

2020-10

# Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants

Wasso Kiseki, Dominique

UB, FSI

---

<https://repository.ub.edu.bi/handle/123456789/306>

*Téléchargé depuis le dépôt institutionnel officiel de l'Université du Burundi*

République du Burundi  
Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique



Université du Burundi  
Faculté des Sciences de l'Ingénieur

Département de TIC  
Mastère en Génie  
Informatique

Année Académique :  
2018-2019

---

---

**MISE EN PLACE D'UN SYSTEME DE GESTION  
UNIVERSITAIRE DES ENSEIGNANTS,  
ENSEIGNEMENTS ET ETUDIANTS**

---

---

**MEMOIRE**  
Présenté Par  
**WASSO KISEKI Dominique**  
à la  
**Faculté des Sciences de l'Ingénieur (FSI)**  
*En vue de l'obtention du grade de*  
**MASTER**  
en  
**Génie Informatique**

---

---

***Soutenu le 09/octobre/2020, devant le jury composé de :***

- |                                |                  |
|--------------------------------|------------------|
| 1. Docteur SAHINGUVU William   | : Président      |
| 2. Prof. NDIKUMAGENGE Jérémie  | : Vice-Président |
| 3. Docteur NDAYISABA Longin    | : Secrétaire     |
| 4. Prof. MANIRABONA Audace     | : Directeur      |
| 5. Docteur MUKESHIMANA Michele | : Co-Directrice  |
| 6. Docteur KIBEYA Hassan       | : Membre         |

**Bujumbura, Octobre 2020.**

### **Identification des membres du jury**

Ce travail a été évalué par le jury composé de:

1. Docteur SAHINGUVU William Président;
2. Professeur NDIKUMAGENGE Jérémie Vice-Président;
3. Docteur NDAYISABA Longin Secrétaire;
4. Professeur MANIRABONA Audace Directeur;
5. Docteur MUKESHIMANA Michele Co-Directrice;
6. Docteur KIBEYA Hassan Membre.

### **Epigraphe**

Vers toi, Yahweh, j'élève mon âme, mon Dieu.  
En toi je me confie : que je n'aie pas de confusion !  
Que mes ennemis ne se réjouissent à mon sujet !  
Non, aucun de ceux qui espèrent en toi ne sera confondu ;  
Ceux-là seront confondus qui sont infidèles sans cause.  
Yahweh, fais-moi connaître tes voies,  
Enseigne-moi tes sentiers.  
Conduis-moi dans ta vérité, et instruis-moi,  
Car tu es le Dieu de mon Salut ;  
Tout le jour en toi j'espère.  
Souviens-toi de ta miséricorde, Yahweh, et de ta bonté  
Car elles sont éternelles.

**Psaume 25 :1-6**

« Il n'y a quelque chose de pire dans la vie que de n'avoir pas  
Réussi : c'est de n'avoir pas essayé »

**FRANKLIN DELANOLT Roosevelt**

### **Dédicaces**

A nos chers parents KISEKI Tabukima Jean-Paul et BITAMWILO Emeride, pour tout l'amour dont vous nous avez entouré, pour tout ce que vous avez fait pour nous, nous ferons de notre mieux pour rester un sujet de fierté à vos yeux, avec l'espoir de ne jamais vous décevoir que ce modeste travail reste et soit l'exaucement de vos vœux tant formulés et de vos prières quotidiennes.

A notre chère femme Béatrice IKANGAMINO Butoga et nos chers enfants Jean-Paul KISEKI, Marie-Rose FAIDA, Jeanne Bitamwilo, Eloge Lukeka, Zacharie Wenga Kaleshi, Benjamin Itongwa et Julien Dunia nous vous devons ce que nous sommes aujourd'hui grâce à votre amour, prière, patience et vos innombrables sacrifices. Que ce modeste travail, soit pour vous une petite compensation et reconnaissance envers ce que vous avez fait d'incroyable pour nous.

A nos cher(e)s frères et sœurs particulièrement à notre très chère sœur Zawadi Kiseki et toute sa famille, vous occupez des places particulières dans notre cœur.

Nous vous dédions ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès.

**Dominique WASSO Kiseki**

## **Remerciements**

Nous adressons nos remerciements à Dieu Tout Puissant le Créateur du ciel et de la terre d'avoir guidé notre démarche à la réalisation du présent travail, par le témoignage que nous avons à son égard. C'est celui qui nous a gardé dès le ventre de notre mère. Dans sa bonté, il a dirigé ce travail jusqu'à la fin.

Nos remerciements s'adressent à notre directeur de ce mémoire, le Docteur Audace Manirabona pour sa disponibilité, son soutien perpétuel, ses précieuses directives et ses idées scientifiques. Qu'il trouve ici le fruit de nos efforts comme témoignage de notre gratitude et de notre respect.

Nous exprimons également toute notre reconnaissance à notre co-directrice, Docteur Michèle Mukeshimana qui a participé dans une grande partie de ce travail, par ses encouragements continus et ses pertinentes remarques. Qu'elle soit assurée de notre très grand respect et du plaisir que nous avons eu à travailler avec lui.

Nous remercions aussi tous nos enseignants de l'UB en particulier ceux de FSI, du Département de TIC plus particulièrement ceux du mastère en Génie Informatique qui ont contribué à notre formation. Qu'ils trouvent ici le résultat de leurs efforts.

Nous n'oublions pas de saluer fortement tous les membres de notre grande famille notamment nos parents notre père Jean-Paul Kiseki Tabukima et notre mère Eméride Bitamwilo de nous avoir encouragé à poursuivre ce cycle de mastère. Qu'ils trouvent dans ce mémoire une récompense de leurs sacrifices.

Nous n'oublions pas de saluer fortement les soutiens de notre femme Béatrice Ikangamino Butoga et nos enfants Jean-Paul Kiseki Wilondja, Marie-Rose Wasso Faïda, Jeanne-Elodie Wasso Bitamwilo, Eloge Wasso Lukeka, Zacharie Wasso Kaleshi-Wenga, Wasso Itongwa Benjamin et Julien Dunia de nous avoir encouragé et toléré nos absences continues et répétitives surtout pendant les moments difficiles de l'incendie d'Irambo. Qu'ils trouvent dans ce mémoire une récompense de leurs sacrifices et patiences.

Nos remerciements s'adressent également à notre très chère sœur Zawadi Kiseki et toute sa famille, à notre belle mère Jeanne Nyabutunga, à nos très chers oncles Zéphyrin Kikukama, Honoré Kukwasubi et Léandre Kubota, à nos cousins et cousines, à maman Jacqueline Lutala, papa Yali et toute l'équipe de la Bibliothèque centrale, à toute notre promotion et à tous nos amis en particulier : Thomas Mubengwa, Laurent Assani, Alain Mukamba, André Shabani, Chriss Ngama, Médard Kilundu, Gulain Buduge, Christophère Bujiriri, Landry Fakage, Hugue Amisi et Samuel Ahana.

**Dominique WASSO Kiseki**

## Résumé

Ce travail de fin d'études s'inscrit dans le cadre d'un projet de recherche portant sur la thématique gestion universitaire dans lequel sont présentées les différentes étapes de conception et de réalisation d'une application web pour la Gestion Universitaire. Cette application a pour but de mettre en place un système d'information permettant de fonctionner en intranet. Elle doit générer, télécharger et imprimer automatiquement les informations nécessaires, archiver électroniquement de données et documents y relatifs tels que les inscriptions en ligne. Elle doit également permettre les gestionnaires et les agents des services concernés dans la prise de décision quant à l'évolution des enseignements, les notes obtenues par les étudiants pour chaque cours dans chaque promotion, avoir une traçabilité sur le parcours d'un utilisateur dans un établissement. Elle doit également permettre de connaître le nombre des enseignants utilisés dans les départements et leurs charges horaires, d'où le pourquoi, la nécessité et la conception de cette application de gestion universitaire dans les services académiques.

**Mots-clés :** Application, Gestion, Etudiants, Enseignement et Enseignant.

### Abstract

This end of study work is part of a research project on the theme of university management in which the different stages of design and production of a web application for University Management are presented. The purpose of this application is to set up an information system to operate on the intranet. It must generate, download and automatically print the necessary information, electronically archive data and related documents such as online registrations. It must also allow managers and agents of the departments concerned in making decisions about the development of lessons, the marks obtained by students for each course in each class, have traceability on the journey of a user in a establishment. It must also make it possible to know the number of teachers used in the departments and their hourly charges, hence the why and the need and the design of this university management application in academic services.

**Keywords:** Application, Management, Students, Teaching and Teacher.

## Table des matières

Identification des membres du jury .....	I
Epigraphe .....	III
Dédicaces .....	IV
Remerciements .....	V
Résumé .....	VI
Abstract .....	VI
Table des matières .....	VII
Liste des tableaux .....	X
Liste des figures .....	XI
Sigles et abréviations .....	XII
Avant propos .....	XIII
Chapitre I. INTRODUCTION GENERALE ET CONTEXTE .....	1
I.1. Généralités et contextes .....	1
I.2. Objectifs du projet .....	2
I.2.1. Objectif global du travail .....	2
I.2.2. Objectifs spécifiques .....	2
I.3. Problématique et hypothèses .....	3
I.4. Solutions proposées .....	5
I.5. Résultats attendus .....	5
I.6. Apports scientifiques et technologiques .....	5
I.7. Domaine d'application .....	6
I.8. Méthodologie utilisée .....	6
I.9. Outils mathématiques utilisées .....	6
Chapitre II. CONCEPTION, MODELISATION DE LA GESTION UNIVERSITAIRE .....	7
II.1. Généralités sur enseignement, enseignant et étudiants (3e) .....	7
II.1.1. La Gestion des enseignements .....	7
II.1.2 La gestion des enseignants .....	8
II.1.3 La gestion des étudiants .....	8
II.2. Les différentes entités utilisées dans le système .....	9
II.2.1 Objets des diagrammes de classes .....	9
II.2.2. Objets communs .....	10
II.2.3. Liste des attributs des classes(entités) .....	11

II.2.4. Liste des opérations des classes (méthodes) :.....	13
II.2.5. L'Explication des entités utilisées .....	15
II.3. Héritage [5].....	16
II.4. Utilisation de l'UML.....	18
II.4.1 Objectifs d'UML .....	18
II.4.2 Identification des acteurs.....	18
II.5. Identification des messages .....	20
II.6. Identification de cas d'utilisation [8] .....	20
II.7. Diagramme de séquence.....	23
II.7.1. Diagramme de séquence de cas « s'authentifier ».....	23
II.7.2. Diagramme de séquence de cas « consulter site ».....	24
II.7.3. Diagramme de séquence de cas « inscription ».....	25
II.8. Diagramme d'activité.....	26
II.9. LES DIAGRAMMES DE CLASSES.....	27
Chapitre III. FORMALISMES MATHEMATIQUES : Avec Algèbre Relationnelle .....	33
III.1. Contextes.....	33
III.2 L'objectif de l'utilisation de l'algèbre relationnelle .....	34
Conclusion .....	45
Chapitre IV. L'IMPLEMENTATION DU SYSTEME « GESTUNI » .....	46
IV.1. Les technologies utilisées .....	46
IV.1.1 Le langage Java.....	46
IV.1.2. Principales caractéristiques du langage Java .....	46
IV.2. Stratégie d'héritage avec JPA [14].....	47
IV.2.1 Mise en relation d'entités .....	50
IV.2.2 Les plates-formes de Java.....	50
IV.3 Des applications Java avec le framework [op-cit] .....	51
b) Le Spring MVC [15].....	52
IV.4. Quelques implémentations.....	53
IV.5 Les outils utilisés .....	58
IV.5.1. Bootstrap.....	58
IV.5.2 WampServer .....	58
IV.5.3 Powerdesigner [18].....	58
IV.5.4 Apache POI.....	58

IV.5.5 iText [20] .....	59
IV.5.6. Le MySQL [21] .....	59
IV.6. Maquettage du système.....	60
IV.7. Conclusion .....	68
CONCLUSION GENERALE ET RECOMMANDATIONS .....	69
1. Conclusion générale .....	69
2. Recommandations .....	71
Références bibliographiques et webographiques .....	72

### **Liste des tableaux**

Tableau 2.1. Liste des entités qui seront développés.....	9
Tableau 2.2. Liste des associations .....	10
Tableau 2.3. Liste des attributs des entités .....	11
Tableau 2.4. Liste des opérations ou méthodes .....	14
Tableau 2.5. La présentation du dictionnaire de données .....	28

## Liste des figures

Figure 2.1. La présentation de l'héritage.....	17
Figure 2.2. Le diagramme de cas d'utilisation .....	22
Figure 2.3. Diagramme de séquence de cas « s'authentifier ».....	23
Figure 2.4. Illustration du diagramme de séquence consulter site .....	24
Figure 2.5. Illustration du diagramme de séquence de cas « inscription ».....	25
Figure 2.6. Diagramme d'activité globale .....	26
Figure 2.7. Diagramme de classe .....	32
Figure 3.1. La structure de la table Personne .....	35
Figure 3.2. Affiche dans Personne où type_personne=ENSEIGNANT .....	36
Figure 3.3. Affiche des personnes où sexe=M .....	36
Figure 3.4. La projection .....	37
Figure 3.5. L'union des tables classe_etudiant et cours_enseignant .....	38
Figure 3.6. L'intersection des tables : cours_enseignant et cours_classe .....	39
Figure 3.7. La différence entre la table cours_classe et cours_enseignant .....	40
Figure 3.8. Le produit cartésien des cours affectés dans une classe .....	41
Figure 3.9. La jointure entre département, cours et classe .....	42
Figure 3.10. Le Theta-jointure sur département et classe .....	43
Figure 3.11. La jointure naturelle .....	44
Figure 4.1. La présentation de la Table Personne .....	49
Figure 4.2. Architecture MVC de l'application Gestuni .....	52
Figure 4.3. Architecture de l'entité Personne .....	54
Figure 4.4. Architecture de l'entité Classe .....	56
Figure 4.6. Architecture de l'entité Document_Etudiant .....	57
Figure 4.7. La page d'accueil .....	60
Figure 4.8. La page d'authentification .....	61
Figure 4.9. Les rôles de l'administrateur .....	61
Figure 4.10. Les rôles du Chef-Dep .....	62
Figure 4.11. Les rôles ou Espace enseignant .....	62
Figure 4.12. Les rôles d'un Etudiant .....	63
Figure 4.13. Les deux grilles de délibération .....	63
Figure 4.14. L'envoi de cotes par l'enseignant .....	64
Figure 4.15. L'identité de l'étudiant .....	64
Figure 4.16. Le candidat complète l'identité de ses parents .....	65
Figure 4.17. Etudes faites du candidat .....	65
Figure 4.18. Documents du candidat. ....	66
Figure 4.19. Choix de la promotion .....	66
Figure 4.20. Confirmation de son inscription .....	67
Figure 4.21. Réception du mot de passe et du login .....	67
Figure 4.22. Brouillon .....	68

### **Sigles et abréviations**

CRUD	: Create, Read, Update et Delete
CSS	: Cascading Style Sheets
DCU	: Diagramme de Cas d'Utilisation
DSI	: Direction des Systèmes d'Informations
EJB	: Entreprise Java Beans
FSI	: Faculté des Sciences de l'Ingénieur
GESTUNI	: Gestion Universitaire
HTML	: Hypertext Markup Language
Java ME	: Java Micro Edition
JDK	: Java Developpement Kit
JEE	: Java Entreprise Edition
JPA	: Java Persistence API
JRE	: Java Runtime Environnement
JS	: Java Script
JSP	: Java ServerPages
JVM	: Java Virtuel Machine
MVC	: Modèle, Vue, Contrôleur
NTIC	: Nouvelles Technologies de l'Information et de la Communication
OMG	: Object Management Group
ORM	: Object Relational Mapping
SI	: Système d'Information
SGBD	: Système de Gestion de Bases des Données
SGBDR	: Système de Gestion de Bases des Données Relationnelles
TIC	: Technologies de l'Information et de la Communication
URL	: Uniform Ressource Locator
UB	: Université du Burundi
UML	: Unified Modeling Language
www	: word wide web
w3c	: word wide web consortium

### **Avant propos**

Par ce travail, nous avons voulu mettre en place une application web pour la Gestion Universitaire. Cette application a pour but de mettre en place un système d'information permettant de fonctionner en intranet.

La force de travail est qu'après la réalisation de cette application, elle permet de télécharger, imprimer, archiver électroniquement les données et documents y relatifs tels que les inscriptions en ligne. Elle permet également aux gestionnaires dans la prise de décision quant à l'évolution des enseignements, les notes obtenues par les étudiants. Elle doit également permettre de connaître le nombre des enseignants utilisés dans les départements et leurs charges horaires.

Néanmoins, la nécessité et la conception de cette application de gestion universitaire dans les services académiques, d'autres questions restent non répondues comme la gestion des charges horaires, la gestion du personnel, la gestion financière...

## **Chapitre I. INTRODUCTION GENERALE ET CONTEXTE**

### **I.1. Généralités et contextes**

Les techniques de programmation n'ont cessé de progresser depuis l'époque de la programmation en langage binaire à nos jours. La technologie objet est donc la conséquence ultime de la modularisation dictée par la maîtrise de la conception et de la maintenance d'applications toujours plus complexes. Cette nouvelle technique de programmation a nécessité la conception de nouvelles méthodes et outils de modélisation tel que UML et le MVC pour ce qui est de notre cas.

Pour ce faire, nous procéderons dans le contexte de notre recherche scientifique, de développer une application dans le domaine de l'informatique en utilisant le langage de programmation java accompagné des frames-works tels que (Springboot, bootstrap...), qui est aujourd'hui généralement entendu comme contribution au développement des systèmes d'informations. Nous définirons également des outils opérationnels pour la réalisation et la modélisation de notre application tels que l'UML et Powerdesigner pour préciser la mission et la performance de l'application produite.

Le système d'information intègre les dimensions organisationnelles, humaines et technologiques de la gestion de l'information d'entreprise ou d'un pays. Grace aux systèmes d'informations novateurs modernes, tels que les entrepôts de données et les méthodes de gestion de connaissances dont Datamining, l'information est devenue une marchandise dans ce monde. Or, dans le contexte mondial actuel, aucun organisme, aucune institution, aucun pays ne pourrait ignorer le rôle et la place des NTIC dans la maîtrise de la concurrence et l'amélioration des services ainsi que les dividendes que l'usage de ces dernières apportent au quotidien. Dans la gestion universitaire, dans son secteur académique, le besoin d'avoir une application de gestion des inscriptions en ligne, des enseignements, enseignants et étudiants reste un besoin d'actualité. Ceci est dû à l'absence d'un système informatique qui tient compte de besoin de ses secteurs. Ce retard rend toujours la gestion de cette dernière manuelle et parfois les usagers en difficultés de trouver certaines informations électroniques disponibles, tel est l'importance de notre recherche qui est de pouvoir doter les institutions universitaires d'une application informatique qui leur facilitera la tâche dans l'exercice des fonctions académiques.

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

## **I.2. Objectifs du projet**

La réalisation de ce projet nous permettra de mettre en place une application de gestion universitaire en ligne pour la bonne tenue du système d'information dans le secteur académique, étudiant, enseignant et enseignement.

### **I.2.1. Objectif global du travail**

L'objectif principal de ce projet de recherche scientifique est la mise en place d'une application informatique de gestion universitaire dans le secteur académique.

### **I.2.2. Objectifs spécifiques**

Pour notre projet, nous nous sommes donnés comme objectifs spécifiques de mettre en place un système d'information pour la gestion universitaire permettant de :

- Gérer les enseignants, enseignements et étudiants ;
- Fonctionner en intranet ;
- Générer, télécharger et imprimer automatiquement les informations nécessaires ;
- Archiver électroniquement les données et documents y relatifs ;
- Aider les gestionnaires et les agents des services concernés dans la prise de décision quant à l'évolution des enseignements, les notes obtenues par les étudiants pour chaque cours dans chaque promotion ainsi que l'impression de bulletin de notes pour chaque étudiant ;
- Avoir une traçabilité sur le parcours d'un utilisateur dans un établissement.

### **I.3. Problématique et hypothèses**

Les institutions d'Enseignements Supérieurs et Universitaires font parties intégrantes des établissements qui sont touchés par le domaine de la nouvelle technologie de l'information et de la communication et dont l'informatique pourra beaucoup aider dans leur gestion quotidienne. Aujourd'hui pour qu'une entreprise ou institution puisse bien évoluer et s'adapter à un environnement en mouvance continue et perpétuelle générée par une croissance et évolution technique et scientifique rapide, elle se trouve dans l'obligation d'établir une stratégie basée sur des moyens et techniques adaptés aux nouvelles technologies de l'information et de la communication. Dans ce monde en mutations continues, un des défis auquel nous tous, êtres humains, devons faire face, est de comment nous ajuster dans ce vingt-unième siècle, où la vie n'est plus uniquement structurée autour de nos communautés, de nos villages ou de nos villes.

De nos jours, l'information est considérée comme le cinquième facteur de production qui n'est pas du tout inférieur à la terre, au travail, au capital et à l'entrepreneuriat. Pour corroborer cette opinion, BRANDLIN et HARRISSON (1987) ont observé que « la ressource informationnelle est maintenant un nouveau type de capital décrit comme étant le capital du savoir ». Et DRUCKER (1999) de renchérir que l'acquisition systématique et résolue de l'information plutôt que de la science et de la technologie apparaît comme la nouvelle base de travail, de la productivité et de l'effort à travers le monde. Pour ce faire, les organisations tout comme les entreprises doivent savoir gérer leurs ressources informationnelles c'est-à-dire savoir les planifier, les organiser, les diriger et les contrôler en vue d'atteindre les objectifs préalablement définis [1]. L'informatique, perçue dès ses débuts comme un outil précieux pour faciliter les processus de gestion et ainsi améliorer la productivité, est progressivement objet d'appropriation par plusieurs entreprises dans la gestion des données à leur sein.

