

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

Faculté des Mathématiques et de
l'Informatique

Département d'Informatique

N° :



DOMAINE : Mathématiques et
Informatique

FILIERE : Informatique

OPTION : RTIC

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par : Abdallaoui Chaima

Korichi Attahir

Intitulé

**Conception et implémentation d'une
application web pour la gestion de la
planification du département.**

Cas d'étude: Département d'Informatique

Soutenu devant le jury composé de :

Hammani Said
Kamel Mohamed
Therafi Abdellah

Université de M'sila
Université de M'sila
Université de M'sila

Président
Rapporteur
Examineur

Année universitaire : 2022 / 2023

Dédicaces

Avant toute chose, je rends grâce à Dieu le tout-puissant.

Je dédie humblement ce travail avec amour et respect :

À mon cher père, qui a été mon roc, mon soutien indéfectible. Tes encouragements, tes prières et ton amour constant m'ont porté tout au long de ce parcours. Je t'entoure de ma plus profonde affection.

À celle qui occupe la place la plus chère dans mon cœur, ma mère qui n'a jamais cessé de croire en moi, de prier pour ma réussite et de m'entourer de son amour inconditionnel. Tu as tout fait pour me donner le bonheur, et pour cela, je te suis éternellement reconnaissante. Que Dieu te protège.

À mes chers frères et sœurs surtout les petites Mohamed, Alea Arahmane et Youcef, qui ont partagé avec moi les joies et les défis de la vie. Votre soutien et votre présence ont été précieux à mes yeux.

À tous mes enseignants en faculté de mathématiques et d'informatique, qui m'ont transmis leur savoir, leur passion et qui ont façonné mon esprit. Vos enseignements ont été une source d'inspiration et de développement personnel.

À tous mes amis et collègues de la promotion 2018/2023

Que cette dédicace témoigne de ma profonde gratitude envers chacune de ces personnes. Votre présence et votre soutien ont été essentiels dans la réalisation de ce travail. Que Dieu vous bénisse tous.

Chaima

Dédicaces

Je dédie ce travail à mes très chers parents,
Qui ont été présents à mes côtés et m'ont soutenu tout au long de mon parcours.

Leur amour, leur soutien inconditionnel et leurs sacrifices ont été une source
d'inspiration et de motivation pour moi.

Je suis extrêmement reconnaissant envers eux et j'espère pouvoir les rendre fiers
de moi.

Leur amour et leur présence ont été les piliers de ma réussite.
Merci du fond du cœur pour tout ce que vous avez fait pour moi.

Attahir

Remerciement

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(وَإِذْ تَأَذَّنَ رَبُّكُمْ لَئِن شَكَرْتُمْ لَأَزِيدَنَّكُمْ وَلَئِن كَفَرْتُمْ إِنَّ عَذَابِي لَشَدِيدٌ)

[إبراهيم : 7]

Tout d'abord, nous remercions le bon Dieu pour nous avoir donné la force, la santé et le courage nécessaires pour mener à bien ce travail.

Nous souhaitons également exprimer notre gratitude envers tous ceux qui nous ont soutenus, encouragés et guidés tout au long de ce parcours.

En particulier, nous aimerions adresser nos remerciements les plus sincères à notre mentor, **M. Kamel Mohammed**, pour son soutien constant, ses précieux conseils et le temps précieux qu'il nous a accordé. C'est grâce à lui que nous avons pu aborder ce sujet, et nous avons toujours su que nous pouvions compter sur lui. Sa contribution a été inestimable.

Nous tenons également à exprimer notre reconnaissance envers les membres du jury, **M. Hamani S.** et **M. Tarafi A.**, pour l'honneur qu'ils nous font en évaluant notre travail.

Enfin, nous souhaitons exprimer notre gratitude envers toutes les personnes qui ont apporté leur aide et leur encouragement, de près ou de loin, pour la réalisation de ce travail

Table des matières

Remerciement.....	i
Table des matières	ii
Liste des figures.....	vii
Liste des tableaux.....	viii
Liste des abréviations	ix
Introduction Générale	1
Chapitre 1:Etude de L'existant	
1. Introduction :.....	4
2. Présentation de l'organisme d'accueil	4
2.1. Présentation de la faculté MI.....	4
2.2. Département de l'Informatique	5
2.2.1. Les formations offertes par le département.....	5
2.2.3. Personnel d'administration département informatique.....	5
2.2.4. Personnel d'enseignement de département informatique	6
2.2.5. Infrastructures et ressources humaines disponibles	6
2.3. Etude du système pédagogique	6
2.4. L'activité pédagogique	7
2.5. Les ressources.....	7
2.5.3. Les ressources de type « salle »	7
2.5.1 Les ressources de type « enseignant ».....	8
2.5.4. Les ressources de type « groupe.....	8
2.5.5. Les ressources de type « matériel ».....	8
2.6. Modélisation du temps	8
2.6.3. Les entités temporelles.....	8
2.6.4. Les séances et les réservations	8
2.6.5. Les filières.....	9
3. La problématique de la planification d'horaires de travail.....	10
3.2. Qu'est-ce que la planification ?.....	10
3.3. Qu'est Ce qu'un planning ?.....	10
3.4. A Quoi sert un planning ?.....	11
3.5. Comment sont évalués les plannings ?.....	12
3.6. Qui peut se charger de l'élaboration d'un planning ?.....	12
3.7. Planning dans la pédagogie	12
4. Les technologies du web	13
4.2. Un site web	13
4.3. Les applications web	13

Table des matières

4.1.3.	Serveur web	16
4.1.4.	Le fonctionnement d'une application web	16
4.1.5.	L'hébergement d'une application web	16
4.1.6.	Les langages et les outils utilisés pour développer une application web 17	
4.1.7.	Application de Gestion	18
5.	Conclusion	18
Chapitre 2:Analyse et conception		
1.	Introduction	20
2.	Présentation UML	20
2.1.	Définition.....	20
2.2.	Historique	20
2.3.	Les Vues UML	21
2.3.1.	Point de vue structurel.....	21
2.3.2.	Point de vue comportemental.....	21
2.3.3.	Point de vue interaction.....	22
3.	Présentation de notre démarche de modélisation.....	22
4.	Analyse	23
4.1.	Identification des acteurs	23
4.2.	Identification des cas d'utilisation.....	23
4.3.	Diagramme de contexte	24
4.4.	Diagramme de cas d'utilisation	25
4.4.1.	Description des cas d'utilisation avec des scénarios.....	25
4.4.2.	Diagramme de Cas d'utilisation Authentification.....	30
4.4.3.	Diagramme de cas d'utilisation « Héritage ».....	30
4.4.4.	Diagramme de cas d'utilisation concernant l'administrateur.....	31
4.4.5.	Diagramme de cas d'utilisation concernant l'enseignant.....	32
4.4.6.	Diagramme de cas d'utilisation concernant le visiteur	32
4.4.7.	Diagramme de cas d'utilisation génération de planning	32
4.5.	Diagramme de séquence	33
4.5.1.	Diagramme de séquence de cas d'utilisation « Authentification »	33
4.5.2.	Diagramme de séquence de cas d'utilisation « Modifier un enseignant » 34	
4.5.3.	Diagramme de séquence de cas d'utilisation « Ajouter enseignant » ..	34
4.5.4.	Diagramme de séquence de cas d'utilisation « supprimer enseignant » 35	
4.5.5.	Diagramme de séquence de cas d'utilisation « Rechercher un enseignant » 35	

Table des matières

4.5.6.	Diagramme de séquence de cas d'utilisation « Création de planning »	36
5.	Conception	36
5.1.	Diagrammes d'activité.....	36
5.1.1.	Diagramme d'activité du cas d'utilisation « S'authentifier »	37
5.1.2.	Diagramme d'activité du cas d'utilisation « Modification un enseignant »	37
5.1.3.	Diagramme d'activité du cas d'utilisation « Rechercher un enseignant »	38
5.1.4.	Diagramme d'activité du cas d'utilisation « Suppression d'un Enseignant ».....	38
5.1.5.	Diagramme d'activité du cas d'utilisation « Ajouter Enseignant »	39
5.1.6.	Diagramme d'activité du cas d'utilisation « Création de planning »....	39
5.2.	Diagramme de classe	40
5.2.1.	Règles de gestion du diagramme de classe	40
5.2.2.	Diagramme de classe global.....	41
6.	Modèle relationnel	41
7.	Conclusion	42
Chapitre 3: Réalisation		
1.	Introduction :.....	45
2.	L'environnement matériel :	45
3.	Outils et langages de développement utilisés	45
3.	Architecture MVC (Model-View-Controller)	51
4.	Présentation de l'application.....	52
4.1.	Représentation des interfaces de l'application	52
5.	Conclusion :.....	58
Conclusion générale		57
Bibliographies		59
Résumé		

Liste des figures

Figure 1 :Organigramme de la faculté de MI.....	4
Figure 2: Application web hébergée sur un serveur.....	14
Figure 3:Fonctionnement d'une application web.....	16
Figure 4Historique UML	21
Figure 5Démarche de modélisation de l'application.....	22
Figure 6 Diagramme de contexte	25
Figure 7: Diagramme de Cas d'utilisation Authentification	30
Figure 8: Diagramme de cas d'utilisation « Héritage »	30
Figure 9: Diagramme de cas d'utilisation concernant l'administrateur	31
Figure 10: Diagramme de cas d'utilisation concernant l'enseignant	32
Figure 11:Diagramme de cas d'utilisation concernant le visiteur.....	32
Figure 12: Diagramme de cas d'utilisation génération de planning.....	32
Figure 13: Diagramme de séquence de cas d'utilisation « Authentification ».....	33
Figure 14: Diagramme de séquence de cas d'utilisation « Modifier un enseignant ».....	34
Figure 15: Diagramme de séquence de cas d'utilisation « Ajouter enseignant ».....	34
Figure 16!Diagramme de séquence de cas d'utilisation « supprimer enseignant	35
Figure 17: Diagramme de séquence de cas d'utilisation « Rechercher un enseignant ».....	35
Figure 18: Diagramme de séquence de cas d'utilisation « Création de planning	36
Figure 19: Diagramme d'activité du cas d'utilisation « S'authentifier »	37
Figure 20: Diagramme d'activité du cas d'utilisation « Modification un enseignant »	37
Figure 21Diagramme d'activité du cas d'utilisation « Rechercher un enseignant.....	38
Figure 22 d'activité du cas d'utilisation « Suppression d'un Enseignant	38
Figure 23 :Diagramme d'activité du cas d'utilisation « Ajouter Enseignant »	39
Figure 24 Diagramme d'activité du cas d'utilisation « Création de planning ».....	39
Figure 25 :Diagramme de classe global	41
Figure 26 Logo Visual Paradigm	45
Figure 27:Logo PhpStorm.....	46
Figure 28:Logo XAMPP	46
Figure 29 : Logo Apache	47
Figure 30 Logo MySQL.....	47
Figure 31: Logo php.....	47
Figure 32 : Logo Bootstrap	48
Figure 33Logo Git.....	48
Figure 34Logo GitHub.....	49
Figure 35Logo Ajax	50
Figure 36Logo JQuery	50
Figure 37Logo Laravel.....	51
Figure 38:Architecture MVC	52
Figure 39: Interface de notre application	53
Figure 40: L'interface de connexion.....	53
Figure 41:L'interface de connexion de l'administrateur	54
Figure 42:Interface d'Accueil	54
Figure 43: Interface de gestion des enseignants.....	55
Figure 44 :L'interface de créer un planning	55
Figure 45: Interface de consulter d'un planning.....	56
Figure 46:Interface de contacter admin.....	56
Figure 47: Interface de gérer les contacts	57
Figure 48: Interface de configuration.....	57
Figure 49: Interface d'impression d'un planning	58

Liste des tableaux

Tableau 1 Identification des cas d'utilisation	24
Tableau 2 Cas d'utilisation « Authentification »	26
Tableau 3 Cas d'utilisation « gérer les enseignants ».....	27
Tableau 4: Cas d'utilisation relative à l'administrateur	29

Liste des abréviations

Faculté MI : Faculté des Mathématiques et de l'Informatique

UMSILA : Université de M'Sila

TD : Travaux Dirigés

TP : Travaux Pratiques

UML : Unified Modeling Language

ORM: Object-Relational Mapping

XAMPP: Cross-Platform, Apache, MYSQL, PHP ET Perl

DOM: Document Object Model

CSS: Cascading Style Sheets Dis

HTML: Hyper Text Markup Language

AJAX: JavaScript and XML

SQL Structured Query Language

PHP: Hypertext Preprocessor

IDE : Environnement de développement intégré

MVC : Model-View-Controller

Introduction Générale

Préface :

L'évolution constante des technologies de l'information a transformé de nombreux domaines, y compris celui de la gestion et de la planification des départements dans les établissements d'enseignement supérieur. Les facultés et les départements universitaires sont confrontés à des défis majeurs lorsqu'il s'agit d'organiser efficacement les ressources humaines, les cours, les salles de classe et les horaires pour répondre aux besoins des étudiants et des enseignants. Dans ce contexte, les applications web jouent un rôle crucial en offrant des solutions flexibles et automatisées.

La planification des activités au sein d'un département universitaire représente souvent un défi majeur en raison des nombreux changements pédagogiques et des contraintes diverses liées aux enseignants, aux filières, aux salles de classe et aux types d'enseignement. Les méthodes manuelles utilisées actuellement entraînent des difficultés organisationnelles, des erreurs potentielles et une perte de temps considérable. Afin de remédier à ces problèmes, nous avons choisi de développer une application web qui automatisera et facilitera la gestion du planning du département d'informatique.

Problématiques :

Ce mémoire se focalise sur les problèmes liés à la planification et à la gestion du planning au sein du département d'informatique de la Faculté des Mathématiques et d'Informatique de Msila. Nous avons identifié plusieurs défis importants, notamment :

- Traitement fastidieux et manuel de nombreuses informations cruciales, entraînant un risque accru d'erreurs et de conflits. L'utilisation de méthodes manuelles pour gérer les informations relatives au planning expose le département à des erreurs humaines, telles que des confusions d'horaires ou des doublons de réservations.
- Perte de temps significative due à la gestion manuelle du planning, ce qui limite la flexibilité et l'efficacité globale du département. Les tâches manuelles répétitives nécessaires pour mettre à jour, ajuster ou modifier le planning consomment des ressources précieuses qui pourraient être mieux utilisées dans d'autres aspects de la gestion académique.
- Manque de souplesse dans le travail, avec des difficultés à effectuer des ajustements et des modifications rapides en cas de changements imprévus.

Introduction Générale

Ces problèmes soulignent la nécessité d'une solution informatisée efficace pour la gestion du planning, permettant une meilleure optimisation, une réduction des erreurs, une utilisation plus efficace du temps et une plus grande flexibilité face aux changements.

L'objectif :

Pour remédier à ces problèmes, nous proposons de développer une application web qui facilitera la gestion de planning de notre département.

L'objectif principal de notre application est de fournir une solution informatisée de planification et de gestion du planning pour notre département d'informatique de la Faculté des Mathématiques et d'Informatique de Msila. Cette application vise à améliorer l'efficacité et la cohérence de la planification, à réduire les erreurs et les conflits, et à faciliter la tâche des administratifs, des enseignants et des étudiants. Elle permettra une communication transparente et réactive entre les différentes parties concernées.

Organisation du mémoire :

Dans ce mémoire, nous avons structuré notre travail en trois chapitres pour aborder de manière approfondie la conception et l'implémentation de cette application.

Le premier chapitre : Étude de l'existant, présentation de l'organisme d'accueil et des technologies web utilisées.

Le deuxième chapitre : Analyse et conception du système, avec une description des étapes de conception, la modélisation des données, l'architecture logicielle et les fonctionnalités attendues.

Le troisième chapitre : Réalisation concrète de l'application, en mettant en évidence les langages et les logiciels utilisés, ainsi que quelques interfaces essentielles et les fonctionnalités clés.

Conclusion : Synthèse du travail réalisé et perspectives d'amélioration et d'extension de l'application dans le futur.

Chapitre 1 :
Etude de L'existant

1. Introduction :

L'étape de l'étude représente une phase essentielle dans le déroulement d'un projet informatique. Dans ce chapitre, nous aborderons trois phases distinctes.

- La première phase se concentre sur la présentation détaillée de l'organisme d'accueil, tout en examinant minutieusement le système d'étude et l'activité pédagogique de cet organisme, ainsi que les différentes ressources disponibles. Une attention particulière sera portée aux notions du temps.
- Dans La deuxième phase examine en détail la planification des horaires dans les organismes, mettant en lumière sa complexité et les défis quotidiens qui y sont associés. Il vise à approfondir notre compréhension de la nature de la planification et de la complexité qui lui est inhérente.
- La dernière phase dans ce chapitre se focalise sur le développement des applications web, en soulignant les spécificités techniques et ergonomiques qui leur sont propres. Dans ce contexte, des méthodes de conception et de travail spécifiques sont présentées et recommandées afin d'assurer l'efficacité des applications web. De plus, cette partie apporte des informations générales sur les technologies web.

2. Présentation de l'organisme d'accueil :

2.1. Présentation de la faculté MI :[13]

La création de la faculté des mathématiques et d'informatique de M'Sila a été officialisée par le décret du 24 février 2010, qui établit les départements qui la composent. Cette faculté est composée de deux départements et Socle commun, répartis de la manière suivante :

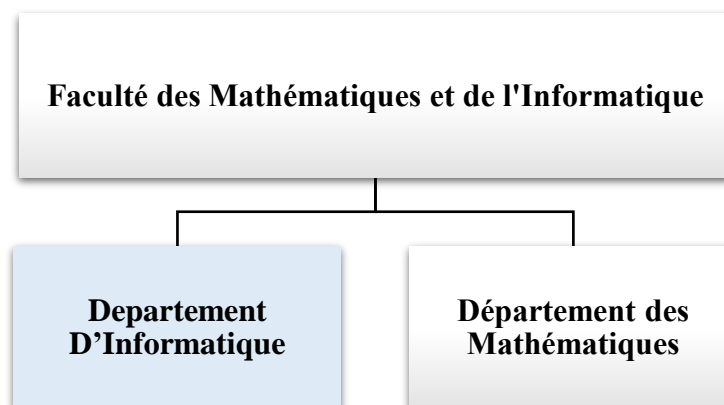


Figure 1 : Organigramme de la faculté de MI

Chapitre 1 : Etude de L'existant

2.2. Département de l'Informatique : [13]

Dans le cadre de notre étude, notre intérêt se porte exclusivement sur un département spécifique. C'est le département d'informatique. En 1997, l'institut d'informatique a été établi en tant qu'annexe du centre universitaire Mohamed Boudiaf. Après que le centre universitaire a été élevé au statut d'université en 2000 et que le système des facultés a été mis en place, notre institut est devenu un département de la faculté des sciences et de l'ingénierie. Par la suite, conformément au décret mentionné précédemment, il a intégré la faculté des mathématiques et d'informatique en tant que département.

2.2.1. Les formations offertes par le département :

Le département d'informatique propose une variété de formations spécialisées dans différents domaines de l'informatique, adaptées à différents niveaux d'études. Voici un aperçu des différences spécialités proposées :

- ❖ Deux licences académiques en Informatique :
 - Ingénierie des Systèmes d'Information et du Logiciel (ISIL)
 - Systèmes Informatiques (SI)
- ❖ Quatre Masters académiques :
 - Réseaux et Technologies de l'Information et de la Communication (RTIC) .
 - Informatique Décisionnelle et Optimisation (IDO)
 - Système d'Information et Génie Logiciel (SIGL)
 - Intelligence Artificielle (IA)
- ❖ Doctorat en Informatique :
 - Intelligence Artificielle
 - Génie Logiciel
 - Réseaux et Systèmes Distribués.

2.2.3. Personnel d'administration département informatique :

L'administration du département d'informatique sont composés du :

- ❖ Chef de Département
- ❖ Adjoint au Chef de Département
- ❖ Secrétaire du chef de département
- ❖ Adjoint chargé de la pédagogie

2.2.4. Personnel d'enseignement de département informatique :

Le département d'informatique dispose d'un corps enseignant spécialisé et compétent chargé de l'encadrement des étudiants.

2.2.5. Infrastructures et ressources humaines disponibles :

Afin d'assurer son fonctionnement, la direction des études dispose d'installations et de ressources humaines considérables, comprenant :

- ❖ Quater amphithéâtres
- ❖ Huit salles de travaux dirigés
- ❖ Vingt laboratoires de travaux pratiques dispose d'équipements d'instruments de pointe.

- (Les deux amphithéâtres et les huit salles TD sont commun avec les étudiants de Département des Mathématiques).

2.3. Etude du système pédagogique : [1]

Le département d'informatique de L'UMSILA propose différentes formations du LMD, d'une durée de trois à huit ans. Les étudiants s'inscrivent en début d'année universitaire, généralement en septembre, selon les formations. Chaque formation a un programme pédagogique détaillé indiquant les matières, les horaires et les informations relatives aux cours, travaux dirigés (TD) et travaux pratiques (TP). Les formations sont organisées en sections et peuvent être divisées en groupes en fonction des besoins et des ressources disponibles. Les enseignants ont des obligations horaires définies par leur statut, avec des coefficients attribués selon le type d'enseignement. Les plannings sont établis par les responsables pédagogiques en collaboration avec les enseignants, et les salles sont attribuées après la création des plannings. Les plannings sont également utilisés par le personnel administratif pour la gestion des intervenants, l'achat d'équipements et la planification des locaux. L'analyse des plannings permet de détecter les problèmes et d'améliorer la planification pour les années à venir. Cependant, la croissance des ressources disponibles ne suit pas la diversification des formations, ce qui complexifie la gestion des ressources. Un outil d'aide est donc nécessaire pour une gestion efficace, prenant en compte les profils et les compétences informatiques des utilisateurs. La robustesse, la rapidité et la puissance de l'outil sont des critères importants pour faciliter son utilisation par différents utilisateurs.

2.4. L'activité pédagogique : [1]

La modélisation de l'activité pédagogique en utilisant trois entités principales : les matières, les enseignements et les modules.

Chaque **matière** est définie par son nom, sa description pédagogique, son type (cours, travaux dirigés, travaux pratiques, etc.) et sa discipline.

Chaque **enseignement** est caractérisé par son nom, ses enseignants, ses groupes d'étudiants, ses matériels par défaut (ou ressources par défaut de manière abstraite), sa matière associée, son volume horaire total et sa durée par défaut. De plus, chaque enseignement indique la liste des enseignements préalables requis. Les enseignants, les groupes d'étudiants et les matériels par défaut sont considérés comme les ressources par défaut de l'enseignement. Les matériels par défaut sont ceux qui sont attribués par défaut aux séances de l'enseignement, mais il est possible que les matériels réellement utilisés lors de chaque séance diffèrent de ceux par défaut. Certains enseignements peuvent être assurés simultanément par plusieurs enseignants, comme c'est le cas pour certains travaux pratiques d'informatique. La durée par défaut correspond à la durée choisie par défaut pour chaque séance de l'enseignement, bien que la durée réelle de chaque séance puisse différer de cette durée par défaut.

Enfin, les **modules** regroupent plusieurs enseignements au sein d'un même ensemble.

2.5. Les ressources : [2]

Cette étude se concentre sur les ressources nécessaires à l'élaboration d'un planning, notamment les salles, les enseignants, les étudiants et le matériel. Ces ressources sont définies par des données abstraites et spécifiques. Les données abstraites incluent un code d'identification, un calendrier indiquant la disponibilité et l'indisponibilité, ainsi qu'une description générale. Chaque ressource possède également des caractéristiques spécifiques propres à son type. Cette distinction est importante car elle permet à l'outil de s'adapter facilement à de nouveaux types de ressources.

