

UNIVERSITE SIDI MOHAMED BEN ABDELLAH

FACULTÉ DES SCIENCES ET TECHNIQUE FES

DÉPARTEMENT D'INFORMATIQUE



PROJET DE FIN D'ÉTUDES

MASTER SCIENCES ET TECHNIQUES

SYSTÈMES INTELLIGENTS & RÉSEAUX

Conception et réalisation d'une application web de gestion d'école



LIEU DE STAGE : ABNSOFT Casablanca

Réalisé par : KABLY HAMID

Soutenu le : 15 / 06 / 2017

ENCADRÉ PAR :

MR Med Chaouki Abounaima

MR Badr Bennani Naciri

DEVANT LE JURY COMPOSÉ DE :

MR Rachid Ben Abbou

MR Med Chaouki Abounaima

MR Abderrahim Benabbou

MR Mohamed Ouzarf

ANNÉE UNIVERSITAIRE: 2016 - 2017

Dédicace

Je dédie ce projet :

A ma chère mère,

A mon cher père,

*Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir
et de m'épauler pour que je puisse atteindre mes objectifs.*

A mes frères,

A mes chers oncles,

Pour leur soutien et leurs conseils précieux tout au long de mes études.

A mes chers amis Amine, Hamza, Abdelmoughit, Abdelwahed,

Pour leur aide et support dans les moments difficiles.

A toute ma famille,

A tous mes autres ami(e)s,

A tous ceux que j'aime et ceux qui m'aiment.

Remerciements

*Louanges tout d'abord à **ALLAH** qui est l'origine de toute réussite dans notre vie.*

*J'adresse mes sincères remerciements à Monsieur **Badr Naciri Bennani** qui m'a confié ce sujet et qui a assumé l'encadrement de mon projet, l'intérêt qu'il a porté à mon travail, sa bienveillance, sa rigueur scientifique, ses discussions fructueuses, ses hautes qualités humaines, ont constitué une aide précieuse et m'ont permis de mener à terme ce travail.*

*Par la même occasion, j'adresse mes remerciements à **Mr. Med Chaouki Abounaïma** pour ses efforts et sa bonne coordination durant les quatre mois de mon stage.*

Je remercie également tous mes enseignants pour leurs efforts qui m'ont guidé et qui ont enrichi mes travaux tout au long de mes études universitaires.

Enfin, mes remerciements s'adressent aussi à tous ceux qui ont participé, de près ou de loin, à l'élaboration de ce projet de fin d'études et en particulier à ma famille et mes amis.

Résumé

Dans le cadre de mon Projet de Fin d'Etudes, la société ABNSOFT souhaite mettre en place une application de gestion d'école pour ses clients, qui aura besoin de gérer l'ensemble de ses élèves, professeurs, parents... à travers une interface web simple et pratique.

Pour ce projet on a opté comme démarche, les étapes suivantes :

- Recenser les besoins fonctionnels et non fonctionnels du projet.
- L'étude technique et la conception détaillée de l'application.
- Réalisation.

Pour bien mener le développement de ce projet, la méthodologie agile qui part du principe que spécifier et planifier dans les détails l'intégralité d'un produit avant de le développer, semble plus adéquate à notre contexte, et plus précisément la méthode **SCRUM**, avec Le langage de modélisation **UML**.

Pour ce qui concerne le volet technique, il y avait eu recours aux nouvelles technologies et aux bonnes pratiques de développement, depuis Maven qui permet de créer une arborescence standard du code et de ses ressources, aux Framework & technologies : Hibernate/JPA, Spring, Spring security, JavaEE, JQuery...etc

Ma mission consiste à développer un ensemble de modules de gestion d'école, tels que la gestion des : élèves, professeurs, parents, cantine, transport, activités.... En utilisant les différentes techniques et outils de développement mentionné auparavant.

Mots clés: ABNSOFT, SCRUM, GESTION D'ÉCOLE, UML, SPRING, HIBERNATE, MAVEN, TWILIO, TFS.

Abstract

As part of my End-of-Study Project, ABNSOFT wants to set up a school management application for its clients, which will need to manage all its students, teachers, parents ... through a Simple and convenient web interface.

For this project, the following steps were taken as a project approach:

- Identify the functional and non-functional needs of the project.
- Technical study and detailed design of the application.
- Production.

In order to carry out the development of this project, the agile methodology, which assumes that specifying and planning in detail the entirety of a product before developing it, seems more appropriate to our context, and more precisely the **SCRUM** method, with The **UML** modeling language.

As for the technical aspect, new technologies and good development practices have been used since Maven, which allows to create a standard tree of the code and its resources, to Framework & technologies: Hibernate / JPA, Spring, Spring security, JavaEE, JQuery ... etc.

My mission is to develop a set of school management modules, such as the management of: students, teachers, parents, canteen, transport, activities.... Using the different techniques and development tools mentioned above.

Keywords: ABNSOFT, SCRUM, MANAGEMENT SCHOOL, UML, SPRING, HIBERNATE, MAVEN, TWILIO, TFS.

Liste des figures

FIGURE 1 : LES MODULES DE GESTION COUVERTS PAR SYNERGIE.....	17
FIGURE 2 : ORGANIGRAMME D'ABNSOFT	20
FIGURE 3 : DEMARCHE DU PROJET	32
FIGURE 4 : PLANNING DU PROJET	37
FIGURE 5 : DIAGRAMME DE GANTT.....	38
FIGURE 6 : EXEMPLE DE REPRESENTATION D'UN ACTEUR	43
FIGURE 7 : EXEMPLE DE REPRESENTATION D'UN CAS D'UTILISATION.....	43
FIGURE 8 : DIAGRAMME DE CAS D'UTILISATION D'ACTEUR «ÉLEVE»	46
FIGURE 9 : DIAGRAMME DE CAS D'UTILISATION D'ACTEUR «PROFESSEUR»	46
FIGURE 10 : DIAGRAMME DE CAS D'UTILISATION D'ACTEUR «ADMINISTRATION»	47
FIGURE 11 : DIAGRAMME DE CAS D'UTILISATION «HERITAGE »	48
FIGURE 12 : DIAGRAMME DE SEQUENCE AUTHENTIFICATION	49
FIGURE 13 : DIAGRAMME DE SEQUENCE D'INSCRIPTION D'UN ELEVE	51
FIGURE 14 : DIAGRAMME DE SEQUENCE D'AJOUT D'UN PROFESSEUR	53
FIGURE 15 : DIAGRAMME DE SEQUENCE D'AFFECTION DES DISCIPLINES AUX NIVEAUX	55
FIGURE 16 : DIAGRAMME DE SEQUENCE D'AJOUT DES DISCIPLINES.....	57
FIGURE 17 : REPRESENTATION GRAPHIQUE DES NEUDS D'ACTIVITE.....	58
FIGURE 18 : DIAGRAMME D'ACTIVITES D'INSCRIPTION D'UN ELEVE	59
FIGURE 19 : DIAGRAMME D'ACTIVITES D'AFFECTION DES NOTES D'ELEVES	60
FIGURE 20 : DIAGRAMME D'ACTIVITES CREATION D'EMPLOI DU TEMPS	61
FIGURE 21 : DIAGRAMME DE COLLABORATION DE REINSCRIPTION D'UN ELEVE	62
FIGURE 22 : DIAGRAMME DE PACKAGE	63
FIGURE 23 : DIAGRAMME DE CLASSE « GESTION D'ÉLEVE »	64
FIGURE 24 : DIAGRAMME DE CLASSE « GESTION DES PROFESSEURS »	65
FIGURE 25 : DIAGRAMME DE CLASSE « GESTION DE LA CANTINE »	65
FIGURE 26 : DIAGRAMME DE CLASSE « GESTION DU TRANSPORT »	66
FIGURE 27 : DIAGRAMME DE CLASSE « GESTION DU PAIEMENTS D'ÉLEVES ».....	66
FIGURE 28 : DIAGRAMME DE CLASSE « GESTION DES ROLES ».....	67
FIGURE 29 : ARCHITECTURE DU SYSTEME.....	74
FIGURE 30 : PATTERN MVC.....	75
FIGURE 31 : STRUCTURE DU SPRING MVC AUTOUR DU DISPATCHERSERVLET.....	76
FIGURE 32 : PAGE D'AUTHENTIFICATION « ADMINISTRATION »	78

FIGURE 33 : PAGE D'ACCUEIL	78
FIGURE 34 : PAGE D'INSCRIPTION ETAPE 1.....	80
FIGURE 35 : PAGE D'INSCRIPTION ETAPE 2.....	82
FIGURE 36 : PAGE D'INSCRIPTION ETAPE 3.....	83
FIGURE 37 : PAGE D'INSCRIPTION ETAPE 4.....	84
FIGURE 38 : PAGE DE LA LISTE D'ELEVES	85
FIGURE 39 : PAGE AJOUTER MATIERE ET DISCIPLINES (CAS 1).....	86
FIGURE 40 : PAGE AJOUTER MATIERE ET DISCIPLINES (CAS 2).....	86
FIGURE 41 : PAGE AFFECTATION DISCIPLINES VERS LES NIVEAUX.....	87
FIGURE 42 : PAGE D'ELABORATION D'EMPLOI DU TEMPS (1)	88
FIGURE 43 : PAGE D'ELABORATION D'EMPLOI DU TEMPS (2)	88
FIGURE 44 : PAGE D'ELABORATION D'EMPLOI DU TEMPS (3)	89
FIGURE 45 : PAGE D'ELABORATION D'EMPLOI DU TEMPS (4)	89
FIGURE 46 : PAGE D'ENVOI DES SMS	90

Liste des tableaux

TABLEAU 1 : FICHE TECHNIQUE D'ABNSOFT	16
TABLEAU 2 : LIVRABLES DU PROJET	28
TABLEAU 3 : LES RISQUES DU PROJET	28
TABLEAU 4 : ACTEURS PRIMAIRES	44
TABLEAU 5 : DESCRIPTION TEXTUEL D'AUTHENTIFICATION	49
TABLEAU 6 : DESCRIPTION TEXTUEL D'INSCRIPTION D'UN ETUDIANT	50
TABLEAU 7 : DESCRIPTION TEXTUEL D'AJOUT D'UN PROFESSEUR.....	52
TABLEAU 8 : DESCRIPTION TEXTUEL D'AFFECTATION DES DISCIPLINES AUX NIVEAUX.....	54
TABLEAU 9 : DESCRIPTION TEXTUEL D'AJOUT D'UNE DISCIPLINE	56

Table des matières

Dédicace	1
Remerciements	2
Résumé	3
Abstract.....	4
Liste des figures.....	5
Liste des tableaux	7
Introduction générale.....	12
Chapitre 1 : Présentation d’organisme d’accueil	14
1. Introduction.....	15
2. Présentation d’ABNSOFT	15
a. C’est quoi ABNSOFT ?	15
b. Fiche technique	16
c. Objectifs	16
d. Solutions ABNSOFT.....	17
3. Logiciel de développement « SYNERGIE »	17
a. Qu’est-ce que SYNERGIE ?	17
b. Quelques avantages de SYNERGIE	18
4. Procédures d’implantation d’une solution de gestion intégrée	19
5. Organigramme d’ABNSOFT	20
6. Présentation des principales directions	20
7. Conclusion	21
Chapitre 2 : Cahier de charge et capture des besoins.....	22
1. Introduction.....	23

2.	Besoins du client	23
3.	Besoins fonctionnels	27
4.	Besoins non fonctionnels	27
5.	Livrables du projet	28
6.	Risques du projet.....	28
7.	Conclusion	29
Chapitre 3 : Démarche du projet		30
1.	Introduction.....	31
2.	Le choix de la méthode de gestion de projet.....	31
3.	Principe de SCRUM	31
a.	Product backlog.....	32
b.	Sprint planning	33
c.	Sprint execution.....	33
d.	Daily scrum	33
e.	Potentially shippable product increment	33
f.	Sprint Review	34
g.	Sprint retrospective	34
4.	Les Rôles de Scrum	34
5.	Gestion du projet.....	34
a.	Planification	35
b.	Diagramme de gantt	36
6.	Equipe de réalisation du projet.....	39
7.	Conclusion	39
Chapitre 4 : Analyse et conception		40
1.	Introduction.....	41

2.	UML.....	41
a.	Choix d'UML.....	41
b.	Pourquoi modéliser ?.....	41
3.	Présentation des outils.....	42
4.	Identification des acteurs	43
a.	Acteur.....	43
b.	Cas d'utilisation.....	43
c.	Acteur direct et indirect.....	43
5.	Diagramme de cas d'utilisation	45
a.	Diagramme de cas d'utilisation d'acteur «Élève».....	46
b.	Diagramme de cas d'utilisation d'acteur «Professeur».....	46
c.	Diagramme de cas d'utilisation d'acteur «Administration».....	47
d.	Diagramme de cas d'utilisation «Héritage ».....	48
6.	Diagramme de séquence	48
a.	Diagramme de séquence « d'inscription d'un élève ».....	51
b.	Diagramme de séquence « d'ajout d'un professeur »	53
c.	Diagramme de séquence « d'affectation des disciplines aux niveaux ».....	55
d.	Diagramme de séquence « d'ajout des disciplines »	57
7.	Diagramme d'activités	58
a.	Diagramme d'activités « d'inscription d'un élève »	59
b.	Diagramme d'activités « d'affectation des notes d'élèves ».....	60
c.	Diagramme d'activités « création d'emploi du temps ».....	61
8.	Diagramme de collaboration.....	62
9.	Diagramme de package.....	63
10.	Diagramme de classes.....	63
a.	Diagramme de classe « gestion d'élève »	64

b.	Diagramme de classe « gestion des professeurs ».....	65
c.	Diagramme de classe « gestion de la cantine »	65
d.	Diagramme de classe « gestion du transport »	66
e.	Diagramme de classe « gestion des paiements d’élèves »	66
f.	Diagramme de classe « gestion des rôles »	67
11.	Conclusion	67
Chapitre 5 : Réalisation		68
1.	Introduction.....	69
2.	Environnement et outils de développement.....	69
a.	Environnement de développement	69
b.	Technologies et Framework adoptées	71
3.	Architecture générale de l'application.....	74
a.	Architecture logiciel du système	74
b.	Le modèle MVC	75
c.	Le DispatcherServlet	76
4.	Principales interfaces graphiques.....	77
a.	Authentification.....	77
b.	Inscription.....	79
c.	Liste d’élèves.....	84
d.	Ajouter matière et ses disciplines	85
e.	Affectation des disciplines sur les niveaux	87
f.	Génération d’emploi du temps	87
g.	Envoi des SMS	90
5.	Conclusion	90
Conclusion général		91
WEBOGRAPHIE.....		92

Introduction générale

De nos jours, la gestion des données d'une façon automatisée occupe une place privilégiée dans le monde de l'informatique. Grâce à cette technologie, le temps de recherche des données a chuté de manière considérable, ce qui laisse croire qu'elle a encore de beaux jours devant elle.

C'est dans ce contexte que plusieurs sociétés essayent de profiter au maximum possible de ces technologies afin d'améliorer leurs productivités et de faire face à quelques problèmes pénibles qui peuvent constituer un obstacle de progression.

Dans ce cadre, la société **ABNSOFT** m'a confié la responsabilité de développer une application web permettant de gérer une école privée, le cas de l'école « **Molière** », installé à Casablanca.

La naissance de cette idée est due pour répondre à un ensemble des besoins notamment : gestion des élèves (inscription - réinscription - création des emplois du temps ...); gestion des professeurs (gestion des informations et diplômes - affectation aux classes ...); gestion des niveaux et classes (créations des niveaux avec leurs classes, affecter à chaque niveau ses disciplines ...); ...

L'implémentation n'est pas la seule difficulté à surmonter, il faut savoir que la gestion d'un établissement scolaire privé est beaucoup plus complexe et différente que celle dans un établissement public, en effet les établissements privés ont un besoin de gérer le transport et la cantine, les activités parascolaires...etc.

Pour mener à bien le développement de ce projet, le choix de la méthode de développement s'est porté sur la méthode agile "**SCRUM**". En effet, la méthode SCRUM est une solution de développement logiciel adapté à tous types de projets, et parmi l'une des caractéristiques de SCRUM, c'est que le personnel participe activement à la définition des activités et des horaires, de sorte que le degré d'engagement et la motivation sont plus élevés.

Il faut noter que l'outil PowerAMC m'a aidé énormément à élaborer et gérer les différents diagrammes UML.

Pour ce qui concerne le volet technique, j'utiliserai un ensemble d'outils tels que : JavaEE(Hibernate/JPA, Spring, Spring security), JQuery, Ajax, Json, Html, Css , bootstrap.

Ayant présenté les outils et les méthodes adoptés, je vais exposer maintenant le plan de ce rapport de mémoire de fin d'étude, qui se subdivise en cinq chapitres principaux :

- Chapitre 1 : présentation d'organisme d'accueil.
- Chapitre 2 : cahier de charge et capture des besoins.
- Chapitre 3 : démarche du projet.
- Chapitre 4 : analyse et conception.
- Chapitre 5 : réalisation.

Enfin, une conclusion sera donnée enfin de ce rapport.

Chapitre 1 : Présentation d'organisme d'accueil

1. Introduction

Dans ce chapitre, je vais vous présenter l'organisme d'accueil ou j'ai effectué mon stage de projet de fin d'études, tel que la fiche technique, les objectifs, les solutions, quelques références, ainsi que le logiciel de développement utilisé par la société et enfin l'organigramme de la société.

2. Présentation d'ABNSOFT

a. C'est quoi ABNSOFT ?

Spécialisée depuis plusieurs années dans le développement, la commercialisation, la mise en place et le soutien technique de logiciels de gestion spécialisés, la société ABNSOFT est l'un des seuls éditeurs de logiciels au Maroc qui développe des solutions de gestion selon les besoins de chaque client et accompagne la PME tout au long de sa croissance. En effet, grâce à la qualité de ses produits et services et à sa profonde connaissance des besoins de la PME marocaine, l'équipe ABNSOFT est en mesure de suivre la croissance et l'évolution de la gestion de chaque client.

Son fondateur et Directeur Général Mr Abdelfattah Bennani Naciri détient une expérience de plus de 25 ans dans le développement des applications informatiques de gestion conçues pour répondre aux besoins croissants de plusieurs industries au Maroc. Il a gardé le focus depuis plus de vingt ans sur la recherche et la formation continue. De plus, Mr Bennani a acquis une expérience internationale en dirigeant quelques projets de développement au Canada.

Aujourd'hui, ABNSOFT déploie toutes ses ressources humaines et techniques et utilise une technologie moderne et performante pour mieux répondre aux besoins de ses clients. Son principal intérêt est d'offrir des solutions de gestion très fiables qui s'adaptent facilement aux exigences du marché.

b. Fiche technique

Date de création	2007
Forme juridique	S.a.r.l.
Chiffre d'affaires	1/5 Mdh
Registre commercial	171213
Adresse	52, bd Abdelmoumen, résid. Almanar 7^oét. - 20340 Casablanca
Activités	Editeur de logiciels de gestion

Tableau 1 : Fiche technique d'ABNSOFT

c. Objectifs

La Société ABNSOFT s'engage à offrir les meilleures applications de gestion adaptées aux entreprises marocaines (petites, moyennes et grandes) afin de les rendre plus compétitives.

Leur objectif est de perfectionner les solutions de gestion pour aider l'entreprise marocaine à mieux exploiter les systèmes informatiques, accroître son rendement et participer ainsi au progrès informatique, scientifique et économique du pays.

Leur vision mondiale est d'offrir des solutions innovatrices à l'échelle internationale avec un projet en cours pour le Canada et les Etats-Unis.

ABNSOFT est toujours à l'écoute de ses prospects, clients et partenaires afin d'offrir un outil de gestion efficace et un service de qualité.

d. Solutions ABNSOFT

- Logiciel Paie
- Logiciel Comptabilité
- Logiciel Finances
- Logiciel Gestion commerciale
- Logiciel Gestion de stock
- Logiciel ERP
- Logiciel Gestion des ressources humaines
- Logiciel Facturation
- Logiciel Gestion des immobilisations
- Logiciel Liasse fiscale
- Logiciel Gestion de la caisse
- Logiciel CRM
- Logiciel SRM
- Logiciel SAV
- Logiciel Point de vente
- Logiciel Parc-auto
- Logiciel Gestion du lait
- Progiciel de gestion intégrée - PGI
- Solution Expert-comptable
- Solution Fiduciaire

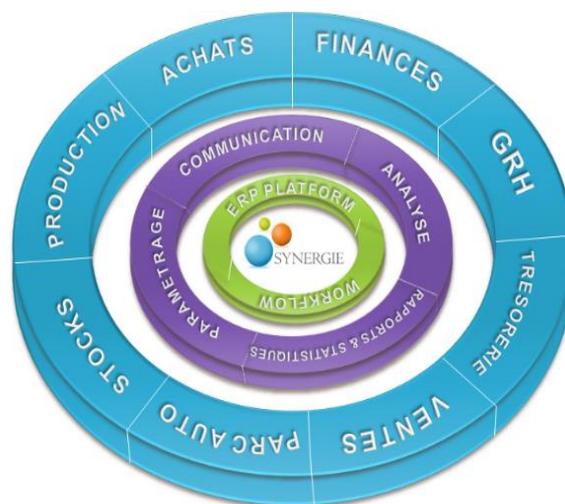


Figure 1 : Les modules de gestion couverts par SYNERGIE

3. Logiciel de développement « SYNERGIE »

a. Qu'est-ce que SYNERGIE ?

SYNERGIE est un ERP performant, efficace et simple qui couvre parfaitement les besoins en gestion et en pilotage de la PME marocaine. Conçu dans l'ultime but d'accroître la productivité et les performances des entreprises, ce logiciel est doté d'une interface conviviale, de plusieurs outils intelligents et de fonctionnalités avancées permettant un accès simple et facile à l'information.

SYNERGIE répond aux besoins de plusieurs secteurs ou métiers et englobe plusieurs modules de gestion qui peuvent être installés de façon intégrale ou modulaire-évolutive selon la croissance, les besoins et les priorités de chaque entreprise.

b. Quelques avantages de SYNERGIE

- Mise en œuvre rapide et récupération des anciennes données
- Simplicité du paramétrage et de l'utilisation
- Développement, assistance et suivi personnalisé
- Normalisation des procédures et unicité de la base de données
- Conformité avec la législation, l'administration fiscale et la gestion de l'entreprise marocaine
- Partage des données en temps réel en évitant toute redondance d'information
- Modification simple et flexible de tous les paramètres du logiciel
- Personnalisation du menu avec des images raccourcis
- Recherche rapide de toutes les fonctionnalités du logiciel
- Gestion des droits et accès par utilisateur et par dossier
- Importation et exportation facile des données sur Excel
- Exploitation graphique des données en deux clics
- Exportation de tous les états émis par le logiciel en format WORD, EXCEL et PDF
- Envoi par email en un clic de tous les états en format PDF
- Consultation rapide des informations qui proviennent de tous les exercices stockés dans la base de données.

En effet, SYNERGIE est un outil de communication intelligent, un moyen de pilotage efficace et une solution métier dotée de plusieurs tableaux de bord décisionnels parfaitement adaptés aux besoins de la PME Marocaine.

4. Procédures d'implantation d'une solution de gestion intégrée

L'objectif d'ABNSOFT est d'offrir à leurs clients une solution simple, efficace et complète qui leur assure une exploitation optimale des ressources et un accès instantané à l'information tout en respectant leur gestion stratégique et budgétaire.

