

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

Université Assane Seck de Ziguinchor

UFR Sciences et Technologies

Département Informatique



Mémoire de fin d'études

Pour l'obtention du diplôme de Master

Mention : Conduite de Projet

Spécialité : Génie Logiciel

Sujet

Conception et développement d'une application web nommée Nema2SchoolApp, pour la gestion des notes scolaires pour le collège d'enseignement moyen Néma 2 de Ziguinchor

Présenté par :

M. Ibrahima Gnamo SONKO

Soutenance le 10/05/2021

Membres du jury

- Pr. Youssou FAYE (**Président de jury**)
- Dr. Papa Alioune CISSE (**Examineur**)
- Dr. Mouhamadou GAYE (**Examineur**)
- Dr. Ibrahima DIOP (**Encadreur**)

Sous la direction de :

- Dr. Ibrahima DIOP

Sous la supervision de :

- Pr. Youssou FAYE

Année Universitaire 2019 - 2020

Résumé

L'année scolaire 2102-2013 fut un tournant au CEM Néma 2 de Ziguinchor, car elle fut le début de l'automatisation du processus de gestion des notes des élèves avec un logiciel nommé : Best Grade. Best Grade est une application Excel et VBA, fruit d'un projet du même nom.

Au fil du temps, son utilisation a révélé beaucoup de faiblesses, nous pouvons citer : la répétition annuelle de certaines tâches comme la saisie des données d'identification des élèves et le « copier-coller » pour toutes les pages des professeurs, la quasi inexistence de sécurité avec la détention de tous les mots de passe des professeurs par l'administrateur. Il y a aussi le risque élevé de perte de données du fait de l'externalisation des pages des professeurs (**Annexe 2 et 3**), la perte de temps dans le traitement des notes due à la répétition annuelle de certaines tâches et à la lenteur dans le traitement des pages des professeurs, la centralisation de tout le travail autour de l'administrateur car le fichier central n'est utilisé que par ce dernier, etc.

Des applications fonctionnelles pouvant pallier les manquements ou les faiblesses de Best Grade cités ci-dessus existent. Les unes : Sama-Ecole, MIRACLE, Daarasytem et SUNU-ECOLE, etc. sont payantes, l'acquisition d'une de ces d'elles impacterait négativement sur le maigre budget de l'établissement. L'autre, PLANETE, développée par le ministère de l'éducation nationale du Sénégal et fonctionnelle depuis 2014, malgré sa gratuité souffre : de bogues répétitifs du système, de surcharge du serveur, de fermetures répétées de la plateforme pour maintenance, de la quasi centralisation de son administration, de la privatisation de son code source, etc.

Au vu des éléments précités, nous nous sommes fixés comme objectif de mettre en place une application web responsive et sécurisée permettant de faire intervenir beaucoup plus d'utilisateurs dans la gestion des notes.

Afin de faire intervenir tout le monde : équipe de développement, représentants des utilisateurs finaux et propriétaire du projet tout le long du processus de développement, nous avons utilisé une des méthodologies agiles, en l'occurrence la méthodologie Scrum.

UML nous a permis de modéliser la spécification des besoins. Pour la structuration et le stockage des données, nous avons utilisé le SGBDR MySQL. Java est utilisé pour l'implémentation afin d'avoir une application simple à utiliser, sécurisée et centralisant les données. L'utilisation du Framework Spring, nous a facilité son implémentation.

Remerciements

Nous remercions tout d'abord Dieu, pour nous avoir donné la force, le courage et l'endurance depuis le tout début de notre périple scolaire puis universitaire jusqu'à l'accomplissement de ce travail. En lui demandant de nous en procurer davantage pour la suite de notre cursus.

Au Dr. Ibrahima Diop nous vous disons grand merci pour tous les sacrifices que vous avez accepté de consentir afin que ce travail soit le meilleur possible. La disponibilité, l'écoute, les critiques positives, la pertinence de vos idées pendant les séances de partage, sont parmi tant d'autres ceux que nous qualifions de sacrifice. Car rien ne vous obligeait à nous accepter dans votre équipe d'encadrement. Nous vous disons encore une fois merci.

Nous tenons à remercier tous les membres de notre jury, nous voulons nommer : le Pr. Youssou Faye pour avoir accepté de présider ce jury, le Dr. Papa Alioune Cissé et le Dr. Mouhamadou Gaye pour avoir examiné ce travail.

A tous les enseignants des départements de mathématique et de physique en général et ceux du département d'informatique en particulier, nous vous disons merci pour toutes les compétences que vous nous avez permis d'acquérir de la licence 1 à nos jours.

Nous tenons à remercier nos camarades de promotion pour l'esprit de partage, les conseils, les suggestions et l'entraide qui vous ont toujours animé. Nous ne sommes pas des camarades de promotion, mais une famille que l'amour des sciences et technologies a pu fonder.

A monsieur Moussa Ndaw du CEM Soucoupapaye, à monsieur Moise Malou du CEMT Amílcar Cabral et à monsieur le Principal Sy du CEM Malick Fall, nous vous transmettons nos remerciements pour votre disponibilité dans le cadre de notre enquête sur l'application PLANETE du ministère de l'éducation nationale.

Nous ne pouvons pas oublier dans nos remerciements l'ensemble du corps professoral et du personnel administratif du CEM Néma 2 de Ziguinchor.

En fin nous tenons à remercier toute personne qui de près ou de loin, a contribué à l'accomplissement de ce travail et à notre réussite dans notre quête du savoir.

Dédicaces

*A notre très chère maman, **Mansata Diakhité**, nous te dédions ce travail pour nous avoir donné la vie, pour nous avoir inculqué les bonnes valeurs, tu es notre premier mur des lamentations.*

*A notre cher défunt père, **Ousmane Sonko**, nous te disons que nous y sommes presque. Nous vous dédions ce mémoire à titre posthume.*

*Aux **deux coépouses** de notre maman, nous vous dédions ce mémoire.*

*A **madame Sonko** née **Gnima Sané**, nous te disons mention spéciale à toi pour ta compréhension et ton aide. Car nous sommes rarement à la maison pendant la journée et le soir l'ordinateur occupe tout notre temps jusque tard dans la nuit.*

A tous les frères et sœurs, nous vous disons merci pour votre amour et vos encouragements.

A tous les amis d'hier et d'aujourd'hui, pour les prières, les encouragements et la relation sincère qui nous lie.

*A notre ami et frère, qui serait sans doute très content de nous voir franchir cette étape avec bravoure. Nous te dédions spécialement ce travail, nous prions et nous continuerons de prier pour le repos de ton âme. **Yaya Mabo Sadio** que le paradis céleste soit ta demeure éternelle.*

Table des matières

INTRODUCTION	1
CHAPITRE I : PRESENTATION DU SUJET DE MEMOIRE	5
I- Présentation du CEM Néma 2	5
II- Le système de gestion de notes existant	6
II.1- Les acteurs	6
II.2- Description du processus actuel de gestion des notes	7
II.3- Problèmes liés à la gestion des notes avec Best Grade	9
III- Problématique du sujet	11
III.1- Solution liée au problème	12
III.2- Les objectifs	14
IV- Processus de développement de l'application	15
IV.1- Introduction aux méthodes agiles	15
IV.2- La méthode SCRUM	15
IV.3- Adaptation de la méthodologie Scrum à notre contexte	18
CHAPITRE II : SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS 21	
I- Spécification des besoins fonctionnels	21
I.1- Identification des acteurs du système	21
I.2- Identification des fonctionnalités du système	22
I.3- Diagrammes des cas d'utilisation	24
II- Analyse des besoins fonctionnels	29
II.1- Analyse de l'authentification	30
II.2- Analyse de l'archivage d'un agent (utilisateur)	32
II.3- Analyse du cas affecter une classe et une matière à un professeur	34
II.4- Analyse du cas d'utilisation ajouter des notes pour un semestre	36
II.5- Analyse du cas d'utilisation orienter un élève dans une classe	39

CHAPITRE III : CONCEPTION DU SYTEME	42
I- Conception générale	42
I.1- Architecture de l'application	42
I.2- Diagrammes de composants	44
I.3- Diagramme de packages	45
I.4- Diagramme de déploiement	46
II- Conception détaillée	47
II.1- Diagramme de classes	47
II.1.5- Diagramme de classes participantes aux fonctionnalités de la gestion des affectations de classes et matières aux professeurs	50
II.2- Dictionnaire de données	51
CHAPITRE IV : IMPLEMENTATION DE L'APPLICATION	55
I- Outils et technologies utilisées	55
I.1- Outils de conception : StarUML	55
I.2- Outils de développement : Eclipse	55
I.3- Le SGBDR : MySQL	55
I.4- Framework Bootstrap	56
I.5- Langages de programmation : Java et JavaScript	56
I.6- Langage de balisage : HTML	56
I.7- Outil de création de rapports (Jaspersoft Studio)	57
I.8- Framework Spring (Spring Boot)	57
II- Implémentation	58
II.1- Implémentation de la base de données	58
II.2- implémentation de l'application	60
CHAPITRE V : PRESENTATION DE L'APPLICATION	65
I- Vue globale de l'application	65
II- Page d'accueil	65

III- Page d'authentification	66
IV- Page avec le menu principal	67
V- Tableau de bord de l'administrateur	67
VI- Formulaire d'ajout d'un agent (utilisateur)	68
.....	68
VII- Liste des agents	69
VIII- Formulaire d'ajout d'un élève	69
IX- Liste des élèves et formulaire de recherche par nom	70
X- Formulaire d'affectation d'une classe à un surveillant	71
XI- Liste des élèves à orienter	72
XII- Formulaire d'insertion de notes	73
XIII- Modification de notes	74
XIV- Affectation d'une classe et d'une matière	75
XV- Impression de listes	76
XVI- Impression d'un bulletin de notes	77
CONCLUSION ET PERSPECTIVES	80
BIBLIOGRAPHIE	82
WEBOGRAPHIE	82
ANNEXE	84

Liste de figures

Figure 1: Organigramme du CEM Nema 2 de Ziguinchor.....	6
Figure 2: Fonctionnement du Scrum [1]	17
Figure 3: Diagramme de cas d'utilisation pour l'Administrateur.....	25
Figure 4: Diagramme de cas d'utilisation pour le Principal	26
Figure 5: Diagramme de cas d'utilisation pour le Professeur	27
Figure 6: Diagramme de cas d'utilisation pour le Surveillant Général.....	28
Figure 7: Diagramme de cas d'utilisation pour le Surveillant	29
Figure 8: Diagramme d'activité du cas "s'authentifier"	31
Figure 9: Diagramme de séquences du cas "s'authentifier"	31
Figure 10: Diagramme d'activité du cas "Archiver un agent"	33
Figure 11: Diagramme de séquences du cas "Archiver un agent"	34
Figure 12: Diagramme d'activité du cas "Affecter classe-matière".....	35
Figure 13: Diagramme de séquences du cas "Affecter classe-matière"	36
Figure 14: Diagramme d'activité du cas "Ajouter notes"	38
Figure 15: Diagramme de séquences du cas "Ajouter notes".....	39
Figure 16: Diagramme d'activité du cas "Orienter"	40
Figure 17: Diagramme de séquences du cas "Orienter"	41
Figure 18: Architecture de l'application	43
Figure 19: Diagramme de composants	44
Figure 20: Diagramme de package.....	45
Figure 21: Diagramme de déploiement	46
Figure 22: Diagramme de classes participantes aux fonctionnalités de la gestion des utilisateurs et des accès	47
Figure 23: Diagramme de classes participantes aux fonctionnalités de la gestion des affectations de classes aux surveillants	48
Figure 24: Diagramme de classes participantes aux fonctionnalités de la gestion des absences	49
Figure 25: Diagramme de classes participantes aux fonctionnalités de la gestion des orientations	49
Figure 26: Diagramme de classes participantes aux fonctionnalités de la gestion des affectations de classes et matières aux professeurs	50
Figure 27: Diagramme de classes participantes aux fonctionnalités de la gestion des notes..	51

Figure 28: Schéma relationnel du MPD	59
Figure 29: Fichier application.properties	59
Figure 30: Architecture de l'application	61
Figure 31: Entité "Agent"	62
Figure 32: Interface "AgentRepository"	63
Figure 33: Contrôleur "AgentController"	63
Figure 34: Fichier HTML du formulaire d'ajout d'un agent	64
Figure 35: Vue globale de l'application.....	65
Figure 36: Page d'accueil de l'application	66
Figure 37: Page d'authentification.....	66
Figure 38: Page avec menu principal	67
Figure 39: Tableau de bord de l'administrateur.....	68
Figure 40: Interface avec le formulaire d'ajout d'un agent	68
Figure 41: Liste paginée des agents (utilisateurs)	69
Figure 42: Formulaire d'ajout d'un élève.....	70
Figure 43: Liste paginée des élèves.....	71
Figure 44: Formulaire de recherche d'un élève par nom	71
Figure 45: Formulaire d'affectation d'une classe à un surveillant	72
Figure 46: Liste paginée des élèves à orienter et formulaire d'orientation.....	72
Figure 47: Liste élèves par classe pour insertion de notes	73
Figure 48: Formulaire d'insertion de notes.....	74
Figure 49: Liste des élèves d'une classe pour modification	74
Figure 50: Formulaire de modification.....	75
Figure 51: Formulaire d'affectation classe-matière à un professeur.....	76
Figure 52: Interface avec les formulaires d'impression de listes par classe	76
Figure 53: Exemple de liste détaillée d'une classe générée.....	77
Figure 54: Interface d'impression d'un bulletin de notes.....	78
Figure 55: Exemple de bulletin généré.....	79
Figure 56: Interface d'administration de Best Grade.....	84
Figure 57: Page d'un professeur dans Best Grade	85
Figure 58: Page d'insertion de notes dans Best Grade.....	86
Figure 59: Fiche de notes et/ou liste d'appel	86
Figure 60: Bulletin du 2nd semestre.....	87

Liste des tableaux

Tableau 1: Les étapes du système de gestion des notes avec Best Grade.....	7
Tableau 2: Les rôles dans SCRUM.....	16
Tableau 3: Extrait du Product Backlog	18
Tableau 4: Identification des acteurs.....	21
Tableau 5: Liste des fonctionnalités du système.....	22
Tableau 6: Description du cas d'utilisation "S'authentifier"	30
Tableau 7: Description du cas d'utilisation "Archiver un agent"	32
Tableau 8: Description du cas d'utilisation "Affecter classe-matière"	34
Tableau 9: Description du cas d'utilisation « Ajouter notes »	36
Tableau 10: Description du cas d'utilisation "Orienter"	39
Tableau 11: Dictionnaire de données	51

Liste des abréviations

CEM : Collège d'Enseignement Moyen

PLANETE : Paquet de Logiciels Académiques Normalisés pour les Etablissements et Ecole

VBA : Visual Basic pour Application

SCRUM : Scrum est un Framework ou cadre de développement de produits logiciels complexes

UML : Unified Modeling Language (Langage de Modélisation Unifié en français)

UASZ : Université Assane Seck de Ziguinchor

UFR : Unité de Formation et de Recherche

SG : Surveillant Général

Ext : tâche qui est exécutée sans utilisation de Best Grade

MVC : Modèle-Vue-Contrôleur

SGBDR : Système de Gestion de Base de Données Relationnelles

SQL : Structured Query Language

INTRODUCTION

Entre 1990 et 2010, l'effectif de scolarisés au Sénégal a connu une augmentation fulgurante avec l'augmentation de la population nationale et les nombreux programmes et projets dans le domaine de l'éducation. Les structures d'accueil et le personnel enseignant n'ont pas pu suivre malgré les efforts de l'Etat dans ce sens. La conséquence est la pléthore d'apprenants dans les établissements d'enseignement publics et privés.

Le CEM Néma 2 de Ziguinchor depuis sa création en octobre 2008, ne cesse de recevoir chaque année, un nombre d'élèves de plus en plus élevé.

Au démarrage, en 2008, le CEM Néma 2 comptait environ deux cents (200) élèves répartis dans quatre (4) classes de sixième, un (1) chef d'établissement, un (1) surveillant et sept (7) professeurs. Aujourd'hui avec ses treize (13) classes pédagogiques, le collège compte six cent seize (616) élèves, quatre (4) surveillants, un (1) surveillant général et vingt-cinq (25) professeurs.

Le nombre important d'élèves comparé au nombre de membres du personnel administratif et pédagogique chargés de gérer le processus de gestion des notes des élèves, nous permet de mesurer la difficulté de la tâche et la probabilité élevée de faire des erreurs si un tel processus reste toujours manuel.

Fort heureusement en 2012-2013, avec la dotation d'un ordinateur fixe et l'achat d'une imprimante, le processus de gestion des notes au CEM Néma 2, a connu un tournant avec l'utilisation de Best Grade qui est une application faite avec Excel et VBA, fruit d'un projet du même nom.

La mise en pratique de Best Grade, nous a permis de déceler beaucoup de manquements, comme :

- la répétition annuelle de certaines tâches comme la saisie des données d'identification des élèves et le « copier-coller » pour toutes les matières des pages des professeurs afin de transférer les notes saisies dans ces dernières vers le fichier central ;
- la quasi inexistence d'une sécurité permettant de protéger les notes saisies par les professeurs dans leur page. Ce manque de sécurité est dû au fait que l'administrateur est celui qui attribue les mots de passe et les conserve pour

pouvoir accéder à toutes les pages des professeurs afin de transférer les notes vers le fichier central ;

- le risque élevé de perte de données lié à l'externalisation des pages des professeurs (**Annexe 2 et 3**) ;
- la perte de temps dans le traitement des notes due à la répétition annuelle de certaines tâches voir **Tableau 1** et la lenteur dans le traitement des pages des professeurs ;
- la centralisation de tous le travail autour de l'administrateur car le fichier centrale n'est utilisé que par ce dernier. En effet il est chargé de l'installation de Best Grade, de sa configuration, de la saisie des données d'identification des élèves et des noms des professeurs, de l'externalisation des pages des professeurs et du transfert des notes des pages des professeurs vers le fichier central ;
- etc.

Des solutions pouvant pallier les manquements de Best Grade existent, mais la plupart sont payantes. Parmi ces dernières, nous pouvons citer Sama-Ecole, MIRACLE, DaaraSystem et SUNU-ECOLE, entre autres.

La plateforme web PLANETE, lancée en 2014 par le ministère de l'éducation nationale et qui est destinée à la gestion des établissements sénégalais du public et du privée, malgré sa gratuité n'a pas été utilisé au CEM car l'établissement ne disposait pas d'une connexion à internet. Le CEM Néma 2 faisait partie des établissements pilotes qui devaient utiliser cette plateforme après une formation. De plus, nous avons mené une enquête dans quelques établissements de la commune de Ziguinchor qui l'utilisent ou qui ont eu à l'utiliser, qui nous a permis de recueillir des faiblesses de cette plateforme. Parmi lesquelles, nous avons :

- les bogues du système ;
- les fermetures répétées de la plateforme pour maintenance ;
- la lenteur des réponses du serveur ;
- les problèmes de mise à jour;
- le code non open source ne permettant pas de modification ;
- la lenteur dans les interventions quand des soucis se présentent;
- etc.

Pour pallier à tous ces problèmes empêchant le CEM Néma 2 de gérer la saisie, le stockage et la génération des notes de ses élèves de façon conviviale, efficace et localement

avec les moyens disponibles, nous nous sommes proposés de travailler à la *conception et au développement d'une application web pour la gestion des notes scolaires pour le CEM Néma 2 de Ziguinchor*.

Ainsi, le système développé et qui va être déployé courant cette année scolaire va permettre de :

- ✓ Gérer les informations sur l'établissement ;
- ✓ Gérer les utilisateurs ;
- ✓ Gérer les années scolaire ;
- ✓ Gérer les semestres ;
- ✓ Gérer les classes pédagogiques ;
- ✓ Gérer les matières ;
- ✓ Gérer les affectations de classes et matières aux professeurs ;
- ✓ Gérer les affectations de classes aux surveillants simples ;
- ✓ Gérer les données d'identification des élèves ;
- ✓ Gérer les absences ;
- ✓ Gérer la saisie des notes ;
- ✓ Générer des bulletins ;
- ✓ Gérer les orientations annuelles des élèves ;
- ✓ Editer et imprimer des documents (listes détaillées, listes d'appel, fiches de notes, bulletins de notes).

Cinq (5) chapitres composent ce rapport de mémoire :

- **Chapitre I : Présentation du sujet de mémoire**, dans lequel nous allons déterminer les contours du sujet ainsi que les éléments qui traitent de la pertinence de notre sujet. Nous le terminerons par les détails de la méthodologie SCRUM, utilisée dans notre travail comme processus de développement de la solution.
- **Chapitre II : Spécification et analyse des besoins fonctionnels**, qui permettra d'identifier d'abord les acteurs et les fonctionnalités du système. Nous y présenterons aussi les diagrammes UML (cas d'utilisation, de séquences et d'activités).
- **Chapitre III : Conception du système**, qui est une description de la façon la plus claire possible avec des diagrammes UML du fonctionnement du futur système. Une partie de cette conception est générale et une autre détaillée.

- **Chapitre IV : Implémentation de l'application**, dans ce chapitre, nous reviendrons sur les outils et technologies utilisés, sur l'implémentation de la base de données et sur celle de l'application
- **Chapitre V : Présentation de l'application**, nous y ferons une présentation de l'application.

A la fin de ce mémoire nous proposons une conclusion et des perspectives d'amélioration de l'application.

CHAPITRE I : PRESENTATION DU SUJET DE MEMOIRE

Ce chapitre va nous permettre de poser la problématique de notre sujet de mémoire, de justifier le choix porté sur notre solution et la méthodologie utilisée dans notre travail. Pour ce faire, nous présenterons d'abord la structure pour laquelle cette solution est destinée. Ensuite nous exposerons la manière dont le processus de gestion des notes des élèves se déroule actuellement et sur la problématique du sujet. Nous terminerons par la méthodologie Scrum.

I- Présentation du CEM Néma 2

Créé en octobre 2008, le CEM Néma 2 appartient à l'Inspection d'Académie (IA) de Ziguinchor et plus précisément à l'Inspection de l'Education et de la Formation (IEF) de Ziguinchor. Il est situé dans la commune de Ziguinchor au quartier Château d'eau à côté de la route de l'UASZ entre l'école primaire Amath Tidiane Beye et le collège privé ACAPES.

Au CEM Néma 2, comme dans la plupart des écoles de ce type, il y a :

- **Le Principal** : qui est le chef de l'établissement, le supérieur hiérarchique direct de tous les agents ;
- **L'adjoint au Principal** : qui seconde le Principal et le remplace en cas d'absence ou d'incapacité provisoire. Il est du corps professoral, il n'est pas dispensé ;
- **Les professeurs** : ils sont les plus nombreux parmi le personnel, ils sont chargés des enseignements-apprentissages ;
- **Le Surveillant général** : il coordonne le travail des surveillants simples et est le trait d'union entre le Principal et ces derniers ;
- **Les surveillants** : ils sont les responsables des classes pédagogiques que le Surveillant général les a octroyés pour une année scolaire donnée ;
- **Le Gestionnaire** : il est un surveillant proposé par le chef de l'établissement et entériné par l'autorité ;
- **La bibliothécaire** : elle est surveillante et sa responsabilité est la gestion de la bibliothèque, à certaines occasions elle appuie les autres surveillantes dans leurs tâches ;
- En fin on a **les élèves** : qui sont au centre du processus d'enseignement-apprentissage.

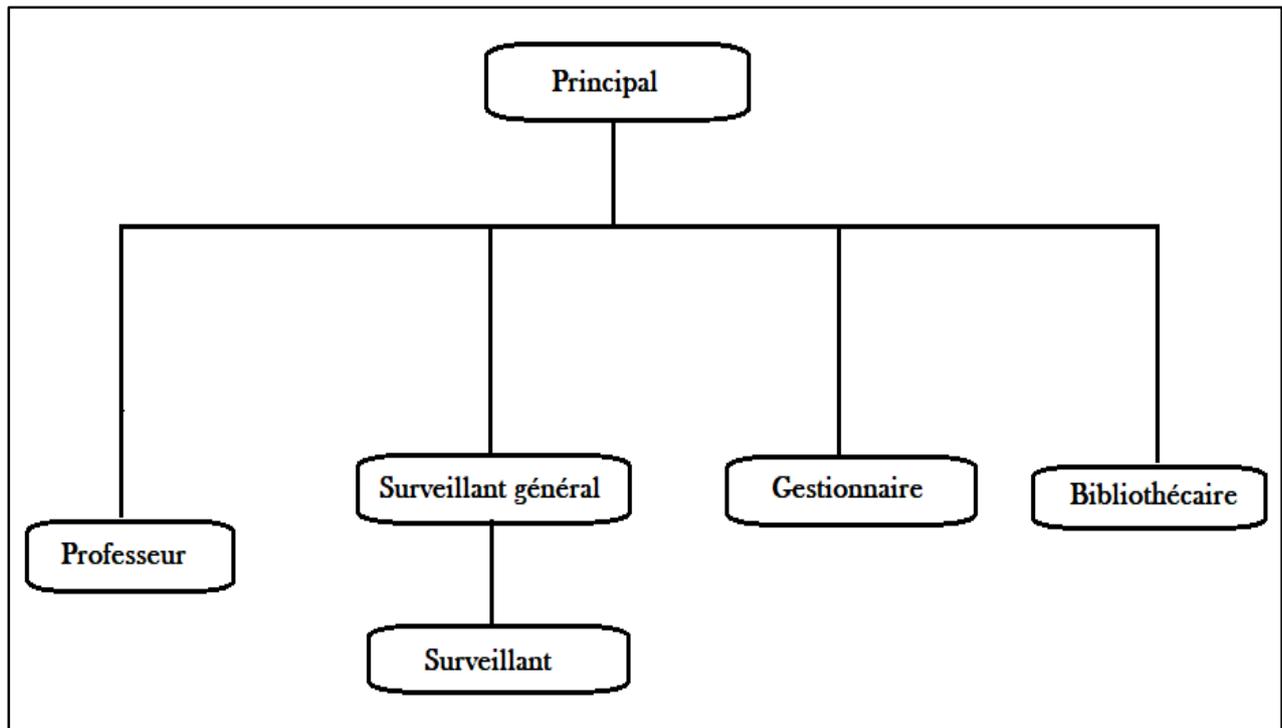


Figure 1: Organigramme du CEM Néma 2 de Ziguinchor

Des structures annexes existent, elles participent dans la vie et le bon fonctionnement de l'établissement. Nous pouvons citer :

- **Le Conseil de Gestion (CGE)** dans lequel toutes les composantes de la communauté éducative sont représentées (Principal, Surveillant général, le gestionnaire, le/la représentant(e) des autorités locales, les représentants des parents d'élèves, les représentants du personnel enseignant, les représentants du personnel de surveillance et les représentants des élèves) ;
- **Le gouvernement scolaire** ancien Foyer Socio-Educatif (FSE) constitué d'élèves et encadrés par un ou plusieurs professeurs et/ou surveillants ;
- **Les cellules pédagogiques** qui existent pour chaque matière. Elles sont uniquement composées de personnel enseignant ;

II- Le système de gestion de notes existant

II.1- Les acteurs

Tous les acteurs figurant dans l'organigramme ci-dessus n'interviennent pas dans le système de gestion des notes au sein du CEM Néma 2. Par exemple le gestionnaire et la bibliothécaire n'interviennent à aucune étape du processus. Le Principal, le Surveillant Général

et les surveillants n'interagissent pas directement avec l'application, ils ne sont donc pas considérés comme des acteurs du système.

Le système de gestion de note actuel du CEM Néma 2 compte que deux acteurs principaux :

- Les **professeurs** : ils enseignent et évaluent les apprenants afin de pouvoir noter leurs productions dans les différentes matières enseignées, ils sont chargés de la saisie des notes dans leur page respective ;
- L'**Administrateur** du système qui est un professeur de l'établissement, est l'acteur principal de ce système. C'est à lui d'installer, de configurer, de saisir les données d'identification des élèves et les noms des professeurs, de générer les pages des professeurs et de transférer les notes saisies par les professeurs de leurs pages vers le fichier central de l'application.

II.2- Description du processus actuel de gestion des notes

Dans cette partie de notre travail, nous allons décrire le processus actuel de la gestion des notes. Pour ce faire nous allons utiliser un tableau avec une colonne pour les étapes (elles sont numérotées), une deuxième pour les tâches dans chaque étape et une troisième pour les acteurs qui exécutent les tâches.

Tableau 1: Les étapes du système de gestion des notes avec Best Grade

Etape	Tâche	Acteur
Etape 1	Installation et configuration de Best Grade	Administrateur
Etape 2	Modification des informations de l'établissement et de l'année scolaire (voir Annexe 1)	Administrateur
	Saisie des noms des classes (voir Annexe 1)	Administrateur
	Choisir la formule de calcul de la moyenne (voir Annexe 1)	Administrateur
Etape 3	Insertion des matières et de leur coefficient	Administrateur
Etape 4	Insertion des données d'identification des élèves dans les classes où ils sont orientés	Administrateur
Etape 5	Impression des listes détaillées et des fiches de notes provisoires du 1 ^{er} semestre	Administrateur
Etape 6	Stabilisation des données d'identification des élèves	Administrateur

Etape 7	Impression des listes détaillées et des fiches de notes définitives du 1 ^{er} semestre	Administrateur
Etape 8	Création des pages des professeurs (ce sont des fichiers Excel externes qui sont générés pour chaque professeur) (voir Annexe 2)	Administrateur
Etape 9	Pour chaque page de professeur : <ul style="list-style-type: none"> - Saisir son prénom et son nom ; - Sélection dans la liste déroulante une classe détenue ; - Saisir dans le champ correspondant la matière enseignée dans cette classe ; - Enregistrer le fichier avec un mot de passe de lecture et écriture. (Ext) 	Administrateur
Etape 10	Remplissage des notes pour chaque professeur dans sa page, il doit au préalable renseigner son mot de passe deux fois pour la lecture et l'écriture (Ext) (voir Annexe 3)	Professeur
Etape 11	Pour chaque page de professeur, ouvrir puis copier les notes et les coller dans Best Grade. A faire pour toutes les matières du professeur. Après avoir collé il faut cliquer sur le bouton TH (tableau d'honneur) pour cocher ceux qui ont le TH dans la matière.	Administrateur
Etape 12	Saisir le nombre d'absences semestriel pour chaque élève	Administrateur
Etape 13	Imprimer pour chaque classe la liste de classement et les bulletins du 1 ^{er} semestre des élèves	Administrateur
Etape 14	Saisir les observations du conseil pour chaque élève pendant le conseil. C'est pour avoir les mêmes observations quand un élève va demander un duplicata de son bulletin	Administrateur
Etape 15	Supprimer les élèves omis lors de la stabilisation du 1 ^{er} semestre et modification de données d'identification d'un élève s'il y en a dans Best Grade	Administrateur
Etape 16	Impression des listes détaillées et des fiches de notes du 2 nd semestre	Administrateur

Etape 17	Remplissage des notes pour chaque professeur dans sa page, il doit au préalable renseigner son mot de passe deux fois pour la lecture et l'écriture (Ext)	Professeur
Etape 18	Pour chaque page de professeur, ouvrir puis copier les notes et les coller dans Best Grade. A faire pour toutes les matières du professeur. Après avoir collé il faut cliquer sur le bouton TH (tableau d'honneur) pour cocher ceux qui ont le TH dans la matière.	Administrateur
Etape 19	Imprimer pour chaque classe la liste de classement et les bulletins du 2 nd semestre des élèves	Administrateur
Etape 20	Saisir les observations du conseil pour chaque élève pendant le conseil	Administrateur
Etape 21	Imprimer les dossiers scolaires pour les agraffer dans les livrets de baccalauréat des élèves	Administrateur

II.3- Problèmes liés à la gestion des notes avec Best Grade

Best Grade est une application Excel et VBA contenant des macros afin d'automatiser certaines actions avec des boutons. La macro est une suite d'instructions que nous pouvons exécuter en cliquant sur un bouton appelant la macro par son nom. Les limites d'Excel sont aussi celles de Best Grade.

