

RÉPUBLIQUE DU SÉNÉGAL



Un peuple - un but - une foi

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR, DE LA RECHERCHE ET DE
L'INNOVATION

XXXXXXXXXXXXXXXXXXXXXXXXXXXX



L'excellence ma référence

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Mention : Management des systèmes d'informations Automatisés

Département : Économie Gestion

UFR : Science Économique Sociale

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

SUJET :

**Etude et conception d'un système d'information de
gestion des heures complémentaires des
enseignements : le cas de l'UASZ**

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Présenté par :

Bakanding GOUDIABY

Sous la direction de :

Professeur Edouard Ngor SARR

Membres du jury :

Professeur Amon Anike DEH (**Président**)

Docteur Abel DIATTA (**Examineur**)

Docteur Bala Moussa BIAYE (**Examineur**)



Dédicace

Je dédie ce travail à ma mère **Marietou Badji.**

Remerciement

J'adresse mes sincères remerciements à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.

- Je tiens à exprimer ma profonde gratitude à mon directeur de mémoire, **Prof Édouard Ngor SARR**, pour sa disponibilité, ses conseils avisés, son accompagnement constant et sa rigueur scientifique. Son expertise, ses orientations et son soutien ont été d'un apport considérable dans l'aboutissement de ce travail.
- J'adresse aussi mes remerciements aux membres du jury, pour l'honneur qu'ils me font en acceptant d'évaluer ce travail, ainsi que pour leurs observations et recommandations constructives.
- Je tiens à remercier l'ensemble des enseignants de la filière **Management Informatisé des Organisations (MIO)**, dont les enseignements, les conseils et les orientations m'ont permis d'acquérir les compétences nécessaires à la réalisation de ce mémoire.
- Mes remerciements s'adressent également au personnel administratif et technique de l'**Université Assane Seck de Ziguinchor**, pour leur accueil, leur disponibilité et leur appui logistique.
- Je remercie particulièrement mon tuteur **Bacary Bodian**, pour son soutien indéfectible, ses encouragements, ses conseils et sa présence constante.
- A mes parents **Mamadou Goudiaby** et **Bacary Goudiaby**.
- À mes proches, amis et mentors **Dia Sow, Fatou Fall, Ismaila coly et El hadji Sagna**
- Je remercie également mes camarades de promotion, notamment **Priscilla, Alima .S. Sarr, Will et Mbathie**, pour leur esprit d'entraide, leur collaboration et les moments de partage. Merci pour tout.
- Enfin, j'exprime ma profonde gratitude à ma famille, véritable pilier de mon existence, pour son amour, sa patience, sa confiance et son soutien indéfectible. Sa présence a été essentielle tout au long de ce parcours.
- À toutes celles et à tous ceux qui, de manière visible ou discrète, ont contribué à la réussite de ce travail, je vous adresse ma profonde reconnaissance.
- **Merci à toutes et à tous.**

Résumé

Dans ce mémoire, nous abordons la problématique liée à la gestion des heures complémentaires des enseignements à l'Université Assane Seck de Ziguinchor. En effet, ce processus reste encore semi-automatisé, ce qui entraîne des erreurs de calcul, des retards de traitement, un manque de traçabilité et une faible visibilité pour les enseignants. L'objectif principal de ce travail est de concevoir un système d'information web permettant de déclarer les heures effectuées, de suivre les demandes, d'assurer une validation hiérarchique et de faciliter le calcul des montants dus. La solution proposée intègre plusieurs espaces selon les profils des utilisateurs, notamment l'espace enseignant, l'espace responsable, l'espace vice-recteur et l'espace administrateur. Des modules de gestion des cours, des salles, des pointages, des utilisateurs, ainsi que des états et rapports y sont intégrés afin d'améliorer la fiabilité, la transparence et l'efficacité du processus de gestion des heures complémentaires.

Mots-clés : système d'information, heures complémentaires, application web, validation hiérarchique.

Abstract

In this dissertation, we address the issue of managing additional teaching hours at Assane Seck University of Ziguinchor. Indeed, this process is still semi-automated, which leads to calculation errors, processing delays, lack of traceability, and limited visibility for teachers. The main objective of this work is to design a web-based information system that allows users to declare completed hours, track requests, ensure hierarchical validation, and facilitate the calculation of amounts due. The proposed solution includes several areas according to user profiles, namely the teacher area, manager area, vice-rector area, and administrator area. It also integrates modules for managing courses, classrooms, attendance records, users, as well as reports and statements in order to improve the reliability, transparency, and efficiency of the management process of additional teaching hours.

Keywords: information system, additional teaching hours, web application, hierarchical validation.



Sommaire

Dédicace	i
Remerciement	ii
Résumé	iii
Abstract	iv
Sommaire	v
Liste des figures	vi
Liste des tableaux	viii
Sigle.....	ix
Introduction générale	1
Chapitre 1 : Généralité sur les systèmes d'informations et les heures complémentaires.....	5
Chapitre 2 : Revue de la littérature et Cahier des charges et.....	19
Chapitre 3 : Analyse, conception et réalisation du SI	34
Chapitre 4 : présentation de la solution.....	67
Conclusion et perspective	81
Références	82
Webographie	84
Table des matières	85
Annexe 1	90
Annexe 2	94

Liste des figures

Figure 1:architecture système d'information	8
Figure 2: Diagramme de séquences de la procédure de transmission des heures complémentaires (explication : voir annexe 2)	17
Figure 3:plateforme de kairos UCAO	21
Figure 4: page de ERP5	21
Figure 5:université Avignon.....	22
Figure 6:université de Toulouse	23
Figure 7:université champillon.....	24
Figure 8: logiciel Apogée	24
Figure 9:logiciel Ypareo.....	25
Figure 10:plateforme de jibble	26
Figure 11: logiciel TimeCamp	26
Figure 12: logiciel Clockify	27
Figure 13:logiciel JOUROFF	28
Figure 14: logiciel MyScol.....	28
Figure 40: Diagramme de cas d'utilisation	49
Figure 41: Diagramme de classe	51
Figure 42: séquence pour créer un compte.....	53
Figure 43: séquence pour s'authentifier.....	54
Figure 44: séquence pour pointer un cours	54
Figure 45: séquence pour ajouter un cours.....	55
Figure 46: séquence pour vérifier le cours	56
Figure 47: le modèle relationnel.....	57
Figure 48: modèle physique de donnée	58
Figure 49: création de la base de données	59
Figure 50:importation de la base de données	59
Figure 51: structure de mase de données.....	60
Figure 52:interface PHPRAD.....	60
Figure 53; création du projet sur PHPRad.....	61
Figure 54: enregistrement du projet	61
Figure 55: le fichier du projet.....	62
Figure 56: choix du thème.....	62

Figure 57: gestion des menus	63
Figure 58:gestion des pages	64
Figure 59: gestion Page Design.....	64
Figure 60: Onglet Authentification	65
Figure 61: gestion des rôles.....	65
Figure 62: gestion de l'audites	66
Figure 63: page d'accueil plateforme	68
Figure 64: creer un compte.....	69
Figure 65: tableau de bord enseignant.....	69
Figure 66: pointer un cours	70
Figure 67: la liste des pointages	70
Figure 68: tableau de bord responsable	71
Figure 69: faire la répartition.....	71
Figure 70: gestion des cours.....	72
Figure 71: gestion des salles.....	72
Figure 72: gestion des pointages	73
Figure 73: vue complet des pointages	73
Figure 74: vue sur les enseignants.....	74
Figure 75:tableau de bord recteur.....	74
Figure 76: gestion des utilisateurs	75
Figure 77: gestion des enseignants	75
Figure 78:gestion des cours.....	76
Figure 79:gestion des enseignements	76
Figure 80: gestion des salles.....	77
Figure 81: vue globale des enseignements	77
Figure 82: gestion pédagogique	78
Figure 83: états et rapport.....	78
Figure 84: tableau de bord administrateur.....	79
Figure 85: App-logs.....	79

Liste des tableaux

Tableau 1: les charges statutaire.....	14
Tableau 2: la transformation.....	14
Tableau 3: Exemple de calcul général.....	15
Tableau 4: La tarification des heures	15
Tableau 5:Exemple : Si c'est un Professeur Titulaire	16
Tableau 6: Tableau de comparaison des logiciels	29
Tableau 7: Tableau de comparaison entre UML et MERISE	37
Tableau 8:Tableau de comparaison des outils de modélisation	39
Tableau 9:Tableau de comparaison Oracle et MySQL	40

Sigle

- **API** : Application Programming Interface / Interface de programmation d'application
- **CFA** : Centre de Formation d'Apprentis
- **CM** : Cours Magistral
- **CRUD**: Create, Read, Update, Delete / Créer, Lire, Modifier, Supprimer
- **CSS** : Cascading Style Sheets / Feuilles de style en cascade
- **DOM** : Document Object Model
- **DSS** : Decision Support Systems / Systèmes d'aide à la décision
- **ENT** : Environnement Numérique de Travail
- **ERP** : Enterprise Resource Planning / Progiciel de Gestion Intégré
- **ERP5** : Progiciel de gestion intégré open source
- **HCE** : Heures Complémentaires d'Enseignement
- **HCM** : Heures de Cours Magistraux
- **HTML** : HyperText Markup Language
- **HTD** : Heures de Travaux Dirigés
- **HTP** : Heures de Travaux Pratiques
- **HTTPS** : HyperText Transfer Protocol Secure
- **IDE** : Integrated Development Environment / Environnement de développement intégré
- **IHM** : Interface Homme-Machine
- **JS** : JavaScript
- **JVM** : Java Virtual Machine / Machine virtuelle Java
- **KPI** : Key Performance Indicator / Indicateur clé de performance
- **MCD** : Modèle Conceptuel de Données
- **MCT** : Modèle Conceptuel de Traitements
- **MERISE** : Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise
- **MIS** : Management Information Systems / Systèmes d'information de gestion
- **MLD** : Modèle Logique de Données
- **MLT** : Modèle Logique de Traitements
- **MPD** : Modèle Physique de Données
- **MPT** : Modèle Physique de Traitements
- **MTV** : Model Template View

- **MVC** : Model View Controller / Modèle Vue Contrôleur
- **NPM** : Node Package Manager
- **OLAP** : Online Analytical Processing / Analyse en ligne des données
- **ORM** : Object Relational Mapping / Mapping objet-relationnel
- **OSE** : Organisation des Services d'Enseignement
- **PECHE** : Programme d'Élaboration des Charges d'Enseignement
- **PHP** : Hypertext Preprocessor
- **PHPRad** : PHP Rapid Application Development
- **REST** : Representational State Transfer
- **SAGHE** : Système Automatisé de Gestion des Heures d'Enseignement
- **SGBD** : Système de Gestion de Base de Données
- **SGBDR** : Système de Gestion de Base de Données Relationnelle
- **SI** : Système d'Information
- **SID** : Système d'Information Décisionnel
- **SIG** : Système d'Information de Gestion
- **SIO** : Système d'Information Opérationnel
- **SQL** : Structured Query Language / Langage de requête structuré
- **SVG** : Scalable Vector Graphics
- **TD** : Travaux Dirigés
- **TP** : Travaux Pratiques
- **UASZ** : Université Assane Seck de Ziguinchor
- **UE** : Unité d'Enseignement
- **UFR** : Unité de Formation et de Recherche
- **UML** : Unified Modeling Language / Langage de modélisation unifié

Introduction générale

La gestion des heures complémentaires des enseignants-chercheurs représente un enjeu administratif et financier crucial pour les établissements universitaires. À l'Université Assane Seck de Ziguinchor (UASZ), comme dans la plupart des universités sénégalaises, ce processus repose encore largement sur des procédures manuelles utilisant des supports papier ou des fichiers Excel décentralisés. Cette approche traditionnelle génère des défis significatifs : calculs fastidieux, risques d'erreurs, difficultés de traçabilité, et retards dans le traitement et le paiement des enseignants. La transformation numérique des processus administratifs universitaires offre des perspectives prometteuses pour optimiser cette gestion. Les systèmes d'information modernes, notamment les applications web et les bases de données relationnelles, permettent d'automatiser et de rationaliser ces processus complexes. Des solutions de gestion académique existent sur le marché, mais leur adaptation spécifique à la gestion fine des heures complémentaires, tenant compte du contexte réglementaire sénégalais et des particularités organisationnelles de l'UASZ, nécessite une approche sur mesure.

Dans ce contexte, la conception d'un système d'information dédié à la gestion des heures complémentaires pourrait révolutionner ce processus administratif en améliorant son efficacité, sa transparence et sa fiabilité. Cette gestion des heures complémentaires implique plusieurs acteurs (enseignants, chefs de département, responsables financiers), suit un processus **workflow**¹ complexe (saisie, validation, calcul, paiement) et doit respecter un cadre réglementaire précis. Cette problématique soulève des questions techniques, organisationnelles et sécuritaires liées à la digitalisation d'un processus critique.

En effet, le problème majeur qui se pose est : **Comment concevoir et mettre en œuvre un système d'information capable de gérer de manière efficace, précise et sécurisée le processus des heures complémentaires des enseignants à l'Université Assane Seck de Ziguinchor ?** En d'autre terme Comment transformer le processus actuel de gestion des heures complémentaires, actuellement basé sur des procédures manuelles et des supports papier, en un workflow numérique intégré qui garantisse à la fois l'efficacité opérationnelle, la précision des calculs et la sécurité des données sensibles au sein de l'UASZ ?

Ainsi, la question centrale se décline en plusieurs sous-questions :

- Quelles sont les spécificités du processus de gestion des heures complémentaires à l'UASZ et quels sont les besoins réels des différents acteurs ?

¹ Désigne une suite organisée d'activités, de tâches ou d'étapes interconnectées qui impliquent plusieurs acteurs, systèmes ou règles métier.

- Quelle architecture technique et quels outils de développement choisir pour concevoir un système adapté, robuste et évolutif ?
- Comment modéliser le workflow de validation et le mécanisme de calcul automatique des heures et des montants dus ?
- Comment garantir la sécurité des données et l'intégrité des transactions, notamment celles liées aux aspects financiers ?
- Comment assurer l'ergonomie² du système pour favoriser son adoption par les utilisateurs ?

Pour répondre à cette problématique, nous proposons :

- Une analyse approfondie du processus existant de gestion des heures complémentaires à l'UASZ et une collecte des besoins auprès des parties prenantes ;
- La conception et le développement d'une application web sécurisée avec une base de données relationnelle ;
- L'implémentation d'un workflow de validation numérique et d'un module de calcul automatique basé sur les grilles tarifaires en vigueur ;
- Une méthodologie de test et de validation incluant des cas d'usage réels avec des utilisateurs finaux.

Ces propositions s'appuieront sur des méthodes de génie logiciel éprouvées (Merise, UML) et des technologies web modernes et adaptées au contexte.

Ainsi, nous fixons comme objectif général de cette étude : Concevoir un système d'information web intégré pour la gestion numérique des heures complémentaires des enseignants à l'UASZ.

Les objectifs spécifiques incluent :

- Analyser l'état actuel du processus et modéliser les besoins fonctionnels et non fonctionnels du système ;
- Spécifier l'architecture technique du système et choisir les technologies appropriées ;
- Concevoir le modèle de données et les interfaces utilisateur (IHM) ;
- Développer un prototype fonctionnel du système ;
- Tester et évaluer le prototype avec des données et des scénarios réalistes.

Cette étude présente des intérêts multiples :

² Désigne la capacité d'un système informatique à permettre à ses utilisateurs d'accomplir leurs tâches

- **Pour l'administration de l'université** : L'implémentation d'un système d'information dédié à la gestion des heures complémentaires apporte à l'administration universitaire une rationalisation profonde et une accélération significative des processus. En automatisant les flux de travail, depuis la soumission des demandes jusqu'à leur validation et leur transmission à la comptabilité, le système élimine les allers-retours papiers et les délais d'attente physiques entre les services. Cette digitalisation réduit considérablement les erreurs de saisie et de calcul, sources fréquentes de conflits et de rectifications chronophages. Sur le plan financier, l'administration acquiert une bien meilleure maîtrise budgétaire grâce à une visualisation en temps réel des engagements et des dépenses, permettant des prévisions plus fines et un pilotage éclairé des enveloppes allouées aux enseignements. Cette transparence facilite grandement les audits internes et externes, renforce la conformité aux règlements et sert de base factuelle en cas de litige, sécurisant ainsi toute la chaîne administrative.
- **Pour les enseignants** : le principal intérêt réside dans la transparence totale offerte sur le statut de leurs demandes d'heures complémentaires. Grâce à un portail personnel sécurisé, chaque enseignant peut suivre en temps réel l'état d'avancement de son dossier, savoir précisément à quel stade il se trouve (soumis, en cours d'instruction, validé, envoyé en paiement) et identifier le responsable de son traitement. Cette visibilité élimine l'incertitude et les démarches de relance fastidieuses auprès des secrétariats. Concrètement, cette automatisation et ce suivi intégré se traduisent par une réduction substantielle des délais de paiement, les traitements étant accélérés et les erreurs de liquidation minimisées. Enfin, le système simplifie radicalement les démarches pour l'enseignant : une interface unique et intuitive permet de formuler ses demandes, de joindre les justificatifs nécessaires et de consulter son historique, remplaçant avantageusement un parcours complexe mêlant formulaires papier, emails et déplacements physiques. Cette simplification administrative constitue un gain de temps et de sérénité non négligeable, contribuant ainsi à la qualité de vie au travail.

Notre étude adopte une méthodologie mixte. Une première phase quantitative, via un questionnaire, établit un diagnostic chiffré des dysfonctionnements du processus actuel. La phase qualitative qui suit, s'appuyant sur des entretiens, approfondit les causes de ces problèmes et précise les besoins des utilisateurs. Cette triangulation des données assure une compréhension globale et robuste, garantissant que la conception du système réponde à la fois aux exigences de performance administrative et aux attentes des enseignants.

Pour répondre à la problématique posée, ce travail s'articulera autour de quatre chapitres principaux :

- **Le premier chapitre** intitulé « **Généralités sur les systèmes d'information et les heures complémentaires** », présente les notions de base liées aux systèmes d'information ainsi que le contexte des heures complémentaires dans l'enseignement supérieur ;
- **Le deuxième chapitre**, « **Cahier des charges et revue de la littérature** », expose les besoins fonctionnels du système, les acteurs concernés ainsi que quelques solutions existantes ;
- **Le troisième chapitre** : « **Analyse, conception et réalisation du SI** », est consacré à la modélisation du système, aux choix techniques et à la mise en œuvre de la plateforme ;
- **Le quatrième chapitre** intitulé « **Présentation de la solution** », présente les principales interfaces développées et les fonctionnalités offertes aux différents utilisateurs.



Chapitre 1 : Généralité sur les systèmes d'informations et les heures complémentaires

1.1. Introduction

Dans le contexte actuel de la gestion universitaire, la rigueur administrative et la transparence financière sont des impératifs essentiels. La gestion des heures complémentaires, élément courant mais complexe de la vie académique, en est une parfaite illustration. Traditionnellement traitée par des procédures manuelles et des flux papier, cette gestion est souvent source de lenteurs, d'erreurs et d'opacité, tant pour l'administration que pour les enseignants. C'est précisément pour répondre à ce type de défis organisationnels que les systèmes d'information se sont imposés comme des outils stratégiques de modernisation, de rationalisation et de pilotage. Ce premier chapitre pose donc les fondements théoriques nécessaires à notre étude. Il s'agira d'une part d'explorer le concept de système d'information, ses composantes et ses finalités dans un environnement organisationnel. D'autre part, nous définirons le cadre spécifique des heures complémentaires dans l'enseignement supérieur, en explicitant leurs enjeux administratifs, financiers et pédagogiques.

1.2. Les systèmes d'information

1.2.1. Définitions

La notion de système d'information (SI) est fondamentale pour analyser et concevoir les outils de gestion des organisations modernes. Une définition académique largement partagée le présente comme un ensemble organisé de ressources (données, matériels logiciels, personnel, procédures) qui permet de collecter, stocker, traiter et diffuser l'information afin de faciliter la prise de décision, le contrôle des activités, la coordination et l'analyse au sein d'une organisation [1]. Cette vision systémique souligne que le SI dépasse la simple informatique pour englober les dimensions humaines et organisationnelles.

Dans une perspective managériale et opérationnelle, le SI est mieux compris comme un dispositif sociotechnique. Ce concept clé, développé par des auteurs comme Alter (2008), insiste sur l'interdépendance et l'alignement nécessaire entre les composantes sociales (les utilisateurs, leurs compétences, la structure organisationnelle, la culture) et les composantes techniques (les logiciels, les bases de données, l'infrastructure). Le succès d'un SI réside dans cette articulation, où la technologie est conçue pour supporter et transformer le travail humain, et où les acteurs s'approprient l'outil pour atteindre des objectifs communs [2].

Dans le contexte des administrations publiques et universitaires, notamment en Afrique, le SI est un levier stratégique de réforme. Il vise à automatiser et rationaliser les processus métier, à fluidifier la circulation de l'information et à en améliorer la qualité et la fiabilité. L'objectif final est d'accroître l'efficacité, la transparence et la capacité de pilotage de l'institution, en

remplaçant les circuits fragmentés et papiers par des flux numériques intégrés [3]. Cette compréhension est centrale pour notre étude, où la conception d'un SI pour les heures complémentaires doit résoudre des problèmes à la fois techniques (calculs, workflows) et sociaux (adhésion des enseignants, changement des pratiques administratives).

1.2.2. Missions d'un SI

Les missions d'un système d'information dépassent sa simple fonction technique de traitement de données pour englober des rôles stratégiques au sein de l'organisation. Sa mission fondamentale est d'assurer le traitement efficace et fiable des opérations courantes, en automatisant les processus métier pour en augmenter la rapidité, la précision et la productivité [4]. Dans le contexte d'une gestion des heures complémentaires, cela se traduit par la digitalisation du circuit de la demande, de la validation, du calcul et de la transmission à la paie. Au-delà de cette automatisation opérationnelle, une mission essentielle du SI est de produire une information fiable, consolidée et accessible en temps utile pour éclairer la décision. Il remplit ainsi un rôle de pilotage, en transformant des données brutes en indicateurs de gestion et en tableaux de bord synthétiques pour les responsables administratifs et financiers [5]. Au niveau de l'UASZ, cela permettrait un suivi budgétaire en temps réel des engagements sur les heures complémentaires et une analyse des volumes par département.

Enfin, dans une perspective plus large, le SI a une mission transformatrice et stratégique. Il est un instrument de modernisation qui participe à la refonte des processus, à l'amélioration de la qualité de service et au renforcement de la gouvernance. En offrant une transparence accrue et en fluidifiant la communication entre les parties prenantes, il contribue à la performance globale et à l'équité de l'organisation [6]. Appliqué à notre étude, un SI dédié ne se contenterait pas de compter des heures ; il servirait de levier pour une gestion plus rationnelle, transparente et équitable de la rémunération des enseignants, soutenant ainsi la mission éducative de l'université.

1.2.3. Architecture de base d'un SI

L'architecture d'un système d'information désigne l'organisation structurelle de ses différents composants et les relations qui les unissent pour assurer ses missions. Selon une modélisation classique, cette architecture repose sur une logique emboîtée et interdépendante des ressources. **Robert Reix** décrit cette architecture fondamentale autour de trois couches essentielles. La première est la couche des données, qui constitue le socle informationnel du système ; il s'agit des fichiers ou bases de données structurées contenant l'ensemble des informations nécessaires au traitement. La seconde est la couche de traitement, matérialisée par les logiciels applicatifs

et les procédures qui transforment les données en informations actionnables, comme le calcul automatique des montants dus ou la gestion des validations workflow. Enfin, la couche de diffusion assure l'interface avec les utilisateurs (via des écrans, des états, des alertes) et la communication avec d'autres systèmes, garantissant ainsi l'accès à l'information et son partage.

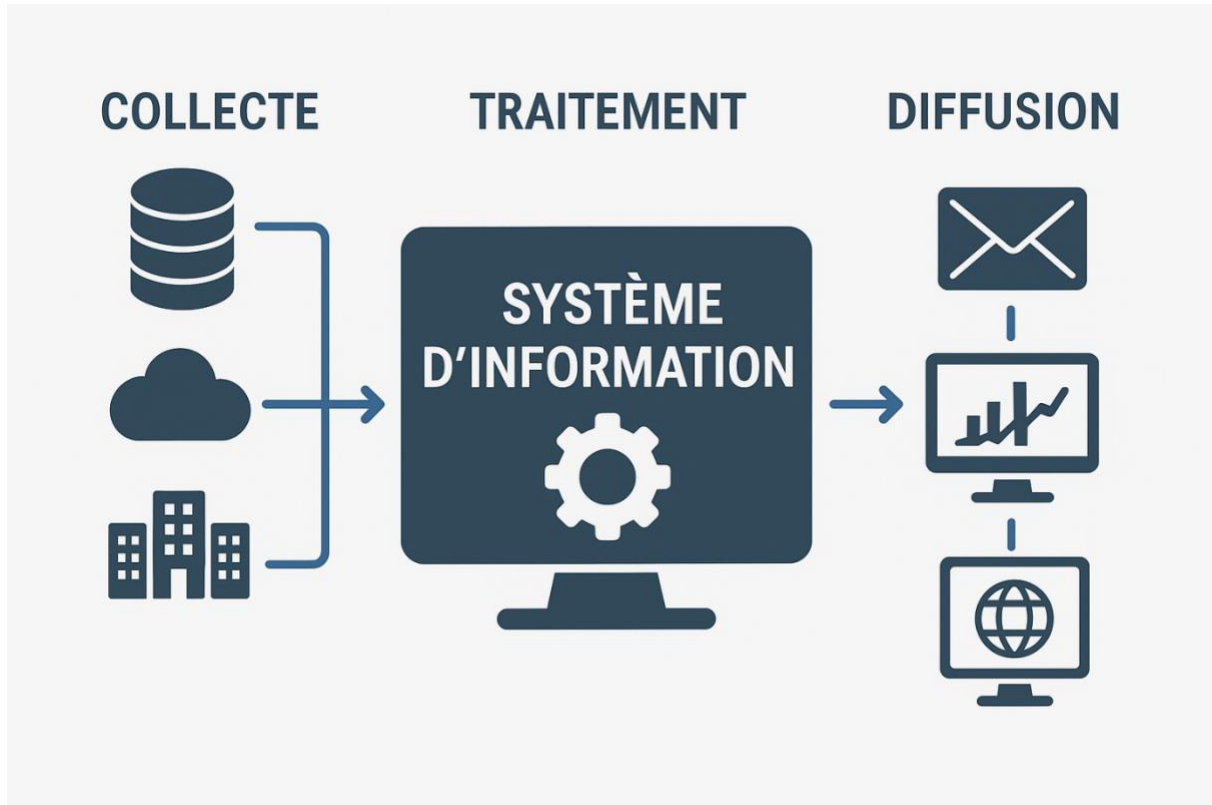


Figure 1:architecture système d'information

Cette architecture intégrée est la condition sine qua non pour que le système soit non seulement technologiquement opérationnel, mais aussi aligné sur les besoins réels, adopté par ses utilisateurs et capable de supporter la transformation des processus visée par l'institution [1]. Une conception architecturale réussie permet ainsi d'éviter l'écueil d'un simple outil informatique déconnecté des pratiques, pour construire un véritable dispositif de gestion au service de la performance administrative et de l'équité.

1.2.4. Les types de SI

3.1.1.1 Les SIO

Les Systèmes d'Information Opérationnels (SIO), également désignés sous le terme de systèmes de production ou de transaction, constituent la base transactionnelle et administrative des organisations. Leur mission centrale est de supporter et d'automatiser les activités routinières et répétitives du quotidien, en traitant un volume élevé de transactions élémentaires avec une

exigence absolue de fiabilité, de rapidité et de précision [4]. Ils sont au cœur du fonctionnement opérationnel, enregistrant et structurant les données de base générées par chaque action courante, telles qu'une vente, un paiement ou, dans le contexte universitaire, une inscription d'étudiant ou une déclaration d'heure d'enseignement.

Le rôle principal d'un SIO est donc de fluidifier et de sécuriser l'exécution des processus métier de base. Il permet de collecter, stocker et traiter les données opérationnelles de manière standardisée, assurant ainsi la cohérence et l'intégrité de l'information tout au long de la chaîne de valeur administrative. Ces systèmes sont généralement conçus pour une efficacité optimale dans la réalisation de tâches prédéfinies, avec des interfaces souvent standardisées, et ils alimentent en données brutes les autres couches du système d'information [1]. Leur performance se mesure à leur capacité à réduire les coûts de transaction, à minimiser les erreurs manuelles et à fournir une base de données fiable et à jour.

Dans le cadre spécifique d'une université comme l'UASZ, les SIO concernent des fonctions vitales telles que la gestion de la scolarité (inscriptions, notes), la paie, la gestion des ressources humaines ou la comptabilité. Appliqué à la problématique des heures complémentaires, un SIO serait précisément le système chargé d'enregistrer chaque déclaration soumise par un enseignant, de tracer chaque validation hiérarchique, d'effectuer les calculs de rémunération selon les barèmes en vigueur et de générer les ordres de paiement. Il automatisera ainsi le processus manuel existant, garantissant la rapidité du traitement, l'exactitude des calculs et la traçabilité de chaque étape, répondant directement aux besoins d'efficacité opérationnelle de l'administration et de fiabilité pour les enseignants [7].

3.1.1.2 Les SID

Les Systèmes d'Information de Décision (SID), souvent désignés sous l'appellation anglaise de Decision Support Systems (DSS), constituent une catégorie de systèmes dont la finalité principale est d'éclairer et d'améliorer le processus décisionnel au sein des organisations. À la différence des SIO qui automatisent et contrôlent les opérations courantes, les SID sont conçus pour analyser, synthétiser et modéliser des informations complexes en vue de résoudre des problèmes semi-structurés ou non structurés, et de soutenir des décisions stratégiques ou tactiques [4]. Leur rôle n'est pas d'exécuter des transactions, mais de fournir aux décideurs une capacité d'investigation, de simulation et d'anticipation à partir des données historiques et courantes collectées par les systèmes opérationnels.

Leur architecture repose typiquement sur une triade fonctionnelle. Premièrement, un entrepôt de données (data warehouse) qui consolide, nettoie et organise les données provenant de diverses sources opérationnelles (SIO) dans un format adapté à l'analyse. Deuxièmement, des outils d'analyse en ligne (OLAP) et de forage de données (data mining) qui permettent d'explorer ces informations sous différents angles (par département, par période, par type d'enseignement) et d'en extraire des tendances, des corrélations ou des prévisions. Troisièmement, des interfaces de visualisation (tableaux de bord interactifs, rapports synthétiques) qui présentent les résultats de manière claire et exploitable pour le décideur [8]. Dans le contexte de la gestion universitaire, et plus spécifiquement pour le pilotage des heures complémentaires à l'UASZ, un SID jouerait un rôle stratégique. Il permettrait par exemple d'analyser l'évolution des volumes et des coûts des heures complémentaires sur plusieurs années, d'en identifier les déterminants principaux (par filière, par période), de simuler l'impact budgétaire de différentes politiques de recours à ces heures, ou de détecter des écarts inhabituels. Ainsi, il passerait d'une simple comptabilisation des heures à une véritable aide à la décision pour une allocation optimale des ressources, un contrôle budgétaire prospectif et une planification stratégique des moyens pédagogiques.

3.1.1.3 Les SIG

Les Systèmes d'Information de Gestion, ou Management Information Systems (MIS) en anglais, représentent une catégorie essentielle de systèmes conçus pour répondre aux besoins d'information et de contrôle des cadres et gestionnaires opérationnels. Positionnés stratégiquement entre les systèmes opérationnels (SIO) et les systèmes d'aide à la décision (SID), les SIG ont pour mission principale de produire des rapports structurés, périodiques et synthétiques à partir des données transactionnelles brutes collectées par les SIO [9]. Leur finalité n'est pas de traiter les transactions individuelles, mais d'en agréger les résultats pour fournir une vue d'ensemble sur le fonctionnement des différents services ou processus, permettant ainsi un suivi régulier de la performance et un contrôle de gestion efficace.