En effet, elle a permis de réduire le temps d'exécution des tâches, les effectuer avec le plus de précision et d'exactitude possibles et ainsi réduire le taux d'erreurs dues, dans la plupart des cas, au traitement manuel ou mécanique des données. A cet effet, la conception des logiciels appropriés dans chaque domaine a permis de répondre aux besoins spécifiques en matière de gestion des données. Dans cette perspective, l'intégration des technologies du traitement de l'information par l'ordinateur par exemple est véritablement devenue la clé du succès dans la conception des systèmes d'information intelligents. Les fonctions de stockage des données et

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

d'accès aux diverses bases et banques des données sont rendues faciles grâce à un système automatisé assuré par un système de gestion des bases de données.[2]

Ainsi, le système informatique intervient comme soutien primaire au système d'information dans les tâches d'automatisation, d'analyse, de l'archivage ainsi que la diffusion de cette information au sein de l'entreprise. L'état déplorable aujourd'hui de l'infrastructure de l'information dans certains pays d'Afrique reste connu comme étant le principal goulot d'étranglement de la gestion moderne de l'information. En effet, nombreuses entreprises et organisations telles que les universités et institutions d'enseignements supérieurs, bien qu'ayant intégré l'ordinateur dans leur gestion de données, n'ont pas encore atteint le niveau de concevoir des systèmes automatisés de domaine. Les données, dont parfois le volume est gigantesque, constituent pourtant la matière première de leur activité et nécessitent ainsi une attention et une rigueur particulières dans leur gestion. Au fur et à mesure aujourd'hui que nous retrouvons dans certaines institutions d'enseignements supérieurs et universitaires l'absence d'un outil informatique moderne particulièrement dans le secteur académique, nous sommes partis des problèmes que, au fur et à mesure aujourd'hui on retrouve la croissance des étudiants sollicitant les inscriptions, des enseignants avec plusieurs activités à réaliser dans les universités, la difficulté dans la gestion manuelle dans le secteur académique, cette gamme de question sera vérifiée par nos hypothèses à savoir:

Du côté étudiant, l'inscription en ligne et la notification par la voie électronique en utilisant même les téléphones smart, la production des listes des étudiants inscrits ainsi que les résultats obtenus à la fin de la session ;

Du côté enseignant : la production des listes des enseignants en leur notifiant la charge horaire de tout un chacun, le calcul de note des étudiants et la production des résultats ;

Du côté enseignement : le suivi des différentes affectations de cours aux enseignants et dans les classes par le chef de département, la composition et gestion d'un jury pour une promotion donnée ; la mise en place d'une base de données selon un SGBDR MySQL, la connexion à cette base de données par l'outil de développement Java. L'échange entre client et le serveur par la création d'un site web.

#### **I.4. Solutions proposées**

La mise en place d'une nouvelle application web de gestion universitaire dans son secteur académique et plus particulièrement dans la gestion des étudiants, enseignants et enseignements, la mise en place d'une base de données en ligne qui va permettre l'interconnexion et l'échange direct des informations entre les différents services et aux chercheurs désireux, concevoir une base de données stockant l'enregistrement des nouveaux étudiants en ligne et la mise à jour de nouvelles affectations des cours aux enseignants.

#### **I.5. Résultats attendus**

Les utilisateurs de notre application bénéficieront d'un outil de travail moderne, actualisé qui leur permettra d'avoir une base de données à jour, de conserver les fichiers sur des supports électroniques, diminuer le temps de la recherche de documents, assurer une traçabilité nette des prestations de chaque enseignant et de cursus de chaque étudiant inscrit. Alors, la qualité de service se retrouverait amélioré ainsi que l'efficacité par les agents du secteur académique et d'une institution universitaire dans sa globalité.

Ensuite, la conception d'une base de données selon le SGBDR, la connecter au modèle client-serveur par la technologie MVC. Disponibiliser les listes et les notes des étudiants inscrits au moment voulu et l'évolution de ces derniers dans le cursus académique, l'attribution des cours dans les départements et le suivi des enseignements, fournir les données et renseignements aux services demandeurs et aux désirants de ces services, la formation des utilisateurs pour usage de la nouvelle application. Enfin, démontrer l'efficacité d'un usage accru et intensif de l'outil informatique dans les milieux académiques et universitaires.

#### **I.6. Apports scientifiques et technologiques**

La réalisation de ce projet apporte une application web conçue à base des technologies modernes tels que MVC, Spring-Boot, Hibernnet qui permettent l'automatisation du processus universitaire dans le secteur académique utilisable aussi sur les Smartphones, une application web sécurisée observant les critères de sécurités ci-après : la disponibilité, l'intégrité, la confidentialité, la non répudiation, l'authentification, l'identification, l'autorisation et la traçabilité.

La confidentialité : la communication de données reste privée entre un émetteur et un destinataire et demande que l'information sur le système ne puisse être lue que par les personnes autorisées. Pas de divulgation à des tiers non autorisés par le chiffrement. Pour notre cas nous y sommes arrivés au chiffrement des données à travers MD5 (Message-Digest Algorithm 5 qui

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

est une fonction de hachage cryptographique largement utilisée pour vérifier l'intégrité de données) selon les principes de Spring Sécurité pour crypter et décrypter le mot de passe.

Cette application donne un modèle de sauvegarde, conservation et archivage des prestations des corps enseignants comme moyen de suivi et de monitoring des activités académiques au sein d'une institution universitaire. Nous pensons que ce travail constituera sans doute une référence aux futurs chercheurs. Quant aux institutions qui bénéficieront de cette application informatique de l'adopter et de l'utiliser efficacement.

### **I.7. Domaine d'application**

Domaine universitaire dans la recherche et enseignement ; Systèmes informatiques.

### **I.8. Méthodologie utilisée**

Une méthode définit une démarche reproductible qui produit des résultats fiables. Une méthode d'élaboration de logiciels décrit comment modéliser et construire des systèmes logiciels de manière fiable et reproductible [3]. De manière générale, une méthode définit : des éléments de modélisation, une représentation graphique, du savoir-faire et des règles. Les objectifs d'une méthode sont les suivants : se donner toutes les chances de mener à bien un projet informatique, établir un plan projet réaliste en définissant, estimant et planifiant les moyens à mettre en œuvre, maîtriser le projet en mesurant son avancement et les écarts éventuels avec les engagements pris, s'assurer que la qualité définie est respectée. Pour réaliser un travail scientifique, il est nécessaire d'employer certaines voies et moyens judicieux au profit d'atteindre l'objectif visé. Pour recueillir les informations et données nécessaires ayant servis à la préparation et l'élaboration de ce travail, nous avons fait recours aux méthodes et techniques diverses et diversifiées telles que la méthode Analytique qui consiste à analyser en détails les données récoltées durant la période de recherche, la méthode UML qui est une méthode de modélisation orientée objet. Cette méthode nous a permis de comprendre systématiquement les besoins et attentes des institutions universitaires, la technique d'observation, d'interview et la technique documentaire nous ont aussi servis pour la mise en œuvre de nos recherches.

### **I.9. Outils mathématiques utilisés**

Nous avons fait recours à l'algèbre relationnelle.

Notre projet de recherche est circonscrit en quatre chapitres ci-après :

- Le chapitre premier est consacré à l'introduction générale ;
- Le deuxième chapitre consacré à la conception du système de gestion 3<sup>E</sup> ;
- Le troisième chapitre consacré aux formalismes mathématiques avec algèbre relationnelle ;
- Le quatrième chapitre est consacré à l'implémentation du système « Gestuni »

## **Chapitre II. CONCEPTION, MODELISATION DE LA GESTION UNIVERSITAIRE**

Ce chapitre va s'intéresser en premier lieu sur la conception du système de gestion de 3E (Etudiant, Enseignant et Enseignement) au sein d'une université. Il s'agira question d'aborder les points tel que les différentes entités, associations, opérations ou méthodes et attributs qui seront développés par le nouveau système. Tout en continuant, en second lieu le développement de notre projet, ce chapitre nous aide à comprendre comment la mise en place de la modélisation de notre système d'études avec les modèles UML sera mise en marche tout en intégrant également la notion d'héritage qui sera aussi au rendez-vous.

### **II.1. Généralités sur enseignement, enseignant et étudiants (3e).**

#### **II.1.1. La Gestion des enseignements**

Alors que la mission principale d'un établissement est la formation des futurs cadres par l'enseignement, l'enseignement supérieur et universitaire reste et joue un rôle fondamental dans le développement d'une société libre et démocratique, par l'enseignement, la recherche, la création et les services à la collectivité, les universités contribuent à préparer les personnes qui, à titre de citoyennes, citoyens, travailleuses et travailleurs, façonnent la société et préparent son avenir.

- **Cursus académique**

Un cursus académique désigne généralement un parcours ; un ensemble des études (scolaires, universitaires...) d'un domaine. Il représente également l'ensemble des informations concernant un candidat au cours d'un cycle universitaire se terminant éventuellement par une remise de diplôme.[4]

Pour notre cas, le cursus académique désigne un processus englobant un ensemble d'informations relatives à l'évolution de l'étudiant pendant un cycle, depuis son inscription jusqu'à sa sortie de l'institution, et qui permettent de faire le suivi de son évolution académique au sein de l'institution. Le parcours des étudiants devra donc être un processus qui comprend un ensemble de données enregistrées et conservées par le service chargé de cette matière.

Les informations nécessaires au cursus de l'étudiant se résument en trois étapes principales :

1. Les informations concernant l'inscription de l'étudiant au sein de l'institution ;
2. Les informations concernant son évolution académique au sein de l'institution ;
3. Les informations à l'homologation des diplômes académiques de l'étudiant.

Etant donné que le traitement de toutes ces informations exige une gestion très efficace, nous comptons, pour cette fin, concevoir et formaliser un système d'information et de traitement automatisé, adapté à la configuration actuelle.

### **II.1.2 La gestion des enseignants**

- **Mission des enseignants chercheurs.**

Les enseignants chercheurs, à travers l'enseignement et la recherche, accomplissent une mission de service public d'enseignement supérieur.

A ce titre ils sont tenus de :

- Dispenser un enseignement de qualité actualisé, lié aux évolutions de la science et des connaissances, de la technologie et des méthodes pédagogiques et didactiques, confirmé avec les normes éthiques et professionnelles ;
- Participer à l'élaboration du savoir et assurer la transmission des connaissances au titre de la formation initiale et continue ;
- Mener des activités de recherche-formation pour développer leurs aptitudes et leurs capacités à exercer la fonction d'enseignant chercheur.

### **II.1.3 La gestion des étudiants**

La gestion de l'étudiant c'est l'admission de ce dernier pour des formations en langues étrangères, ainsi que la gestion de son mouvement au sein de l'université. Lors de son entrée, l'étudiant est accueilli au sein de l'université en fonction de son choix et la réussite au concours. L'étudiant est un élément essentiel pour l'existence et le fonctionnement du centre de formation, où il doit avoir un dossier propre à lui. Le but essentiel de la gestion de l'étudiant est le suivi, l'historique et l'intérêt de l'information est de (d') :

- Avoir un dossier numérique pour chaque étudiant ;
- Planifier des inscriptions et les durées de formations (date début et date de fin de formation) ;
- Contrôler les paiements des frais de formations.

- **Le dossier étudiant**

Un dossier étudiant est la mémoire dans laquelle sont consignées toutes informations nécessaires à la prise en charge et la surveillance d'un étudiant. En d'autres termes, un dossier étudiant englobe tout ce qui peut être mémorisé chez un étudiant.

- **Les objectifs de la gestion des étudiants**

La gestion des étudiants a pour objectifs de :

- Savoir traiter les démarches d'accès aux dossiers des étudiants ;
- Améliorer la gestion des informations en les centralisant ;
- Constituer une banque de données ;
- Accéder à l'historique des étudiants précédents, formation prise, ...

## II.2. Les différentes entités utilisées dans le système

Pour bien mener à bon port notre recherche, nous aurons besoin d'utiliser les différentes entités ci-après : Administrateur, RolePersonne, ClasseEtudiant, Classe, Cours, CoursClasse, Departement, DocumentEtudiant, Enseignant, EtudesFaites, Etudiant, Parents, Jury, Note, Personne, CoursEnseignant et Role.

### II.2.1 Objets des diagrammes de classes

Dans cette partie de notre travail, nous allons exprimer les relations qui existent entre les objets mais aussi l'état des objets, tout en essayant de montrer l'état des instances d'objet pendant une interaction, autrement dit c'est une photographie à un instant précis des attributs et objets existants. Et cela sera représenté sous formes de différents tableaux représentant la liste de classes ou entités, la liste des associations, la liste des attributs et la liste des opérations des classes ou des méthodes.

- Liste des classes (entités)

La classe étant une entité importante dans le processus de développement de notre application, il nous revient de donner les détails nécessaires en rapport avec les entités que nous allons développer dans le tableau ci-après:

Tableau 2.1. Liste des entités qui seront développées

Numéro	Nom	Visibilité	Classificateur
1	Administrateur	Public	Classe
2	RolePersonne	Public	Classe
3	ClasseEtudiant	Public	Classe
4	Classe	Public	Classe
5	Cours	Public	Classe
6	CoursClasse	Public	Classe
7	Departement	Public	Classe
8	DocumentsEtudiant	Public	Classe
9	Enseignant	Public	Classe
10	EtudesFaites	Public	Classe
11	Etudiant	Public	Classe
12	Parents	Public	Classe
13	Jury	Public	Classe
14	Note	Public	Classe
15	Personne	Public	Classe
16	CoursEnseignant	Public	Classe
17	Roles	Public	Classe

Tableau 2.1. Liste des entités qui seront développés

Il ressort de ce tableau 2.1, que dans le nouveau système à produire, nous aurons effectivement besoins de ces 17 entités pour la conception et la modélisation de notre système.

## II.2.2. Objets communs

- Liste des associations

Pour la connexion sémantique entre deux classes (relation logique) dans notre modélisation, nous avons fait recours à des associations que nous n'avons utilisées que dans les diagrammes de classes. Nous avons fait recours à la multiplicité qui nous a servi à compter le nombre minimum et maximum d'instances de chaque classe dans la relation liant 2 ou plusieurs classes, à la navigabilité si on pourra accéder d'une classe à l'autre. Si la relation est entre les classes A et B et que seulement B est navigable, alors on pourra accéder à B à partir de A mais pas réciproquement. Par défaut, la navigabilité est dans les deux sens.

Tableau 2.2. Liste des associations

Num	Code	Classe B	Classe A	Rôle A	Rôle B	Multipli cité A	Multipli cité B
1	Association_1	DocumentEtu	Etudiant	Présenter		1..*	1..1
2	Association_2	Etudiant	Parents	Donner		1..*	1..1
3	Association_4	Departement	Classes		Appart enir	1..*	1
4	Association_5	Cours	Classes	Dispenser		1	1..*
5	Association_6	Classes	Etudiant			0..1	0..*
6	Association_7	Classes	Inscription	Inscrire		1	1
7	Association_8	Cours	Enseignant	Enseigner		1..*	0..*
8	Association_11	EtudesFaites	Etudiant	Posséder		1..*	1
9	Association_13	Inscription	Etudiant	S'inscrire		1..*	1
10	Association_15	Roles	Etudiant	Associer		1..*	1..*
11	Association_16	Notes	Enseignant			0..1	0..*
12	Association_17	Notes	Etudiant			0..1	0..*
13	Association_18	Jury	Classes			0..1	0..*
14	Association_19	Jury	Enseignant			0..1	0..*
15	Association_20	Jury	Classes			0..1	0..*
16	Association_21	Roles	Administ	Associer		1..*	1..*
17	Association_22	Roles	Enseignant	Associer	Associ er	1..*	1..*
18	Association_23	Roles	RolesPerso	Associer		1..*	1..*

Il ressort de ce tableau 2.2, que dans le nouveau système à produire, nous aurons effectivement besoins de ces 18 associations qui seront mises en concert pour la conception et la modélisation de notre système.

### II.2.3. Liste des attributs des classes(entités)

Pour être explicite dans notre travail, nous essayons de donner les attributs utilisés dans la suite de notre application, cela signifie que chaque attribut possède sa propre valeur, son propre espace mémoire pour mémoriser sa valeur. Ainsi, on peut tracer le tableau suivant, indiquant l'accès à un attribut, sa classe, type de données et le code :

Tableau 2.3. Liste des attributs des entités

Num	Code	Type de données	Visibilité	Classificateur
1	IdPersonne	Long	Public	Personne
2	Nom	String	Public	Personne
3	Postnom	String	Public	Personne
4	Prenom	String	Public	Personne
5	Sexe	String	Public	Personne
6	DateNaissance	Date	Public	Personne
7	LieuNaissance	String	Public	Personne
8	Login	String	Public	Personne
9	Pasword	String	Public	Personne
10	Promotion	String	Public	Personne
11	Telephone	String	Public	Personne
12	TypePersonne	Int	private	Personne
13	Adresseemail	String	private	Personne
14	IdEtudiant	Long	Public	Etudiant
15	MatriculeEtudiant	String	private	Etudiant
16	IdEnseignant	Long	Public	Enseignant
17	MatriculeEnseignant	String	Public	Enseignant
18	Grade	String	private	Enseignant
19	Domaine	String	private	Enseignant
20	IdAdmini	Long	Public	Administrateur
21	MatriculeAdmin	String	Public	Administrateur
22	IdClasse	Long	Public	Classes
23	Designation	String	Public	Classes
24	Capacite	String	Public	Classes
25	IdCours	Long	Public	Cours
26	Designation	String	Public	Cours
27	Credit	Int	Public	Cours
28	CodeCours	String	Public	Cours
29	IdDepartement	Int	Public	Departement
30	CodeDepartement	String	Public	Departement
31	NombreEnseignant	String	Public	Departement

32	Designation	String	Public	Departement
33	Id	Long	private	CoursClasse
34	Annee_Acad	String	private	CoursClasse
35	Semestre	String	private	CoursClasse
36	Classe_Id	Long	private	CoursClasse
37	Cours_Id	Long	private	CoursClasse
38	Id	Long	private	ClasseEtudiant
39	Annee_Acad	String	private	ClasseEtudiant
40	Classe_Id	Long	private	ClasseEtudiant
41	Etudiant_Id	Long	private	ClasseEtudiant
42	Id	Long	private	Parents
43	NomsPere	String	private	Parents
44	NomsMere	String	private	Parents
45	Province	String	private	Parents
46	Territoire	String	private	Parents
47	Nationalite	String	private	Parents
48	Sexe	String	private	Parents
49	Adresse	String	private	Parents
50	Telephone	String	private	Parents
51	Id	Long	private	EtudesFaites
52	Section	String	private	EtudesFaites
53	Option	String	private	EtudesFaites
54	AnneeDiplome	String	private	EtudesFaites
55	Pourcentage	String	private	EtudesFaites
56	EcoleProvenance	String	private	EtudesFaites
57	Id	Long	private	DocumentsEtudiant
58	AptitudePhysique	String	private	DocumentsEtudiant
59	BonneConduite	String	private	DocumentsEtudiant
60	Bulletin5	String	private	DocumentsEtudiant
61	Bulletin6	String	private	DocumentsEtudiant
62	Diplôme	String	private	DocumentsEtudiant
63	IdentitePiece	String	private	DocumentsEtudiant
64	AttestNaissance	String	private	DocumentsEtudiant
65	AttestNationalite	String	private	DocumentsEtudiant
66	Passeport	String	private	DocumentsEtudiant
67	PreuvePayment	String	private	DocumentsEtudiant
68	Id	Long	private	Notes
69	IdEtudiant	Long	private	Notes
70	IdEnseignant	Long	private	Notes

71	Session1	String	private	Notes
72	Session2	String	private	Notes
73	Pourcentage	Float	private	Notes
74	Mention	String	private	Notes
75	Id	Long	private	Jury
76	Role	String	private	Jury
77	Annee	Date	private	Jury
78	IdClasse	Long	private	Jury
79	IdEnseignant	Long	private	Jury
80	Id	Int	private	Roles
81	Designation	String	private	Roles
82	Id	Int	private	RolePersonne
83	Enseignant_Id	Int	private	RolePersonne
84	Role_Id	Int	private	RolePersonne
85	Id	Long	private	CoursEnseignant
86	Annee_Academique	String	private	CoursEnseignant
87	Cours_Id	Long	private	CoursEnseignant
88	Enseignant_Id	Long	private	CoursEnseignant

Il ressort de ce tableau 2.3, que dans le nouveau système à produire, nous aurons effectivement besoins de ces 88 attributs qui seront en concert pour la conception et la modélisation de notre système en termes d'utilisation des données.

#### II.2.4. Liste des opérations des classes (méthodes) :

Dans la liste des choses qui nous ont servi, nous avons rajouter les opérations de classes communément appelées méthodes. Nous les avons utilisées pour nous fournir des informations complémentaires sur les méthodes, les classes, les variables, etc. Ils nous ont servi régulièrement dans les opérations que vont jouer les entités de notre application.

Tableau 2.4. Liste des opérations ou méthodes

Num	Code	Type de résultat	Visibilité	Classificateur
1	Ajouter	Void	Public	Personne
2	S'authentifier		Public	Personne
3	Envoyer demande d'inscription		Public	Etudiant
4	Envoie formulaire d'inscription		Public	Etudiant
5	Ajouter_		Public	Etudiant
6	Modifier		Public	Etudiant
7	Supprimer		Public	Etudiant
8	Donner_cote		Public	Enseignant
9	Ajouter		Public	Enseignant

10	Modifier		Public	Enseignant
11	Supprimer		Public	Enseignant
12	Creer_compte	Void	Public	Administrateur
13	Modifier_compte	Void	Public	Administrateur
14	Supprimer_compte		Public	Administrateur
15	Ajouter_utilisateur		Public	Administrateur
16	AjouterClasse	Void	Public	Classes
17	ModifierClasse	Void	Public	Classes
18	SupprimerClasse	Void	Public	Classes
19	Ajouter		Public	Cours
20	Modifier		Public	Cours
21	Supprimer		Public	Cours
22	Enregistre	Void	Public	Departement
23	Modifier		Public	Departement
24	Supprimer		Public	Departement
25	Prendre	Void	Public	Inscription
26	Confirmer	Void	Public	Inscription
27	Ajouter		Public	CoursClasse
28	Modifier		Public	CoursClasse
29	Supprimer		Public	CoursClasse
30	Ajouter		Public	ClasseEtudiant
31	Modiefier		Public	ClasseEtudiant
32	Supprimer		Public	ClasseEtudiant
33	Ajouter		Public	IdentiteParent
34	Modifier		Public	IdentiteParent
35	Supprimer		Public	IdentiteParent
36	Ajouter		Public	EtudesFaites
37	Modifier		Public	EtudesFaites
38	Supprimer		Public	EtudesFaites
39	Ajouter		Public	DocumentsEtud
40	Modifier		Public	DocumentsEtud
41	Supprimer		Public	DocumentsEtud
42	Ajouter		Public	Notes
43	Modifier		Public	Notes
44	Supprimer		Public	Notes

Il ressort de ce tableau 2.4, que dans le nouveau système à produire, nous aurons effectivement besoins de ces méthodes énumérées pour la conception et la modélisation de notre système.

### II.2.5. L'Explication des entités utilisées

Comme nous avons énuméré les entités à utiliser dans notre application, nous essayons de développer dans ce point chaque entité en expliquant le pourquoi de son existence et ce qu'elle fait et comment elle nous a aidée dans l'implémentation de notre application.