Au fil du temps, son utilisation dans la gestion des notes au CEM Néma 2, a permis de déceler ces problèmes. Nous pouvons citer :

- **niveau de sécurité faible** : avec Best Grade la sécurité est basée sur les mots de passe (lecture, écriture) des fichiers Excel externes (pages des professeurs) et sur une confiance mutuelle entre l'administrateur et les professeurs. Car ce dernier octroie les mots de passe et peut accéder à toutes les pages des professeurs et y modifier des notes déjà saisies et enregistrées. L'administrateur doit connaître tous les mots de passe afin de pouvoir accéder à toutes les pages des professeurs pour transférer les notes vers Best Grade.
- **pas de partage** : le fichier Best Grade mis en réseau c'est-à-dire partagé, risque de créer beaucoup de déconvenues surtout en ce qui concerne la perte de données. L'alternative est l'externalisation des fichiers de saisie de notes avec tous ces inconvénients (perte de

fichiers, perte du dossier en entier parce qu'un professeur a fait du couper au lieu du copier, ...).

- **administration fastidieuse** : administrer Best Grade n'est pas du tout facile car nécessitant la répétition des mêmes actions chaque année et chaque semestre. Le **Tableau 1**, illustre parfaitement la taille du travail d'administration avec les différentes tâches dédiées.
- **les risques** : les risques avec cette application sont très nombreux, nous pouvons citer :
 - La perte d'un fichier externe après remplissage ou du dossier contenant les pages des professeurs ;
 - La perte du dossier contenant les pages des professeurs, parce qu'un professeur au lieu de copier son fichier afin de travailler à la maison, coupe le dossier au lieu de le copier ;
 - La modification de notes non prise en compte car le fichier est déjà copié par l'administrateur.
- **la non activation des macros** : plusieurs professeurs oublient souvent d'activer les macros malgré l'alerte à l'ouverture. La non activation des macros avant le remplissage des notes vous oblige à utiliser l'outil « enregistrer » par défaut de Excel. Certain ferment la fenêtre après plusieurs tentatives de faire fonctionner le bouton « Enregistrer et Fermer ». Ils ne se rendent compte que tout le travail n'a pas été sauvegardé qu'après l'interpellation de l'administrateur. Un autre problème due à la non activation des macros est que même si l'administrateur réactive les macros, le bouton « copier » qui devrait lui faciliter la tâche dans ce fichier ne sera plus fonctionnel. Il est obligé de sélectionner les plages de cellules puis copier et coller en utilisant les outils par défaut.
- **absence de contrôle de la saisie** : Best Grade ne refuse pas la sauvegarde même si le remplissage est mal fait, il colorie en jaune toute cellule contenant une valeur numérique supérieure ou égale à vingt (20). Aucune alerte si une cellule devant contenir des valeurs numériques contenait une chaîne de caractères ou un caractère.
- **suppression ou ajout impossible** : après la création des pages de professeurs il est déconseillé d'ajouter ou de supprimer un ou plusieurs élèves. Si une telle opération est effectuée après création des pages des professeurs, au moment de coller les notes dans l'application, il se produira un décalage dans les cellules. Les notes d'un élève dans une matière vont être attribuées à un autre.

Un tel système montre beaucoup plus ses faiblesses dans le cas d'un établissement avec beaucoup d'élèves comme le CEM Néma 2.

La partie suivante abordera la problématique du sujet. Nous y proposerons notre solution et les objectifs de cette dernière.

III- Problématique du sujet

Depuis l'année scolaire 2012-2013, au CEM Néma 2, la gestion des notes est faite avec une application Excel et VBA appelée Best Grade. Mais avant, la gestion des notes se faisait avec des registres, qui sont des versions papiers des bulletins de notes confectionnées par des imprimeurs. Avec l'acquisition d'un ordinateur et d'une imprimante, l'administration d'alors a jugé opportun d'aller vers l'automatisation afin de réduire l'utilisation du papier. Best Grade étant open source et utilisée dans beaucoup d'établissement est choisi comme alternative.

Au fil des années scolaires, certaines faiblesses de cette dernière sont décelées avec les réclamations des utilisateurs concernant des problèmes rencontrés et le regard de personnes averties comme son administrateur.

Best Grade permet de produire des bulletins de notes, des listes d'appel et/ou fiches de notes, d'archiver les données de façon plus ou moins durable, etc. Vu sous cet angle, il n'est pas nécessaire de le changer. Pour voir ses faiblesses, nous devons nous poser la question suivante : « quelles sont les tâches nécessaires pour aboutir aux résultats précités ? ». Ces tâches et les acteurs devant exécuter ces dernières sont consignés dans le **Tableau 1**.

Des applications pour la gestion des notes scolaires qui sont développées au Sénégal et qui peuvent remédier considérablement aux manquements de Best Grade existent. Parmi elles nous avons : PLANETE, Sama-Ecole, MIRACLE, Daarasytem, SUNU-ECOLE, etc. Sama-Ecole développée par DevChrono, MIRACLE développée par Sunusoft, Daarasytem développée par Toubasoft IT Corporation et SUNU-ECOLE développée par DG CONSULTING sont des applications payantes, leur acquisition par le CEM Nema 2 impactera négativement sur ses maigres ressources financières.

La plateforme PLANETE du ministère de l'éducation nationale du Sénégal conçu pour les établissements et qui est gratuite devrait être la solution. Malheureusement, en 2014, année de son lancement et pendant laquelle des établissements pilotes parmi lesquels Néma 2 étaient conviés à une demi-journée de formation, le CEM n'avait pas pu utiliser cette plateforme. La raison principale était l'absence d'une connexion internet dans l'établissement. En 2015, le

CEM avait bénéficié de l'internet par le biais d'un projet géré par le ministère de l'éducation nationale en collaboration avec la Sonatel. Ce réseau était très instable avec un débit très faible, le point d'accès WIFI installé avait une faible portée.

L'enquête que nous avons menée dans certains établissements de la commune de Ziguinchor qui faisaient partie du groupe des établissements pilotes, a révélé d'autres problèmes qui sont les raisons de la non utilisation ou de l'abandon de la plateforme PLANETE malgré sa gratuité. Parmi ces raisons, nous avons :

- **les bogues (ou bugs) du système et la lenteur des réponses du serveur**, qui ont fini par installer un climat de suspicion entre l'administrateur local et ses collègues. En effet beaucoup de professeurs après soumission du formulaire de saisie de notes se déconnectent et partent à force d'attendre, alors que la soumission a échoué sans qu'ils le sachent. Ces derniers ne comprennent pas quand l'administrateur leur signale qu'ils n'ont pas introduit leurs notes ;
- **les fermetures répétées de la plateforme pour maintenance**, pendant les deux tiers de l'année scolaire, PLANETE est inaccessible pour des raisons de maintenance ;
- **la totale centralisation de l'administration de la plateforme**, qui nécessite des appels à l'aide à l'administrateur central ;
- **le code qui n'est pas open source**, ceci nous empêche de faire des modifications dans le code source de la plateforme ;
- **la lenteur dans les interventions**, le temps qui s'écoule entre l'interpellation par téléphone ou par email de l'administrateur local et la réponse de l'administrateur centrale peut faire des semaines ;
- **etc.**

Nous allons donc proposer une solution et se fixer des objectifs à atteindre comme tableau de bord afin que la solution proposée soit plus proches des attentes.

III.1- Solution liée au problème

Les nombreux problèmes évoqués plus haut (II.3) dans le processus de gestion des notes au CEM Néma 2 et la taille actuelle des données liées aux élèves et à leurs notes, nous ont poussé à proposer une application web afin de réduire considérablement les manquements de Best Grade. Cette application nommée « **Nema2SchoolApp** » va permettre entre autres, de :

- **Gérer les utilisateurs** : l'accès de tout utilisateur (ou agent) à l'application est conditionné par une identification en renseignant son matricule et son mot de passe. Ces derniers ne sont pas les seules informations de l'utilisateur qui sont renseignées lors de la création d'un utilisateur. Rappelons qu'un mot de passe est octroyé par défaut à chaque utilisateur, qui doit le modifier lors de sa première connexion. Ce nouveau mot de passe sera alors la propriété privée de l'utilisateur et modifiable autant de fois qu'il est nécessaire si le besoin se fait sentir. L'administrateur n'a pas besoin de supprimer un ancien utilisateur, il lui suffit seulement de l'archiver en transformant son statut en « false » c'est-à-dire inactif.
- **Gérer les données non évolutives** : ces données sont saisies, modifiées et supprimées par l'administrateur. Mais ces données, si elles sont bien renseignées ne changeront presque plus, d'où ce qualificatif. Ces données sont : les informations liées à l'école, aux classes pédagogiques, aux matières avec leur coefficient, aux années scolaires et aux semestres. Retenons que pour les semestres et les années scolaires, leur changement c'est-à-dire, la sélection de l'année scolaire en cours et du semestre se font par activation et désactivation. L'activation et la désactivation sont gérées par les utilisateurs qui ont le profil administrateur.
- **Gérer les professeurs** : tous les professeurs sont gérés par le principal. Cette gestion consiste à les affecter des classes avec la matière à y enseigner pour une année scolaire donnée. Un professeur, quand il va se connecter il ne verra que les classes qui lui sont affectées pour l'année scolaire en cours.
- **Gérer les surveillants** : les surveillants sont gérés par le SG. Il les affecte les classes qui seront sous leur responsabilité pendant une année scolaire. C'est pour qu'ils puissent saisir les absences après avoir fait le décompte.
- **Gérer les élèves** : les élèves sont ajoutés, modifiés et supprimés par les surveillants. Ils peuvent visualiser la liste des élèves qui sont orientés dans les classes qui sont sous leur responsabilité. Seul le Principal et le SG peuvent voir la liste de tous les élèves de l'établissement.
- **Gérer les absences** : la gestion des absences est assurée par les surveillants, chacun pour les classes qu'il détient peut ajouter les absences avant la fin du semestre.
- **Noter un élève** : seuls les professeurs peuvent saisir ou modifier les notes des élèves. Ils doivent tous le faire avant la fin du semestre, sinon toute note introduite après va être prise en compte pour l'autre semestre au lieu du semestre concerné.

- **Gérer les orientations des élèves** : l'orientation des élèves dans les classes, est assurée par le SG après leur ajout par les surveillants. L'ajout des élèves ne concernent que les nouveaux élèves, les anciens étant déjà dans la base de données.
- **Gérer l'impression de documents** : l'impression des documents comme les bulletins, les fiches de notes et les listes d'appel est faite par le Principal. Il suffit seulement de sélectionner une classe puis de cliquer sur « **Imprimer** » pour avoir la version papier de la liste des élèves de cette classe.

III.2- Les objectifs

L'objectif général est de proposer une application web responsive qui permettra d'optimiser le processus de gestion des notes au CEM Néma 2.

De façon spécifique, nous nous sommes fixés comme objectifs de :

- permettre une utilisation synchrone de « **Nema2SchoolApp** » ;
- sécuriser l'application en permettant à chaque utilisateur d'avoir son propre mot de passe, qu'il peut modifier à sa guise ;
- simplifier le processus par la réduction de plusieurs tâches semestrielles ou annuelles car répétitives, en stockant de façon durable les données associées dans une base de données ;
- archiver tout utilisateur ou élève inactif au lieu de le supprimer ;
- octroyer à chaque élève un matricule, dont les chiffres qui le composent, indiquent l'année d'arrivée dans l'établissement ;
- créer une interface homme-machine (IHM) ergonomique et facile d'utilisation ;
- adapter l'application aux terminaux les plus récents au cas où le partage se fait par la communication sans fil ;
- changer les années scolaires et les semestres par activation ou désactivation ;
- faciliter l'insertion, la modification et la suppression des données ;
- optimiser la recherche d'enregistrements ;
- vérifier la validité des données saisies avant de les enregistrer ;
- imprimer des listes d'appel, des fiches de notes et des bulletins ;
- filtrer les accès suivant les profils des utilisateurs.

Pour atteindre ces objectifs, il faut une collaboration entre la structure bénéficiaire (CEM Néma 2) et l'équipe de développement. Le choix d'une méthodologie de développement

adaptée s'impose. La méthodologie **SCRUM** qui fait partie des méthodes agiles a été choisie, nous allons dans le point suivant essayer de revenir en détail sur cette méthodologie.

IV- Processus de développement de l'application

Nous allons dans cette partie faire une brève introduction sur les méthodes agiles, ensuite sur la méthodologie SCRUM et nous terminerons par son adaptation à notre projet.

IV.1- Introduction aux méthodes agiles

Dans les méthodes traditionnelles, le client doit faire une expression détaillée et validée du besoin avant la réalisation et ne laissent presque pas de place à un éventuel changement. Le client attend la fin de la réalisation pour recevoir le produit.

Afin d'éviter l'effet tunnel, qui est une faiblesse des méthodes traditionnelles, les méthodes agiles optent pour une autre philosophie, que Florent Lothon décline sur quatre axes. Elles valorisent :

- **les individus et leurs interactions** plus que le processus et les outils ;
- **des logiciels opérationnels** plus qu'une documentation exhaustive ;
- **la collaboration avec les clients** plus que la négociation contractuelle ;
- **l'adaptation au changement** plus que le suivi du plan. [4]

Cette approche n'exclut ni de spécifications, ni de plan, ni d'outils, ni de contrat, elle ne dit pas aussi que le client peut changer d'avis à sa guise.

IV.2- La méthode SCRUM

La méthode SCRUM est de loin la plus utilisée parmi les méthodes agiles existantes car elle est la plus documentée avec des livres, des blogs, des vidéos, etc. La méthode SCRUM est simple à comprendre mais sa maîtrise en revanche est difficile. [4]

SCRUM peut être considérée comme un groupe ou un cadre de travail qui se focalise sur les buts communs en livrant de manière productive et créative des produits de la plus grande valeur possible. Elle s'appuie sur trois piliers :

- **la transparence** : communiquer avec le même langage pour permettre à tout le monde de comprendre (équipe et management), afin de rendre visible les problèmes ;
- **l'inspection** : pour recueillir à temps d'éventuels écarts entre l'objectif de l'itération et le travail réalisé. Cette inspection dans le sprint a lieu principalement lors du Daily Scrum, de la Sprint Review et de la Sprint Rétrospective ;
- **l'adaptation** : après inspection on peut s'adapter grâce à des actions visant à améliorer la situation. [4]

IV.2.1- Fonctionnement de SCRUM

Avant de se lancer dans le premier **Sprint** (itération), il faut tout d'abord mettre en place le « **Backlog** » du produit qui est une liste des tâches et des spécificités du produit. Il est le cahier des charges et le « **Product Owner** » représentant le client est le responsable. Il doit être priorisée afin de permettre à l'équipe de développeurs de savoir comment les Sprints doivent se suivre.

Le reste du processus est rythmé par des réunions clairement définies et limitées dans le temps.

- **la planification du Sprint** en sélectionnant les éléments prioritaires et réalisables dans le Sprint du « **Product Backlog** » par l'équipe de développement.
- **la revue de Sprint** tenue en fin de Sprint pendant laquelle l'équipe de développement présente les fonctionnalités terminées et recueille les feedbacks du « **Product Owner** ». C'est une occasion pour passer en revue les contours des prochains sprints et au besoin de faire un ajustement.
- **rétrospective de Sprint**, il réunit toute l'équipe pour partager sur les retours. Il se tient après la revue de sprint. C'est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc.) et de discuter sur les prochains sprints.
- **mêlée quotidienne**, elle dure au maximum 15 minutes et les acteurs restent debout, elle est appelée aussi « stand up meeting ». Chacun répond essentiellement à trois questions : Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici-là prochaine mêlée ? Quels obstacles me retardent ?

IV.2.2- Les rôles

Utiliser la méthode SCRUM dans le processus de développement d'un logiciel, nécessite la définition de trois rôles, comme le présente le tableau ci-dessous.

Tableau 2: Les rôles dans SCRUM

Rôle ou Acteur	Description
SCRUM Master (ou chef d'équipe)	<ul style="list-style-type: none">• Animer l'équipe de développement• Faciliter le travail de l'équipe de développement• Faciliter la communication interne et le dialogue avec le Product Owner• Veiller à l'application correcte de la méthodologie Scrum

	<ul style="list-style-type: none"> • S'acquitter de toutes les tâches administratives • Protéger l'équipe des interférences extérieures
Product Owner (ou le propriétaire du projet)	<ul style="list-style-type: none"> • Participer à l'élaboration des besoins • Porter la vision client du produit final • Travailler étroitement avec l'équipe de développement • Représenter le client et les utilisateurs finaux • Donne son accord ou son désaccord sur les résultats • Se charger du backlog produit et décider des priorités des différents user stories • Être le responsable du retour sur investissement
SCRUM Team (ou équipe de développement)	<ul style="list-style-type: none"> • Toutes les personnes chargées de l'implémentation des différents besoins du client • Travailler de façon incrémentale • Livrer une partie du produit final utilisable et testable à la fin de chaque sprint ou itération • Être pluridisciplinaire et auto organisée • Définir les spécifications techniques • Être en plein temps dans le projet

La figure ci-dessous résume le fonctionnement de la méthode Scrum.

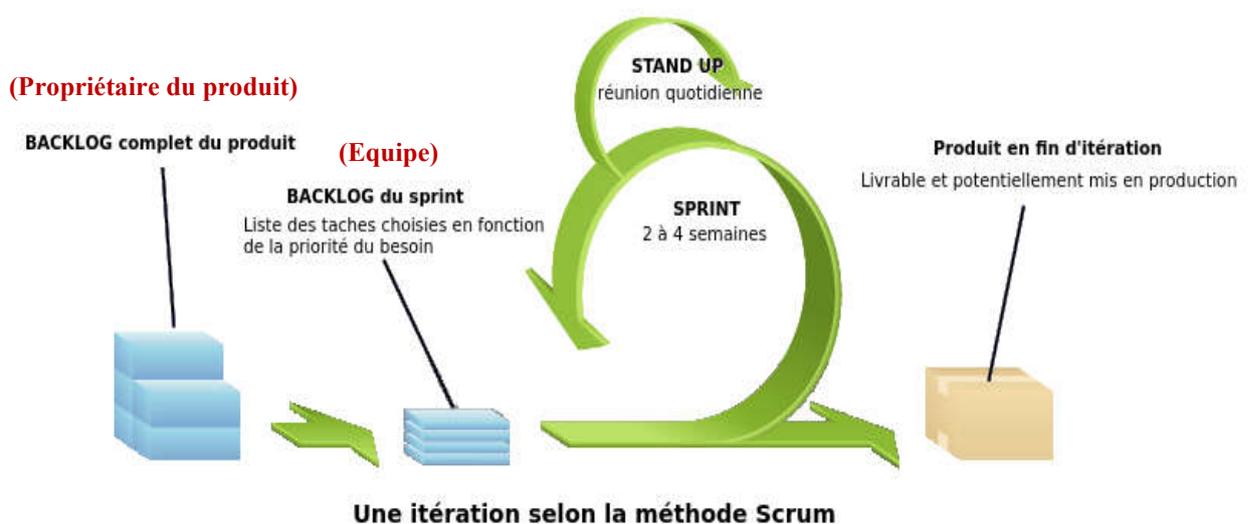


Figure 2: Fonctionnement du Scrum [1]

IV.3- Adaptation de la méthodologie Scrum à notre contexte

Dans cette partie, nous allons montrer comment la méthodologie Scrum est appliquée à notre projet. Nous commencerons par présenter un extrait du « **Product Backlog** » avec un tableau contenant l'ordre de priorité, quelques « **User story** » et les tâches du « **Sprint Backlog** ». Nous terminerons la partie par le déroulement des événements du projet.

IV.3.1– Extrait du « **Product Backlog** » ou « **Backlog Produit** »

Le tableau suivant contient les différents acteurs de l'équipe de pilotage et leurs rôles.

Tableau 3: Extrait du Product Backlog

Conception et développement d'une application web pour la gestion des notes scolaires		
N°	Product Backlog	Sprint Backlog
1	Navigation sur l'application	Page d'accueil
		Page avec le menu principal
		Tableaux de bord des différents profils avec leur menu spécifique
2	Gestion des utilisateurs	Ajouter, modifier et supprimer un utilisateur
		Ajouter, modifier et supprimer un rôle
		Lister les utilisateurs et les rôles
		Octroyer un rôle à un utilisateur
		Archiver un utilisateur
		Configuration du système d'authentification
		Modification mot de passe
3	Gestion des années scolaires	Ajouter une année scolaire
		Lister les années scolaires
		Changer une année scolaire par simple clic
4	Gestion des semestres	Ajouter les semestres dans la base de données
		Lister les semestres
		Changer de semestre par un simple clic
5	Gestion des classes pédagogiques	Ajouter une classe pédagogique
		Modifier une classe pédagogique
		Lister l'ensemble des classes
6	Gestion des matières	Ajouter une matière
		Modifier une matière

		Supprimer une matière
		Lister les matières
7	Gestion des élèves	Ajouter un élève
		Modifier un élève
		Supprimer un élève
		Archiver un élève
		Rechercher un élève
		Voir détails d'un élève
		Lister l'ensemble des élèves
		Lister les élèves d'une classe
8	Gestion des orientations des élèves dans une classe	Orienter ou réorienter un élève dans une classe
		Modifier une orientation
		Supprimer une orientation
		Lister les orientations
9	Gestion des professeurs	Lister les professeurs
		Affecter une classe-matière à un professeur
		Modifier une affectation
		Supprimer une affectation
		Lister les affectations
10	Gestion des surveillants	Lister les surveillants
		Affecter une classe à un surveillant
		Modifier une affectation de classe
		Supprimer une affectation
		Lister les affectations de classe
11	Gestion des notes	Saisir une note
		Modifier la saisie d'une note
		Consulter ses saisies
12	Gestion des absences	Saisir le nombre d'absences d'un élève
		Modifier une saisie d'absence
		Lister la saisie des absences

IV.3.2- Déroulement du projet

Dans le déroulement de notre projet, nous avons essayé dans la mesure du possible, d'appliquer la méthodologie Scrum. La raison principale qui nous a empêché de pouvoir appliquer à la lettre la méthodologie est la fonction de chaque acteur de l'équipe de pilotage. Cette difficulté se ressent surtout quand une réunion nécessite la présence du **Product Owner** et du **Scrum Master**. L'alternative trouvée, est de faire jouer à la **Scrum Team** le rôle d'intermédiaire, heureusement que M. Ibrahima Gnamo Sonko de l'équipe de développement est enseignant au CEM Néma 2 donc pouvant jouer le rôle de Product Owner et d'utilisateur final.

Nous (**Product Owner et Scrum Team**) avons donc travaillé sur l'ensemble des fonctionnalités du produit et leur priorisation, ce qui nous a permis d'avoir notre **product backlog**. Nos **sprints** ont une durée de 2 semaines au minimum et de 3 semaines au maximum, contrairement à la règle qui stipule que tous les sprints doivent avoir une durée constante et égale à 4 semaines. Ceci est dû à la difficulté évoquée plus haut et au fait que notre équipe de pilotage est très réduite.

Toutes les réunions (planification du sprint, revue de sprint, rétrospective de sprint et mêlée quotidienne), respectent la durée normale établie par la méthodologie Scrum.

Nous venons de terminer dans ce chapitre, la présentation de notre sujet de mémoire, par la présentation du CEM Néma 2, la description du système de gestion existant avec ses manquements, la proposition d'une solution et la méthodologie utilisée dans notre processus de développement. Nous allons alors utiliser le Product Backlog comme tableau de bord pour faire la spécification et l'analyse des besoins

CHAPITRE II : SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS

Le choix sur la méthodologie de travail étant fait et cette dernière détaillée, alors un autre pan de notre travail peut être entamé. Nous allons dans ce chapitre qui sera constitué de deux grandes parties, identifier toutes les actions que le système doit faire en réponse à une demande et faire leur analyse.

Pour ce faire, nous avons choisi d'utiliser UML pour modéliser notre système et son fonctionnement car très adapté à la programmation orientée objet, possédant un langage unifié et faisant partie des plus utilisés.

I- Spécification des besoins fonctionnels

Nous allons tout d'abord identifier les acteurs du système ensuite les fonctionnalités du système et en fin nous terminerons par les différents diagrammes de cas d'utilisation.

I.1- Identification des acteurs du système

Un acteur du système n'est pas forcément un être humain, il a un rôle à jouer et ce dernier peut être interne ou externe. L'activité d'un système vise à satisfaire aux besoins de l'acteur.

Concernant notre projet, nous avons identifié plusieurs acteurs comme consigné dans le tableau ci-dessous.

Tableau 4: Identification des acteurs

<u>Acteurs</u>	<u>Rôles</u>
Administrateur (ADMIN)	Il est chargé du choix l'année scolaire et du semestre en cours et de la gestion des utilisateurs. Parfois il fait une mise à jour de classes, de matières, etc.
Principal (PRINCE)	Affecter des classes et les matières associées aux professeurs, et imprimer des documents
Professeur (PROF)	Saisir les notes des élèves et vérifier la saisie

Surveillant Général (SURVGEN)	Affecter des classes aux surveillants et orienter les élèves
Surveillant (SURV)	Inscrire les élèves et saisir les absences pour chaque semestre

Après identification des acteurs de notre système, nous allons maintenant passer à l'identification des fonctionnalités de notre système.

I.2- Identification des fonctionnalités du système

Les besoins fonctionnels sont toutes les actions du système exécutées afin de rendre service à ses futurs usagers. L'étude du **product backlog**, nous a permis d'identifier les différentes fonctionnalités de notre système qui sont consignées dans un tableau (**Tableau 5**).

Tableau 5: Liste des fonctionnalités du système

<u>Fonctionnalités</u>	<u>Acteurs</u>
S'authentifier	ADMIN, PRINCE, PROF, SURVGEN et SURV
Gérer un agent (ajouter, modifier, supprimer, archiver)	ADMIN
Gérer un rôle (ajouter, modifier, supprimer, lister)	ADMIN
Attribuer un rôle, modifier une attribution et supprimer une attribution	ADMIN
Modifier son mot de passe	ADMIN, PRINCE, PROF, SURVGEN et SURV
Gérer une année scolaire (ajouter, modifier, supprimer, lister, changer)	ADMIN
Gérer un semestre (changer de semestre)	ADMIN
Gérer une classe pédagogique (ajouter, modifier)	ADMIN
Gérer une matière (ajouter, modifier, supprimer)	ADMIN
Modifier les informations de l'école	ADMIN

Voir l'effectif des utilisateurs, des élèves inscrits, des garçons inscrits et des filles inscrites	ADMIN, PRINCE
Voir la liste des agents	ADMIN, PRINCE
Voir la liste des rôles attribués aux agents	ADMIN, PRINCE
Voir la liste des attributions de classes aux surveillants	PRINCE, SURVGEN
Voir la liste de tous les élèves actuels et anciens et rechercher par nom	PRINCE, SURVGEN, SURV
Voir la liste des élèves d'une classe	PRINCE, SURVGEN, PROF
Voir la liste des classes pédagogiques	ADMIN, PRINCE
Voir la liste des matières	ADMIN, PRINCE
Gérer les professeurs (lister les professeurs, affecter une classe et une matière, lister toutes leurs attributions de classes, modifier, supprimer)	PRINCE
Générer le bulletin d'un élève pour un semestre	PRINCE
Insérer une observation particulière dans un bulletin	PRINCE
Imprimer une liste de classe, une fiche de notes, un bulletin	PRINCE
Voir la liste de ses classes et matières	PROF
Noter un élève	PROF
Voir la liste des notes des élèves d'une classe ainsi que leurs moyennes et modifier une note	PROF
Voir l'effectif des élèves inscrits, des garçons inscrits et des filles inscrites	SURVGEN, SURV
Gérer les surveillants (lister les surveillants, affecter des classes, modifier, supprimer)	SURVGEN
Orienter, réorienter	SURVGEN
Gérer les orientations (modifier, supprimer, lister les orientations)	SURVGEN

Gérer les élèves (ajouter, modifier, supprimer, voir les détails d'un élève)	SURV
Gérer les absences (saisir, modifier, lister)	SURV

Les diagrammes de cas d'utilisation vont nous permettre de pouvoir avoir un aperçu des liens entre les acteurs et les fonctionnalités du système. La section suivante va nous permettre de présenter les différents diagrammes de cas d'utilisation de notre système.

I.3- Diagrammes des cas d'utilisation

Les diagrammes de cas d'utilisation font partie des diagrammes UML qui permettent de donner une vision globale du comportement fonctionnel du système. Les diagrammes de cas d'utilisation sont des supports de communication entre les acteurs d'un projet, par exemple ils peuvent servir lors de la présentation visuelle du projet au **Product Owner** par la **Scrum Team**. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (**actors**), ils interagissent avec les cas d'utilisation (**use cases**).

Nous allons présenter les diagrammes de cas d'utilisation par utilisateur.

➤ Diagrammes des cas d'utilisation pour l'administrateur

L'administrateur (ADMIN) gère les agents (les utilisateurs) par l'ajout, la modification, la suppression, l'archivage (en changeant le statut en « **false** », ce qui l'empêche de pouvoir se connecter au système), l'attribution de rôles, la consultation de la liste des agents et de la liste des agents avec leurs attributions de rôles.