Voici les procédures d'implantation d'une solution de gestion intégrée :

- Détermination des besoins réels du client par l'écoute, le partage d'idées et les propositions.
- Exploitation efficace des ressources et valorisation des moyens.
- Respect des budgets et contrôle des coûts.
- Respect des budgets et contrôle des coûts.
- Etude et analyse précise du projet.
- Désignation de l'équipe appropriée.
- Evaluation des procédures et systèmes existants.
- Vérification et validation du cahier de charges.
- Gestion et exécution du projet.
- Test et évaluation de tous les modules de la solution.
- Assistance à l'installation et à l'utilisation de la solution.
- Formation complète du personnel concerné.
- Maintenance, soutien et garantie des prestations.
- Suivi et optimisation des solutions.

5. Organigramme d'ABNSOFT

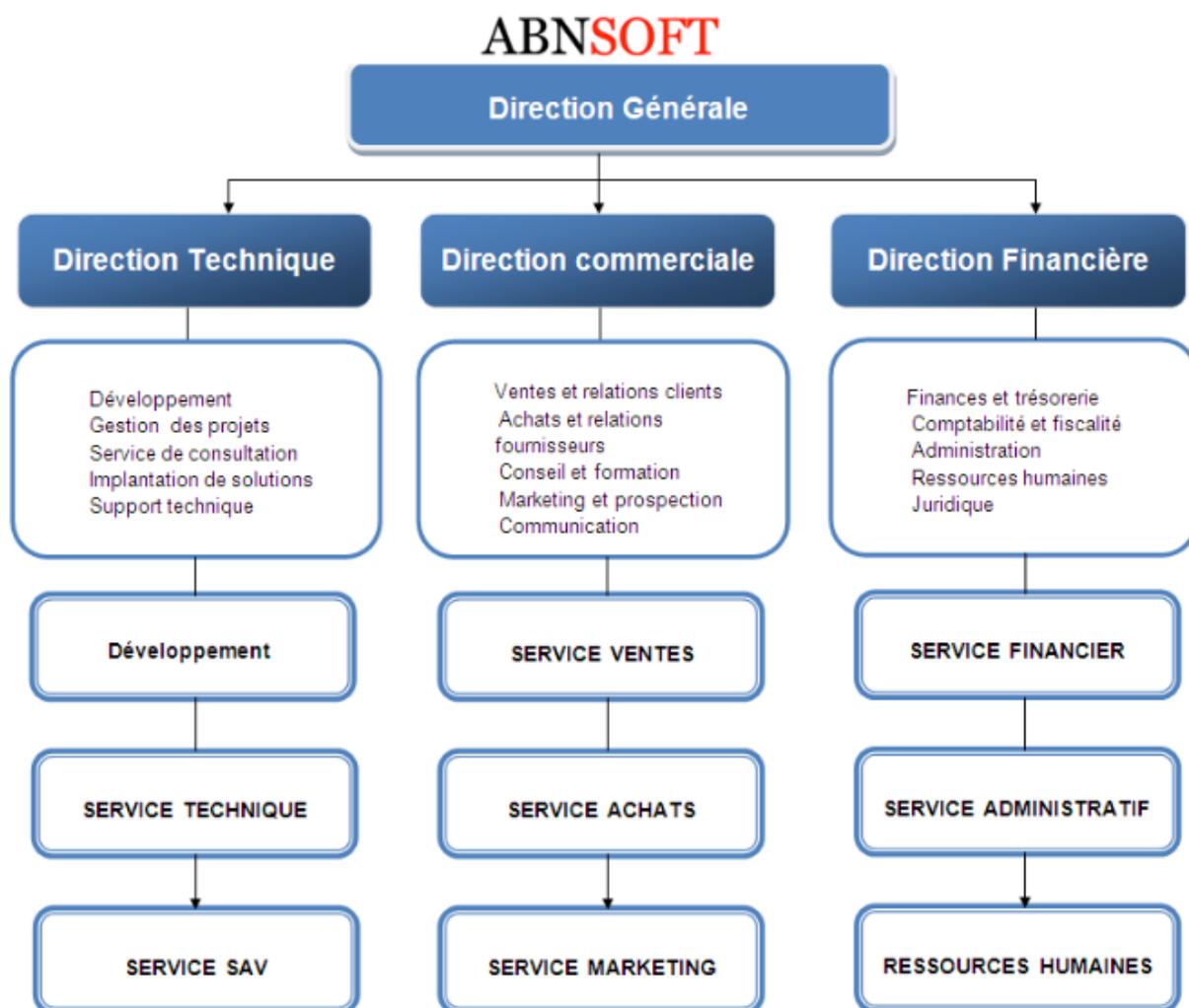


Figure 2 : Organigramme d'ABNSOFT

6. Présentation des principales directions

ABNSOFT se présente comme une entreprise dynamique et innovante, elle se divise en Plusieurs directions, chaque direction se caractérise par un rôle précis :

- **La direction générale:**

Supervisée par un Directeur Administratif et financier qui veille à L'accomplissement de toutes les opérations budgétaires, le contrôle et coordination, L'approvisionnement, le marché, les achats, la gestion du personnel et des affaires sociales.

- **La direction technique :**

A pour principale mission est de superviser l'ensemble des projets informatiques de l'entreprise. Qui se charge aussi de définir la stratégie de la société en matière d'innovation. Il faut savoir que ce métier n'est pas seulement technique : il joue aussi le rôle de commercial auprès des prospects et des clients. En effet, pour chaque projet, il se doit de proposer une solution innovante et répondant parfaitement aux besoins des clients.

- **La direction commerciale :**

Cette direction a la responsabilité de gérer au mieux les besoins de la clientèle de la société et d'offrir des produits de qualité à des prix compétitifs, d'assurer une meilleure rentabilité et satisfaire les clients.

- **La direction financière :**

Cette direction a la responsabilité de gérer au mieux la comptabilité, trésorerie, fiscalité, contrôle de gestion, audit, droit, ressources humaines, communication, système d'information, services généraux, achats.

7. Conclusion

A travers ce chapitre, on a maintenant une idée sur la société ABNSOFT avec son activité, son objectif, le logiciel de développement, quelques références, la répartition des responsables, d'ensembles de tâches entre les postes, et les relations de commandement qui existent entre eux.

Chapitre 2 : Cahier de charge et capture des besoins

1. Introduction

Ce chapitre a pour rôle de montrer les besoins du client, ainsi que les besoins fonctionnels et non fonctionnels de la future application, de préciser les livrables et les risques les plus critiques du projet.

2. Besoins du client

Au cours de mon stage concernant le projet de fin d'études au sein du service informatique, la société **ABNSOFT** souhaite établir la mise en œuvre d'un portail-web sur la gestion d'école privée **Molière** qui comporte un ensemble de modules :

➤ Gestion des élèves :

Ce module contient quatre volets qui sont : l'inscription, la réinscription, les absences et les retards ainsi que l'exploitation des notes obtenues.

- **L'inscription** qui se divise en deux.

Celle des élèves qui vont s'inscrire pour la première fois en première année du primaire. En effet, cet acte est statué par l'obligation d'avoir l'âge réglementaire de scolarisation qui est fixé à 6 ans avec autorisation d'accepter les élèves nouvellement inscrits et d'allonger la date d'inscription à 3 mois avant l'âge de six ans (c'est-à-dire : les élèves qui sont nés du 01 janvier jusqu'au 31 mars).

Les parents des intéressés doivent présenter à l'école un dossier orange (à tirer de l'école ou à acheter chez les libraires) après avoir rempli la partie réservée à l'état civil concernant la date de naissance de l'élève. Parmi les pièces d'inscription, on trouve les photos d'identité de l'élève, la photocopie de la CIN des parents ou tuteurs, l'adresse, le GSM et le fixe s'il existe.

Enfin, les parents ou tuteurs doivent régler les frais d'inscription, les frais de scolarisation, les frais du transport et de la cantine (les deux derniers sont facultatifs).

Les élèves nouveaux dans l'établissement ayant présenté un certificat de sortie réglementaire certifié homologué par la délégation régionale, ces élèves qui ont un niveau supérieur à la 1^{er} année du primaire peuvent bénéficier aussi d'une inscription à condition d'avoir une place vacante dans le niveau souhaité et remplir les clauses de l'inscription telle

que : la photo de l'élève, son acte de naissance et les informations personnelles, les coordonnées des parents ou tuteurs.

Le règlement des frais d'inscription, de scolarisation, des droits de transport et de la cantine sont obligatoires sauf les deux derniers points qui sont facultatifs.

- **La réinscription** des anciens élèves de l'école se fait si :

L'élève est en situation réglementaire, c'est-à-dire si son nom figure sur la liste des élèves admis en classe supérieure ou ayant le droit au doublage.

La réinscription se fait après la présentation des pièces demandées et après le règlement des frais d'inscription, des droits de scolarisation et des services désirés tel que le transport et la cantine.

- **Les retards et les absences :**

Le traitement des absences et retards des élèves est réglementé par le code interne de l'établissement.

Les absences et les retards peuvent être justifiés ou non, dans le deuxième cas : les parents ou tuteurs doivent signer un engagement pour ne pas donner la totalité de la responsabilité à l'école vis-à-vis de l'absence ou du retard non justifié de leur enfant.

L'exploitation et le traitement des notes obtenues par les élèves dans les contrôles continus visent à suivre et à mieux contrôler le travail de l'élève, son évolution ou son retard scolaire est l'une des préoccupations de l'administration.

Les parents sont avisés aussi du comportement de leurs enfants au moment opportun, ils sont convoqués à l'école pour décider de la prise d'une mesure éducative à prendre.

Exemple : Soutien de l'élève dans telle ou telle matière...

- **Gestion des emplois du temps :**

Ce module est divisé en deux, celui des professeurs et celui des élèves.

Celui des professeurs : l'administration établit un emploi de temps cohérent répondant aux directives et aux normes des instructions officielles du ministère de l'éducation nationale pour

chaque niveau et pour chaque discipline, en respectant le nombre d'heures pour tel ou tel niveau concernant telle ou telle matière.

L'emploi du temps une fois établi par l'administration qui l'envoie à l'inspecteur de la matière pour visa.

Celui des élèves : l'emploi du temps doit être compatible non chargé (maximum 7 heures/jours), il doit comporter les matières autorisées à être enseignée dans tel ou tel niveau.

➤ **Gestion des professeurs :**

Deux catégories de professeurs sont à mentionner :

Les vacataires : ceux qui travaillent par heure.

Les permanents : sont ceux qui sont engagés mensuellement par contrat.

La gestion des professeurs consiste à tenir compte de leur dossier professionnel (diplômes, ancienneté,). Elle consiste à tenir compte de leurs absences et retards (justifiés ou non) durant l'année scolaire.

➤ **Gestion des parents :**

Consiste à noter les informations relatives aux parents et tuteurs des élèves inscrits, tels que : l'adresse familiale, le GSM ou le N° du fixe.

L'intérêt et l'avantage de cette application vise à rester en contact avec l'administration de l'école pour donner une vision vivante sur le travail et le comportement journalier de l'élève.

➤ **Gestion de la cantine :**

Le programme de la cantine scolaire est présenté à l'avance sur un calendrier visé par l'hygiène scolaire.

La présentation du menu de la semaine suivant un emploi de temps précis qui ne perturbe pas le déroulement des cours. Ensuite, les élèves bénéficiaires de la cantine scolaire sont affectés au sein d'un groupe de la cantine sous la surveillance des responsables (maitre de garde, accompagnateur, ...).

➤ Gestion des activités parascolaires :

Parmi les activités parascolaires, on trouve les activités artistiques, les activités écologiques et les excursions....

Chaque activité possède un leader ou chef de groupe, ce dernier est le responsable direct vis-à-vis de l'administration qui lui demande d'établir une programmation à réaliser suivant un calendrier trimestriel, semestriel ou annuel. Ce calendrier doit comporter et préciser les différents intervenants, les bénéficiaires, les dates d'exécution, les ressources et les moyens disponibles.

A la de l'activité, le responsable doit établir un rapport détaillé sur l'exécution de telle ou telle activité, les inconvénients et les bienfaits de l'activité, comme il doit présenter un dossier illustré par des photos, des CD montrant la réalisation concrète de l'activité (classement du dossier de l'activité dans le patrimoine culturel de l'école).

La réalisation des activités parascolaires vise à casser la routine scolaire à dynamiser et faire renouveler l'effort scolaire de l'élève.

➤ Les campagnes SMS :

La messagerie SMS reste le moyen de communication le plus efficace. La procédure de lancement d'une campagne SMS est très facile.

Un module de communication par SMS pour communiquer facilement avec les parents d'élèves afin de les notifier pour :

- Des absences des élèves.
- Des rappels de paiement.
- ...

La communication par SMS présente plusieurs avantages:

- L'utilisateur du téléphone portable est averti instantanément.
- Le SMS est un moyen percutant susceptible d'attirer l'attention plus qu'un email.
- Le message est lu immédiatement.

3. Besoins fonctionnels

Avant d'imposer une solution, il faut se tourner vers le demandeur, pour aboutir de manière structurée à la solution. En effet, le but du projet est de satisfaire le besoin. Il faut exprimer clairement les objectifs à atteindre du projet, afin d'éviter toute confusion entre nous et le demandeur.

Le futur système doit permettre à l'utilisateur de gérer un ensemble de modules, dont on trouve :

- ✓ Gestion des élèves.
- ✓ Gestions des emplois du temps.
- ✓ Gestions des professeurs.
- ✓ Gestions des parents.
- ✓ Gestion de la cantine et du transport.
- ✓ Gestion des activités parascolaires.
- ✓ L'accompagnement du pack SMS.

4. Besoins non fonctionnels

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d'objectifs spécifiques que doit atteindre le système.

La maintenabilité : le code doit être compréhensible par simple lecture, notamment en respectant les règles de gestion et les normes de développement.

Exploitabilité : les impacts en termes de performances doivent être pris en compte lors de développement, ainsi que la consommation des ressources (CPU, mémoire, etc.) qui doit être minimisée.

Capacité fonctionnelle et convivialité : les composants développés doivent respecter les spécifications fournies par ABNSOFT. Le système doit être facilement utilisable et disposer d'interfaces conviviales, notamment par le respect des règles d'ergonomie d'ABNSOFT.

Sécurité : Chaque utilisateur, pour accéder à l'application, est obligé de s'authentifier par un nom d'utilisateur et un mot de passe. Il ne pourra accéder qu'aux pages qui lui sont permises par son profil ou les droits d'accès qui lui sont affectés par l'administrateur.

Les mots de passe ne sont enregistrés dans la base qu'après un chiffrement (MD5).

5. Livrables du projet

Phase	Livrable	Responsable
Etude des besoins	Cahier de charge	Hamid kably
Analyse et conception	Diagrammes UML	Hamid kably
codage et test	Application web	Hamid kably
Documentation	Rapport du projet	Hamid kably

Tableau 2 : Livrables du projet

6. Risques du projet

Les risques	Le type	Impact	Action correctives
Cahier des charges	Risque non bloquant	Créer une ambiguïté ce qui pourra générer un retard qui influencera probablement la date de livraison.	Prévoir des réunions et des points de validation avec l'encadrant au fur et à mesure de l'avancement du projet.
Les pannes inattendues du Matériel	Risque non bloquant	Ralentissement des travaux	Utiliser les autres matériaux disponibles. Recours à une réparation rapide.
Absence ou maladie	Risque non bloquant	Ralentissement des travaux	Doubler l'effort et travailler un temps extra.

Tableau 3 : Les risques du projet

7. Conclusion

A la fin de ce chapitre, on a bien étudié les besoins du client ; on a présente l'ensemble des fonctionnalités du futur portail de manière organisée dans les différents cycles de l'application soit fonctionnel ou non fonctionnel, les livrables ainsi que les risques du projet.

Chapitre 3 : Démarche du projet

1. Introduction

Le présent chapitre décrit la conduite du projet. Il présente dans un premier lieu le choix de sa démarche, puis le principe de cette démarche, les rôles du SCRUM, et enfin l'organisation du projet.

2. Le choix de la méthode de gestion de projet

Le choix de la méthode de développement s'est porté vers la méthode **SCRUM**.

SCRUM est la méthode Agile la plus utilisée parmi les autres méthodes Agile. Et de fait, la plus éprouvée.

D'autre part, **SCRUM** est un processus itératif et incrémental, représente un framework de développement logiciel agile pour la gestion du développement des produits.

Il définit « une approche souple, stratégie de développement de produits holistique et permet aux équipes de développement de s'organiser comme une unité pour atteindre un objectif commun ».

L'une des particularités de **SCRUM** est que pendant le développement de produits, les clients peuvent changer d'avis sur ce qu'ils veulent et ont besoin (souvent appelé la volatilité des exigences).

3. Principe de SCRUM

SCRUM est une méthode agile dédiée à la gestion de projet. Cette méthode de gestion a pour objectif d'améliorer la productivité de son équipe.

La méthode SCRUM implique que le projet progresse à travers la mise en place de séries de « sprints ». A chaque lancement d'un sprint, une réunion de planification est organisée afin que chaque membre de l'équipe puisse s'engager sur le nombre de tâches qu'il pourra exécuter, ainsi que sur la création du « sprint backlog », qui est la liste globale des tâches à réaliser lors du sprint.

Chaque jour du sprint, tous les membres de l'équipe (ainsi que le responsable produit et le SCRUM Master) doivent assister à la réunion SCRUM quotidienne. Cette dernière ne doit pas durer plus de 15 minutes, et permet aux membres de l'équipe de partager aux autres ce qu'ils

ont fait la veille, ce sur quoi ils travaillent le jour même, ainsi que l'identification de tout problème pouvant entraver le bon déroulement du sprint. Cette réunion permet ainsi de synchroniser tous les membres de l'équipe.

La fin d'un sprint est marquée par une session de débriefing permettant de présenter le travail achevé au responsable produit, et de partager des informations pouvant influencer sur le sprint suivant.

Voilà un schéma qui représente le processus de la méthodologie SCRUM, avec un détail de chaque étape :

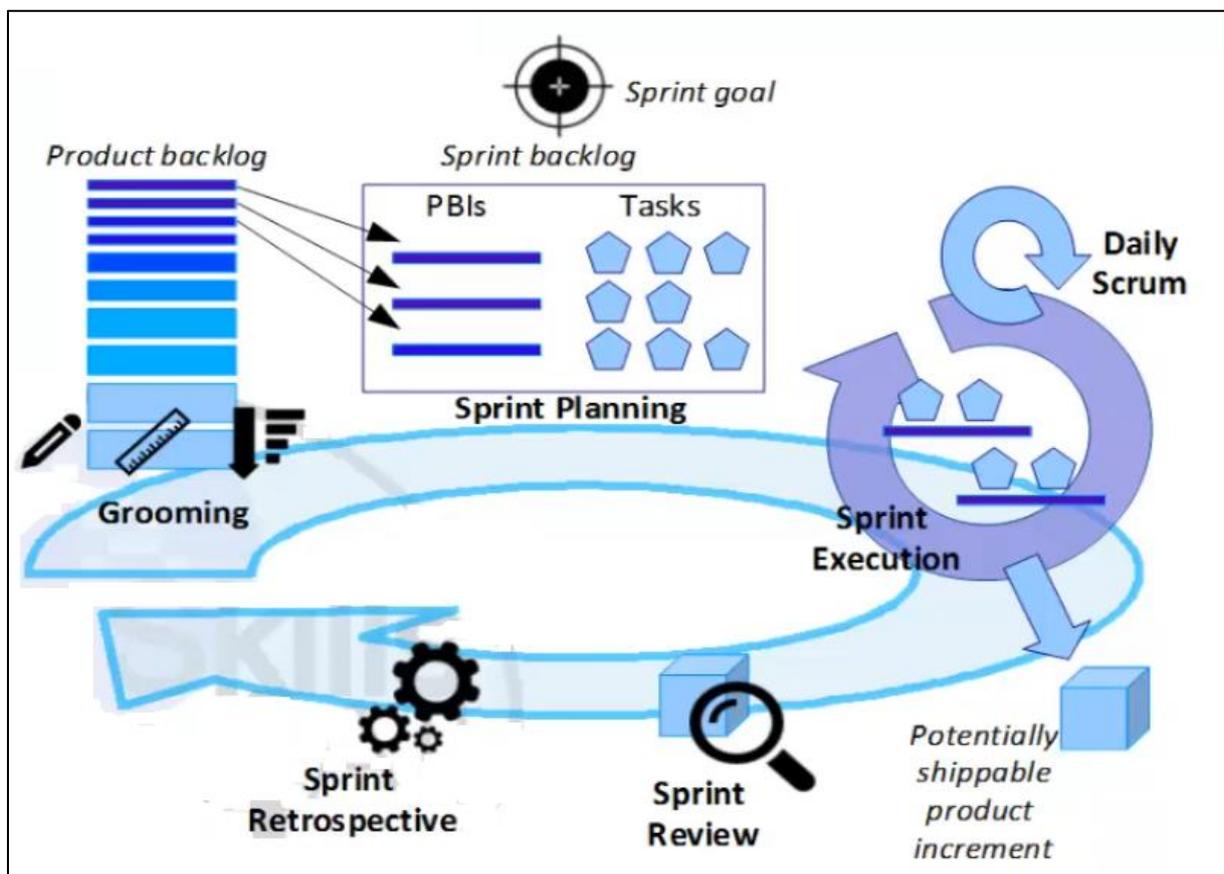


Figure 3 : Démarche du projet

a. Product backlog

Les utilisateurs constituent un **produit de backlog**, qui va être composé de toutes les demandes de fonctionnalités prioritaires. C'est pour cela que dans le produit backlog, on le dépile par le haut, et le haut du produit backlog représente les fonctionnalités les plus demandées et les plus urgentes, qu'il va falloir réaliser en premier.

b. Sprint planning

Comme l'indique le schéma, on va sélectionner les PBI « les produits de backlog items » qui sont les plus demandés par l'utilisateur, c'est-à-dire qui sont en haut de la pile. Puis dans le **sprint planning** on va décomposer les PBIs en tâches auxquelles on va donner une durée en heures.

c. Sprint execution

Une fois on a fait ce sprint planning c'est-à-dire qu'on a déterminé l'ensemble des tâches qui vont être réalisées pendant cette itération ce qu'on appelle un sprint. Eh bien là, on va exécuter l'itération, dans **sprint execution**, et c'est là où on a une équipe de projet, une équipe de développement qui vont réaliser les incréments de fonctionnalités.

d. Daily scrum

Tous les jours on a un petit événement qui s'appelle le **daily scrum** pendant lequel on passe en revue ce qui a été développé la veille, ce qui sera développé demain et comment on va s'y prendre.

Cet événement très court qui prend peu de temps « point rapide sur ce qui s'est déroulé la veille, sur ce qui doit se dérouler aujourd'hui et puis suivant le travail ».

e. Potentially shippable product increment

Une fois qu'on a fait l'ensemble des itérations, normalement on devrait avoir vingt itérations à peu près égales, maintenant on va avoir probablement un ou plusieurs incréments de fonctionnalités qui sont potentiellement intégrables dans le produit.

C'est pour cela qu'on parle en anglais de **Potentially shippable product increment**, c'est-à-dire à la fin de mon itération je ne vais pas forcément livrer quelque chose, en tout cas, j'ai quelque chose qui devrait pouvoir être mis en production, peut-être pas la fin de cette itération, peut-être la fin d'une autre itération, mais c'est quelque chose de tangible.

