



Université Mohamed khider de Biskra
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département d'informatique

Mémoire de master

Systeme d'Information Optimisation et Décision (SIOD)

Présenté et soutenu par :

Mesmoudi Wissem

Gestion numérique des formulaires administratifs : application à des tâches pédagogiques universitaires

Jury :

Dr.Moufida Benchaabane

Université mohammed khider biskra

Rapporteur

Dr.Babahenini Djihane

Université mohammed khider biskra

Président

Dr.Houhou Okba

Université mohammed khider biskra

Examineur

Année universitaire : 2023/2024

Remerciement

*Tout d'abord, je tiens à remercier **Dieu**, le tout-puissant pour les bienfaits qu'il m'a accordés tout au long de mon travail de recherche et qui m'ont permis de l'achever avec succès.*

*Je suis reconnaissant envers **moi-même** pour ma persévérance, ma détermination et ma capacité à surmonter les défis rencontrés.*

Je tiens tout à exprimer ma profonde gratitude à ma directrice de mémoire

*Mme **Benchabane Moufida**,*

pour son encadrement, sa patience et sa confiance tout au long

de ce travail de recherche. Ses précieux conseils,

son expertise et son soutien inébranlable ont été d'une aide inestimable et ont

grandement contribué à l'aboutissement de ce projet.

*Je tiens à remercier très vivement Monsieur **Ababsa Hachemi***

docteur en science de gestion, responsable de la cellule de tutorat de la faculté

des sciences économiques pour son accueil , son aide

et ses remarques et discussions utiles.

À l'ensemble des professeurs du département d'Informatique

*de l' **Université Mohamed Khider**, et surtout à Monsieur*

Abdelhak Merizig

*pour leur enseignement de qualité et les connaissances qu'ils m'ont transmises
durant mes années d'études. Leur passion et leur dévouement
pour la recherche m'ont inspiré et motivé à poursuivre
mes propres questionnements scientifiques.*

*À tous les membres du jury de ma soutenance de mémoire pour avoir accepté de
faire partie du jury. Leurs remarques et suggestions ont été précieuses
et m'ont permis d'en améliorer la qualité.*

*À ma famille,
pour leur soutien indéfectible et pour avoir toujours cru en moi. Leurs
encouragements ont été mon refuge et ma motivation
durant tout le parcours académique.*

*Enfin, à toutes les personnes qui, de près ou de loin, ont contribué à la
réalisation de ce mémoire. Que ce soit à travers des discussions
enrichissantes, des conseils ou simplement par leur présence,
leur contribution a été précieuse.*

*Ce mémoire est le fruit d'un travail collectif autant que personnel,
et je suis profondément reconnaissant envers tous ceux qui m'ont accompagné.*

Résumé

L'évolution de l'enseignement supérieur a engendré diverses initiatives visant à améliorer ses résultats, parmi lesquelles figure l'instauration du tutorat pour les nouveaux étudiants. Ce processus vise à les orienter dans leurs études et à les soutenir face aux défis propres à l'enseignement supérieur.

Conscient des avantages que cette tâche offre, il est décisif d'en favoriser le développement afin de favoriser la réussite académique. Notre mémoire se concentre sur cette problématique en proposant une solution novatrice : une application web dédiée à la numérisation des formulaires administratifs utilisés dans le cadre du tutorat, ainsi qu'à la gestion efficace des séances.

En facilitant les tâches des acteurs impliqués dans le tutorat, notre travail aspire à optimiser le processus d'accompagnement pédagogique et à renforcer l'expérience éducative au sein de l'enseignement supérieur.

Abstract

The evolution of higher education has given rise to a number of initiatives aimed at improving results, including the introduction of tutoring for new students. This process aims to guide them through their studies and support them as they face the challenges of higher education.

Aware of the advantages that this practice offers, it is crucial to encourage its development in order to promote academic success. Our dissertation focuses on this issue by proposing an innovative solution : a web application dedicated to digitizing the administrative forms used in tutoring, as well as to efficiently managing the sessions.

By facilitating the tasks of those involved in tutoring, our work aims to optimize the tutoring process and enhance the educational experience within higher education.

ملخص

لقد أدى تطور التعليم العالي إلى ظهور العديد من المبادرات التي تهدف إلى تحسين أدائه، نذكر من بينها التدريس الخصوصي للطلاب الجدد أو ما يسمى بالمرافقة البيداغوجية. الهدف من هذه العملية هو إرشاد الطلبة في دراستهم ودعمهم في مواجهة تحديات التعليم العالي. إدراكاً للمزايا التي تقدمها هذه الممارسة، من الضروري تشجيع تطويرها من أجل تعزيز النجاح الأكاديمي. تركز أطروحتنا على هذه المسألة من خلال اقتراح حل مبتكر : تطبيق ويب مخصص لرقمنة النماذج الإدارية المستخدمة في المرافقة البيداغوجية، وكذلك لإدارة الجلسات بكفاءة. يهدف عملنا إلى تسهيل مهام المشاركين في المرافقة البيداغوجية وتحسين عملية التدريس وتعزيز التجربة التعليمية في التعليم العالي.

Table des matières

Introduction générale	1
Chapitre I : Contexte de travail	
1. Introduction	3
2. Définition du tutorat	3
3. Description des différentes tâches	3
3.1 Rôle et responsabilité du recteur de tutorat	3
3.2 Rôle et responsabilité du tuteur	4
3.3 Rôle et responsabilité du tuteur	4
4. Aspects et niveaux de supervision	5
4.1 Aspect média et administratif	5
4.2 Aspect Méthodologique	6
4.3 Aspect pédagogique	6
4.4 Aspect professionnel	7
4.5 Aspect technique	7
4.6 Aspect psychologique	8
5. Etude de l'existant	9
5.1 Bureau de tutorat	9
5.2 Formulaire administratifs du tutorat	10
5.2.1 Fiche de tutorat	10

5.2.2	Attestation du tuteur pédagogique	11
5.2.3	Registre de présence des tuteurs	12
5.2.4	Registre de présence des étudiants	13
6.	Système de gestion des formulaires administratifs	13
6.1	Problématique	13
6.2	Objectif	14
7.	Conclusion	15

Chapitre II :Les technologies web

1.	Introduction	17
2.	Historique du Web	17
3.	Les applications web	18
3.1	Définition	18
3.2	Types des applications	18
3.2.1	Applications web statiques	19
3.2.2	Applications web e-commerce	19
3.2.3	Applications web Portails	20
3.2.4	Applications web progressives	20
3.2.5	Applications web de gestion de contenu	20
3.2.6	Applications web dynamiques	21
4.	Architecture des applications web	22
4.1	Architecture Client-Serveur	22
5.	fonctionnement d'une application web	26
6.	Technologies des applications web	27
6.1	Langages de programmation	27
6.2	Technologies côté client	27
6.3	Technologies côté serveur	29
7.	avantages des applications web	29

8. Travaux connexes	31
---------------------	----

Chapitre III : Conception et architecture de système

1. Introduction	35
2. Méthode en cascade	35
3. Architecture proposée	36
3.1 Description	36
3.2 Avantages	37
4. langage de modélisation UML	38
5. Diagramme UML	38
6. Diagramme de cas d'utilisation	41
6.1 Administrateur	41
6.2 Enseignant	43
6.3 Etudiant	45
7. Diagramme de séquence	46
7.1 authentification	46
7.2 Demande séance	47
7.3 Traiter une demande de séance	49
7.4 Ajouter une attestation	51
7.5 Ajouter une entité	52
7.6 Ajouter une fiche	53
8. Diagramme de classe	54
9. Conception de la base de données	55
9.1 Le modèle relationnel	55
10. Conclusion	57

Chapitre IV : Implémentation

1. Introduction	59
-----------------	----

2.	Outils de développement	59
2.1	Modélisation	59
2.1.1	Modelio Open Source 4.1	59
2.2	Front-End	60
2.3	Back-End	62
2.4	Environnement de Développement Intégré (IDE)	64
2.5	Visual Studio Code	64
2.6	Navigateur Internet	64
3.	Description de l'application	65
3.1	Interface d'authentification	65
3.2	Administrateur	67
3.2.1	Page d'accueil	67
3.2.2	Gestion des demandes	68
3.2.3	Opérations d'ajout	68
3.3	Enseignant	71
3.3.1	Page d'accueil	71
3.3.2	Gestion des séances	72
3.3.3	Demande séance	72
3.3.4	Gestion des fiches	74
3.4	Etudiant	75
3.4.1	Page d'accueil	75
3.4.2	Consultation	76
4.	Conclusion	78
	Références	82
	Liste des termes	90

Table des figures

I.1	aspects de l'accompagnement pédagogique.	8
I.2	Plan de l'unité de tutorat. [2]	10
I.3	Fiche d'accompagnement de l'étudiant de première année licence. [2]	11
I.4	Certificat d'accompagnement ou attestation de tuteur pédagogique. [2]	12
I.5	Registre de présence des tuteurs. [2]	13
I.6	Registre de présence des tutorés.[2]	13
II.1	Les types des applications web. [7]	19
II.2	Le principe du fonctionnement de l'architecture Client/Serveur. [16]	23
II.3	Principe du fonctionnement de l'architecture en trois couches. [16]	25
II.4	Principe du fonctionnement de l'architecture multicouche. [18]	25
II.5	Principe du fonctionnement des applications web. [19]	26
III.1	Le modèle en cascade présenté schématiquement.[26]	36
III.2	Principe du fonctionnement de l'architecture en deux couches. [16]	37
III.3	Schéma des diagrammes UML. [31]	39
III.4	Diagramme de cas d'utilisation de l'administrateur.	41
III.5	Diagramme de cas d'utilisation de l'enseignant.	43
III.6	Diagramme de cas d'utilisation de l'étudiant.	45
III.7	Diagramme de séquence d'authentification.	47

III.8 Diagramme de séquence :demande d'une séance par un enseignant.	48
III.9 Diagramme de séquence : demande d'une séance par un étudiant.	49
III.10 Diagramme de Séquence :traitement d'une demande par l'administrateur.	50
III.11 Diagramme de Séquence :traitement d'une demande par l'enseignant.	51
III.12 Diagramme de séquence :ajouter une attestation.	52
III.13 Diagramme de séquence : ajouter une nouvelle entité.	53
III.14 Diagramme de séquence : ajouter une fiche.	54
III.15 Diagramme de classe	55
IV.1 Interface d'authentification.	66
IV.2 Interface de réinitialisation du mot de passe.	67
IV.3 Page d'accueil du compte administrateur.	68
IV.4 Formulaire de confirmation des demandes.	68
IV.5 formulaire 01 d'inscription d'un nouvel enseignant.	69
IV.6 formulaire 02 d'inscription d'un nouvel enseignant.	69
IV.7 formulaire d'inscription d'un nouveau groupe.	69
IV.8 Téléchargement d'une liste d'enseignant-étudiant.	70
IV.9 Ajouter une attestation.	70
IV.10 Consulter les attestations.	71
IV.11 Page d'accueil du compte enseignant.	72
IV.12 Formulaire de demande des supplémentaires.	72
IV.13 Liste des demandes des séances supplémentaires.	73
IV.14 Formulaire de demande d'une séance supplémentaire.	73
IV.15 Refuser une demande.	74
IV.16 Formulaire pour remplir une fiche de tutorat.	75
IV.17 Historique des fiches pour un étudiant.	75
IV.18 Page d'accueil du compte étudiant.	76
IV.20 Les statistiques d'un étudiant.	77

IV.19 profile un étudiant	77
IV.21 Liste des étudiants d'un groupe.	78

Liste des tableaux

III.1 Description des types des diagrammes et leurs utilisations	40
III.2 Description des cas d'utilisations de l'administrateur.	42
III.3 Description des cas d'utilisations de l'enseignant.	44
III.4 Description des cas d'utilisations de l'étudiant	46

Introduction générale

Introduction générale

La réforme globale du système d'enseignement supérieur nous a inévitablement amenés à nous pencher sur les exigences des besoins d'accompagnement des nouveaux étudiants universitaires, dans le but de les guider dans leurs recherches et de les aider à faire face aux nouvelles situations dans le système d'enseignement supérieur sous une nouvelle forme appelée accompagnement pédagogique ou tutorat [1]. L'objectif est de combler les différences entre l'étudiant et l'enseignant, d'améliorer la capacité de l'étudiant à définir son identité cognitive et scientifique, et de choisir les meilleures opportunités avec l'utilisation optimale des compétences et des aptitudes nécessaires pour construire son projet de formation dans l'espace universitaire.

Lors de la réalisation de cette nouvelle tâche, l'un des défis majeurs réside dans la gestion efficace d'une quantité substantielle de documents papier. Cette problématique requiert une approche professionnelle et stratégique afin de garantir une transition réussie vers des méthodes de gestion de l'information plus modernes et efficaces. La gestion des documents papier implique non seulement la manipulation et le stockage physique des documents, mais également la préservation de leur intégrité et de leur accessibilité tout au long du processus de réforme. Dans ce contexte, nous constatons que l'amélioration des performances du tutorat est très importante afin d'assurer l'efficacité de cette nouvelle tâche.

A l'ère numérique, qui a conduit à la matérialisation de nombreuses plateformes, notamment Progress, pour assurer la gestion de documents en ligne. Nous nous sommes intéressés à la numérisation des formulaires administratifs de la cellule d'accompagnement pédagogique pour les étudiants de première année de licence à l'université Mohamed Khider de Biskra.

C'est ainsi que nous avons eu l'occasion d'approfondir nos connaissances théoriques et techniques en concevant et réalisant une application web dédiée aux membres de la cellule, comprenant les enseignants, le personnel administratif et les étudiants.

Ce mémoire est constitué de quatre chapitres :

- Un chapitre introductif consacré pour présenter le concept du tutorat au sein de l'université.
- Un deuxième chapitre qui présente quelques notions sur les technologies web.
- Un troisième chapitre qui traite les étapes fondamentales pour la modélisation de notre application web.
- Un quatrième chapitre qui présente les différentes fonctionnalités de notre application. Ce chapitre sera conclu par la présentation de l'application développée et ses différentes captures écrans.

Chapitre I

Contexte de travail

1. Introduction

L'objectif de cette phase est de fournir une analyse sur le domaine du tutorat , mettant en évidence les fonctions de l'administrateur, des enseignants et des étudiants ainsi que ces différents aspects et niveaux de supervision.

En comprenant ces aspects clés, nous serons mieux équipés pour faire face aux débats et aux défis qui limitent son efficacité .

2. Définition du tutorat

Une des avancées majeures du système **L.M.D** prévu par l'article 2 du décret exécutif n° 09-03 du 6 Muharram 1430 correspondant au 3 janvier 2009.

Le tutorat est une mission de suivi et d'accompagnement de l'étudiant visant à faciliter son intégration dans la vie universitaire et à son accès aux informations. Il est destiné aux étudiants de 1ère année du premier cycle. [1]

3. Description des différentes tâches

3.1 Rôle et responsabilité du recteur de tutorat

L'administrateur ou le recteur de tutorat est responsable de la coordination de la supervision au sein de l'université. Il veille à la bonne conduite du processus d'encadrement des étudiants, assure le bon déroulement des séances de tutorat et s'assure de la satisfaction des enseignants et des étudiants. Le recteur de tutorat est également chargé d'assurer les tâches suivantes :

- Assignation d'un enseignant superviseur à chaque groupe.

- Établissement d'un calendrier de réunions de chaque superviseur avec ses groupes.
- Mise en place d'un espace approprié pour la communication avec les étudiants. [2]

3.2 Rôle et responsabilité du tuteur

Un superviseur ou un tuteur est un professeur de recherche à l'université son rôle consiste à encadrer de petites cohortes d'étudiants de première année, en partageant son expérience et ses compétences, ainsi qu'en leur enseignant des méthodes de travail pour les aider à acquérir des connaissances et à réussir leurs études. En site parmi ces tâches :

- La planification collaborative des cours de soutien si besoin.
- L'évaluation des besoins et objectifs de l'étudiant.
- L'utilisation des méthodes pédagogiques variées. [2]

3.3 Rôle et responsabilité du tuteur

Un étudiant universitaire est une personne inscrite dans une université pour suivre un programme académique. Il participe activement à des cours théoriques, des travaux pratiques, des projets de recherche et des activités extrascolaires, visant à développer son expertise et à préparer sa future carrière professionnelle ou académique. Pour soutenir cet objectif, le tutorat a été adopté comme un moyen essentiel d'encadrement. Toutefois, pour garantir l'efficacité de cette tâche et obtenir des résultats optimaux, le tuteur doit :

- Assister régulièrement aux séances de tutorat.

- Communiquer continuellement avec les tuteurs pour tirer le meilleur parti des conseils fournis.
- Se préparer à l'avance pour les séances en lisant les documents requis et en participant aux discussions.
- Participer activement aux activités et séances.[3]

4. Aspects et niveaux de supervision

L'article 2 du décret exécutif 03-09 daté du 6 Muharram 1430 correspondant au 3 janvier 2009, établit un cadre juridique pour la mission de supervision et définit comment elle doit être mise en œuvre. Ce décret précise le champ d'action et les responsabilités liées à la supervision dans le contexte concerné, en plus de définir les règles et les normes qui régissent l'exécution de cette mission. Grâce à cet article, les aspects principaux de la mission de supervision sont définis.[4]

4.1 Aspect média et administratif

Tout d'abord, l'accueil et l'orientation des étudiants de première année revêtent une importance majeure. Un accueil chaleureux et une orientation adéquate dès leur arrivée permettent aux nouveaux étudiants de se familiariser avec le campus et les services disponibles, créant ainsi une transition en douceur vers la vie universitaire. En outre, une explication claire du système LMD (Licence-Master-Doctorat), du système de tutorat et des nouvelles approches pédagogiques est essentielle. Cette compréhension permet aux étudiants de mieux naviguer dans leur parcours académique. Par ailleurs, encourager activement la participation aux séances de tutorat peut se faire par le biais de campagnes de sensibilisation, de rappels réguliers et de la mise en avant des bénéfices du tutorat pour leur développement

académique. Enfin, établir la confiance entre les enseignants et les étudiants est un élément clé. Les enseignants doivent créer un environnement propice où les étudiants se sentent en confiance pour poser des questions, exprimer leurs préoccupations et chercher de l'aide en cas de besoin, favorisant ainsi un apprentissage ouvert et collaboratif.[4]

4.2 Aspect Méthodologique

Cela englobe notamment l'enseignement des méthodes de travail universitaire, tant individuellement qu'en groupe. Tout d'abord, il est essentiel d'offrir aux étudiants des outils pratiques pour développer leurs compétences en matière de travail académique. Cela peut inclure des séances dédiées à l'apprentissage de la gestion du temps, de la prise de notes efficace, de la recherche documentaire et de la rédaction académique. En fournissant une formation méthodologique solide, les étudiants sont mieux équipés pour réussir dans leurs études et au-delà. De plus, l'apprentissage en groupe peut être un moyen efficace d'encourager la collaboration et le partage des connaissances entre pairs. Les sessions de travail en groupe peuvent permettre aux étudiants d'explorer différents points de vue, de résoudre des problèmes ensemble et de renforcer leur compréhension des concepts académiques. En combinant l'enseignement des méthodes de travail individuel et en groupe, les institutions peuvent offrir un soutien complet aux étudiants pour les aider à développer les compétences nécessaires à leur réussite universitaire et professionnelle.[4]

4.3 Aspect pédagogique

L'aspect pédagogique de l'accompagnement est essentiel pour le succès des étudiants. Il comprend plusieurs dimensions clés : l'accompagnement dans l'apprentissage, qui aide les étudiants à maîtriser les contenus académiques ; l'organisation

du travail personnel, qui leur enseigne à gérer efficacement leur temps et leurs études ; et l'aide à la construction de leur parcours éducatif, qui inclut l'élaboration de plans d'études personnalisés, la préparation aux opportunités futures, et le suivi pour combler les lacunes pédagogiques. Ce suivi identifie et adresse les difficultés spécifiques des étudiants avec des ressources adaptées. Ensemble, ces éléments créent un environnement de soutien optimal pour la réussite académique.[4]

4.4 Aspect professionnel

L'aspect professionnel de l'accompagnement est important pour préparer les étudiants à leur avenir. Cela inclut l'aide à la préparation de leur projet professionnel, en les guidant dans la définition de leurs objectifs de carrière, en fournissant des conseils sur les opportunités de stages et d'emplois, et en les aidant à développer les compétences nécessaires pour réussir dans le monde professionnel. Cet accompagnement permet aux étudiants de faire des choix éclairés et de se préparer efficacement à intégrer le marché du travail.[4]

