

REPUBLIQUE DU BENIN

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE



ECOLE PIGIER BENIN

DOMAINE : SCIENCES ET TECHNOLOGIES

MENTION : INFORMATIQUE

SPECIALITE : LICENCE PROFESSIONNELLE EN RESEAUX ET GENIE
LOGICIEL

THÈME : Conception et réalisation d'une plateforme Web de
paiement de scolarité à PIGIER-BENIN

Réalisé par :

FADONOUGBO Espoir

&

GANDONOU Victoire

Sous la Direction de :

MAÎTRE DE STAGE

M. Godson MIDOHOUNGBE

Directeur Technique à SKYRAN

DIRECTEUR DE RECHERCHE

M. HOUNGBEDJI Koffi M.

Enseignant à PIGIER BENIN

Session de Juin 2024

ENGAGEMENT

L'Ecole PIGIER-BENIN n'entend donner aucune approbation ni improbation aux opinions émises dans ce mémoire.

Ces opinions doivent être considérées comme propres à leurs auteurs.

Sommaire

ENGAGEMENT	
<i>Sommaire.....</i>	<i>I</i>
<i>Dédicaces</i>	<i>III</i>
<i>Dédicaces</i>	<i>IV</i>
<i>Remerciements</i>	<i>V</i>
<i>Liste des figures</i>	<i>VI</i>
<i>Liste des tableaux.....</i>	<i>VIII</i>
<i>Sigles et abréviations.....</i>	<i>IX</i>
<i>Résumé</i>	<i>X</i>
<i>Abstract.....</i>	<i>X</i>
INTRODUCTION	- 1 -
Contexte et justification	- 1 -
Problématique.....	- 1 -
Objectifs du projet de fin de formation	- 2 -
Plan du document	- 2 -
Chapitre 1 : Déroulement du Stage	- 4 -
Introduction.....	- 4 -
1.1 Présentation de la structure	- 4 -
1.2 Fonctionnement de l'entreprise	- 5 -
1.3 Travaux effectués.....	- 6 -
1.4 Apports du stage sur le plan professionnel	- 6 -
1.5 Difficultés rencontrées	- 7 -
Conclusion	- 7 -
Chapitre 2 : Analyse et conception	- 8 -
Introduction.....	- 8 -

2.1	État de l'art	- 8 -
2.2	Présentation du projet :	- 15 -
2.3	Présentation des règles de gestion	- 16 -
2.4	Analyse UML	- 18 -
2.5	Logiciels et langages de programmation utilisés :	- 36 -
	Conclusion	- 46 -
	<i>Chapitre 3 : Résultats et Discussion</i>	- 46 -
	Introduction.....	- 46 -
3.1	Résultats	- 46 -
3.2	Discussions	- 55 -
	Conclusion	- 56 -
	<i>Conclusion générale et Perspectives</i>	- 56 -
	Perspectives	- 56 -
	<i>Références bibliographiques et Webographiques.....</i>	A
	Bibliographie.....	A
	Webographie	A
	<i>Table des Matières.....</i>	

Dédicaces

Dédicace 1

Je dédie ce mémoire :

- À mon père GANDONOU Magloire et à ma mère HOUNSA Thècle, qui m'ont toujours accompagnée et soutenue tout au long de ces trois années d'étude.
- À chacun des membres de ma famille qui m'ont accordé leur assistance continue.
- À toutes les personnes qui ont contribué à la réalisation de ce mémoire.

GANDONOU Victoire

Dédicaces

Dédicace 2

Je dédie ce mémoire

- À mon père FADONOUGBO Alexandre et à ma mère DANSOU Gisèle, qui m'ont encouragé et assisté tout au long de ces trois années d'étude universitaire.
- À chaque personne ayant contribué à la réalisation de ce mémoire.

FADONOUGBO Espoir

Remerciements

Nous tenons à remercier nos parents respectifs pour tous les efforts et le soutien, aussi bien moral que financier qu'ils nous ont accordé durant ces trois années d'étude.

Nous remercions également notre directeur de mémoire, monsieur HOUNGBEDJI Koffi M. qui a toujours su nous accorder du temps pour nous orienter, nous conseiller tout au long de la rédaction de ce mémoire.

Nous exprimons nos remerciements aux membres de l'administration de Pigier ainsi qu'aux professeurs qui nous ont offert un cadre d'apprentissage adéquat.

Enfin, nous remercions tous celles et ceux qui nous ont assisté tout au long de notre formation et durant la réalisation de ce projet.

Liste des figures

FIGURE 1: LOGO DE SKYRAN-GROUP	- 4 -
FIGURE 2: ORGANIGRAMME DE SKYRAN GROUP	- 6 -
FIGURE 3: PAGE D'ACCUEIL DE LA PLATEFORME PAIEMENT.TRESORBENIN.BJ	- 10 -
FIGURE 4: LOGO CAMPUS FASO	- 11 -
FIGURE 5: PAGE D'ACCUEIL DE LA PLATEFORME EDUS	- 12 -
FIGURE 6: PAGE D'ACCUEIL DE LA PLATEFORME SCHOOLMATIK	- 13 -
FIGURE 7: LOGO-UML	- 18 -
FIGURE 8: DIAGRAMME DE CAS D'UTILISATION GESTION DES ETUDIANTS	- 23 -
FIGURE 9: DIAGRAMME DE CLASSE	- 29 -
FIGURE 10: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « DEFINIR SES IDENTIFIANTS DE CONNEXION »	- 32 -
FIGURE 11: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « S'AUTHENTIFIER »	- 33 -
FIGURE 12: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « PAYER LA SCOLARITE »	- 34 -
FIGURE 13: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « RECHERCHER UNE QUITTANCE »	- 35 -
FIGURE 14: DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « CONSULTER L'HISTORIQUE DE PAIEMENT »	- 36 -
FIGURE 15: LOGO DE MYSQL	- 37 -
FIGURE 16: LOGO PHPMYADMIN	- 37 -
FIGURE 17: LOGO DE GITHUB	- 38 -
FIGURE 18 : LOGO-VISUAL STUDIO CODE	- 38 -
FIGURE 19: LOGO-DRAW.IO	- 39 -
FIGURE 20: LOGO-HTML	- 40 -
FIGURE 21: LOGO-CSS	- 40 -
FIGURE 22: LOGO-JAVASCRIPT	- 41 -
FIGURE 23: LOGO-PHP	- 42 -
FIGURE 24: LOGO DE LARAVEL	- 43 -
FIGURE 25: LOGO- BOOTSTRAP	- 45 -
FIGURE 26: FEDAPAY	- 46 -
FIGURE 27: PAGE D'ACCUEIL	- 46 -

FIGURE 28: FORMULAIRE DE CONFIGURATION DES IDENTIFIANTS DE CONNEXION	- 47 -
FIGURE 29: FONCTION-CONFIGURATION D'UN NOUVEAU MOT DE PASSE.....	- 48 -
FIGURE 30: FORMULAIRE DE CONNEXION	- 48 -
FIGURE 31: FONCTION GERANT LA CONNEXION D'UN ETUDIANT	- 49 -
FIGURE 32: PAGE DE RECAPITULATIF DES INFORMATIONS.....	- 50 -
FIGURE 33: FORMULAIRE DE PAIEMENT	- 51 -
FIGURE 34: FORMULAIRE DE DETAILS DE PAIEMENT	- 52 -
FIGURE 35: FORMULAIRE DE RECHERCHE D'UNE QUITTANCE	- 53 -
FIGURE 36:FORMULAIRE D'ENREGISTREMENT DES ETUDIANTS	- 54 -
FIGURE 37: LISTE DES ETUDIANTS	- 55 -

Liste des tableaux

TABLEAU 1:FORCE ET FAIBLESSE DE QUELQUE SOLUTIONS EXISTANTES	- 14 -
TABLEAU 2: DESCRIPTION DU CAS D'UTILISATION « DEFINIR SES IDENTIFIANTS DE CONNEXION »	- 24 -
TABLEAU 3 : DESCRIPTION DU CAS D'UTILISATION "S'AUTHENTIFIER"	- 25 -
TABLEAU 4:DESCRIPTION DU CAS D'UTILISATION "PAYER LA SCOLARITE".....	- 26 -
TABLEAU 5: DESCRIPTION DU CAS D'UTILISATION « RECHERCHER UNE QUITTANCE »...-	- 28 -

Sigles et abréviations

API : Application Programming Interface

CSS : Cascading Style Sheets

EPAC : Ecole Polytechnique d'Abomey Calavi

FASEG : Faculté Des Sciences Economiques et de Gestion

FASHS : Faculté des Sciences Humaines et Sociales

FLLAC : Faculté des Lettres, langues, Arts et Communications

HTML : HyperText Markup Language

HTTP : Hypertext Transfer Protocol

MLD : Modèle Logique de Données

ORM : Object-Relational Mapping

PHP : Hypertext Preprocessor

UAC : Université d'Abomey-Calavi

UML : Unified Modeling Language

Résumé

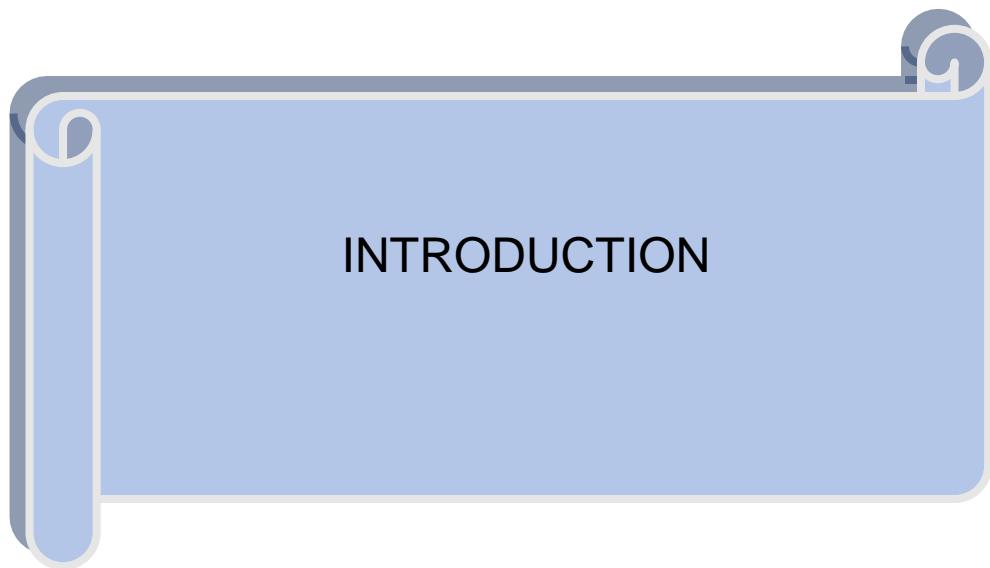
L'enseignement supérieur au Bénin a connu une croissance importante, avec une augmentation notable du nombre d'étudiants inscrits dans les universités au cours des dernières décennies. Malgré cette expansion, le paiement de scolarité par les étudiants dans les établissements privés, reste dominé par des processus manuels à l'origine de perte, et d'erreur. Ainsi, dans le cadre de notre projet de fin d'étude, nous avons, réalisé à l'aide du framework Laravel, une plateforme web visant à automatiser le processus de paiement des frais de scolarités. Notre solution permet principalement à un étudiant de payer sa scolarité en ligne, puis d'obtenir une quittance liée à son paiement. Ce dernier peut consulter l'historique de ses paiements. Le comptable quant à lui peut gérer les informations des étudiants, suivre aisément les différents paiements effectués, puis envoyer des emails de rappel aux étudiants en retard par rapport au paiement de leur scolarité.

Mots clés : plateforme, scolarité, paiement, université

Abstract

Higher education in Benin has experienced significant growth, with a notable increase in the number of students enrolled in universities over the past decades. Despite this expansion, tuition payments by students in private institutions are still dominated by manual processes, leading to loss and errors. Therefore, as part of our final year project, we have developed a web platform using the Laravel framework aimed at automating the tuition payment process. Our solution primarily allows a student to pay their tuition online and then obtain a receipt for their payment. The student can also view their payment history. As for the accountant, they can manage student information, easily track various payments made, and send reminder emails to students who are late in paying their tuition.

Keywords : platform, tuition, payment, university



INTRODUCTION

Contexte et justification

Au Bénin, l'enseignement supérieur connaît une expansion significative avec une augmentation remarquable du nombre d'étudiants inscrits dans les universités et les établissements d'enseignement supérieur au cours des dernières décennies. Selon les données du site Bénin Data Portal, le nombre total d'étudiants inscrits dans les universités publiques comme privées depuis 2022 serait de 123.562¹. Cette affluence massive de jeunes bacheliers vers les universités souligne l'urgence pour ces établissements de moderniser et d'automatiser leurs processus administratifs. Dans cette lancée, le gouvernement béninois a pris l'initiative de développer des plateformes visant à faciliter les processus d'inscription, de paiement de frais académiques et de demande de bourses dans les universités publiques du pays. Cependant, il est important de noter que de nombreuses universités privées n'ont pas encore bénéficié de cette automatisation, en particulier ce qui concerne le paiement des frais de scolarité par les étudiants. Cette insuffisance dans l'automatisation des processus administratifs dans les universités privées constitue un défi majeur, entraînant souvent des inefficacités, des retards dans le traitement des paiements. C'est donc de ce contexte qu'est né notre projet de fin d'étude. Dans celui-ci, nous nous concentrerons sur l'Ecole Pigier, une université privée du Bénin qui ne fait pas exception à ce manque d'automatisation en matière de paiement des frais de scolarité.