Il gère les années scolaires et les semestres par l'ajout des nouvelles années scolaires, le changement en désactivant l'ancienne et en activant la nouvelle année scolaire ou le nouveau semestre.

Il gère les classes pédagogiques et les matières par l'ajout, la modification et la suppression si nécessaire, car ces actions seront rarement nécessaires. Il peut aussi consulter la liste des classes ou celle des matières.

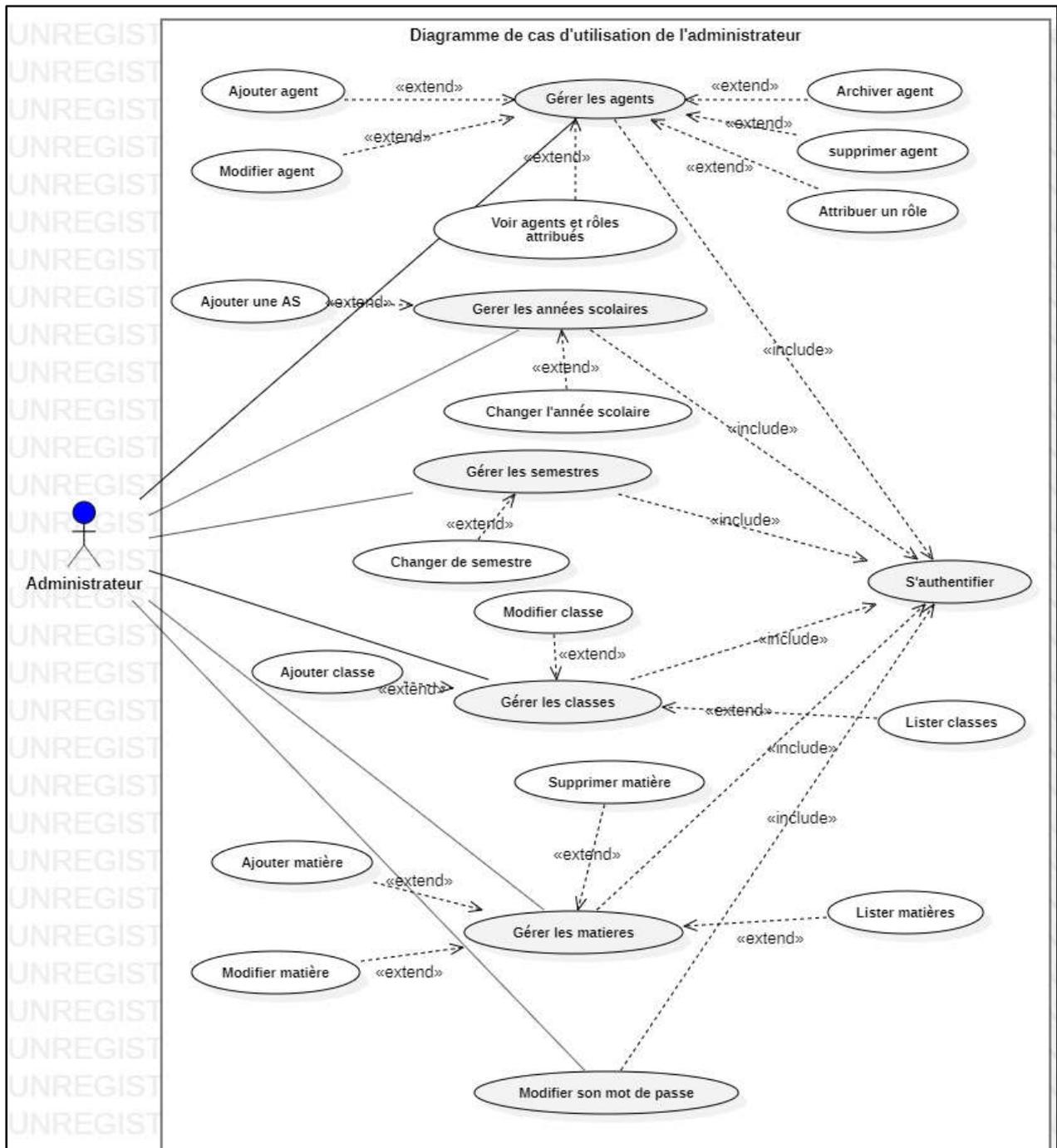


Figure 3: Diagramme de cas d'utilisation pour l'Administrateur

➤ Diagrammes des cas d'utilisation pour le Principal

Le Principal (PRINCE) gère les professeurs par l'affectation de classes et matières, la modification ou la suppression d'une affectation, la consultation de la liste des affectations.

Il peut aussi voir la liste de tous les professeurs, de toutes les classes pédagogiques, de toutes les matières, de tous les agents avec les rôles qui leur sont attribués, de tous les surveillants

avec les classes qui leurs sont affectées par le Surveillant général, de tous les élèves de l'établissement anciens ou actuels.

Il peut aussi rechercher un élève par son nom de famille, générer un bulletin de notes et imprimer un document (fiche de notes, liste d'appel, bulletin de notes, ...).

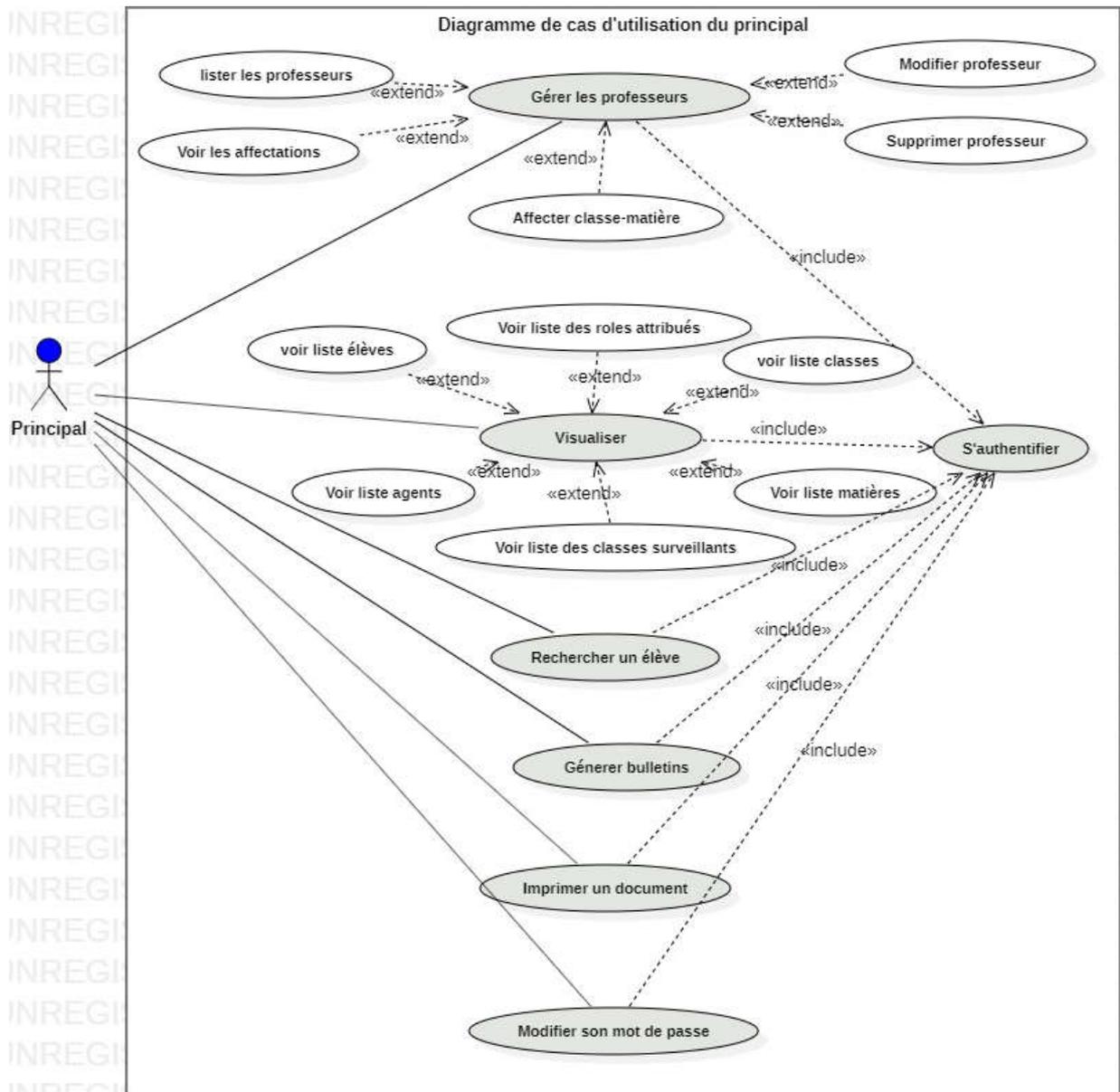


Figure 4: Diagramme de cas d'utilisation pour le Principal

➤ Diagrammes des cas d'utilisation pour le Professeur

Le professeur (PROF) dans les classes qui lui sont affectées pour une année scolaire donnée saisie les notes des élèves, consulte ses saisies et modifie si besoin ses saisies. Après

authentification, le professeur voit la liste de toutes les classes qui lui sont affectées ainsi que les matières avec leur coefficient qu'il doit dispenser dans chacune d'elles.

Le professeur peut aussi voir la liste de tous les élèves d'une de ses classes.

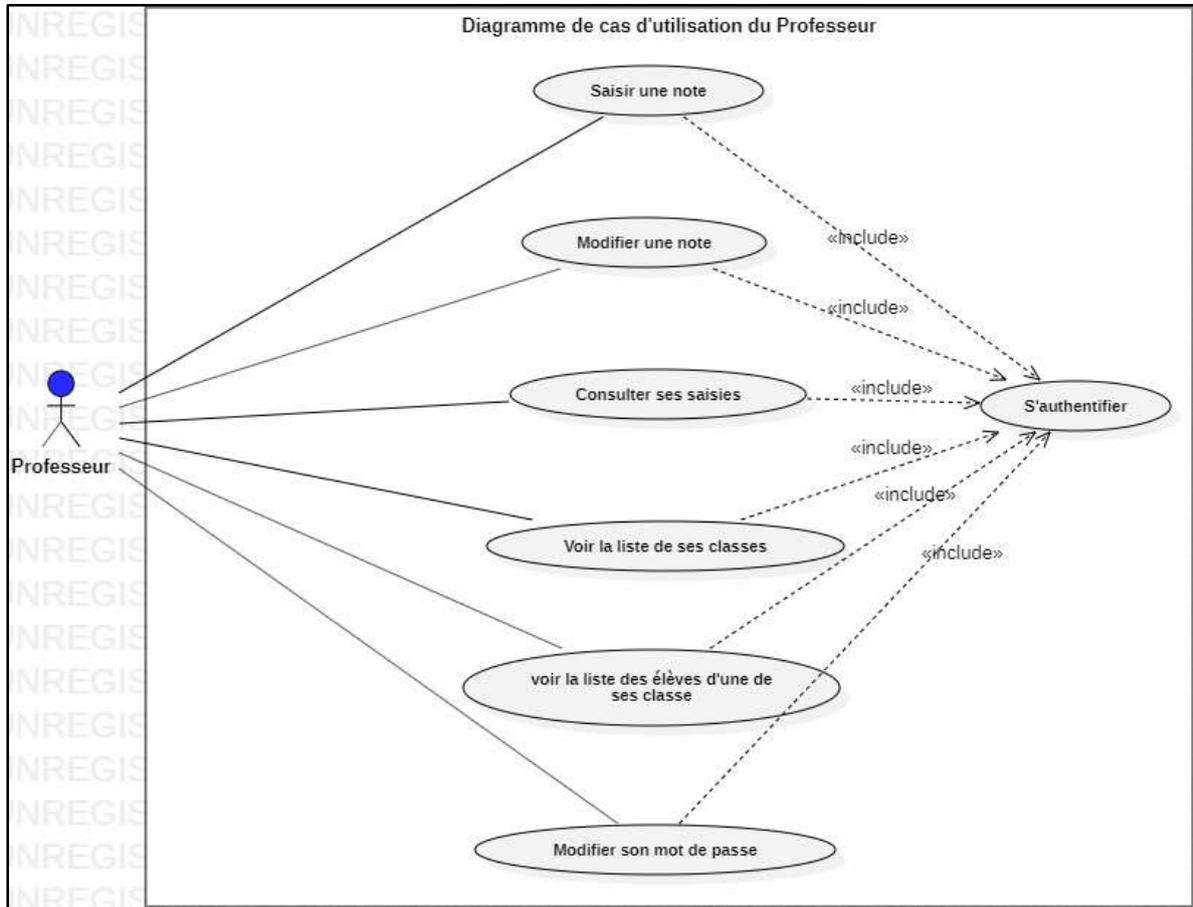


Figure 5: Diagramme de cas d'utilisation pour le Professeur

➤ Diagrammes des cas d'utilisation pour le Surveillant Général

Le surveillant général étant le responsable des surveillants simples doit donc se charger de leur gestion. C'est pourquoi il peut consulter la liste des surveillants, affecter des classes aux surveillants simples pour une année scolaire donnée, voir la liste de toutes les affectations effectuées, modifier une affectation et supprimer une affectation.

Il oriente ou réoriente les élèves dans les classes pour une année scolaire, liste l'ensemble des orientations, modifie et supprime au besoin une orientation.

Il peut consulter la liste de tous les élèves, la liste des élèves d'une classe donnée et rechercher un élève par son nom.

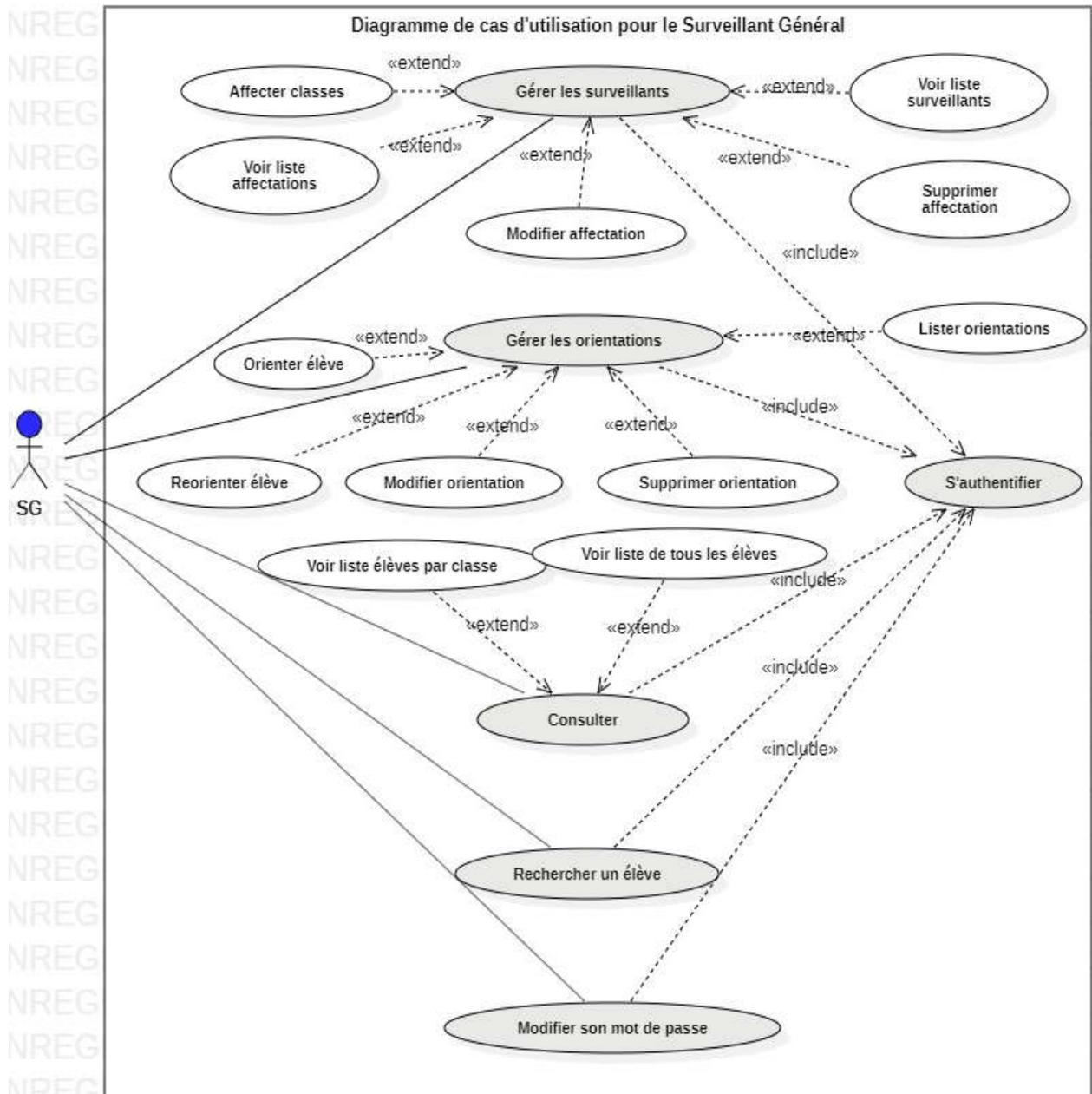


Figure 6: Diagramme de cas d'utilisation pour le Surveillant Général

➤ Diagrammes des cas d'utilisation pour le Surveillant

Responsable direct des élèves, le surveillant ajoute, modifie ou supprime un élève. Un surveillant peut voir la liste de tous les élèves, rechercher un élève par son nom et voir les détails d'un élève à partir de la liste de tous les élèves.

Il est chargé de saisir le nombre d'absences d'un élève pour un semestre, lister l'ensemble des saisies d'une classe et les modifier.

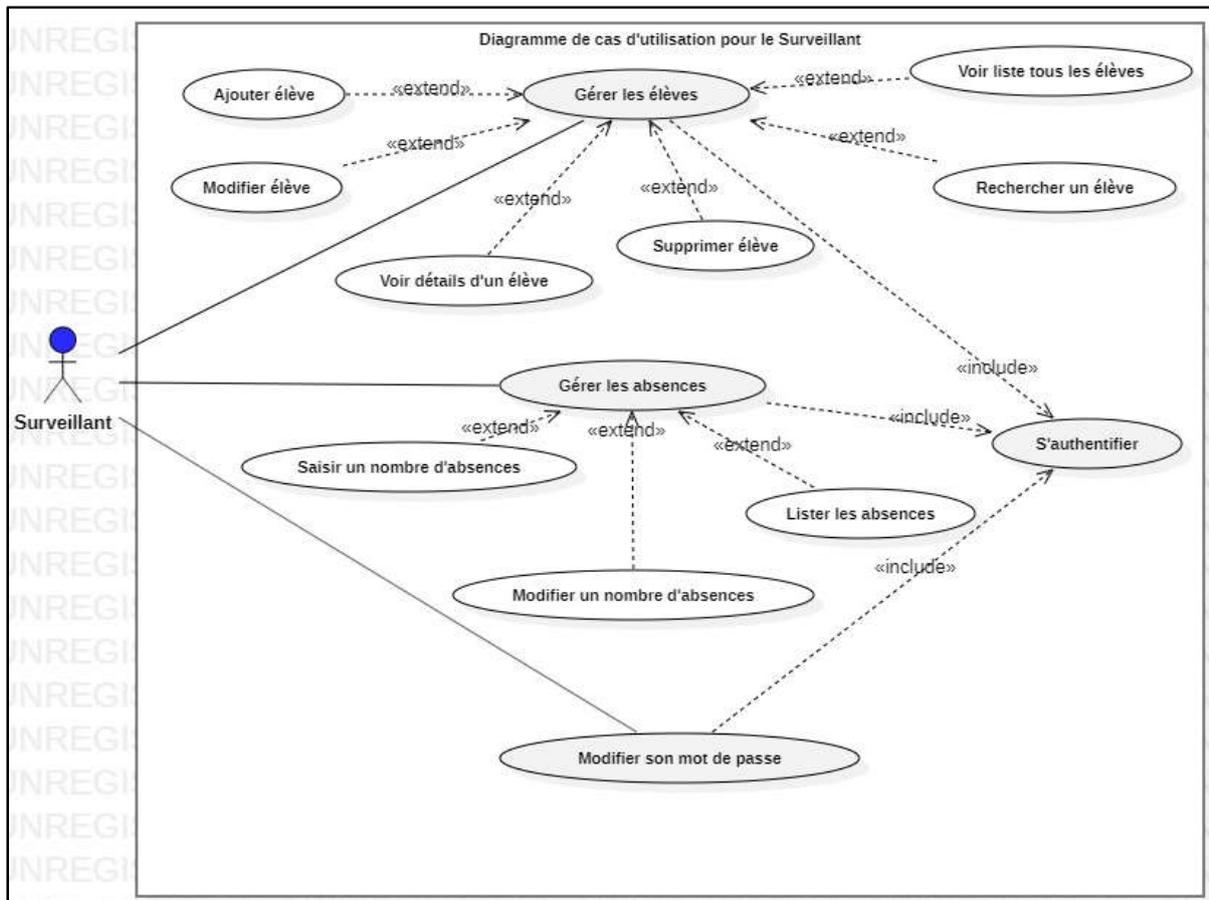


Figure 7: Diagramme de cas d'utilisation pour le Surveillant

Deux cas d'utilisation (Modifier son mot de passe et s'authentifier) n'ont pas été détaillés pour chaque acteur parce qu'ils sont communs à tous. L'administrateur en ajoutant un utilisateur, ne remplit pas le mot de passe de celui-ci car il est fourni par défaut par le système. Donc c'est à l'utilisateur de le modifier lors de sa première connexion.

L'authentification est obligatoire pour accéder à toutes les vues sauf pour la page d'accueil et celle d'authentification.

II- Analyse des besoins fonctionnels

La spécification des besoins fonctionnels, nous a permis de voir les différents liens qui existent entre les acteurs du système et les cas d'utilisation (c'est-à-dire les fonctionnalités). L'analyse des besoins fonctionnels va nous permettre de voir les détails de chaque cas d'utilisation. C'est-à-dire comment les fonctions offertes par le système satisfont les besoins de son utilisateur.

Nous allons dans la suite faire l'analyse des besoins fonctionnels pour quelques cas d'utilisation, par une description et une représentation schématique de ces derniers.

II.1- Analyse de l'authentification

Le premier cas d'utilisation que nous allons faire l'analyse est « S'authentifier ». Nous débuterons par faire sa description ensuite son diagramme d'activités et en fin son diagramme de séquence.

1. Description du cas d'utilisation « s'authentifier »

Le tableau ci-dessous, donne la description du cas d'utilisation

Tableau 6: Description du cas d'utilisation "S'authentifier"

Description du cas d'utilisation « s'authentifier »	
Titre	S'authentifier
Résumé	Permet de contrôler l'accès au système
Acteur (s)	ADMIN, PRINCE, PROF, SURVGEN et SURV
Pré condition	Avoir un compte d'utilisateur
Scenari nominal	<ul style="list-style-type: none">✓ L'utilisateur saisit son identifiant et son mot de passe✓ Le système vérifie l'existence de ces informations✓ Le système récupère le profil de l'utilisateur
Post condition	Accéder à la page avec le menu principal et son prénom et nom affichés
Exception	Saisie d'un identifiant

2- Diagramme d'activités du cas d'utilisation « s'authentifier »

Le cas d'utilisation « s'authentifier » est composé de trois activités. La première activité est la **saisie de l'identifiant et du mot de passe**, si le mot de passe ou l'identifiant est fausse alors **s'affiche un message d'erreur** et on revient à la première activité. Dans le cas où la saisie est correcte alors on accède à la **page avec le menu principal** avec le nom et le prénom de l'utilisateur connecté. L'ensemble de ces enchainements est décrit dans le diagramme d'activité ci-dessous. (Figure 8)

3- Diagramme de séquences du cas d'utilisation « s'authentifier »

L'utilisateur saisit son login et son mot de passe, le système vérifie ensuite si les informations saisies se trouvent dans la base de données. Après vérification, le système renvoie la page avec le menu principal. (Figure 9)

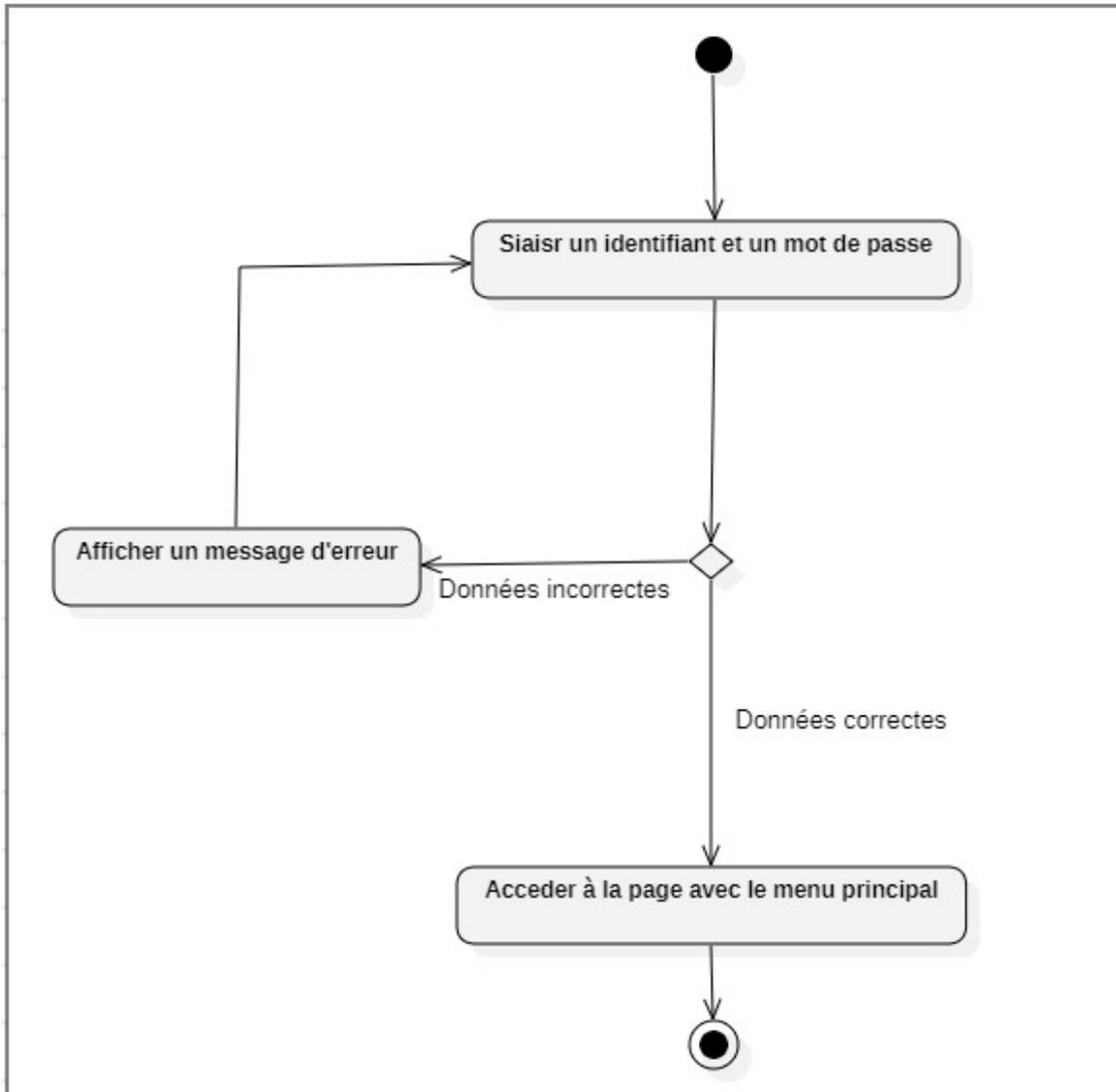


Figure 8: Diagramme d'activité du cas "s'authentifier"

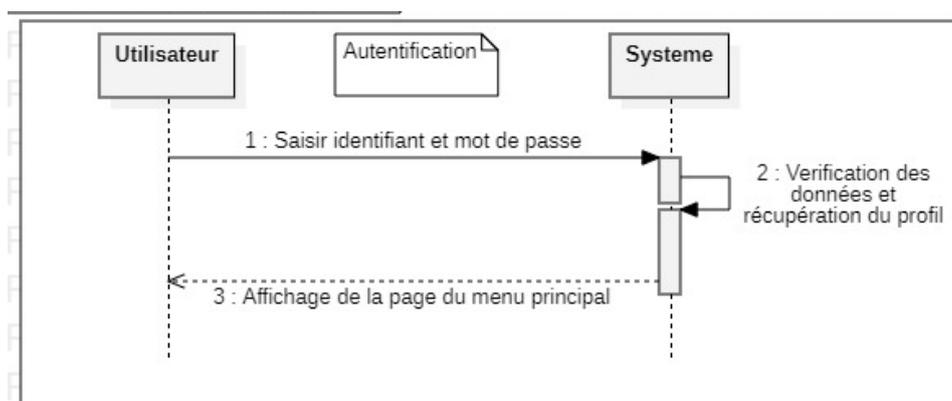


Figure 9: Diagramme de séquences du cas "s'authentifier"

II.2- Analyse de l'archivage d'un agent (utilisateur)

Le cas d'utilisation « Archiver un agent », c'est-à-dire empêcher l'accès d'un utilisateur à l'application sans pour autant supprimer les données le concernant est notre deuxième cas d'utilisation à analyser.

Nous le présenterons en détails par une description, ensuite nous travaillerons sur ses activités et en fin nous présenterons son digramme de séquences.

1- Description du cas d'utilisation « Archiver un agent »

La description de ce cas d'utilisation est faite avec un tableau (**Tableau 7**)

Tableau 7: Description du cas d'utilisation "Archiver un agent"

Description du cas d'utilisation « archiver un agent »	
Titre	Archiver un agent
Résumé	Permet l'archivage d'un agent
Acteur (s)	ADMIN
Pré condition	Authentification en tant qu'administrateur
Scenari nominal	<ul style="list-style-type: none">✓ Accéder à la liste de tous les agents✓ Chercher par défilement et pagination l'agent à archiver✓ Cliquer sur le bouton « Archiver »
Post condition	Le changement de son statut devant son nom en « Archivé », le changement de couleur et de texte du bouton sur lequel on avait cliqué
Exception	L'agent à archiver l'est déjà ou n'est pas trouvé

2- Diagramme d'activité du cas d'utilisation « Archiver un agent »

Pour archiver un agent, l'administrateur après authentification accède à la page avec le menu administrateur après celle du menu principal. Sur ce menu il choisit le sous onglet « **liste des agents** » de l'onglet « **Agents et Rôles** », la **liste de tous les agents** est alors chargée. Il cherche le nom de l'agent à archiver par défilement verticale et/ou pagination, devant le nom de ce dernier, il y a un bouton de couleur orange avec le texte « Archiver ». Il suffit de cliquer sur ce bouton pour archiver (transformer son statut en archivé) cet agent. Le bouton devient bleu et son texte devient « Activer » (voir **Figure 10**).

