****

SYNETCOM

Empowered by innovation

**I**nstitut **A**fricain d’**I**nformatique

Représentation du Niger

**(I.A.I-Niger)**

**Tel:** (227) 20 72 56 72

**BP:** 12 078

**MEMOIRE DE FIN DE CYCLE**

**POUR L’OBTENTION DU DIPLOME D’INGENIEUR DES**

**TRAVAUX INFORMATIQUES**

**THEME**

**Conception d’une plateforme web de gestion d’une école supérieure**

**Lieu de stage : SYNETCOM**

**Présenté par :**

**Mlle** Sani Issaka Samira

**Mr** Ibrahim Issa Mahamadou

**Mr** Hassoumi Kadri Fayçal

**Présenté par:**

**Mlle** Sani Issaka Samira

**Mr** Ibrahim Isaa Mahamadou

**Mr** Hassoumi Kadri Fayçal

**Encadreur :**

**Mr** Chaibou Abdou

Administrateur Général

**Année académique : 2017-2018**

Sommaire

[Dédicaces i](#_Toc526902984)

[Remerciements ii](#_Toc526902985)

[Sigles et Abréviations iii](#_Toc526902986)

[Table des figures iv](#_Toc526902987)

[Table des tableaux vi](#_Toc526902988)

[Introduction 1](#_Toc526902989)

[CHAPITRE I : 2](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526902990)

[CONTEXTE GENERAL DU PROJET 2](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526902991)

[I. Présentation du centre d’accueil 3](#_Toc526902992)

[II. Cadre général du projet 4](#_Toc526902993)

[II.1 Contexte du projet 4](#_Toc526902994)

[II.2 Objectifs attendus 4](#_Toc526902995)

[CHAPITRE II : 5](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526902996)

[DEMARCHE DU PROJET 5](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526902997)

[I. Le choix de la méthode de gestion de projet 6](#_Toc526902998)

[I.1 Définition d’un processus de développement logiciel 6](#_Toc526902999)

[I.2 Le processus unifié 6](#_Toc526903000)

[I.3 Le processus 2TUP 7](#_Toc526903001)

[I.4 Un processus de modélisation avec UML 10](#_Toc526903002)

[II. Le planning prévisionnel 11](#_Toc526903003)

[CHAPITRE III : 12](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903004)

[ÉTUDE PRELIMINAIRE 12](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903005)

[I. Recueil des besoins fonctionnels 13](#_Toc526903006)

[II. Acteurs du système 16](#_Toc526903007)

[III. Identification des messages 17](#_Toc526903008)

[IV. Modélisation du contexte 19](#_Toc526903009)

[CHAPITRE IV : 21](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903010)

[CAPTURE DES BESOINS FONCTIONNELS 21](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903011)

[I. Identification des cas d’utilisation 22](#_Toc526903012)

[II. Structuration des cas d’utilisations en packages 25](#_Toc526903013)

[III. Description détaillée des cas d’utilisation 33](#_Toc526903014)

[IV. Identification des classes candidates 53](#_Toc526903015)

[CHAPITRE V : 56](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903016)

[ANALYSE 56](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903017)

[I. Diagramme de package 57](#_Toc526903018)

[II. Développement du modèle statique 58](#_Toc526903019)

[CHAPITRE VI : 63](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903020)

[CAPTURE DES BESOINS TECHNIQUES 63](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903021)

[I. Architecture Matérielle 64](#_Toc526903022)

[I.1 L’architecture 1 tiers 64](#_Toc526903023)

[I.2 L’architecture 2 tiers 64](#_Toc526903024)

[I.3 L’architecture 3 tiers 65](#_Toc526903025)

[I.4 Le choix de l’architecture 66](#_Toc526903026)

[CHAPITRE VII : 67](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903027)

[REALISATION 67](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526903028)

[I. Technologies et outils de développement 68](#_Toc526903029)

[I.1 Matériel technique 68](#_Toc526903030)

[I.3 Système de gestion de base des données utilisé : MySQL 70](#_Toc526903031)

[I.4 Framework de développement 70](#_Toc526903032)

[1. Pourquoi utiliser un Framework ? 70](#_Toc526903033)

[2. Laravel 70](#_Toc526903034)

[II. Quelques captures d’écrans 71](#_Toc526903035)

[II.1 Authentification 71](#_Toc526903036)

[II.2 Gestion des droits d’accès 72](#_Toc526903037)

[II.3 Profil des utilisateurs 74](#_Toc526903038)

[II.4 Affectation des notes 75](#_Toc526903039)

[II.5 Emplois du temps 76](#_Toc526903040)

[II.6 Absences 77](#_Toc526903041)

[Conclusion générale 78](#_Toc526903042)

[Bibliographie 79](#_Toc526903043)

[Webographie 80](#_Toc526903044)

# Dédicaces

A

Nos parents

Pour les sacrifices déployés à nos égards ; pour leur patience leur amour et leur confiance en nous. Ils ont tout fait pour notre bonheur et notre réussite. Qu'ils trouvent dans ce modeste travail, le témoignage de notre Profonde affection et de notre attachement indéfectible. Nulle dédicace ne puisse exprimer ce que nous leur devons.

Que Dieu leur réserve la bonne santé et une longue vie.

A

Nos amis

En témoignage de nos sincères reconnaissances pour les efforts Qu'ils ont consentis pour nous soutenir au cours de nos études. Que Dieu nous garde toujours unis.

A

Toutes les personnes qui ont contribué de près ou de loin à la réalisation de présent projet.

# Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voudrons témoigner toute notre reconnaissance.

En préambule à ce mémoire nous remercions ALLAH qui nous aide et nous donne la santé, la patience ainsi que le courage durant ces longues années d’étude.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l’élaboration de ce mémoire ainsi qu’à la réussite de cette formidable année universitaire.

Ces remerciements vont également à l’égard du corps professoral et administratif de notre école IAI-Niger, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Nous tenons à remercier sincèrement Monsieur Chaibou Abdou, qui, en tant que Directeur de mémoire, s’est toujours montré à l’écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l’inspiration, l’aide et le temps qu’il a bien voulu nous consacrer et sans qui, ce mémoire n’aurait jamais vu le jour.

On n’oublie pas nos très chers parents qui ont toujours été là pour nous, pour leur contribution, leur soutien et leur patience. Vous nous avez donné un magnifique modèle de labeur et de persévérance. Nous sommes redevables d’une éducation dont on est fier.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragé.

Merci à tous et à toutes.

# Sigles et Abréviations

LMD: Licence-Master-Doctorat

UE: Unité d’Enseignement

TIC : Technologies d’Information et la Communication

NTIC: Nouvelles Technologies d’Information et de la Communication

TD: Travaux dirigés

TP: Travaux pratiques

SMS: Short Message Service (service de messages succincts)

UML: Unified Modeling Language

ETEC : Ecole des Techniques Economiques et Comptables et de Communication

UP: Unified Process (Processus Unifié)

T2UP: Two Tracks Unified Process

PHP: Hypertext Preprocessor

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheets (feuilles de style en cascade)

JS: JavaScript

POO : Programmation Orienté Objet

# Table des figures

[Figure 1:Organigramme de SYNETCOM 3](#_Toc526901324)

[Figure 2 : Le système d’information soumis à deux types de contraintes 8](#_Toc526901325)

[Figure 3 : Le processus de développement en Y 9](#_Toc526901326)

[Figure 4 : Planning du projet 11](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901327)

[Figure 5 : Diagramme de contexte statique 17](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901328)

[Figure 6 : Diagramme de packages 28](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901329)

[Figure 7 : Diagramme de cas d’utilisation, package « gestion de la bibliothèque » 29](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901330)

[Figure 8 : Diagramme de cas d’utilisation, package « gestion administrative » 29](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901331)

[Figure 9: Diagramme de cas d’utilisation, package « gestion des enseignements » 30](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901332)

[Figure 10 : Diagramme de cas d’utilisation, package « gestion des promotions » 31](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901333)

[Figure 11 : Diagramme de cas d’utilisation, package « gestion des notes » 31](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901334)

[Figure 12 : Diagramme de cas d’utilisation, package «Gestion des absences/emplois du temps» 32](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901335)

[Figure 13 : Diagramme de cas d’utilisation, package « Gestion des droits d’accès » 33](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901336)

[Figure 14 : Cas d’utilisation gérer les filières, package « Gestion des enseignements » 35](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901337)

[Figure 15:Diagramme d’activité gérer les filières, package « Gestion des enseignements » 38](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901338)

[Figure 16: Diagramme de séquence gérer les filières, package « Gestion des enseignements » 39](#_Toc526901339)

[Figure 17 : cas d’utilisation : Gérer les classes, package « Gestion des enseignements » 40](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901340)

[Figure 18 : Diagramme d’activité gérer les classes, package « Gestion des enseignements » 43](#_Toc526901341)

[Figure 19 : Diagramme de séquence gérer les classes, package « Gestion des enseignements » 44](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901342)

[Figure 20 : Cas d’utilisation : gérer les années scolaires, package « Gestion des promotions » 45](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901343)

[Figure 21 : Diagramme de séquence gérer les années scolaires, package « Gestion des promotions» 47](#_Toc526901344)

[Figure 22 : Cas d’utilisation maintenir les notes, package « Gestion des notes » 48](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901345)

[Figure 23 : Diagramme de séquence maintenir les notes, package « Gestion des notes» 49](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901346)

[Figure 24 : Diagramme d’activité gérer les étudiants, package « Gestion Administrative » 52](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901347)

[Figure 25 : Diagramme de classe participante des cas d’utilisation, package « gestion des enseignements » 53](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901348)

[Figure 26: Diagramme de classe participante des cas d’utilisation, package « Gestion des promotions » 54](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901349)

[Figure 27 : Diagramme de classe participante des cas d’utilisation, package « Gestion des absences/emplois du temps » 55](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901350)

[Figure 28 : Diagramme de package 57](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901351)

[Figure 29:Diagramme de classe package « gestion des enseignements » 58](#_Toc526901352)

[Figure 30 : Diagramme de classe package gestion des emplois 59](#_Toc526901353)

[Figure 31 : Diagramme de classe package gestion des utilisateurs. 60](#_Toc526901354)

[Figure 32 : Diagramme de package gestion de la scolarité. 61](#_Toc526901355)

[Figure 33: Diagramme de classe package gestion des évaluations 62](#_Toc526901356)

[Figure 34 : Architecture 1 tiers 64](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901357)

[Figure 35 : Architecture 2 tiers 65](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901358)

[Figure 36 : Architecture 3 tiers 65](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901359)

[Figure 37: Page d'authentification 71](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901360)

[Figure 38: Page mot de passe oublié 72](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901361)

[Figure 39 : Interface d’ajout d’un nouveau rôle 73](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901362)

[Figure 40 : Liste des rôles 73](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901363)

[Figure 41 : Profil d’un personnel administratif (Secrétaire) 74](#_Toc526901364)

[Figure 42 : Profil d’un enseignant 74](#_Toc526901365)

[Figure 43 : Profil d’un étudiant 75](#_Toc526901366)

[Figure 44:Affectation de note 76](#_Toc526901367)

[Figure 45: Emplois de temps 77](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901368)

[Figure 46 : Gestion des Absences 77](file:///C:\Users\hp\Downloads\Memoire%203aap%20Final.docx#_Toc526901369)

# Table des tableaux

[Table 1 : Exemple de calcul de la moyenne d’une unité d’enseignement 13](#_Toc526903045)

[Table 2 : Exemple de calcul de la moyenne d'un élément. 14](#_Toc526903046)

[Table 3 : Description des besoins fonctionnels 20](#_Toc526903047)

[Table 4 :Liste des cas d’utilisation 25](#_Toc526903048)

[Table 5 : Liste des cas d’utilisation par package 27](#_Toc526903049)

[Table 6 : Formalisme de description d’un cas d’utilisation 34](#_Toc526903050)

[Table 7 : Description détaillé, cas d’utilisation gérer les filières, package « Gestion des enseignements » 37](#_Toc526903051)

[Table 8 : Description détaillé, cas d’utilisation gérer les classes, package « Gestion des enseignements » 42](#_Toc526903052)

[Table 9 : Description détaillé, cas d’utilisation gérer les années scolaires, package « Gestion des promotions » 46](#_Toc526903053)

[Table 10 : Description détaillée, cas d’utilisation maintenir les notes, package « Gestion des notes » 49](#_Toc526903054)

[Table 11 : Description détaillée, cas d’utilisation :gérer les étudiants , package « Gestion administrative » 51](#_Toc526903055)

# Introduction

La révolution des technologies de l’information et de la communication (TIC), a engendré au cours de ces dernières années une progression notable de la création et de l’usage des applications de gestion dans la vie courante et professionnelle. Dans ce contexte de transformation numérique, et afin d’aider les établissements scolaires à mettre en place un système de gestion automatique et rationalisé, SYNETCOM a initié un projet de développement d’une plateforme web standard qui répondra aux attentes des gestionnaires des établissements scolaires supérieurs, objet de la présente étude.

La présente étude qui nous a été soumise dans le cadre de notre projet de fin d’étude pour l’obtention du diplôme d’ingénieurs des travaux informatiques, vient pour répondre aux besoins d’un secteur de base intéressant et très sensible, à savoir l’éducation.

En effet, l’éducation fait partie des domaines qui sont à la traine dans l’adoption des outils TIC alors qu’il en a plus besoin compte tenu du volume d’information géré tant au niveau des enseignements qu’au niveau des enseignés mais aussi au niveau des différents acteurs impliqués.

Pour concevoir et mettre en place un système numérique répondant à notre thème « Conception d’une plateforme web de gestion d’une école supérieure », et qui réunit les différentes fonctionnalités nécessaires à la gestion des établissements supérieurs, prenant en compte la reforme LMD (Licence-Master-Doctorat) d’une école supérieure, le présent mémoire s’articulera autour de six chapitres :

* Premier chapitre intitulé : Contexte général du projet
* Deuxième chapitre : Démarche du projet
* Troisième chapitre : Etude préliminaire
* Quatrième chapitre : Capture des besoins fonctionnels
* Cinquième chapitre : L’Analyse
* Sixième chapitre : Capture des besoins techniques
* Septième chapitre : Réalisation

# CHAPITRE I :

# CONTEXTE GENERAL DU PROJET

Ce chapitre a pour but de situer le projet dans son environnement organisationnel et contextuel. Il présente dans la première partie l’organisme d’accueil, sa structure et ses activités, tandis que dans la deuxième partie, présente le cadre général du projet.

# Présentation du centre d’accueil

SYNETCOM est une jeune entreprise spécialisée dans les domaines des nouvelles technologies de l’information et de la communication.

Vecteur d'innovation, levier d'efficacité, accélérateur de développement, grâce à son expérience et à son anticipation permanente sur les besoins des clients, SYNETCOM agit en véritable partenaire technologique pour lever les défis et les difficultés auxquels heurtent ses partenaires et optimiser les performances de leur système de gestion.