Donc on va montrer tout cela aux utilisateurs pour avoir le retour hi-média, et d'ailleurs une fois qu'on a montré aux utilisateurs ce qui convient de développer, ils vont pouvoir donner l'accord, faire des modifications ou ajouter de nouveaux PBIs.

f. Sprint Review

L'évènement suivant c'est sprint review, dans lequel on passe en revue ce qui a été réalisé pendant l'itération avec l'utilisateur et puis le passer à ce moment-là avec ce qui «est demandé par les utilisateurs, de ce qu'il va falloir encore ajouter au produit backlog, ou supprimer. C'est à ce moment-là qu'on fait la revue de ce que le sprint a délivré.

g. Sprint retrospective

C'est la dernière étape dans cette méthodologie, est là on passe en revue les éléments de la méthodologie Scrum dans l'entreprise « ce qui a fonctionné ou ce qui n'a pas fonctionné, en fait, c'est une revue de la méthodologie ».

4. Les Rôles de Scrum

Scrum définit seulement 3 rôles :

- **Le Product Owner** : qui porte la vision du produit à réaliser et travaille en interaction avec l'équipe de développement. Il s'agit généralement d'un expert du domaine métier du projet.
- **L'Equipe de Développement** qui est chargée de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables. Elle est pluridisciplinaire et peut donc encapsuler d'autres rôles tels que développeur, architecte logiciel, DBA, analyste fonctionnel, graphiste/ergonome, ingénieur système.
- **Le Scrum Master** qui doit maîtriser SCRUM et s'assurer que ce dernier est correctement appliqué. Il a donc un rôle de coach à la fois auprès du Product Owner et auprès de l'équipe de développement. Il doit donc faire preuve de pédagogie. Il est également chargé de s'assurer que l'équipe de développement est pleinement productive.

5. Gestion du projet

Cette partie a pour objectif de présenter la manière dont j'ai géré le projet. Je vais présenter dans un premier temps comment le projet a été découpé en tâches afin d'assurer son bon déroulement.

Pour cela, pendant la première semaine, j'ai pris connaissance du sujet, établi les fonctions nécessaires pour remplir les besoins, fait des recherches sur les technologies que je pourrai utiliser et déterminer les tâches à accomplir.

J'ai choisi comme outil pour la planification de mon projet Gantt Project qui est un logiciel de gestion de projet. Gantt Project permet de planifier les projets et les ressources, et d'assurer le suivi des projets durant leur réalisation. Cet outil permet ainsi au chef de projet d'assurer une gestion de projet professionnelle, conforme à l'état de l'art, et ainsi garantir le respect des délais.

a. Planification

Comme le montre le diagramme de Gantt (fig : 5), le projet s'étale sur quatre mois où j'ai commencé par :

- **Analyse du problème** : un projet ne peut pas être entrepris sans connaître l'environnement et sans avoir analysé en détail le sujet, afin de pouvoir dégager les fonctionnalités à développer.
- **Rédaction du cahier des charges** : est un document qui doit être respecté lors de la réalisation d'un projet.
- **Spécification des besoins fonctionnels et non fonctionnels** : dégager les fonctionnalités du système, afin de décrire de la façon la plus précise possible les besoins auxquels le nouveau système doit répondre.
- **Etude technique** : le choix des outils de développement.
- **Conception de l'application**.

Après ces cinq tâches, on a débuté la réalisation de la future application, mais d'une manière découpée. C'est-à-dire on a divisé la réalisation de l'application sur des modules (sprint) avec des livrables, pour que l'on puisse à la fin avoir un projet complet à livrer. Donc pour cela, on a découpé le projet sur trois modules qui représentent des sprints :

- Module 1 : gestion des élèves – gestion des parents – gestion des professeurs.
- Module 2 : gestion des niveaux et classes – gestion des emplois du temps – gestion des notes et évaluations.
- Module 3 : gestion du transport et cantine – gestion des activités para scolaires – la campagne sms.

Dans chaque sprint on trouve quatre sous tâches :

- Sprint planning: dans le sprint planning on va décomposer le besoin en tâche et ces tâches on va leur donner une durée.
 - Sprint execution: là où je vais faire le développement de l'application, et de réaliser les incréments de fonctionnalités.
 - Sprint review: dans lequel, on passe en revue ce qui a été réalisé pendant l'itération.
 - Sprint retrospective: là on passe en revue les éléments de la méthode Scrum dans l'entreprise « ce qui a fonctionné ou ce qui n'a pas fonctionné».
- Teste et livraison du projet.
 - Rédaction du rapport.

b. Diagramme de gantt

Le diagramme de Gantt, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet.

Ce diagramme permet donc de visualiser d'un seul coup d'œil :

- Les différentes tâches à envisager
- La date de début et la date de fin de chaque tâche
- La durée escomptée de chaque tâche
- Le chevauchement éventuel des tâches, et la durée de ce chevauchement
- La date de début et la date de fin du projet dans son ensemble



Nom	Date de début	Date de fin
• Analyse du Problème	20/01/17	23/01/17
• Redaction du cahier des charges	24/01/17	26/01/17
• Specification des besoins fonctionnels et n...	27/01/17	30/01/17
• Etude technique	31/01/17	02/02/17
• Conception du systeme	03/02/17	20/02/17
☐ • Sprint 1	21/02/17	24/03/17
• Sprint planning	21/02/17	22/02/17
• Sprint execution	23/02/17	20/03/17
• Sprint review	21/03/17	22/03/17
• Sprint retrospective	23/03/17	24/03/17
☐ • Sprint 2	27/03/17	27/04/17
• Sprint planning	27/03/17	28/03/17
• Sprint execution	29/03/17	21/04/17
• Sprint review	24/04/17	25/04/17
• Sprint retrospective	26/04/17	27/04/17
☐ • Sprint 3	28/04/17	29/05/17
• Sprint planning	28/04/17	01/05/17
• Sprint execution	02/05/17	25/05/17
• Sprint review	26/05/17	26/05/17
• Sprint retrospective	29/05/17	29/05/17
• Teste et livraison du projet	30/05/17	02/06/17
• Rédaction du rapport	27/01/17	06/06/17

Figure 4 : planning du projet

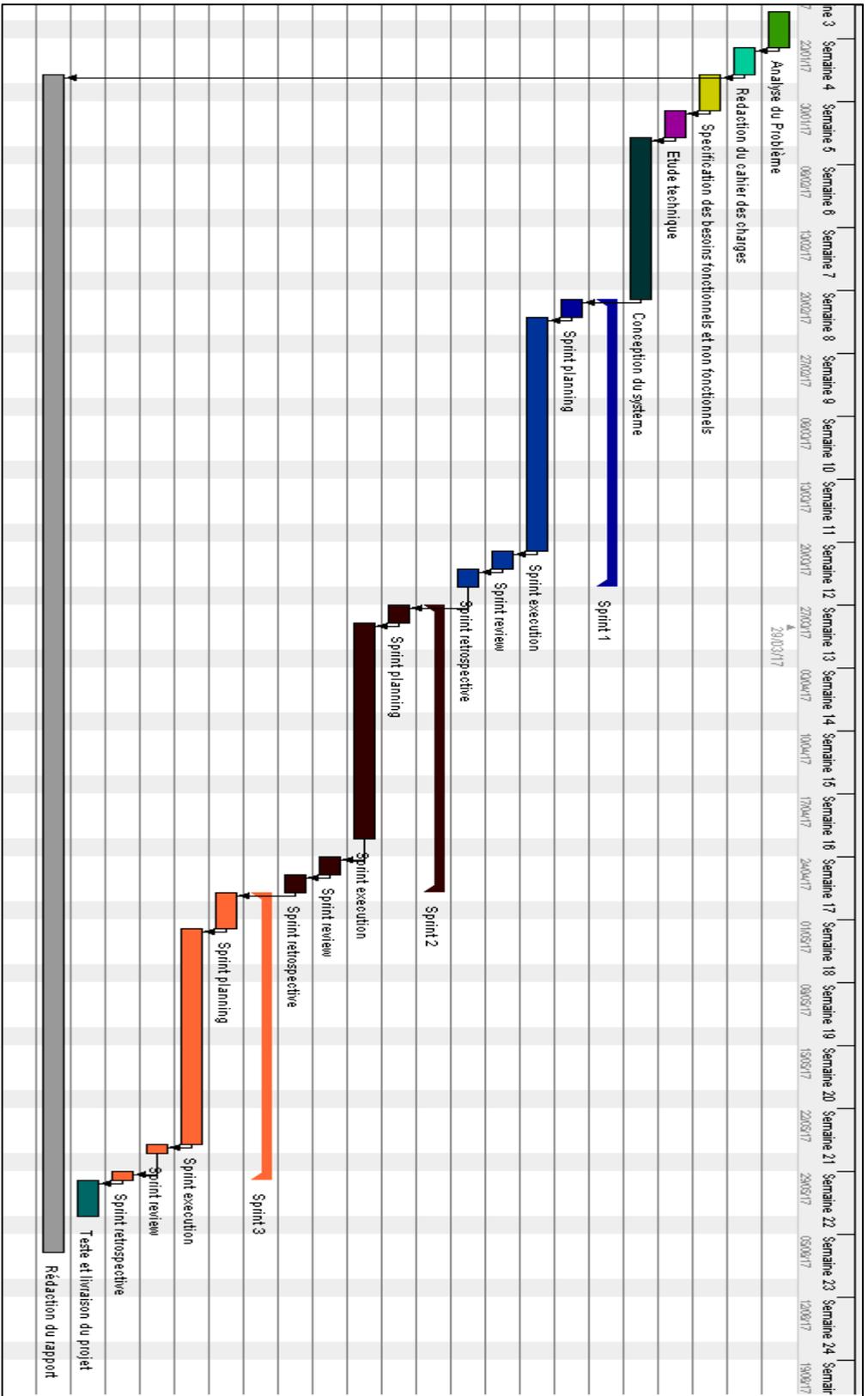


Figure 5 : diagramme de gantt

6. Equipe de réalisation du projet

Nom & prénom	Fonction / Rôle dans le projet
Hamid Kably	Stagiaire / développeur
Med Chaouki Abounaima	Encadrant
Badr Bennani Naciri	SCRUM MASTER

- **Développeur :**
 - Elaboration du dossier de gestion de projet.
 - Réalisation de la spécification détaillée.
 - Codage de l'application.
 - Effectuation des tests unitaires.
- **SCRUM MASTER :**
 - Valide le dossier des spécifications fonctionnelles.
 - Valide le codage.
 - Présentation des besoins fonctionnels du projet.
 - Valide les livrables.
 - Contrôle le respect des demandes.

7. Conclusion

Dans ce chapitre, on a organisé notre projet on utilisant la méthode SCRUM et en faisant un planning bien détaillé avec la définition d'équipe de projet.

Chapitre 4 : Analyse et conception

1. Introduction

Dans cette partie, on va analyser et modéliser les besoins du client avec le langage UML.

L'activité d'analyse et de conception permet de traduire les besoins fonctionnels et les contraintes issues du cahier des charges et de la spécification des exigences dans un langage plus professionnel et compréhensible par tous les individus intervenants dans la réalisation et l'utilisation de l'application.

2. UML

a. Choix d'UML



UML, c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

b. Pourquoi modéliser ?

De la même façon qu'il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de le réaliser.

Modéliser, c'est décrire de manière visuelle et graphique les besoins, les solutions fonctionnelles et techniques du projet.

Modéliser pour :

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Faire des simulations avant de construire un système.

3. Présentation des outils



PowerDesigner (anciennement PowerAMC) est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

PowerAMC propose différentes techniques de modélisation, chacune accessible aux informaticiens de tout niveau, parmi elles : Merise, UML, Data Warehouse, et processus métiers. Simple d'utilisation, personnalisable et dotée d'une interface intuitive, cette application optimise les productivités individuelle et collective. Elle intègre en outre des fonctions de génération de code pour plus de 45 bases de données et divers langages de programmation.

4. Identification des acteurs

a. Acteur

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Il se représente par un petit bonhomme (figure 6) avec son nom (son rôle) inscrit dessous.

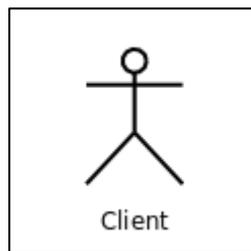


Figure 6 : Exemple de représentation d'un acteur

b. Cas d'utilisation

Un cas d'utilisation est un service rendu à un acteur : c'est une fonctionnalité de son point de vue.

Un cas d'utilisation se représente par une ellipse (figure 7) contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.

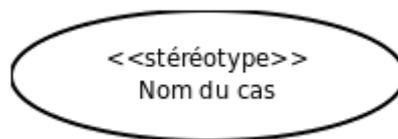


Figure 7 : Exemple de représentation d'un cas d'utilisation

c. Acteur direct et indirect

- Acteurs directs :

Les acteurs directs, ce sont les utilisateurs de l'application, qui touchent directement l'application.

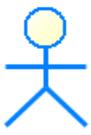
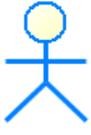
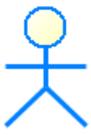
Type d'acteur	Description fonctionnelle
 Élève	L'acteur le plus important, qui aura les fonctionnalités suivantes : <ul style="list-style-type: none"> • Consulter son emploi du temps. • Consulter le planning du transport et cantine. • Voir ses absences et retards. • Etc...
 Professeur	A pour fonctionnalités : <ul style="list-style-type: none"> • Consulter son emploi du temps. • Consulter la liste de ses élèves. • Organise et insère les évaluations. • Etc...
 Administration	Ce type d'acteur aura les fonctionnalités suivantes : <ul style="list-style-type: none"> • Gestion des élèves. • Gestion des parents et tuteurs. • Gestion des personnels. • Etc...

Tableau 4 : Acteurs primaires

- **Acteurs indirects :**

Ce sont les acteurs qui n'utilisent pas l'application, mais participent d'une façon indirecte dans le processus.

Chauffeurs :

- Conduite d'un véhicule scolaire (aller – retour).
- Entretien du véhicule et sa propreté.
- Faire des courses pour l'école.

Accompagnateur :

- Assure les ordres des entrées et sorties des élèves.
- Responsable de la porte principale du bus scolaire (entrée + sortie).
- Donne le signal de départ et ordonne l'arrêt du bus.
- Accompagne les petits élèves vers les WC et dans les bus.

Cuisiniers :

- Préparent à manger pour les élèves qui sont inscrits à la cantine à midi suivant un programme de restauration visé par les autorités compétentes (santé – délégation..).
- Préparent le goûter lors des récréations (matin – après midi).
- L'hygiène et la propreté de la cuisine et du local de restauration.

Stagiaires :

- Sont des aides enseignants qui font un stage de formation limité et sans engagement.

5. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

Dans les figures qui suivent, nous présenterons les cas d'utilisation qui mettent en évidence les principales fonctionnalités de chaque acteur dans le système.

a. Diagramme de cas d'utilisation d'acteur «Élève»

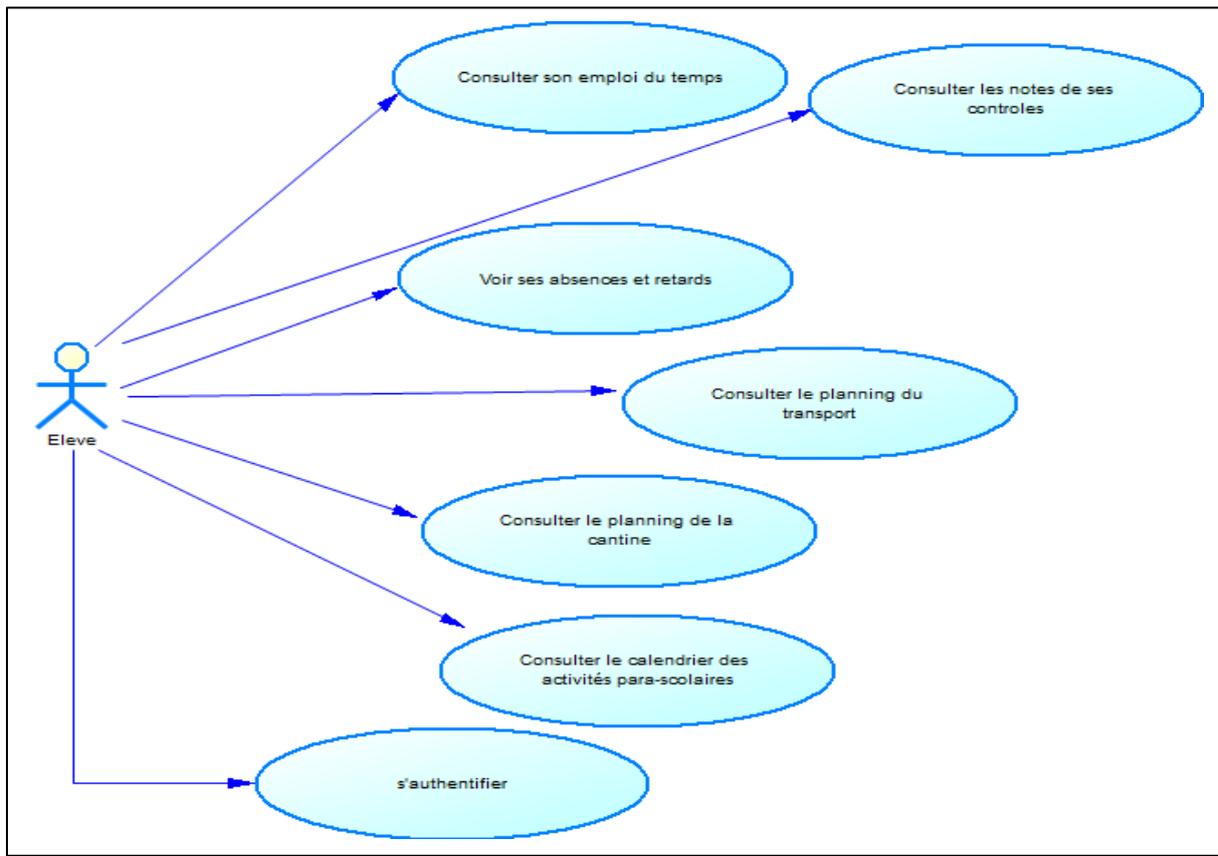


Figure 8 : Diagramme de cas d'utilisation d'acteur «Élève»

b. Diagramme de cas d'utilisation d'acteur «Professeur»

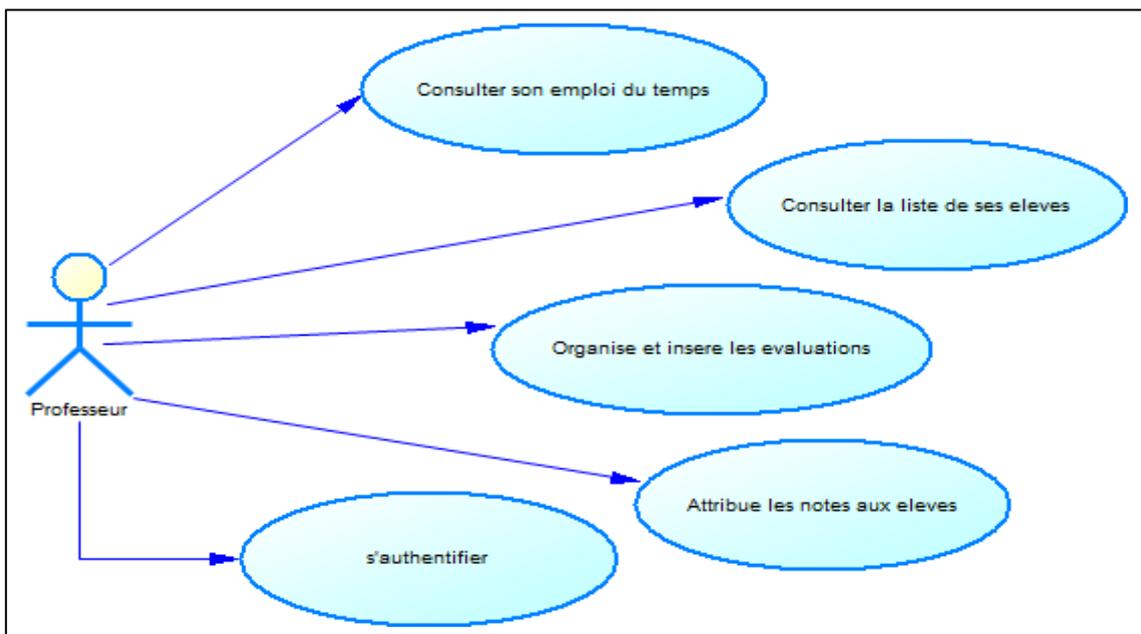


Figure 9 : Diagramme de cas d'utilisation d'acteur «Professeur»

c. Diagramme de cas d'utilisation d'acteur «Administration»

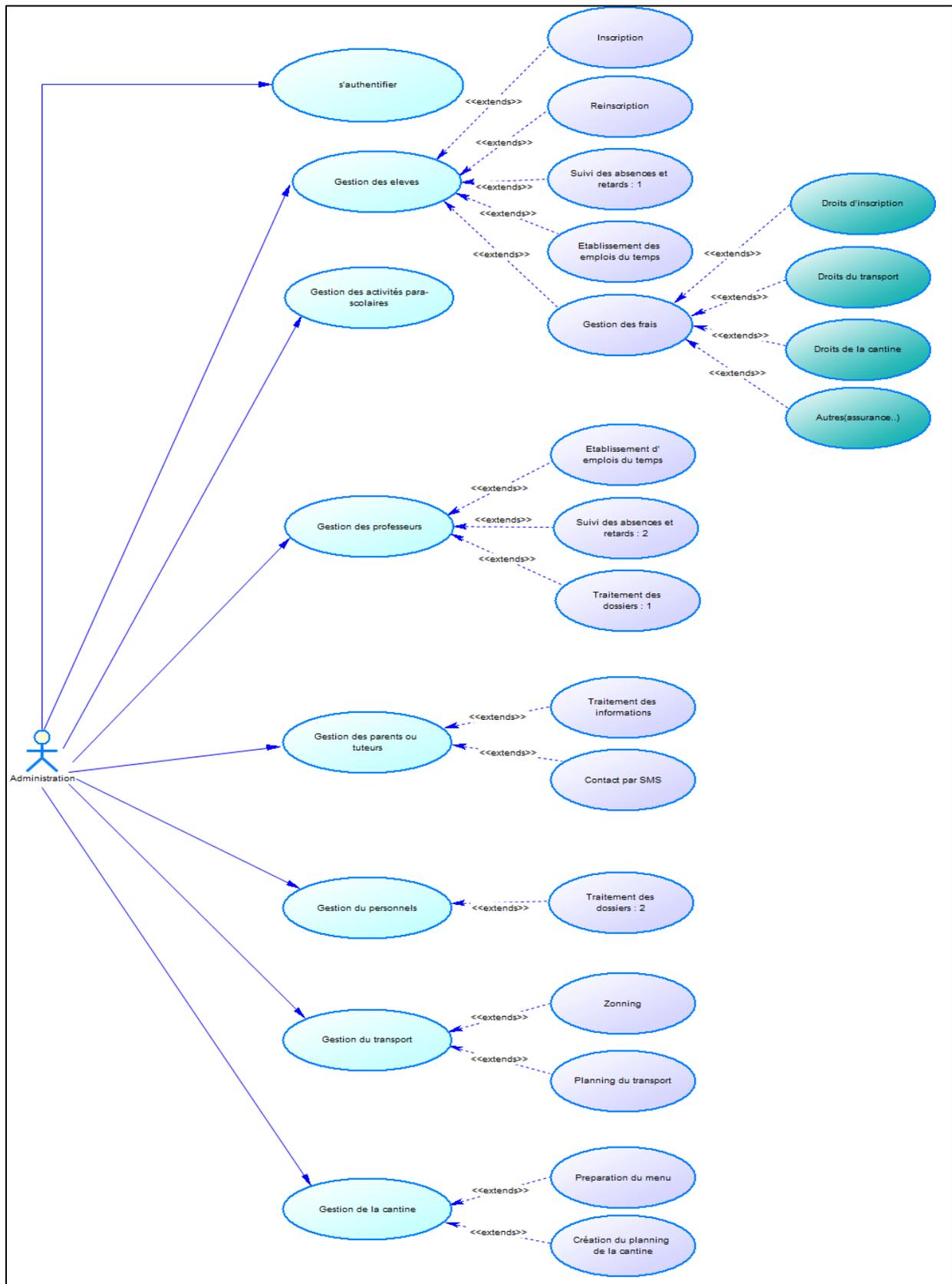


Figure 10 : Diagramme de cas d'utilisation d'acteur «Administration»

Dans notre cas, les trois acteurs (élèves, professeur, administration) possèdent une association commune « consulter le calendrier des évaluations ».