Problématique

La méthodologie administrative traditionnelle de paiement des frais de scolarité académiques se caractérise par des processus complexes et manuels, entraînant des retards et des erreurs. Parallèlement, les erreurs potentielles commises lors des transactions financières créent des inconvénients pour les étudiants, les parents et le personnel administratif. La sécurité des transactions émerge comme une

¹ Voir le numéro [1] dans la liste des références webographiques

préoccupation majeure, exigeant des mesures robustes pour protéger les informations financières des parties prenantes. Enfin, le suivi manuel des paiements et les rappels d'échéances posent des défis en termes d'efficacité et d'anticipation des besoins. Ainsi, la complexité administrative, l'accessibilité, la sécurité des transactions, et le suivi des paiements et rappels automatisés forment un ensemble de défis cruciaux auxquels les établissements éducatifs font face dans la gestion des frais de scolarité. La nécessité de mettre en place une plateforme de paiement de scolarité en ligne est essentiellement motivée par le besoin de surmonter ces problématiques complexes liées aux processus de paiements de scolarité en milieu universitaire.

Objectifs du projet de fin de formation

La solution proposée s'inscrit dans le cadre de notre projet de fin de formation. Ainsi, l'objectif principal de la plateforme de paiement de scolarité en ligne réside dans la transformation et la modernisation des processus administratifs associés aux frais de scolarité, visant ainsi à accroître l'efficacité opérationnelle des établissements éducatifs. Cette initiative cherche à réduire significativement les coûts administratifs tout en éliminant les erreurs et les pertes potentielles liées à la collecte manuelle des paiements. Parmi les objectifs spécifiques figure, la simplification du processus de paiement grâce à une interface conviviale, permettant aux étudiants et aux parents de régler facilement les frais de scolarité. Cette plateforme vise également à offrir une solution moderne, répondant aux besoins contemporains du secteur éducatif, tout en assurant la conformité aux normes en vigueur. Elle vise également à réduire voir à éliminer les pertes de temps liées à la lenteur des paiements bancaires.

Plan du document

Le mémoire sera structuré autour de trois chapitres afin d'explorer de manière approfondie les divers aspects de la conception et de la mise en œuvre de la plateforme de paiement de scolarité en ligne. Le premier chapitre abordera le déroulement du stage. Le deuxième chapitre, d'une part, présentera quelques plateformes de paiements en ligne de scolarité. Il détaillera d'autre part la méthodologie employée pour concevoir et développer la plateforme, mettant en

INTRODUCTION

lumière les choix techniques et les étapes clés du processus. Le troisième chapitre analysera les résultats obtenus suite à la réalisation de la plateforme et puis nous y relèverons les possibilités d'amélioration.

Chapitre 1 : Déroulement du stage

Chapitre 1 : Déroulement du Stage

Introduction

Dans ce chapitre, nous exposerons le déroulement de notre stage dans l'entreprise SKYRAN. Nous commencerons par décrire le fonctionnement de l'entreprise, en mettant en lumière ses principales activités et sa structure organisationnelle. Ensuite, nous présenterons les différents travaux et projets auxquels nous avons participé, en soulignant les tâches spécifiques que nous avons réalisées. Nous aborderons également les difficultés rencontrées au cours de notre stage, en expliquant les défis professionnels et personnels auxquels nous avons dû faire face. Enfin, nous analyserons l'apport de cette expérience sur notre développement professionnel. Cette présentation vise à offrir une vue complète de notre immersion en milieu professionnel.

1.1 Présentation de la structure

Fondée en 2019, SKYRAN- GROUP est une entreprise dynamique opérant dans le secteur de l'ingénierie informatique. Située dans la rue du lycée COULIBALY à Cotonou, elle se caractérise dans son fonctionnement par l'innovation, la qualité et la satisfaction client.



Figure 1: Logo de SKYRAN-GROUP

Dans le domaine de l'ingénierie informatique, **SKYRAN-GROUP** offre plusieurs services à savoir :

- ✓ La fourniture de matériel informatique.

- ✓ Les études statistiques et sociologiques,
- ✓ Le suivi et l'évaluation de projets,
- ✓ La gestion de projets.
- ✓ Le développement d'applications.

1.2 Fonctionnement de l'entreprise

SKYRAN GROUP dispose d'une structure organisationnelle mise en place pour assurer le bon fonctionnement et la coordination efficace des différentes fonctions de l'entreprise. À la tête de celle-ci se trouve la Directrice Générale (DG). Sous sa direction, le Directeur Technique (DT) et le département commercial. Cette structure organisationnelle simple et claire permet à l'entreprise de coordonner efficacement ses activités et de répondre aux besoins de ses clients avec professionnalisme et efficacité.

- ✓ La Directrice Générale (DG)

En tant que chef de l'entreprise, elle prend les décisions exécutives importantes concernant les opérations, les investissements et les partenariats. La Directrice générale supervise les dirigeants et les hauts responsables de l'entreprise, s'assurant qu'ils remplissent leurs fonctions et atteignent les objectifs fixés.

- ✓ Le Directeur Technique (DT)

Il supervise l'équipe technique chargée du développement et de la maintenance des produits ou services de l'entreprise.

- ✓ L'équipe technique

Il s'agit d'un ensemble de techniciens et d'ingénieurs spécialisés dans divers domaines de la technologie. Cette équipe est supervisée par le Directeur Technique

- ✓ Les commerciaux

Cette équipe compte les acteurs au niveau financier de l'entreprise, notamment l'expert-comptable et ses assistants.

La figure suivante présente l'organigramme de l'entreprise.

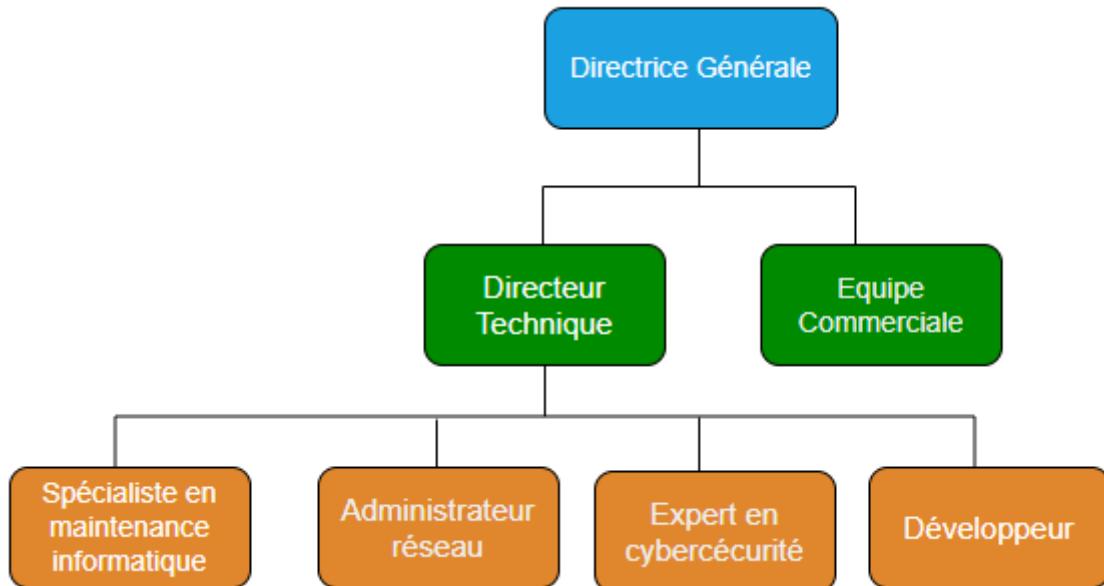


Figure 2:organigramme de SKYRAN GROUP

1.3 Travaux effectués

Au cours de notre stage à SKYRAN, nous avons effectué divers travaux parmi lesquels nous pouvons citer :

- ✓ La configuration des périphériques réseaux
- ✓ La Maintenance des équipements informatiques
- ✓ L'Installation et configuration d'un serveur HP ProLiant dl380 G10
- ✓ La mise en place d'une application de gestion des ressources humaines de l'entreprise

1.4 Apports du stage sur le plan professionnel

Ce stage nous a apporté des avantages à plus d'un titre. Il nous a permis de mettre en pratique les notions apprises au cours de notre parcours académique. De plus, nous avons pu développer des compétences telles que l'esprit d'équipe, la discipline en entreprise et l'organisation.

Enfin, il nous a permis d'acquérir une expérience pratique en ce qui concerne le milieu professionnel.

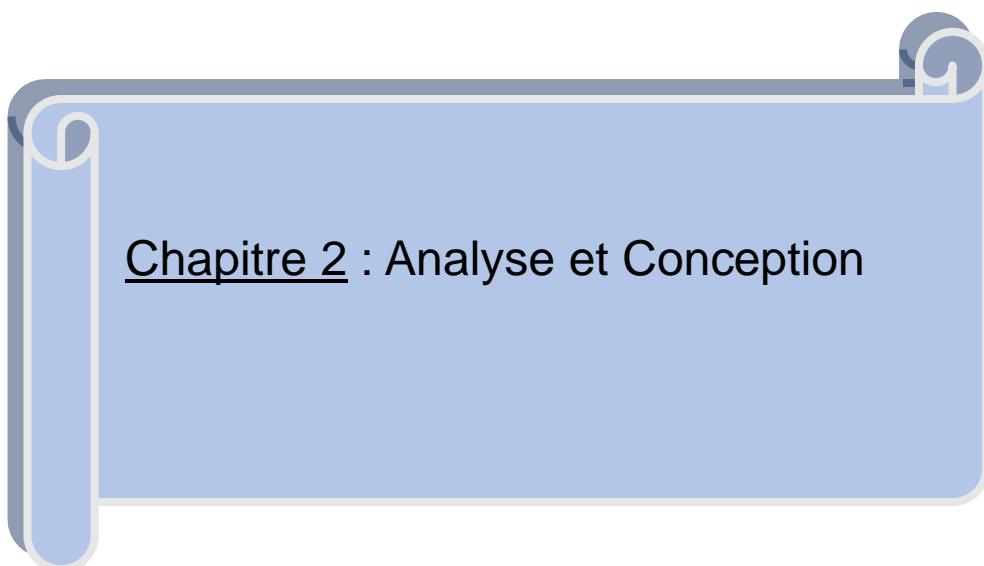
1.5 Difficultés rencontrées

Durant notre stage, nous avons rencontré certaines difficultés qui sont entre autres :

- La difficulté d'adaptation au fonctionnement de l'entreprise
- Difficulté d'adaptation aux interventions sur site au début du stage
- Difficulté à manipuler certains nouveaux équipements

Conclusion

Ce chapitre nous a permis de présenter SKYRAN GROUP, l'entreprise où s'est déroulé notre stage. Dans celui-ci, nous avons d'abord décrit l'entreprise et sa structure organisationnelle. Nous avons fait part des différents travaux effectués lors de ce stage, de l'apport sur le plan professionnel ainsi que les difficultés qui ont été rencontrées lors du stage. Dans le prochain chapitre, nous nous concentrerons sur le projet spécifique réalisé dans le cadre de ce mémoire. Nous y effectuerons une analyse approfondie, en détaillant les objectifs du projet, les méthodologies employées, les différentes étapes de sa mise en œuvre.



Chapitre 2 : Analyse et conception

Introduction

Dans ce chapitre, nous entreprendrons une analyse approfondie des solutions existantes dans le domaine des plateformes de paiement de scolarité en ligne, mettant en lumière leurs avantages et leurs limites. Nous présenterons la modélisation de notre système, détaillerons la méthodologie de travail adoptée, ainsi que les outils et les matériaux utilisés tout au long du processus de conception et de mise en œuvre de notre plateforme. Cette section fournira un aperçu complet des solutions déjà présentes sur le marché, éclairant ainsi sur les choix stratégiques et les innovations qui ont guidé notre approche dans le développement de notre solution.

2.1 État de l'art

2.1.1 Présentation du processus de paiement des frais de scolarité à Pigier

À Pigier-Bénin, le paiement des frais de scolarité est un processus nécessitant l'intervention de plusieurs acteurs, à savoir l'étudiant et le personnel de la comptabilité.

Pour tout étudiant inscrit, il se déroule en plusieurs étapes que voici :

- Retrait d'un bordereau de paiement bancaire.
 - Versement de la somme par l'étudiant à la banque dans le compte de l'école.
 - Présentation de la quittance de paiement bancaire à la comptabilité.
 - Établissement d'un reçu par le service comptable.
 - Retrait du reçu par l'étudiant
- Il faut également noter que :
- Pour connaître avec précision le solde restant par rapport à sa scolarité, l'étudiant ou le parent doit chaque fois se rendre au service comptable.

- Des étudiants peuvent parfois perdre les sommes en espèces du fait d'indélicatesse ou de manque d'attention de leur part.
- Le processus de versement bancaire peut prendre du temps.

L'analyse du processus de paiement de la scolarité (qui est souvent le même pour la majorité des universités privées) révèle, un manque d'automatisation. Cela engendre des pertes en temps, un manque de suivi et des erreurs. Conscient de ces difficultés qu'engendre le manque d'automatisation du processus de paiement des frais de scolarité, plusieurs sociétés ont décidé d'apporter des solutions afin d'y remédier. Le point suivant de ce chapitre présente quelques-unes de ces solutions.

