

Dédicaces

Je dédie ce Modeste travail

A ma mère **Coulibaly Dedianwa**.

A mon regretté père **Coulibaly Ténin**.

A mon grand-frère **Coulibaly Doheré** .

A mes chères frères et sœurs.

A tous mes amis du master 2 Génie Informatique.

A tous les gens qui ont cru en moi et qui me donnent envie d'avancer.

Je vous remercie tous pour votre soutien et pour vos encouragements.

Remerciement

Mes remerciements s'adressent au directeur de master et responsable de la filière Génie Informatique de l'UFR SFA, le professeur **AKA Boko**, à mon enseignant et mentor le docteur **ZEZE DJEDJE Sylvain**, responsable de la licence 3 informatiques de l'UFR SFA pour m'avoir confié cette mission et surtout pour sa disponibilité et ses encouragements.

Mes remerciements vont également à l'endroit du professeur **BOA David** Doyen de l'UFR des Sciences Fondamentales et appliquées et à la Secrétaire Principale Madame **KOUASSI Bienvenue** pour sa disponibilité.

Je remercie l'ensemble du corps professoral du master Génie Informatique pour la qualité de la formation reçue et pour les conseils avisés.

Résumé

Ce travail ambitionne de faciliter les tâches administratives de l'UFR des sciences fondamentales et appliquées (SFA). Nous avons tout d'abord identifié les besoins de l'UFR, ensuite nous avons étudié les moyens de mise en place d'une solution et enfin l'implémentation de cette solution a été faite.

Il s'agit de la mise en place d'une application web pour la gestion des effectifs et des notes des étudiants. Ainsi elle permettra La formation des classes basées sur les unités d'enseignements et les parcours, la saisie des notes après chaque examen et une production des différents états (relevés de notes, procès-verbaux, listes des classes...).

Pour la réalisation de ce travail le langage UML (langage de modélisation unifié) a été utilisé pour la modélisation. Le langage de programmation choisie est le langage PHP (Hypertext Preprocessor) et Le système de gestion de base de données (SGBD) est MySQL

Abstract

This work aim to make easy the administrative tasks of UFR's (university department) fundamental and applied sciences (SFA). In first the needs of UFR has been identified, then we studied the means of setting up a solution and finally the implementation of this solution has been made.

It is about the setting up of a web application for the management of the effectives and the notes of the students. So it will permit the formation of the classes based on the units of teachings and the courses, the seizure of the notes after every exam and a production of the different states (raised of notes, minutes, lists of the classes...).

For the realization of this work the UML language (language of modelling unified) has been used for the modelling. The chosen programming language is the PHP language (Hypertext Preprocessor) and the system of database managements (SGBD) is MySQL

Table des matières

Résumé	2
Abstract	3
Liste des tableaux	ii
Liste des figures	iv
Liste des symboles	iv
Introduction	1
1 CONTEXTE GENERAL DU PROJET	3
1.1 L'UFR SFA	4
1.1.1 Présentation	4
1.1.2 Organisation et Fonctionnement	4
1.2 Cadre générale du projet	6
1.2.1 Etude de l'existant	6
1.2.2 Présentation générale du projet	7
1.2.3 Problématique	7
1.2.4 Objectif du projet	8
1.2.5 Definition du cahier des charges	9
1.2.6 Planification du projet	9
1.3 Conclusion partielle	11
2 ETUDE FONCTIONNELLE ET TECHNIQUE	12
2.1 Analyse et méthodologie de travail	13
2.1.1 Analyse du projet	13
2.1.2 Choix methodologique	13
2.1.3 Technologie Orienté Objet	15
2.1.4 Architecture MVC (Model-View-Controller)	16
2.2 Outils de mise œuvre	17

TABLE DES MATIÈRES

2.2.1	Le langage PHP	17
2.2.2	Le Framework Laravel	18
2.3	Modélisation	22
2.3.1	UML	22
2.3.2	Mise en place de la base de donnée	35
2.4	Conclusion partielle	37
3	REALISATION DE L'APPLICATION	38
3.1	Environnement logiciels	39
3.2	Architecture et fonctionnement de l'application	39
3.2.1	Le modèle MVC de Laravel	39
3.2.2	Structure	40
3.2.3	Organisation	42
3.3	Les interfaces utilisateurs	45
3.3.1	Authentification	45
3.3.2	Gestion des étudiants	47
3.3.3	Gestion des notes	49
3.3.4	Gestion des parcours et production des états	54
3.4	Conclusion partielle	57
	Conclusion générale	57

Liste des tableaux

1.1	Structure des etudes	5
1.2	Tableau récapitulatif des fonctionnalités.	10
1.3	Planning prévisionnel de réalisation	11
2.1	Tableau descriptif des classes métiers.	34
3.1	Présentation des dossiers de Laravel	42

Table des figures

1.1	organigramme	6
2.1	Cas d'utilisation Création d'une année académique.	23
2.2	Cas d'utilisation de Gestion des parcours.	24
2.3	Cas d'utilisation général de l'étudiant.	25
2.4	Cas d'utilisation de Gestion des unité d'enseignement (UE).	26
2.5	Cas d'utilisation de Gestion des classes.	26
2.6	Cas d'utilisation de Gestion des notes.	27
2.7	Cas d'utilisation de Gestion des étudiants.	28
2.8	Cas d'utilisation Etablir les procès-verbaux.	29
2.9	Cas d'utilisation de Gestion des examens.	30
2.10	Cas d'utilisation Paramétrage	30
2.11	Diagramme des classes.	32
2.12	Modèle physique des données.	36
3.1	Structure MVC de Laravel	40
3.2	Structure des fichiers sous Laravel	41
3.3	Organisation général du framework Laravel.	43
3.4	Conténu du dossier app.	43
3.5	Conténu du dossier public.	44
3.6	Conténu du dossier ressources.	45
3.7	Interface d'authentification.	45
3.8	Page d'accueil.	46
3.9	Menu général de l'application.	46
3.10	Liens de navigation rapides.	46
3.11	Formulaire d'inscription des étudiants.	47
3.12	Formulaire de connexion des étudiants.	48
3.13	Liste des notes d'un étudiant.	48
3.14	Liste des classes.	49
3.15	Liste des unités d'enseignement(UE).	49
3.16	Interface d'insertion des notes.	50

3.17	Liste des classes.	50
3.18	Interface de consultation des notes par étudiant.	51
3.19	Interface de présentation des notes par unité d'enseignement	52
3.20	Tableau de modification des notes par classe	52
3.21	Liste des étudiant de la licence 3 chimie	53
3.22	Interface de modification des notes	53
3.23	Liste des étudiants de la licence 2 informatique	54
3.24	Liste des admis	55
3.25	Liste des récalés	55
3.26	Procès-verbal provisoire	56
3.27	Procès-verbal provisoire exporté en excel	56
3.28	Tableau des notes par semestre	58
3.29	Un relevé de note en PDF	59

Liste des symboles

CSS : feuilles de style en cascade, de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.

ECUE : Element Constitutif d'Unite d'Enseignement

HTML : 'HyperText Markup Language, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte

HTTP : Hypertext Transfer Protocol en français protocole de transfert hypertexte

IDE : Environnement de Développement Intégré

IIS : Internet Information Services communément appelé IIS (prononcé 2 i s), est un serveur Web (FTP, SMTP, HTTP etc.) des différents systèmes d'exploitation Windows

LMD : Licence Master Doctorat

MPD : Modèle Physique des Données

OMG : L'Object Management Group est une association américaine à but non lucratif créée en 1989 dont l'objectif est de standardiser et promouvoir le modèle objet sous toutes ses formes.

ORM : Object Relation Mapping est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle

PHP : Hypertext Preprocessor, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP

POO : Programmation Orientée Objet

PV : Procès-verbal

SFA : Sciences Fondamentales et Appliquées

SGBD : Système de Gestion de Base de Données

SQL : Structured Query Language, en français langage de requête structurée est un langage informatique normalisé servant à exploiter des bases de données relationnelles.

UFR : Unité de Formation et de Recherche

UML : Unified Modeling Language en français Langage de Modelisation Unifié

URI : Uniform Resource Identifier, soit littéralement identifiant uniforme de ressource, est une courte chaîne de caractères identifiant une ressource sur un réseau (par exemple une ressource Web) physique ou abstraite, et dont la syntaxe respecte une norme d'Internet mise en place pour le World Wide Web

XML : L'Extensible Markup Language (langage de balisage extensible en français)

Introduction

L'évolution des Nouvelles Technologies de l'information et de la communication a favorisé l'utilisation de l'outil informatique dans un nombre croissant de domaines. L'informatique étant la science du traitement de l'information, elle facilite aujourd'hui le travail dans plusieurs domaines d'activités.

C'est dans cette vision que l'unité de formation et de recherche des Sciences fondamentales et Appliquées (SFA) de l'université Nangui Abrogoua, veut s'approprier une application informatique. Cette application va permettre :

- De gérer l'effectif de ses étudiants ;
- De faire le calcul des notes d'examen ;
- De fournir les différents états.

La mise en place de cette application fait l'objet de notre mémoire de fin d'étude. Notre tâche sera la conception et l'élaboration de cet outil informatique qui devra être accessible via internet et devra permettre une gestion souple et rigoureuse des étudiants de l'UFR SFA.

Ce rapport présente, en trois parties distinctes, l'ensemble du projet. Il rassemble les réflexions nées durant les phases d'analyse et les défis techniques relevés durant l'implémentation. Il nous permet également d'exprimer une opinion motivée sur certaines technologies et d'exprimer nos idées sur certaines décisions d'ordre stratégique. Cet aspect est fondamental dans la mesure où la définition précise du cahier des charges a été un élément clé sur lequel le plan d'implémentation a pu être conçu.

Dans la première partie, il s'agira de rappeler le contexte. Nous aborderons ensuite l'étude fonctionnelle et technique du projet et dans la dernière partie la réalisation de l'application.

Chapitre 1

CONTEXTE GENERAL DU PROJET

1.1 L’UFR SFA

1.1.1 Présentation

L’UFR des Sciences Fondamentales et Appliquées(SFA) a été créé en 1996 au sein de l’université Nangui Abrogoua (ex Abobo-Adjamé).Les enseignements y sont dispensés dans les quatre grands domaines de la science. A savoir la Chimie, la Physique, les Mathématiques et l’Informatique.

Dans le cadre de l’harmonisation des cursus d’enseignement supérieurs africains, le cursus universitaire de l’UFR-SFA s’organise autour de trois diplômes nationaux : la Licence, le Master et le Doctorat. Cette organisation dite LMD permet d’accroître la mobilité des étudiants, la mobilité entre les disciplines et les chances de trouver un emploi pour les étudiants diplômés de l’UFR-SFA. La structure des études est présentée dans le tableau 1.1.

1.1.2 Organisation et Fonctionnement

L’UFR des Sciences Fondamentales et Appliquées est organisée en une direction, un personnel, un Conseil d’UFR, un Conseil Scientifique et un Conseil Pédagogique.

La direction est dirigée par un doyen. Le doyen de l’UFR est nommé par le Ministère de l’Enseignement Supérieur et de la Recherche Scientifique pour une durée de 3 ans renouvelable. Il est assisté d’un vice doyen, d’une secrétaire principale, de 14 responsables de parcours et de 4 directeurs de laboratoire.

Le personnel est constitué de 49 enseignants permanents et de 48 enseignants vacataires.

La figure 1.1 présente l’organigramme de l’UFR SFA.

Le Conseil de l’UFR est une instance qui fixe les grandes orientations de l’UFR. C’est l’instance qui est au-dessus du doyen dans la hiérarchie. Il est composé de 26 représentants élus dans différents collèges :

- 18 enseignants permanents,
- 2 administrateurs et techniciens,
- 6 étudiants.

Le Conseil Scientifique (CS) est l’instance qui est consulté sur les orientations des politiques de recherche, de documentation scientifique et technique au sein de l’UFR. Le CS de l’UFR-SFA comprend 6 membres.

Le Conseil Pédagogique (CP) est consulté sur les orientations pédagogiques au sein de l’UFR. Il donne son avis sur les parcours de formation de l’UFR SFA. Il donne son avis sur les demandes d’équivalence pour des étudiants extérieurs désirant poursuivre leurs études à l’UFR-SFA. Le CP de l’UFR-SFA comprend 8 membres.

GRADE	MENTION	PARCOURS - SPECIALITE
LICENCE	Mathematique-Informatique	<ul style="list-style-type: none">• Parcours Mathématiques.• Parcours Informatique.
	Physique-Chimie	<ul style="list-style-type: none">• Parcours Physique ;• Parcours Chimie ;• Parcours Physique-Chimie.
MASTER	Mathematique	<ul style="list-style-type: none">• Spécialité Algèbre Commutative et Cryptographie (ACC)• Spécialité Probabilités et Statistiques (PS)• Spécialité Optimisation, Equations Différentielles (ED) et Analyse Numérique (AN)
	Informatique	<ul style="list-style-type: none">• Spécialité Méthodes Informatiques Appliquées à la Gestion et à l'Economie (MIAGE)• Spécialité Génie Informatique (GI)
	Physique	<ul style="list-style-type: none">• Spécialité Physique du Vivant (PV)• Spécialité Photonique
	Chimie	<ul style="list-style-type: none">• Spécialité : Physico-chimie des Matériaux et Procédés (PCMP)• Spécialité : Chimie et Physico-chimie des Substances Naturelles (CPCSN)
DOCTORAT		

TABLE 1.1 – Structure des etudes

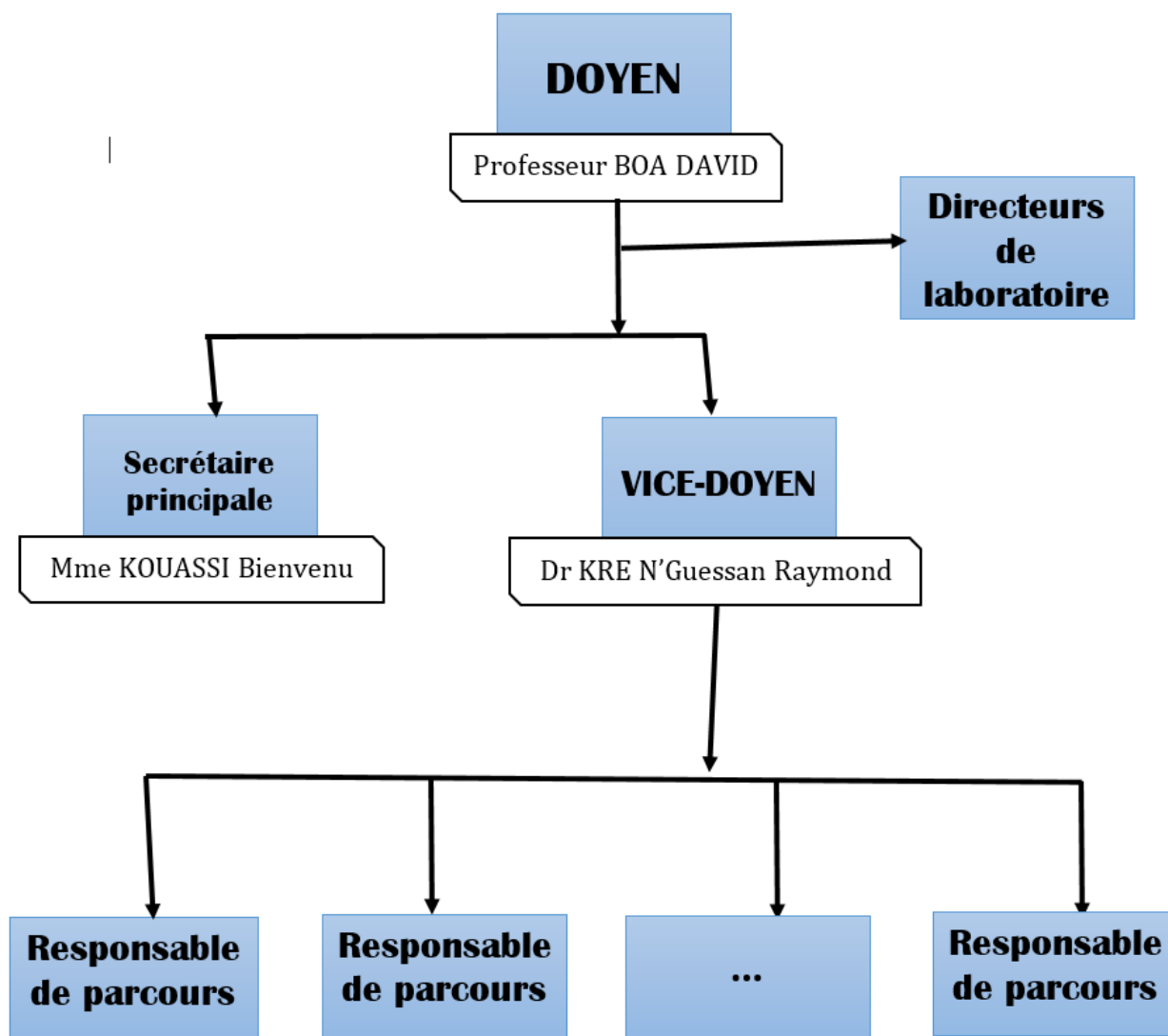


FIGURE 1.1 – organigramme

1.2 Cadre générale du projet

1.2.1 Etude de l'existant

L'étude de l'existant est une phase importante pour comprendre le système actuel et définir ses objectifs.