Afin d'éviter de surcharger le diagramme, il est possible de définir une relation d'héritage entre chacun de ces acteurs avec un acteur «utilisateur », qui possède l'association commune.

d. Diagramme de cas d'utilisation «Héritage »

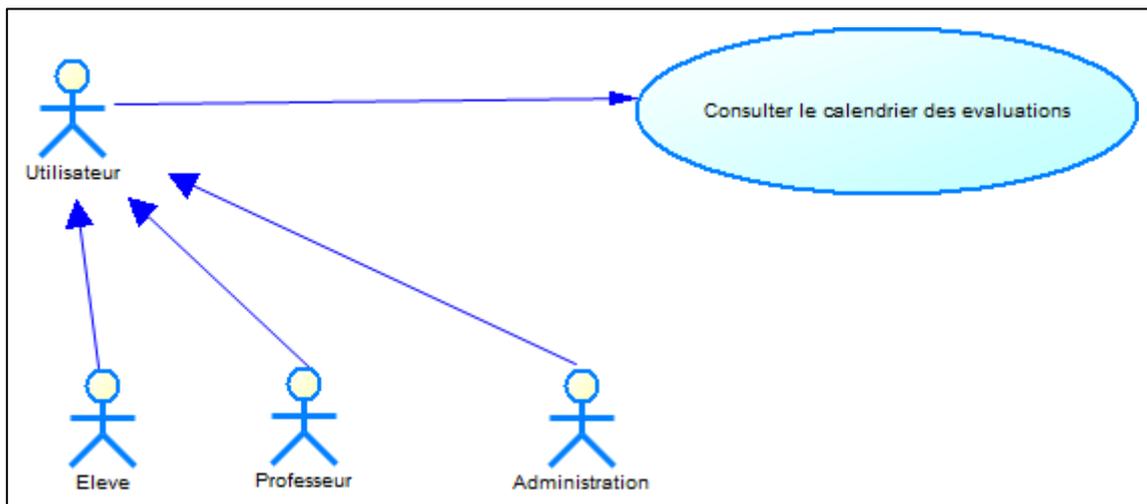


Figure 11 : Diagramme de cas d'utilisation «Héritage »

6. Diagramme de séquence

Pour mieux concrétiser les interactions entre les acteurs du système vis-à-vis de l'application et la base de données, nous sommes amenés à traduire nos scénarios en diagrammes de séquence.

Les diagrammes de séquence servent à illustrer les cas d'utilisation. Ils permettent de représenter des collaborations entre les objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois des messages.

- **Fiche de description « Authentification » :**

Sommaire d'identification :

- **Titre :** Authentification.
- **Objectif :** Permettre à chaque utilisateur de l'application, de s'authentifier vers ses rôles, à travers un login et mot de passe.
- **Acteurs :** utilisateur – système.

Description des enchainements :

- **Enchainements :**
 - **Scenario :**
 1. Saisie le login et mot de passe.
 2. Envoyer login et mot de passe.
 3. Traitement des informations envoyées.
 4. En cas d'erreur, l'authentification est rejetée.
 5. Si non, redirection d'utilisateur vers ses rôles.

Tableau 5 : Description textuel d'authentification

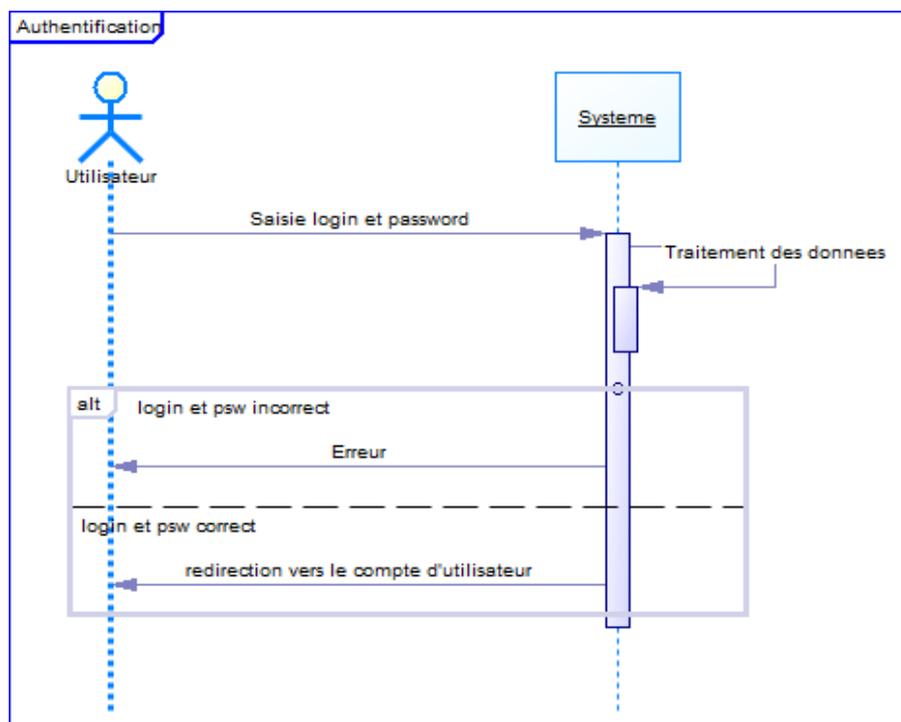


Figure 12 : Diagramme de séquence authentification

- **Fiche de description « Inscription d'un élève » :**

Sommaire d'identification :

- **Titre :** Inscription d'un élève.
- **Objectif :** Permettre à l'administration d'ajouter toutes les informations concernant un élève, y compris l'état civil, les informations sur l'ancienne école, les informations des parents ou tuteurs, les informations sur la santé, les informations complémentaires, et enfin l'affectation d'élève dans une classe.
- **Acteurs :** utilisateur – système.

Description des enchainements :

- **Pré conditions :**
 - Authentification.
 - L'inscription d'un élève doit répondre aux conditions d'inscription établie par la direction régionale.
- **Enchainements :**
 - **Scenario :**
 1. Saisie les informations de l'élève.
 2. Contrôle des données en temps réel (matricule – cne – cin) en cas de duplication.
 3. Validation de la saisie.
 4. Traitement des informations envoyé.
 5. En cas d'une anomalie, l'inscription est rejetée on précisant l'erreur effectuée.
 6. Si non, l'inscription est effectuée avec succès avec redirection d'utilisateur vers la liste d'élèves.

Tableau 6 : Description textuel d'inscription d'un élève

a. Diagramme de séquence « d'inscription d'un élève »

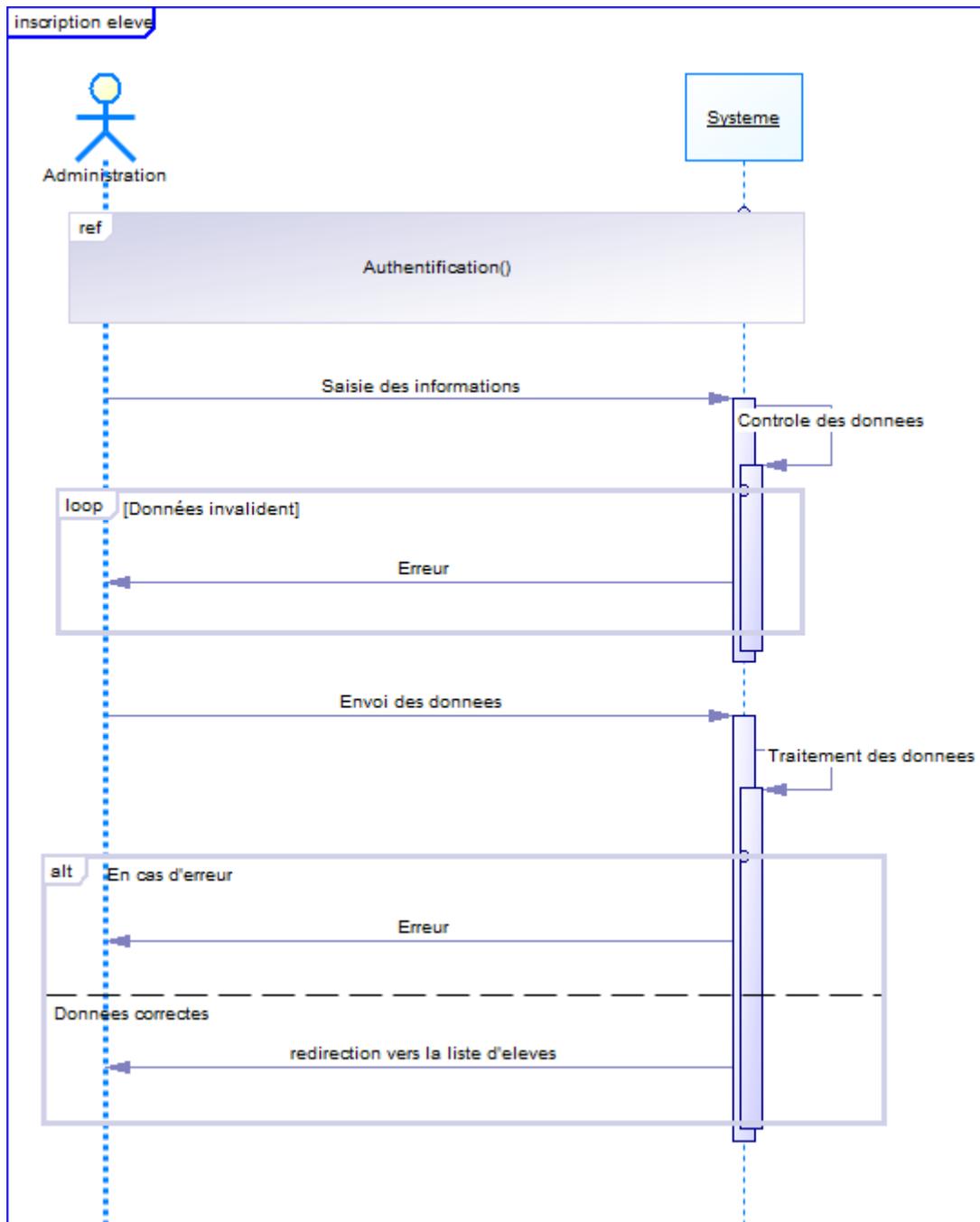


Figure 13 : Diagramme de séquence d'inscription d'un élève

- **Fiche de description « Ajouter professeur » :**

Sommaire d'identification :

- **Titre :** Ajouter professeur.
- **Objectif :** Permettre à l'administration d'ajouter toutes les informations concernant le professeur, y compris les informations personnelles, et la possibilité d'ajouter les diplômes obtenus par le professeur.
- **Acteurs :** utilisateur – système.

Description des enchainements :

- **Pré conditions :**
 - Authentification.
 - L'ajout d'un professeur doit répondre aux conditions de recrutement établies par la direction régionale.
- **Enchainements :**
 - **Scenario :**
 1. Saisie des informations concernant le professeur.
 2. Contrôle des données en temps réel, en cas de duplication.
 3. Validation de la saisie.
 4. Traitement des informations envoyées.
 5. En cas d'une anomalie, l'ajout est rejeté en précisant l'erreur effectuée.
 6. Si non, l'ajout est effectué avec succès avec redirection d'utilisateur vers la liste des professeurs.

Tableau 7 : Description textuel d'ajout d'un professeur

b. Diagramme de séquence « d'ajout d'un professeur »

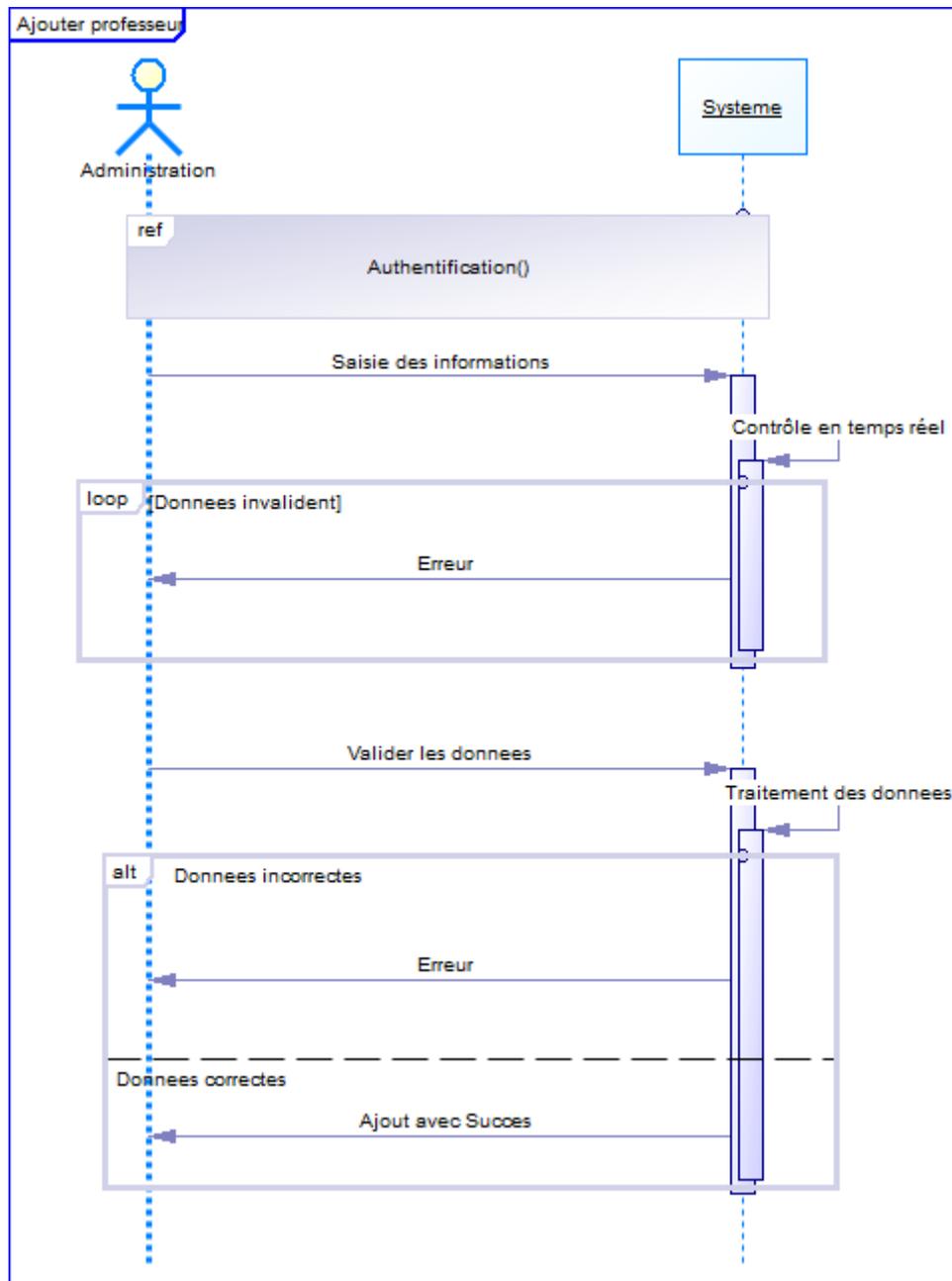


Figure 14 : Diagramme de séquence d'ajout d'un professeur

- **Fiche de description « Affectation des disciplines aux niveaux » :**

Sommaire d'identification :

- **Titre :** Affectation des disciplines aux niveaux.
- **Objectif :** Permettre à l'administration d'affecter à chaque niveau ses disciplines, en précisant le coefficient de chaque discipline.
- **Acteurs :** utilisateur – système.

Description des enchainements :

- **Pré conditions :**
 - Authentification.
- **Enchainements :**
 - **Scenario :**
 1. Choisir le niveau.
 2. Choisir la matière, ce qui vous donne les disciplines.
 3. Choisir la discipline et attribuer un coefficient.
 4. Valider les choix.
 5. En cas d'une erreur, l'ajout est rejeté en précisant l'erreur effectuée.
 6. Si non, l'ajout est effectué avec succès.

Tableau 8 : Description textuel d'affectation des disciplines aux niveaux

c. Diagramme de séquence « d'affectation des disciplines aux niveaux »

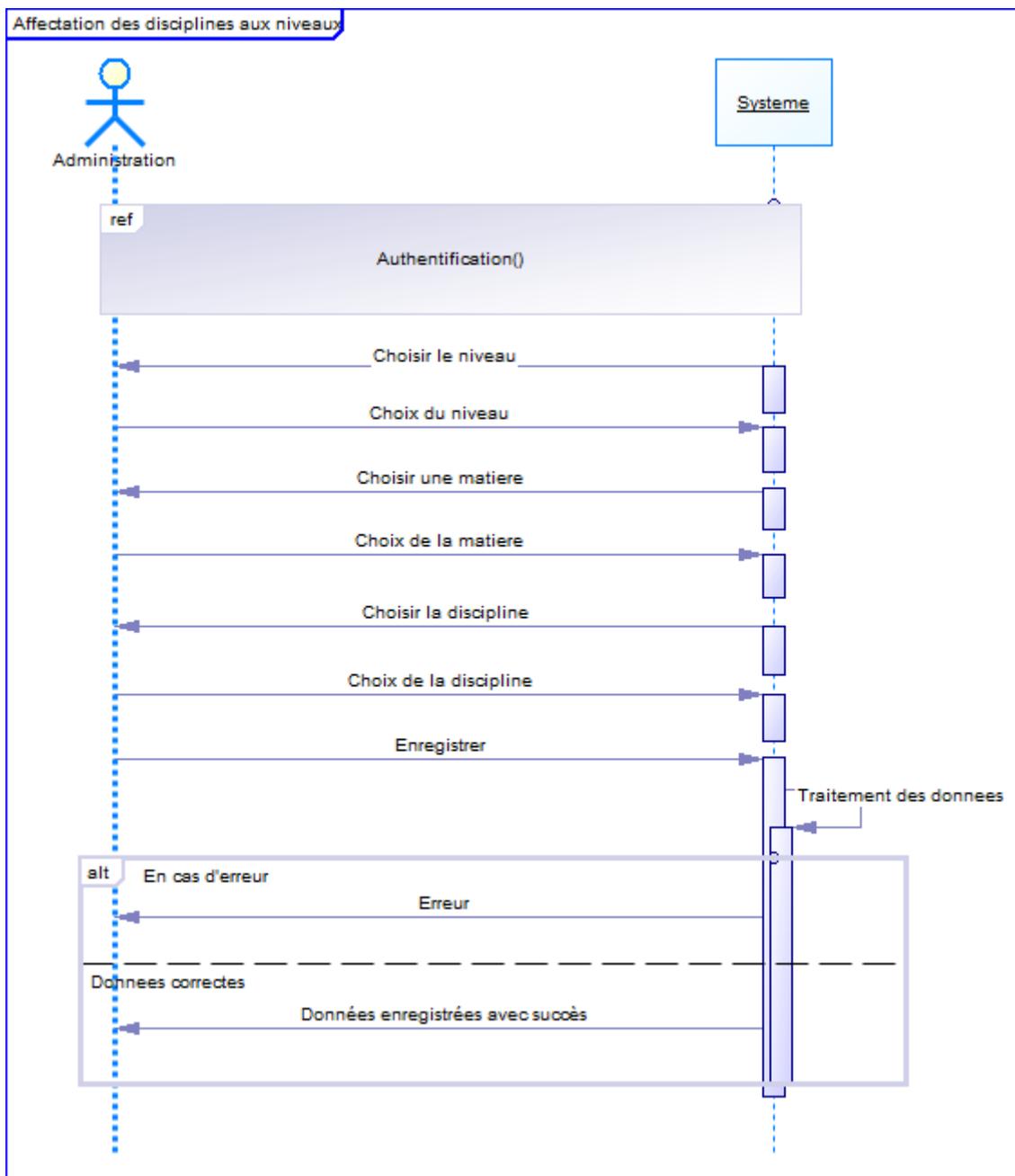


Figure 15 : Diagramme de séquence d'affectation des disciplines aux niveaux

- **Fiche de description « Ajouter discipline » :**

Sommaire d'identification :

- **Titre :** Ajouter discipline.
- **Objectif :** Permettre à l'administration un ajout rapide des disciplines de chaque matière.
- **Acteurs :** utilisateur – système.

Description des enchainements :

- **Pré conditions :**
 - Authentification.
- **Enchainements :**
 - **Scenario :**
 1. En cas d'existence de la matière :
 - a) Sélectionner la matière.
 - b) Saisie des informations des disciplines.
 - c) Validation d'ajout.
 - d) Traitement des informations.
 - e) En cas d'erreur, l'ajout est annulé.
 - f) Si non, l'ajout est effectué avec succès avec redirection d'utilisateur vers la liste des disciplines.
 2. En cas d'absence de la matière :
 - a) Saisie des informations concernant la matière.
 - b) Saisie des informations concernant les disciplines.
 - c) Validation d'ajout.
 - d) Traitement des informations.
 - e) En cas d'erreur, l'ajout est annulé.
 - f) Si non, l'ajout est effectué avec succès avec redirection d'utilisateur vers la liste des disciplines.