2.1.2 Présentation des solutions existantes

Avec les avancées technologiques et l'essor des technologies numériques en Afrique, de nouvelles tendances ont émergé dans la gestion des processus éducatifs. C'est le cas des plateformes de paiement de scolarité en ligne, accessibles via des appareils mobiles et des ordinateurs. Dédiées à faciliter les transactions financières liées à la scolarité, ces plateformes permettent notamment le paiement des frais de scolarité, la consultation des relevés de paiement tout en offrant une expérience pratique aux étudiants et aux parents. Parmi ces plateformes, nous pouvons identifier quelques-unes au Bénin, et dans les pays de la sous-région :

- Paiement.tresorbenin.bj²



Figure 3: page d'accueil de la plateforme paiement.tresorbenin.bj

D'après l'article du site « Le Potentiel », publié le 19 Juin 2023, depuis le 23 Mai de la même année, toutes les entités de l'UAC ont été invitées à disposer de comptes ouverts au trésor public.³ Sur ces comptes seront collectés les différents frais académiques payés par les étudiants desdites entités. Ainsi, les étudiants des facultés comme l'EPAC, La FASEG, FASHS et FLLAC paient désormais leurs frais d'inscription semestriels sur leur compte respectif via la plateforme du trésor. Lors du paiement, les frais appliqués à la transaction représentent 1% du montant nominal du versement. La quittance de versement est automatiquement transmise à l'adresse électronique indiquée par la partie versante.

² Voir le numéro [2] dans la liste des références webographiques

³ Voir le numéro [3] dans la liste des références webographiques

- CAMPUSFASO au Burkina Faso



Figure 4:Logo CAMPUS FASO

CAMPUSFASO est une plateforme en ligne conçue spécifiquement pour les nouveaux bacheliers burkinabè et étrangers qui souhaitent s'inscrire dans les formations de l'enseignement supérieur au Burkina Faso.⁴ Grâce à CAMPUSFASO, les candidats peuvent suivre les campagnes d'orientation en ligne, gérer leurs inscriptions et effectuer des paiements sécurisés pour les frais d'inscription. Que ce soit pour les parents ou les étudiants, cette plateforme offre une solution pratique et efficace pour faciliter le processus d'accès à l'enseignement supérieur. Le mode de paiement accepté pour les frais d'inscription à CAMPUSFASO est : Orange Money

⁴ Voir le numéro [4] dans la liste des références webographiques

- **EDUS en côte d'Ivoire⁵**



Figure 5: Page d'accueil de la plateforme EDUS

EDUS est une plateforme de paiement en ligne des frais d'éducation et de formation qui est intégrée à plusieurs moyens de paiement, banques et portemonnaies électroniques. Les étudiants ou leurs parents peuvent effectuer des paiements en ligne de n'importe où et à n'importe quel moment.

⁵ Voir le numéro [5] dans la liste des références webographiques

- **SchoolMatik au Sénégal⁶**



Figure 6: page d'accueil de la plateforme SchoolMatik

SchoolMatik est une plateforme multiservices conçue depuis 2014 spécialement en vue de la digitalisation de la gestion scolaire. Plus Particulièrement, SchoolMatik est spécialisée dans la gestion monétique et digitale des paiements de frais scolaires. La plateforme SchoolMatik fait interagir les principaux acteurs de l'écosystème (Gestionnaires, comptables, Directeurs, parents ou payeurs) en temps réel 24h/24 et 7j/7.

Le tableau suivant présente un récapitulatif des forces et des faiblesses des plateformes existantes.

⁶ Voir le numéro [6] dans la liste des références webographiques

Tableau 1: force et faiblesse de quelques solutions existantes

Solutions existantes	Forces	Faiblesses
Trésor Bénin	<ul style="list-style-type: none"> - Permet aux étudiants de l'UAC de payer leurs frais académiques sur des comptes ouverts au trésor public - Favorise un meilleur suivi des paiements des frais académiques 	<ul style="list-style-type: none"> - N'est disponible que pour les universités publiques. - Ne prends en compte que les frais d'inscription par semestre.
CAMPUSFASO	<ul style="list-style-type: none"> - Facilité d'utilisation : Orange Money est largement utilisé au Burkina Faso, ce qui signifie que de nombreux étudiants sont déjà familiers avec ce service. - Disponibilité : Les points de vente Orange Money sont répartis dans tout le pays, ce qui facilite l'accès au service. - Rapidité : Les transactions via Orange Money sont généralement rapides et permettent aux étudiants de payer leurs frais d'inscription sans délai. 	<ul style="list-style-type: none"> - Interface complexe : Certains utilisateurs trouvent l'interface de CampusFaso complexe à naviguer, surtout s'ils ne sont pas familiers avec les procédures en ligne. - Modes de paiement limité : La plateforme n'accepte que les paiements via Orange Money
EDUS	<ul style="list-style-type: none"> - Intègre plusieurs moyens de paiement - Permet à différents établissements de 	<ul style="list-style-type: none"> - Frais de transaction élevés - Support client limité

	s'inscrire sur la plateforme - Permet aux étudiants et aux parents de payer les frais académiques	
SchoolMatik	- Prend en charge plusieurs établissements - Permet aux étudiants de payer leur scolarité	- Interface peu attrayante - Frais de transaction élevés

Les plateformes présentées ci-dessus, sont autant de solutions de paiement en ligne des frais de scolarité en milieu universitaire. Mais il faut noter qu'au Bénin, les universités privées ne disposent pas de solution leur permettant de faciliter le processus de paiement des frais de scolarité. D'où l'importance de réaliser une plateforme pour remédier à ce problème.

2.2 Présentation du projet :

Notre projet de fin de formation porte sur la « Conception et la réalisation d'une application Web de paiement de scolarité à Pigier-Bénin ». Ce projet vise à réaliser une plateforme permettant aux étudiants et à leurs parents de payer en ligne les frais de scolarité. Elle permettra également un meilleur suivi des paiements par le service comptable et les parents. La solution proposée dans ce projet sera destinée aux universités privées en général, et Pigier-Bénin a été pris comme exemple illustratif pour contextualiser notre travail.

2.2.1 Présentation des spécifications fonctionnelles et non fonctionnelles

❖ Spécifications fonctionnelles

1. Authentification et Autorisation

Les utilisateurs de la plateforme doivent pouvoir s'inscrire au préalable. De plus, ils doivent se connecter avant toute utilisation et peuvent gérer leur compte.

2. Gestion des Factures

Les étudiants doivent pouvoir visualiser les factures issues des différents paiements qu'ils ont effectués.

3. Paiement en Ligne

Les utilisateurs doivent pouvoir effectuer des paiements en ligne de manière sécurisée.

4. Historique des Transactions

Une historique détaillée des transactions doit être accessible pour les étudiants et le personnel de la comptabilité en la personne du chef comptable.

5. Intégration avec le Système Universitaire

L'application doit être intégrée au système universitaire pour récupérer automatiquement les informations sur les étudiants.

❖ Spécifications non fonctionnelles

1. Sécurité

Le système doit être conforme aux normes de sécurité pour protéger les informations sensibles.

2. Convivialité (Usabilité)

L'interface utilisateur doit être conviviale, avec des instructions claires pour faciliter l'utilisation par tous les utilisateurs.

3. Conformité aux normes de paiement

L'application doit respecter les normes de sécurité des paiements en ligne et être conforme aux réglementations locales sur les transactions financières.

4. Intégration avec les Services de Paiement

L'application doit être intégrée de manière transparente avec les services de paiement tiers.

2.3 Présentation des règles de gestion

Les règles de gestion sont un ensemble de directives ou d'ordres spécifiques qui dictent comment un logiciel doit fonctionner dans différentes situations. Ces règles peuvent concerner la manière dont les données sont gérées, les actions à prendre en cas d'événements spécifiques, ou encore les restrictions sur ce que les utilisateurs peuvent faire dans le système. En somme, elles aident à maintenir l'ordre et la cohérence dans le fonctionnement des logiciels.

Sur la base des différents besoins présentés ci-dessus, nous avons établi les règles de gestion qui régissent le système que nous comptons mettre en place. Afin de permettre une meilleure compréhension de ces règles de gestion, nous les avons décrites d'abord par rapport à tous les utilisateurs du système, puis par rapport à chaque utilisateur en particulier. Elles se présentent comme suit :

➤ Règles de gestion relatives à tout utilisateur du système

1. Un utilisateur peut être un étudiant, le chef comptable ou l'administrateur.
2. Tout utilisateur dispose d'un identifiant, d'un nom, d'un email et d'un mot de passe.
3. Pour effectuer une opération, l'utilisateur doit s'authentifier au préalable.
4. Les parents peuvent, à partir d'un profil d'étudiant, se connecter à la plateforme et consulter l'activité de leurs enfants.

➤ Règles de gestion relatives à l'étudiant/au parent

1. Tout étudiant de Pigier, lors d'une première utilisation de la plateforme, doit définir ses identifiants de connexion en renseignant son numéro matricule, son e-mail et son mot de passe.
2. Un étudiant s'inscrit pour le compte d'une année académique donnée, dans une certaine filière, et pour un niveau d'étude donné (Exemple de niveau d'étude : Licence 1, licence 2, Master 1...).
3. Durant son cursus, un étudiant est concerné par différents niveaux d'étude.
4. Chaque étudiant se connecte à la plateforme en renseignant son numéro matricule, et son mot de passe.
5. Les parents peuvent se connecter à la plateforme à partir d'un profil étudiant.
6. Tout étudiant (ou parent) peut payer en ligne la scolarité, consulter l'historique des paiements, ou voir le solde de sa scolarité.
7. Un étudiant peut effectuer plusieurs paiements.

8. Un paiement concerne un étudiant bien précis.
 9. Tout paiement d'un étudiant peut être lié à un niveau d'étude donné (exemple : Licence 1, Licence 2, etc.).
 10. Tout paiement s'effectue pour le compte d'une année académique donnée (exemple : 2023-2024).
 11. Une quittance est délivrée pour chaque paiement effectué.
 12. L'étudiant peut rechercher et télécharger sa quittance de paiement.
- Règles de gestion relatives au comptable et à l'administrateur
1. Le comptable crée un compte sur la plateforme en renseignant son nom d'utilisateur, son adresse mail et un mot de passe.
 2. Le comptable définit le montant total annuel à payer par filière et par année d'étude
 3. Le comptable peut voir la liste des étudiants, ajouter et mettre à jour les informations des étudiants.
 4. Le comptable peut voir la liste des étudiants inscrits sur la plateforme et le solde de leur scolarité. Il peut également consulter l'historique de paiement pour chaque étudiant.
 5. Le comptable peut ajouter ou mettre à jour une filière.
 6. L'administrateur gère les différents utilisateurs de la plateforme.

Les règles de gestions ainsi établies, serviront de base pour l'analyse UML

2.4 Analyse UML



Figure 7: logo-UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. Il se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

➤ **Pourquoi le langage UML ?**

1. Clarté et compréhension :

UML fournit des diagrammes visuels qui permettent une représentation claire et concise des différents aspects d'un système logiciel. Cela facilite la compréhension des concepts et des relations entre les différents éléments du système, ce qui est essentiel pour les développeurs, les architectes et les parties prenantes du projet.

2. Communication efficace :

Les diagrammes UML servent de langage commun entre les membres de l'équipe de développement, ce qui facilite la communication et la collaboration. Ils permettent aux développeurs de partager leurs idées, de discuter des concepts et de résoudre les problèmes plus efficacement.

3. Analyse des besoins :

UML offre des outils puissants pour modéliser les besoins des utilisateurs et les fonctionnalités attendues du système. Les diagrammes de cas d'utilisation permettent de définir les interactions entre les utilisateurs et le système, aidant ainsi à clarifier les exigences et à s'assurer que le système répond aux besoins des utilisateurs.

4. Conception structurée :

Les diagrammes de classes, de séquence et d'activité UML aident à concevoir la structure interne du système en identifiant les classes, les relations entre les

objets, les flux de contrôle et les interactions entre les composants. Cela permet aux développeurs de concevoir des solutions robustes et évolutives.

5. Gestion de la complexité :

UML permet de gérer la complexité des systèmes logiciels en les décomposant en composants plus simples et en représentant les interactions entre ces composants de manière intuitive. Cela facilite la gestion des projets de grande envergure et la répartition des tâches entre les membres de l'équipe de développement.

6. Détection précoce des problèmes :

En utilisant UML pour modéliser un système logiciel, il est possible de détecter les problèmes potentiels et les erreurs de conception dès les premières étapes du développement.

Les diagrammes UML permettent de visualiser les interactions entre les différents composants du système et d'identifier les points de friction ou les incohérences.

Cela permet de corriger les problèmes plus tôt dans le cycle de développement, ce qui peut réduire considérablement les coûts et les délais.

➤ Diagrammes UML utilisés

Dans ce projet, nous illustrerons trois de ces diagrammes, à savoir :

- Le diagramme de cas d'utilisation.
- Le diagramme de classe.
- Le diagramme de séquence.

2.4.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un type de diagramme UML comportemental et est fréquemment utilisé pour analyser divers systèmes. Ils permettent de visualiser

les différents types d'acteurs d'un système et la façon dont ces acteurs interagissent avec le système.