Dans la suite, nous détaillons les caractéristiques spécifiques des ressources considérées dans cette étude :

2.5.3. Les ressources de type « salle » :

Une **salle** représente un espace où des enseignements sont dispensés. Le type d'une salle indique le genre d'enseignement qui peut y être réalisé, par exemple, si elle est spécifiquement destinée aux travaux dirigés (TD) ou aux travaux pratiques (TP).

Chapitre 1 : Etude de L'existant

2.5.1 Les ressources de type « enseignant » :

L'enseignant est une personne qualifiée pour dispenser des cours. Chaque enseignant possède les caractéristiques suivantes : son nom et son prénom, son grade, sa résidence administrative et sa spécialité. La spécialité indique les matières que l'enseignant est apte à enseigner.

2.5.4. Les ressources de type « groupe » :

Un **groupe** est constitué d'étudiants et peut inclure d'autres groupes qui peuvent être réduits à un seul étudiant, formant ainsi un binôme ou un monôme. Dans cette modélisation, il n'y a pas de distinction entre un groupe et un étudiant individuel. Généralement, les étudiants d'une filière sont organisés en sections, et chaque section est ensuite subdivisée en plusieurs groupes.

2.5.5. Les ressources de type « matériel » :

Afin d'assurer le bon déroulement de l'enseignement, l'enseignant utilise un **matériel** pédagogique. En général, ce matériel se limite à un tableau, des craies, éventuellement un rétroprojecteur et parfois un vidéoprojecteur. Cependant, pour certains enseignements spécifiques, l'enseignant peut avoir besoin d'un matériel spécifique.

2.6. Modélisation du temps : [1]

Les études utilisateur ont révélé différentes perceptions du temps, nécessitant deux notions fondamentales : l'instant et la durée. Pour les plannings, la notion de fréquence est introduite, bien qu'elle puisse être exprimée à partir de l'instant et de la durée. La fréquence indique la répétition d'un événement, avec une durée minimale entre chaque occurrence. Par exemple, un cours ayant lieu tous les 15 jours signifie qu'il y a une durée d'au moins 15 jours entre deux séances consécutives. Ainsi, on distingue la durée de chaque séance et la durée entre les séances.

2.6.3. Les entités temporelles :

Afin de représenter le temps de manière modélisée, plusieurs entités sont définies : la date, l'heure, la durée, le créneau et le calendrier.

Une date représente un instant précis et est définie par un ensemble de trois valeurs (jour, mois, année). À partir de ce triplet, une valeur est attribuée pour situer cette date sur l'axe des jours.

2.6.4. Les séances et les réservations :

Une **séance** représente une occurrence temporelle d'un enseignement à une date précise, pendant un créneau spécifique. L'ensemble des séances d'une ressource est appelé le planning de cette ressource, comprenant toutes les séances où la ressource est impliquée.

Chapitre 1 : Etude de L'existant

Ainsi, il est possible de considérer le planning d'une salle de la même manière que celui d'un enseignant. La notion de planning est également appliquée aux enseignements, ce qui permet de suivre l'évolution temporelle d'un enseignement spécifique.

Une **réservation** correspond à l'ajout d'un créneau au planning d'une ressource. Elle représente une option posée sur l'utilisation de cette ressource. Contrairement à une séance, une réservation n'est pas associée à un enseignement particulier. Les réservations offrent de la flexibilité dans l'utilisation de l'outil, notamment lors de la création des emplois du temps par les différents responsables.

2.6.5. Les filières :

La notion de filière est utile pour réduire le nombre de ressources à prendre en compte. En effet, le responsable d'une filière n'a besoin de connaître que le planning des ressources spécifiques à sa filière. Cependant, certains conflits ne peuvent être détectés que lorsque les différentes filières sont regroupées dans une base de données commune. Cela concerne principalement les conflits d'enseignants et de salles, car il est fréquent qu'un enseignant intervienne dans plusieurs filières ou qu'une salle soit partagée par plusieurs filières.

2.6.6. Les contraintes sur les données : [3]

Les observations sur le terrain révèlent que les données gérées doivent respecter certaines contraintes afin d'assurer leur cohérence. De manière générale, ces contraintes peuvent être regroupées en deux catégories : **les contraintes physiques** et **les contraintes pédagogiques**.

- Les contraintes physiques :

Le non-respect de ces contraintes entraînerait des situations conflictuelles. On parle de "conflit physique de ressource" entre deux séances S1 et S2 lorsque ces séances partagent une ressource pendant une période de temps non nulle. Les contraintes physiques sont les suivantes :

- Une ressource ne peut pas être occupée simultanément par deux séances différentes.
- Il est impossible de placer plus d'étudiants qu'il n'y a de places disponibles dans une salle.
- Les calendriers doivent être respectés pour toutes les ressources.

- Les contraintes pédagogiques :

Les contraintes pédagogiques se distinguent des contraintes physiques par leur possibilité d'être éventuellement enfreintes, ce qui entraîne des plannings de moindre qualité

du point de vue pédagogique. Ces contraintes sont généralement utilisées pour définir ce qu'est un "bon" planning. Les contraintes pédagogiques visent à :

- Uniformiser la durée des séances d'un cours.
- Éviter les créneaux horaires inoccupés dans le planning.
- Éviter de programmer plusieurs séances consécutives du même cours dans une journée.

3. La problématique de la planification d'horaires de travail : [4]

La planification des horaires de travail est un processus extrêmement complexe qui vise à organiser les activités humaines, principalement liées au travail, de manière optimale dans le temps. Son objectif est d'optimiser l'utilisation des ressources afin de répondre à une charge de travail prévisionnelle, tout en respectant différentes contraintes. Ce processus aboutit à l'établissement de programmes définissant les horaires de travail et de repos. Afin d'approfondir notre compréhension de ce qu'est la planification et de la complexité qui y est associée, nous examinons un ensemble de des questions :

3.2. Qu'est-ce que la planification ? [5]

La planification est un outil de gestion essentiel visant à créer des programmes qui organisent et structurent le travail. Son objectif principal est d'évaluer les compétences individuelles de chaque personne, d'anticiper les activités futures et de répondre aux besoins en personnel. L'objectif de la planification est d'attribuer de manière optimale les ressources humaines à chaque période de temps, en prenant en compte les contraintes et exigences spécifiques à chaque intervalle. Ce processus permet d'optimiser l'utilisation des ressources disponibles et d'assurer une exécution efficace des tâches. Une planification bien élaborée contribue à maximiser la productivité, à minimiser les retards et à coordonner harmonieusement toutes les parties prenantes impliquées dans le processus.

3.3. Qu'est Ce qu'un planning ? [4]

Les plannings sont des calendriers qui regroupent les informations relatives au temps, à l'affectation du personnel, aux jours et horaires de travail, ainsi qu'aux congés et repos. Ils servent à organiser les horaires de présence du personnel et à planifier les différentes tâches qu'ils effectuent.

Chapitre 1 : Etude de L'existant

- Le planning des horaires de présence est utilisé pour anticiper les heures de travail du personnel sans spécifier les tâches quotidiennes. Cela peut être fait pour des raisons de sécurité ou pour une plus grande flexibilité.
- Le planning des tâches est utilisé dans les entreprises qui ont des métiers et compétences distincts, afin d'affecter le personnel en fonction des tâches spécifiques. Cela nécessite une décomposition détaillée des opérations et l'identification des tâches que chaque personne peut accomplir.

Les plannings peuvent être quotidiens (précisant les pauses et les périodes de travail de chaque employé pour la journée), hebdomadaires (utilisés pour la rémunération hebdomadaire), mensuels (utilisés pour le calcul des coûts liés à la paie mensuelle) ou annuels (permettant de gérer les congés annuels des employés).

Selon leur nature et les secteurs d'activité concernés, les plannings peuvent avoir différentes appellations. Un tableau de service est un planning qui spécifie les horaires de travail de chaque employé de manière nominative sur une période d'un mois. Une grille de travail, quant à elle, représente les horaires de travail et de repos de manière non nominative sur un nombre entier de semaines.

Certains plannings sont périodiques, ce qui signifie qu'ils suivent une certaine périodicité dans les horaires individuels, c'est-à-dire qu'ils se répètent après une durée définie. En revanche, les plannings acycliques diffèrent chaque semaine et ne suivent pas de modèle régulier.

3.4. A Quoi sert un planning ? [4]

Depuis les années 80, la gestion des ressources humaines est devenue une activité stratégique cruciale pour les organisations et les entreprises. L'importance de développer des plannings a augmenté considérablement, car ils offrent divers avantages :

- Pour les entreprises qui fonctionnent en continu ou presque, les plannings permettent une répartition adéquate du personnel (compagnies aériennes, entreprises de transport, hôpitaux, établissements éducatifs, etc.).
- Les entreprises cherchant à être plus accessibles à leur clientèle peuvent étaler leurs horaires d'ouverture (grands magasins, banques, etc.).
- Toutes les entreprises peuvent ainsi répondre à leurs exigences de productivité et gérer plus efficacement les présences et les absences du personnel.

Les situations où l'utilisation d'un planning est bénéfique sont nombreuses, ce qui justifie la présence de différentes formes de plannings au sein d'un même système.

3.5. Comment sont évalués les plannings ? [4]

Afin d'obtenir des plannings satisfaisants, il est nécessaire de respecter plusieurs contraintes et de parvenir à un compromis équilibré entre les différents acteurs impliqués, tels que le chef d'entreprise, le planificateur, le commercial, le syndicaliste et le salarié. Lorsque différentes solutions alternatives sont disponibles, une négociation s'engage, au cours de laquelle chaque acteur exprime son opinion. Les points d'accord sont rapidement résolus, tandis que les points litigieux font l'objet de débats, aboutissant à des solutions de compromis. La complexité des négociations augmente avec le nombre d'acteurs et de solutions alternatives. Les outils informatiques peuvent fournir une assistance précieuse, notamment pour l'acquisition et la confrontation des données individuelles.

3.6. Qui peut se charger de l'élaboration d'un planning ? [4]

La délégation de cette tâche aux planificateurs, des cadres au sein des entreprises et organismes, est courante. Ces planificateurs doivent choisir la meilleure décision en fonction des préférences des différentes parties prenantes. Ils doivent également justifier leur choix en s'appuyant sur leur expérience, car leur expertise leur permet d'évaluer rapidement les différentes options et de prendre des décisions éclairées pour orienter la recherche de solutions de qualité supérieure, aboutissant ainsi à un choix pertinent.

3.7. Planning dans la pédagogie : [2]

La réalisation des emplois du temps dans les établissements scolaires est une tâche difficile et cruciale. Pour parvenir à une solution, il est nécessaire de s'adapter aux changements dynamiques de l'environnement tout en prenant en compte les nombreuses contraintes telles que l'interdépendance des programmes d'enseignement, la diversité des matières étudiées, les différents types de cours, la durée des cours, la disponibilité des enseignants et des salles. Ce problème peut être divisé en deux catégories principales : la planification des horaires des cours et celle des examens. La création manuelle de ces plannings est extrêmement exigeante, ce qui a suscité un grand intérêt de la part de la communauté scientifique. Les problèmes d'emplois du temps vont de la construction des plannings semestriels ou annuels dans les universités, écoles et collèges, à l'organisation des examens à la fin de ces périodes. Les premières tentatives d'établissement des emplois du temps étaient réalisées manuellement, et une fois créés, ils restaient généralement statiques, avec seulement quelques modifications nécessaires. Cependant, au fil des années, la nature de l'enseignement a considérablement évolué, rendant la planification des emplois du temps beaucoup plus complexe qu'auparavant. Par conséquent, la demande de génération automatisée des emplois du temps augmente, et il est essentiel de développer un système

capable de produire des solutions valides. Au cours des 30 dernières années, de nombreuses approches liées à l'automatisation des emplois du temps ont été publiées, et plusieurs applications ont été développées et mises en œuvre avec des degrés de réussite variables.

4. Les technologies du web : [6]

Le « Web », raccourci de « World Wide Web », est un système d'information hypermédia comprenant des documents multimédias (textes enrichis de sons, graphiques, images fixes et animées, vidéos...) qui sont interconnectés via des liens hypertextes. Nous allons maintenant fournir des définitions pour les termes "site web" et "application web" et ensuite présenter brièvement la différence entre les deux.

4.2. Un site web : [6]

- Un site web est composé d'un ensemble de documents structurés appelés pages, stockées sur un serveur connecté à Internet. Les sites web peuvent être classés en deux types :
- Site web statique :

Dans ce type de site web, les pages ont un contenu fixe, ce qui signifie que chaque visiteur voit les mêmes informations à chaque fois. Les sites statiques ne contiennent généralement pas de langage de programmation ou de bases de données. Les langages couramment utilisés pour ces types de sites sont les langages de balisage tels que HTML et CSS.

- Un site web dynamique :

Fonctionne de manière différente d'un site statique. Les pages d'un site dynamique ont un contenu modifiable, ce qui signifie que les visiteurs peuvent interagir avec le site et voir des contenus différents. Chaque visiteur peut donc voir des informations personnalisées en fonction de ses actions ou préférences.

Pour permettre cette interactivité, le serveur du site utilise un langage de traitement tel que PHP, ainsi qu'une base de données comme MySQL. Le langage de traitement est utilisé pour effectuer des opérations et des traitements sur les données, tandis que la base de données est utilisée pour stocker et récupérer les informations nécessaires.

4.3. Les applications web : [15]

Une application web est un programme ou un logiciel qui s'exécute dans un navigateur web. Sa partie visuelle est créée à l'aide de langages tels que HTML, CSS et/ou JavaScript, tandis que son traitement utilise des serveurs tels que AMP, MAMP, XAMP, etc.

Chapitre 1 : Etude de L'existant

Pour qu'une application web soit fonctionnelle et permette un traitement, trois composants principaux doivent être disponibles. Ces composants constituent l'architecture d'une application web :

- Un serveur web
- Une connexion réseau
- Des navigateurs clients

En d'autres termes, une application web est utilisée pour manipuler et traiter des informations, tandis qu'un site web est utilisé pour exposer ces informations.

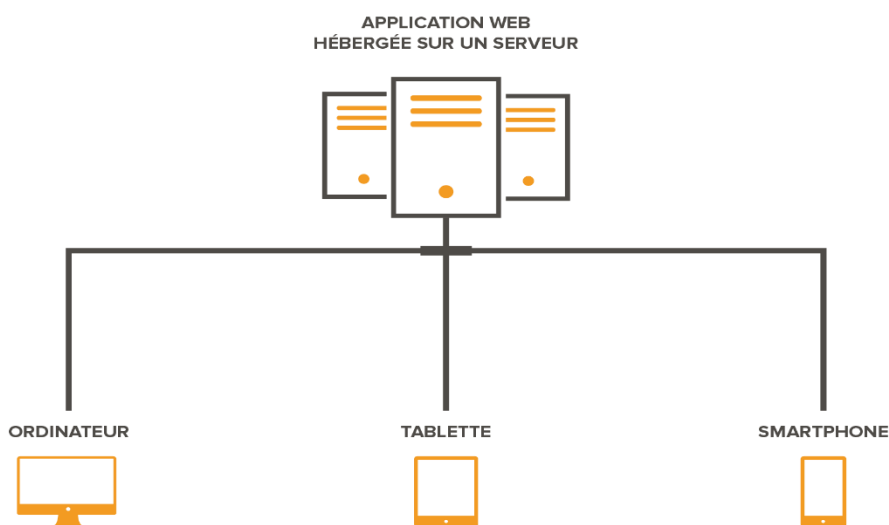


Figure 2: Application web hébergée sur un serveur [14]

4.1.1. Les types d'applications web : [15]

Nous allons fournir une brève définition pour chacun des cinq types d'applications web qui existent.

- Application web statique :

Une application web statique présente un contenu limité et reste généralement immuable. Elle est principalement développée en langage HTML et CSS, et peut inclure des éléments tels que des animations et des vidéos. Le développement d'applications web avec des technologies telles que jQuery et Ajax peut également être mentionné.

Cependant, modifier le contenu d'une application statique n'est pas une tâche facile. Cela nécessite d'éditer le code HTML et de mettre à jour les fichiers sur le serveur.

- Application web dynamique :

Les applications web dynamiques présentent une complexité technique supérieure, nécessitant l'utilisation de bases de données pour charger des informations actualisées à chaque accès de l'utilisateur. Plusieurs langages de programmation

sont disponibles pour le développement de telles applications, avec PHP et ASP en tête de liste en raison de leur capacité à bien structurer le contenu. Telle que Booking.com, TripAdvisor.

- Application web portail :

Ce genre d'application web offre aux utilisateurs la possibilité d'accéder à une page d'accueil comprenant diverses sections et catégories. Voici quelques exemples :(Forums/ Chats / Messagerie /électronique / Moteurs de recherche/Espace d'accès nécessitant une inscription)

- Application web E-Commerce :

Un portail Web est un service en ligne offrant diverses ressources pratiques telles que l'e-mail, les forums, les chats, le moteur de recherche et un accès sécurisé. Il agit comme une passerelle organisée facilitant l'accès aux informations disponibles sur Internet. Les portails Web offrent une apparence cohérente et conviviale, ainsi que des mécanismes de contrôle et d'accès pour utiliser plusieurs applications et bases de données de manière pratique.

- Application web avec gestionnaire de contenu :

La plupart des CMS (Content Management System) disponibles sur le marché sont des applications web payantes ou en Open source. Ils offrent une solution complète pour la gestion de tous les contenus d'un site web, qu'ils soient textuels, visuels ou sonores, y compris les aspects liés à l'ergonomie, à la mise en page et au web design. Ces gestionnaires sont connus pour leur simplicité d'utilisation et leur interface intuitive.

4.1.2. Architecture des applications web : [6]

Une architecture est un modèle générique décrivant la fonctionnalité, la structure et les relations entre les éléments d'un domaine spécifique, tels que le matériel, les logiciels et l'infrastructure. Les applications sont généralement divisées en trois niveaux d'abstraction : la couche présentation, qui est l'interface utilisateur visible dans un navigateur web ; la couche métier, qui met en œuvre la logique de l'application ; et la couche d'accès aux données, qui gère l'accès à la base de données. Différentes architectures web existent, allant de l'architecture 1-tiers à l'architecture n-tiers.

Chapitre 1 : Etude de L'existant

4.1.3. Serveur web : [6]

Un serveur web est un logiciel qui répond aux requêtes des navigateurs web en fournissant des pages web. Lorsqu'un utilisateur clique sur un lien sur une page web ou saisit une URL (Uniform Resource Locator) dans la barre d'adresse du navigateur, une demande de page est générée. Le serveur web reçoit cette demande et renvoie la page correspondante en réponse à la requête.

4.1.4. Le fonctionnement d'une application web : [7]

Pour une page web simple, le navigateur envoie une requête au serveur pour obtenir la page statique, qui est ensuite transmise au navigateur. En revanche, pour une page web dynamique, le serveur transfère cette page à un serveur d'application. Ce dernier lit le code de la page, exécute les instructions et génère une page statique qui est renvoyée au serveur web, puis transmise au navigateur. Le serveur d'application facilite également l'utilisation d'un serveur de base de données en traitant les balises contenant des requêtes de bases de données (comme le langage SQL) qui sont incluses dans le code HTML.

- ❖ La Figure 3 le processus d'interrogation de la base de données par le biais de requêtes, ainsi que la transmission des données résultantes au navigateur.

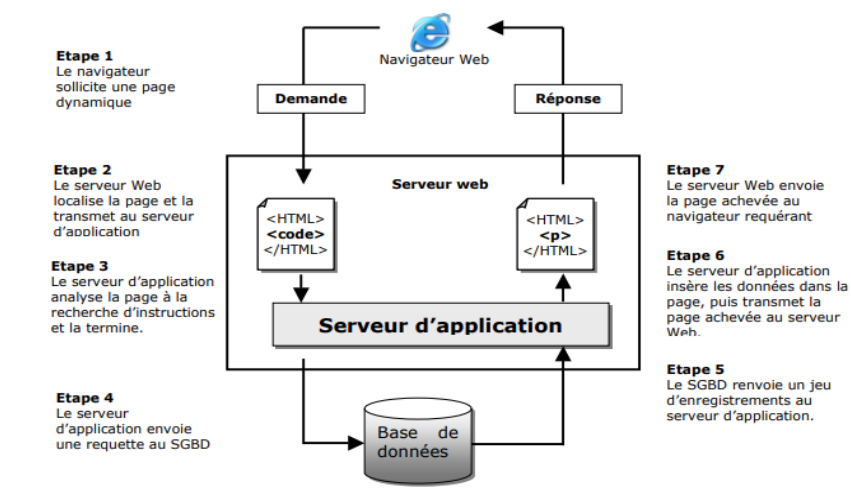


Figure 3: Fonctionnement d'une application web [7]

Une fois l'application réalisée et implémentée, une étape finale consiste généralement à l'héberger afin de la rendre accessible et utilisable par les utilisateurs. [7]

4.1.5. L'hébergement d'une application web : [6]

Pour rendre une application web accessible en ligne, elle doit être déployée sur un serveur connecté à Internet. Les hébergeurs web sont des sociétés spécialisées dans l'hébergement de sites et d'applications web sur des serveurs, offrant un déploiement rapide

et sécurisé. Ils assurent également la maintenance en cas de panne pour assurer la disponibilité continue de l'application.

❖ Il y a différents types d'hébergement, notamment :

- L'hébergement gratuit offre un service basique avec un espace de stockage limité et une fiabilité variable.
- L'hébergement payant propose un service de qualité supérieure avec un espace de stockage important, une grande fiabilité et un support technique réactif.

4.1.6. Les langages et les outils utilisés pour développer une application web : [17]

Pour réaliser et développer une application web, plusieurs langages et outils sont couramment utilisés. Voici quelques-uns d'entre eux :

❖ Langages de programmation :

- HTML (HyperText Markup Language) : utilisé pour définir la structure et le contenu des pages web.
- CSS (Cascading Style Sheets) : utilisé pour styliser et mettre en forme les éléments HTML.
- JavaScript : un langage de script côté client qui permet d'ajouter des fonctionnalités interactives et dynamiques aux pages web.
- PHP (Hypertext Preprocessor) : un langage de script côté serveur utilisé pour générer des pages web dynamiques.
- Python, Ruby, Java, C#, etc. : d'autres langages de programmation couramment utilisés pour le développement web.

❖ Frameworks et bibliothèques :

- Angular, React, Vue.js : des frameworks JavaScript populaires pour le développement d'applications web côté client.
- Node.js : un environnement d'exécution JavaScript côté serveur.
- Laravel, Django, Ruby on Rails : des frameworks web utilisés pour le développement d'applications web côté serveur dans leurs langages respectifs.

❖ Bases de données :

- MySQL, PostgreSQL, MongoDB : des systèmes de gestion de bases de données couramment utilisés dans le développement web.
- ORM (Object-Relational Mapping) : des outils qui facilitent l'interaction avec la base de données en utilisant des objets dans le code.

Chapitre 1 : Etude de L'existant

- ❖ Outils de développement :
 - Éditeurs de code : Visual Studio Code, Sublime Text, Atom, etc.
 - Contrôle de version : Git, SVN (Subversion).
- ❖ Environnements de développement intégrés (IDE) :
 - Eclipse, Visual Studio, IntelliJ IDEA, etc.
 - Outils de test : Jasmine, Jest, PHPUnit, etc.