- **Administrateur** : cette entité comme le nom l'indique deviendra la table administrateur, c'est la table dans laquelle seront enregistrés tous les administrateurs du système et qui auront toute la gestion, le contrôle et la mise à jour de la nouvelle application, elle sera l'une héritée de la super classe personne ;
- **RolePersonne** : elle deviendra plus tard la table RolePersonne dans laquelle tous les rôles attribués à chaque utilisateur de l'application seront stockés et enregistrés.
- **ClasseEtudiant** : elle deviendra plus tard la table ClasseEtudiant dans laquelle seront générés automatiquement les différentes années académiques, les id des classes et étudiants feront aussi partie.
- **Classe** : cette entité comme le nom l'indique deviendra la table classe, c'est la table dans laquelle seront enregistrées toutes les classes ou promotions de la nouvelle application ;
- **Cours** : cette entité deviendra la table cours, et comme son nom l'indique c'est la table dans laquelle seront enregistrés tous les cours de la nouvelle application ;
- **CoursClasse** : elle deviendra plus tard la table CoursClasse dans laquelle seront générés automatiquement les différentes années académiques, les id des classes et des cours plus le semestre feront aussi partie de l'application.
- **Departement** : cette entité comme le nom l'indique deviendra la table departement, c'est la table dans laquelle seront enregistrés tous les départements du système, le code, le nombre d'enseignants ainsi que la désignation de chaque département enregistré dans notre nouvelle application ;
- **DocumentEtudiant** : cette entité deviendra la table DocumentEtudiant, et comme son nom l'indique c'est dans laquelle que seront stockées tous les documents fournis par chaque candidat lors de son inscription ;
- **Enseignant** : cette entité comme le nom l'indique deviendra la table enseignante, c'est la table dans laquelle seront enregistrés tous les enseignants de la nouvelle application, elle sera l'une héritée de la super classe personne ;
- **EtudesFaites** : elle deviendra plus tard la table EtudesFaites dans laquelle seront enregistrées les différentes informations des études suivies par le candidat aux humanités dans la nouvelle application.

- **Etudiant** : cette entité comme le nom l'indique deviendra la table Etudiant, c'est la table dans laquelle seront enregistrés tous les étudiants de la nouvelle application, elle sera l'une héritée de la super classe personne ;
- **Parents** : cette entité deviendra la table Parent, c'est la table dans laquelle seront enregistrés les différentes identités des parents de chaque candidat, leurs Id, noms, adresses, téléphones...dans la nouvelle application ;
- **Jury** : cette entité deviendra la table Jury, c'est la table dans laquelle seront enregistrés les différents Id, rôles noms des membres constituant le jury ainsi que la promotion concernée par le jury dans la nouvelle application ;
- **Note** : cette entité deviendra la table Note, c'est la table dans laquelle seront enregistrés les différentes côtes obtenues par un étudiant ainsi que la session, mention classe et pourcentage dans la nouvelle application ;
- **Personne** : cette entité comme le nom l'indique deviendra la table Personne, c'est la table dans laquelle sera enregistrée toute identité de chaque personne. C'est la super classe dans laquelle toutes les classes filles hériteront dans la nouvelle application ;
- **CoursEnseignant** : elle deviendra plus tard la table CoursEnseignant dans laquelle seront enregistrés les cours ou différentes charges accordées à un enseignant durant l'année académique dans notre nouvelle l'application.
- **Role** : cette entité deviendra la table Role, c'est la table dans laquelle seront enregistrés les rôles accordés aux enseignants (Chefs des Départements) et la scolarité de la nouvelle application ;

### II.3. Héritage [5]

Transmission automatique des propriétés d'une classe de base vers une sous-classe. Il existe différentes sémantiques de la généralisation et de l'héritage, qui sont deux concepts associés. L'héritage est le troisième des paradigmes de la programmation orientée objet (le 1<sup>er</sup> étant l'encapsulation, le 2<sup>e</sup> la structure de classe). L'emploi de l'héritage conduit à un style de programmation par raffinements successifs et permet une programmation incrémentielle effective. Il représente la relation : **EST-UN**

Exemple :

- Un Etudiant est une Personne ;
- Une Enseignant est une Personne ;
- Un ChefDep est un Enseignant.

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

Alors que l'objet membre représente la relation : **A-UN**

Une Personne a un nom

L'héritage est mis en œuvre par la construction de classes dérivées.

Pour notre projet, le graphe de l'héritage est comme suit :

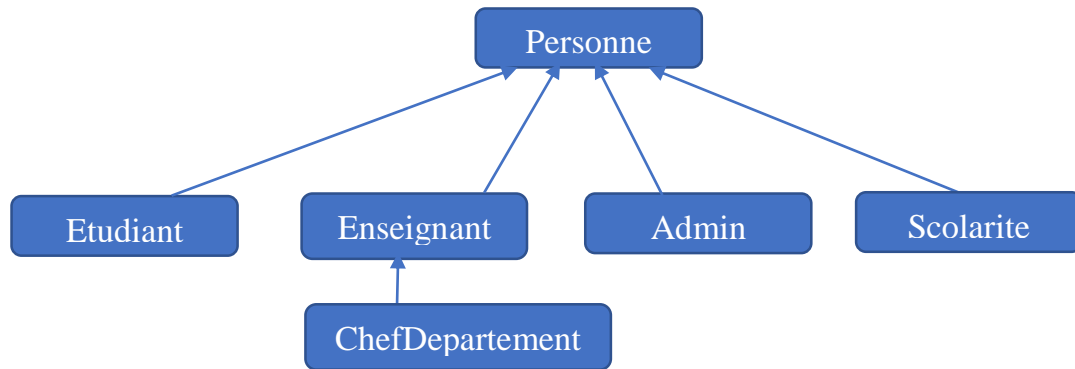


Figure 2.1. La présentation de l'héritage

- La classe dont on dérive est dite Classe de base, Super Classe, Classe mère : **Personne** est la classe de base (classe supérieure) ou super classe ;
- Les classes obtenues par dérivation sont dites Classes dérivées, sous-classes, classes filles : Etudiant, Enseignant, Admin et Scolarite sont des classes dérivées (sous-classes).

La vision descendante : la possibilité de reprendre intégralement tout ce qui a déjà été fait et de pouvoir l'enrichir et la vision ascendante qui est la possibilité de regrouper en un seul endroit ce qui est commun à plusieurs.

Vers le haut : on regroupe dans une classe ce qui est commun à plusieurs classes. Dans la classe Personne, on regroupe les caractéristiques communes aux Etudiants et aux Enseignants. Vers le bas (lors de la réutilisabilité) : la classe de base étant définie, on peut la reprendre intégralement pour construire la classe dérivée. La classe Personne étant définie, on peut la reprendre intégralement, pour construire la classe Enseignant.

Une classe dérivée modélise un cas particulier de la classe de base, et est enrichie d'informations supplémentaires.

La classe dérivée possède les propriétés suivantes :

- Contient les données membres de la classe de base ; Peut en posséder de nouvelles ;
- Possède (à priori) les méthodes de sa classe de base.

## II.4. Utilisation de l'UML

UML (Unified Modeling Language en anglais) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. Il permet de faciliter les transitions lors de développement d'un projet, du besoin originel à la phase d'implémentation. [6]

### II.4.1 Objectifs d'UML

Le langage UML est une synthèse de tous les concepts et formalismes méthodologiques les plus utilisés, pouvant être utilisé, grâce à sa simplicité et à son universalité, comme langage de modélisation pour la plupart des systèmes devant être développés. Le langage UML permet ainsi d'apporter des solutions lors du développement de systèmes informatisés :

- Décomposer le processus de développement en distinguant la phase d'analyse de la phase de réalisation ;
- Décomposer le système en sous-systèmes plus facilement abordables : réduction de la complexité, répartition du travail, réutilisation des sous-systèmes ;
- Utiliser une technologie de haut niveau proche de la réalité pour aborder le développement ;
- Indépendant des langages de programmation.

Alors, est un langage : pour visualiser chaque symbole graphique a une sémantique, spécifier de manière précise et complète sans ambiguïté, construire (les classes, relations...), documenter (les différents diagrammes, notes, contraintes, exigences seront présentés dans un document).

### II.4.2 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Pour notre travail, ci-après nous avons identifié les acteurs qui interagissent avec le système :

- **Administrateur** : il organise et gère en toute fiabilité les systèmes de gestion des données, il en assure la cohérence, la qualité et la sécurité. Son rôle est d'installer, de configurer et de gérer cet outil stratégique, performant et facile à utiliser. Il veille à ce que les informations soient disponibles à tout moment. Ensuite, il fait évoluer la base de données en lui apportant des modifications.

- Il crée les comptes ;
  - Il gère le droit d'accès ;
  - Il modifie les comptes ;
  - Il supprime le compte ;
  - Il ajoute des nouveaux utilisateurs.
- **Etudiant** : c'est en premier lieu un candidat à l'inscription, il devient étudiant lorsqu'il est admis.
    - Envoyer la demande d'inscription (remplir le formulaire d'inscription);
    - Recevoir la notification par E-mail ;
    - Consulter les listes d'inscriptions ;
    - Consulter ses notes.
- **Scolarité** : il est le responsable du bureau de l'inscription et de scolarité
    - Recevoir les demandes d'inscriptions ;
    - Contrôler les dossiers des candidats ;
    - Envoyer les notifications aux candidats ;
    - Confirmer inscription;
    - Inscrire les nouveaux candidats;
    - Constituer les listes.
- **Chef de département** : il est le responsable du département.
    - Affecter cours dans une classe ;
    - Affecter cours à un enseignant ;
    - Gérer les enseignants ;
    - Donner la charge horaire ;
    - Programmer les enseignants soit affecter les enseignants dans différentes classes.
- **Enseignant** : il est enseignant toute personne engagée ou pas dans l'institution et qui donne cours ou ayant une charge horaire au sein du département.
    - Demander la liste des étudiants inscrits ;
    - Recevoir la notification émanant du Chef de Département qui l'attribuant sa charge ;
    - Participer à une délibération s'il est membre d'un jury ;
    - Donner et envoyer les notes.

## **II.5. Identification des messages**

En informatique un message est un ensemble de données destinées à être transmises par un système informatique, une information donnée par le système d'exploitation d'un ordinateur à un opérateur pour signaler une erreur, l'avertir etc. [7]

Un message représente la spécification d'une communication unidirectionnelle entre objets et qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur du message. Un message est normalement associé à deux occurrences d'évènement : un évènement d'envoi et un autre de réception. Les messages répertoriés entre le système et ses acteurs sont :

### **1. Messages émis par le système**

- Affichage de la liste des étudiants inscrits (retenus) ;
- Confirmation de la réussite de la demande d'inscription (notification) ;
- Notifications : message de confirmation d'inscription, les notes obtenues par un étudiant, l'attribution des charges horaires aux enseignants.

### **2. Messages reçus par le système**

- Les informations de demandes d'inscription ;
- L'ajout et la modification des éléments d'un candidat et
- La demande de visualisation de la liste des étudiants retenus.

## **II.6. Identification de cas d'utilisation [8]**

Les diagrammes de cas d'utilisation (DCU) sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases). Il représente un ensemble de séquence d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera. L'objectif est le suivant : l'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Nommez les cas d'utilisation par un verbe à l'infinitif suivi d'un complément, du point de vue de l'acteur (et non pas du point de vue du système).

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

#### **a) Les relations**

Trois types de relations sont prises en charge par la norme UML et sont graphiquement représentées par des types particuliers de ces relations. Il s'agit de l'inclusion, l'extension et la généralisation.

#### **b) Comment les représenter ?**

Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisation reliés par des associations à leurs acteurs par représentation graphique équivalente. Chaque association signifie simplement « participe à ». Un cas d'utilisation doit être relié à au moins un acteur.



## II.7. Diagramme de séquence

Le diagramme de séquence est un diagramme d'interaction entre les objets, qui met l'accent sur le classement des messages par ordre chronologique durant l'exécution du système. Un diagramme de séquence est un tableau dans lequel les objets sont rangés sur les axes des abscisses et les messages par ordre d'apparition sur les axes des ordonnés.[9]

Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés. Pour l'étude de ce projet, nous vous montrerons les diagrammes de séquence de quelques cas qui sont : authentifier, inscription, consulter le site.

### II.7.1. Diagramme de séquence de cas « s'authentifier »

Ce présent diagramme décrit l'authentification de chaque acteur dans le système.

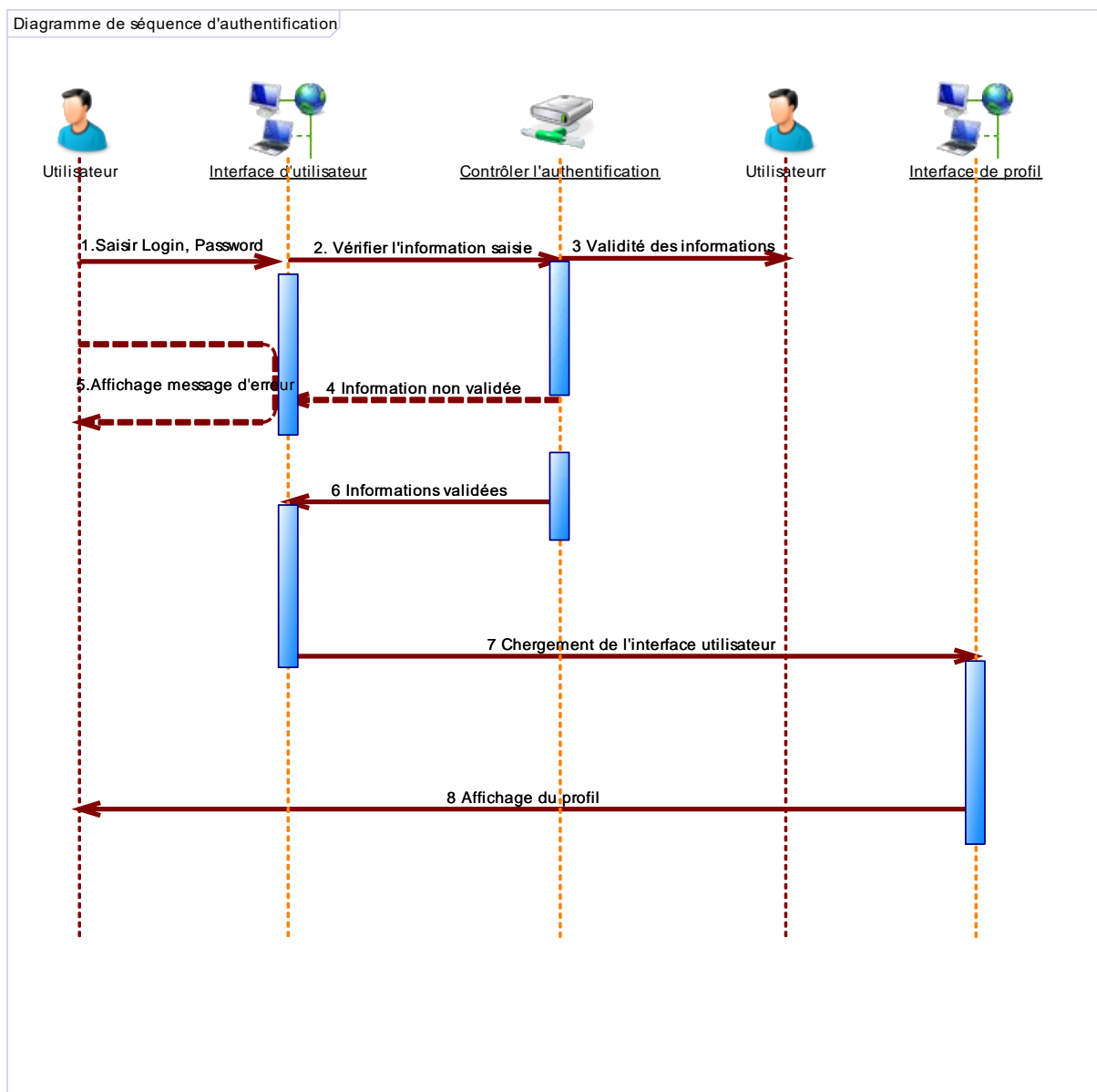


Figure 2.3. Diagramme de séquence de cas « s'authentifier »

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

## II.7.2. Diagramme de séquence de cas « consulter site »

Ce présent diagramme explique les différents scénarios pour qu'un candidat puisse consulte le site de l'institution. Le diagramme se présent de la manière dont voici sa structure :

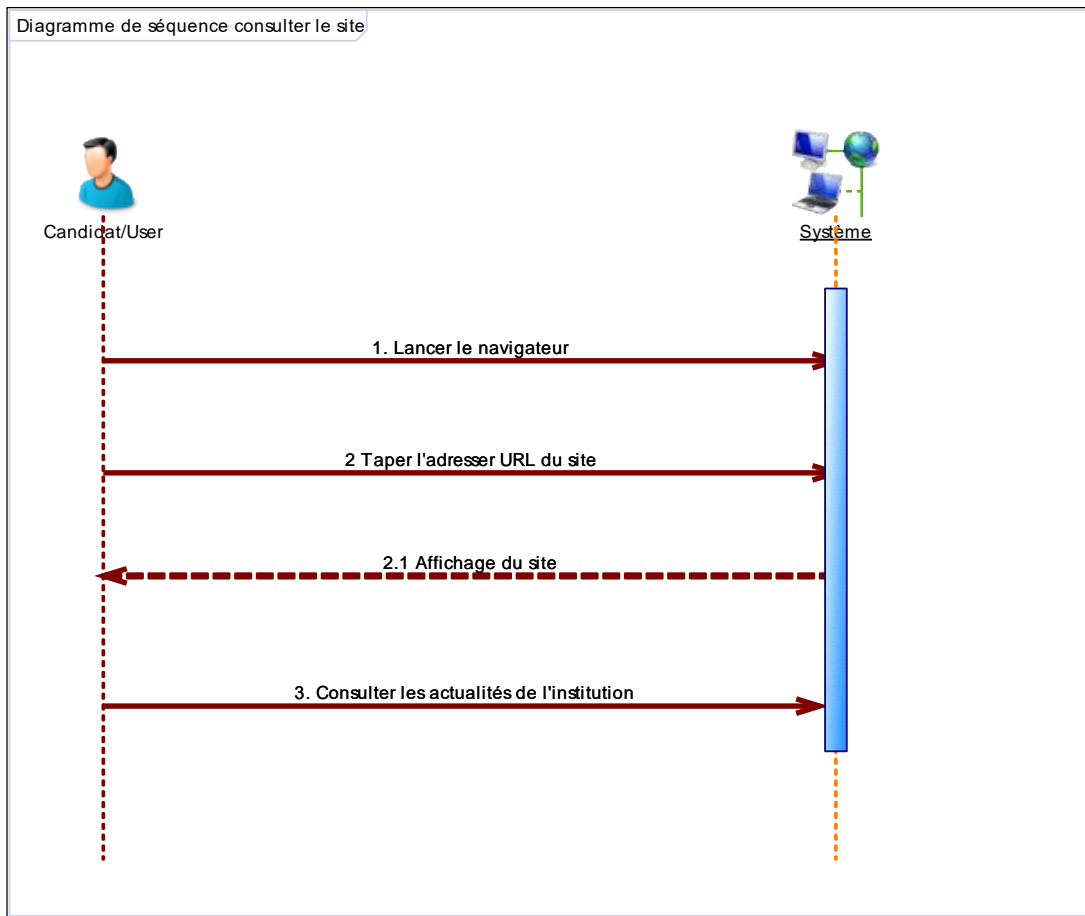


Figure 2.4. Illustration du diagramme de séquence consulter site

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

### II.7.3. Diagramme de séquence de cas « inscription »

Le diagramme de séquence suivant décrit le processus d'inscription d'un étudiant et son enregistrement dans la base de données :



Figure 2.5. Illustration du diagramme de séquence de cas « inscription »

## II.8. Diagramme d'activité

Un diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. Les actions sont les unités de base du comportement du système. Un diagramme d'activités permet de grouper et de dissocier des actions. Si une action peut être divisée en plusieurs actions en séquence, vous pouvez créer une activité les représentant.[10]

Une action représente une seule étape à l'intérieur d'une activité, c'est-à-dire une étape qui ne peut plus être décomposée. Une activité représente un comportement constitué d'éléments qui sont des actions. Une action est un nœud d'activité exécutable, qui est l'unité fondamentale d'une fonctionnalité exécutable dans une activité, contrairement au contrôle et au flot de données entre les actions. L'exécution d'une action représente une transformation ou un traitement dans le système modélisé, que ce soit un ordinateur ou autre chose. Le diagramme d'activité du cas authentifier est représenté dans la figure suivante :

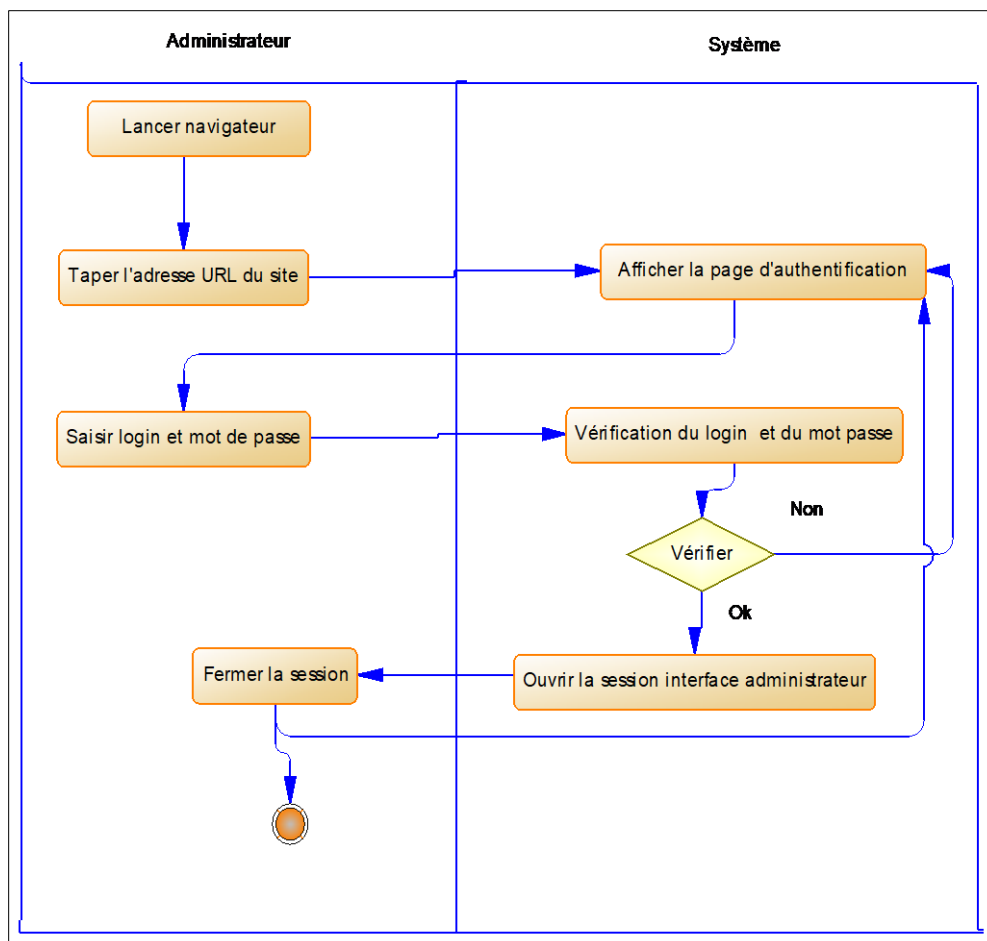


Figure 2.6. Diagramme d'activité globale

## II.9. LES DIAGRAMMES DE CLASSES

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.[11]

Le diagramme de classes est l'un des types les plus populaires en langage UML. Très utilisé par les ingénieurs logiciels pour documenter l'architecture des logiciels, les diagrammes de classes sont un type de diagramme de structure, car ils décrivent ce qui doit être présent dans le système modélisé. Quel que soit votre degré de familiarité avec les diagrammes UML ou les diagrammes de classes, UML est conçu pour être simple à utiliser.