2.4.1.1 Identification des acteurs

Un acteur est une entité externe qui intervient dans le fonctionnement du système. Il peut être un utilisateur humain, une organisation, une machine ou un autre système externe. Parmi les catégories d'acteurs, nous pouvons citer les acteurs principaux et les acteurs secondaires. Les différents acteurs intervenant dans notre système sont :

- ✓ L'étudiant : qui peut effectuer le paiement de sa scolarité et mettre à jour son compte.
- ✓ Le parent : qui peut effectuer les mêmes actions que l'étudiant.
- ✓ L'administrateur : qui peut gérer les utilisateurs, administrer la plateforme.
- ✓ Chef comptable : Il peut ajouter, voir et mettre à jour des étudiants. Il peut en outre consulter le solde de leur scolarité.

2.4.1.2 Identification des cas d'utilisations

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

1- Cas d'utilisation relatifs à l'étudiant / parent :

- ✓ Payer la scolarité
- ✓ Consulter l'historique des paiements
- ✓ Rechercher une quittance
- ✓ Télécharger une quittance de paiement
- ✓ Consulter le solde de la scolarité
- ✓ Mettre à jour son compte

2- Cas d'utilisation relatifs à l'administrateur

- ✓ Administrer la plateforme
- ✓ Gérer les utilisateurs

3- Cas d'utilisation relatifs au Chef comptable

- ✓ Voir la liste des étudiants
- ✓ Mettre à jour les informations d'un étudiant
- ✓ Paramétriser les scolarités à payer
- ✓ Faire le suivi des paiements
- ✓ Ajouter une filière
- ✓ Mettre à jour une filière

Après identification des différents acteurs du système, ainsi que les cas d'utilisation liés à chacun d'eux, nous avons pu établir le diagramme de cas d'utilisation de notre système qui se présente comme suit :

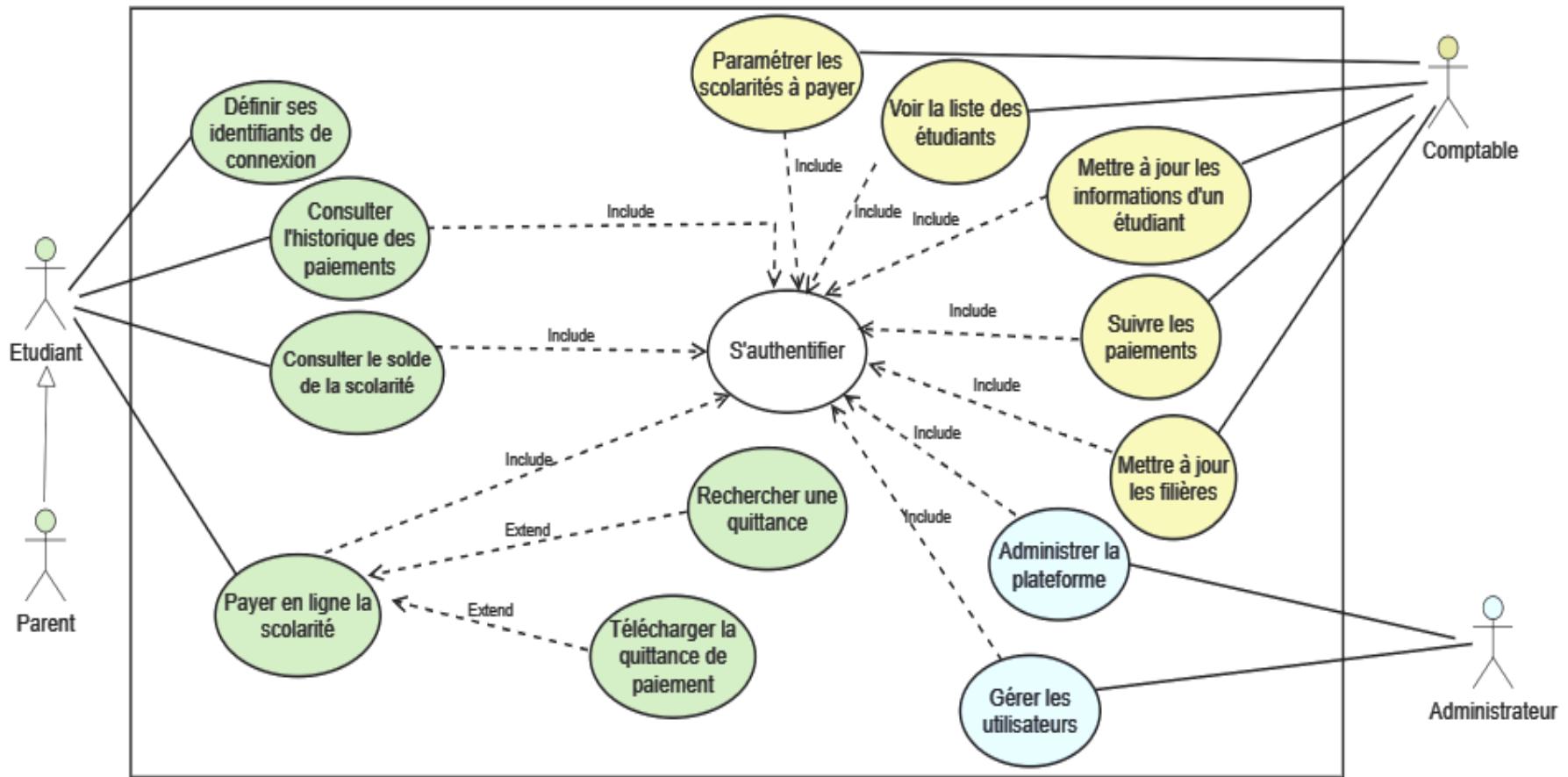


Figure 8: Diagramme de cas d'utilisation Gestion des étudiants

2.4.1.3 Description des cas d'utilisations

Cas d'utilisation 1 : « Définir ses identifiants de connexion »

Tableau 2: Description du cas d'utilisation « Définir ses identifiants de connexion »

Résumé :	Ce cas d'utilisation permet à un étudiant de s'inscrire sur la plateforme.
Acteur :	Étudiant
Précondition :	<ul style="list-style-type: none"> - L'étudiant doit être inscrit (ou réinscrit) à l'université. - L'étudiant doit posséder un numéro matricule. - L'étudiant possède un mot de passe par défaut
Postcondition :	L'étudiant est redirigé vers la page de connexion.
Scénario nominal :	<ol style="list-style-type: none"> 1- L'étudiant clique sur le bouton « Configurer mes identifiants » 2- Le système affiche un formulaire au niveau duquel l'étudiant renseigne son numéro matricule et son mot de passe actuel (ou par défaut). 3- L'étudiant soumet le formulaire. 4- Le système vérifie les informations. 5- Lorsque les informations sont vérifiées, le système affiche le formulaire de configuration des nouveaux identifiants. 6- L'étudiant renseigne le formulaire avec son adresse mail et le nouveau mot de passe puis le soumet. 7- Le système enregistre les nouvelles informations. 8- L'utilisateur est redirigé vers la page de connexion.
Scénario alternatif :	<p>SA1 : Le numéro matricule entré par l'étudiant est erroné. SA1 commence au point 3 du scenario nominal. Un message s'affiche indiquant que le numéro matricule n'existe pas. Le scénario reprend au point 1 du scénario nominal.</p>

Cas d'utilisation 2 : « S'authentifier »

Tableau 3 : Description du cas d'utilisation "s'authentifier"

Résumé :	Ce cas d'utilisation permet à un étudiant de s'authentifier.
Acteur :	Étudiant
Précondition :	L'étudiant s'est déjà inscrit sur la plateforme
Postconditions :	L'étudiant est redirigé vers la page d'accueil.
Scénario nominal :	<ol style="list-style-type: none"> 1. Le système affiche la page d'authentification. 2. L'étudiant entre les informations demandées dans le formulaire (son numéro matricule et son mot de passe). 3. Le système compare les informations entrées par l'étudiant à celles se trouvant dans la base de données. 4. Lorsque les informations sont vérifiées, l'étudiant est redirigé vers la page d'accueil.
Scénario alternatif :	<p>SA1 : Les informations entrées par l'étudiant sont erronées.</p> <p>SA1 commence au point 3 du scenario nominal.</p> <p>Un message s'affiche indiquant que les informations renseignées sont erronées</p> <p>Le scénario reprend au point 1 du scénario nominal.</p>

Cas d'utilisation 3 : « Payer la scolarité »

Tableau 4: Description du cas d'utilisation "payer la scolarité"

Résumé :	Ce cas d'utilisation permet à un étudiant de payer en ligne sa scolarité
Acteurs :	Étudiant, Système api de paiement
Précondition :	L'étudiant s'est authentifié.
Postconditions :	Un message de succès est envoyé.
Scénario nominal :	<ol style="list-style-type: none"> 1- L'étudiant clique sur le bouton “payer ma scolarité”. 2- L'étudiant est redirigé vers un formulaire où il renseigne les informations de paiement telles que le numéro de téléphone ainsi que la somme à payer. 3- L'étudiant soumet le formulaire 4- Le système d'API de paiement vérifie les informations. 5- Une fiche de détails du paiement lui est présentée par le système d'API 6- L'étudiant renseigne et soumet la fiche. 7- Le système d'API de paiement initie le paiement avec l'opérateur de paiement. 8- Un message de confirmation lui est envoyé par l'opérateur de paiement. 9- L'étudiant entre son code mobile money pour confirmer le paiement. 10- Le paiement est exécuté et un message de succès est envoyé à l'étudiant. 11- L'étudiant est ensuite invité à télécharger sa quittance.

Scénarios alternatifs :	<p>SA1 : L'étudiant a déjà payé la totalité de sa scolarité. Un message s'affiche, indiquant que l'étudiant a déjà soldé.</p> <p>SA2 : Les informations entrées par l'étudiant sont erronées Un message d'erreur s'affiche, puis le scénario reprend au point 2 du scenario nominal.</p> <p>SA3 : L'étudiant annule la transaction ou ne la confirme pas avec son code mobile money La transaction est abandonnée, puis le scénario reprend au point 2 du scenario nominal.</p> <p>SA4 : Les fonds de l'étudiant sont insuffisants pour effectuer le paiement. La transaction est abandonnée, puis le scénario reprend au point 2 du scenario nominal.</p>
--------------------------------	--

Cas d'utilisation 4 : « Rechercher une quittance »

Tableau 5: Description du cas d'utilisation « rechercher une quittance »

Résumé :	Ce cas d'utilisation permet à un étudiant de rechercher la quittance associée à un paiement.
Acteurs :	Étudiant, parents
Précondition :	- L'étudiant s'est authentifié et a effectué un paiement.
Postconditions :	La quittance est retrouvée et l'étudiant peut la télécharger.
Scénario nominal :	<ol style="list-style-type: none"> 1- L'étudiant clique sur le bouton "retrouver ma quittance". 2- L'étudiant est redirigé vers un formulaire où il est invité à renseigner les informations de paiement telles que le numéro de référence du paiement et la somme payée. Il soumet le formulaire. 3- Le système compare les informations entrées par l'étudiant avec les informations de sa base de données. 4- Une fiche de détails du paiement lui est présentée, puis il est invité à télécharger sa quittance.
Scénario alternatif :	<p>SA1 : Les informations entrées par l'étudiant sont erronées. Un message s'affiche, indiquant que les informations ne sont pas correctes.</p> <p>Le scénario reprend au point 2 du scenario nominal</p>

2.4.2 Diagramme de classe

Un diagramme de classe⁷ est un type de diagramme UML qui décrit un système en visualisant les différents types d'objets au sein d'un système et les types de relations statiques qui existent entre eux. Il illustre également les opérations et les attributs des classes. Ils sont généralement utilisés pour explorer les concepts de domaine, comprendre les exigences logicielles et décrire les conceptions détaillées.

