

MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA RECHERCHE
(MESR)

REPUBLIQUE TOGOLAISE
Travail-Liberté-Patrie



ECOLE SUPERIEURE DE GESTION D'INFORMATIQUE ET DES SCIENCES

RAPPORT DE STAGE
EN VUE DE L'OBTENTION DU DIPLOME DE LICENCE
PROFESSIONNELLE

Domaine : Science de l'ingénieur

Mention : Informatique Réseaux et Télécommunications

Spécialité : Architecture des logiciels (AL)

THEME DU RAPPORT :

**GESTION AUTOMATISÉE DES ACTIVITÉS D'UNE
ÉCOLE SECONDAIRE (INSCRIPTION, FRAIS DE SCOLARITÉ,
ÉVALUATIONS)**

Rédige par : NGARTOBAYE OUMAROU BILLY

Superviseur du rapport

N'SOUGAN Folly Emmanuel, M.

ANNEE ACADEMIQUE: 2024-2025

Remerciements

Avant toute chose, je tiens à rendre grâce à Dieu Tout-Puissant pour le don précieux de la vie et pour m'avoir soutenu à chaque étape de ce long parcours. Sans Sa présence et Son soutien, rien n'aurait été possible.

Je tiens à exprimer toute ma gratitude au Président du Conseil d'Administration de l'ESGIS, Monsieur Marcel Macy Akakpo, pour son engagement et sa vision qui permettent à tant d'étudiants comme moi d'accéder à une éducation de qualité. Merci également à toute l'administration et au corps enseignant de l'ESGIS, pour leur disponibilité et leur patience, qui nous ont permis de progresser et de nous épanouir tout au long de ces années d'études.

Un grand merci aussi aux agents d'entretien, qui veillent chaque jour à nous offrir un cadre propre et agréable, propice à l'apprentissage et à la réussite.

Je tiens à remercier du fond du cœur mes parents, qui ont été là à chaque instant, par leur amour, leur soutien moral et matériel, et qui ont tout fait pour que je puisse poursuivre ces études. Merci pour tous les sacrifices consentis et pour la force que vous m'avez donnée.

Je voudrais aussi adresser des remerciements tout particuliers à ma tante, Madame KPODAR Dédé Mawule, et à toute sa famille, pour leur présence rassurante et leur amour inconditionnel. Vous avez été un véritable pilier pour moi, et je vous en suis profondément reconnaissant.

Je n'oublie pas non plus tous mes camarades, avec qui j'ai traversé ces années souvent difficiles, remplies d'épreuves, de doutes et de moments de joie. Merci pour votre amitié, vos encouragements et votre soutien.

Enfin, je tiens à exprimer ma reconnaissance à toutes les personnes qui, de près ou de loin, par un mot, un geste, une prière, ont contribué à mon parcours et à ma réussite.

Que Dieu bénisse chacun de vous.

SOMMAIRE

Page de garde

Remerciements

Sommaire

Résumé

Abstract

Glossaire

Liste des figures

Liste des tableaux

Liste des participants au projet

Introduction

Première Partie : Cahier des charges

Chapitre 1 : Présentations

Chapitre 2 : Thème du stage

Chapitre 3 : Proposition de solutions

Chapitre 4 : Choix de la solution

Deuxième Partie : Analyse et Conception

Chapitre 1 : Présentation des outils d'analyse et de modélisation

Chapitre 2 : Étude détaillée de la solution

Conclusion

Troisième Partie : Réalisation et mise en œuvre

Chapitre 1 : Mise en œuvre

Chapitre 2 : Présentation de l'application

Quatrième partie : Exploitation et utilisation

Chapitre 1 : Configuration matérielle et logicielle

Chapitre 2 : Utilisation

Conclusion

Bibliographie

Webographie

Documents annexes

Tables des matières

Résumé

Ce rapport présente la conception et la mise en œuvre d'un système de gestion automatisée des activités d'un établissement secondaire, couvrant notamment la gestion des inscriptions, le suivi des frais de scolarité et la gestion des évaluations. Ce projet a été réalisé dans le cadre d'un travail personnel de fin de cycle, en troisième année de licence en informatique, parcours architecture des logiciels.

L'objectif principal est de proposer une solution numérique moderne permettant de remplacer les méthodes traditionnelles de gestion scolaire (registres papier, carnets, fichiers Excel) par un système centralisé, plus fiable, sécurisé et efficace. Ce système permet aux établissements scolaires de gérer les inscriptions, de suivre les paiements, de saisir et consulter les résultats des élèves, et de générer des rapports automatisés.

Le projet s'appuie sur une architecture logicielle moderne, combinant Next.js pour le développement de l'interface utilisateur (frontend), Django pour la logique métier (backend), et PostgreSQL pour la gestion des données. Ces technologies ont été choisies pour leur robustesse, leur performance et leur adaptabilité aux besoins spécifiques du domaine éducatif. Ce travail démontre la capacité d'un développement logiciel bien structuré à répondre aux défis quotidiens de la gestion scolaire.

Abstract

This report presents the design and implementation of an automated management system for secondary school activities, including student registration, tuition fee tracking, and evaluation management. The project was conducted as part of a personal academic endeavor in the third year of a software architecture degree program.

The main objective is to offer a modern digital solution that replaces traditional school management methods (paper records, notebooks, Excel spreadsheets) with a centralized, more reliable, secure, and efficient system. The developed system enables school staff to manage registrations, monitor tuition payments, input and view student results, and automatically generate detailed reports.

The system architecture is based on modern technologies: Next.js for the user interface (frontend), Django for the business logic (backend), and PostgreSQL for data management. These technologies were chosen for their robustness, performance, and ability to adapt to the specific needs of the education sector. This project demonstrates how well-structured software development can effectively address the practical challenges of school management.

Glossaire

- **Next.js** : Framework JavaScript open source basé sur **React**, utilisé pour le développement d'interfaces web dynamiques. Il permet un rendu côté serveur (SSR), une génération de pages statiques (SSG), et offre des performances optimisées pour les applications modernes, tout en facilitant le SEO.
- **React** : Bibliothèque JavaScript développée par Meta (anciennement Facebook), permettant de créer des interfaces utilisateurs interactives, modulaires et réactives sous forme de composants.
- **Django** : Framework web de haut niveau écrit en Python, qui encourage le développement rapide et une conception propre. Il fournit un ORM, une gestion automatique des routes, une interface d'administration intégrée, ainsi que des outils de sécurité avancés (protection CSRF, gestion des rôles, etc.).
- **Django REST Framework (DRF)** : Extension de Django permettant la création d'API RESTful robustes, bien structurées et sécurisées. Il facilite l'exécution d'opérations CRUD sur les données via des endpoints API.
- **PostgreSQL** : Système de gestion de base de données relationnelle (SGBDR) open source, reconnu pour sa robustesse, sa conformité aux standards SQL et sa capacité à gérer de grandes quantités de données avec des relations complexes.
- **JWT (JSON Web Token)** : Méthode d'authentification sécurisée consistant à échanger des jetons encodés, utilisée pour authentifier les utilisateurs sans stocker leur session côté serveur.
- **ORM (Object-Relational Mapping)** : Technique de programmation permettant de manipuler les données d'une base relationnelle via des objets métiers. Django ORM, par exemple, simplifie les requêtes sans avoir à écrire du SQL brut.
- **SSR (Server Side Rendering)** : Technique de rendu des pages côté serveur avant leur envoi au navigateur. Utilisée notamment dans Next.js, elle améliore les performances et le référencement.
- **SaaS (Software as a Service)** : Modèle de distribution logicielle où les applications sont hébergées à distance et accessibles via Internet, souvent par abonnement.
- **MERISE** : Méthodologie française d'analyse et de conception de systèmes d'information, utilisée pour modéliser les données (MCD, MLD) et les traitements de manière structurée.
- **UML (Unified Modeling Language)** : Langage de modélisation graphique utilisé pour représenter les composants d'un système logiciel (cas d'utilisation, diagramme de classe, séquence, etc.).
- **API REST** : Interface de programmation permettant à différentes applications de communiquer entre elles en respectant les principes REST (Représentation, État, Ressource, Verbe HTTP).
- **Responsive Design** : Approche de conception d'interfaces web garantissant une expérience utilisateur optimale quel que soit le terminal utilisé (ordinateur, tablette, smartphone).
- **Sécurité informatique** : Ensemble des méthodes et outils mis en œuvre pour protéger les données numériques contre les accès non autorisés, les fuites, les pertes ou les modifications malveillantes.

Liste des figures

• Figure 1 : Localisation ESGIS Kodjoviakope	5
• Figure 2 : Localisation ESGIS Avedji	6
• Figure 3 : Localisation ESGIS Adidogome	7
• Figure 4 : Logo UML	33
• Figure 5 : Diagramme de contexte statique	38
• Figure 6 : Diagramme de cas d'utilisation	40
• Figure 7 : Diagramme de séquence	41
• Figure 8 : Diagramme d'activité.....	42
• Figure 9 : Model Conceptuel de Données	47
• Figure 10 : Diagramme de classe	48
• Figure 11 : Page de connexion	72
• Figure 12 : Tableau de bord	73
• Figure 13 : Page d'inscription	74
• Figure 14 : Historique de paiement	75

Liste des tableaux

- Tableau 1 : Planning prévisionnel de réalisation 28
- Tableau 2 : Cas d'utilisation..... 39
- Tableau 3 : outils de développement et de programmation 52
- Tableau 4 : Matériel Cote client 60
- Tableau 5 : Matériel Cote serveur..... 61
- Tableau 6 : Technologies utilisées dans le projet 62
- Tableau 7 : Structure générale de la navigation 70

Liste des participants au projet

Nom	Rôle
NGARTOBAYE OUMAROU BILLY	Stagiaire développeur / Concepteur du système
M. N'SOUGAN Folly W.	Tuteur académique – Encadrement pédagogique et validation du projet

Introduction

Le stage en entreprise représente un moment essentiel dans la formation d'un étudiant en informatique, car il offre l'occasion de mettre en pratique les connaissances théoriques acquises tout au long de son cursus. À travers ce rapport, je retrace mon expérience de stage réalisée dans le cadre de ma troisième année de Licence Professionnelle en Informatique, spécialité Architecture des Logiciels, à l'ESGIS (École Supérieure de Gestion d'Informatique et des Sciences).

Le thème de votre stage porte sur la « **Gestion automatisée des activités d'une école secondaire (inscription, frais de scolarité, évaluations)** ». Ce projet a consisté à concevoir et développer un système informatique permettant de simplifier et de centraliser la gestion des inscriptions des élèves, le suivi des paiements des frais de scolarité, ainsi que l'enregistrement et la gestion des évaluations.

L'objectif principal de ce stage a été d'acquérir des compétences pratiques en développement d'applications, tout en répondant à un besoin concret rencontré par une école secondaire. À travers cette expérience, j'ai pu approfondir mes connaissances en architecture des logiciels, en conception de bases de données, en modélisation UML/Merise et en programmation. Cette immersion dans le milieu professionnel m'a également permis de comprendre les contraintes réelles d'un projet informatique, les attentes des utilisateurs finaux et l'importance d'un système fiable et sécurisé.

Dans ce rapport, je présenterai dans un premier temps le cadre général de mon stage et l'environnement de travail. Ensuite, j'exposerai la problématique à laquelle répond le projet, avant de détailler les différentes étapes de conception, de réalisation et de mise en œuvre de la solution retenue. Enfin, je présenterai l'exploitation de l'application, les enseignements tirés de cette expérience et les perspectives qu'elle ouvre pour la suite de mon parcours.

Première Partie : Cahier des charges

Chapitre 1 : Présentations

I. Présentation de l'ESGIS

L'École Supérieure de Gestion d'Informatique et des Sciences (ESGIS) est un établissement privé d'enseignement supérieur agréé par les ministères de l'Enseignement Supérieur au Togo, au Bénin et au Gabon, et accrédité par le CAMES. Fondée en 1994 au Togo, elle a été étendue au Bénin en 2005 et au Gabon en 2012. Depuis 30 ans, l'ESGIS s'attache à former des cadres de haut niveau, dynamiques et intègres, capables de répondre aux défis économiques et technologiques en Afrique.

L'institution repose sur une vision panafricaine : produire des professionnels compétents dans les secteurs de la gestion, des sciences et des technologies grâce à une formation alliant théorie et pratique. Elle se distingue par un taux d'insertion professionnelle élevé, plus de 20 000 diplômés, un corps enseignant qualifié et des infrastructures modernes de pointe (ordinateurs, laboratoires, serveurs, équipements réseau...).

ESGIS fonctionne sur plusieurs campus ; au Togo, elle en compte trois principaux à Lomé :

- **Pôle de Kodjoviakope**

Ouvert en 1994, dans le quartier de Kodjoviakope. L'adresse officielle est le 8, rue Agnès Gaba, non loin de la zone Ocam-Bè Souza Netimé. Il s'agit du premier site togolais de l'ESGIS, implanté dans un environnement urbain proche d'assurances et de services publics, bien desservi par les transports en commun.