4.1.7. Application de Gestion : [18]

Nous concevons des applications personnalisées qui répondent à vos besoins spécifiques. Cette solution offre de nombreux avantages, tels que la gestion des interventions, la hiérarchisation des tâches, un historique complet des clients, la centralisation des éléments clés (agenda, clients, interventions, e-mails, etc.) et l'amélioration de la productivité. Accessible depuis n'importe quel périphérique, l'application web garantit la sécurité grâce à des identifiants personnels, des sauvegardes multiples et une autorisation d'accès restreinte. Évolutive et basée sur des technologies open source, cette solution représente l'avenir des logiciels de gestion en entreprise

5. Conclusion :

Ce chapitre a permis de présenter l'organisme d'accueil, ainsi que les aspects liés à la planification des horaires dans un contexte universitaire. Il a souligné la complexité du problème de planification et l'importance d'adopter des approches spécifiques à chaque institution. De plus, les particularités du développement des applications web ont été mises en évidence, mettant en avant l'importance des méthodes de conception adaptées. Ces connaissances fournissent une base solide pour la suite de la recherche sur la gestion efficace des horaires dans le domaine pédagogique et l'utilisation des technologies web. Le prochain chapitre abordera l'analyse et la conception de notre application.

Chapitre 2 : **Analyse et conception**

1. Introduction :

Le développement de tout système d'information requiert une approche cruciale dans le cycle de vie d'un logiciel, et la modélisation pratique joue un rôle essentiel dans ces développements. Dans notre projet, nous explorerons la démarche de modélisation basée sur UML. Cette approche méthodologique fournit un cadre structuré pour guider la conception et l'analyse d'une application, et nous présenterons en détail cette démarche dans ce chapitre.

2. Présentation UML :

2.1. Définition : [8]

UML (Unified Modeling Language), qui peut être traduit en français par "Langage de Modélisation Unifié", est un langage de modélisation graphique utilisé dans le domaine de l'ingénierie logicielle. Il offre une manière standardisée de décrire les différentes composantes d'un système, leurs interactions et leurs structures.

UML propose un ensemble de diagrammes qui permettent de capturer les différentes perspectives d'un système, tels que le diagramme de classes, le diagramme de séquence, le diagramme d'états-transitions, le diagramme d'activités, etc. Chaque diagramme fournit une vue spécifique sur le système et permet de communiquer efficacement les informations entre les différents acteurs impliqués dans le processus de développement logiciel.

Donc, UML est un langage de modélisation graphique utilisé pour représenter et communiquer la conception des systèmes logiciels de manière claire et standardisée.

2.2. Historique : [18]

UML (Unified Modeling Language) est un langage de modélisation développé par l'OMG (Object Management Group) à partir des années 1990. Il est issu de la fusion de trois méthodologies de modélisation : Booch, OMT et OOSE. La première version d'UML, UML 1.0, est sortie en 1997, et elle comprenait des diagrammes tels que les diagrammes de classes, de cas d'utilisation, de séquence, d'état et d'activité. UML 2.0, publié en 2004, est la version la plus significative et la plus utilisée, introduisant des améliorations majeures et de nouveaux concepts. UML est largement utilisé dans l'industrie du logiciel pour la modélisation et la conception de systèmes logiciels,

fournissant un ensemble de diagrammes standardisés pour faciliter la communication entre les développeurs et les concepteurs.

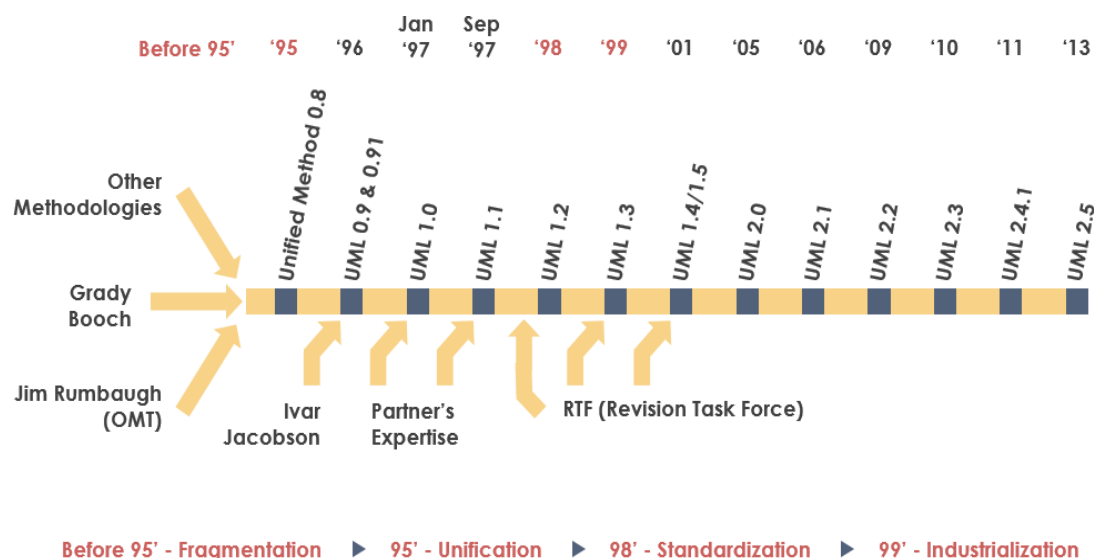


Figure 4 Historique UML [19]

2.3. Les Vues UML : [19]

La modélisation UML offre une variété de diagrammes pour représenter les différents aspects d'un système logiciel. En comprenant les trois points de vue de modélisation (structurel, comportemental et interaction), ainsi que les types de diagrammes associés à chacun, les concepteurs peuvent créer des modèles complets et précis. Nous allons explorer les trois points de vue de modélisation UML et classerons les types de diagrammes associés à chaque point de vue. Nous fournirons également des références pour approfondir chaque point de vue.

2.3.1. Point de vue structurel : [19]

Ce point de vue se concentre sur la structure statique du système, en mettant l'accent sur les classes, les objets, les relations entre les classes et les interfaces.

Les types de diagrammes couramment utilisés pour représenter ce point de vue sont les diagrammes de classes, les diagrammes d'objets, les diagrammes de composants et les diagrammes de déploiement.

2.3.2. Point de vue comportemental : [21]

Ce point de vue se concentre sur le comportement dynamique du système, en décrivant les interactions entre les objets et le flux d'activités à l'intérieur du système.

Les types de diagrammes couramment utilisés pour représenter ce point de vue sont les diagrammes de séquence, les diagrammes d'état, les diagrammes d'activité et les diagrammes de communication.

2.3.3. Point de vue interaction : [22]

Ce point de vue se concentre sur les échanges de messages entre les objets pour accomplir des tâches spécifiques et les collaborations entre les objets.

Le type de diagramme couramment utilisé pour représenter ce point de vue est le diagramme de séquence.

3. Présentation de notre démarche de modélisation : [23]

Notre démarche de modélisation comprend deux étapes essentielles : l'analyse et la conception. Dans la phase d'analyse, nous identifions les acteurs du système et examinons leurs interactions. Cette étape nous permet de comprendre les besoins et les exigences du système. Ensuite, lors de la phase de conception, nous créons des descriptions détaillées basées sur les résultats de l'analyse. Nous utilisons une représentation graphique de notre démarche de modélisation, comme illustrée dans la Figure 5 , pour guider le processus de conception de notre application. Cette approche nous permet de capturer les acteurs, les interactions et de fournir une vision claire des fonctionnalités à développer.

Cette représentation graphique illustre les différentes étapes de la démarche de modélisation, avec des flèches indiquant la progression linéaire du processus. Chaque étape est numérotée et décrit brièvement la tâche à accomplir. Cette visualisation permet de mieux comprendre la séquence des étapes et l'enchaînement logique de la démarche de modélisation.

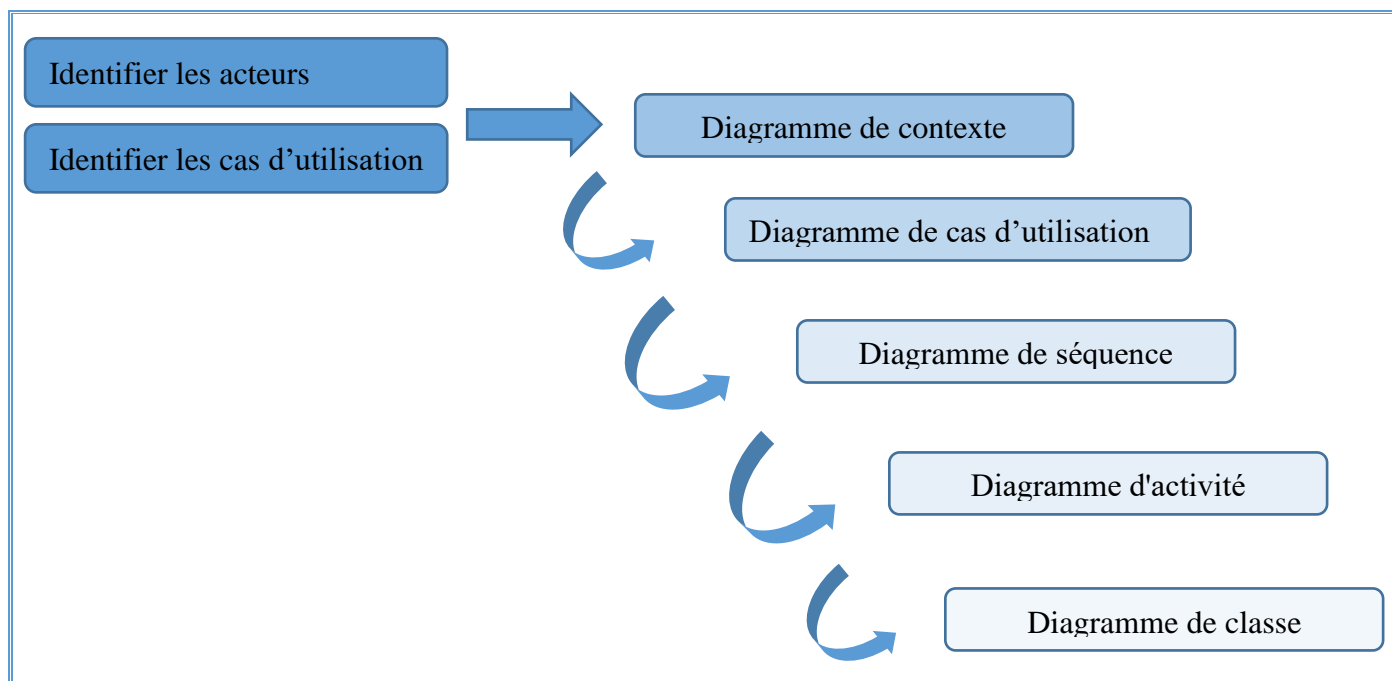


Figure 5 Démarche de modélisation de l'application

4. Analyse : [9]

L'objectif de cette phase est de spécifier clairement l'application. Pour ce faire, il est essentiel de déterminer de manière globale ce qui est inclus dans le champ de l'application. Dans cette phase, nous nous concentrons sur l'identification des acteurs du système, leurs rôles et le contexte de l'application.

4.1. Identification des acteurs : [10]

Les acteurs sont des entités externes qui interagissent avec le système étudié. Ils peuvent être des utilisateurs, des systèmes externes ou d'autres entités qui ont un rôle à jouer dans le fonctionnement du système. L'identification des acteurs implique d'identifier leurs caractéristiques, leurs besoins et leurs interactions avec le système, afin de comprendre leur impact sur la conception et les fonctionnalités du système. Ce processus est essentiel pour une modélisation précise et complète du système étudié.

Dans notre système les acteurs sont :

- **L'administrateur (l'acteur principal)** : qui gère toutes les fonctionnalités de notre application.
- **Enseignant** : qui utilise notre application et consulte le planning.
- **Visiteur (soit étudiant ou autre utilisateur)** : qui consulte le planning, contacter l'admin.

4.2. Identification des cas d'utilisation : [24]

Les cas d'utilisation sont des entités autonomes qui représentent des fonctionnalités identifiées depuis l'extérieur du système. Chaque cas d'utilisation encapsule un service complet avec un déclencheur, une séquence d'actions et une finalité pour l'acteur qui l'initie. Ils modélisent les services fournis par le système, sans imposer de contraintes sur la manière dont ces services sont réalisés concrètement.

Une fois les acteurs identifiés, nous pouvons associer des cas d'utilisation à chacun d'eux, tels que l'administrateur, l'enseignant et le visiteur, afin de recenser toutes les fonctionnalités nécessaires pour répondre à leurs besoins. Cette approche nous permet de créer un tableau récapitulatif des cas d'utilisation spécifiques à chaque acteur de notre application web.

Chapitre 2 : Analyse et conception

Acteur	Cas d'utilisation
Administrateur	<ul style="list-style-type: none"> • Authentification • Gérer son profile: <ul style="list-style-type: none"> – Ajouter (comme : photo de profile/information), modifier (son mot passe/email/numéro téléphone) ou supprimer. • Gérer les contacts: <ul style="list-style-type: none"> – Consulter. – Traiter les messages. • Gérer les semestres /les modules/ les groups/ les section/ les spécialités : <ul style="list-style-type: none"> – Consulter, Ajouter, modifier ou supprimer. • Gérer les sales et Amphis : <ul style="list-style-type: none"> – Consulter, Ajouter, rechercher, modifier ou supprimer. • Gérer les enseignantes: <ul style="list-style-type: none"> – Consulter, Ajouter, rechercher, modifier ou supprimer. • Gérer le planning <ul style="list-style-type: none"> – Consulter, créer, imprimer, modifier ou supprimer • Se déconnecter
Enseignant	<ul style="list-style-type: none"> – S'authentifier. – Consulter son planning – Imprimer le planning – Demander de change Séance – Gérer son profile – Ajouter (comme : photo de profile/information), modifier (son mot passe/email/numéro téléphone) ou supprimer – Se déconnecter.
Visiteur	<ul style="list-style-type: none"> – Consulter le planning. – Imprimer le planning – Contacter l'admin.

Tableau 1 Identification des cas d'utilisation

4.3. Diagramme de contexte : [25]

Le diagramme de contexte joue un rôle essentiel en délimitant le périmètre de l'étude. Il positionne le diagramme dans son contexte en identifiant les acteurs ou éléments qui interagissent avec lui. Le niveau de détail de ce diagramme dépend de l'objectif de la représentation. Il vise à répondre à la question : "Quels sont les acteurs et les éléments qui entourent le système ?".

Nous répondrons à ces questions à l'aide du diagramme suivant où chaque acteur interagit directement avec l'application web afin d'accéder aux fonctionnalités spécifiques à son rôle.

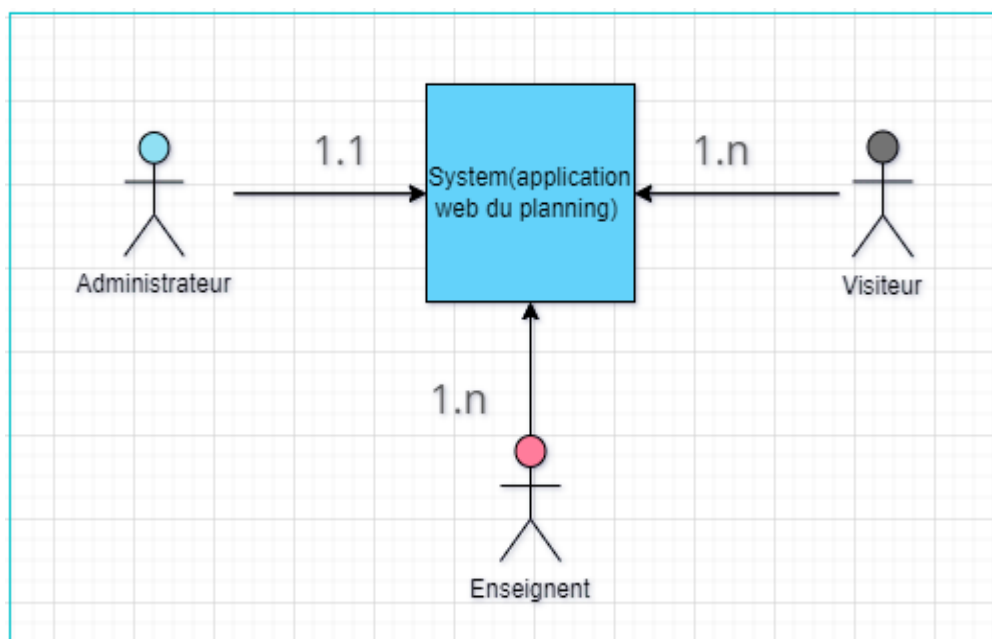


Figure 6 Diagramme de contexte

4.4. Diagramme de cas d'utilisation : [11]

Diagramme de cas d'utilisation représente les interactions entre les acteurs (utilisateurs, systèmes externes) et le système étudié. Il met en évidence les fonctionnalités offertes par le système du point de vue des utilisateurs. Ainsi, il permet de définir les fonctionnalités que le système devra fournir, sans entrer dans les détails de leur implémentation.

- ❖ UML propose trois types de relations standardisées entre les cas d'utilisation, qui sont les suivantes : [26]

L'inclusion, représentée par le mot-clé "`<<include>>`", permet d'exprimer qu'un cas d'utilisation inclut un autre cas d'utilisation.

L'extension, représentée par le mot-clé "`<<extend>>`", permet de spécifier qu'un cas d'utilisation peut être étendu par l'ajout d'autres fonctionnalités optionnelles.

La généralisation/spécialisation est une relation qui permet de définir une hiérarchie entre les cas d'utilisation, où un cas d'utilisation général peut être spécialisé par des cas d'utilisation plus spécifiques.

4.4.1. Description des cas d'utilisation avec des scénarios :

Dans cette section, nous fournissons une description approfondie de chaque cas d'utilisation, en définissant les différents scénarios possibles. L'objectif est d'explorer en détail les différentes actions et interactions qui peuvent se produire dans chaque cas d'utilisation.

Chapitre 2 : Analyse et conception

❖ Description de cas d'utilisation « Authentification » :

Acteurs:	Utilisateur soit: <ul style="list-style-type: none"> - L'administrateur - L'enseignant
Objectifs:	<ul style="list-style-type: none"> - Permettre aux utilisateurs de s'identifier et d'accéder à son espace - Assurer la sécurité et la confidentialité des informations en ne permettant l'accès qu'aux utilisateurs autorisés.
Scénario nominal:	<ol style="list-style-type: none"> 1. L'utilisateur ouvre l'application web 2. Le système affiche l'écran de connexion 3. L'utilisateur saisit son nom d'utilisateur et son mot de passe. 4. Le système effectue une vérification pour s'assurer que les champs ne sont pas vides. Si une erreur est détectée Alternative [A1]. 5. Le system vérifie les informations d'identification fournies par l'utilisateur. Si les informations sont valides, Le system authentifie l'utilisateur et lui permet d'accéder à son espace. Sinon alors Alternative [A2]. 6. Si l'utilisateur a oublié son mot de passe Alternative [A3].
Scénario nominal	<ul style="list-style-type: none"> - Alternative [A1] : L'utilisateur procède à une deuxième tentative d'identification. Le system Affiche « Un champ est laissé vider ». Veuillez-vous identifier à nouveau." Revenez au scénario nominal, étape 3. - Alternative [A2] : la réinitialisation du mot de passe. - Alternative [A3] : Récupération de mot de passe oublié : Clic sur "Mot de passe oublié". Saisie de l'adresse e-mail. Réception d'un e-mail avec un lien de réinitialisation. Clic sur le lien et saisie d'un nouveau mot de passe.
Préconditions	L'application web est opérationnelle et accessible. Les utilisateurs sont enregistrés avec des identifiants valides. Les rôles et les autorisations des utilisateurs sont configurés correctement.
Post condition	L'utilisateur est authentifié par le système

Tableau 2 Cas d'utilisation « Authentification »

Chapitre 2 : Analyse et conception

❖ Description de cas d'utilisation « gérer les enseignants » :

Acteurs	L'administrateur
Objectifs	Permettre à l'administrateur de gérer les enseignants de manière efficace dans web
Scénario nominal	L'administrateur se connecte à l'application web. L'administrateur accède à la fonctionnalité de gestion des enseignants. Le system affiche la liste des enseignants existants. L'administrateur choisit une des actions suivantes : Consulter un enseignant Ajouter un enseignant Rechercher un enseignant Modifier un enseignant Supprime un enseignant L'application enregistre les modifications apportées à la liste des enseignants
Préconditions	L'administrateur est connecté à l'application en tant qu'utilisateur ayant les droits d'administration

Tableau 3 Cas d'utilisation « gérer les enseignants »

❖ Description des cas d'utilisation relative à l'administrateur :

Cas D'utilisation	Acteurs	L'administrateur
Cas d'utilisation « Ajouter un enseignant »	Objectives	Permettre à l'administrateur d'ajouter un nouvel enseignant dans l'application web.
	Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande d'ajouter un enseignant. 2. Le système affiche un formulaire vide pour saisir les informations de l'enseignant 3. L'administrateur remplit le formulaire en fournissant les détails requis tels que le nom, le prénom, l'adresse e-mail, etc. formulaire et valide. Si enseignant existe déjà Alternative [A1]. 4. Le système vérifie les informations remplis 5. Si toutes les champs complètes et les informations sont valides, sinon Alternative [A2]. 6. Le système affiche un message de confirmation. 7. L'administrateur confirme l'ajout. Sinon Alternative [A3]. 8. Le système enregistre les informations d'enseignant à la liste des enseignants existants. 9. Le système affiche un message « Enseignant ajouté ».

Chapitre 2 : Analyse et conception

	Scénario alternative	<ul style="list-style-type: none"> – Alternative [A1] : Le système informe l'administrateur que l'enseignant existe. Retourner au scénario nominal au point 2 – Alternative [A2] : Le système affiche un message d'erreur indiquant "Veuillez vérifier les champs, ils sont incomplets ou incorrects" Revenez au scénario nominal, étape 2, et signalez les champs concernés. – Alternative [A3]: L'administrateur annule l'opération Retourner au scénario nominal au point 2.
	Pré-conditions	L'administrateur est authentifié.
	Post condition	Nouvelle enseignant sera ajouté par le système.
Cas d'utilisation « Modifier un enseignant »	Objectives	Permettre à l'administrateur de modifier les informations d'un enseignant existant
	Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande la modification des informations de l'enseignant. 2. Le système affiche un formulaire pré-rempli avec les informations actuelles de l'enseignant. 3. L'administrateur modifie les informations nécessaires de l'enseignant (nom, prénom, adresse e-mail, etc.) sur les champs et valide. 4. Le système effectue des vérifications 5. Si toutes les informations sont valides Le system affiche un message de confirmation. Sinon Alternative [A2] 6. L'administrateur valide l'opération 7. Le système met à jour les informations de l'enseignant dans la liste et affiche un message de succès.
	Pré-conditions	<ul style="list-style-type: none"> – L'administrateur est connecté. – Il existe au moins un enseignant dans la liste des enseignants.
	Post condition	<ul style="list-style-type: none"> – Les modifications sont enregistrées par le système. – La liste des enseignants est mise à jour avec les nouvelles informations.
Cas d'utilisation « Supprimer un Enseignant »	Objectives	Permettre à l'administrateur de supprimer un enseignant de la liste des enseignants
	Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande la suppression d'un enseignant. 2. Le système affiche la liste des enseignants existants. 3. L'administrateur sélectionne l'enseignant qu'il souhaite supprimer de la liste. 4. Le système affiche une confirmation de la suppression de l'enseignant. 5. L'administrateur confirme la suppression. Sinon Alternative [A3] 6. Le système supprime l'enseignant de la liste.