Sa vocation est d’apporter les ressources nécessaires pour faire bénéficier au mieux à ses partenaires les atouts des nouvelles technologies

SYNETCOM réfléchit aux problèmes du marché et conçoit les solutions novatrices et les services d'accompagnement adaptés aux besoins des clients.

**Organigramme**

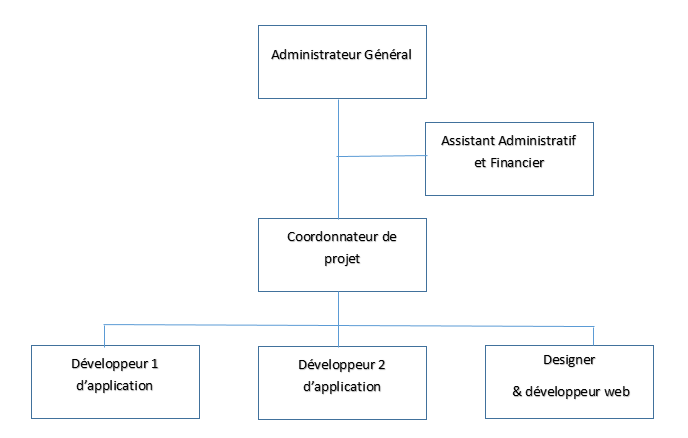


Figure 1:Organigramme de SYNETCOM

# Cadre général du projet

## Contexte du projet

Le besoin pressant d’optimisation, de rationalisation et de rentabilisation des systèmes d’information de gestion impose aux organisations l’automatisation de toutes leurs chaines de gestion. Comme tout autre domaine, l’éducation ne déroge pas à cette règle.

En effet, avec la mise en place récente du système LMD dans la sous-région, la gestion des écoles supérieures est devenue plus complexe. Alors pour apporter une solution aux acteurs chargés de la gestion de ces écoles, SYNETCOM s’est donnée comme objectif la réalisation d’une application permettant à ces responsables d’écoles supérieures d’avoir une maîtrise permanente de la gestion des activités principales de leurs établissements dont entre autres : la gestion de la scolarité, la gestion des notes, la gestion des emplois du temps, la gestion des absences, la gestion des professeurs, la gestion et le suivi du cursus des étudiants…etc.

## Objectifs attendus

L’objectif de ce projet est le développement d’une application web baptisée « Universchool » prenant en compte la gestion d’un ensemble de module dont entre autres :

* Gestion de la scolarité ;
* Gestion des dossiers des étudiants ;
* Consultation des profils étudiants, notes, absence et tableau de discipline ;
* Réception des notifications par les élèves via Mail et/ou SMS sur des mises à jour d’informations (notes, emploie du temps…) ;
* Permettre aux étudiants de consulter et télécharger des documents (anciens devoirs, examen, TP, TD…) ;
* Gestion des notes par l’administration ou les enseignants ;
* Gestion d’affectation des modules ou des matières aux enseignants ;
* Gestion des frais d’inscription et de scolarité ;
* Etablir les états ;
* Impression des états et autres indicateurs ;

Le processus d’analyse est une étape très importante, elle nécessite des méthodes permettant de mettre en place un modèle sur lequel nous allons nous appuyer. Le présent chapitre détaille le langage UML, ainsi que la démarche utilisée pour le développement du projet, et enfin l’organisation du projet.

# CHAPITRE II :

# DEMARCHE DU PROJET

# Le choix de la méthode de gestion de projet

Il existe plusieurs méthodes (2TUP, RUP, XP, AUP et OpenUP …etc.). Cependant, pour mieux guider notre analyse, nous allons porter notre choix sur la méthode qui répond aux questions ci-dessous :

* + Comment vais-je organiser les équipes de développement ;
  + Quelles tâches attribuer à qui ;
  + Quel temps faudrait-il pour livrer le produit ;
  + Comment faire participer le client au développement afin de capter les besoins de celui-ci ;
  + Comment éviter des dérives et de mauvaises estimations qui vont allonger les coûts et le temps de développement.
  + Comment vais-je procéder pour que le produit soit évolutif et facilement maintenable.

Notre choix s’est porté vers la méthode 2TUP, du fait de son approche nouvelle, originale. A ce titre nous expliquerons les concepts intervenant dans la méthode.

## Définition d’un processus de développement logiciel

Un processus définit une séquence d’étapes, en partie ordonnées, qui concourent à l’obtention d’un système logiciel ou à l’évolution d’un système existant. L’objet d’un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

## Le processus unifié

Le Processus Unifié (PU ou UP en anglais pour Unified Process) est une méthode de développement logiciel construite sur UML ; elle est itérative et incrémentale, centrée sur l’architecture, conduite par les cas d’utilisation et pilotée par les risques :

* **Itérative et incrémentale :** la méthode est itérative dans le sens où elle propose de faire des itérations lors de ses différentes phases, ceci garantit que le modèle construit à chaque phase ou étape soit affiné et amélioré. Chaque itération peut servir aussi à ajouter de nouveaux incréments.
* **Conduite par les cas d’utilisation :** elle est orientée utilisateur pour répondre aux besoins de celui-ci.
* **Centrée sur l’architecture :** les modèles définis tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide.
* **Pilotée par les risques :** en définissant des priorités pour chaque fonctionnalité, on peut minimiser les risques d’échec du projet.

La gestion d’un tel processus est organisée d’après les 4 phases suivantes :

* **Pré-étude :** c’est ici qu’on évalue la valeur ajoutée du développement et la capacité technique à le réaliser (étude de faisabilité).
* **Elaboration :** sert à confirmer l’adéquation du système aux besoins des utilisateurs et à livrer l’architecture de base.
* **Construction :** sert à livrer progressivement toutes les fonctions du système.
* **Transition :** déployer le système sur des sites opérationnels.

Chaque phase est elle-même décomposée séquentiellement en itérations limitées dans le temps (entre 2 et 4 semaines). Le résultat de chacune d’elles est un système testé, intégré et exécutable. L’approche itérative est fondée sur la croissance et l'affinement successif d’un système par le biais d’itérations multiples. Le système croît avec le temps de façon incrémentale, itération par itération, et c’est pourquoi cette méthode porte également le nom de développement itératif et incrémental. Il s’agit là du principe le plus important du Processus Unifié.

Ces activités de développement sont définies par 6 disciplines fondamentales qui décrivent la capture des besoins, la modélisation métier, l’analyse et la conception, l’implémentation, le test et le déploiemen*t.*

Notons que ces différentes étapes (ou disciplines) peuvent se dérouler à travers plusieurs phases. Le processus unifié doit donc être compris comme une trame commune des meilleures pratiques de développement.

## Le processus 2TUP

On dit du processus UP qu’il est générique c’est à dire qu’il définit un certain nombre de critères de développement, que chaque société peut par la suite personnaliser afin de créer son propre processus plus adapté à ses besoins. C’est dans ce cadre que la société Valtech a créé le processus 2TUP.

2TUP signifie « 2 Track Unified Process ». C’est un processus qui répond aux caractéristiques du Processus Unifié. Le processus 2TUP apporte une réponse aux contraintes de changement continuel imposées aux systèmes d’information de l’entreprise. En ce sens, il renforce le contrôle sur les capacités d’évolution et de correction de tels systèmes.

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à :

* Identifier les acteurs qui vont interagir avec le système à construire ;
* Identifier les messages qu'échangent les acteurs et le système ;
* Produire le cahier des charges ;
* Modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de 3 phases essentielles :

**

Figure 2 : Le système d’information soumis à deux types de contraintes

La branche gauche (fonctionnelle) : capitalise la connaissance du métier de l’entreprise. Elle constitue généralement un investissement pour le moyen et le long terme.

Les fonctions du système d’information sont en effet indépendantes des technologies utilisées.

Cette branche comporte les étapes suivantes :

* La capture des besoins fonctionnels, qui produit un modèle des besoins focalisés sur le métier des utilisateurs.
* L’analyse.

La branche droite (architecture technique) : capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l’être en effet indépendamment des fonctions à réaliser.

Cette branche comporte les étapes suivantes :

* La capture des besoins techniques.
* La conception générique.

La branche du milieu : à l’issue des évolutions du modèle fonctionnel et de l’architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l’obtention d’un processus en forme de Y.

Cette branche comporte les étapes suivantes :

* La conception préliminaire.
* La conception détaillée.
* Le codage.
* L’intégration.

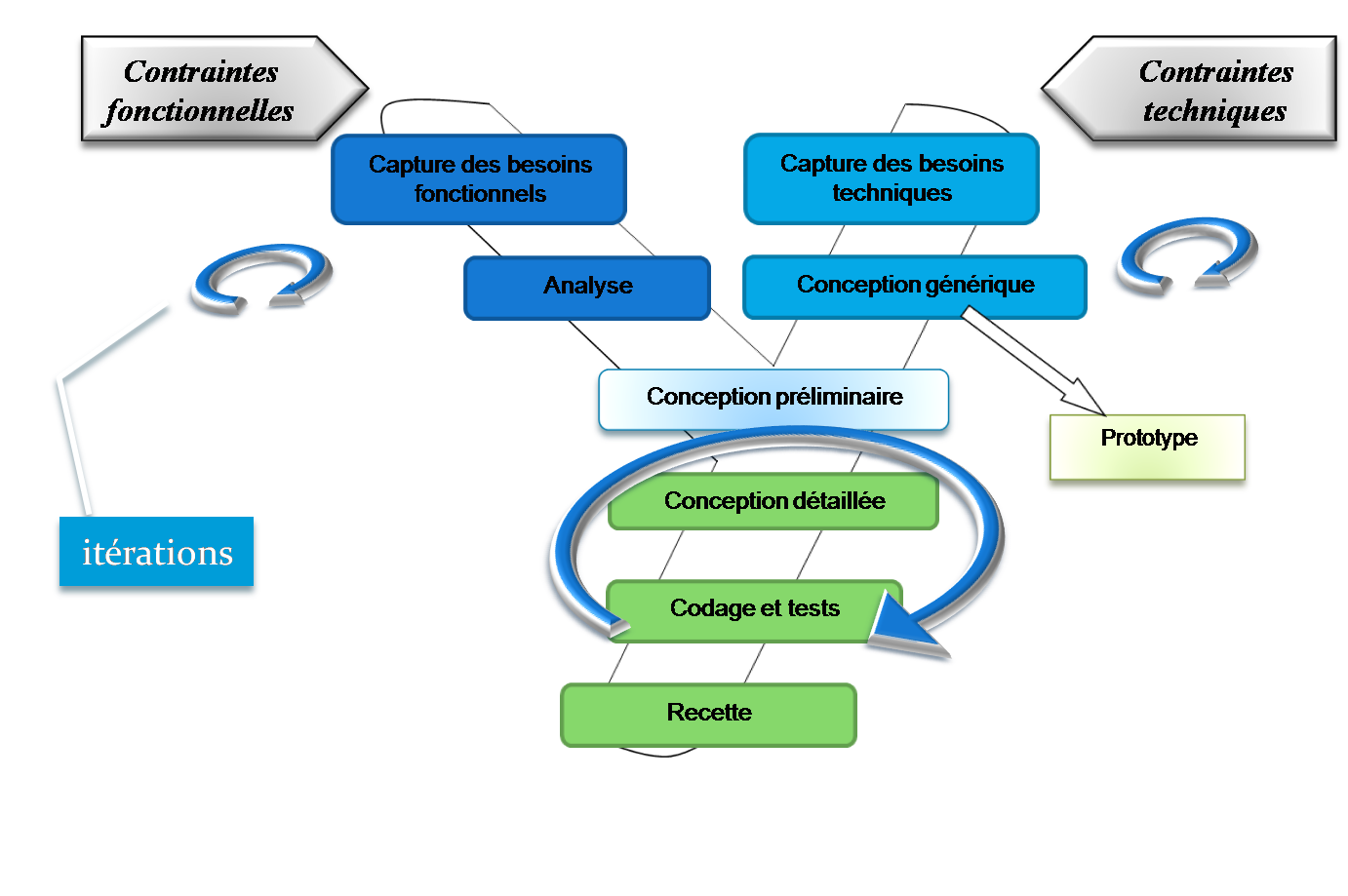


Figure 3 : Le processus de développement en Y

## Un processus de modélisation avec UML

Le processus 2TUP s’appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de par leur facilité et clarté, de bien modéliser le système à chaque étape.

« Unified Modeling Language » : UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins ; spécifier, concevoir des solutions et communiquer des points de vue. (Pitman, 2006)

UML unifie à la fois les notations et les concepts orientés objets. Il ne s’agit pas d’une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d’un langage, c’est pour ça qu’UML est présenté parfois comme une méthode alors qu’il ne l’est absolument pas.

UML unifie également les notations nécessaires aux différentes activités d’un processus de développement et offre, par ce biais, le moyen d’établir le suivi des décisions prises, depuis la définition des besoins jusqu’au codage. (Roques, 2006)

Voici une présentation rapide des différents diagrammes UML qui vont être utilisés tout au long du projet :

**Le diagramme des cas d’utilisation :** représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques.

**Le diagramme d’activités :** représente les règles d’enchaînement des activités et actions dans le système. Il peut être assimilé comme un algorithme mais schématisé.

**Le diagramme de packages :** présent depuis UML 2.0, ce diagramme modélise des catégories cohérentes entre elles, pour un souci de partage des rôles.

**Le diagramme de classes :** sûrement l’un des diagrammes les plus importants dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs.

En conception, le diagramme de classes représente la structure d’un code orienté objet.

**Le diagramme de séquence :** représente les échanges de messages entre objets, dans le cadre d’un fonctionnement particulier du système.

# Le planning prévisionnel

La clé principale de la réussite d’un projet est un bon planning. En effet, le planning aide à bien subdiviser le travail et séparer les taches à réaliser, il offre une meilleure estimation et gestion de temps nécessaire pour chaque tâche. De plus, il donne assez de visibilité permettant d’estimer approximativement la date d’achèvement de chaque tâche.

Nous avons choisi comme outil pour la planification de notre projet Gantt Project qui est un logiciel de gestion de projet.

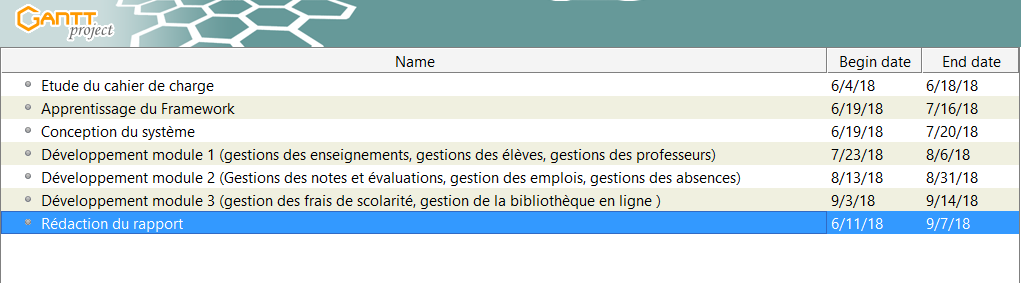
Gantt Project permet de planifier les projets et les ressources, et d’assurer le suivi des projets durant leur réalisation. Il permet de réaliser une représentation graphique du déroulement d'un projet et de rendre compte de son avancement.