Le constat est que l'UFR SFA enregistre ses données dans des fichiers Excel. Bien qu'Excel offre beaucoup d'avantages, il n'offre pas une très bonne organisation des données (en fonction du temps, du type ...). A chaque nouveau besoin il est nécessaire de créer un nouveau fichier et la lecture des informations ne peut se faire que sous forme de tableur.

Les données sont stockées en local sur les ordinateurs de l'UFR. Une panne matériel peut occasionner des pertes de données importantes.

On observe également que l'ensemble des activités de l'UFR est géré par traitement manuel tout en utilisant des supports papiers. La gestion du papier ne rassure pas car il est vulnérable (perte de document, dégradation du papier, difficulté de classement, recherche difficile ...).

L'étude de l'existant nous permet d'affirmer que l'UFR SFA ne possède pas un système optimisé pour organiser et gérer ses données.

1.2.2 Présentation générale du projet

Notre projet de fin d'études s'inscrit dans le cadre d'une solution optimisée pour la gestion des effectifs et des notes. Cette solution doit être adaptée au défi du système LMD nouvellement instauré dans le système universitaire ivoirien.

Ainsi ce projet consiste à mettre en place une application web pour la gestion des effectifs et des notes de l'UFR SFA.

Une application est, dans le domaine informatique, un programme (ou un ensemble logiciel) directement utilisé par l'utilisateur pour réaliser une tâche, ou un ensemble de tâches élémentaires d'un même domaine ou formant un tout.

Cette application devra permettre :

- l'établissement efficace **des procès-verbaux (PV)** ;
- l'élaboration rapide **des relevés de notes** ;
- l'acquisition **d'une banque de données des étudiants sur plusieurs générations** ;
- de produire **une situation** de chaque étudiant, chaque classe, chaque filière pour une année donnée ;
- de gérer **les inscriptions et les réinscriptions pédagogiques** pour chaque année universitaire.

La finalité de notre projet est d'organiser et centraliser les données. Avoir un accès facile aux données en consultation et produire les relevés de notes, les procès-verbaux et les listes des étudiants.

1.2.3 Problématique

L'UFR SFA accueille chaque année environ 1000 étudiants pour un personnel administratif restreint. Le constat est que ce personnel éprouve des difficultés face à la gestion d'un aussi grand nombre d'étudiants.

La préparation des listes étudiants dans l'objectif de former les groupes de cours magistraux (CM) et de travaux dirigés (TD), de faire les examens est difficile. Plusieurs raisons expliquent cela :

- L'administration de l'UFR est dans l'attente des listes des nouveaux étudiants venant du ministère à cause des inscriptions à la scolarité qui ne débutent pas vite et prennent beaucoup de temps pour finir. Ceci a un impact négatif sur le calendrier universitaire (calendrier des cours et examens).
- Le faible nombre de personne (la secrétaire principale) habilitée à ces tâches se retrouve très vite surmené par la charge de travail.

Le traitement des notes après les examens est à la charge de la secrétaire principale. Seul face à cette tâche fastidieuse, un retard est accusé dans la proclamation des résultats et la disponibilité des procès-verbaux.

Une difficulté dans le traitement des relevés de notes car la recherche des données d'un étudiant sur plusieurs années est fastidieuse. Celle-ci est dû au fait que les données d'un étudiant ne sont pas centralisées (le stockage des données est fait de façon indépendante d'une année à une autre).

La perte de données liées à des problèmes matériels ou à une mauvaise manipulation. La récupération est très souvent impossible car il n'existe pas une sauvegarde des données fiable et sécurisée.

La problématique soulevée ici est donc de trouver comment améliorer de façon optimale le travail administratif de l'UFR SFA, de la formation des groupes de CM (cours magistral) et TD (travaux dirigés) ou des classes à la génération des différents états (Procès-verbaux et relevés de note).

La solution choisie est la mise en place d'un système informatique pour la gestion des effectifs et des notes. La mise en place d'une application adaptée peut réduire la charge de travail de l'administration de l'UFR SFA.

1.2.4 Objectif du projet

Cette application web va à terme permettre d'atteindre les objectifs suivants :

- Sécuriser les données des étudiants durant leur cursus dans une base de données ;
- Offrir un outil de saisie adapté à la saisie rapide et sécurisé des notes d'examen,
- faire le stockage et le calcul automatique des notes d'examens dans une base de données ;
- Exporter facilement les données de la base pour établir les états (listes des classes, procès-verbaux, relevés de note) ;
- Offrir un formulaire pour réaliser les inscriptions pédagogiques des étudiants chaque année universitaire ;

1.2.5 Definition du cahier des charges

1.2.5.1 Description du portail

Afin de mener à bien notre projet il est nécessaire d'organiser les besoins exprimé et d'établir une liste des différentes fonctionnalités à mettre en œuvre. Les fonctionnalités requises sont présentées dans le tableau 1.2. Nous avons organisé notre application en plusieurs modules qui sont représentés par les rubriques.

1.2.6 Planification du projet

La planification consiste à prévoir le déroulement du projet tout au long des phases constituant le cycle de vie prévu. En raison de la spécificité de notre étude notre projet de mise en place d'une application web se divise en trois grandes parties telles que la description et l'analyse des besoins spécifiques, la mise en production et la bascule.

- **Description et analyse des besoins spécifiques** : Les objectifs de cette phase sont de bien cerner le sujet d'une part, et de délimiter le périmètre du projet d'autre part au niveau fonctionnel. Ici il s'agit de la définition de notre cahier des charges fonctionnelles réalisées en fonction des besoins exprimé.
- **La mise en production** : L'objectif de cette phase est de construire l'application web, puis d'en vérifier la fiabilité des différents traitements. Les tâches typiques associées à cette étape sont :
 - La personnalisation et le paramétrage.
 - La migration des données.
 - Le développement spécifique des fonctionnalités non traitées par l'application web.
 - La mise à jour finale de la documentation du système final.
- **La bascule** : L'objectif de cette phase est le démarrage opérationnel de l'application web. Les tâches typiques associées à cette étape sont :
 - La formation des utilisateurs finaux et la documentation des utilisateurs.
 - Le pilotage : Test d'intégration sur la base des processus et validation des modes opératoires par les utilisateurs.

Le tableau 1.3 nous donne le planning prévisionnel pour la réalisation de notre projet.

Rubriques	Fonctionnalité
Gestion des étudiants	<ol style="list-style-type: none"> 1. Faire l'inscription pédagogique des étudiants 2. Consulter la liste des étudiants par classe 3. Consulter la liste des étudiants par UE 4. Faire les passages en année supérieur 5. Faire les compensations 6. Consulter les résultats des étudiants 7. Consulter les procès-verbaux de chaque classe 8. Générer les relevés de note en format PDF 9. Générer les procès-verbaux en fichier Excel
Gestion des parcours	<ol style="list-style-type: none"> 1. Ajouter des parcours 2. Définir les parcours autorisés pour une années universitaire 3. Ajouter des classes en fonction d'un parcours 4. Définir les classes autorisées pour une années universitaire 5. Ajouter des unités d'enseignement (UE) en fonction d'un semestre 6. Ajouter des ECUE et les lier à leur UE respective
Gestion des notes	<ol style="list-style-type: none"> 1. Insérer les notes des étudiants après chaque examen 2. Modifier les notes des étudiants sur réclamation 3. Consulter les notes des étudiants par unités d'enseignement 4. Consulter les notes des étudiants par classe 5. Générer la liste des admis et des récalés dans une UE en format PDF et Excel 6. Générer la liste des admis et des récalé par classe en format PDF et Excel
Gestion des examens	<ol style="list-style-type: none"> 1. Faire le programme des examens 2. Générer le programme des examens en format PDF et Excel 3. Etablir le calendrier des examens ;

TABLE 1.2 – Tableau récapitulatif des fonctionnalités.

ETAPES	PERIODES	DUREE
Etude fonctionnelle et technique	du 03 février 2016 au 01 mars 2016	1 mois
Réalisation	du 05 mars au 01 juin 2016	3 mois
Formation et assistance des utilisateurs		2 mois

TABLE 1.3 – Planning prévisionnel de réalisation .

1.3 Conclusion partielle

Dans ce chapitre nous avons décrit le contexte général dans lequel s'inscrit notre projet de fin d'études. Nous avons présenté la structure d'accueil qui est l'UFR SFA. La problématique nous a permis de définir les objectifs du projet qui se résument à la mise en place d'une application web.

Nous pouvons dès lors passer à la deuxième partie qui est l'étude fonctionnelle et technique.

Chapitre 2

ETUDE FONCTIONNELLE ET TECHNIQUE

Dans ce chapitre, nous allons décrire la méthodologie de conception et les besoins de l'UFR SFA, par la suite nous allons les analyser en utilisant le formalisme UML.

2.1 Analyse et méthodologie de travail

2.1.1 Analyse du projet

Une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.

Les premières méthodes d'analyse sont apparues dans les années 70, elles sont basées sur une découpe cartésienne (fonctionnelle et hiérarchique) d'un système.

Les années 80 ont vu l'avènement de l'approche systémique qui se résume : Modélisation des données + modélisation des traitements (Merise, Axial, IE...).

L'émergence des méthodes objet dans les années 1990-1995 vient entamer une nouvelle ère dans le monde du développement logiciel. Une question s'est posée avec ces méthodes : Comment structurer un système sans centrer l'analyse uniquement sur les données ou uniquement sur les traitements (mais sur les deux) ? Plus de 50 méthodes objet sont apparues durant cette période (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE...) !

Afin de comprendre le fonctionnement du système, les besoins identifiés nous ont permis de guider notre analyse par **la méthode de modélisation UML**. UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG). Les groupes de travail UML ont donc travaillé à unifier non pas les processus, mais plus exactement les meilleures pratiques de développement objet. Ces processus se distingueront par le générique UNIFIED PROCESS.

Notre choix de la méthodologie de travail découle de la prise en compte des objectifs de l'UFR SFA. Ces objectifs dépendent des besoins des utilisateurs.

2.1.2 Choix methodologique

Pour mener à bien notre projet, nos choix se sont portés sur les méthodologies et technologies suivantes :

- La technologie Orienté Objet
- Le langage UML
- Le modèle MVC

2.1.2.1 Le langage UML

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes (représentation graphique

schématique) conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. UML permet de comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue[3].

UML est un langage formel et normalisé. Il encourage l'utilisation d'outil, favorise un gain en précision et est un gage de stabilité.

UML est également un support de communication performant car il cadre l'analyse, facilite la compréhension de représentations abstraites complexes et son caractère polyvalent et sa souplesse en font un langage universel.

UML est stable, une modification mineure dans les besoins ne nécessite pas forcément des changements importants du système en construction. Il est adaptable, n'impose pas de méthode de développement et en plus il est possible d'intégrer son utilisation à un processus de développement existant.

UML s'articule autour de **treize types de diagrammes**, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ses diagrammes sont répartis en deux grands groupes :

Six diagrammes structurels : Ils ont vocation à représenter l'aspect statique d'un système (classes, objets, composants...). Il s'agit **du diagramme de classes, du diagramme d'objets, du diagramme de composants, du diagramme de déploiement, du diagramme de packages et du diagramme de structure composite** ;

- Diagramme de classes (class diagram) : représentation des classes intervenant dans le système.
- Diagramme d'objets (object diagram) : représentation des instances de classes (objets) utilisées dans le système.
- Diagramme de composants (component diagram) : représentation des composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)
- Diagramme de déploiement (deployment diagram) : représentation des éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
- Diagramme des paquets (package diagram) : représentation des dépendances entre les paquets (un paquet étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML), c'est-à-dire entre les ensembles de définitions.

- Diagramme de structure composite (composite structure diagram) : représentation sous forme de boîte blanche les relations entre composants d'une classe (depuis UML 2.x).

Sept diagrammes comportementaux : Ils représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Ce sont le **diagramme de cas d'utilisation**, **diagramme d'état-transition**, **diagramme d'activité**, **diagramme de séquence**, **diagramme de communication**, **diagramme global des interactions**, **diagramme de temps**.

- Diagramme des cas d'utilisation (use-case diagram) : représentation des possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire de toutes les fonctionnalités que doit fournir le système.
- Diagramme états-transitions (state machine diagram) : représentation sous forme de machine à états finis le comportement du système ou de ses composants.
- Diagramme d'activité (activity diagram) : représentation sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.
- Diagramme de séquence (sequence diagram) : représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
- Diagramme de communication (communication diagram) : représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets (depuis UML 2.x).
- Diagramme global d'interaction (interaction overview diagram) : représentation des enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité) (depuis UML 2.x).
- Diagramme de temps (timing diagram) : représentation des variations d'une donnée au cours du temps (depuis UML 2.3).

Dans le cadre de notre étude, nous verrons seulement les diagrammes des cas d'utilisation et les diagrammes de classe.

2.1.3 Technologie Orienté Objet

a) Présentation

Face à la crise logicielle due essentiellement à l'évolutivité des besoins, à la complexité des logiciels et aux échecs dans la gestion des risques, les développeurs se tournent de plus en plus vers la technologie orientée objet.

Le principe de base de la technologie objet est la construction de logiciel à partir de composants, si possible standardisés et réutilisables. Ils encapsulent les informations et les opérations qui définissent un comportement. Ils communiquent entre eux par l'intermédiaire de messages. Les objets informatiques correspondent souvent à des objets du monde réel. Des systèmes d'information complexes peuvent être construits par composition, classification ou associations d'objets, comme dans la nature, où des systèmes vivants complexes sont construits à partir de cellules.

La technologie objet est la voie la plus prometteuse pour déployer des systèmes basés sur des composants qui peuvent être adaptés et changés sans avoir à examiner la totalité du code.

b) Avantages

La technologie orientée objet a plusieurs avantages. Elle présente entre autre une meilleure modélisation du monde réel en décrivant plus précisément les données et les processus. Par ailleurs, elle permet un développement plus rapide, une plus grande modularité et flexibilité dans la construction des systèmes. Il en découle une réduction des coûts de maintenance, parce que des objets peuvent être modifiés sans conséquence pour les autres objets. Cette technologie possède également une grande stabilité car une modification mineure dans les besoins ne nécessite pas forcément des changements importants du système en construction.

Tous ces atouts nous ont amené à adopter l'approche orientée objet pour modéliser notre système. Notre modèle se base sur les diagrammes UML.

2.1.4 Architecture MVC (Model-View-Controller)

a) Définition et fonctionnement

Le Modèle-Vue-Contrôleur est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle. C'est une façon d'organiser son code en appliquant le principe de séparation des responsabilités, en l'occurrence celles du traitement de l'information et de sa mise en forme. Il divise l'IHM en un modèle, une vue et un contrôleur, chacun ayant un rôle précis dans l'interface.

Son principe de base est la séparation des composants suivants :

- **Le modèle** qui représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données, garantit leur intégrité et offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation ;
- **La vue** qui correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir

toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur ;

- **Le contrôleur** prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle, ce dernier avertit la vue que les données ont changé pour qu'elle se mette à jour. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

En résumé, lorsqu'un client envoie une requête à l'application, la requête envoyée depuis la vue est analysée par le contrôleur (par exemple un clic de souris pour lancer un traitement de données). Le contrôleur demande ensuite au modèle approprié d'effectuer les traitements et notifie à la vue que la requête est traitée. La vue notifiée fait une requête au modèle pour se mettre à jour (par exemple affiche le résultat du traitement via le modèle).

b) Avantages

Le modèle MVC présente de nombreux avantages en présentant une clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur sur le projet. De plus, les trois (3) parties de l'application (logique de présentation, logique métier et logique applicative) sont parfaitement indépendantes. On peut dès lors confier la programmation de chaque partie à différentes équipes, la modification des traitements n'affecte pas les vues. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne sont pas affectées. La maintenance de l'application est plus souple. Ce découpage permet de modifier la présentation sans toucher à la structure de l'application ni à la logique métier.

