UNIVERSITÉ DU BURUNDI
GRENIER DU SAVOIR DU BURUNDI
Dépôt institutionnel officiel
https://repository.ub.edu.bi

Mise en place d’un système automatisé d’archivage des documents administratifs et académiques entrants d’une faculté : Cas de la FSI

Niyinkunda, Anne Marie; Sous la direction de : Dr. Mukeshimana Michèle

2023
UB, FSI
https://repository.ub.edu.bi/handle/123456789/1677

Document téléchargé depuis le **Grenier du Savoir du Burundi** — https://repository.ub.edu.bi

République du Burundi
Ministère de l'Education Nationale et de la Recherche Scientifique

Université du Burundi



Master en Génie Informatique

Année académique 2022-2023

Mise en place d'un système automatisé d'archivage des documents administratifs et académiques entrants d'une faculté : Cas de la FSI

MEMOIRE

Présenté à la

Faculté des Sciences de l'Ingénieur (FSI)

En vue de l'obtention du

MASTER

en

Génie Informatique

Par

NIYINKUNDA Anne Marie

Soutenu le 04/12/2023devant le jury composé de :

Pr.	NDIKUMAGENGE	Jérémie	Président
Dr.	NDAYISABA	Longin	Secrétaire
Dr.	MUKESHIMANA	Michèle	Directeur
Dr.	SINZOBAKWIRA	Issa	Membre

BUJUMBURA, décembre 2023

Identification des membres du jury

Dr. MUKESHIMANA Michèle	: Directeur
Pr. NDIKUMAGENGE Jérémie	: Président
Dr. NDAYISABA Longin	: Secrétaire
Dr. SINZOBAKWIRA Issa	: Membre

Dédicaces

A Dieu tout puissant ;

A mon Père ;

A ma regrettée Mère qui nous a quittés sitôt,

A ma très chère Marâtre ;

A mes oncles;

A mes frères et sœurs ;

A mes cousins et cousines ;

A tous mes camarades de classe ;

A tous mes amis et connaissances.

Remerciements

J'adresse ma plus profonde gratitude à Dieu le Tout Puissant, le très Miséricordieux pour m'avoir accordé la vie et le courage de réaliser ce travail jusqu'au bout.

Mes remerciements sont adressés à ma famille qui a été toujours à mes côtés en sacrifiant tout pour moi. Elle m'a montré un modèle de labeur et de persévérance à suivre, et je suis redevable d'une très bonne éducation dont je suis fière. Qu'ils trouvent ici l'expression de ma profonde gratitude.

Je remercie plus spécialement Dr Michèle MUKESHIMANA, mon directeur de mémoire, qui malgré ses multiples responsabilités, a accepté sans hésiter de diriger ce travail. Son expérience, sa rigueur scientifique, ses conseils judicieux, ses critiques et suggestions pertinentes m'ont été très utiles et très bénéfiques dans la réalisation de ce travail.

Je tiens également à remercier tout le personnel du décanat de la FSI qui a accepté de prêter une oreille attentive à mes différentes préoccupations. Je les remercie pour leur implication et leur grand apport dans ce projet

Mes remerciements les plus vifs s'adressent aussi aux membres du jury qui ont accepté de lire et d'évaluer ce mémoire. J'adresse ma gratitude aux enseignants qui ont contribué à ma formation ainsi qu'à mes collègues et au personnel du service administratif de l'université du Burundi pour l'accueil chaleureux qu'ils m'ont réservé lors de mes recherches.

J'adresse ma gratitude au corps professoral de la Faculté des Sciences de l'Ingénieur plus particulièrement à tous les professeurs qui nous ont enseigné dans le Département des Technologies de l'Information et de la Communication, option Génie Informatique ainsi qu'à tous ceux qui ont contribué moralement et financièrement à notre formation durant les deux années d'études du deuxième cycle à l'Université du Burundi. Mes remerciements s'adressent également à toute personne qui, de près ou de loin, a contribué activement dans la réalisation de ce travail.

Résumé

L'archivage des documents administratifs et académiques est un service très sollicité et d'une grande importance dans les activités quotidiennes des facultés et instituts. Les documents archivés doivent être gardés de façon permanente et fiable car ils sont souvent sollicités dans les situations importantes de la vie. A la FSI cet archivage est manuel et comporte des anomalies diverses à savoir, la recherche très difficile des documents ciblés, manque de référence stable des index des documents, usure des documents etc.

Ce mémoire propose une solution basée sur une analyse approfondie portant sur la conception et la mise en œuvre d'un système automatisé d'archivage des documents entrants, tant administratifs qu'académiques, au sein de la Faculté des Sciences de l'Ingénieur de l'Université du Burundi. L'objectif principal est d'améliorer l'archivage des documents, l'efficacité opérationnelle créant un environnement de travail numérique efficace.

La méthodologie de recherche débute par une évaluation minutieuse des besoins spécifiques de la Faculté des Sciences de l'Ingénieur, suivie d'une étude des technologies et normes pertinentes, notamment les normes de gestion électronique des documents et l'utilisation d'UML pour modéliser les processus. J'ai développé ensuite un système automatisé intégrant des fonctionnalités de consultation, de classement et de recherche, tout en garantissant la sécurité et la préservation à long terme des documents.

Ce système offre des avantages majeurs tels que la réduction significative des coûts liés à la gestion documentaire, une meilleure utilisation de l'espace physique, une recherche et une récupération d'informations plus efficaces, ainsi qu'une réponse plus rapide aux demandes académiques et administratives spécifiques à la Faculté des Sciences de l'Ingénieur.

Mots-clés : archivage, documents entrants, classement, gestion électronique des documents (GED).

Abstract

The archiving of administrative and academic documents is a highly requested service and of great importance in the daily activities of faculties and institutes. Archived documents must be kept permanently and reliably because they are often called upon in important life situations. At the FSI, this archiving is manual and involves various anomalies, namely the very difficult search for targeted documents, lack of stable references to document indexes, wear and tear on documents, etc.

This dissertation proposes a solution based on in-depth analysis relating to the design and implementation of an automated archiving management system for incoming documents, both administrative and academic, within the Faculty of Engineering Sciences of the University of Burundi. The main objective is to optimize document management and improve operational efficiency creating an effective digital working environment.

The research methodology begins with a careful assessment of the specific needs of the Faculty of Engineering Sciences, followed by a study of relevant technologies and standards, in particular electronic document management standards and the use of UML to model processes. I then developed an automated system integrating consultation, classification and search functionalities, while guaranteeing the security and long-term preservation of the documents.

This system offers major benefits such as significant reduction in document management costs, better utilization of physical space, more efficient search and retrieval of information, and faster response to academic and administrative procedures specific to the Faculty of Engineering Sciences.

Keywords: Archiving, Incoming documents, Filing, Electronic document management (EDM).

Table des matières

Identification des membres du jury i
Dédicaces ii
Remerciements iii
Résumé iv
Abstract v
Table des matières vi
Liste des tableaux ix
Liste des figures x
Liste des sigles et abréviations xi
Avant-Propos xiii
CHAPITRE I. INTRODUCTION GENERALE 1
I.1. Justification du contexte 1
I.2. Objectifs 2
I.2.1. Objectif global 2
I.2.2. Objectifs spécifiques 2
I.3. Problématique 2
I.4. Solutions proposées 3
I.5. Résultats attendus 3
I.6. Apports scientifiques et technologiques 3
I.7. Outils mathématiques et plateforme de conception 4
I.8. Méthodologie utilisée 4
I.9. Conclusion 4
CHAPITRE II. PRESENTATION GENERALE DU CAS D'ETUDE 5
II.1. Présentation de l'Université du Burundi 5
II.2.Présentation de la Faculté des sciences de l'ingénieur 6
II.2.1. Historique 6
II.2.2. Offre de la formation 7
II.3. Organisation et fonctionnement de la FSI 9

II.3.1. Décanat 9
II.3.2. Secrétariat 9
II.3.3. Conseil de la faculté 10
II.3.4. Jury de délibération 11
II.4. Conclusion 11
CHAPITRE III. MODELISATION DU SYSTEME AVEC LE LANGAGE UML 12
III.1. Langage UML 12
III.1.1. Définition 12
III.1. 2. Historique 12
III.1.3. Motivation du choix du langage UML comme outil de modélisation 14
III.2. Diagrammes du langage UML 15
III.2.1. Diagramme des classes 16
III.2.2. Diagramme des cas d'utilisation 17
III.2.3. Diagramme des séquences système pour chaque cas d'utilisation 21
III.2.4. Diagramme de déploiement 30
III.2.5. Diagramme d'activité 31
III.3. Conclusion 35
CHAPITRE IV. MODELISATION MATHEMATIQUE AVEC MODELE RELATIONEL 36
IV.1. Modèle relationnel 36
IV.1.1. Définition 36
IV.1.2. Justification du choix du modèle 37
IV.2. Opérateurs de l'algèbre relationnelle 37
IV.2.1. Opérateurs unaires 37
IV.2.2. Opérateurs n-aires 40
IV.3. Fonctions d'agrégation 48
IV.4. Conception de nos requêtes en Algèbre Relationnelle 49
IV.5. Conclusion 50
CHAPITRE V. REALISATION DE L'APPLICATION 51
V.1.Présentation des outils de développement 51
V.2. Sécurité de l'application 56
V.2.1. Introduction 56
V.2.2. Importance de la sécurité des données 56

V.2.3. Protection de la base de données 57
V.3. Présentation de l'application 59
CONCLUSION GENERALE ET RECOMMANDATIONS 71
Conclusion générale 71
Recommandations 72
REFERENCES BIBLIOGRAPHIQUES 73

Liste des tableaux

Tableau 1: Priorisation des cas d'utilisation 21
Tableau 2: Diagramme de cas d'utilisation authentification..... 22
Tableau 3: Diagramme du cas d'utilisation ajouter un document..... 24
Tableau 4: Diagramme du cas d'utilisation consulter un document 25
Tableau 5: Diagramme de séquence du cas d'utilisation modifier un document 27
Tableau 6: Diagramme de séquence du cas d'utilisation télécharger un document 29
Tableau 7: Tableau de formalisme des éléments du diagramme d'activité 32
Tableau 8: Diagramme d'activité pour s'authentifier..... 34

Liste des figures

Figure 1: Diagramme UML 16
Figure 2: Diagramme de classe de l'application 17
Figure 3: Diagramme de cas d'utilisation global de l'application 20
Figure 4: Diagramme de séquence du cas d'utilisation authentification 21
Figure 5: Diagramme de séquence du Cas d'utilisation : « ajouter un document » 23
Figure 6: Diagramme de séquence du Cas d'utilisation : « consulter un document » 25
Figure 7:Diagramme de séquence du Cas d'utilisation : « modifier un document » 27
Figure 8: Diagramme de séquence du cas d'utilisation « télécharger un document » 28
Figure 9: Diagramme de séquence pour la gestion des utilisateurs 30
Figure 10:Diagramme de déploiement 31
Figure 11: Diagramme d'activité de notre application 33
Figure 12: Diagramme d'activité du cas d'utilisation d'authentification 34
Figure 13: identification d'authentification 60
Figure 14: interface d'accueil 62
Figure 15: interface pour ajouter un document 63
Figure 16: interface pour ajouter une lettre 64
Figure 17: interface pour toutes les lettres archivées 65
Figure 18: Interface des documents entrants archivés par le Doyen de la FSI 66
Figure 19: Interface des documents archivés par le secrétaire administratif 67
Figure 20: Interface des documents archivés par le secrétaire simple 68
Figure 21: Interface de gestion des utilisateurs 69

Liste des sigles et abréviations

A&U : Aménagement et Urbanisme

BMD : Baccalauréat Master Doctorat

CRUD : Create Read Update Delete

CSS : Cascading Style Sheets

EM : Electromécanique

FSA : Faculté des Sciences Appliquées

FSI : Faculté des Sciences de l'Ingénieur

GC : Génie Civil

GED : Gestion Electronique des Documents

HTML : HyperText Markup Language

HTTP : HyperText Transfer Protocol

I.T.S. : Institut Technique Supérieur

ISO : International Organization for Standardization

JSON : JavaScript Object Notation,

LMD : Licence Master Doctorat

MOF : Fonction Méta Objet

MVC : Modèle-Vue –Contrôleur

OCL : Langage de Contrainte d'Objet

OMG : Object Management Group

OMT : Object Modeling Technique

OOSE : Object-Oriented Software Engineering method

PHP : Hypertext Preprocessor

PNUD : Programme des Nations Unies pour le développement

POO : Programmation Orientée objet

SGBDR : Système de Gestion de Base de Données Relationnel

SQL : Structured Query Language

UML : Unified Modeling Langage

UNESCO : United Nations Educational, Scientific and Cultural Organization.

XLS : Excel Spreadsheet

XMI : Metadata Interchange

XML : Extensible Markup Language

Avant-Propos

L'évolution rapide des technologies de l'information et de la communication a profondément transformé les méthodes de gestion documentaire dans les institutions modernes. Dans un contexte marqué par une augmentation significative des flux d'informations et une exigence accrue de transparence et d'efficacité, la gestion des documents administratifs et académiques constitue un enjeu stratégique majeur pour toute organisation.

Ce mémoire s'inscrit dans cette dynamique en proposant une solution innovante : la mise en place d'un système automatisé d'archivage des documents entrants au sein de la Faculté des Sciences de l'Informatique (FSI). L'objectif principal de ce projet est de répondre aux défis posés par la gestion manuelle des archives, notamment en matière de perte de temps, de risques de perte d'information, et de difficultés d'accès aux documents.

En combinant rigueur académique et méthodologie pratique, ce travail ambitionne de démontrer l'importance et les bénéfices de l'automatisation pour une gestion documentaire efficace et durable. Il met également en lumière les étapes clés de la conception, de la mise en œuvre, et de l'évaluation d'un tel système, tout en tenant compte des spécificités organisationnelles et technologiques de la FSI.

Je tiens à exprimer ma profonde gratitude envers toutes les personnes et institutions qui ont contribué à la réalisation de ce mémoire. Leur soutien, leurs conseils, et leurs encouragements ont été une source précieuse de motivation tout au long de cette aventure académique.

Ce document, au-delà de ses apports techniques, se veut aussi une contribution à l'amélioration des pratiques de gestion documentaire dans les établissements d'enseignement supérieur.

Je souhaite que cet avant-projet puisse inspirer d'autres initiatives et contribuer à la modernisation des processus administratifs et académiques au sein de la FSI.

CHAPITRE I. INTRODUCTION GENERALE

I.1. Justification du contexte

Dans le contexte actuel de l'essor des technologies de l'information et de la numérisation des processus, l'archivage des documents administratifs et académiques est devenu une préoccupation majeure pour les établissements universitaires. La faculté, en tant qu'institution responsable de la gestion de nombreux documents essentiels, tels que les dossiers d'inscription, les relevés de notes, les demandes de congé, etc., est confrontée à un défi constant pour organiser, sécuriser et rendre ces informations accessibles aux parties concernées. Cependant, les méthodes traditionnelles d'archivage des documents, qui reposent souvent sur des processus manuels et des systèmes d'archivage physiques, sont souvent inefficaces, sujets aux erreurs et difficiles d'accès. C'est pourquoi la mise en place d'un système automatisé d'archivage des documents entrants administratifs et académiques est devenue une nécessité pour les facultés universitaires.

L'objectif de ce système automatisé est de simplifier et d'optimiser les processus de gestion documentaire en utilisant des outils technologiques avancés tels que des logiciels de gestion de documents et des bases de données. En permettant la numérisation, l'indexation et le stockage sécurisé des documents, ce système offre plusieurs avantages essentiels. Il réduit considérablement les tâches administratives manuelles, minimisant ainsi les erreurs et les pertes de documents. De plus, il améliore l'efficacité des processus administratifs et académiques en offrant un accès rapide et facile aux archives, permettant ainsi aux parties concernées de trouver et de consulter les documents pertinents en un temps record.

Dans cette étude, nous explorerons les différentes étapes nécessaires à la mise en place d'un système automatisé d'archivage des documents entrants administratifs et académiques de la faculté. Nous discuterons des défis potentiels liés à cette transition vers un environnement numérique, ainsi que des avantages attendus en termes d'efficacité, de sécurité et d'accessibilité. Nous examinerons également les solutions technologiques disponibles et les meilleures pratiques en matière de gestion documentaire dans un contexte universitaire.

En somme, la mise en place d'un système automatisé d'archivage des documents entrants administratifs et académiques représente une étape cruciale pour améliorer les processus d'archivage des documents dans les facultés universitaires. Cette étude vise à explorer les possibilités offertes par ces systèmes, en mettant l'accent sur les avantages qu'ils offrent en termes d'efficacité, de sécurité et d'accessibilité des documents essentiels à la faculté.

I.2. Objectifs

I.2.1. Objectif global

L'objectif global de notre sujet est de proposer et de mettre en œuvre un système informatisé qui permettra à une faculté/ institut d'une Institution d'Enseignement Supérieur (IES) de gérer de manière efficace, sécurisée et organisée tous les documents entrants liés à ses activités administratives et académiques.

I.2.2. Objectifs spécifiques

Les objectifs spécifiques de notre sujet sont :

Faciliter l'accès rapide aux documents administratifs et académiques de la faculté ; éviter la détérioration des documents par usure ; mieux sécuriser les documents ; éviter la perte des documents ; améliorer la collaboration entre différents utilisateurs ; optimiser l'espace physique de stockage des documents ; réduire des erreurs et des redondances ; numériser les documents.

I.3. Problématique

La problématique de la, "Mise en place d'un système automatisé d'archivage des documents entrants administratifs et académiques de la faculté", est la suivante : "Comment améliorer efficacement l'archivage des documents administratifs et académiques au sein de la faculté en mettant en place un système automatisé d'archivage, tout en garantissant l'accessibilité, la sécurité et l'efficacité des processus d'archivage?". Cette problématique soulève la question de l'amélioration du processus d'archivage au sein de la faculté en utilisant des technologies automatisées. Elle met l'accent sur les défis liés à l'archivage des documents entrants, tels que la recherche, la classification, l'accessibilité, la sécurité et l'efficacité des

processus. L'objectif est de trouver des solutions pour surmonter ces défis et améliorer globalement l'archivage des documents administratifs et académiques de manière automatisée. Ainsi, les différents problèmes sont : La destruction des papiers, les risques naturels tels que les incendies, les inondations etc., la perte des documents, manque d'espace physique à cause d'un grand nombre de documents, diminuer le temps de recherche d'un document.

I.4. Solutions proposées

Face aux différents problèmes dus à l'archivage manuel, Nous proposons comme solution la mise en place d'un système automatise d'archivage des documents entrants administratifs et académiques au sein de la faculté car ce dernier permet le traitement des documents entrants plus rapidement, ainsi que le temps nécessaire pour les trier, les classer et les enregistrer. Cela permet d'améliorer l'efficacité globale du flux de travail ; Avec un système automatisé, il est beaucoup plus facile de rechercher et d'accéder aux documents, le système automatisé permet de centraliser l'archivage des documents entrants.

I.5. Résultats attendus

Une fois le système mis en place permettra l'amélioration de la qualité du service, la diminution de perte de données liée à la détérioration du support papier, la réduction de l'encombrement des dossiers (papiers), la garantie de conservation aussi longtemps que possible des informations et la diminution du temps de recherche et de traitement des données. Toutes ces solutions envisagées ici vont concourir à l'exécution rapide, fiable, juste et sans beaucoup de peine lors les activités liées à l'archivage des documents au sein de la FSI. Ceci dans la mesure où elle assura l'accès immédiat à l'information simple à manipuler de façon à permettre au personnel de la FSI de pouvoir manipuler les données permettra l'accès au moment opportun aux personnes habilitées.

I.6. Apports scientifiques et technologiques

Le présent travail permet de résoudre des problèmes complexes à savoir l'absence d'un système d'information, l'automatisation des rapports et le développement d'un système

automatisé lors de l'archivage des documents au sein de la faculté. Les principaux apports de ce travail de recherche sont entre autres l'automatisation des opérations et la sécurité des données.

I.7. Outils mathématiques et plateforme de conception

La modélisation mathématique peut être utilisée pour représenter les processus d'archivage des documents de manière formelle. Cela peut inclure la modélisation des flux de travail, des processus de recherche et de récupération de documents, ainsi que des mécanismes de classification et d'indexation.

Pour notre système d'archivage des documents administratifs et académiques au sein de la faculté, nous avons choisir le modèle relationnel comme outil mathématique. Ainsi, les outils de conception et de développement que nous avons utilisé sont : UML pour la conception du système d'information ; le langage PHP ; le HTML, Bootstrap, JAVASCRIPT ; le langage SQL etc.

I.8. Méthodologie utilisée

Compte tenu de la nature du sujet, nous avons fait recours aux différentes techniques de collecte et d'analyse de données (enquête par sondage, entretien, l'observation, la documentation), l'analyse et modélisation des systèmes informatiques (modélisation par UML), les modèles mathématiques (l'algèbre relationnelle) ainsi que, la lecture des livres et des mémoires, des articles scientifiques et la recherche sur internet.

I.9. Conclusion

Dans ce chapitre, nous avons justifié le contexte du projet tout en précisant son objectif principal, sa problématique, son apport scientifique, la méthodologie pour définir les techniques ou les procédures utilisées pour identifier et analyser les informations concernant ce sujet de recherche spécifique et enfin on a fait une approche de solution qui consiste à concevoir et à développer ce système qui facilitera les fonctionnalités énumérées précédemment.

CHAPITRE II. PRESENTATION GENERALE DU CAS D'ETUDE

II.1. Présentation de l'Université du Burundi

Créée en 1964, l'Université du Burundi (UB) compte actuellement plus de 16000 étudiants répartis dans 8 Facultés et 7 Instituts [19]. Avec ses effectifs des étudiants inscrits mais aussi en termes d'effectifs du personnel et des filières d'enseignements, elle reste la plus importante institution publique de l'Enseignement Supérieur du Burundi. Elle ambitionne une formation d'excellence basée sur un enseignement de qualité, et se veut être une université de référence dans la sous -région.

L'Université du Burundi a évolué dans son organisation sous la tutelle du Ministre de l'Enseignement Supérieur et de la Recherche Scientifique L'Université du Burundi est organisée en Facultés et Instituts qui sont organisés en départements et en sections. L'Université du Burundi comprend en outre les services administratifs et techniques, les régies, les centres et organismes de recherches nécessaires à l'accomplissement de sa mission. La gestion quotidienne est assurée par le Recteur assisté par le Vice-Recteur tous nommés par Décret sur proposition du Ministre ayant en charge l'Enseignement Supérieur dans ses attributions ;

L'Université du Burundi est administrée par un Conseil d'Administration de 9 membres ; Un Secrétaire Général qui est nommé par décret sur proposition du Ministre de l'Enseignement Supérieur et de le Recherche Scientifique chargé de la coordination administrative de l'Université du Burundi. Comme nous l'avons évoqué précédemment, l'Université du BURUNDI comporte 8 facultés et 7 instituts suivants :

Les 8 facultés sont : Faculté des Sciences Economiques et de Gestion (Campus Mutanga) ; Faculté des Sciences de l'ingénieur (Campus Kiriri) ; Faculté d'Agronomie et de Bio-Ingénierie (Campus Mutanga ; Campus Zege (centre-ville de Gitega) ; Faculté de la Psychologie et Sciences de l'Education (Campus Kamenge) ; Faculté de Médecine (CHUK) ; Faculté des Sciences ; Faculté des Lettres et Sciences Humaines (Campus Mutanga) ; Faculté des Droits, Sciences Politiques et Relations Internationales (Campus Mutanga).