Figure 4 : Planning du projet

L’étude préliminaire est la toute première étape du processus 2TUP dont l’objet est de délimiter le périmètre du projet. Elle consiste à établir un recueil initial des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou diagrammes très simples. Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture techniques.

# CHAPITRE III :

# ÉTUDE PRELIMINAIRE

# Recueil des besoins fonctionnels

Cette phase correspond à une recherche sur le terrain pour bien définir le cadre de notre système. Nous avons effectué plusieurs recherches pour identifier au mieux les besoins auxquels l’application doit répondre, et ceci afin de répondre aux attentes des potentiels utilisateurs.

A cet effet nous sommes allés chercher les informations à travers des interviews menées avec le Directeur des études de l’école supérieure ETEC (Ecole des Techniques Economiques et Comptables et de Communication).

Tout ceci dans le but de comprendre le fonctionnement du système LMD, et avoir une idée de comment s’effectue la gestion d’une école supérieure basée sur le système LMD :

**Le Système LMD :**

La réforme LMD pour « Licence-Master-Doctorat » désigne un ensemble de mesures modifiant le système d’enseignement supérieur pour l’adapter aux standards internationaux. Elle met en place principalement une architecture basée sur trois grades : licence, master et doctorat ; une organisation des enseignements en semestres et unités d’enseignement.

***Semestrialisation et système de crédits :***

Chaque diplôme est organisé en semestre ; 6 pour la Licence et 4 semestres supplémentaires pour le Master. Un semestre de formation est constitué d'Unités d'Enseignement (UE) constituée d’une ou de plusieurs « matières » dispensés par toute forme d’enseignements (Cours Magistraux, Travaux Dirigés, Travaux Pratiques, projets).

A chaque Unité d'Enseignement est affectée une valeur en crédits qui correspond au volume de travail global (présentiel et personnel) que l'étudiant doit fournir pour obtenir son UE. Chaque semestre permet de valider 30 crédits, 180 crédits sont donc nécessaires pour obtenir une Licence et 120 crédits supplémentaires pour le Master (soient 300 crédits au total pour le niveau Bac +5).

***Les modalités d’évaluations :***

Le passage de l’étudiant du niveau courant au niveau supérieur de son parcours repose sur l’évaluation des unités d’enseignement. L’évaluation consiste à passer un ou plusieurs devoirs par l’étudiant pour chaque unité du parcours qu’il étudie. La moyenne d’une unité est calculée à travers les notes de ses éléments, pondérées par ses coefficients. Le tableau ci-dessous est un exemple illustratif.

|  |  |  |  |
| --- | --- | --- | --- |
| Unité | Elément Constitutif | Coefficient | Moyenne d’un élément |
| Unité Y | Elément 1 | 3 | 10 |
| Elément 2 | 1.5 | 12.5 |
| Elément 3 | 4 | 13 |

Table 1 : Exemple de calcul de la moyenne d’une unité d’enseignement

La moyenne de l’unité *Unité Y* est ((3×10) + (1.5×12.5) + (4×13)) / 8.5 = 11.85

Le régime d’évaluation est soit le régime mixte, soit le régime contrôle continu :

* Le régime mixte : Il s’agit d’un régime d’évaluation qui regroupe les examens finaux (à raison de 70% de la note finale) ainsi que le contrôle continu (à raison de 30% de la note finale). La note du contrôle continu est scindée en devoirs surveillés et/ou travaux pratiques (20%) et d’exercices, de tests oraux, de présentations etc. (10%).

L’étudiant qui n’est pas admis en session principale a le droit de repasser dans la session de rattrapage les examens finaux des unités de régime mixte qu’il n’a pas réussies.

|  |  |  |
| --- | --- | --- |
| Elément Y | Examen final | 11.00 |
| Contrôle continu | 13.50 |
| Autre note | 14.00 |

Table 2 : Exemple de calcul de la moyenne d'un élément.

Ici, la moyenne de l’élément « Élément Y » est (11×0.7) + (13.5×0.2) + (14×0.1) = 11.80

* Le régime contrôle continu : Ce régime d’évaluation repose exclusivement sur le contrôle continu (devoirs surveillés, tests oraux, présentations, travaux pratiques etc.). Le régime contrôle continu applique les taux de 80% pour les devoirs surveillés et 20% pour les autres modalités d’examen tels que les exercices, les travaux pratiques et les exposés.

***Les modalités de validation :***

Chaque unité d’enseignement est validée soit par capitalisation définitive, soit par compensation.

La validation par capitalisation définitive se produit quand la moyenne de l’ensemble des notes obtenues entre les éléments constitutifs de l’UE est égale ou supérieur à 10.

Dans le cas de la validation par compensation d’une unité, l’étudiant ne capitalise pas définitivement l’unité. Il capitalise néanmoins les crédits des éléments constitutifs pour lesquels il a eu une moyenne supérieure à dix (10).

Pour chaque semestre d’enseignement, deux sessions de contrôle sont organisées la deuxième session est une session de rattrapage. Le semestre est acquis pour tout étudiant ayant obtenu l’ensemble des UE qui le composent.

Le semestre peut être également acquis par compensation entres les différentes UE. La moyenne générale est calculée sur la base des moyennes obtenues aux UE composant le semestre pondéré par leurs coefficients respectifs. Le semestre est alors acquis si cette moyenne est égale ou supérieure à 10.

En cas d’échec à la première session, l’étudiant se présente aux examens de rattrapage pour les UE non acquises.

L’étudiant garde le bénéfice des matières de l’UE pour lesquelles il a obtenu une moyenne égale ou supérieure à 10.

***Les modalités de passage :***

L’évaluation des étudiants est semestrielle, mais le passage d’un niveau à un autre est annuel. L’étudiant passe d’une année à l’autre par :

* L’obtention d’une note supérieure ou égale à 10 sur 20 dans toutes les unités d’enseignement de l’année concernée. Dans ce cas l’étudiant capitalise définitivement toutes ses unités.
* L’obtention d’une moyenne annuelle supérieure ou égale à 10 sur 20 par compensation entre toutes les unités d’enseignement.
* La réussite conditionnée : l’étudiant qui n’a pas obtenu la moyenne annuelle (10/20) peut passer :
* De la première année à la deuxième année, s’il capitalise 75% des crédits de la première année, c’est‐à‐dire, au moins, quarante-cinq (45) crédits des soixante (60) crédits requis.
* De la deuxième année à la troisième année s’il capitalise 75% des crédits de la deuxième année. Dans tous les cas, l’obtention des crédits objet du passage conditionné de la première année est exigée pour passer à la troisième année.

***La session de rattrapage :***

La session de rattrapage concerne les étudiants qui n’ont pas été admis en session principale.

Elle ne concerne que les unités à régime mixte pour lesquelles l’étudiant a eu une moyenne inférieure à dix (10) sur vingt (20), et plus particulièrement les éléments constitutifs (de ces unités) pour lesquels il a eu moins que dix (10) sur vingt (20).

En d’autres termes, l’étudiant recalé à la session de rattrapage repasse les examens finaux des éléments constitutifs dans les unités d’enseignement à régime mixte où il a eu une moyenne strictement inférieure à dix (10).

# Acteurs du système

Définition : un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l’état du système, en remettant et/ou en recevant des messages éventuellement porteurs de données.

Identification des acteurs : Les acteurs qui interagissent avec le système dans un premier temps sont :

* Les étudiants : Un étudiant peut consulter son dossier scolaire, ses notes et absences, son emploi du temps.
* Le personnel administratif : le personnel administratif permet de gérer le dossier scolaire des étudiants (cursus de l’étudiant), de gérer les enseignements (UE, module, cycle, filière, niveau), gérer les relever de notes, gérer les absences et emplois du temps, gérer les enseignants, gérer les frais de scolarité.
* Les enseignants : l’enseignant affecte les notes des étudiants de ses évaluations.
* L’administrateur : l’administrateur a accès à toutes les fonctionnalités de bases et avancées de l’application. Il effectue les paramétrages de base. Il crée les profils utilisateurs et attribue les droits d’accès.

Diagramme de contexte statique :

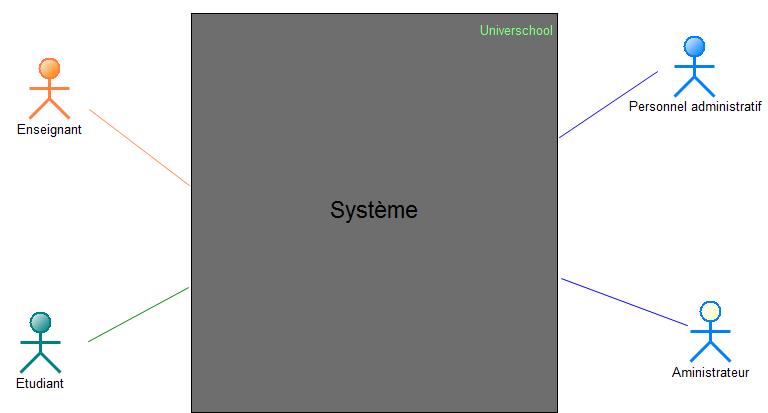


Figure 5 : Diagramme de contexte statique

# Identification des messages

Définition : un message représente la spécification d’une communication unidirectionnelle entre les objets qui transporte de l’information avec l’intention de déclencher une activité chez le récepteur.

Les messages répertoriés entre le système et ses acteurs sont :

**Message émis par le système** :

* La liste des étudiants inscrit par niveau/filière.
* La liste des étudiants ayant acquis leurs diplômes.
* Les relevés de notes des étudiants.
* Les bulletins de note des étudiants.
* La liste des étudiants passés et recalés à la fin d’une année scolaire.
* La liste des étudiants en session de rattrapage à la fin d’un semestre.
* La liste des absences des étudiants.
* Les fiches des enseignants.
* La situation académique des étudiants.
* Les notifications sur les mises à jour d’informations (notes, emplois du temps).
* Les informations de l’étudiant liées à l’année.
* La liste des enseignements (UE, module, filière, niveau).
* Les statistiques relatives aux informations enregistrées.
* Les états de paiements des étudiants.
* Les plannings des emplois du temps.
* La liste des évaluations.

**Message reçus par le système :**

* Les créations, modifications, suppressions de profils utilisateurs.
* Les créations, modifications, suppressions de dossiers des étudiants/enseignants.
* Les créations, modifications, des années scolaires.
* L’affectation des étudiants/enseignants à une classe par année scolaire.
* Les créations, modification, suppressions des enseignements (filière, modules UE, cycle, niveau, classe).
* Les paramétrages des calculs des notes.
* Les éditions des emplois du temps.
* Les éditions des absences.
* Les mises à jour des paiements des étudiants.
* Les ajouts, modifications, des notes des étudiants.
* Les ajouts, modifications, des évaluations.
* Les éditions des modèles de relevé de note, bulletin, attestations.

# Modélisation du contexte

A partir des informations obtenues lors des deux précédentes étapes, nous allons modéliser le contexte de notre application. Ceci va nous permettre dans un premier temps, de définir les fonctionnalités spécifiques de chaque acteur dans le système :

|  |  |
| --- | --- |
| Utilisateurs finaux | Description des besoins fonctionnels |
| Administrateur | Cet utilisateur a accès à toutes les fonctionnalités de bases et avancées de l’application.  L’application doit permettre à l’administrateur de :   * S’authentifier * Créer les profils utilisateurs * Donner des droits d’accès. |
| Enseignant | L’application doit permettre à l’enseignant de :   * S’authentifier * Affecter les notes de ses étudiants * Consulter son emploi de temps * Consulter son profil * Poster et télécharger des documents (devoirs, exercices, corrections, examens, TP, TD…) |
| Personnel administratif | L’application doit permettre au personnel administratif de :   * S’authentifier * Consulter son profil * Créer/modifier/Supprimer les dossiers des étudiants * Créer/modifier/Supprimer les dossiers des enseignants * Paramétrer le système * Affecter les notes des étudiants * Saisir les absences des étudiants/enseignants * Notifier les enseignants n’ayant pas saisie leurs notes par mail/sms * Créer/modifier/Editer les emplois de temps * Poster/télécharger/supprimer des documents (devoirs, exercices, corrections, examens, TP, TD…) * Editer les bulletins * Editer les attestations * Gérer les années scolaires * Créer les cycles et filières * Créer les UE et les Modules * Créer les classes en fonction des années scolaire * Suivre les années scolaires en cours * Affecter les étudiants/enseignants à une classe * Consulter/Editer les états de paiement des étudiants |
| Etudiant | L’application doit permettre aux étudiants de :   * S’authentifier * Consulter ses relevés de notes * Consulter ses absences * Consulter son emploi du temps * Consulter son profil * Consulter ses états de paiement * Télécharger des documents utiles (anciens devoirs, examens, TP, TD…) |

Table 3 : Description des besoins fonctionnels

Cette phase représente un point de vue fonctionnel de l’architecture système. Par le biais des cas d’utilisation, nous allons compléter la capture des besoins fonctionnels ébauchés durant l’étude préliminaire. La technique des cas d’utilisation est la pierre angulaire de cette étape. Chaque cas d’utilisation spécifie un comportement attendu du système comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

# CHAPITRE IV :

# CAPTURE DES BESOINS FONCTIONNELS

# Identification des cas d’utilisation

Un cas d’utilisation (« use case ») représente un ensemble de séquences d’actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Pour constituer les cas d’utilisation, il faut considérer l'intention fonctionnelle de l'acteur par rapport au système dans le cadre de l'émission ou de la réception de chaque message. En regroupant les intentions fonctionnelles en unités cohérentes, on obtient les cas d'utilisations.