2.2 Outils de mise œuvre

2.2.1 Le langage PHP

2.2.1.1 Définition et avantages

Plusieurs centaines de langages ont ainsi vu le jour depuis la naissance de l'informatique dans les années 1950. Parmi les plus connus, peuvent être cités les langages C, C++, C#, Java, Perl, Python, Basic, ActionScript, etc.

PHP (Hypertext Preprocessor) est un langage de programmation. Il permet d'écrire des programmes, tout comme les mathématiques permettent de résoudre des problèmes. Très en vogue actuellement, il est cependant loin d'être le seul dans sa catégorie.

Le PHP est un langage interprété qui a été conçu dès son origine pour le Web. Il est aujourd'hui devenu le leader incontesté dans ce domaine. Plus de 9 millions de sites l'ont aujourd'hui choisi comme plateforme de développement web[1].

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP (protocole de transfert hypertexte) tel qu'IIS (Internet Information Services). Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct.

L'utilisation de PHP commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est facilité par l'installation des modules correspondants effectuée sur le serveur. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Il est multiplateforme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (prendre en compte les règles d'arborescences de répertoires qui peuvent changer). Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite, comme tout langage de programmation, une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage.

2.2.2 Le Framework Laravel

2.2.2.1 Définition

Un Framework informatique est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel. Autrement dit c'est une base cohérente avec des briques toutes prêtes à disposition. Il existe des Framework pour tous les langages de programmation et en particulier pour PHP [2].

L'utilité d'un Framework est d'éviter de passer du temps à développer ce qui a déjà été fait par d'autres souvent plus compétents et qui a en plus été utilisé et validé par de nombreux utilisateurs. On peut imaginer un Framework comme un ensemble d'outils à disposition.

2.2.2.2 Constitution de Laravel

Laravel a été créé par **Taylor Otwell** en juin 2011.

Le référentiel Laravel/laravel présent sur le site GitHub contient le code source des premières versions de Laravel. À partir de la cinquième version, le Framework est développé au sein du référentiel Laravel/framework.

En peu de temps, une communauté d'utilisateurs du Framework s'est constituée [2].

Laravel, initie une nouvelle façon de concevoir un Framework en utilisant ce qui existe de mieux pour chaque fonctionnalité. Par exemple toute application web a besoin d'un système qui gère les requêtes HTTP. Plutôt que de réinventer quelque chose, le concepteur de Laravel a tout simplement utilisé celui de **Symfony** en l'étendant pour créer un système de routage efficace. En quelque sorte Otwell a fait son marché parmi toutes les bibliothèques disponibles.

Laravel ce n'est pas seulement le regroupement de bibliothèques existantes, c'est aussi de nombreux composants originaux et surtout une orchestration de tout ça[2].

Laravel est constitué de :

- un système de routage perfectionné (RESTful et ressources),
- un créateur de requêtes SQL (langage de requête structurée) et un ORM (object-relational mapping) performants,
- un moteur de template efficace,
- un système d'authentification pour les connexions,
- un système de validation,
- un système de pagination,
- un système de migration pour les bases de données,
- un système d'envoi d'emails,
- un système de cache,
- un système d'événements,
- un système d'autorisations,
- une gestion des sessions...

2.2.2.3 Le meilleur de PHP

La version de Laravel utilisée pour notre projet est la 5.2, elle nécessite au minimum la version 5.5.9 de PHP. Pour aborder de façon efficace ce framework, il faut être familiarisé avec certaines notions :

les espaces de noms : c'est une façon de bien ranger le code pour éviter des conflits de nommage. Laravel utilise cette possibilité de façon intensive. Tous les composants sont rangés dans des espaces de noms distincts, de même que l'application créée.

les fonctions anonymes : Ce sont des fonctions sans nom (souvent appelées closures) qui permettent d'améliorer le code. Les utilisateurs de Javascript y sont habitués. Les utilisateurs de PHP un peu moins parce qu'elles y sont plus récentes. Laravel les utilise aussi de façon systématique.

les méthodes magiques : Ce sont des méthodes qui n'ont pas été explicitement décrites dans une classe mais qui peuvent être appelées et résolues.

les interfaces : Une interface est un contrat de constitution des classes. En programmation objet c'est le sommet de la hiérarchie. Une interface de programmation est une façade clairement délimitée par laquelle un logiciel offre des services à d'autres logiciels. L'objectif est de fournir une porte d'accès à une fonctionnalité en cachant les détails de la mise en œuvre. Une interface de programmation peut comporter des classes, des méthodes ou des fonctions, des types de données et des constantes. Tous les composants de Laravel sont fondés sur des interfaces. La version 5 a même vu apparaître un lot de contrats pour étendre de façon sereine le Framework.

les traits : C'est une façon d'ajouter des propriétés et méthodes à une classe sans passer par l'héritage (principe basé sur des classes dont les « filles » héritent des caractéristiques de leur(s) « mère(s) »), ce qui permet de passer outre certaine limitation de l'héritage simple proposé par défaut par PHP.

2.2.2.4 MVC

Laravel est basé sur le patron de conception MVC (Modèle-Vue-Controller) qui est un modèle d'organisation du code :

- Le modèle est chargé de gérer les données.
- La vue est chargée de la mise en forme pour l'utilisateur.
- Le contrôleur est chargé de gérer l'ensemble.

Le fonctionnement de ce patron avec Laravel est le suivant

- Le modèle correspond à une table d'une base de données. C'est une classe qui étend la classe **Model** qui permet une gestion simple et efficace des manipulations de données et l'établissement automatisé de relations entre tables.
- La vue est soit un simple fichier avec du code HTML, soit un fichier utilisant le système de template **Blade** de Laravel.
- Le contrôleur se décline en plusieurs catégories : contrôleur classique, contrôleur RESTfull et contrôleur de ressource.

Laravel n'impose pas le modèle MVC, ce qui est très intéressant, car il donne une ouverture sur les autres patrons de conception. Le fait qu'il n'impose pas de patron de conception le place au-dessus de plusieurs Framework PHP.

2.2.2.5 POO(Programmation Orienté Objet)

Laravel est fondamentalement orienté objet. La POO est un design pattern qui s'éloigne radicalement de la programmation procédurale. Avec la POO tout le code est placé dans des classes qui découlent d'interfaces qui établissent des contrats de fonctionnement. Avec la POO on manipule des objets (un objet représente un concept, une idée ou toute entité du monde physique).

Avec la POO, la responsabilité du fonctionnement est répartie dans des classes (Une classe est une collection d'objets), alors que dans l'approche procédurale tout est mélangé. Le fait de répartir la responsabilité évite la duplication du code qui est le lot presque forcé de la programmation procédurale.

2.2.2.6 En résumé

- Un Framework fait gagner du temps et donne l'assurance de disposer de composants bien codés et fiables.
- Laravel est un framework novateur, complet, qui utilise les possibilités les plus récentes de PHP et qui est impeccablement codé et organisé.
- La documentation de Laravel est complète et précise.
- Laravel adopte le patron MVC mais ne l'impose pas, il est totalement orienté objet.

Sachant que nous sommes à notre première utilisation de Laravel, ce Framework nous a permis d'aller plus vite lors de la mise en place de ce projet. Avec une bonne base de connaissance du langage PHP, le temps d'assimilation de Laravel est considérablement réduit.

2.3 Modélisation

2.3.1 UML

2.3.1.1 Identification des acteurs

Un acteur représente un rôle joué par une personne qui interagit avec le système. Par définition, les acteurs sont à l'extérieur du système. Leur recrutement se fait au sein des utilisateurs du système et les responsables de sa configuration. Les acteurs potentiels qui vont interagir avec l'application sont :

Le Doyen : C'est le gérant de l'application, il a une visibilité totale sur tout le système. Son rôle est de spécifier les utilisateurs, définir leurs droits et il a accès à tous les modules de l'application.

La secrétaire principale : Elle est chargée de la manipulation des données de l'application. Ces droits lui sont attribués par le doyen dont elle seconde.

Les opérateurs de saisir : Ils sont chargés de faire toutes les opérations de saisie de l'application principalement la saisie des notes.

NB : Il faut signaler que les enseignants contrôlent la conformité des notes avec leurs états et ceux de l'administration avant toute opération de saisie dans l'application pour garantir la fiabilité des notes. La possibilité pour qu'ils puissent le faire directement n'est pas exclu mais cela ne figure pas dans le cahier des charges que nous avons reçus.

Le système : Il est chargé d'effectuer toutes les tâches automatiques de l'application entre autre le calcul des notes et les paramétrages.

L'étudiant : Il fait son inscription pédagogique et a la possibilité de consulter ses notes.

2.3.1.2 Description des différents cas d'utilisation

L'étude de cas d'utilisation a pour objectif de déterminer ce que chaque utilisateur attend du système. La détermination du besoin est basée sur la représentation de l'interaction entre l'acteur et le système.

Authentification

Objectif : Permet l'identification de chaque utilisateur lui donnant accès aux fonctionnalités propices.

Acteurs principaux : Le doyen, la secrétaire principale et l'opérateur de saisie.

Pré condition : L'utilisateur dispose d'un compte d'accès au système.

Description : L'utilisateur saisi son identifiant et son mot de passe. Il est reconnu par le système. Il peut accéder en fonction de son profil utilisateur.

Création d'une année académique (figure 2.1)

Objectif : L'administrateur principal débute une nouvelle année académique et mets fin à une autre.

Acteurs principaux : Le doyen.

Pré condition : L'utilisateur doit être connecté au système

Description : L'utilisateur dispose d'un formulaire qu'il renseigne avec le libellé de la nouvelle année académique, la date de début et la date de fin. Après avoir renseigné ce formulaire il valide pour créer la nouvelle année universitaire.

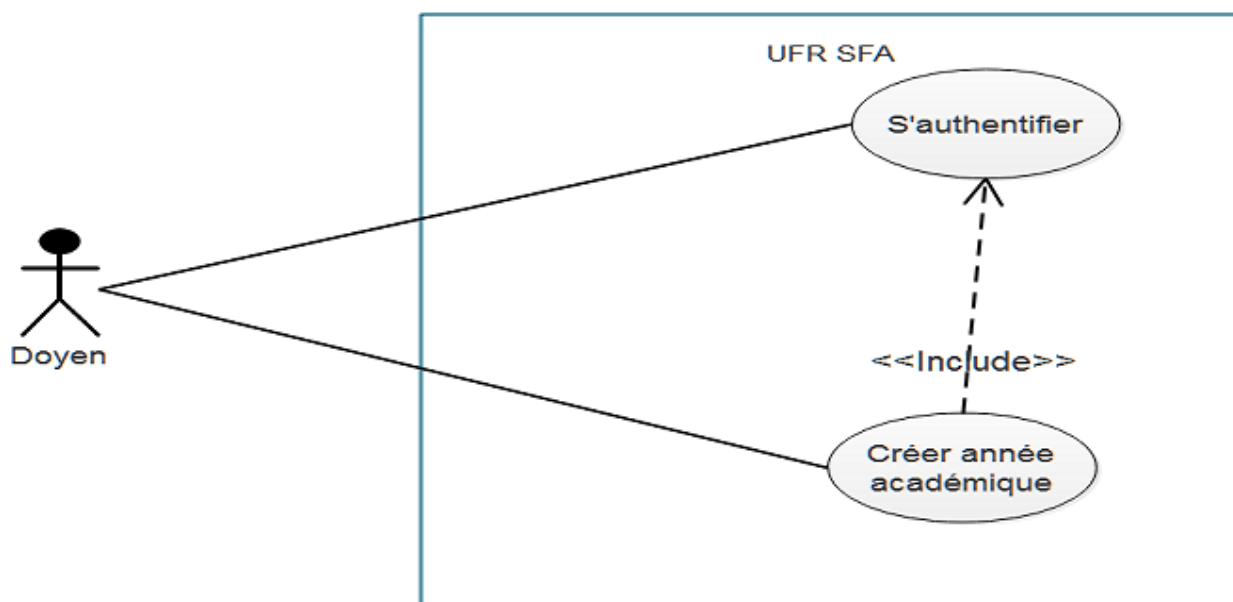


FIGURE 2.1 – Cas d'utilisation Création d'une année académique.

Gestion des Parcours (figure 2.2)

Objectif : Ajouter, modifier et supprimer les parcours.

Acteurs principaux : Le doyen

Pré condition : L'utilisateur doit être connecté au système

Description : L'utilisateur dispose d'un formulaire pour créer de nouveaux parcours. En plus d'une liste de tous les parcours autorisés pendant une année universitaire, il a également la possibilité de modifier et supprimer un parcours.

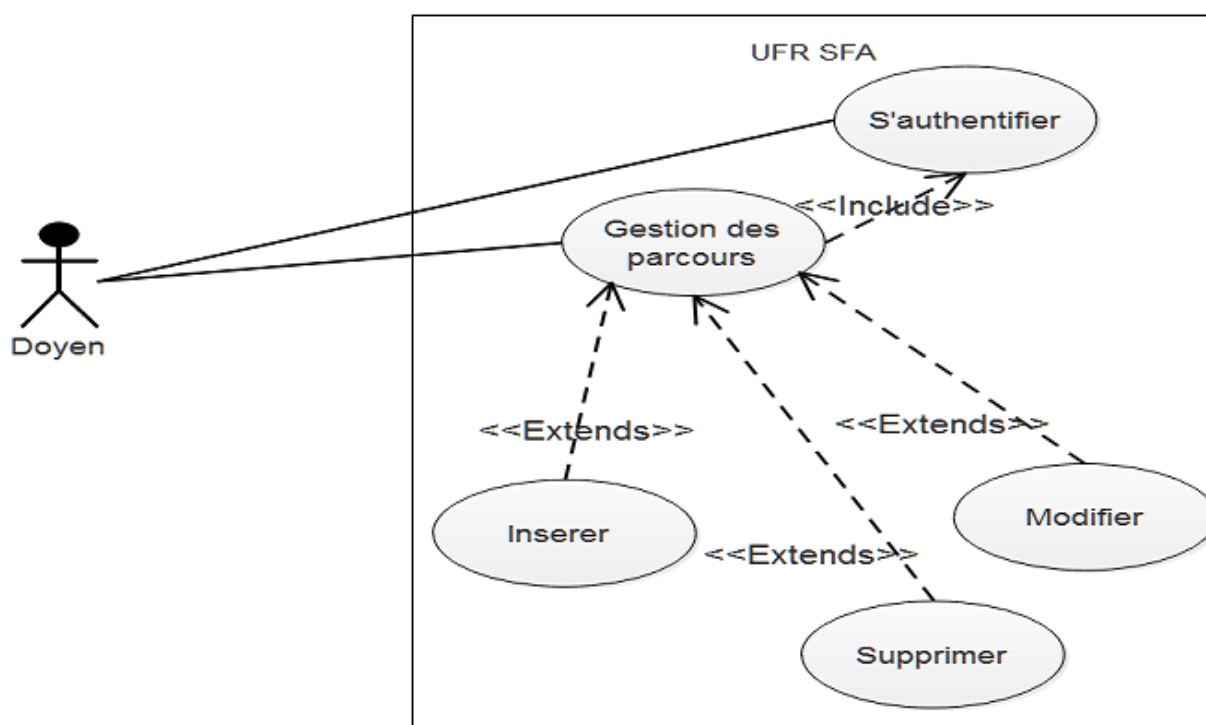


FIGURE 2.2 – Cas d'utilisation de Gestion des parcours.

Cas d'utilisation général de l'étudiant (figure 2.3)

Inscription pédagogique

Objectif : Permettre aux nouveaux étudiants de s'enregistrer dans le système.

Acteurs principaux : Les étudiants

Pré condition : Disposer d'un numéro de carte étudiant.

Description : L'utilisateur dispose d'un formulaire pour s'enregistrer dans la base. Il renseigne les informations concernant sa personne et son cursus scolaire.

Consulter notes

Objectif : Permettre aux étudiants de consulter leurs notes via internet.

Acteurs principaux : Les étudiants

Pré condition : Disposer d'un numéro de carte étudiant.

Description : L'utilisateur dispose d'un formulaire d'authentification dans lequel il renseigne son nom et son numéro de carte d'étudiant pour avoir accès à ses notes.