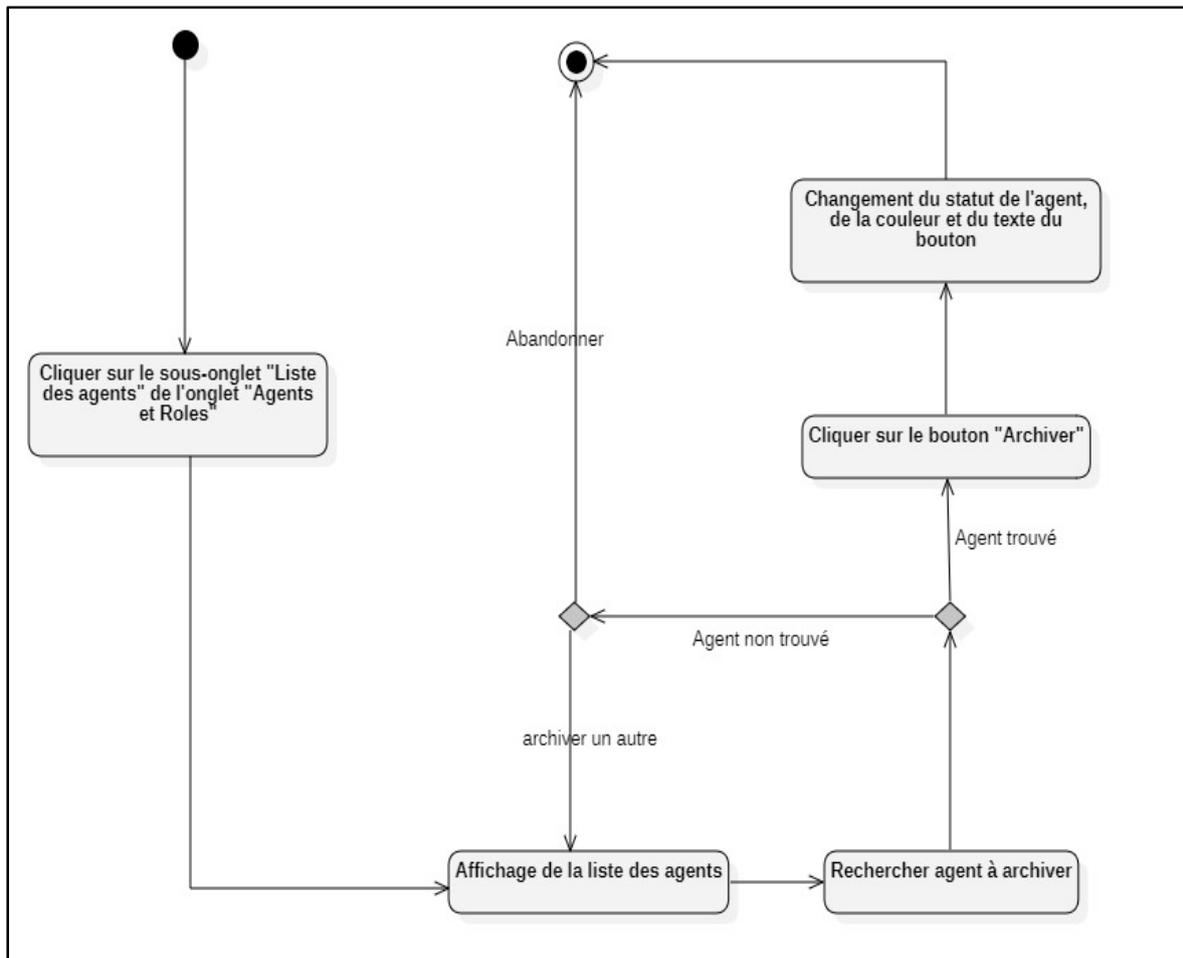


Figure 10: Diagramme d'activité du cas "Archiver un agent"

3- Diagramme de séquences du cas d'utilisation « Archiver un agent »

Après connexion avec son login et son mot de passe, l'administrateur à partir de sa page d'accueil, accède à la page contenant la liste des agents par un clique sur le sous-onglet « **Liste des agents** ». L'utilisateur clique sur le bouton « **Archiver** » devant le nom de l'agent concerné. (Figure 11)

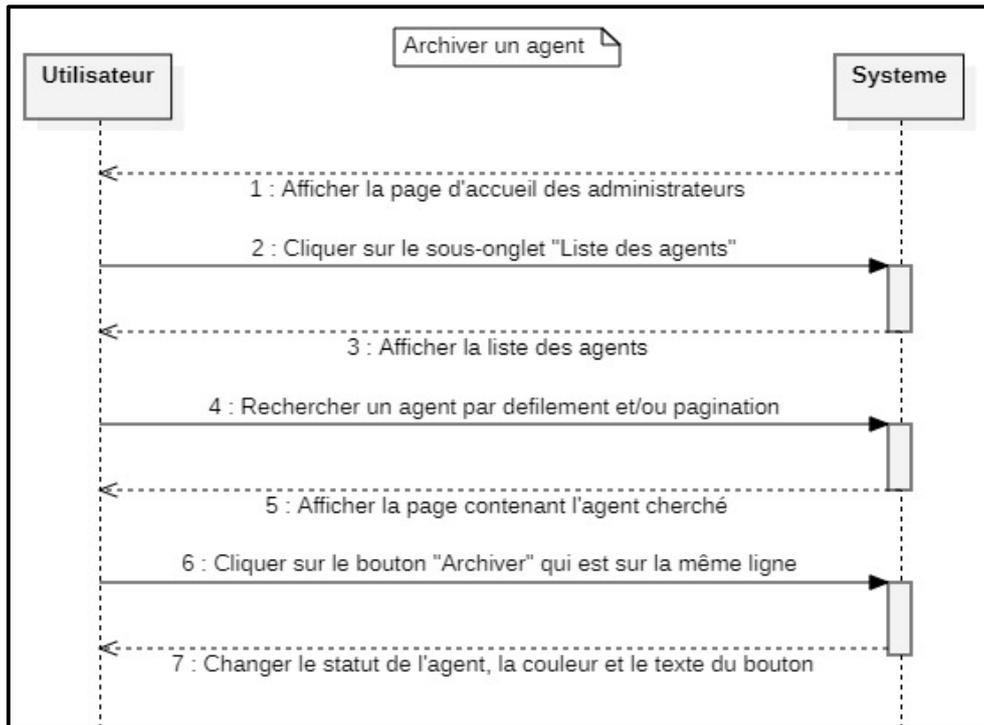


Figure 11: Diagramme de séquences du cas "Archiver un agent"

II.3- Analyse du cas affecter une classe et une matière à un professeur

Nous allons dans cette partie étudier le cas d'utilisation « affecter classe-matière » à un professeur. Cette analyse sera faite par une description, un diagramme d'activité et un diagramme de séquences.

1- Description du cas d'utilisation « Affecter classe-matière »

Le tableau (**Tableau 8**) ci-dessous donne la description du cas d'utilisation.

Tableau 8: Description du cas d'utilisation "Affecter classe-matière"

Description du cas d'utilisation « Affecter classe-matière »	
Titre	Affecter classe-matière
Résumé	Il permet d'affecter une classe et une matière à y dispenser à un professeur.
Acteurs	PRINCE
Pré condition	Authentification en tant que principal, activation de la nouvelle année scolaire
Scenario nominal	✓ Accéder à la liste des professeurs

	<ul style="list-style-type: none"> ✓ Défiler et/ou paginer pour trouver le professeur ✓ Cliquer sur « Affecter classe-matière »
Post condition	Retour à la liste des professeurs sans génération d'un message d'erreur
Exception	Affecter le même couple classe et matière au même professeur

2- Diagramme d'activité du cas d'utilisation « Affecter classe-matière »

Le Principal accède à la **page contenant son menu** après authentification et clique sur l'onglet « Principal ». Dans sa page il affiche la **liste des professeurs** en passant par le sous-onglet « Liste des professeurs » de l'onglet « Gestion Professeurs ». Il cherche le professeur dont il doit affecter des classes pour une année scolaire, le clique sur le lien « **Affecter une classe-matière** » conduit vers le **formulaire d'affectation**. A ce niveau il suffit de **choisir la classe et la matière** à y enseigner puis **sauvegarder**. Si la sauvegarde est réussie, vous serez redirigé vers la liste des professeurs sinon vous recevez un message d'erreur (quand le couple classe-matière est déjà choisi pour une année scolaire donnée). (**Figure 12**)

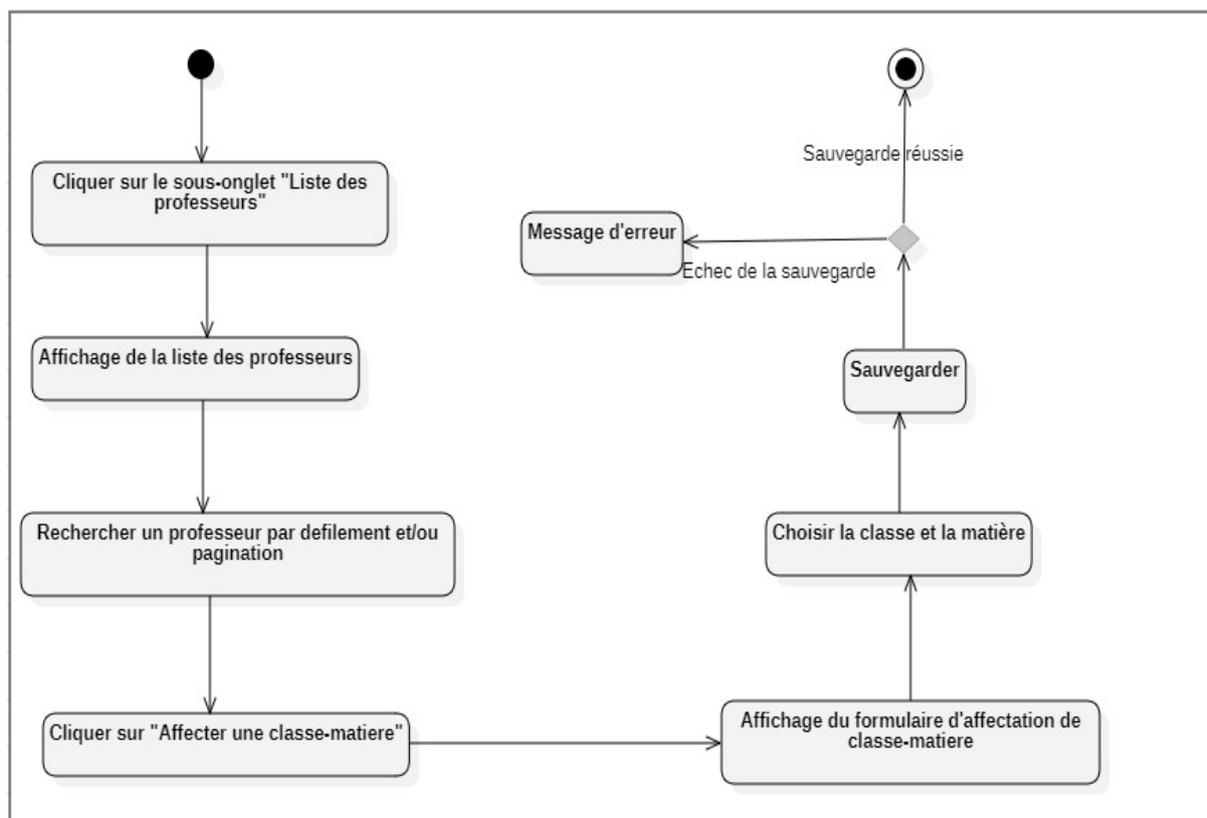


Figure 12: Diagramme d'activité du cas "Affecter classe-matière"

3- Diagramme de séquences du cas « Affecter classe-matière »

Pour pouvoir procéder à l'affectation d'une classe et d'une matière à un professeur, le Principal doit après connexion et accès à sa page d'accueil, accéder à la liste des professeurs. A ce niveau il cherche le professeur concerné par défilement et/ou pagination puis clique sur le bouton « **Affecter classe-matière** » pour afficher le formulaire. Dans le formulaire, il sélectionne la classe et la matière puis valide par le bouton « **Sauvegarder** ». (Figure 13)

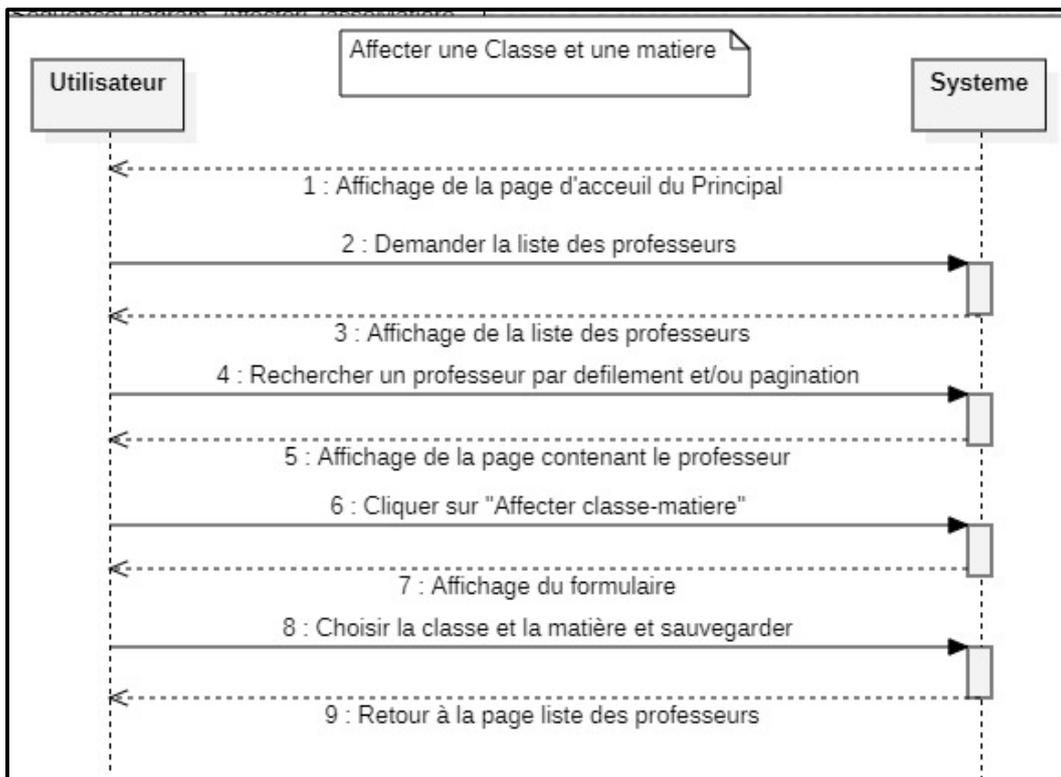


Figure 13: Diagramme de séquences du cas "Affecter classe-matière"

II.4- Analyse du cas d'utilisation ajouter des notes pour un semestre

Ajouter des notes est une tâche qui est exclusivement réservé aux professeurs, nous allons faire l'analyse de ce cas d'utilisation. Cette analyse va débiter par la description du cas, ensuite nous allons représenter son diagramme d'activité et son diagramme de séquences.

1- Description du cas d'utilisation « Ajouter des notes »

Le tableau ci-dessous donne la description du cas d'utilisation.

Tableau 9: Description du cas d'utilisation « Ajouter notes »

Description du cas d'utilisation « ajouter notes »	
Titre	Ajouter des notes

Résumé	Il permet de saisir et de sauvegarder des notes semestrielles
Acteurs	PROF
Pré condition	Authentification, activation de la nouvelle année scolaire et du semestre concerné
Scenari nominal	<ul style="list-style-type: none"> ✓ Accéder à sa page ✓ Choisir l'onglet « Insérer notes » ✓ Choisir la classe puis charger la liste des élèves ✓ Cliquer sur « Noter » pour avoir le formulaire d'insertion
Post condition	Retour à la page contenant la liste des élèves de la classe choisie
Exception	Insérer une note qui n'est pas « Double » ou qui n'est pas comprise entre 0 et 20 inclus

2- Diagramme d'activité du cas d'utilisation « Ajouter notes »

Le professeur, après authentification, accède à sa **page d'accueil** après avoir choisi l'onglet « Professeurs » du menu principal. A ce niveau il choisit l'onglet « **Insérer notes** » et accède à une page presque vide contenant uniquement un formulaire à remplir pour charger la liste des élèves d'une de ses classes. Il **choisit dans la liste de choix la classe** et clique sur le bouton « **charger la page** », une liste de tous les élèves de la classe est alors chargée. Devant chaque nom d'élève il y'a un bouton-lien « **Noter** » pour aller vers le formulaire d'insertion de notes. Apres insertion de toutes les notes, il les sauvegarde en cliquant sur le bouton « **Sauvegarder** ». Si la saisie est bien faite, une redirection vers la liste des élèves de la classe choisie sera faite, sinon des messages spécifiques aux erreurs seront affichés en dessous des champs concernés. (Figure 14)

3- Diagramme de séquences du cas « Ajouter notes »

Le professeur pour pouvoir attribuer des notes à un élève doit accéder à la page d'accueil des professeurs en se connectant puis cliquer sur l'onglet « **Insérer notes** » pour afficher le formulaire qui permet de charger la classe concernée. Dans ce formulaire, il sélectionne la classe puis clique sur le bouton « Charger la page » pour voir la liste des élèves de cette classe. Devant le nom de l'élève, il clique sur le bouton « Noter » pour ouvrir le formulaire de saisie

de notes. Il renseigne les notes de l'élève et sauvegarde en cliquant sur le bouton « Sauvegarder ». (Figure 15)

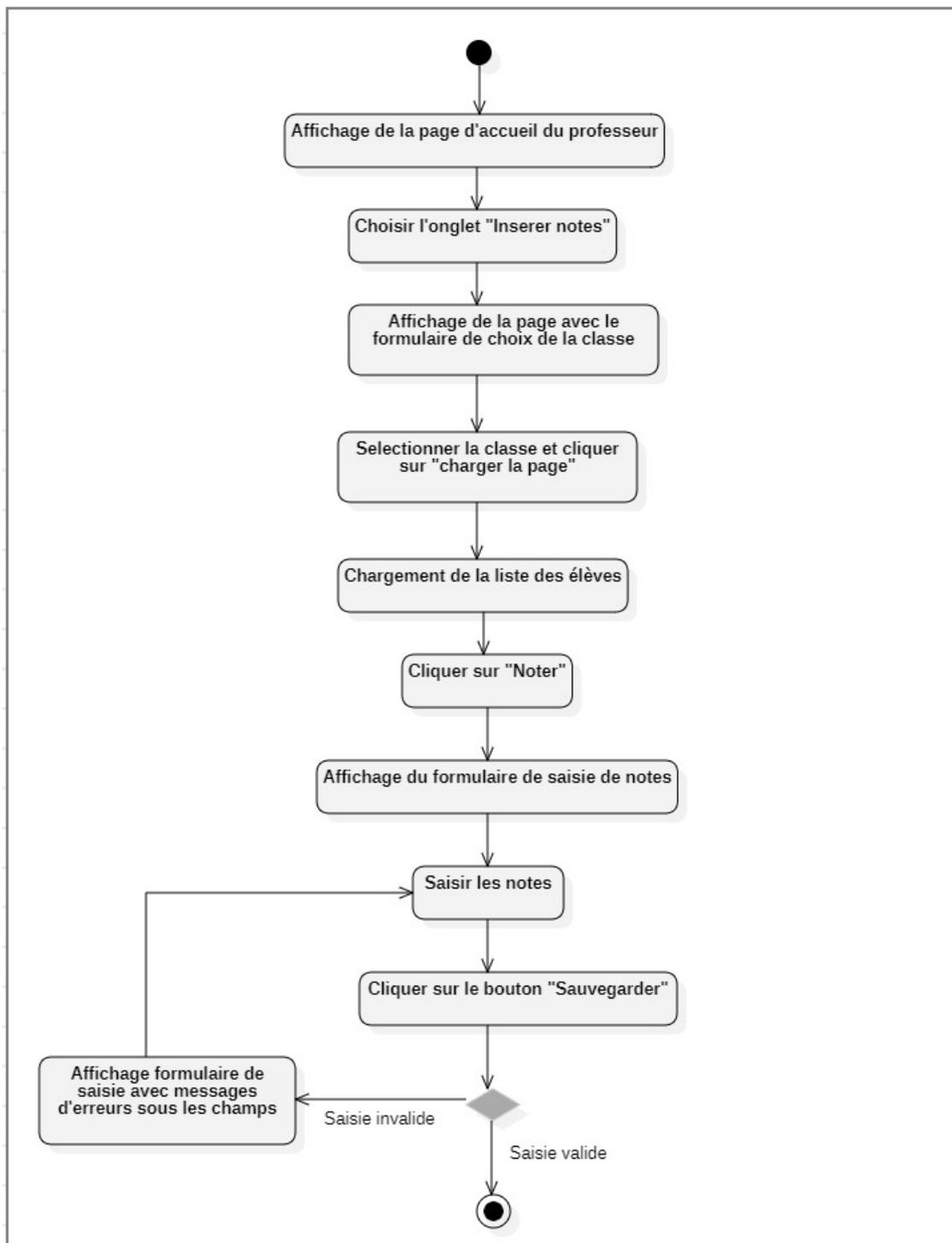


Figure 14: Diagramme d'activité du cas "Ajouter notes"

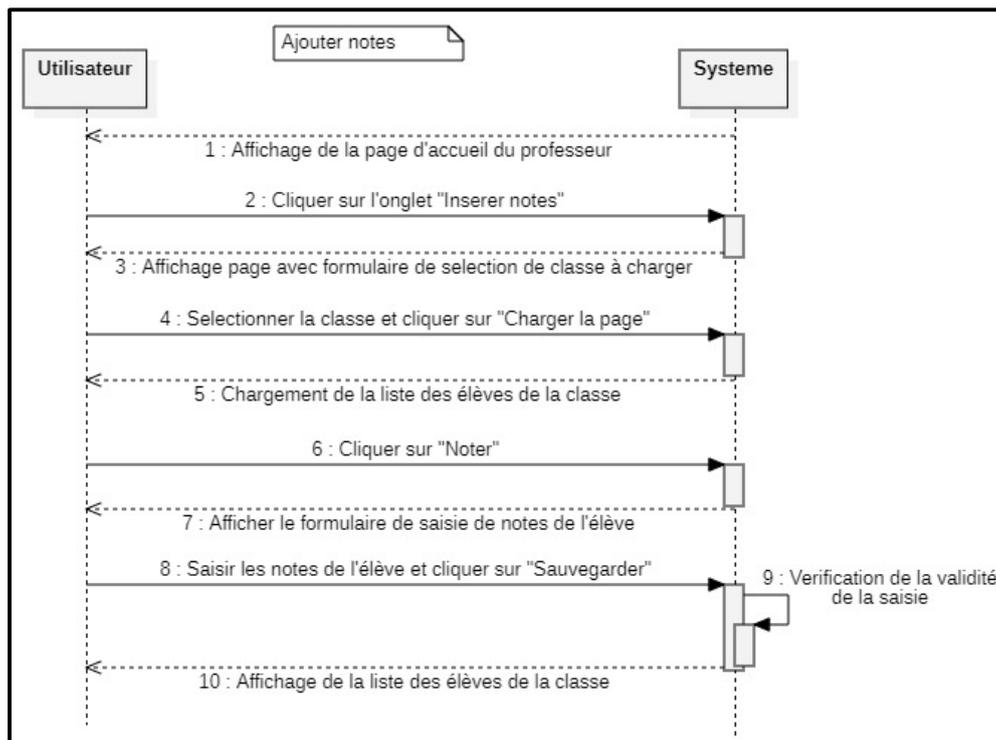


Figure 15: Diagramme de séquences du cas "Ajouter notes"

II.5- Analyse du cas d'utilisation orienter un élève dans une classe

Orienter un élève dans une classe pour une année scolaire donnée est une affaire exclusivement réservée au SG. Ce cas d'utilisation est identique au cas « Réorienter » un élève, leur différence se situe sur le fait que le premier concerne les nouveaux élèves et le deuxième les anciens.

Nous allons donc faire l'analyse de ce cas dans cette partie. Il s'agit de faire sa description puis présenter son diagramme d'activité et son diagramme de séquences.

1- Description du cas d'utilisation « Orienter » un élève

Nous allons faire la description de ce cas d'utilisation avec un tableau (Tableau 10).

Tableau 10: Description du cas d'utilisation "Orienter"

Description du cas d'utilisation « s'authentifier »	
Titre	Orienter un élève
Résumé	Il permet de lier chaque élève à une classe pédagogique pour une année scolaire donnée
Acteurs	SURVGEN (SG)

Pré condition	Authentification, activation de la nouvelle année scolaire, ajout de l'élève dans l'école
Scenario nominal	<ul style="list-style-type: none"> ✓ Accéder à la liste des élèves à orienter par le sous-onglet « Nouvelle orientation » de l'onglet « Elèves-Orientations » ✓ Rechercher l'élève par défilement/pagination ✓ Choisir la classe dans la liste de choix ✓ Cliquer sur le bouton « Orienter »
Post condition	Retour à la liste des élèves à orienter
Exception	Orienter pour une deuxième fois un élève dans la même classe pour la même année scolaire

2- Diagramme d'activité du cas d'utilisation « Orienter »

Pour orienter un élève dans une classe, le SG doit après authentification et accès à sa page d'accueil, cliquer sur le sous-onglet « **Nouvelle orientation** » de l'onglet « **Elèves-Orientations** » pour accéder à la page contenant la liste (formulaire) des élèves à orienter. L'élève à orienter est **recherché par défilement et/ou pagination**. Devant le nom de l'élève le SG **choisit la classe** dans laquelle il sera orienté et il termine en cliquant sur le bouton « **Orienter** ». (Figure 16)

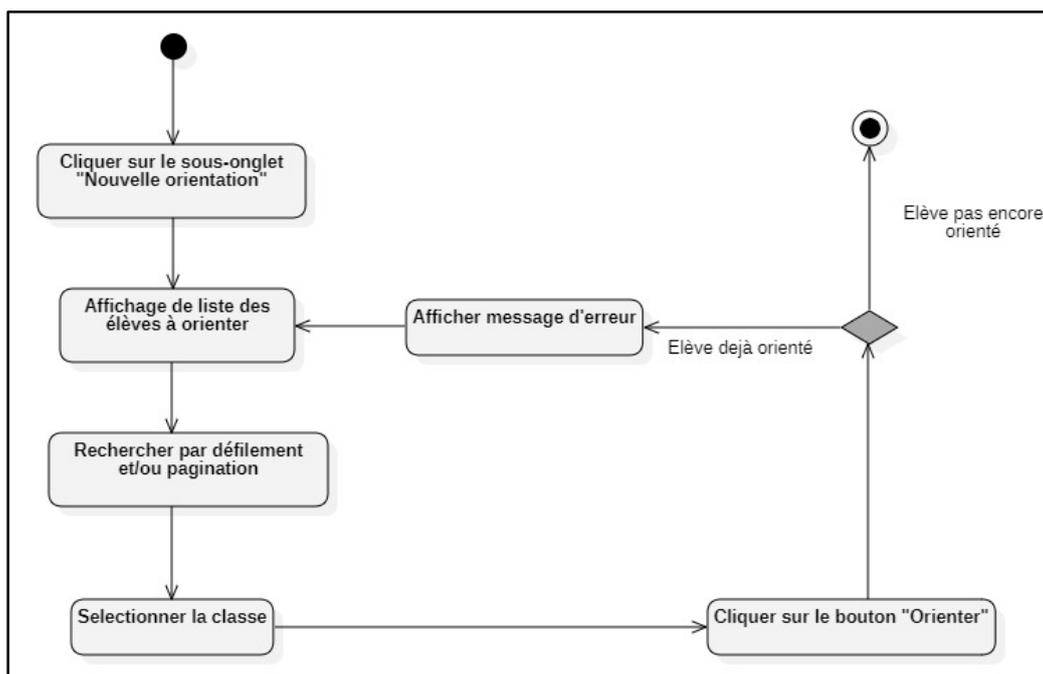


Figure 16: Diagramme d'activité du cas "Orienter"

3- Diagramme de séquences du cas « Orienter »

Afin d'orienter un élève, le SG après avoir accédé à sa page d'accueil par connexion, clique sur le sous-onglet « **Nouvelle orientation** » pour afficher la liste des élèves à orienter. Sur la liste, il cherche le nom du concerné par défilement et/ou pagination, devant celui-ci il sélectionne la classe à partir d'une liste de choix et clique sur le bouton « **Orienter** ». (Figure 17)

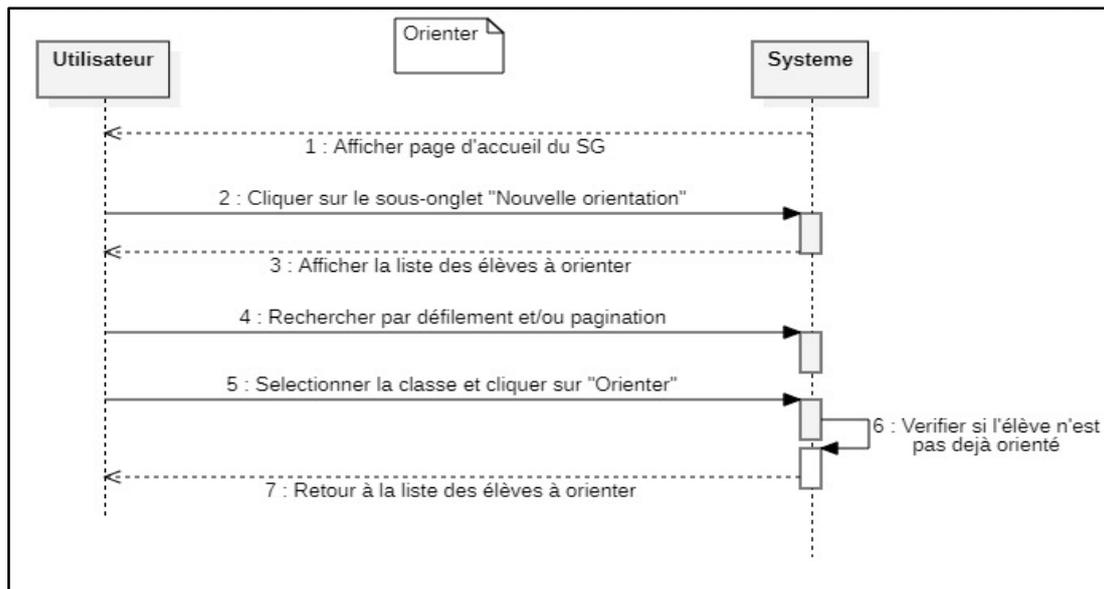


Figure 17: Diagramme de séquences du cas "Orienter"

Nous avons dans ce chapitre présenté la spécification et l'analyse des besoins fonctionnels de notre système. Nous avons aussi modélisé notre système avec des diagrammes de cas d'utilisation, des diagrammes d'activités et les diagrammes de séquences. Ce qui nous a permis de cerner de façon visuelle les contours de notre système.

