

Université Ibn Tofail
Faculté des Sciences
Département d'Informatique



LICENCE FONDAMENTALE EN SCIENCES
MATHÉMATIQUES ET INFORMATIQUES

PROJET DE FIN D'ETUDE

APPLICATION WEB PROFILEE POUR AFFICHAGE ET
MODIFICATION DES EMPLOIS DU TEMPS

Réalisé par :

- Rim Jamaa
- Salma Abdillah

Encadré par :

- Mr Khalid Benlhachmi

Jury :

- Mme Khadija Louzaoui

Année Universitaire : 2022-2023

REMERCIEMENT

Nous tenons à exprimer nos profonds remerciements avec un grand plaisir et un grand respect à notre encadrant Pr. **Khalid Benlhachmi** pour ses conseils, sa disponibilité et ses encouragements qui nous ont permis de réaliser ce travail.

Nous adressons aussi nos vifs remerciements et notre gratitude à Pr. **Khadija Louzaoui** pour avoir bien voulu examiner et juger ce travail.

Nous exprimons de même notre gratitude envers tous ceux qui nous ont accordé leur soutien, tant par leur gentillesse que par leur dévouement.

A tous les enseignants qui nous ont aidés pendant les trois ans passés à l'université Ibn Tofail.

A toute personne ayant contribué de près ou de loin à l'avancement de notre projet.

Nous ne pouvons nommer ici toutes les personnes qui nous ont aidés et encouragés de près ou de loin et nous les remercions vivement.

RESUME

Dans le cadre de l'obtention de notre diplôme de Licence Fondamentale en Sciences Mathématiques et Informatique (SMI) à l'Université « **IBN TOFAIL** » **Kénitra**, nous avons été appelés à réaliser un projet de fin d'études afin d'illustrer tout ce qu'on a étudié au long de notre formation. C'est ainsi que nous avons eu l'occasion d'approfondir nos connaissances théoriques par la conception et la réalisation d'une application de gestion des emplois du temps , la modification et l'affichage des emplois au sein de la faculté des sciences de Kénitra (FSK) sous le tutorat de **Monsieur Khalid Benlhachmi**.

Pour ce qui concerne le volet technique, nous avons utilisé un ensemble d'outils tels que : **Html, CSS, JavaScript, JQuery, PHP, MySQL**.

TABLE DES MATIERES

REMERCIEMENT	2
RESUME	3
TABLE DES FIGURES.....	5
INTRODUCTION GENERALE.....	6
CHAPITRE 1 : CADRE GLOBAL	8
1- CADRE DU PROJET.....	8
2-ETUDE DU CAS.....	8
3- PROBLEMATIQUE :	8
1.3-BESOINS FONCTIONNELS :	9
4- SOLUTIONS PROPOSEES :	10
4.1-PRESENTATION DU PATTERN « MVC » :	10
CHAPITRE 2 : CONCEPTION DU SYSTÈME	12
1-MODELISATION DU SYSTEME :	12
1.1 - DIAGRAMME DE CAS D'UTILISATION :	12
1.2- DIAGRAMME DE CLASSE :	13
1.3- DIAGRAMME DE SEQUENCE :	14
2- NOYAU FONCTIONNEL DU SYSTEME :	16
3- INTERFACE DU SYSTEME :	19
3.1 UTILITE DU PATTERN « MVC » :	19
3.2 PRESENTATION DU MODELE D'ABSTRACTION	20
3.3 PRESENTATION DU CONTROLEUR.....	21
3.4 PRESENTATION DES VUES	23
CHAPITRE 3 : REALISATION ET PRESENTATION	25
1- INTRODUCTION :	25
2- OUTILS DE DEVELOPPEMENT :	25
3-PRESENTATION DE L'APPLICATION.....	27
3.1 PAGE D'ACCUEIL.....	27
3.2 PRESENTATION DE LA PLATEFORME.....	28
3.3 FORMULAIRE DE CONTACT :	29
3.4 FOOTER :	29
3.5 PREMIER PROFIL : PROFIL ETUDIANT	30
3.6 DEUXIEME PROFIL : PROFIL PROFESSEUR	32
3.7 TROISIEME PROFILE : PROFIL ADMINISTRATEUR.....	33

4- CONCLUSION	37
CONCLUSION ET PERSPECTIVES	38
BIBLIOGRAPHIE	39

TABLE DES FIGURES

Figure 1: Diagramme de cas d'utilisation	12
Figure 2: Diagramme de classe	13
Figure 3: Diagramme de séquence (Authentification).....	14
Figure 4: Diagramme de séquence (Update Data)	15
Figure 5: Représentation de la base de données	16
Figure 6: Concepteur des tables sur PhpMyAdmin	17
Figure 7: Schéma explicatif du pattern MVC	20
Figure 8: Table représentative des outils et langages utilisés	26
Figure 9: Page d'accueil	27
Figure 10: Page Présentation.....	28
Figure 11: Formulaire de contact	29
Figure 12: Footer	29
Figure 13: Page d'accueil "ETUDIANT"	30
Figure 14: "Pop up".....	30
Figure 15: Page planning ETUDIANT	31

Figure 16: Page d'accueil PROFESSEUR	32
Figure 17: Page planning PROFESSEUR	33
Figure 18: Page d'accueil ADMIN	33
Figure 19: Modification des séances	35
Figure 20: Ajouter utilisateur.....	36
Figure 21: Ajouter module	37