4.5 Aspect technique

L'aspect technique de l'accompagnement est principal pour le développement des compétences pratiques des étudiants. Cela inclut l'orientation sur l'utilisation des outils pédagogiques, afin qu'ils puissent tirer pleinement parti des ressources disponibles pour leur apprentissage. En outre, l'accompagnement aide les étudiants à acquérir des compétences techniques essentielles à leur réussite, comme la recherche de références académiques, ainsi que l'utilisation des outils audiovisuels et d'Internet. Ce soutien technique permet aux étudiants de naviguer efficacement dans l'environnement numérique et d'exploiter les technologies pour enrichir leur parcours éducatif.[4]

4.6 Aspect psychologique

L’aspect psychologique de l’accompagnement est crucial pour le bien-être et la réussite des étudiants. Il inclut l’encouragement des étudiants à poursuivre leur parcours éducatif, en les motivant et en les soutenant face aux défis académiques et personnels. De plus, il aide les étudiants à s’intégrer dans la vie universitaire et associative, favorisant un sentiment d’appartenance et de soutien communautaire. Cet accompagnement psychologique contribue à créer un environnement propice à l’épanouissement personnel et académique des étudiants.[4]

Le schéma ci-dessous résume les différents aspects de l’accompagnement pédagogique :

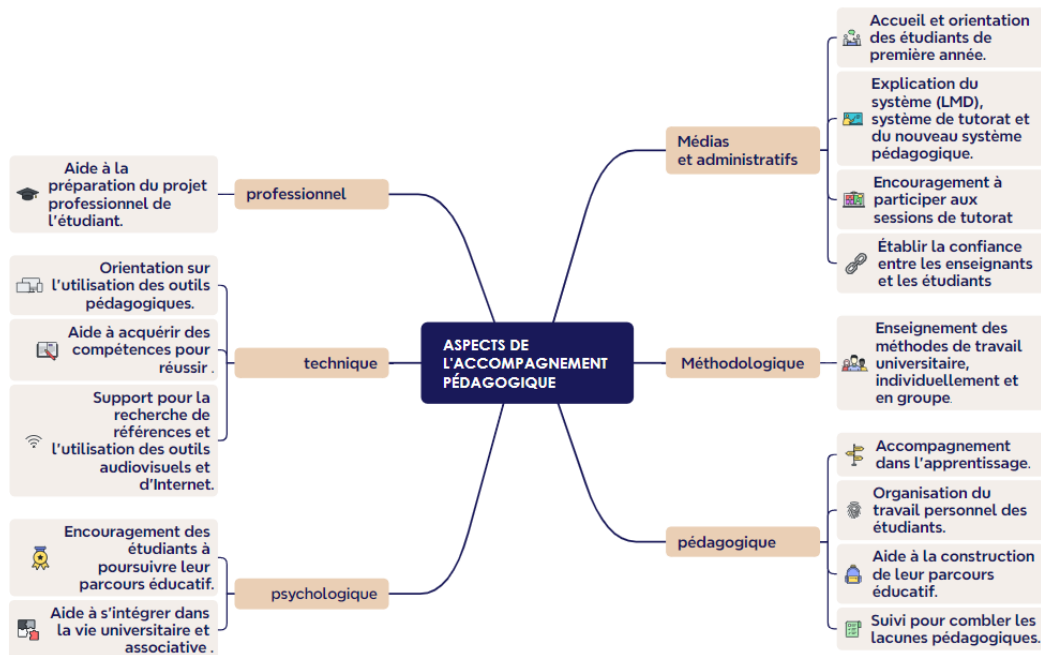


Figure I.1 : aspects de l’accompagnement pédagogique.

5. Etude de l'existant

Nous avons sélectionné la faculté de gestion des sciences économiques, commerciales et sciences de gestion de l'université Mohamed Khider de Biskra comme exemple en raison du succès de l'application de la tâche de tutorat. Les paragraphes suivants sont consacrés à la description de la cellule de tutorat dans cette dernière.

5.1 Bureau de tutorat

Le local dédié à la cellule de tutorat est situé au deuxième étage de la bibliothèque de la faculté des sciences économiques, commerciales et sciences de gestion de l'université Mohamed Kheider de Biskra. Cette cellule se compose de deux bureaux et d'un grand hall. Le premier bureau est attribué au responsable de la cellule de tutorat, tandis que le second est destiné au secrétariat. Le grand hall est aménagé avec tout le nécessaire pour accueillir les rencontres entre les enseignants tuteurs et les étudiants. Il convient de souligner que la taille du hall permet d'accueillir simultanément plus de dix groupes de tutorat, offrant ainsi une grande flexibilité au responsable pour planifier les sessions d'accompagnement.[2]

la figure suivante présente le plan de ce local au sein de la faculté :



Figure I.2 : Plan de l'unité de tutorat. [2]

5.2 Formulaire administratifs du tutorat

5.2.1 Fiche de tutorat

La fiche du tutorat est un outil de suivi qui permet à l'enseignant de consigner le contenu des séances du tutorat, les difficultés rencontrées par l'étudiant et les solutions proposées. Elle aide à suivre la progression de l'étudiant et garantit une approche structurée pour résoudre les problèmes académiques. En documentant les étapes du tutorat, la fiche permet également d'ajuster les stratégies pédagogiques en fonction des besoins des étudiants.[2]

الجمهورية الجزائرية الديمقراطية الشعبية
 République algérienne démocratique et populaire.
 وزارة التعليم العالي والبحث العلمي.
 Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Mohamed Khider - Biskra
 Faculté des Sciences Economiques,
 Commerciales et Sciences de
 Gestion.
 Tronc Commun (LMD1)



جامعة محمد خيضر بسكرة
 كلية العلوم الاقتصادية والتجارية وعلوم
 التسيير.
 قسم جذع مشترك سنة أولى.
 الرقم: أ. ج. م.

بطاقة مرافقة طالب السنة الأولى ليسانس

جلسة المرافقة رقم: بتاريخ: المكان: التوقيت:

اسم ولقب الأستاذ المرافق:	قسم:
اسم ولقب الطالب:	الفوج:
موضوع الجلسة:	
نصائح وتوجيهات الأستاذ:	
ملاحظات:	

توقيع الطالب: توقيع الأستاذ: توقيع مسؤول المرافقة البيداغوجية:
 رئيس القسم:


Figure I.3 : Fiche d'accompagnement de l'étudiant de première année licence. [2]

5.2.2 Attestation du tuteur pédagogique

L'attestation de tuteur est un document officiel remis par l'administration aux tuteurs qui ont atteint un certain nombre de séances du tutorat, selon le seuil fixé par l'administrateur. Elle certifie que le tuteur a apporté un soutien pédagogique aux étudiants de première année. Cette attestation démontre l'implication du tuteur et peut servir de référence pour des opportunités professionnelles ultérieures.[2]

الجمهورية الجزائرية الديمقراطية الشعبية
 République algérienne démocratique et populaire.
 وزارة التعليم العالي والبحث العلمي.
 Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Mohamed Khider - Biskra
 Faculté des Sciences Economiques,
 Commerciales et Sciences de
 Gestion.
 Tronc Commun (LMD1)



جامعة محمد خيضر بسكرة:
 كلية العلوم الاقتصادية والتجارية وعلوم
 التسيير.
 مجال جذع مشترك سنة أولى.
 الرقم: / م. ج. م.

شهادة أداء مهمة المرافقة لطلبة السنة الأولى ليسانس

يشهد السيد رئيس مجال جذع مشترك سنة أولى ليسانس – تحت إشراف السيدة عميد الكلية - أن
 الأستاذ(ة)..... ذا الرتبة..... من قسم.....
 قد قام بمهمة المرافقة البيداغوجية لطلبة السنة الأولى ليسانس، الفوج.....، وذلك للموسم الجامعي

ملاحظة: لا تستعمل هذه الشهادة إلا في نطاق ما يسمح به القانون، وهي تشكل إثباتا بقيام الأستاذ
 بمهمة المرافقة لطلبة السنة الأولى ليسانس.

عميد الكلية:
رئيس مجال جذع مشترك سنة أولى:

.....
.....

Figure I.4 : Certificat d’accompagnement ou attestation de tuteur pédagogique. [2]

5.2.3 Registre de présence des tuteurs

Le registre de présence des tuteurs enregistre les détails des séances de tutorat. Il contient le nom du tuteur, la date, l’heure, la présence ou l’absence des étudiants, et des observations. Chaque séance est signée par le tuteur pour confirmer la tenue de la session. Cela assure un suivi administratif précis.[2]

الإمضاء	ملاحظات	حضور الطلبة (X)		اسم الأستاذ المرافق	التوقيت	التاريخ
		لا	نعم			

Figure I.5 : Registre de présence des tuteurs. [2]

5.2.4 Registre de présence des étudiants

Le registre de présence des étudiants est rempli par l'enseignant à la fin de chaque séance de tutorat. Il répertorie les noms des étudiants présents, ainsi que des détails comme la date, l'heure, et des observations éventuelles. En signant le registre, l'enseignant confirme la tenue de la séance et la présence des étudiants. Ce registre est essentiel pour le suivi de l'assiduité des étudiants et la documentation des séances de tutorat. [2]

ملاحظات	الإمضاء		اسم الطالب	اسم الأستاذ	التوقيت	التاريخ
	الطالب	الأستاذ				

Figure I.6 : Registre de présence des tutorés.[2]

6. Système de gestion des formulaires administratifs

6.1 Problématique

Le responsable (administrateur) de la cellule de tutorat doit rassembler et organiser divers formulaires, comme les registres de présence et les rapports de suivi, pour gérer efficacement le tutorat et créer des archives administratives papier. Avec le nombre important d'étudiants de première année, cette tâche peut être ardue, ce

qui soulève des questions sur la gestion des ressources et la capacité à résoudre les problèmes qui peuvent survenir. Ont site parmi eux :

- La gestion substantielle de documents papier.
- Les défis de la communication : comprennent les difficultés de communication entre les accompagnateurs pédagogiques et les apprenants.
- Les défis de l'organisation et de la planification : peuvent inclure les pressions temporelles et la détermination des priorités lors de la planification et de l'organisation des séances d'accompagnement pédagogique.
- Les défis de l'apprentissage : englobent les difficultés d'apprentissage auxquelles sont confrontés les apprenants et que les accompagnateurs pédagogiques doivent les aider à surmonter.
- Les défis de l'évaluation : comprennent les difficultés d'évaluation des progrès éducatifs des apprenants de manière efficace et utile.

6.2 Objectif

Après avoir souligné les principaux problèmes des membres du tutorat, l'objectif est maintenant de présenter la solution via une application web conçue pour numériser les formulaires administratifs liés au tutorat.

Notre application facilite la gestion des documents administratifs, comme les registres de présence et les rapports de suivi, réduisant ainsi le travail manuel pour les enseignants et l'administrateur. Pour les étudiants, elle offre un accès facile aux horaires des séances, des rappels et des ressources pédagogiques. Cela crée un environnement de tutorat plus fluide et réduit les charges administratives.

7. Conclusion

Avec cette compréhension approfondie de la mission du tutorat, de ses rôles et des défis rencontrés par ses membres, le prochain chapitre se concentrera sur les technologies des applications web et leur potentiel pour répondre à ces défis et simplifier la gestion de cette tâche.

Chapitre II

Les technologies web

1. Introduction

Ce chapitre se concentre sur les principes fondamentaux des applications web. Il commence par un bref historique du **Web**, puis introduit la définition des applications web. Ensuite, il examine les différents types d'applications, avec un accent particulier sur le type que nous avons choisi, mettant en évidence ses caractéristiques. Nous présentons également quelques architectures utilisées pour le développement de ces applications. Enfin, nous expliquons comment nous avons tiré parti des avantages des applications web dans le contexte de notre projet, concluant avec des travaux connexes.

2. Historique du Web

Le World Wide Web (WWW), ou Web, est un système qui permet de consulter des documents et des ressources via Internet à l'aide de navigateurs web. Il a été inventé par Tim Berners-Lee en 1989 au **CERN** pour permettre aux scientifiques du monde entier de partager des informations. Le projet a d'abord été connu sous le nom de WorldWideWeb, puis a été changé en World Wide Web pour améliorer la lisibilité. Le Consortium **W3C** recommande actuellement de l'écrire sous cette forme.

Le développement initial du Web a été dirigé par Tim Berners-Lee et Robert Cailliau, mais l'apparition du navigateur **NCSA Mosaic**, créé par Eric Bina et Marc Andreessen, a joué un rôle clé en introduisant des interfaces graphiques. Cela a mené à l'émergence de Netscape Navigator et d'autres navigateurs comme Internet Explorer, qui ont aidé à propulser la croissance du Web.

En 1993, le CERN a rendu les technologies du WWW accessibles au public, ce qui a accéléré la croissance du Web. Au fil du temps, des entreprises comme

Amazon.com, fondée par Jeff Bezos en 1995, ont commencé à utiliser le Web pour le commerce électronique. Dans les années 2000, Google est devenu un acteur majeur avec son moteur de recherche, et des concepts comme "Web 2.0" ont émergé, indiquant des changements dans la façon dont les utilisateurs interagissent avec le Web. Des technologies comme JavaScript et **PHP** ont permis la création de sites web plus interactifs et dynamiques.

En 2010, le **HTML** a évolué vers HTML5, apportant des améliorations significatives au langage de balisage. En 2020, l'utilisation des médias sociaux avait augmenté, avec environ 3,8 milliards de personnes, soit près de la moitié de la population mondiale, utilisant ces plateformes, indiquant une croissance rapide de l'engagement en ligne. [5]

3. Les applications web

3.1 Définition

Une application web est un logiciel applicatif hébergé sur un serveur, accessible via un navigateur web, et manipulable directement en ligne. Contrairement à un logiciel traditionnel, il n'est pas nécessaire d'installer une application web sur son ordinateur : l'utilisateur peut simplement y accéder via son navigateur favori. Ces applications se connectent généralement par le biais d'un réseau informatique, comme Internet ou un réseau local.[6]

3.2 Types des applications

On peut distinguer six types d'applications web, comme illustré dans la figure suivante :

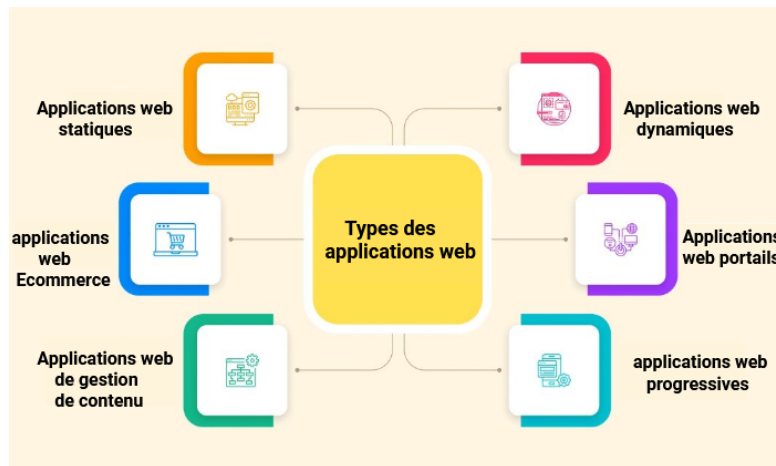


Figure II.1 : Les types des applications web. [7]

3.2.1 Applications web statiques

Les applications web statiques fournissent les mêmes informations à tous les utilisateurs, sans personnalisation et avec une flexibilité limitée. Chaque changement ou mise à jour nécessite un rechargement complet de la page. [8]

3.2.2 Applications web e-commerce

Les applications web de commerce électronique, comme leur nom l'indique, sont des logiciels destinés à faciliter diverses activités commerciales en ligne, telles que la vente et l'achat de produits, la passation de commandes, et le traitement des paiements.

Le développement d'une application web de commerce électronique exige la mise à jour régulière d'une base de données pour la gestion des produits et des commandes. Cette base de données est essentielle pour assurer le bon fonctionnement de l'application, en permettant une gestion efficace des stocks, des transactions et des informations client.[8]

3.2.3 Applications web Portails

Un portail web est un site web spécial conçu pour servir de passerelle et donner un accès pratique à d'autres sites connexes. Ces applications agissent en effet comme des portes d'entrée vers d'autres ressources en ligne, agrégeant des informations provenant de diverses sources pour les rendre disponibles aux utilisateurs. De plus, elles fournissent souvent des services supplémentaires tels que l'orientation des utilisateurs et la création de communautés virtuelles basées sur des intérêts communs. Cependant, la manière dont chaque portail représente les intérêts de sa communauté peut varier et être sujette à interprétation.[9]

3.2.4 Applications web progressives

Les applications web progressives représentent une nouvelle catégorie d'applications en ligne qui fusionnent les avantages des applications natives avec la facilité d'utilisation caractéristique des applications web. Initialement simples, elles évoluent progressivement en acquérant de nouvelles fonctionnalités au fur et à mesure que l'utilisateur interagit avec elles, se transformant ainsi en une expérience proche de celle des applications natives traditionnelles.[10]

3.2.5 Applications web de gestion de contenu

Pour les applications web nécessitant des mises à jour constantes du contenu, l'utilisation d'un système de gestion de contenu **CMS** est recommandée. Cela permet à l'administrateur d'effectuer les modifications et les mises à jour de manière autonome. Ces plateformes sont conviviales et faciles à utiliser. Dans le cadre de mon étude sur les différents types d'applications web, j'ai choisi de me concentrer sur les applications dynamiques, car ma propre création correspond à ce type d'application.[11]

Dans le cadre de notre étude sur les différents types d'applications web, nous avons choisi de nous concentrer sur les applications dynamiques, car notre propre création correspond à ce type d'application.

3.2.6 Applications web dynamiques

Une application web dynamique est un ensemble de pages statiques et dynamiques dont le contenu de ces pages est partiellement ou entièrement indéterminé. Le contenu final d'une page est déterminé uniquement lorsque l'utilisateur requiert une page depuis le serveur web, et ce dernier varie d'une requête à une autre en fonction des actions de l'utilisateur.

Une page statique est une page qui n'est pas modifiée lorsqu'un visiteur la consulte. Le serveur web la transmet au navigateur qui la sollicite sans la modifier, par contre une page dynamique elle est modifiée par le serveur avant d'être transmise au navigateur qui la sollicite. L'application web dynamique est plus complexe sur le plan technique, car elle utilise des bases de données pour charger des informations, et le contenu est mis à jour à chaque fois que l'utilisateur se connecte à l'application. En général, elle a un panneau d'administration depuis lequel l'administrateur peut corriger ou modifier le contenu.[12]

- **Caractéristiques des applications web dynamiques**
- Contenu dynamique et interactive : permettent aux utilisateurs de s'engager avec le contenu de manière interactive, ce qui peut entraîner des changements en temps réel en fonction de divers facteurs, tels que les interactions des utilisateurs, les mises à jour de bases de données ou des événements extérieurs.
- Interaction avec les bases de données : interagissent généralement avec des bases de données pour stocker, récupérer et manipuler des données. Cette

interaction permet de personnaliser le contenu et de le rendre pertinent pour chaque utilisateur.

- Utilisation de technologies côté Client et côté Serveur : Les applications web dynamiques utilisent généralement des technologies côté Client, comme JavaScript, pour améliorer l'interactivité et la réactivité. Elles utilisent également des technologies côté Serveur, comme PHP, Python, Ruby, ou Node.js, pour traiter des requêtes, gérer des sessions et interagir avec des bases de données.
- Communication asynchrone : Les applications web dynamiques peuvent utiliser des techniques de communication asynchrone, comme **AJAX** (Asynchronous JavaScript and XML), pour envoyer et recevoir des données sans recharger la page entière. Cela améliore l'expérience utilisateur en rendant l'application plus réactive.[13]

4. Architecture des applications web

L'architecture des applications web représente un modèle qui joue un rôle essentiel dans le développement réussi d'une application web. Il englobe les interactions et les relations entre divers éléments de l'application, comme les interfaces utilisateur, les systèmes middleware et les bases de données. Ce modèle inclut un ensemble de composants et décrit la manière dont ils interagissent de façon logique.[14]

4.1 Architecture Client-Serveur

L'architecture Client-Serveur est une structure informatique où le serveur héberge, fournit et gère la plupart des ressources et des services destinés au client.

