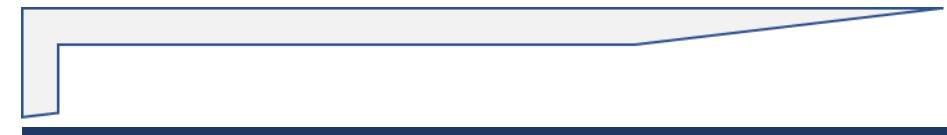


Université Hassan II  
Ecole Supérieure de Technologie  
Département génie informatique

# **Rapport de Projet Fin d'Etude**



## **Conception et réalisation d'une application de notification de paiement**



**Réaliser par :**

Mohammed Yasser

Kawtar BAKIR

Maha ELOUATRI

Loubna BOUKATE

**Encadré par :**

Mme. AFIFI

**Année Universitaire :2022/2023**



## Remerciement :

C'est un devoir agréable d'exprimer en quelques lignes la reconnaissance qu'on doit à tous ceux dont on a sollicité l'aide et la collaboration.

Nous souhaitons remercier, le corps professoral, de nous avoir bien accueilli au sein de cet établissement et la formation de qualité qu'ils nous ont prodigué durant cette année, et spécialement M. Mohamed OUZZIF et M. Khalid BOURAGBA pour tous les efforts extrascolaires et administratifs qu'ils ont veillé à prodiguer sans retenue.

On tient à remercier en particulier Mme Nadia AFIFI pour son aide et ses conseils tout au long des missions évoquées dans ce rapport, et apporté lors des différents suivis.

Nous profitons de cette tribune pour remercier M. Hicham BELHADAoui et toutes les personnes qui de passage, ont pu nous apporter leur contribution, que ce soit au niveau des idées qu'à celui des conceptions. Qu'elles trouvent ici l'expression de notre sincère reconnaissance.

Nous remercions Dieu tout-Puissant de nous avoir permis de mener à terme ce travail qui est pour nous le point de départ d'une merveilleuse aventure, celle de la recherche, source de remise en cause permanente et de perfectionnement perpétuel.

## Résumé

Le projet réalisé lors de la période du PFE vise à développer une application web de paiement pour l'école JOHN NASH SCHOOL.

Cette application facilitera le travail du service financier et d'améliorer en particulier la gestion de paie dans une école.

Avant de passer à l'implémentation, une étude conceptuelle a été réalisée en appliquant les acquis du cours modélisation en UML pour faciliter la réalisation de l'application, et en implémentant Bootstrap5, Django et PostgreSQL qui sont des Framework permettant de créer efficacement des sites complexes et flexibles.

## Table de matière :

Remerciement :	2
Résumé .....	3
Table de matière :	4
Table de figures .....	6
Introduction générale .....	7
Chapitre 1 : Présentation de projet .....	9
1. Introduction .....	10
2. Objectifs .....	10
3. Problématique .....	10
4. Solution proposée .....	11
5. Description de l'application .....	11
6. Conclusion .....	12
Chapitre 2 : Analyse des besoins.....	13
Introduction :	14
Structure de l'application .....	14
Les fonctionnalités de l'application :	14
Avantages de l'application :	15
Gestion de projet :	16
Organigramme Technique des produits : (PBS).....	16
Organigramme Technique des tâches : (WBS) .....	17
Organigramme technique des ressources : (RBS) .....	17
Tableau des tâches :	18
Diagramme de Pert :	18
Diagramme de Gantt :	19
Conclusion :	19
Chapitre 3 : Conception .....	20
Introduction .....	21
Méthode merise :	21
a) Modèle conceptuel de données :	22

b)     Modèle logique de données : .....	24
UML : .....	25
Diagramme de cas d'utilisation .....	26
Acteur 1 : Administrateur .....	28
Diagramme de classe .....	28
Diagramme de séquence : .....	32
Diagramme de déploiement : .....	34
Conclusion .....	35
Chapitre 4 : Outils et technologies utilisés .....	36
Introduction .....	37
HTML5 .....	37
CSS 3 .....	38
JavaScript .....	39
PostgreSQL .....	39
Python .....	40
Django .....	41
Conclusion .....	41
Chapitre 5 : Réalisation .....	42
Introduction : .....	43
Interfaces de l'application : .....	43
Conclusion générale : .....	50
Bibliographies: .....	51

## Table de figures

Figure 1:Organigramme technique des produits .....	16
Figure 2:Organigramme Technique des tâches.....	17
Figure 3:Organigramme technique des ressources.....	18
Figure 4:Diagramme de Pert.....	19
Figure 5:Diagramme de Gantt.....	19
Figure 6:cycle d'abstraction pour la conception des systèmes d'information .....	22
Figure 7:Modèle conceptuel de données .....	23
Figure 8:Modèle logique de données .....	25
Figure 9:Diagramme UML .....	26
Figure 10:Diagramme cas d'utilisation en général .....	27
Figure 11: Diagramme de cas d'utilisation .....	28
Figure 12:Diagramme de classe en général .....	30
Figure 13:Diagramme de classe .....	32
Figure 14:Diagramme de séquence .....	34
Figure 15:Diagramme de déploiement .....	35
Figure 16:Dashboard .....	43
Figure 17:Dashboard (suite) .....	43
Figure 18:Sign up Interface .....	44
Figure 19:Page administrateur .....	44
Figure 20:Table élève .....	45
Figure 21:Table paiement .....	45
Figure 22:Formulaire de paiement .....	46
Figure 23:Facture PDF de paiement.....	46
Figure 24:Table des administrateurs .....	47
Figure 25:Notification des impayés .....	47
Figure 26:Mode de SideBar .....	48
Figure 27: To Do List .....	49
Figure 28:Recherche élève.....	49

## Introduction générale

La technologie des applications Web a évolué très rapidement, de nos jours, où plusieurs logiciels Web réservés à cet usage sont sortis sur le marché un après l'autre, les écoles de leur part utilisent cette plateforme pour accomplir l'automatisation et la simplification des tâches et pour offrir une communication fluide et disponible en permanence entre ses administrateurs, ses professeurs et les clients (parents).

Dans cette optique, on a réalisé un projet dont notre mission était de développer une application web qui permet de faciliter le processus de la facturation et du paiement.

Le serveur d'application est le système d'exploitation qui héberge l'application de gestion de paiement. C'est là où l'application est installée et s'exécute.

La base de données est un système de gestion de base de données qui stocke toutes les données relatives aux paiements des étudiants.

L'application web est l'interface utilisateur qui permet aux utilisateurs d'interagir avec le système pour effectuer des paiements, consulter leur historique de paiement, etc. Les navigateurs des utilisateurs sont les clients qui accèdent à l'application via Internet.

En résumé, les utilisateurs accèdent à l'application via leur navigateur web, qui communique avec l'application hébergée sur le serveur d'application.

L'application interagit avec la base de données pour stocker et récupérer les données nécessaires.

Le présent mémoire a pour but d'explicitier la démarche adoptée afin de réaliser les objectifs et les défis que représente ce projet et met le point sur les procédés technologiques dont nous avons fait usage pour aboutir à un résultat palpable. Il est structuré en quatre chapitres :

**Chapitre 1 : Présentation du projet :** Ce chapitre présente un contexte général du projet : la capture et l'analyse détaillée de l'application et ses objectifs généraux.

**Chapitre 2 : Analyse des besoins :** Ce chapitre fera l'objet de capture des besoins techniques.



**Chapitre 3 : Conception :** Ce chapitre décrit la méthodologie de développement qu'on a opté pour suivi de la planification des tâches à la spécification des besoins fonctionnels et des différents modèles de conception.

**Chapitre 4 : Outils et logiciels utilisés :** Dans ce chapitre seront exposés l'ensemble des technologies utilisées lors de la phase de développement.

**Chapitre 5 : Réalisation :** Ce dernier chapitre présente la phase de réalisation et d'intégration de la solution, où nous décrivons le travail réalisé.

# Chapitre 1 : Présentation de projet

# 1. Introduction

La présentation du cadre général du projet a pour but de situer le projet dans son environnement organisationnel et contextuel. Ce chapitre décrit la problématique à traiter et présente la démarche suivie pour la réalisation du projet.

## 2. Objectifs

Les objectifs d'une application de paiement scolaire pour l'administrateur peuvent inclure :

- **Simplifier la gestion des paiements :** L'application peut permettre à l'administrateur de suivre facilement les paiements effectués, de générer des reçus et des rapports de paiement, et de simplifier la comptabilité.
- **Réduire les tâches administratives :** En automatisant la gestion des paiements, l'application peut aider l'administrateur à réduire les tâches administratives liées à la collecte de paiements, telles que la vérification des paiements, la mise à jour des registres de paiement, la création de reçus, etc.

En résumé, une application de paiement scolaire peut aider l'administrateur à simplifier, sécuriser et faciliter la gestion des paiements liés à l'éducation, tout en réduisant les tâches administratives, en améliorant la communication et en accroissant la transparence.

## 3. Problématique

Pour gérer les factures et les paiements, les écoles s'appuient sur le tableur Excel.

Cette technique entraîne des problèmes sur plusieurs niveaux :

- **La cohérence de l'état financier :** une mise à jour n'est pas automatiquement distribuée ce qui peut causer des documents erronés, en plus, sans parcourir des listes élément par élément les administrateurs ne peuvent pas savoir une mise à jour effectuée par un autre administrateur.
- **L'accès et modification:** pour accéder à une information il faut parcourir tous le(s) tableau(x) aussi si une ligne n'est pas ajoutée dans l'ordre on doit refaire toutes les lignes qui suivent ce qui consomme du temps.
- **Sécurité :** la main de manipulation est donnée à n'importe qui et n'est pas contrôlée en plus que ces fichiers sont faciles à perdre.

- **Confidentialité** : n'importe qui peut savoir les détails des activités de l'école.

A partir de toutes ces observations, nous constatons qu'il y a un manque dans la structuration du travail au sein de l'école. En effet, Il n'y a pas une bonne gestion et une attribution claire des rôles et des tâches pour chaque intervenant dans l'école.

## 4. Solution proposée

On souhaite créer une application web qui gère le processus de facturation scolaire, cette gestion comprend la gestion des paiements, la création, la modification... la gestion de la facturation en plus du suivi des opérations financières de l'école.

Cette application web sera l'intermédiaire entre les administrateurs qui font partie de cette procédure en offrant les services suivants :

- Assurer la confidentialité et la sécurité : la nécessité d'une authentification et la limitation d'accès par rapport à chaque utilisateur.
- Assurer le partage immédiat des mises à jour.
- Mettre en place un système de notifications.
- Simplifier l'accès aux informations grâce aux filtres.
- Garder trace de toutes les activités financières.

## 5. Description de l'application

Une application de paiement scolaire pour un administrateur est un logiciel qui permet aux établissements éducatifs de gérer les paiements des frais de scolarité, des fournitures scolaires, des activités parascolaires et autres frais liés à l'éducation. Cette application permet aux administrateurs de suivre les paiements des étudiants, de gérer les comptes des étudiants, de générer des rapports financiers et d'automatiser les processus de facturation.

Voici une description détaillée des fonctionnalités qu'une telle application pourrait offrir à un administrateur :

- **Tableau de bord** : L'application fournit un tableau de bord qui permet aux administrateurs de voir rapidement les statistiques financières importantes, telles que le total des revenus, le total des frais impayés et le nombre d'étudiants inscrits.

- **Gestion des frais** : L'application permet aux administrateurs de créer et de gérer les frais liés à l'éducation, y compris les frais de scolarité, les frais de fournitures scolaires, les frais d'activités parascolaires et autres frais. Les administrateurs peuvent également fixer les dates d'échéance et les montants pour chaque frais.
- **Suivi des paiements** : L'application permet aux administrateurs de suivre les paiements des étudiants, y compris les paiements effectués, les paiements en attente et les paiements en retard. Les administrateurs peuvent également voir les détails des paiements, tels que les dates et les montants.
- **Gestion de compte** : L'application permet aux administrateurs de gérer les comptes des étudiants, y compris les informations personnelles, les coordonnées et les détails de paiement. Les administrateurs peuvent également ajouter ou supprimer des étudiants de leur compte et gérer les autorisations de paiement pour chaque étudiant.
- **Rapports financiers** : L'application fournit des rapports financiers détaillés pour aider les administrateurs à suivre les revenus et les dépenses de l'établissement. Les rapports peuvent être filtrés par date, frais, étudiant et autres critères.