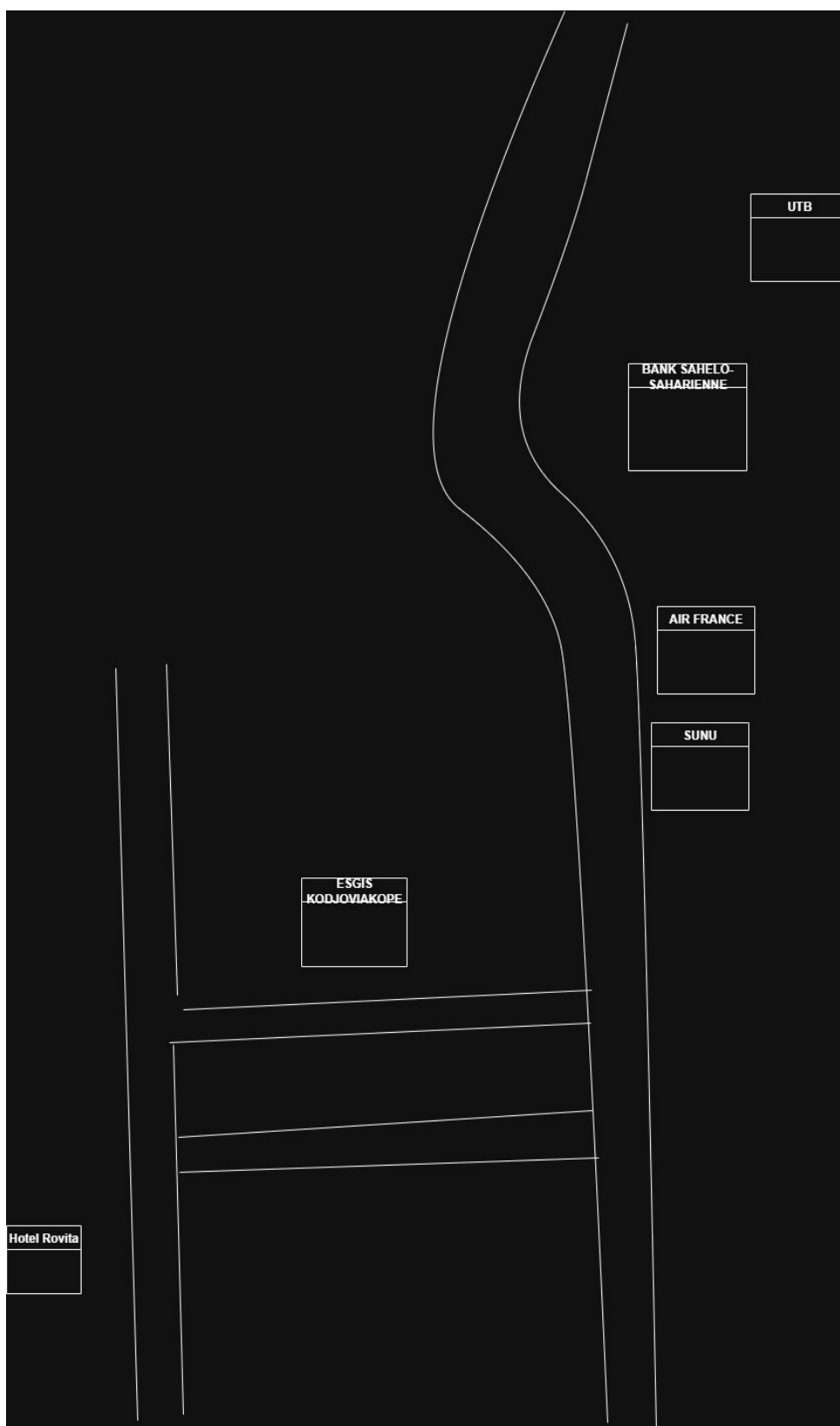


Figure 1 : Localisation ESGIS Kodjoviakope

- **Pôle d'Avédji**

Ouvert en 2005, ce campus se trouve à Avédji, en face de la pâtisserie Opéra et à proximité de l'hôtel La Concorde, ainsi que de l'école René-Mariam (RM). Il sert de relais pour les étudiants habitant l'est de Lomé, dans un quartier à vocation résidentielle et commerciale.

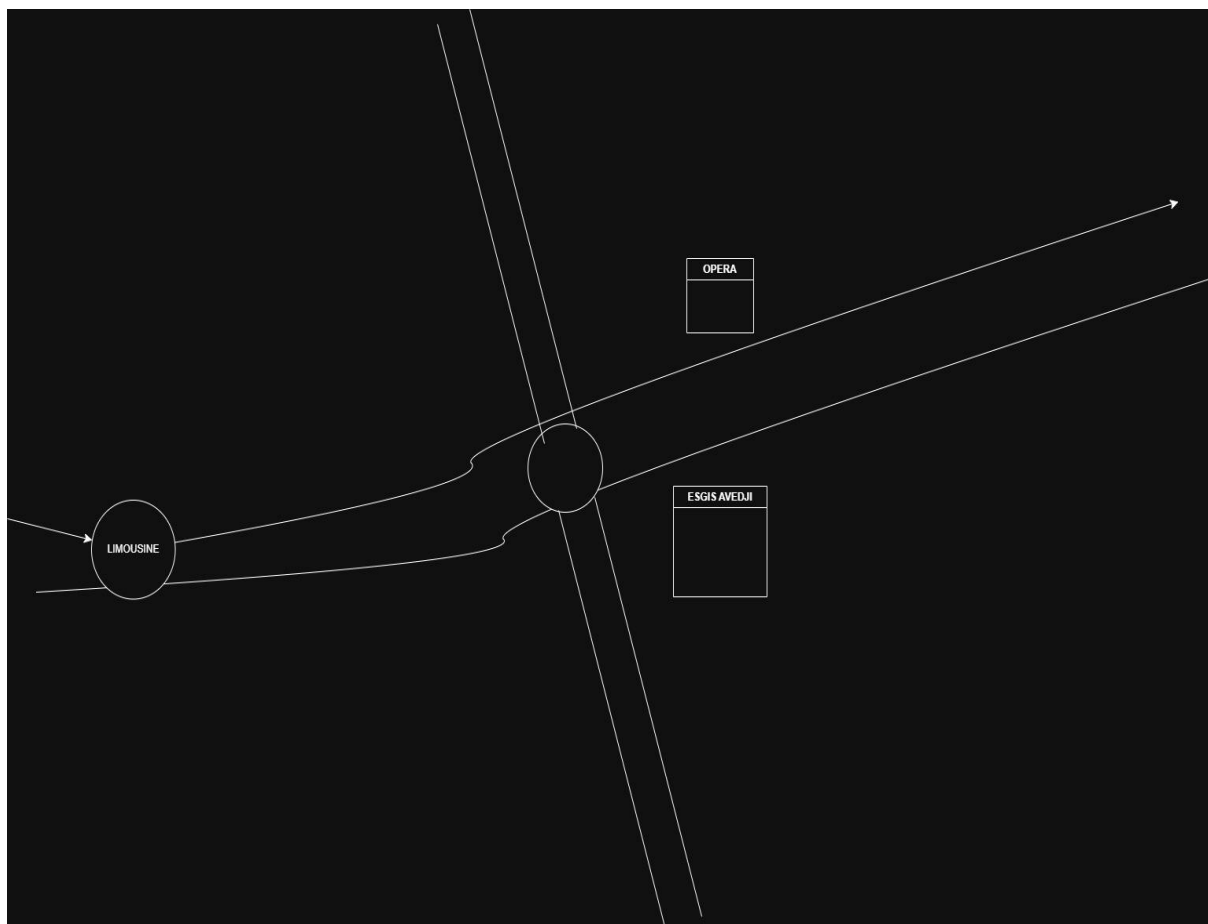


Figure 2 : Localisation ESGIS Avédji

- **Pôle d'Adidogomé**

C'est le plus récent, inauguré officiellement le 30 août 2024, en bordure du boulevard du 30 Août, près du rond-point de l'ancienne douane et en face du camp militaire d'Adidogomé. Ce campus moderne sur deux étages (capacité de 2 500 places, extensible) possède des infrastructures récentes et adaptées à un apprentissage de qualité.

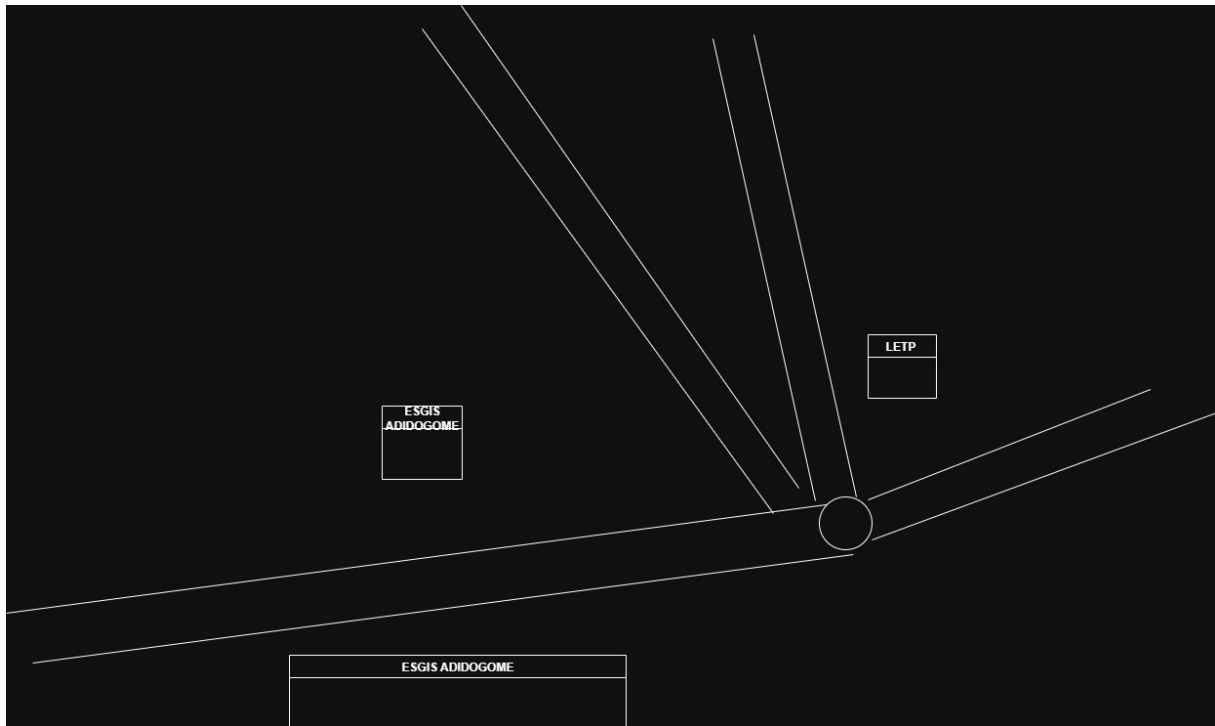


Figure 3 : Localisation ESGIS Adidogomé

Ces trois campus permettent à l'ESGIS de proposer une couverture géographique complète, facilitant l'accès aux études supérieures dans des zones stratégiques de Lomé. Ils symbolisent également la vision de l'école d'offrir des infrastructures modernes et adaptées, répondant aux attentes des étudiants et du milieu professionnel.

Grâce à cette présence physique renforcée, l'ESGIS assure à ses étudiants un environnement d'apprentissage efficace : salles de cours équipées, laboratoires informatiques, accès internet haut débit, bibliothèques, zones d'étude et services d'appui administratif.

Conclusion du chapitre

En somme, l'ESGIS se distingue comme un acteur majeur de la formation supérieure en Afrique de l'Ouest, notamment dans les domaines du numérique, de la gestion et des sciences appliquées. Son implantation stratégique, ses infrastructures modernes, et son encadrement de qualité en font un cadre idéal pour l'éclosion de compétences solides et adaptées aux besoins du marché. C'est dans cette dynamique d'excellence que s'inscrit la réalisation de ce stage de fin de cycle.

Chapitre 2 : Thème du stage

I- Présentation du sujet

Le thème de ce travail est intitulé « Gestion automatisée des activités d'une école secondaire », avec une attention particulière portée sur trois aspects majeurs de la gestion administrative scolaire : l'automatisation des inscriptions, la gestion informatisée des frais de scolarité, et la centralisation des évaluations académiques.

Ce sujet est né d'un constat simple mais récurrent dans plusieurs établissements secondaires : la majorité des processus de gestion restent manuels, fastidieux et sujets à des erreurs fréquentes. Les inscriptions se font encore souvent sur papier, les paiements sont suivis dans des carnets ou fichiers Excel non sécurisés, et les résultats sont compilés à la main, ce qui ralentit considérablement la communication avec les parents et les élèves.

Confronté à cette réalité, l'idée de développer une application web scolaire moderne s'est imposée naturellement. Le système conçu vise à répondre aux défis concrets rencontrés par les écoles dans la gestion de leurs activités quotidiennes, en leur apportant des outils adaptés, simples d'utilisation et sécurisés.

Plus précisément, cette solution permet :

- aux établissements de gérer automatiquement les inscriptions, en assurant la collecte et le stockage sécurisé des informations des élèves ;
- de suivre en temps réel les paiements des frais de scolarité, en permettant une gestion transparente et traçable de la comptabilité scolaire ;
- de centraliser les évaluations, en facilitant la saisie, la consultation et la diffusion des résultats via une interface intuitive.

Ce projet s'inscrit donc dans une démarche de digitalisation intelligente du secteur éducatif, tout en valorisant des technologies modernes comme Next.js, Django, et PostgreSQL. L'ensemble de ces solutions vise à améliorer non seulement la productivité du personnel administratif, mais aussi l'expérience globale des élèves, parents et enseignants.