Les 7 instituts sont : Institut de l'Education Physique et Sport (Campus Kiriri) ; Institut Supérieur et Commerce (Campus Rohero) ; Institut de Pédagogie Appliquée (Campus Rohero) ; Institut Supérieur de Formation Agricole (campus Zege) ; Institut de Statistique Appliquée (Campus Nyamugerera) ; Institut d'administration et cartographies foncières (campus Buhumuza) ; Institut Supérieur de sciences Appliques.Pour notre cas, nous allons nous intéresser sur l'une des facultés ci-haut cite : la faculté des sciences de l'ingénieur FSI sur lequel présente le sujet d'étude.

II.2.Présentation de la Faculté des sciences de l'ingénieur

II.2.1. Historique

La Faculté des Sciences Appliquées a vu le jour en 1982 pour la formation des Ingénieurs en Génie Electromécanique et en Génie Civil. Elle a été mise sur pied avec la collaboration de l'UNESCO/PNUD (pour le Département d'Electromécanique) et de plusieurs institutions de la République Fédérale d'Allemagne (pour le Département du Génie Civil). La formation était subdivisée en deux cycles : le 1er cycle avait les candidatures en 2 ou 3 ans et le 2ème cycle l'Ingéniorat en 3 ans. La durée normale de la formation se chiffrait donc à cinq ans et le grade obtenu était celui d'Ingénieur Civil des constructions ou celui d'Ingénieur Civil en Electromécanique. En juin 1983, L'Institut Technique Supérieur (I.T.S.) a été créé au sein de ladite Faculté dans le souci de former (en 3ans) des Ingénieurs techniciens, intermédiaires entre l'Ingénieur de conception et le technicien de niveau A2.En 1989, sur demande des employeurs, la formation à l'Institut Technique Supérieur (I.T.S.) a passé de 3 à 4 ans avec trois départements (Electromécanique, Génie civil, Aménagement & Urbanisme) pour former des Ingénieurs industriels intermédiaires entre l'Ingénieur de conception et l'Ingénieurs technicien, avec équivalence de la licence délivrée par les autres facultés de l'Université. Par l'ordonnance N°610/890 du 18/03/2009, un département des Technologies de l'Information et de la Communication organisant la formation dans le système LMD a été ouvert au sein de l'ITS afin d'adapter l'enseignement supérieur aux standards internationaux, en particulier à ceux de la Communauté est-africaine,

Le système BMD, -master-doctorat a été mis en place.

Il repose sur trois niveaux de diplômes : bachelier sur trois ans, master sur deux ans et doctorat, sur trois ans et met l'accent sur la professionnalisation. C'est ainsi qu'en décembre 2011, la Faculté des Sciences de l'Ingénieur (FSI) a vu le jour par fusion de la Faculté des Sciences Appliquées (FSA) avec l'Institut Technique Supérieur (ITS) survenue à l'issue de la mise en place du système BMD par la loi N°1/22 du 30 décembre 2011 portant réorganisation de l'Enseignement Supérieur au Burundi.

II.2.2. Offre de la formation

1. Introduction

La Faculté des Sciences de l'Ingénieur dispense la formation en cinq ans (dans le système BMD) sanctionnée d'un diplôme de « Master –Ingénieur Civil ».A la fin de la 3ème année (six semestres), la réussite des 180 crédits est sanctionnée d'un diplôme de « Bachelier – Ingénieur des travaux » donnant accès directement au marché du travail ou à la poursuite des études de Master Professionnel-Ingénieur Civil ou de Master Recherche, pour ceux qui ont obtenu la note requise pour être admis aux dits mastères ;

La faculté est composée de quatre Départements avec différentes options : Département de Génie Civil et Environnemental (GCE) ; Département d'Architecture et Urbanisme (AU) ; Département de Génie Mécanique et Electrique (GEM) ; Département des Technologies de l'Information et de la Communication (TIC).

2. Organisation de la formation

2.1. Domaine de la formation : « Sciences et Technologies »

La Faculté des Sciences de l'Ingénieur dispense la formation dans le domaine des Sciences et Technologie organisée comme suit :

Baccalauréat (6semestres, 180 crédits) : Les deux premiers semestres (1ère année du Baccalauréat) sont communs à tous les trois premiers Départements. Pour les 4 semestres suivant (semestres 3-6), le programme des études diffère selon l'option suivi par l'étudiant. Quatre filières sont proposées au niveau du Baccalauréat : Mécanique, Electricité, Génie Civil et environnemental, Architecture et Urbanisme, avec différentes options à partir du 5ème semestre.

A la fin du 6ème semestre les étudiants remplissant les conditions exigées obtiennent un diplôme de « Bachelier-Ingénieur des travaux ». (Dénomination encore à convenir) Mastère professionnel et ou Mastère Recherche (4semestres, 120crédits)

L'organisation des Mastère professionnels est envisagée à partir de l'année académique 2015-2016. Toutefois, pour le département des TIC, deux Mastères professionnels sont déjà organisés, à savoir, Mastère Génie Informatique et Mastère Génie des Communications.

2.2. Objectif de la formation

Le programme de formation vise à former des Bacheliers-Ingénieurs des travaux et des Maîtres-Ingénieurs civils. La formation des Bacheliers-Ingénieurs des travaux a pour objectifs principaux de permettre l'acquisition des compétences techniques suivantes :Interpréter un projet ;Conduire et créer de manière optimale un projet ;Conduire et créer de manière optimale un projet ;Travailler en équipe ; Etre capable de gérer son équipe ; Interpréter les schémas, les techniques et les exécuter ; Exécuter correctement le plan de travail ; Comprendre le langage employé dans son domaine ;Avoir de la dextérité dans la pratique ; Vérifier le respect des normes ; Flexibilité intellectuelle (apprendre par les anciens et subalternes plus expérimentés) ; Esprit de curiosité technique ; Vérifier la faisabilité et la vraisemblance des résultats ; Exploitation des ressources de façon optimale.

A la fin de la formation, les « Maîtres-Ingénieurs civils auront acquis en plus des compétences des « Bacheliers-Ingénieur des travaux », les compétences techniques suivantes :Créer et gérer un projet ; Savoir-faire pratique ;Savoir intégrer les connaissances acquises dans la pratique ; Savoir organiser et connaître les gens, les collègues ; Savoir identifier, poser et résoudre un problème (technique ou autre) ; Capacité de décision ; Esprit de recherche, d'innovation ; Etre capable de concevoir ; Planifier son travail, encadrer et organiser efficacement ses subalternes ;Etre capable d'adapter la technologie aux conditions locales/problèmes tout en respectant les normes ;Etre capable de communiquer son savoir (rédiger des notes techniques ; faire une présentation ; Etre capable de mettre en pratique son enseignement (traduire son bagage théorique en application).

II.3. Organisation et fonctionnement de la FSI

II.3.1. Décanat

La gestion académique et administrative de la FSI est assurée par le Décanat sous la direction d'un Doyen élu par le conseil de la faculté, approuvé par le Recteur de l'Université du BURUNDI. Le doyen est le chef de la faculté, responsable de la direction et de la coordination globale des activités académiques et administratives. Il est souvent assisté par le Vice Doyen qui soutient le doyen dans la gestion des affaires académiques et administratives, et peut assumer les responsabilités du doyen en son absence. [20]

Le Secrétaire administratif assiste le doyen dans la gestion des tâches administratives, la communication interne et externe, et la coordination des activités du décanat. Il fournit ainsi un soutien administratif général et traite la correspondance, organise les réunions, gère les archives.

II.3.2. Secrétariat

Le secrétariat de la faculté est une entité administrative au sein d'une université ou d'une institution éducative qui s'occupe des tâches administratives et organisationnelles nécessaires au bon fonctionnement de la faculté. Le secrétariat de la faculté joue un rôle crucial dans le soutien des activités académiques et administratives de la faculté, en veillant à ce que les opérations se déroulent de manière ordonnée et efficace.

Voici une liste générale des rôles typiques que l'on peut trouver au sein d'un secrétariat : Un Secrétaire en Chef ou Secrétaire Principal(e) qui supervise et coordonne toutes les activités du secrétariat, s'assurant que les tâches administratives sont effectuées de manière efficace et en temps opportun ; Un Assistant(e) de Direction ou Secrétaire de Direction qui fournit un soutien administratif direct à un ou plusieurs membres de la direction, gérant les agendas, la correspondance et les communications ;

Un Secrétaire Administratif ou Assistant Administratif qui traite la correspondance, rédige des documents, organise des réunions, prépare des rapports et effectue d'autres tâches administratives générales ;Un Réceptionniste ou Secrétaire d'Accueil qui accueille les visiteurs, répond aux appels téléphoniques, gère les rendez-vous et assure un premier contact professionnel avec les personnes entrant en contact avec l'organisation ;

Un Assistant(e) de Bureau ou Secrétaire Général(e) qui assiste divers départements ou services de l'organisation en fournissant un soutien administratif, en traitant la correspondance et en organisant des activités spécifiques ;Un Secrétaire de Comité ou Coordonnateur de Réunions qui Organise et coordonne les réunions, rédige les ordres du jour, prépare les documents et prend des notes lors des réunions ;Un Archiviste ou Spécialiste de l'Archivage qui Gère la classification, le stockage, la récupération et la disposition des documents et des archives de l'organisation ;un Secrétaire de Service qui Fournit un soutien administratif à un service spécifique de l'organisation, comme les ressources humaines, les finances, les ventes, etc. ;Un Agent de Communication ou Coordonnateur des Communications qui gère les communications internes et externes de l'organisation, y compris la coordination des annonces, des bulletins d'information et des publications ;Un Assistant(e) de Gestion des Fournitures ou Gestionnaire des Stocks qui gère les fournitures de bureau, les équipements et les stocks nécessaires pour le fonctionnement du bureau ; Un Agent d'Accueil ou Agent de Services Généraux qui s'occupe des services généraux, tels que la réception, la gestion des locaux, l'approvisionnement et la coordination des services aux employés.

Ces postes varient en fonction de la taille, de la structure et des besoins spécifiques de l'organisation, mais ils sont essentiels pour garantir le bon fonctionnement administratif et la coordination efficace des activités au sein de l'organisation.

II.3.3. Conseil de la faculté

Le conseil de la faculté est constitué du personnel académique, des chefs de travaux, d'un représentant des étudiants de chaque département.

Le conseil de faculté exerce les attributions suivantes : Il approuve les programmes de recherche et d'enseignement, et organise des réunions scientifiques ;

Il donne ses avis en matière de nomination et de promotion du personnel académique et scientifique ; Il effectue les attributions des enseignants chaque année académique.

II.3.4. Jury de délibération

Chaque année académique, sur proposition du Doyen, le conseil de la faculté constitue le jury pour chaque classe. Il en nomme les présidents, les secrétaires, ainsi que les suppléants.

Les jurys sont composés de membres du personnel enseignant et scientifiques qui participent à la formation des étudiants. C'est une instance académique chargée de l'évaluation des compétences et de leur certification ainsi que l'organisation des épreuves correspondante.

Les jurys ont pour fonction de : Veiller par l'entremise de leur président à la bonne marche des évaluations, au secret de délibération ainsi qu'à la publication des résultats ; Délibérer sur l'ensemble des résultats de chaque candidat et de statuer sur la validation des crédits, de décision et la mention qu'il convient de lui attribuer ; Recevoir et statuer sur les éventuels recours.

Il est à souligner que ce jury est composé par des enseignants à temps plein de ladite faculté, et, est formellement indépendant. C'est à dire que lorsqu'il n'est pas encore abrogé, aucun autre organe interne ou externe à la faculté ne peut lui dicter des ordres ou des directives à suivre. En plus les travaux du jury sont strictement gardés secrets.

II.4. Conclusion

Dans ce chapitre, j'ai fait une présentation sur l'université du BURUNDI plus particulièrement l'organisation administratifs et académiques de la FSI, Sa mission ; son historique ainsi que son fonctionnement.

CHAPITRE III. MODELISATION DU SYSTEME AVEC LE LANGAGE UML

III.1. Langage UML

III.1.1. Définition

Le langage UML (Unifié Modeling Langage) est un langage graphique de modélisation utilisé pour représenter visuellement et communiquer les concepts, les structures, les comportements et les relations d'un système logiciel ou d'un processus. Il fournit un ensemble de notations standardisées qui permettent aux concepteurs, aux développeurs, aux architectes logiciels et aux autres parties prenantes de comprendre et de documenter les aspects d'un système de manière claire et cohérente. [16]

L'UML est constitué de diagrammes qui servent à visualiser et décrire la structure et le comportement des objets qui se trouvent dans un système. Il permet de présenter des systèmes logiciels complexes de manière plus simple et compréhensible qu'avec du code informatique. L'UML a des applications dans le développement logiciel, mais aussi dans l'industrie (pour modéliser les flux de processus par exemple), dans l'ingénierie ou le marketing.

III.1. 2. Historique

Avant même l'introduction d'UML dans le développement logiciel, le domaine de la programmation orientée objet (OOP) était déjà en plein essor. Ce style de programmation est basé sur le concept que tout est un objet : les éléments constitutifs d'un programme sont des objets qui interagissent les uns avec les autres. Les messages envoyés dans les deux sens sont également constitués d'objets. Chaque objet individuel est un exemple de sa classe supérieure. La classe elle-même agit également comme un objet et détermine le comportement des instances d'objet qu'elle contient. Les objets sont constitués de données et de code. L'objet organise les données en zones, également appelées attributs. Le code détermine leur procédure ou leur méthode. De la fin des années 1980 aux années 1990, de nombreuses méthodes et langages pour la représentation de la POO ont été développés et mis en œuvre. Il en est résulté une variété de méthodes très dissemblables. Pour unifier ces langages, les trois développeurs James Rumbaugh, Grays Booch et Ivar Jacobson ont décidé de fusionner plusieurs langages existants en un langage commun et standardisé.

Les trois avaient déjà créé leurs propres méthodes de développement logiciel orienté objet :

La méthode Booch ; Object mode Ling technique (OMT) ; Object-oriented software engineering method (OOSE).UML devrait définir la sémantique pour la représentation de ces méthodes comme langage de modélisation. Sous le nom de "UML Partners", les développeurs ont commencé à travailler sur l'achèvement d'UML dans une équipe en 1996. Ils l'ont ensuite remis à l'Object Management Group (OMG), qui a introduit la version 1.1 du langage de modélisation unifié comme norme en 1997.

Non satisfaits, les développeurs ont mis en place un groupe de travail pour améliorer le langage sur plusieurs versions. Les critiques existantes comprenaient une sémantique imprécise et inutilement complexe, un manque d'adaptabilité et une normalisation assez faible. Pour cette raison, une révision majeure a été effectuée. Le résultat a été UML 2.0, qui a établi la nouvelle norme en 2005. La version 2.4.1 constitue la base de la normalisation ISO 19505-1 (Infrastructure) et 19505-2 (Superstructure) de 2012. UML Version 2.5.1 est apparu en décembre 2017 ; Certains appellent "langage de modélisation unifié" parmi les langages de modélisation, ce qui est vraiment ce qu'elle visait à devenir. Comme mentionné ci-dessus, UML visualise les états des objets et les interactions entre eux au sein d'un système. Son utilisation répandue peut être due à la grande influence des membres du groupe de gestion d'objets (OMG) (y compris IBM, Microsoft et HP). La sémantique structurée y contribue également.

Les diagrammes UML montrent les composants système suivants : Objets individuels (composants de base) ; Classes (combinaison d'éléments ayant les mêmes propriétés ; Relations entre objets (hiérarchie et comportement / communication entre objets) ; Activité (combinaison complexe d'actions et de composantes comportementales) ; Interactions entre les objets et les interfaces.UML 2.0 définit des unités de langage qui fonctionnent à différents niveaux. Vous les utilisez pour exprimer la structure et le comportement d'un système. La méta-modélisation inclut tous les éléments d'UML, y compris ceux qui décrivent UML lui-même. Il utilise quatre niveaux hiérarchisés (M0 à M3).Le niveau méta-méta M3 spécifie les métadonnées du langage de modélisation et ses relations à l'aide de la fonction méta objet (MOF). Il définit le méta-modèle et permet également le transfert des métadonnées.

Le format XMI défini par les membres du groupe de gestion d'objets est un outil pratique pour partager des données orientées objet au niveau méta-méta entre les outils de développement. Le langage de contrainte d'objet (OCL), un langage de programmation déclaratif, complète UML et régule les conditions limites de la modélisation. En tant que langage texte, cependant, il n'est qu'un support et ne peut pas être utilisé pour la modélisation.

Il représente des objets concrets, réels et des enregistrements de données individuels - par ex. un objet ou un composant. Le niveau M1 comprend tous les modèles qui décrivent et structurent les données du niveau M0. Ce sont des diagrammes UML tels que le diagramme d'activité ou le diagramme de package (expliqué ci-dessous). Pour définir la structure de ces modèles, les méta-modèles M2 définissent les spécifications et la sémantique des éléments du modèle.

Si vous voulez créer un diagramme UML compréhensible, vous devez connaître le méta-modèle UML et ses règles. Le niveau le plus élevé, M3, est un méta-modèle du méta-modèle. La facilité de méta-objet mentionnée fonctionne à un niveau abstrait qui définit les méta-modèles. Ce niveau se définit lui-même, car autrement, des méta-niveaux supérieurs seraient créés.

III.1.3. Motivation du choix du langage UML comme outil de modélisation

L'UML est un langage graphique permettant de représentation des concepts de la programmation orientée objet grâce à des diagrammes spécifiques. C'est un langage de modélisation unifié, fruit d'un consensus entre industriels du logiciel et le monde de la recherche. Il permet de spécifier, visualiser et documenter les modèles de systèmes logiciels.

Grâce aux outils de modélisation UML (ou modeleur UML), vous pouvez analyser les besoins utilisateurs et concevoir vos futures applications en les représentant grâce aux 13 digrammes UML 2.0. UML permet de modéliser n'importe quel type d'applications, déployées sur n'importe quel type d'infrastructure (matériel, système d'exploitation, langage de programmation et réseau). Enfin, UML s'adapte naturellement aux langages de programmation orientés-objet manipulant par exemple des classes, des méthodes… mais il

peut être utilisé pour modéliser d'autres langages et environnements non-objets, notamment grâce à son mécanisme d'extension : les profiles UML. [3]

III.2. Diagrammes du langage UML

UML s'articule autour de 14 types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. [2]

Ces types de diagrammes sont répartis en deux grands groupes :

Six diagrammes structurels :

Diagramme de classes qui montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc. ;Diagramme d'objets qui montre les instances des éléments structurels et leurs liens à l'exécution ;Diagramme de packages qui montre l'organisation logique du modèle et les relations entre packages ; Diagramme de structure composite montre l'organisation interne d'un élément statique complexe ;Diagramme de composants qui montre des structures complexes, avec leurs interfaces fournies et requises ;Diagramme de déploiement montre le déploiement physique des « artefacts » sur les ressources matérielles.

Huit diagrammes comportementaux :

Diagramme de cas d'utilisation qui montre les interactions fonctionnelles entre les acteurs et le système à l'étude ; Diagramme de vue d'ensemble qui fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots ; Diagramme de séquence qui montre la séquence verticale des messages passés entre objets au sein d'une interaction ;Diagramme de communication qui montre la communication entre objets dans le plan au sein d'une interaction ;Diagramme de temps qui fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps ; Diagramme d'activité qui montre l'enchaînement des actions et décisions au sein d'une activité ;Diagramme d'états qui montre les différents états et transitions possibles des objets d'une classe ;Diagramme d'interaction qui fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.

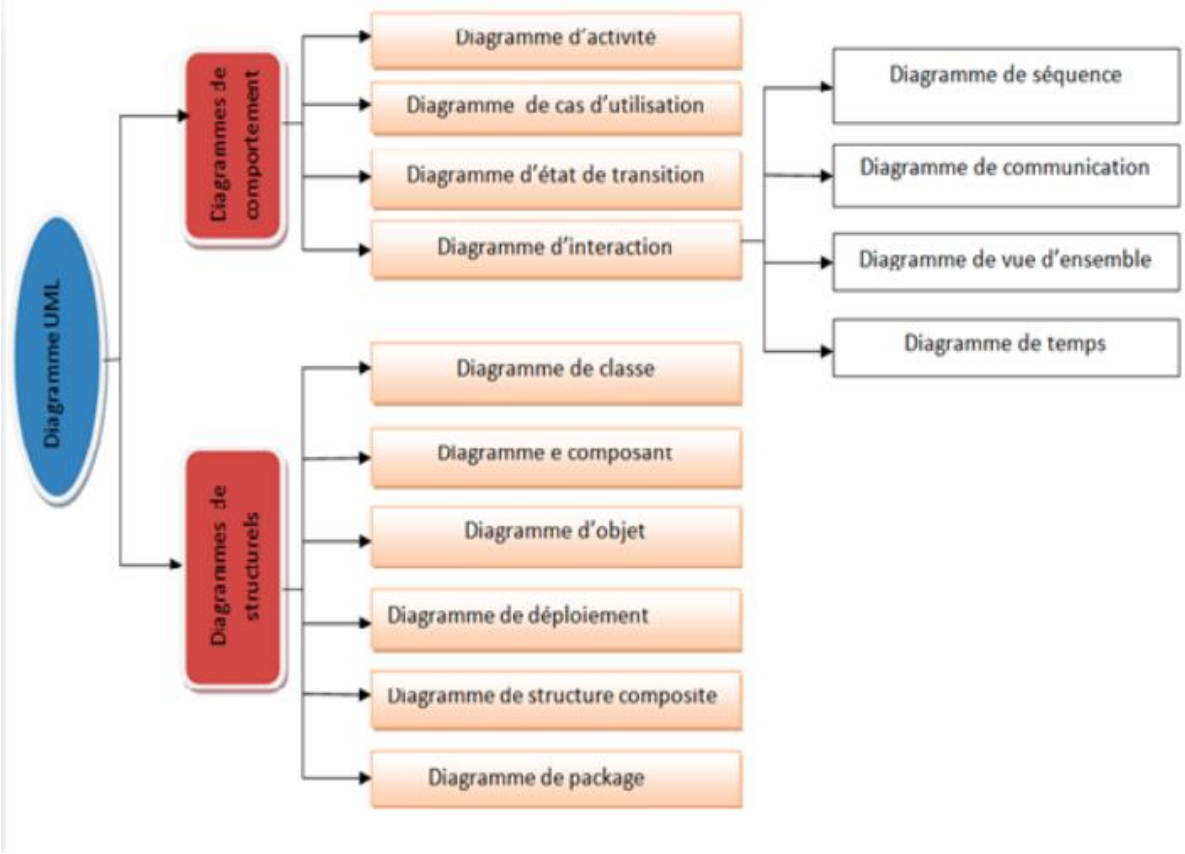


Figure 1: Diagramme UML

III.2.1. Diagramme des classes

Le diagramme des classes constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association.

Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme des classes ; seul l'algorithme général et le pseudocode correspondant peuvent être associés à la modélisation.

La description du diagramme des classes est fondée sur : le concept d'objet ; le concept de classe comprenant les attributs et les opérations ainsi que les différents types d'association entre classes.

Le schéma ci-dessous nous donne une vue globale de notre application. On a les classes principales qui vont nous servir à réaliser l'application :

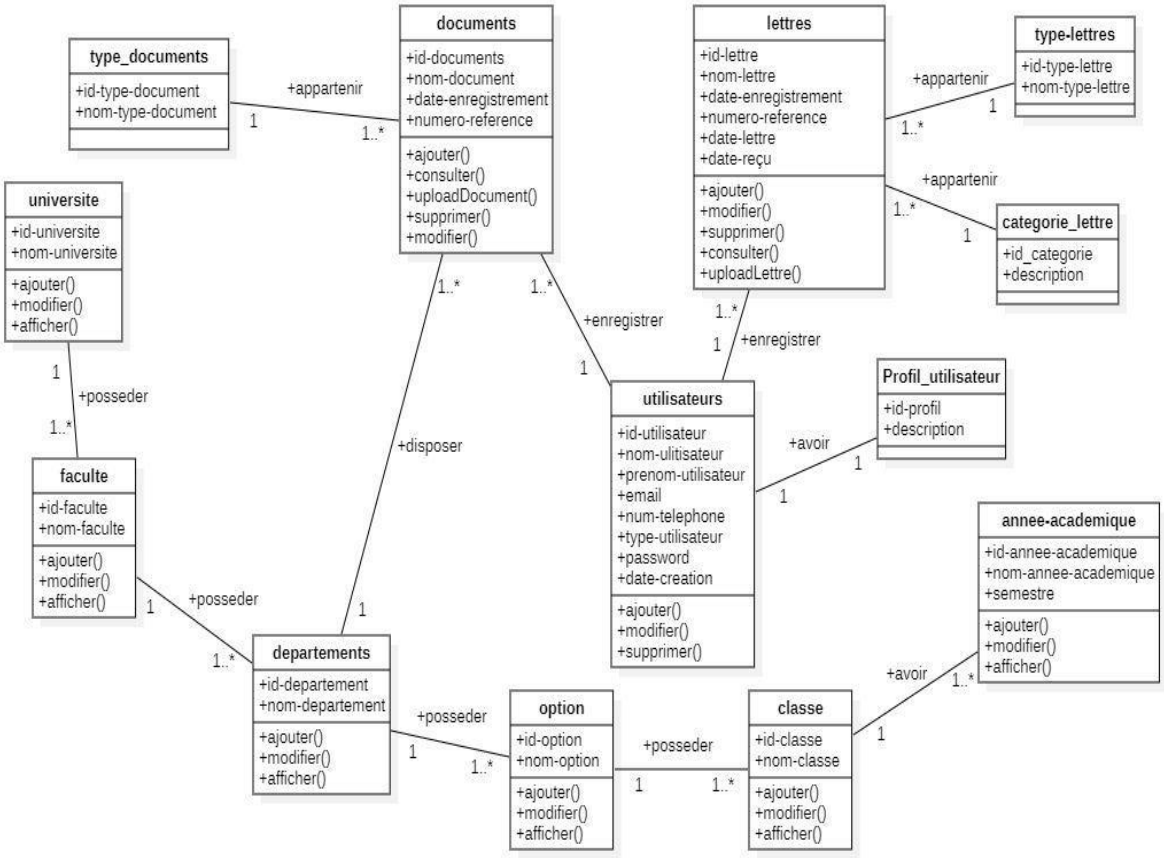


Figure 2: Diagramme de classe de l'application

III.2.2. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation permet de recueillir, d'analyser et d'organiser les besoins auxquels devrait répondre un système, pour donner une vision globale du comportement fonctionnel de ce système. L'activité du système a pour objectif de satisfaire les besoins de l'acteur. Il décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Dans tous les cas, il faut bien garder à l'esprit qu'il n'y a pas de notion temporelle dans un diagramme de cas d'utilisation.

Ainsi, Il existe différents éléments de modélisation des cas d'utilisation :

L'acteur est une entité qui agit sur le système et représente un ensemble cohérent de rôles que l'utilisateur peut effectuer.

Il existe 4 catégories d'acteurs : Les acteurs principaux c'est à dire les personnes qui utilisent les fonctions principales du système ; Les acteurs secondaires qui sont des personnes qui effectuent des tâches administratives ou de maintenance ; Le matériel externe qui sont des dispositifs matériels incontournables qui font partie du domaine de l'application et qui doivent être utilisés ; Les autres systèmes c'est-à-dire les systèmes avec lesquels le système doit interagir.

Un acteur est symbolisé de la manière suivante :

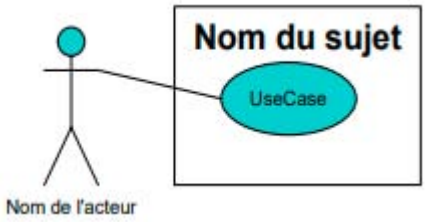


Le Cas d'utilisation, également dénommé « use cases » est une unité significative de travail représentant un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur. Chaque association signifie simplement « participe à ». Un cas d'utilisation doit être relié à au moins un acteur.

Il est symbolisé de la manière suivante :

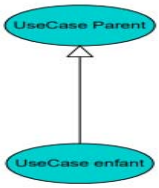


La relation : Elle exprime l'interaction existant entre un acteur et un cas d'utilisation.

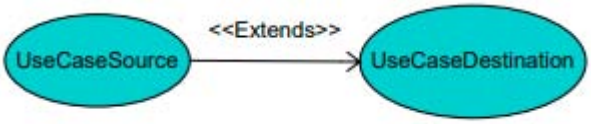


Il existe 3 types de relations entre cas d'utilisation :

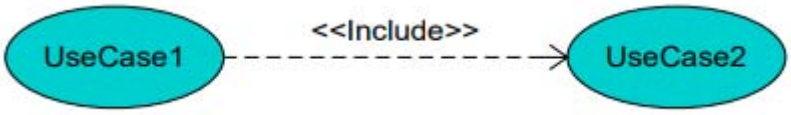
La relation de généralisation où le cas d'utilisation enfant est une spécialisation du cas d'utilisation parent.



La relation d'extension : Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à condition. Le comportement ajouté est inséré au niveau d'un point d'extension défini dans le cas d'utilisation destination.



La relation d'inclusion : Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. L'inclusion a un caractère obligatoire, la source spécifiant à quel endroit le cas d'utilisation cible doit être inclus.



Le schéma ci-dessous nous montre le diagramme des cas d'utilisation globale de notre application.

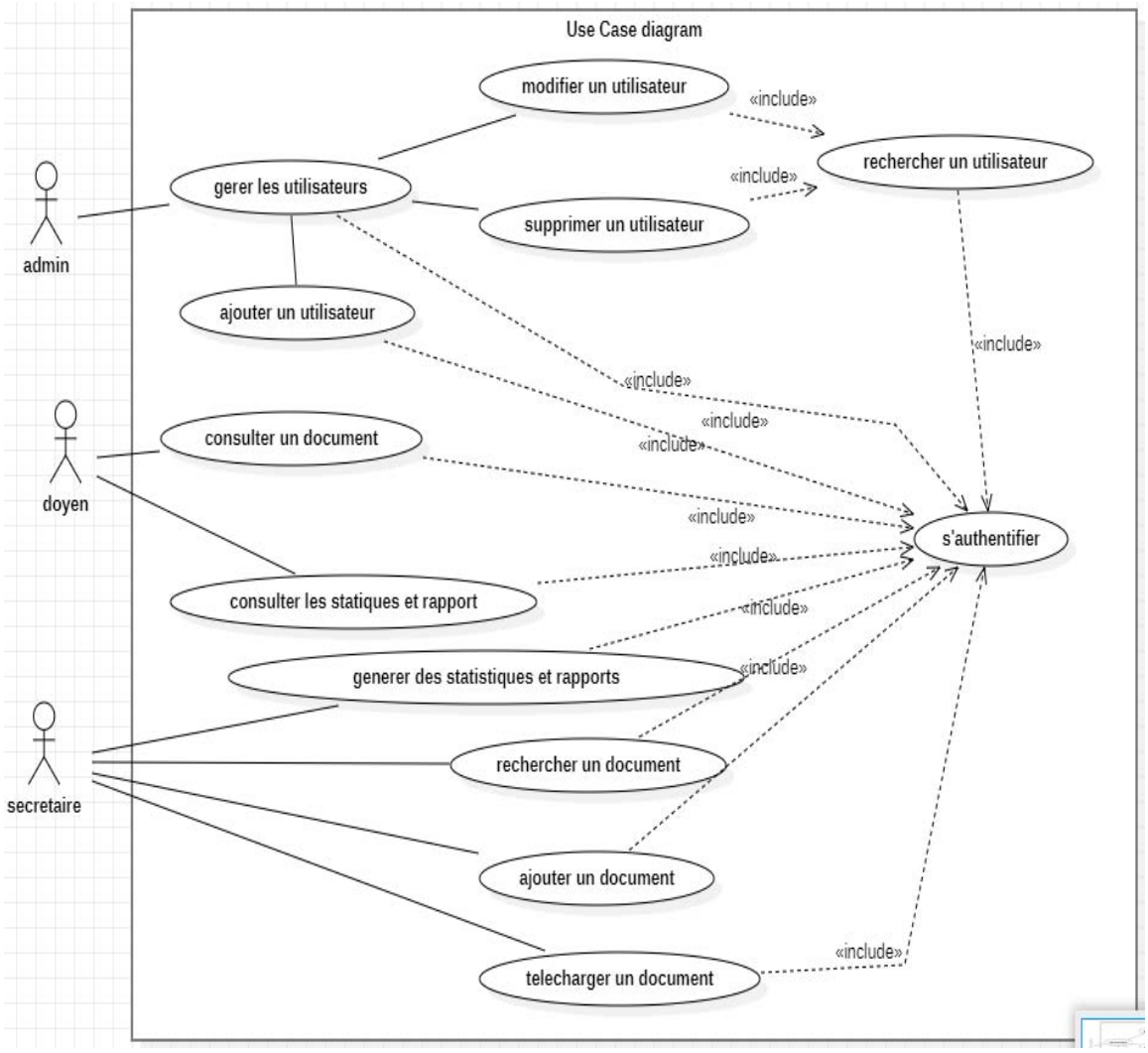


Figure 3: Diagramme de cas d'utilisation global de l'application

Ce diagramme montre les différentes actions possibles pour chaque type d'utilisateur et met en évidence les dépendances et les étapes nécessaires pour effectuer certaines actions. L'authentification est une étape clé pour tous les utilisateurs avant d'effectuer toute autre action. Les cas d'utilisation tels que "rechercher un utilisateur" sont des actions incluses nécessaires pour les opérations comme "modifier" ou "supprimer" un utilisateur.

Priorisation des cas d'utilisation

La priorisation des cas d'utilisation permet de décrire chronologiquement les opérations de l'utilisateur et de les classées selon leur ordre d'importance.

Le Tableau ci-dessous représente la priorisation des cas d'utilisation.

Tableau 1: Priorisation des cas d'utilisation

ACTEURS	Cas d'utilisation
Tous les utilisateurs	S'authentifier
Secrétaire administratif ou secrétaire simple	Ajouter un document
Secrétaire administratif ou secrétaire simple	Afficher un document
Doyen	Rechercher un document
Doyen	Consulter un document
Doyen ou Secrétaire administratif	Modifier un document

III.2.3. Diagramme des séquences système pour chaque cas d'utilisation

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont.

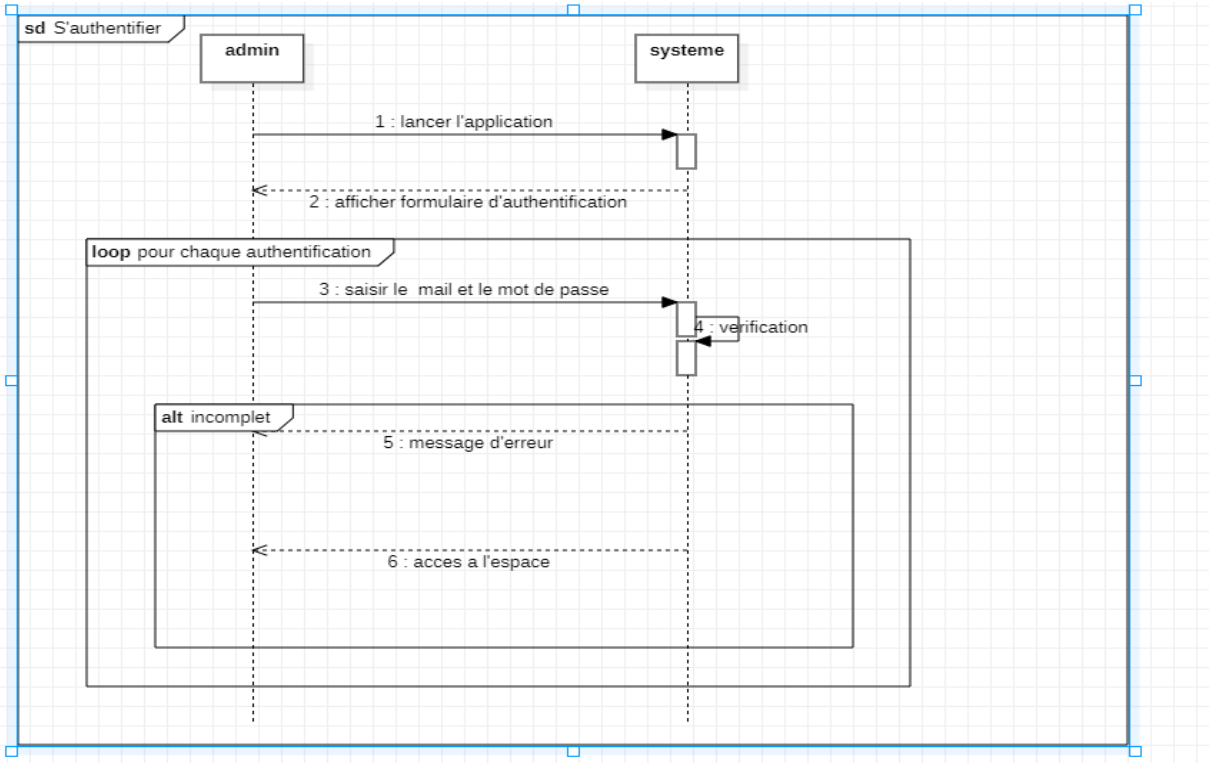


Figure 4: Diagramme de séquence du cas d'utilisation authentification

Voici une explication détaillée du principe de fonctionnement de ce diagramme :

Tableau 2: Diagramme de cas d'utilisation authentification

Étape	**Acteurs**	**Actions/Interactions**	**Messages échangés**
Lancer l'application	Admin-> Système	L'utilisateur lance l'application.	Lancer l'application
Afficher le formulaire	Système-> Admin	Le système affiche le formulaire d'authentification.	Afficher le formulaire d'authentification
Saisie des identifiants	Admin	L'utilisateur saisit son adresse e-mail et son mot de passe.	Saisir le mail et le mot de passe
Vérification des identifiants	Système	Le système vérifie les identifiants fournis par l'utilisateur.	Vérification
Identifiants incorrects	Système-> Admin	Si incorrects : Le système envoie un message d'erreur.	Message d'erreur
Identifiants corrects	Système-> Admin	Si corrects : Le système permet à l'utilisateur d'accéder à l'espace réservé.	Accès à l'espace réservé
Explication		Description du Processus	
Lancer l'application	Admin-> Système	L'utilisateur initie le processus d'authentification en lançant l'application	
Affichage du formulaire	Système-> Admin	Le système affiche un formulaire pour que l'utilisateur puisse entrer ses identifiants.	
Saisie des identifiants	Admin -> Système	L'utilisateur saisit ses identifiants (adresse e-mail et mot de passe).	
Vérification des informations	Système	Le système vérifie les informations fournies par l'utilisateur.	

En somme, le diagramme de séquence montre clairement les étapes que le système et l'utilisateur suivent pour s'authentifier, incluant les boucles pour les tentatives multiples et les conditions alternantes pour les échecs et les succès de l'authentification.

Diagramme de séquence du Cas d'utilisation : « ajouter un document »

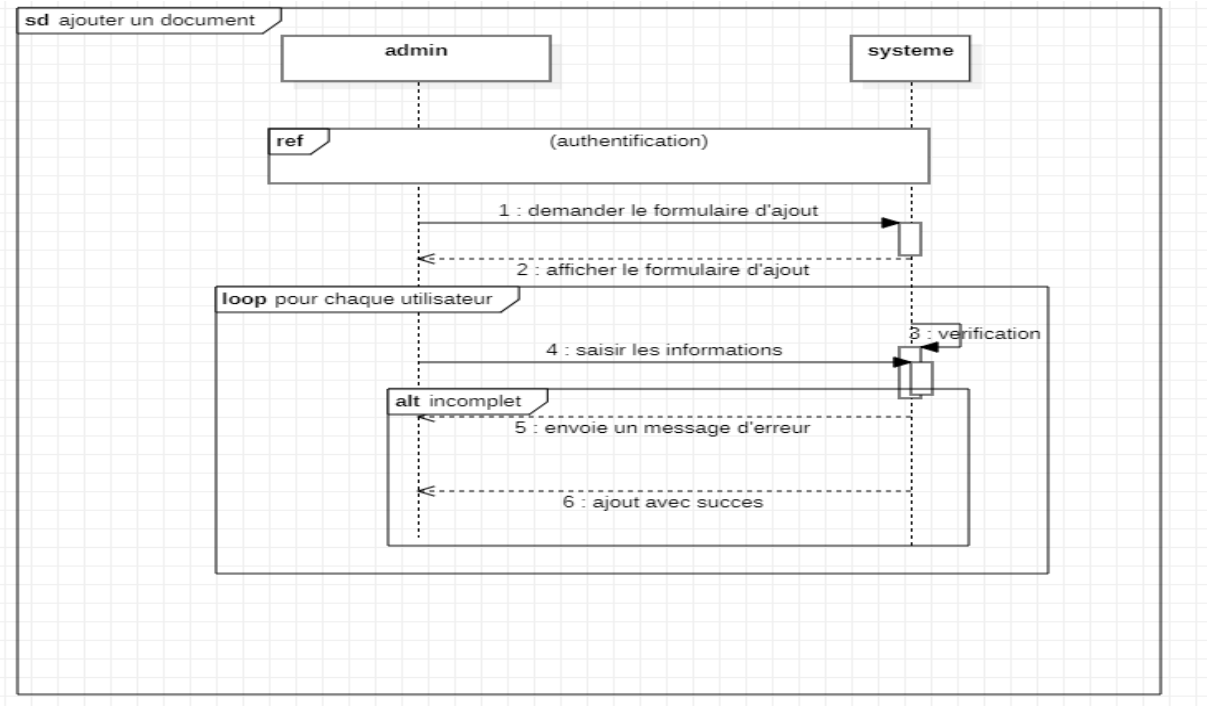


Figure 5: Diagramme de séquence du Cas d'utilisation : « ajouter un document »

Ce diagramme de séquence illustre le flux d'interactions pour le cas d'utilisation "ajouter un document". Voici une explication détaillée des éléments et du principe de fonctionnement de ce diagramme qui se résume dans le tableau suivant :

Tableau 3: Diagramme du cas d'utilisation ajouter un document

Étape	Acteurs	**Actions/Interactions**	Messages échangés
Authentification	Admin -> Système	L'utilisateur doit s'authentifier avant d'ajouter un document (référence à une étape non détaillée ici).	Référence (ref) : Authentification requise
Demander le formulaire d'ajout	Admin-> Système	L'admin envoie une requête au système pour obtenir le formulaire d'ajout de document.	1. Demander le formulaire d'ajout
Afficher le formulaire d'ajout	Système -> Admin	Le système répond en affichant le formulaire d'ajout de document à l'admin.	2. Afficher le formulaire d'ajout
Saisie des informations	Admin	L'admin remplit le formulaire avec les informations nécessaires.	Boucle : Pour chaque utilisateur
Vérification des informations	Système	Le système vérifie les informations saisies par l'admin.	Vérification
Résultat de la vérification		Alternative 1 : Informations incorrectes	
	Système -> Admin	Si les informations sont incorrectes ou incomplètes, le système envoie un message d'erreur à l'admin.	Message d'erreur
		Alternative 2 : Informations correctes Si les informations sont correctes, le document est ajouté avec succès.	Ajout avec succès

Ce tableau présente une vue organisée et détaillée des interactions et des messages échangés entre l'Admin et le Système.