Le fonctionnement caractéristique d'un SIG repose sur sa capacité à consolider, résumer et standardiser l'information. Contrairement aux SID qui explorent des scénarios futurs, les SIG se concentrent sur la présentation claire de la situation passée et présente, souvent sous forme de tableaux de bord, d'états prédéfinis (rapports sur les écarts budgétaires, analyses de productivité par département) et d'indicateurs clés de performance (KPI). Ils répondent ainsi à des besoins d'information récurrents et semi-structurés des managers, en mettant l'accent sur la cohérence, la ponctualité et la comparabilité des données dans le temps [4].

Dans le cadre spécifique de la gestion universitaire à l'UASZ, un SIG dédié aux heures complémentaires constituerait l'outil de pilotage tactique indispensable. Il transformerait les milliers de transactions individuelles issues du SIO (chaque demande validée) en informations de gestion actionnables. Par exemple, il pourrait générer automatiquement des rapports mensuels sur le volume d'heures complémentaires consommées par faculté, en comparant l'engagement budgétaire initial avec les dépenses réelles. Il permettrait également de suivre les tendances saisonnières, d'identifier les départements ou les types d'enseignement qui en recourent le plus, et de mesurer l'efficacité des processus administratifs (comme le délai moyen de traitement). Ainsi, le SIG fournirait aux responsables de l'UASZ la vision consolidée nécessaire pour ajuster les allocations budgétaires, optimiser l'offre pédagogique et assurer un contrôle financier rigoureux, tout en garantissant la transparence et l'équité dans la gestion de cette ressource.

1.3. La gestion des heures complémentaires

La gestion des heures complémentaires désigne l'ensemble des processus administratifs, financiers et pédagogiques par lesquels une institution planifie, contrôle, valide et rémunère le travail d'enseignement excédant le service statutaire de ses enseignants. Dans le contexte de l'enseignement supérieur, cette gestion constitue un levier essentiel de flexibilité pédagogique permettant d'ajuster l'offre de formation aux besoins réels, mais elle représente également un enjeu administratif complexe. Traditionnellement assurée par des procédures semi-automatisées, cette gestion est souvent source de lenteurs, d'erreurs de calcul et d'un manque de transparence, générant des mécontentements parmi les enseignants et des difficultés de pilotage budgétaire pour l'administration (Ndiaye, 2019). Une gestion optimisée requiert donc un circuit rigoureux intégrant la soumission des besoins, la validation hiérarchique, le calcul financier selon les barèmes en vigueur, l'émission des ordres de paiement et le suivi analytique. L'instauration d'un système d'information dédié apparaît ainsi comme une réponse stratégique pour transformer cette contrainte administrative en un processus fiable, rapide et transparent, contribuant à la fois à la performance organisationnelle et à la qualité de vie au travail des enseignants [4].

1.1.1. Heures normales

Les heures normales, ou service statutaire d'enseignement, constituent le volume obligatoire et fixe de travail pédagogique inclus dans le contrat et la rémunération de base de l'enseignant-chercheur. Défini par la réglementation nationale, notamment le Code général de la Fonction publique sénégalais et les décrets statutaires spécifiques au secteur de l'enseignement supérieur,

ce contingent représente le socle légal de l'engagement de l'enseignant envers son institution [W3]. Il s'exprime généralement sous la forme d'un forfait annuel équivalent temps plein, englobant non seulement les heures d'enseignement en présentiel mais aussi les activités pédagogiques connexes inhérentes à la fonction, telles que la préparation des cours, l'encadrement des étudiants et la participation aux réunions pédagogiques.

La fonction première de ce référentiel statutaire est de définir la limite entre le travail inclus dans le salaire de base et le travail donnant droit à rémunération supplémentaire. Il sert de seuil déclencheur pour le calcul des heures complémentaires et constitue ainsi la donnée de référence essentielle à toute gestion prévisionnelle de la masse salariale et de la charge pédagogique d'un établissement. Sa gestion précise est donc un impératif administratif et financier, car toute erreur ou ambiguïté sur ce volume de base se répercute directement sur la légalité et le coût des rémunérations complémentaires [7].

Dans le cadre de la conception d'un système d'information pour l'UASZ, la prise en compte des heures normales est fondamentale. Le système doit pouvoir intégrer et maintenir à jour le référentiel du service statutaire de chaque enseignant (qui peut varier selon le corps, le grade et les décharges éventuelles). Cette donnée de référence alimentera ensuite un moteur de calcul automatique qui comparera le volume d'heures déclarées par un enseignant sur une période à son plafond statutaire pour identifier, valider et liquider les éventuelles heures excédentaires. Une modélisation fiable de ce concept est la condition préalable à l'automatisation du processus de gestion des heures complémentaires et à la garantie d'une équité de traitement entre les enseignants.

1.1.2. Heures complémentaires

Les heures complémentaires désignent l'ensemble des activités d'enseignement effectuées au-delà du service statutaire annuel, et donnant lieu à une rémunération additionnelle calculée sur la base de barèmes officiels. Elles représentent un instrument de flexibilité pédagogique crucial pour les universités, permettant d'ajuster l'offre d'enseignement aux fluctuations des effectifs étudiants et des besoins spécifiques de formation. Leur gestion administrative forme un processus transversal critique, mobilisant plusieurs acteurs (l'enseignant demandeur, le responsable pédagogique, l'administration de la faculté et le service financier) et devant respecter un cadre réglementaire strict pour leur validation, leur liquidation et leur imputation budgétaire.

La fonction centrale de la gestion des heures complémentaires est d'assurer un circuit sécurisé, transparent et efficace entre la réalisation du travail et son paiement. Ce processus comprend

typiquement des étapes clés : l'initiation et la justification de la demande par l'enseignant, la validation pédagogique et budgétaire par l'autorité hiérarchique, le calcul administratif conforme aux barèmes, la transmission à la paie, et enfin le paiement. Lorsque ce processus est géré manuellement, via des formulaires papier et des feuilles de calcul dispersées, il est particulièrement vulnérable aux erreurs, aux pertes de dossiers, aux délais prolongés et à un manque de visibilité pour les parties prenantes, des problèmes fréquemment relevés dans les études sur la gestion universitaire en Afrique [10].

Pour le système d'information à concevoir pour l'UASZ, les heures complémentaires en sont l'objet central et le moteur fonctionnel principal. Le système devra numériser et fluidifier l'intégralité du circuit décrit. Concrètement, il devra offrir une interface de saisie et de suivi pour les enseignants, un workflow électronique de validation pour les responsables, un module de calcul automatique lié aux barèmes et au référentiel des heures normales, et une interface de contrôle et de reporting pour l'administration financière. En automatisant ce processus, le système visera à éliminer les goulots d'étranglement, à réduire drastiquement les erreurs de calcul, à garantir la traçabilité de chaque demande et à accélérer significativement les délais de paiement, répondant ainsi aux attentes de rationalisation administrative et d'équité pour les enseignants.

1.1.3. Charge statutaires

La charge statutaire désigne l'ensemble des obligations professionnelles fondamentales et réglementaires que l'enseignant-chercheur doit accomplir dans le cadre de son contrat de travail avec l'institution universitaire. Elle ne se limite pas aux seules heures d'enseignement en présentiel (heures normales) mais englobe le triptyque classique des missions de l'enseignement supérieur : l'enseignement, la recherche et les services à la collectivité (administration, encadrement, valorisation).

Le système de gestion doit impérativement intégrer ces distinctions statutaires, car la nature du grade et des missions de l'enseignant détermine directement son plafond d'heures normales et le mode de calcul des éventuelles heures complémentaires. La référence normative pour l'UASZ peut se décliner ainsi pour les corps principaux d'enseignants-chercheurs :

- **Professeur Titulaire ou Professeur Assimilé** : Ce grade représente le sommet de la carrière académique. Le titulaire est un enseignant-chercheur senior, responsable de cours magistraux (CM) de haut niveau, de la direction de recherches et de charges importantes d'animation scientifique. Sa charge statutaire d'enseignement est traditionnellement fixée à **72 Heures de Cours Magistraux (HCM)** par semestre. Cette

équivalence implique que toutes ses heures d'enseignement, quel que soit leur format pédagogique (CM, TD, TP), sont comptabilisées et converties en équivalent HCM pour le calcul du respect de son service statutaire.

- **Maître de Conférences (Titulaire ou Assimilé) et Assistant dispensant des Cours Magistraux (CM) :** Ce corps constitue le vivier des enseignants-chercheurs en milieu de carrière, assurant à la fois des enseignements fondamentaux (CM, TD) et une activité de recherche soutenue. Leur charge statutaire est définie de manière plus flexible pour refléter la diversité de leurs interventions pédagogiques. Elle est fixée à **72 HCM**, ou à l'équivalent en heures d'autres types d'enseignement
- **Assistant (ne dispensant pas de CM) :** Les assistants sont généralement des enseignants en début de carrière ou des doctorants chargés d'enseignement. Leur mission statutaire se concentre principalement sur l'encadrement pédagogique des étudiants via les travaux dirigés et pratiques. Leur charge d'enseignement est conséquente et fixée à **144 HTD ou 156 HTP** par semestre. Étant donné qu'ils ne donnent pas de cours magistraux, leur service est directement exprimé dans ces unités, sans conversion en HCM dans leur cas. Le système devra donc gérer un référentiel distinct pour ce corps, en vérifiant que le volume total de leurs HTD ou HTP déclarées n'excède pas ce plafond avant de qualifier les heures supplémentaires comme complémentaires.

Tableau 1: les charges statutaire

Statut	CM	TD	TP
Professeur Titulaire	72	-	-
Professeur Assimilé	72	-	-
Maitre de conférences Titulaire	72	114	156
Maitre de conférences Assimilé	72	114	156
Assistant	-	144	156

1.1.4. Règles de conversion

La première phase repose sur des règles de conversion standardisées, établies par la réforme des titres et des fonctions. Ces règles ont pour objectif exclusif de calculer le prélèvement sur le contingent statutaire. Elles s'appliquent ainsi :

Tableau 2: la transformation

01H de CM	2.16H de TP
01H de CM	1.58H de TD
01H de TD	1.37H de TP

Tableau 3: Exemple de calcul général

Statut	Heures Effectuées		
Statut	CM	TD	TP
Enseignant	34	20	30

- Si l'enseignant choisi de mettre tout en CM alors :
 - o Total CM = $34+(20/1.58)+(30/2.16) = \mathbf{60.65\ H}$
- Si l'enseignant choisi de mettre tout en TD alors :
 - o Total TD = $20+(34*1.58)+(30/1.37) = \mathbf{95.62\ H}$
- Si l'enseignant décide de mettre tout en TP alors :
 - o Total TP = $30+(20*1.37)+(34*2.16) = \mathbf{130.84\ H}$

1.1.5. Tarification des heures commentaire

La tarification des heures complémentaires constitue l'étape finale et financièrement critique du processus de gestion, consistant à calculer le montant dû à un enseignant pour chaque heure excédentaire validée. Cette opération doit respecter un cadre réglementaire précis, défini par l'institution et fondé sur les textes en vigueur, tel que le Code de l'Enseignement Supérieur, qui fixe les principes de rémunération des personnels enseignants-chercheurs. Le processus de tarification est structuré en deux phases logiques successives et automatisables : la conversion des heures déclarées en une unité de compte commune pour le retrait du service statutaire, puis l'application d'un tarif horaire spécifique au grade de l'enseignant sur les heures complémentaires identifiées.

Tableau 4: La tarification des heures

Statut	Tarif(FCFA)		
	01H de CM	01H de TP	01H de TD
Professeur Titulaire	7500	6 000	6 000
Professeur Assimilé	7500	5 500	5 500
Maître de conférences Titulaire	7500	5 000	5 000
Maître de conférences Assimilé	7500	5 000	5 000
Assistant	7500	5 000	5 000

Tableau 5:Exemple : Si c'est un Professeur Titulaire

Conversion	CM	TD	TP	Montant /Heure	Montant
	54	98	56		
CM	$54+(98/1.58)+(56/2.16) = \mathbf{141.95}$			7500	=1 064 625
TD	$98+(54*1.58)+(56/1.37) = \mathbf{224.20}$			6000	=1 345 200
TP	$56+(98*1.37)+(54*2.16)= \mathbf{306.9}$			6000	=1 841 400

1.1.6. Procédure de déclaration des heures complémentaires

Le processus actuel de déclaration des heures complémentaires repose encore largement sur des formulaires papier remplis et signés par les enseignants, avec un modèle uniforme. Ces fiches sont vérifiées au secrétariat de département pour vérification du remplissage, des signatures et de la cohérence des heures dont les déclarations ne sont pas reçus parfois dans les délais fixés puis transmises à l'UFR par un courrier interne, où un contrôle budgétaire est censé être effectué. Mais l'absence d'outils centralisés, la circulation manuelle des documents, les retards et parfois même les pertes rendent le suivi difficile. L'enseignant ne sait jamais où en est sa demande, et une simple erreur ou une fiche égarée peut entraîner un non-paiement.

Les acteurs s'accordent sur la nécessité d'un formulaire numérique unique et standardisé, permettant d'éviter les retards et les risques de perte mais aussi un envoi automatique. Ce formulaire intégré dans le système nous permettra de faire un suivi transparent, des validations en ligne et un calcul automatique des montants. Un tel système apporterait à la fois rapidité, fiabilité, visibilité pour les enseignants et une meilleure maîtrise budgétaire pour l'administration (**voir annexe 1**).

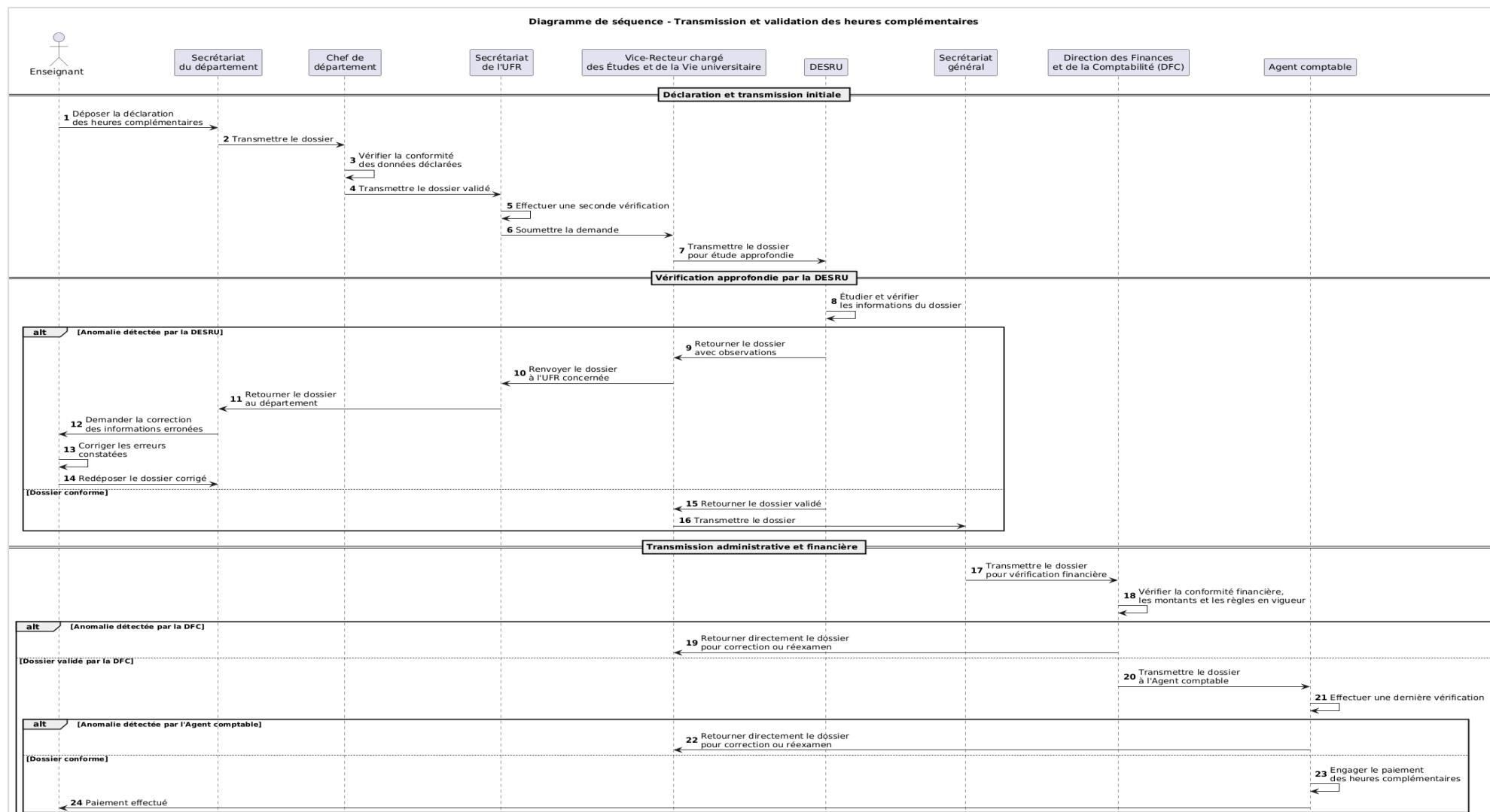


Figure 2: Diagramme de séquences de la procédure de transmission des heures complémentaires (explication : voir annexe 2)

1.4. Conclusion

Ce premier chapitre nous a permis d'établir le cadre théorique et conceptuel fondamental de notre étude. En définissant de manière approfondie les systèmes d'information, leurs missions, leur architecture et leurs principaux types (SIO, SID, SIG), nous avons identifié les caractéristiques et les potentialités d'un tel outil pour moderniser la gestion administrative. Parallèlement, l'analyse détaillée des concepts d'heures normales, complémentaires, de charge statutaire, de tarification et de procédure de déclaration a permis de cerner la complexité et les enjeux spécifiques du processus à optimiser. Cette double approche théorique démontre que la conception d'un système d'information dédié n'est pas un simple projet technique, mais bien une refonte sociotechnique d'un processus métier critique. Les définitions, règles et contraintes dégagées constituent ainsi le référentiel normatif et fonctionnel indispensable. Ce socle théorique nous permet désormais d'aborder, dans le chapitre suivant, le cahier des charges et la revue de la littérature.



Chapitre 2 : Revue de la littérature et Cahier des charges et

2.1 Introduction

Dans le premier chapitre, on a posé les bases en expliquant ce qu'est un système d'information et en détaillant tous les soucis que pose la gestion manuelle des heures complémentaires à l'UASZ. Maintenant, il s'agit de passer à l'étape suivante : concrètement, que faudrait-il faire pour améliorer les choses ? Ce deuxième chapitre sert exactement à ça. On va d'abord établir un cahier des charges, c'est-à-dire une liste détaillée de tout ce que le système idéal devrait faire, écrit dans un langage clair pour que les enseignants, les administratifs et les informaticiens se comprennent. C'est le document de référence qui dit : "Voilà nos problèmes, et voilà les fonctionnalités qui les résoudraient". Ensuite, pour ne pas partir de zéro, on va faire un tour d'horizon des solutions qui existent déjà ailleurs, dans d'autres universités. Cette revue de la littérature nous permettra de voir ce qui marche, ce qui ne marche pas, et de nous inspirer des bonnes idées. En somme, ce chapitre c'est le moment de la conception sur papier, avant de se lancer dans la réalisation technique

2.2. Revue de la littérature

La gestion des heures complémentaires des enseignants fait l'objet de plusieurs solutions informatiques mises en œuvre dans différentes universités et organisations. Cette revue de la littérature présente quelques outils existants afin d'analyser leurs apports et leurs limites, dans le but d'en dégager des éléments utiles pour la conception d'un système d'information adapté au contexte de l'université Assane Seck de Ziguinchor.

2.3.1. KairosSuite

KairosSuite³ est une plateforme de gestion intégrée conçue pour le secteur de l'éducation et de la formation. Ses fonctionnalités dédiées aux heures complémentaires s'inscrivent dans une logique d'automatisation et de traçabilité des activités pédagogiques. Le système permet d'abord la planification des enseignements en affectant les cours aux enseignants tout en générant les emplois du temps, ce qui facilite une répartition maîtrisée des charges horaires. Il assure ensuite un suivi précis des volumes horaires en distinguant les heures prévues des heures réellement effectuées, permettant ainsi d'identifier automatiquement les dépassements constituant des heures complémentaires. Par ailleurs, KairosSuite intègre des mécanismes de calcul automatisé basés sur des règles paramétrables, garantissant une évaluation fiable des heures supplémentaires. À cela s'ajoute un processus de validation administrative impliquant

³ https://ucao.kairossuite.com/kairos_ucao/login/auth

les responsables pédagogiques, ainsi que la production de rapports détaillés pour le suivi et le contrôle.

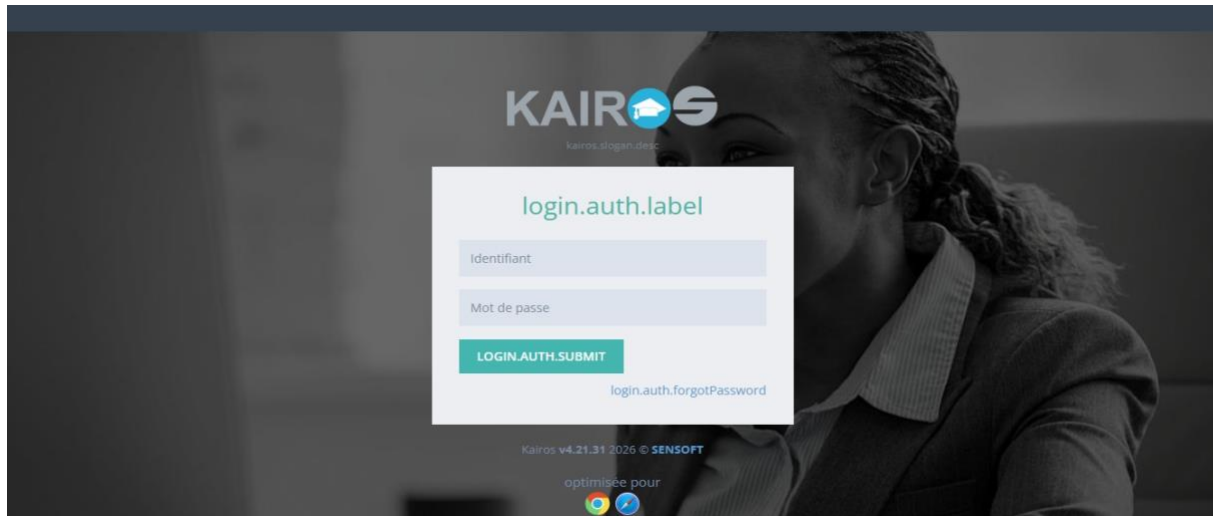


Figure 3: plateforme de kairos UCAO

2.3.2. ERP5

ERP5⁴ est un progiciel de gestion intégré open source utilisé dans plusieurs contextes, y compris le domaine universitaire. Dans le cadre académique, il s'inscrit dans une logique de gestion globale des activités pédagogiques, administratives et financières. Ses fonctionnalités permettent la gestion des cours, des enseignants, des ressources humaines et des processus organisationnels, avec une forte capacité de personnalisation.

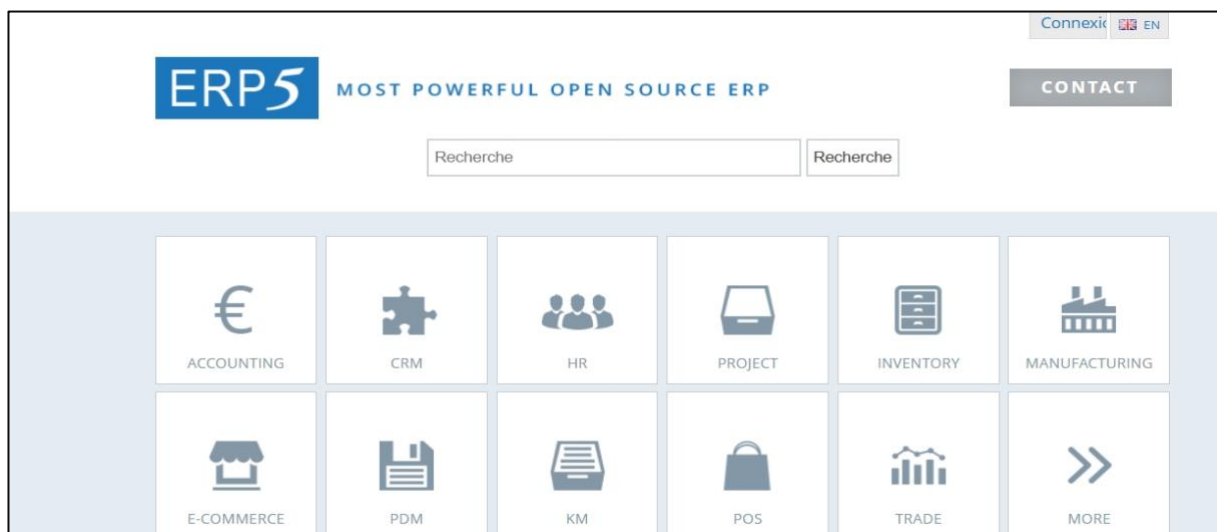


Figure 4: page de ERP5

⁴ <https://www.erp5.com/>

Concernant les activités d'enseignement, ERP5 permet de modéliser les charges pédagogiques et de suivre les activités réalisées, ce qui peut servir de base pour le suivi des volumes horaires. Son architecture flexible permet également d'intégrer des modules spécifiques pour le calcul et la gestion des heures complémentaires selon les besoins de l'établissement. Il est notamment utilisé dans le programme "One Student One ERP" déployé à Université Cheikh Anta Diop en collaboration avec Nexedi⁵.

2.3.3. OSE (Organisation des Services d'Enseignement).

OSE⁶ Est une application bien connue dans de nombreuses universités françaises. Elle aide à planifier et à suivre les heures d'enseignement des professeurs. Concrètement, les enseignants peuvent y déclarer en début d'année les cours qu'ils prévoient d'assurer, puis renseigner au fil des mois les heures réellement effectuées. L'outil s'adapte aux différents statuts – titulaires, contractuels ou vacataires – ce qui simplifie la gestion des aspects administratifs et financiers, notamment pour les heures complémentaires.

Généralement accessible via l'ENT de l'université, OSE inclut aussi un circuit de validation : les responsables pédagogiques et les services administratifs vérifient et valident les déclarations. Cela permet de mieux tracer les informations et limite les erreurs qui pourraient survenir avec des calculs manuels.



Figure 5: université Avignon

⁵ <https://www.nexedi.com/>

⁶ <https://e-doc.univ-avignon.fr/ressources-humaines/pole-gestion/heures-complementaires-2/>

2.3.4. SAGHE (Système Automatisé de Gestion des Heures d'enseignement)

Le logiciel SAGHE⁷ développé par l'Université Toulouse Jean-Jaurès offre une solution complète allant de la négociation des enveloppes de charge d'enseignement jusqu'au paiement des heures complémentaires et le suivi budgétaire SAGHE permet également un recrutement des vacataires 100 % dématérialisé et un accompagnement personnalisé des gestionnaires et directions d'UFR dans le suivi des heures d'enseignement et la préparation de la paie des heures complémentaires.



Figure 6: université de Toulouse

2.3.5. PECHE (Programme d'Élaboration des Charges d'Enseignement)

PECHE⁸ est un logiciel utilisé dans certaines universités pour organiser et suivre les charges d'enseignement des enseignants. Il permet de planifier les heures prévues, d'enregistrer les heures réellement effectuées et de gérer les heures complémentaires. En automatisant les validations et la transmission des informations vers l'administration, PECHE facilite le suivi pédagogique et budgétaire tout en réduisant les erreurs liées à la gestion manuelle.

⁷ <https://www.univ-tlse2.fr/accueil/universite/organisation/saghe>

⁸ <https://numerique.univ-jfc.fr/service/heures-denseignement>

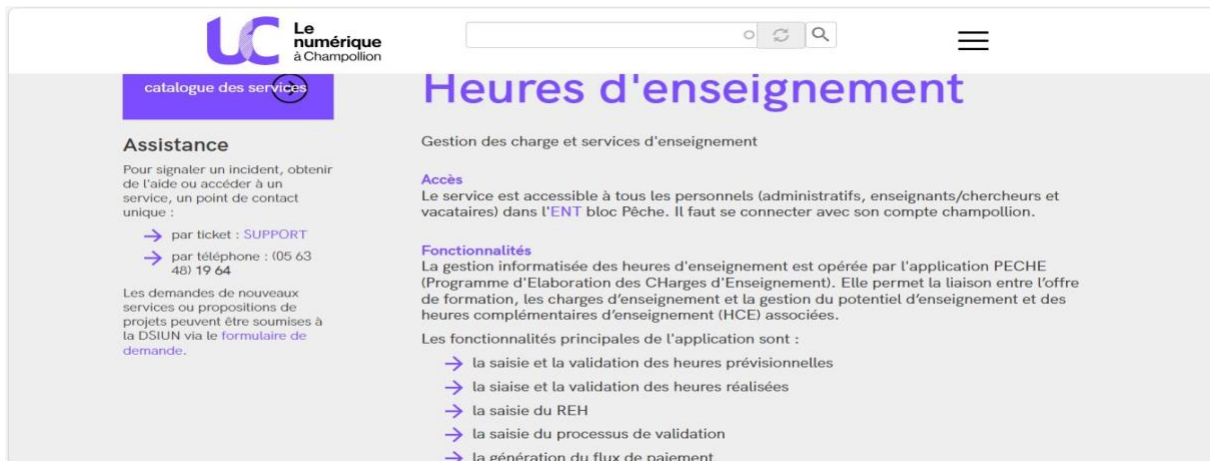


Figure 7: université champillon

2.3.6. Apogée

Apogée⁹ est un système d'information académique largement utilisé dans les universités françaises pour la gestion du parcours étudiant et des activités pédagogiques. Bien qu'il ne soit pas spécifiquement dédié aux heures complémentaires, ses fonctionnalités permettent d'encadrer indirectement leur gestion à travers une structuration rigoureuse de l'offre de formation et des enseignements. Le système assure la gestion des inscriptions administratives et pédagogiques, la planification des formations, ainsi que le suivi des parcours des étudiants. En lien avec les activités d'enseignement, il permet de structurer les maquettes pédagogiques et les volumes horaires associés, facilitant ainsi une meilleure visibilité des charges d'enseignement.



Figure 8: logiciel Apogée

⁹ <https://www.amue.fr/offre-de-solutions-et-services/solutions-et-services/logiciels-si/apogee>

Cette organisation contribue à identifier les écarts entre les volumes prévus et réalisés, qui peuvent être exploités dans d'autres systèmes pour la gestion des heures complémentaires. Apogée est notamment utilisé par Université Paris 1 Panthéon-Sorbonne et de nombreuses universités françaises.

2.3.7. Ypareo

Ypareo¹⁰ est une solution de gestion dédiée aux établissements de formation et d'enseignement supérieur, particulièrement en Europe. Elle propose une gestion intégrée des activités pédagogiques et administratives, incluant la planification des cours, la gestion des ressources et le suivi des apprenants. Dans le cadre des enseignements, Ypareo permet d'organiser les emplois du temps, de suivre les heures dispensées et de gérer les affectations des formateurs. Ces fonctionnalités offrent une visibilité sur les volumes horaires réalisés et facilitent l'identification des dépassements pouvant correspondre à des heures complémentaires. Le système intègre également des outils de reporting permettant d'analyser les charges de travail et d'optimiser la planification pédagogique. Ypareo est utilisé dans plusieurs établissements, notamment au CFA¹¹ des Universités Centre-Val de Loire.



Figure 9: logiciel Ypareo

2.3.8. Jibble

Jibble¹² est à la base un outil conçu pour les entreprises qui souhaitent suivre le temps de travail de leurs équipes. Mais il peut aussi, dans certaines situations, servir à tracer les heures supplémentaires ou complémentaires. L'idée est simple : depuis une interface épurée,

¹⁰ <https://www.ypareo.com/>

¹¹ Centre de Formation d'Apprentis

¹² <https://www.jibble.io/help/getting-started-with-jibbles-mac-time-tracker-app>

accessible sur ordinateur ou mobile, on enregistre ses heures en les associant à une activité ou à une plage horaire. Ensuite, toutes ces données sont synthétisées dans des tableaux de bord et des rapports clairs, ce qui permet de visualiser facilement le volume d'heures effectuées. Le logiciel calcule même automatiquement les heures supplémentaires et offre la possibilité d'exporter les informations vers des logiciels de paie ou de comptabilité.