Cette architecture est couramment utilisée dans des applications nécessitant des échanges fréquents de requêtes et de réponses, le serveur étant responsable de traiter les données et de les distribuer aux clients.[15]

la figure suivante résume la communication Client-Serveur :

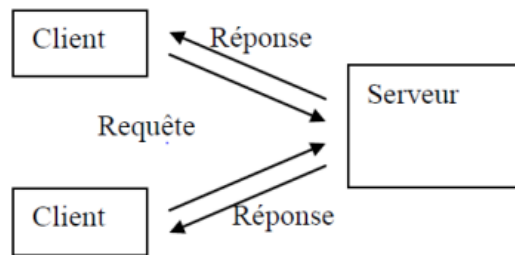


Figure II.2 : Le principe du fonctionnement de l'architecture Client/Serveur. [16]

De plus, l'architecture Client-Serveur peut être implémentée à différents couches, tels que l'architecture en deux couches, l'architecture en trois couches et l'architecture multicouche, chacun offrant divers degrés de modularité, de scalabilité et de gestion de la complexité.

- **architecture en deux couches (2 tiers)**

Dans cette architecture, le client demande un service au serveur, qui répond directement sans besoin d'un serveur intermédiaire.[16]

- **architecture en trois couches (3 tiers)**

Cette architecture vise à simplifier et à améliorer le processus de développement d'applications en résolvant divers problèmes récurrents de conception et de développement. Elle est appelée architecture en trois couches car elle est divisée en trois niveaux distincts : la couche d'interface utilisateur, la couche de logique d'application et la couche de base de données.

- la couche d'interface utilisateur représente la composante du logiciel avec laquelle les utilisateurs interagissent directement. Elle gère les données d'entrée/sortie et leur affichage, mais son rôle va au-delà de la simple présentation des informations. Une interface utilisateur efficace vise à offrir une expérience utilisateur intuitive et conviviale, facilitant ainsi l'interaction avec le système. Elle peut prendre différentes formes selon la plateforme ou la technologie utilisée, et doit être conçue de manière à être accessible via divers appareils tels que les smartphones, les tablettes et les ordinateurs.
- la couche logique d'application agit comme un pont entre l'interface utilisateur et la couche de base de données, Les composants de ce niveau reçoivent des demandes provenant de l'interface utilisateur, interagit avec la base de données pour obtenir les réponses appropriées, puis renvoie ces réponses à l'interface utilisateur pour les présenter aux utilisateurs. En d'autres termes, il joue un rôle essentiel dans le traitement des interactions entre l'utilisateur et les données en arrière-plan, garantissant que les informations appropriées sont échangées de manière efficace et sécurisée.
- La couche de base de données Le rôle de cette couche est d'assurer la modélisation, le stockage, et l'optimisation d'accès aux informations nécessaires au système. Les données nécessaires par la couche de logique d'application sont récupérées depuis la base de données. Ces dernières sont organisées dans des tables bien structurées pour faciliter l'accès, la gestion et l'interaction avec l'application. [17]

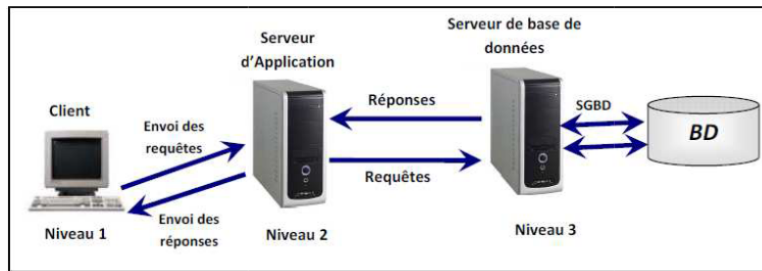


Figure II.3 : Principe du fonctionnement de l’architecture en trois couches. [16]

- **architecture multicouche**

L’architecture multicouche fonctionne de manière similaire à l’architecture en trois couches , mais elle offre la possibilité d’ajouter des niveaux supplémentaires. Concrètement, cela signifie que le serveur de niveau intermédiaire peut faire appel à un autre serveur pour accéder aux ressources nécessaires.[16]

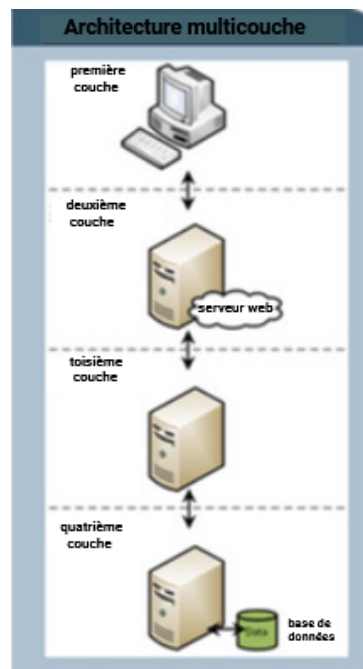


Figure II.4 : Principe du fonctionnement de l’architecture multicouche. [18]

5. fonctionnement d'une application web

Lorsqu'une demande est adressée au serveur web pour une page web statique, celui-ci la transfère simplement au navigateur qui l'a sollicitée. En revanche, lorsqu'il reçoit une demande pour une page dynamique, il la transfère à un serveur d'application chargé de la compléter. Ce serveur lit le code contenu dans la page, exécute les instructions, intègre les résultats dans la page, puis élimine le code. Le résultat est une page statique que le serveur d'application renvoie au serveur web. Ce dernier transmet ensuite cette page au navigateur qui l'a demandée.[19]

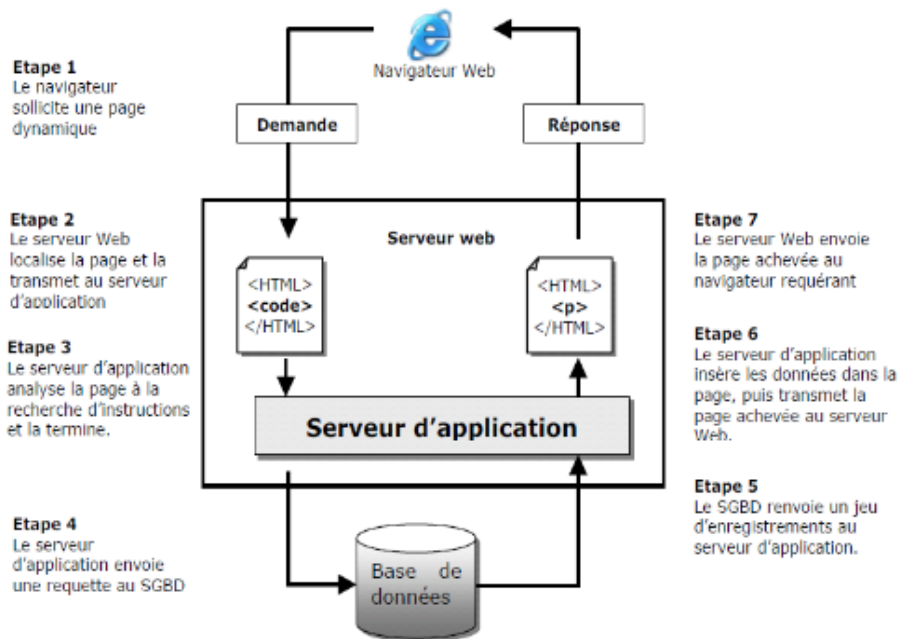


Figure II.5 : Principe du fonctionnement des applications web. [19]

6. Technologies des applications web

Cette section présente des sous-titres qui explorent les différents aspects de ces technologies, y compris les langages de programmation, les technologies côté client, et les technologies côté serveur.

6.1 Langages de programmation

Les langages de programmation sont au cœur du développement des applications web, chacun offrant des avantages spécifiques pour différents types de tâches. Parmi eux, on trouve Python, apprécié pour sa syntaxe simple et ses frameworks puissants. Pour développer des projets complexes, PHP et C Sharp sont privilégiés. Le langage Java est également reconnu pour sa portabilité, sa gestion automatisée de la mémoire et son modèle de sécurité robuste. De plus, JavaScript, essentiel pour le développement web, rend les sites internet interactifs et est également utilisé côté serveur avec Node.js, ainsi qu'avec des frameworks comme React Native et Ionic. Parallèlement, **SQL** est un langage de programmation essentiel pour les bases de données relationnelles, permettant de stocker, manipuler et récupérer des données de manière standardisée et efficace.[20]

6.2 Technologies côté client

Les technologies côté client, souvent appelées technologies frontend, sont essentielles pour concevoir l'expérience utilisateur des applications web. Elles comprennent des langages clés tels que HTML5, CSS3 et JavaScript, qui servent à définir la structure, le contenu et l'apparence des pages web, ainsi qu'à les rendre interactives. Ces technologies permettent de traiter et de gérer les entrées utilisateur directement sur l'appareil de l'utilisateur, que ce soit un ordinateur ou un

smartphone, sans nécessiter une communication constante avec un serveur. ces technologies ont plusieurs types, On compte :

- **Client lourd** : Un client lourd se caractérise par le fait qu'il conserve une part importante de ses capacités de traitement et de stockage sur son propre dispositif. Grâce à cela, il peut fournir une interface utilisateur sophistiquée ainsi que des fonctionnalités avancées, sans nécessiter une connexion constante à un serveur distant.
- **Client riche** : Un client riche en fonctionnalités propose une expérience utilisateur avancée, caractérisée par des fonctionnalités interactives et dynamiques. Ces clients peuvent exploiter des technologies telles que JavaScript et AJAX afin de créer une interface utilisateur réactive et une expérience utilisateur fluide.
- **Client web** : Les clients web sont des applications accessibles via un navigateur web. Ils peuvent être utilisés sur différents appareils et systèmes d'exploitation sans nécessiter d'installation préalable, ce qui les rend facilement accessibles à un large public.
- **Client monopage (SPA)** : Les applications à page unique **SPA** sont des applications web qui chargent le contenu de manière dynamique et mettent à jour l'interface utilisateur en fonction des actions de l'utilisateur, sans nécessiter de rechargement de la page entière. Cela permet de créer une expérience utilisateur fluide et rapide, similaire à celle des applications natives.
- **client léger** : Un client léger est défini comme un dispositif ou une application qui repose largement sur les ressources et le traitement offerts par un serveur distant. Typiquement, il se caractérise par une interface utilisateur minimale et une capacité de traitement limitée.[21]

6.3 Technologies côté serveur

Ce sont des technologies chargées du logique métier, du traitement des données et de la communication avec les bases de données, et Assurent le traitement des requêtes, l'exécution de scripts côté serveur, la gestion des bases de données relationnelles telles que MySQL et PostgreSQL, ainsi que des bases de données non relationnelles telles que MongoDB et Firebase, etc. Parmi les technologies les plus couramment utilisées côté serveur on retrouve JavaScript avec Node.js, Python avec Django et Flask, Ruby avec Ruby on Rails, ainsi que PHP avec Laravel. Ces frameworks fournissent des fonctionnalités robustes pour gérer les requêtes **HTTP**, gérer l'authentification et l'autorisation.[21]

7. avantages des applications web

- Accessibilité multiplateforme : Fonctionne en ligne via un ce qui les rend accessibles depuis n'importe quel type d'appareil, que ce soit un pc, un mobile, une tablette, un ordinateur, etc.
- Aucune installation nécessaire : Ne nécessite pas de téléchargement pour consulter l'application donc pas de consommation de stockage
- Compatibilité avec tous les systèmes d'exploitation : fonctionne sur une grande variété de systèmes d'exploitation, sans souci de plateformes.
- Mises à jour automatiques : Réduire le besoin de gérer des installations et des mises à jour manuelles, car elles sont hébergées sur des serveurs centraux.

D'après les avantages mentionnés précédemment, nous avons choisi ce type d'application pour notre travail car il correspond parfaitement à nos besoins opérationnels. Cela nous garantit les éléments suivants :

- Une application offre diverses options d'accessibilité pour les membres du tutorat, avec l'avantage supplémentaire de ne nécessiter aucune installation. L'administrateur peut accéder depuis son ordinateur de bureau, ce qui simplifie la gestion sans ajouter de charge supplémentaire. L'enseignant peut utiliser son ordinateur portable pour un accès instantané, sans configurations complexes. Quant à l'étudiant, il peut utiliser son téléphone mobile pour une connexion rapide, sans occuper de stockage ni compromettre les performances de son appareil.
- L'administrateur obtient instantanément les dernières fonctionnalités et correctifs sans avoir à gérer des installations manuelles, l'enseignant bénéficie de nouvelles améliorations et de corrections de bugs sans interrompre son flux de travail, et l'étudiant a toujours accès à la version la plus récente, ce qui garantit une expérience utilisateur cohérente et à jour.
- L'administrateur peut travailler sur son ordinateur Windows ou Mac, l'enseignant peut utiliser son ordinateur portable sous Linux ou MacOS, et l'étudiant peut accéder à l'application depuis son téléphone mobile Android ou iOS. Cette compatibilité universelle assure une expérience transparente pour tous, quel que soit le système d'exploitation utilisé.[22]

8. Travaux connexes

Dans divers secteurs, la numérisation des documents et des formulaires est devenue une pratique répandue, en réponse à la demande croissante d'efficacité opérationnelle et de réduction des coûts. Cette partie se concentre sur la présentation de quelques travaux récents ayant pour objectif le développement d'applications web visant à numériser les documents dans divers domaines.

Kufre Effiong Okon et al [23] ont développé une application web de gestion de revues pour une étude de cas du Journal de Recherche et d'Innovations en Ingénierie (JORIE) de l'Université de Uyo. L'objectif principal de leur travail est de créer un système de gestion de revues pour améliorer la visibilité et l'efficacité de la publication des recherches académiques. En utilisant un modèle de développement itératif-incrémental pour la conception du logiciel, et la mise en œuvre d'une architecture à trois niveaux pour l'application, en utilisant le modèle-vue-contrôleur (MVC) pour séparer les composants logiques.

Les auteurs ont présenté les résultats de l'implémentation du système avec des captures d'écran des pages web nécessaires, montrant le fonctionnement de l'application de gestion de revues. L'application permet aux utilisateurs de visualiser, lire en ligne, rechercher et télécharger des articles publiés, tout en offrant une gestion efficace du flux de travail éditorial, y compris la soumission de manuscrits, l'affectation aux éditeurs et aux relecteurs, la décision éditoriale et la publication des articles.

Le système a été développé en utilisant le framework ASP.Net Core 3.1, le framework Vue.js, la base de données Microsoft SQL Server (MSSQL), il a été hébergé sur le serveur Internet Information Services (IIS) [23].

Nurfitria Khoirunnisa et al [24] ont présenté leur travail intitulé « la conception d'un système d'information de gestion administrative basé sur un site web »

pour l'association des étudiants en gestion informatique (HIMMI) sur le campus du Polytechnic d'État de Subang en Indonésie. ils ont créé un système qui facilite le stockage de documents, rationalise le processus d'émission de lettres sortantes, résout le problème du numéro de plusieurs e-mails, rende le processus de présence plus efficace, fournisse un calendrier des programmes de travail et offre des informations relatives aux rapports requis.

L'architecture utilisée dans leur travail est le framework Laravel avec une architecture MVC (Modèle-Vue-Contrôleur). Le modèle MVC sépare le système en trois composants principaux : le Modèle, la Vue et le Contrôleur, pour gérer les données, afficher les pages web et connecter les données avec la vue, respectivement. Les outils de programmation utilisés pour développer ce système comprennent le langage de programmation PHP 8 avec le framework Laravel pour l'implémentation du système, et MySQL pour la base de données. De plus, HTML, jQuery, JavaScript et CSS ont été utilisés pour les composants de vue du système.

Les résultats de ce travail comprennent la mise en œuvre réussie du système d'information de gestion administrative basé sur un site web, qui a été testé par 3 testeurs sur 46 éléments de test. Le système a montré des procédures d'action valides à 100 pour cent. [24]

Samuel M. Alade concentre sur la conception et la mise en œuvre d'un système de gestion de documents basé sur le Web en utilisant les technologies web et la méthodologie de conception hypermédia orientée objet (OOHDM), pour répondre aux problèmes récurrents causés par une gestion de documents inadéquate dans les organisations en introduisant une gestion systématique des documents et en promouvant des méthodes électroniques pour le suivi du flux de travail des documents et leur récupération. L'objectif de son travail est de concevoir et mettre en œuvre un système convivial basé sur le Web pour une gestion efficace des documents et des enregistrements, d'analyser les systèmes existants, de modéliser,

concevoir et développer les fonctionnalités du système de gestion de documents en fonction des exigences recueillies, et d'évaluer le système.

L'architecture du système utilisée dans son travail est une architecture à trois niveaux, avec une interface client, un middleware et un référentiel de base de données. Les outils de programmation utilisés pour développer ce système comprennent l'outil de programmation NetBeans pour l'implémentation, qui fournit un environnement de développement intégré (IDE). De plus, le système a été développé en utilisant les langages de programmation HTML, CSS, JavaScript, PHP, et MySQL pour la conception et la programmation de la base de données.

Les résultats montrent que le système de documents développé aide à l'organisation efficace et sécurisée des documents et des enregistrements d'une organisation, ce qui conduit à une productivité accrue et à une meilleure gestion des projets de construction.[25]

Chapitre III

Conception et architecture de système

1. Introduction

Ce chapitre se concentre sur la conception de notre application, en mettant en avant trois éléments clés : la méthode en cascade, l'architecture en deux couches et le langage de modélisation UML. Nous débuterons par la définition de l'architecture utilisée et son fonctionnement. Ensuite, nous explorerons l'utilisation du langage UML pour représenter graphiquement les différentes facettes de notre système et pour modéliser ses fonctionnalités. Enfin, nous soulignerons l'importance de cette phase de conception dans le processus global de développement de notre application.

2. Méthode en cascade

Le modèle de développement en cascade, introduit par W. Royce en 1970, est l'un des premiers modèles formels de gestion de projet. Il se caractérise par une approche linéaire et séquentielle, découpant le développement en phases distinctes et ordonnées. Traditionnellement, il comprend cinq phases principales : l'analyse des besoins, la conception, l'implémentation, le test et la mise en production. Chaque phase doit être complétée avant de passer à la suivante, assurant ainsi une progression méthodique du projet. Ce modèle vise à minimiser les risques en identifiant clairement les exigences et en testant rigoureusement chaque étape avant la livraison finale du produit logiciel.[26]

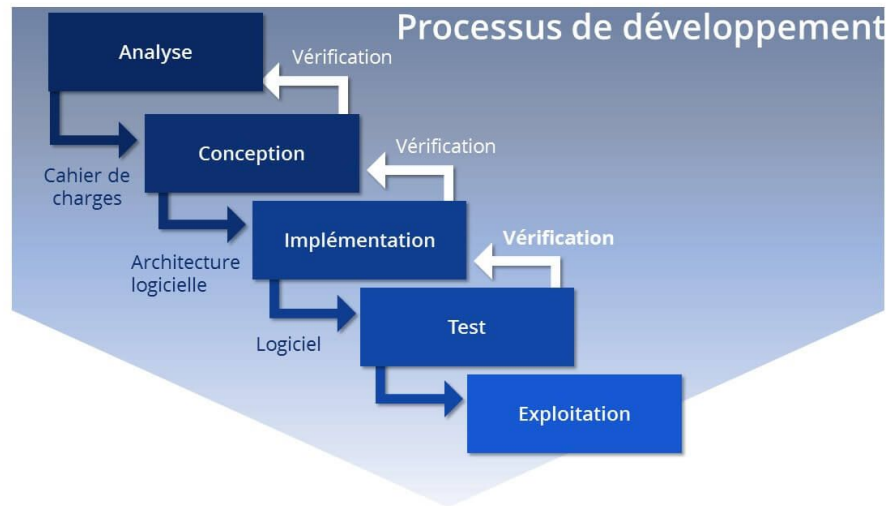


Figure III.1 : Le modèle en cascade présenté schématiquement.[26]

3. Architecture proposée

L'architecture proposée dans notre projet est l'architecture à deux niveaux, également connue sous les termes de Client-Serveur de données ou Client-Serveur de première génération. Pour mieux comprendre cette architecture, nous avons détaillé sa présentation comme suit :

3.1 Description

cette architecture opère selon le principe suivant : le client sollicite un service auprès du serveur, lequel exécute cette requête et fournit directement la réponse au client, sans nécessiter l'intervention d'un serveur intermédiaire. [27]

- **Client** : généralement un navigateur web comme Firefox, chrome, etc. qui exécute le code front end et interagit avec le serveur par l'envoi d'une requête en utilisant son adresse IP et le port associé, qui identifie un service

spécifique sur le serveur. [16]

- **Serveur** : c'est une application qui gère la logique métier, interagit avec la base de données, et traite les requêtes envoyées par les clients et répond en utilisant l'adresse IP et le port de la machine cliente. [16]

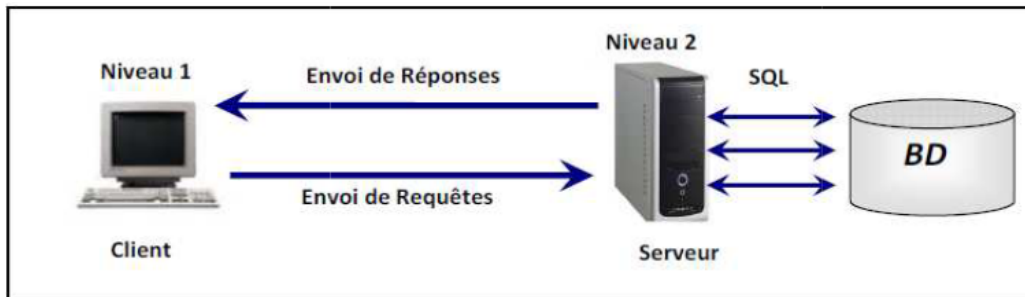


Figure III.2 : Principe du fonctionnement de l'architecture en deux couches. [16]

3.2 Avantages

- **Performance** : En réduisant le nombre de couches intermédiaires, les échanges entre le client et le serveur deviennent plus rapides.
- **Centralisation des données** : Les données sont stockées et gérées sur un serveur centralisé, ce qui simplifie leur gestion, leur sauvegarde et leur sécurité.
- **Interfaces utilisateur riches et interactives** : Cela permet d'améliorer l'expérience utilisateur grâce à des applications plus réactives et personnalisées, adaptées aux besoins spécifiques des utilisateurs finaux.
- **Facilité de déploiement** : En raison de sa simplicité intrinsèque, les applications en deux tiers sont souvent plus simples à déployer et à mettre à l'échelle. [28]

Après avoir présenté l'architecture Client-Serveur qui structure notre système, nous passons maintenant à la modélisation de notre application de manière claire et organisée. Pour ce faire, nous utilisons le langage de modélisation UML (Unified Modeling Language).