INTRODUCTION GENERALE

Pour acquérir une bonne et parfaite qualité, la formation théorique à elle seule ne suffit pas, il est donc nécessaire de suivre une démarche réelle permettant de voir comment se déroulent les tâches dans la vie professionnelle. A cet effet, afin de valider nos études acquises au fil de trois ans au sein de notre Faculté des Sciences et en vue de l'obtention de la License en Informatique, nous sommes appelées à effectuer un projet de fin d'études. Ce PFE permet également de s'introduire dans la vie pratique de notre domaine et son environnement, et d'acquérir une première expérience professionnelle. L'objectif de notre PFE est de développer une application web de « GESTION DES EMPLOIS DU TEMPS UNIVERSITAIRES » afin de gérer les séances, les modules,

ajouter des utilisateurs et afficher les plannings. Dans ce rapport, nous volons présenter dans un premier temps la problématique et les solutions proposées, puis, la deuxième partie va exposer la conception, la méthodologie, et la modélisation du projet et enfin un troisième chapitre dont on cite les outils et les langages utilisés et aussi un ensemble de paragraphes présentant les interfaces graphiques de l'application avec une description de chacune d'elles.

CHAPITRE 1 : CADRE GLOBAL

1-CADRE DU PROJET

Notre projet s'intitule "Mise en place d'une application web de gestion des emplois du temps universitaires".

2-ETUDE DE CAS

Cette étape est primordiale pour la mise en route de tout projet informatique ou autre, et qui permet de définir le contexte de fonctionnement, ou bien le processus métier, et de dégager les différentes imperfections dans le système actuel afin de les corriger. Pour faire cette étude, il nous était indispensable de collecter les informations nécessaires avant le démarrage du projet.

3- PROBLEMATIQUE :

Notre projet a pour objectif de développer une application web permettant d'afficher et modifier les plannings universitaires de façon automatisée en respectant un système d'utilisation profilé.

Suite à une observation approfondie aux fonctionnements des emplois du temps universitaires, on a pu relever divers problèmes :

- Les emplois du temps générés par l'administration sont conçus par semestre, et vue que l'enseignement universitaire fonctionne par module. Un étudiant peut avoir des modules dans plusieurs semestres. Ce qui entraîne un travail laborieux à l'étudiant pour récupérer son emploi du temps. La grande majorité d'eux font même recours à des plateformes de planning pour créer leur propre calendrier.
- Le changement des emplois est un grand souci, les emplois du temps ne se modifient pas à la même cadence que les changements sont effectués en réalité. Si le

changement ne concerne qu'une semaine précise et ne sera pas définitive, il ne sera pas pris en considération dans les changements de ses emplois.

- Le format de ces emplois aussi cause un problème. Généralement ils sont en format PDF dans des fichiers google drive. Donc pour chaque modification il faut créer un nouveau fichier et supprimer l'ancien. Ce qui veut dire qu'à chaque fois que l'étudiant veut revoir son emploi, il doit rechercher le lien de consultation de ce fichier.
- Une capture d'écran qui sera au final perdue dans sa galerie, et il sera obligé de refaire l'opération à chaque fois.
- Les emplois générés par l'administration sont accessibles au grand public. Ce qui peut causer des problèmes de sécurité de l'information (Risque de modification par des inconnues).

Notre projet tente de remédier à un ensemble de besoins fonctionnels.

1.3-BESOINS FONCTIONNELS :



Les besoins fonctionnels se rapportent aux fonctionnalités que l'application en question doit offrir pour satisfaire les utilisateurs.

Les fonctionnalités que doit intégrer l'application à développer peuvent être décrites comme suit :

- L'application offre aux étudiants et aux professeurs une interface qui leur permet de consulter leurs informations personnelles.
- L'application offre au professeur une interface qui lui permet de consulter son emploi du temps personnalisé selon les filières qu'il désire.
- L'application offre à l'étudiant une interface qui lui fournit un emploi du temps automatisé selon les modules qu'il désire, maintenable par l'administrateur.
- Le système doit permettre à l'utilisateur de saisir son login et son mot de passe pour accéder au système. Cette opération assure la sécurité du système et limite l'accès à la plateforme.

Cet ensemble de besoins doit respecter un ensemble de contraintes notamment :

- L'accès à la base de données doit être rapide et sécurisée
- Permettre aux utilisateurs de se familiariser rapidement avec le contenu
- Être simple et compréhensible
- L'application doit être toujours effective
- Temps de réponse minimum

4- SOLUTIONS PROPOSEES :

Mettre en place un nouveau système de Gestion des emplois sous forme d'une application web facilitant le partage des plannings des cours, TD, TP... Cette application va permettre :

- ✓ Minimiser le temps relatif au traitement des données.
- ✓ Simplifier la tâche des utilisateurs.
- ✓ Centraliser les données dans un seul endroit (base de données unique) qui sera partagée par tous les modules de l'application.
- ✓ L'enregistrement des utilisateurs qui vont utiliser l'application.

Pour maîtriser la complexité du problème et faciliter la maintenance de la solution proposée, nous avons adoptées à une modélisation orienté objet :

- Dans la conception du système, on a utilisé le pattern MVC.
- Dans la phase programmation on a adopté la langage PHP.

4.1-PRESENTATION DU PATTERN « MVC » :

Le pattern MVC (Modèle-vue-contrôleur) est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

- **Modèle** : cette partie gère les données de notre application. Son rôle est d'extraire les informations des tables figurant dans la base de données sous forme de classe, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur.
- **Vue** : cette partie se concentre sur l'affichage, elle contient les différentes interfaces de nos utilisateurs. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher
- **Contrôleur** : C'est l'intermédiaire entre le modèle et la vue. C'est lui qui reçoit la requête de l'utilisateur et qui contacte les autres modules pour échanger des informations avec eux.