## 6. Conclusion

Ce chapitre avait pour objet la présentation du cadre général de mon PFE qui consiste sur la réalisation d'une application de notification de paiement. Ainsi, nous avons présenté le contexte du projet, la problématique à laquelle il vient répondre ainsi que ses solutions afin de mettre le projet dans un cadre d'étude transversale avant de pouvoir l'analyser et spécifier les besoins fonctionnels.

# Chapitre 2 : Analyse des besoins

## Introduction :

L'étude technique consiste à mener une analyse des besoins techniques puis trouver une implémentation qui répond à ces besoins indépendamment des choix fonctionnels. Ce chapitre fera donc l'objet de capture des besoins techniques.

## Structure de l'application

Une application de paiement scolaire pour l'administrateur pourrait être structurée de la manière suivante :

- **Tableau de bord :** Une fois connecté, l'utilisateur aurait accès à son tableau de bord. Celui-ci afficherait les informations de paiement en cours et les transactions passées.
- **Notifications :** L'administrateur pourrait recevoir des notifications des étudiants qui n'ont pas payer.
- **Paramètres de compte :** Les utilisateurs pourraient gérer leur profil, ajouter des méthodes de paiement et mettre à jour leurs informations personnelles.
- **Sécurité :** Une attention particulière devrait être portée à la sécurité de l'application, y compris la protection des données personnelles et des informations de paiement.
- **Rapports :** L'administrateur pourrait accéder à des rapports sur les paiements pour suivre les transactions, identifier les tendances et évaluer la situation financière de l'école.

## Les fonctionnalités de l'application :

- **Notification en temps réel :** L'application envoie des notifications en temps réel aux administrateurs pour les paiements dus et effectués.
- **Tableau de bord :** L'application de paiement scolaire doit offrir un tableau de bord pour les administrateurs qui leur permet de surveiller les paiements effectués, les factures en attente et les rapports de paiement. Le tableau de bord doit offrir des fonctionnalités de filtrage et de tri pour une recherche facile des informations.

- **Gestion des factures** : Les administrateurs doivent pouvoir générer des factures pour les frais de scolarité. Les factures doivent inclure les détails du paiement, les dates et les montants dus.
- **Rapports financiers** : L'application de paiement scolaire doit offrir des rapports financiers pour permettre aux administrateurs de surveiller les paiements effectués, les factures en attente et les remboursements effectués. Les rapports doivent être personnalisables pour répondre aux besoins spécifiques des établissements scolaires.

## Avantages de l'application :

- **Facilité d'utilisation** : L'application de paiement scolaire est facile à utiliser et offre une expérience utilisateur conviviale pour les parents/tuteurs et les établissements scolaires.
- **Sécurité** : L'application utilise une passerelle de paiement sécurisée pour garantir la sécurité des informations de paiement.
- **Suivi des paiements** : Les administrateurs peuvent suivre les paiements effectués et en attente à partir du tableau de bord de l'application de paiement scolaire. Cela permet aux administrateurs de surveiller les paiements et de s'assurer que les paiements sont effectués à temps.
- **Réduction des coûts** : Les applications de paiement scolaire permettent de réduire les coûts liés à la gestion manuelle des paiements, tels que les coûts de personnel, les coûts de traitement de papier et les coûts de traitement des chèques. Cela permet aux administrateurs d'allouer les ressources financières et humaines à d'autres tâches importantes.
- **Accessibilité aux données** : Les applications de paiement scolaire fournissent aux administrateurs un accès facile aux données de paiement des étudiants, ce qui leur permet de mieux comprendre les tendances de paiement et de planifier les budgets de manière plus efficace. Les administrateurs peuvent également générer des rapports sur les paiements effectués, les factures en attente et les remboursements.
- **Réduction des tâches administratives** : Les applications de paiement scolaire éliminent la nécessité de gérer manuellement les paiements, ce qui réduit la charge de travail administrative des administrateurs. Cela permet aux administrateurs de se



concentrer sur d'autres tâches importantes, telles que la planification stratégique et la gestion des ressources humaines.

En somme, les applications de paiement scolaire offrent de nombreux avantages pour les administrateurs, notamment une gestion efficace des paiements, une réduction des coûts, une accessibilité aux données, une communication efficace et une réduction des tâches administratives.

## Gestion de projet :

### Organigramme Technique des produits : (PBS)

L'organigramme technique des produits (PBS) est un outil de gestion de projet qui représente la décomposition hiérarchique des produits en sous-produits, composants et autres éléments constitutifs nécessaires à leur réalisation.

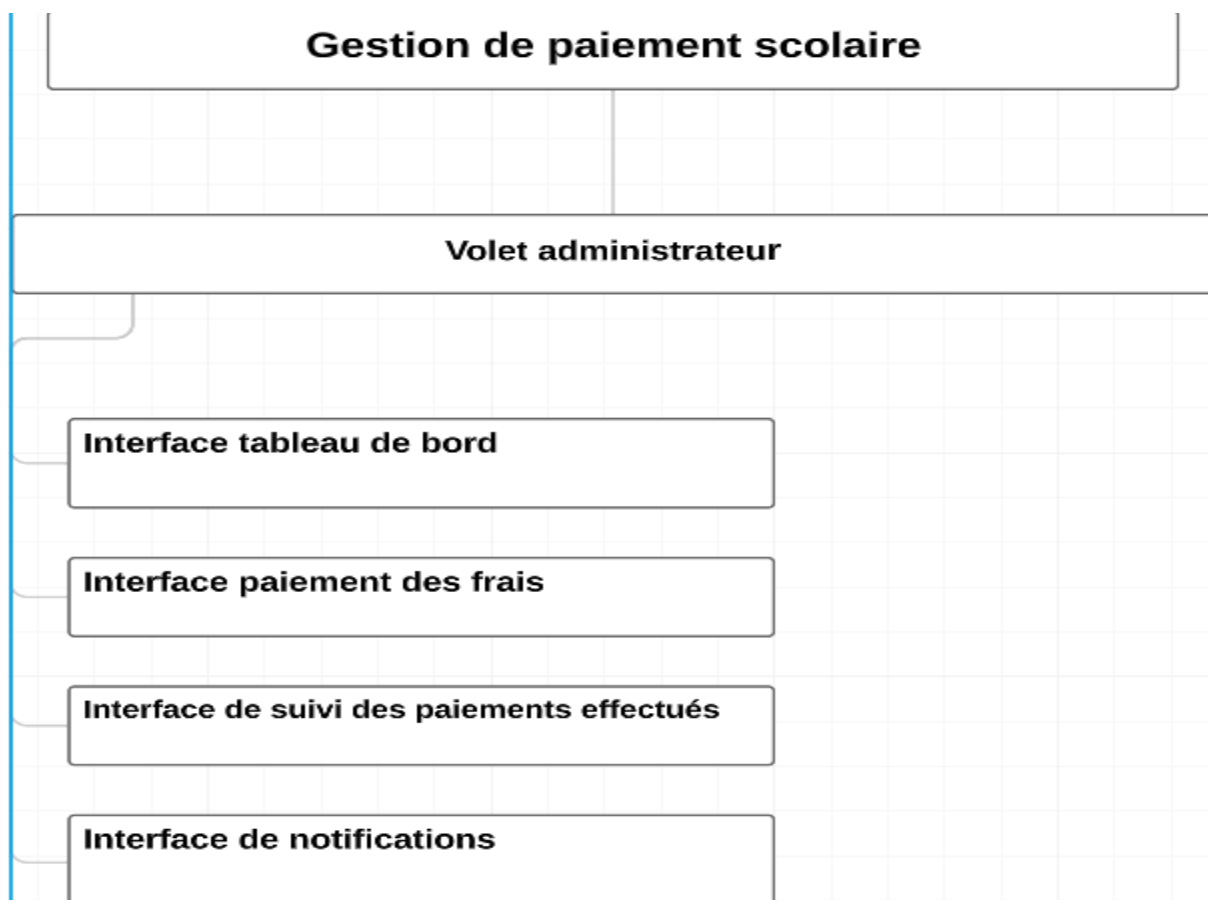


Figure 1: Organigramme technique des produits

## Organigramme Technique des tâches : (WBS)

L'organigramme technique des tâches, ou WBS (Work Breakdown Structure), est un outil de gestion de projet qui permet de décomposer un projet en tâches plus petites et plus gérables.

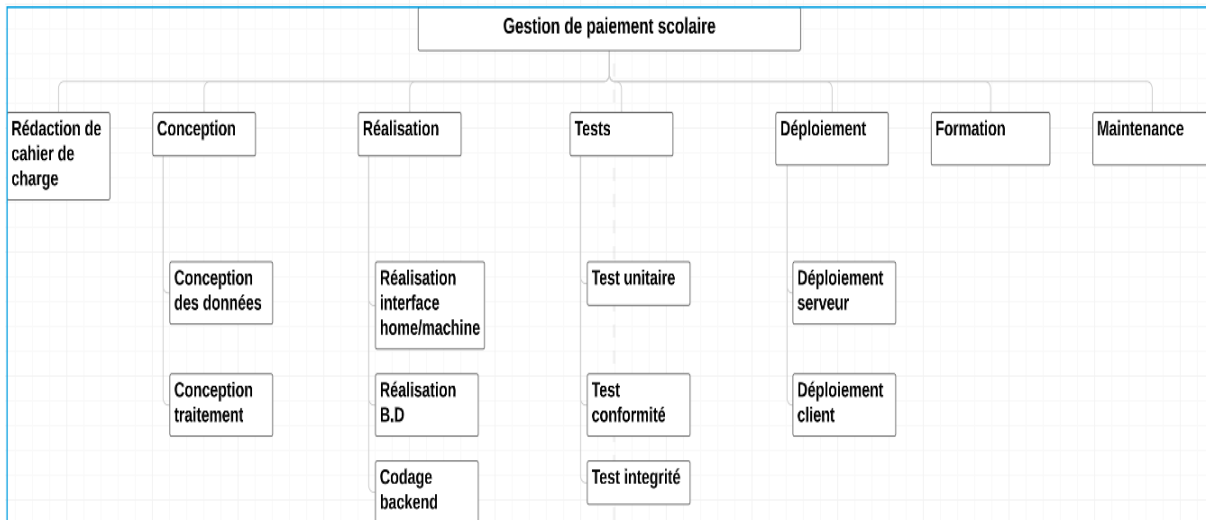


Figure 2: Organigramme Technique des tâches

## Organigramme technique des ressources : (RBS)

L'organigramme technique des ressources, également appelé la structure de la répartition des ressources (RBS), est un outil de gestion de projet qui permet de décomposer les différentes ressources nécessaires à la réalisation d'un projet en une hiérarchie de sous-catégories.

Le RBS permet d'évaluer la quantification des ressources nécessaires au projet, dont la charge est issue des estimations du temps à passer par tâche et par profil.

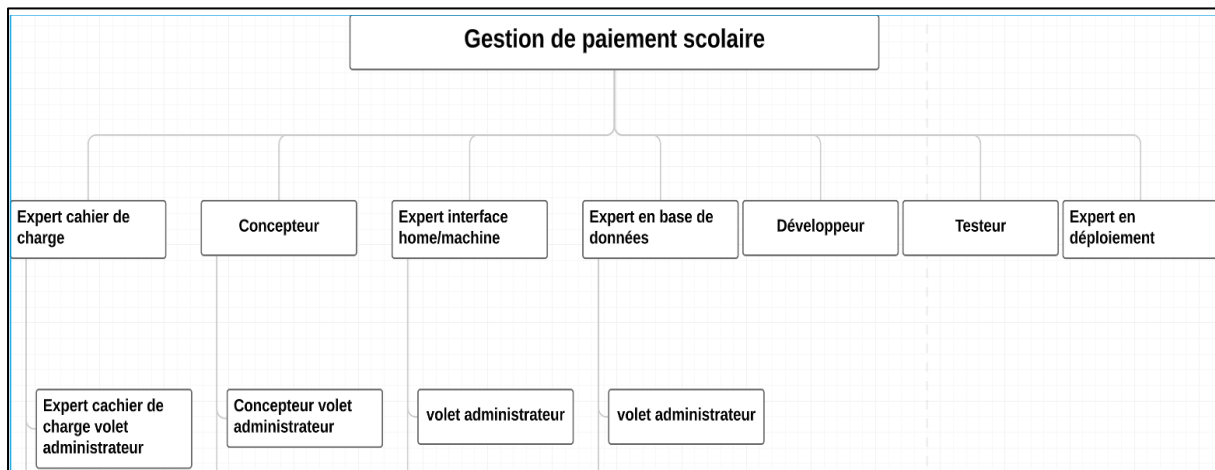


Figure 3: Organigramme technique des ressources

## Tableau des tâches :

Nom	Nom de la tâche	Durée	Antécédent
A	Discuter le projet	1	-
B	Discuter les outils	2	A
C	Conception	7	B
D	Développement du front-end	5	C
E	Développement du back-end	12	C
F	Liaison du front-end et back-end	15	D,E

## Diagramme de Pert :

Un diagramme de PERT est un outil servant à analyser les différentes tâches qui entrent dans l'exécution d'un projet. Pensés pour les projets d'envergure, les diagrammes PERT permettent de déterminer le temps requis pour chaque tâche. Résultat : on peut aisément estimer le délai de réalisation (minimum, maximum et médian) du projet dans son ensemble.

