|  |  |
| --- | --- |
|  |  |



**UNIVERSITE DE FIANARANTSOA**

**ECOLE NATIONALE D’INFORMATIQUE**

# MEMOIRE DE FIN D’ETUDES POUR L’OBTENTION DU DIPLOME DE

**LICENCE PROFESSIONNELLE**

Mention : informatique

Parcours : Informatique Générale

**Intitulé**

CONCEPTION ET REALISATION D’UNE APLLICATION

POUR LA GESTION DES INCRIPTIONS ET ECOLAGES

Présenté le 20 janvier 2024

Par Monsieur NAKAZANDRY Aaron Bertin

**Membres du jury :**

Président : Monsieur RAFAMANTANANTSOA Fontaine, Professeur

Examinateur : Monsieur RAZAFINDRAMONJA Clément Auber, Assistant d’Enseignement

Supérieur et de Recherche

Rapporteurs :

* Monsieur GILANTE Gesazafy, Assistant d’Enseignement Supérieur et de Recherche
* Monsieur RATEFIARIVONY Rado Heriniaina, Ingénieur en Informatique

# CURRICULUM VITAE



NAKAZANDRY Aaron Bertin

**Age :** 21ans, Célibataire

**Adresse :** Bateravola Fianarantsoa

**Contact :** 0349974748

**E- mail :** [nakazandryaaron@gmail.com](mailto:nakazandryaaron@gmail.com)

**FORMATION ET DIPLÔME**

**2023-2024 :** Troisième année de formation en Licence professionnelle à l’Ecole Nationale d’Informatique de l’Université de Fianarantsoa, Parcours : IG.

**2022-2023 :** Deuxième année de Licence professionnelle à l’Ecole Nationale d’Informatique de l’Université de Fianarantsoa, Parcours : IG.

**2021-2022 :** Première annéede formation en Licence professionnelle à l’Ecole Nationale d’Informatique de l’Université de Fianarantsoa, Parcours : IG.

**2020-2021 :** Obtention du diplôme baccalauréat série D au Lycée Fo Masin’i Jesoa Talatamaty.

**STAGES ET EXPERIENCES PROFESSIONNELLES**

**2023-2024 :** Stage aux seins du lycée LOTERANA RAKOTO ANDRIANARIJAONA IVORY ATSIMO portant le thème de conception et réalisation d’une application de gestion pour l’inscriptions et écolages

**2022-2023 :** Stage aux seins de l’association Fonds d'Intervention pour le Développement (FID) Manakara, pour thème Conception et réalisation d’une application de gestion d’un parc informatique.

* Configuration des protocoles de routage RIP et OSPF pour assurer la connectivité et l’acheminement des données avec le logiciel de simulation GNS3.

**2021-2022 :**

* Miseen place d’un système pour gérer les réservations au sein de coopératives.
* Mise en œuvre et paramétrage d’un serveur de basculement de fichiers.

**COMPETENCES EN INFORMATIQUE**

**Système et Réseaux :**

* Systèmes d’exploitation : WINDOWS et LINUX.
* GNS3 et Virtualisation.
* Analyse des malwares

**Technologies Web :** HTML ET CSS, Javascript, PHP.

**Langages de programmation** : JAVA, Langage C, C++, C#, R, Assembleur

**Frameworks :** VueJS, Laravel, ExpressJS

**SGBD :** Mysql, PostgreSQL.

**Technologies de virtualisation :** Virtual Box, VMware

**Méthodes de conception:**MERISE, 2TUP

**Outils :** Git et GitHub, Win design.

**Outils Bureautiques :** Microsoft Offices.

**CONNAISSANCES LINGUISTIQUES**

|  |  |  |
| --- | --- | --- |
| Expression oral | Expression écrite | Compréhension |
| Français | Moyen | Très bien | Très bien |
| Anglais | Moyen | Moyen | Moyen |

**Divertissement**

* Foot Ball
* Basket Ball

**SOMMAIRE**

[**CURRICULUM VITAE I**](#_Toc187873717)

[**Remerciements VI**](#_Toc187873718)

[**Liste des figures VII**](#_Toc187873719)

[**Liste des tableaux IX**](#_Toc187873720)

[**Liste des abréviations X**](#_Toc187873721)

[**PARTIE I : PRESENTATIONS 2**](#_Toc187873722)

[**Chapitre 1. Présentation de l’École Nationale d’Informatique 3**](#_Toc187873723)

**1.1** [**Domaine de spécialisation 6**](#_Toc187873724)

**1.2** [**Relation de l’ENI avec les organismes externes 8**](#_Toc187873725)

**1.3** [**Débouchés professionnels et diplômés 9**](#_Toc187873726)

**1.4** [**Ressources humaines 10**](#_Toc187873727)

[**Chapitre 2 : Présentation de Lycée Loterana Rakoto Andrianarijaona 11**](#_Toc187873728)

[**2.1 Information d’ordre générale 11**](#_Toc187873729)

[**2.2 Historiques en général 11**](#_Toc187873730)

[**2.3 Ces dirigeants se sont relayes pour diriger l’établissement jusqu’à aujourd’hui : 12**](#_Toc187873731)

[**2.3 Organigramme 12**](#_Toc187873732)

[**2.5 Ressources humaines 13**](#_Toc187873733)

[**Chapitre 3. Description du projet 14**](#_Toc187873734)

[**3.1 Formulation 14**](#_Toc187873735)

[**3.2 Objectif et besoins de l’utilisateur 14**](#_Toc187873736)

[**3.3 Moyens nécessaires à la réalisation du projet 15**](#_Toc187873737)

[**3.4 Résultats attendus 16**](#_Toc187873738)

[**3.5 Chronogramme du projet 16**](#_Toc187873739)

[**Chapitre 4. Analyse préalable 17**](#_Toc187873740)

[**4.1 Analyse de l’existant 17**](#_Toc187873741)

[**4.2 Critique de l’existant 18**](#_Toc187873742)

[**a. Choix de méthode de conception et de modélisation 19**](#_Toc187873743)

[**b. Choix du système de gestion de base de données 21**](#_Toc187873744)

[**c. Outil de modélisation 22**](#_Toc187873745)

[**d.Langage de programmation 23**](#_Toc187873746)

[**Chapitre 5. Analyse conceptuelle 27**](#_Toc187873747)

[**5.1 Dictionnaire des données 27**](#_Toc187873748)

[**5.3 Règles de gestion 29**](#_Toc187873749)

[**5.4. Représentation et spécification des besoins 30**](#_Toc187873750)

[**5.5 Spécification des besoins techniques 47**](#_Toc187873751)

[**5.6 Modélisation du domaine 47**](#_Toc187873752)

[**Chapitre 6. Conception détaillé 49**](#_Toc187873753)

[**6.1. Architecture système 49**](#_Toc187873754)

[**6.2 Diagramme de séquence de conception pour chaque cas d’utilisation 50**](#_Toc187873755)

[**6.3 Diagramme de classe de conception pour chaque cas d’utilisation 57**](#_Toc187873756)

[**6.2 Diagramme de classe globale 60**](#_Toc187873757)

[**6.3 Diagramme de paquetage 61**](#_Toc187873758)

[**6.4 Diagramme de déploiement 62**](#_Toc187873759)

[**Partie III : Réalisation 63**](#_Toc187873760)

[**Chapitre 7. Mise en place de l’environnement de développement 64**](#_Toc187873761)

[**7. 1 Installation et configuration des outils 64**](#_Toc187873762)

[**7.1.3 Installation de Visual Paradigm 65**](#_Toc187873763)

[**7.1.1. Installation de Node.js 66**](#_Toc187873764)

[**7.1.4 Installation de PostgreSQL xi**](#_Toc187873765)

[**7.2 Architecture de l’applications xii**](#_Toc187873766)

[**Chapitre 8. Développement de l’application xiii**](#_Toc187873767)

[**8.1. Création de la base de données xiii**](#_Toc187873768)

[**8.2. Codage de l’application xiii**](#_Toc187873769)

[**8.3. Présentation de l’application xvii**](#_Toc187873770)

[**Conclusion xvii**](#_Toc187873771)

[**Références bibliographiques xxi**](#_Toc187873772)

[**Références webographies xxii**](#_Toc187873773)

[**Glossaire xii**](#_Toc187873774)

[**Table des matières xiii**](#_Toc187873775)

[**Résumé………………... xvii**](#_Toc187873776)

[**Abstract xvii**](#_Toc187873777)

## Remerciements

Un succès c’est le résultat d’un dur labeur, des efforts, des persévérances, des années d’expérience, d’acquisition de savoir, d’aide et des soutiens extérieur. D’abord, j’offre mes plus humbles remerciements aux personnes qui m’a donné les connaissances nécessaires, de m’avoir aidé à y voir plus claire en ce qui concerne l’informatique. Dorénavant, j’ai pu connaitre en profondeur le développement d’application même si cependant, j’ai toujours envie d’en savoir un peu plus. Grâce aux connaissances que j’ai acquise, j’ai pu achever le projet que l’on m’a confié dans le temps. Mes remerciements s’adressant à vous :

* Monsieur HAJALALAINA Aimé Richard, Professeur, Président de l’université de Fianarantsoa pour le bon déroulement de l’année universitaire.
* Monsieur MAHATODY Thomas, Professeur, Directeur de l’Ecole Nationale d’Informatique, Assistant d’Enseignement supérieur et de la Recherche, d’avoir pris soin de l’Ecole et de ses étudiants en offrant une meilleure formation et gestion des administrations.
* Monsieur RASOLOFONDRADIMBY Andriantsoa, Proviseur de Lycée Loterana Rakoto Andrianarijaona, de nous avoir accueillis durant ces périodes de stage.
* Monsieur RABETAFIKA Louis Haja, Maître de conférences et Responsable de Mention à l’Ecole National d’Informatique à l’Université de Fianarantsoa, de nous avoir aidé en donnant des conseils attentifs durant la réalisation de notre stage.
* Monsieur GILANTE Gesazafy, Assistant d’Enseignement Supérieur et de la Recherche, Responsable de parcours « Informatique Générale » à l’Ecole Nationale d’Informatique, d’avoir planifié le bon fonctionnement de notre formation théorique et pratique et aussi l’encadreur pédagogique.

-Monsieur RATEFIARIVONY Rado Heriniaina, Ingénieur en Informatique, encadreur professionnel et responsable de conseil administrative et direction collégiale au sein de Lycée LRRA.

* Les membres de jury, pour avoir accepté de juger mon projet afin d’améliorer mes futurs projets.
* Les enseignants de l’Ecole Nationale d’Informatique pour toutes connaissances qu’ils ont transmises.
* Les responsables de l’administration et scolarité pour la gestion des fonctions de l’école.
* Et spécialement, à mes parents qui m’ont donné du courage et du soutient tout au long de mon périple.
* Les proches et amis, qui m’ont aussi soutenu tout le long de la réalisation de ce mémoire.

# Liste des figures

[Figure 1 Organigramme actuel de l’Ecole 5](#_Toc188002624)

[Figure 2 l’architecture des études correspondant au système LMD. 7](#_Toc188002625)

[Figure 3 Organigramme LLRA 13](#_Toc188002626)

[Figure 4 Chronogramme de la réalisation du projet 16](#_Toc188002627)

[Figure 5 Acteur 30](#_Toc188002628)

[Figure 6 Diagramme de cas d'utilisation global 31](#_Toc188002629)

[Figure 7 Diagramme de séquence système pour le cas d'utilisation S'authentifier 40](#_Toc188002630)

[Figure 8 Diagramme de séquence système pour le cas d’utilisation gérer classes 41](#_Toc188002631)

[Figure 9 Diagramme de séquence système pour le cas d'utilisation Gérer inscription 42](#_Toc188002632)

[Figure 10 Diagramme de séquence système pour le cas d’utilisation gérer l’écolage 43](#_Toc188002633)

[Figure 11. Diagramme de séquence système pour le cas d'utilisation Suivi des paiements 44](#_Toc188002634)

[Figure 12 Diagramme de séquence système pour le cas d'utilisation Gérer les professeurs 45](#_Toc188002635)

[Figure 13 Diagramme de séquence système pour le cas d’utilisation « Gérer les utilisateurs » 46](#_Toc188002636)

[Figure 14 Modélisation du domaine 48](#_Toc188002637)

[Figure 15 Architecture MVC 49](#_Toc188002638)

[Figure 16 Diagramme de séquence de conception du cas d'utilisation S'authentifier 50](#_Toc188002639)

[Figure 17 Diagramme de séquence du cas d’utilisation gérer classes 51](#_Toc188002640)

[Figure 18 Diagramme de séquence de conception du cas gérer inscription 52](#_Toc188002641)

[Figure 19 Diagramme de séquence de conception du cas gérer écolage 53](#_Toc188002642)

[Figure 20 Diagramme de séquence de conception du cas Suivi des paiements 54](#_Toc188002643)

[Figure 21 . Diagramme de séquence de conception du cas Gérer les professeurs 55](#_Toc188002644)

[Figure 22 Diagramme de séquence de conception du cas gérer les utilisateurs 56](#_Toc188002645)

[Figure 23 Diagramme de classe de conception du cas d'utilisation S'authentifier 57](#_Toc188002646)

[Figure 24 Diagramme de classe de conception pour le cas d’utilisation 58](#_Toc188002647)

[Figure 25 Diagramme de classe se de conception du cas d'utilisation gérer inscription 58](#_Toc188002648)

[Figure 26 Diagramme de classe se de conception du cas d'utilisation gérer écolage 59](#_Toc188002649)

[Figure 27 Diagramme de classe se de conception du cas d'utilisation gérer professeurs 59](#_Toc188002650)

[Figure 28 Diagramme de classe globale 60](#_Toc188002651)

[Figure 29 Diagramme de paquetage 61](#_Toc188002652)

[Figure 30 Diagramme de déploiement 62](#_Toc188002653)

[Figure 31 Contrat d'utilisation de Visual Studio Code 64](#_Toc188002654)

[Figure 32 Progression de l'installation de Visual Studio Code 65](#_Toc188002655)

[Figure 33 Fenêtre d'installation de visual paradigm 65](#_Toc188002656)

[Figure 34 . Etapes d'installation de visual paradigm 66](#_Toc188002657)

[Figure 35 Contrat de licence de Node.js 67](#_Toc188002658)

[Figure 36 Sélection du répertoire d'installation de Node.js 67](#_Toc188002659)

[Figure 37 Option "Configuration personnalisée" dans l'installateur de Node.js x](#_Toc188002660)

[Figure 38 Finalisation de l'installation de Node.js x](#_Toc188002661)

[Figure 39 Page d’installation PostgreSQL xi](#_Toc188002662)

[Figure 40 Configuration PostgreSQL xi](#_Toc188002663)

[Figure 41 Architecture 3-tiers xii](#_Toc188002664)

[Figure 42 Architecture globale de l'application xiii](#_Toc188002665)

[Figure 43 Dossier frontend Reactjs xiv](#_Toc188002666)

[Figure 44 Modal Utilisateurs xiv](#_Toc188002667)

[Figure 45 Frontend Utilisateurs xv](#_Toc188002668)

[Figure 46 Route pour la page Utilisateurs xv](#_Toc188002669)

[Figure 47 Routes reactJS xvi](#_Toc188002670)

[Figure 48 Page d’authentification xvii](#_Toc188002671)

[Figure 49 Page d’ajout d’une inscription xvii](#_Toc188002672)

[Figure 50 Page suivie des paiements xviii](#_Toc188002673)

[Figure 51 page utilisateur xviii](#_Toc188002674)

[Figure 52 Ajouter prix xviii](#_Toc188002675)

[Figure 53 Ajouter Ecolage xix](#_Toc188002676)

[Figure 54 Ajouter Classe xix](#_Toc188002677)

[Figure 55 Ajouter Professeur xix](#_Toc188002678)

# Liste des tableaux

[Tableau 1 Organisation du système de formation pédagogique de l’Ecole 6](#_Toc188000499)

[Tableau 2 Mention et parcours au sein de l’ENI 6](#_Toc188000500)

[Tableau 3 Liste des formations existantes à l’ENI 7](#_Toc188000501)

[Tableau 4 Débouchés éventuels des jeunes diplômés 9](#_Toc188000502)

[Tableau 5 Moyens matériels utilisés. 15](#_Toc188000503)

[Tableau 6 Logiciels utilisés. 15](#_Toc188000504)

[Tableau 7 Moyens humains pour la réalisation du projet 16](#_Toc188000505)

[Tableau 8 Moyens matériels utilisés 17](#_Toc188000506)

[Tableau 9 Moyens logiciels utilisés 18](#_Toc188000507)

[Tableau 10 Comparaison des deux solutions 19](#_Toc188000508)

[Tableau 11 Comparaison des méthodes de conception et de modélisation 20](#_Toc188000509)

[Tableau 12. Comparaison des SGBD 21](#_Toc188000510)

[Tableau 13. Comparaison Outil de modélisation 22](#_Toc188000511)

[Tableau 14 Comparaison entre PHP et JS 23](#_Toc188000512)

[Tableau 15 Comparaison entre VueJS et ReactJS 24](#_Toc188000513)

[Tableau 16 Comparaison ReactJS et NextJS 26](#_Toc188000514)

[Tableau 17 Dictionnaire des données. 27](#_Toc188000515)

[Tableau 18 Description textuelle « S’authentifier » 32](#_Toc188000516)

[Tableau 19 Description textuelle « Gérer classes » 33](#_Toc188000517)

[Tableau 20 Description textuelle « Gérer l’inscription » 34](#_Toc188000518)

[Tableau 21 Description textuelle « Gérer l’écolage » 35](#_Toc188000519)

[Tableau 22 Description textuelle « Suivi des paiements » 36](#_Toc188000520)

[Tableau 23 Description textuelle « Gérer professeurs » 37](#_Toc188000521)

[Tableau 24 Description textuelle « Gérer Utilisateur » 38](#_Toc188000522)

[Tableau 25 : Priorité des cas d’utilisations 39](#_Toc188000523)

# Liste des abréviations

**2TUP** : 2 Track Unified Process

**BTS** : Brevet de Technicien Supérieur

**CPU** : Centtral Proccessing Unit

**ENI** : Ecole Nationale d’Informatique

**HDD**: Hard Disk Drive

**HTML:** HyperText Markup Language

**HTTP**: HyperText Transfer Protocol

**IDE**: Integrated Development Environment

**LLRA :** Lycée Loterana Rakoto Andrianarijaona

**LMD** : Licence Master Doctorat

**MVC** : Modèle Vue Contrôleur

**PHP**: Personnal Home Page

**RAM**: Random Access Memory

**SGBD** : Système de Gestion de Base de Données

**SQL**: Structured Query Language

**UI**: User Interface

**UML**: Unified Modeling Language

**UP:** Unified Process

**URI:** Uniform Resource Identifier

**URL**: Uniform Resource Locator

**WAMP**: Windows Apache MySQL PHP

**XML** : eXtensible Markup Languag

**Introduction Générale**

Au cours des dernières décennies, l'informatisation s'est imposée comme un élément incontournable de la vie quotidienne. Dans une ère où la technologie régit nos interactions, les organismes doivent nécessairement intégrer l'automatisation des tâches pour garantir l'efficacité de leurs systèmes

Face à ces défis, se pose la problématique d'une gestion inefficace et non centralisée des inscriptions et des écolages. Cette situation entraîne des complications telles que des retards dans le traitement des inscriptions, un suivi inadéquat des paiements, et une absence de transparence dans la gestion des informations administratives. Ces lacunes freinent le bon fonctionnement des établissements et nuisent à leur productivité. Pour répondre à cette problématique, ce projet propose la conception et la mise en œuvre d’une application centralisée de gestion des inscriptions et des écolages. Ce système vise à rationaliser les processus administratifs liés aux inscriptions, au suivi des paiements, et à la gestion des données des professeurs et utilisateurs.

Pour élaborer ce projet on aura recouru à des Frameworks, un environnement de développement Web moderne et robuste garantissant ainsi une interaction fluide et une gestion efficace des données et un système de gestion de base de données.

Afin de mieux présenter ce mémoire, il sera divisé en trois parties. La première partie débutera par les présentations dont l’Ecole Nationale d’Informatique, Lycée Loterana Rakoto Andrianarijaona et la description du projet. La seconde partie concernera l’analyse et la conception du projet. Enfin dans la troisième partie, nous terminerons par la réalisation de l’application.

# PARTIE I : PRESENTATIONS

## Chapitre 1. Présentation de l’École Nationale d’Informatique

L’ENI se positionne sur l’échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques

* 1. **Information d’ordre générale**

L’École Nationale d’Informatique, en abrégé ENI, est un établissement d’enseignement supérieur rattaché académiquement et administrativement à l’Université de Fianarantsoa. Le siège de l’Ecole se trouve à Tanambao-Antaninarenina à Fianarantsoa. L’adresse pour la prise de contact avec l’Ecole est la suivante : Ecole Nationale d’Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 034 05 733 36 ou 032 15 204 28. Son adresse électronique est la suivante : [**eni@eni.mg**.](mailto:eni@eni.mg) Il dispose également d'un site web : [**www.eni.mg**](http://www.eni.mg/)

* 1. **Missions et historiques**

L’ENI se positionne sur l’échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd’hui comme la vitrine et la pépinière des élites informaticiennes du pays.

De façon formelle, l’ENI était créée par le décret N° 83- 185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d’Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L’ENI a pour conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

* + - En fournissant à des étudiants des connaissances de base en informatique ;
    - En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises ;
    - En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l’Information et de la Communication (TIC) ;

La filière de formation d’Analystes Programmeurs a été mise en place à l’Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d’ingénieurs a été ouverte à l’Ecole en 1986.

Une formation de troisième cycle a été ouverte à l’Ecole a été ouverte à l’Ecole depuis l’année 2003 – 2004 grâce à la coopération académique et scientifique entre l’Université de Fianarantsoa pour le compte de l’ENI et l’Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l’Informatique, et notamment pour préparer la relève des Enseignants- Chercheurs qui étaient en poste.

Pendant l’année 2007-2008, la formation en vue de l’obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l’ENI avec les deux parcours de formation

:

* + - Génie Logiciel et base de Données.
    - Administration des Système et réseaux.

La mise en place à l’Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

En vue de surmonter les difficultés de limitation de l’effectif des étudiants accueillis à l’Ecole, notamment à cause du manque d’infrastructures, un système de « Formation Hybride » a été mise en place à partir de l’année 2010. Il s’agit en effet d’un système de formation semi présentielle et à distance avec l’utilisation de la visioconférence pour la formation à distance. Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu’Université de Toliara. Cette formation est à l’origine du parcours Informatique Générale.