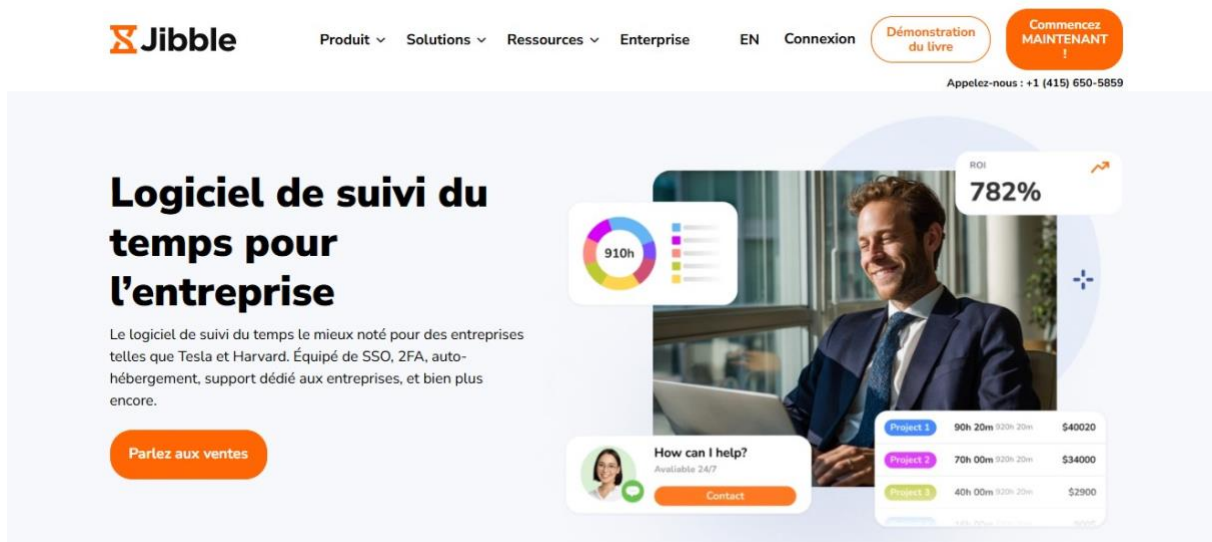


Figure 10: plateforme de jibble

2.3.9. TimeCamp

TimeCamp¹³ est un outil de gestion du temps qui met l'accent sur l'analyse détaillée de la charge de travail. Il est parfois utilisé dans des contextes académiques pour documenter les heures complémentaires, notamment dans une logique de suivi et d'optimisation des activités.



Figure 11: logiciel TimeCamp

¹³ <https://www.timecamp.com/>

Le logiciel permet de suivre les heures par enseignant et par projet, et génère des rapports détaillés pouvant servir à la planification des ressources humaines. TimeCamp offre également des possibilités d'intégration avec certains systèmes de paie, ce qui peut faciliter le processus de rémunération. Cependant, son orientation reste principalement managériale et analytique, ce qui limite son adéquation avec les réalités administratives des universités, où les règles de gestion des heures d'enseignement sont souvent plus spécifiques et encadrées.

2.3.10. Clockify

Clockify¹⁴ est une application de suivi du temps largement utilisée pour la gestion des feuilles de temps dans différents types d'organisations. Elle permet aux utilisateurs de saisir leurs heures de travail manuellement ou à l'aide d'un système de minuterie, puis de consulter des rapports récapitulatifs. Les données peuvent être exportées sous plusieurs formats, facilitant ainsi leur exploitation ultérieure. La simplicité de l'outil et son accessibilité, notamment grâce à sa version gratuite, constituent ses principaux atouts. Néanmoins, Clockify demeure un outil généraliste qui ne prend pas en compte les exigences spécifiques de la gestion universitaire, en particulier les procédures de validation hiérarchique et les règles propres aux heures complémentaires des enseignants.

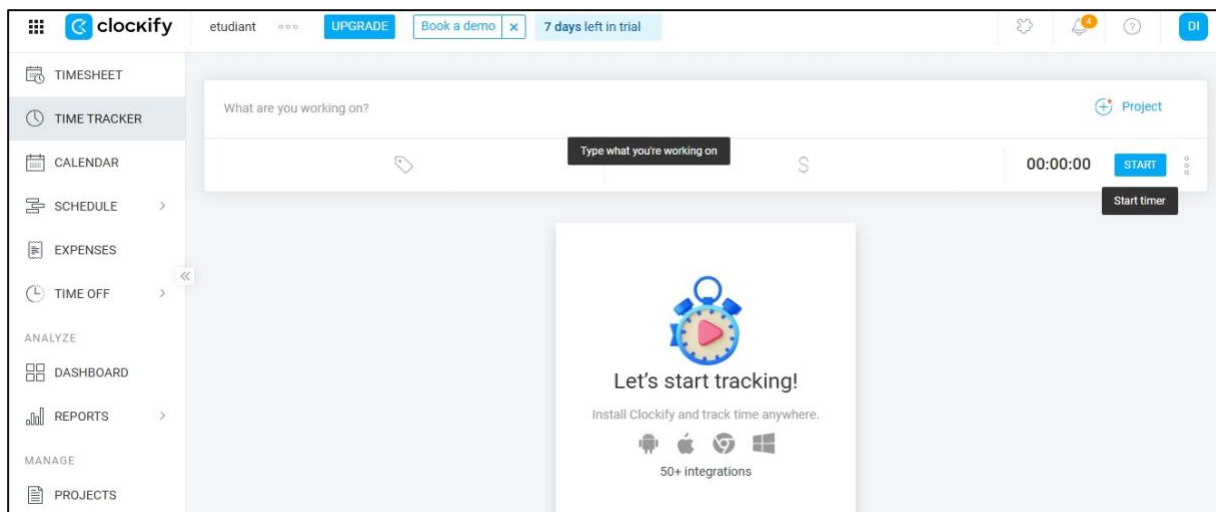


Figure 12: logiciel Clockify

2.3.11. JOUROFF

Jouroff¹⁵ est un logiciel de gestion du temps de travail qui se distingue par la prise en compte de règles complexes de calcul des heures supplémentaires et complémentaires. Il permet de paramétrer des règles liées aux conventions internes, telles que les majorations pour le travail

¹⁴ <https://clockify.me/fr/>

¹⁵ <https://jouroff.com/>

de nuit, les jours fériés ou les week-ends, ainsi que la gestion des heures de récupération ou de rémunération. Le système offre une certaine souplesse dans la saisie des heures, ce qui peut répondre aux besoins d'organisations ayant des contraintes horaires variées. Toutefois, JOUROFF n'est pas spécifiquement conçu pour le secteur de l'enseignement supérieur et nécessite donc une adaptation importante pour être utilisé dans un cadre universitaire, notamment en ce qui concerne la gestion pédagogique des heures d'enseignement.

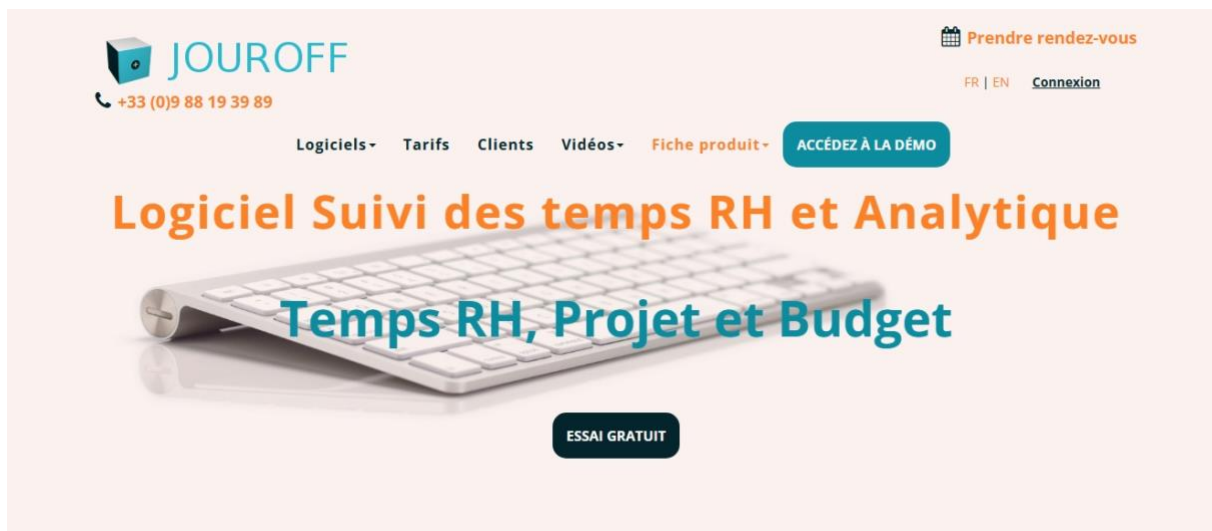


Figure 13: logiciel JOUROFF

2.3.12. MyScol

MyScol¹⁶ est un logiciel tout-en-un conçu pour les universités et les grandes écoles, qui rassemblent en un seul endroit les principaux processus pédagogiques et administratifs. Il facilite le suivi des enseignants, la gestion des emplois du temps, et permet de dématérialiser certaines démarches, comme les feuilles de présence par exemple.

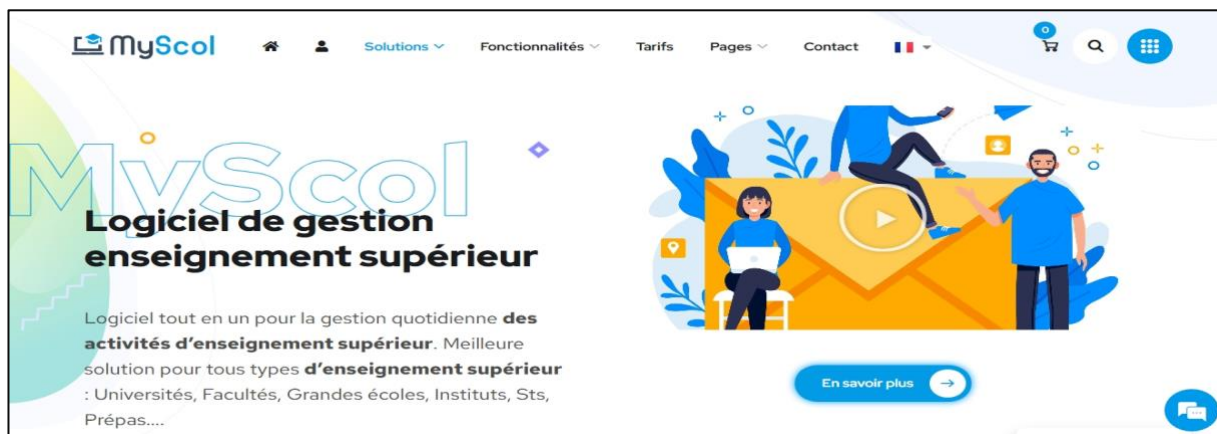


Figure 14: logiciel MyScol

¹⁶ <https://myscol.com/logiciel-de-gestion-enseignement-superieur/>

Selon les besoins, il peut même intégrer un suivi des heures travaillées par les enseignants, en lien avec les outils de gestion des ressources humaines. Le système propose aussi des espaces personnalisés pour les enseignants et les étudiants, ce qui rend l'accès aux informations bien plus simple et fluide. Cependant, il faut bien reconnaître que mettre en place un tel outil représente un investissement, à la fois financier et en termes de paramétrage, ce qui peut être un vrai défi pour les établissements aux budgets plus serrés.

Ce tableau 6 montre que les applications développées spécifiquement pour les universités fonctionnent majoritairement en intranet, intégrées aux ENT, afin de garantir la sécurité des données et le respect des procédures internes. À l'inverse, les solutions généralistes reposent sur un accès Internet, de type SaaS, offrant une grande facilité d'accès mais une intégration limitée aux processus académiques.

Tableau 6: Tableau de comparaison des logiciels

Logiciel	Gratuit	Open-source	Déclaration faite par le professeur	Validation hiérarchique	Suivi par le professeur	Accessible
KairosSuite	Non	Non	Oui	Oui	Oui	Intranet / Internet
OSE	Oui	Oui	Oui	Oui	Oui	Intranet
SAGHE	Non	Non	Oui	Oui	Oui	Intranet
PECHE	Non	Non	Oui	Oui	Oui	Intranet
Jibble	Oui	Non	Oui	Non	Oui	Internet
TimeCamp	Oui	Non	Oui	Non	Oui	Internet
Clockify	Oui	Non	Oui	Non	Oui	Internet
JOUROFF	Non	Non	Oui	Oui	Oui	Internet
MyScol	Non	Non	Oui	Oui	Oui	Internet
Apogée	Non	Non	Non	Oui	Non	Intranet
ERP5	Oui	Oui	Oui	Oui	Oui	Intranet / Internet
Ypareo	Non	Non	Oui	Oui	Oui	Internet

2.3.13. Positionnement

L'analyse comparative des différentes solutions met en évidence certaines limites importantes. D'une part, les outils spécifiques aux universités tels que **OSE**, **SAGHE** ou **PECHE** offrent des fonctionnalités complètes en matière de validation hiérarchique et de suivi des heures, mais ils sont souvent limités à des environnements intranet, peu flexibles et difficilement adaptables à d'autres contextes. D'autre part, les solutions généralistes comme **Jibble**, **TimeCamp** ou **Clockify** permettent une déclaration simple et accessible via Internet, mais elles ne prennent pas en charge les spécificités du milieu universitaire, notamment le circuit de validation des heures complémentaires. Par ailleurs, les ERP comme **Apogée**, **ERP5** et **Ypareo** proposent une gestion globale des activités académiques, mais ils ne sont pas toujours centrés sur la gestion fine et dédiée des heures complémentaires.

Dans ce contexte, nous avons choisi de proposer une solution qui se veut hybride et adaptée aux réalités de l'Université Assane Seck de Ziguinchor. L'objectif est de tirer parti des points forts des solutions existantes, notamment en permettant aux enseignants de déclarer directement leurs heures, en mettant en place un circuit de validation hiérarchique clair, ainsi qu'un système de suivi transparent des demandes, accessible via Internet. À travers cette approche, nous cherchons à concevoir un outil simple d'utilisation, mais efficace, capable d'améliorer la transparence, la fiabilité et l'efficacité dans la gestion des heures complémentaires.

2.3. Cahier des charges

2.3.5. Contexte

Actuellement, à l'UASZ, la gestion des heures complémentaires des enseignants repose sur un processus entièrement manuel. Cela signifie qu'un enseignant qui a effectué des heures complémentaires doit remplir une fiche papier, la faire signer, puis la déposer au secrétariat de son département. Ce document chemine ensuite, souvent par courrier interne, vers le secrétariat de l'UFR, puis vers les services centraux du rectorat pour calcul et transmission à la paie. Ce système papier est à l'origine de la majorité des problèmes rencontrés : lenteur extrême (parfois plusieurs mois de délai), risque élevé de perte ou d'erreur dans les calculs manuels, opacité totale pour l'enseignant qui ne peut pas suivre l'avancement de son dossier, et difficulté de pilotage budgétaire pour l'administration qui n'a pas de vision en temps réel des engagements. La mission consiste donc à remplacer ce circuit (courrier) par un outil numérique fiable.

2.3.6. Mission

La mission du projet est de concevoir et de mettre en œuvre un système d'information web, sécurisé et centralisé, dédié à la gestion du cycle complet des heures complémentaires à l'UASZ. L'objectif principal est de digitaliser et d'automatiser l'intégralité du processus, depuis la déclaration par l'enseignant jusqu'à la transmission des données de paiement, en passant par les validations hiérarchiques. Le système doit éliminer les supports papier, réduire radicalement les délais de traitement, garantir l'exactitude des calculs financiers et offrir une transparence complète à chaque acteur sur l'état d'avancement des dossiers.

2.2.3. Le périmètre du projet

Le projet se concentre uniquement sur la gestion des heures complémentaires des enseignements. Ces points suivants seront inclus dans le périmètre :

- La saisie et la soumission des déclarations par les enseignants.
L'enseignant aura l'accès sur la plateforme pour pouvoir saisir sa demande d'heure complémentaire
- Le workflow électronique de validation (responsable => Vice-recteur).
La plateforme sera conçue de tel sorte qu'il y'aura des validations par étape de manière hiérarchique.
- Le calcul automatique des montants
Une fois les heures complémentaires seront déclarer le système les calculera de manière automatique en donnant le montant selon l'heure déclarés (HTP, HTD, HCM) mais aussi selon le grade de l'enseignant ce qui évitera les erreurs.
- La génération de tableaux de bord
Elle offrira une vision en temps réel des indicateurs essentiels, comme le suivi des demandes pour les enseignants ou le pilotage budgétaire pour les administrateurs.

2.2.4. Etude de l'existant

À l'Université Assane Seck de Ziguinchor, la gestion des heures complémentaires des enseignants se fait encore de manière semi-automatiser. Pour déclarer ses heures, un enseignant doit remplir une fiche papier, la signer et l'apporter lui-même au secrétariat de son département. De là, elle prend le chemin du courrier interne jusqu'au secrétariat de l'UFR, où elle rejoint d'autres demandes avant d'être envoyée aux services centraux du rectorat pour être calculée et traitée.

Ce système, pose pas mal de problèmes au quotidien. Les délais de traitement sont souvent longs et difficiles à prévoir. L'enseignant, lui, n'a aucun moyen de savoir où en est son dossier. Les calculs des paiements, faits à la main à partir des grilles tarifaires, sont aussi source d'erreurs.

2.3.7. Besoins Fonctionnelle

Le système répondra aux besoins spécifiques de chaque type d'utilisateur :

- **Pour Enseignant**
 - **Consulter ses cours programmés** : permet à l'enseignant de visualiser les cours qui lui sont attribués selon le planning établi.
 - **Consulter l'historique de ses pointages** : permet de suivre l'ensemble des heures déjà déclarées et validées ou en attente.
 - **Ajouter ou déclarer un cours** : permet à l'enseignant de proposer ou enregistrer un cours sous condition d'autorisation du système.
 - **Suivre l'état de validation de ses heures** : permet de vérifier si les heures déclarées sont en cours de traitement, validées ou rejetées.
- **Pour l'administrateur**
 - **Gérer les utilisateurs** : permet de créer, modifier, supprimer les comptes et attribuer des rôles aux différents utilisateurs.
 - **Gérer les cours** : permet d'ajouter, modifier ou supprimer les cours disponibles dans le système.
 - **Gérer les enseignements** : permet d'organiser et d'associer les cours aux enseignants et aux formations.
 - **Gérer les salles** : permet de configurer et gérer les salles disponibles pour les cours.
 - **Gérer le pointage des enseignants** : permet de superviser et corriger les heures déclarées par les enseignants.
 - **Superviser et contrôler les données du système** : permet de garantir la cohérence et la fiabilité des informations enregistrées.
- **Pour le responsable**
 - **Faire la répartition des cours entre les enseignants** : permet d'attribuer les cours aux enseignants selon les besoins pédagogiques.

- **Gérer les salles (affectation et organisation)** : permet d'organiser l'utilisation des salles en fonction des emplois du temps.
- **Consulter les informations sur les cours et les enseignants** : permet d'avoir une vue globale sur les affectations et les disponibilités.
- **Suivre la planification des enseignements** : permet de contrôler l'organisation générale des cours et leur bon déroulement.
- **Pour le vice-recteur**
 - **Faire état** (générer ou consulter les rapports globaux du système)
 - Possibilité de valider ou de rejeter les demandes
 - Hérite tous les fonctionnalités de l'administrateur

2.2.6 Spécifications Technique / exigences technique

De manière Technique la plateforme devra être/avoir :

- Web accessible via HTTPS : le protocole https est une technologie de sécurité standard qui permet d'établir une liaison chiffrée entre un serveur Web et un client Web. HTTPS facilite la sécurisation des communications sur le réseau en identifiant et en authentifiant le serveur, ainsi que la confidentialité et l'intégrité des données transmises ;
- Logée dans le serveur de l'université ;
- Sécuriser mais aussi de contrôler qui a accès à la plateforme (authentification) ;
- Multi-utilisateurs : permet plusieurs utilisateur de se connecter.
- Une base de donnée relationnelles.

2.4. Conclusion

L'analyse des dysfonctionnements liés à la gestion manuelle des heures complémentaires à l'UASZ a permis d'identifier des insuffisances organisationnelles et fonctionnelles majeures, ayant conduit à l'élaboration d'un cahier des charges précisant les besoins essentiels du futur système, notamment la déclaration des heures par les enseignants, un circuit de validation hiérarchique structuré et un suivi transparent des demandes. Par ailleurs, la revue des solutions existantes a montré que plusieurs outils efficaces sont déjà utilisés dans d'autres contextes, mais que tous ne répondent pas aux réalités et contraintes des universités publiques africaines, ce qui justifie la conception d'un système d'information adapté au contexte spécifique de l'Université Assane Seck de Ziguinchor.



Chapitre 3 : Analyse, conception et réalisation du SI

3.1. Introduction

Ce chapitre constitue une étape essentielle du mémoire, car il porte sur l'analyse, la conception et la réalisation du système d'information proposé. Il vise d'abord à traduire les besoins identifiés en solutions concrètes à travers une modélisation adaptée, en s'appuyant sur des outils et méthodes reconnus. Ensuite, il présente les choix techniques retenus ainsi que les différentes étapes de mise en œuvre du système, depuis la conception jusqu'à la réalisation. L'objectif est ainsi de proposer une solution cohérente, fonctionnelle et adaptée au contexte de notre projet.

3.2. Choix des outils

3.2.1. Outils d'analyse

Dans le cadre de la conception de notre système d'information, le choix des outils s'avère essentiel afin d'assurer une modélisation claire, structurée et compréhensible du système. Ainsi, nous avons opté pour des outils d'analyse reconnus dans le domaine de l'ingénierie des systèmes d'information, notamment UML et MERISE.

3.2.1.1. UML

UML est un langage de modélisation standard largement utilisé dans le domaine du génie logiciel pour spécifier, visualiser et documenter les systèmes d'information. Il s'inscrit dans une approche orientée objet et propose plusieurs types de diagrammes, tels que les diagrammes de cas d'utilisation, de classes, de séquence ou d'activités, permettant de représenter respectivement les interactions, la structure et le comportement du système. Grâce à cette diversité de représentations, UML facilite la compréhension globale du système et améliore la communication entre les différents acteurs du projet. Il est aujourd'hui reconnu comme un standard international dans la conception des systèmes complexes [11]. UML dispose 14 diagrammes découpés en 3 groupes [W4]:

➤ Diagrammes structurels

- **Diagramme de classe** : représente les classes du système ainsi que leurs attributs, méthodes et relations ;
- **Diagramme d'objets** : montre une instance concrète des classes à un moment donné ;
- **Diagramme de composants** : décrit l'organisation des composants logiciels et leurs interactions ;
- **Diagramme de déploiement** : illustre la répartition physique des composants sur les équipements matériels.

- **Diagramme de packages** : organise les éléments du système en groupes logiques appelés packages ;
 - **Diagramme de structure composite** : détaille la structure interne d'une classe ou d'un composant ;
 - **Diagramme de profils** : permet d'adapter UML à des besoins spécifiques via des extensions.
- **Diagrammes comportementaux**
- **Diagramme de cas d'utilisation** : décrit les interactions entre les utilisateurs et le système ;
 - **Diagramme d'activités** : représente le déroulement des processus ou des flux de travail ;
 - **Diagramme d'états-transitions** : montre les différents états d'un objet et leurs évolutions.
- **Diagrammes d'interaction**
- **Diagramme de séquence** : illustre les échanges de messages entre objets dans le temps ;
 - **Diagramme de communication** : met en évidence les interactions entre objets sous forme de réseau ;
 - **Diagramme de timing** : représente l'évolution des états des objets en fonction du temps ;
 - **Diagramme global d'interaction** : fournit une vue d'ensemble des interactions en combinant plusieurs diagrammes.

Ces diagrammes permettent de modéliser de manière complète les aspects structurels, comportementaux et dynamiques du système d'information.

3.2.1.2. MERISE

MERISE est une méthode d'analyse et de conception des systèmes d'information orientée données, largement utilisée dans les environnements francophones. Elle repose sur une séparation claire entre les données et les traitements, ainsi que sur une modélisation en différents niveaux (conceptuel, logique et physique), ce qui permet d'assurer une meilleure structuration du système. Cette approche favorise la conception de bases de données cohérentes, fiables et adaptées aux besoins de l'organisation. MERISE est particulièrement adaptée aux systèmes de gestion reposant sur des bases de données relationnelles [12]. La méthode

MERISE repose sur trois niveaux de modélisation, permettant de structurer progressivement la conception du système d'information [W4]:

➤ **Niveau conceptuel**

- **Modèle Conceptuel de Données (MCD)** : représente les entités, leurs attributs et les relations entre elles indépendamment de toute contrainte technique ;
- **Modèle Conceptuel de Traitements (MCT)** : décrit les processus métiers et les traitements réalisés sans tenir compte des aspects techniques.

➤ **Niveau logique**

- **Modèle Logique de Données** : traduit le MCD en structures adaptées à un système de gestion de base de données relationnelle (tables, clés, relations) ;
- **Modèle Logique de Traitements** : précise l'organisation des traitements en tenant compte des choix logiques du système.

➤ **Niveau physique**

- **Modèle Physique de Données (MPD)** : définit la structure réelle de la base de données (tables, types de données, index) dans un SGBD.
- **Modèle Physique de Données (MPD)** : définit la structure réelle de la base de données (tables, types de données, index) dans un SGBD.

Tableau 7: Tableau de comparaison entre UML et MERISE

Critère	UML	MERISE
Nature	Langage de modélisation	Méthode d'analyse et de conception
Approche	Orientée objet	Orientée données et traitements
Représentation	Diagrammes (14 types)	Modèles (MCD, MLD, MCT...)
Objectif	Modéliser les systèmes logiciels	Concevoir les systèmes d'information
Niveau d'utilisation	Conception logicielle	Analyse organisationnelle et SI
Flexibilité	Très flexible et moderne	Plus rigide et structuré
Domaine d'usage	Développement logiciel	Systèmes d'information d'entreprise
Popularité actuelle	Très utilisée	Moins utilisée (mais encore enseignée)

Ainsi, dans le cadre de notre projet, le choix s'est porté sur UML. Ce choix se justifie par plusieurs raisons. D'une part, notre système implique une interaction directe entre différents acteurs (enseignants, responsables pédagogiques, administration), ce qui nécessite une modélisation claire des scénarios d'utilisation. D'autre part, la mise en œuvre envisagée repose sur des technologies orientées objet, notamment dans un environnement de développement web, rendant UML plus cohérent avec l'architecture du système.

Par conséquent, UML apparaît comme l'outil le plus approprié pour assurer une modélisation claire, flexible et évolutive de notre système d'information, tout en facilitant sa réalisation technique et sa maintenance.

3.2.2. Outils de modélisation

Dans le cadre de la conception du système d'information, le choix des outils de modélisation revêt une importance particulière. Ces outils permettent de représenter de manière graphique et structurée les différents aspects du système, facilitant ainsi la compréhension, la communication entre les acteurs du projet et la préparation à la phase de développement.

3.2.2.1. ARGOUML

ArgoUML est un outil de modélisation UML open-source écrit en Java et publié sous licence Eclipse Public License. Il supporte l'édition de diagrammes **UML 1.4**, notamment les diagrammes de cas d'utilisation, de classes, de séquence, d'activités et d'états-transitions, tout en offrant des fonctionnalités complémentaires telles que l'exportation des diagrammes dans différents formats et la génération de squelettes de code à partir de diagrammes UML [13]. ArgoUML a été développé dès 1999 et reste l'un des outils libres les plus utilisés pour l'apprentissage et la pratique de UML, même si son développement actif a ralenti après 2011, ce qui peut limiter son support des dernières versions de la norme UML [W6].

3.2.2.2. Start UML

StarUML est un outil de modélisation utilisé aussi pour créer des diagrammes selon la notation UML et supporte également d'autres normes de modélisation. Initialement publié comme projet open source, StarUML a évolué vers une solution plus moderne et multiplateforme, disponible sur Windows, MacOS et Linux [W7]. Il est conçu pour prendre en charge l'ensemble des diagrammes définis par la norme **UML 2.0**, y compris les diagrammes de classes, d'activités, de séquences, d'états et de composants, ce qui le rend adapté à la conception de systèmes complexes [W7]. StarUML propose également une interface ergonomique et des fonctionnalités avancées telles que l'exportation des modèles vers différents formats, facilitant

la documentation et la communication des modèles. De plus, il est souvent cité dans la littérature comme l'un des outils les plus utilisés dans l'enseignement et l'industrie pour la réalisation des modèles UML [14].

Tableau 8: Tableau de comparaison des outils de modélisation

Outil	Type	Langage supporté	Open-source	Multi-plateforme	Export diagrammes	Génération de code
ArgoUML	Logiciel	UML 1.4	Oui	Windows, MacOS, Linux	PNG, JPG, SVG	Oui (Java)
StarUML	Logiciel	UML 2.0	Non	Windows, MacOS, Linux	PNG, JPG, PDF, HTML	Oui (Java, C#, PHP)

Pour la modélisation de notre système d'information, nous avons choisi **StarUML** en raison de sa compatibilité avec UML 2.0 et de ses fonctionnalités avancées d'export et de génération de code. Cet outil moderne facilite la création de diagrammes clairs et complets, tout en permettant une documentation exploitable pour le développement. Ainsi, StarUML constitue une solution adaptée à notre projet, alliant efficacité, précision et praticité pour la conception.

3.2.3. Bases de données

3.2.3.1. Oracle

Oracle est un système de gestion de base de données relationnelle largement utilisé dans les environnements professionnels pour la gestion de données à grande échelle. Développé par Oracle Corporation, il se distingue par sa robustesse, sa sécurité avancée et ses performances élevées, ce qui en fait une solution adaptée aux systèmes critiques [W8]. Oracle intègre des fonctionnalités avancées telles que la gestion des transactions, la haute disponibilité, les mécanismes de sauvegarde et de récupération, ainsi que des outils d'optimisation des requêtes, permettant d'assurer une gestion fiable et efficace des données [15]. Par ailleurs, il est souvent utilisé dans les grandes organisations et les institutions nécessitant une forte capacité de traitement et une sécurité renforcée, bien que son coût de licence puisse constituer une limite dans certains contextes académiques [16].

3.2.3.2. MySQL

MySQL est un système de gestion de base de données relationnelle open-source très répandu, particulièrement dans le développement d'applications web et de systèmes d'information académiques. Il est apprécié pour sa simplicité d'utilisation, sa rapidité et sa compatibilité avec de nombreux langages de programmation, ce qui facilite son intégration dans divers environnements [W9].

MySQL repose sur le langage SQL standard et permet une gestion efficace des données grâce à une architecture client-serveur flexible et performante [15]. En outre, sa gratuité et sa large communauté d'utilisateurs en font une solution accessible et adaptée aux projets de taille moyenne, tout en offrant des performances satisfaisantes pour la gestion des données [16].

Tableau 9: Tableau de comparaison Oracle et MySQL

Critères	Oracle	MySQL
Type	SGBDR propriétaire	SGBDR open-source
Coût	Payant	Gratuit
Utilisation	Complexe	Simple et facile à prendre en main
Performance	Très élevée (grandes entreprises)	Élevée (applications web et SI moyens)
Maintenance	Nécessite expertise avancée	Facile à administrer
Sécurité	Très avancée	Bonne
Communauté	Restreinte (professionnelle)	Très large (open-source)
Déploiement	Complexe	Rapide et simple

Ce tableau met en évidence que, même si Oracle est une solution très puissante, **MySQL apparaît comme le choix le plus pertinent** dans le cadre d'un projet académique. En effet, sa gratuité, sa simplicité d'utilisation, sa facilité de déploiement et sa forte intégration avec les technologies web en font une solution plus adaptée aux contraintes techniques et financières du projet.

3.2.4. Langage de programmation

Les langages de programmation constituent un élément fondamental dans la réalisation d'un système d'information, car ils permettent de traduire les besoins fonctionnels et techniques en instructions exécutables par la machine. Ils interviennent à différents niveaux du système,

notamment pour le développement de l'interface utilisateur (**front-end**) nous avons **HTML**, **CSS**, **JavaScript** mais aussi du logique métier (**back-end**) le **C#** et de la gestion des données. Le choix des langages de programmation dépend de plusieurs critères tels que la performance, la compatibilité avec les technologies utilisées, la facilité de développement et la maintenabilité du système.

