler précisément les couleurs, espacements, et typographies via un fichier de configuration, tout en offrant un support natif pour le responsive design et le mode sombre.

Comparé à d'autres outils comme Bootstrap, qui propose des composants prédéfinis et un design plus standardisé, Tailwind offre plus de liberté pour créer des interfaces uniques sans être limité par des styles préconçus. Par exemple, Bootstrap facilite la mise en place rapide d'interfaces cohérentes avec peu de personnalisation, tandis que Tailwind permet une personnalisation fine et un contrôle total du style. D'autres alternatives comme Bulma ou Foundation ont des approches similaires à Bootstrap, avec des composants prêts à l'emploi mais moins modulables que Tailwind.

6.2 GSAP



FIGURE 3.13

– GSAP

GSAP (GreenSock Animation Platform) est une bibliothèque JavaScript très performante pour créer des animations web fluides et précises. Elle permet un contrôle avancé des transitions, des interactions, et des animations complexes, notamment grâce à des plugins comme ScrollTrigger (animation au scroll) ou MorphSVG (animation de formes SVG). Son intégration facile avec des frameworks comme React, Vue ou Angular en fait un choix privilégié pour les projets modernes.

On choisit GSAP plutôt que d'autres outils comme Animate.css ou Framer Motion parce qu'il offre plus de flexibilité et de puissance. Animate.css est simple et rapide pour des animations basiques prédéfinies, mais ne permet pas de gérer des séquences complexes ou des interactions poussées. Framer Motion, lui, est excellent pour React et propose des animations déclaratives et faciles à utiliser, mais GSAP reste plus performant pour des animations très personnalisées et complexes, avec un meilleur contrôle au niveau du timing et de la synchronisation.

6.3 Axios



FIGURE 3.14

– Chartjs

Chart.js [11] est une bibliothèque JavaScript open-source qui permet de créer facilement des graphiques interactifs et responsives sur des pages web. Elle supporte différents types de graphiques tels que les barres, les lignes, les secteurs, les radars, et plus encore. Grâce à sa simplicité d'utilisation et à sa documentation claire, Chart.js est largement utilisée pour visualiser des données de manière esthétique et dynamique. Elle fonctionne avec le <canvas> et offre des animations fluides, des options de personnalisation variées et une intégration rapide avec d'autres frameworks JavaScript. Son principal avantage est de permettre aux développeurs de transformer des données brutes en graphiques visuels intuitifs et attractifs.

6.4 jsonwebtoken



FIGURE 3.15
– JWT

jsonwebtoken est une bibliothèque populaire utilisée pour créer et vérifier des JSON Web Tokens (JWT) dans les processus d'authentification et d'autorisation. Elle permet de signer les jetons avec une clé secrète ou une paire de clés publique/privée, garantissant ainsi leur intégrité et authenticité. Cette bibliothèque supporte aussi les claims personnalisés, permettant d'ajouter des informations spécifiques au token. jsonwebtoken est choisie principalement pour sa maturité, sa documentation complète et sa large adoption, ce qui facilite son intégration dans les projets, notamment avec des outils comme Passport.js.

D'autres bibliothèques existent, comme jsonwebtoken-simple qui est plus léger mais offre moins de fonctionnalités avancées, ou jose, une solution plus récente et complète prenant en charge les dernières normes de sécurité mais plus complexe à utiliser. En parallèle, on peut gérer l'authentification via des cookies simples ou le localStorage, mais ces méthodes ne garantissent pas la signature ni la vérification des données, ce qui peut poser des problèmes de sécurité.

6.5 SendGrid



FIGURE 3.16
– SendGrid

SendGrid [5] est un service d'envoi d'emails choisi pour sa fiabilité, son évolutivité, et ses fonctionnalités avancées comme le suivi des emails, la gestion des contacts, et l'analyse des campagnes. Il facilite l'envoi d'emails transactionnels et marketing grâce à des templates dynamiques et une API robuste. Par rapport à d'autres outils comme Mailgun, Amazon SES ou Postmark, SendGrid offre un équilibre entre facilité d'utilisation et fonctionnalités complètes, ainsi qu'une bonne intégration avec des services comme Twilio. Sa documentation claire et sa communauté active renforcent son attractivité pour les applications web. Brevo est limité à 300 emails par jour, ce qui est largement suffisant pour notre application.

6.6 Socket.IO

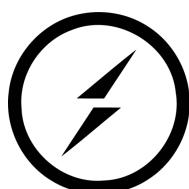


FIGURE 3.17
– Socket.IO

Socket.IO est une bibliothèque qui facilite la communication temps réel bidirectionnelle entre client et serveur, principalement via WebSockets. Elle est idéale pour des applications interactives comme les chats en ligne, les jeux ou tableaux de bord collaboratifs. Ce choix s'explique par ses fonctionnalités avancées comme la reconnexion automatique et la gestion d'erreurs, assurant une expérience fluide.

Comparé à d'autres outils comme WebSocket natif (moins complet), SignalR (spécifique à .NET) ou Firebase Realtime Database (plus orienté stockage temps réel), Socket.IO offre plus de flexibilité, une API riche et une grande compatibilité avec divers frameworks. Sa large communauté et sa documentation en font un outil fiable et facile à intégrer, y compris pour des applications IoT nécessitant un échange rapide et stable entre appareils.

6.7 Nodemailer



FIGURE 3.18
– Nodemailer

Nodemailer est une bibliothèque Node.js simple et puissante pour envoyer des emails. On la choisit souvent pour sa facilité d'usage, son support natif de nombreux services comme Gmail ou Outlook, et sa capacité à gérer des contenus HTML, pièces jointes, et templates. Par rapport à d'autres outils comme SendGrid ou Mailgun, qui sont des services externes spécialisés dans l'email marketing et la délivrabilité à grande échelle, Nodemailer est idéal pour des projets où on veut garder le contrôle total et une intégration directe côté serveur, sans dépendre d'une API tierce. SendGrid ou Mailgun offrent des fonctionnalités avancées comme l'envoi en masse, les statistiques détaillées et la gestion des listes, mais nécessitent une inscription et une configuration plus complexe. Nodemailer reste donc une solution légère, flexible et rapide à mettre en place, adaptée à la plupart des besoins d'envoi d'emails dans les applications Node.js.

6.8 Puppeteer



FIGURE 3.19
– Puppeteer

Puppeteer [9] est une bibliothèque permettant de contrôler des navigateurs headless pour automatiser des tâches web comme le scraping, les tests ou la génération de PDF. On l'a choisi pour sa compatibilité avec Chromium, sa capacité à simuler des interactions complexes et son API puissante, facilitant l'exécution de scripts JavaScript directement dans le navigateur.

Comparé à d'autres outils comme Selenium (plus généraliste, compatible avec plusieurs navigateurs) ou Playwright (plus récent et multi-navigateurs), Puppeteer est souvent préféré pour sa simplicité et son intégration étroite avec Chromium. Par exemple, Selenium est adapté si on veut tester sur Firefox ou Safari, tandis que Playwright offre une gestion plus avancée du multi-navigateurs mais avec une courbe d'apprentissage un peu plus élevée. Puppeteer reste donc un bon compromis pour des automatisations ciblées sur Chromium avec une documentation riche et une communauté active.

7 Outils AI

7.1 TensorFlow.js



FIGURE 3.20
– TensorFlow

TensorFlowjs [6] est une bibliothèque JavaScript qui permet d'exécuter et d'entraîner des modèles d'IA directement dans le navigateur ou sur Node.js. Elle est choisie pour sa flexibilité, sa compatibilité GPU, et sa large communauté, facilitant le développement d'applications comme la vision par ordinateur ou le traitement du langage naturel sans serveur. Comparé à d'autres outils d'IA comme Brain.js, plus simple mais moins puissant, ou ONNX.js, orienté vers l'exécution de modèles pré-entraînés de différents frameworks, TensorFlowjs offre un équilibre entre performance, personnalisation et support. Par exemple, Brain.js est idéal pour des projets basiques, tandis que TensorFlowjs convient mieux à des applications complexes nécessitant des capacités avancées et une optimisation GPU.

IV Intégration de l'intelligence artificielle

1 Intégration d'un moteur IA pour la gestion des dettes

Cette application web intègre un moteur d'intelligence artificielle afin d'évaluer les dettes les plus urgentes à rembourser selon plusieurs critères financiers. Cette décision est assistée par un modèle mathématique appelé **réseau de neurones**, implémenté à l'aide de la bibliothèque `TensorFlow.js`.

1.1 Qu'est-ce qu'un réseau de neurones ?

Un **réseau de neurones artificiels** est un modèle mathématique inspiré du fonctionnement des neurones biologiques. Il est composé de couches de **neurones** artificiels (ou noeuds), connectés entre eux, et capables d'apprendre à partir de données d'entrée. Chaque neurone prend des valeurs en entrée, les combine selon des *poids* (paramètres appris ou fixés), applique une fonction d'activation (souvent non linéaire), puis transmet un signal en sortie. En empilant plusieurs couches de neurones, on obtient un réseau capable de modéliser des relations complexes entre des variables.

L'objectif d'un réseau est de produire une sortie — ici, un **score de priorité** — en fonction d'un ensemble de caractéristiques numériques associées à chaque dette.

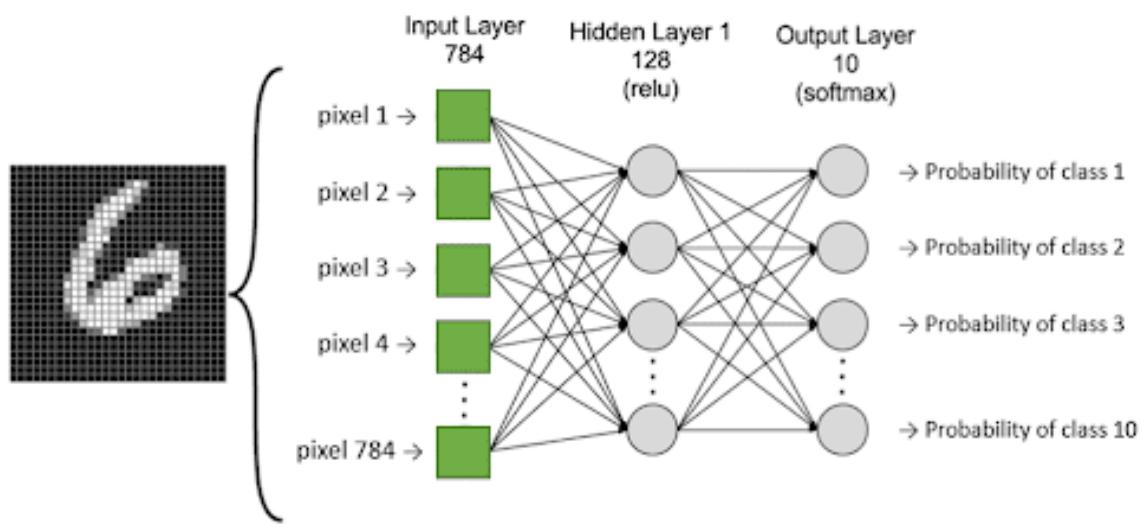


FIGURE 3.21 – Fonctionnement d'un réseau de neurones

1.2 Que sont les tenseurs ?

Les **tenseurs** sont des structures de données représentant des caractéristiques à évaluer. Dans le cadre de l'apprentissage automatique, les tenseurs sont utilisés pour représenter les entrées (features), les poids du modèle, les biais, ainsi que les sorties. Ils seront par la suite attribués à des scores normalisés (entre 0 et 1).

Par exemple :

- Un vecteur $[0.2, 0.5, 0.9, 0.1]$ représentant les caractéristiques normalisées d'une dette est un tenseur 1D.

La bibliothèque `TensorFlow.js` permet de manipuler ces tenseurs, de construire des modèles de réseaux de neurones, et d'exécuter des calculs complexes directement dans le navigateur.

1.3 Prétraitement des Données

Les données des dettes sont d'abord filtrées pour ne conserver que celles qui sont encore non payées. Pour chaque dette, on calcule les caractéristiques suivantes :

- **Jours restants** avant la deadline (`joursRestants`).
- **Ancienneté** de la dette (nombre de jours depuis sa création).
- **Ratio du montant de la dette par rapport au budget réel** (`ratioBudget`).
- **Indication de payabilité** : 1 si la dette peut être couverte par le budget actuel, 0 sinon.

Les variables continues sont normalisées dans des plages définies ($[0,1]$) pour assurer un bon comportement du modèle d'apprentissage.

1.4 Architecture du Modèle IA

Le modèle IA est un réseau de neurones dense séquentiel à trois couches :

1. **Première couche** : 8 neurones, fonction d'activation `ReLU`, connectée à l'entrée à 4 dimensions.
2. **Deuxième couche** : 4 neurones, également avec la fonction d'activation `ReLU`.
3. **Sortie** : 1 neurone, avec la fonction d'activation `sigmoïde`, produisant un score entre 0 et 1 indiquant la priorité de remboursement.

1.5 Initialisation et Prédiction

Le modèle n'est pas entraîné dynamiquement sur les données utilisateur, mais initialisé avec des poids spécifiques pour simuler un comportement de recommandation :

$$W = [0.6, 0.3, 0.9, 0.1]$$

Ces poids reflètent des priorités approximatives :

- L'urgence de la dette (jours restants) est fortement valorisée (0.6) ;
- L'ancienneté est modérément importante (0.3) ;
- Le fait que la dette soit remboursable par le budget est critique (0.9) ;
- Le ratio du budget est ici le moins important (poids 0.1).

Chaque dette reçoit alors un **score IA**, qui est utilisé pour classer les dettes de la plus prioritaire à la moins prioritaire.

1.6 Affichage et Alertes

Une fois les scores calculés, les dettes sont triées. Le système affiche :

- Des alertes si certaines dettes sont proches de leur échéance ou arrivent à leur deadline aujourd’hui ;
- Une recommandation prioritaire si le budget permet de rembourser la dette la plus urgente.

Enfin, une interface graphique présente les résultats dans un tableau avec des barres de couleur illustrant visuellement la priorité de chaque dette.

1.7 Mode de Secours

En cas d'échec du chargement ou de l'utilisation du modèle TensorFlow, une méthode heuristique manuelle est utilisée pour estimer un score basé sur :

$$Score = (30 - \text{joursRestants}) \times 3 + \frac{\text{anciennete}}{10}$$

Cela garantit que l'utilisateur reçoit toujours des recommandations, même sans accès complet au modèle IA.

2 Utilisation de l'API Gemini pour les Recommandations Financières

L'application utilise l'**API Gemini** de Google pour générer automatiquement des recommandations financières adaptées à la situation financière de l'académie selon les données passées à elle, comme : , budget réel, .

- Le budget brut **brut**, et le budget brut **réel**.
- Les **revenus** et les **dépenses**.
- Les **dettes** payées et non payées.
- Les **salaires** des professeurs.

L'appel de l'api est basé sur ce logique :

- Les données de **revenus**, **dépenses**, **dettes** et le **budget** sont formatées en JSON.
- Un **prompt structuré** est envoyé à l'**API gemini-2.0-flash** via une requête **POST**.
- L'API retourne une liste de recommandations sous forme de tableau JSON.
- Le texte reçu est nettoyé et parsé en JavaScript, puis affiché dans l'interface utilisateur.

Cette intégration permet d'obtenir des suggestions intelligentes et actionnables en temps réel, sans intervention manuelle.

V Communication en temps réel avec WebRTC

Pour permettre une communication en temps réel entre les utilisateurs de l'application, notamment entre les professeurs et les étudiants dans une salle virtuelle, nous utilisons la technologie **WebRTC** (Web Real-Time Communication), couplée à la bibliothèque **Socket.io** côté serveur.

1 Principe de WebRTC

WebRTC [3] est une API supportée par les navigateurs modernes permettant d'établir une communication directe (peer-to-peer) entre deux clients, pour transmettre de l'audio, de la vidéo ou tout autre type de données. Ce canal direct permet une faible latence, idéale pour des conférences audio/vidéo ou du partage d'écran. Cependant, l'établissement de cette connexion directe nécessite une phase initiale d'échange d'informations via un serveur de signalisation. C'est là que **Socket.io** intervient.

2 Rôle de Socket.io comme serveur de signalisation

Le serveur Node.js met en place une logique de gestion des salles, de connexion des utilisateurs et de communication des messages WebRTC via des événements `socket.io`.

