

Faculté des Sciences Exactes et d'Informatique
Département de Mathématique et Informatique

Filière : Informatique

Mémoire du Fin d'Etude

Option : Système d'Information

Thème :

Gestion d'une école Privée

Etudiants :

BENZIANE Kheira.

BENGUIRECHE Khaoula.

Sous la supervision du Mme.BETOUATI Fatiha

Année Universitaire 2022-2023

Résumé :

Ce document est le rapport final du projet de fin d'études de licence en système informatique. Le projet vise à concevoir et à développer une application de bureau pour la gestion d'une école privée. Cette application a pour objectif d'automatiser les tâches administratives et de faciliter la gestion des élèves, des enseignants, des programmes d'études, des notes, des absences, des horaires, des finances, etc.

Le développement de cette application a nécessité la mise en œuvre de plusieurs compétences techniques telles que la programmation, la base de données, l'interface utilisateur et la gestion de projet. L'application est conçue pour répondre aux besoins spécifiques de l'école et pour offrir une expérience intuitive et conviviale aux utilisateurs.

Ce projet a permis de contribuer à la résolution d'un problème réel dans le domaine de l'éducation.

Notre application offre les services suivants :

- Pour l'admin : mise à jour les profils (élèves, enseignants), gérer les absences et les paiements, production des factures... Etc.
- Pour l'enseignant : consultation des listes des étudiants, saisit les notes et les absences pour chaque élèves, consultation de l'horaire de cours ...etc.

Abstract :

This document is the final report of the Computer System Graduate Project. The project aims to design and develop an desktop application for the management of a private school.

This application aims to automate administrative tasks and facilitate the management of students, teachers, study programs , grades, absences, schedules, finances, etc. The development of this application required the implementation of several technical skills such as programming, database, user interface and project management. The application is designed to meet the specific requirements of the school and provide an intuitive and easy-to-use user experience.

This project helped to solve a real problem in the field of education.

Our application offers the following services:

- For the admin: updating profiles (student, teachers), managing absences and payments, production of invoices... Etc.
- For the teacher: consultation of student lists, entry of notes and absences for each student, consultation of the course schedule ...etc

المخلص :

هذه الوثيقة هي التقرير النهائي للمشروع لدرجة الليسانس في أنظمة الكمبيوتر. يهدف المشروع إلى تصميم وتطوير تطبيق سطح المكتب لإدارة مدرسة خاصة. يهدف هذا التطبيق إلى أتمتة المهام الإدارية وتسهيل إدارة الطلاب والمعلمين والبرامج الدراسية والدرجات والغياب والجدول الزمنية والمالية وما إلى ذلك. يتطلب تطوير هذا التطبيق تنفيذ العديد من المهارات التقنية مثل البرمجة وقاعدة البيانات وواجهة المستخدم وإدارة المشاريع. تم تصميم التطبيق لتلبية الاحتياجات المحددة للمدرسة ولتوفير تجربة بديهية وسهلة الاستخدام للمستخدمين. وقد ساهم هذا المشروع في حل مشكلة حقيقية في مجال التعليم. يقدم تطبيقنا الخدمات التالية

للمشرف: تحديث الملفات الشخصية (الطالب والمعلمين)، وإدارة الغياب والمدفوعات، وإنتاج الفواتير ... الخ

بالنسبة للمعلم: الاطلاع على قوائم الطلاب، وإدخال الملاحظات والغياب لكل طالب، والتشاور مع الجدول الدراسي ... الخ

Remercîment :

Tout d'abord, nous remercions le bon Dieu Tout-Puissant, qui nous a donné la force de survivre, et nous a aussi donné le courage de surmonter toutes les difficultés et le courage d'achever ce travail.

Nos les plus sincères remerciements et gratitude à notre encadrante, Mme BETOUATI Fatiha, pour sa disponibilité et sa confiance en nous.

Nous remercions nos très chers parents, qui ont toujours été là pour nous. Nous remercions nos sœurs , nos frères et nos chers amis pour leurs encouragements.

Un remerciement a monsieur DELLALI Farouk pour sa disponibilité et pour son aide durant le travail.

À tous ces intervenants, nous présentons nos remerciements, nos respects et nos gratitude

Table des matières

Résumé.....	1
Remercement.....	2
Table des matières.....	3
Liste des figures.....	5
Liste des abréviations.....	6
Introduction générale.....	7
Chapitre 01 Analyse des besoins et présentation de spécifications d'exigences de logiciel	8
1.1 Introduction.....	8
1.2 Cadre du projet	8
1.3 Ecole privée.....	8
1.3.1 Définition.....	8
1.3.2 Problématique.....	8
1.4 EduABSHeer.....	8
1.4.1 Définition.....	8
1.4.2 Organigramme.....	9
1.5 Présentation de spécifications d'exigences de logiciel.....	9
1.5.1 Introduction.....	9
1.5.1.1 Objectif du projet	9
1.5.1.2 Portée du projet	9
1.5.1.3 Définition , acronymes et abréviation du SEL	10
1.5.1.4 Vue d'ensemble	10
1.5.2 Description générale	10
1.5.2.1 L'environnement.....	10
1.5.2.2 Fonctionnalités principales	10
1.5.2.3 Caractéristiques des utilisateurs	10
1.5.2.4 Les contraintes	10
1.5.2.5 Hypothèse	11
1.5.3 Exigences spécifiques	11
1.5.3.1 Exigences fonctionnelles	11
1.5.3.2 Exigences de performance	13
1.5.3.3 Exigences non fonctionnelles	13
1.6 Conclusion.....	13
Chapitre 02 Conception et modélisation.....	14
2.1 Introduction.....	14
2.2 Méthode d'analyse et conception.....	14
2.2.1 Qu'est-ce que UML	14
2.2.2 A quoi sert UML.....	14
2.2.4 Les différents types de diagrammes UML.....	14
2.2.5 La méthode RUP.....	15
2.2.6 Pourquoi la méthode RUP.....	15
2.3 Conception de notre application	15
2.3.1 Diagramme de classe	15
2.3.2 Diagramme de profil	15
2.3.3 Diagramme de contexte statique	16
2.3.4 Diagramme de package	16
2.3.5 Diagramme de cas d'utilisation	16
2.3.6 Diagramme de séquence	17
2.4 Conclusion.....	22

Chapitre 03 Implémentation.....	23
3.1 Introduction.....	23
3.1 Environnement de développement.....	23
3.2 Langage de programmation.....	24
3.3 Base de données	24
3.4 Présentation des interfaces	25
3.5 Conclusion.....	32
Conclusion générale.....	34
Bibliographie	35
Annexe	35
4.1 Les diagrammes	35
4.2 Les interfaces	

Liste des figures

Figure N°	Titre	Page N°
Figure 1	Exemple d'une école privée.	8
Figure 2	Organigramme de EduABSher	9
Figure 3	Diagrammes UML	15
Figure 4	Diagramme de classe	16
Figure 5	Diagramme de profil de notre approche	17
Figure 6	Diagramme de contexte statique	17
Figure 7	Diagramme de package	17
Figure 8	Diagramme de cas d'utilisation « enseignant »	18
Figure 9	Diagramme de cas d'utilisation « admin »	18
Figure 10	Diagramme de séquence « S'identifie »	19
Figure 11	Diagramme de séquence « inscription »	21
Figure 12	Diagramme de séquence « Frais scolaire »	22
Figure 13	Diagramme de séquence « emploi de temps »	23
Figure 14	Diagramme de séquence « charges »	24
Figure 15	Logo drawio	29
Figure 16	Logo intelliJ	29
Figure 17	Logo scene builder	30
Figure 18	Logo Microsoft office word 2007	30
Figure 19	Logo Power Point	31
Figure 20	Logo Java	31
Figure 21	Logo MySQL	31
Figure 22	Logo phpMyAdmin	32
Figure 23	Logo XAMPP	32
Figure 24	Logo JasperReport	32
Figure 25	Interface d'accueil « admin »	33
Figure 26	Interface d'accueil « Enseignant »	33
Figure 27	Interface d'authentification	33
Figure 28	Interface de liste des élèves	34
Figure 29	Interface d'ajouter un employé	34
Figure 30	Interface des classes et groupes pour enseignants	35
Figure 31	Interface de gestionnaire de notes	35
Figure 32	Diagramme de séquence « absences »	39
Figure 33	Diagramme de séquence « charges »	39
Figure 34	Interface d'ajouter classes et groupes.	39
Figure 35	Interface des charges	40
Figure 36	Logo EduManage	40

Liste des abréviations

Abréviation	Expression	Page
SGS	Système de gestion scolaire	7
FSEI	Faculté des sciences exacte et informatique	9
SEL	Spécifications d'exigences de logiciel	11
UML	Unified Modeling Language	17
RUP	Processus Unifié	18

Introduction générale :

Le système de gestion scolaire est un logiciel destiné à aider les institutions. C'est un logiciel qui aide à gérer les opérations administratives et académiques quotidiennes de l'école en douceur à distance. Un système de gestion scolaire est développé pour aider le personnel à faire moins de travail, aider les étudiants et les enseignants à gagner du temps, à réduire les coûts et à améliorer la sécurité des données.

Les systèmes de gestion scolaire traditionnel posent des différents problèmes parce qu'elles n'avaient pas de moyen systématique de stocker des informations, comme :

- La gestion manuelle et papier : le système de gestion scolaire traditionnel implique souvent une gestion manuelle et sur papier de nombreuses tâches administratives, telles que l'enregistrement des notes, l'inscription des étudiants, la gestion des absences, etc. Cette méthode est souvent fastidieuse et peut entraîner des erreurs humaines et une perte de temps.
- Manque d'efficacité : les processus administratifs manuels sont souvent lents et inefficaces, ce qui peut causer des retards dans les opérations de l'école. Cela peut également rendre difficile l'accès aux informations, car les dossiers sont souvent dispersés et difficilement accessibles.
- Manque de sécurité des données : Les données importantes des étudiants, telles que les notes et les informations personnelles, peuvent être facilement perdues ou compromises avec un système de gestion scolaire traditionnel. Cela peut entraîner des problèmes de confidentialité et de sécurité pour les étudiants et l'école.
- Manque de flexibilité : le système de gestion scolaire traditionnel peut être rigide et ne pas s'adapter aux besoins changeants de l'école. Les modifications et les mises à jour peuvent prendre du temps et entraîner des coûts supplémentaires.

Pour pallier les problèmes précédents, les systèmes de gestion c'est une solution pour couvrir les tâches suivantes :

- Utilisation de logiciels de gestion scolaire : Les logiciels de gestion scolaire peuvent aider à automatiser les tâches administratives telles que l'enregistrement des notes, l'inscription des étudiants, la gestion des absences, etc. Les logiciels de gestion scolaire peuvent également fournir un accès facile aux informations, une meilleure communication entre les étudiants, les enseignants et les administrateurs de l'école.
- Stockage numérique des données : Les systèmes de gestion scolaires informatisés peuvent stocker les données importantes des étudiants de manière numériques, ce qui peut améliorer la sécurité et la confidentialité des données. Les données sont stockées dans un emplacement centralisé et sont facilement accessibles aux personnes autorisées.
- Flexibilité : Les systèmes de gestion scolaires numériques peuvent être adaptés pour répondre aux besoins spécifiques de l'école. Des modules supplémentaires peuvent être ajoutés pour gérer les opérations académiques, financières et administratives spécifiques à l'école.

En fin de compte, l'utilisation de systèmes de gestion scolaire numériques peut résoudre de nombreux problèmes associés au système de gestion scolaire traditionnel, en fournissant une gestion efficace, une meilleure communication, une sécurité accrue des données et une flexibilité pour s'adapter aux besoins changeants de l'école.

Dans l'objectif de répondre à la problématique exposée précédemment nous allons dans ce projet concevoir et implémenter un système de gestion scolaire numérique pour une école privée. Et pour cela notre rapport est constitué de trois chapitres suivants :

1. Chapitre 1 : Analyse des besoins et présentation de cahier de charge

Nous commençons par une présentation de notre thème : Application de gestion de l'école privée, ensuite une présentation de cahier des charges de notre projet.

2. Chapitre 2 : Modélisation en utilisant le langage UML.

Nous présentons la conception UML qui présente en détailles notre projet.