Tableau 9 : Description textuel d'ajout d'une discipline

d. Diagramme de séquence « d'ajout des disciplines »

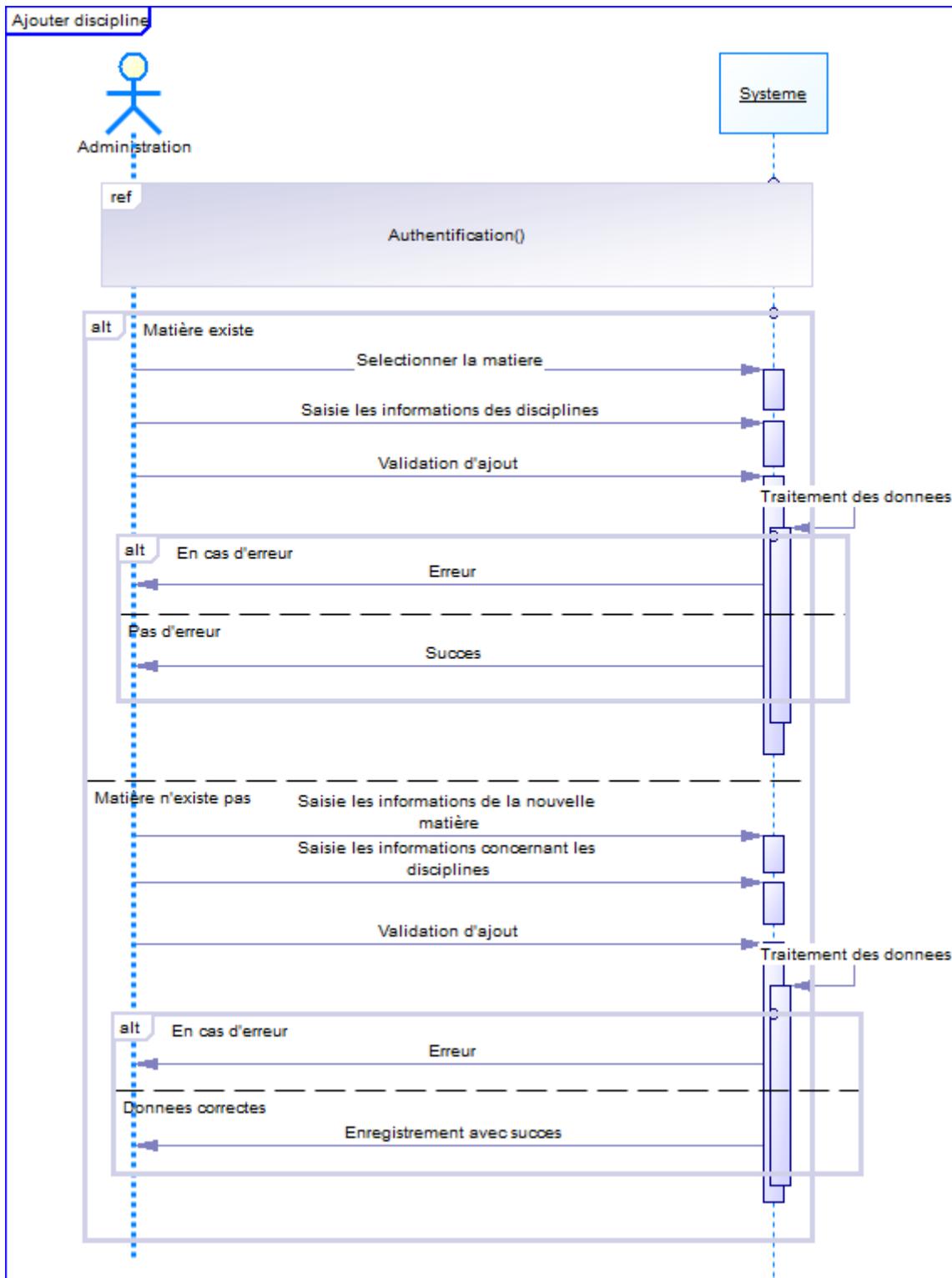


Figure 16 : Diagramme de séquence d'ajout des disciplines

7. Diagramme d'activités

Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation. De plus, leur représentation sous forme d'organigrammes les rend facilement intelligibles et beaucoup plus accessibles que les diagrammes d'états-transitions. On parle généralement dans ce cas de modélisation de workflow. On se concentre ici sur les activités telles que les voient les acteurs qui collaborent avec le système dans le cadre d'un processus métier.

- **Nœuds d'activités :**

De la gauche vers la droite, on trouve : le nœud représentant une action, qui est une variété de nœud exécutable, un nœud objet, un nœud de décision ou de fusion, un nœud de bifurcation ou d'union, un nœud initial, un nœud final et un nœud final de flot.

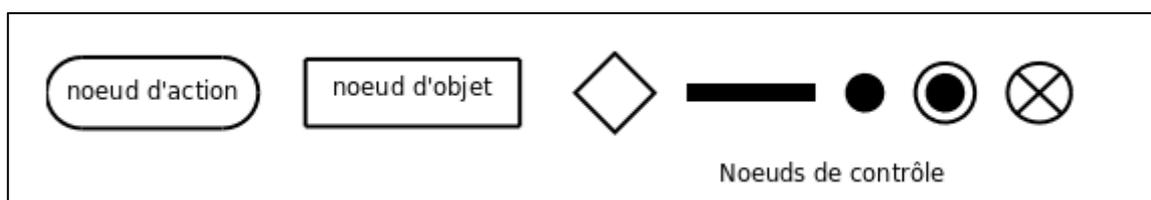


Figure 17 : Représentation graphique des nœuds d'activité

- **Description du processus de diagramme d'activités «Inscription d'un élève» :**

- L'élève demande l'inscription dans un niveau.
- L'administration vérifie les conditions d'inscriptions pour l'élève.
- Si l'élève ne répond pas aux conditions de l'établissement, donc la demande est refusée.
- Si non, l'élève doit fournir les pièces et les informations nécessaires pour l'inscription.
- L'administration donne les informations personnelles de l'élève.
- L'administration introduit les informations complémentaires et celles concernant la santé de l'élève.

- L'administration affecte le niveau et valide l'inscription.
- Le système traite les informations envoyées.
- En cas d'une anomalie, le système refuse l'inscription demandant à l'administration de vérifier l'anomalie.
- Si non, l'inscription est effectuée avec succès.

a. Diagramme d'activités « d'inscription d'un élève »

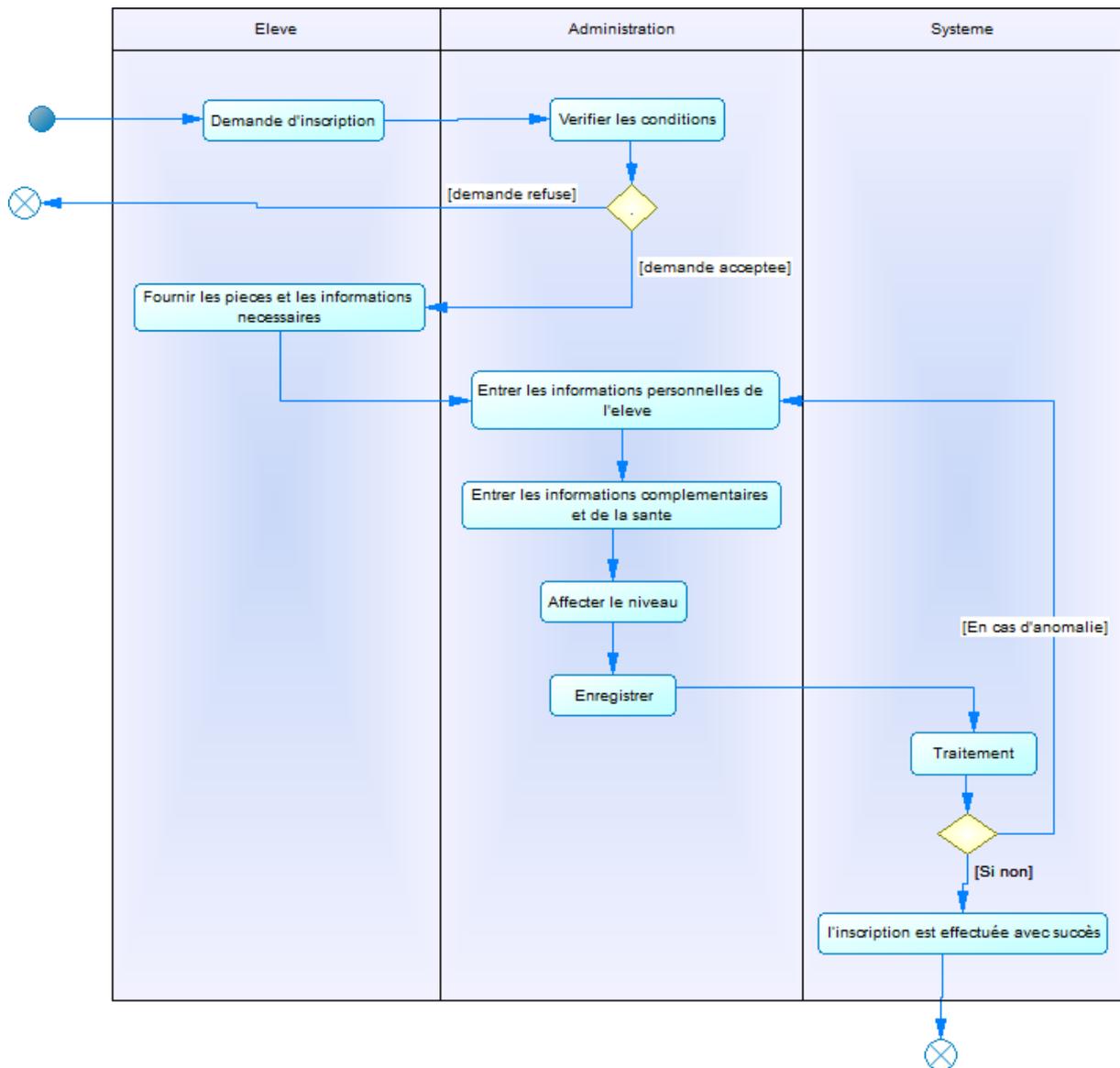


Figure 18 : Diagramme d'activités d'inscription d'un élève

- **Description du processus de diagramme d'activités «Affecter les notes d'élèves» :**

- L'administration demande la liste des évaluations.
- Le système affiche la liste des évaluations.
- En cas d'absence d'évaluation, l'administration doit créer l'évaluation.
- Si non, l'administration choisit l'évaluation et affecte les notes pour chaque élève.
- Puis envoyer les notes pour les sauvegarder.
- Le système traite les informations envoyées.
- En cas, d'une anomalie l'ajout est annulé.
- Si non, l'ajout est effectué avec succès.

b. Diagramme d'activités « d'affectation des notes d'élèves »

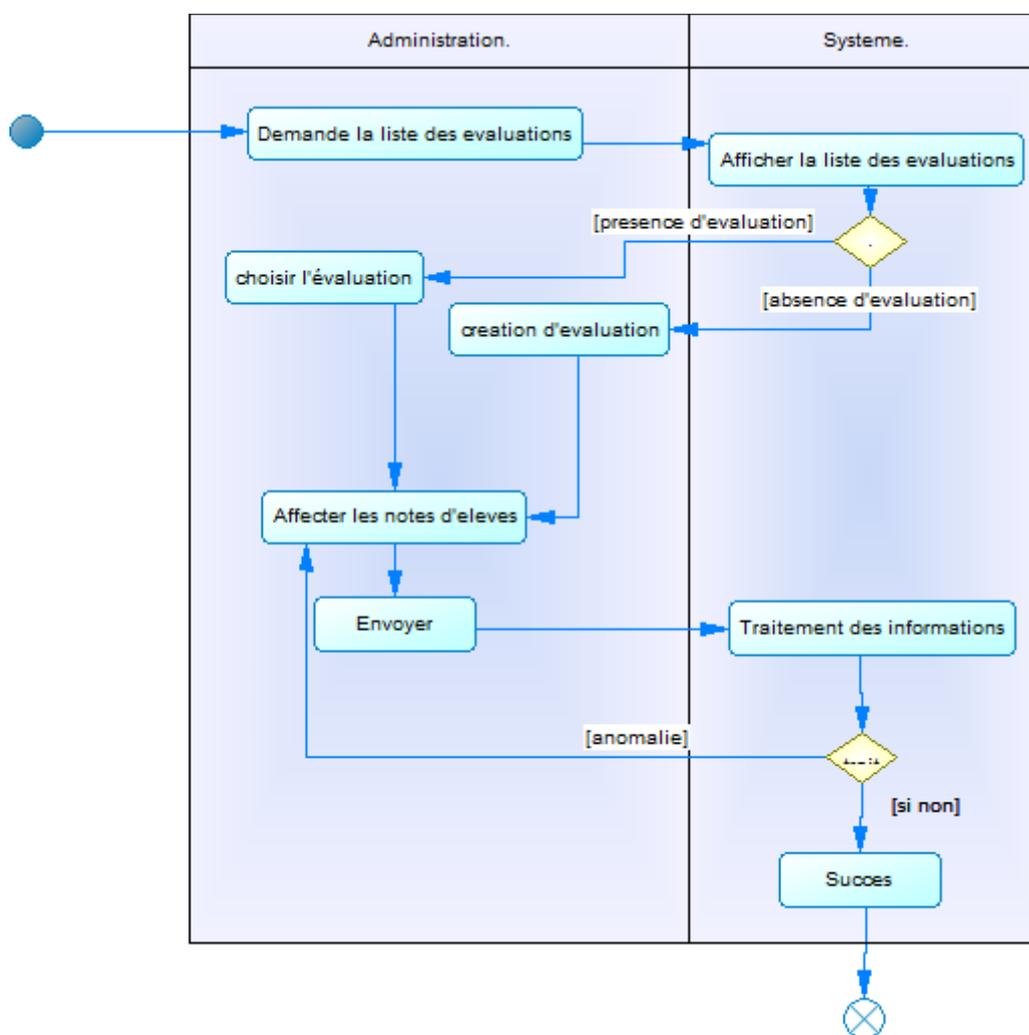


Figure 19 : Diagramme d'activités d'affectation des notes d'élèves

• **Description du processus de diagramme d'activités «création d'emploi du temps»**

- L'administration choisit la classe voulue.
- Puis choisir l'année scolaire et la période, et puis cliquer sur valider.
- Le système affiche l'agenda de la semaine du lundi jusqu'au samedi.
- L'administration renseigne les champs de l'agenda.
- Le système contrôle les informations en temps réel.
- En cas d'une erreur, le système demande à l'utilisateur de corriger l'erreur.
- Si non, l'administration envoie les données et le système enregistre l'agenda.

c. Diagramme d'activités « création d'emploi du temps »

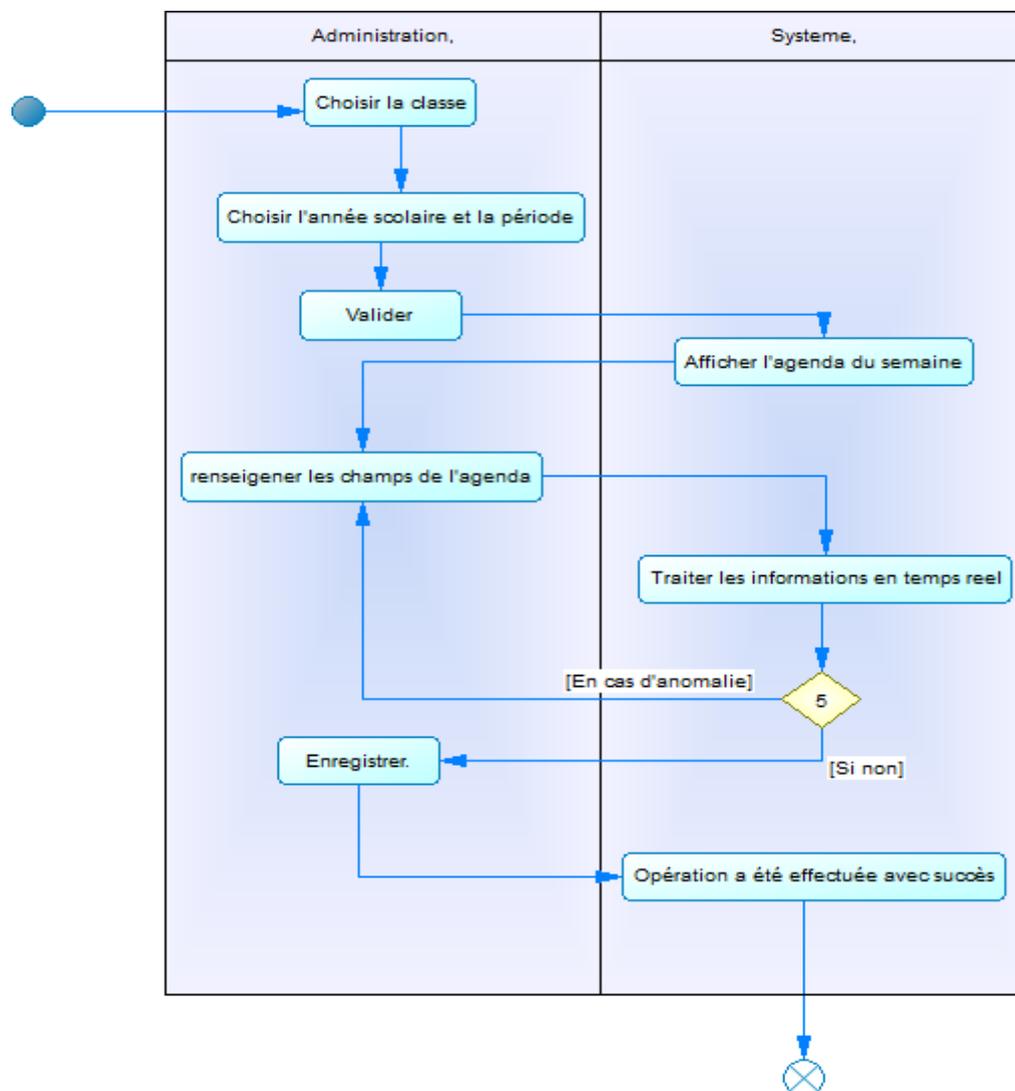


Figure 20 : Diagramme d'activités création d'emploi du temps

8. Diagramme de collaboration

Un diagramme de collaboration est un diagramme d'interactions, représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets, et où la chronologie n'intervient que de façon annexe.

Cela consiste en un graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) et les échanges entre ces objets.

- **Diagramme de collaboration «réinscription d'un élève»**

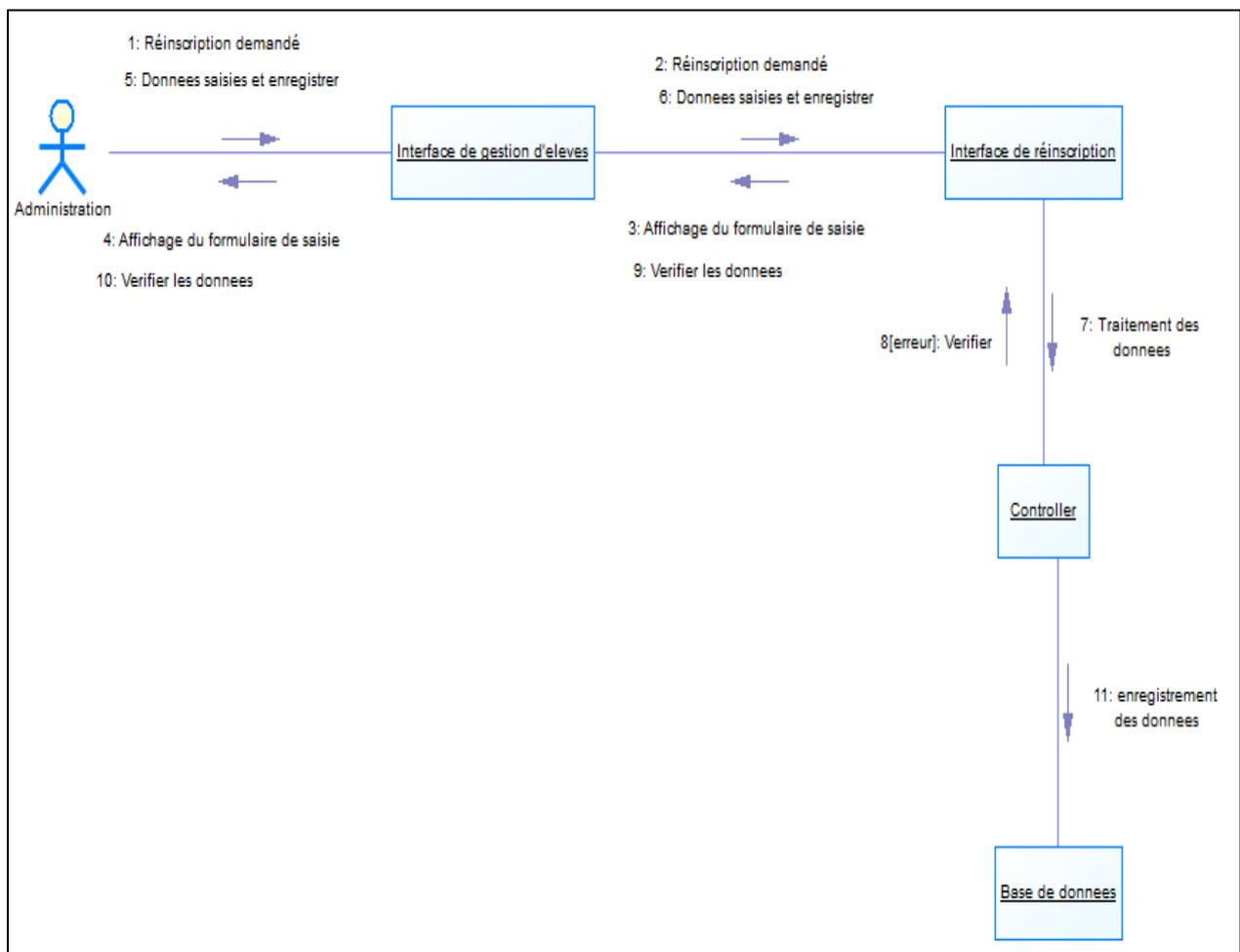


Figure 21 : Diagramme de collaboration de réinscription d'un élève

9. Diagramme de package

L'utilisation la plus courante pour diagrammes de paquetages est d'organiser des Diagrammes de Cas d'Utilisation et des Diagrammes de Classes.

Le diagramme de package peut afficher à la fois la structure et les dépendances entre les sous-systèmes ou les modules.

- **Diagramme de package du système :**

Le diagramme de packages, appelé aussi diagrammes de paquetages ; permet de découper l'ensemble de classes (d'un modèle de classes) en sous-ensemble de classes en fonction de leurs dépendances, d'une tâche commune. En clair, on essaye de regrouper toutes les classes qui sont fortement dépendantes entre-elles avec le reste des classes. Par exemple toutes les classes qui servent aux techniques de marketing, toutes les classes qui servent au fonctionnement des servlets.

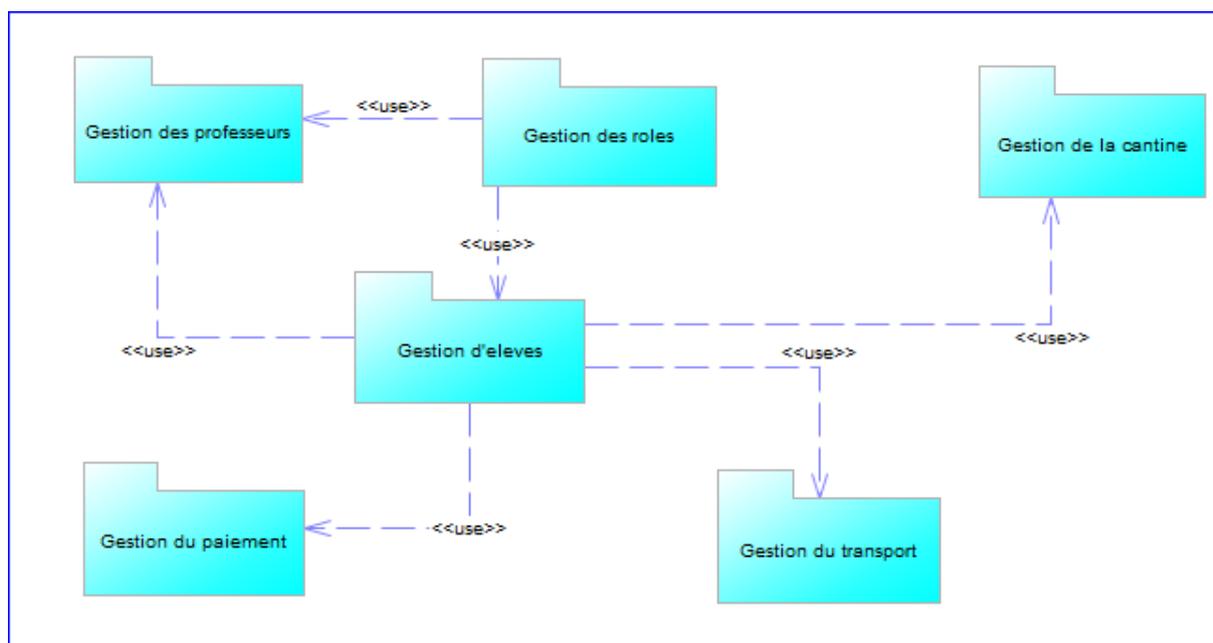


Figure 22 : Diagramme de package

10. Diagramme de classes

Le diagramme de classe est une description statique du système focalisé sur le concept de classe et d'association. Une classe représente un ensemble d'objets qui possèdent des propriétés similaires et des comportements communs décrivant en terme d'attributs et d'opérations.