II- Problématique du sujet

Dans de nombreux établissements secondaires, la gestion des inscriptions, du suivi des paiements et des évaluations académiques repose encore majoritairement sur des procédés manuels ou semi-informatisés, souvent limités à l'usage de tableurs ou de registres papier. Si ces méthodes ont longtemps fait leurs preuves, elles s'avèrent aujourd'hui inadaptées face aux exigences croissantes de rapidité, de fiabilité et de transparence.

Cette gestion traditionnelle engendre plusieurs difficultés notables, parmi lesquelles :

- des risques fréquents d'erreurs de saisie, d'omissions ou de doublons dans les informations traitées ;
- une absence de centralisation des données, rendant difficile l'accès instantané aux informations essentielles ;
- une charge de travail importante pour le personnel administratif, contraint de répéter les mêmes tâches manuelles à plusieurs reprises ;
- une faible réactivité dans la production de rapports ou l'analyse des données scolaires, ce qui nuit à la prise de décision rapide et éclairée.

Dans ce contexte, il devient impératif de repenser ces processus à travers des solutions numériques intégrées, capables de garantir l'automatisation, la sécurisation et la fiabilité des opérations.

La problématique principale ayant guidé ce travail est donc la suivante : Comment concevoir et mettre en œuvre un système numérique sécurisé, capable de gérer de manière centralisée, efficace et automatisée les inscriptions, les frais de scolarité et les évaluations au sein d'un établissement secondaire ?

Ce questionnement a orienté la réflexion autour des enjeux techniques, organisationnels et fonctionnels d'un tel système, en vue de proposer une solution concrète et durable.

III- Intérêt du sujet

Le choix de ce sujet de stage est motivé par plusieurs raisons majeures :

a. Objectifs

Le projet vise à répondre à un besoin réel de modernisation des outils de gestion dans les établissements secondaires. Pour cela, plusieurs objectifs fonctionnels et techniques ont été définis :

- Automatiser le processus d'inscription
L'objectif est de mettre en place une procédure d'inscription en ligne intuitive, qui réduit les délais de traitement, évite les erreurs de saisie manuelle, et simplifie l'enregistrement des nouveaux élèves grâce à un système de vérification automatisée des données.
- Assurer un suivi fiable et centralisé des frais de scolarité
Il s'agit d'intégrer un module permettant d'enregistrer, de suivre et d'archiver tous les paiements effectués par les parents ou responsables d'élèves. Le système devra générer des rapports détaillés, utiles à la fois pour la comptabilité interne et pour la transparence envers les usagers.
- Faciliter la gestion des évaluations
L'application doit offrir aux enseignants une interface simple pour saisir les notes, et aux parents/élèves une possibilité de les consulter en temps réel. Cela permet une meilleure communication des résultats et une réduction du travail administratif lié à la compilation manuelle des bulletins.
- Améliorer la transparence et la traçabilité des opérations
Toutes les actions effectuées dans le système (inscriptions, paiements, saisies de notes) doivent être enregistrées, datées et sécurisées, afin de pouvoir être vérifiées en cas de besoin. Cela renforce la confiance dans le fonctionnement de l'établissement.
- Offrir un outil ergonomique, sécurisé et accessible
L'interface doit être conçue pour être simple d'utilisation, agréable visuellement et utilisable sur plusieurs supports (ordinateur, tablette, smartphone). L'accès au système doit être protégé par des niveaux de permission, en fonction du profil de l'utilisateur (administrateur, enseignant, parent...).

b. Résultats attendus

Au terme de la conception et de la mise en œuvre du projet, plusieurs résultats concrets et mesurables sont attendus. Ces résultats traduisent l'impact direct que l'application vise à produire dans l'environnement scolaire :

- Un système web fonctionnel, intuitif et performant
L'application doit être accessible via un navigateur web, rapide à charger, fluide dans la navigation, et suffisamment simple pour être utilisée sans formation poussée. Elle doit répondre efficacement aux besoins des différents profils utilisateurs tout en garantissant une expérience fluide et agréable.
- Une base de données centralisée, garantissant l'unicité et la sécurité des informations
Toutes les données relatives aux élèves, aux paiements et aux évaluations seront stockées dans une base de données unique et bien structurée, évitant ainsi les doublons, les pertes de données ou les incohérences. Des mécanismes de sécurité (authentification, rôles, journalisation) seront mis en place pour protéger les données sensibles.
- Une meilleure organisation du travail administratif et une réduction notable de la charge de travail
Grâce à l'automatisation des tâches répétitives (inscriptions, calculs de moyennes, suivi des paiements), le personnel administratif pourra se consacrer à des tâches à plus forte valeur ajoutée. Cela se traduira par un gain de temps, une diminution des erreurs et une efficacité accrue au sein de l'établissement.
- Des rapports clairs et fiables pour un suivi rigoureux de l'évolution des activités scolaires
Le système permettra de générer automatiquement des tableaux de bord, des statistiques, et des rapports (par classe, par élève, par période). Ces outils décisionnels seront utiles à l'administration pour piloter l'activité de l'établissement avec plus de rigueur et de transparence.
- Une plus grande satisfaction des utilisateurs (élèves, parents, administration)
En facilitant l'accès aux informations (notes, reçus de paiement, bulletins...), en simplifiant les démarches (inscription, suivi, consultation) et en garantissant un environnement sécurisé, le système contribuera à renforcer la confiance et l'implication des différents acteurs dans la vie scolaire.

IV- Étude de l'existant

Avant de concevoir une solution automatisée pour la gestion scolaire, une observation approfondie du fonctionnement traditionnel a été menée dans un établissement secondaire type. Cette étude s'est appuyée sur les méthodes actuellement utilisées pour gérer les inscriptions, les paiements de frais de scolarité et les évaluations des élèves. Il en ressort les constats suivants :

- **Gestion des inscriptions**

Les inscriptions s'effectuent sur place, à l'aide de fiches papier remplies manuellement par les parents ou par le personnel administratif. Ces fiches comprennent les informations personnelles et scolaires de l'élève, ainsi que la liste des pièces justificatives requises. Une fois complétées, elles sont archivées dans des dossiers physiques et classées dans des armoires.

- **Enregistrement des paiements**

Les règlements des frais de scolarité sont effectués directement auprès du secrétariat ou du service comptable, généralement en espèces. Chaque paiement est noté manuellement dans un cahier dédié. Un reçu manuscrit est remis aux parents. À la fin de chaque période, les données sont reportées dans un tableur Excel pour produire les états financiers.

- **Saisie et gestion des évaluations**

Les enseignants notent les résultats des élèves dans des carnets personnels. En fin de trimestre, ces notes sont communiquées à l'administration ou recopiées sur des bulletins imprimés. Les moyennes sont calculées manuellement ou à l'aide d'outils rudimentaires comme des feuilles Excel. Cette étape mobilise beaucoup de temps, notamment pour la saisie, la relecture et l'édition des bulletins.

- **Traitement et stockage des données**

Chaque service (administration, comptabilité, pédagogie) fonctionne de manière indépendante, avec ses propres fichiers et supports physiques. Les données ne sont ni synchronisées ni centralisées, ce qui limite la fluidité de l'information et accroît les risques de perte ou d'incohérence. Aucun système de sauvegarde systématique n'est mis en place, ce qui rend l'ensemble vulnérable à tout incident (incendie, vol, erreur humaine, etc.).

V- Critique de l'existant

À la lumière de l'étude menée sur le système de gestion actuel des établissements secondaires, il est possible de dresser un bilan objectif en mettant en évidence aussi bien les points positifs que les limites structurelles.

Forces observées dans le système existant

- Le personnel administratif a développé une **maîtrise empirique** des outils simples comme les registres papier ou les tableurs Excel, facilitant une certaine continuité dans la gestion scolaire.
- Les documents papier, bien archivés, peuvent constituer une **trace physique durable** et accessible même sans ressources numériques.
- Les interactions humaines entre le personnel et les parents lors des inscriptions favorisent une **relation de proximité** et permettent des échanges personnalisés.
- Les outils utilisés (carnets, fiches, cahiers de comptes) sont **économiquement accessibles**, ne nécessitant ni maintenance informatique ni infrastructures techniques coûteuses.

Faiblesses et limites du système existant

- L'usage de **registres papier et de fichiers Excel** atteint rapidement ses limites lorsqu'il s'agit de gérer un grand volume d'élèves, de paiements et de notes, rendant les traitements longs et fastidieux.
- Les processus sont **entièrement manuels**, ce qui augmente considérablement les risques d'erreurs (saisie, calculs, oublis) et allonge les délais de traitement.
- L'information étant dispersée entre plusieurs supports et services, **aucune centralisation** n'est possible, ce qui nuit à la coordination entre les départements (comptabilité, direction, enseignants).
- **Le manque d'automatisation** empêche la génération rapide de statistiques ou de rapports, rendant la prise de décision lente et approximative.
- **Aucune solution de sauvegarde ou de sécurisation** n'est formellement mise en place, exposant l'établissement à une perte totale des données en cas de sinistre ou de mauvaise manipulation.

- Le système existant n'offre **aucun accès distant** aux informations (paiements, bulletins, inscriptions), ce qui empêche les parents ou les élèves d'interagir avec l'administration en ligne.

Conclusion de l'analyse critique

Ces constats ont permis de mesurer l'écart entre les pratiques actuelles et les besoins réels d'un établissement moderne. Ils ont justifié la nécessité de concevoir un **système numérique intégré**, capable de centraliser les données, d'automatiser les tâches administratives et de renforcer la sécurité et la transparence dans la gestion des opérations scolaires.

Conclusion du chapitre

Le thème du stage s'inscrit dans une problématique concrète et pertinente : l'automatisation des activités administratives et pédagogiques dans un établissement secondaire. Il vise à répondre aux défis quotidiens liés à la gestion manuelle des inscriptions, des frais de scolarité et des évaluations. À travers l'analyse des besoins, la formulation d'objectifs clairs et l'étude critique de l'existant, ce thème s'est imposé comme un projet à fort impact à la fois pédagogique et technologique.

Chapitre 3 : Proposition de solutions

I. Spécification de la solution

a. Fonctionnelles

Les spécifications fonctionnelles définissent les comportements attendus du système à mettre en place, en réponse aux besoins exprimés par les utilisateurs finaux. Dans le cadre de ce projet, ces besoins ont été identifiés à travers l'analyse des processus existants dans les établissements secondaires et les entretiens avec des responsables pédagogiques.

Le système doit offrir les fonctionnalités clés suivantes :

- **Gestion des inscriptions**

Le système doit permettre l'enregistrement complet d'un élève, en recueillant ses informations personnelles (nom, prénom, date de naissance, sexe, adresse), ainsi que les données relatives à sa scolarité (classe, année scolaire, tuteur légal).

L'administrateur pourra imprimer automatiquement des fiches d'élève et des reçus d'inscription.

- **Suivi des frais de scolarité**

Chaque paiement effectué est enregistré avec les détails nécessaires : montant, élève concerné, trimestre et date. Le système permet de suivre l'historique de chaque élève, de générer des reçus et d'alerter l'administration en cas de retard. Des options d'exportation comptable (PDF ou Excel) sont également intégrées.

- **Gestion des évaluations**

Les enseignants peuvent saisir les notes par matière et par période (trimestre ou semestre), tandis que le système calcule automatiquement les moyennes et les classements. Des bulletins de notes peuvent être générés pour chaque élève, avec un système de validation avant publication.

- **Accès personnalisé selon le profil utilisateur**

Le système prévoit plusieurs types d'accès :

- **Administrateur** : contrôle global du système, gestion des utilisateurs et de la configuration.
- **Enseignant** : accès restreint à la saisie et consultation des notes.
- **Comptable** : suivi des paiements et édition des rapports financiers.
- **Parent / Élève** : consultation des notes, de l'historique des paiements et des informations scolaires.

- **Tableaux de bord dynamiques**

Des tableaux de bord interactifs affichent en temps réel les indicateurs clés : nombre d'élèves inscrits, paiements en attente, moyenne générale de la classe, répartition par niveau, etc.

- **Historique et archivage**

Les données d'une année scolaire sont archivées et consultables même après clôture de l'année, assurant une traçabilité complète du parcours de chaque élève.

Ces spécifications visent à automatiser les tâches répétitives, à améliorer la réactivité administrative et à offrir une meilleure transparence entre les différents acteurs du système scolaire.

b. Spécifications de sécurité

La sécurité constitue un pilier central du projet, en raison de la nature confidentielle des informations manipulées (données personnelles, paiements, évaluations). Pour répondre à ces exigences, plusieurs mesures techniques et organisationnelles ont été définies :

- **Gestion des accès et authentification**

Chaque utilisateur se connecte à l'aide de ses identifiants personnels. Un système d'authentification sécurisé basé sur des jetons JWT (JSON Web Tokens) est mis en œuvre. Les droits d'accès sont gérés selon le rôle de l'utilisateur.

- **Chiffrement des données sensibles**

Les mots de passe sont stockés de manière sécurisée à l'aide d'un algorithme de hachage robuste (bcrypt). Toutes les communications entre le client et le serveur sont sécurisées par le protocole HTTPS.