- `join-room` : Lorsqu'un utilisateur rejoint une salle, son profil est enregistré, et il est notifié des autres participants présents. Si l'utilisateur est un professeur, son rôle est marqué de façon spécifique.
- `webrtc-offer`, `webrtc-answer` : Ces événements transportent les messages d'offre et de réponse SDP (Session Description Protocol) pour initier une connexion entre pairs.
- `webrtc-ice-candidate` : Les candidats ICE sont échangés afin de déterminer le chemin réseau optimal entre les deux pairs (via STUN ou TURN).
- `start-screen-share`, `stop-screen-share` : Ces messages signalent aux autres participants que le professeur a commencé ou arrêté le partage d'écran.
- `toggle-mute`, `toggle-hand-raise` : Ces événements permettent de gérer dynamiquement l'état du microphone ou de lever la main virtuellement, pour une meilleure gestion des interactions.
- `disconnect` : Lorsqu'un utilisateur quitte la salle, il est retiré de la liste des participants et les autres membres sont informés.

3 Avantages de cette architecture

- **Faible latence** : grâce à la connexion directe WebRTC.
- **Scalabilité** : chaque salle est isolée, ce qui permet de gérer plusieurs classes virtuelles indépendamment.
- **Flexibilité côté client** : en manipulant les flux audio/vidéo ou le partage d'écran directement via les API WebRTC du navigateur.
- **Séparation des rôles** : distinction entre professeur (initiateur de la salle, partage d'écran) et étudiants (participants interactifs).

VI Automatisation et optimisation de la gestion des paiements mensuels

1 Problématique

Dans le cadre de la gestion des formations, chaque étudiant inscrit doit régler des paiements mensuels à partir de la date de son inscription. La méthode classique consistant à générer automatiquement chaque mois une ligne de paiement via un processus individuel ou un compteur déclenché à l'inscription présente plusieurs limitations :

- Risque de ralentir l'application à cause de la vision non optimale.
- Difficulté de mise à l'échelle lorsque le nombre d'étudiants augmente.
- Complexité de gestion en cas de désinscription ou de modification de la formation.
- Risque accru de pertes de cohérence ou d'erreurs système.

2 Solution proposée

Pour remédier à ces limites, nous avons mis en place une tâche planifiée (cron job) qui exécute quotidiennement une API. Cette API vérifie toutes les inscriptions, puis pour chaque étudiant et chaque mois écoulé depuis l'inscription, elle insère une ligne de paiement si celle-ci n'existe pas déjà.

Algorithme résumé :

1. Récupérer toutes les inscriptions avec les montants associés à leurs formations.
2. Pour chaque inscription, calculer les mois écoulés depuis la date d'inscription.
3. Pour chaque mois :
 - Vérifier si un paiement a déjà été effectué pour ce mois.
 - Si non, insérer une ligne de paiement avec le statut **nonpaye**.

```
pour chaque formation
    pour chaque étudiant
        date = dateInscription(étudiant)
        tant que date < dateActuelle()
            date += 1 mois
            si !paiement
                insérerPaiement()
```

FIGURE 3.22 – Algorithme des paiements mensuels

3 Automatisation via Cron Job

Afin d'assurer l'insertion automatique des lignes de paiements mensuels pour chaque étudiant inscrit à une formation, nous avons mis en place un mécanisme d'exécution planifiée appelé **Cron Job**.

3.1 Définition :

Un *Cron Job* est une tâche planifiée sous Unix/Linux permettant d'exécuter automatiquement un script ou une commande à des intervalles réguliers (minute, heure, jour, mois, etc.). Il s'appuie sur le démon `cron` qui vérifie en permanence les instructions du fichier `crontab`.

3.2 Objectif :

Notre Cron Job est chargé d'exécuter quotidiennement une requête vers une API développée avec Next.js. Cette API vérifie les inscriptions des étudiants et insère automatiquement les lignes de paiements mensuels manquantes dans la base de données.

3.3 Exemple de configuration :

Voici un exemple de ligne `crontab` qui exécute l'appel à l'API chaque jour à minuit :

```
0 0 * * * curl -s https://noboughacademy.com/api/_lib/scheduler > /dev/null
```

FIGURE 3.23 – Cron-job commande

- **0 0 * * *** signifie : tous les jours à 00h00 (minuit).
- **curl** : outil pour faire une requête HTTP.
- **-s** : mode silencieux (pas d'affichage, même des erreurs).
- **/dev/null** : est un "trou noir" : tout ce qu'on y écrit est supprimé (aucune sortie ne sera visible ou enregistrée dans les logs cron).

3.4 Avantages :

- Évite la complexité d'un compteur manuel ou d'une logique embarquée côté base de données.
- Assure la régularité et la fiabilité des insertions de paiements.
- Permet un contrôle centralisé via les journaux (`logs`) du système.

3.5 Sécurité et robustesse :

L'API utilisée par le Cron Job peut être protégée par un jeton secret (token) ou une authentification IP pour éviter les appels non autorisés. Une gestion des erreurs est également intégrée pour notifier l'administrateur en cas d'échec d'exécution.

4 Optimisation par indexation

Pour optimiser les performances des requêtes les plus fréquentes sur la table `Paiements`, plusieurs index ont été créés. Chaque index a un rôle spécifique :

- L'index composite (`etudiantID`, `formationID`, `created_At`) est le plus important, car il correspond exactement à la majorité des requêtes de vérification de paiements mensuels.
- Les index simples sur `etudiantID` et `formationID` sont utiles si ces colonnes sont utilisées seules dans les requêtes ou dans des jointures.
- L'index sur `datePaiement` est très pertinent pour les rapports mensuels ou les filtres temporels.

MySQL choisit automatiquement le meilleur index selon le type de requête. Une combinaison équilibrée d'index simples et composites permet d'optimiser à la fois les recherches ponctuelles et les requêtes complexes.

Index	Requêtes optimisées
<code>etudiantID</code>	<code>WHERE etudiantID = ?</code>
<code>formationID</code>	<code>WHERE formationID = ?</code>
<code>datePaiement</code>	<code>WHERE datePaiement BETWEEN ...</code>
<code>etudiantID</code> , <code>formationID</code> , <code>created_At</code>	Requêtes combinées sur les 3 colonnes

5 Avantages de la solution

- Traitement centralisé, fiable et facile à auditer.
- Faible charge au moment de l'inscription.
- Possibilité de relancer ou rejouer l'algorithme sans risque de duplication.
- Meilleure évolutivité et tolérance aux erreurs.

VII Conclusion

Dans ce chapitre, nous avons présenté les outils, technologies et l'architecture technique utilisés pour mettre en œuvre notre solution de gestion pour l'Académie Nobough. Nous avons adopté une architecture modulaire basée sur les dossiers (MFBA) pour organiser le code de manière claire et maintenable. Nous avons également utilisé des technologies modernes comme Next.js, React, et Node.js pour construire une application web robuste et évolutive.

Les bibliothèques et frameworks que nous avons intégrés, tels que Tailwind CSS, GSAP, et Axios, nous ont permis de créer une interface utilisateur attrayante et interactive, tout en assurant des performances optimales. Les outils d'authentification comme Next-Auth et jsonwebtoken ont été essentiels pour sécuriser l'application et gérer les sessions utilisateur. Enfin, nous avons utilisé des bibliothèques d'apprentissage automatique comme Brain.js et TensorFlow.js pour intégrer des capacités d'IA dans notre application, améliorant ainsi l'expérience utilisateur avec des fonctionnalités intelligentes.

Ce projet démontre l'importance de choisir les bonnes technologies et de les intégrer de manière cohérente pour créer une application web moderne et performante.

Chapitre 4 : Réalisation de l'application

Chapitre 4

Réalisation de l'application

I Introduction

Dans la phase de réalisation, nous mettrons en avant notre application à travers une série d'interfaces illustratives. Cette présentation visuelle offrira un aperçu concret de la solution développée et permettra de mieux comprendre ses principales fonctionnalités.

II Logo de notre application



FIGURE 4.1 – Logo de l'académie

Le logo de l'académie Nobough (signifiant « brillance intellectuelle » en arabe) est une composition graphique équilibrée et hautement symbolique, illustrant les valeurs fondamentales de l'institution.

Éléments Visuels :

- **Lettrage Nobough en vert** : le mot Nobough, écrit en calligraphie arabe moderne, est présenté dans une teinte verte vive, évoquant la croissance, l'espoir et la quête de connaissance. La fluidité des courbes exprime le dynamisme et l'élan vers l'excellence.
- **Forme de cœur rouge** : entourant le texte, un cœur stylisé en rouge symbolise la passion, l'engagement et l'amour pour l'apprentissage.
- **Cerveau en 3D réaliste** : placé au sommet du logo, un cerveau tridimensionnel illustre le développement cognitif, la créativité et l'intelligence, valeurs centrales de l'académie.

Symbolique Générale : Ce logo crée une synergie visuelle entre l'émotion (le cœur) et la raison (le cerveau), mettant en évidence l'importance d'un apprentissage équilibré, fondé à la fois sur la passion et sur la réflexion intellectuelle. Il reflète la mission de l'académie : cultiver le potentiel de chaque apprenant dans un cadre stimulant et bienveillant.

III Présentation des interfaces de l'application

1 Page d'accueil

Cette page présente de manière simple, claire et rapide des informations essentielles sur notre académie. Elle propose également une section FAQ ainsi que des rappels spirituels à travers des versets du Coran et des paroles du Prophète (paix et bénédictions sur lui).

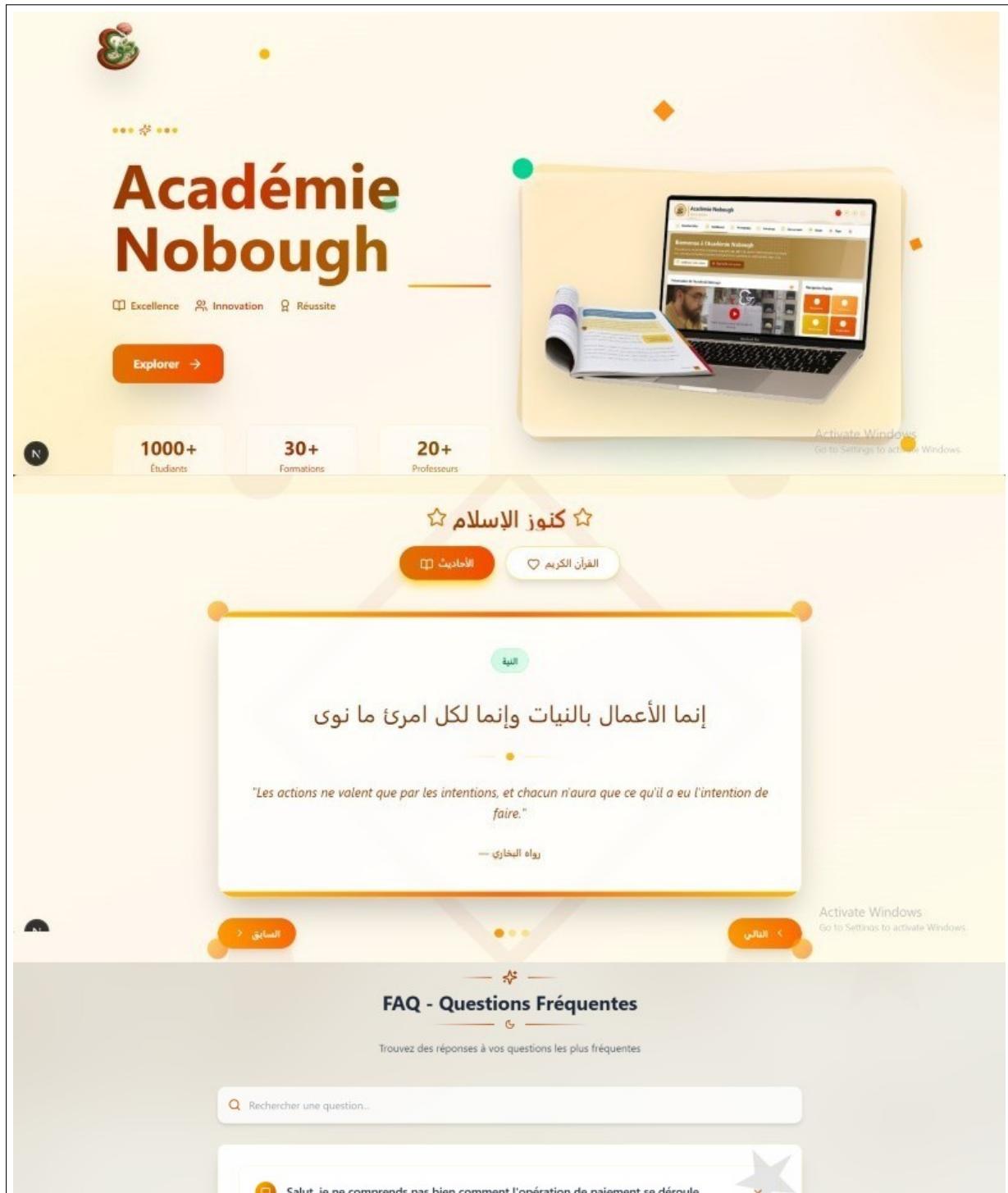


FIGURE 4.2 – Page d'accueil

2 Première visite du site

2.1 Navigation sans authentification

L'utilisateur peut naviguer librement sur le site sans se connecter. Cependant, certaines fonctionnalités comme l'accès aux pages "/Payer" et "/Room" sont restreintes.

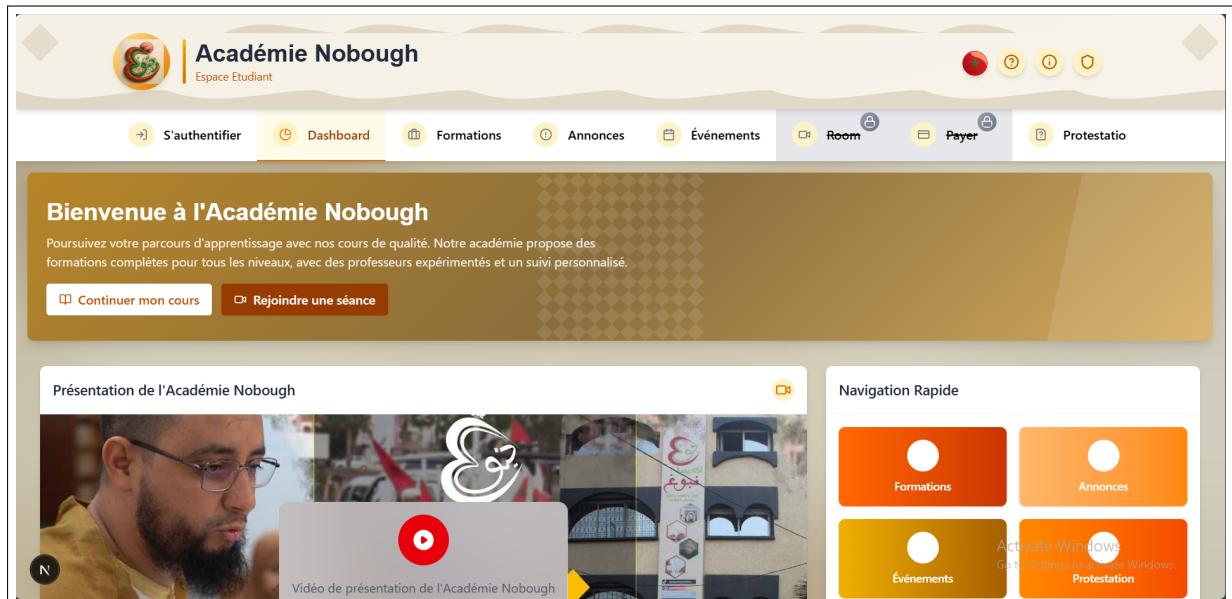


FIGURE 4.3 – Accès restreint

2.2 Rejoindre une formation

Les utilisateurs non authentifiés ne peuvent pas rejoindre une formation, ce qui garantit que seuls les étudiants inscrits peuvent accéder aux ressources pédagogiques.