|  |  |  |
| --- | --- | --- |
| Cas d’utilisation | Acteurs | Messages émus/reçus |
| Gérer les profils | Administrateur | **Emet :**   * Création, modification, suppression de profils. * Attribution de droit   **Reçoit :**   * Confirmation * Information sur profils et droits |
| Gérer emplois du temps | Personnel administratif | **Emet :**   * Mises à jour des emplois de temps * Affections des modules aux enseignants   **Reçoit :**   * Confirmation * Information sur les emplois de temps |
| Consulter l’emploi temps | Etudiants/Professeur | **Reçoit :**   * Informations sur l’emploi du temps. |
| Télécharger des documents | Etudiant/Enseignant | **Reçoit :**   * Liste des documents à télécharger |
| Gérer la bibliothèque en ligne | Personnel administratif/Enseignant | **Emet :**   * Mises à jour des documents   **Reçoit :**   * Confirmation * Liste des documents |
| Maintenir les notes | Personnel administratif/Enseignant | **Emet :**   * Mises à jour des notes * Gérer session de rattrapage * Délibération des notes   **Reçoit :**   * Confirmation des mises à jour * Relevé de notes |
| Organiser et créer les évaluations | Personnel administratif/Enseignant | **Emet :**   * Mises à jour des évaluations   **Reçoit :**   * Confirmation des mises à jour des évaluations * Notes étudiants par évaluation |
| Consulter les notes | Etudiant | **Reçoit :**   * Relevé de notes * Bulletin de notes |
| Gérer les étudiants | Personnel administratif | **Emet :**   * Enregistrements d’un étudiant * Mises à jour des dossiers des étudiants   **Reçoit :**   * Confirmation * Situations académiques des étudiants * Liste des étudiants (par promotion, par niveau…) |
| Gérer la comptabilité | Comptable | **Emet :** |
| Gérer les filières | Personnel administratif | **Emet :**   * Ajout/Mises à jour des filières par cycle   **Reçoit :**   * Confirmation * Informations sur les filières |
| Gérer les UE | Personnel administratif | **Emet :**   * Ajout/Mises à jour des UE par filière   **Reçoit :**   * Confirmation * Informations sur les UE |
| Gérer les Modules | Personnel administratif | **Emet :**   * Ajout/Mises à jour des modules par UE   **Reçoit :**   * Confirmation * Informations sur les modules |
| Gérer les classes | Personnel administratif | **Emet :**   * Ajout/Mises à jour des classes d’enseignements par niveau et filières   **Reçoit :**   * Confirmation * Informations sur les classes d’enseignements |
| Gérer les absences | Personnel administratif/Enseignant | **Reçoit :**   * Enregistrements des absences * Edition des absences   **Emet :**   * Confirmation * Historiques des absences |
| Consulter ses absences | Enseignant/Etudiant | **Reçoit :**   * Informations sur ses absences |
| Gérer les années scolaires | Personnel administratif | **Emet :**   * Création/Edition d’une année scolaire * Affectation des classes d’enseignements pour une année scolaire donnée   **Reçoit :**   * Confirmation * Liste des classes par année scolaire |
| Gérer les classes par année scolaire | Personnel administratif | **Emet :**   * Affectation des étudiants à une classe d’enseignement par année scolaire * Affecter les enseignants à une classe d’enseignement par année scolaire   **Reçoit :**   * Liste des étudiants passés et recalés * Liste des étudiants par classe d’enseignement |
| Gérer les enseignants | Personnel administratif | **Emet :**   * Enregistrements d’un enseignant * Mises à jour des dossiers des enseignants   **Reçoit :**   * Confirmation * Liste des enseignants par compétence |
| S’authentifier | Tous les acteurs | **Emet :**   * Nom d’utilisateur/Mot de passe   **Reçoit :**   * Confirmation |

Table 4 :Liste des cas d’utilisation

# Structuration des cas d’utilisations en packages

Les besoins variés des acteurs et le nombre de fonctionnalités dont le futur logiciel devra disposer nous semble assez souvent compliqués. Pour mieux s’y prendre et nous faciliter la tâche, on peut découper le futur logiciel en parties distinctes, en fonction des « familles » de fonctionnalités de façon à pouvoir les analyser séparément. Chacune de ces parties correspond à un domaine fonctionnel ou package.

Pour définir la stratégie de regroupement des cas d’utilisation pour un projet, trois critères de regroupements sont souvent utilisés :

* Par domaine d'expertise métier : le plus intuitif et souvent le plus efficace. Le but étant de trouver des parties qui regroupent des fonctionnalités qui ont un lien ou qui font partie d’une même « famille » et que l’on peut analyser séparément ;
* Par acteur: simple à mettre en œuvre uniquement si chaque cas d'utilisation est  
  relié à un et un seul acteur, sinon il s'apparente souvent au critère précédent;
* Par lot de livraison : dans le cadre d’un développement itératif et incrémental, il est intéressant de regrouper dans un même package les cas d’utilisations qui seront livrés ensemble au client.

Pour mieux s’organiser le regroupement par domaine d'expertise métier sera utilisé pour notre projet.

|  |  |  |
| --- | --- | --- |
| Cas d’utilisation | Acteurs | Package |
| Télécharger des documents | Etudiant/Enseignant | **Gestion de la bibliothèque** |
| Gérer la bibliothèque en ligne | Personnel administratif/Enseignant |
| Gérer les enseignants | Personnel administratif | **Gestion administrative** |
| Gérer les étudiants | Personnel administratif |
| Gérer la comptabilité | Personnel administratif |
| Gérer les filières | Personnel administratif | **Gestion des enseignements** |
| Gérer les UE |
| Gérer les modules |
| Gérer les classes |
| Organiser et créer les évaluations | Personnel administratif/Enseignant | **Gestion des notes** |
| Maintenir les notes | Personnel administratif/Enseignant |
| Consulter les notes | Etudiant |
| Gérer les absences | Personnel administratif/Enseignant | **Gestion des absences et emplois du temps** |
| Consulter ses absences | Etudiant, Enseignant |
| Gérer les emplois du temps | Personnel administratif |
| Consulter les emplois du temps | Etudiant, Enseignant |
| Gérer les années scolaires | Personnel administratif | **Gestion des promotions** |
| Gérer les classes par année scolaire |
| Gérer les profils | Administrateur | **Gestion des droits d’accès** |
| S’authentifier | Tous les acteurs | **Authentification** |

Table 5 : Liste des cas d’utilisation par package

Diagramme de package :

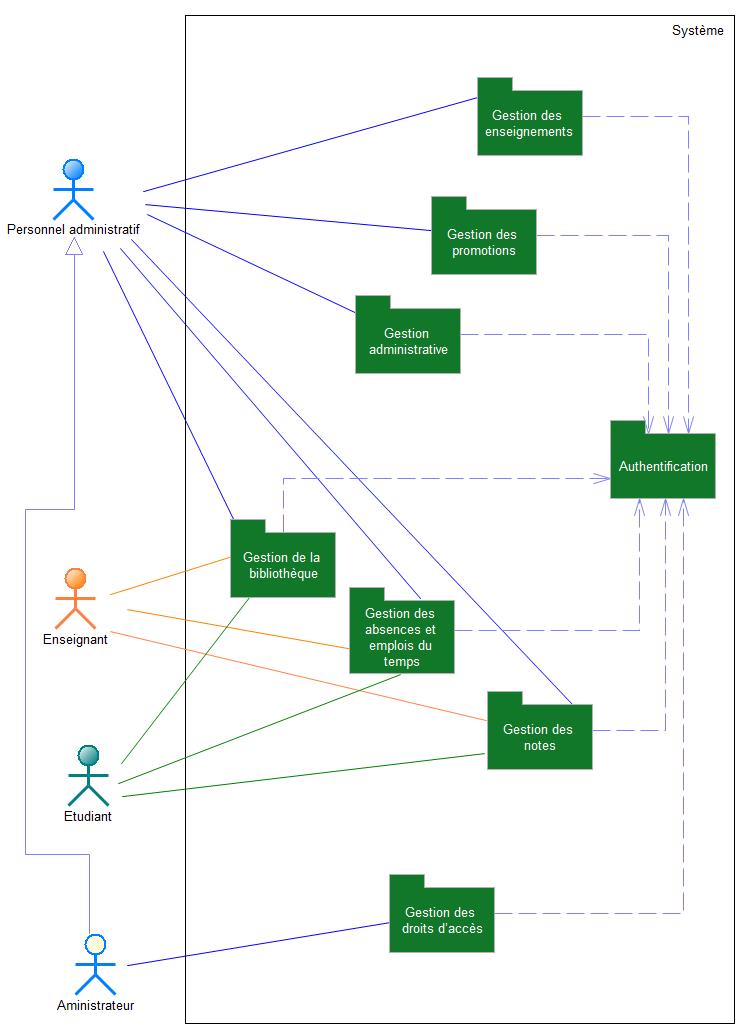


Figure 6 : Diagramme de packages

#### Diagramme de cas d’utilisation, package « gestion de la bibliothèque » :

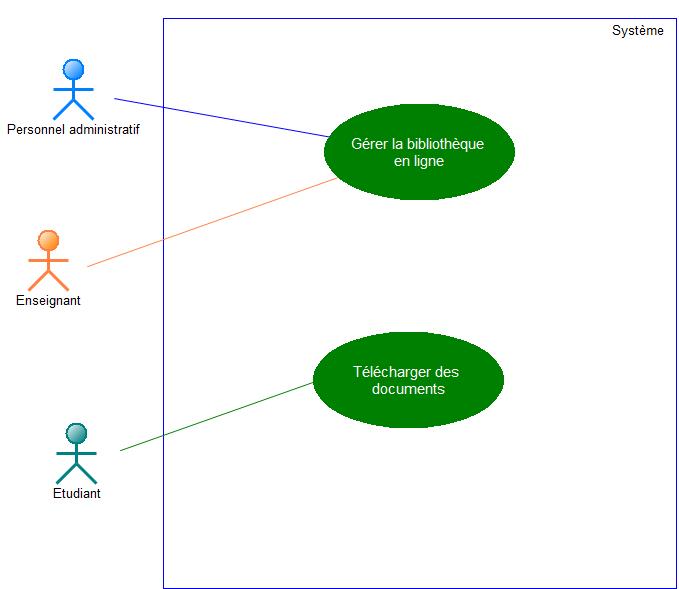


Figure 7 : Diagramme de cas d’utilisation, package « gestion de la bibliothèque »

#### Diagramme de cas d’utilisation, package « gestion administrative » :

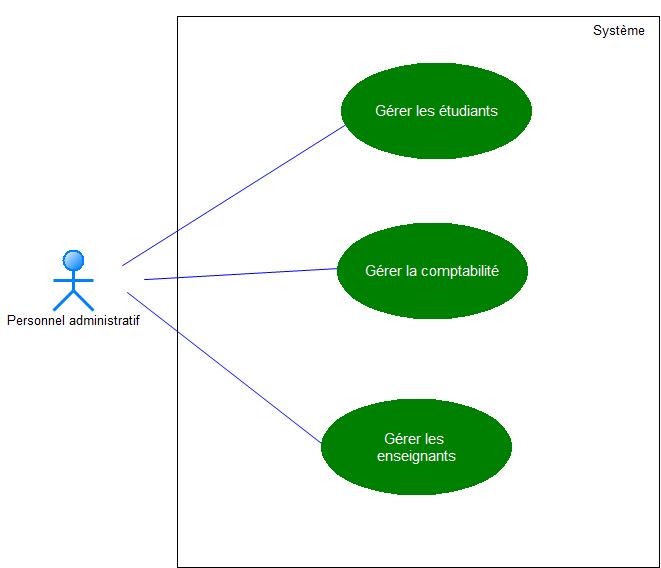


Figure 8 : Diagramme de cas d’utilisation, package « gestion administrative »

#### Diagramme de cas d’utilisation, package « gestion des enseignements » :

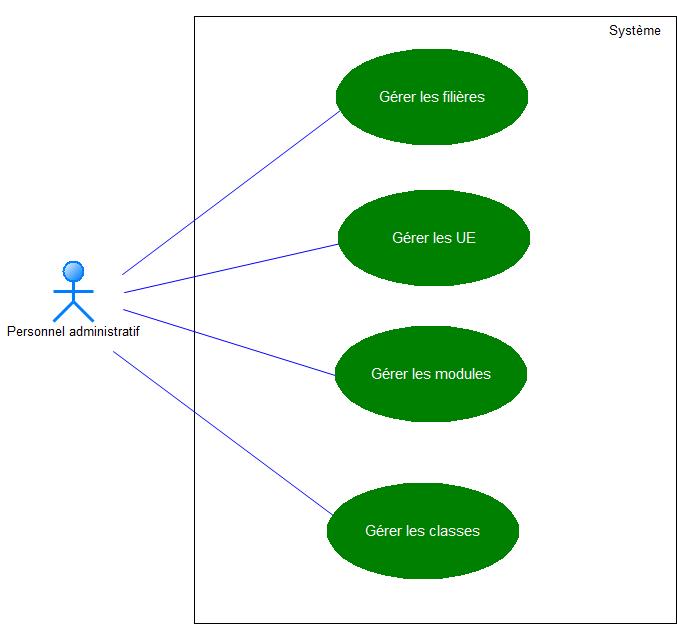


Figure 9: Diagramme de cas d’utilisation, package « gestion des enseignements »

#### Diagramme de cas d’utilisation, package « gestion des promotions » :

Figure 10 : Diagramme de cas d’utilisation, package « gestion des promotions »

C:\Users\hp\Desktop\casPGPR.emf

#### Diagramme de cas d’utilisation, package « gestion des notes » :

Figure 11 : Diagramme de cas d’utilisation, package « gestion des notes »

C:\Users\hp\Desktop\package gestion des notes.emf

#### Diagramme de cas d’utilisation, package « Gestion des absences/emplois du temps »

C:\Users\hp\Desktop\casPGAE.emf

Figure 12 : Diagramme de cas d’utilisation, package «Gestion des absences/emplois du temps»

#### Diagramme de cas d’utilisation, package « Gestion des droits d’accès » :

C:\Users\hp\Desktop\casPGDA.emf

Figure 13 : Diagramme de cas d’utilisation, package « Gestion des droits d’accès »

# Description détaillée des cas d’utilisation

Les diagrammes réalisés jusqu’à maintenant (diagramme de contexte, diagramme de packages, diagramme de cas d’utilisation) nous ont permis de découvrir petit à petit les fonctionnalités (appelées aussi des cas d’utilisation) que l’on devrait avoir dans le futur logiciel.

Nous allons maintenant détailler les cas d’utilisations qui doivent faire l’objet d’une définition a priori qui décrit l’intention de l’acteur lorsqu’il utilise le système et les séquences d’actions principales qu’il est susceptible d’effectuer. Ces définitions servent à fixer les idées et n’ont pas pour but de spécifier un fonctionnement complet et irréversible.

Pour exprimer les cas d’utilisations de notre système, nous avons choisi le formalisme suivant :

|  |  |
| --- | --- |
| Titre | Le titre du cas d’utilisation |
| Acteur | Acteurs participants au cas d’utilisation |
| Description | But du cas d’utilisation |
| Description des enchainements | |
| Préconditions | Condition qui doit être remplie avant le début du cas d’utilisation |
| Scenario nominal  Séquence d’actions normales associées au cas d’utilisation | |
| Scenario alternatif  Séquence d’actions alternatives. C’est le cas des étapes liées à des conditions. | |

Table 6 : Formalisme de description d’un cas d’utilisation

Remarque : Plusieurs diagrammes vont apparaitre (mais pas nécessairement) pour apporter une compréhension additive au cas d’utilisation à savoir :

* Le diagramme de séquence : permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.
* Le diagramme d’activités : représente le déroulement des actions, sans utiliser les objets. La fiche descriptive d’un cas d’utilisation peut contenir plusieurs scénarios alternatifs et/ou d’exception. Il est alors difficile d’avoir une vision de l’ensemble des actions. Le diagramme d’activité est un moyen graphique pour donner cette vision d’ensemble. Ainsi pour les cas d’utilisation les plus complexes, un diagramme d’activité peut aider à y voir un peu plus claire.