- **Sauvegarde régulière des données**

Une routine de sauvegarde automatique est planifiée quotidiennement pour sécuriser l'ensemble de la base de données PostgreSQL. Les sauvegardes sont conservées sur un espace de stockage externe avec contrôle d'intégrité.

- **Journalisation des actions critiques (logs)**

Toutes les actions sensibles (modification de notes, enregistrement de paiements, suppression de comptes) sont tracées dans un journal d'événements sécurisé, permettant un suivi précis et un audit ultérieur.

- **Prévention des vulnérabilités courantes**

Le système intègre des protections contre les attaques de type injection SQL, XSS ou CSRF, grâce à l'utilisation du framework Django (ORM sécurisé) côté serveur, et à une validation rigoureuse des entrées côté client.

Ces mesures garantissent la confidentialité, l'intégrité et la disponibilité des données, tout en assurant une protection renforcée contre les menaces courantes sur les applications web.

II. Propositions de solution

Pour répondre aux besoins identifiés, nous avons envisagé deux approches : l'adoption d'une solution existante du marché ou le développement d'un système personnalisé. Dans cette section, nous présentons en détail une solution réelle existante : **EDUKA**.

1-Solution existante choisie : EDUKA

EDUKA est une plateforme de gestion scolaire en ligne (SaaS – Software as a Service) conçue pour accompagner les établissements d'enseignement dans la digitalisation de leur administration. Développée par la société française Educaschool, cette solution est utilisée par de nombreuses écoles privées en Afrique de l'Ouest (notamment au Sénégal, au Cameroun, en Côte d'Ivoire, au Togo, etc.).

Description de la solution

EDUKA est un logiciel 100 % en ligne, accessible via navigateur web. Il propose une interface multi-utilisateur avec des modules spécialisés selon le profil : direction, enseignants, comptables, parents et élèves.

Parmi les fonctionnalités proposées :

- **Inscription en ligne des élèves** avec génération automatique de fiches.
- **Gestion des frais de scolarité** (facturation, suivi des paiements, alertes de retard).
- **Saisie des notes et bulletins scolaires**.
- **Communication école-famille** (messagerie intégrée, notifications, convocations).
- **Édition automatique de documents officiels** (certificats, bulletins, attestations).
- **Tableaux de bord analytiques** pour le suivi en temps réel.

EDUKA propose également une application mobile pour Android et iOS.

Avantages de la solution EDUKA

- **Gain de temps et mise en œuvre immédiate** : aucune phase de développement n'est nécessaire, l'école peut être opérationnelle en quelques jours après configuration.
- **Solution complète et intégrée** : tous les modules essentiels à la gestion d'un établissement sont présents.

- **Support technique et maintenance inclus** : l'éditeur assure les mises à jour, la sécurité, et le support.
- **Expérience utilisateur fluide** : l'interface a été conçue de façon intuitive pour tous les profils d'utilisateurs.
- **Accessibilité** : étant cloud-based, l'accès peut se faire à distance, 24h/24, avec un simple accès Internet.

Inconvénients de la solution EDUKA

- **Coût d'abonnement mensuel/annuel élevé** : le prix est généralement basé sur le nombre d'élèves ou de modules utilisés, ce qui peut s'avérer coûteux pour les établissements de taille moyenne.
- **Personnalisation restreinte** : les écoles doivent s'adapter aux fonctionnalités existantes, sans possibilité d'ajouter des modules spécifiques à leur fonctionnement.
- **Dépendance technique** : l'école ne contrôle ni l'infrastructure ni le code source.
- **Problématique de souveraineté des données** : les données des élèves sont stockées sur des serveurs à l'étranger, ce qui peut poser problème si la législation locale exige un hébergement sur le territoire national.

2. Solution personnalisée – Développement d'un système sur mesure

Description

Dans l'optique d'offrir une réponse spécifique aux besoins des établissements scolaires locaux, une solution logicielle sur mesure sera développée. Ce système sera conçu pour automatiser les principales tâches administratives telles que les inscriptions, la gestion des frais de scolarité, les évaluations et la production de bulletins.

Le développement de cette solution reposera sur des technologies web modernes et robustes :

- **Frontend** : Next.js – framework basé sur React, combinant génération statique et rendu côté serveur (SSR) pour une interface fluide et rapide.
- **Backend** : Django – framework Python puissant et sécurisé, associé à **Django REST Framework** pour exposer une API structurée et réutilisable.

- **Base de données** : PostgreSQL – système relationnel reconnu pour sa fiabilité, sa performance et sa compatibilité avec les environnements complexes.

Ce système sera déployé soit en hébergement local (serveur de l'établissement), soit dans un cloud contrôlé en fonction des besoins de l'école, garantissant ainsi la souveraineté des données.

Avantages

- **Personnalisation totale** : toutes les fonctionnalités pourront être ajustées aux réalités de l'établissement (gestion spécifique des trimestres, frais modulaires, impression des documents officiels, etc.).
- **Évolutivité garantie** : la solution sera conçue de manière modulaire, permettant l'ajout progressif de nouveaux modules (ex. : gestion de bibliothèque, messagerie interne, portail parents...).
- **Contrôle des données** : l'établissement pourra choisir son mode d'hébergement, garantissant la confidentialité et le respect des réglementations locales sur la protection des données.
- **Indépendance technologique** : l'école ne dépendra d'aucun prestataire externe pour la gestion ou l'évolution de son outil.

Inconvénients (à anticiper)

- **Nécessité de temps pour le développement** : la mise en œuvre complète nécessitera une phase de conception rigoureuse, suivie de développement, de tests et de déploiement.
- **Besoins techniques à long terme** : une équipe technique devra être mobilisée pour assurer la maintenance, les mises à jour et le support aux utilisateurs.
- **Investissement initial plus conséquent** : bien que sans abonnement mensuel, la solution nécessitera un effort financier initial (main-d'œuvre, matériel, formation) avant d'être pleinement fonctionnelle.

Chapitre 4 : Choix de la solution

I. Solution retenue

Après l'analyse comparative des deux approches (solution SaaS existante vs. développement personnalisé), la solution retenue est celle d'un **développement sur mesure**, en raison de sa **souplesse**, de sa **capacité d'adaptation** aux réalités spécifiques de l'école et de sa **durabilité** à long terme.

Ce choix permet notamment :

- Une personnalisation totale de la gestion des trimestres, des frais, des bulletins et des profils utilisateurs.
- Une indépendance technique vis-à-vis de prestataires externes.
- Une meilleure maîtrise des données, hébergées localement ou sur un cloud privé.
- Une évolutivité, avec la possibilité d'intégrer à l'avenir de nouvelles fonctionnalités (messagerie interne, portail parent, etc.).

III. Planning de réalisation

a. Planning prévisionnel de réalisation

Tab 1 : Planning prévisionnel de réalisation

Phase	Tâches principales	Durée estimée (jours)	Dates prévues
Phase 1 : Analyse	Récolte des besoins, rédaction du cahier des charges	5 jours	01/03/2025 – 05/03/2025
Phase 2 : Conception	Maquettage, modélisation base de données, diagrammes	6 jours (en partie parallèle avec Phase 1)	03/03/2025 – 10/03/2025
Phase 3 : Développement	Implémentation frontend (Next.js) et backend (Django)	10 jours	06/03/2025 – 17/03/2025
Phase 4 : Tests	Vérifications fonctionnelles, sécurité, corrections	4 jours	15/03/2025 – 20/03/2025
Phase 5 : Déploiement	Hébergement, configuration serveur, test production	2 jours	21/03/2025 – 22/03/2025
Phase 6 : Rapport	Structuration et rédaction du rapport de stage	6 jours	18/03/2025 – 25/03/2025

Conclusion au chapitre

Le choix final de développer une solution sur mesure s'est révélé être le plus pertinent, en raison de sa flexibilité, de son alignement avec les besoins de l'école et de son caractère évolutif. L'évaluation financière réalisée a permis d'anticiper les ressources nécessaires, tandis que le planning de réalisation a offert un cadre temporel rigoureux. Ce chapitre consolide ainsi la base du projet avant d'aborder la phase d'analyse et de conception détaillée.

Deuxième Partie : Analyse et Conception

Chapitre 1 : Présentation des outils d'analyse et de modélisation

I. Présentation de la méthode d'analyse

Pour mener à bien la conception du système de gestion automatisée des activités scolaires, nous avons adopté une **approche orientée objet**, basée principalement sur la **méthodologie UML (Unified Modeling Language)**.

Ce choix s'explique par plusieurs raisons :

- UML est **universellement reconnu** dans le domaine du développement logiciel et offre une **notation graphique claire** et standardisée pour modéliser les aspects fonctionnels, structurels et comportementaux d'un système.
- Elle permet de **communiquer efficacement** avec toutes les parties prenantes, même non techniques.
- Elle facilite la **transition vers le développement** car la modélisation en objets s'aligne naturellement avec les langages modernes utilisés (comme Python ou JavaScript).

Ainsi, **la méthode UML a guidé toutes les étapes de la modélisation**, depuis l'étude fonctionnelle jusqu'à la conception technique du système.



Fig 4 : Logo UML

II. Langage et outils de modélisation utilisés

a. Langage utilisé : UML (Unified Modeling Language)

Le langage utilisé pour décrire et représenter les composants fonctionnels et techniques du système est **UML**.

Nous avons utilisé plusieurs types de diagrammes UML dans le cadre de ce projet :

- **Diagramme de cas d'utilisation** : pour identifier les acteurs et leurs interactions avec le système.
- **Diagramme de séquence** : pour illustrer le déroulement des échanges entre les acteurs et les composants du système dans un scénario donné.
- **Diagramme d'activité** : pour modéliser le déroulement logique des opérations, notamment les processus métier.
- **Diagramme de classes** : pour représenter la structure statique du système, les entités, leurs attributs et leurs relations.

Le langage UML a été préféré à la méthode MERISE, car il est **plus adapté aux architectures orientées objet** et à des projets modernes basés sur des technologies web telles que Django ou Next.js.

b. Outils de modélisation : Draw.io

Pour la réalisation des différents diagrammes UML, l'outil utilisé est **Draw.io** (aussi connu sous le nom de **diagrams.net**), un outil de modélisation graphique en ligne. Le choix de Draw.io repose sur plusieurs critères :

- Il est **gratuit et accessible en ligne**, sans installation.
- Il propose une large **bibliothèque de formes UML**, parfaitement adaptées aux standards de modélisation.
- Il permet l'**exportation directe des diagrammes** sous plusieurs formats (PNG, PDF, SVG) pour une intégration fluide dans le rapport ou les présentations.
- Il est **intuitif et rapide à prendre en main**, même pour les débutants.

Ainsi, tous les schémas intégrés au rapport (cas d'utilisation, séquence, activité, classe) ont été **réalisés sur Draw.io** à partir du langage UML.

Chapitre 2 : Étude détaillée de la solution

I. Étude fonctionnelle

L'étude fonctionnelle vise à modéliser de manière claire les besoins exprimés par les utilisateurs, à travers l'identification des fonctions principales que doit accomplir le système. Elle constitue la base de la conception et du développement du logiciel.

1. Diagramme de contexte statique (description)

Le **diagramme de contexte statique** présente une vue globale du système étudié, en mettant en évidence les interactions entre ce système et ses différents acteurs externes (appelés *entités externes*).

Dans notre cas, le **système de gestion scolaire** est au centre, et les principaux acteurs interagissant avec lui sont :

- **Administrateur**
- **Enseignant**
- **Parent/Tuteur**
- **Élève**
- **Comptable**

Chaque acteur échange des flux d'informations avec le système :

- Par exemple, l'administrateur fournit des données d'inscription, récupère des rapports.
- Le parent consulte les bulletins ou le statut des paiements.
- L'enseignant transmet les notes.