Diagramme de séquence du Cas d'utilisation : « consulter un document »

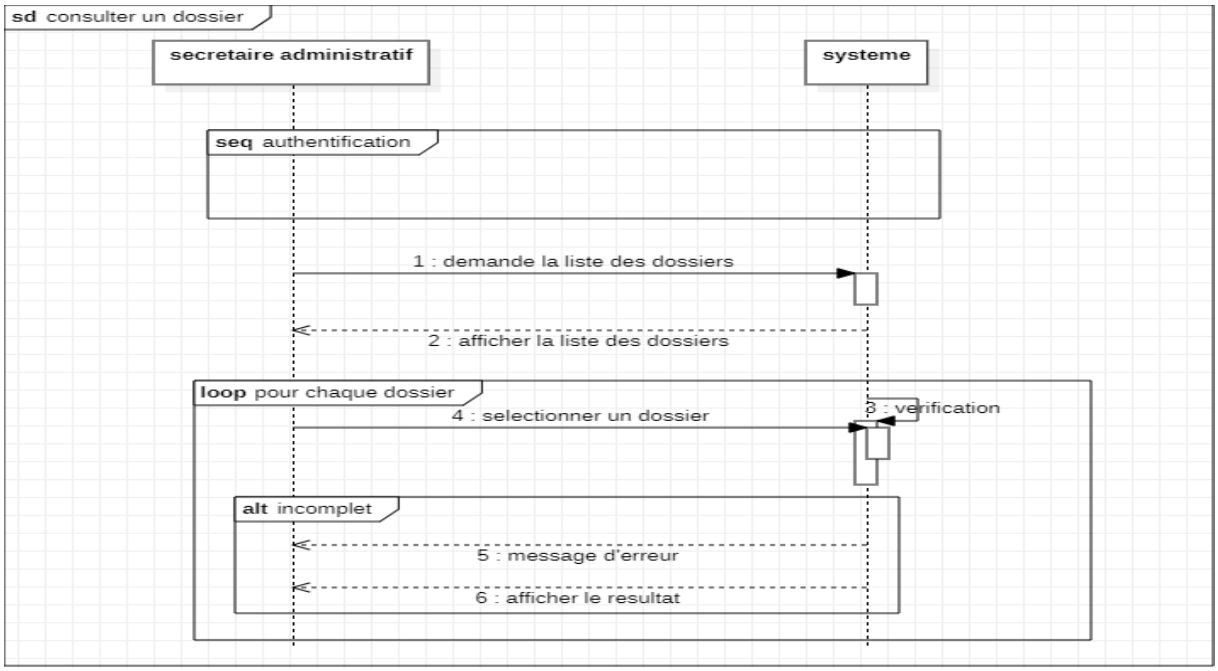


Figure 6: Diagramme de séquence du Cas d'utilisation : « consulter un document »

Tableau 4: Diagramme du cas d'utilisation consulter un document

Étape	**Acteurs**	**Actions/Interactions**	**Messages échangés**
Lancer l'application	Admin -> Système	L'utilisateur (Admin) lance l'application.	Lancer l'application
Afficher le formulaire	Système -> Admin	Le système affiche le formulaire d'authentification pour l'utilisateur.	Afficher le formulaire d'authentification
Saisie des identifiants	Admin	L'utilisateur saisit son adresse e-mail et son mot de passe dans le formulaire.	Saisir le mail et le mot de passe

Vérification des identifiants	Système	Le système vérifie les identifiants fournis par l'utilisateur.	Vérification
Résultat de la vérification		Alternative Identifiants incorrects	
	Système -> Admin	Si les identifiants sont incorrects, le système envoie un message d'erreur à l'utilisateur.	Message d'erreur
Résultat de la vérification	Système -> Admin	Alternative Identifiants corrects Si les identifiants sont corrects, le système permet l'accès à l'espace réservé pour l'utilisateur.	Accès à l'espace

Diagramme de séquence du Cas d'utilisation : « modifier un document »

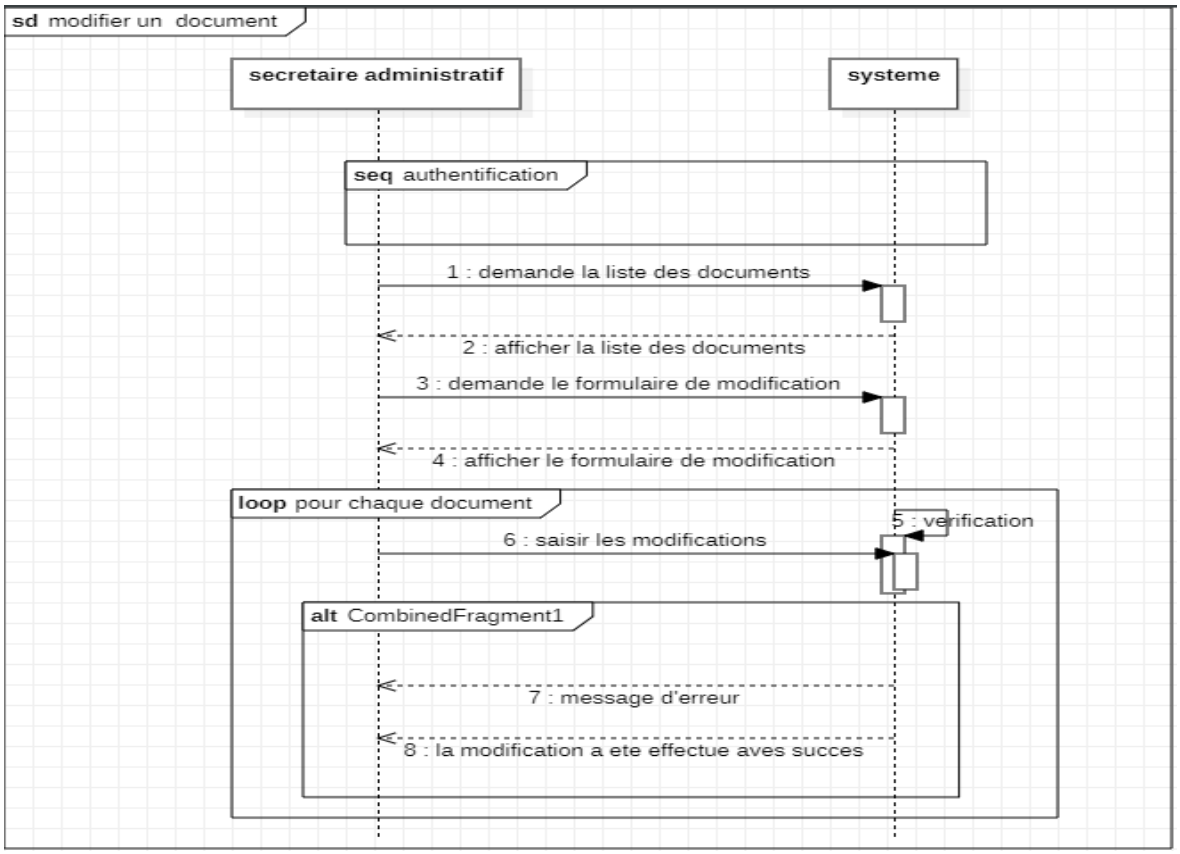


Figure 7:Diagramme de séquence du Cas d'utilisation : « modifier un document »

Le tableau ci-dessus montre une explication détaillée de ce diagramme :

Tableau 5: Diagramme de séquence du cas d'utilisation modifier un document

Étape	Description
Demande Initiale	L'utilisateur demande la liste des documents disponibles dans le système.
Affichage de la Liste	Le système répond en affichant la liste des documents.
Demande de Modification	L'utilisateur demande le formulaire de modification pour un document spécifique.
Affichage du Formulaire de Modification	Le système affiche le formulaire de modification pour le document sélectionné.
Saisie des Modifications	L'utilisateur saisit les modifications nécessaires pour le document.

Vérification et Résultats	
Vérification	Le système vérifie les modifications apportées pour s'assurer qu'elles sont correctes et complètes.
Erreur	Si une erreur est détectée (modifications incorrectes ou incomplètes), le système envoie un message d'erreur.
Succès	Si tout est correct, le système confirme que la modification a été effectuée avec succès.

Diagramme de séquence du cas d'utilisation « télécharger un document »

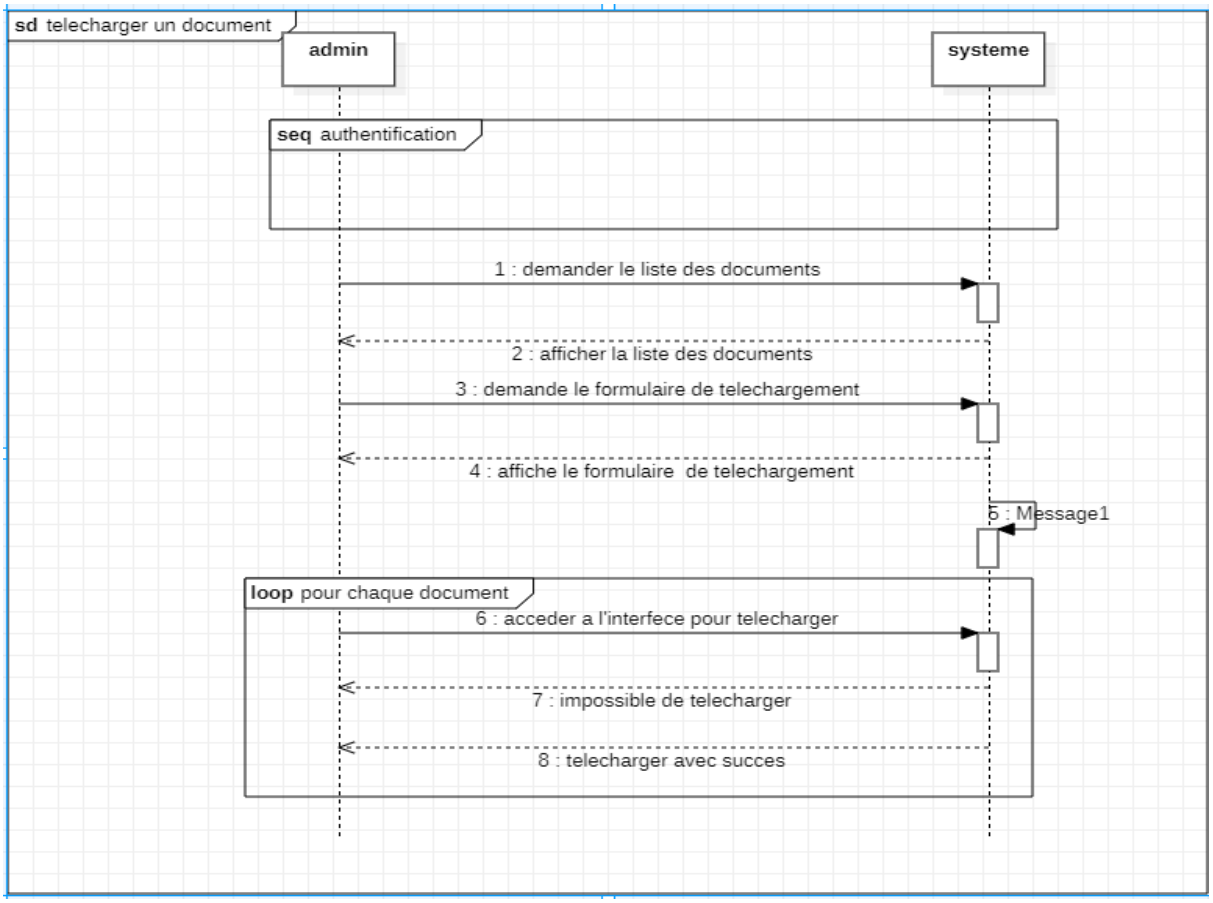


Figure 8: Diagramme de séquence du cas d'utilisation « télécharger un document »

Ce diagramme aide à visualiser l'interaction entre l'utilisateur et le système pour télécharger un document, mettant en évidence les différentes étapes et vérifications nécessaires pour compléter cette action.

Le tableau ci-dessous présente clairement les étapes clés et les résultats attendus dans ce processus de demande et de téléchargement de document.

Tableau 6: Diagramme de séquence du cas d'utilisation télécharger un document

Étape	Acteurs	Actions/Interactions	Messages échangés
Lancement de la demande	Utilisateur -> Système	L'utilisateur (secrétaire administratif) demande la liste des documents disponibles.	Demande la liste des documents
Affichage de la liste des documents	Système -> Utilisateur	Le système renvoie et affiche la liste des documents disponibles.	Affichage de la liste des documents
Sélection d'un document	Utilisateur -> Système	L'utilisateur sélectionne un document spécifique et demande son téléchargement.	Sélectionne le document et demande le téléchargement
Vérification	Système	Le système vérifie si l'utilisateur a l'autorisation et si le document est disponible.	Vérification des permissions et disponibilité
Si vérification réussie	Système -> Utilisateur	Le document est téléchargé et renvoyé à l'utilisateur.	Téléchargement du document
Si vérification échoue	Système -> Utilisateur	Le système renvoie un message d'erreur à l'utilisateur, indiquant l'échec du téléchargement.	Message d'erreur
Boucle	Système	Si plusieurs utilisateurs effectuent cette action simultanément, le système traite chaque demande individuellement pour chaque utilisateur.	-

Diagramme de séquence pour la gestion des utilisateurs

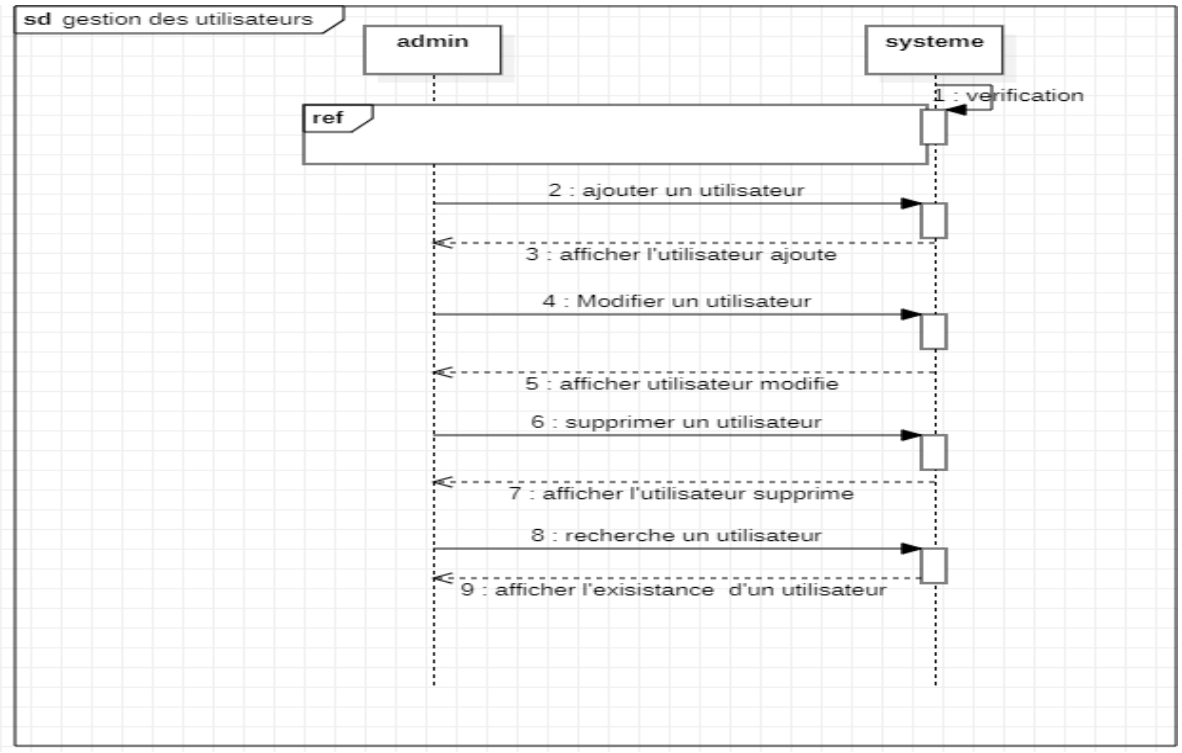


Figure 9: Diagramme de séquence pour la gestion des utilisateurs

La gestion des utilisateurs est une fonctionnalité cruciale pour toute application qui nécessite un contrôle d'accès. Elle permet aux administrateurs de gérer les comptes des utilisateurs, y compris l'ajout, la modification, la suppression, et la consultation des informations des utilisateurs. Ce diagramme de séquence montre clairement les interactions entre l'admin et le système pour gérer les utilisateurs, incluant les étapes de vérification nécessaires pour s'assurer que toutes les opérations sont autorisées et valides.

III.2.4. Diagramme de déploiement

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondant aux supports physiques (serveurs, routeurs…) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables…) sur ces nœuds. C'est un véritable réseau constitué de nœuds et de connexions entre les nœuds qui modélise cette architecture.

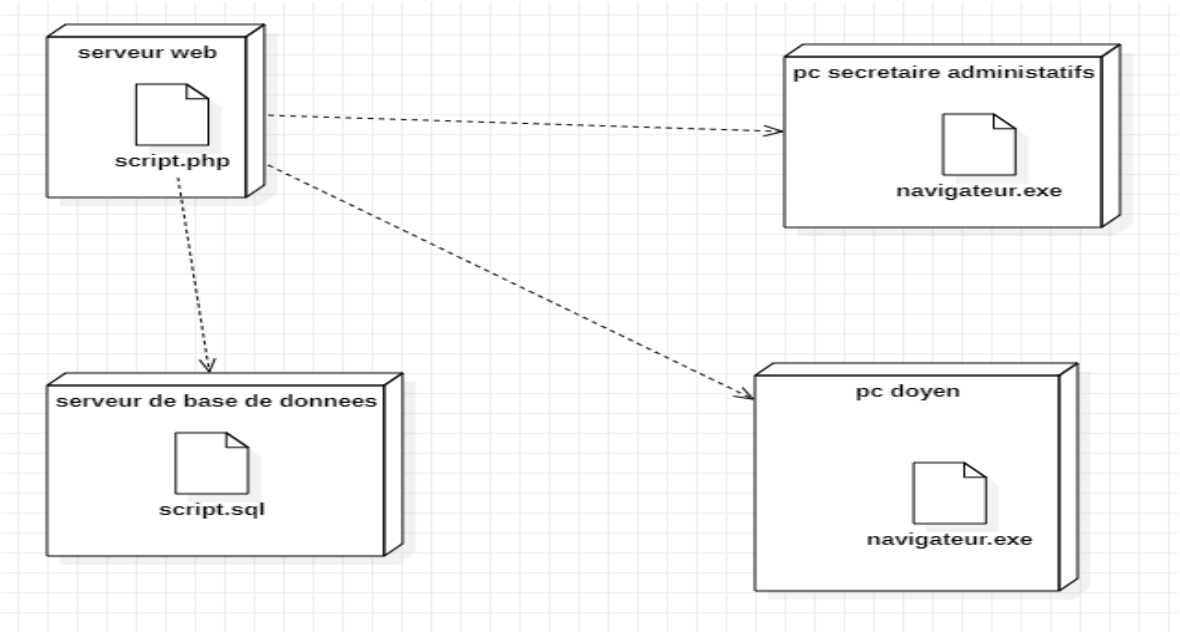


Figure 10:Diagramme de déploiement

En résumé, ce diagramme de déploiement fournit une vue d'ensemble de la configuration physique du système, montrant comment les composants logiciels sont distribués à travers les ressources matérielles.

Il aide à comprendre l'architecture déployée, les connexions et les interactions entre les différents éléments du système.

Ainsi, elles représentent les éléments logiciels déployés sur les nœuds. Ils peuvent être des applications, des services, ou des modules. Les composants sont souvent représentés par des rectangles.

Elles montrent les interactions entre les nœuds. Ces interactions peuvent inclure des protocoles de communication, des messages échangés, ou des flux de données.

III.2.5. Diagramme d'activité

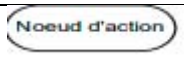
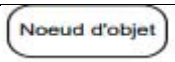
Le diagramme d'activité est l'un des types de diagrammes utilisés dans la modélisation UML.Il est utilisé pour représenter le flux de contrôle d'un processus ou d'un comportement dans un système, que ce soit au niveau logiciel ou au niveau métier.

Le diagramme d'activité est utile pour modéliser des processus complexes, des algorithmes, des workflows, des scénarios d'utilisation, etc. Il permet de décrire les étapes, les décisions, les actions et les flux de contrôle dans un format graphique facile à comprendre.

Voici quelques-uns des composants les plus courants d'un diagramme d'activités que je vais utiliser : Une Action qui est étape dans l'activité où les utilisateurs ou le logiciel exécutent une tâche donnée ; Le nœud objet qui est l'association entre un objet (instance d'une classe) et un état. Il représente un objet dans un état particulier. Le symbole du nœud d'objet est un rectangle ; Un Nœud de décision qui représente l'embranchement conditionnel dans le flux, et est représenté par un losange. Il comporte une seule entrée et au moins deux sorties ; Un Nœud de départ qui est l'élément symbolisant le début de l'activité, que l'on représente par un cercle noir ; et Un Nœud de fin l'élément symbolisant l'étape finale de l'activité, que l'on représente par un cercle noir avec un contour.

Le tableau ci-dessous nous donne le formalisme :

Tableau 7: Tableau de formalisme des éléments du diagramme d'activité

Signification	Formalisme
1. Le nœud représentant une action	
2. Le nœud objet	
3. Le nœud de décision ou de fusion	
4. Le nœud initial	
5. Le nœud final	
6. Symbole de record	

Voici le diagramme d'activité pour notre application :

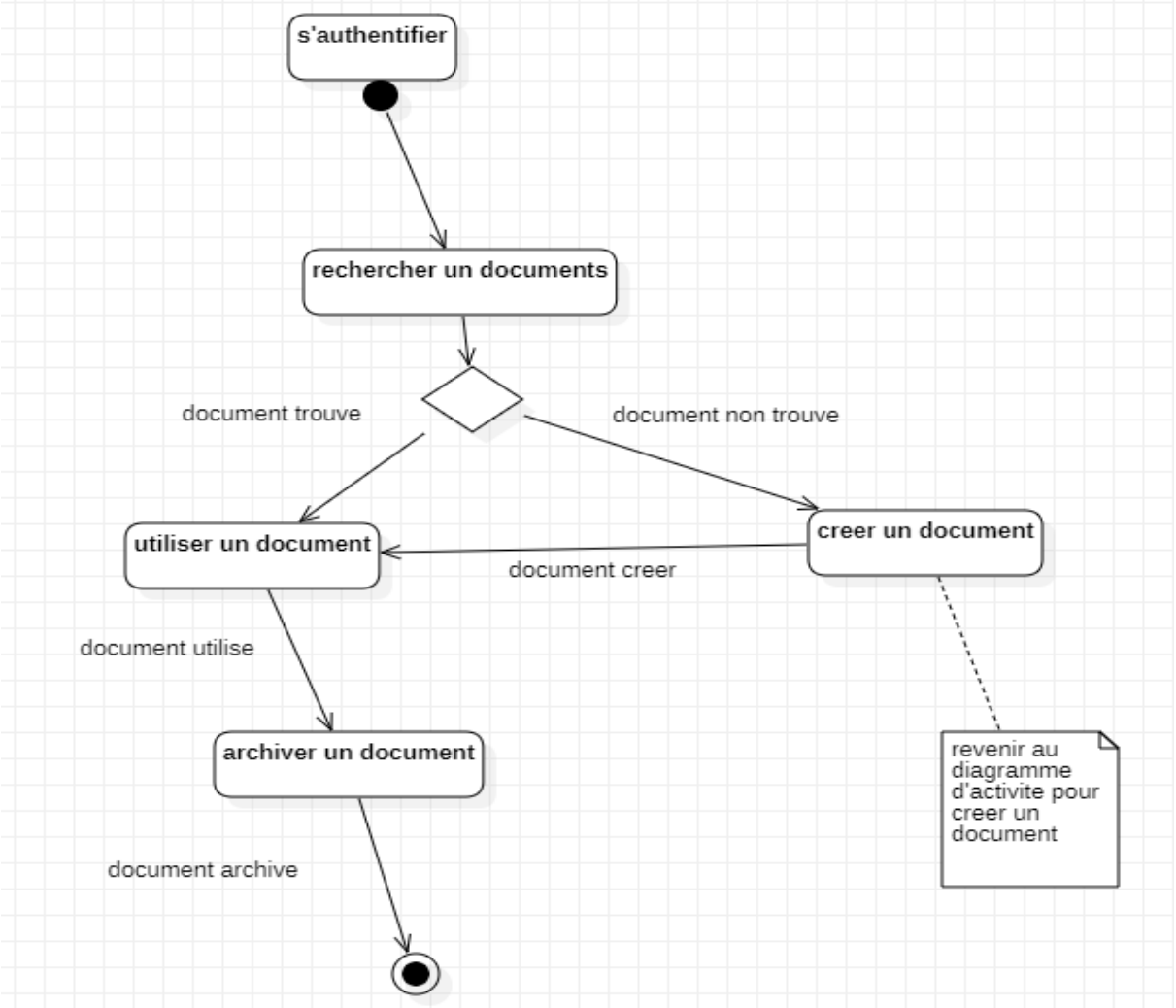


Figure 11: Diagramme d'activité de notre application

Ce diagramme d'activité illustre le flux de travail pour la gestion de documents dans un système. Il montre de manière séquentielle et conditionnelle les différentes étapes de la gestion de documents, depuis la recherche jusqu'à l'archivage ou la création.