En 2023, une nouvelle mention Intelligence Artificielle (IA) a été ouvert au sein de l’École pour répondre les besoins des entreprises. La formation est destinée aux étudiants titulaires du diplôme de licence (Bac +3) en Mathématiques ou en Statistiques ou en Informatique, etc. La mention IA comporte deux parcours :

* + - Gouvernance et Ingénierie de Données (GID),
    - Objets connectés et Cybersécurité (OCC).

Le principe de l’enseignement pour le parcours GID offre aux l’étudiants des compétences scientifiques et techniques spécialisées en Science de données. Pour le parcours OCC, les étudiants octroient la double spécialité premièrement en internet des objets et deuxièmement en cybersécurité. La formation de master est axée sur l’ensemble d’applications de l’Intelligence Artificielle.

* 1. **Organigramme institutionnel**

Organigramme de l’Ecole est inspiré des dispositions du décret N° 83-185 du 24 Mai 1983. L’ENI est administrée par un-Conseil d’Ecole, et dirigée par un directeur nommé par un décret adopté en Conseil des Ministres. Le Collège des enseignants regroupant tous les enseignants-chercheurs permanents de l’Ecole est chargé de résoudre les problèmes liés à l’organisation pédagogique des enseignements. Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l’établissement, en tenant compte notamment de l’évolution du marché de travail et de l’adéquation des formations dispensées par rapport aux besoins des entreprises. La figure 1 représente l’organigramme actuel de l’ENI.

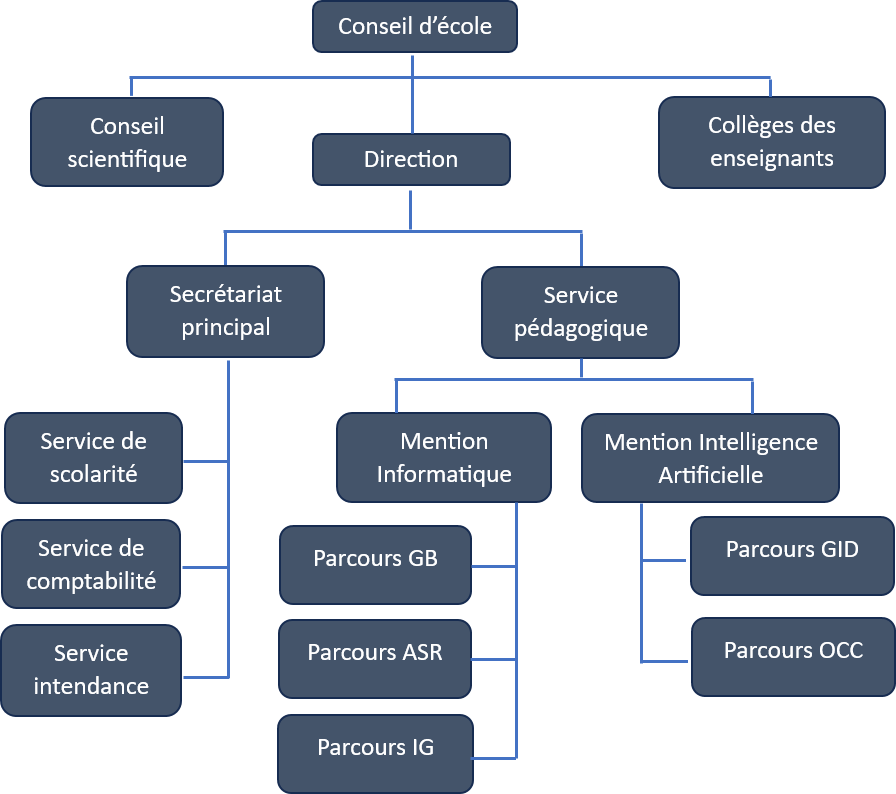


Figure 1 Organigramme actuel de l’Ecole

### Domaine de spécialisation

Les activités de formation et de recherche organisées à l’ENI portent sur les domaines suivants :

* + - Génie logiciel et Base de Données ;
    - Administration des Systèmes et Réseaux ;
    - Informatique Générale ;
    - Modélisation informatique et mathématique des Systèmes complexes ;
    - Intelligence artificielle.

Le tableau 1 décrit l’organisation du système de formation pédagogique de l’Ecole.

Tableau 1 Organisation du système de formation pédagogique de l’Ecole

|  |  |
| --- | --- |
| Formation Théorique | Formation Pratique |
| * Enseignement théorique * Travaux dirigés * Travaux pratiques * Conférences | * Etude de cas * Travaux de réalisation * Projets/ Projets tutorés * Voyages d’Etudes * Stages en entreprise |

* 1. Architecture des formations pédagogiques

Le recrutement des étudiants à l’ENI se fait uniquement par voie de concours d’envergure nationale en première année. Les offres de formation organisées à l’Ecole ont été validées par la Commission Nationale d’Habilitation (CNH). Au sein de l’ENI, il existe deux mentions et cinq parcours. Le tableau 2 récapitule les mentions et les parcours au sein de l’École

Tableau 2 Mention et parcours au sein de l’ENI

|  |  |
| --- | --- |
| **Mention** | **Parcours** |
| **Informatique** | Génie logiciel et Base de Données (GB) |
| Administration des Systèmes et Réseaux (ASR) |
| Informatique Générale (IG) |
| **Intelligence Artificielle** | Gouvernance et Ingénierie de Données (GID) |
| Objets Connectés et Cyber sécurités (OCC) |

La figure 2 représente l’architecture des études correspondant au système LMD.

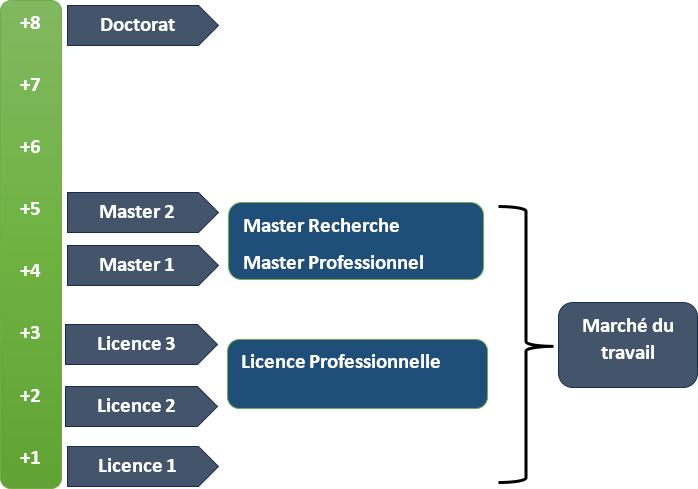


Figure 2 l’architecture des études correspondant au système LMD.

La licence peut avoir une vocation générale ou professionnelle. Le master peut avoir une vocation professionnelle ou de recherche. L’accès en première année de MASTER se fait automatiquement pour les étudiants de l’Ecole qui ont obtenu le diplôme de Licence Professionnelle. Le tableau 3 illustre la liste des formations existantes à l’ENI.

Tableau 3 Liste des formations existantes à l’ENI

|  |  |  |
| --- | --- | --- |
|  | FORMATION | |
| LICENCE PROFESSIONNELLE | MASTER |
| Condition admission | Par voie de concours | Par voie de concours pour la mention IA |
| Condition d’accès | Bac de série C, D ou Technique | Être titulaire de licenceprofessionnelle |
| Durée de Formation | 3 ans | 2 ans |
| Diplôme délivré | Diplôme de Licence  Professionnelle | Diplôme de Master Professionnel  Diplôme de Master Recherche |

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s’inscrire directement dans une Ecole Doctorale.

Les étudiants diplômés de l’Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (Canada, Suisse, France, …)

### Relation de l’ENI avec les organismes externes

Les stages effectués chaque année par les étudiants mettent l’Ecole en rapport permanent avec plus de 400 entreprises et organismes publics, semi-publics et privés, nationaux et

internationaux. L’Ecole dispose ainsi d’un réseau d’entreprises, de sociétés et d’organismes publics et privés qui sont des partenaires par l’accueil en stage de ses étudiants, et éventuellement pour le recrutement après l’obtention des diplômes par ces derniers. Les compétences que l’Ecole cherche à développer chez ses étudiants sont l’adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l’expérimentation et l’innovation.

En effet, la vocation de l’ENI est de former des licenciés et des ingénieurs de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d’évoluer professionnellement dans des secteurs d’activité variés intégrant l’informatique. Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l’Ecole et les besoins évolutifs du marché de l’emploi.

Parmi les sociétés, les entreprises et les organismes partenaires de l’Ecole, on peut citer

: ACCENTURE Mauritius, AKATA Goavana, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie (AUF), AXIAN, B2B, Banque Centrale, , BIANCO, BlueLine, CNaPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, COLAS, Direction Générale des Douanes, DLC, E-Tech Consulting, , FID, FIHARY Soft, FTM, GNOSYS, GENIUS AT WORK, Hello Tana, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MANAO, MEF, MEN, MESupRES, MFB, ,

MININTER, Min des Postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SG Madagasikara SMMC, SMMEC, SNEDADRS Antsirabe, Sénat, Société d’Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN’I BETSILEO, WWF, UGD, ARATO, MANAO, MNDPT, NG ACADEMY.NG, Relia …

### Débouchés professionnels et diplômés

Les formations proposées par l’Ecole permettent aux diplômés d’être immédiatement opérationnels sur le marché du travail avec la connaissance d’un métier complet lié à l’informatique aux TIC.

L’Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante. Les diplômés en LICENCE et en MASTER issus de l’ENI peuvent faire carrière dans différents secteurs.

L’Ecole bénéficie aujourd’hui de 40 années d’expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C’est une Ecole Supérieure de référence en matière informatique.

D’une manière générale, les diplômés de l’ENI n’éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l’ENI recommande à ses diplômés de promouvoir l’entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d’études. Le tableau 4 représente les débouchés éventuels des jeunes diplômés.

Tableau 4 Débouchés éventuels des jeunes diplômés

|  |  |
| --- | --- |
| **LICENCE** | **MASTER** |
| * Analyste - Programmeur * Administrateur de site web/de portail web * Assistant Informatique et internet * Chef de projet web ou multimédia * Développeur Informatique ou multimédia * Intégrateur web ou web designer * Hot liner/Hébergeur Internet * Agent de référencement * Technicien/Supérieur de help desk sur Informatique * Responsable de sécurité web * Administrateur de réseau | * Administrateur de réseau et système * Architecture de système d’information * Développeur d’applications * Ingénieur réseau * Webmaster / Web Designer * Concepteur et réalisateur d’application * Directeur du système d’informations * Chef de projet informatique * Responsable de sécurité informatique * Consultant fonctionnel ou freelance |

### Ressources humaines

Les ressources humaines sont citées ci-dessous selon leurs responsabilités :

* + - * Directeur de l’Ecole : Monsieur MAHATODY Thomas, Docteur HDR
      * Responsable de la Mention « Informatique » : Monsieur RABETAFIKA Louis Haja, Maître de Conférences
      * Responsable de la Mention « Intelligence Artificielle » : Monsieur DIMBISOA William Germain, Maître de Conférences
      * Responsable du Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian, Assistant d’Enseignement Supérieur et de Recherche
      * Responsable du Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA, Assistant d’Enseignement Supérieur et de Recherche
      * Responsable du Parcours « Informatique Générale » : Monsieur GILANTE Gesazafy, Assistant d’Enseignement Supérieur et de Recherche
      * Responsable du Parcours « Gouvernance et Ingénierie de Données » : Madame RATIANANTITRA Volatiana Marielle, Maître de Conférences
      * Responsable du Parcours « Objets Connectés et Cybersécurité » : Monsieur RAZAFIMAHATRATRA Hajarisena, Maître de Conférences

L’ENI compte quinze (15) enseignants permanents dont un (01) Professeur Titulaire, un (01) Professeur, un (01) Docteur HDR, huit (08) Maîtres de Conférences, quatre (04) Assistants d’Enseignement Supérieur et de Recherche, dix (10) enseignants vacataires, quarante un (41) personnel administratif.

# Chapitre 2 : Présentation de Lycée Loterana Rakoto Andrianarijaona

Cette section sera consacrée à la description de Lycée Loterana Rakoto Andrianarijaona (LLRA).

## 2.1 Information d’ordre générale

Le Lycée Loterana Rakoto Andrianarijaona, en abrégé LLRA, est un établissement d’enseignement général. Le siège de l’Ecole se trouve à Ivory Atsimo à Fianarantsoa. L’adresse pour la prise de contact avec l’Ecole est la suivante : Lycée Loterana Rakoto Andrianarijaona, (LLRA) Ivory Atsimo, Fianarantsoa. Téléphone : 034 29 923 86.

## 2.2 Historiques en général

Le Lycée Loterana Rakoto Andrianarijaona était créé en 1896, au début, seules les filles du Betsileo plus de 14 ans y étudiaient pour apprendre à lire pendant 3 ans de formation.

En 1953, le Lycée Loterana Rakoto Andrianarijaona a commencé l’enseignement général jusqu’à la classe de 7éme et il portait le nom d’Ecole primaire élémentaire privé mixte et section ménagère.

En 28 Juin 1962, le lycée a obtenu l’autorisation d’ouverture de la classe de 6ème et 5ème et le 02 Octobre 1969 de la classe de 4ème et 3ème. Ainsi, l’Ecole primaire élémentaire privé mixte et section ménagère devenait « Collège Luthérien Ivory Atsimo » jusqu’en 2001.

L’année scolaire 1983-1984, la classe de 2nde a été ouverte pour la première fois. Et l’année scolaire 2001-2002 a été marqué par l’ouverture de la classe de Terminale A et depuis cela, son nom est devenu Lycée Loterana Rakoto Andrianarijaona.

L’année scolaire 2004-2005, la classe de terminale S ou scientifique a été ouverte.

## 2.3 Ces dirigeants se sont relayes pour diriger l’établissement jusqu’à aujourd’hui :

* Mlle RASMUSSEN
* Mlle FRIDA HOYEM
* Mlle MADSEN
* Mlle MARTA LØTVEIT
* Mlle ANNA SKJŒVOLD
* Mlle KAMILLA TVEIT
* Mlle ANNEKEN RECH
* Mlle ASTRID GIDSKEHAUG
* Mlle ELDRID AAMLAND
* Mlle HELG EIKELAND
* Mlle INGEBORG BERG
* Mr. RAZANADAHY
* Mr. RASOLOFONDRADIMBY ANDRIANTSOA

## Organigramme

Le Lycée Loterana Rakoto Andrianarijaona est constitué d’un comité d’administration, une Direction collégiale et 5 départements

La figure 3 représente l’organigramme actuel du LLRA

Figure 3 Organigramme LLRA

## 2.5 Ressources humaines

Les membres de la direction collégiale avec leurs responsabilités :

* + Président du Comité d’Administration et Responsable de l’Infrastructure : RATEFIARIVONY Rado Heriniaina
  + Proviseur et Responsable administratif : RASOLOFONDRADIMBY Andriantsoa
  + Responsable Pédagogique : ANDRIAMADSON Solofo
  + Aumônier et Gestionnaire : Pasteur RANARSON Remond
  + Responsable de logistique : RATOMBINIAINA Tiana

Le lycée Loterana Rakoto Andrianarijaona compte :

* 9 enseignants permanents
* 5 personnels administratifs
* 10 chargés de cours

## Chapitre 3. Description du projet

Dans ce chapitre, explorons la formulation du projet, l’objectif et les besoins de  
l’utilisateur, les moyens nécessaires à la réalisation du projet, et enfin, les résultats attendus.

### 3.1 Formulation

Le projet vise à développer une application web destinée aux responsables du Lycée Loterana Rakoto Andrianarijaona Ivory Atsimo Fianarantsoa. Ce lycée assure la gestion des étudiants depuis le dépôt de leurs dossiers jusqu'à la fin de leur cursus, tout en respectant les démarches administratives requises d’où la gestion des inscriptions et de l’écolage. Le but c’est d'établir des politiques et des procédures claires pour les inscriptions, de gérer efficacement les frais de scolarité, et de suivre les éventuels retards de paiement afin de garantir leur conformité avec les règles de l'établissement. Ce projet a pour finalité de concevoir un logiciel informatique qui simplifie ces tâches administratives et optimise leur gestion.

### 3.2 Objectif et besoins de l’utilisateur

L’objectif du projet est de concevoir et de réaliser une application pour gérer les inscriptions et l’écolage. Cette application vise à automatiser et à informatiser ces processus pour simplifier et optimiser leur gestion. Elle permettra ainsi aux utilisateurs de :

* Gérer les inscriptions,
* Gérer les classes,
* Gérer les écolages,
* Gérer les professeurs,
* Gérer les utilisateurs.

### 3.3 Moyens nécessaires à la réalisation du projet

Les moyens nécessaires à la réalisation du projet incluent les ressources matériels, logiciels, et les ressours humaines.

#### 3.3.1 Moyens matériels

Les ressources matérielles présentées dans le tableau 5 ont été indispensables pour mener à bien ce projet.

Tableau 5 Moyens matériels utilisés.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Désignation** | **CPU** | **RAM** | **HDD** | **SE** |
| PC | Intel core i5 | 8 GO | 500 GO | Windows 10  Professionnel |

#### 3.3.2 Moyens logiciels

Les divers logiciels pour la réalisation de ce projet sont aperçus dans le Tableau 6.

Tableau 6 Logiciels utilisés.

|  |  |  |
| --- | --- | --- |
| **Logiciel** | **Version** | **Utilité** |
| Chrome | 120.0 (64 bits) | Navigateur web |
| Visual Studio Code | 1.80.0 | Outil de modélisation |
| PostgreSQL | 3.2.3 | Système de gestion de base de données (SGBD) |
| Microsoft Office | Microsoft Office 2016 | Documentation et présentation |

#### 3.3.3 Moyens humains

Les personnes mentionnent sur le tableau 7 ont contribué à la réalisation de ce projet.

Tableau 7 Moyens humains pour la réalisation du projet

|  |  |
| --- | --- |
| **Acteur** | **Rôle** |
| Responsable de l’école | Fournisseurs de donne |
| Stagiaire | Concepteur et réalisateur |

### 3.4 Résultats attendus

Au terme de ce projet, nous attendons d’obtenir une application web permettant de :

* + - Faciliter la gestion des paiements des frais de scolarité,
    - Suivre les paiements et générer des factures,
    - Générer en PDF les listes des paiements non effectués par classe,
    - Générer en PDF les informations détaillées de chaque élève,
    - Générer en PDF les informations des professeurs,
    - Proposer une interface utilisateur ergonomique et intuitive, adaptée aux besoins des responsables de l’établissement.

### 3.5 Chronogramme du projet

Tout projet, pour devenir un succès, doit se baser sur une planification bien régie. La méthodologie pour la conduite d'un projet est un ensemble de processus et de techniques utilisés pour planifier, organiser, exécuter et surveiller un projet.

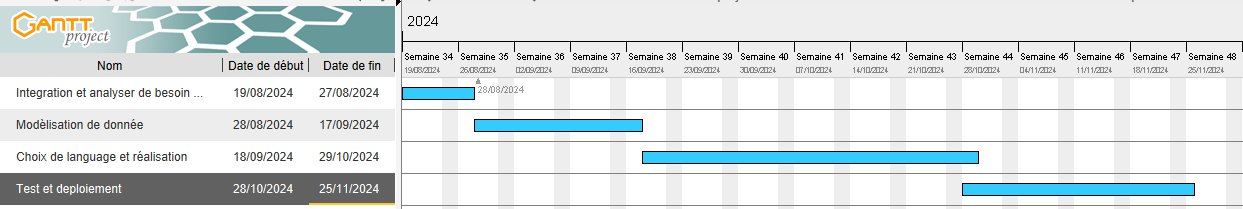
La figure 4 illustre le chronogramme du projet.

Figure 4 Chronogramme de la réalisation du projet

# Chapitre 4. Analyse préalable

Dans ce quatrième chapitre nous allons procéder à l’analyse de l’existant, en prenant

Compte les améliorations pour l’élaboration de notre projet.

### 4.1 Analyse de l’existant

#### 4.1.1 Organisation actuelle

En tant qu’établissement scolaire, le fonctionnement du lycée repose principalement sur la gestion des inscriptions et des écolages des élèves. La première étape consiste à inscrire chaque élève à travers une fiche d’inscription disponible au service de scolarité. Cette fiche rassemble des informations importantes telles que le nom et le prénom de l’élève, sa date de naissance, les noms de ses parents ou tuteurs, l’année scolaire concernée, ainsi que la classe attribuée. Cette inscription s’accompagne du paiement des droits d’inscription ainsi que de la remise des documents requis. Par ailleurs, pour continuer leurs études, les élèves sont tenus de payer un écolage mensuel. Ce paiement est enregistré sur une carte d’écolage, qui sert à suivre les montants versés et constitue une preuve des transactions effectuées. Tous ces paiements se font directement auprès du service de scolarité, qui assure la gestion et le suivi des contributions financières des élèves.

#### 4.1.2 Inventaire des moyens matériels et logiciels

#### a. Moyens matériels

Pour les moyens matériels

Tableau 8 Moyens matériels utilisés

|  |  |  |
| --- | --- | --- |
| **Désignation** | **Caractéristiques** | **Nombre** |
| PC | HP, Intel(R) Core (TM) i5 CPU, RAM 4 Go, 2.4 GHz (4CPUs), HDD 500Go | 03 |
| Imprimantes | Canon MFC 633 | 02 |

#### b. Moyens logiciels

Pour accompagner les moyens matériels, nous avons aussi les moyens logiciels.

Tableau 9 Moyens logiciels utilisés

|  |  |
| --- | --- |
| **Logiciel** | **Utilité** |
| Microsoft office 2019 | Saisie et enregistrement des données |

### 4.2 Critique de l’existant

Cette section est consacrée à l’examen critiques et remarques sur les informations

recensés auprès de l’existant analysé en mettant en avant les avantages et les faiblesses du

système actuel.

1. Points faibles du système actuel :

Les collectes et l'enregistrement des informations sont faits sur des supports physiques et manuellement. Ce qui risque d’être égarés, abîmés ou confondus avec d’autres documents.

1. Points forts du système actuel :

En traitant les dossiers manuellement, le lycée peut potentiellement maintenir un certain niveau de confidentialité, car l'accès physique aux documents est généralement plus restreint et Pas besoin d’électricité.

#### 4.2.1 Conception avant -projet

Pour résoudre à ce problème, deux solutions s’offrent à nous :

Solution 1 : Concevoir et développer une application optimiser aux besoins de l’utilisateur

Solution 2 : télécharger une application existante

Le Tableau 10 établit la comparaison des caractéristiques des 2 solutions proposées.