Les divers éléments d'un diagramme de classes peuvent représenter les classes qui seront effectivement programmées, les principaux objets ou les interactions entre classes et objets.

### a) Avantages de classes

Les diagrammes de classes présentent de nombreux avantages pour n'importe quel type d'organisation. Il s'agit de(d') :

- Illustrer des modèles de données pour des systèmes d'information, quel que soit leur degré de complexité ;
- Mieux comprendre l'aperçu général des schémas d'une application ;
- Exprimer visuellement les besoins d'un système et diffuser cette information dans toute l'entreprise ;
- Créer des schémas détaillés qui mettent l'accent sur le code spécifique qui doit être programmé et mis en œuvre dans la structure décrite ;
- Fournir une description indépendante de l'implémentation des types utilisés dans un système, qui sont ensuite transmis entre ses composants.

## b) Le dictionnaire de données

Pour la bonne collection de métadonnées ou de données de référence nécessaire à la conception de notre base de données relationnelle nous avons fait recours à un dictionnaire de données ou référentielle d'une base de données qui revêt pour nous une importance stratégique particulière, car il est le vocabulaire commun de l'organisation des données dans notre future base de données. Nous avons décrit des données aussi importantes que les personnes, enseignants, étudiants, départements, classes, etc. C'est donc le référentiel principal de l'institution qui adoptera notre application, sur lequel s'appuieront les décisions de celle-ci. Il est souvent représenté par un tableau à quatre colonnes contenant le nom, le code et le type de donnée ainsi que des commentaires. Nous le présentons de la manière suivante :

Tableau 2.5. La présentation du dictionnaire de données

Noms	Codes	Type de données	Tables
IdPersonne	IdPersonne	Long	Personne
Nom	Nom	String [50]	Personne
Postnom	Postnom	String [50]	Personne
Prenom	Prenom	String [50]	Personne
Sexe	Sexe	String [10]	Personne
DateNaissance	DateNaissance	Date	Personne
LieuNaissance	LieuNaissance	String [50]	Personne
Login	Login	String [50]	Personne
Passeword	Pasword	String [200]	Personne
Promotion	Promotion	String [50]	Personne
Telephone	Telephone	String [50]	Personne
TypePersonne	TypePersonne	Int	Personne
adresseEmail	Adressemail	String [100]	Personne
IdEtudiant	IdEtudiant	Long	Etudiant
MatriculeEtudiant	Matricule	String [50]	Etudiant
IdEnseignant	IdEnseignant	Long	Enseignant
MatriculeEnseignant	Matricule	String [50]	Enseignant
GradeEnseignant	Grade	String [70]	Enseignant
DomaineEtudes	Domaine	String [100]	Enseignant
IdAdministrateur	IdAdmini	Long	Administrateur
MatriculeAdministrateur	Matricule	String [50]	Administrateur
IdClasse	IdClasse	Long	Classes
DesignationClasses	Designation	String [100]	Classes
CapaciteClasse	Capacite	String [50]	Classes
IdCours	IdCours	Long	Cours

DesignationCours	Designation	String [100]	Cours
Credit	Credit	Int	Cours
CodeCours	CodeCours	String [50]	Cours
IdDepartement	IdDepartement	Int	Departement
CodeDepartement	CodeDepartement	String [50]	Departement
NombreEnseignant	NombreEnseignant	String [50]	Departement
DesignationDepartement	Designation	String [100]	Departement
IdCoursClasse	Id	Long	CoursClasse
Annee_Academique	Annee_Acad	String [50]	CoursClasse
Semestre	Semestre	String [50]	CoursClasse
Classe_Id	Classe_Id	Long	CoursClasse
Cours_Id	Cours_Id	Long	CoursClasse
IdClasseEtudiant	Id	Long	ClasseEtudiant
Annee_Academique	Annee_Acad	String [50]	ClasseEtudiant
Classe_Id	Classe_Id	Long	ClasseEtudiant
Etudiant_Id	Etudiant_Id	Long	ClasseEtudiant
IdParents	Id	Long	Parents
NomsPere	NomsPere	String [100]	Parents
NomsMere	NomsMere	String [100]	Parents
ProvinceParent	Province	String [100]	Parents
TerritoireParent	Territoire	String [100]	Parents
NationaliteParent	Nationalite	String [100]	Parents
SexeParent	Sexe	String [10]	Parents
AdresseParent	Adresse	String [50]	Parents
TelephoneParent	Telephone	String [50]	Parents
IdEtudesFaites	Id	Long	EtudesFaites
Section	Section	String [50]	EtudesFaites
Option	Option	String [50]	EtudesFaites
AnneeDiplome	AnneeDiplome	String [50]	EtudesFaites
Pourcentage	Pourcentage	String [50]	EtudesFaites
EcoleProvenance	EcoleProvenance	String [50]	EtudesFaites
IdDocumentsEtudiant	Id	Long	DocumentsEtudiant
AptitudePhysique	AptitudePhysique	String [50]	DocumentsEtudiant
BonneConduite	BonneConduite	String [50]	DocumentsEtudiant
Bulletin5	Bulletin5	String [50]	DocumentsEtudiant
Bulletin6	Bulletin6	String [50]	DocumentsEtudiant
Diplôme	Diplôme	String [50]	DocumentsEtudiant
IdentitePiece	IdentitePiece	String [50]	DocumentsEtudiant
AttestNaissance	AttestNaissance	String [50]	DocumentsEtudiant

AttestNationalite	AttestNationalite	String [50]	DocumentsEtudiant
Passeport	Passeport	String [50]	DocumentsEtudiant
IdNotes	Id	Long	Notes
IdEtudiant	IdEtudiant	Long	Notes
IdEnseignant	IdEnseignant	Long	Notes
Session1	Session1	String [50]	Notes
Session2	Session2	String [50]	Notes
Pourcentage	Pourcentage	Float	Notes
Mention	Mention	String [100]	Notes
IdJury	Id	Long	Jury
Role	Role	String [50]	Jury
Annee	Annee	Date	Jury
IdClasse	IdClasse	Long	Jury
IdEnseignant	IdEnseignant	Long	Jury
IdRoles	Id	Int	Roles
DesignationRole	ChefDepartement	String [100]	Roles
IdRolePersonne	Id	Int	RolePersonne
Enseignant_Id	Enseignant_Id	Int	RolePersonne
Role_Id	Role_Id	Int	RolePersonne
IdCoursEnseignant	Id	Long	CoursEnseignant
Annee_Academique	Annee_Academique	String [50]	CoursEnseignant
Cours_Id	Cours_Id	Long	CoursEnseignant
Enseignant_Id	Enseignant_Id	Long	CoursEnseignant

Le tableau 2.5 nous présente un dictionnaire des données, ou Référentiel (base de données), qui, est un « référentiel centralisé de l'information sur les données, leurs signification, relation avec d'autres données, leur origines, utilisation et format ». Nous pouvons encore dire que c'est une collection des tables avec des métadonnées. Il décrit la base de données ou une collection des bases de données, un composant intégrale de SGBD nécessaire pour définir sa structure.



## **Conclusion**

Durant ce chapitre nous avons pu définir les concepts liés à la conception et la modélisation du système de gestion 3E (Etudiant, Enseignant et Enseignement) au sein d'une université. Il était question de bien aborder les points tels que les différentes entités, associations, attributs et opérations ou méthodes qui seront développés par le nouveau système. Nous avons aussi trouvé dans ce chapitre que l'université est le lieu par excellence de la formation des futurs cadres par l'enseignement.

Ainsi, ce chapitre nous a permis de présenter les différents diagrammes utilisés dans la modélisation et la conception de notre système en l'occurrence, le diagramme de cas d'utilisation, de séquence, d'activités et le diagramme de classes précédé par un dictionnaire de donnée. Nous venons de voir comment l'UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente.

### **Chapitre III. FORMALISMES MATHÉMATIQUES : Avec Algèbre Relationnelle**

Ce chapitre va nous lancer dans l'utilisation de l'algèbre relationnelle, un support mathématique cohérent sur lequel repose le modèle relationnel. Également, dans ce chapitre il sera question d'aborder les trois familles d'opérateurs relationnels entre autres les opérateurs unaires (Sélection, Projection), les opérateurs binaires ensemblistes (Union, Intersection Différence) et les opérateurs binaires ou n-aires (Produit cartésien, Jointure, Division). L'approche suivie dans ce chapitre sera donc plus opérationnelle que mathématique. Par conséquent, des grandes lignes sont consacrées à l'épineux modèle mathématique en utilisant l'algèbre relationnel.

#### **III.1. Contextes**

Comment réussir à appréhender le comportement des systèmes technologiques de plus en plus complexes afin de les concevoir, de le réaliser et/ou de les communiquer à partir d'un cahier des charges ? Celui-ci est en général défini par différents intervenants, intéressés par les aspects fonctionnels du produit, les besoins du consommateur, les contraintes de coût, le marketing, etc... Du fait de la complexité de plus en plus forte des systèmes technologiques, il apparaît de plus en plus nécessaire de disposer de méthodes et d'outils de conception, de réalisation et/ou de commande qui soient particulièrement efficaces. Au centre de ces méthodes et de ces outils, se trouve en général la modélisation des processus [12]. La mise en place d'un système de gestion étudiant, enseignant, enseignement demande la modélisation du problème afin de pouvoir le simuler, l'analyser et le vérifier. Ces vérifications peuvent mettre en évidence des dysfonctionnements non visibles. Il existe plusieurs langages de modélisation et les outils qui les implémentent. Ces outils, selon les modèles qu'ils représentent, permettent de simuler et de vérifier certaines propriétés. Ceci permet de suivre le fonctionnement du système pour détecter les faiblesses. Pour cela ils peuvent vérifier entre autres :

- L'activité : Vérifie si le fonctionnement d'une partie ou de tout le système évolue ;
- La répétitivité : Vérifie s'il y a des séquences qui se répètent dans le fonctionnement du système ;
- La bornitude : Caractérise la possibilité qu'une partie ou tout le système n'évolue plus, une fois qu'un état spécifique est atteint ;
- La vivacité : Vérifie qu'un état du système puisse être atteignable, quel que soit l'état dans lequel il se trouve ;
- La concurrence : Vérifie si le passage à un état entraîne la collaboration de deux ou plusieurs parties du système.

## III.2 L'objectif de l'utilisation de l'algèbre relationnelle

L'algèbre relationnelle est un support mathématique cohérent sur lequel repose le modèle relationnel. L'objet de cette section est d'aborder l'algèbre relationnelle dans le but de décrire les opérations qu'il est possible d'appliquer sur des relations pour produire de nouvelles relations. L'approche suivie est donc plus opérationnelle que mathématique.

On appelle algèbre relationnelle un ensemble d'opérations simples sur des tables relationnelles, à partir desquelles des opérations plus complexes sont définies par composition. Ils définissent donc un petit langage de manipulation de données. L'élément fondamental du modèle de base de données relationnelle est la table relationnelle. Une table relationnelle est une représentation en extension d'une relation définie sur un certain domaine. Cette méthode consiste essentiellement à créer ou à recréer de nouvelles relations à partir de relations existantes. On peut distinguer trois familles d'opérateurs relationnels :

- Les opérateurs unaires (Sélection, Projection) : ce sont les opérateurs les plus simples, ils permettent de produire une nouvelle table à partir d'une autre table.
- Les opérateurs binaires ensemblistes (Union, Intersection Différence) : ces opérateurs permettent de produire une nouvelle relation à partir de deux relations de même degré et de même domaine.
- Les opérateurs binaires ou n-aires (Produit cartésien, Jointure, Division) : ils permettent de produire une nouvelle table à partir de deux ou plusieurs autres tables.

### III.2.1. Les opérateurs unaires

#### a) Sélection

La sélection parfois appelée restriction génère une relation regroupant exclusivement toutes les occurrences de la relation  $R$  qui satisfont l'expression logique  $E$ , on la note  $\sigma(E)R$ . Il s'agit d'une opération unaire essentielle dont la signature est : relation  $\times$  expression logique  $\rightarrow$  relation. En d'autres termes, la sélection permet de choisir (*i.e.* sélectionner) des lignes dans le tableau. Le résultat de la sélection est donc une nouvelle relation qui a les mêmes attributs que  $R$ . Si  $R$  est vide (*i.e.* ne contient aucune occurrence), la relation qui résulte de la sélection est vide.

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

Exemple : Soit la relation suivante :

- **Personne**(id, matricula, nom, postnom, prenok2m, sexe, lieuNaissance, dateNaissance, adresse, telephone, adresseemail, password, etat)

type_personne	id	adresse	adresseemail	date_naissance	etat	lieu_naissance	matricule	nom	password	postnom	prenom	sexe	telephone	domaine
ADMIN	1	Bukavu	kiseki@gmail.com	2020-03-08 21:45:41	1	Bukavu	SDF4SSGS745845	Kiseki	9f112bc52a09fc35e9320089d7e37480	Asumani	Jean	Homme	+2438785415	
ENSEIGNANT	2	bAGRE	mugaruka@gmail.com	2019-12-30 08:00:00	1	Bukavu	1234567	Mugarka	25d55ad283aa400af464c76d713c07ad	Buduge	Gulain	M	1134648489489	MEDECINE
ETUDIANT	3	bujavu	jo@gmail.com	2019-12-30 08:00:00	1	Bukavu	Azehhdjeufuru	Jona	25d55ad283aa400af464c76d713c07ad	kab	all	Homme	097454856	
ENSEIGNANT	4	KATANA CENTRE	eloge@gmail.com	2019-12-30 08:00:00	1	KITUTU	002/2019-2020	WASSO	25d55ad283aa400af464c76d713c07ad	LUKEKA	ELODIE	M	545454545454	Medecine Interne
ENSEIGNANT	5	KATANA CENTRE	bingwa@gmail.com	2019-12-30 08:00:00	1	KABARE	002/2019-2020	NTUMWA	25d55ad283aa400af464c76d713c07ad	MUKUNDA	CLEMENT	M	45245648	GENIE INFORMATI
ETUDIANT	6	BAGIRA	jonathankabuka@gmail.com	1997-12-29 08:00:00	0	Bukavu	EKJIED41457744	Jonathan	25d55ad283aa400af464c76d713c07ad	Kabuka	Gulain	Homme	09999999	
ETUDIANT	7	BUKAVU	wakisdome@gmail.com	1983-01-03 08:00:00	0	Bukavu	1234567	KISEKI	25d55ad283aa400af464c76d713c07ad	KILONGO	Jean-Paul	Homme	0963258	

Figure 3.1. La structure de la table Personne

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

Soit deux opérations suivantes :

### Afficher les personnes où type\_personne=ENSEIGNANT

- Requête en algèbre relationnelle :  $\sigma_{(type\_personne=ENSEIGNANT)} Personne$
- Requête en SQL : `SELECT * FROM personne WHERE type_personne = 'ENSEIGNANT'`
- Résultat :

type_personne	id	adresse	adressesmail	date_naissance	etat	lieu_naissance	matricule	nom	password	postnom	prenom	sexe	telephone	domaine
ENSEIGNANT	2	BAGRE	mugaruka@gmail.com	2019-12-30 08:00:00	1	Bukavu	1234567	Mugarka	25d55ad283aa400af464c76d713c07ad	Buduge	Gulain	M	1134648489489	MEDECINE
ENSEIGNANT	4	KATANA CENTRE	eloge@gmail.com	2019-12-30 08:00:00	1	KITUTU	002/2019-2020	WASSO	25d55ad283aa400af464c76d713c07ad	LUKEKA	ELODIE	M	545454545454	Medecine Interne
ENSEIGNANT	5	KATANA	bingwa@gmail.com	2019-12-30 08:00:00	1	KABARE	002/2019-2020	NTUMWA	25d55ad283aa400af464c76d713c07ad	MUKUNDA	CLEMENT	M	45245646	GENIE

Figure 3.2. Affiche dans Personne où type\_personne=ENSEIGNANT

### Afficher les personnes où sexe =M

- Requête en algèbre relationnelle :  $\sigma_{(sexe=M)} Personne$
- Requête en SQL : `SELECT * FROM Personne WHERE sexe='M'`

id	adresse	adressesmail	date_naissance	etat	lieu_naissance	matricule	nom	password	postnom	prenom	sexe	telephone	domaine
2	BAGRE	mugaruka@gmail.com	2019-12-30 08:00:00	1	Bukavu	1234567	Mugarka	25d55ad283aa400af464c76d713c07ad	Buduge	Gulain	M	1134648489489	MEDECINE
4	KATANA CENTRE	eloge@gmail.com	2019-12-30 08:00:00	1	KITUTU	002/2019-2020	WASSO	25d55ad283aa400af464c76d713c07ad	LUKEKA	ELODIE	M	545454545454	Medecine Interne
5	KATANA	bingwa@gmail.com	2019-12-30 08:00:00	1	KABARE	002/2019-2020	NTUMWA	25d55ad283aa400af464c76d713c07ad	MUKUNDA	CLEMENT	M	45245646	GENIE

Figure 3.3. Affiche des personnes où sexe=M

## b) Projection

Une projection est une opération qui consiste à supprimer les attributs autres que  $A_1, \dots, A_n$  d'une relation et à éliminer les n-uplets en double apparaissant dans la nouvelle relation ; on la note  $\Pi(A_1, \dots, A_n)R$ . Il s'agit d'une opération unaire essentielle dont la signature est: relation  $\times$  liste d'attributs  $\rightarrow$  relation

En d'autres termes, la projection permet de choisir des colonnes dans le tableau. Si R est vide, la relation qui résulte de la projection est vide, mais pas forcément équivalente (elle contient généralement moins d'attributs).

### Afficher les personnes où type\_personne=CHIEFDEPARTEMENT

- Requête en algèbre relationnelle :  $\Pi_{(nom, postnom, prenom)}^{Personne}$
- Requête en SQL : `SELECT DISTINCT personne. nom, postnom, prenom FROM personne`

```
SELECT DISTINCT personne. nom, postnom, prenom FROM personne
```

Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette tabl

+ Options

	nom	postnom	prenom
<input type="checkbox"/> Éditer Copier Supprimer	Kiseki	Asumani	Jean
<input type="checkbox"/> Éditer Copier Supprimer	ELOGE	LUKEJOS	BINGWA
<input type="checkbox"/> Éditer Copier Supprimer	JEANNE	BITAMWILO	ELODIE
<input type="checkbox"/> Éditer Copier Supprimer	MARIE	ROSE	WASSO
<input type="checkbox"/> Éditer Copier Supprimer	KISEKI	WILONDJA	JEAN PAUL
<input type="checkbox"/> Éditer Copier Supprimer	WASSO	FAIDA	MARIE ROSE
<input type="checkbox"/> Éditer Copier Supprimer	WASSO	BITAMWILO	JEANNE
<input type="checkbox"/> Éditer Copier Supprimer	WASSO	LUKEKA	ELOGE
<input type="checkbox"/> Éditer Copier Supprimer	WASSO	KALESHI	WENGA
<input type="checkbox"/> Éditer Copier Supprimer	BINGWA	SOLE	KINOISerie
<input type="checkbox"/> Éditer Copier Supprimer	DOMINIQUE	WASSO	BINGWA

Figure 3.4. La projection

### III.2.2. Les opérateurs binaires ensemblistes

#### a) Union

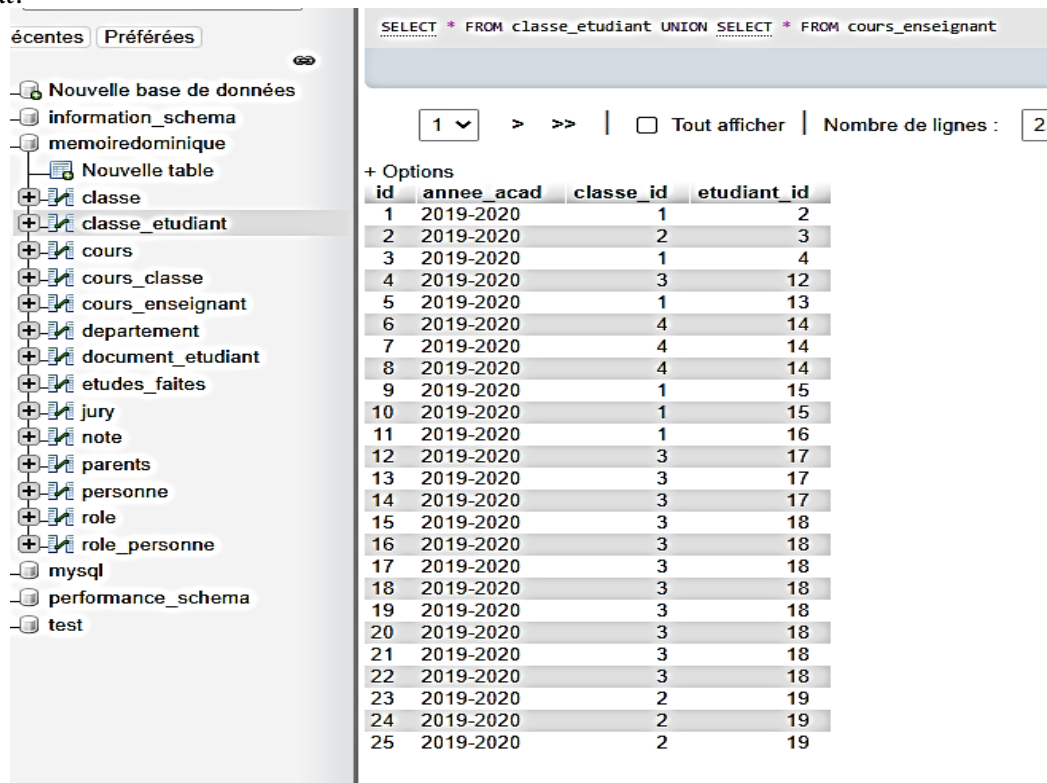
**Définition :** L'union est une opération portant sur deux relations  $R1$  et  $R2$  ayant le même schéma et construisant une troisième relation constituée des n-uplets appartenant à chacune des deux relations  $R1$  et  $R2$  sans doublon, on la note  $R1 \cup R2$ .