Ayant présenté les outils et les méthodes adoptés, nous allons exposer en détails tout ce qui était introduit précédemment, dans les chapitres qui suivent :

Le deuxième chapitre, intitulé « **Conception du Système** » est consacré à la conception des données, ainsi que le traitement, le fonctionnement et l'environnement de développement de chacun des modules « MVC »

Le dernier chapitre intitulé « **Réalisation** » présente l'environnement de développement, l'architecture de déploiement ainsi que les interfaces réalisées.

Ce rapport sera clôturé par une **Conclusion** dans laquelle on présentera les acquis retenus au cours de ce projet ainsi que les perspectives à envisager en vue d'améliorer ses fonctionnalités.

CHAPITRE 2 : CONCEPTION DU SYSTEME

1- MODELISATION DU SYSTEME :

1.1 - DIAGRAMME DE CAS D'UTILISATION :

Afin de préciser le rôle de chaque acteur, et de décrire précisément leur interaction avec le système. On a utilisé un diagramme de cas d'utilisation.

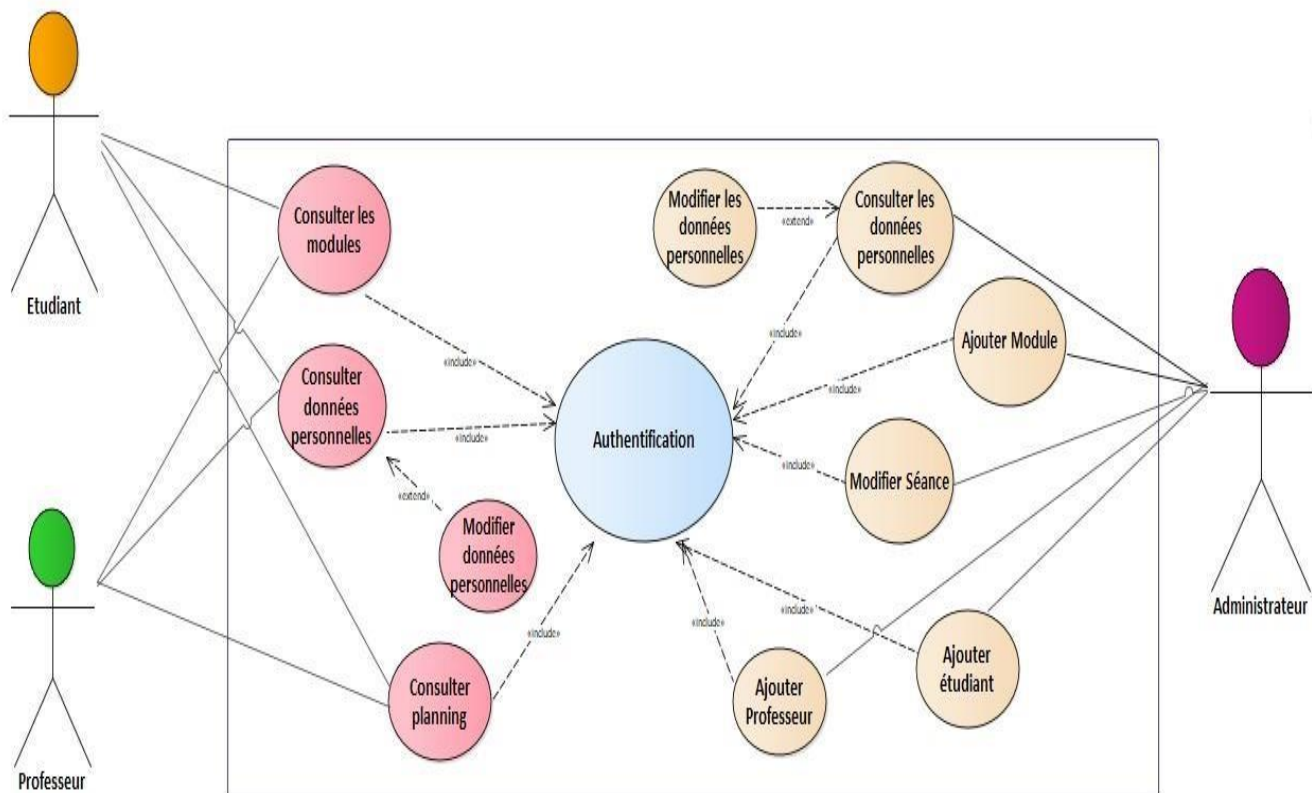


Figure 1: Diagramme de cas d'utilisation

Ce diagramme se compose de trois acteurs de notre système, chacun d'eux peut déclencher diverses fonctionnalités représentées par des cas d'utilisation.

1.2- DIAGRAMME DE CLASSE :

La construction d'un diagramme de classes nous a simplifier l'interaction des objets de notre système qu'on était en train de modéliser. Ce diagramme exprime la structure statique de notre système en termes de classes et les relations entre eux.