Tableau 10 Comparaison des deux solutions

|  |  |  |
| --- | --- | --- |
| **Solutions** | **Solution 1** | **Solution 2** |
| Avantages | * Plus rapide et nécessite moins de ressources | - Développer à partir de zéro   * Répondre aux besoins d’utilisateur   - Ajouter des nouvelles fonctionnalités  - Personnalisation de la plateforme  - Meilleur rapport qualité |
| Inconvénients | * Ne permet pas une possibilité de maintenir le code source | * Coût de temps considérable lors de la réalisation jusqu’à la mise en place. |

Après analyse approfondie des deux solutions proposées, nous avons opté pour la solution deux raison de l’adaptabilité au besoin de l’utilisateur.

#### 4.2.2 Méthodes de conception et outils utilisés

Avant de débuter, il est nécessaire de bien choisir une méthode de conception appropriée et aussi les outils adaptés. Afin d’éviter les bugs, et assurer l’optimisation, l’adaptabilité, la flexibilité, la robustesse du système, il faut commencer par choisir une méthode claire ainsi qu’un langage de modélisation.

### a. Choix de méthode de conception et de modélisation

La modélisation est une étape cruciale pour la réussite de tout projet. Quelle que soit l’architecture applicative choisie pour l’implémentation, la conception du système devra faire appel à une technique de modélisation des données dédiée au décisionnel. Le modèle d'un système informatique sert de document d'échange entre clients et développeurs, d'outil de conception, de référence pour le développement, et aussi de référence pour la maintenance et l'évolution. Concernant le choix de méthode de conception, on a mis en comparaison les avantages et les inconvénients de chacune des deux méthodes de conception 2TUP et MERISE.

Tableau 11 Comparaison des méthodes de conception et de modélisation

|  |  |  |
| --- | --- | --- |
|  | **2TUP** | **Merise** |
| **Avantages** | * Indépendant d’un langage de programmation * Architecture bidirectionnelle | * Structuration en étape et en point de contrôle * Approche conceptuelle, se focalisant sur les métiers et les besoins associées * Approche systématique |
| **Inconvénients** | * La mise en pratique nécessite un apprentissage * L’intégration dans un processus n’est pas triviale | * Ne s’occupe pas de l’interface utilisateur * Ne prends pas ne charge la * Maintenance ni la deuxième informatisation |

Le choix d’utilisé 2TUP est du fait qu’on peut facilement visualiser l'architecture logicielle, identifier les risques potentiels et les gérer efficacement, et planifier et suivre les différentes étapes du développement. De plus, la combinaison de 2TUP au langage de modélisation UML permet de mieux comprendre les besoins des utilisateurs [1].

**Présentation de 2TUP :**

2TUP (Two-Track Unified Process) est une méthodologie de gestion de projet et de développement logiciel qui se concentre sur deux volets principaux : le processus de développement et le processus de gestion. Elle propose une approche itérative et incrémentale, permettant une meilleure adaptation aux changements et une livraison progressive des fonctionnalités. En combinant des cycles courts avec des livraisons fréquentes, 2TUP garantit une forte collaboration entre les parties prenantes, une gestion proactive des risques et une amélioration continue du produit. Cette méthode est particulièrement adaptée aux projets complexes nécessitant une flexibilité et un suivi rigoureux.

### b. Choix du système de gestion de base de données

Pour choisir le système de gestion de base de données (SGBD) le mieux adapté à notre projet, il est important de comparer les avantages et les inconvénients de deux solutions couramment utilisées : MongoDB et PostgreSQL. Le choix de ces deux SGBD repose sur leur popularité, leur performance et leur adéquation avec les besoins spécifiques de notre application. MongoDB, un système NoSQL, offre une grande flexibilité pour le stockage de données non structurées, tandis que PostgreSQL, un SGBD relationnel, est reconnu pour sa robustesse et son respect des normes ACID. Cette comparaison permettra d’identifier lequel de ces outils répond le mieux aux exigences fonctionnelles et techniques de notre projet.

Tableau 12. Comparaison des SGBD

|  |  |  |
| --- | --- | --- |
| **SGBD** | **Avantages** | **Inconvénients** |
| MongoDB | * Flexibilité du schéma. * Performances élevées pour des données non structurées. * Base de données NoSQL. | * Moins approprié pour des données strictement relationnelles. * Manque de prise en charge SQL. * Base de données volumineuses. |
| PostgreSQL | * Robuste, offrant la possibilité de gérer des données structurées de manière efficace. * Requête SQL puissante. * Fiabilité et évolutivité. | * Configuration et maintenance complexe. * Rigidité de schéma * Performances potentiellement |

Dans le cadre de ce projet, le choix s'est porté sur PostgreSQL comme système de gestion de base de données (SGBD) pour sa Requête SQL puissante et Fiabilité et évolutivité.

**Présentation de PostgreSQL :**

PostgreSQL est un SGBDR (Système de Gestion de Base de Données Relationnelle) reconnu pour sa robustesse, sa conformité aux normes ACID, et sa capacité à gérer des données complexes de manière efficace. PostgreSQL offre des fonctionnalités avancées telles que le support des types de données personnalisés, des extensions, et une gestion optimisée des transactions, ce qui le rend particulièrement adapté pour des projets nécessitant une gestion fiable et sécurisée des données [9].

### c. Outil de modélisation

Pour modéliser efficacement un système, il est important de choisir un outil adapté aux besoins spécifiques du projet. Visual Paradigm et StarUML sont deux outils couramment utilisés pour la modélisation UML. Ces deux logiciels ont été retenus pour leur popularité, leurs fonctionnalités principales, et leur capacité à répondre aux exigences des projets informatiques.

Tableau 13. Comparaison Outil de modélisation

|  |  |  |
| --- | --- | --- |
| **Outil de modélisation** | **Visual Paradigm** | **Start UML** |
| Avantages | - Complet et puissant.  -Collaboration en équipe.  - Support avancé d’UML | - Léger et rapide.  - Simple à utiliser.  - Moins coûteux. |
| Inconvénients | |  | | --- | |  |  |  | | --- | | - Version payante coûteuse pour les fonctionnalités avancées.  - Peut être complexe pour les débutants. | | -Fonctionnalités limitées.  -Pas de collaboration native. |

Pour ce projet, nous avons choisi Visual Paradigm car il offre un support complet pour UML et d'autres méthodologies de modélisation comme BPMN. Cet outil permet également une collaboration efficace en équipe et répond aux besoins de modélisation avancés nécessaires pour notre projet. Bien qu'il puisse être complexe au départ.

**Présentation de Visual Paradigme :**

Visual Paradigm est un outil de modélisation puissant et polyvalent, largement utilisé pour la conception et la gestion des projets logiciels. Il prend en charge plusieurs standards de modélisation, notamment UML (Unified Modeling Language), BPMN (Business Process Model and Notation), et ERD (Entity-Relationship Diagram).

### d.Langage de programmation

Nous avons comparé JavaScript et PHP car ce sont deux technologies essentielles dans le développement web, mais avec des usages distincts. PHP est principalement utilisé pour la gestion côté serveur, comme la génération de contenu dynamique et l’interaction avec les bases de données, tandis que JavaScript se distingue par sa polyvalence, étant utilisé à la fois côté client pour l'interactivité et côté serveur via Node.js. Cette comparaison permet de choisir la technologie la plus adaptée en fonction des besoins spécifiques du projet, qu’il s’agisse d’un backend robuste ou d’applications interactives et en temps réel.

Tableau 14 Comparaison entre PHP et JS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Critères** | | |  | | --- | | **PHP** | | |  | | --- | | **JavaScript** | |
| Type | * Langage de script côté serveur. | * Langage polyvalent (client et serveur) |
| Usage principal | * Génération dynamique de contenu côté serveur. | |  | | --- | |  |  |  | | --- | | * Dynamique. | |
| Avantages | - Large communauté et nombreux frameworks   |  |  | | --- | --- | |  | - Excellente compatibilité avec les bases de données. | | - Polyvalent (Node.js pour le backend).  - Rapide pour des applications en temps réel.  - Exécuté directement dans le navigateur. |
| Inconvénients | - Limité au côté serveur. | - Plus complexe pour gérer des bases de données sans framework. |

En tenant compte des différences entre ces deux langages de programmation, le choix de JS nous semble plus adéquate pour réaliser le projet dû à son rapide pour des applications en temps réel et pour sa Dynamicité.

**Présentation de Javascript**

JavaScript est un langage de programmation polyvalent et dynamique, utilisé à la fois côté client et côté serveur grâce à des environnements comme Node.js. Initialement conçu pour ajouter de l'interactivité aux pages web, il est aujourd'hui une technologie incontournable pour le développement d'applications web modernes, mobiles et même desktop. JavaScript permet de créer des interfaces utilisateur réactives, des Single Page Applications (SPA) et des fonctionnalités en temps réel comme les chats ou notifications. Avec une vaste communauté, des frameworks populaires tels que React, Angular, ou Vue.js, et des outils comme Node.js, JavaScript s'impose comme une solution complète pour construire des applications web performantes et interactives.

**d.1 Choix de Framework JavaScript**

**Frontend**

Nous avons comparé React.js et Vue.js car ce sont deux technologies JavaScript les plus populaires pour le développement d’interfaces utilisateur modernes. React.js, maintenu par Facebook, est connu pour sa flexibilité et sa grande communauté, idéal pour les projets complexes nécessitant une personnalisation poussée. Vue.js, quant à lui, est apprécié pour sa simplicité, sa courbe d’apprentissage douce et son intégration facile dans les projets existants. En comparant ces deux frameworks, nous cherchons à identifier celui qui répond le mieux aux besoins de notre projet, notamment en termes de rapidité de développement, d’évolutivité, et de gestion des composants dynamiques.

Tableau 15 Comparaison entre VueJS et ReactJS

|  |  |  |
| --- | --- | --- |
|  | Avantage | Inconvénient |
| VueJS | * Une petite taille * La rapidité de développement et la flexibilité * Une intégration simple et facile * Une documentation détaillée | * Des soucis de stabilité * Le manque d’évolutivité * Une communauté restreinte |
| ReactJS | * Des composants réutilisables * Un contenu référençable * La rapidité * La communauté * La lisibilité du code | * Les modèles des composants sont codés en JSX au lieu du HTML * Offre tellement de possibilités aux développeurs que cela peut en devenir un défaut * Demande assez de pratique |

Le choix de ReactJS comme technologie pour ce projet s'explique par plusieurs raisons pertinentes. Tout d'abord, il permet de créer des interfaces utilisateur dynamiques et performantes grâce à son système de composants réutilisables, ce qui favorise une structure modulaire et une maintenance simplifiée du code. De plus, la large communauté de développeurs et la popularité de React assurent un soutien constant, ainsi qu'un accès à une abondante documentation et à des bibliothèques complémentaires.

**Présentation de ReactJS**

ReactJS est une bibliothèque JavaScript libre. Il s’agit d’un technologie web convivial qui permet la conception des applications web via des composants dépendants. Recommandé par les entreprises adeptes de JavaScript, React peut servir à créer des interfaces graphiques front. Grâce à sa flexibilité et son API, on utilise plus facilement les composants Ces caractéristiques font de React une solution robuste et évolutive, parfaitement adaptée aux exigences de notre projet et aux besoins futurs de développement [6].

**Backend**

Nous avons comparé Express.js et Next.js car ce sont deux technologies couramment utilisées dans le développement web backend et full-stack. Express.js est un framework minimaliste et flexible pour le développement backend avec Node.js, tandis que Next.js est un framework full-stack basé sur React, offrant des fonctionnalités avancées comme le rendu côté serveur (SSR) et la génération de sites statiques (SSG). Cette comparaison nous permet de choisir la technologie qui s’aligne le mieux avec les exigences de notre projet, notamment en termes de simplicité, flexibilité, et fonctionnalités spécifiques.

Tableau 16 Comparaison ReactJS et NextJS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Critères** | | |  | | --- | | **Express.js** | | |  | | --- | | **Next.js** | |
| Type | |  | | --- | | * Framework backend pour Node.js. |  |  | | --- | |  | | * Framework full-stack basé sur React. |
| Usage principal | |  | | --- | | * Création d’API et de serveurs web backend. |  |  | | --- | |  | | * Développement d’applications web complètes avec SSR et SSG. |
| Avantages | |  |  | | --- | --- | | - Léger et hautement personnalisable. | | | - Idéal pour les microservices. |  |  | | --- | |  | | - Intégré avec React pour le front et back.  - Prend en charge SSR et SSG par défaut. |
| Inconvénients | |  | | --- | | - Nécessite plus de configuration pour les fonctionnalités avancées. | | - Plus complexe pour les projets backend uniquement. |

Nous avons choisi Express.js car notre projet nécessite principalement un framework backend pour gérer les API et les interactions avec la base de données. Express.js est léger, simple à configurer et parfaitement adapté pour développer des serveurs RESTful, ce qui correspond aux besoins de notre projet. De plus, sa flexibilité et sa large communauté garantissent un support fiable et des ressources abondantes pour résoudre d’éventuels problèmes.

**Présentation de ExpressJS**

Express.js est un framework web minimaliste et flexible pour Node.js, conçu pour développer des applications backend et des API RESTful. Il offre des outils puissants pour la gestion des requêtes HTTP, des routes, des middlewares et des erreurs. Grâce à sa simplicité et son architecture modulaire, Express.js permet aux développeurs de personnaliser et d’étendre leurs applications selon leurs besoins. Utilisé dans des projets de toutes tailles, Express.js est une solution idéale pour construire des systèmes backend efficaces et évolutifs.

# Chapitre 5. Analyse conceptuelle

Dans ce cinquième chapitre, nous nous pencherons sur l’analyse conceptuelle,

Impliquant l’analyse du système existant. Cette analyse conceptuelle découle de l’observation de la réalité que l’on souhaite représenter de manière plus efficace d’où la présentation des divers aspects de fonctionnalités de notre application. La conception décrit en premier lieu, le fonctionnement de notre système par une étude générale basée sur la spécification des besoins, puis une étude détaillée du projet sur plusieurs approches

### 5.1 Dictionnaire des données

Le Dictionnaire des données est une collection de métadonnées ou de données de référence nécessaire à la conception d’une base de données relationnelle. C’est un document qui permet de recenser, de classer et de trier toutes les informations (les données) collectées lors de l’étude des documents. C’est le référentiel principal de l’entreprise, sur lequel s’appuient les décisions.

Tableau 17 Dictionnaire des données.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rubrique** | **Description** | **Structure** | | **Remarque** |
| **Type** | **Longueur** |
| Adress\_Etu | Adresse de l’étudiant | AN | 20 |  |
| Adress\_Par | Adresse des parents | AN | 20 |  |
| Classe | Classe de l’étudiant | AN | 7 |  |
| Date\_paye | Date des paiements | D | 10 | jj/mm/aaaa |
| Datenaiss | Date de naissance de l’étudiant | D | 10 | jj/mm/aaaa |
| Ecole\_Org | Ecole d’origine | AN | 10 |  |
| Frais\_scol | Montant de frais de scolarité par mois | N | 6 |  |
| IdAnne | Identifiant année scolaire | AN | 2 |  |
| Idclasse | Identifiant classe | AN | 2 |  |
| Idpaye | Identifiant écolage | AN | 5 |  |
| Iduser | Identifiant utilisateur | AN | 2 |  |
| idprof | Identifiant Professeur | AN | 2 |  |
| Matiere | Matière a enseigné | AN | 10 |  |
| Matricule | Identifiant étudiant | AN | 5 |  |
| Mois | Mois de l’année | AN | 7 |  |
| Montant | Montant en Ariary | N | 6 |  |
| Mots\_De\_.Passe | Mots de passe | AN | 12 |  |
| Nom\_etu | Nom de l’étudiant | AN | 20 |  |
| Nom\_Anne | Année scolaire | AN | 5 |  |
| nomClasse | Nom de la classe | AN | 10 |  |
| nomMere | Nom de mère d’élevé | AN | 20 |  |
| nomPere | Nom de père de l’élevé | AN | 20 |  |
| nomRes | Nom de la Responsable | AN | 20 |  |
| Num\_auto | Numéros d’autorisation des professeurs | AN | 5 |  |
| Place | Nombre de places d’une salle | N | 3 |  |
| Remarque | Remarque pour l’élève | AN | 30 |  |
| Salle | Salle de classes | AN | 5 |  |
| Tel | Numéro téléphone des parents de l’étudiant | N | 13 | +261 .. …. .. |
| Type | Type de payement | N | 7 |  |

***Légende :***

|  |  |  |
| --- | --- | --- |
| AN : Alphanumérique | D : Date | N : Numérique |

### 5.3 Règles de gestion

Les règles de gestion sont des informations de commande qui définissent la suite des opérations effectuées pour passer des données de base au résultat. Ainsi, ils indiquent l’ensemble des liens existants sur les données manipulées, les actions effectuées par les postes de travail impliqués et les différentes formules de calcul. Les règles de gestion associées au domaine d’étude sont :

* **RG1** : L'administrateur est responsable de créer et gérer les profils des étudiants dans le système.
* **RG2 :** Chaque du huit du jours du mois les étudiants qui n’ont pas payer les frais de scolarité sont considéré en retard.
* **RG 3** : L'administrateur doit enregistrer les frais d'inscription payés par les étudiants lors de l'inscription.
* **RG 4** : L'administrateur doit pouvoir définir les frais d'écolage par classe et par année scolaire.
* **RG 5** : Les enfants des professeurs bénéficient d’une réduction de vingt pourcents sur leurs frais d’inscription.
* **RG 6 :** L'administrateur peut gérer les informations des enseignants (ajouter, modifier, supprimer).
* **RG 7** : L'administrateur peut générer des rapports pour identifier les paiements en retard ou non effectués.

### 5.4. Représentation et spécification des besoins

#### 5.4.1. Diagramme des cas d’utilisation

En langage de modélisation unifié (UML), un diagramme de cas d'utilisation peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier. Identification des acteurs : Un acteur c’est une entité extérieure en interaction avec le système. Les acteurs comprennent aussi bien humains (utilisateurs) que d’autre type (machine, autre système, tout élément de l’environnement).

La figure 5 comment représente un acteur :

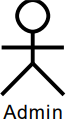


Figure 5 Acteur

Dans notre cas, on a juste un seul acteur et c’est l’Administrateur.

Identification des cas d’utilisation :

Un cas d’utilisation permet de décrire l’interaction entre les acteurs (utilisateurs du cas) et le système. La description de l’interaction est réalisée suivant le point de vue de l’utilisateur. La représentation d’un cas d’utilisation met en jeu trois concepts : l’acteur, le cas d’utilisation et l’interaction entre l’acteur et le cas d’utilisation. La figure 8 montre l’interaction entre l’acteur et le cas d’utilisation

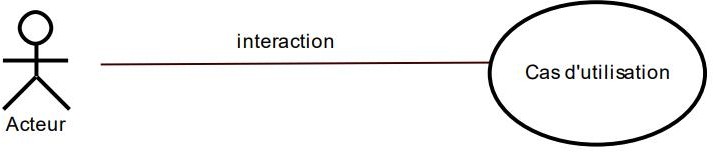


Figure 8. Interaction entre acteur et cas d'utilisation

Relations possibles entre cas d’utilisation UML définit trois types de relations standardisées entre cas d’utilisation qui sont :

1. <include> : Le cas d'utilisation incorpore explicitement et de manière obligatoire un autre cas d'utilisation à l'endroit spécifié.
2. <extend> : Le cas d'utilisation incorpore implicitement de manière facultative un autre cas d'utilisation à l'endroit spécifié.
3. Généralisation : le cas d'utilisation descendants hérite des propriétés de leur parent.

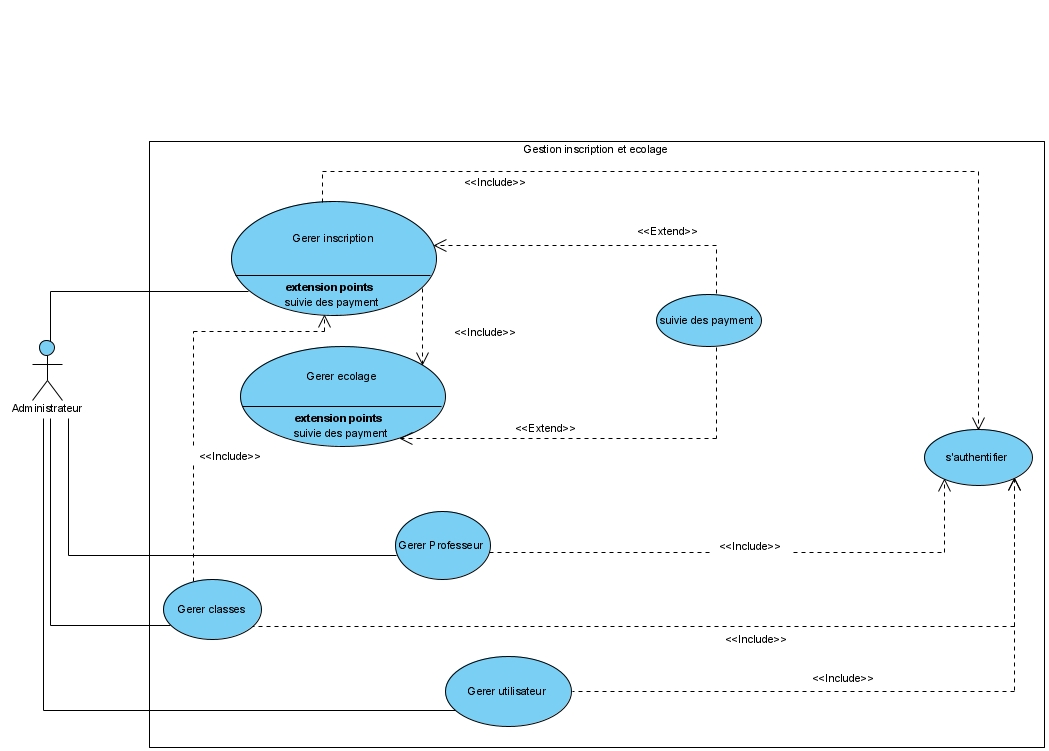
Nous allons voir dans la figure 6 les cas d’utilisation conforme à notre projet

Figure 6 Diagramme de cas d'utilisation global

#### 

#### 5.4.2 Description textuelle pour chaque cas d’utilisation

1. Description textuelle S’authentifier

Le tableau 18 illustre la description textuelle du cas d'utilisation S’authentifier. Ce tableau détaille les étapes et les interactions nécessaires pour permettre aux utilisateurs d'accéder à leur compte

Tableau 18 Description textuelle « S’authentifier »

|  |  |
| --- | --- |
| Cas d’utilisation | S’authentifier |
| Acteur | Administrateur |
| Objectif | Se connecter au système |
| Précondition | Possède un compte d’authentification |
| Postcondition | Le tableau de bord s’affiche |
| Scenario nominal | 1. L’administrateur demande la page de connexion. 2. Le système affiche la page de connexion. 3. L’administrateur saisit son adresse email et son mot de passe. 4. Le système vérifie et redirige l’administrateur vers le tableau de bord. |
| **Scénario d’exception** | a. L’administrateur saisit des données incorrectes. b. Le système affiche un message d’erreur. c. Retour au scénario 3. |

2. Description textuelle Gérer classes

L**e** tableau 19 présente une description textuelle détaillée du cas d'utilisation intitulé « Gérer classes » Ce tableau illustre les différentes étapes et interactions nécessaires pour permettre aux utilisateurs de gérer efficacement au sein de l'application.