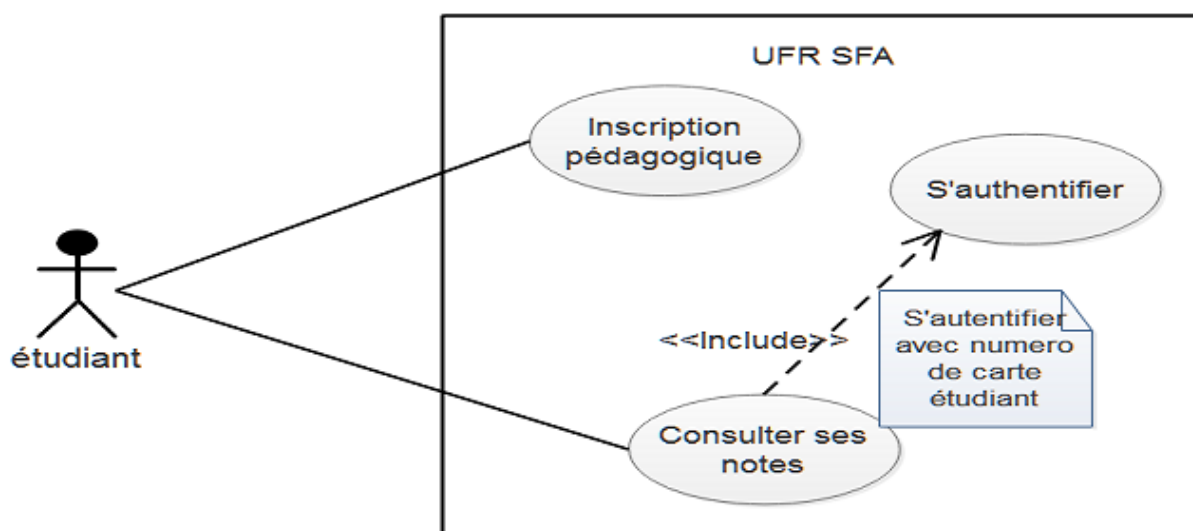


FIGURE 2.3 – Cas d'utilisation général de l'étudiant.

Gestion de UE et ECUE (figure 2.4)

Objectif : Ajouter, modifier et supprimer les unités d'enseignement autorisées pour l'année universitaire.

Acteurs principaux : Le doyen et la secrétaire principale

Pré condition : L'utilisateur doit être connecté au système

Description : L'utilisateur dispose d'un formulaire pour enregistrer les nouvelles unité d'enseignement (UE) et ECUE. Il a également la possibilité de modifier et supprimer une unité d'enseignement (UE).

Gestion des classes (figure 2.5)

Objectif : Permettre à l'utilisateur d'ajouter, modifier ou supprimer des classes.

Acteurs principaux : Le doyen

Pré condition : L'utilisateur doit être connecté au système

Description : Dans le système LMD les classes existent virtuellement car les étudiants s'inscrivent dans les unités d'enseignement (UE). Une classe est un ensemble d'UE, les étudiants s'inscrivent dans ces UE. L'utilisateur lie cet ensemble d'UE à une classe avec un formulaire.

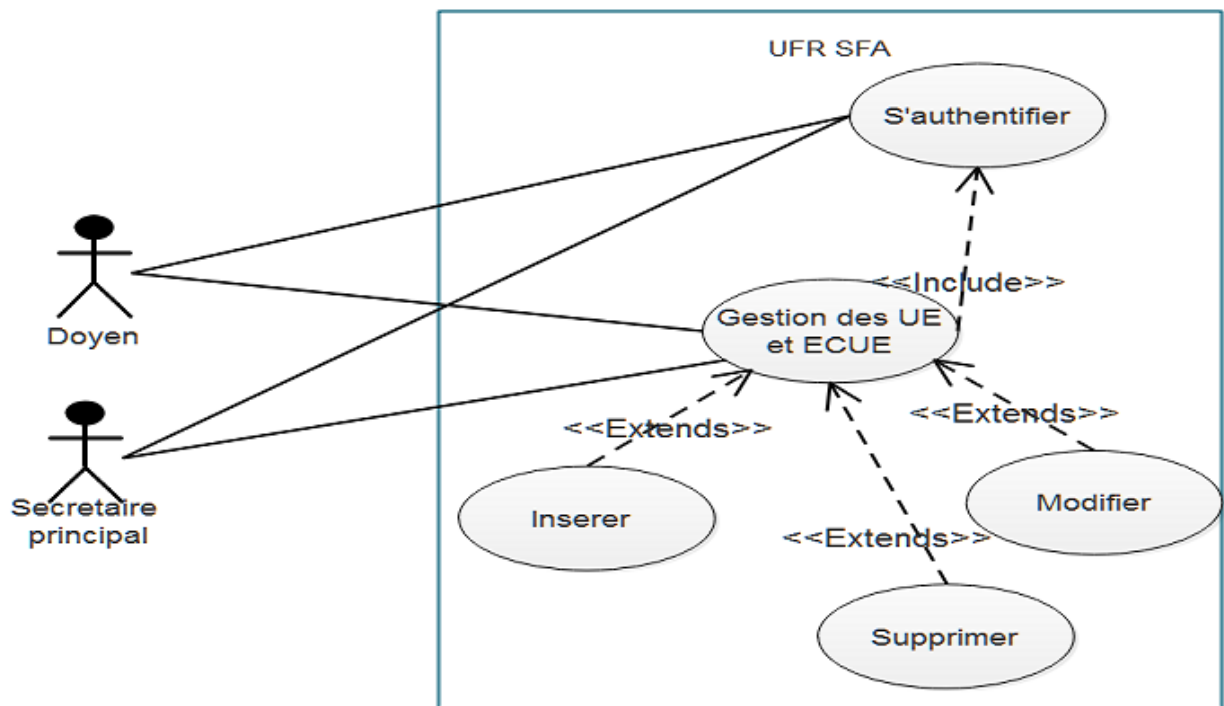


FIGURE 2.4 – Cas d'utilisation de Gestion des unité d'enseignement (UE).

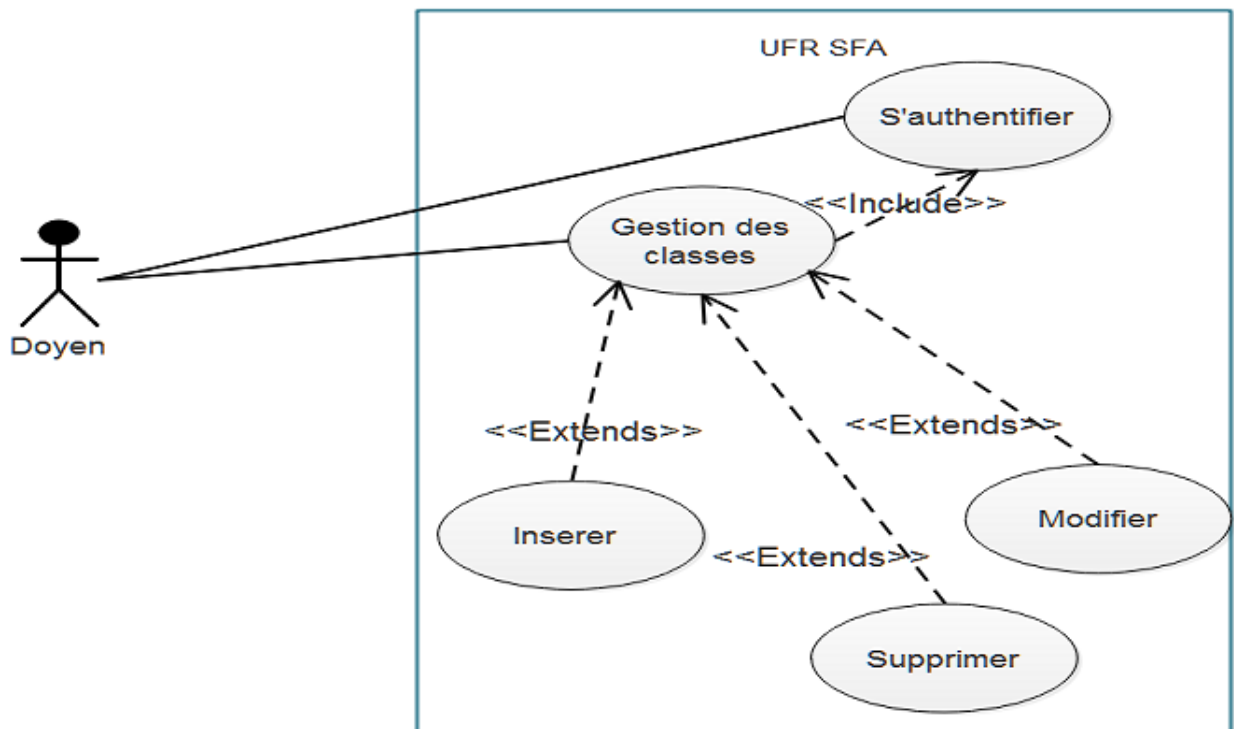


FIGURE 2.5 – Cas d'utilisation de Gestion des classes.

Gestion des notes (figure 2.6)

Objectif : Attribuer des notes aux étudiants après un examen, calculer leurs moyennes et générer leur relevé de note. Pouvoir modifier les notes sur réclamation.

Acteurs principaux : La secrétaire principale, l'opérateur de saisie, le système

Pré-condition : L'utilisateur doit être connecté au système.

Description : L'utilisateur dispose de la liste des étudiants **inscrits par UE** et il renseigne les notes respectives de chacun. La moyenne de chaque UE est calculée de façon automatique par le système. Après le calcul des moyennes, ils peuvent être consulter par UE ou par classe.

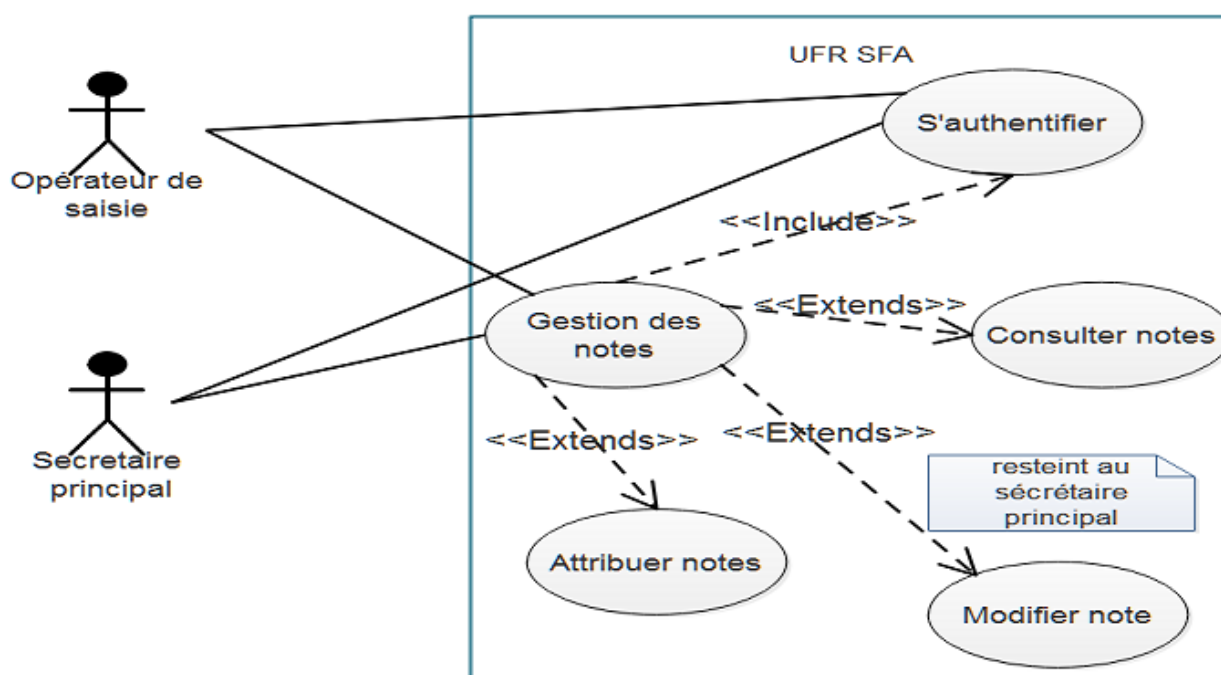


FIGURE 2.6 – Cas d'utilisation de Gestion des notes.

Gestion des étudiants (figure 2.7)

Objectif : Permettre la vérification et la mise à jour des effectifs par classe et par UE. Confirmer les inscriptions des étudiants.

Acteurs principaux : La secrétaire principale

Pré-condition : L'utilisateur doit être connecté au système

Description : L'utilisateur organise les effectifs par parcours, par classes et par UE. Il dispose d'un formulaire d'inscription groupée des étudiants par classe pour valider les inscriptions. Il peut également valider les résultats définitifs, gérer les compensations.

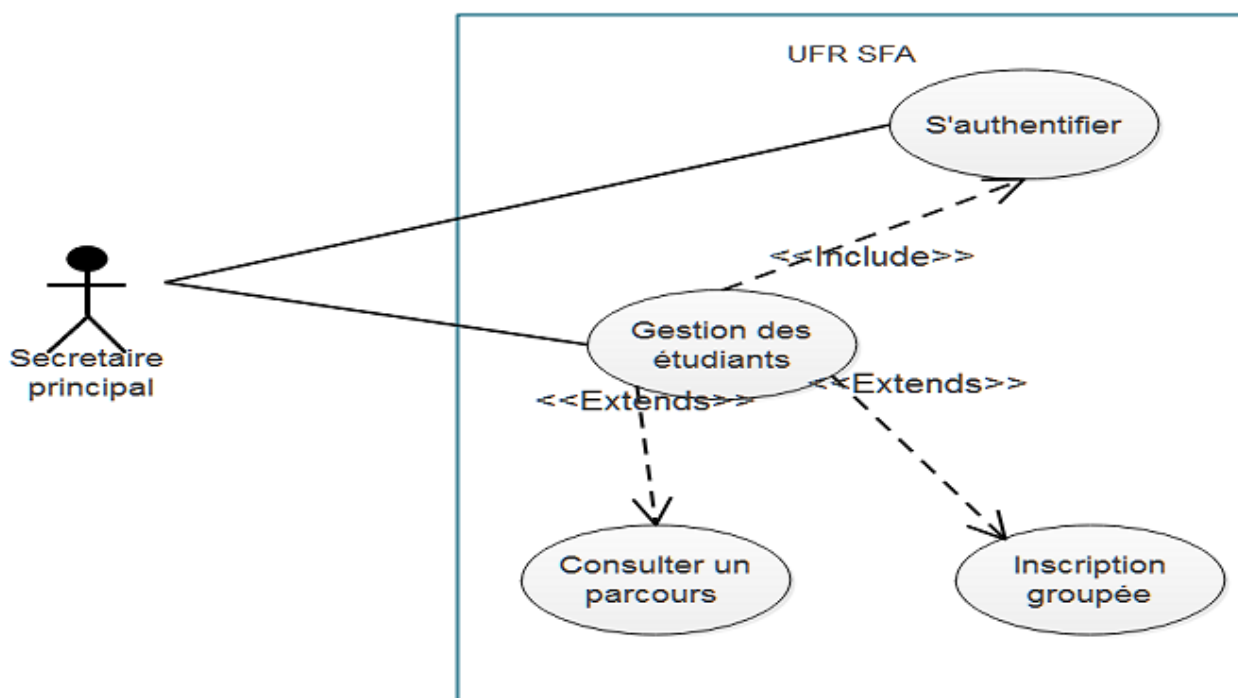


FIGURE 2.7 – Cas d'utilisation de Gestion des étudiants.

Etablir les procès-verbaux (PV) (figure 2.8)

Objectif : Produire les résultats définitifs de l'année académique (établissement du procès-verbal).

Acteurs principaux : Le doyen, la secrétaire principale, le système

Pré-condition : L'utilisateur doit être connecté au système

Description : Lorsque les examens sont terminés pour une année académique et toutes les notes renseignées alors l'utilisateur génère les procès-verbaux. Il a en sa possession la liste des étudiants avec le nombre de crédit validé, cette liste est validée et le PV est généré par le système. Le PV est exportable en format Excel.