Les diagrammes PERT offrent une visualisation claire de la durée de travail requise et de l'interdépendance des tâches, nous aidant ainsi à mieux gérer, entre autres, le séquençage des tâches et les échéances. Grâce à ce document exhaustif, notre équipe peut réaliser les tâches plus rapidement, de façon précise, sans disperser leurs efforts.

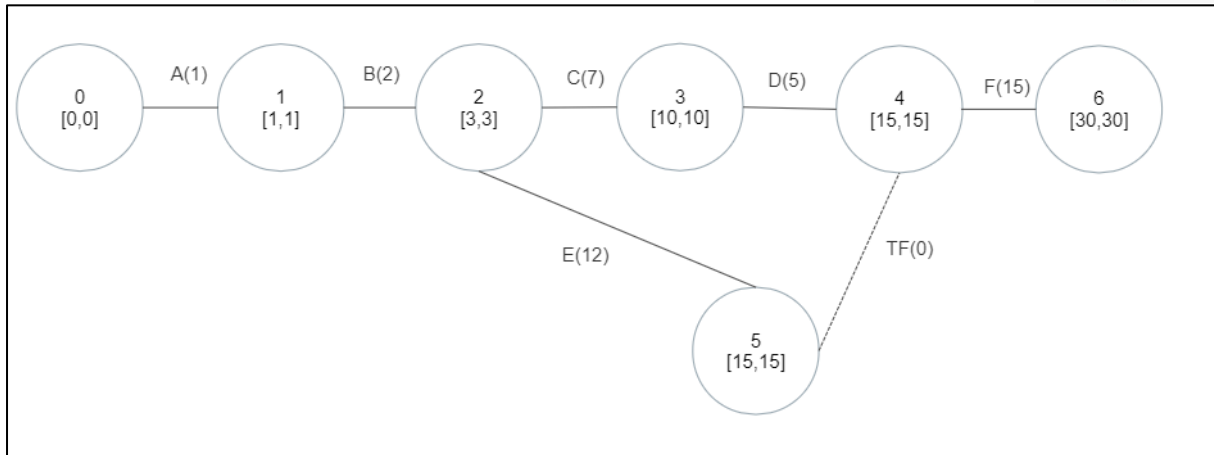


Figure 4:Diagramme de Pert

## Diagramme de Gantt :

Un diagramme de Gantt, par définition, est un graphique composé de barres indiquant les étapes (ou tâches) nécessaires à la réalisation d'un projet selon une échelle de temps. Ces tâches sont affectées à différents membres de l'équipe et représentées par des barres qui concordent avec les dates d'échéance figurant dans la partie supérieure du graphique.

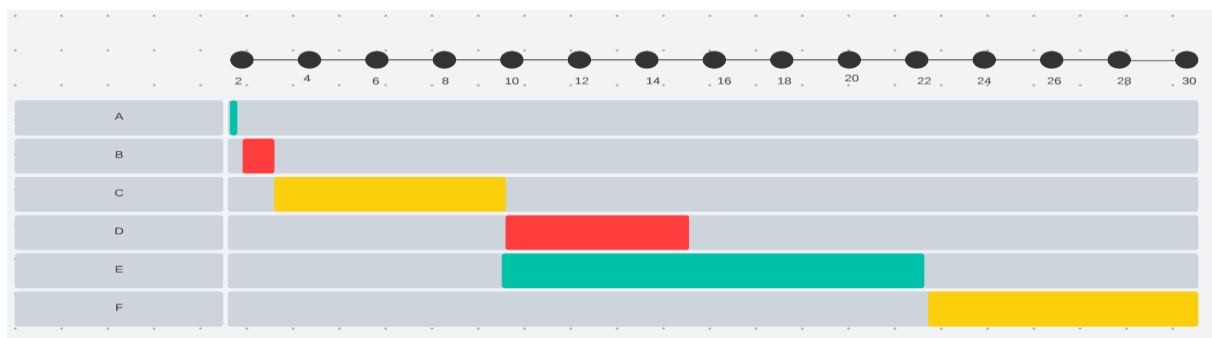


Figure 5:Diagramme de Gantt

## Conclusion :

L'application de paiement scolaire offre une solution technologique innovante pour les établissements scolaires qui cherchent à gérer les paiements des frais de scolarité de manière efficace et sécurisée. Les fonctionnalités de l'application pour les administrateurs permettent une gestion facile et transparente des paiements.

# Chapitre 3 : Conception

## Introduction

Dans cette partie, on va modéliser les besoins avec le langage UML, suivi par une conception du fonctionnement du projet qui comporte une description du fonctionnement de la solution.

Concevoir une application est un processus qui peut être relativement complexe. En effet, lors de l'élaboration d'un projet, de nombreuses données, concept, processus métiers, acteurs et autres se mélangent et peuvent très vite donner le tournis. La modélisation UML fournit un formalisme relativement simple à appréhender mais qui s'avère très puissant à l'utilisation. Il sert notamment de support à la réflexion en permettant de modéliser les interactions entre les différents utilisateurs et la solution logicielle.

## Méthode merise :

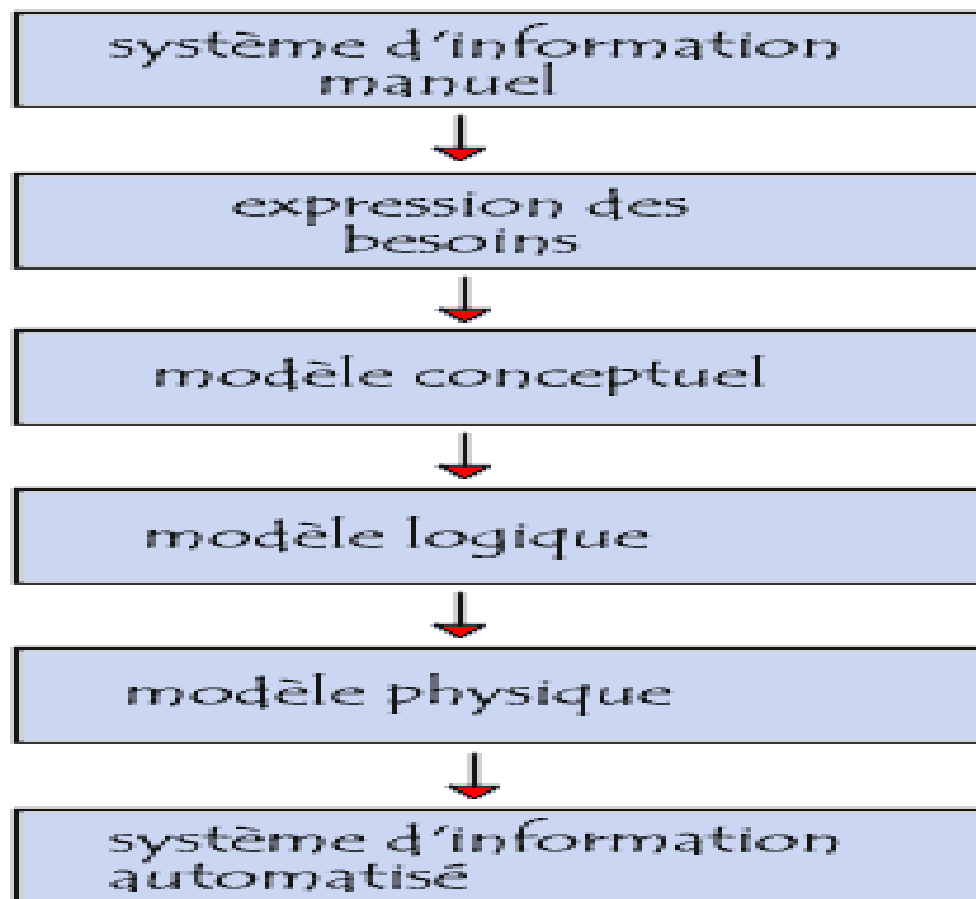
MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

La méthode MERISE date de 1978-1979, et fait suite à une consultation nationale lancée en 1977 par le ministère de l'Industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information. Les deux principales sociétés ayant mis au point cette méthode sont le CTI (Centre Technique d'Informatique) chargé de gérer le projet, et le CETE (Centre d'Etudes Techniques de l'Equipement) implanté à Aix-en-Provence.

La conception du système d'information se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance entre données et

traitements afin de vérifier que toutes les données nécessaires aux traitements sont présentes et qu'il n'y a pas de données superflues.

Cette succession d'étapes est appelée cycle d'abstraction pour la conception des systèmes d'information :



*Figure 6:cycle d'abstraction pour la conception des systèmes d'information*

### **a) Modèle conceptuel de données :**

Un modèle conceptuel de données (MCD) est une représentation visuelle des concepts, des relations et des règles qui décrivent les données d'un système d'information. Le modèle conceptuel de données est généralement utilisé pour planifier, concevoir et décrire les données qui seront stockées et utilisées dans un système.

Le MCD est construit en utilisant un langage de modélisation de données, tel que le modèle entité-association (ER), le modèle objet-relationnel (ORM), ou le modèle relationnel

(MR). Le choix du langage de modélisation dépend du contexte et des objectifs du système d'information.

Le modèle conceptuel de données est souvent utilisé en amont de la conception d'une base de données, afin de comprendre les besoins en données du système et de garantir que toutes les données pertinentes sont prises en compte. Il peut également être utilisé pour communiquer avec les parties prenantes et pour faciliter la compréhension des exigences du système.

Les éléments clés d'un modèle conceptuel de données comprennent les entités, les attributs, les relations et les contraintes. Les entités sont des objets du monde réel ou du système d'information, tels que les clients, les commandes, les produits ou les employés. Les attributs sont les caractéristiques des entités, tels que les noms, les adresses, les numéros de téléphone ou les dates de naissance. Les relations sont les associations entre les entités, telles que la relation entre un client et une commande. Les contraintes sont les règles qui définissent la validité et l'intégrité des données, telles que les contraintes de clé primaire et de clé étrangère.

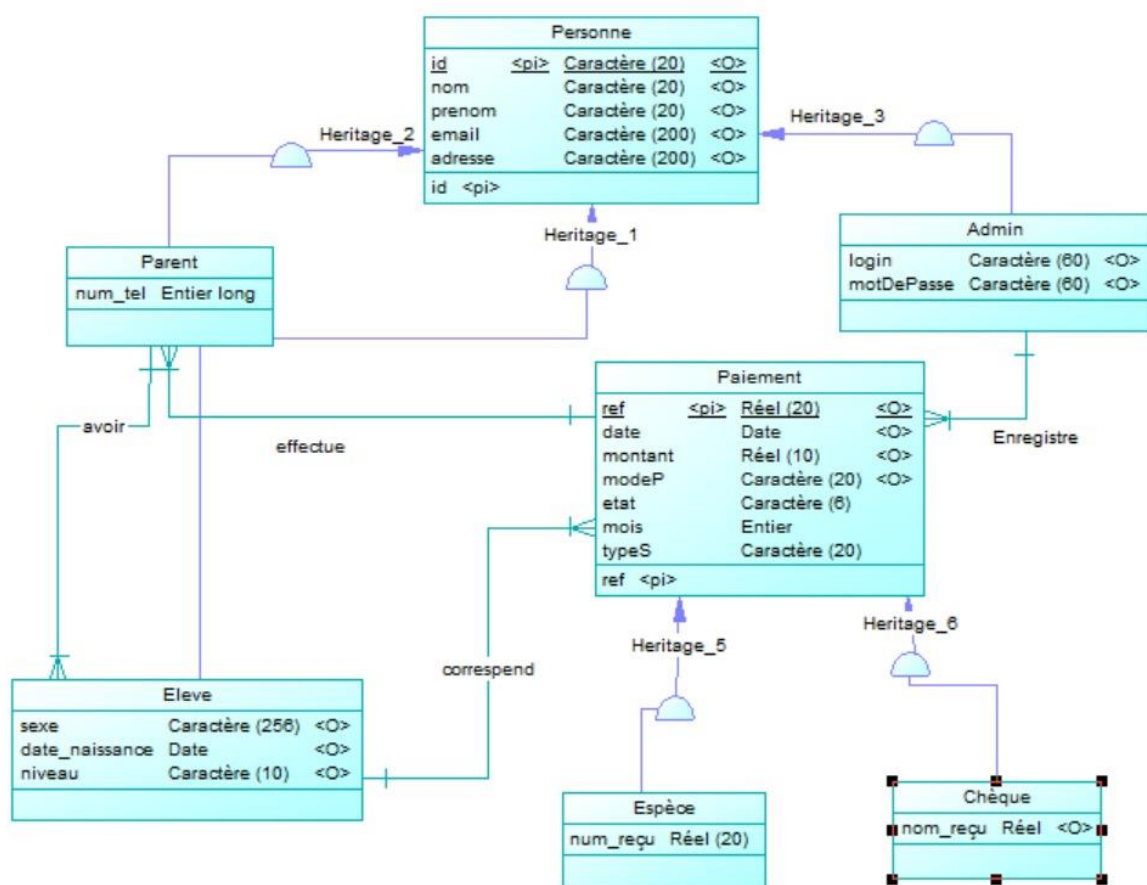


Figure 7:Modèle conceptuel de données



## **b) Modèle logique de données :**

Un modèle logique de données est une représentation abstraite des données d'un système d'information qui se concentre sur les relations entre les différents types de données. Le modèle logique de données est utilisé pour décrire la structure de la base de données et la façon dont les données sont organisées. Il peut être considéré comme une étape intermédiaire entre le modèle conceptuel de données et le modèle physique de données.