Tableau 19 Description textuelle « Gérer classes »

|  |  |
| --- | --- |
| Cas d’utilisation | Gérer classes |
| Acteur | Administrateur |
| Objectif | Gérer les classes |
| Précondition | L’administrateur est connecté au système |
| Postcondition | Les classes sont sauvegardées avec succès |
| Scenario nominal | 1. L’administrateur accède à la page des classes. 2. Le système affiche le formulaire pour les classes.   3. L’administrateur saisit les informations de la classe.  4. Le système valide les données et enregistre la classe |
| **Scénario d’exception** | a. Les informations saisies sont incomplètes ou incorrectes.  b. Le système affiche un message d’erreur. |

3. Description textuelle Gérer Inscriptions

Le tableau 20 présente une description textuelle du cas d'utilisation "Gérer Inscriptions". Ce tableau met en lumière les processus impliqués dans la création d'une nouvelle Inscriptions au sein de l'application.

Tableau 20 Description textuelle « Gérer l’inscription »

|  |  |
| --- | --- |
| Cas d’utilisation | **Gérer inscription** |
| Acteur | Administrateur |
| Objectif | **Gérer inscription** |
| Précondition | L’administrateur est connecté au système |
| Postcondition | Les inscriptions sont sauvegardées avec succès |
| Scenario nominal | 1. L’administrateur accède à la page d’inscription.  2. Le système affiche le formulaire d’inscription.  3. L’administrateur saisit les informations de l’élève  4. Le système valide les données et enregistre l’inscription. |
| **Scénario d’exception** | a. Les informations saisies sont incomplètes ou incorrectes.  b. Le système affiche un message d’erreur. |

4. Description textuelle gérer l’écolage

Le tableau 21 fournit une description textuelle du cas d'utilisation « Gérer l’écolage ». Ce tableau explique comment les utilisateurs peuvent gérer les payements des écolages dans le système.

Tableau 21 Description textuelle « Gérer l’écolage »

|  |  |
| --- | --- |
| Cas d’utilisation | Gérer l’écolage |
| Acteur | Administrateur |
| Objectif | Ajouter, modifier ou consulter les frais de scolarité |
| Précondition | L’administrateur est connecté au système |
| Postcondition | Les données des frais d’écolage sont mises à jour |
| Scenario nominal | 1. L’administrateur sélectionne une classe ou un élève. 2. Le système affiche les détails de l’écolage (montant, statut de paiement). 3. L’administrateur modifie ou ajoute les informations. 4. Le système enregistre les mises à jour. |
| **Scénario d’exception** | a. L’administrateur tente de saisir un montant non valide. b. Le système affiche un message d’erreur. |

**5. Description textuelle** **Suivi des paiements**

Le tableau 22 décrit le cas d'utilisation « Suivi des paiements ». Ce tableau détaille les étapes requises pour que les utilisateurs puissent suivre les paiements.

Tableau 22 Description textuelle « Suivi des paiements »

|  |  |
| --- | --- |
| Cas d’utilisation | Suivi des paiements |
| Acteur | Administrateur |
| Objectif | Suivre les paiements effectués et non payés |
| Précondition | Les frais d’écolage ont été définis |
| Postcondition | Le statut des paiements est mis à jour |
| Scenario nominal | 1. L’administrateur accède à la section « Suivi des paiements ». 2. Le système affiche une liste des paiements par classe ou élève. 3. L’administrateur met à jour le statut d’un paiement ou génère un rapport. |
| **Scénario d’exception** | a. Les données affichées sont incomplètes ou incorrectes. b. Le système affiche une erreur. |

**6. Cas d’utilisation : Gérer les** **professeurs**

Le tableau 23 présente la description textuelle du cas d'utilisation « Gérer professeurs ». Ce tableau explique le processus par lequel un utilisateur peut gérer les professeurs.

Tableau 23 Description textuelle « Gérer professeurs »

|  |  |
| --- | --- |
| Cas d’utilisation | Gérer les professeurs |
| Acteur | Administrateur |
| Objectif | Ajouter, modifier ou supprimer les informations des professeurs |
| Précondition | L’administrateur est connecté |
| Postcondition | Les données des professeurs sont mises à jour |
| Scenario nominal | 1. L’administrateur accède à la section « Professeurs ». 2. Le système affiche la liste des professeurs enregistrés. 3. L’administrateur ajoute ou modifie les informations d’un professeur. 4. Le système sauvegarde les modifications. |
| **Scénario d’exception** | a. L’administrateur tente de supprimer un professeur lié à une classe active. b. Le système affiche une erreur. |

**7. Description textuelle Gérer les utilisateurs**

Le tableau 24 illustre la description textuelle pour le cas d’utilisation Gérer utilisateur.

Tableau 24 Description textuelle « Gérer Utilisateur »

|  |  |
| --- | --- |
| Cas d’utilisation | Gérer les utilisateurs |
| Acteur | Administrateur |
| Objectif | Ajouter, modifier ou supprimer les comptes utilisateurs |
| Précondition | L’administrateur possède les droits requis |
| Postcondition | Les données des utilisateurs sont mises à jour |
| Scenario nominal | 1. L’administrateur accède à la section « Utilisateurs ». 2. Le système affiche une liste des comptes existants. 3. L’administrateur crée un nouveau compte ou modifie un compte existant. 4. Le système enregistre les modifications. |
| **Scénario d’exception** | a. L’administrateur tente de supprimer son propre compte. b. Le système affiche un message d’avertissement. |

#### 5.4.3 Priorisation des cas d’utilisation

La priorisation des cas d’utilisation permet de décrire chronologiquement les opérations à effectuer.

Le tableau 25 présente la liste des cas d’utilisations par leur niveau de priorité.

Tableau 25 : Priorité des cas d’utilisations

|  |  |
| --- | --- |
|  | Cas d’utilisation |
| 1 | S’authentifier |
| 2 | Gérer classe |
| 3 | Gérer l’inscription |
| 4 | Gérer l’écolage |
| 5 | Suivi des paiements |
| 6 | Gérer les professeurs |
| 7 | Gérer les utilisateurs |

#### 5.4.4 Diagramme des séquences système pour chaque cas d’utilisation

Les diagrammes de séquences, au sein du langage de modélisation UML, se révèlent être des outils essentiels pour illustrer de manière chronologique le déroulement des interactions des cas d’utilisations entre acteurs et système. Ce type de diagramme illustre chaque scenario dont l’ordre des messages disposes verticalement souligne l’importance de chaque interaction.

Cette section explorera en détail les scenarios les plus significatifs afin de fournir une représentation visuelle éclairante.

**a. Cas d’utilisation : S’authentifier**

La figure 7 illustre le diagramme de séquence du système pour le cas d'utilisation "S’authentifier ". Ce diagramme décrit les interactions entre les différents composants du système lors du processus de connexion d'un utilisateur.

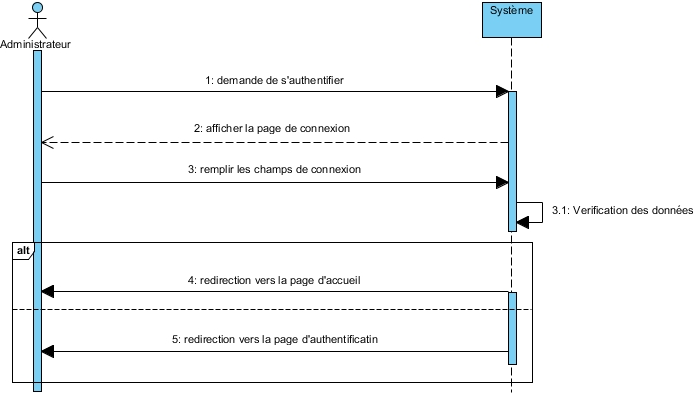


Figure 7 Diagramme de séquence système pour le cas d'utilisation S'authentifier

**c. Cas d’utilisation : Gérer classes**

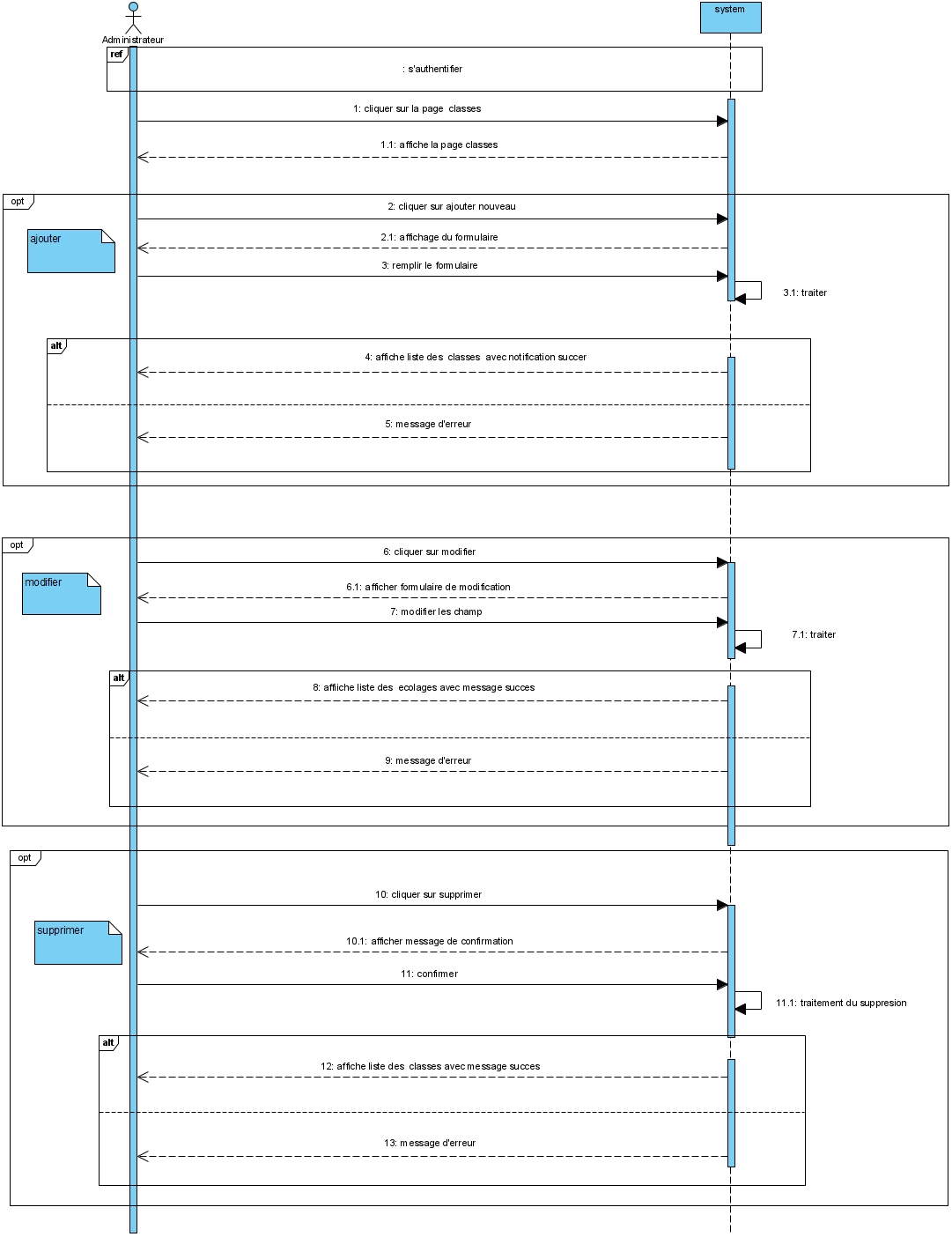
La figure 8 présente le diagramme de séquence du système pour le cas d'utilisation "S’inscrire". Ce diagramme montre les étapes et les interactions nécessaires pour qu’une nouvelle classe soit enregistrer.

Figure 8 Diagramme de séquence système pour le cas d’utilisation gérer classes

d. Cas d’utilisation : Gérer inscription

La figure 9 expose le diagramme de séquence du système pour le cas d'utilisation "gerer incription". Ce diagramme détaille les flux d'information et les actions nécessaires pour permettre à un utilisateur de créer une tâche.

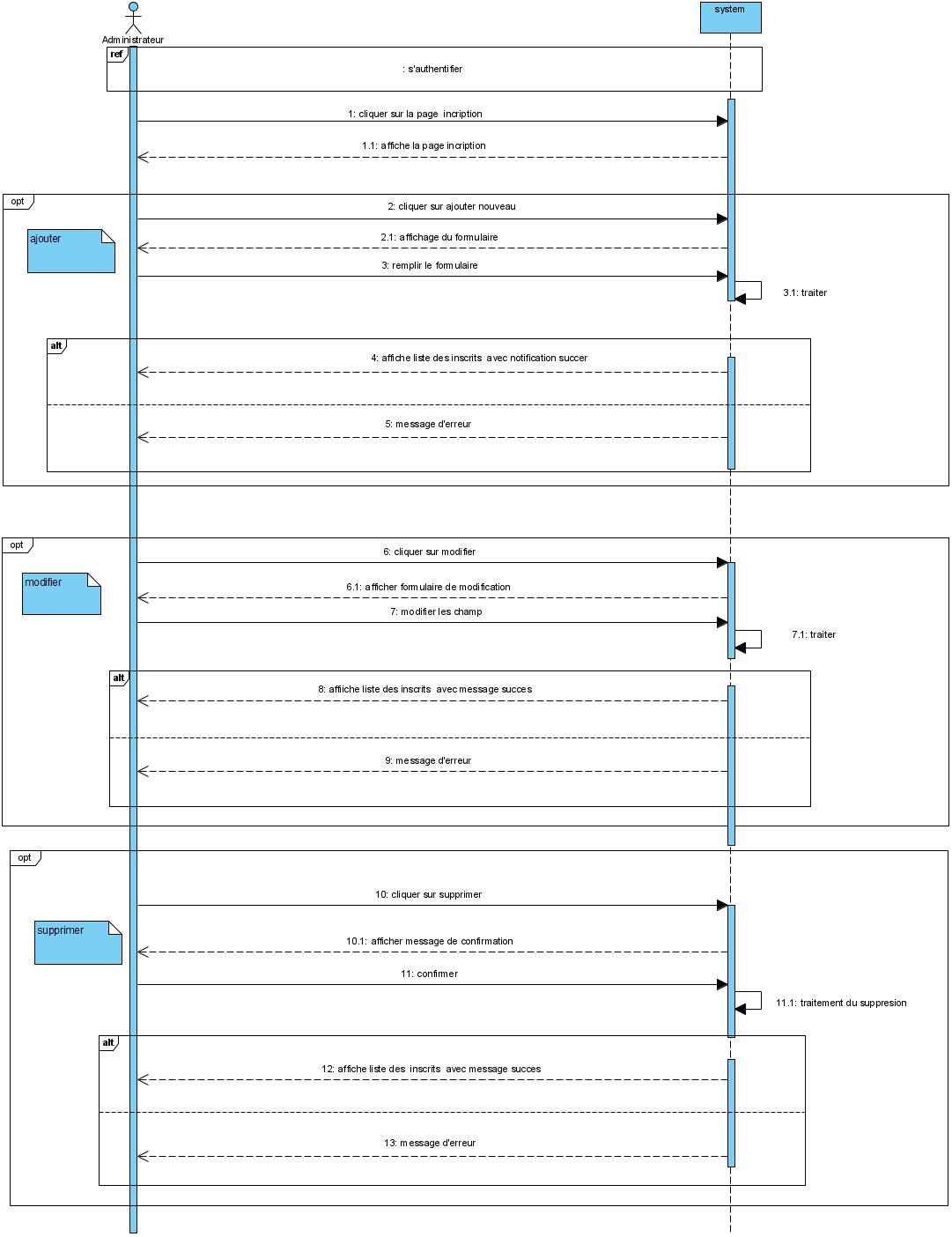


Figure 9 Diagramme de séquence système pour le cas d'utilisation Gérer inscription

**e. Cas d’utilisation : Gérer l’écolage**

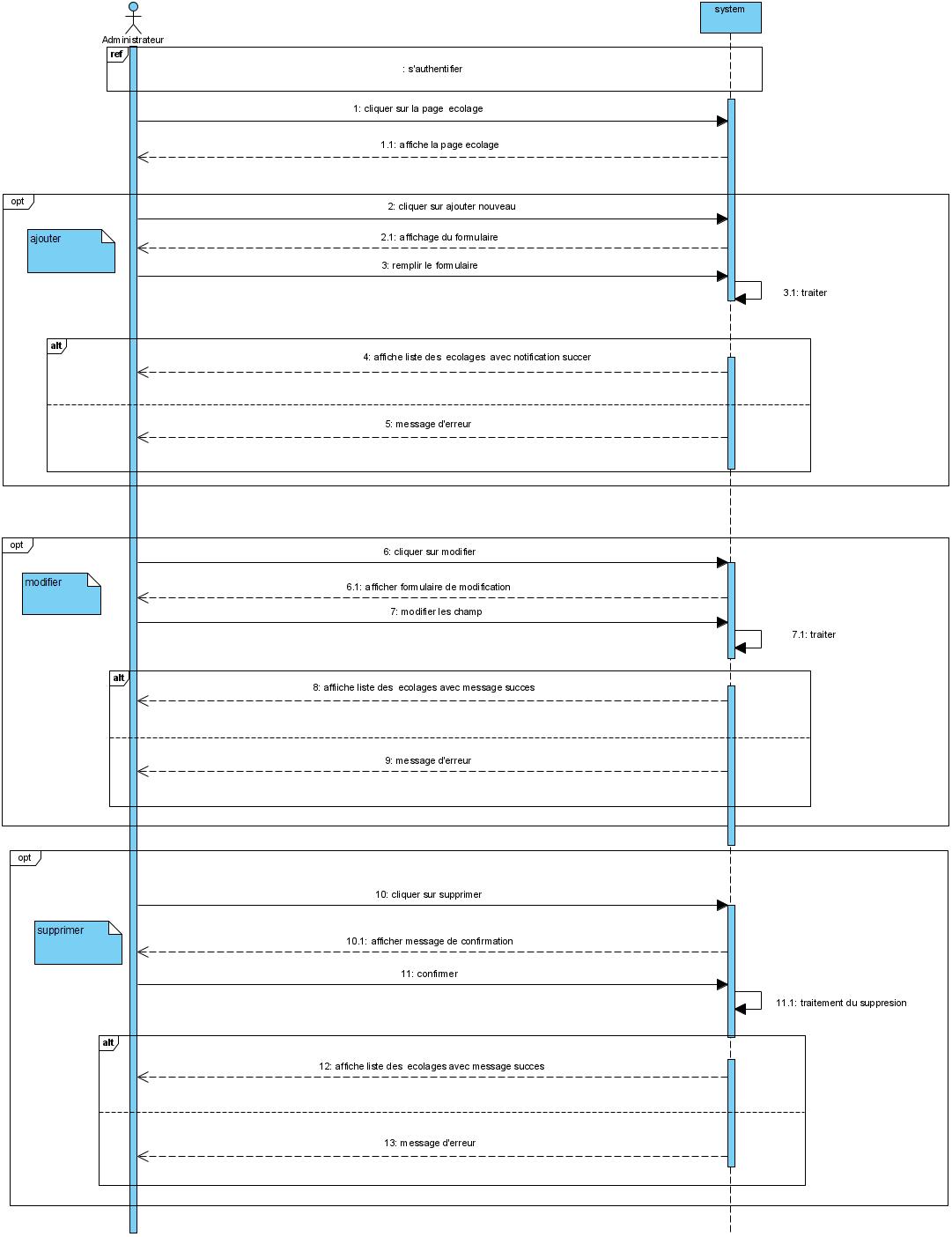
La figure 10 représente le diagramme de séquence du système pour le cas d'utilisation "Gérer écolage". Ce diagramme illustre les étapes à suivre pour qu'un utilisateur puisse crée un paiement d’une écolage.

Figure 10 Diagramme de séquence système pour le cas d’utilisation gérer l’écolage

**f. Cas d’utilisation : Suivi des paiements**

La figure 11 montre le diagramme de séquence du système pour le cas d'utilisation "suivi des paiements". Ce diagramme décrit comment les utilisateurs peut interagir avec le système pour suivre les payements existants.