3.2.4.1. HTML

HTML (HyperText Markup Language) est un langage de balisage utilisé pour structurer le contenu des pages web. Il permet de définir la structure logique d'un document en organisant les éléments tels que les titres, les paragraphes, les images, les liens hypertextes et les formulaires [17]. HTML repose sur un système de balises qui décrivent la nature des contenus, facilitant ainsi leur interprétation par les navigateurs web. Il constitue la base de toute application web et est généralement associé à d'autres technologies pour enrichir les fonctionnalités et améliorer l'expérience utilisateur [W10]. Son évolution continue, notamment avec HTML5, a permis l'intégration de nouvelles fonctionnalités multimédias et interactives.

3.2.4.2. CSS

CSS (Cascading Style Sheets) est un langage de feuille de style utilisé pour décrire la présentation visuelle des documents HTML. Il permet de contrôler l'apparence des éléments en définissant les couleurs, les polices, les marges, les espacements et la mise en page globale [17]. L'un des principaux avantages de CSS réside dans la séparation qu'il établit entre la structure (HTML) et la présentation, ce qui facilite la maintenance et l'évolution des applications web. De plus, CSS permet de créer des interfaces responsives adaptées à différents types d'écrans (ordinateurs, tablettes, smartphones), améliorant ainsi l'accessibilité et l'ergonomie des systèmes web modernes [W11].

3.2.4.3. JavaScript

JavaScript est un langage de programmation interprété, principalement utilisé pour rendre les pages web interactives et dynamiques. Il permet de manipuler le contenu HTML et CSS en temps réel, de gérer les événements utilisateurs (clics, saisies, navigation) et d'implémenter des fonctionnalités avancées telles que les animations, les validations de formulaires et les mises à jour dynamiques sans rechargement de page [18]. JavaScript est aujourd'hui un langage incontournable du développement web, notamment grâce à son utilisation côté client et côté serveur via des environnements comme Node.js. Il dispose également d'un vaste écosystème

de bibliothèques et frameworks qui facilitent le développement d'applications complexes [W12].

3.2.4.4. Python

Python est un langage de programmation de haut niveau, interprété et polyvalent, largement utilisé dans le développement web, l'analyse de données, l'intelligence artificielle et l'automatisation. Sa syntaxe simple et lisible facilite son apprentissage et améliore la productivité des développeurs, ce qui explique son adoption croissante dans les milieux académiques et professionnels [19]. Python dispose également d'un écosystème riche de bibliothèques et de frameworks, tels que Django et Flask, qui permettent de développer rapidement des applications robustes et évolutives [20]. De plus, sa portabilité et sa compatibilité avec différents systèmes d'exploitation en font un outil privilégié pour le développement moderne.

3.2.4.5. PHP

PHP (Hypertext Preprocessor) est un langage de programmation côté serveur principalement utilisé pour le développement d'applications web dynamiques. Il est intégré directement dans le code HTML, ce qui permet de générer des pages web interactives et de gérer efficacement les bases de données [21]. PHP est largement utilisé dans le développement de systèmes d'information en raison de sa simplicité, de sa flexibilité et de sa compatibilité avec de nombreux serveurs web tels qu'Apache [22]. De plus, il est à la base de nombreux systèmes populaires comme WordPress, ce qui témoigne de sa robustesse et de son adoption à grande échelle.

3.2.4.6. Java

Java est un langage de programmation orienté objet, conçu pour être portable grâce au principe « Write Once, Run Anywhere », rendu possible par la machine virtuelle Java (JVM). Il est largement utilisé dans le développement d'applications d'entreprise, de systèmes distribués et d'applications mobiles (Android) en raison de sa robustesse, de sa sécurité et de ses performances [23]. Java propose une gestion automatique de la mémoire et un ensemble riche de bibliothèques standard, facilitant le développement de systèmes complexes et fiables [24]. Son utilisation dans les grandes infrastructures informatiques en fait un langage incontournable dans le domaine des systèmes d'information.

3.2.4.7. C++

C++ est un langage de programmation compilé, orienté objet et basé sur le langage C, permettant de développer des applications performantes et proches du matériel. Il est largement utilisé dans les systèmes embarqués, les logiciels systèmes, les jeux vidéo et les applications nécessitant une gestion fine des ressources [25]. C++ offre un contrôle précis de la mémoire et des performances, ce qui le rend particulièrement adapté aux applications critiques nécessitant une grande efficacité [26]. Malgré sa complexité, il reste un langage fondamental dans le domaine de l'ingénierie logicielle et des systèmes à haute performance.

3.2.4.8. C#

C# est un langage de programmation orienté objet développé par Microsoft, conçu pour le développement d'applications sur la plateforme .NET. Il est largement utilisé pour créer des applications web, desktop et mobiles, grâce à sa syntaxe moderne et à ses fonctionnalités avancées telles que la gestion des exceptions, le typage fort et la programmation orientée objet [27]. C# offre également des mécanismes robustes pour la gestion de la mémoire et la sécurité des applications, ce qui en fait un langage fiable pour le développement de systèmes complexes. Son intégration avec l'environnement .NET permet de bénéficier de nombreux outils et bibliothèques facilitant le développement et la maintenance des applications [W13].

3.2.5. Framework de développement

Les frameworks de développement sont des ensembles d'**outils**, de **bibliothèques** et de **conventions** qui facilitent la conception et la réalisation des applications informatiques. Ils fournissent une architecture préétablie permettant d'organiser le code, de standardiser les pratiques de développement et d'accélérer le processus de réalisation. En offrant des composants réutilisables et des fonctionnalités intégrées, les frameworks permettent de réduire les erreurs, d'améliorer la maintenabilité du code et de garantir une meilleure qualité des applications développées [28].

De plus, les frameworks modernes intègrent souvent des mécanismes avancés tels que la gestion des routes, la sécurité, la gestion des sessions et l'accès aux bases de données, ce qui simplifie considérablement le travail des développeurs. Ils favorisent également l'adoption de bonnes pratiques comme l'architecture MVC (Modèle-Vue-Contrôleur), contribuant ainsi à une meilleure séparation des responsabilités au sein de l'application [29]. Ainsi, l'utilisation d'un framework s'avère essentielle dans le développement de systèmes d'information complexes, en garantissant efficacité, évolutivité et robustesse.

3.2.5.1. ASP.NET MVS

ASP.NET MVC est un framework de développement web proposé par Microsoft, basé sur le modèle architectural MVC (Modèle-Vue-Contrôleur). Il permet de séparer clairement la logique métier, l'interface utilisateur et la gestion des données, ce qui facilite la maintenance et l'évolution des applications [29]. Ce framework offre également des fonctionnalités intégrées telles que la gestion des routes, la sécurité, l'authentification et l'interaction avec les bases de données via des technologies comme Entity Framework. Grâce à sa robustesse et à son intégration avec l'environnement .NET, ASP.NET MVC est largement utilisé pour le développement d'applications web professionnelles [W14].

3.2.5.2. Bootstrap

Bootstrap est un framework front-end open-source utilisé pour concevoir des interfaces web modernes et responsives. Développé initialement par Twitter, il propose un ensemble de composants prédéfinis tels que les boutons, les formulaires, les menus de navigation et les grilles de mise en page [30]. Bootstrap facilite la création d'interfaces adaptables à différents types d'écrans (ordinateurs, tablettes, smartphones) grâce à son système de grille flexible. Il permet ainsi de gagner du temps dans le développement tout en garantissant une cohérence visuelle et une bonne ergonomie des applications web [W15].

3.2.5.3. Entity Framework

Entity Framework est un framework de mapping objet-relationnel (ORM) développé par Microsoft. Il permet aux développeurs de manipuler les bases de données en utilisant des objets du langage de programmation, sans avoir à écrire directement des requêtes SQL complexes. Ce framework facilite la gestion des données en assurant la correspondance entre les classes du programme et les tables de la base de données [31]. Entity Framework contribue ainsi à améliorer la productivité, à réduire les erreurs et à simplifier la maintenance des applications. Il est largement utilisé dans les environnements .NET pour le développement d'applications orientées données [W16].

3.2.5.4. Symfony

Symfony est un framework PHP open source, conçu pour le développement d'applications web robustes et maintenables. Il repose sur une architecture modulaire basée sur des composants réutilisables, permettant aux développeurs de construire des applications évolutives tout en respectant les bonnes pratiques du génie logiciel [32]. Symfony intègre des fonctionnalités avancées telles que la gestion des routes, la sécurité, l'injection de dépendances et la gestion

des bases de données via Doctrine. Il est largement utilisé dans les projets d'entreprise en raison de sa flexibilité et de sa conformité aux standards du web moderne [W17].

3.2.5.5. PHPRad

PHPRad est un framework et outil de génération de code basé sur PHP, permettant de créer rapidement des applications web avec des fonctionnalités CRUD sans nécessiter une programmation avancée. Il propose une interface graphique facilitant la conception des applications, ainsi que des fonctionnalités intégrées telles que l'authentification, la gestion des utilisateurs et l'interaction avec les bases de données [W18]. Ce type d'outil est particulièrement utile pour le prototypage rapide et le développement de systèmes d'information simples, bien qu'il soit moins flexible que des frameworks traditionnels comme Symfony ou Laravel.

3.2.5.6. Laravel

Laravel est un framework PHP moderne, open source, conçu pour simplifier le développement d'applications web en adoptant une syntaxe élégante et expressive. Il repose sur le modèle MVC (Model-View-Controller) et propose des fonctionnalités intégrées telles que le routage, l'ORM Eloquent, la gestion des sessions, l'authentification et la sécurité [33]. Laravel est très populaire dans la communauté des développeurs web en raison de sa facilité d'utilisation, de sa documentation riche et de son écosystème complet, ce qui en fait un choix privilégié pour le développement rapide d'applications performantes.

3.2.5.7. Django

Django est un framework web open source écrit en Python, basé sur le modèle architectural MTV (Model-Template-View). Il est conçu pour faciliter le développement rapide d'applications web sécurisées et maintenables. Django intègre par défaut de nombreuses fonctionnalités telles que l'authentification, l'administration automatique, la gestion des formulaires et la sécurité contre les attaques courantes [34]. Il est largement utilisé dans le développement d'applications web complexes en raison de sa robustesse et de sa philosophie « batteries included », qui réduit le besoin d'outils externes.

3.2.5.8. Node.js

Node.js est un environnement d'exécution JavaScript côté serveur, basé sur le moteur V8 de Google Chrome. Il permet de développer des applications réseau rapides et scalables grâce à son architecture non bloquante et événementielle, particulièrement adaptée aux applications temps réel [35]. Node.js est largement utilisé pour la création d'API REST, de systèmes distribués et d'applications web modernes, en raison de sa performance et de sa capacité à gérer

un grand nombre de connexions simultanées. Son écosystème, riche grâce au gestionnaire de paquets npm¹⁷, facilite l'intégration de nombreuses bibliothèques et outils.

3.2.5.9. React.js

React (souvent appelé React.js) est une bibliothèque JavaScript développée par Meta Platforms, utilisée pour construire des interfaces utilisateur dynamiques et interactives. Elle repose sur le concept de composants réutilisables et utilise un mécanisme appelé Virtual DOM¹⁸ pour optimiser les performances d'affichage [36]. React est particulièrement adapté au développement d'applications web monopage, offrant une expérience utilisateur fluide et réactive. Sa popularité et son large écosystème en font l'un des outils les plus utilisés dans le développement frontend moderne.

3.2.6. Autres outils et technologies

3.2.6.1. Visual Studio

Visual Studio est un environnement de développement intégré (IDE) développé par Microsoft, qui permet de concevoir, coder, déboguer et déployer des applications sur différentes plateformes (Windows, web, mobile, cloud). Il offre des fonctionnalités avancées telles que l'auto-complétion, la gestion des projets multi-langages, l'intégration avec les frameworks .NET et la possibilité de générer automatiquement du code, ce qui facilite le travail des développeurs. Visual Studio est largement utilisé dans l'industrie et l'enseignement pour sa robustesse, sa compatibilité avec divers outils et sa capacité à gérer des projets de grande envergure [W19].

3.2.6.2. Git/GitHub

Git est un système de gestion de versions distribué, créé par Linus Torvalds, qui permet de suivre les modifications du code source, de gérer les branches et de collaborer efficacement au sein d'une équipe [37]. Quant à **GitHub**, c'est une plateforme en ligne qui exploite Git pour héberger des dépôts de code, faciliter le travail collaboratif, la revue de code et l'intégration continue. Ces outils sont devenus des standards dans le développement logiciel moderne, car ils assurent la traçabilité des modifications, simplifient la coordination entre plusieurs développeurs et permettent de revenir facilement à une version précédente en cas de problème [W20].

¹⁷NPM est un outil central de l'écosystème JavaScript, utilisé principalement avec Node.js pour gérer les dépendances d'un projet.

¹⁸ Le DOM est une interface de programmation qui permet à des langages comme JavaScript de lire le contenu HTML modifier les éléments mais aussi gérer les événements

3.2.6.3. Google chrome

Google Chrome est un navigateur web développé par Google, largement utilisé pour l'accès aux applications web et aux systèmes d'information en ligne. Il repose sur le moteur de rendu Blink et le moteur JavaScript V8, ce qui lui permet d'offrir des performances élevées et une exécution rapide des applications web modernes [38]. Google Chrome intègre également des outils avancés destinés aux développeurs, tels que les Chrome DevTools, facilitant le débogage, l'analyse des performances et l'inspection du code des applications web. Grâce à sa compatibilité avec les standards du web et à son système d'extensions, il constitue un environnement essentiel pour le développement et l'utilisation d'applications web interactives.

3.3. Analyse du SI

3.3.1. User CASE

Le diagramme de cas d'utilisation est un diagramme comportemental du langage UML permettant de représenter les interactions entre un système et ses acteurs externes. Il vise principalement à capturer les exigences fonctionnelles du système du point de vue de l'utilisateur, sans décrire les mécanismes internes d'implémentation. Les diagrammes de cas d'utilisation constituent un outil central dans l'analyse des exigences, car ils permettent de structurer et de clarifier les besoins fonctionnels en identifiant les interactions essentielles entre les utilisateurs et le système. Les auteurs démontrent expérimentalement que les use cases facilitent la compréhension des exigences lorsqu'ils sont utilisés conjointement avec d'autres modèles UML [39].

Un diagramme de cas d'utilisation comprend généralement :

- **Acteur** : C'est une entité externe qui interagit avec le système. Les acteurs peuvent être des utilisateurs humains, d'autres systèmes, des dispositifs matériels etc;
- **Cas d'utilisation** : il correspond à une fonctionnalité ou un service que le système fournit à un acteur. Il décrit un objectif atteint par l'acteur à travers une interaction avec le système ;
- **Relation entre acteur et cas d'utilisation** : Les acteurs sont reliés aux cas d'utilisation par des lignes, indiquant l'interaction entre l'acteur et le système. On peut citer les types de relation :
 - **Association** : c'est un lien simple entre un acteur et un cas d'utilisation ;
 - **Include (inclusion)** : indique qu'un cas d'utilisation est toujours réalisé quand un autre cas d'utilisation est effectué ;

- **Extend (extension)** : un cas d'utilisation ajoute un comportement optionnel à un autre ;
 - **Généralisation** : permet de montrer qu'un cas d'utilisation hérite des caractéristiques d'un autre cas d'utilisation ;
- **Le système** : est représentée par un rectangle englobant les cas d'utilisation. Il délimite clairement ce qui appartient au système et celui des acteurs.

Ces éléments permettent de représenter les exigences à un niveau conceptuel élevé, facilitant la communication entre analystes, développeurs et parties prenantes [39].

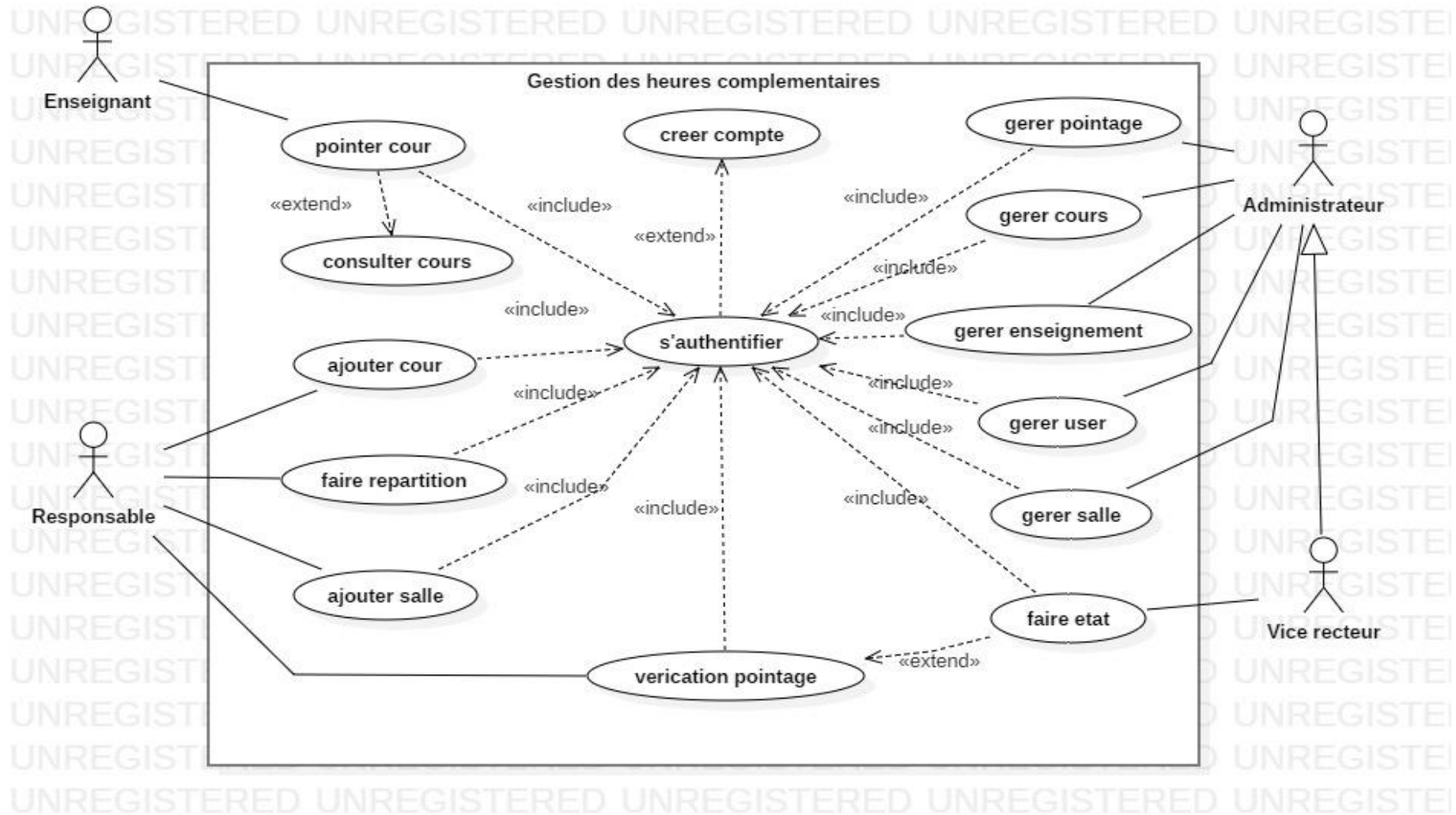


Figure 15: Diagramme de cas d'utilisation

Ce diagramme de cas d'utilisation illustre le fonctionnement du système de gestion des heures complémentaires. On y distingue plusieurs acteurs principaux : l'Enseignant, le Responsable, l'Administrateur et le Vice-recteur, chacun ayant des rôles bien définis.

L'Enseignant : L'enseignant est un acteur principal du système pour la saisie des heures complémentaires. Il doit s'authentifier avant de pointer ses cours, consulter son emploi du temps ou ses cours déjà enregistrés, et éventuellement ajouter des informations liées à ses enseignements. Son rôle est centré sur l'exécution et la déclaration des activités pédagogiques.

L'Administrateur : assure la gestion globale du système. Après authentification, il gère les utilisateurs, les cours, les enseignements, les salles ainsi que le pointage. Il garantit le bon fonctionnement technique et organisationnel de la plateforme en contrôlant toutes les données.

Le responsable : intervient dans l'organisation académique. Il doit s'authentifier pour répartir les cours entre les enseignants (faire la répartition) et gérer les ressources comme les salles. Son rôle est stratégique dans la planification et l'affectation des enseignements.

Le Vice-recteur : est un acteur de niveau supérieur qui **hérite des fonctionnalités de l'administrateur**. Cela signifie qu'il peut également gérer les utilisateurs, les cours, les enseignements, les salles et le pointage. En plus de ces droits, il assure un rôle de contrôle en vérifiant les pointages et en produisant des états pour le suivi et la prise de décision.

3.3.2. Diagramme de classe

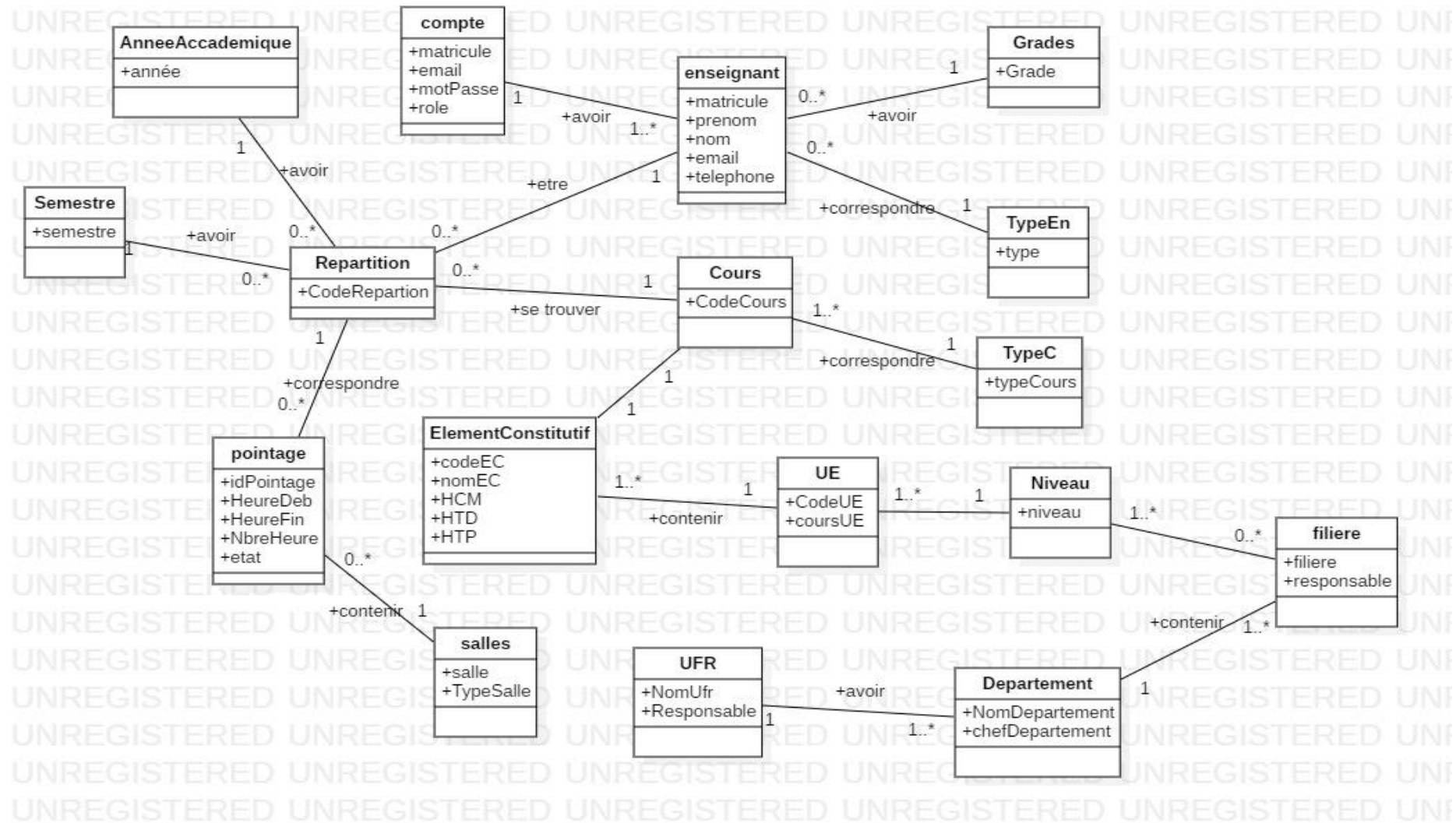


Figure 16: Diagramme de classe

Ce diagramme représente les différentes classes intervenant dans le système de gestion des enseignements et de suivi des heures. Dans notre diagramme, nous avons plusieurs classes qui structurent l'ensemble du système. Cependant, nous avons :

Une classe **"enseignant"** qui permet d'identifier chaque enseignant à travers son matricule et son prénom. Elle est liée à la classe **"grade"** pour connaître son niveau académique ainsi qu'à la classe **"TypeEn"** qui précise son type (permanent, non permanent). Cette classe permet donc de caractériser chaque acteur principal du système.

Une classe **"répartition"** qui joue un rôle central dans le système. Elle permet d'associer un enseignant à un cours, tout en prenant en compte le semestre et l'année académique. Cette classe nous informe sur l'affectation des enseignements et permet de savoir qui enseigne quoi et à quel moment.

Une classe **"cours"** qui nous fournit les informations sur les différents enseignements. Elle est liée à la classe **"TypeC"** afin de préciser la nature du cours (CM, TD, TP). Elle est également rattachée à la classe **"ElementConstitutif"**, ce qui permet de structurer les enseignements.

Une classe **"ElementConstitutif"** qui regroupe les cours appartenant à un même module. Elle est liée à la classe **"UE"**, ce qui permet d'organiser les enseignements en unités pédagogiques cohérentes.

Une classe **"UE"** qui permet d'identifier les unités d'enseignement à travers un code et un nom. Elle est liée à la classe **"Niveau"**, du fait qu'elle permet de situer chaque unité dans un niveau d'étude précis.

Une classe **"Niveau"** qui représente les différents niveaux académiques (Licence, Master, etc.). Elle est liée à la classe **"filiere"**, et permet d'identifier la spécialité ou la formation de l'étudiant.

Une classe **"filiere"** qui contient les informations sur les filières ainsi que leur responsable. Reliée à la classe **"Departement"**, elle permet de rattacher chaque filière à une structure administrative.

Une classe **"Departement"** qui nous renseigne sur les départements ainsi que leurs chefs. Elle est liée à la classe **"UFR"**, permettant ainsi de représenter la hiérarchie universitaire.

Une classe **"UFR"** se charge de l'identification des unités de formation et de recherche ainsi que leurs responsables.

Une classe **"pointage"** nous montre les heures effectuées (heure de début, heure de fin, état). Elle permet de suivre l'exécution réelle des enseignements et est liée à la classe **"répartition"** et **"salles"**, ceci permet de connaître à la fois le cours concerné et le lieu où il s'est déroulé.

Une classe **"salles"** met en évidence les différentes salles ainsi que leur type.

Une classe "**semestre**" définit les périodes académiques dans lesquelles les enseignements sont dispensés.

Une classe "**AnnéeAcademique**" : elle permet de situer les enseignements dans une année universitaire donnée.

Ainsi, ce diagramme de classe se charge de la modélisation de l'ensemble du système en intégrant la gestion des enseignants, des enseignements, de la structure académique ainsi que le suivi des heures effectuées.

3.3.3. Diagramme de séquence

- Créer un compte

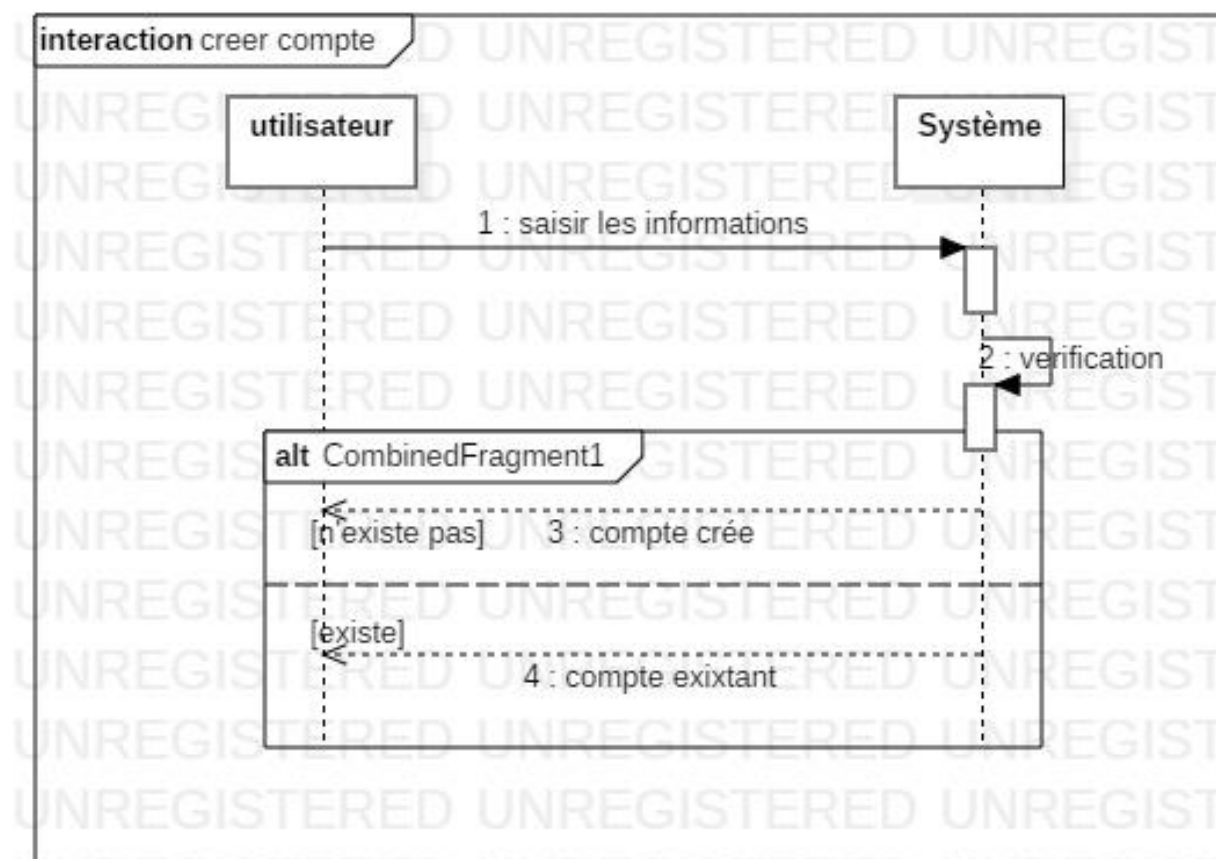


Figure 17: séquence pour créer un compte

Pour créer un compte, l'utilisateur fournit ses informations et le soumet. Le système vérifie d'abord si le compte existe déjà. Si ce n'est pas le cas, il envoie un message confirmant la création du compte ; sinon, une erreur est signalée.