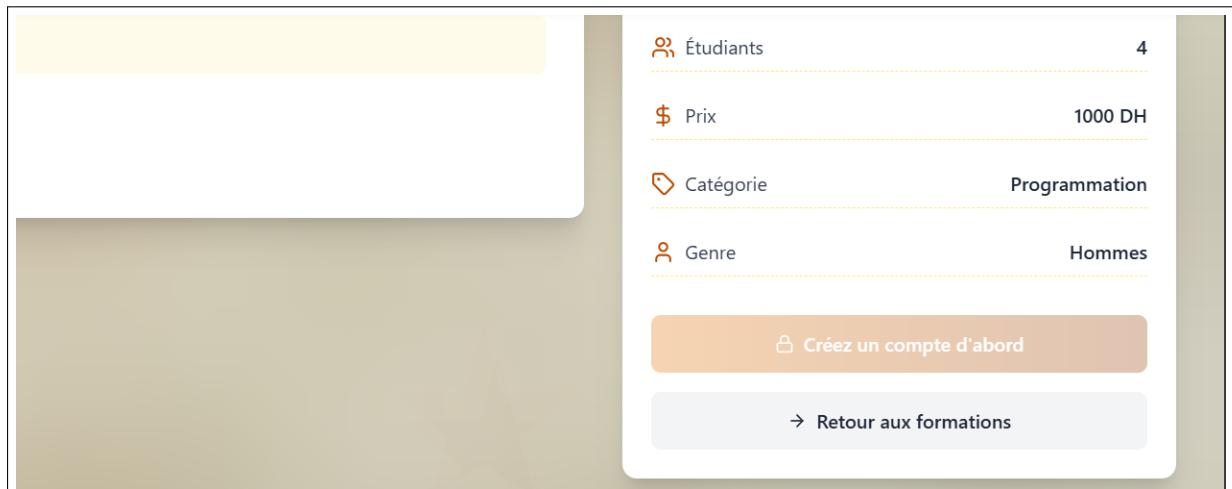


FIGURE 4.4 – Rejoindre une formation

2.3 Tentative de connexion

Lorsqu'un utilisateur tente de se connecter à la plateforme en saisissant des identifiants qui ne correspondent à aucun compte existant dans la base de données, un message d'erreur clair et explicite lui est affiché, l'informant que le compte n'existe pas. Ce message a pour but de lui indiquer que soit l'adresse e-mail saisie est incorrecte, soit qu'il n'a pas encore créé de compte, et il est invité, le cas échéant, à procéder à l'inscription.

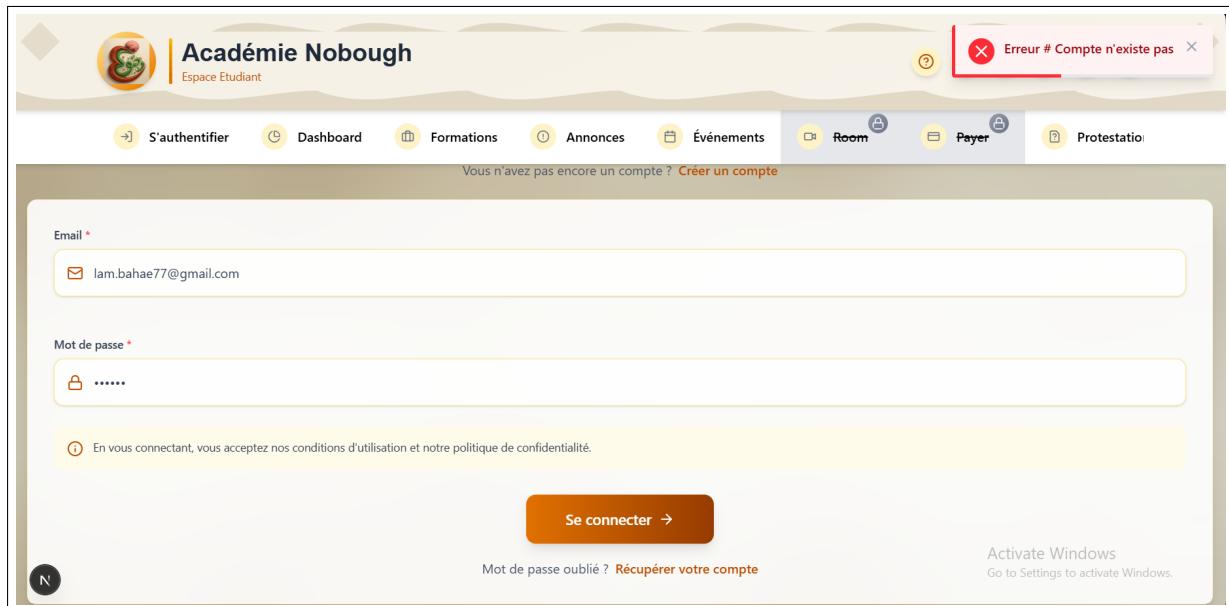


FIGURE 4.5 – Login : compte n'existe pas

2.4 Inscription

L'utilisateur peut s'inscrire en fournissant les informations nécessaires.

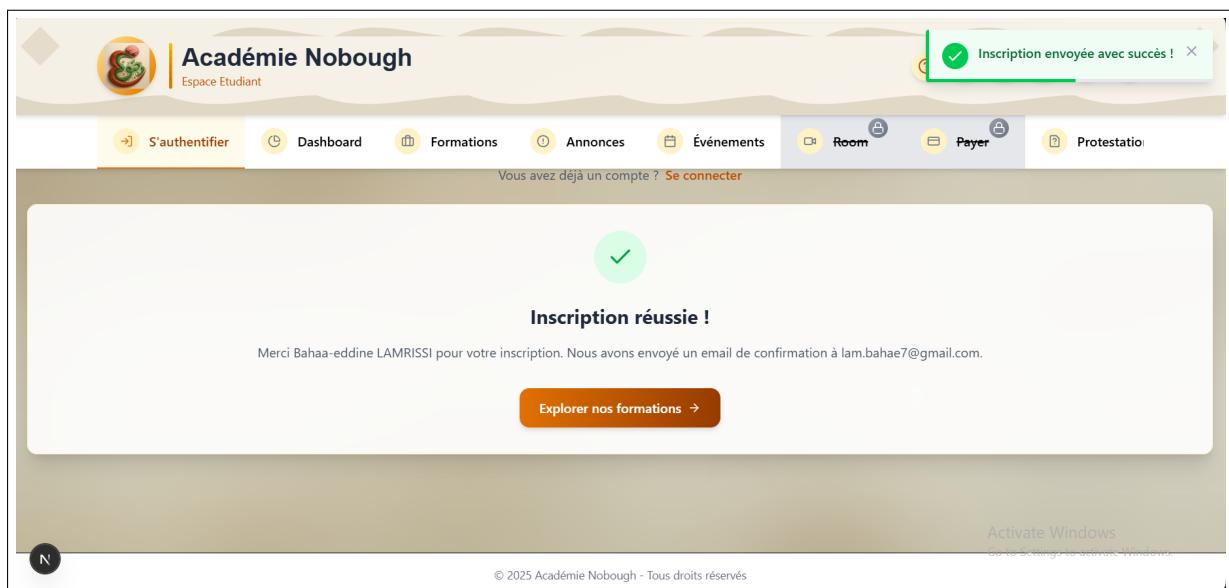


FIGURE 4.6 – Registration

2.5 Tentative de connexion

Si l'utilisateur essaie de faire login sur place (avant que son compte est validé), il aura une erreur.



FIGURE 4.7 – Registration

2.6 Validation par la secrétaire

Une fois la demande d'inscription soumise, elle est examinée par la secrétaire. Celle-ci dispose de la possibilité d'accepter ou de refuser la demande en fonction des informations fournies, des pièces justificatives jointes, et des critères d'admission définis par l'établissement. En cas de refus, un motif peut être communiqué à l'utilisateur afin de l'informer des raisons de la décision.



FIGURE 4.8 – Secrétaire refuse

Si la secrétaire décide de refuser la demande d'inscription, un message est automatiquement envoyé à l'utilisateur sur sa boîte mail. Ce message l'informe que son inscription a été rejetée et précise clairement les raisons de ce refus, afin de lui permettre, le cas échéant, de corriger les informations ou de fournir les données manquantes avant de soumettre une nouvelle demande.

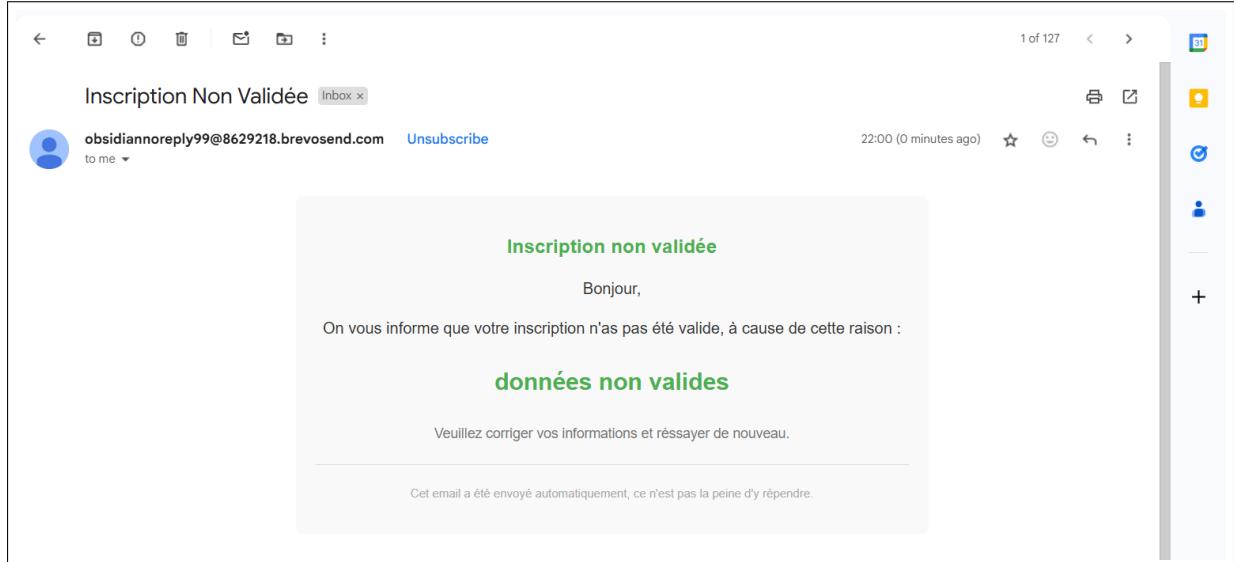


FIGURE 4.9 – Email de refus

3 Processus de paiement et de validation

3.1 Passage au paiement

Si la secrétaire valide la demande, l'utilisateur reçoit un lien de paiement, où il pose une image de preuve de transaction banquaire. Après paiement, il doit attendre la validation de la secrétaire avant de pouvoir accéder à son compte.

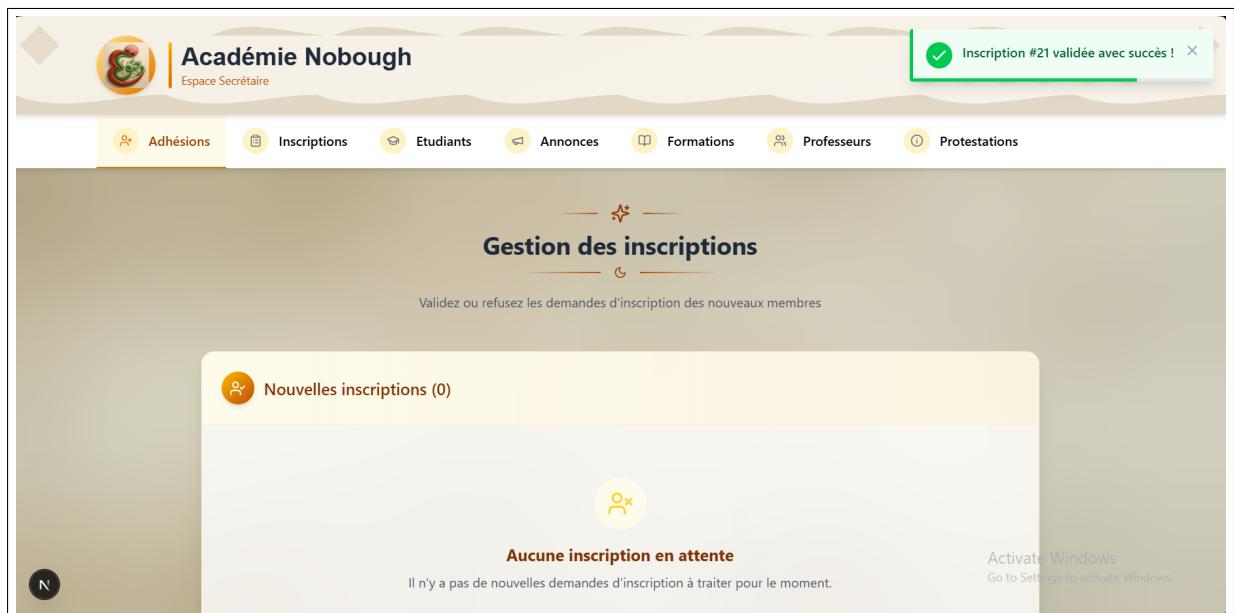


FIGURE 4.10 – Secrétaire valide

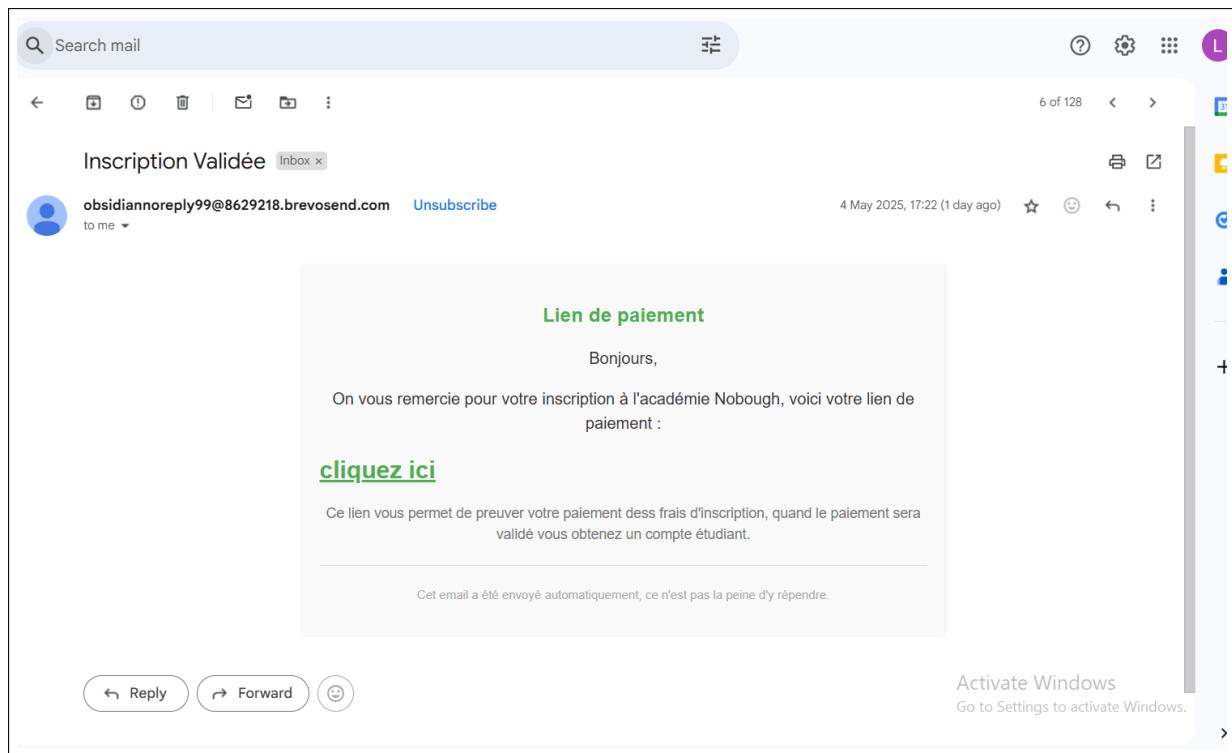


FIGURE 4.11 – Lien de paiement

Le lien envoyé à l'utilisateur par e-mail lui permet d'accéder à un espace sécurisé dans lequel il peut déposer une preuve de transaction (reçu, capture d'écran, ou tout autre justificatif) afin de confirmer son paiement. Cette étape est essentielle pour valider son inscription et permettre l'activation de son compte.