#### Cas d’utilisation : Gérer les filières, package « Gestion des enseignements »

C:\Users\hp\Desktop\casGF.emf

Figure 14 : Cas d’utilisation gérer les filières, package « Gestion des enseignements »

|  |  |
| --- | --- |
| Titre | Gérer les filières |
| Acteur | Personnel administratif |
| Description | Le système doit permettre de gérer et d’organiser les filières à savoir l’ajout, la modification, la recherche et la suppression des filières |
| Description des enchainements | |
| Préconditions | Le personnel administratif doit être authentifié (cas d’utilisation : s’authentifié – package : authentification) et avoir la permission d’effectuer l’opération |
| Scenario nominal   1. Le personnel administratif choisit l’option « Gestion des enseignements » du menu général et choisit l’élément « filière » 2. Le système affiche une page contenant la liste des filières, accompagnée des opérations « ajouter », « éditer » 3. Si le personnel administratif choisit l’option d’ajouter 4. Le système affiche une page contenant un formulaire pour saisir les informations d’une filière 5. Le personnel administratif remplit les champs du formulaire et valide l’opération d’ajout 6. Le système ajoute les données dans la base et affiche un message de validation. 7. Si le personnel administratif choisit l’option d’éditer 8. Le système affiche une page contenant les données de la filière, et aussi une liste des UE qui la compose 9. Le personnel administratif peut supprimer, modifier la filière 10. Si le personnel administratif choisit de supprimer la filière 11. Le système effectue l’opération, et affiche une notification, la suppression a été effectuer avec succès. 12. Si le personnel administratif choisit de modifier la filière 13. Le système affiche un formulaire contenant les informations de la filière 14. Le personnel administratif apporte les modifications et valide l’opération 15. Le système enregistre les changements effectués et affiche un message de validation | |
| Scenario alternatif   1. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de la sélection des données 2. Le personnel administratif décide de quitter l’ajout. Le scenario reprend au point 2 du scenario nominal. 3. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 4 du scenario nominal. 4. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de l'ajout des données. Le scenario reprend au point 4 du scenario nominal. 5. Le personnel administratif décide de quitter l’édition. Le scenario reprend au point 2 du scenario nominal. 6. Le personnel administratif décide de quitter la modification. Le scenario reprend au point 8 du scenario nominal. 7. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 13 du scenario nominal. 8. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de la modification des données. Le scenario reprend au point 13 du scenario nominal. | |

Table 7 : Description détaillé, cas d’utilisation gérer les filières, package « Gestion des enseignements »

C:\Users\hp\Desktop\activitygf.emf

Figure 15:Diagramme d’activité gérer les filières, package « Gestion des enseignements »

C:\Users\hp\Desktop\sequenceGF.emf

Figure 16: Diagramme de séquence gérer les filières, package « Gestion des enseignements »

#### Cas d’utilisation : Gérer les classes, package « Gestion des enseignements »

C:\Users\hp\Desktop\casGC.emf

Figure 17 : cas d’utilisation : Gérer les classes, package « Gestion des enseignements »

|  |  |
| --- | --- |
| Titre | Gérer les classes |
| Acteur | Personnel administratif |
| Description | Le système doit permettre d’ajouter, de modifier de supprimer des classes/groupes dans le but de gérer les étudiants sous forme de groupe (classe). |
| Description des enchainements | |
| Préconditions | Le personnel administratif doit être authentifié (cas d’utilisation : s’authentifié – package : authentification) et avoir la permission d’effectuer l’opération |
| Scenario nominal   1. S’il choisit l’option « Gestion des enseignements » du menu général et choisit l’élément « classes » 2. Le système affiche une page contenant la liste des classes, accompagnée des options « ajouter », « modifier », « supprimer » 3. S’il choisit l’option d’ajouter 4. Le système affiche une page contenant un formulaire pour saisir les informations de la classe 5. Le personnel administratif remplit les champs du formulaire et valide l’opération d’ajout 6. Le système ajoute les données dans la base et affiche un message de validation. 7. S’il choisit de supprimer une classe 8. Le système effectue l’opération, et affiche une notification, la suppression a été effectuer avec succès. 9. S’il choisit de modifier une classe 10. Le système affiche un formulaire contenant les informations de la classe 11. L’utilisateur apporte les modifications et valide l’opération 12. Le système enregistre les changements effectués et affiche un message de validation | |
| Scenario alternatif   1. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de la sélection des données 2. Le personnel administratif décide de quitter l’ajout. Le scenario reprend au point 2 du scenario nominal. 3. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 4 du scenario nominal. 4. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de l'ajout des données. Le scenario reprend au point 4 du scenario nominal. 5. Le personnel administratif décide de quitter la modification. Le scenario reprend au point 2 du scenario nominal. 6. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 10 du scenario nominal. 7. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de la modification des données. Le scenario reprend au point 10 du scenario nominal. | |

Table 8 : Description détaillé, cas d’utilisation gérer les classes, package « Gestion des enseignements »

Figure 18 : Diagramme d’activité gérer les classes, package « Gestion des enseignements »

C:\Users\hp\Desktop\activitygc.emf

C:\Users\hp\Desktop\sequenceGC.emf

Figure 19 : Diagramme de séquence gérer les classes, package « Gestion des enseignements »

#### Cas d’utilisation : Gérer les années scolaires, package « Gestion des promotions »

C:\Users\hp\Desktop\casGestionAnneeScolaire.emf

Figure 20 : Cas d’utilisation : gérer les années scolaires, package « Gestion des promotions »

|  |  |
| --- | --- |
| Titre | Gérer les années scolaires |
| Acteurs | Personnel administratif |
| Description | Le système doit permettre d’organiser les années en affectant les classes/groupes d’enseignements pour une année scolaire donnée. |
| Description des enchainements | |
| Préconditions | L’utilisateur (personnel administratif, enseignant) doit être authentifié (cas d’utilisation : s’authentifié – package : authentification) et avoir les permissions nécessaires pour effectuer les opérations |
| Scenario nominal   1. L’utilisateur choisit l’option « Année scolaire » du menu général 2. Le système affiche une page contenant la liste des années scolaires suivis des options d’opération « ajouter », « organiser », « supprimer » 3. Si l’utilisateur choisit d’ajouter une année scolaire 4. Le système affiche une page contenant un formulaire pour saisir les informations de l’année scolaire 5. Le personnel administratif remplit les champs du formulaire et valide l’opération d’ajout 6. Le système ajoute les données dans la base et affiche un message de validation. 7. Si le personnel administratif choisit de supprimer une année scolaire 8. Le système effectue l’opération, et affiche une notification, la suppression a été effectuer avec succès. 9. Si le personnel administratif choisit l’option d’organiser une année scolaire 10. Le système affiche une page contenant la liste des classes affectées à l’année scolaire 11. Le personnel administratif peut lui affecter une classe, retirer une classe 12. Si le personnel administratif choisit de lui retirer une classe 13. Le système effectue l’opération, et affiche une notification, l’opération a été effectuer avec succès. 14. Si le personnel administratif choisit de lui affecter une classe 15. Le système affiche un formulaire pour saisir les informations de la classe (date début et date fin des enseignements de la classe pour l’année) 16. Le personnel administratif remplit les champs du formulaire et valide l’opération. 17. Le système enregistre les données et affiche un message de validation | |
| Scenario alternatif   1. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de la sélection des données 2. Le personnel administratif décide de quitter l’ajout. Le scenario reprend au point 2 du scenario nominal. 3. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 4 du scenario nominal. 4. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de l'ajout des données. Le scenario reprend au point 4 du scenario nominal. 5. Le personnel administratif décide de quitter l’organisation. Le scenario reprend au point 2 du scenario nominal. 6. Le personnel administratif décide d’annuler l’opération d’affectation. Le scenario reprend au point 10 du scenario nominal. 7. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 15 du scenario nominal. 8. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de l’opération. Le scenario reprend au point 15 du scenario nominal. | |

Table 9 : Description détaillé, cas d’utilisation gérer les années scolaires, package « Gestion des promotions »

C:\Users\hp\Desktop\sequenceGAS.emf

Figure 21 : Diagramme de séquence gérer les années scolaires, package « Gestion des promotions»

#### Cas d’utilisation : Maintenir les notes, package « Gestion des notes »

Figure 22 : Cas d’utilisation maintenir les notes, package « Gestion des notes »

C:\Users\hp\Desktop\cas maintenir note.emf

|  |  |
| --- | --- |
| Titre | Maintenir les notes |
| Auteur | Personnel administratif, enseignant |
| Description | Le système doit permettre au personnel administratif (ou enseignant) d’attribuer des notes aux étudiants. |
| Description des enchainements | |
| Préconditions | L’utilisateur (personnel administratif, enseignant) doit être authentifié (cas d’utilisation : s’authentifié – package : authentification) et doit avoir les permissions lui permettant d’effectuer l’opération. |
| Scenario nominal   1. Ce cas d’utilisation commence lorsque l’utilisateur (personnel administratif, enseignant) choisit d’attribuer les notes des étudiants suite à une évaluation. 2. Le système affiche une page contenant la liste des évaluations. 3. L’utilisateur choisit l’évaluation. 4. Le système affiche la liste des étudiants. 5. L’utilisateur affecte les notes des étudiants puis valide l’opération. 6. Le système ajoute les données dans la base et affiche un message de validation | |
| Scenario alternatif   1. L’utilisateur décide de quitter l’opération. 2. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 4 du scenario nominal. 3. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de l'ajout des données. Le scenario reprend au point 4 du scenario nominal. | |

Table 10 : Description détaillée, cas d’utilisation maintenir les notes, package « Gestion des notes »

C:\Users\hp\Desktop\sequence maintenir notes.emf

Figure 23 : Diagramme de séquence maintenir les notes, package « Gestion des notes»

#### Cas d’utilisation : Gérer les étudiants, package « Gestion administrative »

|  |  |
| --- | --- |
| Titre | Gérer les étudiants |
| Auteur(s) | Personnel administratif |
| Description | Le système doit permettre d’enregistrer les informations d’un étudiant, y compris les informations son bac, ses contact d’urgence... Le système doit aussi permettre de suivre et mètre a jours leurs dossier scolaire. |
| Description des enchainements | |
| Préconditions | Le personnel administratif doit être authentifié (cas d’utilisation : s’authentifié – package : authentification) et avoir les permissions pour pouvoir effectuer les opérations |
| Scenario nominal   1. L’utilisateur choisit l’option « Gestion des étudiants » du menu général 2. Le système affiche une page contenant la liste des étudiants, accompagnée des options d’opération « ajouter », « voir profil », « supprimer » 3. Si l’utilisateur choisit d’ajouter un étudiant 4. Le système affiche une page contenant un formulaire pour saisir les informations de l’étudiant. 5. L’utilisateur remplit les champs du formulaire, et valide l’opération 6. Le système ajoute les données dans la base et affiche un message de validation 7. Si l’utilisateur choisit de consulter le profil d’un étudiant 8. Le système envoie une page contenant les informations (profil, UE obtenue, parcours de formations, situation académique…) sur l’étudiant 9. Le personnel administratif peut éventuellement décider de mettre à jour les informations relatives à l’étudiant. 10. Le système affiche un formulaire contenant les informations à modifier 11. Le personnel administratif effectue les modifications et valide l’opération. 12. Le système met à jour les données et affiche un message de confirmation 13. Si le personnel administratif choisit de supprimer un étudiant 14. Le système effectue l’opération, et affiche une notification, l’opération a été effectuer avec succès. | |
| Scenario alternatif   1. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de la sélection des données 2. Le personnel administratif décide de quitter l’ajout. Le scenario reprend au point 2 du scenario nominal. 3. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 4 du scenario nominal. 4. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de l'ajout des données. Le scenario reprend au point 4 du scenario nominal. 5. Le personnel administratif décide de retourner en arrière. Le scenario reprend au point 2 du scenario nominal. 6. Le personnel administratif décide d’annuler l’opération de modification. Le scenario reprend au point 8 du scenario nominal. 7. Si les champs requis ne sont pas remplis ou invalide, le système affiche une notification, veuillez bien renseigner tous les champs du formulaire. Le scenario reprend au point 10 du scenario nominal. 8. En cas d'erreur, le système notifie l'agent que la connexion a échoué ou une erreur est survenue sur la base de données lors de l’opération. Le scenario reprend au point 10 du scenario nominal. | |

Table 11 : Description détaillée, cas d’utilisation :gérer les étudiants , package « Gestion administrative »

C:\Users\hp\Desktop\activityGEtudiant.emf

Figure 24 : Diagramme d’activité gérer les étudiants, package « Gestion Administrative »

# Identification des classes candidates

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d’analyse.

La technique utilisée pour identifier les classes candidates consiste à chercher les noms communs importants dans les descriptions textuelles des cas d’utilisation regroupés par package.

#### Diagramme de classe participante des cas d’utilisation, package « Gestion des enseignements »

Les classes candidates tirées de la description détaillée des cas d’utilisation du package « gestion des enseignements » sont : Cycle, Filiere, Niveau, Classe, UE, TypeUE, Module, Categorie.

C:\Users\hp\Desktop\classeGestionEnseignment.emf

Figure 25 : Diagramme de classe participante des cas d’utilisation, package « gestion des enseignements »

#### Diagramme de classe participante des cas d’utilisation, package « Gestion des promotions »

Les classes candidates tirées de la description détaillée des cas d’utilisation du package « gestion des promotions » sont : AnneeScolaire, Enseignant, Etudiant, EnseignementAnnee, Classe, Module, ClasseAnnee.

C:\Users\hp\Desktop\classeGestonPromo.emf

Figure 26: Diagramme de classe participante des cas d’utilisation, package « Gestion des promotions »

#### Diagramme de classe participante des cas d’utilisation, package « Gestion des absences/emplois du temps »

Les classes candidates tirées de la description détaillée des cas d’utilisation du package « Gestion des absences/emplois du temps » sont : AnneeScolaire, Categorie, Etudiant, Horaire, Classe, Module, ClasseAnnee. ModuleCategorie, Salle.

C:\Users\hp\Desktop\classe emplois absence.emf

Figure 27 : Diagramme de classe participante des cas d’utilisation, package « Gestion des absences/emplois du temps »

Ce chapitre va nous permettre d’illustrer les principales constructions du diagramme de classes UML durant l’étape d’analyse. Le diagramme de classes a toujours été le diagramme le plus important dans toutes les méthodes orientées objet. C’est également celui qui contient la plus grande gamme de notations et de variantes.