➤ S'authentifier

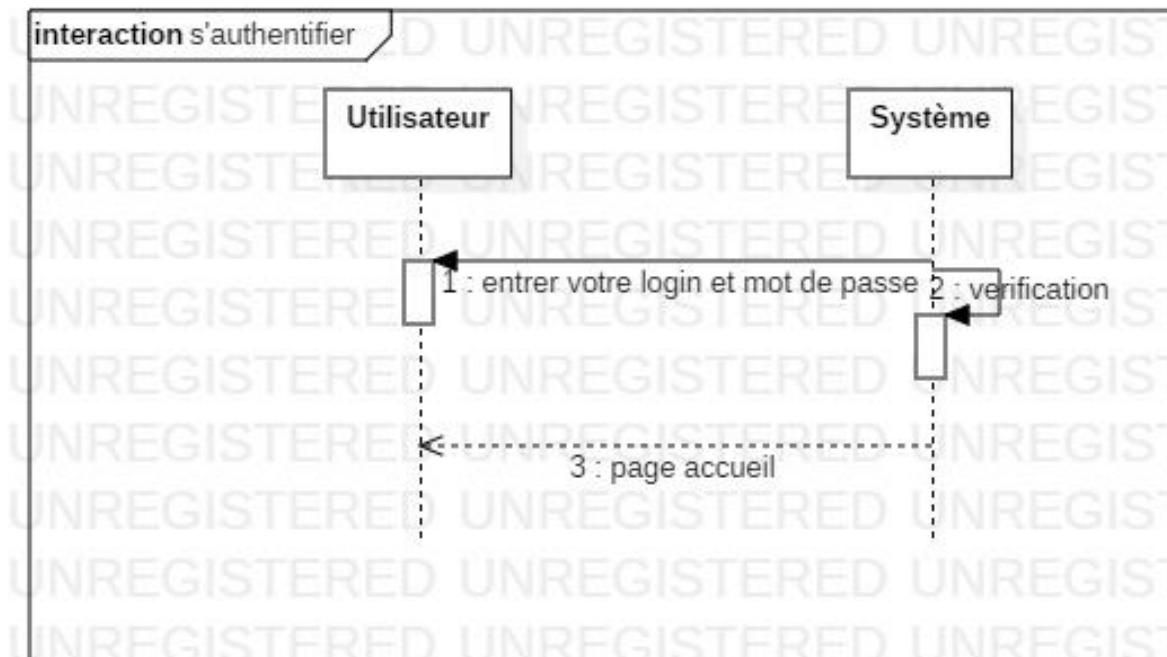


Figure 18: séquence pour s'authentifier

Pour s'authentifier, l'utilisateur saisit ses identifiants et que le système procède à la vérification. Si les informations sont correctes, l'accès est accordé à sa page d'accueil selon rôle; sinon, un message d'erreur est renvoyé.

➤ Pointer cours

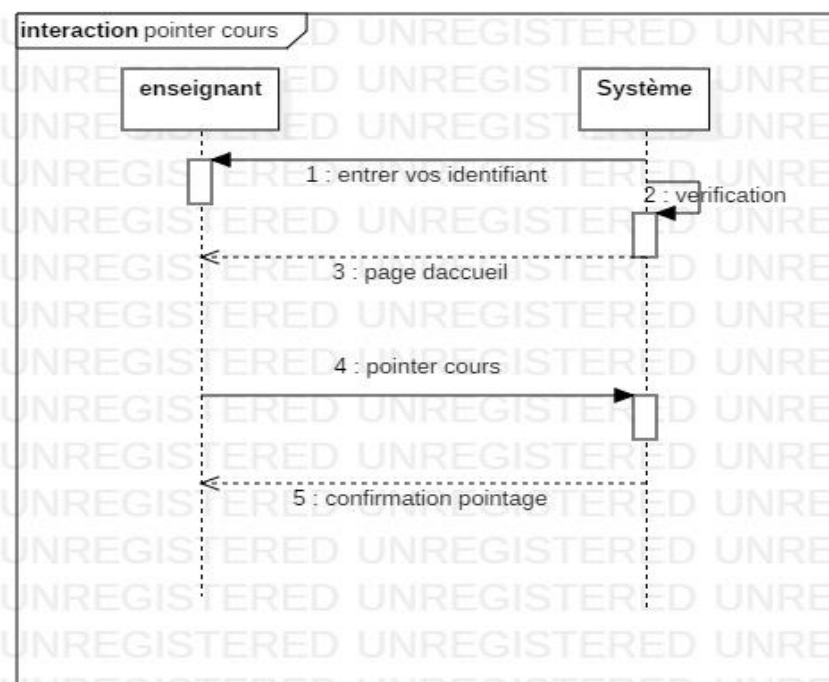


Figure 19: séquence pour pointer un cours

Ce scénario montre comment un enseignant enregistre un cours mais avant l'envoi l'enseignant doit s'authentifier d'abord et le système lui renvoie sa page d'accueil.

Après Il pourra renseigner les informations nécessaires pour pointer son cours, puis le système confirme l'enregistrement.

➤ Ajouter cours

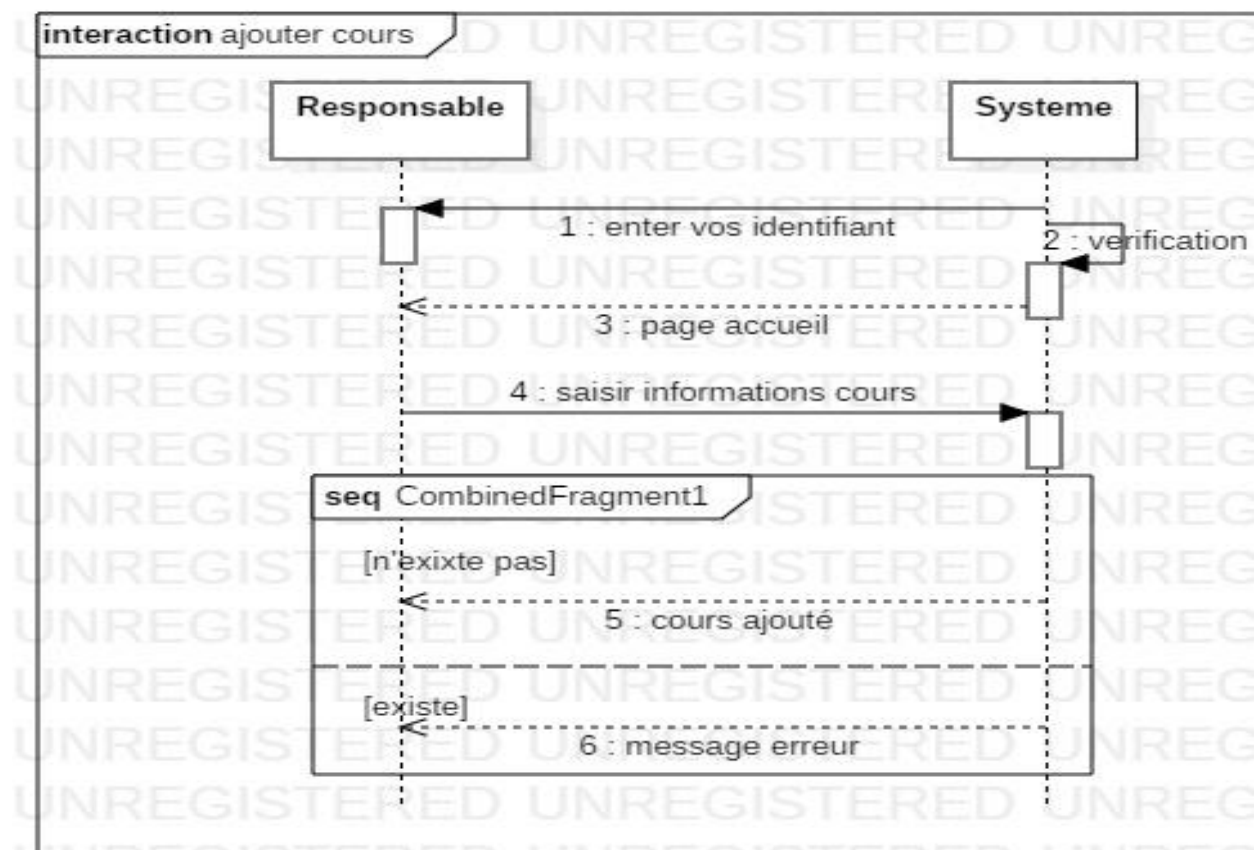


Figure 20: séquence pour ajouter un cours

Pour ajouter un cours, le responsable doit se connecté avant de pouvoir faire quelque chose. Si ses identifiant sont correcte, il renvoyer à sa page d'accueil ou il pourra ajouter un nouveau cours dans le système. Le système vérifie d'abord si ce cours existe déjà. S'il est nouveau, il est enregistré ; sinon, une erreur est retournée.

➤ Vérifier le pointage

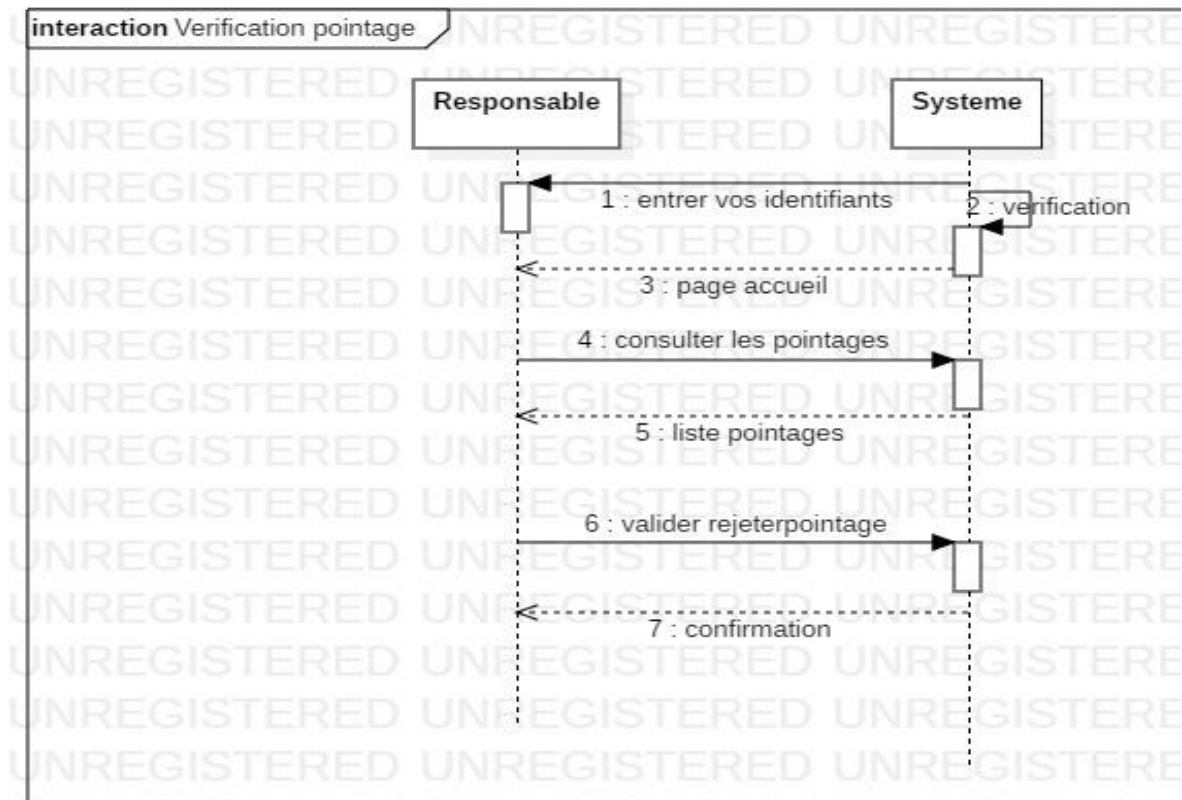


Figure 21: séquence pour vérifier le cours

Ici le responsable doit d'abord se connecter pour que le système lui renvoi sa page d'accueil. Sur la page d'accueil le responsable verra ses fonctionnalités et pourra consulter les pointages effectués par les enseignants. Après visualisation, il peut valider ou rejeter chaque pointage. Le système enregistre ensuite la décision et renvoie une confirmation.

3.3.4. Modèle relationnel

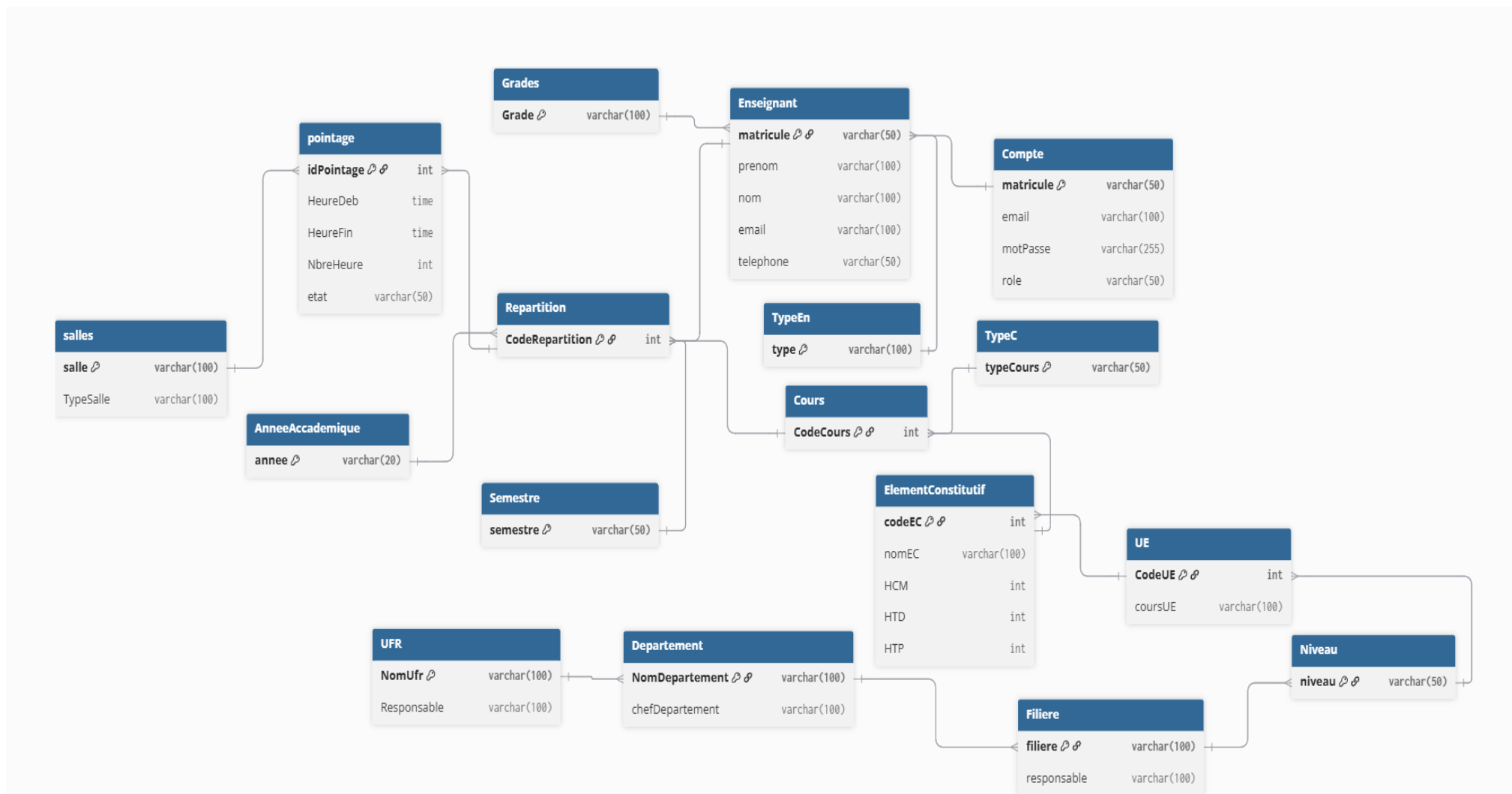


Figure 22: le modèle relationnel

3.3.5. Modèle physique de données

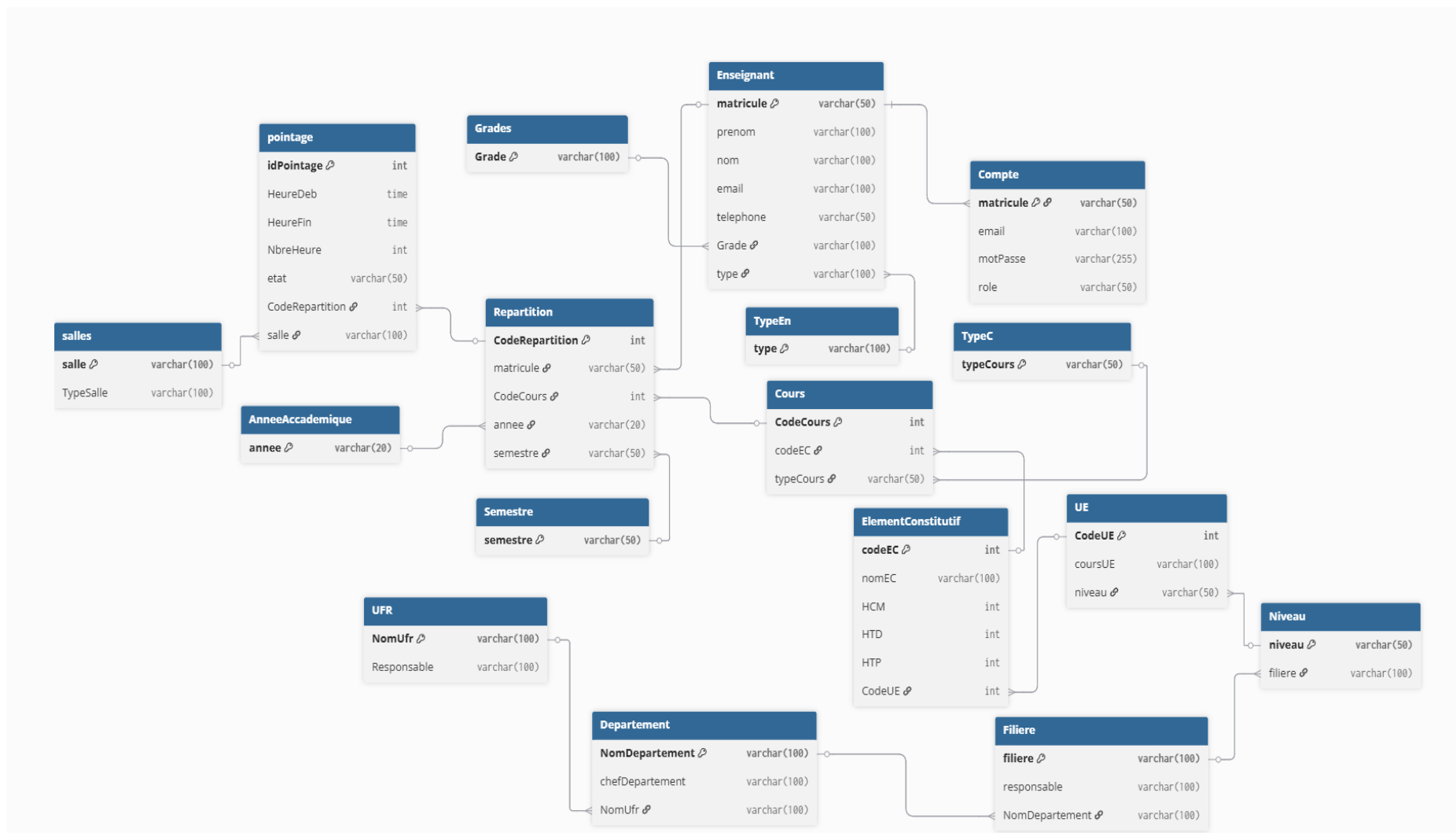


Figure 23: modèle physique de donnée

3.3.6. Création de la base de données

Pour la création de la base de données, nous allons nous connecter à l'interface de **PhpMyAdmin**. Une fois connecter, il suffit de cliquer sur nouvelle base de données et on renseigne le nom de notre base de données qui est « **ghc_uasz** » ; ensuite on clique sur créer et la base de données est créé.



Figure 24: création de la base de données

Après avoir créé la base de données, nous procédons à l'importation du fichier SQL de notre base de données.

Pour le faire on clique sur le menu « Importer », on choisit le fichier SQL à importer et on clique sur le bouton « Importer » qui est au-dessous.

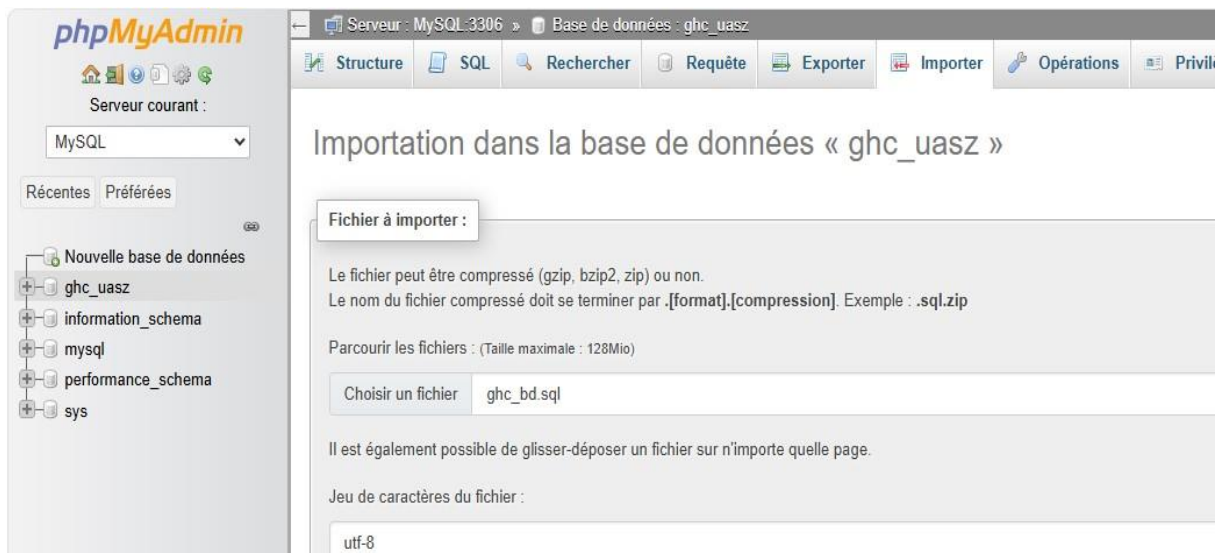


Figure 25: importation de la base de données

❖ La structure de la base donnée

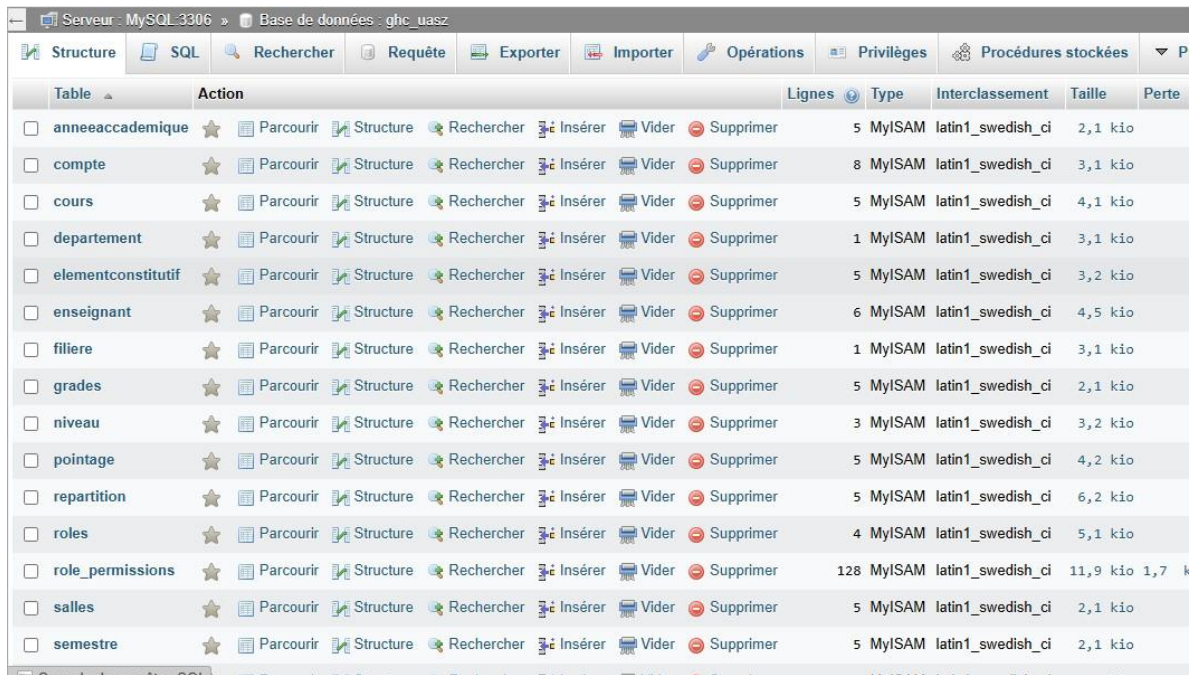


Table	Action	Lignes	Type	Interclassement	Taille	Perte
anneeacademique	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	2,1 kio	
compte	Parcourir Structure Rechercher Insérer Vider Supprimer	8	MyISAM	latin1_swedish_ci	3,1 kio	
cours	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	4,1 kio	
departement	Parcourir Structure Rechercher Insérer Vider Supprimer	1	MyISAM	latin1_swedish_ci	3,1 kio	
elementconstitutif	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	3,2 kio	
enseignant	Parcourir Structure Rechercher Insérer Vider Supprimer	6	MyISAM	latin1_swedish_ci	4,5 kio	
filier	Parcourir Structure Rechercher Insérer Vider Supprimer	1	MyISAM	latin1_swedish_ci	3,1 kio	
grades	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	2,1 kio	
niveau	Parcourir Structure Rechercher Insérer Vider Supprimer	3	MyISAM	latin1_swedish_ci	3,2 kio	
pointage	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	4,2 kio	
repartition	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	6,2 kio	
roles	Parcourir Structure Rechercher Insérer Vider Supprimer	4	MyISAM	latin1_swedish_ci	5,1 kio	
role_permissions	Parcourir Structure Rechercher Insérer Vider Supprimer	128	MyISAM	latin1_swedish_ci	11,9 kio	1,7 k
salles	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	2,1 kio	
semestre	Parcourir Structure Rechercher Insérer Vider Supprimer	5	MyISAM	latin1_swedish_ci	2,1 kio	

Figure 26: structure de base de données

3.4. Réalisation

3.4.1. Installation du framework

Pour démarrer notre projet, nous aurons besoin d'un framework de développement qui est PHPRAD.

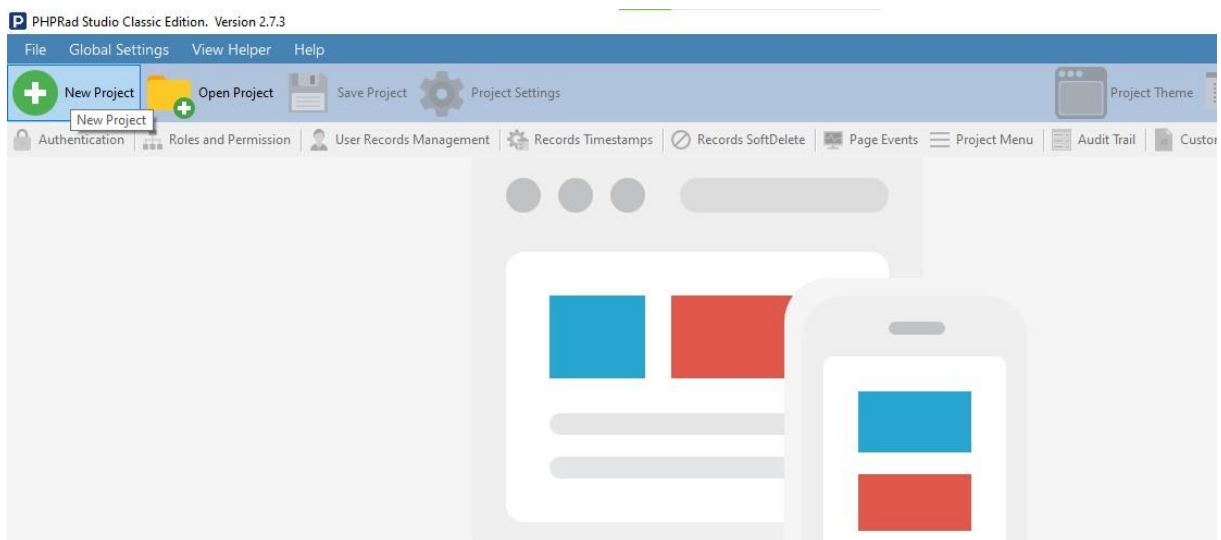


Figure 27: interface PHPRAD

3.4.2. Création du projet sur PHPRAD

Cette interface de création d'un nouveau projet, nous avons défini les informations principales du projet, notamment le nom du projet **GestionHCE**, son emplacement dans le serveur local **WAMP** et son adresse d'accès via le navigateur. Le projet est configuré pour fonctionner avec le serveur web **Apache** et utilise une base de données **MySQL** nommée **ghc_uasz**. La langue par défaut choisie est le **français**,

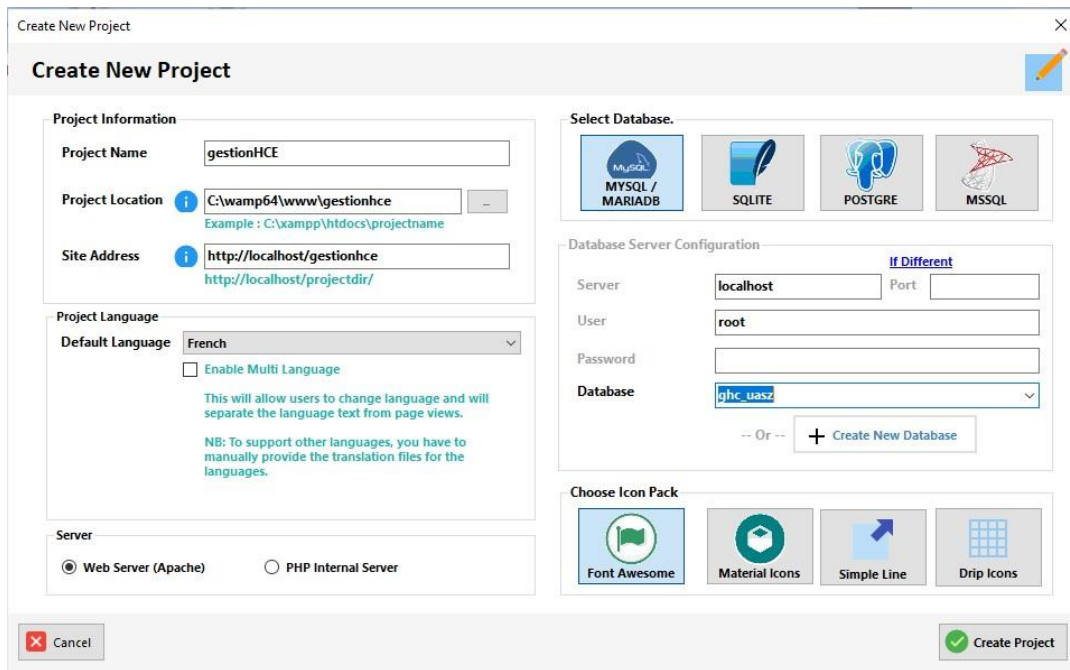


Figure 28; création du projet sur PHPRad

Une fois le projet créer nous verrons dans notre dossier local que le projet a bien été crée dans le C:\wamp64\www\gestionhce.

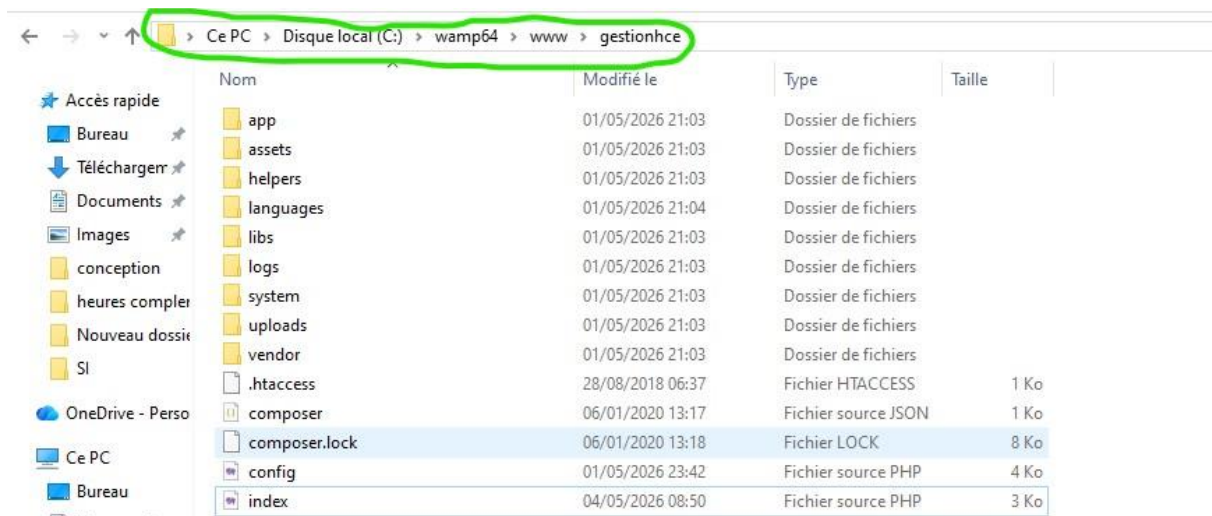


Figure 29: enregistrement du projet

Le fichier du projet est stocké différemment du dossier principal.