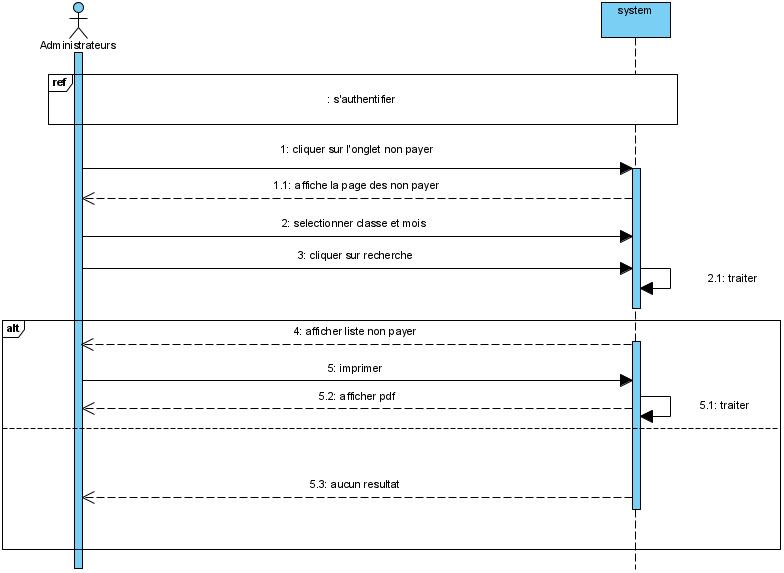


Figure 11. Diagramme de séquence système pour le cas d'utilisation Suivi des paiements

**g. Cas d’utilisation : Gérer les professeurs**

La figure 12 représente le diagramme de séquence système pour le cas d’utilisation

« **Gérer les professeurs** »

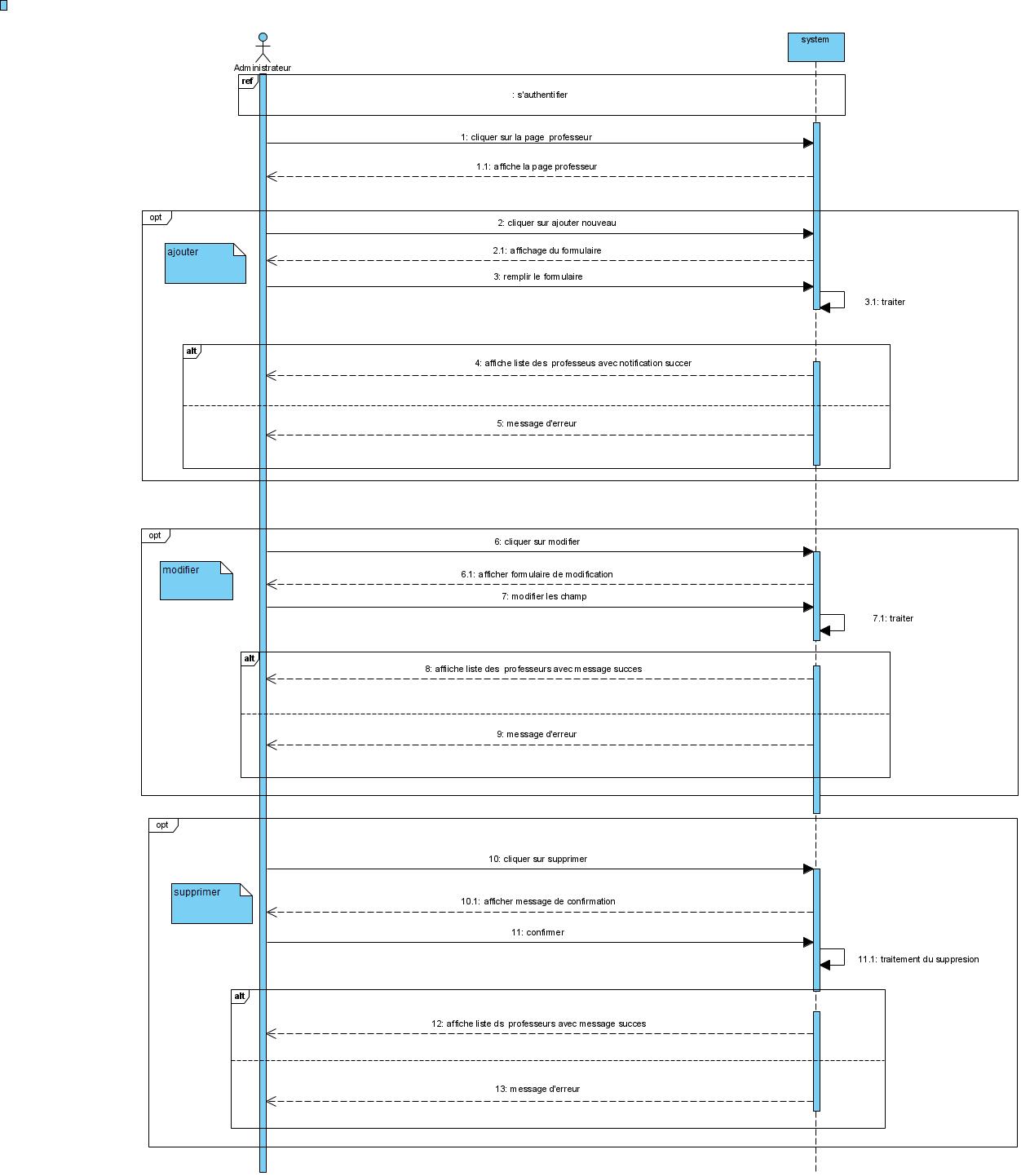


Figure 12 Diagramme de séquence système pour le cas d'utilisation Gérer les professeurs

**h. Cas d’utilisation :** **Gérer les utilisateurs**

La figure 13 montre le diagramme de séquence du système pour le cas d'utilisation "Gérer les utilisateur". Ce diagramme décrit comment les utilisateurs peut interagir avec le système pour modifier supprimer un utilisateur.

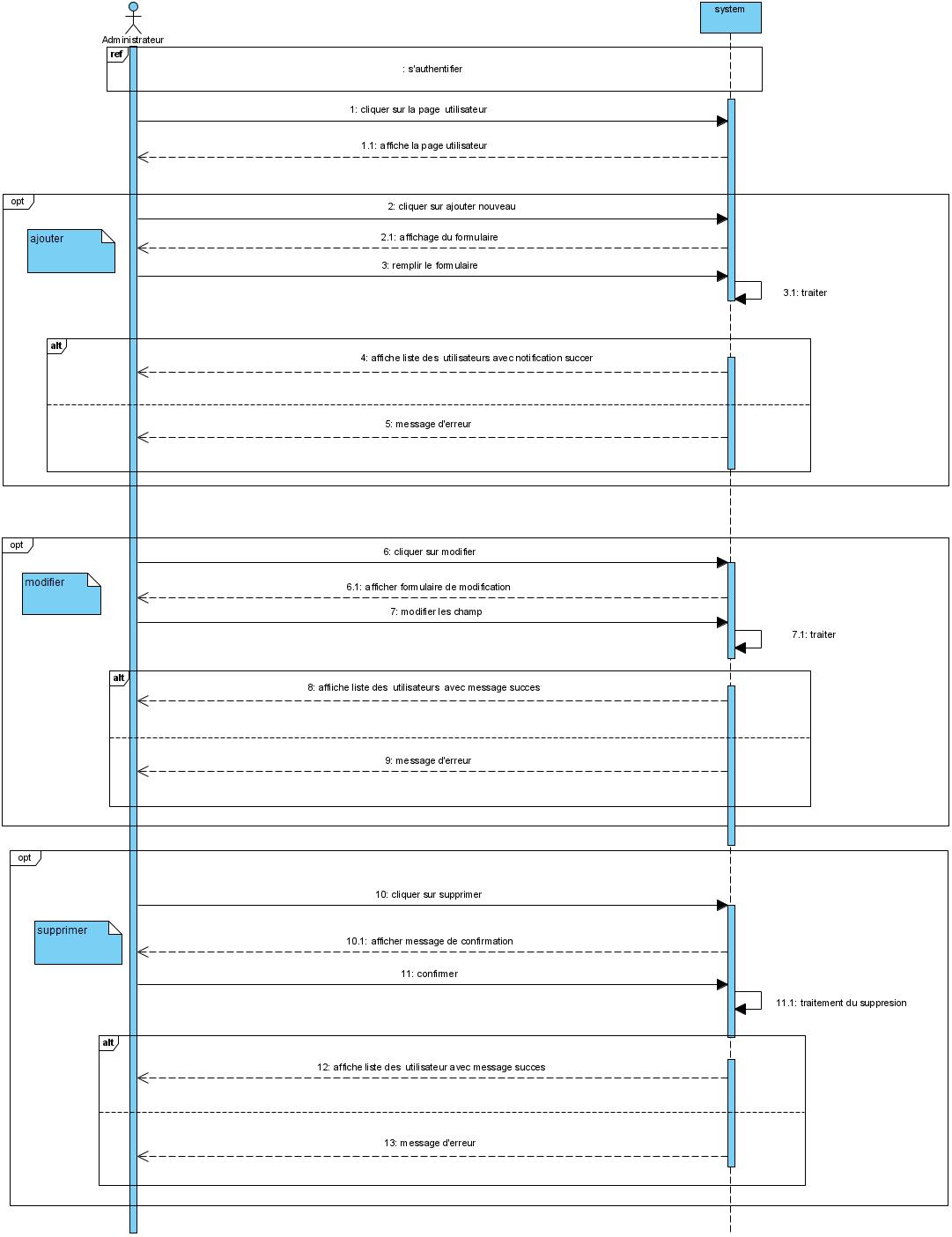


Figure 13 Diagramme de séquence système pour le cas d’utilisation « Gérer les utilisateurs »

## 5.5 Spécification des besoins techniques

Les besoins techniques sont directement liés aux tâches à accomplir, et il est essentiel qu'ils restent aussi transparents que possible pour les utilisateurs. Notre application doit être capable de prendre en charge les fonctionnalités techniques suivantes :

* **Fiabilité :** les données fournies par l'application doivent être fiables.
* **Convivialité de l'interface graphique :** l'application doit offrir une interface conviviale et simple, adaptée à tous les types d'utilisateurs, car elle constitue le premier contact de l'utilisateur avec l'application et détermine la découverte de ses fonctionnalités.
* **Solution ouverte et évolutive :** l'application doit permettre des améliorations grâce à l'ajout de modules supplémentaires, assurant ainsi la souplesse, l'évolutivité et l'ouverture de la solution.
* **La sécurité :** l’accès à l’application doit être protégé par un système d’authentification.
* **Robustesse :** l’application doit s’attendre à bien réagir lorsque l’on s’écarte des conditions normales d’utilisation

## 5.6 Modélisation du domaine

La modélisation du domaine implique une approche qui débute par la représentation des besoins a travers des cas d’utilisations, se rapprochant ainsi d’une analyse fonctionnelle classique. Cette démarche évolue ensuite vers une modélisation objet approfondie, mettant en lumière l’importance de développer le modèle de classe du domaine. Ce dernier a pour tâche de définir de manière précises les classes responsables de représenter les entités ou concepts spécifiques au domaine de l’application. En somme, l’objectif fondamental est de créer un modèle des objets du monde réel au sein du contexte particulier du domaine considéré.

La figure 14 représente la modélisation du domaine de notre projet

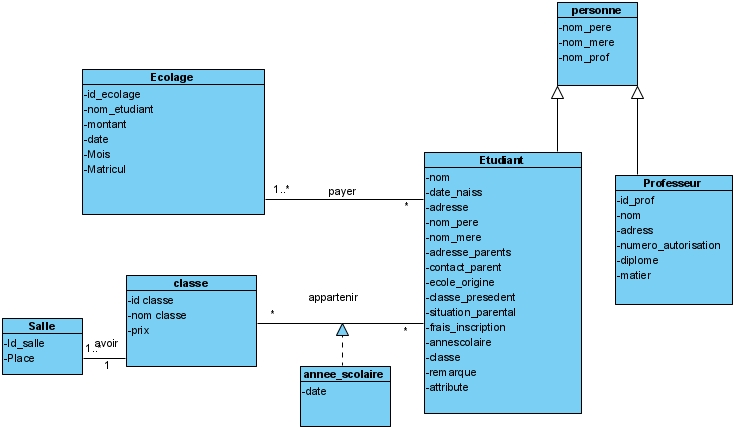


Figure 14 Modélisation du domaine

# Chapitre 6. Conception détaillé

Dans ce sixième chapitre, nous approfondirons la phase de conception en examinant plus en détails chaque cas d’utilisation à travers l’analyse de l’architecture du système, l’élaboration des diagrammes de séquences et la structuration du diagramme de classe.

### Architecture système

MVC, ou modèle-vue-contrôleur (model-view-controller en anglais), est ce qu’on appelle un modèle d’architecture logicielle. Ce type d’architecture, aujourd’hui très répandu dans les projets d’applications web, doit son succès notamment à sa forte maintenabilité et à sa facilité à être utilisé en travail collaboratif.

* **Modèle :** Le modèle, ou model, représente les données qui vont être utilisées dans l’application web. C’est ici que va être stockée la data, et tout ce qui permet de la modifier (getters, setters, etc.), que ça soit en local en en distant (base de données).
* **Vue :** La vue, ou view, c’est l’interface graphique de l’application. C’est via cet élément que vont se faire les interactions entre l’utilisateur et le code métier. Elle ne contient presque aucune logique (contrairement à l’architecture concurrente de MVC dont on parlera plus loin), son but est de construire, à partir de ce que renvoie le serveur, une interface et de l’afficher à l’utilisateur.
* **Contrôleur :** Le contrôleur, ou controller, est l’élément qui contient la logique métier. C’est ici que sont la plupart des algorithmes, calculs, etc.

Dans la figure 15, nous allons voir l’architecture MVC.

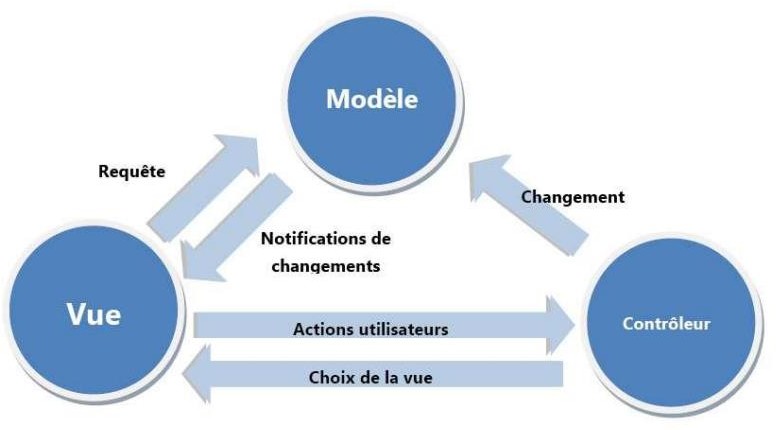


Figure 15 Architecture MVC

### 6.2 Diagramme de séquence de conception pour chaque cas d’utilisation

Les diagrammes de séquences de conceptions sont des représentations graphique détaillant les appels de méthodes et la coopération entre instances et méthodes dans le cade de conception d’un système. Ils offrent une vue en largeur du déroulement d’une opération en mettant en relief les objets impliqués, les méthodes des classes concernées ainsi que la source des paramètres.Ces diagrammes jouent un rôle essentiel offrant une compréhension globale de la mécanique d’ensemble et en clarifiant les responsabilités avant d’entrer dans les détails spécifiques. Ce diagramme de conception se révèle utile comme outil de conception, un moyen de documentation pour transmettre les informations de manière claire et de planification de la structure du fonctionnement du système.

Chaque cas d’utilisation est ainsi accompagné d’un diagramme de séquence de conception, offrant une représentation visuelle et détaillée des interactions entre les composants du système.

**A. Diagramme de séquence de conception cas d’utilisation « S’authentifier »**

La figure 16 représente le diagramme de séquence du cas d’utilisation s’authentifier.

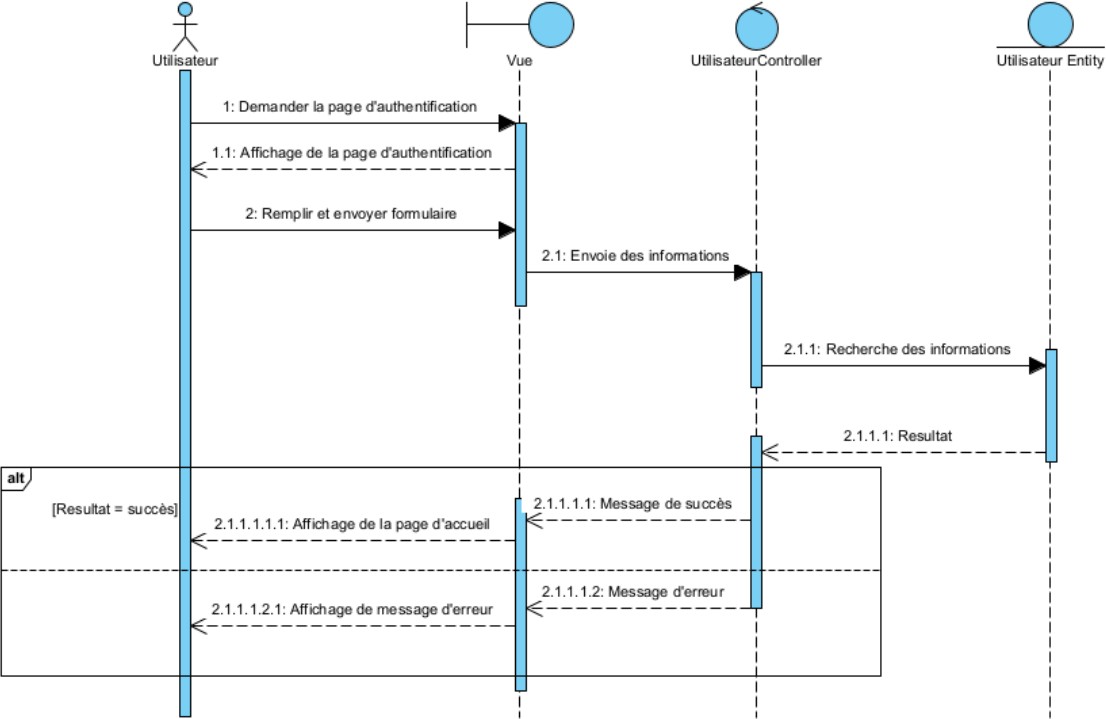


Figure 16 Diagramme de séquence de conception du cas d'utilisation S'authentifier

**B. Diagramme de séquence de conception cas d’utilisation « Gérer classes »**

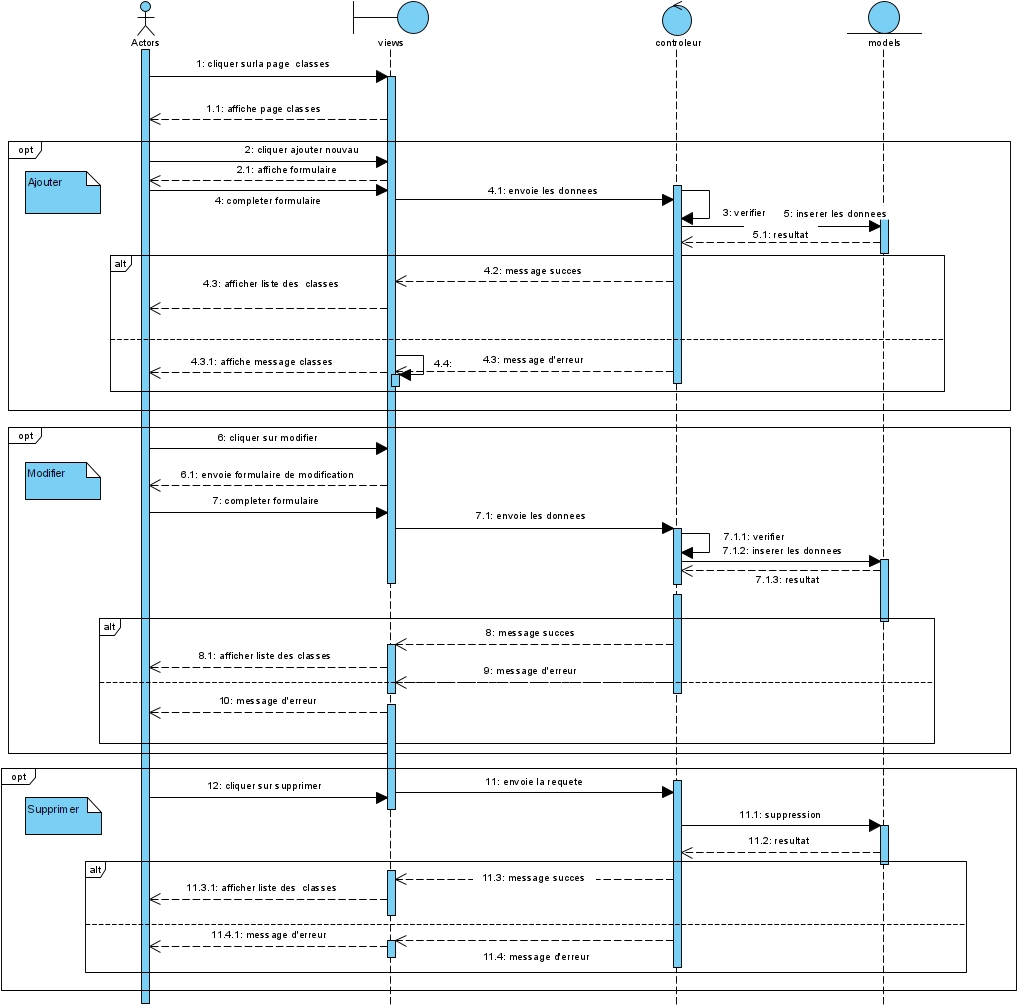
La figure 17 représente le diagramme de séquence du cas d’utilisation gérer classes

Figure 17 Diagramme de séquence du cas d’utilisation gérer classes

**C. Diagramme de séquence de conception cas d’utilisation « Gérer l’inscription »**

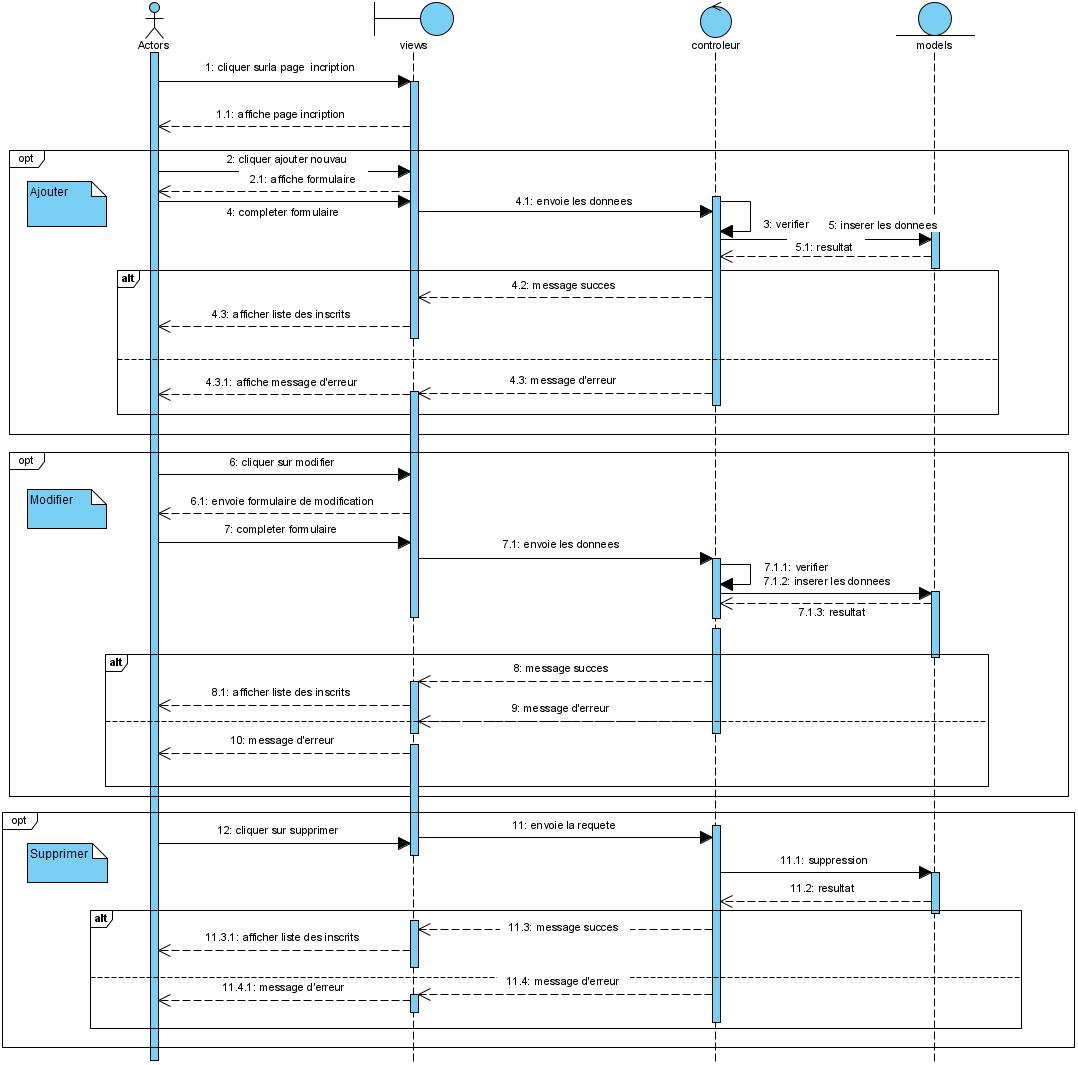
La figure 18 représente le diagramme de séquence du cas d’utilisation gérer inscription.

