

## DÉDICACE

*À nous familles*

## REMERCIEMENTS

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier notre encadrante Pr. **Noura AHERRAHROU**, son précieux conseil et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.



## TABLE DES MATIÈRES

DÉDICACE.....	II
REMERCIEMENTS .....	III
TABLE DES MATIÈRES .....	V
LISTE DES TABLEAUX.....	<b>ERREUR ! SIGNET NON DEFINI.</b>
LISTE DES FIGURES.....	VII
INTRODUCTION.....	8
CHAPITRE 1    CONTEXTE GENERALE DU PROJET .....	9
1.1    Problématique.....	9
1.2    Solution et critique .....	9
1.3    Cahier des charges.....	10
1.4    Gestion de projet .....	12
1.5    Conclusion.....	13
CHAPITRE 2    PHASES D'ANALYSE ET DE CONCEPTION.....	14
2.1    Choix de la méthode de modélisation .....	14
2.1.1    Conception et modélisation.....	15
2.1.1.1    Diagramme de cas d'utilisation.....	16
2.1.1.2    Diagramme de séquence.....	18
2.1.1.3    Diagramme de classe.....	21
2.1.2    Conclusion.....	22
CHAPITRE 3    ENVIRONNEMENT DE TRAVAIL ET MISE EN PLACE DE L'APPLICATION.....	23
3.1    Présentation des solutions techniques .....	23
3.1.1    Architecture MVC.....	23

3.1.1.1	Modèle.....	24
3.1.1.2	Contrôleur.....	24
3.1.1.3	Vue .....	24
3.1.2	Technologie et Framework utilisée .....	25
3.1.2.1	Technologies web.....	25
3.1.2.2	Bootstrap .....	26
3.1.2.3	Serveur Apache .....	26
3.1.2.4	MS-PROJECT.....	27
3.1.3	Outils de développement et de modélisation.....	28
3.1.3.1	StarUML.....	28
3.2	Présentation de l'application .....	28
3.2.1	Page d'accueil .....	29
3.2.2	S'inscrire .....	31
3.2.3	S'identifier.....	33
3.2.4	Panel Professeur .....	34
3.2.4.1	Demande.....	34
3.2.4.2	Création d'une incident .....	38

## LISTE DES FIGURES

Figure 1:Diagramme de Gantt.....	13
Figure 2:Diagramme de cas d'utilisation.....	17
Figure 3: Diagramme de séquence du cas d'utilisation 'Demande de s'inscrire' .....	18
Figure 4:Diagramme de séquence du cas d'utilisation ' Demande de réservation ' .....	19
Figure 5: Diagramme de séquence du cas d'utilisation ' Demande de document ou service ' .....	20
Figure 6:Diagramme de classe .....	21
Figure 7:le modèle MVC.....	25
Figure 8:Page d'accueil .....	29
Figure 9:Fonctionnalité .....	30
Figure 10:Contactez-nous .....	30
Figure 11:Demande de S'inscrire .....	31
Figure 12:Gestion d'erreurs pour l'inscription.....	32
Figure 13:Confirmation d'inscription .....	32
Figure 14:S'identifie .....	33
Figure 15:Panel Professeur.....	34
Figure 16:Types des demandes .....	35
Figure 17:Demande de sale .....	35

## **INTRODUCTION**

De nos jours, la technologie est indéniablement intégrée dans tous les aspects de notre vie.

Par conséquent, il est très important de maintenir et de diffuser l'implication technologique dans nos services en raison de ses nombreux avantages. Le principal avantage de l'utilisation de ces technologies dans toute administration est d'optimiser les aspects organisationnels et de gestion d'une telle organisation.

En outre, l'utilisation des dernières technologies fournir des informations en temps réel, des interfaces hautement personnalisables en fonction des besoins de l'utilisateur et une grande adaptabilité à n'importe quel domaine d'activité.

Dans ce rapport, on va discuter les exigences et les besoins de ce projet, qui sont les exigences fonctionnelles et non fonctionnelles pour fournir une orientation claire à l'utilisateur et les services que notre application va fournir.



## **CHAPITRE 1    CONTEXTE GENERALE DU PROJET**

Dans ce chapitre, nous allons présenter le contexte général de notre projet en décrivant son objectif, son intérêt et son cadre thématique, ainsi que la démarche et la conduite suivie pour sa réalisation.

### **1.1 Problématique**

La plupart des universités et des écoles supérieures ont tendance à être à l'ancienne et à traiter les demandes administratives de manière manuelle et inefficace. Une façon traditionnelle de recevoir et traiter les demandes et les incidents au sein de l'école, de se rendre à l'administration pour régler n'importe quelle affaire.

La manière complexe de passer pour recevoir un service augmente le temps de rendre le service et rend la tâche difficile pour le corps administratif et pour ceux qui demandent le service.

Les anciens styles de gestion à tendance à perdre la traçabilité des services rendus et des incidents non réglés.

Un absent de traçabilité et d'accès rapide au services de l'administration entrainera des malentendus ce qui cause une perte de temps et un fonctionnement mal géré des acteurs concernés.

### **1.2 Solution et critique**

-Par la communication informatisée entre l'administration les professeurs et les étudiants, on aura une traçabilité des services rendus avec tous les détails nécessaires ce qui va permettre d'améliorer le service à chaque fois.

-Faciliter la communication entre tous les intervenants.

-le temps nécessaire pour rendre le service ainsi que le déplacement des individus seront minimiser.

-L'administration aura la possibilité de surveiller et d'accéder à tous les services rendus grâce à l'historique de l'application. Catégories de technologies biométriques

Cahier des charges

### **1.3 Cahier des charges**

Afin de réussir un projet, il faut se lancer dès le départ sur une base solide. De ce fait, l'étape de spécification des besoins et l'élaboration du cahier des charges constitue la base de départ de notre travail.

L'élaboration d'un cahier des charges constitue une étape cruciale dans la réalisation d'une application donnée. En effet, c'est au cours de cette phase qu'on obtient une image précise et globale du produit à réaliser. Le futur d'une application dépend donc de cette phase, elle nous permet d'éviter le développement d'une application non satisfaisante et de réaliser un projet complet qui respecte les critères de qualité, de cout, et de temps.

Donc pour assurer ces objectifs, il est essentiel que nous parvenions à une vue claire des différents besoins escomptés de notre projet, c'est pour cela au niveau de ce cahier des charges nous essayerons d'éclater et de détailler toutes les fonctionnalités attendues du système.

Nous arrivons dans cette partie à spécifier les besoins fonctionnels attendue de notre application :

### ***Manipulation d'une demande.***

A travers une interface graphique simple et facile à utiliser, l'utilisateur (ici : Administrateur, Professeur ou Etudiant) peut créer, envoyer ou répondre à une demande de la façon suivante :

- Créer une demande de salle.
- Créer une demande de service.
- Créer une demande de document.
- Créer une demande simple (pour Administrateur).
- Répondre à une demande.

### ***Manipulation d'une incident.***

A travers une interface graphique simple et facile à utiliser, l'utilisateur (ici : Administrateur, Professeur ou Etudiant) peut créer, affecter (seul l'Administrateur) ou répondre (seul le Technicien) à une incident de la façon suivante :

- Créer une incident.
- Affecter une incident à un technicien.
- Répondre à une incident.

### ***Suivie du déroulement des actions.***

A travers une interface graphique simple et facile à utiliser, l'utilisateur à la possibilité de suivre les actions faites par lui ou diriger vers lui de la façon suivante :

- Voir les demandes, les incidents et les réponse envoyées par l'utilisateur.
- Consulter l'historique des actions avec détails (temps de réponse, message complet, date d'envoi, date de réponse, destinataire) de toutes les actions par l'administrateur.

- Consulter toutes les actions en cours d'exécution avec détails par l'Administrateur.

### **Manipuler les utilisateurs et les salles.**

A travers une interface graphique simple et facile à utiliser, l'administrateur a la possibilité de manipuler les salles et les utilisateurs de la façon suivante :

- Ajouter ou supprimer une salle.
- Ajouter ou supprimer un utilisateur (Professeur ou Etudiant).
- Approuver ou annuler la demande d'inscription d'un utilisateur.
- Cahier des charges

## **1.4 Gestion de projet**

Pour mieux planifier, analyser et contrôler le bon déroulement de la réalisation du projet, ainsi qu'assurer une bonne qualité du produit dans des délais fixés et une conformité entre ce qui est définie et ce qui est obtenu, nous avons utilisé les notions de la gestion des projets informatiques et le respect des normes qualité. Dans ce cadre nous allons présenter le diagramme de GANTT.