3. Chapitre 3 : Implémentation en Java et MySQL Server

L'implémentation de notre application. Le mémoire se termine avec une conclusion et un ensemble des perspectives.

Chapitre 01 : Analyse des besoins et présentation de spécifications d'exigences de logiciel

1.1 Introduction :

Dans ce chapitre, nous allons examiner les besoins spécifiques de notre école privée en matière de gestion et d'administration, afin de définir les spécifications du cahier des charges de notre application de gestion d'école privée.

Ce chapitre est constitué de deux parties :

- La première partie présente l'école privée
- La deuxième partie présente le cahier de charge sous la forme IEEE830 [1]
- 1.2 Cadre du projet :

Le projet fait partie de la préparation finale pour obtenir la licence informatique de la Faculté des sciences exactes et de l'informatique (FSEI) Mostaganem.

1.3 Ecole privée :

1.3.1 Définition :

Ensemble des établissements qui dispensent un enseignement de façon privée en étant sous contrat d'association avec l'Etat, sous contrat simple ou, plus rarement, hors contrat.[2]



Figure 1 : Exemple d'une école privée.

1.3.2 Problématique :

Il n'y a pas de logiciel gratuit ou standard disponible pour faciliter la gestion des écoles privées, notamment pour les opérations d'inscription et de paiement pour tous les étudiants et enseignants.

La création de logiciels peut mener à une meilleure gestion de l'école par rapport à une méthode traditionnelle qui exige beaucoup de temps, d'efforts et de ressources. Donc nous proposons une application de bureau qui va être facile à administrer avec des interfaces faciles à comprendre et à utiliser.

1.4 Education ABSher :

1.4.1 Définition :

Nous allons développer une application nommée « EduABShier » qui vise à faciliter la gestion quotidienne des établissements scolaires privés. Son objectif principal est de réduire la charge administrative en proposant des fonctionnalités pour gérer les enseignants, leurs emplois du temps et leurs absences, ainsi que pour gérer les inscriptions des élèves et suivre leur vie scolaire tout en garantissant la confidentialité de leurs données personnelles. En outre, l'application permet également de gérer les notes des élèves et de suivre leur progression scolaire. Elle vise à améliorer l'efficacité et l'efficience de la gestion scolaire, pour les administrateurs, les enseignants, les élèves et les parents.

1.4.2. Organigramme :

Pour réaliser l'application nous allons utiliser l'architecture client/serveur et voici l'organigramme de notre application :

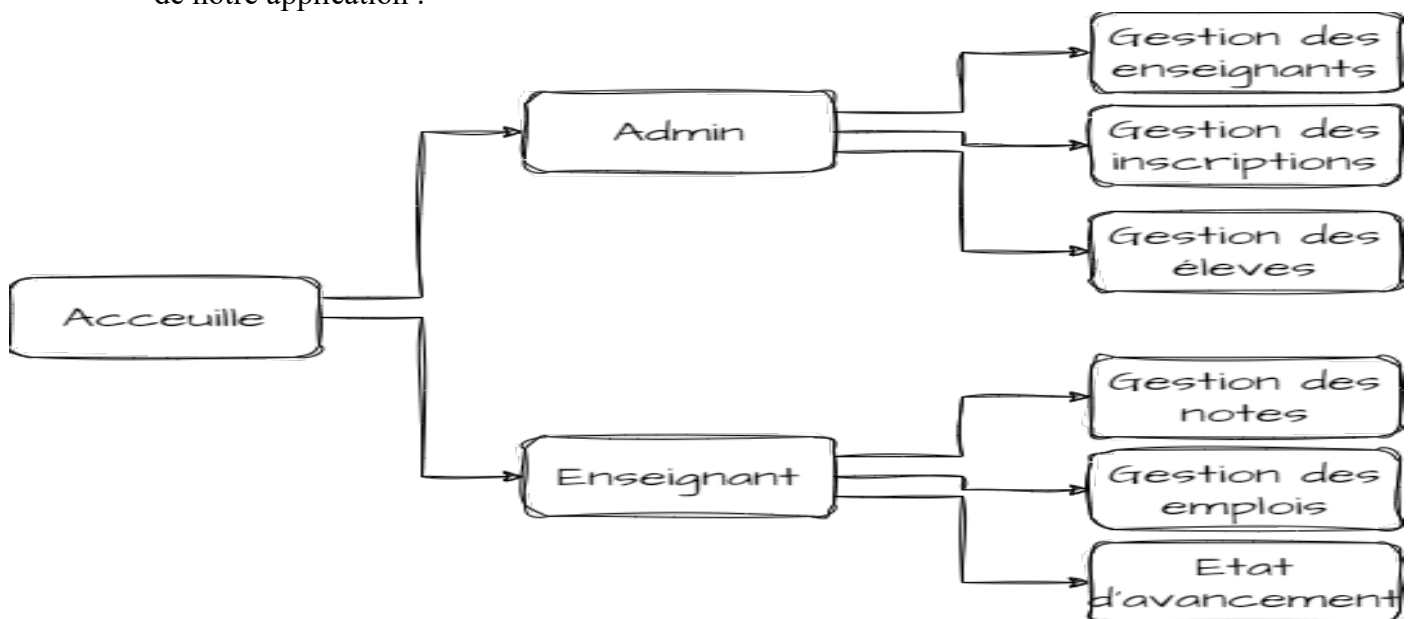


Figure 2 : Organigramme de EduABshier

1.4.2.1 Architecture client-serveur :

L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients du serveur : chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique. Dans notre travail nous allons adopter cette architecture [3].

1.5. Présentation de spécifications d'exigences de logiciel :

5.1. Introduction :

5.1.1 Objectif du projet :

L'objectif du projet est de développer une application de gestion d'école privée, cette spécification sert de guide pour l'ensemble du projet et constitue le document de base qui décrit les exigences que le projet doit atteindre.

- Les destinataires sont :

Le binôme : BENZIANE Kheira, BENGUIRECHE Khaoula, dans le cadre de projet fin d'étude

5.1.2 Portée du produit :

Le système développé sera nommé « EduABSher »

Le but de cette application est de fournir une solution informatique complète pour la gestion d'écoles privées et permettra de faciliter les tâches administratives. Elle a pour objectif de répondre aux besoins spécifiques d'une école privée en offrant des fonctionnalités pour la gestion quotidienne de l'école.

Voici quelque fonctionnalité de EduABSher :

- Gestion des enseignants
- Gestion des élèves
- Sécurité et confidentialité

5.1.3 Définition, acronymes, et abréviation du SEL :

EduABSher : Education ABSher, le nom de notre application

SGS : système de gestion scolaire.

5.1.4 Vue d'ensemble :

Ce document décrit les meilleures pratiques pour développer les spécifications des exigences logicielles.

- Expliquer la portée du document
- Expliquer comment notre projet est organisé.
- Donner une description générale du système
- Définir les exigences spécifiques

5.2 Description général :

5.2.1 L'environnement :

Le système à réaliser est indépendant et parfaitement autonome, et elle n'appartient à aucun système plus vaste.

5.2.2 Fonctionnalités principales :

- Inscrire les élèves
- Gérer les horaires et des emplois du temps
- Enregistrer les notes
- Gérer les absences et des retards
- Gérer les résultats d'examen
- Gérer les enseignants et de leurs affectations
- Gérer les frais de scolarité

5.2.3 Caractéristiques des utilisateurs :

Les utilisateurs du SGS seront principalement des enseignants et des administrateurs de l'école.

5.2.4 Les contraintes :

- Contrainte du développement :
 - Le système n'accepte pas les utilisateurs anonymes.
 - Chaque utilisateur son compte.
- Contrainte d'exploitation : Le système n'est accessible à aucun système d'exploitation. Windows seulement.
- Fonction de vérification :
 - Le champ Nom d'utilisateur ne peut contenir que des caractères alphabétiques.
 - Le champ Mot de passe doit contenir au moins 8 caractères.
- Langage de programmation :
 - Le système est développé en utilisant le langage de programmation java
- Base de données :
 - Le système utilise une base de données relationnelle (MySQL) pour stocker les informations.

5.2.5 Hypothèse :

Les utilisateurs disposeront d'une formation de base sur l'utilisation des applications informatiques

5.3 Exigences spécifiques :

1.Exigences fonctionnelles :

1. Classe d'utilisateur : Administrateur

Fonctionnalité 1 : Inscription et enregistrement des informations des élèves.

Entrées : Informations personnelles des élèves (nom, prénom, date de naissance, adresse, etc.), informations académiques (niveau d'études, programme, etc.)

Sorties : Enregistrement des informations des élèves dans la base de données, confirmation d'inscription, numéro d'identification étudiant (le cas échéant)

Tâches : Saisie des informations des élèves, validation des données, enregistrement dans la base de données, génération de confirmations et d'identifiants élèves

Acteurs : Administrateurs du système.

Fonctionnalité 2 : Gestion des horaires et des emplois du temps

Entrées : Informations sur les élèves, les enseignants, les cours, les classes, les disponibilités

Sorties : Horaires et emplois du temps générés pour les élèves et les enseignants, rapports d'emplois du temps

Tâches : Saisie et mise à jour des informations sur les cours, les élèves, les enseignants et les classes, création des emplois du temps, génération de rapports

Acteurs : Administrateurs du système, enseignants.

Fonctionnalité 3 : Gestion des absences et des retards des élèves

Entrées : Données d'absence et de retard des élèves, informations sur les horaires et les cours

Sorties : Rapports d'absence et de retard générés, notifications d'absence ou de retard pour les élèves et les enseignants

Tâches : Saisie et mise à jour des données d'absence et de retard, vérification des justifications, génération de rapports, envoi de notifications

Acteurs : Administrateurs du système, enseignants.

Fonctionnalité 4 : Gestion des résultats d'examen

Entrées : Résultats des examens des élèves, informations sur les évaluations et les matières

Sorties : Enregistrement et mise à jour des résultats des élèves, génération de rapports de résultats

Tâches : Saisie et validation des résultats d'examen, stockage dans la base de données, calcul des moyennes, génération de rapports, affichage des résultats pour les élèves

Acteurs : Administrateurs du système, enseignant.

Fonctionnalité 5 : Gestion des frais scolaires

Entrées : Informations sur les frais de scolarité.

Sorties : Enregistrement et suivi des informations sur les frais scolaires, génération de rapports financiers.

Tâches : Saisie et mise à jour des informations sur les frais scolaires, calcul des montants, génération de rapports, envoi de factures aux élèves.

Acteurs : Administrateurs du système, personnel administratif.

Fonctionnalité 6 : Génération de rapports.

Entrées : Données académiques, financières, d'inscription et de calendrier.

Sorties : Rapports sur les résultats académiques, les absences, les finances, les inscriptions

Tâches : Analyse des données, génération de rapports structurés, personnalisation des rapports selon les besoins, exportation ou affichage des rapports.

Acteurs : Administrateurs du système, personnel administratif

Fonctionnalité 7 : Création de nouveaux comptes enseignants.

Entrées : Informations personnelles des enseignants (nom, prénom, adresse, etc.), informations d'identification.

Sorties : Comptes enseignants créés avec succès, identifiants d'accès.

Tâches : Saisie des informations des enseignants, validation des données, création des comptes d'accès, envoi des identifiants aux enseignants.

Acteurs : Administrateurs du système

Fonctionnalité 8 : Gestion des données (élèves, classes, filières, niveau, groupes, enseignants, etc.).

Entrées : Informations sur les élèves (nom, prénom, date de naissance, etc.), classes (nom, horaires, etc.), filières (nom, description, etc.), niveaux (nom, description, etc.), groupes (nom, description, etc.), enseignants (nom, prénom, spécialité, etc.).

Sorties : Mises à jour des données des élèves, classes, filières, niveaux, groupes, enseignants, confirmation des modifications ou suppressions.

Tâches : Ajout, modification et suppression des données des élèves, classes, filières, niveaux, groupes, enseignants, vérification des données, mise à jour de la base de données.

Acteurs : Administrateurs du système

2. Classe d'utilisateur : Enseignant

Fonctionnalité 1 : Saisie des notes des élèves par les enseignants.

Entrées : Données des élèves, résultats des évaluations.

Sorties : Enregistrement des notes des élèves.

Tâches : Saisie des notes pour chaque évaluation, vérification des données, enregistrement des notes dans le système, calcul des moyennes.