# CHAPITRE V :

# ANALYSE

# Diagramme de package

Cette phase marque le démarrage de l’analyse objet du système à réaliser. Elle utilise la notion de *package* pour définir des catégories de classes d’analyse et découper le modèle UML en blocs logiques les plus indépendants possibles. Il faut pour cela se baser sur les diagrammes de classes participantes obtenues lors de la capture des besoins fonctionnels.

Figure 28 : Diagramme de package

C:\Users\hp\Desktop\packageClasse.emf

**Package gestion des utilisateurs** : les classes qui constituent le package gestion des enseignements sont : Etudiant, Bac, Professeur, Enseignant, User, Role, Permission.

**Package gestion des évaluations** : il est constitué des classes : Evaluation, TypeEvaluation, RegimeEvaluation, ModaliteEvaluation, Notage.

**Package gestion de la scolarité** : il comporte les classes suivantes : Frais, Reglement, Inscription.

**Package gestion des enseignements** : ce package est constitué des classes : Niveau, Cycle, Filière, UE, Module, Categorie, ModuleCategorie, Classe, AnneeScolaire, ClasseAnnee, EnseignementAnnee, TypeUE

# Développement du modèle statique

Cette étape est très importante. Il va s'agir de raffiner les classes. Nous allons raffiner les attributs des classes.

Nous allons assigner la visibilité des attributs et opérations des classes : de façon générale, les attributs sont privés et les opérations publiques. Nous allons aussi procéder au typage des attributs et opérations.

* Package gestion des enseignements :

C:\Users\hp\Desktop\package gestion enseignement.emf

Figure 29:Diagramme de classe package « gestion des enseignements »

* Package gestion des emplois

C:\Users\hp\Desktop\package classe absence.emf

Figure 30 : Diagramme de classe package gestion des emplois

* Package gestion des utilisateurs

C:\Users\hp\Desktop\package classe utilisateur.emf

Figure 31 : Diagramme de classe package gestion des utilisateurs.

* Package gestion de la scolarité

C:\Users\hp\Desktop\package classe scolarite.emf

Figure 32 : Diagramme de package gestion de la scolarité.

* Package gestion des évaluations

C:\Users\hp\Desktop\package classe evaluation.emf

Figure 33: Diagramme de classe package gestion des évaluations

Les besoins techniques seront exprimés dans les lignes qui suivront. Les différentes Architectures techniques (matérielle, logicielle) seront évoquées.

# CHAPITRE VI :

# CAPTURE DES BESOINS TECHNIQUES

# Architecture Matérielle

L’expression des besoins techniques implique également le choix d’architecture. Ce choix est crucial puisqu’il intervient dans l’évolutivité du système.

## L’architecture 1 tiers

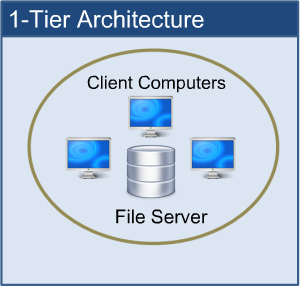
La conception de l’application est élaborée de manière à fonctionner sur un ordinateur unique. En fait, tous les services fournis par l'application résident sur la même machine et sont inclus dans l'application. Toutes les fonctionnalités sont donc comprises dans une seule couche logicielle.

Figure 34 : Architecture 1 tiers

## L’architecture 2 tiers

Appelée aussi architecture client lourd/serveur, elle est assez simple dans sa mise en œuvre. Ce type d'architecture est constitué uniquement de deux parties : le « client lourd » demandeur de service d’une part et le « serveur de données » qui fournit le service d'autre part. Nous aurons donc la base de données qui sera délocalisée sur un serveur dédié appelé le serveur de données qui fournira les données à exploiter.

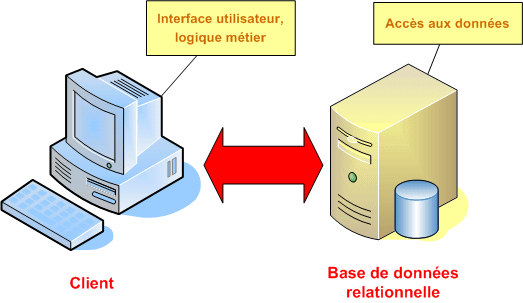


Figure 35 : Architecture 2 tiers

## L’architecture 3 tiers

Cette architecture physique est assez semblable à l’architecture client/serveur, mais en plus des « clients » et du serveur de données évoquées plus haut, un serveur d'application intervient comme un troisième tiers. En effet, les machines clientes également appelées « clients légers » ne contiennent que l'interface de l'application de manière qu’elles soient déchargées de tout traitement. En effet, le traitement est ainsi assuré par le serveur d'application, qui sert de liaison entre l'interface applicative et les données localisées au niveau du serveur de données.

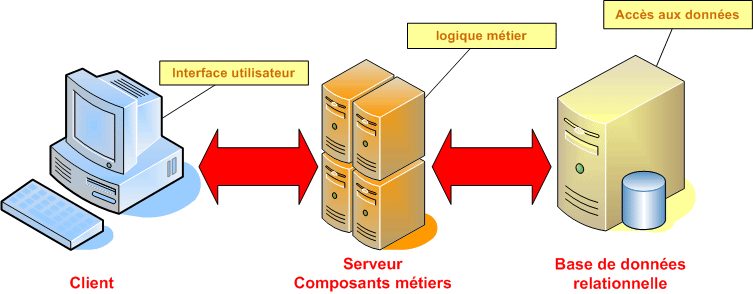


Figure 36 : Architecture 3 tiers

## Le choix de l’architecture

Pour nous, utiliser une architecture à 3 niveaux est beaucoup plus propre. Cela permet de diviser les tâches et par conséquent d’avoir des développeurs spécialisés sur un des trois niveaux. De plus, la flexibilité qu’offre ce genre d’infrastructure est à prendre en considération surtout si c’est un projet qui peut être amené à évoluer.

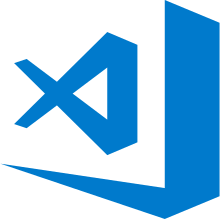
Ce chapitre sera consacré à présenter les outils utilisés pour réaliser notre application ainsi qu’un aperçu sur cette dernière.

# CHAPITRE VII :

# REALISATION

# Technologies et outils de développement

## Matériel technique

* **Visual Studio Code**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et OS X. Visual Studio Code est un nouvel outil qui combine la simplicité d'un éditeur de code avec ce dont les développeurs ont besoin pour le cycle edit-build-debug. Aller au-delà de la coloration syntaxique et de la saisie semi-automatique avec IntelliSense, qui fournit des complétions intelligentes basées sur des types de variables, des définitions de fonctions et des modules importés. Code de débogage à partir de l'éditeur. Lancez ou attachez à vos applications en cours d'exécution et déboguez avec des points d'arrêt, des piles d'appels et une console interactive.

* **PowerDesigner**

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. Il permet de réaliser tous les types de modèles qu’utilise UML.

* **Laragon**

Laragon est un environnement de développement moderne et puissant que d’innombrables personnes aiment utiliser chaque jour. Laragon est créé par Leo Khoa.Il a construit laragon avec un seul objectif : rendre le développement Web rapide, amusant et amusant**.** Laragon offre tout ce dont un développer à besoin pour créer des applications web modernes.

1. **Langages de programmation**

* **PHP**

Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

* **HTML**

L’HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C’est un langage permettant d’écrire de l’hypertexte, d’où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.

* **CSS**

****Les feuilles de style en cascade, généralement appelées CSS de l'[anglais](https://fr.wikipedia.org/wiki/Anglais) Cascading Style Sheets, forment un [langage informatique](https://fr.wikipedia.org/wiki/Langage_informatique) qui décrit la présentation des documents [HTML](https://fr.wikipedia.org/wiki/Hypertext_markup_language) et [XML](https://fr.wikipedia.org/wiki/Extensible_markup_language). CSS permet de gérer tout le style, le visuel d’une page web.

* **Jquery**

JQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. JQuery est la bibliothèque JavaScript la plus utilisée et permet de créer des effets dynamiques sur des pages web comme des changements de couleur, des animations, et des effets de fondu…

* **Booststrap**

Bootstrap est une collection d'outils utiles 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. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

## RÃ©sultat de recherche d'images pour "mysql"Système de gestion de base des données utilisé : MySQL

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. La première version de MySQL est apparue le 23 mai 1995.

## Framework de développement

### Pourquoi utiliser un Framework ?

Un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle, et accompagné du plan de l'architecture cible du Framework.

Les avantages des Frameworks sont nombreux. En effet, un Framework est portable, de la part de son abstraction de la base de données et de la gestion générique du cache. Un Framework permet le développement des applications sécurisées. Grâce aux systèmes d'authentification, à la gestion des injections SQL ainsi qu'à la protection CSRF (Cross-Site Request Forgery) qui est gérée par la plupart des Framework. Les Framework sont des outils communautaires et ont, par conséquent, des forums, des listes de diffusion et des canaux IRC pour les soutenir. De plus vu que les Framework sont largement déployés, la chance de trouver les correctifs des problèmes rencontrés est plus grande.

### Laravel

Comme Framework de développement nous avons utilisé Laravel. Laravel est un Framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub. Laravel, créé par Taylor Otwel, initie une nouvelle façon de concevoir un Framework en utilisant ce qui existe de mieux pour chaque fonctionnalité.

En Mars 2015, Laravel est considéré comme l’un des plus populaires Framework PHP.

# Quelques captures d’écrans

Dans cette partie, nous allons présenter quelques interfaces de l’application. Nous avons choisi l’administration comme utilisateur vu qu’il présente à travers ces interactions la majeure partie des principales fonctionnalités de l’application.

## Authentification

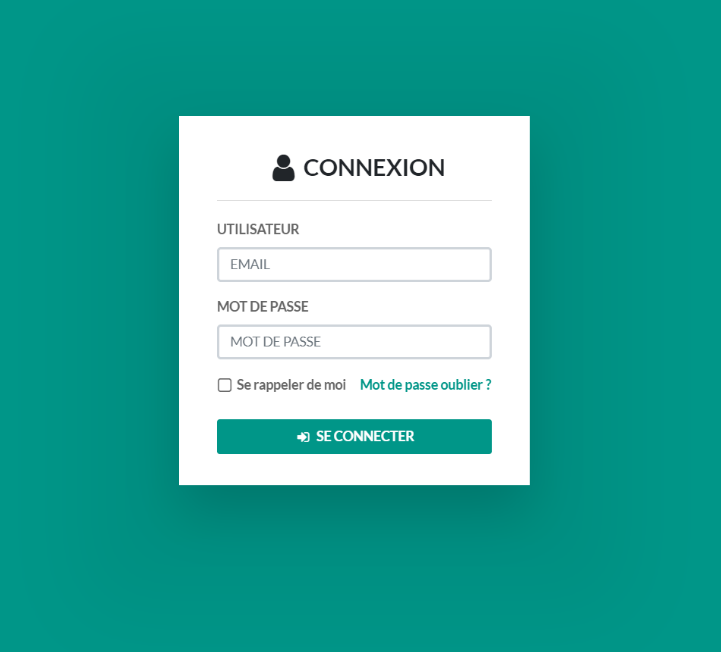
L’authentification (figure 36) est la première étape pour la sécurisation de l’application. En effet il y a souvent des parties de l’application qui ne doivent être accessibles qu'à certains utilisateurs. Ainsi la page d’authentification permet aux utilisateurs (personnel administratif, enseignant, étudiant, administrateur) de s’authentifier à travers un identifiant (e-mail) et un mot de passe qui est stocké sous forme cryptée dans la base de données.

Figure 37: Page d'authentification

En cas d’oubli de mot de passe, un mail sera envois à l’utilisateur lui permettant de changer son mot de passe.

Figure 38: Page mot de passe oublié

## Gestion des droits d’accès

Les rôles et les autorisations constituent une partie importante de l’application. Ainsi il est possible d’organiser dynamiquement les personnels administratifs en plusieurs type d’utilisateur (sécréteur, comptable, directeur des études…), voir figure 39, ayant des droits et autorisations spécifiques. Ainsi un personnel administratif ne peut effectuer les opérations donc il a les droits et permissions ; ce qui permet de rendre l’application plus flexible et plus dynamique.