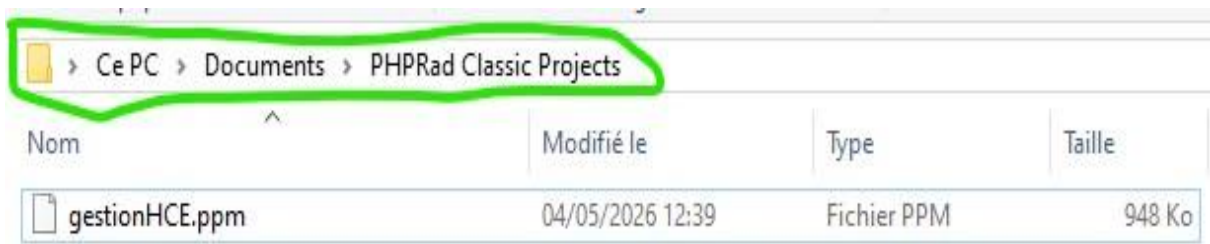


Figure 30: le fichier du projet

3.4.3. Le choix du thème

Pour le choix du thème, nous allons cliquer sur « **Project Theme** ». Dans cette interface, il y'a plusieurs types de Templates où on a la possibilité de choisir un thème que nous voulons en cliquant sur « Select Template ». Nous pouvons même choisir la couleur de fond et la police des écritures de la barre de navigation principale en cliquant sur « Navbar Variant ».

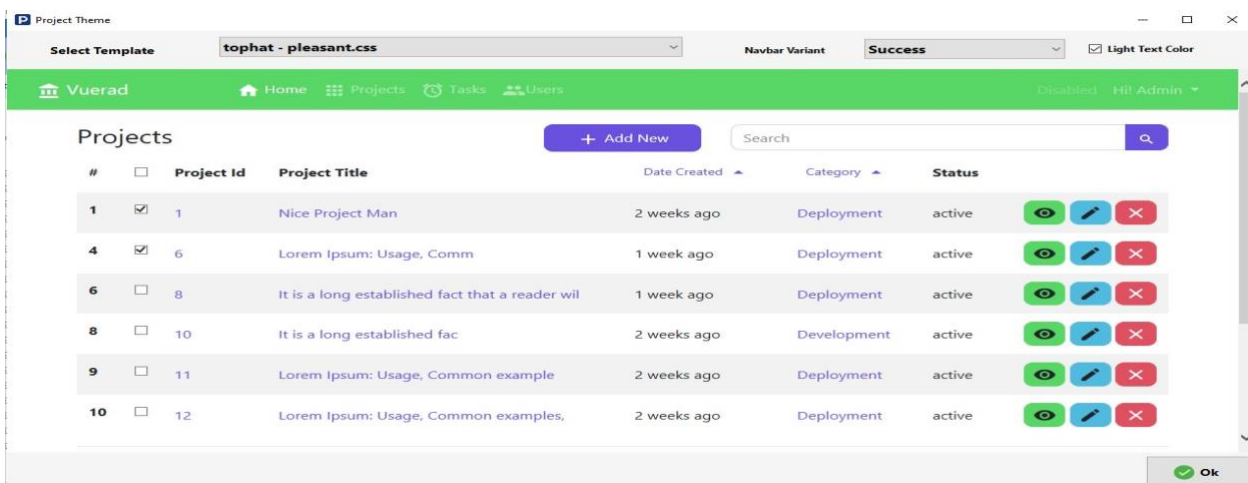
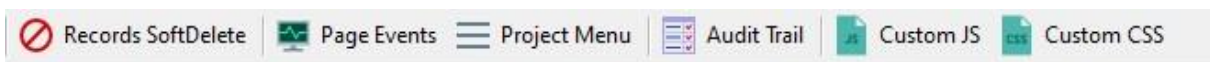


Figure 31: choix du thème

3.4.4. Organisation du menu

Cette interface permet d'ajouter, modifier ou supprimer les éléments du menu. Pour chaque menu, nous avons défini le chemin de la page (**Path**), le libellé à afficher (**Display Label**), la **cible d'ouverture (Target)** et l'**icône (Menu Icon)** correspondante.



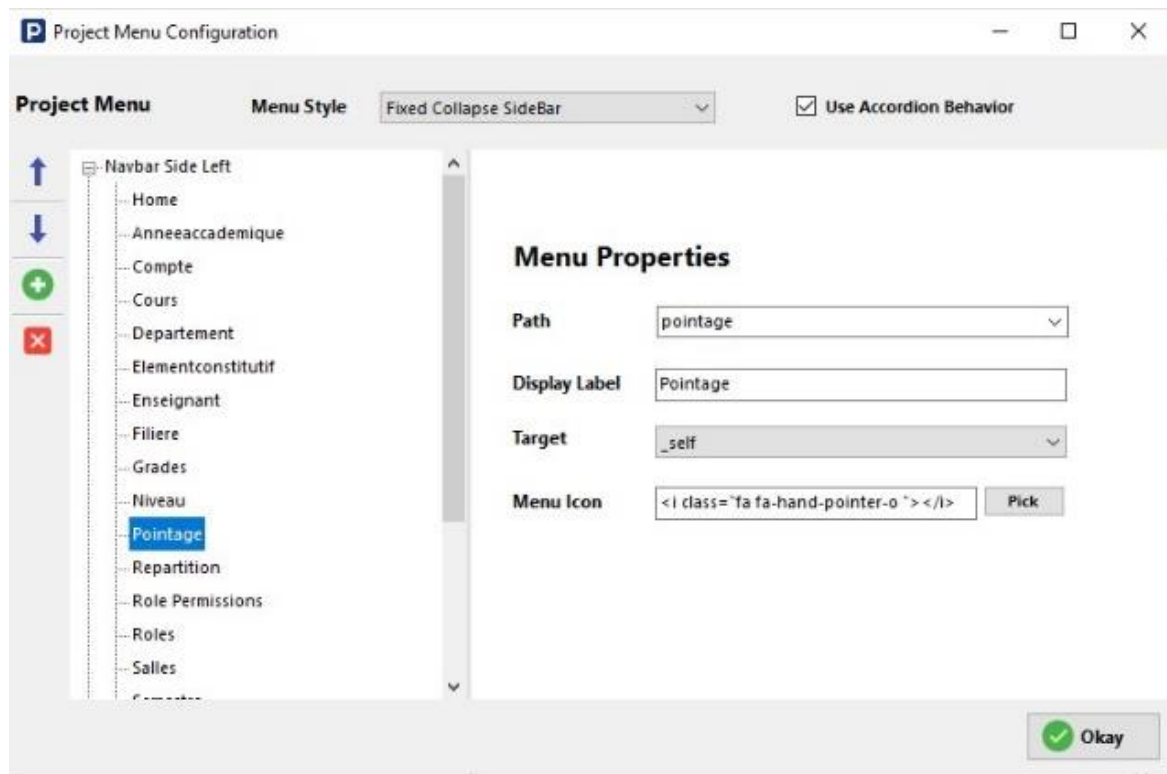


Figure 32: gestion des menus

3.4.5. Gestion des pages

Pour la gestion des pages, nous avons utilisé la section « **Database Tables** », dans laquelle toutes les tables de notre base de données sont affichées. À partir de cette section, il est possible de générer automatiquement les pages nécessaires pour chaque table. Ces pages sont composées de plusieurs modules :

- **List Page** : permet d'afficher la liste complète des enregistrements ;
- **View Page** : permet d'afficher les détails d'un enregistrement ;
- **Add Page** : permet d'ajouter un nouvel enregistrement ;
- **Edit Page** : permet de modifier un enregistrement existant ;
- **Delete Page** : permet de supprimer un enregistrement.

Il existe également une section appelée « **Page Fields** », qui permet de configurer les champs affichés dans chaque page. Elle permet de choisir les champs visibles, les champs utilisés dans les formulaires, ainsi que ceux qui peuvent être exploités dans la barre de recherche.

Enfin, la section « **Field Properties** » permet de personnaliser les propriétés d'un champ sélectionné, comme son type d'affichage, son alignement, sa relation avec une autre table ou encore son comportement dans les formulaires. Cette gestion facilite la création des interfaces

liées aux différents modules du système, notamment les enseignants, les cours, les répartitions, les salles et les pointages.

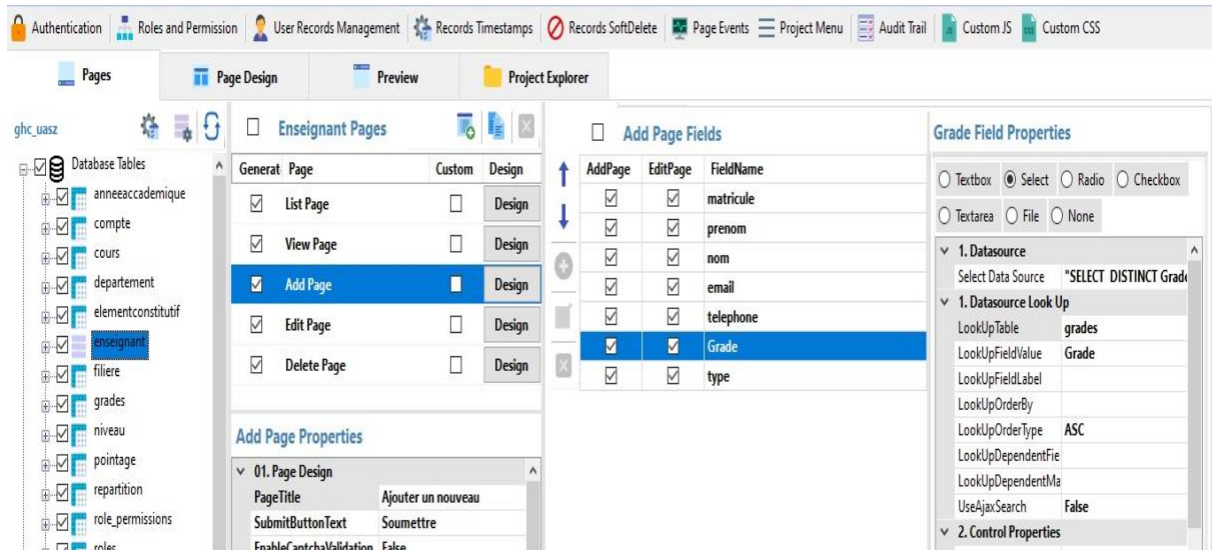


Figure 33: gestion des pages

À cela s'ajoute l'onglet « **Design** », qui permet de personnaliser l'apparence de chaque page générée. L'onglet « **Custom** » permet aussi d'effectuer des modifications manuelles, comme l'ajout de code HTML, CSS ou JavaScript, afin d'adapter davantage l'interface aux besoins du projet.

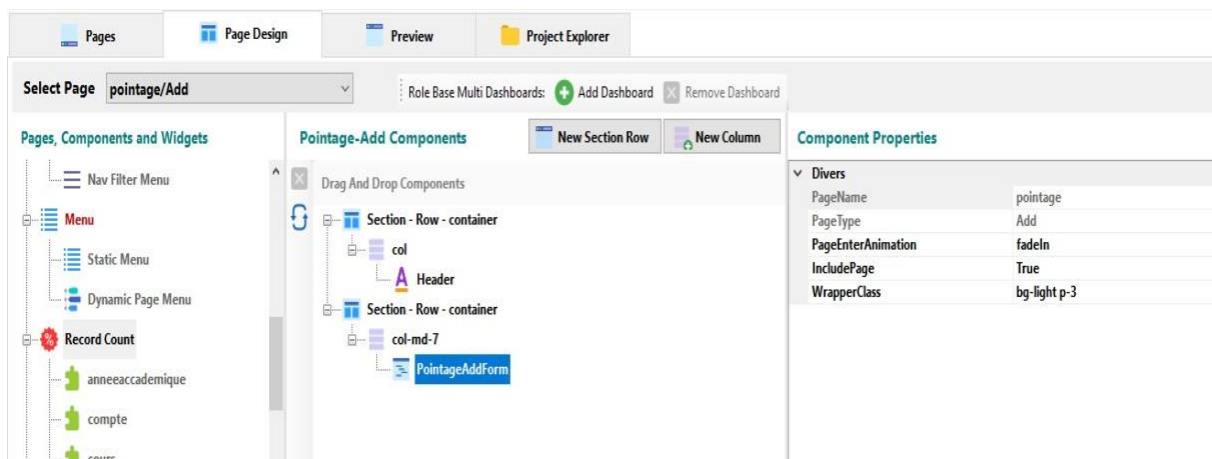


Figure 34: gestion Page Design

3.4.6. Gestion des authentifications

En cliquant sur le menu authentification, nous verrons l'onglet Login and registration. La table compte a été choisie comme table de connexion. Les champs **matricule**, **motPasse** et **email** ont

été associés aux informations d'identification de l'utilisateur. Cette configuration permet aux utilisateurs de se connecter avec leur matricule ou leur email.

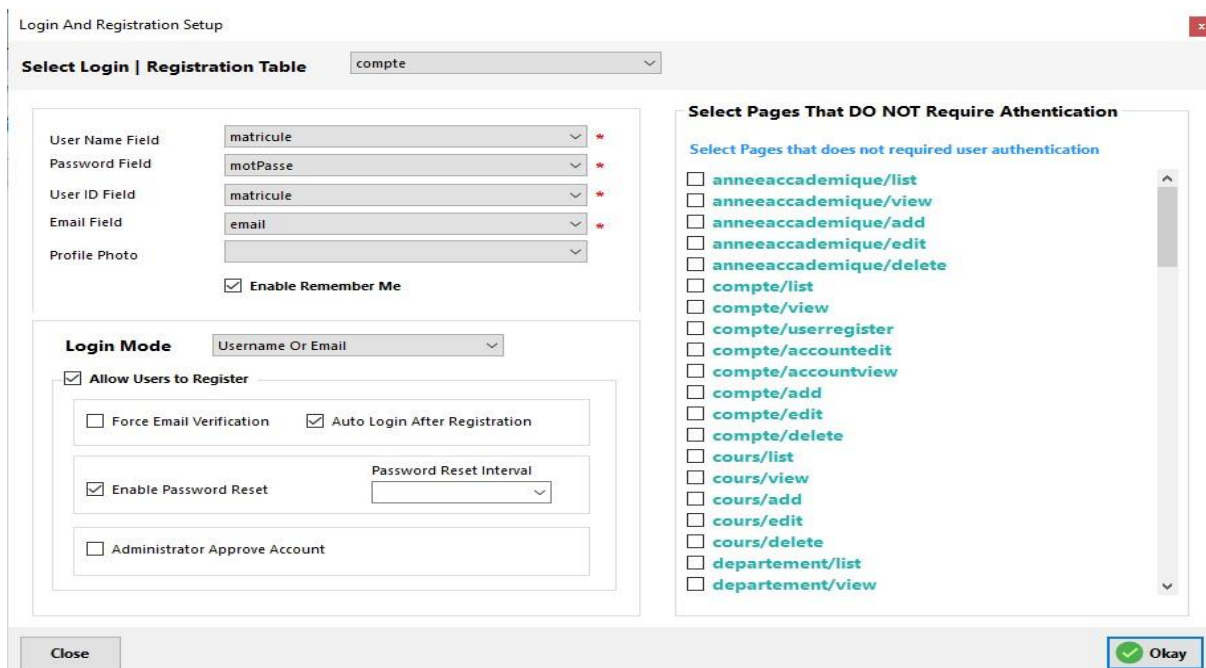
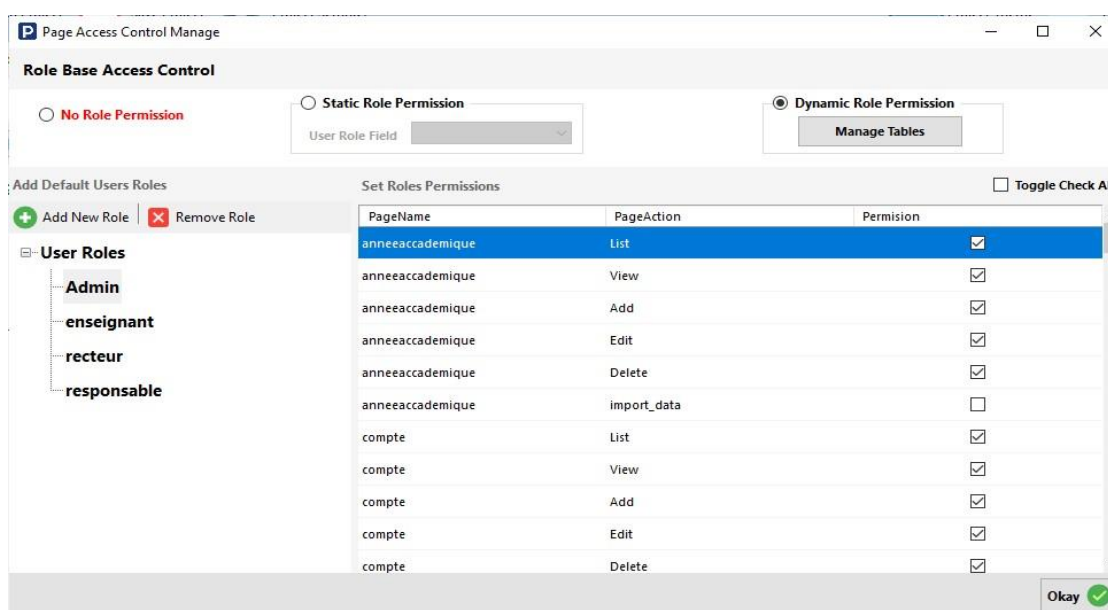


Figure 35: Onglet Authentification

3.4.7. Gestion des rôles

Cette partie permet de définir les différents profils d'utilisateurs du système, comme l'enseignant, le responsable, l'administrateur ou le recteur. Chaque rôle est ensuite associé à des permissions précises, afin de limiter ou d'autoriser l'accès à certaines pages et fonctionnalités.

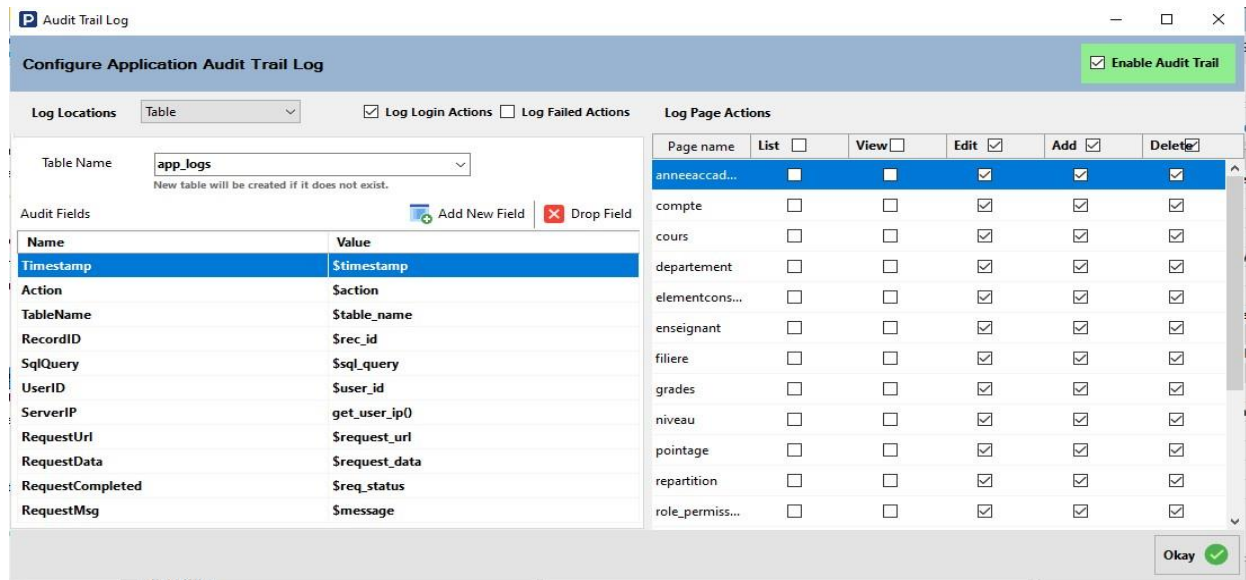


PageName	PageAction	Permission
anneeacademique	List	☒
anneeacademique	View	☒
anneeacademique	Add	☒
anneeacademique	Edit	☒
anneeacademique	Delete	☒
anneeacademique	import_data	☐
compte	List	☒
compte	View	☒
compte	Add	☒
compte	Edit	☒
compte	Delete	☒

Figure 36: gestion des rôles

3.4.8. Gestion de l'audite

Cette fonctionnalité permet d'enregistrer certaines opérations effectuées par les utilisateurs, comme l'**ajout**, la **modification** ou la **suppression** des données. Elle permet aussi d'identifier l'utilisateur concerné, l'action réalisée et la date de l'opération.



Configure Application Audit Trail Log

☒ Enable Audit Trail

Log Locations: Table ☒ Log Login Actions ☐ Log Failed Actions

Log Page Actions

Page name	List	View	Edit	Add	Delete
anneecacad...	☐	☐	☒	☒	☒
compte	☐	☐	☒	☒	☒
cours	☐	☐	☒	☒	☒
departement	☐	☐	☒	☒	☒
elementcons...	☐	☐	☒	☒	☒
enseignant	☐	☐	☒	☒	☒
filiere	☐	☐	☒	☒	☒
grades	☐	☐	☒	☒	☒
niveau	☐	☐	☒	☒	☒
pointage	☐	☐	☒	☒	☒
repartition	☐	☐	☒	☒	☒
role_permiss...	☐	☐	☒	☒	☒

Audit Fields

Name	Value
Timestamp	\$timestamp
Action	\$action
TableName	\$table_name
RecordID	\$rec_id
SqlQuery	\$sql_query
UserID	\$user_id
ServerIP	\$get_user_ip()
RequestUrl	\$request_url
RequestData	\$request_data
RequestCompleted	\$req_status
RequestMsg	\$message

Okay

Figure 37: gestion de l'audite

3.5. Conclusion

Ce chapitre nous a permis de présenter l'analyse, la conception et les choix techniques retenus pour la réalisation du système de gestion des heures complémentaires. À travers l'identification des acteurs, la définition des besoins fonctionnels, la modélisation du système et le choix des outils de développement, nous avons posé les bases nécessaires à la mise en œuvre de la plateforme. Cette étape a donc constitué un passage essentiel entre l'étude théorique du projet et la réalisation concrète de la solution présentée dans le chapitre suivant



Chapitre 4 : présentation de la solution

2.5. Accès à la plateforme

Pour accéder à la plateforme de gestion des heures complémentaires, l'utilisateur doit saisir l'adresse suivante dans le navigateur : <http://localhost/gestionhce>. Cette adresse permet d'ouvrir la page d'accueil de l'application. À partir de cette page, l'utilisateur peut accéder à la plateforme s'il possède déjà un compte et pourra voir son espace de travail selon son rôle dans le système. Par exemple, un enseignant accède à ses cours, à ses pointages et au suivi de ses heures complémentaires, tandis qu'un administrateur ou un responsable dispose d'un accès plus large. Pour cela, il doit renseigner son nom d'utilisateur ou son adresse email, saisir son mot de passe, puis cliquer sur le bouton « S'identifier ».

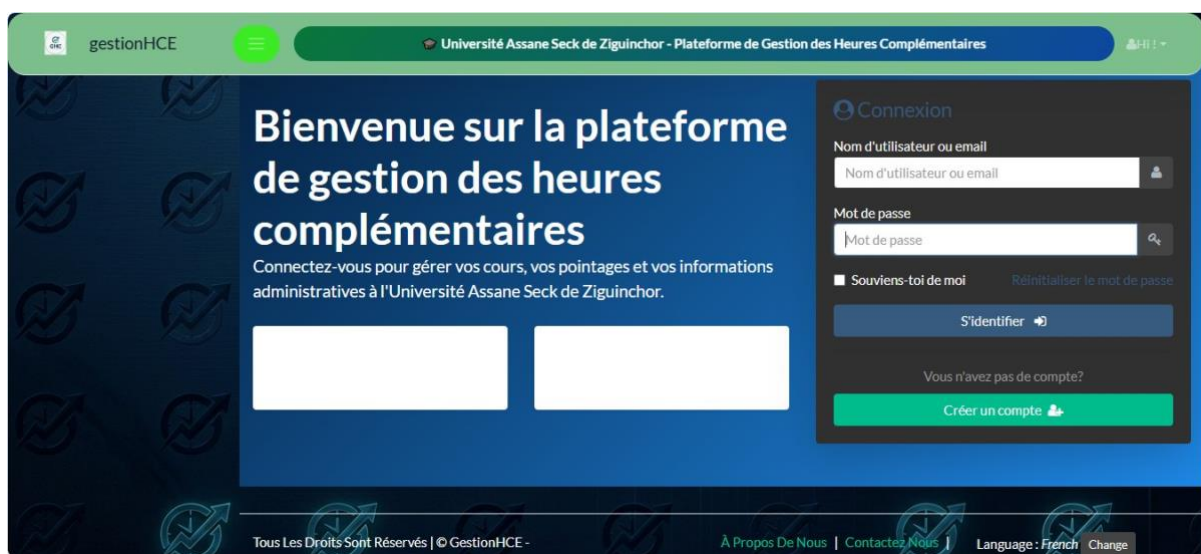


Figure 38: page d'accueil plateforme

2.6. Création de compte

Pour créer un compte, l'utilisateur clique sur « **Créer un compte** » et renseigne les informations d'authentification demandées, notamment le **matricule**, l'**adresse email**, le **mot de passe**, la confirmation du mot de passe et le **rôle**. Après avoir cliqué sur « **Soumettre** », le compte est enregistré dans la table **compte**, puis l'utilisateur est redirigé vers le formulaire d'ajout d'un enseignant afin de compléter son profil avec ses informations personnelles et administratives. Une fois ce formulaire validé, les données sont enregistrées dans la table **enseignant** et l'utilisateur accède à son espace enseignant.

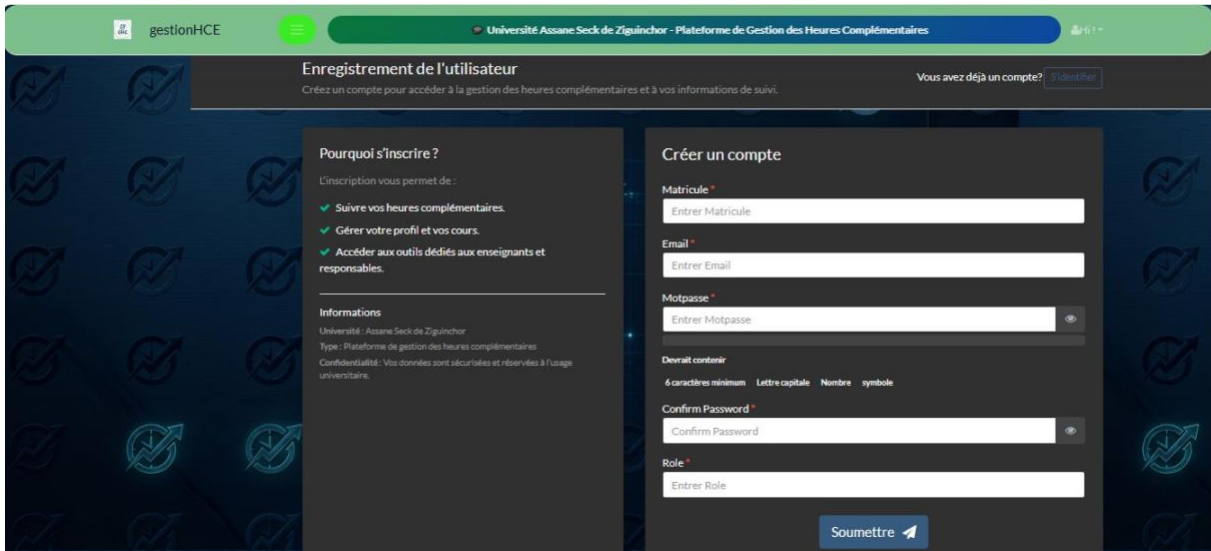


Figure 39: creer un compte

2.7. Espace enseignant

Dans l'espace enseignant, l'utilisateur accède aux fonctionnalités qui lui permettent de gérer ses heures complémentaires. Après authentification, il peut effectuer un pointage de son cours réalisés, consulter ses cours, consulter l'historique de ses pointages et suivre l'état de validation de ses heures. Lorsqu'un enseignant enregistre un pointage, celui-ci est automatiquement mis « **En attente** » afin d'être vérifié et validé par les responsables.

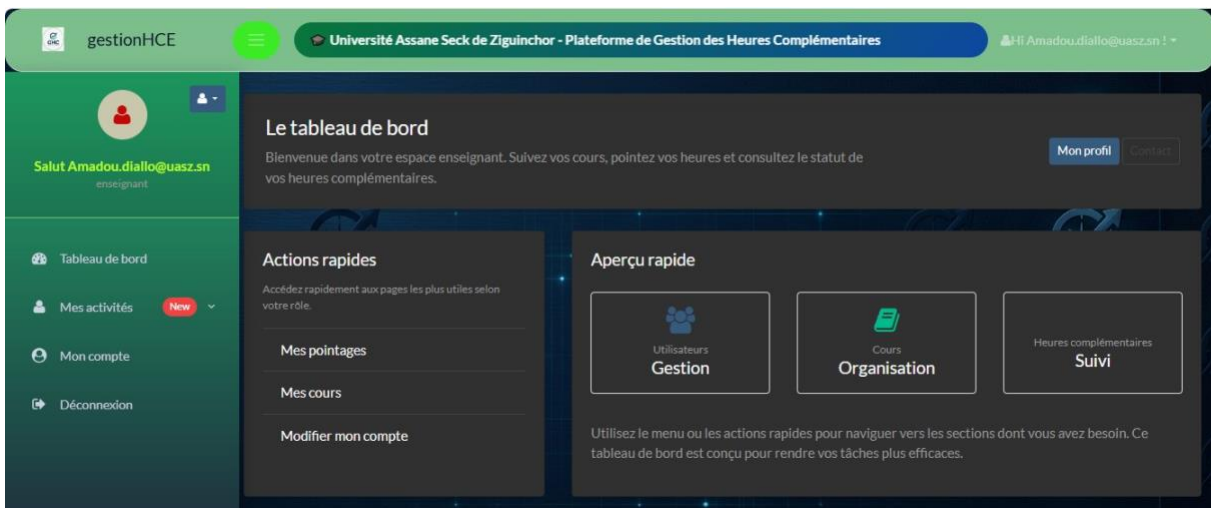
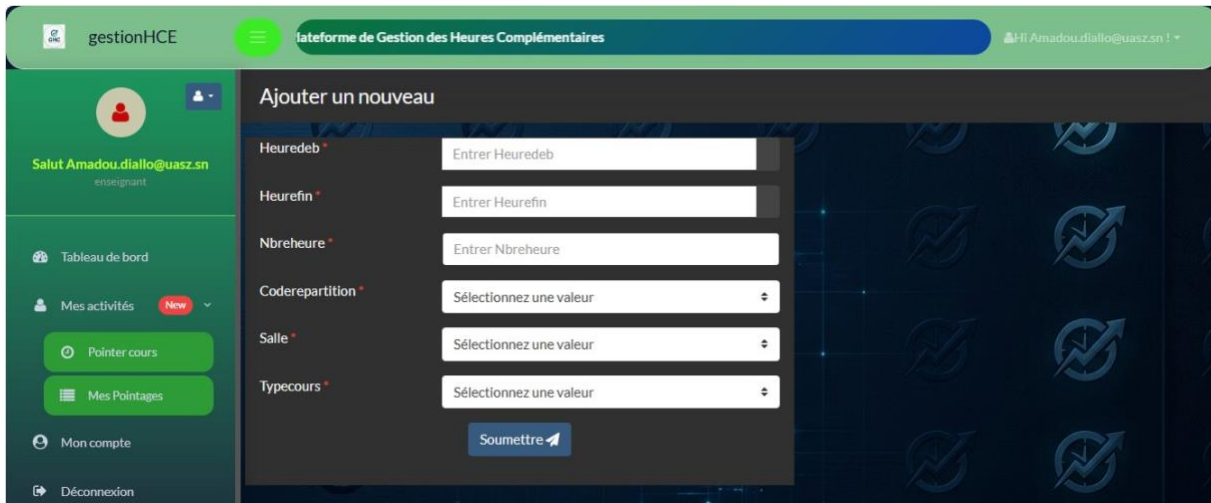


Figure 40: tableau de bord enseignant

4.1.1. Pointer un cours

L'enseignant peut pointer un cours à partir de son espace personnel en cliquant sur « Pointer cours ». Un formulaire s'affiche et lui permet de renseigner l'heure de début, l'heure de fin, le nombre d'heures effectuées, le cours concerné, la salle et le type de cours. Après avoir rempli

les informations demandées, il clique sur « Soumettre ». Le pointage est alors enregistré automatiquement avec l'état « En attente », afin d'être vérifié et validé par le responsables.