Figure 18 Diagramme de séquence de conception du cas gérer inscription

**D. Diagramme de séquence de conception cas d’utilisation « Gérer l’écolage »**

La figure 19 représente le diagramme de séquence du cas d’utilisation gérer écolage



Figure 19 Diagramme de séquence de conception du cas gérer écolage

**E. Diagramme de séquence de conception cas d’utilisation « Suivi des paiements »**

La figure 20 représente le diagramme de séquence du cas d’utilisation Suivi des paiements

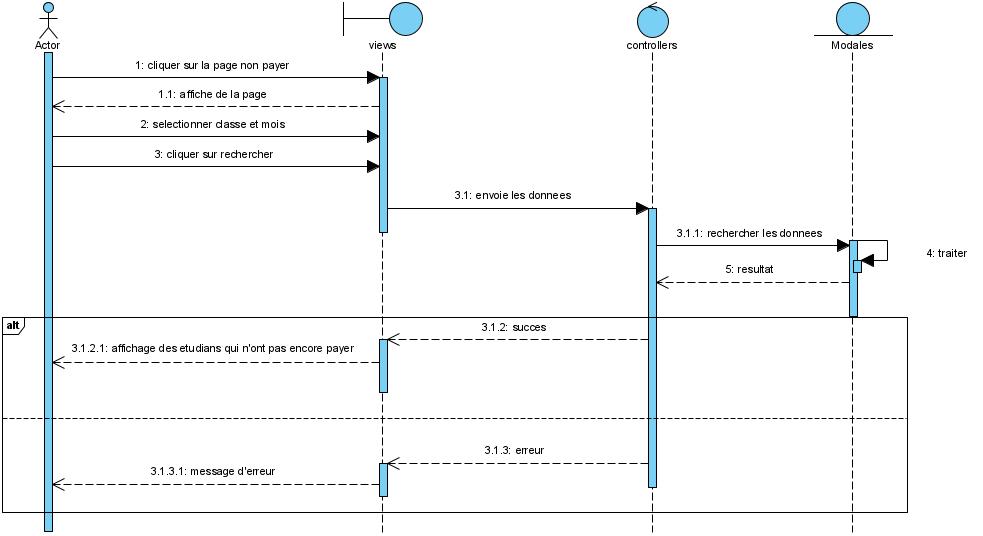


Figure 20 Diagramme de séquence de conception du cas Suivi des paiements

**F. Diagramme de séquence de conception cas d’utilisation « Gérer les professeurs »**

La figure 21 représente le diagramme de séquence du cas d’utilisation Gérer les professeurs

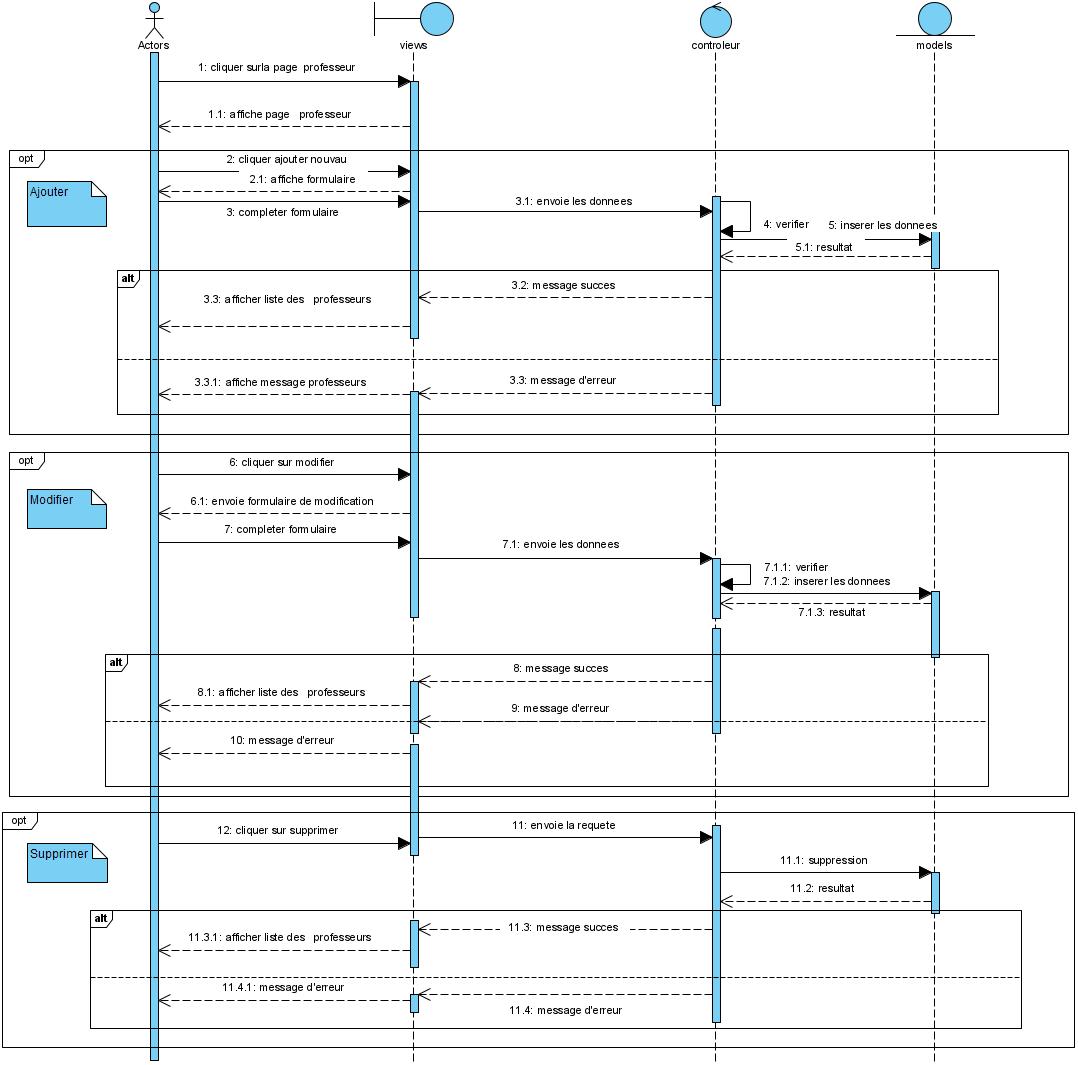


Figure 21 . Diagramme de séquence de conception du cas Gérer les professeurs

**G. Diagramme de séquence de conception pour le cas d’utilisation « Gérer les utilisateurs »**

La figure 22 représente le diagramme de séquence du cas d’utilisation Gérer les utilisateurs

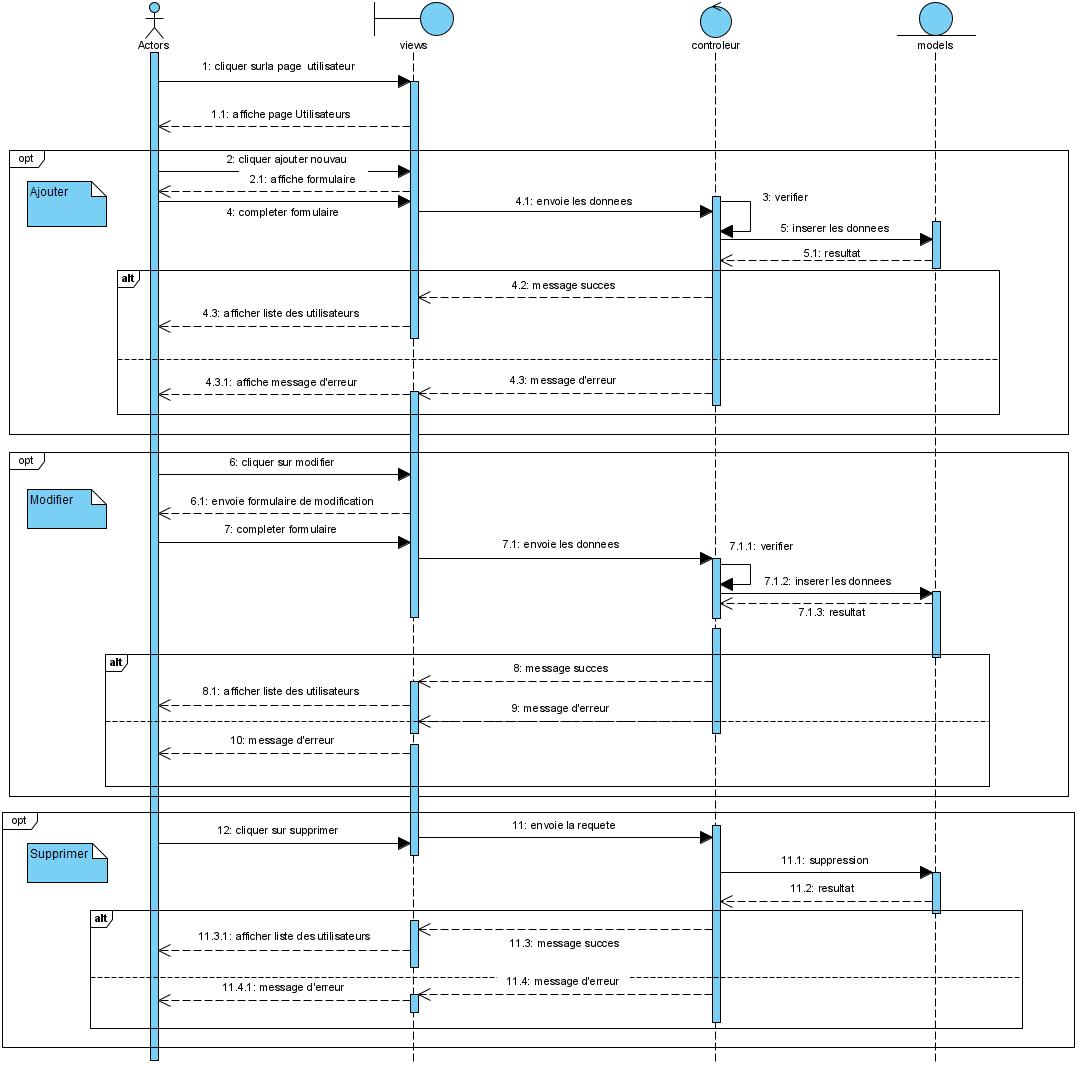


Figure 22 Diagramme de séquence de conception du cas gérer les utilisateurs

### 6.3 Diagramme de classe de conception pour chaque cas d’utilisation

La description du diagramme de classe est fondée sur le concept d’objet, le concept de classe comprenant les attributs et les opérations, les différents types d’associations entre les classes.

* + - Objet : Un objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer à priori les valeurs de ses propriétés. Un objet est une instance d’une classe.
    - Attributs : Un attribut est une donnée élémentaire d’une classe servant à caractériser les classes et les relations.
    - Classe : Une classe décrit un groupe d’objets ayant les mêmes attributs, une même opération, et un même domaine de définition. La classe représente l’abstraction de ses objets.
    - Opération : Une opération est une fonction applicable aux objets d’une classe. Elle permet de décrire le comportement d’un objet.
    - Association : Une association entre classes permet de relier une classe à plusieurs autres classes. Chaque association peut être identifiée par son nom.
    - Multiplicité : La multiplicité indique un domaine de valeurs pour exprimer le nombre minimum et maximum d’objets d’une classe qui peuvent être reliés a des objets d’une autre classe.

**A. Diagramme de classe de conception pour le cas d’utilisation « S’authentifier »**

La figure 23 nous montre le diagramme de classe de conception pour le cas d’utilisation S’authentifier.

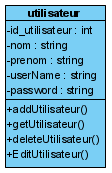


Figure 23 Diagramme de classe de conception du cas d'utilisation S'authentifier

**B. Diagramme de classe de conception pour le cas d’utilisation « Gérer classes »**

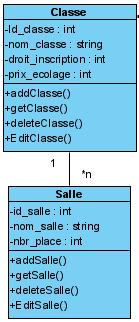
La figure 24 nous montre le diagramme de classe de conception pour le cas d’utilisation Gerer classes

Figure 24 Diagramme de classe de conception pour le cas d’utilisation

**C. Diagramme de classe de conception pour le cas d’utilisation « Gérer l’inscription »**

La figure 25 nous montre le diagramme de classe de conception pour le cas d’utilisation gérer inscription.

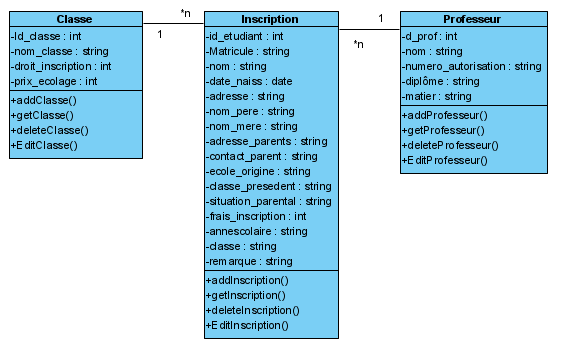


Figure 25 Diagramme de classe se de conception du cas d'utilisation gérer inscription

**D. Diagramme de classe de conception pour le cas d’utilisation « Gérer l’écolage » et « Suivi des paiements »**

La figure 26 nous montre le diagramme de classe de conception pour le cas d’utilisation gérer écolage.

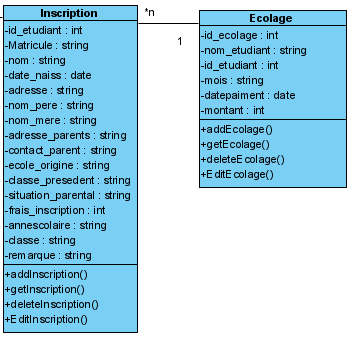


Figure 26 Diagramme de classe se de conception du cas d'utilisation gérer écolage

**E. Diagramme de classe de conception pour le cas d’utilisation « Gérer les professeurs »**

La figure 27 nous montre le diagramme de classe de conception pour le cas d’utilisations gérer les professeurs.

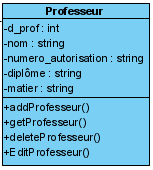


Figure 27 Diagramme de classe se de conception du cas d'utilisation gérer professeurs

### 6.2 Diagramme de classe globale

Cette section fournit une vue d'ensemble du diagramme de classe de conception global, qui met en évidence les principales entités du système de gestion de rappels de tâches. En détaillant leurs attributs, opérations et relations, ce diagramme illustre de manière claire et structurée comment les différentes classes interagissent et s'associent pour former l'architecture intégrée du système, facilitant ainsi la compréhension de son fonctionnement global.

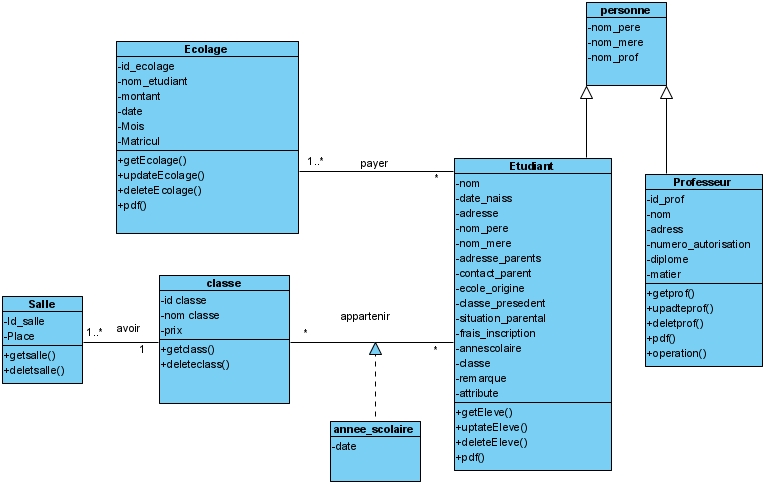


Figure 28 Diagramme de classe globale

### 6.3 Diagramme de paquetage

Le diagramme de paquetage est un outil essentiel pour organiser et structurer les éléments d’un système en groupes logiques, appelés paquets. Il permet de visualiser la dépendance entre ces paquets, facilitant ainsi la gestion et la modularité du projet. En structurant les composants du système de cette manière, le diagramme de paquetage aide les équipes de développements à comprendre les relations entre les différents modules et à planifier efficacement l’architecture du logiciel.

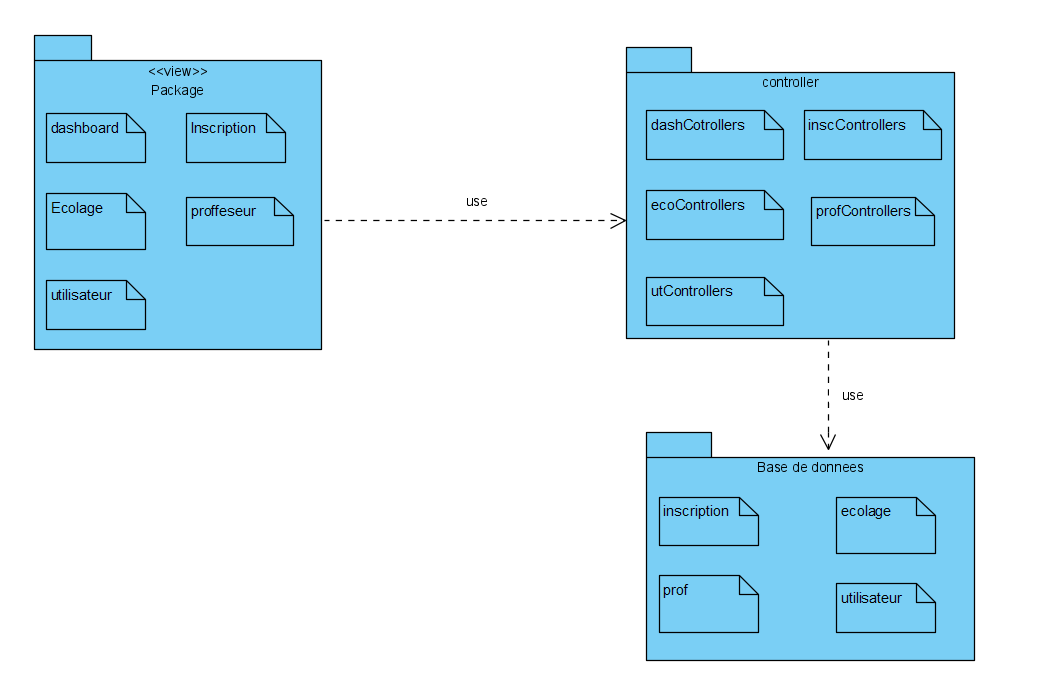


Figure 29 Diagramme de paquetage

### 6.4 Diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montrent la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds.

La figure 30 illustre le diagramme de déploiement de l’application.

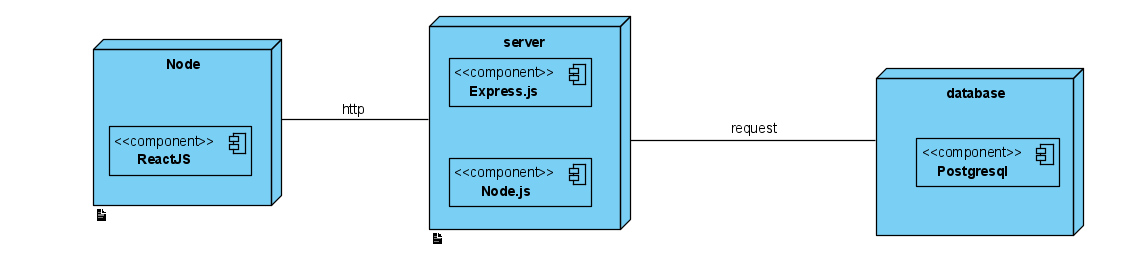


Figure 30 Diagramme de déploiement

# Partie III : Réalisation

# Chapitre 7. Mise en place de l’environnement de développement

Dans ce septième chapitre, nous observerons un aperçu des conditions de travails réelles. Il sera divisé en 2 sections. Tout d'abord, nous allons décrire les logiciels et l'environnement matériel pour atteindre la phase de développement. Ensuite, nous présenterons l'architecture sur laquelle l’application a été développée.

### 7. 1 Installation et configuration des outils

#### 7.1.2 Installation de Visual studio Code

Visual Studio Code est un éditeur de code extensible et gratuit développé par Microsoft. Il est disponible pour les systèmes d’exploitation Windows, Linux et MacOs, et il est utilisé pour le développement informatique d’applications logicielles, de sites web et de services. Les fonctionnalités notables comprennent le support du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, l’utilisation de snippets, la ré-factorisation du code, ainsi que l’intégration de Git. On peut le télécharger dans leur site web officiel <https://code.visualstudio.com/download>

Après le téléchargement, exécuter le setup d’installation en tant qu’administrateur et nous aurons la fenêtre d’installation comme dans la figure suivante.