Le modèle logique de données est souvent représenté sous forme de diagramme entité-association (E/A). Ce type de diagramme utilise des symboles graphiques pour représenter les entités, les attributs et les relations entre les entités. Les entités représentent des objets du monde réel, tels que des personnes, des lieux ou des choses, tandis que les attributs représentent les caractéristiques de ces objets. Les relations entre les entités indiquent comment les entités sont liées les unes aux autres.

Le modèle logique de données est généralement utilisé comme point de départ pour la conception de la base de données. Il permet de définir la structure de la base de données et de préciser les contraintes qui s'appliquent aux données. Le modèle logique de données peut également être utilisé pour communiquer avec les parties prenantes du système d'information, telles que les utilisateurs et les développeurs, afin de s'assurer que tout le monde a une compréhension commune de la structure de la base de données et de son fonctionnement.

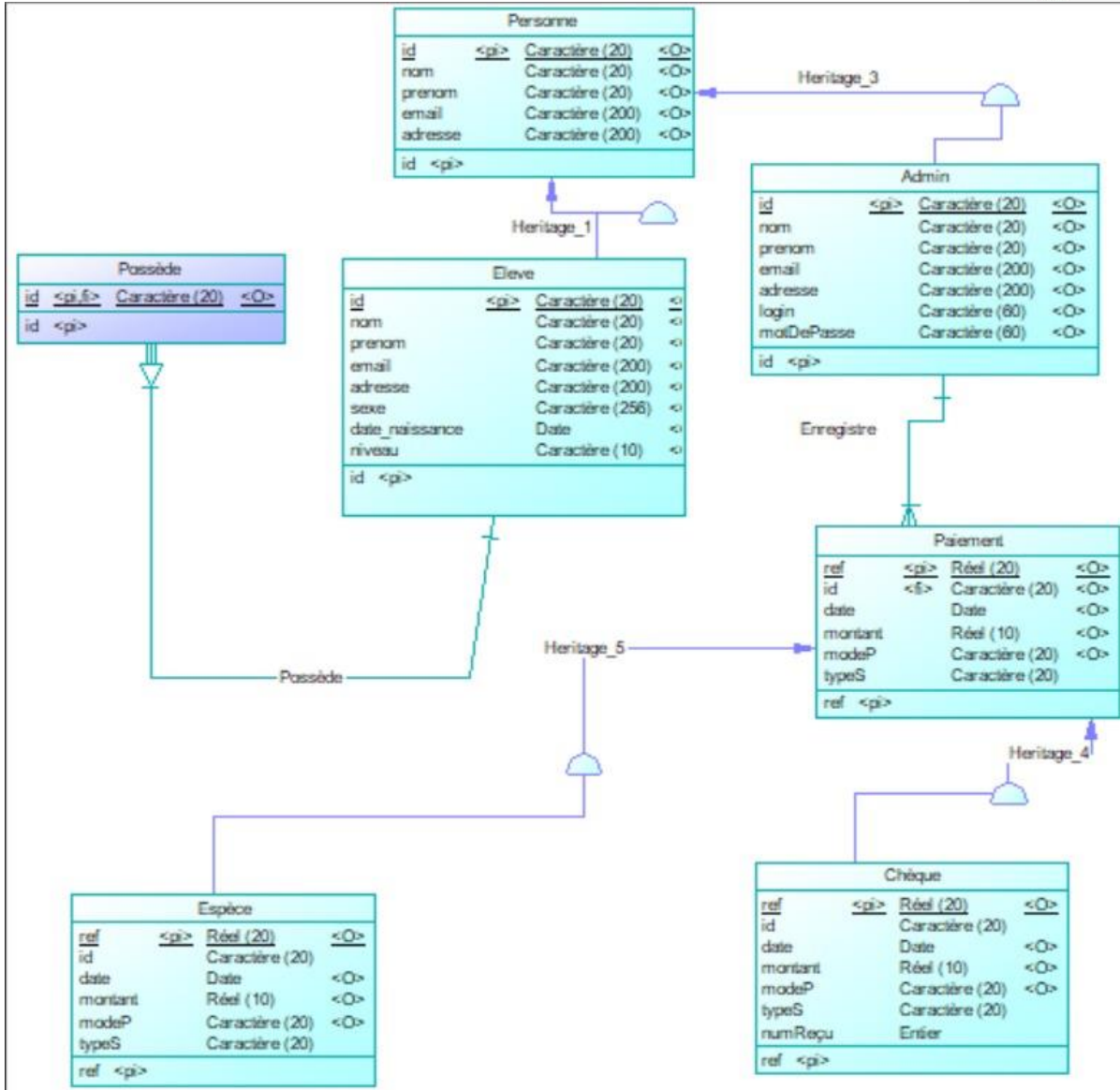


Figure 8:Modèle logique de données

## UML :

Pour modéliser d'une manière claire et précise la structure et le comportement de notre système indépendamment de tout langage de programmation, je vais adopter la démarche le langage de modélisation UML (Unified Modeling Language).

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

UML représente le standard de modélisation objet le plus répandu et le plus utilisé aujourd'hui. Il s'articule autour de treize types de diagrammes (Les diagrammes d'UML comme le montre la Figure, chacun d'eux étant dédié), la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes ; des Diagrammes structurels et comportementaux.

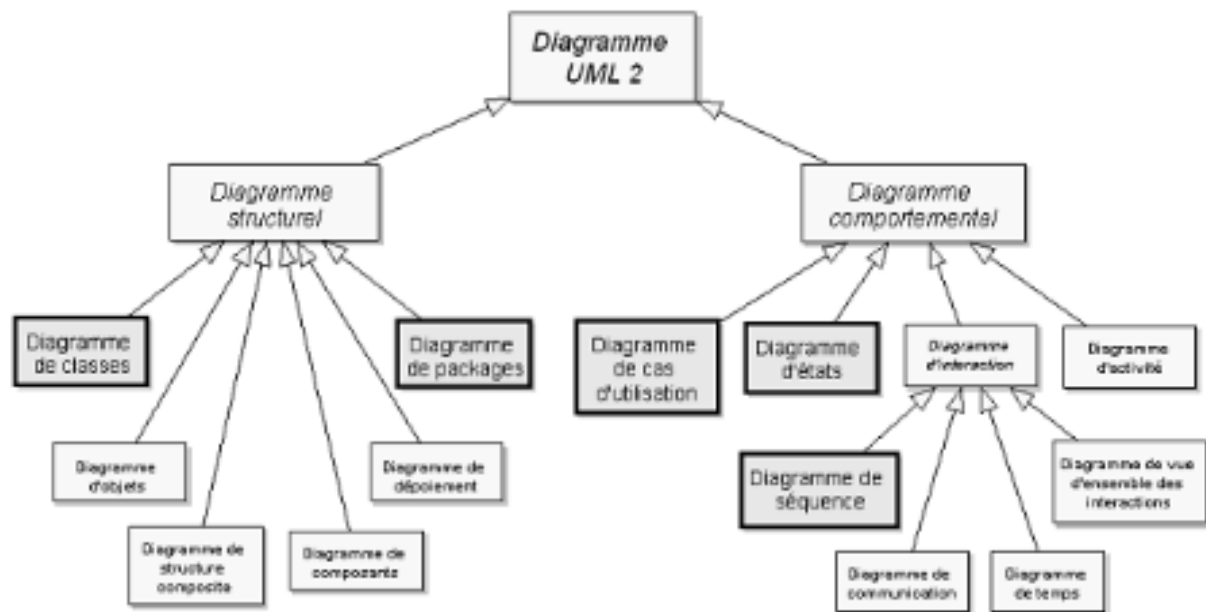


Figure 9: Diagramme UML

## Diagramme de cas d'utilisation

En langage UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système.

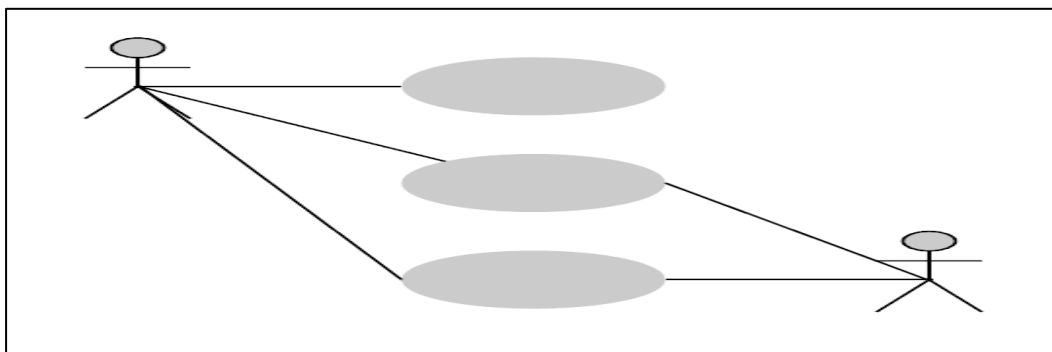
Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne.

Les diagrammes de cas d'utilisation illustrent et définissent le contexte et les exigences d'un système entier, ou des parties essentielles d'un système. On peut modéliser un système complexe avec un seul diagramme de cas d'utilisation, ou créer de nombreux diagrammes de

cas d'utilisation pour modéliser les composants du système. On développe des diagrammes de cas d'utilisation essentiellement dans les premières phases d'un projet et on s'y réfèrera tout au long du processus de développement.

Les diagrammes de cas d'utilisation sont utiles dans les situations suivantes :

- Avant de commencer un projet, on peut créer des diagrammes de cas d'utilisation pour modéliser une entreprise, afin que tous les participants au projet visualisent bien les travailleurs, clients et activités de l'entreprise.
- Lors du recueil des exigences, on peut créer des diagrammes de cas d'utilisation pour capturer les exigences du système et pour présenter aux autres ce que le système doit faire.
- Lors des phases d'analyse et de conception, on peut utiliser les cas d'utilisation et les acteurs de nos diagrammes de cas d'utilisation pour identifier les classes nécessaires.
- Lors de la phase de test, on peut utiliser les diagrammes de cas d'utilisation pour identifier les tests à réaliser pour le système.



*Figure 10:Diagramme cas d'utilisation en général*

Les rubriques suivantes décrivent les éléments de modèle dans les diagrammes de cas d'utilisation :

### **Cas d'utilisation:**

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

### **Acteurs**

Un acteur représente un rôle d'un utilisateur qui interagit avec le système qu'on

modélise. L'utilisateur peut être un utilisateur humain, une organisation, une machine ou un autre système externe.

### Sous-systèmes

Dans les modèles UML, les sous-systèmes sont un type de composant stéréotypé représentant des unités comportementales indépendantes dans un système. Les sous-systèmes sont utilisés dans les diagrammes de classes, de composants et de cas d'utilisation pour représenter des composants de grande taille dans le système qu'on modélise.

### Relations dans les diagrammes de cas d'utilisation :

En langage UML, une relation est une connexion entre des éléments de modèle. Une relation UML est un type d'élément de modèle qui ajoute une sémantique à un modèle en définissant la structure et le comportement entre les éléments de modèle.