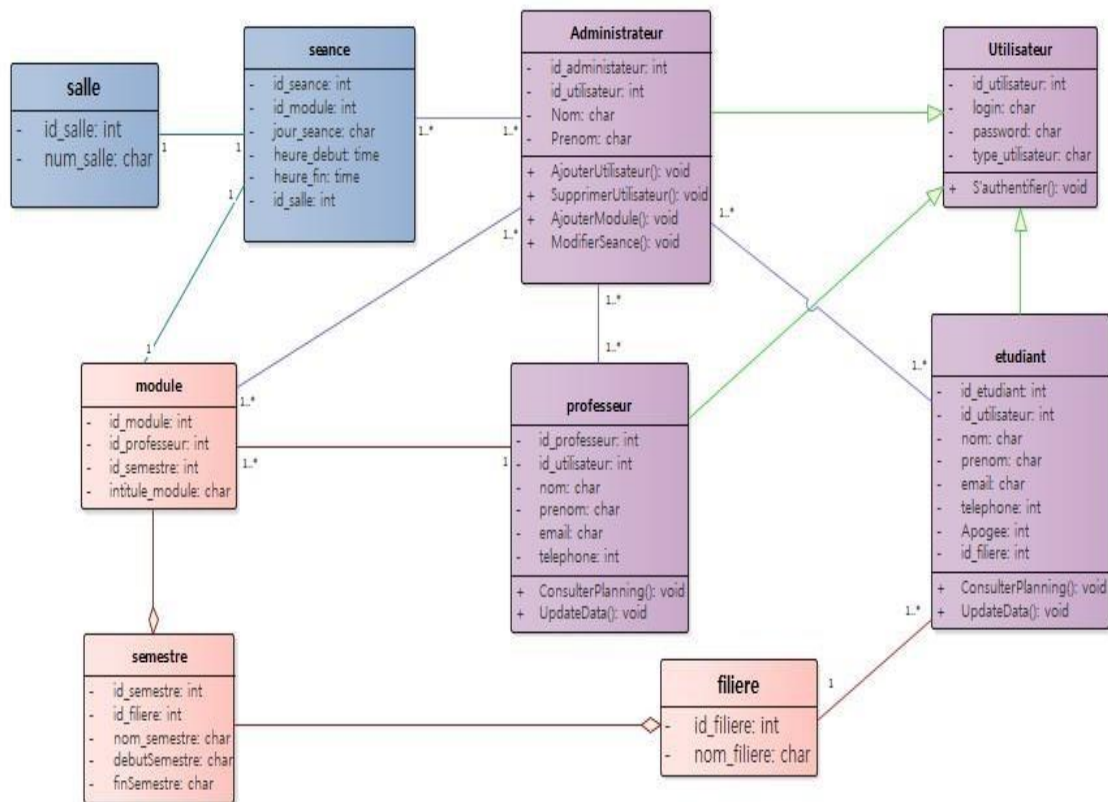


Figure 2: Diagramme de classe

Description du diagramme :

- Un étudiant est inscrit dans une seule filière.
- Une filière se compose de plusieurs semestres.
- Un semestre se compose de plusieurs modules.
- Un module est enseigné par un et un seul professeur, un professeur enseigne un ou plusieurs modules.

- Un module est enseigné dans une séance, et une séance est enseigné dans une salle.

1.3- DIAGRAMME DE SEQUENCE :

Afin de mieux décrire les interactions entre les modules du pattern « MVC », on a utilisé des diagrammes de séquences pour montrer les interactions d'objets dans le cadre d'un des scénarios du Diagramme de cas d'utilisation.

1.3.1-DIAGRAMME DE SEQUENCE (AUTHENTIFICATION) :

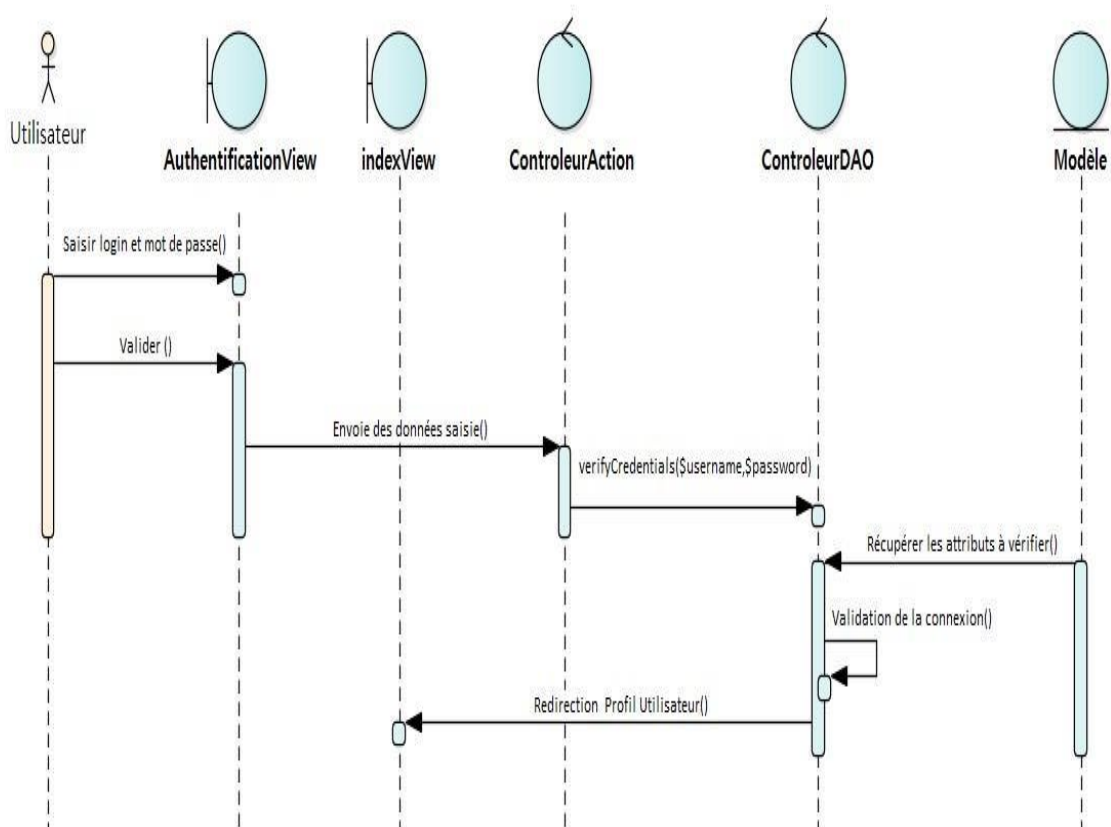


Figure 3: Diagramme de séquence (Authentification)

Ce diagramme représente les opérations nécessaires pour l'authentification de l'utilisateur.