Gestion des examens (figure 2.9)

Objectif : Programmer et établir le calendrier des examens

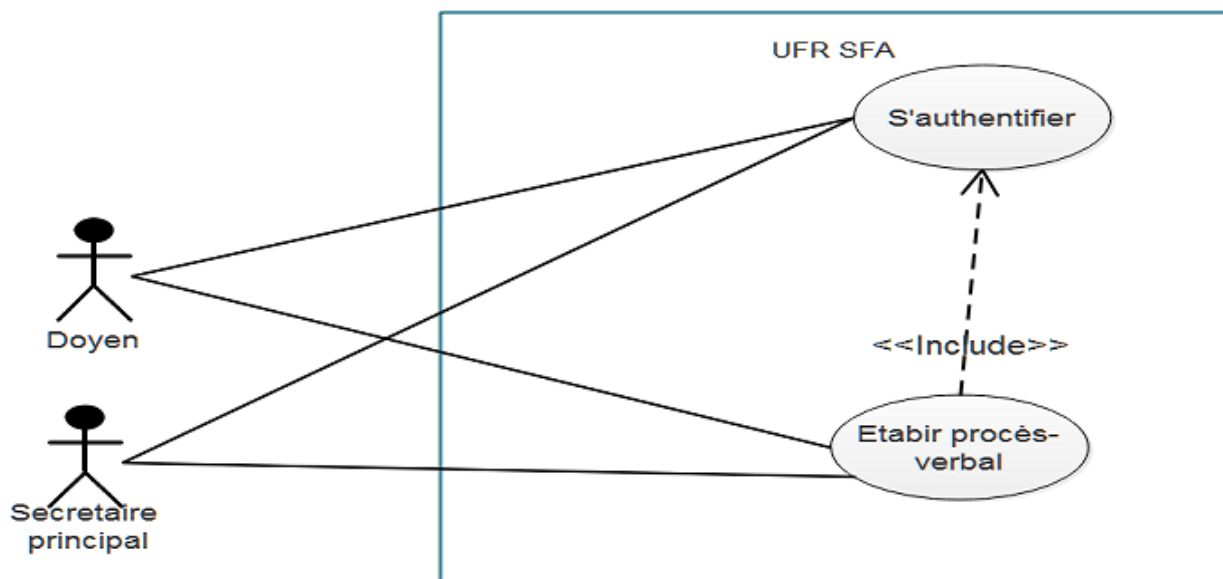


FIGURE 2.8 – Cas d'utilisation Etablir les procès-verbaux.

Acteurs principaux : Le doyen, la secrétaire principale

Pré-condition : L'utilisateur doit être connecté au système

Description : L'utilisateur renseigne la date et l'heure de l'examen dans un formulaire en fonction des jours définis dans la semaine et valide. Le calendrier est ensuite généré et exportable en format PDF.

Paramétrage (figure 2.10)

Objectif : Permettre la reconduite de toutes les données (les unités d'enseignement, les parcours, les classes) à la création d'une nouvelle année universitaire. Permettre également la gestion des utilisateurs de l'application.

Acteur principaux : Le système, le doyen

Pré-condition : Création d'une nouvelle année académique, l'utilisateur doit s'authentifier

Description : A la création d'une nouvelle année académique, les parcours, les classes et les unités d'enseignement de l'année clôturée sont recopier automatiquement pour la nouvelle année. Une vérification manuelle est faite par le doyen pour reconduire ou supprimer certaines données qui ne sont pas autorisée pour la nouvelle année. Le doyen peut créer un nouveau compte utilisateur, consulter, modifier et supprimer un utilisateur.

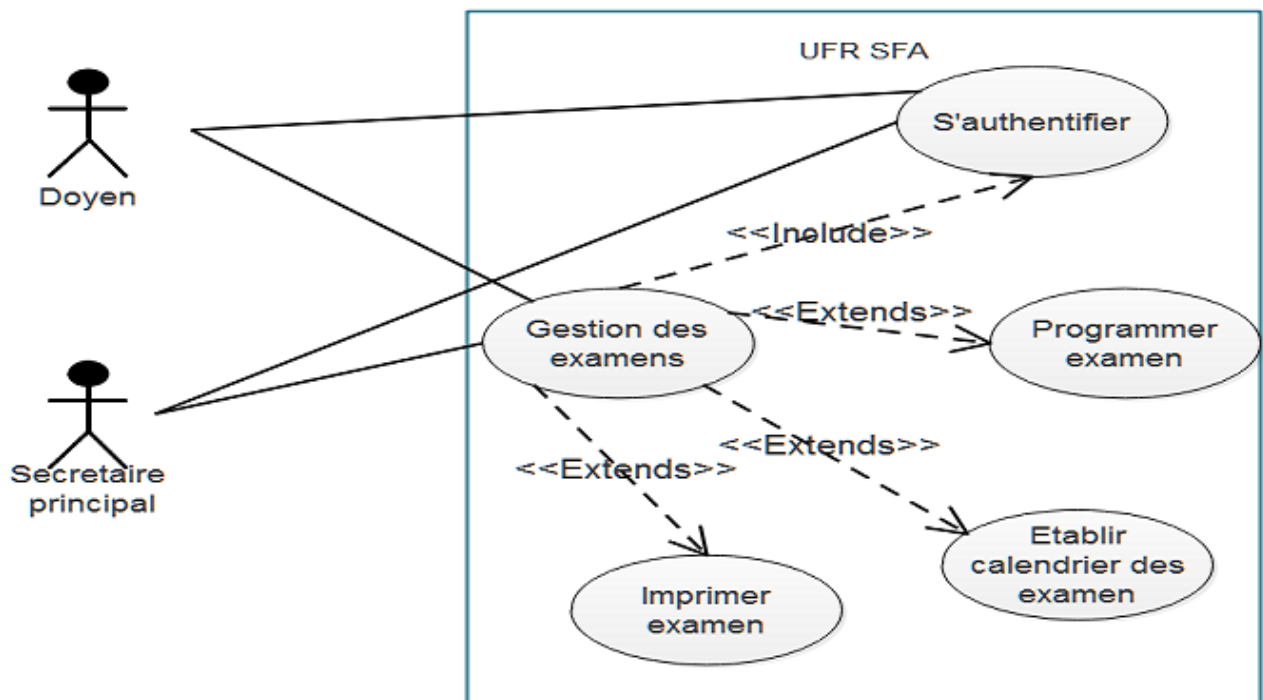


FIGURE 2.9 – Cas d'utilisation de Gestion des examens.

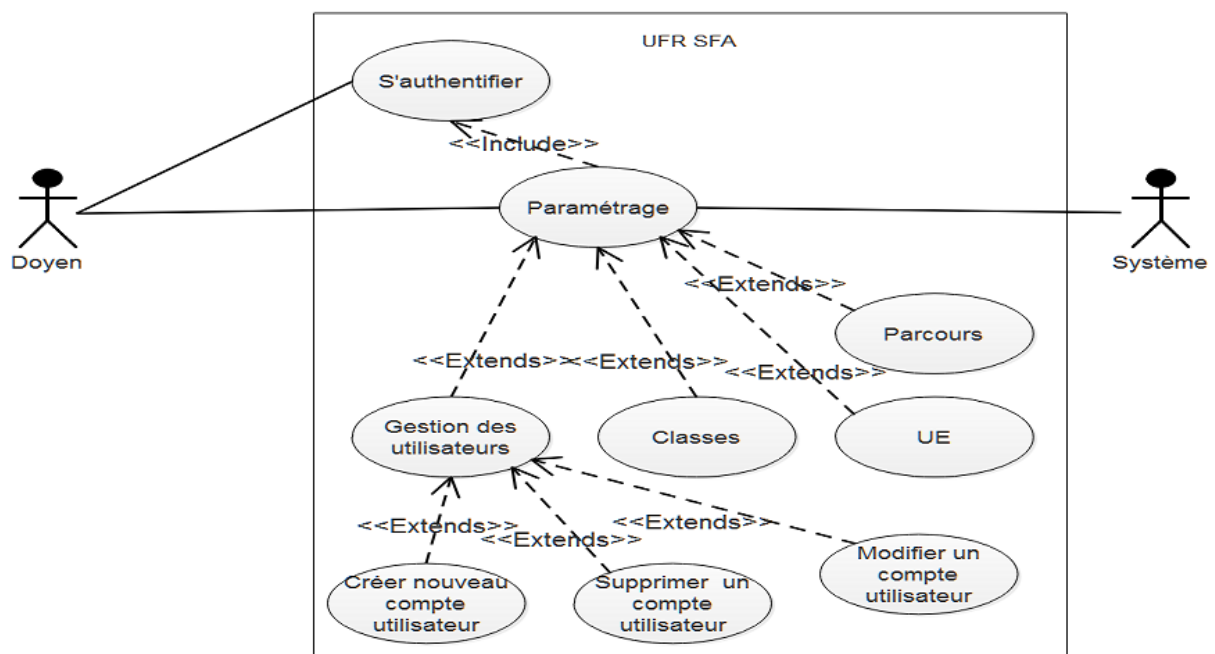


FIGURE 2.10 – Cas d'utilisation Paramétrage

2.3.1.3 Diagramme des classes

Le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

L'étude statique nous permet d'identifier les objets manipulés dans le système. Le diagramme de classe obtenue de notre étude est le suivant :



FIGURE 2.11 – Diagramme des classes.

2.3.1.4 Description des classes métiers du système

Ce diagramme présente les classes métiers de notre application. Ces classes se décrivent comme suit :

Année académique : Elle représente l'**entité centrale** de notre application web. Nous pouvons observer sa relation avec toutes les autres entités du diagramme de classe. Cela se justifie par le fait que le fonctionnement de l'application est fondamentalement lié à une année académique.

Etudiant : Cette entité concerne les étudiants de l'UFR, ils représentent l'effectif que l'application va gérer. Elle récolte tous les attributs (informations personnelles) concernant l'étudiant.

Inscription pédagogique : L'inscription des étudiants dans les unités d'enseignement est réalisée par cette classe. Elle est fondamentale pour le système car elle justifie le fonctionnement même du système LMD dans lequel les étudiants s'inscrivent dans les unités d'enseignements.

Cursus étudiant : Cette classe fait le lien entre un étudiant et une classe. Elle gère les inscriptions des étudiants, établit les listes étudiantes.

Parcours : La classe parcours stocke et organise les différents parcours existant à l'UFR en fonction des grades et des mentions.

Unité d'enseignement(UE) : Cette classe réalise la gestion des UE et ECUE de l'UFR. Elle insère, modifie et supprime des UE.

Notes : Une classe pour prendre en compte les notes d'examens et la gestion de celles-ci. Les notes d'ECUE sont stockées pour chaque UE et le calcul des moyennes d'UE est réalisé par cette classe.

Examen : Elle va permettre d'établir les calendriers d'examen durant l'année académique.

Classe : Cette classe insère, et modifie les classes en fonction des parcours de l'UFR SFA.

User : Cette entité est chargée de la gestion des utilisateurs du système. Elle est chargée de la création d'un utilisateur et de l'attribution des droits à celui-ci.

Voici ici présenté le fonctionnement de notre application d'un point de vue statique, le tableau de la figure 2.1 nous donne un descriptif des classes métiers.

Classe	Attributs	Description
Année académique	Libelle	Libelle de l'année académique
	begindate	Date de début de l'année académique
	enddate	Date de fin de l'année académique
	cloturer	Status de l'année académique
Etudiant	nom	Nom de l'étudiant
	Prenom	Prenom de l'étudiant
	numero carte etudiant	Le numero de carte de l'étudiant
	Lieu de naissance	Le lieu de naissance de l'étudiant
	date de naissance	La date de naissance de l'étudiant
Parcours	mention	Mention
	spécialite	Spécialité
Note	etudiant	l'étudiant concerné par la note
	ecue	l'ECUE concernée
	annee	l'année ou la note est obtenue
	note_examen	la note de l'examen
	session	la session concernée
	valider	Status de la note
Cursus étudiant	annee	L'année académique
	etudiant	l'étudiant concerné
	classe	la classe ou l'étudiant est inscrit
Inscription pedagogique	annee	L'année académique
	etudiant	l'étudiant qui s'inscrit
	ue	l'UE de la classe
Unités d'enseignement(UE)	libelle	Libelle de l'UE
	code	Code de l'UE
	semestre	le semestre ou l'UE est enseignée
Classes	libelle	Libelle de la classe
	parcours	Le parcours où la classe existe
	niveau	Le niveau
Examens	annee	L'année académique
	date_exam	La date de l'examen
	ue	l'UE concernée
Utilisateurs	name	nom et prénom de l'utilisateur
	email	l'email de l'utilisateur
	password	le mot de passe de l'utilisateur
	validated	le status de l'utilisateur
	profil	Le profil de l'utilisateur

TABLE 2.1 – Tableau descriptif des classes métiers.

2.3.2 Mise en place de la base de donnée

2.3.2.1 Modèle physique des données

L'étude du diagramme des classes nous permet d'aboutir sur le **modèle physique des données(MPD)** présenté à la figure 2.12 ci-dessous :

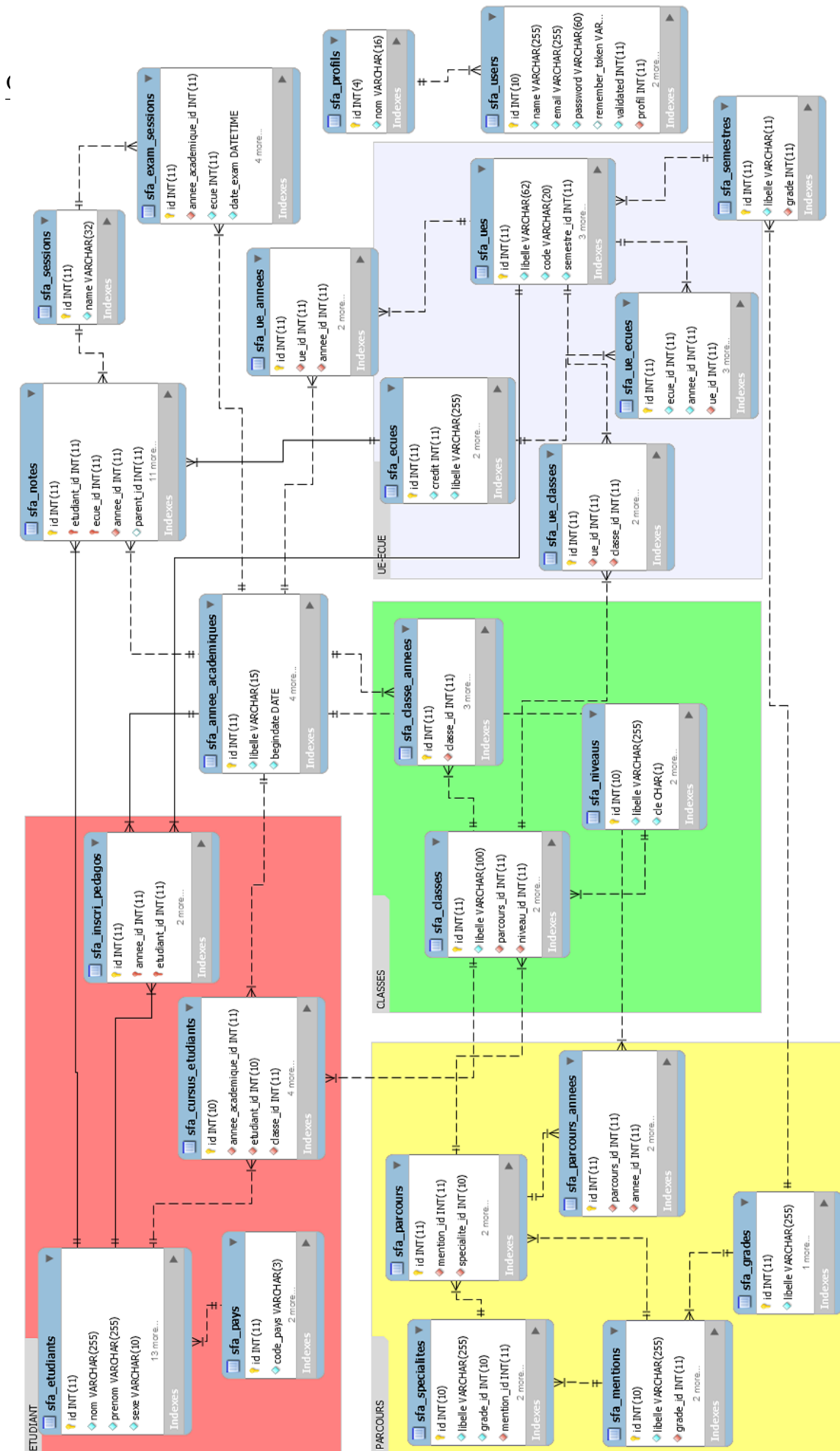


FIGURE 2.12 – Modèle physique des données.