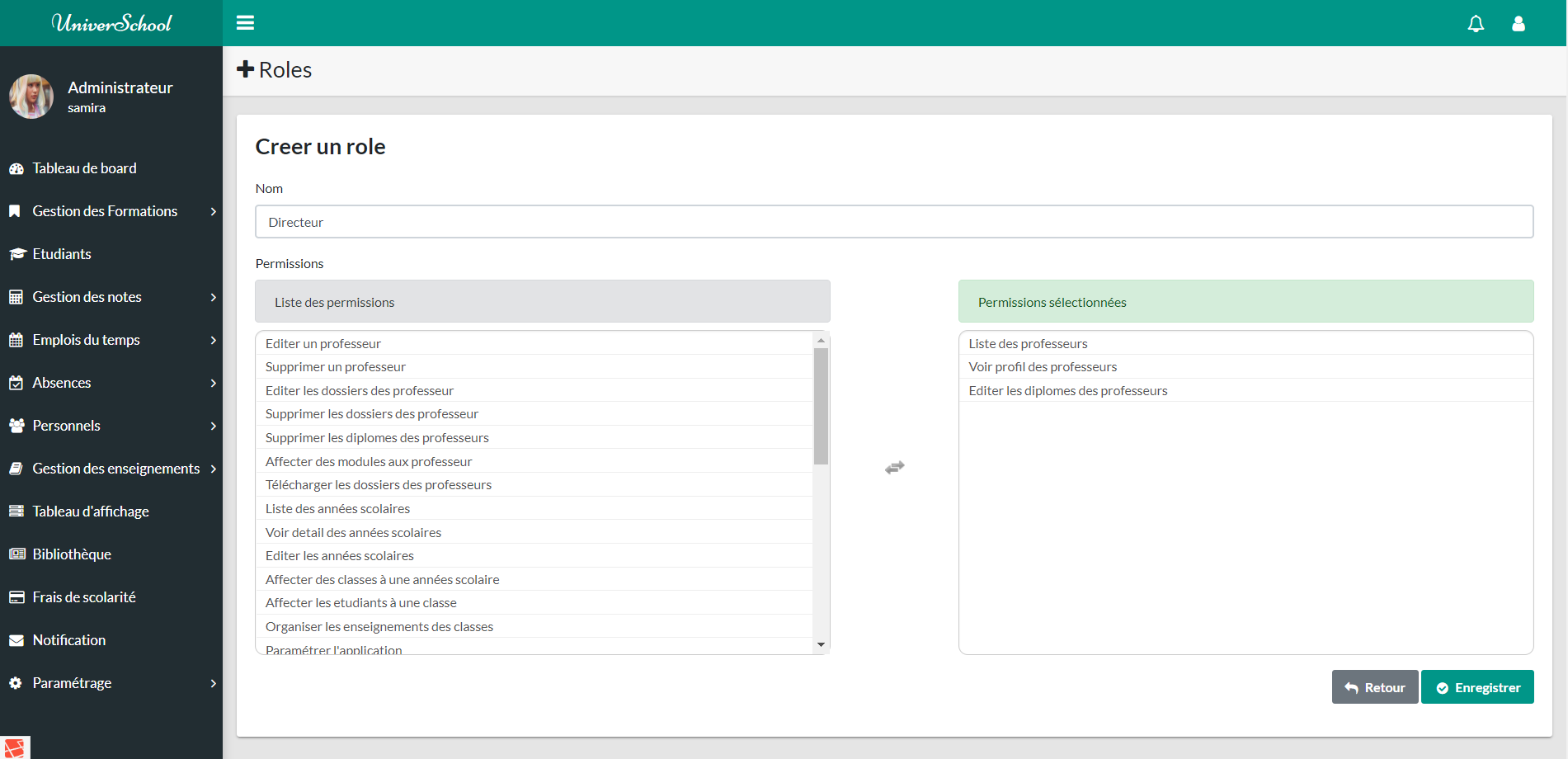


Figure 39 : Interface d’ajout d’un nouveau rôle

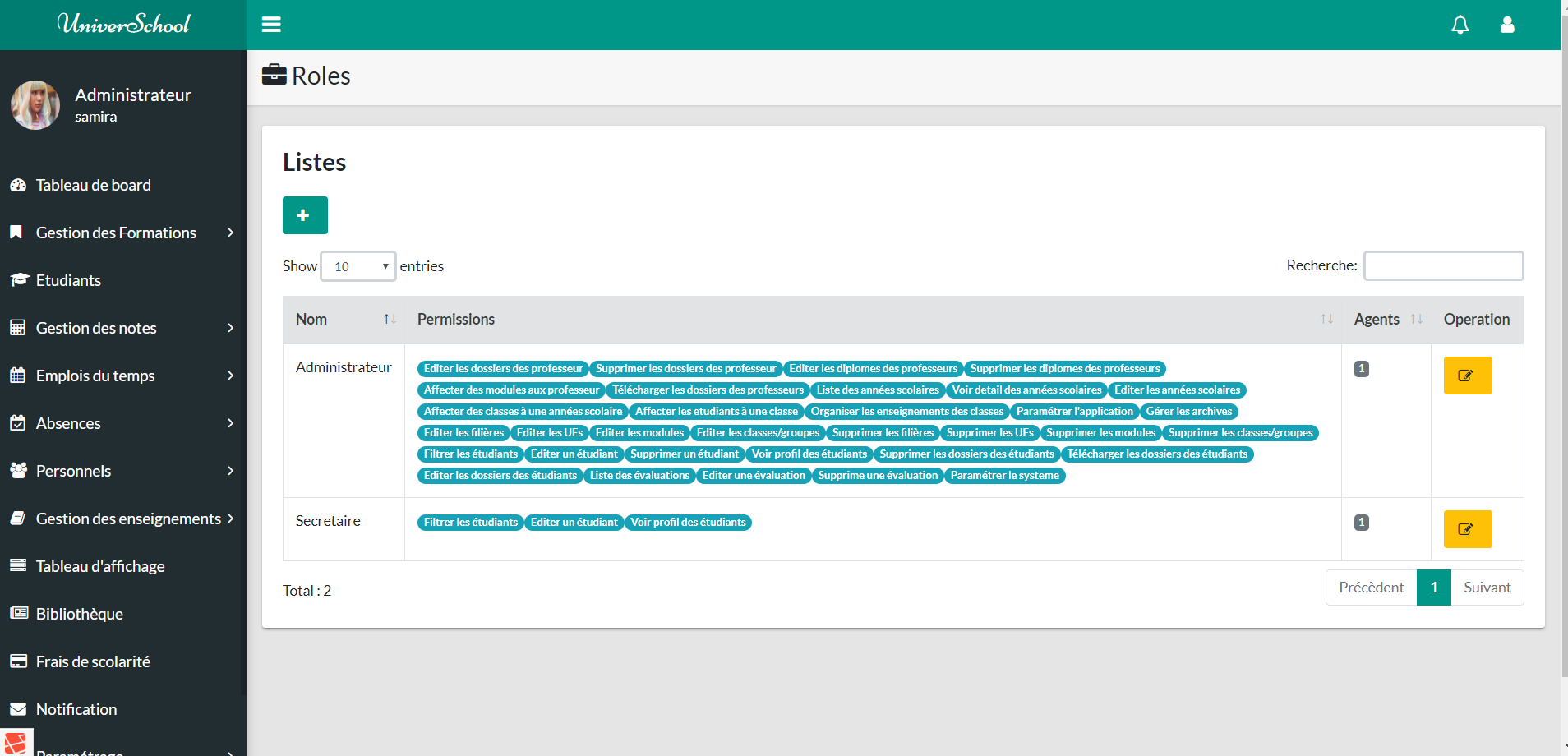


Figure 40 : Liste des rôles

## Profil des utilisateurs

Le profil permet de voir toute les informations concernant l’utilisateur (étudiant, enseignant, personnel administratif).

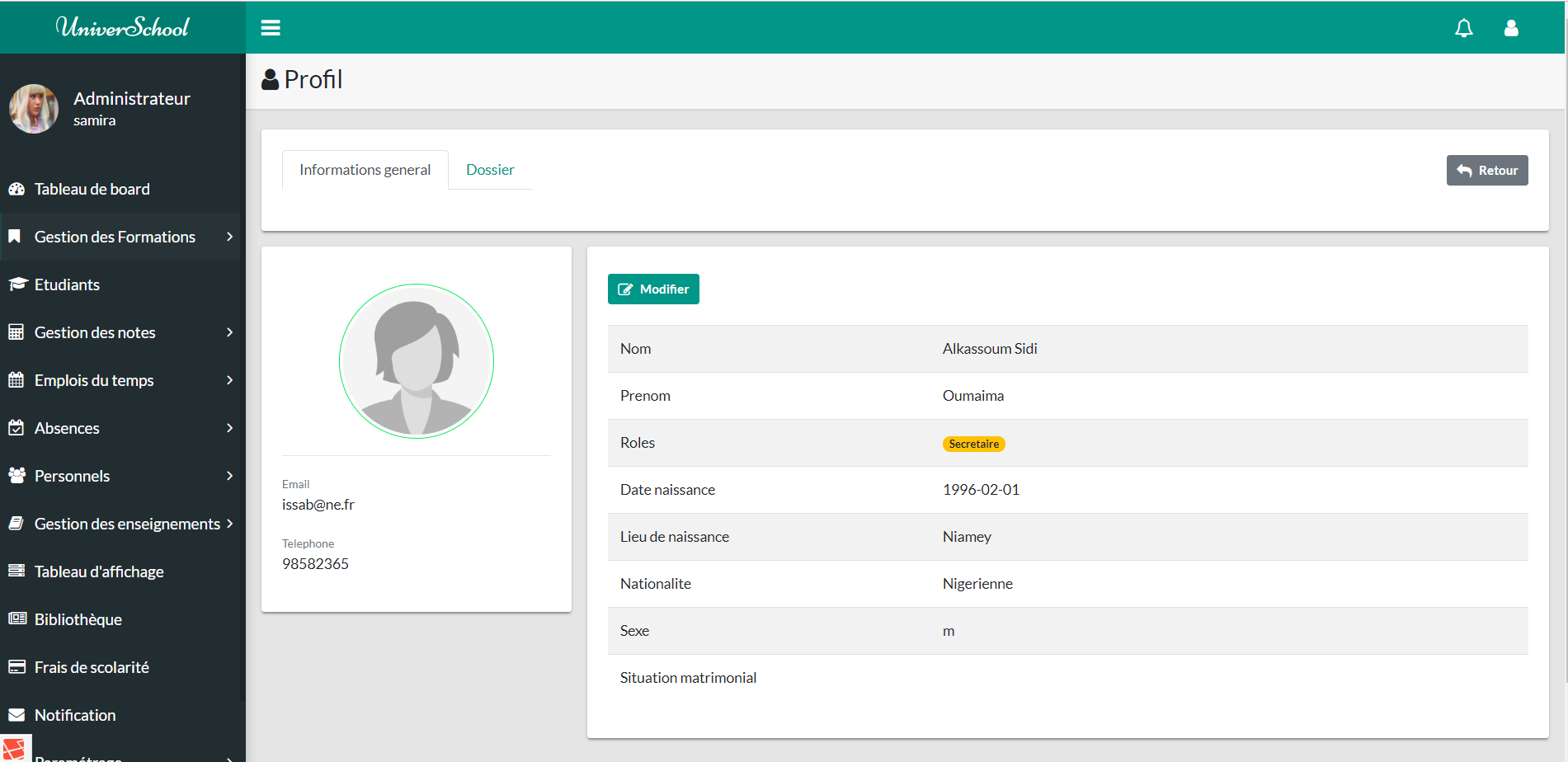


Figure 41 : Profil d’un personnel administratif (Secrétaire)