Chapitre 2 : Analyse et conception

	Pré-conditions	<ul style="list-style-type: none"> - Authentification
	Post condition	<ul style="list-style-type: none"> - L'enseignant sélectionné est supprimé avec succès de la liste des enseignants dans l'application. - La liste des enseignants est mise à jour sans la présence de l'enseignant supprimé.
Cas d'utilisation « Rechercher un enseignant »	Objectives	<ul style="list-style-type: none"> - Permettre à l'utilisateur de rechercher un enseignant Spécifique dans l'application
	Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande de rechercher un enseignant 2. Le système affiche une liste d'enseignant et affiche un champ de recherche où l'utilisateur peut saisir le nom, ID, 3. L'administrateur saisit les informations de recherche dans le champ correspondant. 4. Le système effectue la recherche en fonction des critères fournis et le sélectionne. Sinon Alternative [A1] 5. L'administrateur valide l'opération 6. Le système affiche les résultats de la recherche, qui correspondent aux enseignants correspondant aux critères de recherche.
	Pré-conditions	<ul style="list-style-type: none"> - Authentification
	Post condition	<ul style="list-style-type: none"> - Les résultats de la recherche sont affichés à L'administrateur. - L'administrateur peut consulter les informations des enseignants correspondant aux critères de recherche.
Cas d'utilisation « Création de planning »	Objectives	<ul style="list-style-type: none"> - Permettre à l'administrateur de créer un planning pour le département de chaque spécialité.
	Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur demande de créer un planning. 2. Le système affiche une interface permettant à l'administrateur de spécifier les paramètres du planning tels que la période, les horaires, les jours de la semaine, la salle etc. 3. L'administrateur remplit les cases de planning. 4. L'administrateur valide l'opération. 5. Le système génère automatiquement le planning en utilisant les paramètres spécifiés. Sinon Alternative [A2]. 6. Le système affiche le planning généré à l'administrateur.
	Pré-conditions	<ul style="list-style-type: none"> - L'administrateur doit être connecté (Authentification).
	Post condition	<ul style="list-style-type: none"> - Le planning est créé avec succès. - Le planning généré est affiché à tous les utilisateurs. - Le planning prêt à télécharger.

Tableau 4: Cas d'utilisation relative à l'administrateur

❖ Remarque :

Les descriptions des cas d'utilisation "Gérer section" et "Gérer semestre" et "Gérer salle" suivent le même fonctionnement que la description du cas d'utilisation "Gérer les enseignants" etc.

4.4.2. Diagramme de Cas d'utilisation Authentification :

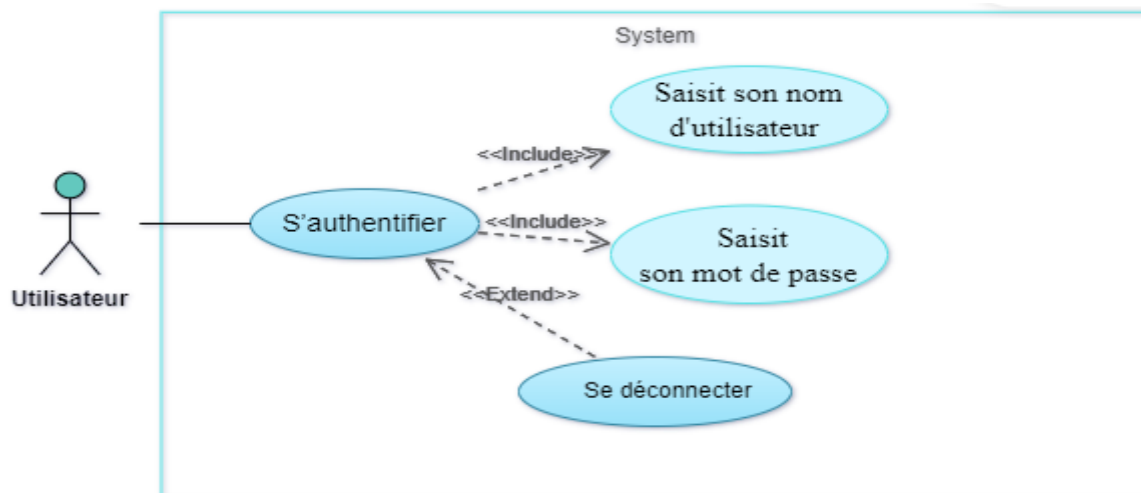


Figure 7: Diagramme de Cas d'utilisation Authentification

4.4.3. Diagramme de cas d'utilisation « Héritage » :

Dans ce diagramme, nous avons l'acteur "Utilisateur" comme acteur de plus haut niveau, représentant tous les utilisateurs de l'application. Le cas d'utilisation "Consulter Planning" est partagé par tous les types d'utilisateurs

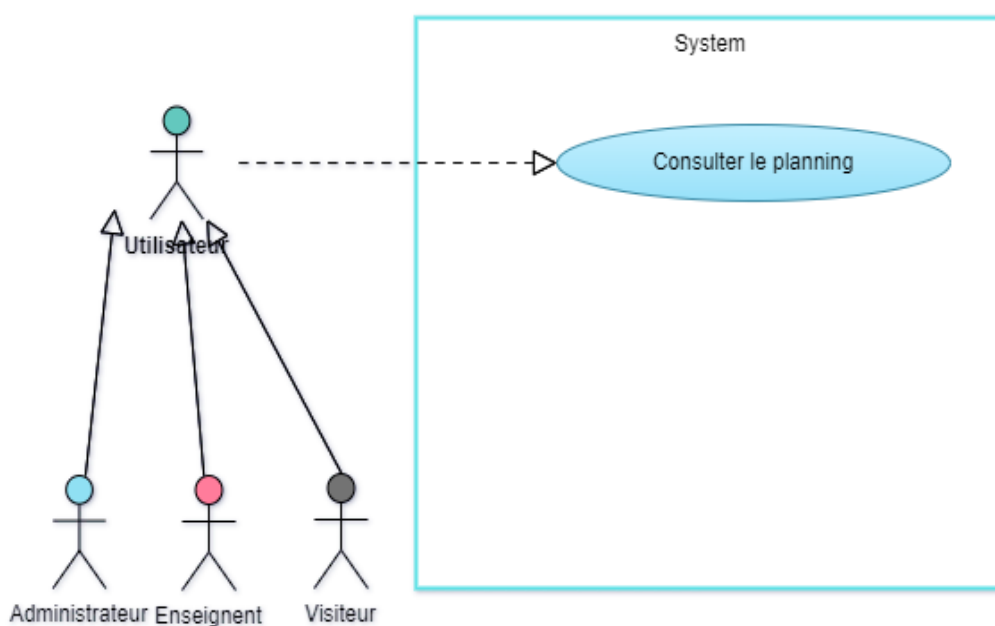


Figure 8: Diagramme de cas d'utilisation « Héritage »

Chapitre 2 : Analyse et conception

4.4.4. Diagramme de cas d'utilisation concernant l'administrateur :

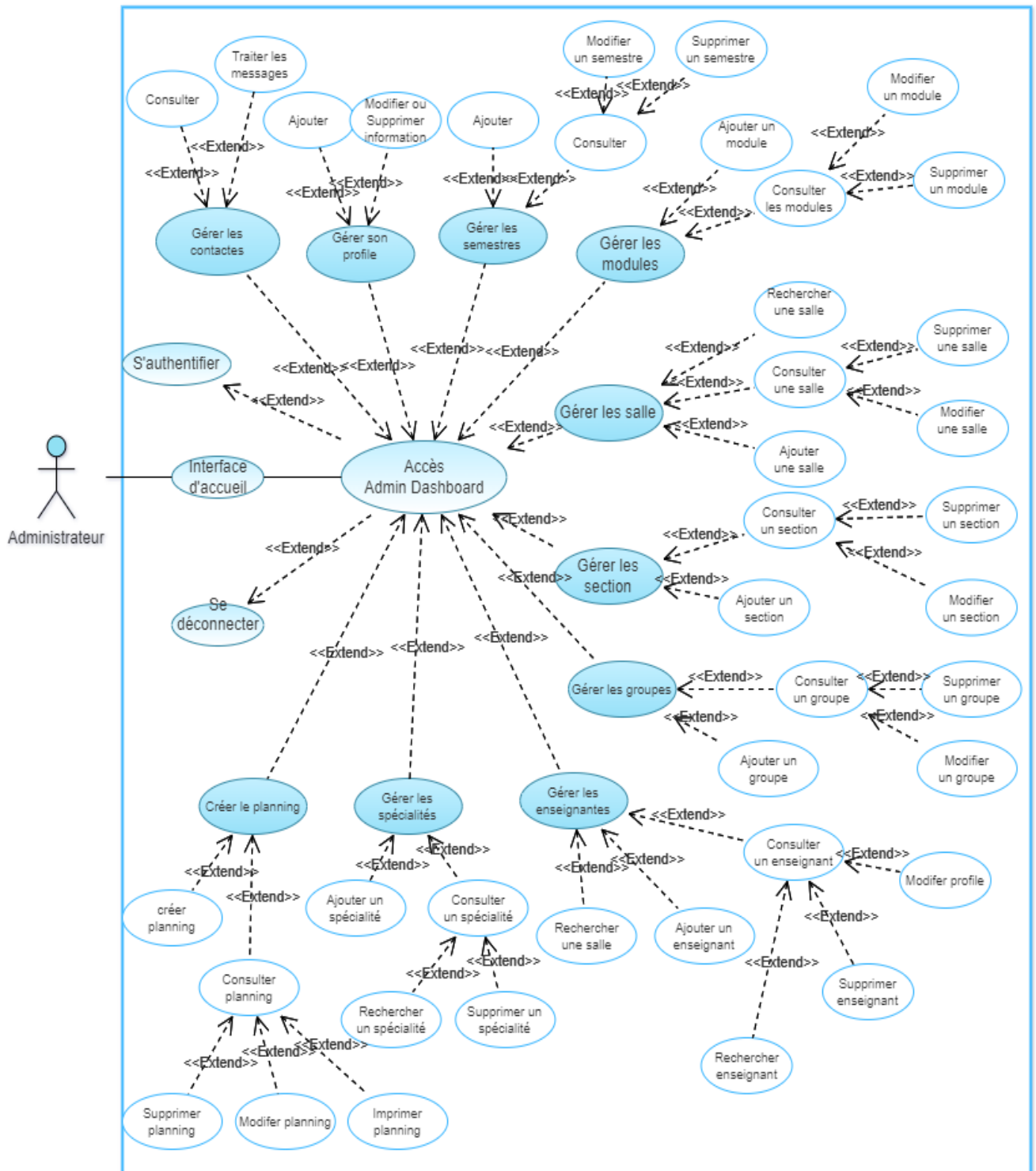


Figure 9: Diagramme de cas d'utilisation concernant l'administrateur

Chapitre 2 : Analyse et conception

4.4.5. Diagramme de cas d'utilisation concernant l'enseignant :

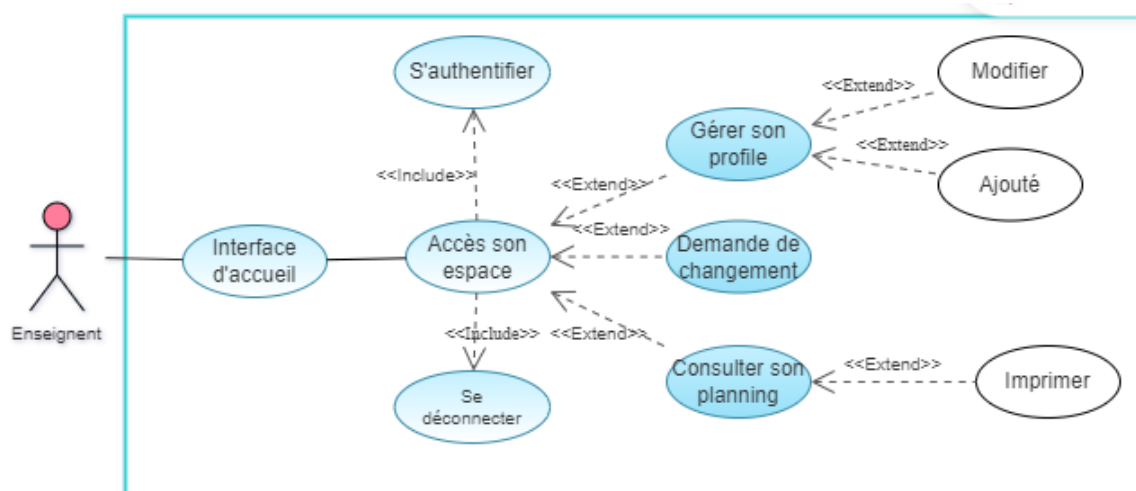


Figure 10: Diagramme de cas d'utilisation concernant l'enseignant

4.4.6. Diagramme de cas d'utilisation concernant le visiteur :

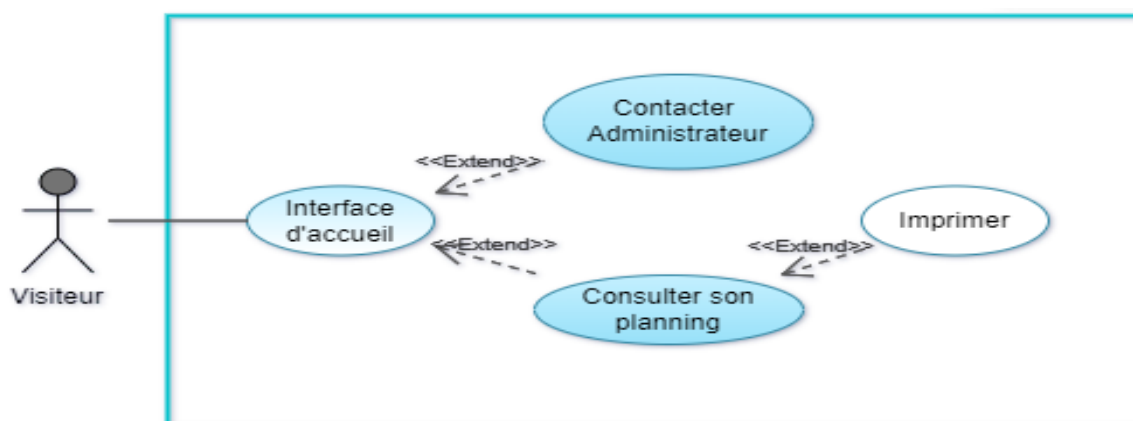


Figure 11: Diagramme de cas d'utilisation concernant le visiteur

4.4.7. Diagramme de cas d'utilisation génération de planning :

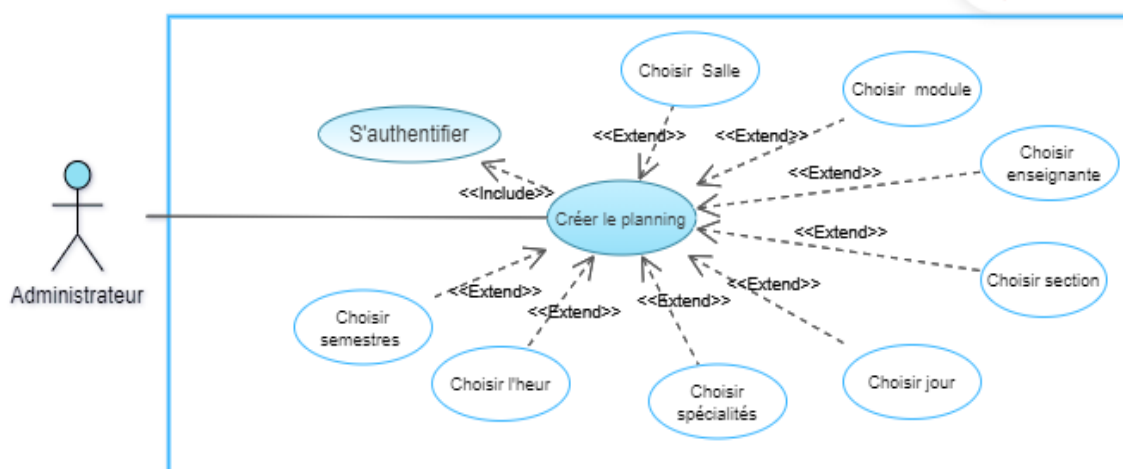


Figure 12: Diagramme de cas d'utilisation génération de planning.

4.5. Diagramme de séquence : [12]

Le diagramme de séquence est un diagramme d'interaction UML qui permet de visualiser les échanges de messages entre les objets d'un système dans un scénario particulier. Il est largement utilisé dans le domaine de l'analyse et de la conception logicielle pour représenter les interactions dynamiques entre les différentes composantes d'un système. Il facilite la communication entre les parties prenantes et permettent de mieux comprendre le fonctionnement et les scénarios d'utilisation d'un système logiciel.

Le diagramme de séquence utilise des "lifelines" pour représenter les objets et des flèches pour les messages échangés, offrant ainsi une représentation graphique claire du flux de contrôle et de communication dans un système. Il permet de visualiser l'ordre chronologique des interactions et d'appréhender la collaboration entre les objets pour atteindre une fonctionnalité spécifique.

4.5.1. Diagramme de séquence de cas d'utilisation « Authentification » :

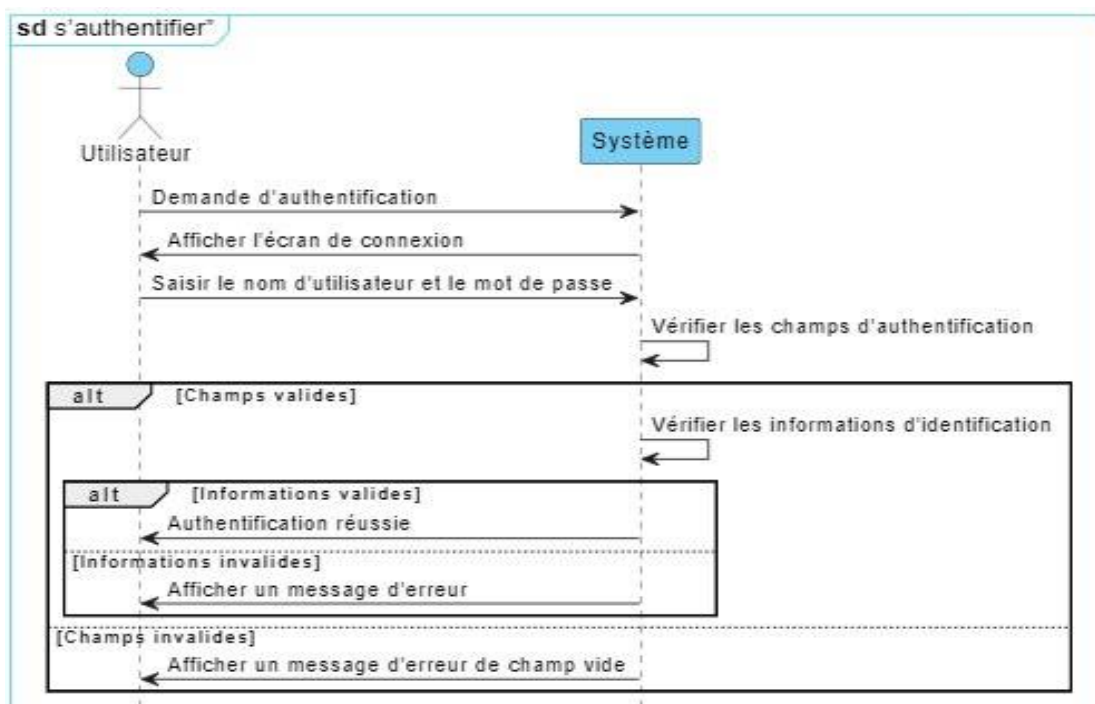


Figure 13: Diagramme de séquence de cas d'utilisation « Authentification »

Chapitre 2 : Analyse et conception

4.5.2. Diagramme de séquence de cas d'utilisation « Modifier un enseignant » :

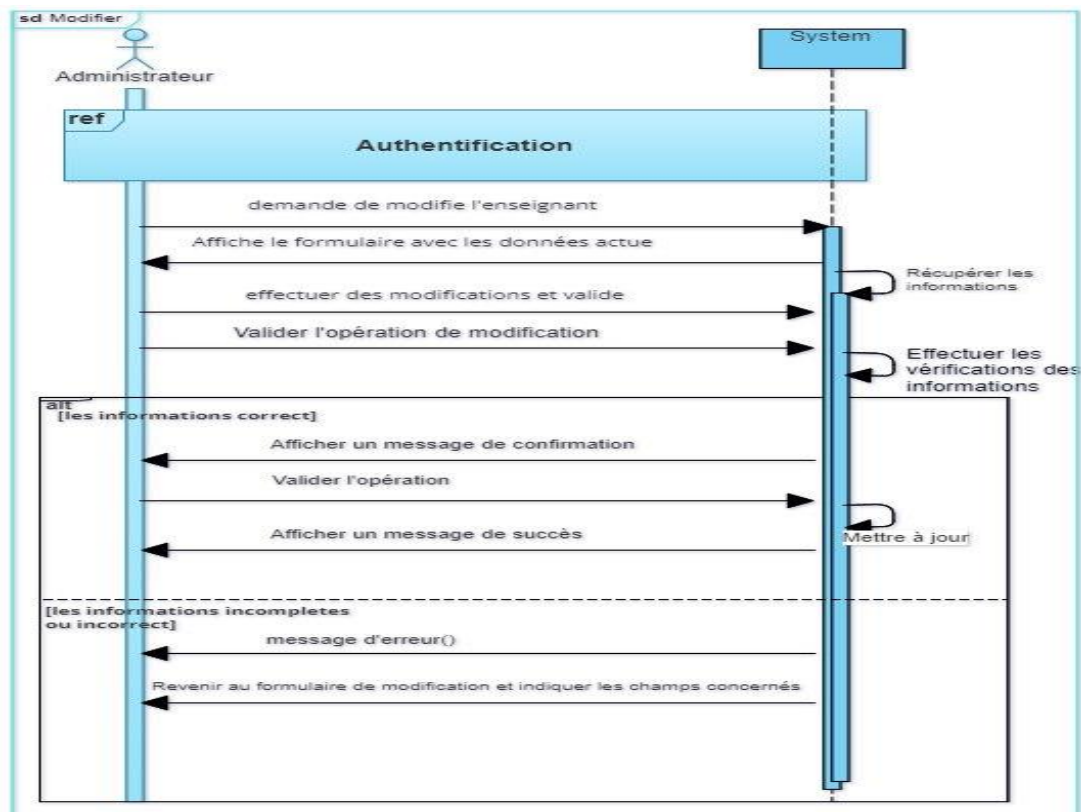


Figure 14: Diagramme de séquence de cas d'utilisation « Modifier un enseignant »