### ➤ **Diagramme de GANTT**

Le diagramme de GANTT est un outil permettant de modéliser la planification de tâches nécessaires à la réalisation d'un projet. Étant donné la relative facilitée de lecture des diagrammes GANTT, cet outil est utilisé par la quasi-totalité des chefs de projet dans tous les secteurs. Le diagramme GANTT représente un outil pour le chef de projet, permettant de représenter graphiquement l'avancement du projet, mais c'est également un bon moyen de communication entre les différents acteurs d'un projet.

	Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾	Predecessors ▾	Resource Names ▾
1	👉	▲ <b>Projet</b>	74 days	Mon 2/17/20	Mon 5/25/20		
2	👉	▲ <b>Etude préalable</b>	5 days	Mon 2/17/20	Fri 2/21/20		
3	👉	Etude de l'existant	2 days	Mon 2/17/20	Tue 2/18/20		Encadrant;MeganeYouness;MoufakkirZohair
4	👉	Etude de problème du projet	1 day	Wed 2/19/20	Wed 2/19/20	3	MeganeYouness;MoufakkirZohair
5	👉	Planing	2 days	Thu 2/20/20	Fri 2/21/20	4	MeganeYouness;MoufakkirZohair
6	👉	▲ <b>Analyse et Conception du projet</b>	3 days	Sat 2/22/20	Mon 2/24/20		
7	👉	Diagramme de cas d'utilisation	1 day	Sat 2/22/20	Sat 2/22/20	5	Encadrant;MeganeYouness;MoufakkirZohair
8	👉	Diagramme de séquence	1 day	Sun 2/23/20	Sun 2/23/20	7	Encadrant;MeganeYouness;MoufakkirZohair
9	👉	Diagramme de Classe	1 day	Mon 2/24/20	Mon 2/24/20	8	Encadrant;MeganeYouness;MoufakkirZohair
10	👉	▲ <b>Réalisation du projet</b>	45 days	Tue 2/25/20	Fri 4/24/20		
11	👉	Apprentissage	15 days	Tue 2/25/20	Fri 3/13/20	7,8,9	MeganeYouness;MoufakkirZohair
12	👉	Implémenter l'application	30 days	Mon 3/16/20	Fri 4/24/20	11	MeganeYouness;MoufakkirZohair
13	👉	▲ <b>Finalisation du projet</b>	6 days	Mon 4/27/20	Mon 5/4/20		
14	👉	Test et détection des erreurs	3 days	Mon 4/27/20	Wed 4/29/20	12	Encadrant;MeganeYouness;MoufakkirZohair
15	👉	Mise à jour de l'application	3 days	Thu 4/30/20	Mon 5/4/20	14	MeganeYouness;MoufakkirZohair
16	👉	▲ <b>Mise en oeuvre</b>	74 days	Mon 2/17/20	Mon 5/25/20		
17	👉	Rédaction du rapport	10 days	Tue 5/5/20	Mon 5/18/20	15	MeganeYouness;MoufakkirZohair
18	👉	Rédaction d'une présentation	5 days	Tue 5/19/20	Mon 5/25/20	17	MeganeYouness;MoufakkirZohair

Figure 1:Diagramme de Gantt

## 1.5 Conclusion

Dans ce chapitre, nous avons essayé d'analyser notre projet en élaborant un cahier de charges bien détaillé afin de mieux comprendre les exigences de notre future application, ce qui va nous éclairer la voie de bien choisir les meilleures technologies à utiliser, qui rendent l'application plus fluide.

## CHAPITRE 2 PHASES D'ANALYSE ET DE CONCEPTION

Dans ce chapitre, nous allons aborder, en premier temps, la démarche suivie pour entreprendre ce projet, la présentation de la méthode de modélisation utilisé dans l'analyse et la conception de notre application, à savoir la méthode UML qui a beaucoup d'avantages et offre divers diagrammes de conception, Par la suite, nous exposerons les différents diagrammes de conception de notre projet.

### 2.1 Choix de la méthode de modélisation

UML (Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié), un langage graphique de modélisation des données et des traitements. C'est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et est devenu désormais la référence en termes de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

#### ➤ Mise en œuvre d'UML

Pour programmer une application, il ne convient pas de se lancer directement dans l'écriture du code : il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. C'est cette démarche antérieure à l'écriture que l'on appelle modélisation ; son produit est un modèle. Nous avons adopté tout au long du projet le langage UML comme langage de modélisation et de conception. Ce choix est justifié par le fait qu'UML est un langage de modélisation de plus en plus utilisé grâce à la puissance des outils qu'il offre et l'efficacité de ses méthodes.

UML propose 13 diagrammes de deux types :

➤ **Diagrammes de comportement** : pour modéliser l'aspect dynamique du système, parmi ces diagrammes nous citons :

- Diagramme de séquence : Les diagrammes de séquence sont une solution de modélisation dynamique. La modélisation dynamique s'intéresse aux interactions se produisant à l'intérieur d'un système. Les diagrammes de séquence sont plus précisément consacrés aux « liens vitaux » d'un objet et comment ils communiquent avec d'autres objets pour accomplir une action avant que le lien vital ne s'interrompe.
- Diagramme d'activité : C'est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements. Le diagramme d'activité est également utilisé pour décrire un flux de travail.
- Diagramme de cas d'utilisation : sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation est une manière spécifique d'utiliser un système.

➤ **Diagramme de structure** : pour modéliser l'aspect statique du système. Parmi ces diagrammes nous citons :

- Diagramme de classe : Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes.

### 2.1.1 Conception et modélisation

Dans cette partie nous allons introduire le diagramme de cas d'utilisation qui va refléter l'ensemble des fonctionnalités du système de point de vue conception.

### 2.1.1.1 Diagramme de cas d'utilisation

Dans notre application, nous avons deux acteurs principaux **Administrateur**, **Utilisateur (Professeur, Etudiant)** et **Technicien**.

Acteur	Rôles
<i>Utilisateur</i>	<ul style="list-style-type: none"> <li>▪ Demande de réservation des salles</li> <li>▪ Demande des documents aux étudiants et aux l'administrateur</li> <li>▪ Demande des Services aux étudiants et aux l'administrateur</li> <li>▪ Créer une incident</li> <li>▪ Reçoive des demandes des documents</li> <li>▪ Reçoive des demandes des services</li> <li>▪ Afficher l'historique des taches effectue</li> <li>▪ Afficher leur profil</li> <li>▪ Modifier leur profil</li> </ul>
<i>Administrateur</i>	<ul style="list-style-type: none"> <li>▪ Même Rôles que les utilisateurs</li> <li>▪ Confirmer les comptes des nouveaux utilisateurs</li> <li>▪ Lister les salle/salle de Tp avec la possibilité d'ajouter des nouveaux ou de supprimer</li> <li>▪ Lister les utilisateurs (Étudiants, Professeurs, Techniciens) avec la possibilité d'ajouter des nouveaux ou de supprimer</li> <li>▪ Affectation des techniciens aux incidents des utilisateur(Prof/Etud)</li> <li>▪ Créer une incident avec le choix de technicien</li> <li>▪ Reçoive des demandes des documents</li> <li>▪ Reçoive des demandes des services</li> <li>▪ Afficher l'historique des tâches effectue</li> <li>▪ Consulter toutes les actions en cours d'exécution</li> </ul>
<i>Technicien</i>	<ul style="list-style-type: none"> <li>▪ Répons aux incidents</li> <li>▪ Imprimer des document</li> </ul>