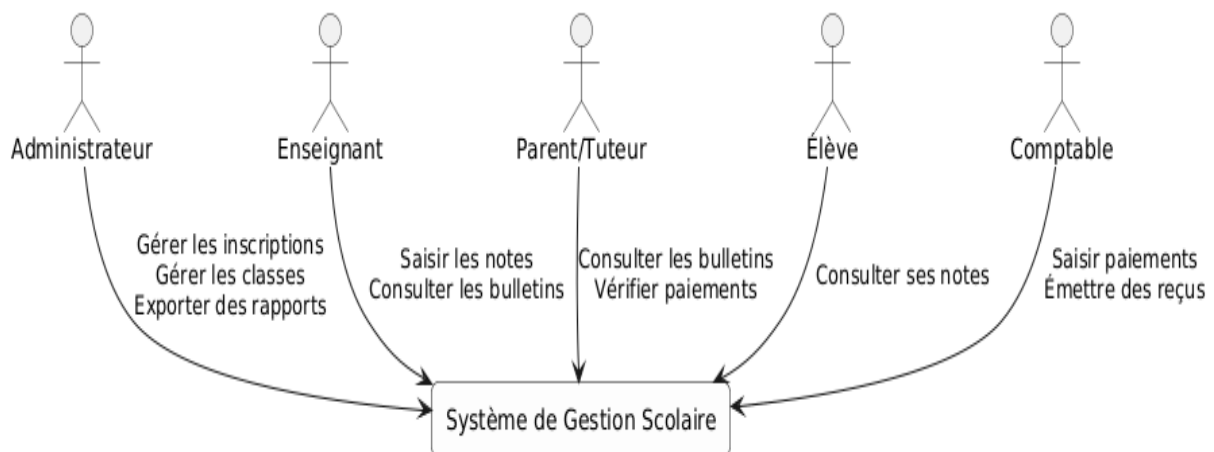


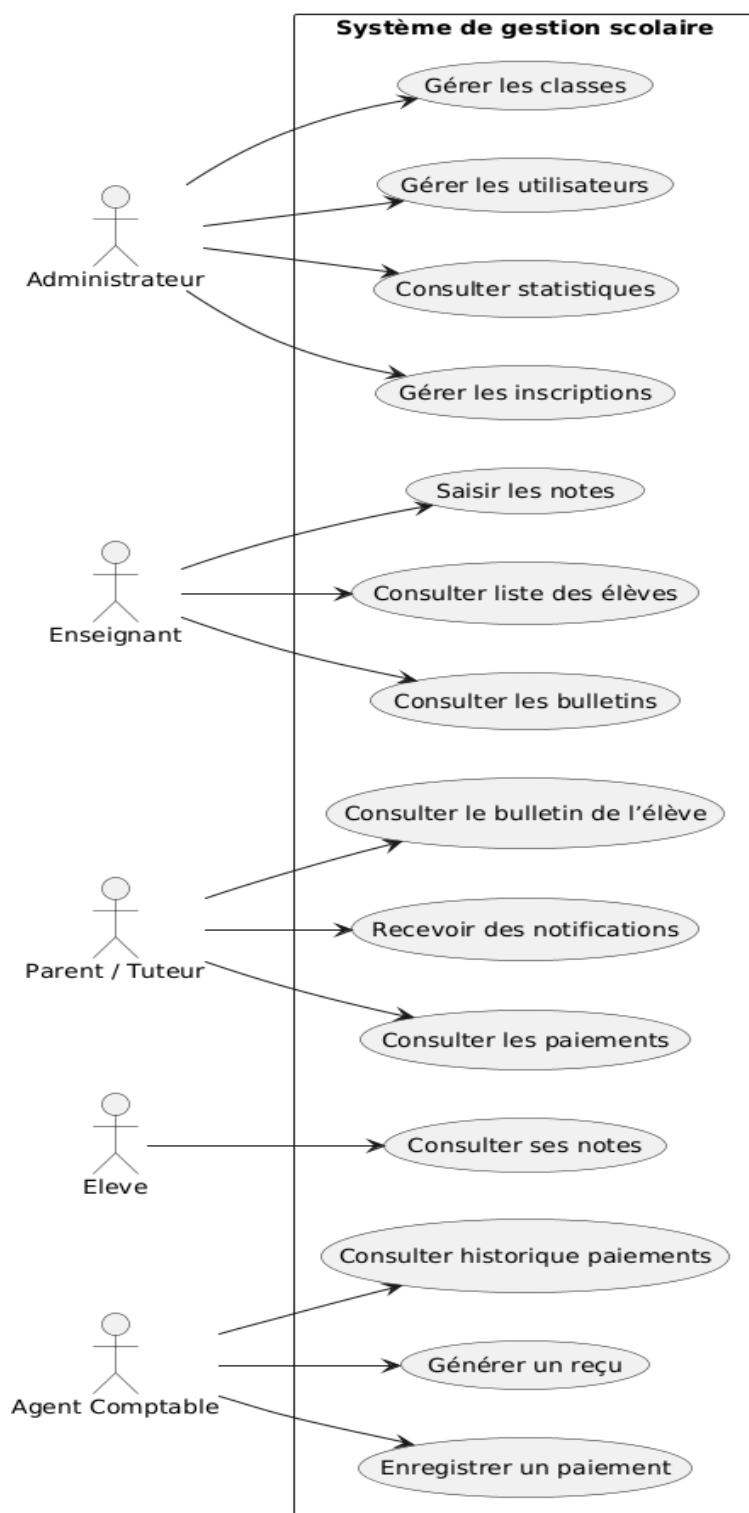
Fig 5 : Diagramme de contexte statique

2. Identification et répartition des cas d'utilisation (CU)

Les **cas d'utilisation** représentent les fonctionnalités principales attendues du système. Ils sont répartis par acteur comme suit :

Tab 2 : Cas d'utilisation

Acteur	Cas d'utilisation (CU)
Administrateur	<ul style="list-style-type: none"> - Gérer les utilisateurs - Enregistrer une inscription - Gérer les classes - Consulter les statistiques
Enseignant	<ul style="list-style-type: none"> - Saisir les notes - Consulter les bulletins - Consulter la liste des élèves
Parent/Tuteur	<ul style="list-style-type: none"> - Consulter le bulletin de l'élève - Consulter les paiements - Recevoir des alertes
Élève (optionnel)	<ul style="list-style-type: none"> - Consulter ses propres résultats - Mettre à jour ses informations personnelles
Agent comptable	<ul style="list-style-type: none"> - Enregistrer un paiement - Générer un reçu - Consulter l'historique des paiements

a. Diagramme de cas d'utilisation*Fig 6 : Diagramme de cas d'utilisation*

b. Diagramme de séquence

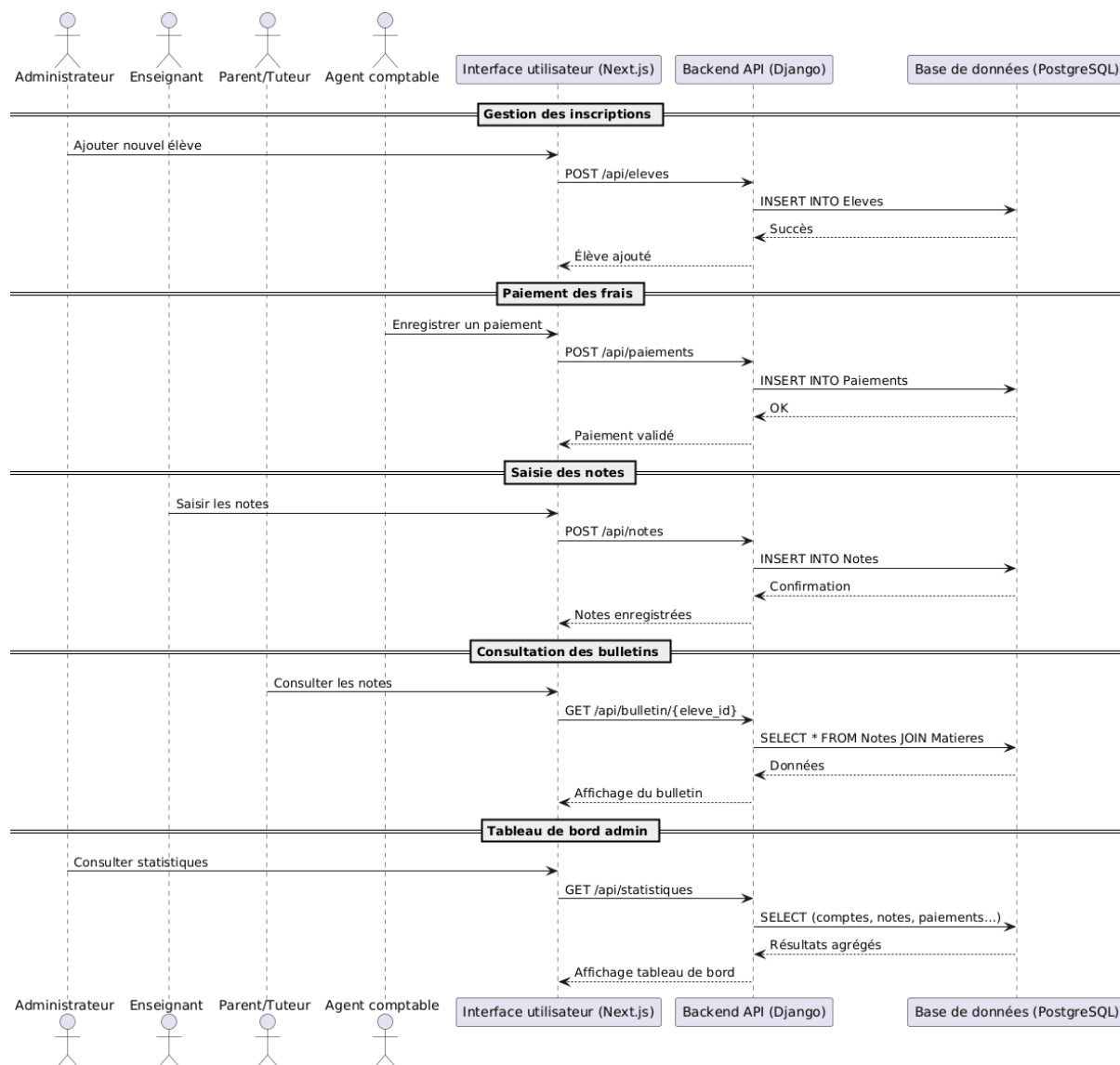


Fig 7 : Diagramme de séquence

c. Diagramme d'activité

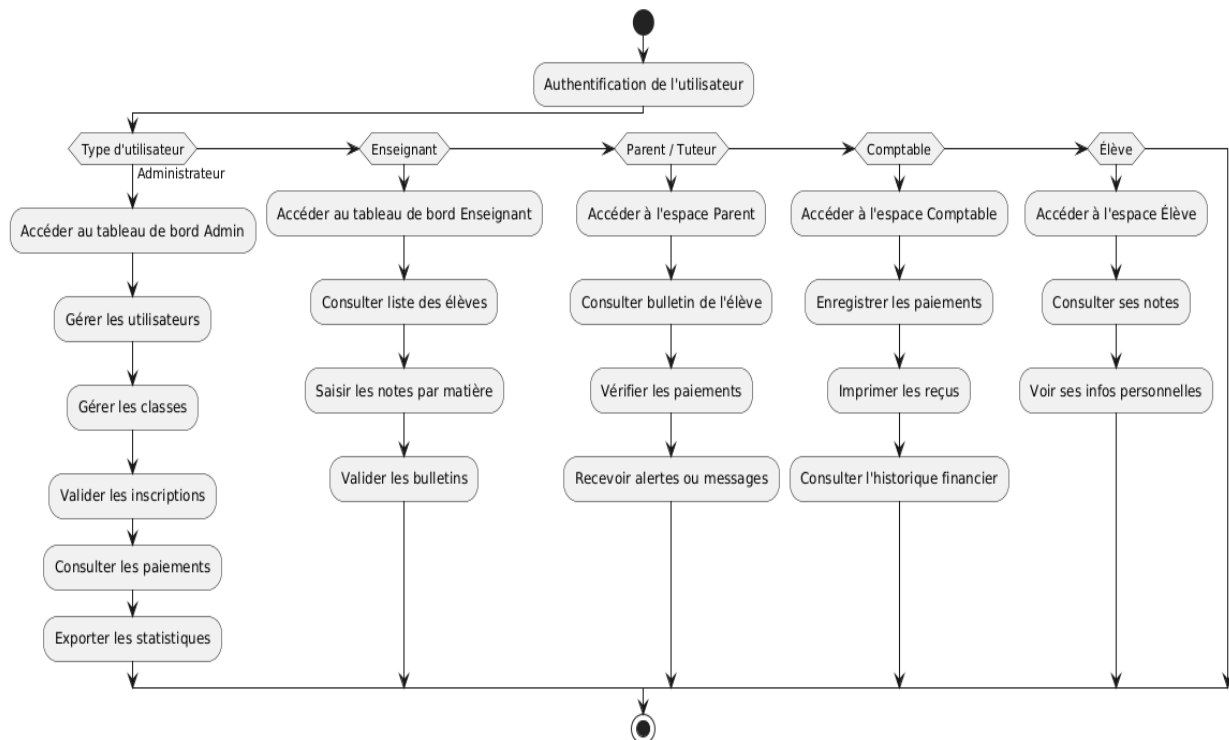


Fig 8 : diagramme d'activité

Description des cas d'utilisation (exemples)

CU01 : Gérer les utilisateurs

- Acteur principal : Administrateur
- Description : Ajouter, modifier ou supprimer des utilisateurs (enseignants, parents, agents).
- Préconditions : L'administrateur est connecté.
- Résultat attendu : Le compte utilisateur est mis à jour dans le système.

CU02 : Saisir les notes

- Acteur principal : Enseignant
- Description : L'enseignant accède à sa liste de classes et saisit les notes par matière et par trimestre.
- Résultat attendu : Les notes sont enregistrées et accessibles pour les bulletins.

CU03 : Enregistrer un paiement

- Acteur principal : Agent comptable
- Description : Enregistrement du montant, de la date, du type de paiement pour un élève donné.

- Résultat attendu : Mise à jour du solde et émission d'un reçu.

CU04 : Consulter le bulletin

- Acteur principal : Parent ou Élève
- Description : Accès sécurisé aux résultats par trimestre.
- Résultat attendu : Affichage lisible des notes et moyennes générales.

Références aux diagrammes

Les diagrammes suivants sont utilisés pour illustrer les interactions du système :

- **Diagrammes de cas d'utilisation** : Présentent visuellement les relations entre acteurs et cas d'utilisation.
- **Diagrammes de séquence** : Illustrent l'ordre d'échange des messages entre un acteur et le système dans un scénario donné (ex. : saisie de note, consultation de bulletin).
- **Diagrammes d'activité** : Représentent graphiquement le déroulement logique d'un processus métier (ex. : processus d'inscription, de paiement ou d'émission d'un bulletin).