A screenshot of a payment submission form titled "Effectuer un paiement". The form is set against a light beige background with decorative horizontal bars above and below the title. It includes fields for "Montant" (amount) containing "200", "Message" (message) containing "merci", and a "Preuve de transaction" (transaction proof) section with a file input field showing "Choose File Untitled (2).png". A note at the bottom of the form states: "En soumettant ce formulaire, vous acceptez nos conditions d'utilisation et notre politique de confidentialité. Vos données personnelles seront traitées conformément à la loi en vigueur." To the right of the form, there is a "Activate Windows" message. At the bottom right of the form area, there is a large orange button labeled "Soumettre le paiement →".

FIGURE 4.12 – Secrétaire valide

3.2 Tentative de connexion

Si l'utilisateur tente de se connecter avant que son paiement n'ait été validé, le système lui renverra un message d'erreur précisant que l'accès à son compte est temporairement restreint. Ce message l'informera que la validation du paiement est une étape préalable indispensable à l'activation complète de son compte, et qu'il devra attendre la confirmation du règlement avant de pouvoir se connecter.

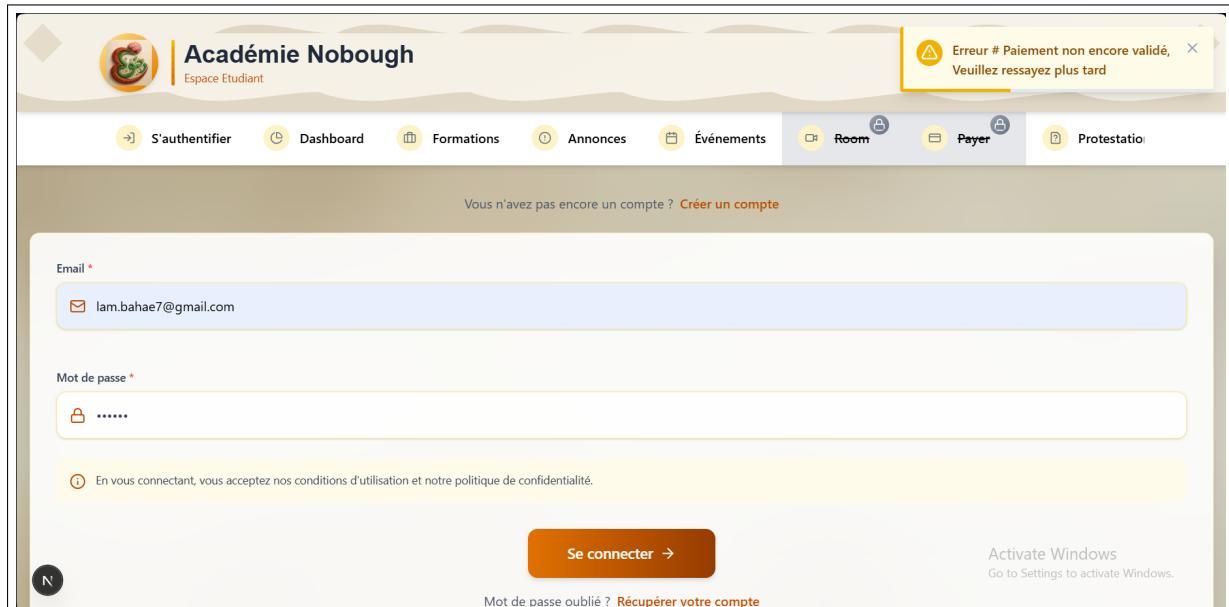


FIGURE 4.13 – Registration

3.3 Validation du paiement

Le paiement effectué sera ensuite examiné par le service administratif ou la secrétaire, qui valide ou rejette le paiement selon sa conformité.

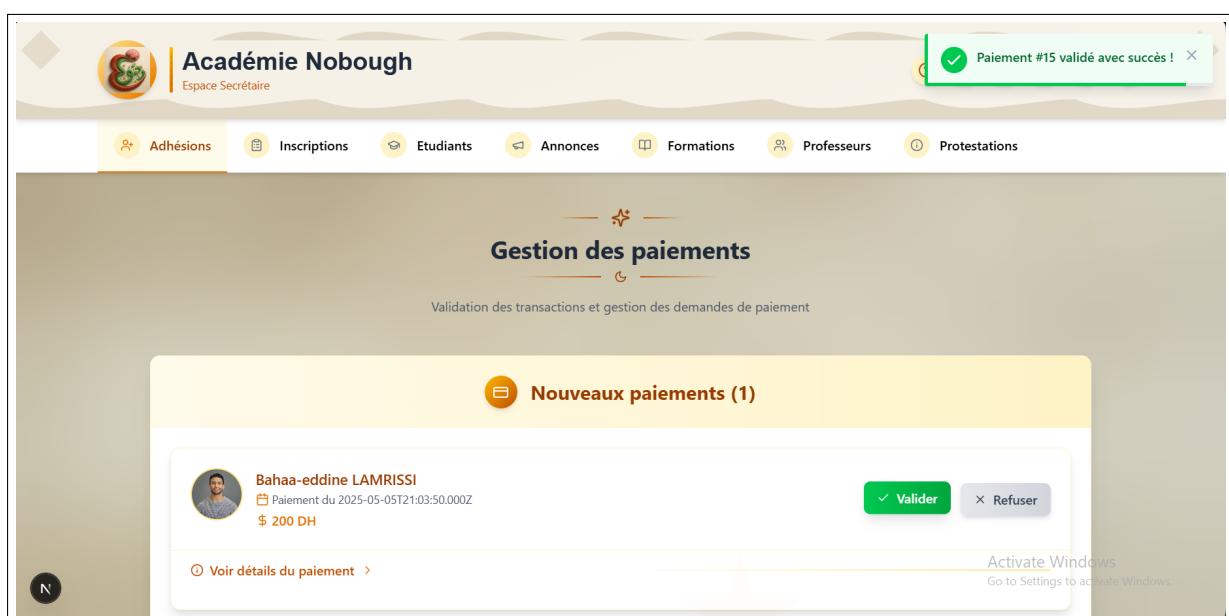


FIGURE 4.14 – Lien de paiement

En cas de validation, le compte de l'étudiant est activé et il reçoit un e-mail de confirmation lui donnant accès à l'espace étudiant. En cas de rejet, l'étudiant est notifié par mail avec les raisons du refus et les éventuelles démarches à suivre pour régulariser la situation.

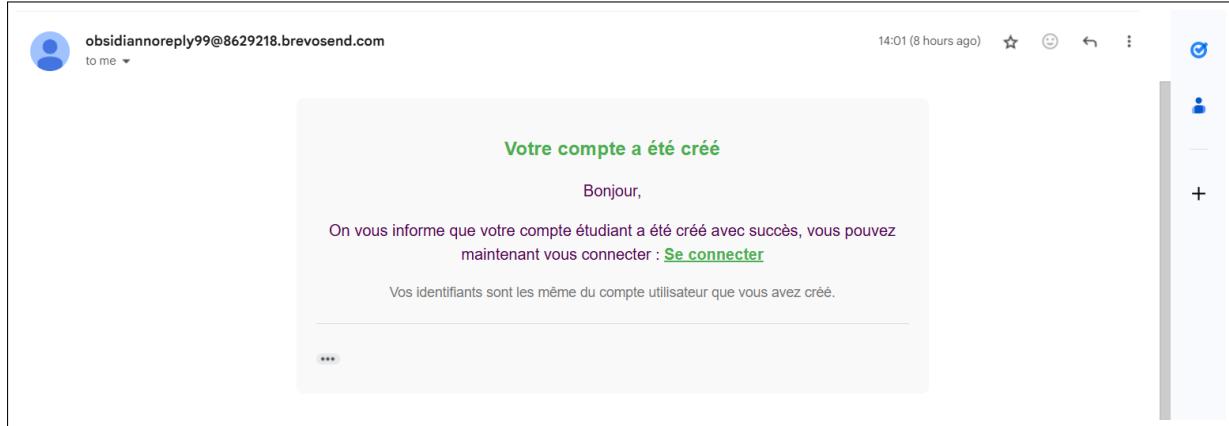


FIGURE 4.15 – Lien de paiement

3.4 Connexion après validation

Une fois le paiement validé, l'utilisateur peut se connecter. Si le mot de passe est incorrect, il reçoit un message d'erreur. En cas de succès, il est redirigé vers la page principale et une session est démarrée.

La session est stockée en côté serveur pour 7 jours et elle n'apparaît pas en côté client (dans le navigateur), ce qui renforce la sécurité.

La session contient l'identifiant, le nom, le prénom, l'email, le téléphone, le cin, la biographie, le rabais et la date d'inscription.

L'image ne sera pas stocké dans la session pour ne pas ralentir la performance de l'application.

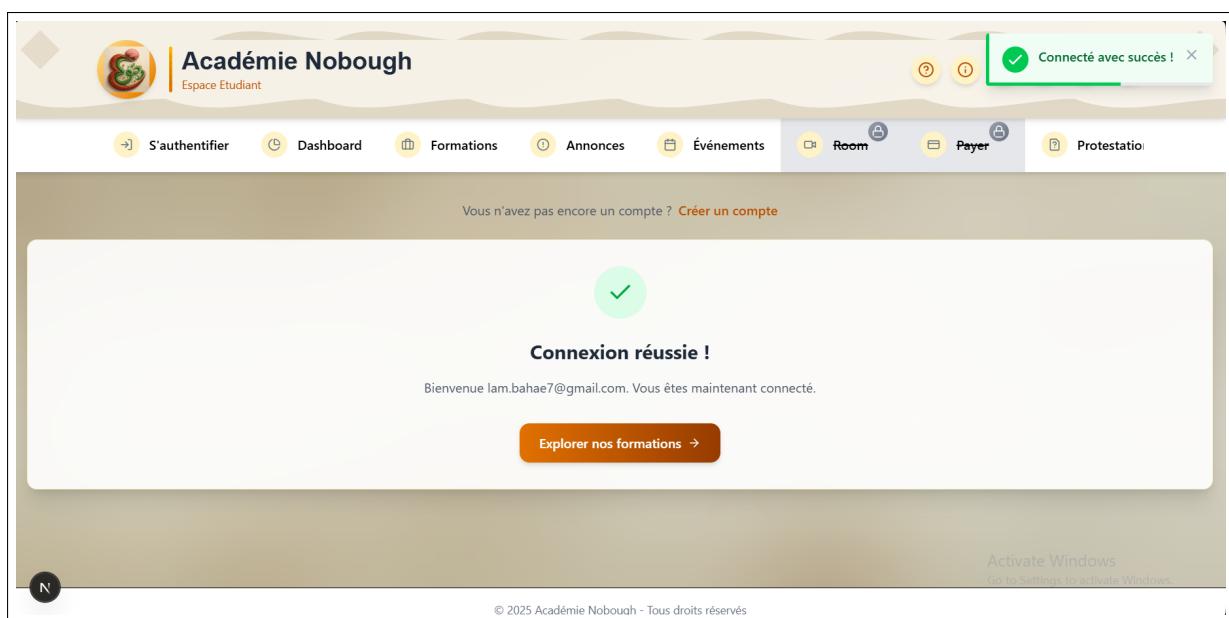


FIGURE 4.16 – Login réussi

4 Gestion de la session et récupération de mot de passe

4.1 Déconnexion

L'utilisateur peut se déconnecter à tout moment.

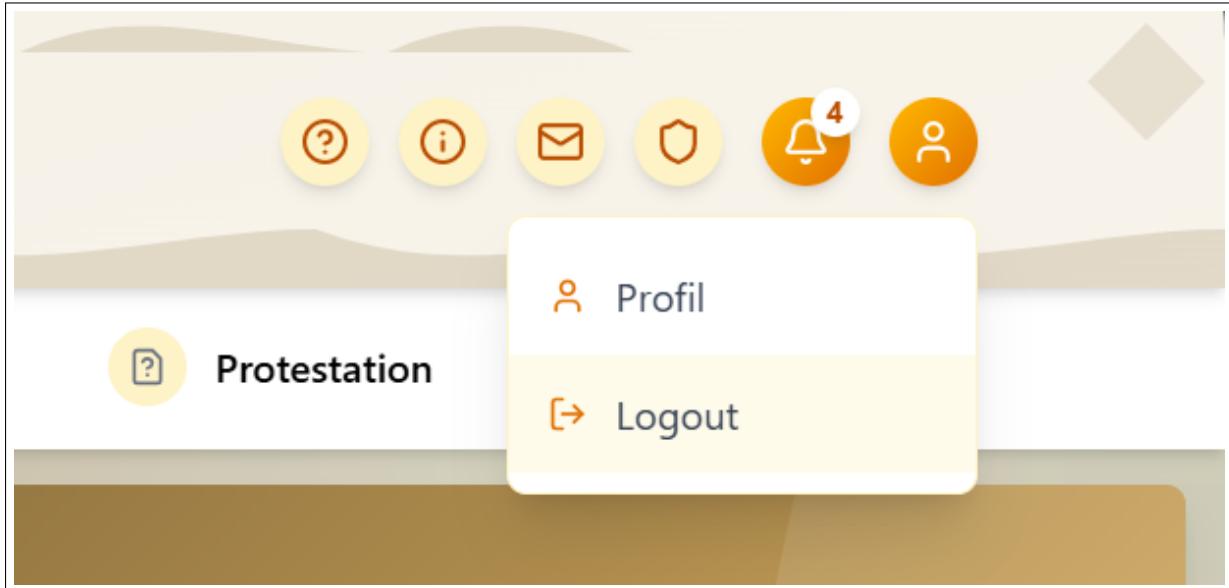


FIGURE 4.17 – Logout

4.2 Récupération de mot de passe

En cas d'oubli de mot de passe, l'étudiant peut demander un code de réinitialisation par email.

Si l'email est invalide ou inexistant une erreur s'affiche, sinon il sera invité à entrer le code qui vient d'être envoyé à sa boîte mail, ce code est valable seulement pour cinq minutes pour assurer plus de sécurité.



FIGURE 4.18 – Mot de passe oublié

4.3 Code envoyé par email

Pour réinitialiser son mot de passe, un code de vérification est envoyé à l'adresse e-mail associée au compte de l'utilisateur. Ce code est valable uniquement pendant une durée de cinq minutes, afin de renforcer la sécurité du processus et de prévenir toute tentative d'accès non autorisé.

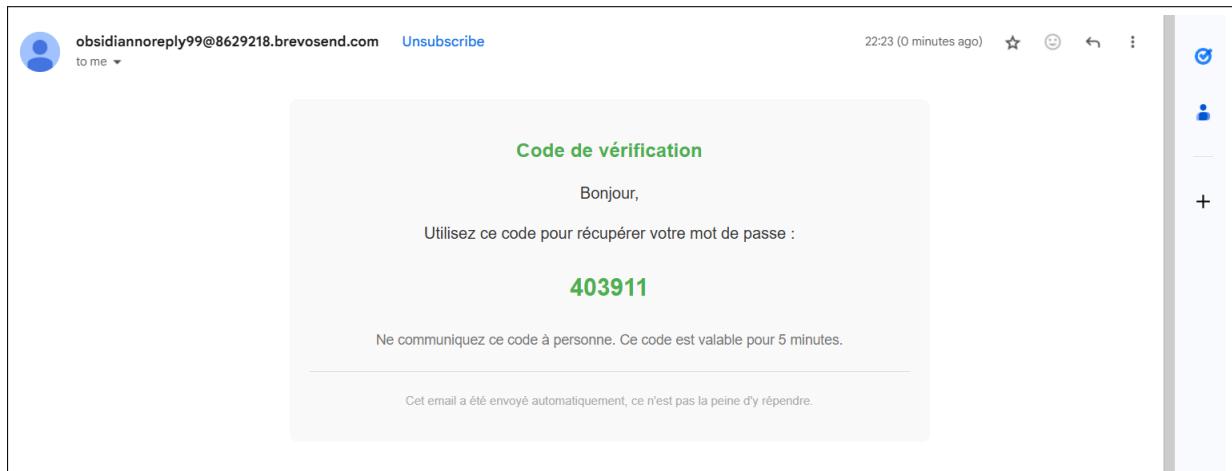


FIGURE 4.19 – Mot de passe oublié

Si le code saisi est correct et toujours valide, l'utilisateur est autorisé à poursuivre la procédure et peut finalement modifier son mot de passe en toute sécurité.