⁷ Voir le numéro [7] dans la liste des références webographiques

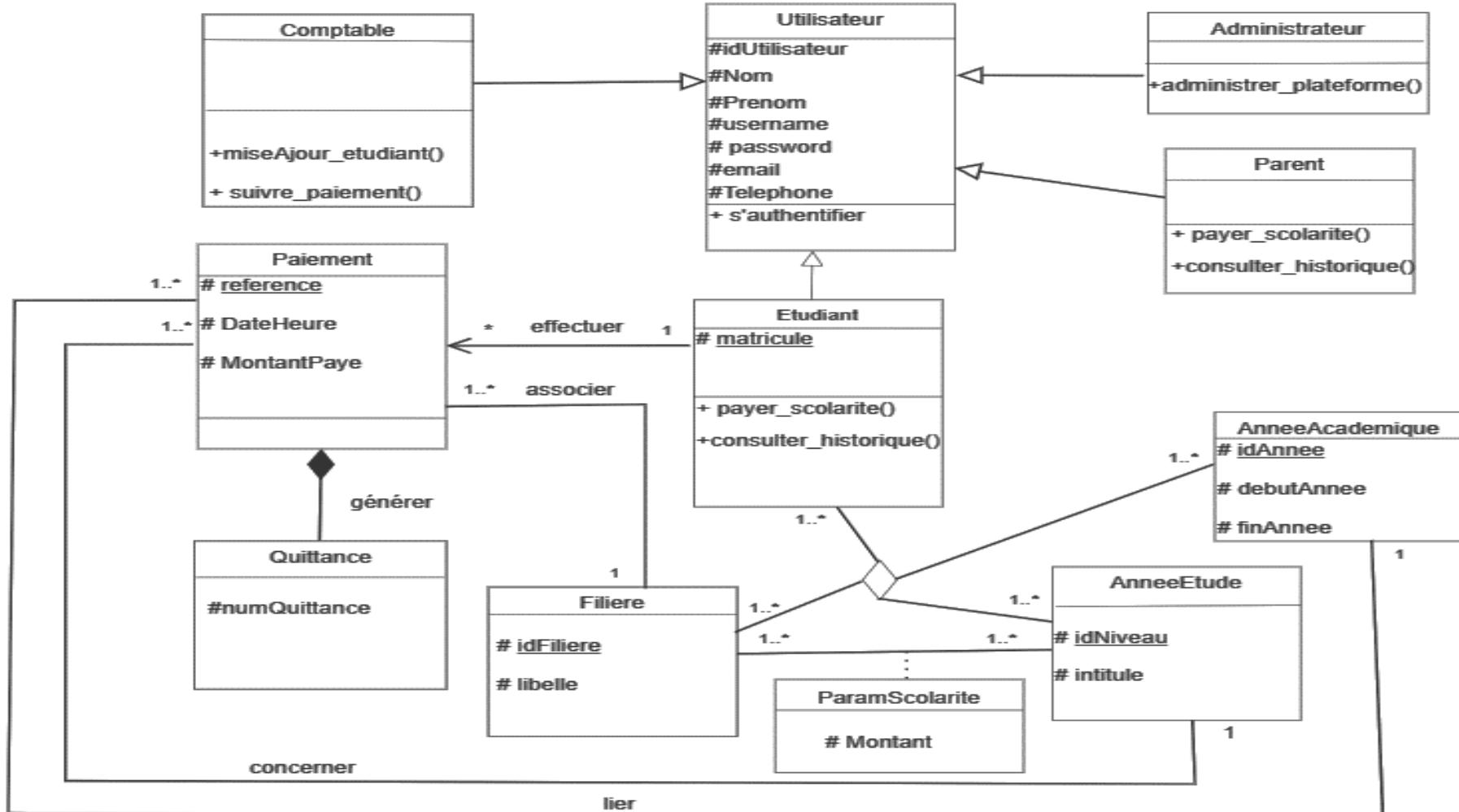


Figure 9: Diagramme de classe

2.4.2.1 Modèle Logique des données

Le MLD ou Modèle Logique des Données est simplement la représentation textuelle du schéma représentant la structure de la base de données. Pour réaliser un MLD, il faut respecter les règles suivantes :

- Chaque ligne représente une table ;
- C'est toujours le nom de la table qui est écrit en premier ;
- Les champs sont listés entre parenthèses et séparés par des virgules ;
- Les clés primaires sont soulignées et placées au début de la liste des champs ;
- Les clés étrangères sont préfixées par un dièse

Le Modèle Logique des Données associé au diagramme de classe se présente comme suit :

- T_Administrateur (idUtilisateur, Nom, Prenom, username, password, email, Telephone)
- T_Comptable (idUtilisateur, Nom, Prenom, username, password, email, Telephone)
- T_Etudiant (matricule, idUtilisateur, Nom, Prenom, username, password, email, Telephone)
- T_Paiement (reference, DateHeure, MontantPaye, #matricule, #idAnnee, #idNiveau, #idFiliere)
- T_Quittance (numQuittance, #reference)
- T_ParamScolaire (#idNiveau, #idFiliere, Montant)
- T_AnneeAcademique (idAnnee, debutAnnee, finAnnee)
- T_AnneeEtude (idNiveau, intitule)
- T_Filiere (idFiliere, libelle)

2.4.3 Diagramme de séquence

Les diagrammes de séquence, couramment utilisés par les développeurs, modélisent les interactions entre les objets dans un cas d'utilisation unique. Ils illustrent la manière dont les différentes parties d'un système interagissent entre elles pour

exécuter une fonction, et l'ordre dans lequel les interactions se produisent lorsqu'un cas d'utilisation particulier est exécuté.

En termes plus simples, un diagramme de séquence montre les différentes parties d'un système qui travaillent dans une "séquence" pour obtenir quelque chose. Un diagramme de séquence est structuré de telle sorte qu'il représente une ligne de temps qui commence en haut et descend progressivement pour marquer la séquence des interactions. Chaque objet possède une colonne et les messages échangés entre eux sont représentés par des flèches.

Nous avons identifié quatre cas d'utilisations clés du système dont les diagrammes de séquences sont présentés comme suit :

- ❖ Diagramme de séquence relatif au cas d'utilisation « Définir ses identifiants de connexion » (interface étudiant)



Figure 10: Diagramme de séquence du Cas d'utilisation « Définir ses identifiants de connexion »

- ❖ Diagramme de séquence relatif au cas d'utilisation « s'authentifier » (interface étudiant)

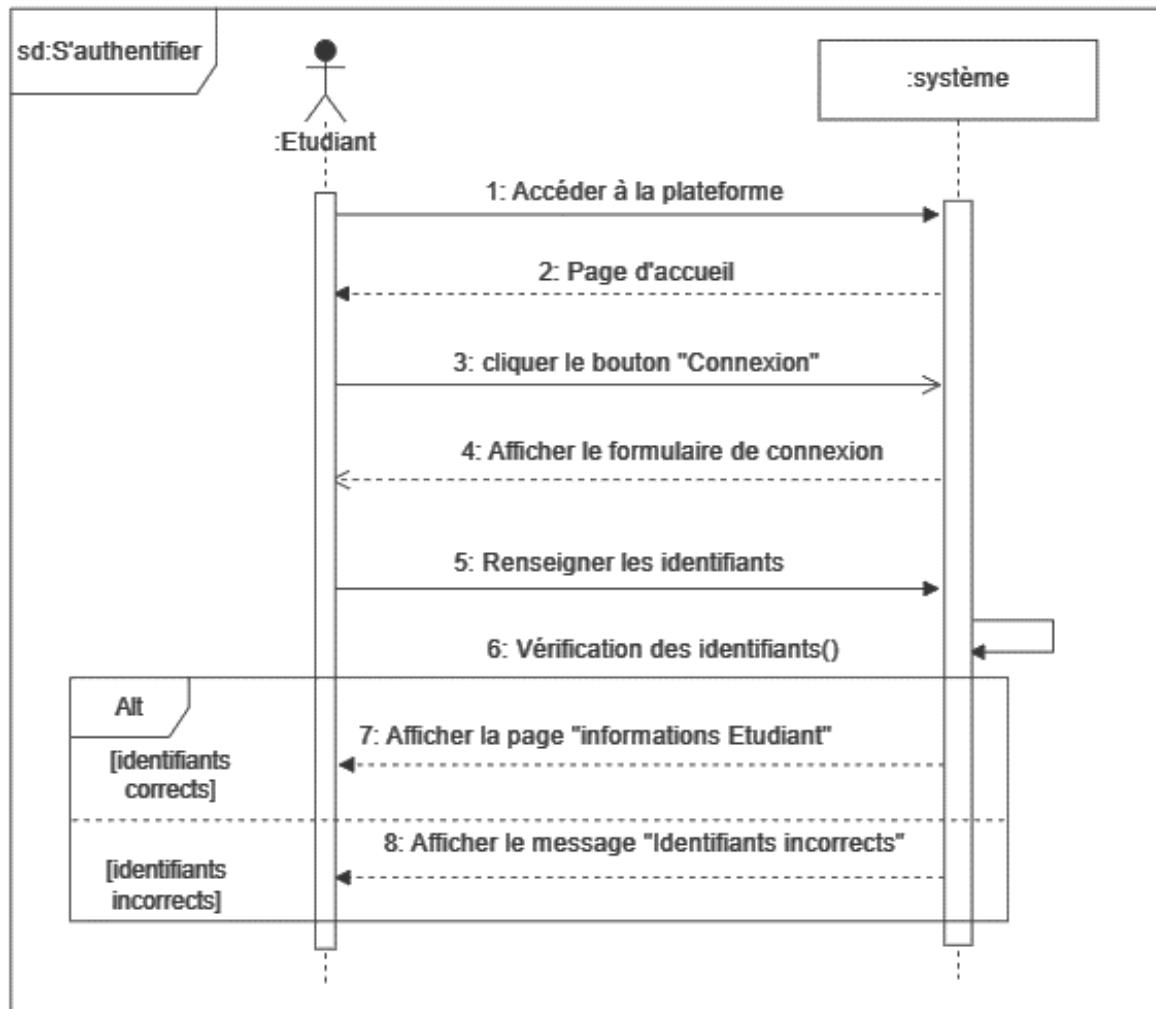


Figure 11: Diagramme de séquence du Cas d'utilisation « S'authentifier »

- ❖ Diagramme de séquence relatif au cas d'utilisation « payer en ligne la scolarité »

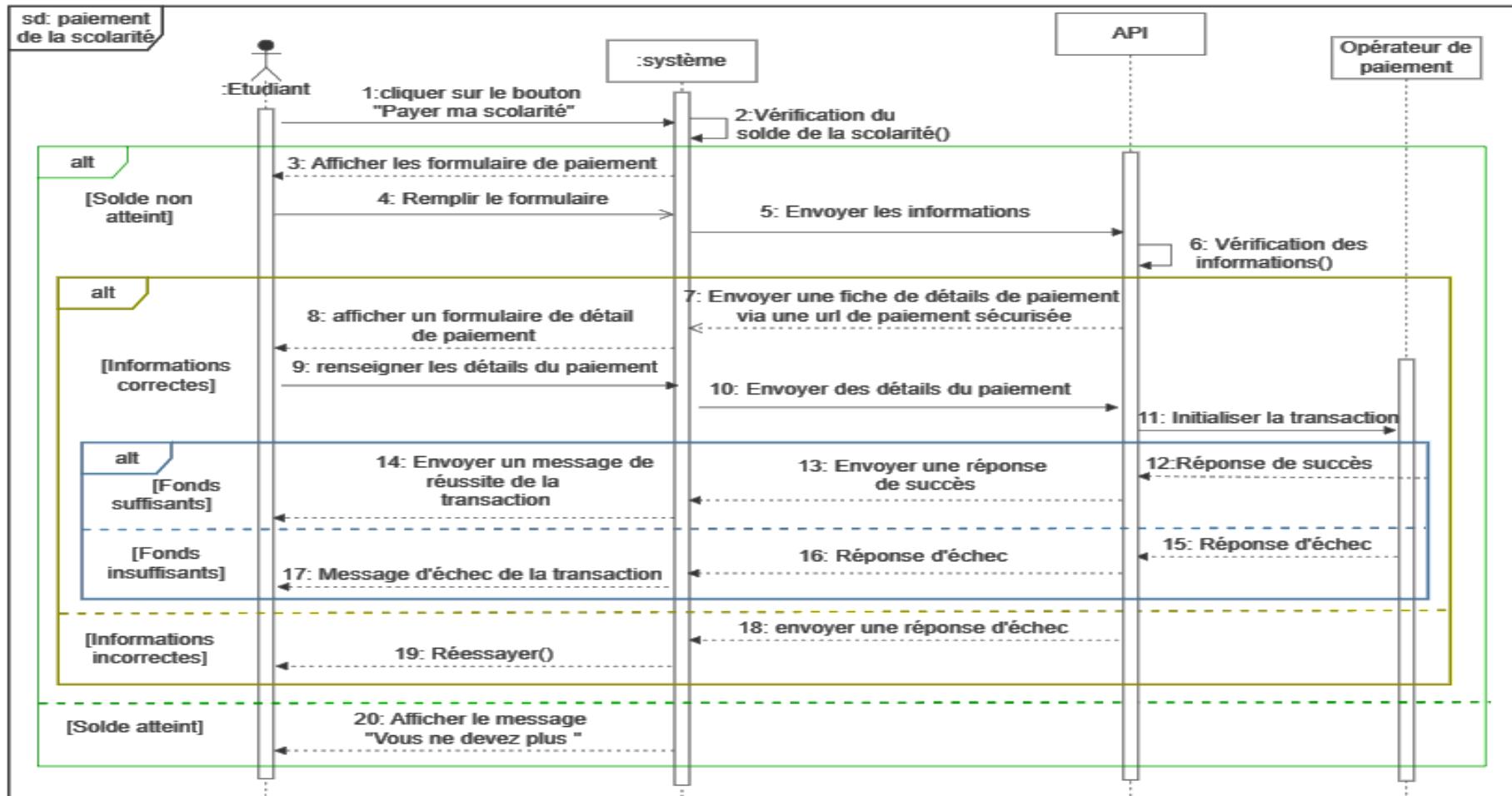


Figure 12: Diagramme de séquence du cas d'utilisation « payer la scolarité »

❖ Diagramme de séquence relatif au cas d'utilisation « Rechercher une quittance »