Diagramme d'activité pour s'authentifier

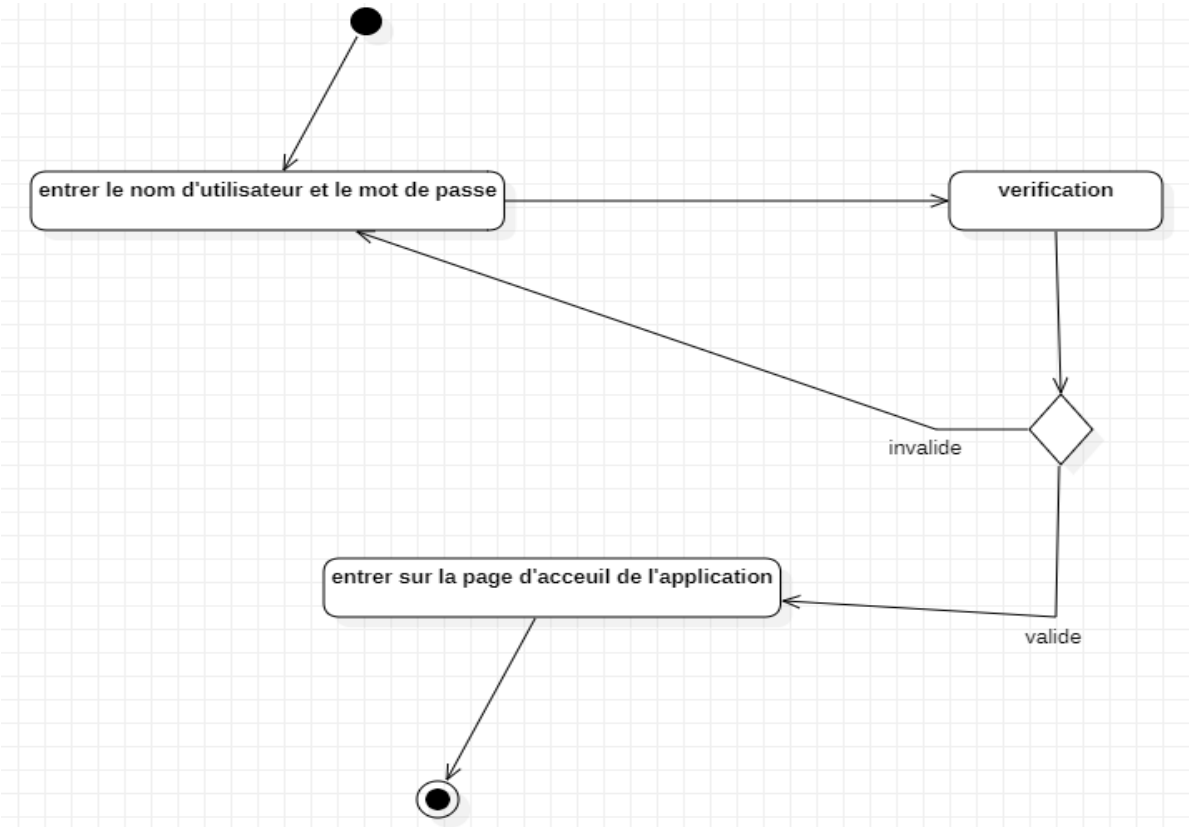


Figure 12: Diagramme d'activité du cas d'utilisation d'authentification

Ce diagramme d'activité décrit le processus d'authentification d'un utilisateur dans une application. Il montre clairement les étapes et les décisions prises pour authentifier un utilisateur, soulignant les flux en cas de succès ou d'échec de la vérification des informations d'authentification. Le tableau ci-dessus va nous montre une explication détaillée :

Tableau 8: Diagramme d'activité pour s'authentifier

Étape	Description
Début	L'utilisateur entre son nom d'utilisateur et son mot de passe.
Vérification	Les informations saisies par l'utilisateur sont vérifiées pour garantir leur exactitude.
Décision	Informations Valides : Si les informations sont correctes, l'utilisateur est redirigé vers la page d'accueil de l'application

	Informations Invalides : Si les informations sont incorrectes, le processus retourne à la saisie des informations d'authentification, permettant à l'utilisateur de réessayer.
Fin	Le processus se termine une fois que l'utilisateur a accédé à la page d'accueil de l'application.
Résumé	Entrée des Informations : L'utilisateur entre son nom d'utilisateur et son mot de passe.
	Si les informations sont valides, l'utilisateur accède à la page d'accueil.
	Si les informations sont invalides, l'utilisateur doit réessayer.
Accès à l'Application	Le processus se termine lorsqu'un accès valide est accordé.

III.3. Conclusion

Dans ce chapitre, nous avons présenté l'étape conception et analyse des besoins, dont nous avons décrit d'une façon détaillé la modélisation en utilisant le langage de modélisation « UML », ainsi que le processus unifié qui nous a aidé de faire une conception détaillée avec les déférents diagrammes. Nous avons aussi donné un petit détail sur les diagrammes couramment utilisés dans la modélisation des systèmes d'information.

CHAPITRE IV. MODELISATION MATHEMATIQUE AVEC MODELE RELATIONEL

La modélisation mathématique est le processus de création d'une représentation mathématique d'un système ou d'un processus. Elle est utilisée en science, en ingénierie et en finance pour faire des prédictions et simuler des systèmes complexes. Les modèles mathématiques peuvent être utilisés pour analyser le comportement d'un système dans différentes conditions et pour optimiser ses performances.

L'objectif de la modélisation est de créer une représentation visuelle d'un système ou d'un processus qui peut être utilisée pour analyser, améliorer et communiquer des informations. La modélisation permet d'identifier les problèmes et d'optimiser les performances. Elle sert à créer des simulations qui peuvent être utilisées pour tester et affiner les conceptions avant qu'elles ne soient construites. [11]Pour notre cas, nous allons utiliser l'algèbre relationnelle comme outil mathématiques de modélisation

IV.1. Modèle relationnel

IV.1.1. Définition

Inventé par Edgar Codd, l'algèbre relationnelle est 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. Il existe 2 types d'opérateurs algébriques: Opérateurs de base ou primitifs ; ainsi que des Opérateurs non essentiels ou dérivés.

C'est un Ensemble d'opérateurs s'appliquant sur l'ensemble des lignes (ou tuples) d'une (ou plusieurs) table(s). Le résultat d'une opération (ou d'une requête) est une nouvelle table qui est exploitable à son tour par une nouvelle opération.

IV.1.2. Justification du choix du modèle

Nous avons choisi d'utiliser l'algèbre relationnelle car cette dernière nous permettra d'écrire et de construire très facilement nos requêtes. La principale application de l'algèbre relationnelle est de fournir une base théorique pour les bases de données relationnelles, en particulier les langages de requête pour ces bases de données, dont le principal est SQL. Les bases de données relationnelles stockent des données tabulaires représentées sous forme de relations. Les requêtes sur les bases de données relationnelles renvoient souvent également des données tabulaires représentées sous forme de relations.

IV.2. Opérateurs de l'algèbre relationnelle

IV.2.1. Opérateurs unaires

Une opération relationnelle agit sur une ou plusieurs tables et a pour résultat une table .on peut citer : La projection et la sélection sont des opérations qui s'appliquent à une table ; Les opérations ensemblistes (union, intersection, différence) ne peuvent être utilisés qu'avec deux tables ayant les mêmes attributs et fournissent une troisième table ayant les mêmes attributs ; Le produit cartésien et la jointure fournissent une troisième table à partie de deux tables quelconque

1. Sélection

L'opérateur de sélection (σ) dans l'algèbre relationnelle est utilisé pour filtrer les tuples (enregistrements) d'une relation (table) en fonction d'une condition spécifique. Cela signifie que l'opérateur de sélection permet de sélectionner uniquement les tuples qui satisfont une condition donnée tout en excluant les autres. L'opérateur de sélection est particulièrement utile pour effectuer des requêtes ciblées sur une base de données relationnelle. En utilisant cet opérateur, vous pouvez extraire uniquement les informations qui correspondent à des critères spécifiques, ce qui permet de réduire la taille du résultat obtenu et de récupérer les données pertinentes pour une analyse ou une utilisation particulière. [14]

Voici comment l'opérateur de sélection est utilisé dans l'algèbre relationnelle

σ(condition)(relation)

σ : Symbole de l'opérateur de sélection ;

Condition : La condition ou la clause de recherche qui doit être satisfaite par les tuples que vous souhaitez sélectionner. Cette condition peut impliquer des opérateurs de comparaison (>, <, =, ≤, ≥, etc.), des opérateurs logiques (AND, OR, NOT), des opérateurs de quantification (POUR TOUT, EXISTE), etc ;

Relation : Le nom de la relation (table) sur laquelle vous souhaitez effectuer la sélection.

Exemple d'utilisation :

Sélectionner tous les documents reçus après une certaine date :

SELECT * FROM `documents` WHERE date_enregistrement ='2023-10-10'

Cette requête sélectionnera tous les documents entrants enregistres en date du 10 octobre 2023.

2. Projection

L'opérateur de projection (π) dans l'algèbre relationnelle sert à sélectionner certaines colonnes (attributs) d'une relation (table) tout en éliminant les autres. En d'autres termes, il permet de projeter une relation sur un sous-ensemble spécifique de ses attributs, ce qui conduit à une nouvelle relation avec une structure simplifiée contenant uniquement les colonnes requises.

L'opérateur de projection est utilisé pour réduire le nombre de colonnes dans le résultat d'une requête, ce qui peut être utile pour des raisons d'efficacité, de confidentialité ou pour se concentrer sur les informations pertinentes. Il permet également de créer des vues personnalisées sur les données en masquant certaines informations confidentielles ou en se concentrant sur des attributs spécifiques pour l'analyse.

Voici comment l'opérateur de projection est utilisé dans l'algèbre relationnelle:

π(attributs)(relation)

π : Symbole de l'opérateur de projection ;

Attributs : La liste des attributs que vous souhaitez conserver dans la nouvelle relation résultante. Vous pouvez spécifier un ou plusieurs attributs, séparés par des virgules ;

Relation : Le nom de la relation (table) sur laquelle vous souhaitez effectuer la projection.

Exemple de cas d'utilisation

Voici un cas d'utilisation pour l'opérateur de projection (π) :

Supposons que vous souhaitiez créer une vue personnalisée pour afficher uniquement l'identifiant du document et la date d'enregistrement des documents pour une recherche rapide des documents les plus récents :

CREATE VIEW VueDocumentsRécents AS

SELECT id_document, date_enregistrement

FROM documents

ORDER BY date_enregistrement DESC ;

Cette requête projettera la table "Document" sur les attributs "id-document" et "date-enregistrement", et le résultat sera une nouvelle relation avec seulement ces colonnes.

Le résultat obtenu pourrait ressembler à ceci :

Id-document	Date-enregistrement
DOC-001	2023-07-15
DOC-002	2023-07-14
DOC-002	2023-07-13
…	….

Avec cette vue personnalisée, vous pouvez rapidement visualiser les identifiants des documents et leurs dates d'enregistrement, ce qui vous permettra de repérer facilement les documents les plus récents. Cette projection peut être utile lorsque vous avez une grande quantité de documents archivés et que vous souhaitez rapidement identifier les documents enregistrés récemment pour une gestion plus efficace de l'archivage.

IV.2.2. Opérateurs n-aires

1. Union

L'opérateur de l'union (∪) dans l'algèbre relationnelle est utilisé pour combiner les tuples (enregistrements) de deux relations (tables) qui ont le même schéma (même nombre et types d'attributs). Cela permet de créer une nouvelle relation qui contient tous les tuples présents dans l'une ou l'autre des relations d'origine, tout en éliminant les duplicatas.L'opérateur de l'union est utilisé lorsque vous souhaitez combiner les enregistrements de deux tables ayant des données similaires ou complémentaires sans duplication. C'est l'équivalent de l'opération "OR" en langage naturel.Voici comment l'opérateur de l'union est utilisé dans l'algèbre relationnelle :

Relation1 ∪ relation2 ;∪ : Symbole de l'opérateur de l'union ;

relation1, relation2 : Les noms des deux relations que vous souhaitez combiner.

Exemple d'utilisation de l'union :

DocumentsEntrants_Semestre1 ∪ DocumentsEntrants_Semestre2

2. Intersection

L'opérateur d'intersection (∩) dans l'algèbre relationnelle est utilisé pour obtenir les tuples (enregistrements) qui se trouvent à la fois dans deux relations (tables) ayant le même schéma. En d'autres termes, l'opérateur d'intersection permet de trouver les enregistrements communs entre deux tables.Cet opérateur est utile lorsque vous souhaitez identifier les éléments qui sont présents à la fois dans deux ensembles de données, c'est-à-dire les enregistrements qui sont communs aux deux tables. Voici comment l'opérateur d'intersection est utilisé dans l'algèbre relationnelle:

Relation1 ∩ relation2

∩ : Symbole de l'opérateur d'intersection ;

relation1, relation2 : Les noms des deux relations que vous souhaitez comparer.

Exemple :

DocumentsEntrants_Semestre1 ∩ DocumentsEntrants_Semestre2

3. Différence

L'opérateur de différence (-) dans l'algèbre relationnelle est utilisé pour obtenir les tuples (enregistrements) qui se trouvent dans une relation (table) donnée mais ne sont pas présents dans une autre relation ayant le même schéma. En d'autres termes, l'opérateur de différence permet d'identifier les enregistrements qui sont exclusifs à la première relation, par rapport à la seconde.

Cet opérateur est utile lorsque vous souhaitez connaître les éléments qui sont présents dans une table mais absents dans une autre, ce qui peut être utile pour comparer des ensembles de données et identifier des différences entre eux.

Voici comment l'opérateur de différence est utilisé dans l'algèbre relationnelle:

Relation1 - relation2

-: Symbole de l'opérateur de différence ;

Relation1, relation2 : les noms des deux relations que vous souhaitez comparer

4. Produit cartésien

L'opérateur de produit cartésien (×) dans l'algèbre relationnelle permet de combiner tous les tuples de deux relations (tables) pour créer une nouvelle relation dont le nombre de tuples est le produit des nombres de tuples des deux relations d'origine. Cela signifie que chaque tuple de la première relation sera combiné avec chaque tuple de la seconde relation pour former le résultat du produit cartésien.

L'opérateur de produit cartésien est utilisé lorsque vous souhaitez combiner toutes les lignes de deux tables, indépendamment de tout critère de correspondance entre les tuples. Le résultat du produit cartésien peut être significativement plus grand que les relations d'origine, surtout lorsque les tables impliquées sont volumineuses.

Voici comment l'opérateur de produit cartésien est utilisé dans l'algèbre relationnelle :

Relation1 × relation2

× : Symbole de l'opérateur de produit cartésien ;

relation1, relation2 : Les noms des deux relations que vous souhaitez combiner.

5. Jointure

L'opérateur de jointure (⋈) dans l'algèbre relationnelle est utilisé pour combiner les tuples (enregistrements) de deux relations (tables) en fonction d'une condition de correspondance entre les attributs spécifiés. Cela permet de créer une nouvelle relation qui contient les informations combinées des deux tables d'origine, en fonction de la condition de jointure.

L'opérateur de jointure est utilisé lorsque vous avez besoin de combiner des données de deux tables qui partagent un attribut commun, et vous souhaitez récupérer des informations de ces tables pour les enregistrements qui satisfont la condition de jointure. C'est l'équivalent de l'opération "AND" en langage naturel.

Voici comment l'opérateur de jointure est utilisé dans l'algèbre relationnelle :

Relation1 ⋈ condition relation2

⋈ : Symbole de l'opérateur de jointure ;

Condition : La condition de correspondance entre les attributs des deux tables. Cette condition spécifie comment les tuples des deux relations doivent être combinés.

relation1, relation2 : Les noms des deux relations que vous souhaitez joindre.

Pour utiliser l'opérateur de jointure entre les tables "Classe" et "AnnéeAcademique", vous devez avoir un attribut commun (clé étrangère) entre ces deux tables. Cette clé étrangère permettra de lier les enregistrements des deux tables qui correspondent à la même classe pour une année académique spécifique.

Supposons que la table "Classe" a les attributs suivants : "IDClasse" (identifiant unique de la classe), "NomClasse" (nom de la classe), et "Niveau" (niveau de la classe). La table "AnnéeAcademique" a les attributs suivants : "IDAnnée" (identifiant unique de l'année académique), "AnnéeDébut" (année de début) et "AnnéeFin" (année de fin) pour chaque année académique.

Exemple d'utilisation de la jointure :

Supposons que vous souhaitiez obtenir une vue qui combine les informations de la table "Classe" avec les informations de l'année académique à laquelle chaque classe appartient.

La jointure serait réalisée sur la base d'une clé étrangère "IDAnnée" dans la table "Classe" correspondant à l'attribut "IDAnnée" dans la table "AnnéeAcademique".

Voici comment vous pouvez effectuer la jointure :

Classe ⋈ Classe.IDAnnée = AnnéeAcademique.IDAnnée AnnéeAcademique

Cette requête joint les deux tables "Classe" et "Année Académique" en fonction de la clé étrangère "IDAnnée".

Le résultat serait une nouvelle relation qui contient les informations combinées des classes avec les informations de l'année académique correspondante.

Le résultat obtenu pourrait ressembler à ceci :

IDClasse	NomClasse	Niveau	IDAnnée	AnnéeDébut	AnnéeFin
1	Classe A	1ère	100	2021	2022
2	Classe B	2ème	1002	2022	2023
…	…	…	….	….	….

Ainsi, la jointure vous permet de récupérer les informations des classes en les associant à l'année académique à laquelle elles appartiennent, ce qui facilite l'analyse et la gestion des classes dans votre système académique.

5.1. Jointure naturelle

La jointure naturelle est une opération de jointure en algèbre relationnelle qui combine automatiquement les enregistrements de deux tables en utilisant toutes les colonnes ayant le même nom dans les deux tables comme critère de jointure. Cela signifie que la jointure naturelle relie les enregistrements des deux tables lorsque les valeurs dans les colonnes partagent le même nom.

Plus spécifiquement, lorsqu'une jointure naturelle est effectuée entre deux tables, le système de gestion de base de données (SGBD) compare les noms des colonnes dans les deux tables et identifie les colonnes ayant des noms identiques. Il utilise ensuite ces colonnes avec les mêmes noms pour établir la correspondance entre les enregistrements. [9]

Il est important de noter que la jointure naturelle ne nécessite pas que vous spécifiiez explicitement les colonnes à utiliser pour la jointure. Cependant, cela peut parfois être risqué, car si les colonnes ont des noms identiques mais ne devraient pas être utilisées pour la jointure, les résultats de la jointure peuvent être inattendus. Il est donc essentiel de comprendre la structure de vos données avant d'utiliser une jointure naturelle.

5.2. Jointure externe

En algèbre relationnelle, la jointure externe (ou outer join en anglais) est une opération de jointure qui combine les enregistrements de deux tables en fonction d'un critère de jointure, tout en incluant également les enregistrements qui ne correspondent pas à ce critère. Elle permet d'inclure les enregistrements qui n'ont pas de correspondance dans l'autre table dans le résultat de la jointure.

Il existe trois principaux types de jointures externes :

Jointure externe gauche (LEFT OUTER JOIN) : la jointure externe gauche retourne toutes les lignes de la table de gauche (première table spécifiée) et les lignes correspondantes de la table de droite (deuxième table spécifiée). Si une ligne de la table de gauche n'a pas de correspondance dans la table de droite, les colonnes de la table de droite seront remplies de valeurs NULL dans le résultat.

Jointure externe droite (RIGHT OUTER JOIN) : la jointure externe droite est l'inverse de la jointure externe gauche. Elle retourne toutes les lignes de la table de droite et les lignes correspondantes de la table de gauche. Si une ligne de la table de droite n'a pas de correspondance dans la table de gauche, les colonnes de la table de gauche seront remplies de valeurs NULL dans le résultat.

Jointure externe complète (FULL OUTER JOIN) : la jointure externe complète retourne toutes les lignes des deux tables, avec des correspondances là où elles existent. Si une ligne n'a pas de correspondance dans l'autre table, les colonnes de la table sans correspondance contiendront des valeurs NULL dans le résultat.

La jointure externe est souvent utilisée lorsqu'il est nécessaire de récupérer des données de deux tables tout en incluant les enregistrements qui peuvent ne pas avoir de correspondance dans l'autre table. Elle est utile pour analyser les données de manière plus complète et pour

présenter les informations de manière significative, même lorsque certaines données manquent dans l'une des tables.Voici un exemple de jointure externe gauche en SQL :

```
SELECT *
FROM Table1
LEFT OUTER JOIN Table2 ON Table1.ID = Table2.ID;
```

Dans cet exemple, "Table1" est la table de gauche et "Table2" est la table de droite. La jointure est basée sur la colonne "ID", et toutes les lignes de "Table1" sont incluses, ainsi que les correspondances de "Table2" ou des valeurs NULL si aucune correspondance n'est trouvée.

5.3. Jointure interne

L'opérateur de jointure interne (join) fusionne deux relations en produisant toutes les concaténations possibles de leurs n-uplets respectifs qui vérifient une certaine condition.

La jointure interne est équivalente à un produit cartésien suivi d'une sélection : P joincond q ≡ σscond (p × q)

Ainsi, la jointure interne (INNER JOIN en anglais) est une opération de jointure qui combine les enregistrements de deux tables en fonction d'un critère de jointure spécifié. Elle ne renvoie que les enregistrements qui ont une correspondance dans les deux tables, en utilisant ce critère comme base de la jointure.

Voici les caractéristiques principales de la jointure interne :

Correspondance des enregistrements : Seuls les enregistrements qui ont une correspondance dans les deux tables en fonction de la condition de jointure spécifiée seront inclus dans le résultat ;

Critère de jointure explicite : Vous devez spécifier explicitement le critère de jointure en indiquant quelle colonne ou quelles colonnes des deux tables seront utilisées pour établir la correspondance ;

Élimination des non-correspondances : Les enregistrements qui n'ont pas de correspondance dans l'autre table sont exclus du résultat de la jointure.