Il s'agit une opération binaire ensembliste commutative essentielle dont la signature est :  
relation  $\times$  relation  $\rightarrow$  relation

Comme nous l'avons déjà dit,  $R1$  et  $R2$  doivent avoir les mêmes attributs et si une même occurrence existe dans  $R1$  et  $R2$ , elle n'apparaît qu'une seule fois dans le résultat de l'union. Le résultat de l'union est une nouvelle relation qui a les mêmes attributs que  $R1$  et  $R2$ . Si  $R1$  et  $R2$  sont vides, la relation qui résulte de l'union est vide. Si  $R1$  (respectivement  $R2$ ) est vide, la relation qui résulte de l'union est identique à  $R2$  (respectivement  $R1$ ).

#### Afficher l'union des tables `classe_etudiant` et `cours_enseignant`.

- Requête en algèbre relationnelle : `TABLE [R1] UNION SELECT * FROM R2;`
- Requête en SQL : `SELECT * FROM classe_etudiant UNION SELECT * FROM cours_enseignant`
- Résultat:



The screenshot displays a database management interface. On the left, a tree view shows the database structure, with 'classe\_etudiant' and 'cours\_enseignant' highlighted. On the right, a query editor shows the SQL statement: `SELECT * FROM classe_etudiant UNION SELECT * FROM cours_enseignant`. Below the query, the result set is displayed as a table with 25 rows and 4 columns: `id`, `annee_acad`, `classe_id`, and `etudiant_id`. The results are as follows:

id	annee_acad	classe_id	etudiant_id
1	2019-2020	1	2
2	2019-2020	2	3
3	2019-2020	1	4
4	2019-2020	3	12
5	2019-2020	1	13
6	2019-2020	4	14
7	2019-2020	4	14
8	2019-2020	4	14
9	2019-2020	1	15
10	2019-2020	1	15
11	2019-2020	1	16
12	2019-2020	3	17
13	2019-2020	3	17
14	2019-2020	3	17
15	2019-2020	3	18
16	2019-2020	3	18
17	2019-2020	3	18
18	2019-2020	3	18
19	2019-2020	3	18
20	2019-2020	3	18
21	2019-2020	3	18
22	2019-2020	3	18
23	2019-2020	2	19
24	2019-2020	2	19
25	2019-2020	2	19

Figure 3.5. L'union des tables `classe_et` et `cours_enseignant`

## b) Intersection

**Définition :** L'intersection est une opération portant sur deux relations  $R1$  et  $R2$  ayant le même schéma et construisant une troisième relation dont les n-uplets sont constitués de ceux appartenant aux deux relations, on la note  $R1 \cap R2$ . Il s'agit d'une opération binaire ensembliste commutative dont la signature est : relation  $\times$  relation  $\rightarrow$  relation

Comme nous l'avons déjà dit,  $R1$  et  $R2$  doivent avoir les mêmes attributs. Le résultat de l'intersection est une nouvelle relation qui a les mêmes attributs que  $R1$  et  $R2$ . Si  $R1$  ou  $R2$  ou les deux sont vides, la relation qui résulte de l'intersection est vide.

**Afficher l'intersection des tables : cours\_enseignant et cours\_classe.**

- Requête en algèbre relationnelle :  $R = R1 \cap R2$
- Requête en SQL : 

```
SELECT id, annee_acad, cours_id FROM cours_classe
INTERSECT
SELECT id, annee_acad, cours_id FROM cours_enseignant
```

Ou soit:

- ```
SELECT cours_classe.id, cours_classe.annee_acad, cours_classe.cours_id
FROM cours_classe
INNER JOIN cours_enseignant
ON cours_classe.id= cours_enseignant.id
```

Ou 

```
SELECT* FROM cours_classe INTERSECT SELECT* FROM cours_enseignant
```

Ou

```
SELECT id, annee_acad, cours_id FROM cours_classe INTERSECT
SELECT id, annee_acad, cours_id FROM cours_enseignant
```

- Résultat:

| + Options |            |          |
|-----------|------------|----------|
| id        | annee_acad | cours_id |
| 1         | 2019-2020  | 1        |
| 2         | 2019-2020  | 2        |
| 3         | 2019-2020  | 3        |
| 4         | 2019-2020  | 4        |
| 5         | 2019-2020  | 5        |
| 6         | 2019-2020  | 6        |
| 9         | 2019-2020  | 9        |
| 10        | 2019-2020  | 10       |

Figure 3.6. L'intersection des tables : cours\_enseignant et cours\_classe

### c) La Différence

La différence est une opération portant sur deux relations R1 et R2 ayant le même schéma et construisant une troisième relation dont les n-uplets sont constitués de ceux ne se trouvant que dans la relation R1 ; on la note  $R1 - R2$ .

Il s'agit une opération binaire ensembliste non commutative essentielle dont la signature est : relation  $\times$  relation  $\rightarrow$  relation

Comme nous l'avons déjà dit, R1 et R2 doivent avoir les mêmes attributs. Le résultat de la différence est une nouvelle relation qui a les mêmes attributs que R1 et R2. Si R1 est vide, la relation qui résulte de la différence est vide. Si R2 est vide, la relation qui résulte de la différence est identique à R1.

**Afficher la différence entre la table cours\_classe et cours\_enseignant.**

- Requête en algèbre relationnelle :  $R = R_1 - R_2$

Requête en SQL : `SELECT cours_classe.id, cours_classe.annee_acad, cours_classe.cours_id FROM cours_classe LEFT JOIN cours_enseignant ON cours_classe.id= cours_enseignant.id`

Ou

`SELECT* FROM cours_classe EXCEPT SELECT* FROM cours_enseignant`

Ou

`SELECT id, annee_acad,cours_id FROM cours_classe EXCEPT SELECT id, annee_acad,cours_id FROM cours_enseignant`

- Résultat:

| + Options |            |          |
|-----------|------------|----------|
| id        | annee_acad | cours_id |
| 1         | 2019-2020  | 1        |
| 2         | 2019-2020  | 2        |
| 3         | 2019-2020  | 3        |
| 4         | 2019-2020  | 4        |
| 5         | 2019-2020  | 5        |
| 6         | 2019-2020  | 6        |
| 7         | 2019-2020  | 7        |
| 8         | 2019-2020  | 8        |
| 9         | 2019-2020  | 9        |
| 10        | 2019-2020  | 10       |

Figure 3.7. La différence entre la table cours\_classe et cours\_enseignant

### III.2.3. Les opérateurs binaires ou n-aires

#### a) Produit cartésien

Le produit cartésien est une opération portant sur deux relations R1 et R2 et qui construit une troisième relation regroupant exclusivement toutes les possibilités de combinaison des occurrences des relations R1 et R2, on la note  $R1 \times R2$ . Il s'agit une opération binaire commutative essentielle dont la signature est :  $relation \times relation \rightarrow relation$

Le résultat du produit cartésien est une nouvelle relation qui a tous les attributs de R1 et tous ceux de R2. Si R1 ou R2 ou les deux sont vides, la relation qui résulte du produit cartésien est vide. Le nombre d'occurrences de la relation qui résulte du produit cartésien est le nombre d'occurrences de R1 multiplié par le nombre d'occurrences de R2.

#### Afficher le produit cartésien des cours affectés dans une classe

- Requête en algèbre relationnelle :  $R = classe \times cours$

- Requête en SQL : `SELECT * FROM classe, cours`

Ou

`SELECT * FROM classe CROSS JOIN cours`

- Résultat:

| + Options |          |             |                |    |        |                           |
|-----------|----------|-------------|----------------|----|--------|---------------------------|
| id        | capacite | designation | departement_id | id | credit | designation               |
| 1         | 500      | BAC1        | 1              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 2         | 300      | BAC2        | 1              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 3         | 200      | BAC3        | 1              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 4         | 1000     | BAC1        | 3              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 5         | 800      | BAC2        | 3              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 6         | 500      | BAC3        | 3              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 8         | 45       | G4          | 1              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 9         | 1000     | G5          | 3              | 1  | 75     | MATHEMATIQUES GENERALES   |
| 1         | 500      | BAC1        | 1              | 2  | 30     | HISTOIRE DU CONGO         |
| 2         | 300      | BAC2        | 1              | 2  | 30     | HISTOIRE DU CONGO         |
| 3         | 200      | BAC3        | 1              | 2  | 30     | HISTOIRE DU CONGO         |
| 4         | 1000     | BAC1        | 3              | 2  | 30     | HISTOIRE DU CONGO         |
| 5         | 800      | BAC2        | 3              | 2  | 30     | HISTOIRE DU CONGO         |
| 6         | 500      | BAC3        | 3              | 2  | 30     | HISTOIRE DU CONGO         |
| 8         | 45       | G4          | 1              | 2  | 30     | HISTOIRE DU CONGO         |
| 9         | 1000     | G5          | 3              | 2  | 30     | HISTOIRE DU CONGO         |
| 1         | 500      | BAC1        | 1              | 3  | 45     | REDACTION FRANCAISE       |
| 2         | 300      | BAC2        | 1              | 3  | 45     | REDACTION FRANCAISE       |
| 3         | 200      | BAC3        | 1              | 3  | 45     | REDACTION FRANCAISE       |
| 4         | 1000     | BAC1        | 3              | 3  | 45     | REDACTION FRANCAISE       |
| 5         | 800      | BAC2        | 3              | 3  | 45     | REDACTION FRANCAISE       |
| 6         | 500      | BAC3        | 3              | 3  | 45     | REDACTION FRANCAISE       |
| 8         | 45       | G4          | 1              | 3  | 45     | REDACTION FRANCAISE       |
| 9         | 1000     | G5          | 3              | 3  | 45     | REDACTION FRANCAISE       |
| 1         | 500      | BAC1        | 1              | 4  | 45     | STATISTIQUES DESCRIPTIVES |

Figure 3.8. Le produit cartésien des cours affectés dans une classe

## b) Jointure, theta-jointure, equi-jointure, jointure naturelle

- **Jointure** : La jointure est une opération portant sur deux relations R1 et R2 qui construit une troisième relation regroupant exclusivement toutes les possibilités de combinaison des occurrences des relations R1 et R2 qui satisfont l'expression logique E. La jointure est notée  $R1 \bowtie_{\langle E \rangle} R2$ .

Il s'agit d'une opération binaire commutative dont la signature est : relation  $\times$  relation  $\times$  expression logique  $\longrightarrow$  relation

Si R1 ou R2 ou les deux sont vides, la relation qui résulte de la jointure est vide.

En fait, la jointure n'est rien d'autre qu'un produit cartésien suivi d'une sélection :  $R1 \bowtie_{\langle E \rangle} R2 = \sigma_E (R1 \times R2)$

### Afficher la jointure entre département, cours et classe.

- Requête en algèbre relationnelle :  $R = \text{departement} \bowtie_{\langle (\text{nombre\_enseignant} \leq \text{capacite}) \wedge (\text{Prixcredit} \geq 30) \rangle} \text{cours, classe}$
- Requête en SQL : `SELECT * FROM departement, cours, classe WHERE nombre_enseignant <= capacite and credit >= 30`
- Résultat:

| + Options |      |                                       |                   |               |    |        |                         |    |          |             |                |
|-----------|------|---------------------------------------|-------------------|---------------|----|--------|-------------------------|----|----------|-------------|----------------|
| id        | code | designation                           | nombre_enseignant | enseignant_id | id | credit | designation             | id | capacite | designation | departement_id |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 1  | 75     | MATHEMATIQUES GENERALES | 1  | 500      | BAC1        | 1              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 1  | 500      | BAC1        | 1              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 1  | 75     | MATHEMATIQUES GENERALES | 1  | 500      | BAC1        | 1              |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 1  | 75     | MATHEMATIQUES GENERALES | 2  | 300      | BAC2        | 1              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 2  | 300      | BAC2        | 1              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 1  | 75     | MATHEMATIQUES GENERALES | 2  | 300      | BAC2        | 1              |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 1  | 75     | MATHEMATIQUES GENERALES | 3  | 200      | BAC3        | 1              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 3  | 200      | BAC3        | 1              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 1  | 75     | MATHEMATIQUES GENERALES | 3  | 200      | BAC3        | 1              |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 1  | 75     | MATHEMATIQUES GENERALES | 4  | 1000     | BAC1        | 3              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 4  | 1000     | BAC1        | 3              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 1  | 75     | MATHEMATIQUES GENERALES | 4  | 1000     | BAC1        | 3              |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 1  | 75     | MATHEMATIQUES GENERALES | 5  | 800      | BAC2        | 3              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 5  | 800      | BAC2        | 3              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 1  | 75     | MATHEMATIQUES GENERALES | 5  | 800      | BAC2        | 3              |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 1  | 75     | MATHEMATIQUES GENERALES | 6  | 500      | BAC3        | 3              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 6  | 500      | BAC3        | 3              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 1  | 75     | MATHEMATIQUES GENERALES | 6  | 500      | BAC3        | 3              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 8  | 45       | G4          | 1              |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 1  | 75     | MATHEMATIQUES GENERALES | 9  | 1000     | G5          | 3              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 1  | 75     | MATHEMATIQUES GENERALES | 9  | 1000     | G5          | 3              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 1  | 75     | MATHEMATIQUES GENERALES | 9  | 1000     | G5          | 3              |
| 1         | GI   | GENIE INFORMATIQUE                    | 50                | 5             | 2  | 30     | HISTOIRE DU CONGO       | 1  | 500      | BAC1        | 1              |
| 2         | MP   | MATHEMATIQUES PHYSIQUES               | 30                | 6             | 2  | 30     | HISTOIRE DU CONGO       | 1  | 500      | BAC1        | 1              |
| 3         | SCA  | SCIENCES COMMERCIALES ADMINISTRATIVES | 60                | NULL          | 2  | 30     | HISTOIRE DU CONGO       | 1  | 500      | BAC1        | 1              |

Figure 3.9. La jointure entre département, cours et classe

### c) Theta-jointure

Une theta-jointure est une jointure dans laquelle l'expression logique E est une simple comparaison entre un attribut A1 de la relation R1 et un attribut A2 de la relation R2. La theta-jointure est notée  $R1 \bowtie E R2$ .

#### Afficher le Theta-jointure sur département et classe

- Requête en algèbre relationnelle :  $R = \text{departement} \bowtie_{(\text{nombre\_enseignant} \leq \text{capacite})} \text{classe}$
- Requête en SQL : `SELECT * FROM departement, classe WHERE nombre_enseignant > capacite`
- Résultat:

| id | code | designation         | nombre_enseignant | id | capacite | designation | departement_id |
|----|------|---------------------|-------------------|----|----------|-------------|----------------|
| 2  | GI   | GENIE INFORMATIQUE  | 60                | 3  | 50       | BAC3        | 1              |
| 1  | MG   | MEDECINES GENERALES | 50                | 5  | 45       | MANAGEMENT  | 1              |
| 2  | GI   | GENIE INFORMATIQUE  | 60                | 5  | 45       | MANAGEMENT  | 1              |
| 1  | MG   | MEDECINES GENERALES | 50                | 7  | 30       | BAC1 Maths  | 6              |
| 2  | GI   | GENIE INFORMATIQUE  | 60                | 7  | 30       | BAC1 Maths  | 6              |
| 1  | MG   | MEDECINES GENERALES | 50                | 8  | 30       | BAC2 Maths  | 6              |
| 2  | GI   | GENIE INFORMATIQUE  | 60                | 8  | 30       | BAC2 Maths  | 6              |
| 1  | MG   | MEDECINES GENERALES | 50                | 9  | 30       | BAC3 Maths  | 6              |
| 2  | GI   | GENIE INFORMATIQUE  | 60                | 9  | 30       | BAC3 Maths  | 6              |

Figure 3.10. Le Theta-jointure sur département et classe

### d) Jointure naturelle

Une jointure naturelle est une jointure dans laquelle l'expression logique E est un test d'égalité entre les attributs qui portent le même nom dans les relations R1 et R2. Dans la relation construite, ces attributs ne sont pas dupliqués mais fusionnés en une seule colonne par couple d'attributs. La jointure naturelle est notée  $R1 \bowtie R2$ . On peut préciser explicitement les attributs communs à R1 et R2 sur lesquels porte la jointure :  $R1 \bowtie \langle A1, \dots, An \rangle R2$ .

Généralement, R1 et R2 n'ont qu'un attribut en commun. Dans ce cas, une jointure naturelle est équivalente à une equi-jointure dans laquelle l'attribut de R1 et celui de R2 sont justement les deux attributs qui portent le même nom.

Lorsque l'on désire effectuer une jointure naturelle entre R1 et R2 sur un attribut A1 commun à R1 et R2, il vaut mieux écrire  $R1 \bowtie \langle A1 \rangle R2$  que  $R1 \bowtie R2$ . En effet, si R1 et R2 possèdent deux attributs portant un nom commun, A1 et A2,  $R1 \bowtie \langle A1 \rangle R2$  est bien une jointure naturelle

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

sur l'attribut A1, mais  $R1 \triangleright \triangleleft R2$  est une jointure naturelle sur le couple d'attributs A1, A2, ce qui produit un résultat très différent !

**Afficher la jointure naturelle.**

- Requête en algèbre relationnelle :  $R = \text{Note} \triangleright \triangleleft \text{Classe\_etudiant}$  ou  $R = \text{Famille} \triangleright \triangleleft \text{etudiant\_id Classe\_etudiant}$
- Requête en SQL : `SELECT * FROM note, classe_etudiant WHERE note.etudiant_id=classe_etudiant.etudiant_id`
- Résultat:

| id | annee     | examen | session | tj | cours_id | etudiant_id | id | annee_acad | classe_id | etudiant_id |
|----|-----------|--------|---------|----|----------|-------------|----|------------|-----------|-------------|
| 1  | 2019-2020 | 5      | S1      | 6  | 1        | 2           | 1  | 2019-2020  | 1         | 2           |
| 5  | 2019-2020 | 7      | S1      | 10 | 2        | 2           | 1  | 2019-2020  | 1         | 2           |
| 9  | 2019-2020 | 6      | S2      | 7  | 1        | 2           | 1  | 2019-2020  | 1         | 2           |
| 13 | 2019-2020 | 6      | S2      | 3  | 2        | 2           | 1  | 2019-2020  | 1         | 2           |
| 2  | 2019-2020 | 7      | S1      | 4  | 1        | 3           | 2  | 2019-2020  | 1         | 3           |
| 6  | 2019-2020 | 6      | S1      | 6  | 2        | 3           | 2  | 2019-2020  | 1         | 3           |
| 10 | 2019-2020 | 4      | S2      | 2  | 1        | 3           | 2  | 2019-2020  | 1         | 3           |
| 14 | 2019-2020 | 7      | S2      | 4  | 2        | 3           | 2  | 2019-2020  | 1         | 3           |
| 3  | 2019-2020 | 8      | S1      | 8  | 1        | 4           | 3  | 2019-2020  | 1         | 4           |
| 7  | 2019-2020 | 2      | S1      | 7  | 2        | 4           | 3  | 2019-2020  | 1         | 4           |
| 11 | 2019-2020 | 6      | S2      | 2  | 1        | 4           | 3  | 2019-2020  | 1         | 4           |
| 15 | 2019-2020 | 6      | S2      | 7  | 2        | 4           | 3  | 2019-2020  | 1         | 4           |
| 4  | 2019-2020 | 4      | S1      | 2  | 1        | 9           | 8  | 2019-2020  | 1         | 9           |
| 8  | 2019-2020 | 2      | S1      | 3  | 2        | 9           | 8  | 2019-2020  | 1         | 9           |
| 12 | 2019-2020 | 4      | S2      | 7  | 1        | 9           | 8  | 2019-2020  | 1         | 9           |
| 16 | 2019-2020 | 2      | S2      | 3  | 2        | 9           | 8  | 2019-2020  | 1         | 9           |
| 17 | 2019-2020 | 9      | S1      | 8  | 6        | 28          | 15 | 2019-2020  | 7         | 28          |
| 18 | 2019-2020 | 7      | S1      | 10 | 7        | 28          | 15 | 2019-2020  | 7         | 28          |

Figure 3.11. La jointure naturelle

## **Conclusion**

Ce chapitre nous a lancé dans l'utilisation de l'algèbre relationnelle. L'algèbre relationnelle étant un outil mathématique qui nous a été utile dans l'exécution de la Base de données dite relationnelle en représentant des tables relationnelles en extension d'une relation définie sur un domaine. Cette méthode a consisté essentiellement à créer ou à recréer de nouvelles relations à partir de relations existantes. Nous avons suivi dans ce chapitre une approche plus opérationnelle que mathématique qui nous a permis à définir donc un petit langage de manipulation de données exploité dans ce chapitre à titre d'exemple la relation de la jointure entre département, cours et classe. Nous avons aussi distingué dans ce chapitre, trois familles d'opérateurs relationnels : les opérateurs unaires, les opérateurs binaires ensemblistes et les opérateurs binaires ou n-aires.

Dans le chapitre suivant, nous allons entamer la phase de l'implémentation.

## **Chapitre IV. L'IMPLEMENTATION DU SYSTEME « GESTUNI »**

Ce chapitre représente la dernière partie de ce travail de mémoire, il traite la phase qui a pour objectif l'implémentation de notre application. Nous débutons, tout d'abord, par la description des technologies utilisées pour développer notre application. Ensuite nous mentionnons les outils technologiques utilisés et en donnant quelques détails sur le SGBD utilisé et enfin la maquette de notre application sera illustrée.

### **IV.1. Les technologies utilisées**

Pour aboutir à l'aboutissement de notre application, nous avons fait recours sur les outils ci-après :

#### **IV.1.1 Le langage Java**

Notre attachement au langage Java est du fait que c'est un langage de programmation le plus utilisé de la planète, et le langage par défaut pour le développement d'applications natives Android. La force de Java est un langage orienté objet est particulièrement utile lorsqu'il s'agit de réaliser des applications commerciales complexes en ligne et hors ligne. Combinées à l'environnement d'exécution Java (JRE), ces applications se démarquent principalement par leur indépendance vis-à-vis des plateformes, ce qui les rend accessibles à un vaste public.