Une association consiste à présenter les liens entre les instances de classe. Durant cette section, nous allons présenter les diagrammes de classes entités à notre application.

a. Diagramme de classe « gestion d'élève »

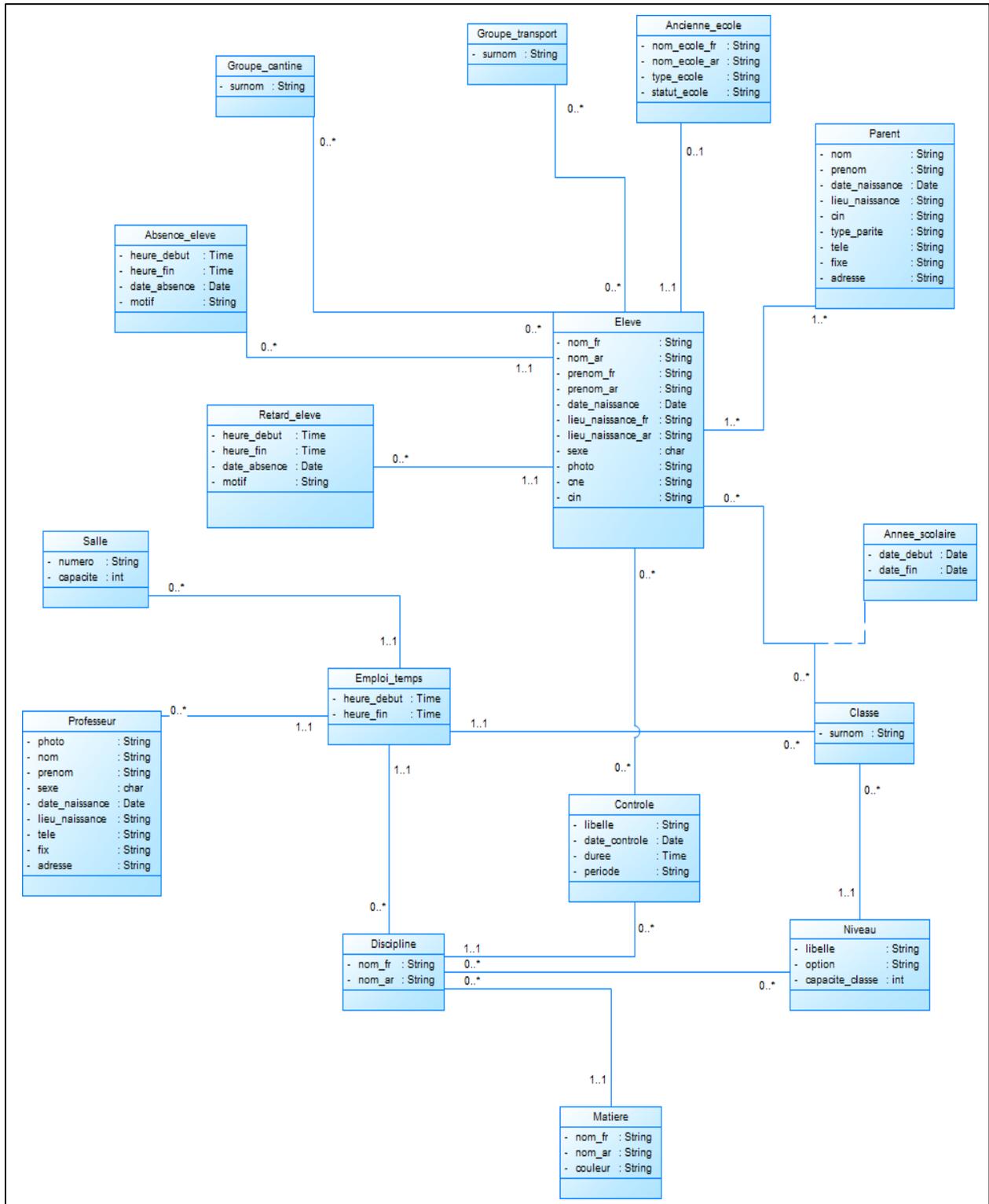


Figure 23 : Diagramme de classe « gestion d'élève »

b. Diagramme de classe « gestion des professeurs »

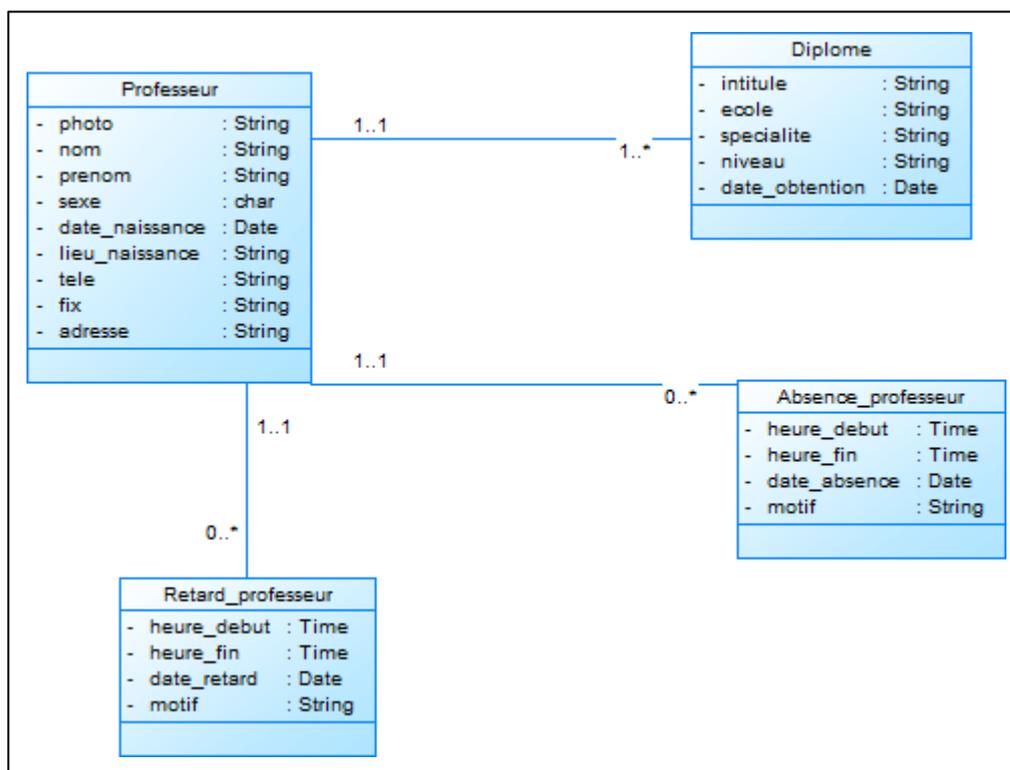


Figure 24 : Diagramme de classe « gestion des professeurs »

c. Diagramme de classe « gestion de la cantine »

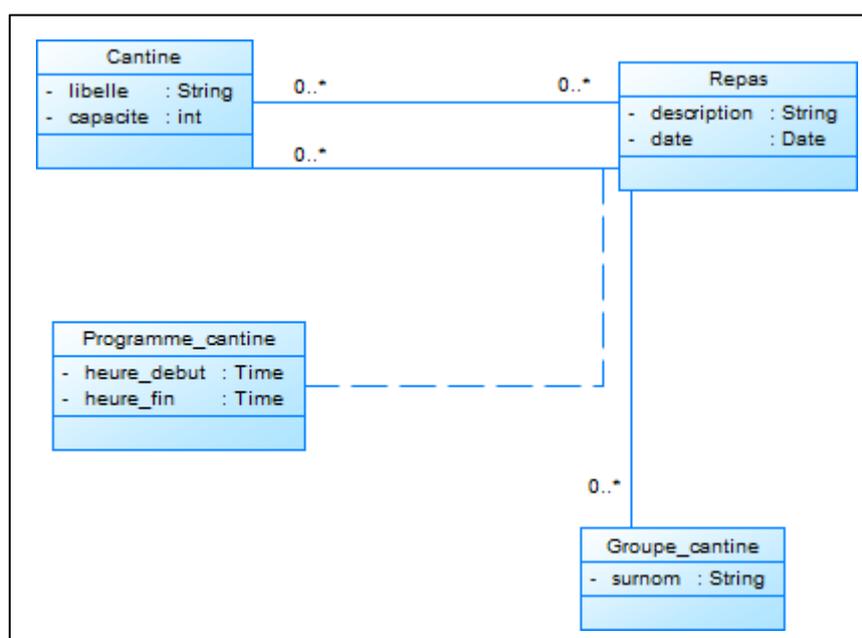


Figure 25 : Diagramme de classe « gestion de la cantine »

d. Diagramme de classe « gestion du transport »

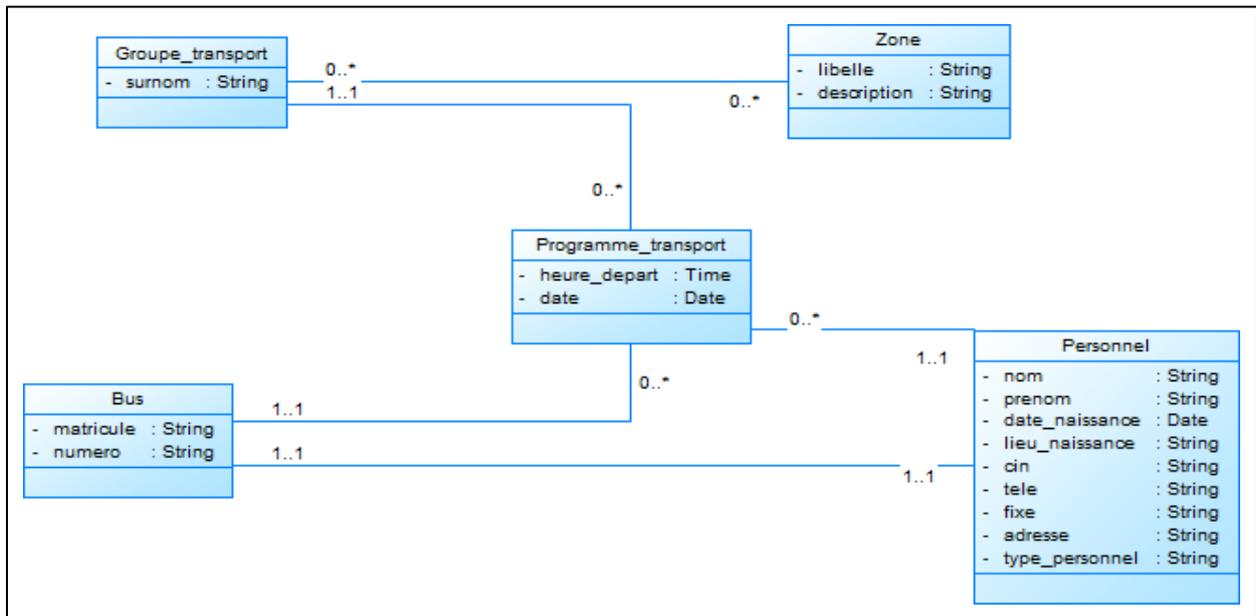


Figure 26 : Diagramme de classe « gestion du transport »

e. Diagramme de classe « gestion des paiements d'élèves »

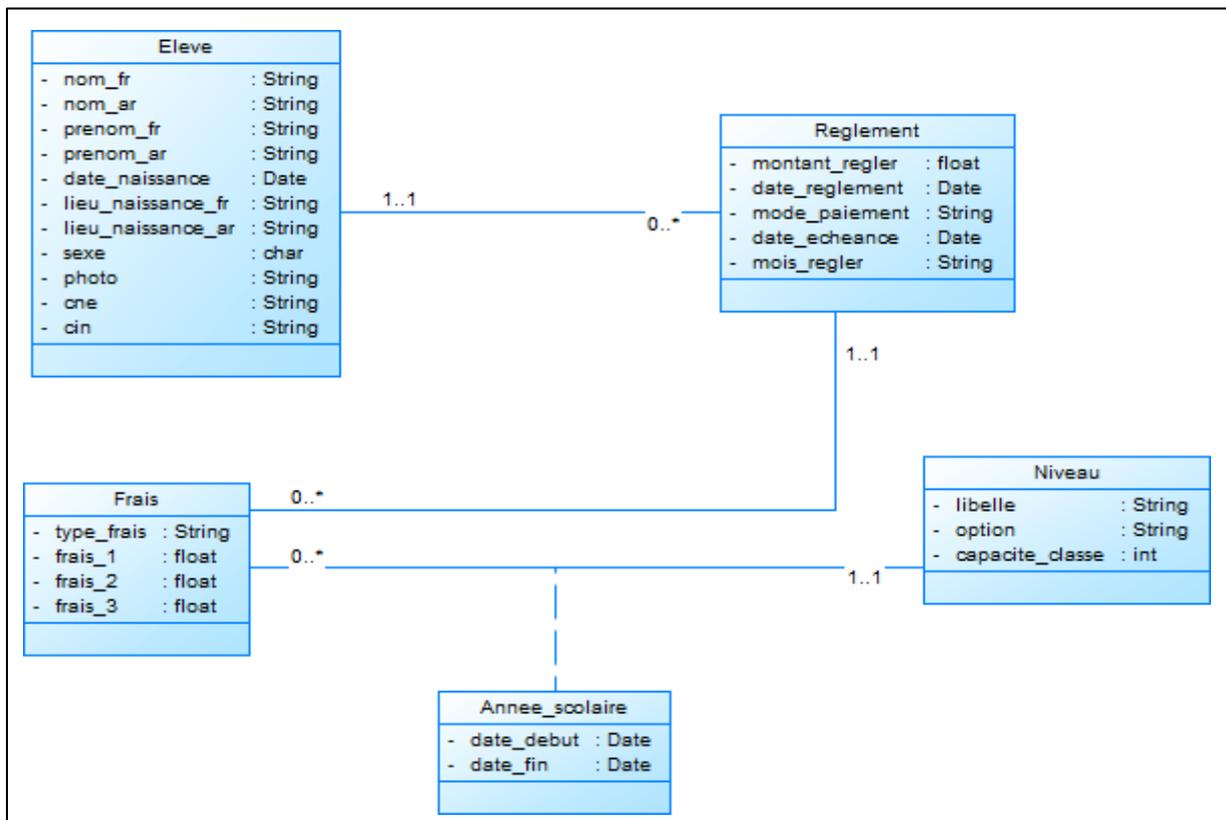


Figure 27 : Diagramme de classe « gestion du paiements d'élèves »

f. Diagramme de classe « gestion des rôles »

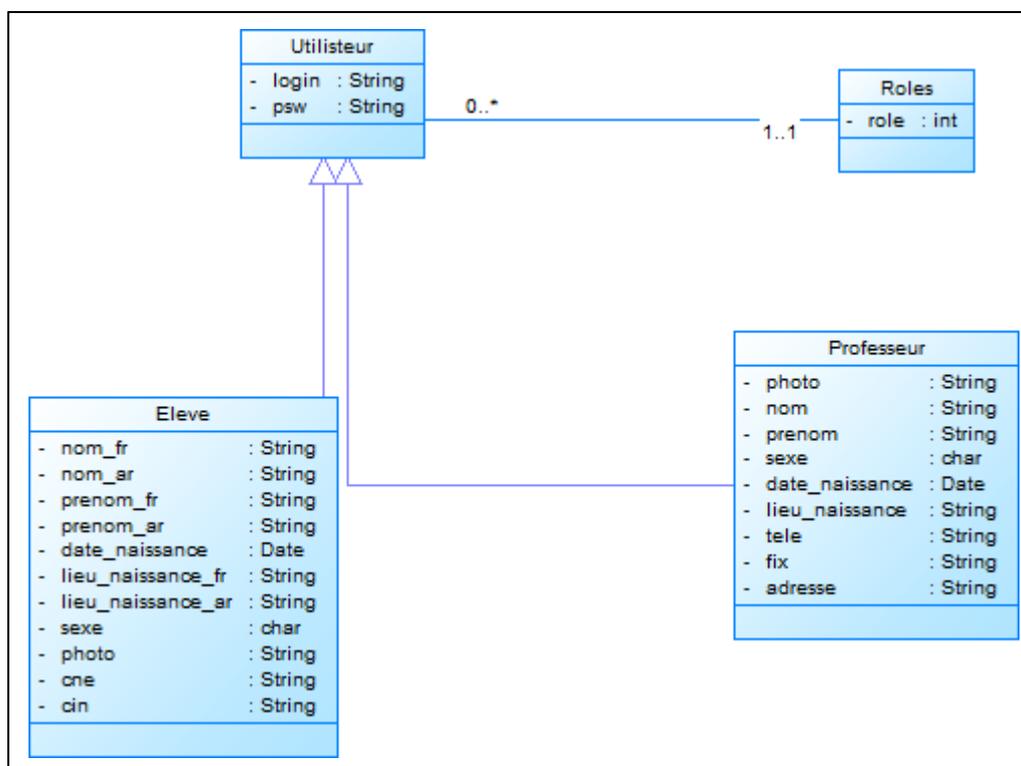


Figure 28 : Diagramme de classe « gestion des rôles »

11. Conclusion

Dans ce chapitre, nous avons conçu et documenté le code que nous devons produire. Dans cette phase, toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (ensemble de diagrammes) prêt à être codé. Dans le chapitre suivant nous allons étudier en détails les outils et les langages utilisés durant la phase de construction.

Chapitre 5 : Réalisation

1. Introduction

Après avoir achevé l'étape d'analyse et conception de l'application, on va entamer dans ce chapitre la partie réalisation et implémentation dans laquelle on s'assure que le système est prêt pour être exploité par les utilisateurs finaux.

2. Environnement et outils de développement

a. Environnement de développement

- **Eclipse :**

Eclipse est un environnement de développement intégré (IDE) utilisé dans la programmation informatique et l'IDE Java le plus utilisé. Il contient un espace de travail de base et un système extensible de plug-in pour personnaliser l'environnement. Eclipse est écrit principalement en Java et son utilisation principale est pour le développement d'applications Java.

- **Apache Tomcat :**

Apache Tomcat, souvent appelé **Tomcat Server**, est un conteneur de servlets Java open source développé par Apache Software Foundation (ASF). Tomcat implémente plusieurs spécifications Java EE, y compris Java Servlet, JavaServer Pages (JSP) et WebSocket, et fournit un environnement de serveur web http « Java pur » dans lequel le code Java peut s'exécuter.

- **Système de gestion de base de données « MySQL » :**

MySQL est un système de gestion de bases de données relationnelles (SGBDR) fonctionnant sous Windows et Linux. Il fait partie des logiciels de gestion de base des données les plus utilisées au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

- **Maven :**

Maven est un outil de construction de projets (build) open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java.

Il permet notamment :

- d'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet.
- de gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet.
- de générer des documentations concernant le projet.

- **Twilio api :**

Twilio (prononcé TWILL-eo) est une plate-forme de communication en nuage comme une société de service, basée à San Francisco, en Californie. Twilio permet aux développeurs de logiciels de programmer et de recevoir des appels téléphoniques et d'envoyer et de recevoir des messages texte en utilisant ses API de service Web. Les services de Twilio sont accessibles via HTTP et sont facturés en fonction de leur utilisation.

Le but de l'amélioration de la communication sur une équipe est de s'assurer qu'aucune information ou travail n'est perdu lorsque les tâches sont transférées d'un membre de l'équipe à l'autre. Le concept central lorsque vous travaillez avec Team Foundation est le projet d'équipe. Un projet d'équipe est stocké sur Team Foundation Server et possède un nom que tout le monde peut identifier facilement.

- **Team Foundation Server :**

TFS signifie Team Foundation Server développé par Microsoft. TFS permet à une équipe de travailler ensemble et d'organiser ses efforts pour mener à bien un projet. Les développeurs utilisent TFS pour le contrôle des sources, le suivi des bogues, la collecte des besoins et pour gérer le cycle de vie complet du développement de logiciels.

b. Technologies et Framework adoptées

- **Plateforme Java EE :**

Le terme « **Java EE** » signifie Java Entreprise Edition, et était anciennement raccourci en « J2EE ». Il fait quant à lui référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE construite sur le langage Java et la plateforme Java SE, et elle ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine.

L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées déployées et exécutées sur un serveur d'applications.

- **Framework spring :**

SPRING est effectivement un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'application J2EE. Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ces objets.

Le gros avantage par rapport aux serveurs d'application est qu'avec **SPRING**, vos classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le Framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que **SPRING** est qualifié de conteneur « léger ».

Le cœur de **SPRING** et ce qui fait sa très grande force est la mise en œuvre du design pattern « Inversion Of Control » ou encore « Dependency Injection », et c'est ça mon premier but d'utilisation de ce Framework dans mon projet.

- **Framework spring security :**

Spring Security est un Framework d'authentification puissante et hautement personnalisable et d'un cadre de contrôle d'accès. Il est le standard pour la sécurisation des applications web.

Spring Security est l'un des projets de spring les plus matures et largement utilisé. Fondée en 2003 et activement maintenu par SpringSource, il est aujourd'hui utilisé pour sécuriser de nombreux environnements les plus exigeants, y compris les agences gouvernementales, les applications militaires et les banques centrales. Il est distribué sous licence Apache 2.0.

- **Framework hibernate/jpa :**

Les applications d'entreprises s'orientent de plus en plus vers des architectures n-tiers. La technologie J2EE et les Framework qui se sont offerts beaucoup d'outils pour répondre aux besoins modernes. Pour la couche présentation, on trouve par exemple, le plus populaire **Struts** respectant le modèle MVC de séparation de code, de l'interface et des données.

Nous allons, cependant, nous intéresser à une couche plus basse d'une architecture applicative, la couche d'accès aux données. Celle-ci permet d'interfacer le code métier avec une source des données. L'intérêt est de pouvoir changer de base de données en n'ayant besoin de ne modifier que la couche d'accès.

Pour réaliser cette tâche, il existe plusieurs solutions fournies par J2EE qui sont appelées **JPA** (suivre l'annexe) (Java Persistence API). Ce mécanisme qui gère la correspondance entre des objets d'une application et les tables de base des données se nomme ORM (Object-Relationnal Mapping).

Pour cela, on a pris la décision d'utiliser **Hibernate** comme un **Framework ORM**.