Ceci étant fait, nous pouvons maintenant nous attaquer à sa conception, c'est l'objet du chapitre suivant.

CHAPITRE III : CONCEPTION DU SYTEME

Dans le chapitre précédent, nous avons étudié notre système dans son aspect fonctionnel. C'est-à-dire répondre aux questions suivantes : Qui utilisera l'application et pour faire quoi ? Comment les actions devront-elles se dérouler ? Quelles informations seront utilisées pour cela ?

Dans ce chapitre, il s'agira d'étudier notre système sous un autre angle, c'est-à-dire sous un angle lié à son architecture. Pour ce faire, nous devons essayer de répondre aux questions suivantes : Quelle style d'architecture client-serveur sera utilisée ? quels sont les composants logiciels à utiliser ? Sur quel matériel chacun de ces composants sera installé ?

Nous décomposerons ce chapitre en deux grandes parties :

- La conception générale avec l'architecture du système, le diagramme de package et le diagramme de de déploiement.
- La conception détaillée avec les diagrammes de classes participantes et le dictionnaire des données.

I- Conception générale

I.1- Architecture de l'application

L'architecture d'une application est la description de façon symbolique et schématique des différents éléments d'un système informatiques, leurs relations, et leurs interactions. Elle ne décrit pas ce que doit faire un système informatique mais comment il doit être fait. Donc il s'agit donc de répondre à la question : comment est-ce que le system doit être fait ?

Dans le cas de notre système, nous avons choisi une architecture trois tiers (trois couches) incluant : la couche de présentation, la couche métier (ou la logique métier) et l'accès aux données.

- **La couche présentation** (niveau 1) : c'est la première couche de l'architecture trois tiers. C'est la partie visible et interactive de l'application pour les utilisateurs. Elle est appelée IHM (Interface Homme Machine), elle permet les interactions entre l'homme (utilisateur) et la machine(application). Elle gère la saisie au clavier, les actions avec la souris et la présentation des informations à l'écran [2]. Nous utiliserons dans notre cas

un moteur de modèles appelé **Thymeleaf** car il est mieux adapté pour diffuser du XML, XHTML et HTML5 sur les vues des applications web basées sur MVC. Nous y avons utilisé aussi du Bootstrap et un peu de JavaScript.

- **La couche métier ou fonctionnelle** (niveau 2) : c'est la seconde couche de l'architecture de notre système. Elle interagit avec les autres couches et sert d'intermédiaire entre le niveau 1 et le niveau 3. L'essentiel du fonctionnement de notre application se situe à ce niveau : les règles de gestion et de contrôle, les traitements, etc. [2] Le langage utilisé dans le cadre de notre projet est **Java**.
- **La couche d'accès aux données** (niveau 3) : troisième couche de notre architecture logicielle. Elle gère l'accès aux données de l'application et leur stockage. Elle correspond au serveur de base de données dans lequel est installé un SGBD [2], dans notre cas MySQL qui utilise un langage de requête appelé SQL.

La **Figure 18** ci-dessous donne un aperçu de l'architecture de notre application

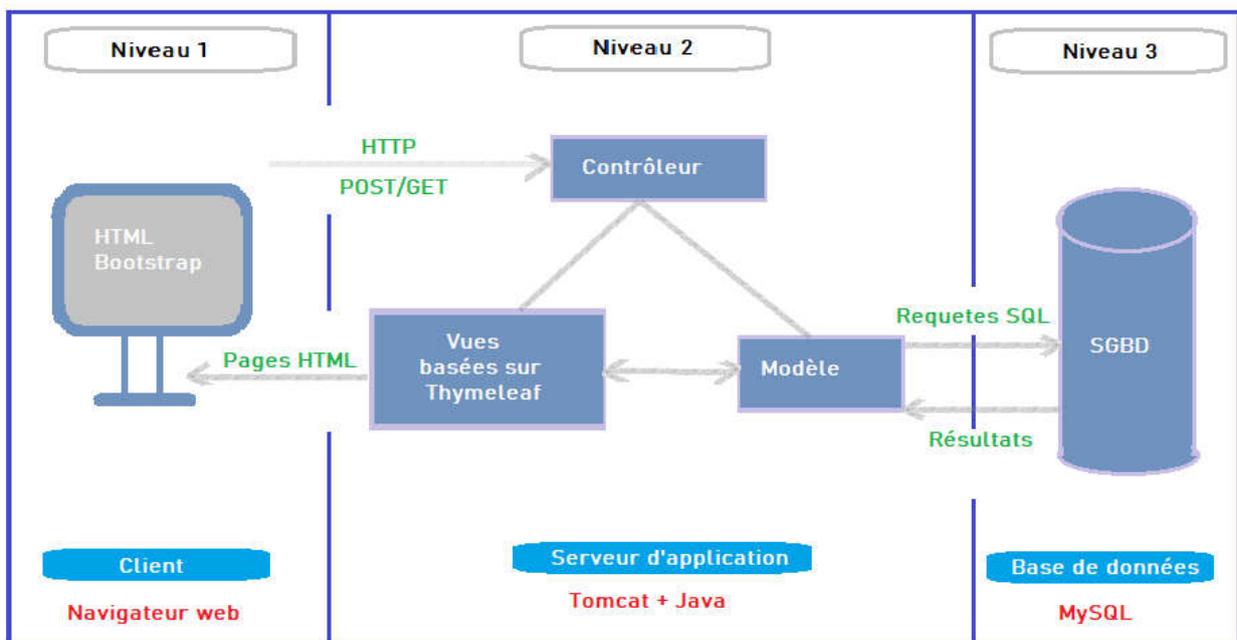


Figure 18: Architecture de l'application

Dans le cas des applications web, le motif d'architecture logicielle destiné aux interfaces graphiques MVC est très populaire. C'est ce motif que nous avons adopté dans la partie serveur d'application afin de faciliter le développement et la maintenance de l'application. Le motif se décompose comme suit :

- Les **modèle** (Model) communiquent avec la base de données.

- Les **vues** (View) sont faites pour la présentation de l'interface utilisateur.
- Les **contrôleurs** (Controller) contiennent les actions effectuées par l'utilisateur.

I.2- Diagrammes de composants

Le diagramme de composants décrit le système modélisé sous forme de composants réutilisables et met en évidence leurs relations de dépendance. Il représente aussi les concepts connus de l'existant pour installer et dépanner le système. Il s'agit de déterminer la structure des composants d'exploitation que sont les instances de base de données, les applications, les objets distribués, les exécutables, etc. Un composant est une unité autonome représentée par un classeur structuré, stéréotypé « *component* », comportant une ou plusieurs interfaces requises ou offertes. Son comportement interne, généralement réalisé par un ensemble de classes, est totalement masqué : seules ses interfaces sont visibles. La seule contrainte pour pouvoir substituer un composant par un autre est de respecter les interfaces requises et offertes. [3]

La figure ci-dessous (**Figure 19**) représente le diagramme de composants de notre système.

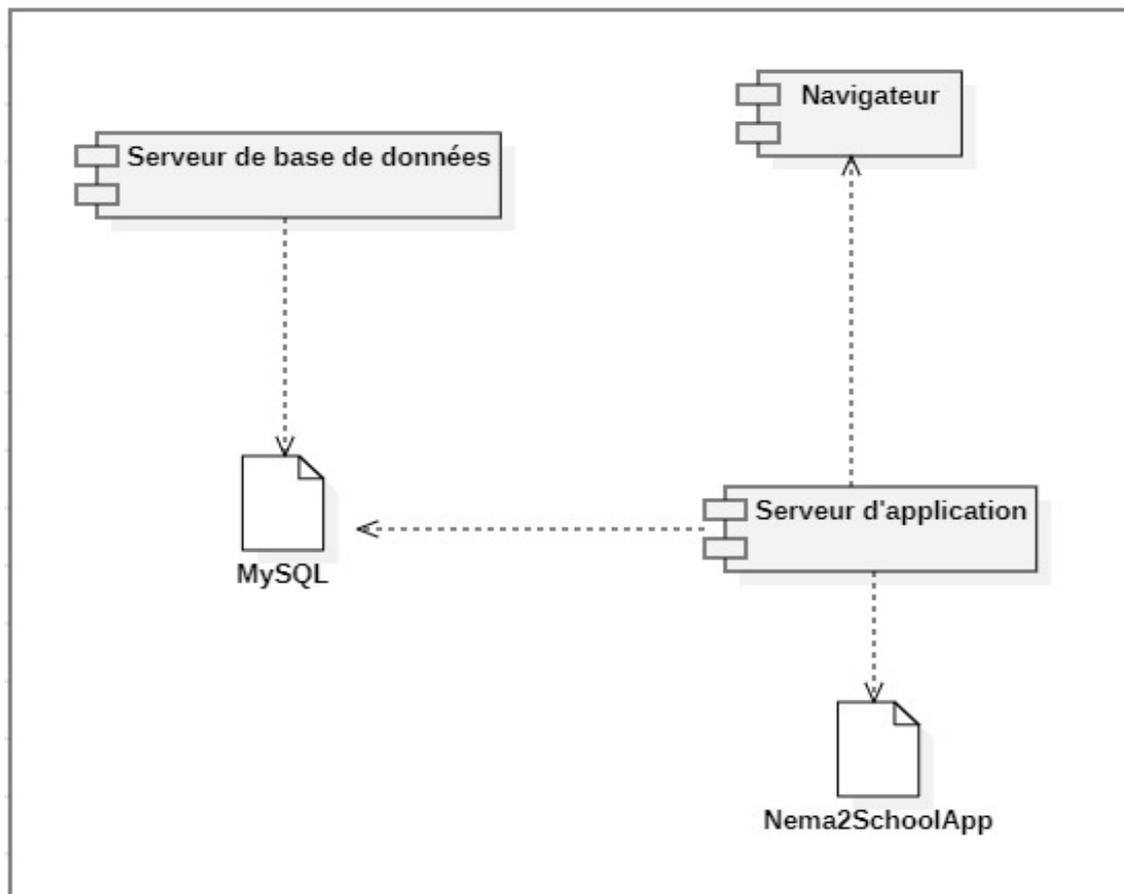


Figure 19: Diagramme de composants

I.3- Diagramme de packages

Un package (paquetage en français) en UML est le regroupement de différents éléments d'un système et fournit un espace de nom pour ces éléments. Un package peut contenir des éléments UML : classes, objets, cas d'utilisation, etc. L'intérêt des packages est de permettre de structurer les diagrammes et de donner une vision globale avec clarté [3]. Le diagramme de package de notre système est donné ci-dessous. (**Figure 20**)

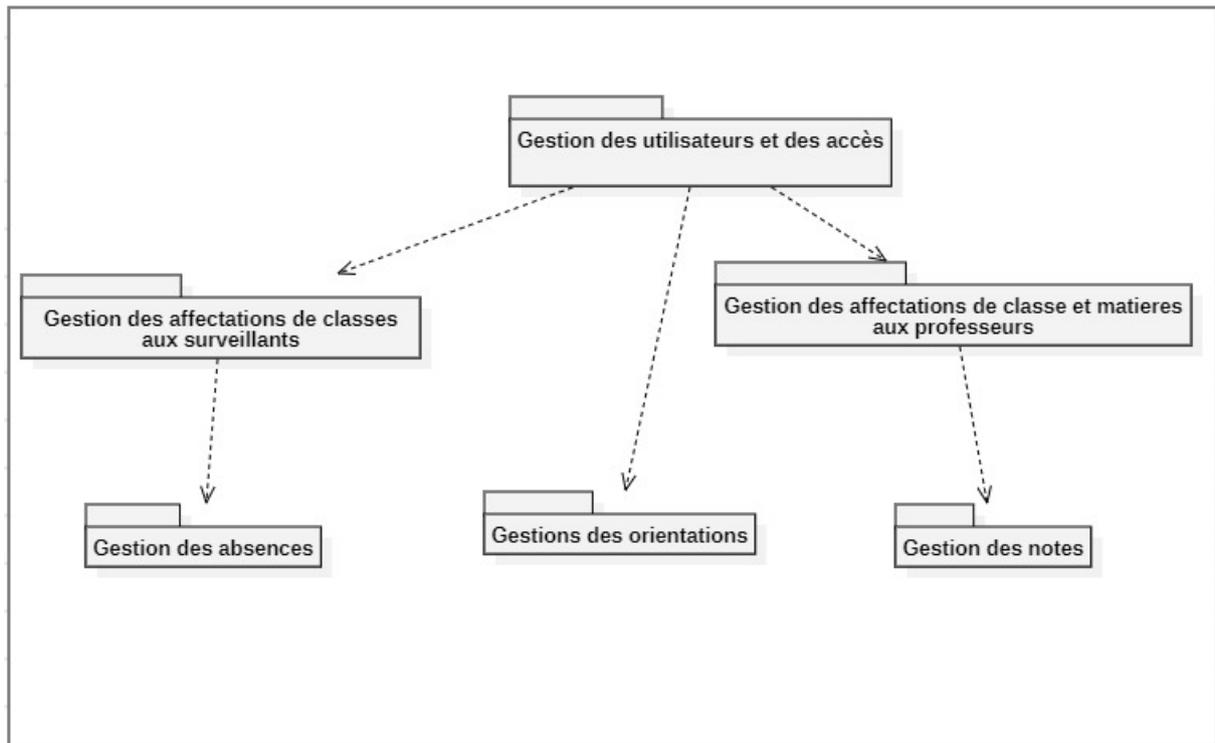


Figure 20: Diagramme de package

Notre diagramme de package ci-dessous est composé de 6 packages :

- ✓ **Gestion des utilisateurs et des accès** : les fonctionnalités présentes dans ce package sont ajouter, modifier, archiver un utilisateur ; ajouter, modifier, supprimer un rôle ; octroyer un rôle, supprimer un octroie. Signalons que les accès sont fonction du rôle ou des rôles de l'utilisateur.
- ✓ **Gestion des affectations de classes aux surveillants** : dans ce package, nous retrouvons les fonctionnalités comme : afficher la liste des surveillants, affecter des classes à un surveillant, modifier une affectation, supprimer une affectation et voir la liste des affectations de tous les surveillants.
- ✓ **Gestion des absences** : dans ce package nous avons comme fonctionnalités : la saisie du nombre d'absences d'un semestre pour chaque classe, lister les saisies et les modifier.

- ✓ **Gestion des affectations de classes et matières aux professeurs** : dans ce package nous avons comme fonctionnalités : lister l'ensemble des professeurs, affecter des classe-matière aux professeurs, modifier ou supprimer une affectation, lister l'ensemble des affectations d'une années scolaire.
- ✓ **Gestion des notes** : par gestion des notes, il faut entendre la saisie, l'affichage et la modification de notes par semestre.
- ✓ **Gestion des orientations** : ce package contient les fonctionnalités suivantes : orienter un élève dans une classe pour une année scolaire donnée, modifier une orientation, supprimer une orientation et lister les élèves par classe.

I.4- Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication serveurs et routeurs peuvent être précisées par stéréotype [1]. (Figure 21)

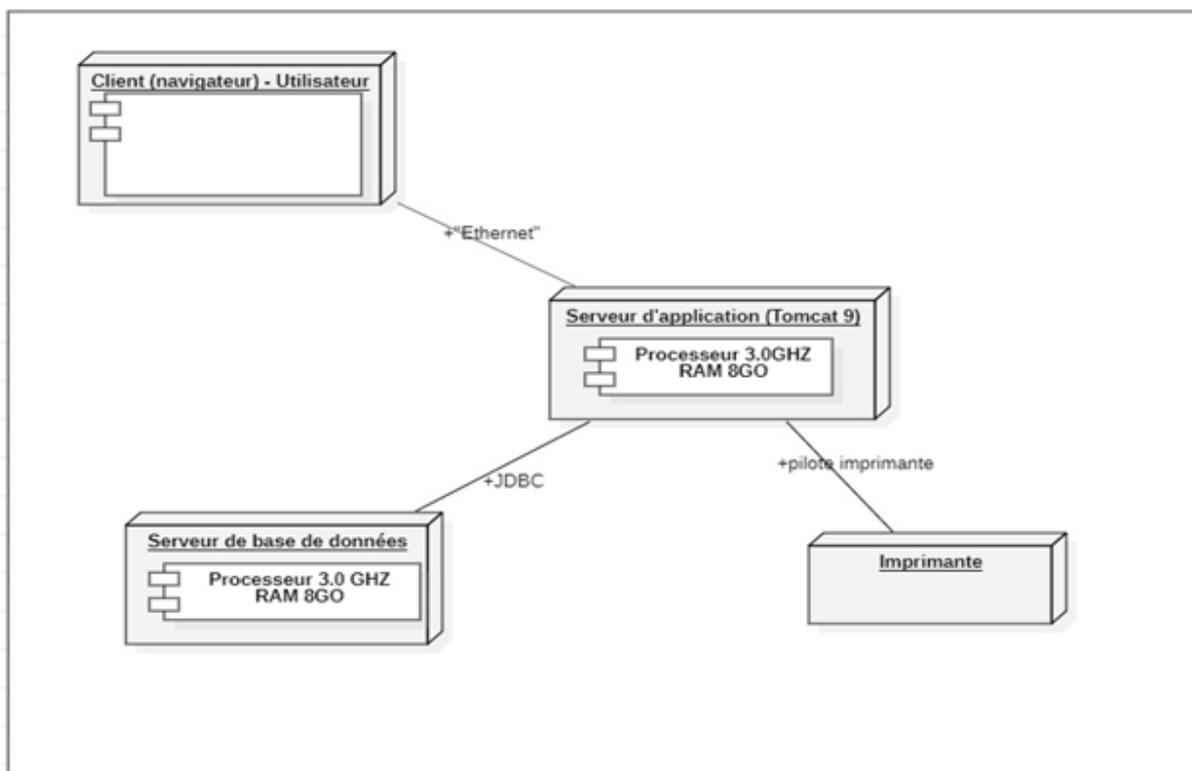


Figure 21: Diagramme de déploiement

La figure ci-dessus nous donne les caractéristiques techniques idéales du matériel nécessaire au déploiement de notre système. Mais le déploiement peut se faire sur du matériel avec des caractéristiques différentes de celles indiquées et qui respectent le minimum requis.

II- Conception détaillée

La conception détaillée est la dernière phase de la conception de notre système. Nous allons dans cette partie, modéliser le diagramme de classes et terminer par le dictionnaire de données.

II.1- Diagramme de classes

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Elles sont utilisées dans la programmation orientée objet. Les classes permettent de modéliser un programme. Elles peuvent être liées entre elles par association, héritage, agrégation, composition, etc. [3]

Le diagramme de classes est un schéma utilisé pour représenter les classe et les interfaces d'un système ainsi que les différentes relations entre celles-ci. [3]

Nous n'allons pas représenter le diagramme de classes de notre système au complet. Mais plutôt le faire par classes participantes aux principales fonctionnalités.

II.1.1- Diagramme de classes participantes aux fonctionnalités de la gestion des utilisateurs et des accès

Seules trois classes participent dans ce diagramme, elles sont : **Agent**, **AgentRole** et **Role**. Tous les agents sont des utilisateurs, la distinction se fait à partir du champ fonction. L'agent est soit principal, soit professeur, soit surveillant général, soit surveillant et jouent le même rôle dans notre système. Certains pourront jouer le rôle d'administrateur qui va s'ajouter à sa fonction dans l'établissement. (**Figure 22**)

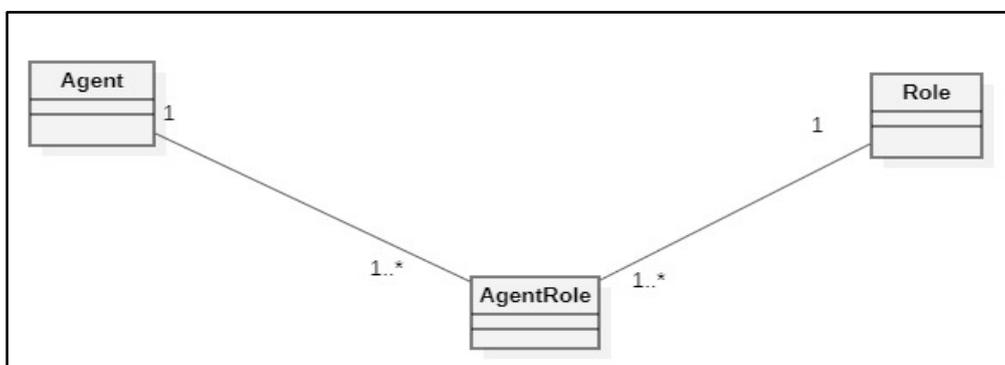


Figure 22: Diagramme de classes participantes aux fonctionnalités de la gestion des utilisateurs et des accès

II.1.2- Diagramme de classes participantes aux fonctionnalités de la gestion des affectations de classes aux surveillants

Les classes participantes dans ce diagramme sont : **Agent**, **Classe**, **SchoolYear** et **AffecterClasseSurv**. A chaque surveillant, le surveillant général affecte des classes pour une année scolaire donnée. Pour ce faire, l'administrateur doit activer au préalable la nouvelle année scolaire. (**Figure 23**)

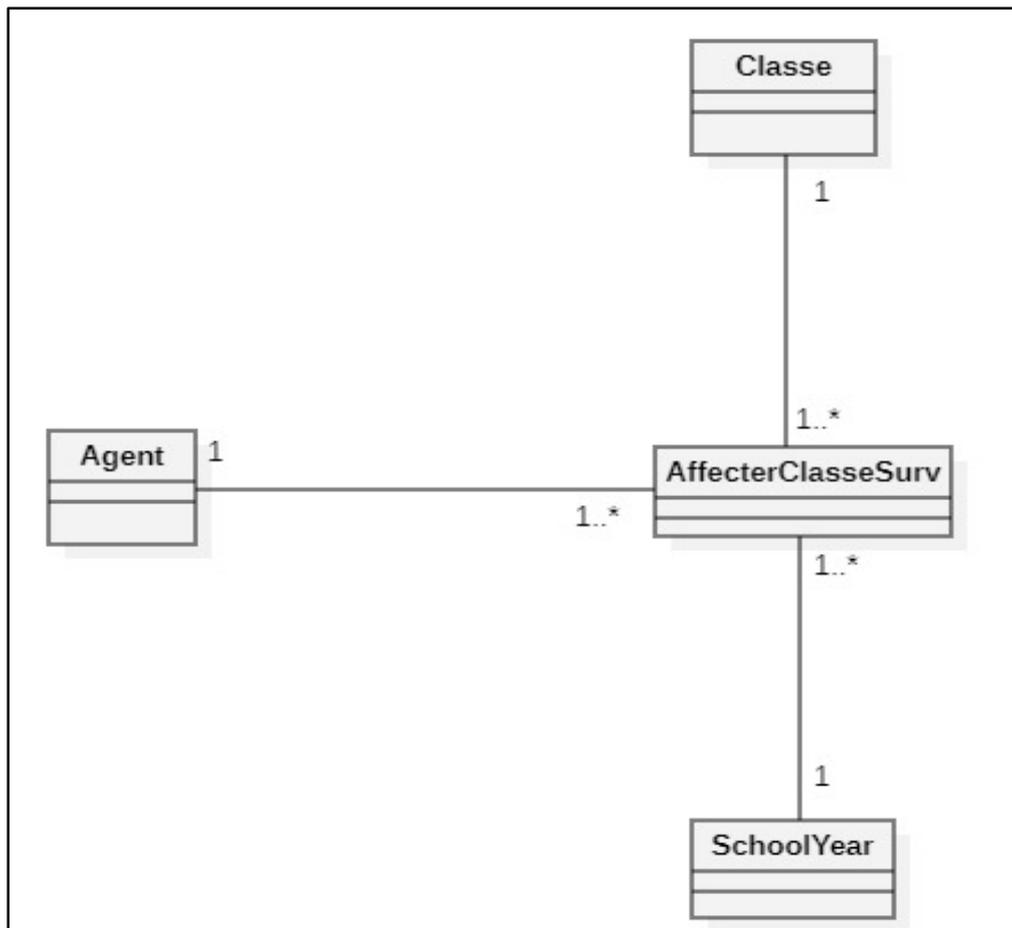


Figure 23: Diagramme de classes participantes aux fonctionnalités de la gestion des affectations de classes aux surveillants

II.1.3- Diagramme de classes participantes aux fonctionnalités de la gestion des absences

Dans ce diagramme, nous retrouvons cinq classes participantes, que sont : **Eleve**, **Classe**, **SchoolYear**, **Semestre** et **Absence**. Il permet à chaque surveillant de saisir le nombre total d'absences de chaque élève pendant le semestre. (**Figure 24**)

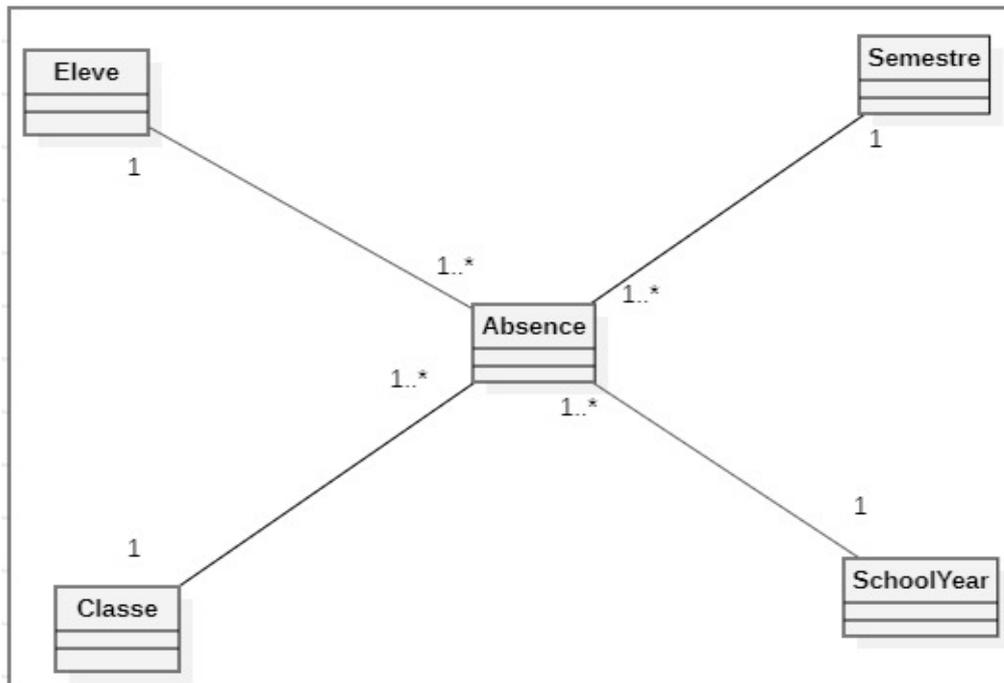


Figure 24: Diagramme de classes participantes aux fonctionnalités de la gestion des absences

II.1.4- Diagramme de classes participantes aux fonctionnalités de la gestion des orientations

Les classes de ce diagramme sont : **Eleve**, **Classe**, **SchoolYear** et **Orientation**. Il s'agit ici d'associer chaque élève à une classe pédagogique pour une année scolaire. (Figure 25)

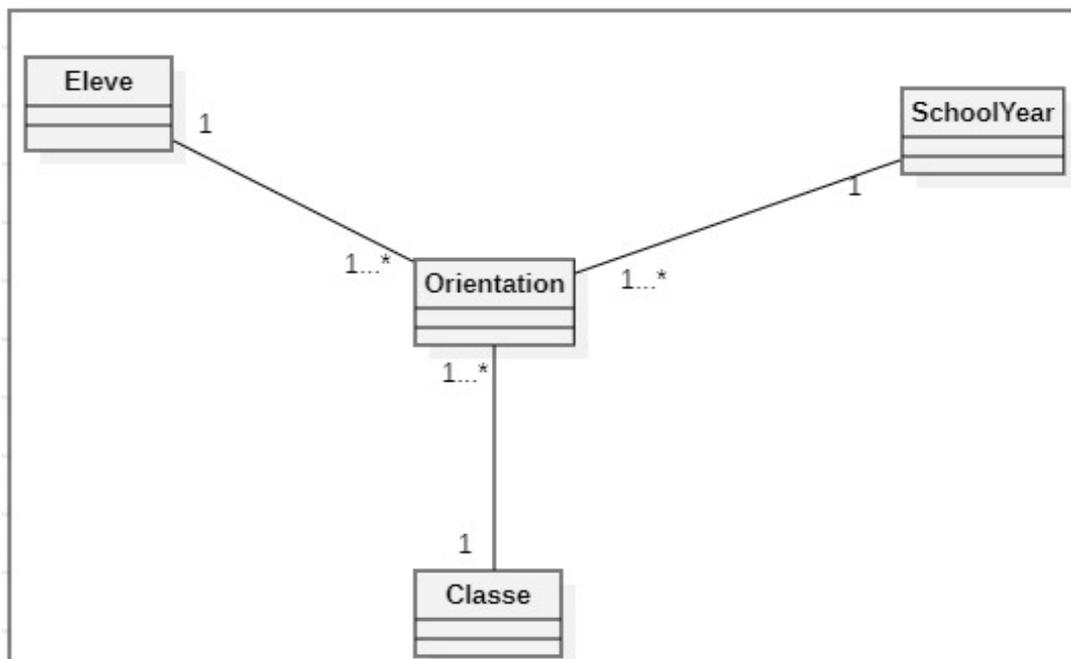


Figure 25: Diagramme de classes participantes aux fonctionnalités de la gestion des orientations