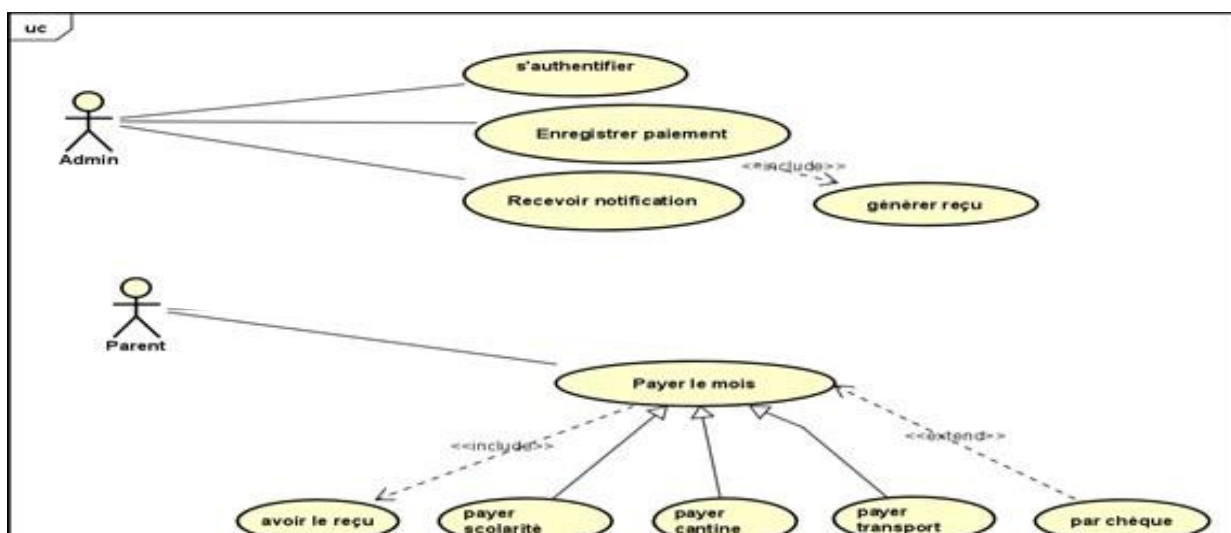


Figure 11: Diagramme de cas d'utilisation

### Acteur 1 : Administrateur

L'administrateur enregistre les paiements effectués pour les étudiants puis générer des reçus afin de les donner aux parents, enfin il reçoit des notifications automatiquement au cas où il y'a des retards au niveau des paiements.

## Diagramme de classe

Dans le langage UML, les diagrammes de classes appartiennent à l'un des six types de diagramme structurel. Les diagrammes de classes sont fondamentaux pour le processus de modélisation des objets et modélisent la structure statique d'un système. Suivant la complexité d'un système, on peut utiliser un seul diagramme de classes pour modéliser un système complet ou bien on peut utiliser différents diagrammes de classes pour modéliser les composants d'un système.

Les diagrammes de classes sont les plans de notre système ou sous-système. On peut utiliser des diagrammes de classes pour modéliser les objets qui constituent le système, pour afficher les relations entre les objets et pour décrire ce que ces objets font et les services qu'ils fournissent.

Les diagrammes de classes sont utiles à de nombreuses étapes de la conception d'un système. Lors de l'étape d'analyse, un diagramme de classes peut nous aider à comprendre les exigences de notre domaine de problème et à identifier ses composants. Dans un projet logiciel orienté objet, les diagrammes de classes qu'on crée pendant les premières étapes du projet contiennent des classes qui se traduiront souvent par des classes et objets logiciels réels lorsqu'on écrira le code. Par la suite, on peut détailler nos modèles d'analyse et de conception antérieurs dans des diagrammes de classes montrant des composants spécifiques de notre système, des interfaces utilisateur, des implémentations logiques, etc. Nos diagrammes de classes deviennent alors une image instantanée décrivant exactement comment notre système travaille.

On peut utiliser des diagrammes de classes pour visualiser, définir et documenter des fonctions structurelles dans nos modèles. Par exemple, pendant les phases d'analyse et de conception du cycle de développement, on peut créer des diagrammes de classes pour réaliser les fonctions suivantes :

- Capturer et définir la structure des classes et autres discriminants
- Définir les relations entre les classes et discriminants
- Illustrer la structure d'un modèle à l'aide d'attributs, d'opérations et de signaux
- Afficher les rôles et responsabilités de discriminant communs qui définissent le comportement du système
- Afficher les classes d'implémentation dans un package
- Afficher la structure et le comportement d'une ou plusieurs classes

- Afficher une hiérarchie d'héritage entre des classes et discriminants
- Afficher les travailleurs et entités comme des modèles objet métier

Pendant la phase d'implémentation d'un cycle de développement d'application, on peut utiliser des diagrammes de classes pour convertir nos modèles en code et pour convertir notre code en modèles.



Figure 12: Diagramme de classe en général

Les rubriques suivantes décrivent les éléments de modèle dans les diagrammes de classes :

## Classes

Dans le langage UML, une classe représente un objet ou un ensemble d'objets qui partagent une structure et un comportement communs. Les classes ou instances de classes, sont des éléments de modèle communs dans les diagrammes UML.

## Objets

Dans les modèles UML, les objets sont des éléments de modèle représentant les instances d'une ou plusieurs classes. On peut ajouter des objets à notre modèle pour représenter des instances concrètes et prototypiques. Une instance concrète représente une personne ou une chose réelle. Par exemple, une instance concrète d'une classe Customer représente un client réel. Une instance prototypique d'une classe Customer contient des données représentant un client type.

## Packages

Les packages regroupent des éléments de modèle associés de tout type, y compris d'autres packages.

## Signaux

Dans les modèles UML, les signaux sont des éléments de modèle indépendants des

discriminants qui les gèrent. Les signaux spécifient des communications asynchrones oneway entre des objets actifs.

### **Enumérations**

Dans les modèles UML, les énumérations sont des éléments de modèle dans les diagrammes de classes qui représentent des types de données définis par l'utilisateur. Les énumérations contiennent des ensembles d'identificateurs nommés qui représentent les valeurs de l'énumération. Ces valeurs sont appelées littéraux d'énumération.

### **Type de données :**

Dans les diagrammes UML, les types de données sont des éléments de modèle qui définissent des valeurs de données. On utilise généralement les types de données pour représenter des types primitifs, tels que les types integer (entier) ou string (chaîne), et des énumérations, telles que des types de données définis par l'utilisateur.

### **Artefacts**

Dans les modèles UML, les artefacts sont des éléments de modèle qui représentent les entités physiques dans un système logiciel. Les artefacts représentent des unités physiques d'implémentation, telles que des fichiers exécutables, des bibliothèques, des composants de logiciel, des documents, et bases de documents.

### **Relations dans les diagrammes de classes**

En langage UML, une relation est une connexion entre des éléments de modèle. Une relation UML est un type d'élément de modèle qui ajoute une sémantique à un modèle en définissant la structure et le comportement entre les éléments de modèle.

### **Qualificateurs sur les extrémités d'association**

Dans UML, les qualificateurs sont des propriétés applicables aux associations binaires et des composants facultatifs des extrémités d'association. Un qualificateur contient une liste d'attributs d'association, doté chacun d'un nom et d'un type. Les attributs d'association modélisent les clés utilisées pour indexer un sous-ensemble d'instances de relation.



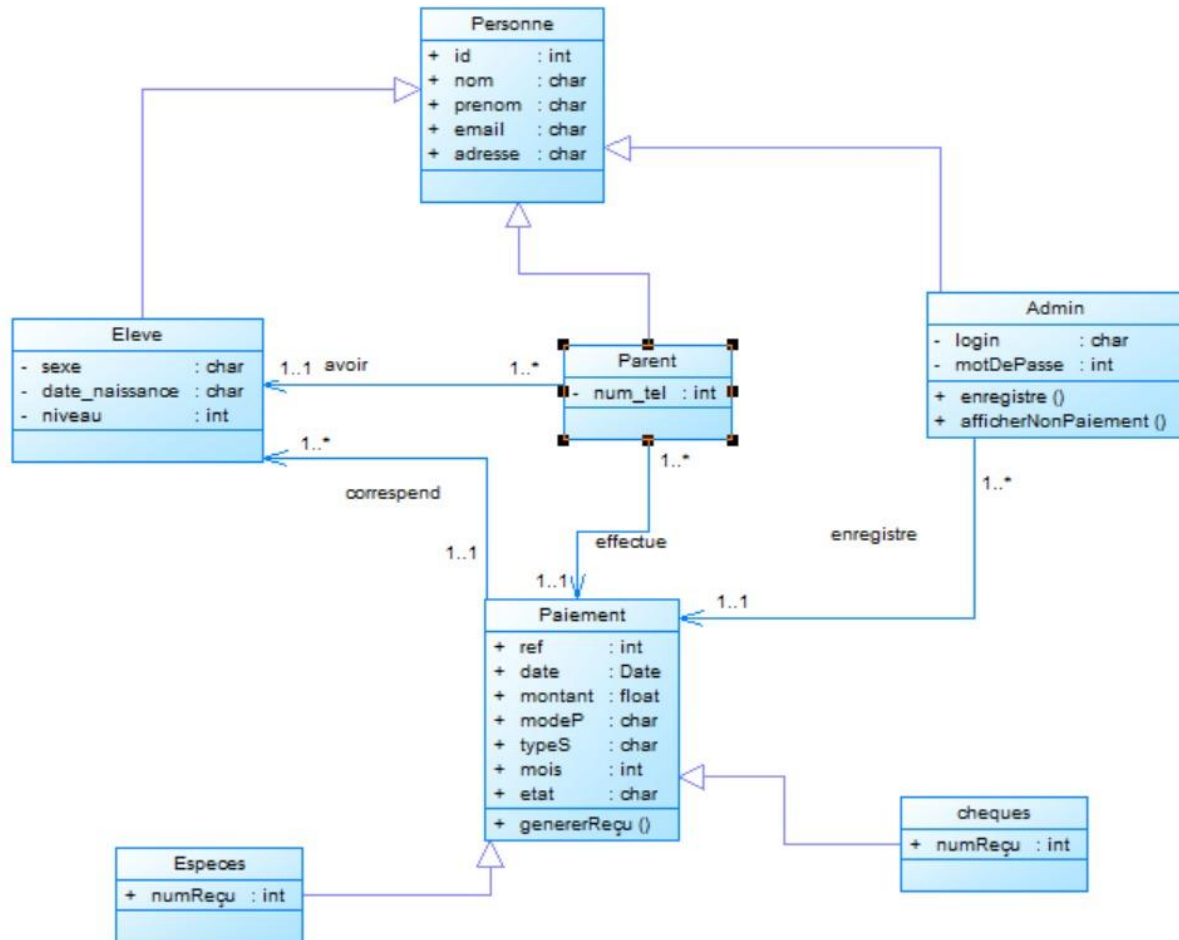


Figure 13:Diagramme de classe

## Diagramme de séquence :

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

Les diagrammes de séquence représentent la séquence de messages transmis entre des objets. Ils peuvent également représenter les structures de contrôle entre des objets. Par exemple, les lignes de vie dans un diagramme de séquence pour un scénario de banque peuvent représenter un client, un guichetier ou un responsable d'agence. Les communications entre le client, le guichetier et le responsable sont représentés par les messages entre ces derniers. Le diagramme de séquence représente les objets et les messages entre ces objets.

Les rubriques suivantes décrivent les éléments compris dans les diagrammes de séquence :

- **Lignes de vie dans les diagrammes UML**

Dans les diagrammes UML tels que les diagrammes de séquence ou de communication, les lignes de vie représentent les objets entrant dans une interaction. Dans un scénario d'une école, par exemple, les lignes de vie peuvent représenter des objets comme le système scolaire ou un client. Chaque instance d'une interaction est représentée par une ligne de vie.

- **Messages dans les diagrammes UML**

Un message est un élément de diagramme Unified Modeling Language (UML) qui définit un type particulier de communication entre les instances au cours d'une interaction. Un message fait circuler des informations d'une instance, représentée par une ligne de vie, à une autre instance au cours d'une interaction.

- **Fragments combinés dans les diagrammes de séquence**

Dans les diagrammes de séquence, les fragments combinés sont des regroupements logiques représentés par un rectangle et contenant les structures conditionnelles qui affectent le flux de messages. Un fragment combiné contient des opérandes d'interaction et est défini par un opérateur d'interaction.