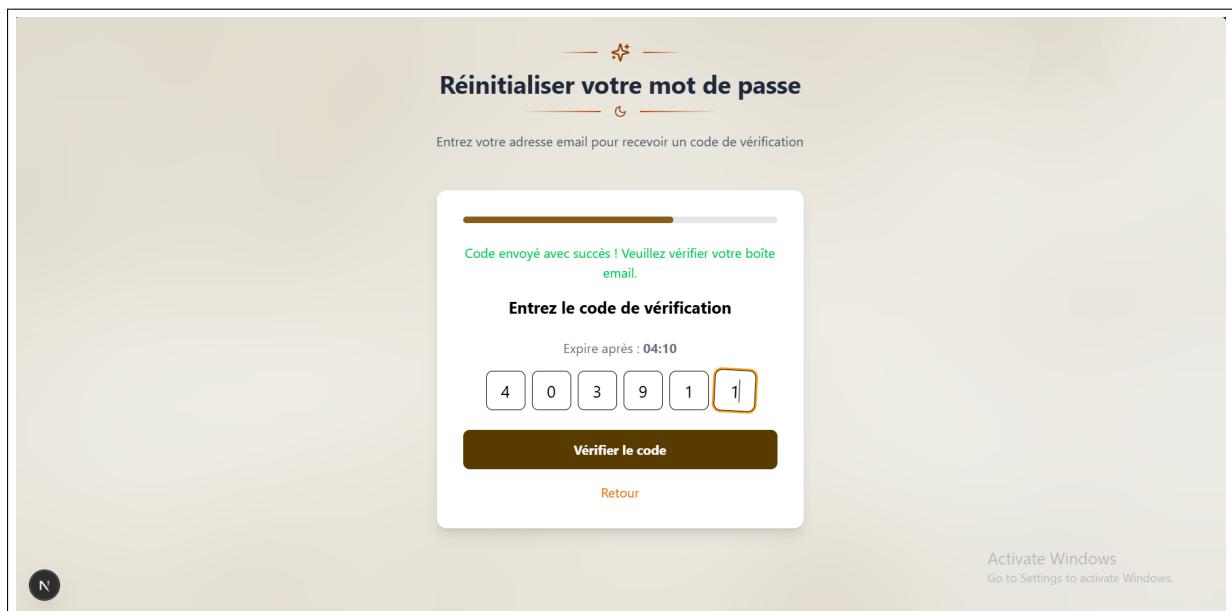


FIGURE 4.20 – Mot de passe oublié

4.4 Réinitialisation du mot de passe

Après avoir reçu le code par email, l'utilisateur peut entrer un nouveau mot de passe.



FIGURE 4.21 – Mise à jour du mot de passe

5 Rejoindre une formation après login

Les étudiants peuvent maintenant demander à rejoindre une formation. La secrétaire reçoit une notification et peut accepter ou refuser la demande. L'étudiant reçoit un email avec la décision.

Il ne peut pas rejoindre sans session ou si il a déjà envoyé une demande d'inscription, une seule demande peut être effectuée.

FIGURE 4.22 – Processus de demande pour rejoindre une formation

6 Validation ou refus par la secrétaire

Une fois l'inscription à la formation soumise par l'utilisateur, celle-ci est transmise à la secrétaire pour vérification. La secrétaire a alors la possibilité soit de valider l'inscription, soit de la refuser en fonction des critères établis. En cas de validation, un e-mail est automatiquement envoyé à l'utilisateur contenant un lien de paiement pour le premier mois. Le processus qui suit est identique à celui mis en place lors de l'inscription initiale, incluant la validation du paiement et l'activation du compte.

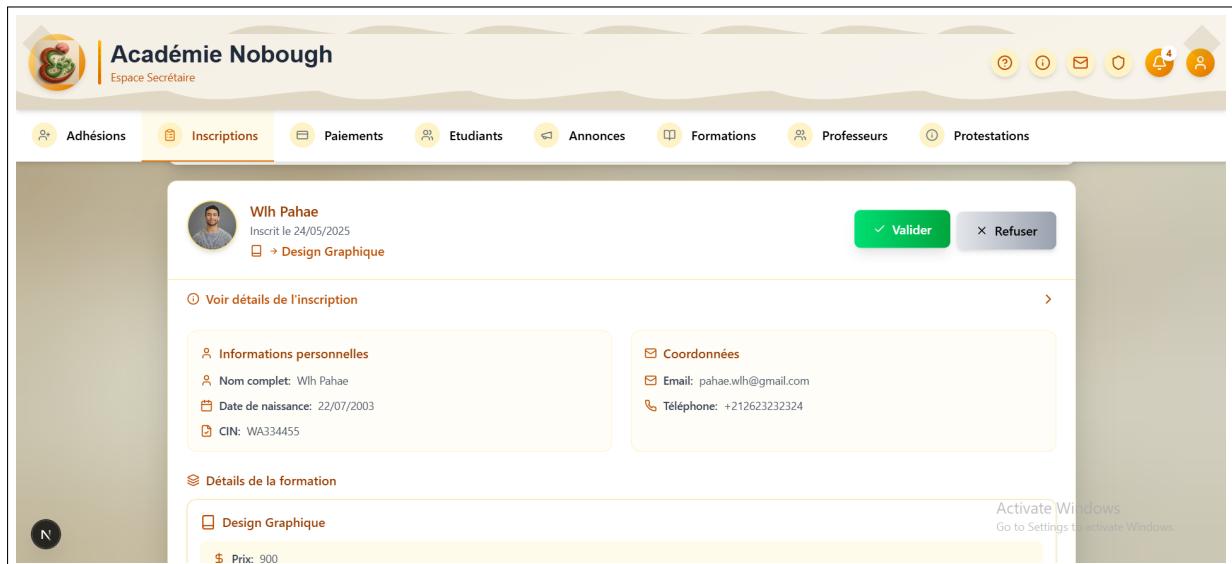


FIGURE 4.23 – Processus de demande pour rejoindre une formation

7 Consultation du paiement

Les étudiants ont une page où ils peuvent voir leurs paiements mensuels et effectuer des paiements en ligne. Cette page affiche les détails des paiements et permet aux étudiants de payer en ligne.

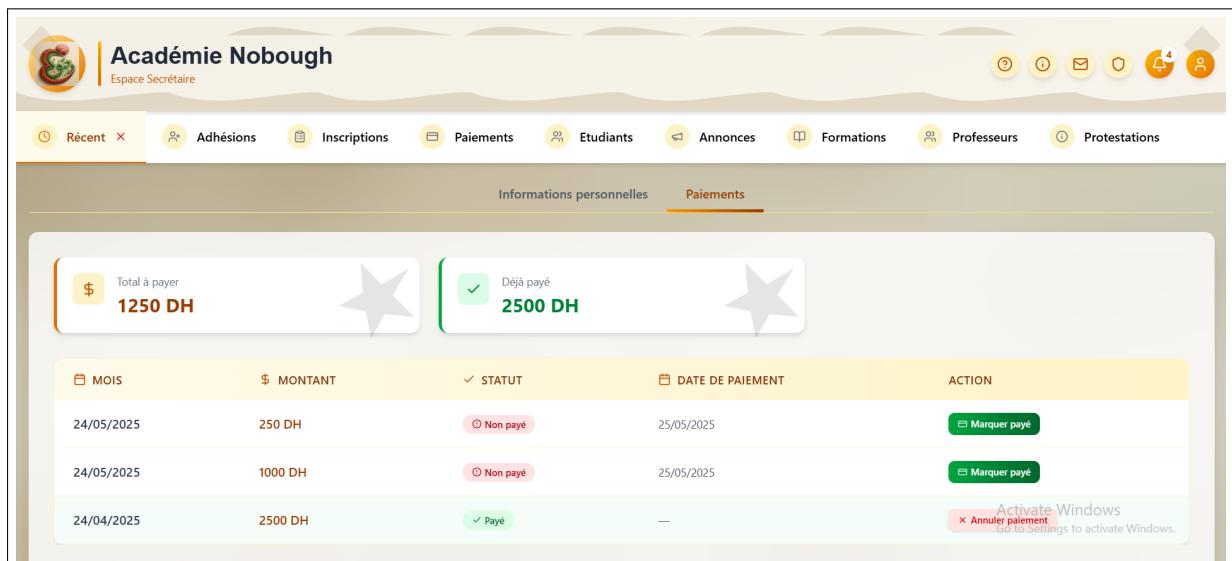


FIGURE 4.24 – Consultation de paiement par l'étudiant

8 Gestion des paiements

La secrétaire dispose d'une liste affichant l'ensemble des étudiants, accompagnés du nombre de mois non payés. Pour faciliter la lecture et le suivi, un système de couleurs est utilisé : les étudiants à jour dans leurs paiements apparaissent en vert, ceux ayant 1 mois de retard en jaune, ceux en retard de 2 mois en orange, et ceux ayant plus de 3 mois de retard en rouge. Ce code couleur permet à la secrétaire d'identifier rapidement les situations nécessitant une relance ou un suivi administratif.

The screenshot shows a list of students with their payment status. The columns include: PHOTO, NOM & PRÉNOM, EMAIL, TÉLÉPHONE, CIN, MOIS NON PAYÉS, DÉTAILS, and SUPPRIMER. The 'MOIS NON PAYÉS' column uses color coding: green for 'A jour', yellow for '1 mois', orange for '2 mois', and red for '3 mois'. A tooltip 'Activate Windows Go to Settings to activate Windows.' is visible on the right side of the screen.

PHOTO	NOM & PRÉNOM	EMAIL	TÉLÉPHONE	CIN	MOIS NON PAYÉS	DÉTAILS	SUPPRIMER
	Youssef Benali Rabais: 0%	youssef.benali@gmail.com	+21262222222	EF345678	1 mois		
	Omar Lahiou Rabais: 5%	omar.lahiou@gmail.com	+21264444444	IJ567890	1 mois		
	Karim Moussaoui Rabais: 15%	karim.moussaoui@gmail.com	+21263333333	GH901234	A jour		
	Wih Pahae Rabais: 10%	pahae.wlh@gmail.com	+212623232324	WA334455	2 mois		
	Hamza Tazi Rabais: 0%	hamza.tazi@gmail.com	+21265555555	KL123456	A jour		

FIGURE 4.25 – Gestion des paiements

En complément de la vue générale, la secrétaire peut accéder à une fiche détaillée pour chaque étudiant, lui permettant de consulter la liste complète des mois payés et non payés. Depuis cette interface, elle a la possibilité de cocher manuellement un mois comme payé en cas de régularisation hors système, ou d'annuler un paiement précédemment enregistré en cas d'erreur ou de remboursement. Cette fonctionnalité offre une flexibilité de gestion tout en garantissant un suivi précis et à jour des paiements.

The screenshot shows a detailed view of a student's payments. It includes sections for 'Informations personnelles' and 'Paiements'. The 'Paiements' section displays two boxes: one for 'Total à payer 1250 DH' and another for 'Déjà payé 2500 DH'. Below this, a table lists payment details: MOIS, MONTANT, STATUT, DATE DE PAIEMENT, and ACTION. The first two rows have 'Non payé' status and 'Marquer payé' action, while the third row has 'Payé' status and 'Annuler paiement' action. A tooltip 'Activate Windows Go to Settings to activate Windows.' is visible on the right side.

MOIS	MONTANT	STATUT	DATE DE PAIEMENT	ACTION
24/05/2025	250 DH	Non payé	25/05/2025	
24/05/2025	1000 DH	Non payé	25/05/2025	
24/04/2025	2500 DH	Payé	—	

FIGURE 4.26 – Gestion des paiements en détails

9 Annonces

La secrétaire dispose d'un espace dédié sur le site lui permettant de publier des annonces à destination des étudiants. Elle peut y saisir un titre, un contenu détaillé, éventuellement y ajouter une image. Une fois l'annonce créée, elle sera automatiquement visible dans la section des annonces du site, garantissant une communication rapide et centralisée avec l'ensemble des étudiants.

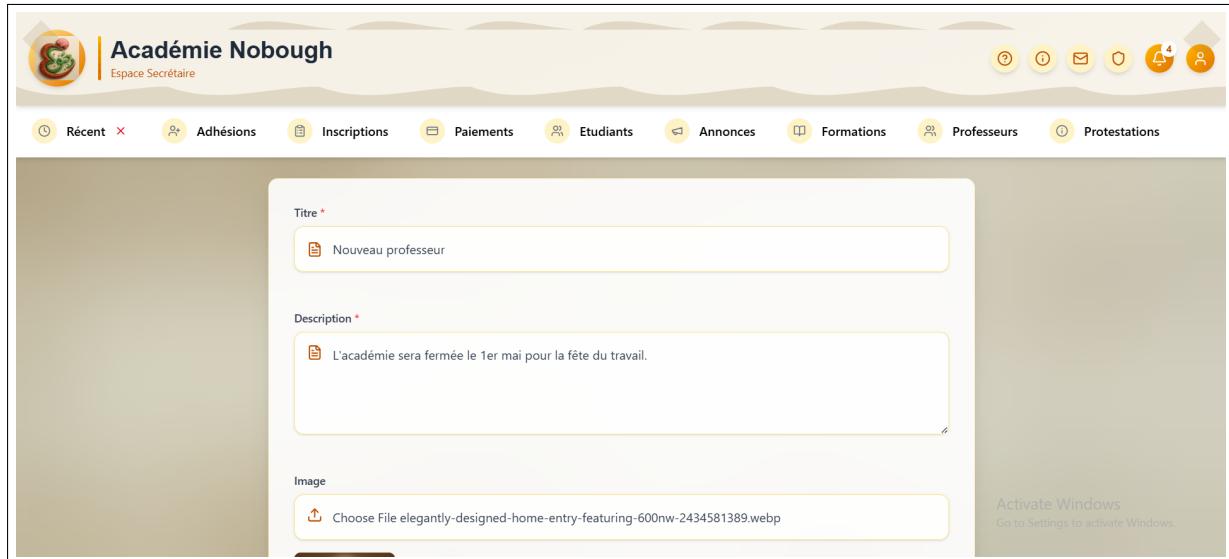


FIGURE 4.27 – Ajouter annonce

Les étudiants ont accès à une section dédiée sur le site où ils peuvent consulter l'ensemble des annonces publiées par la secrétaire. Chaque annonce y est affichée avec son titre, sa date de publication, et son contenu détaillé.

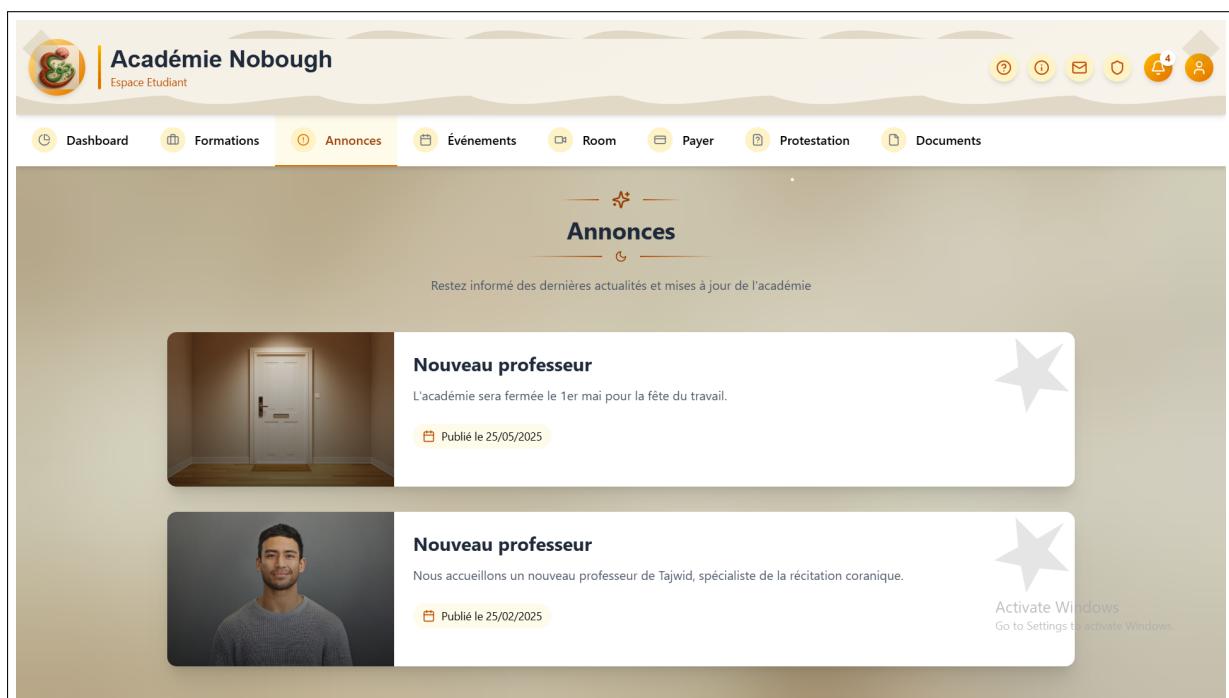


FIGURE 4.28 – Consulter les annonces

10 Protestations

Un utilisateur ou un étudiant a la possibilité de déposer une protestation directement sur le site via un espace dédié. Deux options s'offrent à lui : – S'il sélectionne une question prédefinie dans la liste proposée (ex. : 'Pourquoi mon paiement n'est-il pas validé ?', 'Je ne vois pas mon cours dans l'espace étudiant'), une réponse automatique et immédiate lui est affichée à l'écran. – En revanche, s'il rédige un message personnalisé, sa demande sera automatiquement transmise à la secrétaire, qui pourra l'examiner et y répondre manuellement dans les plus brefs délais.