Exemple simple de jointure interne en SQL :

```
SELECT *
FROM Table1
INNER JOIN Table2 ON Table1.ID = Table2.ID;
```

Dans cet exemple, "Table1" et "Table2" sont les deux tables que nous voulons joindre. La jointure est basée sur la colonne "ID", ce qui signifie que seules les lignes ayant le même "ID" dans les deux tables seront incluses dans le résultat.

La jointure interne est largement utilisée pour combiner des données provenant de différentes tables en ne conservant que les enregistrements qui correspondent aux critères spécifiés. C'est l'un des types de jointure les plus couramment utilisés dans le langage SQL pour interroger une base de données relationnelle.

5.4. Equijointure et Thêta-jointure

L'équijointure et la thêta-jointure sont deux types de jointures utilisées en algèbre relationnelle pour combiner des enregistrements de deux tables en fonction de certains critères.

Voici des explications détaillées pour chacun de ces types de jointures :

Equijointure (Equi-Join) :l'équi-jointure est un type de jointure dans lequel les enregistrements des tables sont combinés en fonction d'une condition d'égalité entre les colonnes spécifiées ; La condition d'égalité est souvent exprimée à l'aide de l'opérateur "=". Par exemple, "table1.colonne = table2.colonne" ;

L'équi-jointure permet de lier les enregistrements des deux tables où les valeurs des colonnes sélectionnées sont égales ; Elle est souvent utilisée pour rechercher des correspondances entre les enregistrements basés sur une relation égalitaire.

Thêta-Jointure (Thêta-Join) : la thêta-jointure est un type de jointure plus général que l'équi-jointure ; elle permet de spécifier une condition de jointure arbitraire basée sur n'importe quel opérateur logique (comme "=", ">", "<", etc.) ; La condition peut être une égalité, une inégalité, une comparaison, etc., et elle est spécifiée par l'utilisateur en fonction de ses

besoins spécifiques ;La thêta-jointure offre plus de flexibilité, car elle permet de spécifier des conditions de jointure complexes autres que l'égalité.

Exemple d'équi-jointure en SQL :

```
SELECT *
FROM Table1
INNER JOIN Table2 ON Table1.ID = Table2.ID;
```

Dans cet exemple, l'équi-jointure est basée sur l'égalité entre les colonnes "ID" de "Table1" et "Table2". Seuls les enregistrements ayant le même "ID" dans les deux tables seront inclus dans le résultat.

Exemple de thêta-jointure en SQL :

```
SELECT *
FROM Table1, Table2
WHERE Table1.Age > Table2.Age;
```

Dans cet exemple, la thêta-jointure est basée sur la condition "Table1.Age > Table2.Age". Cela signifie que seules les lignes pour lesquelles l'âge dans "Table1" est supérieur à l'âge dans "Table2" seront incluses dans le résultat.

En résumé, l'équi-jointure est une jointure basée sur l'égalité, tandis que la thêta-jointure permet de spécifier des conditions de jointure arbitraires en utilisant différents opérateurs logiques.

6. Renommage

Le renommage en algèbre relationnelle est une opération qui consiste à attribuer un nouveau nom (alias) à une relation ou à un attribut dans une relation. Cela permet de rendre les requêtes ou les expressions plus lisibles et compréhensibles sans changer la structure ou les données sous-jacentes. Le renommage est souvent utilisé dans les requêtes SQL et dans les opérations algébriques pour améliorer la clarté du code.

IV.3. Fonctions d'agrégation

Une fonction d'agrégation (ou fonction de regroupement) s'applique aux valeurs du sous-ensemble d'un agrégat en relation avec pour résultat la production d'une valeur atomique unique (entier, chaîne, date, etc.)

Les cinq fonctions prédéfinies sont : [22]

COUNT (Relation. Propriété) Renvoie le nombre de valeurs non nulles d'une propriété pour tous les tuples d'une relation ;

SUM (Relation. Propriété) Renvoie la somme des valeurs d'une propriété des tuples (numériques) d'une relation ;

AVG (Relation. Propriété) Renvoie la moyenne des valeurs d'une propriété des tuples (numériques) d'une relation ;

MIN (relation. Propriété) Renvoie la plus petite valeur d'une propriété parmi les tuples d'une relation.

MAX (relation. Propriété) Renvoie la plus grande valeur d'une propriété parmi les tuples d'une relation.

Exemple d'utilisation

Supposons que nous avons une table "Documents" dans notre application d'archivage des documents, avec les attributs suivants : "ID Document" (identifiant unique pour chaque document), "Type Document" (le type du document, par exemple : facture, lettre, rapport), "Date Enregistrement " (la date à laquelle le document a été enregistré), et "Taille" (la taille du document en octets).Voici quelques exemples d'utilisation de fonctions d'agrégation pour obtenir des informations agrégées à partir de la table "Documents" :

Calculer le nombre total de documents enregistrés dans la base de données :

SELECT COUNT (*) AS Total Documents FROM Documents;

Cette requête utilise la fonction d'agrégation COUNT(*) pour compter le nombre total de documents dans la table "Documents" et affiche le résultat sous le nom d'alias "Total Documents".

Calculer la taille totale de tous les documents enregistrés :

SELECT SUM(Taille) AS TailleTotale FROM Documents;

Cette requête utilise la fonction d'agrégation SUM(Taille) pour calculer la somme des tailles de tous les documents enregistrés dans la table "Documents" et affiche le résultat sous le nom d'alias "TailleTotale".

Trouver la date la plus récente d'enregistrement des documents :

SELECT MAX (Date Enregistrement) AS DatePlusRecente FROM Documents;

Cette requête utilise la fonction d'agrégation MAX(DateEnregistrement) pour trouver la date la plus récente d'enregistrement des documents dans la table "Documents" et affiche le résultat sous le nom d'alias "DatePlusRecente".

Trouver la taille moyenne des documents enregistrés :

SELECT AVG(Taille) AS TailleMoyenne FROM Documents;

Cette requête utilise la fonction d'agrégation AVG(Taille) pour calculer la taille moyenne des documents enregistrés dans la table "Documents" et affiche le résultat sous le nom d'alias "TailleMoyenne".

Ces exemples d'utilisation de fonctions d'agrégation vous permettent d'obtenir rapidement des informations agrégées sur les documents archivés dans votre application d'archivage des documents, ce qui facilite l'analyse et la prise de décisions en fonction des statistiques et des métriques obtenues.

IV.4. Conception de nos requêtes en Algèbre Relationnelle

Nous pouvons présenter des requêtes SQL avec leurs expressions équivalentes en algèbre relationnel de la manière suivante :

Sélection :

Nous voulons rechercher la liste les documents entrants archivés pour une telle date, la commande que nous utiliserons est :

SELECT * FROM `documents` WHERE date_enregistrement='2023-10-10 21:47:10'

Nous voulons afficher la liste de tous les utilisateurs, alors nous allons utiliser la requête suivante :

SELECT*FROM Utilisateurs

Jointure

Requête SQL utilisant la jointure entre la table option et la table département:

SELECT *FROM département INNER JOIN option ON departement.id_departement = option.id_departement ;

Dans cette requête, nous utilisons la clause JOIN pour relier les enregistrements de la table "département" avec les enregistrements correspondants de la table "option" en utilisant la clé étrangère "id_Option". La condition WHERE filtre les résultats en ne montrant que les classes ayant l'option "génie informatique" en fonction du champ "Nom_Option" de la table "option". Assurez-vous que les noms des colonnes et des tables correspondent à votre schéma de base de données pour que la requête fonctionne correctement.

Projection

Requête SQL pour afficher la liste des utilisateurs des départements d'une faculté : SELECT `nom';' prénom` ;`email` ;`phone` ;`date_creation` FROM `utilisateur`

IV.5. Conclusion

Dans ce chapitre, j'ai présenté quelques requêtes que j'ai utilisées pour le bon fonctionnement de l'application surtout dans les bases de données ainsi que l'importance d'utiliser le modèle relationnel comme outil de modélisations.

Je n'ai pas pu évoquer toutes les requêtes du langage SQL que j'ai utilisé mais j'ai cité quelques-unes.

CHAPITRE V. REALISATION DE L'APPLICATION

V.1.Présentation des outils de développement

1. Sublime text

Sublime Text est un éditeur de code multiplateforme principalement utilisé pour la création de sites Web. Il offre de nombreuses fonctionnalités utiles qui facilitent et accélèrent le codage.Cet outil offre aux utilisateurs de sélectionner des colonnes entières en même temps ou de placer plusieurs curseurs sur plusieurs lignes, ce qui permet d'éditer ou d'insérer des codes simultanément. Une fonctionnalité très utile lorsque vous insérez des codes répétitifs sur plusieurs lignes.

La complétion automatique est également une autre fonctionnalité de Sublime qui propose de compléter les entrées. Il agit comme un assistant de texte prédictif dans la plupart des applications basées sur du texte.La nature multi-plateforme de Sublime Text vous permet d'exécuter des codes pour différents langages sans avoir à basculer vers la ligne de commande et revenir.L'application sauvegarde automatiquement vos projets pour vous éviter de perdre votre travail en cas de panne du système ou de courant.Vous pouvez également attribuer des raccourcis clavier pour un fonctionnement plus rapide et même configurer des macros. Le logiciel dispose d'un outil de correction intégré spécialement conçu pour identifier les erreurs de codage. Son interface claire inclut une fonction Minimap qui vous permet de prévisualiser votre code dans une barre latérale.

L'application prend en charge les langages de programmation JavaScript, C, C++, C#, LaTeX, Perl, PHP, Ruby, CSS, SQL, XML et XLS.Sublime Text est disponible sous Windows, macOS et Linux.

2. Xamp

XAMPP est un ensemble de logiciels open source qui fournit un environnement de développement web complet pour créer et tester des applications web localement sur votre ordinateur. XAMPP est souvent utilisé par les développeurs pour créer des sites web et des applications web avant de les déployer sur des serveurs en ligne.

Le "X" dans XAMPP signifie qu'il est multiplateforme, ce qui signifie qu'il peut être utilisé sur différents systèmes d'exploitation tels que Windows, macOS et Linux.

Le "AMPP" dans XAMPP représente les composants principaux inclus dans le package :

Apache : Le serveur web Apache est inclus dans XAMPP pour gérer les requêtes HTTP et servir les pages web ;

MySQL : Il s'agit d'un système de gestion de base de données relationnelle qui permet de stocker, gérer et manipuler les données ;

PHP : PHP est un langage de script côté serveur largement utilisé pour créer des pages web dynamiques. Il est utilisé pour interagir avec les bases de données et générer du contenu dynamique ;

Perl : Perl est un langage de script polyvalent qui peut être utilisé pour diverses tâches de développement web ;

XAMPP Control Panel : C'est une application qui permet de gérer et de contrôler les composants d'un environnement XAMPP, tels que le démarrage et l'arrêt des serveurs Apache et MySQL.

En utilisant XAMPP, les développeurs peuvent créer un environnement de développement local pour tester leurs sites web et applications sans avoir besoin d'un hébergement en ligne. Cela facilite le processus de développement, de débogage et de test. Une fois que les développements locaux sont terminés, les sites web et les applications peuvent être transférés vers des serveurs en ligne pour une utilisation en production.XAMPP est un outil populaire pour les débutants et les développeurs expérimentés qui cherchent à créer et tester des applications web sur leur propre machine avant de les déployer sur Internet.

3. PHP

PHP (Hypertext Preprocessor) est un langage de programmation de scripts côté serveur conçu spécialement pour le développement d'applications web dynamiques. Créé à l'origine en 1994 par Rasmus Lerdorfs, PHP est aujourd'hui l'un des langages de programmation les plus largement utilisés pour la création de sites web interactifs et de nombreuses autres applications en ligne. Le PHP a évolué au fil des ans et continue d'être une technologie essentielle pour le développement web. C'est un excellent choix pour créer des applications web dynamiques, des sites e-commerce, des blogs, des forums et bien plus encore

4. JavaScript

JavaScript est un langage de programmation populaire et largement utilisé pour le développement web. Il permet d'ajouter de l'interactivité, de la dynamique et des fonctionnalités avancées aux pages web. JavaScript est principalement utilisé côté client, ce qui signifie qu'il s'exécute dans le navigateur web de l'utilisateur, mais il peut également être utilisé côté serveur avec des environnements tels que Node.js.[21]

5. Bootstrap

Bootstrap est un Framework open source très populaire pour le développement web front-end. Il a été créé par Twitter et est conçu pour faciliter la conception et le développement de sites web réactifs et esthétiquement agréables. Bootstrap fournit une collection de composants prêts à l'emploi, de styles CSS et de scripts JavaScript pour créer rapidement des interfaces utilisateur attrayantes et adaptables à différents dispositifs et tailles d'écran. En utilisant Bootstrap, les développeurs peuvent accélérer le processus de conception et de développement, ce qui en fait un choix populaire pour la création d'interfaces utilisateur réactives et professionnelles.

6. HTML

HTML (HyperText Markup Language) est le langage de balisage standard utilisé pour créer des pages web. Il définit la structure et l'organisation du contenu sur une page web en utilisant des balises et des éléments. HTML est la base fondamentale du développement web et est utilisé pour décrire la présentation et la sémantique des documents web.

HTML joue un rôle essentiel dans la création de l'infrastructure et de la base d'une page web. En utilisant des balises et des éléments, les développeurs peuvent structurer le contenu, insérer des médias, créer des liens et définir des formulaires interactifs, entre autres.

7. MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) open source très populaire. Il est largement utilisé pour stocker et gérer des données dans une variété d'applications, y compris les sites web, les applications d'entreprise et les applications logicielles. Voici quelques caractéristiques clés de MySQL :

Stockage de données : MySQL permet de stocker des données structurées dans des tables. Les tables sont organisées en lignes et colonnes, créant une structure relationnelle ;

Langage SQL : MySQL utilise le langage SQL (Structured Query Language) pour effectuer des opérations de manipulation de données, telles que la sélection, l'insertion, la mise à jour et la suppression.

Gestion des transactions : MySQL prend en charge les transactions, ce qui permet d'assurer l'intégrité des données lors de l'insertion, la mise à jour ou la suppression de données ;

Indexation : Il offre des fonctionnalités d'indexation pour améliorer les performances des requêtes, en accélérant la recherche et la récupération des données ;

Sécurité : MySQL propose des fonctionnalités de sécurité pour protéger les données, y compris l'authentification des utilisateurs, le contrôle d'accès et le chiffrement ;

Requêtes avancées : MySQL prend en charge des requêtes avancées, telles que les jointures, les sous-requêtes, les agrégats et les fonctions pour effectuer des opérations complexes sur les données ;

Réplication et haute disponibilité : MySQL permet la réplication de données pour maintenir des copies synchronisées de la base de données sur plusieurs serveurs, améliorant ainsi la disponibilité et la tolérance aux pannes ;

Gestion des données volumineuses : MySQL peut gérer des bases de données de grande taille et de nombreux enregistrements, en offrant des performances optimisées ;

Stockage JSON : Il prend en charge le stockage et la manipulation de données au format JSON (JavaScript Object Notation), ce qui permet de travailler avec des données semi-structurées ;

Communauté active : MySQL dispose d'une grande communauté d'utilisateurs et de développeurs qui partagent des ressources, des astuces et des solutions.

MySQL est utilisé dans une variété d'applications, allant des petits sites web personnels aux grandes entreprises et aux applications en ligne à fort trafic. Il offre une solution fiable et évolutive pour la gestion des données, avec des options de personnalisation et d'optimisation pour répondre aux besoins spécifiques de chaque projet.

8. Codeigniter

Codeigniter est un framework de développement d'applications web open source, écrit en PHP, qui simplifie et accélère le processus de création d'applications web interactives et dynamiques. Il est conçu pour être léger, rapide et flexible, ce qui en fait un excellent choix pour les développeurs souhaitant créer des applications web performantes.CodeIgniter est conçu pour être léger en termes de taille et de performances, ce qui le rend adapté aux petites et grandes applications et suit le modèle Modèle-Vue-Contrôleur (MVC), qui sépare la logique de traitement des données (Modèle), la présentation (Vue) et le contrôle de l'application (Contrôleur).Il fournit une gamme de bibliothèques préconstruites (librairies) et de fonctions d'assistance (helpers) pour effectuer des tâches courantes, telles que la manipulation de formulaires, la gestion des sessions, l'envoi de courriels, etc.

Aussi, CodeIgniter permet une personnalisation facile grâce à son approche basée sur la configuration. Vous pouvez choisir les composants à utiliser en fonction des besoins spécifiques de votre application. CodeIgniter est souvent choisi par les développeurs pour sa simplicité, sa flexibilité et sa rapidité d'exécution. Il est particulièrement adapté aux projets de taille moyenne qui nécessitent des fonctionnalités MVC et une gestion de la base de données, tout en minimisant la surcharge.

V.2. Sécurité de l'application

V.2.1. Introduction

Avec la montée en puissance de la technologie numérique, la sécurité informatique est devenue une préoccupation majeure pour les individus, les entreprises et les organisations du monde entier. La sécurité informatique est un domaine essentiel de la technologie moderne qui vise à protéger les systèmes informatiques, les réseaux, les données et les informations contre les menaces, les attaques et les accès non autorisés.

La sécurité informatique englobe un large éventail de mesures, de pratiques et de technologies conçues pour assurer la confidentialité, l'intégrité et la disponibilité des données et des ressources informatiques. Elle vise à protéger les informations sensibles, à prévenir les perturbations des services, à détecter les activités malveillantes et à répondre efficacement aux incidents de sécurité.

V.2.2. Importance de la sécurité des données

La sécurité informatique est un domaine essentiel de la technologie moderne qui vise à protéger les systèmes informatiques, les réseaux, les données et les informations contre les menaces, les attaques et les accès non autorisés. Avec la montée en puissance de la technologie numérique, la sécurité informatique est devenue une préoccupation majeure pour les individus, les entreprises et les organisations du monde entier.

La sécurité informatique englobe un large éventail de mesures, de pratiques et de technologies conçues pour assurer la confidentialité, l'intégrité et la disponibilité des données et des ressources informatiques. Elle vise à protéger les informations sensibles, à prévenir les perturbations des services, à détecter les activités malveillantes et à répondre efficacement aux incidents de sécurité.

Voici les huit principes génériques de la sécurité des données informatiques :

Identification : Information permettant d'indiquer qui vous prétendez être. Une identification élémentaire est le nom d'utilisateur que l'on saisit dans un système informatique. Une identification plus évoluée peut-être le relevé d'empreinte digitale, l'analyse fiscale, rétinienne, etc. [22]

Authentification : Information permettant de valider l'identité pour vérifier que vous êtes celui que vous prétendez être. Une authentification élémentaire est le mot de passe que vous entrez dans le système informatique. Une authentification forte combine une chose que vous possédez et une chose que vous connaissez, (numéro de cane bancaire et code personnel, par exemple).

Autorisation : Information permettant de déterminer quelles seront les ressources de l'entreprise auxquelles l'utilisateur identifié et autorisé aura accès ainsi que les actions autorisées sur ces ressources. Cela couvre toutes les ressources de l'entreprise.

Confidentialité : Ensemble des mécanismes permettant qu'une communication de données reste privée entre un émetteur et un destinataire. La cryptographie ou le chiffrement des données est la seule solution fiable pour assurer la confidentialité des données.

Intégrité : Ensemble des mécanismes garantissant qu'une information n'a pas été modifiée.

Disponibilité : Ensemble des mécanismes garantissant que les ressources de l'entreprise sont accessibles, que ces derniers concernent l'architecture réseau, la bande passante, le plan de sauvegarde, etc.

Non répudiation : Mécanisme permettant de garantir qu'un message a bien été envoyé par un émetteur et reçu par un destinataire.

Traçabilité : Ensemble des mécanismes permettant de retrouver les opérations réalisées sur les ressources de l'entreprise. Cela suppose que tout événement applicatif soit archivé pour investigation ultérieure.

V.2.3. Protection de la base de données

V.2.3.1. Gestion des utilisateurs

La gestion des utilisateurs est le processus contrôlant les utilisateurs autorisés à se connecter au serveur MySQL et définissant les permissions d'accès sur chaque base de données. PhpMyAdmin ne gère pas les utilisateurs : il passe le nom et le mot de passe utilisateur à MySQL, qui détermine alors si un utilisateur est autorisé à réaliser une action particulière. Dans phpMyAdmin, les administrateurs ont tous les droits pour créer, voir et éditer les privilèges ou supprimer les utilisateurs. [23]

Ainsi, l'administrateur peut créer, modifier ou même supprimer un utilisateur.

Les requêtes SQL que nous avons utilisé pour réaliser ces opérations sont :

Pour créer un utilisateur : CREATE USER 'nom_utilisateur" IDENTIFIED BY 'mot_de_passe';

Voici ce que signifient les différents éléments de cette requête :

Nom utilisateur : Remplacez cela par le nom d'utilisateur que vous souhaitez créer.

mot_de_passe : Remplacez cela par le mot de passe souhaiter pour l'utilisateur.

Pour supprimer un utilisateur: DROP USER 'nom utilisateur';

On peut aussi modifier un utilisateur et un mot de passe avec la requête SQL suivante : ALTER USER 'nom_utilisateur' IDENTIFIED BY 'nouveau_mot_de_passe'

V.2.3.2. Gestion des privilèges

Les privilèges sont des droits pour exécuter des requêtes.

Pour accorder à l'utilisateur récemment créé tous les privilèges pour la base de données, exécutez la commande suivante : GRANT ALL PRIVILEGES ON 'base_de_donnees'TO 'nom utilisateur'

Cette commande donne à l'utilisateur 'nouveau_utilisateur' tous les privilèges sur toutes les bases de données. Cela inclut les privilèges tels que SELECT, INSERT, UPDATE, DELETE, ainsi que des privilèges plus élevés comme ALTER, CREATE, DROP, etc.

Vous devriez accorder les privilèges de manière judicieuse en fonction des besoins spécifiques de l'utilisateur pour des raisons de sécurité.Ainsi tout comme on peut attribuer des privilégiés a un utilisateur, on peut aussi donner l'ordre pour supprimer grâce à la requête REVOKE.Ce dernier est utilisé de la manière suivante : REVOKE ALL PRIVILEGES ON 'base_de_donnees' FROM 'nom_utilisateur'

V.2.3.3. Fonctions de hachage

Les fonctions de hachage sont des algorithmes qui prennent en entrée des données de taille variable et produisent une valeur de longueur fixe, généralement une séquence de caractères hexadécimaux. Ces fonctions sont couramment utilisées dans divers domaines de l'informatique, notamment la sécurité, les bases de données, la cryptographie et la gestion des données.

Les fonctions de hachage permettent à partir d'une entrée, de calculer une empreinte unique [24]. Les fonctions de hachages ont diverses applications, notamment la possibilité de calculer l'empreinte unique d'un fichier ou de stocker les mots de passe de manière protégé dans une base de données. On peut aussi l'utiliser pour vérifier la somme de contrôle (checksum) d'une donnée pour vérifier l'intégrité. Il existe plusieurs algorithmes différents qui permettent d'obtenir une empreinte comme par exemple md5, sha1, sha-256

Pour notre cas, nous avons utilisé MD5.