1.3.2-DIAGRAMME DE SEQUENCE (UPDATE DATA) :

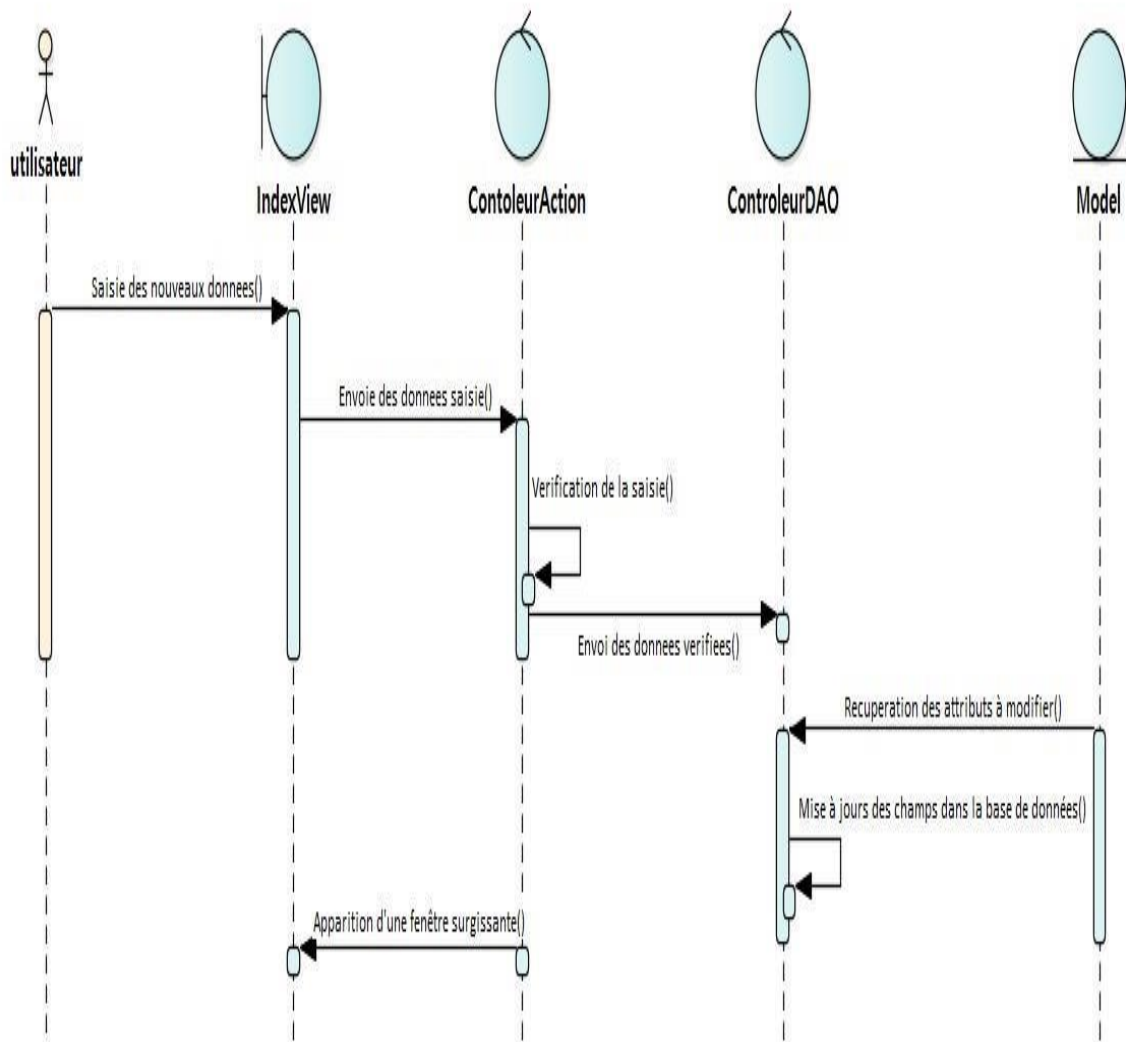


Figure 4: Diagramme de séquence (Update Data)

Après authentification, l'utilisateur accède à son espace personnel, dont lequel il peut visualiser et modifier ces données personnelles.

Ce diagramme représente l'enchaînement d'opération nécessaire pour la mise à jour des données de l'utilisateur.

2- NOYAU FONCTIONNEL DU SYSTEME :

Le noyau fonctionnel représente la pièce centrale d'un système d'information. C'est le mécanisme de traitement et de stockage de l'information, et la gestion des données internes de l'application. Concrètement, c'est la base de données du système.