The screenshot shows the student dashboard of the Académie Nobough. At the top, there is a navigation bar with icons for Dashboard, Formations, Annonces, Événements, Room, Payer, Protestation (which is highlighted in orange), and Documents. Below the navigation bar, a large central box is titled 'Sélectionnez le type de protestation' (Select protest type). It contains fields for Nom (Nom: pahae, Prénom: lamrissi), Email (Email: lam.bahae7@gmail.com), and Téléphone (Phone: +212611111111). A 'Message' field contains the text: 'Salut, je ne comprends pas bien comment l'opération de paiement se déroule.' At the bottom right of this box is an 'Envoyer' (Send) button.

FIGURE 4.29 – Déposer une protestation

La secrétaire peut consulter depuis son espace d'administration l'ensemble des protestations personnalisées envoyées par les utilisateurs ou étudiants. Pour chaque protestation, elle a la possibilité de répondre directement via l'interface. Une fois sa réponse rédigée et validée, celle-ci est automatiquement envoyée par e-mail à l'utilisateur concerné, lui permettant de recevoir un retour clair et rapide à sa demande.

The screenshot shows the administrator dashboard of the Académie Nobough. At the top, there is a navigation bar with icons for Récent, Adhésions, Inscriptions, Paiements, Etudiants, Annonces, Formations, Professeurs, and Protestations (which is highlighted in orange). Below the navigation bar, a section titled 'Protestations des étudiants / utilisateurs' displays a protest from user 'lamrissi pahae'. The protest message is: 'Salut, je ne comprends pas bien comment l'opération de paiement se déroule.' Below the message, it says 'Reçue le 2025-05-25T17:39:11.000Z' and the email address 'lam.bahae7@gmail.com'. A response message is being typed into a text area: 'D'accord, je vais vous envoyer un guide par email tout de suite.' At the bottom right of this area are 'Annuler' and 'Envoyer' buttons.

FIGURE 4.30 – Répondre à une protestation

En complément, la question et la réponse rédigée par la secrétaire sont automatiquement ajoutées à la section FAQ (Foire Aux Questions) du site. Cette base de connaissances est accessible à tous les utilisateurs et permet de consulter les réponses aux questions déjà posées, favorisant ainsi une réduction du nombre de protestations similaires et une meilleure autonomie des utilisateurs dans la recherche d'informations.



FIGURE 4.31 – les FAQs

11 Consultation du programme d'une formation

Le professeur ainsi que les étudiants inscrits peuvent accéder à l'espace dédié à leur formation via le site. Cet espace offre plusieurs fonctionnalités clés : – la consultation du programme de la formation, – l'accès à l'historique des messages échangés dans le chat, – une bibliothèque numérique contenant des ressources pédagogiques (PDF, vidéos, liens utiles), – et la liste des autres membres inscrits à la même formation, facilitant ainsi les échanges, la collaboration et le suivi pédagogique.

11.1 Programme de la formation

Le professeur ainsi que ses étudiants ont accès à un espace dédié à leur formation, dans lequel ils peuvent consulter le programme détaillé, incluant les horaires des séances. Cet espace favorise une vision claire du déroulement de la formation et assure un meilleur suivi de l'apprentissage.



FIGURE 4.32 – Programme d'une formation

11.2 Chat en temps réel

Ils ont également la possibilité de participer à un chat en temps réel au sein de l'espace de formation. Ce chat permet au professeur de partager des informations importantes, d'envoyer des annonces ou des précisions, tandis que les étudiants peuvent poser des questions, échanger entre eux et discuter directement avec le professeur, favorisant ainsi une communication interactive et instantanée.



FIGURE 4.33 – Chat en temps réel

11.3 Bibliothèque numérique

Dans cet espace de formation, une section bibliothèque numérique est mise à disposition. Le professeur peut y déposer des fichiers de tout type (documents PDF, présentations, vidéos, etc.), tandis que les étudiants ont la possibilité de consulter et télécharger ces ressources librement. Cette fonctionnalité facilite l'accès aux supports pédagogiques nécessaires au bon déroulement de la formation.

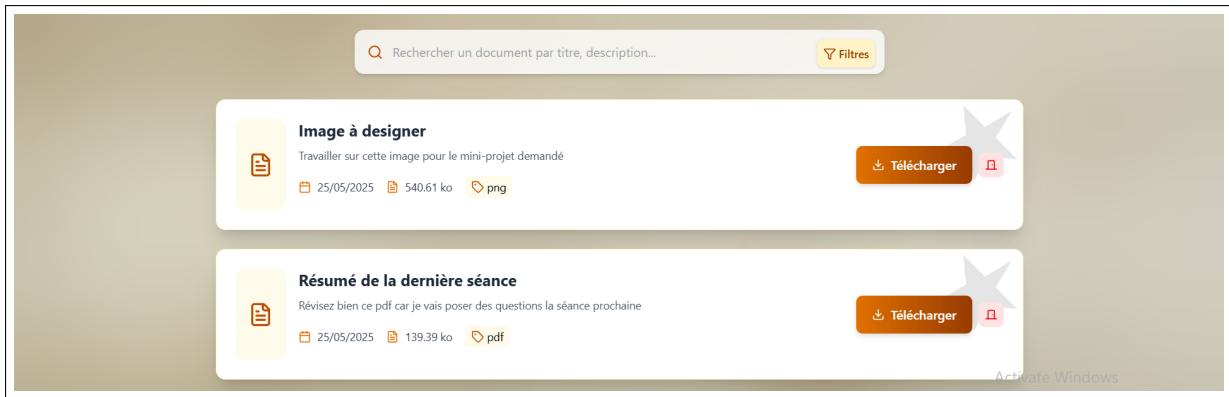


FIGURE 4.34 – Bibliothèque numérique

11.4 Autres participants

Au sein de l'espace de formation, le professeur ainsi que les étudiants ont la possibilité d'accéder à une liste complète des membres inscrits à la même session. Cette fonctionnalité leur permet non seulement de connaître l'identité des autres participants, mais aussi de faciliter la communication, la collaboration et le travail en groupe tout au long de la formation. Cette transparence favorise un esprit de communauté et d'entraide entre les apprenants. Par ailleurs, en matière de certification, seule la responsabilité du professeur est reconnue pour générer et délivrer un certificat officiel à un étudiant. Ce certificat atteste formellement de la réussite de la formation, de la participation active de l'étudiant, ainsi que de la validation des compétences acquises. La génération de ce document est une étape clé dans le parcours pédagogique, garantissant la reconnaissance officielle du travail accompli.

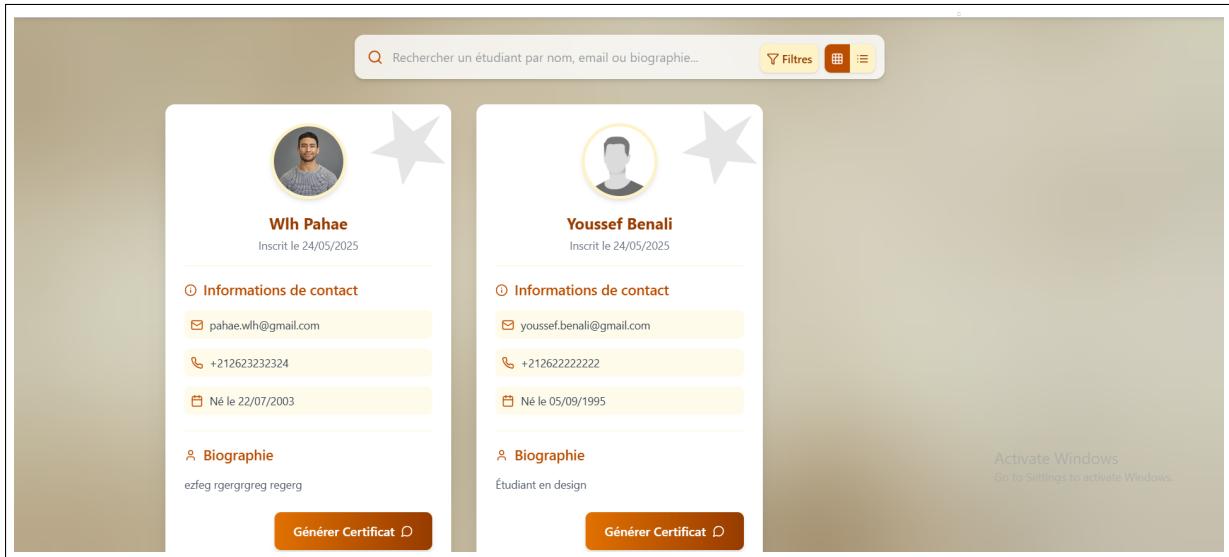


FIGURE 4.35 – Autres participants à la formation

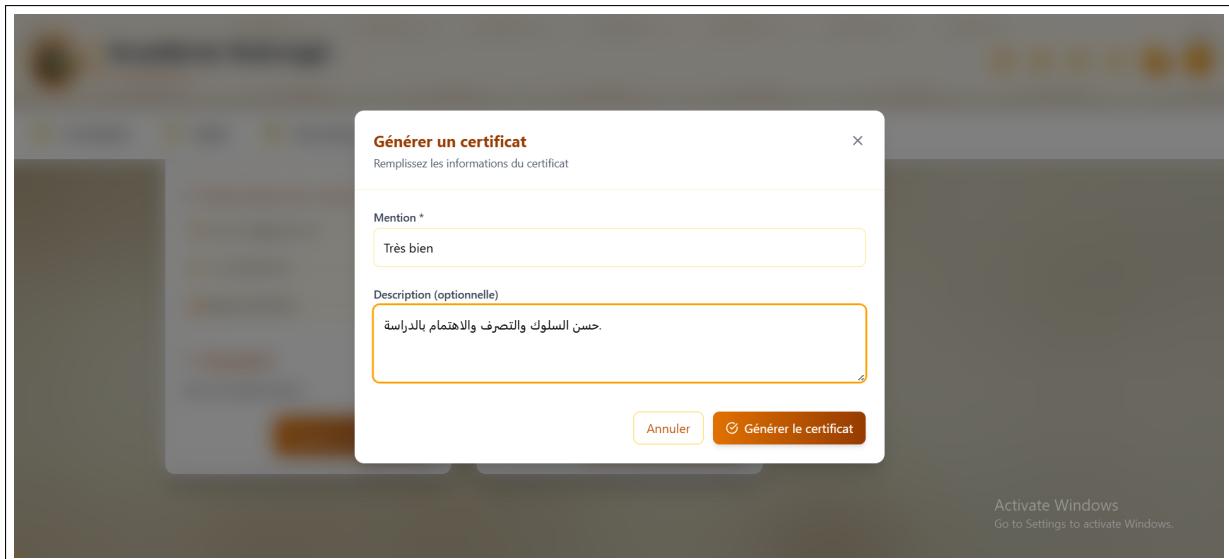


FIGURE 4.36 – Générer un certificat pour un étudiant

11.5 Informations sur le professeur

Seuls les étudiants peuvent voir les informations de leurs professeur.

FIGURE 4.37 – Informations du professeur

12 Séance en ligne

Dans notre application, il y a une possibilité de faire des séances en ligne entre le professeur et les étudiants, pour ne pas aller loin et se baser sur une autre application externe comme Zoom ou Google Meet, etc. Notre but c'est d'avoir tous en une seule application

12.1 Créeer une séance en ligne

Lorsque le professeur clique sur le bouton 'Créer une séance', un code unique est automatiquement généré. Ce code peut ensuite être partagé avec les étudiants, par exemple via le chat intégré à la plateforme, afin qu'ils puissent facilement rejoindre la séance en cours. Ce système facilite la gestion des sessions en garantissant un accès sécurisé et rapide pour tous les participants.

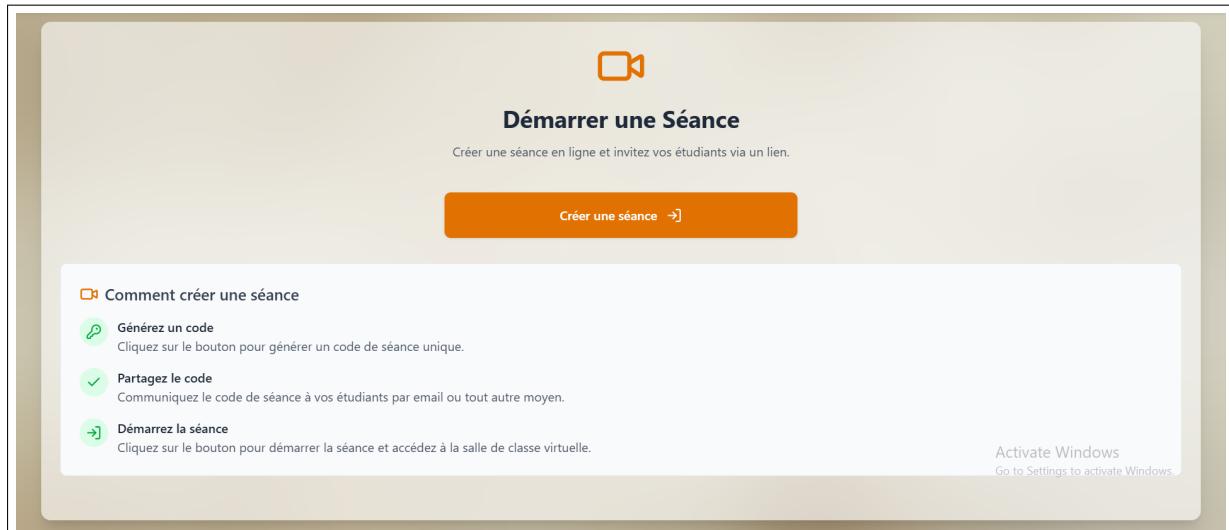


FIGURE 4.38 – Création d'une séance en ligne

12.2 Rejoindre la séance en ligne

Une fois l'étudiant a le code, il peut rejoindre la séance avec le professeur et les autres étudiants.