4. langage de modélisation UML

UML (Unified Modeling Language) est un langage de modélisation visuel polyvalent utilisé pour concevoir et documenter des systèmes logiciels. Il est conçu pour spécifier, visualiser, construire et documenter les composants d'un système. UML permet de modéliser la structure statique d'un système, comme les types d'objets et leurs relations, ainsi que son comportement dynamique, comme les interactions entre objets au fil du temps.[29]

5. Diagramme UML

Un diagramme UML est une représentation graphique organisée qui facilite la compréhension de la structure d'un système, y compris son architecture et sa conception. Ces modèles UML se divisent en deux types : structurels, qui illustrent les éléments statiques d'un système, et comportementaux, qui décrivent le comportement des composants entre eux. La figure suivante présente 14 types principaux de diagrammes utilisés pour modéliser des systèmes logiciels ou des systèmes complexes.[30]

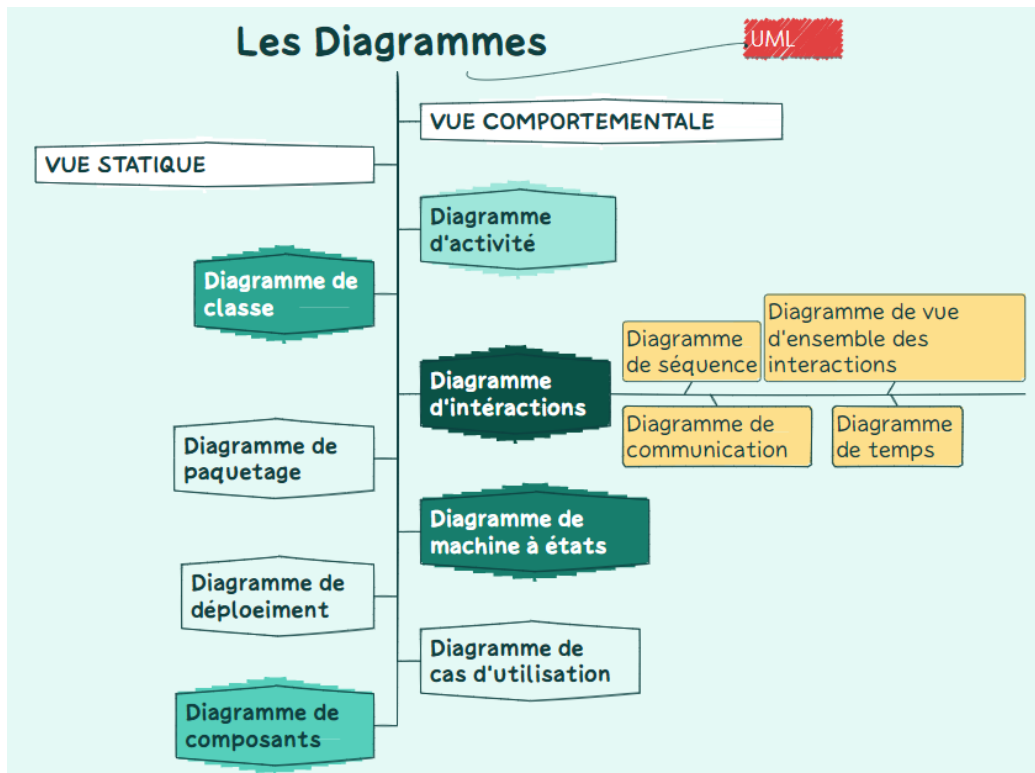


Figure III.3 : Schéma des diagrammes UML. [31]

D’après l’ensemble des 14 types de diagrammes UML, j’ai utilisé certains d’entre eux pour illustrer la vue statique du système, tandis que d’autres ont servi à décrire le comportement et les interactions entre les composants, fournissant ainsi une perspective complète de l’architecture et du fonctionnement de mon système. Le tableau ci-dessous répertorie ces diagrammes :

Diagramme	Type	Permet de
Cas d'utilisation [32]	Comportemental	<ul style="list-style-type: none"> • recueillir les exigences d'un système • obtenir une vue externe d'un système • identifier les facteurs externes et internes qui influencent le système • montrer les interactions entre les exigences et les acteurs
Classe [33]	Statique	<ul style="list-style-type: none"> • Décrire la structure statique du système • Spécifier la structure et le comportement communs d'un ensemble d'objets
Séquence [34]	Comportemental	<ul style="list-style-type: none"> • Décrire la séquence des actions nécessaires pour accomplir une tâche ou un scénario • Fournir une vue dynamique du comportement du système • Explorer la logique d'une opération, d'une fonction ou d'une procédure complexe

Tableau III.1 : Description des types des diagrammes et leurs utilisations

6. Diagramme de cas d'utilisation

6.1 Administrateur

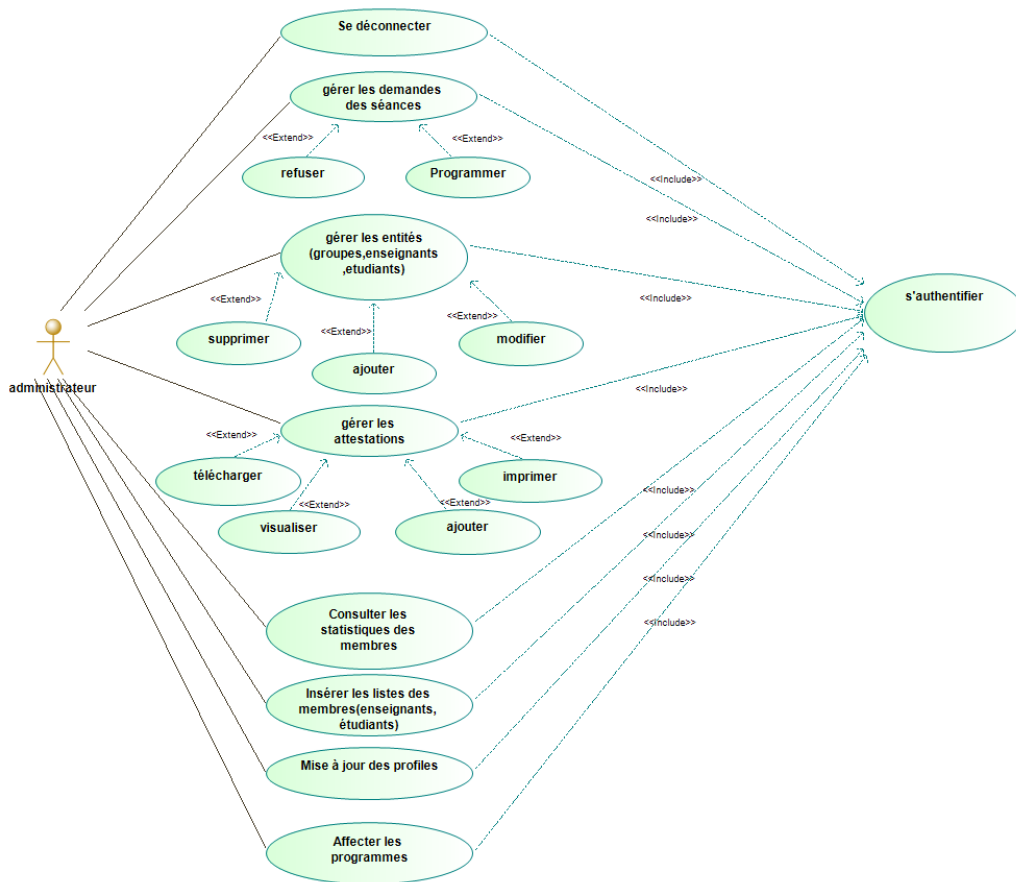


Figure III.4 : Diagramme de cas d'utilisation de l'administrateur.

- **Illustration de diagramme**

Le tableau ci-dessous décrit les différents cas d'utilisation associés au rôle de l'administrateur, avec une brève explication de leur objectif et de leur signification :

Cas d'utilisation	Signification
S'authentifier	désigne le processus par lequel l'administrateur prouve son identité pour accéder au système, à l'aide d'un nom d'utilisateur et d'un mot de passe.
Gérer les demandes	permet à l'administrateur d'accéder à la liste des demandes de séances soumises par les utilisateurs. L'administrateur peut alors choisir de valider et programmer une séance ou de refuser la demande, selon les critères appropriés.
Gérer les entités	englobe les enseignants, les étudiants et les groupes, permet à l'administrateur d'effectuer diverses tâches telles que l'ajout, la suppression ou la modification d'une entité.
Gérer les attestations	permet à l'administrateur de créer des attestations pour les enseignants qui ont atteint un certain nombre de séances de tutorat, conformément au seuil fixé par l'administration. L'administrateur peut également télécharger, imprimer ou visualiser ces attestations.
Insérer les listes des membres	permet à l'administrateur d'importer directement des listes au format Excel d'enseignants ou d'étudiants, qui sont ensuite automatiquement enregistrés dans le système.
Consulter les statistiques et les profils des membres	offre une vue précise sur l'activité des membres, en fournissant des données détaillées sur leur niveau d'interactivité. De plus, la consultation des profils permet à l'administrateur d'accéder à des informations spécifiques sur chaque membre.

Tableau III.2 : Description des cas d'utilisations de l'administrateur.

6.2 Enseignant

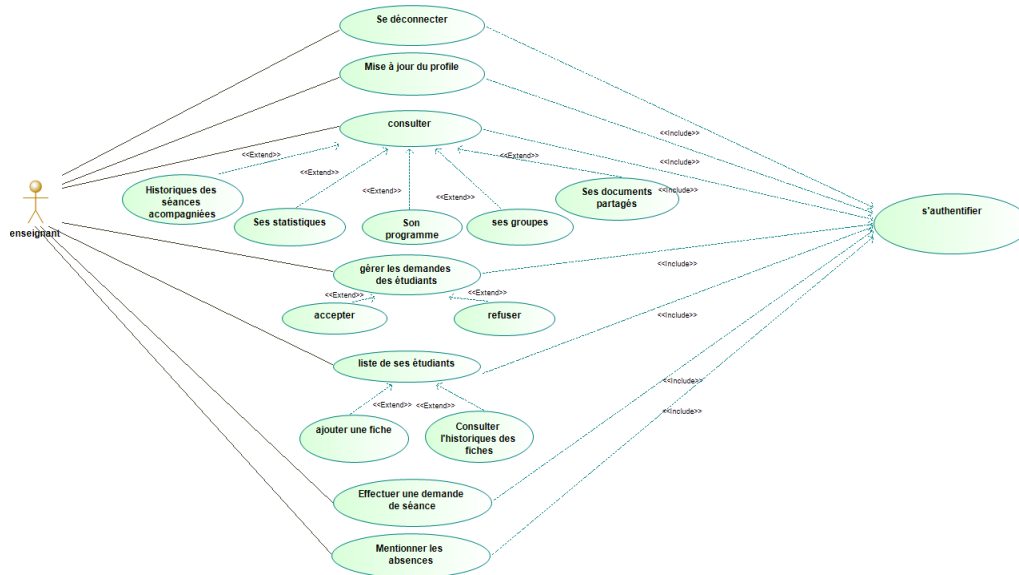


Figure III.5 : Diagramme de cas d'utilisation de l'enseignant.

- **Illustration de diagramme**

Le tableau ci-dessous décrit les différents cas d'utilisation associés au rôle de l'enseignant, avec une brève explication de leur signification.

Cas d'utilisation	Signification
S'authentifier	Désigne le processus par lequel l'enseignant prouve son identité pour accéder au système, à l'aide d'un nom d'utilisateur et d'un mot de passe.
Gérer les demandes des étudiants	Permet à l'enseignant d'accéder à la liste des demandes de séances soumises par les étudiants. L'enseignant peut alors choisir de valider une séance ou de refuser la demande, selon les critères appropriés. Si l'enseignant valide la séance, une demande est soumise à l'administrateur pour la traiter.
Consulter	Englobe plusieurs tâches. L'enseignant peut consulter les groupes auxquels il appartient et les documents qu'il a partagés avec eux. Il peut également consulter l'historique des séances qu'il a enseignées, ainsi que ses statistiques qui fournissent des données détaillées sur son niveau d'interactivité. De plus, il peut accéder à son profil et y apporter des modifications.
Effectuer une demande de séance	Permet à l'enseignant de faire une demande de séance pour des groupes de son choix, ainsi que pour des étudiants spécifiques.
Consulter les fiches, ajouter une fiche	L'enseignant peut attribuer une fiche à chaque étudiant pour une séance qu'il a déjà enseignée, à condition que l'étudiant ait été présent. Il peut également consulter, télécharger ou téléverser les fiches associées à chaque séance.

Tableau III.3 : Description des cas d'utilisations de l'enseignant.

6.3 Etudiant

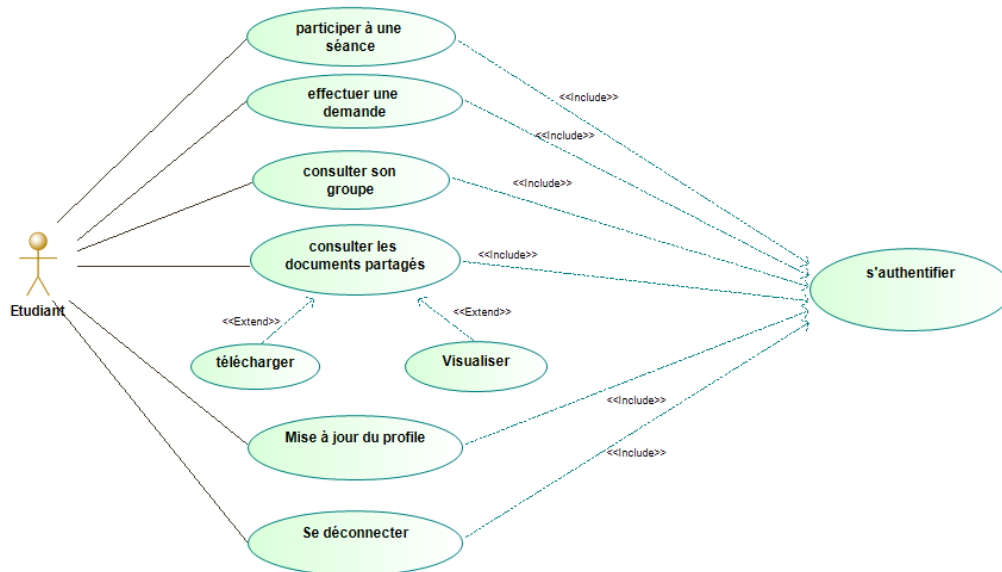


Figure III.6 : Diagramme de cas d'utilisation de l'étudiant.

- **Illustration de diagramme**

Le tableau ci-dessous décrit les différents cas d'utilisation associés au rôle de l'étudiant, avec une brève explication de leur signification.

Cas d'utilisation	Signification
S'authentifier	désigne le processus par lequel l'étudiant prouve son identité pour accéder au système, à l'aide d'un nom d'utilisateur et d'un mot de passe.
Participer à une séance	Permet à un étudiant d'envoyer une notification pour indiquer qu'il est intéressé par cette séance et qu'il souhaite y assister.
Effectuer une demande	Cette fonctionnalité permet à l'étudiant de faire une demande de séance sur un sujet précis, qui sera ensuite acceptée ou refusée par l'enseignant puis l'administrateur.
Consulter les documents partagés	permet à l'étudiant de télécharger les documents partagés par l'enseignant avec son groupe.

Tableau III.4 : Description des cas d'utilisations de l'étudiant

7. Diagramme de séquence

7.1 authentification

L'authentification dans notre application repose sur un nom d'utilisateur et un mot de passe. Elle traite deux scénarios : si les informations saisies sont valides, l'utilisateur peut accéder à l'application. Dans le cas contraire, il doit réessayer jusqu'à ce que les informations soient correctes, cette étape garantit la sécurité de notre application en assurant que seuls les utilisateurs autorisés peuvent accéder aux fonctionnalités et aux données.

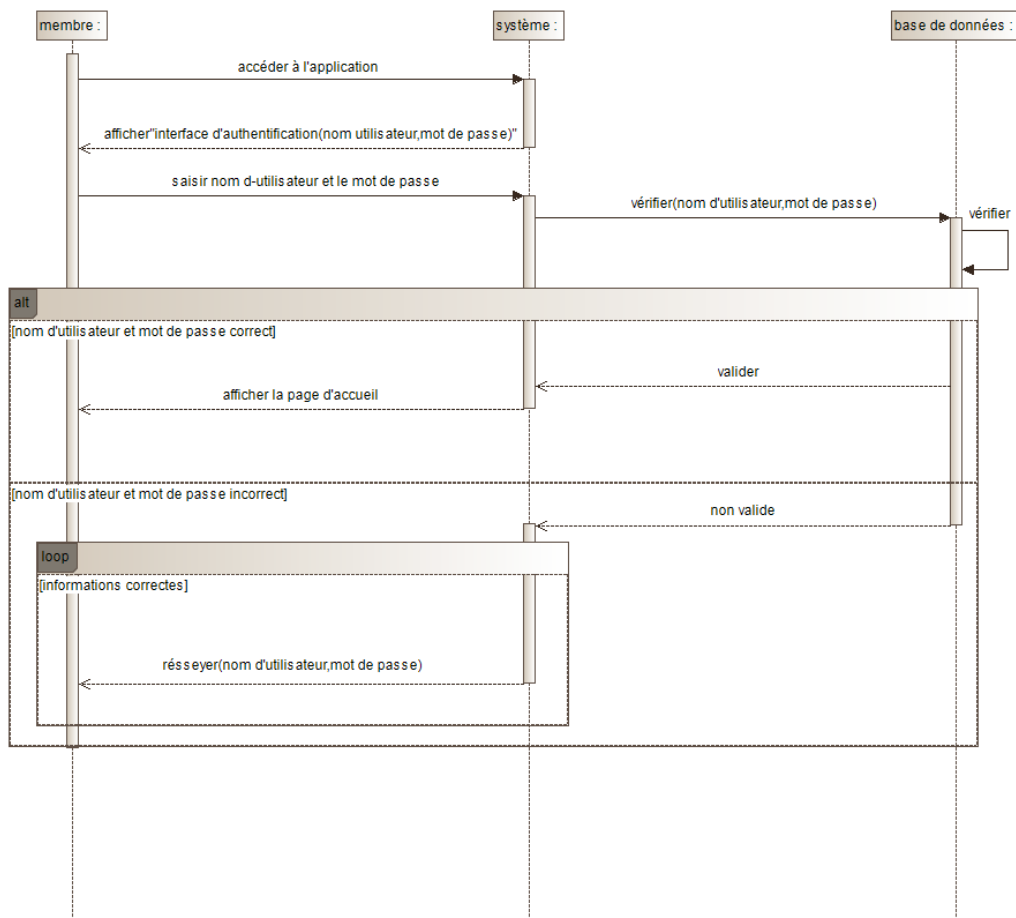


Figure III.7 : Diagramme de séquence d'authentification.