3. Interfaces prévues

Les interfaces ont été pensées de manière intuitive et segmentée par rôle. Un menu dynamique s'adapte selon les droits d'accès, garantissant une bonne ergonomie pour chaque utilisateur.

II. Étude technique

L'étude technique consiste à détailler les choix technologiques, les composants de l'architecture du système, ainsi que les mécanismes de sécurité et d'optimisation mis en œuvre. Ces choix ont été guidés par les exigences fonctionnelles du projet, les contraintes techniques, ainsi que les bonnes pratiques de développement d'applications web modernes.

1. Architecture générale

Le système repose sur une **architecture client-serveur** respectant le modèle **RESTful**, favorisant la séparation des responsabilités, la maintenabilité et la réutilisabilité du code. Cette architecture est structurée en trois couches principales :

a. Couche Frontend – Interface utilisateur

L'interface utilisateur a été développée avec **Next.js**, un framework basé sur React. Ce choix se justifie par plusieurs avantages :

- **Rendu côté serveur (Server-Side Rendering)** : améliore le chargement initial des pages et le référencement naturel.
- **Modularité et réutilisabilité des composants React**.
- **Performances élevées** grâce à la génération statique (Static Generation) et au pré-chargement des routes.
- **Support TypeScript** pour un code plus robuste et maintenable.
- Interface responsive, compatible avec ordinateurs, tablettes et smartphones.

b. Couche Backend – Logique métier et API

Le serveur backend est développé en **Django**, un framework Python réputé pour sa robustesse, sa sécurité et sa rapidité de développement. Il est couplé à **Django REST Framework (DRF)** pour exposer des **API RESTful** claires et sécurisées.

- Django gère la logique métier, les relations entre les entités (élève, classe, paiement...).
- DRF permet de construire rapidement des endpoints REST avec validation des données, sérialisation et permissions.

c. Couche Base de données – Stockage des informations

Le système utilise **PostgreSQL** comme SGBD (Système de Gestion de Base de Données) relationnel. Ce choix s'explique par :

- Sa **puissance de traitement des relations complexes** (jointures, contraintes d'intégrité).
- Son **support de transactions** pour assurer la cohérence des données.
- Sa **compatibilité avec Django ORM** pour faciliter les requêtes.

d. Hébergement

Deux modes de déploiement sont envisagés :

- **Local**, pour les établissements ayant une infrastructure réseau interne.
- **Cloud** (Railway, Render, DigitalOcean...) pour bénéficier d'un hébergement sécurisé, évolutif et à haute disponibilité.

2. Sécurité

La sécurité étant cruciale dans la gestion de données scolaires (informations personnelles, notes, paiements), plusieurs mesures techniques ont été mises en œuvre :

- **Authentification par tokens JWT** : chaque utilisateur se connecte via un identifiant unique et reçoit un jeton sécurisé pour interagir avec l'API.
- **Gestion des rôles et permissions** : les droits d'accès sont strictement définis selon le profil (admin, enseignant, parent...).
- **Chiffrement des données sensibles** :
 - Les mots de passe sont **hachés avec bcrypt** (et non stockés en clair).
 - Les communications client-serveur sont sécurisées via **HTTPS/TLS**.
- **Protection contre les attaques web** :
 - Django ORM protège contre les **injections SQL**.
 - Validation stricte des entrées utilisateur côté client et serveur pour éviter les attaques XSS/CSRF.
- **Journalisation des actions critiques** : chaque action sensible (suppression, modification de note, paiement...) est enregistrée dans des **logs sécurisés**.

3. Performances et optimisation

L'application a été pensée pour être **performante**, fluide et accessible à tous les types d'utilisateurs :

- **Pagination automatique** des listes (élèves, paiements, notes) pour éviter le chargement massif de données.
- **Filtres dynamiques** pour affiner les résultats de recherche selon la classe, le trimestre, le statut, etc.
- **Mise en cache** des requêtes fréquentes côté client et serveur pour accélérer les temps de réponse.
- **Minimisation des appels API** grâce à des composants optimisés côté frontend.
- **Adaptabilité (responsive design)** : l'interface s'ajuste automatiquement à toutes les tailles d'écrans via Tailwind CSS.
- **Surveillance des erreurs** avec des outils de débogage (logs Django, console navigateur, tests unitaires).

III. Diagramme de classe (UML) / Modèle Conceptuel de Données (MERISE)

Le projet étant structuré autour de données fortement liées, nous avons choisi de représenter son **modèle conceptuel de données (MCD)** selon une approche MERISE.

Exemple de MCD simplifié :

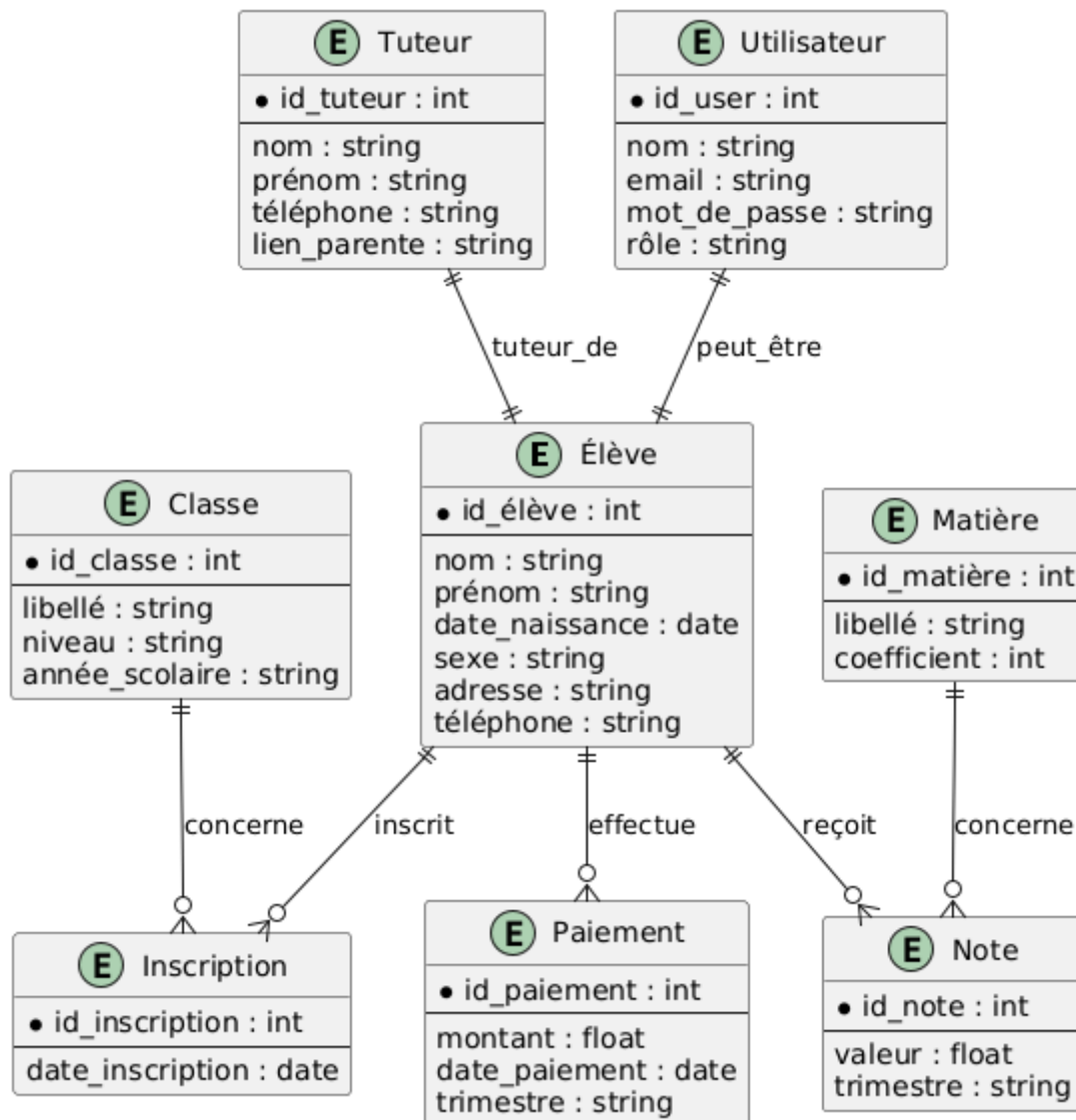


Fig 9 : Modèle conceptuel de données

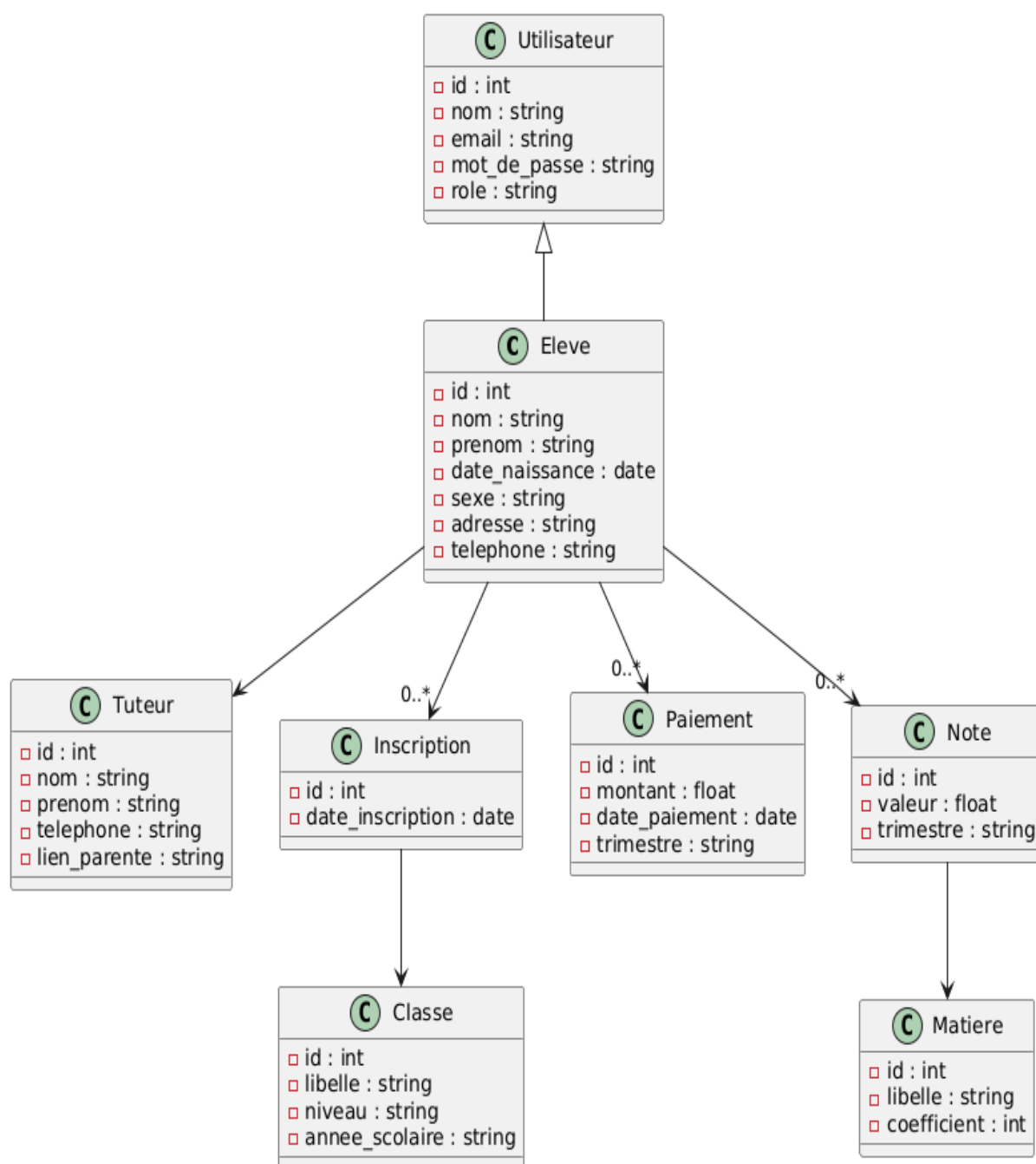
Exemple diagramme de classe :

Fig 10 : Diagramme de classe

Troisième Partie : Réalisation et mise en œuvre

Chapitre 1 : Mise en œuvre

I. Choix matériels

Pour garantir le bon déroulement du développement, des tests et des démonstrations de l'application de gestion automatisée des activités scolaires, un environnement matériel adéquat a été mis en place. Les choix matériels ont été guidés par la nécessité d'assurer à la fois performance, stabilité et accessibilité.

Matériel utilisé pendant le développement

- **Ordinateur portable personnel:**
 - **Modèle** : HP Pavilion
 - **Processeur** : Intel Core i5, 8^e génération, 2.4 GHz
 - **Mémoire vive (RAM)** : 12 Go
 - **Stockage** : SSD 512 Go
 - **Système d'exploitation**: Windows 10 Pro (64 bits)
- **Équipements supplémentaires:**
 - Connexion Internet stable (fibre optique avec débit ≥ 20 Mbps)
 - Clé USB / Disque dur externe pour la sauvegarde régulière du projet
 - Souris, casque audio, smartphone pour les tests responsives

Ce matériel a permis de développer et tester l'application dans des conditions similaires à celles rencontrées dans un environnement professionnel.