II.1.5- Diagramme de classes participantes aux fonctionnalités de la gestion des affectations de classes et matières aux professeurs

Les classes intervenant dans ce diagramme de classes sont : **Agent**, **Classe**, **Matiere**, **SchoolYear** et **AffecterClasseProf**. Comme l'indique le titre, dans ce diagramme il s'agit de se référer sur l'emploi du temps pour affecter des classes et des matières à tous les professeurs pour une année scolaire donnée. (**Figure 26**)

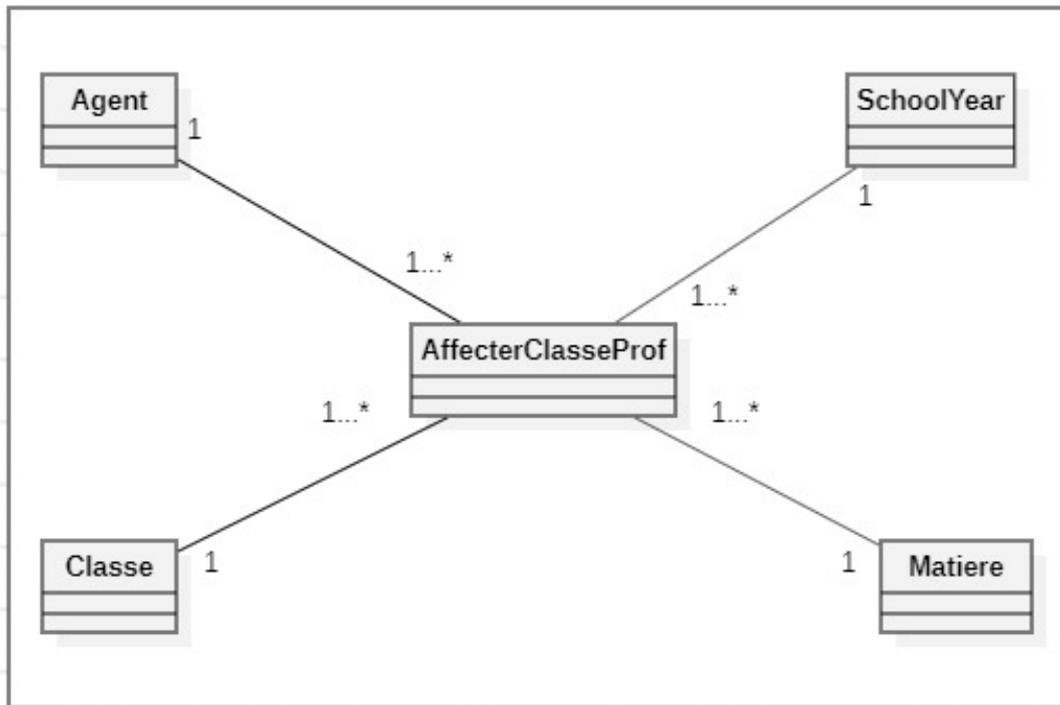


Figure 26: Diagramme de classes participantes aux fonctionnalités de la gestion des affectations de classes et matières aux professeurs

II.1.6- Diagramme de classes participantes aux fonctionnalités de la gestion des notes

Les classes intervenant dans ce diagramme sont : **Eleve**, **Classe**, **Matiere**, **SchoolYear**, **Semestre** et **Note**. Ce diagramme permet que les professeurs saisissent ou modifient les notes des élèves. Un professeur ne peut ni choisir l'année scolaire, ni le semestre dans le formulaire de saisie ou de modification. C'est pour empêcher à un professeur de revenir sur ses saisies après la génération et l'impression des bulletins. (**Figure 27**)

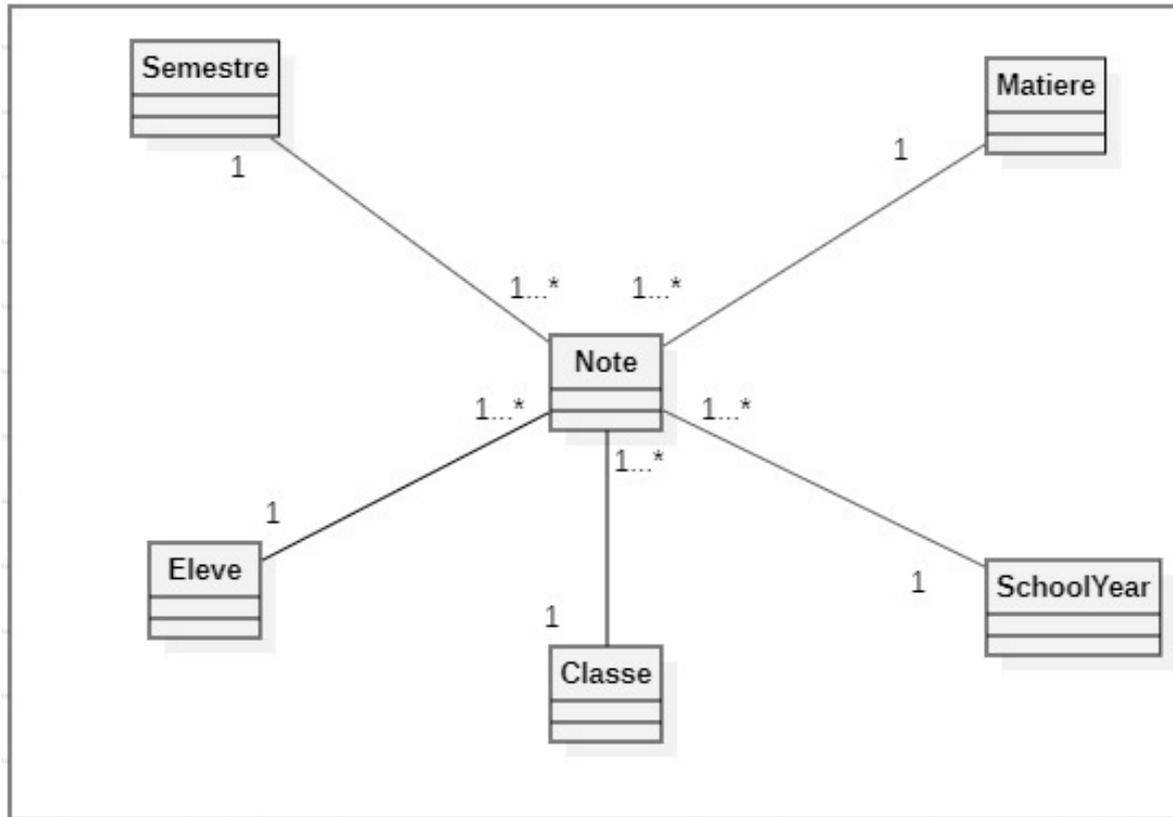


Figure 27: Diagramme de classes participant aux fonctionnalités de la gestion des notes

II.2- Dictionnaire de données

Un dictionnaire des données est une collection de métadonnées ou de données de référence nécessaire à la conception d'une base de données relationnelle.

Dans notre projet, il est représenté par le tableau ci-dessous qui a quatre colonnes que sont : noms des tables, les champs (données), libellé et le type.

Tableau 11: Dictionnaire de données

Nom de table	Champs	Libellé	Type
Agent	matricule	Matricule (identifiant) de l'agent (utilisateur)	String
	motPasse	Mot de passe de l'agent (utilisateur)	String
	nom	Nom de l'agent	String
	prenom	Prenom de l'agent	String
	fonction	La fonction de l'agent dans l'établissement	String

	statutAgent		Boolean
Role	role	Le nom (identifiant) du rôle qui est donné à un agent	String
	description	La description de rôle	String
AgentRole	idAgentRole	Identifiant d'une association entre un agent et un rôle	Long
	matricule	Matricule (identifiant) de l'agent (utilisateur)	String
	rôle	Le nom du rôle qui est donné à un agent	String
Classe	nomClasse	Le nom (identifiant) de la classe	String
	niveau	Le niveau de la classe (niveau 1 : 6 ^{ième} , niveau 2 : 5 ^{ième} , niveau 3 : 4 ^{ième} , niveau 4 : 3 ^{ième})	String)
SchoolYear	anneScolaire	Année scolaire (identifiant) au format : 2019-2020	String
	selectedAS	Indique si l'année scolaire est sélectionnée ou non	Boolean
Matiere	nomMatiere	Nom de la matière (identifiant)	String
	coefficient	Coefficient de la matière	Integer
Semestre	nomSemestre	Nom du semestre (identifiant)	String
	selectedSEM	Indique si le semestre est sélectionné ou non	Boolean
Eleve	codeEleve	Identifiant de l'élève	Integer
	nomElv	Nom de l'élève	String
	prenomElv	Nom de l'élève	String
	dateNaiss	La date de naissance de l'élève	Date
	lieuNaiss	Le lieu de naissance de l'élève	String
	classeRed	Les classes redoublées	String
	sexe	Le sexe de l'élève	String
	anneeArv	L'année d'arrivée de l'élève	String

	statutElv	Indique le statut de l'élève : archivé ou pas	Boolean
Orientation	idOrienta	Identifiant d'une orientation	Long
	codeEleve	Identifiant de l'élève	Integer
	nomClasse	Identifiant la classe	String
	anneScolaire	Identifiant de l'année scolaire	String
AffecterClassSurv	idAffectClassSurv	Identifiant d'une affectation de classe à un surveillant	Long
	matricule	Identifiant d'un agent	String
	nomClasse	Identifiant la classe	String
	anneScolaire	Identifiant de l'année scolaire	String
AffecterClassProf	idAffectClassProf	Identifiant d'une affectation de classe et matière à un professeur	Long
	matricule	Identifiant d'un agent	String
	nomClasse	Identifiant la classe	String
	nomMatiere	Identifiant de la matière	String
	anneScolaire	Identifiant de l'année scolaire	String
Note	idNote	Identifiant d'une note	Long
	codeEleve	Identifiant de l'élève	Integer
	nomClasse	Identifiant la classe	String
	nomMatiere	Identifiant de la matière	String
	anneScolaire	Identifiant de l'année scolaire	String
	nomSemestre	Identifiant du semestre	String
	devoir1	Note du devoir 1	Double
	devoir2	Note du devoir 2	Double
	devoir3	Note du devoir 3	Double
	composition	Note de la composition	Double
Absence	idAbsence	Identifiant d'une absence	Long
	codeEleve	Identifiant de l'élève	Integer
	nomClasse	Identifiant la classe	String
	anneScolaire	Identifiant de l'année scolaire	String
	nomSemestre	Identifiant du semestre	String

	nombreAbsence	Le nombre d'absence pendant un semestre	Integer
Ecole	nomEcole	Identifiant de l'école	String
	IA	L'inspection d'académie	String
	IEF	L'inspection de l'éducation et de la formation	String
	telephone	Le numéro de téléphone de l'école	String
	schoolEmail	L'email de l'école	String

Nous avons dans ce chapitre parlé de l'architecture de notre système, présenté les diagrammes de composants, de packages et de déploiement dans la partie qui concerne la conception générale. Dans la conception détaillée de l'application nous avons présenté les diagrammes de classes. Ce chapitre nous permet de bien préparer le prochain chapitre de notre projet, à savoir l'implémentation de notre application.

CHAPITRE IV : IMPLEMENTATION DE L'APPLICATION

L'implémentation (ou codage) est une partie principale de ce chapitre qui prend en considération le travail de conception qui a été fait dans le chapitre précédent. Mais avant, nous présenterons les outils et technologies utilisées (outils de conception, de développement, Framework, etc.).

I- Outils et technologies utilisées

Le développement de notre application « **Nema2SchoolApp** » a nécessité l'utilisation de plusieurs outils et technologies. Nous allons dans ce qui suit faire une brève présentation de chacun.

I.1- Outils de conception : StarUML

StarUML est un logiciel de modélisation UML. Il est écrit en Delphi et dépend de composants Delphi propriétaires. Il gère la plupart des diagrammes spécifiés dans la norme UML 2.0. L'export peut se faire dans les formats JPEG, WMF, SVG et PNG (formats d'images) [5].

I.2- Outils de développement : Eclipse

Eclipse est un environnement de développement intégré (IDE) écrit en Java, extensible par des greffons, multi-langages et multi-plates-formes. Conçu à l'origine pour le langage Java, il est devenu un IDE pour beaucoup de langages de programmation (C, C++, Python, PHP, ...). Nous avons téléchargé et installé par **marketplace** (espace de téléchargement d'Eclipse) **Spring Tool Suite 4** pour pouvoir travailler avec le Framework Spring Boot.

I.3- Le SGBDR : MySQL

MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous double licence libre et propriétaire. MySQL fait partie des logiciels de gestion de bases de données les plus utilisés au monde, autant par le grand public que par des professionnels. Parmi ses concurrents nous pouvons citer : Oracle Database, PostgreSQL, Microsoft SQL Server, MariaDB, ... Dans le cadre de notre projet nous avons utilisé MySQL qui est contenu dans WampServer, donc nous avons installé WampServer afin d'utiliser MySQL [6].

I.4- Framework Bootstrap

Bootstrap est une collection d'outils utiles à la création du design de site et d'application web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option [7].

I.5- Langages de programmation : Java et JavaScript

Java est un langage de programmation 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 a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation [8].

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel [9].

I.6- Langage de balisage : HTML

Les langages de balisage représentent une classe de langages spécialisés dans l'enrichissement d'information textuelle. Ils utilisent des balises, unités syntaxiques délimitant une séquence de caractères ou marquant une position précise à l'intérieur d'un flux de caractères (par exemple un fichier texte). L'HyperText Markup Language, généralement abrégé HTML est un langage de balisage parmi tant d'autres (DocBook, TEI, WML, etc.). Autrement dit, c'est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom [1].

I.7- Outil de création de rapports (Jaspersoft Studio)

Jaspersoft est un plugin Eclipse équivalent à l'outil de reporting libre **JasperReports** basé sur des fichiers XML (Extensible Markup Language) avec une extension «**.jrxml**» pour la présentation des états. Il est offert sous forme d'une bibliothèque qui peut être embarquée dans tous types d'application Java. Nous l'avons utilisé spécifiquement pour les bulletins de notes, les listes d'appel, les fiches de notes, etc.

I.8- Framework Spring (Spring Boot)

Spring est un Framework open-source (gratuit) destiné aux applications java (J2EE) pour les applications 3-tiers dont il facilite le développement et les tests.

Ce Framework est très apprécié par la communauté Java pour sa souplesse d'utilisation et sa modularité. En effet, il n'est pas nécessaire d'implémenter la totalité de Spring pour pouvoir l'utiliser, et c'est ce qui en fait son principal point fort, vous n'utilisez que ce dont vous avez besoin.

Spring et Spring MVC sont de bons outils de développement, mais nécessitent beaucoup de configurations. Les applications développées avec ces Framework possèdent beaucoup de fichiers XML qui indiquent les configurations des servlets, des vues, des contenus statiques, etc.

Spring Boot rend beaucoup plus facile le travail de configuration en proposant deux fonctionnalités principales :

- L'auto-configuration : une fonctionnalité très importante, elle permet de configurer automatiquement notre application à partir des jars de notre classpath. C'est-à-dire Spring Boot parcourt la liste des dépendances importées et produit les configurations nécessaires pour que tout fonctionne normalement.
- Les starters : ils apportent à notre projet un ensemble de dépendances communément utilisées pour un type de projet donné. Ils viennent compléter l'auto-configuration et font gagner énormément de temps. Un autre avantage pas le moindre est la gestion des versions. Il suffit simplement d'ajouter une dépendance au starter de notre choix dans le fichier **pom.xml**, cette dernière va alors ajouter, à son tour, les éléments dont elle dépend, avec les bonnes versions.

Exemples :

- [Spring-boot-starter-mail](#) : pour les applications et services d'envoi de mails.

- Spring-boot-starter-thymeleaf : si nous souhaitons créer une application qui offre une interface utilisateur en utilisant le moteur de template thymeleaf.

1.8.1- Thymeleaf

Thymeleaf est un moteur de Template (thème), sous licence Apache 2.0, écrit en Java pouvant générer du XML/XHTML/HTML5. Thymeleaf peut être utilisé dans un environnement web (utilisant l'API Servlet) ou non. Son but principal est d'être utilisé dans un environnement web pour la génération de vue pour les applications web basées sur le modèle MVC [10].

1.8.2- Tomcat

Tomcat est un serveur HTTP à part entière. De plus, il gère les servlets et les JSP (par un compilateur Jasper compilant les pages JSP pour en faire des servlets). Il a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant [11].

II- Implémentation

II.1- Implémentation de la base de données

II.1.1- Modèle physique de données

Le modèle physique de données (MPD) consiste à implémenter une base de données dans un SGBDR, c'est-à-dire la traduire dans un langage de définition de données. Le langage utilisé pour ce type d'opération est le SQL.

La migration des clés primaires dépend des cardinalités des tables. Les clés primaires migrent vers les tables où les cardinalités sont minimales. Dans ce modèle, les clés primaires sont soulignées tandis que les clés étrangères sont précédées par un dièse « # ».

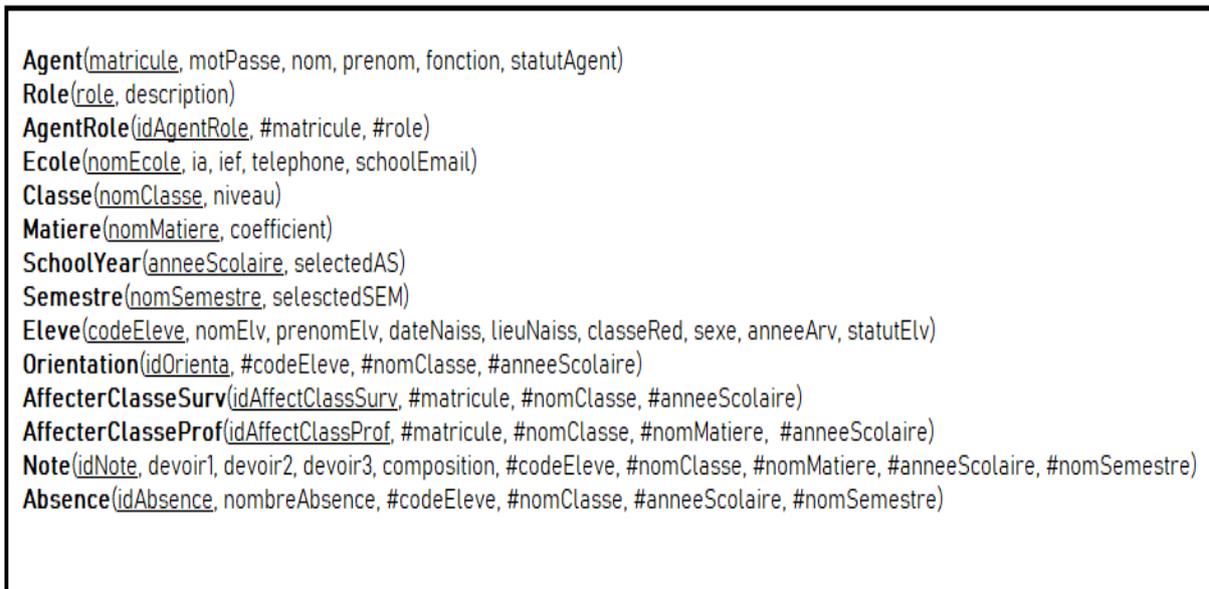


Figure 28: Schéma relationnel du MPD

II.1.2- Création de la base de données sous MySQL de WampServer

Nous démarrons WampServer afin d'accéder à l'interface de connexion de « phpMyAdmin ». Nous nous connectons au serveur MySQL avec root comme identifiant et sans mot de passe. A ce niveau nous créons notre base de données appelée « **nema2schoolapp_bd** ».

Nous allons lier l'application à la base de données en passant par le fichier « **application.properties** » en ajoutant les lignes indiquées dans l'image ci-dessous.

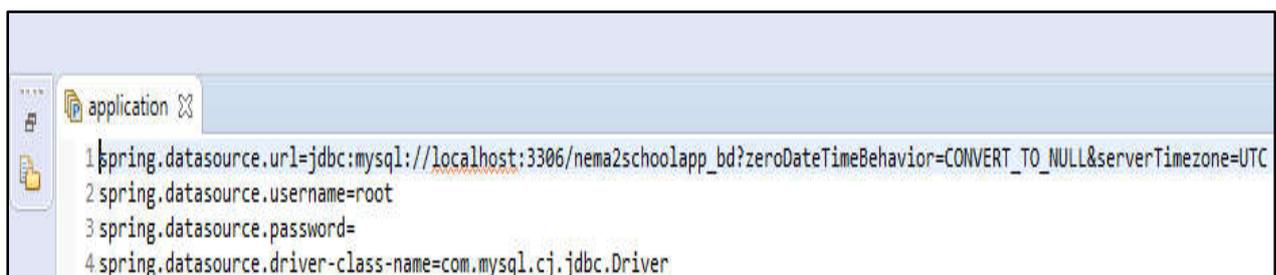


Figure 29: Fichier application.properties

La ligne numéro 1 peut être écrite comme suit :
`spring.datasource.url=jdbc:mysql://localhost:3306/mydatabase`

Dans notre cas il fallait ajouter tout ce qui est après le point d'interrogation à cause d'une erreur de fuseau horaire de mon ordinateur.

Après avoir implémenté toutes les classes et démarré l'application, les tables seront créées automatiquement.

II.2- implémentation de l'application

II.2.1- Création et architecture de l'application

Lors de la création d'une application web Java EE avec Spring Boot dans Eclipse, nous avons certains choix à faire par exemples.

- Le type : Maven ou Gradle qui sont des outils pour gérer les bibliothèques (**Maven**).
- Le packaging de l'application : JAR ou WAR, fichier autonome et exécutable pour le premier ou fichier pour le déploiement sur un conteneur de servlet externe (**JAR** pour l'instant).
- La version de Java : Java Version (**Java 8**).
- Le langage (langage en anglais) : Java, Kotlin, Groovy (**Java**).
- La version de Spring Boot: Spring Boot Version (**2.1.6. RELEASE**).
- Les dependances nécessaires pour application: **Spring Web, Spring Data JPA, MySQL, Thymeleaf, Spring Security, DevTools**, etc. Il n'est pas nécessaire d'ajouter d'un seul coup toutes les dépendances, il est possible de le faire après dans le fichier **pom.xml**.

La figure (**Figure 30**) nous montre de l'architecture générale de notre application.

II.2.2- Codage d'une entité

Avec **JPA (Java Persistence API)** les programmeurs n'ont plus besoin d'être des spécialistes en base de données ou d'un SGBD quelconque. JPA leur permet d'implémenter les classes (entités) qui seront transformées en tables dans une base de données. JPA permet aux programmeurs d'utiliser des notions avancées en base de données tout en restant programmeurs.

@Entity, signifie que cette classe est une table. JPA créera une table de même nom que la classe dans notre base de données.

@Table (name= "nom de la table"), permet de donner un nom différent de la classe à la table.

@Id, signifie que l'attribut qui est en dessous est la clé primaire de la table.

@GeneratedValue, signifie que la clé primaire est générée automatiquement par le SGBD.

@OneToMany, traduit la relation : un élément d'une entité est associé à un ou plusieurs éléments d'une autre entité.

La figure (Figure 31), montre un exemple d'implémentation d'une classe.

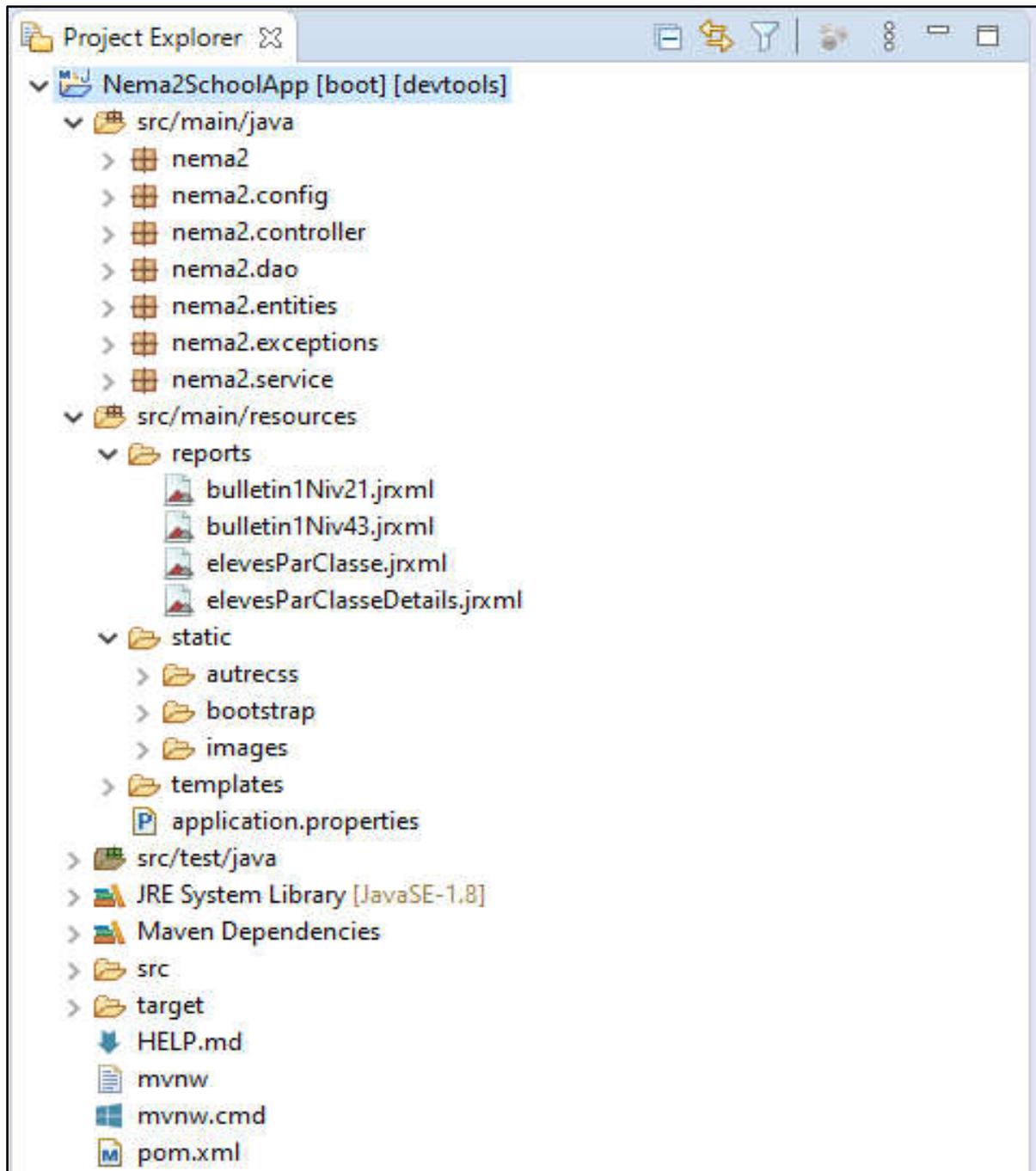


Figure 30: Architecture de l'application

```
Nema2SchoolApp/pom.xml | Eleve.java | Agent.java
1 package nema2.entities;
2
3 import java.io.Serializable;
15
16 @Entity
17 @Table(name = "agent")
18 public class Agent implements Serializable{
19     /**
20      *
21      */
22     private static final long serialVersionUID = 1L;
23     @Id
24     @NotEmpty
25     @Length(min = 8,max = 8,message = "un matricule doit avoir exactement 8 caracteres et ecrit sous la forme")
26     @Column(name = "matricule")
27     private String matricule;
28     @NotNull
29     @Size(min = 3, max = 30)
30     private String password;
31     @NotNull
32     @Size(min = 4, max = 30)
33     private String nom;
34     @NotNull
35     @Size(min = 4, max = 50)
36     private String prenom;
37     @NotNull
38     @Size(min = 4, max = 20)
39     private String fonction;
40
41     private boolean statutAgent;
42
43     @OneToMany(mappedBy = "agent")
44     private Collection<AgentRole> agentRoles;
45
46     @OneToMany(mappedBy = "agent")
47     private Collection<AffectClassProf> affectClassProf;
48
```

Figure 31: Entité "Agent"

II.2.3- Codage d'une interface

Toutes les interfaces de notre application héritent de l'interface **JpaRepository**. Ceci permet l'utilisation des opérations CRUD (Create, Read, Update et Delete) auto-générées c'est-à-dire les opérations (insertion, lecture, mise à jour et suppression), vers la base de données sans avoir à écrire la moindre requête, ni même la moindre implémentation **DAO** (en anglais **Data Access Object** ou **Objet d'Accès aux Données** en français).

Pour certaines opérations spécifiques ou particulières, nous avons utilisé des requêtes **HQL** (**Hibernate Query Language**) qui est un langage de requête du Framework **Hibernate**. Il est presque similaire au **SQL**.

La figure (**Figure 32**) ci-dessous, nous montre un exemple d'implémentation d'une interface.

```
AgentControlleur.java AgentRepository.java
1 package nema2.dao;
2
3 import org.springframework.data.domain.Page;
10
11 public interface AgentRepository extends JpaRepository<Agent, String>{
12     @Query("FROM Agent agt")
13     public Page<Agent> listAgents(Pageable pageable);
14
15     @Query("SELECT a FROM Agent a WHERE a.fonction='professeur'")
16     public Page<Agent> listProfs(Pageable pageable);
17
18     @Query("SELECT a FROM Agent a WHERE a.fonction='surveillant'")
19     public Page<Agent> listSurvs(Pageable pageable);
20
21     @Query("SELECT a FROM Agent a WHERE a.matricule=:x") //Le Agent indique le nom de la class
22     public Agent findOne(@Param("x")String matricule);
23
24     @Query("SELECT a FROM Agent a WHERE a.fonction=:x")
25     public Agent findByFonction(@Param("x")String fonction);
26
27     @Query("SELECT count(*) FROM Agent a WHERE a.statutAgent=TRUE")
28     public long nombreAgt_Actifs();
29
```

Figure 32: Interface "AgentRepository"

II.2.4- Codage d'un contrôleur

```
AgentRepository.java AgentControlleur.java AdminController.java
1 package nema2.controller;
2
3 import java.util.List;
5
6 @Controller
7 public class AgentControlleur {
8
9     @Autowired
10     private AgentRepository agentRepository;
11
12
13     @Autowired
14     private RoleRepository roleRepository;
15
16     @Autowired
17     private AgentRoleRepository agentRoleRepository;
18
19     //Appel pour le chargement du formulaire d'ajout d'un agent
20     @Secured(value = {"ROLE_ADMIN"})
21     @RequestMapping(value = "/agentForm", method=RequestMethod.GET)
22     public String formAgent(Model model) {
23         model.addAttribute("agent", new Agent());
24         return "agentForm";
25     }
26 }
```

Figure 33: Contrôleur "AgentController"

Toute application web développée avec Spring Boot possède un contrôleur frontal par défaut « **DispatcherServlet** » qui intercepte toutes les demandes entrantes et les transfère à tous

les autres composants (contrôleurs) de l'application. Les autres contrôleurs sont codés par le développeur et sont des fichiers java avec l'annotation « **@Controller** » placée avant le nom de la classe.