Pour faciliter le travail avec Java, différentes infrastructures logicielles ont été développées au cours de cette dernière décennie afin de mettre à disposition de l'utilisateur une trame de base prête à l'emploi pour le programme Java qu'on souhaite développer. L'une des plus connues est l'infrastructure logicielle en open source Spring que nous avons utilisé pour le développement de notre application, publiée en 2003, qui diminue nettement la complexité de la spécification Java standard (« J2EE » ou « plateforme Java ») mais aussi de l'architecture de composants Enterprise JavaBeans (« EJB »). Après la mise à disposition de l'extension Spring Boot, qui simplifie encore la configuration de nouveaux projets Spring, cette trame s'est révélée un choix optimal et convivial pour toutes les personnes ne souhaitant pas commencer le développement d'une nouvelle application Java entièrement de zéro.[13]

#### **IV.1.2. Principales caractéristiques du langage Java**

Une première caractéristique du langage Java est qu'il est pleinement orienté objet. La raison de ce choix est que Java se veut portable : le programmeur ne souhaite plus faire des adaptations pour que son programme puisse fonctionner sur telle ou telle machine. La librairie Java est très riche : elle est constituée de plusieurs milliers de classes, elles-mêmes souvent constituées de plusieurs dizaines de méthodes (de fonctions si vous préférez). Pour l'heure, notez seulement que l'on peut quasiment tout faire en Java : de l'accès aux bases de données, de

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

la manipulation de fichiers XML, du graphisme, des applications distribuées, des applications Web, ...

Nous retenons que les caractéristiques essentielles de java selon les concepteurs de Java (James Gosling et Bill Joy, Sun Microsystem) ont conçu Java comme un langage :

- Indépendant de la machine et du système d'exploitation sous-jacent ;
- Complet : Programmation réseau avancée, Programmation d'interface utilisateur graphique ;
- Simple : La gestion de la mémoire n'est pas à la charge d'utilisateur.

#### IV.2. Stratégie d'héritage avec JPA [14]

Les bases de données relationnelles n'ont pas une fonctionnalité permettant de traduire ou mapper une hiérarchie de classe en tables de bases de données, or l'héritage est un des concepts clefs en java en particulier et de la programmation orientée objet en général. JPA et ses différentes implémentations doivent donc fournir un moyen pour traduire ce concept clef en un concept compréhensible par les bases de données relationnelles.

Pour résoudre ce problème, JPA propose plusieurs stratégies :

- Single Table : les entités de la hiérarchie de classe sont placées dans une seule table.
- Table-Per-Class : une table par classe.
- Joined Table : chaque classe a sa table et effectuer une requête sur une sous-classe de la hiérarchie implique de faire une jointure sur les tables.

##### ➤ **Single\_Table**

C'est la stratégie par défaut utilisée par JPA et utilisée dans notre travail. Avec cette stratégie, une seule table est créée et partagée par toutes les classes de la hiérarchie. Le mode **SINGLE\_TABLE** correspond au cas où super-entités et sous-entités sont représentées par une seule table en tout et pour tout en base de données. Cette table contient donc tous les champs de la super-classe plus tous ceux de toutes les sous-classes. De plus, une colonne sert de discriminant pour déterminer de quel type doit être tel ou tel tuple. Dans notre exemple, cela signifie qu'il n'y a qu'une table « Personne » qui comporte les champs id, matricule, nom, postnom, prenom, sexe, lieuNaissance, dateNaissance, adresse, telephone, adressemail et enfin une colonne de discrimination dont le nom par défaut est «[TYPE\\_PERSONNE](#)».

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

L'annotation `@Inheritance` est utilisée sur la classe mère pour préciser à JPA la stratégie d'héritage à utiliser. Alors pour notre cas, la classe mère **Personne** est donc :

```
@Entity
@Data
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name = "TYPE_PERSONNE",discriminatorType =
DiscriminatorType.STRING,length = 10)
public abstract class Personne implements Serializable{
@Id @GeneratedValue (strategy = GenerationType.IDENTITY)
    private Long id;
    private String matricule;
    private String nom;
    private String postnom;
    private String prenom;
    private String sexe;
    private String lieuNaissance;
    private Date dateNaissance;
    private String adresse;
    private String telephone;
    private String adressemail;
    private boolean etat;

// constructeur, getters, setters }
}
```

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

La table se présentera de la manière suivante :

| ▼         | type_personne | id | adresse  | adressemail         | date_naissance      | etat | lieu_naissance | matricule      | nom                 | password                         | postnom    | prenom     | sexe  | telephone   | domaine      | grade   | contrat |
|-----------|---------------|----|----------|---------------------|---------------------|------|----------------|----------------|---------------------|----------------------------------|------------|------------|-------|-------------|--------------|---------|---------|
| Supprimer | ADMIN         | 1  | Bukavu   | kiseki@gmail.com    | 2020-05-14 08:04:42 | 1    | Bukavu         | SDF4SSGS745845 | Kiseki              | 9f112bc52a09fc35e9320089d7e37480 | Asumani    | Jean       | Homme | +2438785415 | NULL         | NULL    | NULL    |
| Supprimer | ETUDIANT      | 2  | KAMENGE  | eloge@gmail.com     | 2017-12-31 22:00:00 | 1    | BUKAVU         | 133222         | ELOGE               | 25f9e794323b453885f5181f1b624d0b | LUKEJOS    | BINGWA     | Homme | 099456876   | NULL         | NULL    | NULL    |
| Supprimer | ETUDIANT      | 3  | RIHERO   | jeanneb@gmail.com   | 2017-01-01 22:00:00 | 1    | BUKAVU         | 46666          | JEANNE              | 25f9e794323b453885f5181f1b624d0b | BITAMWILO  | ELODIE     | M     | 25466598777 | NULL         | NULL    | NULL    |
| Supprimer | ETUDIANT      | 4  | BUKAVU   | faida@gmail.com     | 2013-12-29 22:00:00 | 0    | BUKAVU         | EKJIED41457744 | MARIE               | 25f9e794323b453885f5181f1b624d0b | ROSE       | WASSO      | Homme | 098655755   | NULL         | NULL    | NULL    |
| Supprimer | ENSEIGNANT    | 5  | BUKAVU   | wilcojean@gmail.com | 2012-12-30 22:00:00 | 1    | BUKAVU         | N23336         | KISEKI              | 25f9e794323b453885f5181f1b624d0b | WILONDEJA  | JEAN PAUL  | M     | 0236589988  | INFORMATIQUE | DOCTEUR | NULL    |
| Supprimer | ENSEIGNANT    | 6  | BUKAVU   | marierose@gmail.com | 2013-12-29 22:00:00 | 1    | BUKAVU         | N66666         | WASSO               | 25f9e794323b453885f5181f1b624d0b | FAIDA      | MARIE ROSE | F     | 236989898   | MEDECINE     | DOCTEUR | NULL    |
| Supprimer | ENSEIGNANT    | 7  | BUKAVU   | elodieb@gmail.com   | 2017-12-31 22:00:00 | 1    | BUKAVU         | N56666         | WASSO               | 25f9e794323b453885f5181f1b624d0b | BITAMWILO  | JEANNE     | F     | 2656565959  | MATHEMATIQUE | DOCTEUR | NULL    |
| Supprimer | ENSEIGNANT    | 8  | BUKAVU   | lukekaw@gmail.com   | 2018-12-30 22:00:00 | 1    | BUKAVU         | N466666        | WASSO               | 25f9e794323b453885f5181f1b624d0b | LUKEKA     | ELOGE      | M     | 32326666666 | MEDECINE     | DOCTEUR | NULL    |
| Supprimer | ENSEIGNANT    | 9  | KITUTU   | wenga@gmail.com     | 2018-12-30 22:00:00 | 1    | BUKAVU         | N65656468      | WASSO               | 25f9e794323b453885f5181f1b624d0b | KALESHI    | WENGA      | M     | 66996566    | INFORMATIQUE | DOCTEUR | NULL    |
| Supprimer | ENSEIGNANT    | 10 | BAKONGO  | bingwa@gmail.com    | NULL                | 0    | BILIZA         | N66457         | BINGWA              | 25f9e794323b453885f5181f1b624d0b | SOLE       | KINOISerie | M     | 6666666666  | INFORMATIQUE | DOCTEUR | NULL    |
| Supprimer | ADMIN         | 11 | NDJILI   | domingo@gmail.com   | 1989-12-31 22:00:00 | 1    | BILIZA         | N6666          | DOMINIQUE           | 25f9e794323b453885f5181f1b624d0b | WASSO      | BINGWA     | M     | 6666466     | NULL         | NULL    | NULL    |
| Supprimer | ETUDIANT      | 12 | BUKAVU   | faidrosa@gmail.com  | 2013-12-29 22:00:00 | 0    | BUKAVU         | EKJIED41457744 | MARIE               | 25f9e794323b453885f5181f1b624d0b | ROSE       | WASSO      | Homme | 098655755   | NULL         | NULL    | NULL    |
| Supprimer | ETUDIANT      | 13 | hjhbjbjk | jjjjoj@gmail.com    | 1997-12-28 22:00:00 | 0    | hjhujhi        | EKJIED41457744 | hjqdhqsjdhsjkhdkqsj | 25f9e794323b453885f5181f1b624d0b | sfsfqsfqfq | hiuhuhj    | Homme | 56465456666 | NULL         | NULL    | NULL    |
| Supprimer | ETUDIANT      | 14 | BUKAVU   | bikubu@gmail.com    | 2013-12-29 22:00:00 | 0    | BUKAVU         | EKJIED41457744 | BIKUBU              | a970a7e3b359f88a4732b56050822888 | MAYAYA     | ODETTE     | Femme | 032569988   | NULL         | NULL    | NULL    |
| Supprimer | ETUDIANT      | 15 | BAGIRA   | cirubaka@gmail.cpm  | 2013-12-29 22:00:00 | 0    | BUK            | EKJIED41457744 | CIRUBAKADERHA       | 70873a8580c9900986939611618d7b1e | MUBALAMA   | STANISLAS  | Homme | 02398989    | NULL         | NULL    | NULL    |

Figure 4.1. La présentation de la Table Personne

#### IV.2.1 Mise en relation d'entités

Tout comme en SQL, on peut définir quatre types de relations entre entités JPA :

- Relation 1:1 : annotée par @OneToOne ;
- Relation n:1 : annotée par @ManyToOne ;
- Relation 1:n : annotée par @OneToMany ;
- Relation n:n : annotée par @ManyToMany.

Bien qu'exprimées entre des classes Java, les relations sont définies entre des entités JPA.

Ici nous présentons un exemple de l'entité **CoursEnseignant** dans laquelle nous trouvons le cas de relations entre entités JPA représentée sous la relation 1 :1 : annotée par @OneToOne ;

@Entity

```
public class CoursEnseignant {  
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    @Column(length = 10)  
    private String anneeAcad;  
    @ManyToOne  
    private Cours cours;  
    @ManyToOne  
    private Enseignant enseignant;  
    public CoursEnseignant() {  
        super();  
        // TODO Auto-generated constructor stub }  
}
```

#### IV.2.2 Les plates-formes de Java

**Le Java SE (Java Standard Edition) aussi appelé JDK** : cette distribution est utilisée pour développer des applications Java qui s'exécuteront directement sur un poste client. Il est décliné sous deux principales formes :

- Le JRE (Java Runtime Environment) : il s'agit juste de l'environnement d'exécution. Il est nécessaire pour l'exécution d'un programme Java sur un poste client.
- Le JDK (Java Development Kit) : c'est un JRE complété d'un certain nombre d'outils de développement (compilateurs notamment) et de profilage d'applications Java.

**Le Java EE (Java Enterprise Edition)** : un complément au Java SE permettant le développement d'application d'entreprise. On déploie les applications développées via ce standard sur un serveur d'applications compatible J2EE (Web Sphere, JBoss, Web Logic).

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

**Le Java ME (Java Micro Edition)** : un ensemble de distribution "light" pour des plates-formes embarquées (téléphonie mobile, par exemple).

### **IV.3 Des applications Java avec le framework [op-cit]**

Pour ce qui est de notre cas de recherche, nous avons opté pour le framework spring boot que nous développons dans le point suivant.

#### **a) Spring Boot**

Spring Boot est une solution de « convention plutôt que configuration » destinée à l'infrastructure logicielle Java Spring qui réduit la complexité de la configuration de nouveaux projets Spring. À cette fin, Spring Boot définit une configuration de base incluant des directives pour l'utilisation de l'infrastructure logicielle ainsi que toutes les bibliothèques de prestataires tiers pertinentes, ce qui permet de faciliter autant que possible la création de nouveaux projets. Cette méthode simplifie considérablement la création d'applications indépendantes prêtes pour la production, ce qui explique pourquoi la majorité des nouvelles applications Spring reposent en grande partie sur Spring Boot. Les caractéristiques de Spring Boot est une boîte à outils très riche permettant de structurer, d'améliorer et de simplifier l'écriture d'application JEE. Spring est organisé en module et peut être résumé comme suit :

- L'intégration directe d'applications de serveur Web/de conteneur comme Apache Tomcat ou Jetty sans utiliser de fichiers WAR (Web Application Archive) ;
- La configuration simplifiée de Maven grâce à des POM (Project Object Models) « Starter » ;
- Lorsque c'est possible, la configuration automatique de Spring ;
- Gestion des instances de classes (JavaBean et/ou métier) ;
- Modèle MVC et outils pour les applications WEB ;
- Outils pour la DAO (JDBC) ;
- Outils pour les ORM (Hibernate, iBatis, ...) ;
- Outils pour les applications JEE (JMX, JMA, JCA, EJB, ...)

### b) Le Spring MVC [15]

Le framework Spring MVC a une architecture de type MVC (**M**odel-**V**iew-**C**ontroller) et ses composants servent pour développer des applications Web flexibles et faiblement couplées. Dans Spring-MVC :

- Le modèle correspond à une table d'une base de données. C'est une classe qui étend la classe Model, qui gère simplement et efficacement les manipulations de données et l'établissement automatisé de relation entre tables. Avec spring boot c'est un rôle joué par Hibernate qui s'occupe de tout ce qui est de la base de données. En bref, est chargé de gérer les données.
- Le contrôleur est chargé de gérer l'ensemble des activités. La tâche d'un contrôleur est de réceptionner une requête et définir une réponse appropriée au client.
- La vue est soit un simple fichier avec du code HTML. Produit les pages HTML.

En général, on résume en disant que le modèle gère la base de données, la vue produit les pages HTML et le contrôleur fait tout le reste. Plus précisément, détaillons cette organisation dans Spring.

### c) L'architecture de l'application

La présentation.

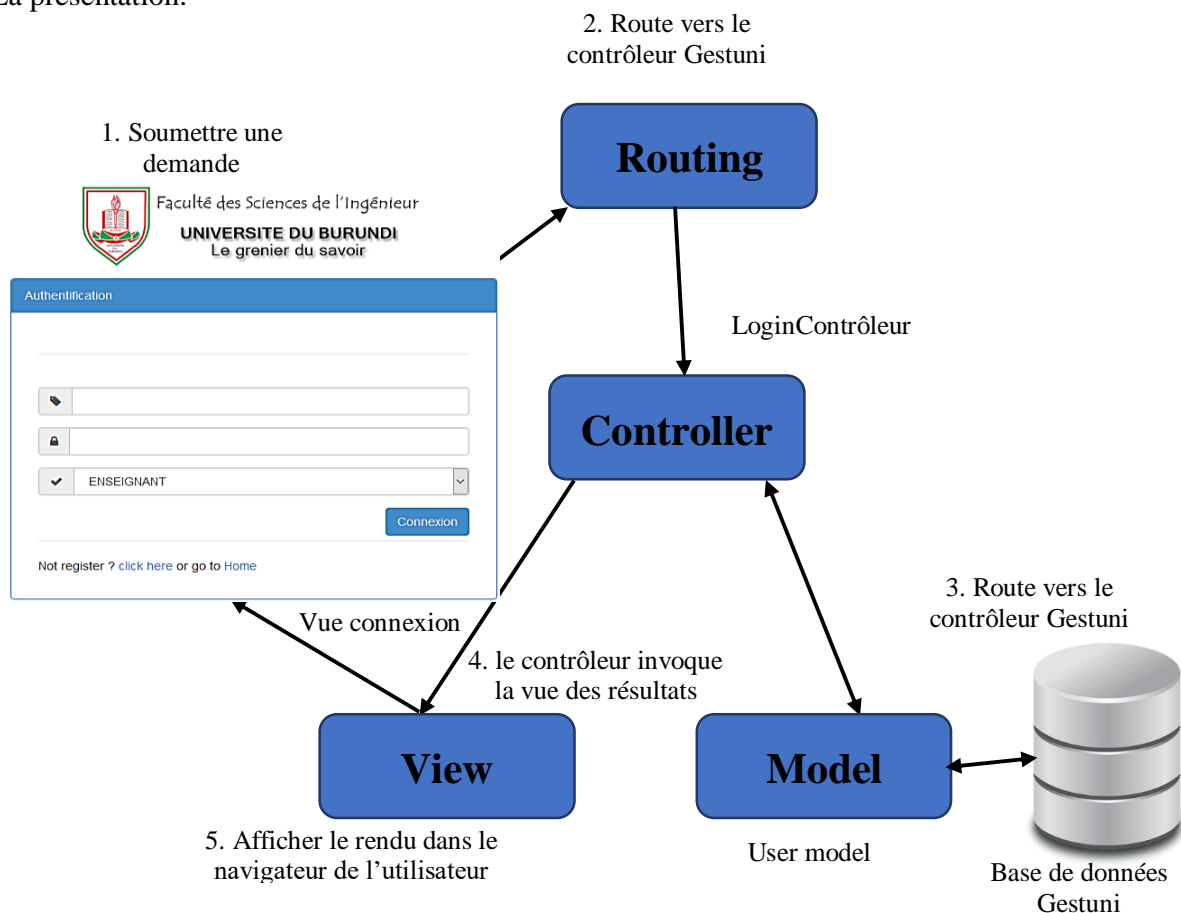


Figure 4.2. Architecture MVC de l'application Gestuni

#### IV.4. Quelques implémentations

Dans ce point, nous prenons en compte l'implémentation de quelques entités utilisées dans notre application Gestuni. Il s'agira de la Super Classe Personne, Enseignant, Classe, Etudiant et Document-Etudiant.

##### L'implémentation de l'entité Super Classe Personne :

```
@Entity
@Data
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name = "TYPE_PERSONNE",discriminatorType =
DiscriminatorType.STRING,length = 10)
public class Personne implements Serializable{
    @Id @GeneratedValue (strategy = GenerationType.IDENTITY)
    private Long id;
    private String matricule;
    @Column(length = 50) @NotEmpty @Size(min = 3, max = 50)
    private String nom;
    @Column(length = 50) @NotEmpty @Size(min = 3, max = 50)
    private String postnom;
    @Column(length = 50) @NotEmpty @Size(min = 3, max = 50)
    private String prenom;
    @Column(length = 6) @NotEmpty @Size(min = 1, max = 6)
    private String sexe;
    @Column(length = 30) @NotEmpty @Size(min = 3, max = 30)
    private String lieuNaissance;
    @DateTimeFormat(pattern = "YYYY-MM-d")
    private Date dateNaissance;
    @Column(length = 100) @NotEmpty @Size(min = 3, max = 100)
    private String adresse;
    @Column(length = 13) @NotEmpty @Size(max = 13)
    private String telephone;
    @Column(length = 70) @NotEmpty @Email
    private String adressemail;
    @Column(length = 200) @NotEmpty
    private String password;
    private boolean etat;
    public Personne() {
    super();
    }
}
```

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

### La présentation de l'entité Super Classe Personne :

| type_personne | id | adresse       | adressemail        | date_naissance      | etat | lieu_naissance | matricule      | nom           | password                         | postnom | prenom    | sexe  | telephone   | domaine            | grade     |
|---------------|----|---------------|--------------------|---------------------|------|----------------|----------------|---------------|----------------------------------|---------|-----------|-------|-------------|--------------------|-----------|
| ADMIN         | 1  | Bukavu        | kiseki@gmail.com   | 2020-03-20 20:54:33 | 1    | Bukavu         | SDF4SSGS745845 | Kiseki        | 9f112bc52a09fc35e9320089d7e37480 | Asumani | Jean      | Homme | +2438785415 |                    | NULL NULL |
| ENSEIGNANT    | 2  | NJILI PLATEAU | jobogo@gmail.com   | 2019-12-30 08:00:00 | 1    | BUKAVU         | 002/2019-2020  | NTUMWA        | 25d55ad283aa400af464c76d713c07ad | JOBOGO  | FRANCOISE | M     | 1496832588  | Economie           | Dr pHd    |
| ENSEIGNANT    | 3  | KATANA CENTRE | jonathan@gmail.com | 2019-12-30 08:00:00 | 1    | ISOGHA         | 001/2019-2020  | FAIDA         | 25d55ad283aa400af464c76d713c07ad | MONGANE | FRANCOISE | M     | 1496832588  | GENIE INFORMATIQUE | Dr pHd    |
| ETUDIANT      | 4  | NJILI PLATEAU | eloge1@gmail.com   | 2014-12-29 08:00:00 | 1    | ISOGHA         | 002/2019-2020  | ZIRIMWABAGABO | 25d55ad283aa400af464c76d713c07ad | MONGANE | CLEMENT   | M     | 1496832585  |                    | NULL NULL |
| ETUDIANT      | 5  | BAGIRA        | kasindi@gmail.com  | 2019-12-30 08:00:00 | 1    | ISOGHA         | 001/2019-2020  | KASINDI       | 25d55ad283aa400af464c76d713c07ad | LUGANZU | FRANCOISE | Femme | 4445465466  |                    | NULL NULL |
| ENSEIGNANT    | 6  | IRAMBO        | mugaruka@gmail.com | 1997-12-29 08:00:00 | 1    | BUKAVU         | 6565JJK        | MUGARUKA      | 25d55ad283aa400af464c76d713c07ad | GULAIN  | BUDUGE    | M     | 69918762    | Economie           | MASTERE   |
| ETUDIANT      | 7  | NJILI PLATEAU | bingwa@gmail.com   | 2019-12-30 08:00:00 | 0    | KABARE         | EKJIED41457744 | KASINDI       | 25d55ad283aa400af464c76d713c07ad | MONGANE | CLEMENT   | Homme | 03654496    |                    | NULL NULL |
| ETUDIANT      | 8  | NJILI PLATEAU | ndik@gmail.com     | 2019-12-30 08:00:00 | 0    | ISOGHA         | EKJIED41457744 | NDIKUMAGENGE  | 25f9e794323b453885f5181f1b624d0b | MONGANE | Zephyrin  | Homme | 4445465466  |                    | NULL NULL |

Figure 4.3. Architecture de l'entité Per Personne 1

### L'implémentation de l'entité Enseignant :

```
@Entity
@DiscriminatorValue("ENSEIGNANT")
public class Enseignant extends Personne{
    @NotEmpty @Size(min = 3, max = 50)
    private String domaine;
    @NotEmpty @Size(min = 3, max = 50)
    private String grade;
    @OneToMany(mappedBy = "enseignant", fetch = FetchType.LAZY)
    private Collection<CoursEnseignant> coursEnseignants;
    @OneToMany(mappedBy = "enseignant", fetch = FetchType.LAZY)
    private Collection<Jury> juries;
    @OneToMany(mappedBy = "enseignant", fetch = FetchType.LAZY)
    private Collection<RolePersonne> rolePersonnes;
    @OneToOne(mappedBy = "enseignant", fetch = FetchType.EAGER)
    private Departement departement;
    public Enseignant() {
        super();
    }
}
```