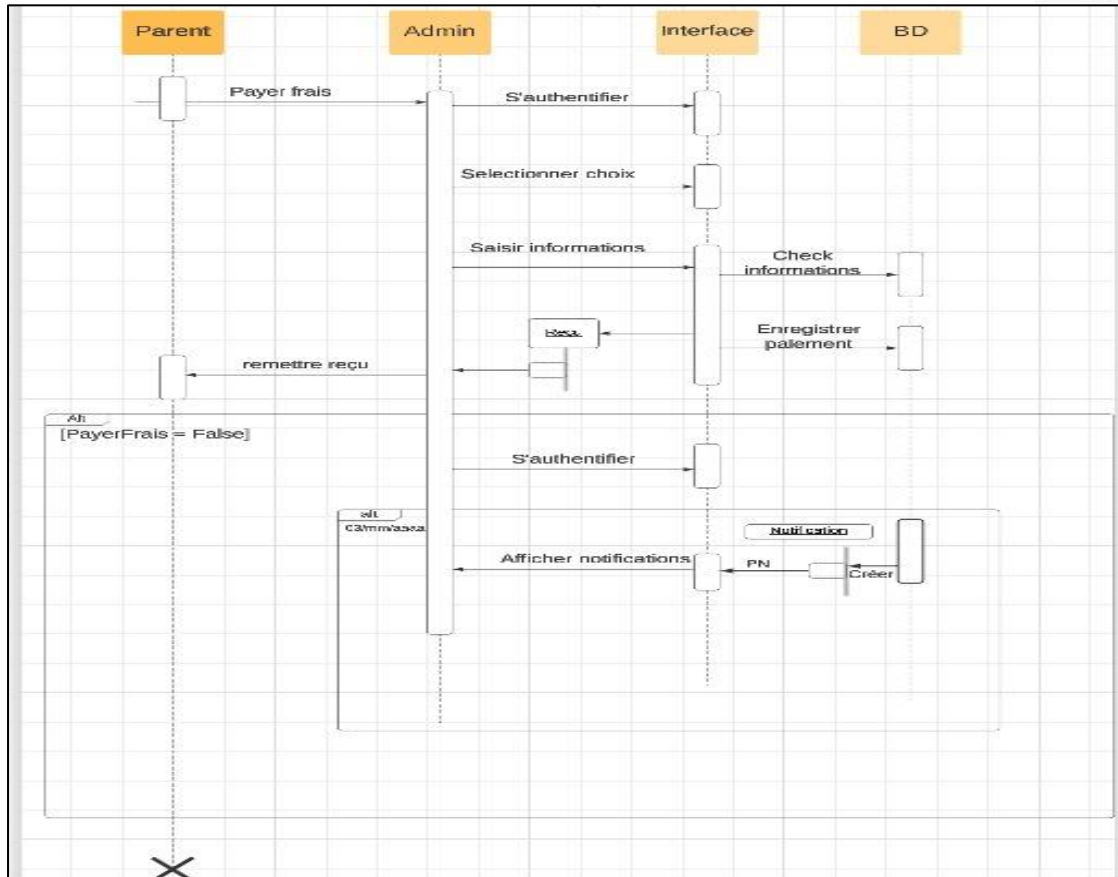


Figure 14:Diagramme de séquence

## Diagramme de déploiement :

Un diagramme de déploiement est un type de diagramme UML qui permet de représenter la distribution physique des composants logiciels sur les différents nœuds d'un système. Il permet de visualiser les différentes machines physiques, les logiciels installés sur ces machines et les connexions entre ces éléments.

Le diagramme de déploiement est utilisé pour planifier et décrire l'infrastructure matérielle nécessaire pour exécuter un système logiciel donné. Il permet également de visualiser les relations entre les différents éléments du système et de comprendre les exigences en matière de communication et de sécurité.

Le diagramme de déploiement est un outil utile pour comprendre la façon dont les différents composants d'un système sont déployés et interagissent les uns avec les autres. Il peut être utilisé pour planifier et concevoir des systèmes complexes, ainsi que pour décrire les exigences en matière d'infrastructure pour un système donné.

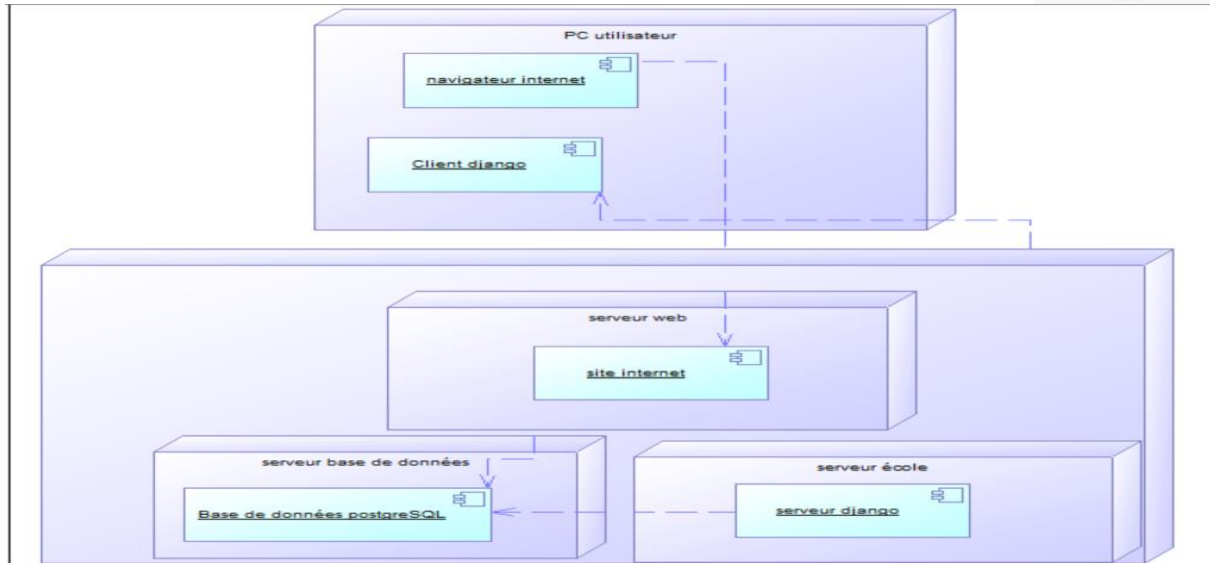


Figure 15:Diagramme de déploiement

## Conclusion

Nous avons présenté dans ce chapitre la conception des différents axes de notre projet. Cette conception a été faite sur la base de diagrammes comportementaux. Arrivés à ce point du projet, toutes les conditions sont rassemblées afin d'entamer la phase de réalisation qui rappelle souvent de constater la pertinence de la conception.

# Chapitre 4 : Outils et technologies utilisés

## Introduction

Dans ce chapitre nous présentons les différentes technologies ainsi que les outils de contrôle utilisés lors de la phase de développement.



### HTML5

**HTML5** est un langage conçu pour organiser le contenu Web. Il est destiné à faciliter la conception et le développement Web en créant un langage de balisage d'interface utilisateur standardisé et intuitif. **HTML5** fournit les moyens de disséquer et de compartimenter vos pages, et il vous permet de créer des composants discrets qui ne sont pas seulement conçus pour organiser votre site de manière logique, mais sont également créés pour donner à votre site des capacités de syndication. **HTML5** pourrait être appelé "l'approche de cartographie de l'information pour la conception de sites Web" car il intègre l'essence de la cartographie de l'information, de la division et de l'étiquetage des informations pour la rendre facile à utiliser et à comprendre. C'est le fondement de l'utilité sémantique et esthétique spectaculaire de **HTML5**. **HTML5** donne aux concepteurs et aux développeurs de tous niveaux la possibilité de tout publier dans le monde entier, du simple contenu textuel au riche,

**HTML5** fournit des outils efficaces de gestion des données, de dessin, de vidéo et d'audio. Il facilite le développement d'applications multi-navigateurs pour le Web ainsi que pour les appareils portables. **HTML5** est l'une des technologies à l'origine des progrès des services d'informatique en nuage mobile, car elle permet une plus grande flexibilité, permettant le développement de sites Web passionnants et interactifs. Il introduit également de nouvelles balises et améliorations, notamment une structure élégante, des contrôles de formulaire, des API, du multimédia, la prise en charge de bases de données et une vitesse de traitement nettement plus rapide.

Les nouvelles balises de **HTML5** sont très évocatrices, encapsulant leur rôle et leur utilisation. Les versions antérieures de **HTML** utilisaient des balises plutôt indéscriptibles. Cependant, **HTML5** a des étiquettes très descriptives et intuitives. Il fournit des étiquettes de contenu riches qui identifient immédiatement le contenu. Par exemple, la <div> balise overworked a été complétée par les balises <section> et <article>. L'ajout des balises <video>, <audio>, <canvas> et <figure> fournit également une description plus précise du type de contenu spécifique.



## CSS 3

CSS (Cascading Style Sheets) est un langage de programmation utilisé pour styliser et mettre en page des pages web. Il fonctionne en conjonction avec HTML (Hypertext Markup Language) et JavaScript pour créer une page web complète. CSS offre un moyen de séparer la présentation d'une page web (c'est-à-dire son apparence visuelle) de son contenu, ce qui facilite la maintenance et la mise à jour d'un site web.

Voici quelques concepts clés de CSS :

- **Sélecteurs** : Les sélecteurs CSS sont utilisés pour cibler des éléments spécifiques dans une page HTML. Par exemple, vous pouvez utiliser un sélecteur pour cibler tous les éléments de paragraphe sur une page et définir leur taille de police à 16 pixels.
- **Propriétés** : Les propriétés CSS sont utilisées pour définir les styles à appliquer aux éléments ciblés. Par exemple, vous pouvez utiliser la propriété "font-size" pour définir la taille de police d'un élément.
- **Valeurs** : Les valeurs CSS sont utilisées pour spécifier la manière dont une propriété doit être appliquée à un élément. Par exemple, vous pouvez utiliser la valeur "16px" pour définir la taille de police à 16 pixels.
- **Héritage** : Les propriétés CSS peuvent être héritées d'un élément parent à un élément enfant. Cela signifie que si vous définissez une propriété de style sur un élément parent, tous les éléments enfants hériteront de cette propriété à moins qu'une propriété différente ne soit définie pour eux.
- **Cascade** : La cascade est un mécanisme de résolution de conflits lorsque plusieurs règles CSS sont appliquées à un même élément. La cascade détermine quelle règle a la priorité en fonction de la spécificité du sélecteur et de l'ordre de déclaration des règles.

En plus des concepts clés, CSS propose également des fonctionnalités avancées telles que les mises en page flexbox et les grilles CSS, qui permettent de créer des mises en page complexes avec facilité. CSS est également compatible avec les médias, ce qui permet de définir des styles spécifiques pour différents types d'appareils et de tailles d'écran.



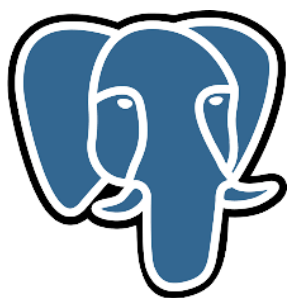
## JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript pour l'interpréter.

JavaScript est aussi employé pour les serveurs Web avec l'utilisation (par exemple) de Node.js ou de Deno.

JavaScript a été créé en 1995 par Brendan Eich et intégré au navigateur web Netscape Navigator 2.0. L'implémentation concurrente de JavaScript par Microsoft dans Internet Explorer jusqu'à sa version 9 se nommait JScript, tandis que celle d'Adobe Systems se nommait ActionScript. JavaScript a été standardisé sous le nom d'ECMAScript en juin 1997 par Ecma International dans le standard ECMA-262. La version en vigueur de ce standard depuis juin 2022 est la 13<sup>e</sup> édition.

C'est un langage orienté objet à prototype : les bases du langage et ses principales interfaces sont fournies par des objets. Cependant, à la différence d'un langage orienté objets, les objets de base ne sont pas des instances de classes. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel.



## PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle objet (ORDBMS) basé sur PostgreSQL. C'est un outil libre disponible selon les termes d'une licence de type BSD. Il a été lancé en 1986 sous la direction de Michael Stonebraker et parrainé par la (DARPA) et la (NSF), entre autres PostgreSQL n'est pas seulement une base de données compatible SQL, mais possède également des fonctionnalités comme :

- Possibilité de requêtes complexes, concept de transactions complètes ainsi que des vues actualisables.
- Triggers lancés automatiquement lors de la saisie et du contrôle, de la confirmation, de la modification, de la suppression ou de l'utilisation facultative de données de référence.
- Contrôle de simultanéité multi version (MVCC) pour une exécution efficace des accès simultanés aux bases de données.