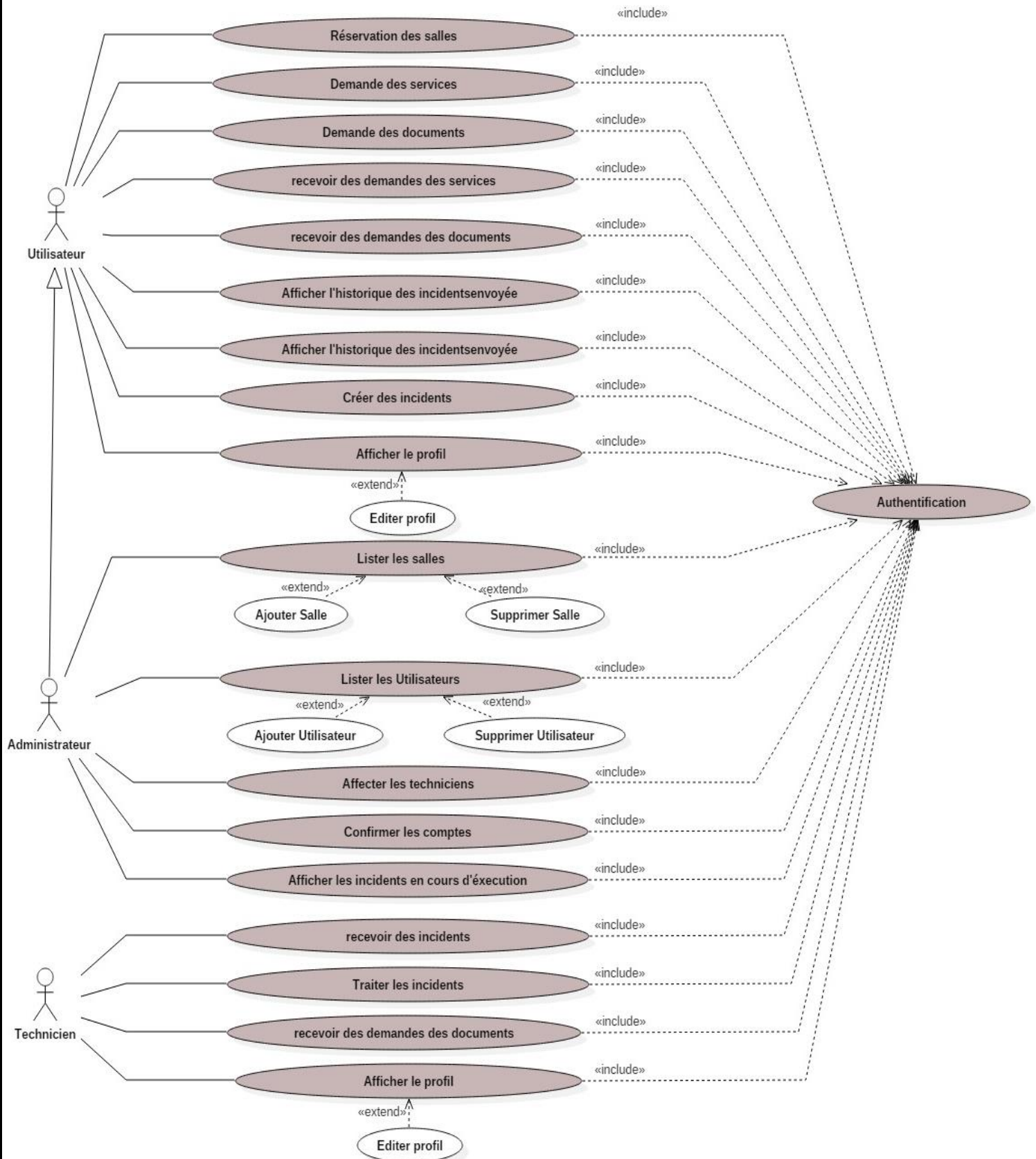


Figure 2:Diagramme de cas d'utilisation

### 2.1.1.2 Diagramme de séquence

Le diagramme de séquence permet d'illustrer les cas d'utilisation et de représenter les interactions dans le temps entre les objets du système.

➤ Diagramme de séquence du cas d'utilisation " Demande de s'inscrire "

1. l'utilisateur (Etudiant, Professeur, Technicien) demande de s'inscrire.
2. Le système envoie le formulaire d'inscription.
3. Le système vérifie la syntaxe des champs.
4. Si la syntaxe des champs est correcte : le système enregistre les informations, puis envoie un message de succès mais il faut accepter par l'administrateur.
5. Si la syntaxe des champs est incorrecte : le système vide les champs et renvoie le formulaire.

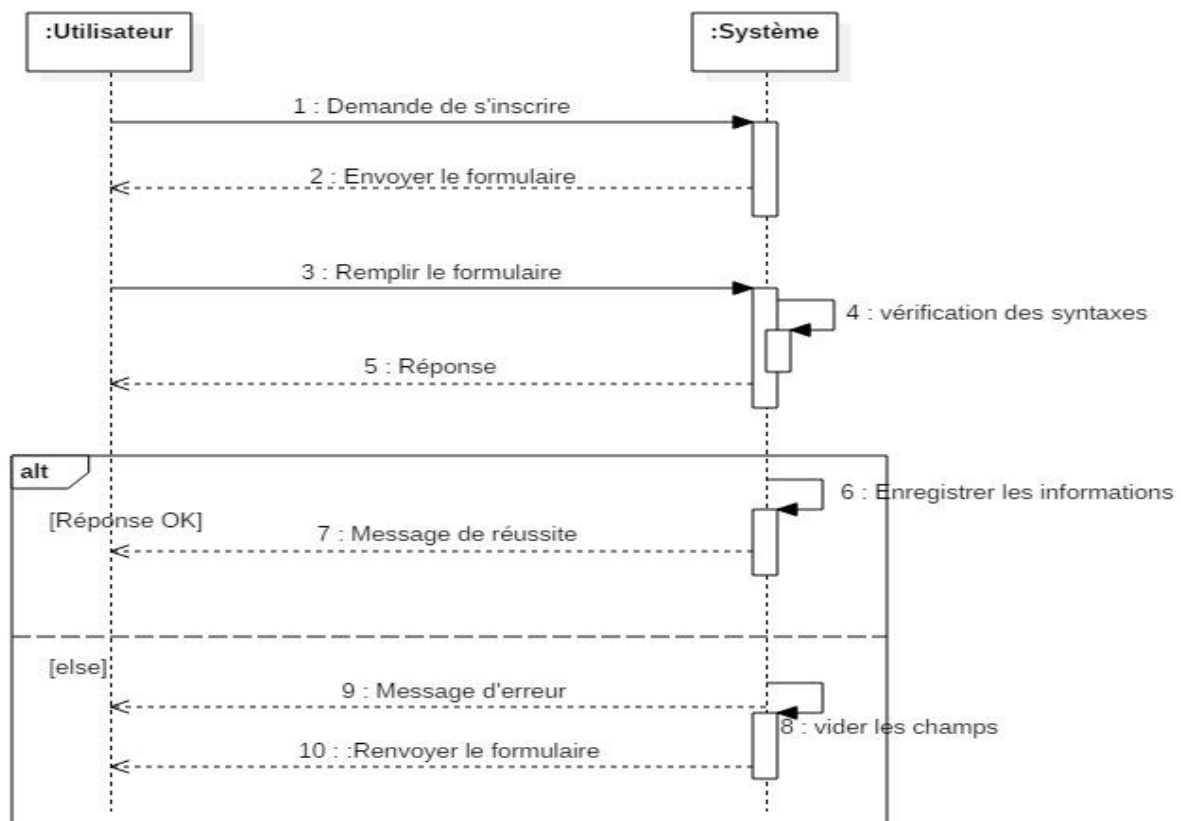


Figure 3: Diagramme de séquence du cas d'utilisation 'Demande de s'inscrire'

➤ Diagramme de séquence du cas d'utilisation " Demande de réservation "

1. L'utilisateur (Etudiant, Professeur) demande de réservation d'une salle.
2. L'administrateur vérifier la disponibilité de la salle demandée.
3. Si la salle est libre l'administrateur affecter la salle demandée et envoie un message de réservation.
4. Sinon l'administrateur envoie un message d'occupation.