7.2 Demande séance

Les deux acteurs qui peuvent soumettre une demande de séance dans notre application sont les enseignants et les étudiants, en fonction de leurs besoins. Chaque demande est stockée dans une liste jusqu'à ce qu'elle soit traitée.

• cas d'une demande par un enseignant

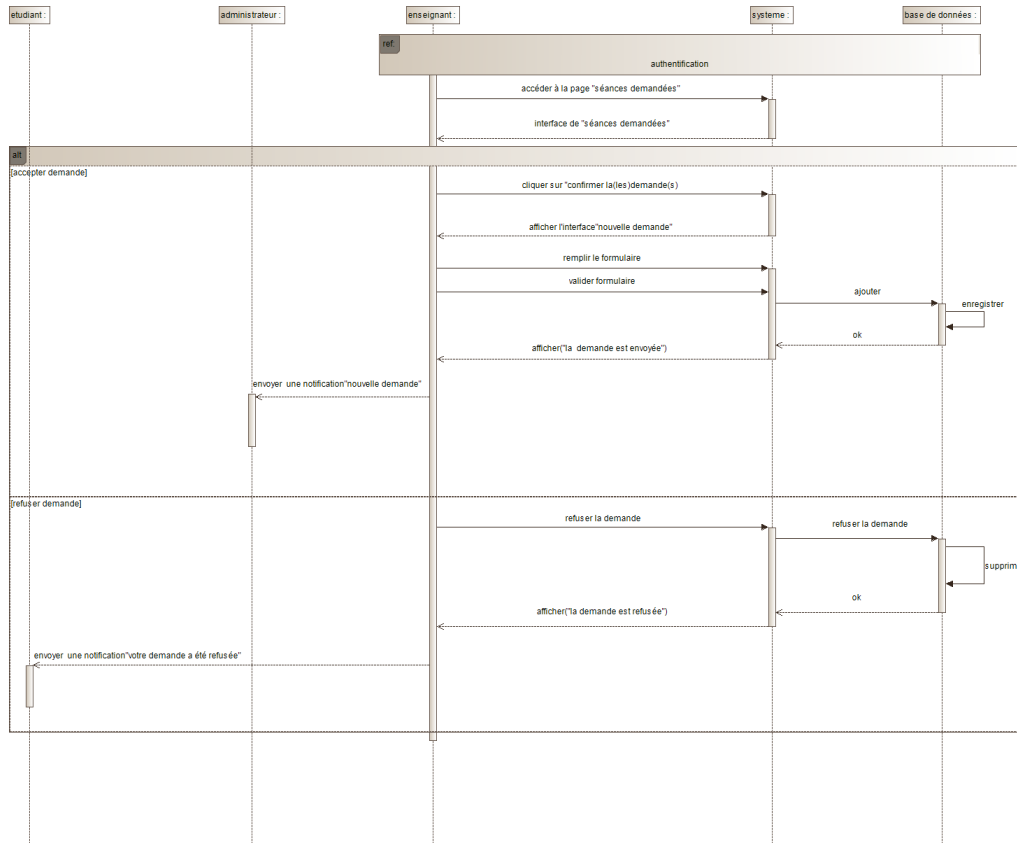


Figure III.8 : Diagramme de séquence :demande d'une séance par un enseignant.

- cas d'une demande par un étudiant

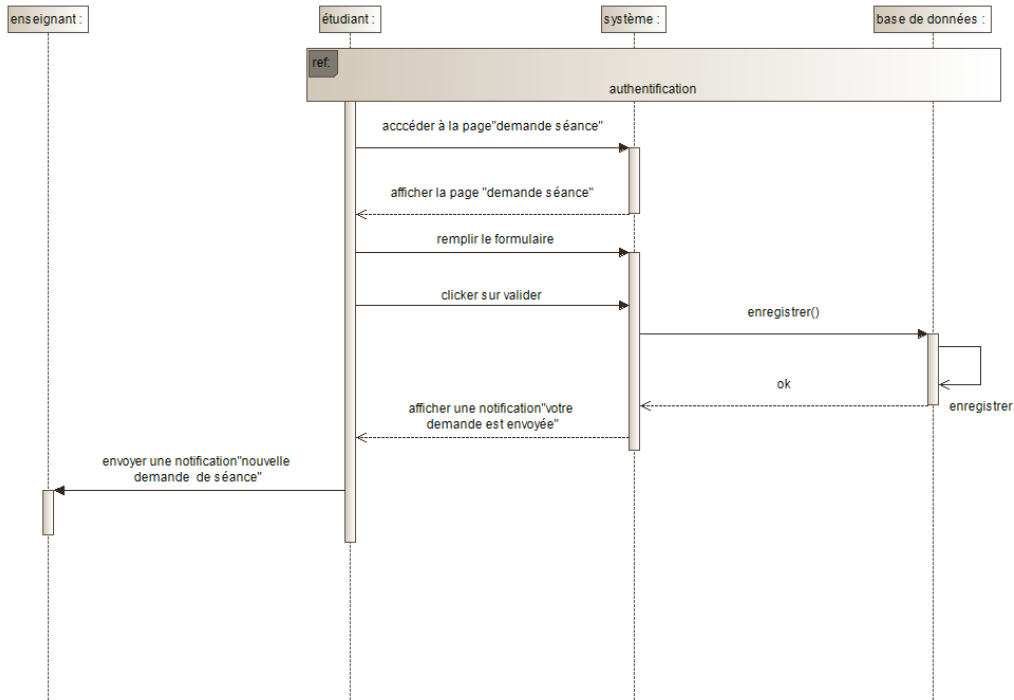


Figure III.9 : Diagramme de séquence : demande d'une séance par un étudiant.

7.3 Traiter une demande de séance

- Administrateur

Les demandes de séances soumises par les enseignants apparaissent comme des notifications sur la page d'accueil de l'administrateur. Celui-ci peut accéder à cette liste pour obtenir plus de détails sur les demandes, tels que le nom complet de l'enseignant, les groupes concernés par la séance demandée, etc. À ce stade, deux scénarios peuvent être envisagés.

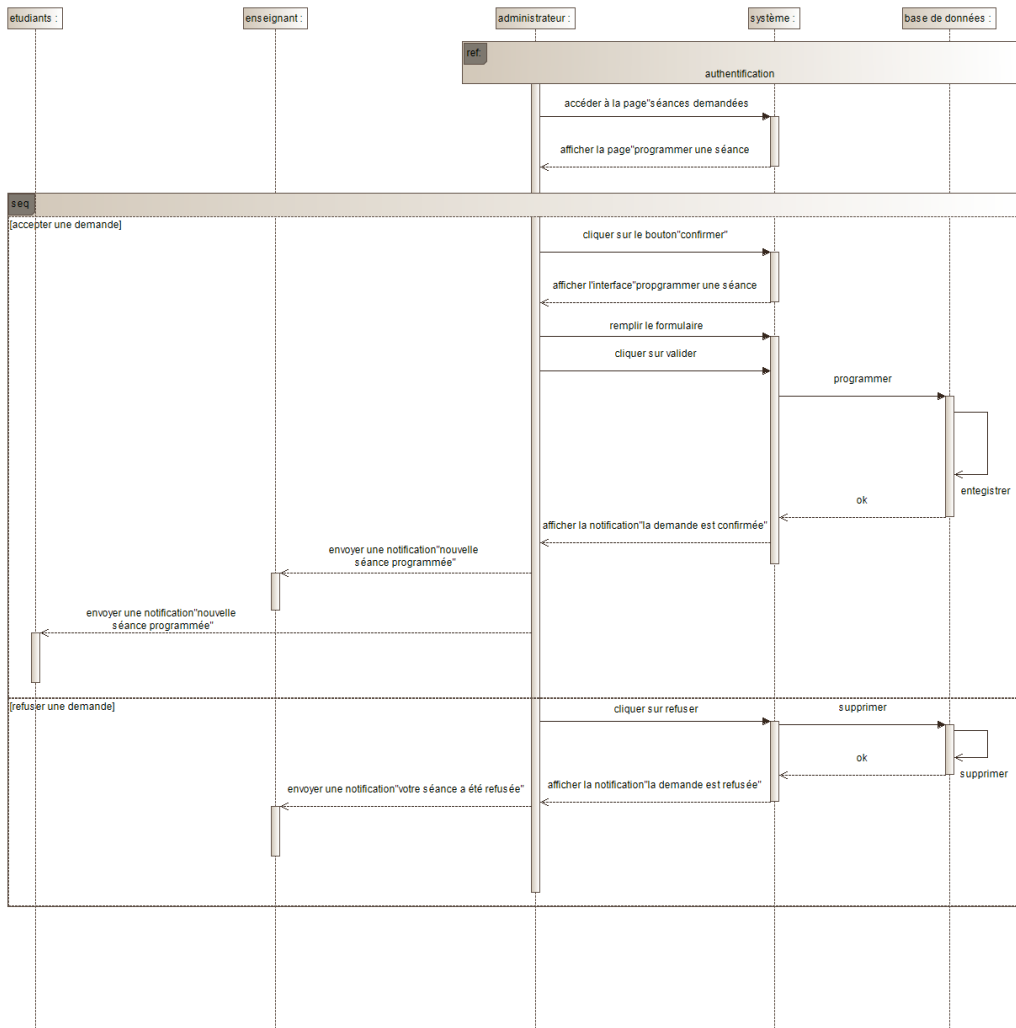


Figure III.10 : Diagramme de Séquence :traitement d'une demande par l'administrateur.

- **Enseignant**

Le processus est le même pour l'enseignant, à la différence près que les demandes de séances sont faites par les étudiants. L'enseignant reçoit des notifications de ces demandes sur sa page d'accueil et peut accéder à la liste pour obtenir des détails supplémentaires, tels que le nom complet de l'étudiant, son groupe, etc.

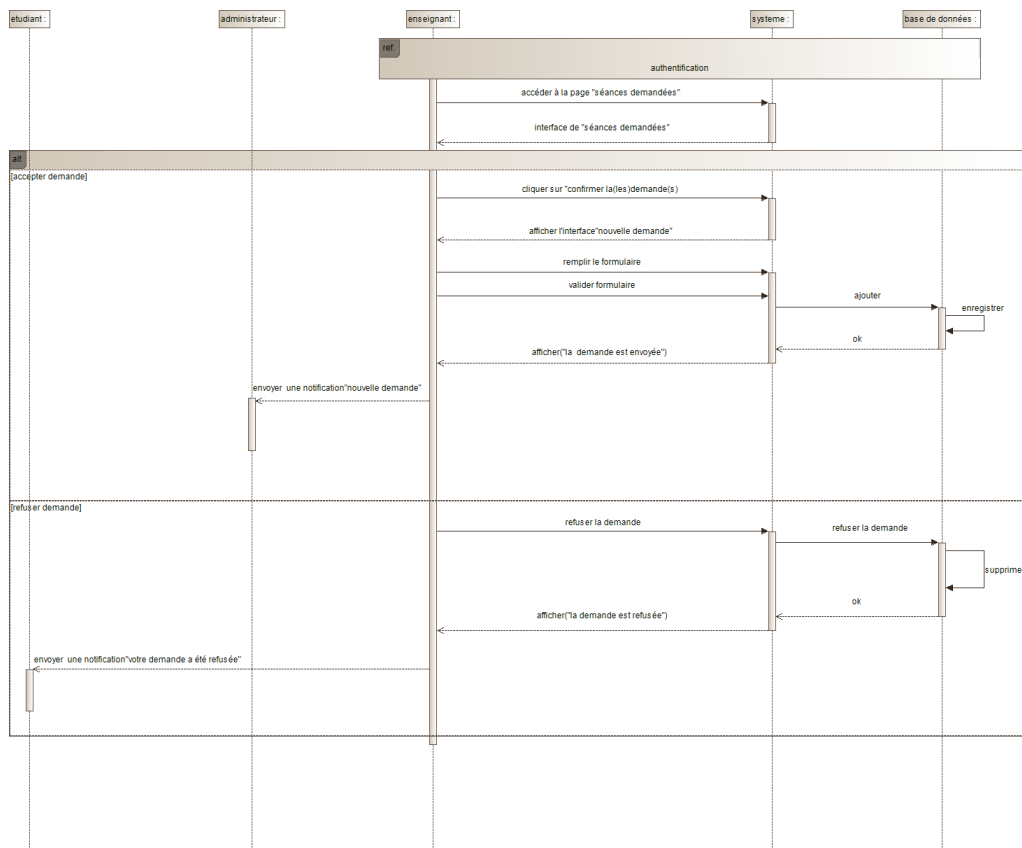


Figure III.11 : Diagramme de Séquence :traitement d'une demande par l'enseignant.

7.4 Ajouter une attestation

Les enseignants qui ont atteint le seuil requis sont automatiquement affichés dans la liste des enseignants éligibles pour recevoir une attestation. L'administrateur attribue ensuite une attestation à chacun, suivant le scénario suivant.

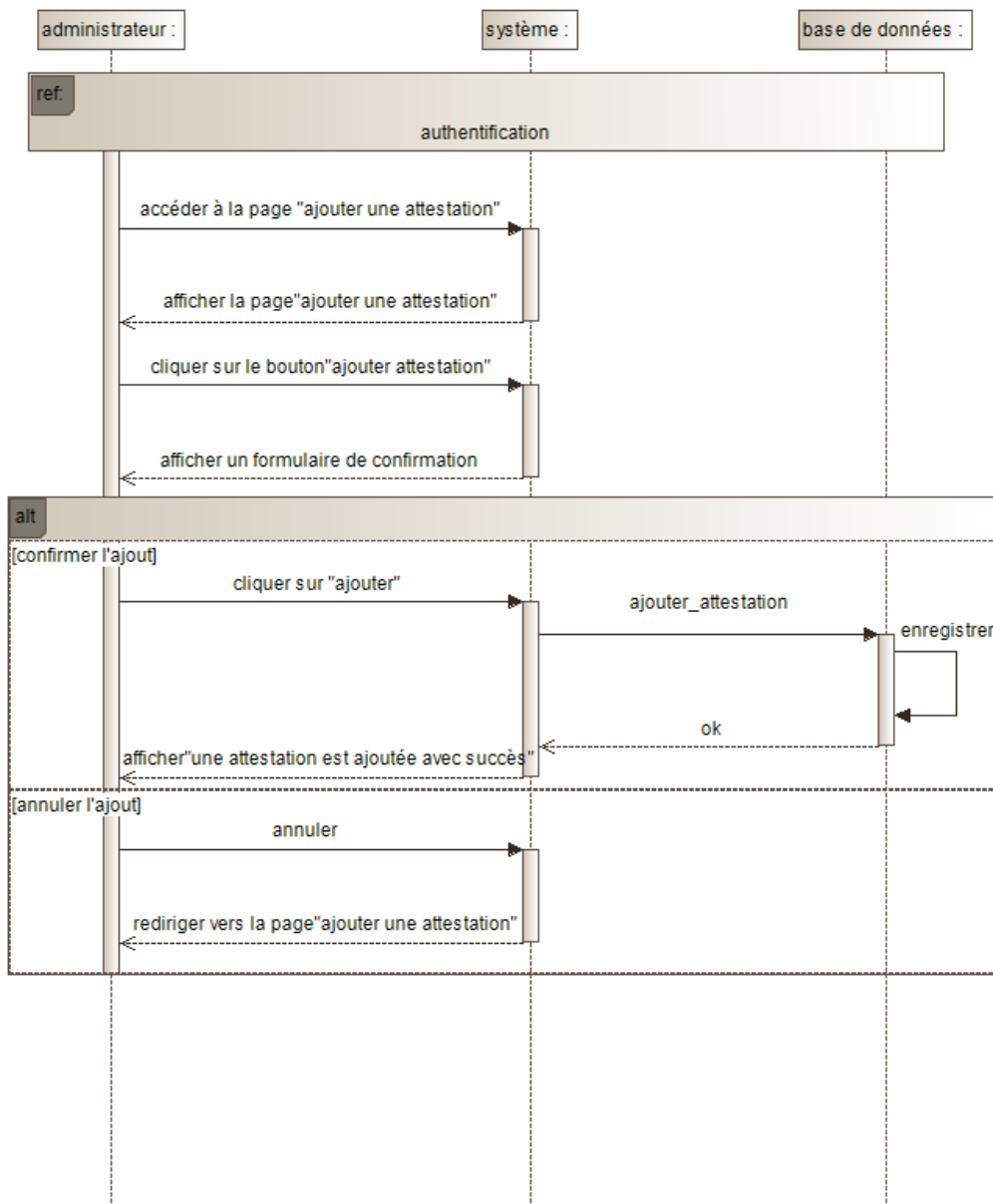


Figure III.12 : Diagramme de séquence :ajouter une attestation.

7.5 Ajouter une entité

Le diagramme ci-dessous illustre le scénario que suit l'administrateur lors de l'ajout d'un nouvel étudiant, enseignant ou groupe.

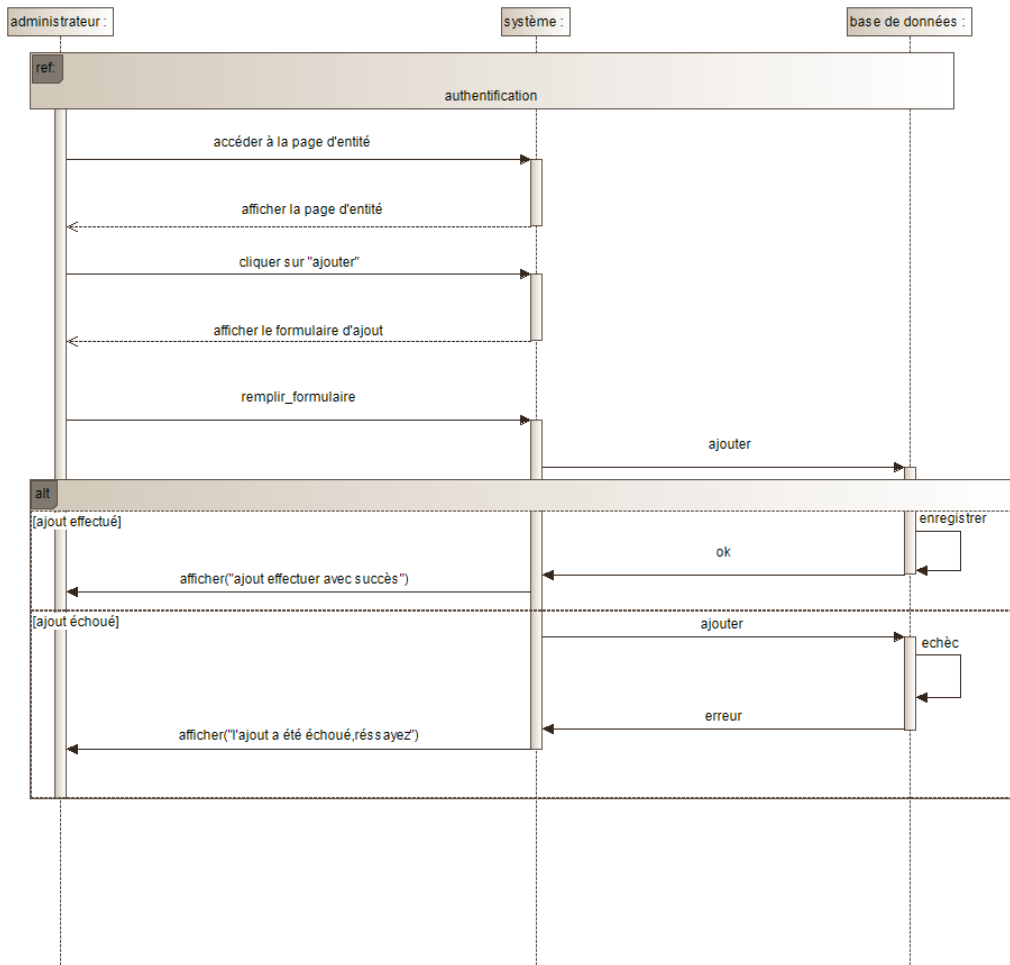


Figure III.13 : Diagramme de séquence : ajouter une nouvelle entité.