4.5.3. Diagramme de séquence de cas d'utilisation « Ajouter enseignant » :

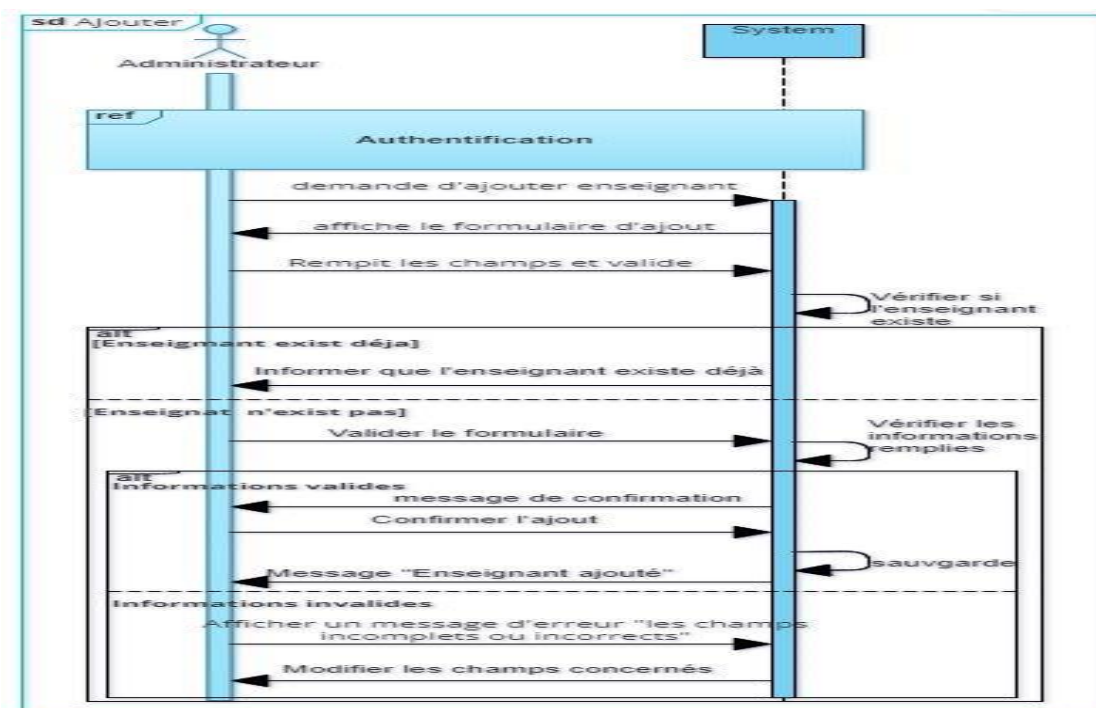


Figure 15: Diagramme de séquence de cas d'utilisation « Ajouter enseignant »

Chapitre 2 : Analyse et conception

4.5.4. Diagramme de séquence de cas d'utilisation « supprimer enseignant » :

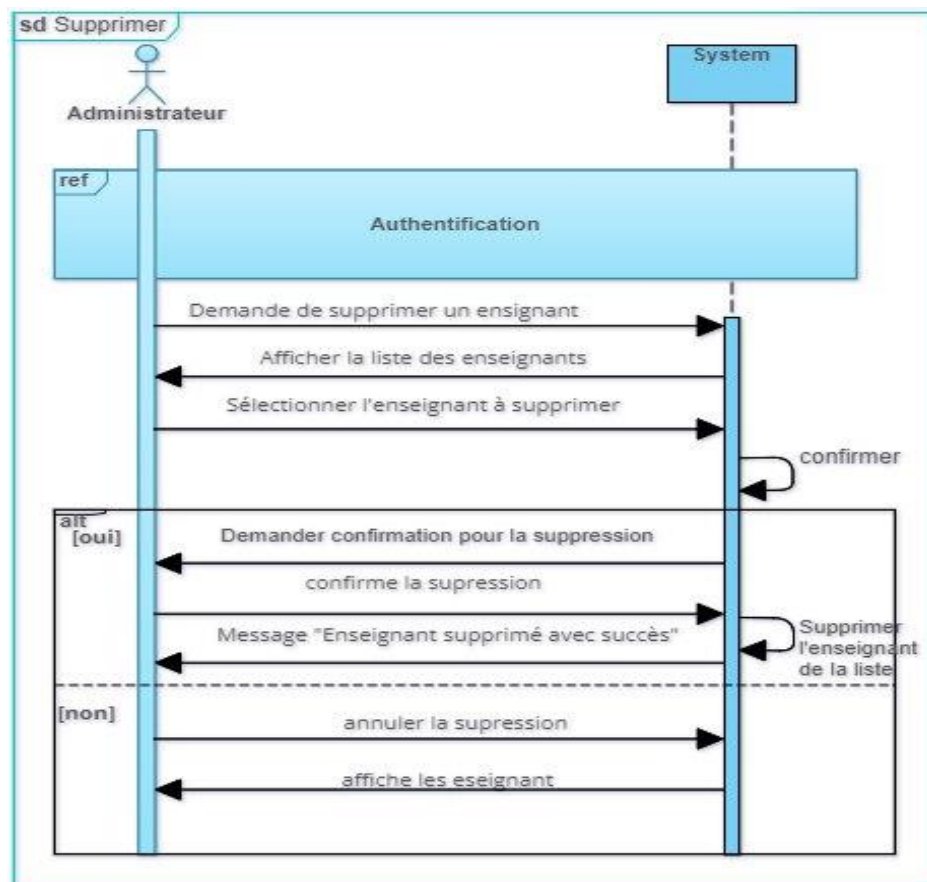


Figure 16! Diagramme de séquence de cas d'utilisation « supprimer enseignant »

4.5.5. Diagramme de séquence de cas d'utilisation « Rechercher un enseignant » :

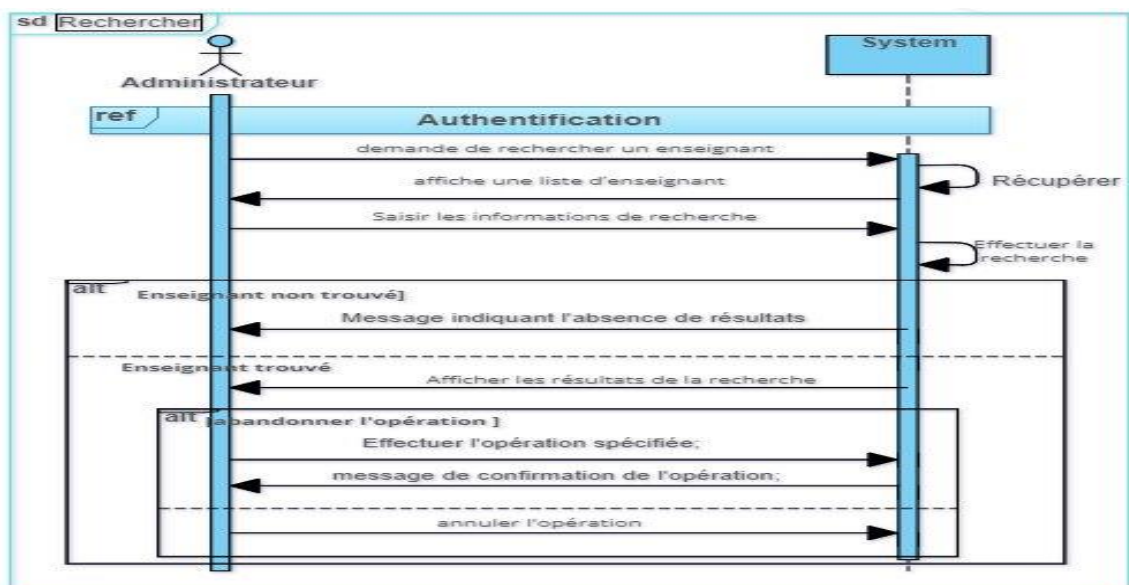


Figure 17: Diagramme de séquence de cas d'utilisation « Rechercher un enseignant »

4.5.6. Diagramme de séquence de cas d'utilisation « Création de planning » :

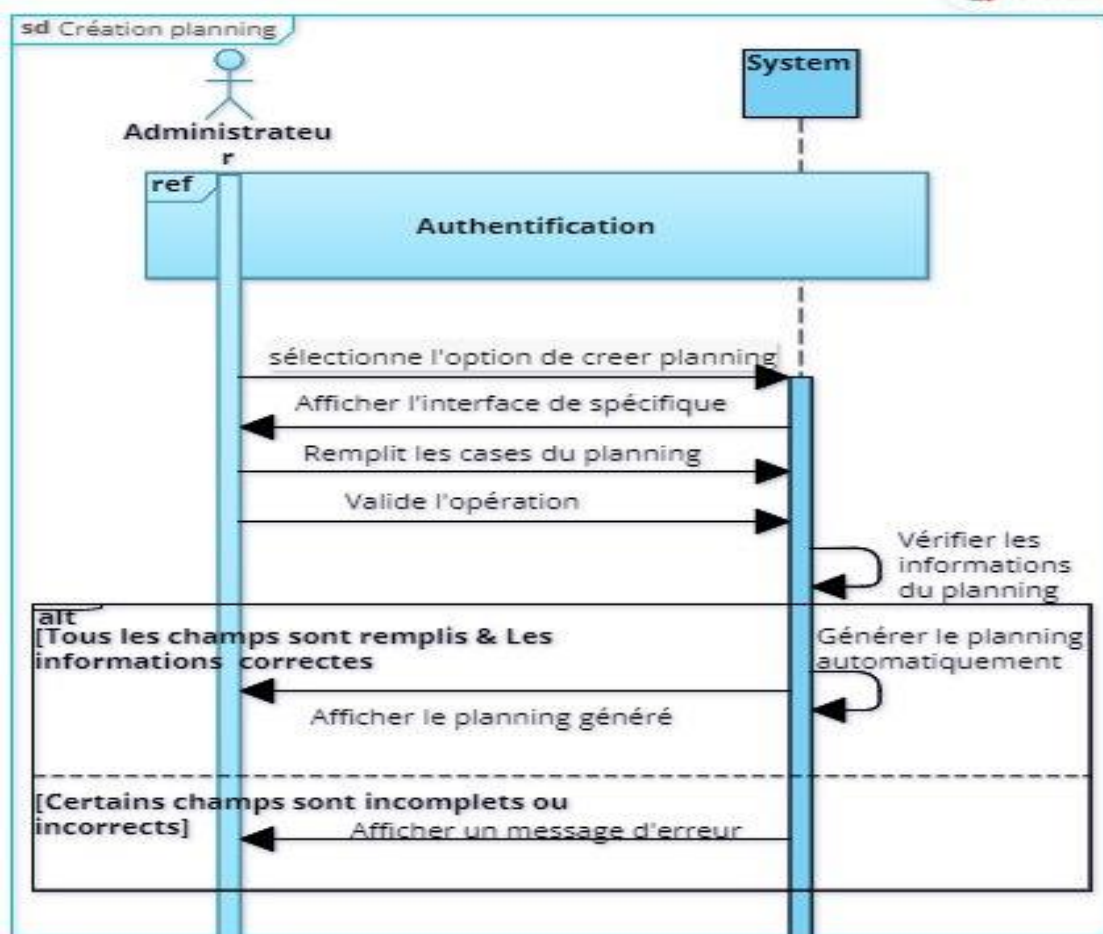


Figure 18: Diagramme de séquence de cas d'utilisation « Création de planning

5. Conception

La phase de conception est essentielle pour concrétiser le modèle d'analyse en une solution logicielle fonctionnelle. Elle comprend la conception de l'application et de la base de données, en tenant compte de vos spécifications personnalisées afin d'assurer le succès du projet.

5.1. Diagrammes d'activité : [24]

Les diagrammes d'activités sont des outils de modélisation qui se concentrent sur les traitements en représentant graphiquement les flux de contrôle et de données. Ils sont utilisés pour visualiser le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Contrairement aux diagrammes d'états-transitions qui sont spécifiques aux systèmes réactifs, les diagrammes d'activités peuvent être attachés à n'importe quel élément de modélisation pour représenter, spécifier, construire ou documenter son comportement. La principale différence avec les diagrammes d'interaction est que les diagrammes d'interaction mettent en évidence le flux de contrôle d'un objet à un autre, tandis que les diagrammes d'activités mettent l'accent sur le flux de contrôle d'une activité à une autre.

Chapitre 2 : Analyse et conception

5.1.1. Diagramme d'activité du cas d'utilisation « S'authentifier » :

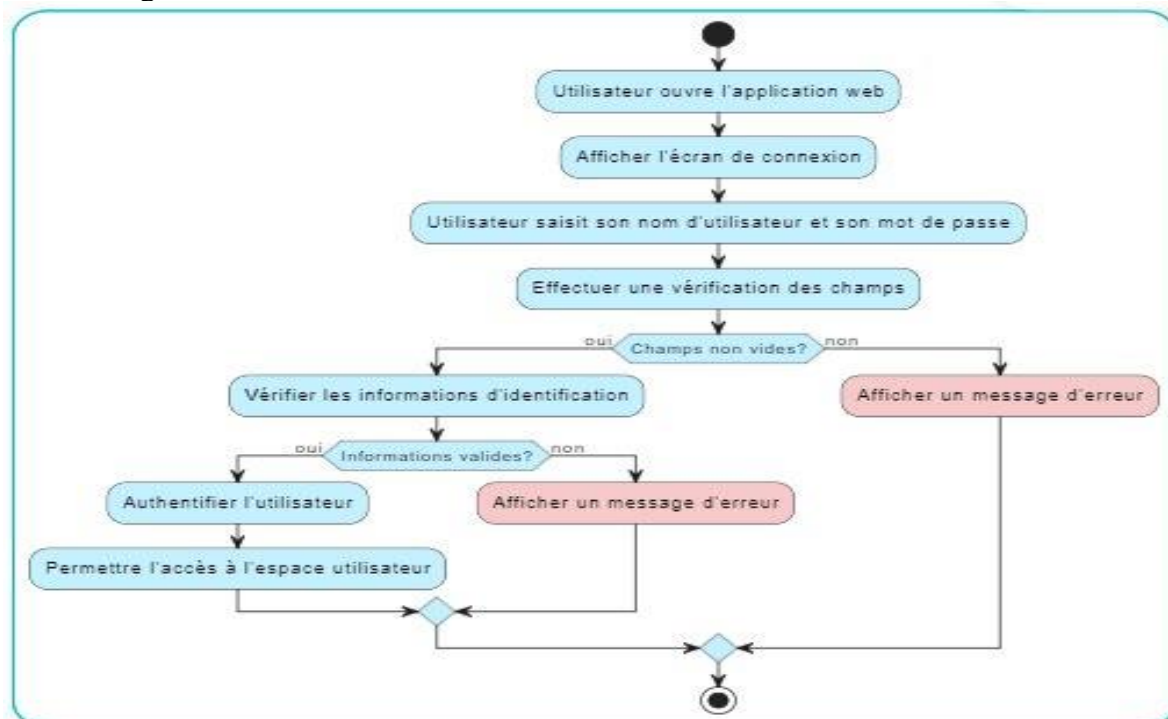


Figure 19: Diagramme d'activité du cas d'utilisation « S'authentifier »

5.1.2. Diagramme d'activité du cas d'utilisation « Modification un enseignant »

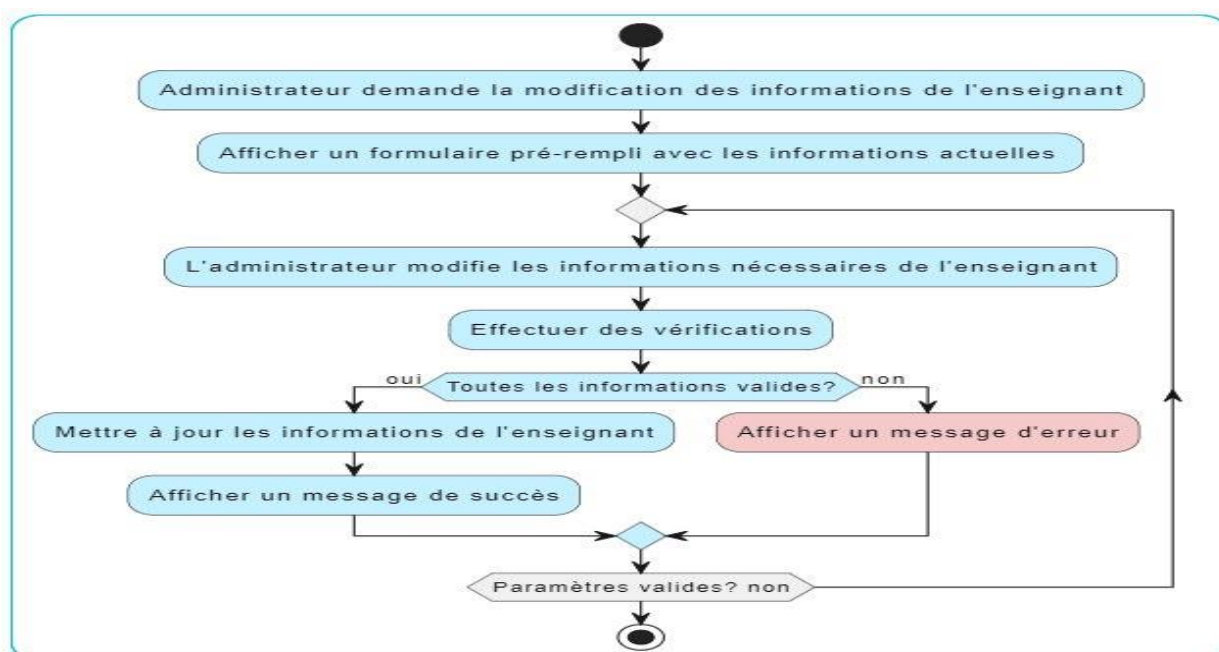


Figure 20: Diagramme d'activité du cas d'utilisation « Modification un enseignant »

Chapitre 2 : Analyse et conception

5.1.3. Diagramme d'activité du cas d'utilisation « Rechercher un enseignant »

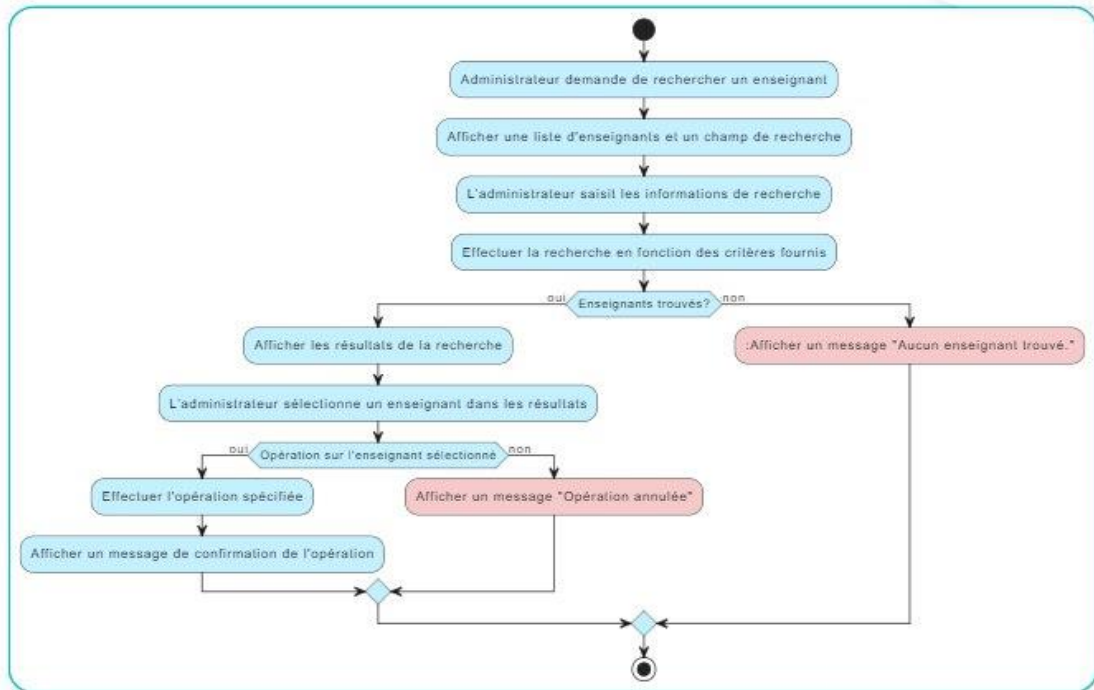


Figure 21 Diagramme d'activité du cas d'utilisation « Rechercher un enseignant

5.1.4. Diagramme d'activité du cas d'utilisation « Suppression d'un Enseignant » :

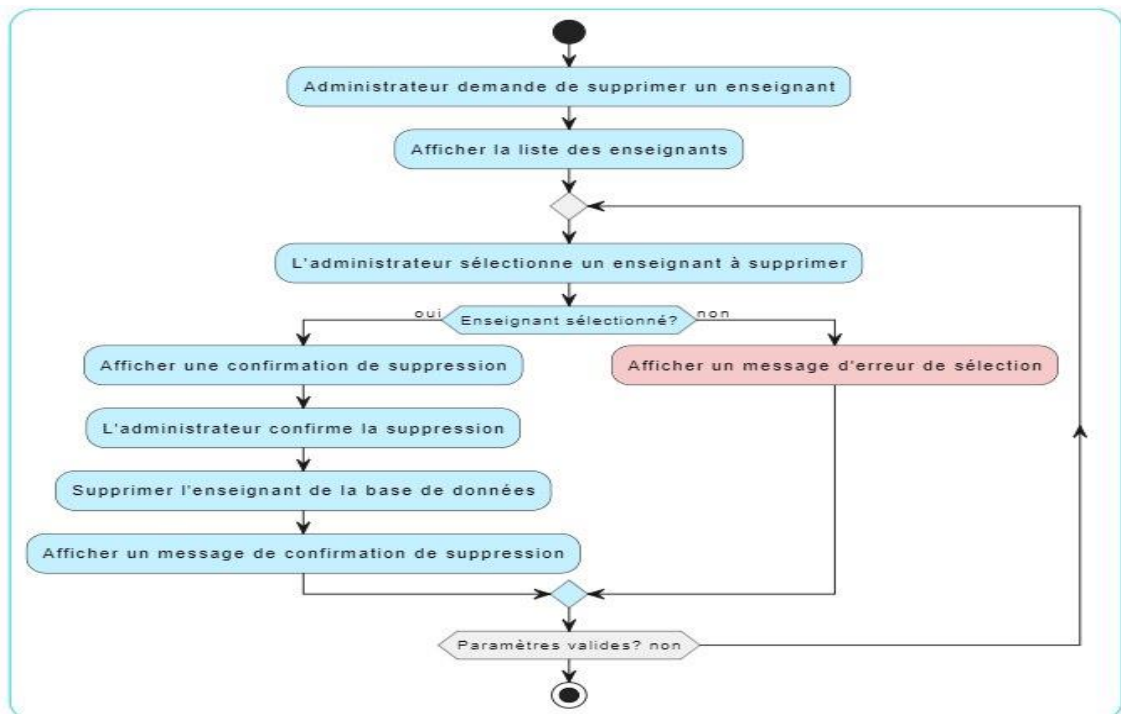


Figure 22 : Diagramme d'activité du cas d'utilisation « Suppression d'un Enseignant