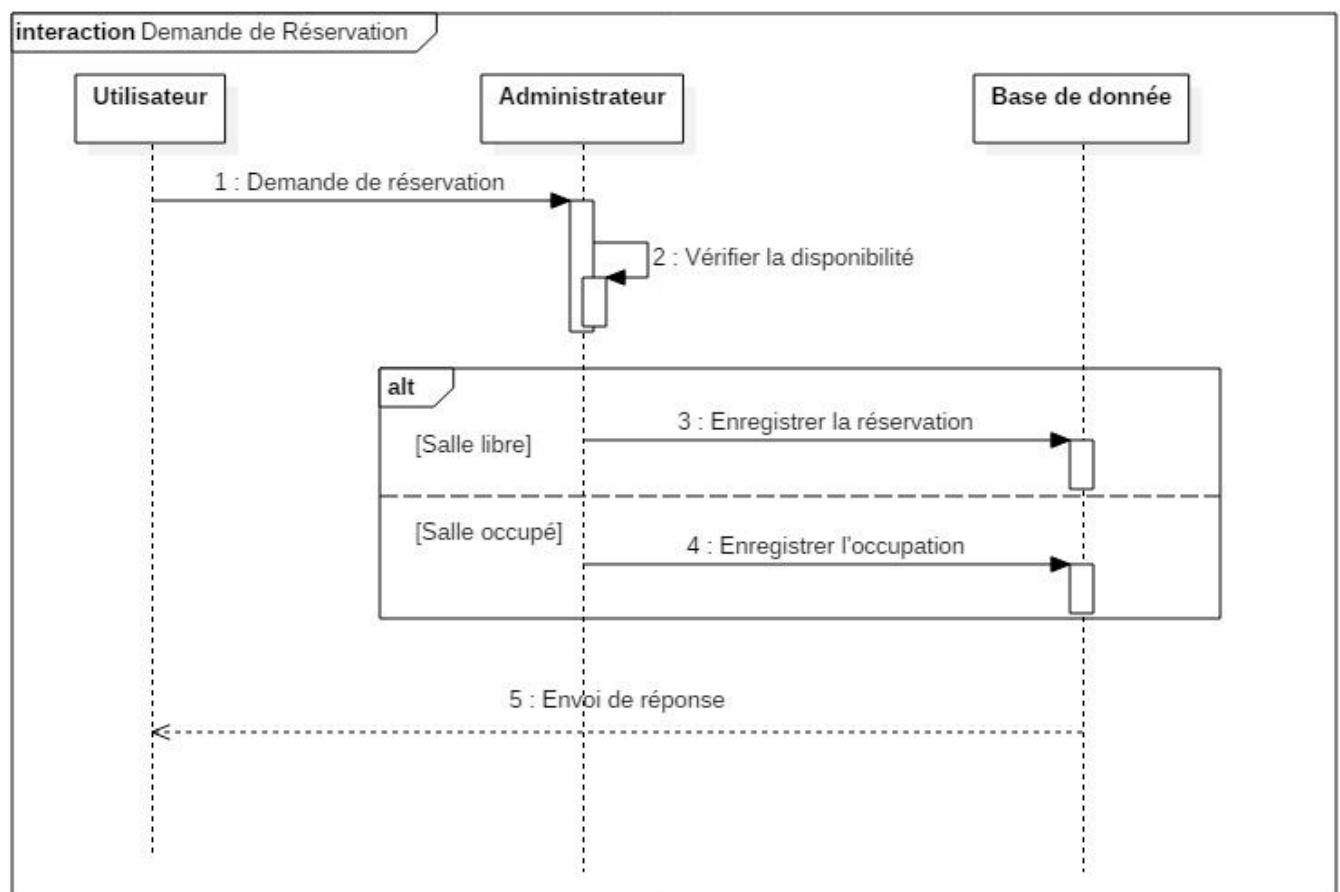


Figure 4:Diagramme de séquence du cas d'utilisation ' Demande de réservation '

- Diagramme de séquence du cas d'utilisation " Demande de document ou service"
- Le Destinateur demande de document ou un service à Destinataire.
- Le Destinataire livrer le document demandée ou service à la base de donne
- L'envoi de document demandée à le Destinateur.

### Remarque :

Le Destinateur peut être soit un Etudiant, Professeur ou Administrateur.

Le Destinataire peut être soit un Etudiant, Professeur ou Administrateur ou un Technicien.

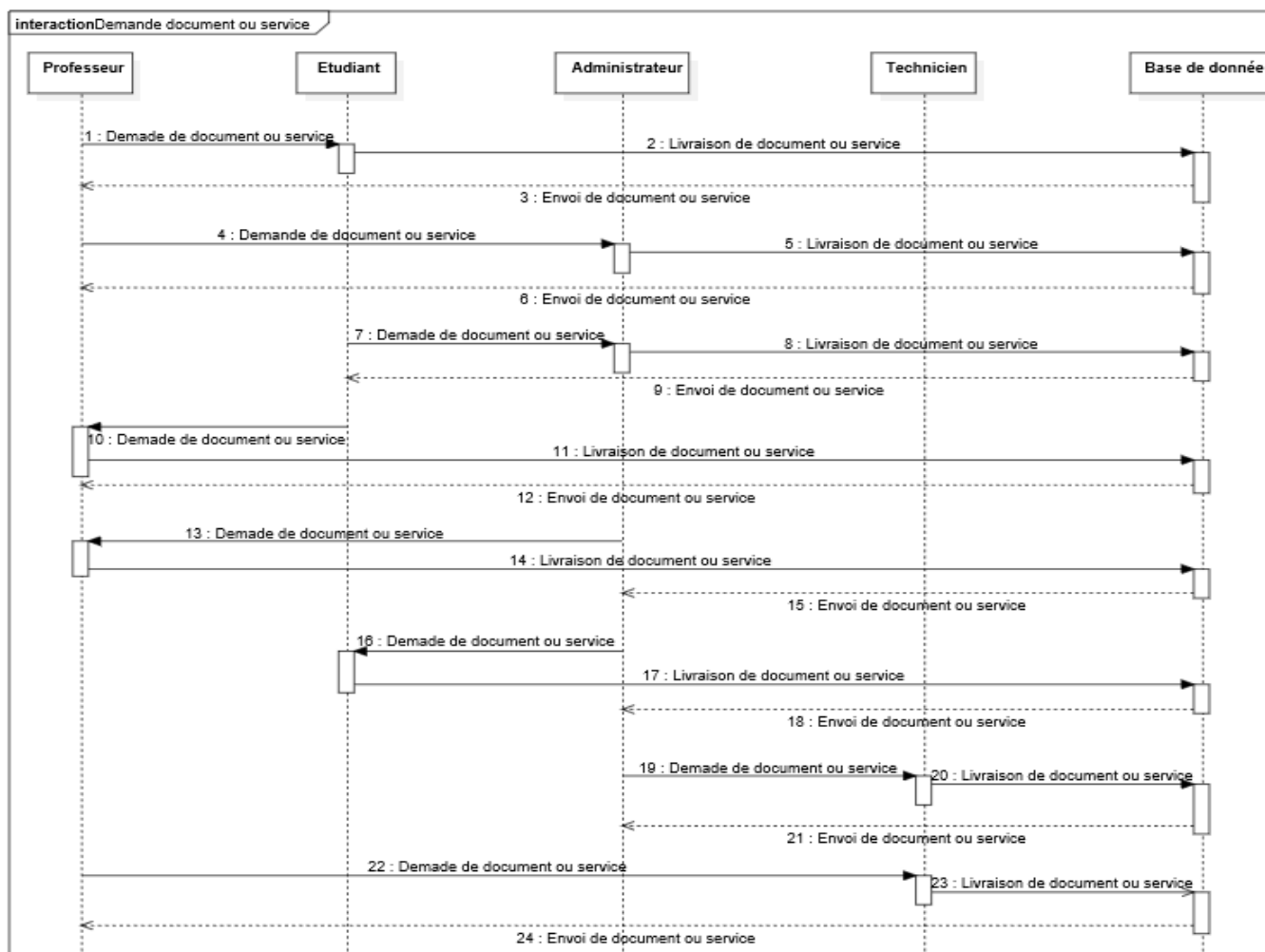


Figure 5: Diagramme de séquence du cas d'utilisation ' Demande de document ou service '

### 2.1.1.3 Diagramme de classe

C'est le point central dans le développement orienté objet. Il représente la structure statique du système sous forme de classes et de leurs relations. Les classes constituent la base pour la génération de code et des schémas de bases de données.