On a pris le choix d'utiliser Hibernate car :

- Génère le code SQL nécessaire, ce qui rend l'application plus portable.
- La persistance est transparente.
- La récupération des données est optimisée.
- Portabilité du code en cas de changement de la base des données.

- **Framework bootstrap :**

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications 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.

- **JSTL :**

JSTL est l'acronyme de Java server page Standard Tag Library. C'est un ensemble de tags personnalisés développé sous la JSR 052 qui propose des fonctionnalités souvent rencontrées dans les JSP.

Elle étend la spécification JSP en ajoutant une bibliothèque de balises pour les tâches courantes, comme le travail sur des fichiers XML, l'exécution conditionnelle, les boucles et l'internationalisation.

- **JQUERY :**

JQuery est un Framework développé en JavaScript qui permet notamment de manipuler aisément la DOM, d'utiliser AJAX, de créer des animations..., La vocation première de ce Framework est de gagner du temps dans le développement des applications: « write less, do more ».

- **AJAX :**

AJAX (Asynchronous Javascript And XML, traduisez Javascript asynchrone et XML) est une méthode de développement web basée sur l'utilisation d'un script Javascript pour effectuer des requêtes web à l'intérieur d'une page web sans recharger la page. AJAX rend plus interactifs les applications web et offre une meilleure ergonomie ainsi qu'une réactivité améliorée en permettant de modifier interactivement une partie de l'interface web seulement.

En effet, le modèle web traditionnel est basé sur une suite de requêtes et de réponses successives, c'est-à-dire une navigation séquentielle de page web en page web.

AJAX permet de ne modifier que la partie de la page web qui nécessite d'être mise à jour en créant une requête HTTP locale et en modifiant tout ou partie de la page web en fonction de la requête HTTP récupérée.

- **XML :**

XML est un langage de balisage qui définit un ensemble de règles pour l'encodage des documents dans un format de qui est à la fois lisible par l'homme et lisible par machine.

- **JSON :**

JSON (JavaScript Object Notation) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple.

Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci.

3. Architecture générale de l'application

a. Architecture logiciel du système

Dans ce projet j'ai fait appel aux Frameworks Hibernate, Spring MVC et Spring Security. L'un des intérêts de l'intégration de ces différents Frameworks et de permettre la mise en place d'une architecture rigoureuse, de manière à garantir la maintenabilité, l'évolutivité et l'exploitabilité de l'application.

La figure ci-dessous montre l'architecture qui a été mise en place dans le cadre de ce projet, cette architecture est largement admise comme la plus efficace et généralisable à n'importe quel projet Web.

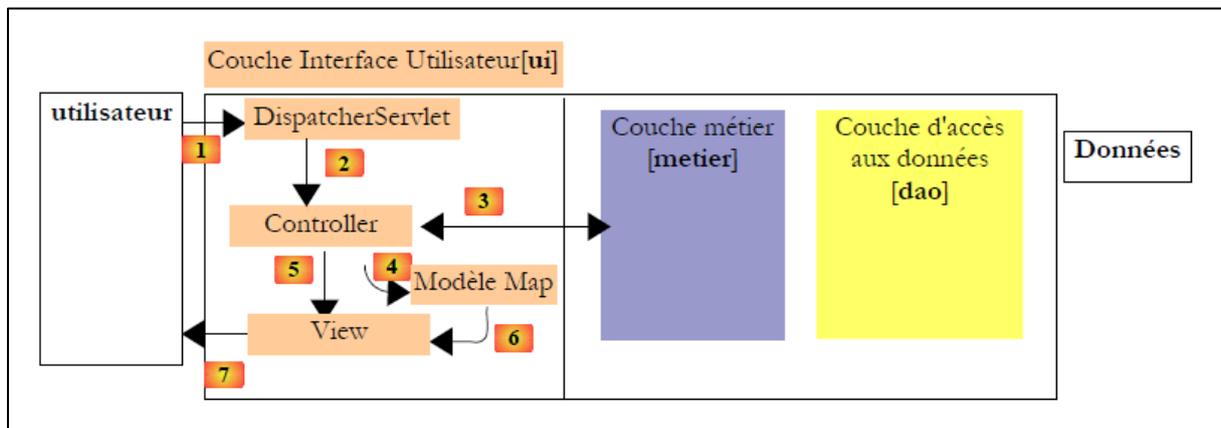


Figure 29 : Architecture du système

La principale caractéristique de cette architecture est la séparation des préoccupations (données, Service et Présentation) grâce à la séparation stricte des couches applicatives. En effet on peut observer les trois couches de l'application :

- ❖ **Couche DAO :** permet les accès à la base de données à travers le Framework Hibernate.
- ❖ **Couche Métier :** contient l'ensemble du code service de l'application, elle organise les accès à la couche DAO et ses aspects transactionnels. Ses aspects sont gérés et organisés par le Framework Spring.
- ❖ **Couche Présentation :** cette couche est la couche d'entrée dans l'application du point de vue d'utilisateur.

Elle appelle les traitements de la couche Service en fonction des actions effectuées par l'utilisateur et récupère les données retournées. Ensuite elle met en forme ces données pour l'affichage, cette couche est réalisée grâce au Framework Struts2, la technologie Ajax et la bibliothèque JQuery.

Ces trois couches sont rigoureusement séparées les unes des autres en ce sens qu'il ne doit exister idéalement aucune dépendance entre elles. Ainsi chaque couche ne connaît que les interfaces de la couche inférieure. Par exemple la couche métier ne connaît que les interfaces de la couche DAO, de même la couche Présentation ne connaît que les interfaces de la couche Métier. Ce qui assure que chaque couche publie via ses interfaces l'ensemble des traitements qu'elle met à la disposition des couches supérieures. Un couplage technologique alors mis en place grâce au lien qui existe entre chaque interface et son implémentation, ce lien est géré par le Framework Spring.

b. Le modèle MVC

Le Framework Spring Web MVC fournit une architecture MVC (Model-View-Controller) et des composants prêts à utiliser pour développer des applications Web flexibles et peu couplées. Le modèle MVC permet de séparer les différents aspects de l'application (logique d'entrée, logique métier et logique UI), tout en fournissant un couplage en vrac entre ces éléments.

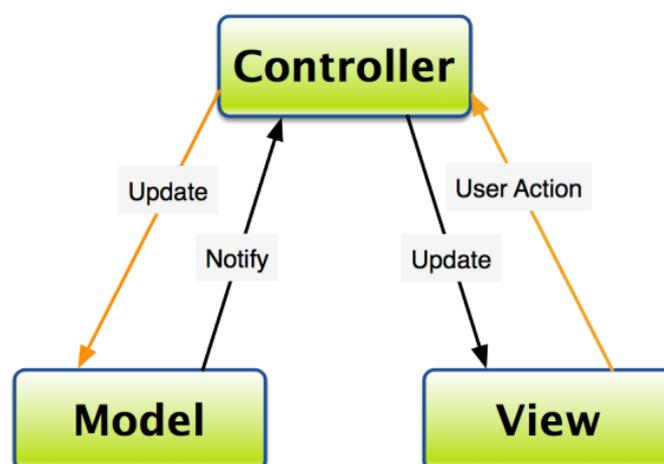


Figure 30 : Pattern MVC

- Le **modèle** encapsule les données de l'application et en général, elles sont constituées de POJO.

POJO est un acronyme qui signifie **plain old Java object**. Cet acronyme est principalement utilisé pour faire référence à la simplicité d'utilisation d'un objet Java en comparaison avec la lourdeur d'utilisation d'un composant EJB.

- La **Vue** est responsable du rendu des données du modèle et, en général, il génère une sortie HTML que le navigateur du client peut interpréter.
- Le **contrôleur** est responsable du traitement des demandes des utilisateurs et de la construction d'un modèle approprié et le transmet à la vue pour le rendu.

c. Le DispatcherServlet

La structure de Spring Web model-view-controller (MVC) est conçue autour d'un DispatcherServlet qui gère toutes les requêtes et réponses HTTP. Le processus de traitement des requêtes du Spring Web MVC DispatcherServlet est illustré dans le diagramme suivant:

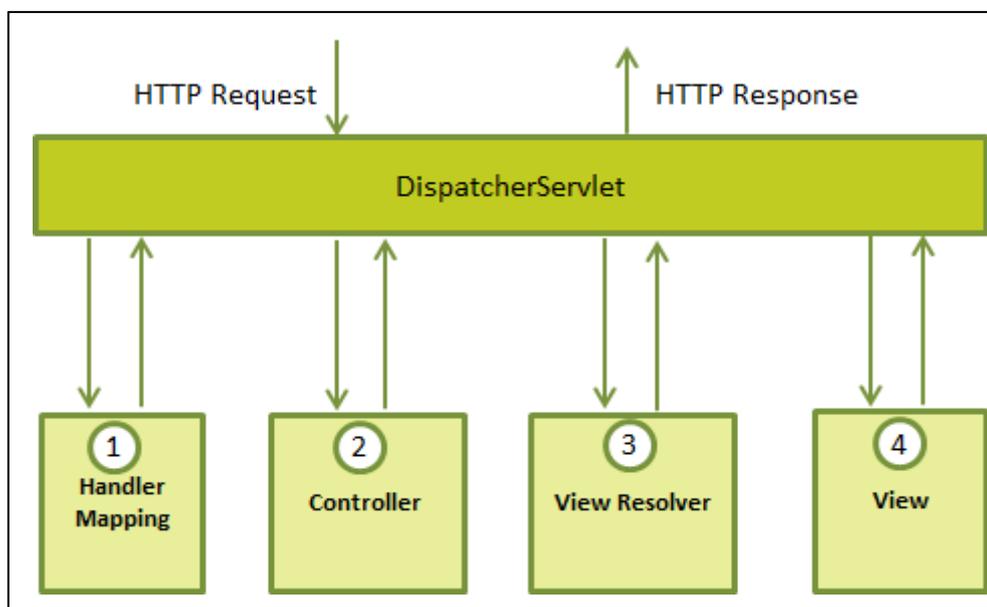


Figure 31 : Structure du spring MVC autour du DispatcherServlet

Voici la séquence d'événements correspondant à une requête HTTP entrante à DispatcherServlet :

- Après avoir reçu une requête http, DispatcherServlet consulteHandlerMapping pour appeler le contrôleur approprié.
- Le contrôleur prend la demande et appelle les méthodes de service appropriées en fonction de la méthode GET ou POST utilisée. La méthode de service définira les données du modèle en fonction de la logique commerciale définie et renverra le nom de la vue au DispatcherServlet.
- Le DispatcherServlet prend l'aide de ViewResolver pour sélectionner la vue définie pour la demande.
- Une fois la vue terminée, le DispatcherServlet passe les données du modèle à la vue qui est finalement rendue sur le navigateur.

4. Principales interfaces graphiques

La conception des interfaces de l'application est une étape très importante puisque toutes les interactions avec le cœur de l'application passent à travers ces interfaces, on doit alors guider l'utilisateur avec les messages d'erreurs et de notification si besoin, ainsi présenter un système complet. Dans cette partie, nous allons présenter quelques interfaces de l'application, répondant aux recommandations ergonomiques de compatibilité, de guidage, de clarté, d'homogénéité et de souplesse. Nous avons choisi l'administration comme utilisateur vu qu'il présente à travers ces interactions la majeure partie des principales fonctionnalités de l'application.

a. Authentification

Pour l'authentification on a opté de créer deux pages d'authentification, une pour l'administration et l'autre pour les élèves et professeurs. Les deux pages d'authentification sont gérées par spring security qui permet aux utilisateurs de l'application de s'identifier par leurs login et leurs mots de passe, ainsi qu'un champ de l'année scolaire pour l'administration, afin de dire de quel exercice il s'agit. La figure ci-dessous représente IHM de l'authentification pour l'administration.

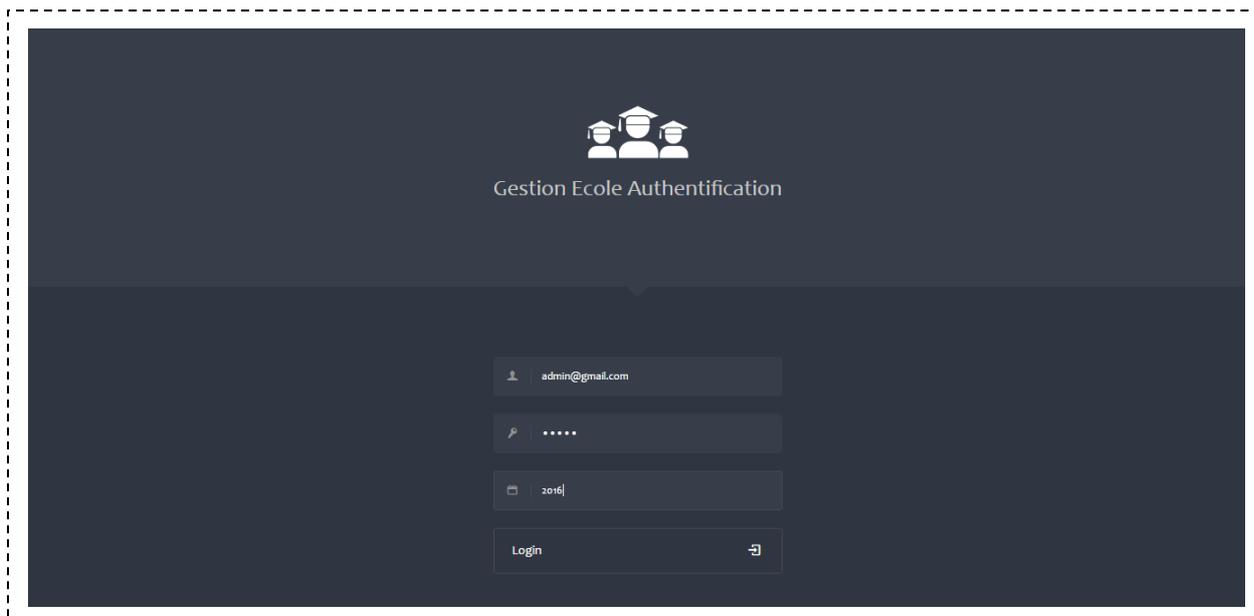


Figure 32 : Page d'authentification « administration »

Si les coordonnées de l'utilisateur sont erronées, le système affiche un message d'erreur et l'invite à ressaisir ses coordonnées.

Sinon l'utilisateur est redirigé vers sa page d'accueil, dans lequel on trouve un menu de tous les modules de l'application et chacun de ces modules contient un ensemble des fonctionnalités sous forme des sous menus relatifs à ce module.

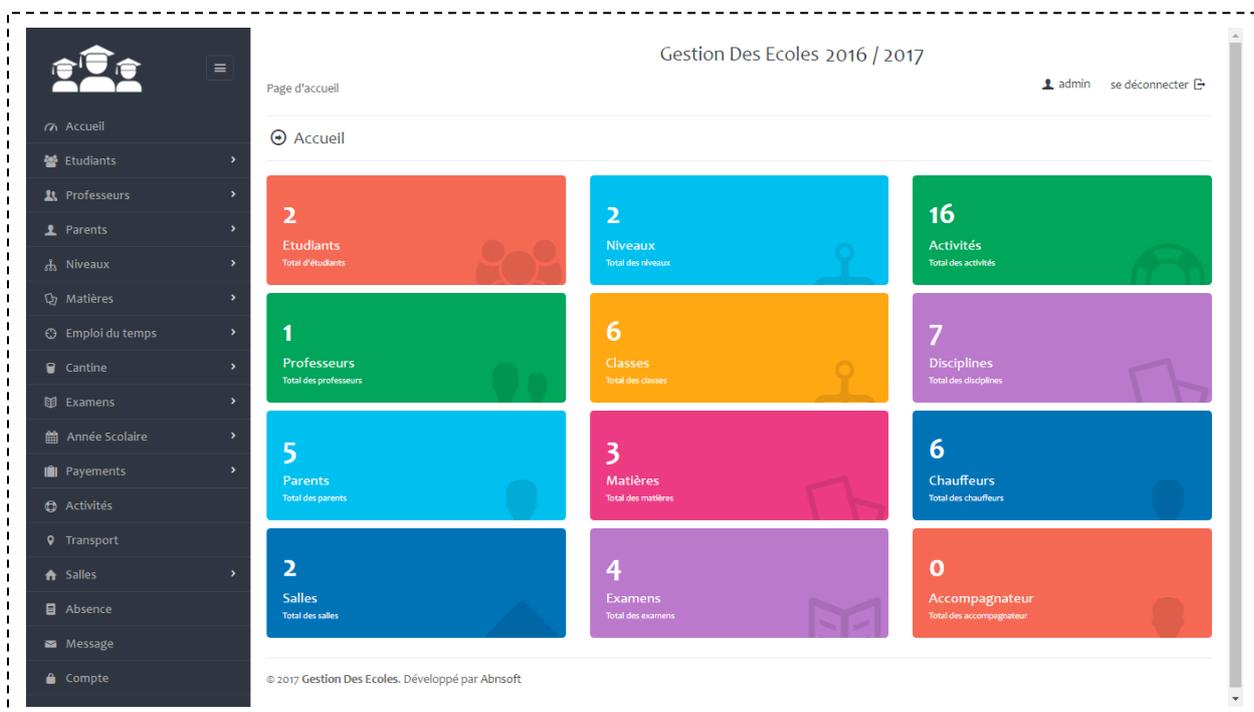


Figure 33 : Page d'accueil

b. Inscription

Les figures (34 – 35 – 36 – 37) sont visualisées lorsque le choix de l'utilisateur, a porté sur le sous menu « Inscription » du menu « Etudiants».

Cet écran permet à l'administration d'ajouter les informations d'un nouvel élève, y compris : les informations personnelles et celles de l'ancienne école, les informations des parents ou tuteurs, les informations complémentaires et de la santé, et enfin l'affectation d'élève vers le niveau d'étude. Afin de faciliter la saisie de cette quantité d'information et d'avoir une interface conviviale, nous avons structuré le formulaire sous forme d'étapes :

Etape 1 : Représente les informations personnelles de l'élève comme la photo d'identité, le nom, prénom, matricule, cne (code national d'élève)... et les informations concernant l'ancienne école y compris le nom, type, et le statut de l'école.

Etape 2 : Contient les informations des parents ou tuteurs. Cet écran vous donne deux options soit les parents ou tuteurs déjà existants, il suffit de les chercher par le frère ou sœur du nouvel élève, soit d'ajouter leurs informations en cas d'absence. La possibilité d'ajouter les informations d'un contact en cas d'urgence.

Etape 3 : Contient les informations complémentaires et des informations sur la santé de l'élève.

Etape 4 : L'étape finale dans laquelle on trouve l'affectation de l'élève vers un niveau, et on précisant les informations concernant la cantine et du transport.



- Accueil
- Etudiants
- Professeurs
- Parents
- Niveaux
- Matières
- Emploi du temps
- Cantine
- Examens
- Année Scolaire
- Payements
- Activités
- Transport
- Salles
- Absence
- Message
- Compte

Gestion Des Ecoles 2016 / 2017

admin [se déconnecter](#)

Inscription d'un élève

Accueil > Ajouter Etudiant

Etape 1

Renseignement d'identité

Etape 2

Informations des parents

Etape 3

Infos complémentaires & médicales

Etape 4

Affectation à la classe

Informations personnelles

Photo



Sélectionner une image

Matricule

Nom الإسم العائلي

Prénom الإسم الشخصي

Sexe Masculin

CNE

CIN

Date de naissance

Lieu de naissance مكان الأزداد

Nationalité الجنسية

Téléphone

Email

Adresse العنوان

Ancienne Ecole

Ancienne école المؤسسة السابقة

Type d'école Crèche Ecole

Statut de l'école Privée Publique

Enregistrer

© 2017 Gestion Des Ecoles. Développé par Abnsoft

Figure 34 : Page d'inscription étape 1

-  ☰
- [Accueil](#)
- [Etudiants](#)
- [Professeurs](#)
- [Parents](#)
- [Niveaux](#)
- [Matières](#)
- [Emploi du temps](#)
- [Cantine](#)
- [Examens](#)
- [Année Scolaire](#)
- [Payements](#)
- [Activités](#)
- [Transport](#)
- [Salles](#)
- [Absence](#)
- [Message](#)
- [Compte](#)

Gestion Des Ecoles 2016 / 2017

Inscription d'un élève

 admin [se déconnecter](#)

☰ Accueil > Ajouter Etudiant

Etape 1
Renseignement d'identité

Etape 2
Informations des parents

Etape 3
Infos complémentaires & médicales

Etape 4
Affectation à la classe

Parent existant déjà

choisir frère ou soeur

Select

Ajouter Père

Nom

Prénom

CIN

Profession

Date de naissance

Lieu de naissance

Tel Portable

Tel Domicile

Tel Professionnel

Email

adresse

Personnel

Ajouter Mère

Nom

Prénom

CIN

Profession

Date de naissance

Lieu de naissance

Tel Portable

Tel Domicile

Tel Professionnel

Email

adresse

Personnel

[Ajouter Tuteur Légal](#)

Nom	<input type="text"/>
Prénom	<input type="text"/>
CIN	<input type="text"/>
Profession	<input type="text"/>
Date de naissance	<input type="text" value="jj/mm/aaaa"/>
Lieu de naissance	<input type="text"/>
Tel Portable	<input type="text" value="06.01.01.01.01"/>
Tel Domicile	<input type="text" value="06.01.01.01.01"/>
Tel Professionnel	<input type="text" value="06.01.01.01.01"/>
Email	<input type="text"/>
adresse	<input type="text"/>
Personnel	<input type="text" value="Non"/>

Contact d'urgence

Nom	<input type="text"/>	Nom	<input type="text"/>
Prénom	<input type="text"/>	Prénom	<input type="text"/>
Lien de parenté	<input type="text"/>	Lien de parenté	<input type="text"/>
Telephone	<input type="text" value="06.01.01.01.01"/>	Telephone	<input type="text" value="06.01.01.01.01"/>

© 2017 Gestion Des Ecoles. Développé par Abnsoft

Figure 35 : Page d'inscription étape 2



- Accueil
- Etudiants
- Professeurs
- Parents
- Niveaux
- Matières
- Emploi du temps
- Cantine
- Examens
- Année Scolaire
- Payements
- Activités
- Transport
- Salles
- Absence
- Message
- Compte

Gestion Des Ecoles 2016 / 2017

Inscription d'un élève

admin se déconnecter

Accueil > Ajouter Etudiant

Etape 1
Renseignement d'identité

Etape 2
Informations des parents

Etape 3
Infos complémentaires & médicales

Etape 4
Affectation à la classe

Informations Complémentaires

1. Quelle(s) est (sont) la(les) langue(s) habituellement parlée(s) à la maison?
 Arabe Français Anglais Amazigh Autre
2. Votre enfant suit-il des cours particuliers? Non Oui
3. Votre enfant est-il aidé dans son travail scolaire? Non Oui
4. Votre enfant est-il plutôt réceptif ou réfractaire face aux règles de vie de l'école et au corps enseignant?
5. Votre enfant habite-t-il chez vous? Oui Non