PostgreSQL est basé sur le modèle client-serveur classique : Le composant serveur central « postmaster » est capable de traiter les requêtes, gérer tous les fichiers de base de données ainsi que toutes les connexions établies pour la communication avec le serveur de base de données. Donc passer alternativement à diverses applications avec des interfaces utilisateur graphiques, telles que :

- PgAdmin est un outil d'administration graphique pour PostgreSQL distribué selon les termes de la licence PostgreSQL.
- PhpPgAdmin est une interface web d'administration pour PostgreSQL. L'outil d'administration est écrit en PHP et supporte les différentes branches de PostgreSQL depuis sa version 7. L'interface s'appuie sur des scripts PHP et sur la base de données PostgreSQL pour favoriser les diverses tâches d'administration via le World Wide Web.

Dans le cas des sites Web interactifs, le serveur Web assume habituellement le rôle du client.

Ce SGBDRO est reconnu pour son comportement stable, proche d'Oracle, mais aussi pour ses possibilités de programmation étendues, directement dans le moteur de la base de données, via PL/pgSQL. Le traitement interne des données peut aussi être couplé à d'autres modules externes compilés dans d'autres langages.

PostgreSQL a été le pionnier de nombreux concepts qui sont devenus disponibles dans certains systèmes de bases de données commerciales beaucoup plus tard.



## Python

Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plateformes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la

productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation.

## Django



Django est un Framework Web Python de haut niveau qui permet le développement rapide de sites Web sécurisés et maintenables. Construit par des développeurs expérimentés, Django prend en charge une grande partie des tracas du développement Web, vous pouvez donc vous concentrer sur l'écriture de votre application sans avoir à réinventer la roue. Il est gratuit et open source, possède une communauté florissante et active, une excellente documentation et de nombreuses options de support gratuit et payant.

Django a été initialement développé entre 2003 et 2005 par une équipe Web chargée de créer et de maintenir les sites Web des journaux. Après avoir créé un certain nombre de sites, l'équipe a commencé à factoriser et à réutiliser de nombreux modèles de code et de conception courants. Ce code commun a évolué vers un Framework de développement Web générique, qui a été open-source sous le nom de projet "Django" en juillet 2005.

Django n'a cessé de croître et de s'améliorer, depuis sa première version importante (1.0) en septembre 2008 jusqu'à la version 4.0 (2022) récemment publiée. Chaque version a ajouté de nouvelles fonctionnalités et des corrections de bogues, allant de la prise en charge de nouveaux types de bases de données, de moteurs de modèles et de mise en cache, à l'ajout de fonctions et de classes d'affichage "génériques" (qui réduisent la quantité de code que les développeurs doivent écrire pour un certain nombre de tâches de programmation).

## Conclusion

Ce chapitre définit les logiciels utilisés afin de réaliser ce projet en décrivant leurs architectures, origines et développement. Ces outils vont nous aider à réaliser une application de paiement scolaire.

# Chapitre 5 : Réalisation

## Introduction :

Ce chapitre vient de concrétiser les phases d'analyse et de conception. Il présente aussi la démarche adoptée pour la réalisation du projet et quelques exemples des outils utilisés.

## Interfaces de l'application :

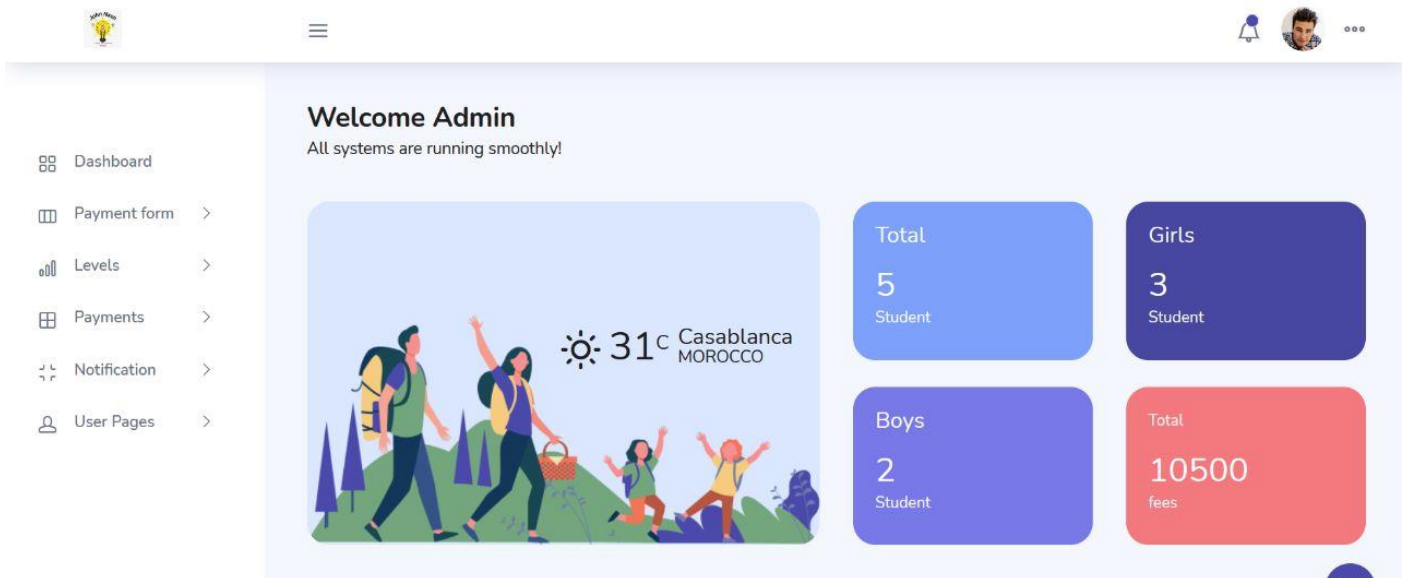


Figure 16:Dashboard

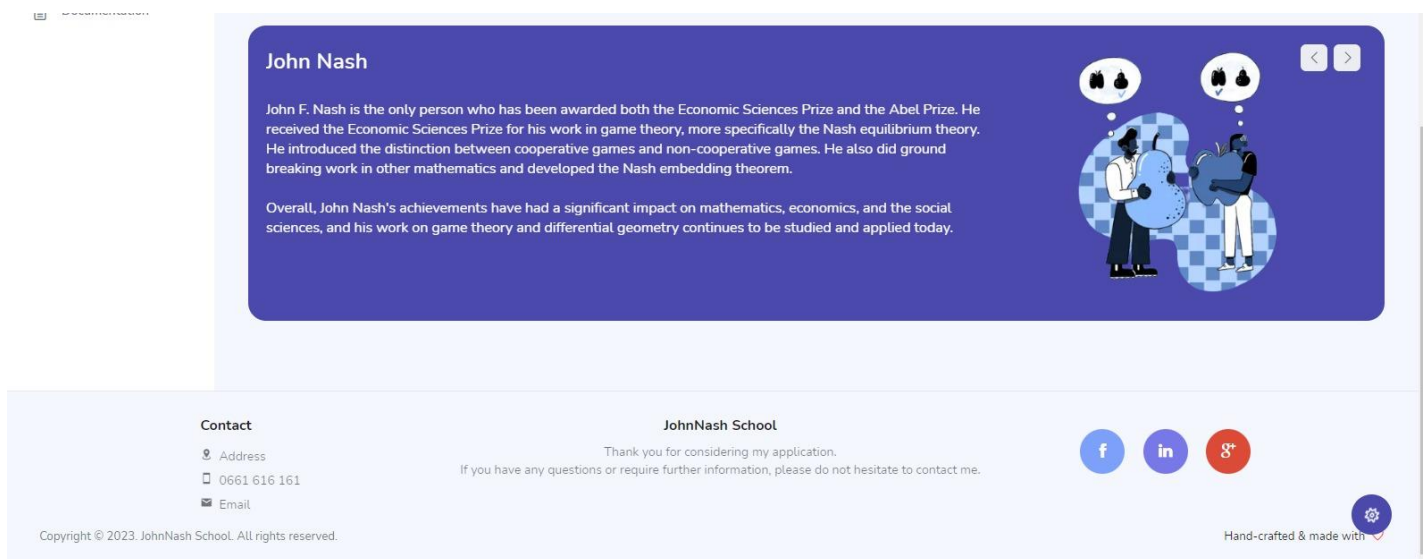


Figure 17:Dashboard (suite)

The sign-up interface features a sidebar menu on the left with options: Payment form, Levels, Payments, Notification, and User Pages (which is expanded to show Login and Register). The main area is titled 'Sign up' and contains a registration form with the following fields: 'Your Name', 'Your Email', 'Password', and 'Repeat your password'. A 'REGISTER' button is located at the bottom of the form. To the right of the form is a colorful illustration of a person sitting on a large number '2', surrounded by various geometric shapes and symbols. The top right corner includes a notification bell and a user profile icon.

Figure 18: Sign up Interface

The Django administration page has a header with 'Django administration' on the left and 'WELCOME, KK. VIEW SITE / CHANGE PASSWORD / LOG OUT' on the right. The main content is divided into two sections: 'Site administration' and 'Recent actions'.

**Site administration**

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change

GESTION	
Admines	+ Add Change
Cheques	+ Add Change
Elevs	+ Add Change
Especes	+ Add Change
Paiments	+ Add Change
Parents	+ Add Change

**Recent actions**

**My actions**

- payment object (20230408221619408707) Payment
- payment object (20230408222211264872) Payment
- payment object (20230408223702948192) Payment
- payment object (20230408225750887708) Payment
- payment object (20230409132906295393) Payment
- BOUKATE Loubna Eleve
- BOUKATE Loubna Eleve

Figure 19: Page administrateur

**1st Primary Year**

Rechercher Search

Id	Last name	First name	E-mail	Address	Sexe	Date of birth	Level
3	BAKIR	jamal	jamal@gmail.com	casa	Male	March 7, 2017	1st primary y
4	ELQADI	Salma	salma@gmail.com	bouskoura	Female	Feb. 4, 2017	1st primary y
2	AZIZI	Yasse	yasser@gmail.com	casa	Male	April 5, 2016	1st primary y
5	BAKIR	Kawtar	k@gmail.com	casa	Female	March 7, 2017	1st primary y
6	BOUKATE	Loubna	loubna@gmail.com	casa	Female	Feb. 4, 2017	1st primary y

Figure 20:Table élève

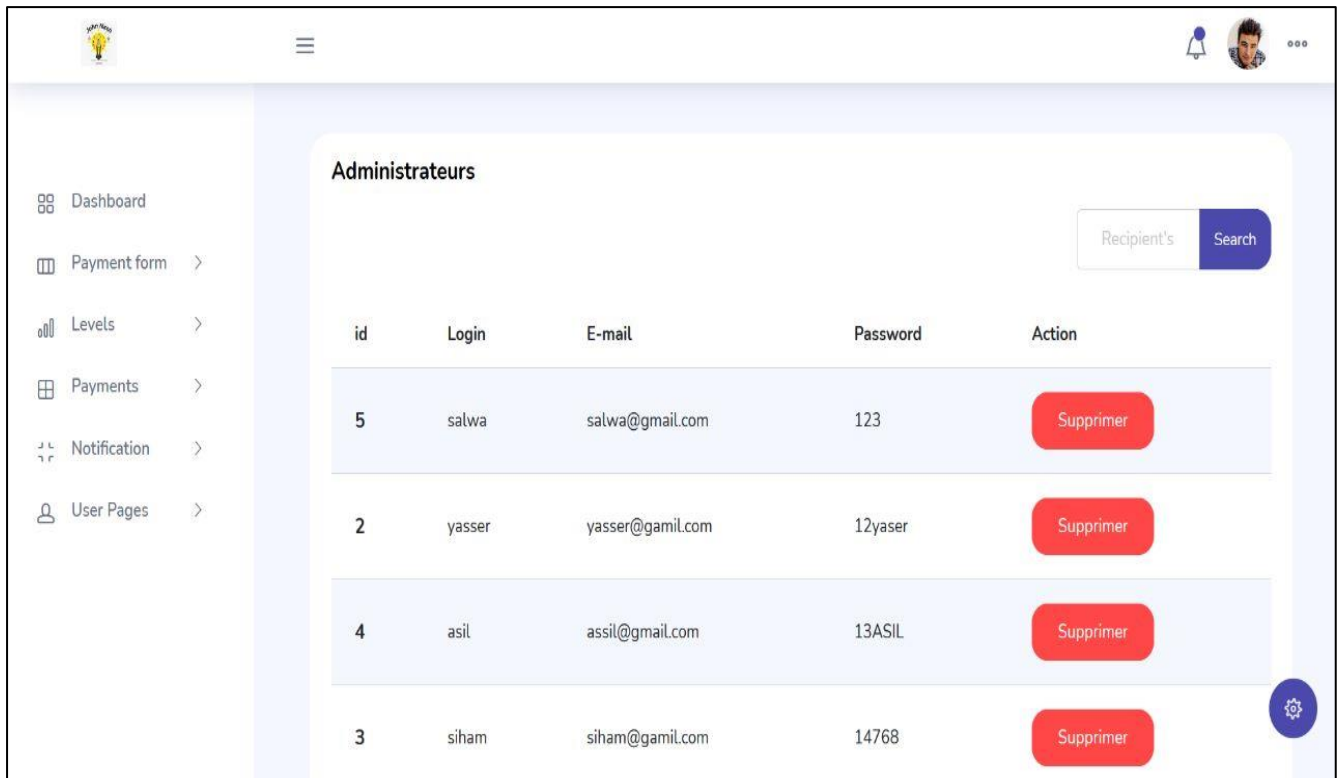
te	Fees	Mode Payment	Eleve	Service type	Month	Action
ril 9, 2023	1500	Espèces	BAKIR jamal	['scolarite']	April	Supprimer
ril 3, 2023	2000	Chèque	ELQADI Salma	['scolarite','cantine']	April	Supprimer
arch 8, 2023	1500	Chèque	AZIZI Yasse	['scolarite','cantine']	March	Supprimer
ril 8, 2023	1500	Chèque	BOUKATE Loubna	['scolarite','cantine']	April	Supprimer
ril 8, 2023	1500	Espèces	BAKIR Kawtar	['scolarite','cantine']	April	Supprimer