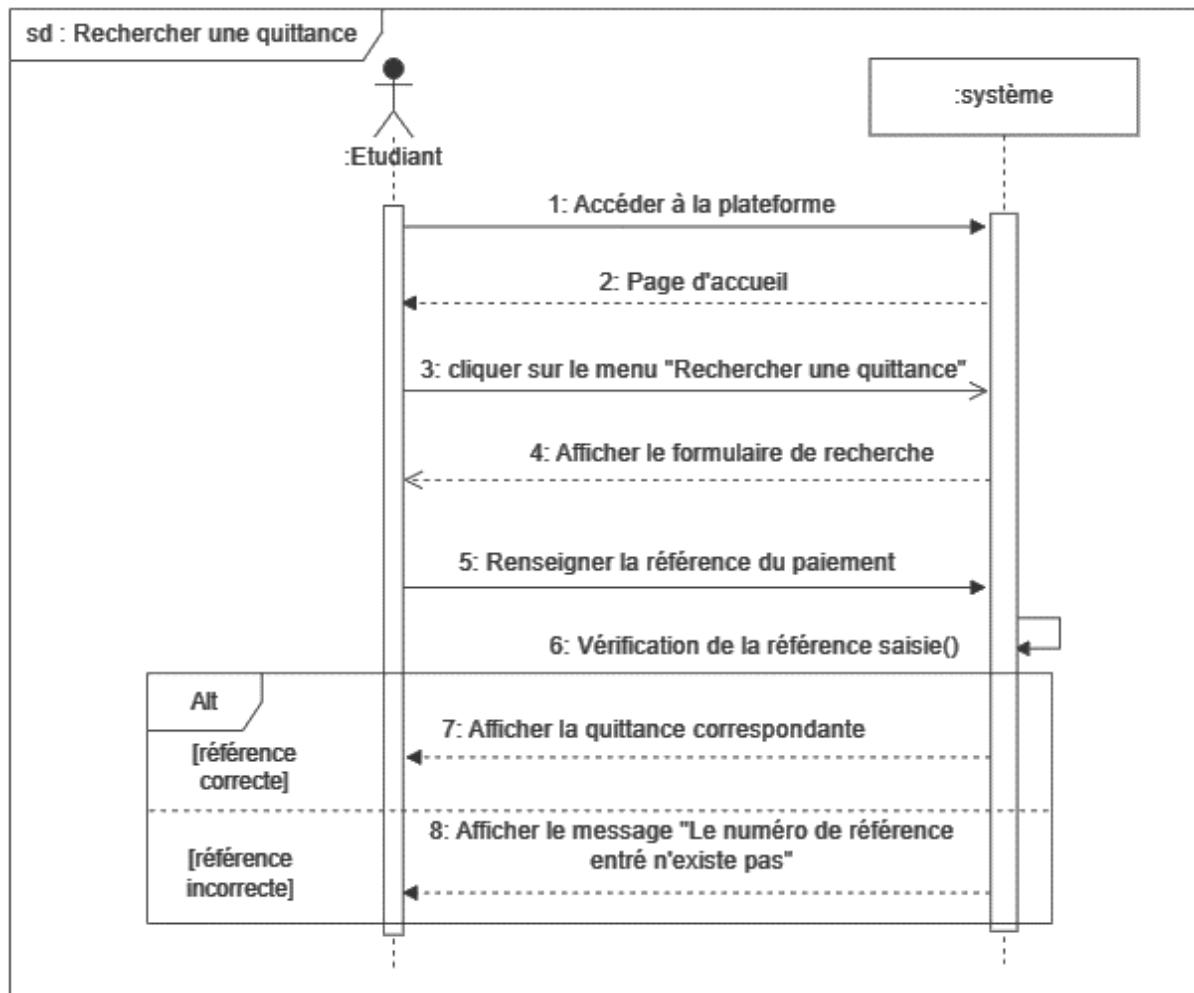


Figure 13: Diagramme de séquence du cas d'utilisation « Rechercher une quittance »

- ✓ Diagramme de séquence relatif au cas d'utilisation « Consulter l'historique de paiement »

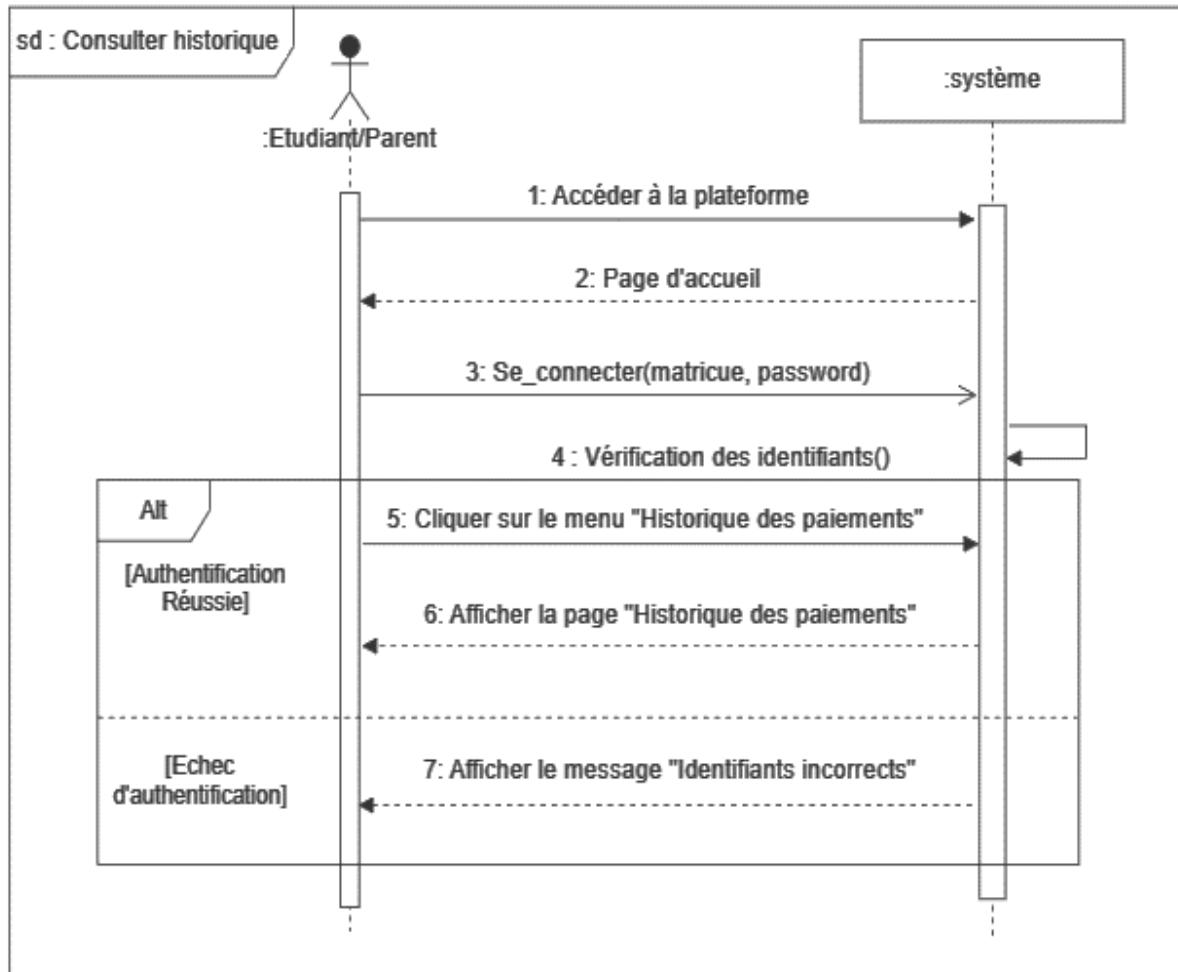


Figure 14: Diagramme de séquence du cas d'utilisation « consulter l'historique de paiement »

2.5 Logiciels et langages de programmation utilisés :

2.5.1 Logiciels utilisés

Durant la réalisation de notre projet, nous avons utilisé différents langages de programmation, Framework et outils collaboratifs.

- Pour la base de données

Pour réaliser et gérer notre base de données, nous avons opté pour **MySQL** qui est un système de gestion de base de données relationnelle (SGBDR) open source basé sur le langage de requête structuré (SQL). SQL est le langage le plus populaire pour ajouter, accéder et gérer du contenu dans une base de données. Il est connu pour sa vitesse de traitement rapide, sa fiabilité, sa facilité et sa flexibilité d'utilisation.



Figure 15: logo de MySQL

Pour interagir avec celle-ci, nous avons utilisé **phpMyAdmin**, qui est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL.



Figure 16:Logo phpMyAdmin

➤ Pour le développement :

Les logiciels utilisés pour le développement de la solution sont les suivants :

1. GitHub :



Figure 17: Logo de GitHub

GitHub est un service basé sur le cloud qui héberge un système de contrôle de version (VCS) appelé Git. Il permet aux développeurs de collaborer et d'apporter des modifications aux projets partagés tout en gardant un suivi détaillé de leur progression.



Figure 18 : logo-Visual studio code

2. Visual Studio Code :

Visual Studio Code est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, Go, Node.js et C++. Il est basé sur le cadre Electron, qui est utilisé pour développer des applications Web Node.js qui s'exécutent sur le moteur de présentation Blink. Visual Studio Code utilise le même composant d'éditeur (nom de code Monaco) utilisé dans Azure DevOps (anciennement appelé Visual Studio Online et Visual Studio Team Services). Le logiciel prend en charge le Windows Subsystem for Linux et,

permet ainsi par exemple, de programmer facilement en C/C++ depuis un ordinateur Windows 10.

- Pour la réalisation des diagrammes



Draw.io

Figure 19: Logo-Draw.io

Draw.iO est un logiciel gratuit pour la schématisation et d'organisation d'idées. Il s'agit d'un outil en ligne plutôt performant et à l'esthétique soignée. Il se distingue ainsi des autres programmes de création de diagrammes.

2.5.2 Langages utilisés :

- Pour le frontend :

En développement web, la notion de « frontend » fait référence à l'ensemble des éléments visibles et accessibles directement sur un site web (voire sur une application web ou une application web mobile). Dans le cadre de notre projet, les différents langages utilisés au niveau du frontend sont les suivants :

1- HTML



Figure 20: logo-HTML

HTML (HyperText Markup Language,) est le langage utilisé pour structurer une page web et son contenu. Il s'agit en outre d'un langage de balisage langage, utilisé pour la création de sites Internet.

2- CSS



Figure 21: logo-css

CSS (Cascading Style Sheets) est un langage informatique de description utilisé pour compléter le **HTML** (HyperText Markup Language) et lui conférer un style personnalisé. En d'autres termes, **CSS** permet de définir la mise en forme des

éléments d'un document web, tels que la couleur, la police, l'espacement, la disposition, etc.

3- JavaScript



Figure 22: logo-JavaScript

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. Lorsqu'une page web va au-delà de l'affichage de contenu statique et inclut des éléments tels que des mises à jour de contenu à intervalles réguliers, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, **JavaScript** intervient. C'est la troisième couche des technologies standards du web, les deux premières étant **HTML** (pour structurer et donner du sens au contenu web) et **CSS** (pour mettre en forme le contenu HTML).

En d'autres termes, **JavaScript** est un langage de programmation dynamique qui, appliqué à un document **HTML**, peut fournir une interactivité dynamique sur les sites Web. Il permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images et bien plus encore.

➤ **Pour le Backend :**

Le backend, c'est toute la partie que l'utilisateur ne voit pas, mais qui lui permet de réaliser des actions sur un site ou une application. Pour la réalisation du backend, nous avons utilisé le langage **PHP** :



Figure 23: logo-PHP

Le PHP⁸, ou « Hypertext Preprocessor », est un langage de programmation open-source, principalement utilisé pour créer des sites web dynamiques et des applications web. Il fonctionne côté serveur, ce qui signifie que le code PHP est exécuté sur le serveur web avant d'être envoyé au navigateur du client sous la forme de HTML, CSS ou JavaScript. Le PHP a été créé en 1994 par Rasmus Lerdorf, un programmeur dano-canadien. À l'origine, il s'agissait d'un ensemble de scripts en Perl pour suivre les visites de son curriculum vitae en ligne. Lerdorf a ensuite développé ces scripts et les a publiés sous le nom de « Personal Home Page Tools » (PHP Tools). Le langage a rapidement évolué et a été réécrit par d'autres programmeurs, donnant naissance au PHP que nous connaissons aujourd'hui. **Le PHP est un langage interprété**, ce qui signifie qu'il n'a pas besoin d'être compilé avant d'être exécuté. Il est également faiblement typé, ce qui permet une certaine souplesse dans la manipulation des variables. De plus, le PHP prend en charge la programmation orientée objet, ce qui facilite la réutilisation du code et la maintenance des applications. (Conallen, 2000)

3.1 2.5.3 Les frameworks utilisés

⁸ Définition du PHP : voir numéro [8] dans la liste des références webographiques

Un **Framework** (ou **infrastructure logicielle** en français) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. L'objectif du Framework est de simplifier et d'uniformiser le travail des développeurs. Il fonctionne comme un cadre ou un patron, mais son maniement suppose d'avoir déjà un profil expérimenté. En général, une infrastructure logicielle est associée spécifiquement à un langage de script ou de programmation. Par exemple, Hibernate est un Framework pour JavaScript et Django pour Python.

Pour la réalisation de notre projet, nous avons utilisé les frameworks suivants :

1- Laravel

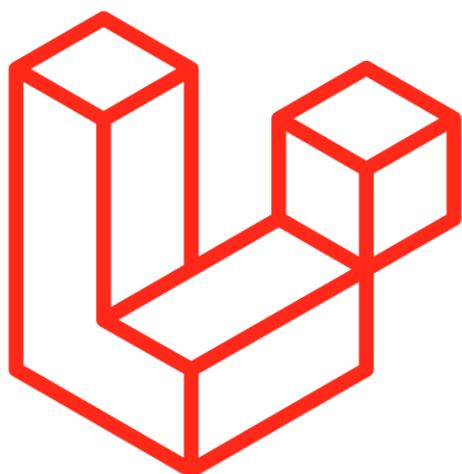


Figure 24: Logo de Laravel

Laravel est un Framework PHP open source conçu pour faciliter le développement des applications web. Créé par Taylor Otwell en 2011, il s'est démarqué avec prestance parmi les plus grands Framework du langage de programmation PHP au monde. Nous avons utilisé la version 10 du Framework Laravel.