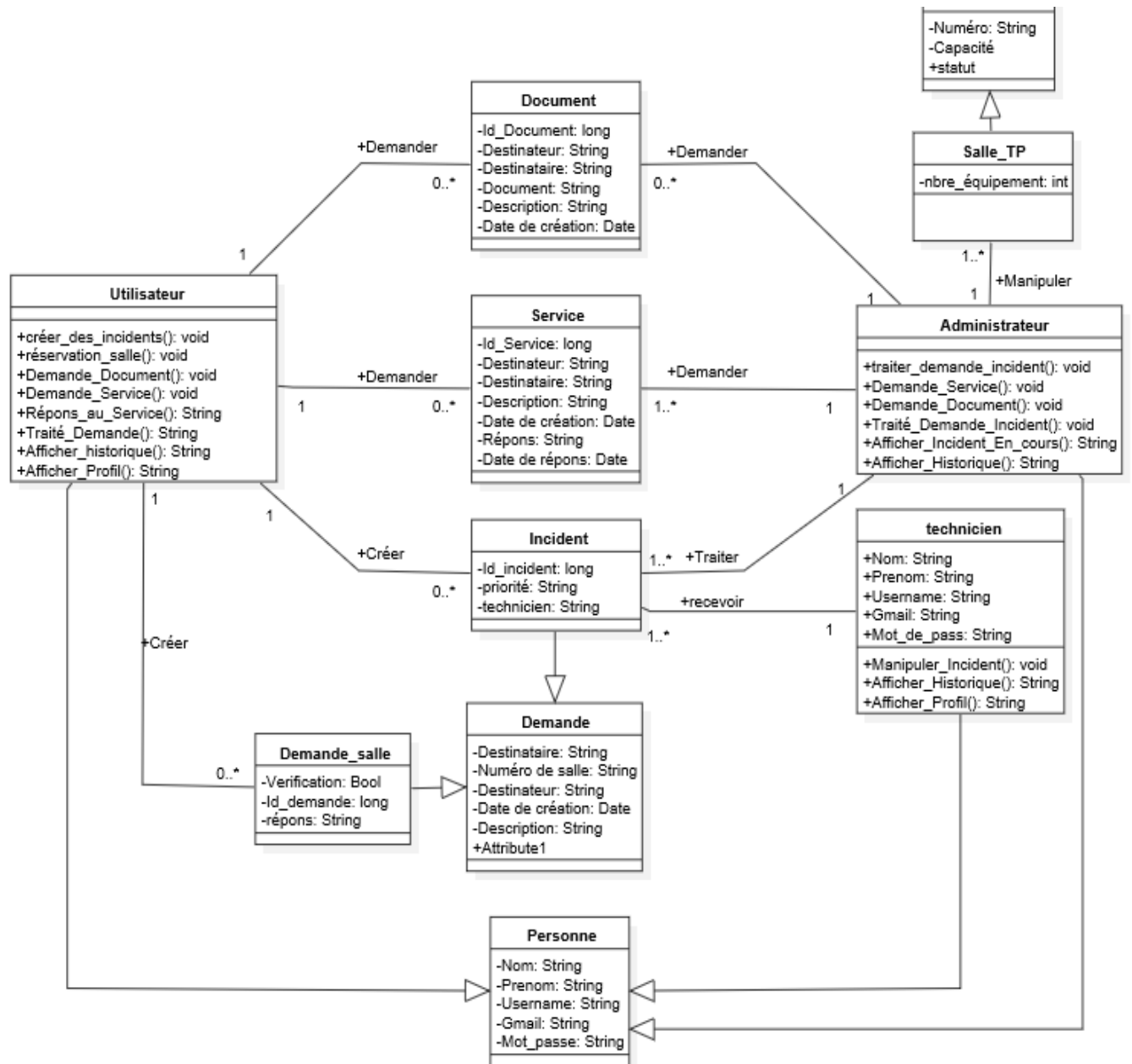


Figure 6:Diagramme de classe

### **2.1.2 Conclusion**

Dans ce chapitre, nous avons essayé de montrer l'architecture générale de notre projet, ainsi que la méthode de conception et de modélisation. Nous avons utilisé l'UML qui nous a offert les diagrammes de conception qui seront considérés comme un socle pour bien structurer notre projet. Le but de ce chapitre est de donner une vue détaillée afin de rendre les tâches plus claires et concrètes. Finalement, la conception nous a aidé à expliquer et mettre le projet compréhensible chez les lecteurs.

## **CHAPITRE 3    ENVIRONNEMENT DE TRAVAIL ET MISE EN PLACE DE L'APPLICATION**

Après l'analyse et la modélisation des différents cas d'utilisation du progiciel vient l'étape du développement des différentes fonctionnalités. Dans ce chapitre, nous allons commencer par une présentation des différents choix technologiques et outils logiciels utilisés pour développer notre application ENSIAS SERVICES ; Ensuite nous allons décrire les principales interfaces constituant l'application.

### **3.1 Présentation des solutions techniques**

#### **3.1.1 Architecture MVC**

Afin de garantir la maintenabilité de l'application, faciliter le travail en équipe et assurer une grande souplesse pour organiser le développement entre différents développeurs, on a choisi à suivre l'architecture MVC 2 (modèle, vue et contrôleur).

L'architecture MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et du contrôleur. Ce concept peut très bien être appliqué à un site internet. De nombreux sites internet disposent d'une plateforme d'administration pour modifier facilement les contenus. Il est bien évident que l'utilisateur qui va utiliser ce système pour créer des pages ne doit pas modifier la structure complète du site, ni les actions ni l'affichage. Pour qu'il ne modifie que les données, celles-ci doivent alors être isolées.

C'est sur ce constat que se base le système MVC. Chaque partie est bien distincte. Les données peuvent provenir d'une source quelconque et la vue peut être conçue par des webdesigners n'ayant pas de connaissance en programmation.

La différence entre MVC et MVC2 c'est que ce dernier il a un seul contrôleur qui se charge de rediriger la requête vers le bon traitement.

L'architecture MVC2 est donc décomposée en trois étapes :

#### **3.1.1.1 Modèle**

Le modèle représente les données et les règles métiers. C'est dans ce composant que s'effectuent les traitements liés au cœur du métier. Les données peuvent être liées à une base de données, myAdmin... Il est important de noter que les données sont indépendantes de la présentation. En d'autres termes, le modèle ne réalise aucune mise en forme. Ces données pourront être affichées par plusieurs vues. Du coup le code du modèle est factorisé : il est écrit une seule et unique fois puis réutilisé par chaque vue.

#### **3.1.1.2 Contrôleur**

Le contrôleur est l'élément qui va utiliser les données pour les envoyer à la vue. Son rôle est donc de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue (par l'utilisateur, exemple : afficher les derniers articles), puis de renvoyer à la vue les données afin d'être affichées.

Le contrôleur peut donc instancier différents objets (classe User, classe Articles, ...) qui enverront des requêtes vers la base de données ou récupéreront des données.

#### **3.1.1.3 Vue**

La vue correspond à des formulaires. Elle se contente d'afficher le contenu qu'elle reçoit sans avoir connaissance des données c'est le côté présentation de l'application



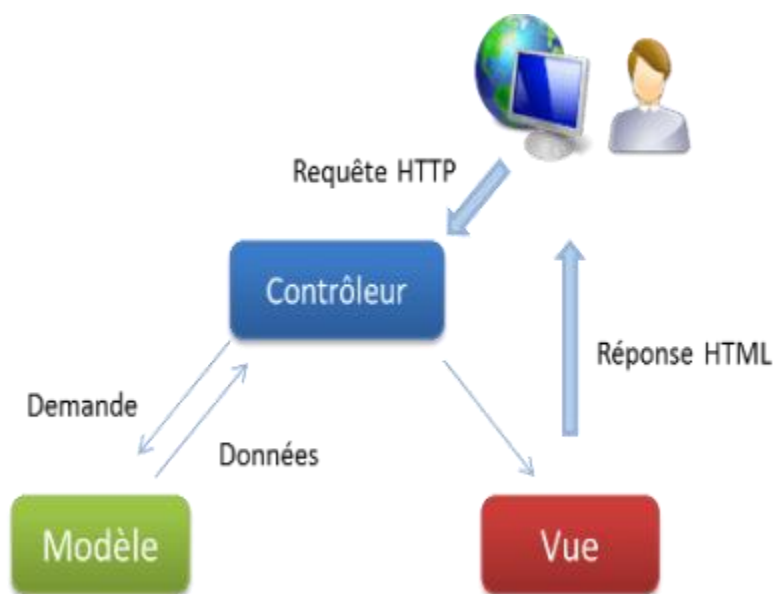


Figure 7:le modèle MVC

### 3.1.2 Technologie et Framework utilisée

#### 3.1.2.1 Technologies web



L'HyperText Markup Language : généralement abrégé HTML, est le langage de Balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom.

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs<sup>2</sup> avec l'utilisation (par exemple) de Node.js<sup>3</sup>. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs

permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés

Les feuilles de style en cascade<sup>1</sup>, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C).

### 3.1.2.2 Bootstrap



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

### 3.1.2.3 Serveur Apache

Deux serveurs distants utilisés dans cette application : un serveur web et un serveur de base de données.

- APPACHE est comme serveur web utilisé, permet de se propager les scripts PHP.
- Le serveur de base de données utilisé est le serveur MySQL, son rôle est le stockage de données.

#### 3.1.2.3.1 Apache



C'est un serveur web, il est un programme connecté internet dont le rôle est d'attendre les requêtes des clients (les navigateurs web qui lui demandent des pages web via le protocole HTTP, d'où le nom de serveur HTTP), et d'y répondre. Ainsi, Apache permet d'accéder des fichiers

statiques sur un serveur : images (. JPG, Ping), documents html, documents word, archive zip, vidéo (.mp4) ...

#### 3.1.2.3.2 MySQL



MySQL est un serveur puissant de base de données open source intégré basé sur un système de gestion de base de données relationnelle (SGBDR) et capable de gérer une grande base de données de connexion simultanée. C'est la base de données open source la plus plébiscitée au monde. De par sa performance, sa fiabilité et sa facilité d'utilisation éprouvées, MySQL s'est imposée comme le choix évident de base de données pour les applications Web, et il fait partie du WAMPER.

#### 3.1.2.3.3 PHP



PHP est une open source langage de script rapide et riche en fonctionnalités pour développer des applications Web ou Internet / Intranet Applications. Principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de fa on locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

#### 3.1.2.4 MS-PROJECT



Est un logiciel de gestion de projets édité par Microsoft. Il permet la planification d'un projet : il est possible à tout moment créer des tâches et des jalons, définir les liens entre chaque tâche, les hiérarchiser. MS Project a également la capacité d'estimer la durée ainsi que la charge de travail nécessaire pour accomplir une tâche définie. Il permet

aussi la création de modèles qui permet à l'utilisateur de respecter une méthodologie ou un processus quelconque. Le projet peut être représenté graphiquement de différentes manières : diagramme de Gantt, réseau des tâches...