Chapitre 2 : Analyse et conception

5.1.5. Diagramme d'activité du cas d'utilisation « Ajouter Enseignant »

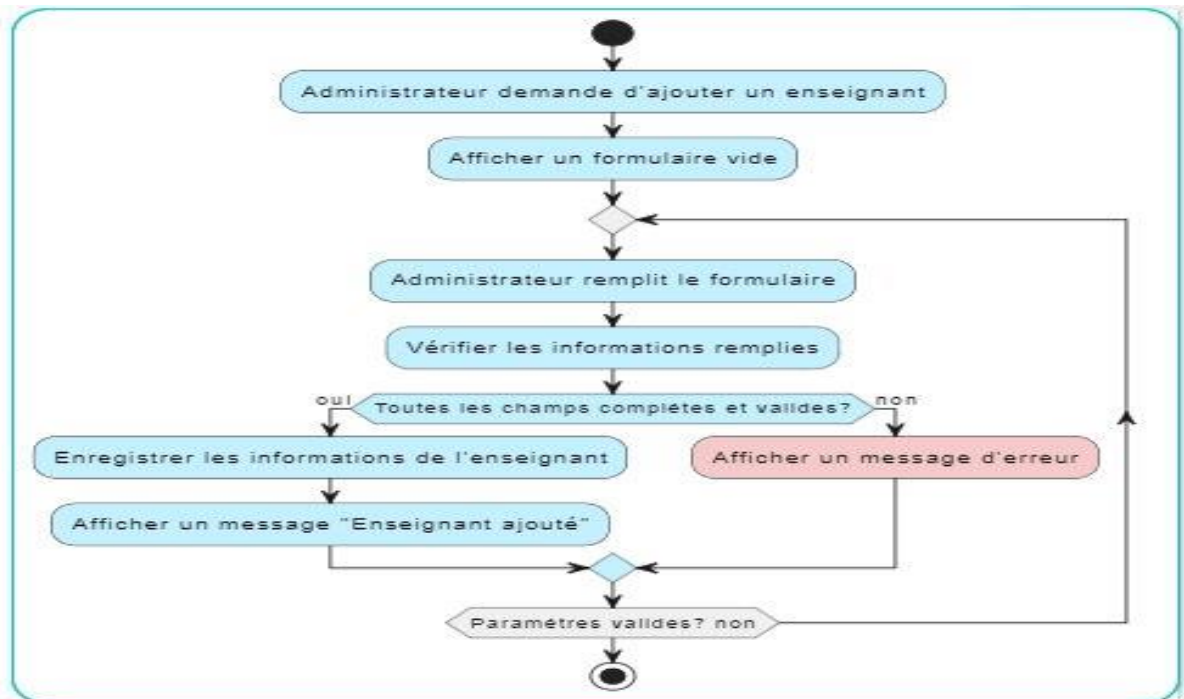


Figure 23 : Diagramme d'activité du cas d'utilisation « Ajouter Enseignant »

5.1.6. Diagramme d'activité du cas d'utilisation « Création de planning ».

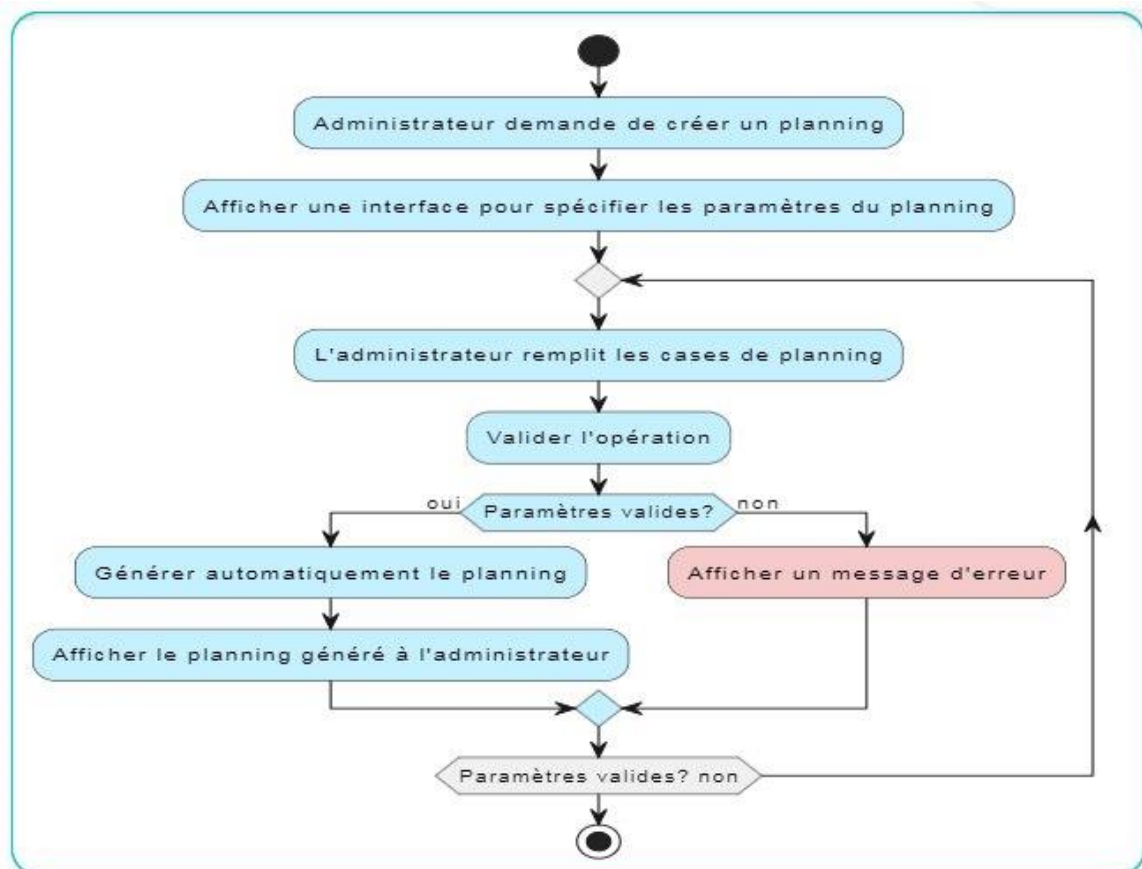


Figure 24 Diagramme d'activité du cas d'utilisation « Création de planning ».

5.2. Diagramme de classe : [18]

Le diagramme de classes est souvent considéré comme l'un des éléments les plus essentiels lors du développement orienté objet. Du point de vue de la branche fonctionnelle, ce diagramme permet de décrire la structure des entités qui sont manipulées par les utilisateurs. En phase de conception, le diagramme de classes représente la structure d'un code orienté objet, et idéalement, les modules du langage de développement sont également pris en compte.

Ce diagramme offre une vue d'ensemble de la structure du système, en identifiant les classes, les attributs, les méthodes et les relations entre les différentes entités. Il permet de définir les objets clés du système, ainsi que leurs caractéristiques et leurs interactions.

5.2.1. Règles de gestion du diagramme de classe :

Ces règles de gestion établissent les relations et les associations entre les différentes classes du diagramme de classe. Elles décrivent comment les entités sont connectées les unes aux autres et permettent de représenter les contraintes et les interactions entre les objets du système :

- Une année académique (AcademicYear) peut avoir plusieurs branches (Branch), mais une branche appartient à une seule année académique.
- Une branche (Branch) peut être associée à plusieurs module (Subject), mais un module appartient à une seule branche.
- Une section (Section) peut contenir plusieurs sous-groupes (Group), mais un groupe appartient à une seule section.
- Une salle (Room) peut être de plusieurs types de salle (RoomType), mais un type de salle peut être associé à plusieurs salles.
- Une salle (Room) peut être utilisée dans plusieurs plannings (Lesson), mais un planning est associé à une seule salle.
- Un planning (Lesson) est attribué à un utilisateur (User), un enseignant (Teacher), une salle (Room) et un module (Subject) spécifiques.
- Un utilisateur (User) peut être associé à plusieurs plannings (Lesson), mais un planning est attribué à un seul utilisateur.
- Un enseignant (Teacher) peut être associé à plusieurs plannings (Lesson), mais un planning est attribué à un seul enseignant.
- Un enseignant (Teacher) appartient à un seul genre (Gender), mais un genre peut être associé à plusieurs enseignants.

Chapitre 2 : Analyse et conception

- Un planning (Lesson) peut être attribué à un groupe spécifique, identifié par son ID de groupe.

5.2.2. Diagramme de classe global :

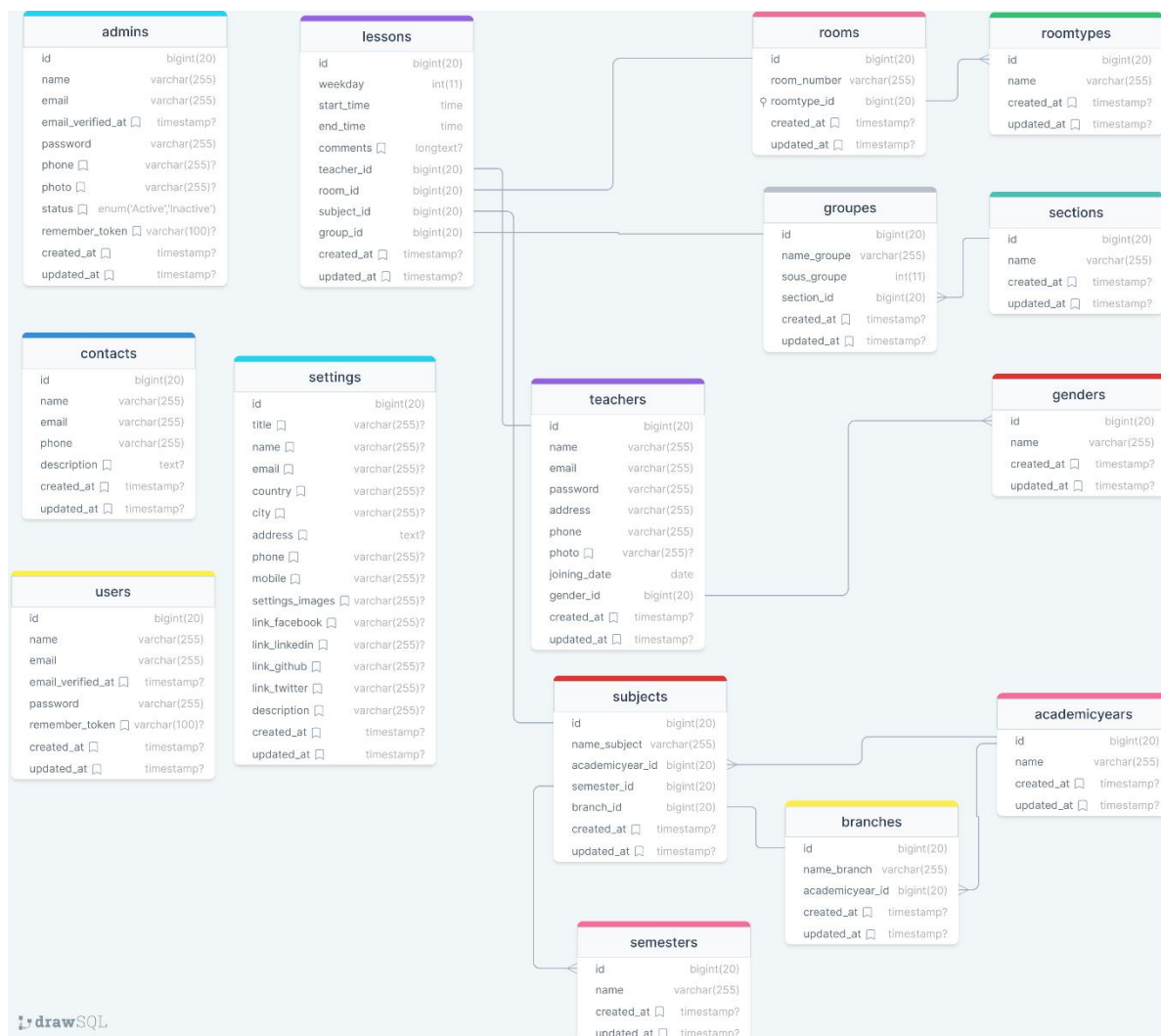


Figure 25 :Diagramme de classe global

6. Modèle relationnel :

Le modèle relationnel de données est représenté par les tables et les relations entre elles.

Voici la représentation des tables du modèle de données dans leur forme relationnelle :

- academicyears (id, name).
- branches (id, name_branch, #academicyears_id).
- genders (id, name).
- groupes (id, name_n_groupe, sous_group, #section_id).
- rooms (id, room_number, #roomtype_id).
- roomtypes (id, room).

Chapitre 2 : Analyse et conception

- sections (id, name).
- subjects (id, name_subject, #semster_id, #academicyears_id, #branche_id).
- contacts (id, name, email, phone, description).
- teachers (id, name, email, password, address, phone, joining_date, #gender_id).
- admins (id, name, email, email_verified_at, password, phone, photo, status, remember_token).
- users (id, name, email, email_verified_at, password, remember_token).
- lessons (id, weekday, start_date, end_date, comments, #user_id, #teacher_id, #room_id, #subject_id, #group_id).
- settings (id, title, name, email, country, city, address, phone, mobile, settings_images, link_facebook, link_linkedin, link_github, link_twitter, description).

7. Conclusion

Dans ce deuxième chapitre, nous avons réalisé une analyse approfondie, une conception détaillée et une documentation complète de notre système d'information en suivant une approche méthodique basée sur UML.

Grâce à cela, nous sommes maintenant prêts à passer à la prochaine phase de développement. Nous allons mettre en œuvre notre application en utilisant les spécifications établies dans ce chapitre. Cette phase de développement nous permettra de concrétiser les concepts et les fonctionnalités que nous avons modélisés, et de créer une application fonctionnelle qui répondra aux besoins identifiés.

Chapitre 3 : Réalisation

1. Introduction :

Dans le cadre de ce projet, nous avons développé une application en utilisant une combinaison de logiciels, d'outils et de langages afin de concrétiser nos idées et de mettre en place les fonctionnalités nécessaires. Dans cette dernière partie de notre travail, nous allons brièvement présenter l'environnement de développement, les langages de programmation et les outils que nous avons utilisés. De plus, nous aborderons l'architecture de notre application et conclurons en mettant en avant les principales interfaces de celle-ci. L'objectif est de donner un aperçu des choix techniques que nous avons faits et de montrer comment ils ont contribué au succès de notre application.

2. L'environnement matériel :

L'environnement matériel fait référence aux composants physiques utilisés pour exécuter une application ou un système informatique. Voici quelques éléments importants de l'environnement matériel qui a été utilisé pour développer mon application :

- Ordinateur Portable: DESKTOP- S9FSUUB.
- Mémoire: 8 Go.
- Disque Dur: 500 Go.
- Processeur: Intel(R) Core (TM) i5- -5010U CPU @ 2.10GHz 2.10 GHz.
- Type de système: Windows 10 Professionnel

3. Outils et langages de développement utilisés :

Voici les différents outils et langages de développement utilisés pour créer le système :

❖ Visual Paradigm online : [27]

Visual Paradigm est un puissant outil de développement et de modélisation logicielle qui offre des fonctionnalités complètes pour concevoir, documenter et générer du code pour des systèmes logiciels. Il prend en charge les langages de modélisation standard, facilite la collaboration d'équipe et propose des fonctionnalités avancées telles que la gestion des exigences et la génération de documentation. Utilisé largement dans l'industrie du développement logiciel, Visual Paradigm améliore la productivité, la qualité et la communication entre les membres de l'équipe.



Figure 26 Logo Visual Paradigm

❖ **PhpStorm:** [28]

PhpStorm est un IDE développé par JetBrains, conçu spécifiquement pour les développeurs PHP et web. Il offre un large éventail de fonctionnalités avancées, telles que la refactorisation de code, l'autocomplétion, la prévention des erreurs en temps réel et un éditeur HTML, CSS et JavaScript complet. PhpStorm est basé sur la plateforme IntelliJ IDEA et prend en charge les versions de PHP 5.3 et ultérieures. Il permet également de travailler avec des bases de données et d'exécuter des requêtes SQL. C'est un outil polyvalent et performant pour le développement web avec PHP.



Figure 27:Logo PhpStorm

❖ **Xampp :** [28]

XAMPP, qui est un acronyme pour Cross-Platform, Apache, MYSQL, PHP et Perl, est un serveur web multiplateforme largement utilisé qui permet aux développeurs de créer et de tester leurs programmes sur un serveur web local. Développé par les Apache Friends, il comprend le serveur Apache HTTP, MariaDB et un interpréteur pour différents langages de programmation tels que PHP et Perl. XAMPP est disponible en plusieurs langues et est pris en charge sur différentes plates-formes.

XAMPP est un ensemble de solutions web open source comprenant une distribution Apache pour de nombreux serveurs, des exécutables en ligne de commande et des modules tels que le serveur Apache, MariaDB, PHP et Perl.

XAMPP est un outil qui permet aux serveurs locaux de tester leurs sites web et leurs clients sur des ordinateurs et des ordinateurs portables avant de les déployer sur le serveur principal. Il offre un environnement idéal pour tester et vérifier le bon fonctionnement de projets basés sur Apache, Perl, MySQL et PHP directement sur la machine hôte. Parmi ces technologies, Perl est un langage de programmation utilisé pour le développement web, PHP est un langage de script côté serveur, et MariaDB, développée par MySQL, est la base de données la plus couramment utilisée.



Figure 28:Logo XAMPP

– **Serveur apache:** [29]

Apache est un logiciel libre qui fonctionne en tant que serveur HTTP. Il a été créé et est maintenu par la fondation Apache. Pendant de nombreuses années, Apache a été le serveur HTTP le plus populaire sur le World Wide Web. Il est distribué sous les termes de la licence Apache, ce qui signifie qu'il peut être utilisé, modifié et distribué librement. Apache offre une solution fiable et flexible pour la gestion des requêtes HTTP et la diffusion de contenu web.



Figure 29 : Logo Apache

– **MYSQL:** [28]

MySQL est un système de gestion de base de données relationnelles (SGBDR) open source, développé et soutenu par Oracle Company. Il permet de stocker, gérer et interroger efficacement des données structurées dans des tables relationnelles. MySQL est largement utilisé dans le développement d'applications web et est apprécié pour sa rapidité, sa scalabilité et sa facilité d'utilisation. Il est couramment utilisé en conjonction avec des scripts PHP pour créer des applications d'entreprise puissantes et dynamiques côté serveur ou basées sur le web.



Figure 30 Logo MySQL

– **PHP:** [28]

PHP (signifie Hypertext Preprocessor) est un langage de script côté serveur, open source, interprété et orienté objet, utilisé principalement pour le développement web. Il permet de créer des applications web dynamiques et peut être intégré dans du code HTML. PHP est rapide, simple à apprendre et offre des fonctionnalités orientées objet. C'est un langage populaire pour la gestion du contenu dynamique des sites web.



Figure 31: Logo php

❖ **Bootstrap : [30]**

Bootstrap est un framework front-end gratuit qui facilite et accélère le développement de sites web. Il propose une variété de modèles de conception prêts à l'emploi pour différents éléments tels que la typographie, les formulaires, les boutons, la navigation, les tableaux, les modales et les carrousels d'images. Bootstrap offre également des plugins JavaScript optionnels pour ajouter des fonctionnalités supplémentaires. Son avantage principal réside dans sa capacité à créer des designs réactifs, adaptés automatiquement aux différents appareils et tailles d'écran. Ainsi, les sites web créés avec Bootstrap offrent une expérience utilisateur optimale sur les ordinateurs de bureau, les tablettes et les smartphones.



Figure 32 : Logo Bootstrap

❖ **Git: [31]**

Git is a powerful and efficient distributed version control system, ideal for projects of all sizes. It is open source and free, offering speed and performance advantages over other SCM tools like Subversion, CVS, Perforce, and ClearCase. With its small footprint, Git is easy to learn and boasts features such as affordable local branching, convenient staging areas, and support for various workflows.



Figure 33 Logo Git

❖ **GitHub: [28]**

GitHub est une plateforme d'hébergement de code qui offre des fonctionnalités de contrôle de version, de collaboration et de gestion de projet. Elle permet aux développeurs de travailler ensemble sur des projets, en utilisant Git comme outil de contrôle de version. GitHub comprend des composants essentiels tels que les dépôts, les branches, les validations et les demandes de tirage. Il facilite également la collaboration grâce à des fonctionnalités telles que le suivi des bugs et la gestion des tâches.



Figure 34 Logo GitHub

❖ **HTML: [30]**

HTML, abréviation de Hyper Text Markup Language, est le langage standard pour la création de pages Web. Il permet de structurer le contenu des pages en utilisant des balises telles que des en-têtes, des paragraphes et des liens. Chaque élément est étiqueté avec une balise spécifique qui définit son objectif et son type de contenu. Lorsque la page est chargée dans un navigateur, celui-ci interprète les balises HTML pour afficher le contenu de manière appropriée, en utilisant les informations fournies par les balises pour déterminer la présentation adéquate du document.

❖ **CSS: [18]**

CSS, acronyme de Cascading Style Sheets (Les feuilles de style en cascade), est le langage utilisé pour styliser les documents HTML. Il permet de décrire comment les éléments HTML doivent être affichés à l'écran, sur papier ou dans d'autres médias. L'utilisation de CSS permet d'économiser beaucoup de travail, car il offre la possibilité de contrôler la mise en page de plusieurs pages Web simultanément. Les règles de style CSS sont généralement stockées dans des fichiers externes, appelés fichiers CSS.

❖ **JavaScript: [30]**

JavaScript est le langage de programmation le plus populaire utilisé pour créer des applications Web interactives. Il permet d'ajouter des fonctionnalités dynamiques aux pages Web, comme des animations, des validations de formulaire et des mises à jour en temps réel. Grâce à sa syntaxe simple et à sa large compatibilité avec les navigateurs modernes, JavaScript est accessible aux débutants en programmation et est largement utilisé dans le développement Web.

❖ **La technique AJAX : [28]**

AJAX (Asynchrones JavaScript and XML) est un ensemble de technologies qui permet l'échange de données de manière asynchrone entre le client et le serveur sans recharger la page web. Grâce à AJAX, il est possible d'envoyer et de recevoir des données

en arrière-plan, ce qui améliore la réactivité et la performance des applications web. En utilisant AJAX, seules les informations pertinentes sont envoyées au serveur, ce qui réduit la quantité de données échangées et rend l'application plus rapide et interactive.



Figure 35 Logo Ajax

❖ La bibliothèque JQuery: [28]

JQuery est une bibliothèque JavaScript légère et polyvalente qui simplifie la manipulation du DOM et facilite le développement d'applications web interactives. Avec JQuery, vous pouvez sélectionner et modifier des éléments HTML, gérer les événements, animer des éléments, manipuler les styles CSS, effectuer des requêtes AJAX, et bien plus encore. Son utilisation permet d'écrire du code JavaScript de manière plus rapide et efficace grâce à sa syntaxe concise et sa compatibilité multiplateforme. JQuery est largement utilisé dans le développement web pour améliorer l'expérience utilisateur et accélérer le développement d'applications web dynamiques.



Figure 36 Logo JQuery

❖ Framework Laravel : [32]

Un framework est, comme son nom l'indique en anglais, un « cadre de travail ». Un framework est un ensemble de composants logiciels que l'on retrouve sur la plupart des projets d'applications. C'est en quelque sorte une boîte à outils qui comprend tous les éléments indispensables à la création d'une application web moderne.

Il existe différents types de frameworks dans le domaine du développement logiciel, chacun avec ses propres objectifs et fonctionnalités. Dans notre mémoire, nous avons spécifiquement travaillé sur le type de framework de développement web, en vous concentrant sur Laravel, qui est un framework de développement web en PHP.