Acteurs : Enseignants

Fonctionnalité 2 : Suivi de l'état d'avancement des matières par les enseignants.

Entrées : Informations sur les matières, contenus d'enseignement, objectifs d'apprentissage, données d'évaluation.

Sorties : État d'avancement des matières, suivi des progrès des élèves.

Tâches : Consultation des informations sur les matières et les contenus d'enseignement, enregistrement des données d'évaluation, suivi de la progression des élèves, identification des besoins supplémentaires d'enseignement ou de révision.

Acteurs : Enseignants

Fonctionnalité 3 : Consultation des emplois du temps des classes par les enseignants.

Entrées : Informations sur les enseignants.

Sorties : Affichage des emplois du temps des classes attribuées à l'enseignant.

Tâches : Consultation des emplois du temps des classes, vérification des horaires et des salles de classe.

Acteurs : Enseignants

Fonctionnalité 4 : Gestion des listes d'élèves pour chaque matière par les enseignants.

Entrées : Informations sur les élèves, matières enseignées.

Sorties : Listes d'élèves mises à jour pour chaque matière.

Tâches : Création et mise à jour des listes d'élèves pour chaque matière enseignée, ajout ou suppression d'élèves, vérification de l'exactitude des données.

Acteurs : Enseignants

5.4 Exigences de performances :

Notre application est compatible avec toutes les versions de Windows., mais nous la programmons sur un ordinateur avec ces exigences de performances suivantes :

- RAM :4.00 GO.
- Système d'exploitation : Windows10bit.
- Processeur : Intel(R) Core (TM) 3 CPU 4400 2.00GHZ
- Sureté : antivirus Kaspersky interne et Security.

5.5 Exigences non fonctionnelles :

1.Performance :

Le système doit offrir à une performance optimale pour gérer un grand nombre d'utilisateurs simultanément, sans ralentissement ou interruption.

2.Architecture client-serveur :

Le système est basé sur une architecture client-serveur, ou les clients doivent communiquer avec le serveur pour accéder aux fonctionnalités du système.

3.Sécurité :

Le système devrait être sécurisé, avec une authentification à plusieurs niveaux, des mesures de protection des données et une gestion des autorisations d'accès.

4.Ergonomie :

Le système devrait être ergonomique et convivial pour les utilisateurs, avec une interface utilisateur intuitive, des menus déroulants, des boutons, des messages d'erreur et des notifications claires.

1.6. Conclusion :

Après avoir donné une description d'un bon fonctionnement de l'établissement pour lequel nous allons concevoir une application, nous avons établi un cahier de charge conforme à la IEEE 830 std, le chapitre suivant nous allons procéder à la modélisation de notre application telle que décrite dans ce cahier de charge.

Chapitre 02 : Conception et Modélisation

2.1 Introduction :

La conception et le développement d'un système d'information nécessitent une approche méthodique, avec une étape essentielle de modélisation. Dans notre projet, nous avons choisi d'utiliser le langage UML pour représenter notre conception. Les diagrammes de cas d'utilisation, de séquence et de classe seront utilisés pour illustrer notre démarche. La conception est cruciale pour développer une application performante et cohérente, et la méthode UML est reconnue comme étant la plus adaptée à cet effet. Notre présentation comprendra plusieurs diagrammes pour représenter les différentes perspectives du système.

2.2 Méthode d'analyse et conception :

2.2.1 Qu'est-ce que UML :

Le Langage de Modélisation Unifié, de l'anglais *Unified Modeling Language* (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet[4].

L'UML est une synthèse de langages de modélisation objet antérieurs : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). UML 1.0 a été normalisé en janvier 1997; UML 2.0 a été adopté par l'OMG en juillet 2005. La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017)[5].

2.2.2 A quoi sert UML :

UML a été principalement utilisé comme langage de modélisation à usage général dans le domaine du génie logiciel. Cependant, il a maintenant trouvé sa place dans la documentation de plusieurs processus métier ou workflows. Par exemple, les diagrammes d'activités, un type de diagramme UML, peuvent être utilisés comme remplacement des organigrammes. Ils fournissent à la fois un moyen plus standardisé de workflows de modélisation ainsi qu'un plus large éventail de fonctionnalités pour améliorer lisibilité et efficacité.

Il existe plusieurs types de diagrammes UML et chacun d'eux sert un objectif différent, qu'il ait été conçu ou non avant la mise en œuvre ou après.

Les deux catégories les plus larges qui englobent tous les autres types sont comportementaux et structurels. Comme le nom le suggère, certains diagrammes UML tentent d'analyser et de décrire la structure d'un système ou processus, tandis que d'autres décrivent le comportement du système, ses acteurs et ses éléments constitutifs.

2.2.3 Les différents types de diagrammes UML :

Il y a deux types de diagrammes UML :

- Les diagrammes de structure représentent les éléments du système, leurs propriétés et leurs relations entre eux :

- Diagramme de classes.
 - Diagramme d'objets.
 - Diagramme de paquetage.
 - Diagramme de déploiement.
- Les diagrammes de comportement représentent les processus et les interactions entre les objets :
- Diagramme de cas d'utilisation.
 - Diagramme de séquence.
 - Diagramme de transition d'état.

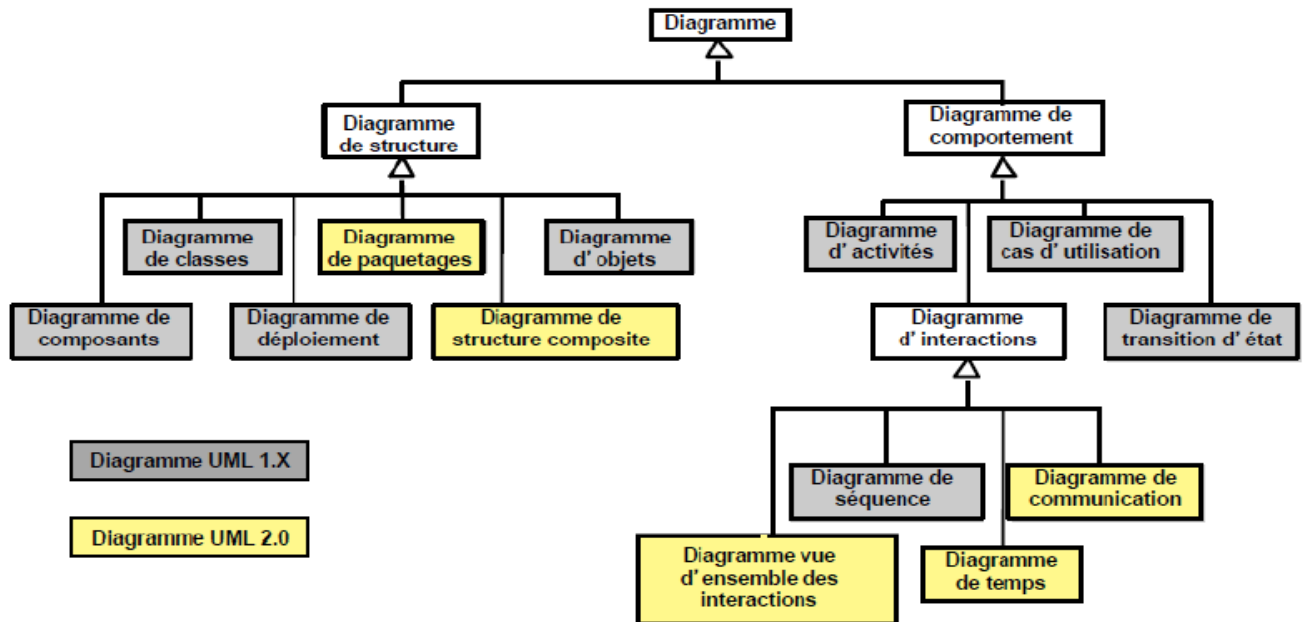


Figure 3 : Diagramme UML[6]

2.2.4 La méthode RUP :

La méthode de Processus Unifié (UP) s'intéresse au cycle de vie d'un logiciel et le gère. En ce qui concerne les logiciels orientés objet, elle se concentrera sur son développement. La méthode RUP (Rational Unified Process), quant à elle, est une des émanations de la méthode PU, qui s'attache à donner un cadre précis au développement du logiciel. C'est une méthode générique, itérative et incrémentale assez lourde mais qui s'adapte très facilement aux processus et aux besoins du développement[7].

2.2.5 Pourquoi la méthode RUP :

Cette méthode permet de répondre aux besoins des utilisateurs rapidement, dans les délais impartis, tout en respectant les budgets. En effet, elle canalise et modélise toutes les étapes du développement d'un logiciel. Elle en ordonne aussi très clairement les différents jalons.

2.3 Conception de notre application :

1. Diagramme de classe :

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction[8]

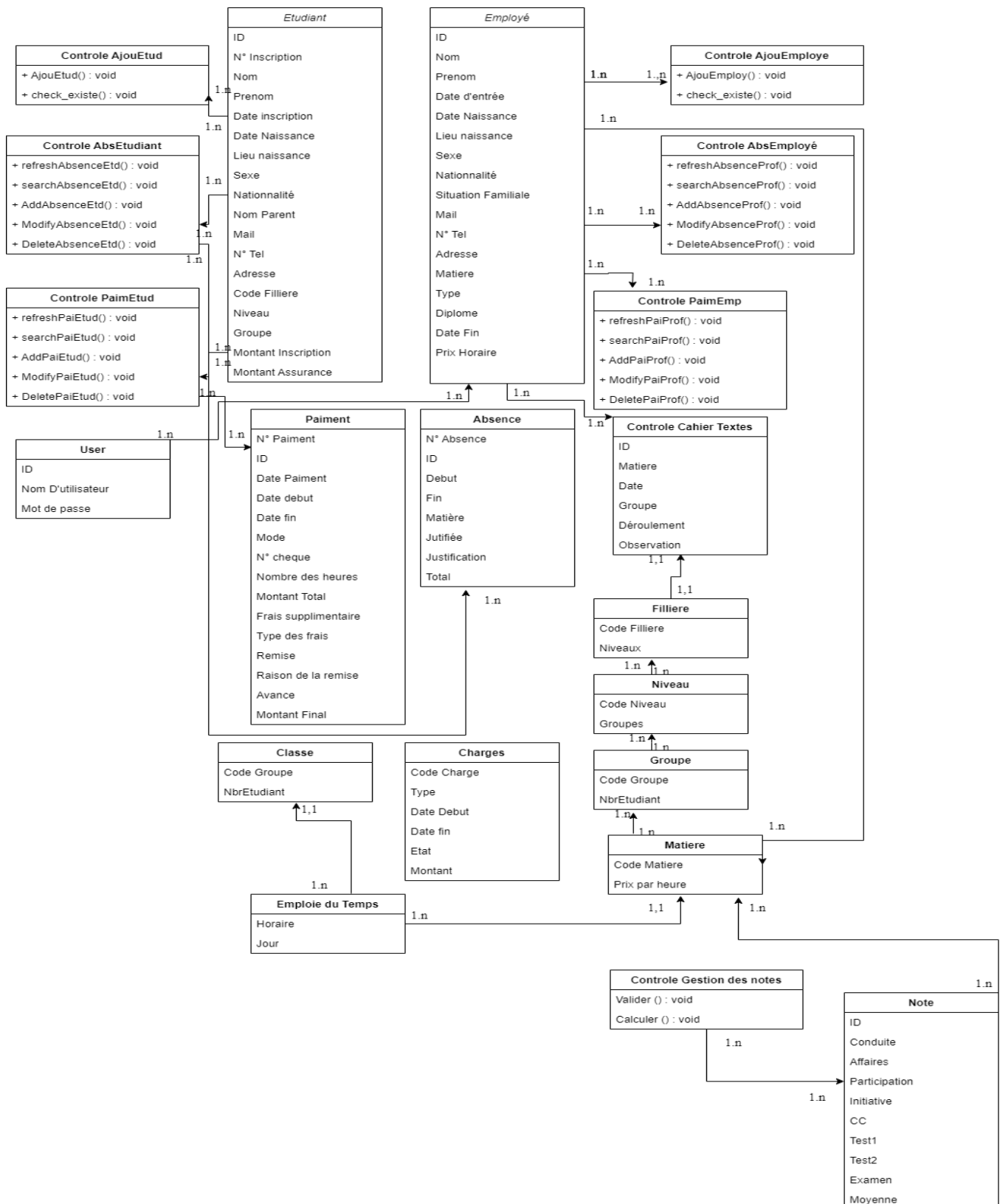


Figure 4 : Diagramme de classe de notre approche

2. Diagramme de profil :

Ce diagramme fournit une représentation des concepts utilisés dans la définition des profils.