Dans ce projet, on utilise le système de gestion de bases de données relationnelles MySQL, notre base de données se composent de 9 tables :

Table	Action
<input type="checkbox"/> administrateur	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> etudiant	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> filiere	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> module	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> professeur	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> salle	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> seance	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> semestre	★ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> utilisateur	★ Parcourir Structure Rechercher Insérer Vider Supprimer

Figure 5: Représentation de la base de données

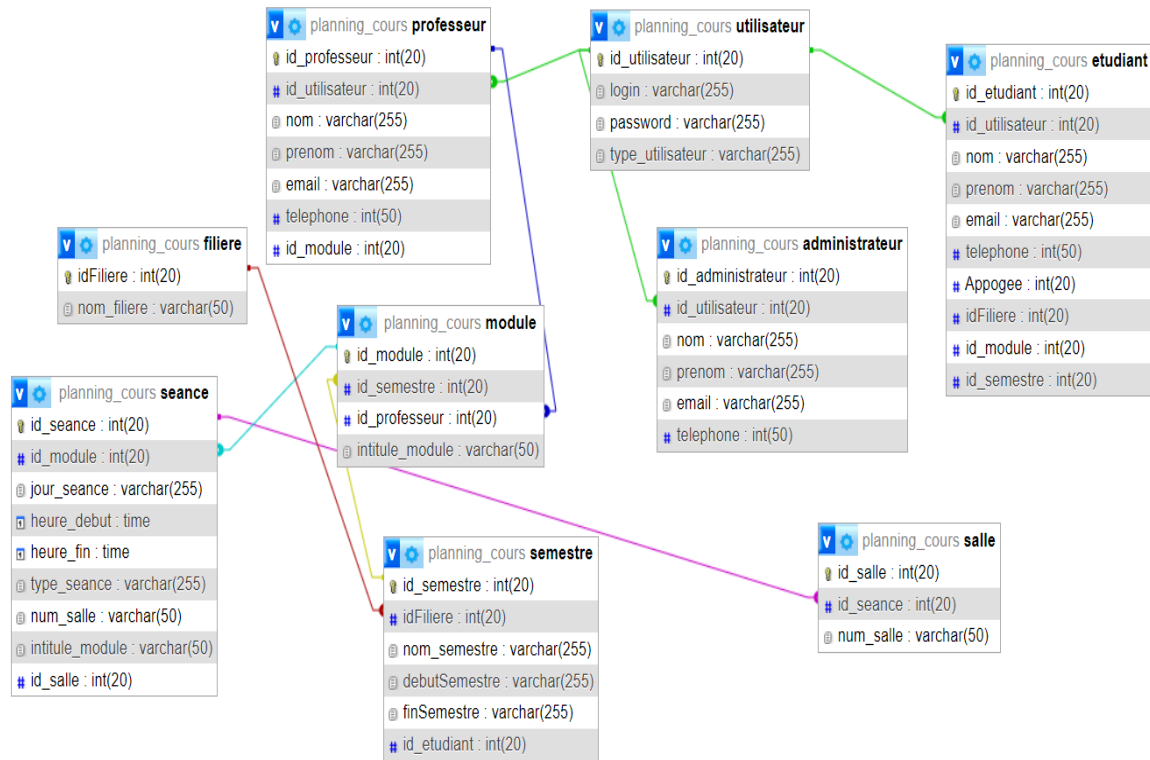


Figure 6: Concepteur des tables sur PhpMyAdmin

Les figures ci-dessous représentent les tables de notre base de données et les liaisons entre eux :

Table utilisateur :

« id_utilisateur » représente la clé primaire dans la table utilisateurs

Table étudiant :

« id_etudiant » représente la clé primaire dans la table etudiant

« id_utilisateur » représente une clé étrangère dans la table etudiant «

idFiliere » représente une clé étrangère dans la table étudiant

Table administrateur :

« id_administrateur » représente la clé primaire dans la table administrateur

« id_utilisateur » représente une clé étrangère dans la table administrateur

Table professeur :

« id_professeur » représente la clé primaire dans la table professeur

« id_utilisateur » représente une clé étrangère dans la table professeur

Table filière :

« idFiliere » représente la clé primaire dans la table filière

Table Semestre :

« id_semestre » représente la clé primaire dans la table semestre

« idFiliere » représente une clé étrangère dans la table semestre

Table module :

« id_module » représente la clé primaire dans la table module

« id_semestre » représente une clé étrangère dans la table module

« id_professeur » représente une clé étrangère dans la table module

Table Séance :

« id_seance » représente la clé primaire dans la table séance

« id_module » représente une clé étrangère dans la table séance

« id_salle » représente une clé étrangère dans la table séance

Table Salle :

« id_salle » représente la clé primaire dans la table salle

3- INTERFACE DU SYSTEME :

Afin de mettre en œuvre des interfaces graphiques pour notre système, nous avons décidé d'opter pour une architecture « MVC ».

3.1 UTILITE DU PATTERN « MVC » :

AU NIVEAU CONCEPTION :

« MVC » est un pattern très puissant qui intervient dans l'implémentation d'une application. Son principal intérêt est la séparation du modèle (classes), de la vue (affichage) et du contrôleur (actions), ce qui simplifie la clarté de l'architecture.

AU NIVEAU REALISATION :

- Il est **modulaire** : permet de découper le code de notre application en de nombreux fichiers, chaque fichier a un et un seul rôle à la fois.
- Il est **découplé** : chacun des fichiers de notre code est conçu pour fonctionner indépendamment l'un de l'autre.
- Il est **clair** : nos fichiers sont généralement assez courts et le code à l'intérieur peut se lire facilement.
- Il est **facile d'y travailler à plusieurs** : chacun des fichiers étant indépendant, on a pu facilement y travailler en binôme pour notre projet.
- Il est **évolutif** : il est facile d'ajouter une nouvelle fonctionnalité. Sans avoir peur de tout casser, et sans pour autant compliquer notre code.