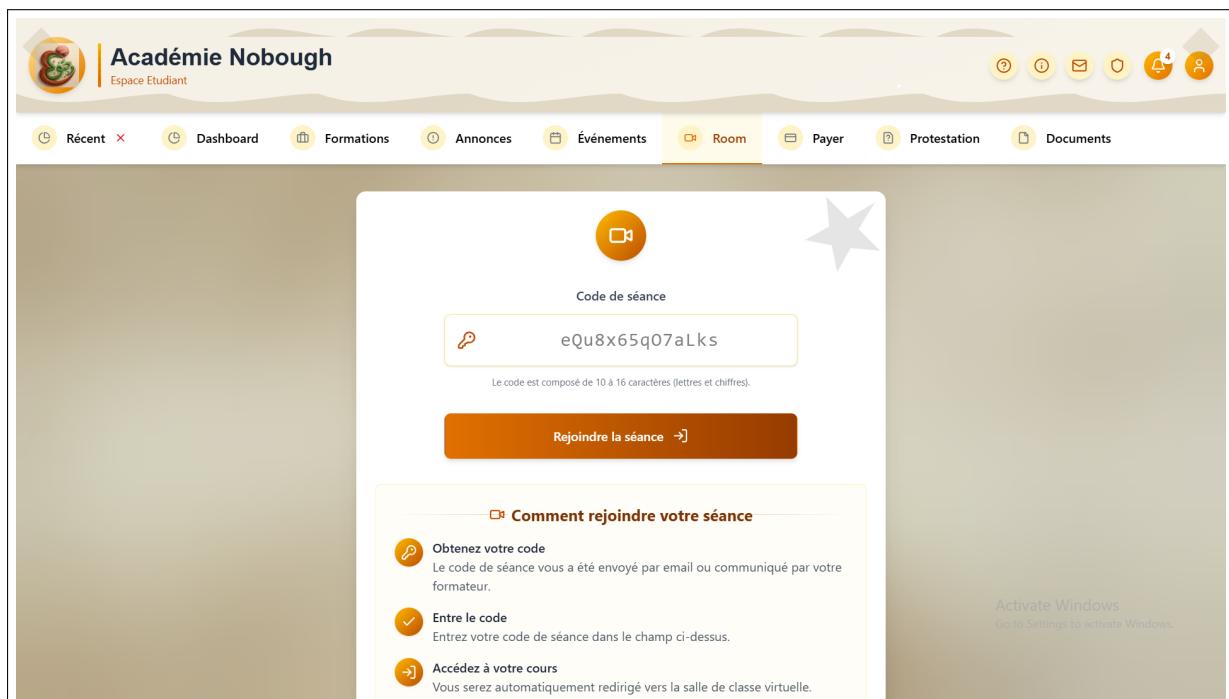


FIGURE 4.39 – rejoindre la séance en ligne

12.3 Déroulement de la séance en ligne

Durant la séance, plusieurs fonctionnalités sont disponibles pour les participants :
Étudiant :

- **Couper son micro** pour éviter les bruits parasites.
- **Lever la main virtuelle** pour demander la parole sans interrompre.
- **Se déconnecter** à tout moment pour quitter la session.

Professeur :

- **Couper son micro** afin d'améliorer la qualité sonore.
- **Partager son écran** pour diffuser des documents ou présentations.
- **Se déconnecter** pour terminer la séance ou interrompre l'accès.

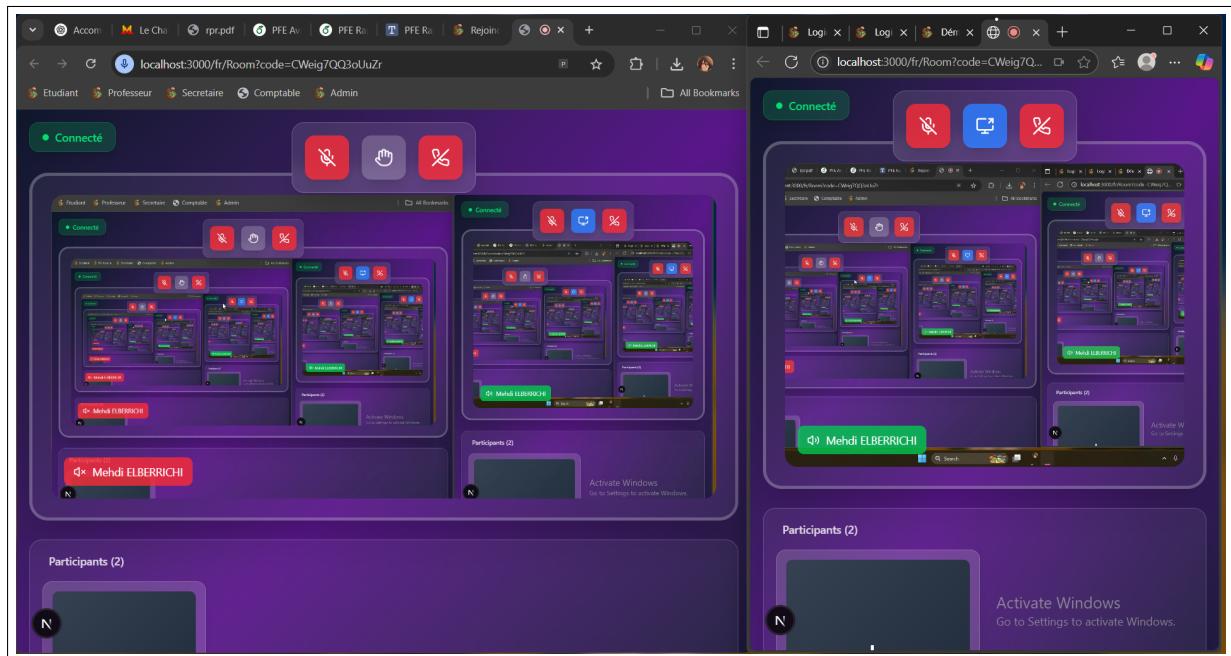


FIGURE 4.40 – Déroulement de la séance en ligne

13 Génération des documents

13.1 Côté étudiant

L'étudiant dispose d'un espace personnel où il peut télécharger plusieurs documents importants, notamment : son attestation d'inscription, son emploi du temps, l'historique de ses reçus de paiement, ainsi que les certificats des formations qu'il a déjà obtenus. Cette fonctionnalité lui permet de centraliser et d'accéder facilement à tous ses documents administratifs et pédagogiques essentiels.

The screenshot shows the 'Mes Documents' section of the Académie Nobough student dashboard. It includes:

- Documents Principaux:**
 - Attestation d'inscription (Document officiel d'inscription)
 - Emploi du temps (Planning des cours actuels)
- Reçus de Paiement:**
 - 2500 DH (24/04/2025)
- Certificats:**
 - Web Design (très bien, 24/05/2025)
 - Web Design (bien, 24/05/2025)

FIGURE 4.41 – Documents pour l'étudiant

Voici quelques exemples des documents qu'un étudiant peut télécharger depuis son espace personnel :



FIGURE 4.42 – Attestation d'inscription

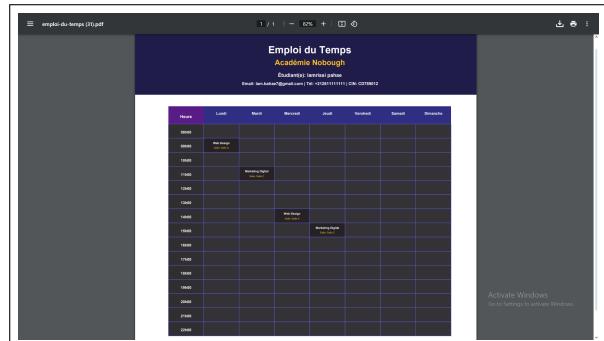


FIGURE 4.43 – Emploi du temps 1



FIGURE 4.44 – Reçu de paiement



FIGURE 4.45 – Certificat

13.2 Côté Professeur

Le professeur dispose également d'un espace personnel lui permettant d'effectuer plusieurs opérations importantes. Il peut notamment télécharger son attestation de travail officielle, un document qui certifie sa position au sein de l'établissement, ainsi que consulter ou télécharger son emploi du temps détaillé, mis à jour selon les modifications administratives ou pédagogiques. De plus, il a la possibilité de générer un certificat personnalisé pour un étudiant, comme une attestation de réussite, de présence ou tout autre document justificatif requis, facilitant ainsi les démarches administratives des apprenants.

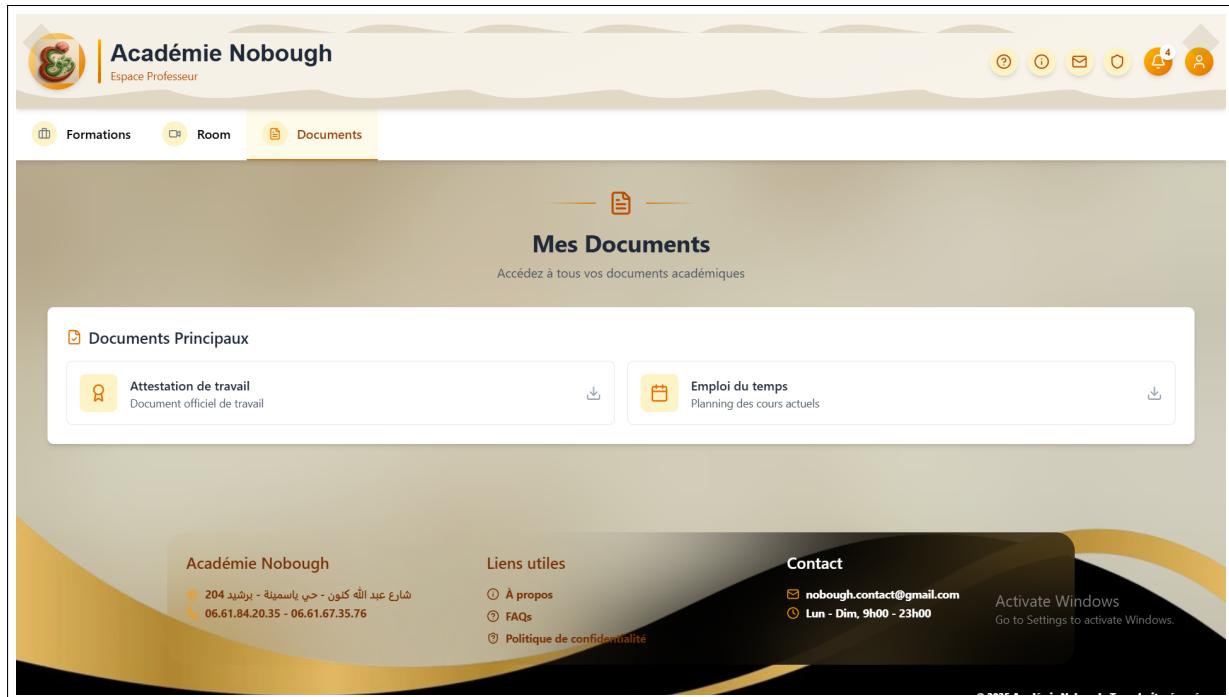


FIGURE 4.46 – Documents pour le professeur



FIGURE 4.47 – Attestation de travail

A screenshot of a PDF document titled "Emploi du Temps Académie Nobough". The header includes the Académie Nobough logo and Arabic text. The main content is a weekly employment schedule grid from Monday to Sunday. The grid shows various tasks and their times for each day. A digital signature of "Berriichi Mehdi" is at the bottom right.

FIGURE 4.48 – Emploi du temps 2

14 Pages de profil pour l'étudiant et le professeur

Les professeurs comme les étudiants disposent d'une page dédiée à la gestion de leur profil personnel. Sur cette page, ils peuvent :

- modifier leurs informations personnelles (nom, prénom, email, etc.).

- changer leur mot de passe pour renforcer la sécurité de leur compte.
- et enfin, supprimer définitivement leur compte s'ils le souhaitent, ce qui entraînera la suppression complète de leurs données de la plateforme.

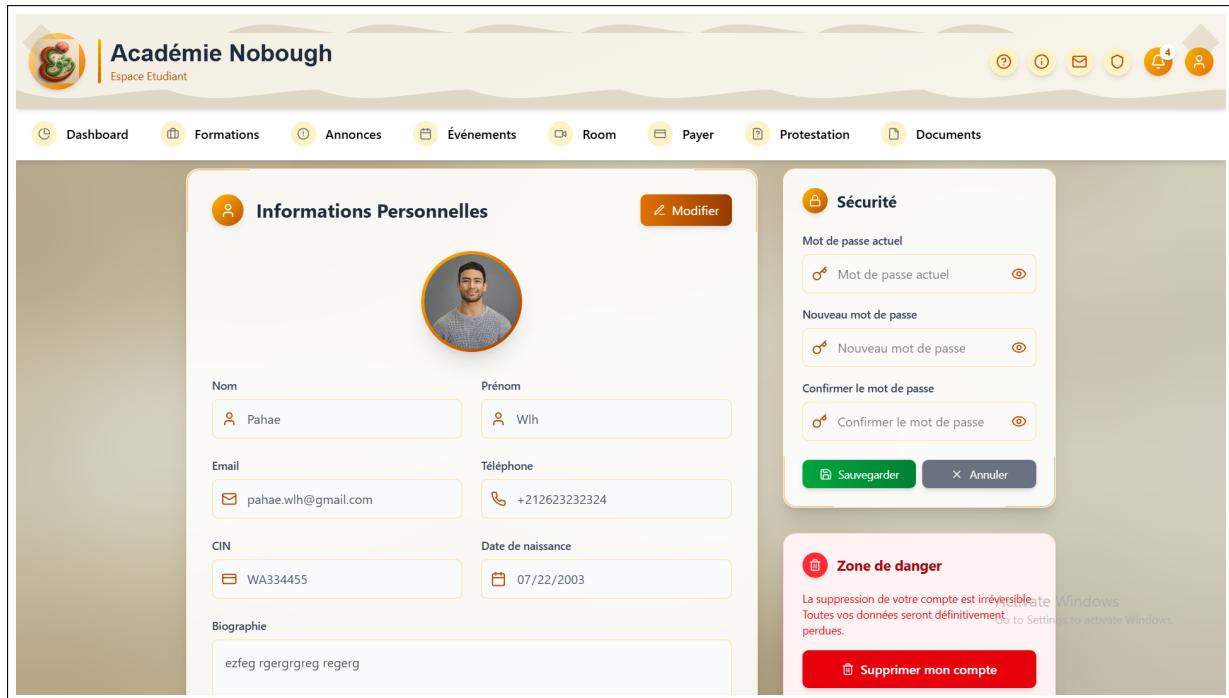


FIGURE 4.49 – Pages de profil pour l'étudiant et le professeur

15 Partie Comptable - Admin

15.1 Tableau de bord du le comptable

Le comptable bénéficie d'un espace dédié au sein de la plateforme, lui offrant une vue complète et détaillée de la situation financière de l'établissement. Cet espace lui permet de consulter les totaux des revenus, des dépenses, des dettes, ainsi que l'état du budget. Ce dernier est présenté sous deux formes :

- **Le budget total** : représentant l'ensemble des ressources financières prévues.
- **Le budget réel** : calculé dynamiquement en soustrayant, du budget total, les salaires des professeurs encore non réglés, offrant ainsi une vision plus précise de la trésorerie disponible.

Pour faciliter l'analyse et la prise de décision, le comptable peut également accéder à des graphiques interactifs illustrant la répartition des finances par mois, par catégorie, ou encore des comparaisons entre différentes périodes ou postes budgétaires. Ces visualisations permettent une compréhension rapide et intuitive des tendances économiques.

Enfin, un assistant intelligent (bot AI) est intégré à l'espace comptable. Ce dernier analyse en temps réel les données financières et peut proposer des prédictions basées sur les tendances observées (par exemple : estimation des revenus ou dépenses futurs), ou encore fournir des recommandations stratégiques (réduction de coûts, réaffectation de fonds, alerte sur des anomalies, etc.), afin d'optimiser la gestion financière de l'établissement.