D'autres annotations sont utilisées dans notre contrôleur, comme :

@Autowired, qui signifie que la dépendance d'objet est implicitement injectée.

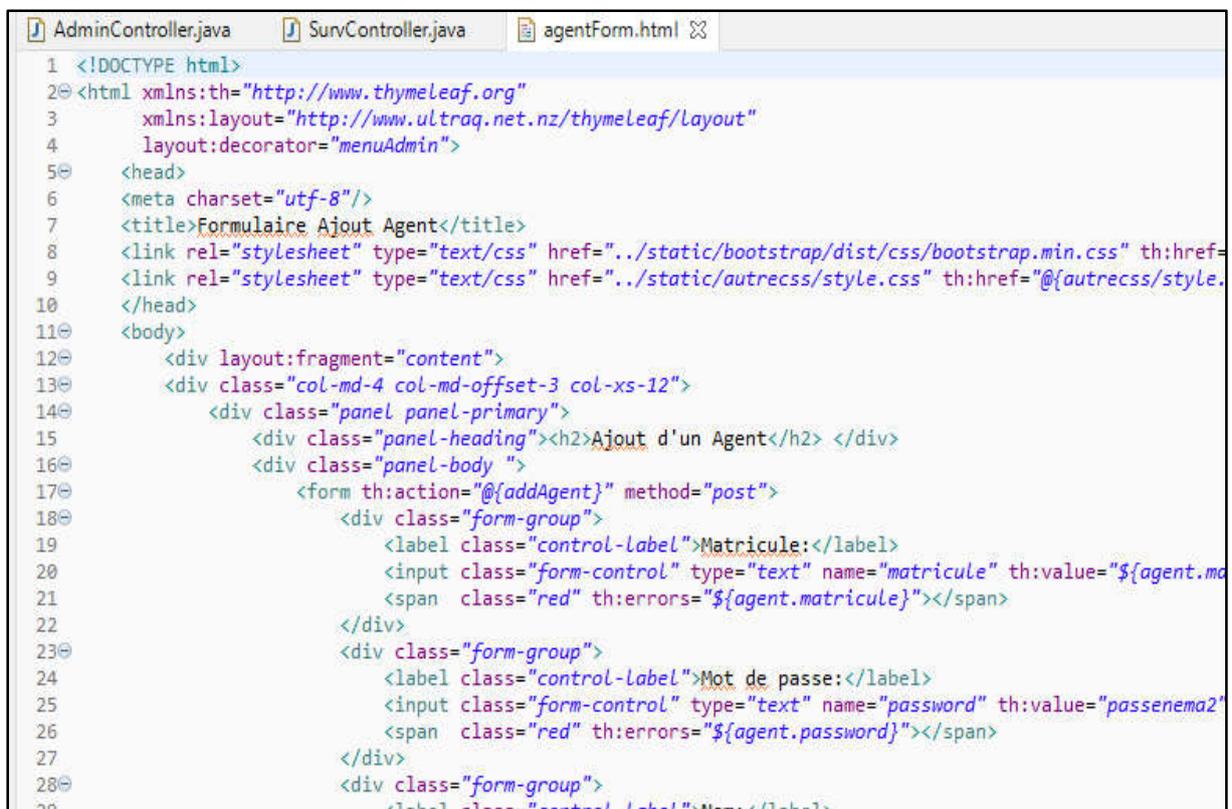
@Secured, permet de déterminer les utilisateurs qui peuvent appeler la méthode en-dessous

@RequestMapping, elle est utilisée pour mapper des requêtes Web sur des classes et des méthodes de gestionnaires spécifiques.

II.2.5- Codage d'une vue

Les vues de notre application sont dans le dossier `src/main/resources/templates`. Dans nos vues, nous avons utilisé du HTML avec l'espace de nom du moteur de Template Thymeleaf, du Bootstrap et un peu de JavaScript dans certaines.

Ci-dessous (**Figure 34**), nous vous présentons un exemple de vue « **agentForm.html** », notre formulaire d'ajout d'un agent (utilisateur).