Parmi les raisons qui justifient le choix de ce Framework, nous pouvons citer :

- ✓ **Architecture élégante et expressive** : Laravel offre une syntaxe élégante et expressive qui simplifie le processus de développement. Son architecture bien conçue facilite la création de code propre et maintenable.

- ✓ **ORM puissant** : Eloquent, l'ORM (Object-Relational Mapping) intégré à Laravel, simplifie la manipulation des bases de données en utilisant des modèles orientés objet. Cela rend les opérations de base de données plus simples et plus intuitives.
- ✓ **Migration de base de données** : Laravel propose un système de migration de base de données qui permet aux développeurs de versionner et de partager facilement la structure de la base de données. Cela facilite la gestion des schémas de base de données tout au long du cycle de développement.
- ✓ **Système de routage** : Laravel propose un système de routage simple et puissant qui permet de définir les points d'entrée de manière claire et organisée. Cela facilite la gestion des URL et des contrôleurs.
- ✓ **Système de templating Blade** : Blade, le moteur de templating de Laravel, offre une syntaxe simple et expressive pour créer des vues. Il facilite la création de vues élégantes et maintenables.
- ✓ **Sécurité** : Laravel intègre des fonctionnalités de sécurité telles que la protection CSRF (Cross-Site Request Forgery), la validation des données, la protection contre les injections SQL, etc.
- ✓ **Écosystème robuste** : Laravel dispose d'un écosystème étendu avec de nombreuses bibliothèques tierces, des packages Composer et des outils qui facilitent le développement rapide et la gestion des dépendances.

2- Bootstrap



Figure 25: logo- Bootstrap

Bootstrap est un **Framework frontend** développé par l'équipe du réseau social Twitter. Proposé en open source, ce Framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce Framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de Framework un "Frontend Framework".

➤ API utilisée

Une interface de programmation d'application (API pour application programmation interface) correspond à un ensemble de règles grâce auxquelles un logiciel transmet des données à un autre logiciel.⁹

Les API évitent aux développeurs des tâches redondantes ; au lieu de créer sans cesse des fonctions d'application qui existent déjà, les développeurs peuvent intégrer les fonctions qui existent déjà dans leurs nouvelles applications en formatant les requêtes en fonction des exigences de l'API.

⁹ Voir numéro [9] dans la liste des références bibliographiques

Dans le cadre de notre mémoire, nous avons utilisé l'API fourni par l'agrégeateur de paiement FedaPay

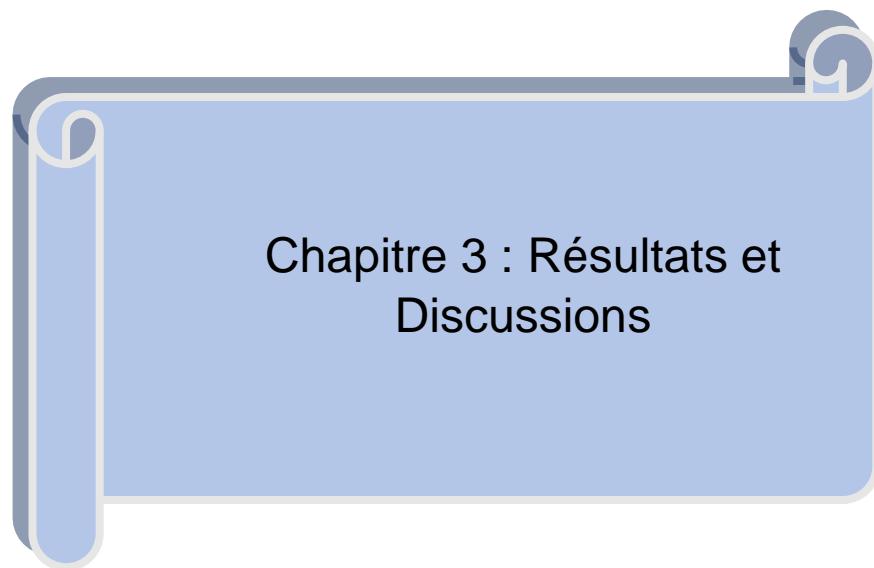


Figure 26: FedaPay

FedaPay une solution de Nautilus Technology est un système facile et sécurisé de paiement par cartes de crédit et par Mobile Money. Elle élimine pour les marchands les tracasseries de gestion de plusieurs systèmes de paiements en leur permettant de se concentrer sur l'essentiel et réduit pour les développeurs le temps d'intégration.

Conclusion

Dans ce chapitre, nous avons présenté quelques solutions existantes qui portent sur le domaine des paiements en ligne en milieu universitaire. Nous y avons également exposé la méthode utilisée, les différentes technologies utilisées dans la conception et la réalisation de notre projet. Dans le prochain chapitre, nous présenterons et les résultats obtenus et discuterons des implications pratiques de notre projet, puis évaluerons les perspectives futures et les recommandations pour l'amélioration continue de notre solution.



Chapitre 3 : Résultats et Discussions

Introduction

Ce chapitre présente en détail les résultats obtenus suite aux différentes analyses et travaux menés tout au long de notre projet. Ces résultats feront l'objet d'une analyse, afin de mettre en évidence les points forts et les axes d'amélioration. Cette évaluation nous permettra d'évaluer l'efficacité de la solution proposée et de suggérer des ajustements pour optimiser les performances futures.

3.1 Résultats

✓ Page d'accueil de la plateforme

La figure suivante présente la page d'accueil. Il s'agit de la première interface à laquelle aura accès l'étudiant lorsqu'il accède à la plateforme.

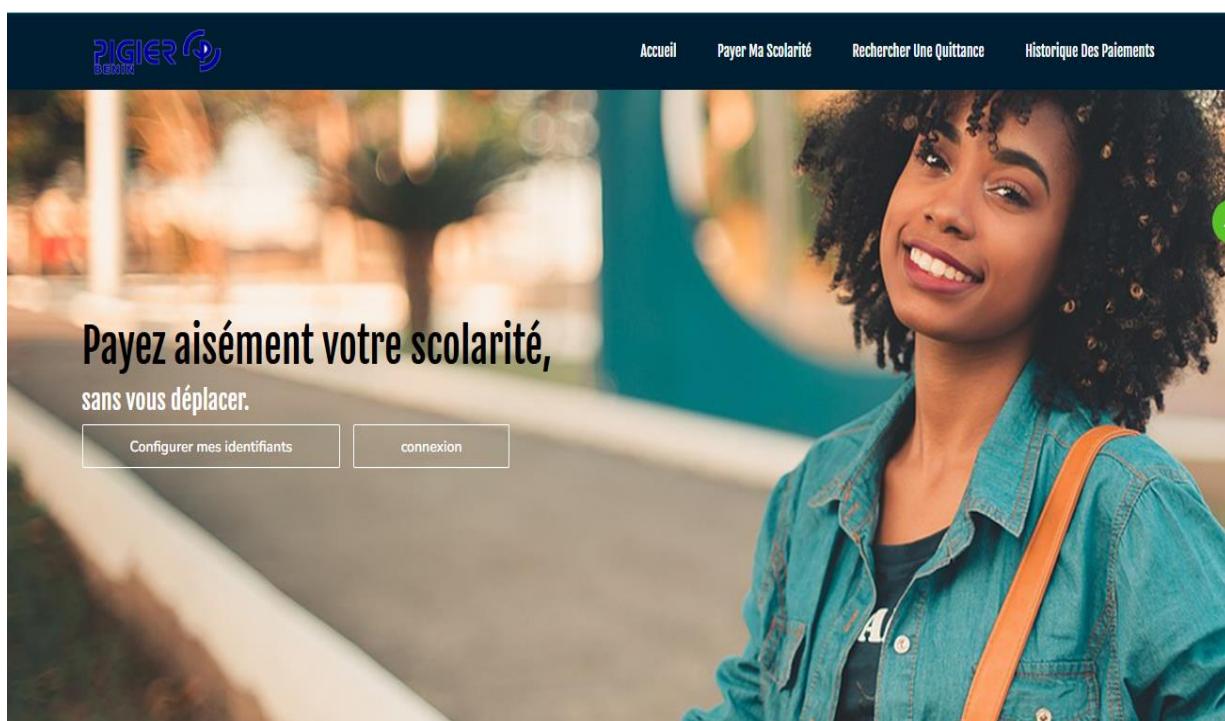


Figure 27: Page d'accueil

✓ **Formulaire de configuration des identifiants de connexion (interface étudiant)**

Lorsque l'étudiant clique sur le bouton « configurer mes identifiants », il est redirigé vers un formulaire où il renseigne son numéro matricule et le mot de passe qui lui a été attribué par défaut puis valide les informations. Il est ensuite redirigé vers un autre formulaire où il peut ajouter une adresse mail et créer un nouveau mot de passe.

The screenshot shows a user interface for configuring student identification. It consists of several input fields and labels:

- Matricule:** A field containing the value "123456789".
- Ajoutez un email:** A field labeled "Ajoutez une adresse email".
- Nouveau mot de passe:** A field labeled "Créez un mot de Passe".
- Confirmer le mot de passe:** A field labeled "Confirmez le mot de passe".
- Soumettre:** An orange button labeled "Soumettre".
- Link:** A link at the bottom left of the form area that says "Identifiants déjà configurés ? Connectez-vous".

Figure 28: formulaire de configuration des identifiants de connexion

La figure suivante présente la fonction permettant à un étudiant de configurer ses identifiants.

```

public function EtudiantRegisterCreate(Request $request){
    $etudiant=Etudiant::where('matricule' , $request->input('matricule'))->first();
    $request->validate([
        'email'=>'required|email',
        'matricule' => 'required|unique:etudiants',
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
    $etudiant->update([
        'email' => $request->email,
        'password' => Hash::make($request->password),
        'created_at' => Carbon::now(),
    ]);
    return redirect()->route(['etudiant_login_from']);
}

```

You, 1 second ago • Uncommitted

Figure 29: fonction-configuration d'un nouveau mot de passe

✓ **Page de connexion (étudiant)**

Lorsqu'un étudiant clique sur le bouton "Connexion" de la page d'accueil, il est redirigé vers une page où il doit remplir un formulaire de connexion. L'étudiant inscrit doit entrer son numéro matricule et son mot de passe correct.

The screenshot shows a web form titled "Formulaire de Connexion" with a purple header and a white background. The title "Formulaire de Connexion" is at the top center. Below it, the text "Connectez-Vous" is centered. The form has two input fields: "Matricule" and "Mot de passe", each with a placeholder text ("Matricule" and "Mot de passe" respectively). Below the inputs is a link "Mot de passe oublié?". At the bottom is a large purple button labeled "Connexion".

Figure 30: Formulaire de connexion

La fonction permettant à un étudiant de se connecter se présente comme suit :

```
public function Login(Request $request){  
    $request->validate([  
        'matricule' => 'required',  
        'password' => 'required',  
    ]);  
    $credentials = $request->only('matricule', 'password');  
    if(Auth::guard('etudiant')->attempt($credentials)) {  
        $etudiant = Etudiant::where('matricule', $request->input('matricule'))->first();  
        Auth::guard('etudiant')->login($etudiant);  
        return view('etudiant.home', compact('etudiant'));  
    }  
    else {      You, 1 second ago • Uncommitted changes  
        return back()->with('error', 'matricule ou mot de passe invalide');  
    }  
}
```

Figure 31: Fonction gérant la connexion d'un étudiant

- ✓ Page de récapitulatif des informations de l'étudiant

Après s'être connecté, l'étudiant a accès à la page où il a un récapitulatif de ses informations, y compris le montant total de sa scolarité, le montant payé et celui restant.