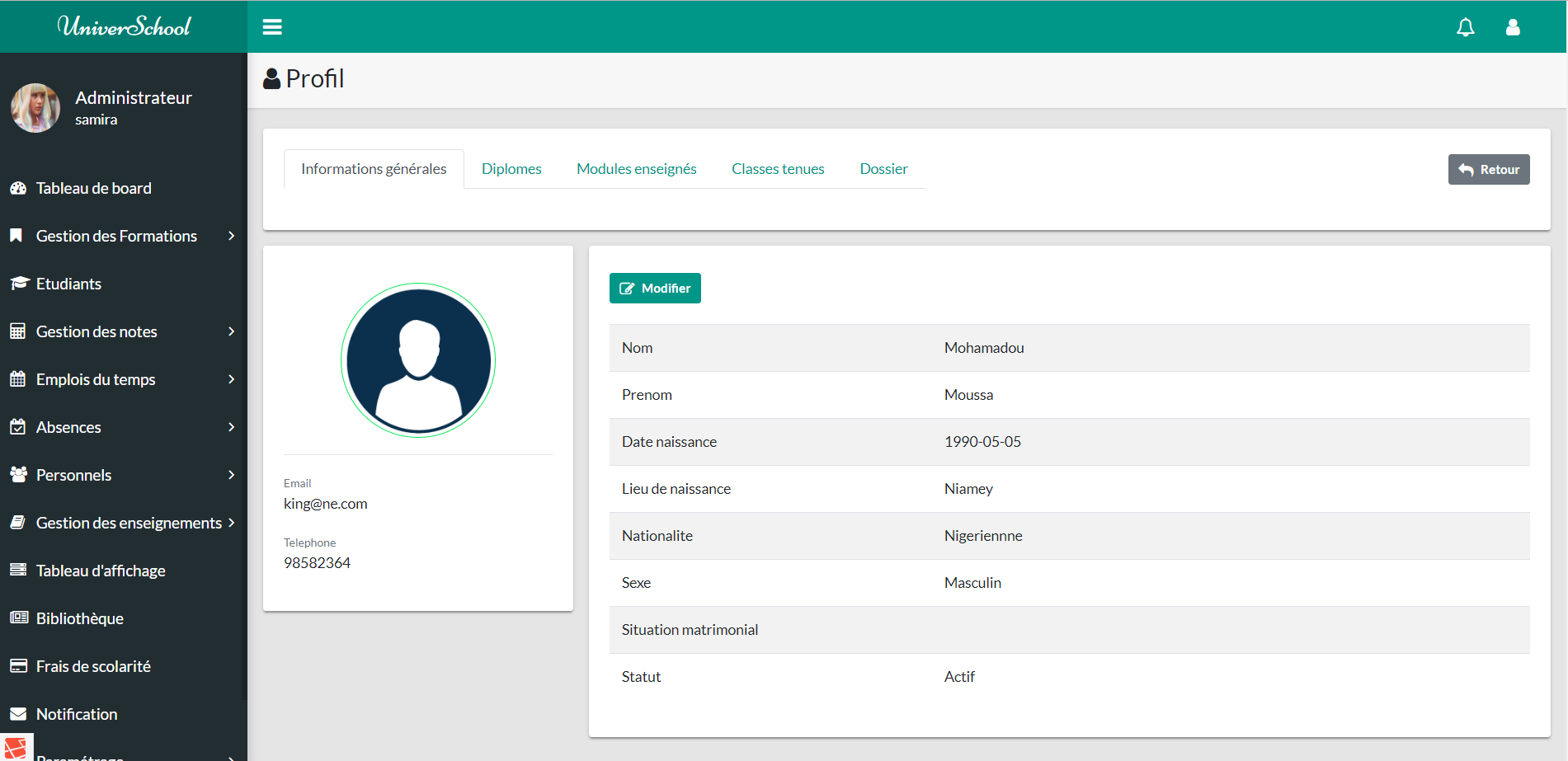


Figure 42 : Profil d’un enseignant

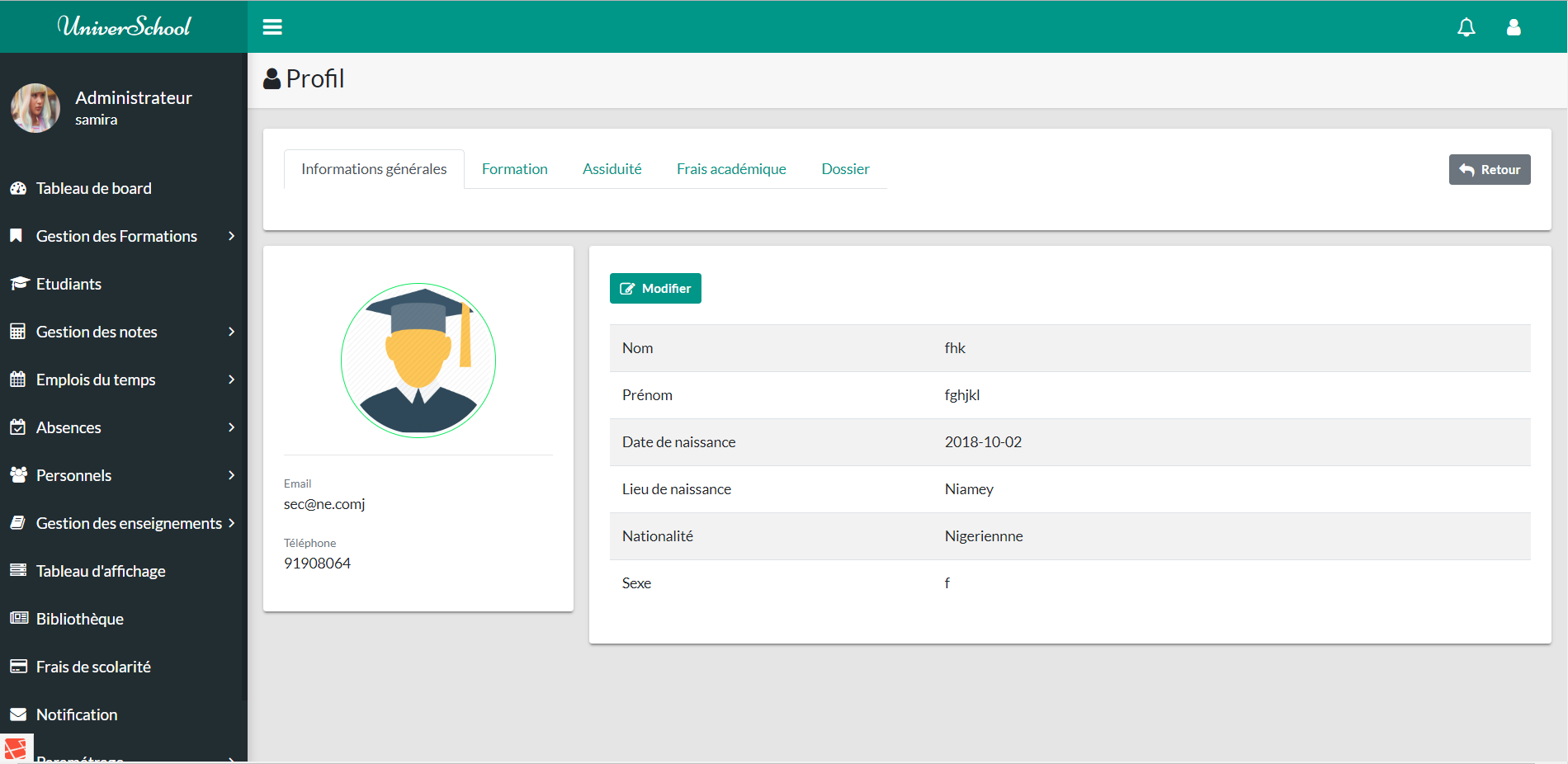


Figure 43 : Profil d’un étudiant

## Affectation des notes

L’affectation des notes d’une évaluation est effectuée par un personnel administratif (ayant le droits et permissions nécessaire pour effectuer l’opération) ou l’enseignant à qui le module (sur lequel porte l’évaluation) est affecté.

Ici il est possible d’affecter une même note à plusieurs étudiants avec l’option « notes groupées ». L’option « importer » permet d’importer les notes à partir d’un fichier Excel.

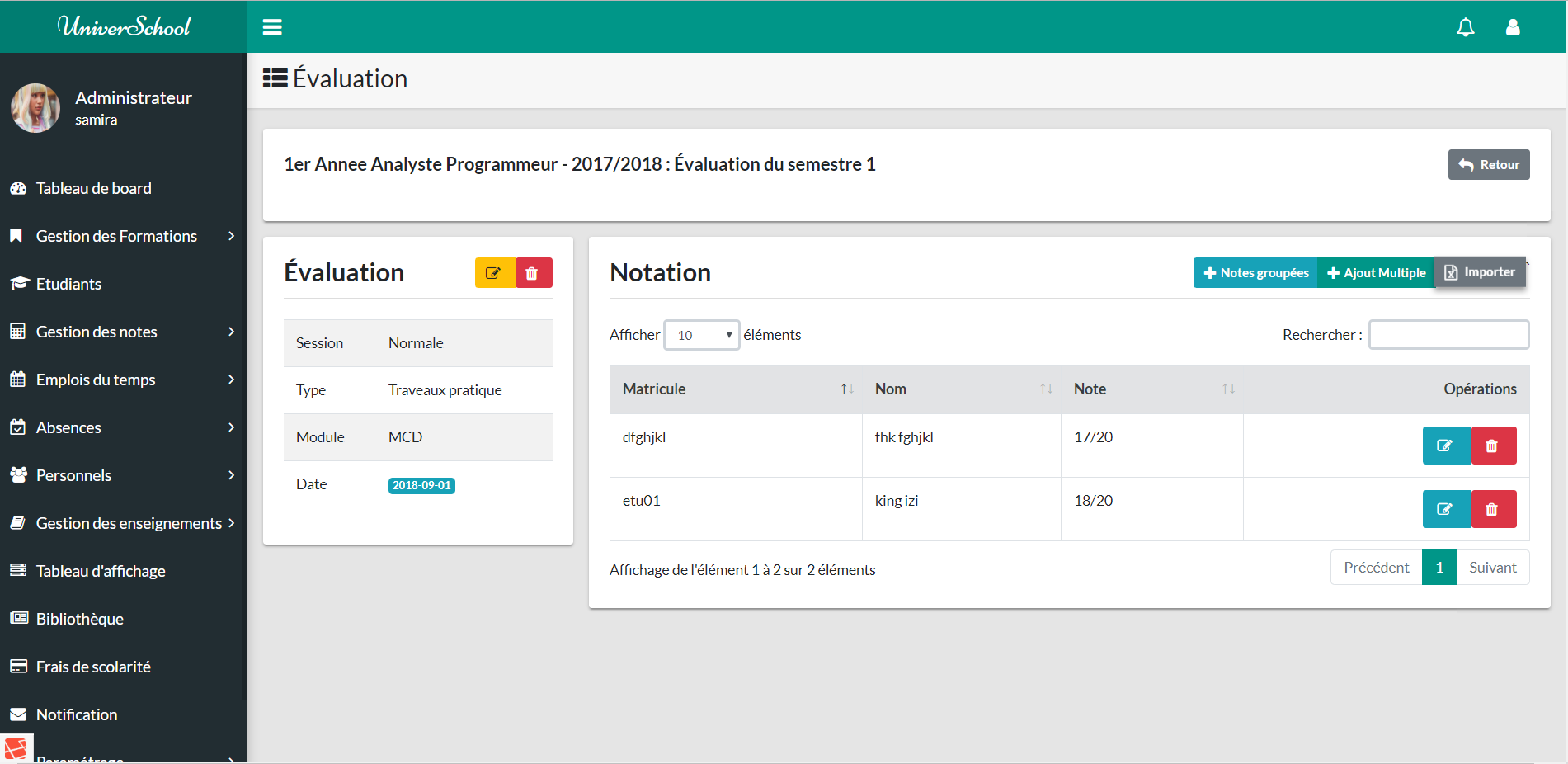


Figure 44:Affectation de note

## Emplois du temps

Cette interface (figure 44) permet à l’utilisateur (ayant tous les droits nécessaires pour effectuer l’opération) de programmer et organiser les séances de cours, TP, TD etc… pour une classe donnée. Et aussi de confirmer la présence d’un professeur à une séance donnée pour qu’il puisse avoir le droit de saisir la liste des absences (voir figure 45).

## C:\Users\hp\Documents\Captureemplois.PNG Absences

Figure 45: Emplois de temps

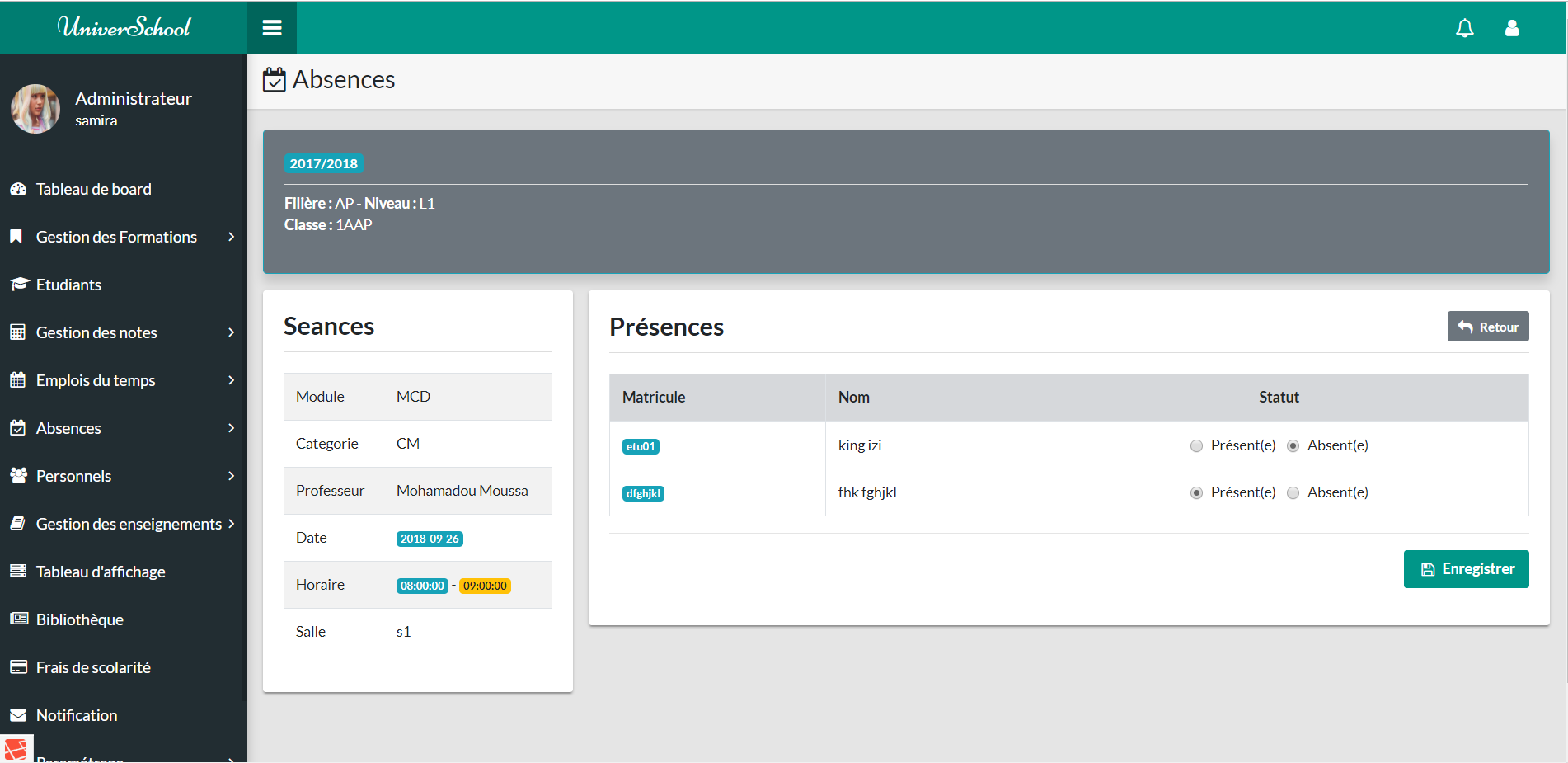
La saisie des absences des étudiants pour une séance donnée est effectuée par l’enseignant ou un personnel administratif ayant les droits nécessaires pour effectuer l’opération.

Figure 46 : Gestion des Absences

# Conclusion générale

L’objectif de notre projet était de concevoir et d’implémenter une application de gestion d’une école supérieure prenant en compte le système LMD.

Dans ce présent document, nous exposons les différentes étapes suivie pour l'élaboration du projet. Nous avons d'abord présenté le contexte général ainsi que la méthodologie adoptée 2TUP.

Une phase d'analyse des besoins nous a permis d'identifier les différents acteurs et spécifier les besoins fonctionnels. Ensuite, nous avons présenté une conception détaillée à travers la représentation des vues statiques et dynamiques du système. Par la suite, les différents choix technologiques et techniques adoptés pour la réalisation de ce projet.

Notre projet est clôturé par une phase de réalisation qui comporte une exposition du travail effectué, illustrée par des captures d'écran.

La réalisation de ce projet nous a permis d’améliorer nos connaissances en création d'application web, et notamment en ce qui concerne le respect strict des standards du Web et l'utilisation poussée de technologies comme l'UML, le SQL, l’Ajax, le JavaScript, et surtout la POO à travers l’utilisation du Framework Laravel. Cette expérience nous a également fait découvrir le monde professionnel et cela nous a été bénéfique autant d'un point de vue professionnel que relationnel en matière de communication et de coopération au sein d'une équipe.

Cependant, des améliorations de ce projet peuvent être envisagées. En effet, l'application peut être étendue en y ajoutant d'autres fonctionnalités comme un système de messagerie instantané, un système d’inscription en ligne etc...

En somme, ce stage nous a permis de mettre en œuvre des compétences scolaires, professionnelles et humaines pour un sujet intéressant. Il nous a aussi permis de suivre une méthodologie de travail bien étudiée, d’approfondir nos connaissances dans le monde du développement des applications et de bien nous exercer sur le Framework Laravel utilisé.

# Bibliographie

* P. Roques, UML 2 par la pratique : Etude de cas et exercices corrigés, EYROLlES, 2011.
* Sandhu k., “Automating class schedule generation in the context of university timetabling information system “, School of Management, Nathan Campus, Griffith University, 21 september 2001.
* Meyer, B. (2000). Conception et Programmation orientées objet. Eyrolles.
* Maurice Chavelli, Découvrez le framework PHP Laravel.
* Guide de l’Etudiant en LMD de l’Institut Supérieur des Technologies de l’Information et de la Communication, Tunisie

# Webographie

|  |  |
| --- | --- |
| Url | Date de consultation |
| <https://www.manager-go.com/gestion-de-projet/dossiers-methodes/elaborer-un-cdc> | 7 juin 2018 à 10h15 |
| https://cahiersdescharges.com/comment-faire-cahier-des-charges/ | 11 juin 2018 à 9h40 |
| https://laravel.com/docs/5.4 | 21 juin 2018 à 13h11 |
| https://laravel.com/docs/5.4/routing | 26 juin 2018 à 16h05 |
| https://laravel.com/docs/5.4/eloquent-relationships | 6 juillet 2018 à 14h18 |
| <https://openclassrooms.com/fr/courses/3613341-decouvrez-le-framework-php-laravel> | 22 juin 2018 au 13 juillet 2018 heures différentes |
| http://www.e-bancel.com/Processus\_2TUP.php | 12 juin au 16 juin 2018 heures différentes |
| https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2035851-uml-c-est-quoi | 10 juillet au 19 juillet 2018 heures différentes |
| <https://fr.wikipedia.org/wiki/R%C3%A9forme_Licence-Master-Doctorat> | 11 juillet 2018 à 10h25 |
| https://www.iae-eiffel.fr/LMD | 26 juillet 2018 à 17h20 |
| https://laravel.com/docs/5.5/authentication | 15 août 2018 à 15h10 |
| https://laravel.com/docs/5.5/queries | 28 août 2018 à 18h35 |
| https://docplayer.fr/99389-Comment-lire-et-comprendre-son-releve-bulletin-de-notes.html | 13 septembre 2018 à 12h13 |