Laravel est un framework PHP puissant et élégant, basé sur le modèle MVC, destiné aux développeurs souhaitant créer des applications web complètes. Créé par Taylor Otwell, il propose une structure de développement solide et offre de nombreuses fonctionnalités préconfigurées telles que la gestion des bases de données, les sessions, l'authentification utilisateur et la génération de code. Laravel se distingue par sa syntaxe élégante, sa facilité

d'utilisation et sa documentation complète, ce qui en fait un choix populaire parmi les développeurs PHP. En favorisant les meilleures pratiques de développement, Laravel permet de créer des applications web fiables, sécurisées et évolutives. [33]



Figure 37 Logo Laravel

- **Pourquoi choisi Laravel ?**

Laravel est choisi pour ses fonctionnalités avancées, sa facilité d'utilisation, son écosystème riche et son excellente documentation, ce qui en fait un choix populaire pour le développement web avec PHP.

3. Architecture MVC (Model-View-Controller): [34]

Nous avons mis en place une architecture MVC (Modèle-Vue-Contrôleur) dans notre application, composée du modèle, de la vue et du contrôleur

Comme de nombreux autres frameworks PHP, Laravel adopte une architecture MVC (Modèle-Vue-Contrôleur).

Le patron de conception MVC (Modèle-Vue-Contrôleur) est largement utilisé dans le développement d'applications web pour organiser les différents concepts de nos pages PHP. Il divise une application en trois composants logiques : le modèle, la vue et le contrôleur. Laravel, en tant que framework, repose sur cette architecture MVC pour rendre le développement accessible, puissant et faciliter la gestion d'applications volumineuses et robustes.

- 12.** Le modèle gère les interactions avec la base de données, regroupe et traite les données. Il contient à la fois les données et leur logique associée.
- 13.** La vue se charge de l'affichage des données renvoyées par le modèle. Elle est responsable de l'interface graphique présentée à l'utilisateur.
- 14.** Le contrôleur coordonne le modèle et la vue. Il capture les actions de l'utilisateur et, en fonction de ces actions, effectue les modifications nécessaires dans l'application. Le contrôleur gère les requêtes utilisateur, demande au modèle d'effectuer les changements et transmet les données à la vue.

Cette séparation en trois composants facilite la clarté de l'architecture du code et simplifie considérablement le travail des développeurs. L'architecture MVC de Laravel est illustrée dans la figure ci-dessous.

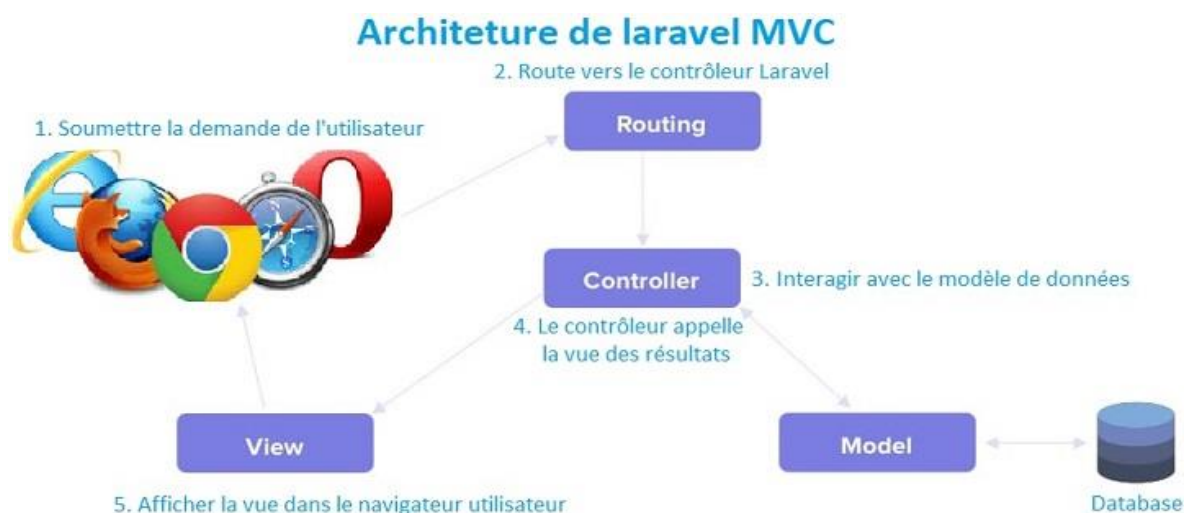


Figure 38: Architecture MVC

15. Le modèle détient à la fois les données et la logique associée.
16. La vue se concentre sur la présentation graphique destinée à l'utilisateur.
17. Le contrôleur gère les actions de l'utilisateur, communique avec le modèle pour effectuer les modifications nécessaires, puis transmet les données à la vue.
18. Concernant le routage, bien qu'il soit indépendant de l'architecture MVC, il constitue une partie essentielle de tous les frameworks PHP.

4. Présentation de l'application

4.1. Représentation des interfaces de l'application :

Les interfaces de l'application sont les représentations visuelles des écrans, formulaires et éléments interactifs. Voici quelques interfaces de notre application conçues pour offrir une expérience conviviale et intuitive aux utilisateurs :

❖ Interface Dashboard :

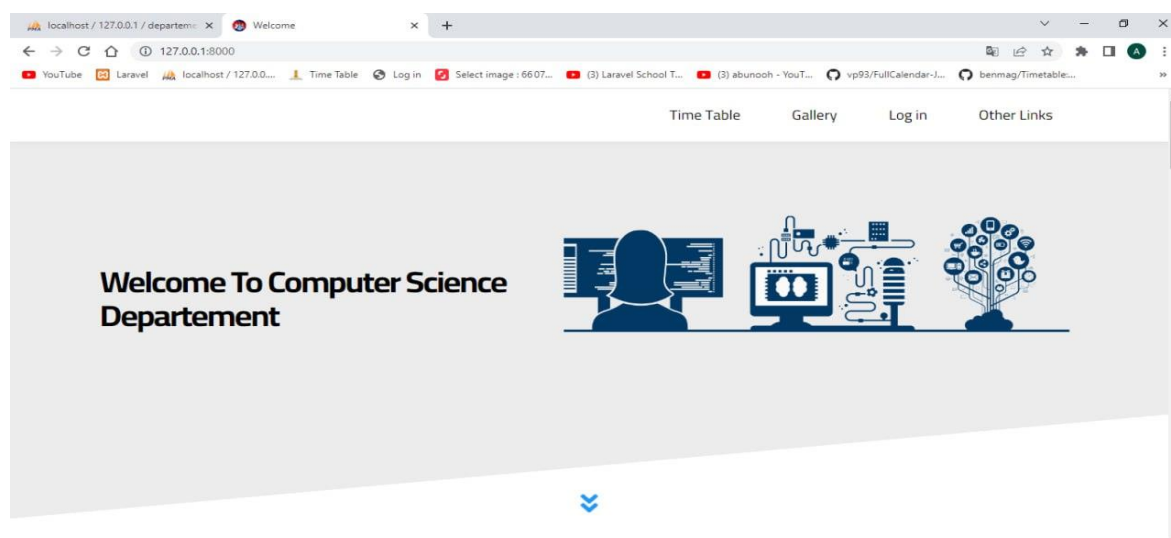


Figure 39 : Interface Dashboard

Chapitre 3 : Réalisation

❖ Interface de notre application :

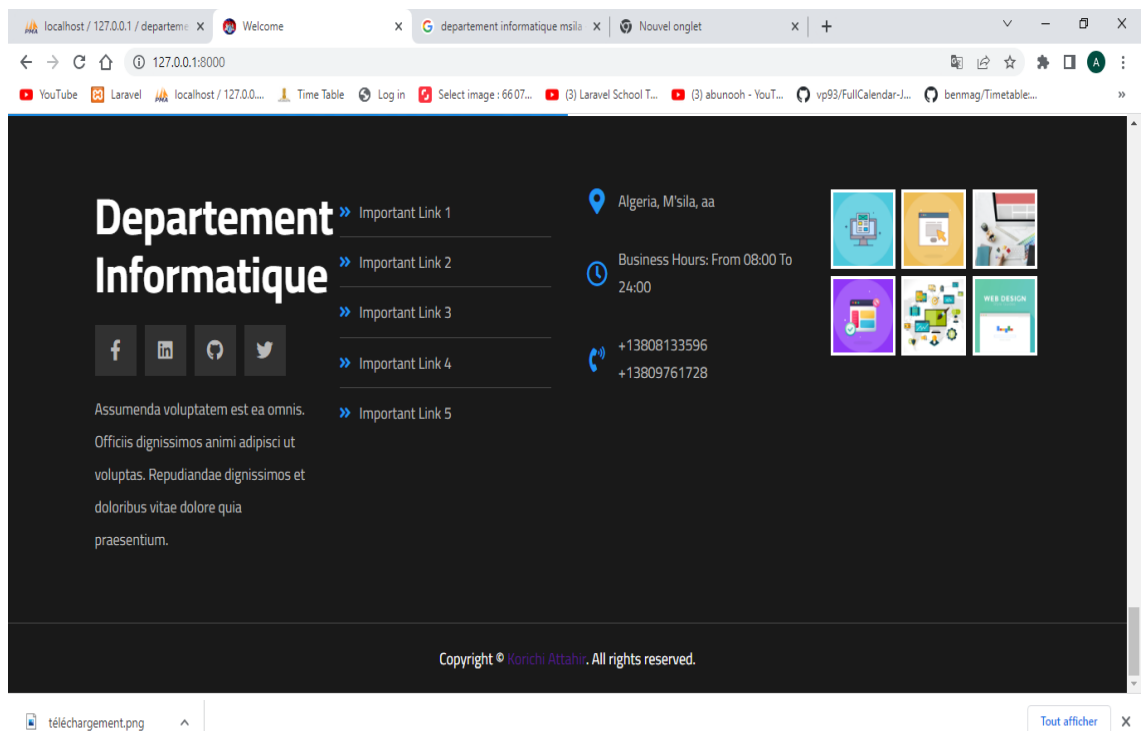


Figure 40: Interface de notre application

❖ L'interface de connexion :

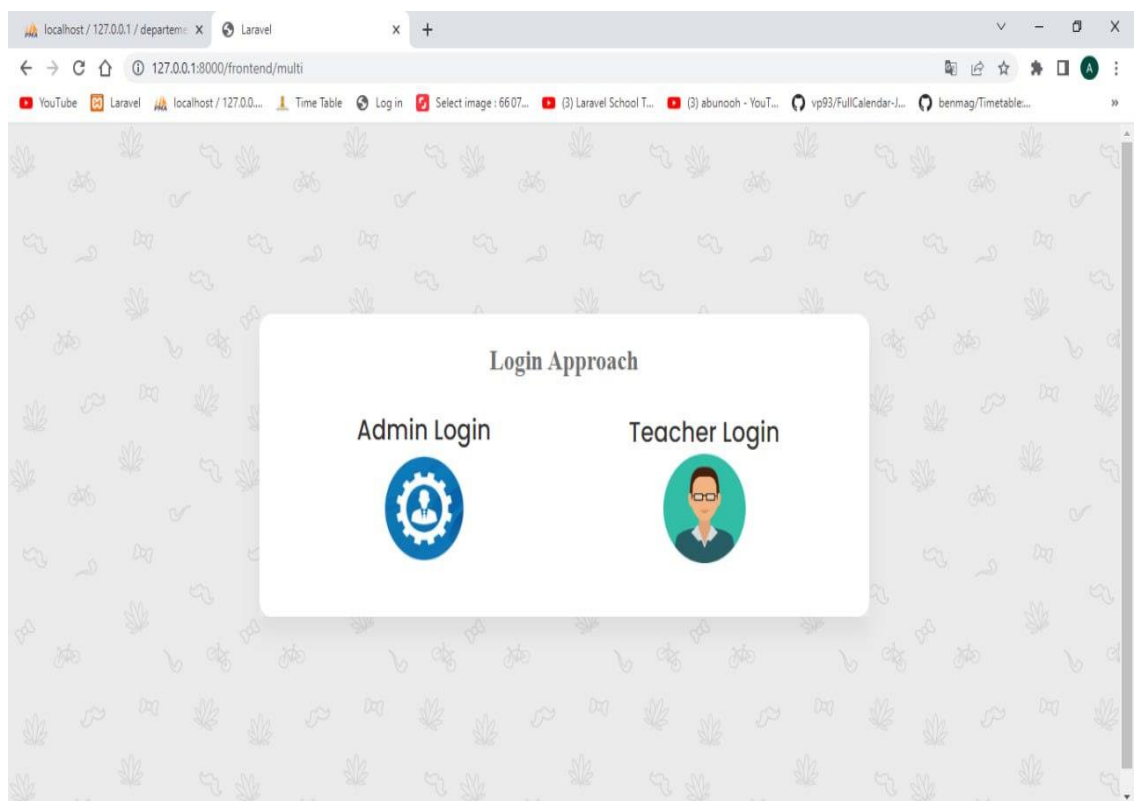


Figure 41: L'interface de connexion

Chapitre 3 : Réalisation

- ❖ **L'interface de connexion de l'administrateur** : Affichez un formulaire avec deux champs de saisie : un pour l'adresse e-mail ou le nom d'utilisateur et un pour le mot de passe.

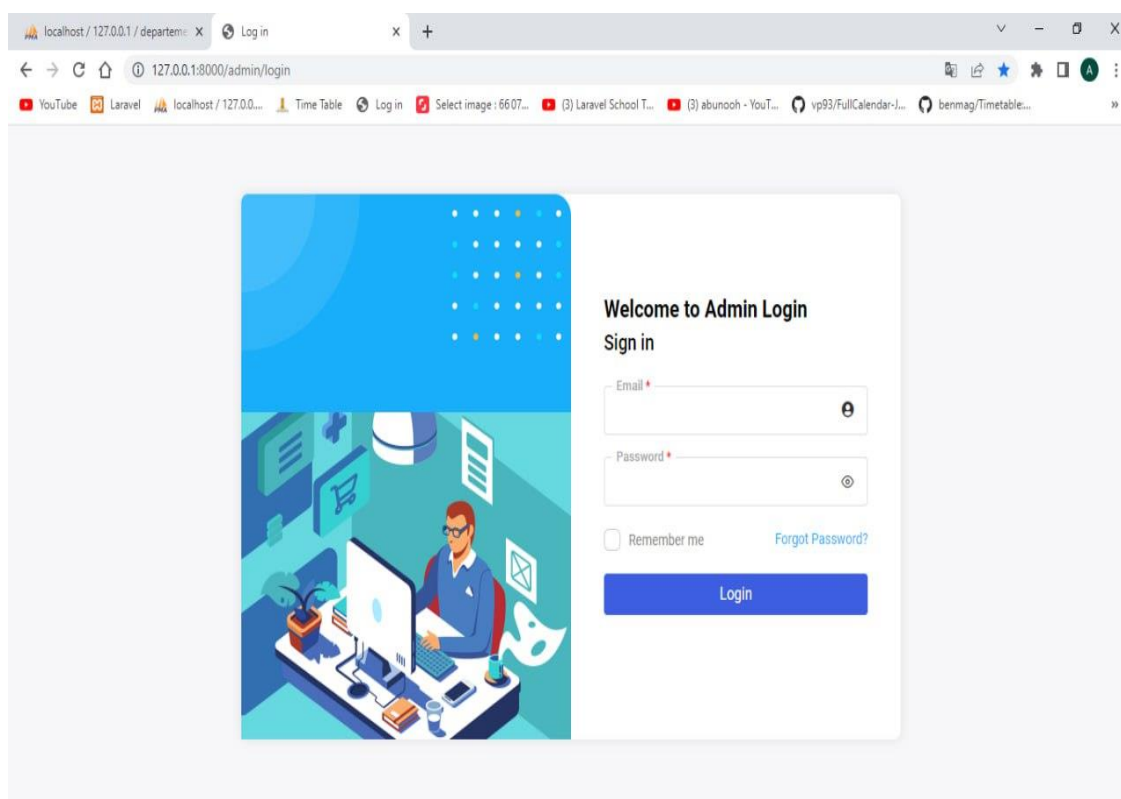


Figure 42:L'interface de connexion de l'administrateur

- ❖ **Interface d'Accueil** : également appelée tableau de bord (Dashboard), est une partie essentielle de toute application. Elle offre un aperçu des informations pertinentes et des fonctionnalités clés à l'administrateur dès qu'ils se connectent.

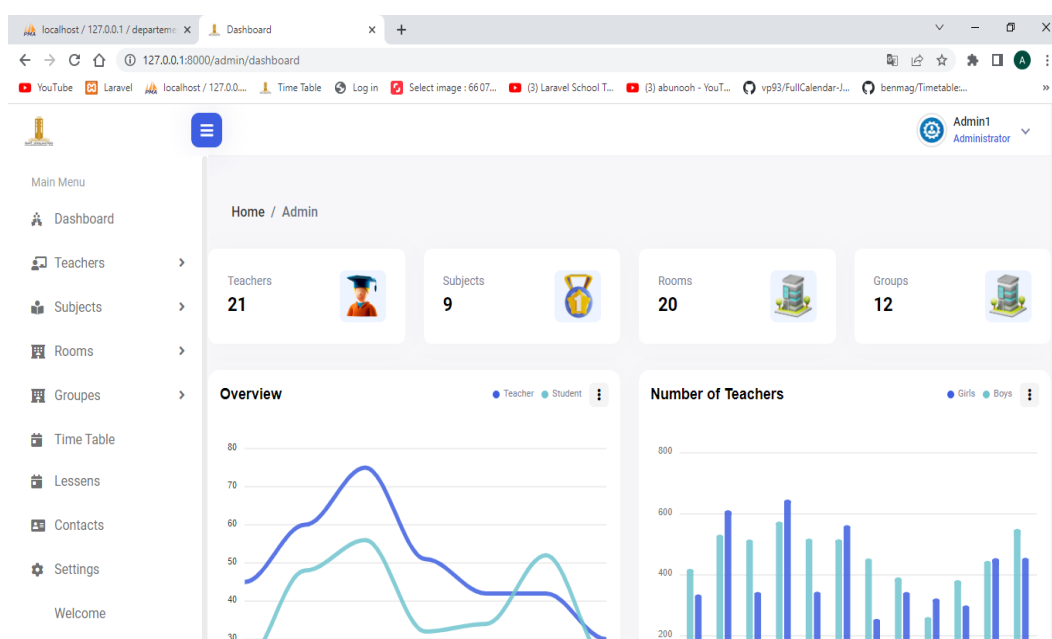


Figure 43:Interface d'Accueil

Chapitre 3 : Réalisation

- ❖ **Interface de gestion des enseignants :** Pour ajouter, sélectionner, rechercher ; modifier ou supprimer un enseignant. :

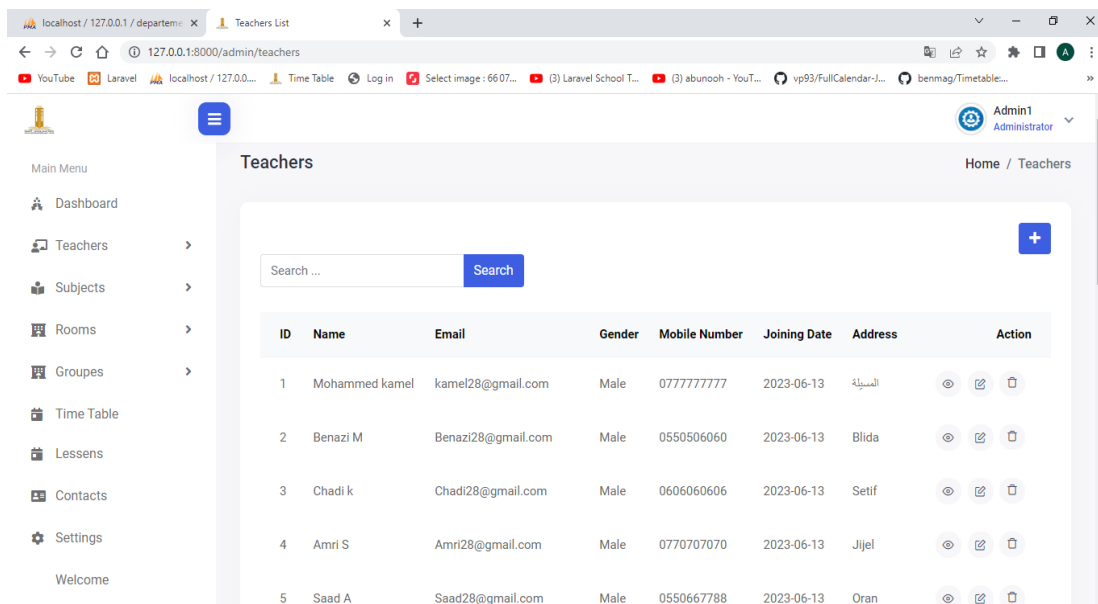


Figure 44: : L'interface de gestion des enseignants

- ❖ **Interface de créer un planning :**
L'interface de création d'un planning permet aux utilisateurs de concevoir et de planifier des activités, des tâches ou des événements dans l'application :

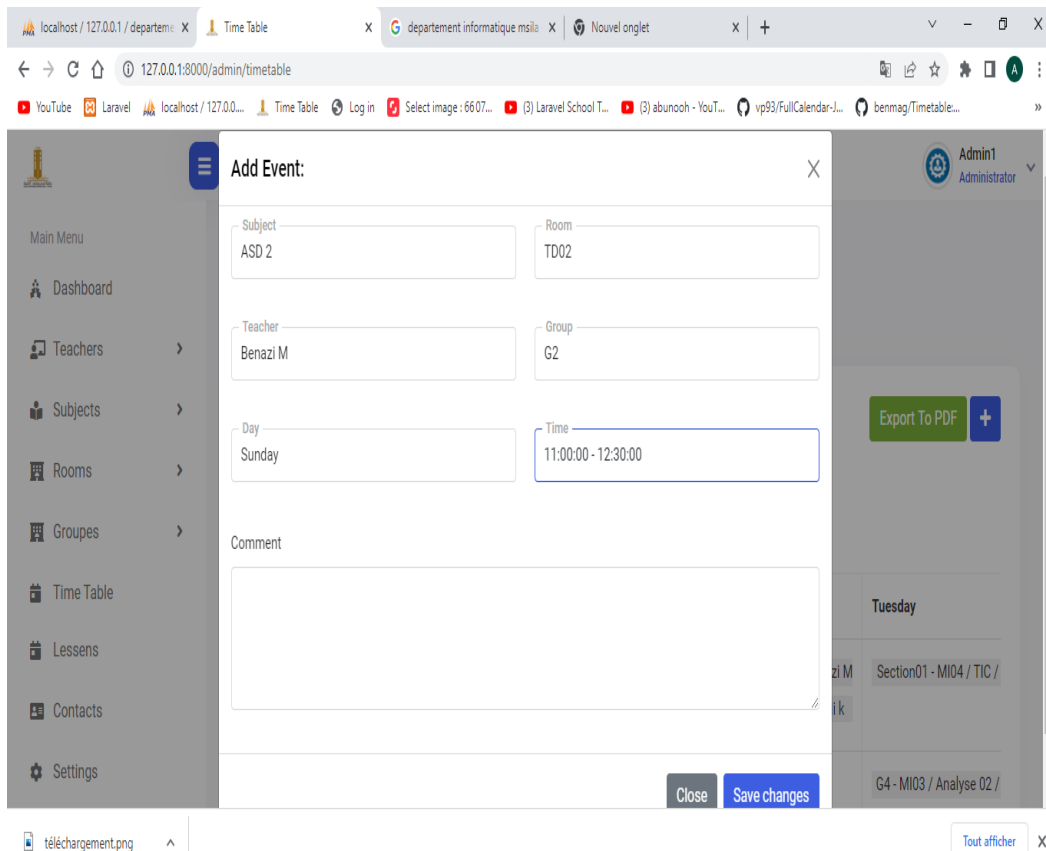


Figure 45 : L'interface de créer un planning :

- ❖ **Interface de consulter d'un planning :** Permet d'afficher le planning par l'enseignant ou le groupe :

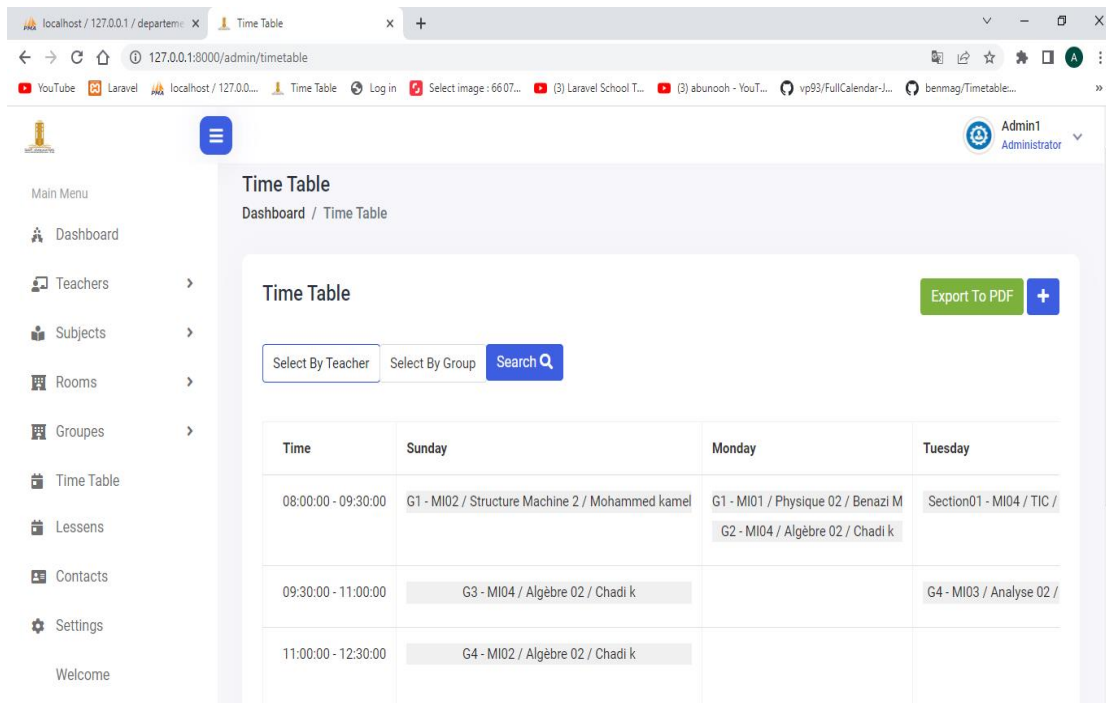


Figure 46: Interface de consulter d'un planning

- ❖ **Interface de contacter admin :**

L'interface de contact avec l'administrateur permet aux utilisateurs de communiquer facilement avec l'administrateur de l'application pour poser des questions, signaler des problèmes ou demander de l'assistance.