7.6 Ajouter une fiche

Le diagramme suivant montre le scénario suivi par l'enseignant pour attribuer une fiche à un étudiant qui a assisté à une séance.

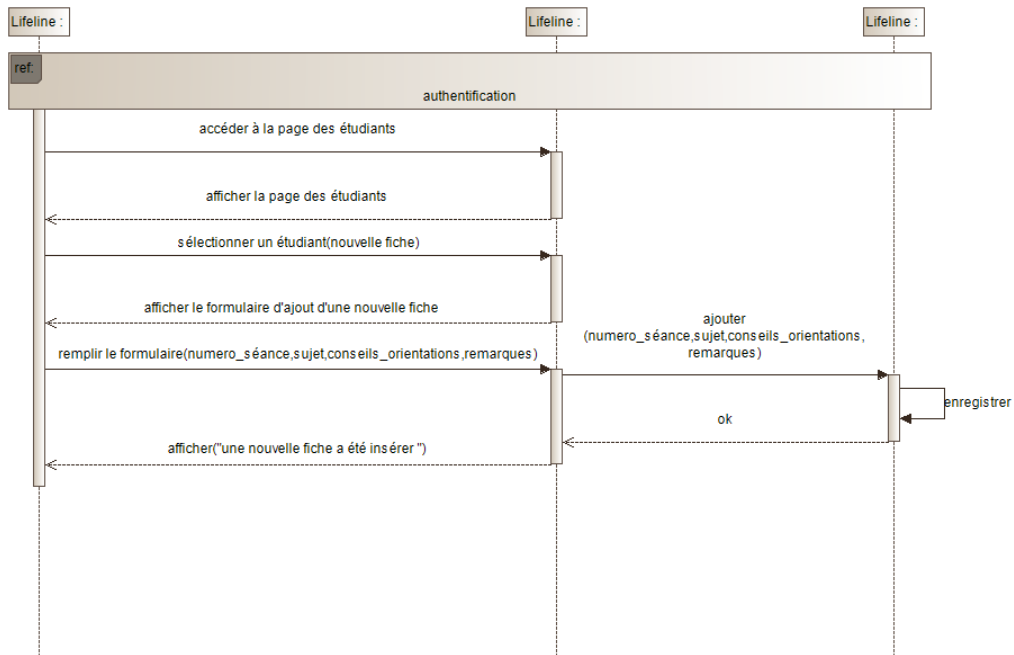


Figure III.14 : Diagramme de séquence : ajouter une fiche.

8. Diagramme de classe

Le diagramme suivant montre la structure de notre système, les relations entre les objets, les attributs et les opérations qui caractérisent chaque classe d'objet.

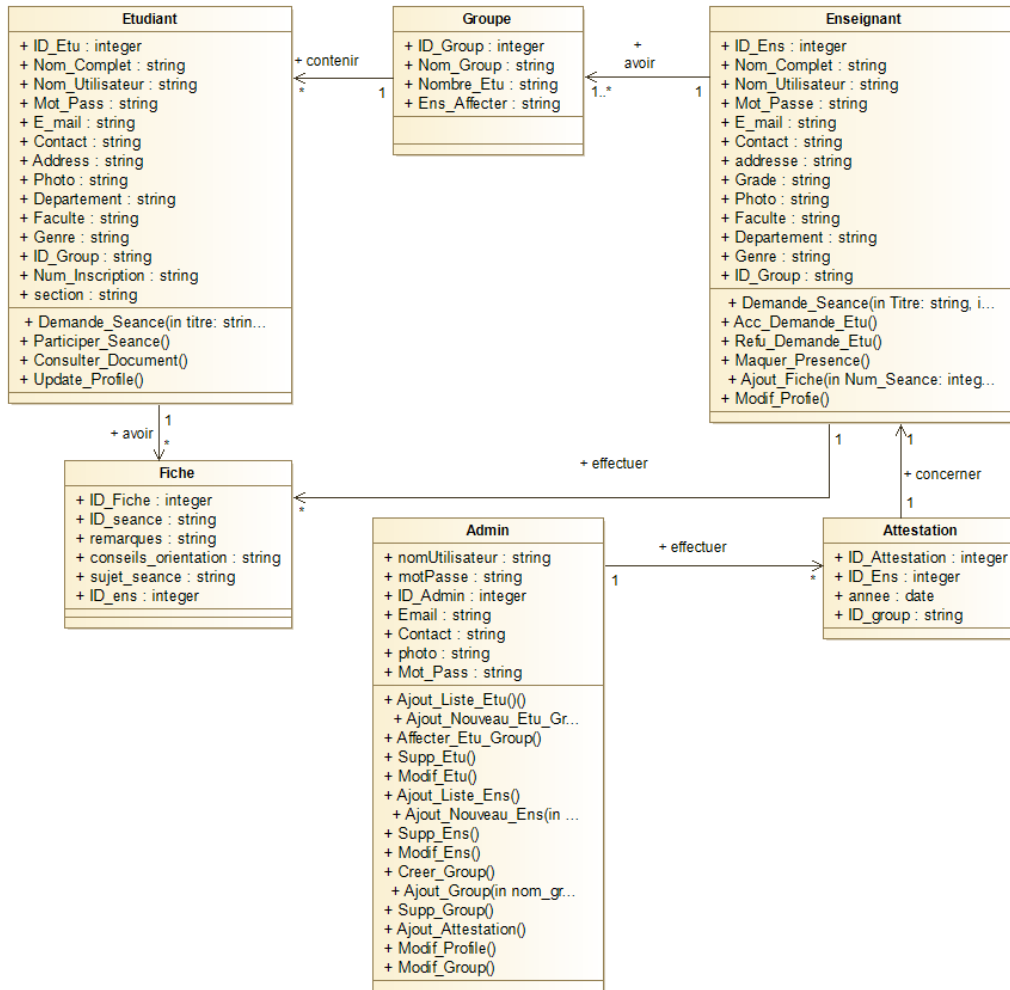


Figure III.15 : Diagramme de classe

9. Conception de la base de données

9.1 Le modèle relationnel

- **Admin**

(ID_admin, Nom_Utilisateur, Mot_Pass, Nom_Complet, Contact, E-Mail, Photo).

- **Enseignant**

(ID_Enseignant, Nom_Utilisateur, Mot_Pass, Nom_Complet, Sexe, Grade, Contact, E_Mail, Adresse, Faculté, Département, Photo, #ID_Groupe).

- **Etudiant**

(ID_Etudiant, Numéro_Inscription, Nom_Utilisateur, Mot_Pass, Nom_Complet, Sexe, Contact, E_Mail, Adresse, Faculté, Département, Section, Photo, #ID_Groupe).

- **Groupe**

(ID_Groupe, Nom_Groupe, Nombre_Etudiants, Faculté, Département, #ID_Enseignant).

- **Fiche**

(ID_Fiche, Sujet_séance, Remarques, Conseils_orientations, #ID_Étudiant, #ID_Séance).

- **Attestation**

(ID_Attestation, Année_Attestation, #ID_Enseignant).

10. Conclusion

La conception de mon application repose sur trois diagrammes principaux. Le diagramme de cas d'utilisation décrit les interactions utilisateur et les fonctionnalités requises. Le diagramme de séquence illustre le flux des opérations entre les composants du système. Le diagramme de classe détaille les structures, avec les relations entre les objets clés. Ces diagrammes fournissent une vision claire de notre application et constituent une base solide pour la phase de développement. Passons maintenant à la mise en œuvre tout en respectant la structure définie.

Chapitre IV

Implémentation

1. Introduction

Dans ce chapitre, nous plongerons dans une exploration détaillée de notre application, mettant en lumière à la fois ses fonctionnalités et les outils de développement qui ont rendu sa création possible. Nous commencerons par les outils de développement utilisés pour donner vie à notre application. Nous passerons en revue les langages de programmation, les frameworks, les bibliothèques et les autres technologies clés qui ont été employés dans le processus de développement. Ensuite, nous présenterons une série de captures d'écran soigneusement sélectionnées, offrant ainsi un aperçu visuel de l'interface utilisateur et des diverses fonctionnalités offertes aux utilisateurs, qu'ils soient administrateurs, enseignants ou étudiants.

2. Outils de développement

2.1 Modélisation

2.1.1 Modelio Open Source 4.1

Modelio est un logiciel open source de qualité qui permet la modélisation de systèmes logiciels et de bases de données. Il offre une large gamme de fonctionnalités pour les architectes d'entreprise, de logiciels et d'infrastructures. Voici quelques-unes des caractéristiques principales :

- **Gestion du temps et réduction des coûts :**
 - Adaptable aux besoins spécifiques, ce qui permet de ne payer que pour les fonctionnalités réellement nécessaires, assurant ainsi un bon retour sur investissement (ROI).

- Prend en charge la génération de code, le reverse engineering et le refactoring rapide de modèles, réduisant le temps de développement et accélérant la mise sur le marché des produits.
- **Maximisation de la flexibilité métier :**
 - Favorise la personnalisation, l’innovation et le développement de nouvelles solutions de modélisation grâce à son écosystème de modules. Les développeurs peuvent créer et partager leurs propres modules, créant ainsi une base solide pour l’extensibilité.
 - Offre une version open source, permettant l’ajout de fonctionnalités supplémentaires ou le développement de modules personnalisés. [34]

2.2 Front-End

Le front-end d’une application est la partie visible et interactive avec laquelle les utilisateurs interagissent directement. Dans cette section, nous présenterons les outils et les technologies que nous avons utilisés pour développer cette interface utilisateur, offrant ainsi une expérience agréable et intuitive aux utilisateurs finaux.

- **Html (HyperText Markup Language)**

Un langage de balisage sert à faciliter la création des documents bien structurés et la mise en page des sites web en utilisant des balises pour organiser les éléments tels que les titres, les paragraphes, les liens, les images, etc. Les premières versions de HTML s’appuyaient sur le SGML (Standard Generalized Markup Language, norme [ISO 8879](#)), mais HTML 5 n’est plus basé sur le [SGML](#) et fonctionne désormais comme une norme autonome.[35]

- **[CSS](#) (Cascading Style Sheet)**

est un langage de feuille de style utilisé pour définir les aspects visuels des

éléments d'une page web selon les préférences du développeur.[46]

- **Les frameworks**

Le framework, ou «cadre de travail», est un ensemble d'outils de haut niveau et de bibliothèques qui facilite la création et l'implémentation des applications, met à la disposition des développeurs des méthodes visant à améliorer la productivité, et permet la réutilisation des codes pour le développement de diverses applications telles que des sites web, des applications mobiles, des jeux, des extensions, etc. Les frameworks sont disponibles dans diverses formes : gratuits ou payants, open-source ou propriétaires. Cette diversité offre aux développeurs une gamme étendue d'options en fonction de leurs besoins et de leurs préférences.[37] est un Framework open source utilisé pour créer des sites web réactifs avec HTML, CSS, et JavaScript, afin d'assurer un fonctionnement optimal sur toutes les tailles d'écran. Les trois fichiers principaux de Bootstrap sont les suivants :

- Bootstrap.css : Fichier CSS pour la mise en page, les grilles, et les composants.
- Bootstrap.js : Fichier JavaScript qui ajoute de l'interactivité aux éléments de l'interface. [38]

nous avons utilisé le template AdminLTE c'est un Thème basé sur Bootstrap pour les tableaux de bord et les interfaces d'administration.

- Librairie : un ensemble de modèles et des fonctions liées à un thème spécifique conçus pour accomplir des tâches spécifiques ou fournir des fonctionnalités liées à un thème particulier. [39]

Parmi les librairies que nous avons utilisé, on trouve les suivantes :

jQuery : Bibliothèque JavaScript pour la manipulation du **DOM**, les animations et les requêtes AJAX.

Morris.js : Pour créer des graphiques (visuels).

Toaster (Toast Master) : Pour les notifications et les messages contextuels.

Wave Effects et Sidebarmenu : Scripts pour des effets visuels et la gestion des menus latéraux.

2.3 Back-End

Le back-end d'une application constitue le moteur invisible qui assure le bon fonctionnement de toutes les opérations en coulisses. Dans cette section, nous explorerons les outils et les technologies que nous avons sélectionnés pour développer cette composante essentielle de l'application, garantissant ainsi sa robustesse et son efficacité.

- **PHP (acronyme récursif de PHP : Hypertext Preprocessor)**

Un langage de programmation largement utilisé pour le développement de sites web. Conçu principalement pour s'exécuter sur des serveurs web, il joue un rôle central dans les interactions entre les navigateurs des utilisateurs et les serveurs hébergeant des applications web, Il est également utilisé côté serveur pour la logique métier et la gestion des bases de données. [40]

- **JavaScript**

un langage de script orienté, dont la syntaxe est basée sur le style de C, il a été évolué indépendamment et a ses propres paradigmes de programmation, y compris un système d'objets basé sur des prototypes plutôt que sur des classes traditionnelles, Il est généralement utilisé côté client pour rendre les interfaces utilisateur interactives et dynamiques.[41]

- **Serveur XAMPP**

est un acronyme créé par les "Apache Friends" où le "X" signifie "Cross-Platform" (multiplateformes), le "A" signifie "Apache", le "M" signifie "MySQL", et les deux "P" représentent respectivement "PHP" et "Perl". C'est un serveur web multiplateformes qui permet le développement et le test de code sur un hôte ou un serveur local. Il comprend le serveur HTTP Apache, Maria DB, ainsi que des interprètes pour PHP et Perl, entre autres composants.[42]

- **MySQL**

un système de gestion de bases de données relationnelles (SGBDR) open source, distribué, développé et soutenu par Oracle Corporation. Il stocke les données sous forme de tableaux et utilise le langage de requête structuré (SQL) pour accéder aux données. MySQL permet de réaliser diverses opérations comme les Jointures, les Indexes, les vues, Importation/Exportation des données dans différents formats, comme CSV ou SQL, etc.[43]

- **PhpMyAdmin**

est une interface web open source qui gère l'administration de MySQL, il permet la création, la modification et la suppression des bases de données et des tables, l'insertion, la modification et la suppression des enregistrements de données, et l'importation et l'exportation des bases de données entières, parmi de nombreuses autres tâches administratives. [44]

- **Apache Web Server**

est un projet collaboratif open source visant à créer un serveur HTTP de qualité professionnelle. Le serveur Apache est un produit supervisé conjointement, géré par un groupe de bénévoles appelé le "Groupe Apache". Ce groupe est renforcé par la communauté open source, qui a apporté une contribution inestimable en termes de temps et d'effort au développement du

serveur. [45]

2.4 Environnement de Développement Intégré (IDE)

2.5 Visual Studio Code

Éditeur de code source puissant disponible pour Windows, MacOS et Linux, Il inclut un support intégré pour JavaScript, Type Script et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages et environnements d'exécution (comme C++, C sharp, Java, Python, PHP, Go, .NET). Visual Studio Code prend en charge des fonctionnalités telles que la coloration syntaxique, la correspondance des crochets, l'auto-indentation, les snippets, et des raccourcis clavier intuitifs, il inclut aussi des outils de complétion de code IntelliSense, une navigation sémantique, et des options de refactoring. Le débogage interactif permet de parcourir le code source, d'inspecter les variables et de visualiser les piles d'appels. [46]

2.6 Navigateur Internet

Un navigateur web est un logiciel applicatif qui permet d'explorer le www (World Wide Web). Il agit comme un compilateur, interprétant le HTML utilisé pour concevoir des pages web. Ces pages peuvent inclure du texte, des liens, des images, ainsi que d'autres éléments comme des feuilles de style et des fonctions JavaScript. Google Chrome, Microsoft Edge, et Mozilla Firefox sont des exemples de navigateurs web. [47]

3. Description de l'application

La section suivante procède à l'examen des diverses interfaces graphiques de notre application, accompagnée d'une brève description pour chacune. Cette démarche vise à identifier les composants visuels caractéristiques, à mettre en évidence les fonctionnalités disponibles ainsi qu'à clarifier les objectifs spécifiques de chaque interface.

3.1 Interface d'authentification

Il est évident que l'étape initiale essentielle pour accéder à toute application réside dans l'authentification, qui revêt une importance cruciale en tant que pilier fondamental de la sécurité. En effet, l'authentification garantit la protection des données, des systèmes, des réseaux, ainsi que des applications web, en prévenant les attaques susceptibles d'engendrer des pertes de données ou des interruptions des activités. Dans mon système, la sécurité est assurée par l'utilisation d'un nom d'utilisateur et d'un mot de passe uniques, cryptés à l'aide d'un algorithme de hachage spécifique. En démarrant l'application, l'utilisateur sera dirigé vers l'écran suivant :

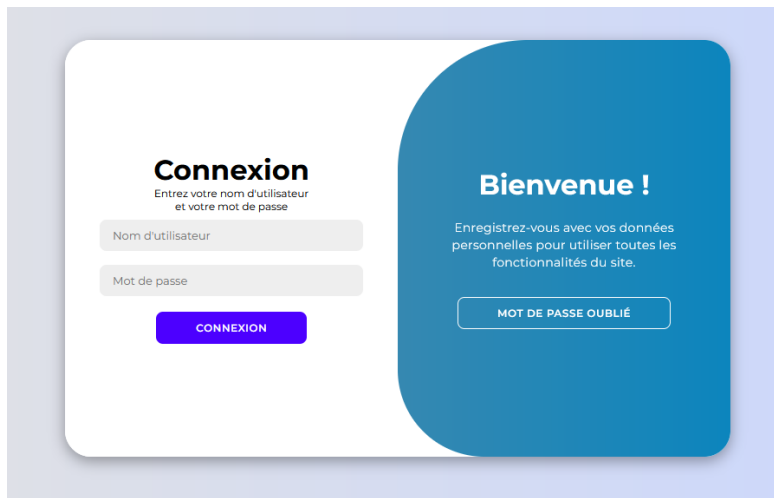


Figure IV.1 : Interface d'authentification.

Dans le cas où l'utilisateur oublie son mot de passe, il doit cliquer sur "Mot de passe oublié". Il sera alors redirigé vers cette page où il peut entrer son adresse e-mail enregistrée. Après avoir soumis cette information, l'utilisateur recevra un e-mail contenant des instructions pour réinitialiser son mot de passe, garantissant ainsi un accès sécurisé et sans interruption à l'application. Cette fonctionnalité assure une récupération rapide et facile des comptes utilisateur, améliorant ainsi l'expérience globale des utilisateurs.



Figure IV.2 : Interface de réinitialisation du mot de passe.

Après une procédure de connexion réussie, les utilisateurs (administrateur, enseignant, étudiant) sont dirigés chacun vers un écran principal simple, où ils ont accès à des fonctionnalités spécifiques en fonction de leurs rôles et de leurs privilèges respectifs.

3.2 Administrateur

3.2.1 Page d'accueil

Cette page présente diverses informations sur le taux de présence des enseignants et des groupes, ainsi que le nombre des demandes de séances s'il y en a, sous forme de notifications (consulter la figure suivante).

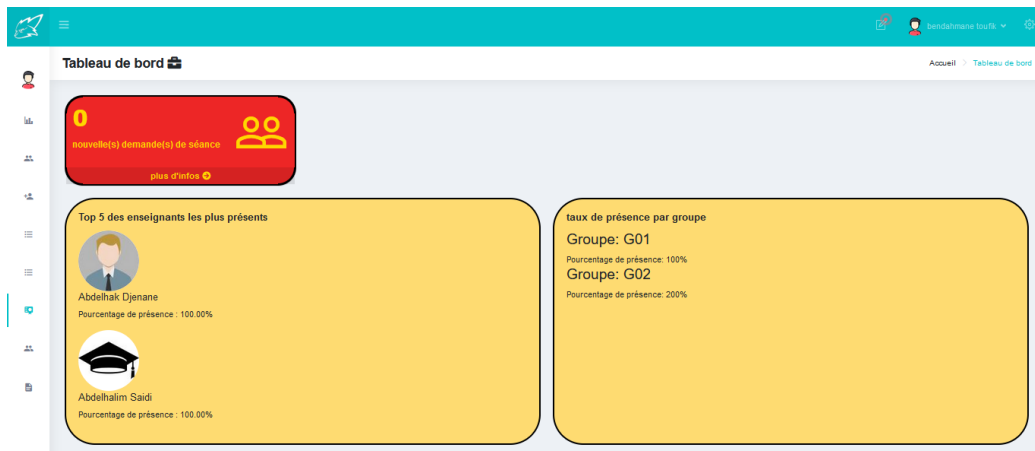


Figure IV.3 : Page d'accueil du compte administrateur.