### 3.1.3 Outils de développement et de modélisation

#### 3.1.3.1 StarUML



StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose1 ou Together2. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation.

## 3.2 Présentation de l'application

Nous avons essayé dans la réalisation de l'application de répondre aux facteurs de qualité à savoir :

- **Maintenabilité** : Minimiser l'effort pour localiser et corriger une anomalie, c'est pour cela on a opté au modèle MVC (Modèle Vue Contrôleur) qui facilite la maintenance du code.
- **Conformité** : Contenir minimum d'erreur à satisfaire aux spécifications et à remplir ses missions, les spécifications définies au départ doivent être conforme à ce qui est réalisé.
- **Adaptabilité** : Minimiser l'effort nécessaire pour le modifier par suite d'évolution des spécifications.
- **Testabilité** : Faciliter les procédures de test permettant de s'assurer de l'adéquation des fonctionnalités.

### 3.2.1 Page d'accueil

#### ▪ *Accueil*

Lorsqu'un visiteur de L'ENSIAS naviguer sur un navigateur et cherche une application de gestion des services scolaires il va trouver notre application, et la première chose qui va voir c'est la page d'accueil.

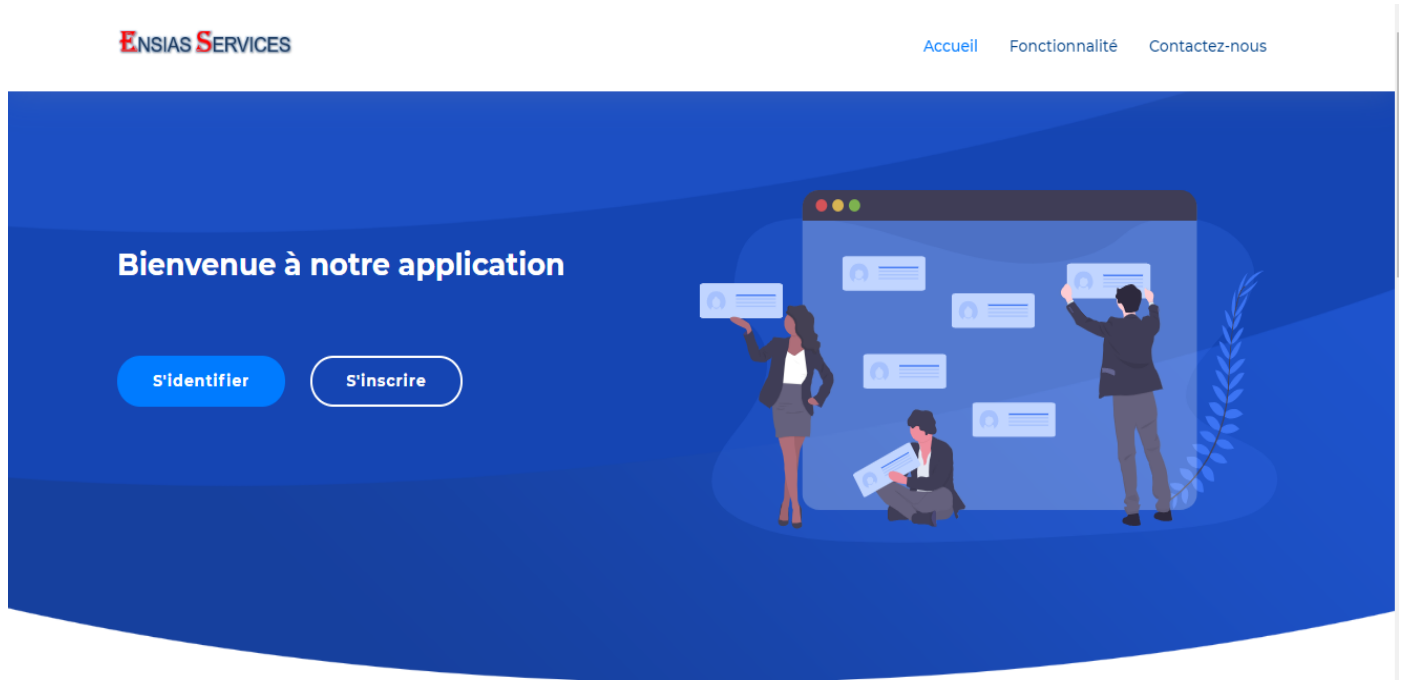


Figure 8:Page d'accueil

## ▪ Fonctionnalité

Lorsqu'on clique sur la bouton service elle va nous ramener dans l'image ci-dessous, et à partir de c'est dernier on peut savoir les fonctionnalités de l'application.



Figure 9:Fonctionnalité

## ▪ Contactez-nous

Parfois un visiteur veut contacter l'administrateur de site dans le cas d'un problème ou bien il veut quelle qu'information concernant notre site, donc c'est mieux de réserver une place dans la page d'accueil pour qu'il puisse nous contacter, et c'est ça le rôle du bouton Contactez-nous.

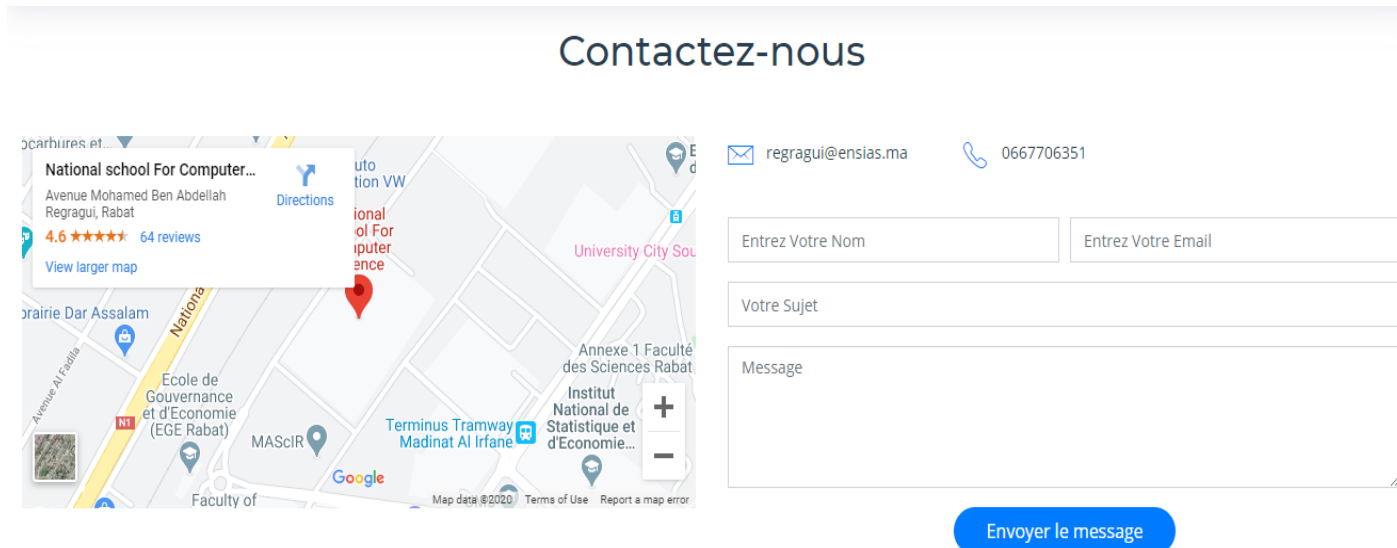


Figure 10:Contactez-nous

### 3.2.2 S'inscrire

Lorsque l'utilisateur n'admet pas de compte, il pourrait accéder aux fonctionnalités d'ajout d'un membre tout en saisissant quelques informations personnelles (nom d'utilisation, Nom, Prénom, Email, Mot de passe) et les paramètres d'accès (Professeur, Etudiant, Technicien) :