Figure 21:Table paiement

## Payment invoice

Student: BAKIR Kawtar

46



id	Login	E-mail	Password	Action
5	salwa	salwa@gmail.com	123	Supprimer
2	yasser	yasser@gamil.com	12yaser	Supprimer
4	asil	assil@gmail.com	13ASIL	Supprimer
3	siham	siham@gamil.com	14768	Supprimer

Figure 24: Table des administrateurs

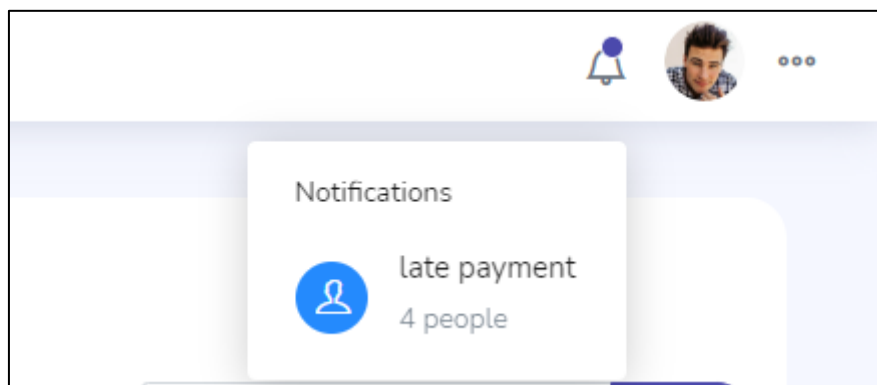


Figure 25: Notification des impayés



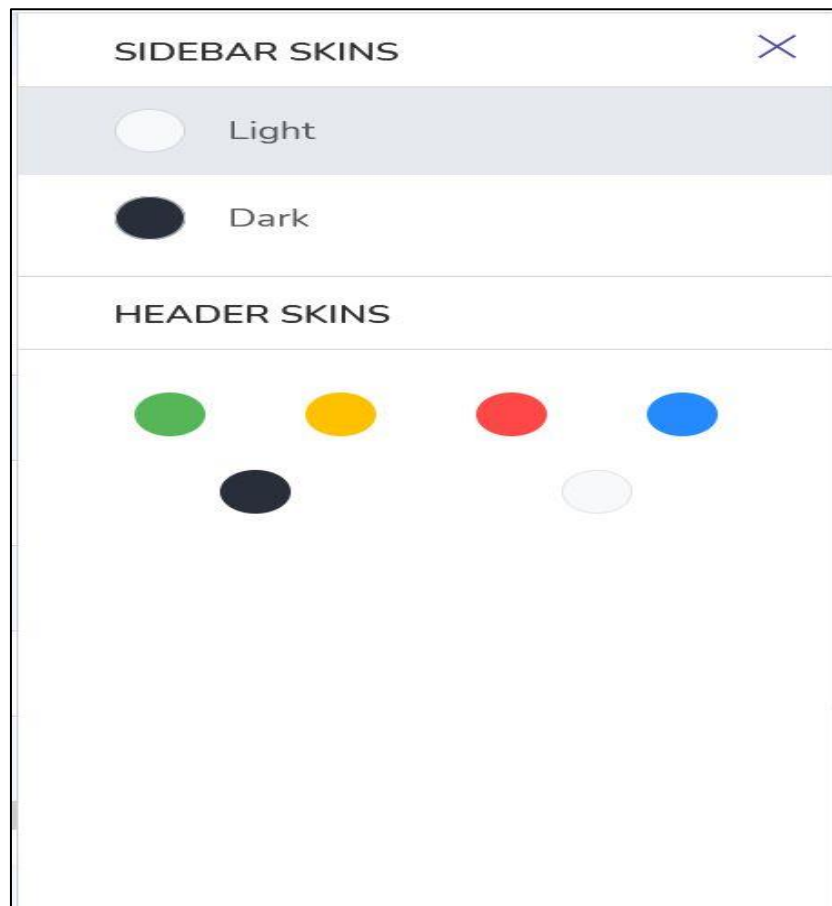


Figure 26: Mode de SideBar

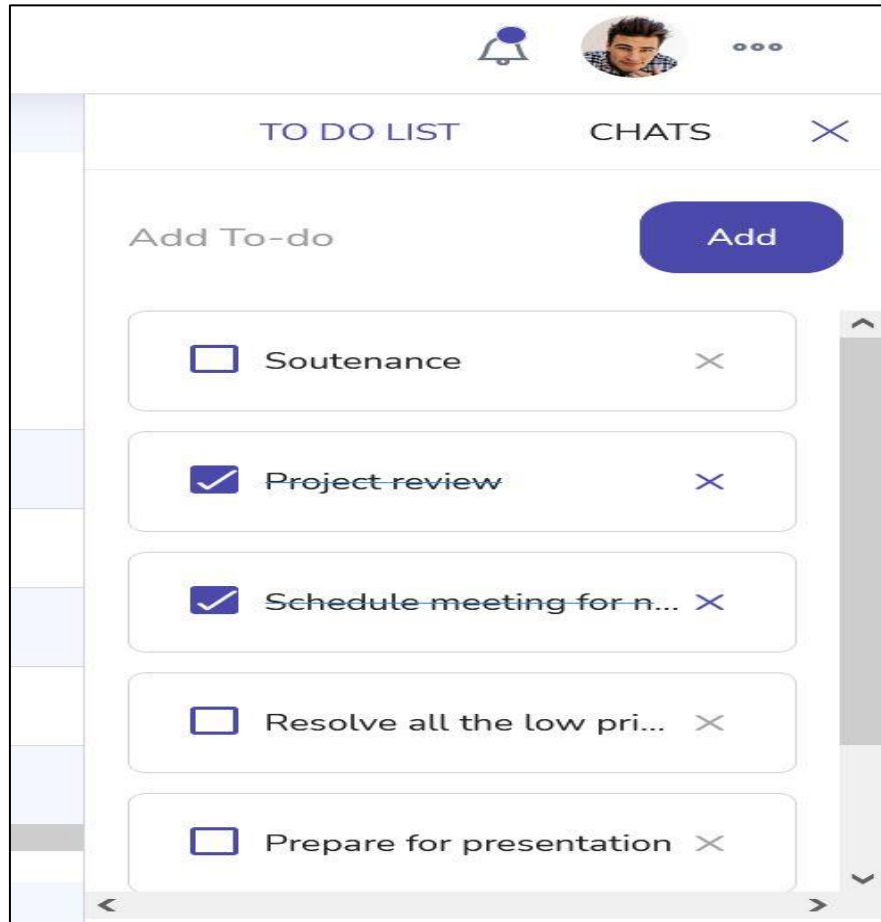


Figure 27: To Do List

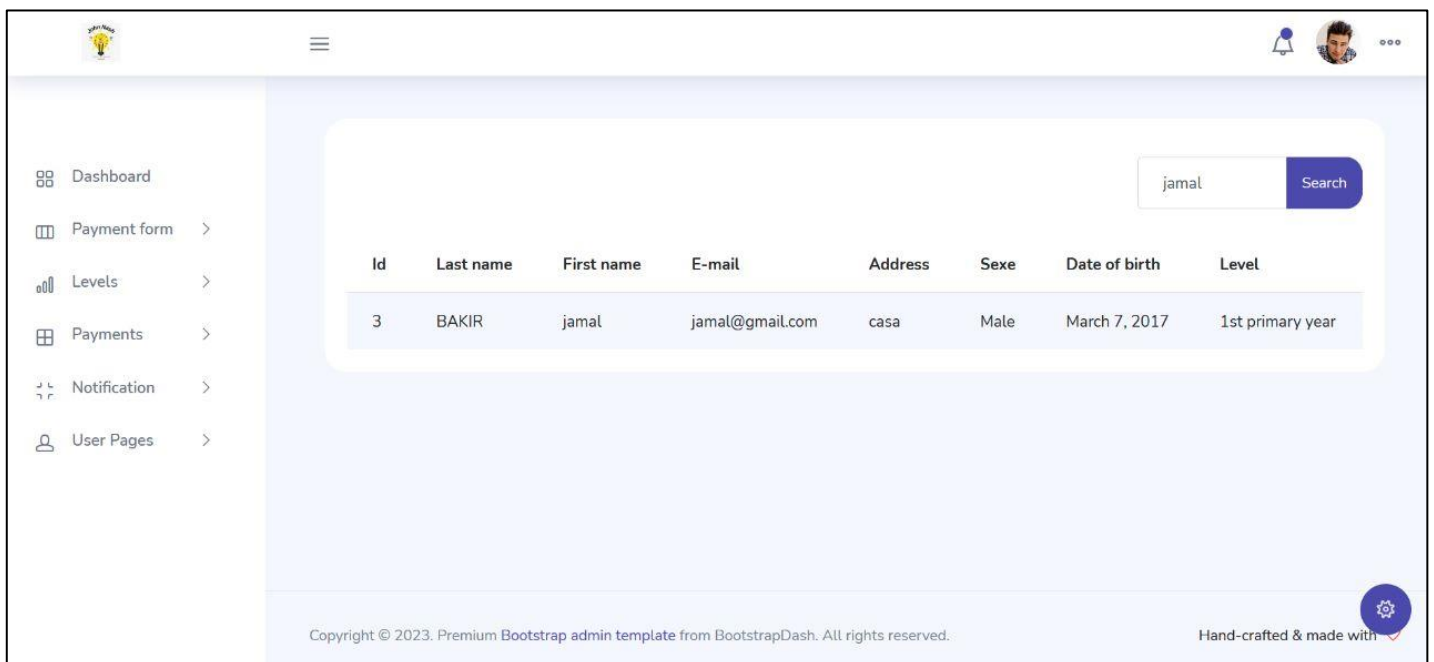


Figure 28: Recherche élève

## Conclusion générale :

En conclusion de ce rapport, l'application de paiement scolaire répond efficacement aux besoins des établissements scolaires.

L'application offre une alternative pratique et sécurisée pour le paiement des frais scolaires, permettant aux administrateurs de gérer les factures scolaires de manière efficace, tout en offrant une traçabilité et une transparence des paiements.

De plus, l'application permet une gestion plus efficace des transactions de paiement, réduisant ainsi les erreurs de comptabilité et permettant une plus grande efficacité dans la gestion financière des écoles.

Enfin, l'application de paiement scolaire que nous avons développée est facile à utiliser et intuitive.

En somme, nous sommes convaincus que notre application de paiement scolaire représente une solution complète, efficace et sécurisée pour répondre aux besoins de gestion financière des écoles.

## Bibliographies:

<https://www.geeksforgeeks.org/python-program-to-get-month-name-from-month-number/>

<https://www.w3schools.com/django/>

<https://stackoverflow.com/questions/70472865/django-3-2-8-0-static-files-copied-to-static>

<https://docs.djangoproject.com/fr/4.1/topics/db/models/#~:text=Dans%20Django%2C%20l'h%C3%A9ritage%20de,une%20sous%2Dclasse%20de%20django>

<https://lucid.app/>

<https://cloud.smartdraw.com/editor.aspx?credID=-47257684&depoId=44298115&flags=144#>