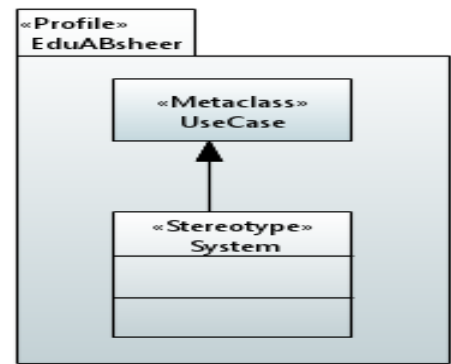


Figure 5 : Diagramme de profile de notre approche

3 Diagramme de contexte statique :

Nous représentons les interactions des acteurs avec le système étudié.

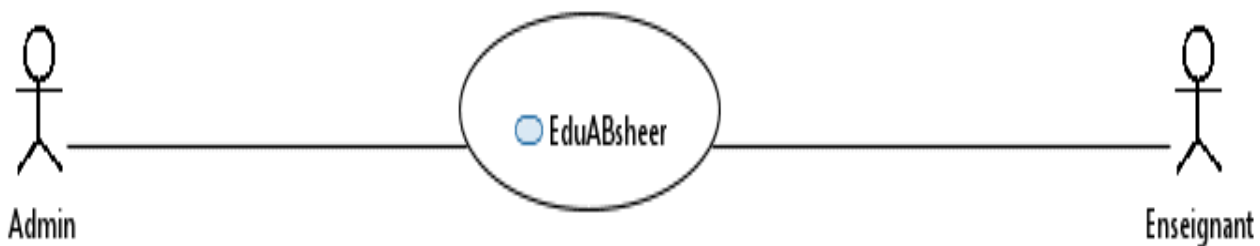


Figure 6 : Diagramme de contexte statique

4 Diagramme de package :

Les diagrammes de paquets sont la représentation graphique des relations existant entre les paquets (ou espaces de noms) composant un système, dans le langage Unified Modeling Language (UML)[9].



Figure 7 : Diagramme de package

5. Diagramme de cas d'utilisation (use-case diagram) :

L'objectif d'un diagramme de cas d'utilisation UML est de représenter les différentes façons dont un utilisateur peut interagir avec un système[10].

Dans cette section nous allons détailler les diagrammes de cas d'utilisation réalisés

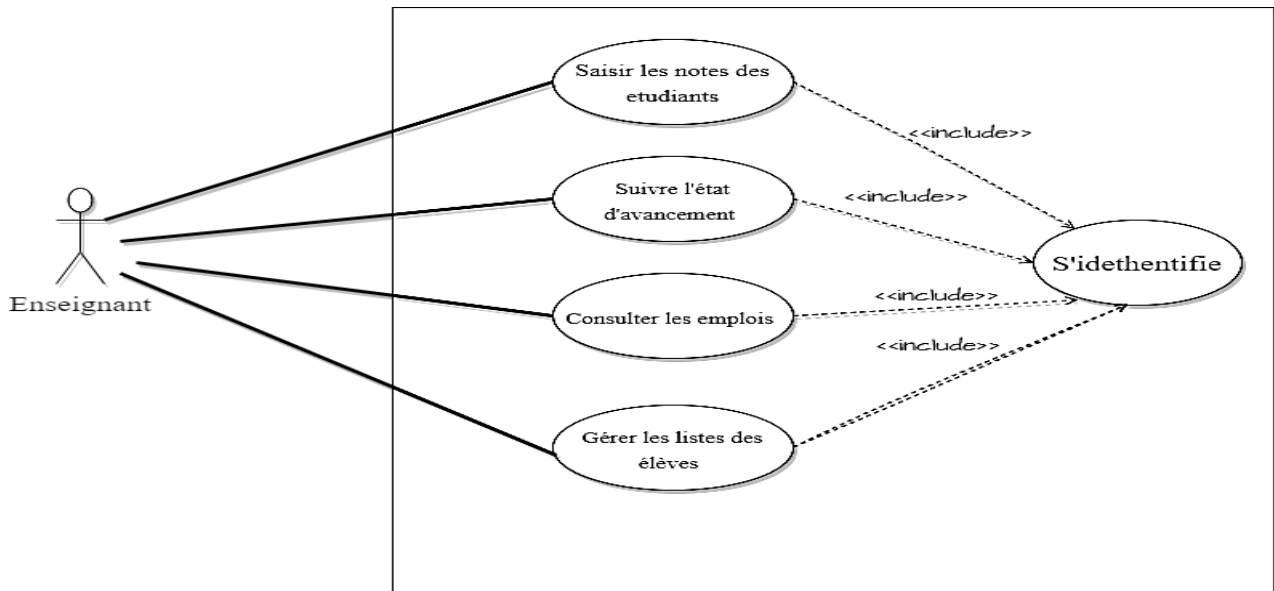


Figure 8 : Diagramme de cas d'utilisation « enseignant ».

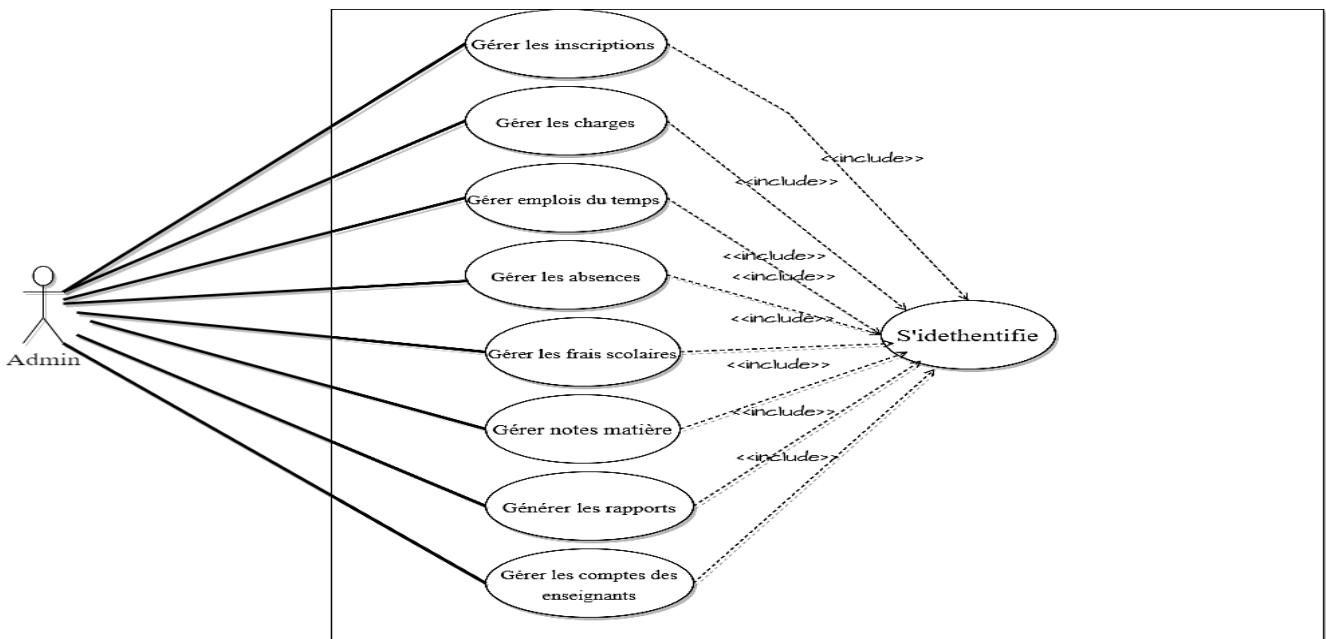


Figure 9 : Diagramme de cas d'utilisation « admin »

6. Diagramme de séquence (*sequence diagram*) :

Représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

7. Description textuelle du cas d'utilisation et les diagrammes de séquences

1. S'identifier :

Objectif : authentification dans l'application

Acteurs principaux : Admin et Enseignant

Pré conditions : avoir un compte

Postconditions : accès à son compte

Scénario nominal :

1. L'utilisateur ouvre l'application
2. Le système affiche l'interface d'authentification
3. L'utilisateur introduit son nom d'utilisateur et son mot de passe
4. Le système vérifie les champs introduit par l'utilisateur
5. Le système vérifie l'existence de l'utilisateur
6. Si les informations introduites sont correctes, le système affiche l'accueil du system

Scenarios alternatifs :

E1 : Champs obligatoires non valide et/ou vides

3. Le système affiche un message d'erreur
4. Le système reprend à l'étape 2 du scénario nominal

E2 : Email et/ ou mot de passe incorrect

3. Le système affiche un message d'erreur indiquant « Veuillez vérifier vos coordonnées et réessayer » »
4. Le système reprend à l'étape 2 du scénario nominal

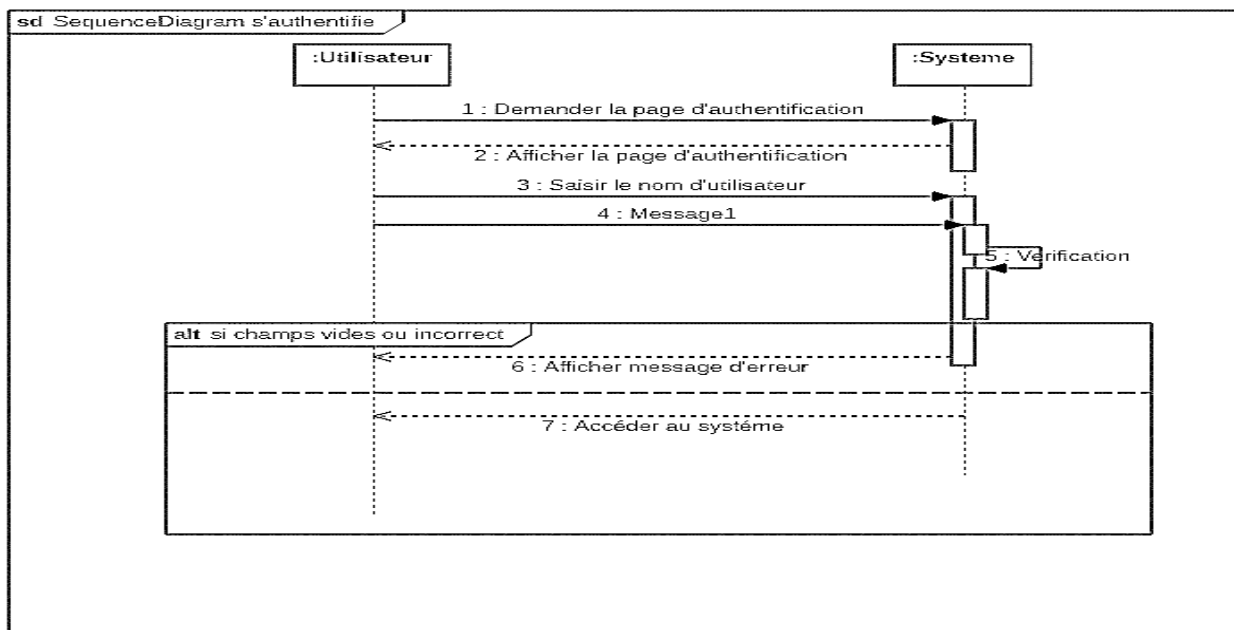


Figure 10 : Diagramme de séquence « S'identifie»

2.diagramme de séquence de gestion de l'inscription :

Objectif : Permet à l'administrateur d'ajouter des étudiants et des enseignants dans le système.

Acteurs principaux : Admin, système.

Précondition : L'administrateur est connecté au système et avoir les autorisations nécessaires pour gérer les inscriptions.

Postcondition : Les informations des étudiants et des enseignants sont enregistrées dans le système.