Informations médicales

1. Depuis sa naissance, votre enfant a-t-il eu un (des) grave(s) problème(s) de santé? Non Oui
2. Votre enfant a-t-il subi une opération? Non Oui
3. Votre enfant prend-t-il des médicaments de manière chronique : Non Oui
4. Votre enfant présente-t-il des problèmes de vue? : Non Oui
5. Votre enfant présente-t-il des problèmes d'audition? : Non Oui
6. Le calendrier de vaccination de votre enfant est-il à jour? Oui Non
7. Votre enfant a-t-il déjà consulté chez un :
 - ORL : Non Oui
 - Ophtalmologue : Non Oui
 - Pédiopsychiatre : Non Oui
 - Psychologue : Non Oui
 - Orthophoniste : Non Oui
 - Psychomotricien : Non Oui
 - Souhaitez-vous que votre enfant participe aux excursions et sorties pédagogiques organisées par le groupe scolaire?
 Oui Non
 - Remarques particulières :

© 2017 Gestion Des Ecoles. Développé par Abnsoft

Figure 36 : Page d'inscription étape 3

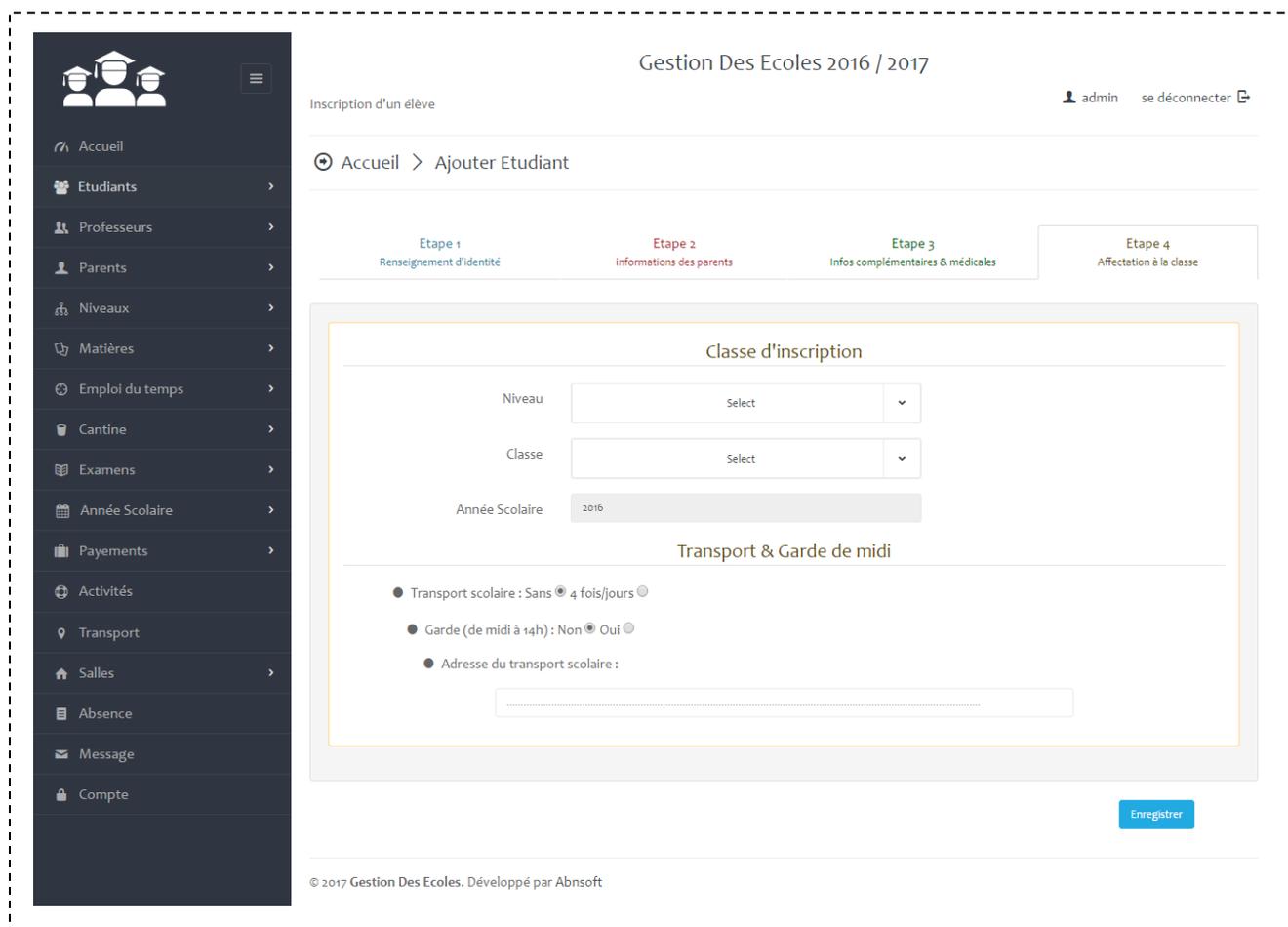


Figure 37 : Page d'inscription étape 4

Après la saisie de toutes ces informations, l'utilisateur clique sur le bouton enregistrer, dans le cas d'une erreur, l'enregistrement est annulé en précisant l'erreur effectuée.

Sinon l'utilisateur est redirigé vers la liste d'élèves dont on va voir les détails dans la partie qui suit.

c. Liste d'élèves

Cette interface qui contient l'ensemble des élèves, avec un certain nombre d'options :

- Possibilité de filtrage par : niveau – classe – année scolaire.
- La recherche rapide.
- La possibilité de voir toutes les informations concernant un élève.
- Modifier les données d'un élève.
- Supprimer un élève.

Gestion Des Ecoles 2016 / 2017

Liste des Etudiants admin se déconnecter

Accueil > Liste Des Etudiants

Niveau Classe Année Scolaire Ajouter Etudiant

Liste Des Etudiants

10 per page Search:

Photo ^	Matricule	Nom	Prenom	CNE	Adresse	Options
	matricule	kably	hamid	1210284852	266 RUE MEKKA hay Ibaraka 1 berrechid	
	matricule2	kably	ahmed	1210284855	266 rue mekka hay Ibaraka 1 berrechid	

Showing 1 to 2 of 2 entries

© 2017 Gestion Des Ecoles. Développé par Abnsoft

Figure 38 : Page de la liste d'élèves

d. Ajouter matière et ses disciplines

Cette interface permet à l'administration d'ajouter les matières avec ses disciplines d'une manière rapide. Permet aussi de donner la possibilité soit d'ajouter des disciplines sous une matière déjà existante, soit de créer d'abord la matière et puis de lui ajouter les disciplines.

Dans le cas où la matière existe, il suffit de choisir la matière et de lui ajouter ses disciplines, voir (fig : 40).

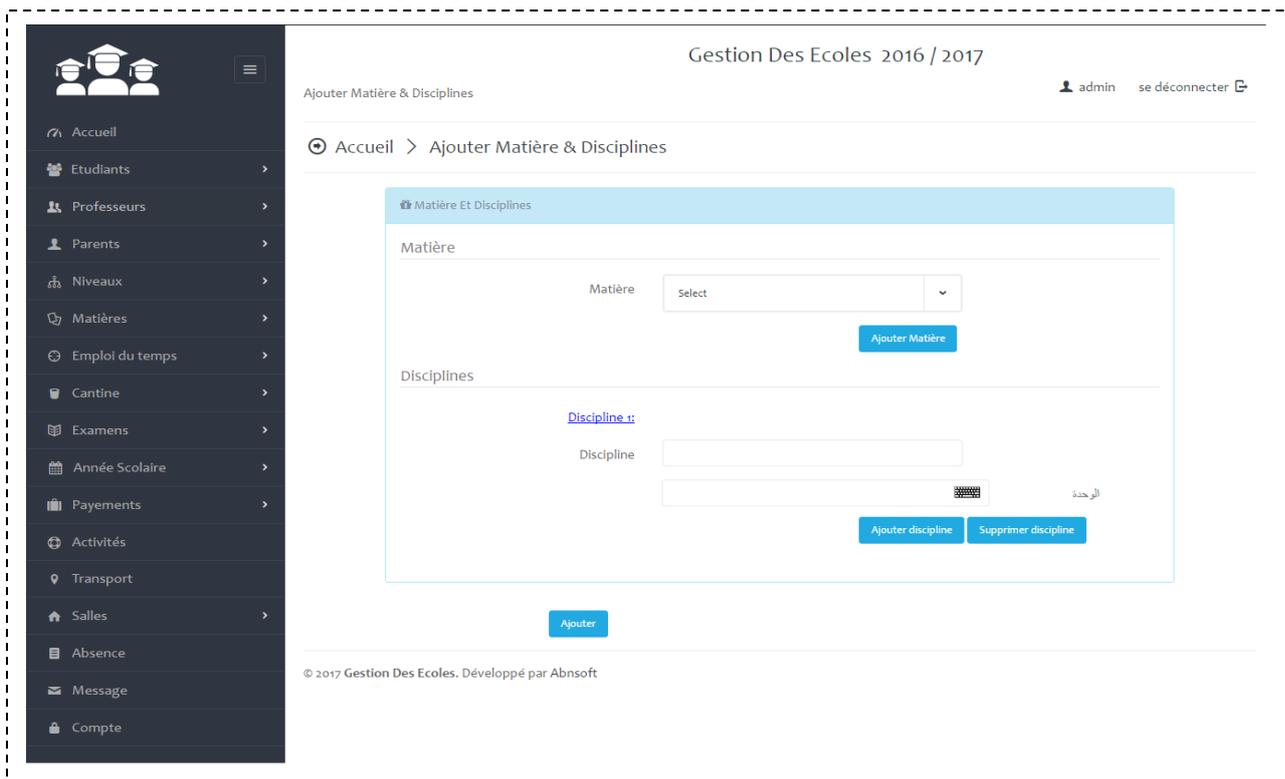


Figure 39 : Page ajouter matière et disciplines (cas 1)

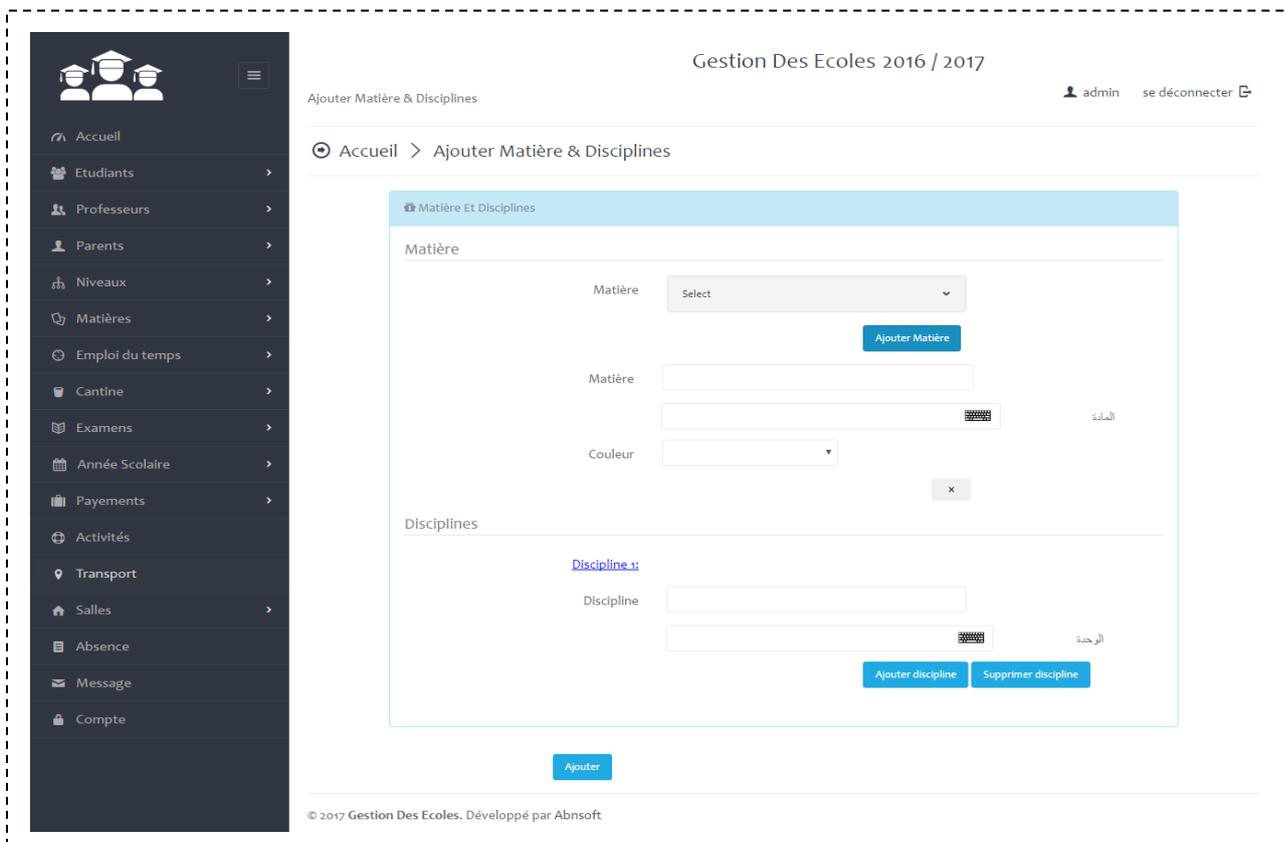


Figure 40 : Page ajouter matière et disciplines (cas 2)

e. Affectation des disciplines sur les niveaux

Dans cette interface on va voir comment on peut affecter les disciplines sur les niveaux, on précisant dans chaque discipline le coefficient dans le niveau.

Gestion Des Ecoles 2016 / 2017

Affectation Disciplines Niveaux

admin se déconnecter

Accueil > Affectation Disciplines Niveaux

Affectation Disciplines Niveaux

Niveau

Niveau

Disciplines

[Discipline 1](#)

Matiere

Discipline

Coefficient

Ajouter Discipline Supprimer Discipline

Ajouter

© 2017 Gestion Des Ecoles. Développé par Abnsoft

Figure 41 : Page affectation disciplines vers les niveaux

f. Génération d'emploi du temps

Cette interface permet à l'administration, d'élaborer l'emploi du temps d'une classe d'un niveau bien déterminé.

La gestion d'emplois du temps met en relation différentes ressources : les professeurs, les salles, la durée, le jour, les matières et la classe. L'objectif est d'affecter des enseignements à un professeur, pour une matière située dans un niveau, enseigner dans une salle, pour une durée pendant la journée et sans qu'il y ait des conflits.

Les figures ci-dessous représentent un exemple d'élaboration d'emploi du temps de la classe C2 du niveau BAC.

Après le choix de la classe, une interface est affichée sous forme d'un tableau, qui contient sur les lignes les heures et sur les colonnes les jours.

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
8 ⁰⁰					
9 ⁰⁰					
10 ⁰⁰					
11 ⁰⁰					
12 ⁰⁰					
13 ⁰⁰					
14 ⁰⁰					
15 ⁰⁰					
16 ⁰⁰					
17 ⁰⁰					

Figure 42 : Page d'élaboration d'emploi du temps (1)

Si vous cliquez, sur l'une des cases du tableau un pop-up est affiché, on vous donne la possibilité d'entrer la durée, puis la salle, le professeur et la matière.

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
8 ⁰⁰					
9 ⁰⁰					
10 ⁰⁰					
11 ⁰⁰					
12 ⁰⁰					
13 ⁰⁰					
14 ⁰⁰					
15 ⁰⁰					
16 ⁰⁰					
17 ⁰⁰					

Figure 43 : Page d'élaboration d'emploi du temps (2)

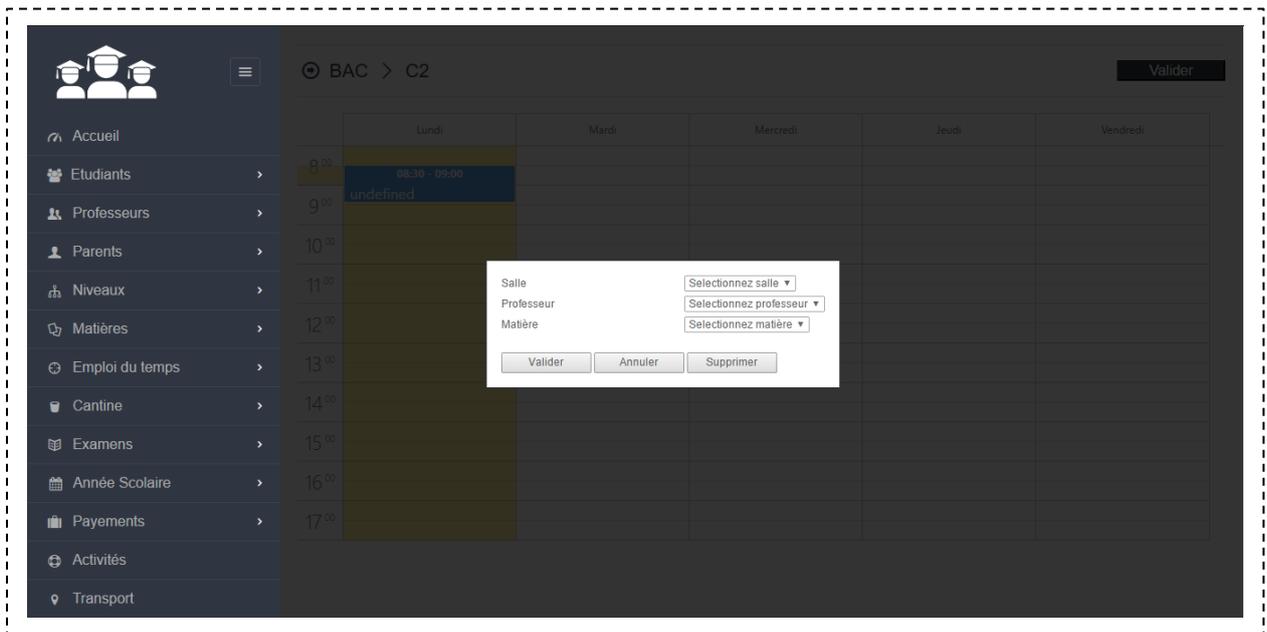


Figure 44 : Page d'élaboration d'emploi du temps (3)

Après une multiple saisie, le résultat est apparu comme la figure ci-dessous.



Figure 45 : Page d'élaboration d'emploi du temps (4)

g. Envoi des SMS

Cette interface permet à l'administration d'envoyer un SMS, ont sélectionnant le destinataire et de rédiger le message à envoyer.

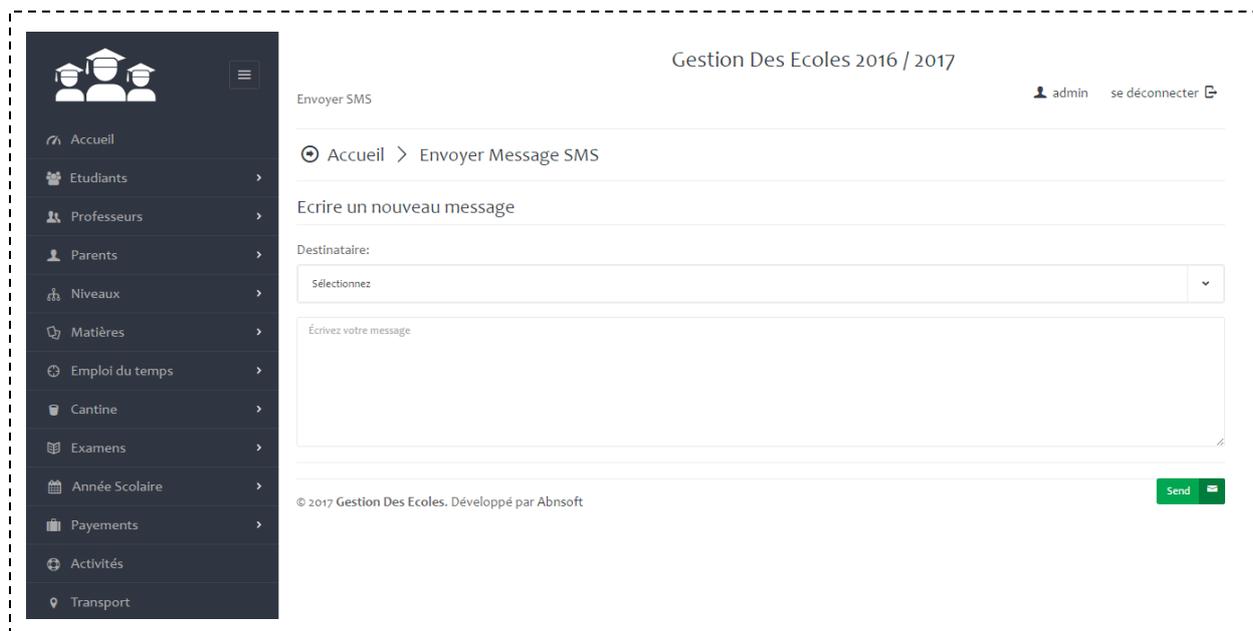


Figure 46 : Page d'envoi des SMS

5. Conclusion

A travers ce chapitre, nous avons présenté la réalisation de l'application en justifiant nos choix technologiques, en représentant quelques interfaces graphiques que nous avons jugées les plus importantes.

Conclusion général

L'objectif de notre projet de fin d'études était de concevoir et implémenter une application de gestion d'école privée, le cas de l'école « Molière ».

Le point de départ de la réalisation de ce projet était une récolte d'informations nécessaires pour dresser un état de l'existant, présenter un aperçu sur la problématique ainsi que l'architecture utilisée au sein des réseaux des établissements.

Par la suite, nous nous sommes intéressés à l'analyse et la spécification des besoins qui nous a permis de distinguer les différents acteurs interagissant avec l'application visée.

L'objectif de la partie suivante était la conception détaillée, dans laquelle nous avons fixé la structure globale de l'application. Le dernier volet de notre projet était la partie réalisation qui a été consacrée à la présentation des outils du travail et les interfaces les plus significatives de notre application.

L'apport de ce travail a été d'une importance très considérable. En effet, il nous a permis : de suivre une méthodologie de travail bien étudiée, d'approfondir nos connaissances dans le monde de développement des applications et de bien nous exercer sur le Framework spring et hibernate.

La réalisation d'un tel projet, nous a permis d'apprendre et de toucher du doigt une partie des divers aspects du métier de développeur et de celui du concepteur.

WEBOGRAPHIE

Régis pouiller, (19 mai 2014), « Réalisation d'application Web simple avec Spring »

<http://rpouiller.developpez.com/tutoriels/spring/application-web-spring-hibernate/>

Thomas Risberg, Rick Evans, Portia Tung « Développement d'une application MVC Spring Framework étape par étape »

<https://docs.spring.io/docs/Spring-MVC-step-by-step/>

Pankaj, (12 OCTOBRE 2016), « Spring MVC File Upload »

<http://www.journaldev.com/2573/spring-mvc-file-upload-example-single-multiple-files>

Mkyong, (23 août 2011), « Spring security »

<http://www.mkyong.com/tutorials/spring-security-tutorials/>

Sivateja, (4 mai 2011), « Mapping et configuration des fichiers dans Hibernate »

<http://www.java4s.com/hibernate/mapping-and-configuration-in-hibernate/>

OpenClassrooms, (9 mars 2017), « Prenez en main Bootstrap »

<https://openclassrooms.com/courses/prenez-en-main-bootstrap>

Agustin Camino, (17 Avril 2017), « SMS and MMS Notifications avec Java et Spring »

<https://www.twilio.com/docs/tutorials/server-notifications-java-spring>

Pierre Emmanuel, (2009), « Team Foundation Server »

http://pedautreppe.developpez.com/tutoriels/dotnet/integration-continue-visual-studio-team-foundation-server-partie2-msbuild-tfsbuild/?page=page_6