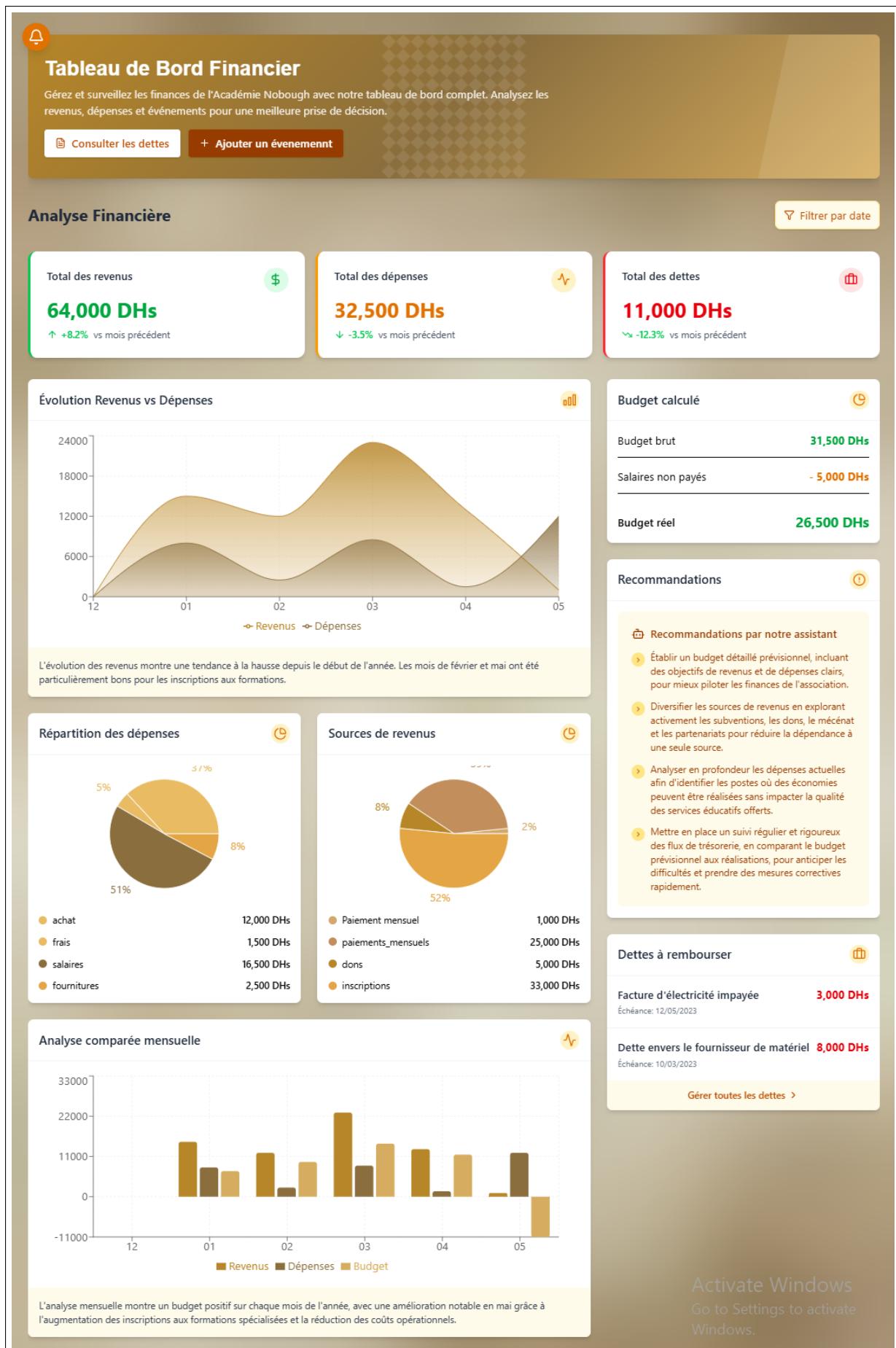


FIGURE 4.50 – Tableau de bord du le comptable

15.2 Consultation des Revenus, Dépenses et Dettes

Le comptable dispose d'un accès complet lui permettant de consulter en détail l'ensemble des données financières, notamment les revenus, les dépenses et les dettes de l'établissement. Chaque catégorie est présentée de manière claire avec des tableaux ou des graphiques explicites, lui offrant une vision précise de l'évolution financière dans le temps. En plus de la consultation, le comptable bénéficie également de fonctionnalités avancées lui permettant de gérer activement ces données. Il peut :

- **Ajouter** : de nouveaux enregistrements financiers (comme un revenu ponctuel ou une nouvelle dépense),
- **Modifier** : les informations existantes en cas d'erreur ou de mise à jour,
- **Supprimer** : des éléments obsolètes ou incorrects.

Ces actions sont protégées par des contrôles d'accès sécurisés et peuvent être accompagnées de systèmes de validation ou de journalisation pour garantir la traçabilité des modifications.

Récapitulatif des revenus				
SOURCE	DATE	MONTANT	REMARQUE	ACTIONS
Paiement de Wlh Pahae	25/05/2025	+250 DH	Paiement pour le mois de mai 2025	
Paiement de Wlh Pahae	25/05/2025	+250 DH	Paiement pour le mois de mai 2025	
Paiement de Wlh Pahae	25/05/2025	+250 DH	Paiement pour le mois de mai 2025	
Paiement de Wlh Pahae	25/05/2025	+250 DH	Paiement pour le mois de mai 2025	
Palements mensuels Palements mensuels avril	15/04/2023	+13,000 DH	Palements mensuels des étudiants	
Dons Don de Monsieur X	20/03/2023	+5,000 DH	Don généreux de Monsieur X	

FIGURE 4.51 – Consultation des revenus, dépenses et dettes

L'ajout d'un revenu peut se faire manuellement via un formulaire (type, montant, date, libellé, remarque), utile pour les revenus exceptionnels. Toutefois, la plupart sont générés automatiquement, notamment lors des paiements étudiants, assurant ainsi traçabilité et fiabilité des données.

← Ajouter un revenu

Informations du revenu
Veuillez remplir tous les champs obligatoires (*)

Type de revenu * <input type="text" value="Inscriptions"/>	Montant (DH) * <input type="text" value="0.00"/>
Date * <input type="date" value="05/26/2025"/>	Libellé * <input type="text" value="Entrez un libellé descriptif"/>
Remarque (optionnel) <small>Ajoutez des détails supplémentaires si nécessaire</small>	

Annuler
Enregistrer

FIGURE 4.52 – Ajout d'un revenu

15.3 Rembourser les dettes Intelligemment

Le comptable a également la possibilité d'accéder à une Prédiction Intelligente, propulsée par l'intelligence artificielle, lui permettant d'identifier les meilleures dettes à rembourser en priorité. Ce système d'aide à la décision analyse chaque dette en se basant sur quatre caractéristiques clés, soigneusement choisies pour évaluer la pertinence d'un remboursement :

- **Le degré d'urgence** : Certaines dettes peuvent impliquer des pénalités ou des conséquences juridiques si elles ne sont pas réglées rapidement.
- **L'ancienneté** : Plus une dette est ancienne, plus elle est prioritaire à rembourser.
- **La payabilité** : Capacité actuelle de l'établissement à rembourser la dette, en fonction de la trésorerie disponible.
- **Le pourcentage du budget** : Proportion que représente la dette par rapport au budget global, permettant d'estimer son impact financier.

Chaque dette est ainsi évaluée et notée selon ces quatre critères, et un score global est calculé pour chacune. Le système recommande alors en priorité celle(s) ayant le meilleur score, c'est-à-dire les dettes les plus stratégiques à rembourser dans l'intérêt de l'établissement.

Cette approche intelligente permet au comptable de prendre des décisions plus éclairées, plus rapides et plus efficaces, tout en minimisant les risques financiers.



FIGURE 4.53 – Rembourser les dettes intelligemment

15.4 L'administrateur

L'administrateur dispose d'un espace personnel centralisé, qui lui confère un ensemble complet de privilèges. Cet espace regroupe toutes les fonctionnalités accessibles à la fois à la secrétaire et au comptable, lui offrant ainsi une vue globale et un contrôle total sur les aspects organisationnels et financiers de l'établissement.

En plus de ces privilèges combinés, l'administrateur possède des droits exclusifs lui permettant de gérer les comptes du personnel administratif. Il peut notamment :

- **Créer de nouveaux comptes** : pour les secrétaires ou les comptables,
 - **Modifier les informations** : de comptes existants (nom, email, permissions, etc.),
 - **Supprimer des comptes** : si ils sont devenus inactifs ou obsolètes.
- Grâce à cet espace, l'administrateur agit comme le superviseur principal du système, garantissant la bonne coordination entre les différents services et assurant la gestion sécurisée des accès.

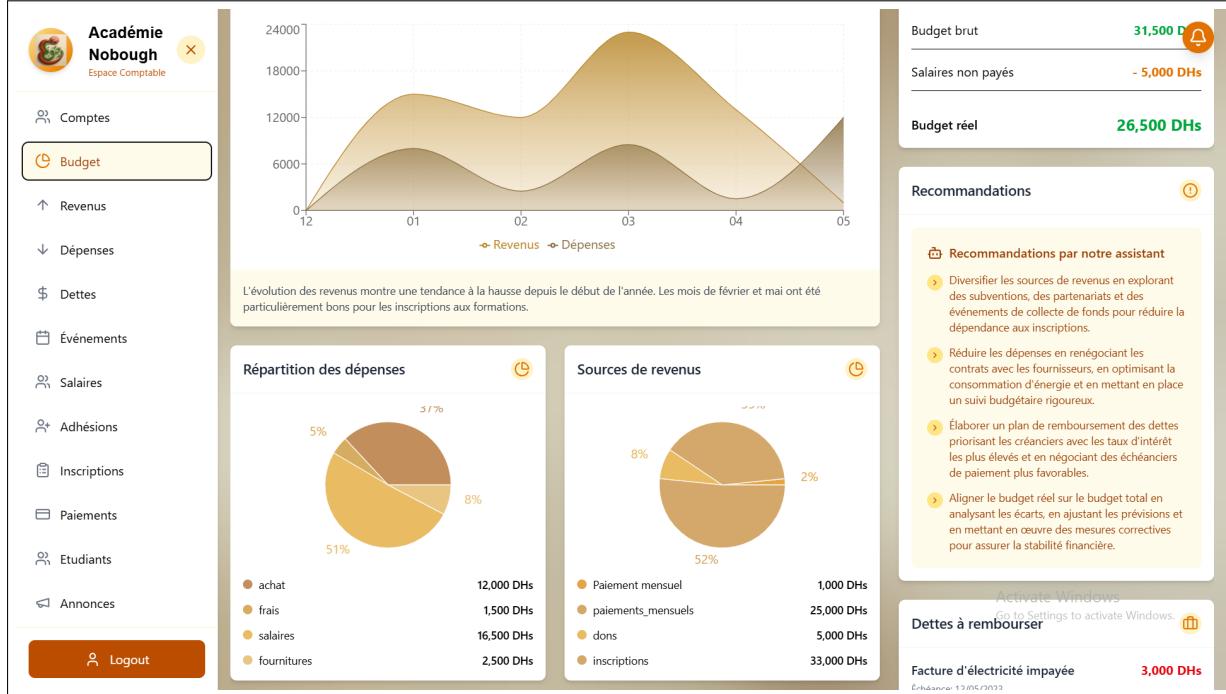


FIGURE 4.54 – Tableau de bord de l'admin

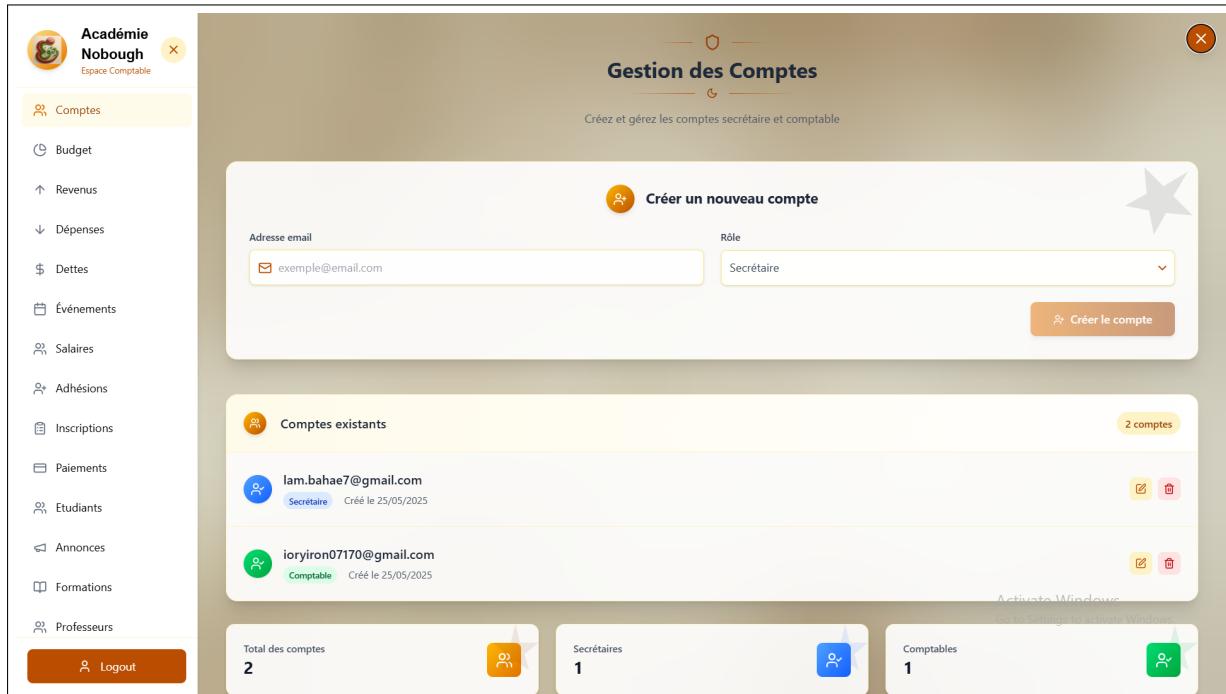


FIGURE 4.55 – Crédit de comptes secrétaire et comptable

IV Conclusion

L'application de l'Académie Nobough offre une solution complète pour la gestion des étudiants, professeurs, formations, et aspects administratifs et financiers. Elle inclut des fonctionnalités pour l'inscription et la validation des étudiants, la gestion des paiements, l'accès aux formations, et la consultation des annonces et protestations. Les professeurs peuvent créer des séances en ligne, interagir avec les étudiants, et générer des certificats. La secrétaire gère les inscriptions, annonces, et paiements, tandis que le comptable supervise les finances et utilise une IA pour le remboursement intelligent des dettes. L'administrateur a un contrôle total sur les comptes et les fonctionnalités. L'application assure une gestion efficace et sécurisée des processus académiques et administratifs.

Conclusion générale et perspectives

Ce projet de fin d'études a permis de concevoir et de développer une application web complète pour la gestion de l'Académie Nobough, visant à moderniser et automatiser les processus administratifs, pédagogiques et financiers de l'académie, tout en offrant une expérience utilisateur fluide et intuitive.

Au cours de ce projet, nous avons mené une analyse approfondie des besoins de l'académie, identifié les différents acteurs impliqués, et défini les fonctionnalités essentielles du système. Nous avons ensuite conçu l'interface utilisateur en utilisant des outils de prototypage visuel et modélisé le système à l'aide de diagrammes UML.

Pour le développement du site web, nous avons opté pour des technologies modernes du web, notamment Next.js et Tailwind CSS, afin de garantir une personnalisation rapide et un rendu visuel professionnel.

L'application développée offre une multitude de fonctionnalités pour les différents acteurs de l'académie, notamment pour les étudiants, les professeurs, la secrétaire, le comptable et l'administrateur. L'application intègre également des technologies avancées telles que l'intelligence artificielle pour la gestion des dettes et des recommandations financières, ainsi que la communication en temps réel avec WebRTC pour les séances en ligne.

Bien que l'application développée réponde aux besoins actuels de l'Académie Nobough, plusieurs perspectives d'amélioration et d'extension peuvent être envisagées pour l'avenir, telles que l'amélioration de l'interface utilisateur, l'extension des fonctionnalités, l'intégration de nouvelles technologies, l'amélioration de la sécurité, l'optimisation des performances, et le déploiement et la maintenance de l'application.

En conclusion, ce projet a été une expérience enrichissante qui a permis de mettre en pratique les compétences acquises tout au long de ma formation en Génie Informatique, et a également été l'occasion de se familiariser avec des outils et des technologies modernes, essentiels pour le développement web professionnel actuel. Les perspectives d'amélioration et d'extension ouvrent la voie à de futures innovations et améliorations pour répondre aux besoins évolutifs de l'Académie Nobough.

Bibliographie / Webographie

1. Prof. Said Kafhali – Cours génie logiciel et UML.
2. Next.js Documentation – <https://nextjs.org/docs> - Consulté pendant le 12/04/2025 jusqu'à 08/06/2025 en ligne.
3. WebRTC Official Guide – <https://webrtc.org/> - Consulté le 10/05/2025 en ligne.
4. Express.js Documentation – <https://expressjs.com/> - Consulté le 09/05/2025 en ligne.
5. Brevo (SendGrid alternative) SMTP Integration – <https://www.brevo.com/docs/> - Consulté le 20/04/2025 en ligne.
6. Guide YouTube pour TensorFlow et l'AI – https://youtu.be/EoYfa6mY0G4?si=WoNpL_u-gWR7-PHA - Consulté le 15/05/2025 en ligne.
7. TexPage – <https://www.texpage.com/> - Utilisé entre le 15/04/2025 et le 10/06/2025 en ligne.
8. Site web pour s'inspirer du design – <https://zad-academy.com/en> - Consulté le 09/04/2025 en ligne.
9. Puppeteer Documentation – <https://pptr.dev/> - Consulté le 25/05/2025 en ligne.
10. Tutoriel pour Puppeteer – https://youtu.be/9VgghGKx_1c?si=Rue2p20z__ZgsVPT - Consulté le 25/05/2025 en ligne.
11. Recharts (React) – <https://recharts.org/en-US> - Consulté le 18/04/2025 en ligne.