Le modèle physique de donnée nous permet de créer la base la données avec le système de gestion des bases de données (SGBD). Le SGBD permettra de faire les requêtes pour répondre aux besoins de l'application.

2.3.2.2 Choix du Système de Gestion de Base de Données(SGBD)

De nombreux SGBD (Système de Gestion de Bases de Données) sont disponibles sur le marché, partant des SGBD gratuits jusqu'à ceux destinés spécialement aux professionnels, comportant de plus nombreuses fonctionnalités, mais plus coûteux.

Nous pouvons citer entre autre MsAccess, SQL Server, Oracle & TimesTen & DBLibrary, DB2, Interbase (Delphi) Sybase, SQL Anywhere, Adabas, MySQL, PostgreSQL, OpenOfficeBase, Monet, SD-SQL-Server...

MySQL est un système de gestion de base de données relationnelles à la fois robuste et rapide. Une base de données permet de manipuler les informations d'une manière efficace, de les enregistrer, de les trier, de les lire et d'y effectuer des recherches[4].

Notre choix s'est porté sur le SGBD **MySQL** pour les avantages qu'il nous offre :

Un serveur de bases de données relationnelles : Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête.

Rapide, fiable et facile à utiliser : Le serveur MySQL a été développé à l'origine pour gérer de grandes bases de données plus rapidement que les solutions existantes, et a été utilisé avec succès dans des environnements de production très contraintes et très exigeants, depuis plusieurs années. Bien que toujours en développement, le serveur MySQL offre des fonctions nombreuses et puissantes. Ses possibilités de connexions, sa rapidité et sa sécurité font du serveur MySQL un serveur hautement adapté à Internet.

2.4 Conclusion partielle

Cette phase d'étude du projet nous a permis d'élaborer un modèle qui répond aux attentes de l'UFR. Pour ce faire, nous avons utilisé un ensemble d'outils et technologies pour mettre en place la solution.

Nous pouvons dès lors passer à la troisième partie qui est la réalisation de l'application.

Chapitre 3

REALISATION DE L'APPLICATION

Dans ce chapitre, notre travail va porter sur la réalisation et l'implémentation de l'application.

3.1 Environnement logiciels

Pour la réalisation de l'application, notre environnement de développement se présente comme suit :

- Wamp server qui permet de développer des applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données ORACLE en occurrence MySQL qui sont déjà intégrés. Il possède également PHPMyAdmin qui lui permet de gérer plus facilement les bases de données.
- PhpStorm 10.0.3 comme Environnement de développement intégré(IDE).
- MySQL Workbench comme logiciel de conception pour la modélisation informatique de base de données.
- Laravel comme Framework de développement d'application web.
- Edraw Max Fr comme outil de modélisation.

3.2 Architecture et fonctionnement de l'application

Dans ce paragraphe, nous allons présenter l'architecture du Framework Laravel et son impact dans le développement de notre application.

3.2.1 Le modèle MVC de Laravel

Laravel se base effectivement sur le patron de conception MVC, c'est-à-dire modèle-vue-contrôleur. Le modèle interagit avec la base de données, regroupe, traite et gère les données. La vue s'occupe principalement de faire afficher ce que le modèle renvoie. Ensuite, elle s'occupe de recevoir toute interaction de l'utilisateur. Ce sont ces actions que le contrôleur gère. Celui-ci prend en charge la synchronisation du modèle et de la vue. Il capte toutes les activités de l'utilisateur et, en fonction de ces activités, il actionne les changements à effectuer sur l'application.

La séparation des composants d'une application en ces trois catégories permet une clarté de l'architecture des dossiers et simplifie grandement la tâche aux développeurs.

Ainsi la figure 3.1 nous décrit l'architecture MVC de Laravel.

Nous avons une organisation des fichiers comme suit :

3.2.1.1 Routes (Dispatcher) :

Il contient les définitions des chemins d'entrées pour l'utilisateur, autrement dit les URI(identifiant d'une ressource sur un réseau informatique) possibles et les dirige sur la classe définit dans le contrôleur qui doit traiter l'information.

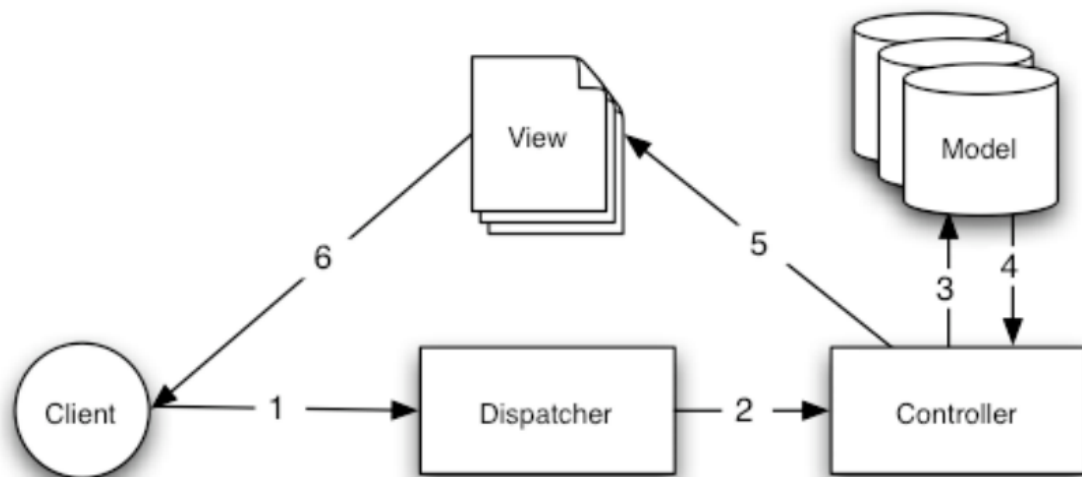


FIGURE 3.1 – Structure MVC de Laravel

3.2.1.2 Modèle :

A Chaque table de notre base de données correspond un modèle unique. Il permet de décrire la méthode d'accès aux données de la base à travers un objet défini par ORM Eloquent (Object-Relational Mapping). Un mapping objet-relationnel ou ORM est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé.

3.2.1.3 Contrôleur

Il permet de récupérer les informations du modèle et de l'envoyer vers la vue pour la mise en forme. Nous avons fait le choix de créer un contrôleur pour certaines tables jugées primordiales pour la bonne marche de notre application.

3.2.1.4 Vue

La vue réceptionne la réponse qui est envoyée par le contrôleur. Laravel a un moteur de Template dénommé **Blade**. Un moteur de Template sert à rassembler le code de présentation (tout ce qui est (x)HTML et CSS) et le code d'application (votre requête en PHP et autres).

3.2.2 Structure

L'emplacement de ces différents fichiers dans Laravel est donné par la structure présentée à la figure 3.2. Le dossier **app** contient tous les éléments nécessaires à la programmation « back-end » de l'application.


























Nom	Modifié le	Type	Taille
 bootstrap	20/01/2016 14:42	Dossier de fichiers	
 database	20/01/2016 14:42	Dossier de fichiers	
 resources	20/01/2016 14:42	Dossier de fichiers	
 tests	20/01/2016 14:42	Dossier de fichiers	
 nbproject	21/01/2016 02:49	Dossier de fichiers	
 storage	06/02/2016 12:25	Dossier de fichiers	
 vendor	22/03/2016 12:10	Dossier de fichiers	
 public	24/03/2016 11:35	Dossier de fichiers	
 config	30/03/2016 11:38	Dossier de fichiers	
 app	09/04/2016 11:15	Dossier de fichiers	
 .idea	02/06/2016 14:31	Dossier de fichiers	
 .env.example	20/01/2016 14:42	Fichier EXAMPLE	1 Ko
	20/01/2016 14:42	Document texte	1 Ko
	20/01/2016 14:42	Document texte	1 Ko
 artisan	20/01/2016 14:42	Fichier	2 Ko
 gulpfile	20/01/2016 14:42	JetBrains PhpStorm	1 Ko
 package.json	20/01/2016 14:42	Fichier JSON	1 Ko
 phpspec.yml	20/01/2016 14:42	Fichier YML	1 Ko
 phpunit	20/01/2016 14:42	Fichier XML	1 Ko
 readme	20/01/2016 14:42	Fichier MD	2 Ko
 server	20/01/2016 14:42	Fichier PHP	1 Ko
 _ide_helper	24/01/2016 06:45	Fichier PHP	370 Ko
 .env	01/03/2016 13:22	Fichier ENV	1 Ko
 composer.json	22/03/2016 11:18	Fichier JSON	2 Ko
 composer.lock	22/03/2016 11:23	Fichier LOCK	137 Ko

FIGURE 3.2 – Structure des fichiers sous Laravel

Dossier	Présentation
app	Il contient tous les éléments nécessaires à la programmation back-end (les Modèles et les Contrôleurs) de l'application. Ici se trouve toute la logique pour la performance de chaque opération. Les Modèles sont directement placés dans le dossier.
bootstrap	Ce dossier contient les paramètres de base pour le démarrage de l'application
config	Ce dossier contient tous les paramètres de configuration de l'application comme la connexion à la base de données, l'inclusion des différentes classes, les paramètres d'email...
database	Ce dossier contient les codes pour les transactions pour la base de données comme la création des tables, la modification des colonnes, ajout de lignes, etc...
public	Ce dossier peut être vu par tout le monde. C'est le répertoire qui est pointé sur le serveur web. Il contient les médias et les autres langages de programmation, soit le CSS, le JS
resource	Ce dossier contient les ressources de l'application comme les vues
Storage	Ce dossier stocke les données locales comme les sessions ; les caches, etc...
tests	Ce dossier contient tous les tests de l'application
vendor	Ce dossier contient les dépendances, les packages additionnels pour les plugins et les fichiers de code du framework Laravel

TABLE 3.1 – Présentation des dossiers de Laravel .

3.2.3 Organisation

L'organisation de notre Framework est présentée à la figure 3.3. Il est subdivisé en deux grandes parties, le dossier **Public** qui est la **partie accessible** aux utilisateurs et un lot de dossier présenté comme la **partie non accessible**. Ce paragraphe va nous permettre de définir tous ces dossiers.

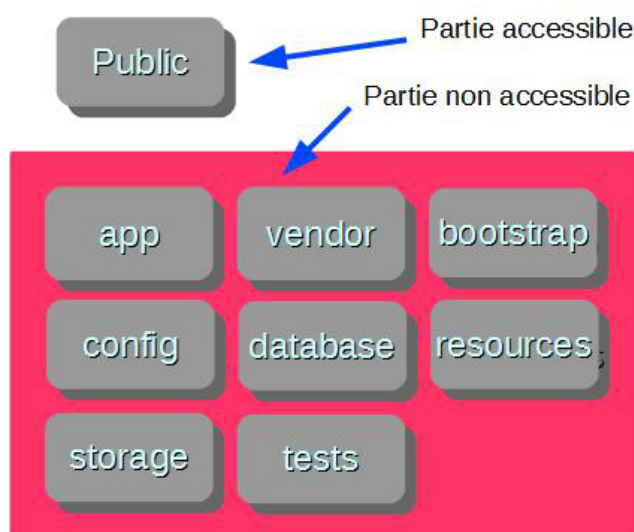


FIGURE 3.3 – Organisation général du framework Laravel.

3.2.3.1 Le dossier app (figure 3.4)

C'est le dossier dans lequel est créée l'application. Il contient les éléments essentiels au développement back-end de notre application.

Commands	20/01/2016 14:42	Dossier de fichiers	
Console	20/01/2016 14:42	Dossier de fichiers	
Events	20/01/2016 14:42	Dossier de fichiers	
Exceptions	20/01/2016 14:42	Dossier de fichiers	
Handlers	20/01/2016 14:42	Dossier de fichiers	
Http	02/05/2016 13:03	Dossier de fichiers	
Providers	20/01/2016 14:42	Dossier de fichiers	
Services	20/01/2016 14:42	Dossier de fichiers	
AnneeAcademique	07/04/2016 02:48	Fichier PHP	1 Ko
Classe	06/02/2016 10:41	Fichier PHP	1 Ko
CursusEtudiants	10/02/2016 02:15	Fichier PHP	1 Ko
Ecue	01/04/2016 15:16	Fichier PHP	1 Ko
Etudiants	29/02/2016 17:17	Fichier PHP	1 Ko
Grade	01/02/2016 00:03	Fichier PHP	1 Ko
InfoPersonne	01/02/2016 02:12	Fichier PHP	1 Ko
Mention	01/02/2016 00:03	Fichier PHP	1 Ko

FIGURE 3.4 – Contenu du dossier app.

- **Http** : Ce dossier contient les contrôleurs, les routes et les middlewares. Les routes sont définies dans le fichier route.php et les contrôleurs dans le dossier Controllers. Ici il s'agit de tout ce qui concerne la communication entre les composants de l'application.
- **Exception** : Ce dossier contient les classes d'exception.
- **Services** : Ce dossier est utilisé pour définir les nouveaux services. Il contient par défaut le service registrar utilisé pour enregistrer les nouveaux utilisateurs.
- **Providers** : Ce dossier enregistre et exécute les services. Il faut remarquer que les fichiers AnnéeAcademique.php, Classe, CoursEtudiants ... situés à la racine du dossier app sont les modèles de notre application. La convention de nommage de Laravel voudrait que le nom de ces fichiers commence par une lettre majuscule. Chaque modèle correspond à une table de notre base de données.

3.2.3.2 Le dossier public

Au départ ce dossier ne contient pas grand chose, essentiellement le fichier index.php qui est la porte d'entrée de l'application :









 css	05/02/2016 01:41	Dossier de fichiers	
 documents	24/03/2016 13:38	Dossier de fichiers	
 fonts	20/01/2016 14:42	Dossier de fichiers	
 doc	20/01/2016 14:42	Fichier HTACCESS	1 Ko
 favicon	22/03/2016 13:40	Foxit Reader PDF ...	0 Ko
 index	20/01/2016 14:42	Icône	0 Ko
 robots	25/02/2016 13:47	Fichier PHP	2 Ko
 robots	20/01/2016 14:42	Fichier TXT	1 Ko

FIGURE 3.5 – Contenu du dossier public.

3.2.3.3 Le dossier resources

Il contient trois(3) sous-dossiers, les dossiers assets, lang et views.

- **le dossier views** : Il contient les fichiers de vue.
- **le dossier lang** : Ce dossier contient les fichiers de langue qui sont chargés de la gestion de la traduction, le fichier de langue par défaut est celui de l'anglais.
- **le dossier assets** : Ce dossier contient un dossier nommé **less**. LESS est un langage dynamique de génération de feuilles de style conçu par Alexis Sellier.




 assets	20/01/2016 14:42	Dossier de fichiers
 lang	12/02/2016 15:16	Dossier de fichiers
 views	01/05/2016 11:16	Dossier de fichiers

FIGURE 3.6 – Contenu du dossier resources.

3.3 Les interfaces utilisateurs

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 allons présenter les interfaces de quelque module de l'application.

3.3.1 Authentification



FIGURE 3.7 – Interface d'authentification.

Cette interface (figure 3.7) permet à l'utilisateur de s'authentifier et de se connecter au tableau de bord de l'application. L'utilisateur doit entrer son identifiant et son mot de passe pour accéder à l'application. En cas d'échec un message d'alerte s'affiche.

En cas de succès, l'utilisateur accède à la page d'accueil de l'application présentée à la figure 3.8. Cet interface est de type tableau de bord avec le menu général de l'application (figure 3.9) et la présentation de tous les modules. Il propose également les liens de navigation rapides à l'utilisateur connecté (figure 3.10).

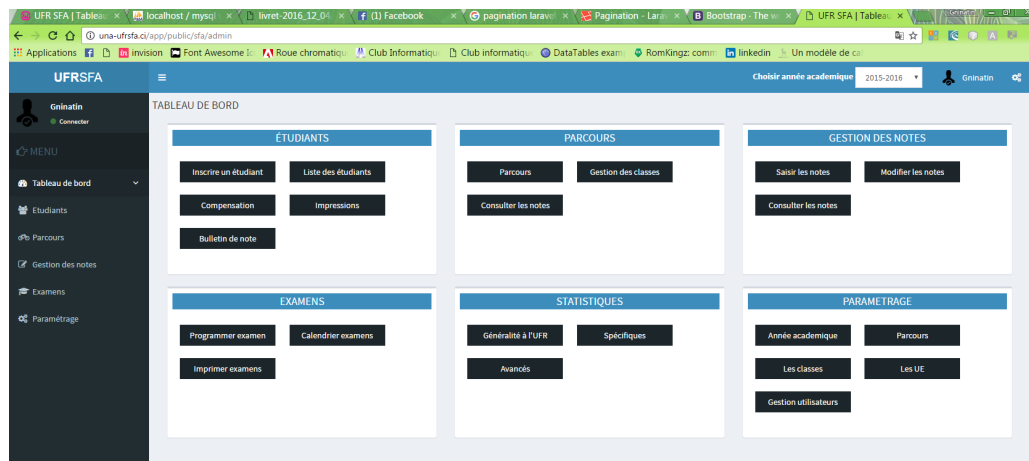


FIGURE 3.8 – Page d'accueil.