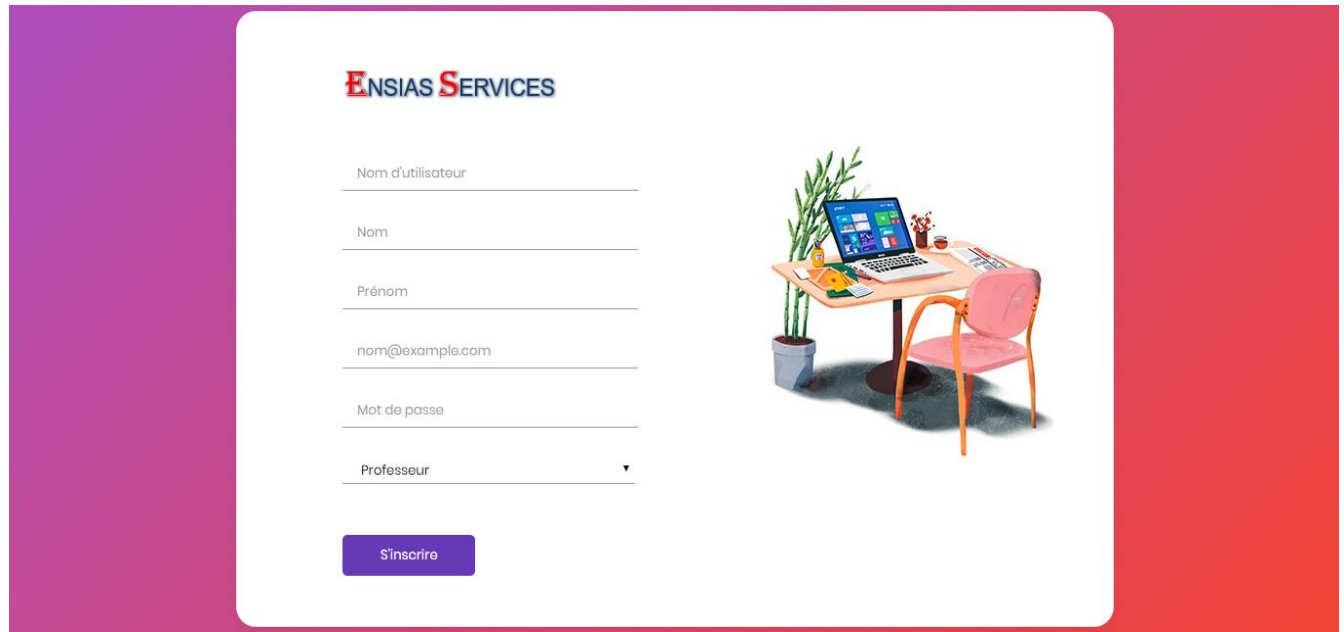
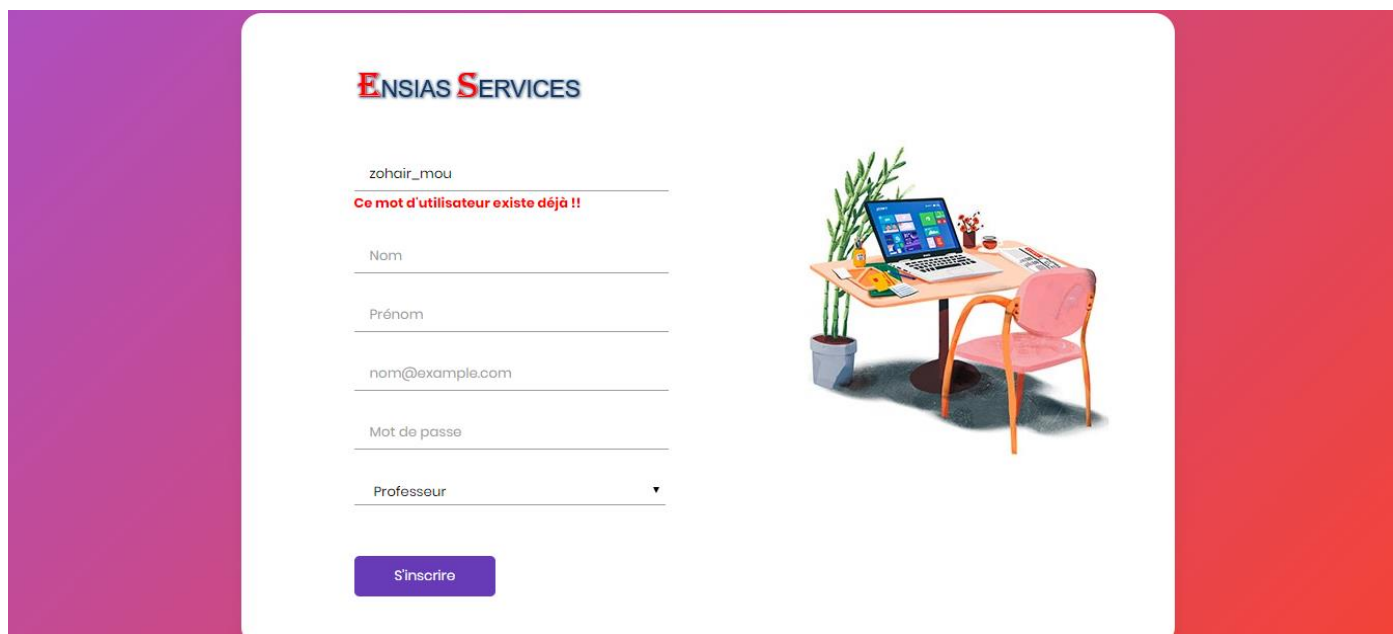
The image shows a registration form for 'ENSIAS SERVICES'. The form is set against a white background with a purple-to-red gradient border. It includes input fields for 'Nom d'utilisateur', 'Nom', 'Prénom', 'Email' (with the placeholder 'nom@example.com'), 'Mot de passe', and a dropdown menu for 'Professeur'. A purple 'S'inscrire' button is at the bottom. To the right of the form is an illustration of a desk with a laptop, a potted plant, and a pink chair.

Figure 11: Demande de S'inscrire

- **Gestion d'erreurs :**

Lorsque l'utilisateur tape un nom d'utilisateur qui existe déjà, un message d'erreur s'affiche à l'utilisateur pour ne perdre pas du temps à remplir la totalité du formulaire.



**ENSIAS SERVICES**

zohair\_mou

**Ce mot d'utilisateur existe déjà !!**

Nom

Prénom

nom@example.com

Mot de passe

Professeur ▼

S'inscrire




Figure 12: Gestion d'erreurs pour l'inscription

Après l'inscription d'un utilisateur, il sera redirigé vers la page de confirmation d'inscription tant que l'administrateur n'a pas encore accepté sa demande d'inscription.



**ENSIAS SERVICES**



**FÉLICITATIONS**

**compte créé**

félicitations votre compte est bien crée dans l'attente d'être accepté par l'administrateur

retour à la page d'accueil

Figure 13: Confirmation d'inscription



### 3.2.3 S'identifier

Après la création de son compte l'utilisateur peut l'accéder et bénéficier des services de l'application en entrant son nom d'utilisateur et son mot de passe. Et selon son statut il va être rediriger vers le panel convenable.

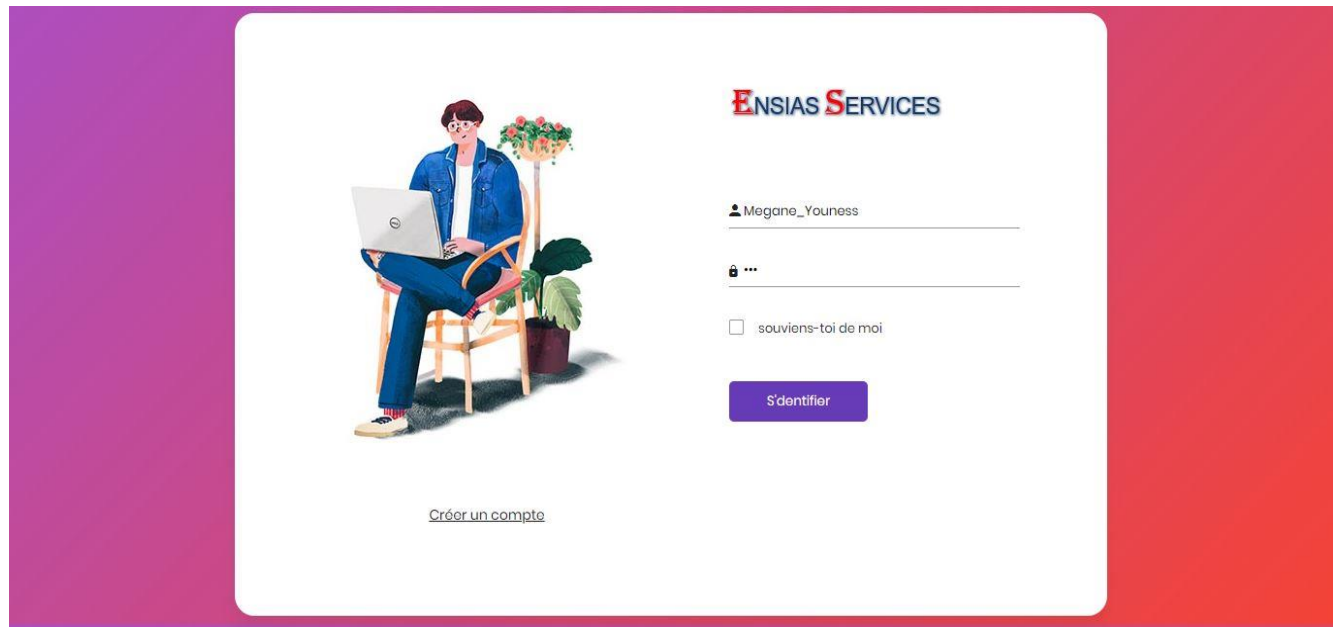


Figure 14:S'identifie

### 3.2.4 Panel Professeur

Lorsqu'un professeur s'identifie il va être dirigé vers ce panel ce tableau de bord affiche le nom de l'utilisateur connecté en haut droite de l'écran aussi que le nombre de demandes et incidents créés par cet utilisateur selon le type durant son activité dans le site dès la création de son compte. En haut à gauche on a le logo du site et au-dessous de lui le menu de l'application.