Scénario nominal :

- 1.L'administrateur sélectionne l'option "Gérer les inscriptions" dans le système.
- 2.Le système affiche un formulaire d'inscription.
- 3.L'administrateur saisit les informations de l'étudiant dans le formulaire.
4. L'administrateur valide les informations d'inscription.
- 5.Le système effectue une vérification des données d'inscription.
6. Le système génère un identifiant étudiant unique pour l'étudiant.
7. Les informations de l'étudiant sont enregistrées dans la base de données du système.
- 8.Le système affiche un message de confirmation d'inscription.
- 9.L'administrateur sélectionne l'option "Ajouter un enseignant" dans le système.
- 10.Le système affiche un formulaire d'inscription spécifique à l'enseignant.
- 11.L'administrateur saisit les informations de l'enseignant dans le formulaire.
- 12.L'administrateur valide les informations d'inscription de l'enseignant.
13. Le système effectue une vérification des données d'inscription de l'enseignant.
- 14.Le système génère un identifiant unique pour l'enseignant.
- 15.Les informations de l'enseignant sont enregistrées dans la base de données du système.
- 16.Le système affiche un message de confirmation d'inscription de l'enseignant.
17. L'administrateur a le droit d'imprimer une attestation d'inscription
- 18.Le processus de gestion des inscriptions se termine.

Scénarios alternatifs :

E 5 : Si les données d'inscription sont invalides, le système affiche un message d'erreur et retourne à l'étape 3 pour permet à l'administrateur de corriger les informations.

E 13 : Si les données d'inscription de l'enseignant sont invalides, le système affiche un message d'erreur et retourne à l'étape 10 pour permet à l'administrateur de corriger les informations.

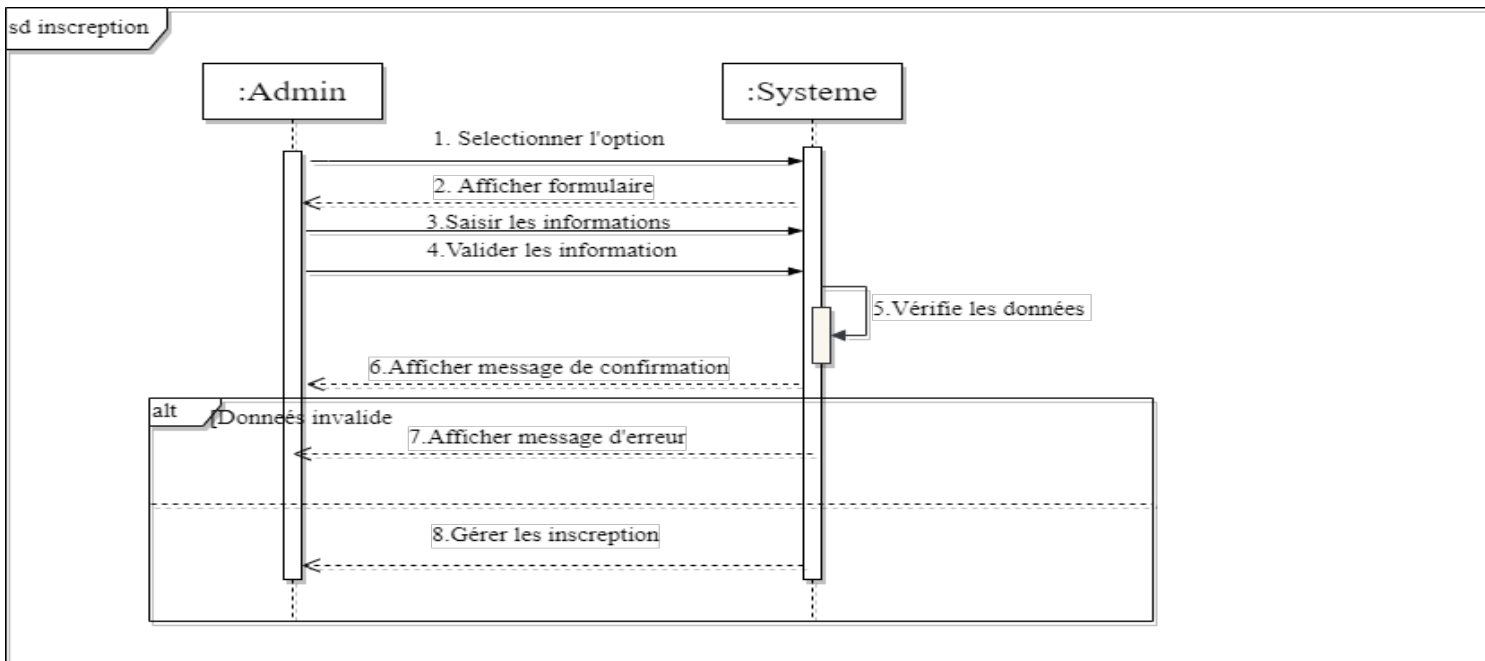


Figure 11 : Diagramme de séquence « inscription »

3. diagramme de séquence de Frais scolaires :

Objectif : Permet aux administrateurs de gérer les paiements des étudiants.

Acteurs principaux : Administrateur, Étudiant

Précondition : L'administrateur est connecté au système et il a les autorisations nécessaires pour gérer les paiements.

Les informations des étudiants et des frais scolaires doivent être préalablement enregistrées dans le système.

Postcondition : Les paiements des étudiants sont enregistrés dans le système.

Les mises à jour des soldes des étudiants sont effectuées.

Scénario nominal :

1. L'administrateur sélectionne l'option "Gérer paiement" dans le système.
2. Le système affiche la fenêtre du frais scolaire qui contient un formulaire.
3. L'administrateur saisit les informations nécessaires pour paiement dans le formulaire.
4. L'administrateur valide le paiement.
5. Le système enregistre le paiement dans la base de données.
6. Le système affiche un message de confirmation du paiement enregistré.
7. L'administrateur a le droit d'imprimer le reçu de paiement
9. Le processus de gestion des paiements se termine.

Scénarios alternatifs :

E 6 : Si le montant du paiement saisi par l'administrateur est invalide, le système affiche un message d'erreur et retourne à l'étape 5 pour permettre à l'administrateur de corriger le montant

E7 : Si une erreur survient lors de l'enregistrement du paiement, le système affiche un message d'erreur et retourne à l'étape 5 pour permet à l'administrateur de réessayer.

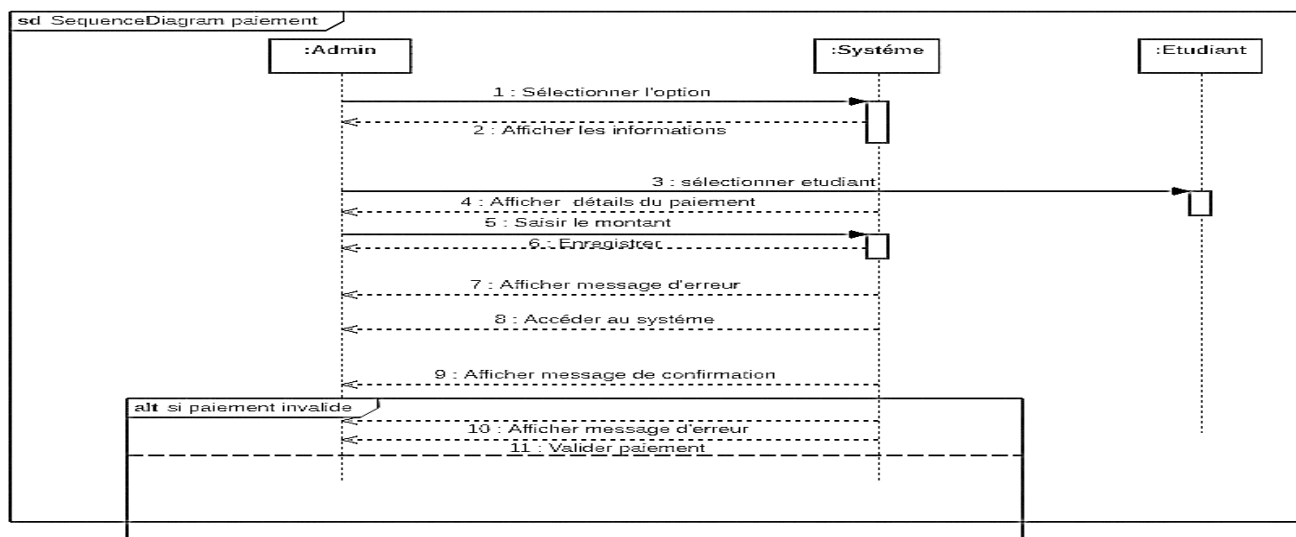


Figure 12 : Diagramme de séquence « frais scolaire »

4. diagramme de séquence de gérer emplois :

Objectif : Permet aux administrateurs de gérer les emplois du temps des étudiants et des enseignants.

Acteurs principaux : Administrateur, Enseignant

Précondition : L'administrateur est connecté au système et avoir les autorisations nécessaires pour gérer les emplois du temps. Les informations des étudiants, des enseignants et des cours doivent être préalablement enregistrées dans le système.

Postcondition : Les modifications apportées aux emplois du temps sont enregistrées dans le système.

Scénario nominal :

1. L'administrateur sélectionne l'option "Gérer emplois" dans le système.
2. Le système affiche la fenêtre d'emploi du temps
3. L'administrateur choisit une filière, un niveau, un enseignant et une matière pour afficher, modifier et créer l'emploi du temps.
4. L'administrateur remplit l'heure, le jour, pour effectuer les enseignants.
5. Le système affiche l'emploi du temps.
6. L'administrateur peut effectuer les actions suivantes :
 - Ajouter un cours à l'emploi du temps.
 - Modifier les détails d'un cours existant.
 - Supprimer un cours de l'emploi du temps.
7. L'administrateur enregistre les modifications apportées à l'emploi du temps.
8. Le système met à jour les données de l'emploi du temps dans la base de données.
9. Le système affiche un message de confirmation des modifications enregistrées.
10. Le processus de gestion des emplois du temps se termine.

Scénario alternatif :

E 6 : Si l'administrateur décide d'annuler les modifications, le système ne sauvegarde pas les changements et retourne à l'étape 4 pour permettre à l'administrateur de continuer à modifier l'emploi du temps.

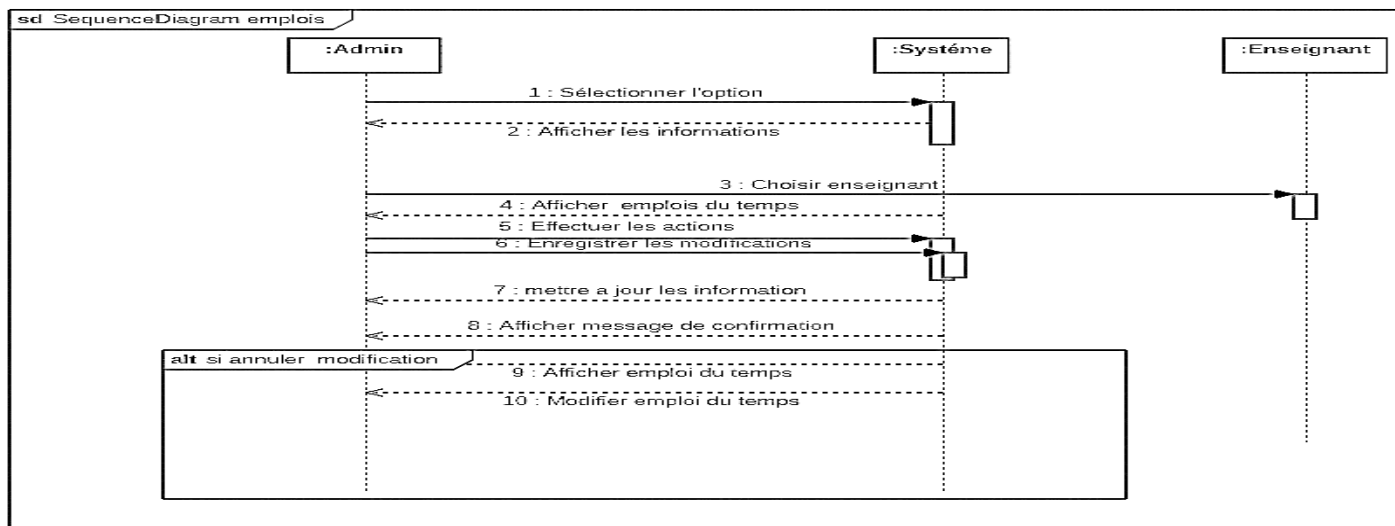


Figure 13 : diagramme de séquence « emplois du temps »

4.Gérer absences :

Objectif : Permettre aux administrateurs de gérer les absences des étudiants et des enseignants.