The screenshot shows the 'Ajouter un nouveau' form in the 'gestionHCE' application. The form is titled 'Ajouter un nouveau' and is located in the 'Plateforme de Gestion des Heures Complémentaires' section. The form includes the following fields:

- Heuredeb: A text input field with the placeholder 'Entrer Heuredeb'.
- Heurefin: A text input field with the placeholder 'Entrer Heurefin'.
- Nbreheure: A text input field with the placeholder 'Entrer Nbreheure'.
- Coderepartition: A dropdown menu with the placeholder 'Sélectionnez une valeur'.
- Salle: A dropdown menu with the placeholder 'Sélectionnez une valeur'.
- Typecours: A dropdown menu with the placeholder 'Sélectionnez une valeur'.

A 'Soumettre' button is located at the bottom of the form.

Figure 41: pointer un cours

4.1.2. Voir les pointages

Pour consulter ses pointages, l'enseignant clique sur « Mes Pointages ». La page affiche la liste des cours déjà pointés avec les informations comme l'heure de début, l'heure de fin, le nombre d'heures, la salle, le cours concerné, le semestre, l'année académique, le tarif et le montant. Elle permet aussi à l'enseignant de suivre l'état de ses pointages.



gestionHCE



 Université Assane Seck de Ziguinchor - Plateforme de Gestion des Heures Complémentaires

 Amanlou.diallo@uas-z.sn



Salut Amanlou.diallo@uas-z.sn

amougant



Tableau de bord

Mes activités

New

 Pointer cours

 Mes Pointages

Mon compte

Déconnexion

Vue Mes Pointages



#	Idpointage	Heuredeb	Heurefin	Nbreheure	Etat	Salle	Codercours	Année	Semestre	Codenc	Typecours	Nomec	Codecours	Coursue	Niveau	Tarif	Montant
1	8	08:00:00	10:00:00	2	En attente	Salle Info 1	MIO12111	2024-2025	Semestre 2	MIO1211 TP	Technologies de l'information et de la communication	MIO121 - Informatique de Gestion 2	MIO121	Licence 1	5000	0	

 Exportation

Records: 1 sur 1

























Figure 42: la liste des pointages

2.8. Espace responsable

L'espace du responsable vous permet de gérer les activités pédagogiques liées aux heures complémentaires. Après authentification, le responsable peut accéder rapidement à la répartition des cours, à la gestion des salles et au planning des enseignements. Cet espace lui permet aussi de suivre l'organisation pédagogique et de consulter les informations nécessaires afin d'assurer un meilleur contrôle des cours programmés et réalisés.

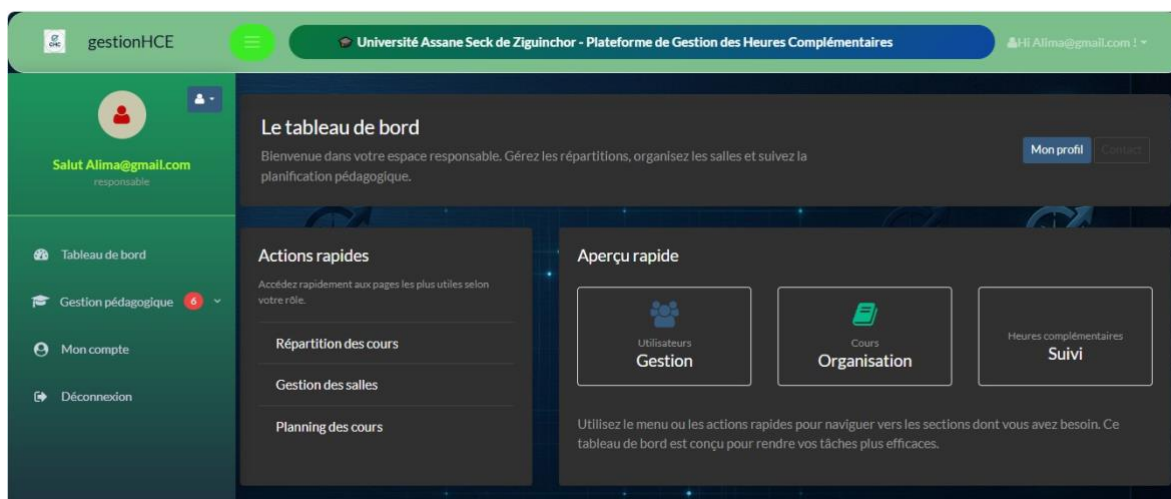


Figure 43: tableau de bord responsable

4.1.3. Faire la répartition

Pour faire la répartition des cours, le responsable clique sur « **Répartition** » dans le menu Gestion pédagogique. La page affiche la liste des répartitions déjà enregistrées avec les informations comme le **matricule de l'enseignant**, le **code du cours**, l'**année académique** et le **semestre**. Le responsable peut aussi cliquer sur « **Ajouter un nouveau** » pour affecter un cours à un enseignant. Il peut ensuite **consulter**, **modifier** ou **supprimer** une répartition selon les besoins de l'organisation pédagogique.



Figure 44: faire la répartition

4.1.4. Gestion des cours

Pour gérer les cours, le responsable accède à la rubrique « **Cours** » dans le menu Gestion pédagogique. Cette page affiche la liste des cours enregistrés avec les informations comme le code du cours, le code de l'élément constitutif et le type de cours. Le responsable peut ajouter un nouveau cours, consulter les détails d'un cours existant ou le modifier.

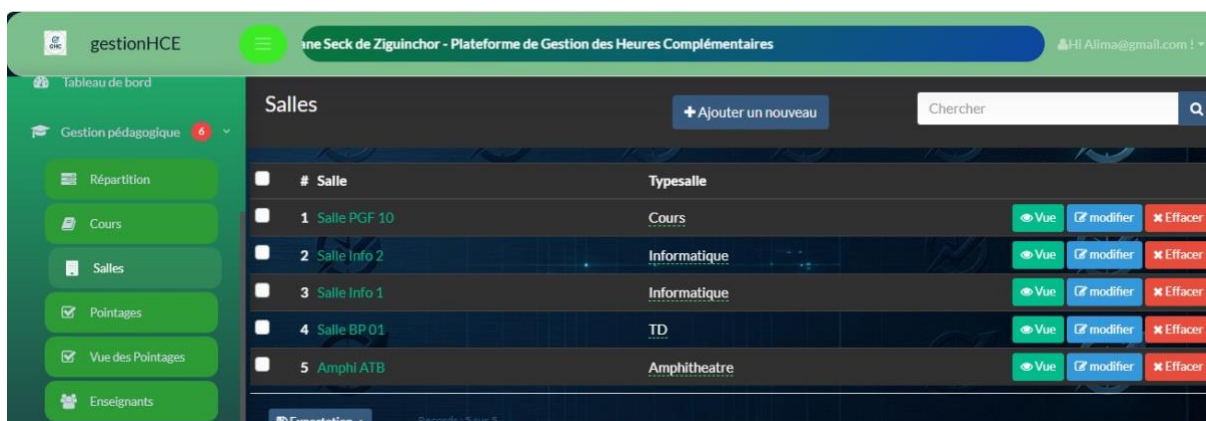


#	Codecours	Codeec	Typecours		
1	MIO31212	MIO3121	TD	Vue	modifier
2	MIO31211	MIO3121	CM	Vue	modifier
3	MIO24221	MIO2422	CM	Vue	modifier
4	MIO24121	MIO2412	CM	Vue	modifier
5	MIO12311	MIO1231	TP	Vue	modifier
6	MIO12211	MIO1221	TD	Vue	modifier
7	MIO12112	MIO1211	TD	Vue	modifier
8	MIO12111	MIO1211	TP	Vue	modifier

Figure 45: gestion des cours

4.1.5. Gestion des salles

Pour gérer les salles, le responsable accède à la rubrique « Salles » dans le menu Gestion pédagogique. Cette page permet d'afficher la liste des salles disponibles avec leurs informations principales. Le responsable peut ajouter une nouvelle salle, consulter ses détails ou modifier ses informations.



#	Salle	Typesalle			
1	Salle PGF 10	Cours	Vue	modifier	Effacer
2	Salle Info 2	Informatique	Vue	modifier	Effacer
3	Salle Info 1	Informatique	Vue	modifier	Effacer
4	Salle BP 01	TD	Vue	modifier	Effacer
5	Amphi ATB	Amphitheatre	Vue	modifier	Effacer

Figure 46: gestion des salles

4.1.6. Gestion des pointages

Pour gérer les pointages, le responsable accède à la rubrique « **Pointages** » dans le menu Gestion pédagogique. Cette page affiche la liste des pointages enregistrés par les enseignants avec l'heure de début, l'heure de fin, le nombre d'heures, l'état, la répartition et la salle. Le responsable peut consulter les détails d'un pointage, le modifier et vérifier son état afin de valider ou rejeter les heures déclarées selon leur conformité.

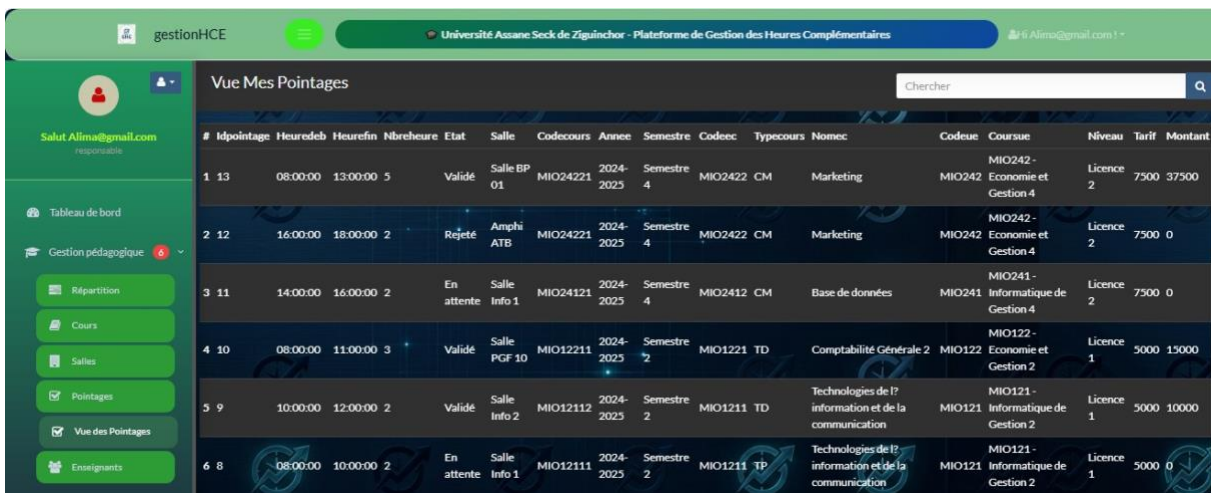


#	Idpointage	Heuredeb	Heurefin	Nbreheure	Etat	Coderepartition	Salle	
1	13	08:00:00	13:00:00	5	Validé	5	Salle BP 01	Vue modifier
2	12	16:00:00	18:00:00	2	Rejeté	5	Amphi ATB	Vue modifier
3	11	14:00:00	16:00:00	2	En attente	4	Salle Info 1	Vue modifier
4	10	08:00:00	11:00:00	3	Validé	3	Salle PGF 10	Vue modifier
5	9	10:00:00	12:00:00	2	Validé	2	Salle Info 2	Vue modifier
6	8	08:00:00	10:00:00	2	En attente	1	Salle Info 1	Vue modifier

Figure 47: gestion des pointages

4.1.7. Vue globale des pointages

Pour consulter la vue des pointages, le responsable clique sur « Vue des Pointages ». Cette page affiche les pointages détaillés avec les informations liées aux heures effectuées. Elle permet ainsi au responsable d'avoir une vision globale des heures complémentaires déclarées par les enseignants et de suivre leur état de traitement, notamment **validé, rejeté ou en attente**.

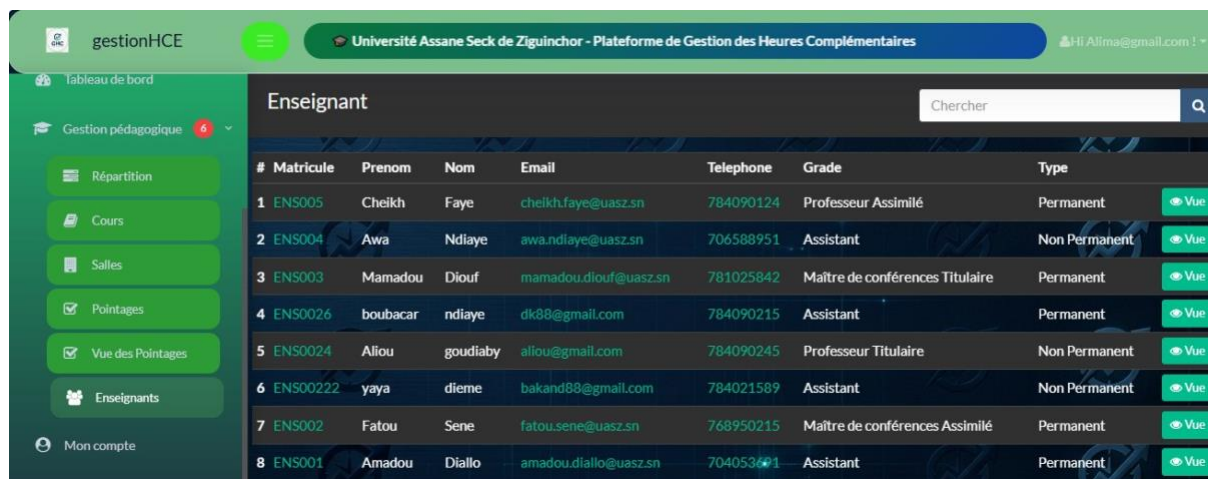


#	Idpointage	Heuredeb	Heurefin	Nbreheure	Etat	Salle	Codecours	Annee	Semestre	Codeec	Typecours	Nomec	Codeue	Coursue	Niveau	Tarif	Montant
1	13	08:00:00	13:00:00	5	Validé	Salle BP 01	MIO24221	2024-2025	Semestre 4	MIO2422	CM	Marketing	MIO242	MIO242 - Economie et Gestion 4	Licence 2	7500	37500
2	12	16:00:00	18:00:00	2	Rejeté	Amphi ATB	MIO24221	2024-2025	Semestre 4	MIO2422	CM	Marketing	MIO242	MIO242 - Economie et Gestion 4	Licence 2	7500	0
3	11	14:00:00	16:00:00	2	En attente	Salle Info 1	MIO24121	2024-2025	Semestre 4	MIO2412	CM	Base de données	MIO241	MIO241 - Informatique de Gestion 4	Licence 2	7500	0
4	10	08:00:00	11:00:00	3	Validé	Salle PGF 10	MIO12211	2024-2025	Semestre 2	MIO1221	TD	Comptabilité Générale 2	MIO122	MIO122 - Economie et Gestion 2	Licence 1	5000	15000
5	9	10:00:00	12:00:00	2	Validé	Salle Info 2	MIO12112	2024-2025	Semestre 2	MIO1211	TD	Technologies de l'information et de la communication	MIO121	MIO121 - Informatique de Gestion 2	Licence 1	5000	10000
6	8	08:00:00	10:00:00	2	En attente	Salle Info 1	MIO12111	2024-2025	Semestre 2	MIO1211	TP	Technologies de l'information et de la communication	MIO121	MIO121 - Informatique de Gestion 2	Licence 1	5000	0

Figure 48: vue complet des pointages

4.1.8. Vue sur les enseignants

Cette page affiche la liste des enseignants enregistrés avec leurs informations principales, notamment le matricule, le prénom, le nom, l'email, le téléphone, le grade et le type d'enseignant. Elle permet au responsable de consulter les profils des enseignants afin de mieux organiser la répartition des cours et le suivi des heures.



#	Matricule	Prenom	Nom	Email	Telephone	Grade	Type	
1	ENS005	Cheikh	Faye	cheikh.faye@uaszn	784090124	Professeur Assimilé	Permanent	Vue
2	ENS004	Awa	Ndiaye	awa.ndiaye@uaszn	706588951	Assistant	Non Permanent	Vue
3	ENS003	Mamadou	Diouf	amadou.diouf@uaszn	781025842	Maitre de conférences Titulaire	Permanent	Vue
4	ENS0026	boubacar	ndiaye	dk88@gmail.com	784090215	Assistant	Permanent	Vue
5	ENS0024	Aliou	goudiaby	aliou@gmail.com	784090245	Professeur Titulaire	Non Permanent	Vue
6	ENS00222	yaya	dieme	bakand88@gmail.com	784021589	Assistant	Non Permanent	Vue
7	ENS002	Fatou	Sene	fatou.sene@uaszn	768950215	Maitre de conférences Assimilé	Permanent	Vue
8	ENS001	Amadou	Diallo	amadou.diallo@uaszn	70405361	Assistant	Permanent	Vue

Figure 49: vue sur les enseignants

2.9. Espace Recteur

Le tableau de bord du recteur donne une vue globale sur la gestion des heures complémentaires. Après authentification, le recteur peut consulter les heures complémentaires déclarées, suivre les validations, accéder aux états et rapports, et superviser les informations pédagogiques. Cet espace lui permet d'avoir un aperçu général du fonctionnement du système et de prendre des décisions à partir des données disponibles.

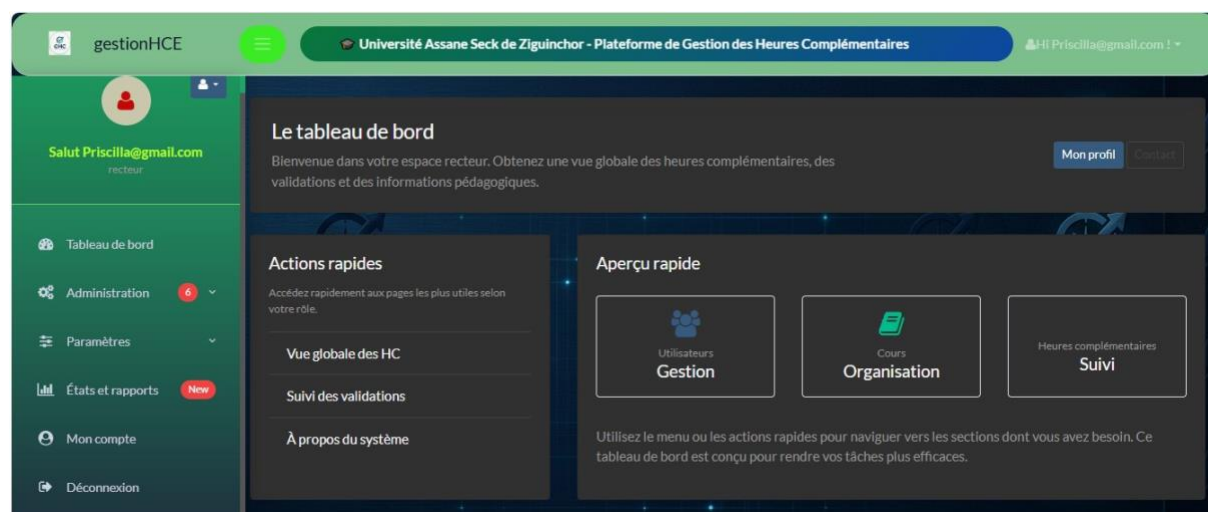
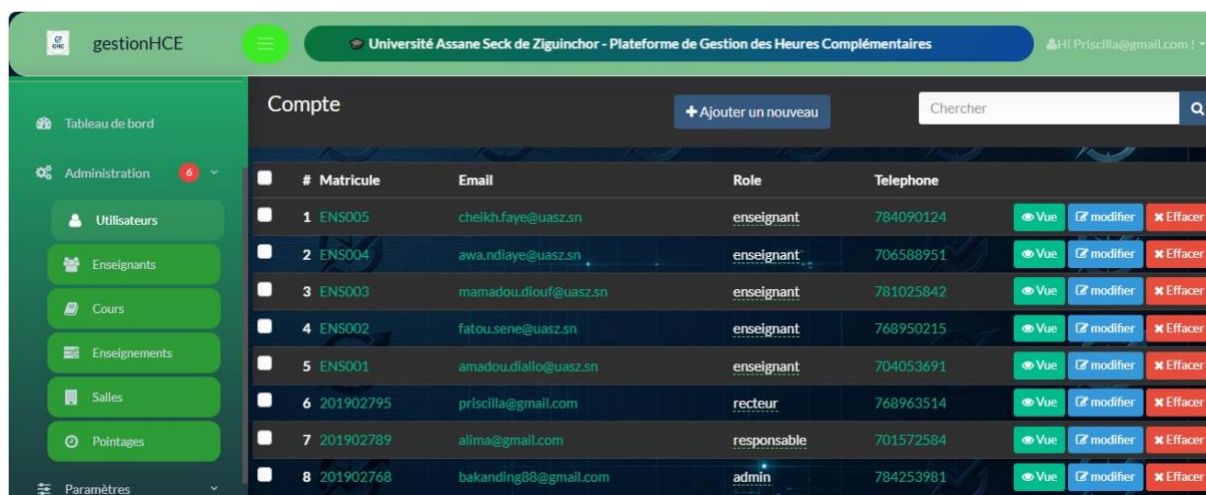


Figure 50: tableau de bord recteur

4.1.9. Gestion des utilisateurs

Pour gérer les utilisateurs, le recteur accède à la rubrique « Utilisateurs » dans le menu Administration. Cette page affiche la liste des comptes créés avec le matricule, l'email, le rôle et le téléphone de chaque utilisateur. Elle permet d'ajouter un nouvel utilisateur, de consulter ses informations, de les modifier ou de supprimer un compte.

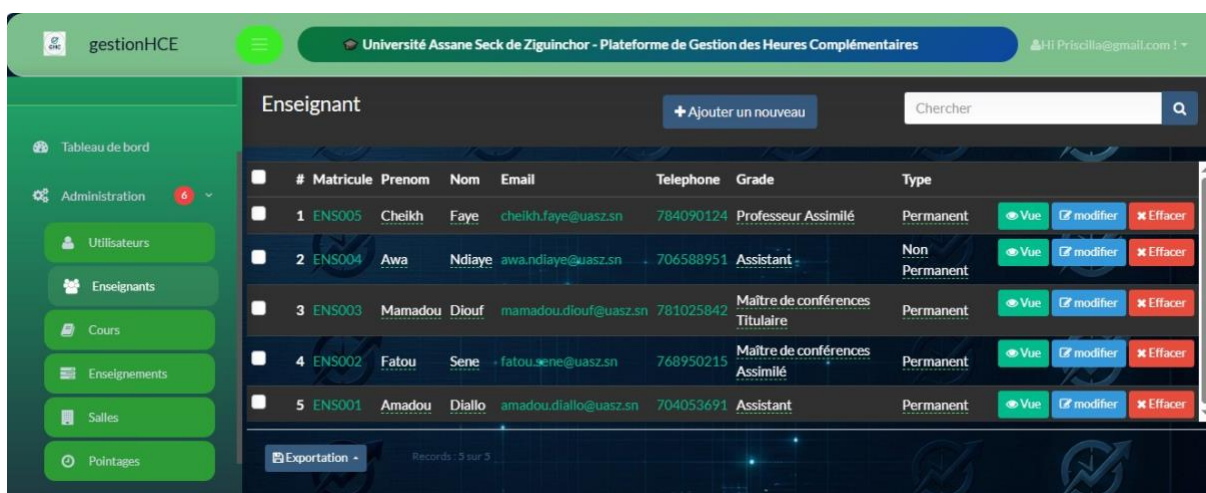


#	Matricule	Email	Role	Telephone	Actions
1	ENS005	cheikh.faye@uaszs.sn	enseignant	784090124	Vue modifier Effacer
2	ENS004	awa.ndiaye@uaszs.sn	enseignant	706588951	Vue modifier Effacer
3	ENS003	mamadou.diouf@uaszs.sn	enseignant	781025842	Vue modifier Effacer
4	ENS002	fatou.sene@uaszs.sn	enseignant	768950215	Vue modifier Effacer
5	ENS001	amadou.diallo@uaszs.sn	enseignant	704053691	Vue modifier Effacer
6	201902795	priscilla@gmail.com	recteur	768963514	Vue modifier Effacer
7	201902789	alima@gmail.com	responsable	701572584	Vue modifier Effacer
8	201902768	bakandling88@gmail.com	admin	784253981	Vue modifier Effacer

Figure 51: gestion des utilisateurs

4.1.10. Gestion des enseignants

Pour la rubrique « **Enseignants** », le recteur dispose d'une vue détaillée sur l'ensemble des enseignants enregistrés dans la plateforme. Cette interface présente les informations essentielles de l'enseignant. À partir de cette page, il peut ajouter un nouvel enseignant, consulter son profil, modifier ses informations ou supprimer un enregistrement si nécessaire.



#	Matricule	Prenom	Nom	Email	Telephone	Grade	Type	Actions
1	ENS005	Cheikh	Faye	cheikh.faye@uaszs.sn	784090124	Professeur Assimilé	Permanent	Vue modifier Effacer
2	ENS004	Awa	Ndiaye	awa.ndiaye@uaszs.sn	706588951	Assistant	Non Permanent	Vue modifier Effacer
3	ENS003	Mamadou	Diouf	amadou.diouf@uaszs.sn	781025842	Maitre de conférences Titulaire	Permanent	Vue modifier Effacer
4	ENS002	Fatou	Sene	fatou.sene@uaszs.sn	768950215	Maitre de conférences Assimilé	Permanent	Vue modifier Effacer
5	ENS001	Amadou	Diallo	amadou.diallo@uaszs.sn	704053691	Assistant	Permanent	Vue modifier Effacer

Figure 52: gestion des enseignants

4.1.11. Gestion des cours

Pour la rubrique « **Cours** », le recteur peut consulter l'ensemble des cours disponibles dans la plateforme. Cette page présente les informations principales. Elle permet également d'**ajouter** un nouveau cours, de **consulter** les détails, de **modifier** les informations existantes ou de **supprimer** un cours afin de maintenir une base de données pédagogique fiable et à jour.



#	Codecours	Codeec	Typecours	Vue	modifier	Effacer
1	MIO31212	MIO3121	TD			
2	MIO31211	MIO3121	CM			
3	MIO24221	MIO2422	CM			
4	MIO24121	MIO2412	CM			
5	MIO12311	MIO1231	TP			
6	MIO12211	MIO1221	TD			
7	MIO12112	MIO1211	TD			
8	MIO12111	MIO1211	TP			

Figure 53:gestion des cours

4.1.12. Gestion des enseignements

Pour la rubrique « **Enseignements** », le recteur peut suivre l'affectation des cours aux enseignants. Cette interface présente les répartitions déjà enregistrées avec le code de répartition, le matricule de l'enseignant, le code du cours, l'année académique et le semestre. Elle permet de vérifier si chaque cours est bien attribué à un enseignant et de contrôler l'organisation des enseignements avant le suivi des heures complémentaires.



#	Coderepartition	Matricule	Codecours	Année	Semestre	Vue	modifier	Effacer
1	6	ENS005	MIO31211	2024-2025	Semestre 6			
2	5	ENS005	MIO24221	2024-2025	Semestre 4			
3	4	ENS004	MIO24121	2024-2025	Semestre 4			
4	3	ENS003	MIO12211	2024-2025	Semestre 2			
5	2	ENS002	MIO12112	2024-2025	Semestre 2			
6	1	ENS001	MIO12111	2024-2025	Semestre 2			

Figure 54:gestion des enseignements

4.1.13. Gestion des salles

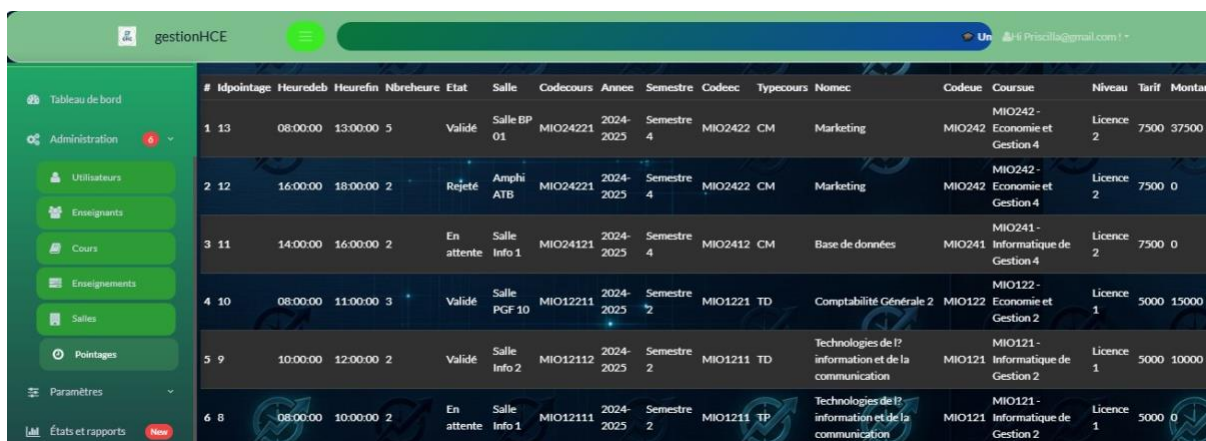
Pour le menu « **Salles** », le recteur peut consulter et gérer les salles utilisées pour les enseignements. Cette page présente les informations relatives aux salles disponibles et permet d'ajouter une nouvelle salle, de modifier ses informations ou de supprimer un enregistrement. Elle facilite ainsi l'organisation des cours et permet d'éviter les conflits liés à l'occupation des salles.



Figure 55: gestion des salles

4.1.14. Vue globale des enseignements

Pour la rubrique « **Pointages** », le recteur ou l'administrateur dispose d'un **tableau récapitulatif des heures déclarées par les enseignants**. Cette interface présente les informations détaillées sur chaque pointage. Elle permet ainsi de contrôler les heures complémentaires enregistrées et de suivre leur traitement selon leur statut : **validé, rejeté ou en attente**.



#	Idpointage	Heuredeb	Heurefin	Nbreheure	Etat	Salle	Codocours	Annee	Semestre	Codecc	Typecours	Nomcc	Codeue	Coursue	Niveau	Tarif	Montant
1	13	08:00:00	13:00:00	5	Validé	Salle BP 01	MIO24221	2024-2025	Semestre 4	MIO2422	CM	Marketing	MIO242-	MIO242- Economie et Gestion 4	Licence 2	7500	37500
2	12	16:00:00	18:00:00	2	Rejeté	Amphi ATB	MIO24221	2024-2025	Semestre 4	MIO2422	CM	Marketing	MIO242-	MIO242- Economie et Gestion 4	Licence 2	7500	0
3	11	14:00:00	16:00:00	2	En attente	Salle Info 1	MIO24121	2024-2025	Semestre 4	MIO2412	CM	Base de données	MIO241-	MIO241- Informatique de Gestion 4	Licence 2	7500	0
4	10	08:00:00	11:00:00	3	Validé	Salle PGF 10	MIO12211	2024-2025	Semestre 2	MIO1221	TD	Comptabilité Générale 2	MIO122-	MIO122- Economie et Gestion 2	Licence 1	5000	15000
5	9	10:00:00	12:00:00	2	Validé	Salle Info 2	MIO12112	2024-2025	Semestre 2	MIO1211	TD	Technologies de l'information et de la communication	MIO121-	MIO121- Informatique de Gestion 2	Licence 1	5000	10000
6	8	08:00:00	10:00:00	2	En attente	Salle Info 1	MIO12111	2024-2025	Semestre 2	MIO1211	TP	Technologies de l'information et de la communication	MIO121-	MIO121- Informatique de Gestion 2	Licence 1	5000	0

Figure 56: vue globale des enseignements

4.1.15. Paramètres

Pour la rubrique « **Paramètres** », le recteur peut gérer les données de base nécessaires au fonctionnement de la plateforme. Ce menu regroupe plusieurs éléments comme les **UFR**, les **départements**, les **filières**, les **niveaux**, les **semestres**, les **années académiques**, les éléments **constitutifs**, les **unités d'enseignement**, les **grades** et les **types d'éléments constitutifs**. Ces paramètres permettent de structurer correctement les informations pédagogiques utilisées dans la répartition des cours, le pointage et le suivi des heures complémentaires.