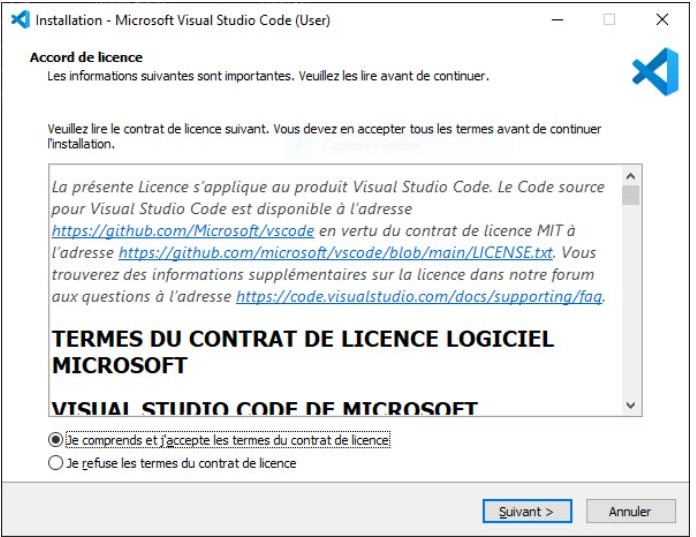


Figure 31 Contrat d'utilisation de Visual Studio Code

Ensuite, nous enchaînons avec les autres étapes.

Après avoir cliqué sur le bouton « Installer », nous aurons alors la barre de progression de

L’installation et après il suffit de cliquer sur le bouton « terminer » pour terminer l’installation

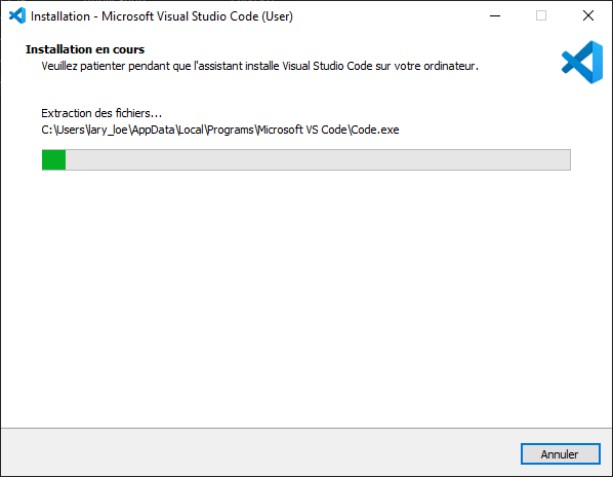


Figure 32 Progression de l'installation de Visual Studio Code

### 7.1.3 Installation de Visual Paradigm

Visual Paradigm est une suite de modélisation et de conception de logiciels qui offre une gamme complète d'outils pour les professionnels du développement logiciel, les analystes métier, et les architectes. On peut le télécharger dans leur site officiel [https://www.visual-](https://www.visual-paradigm.com/) [paradigm.com](https://www.visual-paradigm.com/)

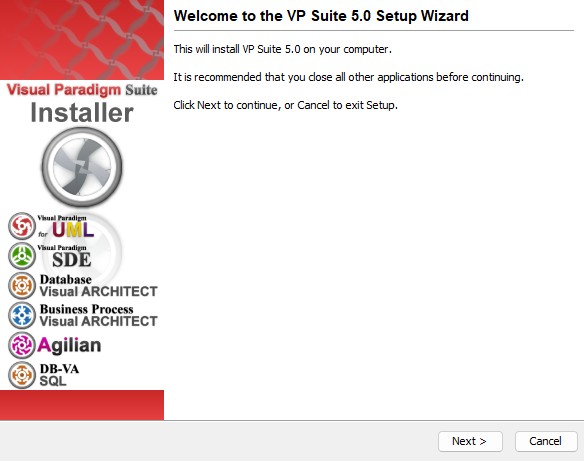
Pour commencer l’installation, nous aurons cette fenêtre d’installation suivant

Figure 33 Fenêtre d'installation de visual paradigm



Figure 34 . Etapes d'installation de visual paradigm

Après cela, il suffit de personnaliser selon les besoins de l’utilisateur.

### Installation de Node.js

Node.js propose deux versions différentes que nous pouvons télécharger : la version LTS et la version actuelle

La première est le support à long terme (Long Term Support ou LTS), qui indique la version qui est sur le marché depuis un certain temps et qui est fournie avec tout le support obligatoire. Cette version LTS est recommandée à la plupart des utilisateurs en raison de sa durabilité et de son cycle de support de 18 mois. Comme il s’agit d’une version stable, son utilisation pour produire des backends peut nous aider à obtenir un résultat robuste.

La version actuelle (Current) indique la dernière version publiée de Node avec les caractéristiques les plus récemment ajoutées et mises à jour. Mais cette version a moins de support derrière elle (environ huit mois) et une possible exposition aux bogues. Par conséquent, les experts suggèrent d’utiliser cette version uniquement pour le développement forntend. [E]

Pour la réalisation du projet, nous avons installé la version stable qui est la version LTS, qui est disponible dans le lien suivant : <http://nodejs.org/en/download/>

Après avoir télécharger ce dernier, ouvrir le fichier pour lancer l’installation.

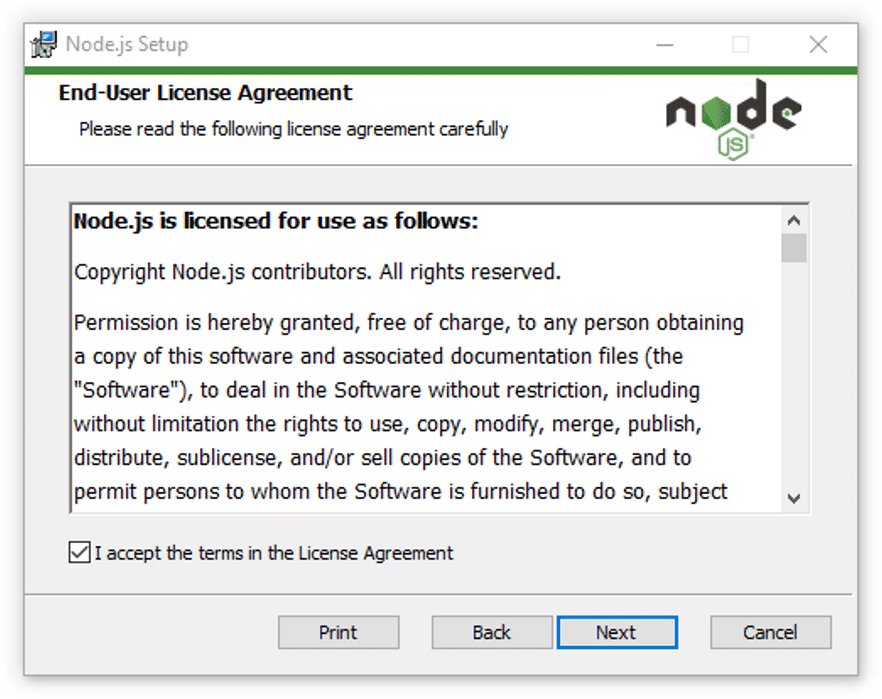


Figure 35 Contrat de licence de Node.js

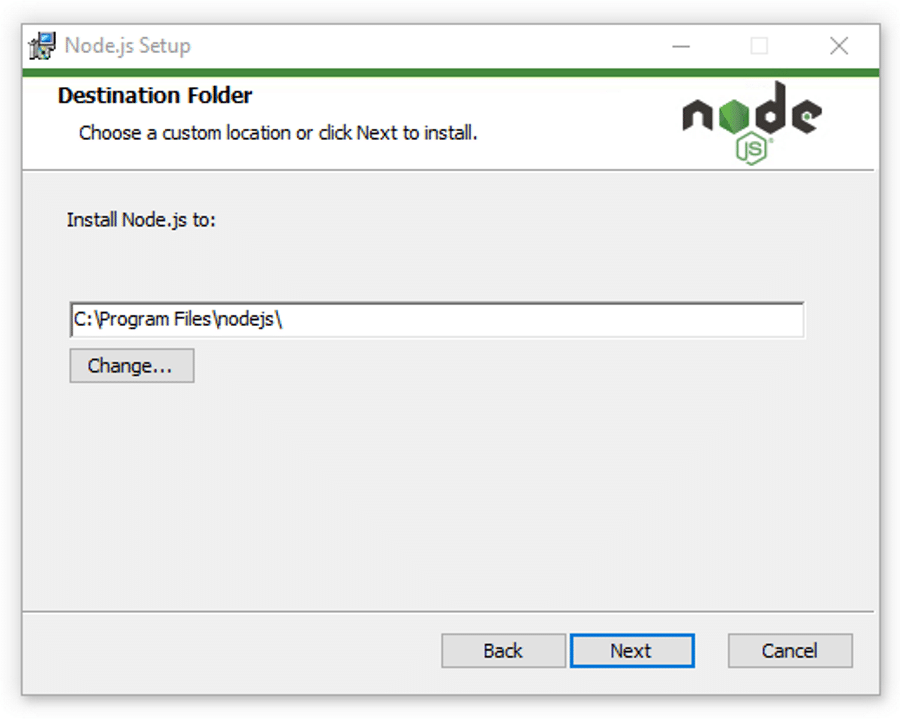


Figure 36 Sélection du répertoire d'installation de Node.js

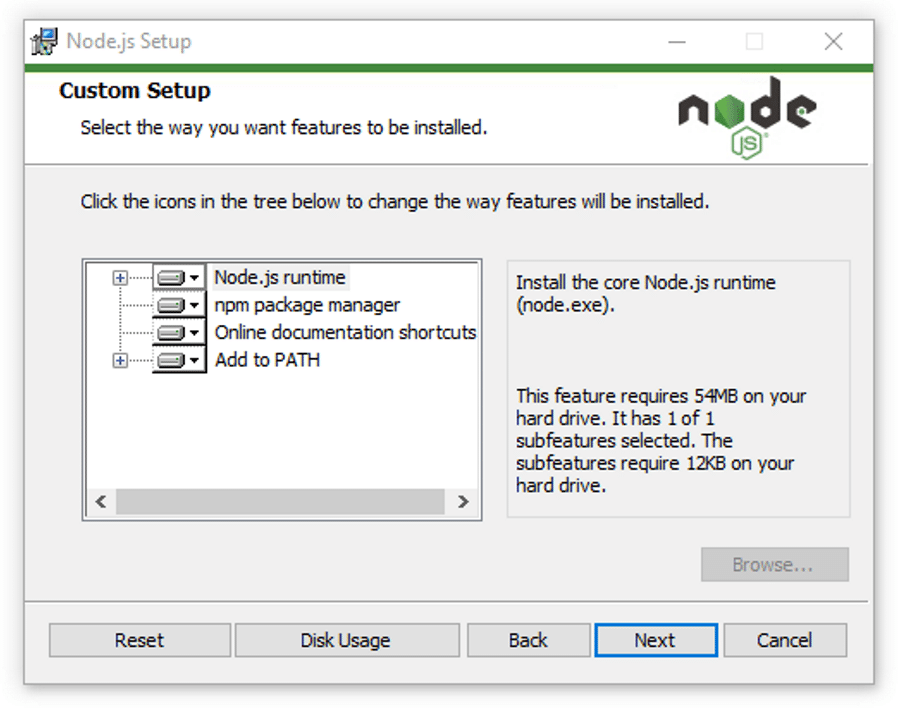


Figure 37 Option "Configuration personnalisée" dans l'installateur de Node.js

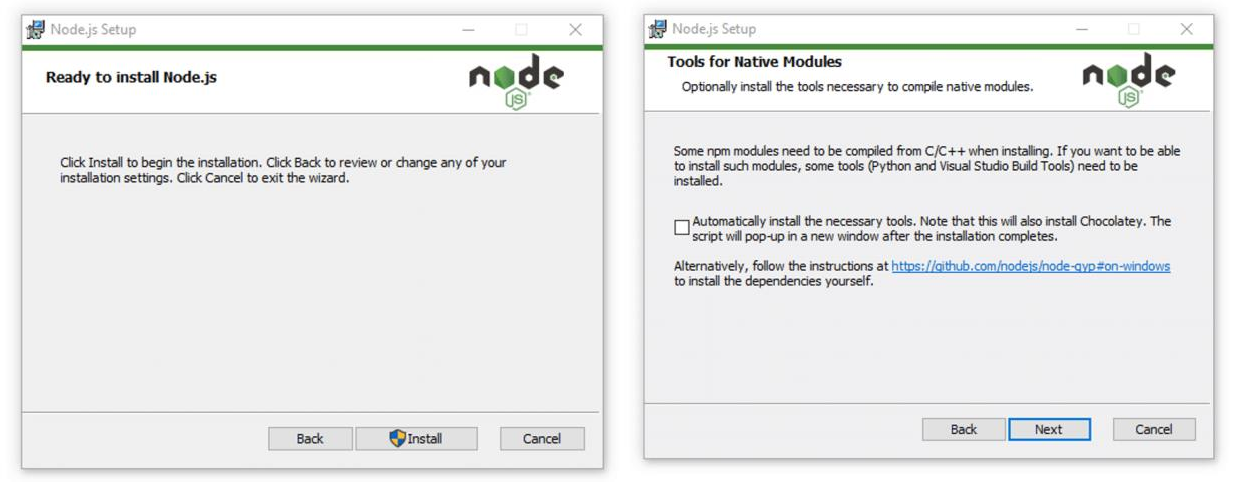


Figure 38 Finalisation de l'installation de Node.js

### 7.1.4 Installation de PostgreSQL

PostgreSQL c’est notre serveur pour la base de données voici les étapes pour l’installer

Apres avoir télécharger PostgreSQL on l’installe

La figure 39 représente la page d’installions de PostgreSQL

Figure 39 Page d’installation PostgreSQL

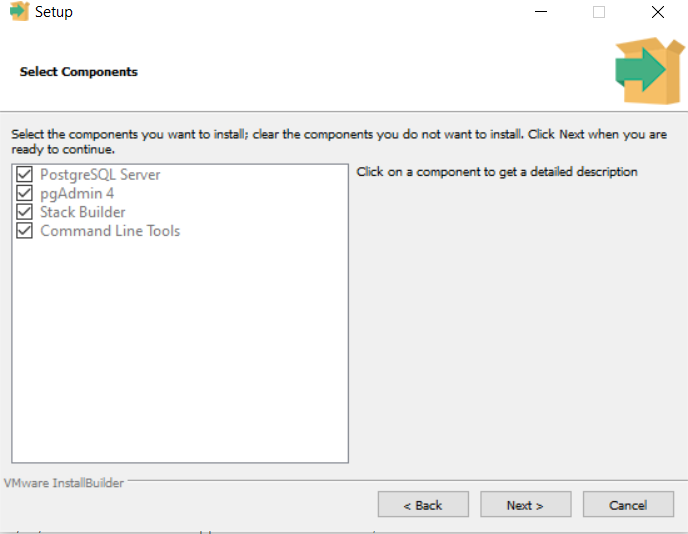
Apres avoir cliquer sur Next une nouvelle page s’affiche, un page pour la configuration pour l’installer correctement on doit cocher toutes les options pour éviter le bug a l’avenir

Figure 40 Configuration PostgreSQL

## 7.2 Architecture de l’applications

Nous avons adopté une architecture 3-tiers. Cette architecture est à trois niveaux : client, serveur d’application et serveur de données. Le client est l’ordinateur demandeur de ressources équipe d’une interface utilisateur. Le serveur d’application ou serveur web se charge de fournir la ressource mais faisant d’appel à un autre serveur. Le serveur de données fournissant au serveur d’application les données fournissant au serveur d’application les données requises.

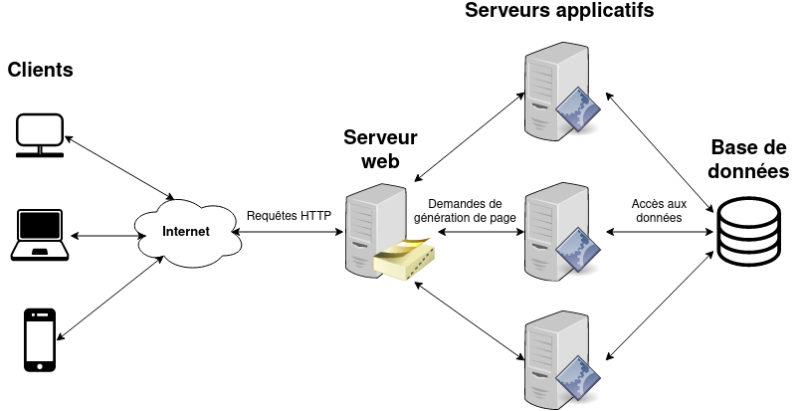


Figure 41 Architecture 3-tiers

# Chapitre 8. Développement de l’application

Dans ce huitième chapitre, nous plongerons dans le cœur du projet en explorant le processus de développement. Nous aborderons création de la base de données, mettrons en lumière les codages nécessaires et quelques captures d’écran de l’application pour une meilleure immersion visuelle.

### Création de la base de données

Pour créer notre base de données, nous avons initiés directement les schémas dans le backend Node.js. De plus, nous avons utilisé postgressql pour avoir une interface graphique de la base en question.

La figure suivante représente le schéma de la base de données de notre projet englobé dans un dossier models dont chaque fichiers javascript dans ce dossier représente chaque classe.

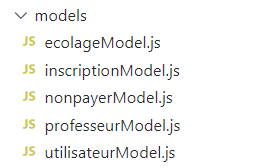


Figure 42 Architecture globale de l'application

### Codage de l’application

On peut voir dans cette partie, quelques captures des lignes de codes et leurs descriptions.

Mais avant d’entamer tout ça, il est crucial de présenter la structure de notre application. Dans notre projet, nous avons deux (02) dossiers parents dont ‘Client’ qui englobe tous les codes pour le côté client, et ‘Server’ pour le côté serveur.

La figure 43 représente les dossier frontend Reactjs

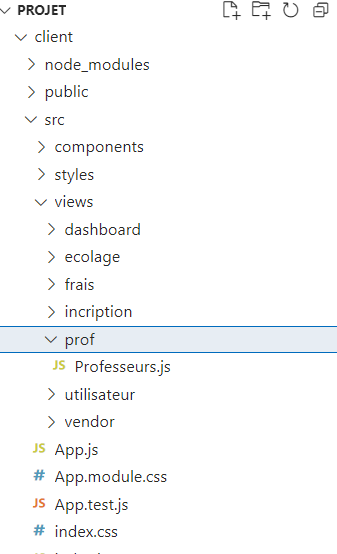


Figure 43 Dossier frontend Reactjs

La figure 44 présente un extrait de code notre projet depuis le backend, c’est ce code qui permet à notre application de communiquer avec la base de donner nous avons utilisé une bibliothèque de nodejs nommer sequelize pour cela

Figure 44 Modal Utilisateurs

La figure 45 illustre les importations nécessaires pour l’exécution de notre projet dans le frontend ce code est souscrite dans le page utilisateur frontend de notre projet

Figure 45 Frontend Utilisateurs

La figure 46 représente les route pour l’ajout le modification et la suppression d’un utilisateur pour la page utilisateur dans le backend

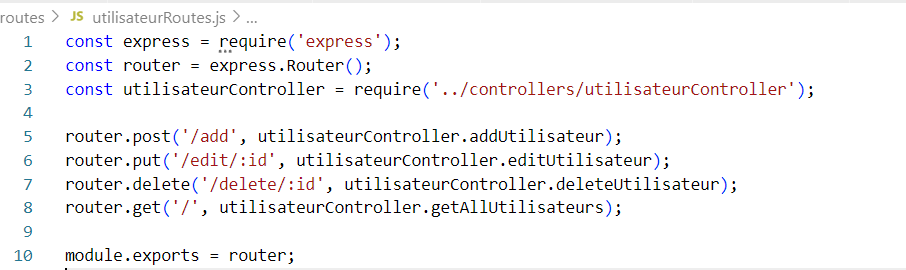


Figure 46 Route pour la page Utilisateurs

La figure 47 représente un extrait de code dans notre frontend, code des routes pour notre frontend avec car dans reactJS il y a des routes seulement pour les frontends

 Figure 47 Routes reactJS

### Présentation de l’application

Avant tout, un login se présente dès l’ouverture de l’application sur le navigateur. Puis c’est après que l’on pourra accéder à sa manipulation. Les différentes fonctionnalités de notre application sont présentées par quelques captures d’écran

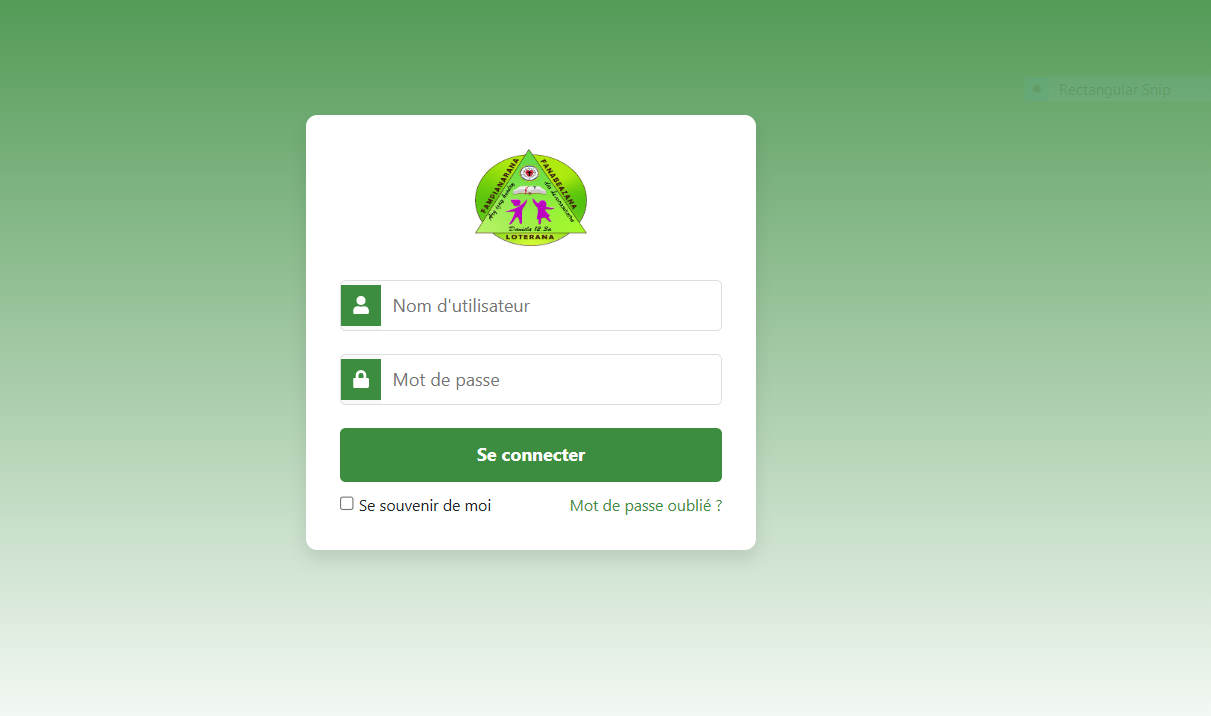


Figure 48 Page d’authentification

La figure 49 illustre la page pour ajouter une nouvelle inscription

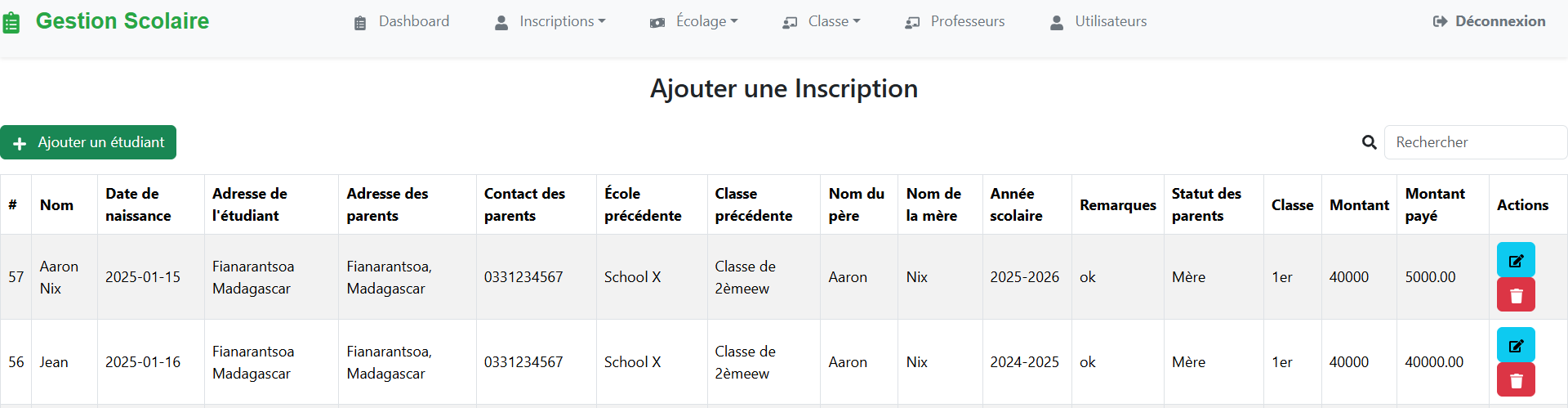


Figure 49 Page d’ajout d’une inscription

La figure 50 représente la page pour suivre les paiements pour les écolages.