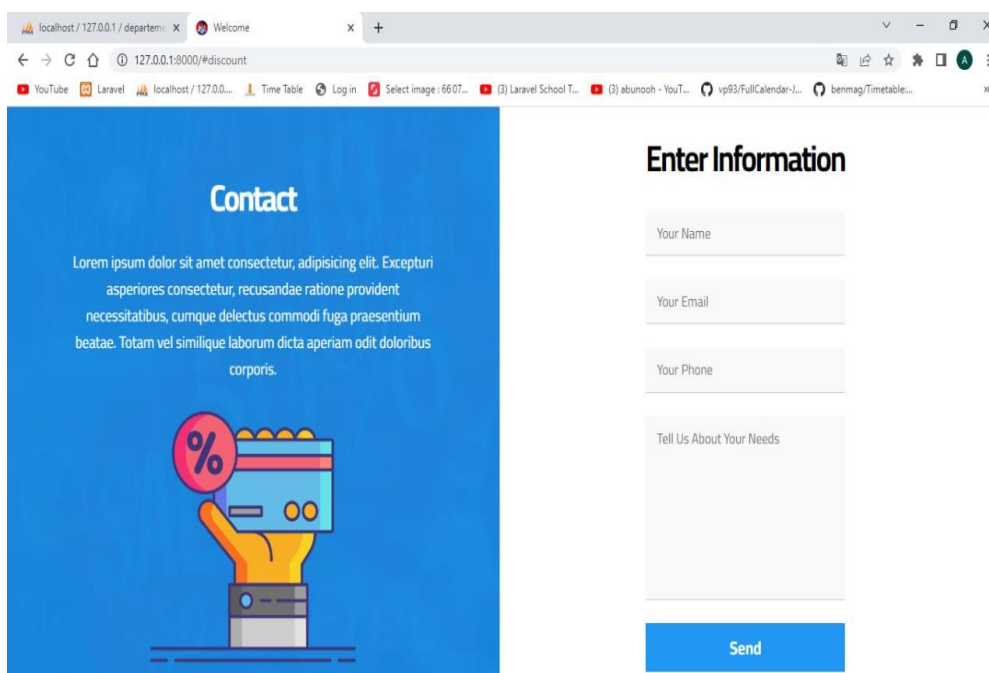


Figure 47: Interface de contacter admin

Chapitre 3 : Réalisation

❖ Interface de gérer les contacts :

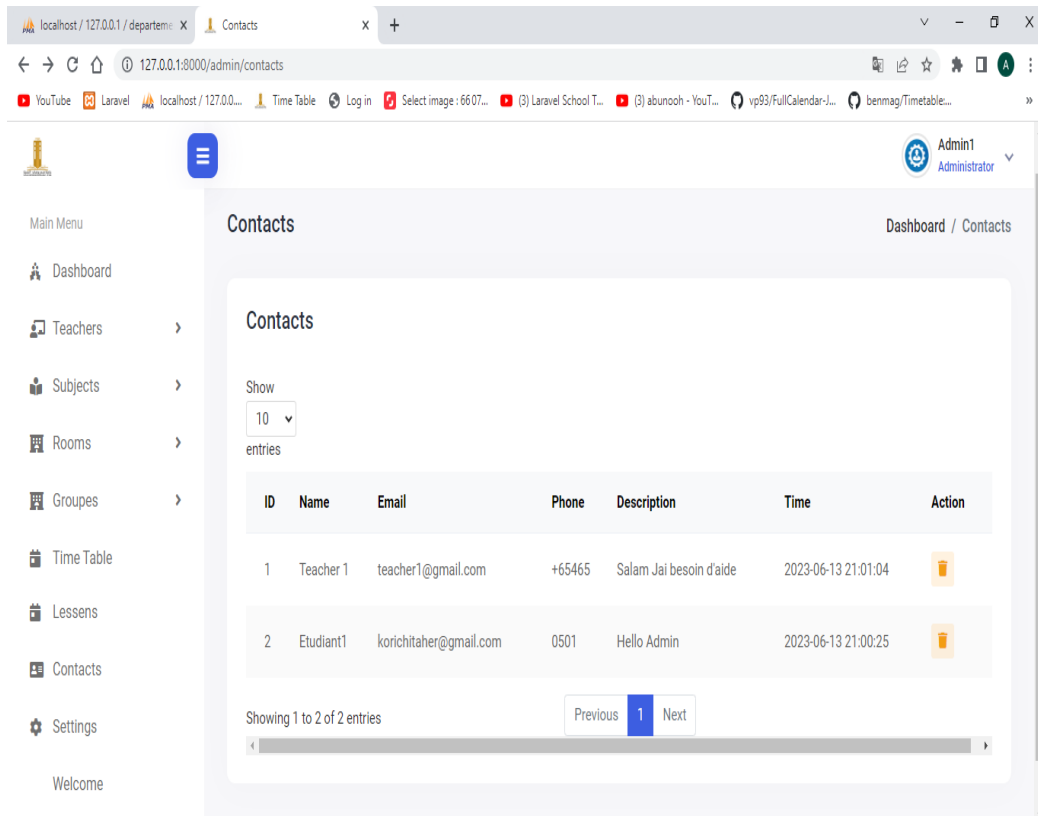


Figure 48: Interface de gérer les contacts

❖ : Interface de configuration :

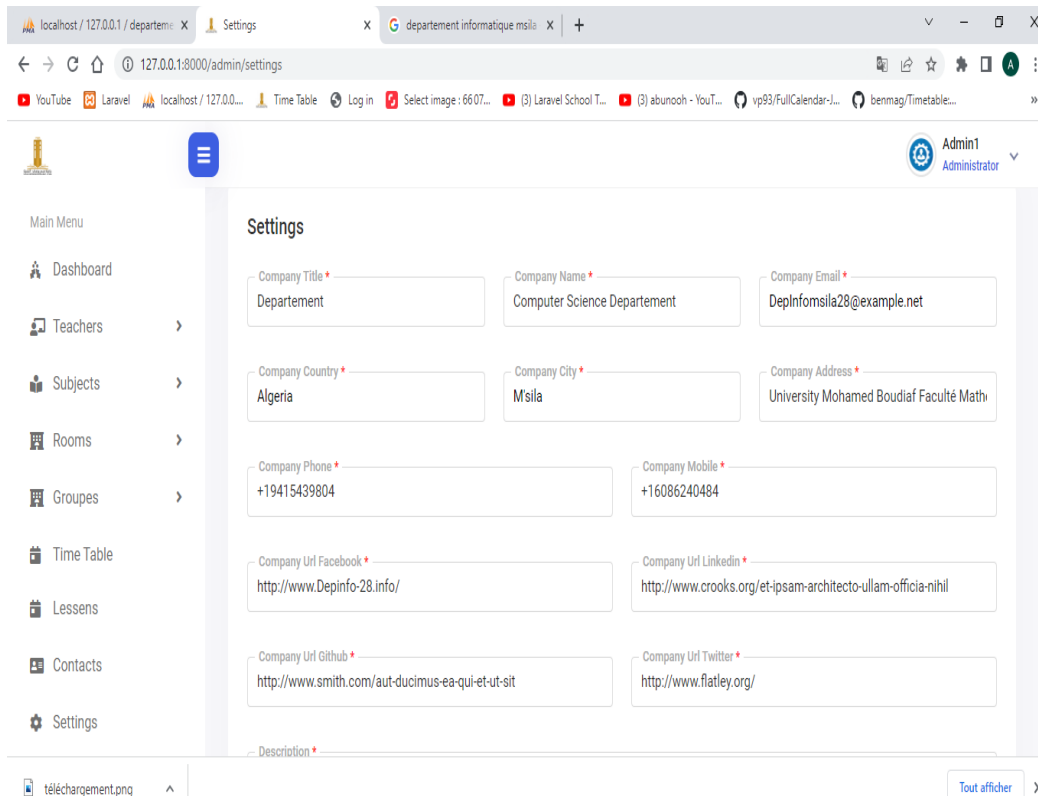


Figure 49: Interface de configuration

Chapitre 3 : Réalisation

- ❖ **Consulter le planning** : Permet de consulter le planning concerné par l'utilisateur :

• Time Table

Time	Sunday	Monday	Tuesday	Wednesday	Thursday
08:00.00 - 09:30.00	G1 - MI02 / Structure Machine 2 Teacher: Mohammed kamel G2 - TD02 / TIC Teacher: Noubat T G3 - TD03 / Analyse 02 Teacher: Bentleria R G4 - TD03 / Algèbre 02 Teacher: Lounes B	G1 - MI01 / Physique 02 Teacher: Benazi M G2 - MI04 / Algèbre 02 Teacher: Chadi k	Section01 - MI04 / TIC Teacher: Amri S	G9 - TP06 / English Teacher: Chouder R	G8 - TD02 / Physique 02 Teacher: Lounes B
09:30.00 - 11:00.00	G3 - MI04 / Algèbre 02 Teacher: Chadi k	Section01 - MI04 / Structure Machine 2 Teacher: Amri S	G4 - MI03 / Analyse 02 Teacher: Amri S		G2 - MI03 / Algèbre 02 Teacher: Benazi M
11:00.00 - 12:30.00	G4 - MI02 / Algèbre 02 Teacher: Chadi k G2 - TD02 / ASD 2 Teacher: Benazi M	Section02 - MI03 / English Teacher: Khetlaf A	Section01 - MI04 / Prob&Stat Teacher: Chouder R	Section02 - MI03 / OPM Teacher: Debbi H	Section02 - TD05 / Prob&Stat Teacher: Noubat T Section01 - MI03 / Physique 02 Teacher: Tahni Z
12:30.00 - 14:00.00					G3 - MI03 / Analyse 02 Teacher: Mohammed kamel
14:00.00 - 15:30.00	G5 - TD03 / OPM Teacher: Benazi M	G10 - TP02 / TIC Teacher: Amri S			G3 - MI03 / Algèbre 02 Teacher: Benazi M
15:30.00 - 17:00.00	G2 - MI02 / Analyse 02 Teacher: Benazi M	G6 - TD01 / Prob&Stat Teacher: Chouder R	G4 - TD05 / Physique 02 Teacher: Fakher	G6 - TD04 / Prob&Stat Teacher: Ghemoughi A	G6 - TD04 / OPM Teacher: Touahria M

Figure 50: Interface d'impression d'un planning

5. Conclusion :

Dans cette conclusion, nous avons présenté de manière générale notre application, en soulignant les outils, les langages et les environnements de développement utilisés. Nous avons également mis en avant les principales fonctionnalités à travers plusieurs interfaces. Ce chapitre met en évidence la réalisation et la réussite de notre application, en démontrant l'impact des choix techniques sur son développement.

Conclusion générale

La conception et l'implémentation de cette application web de planification pour le département d'informatique de la Faculté des Mathématiques et d'Informatique de Msila ont été une expérience réussie. En utilisant des méthodes de développement logiciel telles que le procédé UP et le langage de modélisation UML, ainsi qu'en acquérant des compétences techniques avancées dans des langages de programmation tels que **PHP**, **MySQL**, **JavaScript**, **HTML**, **CSS** et l'utilisation de bibliothèques et de **frameworks** tels que **Laravel** et **Bootstrap**, nous avons créé une application fonctionnelle et adaptée aux besoins spécifiques du département.

Cette application permet aux utilisateurs de consulter facilement et de gérer les plannings, offrant ainsi une plateforme centralisée pour la communication et la coordination au sein du département. Elle simplifie et optimise la gestion des plannings, contribuant à une meilleure organisation et planification des activités académiques.

Perspectives :

Malgré le succès de l'application, il existe des possibilités d'amélioration et d'expansion futures :

- Amélioration de l'interface utilisateur : Continuer à améliorer l'interface utilisateur en se basant sur les retours des utilisateurs et en adoptant les meilleures pratiques en matière de conception d'interfaces conviviales et intuitives.
- Ajout de fonctionnalités supplémentaires : Explorer la possibilité d'ajouter de nouvelles fonctionnalités, telles que la génération automatique de plannings en tenant compte des contraintes spécifiques, l'intégration avec d'autres systèmes académiques, la prise en charge de notifications et de rappels, etc.
- Adaptation à d'autres départements ou institutions : Évaluer la possibilité d'adapter l'application pour répondre aux besoins d'autres départements ou institutions, en personnalisant les fonctionnalités selon leurs exigences spécifiques.
- Maintenance et mise à jour continue : Assurer une maintenance régulière de l'application, en garantissant sa stabilité, en résolvant les problèmes potentiels et en appliquant les mises à jour nécessaires pour suivre l'évolution des technologies et des besoins des utilisateurs.
- Utilisation de l'intelligence artificielle : L'intégration de techniques d'intelligence artificielle, telles que l'apprentissage automatique et l'optimisation basée sur des

Conclusion générale

algorithmes avancés, pourrait permettre d'améliorer encore davantage la précision et l'efficacité de la planification.

En poursuivant ces perspectives, l'application de planification et de gestion des plannings peut continuer à évoluer et à répondre aux besoins croissants du département et de l'institution, contribuant ainsi à l'amélioration globale de l'organisation et de la qualité de l'enseignement.

Bibliographies

- [1] : S. Piechowiak, C. Kolski « Analyse et conception d'un outil interactif d'aide à la gestion des emplois du temps basé sur les points de vue », LAMIH - UMR CNRS 8530 – Université de Valenciennes et du Hainaut-Cambrésis 59313 Valenciennes Cédex 9– France, Revue d'Interaction Homme Machine (RIHM), volume 5, n° 2, pp. 1-30, 2004.
- [2] : Tigrine Youva- Tahir Oussama/Conception et réalisation d'une application web pour la gestion des emplois du temps Cas : UMMTO/ Promotion 2016/2017/ UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU/pp :22---25)
- [3] L. RAMDANE /Automatisation De La Génération De l'Emploi Du Temps/ Master /UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM/ Promotion 2015/2016 /pp :10)
- [4] : Remy-Robert, Alexandre Joseph, « Systèmes interactifs d'aide à l'élaboration de plannings de travail de personnel », Thèse de doctorat, Laboratoire TIMC, Institut IMAG, Université Joseph Fourier-Grenoble, 07 novembre 2003
- [5] : [Chan et all, 2002] : Chan Yew Chéong, Peter, « La planification du personnel : acteurs, actions et termes multiples pour une planification opérationnelle des personnes », Thèse de doctorat, Institut IMAG, Université Joseph FourierGrenoble, 1 octobre 2002.
- [6] : Melle SAICHE Cylia, Mr OUYOUGOUTE Abdelatif : Mémoire de fin de cycle de master professionnel : Conception et réalisation d'une application web pour la gestion des étudiants d'une école privée. Cas d'étude : "ISA School", Université A/Mira de Bejaia 2014/2015.
- [7] : M. MEDDAH - / Conception et réalisation d'une application web de gestion de projets de fin d'études / Université ABDELHAMID IBN BADIS – MOSTAGANEM- /2020-2021
- [8] : V.Foundikou, Méthodes de Conception Orientés Objet (MCOO), Recueil de cours, 2012
- [9]: Larman, C. (2011). "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development". Pearson Education.
- [10] : P. Roques, F. Vallée. UML en action - 4ème édition, de l'analyse des besoins à la conception, EYROLLES, 2004.)
- [11] : J. Conallen. Concevoir des applications web avec UML. EYROLLES, 2000.
- [12] : R. P. Franck Vallée, UML 2 en action, de l'analyse des besoins à la conception. 4ème édition, EYROLLES, 2007
- [13] : virtuelcampus.univ-msila.dz, <http://virtuelcampus.univ-msila.dz/fmi/index.php/fr/>
consulté le : 09/05/2023.

Bibliographies

- [14] : ideematic.com, <https://www.ideematic.com/dictionnaire-digital/application-web/#:~:text=D%C3%A9finition%20de%20l%27application%20web,l%27installer%20sur%20son%20ordinateur> consulté le : 15/05/2023.
- [15] : yeeply.c , <https://fr.yeeply.com/blog/5-types-developpement-d-applications-web/> consulté le : 15/05/2023.
- [16] : apprendre-la-programmation.net, [Programmation web : langages et logiciels | Apprendre la programmation \(apprendre-la-programmation.net\)](#) consulté le: 18/05/2023.
- [17] : dcllic.info, consulté le : 18/05/2023,<https://www.dcllic.info/services/developpement-application-gestion-specifique/>
- [18] : Object Management Group (OMG) : <https://www.omg.org/spec/UML/2.5.1/> consulté le : 19/05/2023
- [19] : cybermedian,<https://www.cybermedian.com/fr/a-comprehensive-guide-to-uml-class-diagram> Consulté le : 19/05/2023
- [20] : UML 2.5 Specification - Chapter 9 : "Structural Diagrams" (<http://www.omg.org/spec/UML/2.5/PDF/>) consulté le : 20/05/2023
- [21] : UML 2.5 Specification - Chapter 14 : "Behavioral Diagrams" (<http://www.omg.org/spec/UML/2.5/PDF/>) consulté le : 20/05/2023
- [22] : UML 2.5 Specification - Chapter 17: "Interaction Overview Diagrams" (<http://www.omg.org/spec/UML/2.5/PDF/>) consulté le: 21/05/2023
- [23] : cours-gratuit,<https://www.cours-gratuit.com/cours-uml/demarche-d-analyse-et-de-conception-avec-le-langage-uml> consulté le: 21/05/2023
- [24] developpez. ,<https://www.developpez.com/> consulté le: 21/05/2023
- [25] : Ingénierie Système et SysML ,<http://www.xr6805.fr/IS/Diagrammedecontexte.html> ; consulté le 24/5/2023
- [26] : <http://projet.eu.org/pedago/sin/term/3-UML.pdf>, consulté le 24/5/2023
- [27] : visual-paradigm, <https://www.visual-paradigm.com/> ,consulté le 24/5/2023
- [28] : javatpoint , <https://www.javatpoint.com/>, consulté le 24/5/2023
- [29] : wikipedia, https://fr.wikipedia.org/wiki/Apache_HTTP_Server ,consulté le 24/5/2023
- [30]: W3Schools ,[W3Schools Online Web Tutorials](#) ;consulté le 24/5/2023
- [31] : git ,<https://git-scm.com/>,consulté le 24/5/2023
- [32] : hexium , <https://www.hexium.io/blog/pourquoi-utiliser-le-framework-PHP-laravel> ,consulté le 28/5/2023
- [33] : Dataquest. didaquest.org, <https://didaquest.org/wiki/Laravel.>, consulté le 28/5/2023
- [34] : laravel , <https://apcpedagogie.com/le-modele-mvc-de-laravel/>., consulté le 28/5/2023

Résumé

Notre projet consiste à concevoir et implémenter une application web de gestion des emplois du temps pour le département d'informatique de la Faculté des Mathématiques et d'Informatique de Msila. Cette application permettra aux administrateurs du département de créer facilement des plannings semi-automatiques, tandis que les enseignants et les étudiants pourront consulter leur emploi du temps. En utilisant des technologies web modernes, notre application vise à améliorer l'efficacité, l'organisation et la flexibilité du processus de planification, tout en facilitant la communication entre les différents acteurs impliqués.

Mots clés : Planning, Laravel, calendrier, UML,

Abstract

Our project consists of designing and implementing a web application for timetable management in the Department of Computer Science at the Faculty of Mathematics and Computer Science in Msila. This application will enable department administrators to easily create semi-automated schedules, while teachers and students will be able to view their timetables. Using modern web technologies, our application aims to improve efficiency, organization, and flexibility in the planning process, while facilitating communication among the different stakeholders.

Keywords: Timetable, Laravel, calendar, UML,

ملخص

يهدف مشروعنا إلى تصميم وتنفيذ تطبيق ويب لإدارة جداول الأوقات في قسم علوم الحاسوب بكلية الرياضيات وعلوم الحاسوب في مسيلة. سيتيح التطبيق لمسؤولي القسم إنشاء جداول زمنية شبه آلية بسهولة، بينما سيتمكن المعلمون والطلاب من الاطلاع على جداول أوقاتهم الشخصية. من خلال استخدام تقنيات الويب الحديثة، يهدف تطبيقنا إلى تحسين الكفاءة والتنظيم والمرونة في عملية التخطيط، مع تسهيل التواصل بين جميع الأطراف المعنية.

كلمات مفتاحية: جدول الزمني، لارافل، جدول المواعيد