### L'implémentation de l'entité Etudiant :

```
@Entity
@DiscriminatorValue("ETUDIANT")
public class Etudiant extends Personne{
    @OneToMany(mappedBy = "etudiant", fetch = FetchType.LAZY)
    private Collection<ClasseEtudiant> classeEtudiants;
    @OneToMany(mappedBy = "etudiant", fetch = FetchType.LAZY)
    private Collection<DocumentEtudiant> documentEtudiants;
    @OneToMany(mappedBy = "etudiant", fetch = FetchType.LAZY)
    private Collection<EtudesFaites> etudesFaites;
    @OneToMany(mappedBy = "etudiant", fetch = FetchType.LAZY)
    private Collection<Parents> parents;
    @OneToMany(mappedBy = "etudiant", fetch = FetchType.LAZY)
    private Collection<Note> notes;
    public Etudiant() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

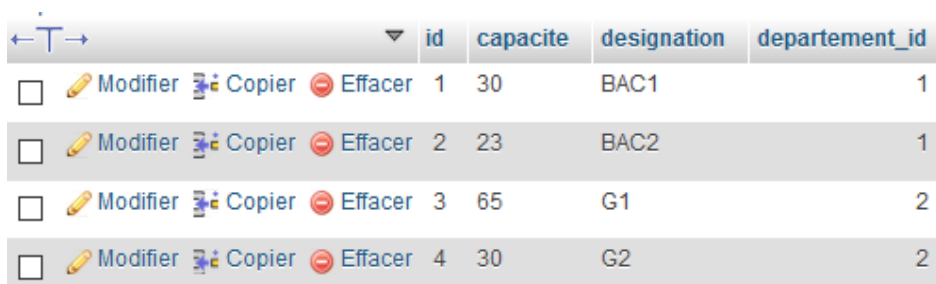
**N.B:** La présentation des entités Enseignant et Etudiant c'est par la Super Classe Personne.

### L'implémentation de l'entité Classe :

@Entity

```
public class Classe implements Serializable {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    @Column(length = 50) @NotEmpty @Size(min = 2, max = 50)  
    private String designation;  
    @Column(length = 10) @NotEmpty @Size(min = 2, max = 10)  
    private String capacite;  
    @OneToMany(mappedBy = "classe", fetch = FetchType.LAZY)  
    private Collection<CoursClasse> coursClasses;  
    @OneToMany(mappedBy = "classe", fetch = FetchType.LAZY)  
    private Collection<ClasseEtudiant> classeEtudiants;  
    @OneToMany(mappedBy = "classe", fetch = FetchType.LAZY)  
    private Collection<Jury> juries;  
    @ManyToOne  
    private Departement departement;  
    public Classe() {  
    super();  
    // TODO Auto-generated constructor stub  
    }  
}
```

Sa présentation :



|                                                  | id | capacite | designation | departement_id |
|--------------------------------------------------|----|----------|-------------|----------------|
| <input type="checkbox"/> Modifier Copier Effacer | 1  | 30       | BAC1        | 1              |
| <input type="checkbox"/> Modifier Copier Effacer | 2  | 23       | BAC2        | 1              |
| <input type="checkbox"/> Modifier Copier Effacer | 3  | 65       | G1          | 2              |
| <input type="checkbox"/> Modifier Copier Effacer | 4  | 30       | G2          | 2              |

Figure 4.4. Architecture de l'entité Classe

## L'implémentation de l'entité Document\_Etudiant :

@Entity