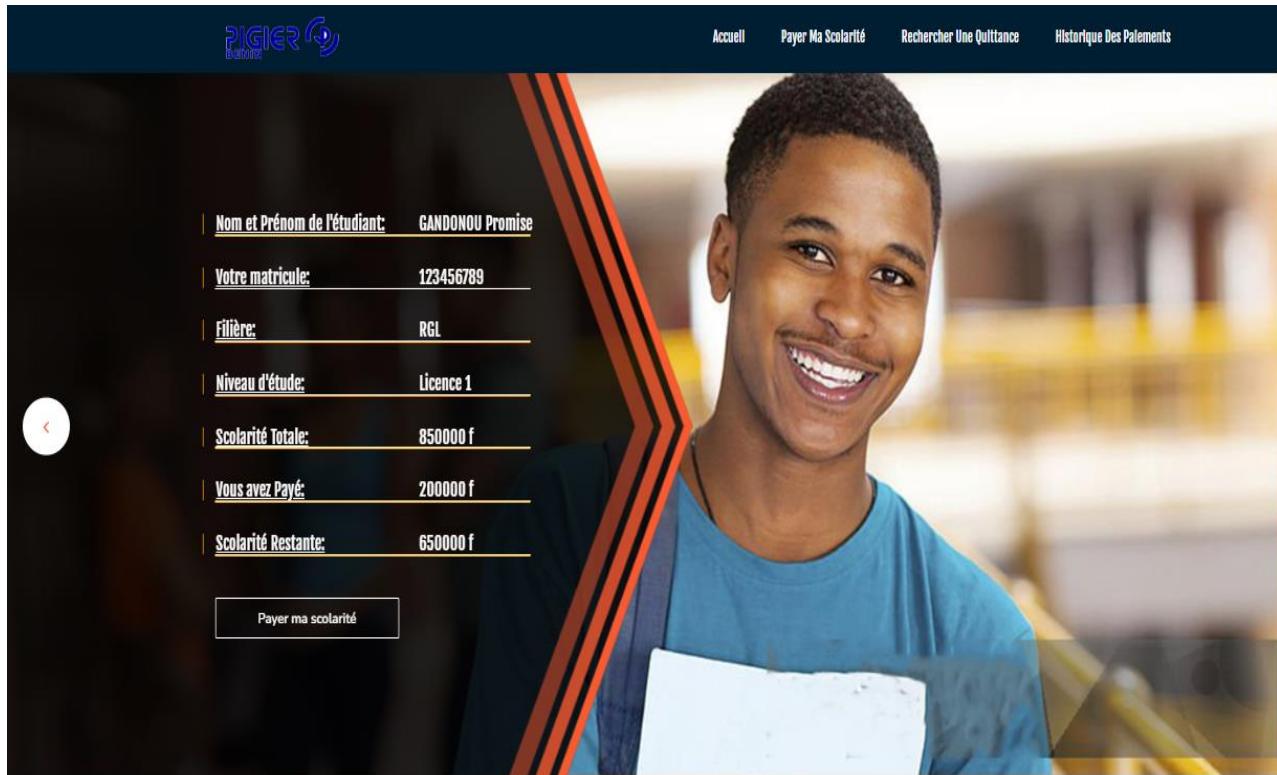


Figure 32: Page de récapitulatif des informations

✓ Formulaire de paiement :

Lorsque l'étudiant (ou le parent se connecte), ce dernier est dirigé vers une interface où il a un récapitulatif de ces informations (figure précédente), ainsi que le solde de sa scolarité. Sur cette même page se trouve le bouton « Payer ma scolarité », qui le redirige vers la page de paiement. Sur cette page, il renseigne le montant qu'il souhaite payer, sachant que les autres champs du formulaire ont été prérempli en tenant compte de l'étudiant actuellement connecté. La figure suivante présente le formulaire de paiement.

Formulaire de paiement

Nom
GANDONOU

prénom
Promise

telephone
56243310

Adresse mail
promisegando@gmail.com

Montant
Entrez le montant

Soumettre

Figure 33: formulaire de paiement

- ✓ Formulaire de détails du paiement envoyé par l'api de FedaPay

Lorsque l'étudiant remplit le formulaire de paiement et le soumet, l'API de FedaPay envoie à ce dernier un formulaire de détail de paiement qu'il renseigne puis valide.

Figure 34: Formulaire de détails de paiement

✓ Formulaire de recherche d'une quittance

Après avoir effectué un paiement via la plateforme, l'utilisateur a la possibilité de rechercher la quittance associée à un paiement qu'il aurait effectué. Il suffit à ce dernier de renseigner la référence de paiement.

La figure suivante présente le formulaire de recherche d'une quittance.

The screenshot shows a search interface with a red border. At the top, the title "Rechercher une quittance" is displayed in a large, bold, dark font. Below the title, there is a label "Référence du paiement" in a smaller, dark font. Underneath this label is a light gray input field with the placeholder text "Référence du paiement". At the bottom of the form is a solid orange button labeled "Soumettre" in white text.

Figure 35: Formulaire de recherche d'une quittance

✓ **Formulaire d'ajout des étudiants (interface comptable)**

Le comptable, après s'être authentifié peut enregistrer un étudiant spécifique. Il peut aussi effectuer l'enregistrement de plusieurs étudiants par le biais d'un fichier Excel qui contient entre autres :

- Leur numéro matricule ;
- Leur nom et prénom ;
- Leur numéro de téléphone ;
- Leur filière et leur année d'étude ;

The screenshot shows a web page with a red border. At the top, the title "enregistrer des étudiants" is displayed in a large, bold, dark font. Below the title, there is a text input field labeled "Charger un fichier". Inside this field, a button labeled "Choisir un fichier" is visible, along with the message "Aucun fichier ...été sélectionné". At the bottom of the form is a large orange button labeled "Soumettre".

Figure 36:Formulaire d'enregistrement des étudiants

✓ **Page « Liste des étudiants » (Interface Chef Comptable)**

Le comptable peut accéder à la liste des étudiants par niveau d'étude et par filière.

GANDONOU Promise

matricule: 123456789
Filière: RGL
Niveau: Licence 3

[VOIR LES DÉTAILS](#) • [MODIFIER](#) • [SUPPRIMER](#) • [HISTORIQUE DE PAIEMENT](#)

DOSSA Luc

matricule: 234567891
Filière: RGL
Niveau: Licence 3

[VOIR LES DÉTAILS](#) • [MODIFIER](#) • [SUPPRIMER](#) • [HISTORIQUE DE PAIEMENT](#)

GOZO Jaures

matricule: 345678912
Filière: RGL

Figure 37: Liste des étudiants

3.2 Discussions

Les différents travaux menés tout au long de ce projet ont abouti à la réalisation d'une plateforme permettant principalement aux étudiants de l'université Pigier de payer plus aisément leur frais de scolarité. Ces derniers pourront, à la suite de leur paiement, rechercher puis télécharger leur quittance, et enfin, consulter l'historique des paiements. Chaque étudiant peut ainsi suivre l'évolution de sa scolarité sans consulter à chaque fois la comptabilité. Le comptable quant à lui, peut enregistrer et mettre à jour les informations des étudiants. Ce dernier a également la possibilité de suivre les paiements effectués par chaque étudiant.

Enfin, l'administrateur de la plateforme de son côté peut gérer les différents utilisateurs.

À travers ces différentes fonctionnalités, la plateforme assure l'automatisation du processus de paiement des frais de scolarité à Pigier. Il sera en effet, plus facile pour un étudiant de payer ses frais de scolarité sans perdre du temps dans une longue file d'attente à la banque. Il pourra avoir un aperçu des différents paiements qu'il a effectué.

Bien que les objectifs principaux de ce projet aient été atteints, il revient de noter que des fonctionnalités supplémentaires pourraient être ajoutées à la solution afin de la rendre plus optimale. Il s'agit entre autres de :

- ✓ La mise en place d'une fonctionnalité de gestion des bourses d'étudiants.
- ✓ L'intégration d'un système de notification automatique sur les échéances et retard de paiement de scolarité.
- ✓ L'ajout d'une fonctionnalité de mise à jour automatique du niveau d'étude d'un étudiant.

Conclusion

Dans ce chapitre, nous avons exposé les résultats obtenus suite à la conception et la réalisation de notre solution. Nous y avons présenté quelques interfaces de l'application. En outre, nous avons proposé des pistes d'amélioration pour optimiser la performance et l'ergonomie de l'application. Cette analyse nous permet de mettre en perspective les accomplissements réalisés et d'identifier les étapes futures pour perfectionner notre solution.



CONCLUSION GENERALE ET PERSPECTIVES

Conclusion générale et Perspectives

Conclusion

Notre projet de fin d'étude portant sur la conception et la réalisation d'une plateforme de paiement des frais de scolarité à Pigier représente une avancée significative dans l'optimisation des processus administratifs et financiers de l'établissement. Dans l'optique de mettre en place cette plateforme, nous avons effectué un stage à SKYRAN, qui nous a permis de mettre en pratique les différentes notions acquises au cours de nos trois années de formation. Pour mieux aborder le projet, nous avons au préalable effectué une analyse de l'existant, puis défini les différentes fonctionnalités que doit assurer la plateforme. Cette analyse a servi de base pour établir la structure de notre plateforme et pour sélectionner judicieusement les différents langages et outils à utiliser dans l'implémentation de la solution. Ce processus a abouti à la mise en place de la plateforme de paiement de frais de scolarité à Pigier. Elle accorde entre autres aux étudiants (ou leurs parents) de payer leurs frais de scolarité et de suivre les différents paiements effectués. Ainsi, cette solution offre une alternative moderne et efficace aux méthodes traditionnelles de paiement, facilitant ainsi la vie des étudiants, des parents et du personnel administratif. Enfin, elle limitera les différentes situations de perte d'argent et de temps.

Perspectives

À l'instar de toute solution informatique, celle que nous avons développée tout au long de ce projet pourra connaître des améliorations, qui incluent l'intégration de nouvelles fonctionnalités telles que les notifications automatisées de rappel de paiement, la gestion des bourses universitaires. La plateforme pourrait également permettre à ses principaux utilisateurs d'avoir un bilan général des paiements effectués.

En somme, la plateforme pourrait évoluer pour devenir un écosystème complet de gestion universitaire, intégrant plusieurs universités et répondant aux besoins de tous les acteurs impliqués tout en tenant compte des exigences technologiques futures.

Références bibliographiques et Webographiques

Bibliographie

- [1] Conallen, J. (2000). *Concevoir des applications Web avec UML*. Eyrolles.
Consulté le Mai 05, 2024
- [2] Engelbert, A. (2023). *UML*.
- [3] HEURTEL, O. (s.d.). *PHP et MySQL Maîtrisez le développement d'un site Web dynamique* (éd. 3). (eni, Éd.)
- [4] Luke Welling, L. T. (2009). *PHP and MySQL Web Development*. Pearson Education. Consulté le Mai 03, 2024
- [5] Matine, O. A. (s.d.). *XHTML & CSS*.

Webographie

- [1] <https://benin.opendataforafrica.org/udeaphd/inscription-et-effectifs-d-%C3%A9tudiants?lang=en> Consulté le 25 Avril 2024, sur Benin Data Portal
- [2] <https://paiement.tresorbenin.bj/#/> Consulté le 28 Février 2024, sur paiement.tresorbenin.bj:
- [3] <https://lepotentiel.bj/2023/06/19/suspension-du-paiement-des-frais-academiques-sur-les-comptes-des-établissements-de-luac-loption-tresor-public-pour-desormais-plus-de-tracabilite/> Consulté le 06 Mai 2024
- [4] <https://www.campusfaso.bf/memoOrangeMoney.pdf> Consulté le 27 Avril 2024, sur CAMPUSFASO :
- [5] <https://edus.education/> Consulté le 28 Février 2024, sur EDUS
- [6] <https://schoolmatik.com/> Consulté le 28 Février 2024, sur SchoolMatik
- [7] <https://creately.com/blog/fr/uncategorized-fr/tutoriel-sur-les-diagrammes-de-classe/#d%C3%A9finition> Consulté le 27 Février 2024, sur creately.com
- [8] <https://ouille.info/technologie/php-definition/#Une-definition-du-PHP-pour-commencer> Consulté le 25 Avril 2024, sur Blog Ouille
- [9] <https://www.cloudflare.com/fr-fr/learning/security/api/what-is-an-api/>
Consulté le 29 Avril 2024

Table des Matières

ENGAGEMENT	
<i>Sommaire.....</i>	<i>I</i>
<i>Dédicaces</i>	<i>III</i>
<i>Dédicaces</i>	<i>IV</i>
<i>Remerciements</i>	<i>V</i>
<i>Liste des figures</i>	<i>VI</i>
<i>Liste des tableaux.....</i>	<i>VIII</i>
<i>Sigles et abréviations.....</i>	<i>IX</i>
<i>Résumé</i>	<i>X</i>
<i>Abstract.....</i>	<i>X</i>
INTRODUCTION	- 1 -
Contexte et justification	- 1 -
Problématique.....	- 1 -
Objectifs du projet de fin de formation	- 2 -
Plan du document	- 2 -
Chapitre 1 : Déroulement du Stage	- 4 -
Introduction.....	- 4 -
1.1 Présentation de la structure	- 4 -
1.2 Fonctionnement de l'entreprise	- 5 -
1.3 Travaux effectués.....	- 6 -
1.4 Apports du stage sur le plan professionnel	- 6 -
1.5 Difficultés rencontrées	- 7 -
Conclusion	- 7 -
Chapitre 2 : Analyse et conception	- 8 -
Introduction.....	- 8 -
2.1 État de l'art	- 8 -
2.1.2 Présentation des solutions existantes	- 9 -
2.2 Présentation du projet :	- 15 -
2.2.1 Présentation des spécifications fonctionnelles et non fonctionnelles ..	- 15 -
15 -	

Table des Matières

2.3 Présentation des règles de gestion	- 16 -
2.4 Analyse UML	- 18 -
2.4.1.1 Identification des acteurs	- 21 -
2.4.1.2 Identification des cas d'utilisations	- 21 -
2.4.1.3 Description des cas d'utilisations.....	- 24 -
2.4.2.1 Modèle Logique des données.....	- 30 -
2.4.3 Diagramme de séquence	- 30 -
2.5 Logiciels et langages de programmation utilisés :	- 36 -
2.5.1 Logiciels utilisés	- 36 -
2.5.2 Langages utilisés :	- 39 -
3.1 2.5.3 Les frameworks utilisés	- 42 -
Conclusion	- 46 -
<i>Chapitre 3 : Résultats et Discussion</i>	- 46 -
Introduction.....	- 46 -
3.1 Résultats	- 46 -
3.2 Discussions	- 55 -
Conclusion	- 56 -
Conclusion générale et Perspectives	- 56 -
Perspectives	- 56 -
Références bibliographiques et Webographiques.....	A
Bibliographie.....	A
Webographie	A
<i>Table des Matières.....</i>	