Les différentes interactions entre modèle, la vue et le contrôleur sont résumées par le schéma dans la figure ci-dessous :

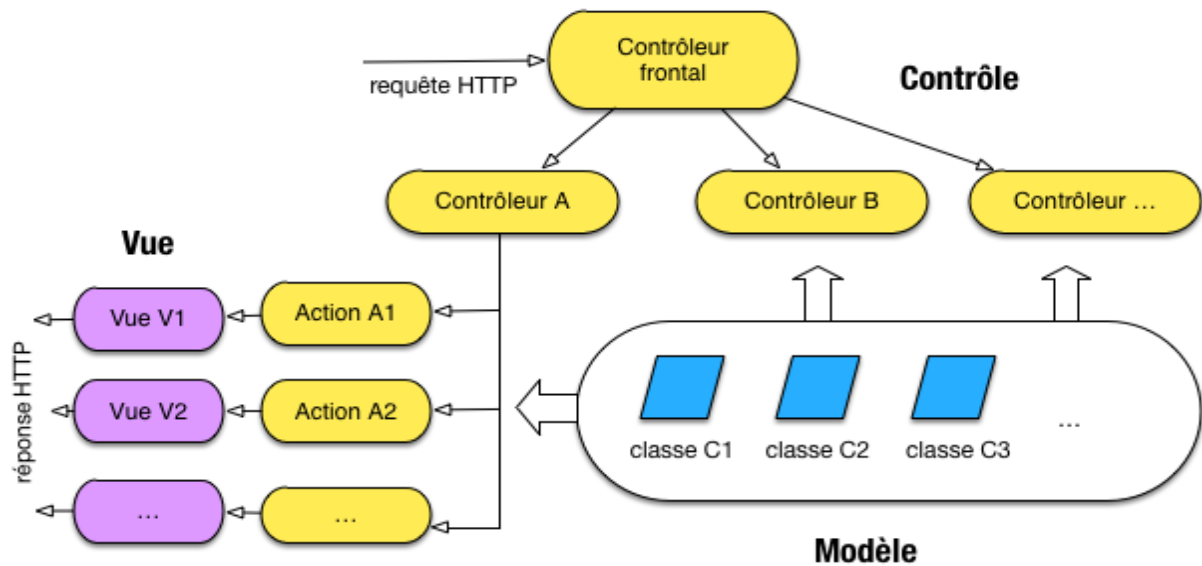
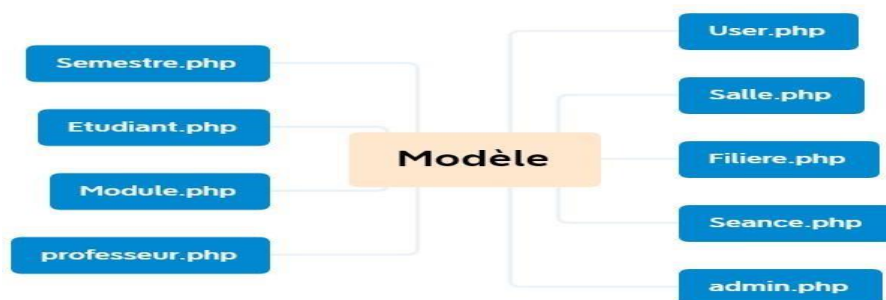


Figure 7: Schéma explicatif du pattern MVC

Comme le montre la figure, le module « modèle » se compose de plusieurs classes, chacune liée à un des fichiers du module « contrôleur », chaque contrôleur est lié à une ou plusieurs actions, qui interagissent avec le module « vue ».

3.2 PRESENTATION DU MODELE D'ABSTRACTION

Suivant la conception du diagramme de classe, et la base de données qui lui y est conforme. On a créé différents modèles miroirs sous forme de fichiers chacun contenant une classe dont les attributs représentent les champs de la table associée au modèle.



EXEMPLE DU CONTENUE D'UN FICHIER MODELE :

En ce qui concerne le contenu du fichier du modèle « Etudiant.php », Il n'est constitué que de **constructeur** permettant d'allouer la mémoire nécessaire à l'objet et d'initialiser ses attributs, d'accesseur(**getters**) permettant d'accéder à un attribut, et de mutateur(**setters**) permettant de modifier la valeur d'un attribut, c'est à dire toutes les méthodes primordiales qui sont nécessaire pour le bon fonctionnement d'un objet.

Dans aucun de ces modèles ne figure ni de requête **SQL**, ni de balise **HTML**, ou encore du **JavaScript**, il ne contient que du **PHP**.

Et c'est là que réside l'une des principales utilités du pattern MVC. Tout est bien répartie de façon à faciliter l'accès aux attributs des objets. Ainsi que la possibilité de les utiliser partout dans le code source.

3.3 PRESENTATION DU CONTROLEUR

Les seuls fichiers auxquels les modèles sont directement liés sont les fichiers **Contrôleurs**.

Le Contrôleur connaît les besoins de la Vue et du Modèle et sait quel moyen la Vue utilise pour modifier ou demander des informations au Modèle. Il transforme les demandes de la Vue en action, ou suite d'actions.

Dans ce projet, le module contrôleurs se divisent en 2 types de fichiers : fichiers DAO (Data Access Object) & fichiers actions.

LES FICHIERS « DAO » :

Les DAO se composent de différents fichiers chacun d'eux contenant diverses fonctions qui ont pour seul mission manager les différentes requêtes SQL nécessaires pour mener à bien les actions nécessaires par chacune des tables de notre base de données.

Autrement dit, chacune de nos tables à un fichier contenant un objet d'accès aux données.