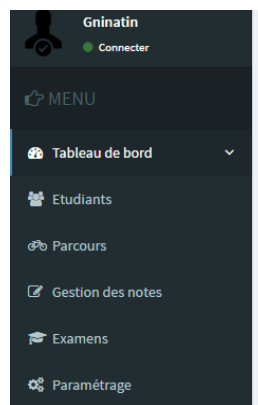


FIGURE 3.9 – Menu général de l'application.

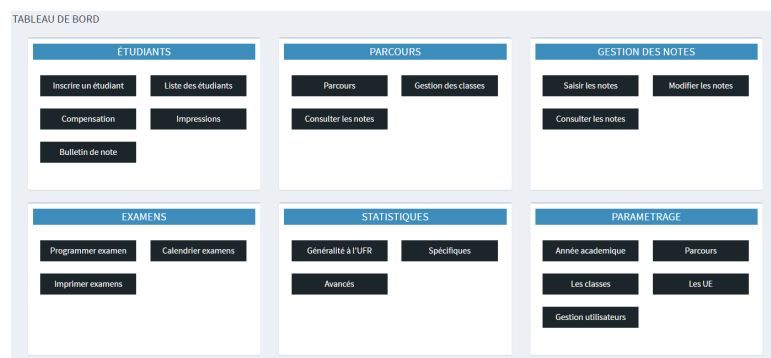


FIGURE 3.10 – Liens de navigation rapides.

3.3.2 Gestion des étudiants

3.3.2.1 Inscription des étudiants et consultation des notes

Le formulaire de la figure 3.11 permet d'enregistrer les étudiants de la première année qui n'existent pas encore dans le système. Il collecte les informations personnelles (nom, prénom, sexe, nationalité etc...) et les informations scolaires (année du bac, numéro du bac, numéro de la carte d'étudiant, établissement d'origine etc...) de l'étudiant.

Le formulaire de la figure 3.12 permet aux étudiants de consulter leurs notes directement en ligne. L'étudiant doit renseigner son numéro de carte étudiant et son nom pour accéder à ses notes. Ainsi nous pouvons voir la liste des notes de l'étudiant **AMON SABLIN XAVIER PRIVAT** à la figure 3.13.

The screenshot shows a web browser window with the URL `calhost81/Laravel5/public/form/nouveau`. The page header features the SFA logo (Sciences Fondamentales et Appliquées) and an orange navigation bar. The main content area is titled 'Inscription Etudiant' and contains two form sections. The 'Informations Personnelles' section has fields for personal details, and the 'Information Scolaire' section has fields for academic details. A blue button labeled 'S'inscrire' is at the bottom.

FIGURE 3.11 – Formulaire d'inscription des étudiants.

SFA
Sciences Fondamentales et Appliquées

Consulter ses notes

Veuillez entrer le nom en Majuscule!

Entrez votre Numero de carte étudiant:

Entrez votre nom:

FIGURE 3.12 – Formulaire de connexion des étudiants.

Consulter ses notes

Veuillez entrer le nom en Majuscule!

Entrez votre Numero de carte étudiant:

Entrez votre nom:

NOM ET PRENOM: AMON SABLIN XAVIER-PRIVAT

SEXE: MASCULIN

DATE DE NAISSANCE: 0000-00-00

LIEU DE NAISSANCE: BOUAKE

CLASSE: LICENCE 2 INFORMATIQUE

UE	CODE	NOTES
PROGRAMMATION STRUCTUREE	PRS2203	13,00/20
ALGEBRE 3	ALB2203	12,00/20
ARITHMETIQUE	ARI 2203	10,00/20
INITIATION A LA GESTION DES PROJETS	GPR 2203	15,00/20
MACROECONOMIE	MAC 2203	12,50/20

FIGURE 3.13 – Liste des notes d'un étudiant.

3.3.3 Gestion des notes

3.3.3.1 Insertion des notes

L'interface présentée par la figure 3.14 donne la liste des classes dont on n'a besoin pour renseigner les notes des étudiants, il faut impérativement choisir une session et cliquer sur une classe ou sur le bouton de la colonne action pour accéder à la liste des UE de la classe présentée à la figure 3.15.

Après un clic sur une UE on accède à un formulaire avec la liste des étudiants inscrit dans l'UE (figure 3.16), ainsi on peut entrer les notes des étudiants.



FIGURE 3.14 – Liste des classes.

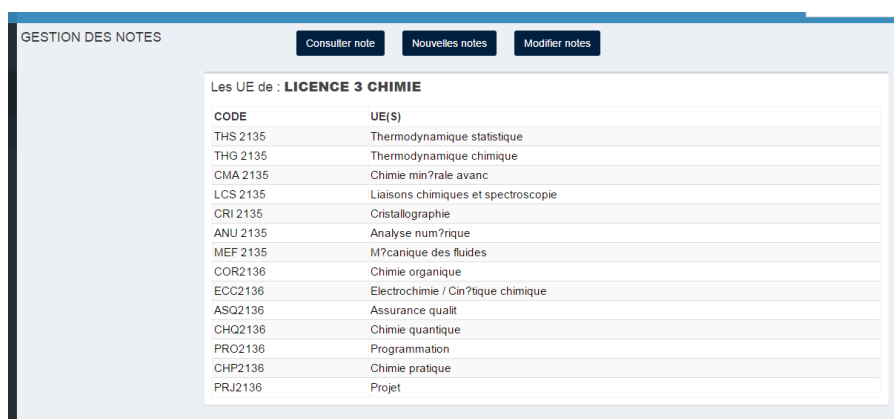


FIGURE 3.15 – Liste des unités d'enseignement(UE).

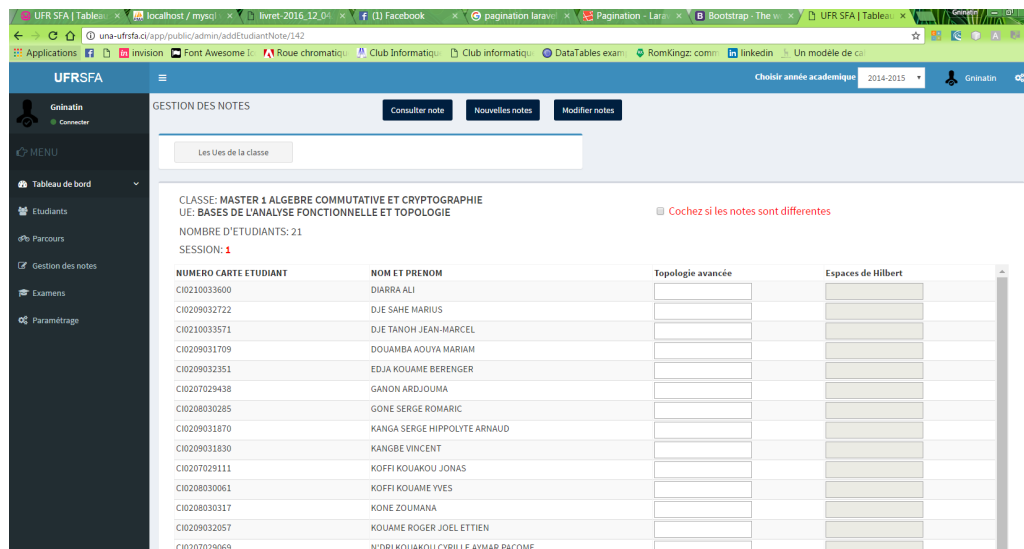


FIGURE 3.16 – Interface d'insertion des notes.

3.3.3.2 Consulter les notes

Le premier bouton de la colonne action donne la liste des étudiants de la classe, à travers cette liste nous pouvons consulter les notes de chaque étudiant. Il suffit de faire un clic sur celui-ci (figure 3.18)

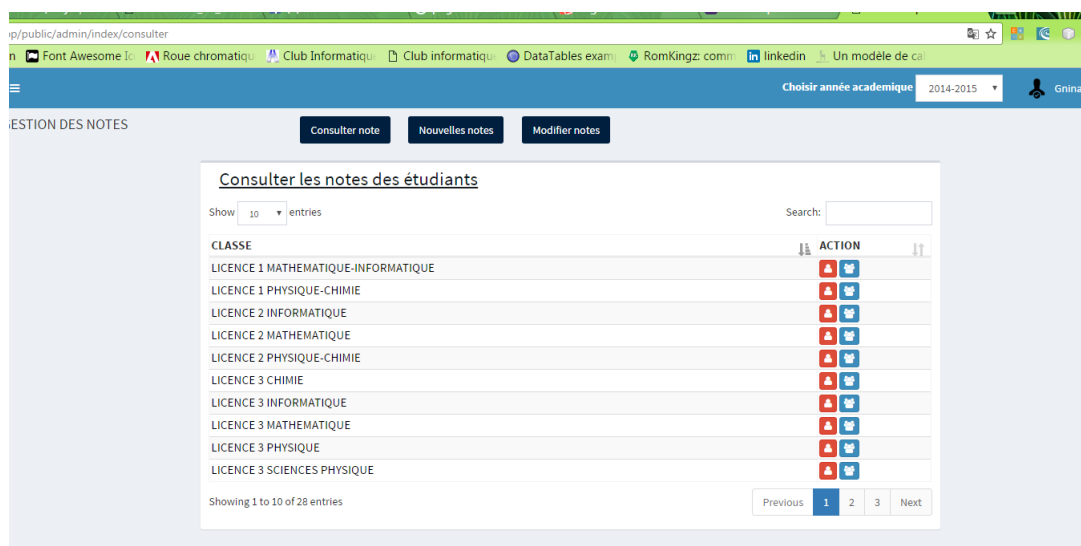


FIGURE 3.17 – Liste des classes.

Le second bouton de la colonne action de la figure 3.17 nous permet de faire la consultation par groupe d'étudiant inscrit dans une unité d'enseignement. Ainsi nous avons la liste des unités d'enseignement de la classe présentée à la figure 3.19. Ce tableau nous permet de consulter les

GESTION DES NOTES

Consulter note Nouveilles notes Modifier notes

Les Etudiants inscrit en : **LICENCE 2 INFORMATIQUE**

Show 25 entries Search:

NUMERO CARTE ETUDIANT	NOM ET PRENOM
CI0204024428	GNAGNE BEDI RAYMOND
CI0204034556	YEO KLOFOR Y ISSOUFOU
CI0208030577	SILUE KAGNAN YACOUBA
CI0209031631	DIKITE MARIAM
CI0209031892	SANOGO ADAMA
CI0209031985	TEHIA KOUAHO N'GUESSAN N.
CI0209032147	GBOGBO YATANA ADOLPHE
CI0209032307	SILUE GNINAHOUOGUI
CI0210033331	BROU N'GUESSAN LUC
CI0210033451	AMANI N'GUESSAN HONORE
CI0210033471	DECHI KOFFI GHISLAIN PACOME
CI0210033552	KONE NAKATANLAN EMMANUEL
CI0210033559	KANE KAMAKA MARIUS
CI0210033579	YAPO AKRE MATHEU JUNIOR
CI0210033591	N'GUESSAN AYA AGGEE EUNICE
CI0210033699	KOUADIO KOFFI SIMEON
CI0210033707	KONAN KOUADIO PATRICK
CI0210033911	KONE MIGNINLIKAN KLINDJO GENEVIEVE
CI0210033994	DJAKO AKASSA MARIUS BERNARD
CI0213035087	NANOU KAKOU JEAN PAUL
CI0213036584	CISSE IBRAHIM KALIL
CI0213036749	AMON SABLIN XAVIER-PRIVAT

Showing 1 to 22 of 22 entries Previous 1 Next

NOM ET PRENOM: GNAGNE BEDI RAYMOND Imprimer

SEXE: MASCULIN

DATE DE NAISSANCE: 0000-00-00

LIEU DE NAISSANCE: DABOU

CODE	UE	NOTES
ALB2203	Algèbre 3	9,50 /20
ANL 2203	Analyse 3	10,00/20
ARI 2203	Arithmétique	10,00/20
GPR 2203	Initiation à la Gestion des projets	14,00/20
MAC 2203	Macroéconomie	11,00/20
MST2203	Probabilités- Statistiques	15,00/20
PRS2203	Programmation structur	12,00/20
APL 2224	Applications linéaires	10,00/20
ASC 2224	Anglais scientifique	12,00/20
BDO 2224	Base de données	14,00/20
CPG2224	Comptabilité générale	16,00/20
ELN 2224	Electronique numérique	9,50 /20
GSA 2224	Gestion des stocks et de l'approvisionnement	11,50/20
ISE 2224	Introduction aux systèmes d'exploitation	10,00/20
MAD 2224	Mathématiques discrètes	10,00/20

FIGURE 3.18 – Interface de consultation des notes par étudiant.

notes par groupe d'admis et par groupe de recalé avec les boutons (admis et recalcé) en rouge. En un clic sur une unité d'enseignement nous avons la liste soit des admis, soit des recalés à droite.

CLASSE: LICENCE 3 CHIMIE		<input checked="" type="radio"/> ADMIS <input type="radio"/> RECALES	UE: Analyse num?rique Etudiant(s) admis: 9 / 9	Imprimer
CODE	LIBELLE			
ANU 2135	Analyse num?rique			
CMA 2135	Chimie min?rale avanc			
CRI 2135	Cristallographie			
LCS 2135	Liaisons chimiques et spectroscopie			
MEF 2135	M?canique des fluides			
THG 2135	Thermodynamique chimique			
THS 2135	Thermodynamique statistique			
ASQ2136	Assurance qualitt			
CHP2136	Chimie pratique			
CHQ2136	Chimie quantique			
COR2136	Chimie organique			
ECC2136	Electrochimie / Cin?tique chimique			
PRJ2136	Projet			
PRO2136	Programmation			

NUMERO CARTE ETUDIANT	NOM ET PRENOM	NOTES
CI0207028271	BAMBA YAYA	15,00/20
CI0208030996	MAH GUEU FULGENCE	18,00/20
CI0210033941	N'DRI KONAN ARCENE	11,00/20
CI0214039324	KOBENAN YAO BIENVENU	14,00/20
CI0213037528	KOIDIO BINDE BORIS	14,00/20
CI0214039195	KOUADJA RIKA JUSTIN	13,00/20
CI0213037408	KISSA OUMAR	11,00/20
CI0213035137	KOBENAN MOUSSA ALEXIS	12,00/20
CI0213037775	N'GATTA ANGE ARMAND VIRGILE	10,00/20

FIGURE 3.19 – Interface de présentation des notes par unité d'enseignement

3.3.3.3 Modifier les notes

La modification des notes se fait par classe avec le tableau de la liste des classes de la figure ci-dessous Ce tableau nous donne accès à la liste des étudiants de la classe (figure 3.21). Nous

Modifier les notes des étudiants	
Show 10 entries	Search: <input type="text"/>
CLASSE	ACTION
LICENCE 1 MATHEMATIQUE-INFORMATIQUE	<input checked="" type="checkbox"/>
LICENCE 1 PHYSIQUE-CHIMIE	<input checked="" type="checkbox"/>
LICENCE 2 INFORMATIQUE	<input checked="" type="checkbox"/>
LICENCE 2 PHYSIQUE-CHIMIE	<input checked="" type="checkbox"/>
LICENCE 3 CHIMIE	<input checked="" type="checkbox"/>
LICENCE 3 INFORMATIQUE	<input checked="" type="checkbox"/>
LICENCE 3 MATHEMATIQUE	<input checked="" type="checkbox"/>
LICENCE 3 PHYSIQUE	<input checked="" type="checkbox"/>
LICENCE 3 SCIENCES PHYSIQUE	<input checked="" type="checkbox"/>
MASTER 1 GENIE INFORMATIQUE	<input checked="" type="checkbox"/>
Showing 1 to 10 of 11 entries	
Previous 1 2 Next	

FIGURE 3.20 – Tableau de modification des notes par classe

avons la liste des unités d'enseignement dans lesquelles l'étudiant est inscrit (figure 3.22). Un clic sur une unité d'enseignement et les notes sont accessibles en modification par ECUE.