II. Choix logiciels

Le choix des outils logiciels a été fait de manière stratégique pour garantir la fiabilité, la maintenabilité et l'évolutivité du système. Le projet repose principalement sur des technologies web modernes, avec une architecture séparée entre le frontend (interface utilisateur) et le backend (logique métier et base de données).

Outils de développement et de programmation

Tab 3 : outils de développement et de programmation

Logiciel / Technologie	Rôle dans le projet	Remarques
Next.js	Développement de l'interface utilisateur (frontend)	Framework React.js permettant le SSR (Server Side Rendering), performant et SEO-friendly
Django	Développement du backend (API REST)	Framework Python robuste, modulaire et sécurisé
PostgreSQL	Système de gestion de base de données (SGBD)	Base relationnelle open-source fiable et puissante
Visual Studio Code	Environnement de développement (IDE)	Léger, extensible avec de nombreuses extensions
Postman	Test des requêtes API	Simule les appels HTTP pour vérifier les endpoints
Draw.io	Création des diagrammes (UML, MCD, etc.)	Outil gratuit et intuitif pour la modélisation visuelle
Git / GitHub	Gestion de version du code source	Suivi de l'évolution du code, collaboration et sauvegarde
Figma (<i>optionnel</i>)	Prototypage UI avant développement	Utilisé pour simuler certaines interfaces clés

Chapitre 2 : Présentation de l'application

I. Présentation générale de l'application

L'application web réalisée vise à **numériser et centraliser** la gestion des activités administratives et pédagogiques d'un établissement secondaire. Elle répond aux limites de la gestion manuelle traditionnelle, tout en s'adaptant aux contraintes et aux réalités du système éducatif local.

L'application couvre principalement trois modules fonctionnels :

- **Gestion des inscriptions** : les responsables peuvent inscrire de nouveaux élèves, modifier leurs informations, et visualiser l'historique des inscriptions par année scolaire.
- **Gestion des frais de scolarité** : enregistrement des paiements par élève et génération d'états financiers par classe, trimestre ou année scolaire.
- **Gestion des évaluations** : saisie des notes par matière, trimestre et élève, avec calcul automatique de la moyenne et génération de bulletins.

L'application est accessible via un navigateur web, depuis n'importe quel terminal (ordinateur, tablette, smartphone), sans installation locale. Cela permet une grande **portabilité** et une adoption facile par les utilisateurs.

Chaque utilisateur dispose d'un **compte sécurisé** et voit des interfaces et fonctionnalités différentes selon son rôle : administrateur, enseignant, parent ou élève.

II. Architecture de l'application

L'architecture adoptée est de type **client-serveur**, avec séparation stricte entre :

- Le **frontend** (interface utilisateur)
- Le **backend** (traitement des données et logique métier)
- La **base de données** (stockage des informations)

1. Frontend – Interface utilisateur (Next.js)

Développé avec **Next.js**, un framework basé sur React, le frontend assure :

- Un rendu côté serveur (SSR) pour de meilleures performances
- Un design responsive avec Tailwind CSS
- Une navigation fluide entre les pages (sans rechargement complet)
- Une interface claire, intuitive et adaptée à chaque profil utilisateur

Quelques exemples de composants développés :

- Formulaire d'inscription dynamique
- Tableau de bord avec statistiques en temps réel
- Modules de saisie de notes avec contrôle des trimestres
- Interfaces de paiement avec génération de reçus

2. Backend – API sécurisée (Django + DRF)

Le backend est basé sur le framework **Django**, avec l'extension **Django REST Framework** pour exposer des API RESTful.

Fonctionnalités principales :

- Authentification et gestion des rôles
- Création/modification/suppression des élèves, paiements, matières, notes, etc.
- Protection des routes sensibles avec des permissions personnalisées
- Validation des données avant enregistrement
- Génération d'exports (PDF, Excel) à venir

3. Base de données relationnelle (PostgreSQL)

La base de données est centralisée et normalisée. Elle garantit l'intégrité des données, évite les doublons et permet des requêtes puissantes.

- Relations 1-N et N-N (ex. : un élève peut être inscrit plusieurs fois)
- Indexation des clés primaires et étrangères
- Sauvegardes régulières prévues via pgAdmin

Quatrième Partie : Exploitation et Utilisation

Chapitre 1 : Configuration matérielle et logicielle

Ce chapitre a pour objectif de présenter les éléments nécessaires au bon fonctionnement de l'application développée, tant du point de vue matériel que logiciel. Il est crucial de garantir que l'environnement d'exécution est cohérent avec les spécifications techniques de l'application pour assurer sa stabilité, ses performances et sa sécurité. Il aborde également le processus de déploiement et le suivi post-mise en service.

I. Configuration matérielle

L'application étant une solution web, elle ne nécessite pas d'installation physique chez chaque utilisateur, ce qui constitue un grand avantage en termes de coût et de maintenance. Néanmoins, certains équipements sont nécessaires pour le bon fonctionnement côté serveur et pour une expérience utilisateur fluide côté client.

Côté client (utilisateurs finaux)

Les utilisateurs finaux (administrateurs, enseignants, parents) peuvent accéder à la plateforme depuis un simple navigateur web moderne (Chrome, Firefox, Edge). Voici les prérequis recommandés pour une navigation optimale :

Tab 4 : Matériel Côté client

Équipement	Configuration minimale recommandée
Processeur	Dual-core 1.8 GHz ou plus
RAM	4 Go minimum
Stockage	100 Mo libres (cache et stockage local)
Système d'exploitation	Windows 10+, macOS, Linux ou Android/iOS récent
Connexion Internet	1 Mbps minimum (idéalement 3 Mbps ou plus)
Navigateur Web	Chrome v90+, Firefox v85+, Edge v90+

Côté serveur (hébergement de l'application)

Le backend (Django/PostgreSQL) et le frontend (Next.js) doivent être hébergés sur un serveur capable de gérer les requêtes API et le rendu côté serveur. En cas de déploiement sur une plateforme cloud (Render, Railway, Heroku, etc.), voici les prérequis typiques :

Tab 5 : Matériel Cote serveur

Composant	Configuration recommandée
Processeur	2 vCPU minimum
RAM	4 Go minimum
Stockage	10 Go minimum
Base de données	PostgreSQL 14+
OS serveur	Ubuntu 20.04+ ou Debian stable
Sécurité réseau	Pare-feu activé, ports ouverts : 80 (HTTP), 443 (HTTPS), 5432 (PostgreSQL)

II. Configuration logicielle

Le choix des outils logiciels utilisés repose sur leur stabilité, leur documentation riche, leur compatibilité entre eux et leur communauté active.

Technologies utilisées dans le projet

Tab 6 : Technologies utilisées dans le projet

Composant	Technologie	Rôle
Frontend	Next.js (v13+)	Interface utilisateur
Backend	Django (v4.2) + DRF	API et logique métier
Base de données	PostgreSQL (v14+)	Stockage relationnel des données
ORM	Django ORM	Requêtes en base de données
Sécurité	JWT (via djangorestframework-simplejwt)	Authentification sécurisée
Déploiement (test)	Render.com	Hébergement du backend et frontend
Gestion de projet	Git + GitHub	Suivi des versions et collaboration
Documentation	Markdown + PDF	Présentation des APIs, modèles et interfaces

Navigateur et compatibilité

L'interface a été testée avec succès sur :

- Google Chrome (Windows, Android)
- Mozilla Firefox (Windows, Linux)
- Microsoft Edge
- Safari (macOS et iOS)

Elle reste responsive et s'adapte automatiquement à tous les formats d'écran (ordinateur, tablette, smartphone).

Conclusion du chapitre

La réussite d'un projet informatique ne dépend pas uniquement du développement, mais aussi de sa capacité à être déployé, maintenu et utilisé efficacement. Ce chapitre a montré que l'application dispose d'un environnement technique solide, accessible et adaptable, garantissant sa stabilité dans le temps. La rigueur apportée au choix des outils et à la configuration garantit non seulement un bon démarrage, mais aussi une durabilité du système en environnement réel.

Chapitre 2 : Utilisation

I. Plan de navigation

Le plan de navigation représente la structure logique de l'application, c'est-à-dire la manière dont les utilisateurs peuvent accéder aux différentes fonctionnalités via l'interface.

L'application a été conçue de manière intuitive pour s'adapter aux profils des utilisateurs (administrateurs, enseignants, parents/élèves), avec une navigation claire, fluide et responsive.

Structure générale de la navigation

Tab 7 : *Structure générale de la navigation*

Page	Rôle / Description
Accueil (Landing page)	Page d'accueil avec présentation de l'école et de l'application
Connexion / Inscription	Authentification selon le profil utilisateur (admin, parent, enseignant)
Tableau de bord	Vue personnalisée selon le rôle : tableau de bord admin, enseignant ou parent
Élèves	Liste des élèves inscrits, formulaires d'inscription, édition de profil
Paielements	Suivi des paiements par élève / par classe / par trimestre
Évaluations	Saisie, consultation et génération de bulletins de notes
Rapports	Export des données (PDF, Excel) pour les paiements, inscriptions, résultats
Paramètres / Profil	Gestion du mot de passe, déconnexion, préférences utilisateur
Aide / Contact	Informations utiles, contact support, documentation utilisateur

Parcours utilisateur (exemples)

1. Administrateur :

- Se connecte → Tableau de bord admin
- Gère les inscriptions → Ajoute ou modifie un élève
- Suit les paiements → Consulte les paiements récents ou en attente
- Génère des rapports → Exporte les bulletins ou la liste des paiements

2. Enseignant :

- Se connecte → Tableau de bord enseignant
- Accède à sa liste de classes → Saisit les notes
- Consulte les moyennes → Génère les bulletins
- Transmet les résultats à l'administration

3. Parent / Élève :

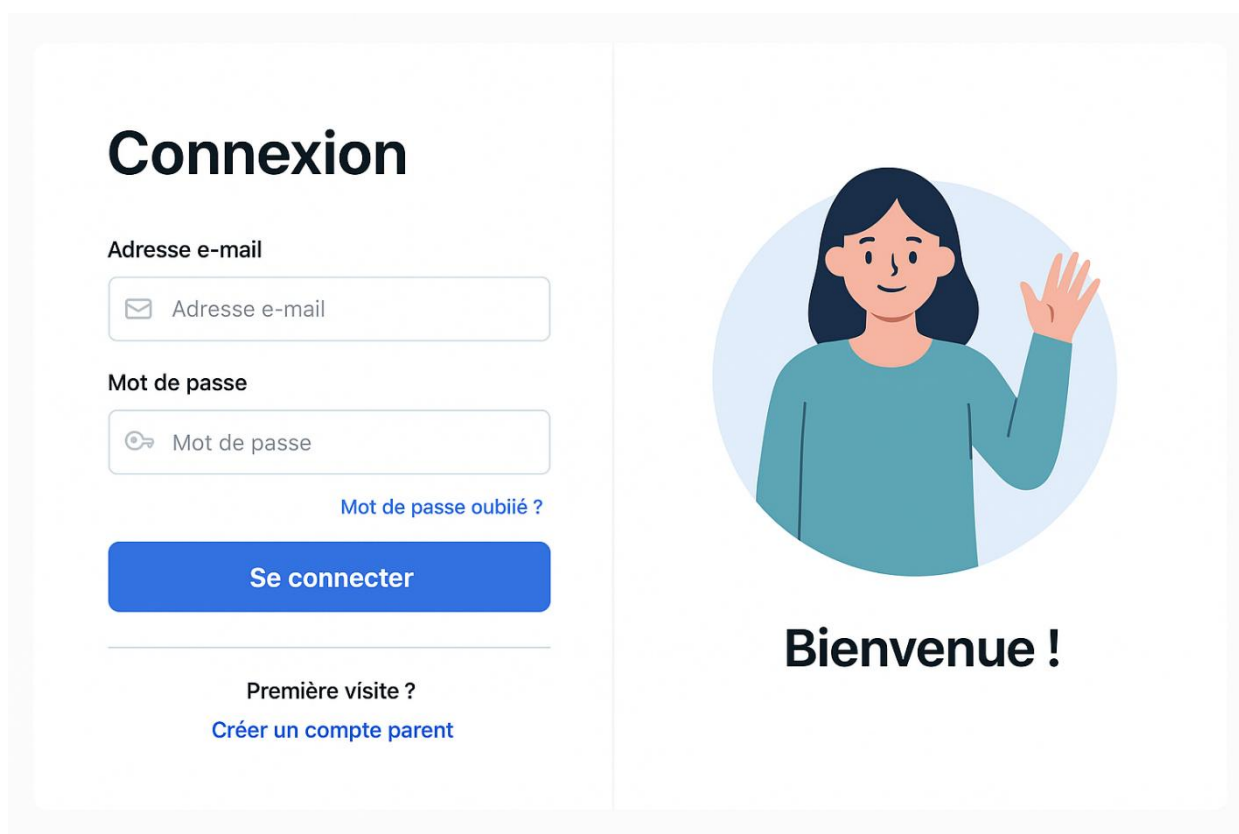
- Se connecte → Accède au profil élève
- Consulte l'état de paiement → Reçoit des alertes si retard
- Vérifie les résultats → Télécharge le bulletin du trimestre

II. Présentation des principales interfaces de l'application

Voici un aperçu des interfaces clés.