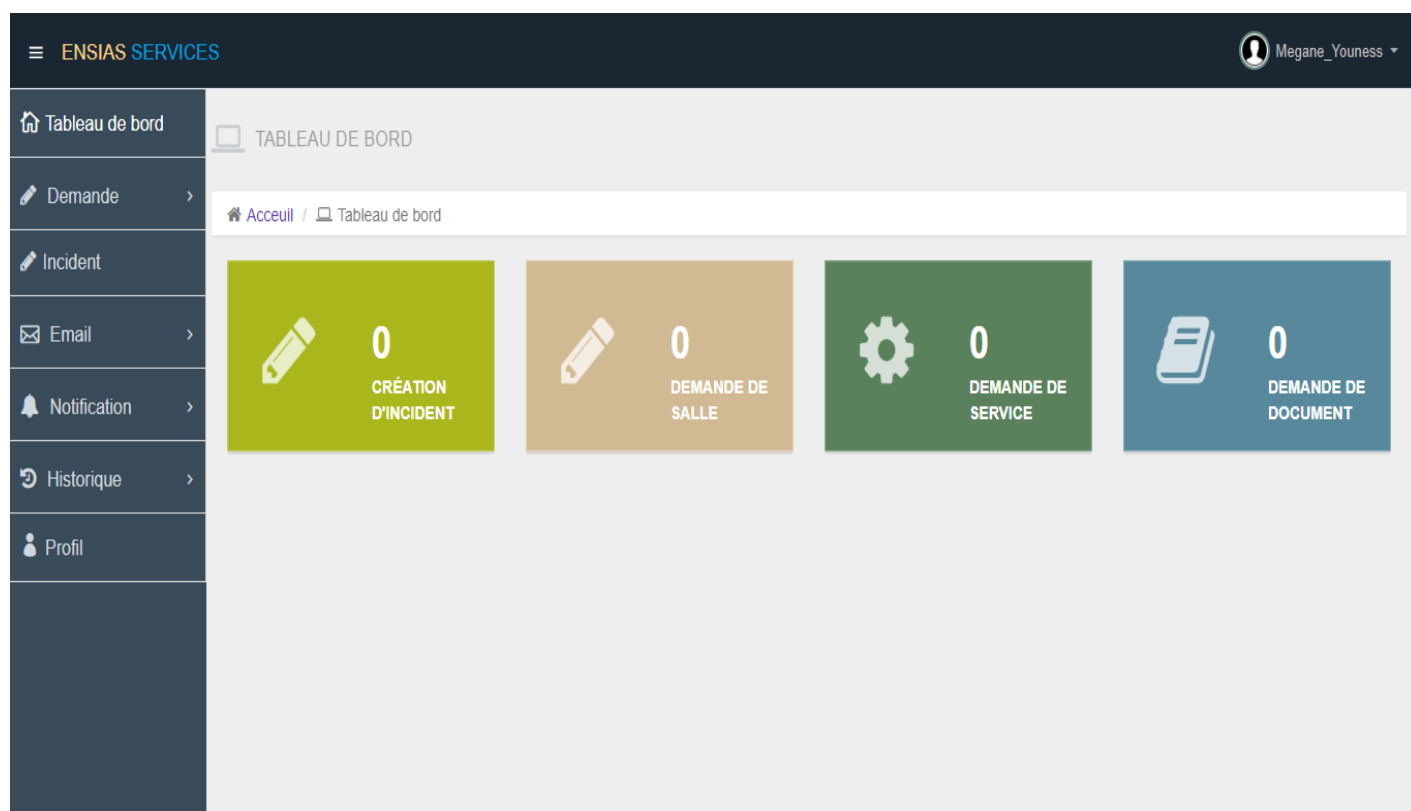


Figure 15:Panel Professeur

#### 3.2.4.1 Demande

En cliquant sur demande dans le menu, il s'affiche trois cases sont salle, document et service, en cliquant sur l'un de ces boutons l'application affiche un formulaire pour créer la demande spécifiée.

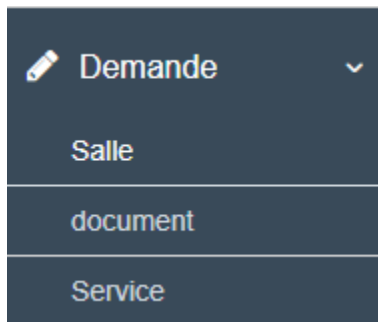


Figure 16:Types des demandes

#### 3.2.4.1.1 Demande de Salle

Le formulaire de la demande de sale, où le destinataire est fixé (le nom d'utilisateur connecté) donne la liste les salles de l'établissement pour choisir une et la zone de description pour décrire la raison de la réservation, la date et l'heure et autres choses, et en dessous deux boutons, un pour affirmer et envoyer la demande et l'autre pour l'annuler.

ENSIAS SERVICES

Megane\_Youness

Tableau de bord

DEMANDE

Accueil / Demande / Salle

Demande d'une salle

Destinateur: Megane\_Youness

Numéro de salle: L15

Description: Bonjour monsieur SVP je voudrais la salle numéro L15 demain à 08:30

Envoyer Annuler

Figure 17:Demande de sale

### 3.2.4.1.2 Demande de Document

Le formulaire du demande de document, où le destinataire est fixé (le nom d'utilisateur connecté) donne une liste des destinataire possible pour choisir un pour éviter l'erreur d'écrire un nom qui n'existe pas ou qui n'a pas le droit de recevoir cette demande, un champ de texte pour écrire le nom du document demandé et une zone de description pour décrire la raison de la demande le nombre de copies t autres choses, et en dessous deux boutons, un pour affirmer et envoyer la demande et l'autre pour l'annuler.

The screenshot shows the 'Demande de Document' form within the ENSIAS SERVICES application. The interface includes a dark sidebar with navigation links: Tableau de bord, Demande, Salle, document, Service, Incident, Email, Notification, Historique, and Profil. The main content area has a header 'DOCUMENT' and a breadcrumb trail 'Accueil / Demande / Document'. The form itself is titled 'Demande une document' and contains the following fields:

- Destinateur:** A text input field containing 'Megane\_Youness'.
- Destinataire:** A dropdown menu with 'Moufakkir Zohair' selected.
- Document demandé:** A text input field containing 'Emploi de temps'.
- Description:** A large text area containing the text 'Bonjour Monsieur SVP je voudrais l'emploi du temps de 1A'.

At the bottom of the form are two buttons: 'Envoyer' (blue) and 'Annuler' (white).


### 3.2.4.1.3 Demande de Service


Le formulaire du demande de service, où le destinataire est fixé (le nom d'utilisateur connecté) donne une liste des destinataire possible pour choisir un pour éviter l'erreur d'écrire un nom qui n'existe pas ou qui n'a pas le droit de recevoir cette demande et une zone de description pour décrire le service demander et la raison, et en dessous deux boutons, un pour affirmer et envoyer la demande et l'autre pour l'annuler.


The screenshot shows the 'Demande d'une service' form in the ENSIAS SERVICES application. The interface includes a dark sidebar with navigation links: Tableau de bord, Demande, Incident, Email, Notification, Historique, and Profil. The main content area has a header 'SERVICE' and a breadcrumb trail 'Accueil / Demande / Service'. The form itself is titled 'Demande d'une service' and contains three input fields: 'Destinateur' (pre-filled with 'Megane\_Youness'), 'Destinataire' (a dropdown menu showing 'Moufakkir Zohair'), and 'Description' (a text area containing 'Bonjour monsieur S'ils vous plait demain je serais absent'). At the bottom of the form are two buttons: 'Envoyer' (blue) and 'Annuler' (white).


### 3.2.4.2 Création d'un incident


≡ ENSIAS SERVICES


 Megane\_Youness ▾


 Tableau de bord


 Demande >

 Incident



 Email >

 Notification >

 Historique >

 Profil

INCIDENT

 Accueil /  Incident

Création de l'incident

Destinateur:

Megane\_Youness

sujet:

Matériel

Numéro de salle:

L15 ▾

Description:

Bonjour monsieur le projecteur dans la salle L15 est défaillant

Envoyer