Acteurs principaux : Administrateur, Enseignant

Précondition : L'administrateur doit être connecté au système et avoir les autorisations nécessaires pour gérer les absences. Les informations des étudiants, des enseignants et des cours doivent être préalablement enregistrées dans le système.

Postcondition : Les informations sur les absences et les retards sont enregistrées dans le système.

Scénario nominal :

- 1.L'administrateur sélectionne l'option "Gérer absences " dans le système.
- 2.Le système affiche un formulaire d'absence.
3. .L'administrateur saisit les informations nécessaires pour l'absence dans le formulaire
- 4.l'administrateur valide les informations .
- 5.L'administrateur peut effectuer les actions suivantes : Ajouter une absence pour un étudiant ou un employée
Modifier les détails d'une absence ou d'un retard existant. Supprimer une absence.
- 6.L'administrateur enregistre les modifications apportées aux absences et aux retards.
- 7.Le système met à jour les données des absences dans la base de données.
- 8.Le système affiche un message de confirmation des modifications enregistrées.
- 9.Le processus de gestion des absences se termine.

Scénario alternatif :

E6 : Si l'administrateur décide d'annuler les modifications, le système ne sauvegarde pas les changements et retourne à l'étape 4 pour permettre à l'administrateur de continuer à gérer les absences et les retards.

5.Gérer les charges :

Objectif : Permettre aux administrateurs de gérer les frais scolaires, y compris la saisie des informations sur les frais de scolarité, les charges de restauration.

Acteurs principaux : Administrateur, Étudiant

Précondition : L'administrateur doit être connecté au système et avoir les autorisations nécessaires pour gérer les frais scolaires.

Postcondition : Les informations sur les frais scolaires sont enregistrées dans le système.

Scénario nominal :

- 1.L'administrateur sélectionne l'option "Gérer frais scolaires" dans le système.
- 2.Le système affiche un formulaire
3. L'administrateur saisit les informations nécessaires pour les charges dans le formulaire
4. L'administrateur valide la tâche .
- 5.L'administrateur peut effectuer les actions suivantes :
 - Ajouter de nouveaux frais pour un étudiant.
 - Modifier les détails des frais existants.
 - Supprimer des frais.
- 6.L'administrateur enregistre les modifications apportées aux frais scolaires.
- 7.Le système met à jour les données des frais scolaires dans la base de données.
- 8.Le système affiche un message de confirmation des modifications enregistrées.
- 9.Le processus de gestion des frais scolaires se termine.

Scénarios alternatif :

E6 : Si l'administrateur décide d'annuler les modifications, le système ne sauvegarde pas les changements et retourne à l'étape 4 pour permettre à l'administrateur de continuer à gérer les frais scolaires.

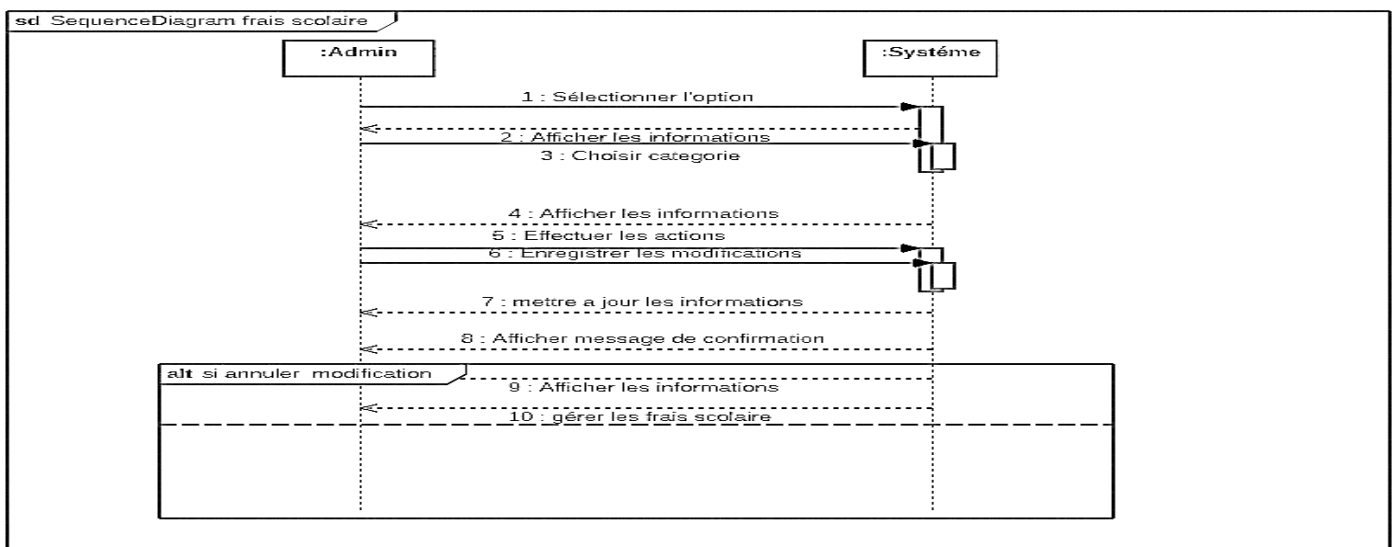


Figure 14 : Diagramme de séquence « Charges »

2.4. Conclusion :

Dans ce chapitre, nous avons réalisé la modélisation d'un système d'information pour les écoles privées en utilisant le langage de modélisation UML. Nous avons présenté plusieurs diagrammes, tels que les diagrammes de cas d'utilisation, de séquence et de classes, afin de comprendre le fonctionnement de

l'application. Une fois la conception prête, dans le chapitre suivant, nous entrons dans la phase de développement et de mise en œuvre de l'application en suivant les principes établis lors de la conception.

Chapitre 03 : Implémentation

3.1 Introduction :

Après avoir détaillé la conception de l'application, nous avons consacré le dernier chapitre du rapport à la partie implémentation.

Ce chapitre présente les outils et les environnements de programmation utilisés pour mettre en œuvre notre application **Environnement de développement :**

1.Draw.io :

C'est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter. Draw.io est un véritable couteau suisse de la frise chronologique, de la carte mentale et des diagrammes de tout genre [11].

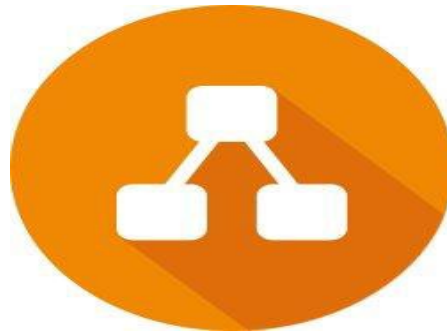


Figure 15 : Logo draw.io

2.IntelliJ IDEA :

IntelliJ IDEA est un environnement de programmation spécial ou un environnement de développement intégré (IDE) largement destiné à Java. Cet environnement est utilisé notamment pour le développement de programmes. Il est développé par une société appelée JetBrains, qui s'appelait officiellement IntelliJ. Il est disponible en deux éditions : l'édition communautaire sous licence Apache 2.0 et une édition commerciale connue sous le nom d'Ultimate Edition. Les deux peuvent être utilisés pour créer des logiciels qui peuvent être vendus. Ce qui rend IntelliJ IDEA si différent de ses homologues, c'est sa facilité d'utilisation, sa flexibilité et sa conception solide.[12]

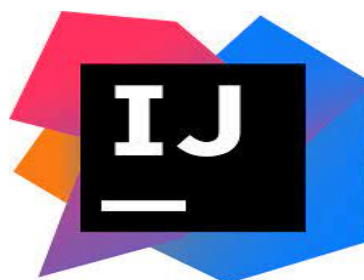


Figure 16 : Logo de IntelliJ

3.Scene Builder :

JavaFX Scene Builder est un outil de mise en page visuelle qui permet aux utilisateurs de concevoir rapidement des interfaces utilisateur pour les applications JavaFX sans codage. Les

utilisateurs peuvent faire glisser et déposer des composants d'interface utilisateur dans l'espace de travail, modifier leurs propriétés et appliquer des feuilles de style. Le code FXML de mise en page que vous créez est automatiquement généré en arrière-plan. Le résultat est un fichier FXML que vous pouvez combiner avec votre projet Java en liant l'interface utilisateur à la logique de votre application [13].



Figure17 : Logo du scene builder

4. Microsoft office Word 2007 :

Microsoft Office 2007, anciennement nommé *Office 12*, est une évolution de la suite bureautique de Microsoft depuis sa création sortie le 30 janvier 2007 en remplacement de Microsoft Office 2003.

Parmi les nouveautés, les applications globales restent les mêmes et vient s'ajouter *Office « 12 » Server* et *Microsoft Expression Web* (succédant à FrontPage). La gamme Groove fait également son entrée dans Office 2007 avec une composante « client », Microsoft Office Groove, et une gamme « Serveurs » contenant Groove Management Server, Groove DataBridge, et Groove Relay Server. La majorité de ces logiciels possède une nouvelle interface graphique basée sur une bande de commandes nommée *Ruban*, dont les fonctions changent en fonction de l'onglet actif.

SharePoint devrait normalement incorporer plusieurs technologies de postes à postes en suivant notamment l'embauche du fondateur de la société Groove Networks, Inc. (créateur de Groove et de Lotus Notes) : Ray Ozzie[14]



Figure 18 : Logo du Microsoft Office 2007

5. Power Point :

Microsoft PowerPoint est un logiciel de présentation édité par Microsoft. Il fait partie de la suite Microsoft Office. Microsoft PowerPoint fonctionne sous Windows et Mac OS.

La version la plus récente est PowerPoint 2021¹. Elle fait partie de la suite Office 2021 et du service Microsoft 365. Depuis avril 2014, l'application PowerPoint est disponible

sur iPad, iPhone, Windows Phone et Android. Cela entre dans la logique « Office Mobile »² de Microsoft de rendre ses logiciels disponibles sur le plus grand nombre de supports (ordinateurs, tablettes, smartphones). PowerPoint est le programme de présentation le plus utilisé dans le monde.
[15]



Figure 19:Logo du Power point

3.2 Langage de programmation :

1. Java :

Java est un langage de programmation de haut niveau orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

La société Sun est rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.[16]



Figure 20 : Logo du java

3.3 Base de données :

1. MySQL :

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données[17].



Figure 21 : Logo du MySQL

2. PhpMyAdmin :

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL [18]



Figure 22 : Logo du phpMyAdmin

3. Xampp :

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

Il est distribué avec différentes bibliothèques logicielles qui élargissent la palette des services de façon notable : OpenSSL, Expat (analyseur syntaxique de fichiers XML), PNG, SQLite, zlib... ainsi que différents modules Perl et Tomcat.

Officiellement, XAMPP permet de configurer un serveur de test local avant la mise en œuvre d'un site Web, et son usage n'est pas recommandé pour un serveur dit *de production*². [19]



Figure 23 : Logo du XAMPP

4. JasperReports :

JasperReports est un outil de reporting *open source*, offert sous forme d'une bibliothèque qui peut être embarquée dans tous types d'applications Java [20].