1. Page de connexion

- Champs : email, mot de passe
- Bouton de connexion
- Lien vers mot de passe oublié ou création de compte parent



The screenshot displays a login interface titled "Connexion". On the left side, there are two input fields: "Adresse e-mail" with an envelope icon and "Mot de passe" with a key icon. Below the password field is a link "Mot de passe oublié ?". A blue button labeled "Se connecter" is positioned below the fields. At the bottom, there is a link "Créer un compte parent" preceded by the text "Première visite ?". The right side of the interface features a circular illustration of a woman with dark hair waving, and the word "Bienvenue !" in a large, bold font.

Fig 11 : Page de connexion

2. Tableau de bord (personnalisé)

- Cartes dynamiques (nombre d'élèves, paiements en attente, moyennes)
- Liens rapides vers les actions fréquentes (ajouter élève, voir paiements)

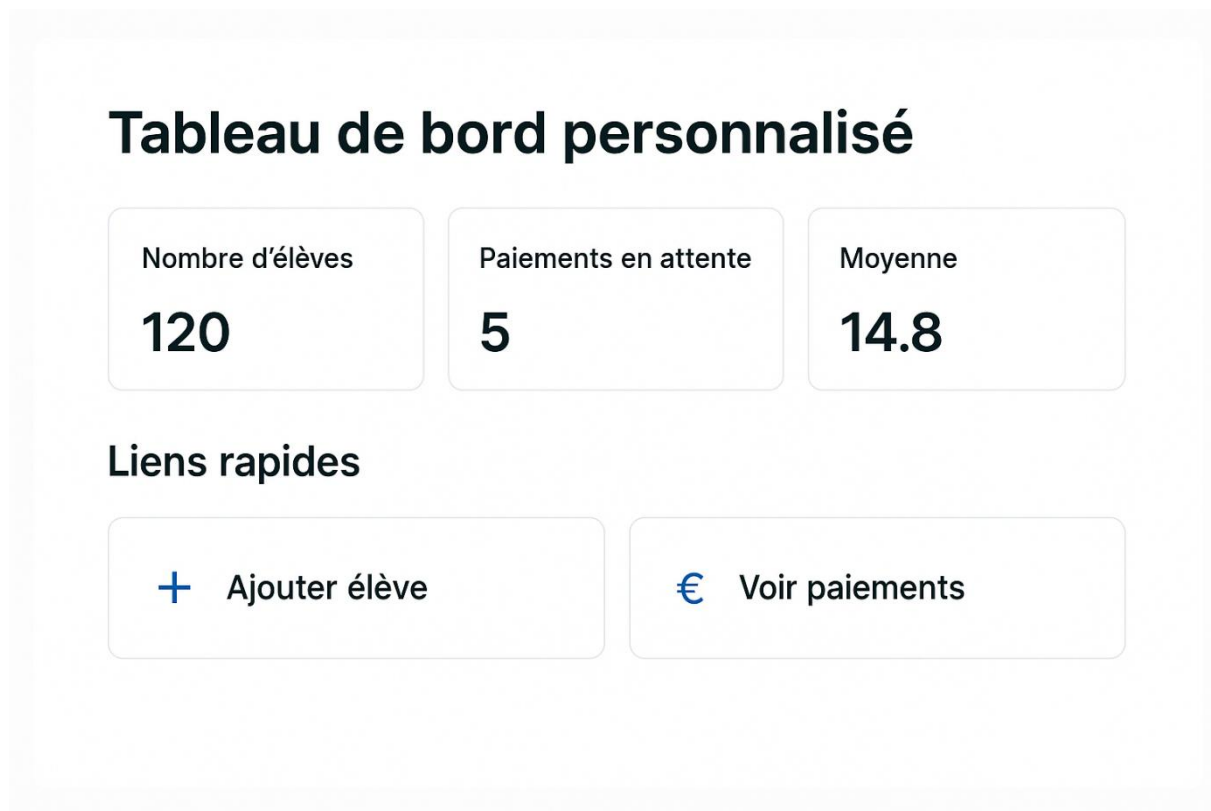


Fig 12 : Tableau de bord

3. Page d'inscription

- Formulaire avec champs : nom, prénom, sexe, classe, tuteur, etc.
- Upload de photo / pièce
- Validation côté serveur

Créer un compte parent

Nom

Adresse e-mail

Adresse e-mail

Mot de passe

Confirmer le mot de passe

Confirmer le mot de passe

S'inscrire

Déjà un compte ? [Se connecter](#)

Fig 13 : Page d'inscription

4. Paiements

- Historique des paiements
- Statut (payé, partiel, en attente)
- Filtrage par élève / classe / trimestre

Tableau de bord de gestion scolaire			
Paiements			
Étudiant	Classe	Trimestre	
Nom de l'étudiant	Montant	Date	Statut
Antoine Lefevre	250 €	25/08/2021	Payé
Claire Marchand	150 €	28/08/2021	Partiel
François Dubois	350 €	30/03/2022	En attente
Julie Bernard	350 €	24/05/2022	Payé
Lucas Moreau	350 €	30/06/2022	Partiel
Marie Lambert	250 €	23/06/2022	En attente
Nicolas Caron	250 €	30/06/2022	Payé

Fig 14 : Historique de paiement

5. Gestion des notes

- Liste des élèves par matière
- Saisie rapide des notes par trimestre
- Calcul automatique des moyennes

6. Génération de bulletins

- Sélection de la classe et du trimestre
- Affichage des résultats
- Export PDF

7. Paramètres utilisateur

- Modification de mot de passe
- Préférences d'affichage

Conclusion du chapitre

Ce chapitre a permis de mettre en évidence la simplicité et la richesse fonctionnelle de l'application développée. Grâce à une navigation intuitive et à des interfaces ergonomiques, chaque utilisateur dispose d'un accès personnalisé aux données dont il a besoin. L'intégration des états imprimables et exportables renforce l'utilité administrative du système, en facilitant les prises de décision, le suivi pédagogique et la transparence des opérations.

CONCLUSION GENERAL

Au terme de cette mission, il apparaît clairement que le développement d'un système de gestion automatisée des activités d'une école secondaire constitue une solution efficace, adaptée et indispensable pour répondre aux besoins modernes des établissements scolaires.

Ce projet m'a permis de mobiliser un large éventail de compétences acquises au cours de ma formation, notamment en architecture logicielle, développement web (frontend avec Next.js, backend avec Django), gestion de bases de données avec PostgreSQL, ainsi qu'en analyse et modélisation des besoins. La réalisation du système a également favorisé une meilleure compréhension des enjeux de la digitalisation dans le domaine de l'éducation.

L'application mise en œuvre offre aujourd'hui une réponse concrète aux défis liés à la gestion manuelle des inscriptions, des paiements et des évaluations. Elle facilite la centralisation des données, la traçabilité des opérations, et renforce l'efficacité du travail administratif.

Cependant, comme tout projet technologique, certaines limites subsistent, notamment l'absence d'une application mobile native ou d'un système d'alerte en temps réel pour les parents. Ces éléments pourraient constituer des pistes d'amélioration dans les versions futures.

Bibliographie

- ❖ Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7e éd.). McGraw-Hill.
→ Référence incontournable dans la conception et le développement logiciel.
- ❖ Sommerville, I. (2011). *Software Engineering* (9e éd.). Pearson Education.
→ Un ouvrage de référence pour comprendre le cycle de vie d'un logiciel.
- ❖ Faidherbe, L. (2007). *Méthode MERISE : Analyse et Conception* (3e éd.). Dunod.
→ Utilisé pour la modélisation du système via la méthode MERISE.
- ❖ Tavares, M. (2014). *Gestion de projet informatique*. Eyrolles.
→ Apports sur les plannings, l'évaluation des charges et les méthodes de suivi de projet.

Webographie

- ❖ Django Software Foundation. (2024). *Django Web Framework*. Disponible sur : <https://www.djangoproject.com>
→ Documentation officielle utilisée pour le backend de l'application.
- ❖ Vercel Inc. (2024). *Next.js Documentation*. Disponible sur : <https://nextjs.org>
→ Documentation officielle utilisée pour le frontend (React + Next.js).
- ❖ PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*. Disponible sur : <https://www.postgresql.org>
→ Documentation de référence pour la base de données relationnelle.
- ❖ Mozilla. (2024). *MDN Web Docs*. Disponible sur : <https://developer.mozilla.org>
→ Référence complète pour HTML, CSS, JavaScript.
- ❖ OpenClassrooms. (2024). *Cours et tutoriels pour développeurs*. Disponible sur : <https://openclassrooms.com>
→ Apports sur les bonnes pratiques de développement web.
- ❖ W3Schools. (2024). *Web Development Tutorials*. Disponible sur : <https://www.w3schools.com>
→ Tutoriels pratiques utilisés pour des éléments HTML et CSS.
- ❖ draw.io (2024). *Diagram Editor*. Disponible sur : <https://drawio.app>
→ Outil utilisé pour la création des diagrammes UML.

ANNEXES

Annexe I – Modèle Conceptuel de Données (MCD)

Représentation textuelle du MCD utilisé pour structurer les données dans la base PostgreSQL.

Élève(id, nom, prénom, date_naissance, sexe, adresse, téléphone, tuteur_id)
Classe(id, nom_classe, niveau, annee_scolaire)
Inscription(id, eleve_id, classe_id, date_inscription)
Paiement(id, eleve_id, montant, date_paiement, trimestre)
Note(id, eleve_id, matiere_id, valeur, trimestre)
Matière(id, libelle, coefficient)
Utilisateur(id, nom, email, rôle, mot_de_passe)

Annexe II – Diagrammes UML

Diagrammes produits pour représenter les différents aspects du système :

- Diagramme de cas d'utilisation
- Diagrammes de séquence (Admin, Enseignant, Parent/Élève)
- Diagramme d'activité
- Diagramme de classes

Ces diagrammes ont été générés avec l'outil Draw.io.

Annexe III – Interfaces de l'application

Captures d'écran ou maquettes des principales interfaces développées dans le projet :

- Page de connexion
- Tableau de bord
- Gestion des élèves
- Paiements
- Saisie des notes
- Résultats / Relevés
- Profil utilisateur

Annexe IV – Extraits de code source

Quelques extraits significatifs du code utilisé dans le projet :

- Modèles Django : fichier models.py
- API Views Django : fichier views.py
- Interface Next.js : fichier Dashboard.tsx
- Fichier de configuration environnementale : .env (version anonymisée)

Annexe V – Planning prévisionnel vs réalisé

Étape	Prévision	Réalisation	Commentaire
Étude de l'existant	5 jours	4 jours	Conforme
Analyse & Conception	7 jours	9 jours	Ajustement des exigences
Développement Frontend	10 jours	12 jours	Interface responsive plus longue
Développement Backend	10 jours	8 jours	Django bien maîtrisé
Tests & Déploiement	5 jours	5 jours	Conforme

Annexe VI – Autres documents utiles

Cette section contient des éléments supplémentaires liés au projet :

- Plan de localisation des pôles ESGIS
- Attestation fictive de stage
- Captures d'écrans de démonstration
- Résultats ou observations diverses

Table des matières

Remerciements	i
SOMMAIRE	ii
Résumé	iv
Abstract	v
Glossaire	vi
Liste des figures	vii
Liste des tableaux	viii
Liste des participants au projet	ix
Introduction	1
Première Partie : Cahier des charges	2
Chapitre 1 : Présentations	3
I. Présentation de l'ESGIS	4
Conclusion du chapitre	8
Chapitre 2 : Thème du stage	10
I- Présentation du sujet	11
II- Problématique du sujet	12
III- Intérêt du sujet	13
IV- Étude de l'existant	15
V- Critique de l'existant	16
Conclusion de l'analyse critique	17
Conclusion du chapitre	18
Chapitre 3 : Proposition de solutions	20
I. Spécification de la solution	21
II. Propositions de solution	24
Chapitre 4 : Choix de la solution.....	27
I. Solution retenue	28
III. Planning de réalisation	29
Conclusion au chapitre	31
Deuxième Partie : Analyse et Conception	33
Chapitre 1 : Présentation des outils d'analyse et de modélisation.....	34
I. Présentation de la méthode d'analyse	35
II. Langage et outils de modélisation utilisés	36
Chapitre 2 : Étude détaillée de la solution	37
I. Étude fonctionnelle	38

II. Étude technique	44
III. Diagramme de classe (UML) / Modèle Conceptuel de Données (MERISE).....	47
Troisième Partie : Réalisation et mise en œuvre.....	49
Chapitre 1 : Mise en œuvre.....	50
I. Choix matériels	51
II. Choix logiciels	52
Chapitre 2 : Présentation de l'application.....	53
I. Présentation générale de l'application.....	54
II. Architecture de l'application	55
Quatrième Partie : Exploitation et Utilisation.....	57
Chapitre 1 : Configuration matérielle et logicielle	58
I. Configuration matérielle	60
II. Configuration logicielle.....	62
Conclusion du chapitre	67
Chapitre 2 : Utilisation.....	69
I. Plan de navigation	70
II. Présentation des principales interfaces de l'application.....	72
1. Page de connexion.....	72
2. Tableau de bord (personnalisé).....	73
3. Page d'inscription.....	74
4. Paiements	75
5. Gestion des notes	75
6. Génération de bulletins	75
7. Paramètres utilisateur.....	76
CONCLUSION GENERAL	79
Bibliographie.....	81
Webographie.....	83
ANNEXES.....	85