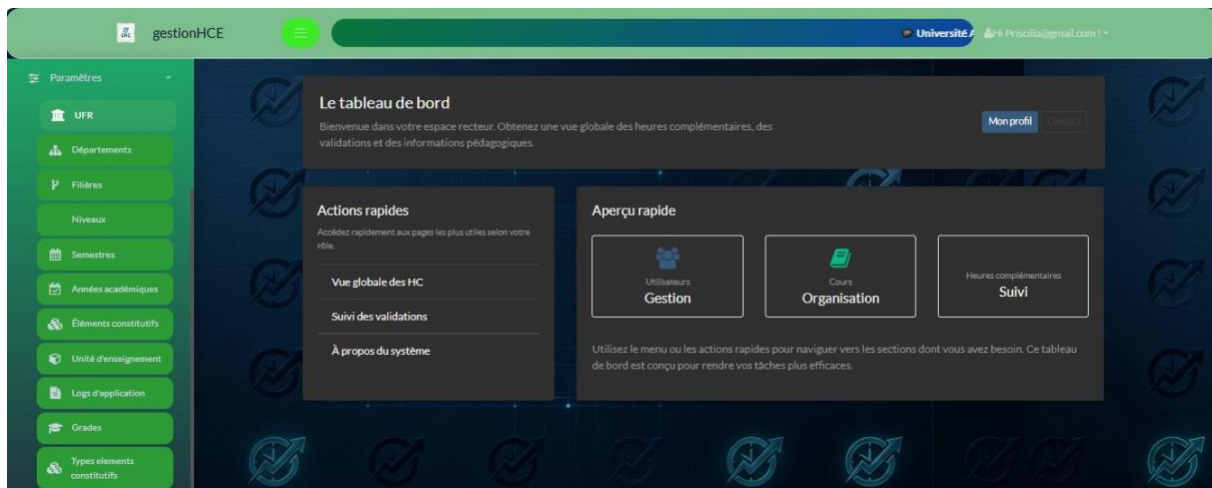
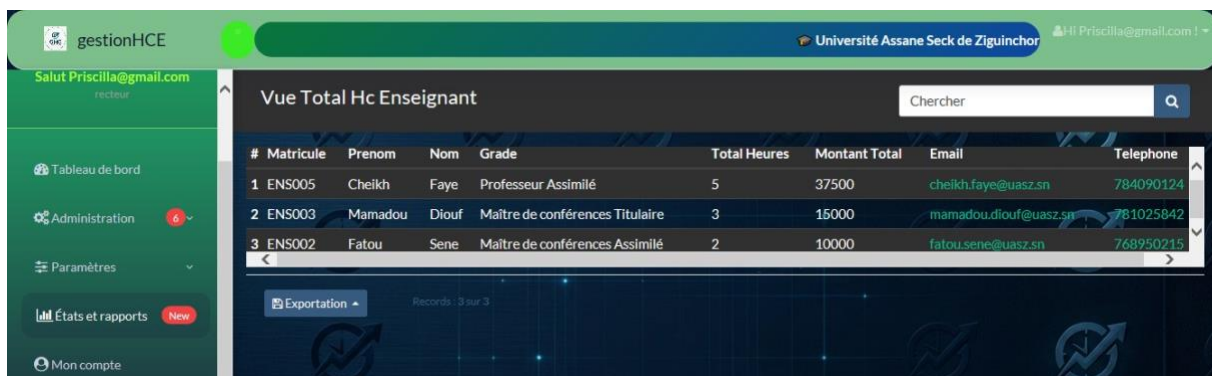


Figure 57: gestion pédagogique

4.1.16. Etats et rapport

Pour la rubrique « **États et rapports** », le recteur peut consulter les synthèses liées aux heures complémentaires. Cette page présente le total des heures effectuées par enseignant ainsi que le montant correspondant à payer. Elle affiche aussi des informations comme le matricule, le prénom, le nom, le grade, l'email et le téléphone. Grâce à cette rubrique, les responsables disposent d'un état récapitulatif clair pour contrôler les heures validées et préparer **les rapports ou les exports nécessaires**.



#	Matricule	Prenom	Nom	Grade	Total Heures	Montant Total	Email	Telephone
1	ENS005	Cheikh	Faye	Professeur Assimilé	5	37500	cheikh.faye@uaszs.sn	784090124
2	ENS003	Mamadou	Diouf	Maitre de conférences Titulaire	3	15000	mamadou.diouf@uaszs.sn	781025842
3	ENS002	Fatou	Sene	Maitre de conférences Assimilé	2	10000	fatou.sene@uaszs.sn	768950215

Figure 58: états et rapport

2.10. Espace administrateur

Le tableau de bord de l'administrateur constitue l'espace principal de pilotage de la plateforme. Après authentification, l'administrateur peut accéder rapidement aux fonctionnalités essentielles telles que la gestion des utilisateurs, des rôles et permissions, des cours, des salles et des pointages. Cet espace lui offre ainsi une vue d'ensemble sur le fonctionnement du système et lui permet d'assurer l'administration, l'organisation et le suivi des différentes données de la plateforme.

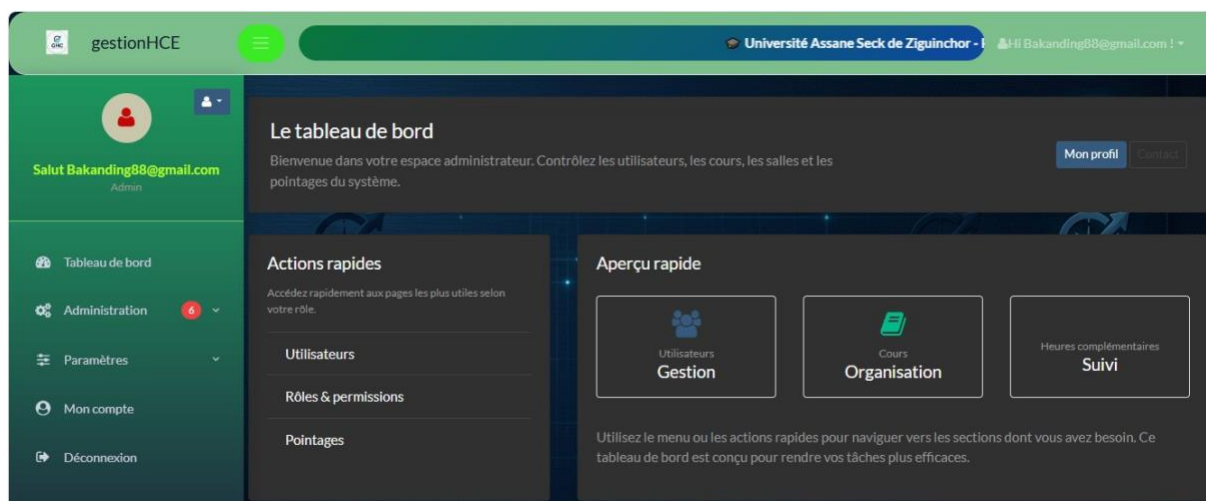


Figure 59: tableau de bord administrateur

4.1.17. App-logs

Pour la rubrique « **Logs d'application** », l'administrateur consulte l'historique des actions effectuées dans la plateforme. Cette page permet de suivre les connexions, les déconnexions et certaines opérations réalisées par les utilisateurs, avec des informations comme la date, l'action effectuée, la table concernée et l'identifiant de l'utilisateur. Elle facilite ainsi le contrôle, la traçabilité et l'audit du système.

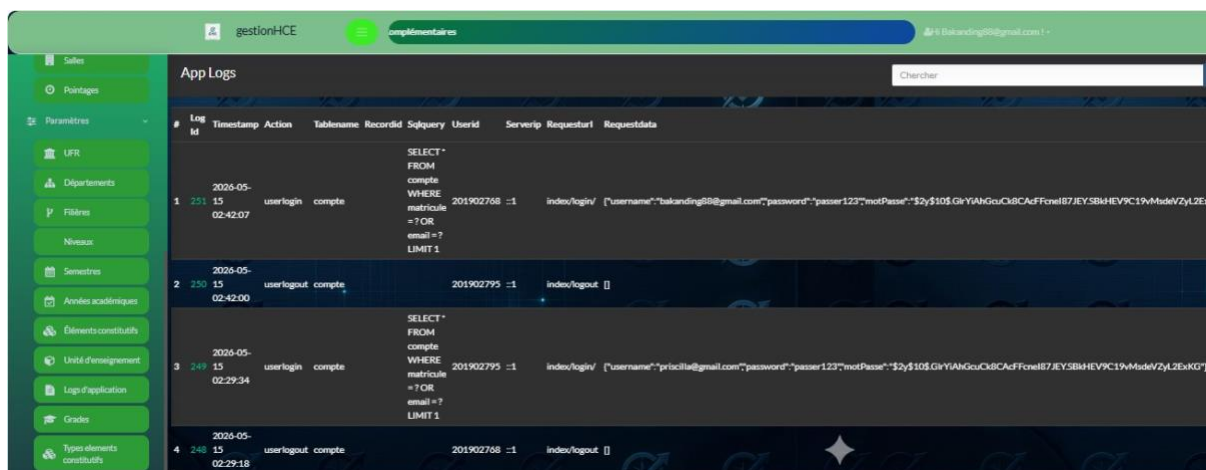


Figure 60: App-logs

NB : Les menus « **Administration** » et « **Paramètres** » sont accessibles à la fois au **recteur** et à l'**administrateur**. Ces deux acteurs partagent ces rubriques afin de gérer les informations essentielles de la plateforme, notamment les utilisateurs, les enseignants, les cours, les salles, les pointages ainsi que les données de configuration comme les UFR, départements, filières, niveaux, semestres et années académiques. Toutefois, les actions disponibles peuvent varier selon les droits attribués à chaque profil.

En résumé, ce chapitre a présenté la solution développée pour la gestion des heures complémentaires, en décrivant les principales interfaces et les fonctionnalités associées à chaque profil utilisateur. De l'accès à la plateforme jusqu'aux espaces dédiés à l'enseignant, au responsable, au recteur et à l'administrateur, chaque module a été conçu pour répondre à des besoins précis. Cette organisation permet de faciliter le pointage des cours, le suivi des validations, la gestion pédagogique et l'administration globale du système.

Conclusion et perspective

Au terme de ce mémoire, nous pouvons retenir que la gestion des heures complémentaires constitue un enjeu important dans le fonctionnement administratif et pédagogique des établissements d'enseignement supérieur. À l'Université Assane Seck de Ziguinchor, cette gestion implique plusieurs acteurs et nécessite un suivi rigoureux depuis la déclaration des heures jusqu'à leur validation et leur transmission pour traitement. Cependant, les méthodes semi-automatisées peuvent entraîner des difficultés telles que les erreurs de comptage, la redondance des informations, le manque de traçabilité et le retard dans le traitement des dossiers. C'est dans ce contexte que nous avons mené cette étude portant sur la conception et la réalisation d'un système d'information de gestion des heures complémentaires des enseignements. L'objectif principal était de proposer une solution capable de faciliter la déclaration, le pointage, le suivi, la validation et la consultation des heures complémentaires. Pour atteindre cet objectif, nous avons d'abord analysé le fonctionnement existant, identifié les acteurs concernés ainsi que leurs besoins fonctionnels, puis modélisé le système à l'aide des outils d'analyse et de conception adaptés.

La solution développée permet aux différents utilisateurs d'accéder à des espaces adaptés à leurs rôles. **L'enseignant** peut pointer ses cours, consulter ses pointages et suivre l'état de ses heures. **Le responsable** peut gérer les répartitions, les cours, les salles et contrôler les pointages. **Le recteur et l'administrateur** disposent d'une vue plus globale leur permettant de suivre les validations, gérer les utilisateurs, consulter les états et rapports, et administrer les paramètres du système. Cette organisation contribue à améliorer la transparence, la fiabilité des données et la rapidité du traitement des heures complémentaires. La mise en œuvre de cette plateforme montre ainsi l'importance d'un système d'information dans la modernisation des procédures administratives universitaires. En centralisant les données et en définissant clairement les rôles de chaque acteur, la solution proposée permet de réduire les erreurs, de renforcer le contrôle et d'assurer une meilleure traçabilité des opérations effectuées. Elle constitue donc un outil d'aide à la gestion et à la prise de décision pour l'université.

Toutefois, comme tout projet informatique, cette solution peut encore être améliorée. Parmi les perspectives, nous pouvons envisager l'intégration d'un système de notification automatique, la génération avancée de rapports statistiques, l'export sécurisé des données vers le service financier, ainsi que le renforcement de la sécurité et de l'audit des actions réalisées dans la plateforme. Ces améliorations permettraient de rendre le système plus complet, plus performant et mieux adapté aux besoins futurs de l'institution.

Références

1. Kalika, M., Bouzidi, L., & Drissi, S. (2010). *Le mythe de la génération Y au travail*. *Revue française de gestion*, (2), 67-83.
2. Alter, S. (2008). *Defining information systems as work systems: Implications for the IS field*. *European Journal of Information Systems*, 17(5), 448-469.
3. N'Guessan, G. (2017). *Numérisation de l'administration universitaire en Côte d'Ivoire : enjeux et défis*. *Revue Française d'Administration Publique*, (3), 163-175.
4. Reix, R. (2004). *Systèmes d'information et management des organisations (5e éd.)*. Vuibert.
5. Lauriol, J., Perret, V., & Tannery, F. (2008). *Stratégie et systèmes d'information*. Dunod.
6. Besson, P., & Rowe, F. (2012). *Stratégie et systèmes d'information : Un regard sur 30 ans de recherche et des perspectives post 2010*. *Systèmes d'Information et Management*, 17(2), 1-11.
7. Ndiaye, I. (2019). *Gestion des ressources humaines et performance des établissements d'enseignement supérieur au Sénégal : défis et perspectives*. *Revue Africaine de Management*, 5(2), 45-62.
8. Turban, E., Sharda, R., & Delen, D. (2011). *Decision Support and Business Intelligence Systems (9th ed.)*. Pearson Education. (Ouvrage de référence sur les SID).
9. Laudon, K. C., & Laudon, J. P. (2020). *Management information systems: Managing the digital firm (16th ed.)*. Pearson. (Ouvrage de référence détaillant le rôle et l'architecture des SIG).
10. Sourang, M. (2021). *Transformation numérique et administration publique : l'exemple de la gestion académique*. Dans A. Diop (Éd.), *Innovation et réforme dans les services publics au Sénégal* (pp. 112-135). Presses Universitaires de Dakar.
11. Kakunga Lushimba, C., & Banantamba Kalunde, T. (2025). *Comparative study between MERISE and UML as tools for analyzing and designing information systems*. *International Journal of Academic and Applied Research*, 9(10), 27-39.
12. Avison, D. E. (1991). *MERISE: A European methodology for developing information systems*. *European Journal of Information Systems*, 1(3), 183-191.
13. Springer Nature. (2021). *ArgoUML-SPL and related studies in software engineering tools (Open-source software product line extraction)*.
14. Agner, L. T. W., & Lethbridge, T. C. (2017). *Modeling tools usage in education and industry (ref. ACM/IEEE MODELS conference)*.
15. Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems seventh edition*.
16. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *Database system concepts (Vol. 5)*. New York: McGraw-Hill.
17. Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Wiley
18. Flanagan, D. (2020). *JavaScript: The Definitive Guide (7th ed.)*. O'Reilly Media.
19. Van Rossum, G., & Drake, F. (2009). *Python 3 Reference Manual*. CreateSpace.
20. Lutz, M. (2013). *Learning Python (5th ed.)*. O'Reilly Media.
21. Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development (5th ed.)*. Addison-Wesley.
22. Nixon, R. (2018). *Learning PHP, MySQL & JavaScript*. O'Reilly Media.
23. Bloch, J. (2018). *Effective Java (3rd ed.)*. Addison-Wesley.
24. Gosling, J., Joy, B., Steele, G., & Bracha, G. (2005). *The Java Language Specification*. Addison-Wesley.
25. Stroustrup, B. (2013). *The C++ programming language*. Pearson Education.
26. Lippman, S., Lajoie, J., & Moo, B. (2012). *C++ Primer (5th ed.)*. Addison-Wesley.
27. Troelsen, A., & Japikse, P. (2021). *Pro C# 9 with .NET 5*. Apress.
28. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
29. Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.

30. Spurlock, J. (2013). *Bootstrap: Responsive Web Development*. O'Reilly Media.
31. Lerman, J. (2010). *Programming Entity Framework*. O'Reilly Media.
32. Potencier, F. (2018). *Symfony 4: The Fast Track*. Symfony SAS.
33. Stauffer, M. (2019). *Laravel: Up and Running*. O'Reilly Media.
34. Holovaty, A., & Kaplan-Moss, J. (2009). *The Definitive Guide to Django*. Apress.
35. Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80–83.
36. Banks, A., & Porcello, E. (2020). *Learning React (2nd ed.)*. O'Reilly Media.
37. Chacon, S., & Straub, B. (2014). *Git Internals—Git References*. *Pro Git (2nd ed.)*. Apress, Berkeley, CA, 365-368.
38. Grigorik, I. (2013). *High Performance Browser Networking*. O'Reilly Media.
39. Siau, K., & Lee, L. (2004). Are use case and class diagrams complementary in requirements analysis? *An experimental study on use case and class diagrams in UML*. *Requirements Engineering*, 9(4), 229–237.

Webographie

- [W1] <https://commentouvrir.com/tech/comprendre-les-systemes-dinformation-definition-roles-objectifs-et-types/> consulté le 18/11/2025
- [W2] <https://www.academie-rh.com/post/tout-savoir-sur-les-heures-complementaires-et-leur-majoration> consulter le 27/03/2026
- [W3] République du Sénégal. (2021). Code général de la Fonction publique (Loi n° 2021-34). Journal Officiel. consulter le 27/03/2026
- [W4] <https://www.cybermedian.com/fr/comprehensive-guide-to-the-14-uml-diagram-types/> consulté le 27/03/2026
- [W5] [https://fr.wikipedia.org/wiki/Merise_\(informatique\)#M%C3%A9thode_d'analyse_et_de_conception](https://fr.wikipedia.org/wiki/Merise_(informatique)#M%C3%A9thode_d'analyse_et_de_conception) consulter le 27/03/2026
- [W6] <https://en.wikipedia.org/wiki/ArgoUML> consulter le 27/03/2026
- [W7] <https://en.wikipedia.org/wiki/StarUML> consulté le 27/03/2026
- [W8] [Base de données Oracle - Wikipédia](#) consulté le 28/03/2026
- [W9] https://en.wikipedia.org/wiki/My_SQL consulté le 28/03/2026
- [W10] <https://en.wikipedia.org/wiki/HTML> consulté le 30/03/2026
- [W11] <https://en.wikipedia.org/wiki/CSS> consulté le 30/03/2026
- [W12] <https://en.wikipedia.org/wiki/JavaScript> consulté le 30/03/2026
- [W13] [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) consulté le 30/03/2026
- [W14] https://en.wikipedia.org/wiki/ASP.NET_MVC consulter le 27/03/2026
- [W15] [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)) consulter le 27/03/2026
- [W16] https://en.wikipedia.org/wiki/Entity_Framework consulter le 27/03/2026
- [W17] <https://symfony.com/doc> conslté le 04/04/2026
- [W18] <https://phprad.com/> il est instable le 04/04/2026
- [W19] [Visual Studio - Wikipédia](#) consulter le 27/03/2026
- [W20] [Git - Wikipédia](#) consulté le 30/03/2026

Table des matières

Dédicace.....	i
Remerciement	ii
Résumé.....	iii
Abstract	iv
Sommaire	v
Liste des figures	vi
Liste des tableaux.....	viii
Sigle.....	ix
Introduction générale	1
Chapitre 1 : Généralité sur les systèmes d'informations et les heures complémentaires.....	5
1.1. Introduction	6
1.2. Les systèmes d'information	6
1.2.1. Définitions	6
1.2.2. Missions d'un SI.....	7
1.2.3. Architecture de base d'un SI	7
1.2.4. Les types de SI	8
3.1.1.1 Les SIO.....	8
3.1.1.2 Les SID.....	9
3.1.1.3 Les SIG.....	10
1.3. La gestion des heures complémentaires	11
1.1.1. Heures normales	11
1.1.2. Heures complémentaires	12
1.1.3. Charge statutaires	13
1.1.4. Règles de conversion.....	14
1.1.5. Tarification des heures commentaire	15
1.1.6. Procédure de déclaration des heures complémentaires	16
1.4. Conclusion.....	18
Chapitre 2 : Revue de la littérature et Cahier des charges et	19
2.1 Introduction	20
2.2. Revue de la littérature	20

2.3.1.	KairosSuite	20
2.3.2.	ERP5.....	21
2.3.3.	OSE (Organisation des Services d'Enseignement).	22
2.3.4.	SAGHE (Système Automatisé de Gestion des Heures d'enseignement).....	23
2.3.5.	PECHE (Programme d'Élaboration des Charges d'Enseignement)	23
2.3.6.	Apogée	24
2.3.7.	Ypareo	25
2.3.8.	Jibble	25
2.3.9.	TimeCamp	26
2.3.10.	Clockify	27
2.3.11.	JOUROFF.....	27
2.3.12.	MyScol	28
2.3.13.	Positionnement	30
2.3.	Cahier des charges.....	30
2.3.5.	Contexte	30
2.3.6.	Mission	31
2.2.3.	Le périmètre du projet	31
2.2.4.	Etude de l'existant	31
2.3.7.	Besoins Fonctionnelle	32
2.2.6	Spécifications Technique / exigences technique	33
2.4.	Conclusion.....	33
Chapitre 3 : Analyse, conception et réalisation du SI		34
3.1.	Introduction	35
3.2.	Choix des outils	35
3.2.1.	Outils d'analyse.....	35
3.2.1.1.	UML	35
3.2.1.2.	MERISE	36
3.2.2.	Outils de modélisation.....	38

3.2.2.1.	ARGOUML.....	38
3.2.2.2.	Start UML	38
3.2.3.	Bases de données.....	39
3.2.3.1.	Oracle	39
3.2.3.2.	MySQL.....	40
3.2.4.	Langage de programmation.....	40
3.2.4.1.	HTML.....	41
3.2.4.2.	CSS.....	41
3.2.4.3.	JavaScript	41
3.2.4.4.	Python.....	42
3.2.4.5.	PHP.....	42
3.2.4.6.	Java.....	42
3.2.4.7.	C++.....	43
3.2.4.8.	C#	43
3.2.5.	Framework de développement	43
3.2.5.1.	ASP.NET MVS	44
3.2.5.2.	Bootstrap	44
3.2.5.3.	Entity Framework.....	44
3.2.5.4.	Symfony	44
3.2.5.5.	PHPRad	45
3.2.5.6.	Laravel.....	45
3.2.5.7.	Django	45
3.2.5.8.	Node.js.....	45
3.2.5.9.	React.js	46
3.2.6.	Autres outils et technologies	46
3.2.6.1.	Visual Studio	46
3.2.6.2.	Git/GitHub.....	46
3.2.6.3.	Google chrome	47
3.3.	Analyse du SI	47
3.3.1.	User CASE	47
3.3.2.	Diagramme de classe.....	51

3.3.3.	Diagramme de séquence.....	53
3.3.4.	Modèle relationnel.....	57
3.3.5.	Modèle physique de données	58
3.3.6.	Création de la base de données	59
3.4.	Réalisation	60
3.4.1.	Installation du framework	60
3.4.2.	Création du projet sur PHPRAD	61
3.4.3.	Le choix du thème	62
3.4.4.	Organisation du menu	62
3.4.5.	Gestion des pages	63
3.4.6.	Gestion des authentifications	64
3.4.7.	Gestion des rôles	65
3.4.8.	Gestion de l'audites	66
3.5.	Conclusion.....	66
Chapitre 4 : présentation de la solution.....		67
2.5.	Accès à la plateforme	68
2.6.	Création de compte.....	68
2.7.	Espace enseignant	69
4.1.1.	Pointer un cours.....	69
4.1.2.	Voir les pointages.....	70
2.8.	Espace responsable.....	70
4.1.3.	Faire la répartition	71
4.1.4.	Gestion des cours	71
4.1.5.	Gestion des salles	72
4.1.6.	Gestion des pointages.....	72
4.1.7.	Vue globale des pointages.....	73
4.1.8.	Vue sur les enseignants	73
2.9.	Espace Recteur	74



4.1.9.	Gestion des utilisateurs.....	74
4.1.10.	Gestion des enseignants	75
4.1.11.	Gestion des cours	75
4.1.12.	Gestion des enseignements.....	76
4.1.13.	Gestion des salles	76
4.1.14.	Vue globale des enseignements.....	77
4.1.15.	Paramètres	77
4.1.16.	Etats et rapport	78
2.10.	Espace administrateur	78
4.1.17.	App-logs.....	79
	Conclusion et perspective	81
	Références.....	82
	Webographie	84
	Table des matières.....	85
	Annexe 1	90
	Annexe 2	94

Annexe 1

1. Entretien avec la secrétaire du département Economie et Gestion

Questionnaire – Transmission des heures complémentaires destiné aux secrétaires de département (UFR SES)

Etudiant : Bekandira Gendiahy.....
Encadreur : **Professeur Édouard Ngor SARR**
Formation : Master MSIA

Cible : Mme SANE Rahéya BODIAN.....

1. Informations générales
Département :
☒ Économie
☒ Gestion
☐ Sociologie
Votre fonction :
☒ Secrétaire du département

2. Réception des heures complémentaires
Comment recevez-vous habituellement les heures complémentaires ?
☒ Papier signé
☐ Fichier Excel
☐ E-mail
☐ Autre :

Le format utilisé par les enseignants est-il uniforme ?
☒ Oui
☐ Non
☐ Parfois

Recevez-vous les heures dans les délais fixés ?
☐ Oui
☒ Parfois
☐ Non

3. Traitement et transmission
Après réception, que faites-vous avant transmission ? (plusieurs réponses possibles)
☒ Vérification du remplissage
☒ Vérification des signatures
☒ Vérification de la cohérence des heures
☐ Rien (simple transmission)
☐ Autre :

Comment transmettez-vous les heures à la scolarité/UFR ?
☒ Courrier interne (papier)
☐ Email
☐ Dépôt physique
☐ Plateforme interne
☐ Autre :

Rencontrez-vous des difficultés lors de la transmission ?

- ☐ Oui
☒ Parfois
☐ Non

Si oui, lesquelles ?

4. Suivi

Avez-vous un moyen de vérifier que les heures ont bien été reçues et traitées ?

- ☐ Oui
☒ Non
☐ Parfois

Avez-vous déjà constaté des retards ou pertes de documents ?

- ☒ Oui
☐ Parfois
☐ Non

5. Amélioration

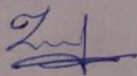
Que faudrait-il améliorer dans la procédure de transmission ?

- ☐ Clarification des étapes
☐ Uniformisation des fiches d'heures complémentaires
☒ Un canal unique (ex : email officiel)
☐ Un formulaire numérique standardisé
☐ Autre :

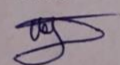
Suggestions supplémentaires :

Je suggère que les fiches de déclaration
soient prises en charge et réglées à l'uf
surtout pour les vacataires ou que le
retard de paiement est éliminé.

Etudiant



Cible



Encadreur



Scanné avec CamScanner

2. Entretien avec la secrétaire du département de Sociologie

Questionnaire – Transmission des heures complémentaires destiné aux secrétaires de département (UFR SES)

Etudiant : Bekanding Gendiaby

Encadreur : **Professeur Édouard Ngor SARR**

Formation : Master MSIA

Cible : Mme Mawéou Ndoye

1. Informations générales

Département :

☐ Économie

☐ Gestion

☒ Sociologie

Votre fonction :

☒ Secrétaire du département

2. Réception des heures complémentaires

Comment recevez-vous habituellement les heures complémentaires ?

☒ Papier signé

☐ Fichier Excel

☐ E-mail

☐ Autre :

Le format utilisé par les enseignants est-il uniforme ?

☒ Oui

☐ Non

☐ Parfois

Recevez-vous les heures dans les délais fixés ?

☐ Oui

☒ Parfois

☐ Non

3. Traitement et transmission

Après réception, que faites-vous avant transmission ? (plusieurs réponses possibles)

☒ Vérification du remplissage

☒ Vérification des signatures

☒ Vérification de la cohérence des heures

☐ Rien (simple transmission)

☐ Autre :

Comment transmettez-vous les heures à la scolarité/UFR ?

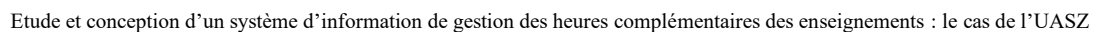
☒ Courrier interne (papier)

☐ Email

☐ Dépôt physique

☐ Plateforme interne

☐ Autre :



Si oui, lesquelles ?

☐ Parfois☐ Non☐ Autre :

This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

2nd

ms. 100

Encadreur

Annexe 2

1. Entretien avec le directeur de la DESRU

Ziguinchor, le 10 avril 2026

Objet : Compte rendu d'entretien sur le processus de transmission des heures complémentaires.

La transmission des heures complémentaires s'inscrit dans un processus structuré et hiérarchisé impliquant plusieurs niveaux de validation afin de garantir la fiabilité et la conformité des informations. Dans un premier temps, l'enseignant effectue la déclaration de ses heures complémentaires en renseignant les informations nécessaires, puis la dépose au **secrétariat de son département**. Le chef de département procède à une première vérification portant sur la conformité des données déclarées avant de transmettre le dossier au **secrétariat de l'UFR**, où une seconde vérification est censée être effectuée.

Par la suite, la demande est soumise au **Vice-Recteur chargé des Études et de la Vie universitaire**, qui la transmet à la **Direction des Études, de la Statistique et de la Réforme universitaire** pour une étude et une vérification approfondies. Une fois cette étape validée, le dossier est retourné au **Vice-Recteur**, qui se charge de le transmettre au **Secrétariat général**.

Le **Secrétaire général** assure ensuite la transmission du dossier à la **Direction des Finances et de la Comptabilité (DFC)**, où une vérification de conformité financière est effectuée. Cette étape permet de s'assurer de l'exactitude des montants calculés ainsi que du respect des règles en vigueur. Après validation par la DFC, le dossier est transmis à l'**Agent comptable**, qui procède, à son tour, à une dernière vérification avant d'engager le paiement des heures complémentaires.

Toutefois, en cas d'anomalie détectée au niveau de la Direction des Finances et de la Comptabilité (DFC) ou de l'agence comptable, le dossier est directement retourné au Vice-Recteur chargé des Études et de la Vie universitaire pour correction ou réexamen avant toute validation définitive. De même, si une anomalie est constatée au niveau de la Direction des Études, de la Statistique et de la Réforme universitaire, le dossier est renvoyé au Vice-Recteur, qui se charge à son tour de le retransmettre au niveau de l'UFR concernée. Le dossier suit alors le circuit inverse en passant par le secrétariat de l'UFR puis celui du département, jusqu'à l'enseignant, afin de permettre la correction des informations erronées.

Dans tout processus, des anomalies ou des erreurs peuvent être observées. Dans notre cas, nous soulignons notamment la redondance d'heures de cours dans l'emploi du temps, les erreurs de comptage des heures effectuées ainsi que le non-respect du canevas établi.



Bakanding Goudiaby