Figure 24 : Logo du JasperReports

3.4 Présentation des interfaces :

1. Interface d'accueil :

1.1 Admin : Cette figure (figure 25) représente l'interface d'accueil de l'admin

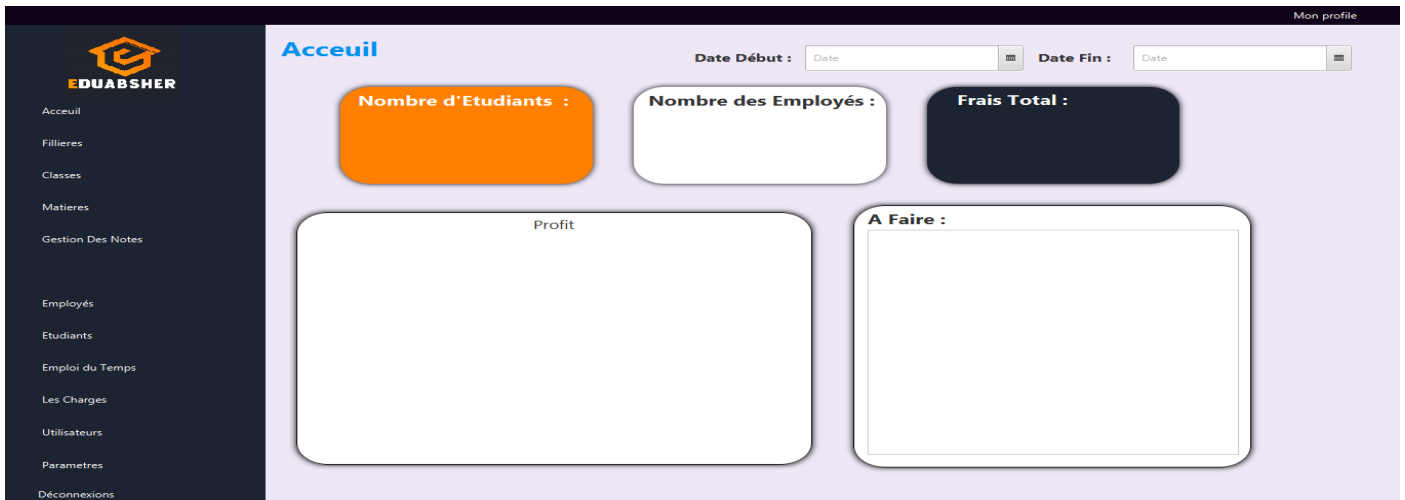


Figure 25 : Interface d'accueil « admin »

1.2 Enseignant : Cette figure (figure 26) représente l'interface d'accueil de l'enseignant



Figure 26 : Interface d'accueil « Enseignant »

2.Interface d'authentification : Cette figure (figure 27) représente l'interface d'authentification de l'admin et l'enseignant



Figure 27 : l'interface d'authentification

4. Interface de liste des élèves : Cette figure (figure 28) représente l'interface de liste des élèves , qui contient les différents tache

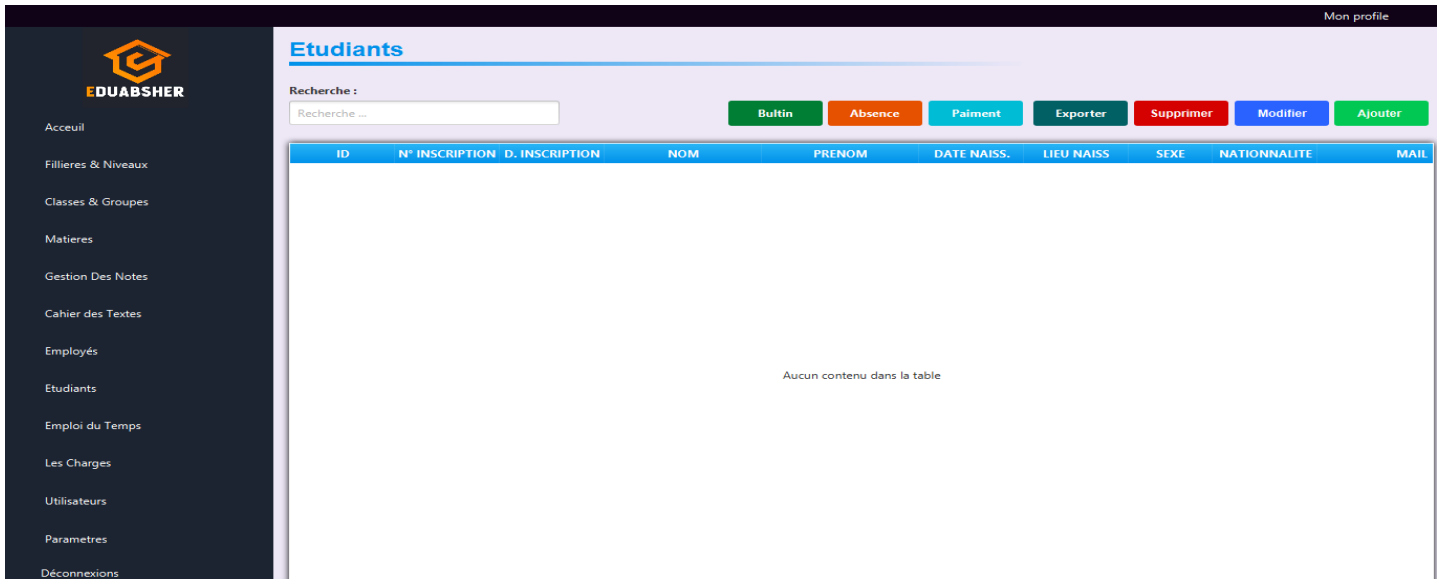


Figure 28 : interface de liste des élèves

5. Interface d'ajouter un employé :

Cette figure (figure 28) représente l'interface d'ajouter des employés .

The screenshot shows a form titled 'Ajouter un Employé'. The form contains the following fields and controls:

- Matricule :** A text input field.
- ID :** A text input field with placeholder 'identifiant ...'.
- Nationalité :** A text input field.
- Nom :** A text input field with placeholder 'Nom ...'.
- Prénom :** A text input field with placeholder 'Prénom ...'.
- Date Naiss. :** A date picker field.
- Lieu Naiss. :** A text input field with placeholder 'ville , Wilaya ...'.
- Sexe :** Radio buttons for 'Homme' and 'Femme'.
- Situation Familiale :** A dropdown menu.
- Adresse :** A text input field with placeholder 'ex : 15 City xvb'.
- Nature d'emploi :** A dropdown menu.
- Diplome :** A text input field.
- Matiere :** A text input field.
- Mail :** A text input field with placeholder 'xsn@example.com'.
- N° Tél. :** A text input field with placeholder '02 154876'.
- Date d'entrée :** A date picker field.
- Date Fin :** A date picker field.
- Salarie Horaire :** A text input field with placeholder '0.00' and a 'DZD' label.
- Photo :** A large empty rectangular area for a photo.
- Buttons:** 'Parcourir ...' (orange), 'Annuler' (red), and 'Enregistrer' (green).

Figure 29 : Interface d'ajouter un employé

6. Interfaces des enseignants :

6.1 Interface de groupe et classe : la figure ci-dessous représente l'interface des classes et groupes pour enseignants

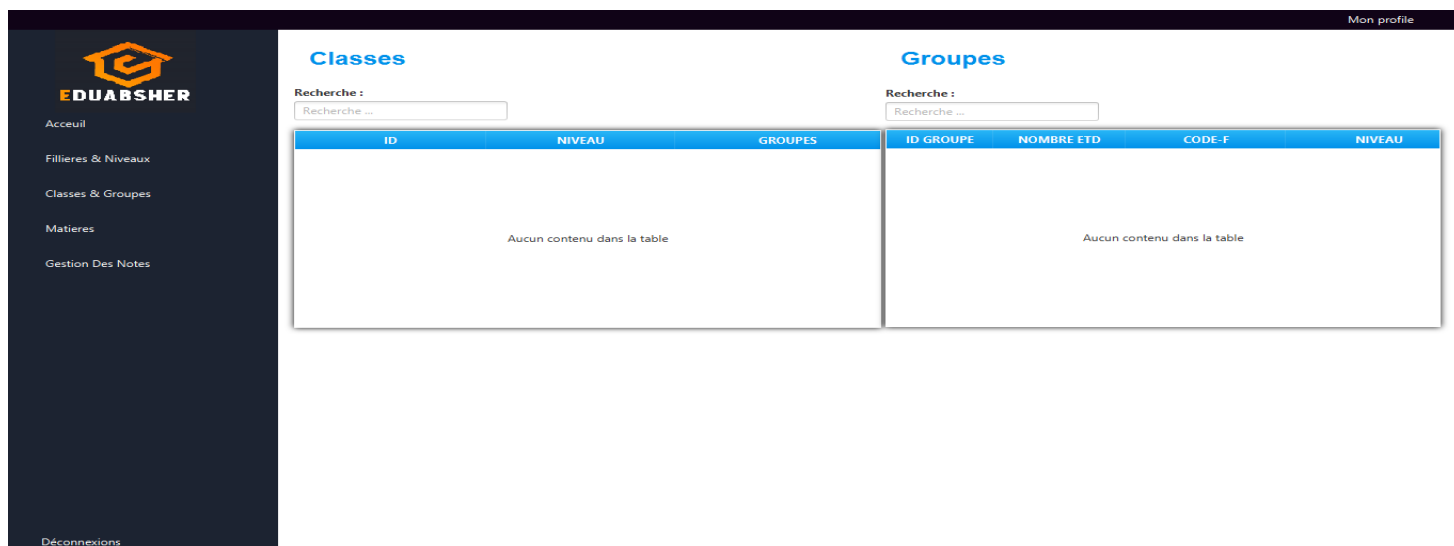


Figure 30 : Interface de groupe et classe

6.2 Interface de gestionnaire de note : la figure ci-dessous représente l'interface de gestionnaire de note

Figure 31 : Interface de gestionnaire de notes

3.5 Conclusion

Dans ce chapitre, nous avons justifié notre choix de langage et d'outils de développement ainsi que certaines interfaces de notre application. Notre but était d'assurer une meilleure compréhension du sujet et des fonctionnalités de notre système.

Conclusion générale

Ce projet nous a offert une expérience complète et une initiation approfondie au cycle de développement logiciel, couvrant toutes les phases, de la conception à la validation, en passant par les différentes étapes itératives de codage et de tests. De plus, il nous a permis d'acquérir des compétences dans la conception d'une base de données complète.

La réalisation de notre application est à présent achevée, elle comporte les fonctionnalités suivantes :

- Inscrire les élèves
- Gérer les horaires et des emplois du temps
- Enregistrer les notes et calculer la moyenne
- Gérer les absences
- Gérer les charges
- Gérer les frais de scolarité

Notre projet se concentre sur l'organisation de la gestion des écoles privées afin d'augmenter la fiabilité et de faciliter aux administratifs de gérer les fonctions principales pour les élèves et leurs enseignants

Nous avons utilisé la méthode RUP et le langage UML pour modéliser le système. Puis nous avons effectué l'implémentation de l'application en utilisant tous les outils nécessaires,

Ce projet nous a permis d'en savoir plus, d'approfondir nos connaissances et d'acquérir de l'expérience dans le domaine de la programmation.

Nous nous sommes focalisés pour délivrer une application desktop de qualité supérieur, pour les écoles privées, cependant nulle application est parfaite. Comme perspectives à ce travail, nous proposons de :

- Ajouter l'option du scanner pour scanner le dossier de l'élève pour éviter plus de papier.
- Pouvoir sauvegarder les images de las élèves dans la base de données pour faciliter les inscriptions.
- Générer le fonctionnement d'emploi du temps telle que l'admin peut vérifier les salles , les profs , et l'heure pour ne pas perdre du temps de remplir l'emplois manuellement.