V.3. Présentation de l'application

L'application d'archivage que j'ai développé possède plusieurs interfaces mais je présenterai quelques-unes jugées importantes

1. **Interface d'authentification**

Une interface d'authentification est la partie de l'application qui permet aux utilisateurs de s'identifier ou de se connecter de manière sécurisée. Elle joue un rôle essentiel dans la sécurité et l'accès aux fonctionnalités de l'application.

Cette interface contient l'adresse mail de l'utilisateur et son mot de passe.

La figure ci-dessous nous montre l'interface d'authentification ou identification de notre système :

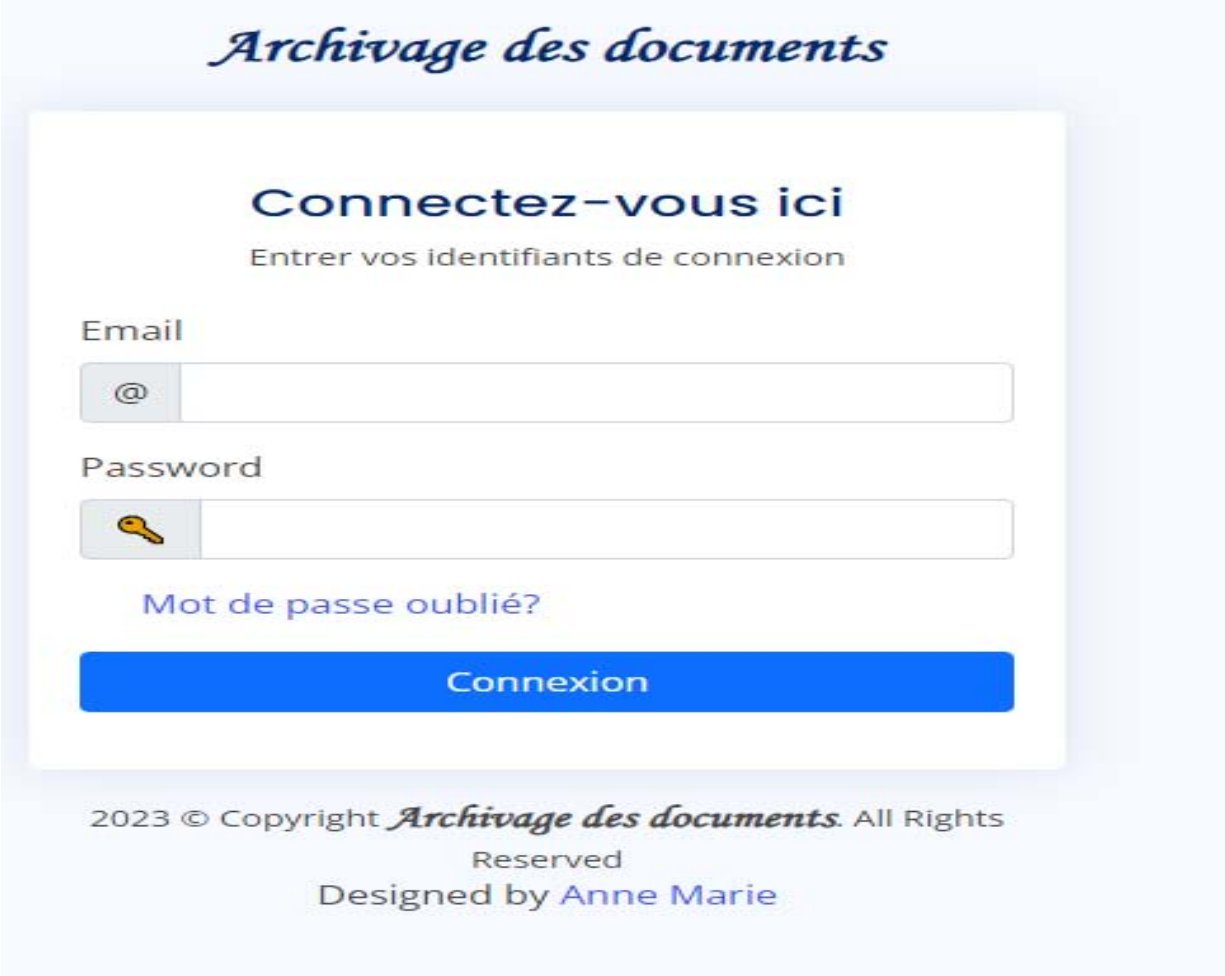


Figure 13: identification d'authentification

L'utilisateur doit entrer son adresse e-mail dans ce champ. Cette adresse est utilisée pour identifier l'utilisateur de manière unique puis entre un mot de passe associé à son compte. Ce mot de passe permet de sécuriser l'accès et de vérifier l'identité de l'utilisateur ;

Apres avoir entre les identifiants, le bouton "Connexion" envoie les informations saisies dans les champs d'e-mail et de mot de passe au serveur pour validation.

Lorsqu'un utilisateur clique sur "Connexion", une requête est envoyée vers le serveur avec les informations d'identification (email et mot de passe).Le serveur compare ces informations avec celles stockées dans sa base de données pour vérifier si elles correspondent à un utilisateur existant. Une fois les données validées le serveur vérifie d'abord que l'e-mail existe dans la base de données. Si oui, il compare le mot de passe fourni avec celui qui est stocké (généralement sous forme chiffrée, pour des raisons de sécurité)

Si les informations sont correctes, l'utilisateur est redirigé vers une page protégée de l'application, lui permettant d'accéder à ses fonctionnalités ;

Si l'une des informations est incorrecte (e-mail ou mot de passe), le serveur renverra un message d'erreur indiquant que les identifiants ne sont pas valides.

Si l'utilisateur a oublié son mot de passe, le lien « mot de passe oublie » est généralement utilisé. En cliquant dessus, il est redirigé vers une page où il peut entrer son adresse e-mail pour recevoir un lien de réinitialisation de mot de passe ;

Le bas de l'interface contient des informations telles que les droits d'auteur et le concepteur de l'application. Ces informations sont courantes pour indiquer la propriété et les droits de l'application ;

Une application d'authentification doit être protégée contre des attaques comme l'injection SQL, le cross-site scripting (XSS), et doit utiliser le chiffrement SSL/TLS pour garantir que les informations de connexion (en particulier le mot de passe) ne sont pas transmises en clair sur le réseau.

De plus, les mots de passe doivent être chiffrés (hachés) dans la base de données en utilisant des algorithmes sécurisés comme bcrypt ou Argon2.

En résumé, cette interface permet de sécuriser l'accès à une application d'archivage de documents en demandant aux utilisateurs de s'authentifier avant d'accéder à leurs documents ou fonctionnalités.

2. Interface d'accueil pour notre système

Cette interface montre toutes les fonctionnalités et composantes de notre système :

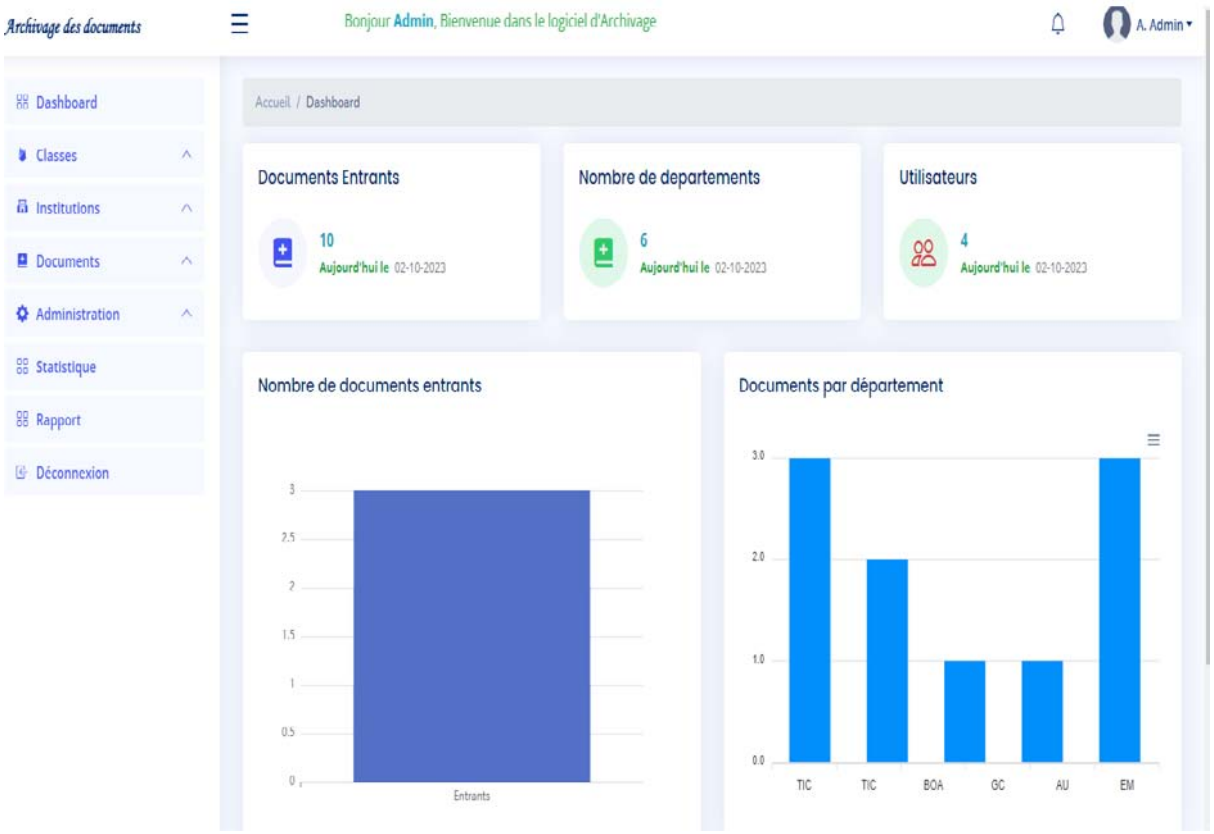


Figure 14: interface d'accueil

Cette interface représente un tableau de bord (Dashboard) pour un logiciel d'archivage de documents. Elle offre une vue d'ensemble des activités et des statistiques liées à l'archivage des documents, permettant aux administrateurs de suivre facilement les informations clés et d'accéder aux différentes sections du système.

Voici une brève explication des différents éléments de cette interface :

Le tableau de bord principal offre une vue d'ensemble intuitive avec plusieurs sections clés : Classes, pour gérer les catégories de documents, et Institutions, pour administrer les établissements associés. La section documents permet d'accéder et de gérer les archives, tandis que l'onglet administration centralise les options du système. Une zone statistique présente les analyses, et l'onglet rapport facilite la génération de rapports détaillés.

En haut à droite, un message de bienvenue est affiché pour l'administrateur, accompagné des options de profil utilisateur et d'un bouton de déconnexion. Cette organisation garantit une navigation fluide et une gestion efficace des fonctionnalités

3. Interface pour ajouter un document

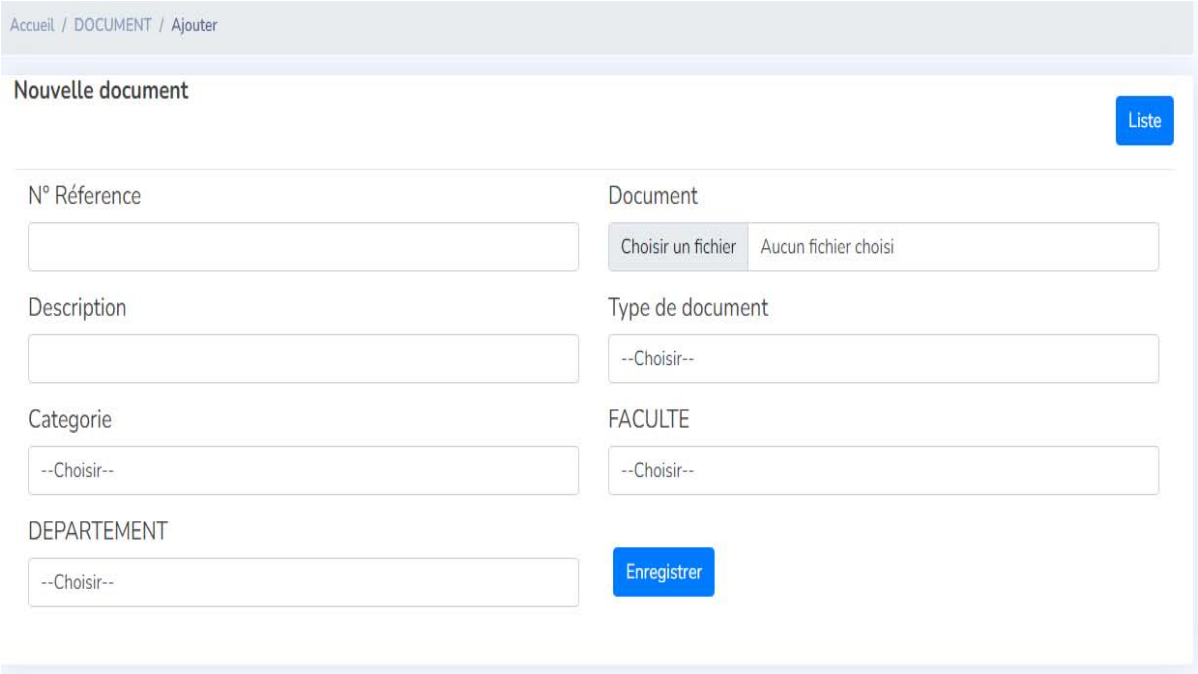


Figure 15: interface pour ajouter un document

Cette fenêtre permet l'ajout d'un document scanné se trouvant dans le disque, c'est-à-dire un document scanné se trouvant dans l'ordinateur. Le secrétaire réceptionniste ou même l'administratif peuvent effectuer ces opérations.

Elle va permettre aux utilisateurs l'ajout de nouveaux documents à une base de données. Elle est divisée en plusieurs sections pour faciliter la saisie des informations nécessaires. Elle est conçue pour simplifier l'ajout de nouveaux documents en fournissant une structure claire et un processus intuitif. Elle assure que toutes les informations pertinentes sont correctement saisies et que les documents sont bien organisés et facilement accessibles dans la base de données.

Comme dans notre application on va archiver les documents et lettres, l'interface ci-dessous va nous montre comment l'ajout (archivage) des lettres se fait au sein de la FSI :

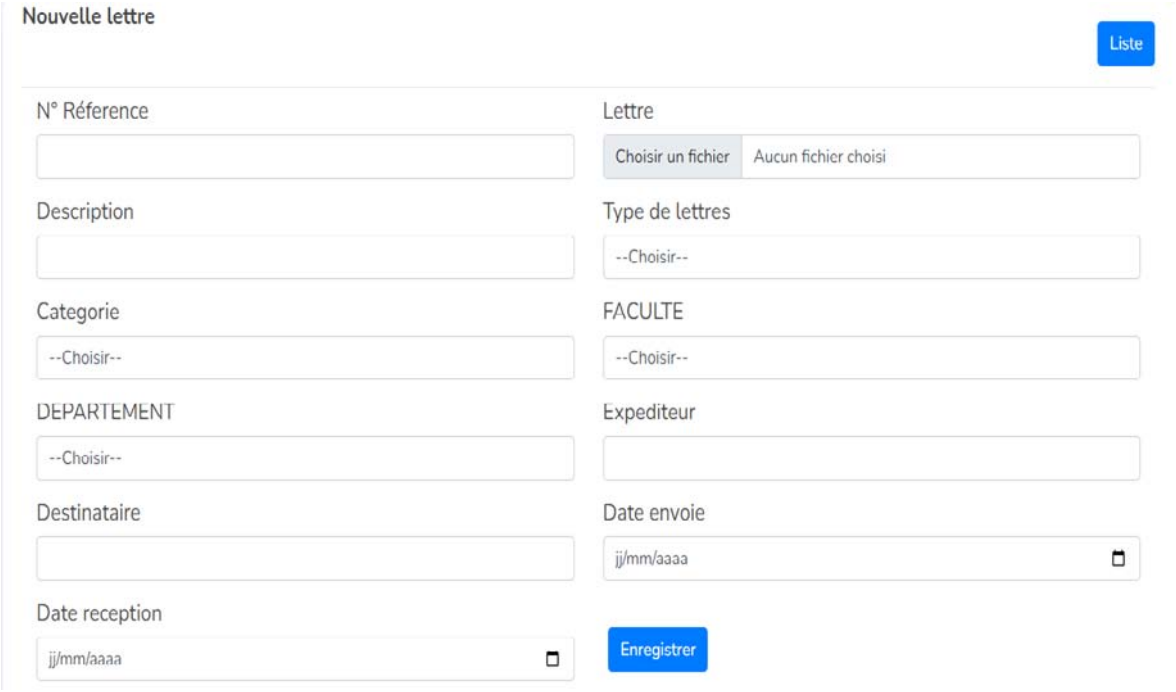


Figure 16: interface pour ajouter une lettre

Le processus de gestion des lettres permet à l'utilisateur de saisir et d'organiser efficacement les informations. Chaque lettre est identifiée par un N° Référence unique, accompagné d'un fichier représentant la lettre, sélectionné et téléchargé via un bouton dédié. Une brève description peut être ajoutée pour contextualiser son contenu. Les menus déroulants permettent de définir le type, la catégorie, la faculté, et le département associés. Les champs permettent également de spécifier l'expéditeur, le destinataire, ainsi que les dates d'envoi et de réception. Une fois le formulaire rempli et validé, un clic sur le bouton Enregistrer soumet les données au serveur. Le fichier est stocké, les métadonnées sont enregistrées dans la base de données, et une confirmation de succès est affichée, garantissant une gestion fluide et organisée des lettres.

Cette interface permet aux utilisateurs d'ajouter de nouvelles lettres à une base de données. L'interface est divisée en plusieurs sections pour faciliter la saisie des informations nécessaires.

5. Interface de tous les documents entrants archivés

Dans cette partie, on va afficher les documents en fonction de l'utilisateur :Les utilisateurs doivent être crée dans la base de données et doivent avoir un certain nombre de droits et privilégiés donnes par l'admin du système.

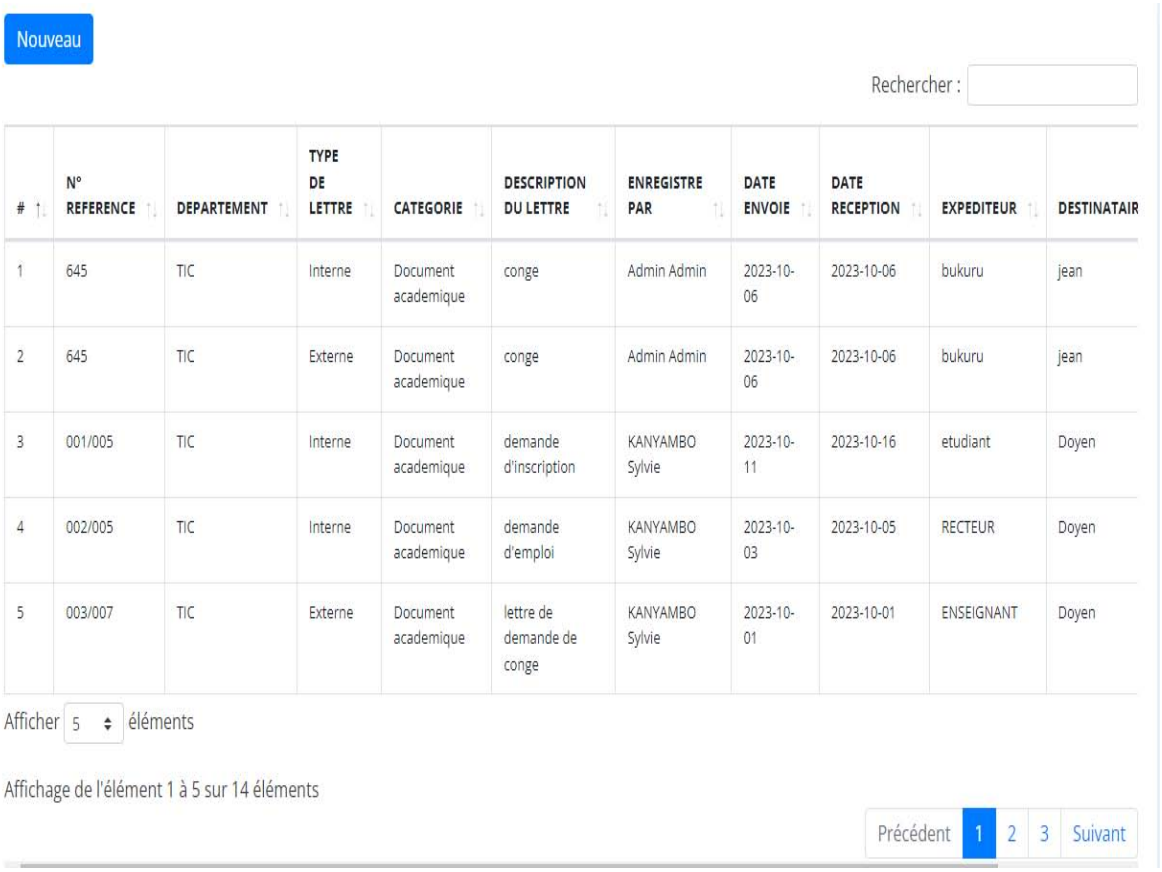


Figure 17: interface pour toutes les lettres archivées

La barre de navigation et les boutons d'action facilitent la gestion des lettres et documents. Le bouton "Nouveau" redirige l'utilisateur vers un formulaire pour ajouter une nouvelle lettre. Un champ de recherche permet de filtrer rapidement les lettres ou documents par mots-clés. Le tableau des lettres affiche les informations essentielles dans des colonnes bien définies : N° Référence, Département, Type de Lettre (Interne ou Externe), Catégorie, Description, Enregistré par, Date d'Envoi, Date de Réception, Expéditeur, et Destinataire. La pagination offre une navigation fluide entre les pages de résultats, tandis que des fonctionnalités permettent l'ajout, la modification, la suppression, ainsi que l'exportation des données sous divers formats (CSV, PDF, etc.). Ce système garantit une gestion claire et efficace des informations archivées.

2) Interface des documents entrants archivés pour le Doyen

Comme l'admin, le doyen a aussi un certain nombre de droits et privilèges sur notre système. Il peut voir la liste de tous les documents archivés ainsi que l'utilisateur qui les a archivés. Il peut également modifier, supprimer ou même télécharger un document. L'ensemble des droits et privilèges est octroyé par l'admin lors de la gestion des utilisateurs.