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org"
3     xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4     layout:decorator="menuAdmin">
5 <head>
6 <meta charset="utf-8"/>
7 <title>Formulaire Ajout Agent</title>
8 <link rel="stylesheet" type="text/css" href="../static/bootstrap/dist/css/bootstrap.min.css" th:href=
9 <link rel="stylesheet" type="text/css" href="../static/autrecss/style.css" th:href="@{autrecss/style.
10 </head>
11 <body>
12 <div layout:fragment="content">
13 <div class="col-md-4 col-md-offset-3 col-xs-12">
14 <div class="panel panel-primary">
15 <div class="panel-heading"><h2>Ajout d'un Agent</h2> </div>
16 <div class="panel-body">
17 <form th:action="@{addAgent}" method="post">
18 <div class="form-group">
19 <label class="control-label">Matricule:</label>
20 <input class="form-control" type="text" name="matricule" th:value="{agent.ma
21 <span class="red" th:errors="{agent.matricule}"></span>
22 </div>
23 <div class="form-group">
24 <label class="control-label">Mot de passe:</label>
25 <input class="form-control" type="text" name="password" th:value="passenema2'
26 <span class="red" th:errors="{agent.password}"></span>
27 </div>
28 <div class="form-group">
29 <label class="control-label">Nom:</label>
```

Figure 34: Fichier HTML du formulaire d'ajout d'un agent

CHAPITRE V : PRESENTATION DE L'APPLICATION

L'interface graphique ou en anglais *user interface* (UI) permet aux personnes d'interagir avec notre application. Une interface graphique doit être conviviale pour procurer du plaisir à l'utilisateur pendant sa navigation.

I- Vue globale de l'application

La figure ci-dessous (Figure 35), donne une vue globale de notre application. Chaque profil a un tableau de bord qui lui est propre et qui donne de façon résumée des informations à l'utilisateur.

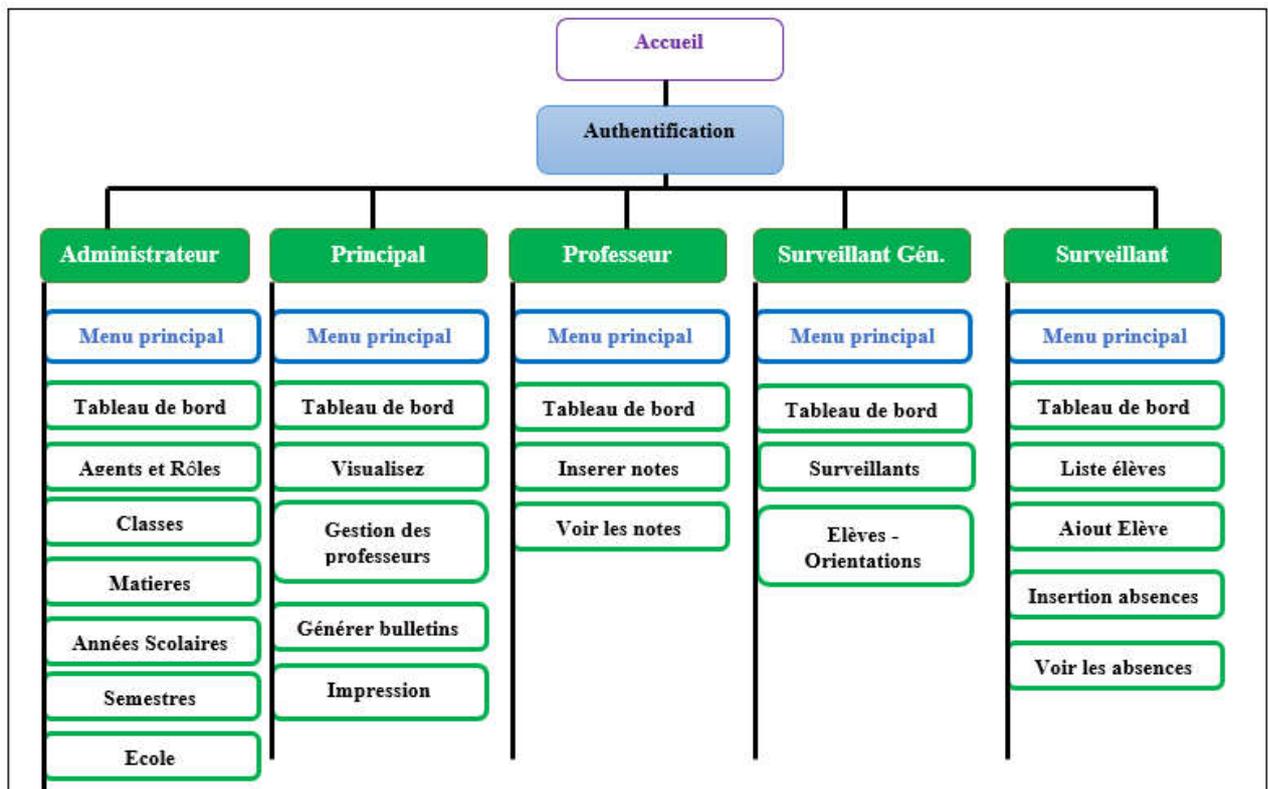


Figure 35: Vue globale de l'application

II- Page d'accueil

La page d'accueil est la page qui s'affiche au lancement de l'application. Tout le monde peut y accéder, elle possède un bouton dans sa partie droite en haut qui permet d'appeler la page d'authentification. La figure ci-dessous (Figure 36) nous montre cette interface utilisateur.



Figure 36: Page d'accueil de l'application

III- Page d'authentification

L'accès à toutes les autres interfaces, nécessite une authentification de l'utilisateur. L'utilisateur doit saisir son identifiant (son matricule) et son mot de passe. Si la saisie est correcte c'est-à-dire l'utilisateur est déjà ajouté par l'administrateur, alors il accède à l'interface avec le menu principal.

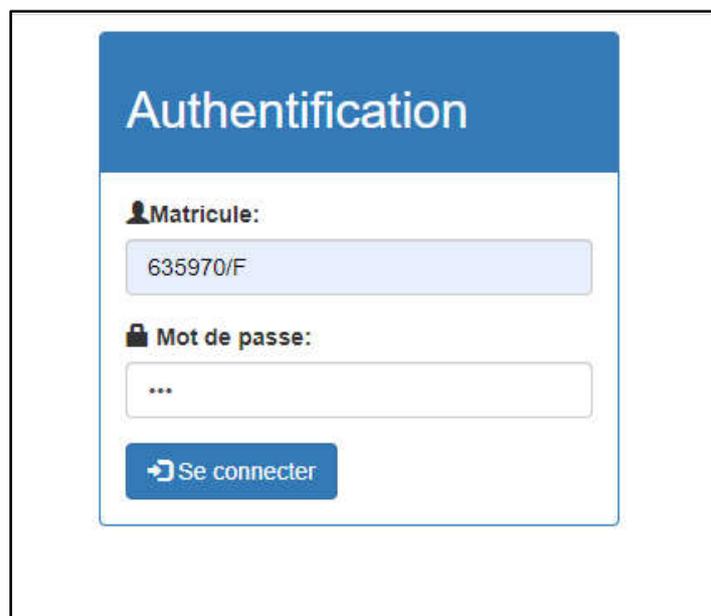


Figure 37: Page d'authentification

IV- Page avec le menu principal

Tous les utilisateurs après connexion accèdent à cette page (**Figure 38**). Suivant le profil de l'utilisateur et/ou sa fonction dans l'établissement, il clique sur l'onglet correspondant à son profil au menu pour accéder au tableau de bord de son profil avec un menu spécifique. Si un utilisateur essaie d'accéder à un profil non autorisé, il voit un message qui lui signifie qu'il n'a pas l'autorisation.

Pour pallier à l'éventualité d'un clic par inadvertance ou pas, nous avons caché tous les profils auxquels un utilisateur ne doit pas accéder à partir de son rôle.

Le bouton  permet à l'utilisateur connecté d'accéder au formulaire de modification de son mot de passe.

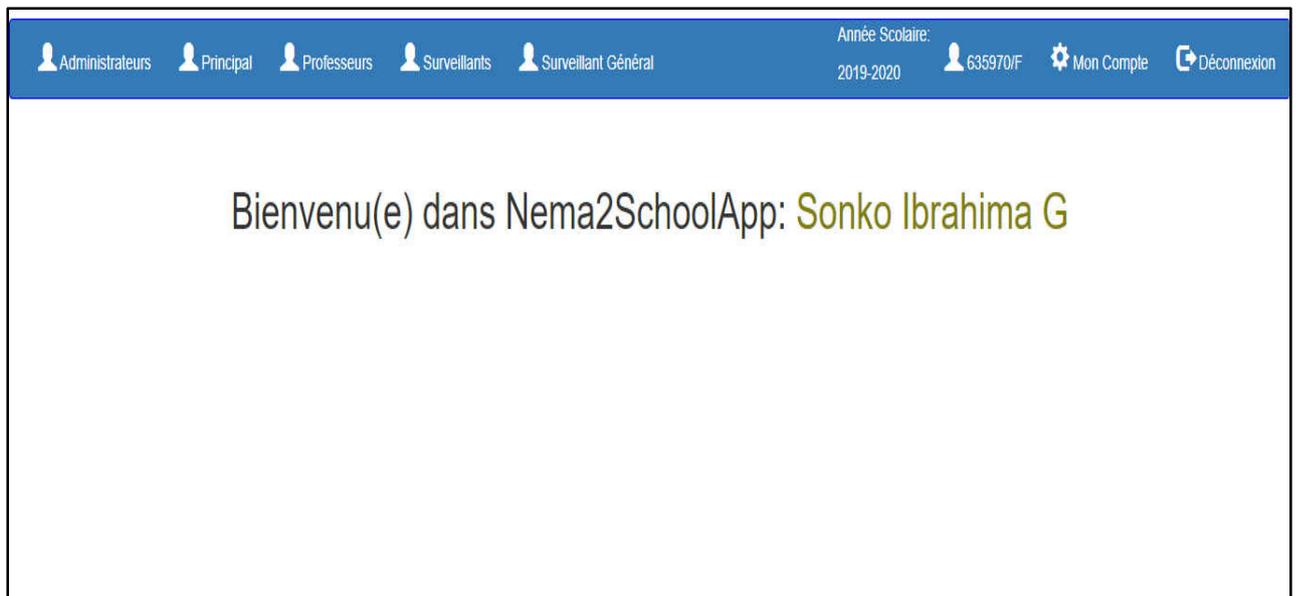


Figure 38: Page avec menu principal

V- Tableau de bord de l'administrateur

Après le choix de son profil dans la page avec la barre de menu principal, on accède au tableau de bord de son profil qui possède une barre de menu spécifique. A ce niveau l'utilisateur peut choisir l'onglet ou le sous-onglet pour soit ajouter, soit supprimer, soit lire, soit modifier des données. La figure (**Figure 39**) ci-dessous, nous montre le tableau de bord de l'administrateur.

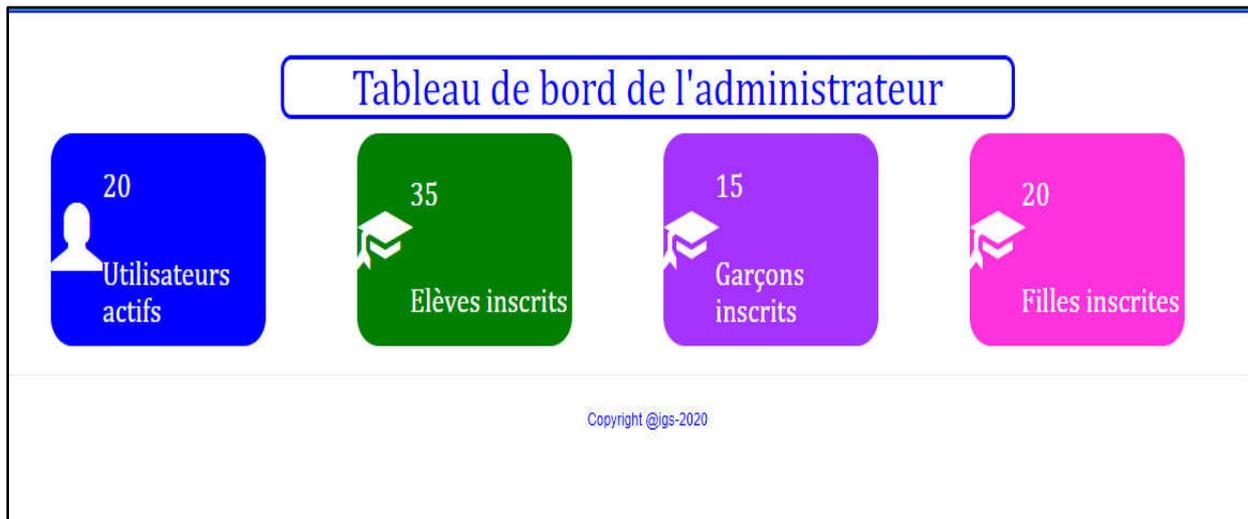
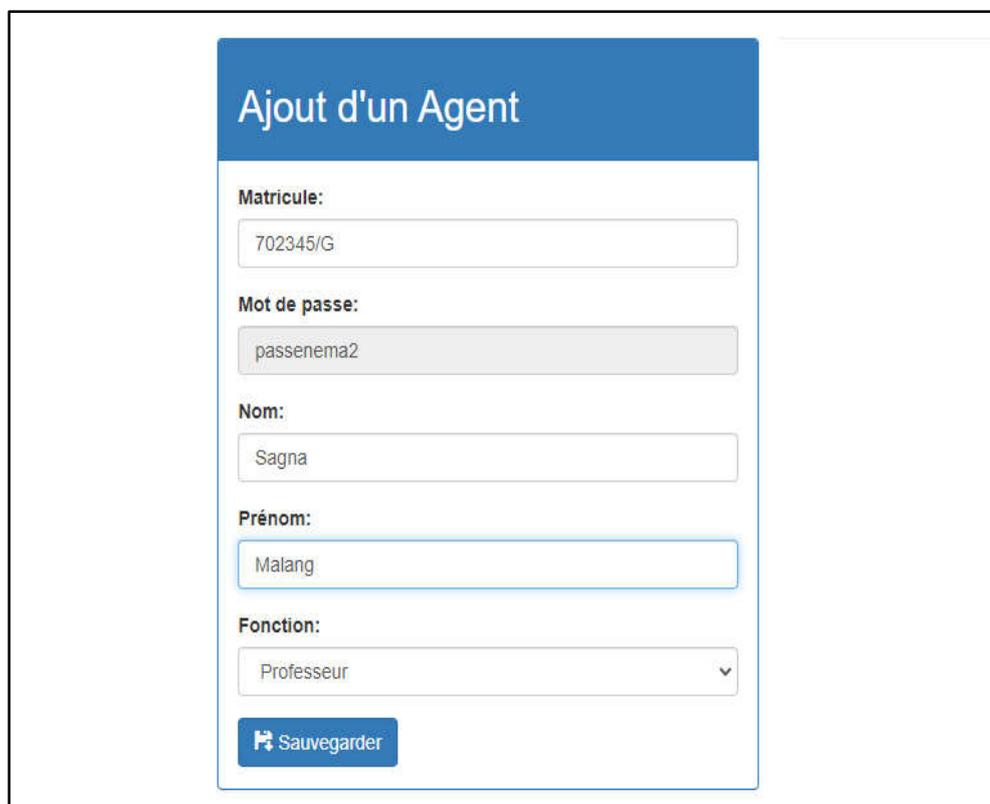


Figure 39: Tableau de bord de l'administrateur

VI- Formulaire d'ajout d'un agent (utilisateur)

Pour ajouter un agent, l'administrateur, à partir de son menu doit cliquer sur le sous-onglet « *Ajouter un agent* » de l'onglet « *Agents et Rôles* » pour que le formulaire d'ajout d'un agent s'affiche. Tous les champs sont à remplir sauf le mot de passe, qui est donné par défaut et qui doit être modifier par l'utilisateur après connexion. Voir figure (Figure 40) ci-dessous.



Ajout d'un Agent

Matricule:

Mot de passe:

Nom:

Prénom:

Fonction:

Figure 40: Interface avec le formulaire d'ajout d'un agent

VII- Liste des agents

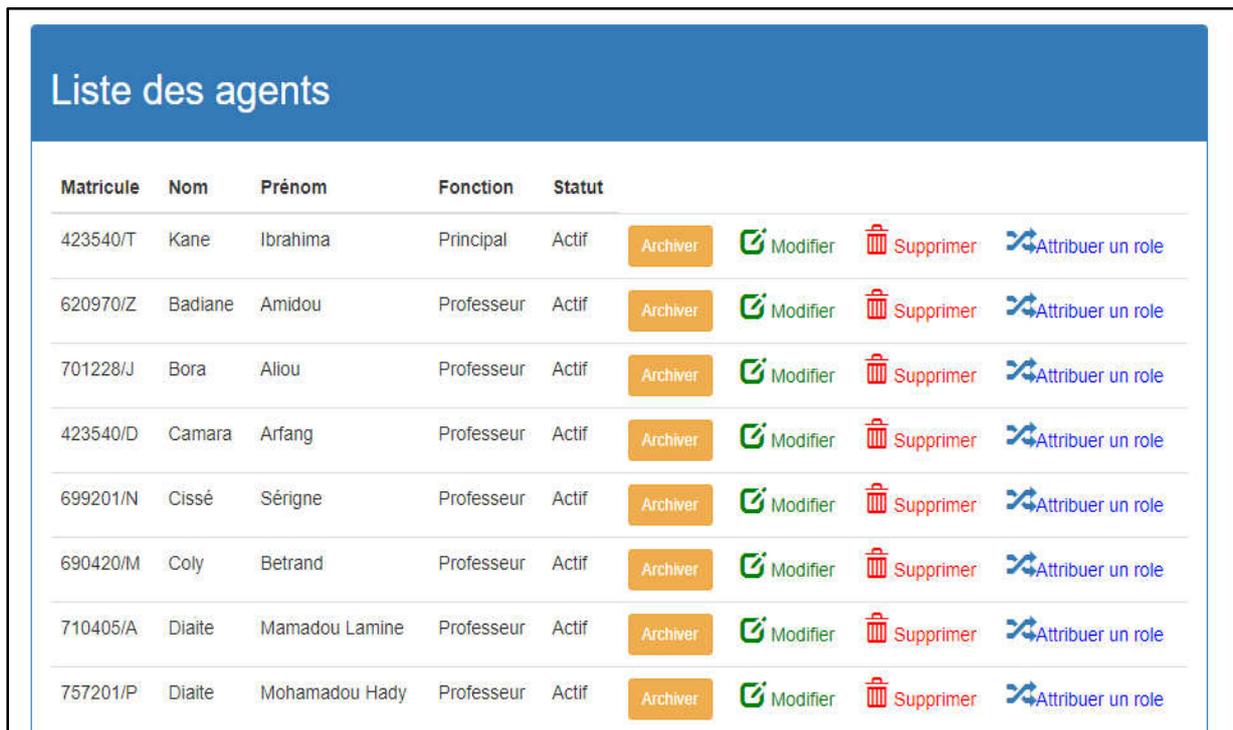
Accéder à la liste des agents, passe par le sous-onglet « *Listes des agents* » de l'onglet « *Agents et Rôles* ». En plus de voir la liste des agents sous forme de pages, l'administrateur peut à partir de cette dernière, procéder à l'archivage d'un agent en cliquant sur le bouton

Archiver

, modifier ou supprimer un agent, ouvrir le formulaire d'attribution d'un rôle à un

Attribuer un rôle

utilisateur par le bouton . Voir la figure (Figure 41) ci-dessous.



Matricule	Nom	Prénom	Fonction	Statut	Archiver	Modifier	Supprimer	Attribuer un rôle
423540/T	Kane	Ibrahima	Principal	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle
620970/Z	Badiane	Amidou	Professeur	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle
701228/J	Bora	Allou	Professeur	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle
423540/D	Camara	Arfang	Professeur	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle
699201/N	Cissé	Sérigne	Professeur	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle
690420/M	Coly	Bertrand	Professeur	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle
710405/A	Diaite	Mamadou Lamine	Professeur	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle
757201/P	Diaite	Mohamadou Hady	Professeur	Actif	Archiver	Modifier	Supprimer	Attribuer un rôle

Figure 41: Liste paginée des agents (utilisateurs)

VIII- Formulaire d'ajout d'un élève

L'ajout d'un élève dans l'application est du ressort des surveillants après avoir accédé à la page avec leur tableau de bord et leur menu. A ce niveau l'onglet « *Ajout Elève* » permet d'accéder au formulaire d'ajout d'un élève. Le code de l'élève, qui est son identifiant constitué de huit chiffres est la concaténation de l'année d'arrivée et d'un nombre à quatre chiffres auto généré. Ce dernier n'est pas visible sur le formulaire mais généré en arrière-plan. Le statut de l'élève est aussi un champ caché car paramétré par défaut à actif. La figure (Figure 42), nous montre un exemple d'ajout d'un élève.

Formulaire Ajout Elève

Nom:
Kane

Prénom:
Madeleine

Date de naissance:
12/02/2003

Lieu de naissance:
Kolda

Classes redoublées:

Sexe: M F

Année d'arrivée:
2017

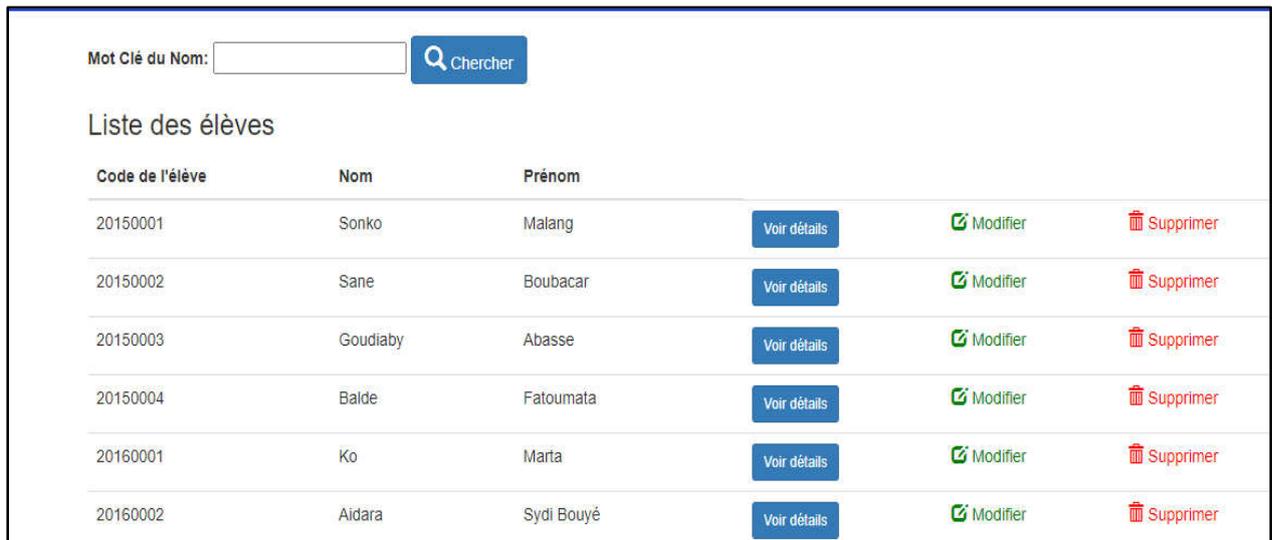
Sauvegarder

Figure 42: Formulaire d'ajout d'un élève

IX- Liste des élèves et formulaire de recherche par nom

Le surveillant, pour voir la liste de tous les élèves archivés ou non, doit à partir de son menu, choisir l'onglet « *Liste élèves* » comme indiqué à la **Figure 43**. Cette liste est paginée et le formulaire de recherche qui se trouve en haut de cette interface, permet de rechercher un élève par son nom de famille (**Figure 44**). L'interface permet aussi de voir les détails des

données d'un élève en cliquant sur le bouton **Voir détails**. Il peut aussi à partir de cette interface modifier ou supprimer un élève.



Mot Clé du Nom: Chercher

Liste des élèves

Code de l'élève	Nom	Prénom			
20150001	Sonko	Malang	Voir détails	Modifier	Supprimer
20150002	Sane	Boubacar	Voir détails	Modifier	Supprimer
20150003	Goudiaby	Abasse	Voir détails	Modifier	Supprimer
20150004	Balde	Fatoumata	Voir détails	Modifier	Supprimer
20160001	Ko	Marta	Voir détails	Modifier	Supprimer
20160002	Aidara	Sydi Bouyé	Voir détails	Modifier	Supprimer

Figure 43: Liste paginée des élèves



Mot Clé du Nom: Chercher

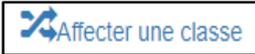
Liste des élèves

Code de l'élève	Nom	Prénom			
20150001	Sonko	Malang	Voir détails	Modifier	Supprimer
20160010	Sonko	Fatou	Voir détails	Modifier	Supprimer

0

Figure 44: Formulaire de recherche d'un élève par nom

X- Formulaire d'affectation d'une classe à un surveillant

L'affectation d'une classe à un surveillant est une action réservée uniquement au SG. Pour se faire il doit accéder à l'interface contenant son menu et son tableau de bord. A partir de ce menu il accède à la liste des surveillants en passant par le sous-onglet « *Liste des surveillants* » de l'onglet « *Surveillants* ». Dans la page contenant la liste des surveillants, devant le nom du surveillant concerné, il y'a un bouton  qui permet d'ouvrir le formulaire d'affectation d'une classe. Il s'agit seulement de sélectionner la classe puis de sauvegarder car le matricule du surveillant et l'année scolaire sont préremplis. (Figure 45)

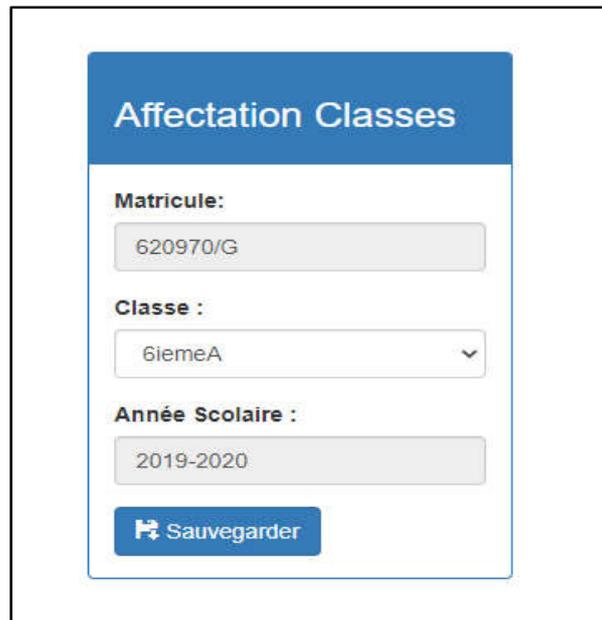
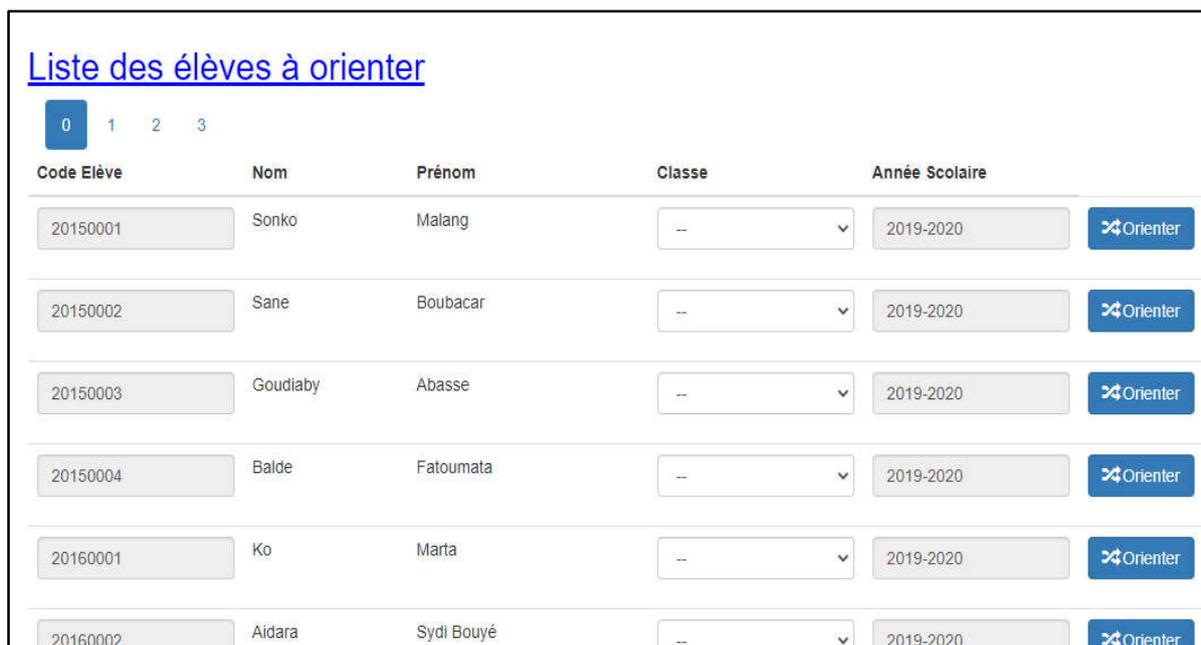


Figure 45: Formulaire d'affectation d'une classe à un surveillant

XI- Liste des élèves à orienter

L'orientation d'un élève dans une classe pour une année scolaire donnée concerne seulement les élèves nouvellement venus et est à la charge du SG après leur ajout par les surveillants. Pour accéder à l'interface de la **Figure 46** ci-dessous, le SG utilise le sous-onglet « *Nouvelle orientation* » de l'onglet « *Elèves-Orientations* ». Il lui suffit de sélectionner la classe dans laquelle l'élève doit être orienté puis cliquer sur le bouton .

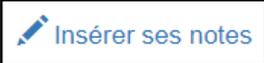


Code Elève	Nom	Prénom	Classe	Année Scolaire	
20150001	Sonko	Malang	--	2019-2020	
20150002	Sane	Boubacar	--	2019-2020	
20150003	Goudiaby	Abasse	--	2019-2020	
20150004	Balde	Fatoumata	--	2019-2020	
20160001	Ko	Marta	--	2019-2020	
20160002	Aidara	Sydi Bouyé	--	2019-2020	

Figure 46: Liste paginée des élèves à orienter et formulaire d'orientation

XII- Formulaire d'insertion de notes

L'insertion d'une note par un professeur nécessite plusieurs étapes. Ce dernier doit après accession à son tableau de bord à partir de l'interface commune à tous les utilisateurs, utiliser l'onglet « *Insérer notes* ». Il accède à l'interface permettant de charger la liste des élèves d'une classe. Le choix de la classe se fait par une liste déroulante, cette dernière ne contient que les classes qui sont affectées à ce professeur. A partir de cette liste (exemple 3^{ème} A, **Figure 47**),

le professeur clique sur le bouton  devant le nom de l'élève pour ouvrir le formulaire d'insertion de notes (**Figure 48**). Le professeur doit insérer toutes les notes du semestre avant une date butoir après laquelle, le Principal doit changer de semestre.

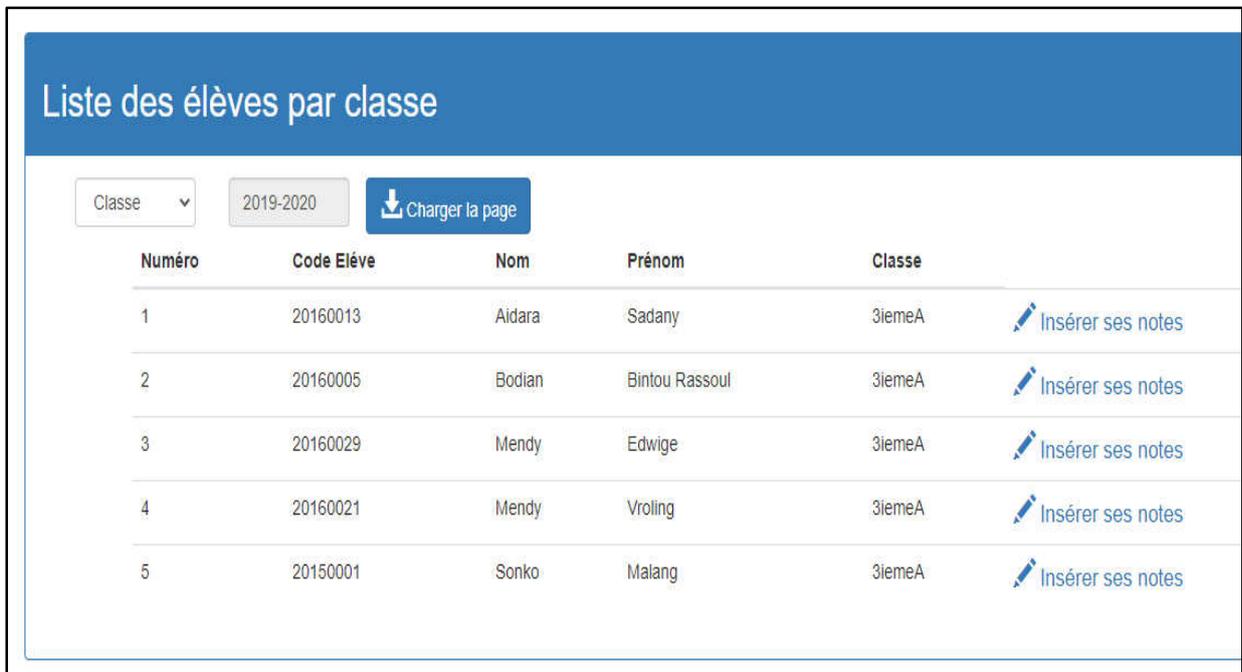


Figure 47: Liste des élèves par classe pour insertion de notes. L'interface affiche un tableau de données avec les colonnes Numéro, Code Elève, Nom, Prénom, Classe et un bouton 'Insérer ses notes' pour chaque ligne. Le tableau est précédé d'un menu déroulant 'Classe', un champ de sélection '2019-2020' et un bouton 'Charger la page'.

Numéro	Code Elève	Nom	Prénom	Classe	
1	20160013	Aidara	Sadany	3iemeA	
2	20160005	Bodian	Bintou Rassoul	3iemeA	
3	20160029	Mendy	Edwige	3iemeA	
4	20160021	Mendy	Vrolling	3iemeA	
5	20150001	Sonko	Malang	3iemeA	

Figure 47: Liste élèves par classe pour insertion de notes

Voir les notes 635

Inserer les notes de l'élève

Code élève: 20160005 Prénom: Bintou Rassoul Nom: Bodian

Classe: 3iemeA Année scolaire: 2019-2020 Semestre: semestre1

Matière: Mathématiques Devoir1: 0.0 Devoir2: 0.0

Devoir3: 0.0 Composition: 0.0 [Sauvegarder](#)

Figure 48: Formulaire d'insertion de notes

XIII- Modification de notes

Après insertion des notes, le professeur peut voir l'ensemble des notes des élèves d'une classe dans une matière et la moyenne de chacun. Pour ce faire, il doit choisir l'onglet « **Voir les notes** » puis sélectionner la classe et la matière et en fin cliquer sur le bouton 

. Dans cette liste (Figure 49), devant chaque nom, il y'a un bouton  pour accéder au formulaire de modification des notes insérées (Figure 50).

Liste des notes par Classe-AnnéeScolaire-Semestre

Classe: 3iemeA Matière: Mathématiques semestre1 2019-2020 [Charger la page](#)

Numéro	Prénom	Nom	Classe	Matière	Devoir1	Devoir2	Devoir3	Composition	Moyenne	
1	Sadany	Aidara	3iemeA	Mathématiques	12.0	15.0	11.0	15.0	13.25	Modifier
2	Bintou Rassoul	Bodian	3iemeA	Mathématiques	9.0	11.0	7.0	11.0	9.5	Modifier
3	Edwige	Mendy	3iemeA	Mathématiques	4.0	5.0	3.0	10.0	5.5	Modifier
4	Vroiling	Mendy	3iemeA	Mathématiques	17.0	15.0	12.0	19.0	15.75	Modifier
5	Malang	Sonko	3iemeA	Mathématiques	13.0	5.0	11.0	16.0	11.25	Modifier

Figure 49: Liste des élèves d'une classe pour modification

Code élève:	Prénom:	Nom:
20160013	Sadany	Aidara
Classe:	Année scolaire:	Semestre:
3iemeA	2019-2020	semestre1
Matière:	Devoir1:	Devoir2:
Mathématiques	12.0	15.0
Devoir3:	Composition:	
11.0	15.0	

[Sauvegarder](#)

Figure 50: Formulaire de modification

XIV- Affectation d'une classe et d'une matière

Comme pour les surveillants, aux professeurs aussi et suivant le dispatching fait pour les emplois du temps, le Principal affecte des classes. La spécificité de cette affectation est l'association avec la matière que le professeur va enseigner dans cette classe. Pour atteindre le formulaire d'affectation d'une classe et d'une matière, le Principal, à partir de l'interface de son tableau de bord utilise le sous-onglet « *Liste des professeurs* » de l'onglet « *Gestion Professeurs* » pour afficher d'abord la liste des professeurs, ensuite il utilise le bouton

 Affecter une classe-matière

, devant le nom d'un professeur, pour ouvrir le formulaire d'affectation du couple classe-matière (Figure 51).



Figure 51: Formulaire d'affectation classe-matière à un professeur

XV- Impression de listes

L'impression des listes d'appel par classe est à la charge du Principal. Il doit accéder à l'interface de la **Figure 52**, en passant par le sous-onglet « **Impression de listes** » de l'onglet « **Impression** ». Dans cette interface il y a deux formulaires, le premier à partir de notre gauche permet d'imprimer la liste d'appel d'une classe et le deuxième la liste d'appel détaillée d'une classe. Il suffit de sélectionner d'abord la classe puis de cliquer sur le bouton . Le type de sortie dépend de l'imprimante par défaut de notre machine. Dans notre cas, c'est l'imprimante « **Foxit Reader PDF Printer** ».

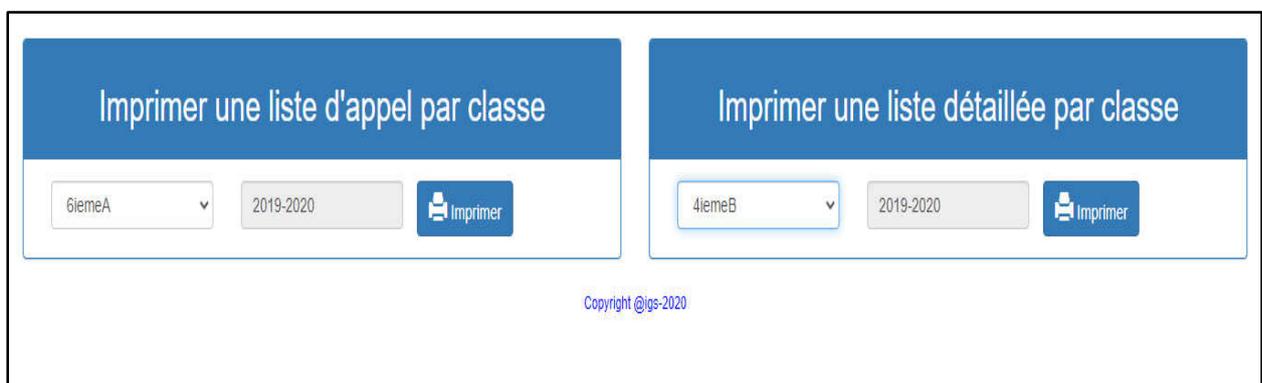


Figure 52: Interface avec les formulaires d'impression de listes par classe

La **Figure 53**, montre un exemple de liste d'élèves d'une classe. Cette liste est une liste détaillée, un exemple de fiche de notes et/ou liste d'appel est donnée en annexe (**Annexe 4**).

The screenshot shows a web browser window with several tabs. The active page displays the following information:

- IA:** ZIGUINCHOR
- IEF:** ZIGUINCHOR
- CEM Néma 2**
- Téléphone:** 339925065
- Email:** cemnema2@gmail.com
- Année Scolaire:** 2019-2020
- Classe:** 5iemeB
- Total élèves:** 4
- Total Filles:** 3
- Total Garçons:** 1

The main title is **Liste des élèves**.

N°	Code	Prénom	Nom	Date Naiss.	Lieu Naiss.	Redouble	Sexe
1	20160002	Sydi Bouyé	Aidara	21-12-2000	Daroul Salam		M
2	20160026	Diarietou	Diamanka	23-01-2004	Goudomp		F
3	20160018	Sire	Mandiang	01-01-2004	Sedhiou	4ièmeC	F
4	20160010	Fatou	Sonko	28-07-2005	Ziguinchor	5ièmeC	F

Figure 53: Exemple de liste détaillée d'une classe générée

XVI- Impression d'un bulletin de notes

L'impression des bulletins de notes comme celle des listes d'appel est du domaine exclusif du Principal. A la fin de la période d'insertion des notes et des absences du semestre, il génère tous les bulletins puis les imprime afin de les remettre aux ayants droit. Pour imprimer un bulletin de notes, il passe par le sous-onglet « *Imprimer un bulletin* » de l'onglet « *Impressions* ». Avant d'accéder à l'interface de la **Figure 54**, il y a une interface au niveau de laquelle, il choisit entre bulletins niveau troisième ou quatrième et niveau cinquième ou sixième. Car le nombre de matières dans les deux premiers niveaux (dix matières) est différent de celui des deux derniers (huit matières).

Wikipédia x Récapitulatif impression bulletin x liste formulaire - Recherche Google x +

43?idOrienta=21

Cours DGB Bootstrap 3 Tutorial Banque de l'Habitat... ZTE Livebox Connexion - CAS ...

ton Professeurs Generer bulletins Impressions

Récapitulatif de l'élève

Vous allez imprimer le bulletin du 1er semestre de l'élève suivant:

Code de l'élève:

Prénom: Sadany

Nom: Aidara

Classe:

Pour l'année scolaire:

 Imprimer

Figure 54: Interface d'impression d'un bulletin de notes

.. Realisation-dune-appli... JasperReports - bulletin... X

IA: ZIGUINCHOR **Année scolaire:** 2019-2020
IEP: ZIGUINCHOR **semestre:** 1
CEM Néma 2
Téléphone: 339925065
Email: cemnema2@gmail.com

BULLETIN DE NOTES

Prénom: Sadany **Nom:** Aldara
Né(e) le: 23-04-2003 à Daroul Khairou **Classe:** 3iemeA
Code: 20160013 **Nombre élèves:** 5 **Redoublement:**

Matières	Dvs	Compo	Moy/20	Coef	Moy x	TH	Rang	Appréciations
Composition française	11,67	15	13,33	2	26,67	TH	2	A Bien
Orthographe-Grammaire	14	13	13,5	2	27	TH	2	A Bien
Anglais	16,17	11	13,58	2	27,17	TH	2	A Bien
Histoire-Géographie	16,83	17	16,92	2	33,83	TH	1	Très bon travail
Education Civique	17,33	20	18,67	1	18,67	TH	1	Excellent travail
Mathématiques	12,67	15	13,83	3	41,5	TH	2	A Bien
BVT	16,33	16	16,17	2	32,33	TH	1	Très bon travail
Physique-Chimie	12,83	16	14,42	2	28,83	TH	2	Bon travail
Espagnol	15,5	13,5	14,5	2	29	TH	1	Bon travail
EPS	17	9	13	2	26	TH	4	A Bien
TOTAL				20	291			
Moyenne	14,55	Rang	1	Absence	1			

Satisfaisant doit continuer	X
Peux mieux faire	
Insuffisant	
Risque de redoubler	
Risque l'exclusion	

Félicitations	X
Encouragement	X
Tableau d'honneur	X
Avertissement	
Blâme	

Le Principal

Observations du conseil des professeurs

Aucune

Figure 55: Exemple de bulletin généré

CONCLUSION ET PERSPECTIVES

Le travail dont les paragraphes ci-dessous marquent la fin, était de concevoir une application web pour la gestion des notes pour le CEM Néma 2 de Ziguinchor, qui doit améliorer considérablement les manquements de celle qui y est utilisée.

Notre application web nommée « *Nema2SchoolApp* » développée à cet effet, nous permet d'atteindre l'objectif principal et les objectifs spécifiques de notre projet. Elle est responsive car offrant une expérience de lecture et de navigation optimale pour l'utilisateur quelle que soit sa gamme d'appareil. Elle permet de gérer la sécurité avec l'authentification, les années scolaires, les semestres, les classes pédagogiques, les matières, les élèves, etc. La génération de documents (listes d'appel, fiches de notes et bulletins de notes) au format papier est aussi assurée par l'application.

Afin de faciliter la compréhension du travail fait dans le cadre de ce projet et sa présentation, nous avons divisé ce document en cinq (5) grands chapitres. Dans le premier, nous avons fait une présentation du contexte justificatif du sujet, les solutions proposées faces aux difficultés rencontrées pendant le processus de gestion des notes au sein du CEM Néma 2 avec Best Grade et le processus de développement avec une brève présentation des méthodologies agiles et l'adaptation de la méthodologie Scrum à notre projet. La spécification et l'analyse des besoins fonctionnels constituent le deuxième chapitre de notre projet. Nous y avons fait une description des besoins fonctionnels et leur modélisation par les digrammes de cas d'utilisation ainsi que leur analyse par une description des cas et une modélisation de certaines fonctionnalités par des diagrammes d'activités. Le troisième chapitre a été consacré à la conception du système en présentant la conception générale puis la conception détaillée. Ensuite nous avons présenté l'implémentation de notre application en revenant sur les outils et technologies utilisés et sur l'implémentation de la base de données. Notre mémoire est consacré dans son dernier chapitre à la présentation de l'application avec des images de certaines vues.

Malgré les améliorations que notre application va apporter au système de gestion des notes des élèves, des perspectives existent, comme :

- ✓ **La gestion des inscriptions et de la comptabilité.** Il s'agit d'ajouter de nouvelles fonctionnalités qui vont prendre en charge le paiement des frais d'inscription, la gestion de la comptabilité (comptabilité financière et comptabilité matière) de l'établissement

et l'impression des documents comptables (reçus de caisses, bons de sortie, bons d'entrée, etc.).

- ✓ **L'ajout de la gestion des imputations budgétaires.** Les agents, pour la plupart sont déjà pris en compte dans la gestion des notes, alors il serait moins fastidieux comme travail de greffer sur notre application, l'automatisation de la gestion des imputations budgétaires. Aujourd'hui, elle est à la charge des établissements et l'agent demandeur doit remplir à la main et en double les renseignements nécessaires.
- ✓ **La génération des dossiers scolaires pour les livrets de baccalauréat.** A la fin de chaque année scolaire les professeurs principaux de classe et les surveillants, doivent remplir les livrets de baccalauréat des élèves et les signer. En permettant à l'application de produire une page à l'image des pages du livret, alors cela réduira considérablement le travail des surveillants et des professeurs principaux.
- ✓ **La gestion des demandes d'autorisation d'absence des agents.** La demande d'autorisation d'absence comme la demande d'une imputation budgétaire se fait toujours à la main au CEM Néma 2. Alors il s'agit de permettre à l'application de produire de façon automatisée les demandes d'autorisation d'absence.

BIBLIOGRAPHIE

[1] : M. Henry Diallo « conception et développement d'une application pour la gestion de la répartition des enseignements à l'UASZ ». Mémoire de Master 2 soutenu le 07/11/2018

WEBOGRAPHIE

[2] : Mérouze, Nicolas. "Architecture des applications web : quelles bonnes pratiques en 2020 ?". Synbioz. Consulté le 15 octobre 2020 sur <https://www.synbioz.com/blog/architecture-applications-web> du 25 juin 2019.

[3] : Conan, Denis et al. "Introduction au langage de modélisation UML". Telecom SudParis. Consulté le 15 octobre 2020 sur <http://www-inf.it-sudparis.eu/COURS/CSC4002/EnLigne/Cours/CoursUML/index.html#> d'octobre 2015.

[4] : Lothon, Florent. "Introduction aux méthodes agiles et Scrum". L'Agiliste. Consulté le 12 juillet 2020 sur <https://agiliste.fr/introduction-methodes-agiles/> du 03 juin 2013.

[5] : Pérochon, Laurent et al. "StarUML : Modélisation UML (similaire à 'Rational Rose' ou 'Borland Together')". PLUME. Consulté le 25 octobre 2020 sur <https://www.projet-plume.org/fiche/staruml> dont la dernière mise à jour date du 19 avril 2013.

[6] : Gribaumont, Chantal. "Administrez vos bases de données avec MySQL". OPENCLASSROOMS. Consulté le 25 octobre 2020 sur <https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql> mise à jour le 15 mai 2020.

[7] : Philibert, Benoit et al. "Créez des sites web responsive avec Bootstrap 4". OPENCLASSROOMS. Consulté le 26 octobre 2020 sur <https://openclassrooms.com/fr/courses/6391096-creez-des-sites-web-responsive-avec-bootstrap-4> mise à jour le 15 décembre 2019.

[8] : Steinmann, Alain et al. "Qu'est-ce que Java ?". CCM Benchmark Group. Consulté le 27 octobre 2020 sur <https://journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203555-java-definition/> mise à jour le 08 janvier 2019.

[9] : Steinmann, Alain et al. "Qu'est-ce que Java ?". CCM Benchmark Group. Consulté le 27 octobre 2020 sur <https://journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/> mise à jour le 10 mars 2020.

[10] : Fernandez, Daniel et al. "Thymeleaf". Consulté le 28 octobre 2020 sur <https://www.thymeleaf.org/index.html> mise à jour le 29 octobre 2018.

[11] : Turk, Mladen et al. "Apache Tomcat". The APACHE Software Foundation. Consulté le 28 octobre 2020 sur <http://tomcat.apache.org/> mise à jour le 29 octobre 2018.

ANNEXE

Annexe 1 : Interface administrateur pour la modification des informations de l'école, de l'année scolaire et pour l'ajout des classes.

L'interface « *Classes et Admin* » permet d'administrer Best grade, en ajoutant des informations propres à l'établissement qui l'utilise ou en paramétrant la méthode de calcul de la moyenne générale. Il s'agit :

- ✓ Du nom de l'établissement ainsi que de l'inspection d'académie et l'inspection de l'éducation et de la formation (ex inspection départementale de l'éducation nationale) ;
- ✓ De l'année scolaire ;
- ✓ D'ajouter un logo, s'il existe ;
- ✓ De saisir les noms des classes pédagogiques.

Ces informations vont être prise en compte dans les bulletins de notes, les fiches de notes et les listes d'appel, etc.

Classes et Admin

Menu

IA	IA THIES
IDEN	IDEN de Tivaouane
Établis.	CEM Ababacar Sy

Année Scolaire: Année Scolaire: 2007 - 2008

Ajouter les Classes

Sélectionner la Formule Pour la Moyenne Annuelle

$(MoySem1 + MoySem2 \times 2) / 3$

$(MoySem1 + MoySem2) / 2$

Ajouter Une Photo Ou Logo Pour l'Établissement

Sélectionner

Figure 56: Interface d'administration de Best Grade

Annexe 2 : Exemple de page de professeur

Les pages des professeurs sont générées automatiquement après insertion des noms de tous les professeurs. Le fichier qui permet cette génération est externe au fichier principal, mais l'utilise pour charger les classes et les élèves. L'administrateur doit sélectionner les classes et saisir les matières pour toutes les pages des professeurs avant de les enregistrer avec un mot de passe.

The screenshot shows a web interface for a teacher. At the top, there is a text input field containing 'El H Dieme' and a button labeled 'Enregistrer et Fermer'. Below this, there is a table with two columns: 'Classe' and 'Discipline'. Each row in the table has a dropdown menu for the class, a text input field for the discipline, and a blue link labeled 'Ajouter les notes'. The rows are as follows:

Classe	Discipline	Action
3eC	Compo-Franç	Ajouter les notes
3eC	Ortho-Gram	Ajouter les notes
3eB	HG	Ajouter les notes
3eB	EC	Ajouter les notes
5eA	HG	Ajouter les notes
5eA	EC	Ajouter les notes
z	-	Ajouter les notes
z	-	Ajouter les notes

Figure 57: Page d'un professeur dans Best Grade

Annexe 3 : Exemple de page d'insertion de notes d'une matière

Pour saisir les notes des élèves d'une classe dans une matière, le professeur pour accéder à l'interface (Figure 58) à partir de l'interface précédente (Figure 57), doit cliquer sur le lien « *Ajouter les notes* » qui est sur la ligne du nom de la classe et de la matière. Après avoir saisi les notes des élèves de la classe, il doit revenir à l'interface précédente par le bouton « *Retour au menu* » pour soit choisir une autre classe, soit enregistrer et fermer par le bouton « *Enregistrer et Fermer* » de l'interface de la Figure 57.

Retour au Menu 

Nom **TH Debut**
Discipline **TH auto**
Classe

Prénoms	NOM	D1	D2	D3	D4	D5	N Comp	TH	Rang	Moyenne
ALPHA OMAR	BALDE						-	X	X	
MARIAMA	BANGOURA						-	X	X	
AMADOU LAMIN	BARRY						-	X	X	
SOULEYMANE	BARRY						-	X	X	
JOELLE	BAZAN						-	X	X	
ROBERT	BLECK						-	X	X	
MAME SIRE	BODIAN						-	X	X	
JACQUES	BOUCAL						-	X	X	
BABA SADIO	CISSE						-	X	X	
NDEYE MAGUET	CISSE						-	X	X	
ANASTASIA PE	CORREA						-	X	X	
MARCEL	CORREA						-	X	X	
AMINATA	DIALLO						-	X	X	
ISMAILA	DIATTA						-	X	X	

Figure 58: Page d'insertion de notes dans Best Grade

Annexe 4 : Exemple de liste d'appel et/ou Fiche de notes générée

Mémoire de fin de cyc... Mémoire de fin de cyc... Mémoire_Master_Hen... liste eleves 3iemeA.pdf x

IA:ZIGUINCHOR **Année Scolaire:2019-2020**
IEF:ZIGUINCHOR **Classe: 3iemeA**
CEM NEMA 2 **Semestre:**
Matière:

Feuille de Notes

N°	Prénom	Nom	DV1	DV2	DV3	Comp
1	Sadany	Aidara				
2	Bintou Rassoul	Bodian				
3	Edwige	Mendy				
4	Vroling	Mendy				
5	Malang	Sonko				

Figure 59: Fiche de notes et/ou liste d'appel

Annexe 5 : Exemple de bulletin du 2nd semestre généré

IA: ZIGUINCHOR		Année scolaire: 2019-2020
IEF: ZIGUINCHOR	BULLETIN DE NOTES	semestre2
CEM Néma 2		
Téléphone: 339925065		
Email: cemnema2@gmail.com		

Prénom: Sadany		Nom: Aidara
Né(e) le: 23-04-2003	à Daroul Khairou	Classe: 3iemeA
Code: 20160013	Nombre élèves: 5	Redoublement:

Matières	Dvs	Compo	Moy/20	Coef	Moy x	TH	Rang	Appréciations
Composition française	13,67	10	11,83	2	23,67		2	Passable
Orthographe-Grammaire	16,33	11	13,67	2	27,33	TH	1	A Bien
Anglais	16,33	18	17,17	2	34,33	TH	1	Très bon travail
Histoire-Géographie	10,33	14	12,17	2	24,33	TH	1	A Bien
Education Civique	13	10,5	11,75	1	11,75		1	Passable
Mathématiques	15,5	11	13,25	3	39,75	TH	1	A Bien
SVT	6	12	9	2	18		2	Insuffisant
Physique-Chimie	17,17	11	14,08	2	28,17	TH	1	Bon travail
Espagnol	13,33	10,5	11,92	2	23,83		1	Passable
EPS	13	17,5	15,25	2	30,5	TH	4	Bon travail
TOTAL				20	261,67			Absence 0
Moyenne	13,08	Rang	1					Abs. Total 1
		Tableau d'honneur		Encouragement				

Décision du conseil	
Admis(e) en classe supérieure	X
Autorisé(e) à redoubler	
Exclusion	

Observations du conseil des professeurs

Aucune

Moy. 1er Semestre	14,55
Moy. 2nd Semestre	13,08
Moy. annuelle	13,82
Rang annuel	1

Le Principal

Figure 60: Bulletin du 2nd semestre