```
public class DocumentEtudiant implements Serializable{
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String designation;
    @ManyToOne
    private Etudiant etudiant;
    @Transient
    private byte[] file;
    public DocumentEtudiant() {
        super();
    }
}
```

Sa présentation :

|                                                                                                                                                                                                                                                                                      | id | designation                                           | etudiant_id |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------------------------------------------------|-------------|
| <input type="checkbox"/>    | 1  | Bulletin de la cinquième (5em) années des humanité... | 7           |
| <input type="checkbox"/>    | 2  | Bulletin de la sixième (6em) années des humanités     | 7           |
| <input type="checkbox"/>    | 3  | Diplôme d'Eta ou journal ou Attestation de réusit...  | 7           |
| <input type="checkbox"/>    | 4  | Attestation de bonne conduite, vie, mœurs et civis... | 7           |
| <input type="checkbox"/>    | 5  | Attestation de nationalité                            | 7           |
| <input type="checkbox"/>    | 6  | Aptitude physique                                     | 7           |
| <input type="checkbox"/>    | 7  | Carte d'identité                                      | 7           |
| <input type="checkbox"/>    | 8  | Attestation de naissance                              | 7           |
| <input type="checkbox"/>    | 9  | Photo passeport                                       | 7           |
| <input type="checkbox"/>    | 10 | Bulletin de la cinquième (5em) années des humanité... | 8           |
| <input type="checkbox"/>    | 11 | Bulletin de la sixième (6em) années des humanités     | 8           |
| <input type="checkbox"/>    | 12 | Diplôme d'Eta ou journal ou Attestation de réusit...  | 8           |
| <input type="checkbox"/>    | 13 | Attestation de bonne conduite, vie, mœurs et civis... | 8           |

Figure 4.6. Architecture de l'entité Document\_Etudiant

## **IV.5 Les outils utilisés**

### **IV.5.1. Bootstrap**

Bootstrap est une boîte à outils open source pour le développement avec HTML, CSS et JS, propose un prototypage rapide des idées, permet de créer une application entière, un système de grille réactif, de nombreux composants prédéfinis et des plugins puissants construits sur jQuery. [16]

Bootstrap est un Framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation...

Il nous a permis ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées en synthétisant celui de html et du css.

### **IV.5.2 WampServer**

WampServer est une plate-forme de développement web sous windows pour des applications web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL [17]. Il nous a permis la mise en place de notre base de données et de la manipulation de l'outil mathématique en l'occurrence l'algèbre relationnelle.

### **IV.5.3 Powerdesigner [18]**

PowerDesigner (Anciennement PowerAMC), le n°1 de la modélisation de données, est un outil tout-en-un de modélisation d'entreprise et de gestion des métadonnées destiné à documenter l'architecture d'entreprise. L'Architecture d'Entreprise permet de décrire l'organisation de manière dynamique afin de disposer d'analyses d'impact rapides et pertinentes, préalables à une conduite du changement axée sur l'amélioration de la Performance. PowerDesigner aligne les différents composants du système d'information : la modélisation de données, le développement d'applications et la mise en œuvre des infrastructures. Il nous a été utile dans la mise en place et à générer tous les diagrammes utilisés dans notre travail.

### **IV.5.4 Apache POI. [19]**

**POI** (Poor Obfuscation Implementation ou mise en œuvre d'une mauvaise obfuscation) est un projet de l'Apache Software Foundation permettant de manipuler avec la technologie Java divers types de fichiers créés par Microsoft Office. Au sein de cette fondation, il faisait auparavant partie du projet Jakarta. Apache POI est une bibliothèque Java open source fournie

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

par Apache, c'est la puissante bibliothèque pour vous aider à travailler avec des documents Microsoft tels que Word, Excel, Powerpoint, Visio...

POI nous a permis de travailler avec des fichiers de Microsoft Office notamment, générer les fichiers Excel par notre application. POI est constitué de plusieurs composants : Apache POI vous permet de travailler avec les formats de Microsoft, ses classes sont habituellement préfixés avec HSSF, XSSF, HPSF,...

#### **IV.5.5 iText [20]**

**iText** est une interface de programmation partiellement à code source ouvert servant à créer et manipuler des documents PDF. Écrit en langage Java, en .NET (iTextSharp) ainsi qu'en Java compatible avec Android (iTextG). Il est distribué sous licence AGPL et propriétaire pour certaines parties , iText permet par exemple de:

- Créer un fichier PDF à la volée et l'afficher dans un navigateur ;
- Créer des documents dynamiques à partir de sources telles que des fichiers XML ou des bases de données ;
- Ajouter ou supprimer de l'interactivité (par exemple, rendre un formulaire interactif non modifiable une fois qu'il est rempli) ;
- Ajouter des marque-pages, des numéros de pages, des filigranes ;
- Découper, concaténer et manipuler des pages de fichiers PDF ;
- Automatiser le remplissage de formulaires au format PDF. Par exemple pré-remplir le nom dans une facture ;
- Ajouter une signature numérique ;

Pour ce qui est de notre application, iText nous a aidé à générer les fichiers PDF avec notre application.

#### **IV.5.6. Le MySQL [21]**

MySQL s'est fait connaître à la fin des années 1990 comme étant un système de gestion de bases de données relationnelles de choix pour les petits projets web, profitant principalement de sa gratuité et de sa rapidité. Puis, dans le courant des années 2000, les géants du Web se sont tous mis à utiliser massivement MySQL. La suite a été plus difficile, puisque dans la seconde moitié des années 2000, tous ces grands acteurs ont été confrontés aux limitations de MySQL concernant sa scalabilité. C'est à ce moment qu'ont émergé de nombreuses solutions NoSQL. Mais finalement, MySQL a su évoluer rapidement au cours des dernières années, alors que les problèmes de jeunesse des solutions NoSQL sont devenus de plus en plus évidents.

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

Aujourd'hui, MySQL reste un choix très répandu pour les projets web, beaucoup moins pour les projets plus traditionnels. Tout d'abord, MySQL est capable d'offrir de bonnes performances même sur des serveurs peu puissants. De plus, sa stabilité est excellente et, sur une instance correctement configurée, il est très rare de voir MySQL planter ou de perdre des données. Enfin, sa gratuité permet à beaucoup d'utilisateurs de s'en approprier et d'envisager son déploiement et son utilisation. Pour ce qui est de notre cas, le serveur MySQL installé sur notre poste via WampServer comme nous utilisons le système d'exploitation windows, il nous a permis de créer la base de données qui nous a servi dans toute la suite de notre application : l'enregistrement des données de façon organisée et hiérarchisée dans notre base de données, la modification, la suppression et la mise à jour en utilisant les instructions SQL.

#### IV.6. Maquettage du système

Le maquettage reste une des étapes la plus importantes à définir afin d'avoir un aperçu visuel de notre application, nous sommes passés par un maquettage sur papier avant de pouvoir définir les figures suivantes (figures 4.13, 4.14, 4.15...4.28). Les maquettes suivantes représentent les pages d'accueil de notre système qui contiennent les principales fonctionnalités « Login », « Je m'inscrit », « Services ».

##### La page d'accueil

La figure suivante nous présente la page d'accueil de notre application.

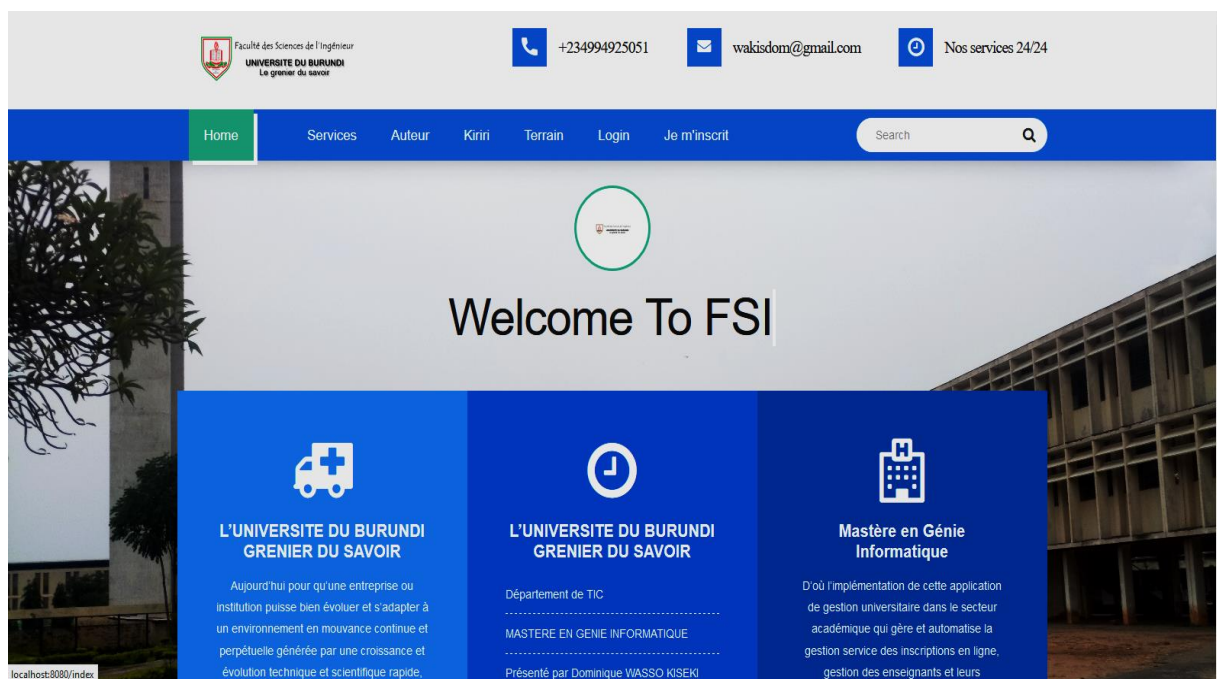


Figure 4.7. La page d'accueil

Cette page comprend cinq menus à savoir : Home, Services, Auteur, Kiriri, Terrain, Login, Je m'inscris. A ce point deux menu nous intéresse, il s'agit du **Login** et **Je m'inscris**.

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

- On clique sur **Login**, il s'agira de la fenêtre d'authentification qui se présente de la manière suivante :

### La page d'authentification

Cette figure nous présente la page d'authentification de notre application.



Figure 4.8. La page d'authentification

Il est à signaler ici qu'on peut se connecter avec les droits ci-après : Admin, Scolarité, Chef-Dep, Enseignant et Etudiant. Lorsqu'on se connecte comme d'administrateur on obtient la page suivante :

### L'Espace Admin

Comme prévu dans la modélisation, l'administrateur détient tous les droits de l'application. C'est pourquoi tous les menus lui sont réservés.

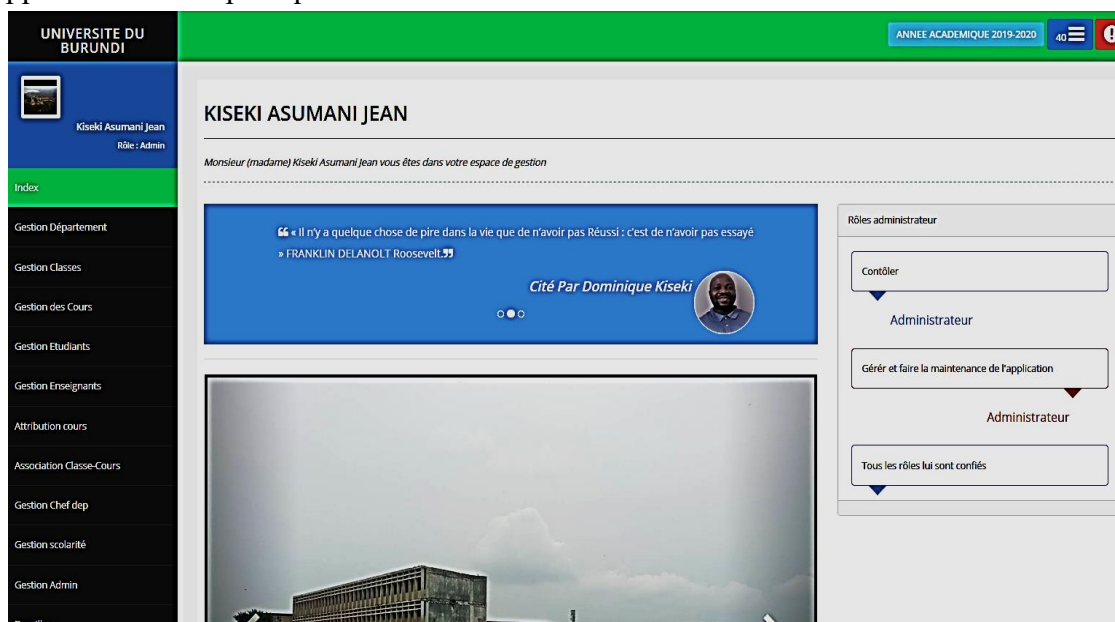


Figure 4.9. Les rôles de l'administrateur

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

## L'Espace Chef-Dep

La figure suivante nous présente l'espace du Chef de Département de notre application.

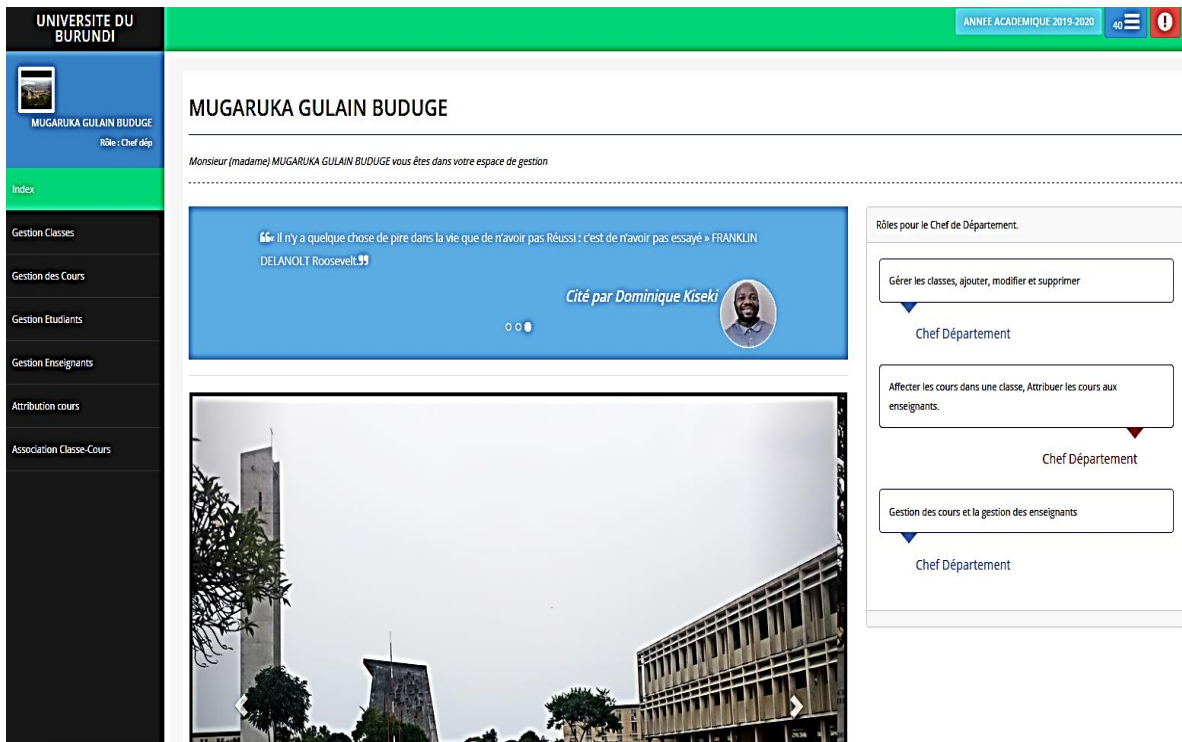


Figure 4.10. Les rôles du Chef-Dep

## L'espace Enseignant

La figure suivante nous présente l'espace de l'enseignant et ses rôles dans l'application.

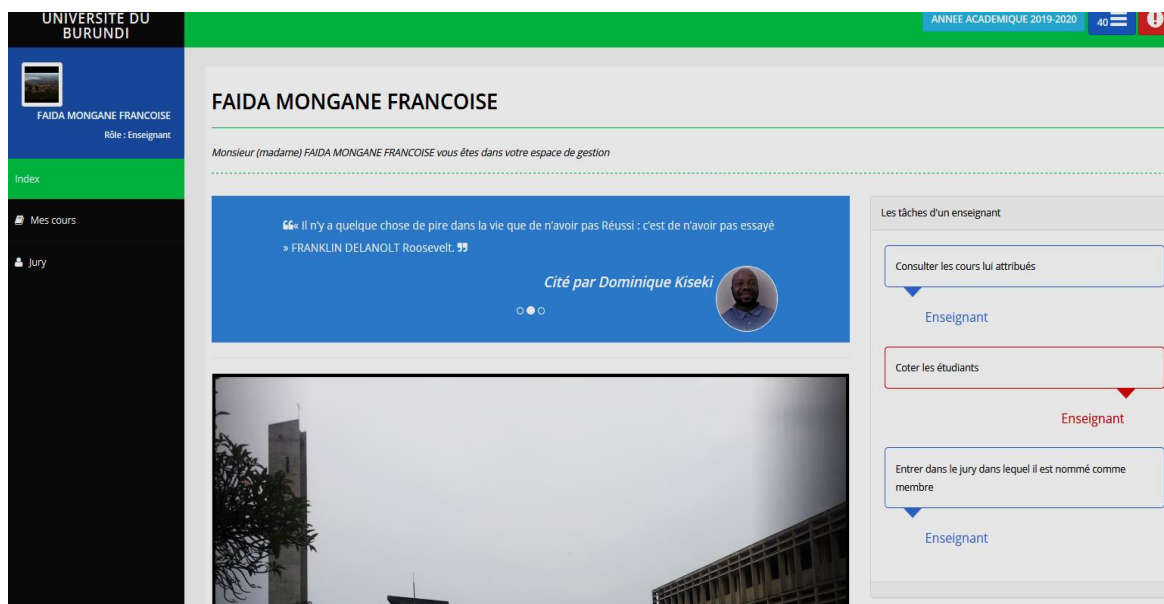


Figure 4.11. Les rôles ou Espace enseignant

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

## L'Espace Etudiant

La figure suivante nous présente l'espace Etudiant de l'application ainsi que ses rôles.

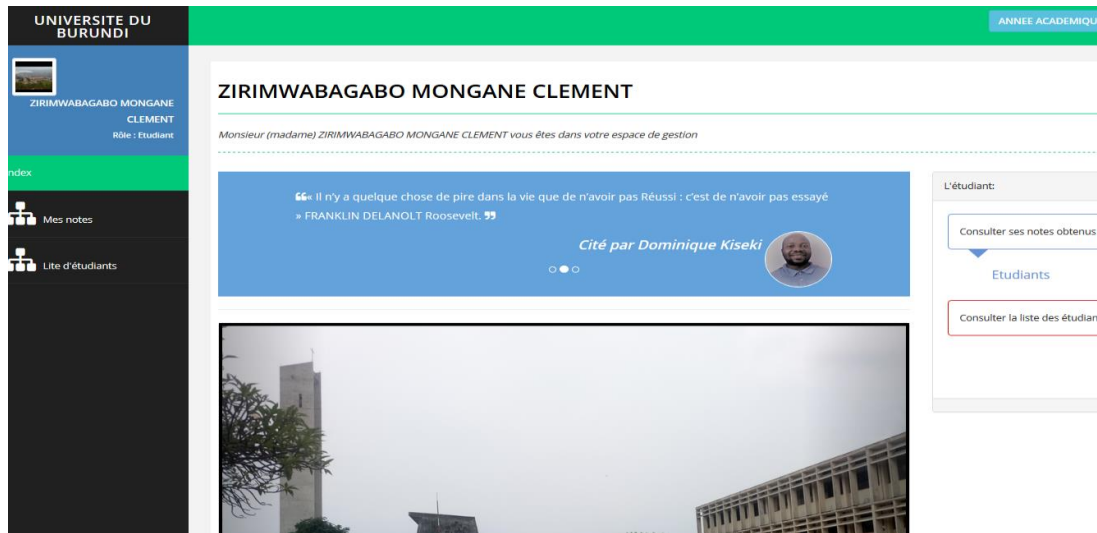


Figure 4.12. Les rôles d'un Etudiant

## Les deux Grilles délibération

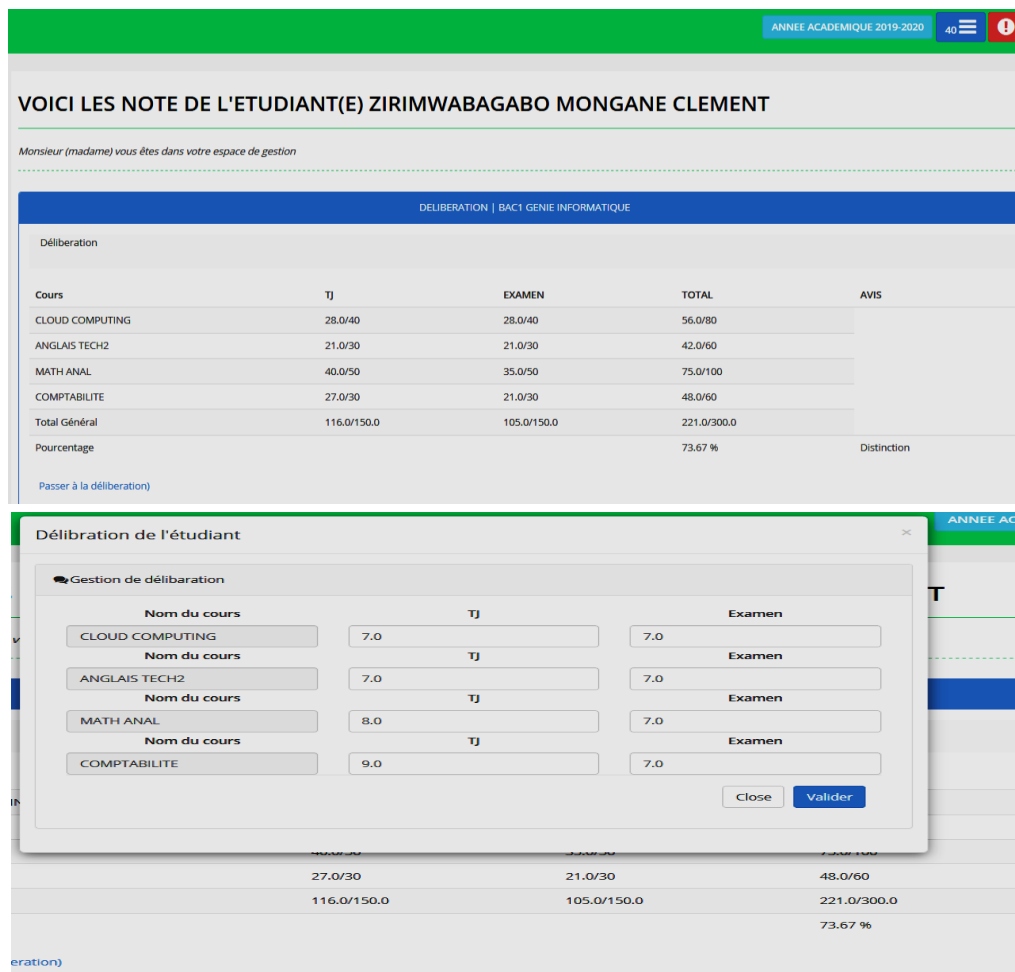


Figure 4.13. Les deux grilles de délibération

## Send cotes

Cette figure nous présente le formulaire de gestion et de l'envoi de notes par l'enseignant.

The screenshot shows a web interface for 'GESTION DES NOTES'. At the top, there is a green header with 'ANNEE ACADEMIQUE 2019-2020', a menu icon, and a notification icon. Below the header, the title 'GESTION DES NOTES' is displayed. A subtitle reads 'Ce formulaire vous accompagne dans l'envoi des notes des étudiants'. The main content area has a blue header 'GESTION DES NOTES' and contains a table with the following data:

| Nom de l'étudiant             | Matricule | Examen |
|-------------------------------|-----------|--------|
| ZIRIMWABAGABO MONGANE CLEMENT | TJ        | Examen |
| KASINDI LUGANZU FRANCOISE     | TJ        | Examen |

Below the table, a light blue message box states: 'Ces cotes ont déjà été envoyées'.

Figure 4.14. L'envoi de cotes par l'enseignant

- La deuxième clique sur Je m'inscris nous amène sur l'inscription en ligne dont voici les différentes étapes

## Inscription en ligne Etape1 : identité de l'étudiant

Cette figure nous plonge sur le formulaire débutant les inscriptions en ligne.

The screenshot shows the 'IDENTITE DE L'ETUDIANT(ETAPE 1 SUR 7)' registration form. At the top, there is a navigation bar with 'Home', 'Services', 'Auteur', 'Kiriri', 'Terrain', 'Login', and 'Je m'inscris'. A search bar is also present. The form fields are as follows:

- Nom de l'étudiant:
- Prénom de l'étudiant:
- Lieu de naissance:
- Adresse:
- E-mail:
- postnom de l'étudiant:
- Sexe:
- date de naissance:
- Téléphone:
- Choisir votre mot de passe:

A 'Suivant' button is located at the bottom right of the form.

Figure 4.15. L'identité de l'étudiant

## Etape 2 : Identité des parents

The screenshot shows the 'IDENTITE DES PARENTS (ETAPE 2 SUR 7)' form. The header includes the university logo and contact information: +234994925051, wakisdom@gmail.com, and 'Nos services 24/24'. The navigation bar has 'Home', 'Services', 'Auteur', 'Kiriri', 'Terrain', 'Login', and 'Je m'inscris'. A search bar is on the right. The form fields are: 'Nom du père' (LUGANZU BAGAZWA), 'Province d'origine' (KASAI CENTRALE), 'Nationalité' (CONGOLAISE), 'Téléphone' (03654496556), 'Nom de la mère' (LEONTINE KABESHA), 'Territoire d'origine' (MASISI), and 'Adresse' (KATANA CENTRE). 'Précédent' and 'Suivant' buttons are at the bottom.

Figure 4.16. Le candidat complète l'identité de ses parents

## Etape 3 : Etudes faites par le candidat aux humanités

The screenshot shows the 'ETUDES FAITES (ETAPE 3 SUR 7)' form. The header and navigation bar are identical to Figure 4.16. The form fields are: 'Section' (PEDAGOGIQUE), 'Année obtention diplôme' (2006-2007), 'Ecole de provenance' (LYCEE KATANA), 'Option' (PEDAGOGIE GENERALE), and 'Pourcentage obtenu' (60.0). 'Précédent' and 'Suivant' buttons are at the bottom.

Figure 4.17. Etudes faites du candidat

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

#### Etape4 : Ici le candidat envoi ou présente les différents documents exigés.

Faculté des Sciences de l'ingénieur  
UNIVERSITE DU BURUNDI  
Le grenier du savoir

+234994925051 wakisdome@gmail.com Nos services 24/24

Home Services Auteur Kiriri Terrain Login Je m'inscrit Search

### 4. DOSSIERS DE L'ETUDIANT (ETAPE 4 SUR 6)

Bulletin de la cinquième (5em) années des humanités Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Bulletin de la sixième (6em) années des humanités Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Diplôme d'Etat ou journal ou Attestation de réussite Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Attestation de bonne conduite, vie, mœurs et civisme Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Attestation de nationalité Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Aptitude physique Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Carte d'identité Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Attestation de naissance Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Photo passeport Déjà envoyé

Parcourir... Aucun fichier sélectionné. Envoyer

Précédent Suivant

Figure 4.18. Documents du candidat.

#### Etape5 : Le choix de la promotion par le candidat

Faculté des Sciences de l'ingénieur  
UNIVERSITE DU BURUNDI  
Le grenier du savoir

+234994925051 wakisdome@gmail.com Nos services 24/24

Home Services Auteur Kiriri Terrain Login Je m'inscrit Search

### 5. CHOISIR LA PROMOTION ET L'OPTION (ETAPE 5 SUR 7)

Sélectionner votre promotion

BAC1 GENIE INFORMATIQUE

Précédent Suivant

Faculté des Sciences de l'ingénieur  
UNIVERSITE DU BURUNDI  
Le grenier du savoir

Le grenier du savoir.

**CONTACT US**  
wakisdome@gmail.com  
(+257) 69916762  
(+243) 994925051

**SUGGESTION**  
Envoyez-nous vos suggestions

Email address

Figure 4.19. Choix de la promotion

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

**Etape 6 : Fin de l'inscription.** Ici il s'agit de confirmer son inscription en cliquant sur je m'inscrit.

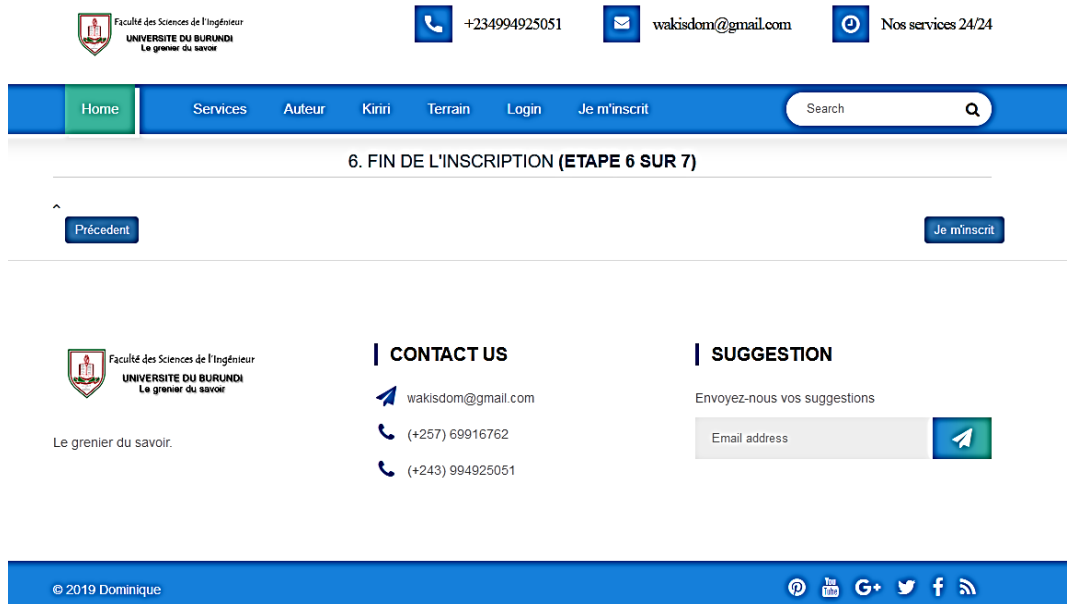


Figure 4.20. Confirmation de son inscription

**Etape7 :** Message de confirmation au candidat en lui offrant un login et le mot de passe pour le futur usage.

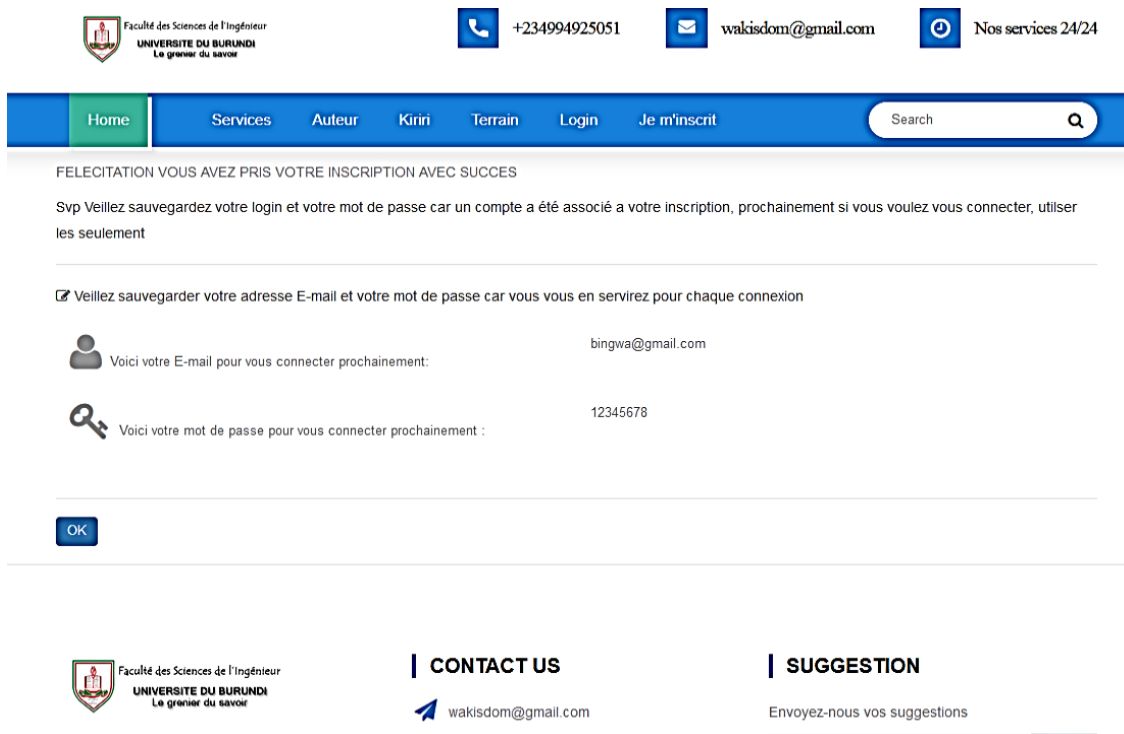


Figure 4.21. Réception du mot de passe et du login

## Brouillon

Cette figure nous présente le brouillon de l'application. Il est à signaler que seul l'administrateur a accès au brouillon pour restaurer un élément supprimer par l'utilisateur.

| Id | Nom     | PostNom | Prenom  | Sexe  | Lieu de Naissance | Date de Naissance     | Matricule      | Adresse       | Téléphone | @Mail            |           |
|----|---------|---------|---------|-------|-------------------|-----------------------|----------------|---------------|-----------|------------------|-----------|
| 7  | KASINDI | MONGANE | CLEMENT | Homme | KABARE            | 2019-12-30 00:00:00.0 | EKJIED41457744 | NJILI PLATEAU | 03654496  | bingwa@gmail.com | Restaurer |

Figure 4.22. Brouillon

## IV.7. Conclusion

A ce chapitre, notre projet d'études atteint sa fin. Tout au long de ce chapitre, nous avons abordé les technologies utilisées ainsi que notre environnement de travail en passant par le langage de programmation orienté objet utilisé (Java), le framework et les différents outils utilisés. Nous avons constaté ce qui suit : aucun outil n'est parfait, mais, vous l'aurez compris, Spring Boot a clairement notre préférence dans le monde des frameworks web modernes. Il propose des réponses simples et efficaces à une très grande majorité des problématiques auxquelles les développeurs d'applications modernes sont confrontés. Aussi, nous avons recourus aux outils tels que : Bootstrap, Power designer, iText, Apache POI et Wamp-server comme serveur Apache.

Par la suite, nous avons expliqué notre architecture d'application à travers le maquettage afin de présenter finalement les différentes principales parties d'implémentation de notre application réalisée.

## CONCLUSION GENERALE ET RECOMMANDATIONS

### 1. Conclusion générale

Les technologies de l'information et de la communication connaissent une croissance vertigineuse pratiquement partout dans le monde et surtout ces dernières années grâce au développement de l'internet. Ce projet de fin d'études avait pour ambition d'établir un système de gestion universitaire en ligne, plus précisément dans la gestion des inscriptions, enseignants et enseignement, gestion de côtes dans le secteur académique, la conception d'une base de données et d'un site web dynamique que nous venons de réaliser au cours de notre recherche.

Il est important de noter aujourd'hui, que pour concevoir une application, une base des données surtout dans le service académique, la conception d'un site web reste inévitable pour permettre aux utilisateurs et aux désireux de consulter à distance ou soit en ligne les publications des listes des étudiants inscrits pour chaque année académique, aux enseignants de voir et consulter les listes des cours, des classes, les enseignants prestataires, les résultats par session obtenus par chaque étudiant produits par cet outil qui est le fruit de notre recherche.

Pour aboutir à des résultats escomptés, nous avons fait recours à des méthodes analytique, descriptive et la méthode UML, les techniques d'observation, d'interview et la technique documentaire nous ont servis pour la mise en œuvre de nos recherches en passant par la vérification avec l'algèbre relationnelle comme outil mathématique utilisé.

Partant de ce qui précède, nous avons vérifié à travers nos hypothèses que l'objectif principal de ce projet de recherche scientifique qui consistait à la mise en place d'une application informatique de gestion universitaire dans le secteur académique est observable par l'effectivité de cette application produite. Avec notre système, on arrive à archiver les documents et les données, télécharger et imprimer automatiquement les informations nécessaires, elle permet aux gestionnaires et agents des services concernés dans la prise de décision quant à l'évolution des enseignements, avoir une traçabilité sur le parcours d'un utilisateur dans un établissement. Elle permet également de connaître le nombre d'enseignants utilisés dans un département et leurs charges horaires. Ensuite, elle facilite les inscriptions et autres services académiques en ligne.

Mise en place d'un système de gestion universitaire des enseignants, enseignements et étudiants.

Eu égard de ce qui précède, sans doute nous confirmons que nos objectifs assignés ont été atteints à 100%. Nous reconnaissons que ce travail nous a été très formateur, puisqu'il a permis de découvrir une nouvelle technologie innovante tel que le framework spring-boot et des outils modernes tels que Powerdesigner, iText, Apache POI... et nous a permis également de se confronter à plusieurs contraintes à la fois : contraintes de temps, contraintes d'expérience et de technologie. En outre, ce projet nous a permis d'approfondir nos connaissances dans les bonnes pratiques de l'ingénierie en génie logiciel.

## **2. Recommandations**

Au regard des résultats obtenus, nous recommandons aux institutions d'enseignements supérieurs et universitaires d'automatiser leurs systèmes de gestion en ce qui concerne notamment, la gestion des enseignements, des programmations des enseignants ainsi que le suivi des dossiers (notes obtenues et autres) des étudiants.

En outre, d'organiser les séances de formation des personnels et utilisateurs de cette application informatique afin de minimiser les erreurs et de rendre plus rapide et facile la gestion du secteur académique.

En fin, nous faisons appel aux autres chercheurs à pouvoir optimiser le fonctionnement de cette application afin de l'améliorer et de la rendre plus performante. C'est seulement de cette manière que cette étude pourra s'inscrire dans la série des travaux déjà existants dans ce domaine.

## Références bibliographiques et webographiques

1. BRANDLIN & HARRISSON, *Contraintes du marché et négociations locales*, Ed. l'Harmattan, Paris, 1987.
2. B. DAVIS GOURDON & MARGUERITE H. OLSON, *Système d'information pour le Management*, Ed. Economica, Hericart, Paris, 1990.
3. H. TARDIEU, A. ROCHEFELD & R. COLETTE, *La Méthode MERISE, Principes et outils*, Ed. d'Organisation, Paris, 1998.
4. Encyclopédie Universalis, 200 1-2002.
5. <http://www-inf.int-evry.fr/cours/CSC4002/EnLigne/Cours/CoursUML/5.16.23.html>, consulté à Bukavu, le 15 décembre 2019.
6. Abdellah Madani, « *UML : Unified Modeling Language* », Support de cours, Université Nancy 2, 2007.
7. Pascal ROQUES, *UML par la pratique*. EYROLLES, Paris 2009. 8 éditions
8. [www.joradp.dz](http://www.joradp.dz), consulté à Bujumbura, Samedi 1<sup>er</sup> Juin 2019 à 6h25
9. P. Roques et F. Vallée, *UML 2 en action de l'analyse des besoins à la conception*, Edition EYROLLES, 2005.
10. [http://docwiki.embarcadero.com/RADStudio/Rio/fr/D%C3%A9finition\\_des\\_diagrammes\\_d%27activit%C3%A9s\\_UML\\_2.0](http://docwiki.embarcadero.com/RADStudio/Rio/fr/D%C3%A9finition_des_diagrammes_d%27activit%C3%A9s_UML_2.0). Consulté à Bujumbura, Dimanche 2 juin 2019.
11. <https://www.lucidchart.com/pages/fr/tutoriel-sur-les-diagrammes-de-classes>. Consulté à Bukavu, le 15 février 2020 à 10h<sup>00</sup>.
12. C. Soutou. *Programmer avec MySQL* (3e édition). N°13719, 2013, 520 pages.
13. (<https://www.ionos.fr/digitalguide/sites-internet/developpement-web/tutoriel-spring-boot/>). Consulté à Bukavu, le 20 janvier 2020 à 12h45.
14. <http://centonni.com/strategies-heritage-jpa/> consulté à Bukavu, le 6 décembre 2019.
15. <http://www.ocamil.com/index.php/spring/spring-mvc>. Consulté à Bukavu, le 20 janvier 2020 à 13h25.
16. [https://www.w3schools.com/bootstrap4/bootstrap\\_get\\_started.asp](https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp). Consulté à Bukavu, le 20 janvier 2020 à 16h45.
17. <http://www.lepetitatelier.be/content/presentation-sommaire-de-wampserver>. Consulté à Bukavu, le 25 mars 2020 à 9h<sup>00</sup>.
18. <https://www.powerdesigner.biz/FR/powerdesigner/powerdesigner-features.html>. Consulté à Bukavu, le 25 mars 2020 à 10h<sup>00</sup>.
19. <https://poi.apache.org/>. Consulté à Bukavu, le 25 mars 2020 à 12h<sup>00</sup>.
20. <https://itextpdf.com/en/resources/api-documentation/itext-5-java>. Consulté à Bukavu, le 26 mars 2020 à 9h<sup>00</sup>.
21. <https://dev.mysql.com/doc/refman/8.0/en/>. Consulté à Bukavu, le 27 mars 2020 à 8h<sup>00</sup>.