Bibliographie :

- [1] H. Famié-Galtier, « Comment créer un cahier des charges : étapes à suivre et modèles à télécharger », *BDM*, 2 janvier 2023. <https://www.blogdumoderateur.com/cahier-des-charges-etapes-modeles/> (consulté le 15 juin 2023).
- [2] « définition ecole privée - Recherche Google ». Consulté le: 12 juin 2023. [En ligne]. Disponible sur: https://www.google.com/search?q=definition+ecole+priv%C3%A9e&rlz=1C1XXVF_frDZ1042DZ1042&oq=definition+ecole+priv%C3%A9e&aqs=chrome..69i57j0i22i30l3.27375j1j7&sourceid=chrome&ie=UTF-8
- [3] « Snapshot ». Consulté le: 12 juin 2023. [En ligne]. Disponible sur: <https://www.techno-science.net/definition/3743.html>
- [4] « définition uml wikipedia - Recherche Google ». https://www.google.com/search?q=definition+uml+wikipedia&rlz=1C1XXVF_frDZ1042DZ1042&sxsrf=APwXEd8vNe-doe6251fnwXB0ZAP8JTItg%3A1686169572436&ei=5OeAZPOUGr-J9u8PmdGewAg&oq=definition+uml+wi&gs_lcp=Cgxnd3Mtd2l6LXNlcnAQAxgAMgUIIRCgATIFCCEQoAEyBQghEKABMgUIIRCgATIICCEQFhAeEB06CggAEecQ1gQQsAM6CggAEIoFELADEEM6BggAEBYQHjoICAAQFhAeEApKBAhBGABQfliVB2C3GGgBcAF4AIABzAKIAYsFkgEHMC4yLjAuMZgBAKABAcABAcgBCg&sclient=gws-wiz-serp (consulté le 7 juin 2023).
- [5] « définition uml wikipedia - Recherche Google ». https://www.google.com/search?q=definition+uml+wikipedia&rlz=1C1XXVF_frDZ1042DZ1042&sxsrf=APwXEd8vNe-doe6251fnwXB0ZAP8JTItg%3A1686169572436&ei=5OeAZPOUGr-J9u8PmdGewAg&oq=definition+uml+wi&gs_lcp=Cgxnd3Mtd2l6LXNlcnAQAxgAMgUIIRCgATIFCCEQoAEyBQghEKABMgUIIRCgATIICCEQFhAeEB06CggAEecQ1gQQsAM6CggAEIoFELADEEM6BggAEBYQHjoICAAQFhAeEApKBAhBGABQfliVB2C3GGgBcAF4AIABzAKIAYsFkgEHMC4yLjAuMZgBAKABAcABAcgBCg&sclient=gws-wiz-serp (consulté le 7 juin 2023).
- [6] « FACULTE DES SCIENCES EXACTES ET DE L'INFORMATIQUE: Se connecter sur le site ». Consulté le: 6 juillet 2023. [En ligne]. Disponible sur: <https://e-fsei.univ-mosta.dz/login/index.php>
- [7] « Méthode RUP : une approche de gestion de projet | Nutcache ». <https://www.nutcache.com/fr/blog/methode-rup/> (consulté le 13 juin 2023).
- [8] « Définition des diagrammes de classes UML 1.5 — RAD Studio ». https://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_de_classes_UML_1.5 (consulté le 13 juin 2023).
- [9] « Diagramme des paquetages », *Wikipédia*. 28 décembre 2018. Consulté le: 13 juin 2023. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Diagramme_des_paquetages&oldid=155209216
- [10] « Les diagrammes de cas d'utilisation UML », *Lucidchart*. <https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml> (consulté le 7 juin 2023).
- [11] « Draw.io : un outil pour dessiner des diagrammes en ligne ». <https://www.tice-education.fr/tous-les-articles-er-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne> (consulté le 12 juin 2023).
- [12] « intellij definition - Recherche Google ». https://www.google.com/search?q=intellij+definition&source=lmns&bih=568&biw=1366&rlz=1C1XXVF_frDZ1042DZ1042&hl=fr&sa=X&ved=2ahUKEwip-IfZ9b3_AhUtvicCHbvIDgcQ_AUoAHoECAEQAA (consulté le 12 juin 2023).
- [13] « Informations sur le générateur de scène JavaFX ». <https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html> (consulté le 12 juin 2023).
- [14] « Microsoft Office 2007 », *Wikipédia*. 21 septembre 2022. Consulté le: 12 juin 2023. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Microsoft_Office_2007&oldid=197136745
- [15] « Microsoft PowerPoint », *Wikipédia*. 2 décembre 2022. Consulté le: 12 juin 2023. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Microsoft_PowerPoint&oldid=199157388
- [16] « Java (langage) — Wikipédia ». [https://fr.wikipedia.org/wiki/Java_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage)) (consulté le 13 juin 2023).
- [17] « Définition | MySQL | Futura Tech ». <https://www.futura-sciences.com/tech/definitions/internet-mysql-4640/> (consulté le 12 juin 2023).
- [18] « phpMyAdmin — Wikipédia ». <https://fr.wikipedia.org/wiki/PhpMyAdmin> (consulté le 12 juin 2023).

- [19] « XAMPP », *Wikipédia*. 13 janvier 2023. Consulté le: 12 juin 2023. [En ligne]. Disponible sur:
<https://fr.wikipedia.org/w/index.php?title=XAMPP&oldid=200395821>
- [20] « JasperReports », *Wikipédia*. 28 janvier 2021. Consulté le: 13 juin 2023. [En ligne]. Disponible sur:
<https://fr.wikipedia.org/w/index.php?title=JasperReports&oldid=179320385>

Annexe :

1. Les diagrammes de séquence :

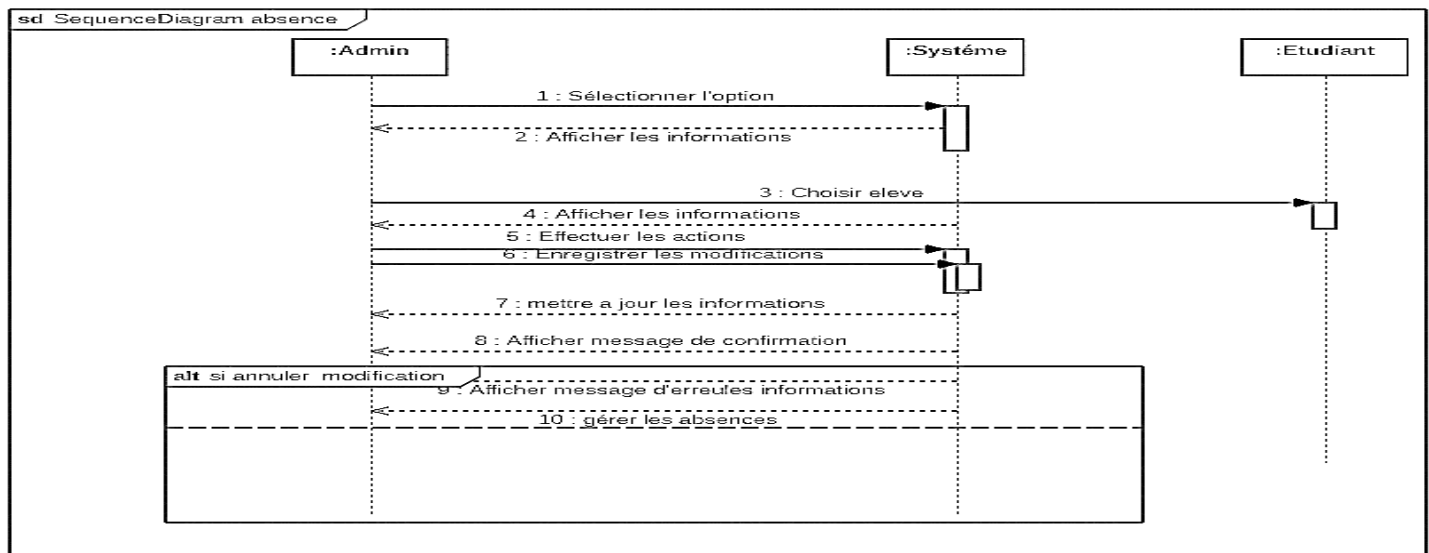


Figure 32: diagramme de séquence « Absence »

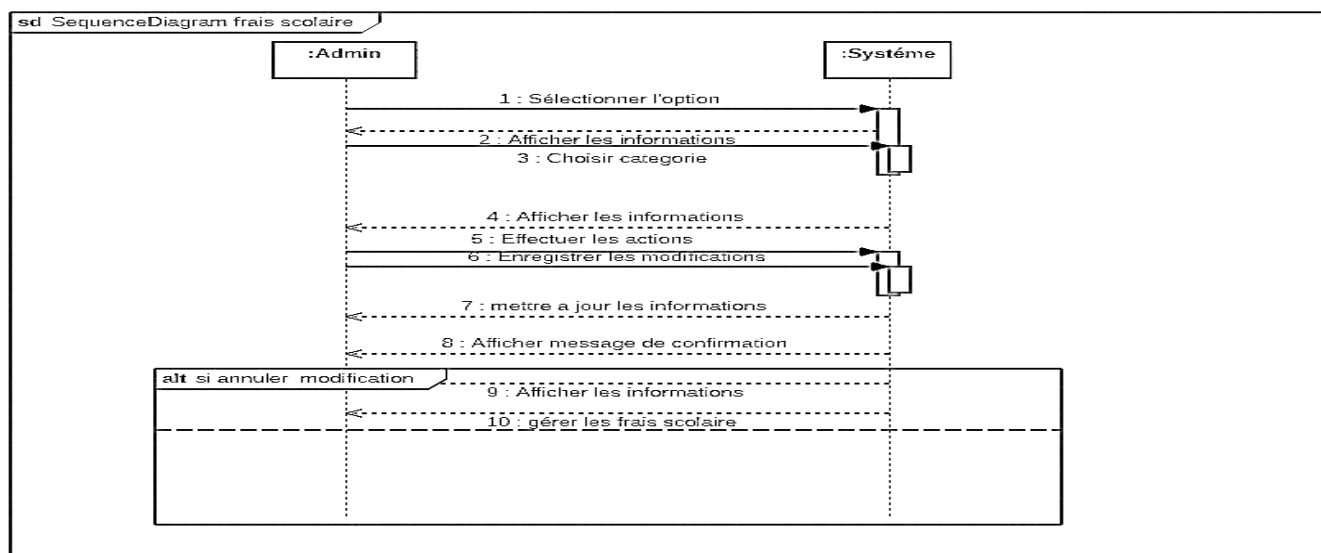


Figure 33: Diagramme de séquence « charges »

2. Les interfaces : Voici d'autre interface de notre application.

Figure 34 :Interface d'ajouter classes et groupes.

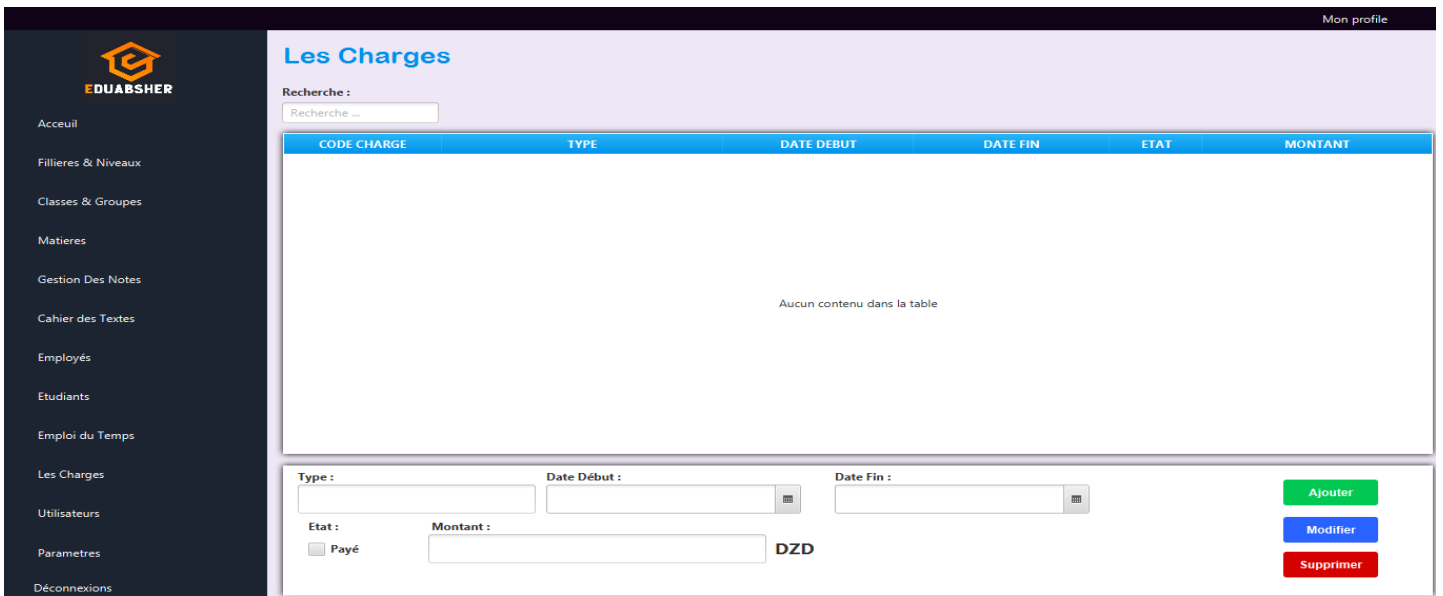
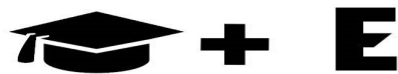


Figure35 :Interface des charges

LOGO EDUMANAGE



Concept



Color - Palette



Figure 36: Logo de notre application