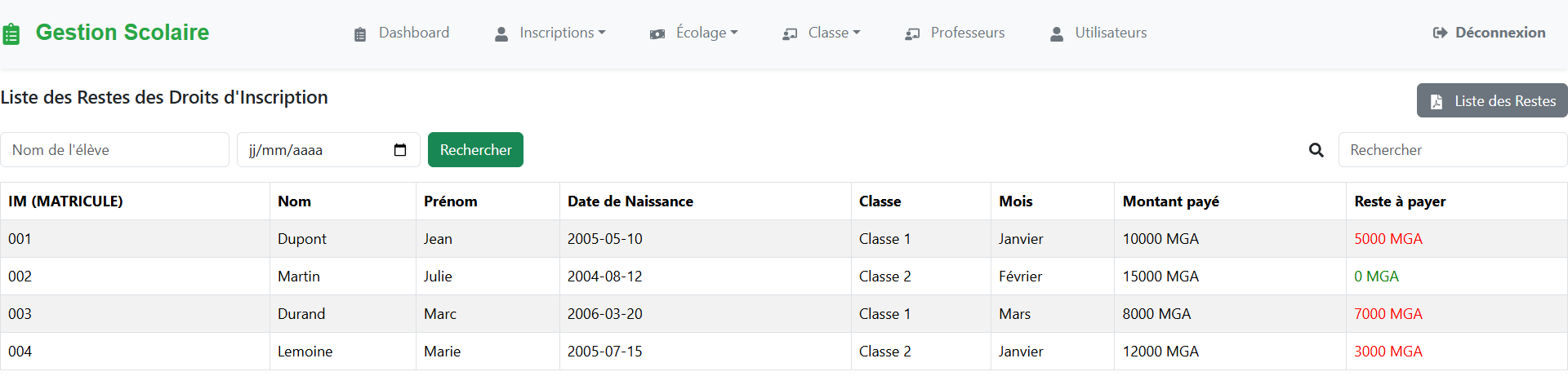


Figure 50 Page suivie des paiements

La figure 51 affiche la page pour ajouter un nouvel utilisateur.



Figure 51 page utilisateur

La figure 52 illustre la page d’ajout des prix pour chaque classe par type soit Ecolage soit Inscriptions.

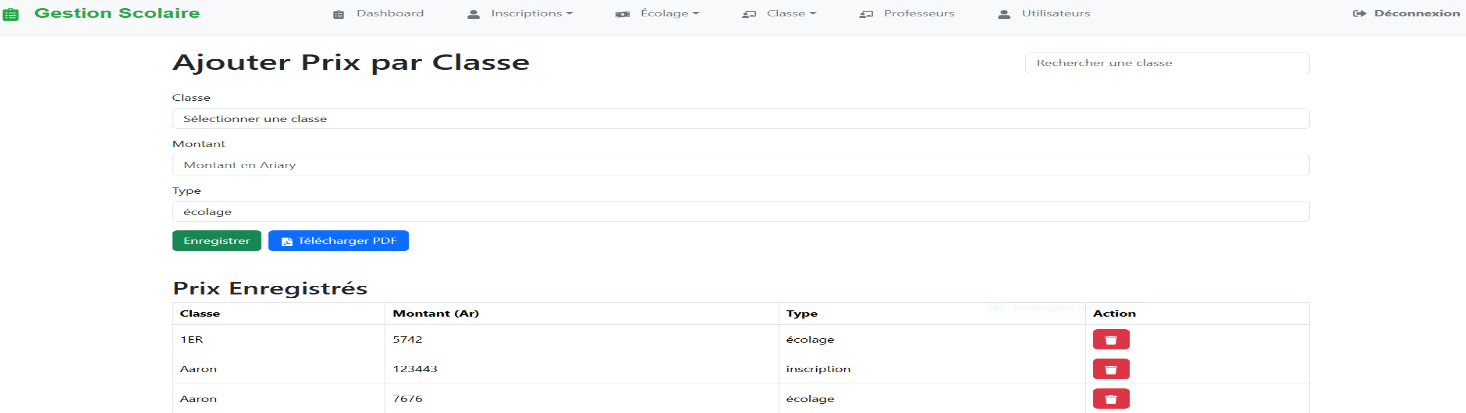


Figure 52 Ajouter prix

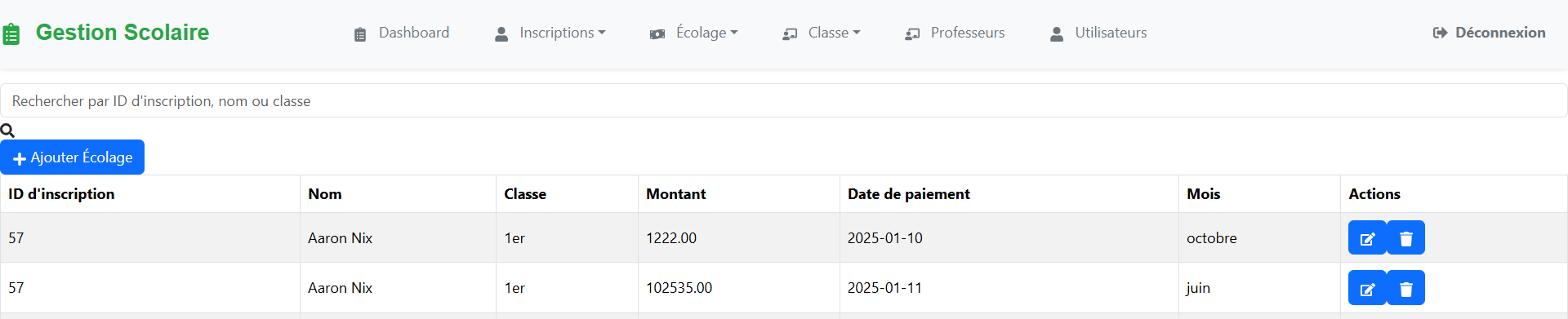
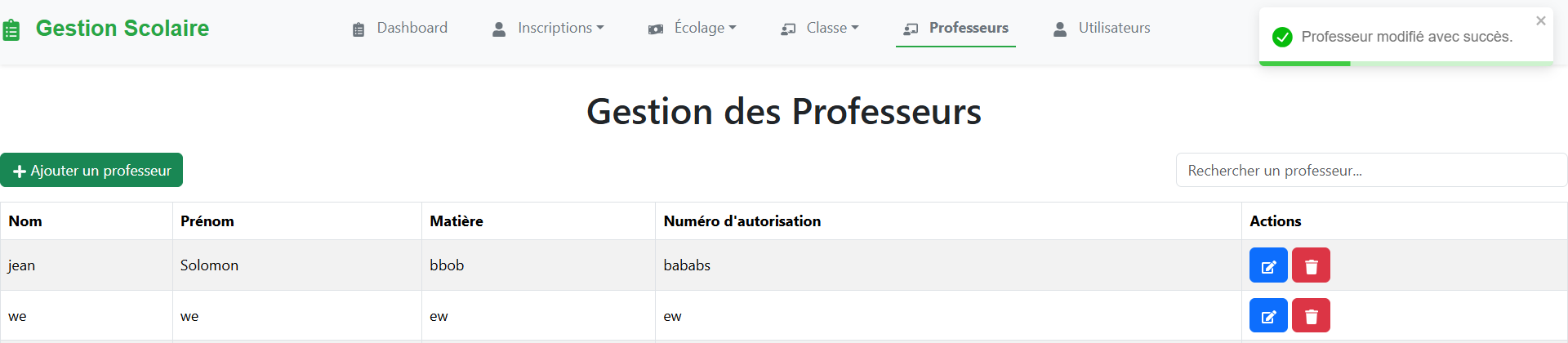
La figure 53 illustre la page d’ajout d’un nouvel écolage.

Figure 53 Ajouter Ecolage

La figure 54 représente la page d’ajout d’une nouvelle classe avec la salle correspondante .



Figure 54 Ajouter Classe



La figure 55 montre la page d’ajout d’un nouveau professeur dans l’établissement.

Figure 55 Ajouter Professeur

# Conclusion

Ce stage a mis en lumière les lacunes du système actuel de gestion des inscriptions et des écolages, principalement caractérisé par une absence de centralisation et une gestion manuelle inefficace. Ces problématiques, telles que les retards dans le suivi des paiements ou la difficulté à retrouver rapidement des informations administratives, entravaient le bon fonctionnement du lycée Loterana Rakoto Andrianarijaona.

Pour répondre à ces défis, nous avons conçu et développé une nouvelle application web, parfaitement adaptée aux besoins spécifiques des utilisateurs. Cette solution propose une gestion centralisée des inscriptions et des écolages, tout en offrant une interface conviviale et intuitive. Elle intègre des fonctionnalités clés telles que la facturation des paiements, la génération de rapports en PDF, et un suivi détaillé des opérations administratives.

Afin de garantir la robustesse et l'efficacité de cette solution, nous avons utilisé des outils modernes et des technologies adaptées : Méthode de conception 2TUP , ReactJS pour le frontend, Express.js pour le backend, et PostgreSQL pour la gestion des données. La méthodologie adoptée, basée sur une analyse approfondie des besoins et une conception orientée utilisateur, a permis de structurer le projet en étapes claires, de l’analyse à la réalisation.

Les résultats obtenus sont conformes aux attentes : une application fiable et performante, répondant aux besoins de centralisation et de simplification des tâches administratives. De plus, ce projet ouvre des perspectives intéressantes pour l’avenir, notamment l’ajout de fonctionnalités avancées telles que l’analyse de données pour des rapports plus détaillés, l’automatisation de certaines tâches récurrentes, et une optimisation continue des performances. L’extension de l’application pour d’autres plateformes est également envisageable.

Enfin, cette expérience a été l’occasion d’acquérir des compétences techniques et professionnelles précieuses, renforçant notre maîtrise des outils modernes de développement et notre capacité à gérer des projets de manière efficace. Elle marque une étape clé dans notre formation et nourrit notre ambition de contribuer au développement de solutions numériques innovantes.

# Références bibliographiques

1. Joseph Gabay et David Gabay, UML2 analyse et conception, Edition Eyrolles, Paris 2008, 242 pages.
2. Pascal Roques et Frank Vallée, 4eme édition, UML2 en action, Paris, EYROLLES, 394p
3. Thibaut Courouble, React vs Vue : Lequel choisir ? Edition Eyrolles, Paris 2023, 240 pages.

# Références webographies

[4] : <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-node-js/>, consulter le 20 septembre 2024

[5] : <https://easypartner.fr/blog/vue-vs-react-quel-est-le-meilleur-framework/> , consulter le 20 septembre 2024

[6] : <https://easypartner.fr/blog/vue-vs-react-quel-est-le-meilleur-framework/> , consulter le 20 septembre 2024

[7] : [https://www.techno-science.net/definition/670.html](https://www.techno-science.net/definition/670.html%20) , consulter le 26 septembre 2024

[8] :[https://kinsta.com/blog/how-to-install-node-js/](https://kinsta.com/blog/how-to-install-node-js/%20) , consulter le 28 septembre 2024

[9] : [https://fr.wikipedia.org/wiki/PostgresSql](https://fr.wikipedia.org/wiki/PostgresSql%20) , consulter le 01 novembre 2024

[10]:<https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-> [mean-stack](https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack) , consulter le 01 novembre 2024

# Glossaire

**Ecolage** : Montant payer par les eleves pour leurs scolarite qui se varie par classes

**Suivie scolaire :** Retard pour les frais de scolarité.

**Carte d'écolage individuelle :** Chaque élève se voit attribuer une carte d'écolage individuelle au début de l'année scolaire. Cette carte contient des informations personnelles telles que le nom de l'élève, son identifiant scolaire, sa classe, et d'autres détails pertinents. Elle sert également de moyen de suivi des paiements de frais de scolarité.

**Inscription :** Le processus administratif par lequel les élèves s'inscrivent à l'école pour une nouvelle année scolaire. Cela peut impliquer la fourniture de documents, le paiement de frais de scolarité, et la mise à jour des informations personnelles.

**Règlement intérieur :** Un ensemble de règles et de normes établies par l'établissement scolaire pour assurer le bon fonctionnement, la sécurité, et le respect au sein de l'école. Les élèves sont tenus de suivre ces règles.

**Professeur** : Membre du personnel enseignat.

**Classe** : une salle utilisée pour les études des élevés.

**Année scolaire :** Période pendant laquelle les cours sont dispensés, généralement d’octobre à juillet dans le cadre du projet.

**Classement des élèves :** Organisation des élèves par classe, basée sur des critères comme l'âge ou le niveau scolaire.

**Paiement impayé :** Montant dû par un élève pour les frais d’incription qui n'a pas encore été réglé.

# Table des matières

[**CURRICULUM VITAE I**](#_Toc187863250)

[**Remerciements VI**](#_Toc187863251)

[**Liste des figures VII**](#_Toc187863252)

[**Liste des tableaux IX**](#_Toc187863253)

[**Liste des abréviations X**](#_Toc187863254)

[**PARTIE I : PRESENTATIONS 2**](#_Toc187863255)

[**Chapitre 1. Présentation de l’École Nationale d’Informatique 3**](#_Toc187863256)

**1.1**[**Domaine de spécialisation 6**](#_Toc187863257)

**1.2** [**Relation de l’ENI avec les organismes externes 8**](#_Toc187863258)

**1.3** [**Débouchés professionnels et diplômés 9**](#_Toc187863259)

**1.4** [**Ressources humaines 10**](#_Toc187863260)

[**Chapitre 2 : Présentation de Lycée Loterana Rakoto Andrianarijaona 11**](#_Toc187863261)

[**2.1 Information d’ordre générale 11**](#_Toc187863262)

[**2.2 Historiques en général 11**](#_Toc187863263)

[**2.3 Ces dirigeants se sont relayes pour diriger l’établissement jusqu’à aujourd’hui : 12**](#_Toc187863264)

[**2.3 Organigramme 12**](#_Toc187863265)

[**2.5 Ressources humaines 13**](#_Toc187863266)

[**Chapitre 3. Description du projet 14**](#_Toc187863267)

[**3.1 Formulation 14**](#_Toc187863268)

[**3.2 Objectif et besoins de l’utilisateur 14**](#_Toc187863269)

[**3.3 Moyens nécessaires à la réalisation du projet 15**](#_Toc187863270)

[3.3.1 Moyens matériels 15](#_Toc187863271)

[3.3.2 Moyens logiciels 15](#_Toc187863272)

[3.3.3 Moyens humains 16](#_Toc187863273)

[**3.4 Résultats attendus 16**](#_Toc187863274)

[**3.5 Chronogramme du projet 16**](#_Toc187863275)

[**Chapitre 4. Analyse préalable 17**](#_Toc187863276)

[**4.1 Analyse de l’existant 17**](#_Toc187863277)

[4.1.1 Organisation actuelle 17](#_Toc187863278)

[4.1.2 Inventaire des moyens matériels et logiciels 17](#_Toc187863279)

[a. Moyens matériels 17](#_Toc187863280)

[b. Moyens logiciels 18](#_Toc187863281)

[**4.2 Critique de l’existant 18**](#_Toc187863282)

[4.2.1 Conception avant -projet 18](#_Toc187863283)

[4.2.2 Méthodes de conception et outils utilisés 19](#_Toc187863284)

[a. Choix de méthode de conception et de modélisation 19](#_Toc187863285)

[b. Choix du système de gestion de base de données 21](#_Toc187863286)

[c. Outil de modélisation 22](#_Toc187863287)

[d.Langage de programmation 23](#_Toc187863288)

[**Chapitre 5. Analyse conceptuelle 27**](#_Toc187863289)

[**5.1 Dictionnaire des données 27**](#_Toc187863290)

[**5.3 Règles de gestion 29**](#_Toc187863291)

[**5.4. Représentation et spécification des besoins 30**](#_Toc187863292)

[5.4.1. Diagramme des cas d’utilisation 30](#_Toc187863293)

[5.4.2 Description textuelle pour chaque cas d’utilisation 32](#_Toc187863294)

[5.4.3 Priorisation des cas d’utilisation 39](#_Toc187863295)

[5.4.4 Diagramme des séquences système pour chaque cas d’utilisation 39](#_Toc187863296)

[**5.5 Spécification des besoins techniques 47**](#_Toc187863297)

[**5.6 Modélisation du domaine 47**](#_Toc187863298)

[**Chapitre 6. Conception détaillé 47**](#_Toc187863299)

[**6.1. Architecture système 49**](#_Toc187863300)

[**6.2 Diagramme de séquence de conception pour chaque cas d’utilisation 50**](#_Toc187863301)

[**6.3 Diagramme de classe de conception pour chaque cas d’utilisation 57**](#_Toc187863302)

[**6.2 Diagramme de classe globale 60**](#_Toc187863303)

[**6.3 Diagramme de paquetage 61**](#_Toc187863304)

[**6.4 Diagramme de déploiement 62**](#_Toc187863305)

[**Partie III : Réalisation 63**](#_Toc187863306)

[**Chapitre 7. Mise en place de l’environnement de développement 64**](#_Toc187863307)

[**7. 1 Installation et configuration des outils 64**](#_Toc187863308)

[7.1.2 Installation de Visual studio Code 64](#_Toc187863309)

[7.1.3 Installation de Visual Paradigm 65](#_Toc187863310)

[7.1.1. Installation de Node.js 66](#_Toc187863311)

[7.1.4 Installation de PostgreSQL xi](#_Toc187863312)

[7.2 Architecture de l’applications xii](#_Toc187863313)

[**Chapitre 8. Développement de l’application xiii**](#_Toc187863314)

[**8.1. Création de la base de données xiii**](#_Toc187863315)

[**8.2. Codage de l’application xiii**](#_Toc187863316)

[**8.3. Présentation de l’application xvii**](#_Toc187863317)

[**Conclusion xvii**](#_Toc187863318)

[**Références bibliographiques xx**](#_Toc187863319)

[**Références webographique xxii**](#_Toc187863320)

[**Glossaire xii**](#_Toc187863321)

[**Table des matières xiii**](#_Toc187863322)

[**Résumé ……………………………………………………………………………xvii**](#_Toc187863323)

[**Abstract xvii**](#_Toc187863324)

### Résumé

Le stage réalisé au sein du Lycée LRRA a pour objectif la conception et la mise en œuvre d'une application web dédiée à la gestion des inscriptions et écolages. À travers l'utilisation du langage de modélisation UML, de l'outil de conception Visual Paradigm, de MySQL comme système de stockage des données, et du langage de programmation Javascript, le développement de l'application a été orchestré via l'IDE VS Code. L'essence de ce projet réside dans la création d'une interface conviviale pour les utilisateurs, offrant des fonctionnalités précisément définies. Parallèlement, une page d'administration a été spécialement conçue pour permettre une gestion efficiente des utilisateurs, avec un accès administratif réservé à un unique administrateur au sein de l'application.

**Mots clés :** *Inscriptions, écolages, Lycée LRA, MySQL,UML.*

### Abstract

The internship conducted at LRA High School aims to design and implement a web application dedicated to register and schooling. Utilizing the UML modeling language, Visual Paradigm as the design tool, MySQL for data storage,web programming language JavaScript, the application development was facilitated using the VS Code IDE.The core objective of this project is to create a user-friendly interface with precisely defined features. Simultaneously, a dedicated administration page has been developed to ensure efficient user management, with administrative access restricted to a single administrator within the application.

**Keywords:** Registration, schooling, *LRRA High School, MySQL, UML, web application.*