3.2.2 Gestion des demandes

L'administrateur peut consulter les différentes demandes afin de les valider via le formulaire suivant :

The screenshot shows the "Confirmer une Demande" form with the following fields and buttons:

- Header:** "Confirmer une Demande" and navigation links "Home" and "Retour".
- User:** "Abdelhalim Saidi".
- Identifiant de la Séance:** Input field containing "5".
- Titre de la Séance:** Input field containing "les listes chaînées".
- Salle:** Empty input field.
- Date et Heure de la Séance:** Date picker showing "mm / dd / yyyy, --:-- --".
- Buttons:** "Réinitialiser" (red) and "Valider" (green).

Figure IV.4 : Formulaire de confirmation des demandes.

3.2.3 Opérations d'ajout

Cette tâche englobe l'ajout d'un nouveau groupe, d'un nouvel enseignant, d'un nouvel étudiant, ou encore la gestion d'une liste complète d'enseignants ou d'étudiants.

Figure IV.5 : formulaire 01 d'inscription d'un nouvel enseignant.

Figure IV.6 : formulaire 02 d'inscription d'un nouvel enseignant.

Figure IV.7 : formulaire d'inscription d'un nouveau groupe.

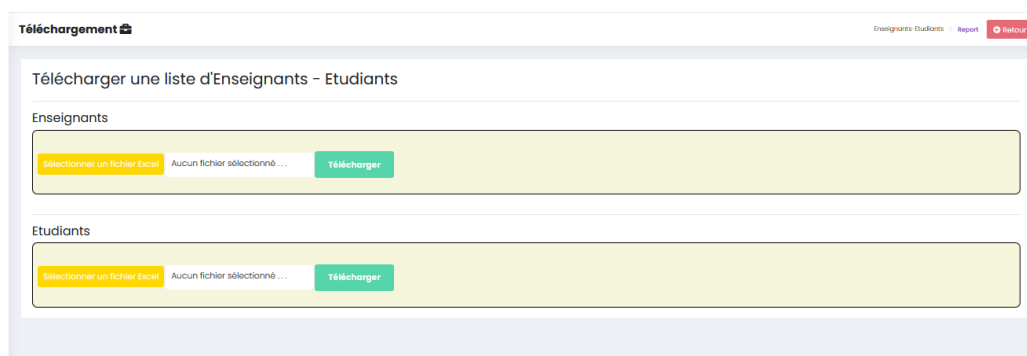


Figure IV.8 : Téléchargement d'une liste d'enseignant-étudiant.

Pour ajouter des attestations, l'administrateur doit d'abord consulter la liste des enseignants autorisés à recevoir une attestation. Ensuite, il clique sur le bouton d'ajout, ce qui fait apparaître un formulaire de confirmation pour finaliser l'ajout.

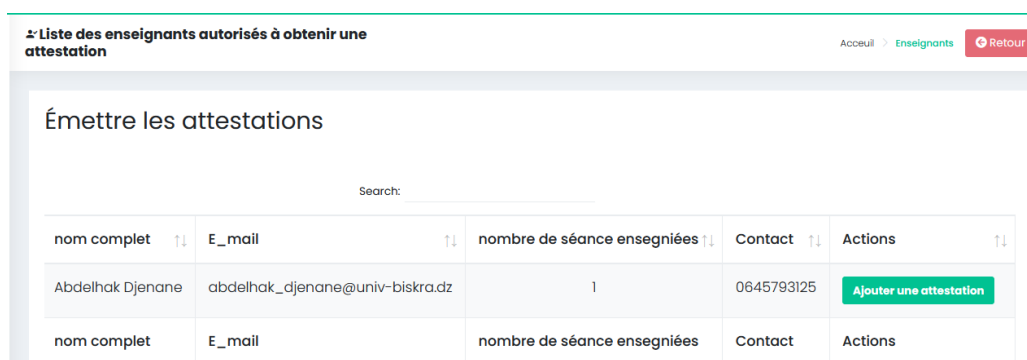


Figure IV.9 : Ajouter une attestation.

Liste Des Attestations Accueil > Attestations > Retour

Liste Des Attestations

cher 10 éléments Rechercher :

#	Identifiant de l'Attestation	Identifiant de l'Enseignant	Nom Complet	Grade	Année	Actions
1	1	2	Abdelhak Djenane	doctorant	2024/2025	Visualiser le PDF Télécharger le PDF Upload PDF
2	2	3	Abdelhalim Saidi	doctorant	2024/2025	Visualiser le PDF Télécharger le PDF Upload PDF

charge de l'élément 1 à 2 sur 2 éléments Précédent | Suivant

Figure IV.10 : Consulter les attestations.

3.3 Enseignant

3.3.1 Page d'accueil

Cette page joue un rôle central en consolidant différentes notifications importantes. En outre, elle met en lumière les étudiants qui se distinguent par leur assiduité, offrant ainsi une reconnaissance de leur engagement. Ce faisant, elle contribue à créer un environnement qui valorise les efforts et l'implication des étudiants dans leur formation. De plus, un graphique est présenté pour visualiser de manière claire et concise le pourcentage des séances assistées et non assistées par l'enseignant. Cette représentation visuelle permet à l'enseignant de mieux évaluer la participation de ses étudiants et de prendre des mesures appropriées pour soutenir ceux qui pourraient avoir besoin d'un accompagnement supplémentaire. En résumé, cette page offre une combinaison d'informations essentielles et de fonctionnalités visuelles qui favorisent la transparence, la reconnaissance des efforts et la prise de décision éclairée dans le cadre de l'enseignement et de l'apprentissage.

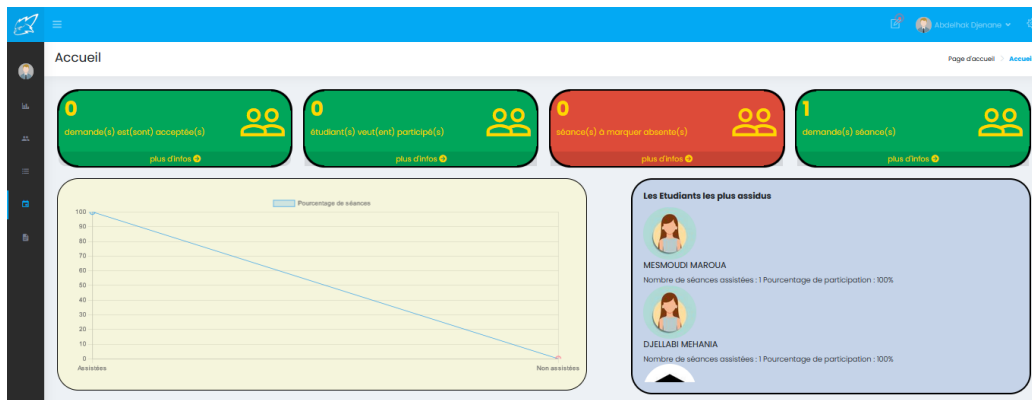


Figure IV.11 : Page d'accueil du compte enseignant.

3.3.2 Gestion des séances

3.3.3 Demande séance

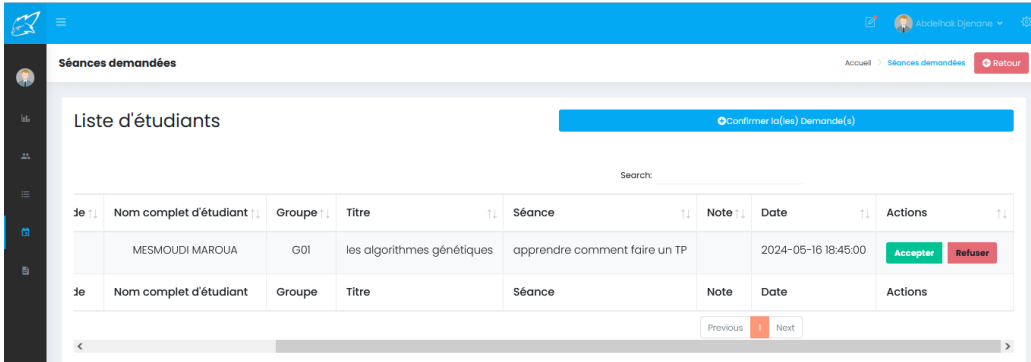
En plus des séances programmées par l'administrateur, l'enseignant a la possibilité de solliciter des séances supplémentaires conformément à ses besoins. Ces demandes sont effectuées au moyen du formulaire ci-dessous :

Figure IV.12 : Formulaire de demande des supplémentaires.

Remarque

- Un étudiant a également la possibilité de soumettre une demande selon ses

besoins. L'enseignant responsable du groupe auquel cet étudiant est affilié reçoit cette demande sous forme de notification et la traite en utilisant le formulaire ci-dessous :

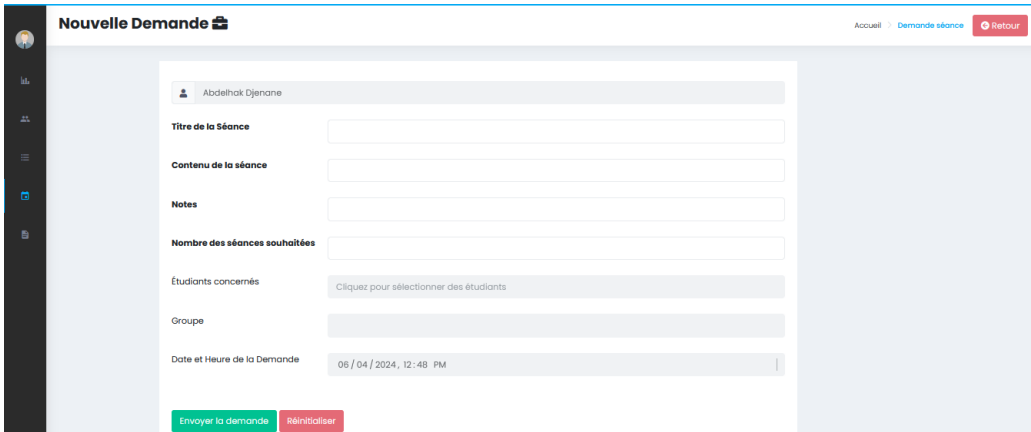


The screenshot shows a web interface titled "Séances demandées". At the top, there is a navigation bar with "Accueil", "Séances demandées", and a "Retour" button. Below the navigation bar, there is a section titled "Liste d'étudiants" with a blue button labeled "Confirmer la(s) Demande(s)". A search bar is present above a table. The table has columns for "Id", "Nom complet d'étudiant", "Groupe", "Titre", "Séance", "Note", "Date", and "Actions". One row is visible with the following data: "MESMOUDI MAROUA", "G01", "les algorithmes génétiques", "apprendre comment faire un TP", an empty "Note" field, and "2024-05-16 18:45:00". The "Actions" column for this row contains "Accepter" and "Refuser" buttons. At the bottom of the table, there are "Previous" and "Next" navigation buttons.

Id	Nom complet d'étudiant	Groupe	Titre	Séance	Note	Date	Actions
	MESMOUDI MAROUA	G01	les algorithmes génétiques	apprendre comment faire un TP		2024-05-16 18:45:00	Accepter Refuser

Figure IV.13 : Liste des demandes des séances supplémentaires.

Dans le cas où plusieurs demandes sont soumises, le bouton "Confirmer les demandes" offre une fonctionnalité de sélection multiple, permettant ainsi de traiter simultanément plusieurs demandes en une seule action (voir la figure suivante) :



The screenshot shows a web interface titled "Nouvelle Demande". At the top, there is a navigation bar with "Accueil", "Demande séance", and a "Retour" button. Below the navigation bar, there is a form for creating a new request. The form includes fields for "Titre de la séance", "Contenu de la séance", "Notes", "Nombre des séances souhaitées", "Étudiants concernés" (with a button "Cliquez pour sélectionner des étudiants"), "Groupe", and "Date et Heure de la Demande" (showing "06 / 04 / 2024, 12:48 PM"). At the bottom of the form, there are two buttons: "Envoyer la demande" and "Annuler".

Figure IV.14 : Formulaire de demande d'une séance supplémentaire.

Cas de refus

- Comme l'enseignant a la possibilité de valider une demande, il peut également la refuser en utilisant la confirmation suivante :

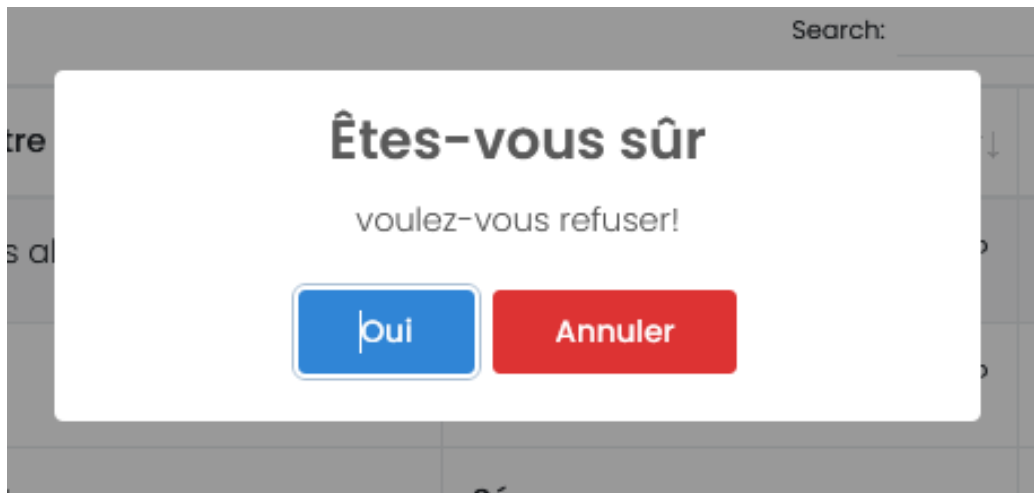


Figure IV.15 : Refuser une demande.

3.3.4 Gestion des fiches

L'attribution d'une fiche à un étudiant pour une séance d'enseignement se fait par le biais du remplissage d'un formulaire après avoir sélectionné l'étudiant spécifique. Les huit premiers champs du formulaire sont automatiquement renseignés, facilitant ainsi le processus. De plus, les trois premiers champs sont pré-remplis dès que le numéro de la séance est sélectionné, ce qui permet de gagner du temps et de réduire les erreurs potentielles. Cette méthode assure une attribution précise et efficace des fiches, simplifiant la gestion administrative et améliorant la fluidité des opérations pour les enseignants et les administrateurs.

Numéro de la séance: Choisissez le numéro de la séance

Date: [input field]

Heure: [input field]

Lieu: [input field]

Nom complet d'enseignant: Abdelhak Djernane

Identifiant d'étudiant: 5

Nom complet d'étudiant: MESMOUDI MAROUA

Identifiant du groupe: 1

Nom du groupe: G01

Sujet de la séance: Ecrivez le sujet ...

Conseils et orientations du professeur: Ecrivez vos Conseils et orientations ...

Remarques: Ecrivez vos remarques ...

Ajouter Réinitialiser

Figure IV.16 : Formulaire pour remplir une fiche de tutorat.

Liste Des Fiches

Home > Fiches Retour

#	Identifiant de la Séance	Date de la Séance	Identifiant de la Fiche	Actions
1	1	2024-05-15	1	Visualiser le PDF Télécharger le PDF Upload PDF
#	Identifiant de la Séance	Date de la Séance	Identifiant de la Fiche	Actions

Figure IV.17 : Historique des fiches pour un étudiant.

3.4 Etudiant

3.4.1 Page d'accueil

Cette page regroupe différentes notifications liées aux séances prévues. Cela englobe à la fois les séances régulièrement prévues et celles organisées à la

demande de l'étudiant. Il est pris en compte chaque demande de séance supplémentaire, avec une notification précise indiquant si la demande a été acceptée ou refusée, ce qui permet à l'étudiant de rester informé en temps réel de l'avancement de ses demandes. De plus, le graphique présenté sur la page montre le pourcentage des séances assistées et non assistées par l'étudiant. Ce graphique permet de visualiser de manière claire et précise la participation de l'étudiant, ce qui facilite le suivi de son implication et de sa régularité lors des séances prévues. Cette caractéristique revêt une grande importance pour les étudiants qui souhaitent améliorer leur assiduité ainsi que pour les enseignants.

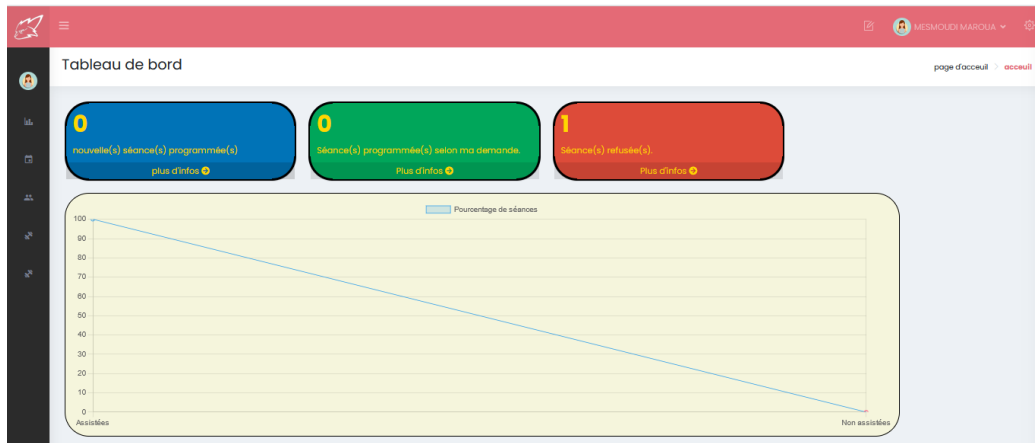


Figure IV.18 : Page d'accueil du compte étudiant.

3.4.2 Consultation

L'étudiant a la possibilité de consulter diverses pages contenant des informations variées, telles que les membres de son groupe, l'historique des séances, les documents partagés par l'enseignant de ce groupe, les statistiques, le profil, etc. (voir les Figures ci-dessous).

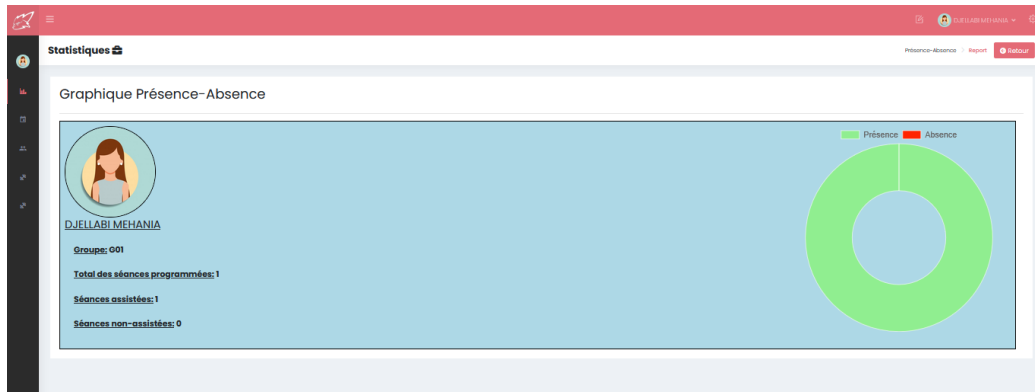


Figure IV.20 : Les statistiques d'un étudiant.

The screenshot shows a web interface for a student's profile information. The page title is "Informations sur le profil". On the left, there is a profile card for "DJELLABI MEHANIA" with the role "Étudiant". On the right, there is a form with the following fields:

- Prénom**: DJELLABI MEHANIA
- nom complet**: DJELLABI MEHANIA
- nom d'utilisateur**: djellabimehania
- mot de passe**: [masked]
- E-mail**: djellabimehania@gmail.com
- Address**: [empty]
- Contact**: 079854912

There is an "Update Profile" button at the bottom of the form.

Figure IV.19 : profile un étudiant.

Remarque

- Cette page permet aux étudiants d'apporter des modifications concernant leurs informations personnelles.