GESTION DES NOTES Consulter note Nouvelles notes Modifier notes

Les Etudiants inscrit en : **LICENCE 3 CHIMIE**

Show entries Search:

NUMERO CARTE ETUDIANT	NOM ET PRENOM
CI0207028271	BAMBA YAYA
CI0208030996	MAH GUEU FULGENCE
CI0210033941	N'DRI KONAN ARCENE
CI0213035137	KOBENAN MOUSSA ALEXIS
CI0213037408	KISSA OUMAR
CI0213037528	KOIDIO BINDE BORIS
CI0213037775	N'GATTA ANGE ARMAND VIRGILE
CI0214039195	KOUADJA RIKA JUSTIN
CI0214039324	KOBENAN YAO BIENVENU

Showing 1 to 9 of 9 entries Previous 1 Next

FIGURE 3.21 – Liste des étudiant de la licence 3 chimie

GESTION DES NOTES Consulter note Nouvelles notes Modifier notes

Les notes de : **BAMBA YAYA**

CODE	UE(S)
ANU 2135	Analyse num?rique
CMA 2135	Chimie min?rale avanc
CRI 2135	Cristallographie
LCS 2135	Liaisons chimiques et spectroscopie
MEF 2135	M?canique des fluides
THG 2135	Thermodynamique chimique
THS 2135	Thermodynamique statistique
ASQ2136	Assurance qualit
CHP2136	Chimie pratique
CHQ2136	Chimie quantique
COR2136	Chimie organique
ECC2136	Electrochimie / Cin?tique chimique
PRJ2136	Projet
PRO2136	Programmation

Modifier notes de l'UE de Chimie min?rale avanc

Elements de transition	<input type="text" value="11"/>
Chimie minerale industrielle	<input type="text" value="14"/>

FIGURE 3.22 – Interface de modification des notes

3.3.4 Gestion des parcours et production des états

Cette section va nous permettre de faire les états sur chaque étudiant en fonction de son parcours. Elle nous donne en premier lieu la liste des étudiants d'une classe (figure 3.23).

NUMERO CARTE ETUDIANT	NOM ET PRENOM	ACTION
CI0204024428	GNAGNE BEDI RAYMOND	
CI0204034556	YEO KLOFOR Y ISSOUFOU	
CI0208030577	SILUE KAGNAN YACOUBA	
CI0209031631	DIAKITE MARIAM	
CI0209031892	SANO GO ADAMA	
CI0209031985	TEHIA KOUAHO N'GUESSAN N.	
CI0209032147	GBOGBO YATANA ADOLPHE	
CI0209032307	SILUE GNINAHOUOGUI	
CI0210033331	BROU N'GUESSAN LUC	
CI0210033451	AMANI N'GUESSAN HONORE	

FIGURE 3.23 – Liste des étudiants de la licence 2 informatique

Sur cette figure nous avons trois boutons dont les actions sont :

- le bouton **Etudiants admis** donne la liste des étudiants qui ont validé la totalité des crédits (60). Cette liste est donnée à la figure 3.24
- le bouton **Etudiants récalés** donne la liste des étudiants ayant validé moins de 60 crédits (figure 3.25).
- le bouton **PV de la classe** permet la génération du procès-verbal provisoire de la classe. Un aperçu du procès verbal est présenté à la figure 3.26. Le procès-verbal peut être exporter en fichier excel (figure 3.27), ceci est l'action du bouton **Exporter CSV**.

La colonne action de la figure 3.23 va afficher les notes de l'étudiant par semestre présentées à la figure 3.28, ce tableau va permettre de générer les relevés de note en un seul clic sur le bouton **imprimer le relevé** en format PDF.

Ainsi nous avons le relevé qui est généré et peut être imprimé (figure 3.29).

Liste des étudiants ayant validé exactement 60 crédits en LICENCE 2 INFORMATIQUE

ETUDIANT	NOMBRE DE CREDIT
AMON SABLIN XAVIER-PRIVAT	60
CISSE IBRAHIM KALIL	60
DECHI KOFFI GHISLAIN PACOME	60
KONE NAKATANLAN EMMANUEL	60
KONE MIGNINLIKAN KLINDJO GENEVIEVE	60
NANOU KAKOU JEAN PAUL	60
N'GUESSAN AYA AGGEE EUNICE	60
GBOGBO YATANA ADOLPHE	60
SILUE GNINAHOUOGUI	60
SILUE KAGNAN YACOUBA	60

[Valider résultat](#)
[Imprimer résultat](#)

FIGURE 3.24 – Liste des admis

Liste des étudiants ayant validé moins de 60 crédits en LICENCE 2 INFORMATIQUE

ETUDIANT	NOMBRE DE CREDIT
DIAKITE MARIAM	48
GNAGNE BEDI RAYMOND	51
KANE KAMAKA MARIUS	50
KONAN KOUADIO PATRICK	52
SANOGO ADAMA	49
TEHIA KOUAHO N'GUESSAN N.	50
YAPO AKRE MATHEU JUNIOR	57
YEO KLOFOR Y ISSOUFOU	57
AMANI N'GUESSAN HONORE	56
BROU N'GUESSAN LUC	56
DJAKO AKASSA MARIUS BERNARD	57
KOUADIO KOFFI SIMEON	47

[Valider résultat](#)
[Imprimer résultat](#)

FIGURE 3.25 – Liste des récalés

CHAPITRE 3. REALISATION DE L'APPLICATION

Imprimer procès verbalFaire les passagesCompensationExporter CSV

entries

Search:

	NUM CARTE	NOM	PRENOM	DATE- NAISS	LIEU NAISS	PRS2203		MST2203		MAC 2203		GPR 2203	
						MOY	CRE	MOY	CRE	MOY	CRE	MOY	C
	CI0213036749	AMON	SABLIN XAVIER-PRIVAT	0000-00-00	BOUAKE	12	4	15	6	12	3	10	
	CI0210033707	KONAN	KOUADIO PATRICK	0000-00-00	AHOUGNASSOU	10	4	17	6	14.5	3	10	
	CI0208030577	SILUE	KAGNAN YACOUBA	0000-00-00	BOUAKE	15	4	18	6	15	3	16	
	CI0209032307	SILUE	GNINAHOUOGUI	0000-00-00	NONGOTIENKAHA	14.5	4	16	6	12	3	15	
	CI0209032147	GBOGBO	YATANA ADOLPHE	0000-00-00	BENOMBA/BIANKOUMA	11	4	11	6	10	3	14	
	CI0210033591	N'GUESSAN	AYA AGGEE EUNICE	0000-00-00	BOUAKE	13	4	14	6	14	3	12	
	CI0213035087	NANOU	KAKOU JEAN PAUL	0000-00-00	ABENGOUROU	19	4	10.5	6	11	3	11	
	CI0210033911	KONE	MIGNINLIKAN KLINDJO GENEVIEVE	0000-00-00	NIAKARAMANDOUGOU	12	4	10	6	14	3	14	
	CI0210033552	KONE	NAKATANLAN EMMANUEL	0000-00-00	TREICHVILLE	10	4	18	6	13	3	10	
	CI0210033471	DECHI	KOFFI GHISLAIN PACOME	0000-00-00	AGBOVILLE	16	4	11	6	11	3	12	

10 of 22 entries

Previous123Next

FIGURE 3.26 – Procès-verbal provisoire

FICHIERACCUEILINSERTIONMISE EN PAGEFORMULESDONNÉESRÉVISIONAFFICHAGE

Calibri11A⁺

Collier

Presse-papiers

Police

Alignement

Nombre

Mise en forme conditionnelle

Mettre sous forme de tableau

Styles de cellules

Insérer Supprimer Format

Cellules

Trier et Rechercher et filtrer > sélectionner >

Édition

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	NOM	PRENOM	NUMERO CARTE	DATE NAISSANCE	LIEU DE NAISSANCE	PRS2203	MST2203	MAC 2203	GPR 2203	ARI 2203	ANL 2203	ALB2203	SEMES
2	AMON	SABLIN XAVIER-PRIVAT	CI0213036749	0000-00-00	BOUAKE	12	15	12	10	16	15,5	12,5	
3	KONAN	KOUADIO PATRICK	CI0210033707	0000-00-00	AHOUGNASSOU	10	17	14,5	10	12,5	10	16,5	
4	SILUE	KAGNAN YACOUBA	CI0208030577	0000-00-00	BOUAKE	15	18	15	16	15	19	14,5	
5	SILUE	GNINAHOUOGUI	CI0209032307	0000-00-00	NONGOTIENKAHA	14,5	16	12	15	17	18,5	11,5	
6	GBOGBO	YATANA ADOLPHE	CI0209032147	0000-00-00	BENOMBA/BIANKOUMA	11	11	10	14	13	11	15	
7	N'GUESSAN	AYA AGGEE EUNICE	CI0210033591	0000-00-00	BOUAKE	13	14	14	12	10	12,5	10,75	
8	NANOU	KAKOU JEAN PAUL	CI0213035087	0000-00-00	ABENGOUROU	19	10,5	11	11	13	10,5	13,75	
9	KONE	MIGNINLIKAN KLINDJO GENEVIEVE	CI0210033911	0000-00-00	NIAKARAMANDOUGOU	12	10	14	14	14	13,25	16	
10	KONE	NAKATANLAN EMMANUEL	CI0210033552	0000-00-00	TREICHVILLE	10	18	13	10	13	10,75	13	
11	DECHI	KOFFI GHISLAIN PACOME	CI0210033471	0000-00-00	AGBOVILLE	16	11	11	12	11	13	10	
12	CISSE	IBRAHIM KALIL	CI0213036584	0000-00-00	ABOBO	11	10	16	17	12	12,5	14	
13	YEO	KLOFOR Y ISSOUFOU	CI0204034556	0000-00-00	ABOBO	12	16	10	10	9	15,78	10,5	
14	YAPO	AKRE MATHEU JUNIOR	CI0210033579	0000-00-00	BOUAKE	15	12	11	10	8	11,725	15,5	
15	DIAKO	AKASSA MARIUS BERNARD	CI0210033994	0000-00-00	DABOU	13	17	10	13	13	16,5	13,5	
16	AMANI	N'GUESSAN HONORE	CI0210033451	0000-00-00	BROUKRO/BROBOM	12	11	12	11	16	17,03	11,5	
17	BROU	N'GUESSAN LUC	CI0210033331	0000-00-00	AHOUNAN/ M'BATTO	15	16	10	12	10,55	14	14,5	
18	GNAGNE	BEDI RAYMOND	CI0204024428	0000-00-00	DABOU	12	15	11	14	10	10	9,5	
19	TEHIA	KOUAHO N'GUESSAN N.	CI0209031985	0000-00-00	BOUAKE	9	8	17	10	15	13,5	10,5	
20	KANE	KAMAKA MARIUS	CI0210033559	0000-00-00	BOUAKE	10	12	12	15,6	10	8,5	10	
21	SANOGO	ADAMA	CI0209031892	0000-00-00	AHOUANNOU/DIVO	11	7	13	11,7	14	11,5	11,5	
22	DIAKITE	MARIAM	CI0209031631	0000-00-00	BOUAKE	13,5	9	10	11	15	16,5	8,5	
23	KOUADIO	KOFFI SIMEON	CI0210033699	0000-00-00	NIENKRO	8,5	16	12	15	12	7,5	13,5	

FIGURE 3.27 – Procès-verbal provisoire exporté en excel

3.4 Conclusion partielle


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 jugé les plus importantes et en décrivant brièvement la planification de notre projet.

PARCOURS DE GNAGNE BEDI RAYMOND					
CREDITS VALIDES			CREDITS NON VALIDES		
51 / 60			9 / 60		
	CODE DE L'UE	UE	CREDITS	NOTE	ANNEE
SEMESTRE 3	MST2203	Probabilit?s- Statistiques	6	15	2014-2015
	GPR 2203	Initiation à la Gestion des projets	2	14	2014-2015
	ANL 2203	Analyse 3	6	10	2014-2015
	PRS2203	Programmation structur	4	12	2014-2015
	MAC 2203	Macro?conomie	3	11	2014-2015
	ARI 2203	Arithm?tique	3	10	2014-2015
	ALB2203	Alg?bre 3	6	9.5	2014-2015
SEMESTRE 4	BDO 2224	Base de donn?es	5	14	2014-2015
	APL 2224	Applications lin?aires	4	10	2014-2015
	MAD 2224	Math?matiques discr?tes	3	10	2014-2015
	GSA 2224	Gestion des stocks et de l?approvisionnement	4	11.5	2014-2015
	CPG2224	Comptabilit? g?n?rale	4	16	2014-2015
	ASC 2224	Anglais scientifique	2	12	2014-2015
	ISE 2224	Introduction aux syst?mes d?exploitation	5	10	2014-2015


FIGURE 3.28 – Tableau des notes par semestre

public/getReleve/34/5

Font Awesome Icons Roue chromatique Club Informatique Club informatique DataTables exam RomKingz: comm linkedin Un modèle de cal



République de Cote d'Ivoire
Union - Discipline - Travail
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique



Unité de Formation et de
Recherche des Sciences
Fondamentales et
Appliquées

**DEMANDE DE RELEVÉ DE NOTES PROVISOIRE DE
LICENCE-SFA PARCOURS : LICENCE 2 INFORMATIQUE**

Nom et Prénoms : GNAGNE BEDI RAYMOND

Date et lieu de naissance : DABOU

N° de Carte d'étudiant : CI0204024428 Redoublant : OUI / NON

	Code de l'UE	Unités d'Enseignement(UE)	Nbre de crédit	Note	Mention	Année
SEMESTRE 3	ARI 2203	Arithmétique	3	10	PASSABLE	2014-2015
	ALB2203	Algèbre 3	6	9.5	ECHEC	2014-2015
	MST2203	Probabilités- Statistiques	6	15	BIEN	2014-2015
	GPR 2203	Initiation à la Gestion des projets	2	14	BIEN	2014-2015
	ANL 2203	Analyse 3	6	10	PASSABLE	2014-2015
	PRS2203	Programmation structur	4	12	ASSEZ BIEN	2014-2015
	MAC 2203	Macroéconomie	3	11	PASSABLE	2014-2015
SEMESTRE 4	ISE 2224	Introduction aux systèmes d'exploitation	5	10	PASSABLE	2014-2015
	ELN 2224	Electronique numérique	3	9.5	ECHEC	2014-2015
	BDO 2224	Base de données	5	14	BIEN	2014-2015
	APL 2224	Applications linéaires	4	10	PASSABLE	2014-2015
	MAD 2224	Mathématiques discrètes	3	10	PASSABLE	2014-2015
	GSA 2224	Gestion des stocks et de	4	11.5	PASSABLE	2014-2015

FIGURE 3.29 – Un relevé de note en PDF

Conclusion général

L'objectif de notre projet de fin d'étude était de concevoir et implémenter une application web de gestion des effectifs et des notes. Cette phase pratique de notre formation consacrée à l'analyse, à la conception et à la programmation nous a conduits à partir des problèmes constatés, à satisfaire les besoins réels en adéquation avec les objectifs prescrits et les contraintes rencontrées.

Nous avons appliqué au maximum les règles de bases permettant d'avoir une application performante. Nous avons utilisé le langage UML pour modéliser le système en suivant les différentes étapes, MySQL comme SGBD et le langage PHP pour implémenter notre application. Du fait que PHP est un langage interopérable et flexible, l'application sera portable et opérationnel indépendamment de toute plateforme.

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 étudié, d'approfondir nos connaissances dans le monde du développement d'applications informatiques.

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

Bien que la plupart des fonctionnalités majeures de notre application soient achevées, certaines sont encore en attente à cause de certaines données que nous attendons encore de l'UFR SFA.

Nous n'avons pas la prétention d'avoir réalisé un travail parfait. C'est pourquoi, nous voudrions au-delà des imperfections et insuffisances que vous constaterez et rencontrerez, tenir compte de vos remarques, critiques, et suggestions afin de rendre possible la perfection de ce projet.

Bibliographie

- [1] François Xavier Bois. *PHP 5 le guide complet*. Micro Application, 2007.
- [2] Maurice Chavelli. Découvrez le framework php laravel, 2016.
- [3] Joseph Gabay. *Merise et UML pour la modélisation des systèmes d'information*. Dunod, 2004.
- [4] Laura Thomson Luke Welling. *PHP & MySQL*. CAMPUSPress, 2004.