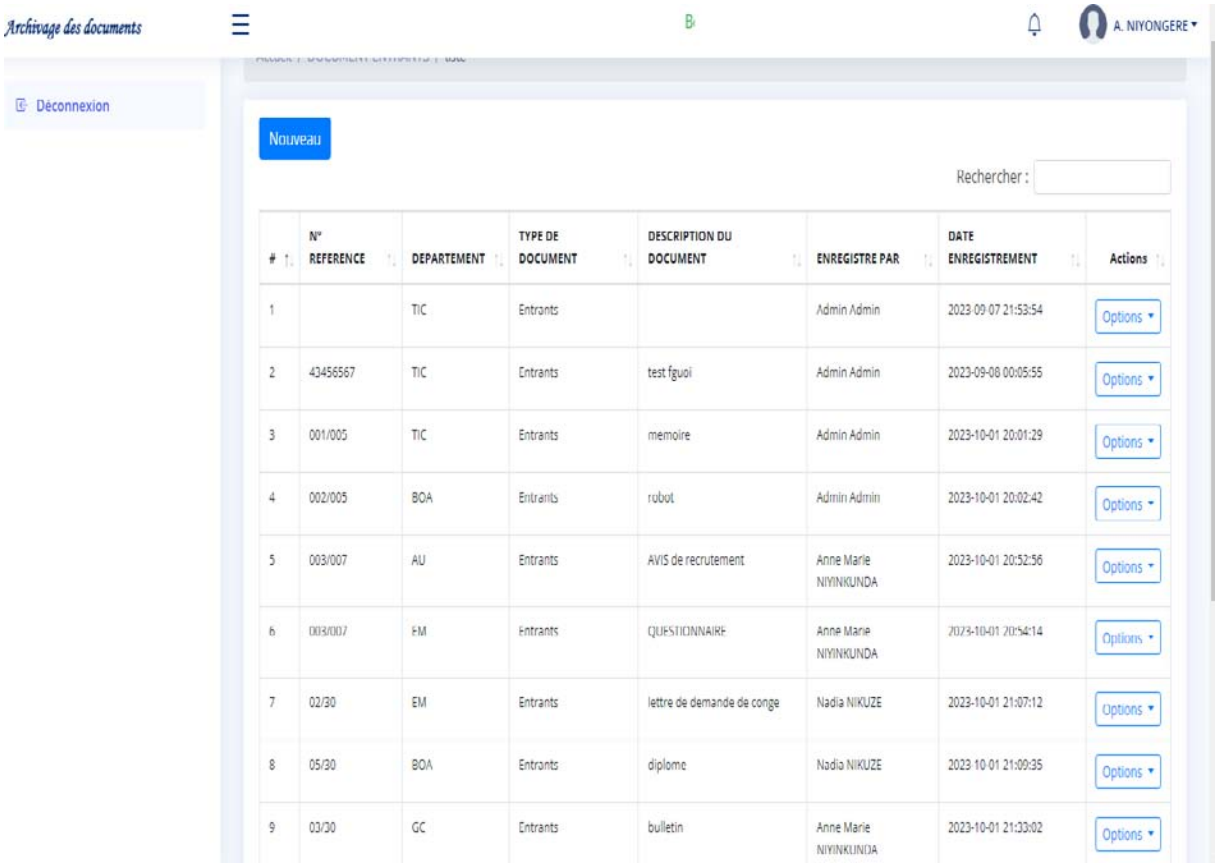


Figure 18: Interface des documents entrants archivés par le Doyen de la FSI

Cette interface représente un tableau de gestion des documents entrants, archivés par le doyen de la faculté des Sciences de l'Ingénieur. Le système de gestion des documents fonctionne de manière simple et intuitive.

Un menu de navigation, situé à gauche, propose des options comme"Déconnexion" et facilite l'accès aux différentes sections. Le bouton "Nouveau" permet d'ajouter un document entrant via un formulaire dédié. Les documents sont affichés dans un tableau structuré avec des colonnes telles que : N° Référence (identifiant unique), département, type de document,

description, enregistré par, date d'enregistrement, et une colonne actions pour interagir avec chaque document (modifier, supprimer, voir les détails).

Un champ de recherche offre un filtrage rapide par mots-clés, tandis que la pagination permet de naviguer entre les pages de résultats. En arrière-plan, les contrôleurs gèrent les requêtes, les modèles interagissent avec la base de données, et des validations assurent l'intégrité des données.

Enfin, un système de gestion des droits garantit que seules les personnes autorisées peuvent effectuer des modifications ou des suppressions, renforçant ainsi la sécurité.

En résumé, cette interface permet une gestion efficace des documents entrants avec des fonctionnalités de navigation, d'ajout, de recherche, et de manipulation des documents.

3) Interface des documents entrants archivés pour le secrétaire administratif

Le Secrétaire administratif peut aussi archiver, modifier, télécharger, supprimer un document entrant :

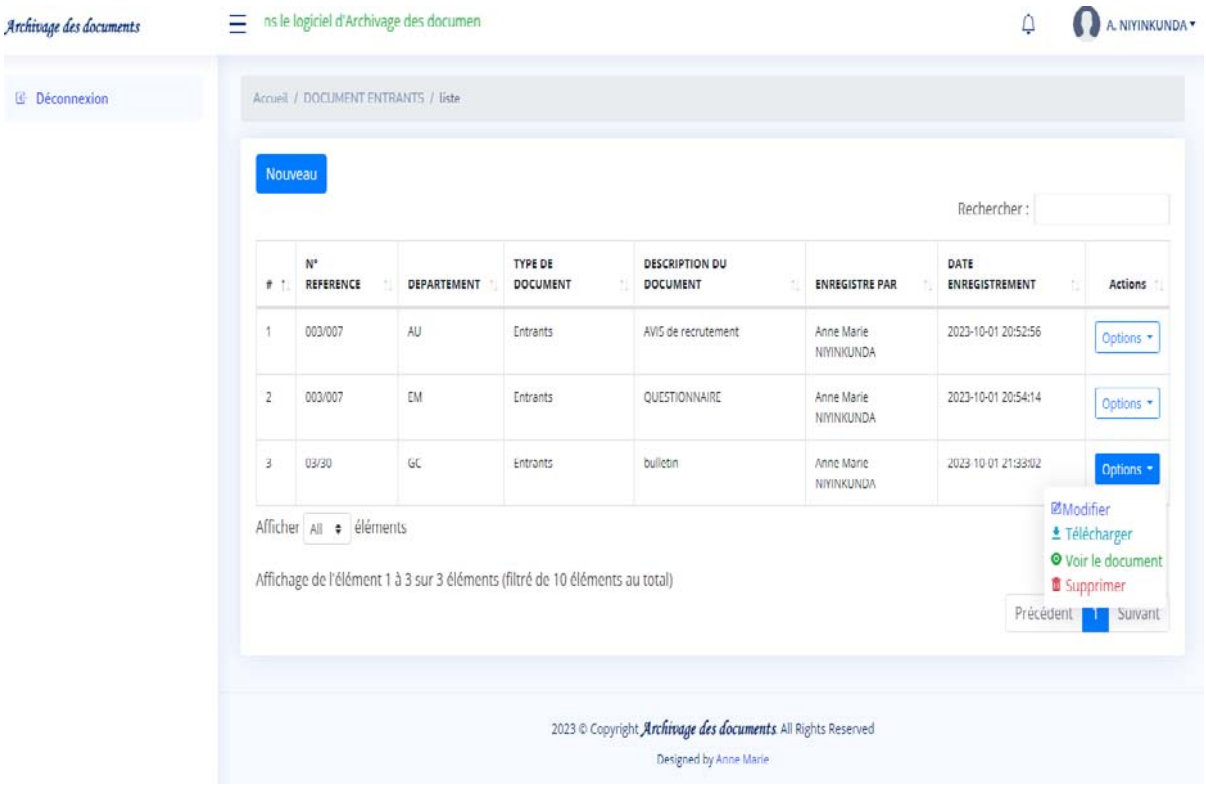


Figure 19: Interface des documents archivés par le secrétaire administratif

Pour ajouter un document à archiver, on va cliquer sur nouveau puis remplir le formulaire d'ajout et soumettre ce dernier afin d'enregistrer le document dans la base de données.

Ainsi, pour modifier un document, cliquez sur "Options" puis sur "Modifier" pour un document spécifique. Cela Apporte les modifications nécessaires dans le formulaire de modification afin de soumettre le formulaire pour mettre à jour les informations du document.Concernant le téléchargement, il suffit de cliquer sur"Options" puis sur "Télécharger" pour télécharger le document et pour voir le document, Cliquez sur "Options" puis sur "Voir le document" pour visualiser le contenu du document etc.

4) Interface des documents entrants archivés pour le secrétaire simple:

Comme dans notre faculté nous avons beaucoup de département, on peut avoir un secrétaire pour chaque faculté comme par exemple un secrétaire pour le département des TIC.

La figure ci-dessus nous montre l'interface des documents entrants archivés par ce dernier :

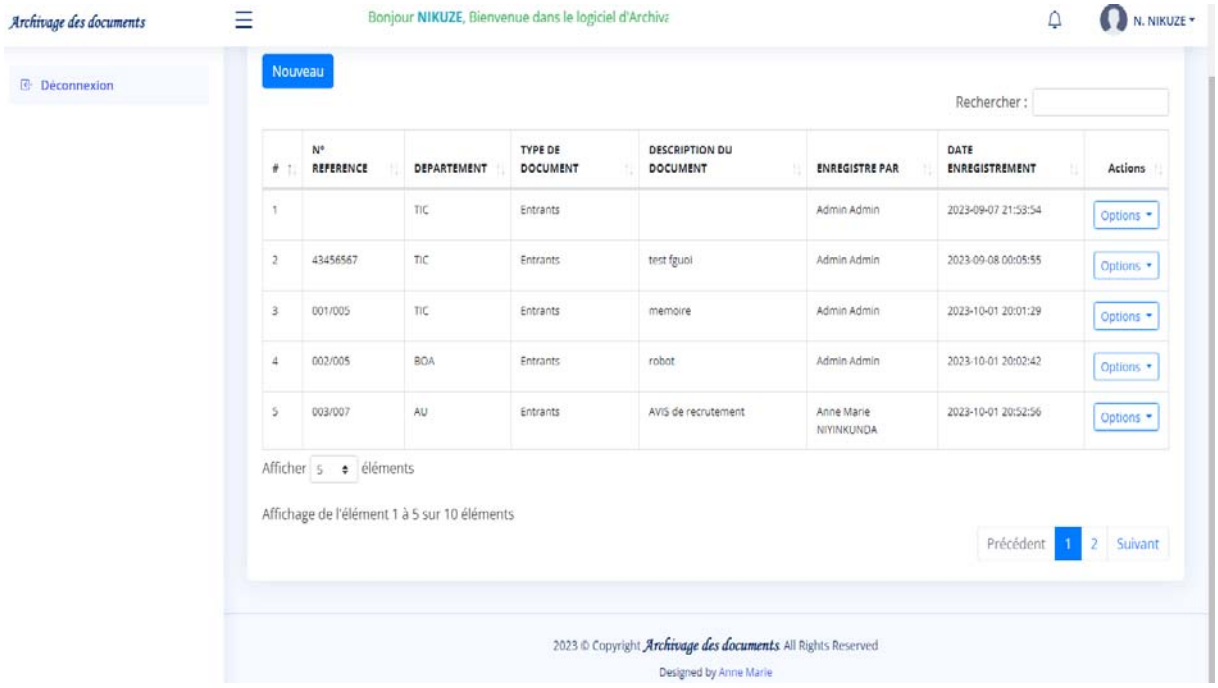


Figure 20: Interface des documents archivés par le secrétaire simple

L'application offre un système d'authentification qui exige que l'utilisateur se connecte pour accéder aux fonctionnalités. Le nom de l'utilisateur connecté est affiché en haut à droite, accompagné d'un bouton "Déconnexion". La navigation est facilitée par un menu latéral permettant d'accéder aux différentes sections.

Les documents enregistrés sont présentés dans un tableau détaillé, affichant des informations clés comme le numéro de référence, le département, le type de document, la description, l'enregistré par, et la date d'enregistrement. Une barre de recherche et une pagination permettent de filtrer et de naviguer efficacement parmi les documents. Les actions possibles incluent la modification, le téléchargement, la consultation, et la suppression de documents. Un bouton "Nouveau" redirige vers un formulaire pour ajouter de nouveaux documents, avec la possibilité de contrôler le nombre de documents affichés par page grâce à des options de pagination.

En résumé, cette interface permet de gérer l'archivage des documents en offrant des fonctionnalités pour ajouter, modifier, télécharger, voir et supprimer des documents, tout en facilitant la navigation et la recherche.

5. Interface de gestion des utilisateurs

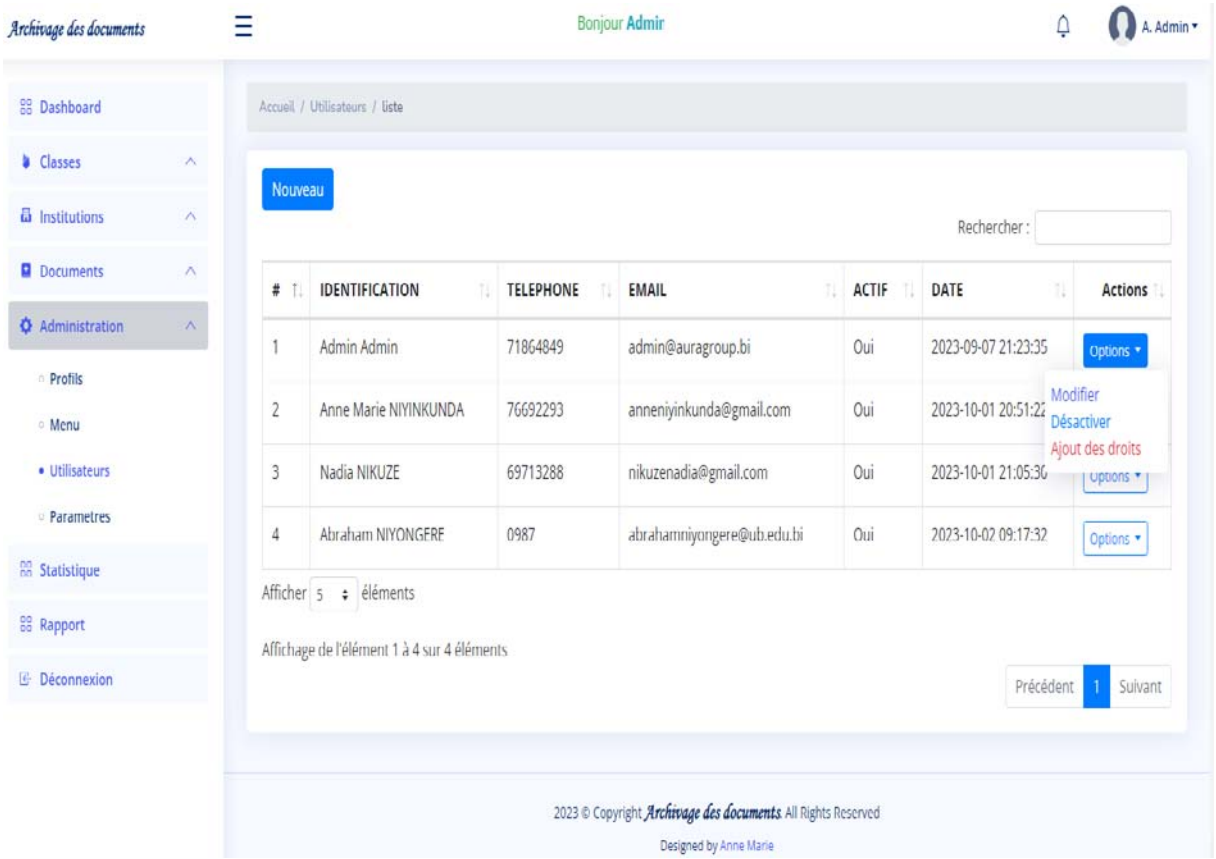


Figure 21: Interface de gestion des utilisateurs

Cette interface permet une gestion efficace des utilisateurs avec des fonctionnalités de navigation, d'ajout, de modification, de désactivation, et de gestion des droits.

Elle utilise des contrôleurs et des modèles pour interagir avec la base de données et des vues pour afficher et gérer les interactions utilisateur.

L'application dispose d'un menu de navigation sous forme de barre latérale, comprenant des sections principales telles que Dashboard, Classes, Institutions, Documents, Administration, Statistiques, Rapport, et Déconnexion.

La section Administration propose des sous-sections comme profils, menu, utilisateurs, et paramètres. Un tableau de gestion des utilisateurs affiche les informations essentielles, notamment : N° (numéro séquentiel), Identification (nom), Téléphone, Email, Actif (statut d'activité), et Date d'enregistrement. Un bouton "Nouveau" permet d'ajouter un utilisateur via un formulaire, tandis qu'un champ de recherche facilite le filtrage des utilisateurs. Les actions disponibles incluent la modification, la désactivation, et l'ajout de droits via un menu déroulant. La pagination gère efficacement l'affichage des utilisateurs sur plusieurs pages si nécessaire.

En backend, les contrôleurs assurent la gestion des requêtes CRUD, les modèles interagissent avec la base de données, et les vues affichent les données à l'utilisateur. Les fonctionnalités supplémentaires incluent la validation des données pour garantir leur intégrité et la gestion des droits d'accès pour renforcer la sécurité.

CONCLUSION GENERALE ET RECOMMANDATIONS

Conclusion générale

La mise en place d'un système automatisé d'archivage des documents entrants au sein de la Faculté des Sciences de l'Informatique (FSI) est une avancée cruciale pour moderniser les processus administratifs, améliorer la gestion de l'information et l'efficacité globale de la faculté. Cette initiative est essentielle pour répondre aux besoins croissants du FSI en matière de traitement des documents et d'accès rapide aux données.

Elle positionne la faculté à l'avant-garde de la technologie de gestion documentaire, créant ainsi une base solide pour l'avenir.

La mise en œuvre de ce système offre une série d'avantages considérables. Tout d'abord, il permettra de gérer de manière efficace et efficiente les flux de documents entrants, notamment les tâches manuelles fastidieuses et minimisant les risques d'erreur humaine. En outre, ce système favorisera une de la satisfaction des utilisateurs grâce à un accès simplifié aux informations et une réduction des délais de traitement des documents.

La conformité aux réglementations en matière de gestion de documents sera renforcée, minimisant ainsi les risques juridiques potentiels.

Enfin, ce système offre la possibilité de mieux gérer les ressources humaines en déchargeant le personnel des tâches administratives routinières et en les redirigeant vers des activités à plus forte valeur ajoutée.

Recommandations

Nous suggérons à l'Université du BURUNDI et aux futurs chercheurs :

- **A l'Université du BURUNDI :**

Adopter le nouveau système mis en place pour faciliter le processus d'archivage des documents administratifs et académiques ;

Mobiliser aux employeurs de l'UB à utiliser cette plateforme pour résoudre certaines erreurs ;

Encourager l'utilisation interdisciplinaire : Le système peut être utilisé dans d'autres facultés ou disciplines pour harmoniser la gestion des documents et améliorer l'efficacité administrative dans toute l'université.

- **Aux futurs chercheurs,**

Les étudiants-chercheurs intéressés par l'informatique peuvent étudier le fonctionnement du système et proposer des améliorations, comme l'intégration de fonctionnalités d'intelligence artificielle ou la mise en place d'une interface encore plus intuitive.

Grâce à ce système, les étudiants peuvent apprendre à organiser et structurer leurs propres travaux de manière à éviter les duplications d'efforts et à adopter des normes standardisées pour le traitement des informations.

Le système peut être un outil précieux pour les étudiants qui souhaitent accéder facilement aux documents académiques et administratifs sans perte de temps. Cela réduira les délais liés aux processus manuels.

REFERENCES BIBLIOGRAPHIQUES

[1]https://www.cybermedian.com/fr/unified-modeling-language-uml-introduction/#Concepts_cles_orientes_objet_dans_UML consulté le 02/07/2023 à 8h

[2] Un guide complet sur 14 types de diagramme UML - Cybermédiane (cybermedian.com) consulté le 05/07/2023 à 14h50min

[3] https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/ consulté le 08/07/2023 à 10h40min

[4] "UML 2.5 Specification" by the Object Management Group (OMG). Consulté le 27/07/2023 à 12h27min

[5]https://commentouvrir.com/info/types-de-modelisation-un-guide-complet consulté le 20/8/2023 à 16h30 min

[6]https://www.youtube.com/watch?v=bxBjAcQD2gs&list=PLp0vdGXHONe2DngJK-GuB22ZhfvvAFTHA&index=5 consulte le 21/07/2023 à 10h16min

https://fr.wikipedia.org/wiki/Alg%C3%A8bre_relationnelle Consulté le 27/07/2023 à 12h27min

[7] https://www.ibm.com/fr-fr/topics/data-security consulté le 04/08/2023 à 12h25min

[8] https://webusers.i3s.unice.fr/~edemaria/cours/c3.pdf consulté le 04/08/2023 à 14h42min

[9]https://sgbd.developpez.com/tutoriels/cours-complet-bdd-sql/?page=algebre-relationnelle#LVI-D-6-a consulté le 10/08/2023 à 10h22min

[10]https://books-library.net/files/books-library.net-11282031Aa9S6.pdf consulté le 15/08/2023 à 8h30min

[11]https://sgbd.developpez.com/tutoriels/cours-complet-bdd-sql/?page=algebre-relationnelle consulté le 20/08/2023 à 21h30

[12] https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-programmation-orientee-objet/ consulté le 23/08/2023 à 10h32min.

[13] M. NEMICHE, Faculté Polydisciplinaire de Ouarzazate Filière : Informatique et Gestion d'Entreprises Edition 2012-2013. Consulté le 25/08/2023 à 20h00min.

[14] Maude Manouvrier, Licence Mathématiques et Informatique - 3ème année Edition2022-2023 consulté le 30/08/2023 à 9h30min.

[15] IUT de Nice, Concepts et langages des Bases de Données Relationnelles

[16] P. Roques. – UML 2 par la pratique. N°12322, 6e édition, 2008 consulté 03/09/2023 à 12h22min.

[17] The Exploring on University Archives Management System Based on Information System: Zuheng Lv and Heyuzi Shi 2020 J. Phys.: Conf. Ser. 1550 032017 consulté le 07/09/2023 à 14h43min.

[18]www.ub.edu.bi consulté le 10/09/2023 à 16h15min.

[19] https://fsi.ub.edu.bi/a-propos-de-la-faculte/ consulté le 15/09/2023 à 19h32min.

[20] Stéphane Crozat, Paternité - Partage des Conditions Initiales à l'Identique : http://creativecommons.org/licenses/by-sa/4.0/fr consulté le 20/09/2023 à 23h35min.

[21] Mozilla Developer Network (MDN) consulté le 25/09/2023 à 20h39min.

[22]https://www.cours-gratuit.com/cours-reseau/cours-sur-les-principes-de-securite-informatique#remosito ryfilelisting Consulté le 25/09/2023 à 10h

[23] https://docs.phpmyadmin.net/fr/latest/privileges.html Consulté le 28/09/2023 à 16h 42min

[24] https://www.malekal.com/hash-md5-sha1-sha256/ Consulté le 30/09/2023 à 18h 02min