#	Faculté	departement	nom complet	groupe	Email	Contact	#
1	مجال العلوم الاقتصادية، الصبير والعلوم التجارية	كلية العلوم الاقتصادية والتجارية وعلوم الصبير	MESMOUDI MAROUA	1	zsdfg@gmailcom		
2	مجال العلوم الاقتصادية، الصبير والعلوم التجارية	كلية العلوم الاقتصادية والتجارية وعلوم الصبير	DJELLABI MEHANIA	1	djellabimehania@gmailcom	679854612	
3	مجال العلوم الاقتصادية، الصبير والعلوم التجارية	كلية العلوم الاقتصادية والتجارية وعلوم الصبير	BERROUK BESMA	1	zsdfg@gmailcom		
4	مجال العلوم الاقتصادية، الصبير والعلوم التجارية	كلية العلوم الاقتصادية والتجارية وعلوم الصبير	TEDJANI MOUHAMED AKRAM	1	zsdfg@gmailcom		
5	مجال العلوم الاقتصادية، الصبير والعلوم التجارية	كلية العلوم الاقتصادية والتجارية وعلوم الصبير	LASTAB KHAOULA	1	zsdfg@gmailcom		
6	مجال العلوم الاقتصادية، الصبير والعلوم التجارية	كلية العلوم الاقتصادية والتجارية وعلوم الصبير	ZOUGGAD NABILA	1	zsdfg@gmailcom		
#	Faculté	departement	nom complet	groupe	Email	Contact	#

Figure IV.21 : Liste des étudiants d’un groupe.

4. Conclusion

Dans ce chapitre, nous plongerons dans une exploration détaillée de notre application mettant en lumière à la fois ses fonctionnalités et les outils de développement qui ont rendu sa création possible, Nous commencerons par les outils de développement utilisés pour donner vie à notre application Nous passerons en revue les langages de programmation, , les Framework, les bibliothèques et les autres technologies clés qui ont été employés dans le processus de développement Ensuite, nous explorerons une série de captures d’écran soigneusement sélectionnées, offrant ainsi un aperçu visuel de l’interface utilisateur et des diverses fonctionnalités offertes aux utilisateurs, qu’ils soient administrateur, enseignants ou étudiants.

Conclusion générale

Conclusion générale

Tout au long de la période de réalisation de notre projet de fin d'études, axé sur la numérisation des formulaires administratifs dans le cadre de la nouvelle tâche intitulée par le tutorat, adoptée par la faculté des sciences économiques, commerciales et de gestion, nous avons tenté de répondre à notre problématique, qui englobe plusieurs défis : les défis de la communication entre les tuteurs et les étudiants, les défis de l'organisation et de la planification des séances, les défis de l'apprentissage, les défis de l'évaluation des étudiants. Pour répondre à cette problématique nous avons structuré notre travail en quatre chapitres distincts, chacun dédié à l'exploration approfondie d'un aspect spécifique de notre projet.

Dans le premier chapitre, nous avons fourni une analyse approfondie du domaine du tutorat afin de mieux comprendre cette tâche et son impact sur l'enseignement universitaire et le système en général. Nous avons examiné les aspects de l'accompagnement touchés par le tutorat, les rôles des différents membres impliqués dans cette tâche, le principe de son fonctionnement ainsi que les formulaires nécessaires à sa mise en œuvre. En conclusion de ce chapitre, nous avons soulevé une problématique, présenté notre mission, et résumé nos observations dans une conclusion analytique.

Dans le deuxième chapitre, nous avons introduit les applications web, en commençant par un bref historique du Web et en définissant les applications web ainsi que leurs différents types. Nous nous sommes concentrés sur le type que nous avons choisi d'adopter et ses caractéristiques distinctives. En outre, nous avons exploré l'architecture des applications web, en mettant en lumière leurs composants et leur fonctionnement. Nous avons aussi souligné les avantages des applications web, tout en expliquant les solutions qu'elles apportent pour traiter notre

problématique spécifique. Pour conclure ce chapitre, nous avons présenté quelques travaux connexes et enfin une conclusion.

Pour le troisième chapitre, qui se concentre sur la conception de notre application, nous avons abordé la méthode de développement utilisée, qui est la méthode en cascade. De plus, nous avons donné en détail l'architecture utilisée pour la réalisation de notre application. Nous avons utilisé les outils offerts par l'UML (Unified Modeling Language) pour établir divers diagrammes à l'aide du logiciel Modelio. Ces diagrammes, tels que les diagrammes de cas d'utilisation, les diagrammes de classes et les diagrammes de séquence, ont pour objectif de faciliter la compréhension du système et de représenter les différents scénarios de manière visuelle.

Le quatrième chapitre a été consacré à la présentation des outils de développement employés pour la réalisation de notre application, ainsi qu'à l'exposition des différentes captures d'écran des interfaces de notre système.

Cette recherche nous a offert l'opportunité d'approfondir nos compétences théoriques et techniques par la réalisation d'une application web qui nous a permis d'atteindre les résultats suivants :

- Minimiser la quantité des documents papiers par la numérisation des formulaires administratifs tels que : le Certificat d'accompagnement, Registre de présence des tuteurs, Registre de présence des tutorés, les Cartes d'accompagnements des étudiants.
- Surveiller la régularité et l'assiduité : Suivre les présences et absences des étudiants et des enseignants permet d'identifier les tendances et les problèmes potentiels, notamment en cas de taux élevés d'absentéisme.
- Créer un environnement favorable au partage des connaissances et des documents.

-
- Faciliter la planification des séances : Ce système permet aux enseignants et aux étudiants de soumettre facilement des demandes de séances supplémentaires selon leurs besoins. Ces demandes sont traitées rapidement via un formulaire pré-rempli, ne nécessitant que la spécification de la date pour la validation. En cas de refus, une réponse peut être donnée en un seul clic.
 - Faciliter la gestion des absences : L'enseignant est chargé de sélectionner uniquement le numéro de la séance, puis de cocher les cases des étudiants présents.

Malgré les avancées significatives réalisées dans ce projet, le système pourrait être complété et approfondi par l'ajout des nouvelles fonctionnalités à savoir :

- concevoir un algorithme de deep learning capable d'analyser et de comprendre les défis spécifiques auxquels sont confrontés les étudiants. L'objectif ultime est de fournir des solutions personnalisées et des recommandations adaptées à chaque individu, afin d'améliorer leur expérience d'apprentissage et de favoriser leur réussite académique..
- L'implémentation de la reconnaissance vocale des enseignants pourrait optimiser davantage le processus de gestion des séances et des présences.
- La gestion automatique des salles du tutorat.

Références

- [1] المرسوم التنفيذي رقم 03-09 المؤرخ في 6 محرم 1430 الموافق ل 03 يناير 2009. الجريدة الرسمية - العدد 01
- [2] هاشمي عبايسة، وهيبة غربي، "المرافقة البيداغوجية لطالب السنة الأولى ليسانس: تجربة كلية العلوم الاقتصادية والتجارية وعلوم التسيير بجامعة بسكرة"، مداخلة في الملتقى الوطني حول المرافقة البيداغوجية للطلاب الجامعي: -نحو فهم جيد لإرساء آليات التفعيل- كلية العلوم الإنسانية والاجتماعية، جامعة بسكرة، 23 نوفمبر 2022
- [3] Disponible sur : <https://etudier.uqam.ca/quelle-est-difference-entre-un-etudiant-regulier-un-etudiant-libre>, consulté le 10/04/2024
- [4] شاكري.س. "المرافقة البيداغوجية كآلية لتحويل الطالب إلى باحث في الجامعات الجزائرية". https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.bsu.edu.eg/Backend/Uploads/PDF/Conference/003%2520%25D8%25B3%259%2585%25D9%258A%25D8%25A9.pdf&ved=2ahUKEwjnkqPFjKGGAxUOUq_uwQEHTQaBXAQFnoECBAQAw&usg=AOvVaw3HoR0RvnBxuzOnlkRxy
- [5] HALIMI, K. (5 décembre 2023). "Développement des Applications Web". Université 8 mai 1945 de Guelma, 112 pages. Disponible sur <http://dspace.univ-guelma.dz/jspui/handle/123456789/15744>, consulté le 05/05/2024
- [6] Disponible sur : <https://www.ideematic.com/dictionnaire-digital/application-web>, consulté le 06/05/2024
- [7] Lomas, A. (2022, December 28). "How to build a web app". Net Solutions.

Disponible sur : <https://www.netsolutions.com/insights/how-to-build-a-web-app/>, consulté le 05/05/2024

[8] Disponible sur : <https://www.example.com/reference3>, consulté le 06/05/2024

[9] Tatnall, A. (2004). "Web Portals : The New Gateways to Internet Information and Services". Hershey, PA : Idea Group Publishing. 363 pages. Disponible sur : <https://books.google.com.br/books?id=5IeT8JhZSmcClpg=PP1hl=pt-BRpg=PR2v=onepageqf=false>, consulté le 10/05/2024

[10] Ater, T. (2017). "Building Progressive Web Apps". Sebastopol, CA : O'Reilly Media, Inc. 80 pages. Disponible sur : https://www.google.dz/books/edition/Building_progressive_web_apps/H8o0DwAAQBAJ?hl=frqbpv=1dq=Building%20Progressive%20Web%20Appsq=PP1printsec=frontcover, consulté le 10/05/2024

[11] Disponible sur : <https://fr.yeeply.com/blog/6-types-developpement-d-applications-web/>, consulté le 10/05/2024

[12] Khalouf, O., et Stoutah, D. (2019). "KhaloufOurdia.pdf", Université Mouloud Mammeri de Tizi Ouzou, 77 pages. Disponible sur : <https://dspace.ummtto.dz/handle/ummtto/12570>, consulté le 10/04/2024

[13] Disponible sur : <https://www.mayasquad.com/glossaire/site-web-dynamique/>, consulté le 10/04/2024

[14] Disponible sur : <https://cynoteck.com/fr/blog-post/reasons-to-treat-web->

[application-architecture-seriously/](#), consulté le 10/04/2024

[15] Ballejos, L. (6 mars 2024). "L'architecture client-serveur". ninjaOne, Disponible sur : <https://www.ninjaone.com/fr/it-hub/endpoint-management/architecture-client-serveur/>, consulté le 06/05/2024

[16] Amrane, S., and Amrane, H. (2013). "Conception et réalisation d'une application web avec J2EE pour la gestion de scolarité". Disponible sur : <https://www.ummo.dz/dspace/bitstream/handle/ummo/12932/AmraneSamia.pdf?sequence=1>, consulté le 03/04/2024

[17] Chen, S.-C., Gulati, S., and Hamid, S. (2003). "A three-tier system architecture design and development for hurricane occurrence simulation". In Information Technology : Research and Education, 2003. Proceedings. ITRE2003. International Conference on (pp. 201-205). IEEE. Disponible sur : https://www.researchgate.net/publication/4056556_A_three_tier_system_architecture_design_and_development_for_hurricane_occurrence_simulation, consulté le 03/05/2024

[18] Disponible sur : https://www.researchgate.net/figure/Tiers-client-server-architecture_fig1_341699804, consulté le 29/05/2024

[19] Chouha, A. (14/10/2023). "Programmation avancée pour le Web", staff.univ-batna2.dz. Disponible sur : <https://staff.univ-batna2.dz/sites/default/files/chouha-adel/files/partie01.pdf%3Fm%3D1697456195>, consulté le 03/04/2024

[20] Disponible sur : <https://www.fedit.ca/actualite-fedit/conseil-emploi/quels-sont-les-langages-informatiques-les-plus-utilises/>, consulté le 07/06/2024

[21] Tanguy, P. (s.d.). Les technologies du Web. Rappels. Bachelor IMTN. Programmation Web 2. IMT Atlantique. 56 pages. Disponible sur : <https://formations.imt-atlantique.fr/bachelor;mtn/ressources/IMTN01TechnoWeb.pdf>, consulté le 07/06/2024

[22] Disponible sur : <https://agence-scroll.com/blog/avantages-et-inconvenients-dune-application-web>, consulté le 05/05/2024

[23] Okon, K., Ozuomba, S., Asuquo, P. (mars 2023). "Development of Journal Management Web Application, Case Study of Journal of Research and Innovations in Engineering." Journal of Research and Innovations in Engineering, 8(3), (2528-9810) pages. Disponible sur : https://www.researchgate.net/publication/37210174017KufreOkon2DEV_EO_F_JOURNAL_MANAGEMENT_WEB_APP_IMJST, consulté le 07/05/2024

[24] Khoirunnisa, N., Haryati, H., Jubaedah, E., Rostiawati, T. (mars 2023). "Designing Website-Based Administration Management Information System." CoreID Journal, 1(1), (21-32) pages. Disponible sur : https://www.researchgate.net/publication/372267653_Designing_Website-Based_Administration_Management_Information_System, consulté le 07/05/2024

[25] Aládé, S. M. (avril 2023). "Design and Implementation of a Web-based Document Management System." International Journal of Information Technology and Computer Science, 15(2), (35-53) pages. Disponible sur : https://www.researchgate.net/publication/370534584_Design_and_Implementation_of_a_Web-based_Document_Management_System, consulté le 07/05/2024

[26] Cheddadi, M.-Y., El Jabri, Z., Hbyaj, A. (2020). Méthode en Cascade (Waterfall model). 13 pages. Disponible sur : <https://www.scribd.com/document/470353004/methode-e-cascade-docx>, Consulté le 05/04/2024

[27] Yādava, S. C. (2009). New Age International (p) Limited. 212 pages. Disponible sur : https://www.google.dz/books/edition/Introduction_à_client_server Computing/AD0TjdCRVEkC?hl=fr&gbpv=0, consulté le 08/05/2024

[28] Olfa Bouchaala.C. (2020/2021). Architecture n/3 et orientée service. Disponible sur : <https://www.scribd.com/document/492075865/Arcitecture-2-tiers>, consulté le 08/05/2024

[29] Rumbaugh, J., Jacobson, I., and Booch, G. (2005). "The Unified Modeling Language Reference Manual" (2nd ed.). Pearson Education, Inc. 742 pages. Disponible sur : <https://www.utdallas.edu/~chung/Fujitsu/UML2.0/Rumbaugh-UML2.0ReferenceCD.pdf>, consulté le 05/05/2024

[30] Disponible sur : <https://pro.arcgis.com/fr/pro-app/latest/help/analysis/geoprocessing/charts/what-is-a-chart-.htm>, consulté le 05/05/2024

[31] Disponible sur : <https://www.researchgate.net/profile/Mohamed-Hadj-Kacem-2/publication/30514387/figure/fig5/AS:669458213376039@1536622684666/Diagrammes-de-comportement.png>, consulté le 04/05/2024

[32] Waykar, Y. (Janvier 2015). "Role of use case diagram in software development" International Journal of Management and Economics. Disponible sur :

https://www.researchgate.net/publication/322991847_role_of_sequence_diagram_in_software_development, consulté le 04/05/2024

[33] Harizi, M. (2012). "The Role of Class Diagram in Estimating Software Size". *International Journal of Computer Applications*, 44(5), (31–33) pages. Disponible sur : <https://doi.org/10.5120/6261-8407>, consulté le 04/05/2024

[34] Grgec, M. (n.d.). "Role of UML sequence diagram constructs in object lifecycle concept". 74 pages. Disponible sur : https://www.researchgate.net/publication/26596376_Role_of_UML_sequence_diagram_constructs_in_object_lifecycle_concept, consulté le 04/05/2024

[35] Disponible sur : <https://www.modeliosoft.com/fr/produits/avantages.html>, consulté le 08/05/2024

[36] Rolland, Christian. (1995). "Présentation de HTML." *Cahiers GUTenberg*, 84 pages. Disponible sur : http://www.numdam.org/item/CG_1995__19870/, consulté le 07/05/2024

[37] Cordier, C. (2023). "Exercices INGINIOUS destinés à l'enseignement secondaire : introduction aux langages HTML/CSS". École polytechnique de Louvain, Université catholique de Louvain. Prom. : Bonaventure, 55 pages. Disponible sur : https://dial.uclouvain.be/downloader/downloader.php?pid=thesis%3A43418&datastream=PDF_01cover=cover-mem, consulté le 07/05/2024

[38] Anoh, L. (19 août 2021). "Le framework". SCRIBD. Disponible sur : <https://www.scribd.com/document/520870339/Le-framework>. Consulté le 07/04/2024.

[39] Disponible sur : <https://www.hostinger.fr/tutoriels/cest-quoi-bootstrap>, consulté le 08/05/2024

[40] Disponible sur : <https://www.data-bird.co/blog/bibliotheque-python>, consulté le 08/05/2024

[41] Sklar, D. (2004). "Développement des Applications Web" (Traduit par É. Jacoboni, en français). O'Reilly, 345 pages. Disponible sur : <https://www.google.dz/books/edition/Introduction%C3%A0PHP5/fAlN55KwKjC?hl=frgbpv=1pq=PR4printsec=frontcover>, consulté le 07/06/2024

[42] Flanagan, D. (2002). "JavaScript" - Pocket Reference (2nd ed.). O'Reilly and Associates. 92 pages. Disponible sur : https://books.google.dz/books?hl=frlr=id=9nifXR0u7xcCoi=fndpg=PA1dq=cour+sur++le+javascriptots=MXQU67hXmKsig=RiDfF90Wlba9RInZheV0ekGQblgredir_esc=yv=onepageq=cour%20sur%20%20le%20javascriptf=false, consulté le 07/05/2024

[43] Kher, J., Parekh, J., Isotiya, P., Malani, P., and Tadhani, H. (2021). "Analysis and Implementation of Domain Hosting and WHOIS Data Web Application". International Journal of Scientific Research and Engineering Development, 68 pages. Disponible sur : https://www.academia.edu/download/66070056/IJSRED_V4I2P10.pdf, Consulté le 07/05/2024.

[44] Dipina Damodaran, B., Salim, S., and Vargese, S. M. (2016). "Performance evaluation of MySQL and MongoDB databases". Int. J. Cybern. Inform. (IJCI), 5, (387-394) pages. Disponible sur : <https://www.academia.edu/download>

[/59431690/PERFORMANCE_EVALUATION_OF_MYSQL_AND_20190528-97716-g7l9vr.pdf](#), consulté le 07/05/2024

[45] Delisle, M. (2012). "Mastering PhpMyAdmin 3.4 for Effective MySQL Management". Packt Publishing. 365 pages. Disponible sur : https://books.google.dz/books?hl=frlr=id=m1cUk-zcyOACoi=fndpg=PT22dq=definition+of++phpMyAdmin+ots=srWeUW9tYGsig=N9Qq05NvZh2omNFiaEzkRyYKQNoredir_esc=yv=onepageq=definition%20of%20%20phpMyAdminf=false, consulté le 07/05/2024

[46] Goodwill, J. (2002). "Integrating the Apache HTTP Server. In Apache Jakarta Tomcat. Berkeley, CA : Apress. (151-158) pages. Disponible sur : https://link.springer.com/content/pdf/10.1007/978-1-4302-0851-8_9.pdf, consulté le 07/05/2024

[47] Disponible sur : <https://code.visualstudio.com/docs/editor/whyvscode>, consulté le 08/05/2024

[48] Disponible sur : <https://www.geeksforgeeks.org/web-browser>, consulté le 08/05/2024

Liste des termes

AJAX JavaScript et XML Asynchrones.

CERN Organisation européenne pour la recherche nucléaire.

CMS Système de Gestion de Contenu.

CSS Feuilles de Style en Cascade.

CSV Valeurs Séparées par des Virgules.

DOM Modèle d'Objet de Document.

e-commerce commerce électronique.

HTML Langage de Balisage Hypertexte.

HTTP Protocole de Transfert Hypertexte.

IDE Environnement de Développement Intégré.

IIS Internet Information Services.

IP Protocole Internet.

ISO Organisation internationale de normalisation .

L.M.D Licence-Master-Doctorat.

MSSQL Microsoft SQL Server.

MVC Modèle-Vue-Contrôleur.

NCSA Mosaic Mosaïque du Centre National des Applications de Superinformatique.

OOHDM Méthodologie de Conception d'HypermédiAs Orientée Objet.

PHP Préprocesseur Hypertexte.

ROI Retour sur Investissement..

SGBDR Système de Gestion de Base de Données Relationnelle.

SGML Langage de Balisage Généralisé Standardisé.

SPA Application Monopage.

SQL Langage de Requête Structurée.

UML Langage de Modélisation Unifié.

W3C Consortium World Wide Web..

Web World Wide Web.

XAMPP Cross-platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P).