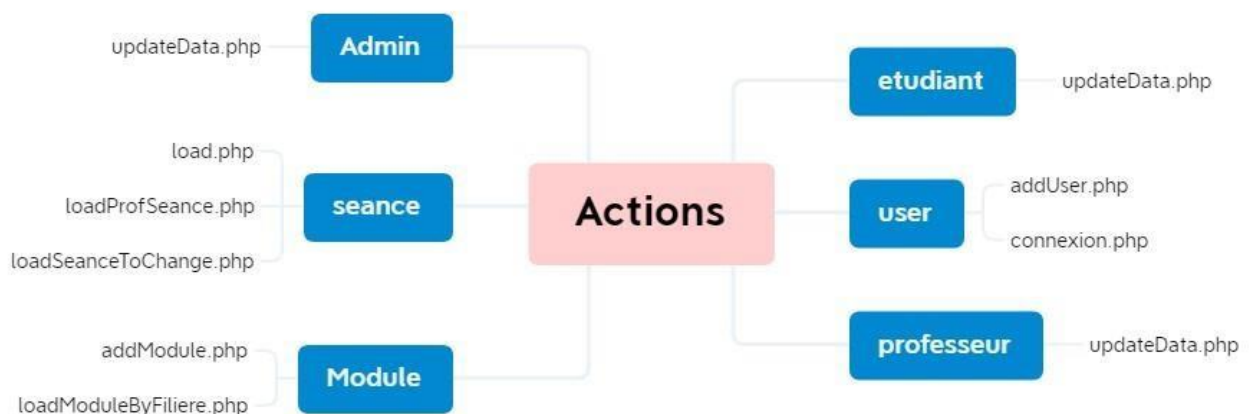


« EtudiantDAO.php » par exemple, il ne fait appel qu'au fichier modèle. Et la classe qui englobe tout le contenu du fichier, ne contient qu'un seul attribut « \$connexion », et des méthodes dont les tâches qui interagissent avec la base de données à l'aide de requêtes.

LES FICHIERS « ACTIONS » :

Les fichiers actions permettent de réaliser le traitement de différents scénarios en utilisant les méthodes des classes DAO.

Ces scénarios sont dans la majorité des cas, des services demander par l'utilisateur (authentification, mise à jour de données, ...). Contrairement à la répartition des DAO, les actions sont réparties en dossier relatif aux tables de la base de données, et au sein de ces dossiers chaque action a un fichier précis.



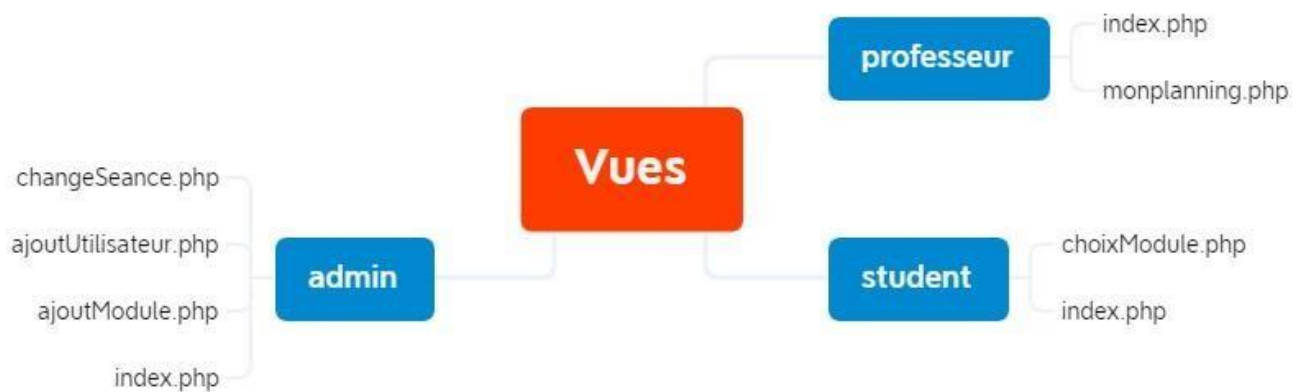
EXEMPLE DU CONTENUE D'UN FICHIER ACTIONS :

Le fichier « UpdateData.php » par exemple, est relative au dossier Étudiant. Il ne fait appel qu'au fichier DAO auquel il est associé. Il a pour unique but de récupérer les nouvelles coordonnées saisies par l'étudiant dans le formulaire et de les passer en argument à la fonction récupérer du fichier EtudiantDAO.

3.4 PRESENTATION DES VUES

Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant des fichiers DAO et Actions.

Notre module « vue » est réparti en 3 dossiers, chacun contenant les interfaces propres à chacun de nos 3 différents types d'utilisateurs.



2.4 CONCLUSION :

Dans ce chapitre, nous avons présenté quelques diagrammes définis par UML, qui ont permis de bien comprendre les besoins du système à développer ainsi que les différentes interactions entre les objets participant à son fonctionnement.




Ainsi que le pattern MVC, qui nous a été utile au cours de la conception de notre système, il nous a permis de bien organiser notre code source. Il nous a aidé non seulement à savoir quels fichiers créer, mais surtout à définir leur rôle.

CHAPITRE 3 : REALISATION

1- INTRODUCTION :

Dans ce chapitre on va parler des outils et des langages de développement, les étapes de la réalisation et nous allons montrer quelques captures d'écran pour avoir une idée sur l'application et son fonctionnement.

2- OUTILS DE DEVELOPPEMENT :

	<p>HTML (HyperText Markup Language) désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web.</p>
	<p>CSS (Cascading Style Sheets) ce langage informatique est utilisé sur l'internet pour mettre en forme les fichiers HTML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.</p>
	<p>JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS.</p>


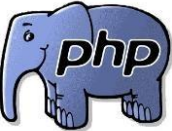



	<p>jQuery est un Framework JavaScript qui permet de faciliter des fonctionnalités de JavaScript. L'utilisation de cette bibliothèque permet de gagner du temps lors de l'interaction sur le code HTML d'une page web.</p>
	<p>HP (Hypertext Preprocessor) ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que MySQL. Exécuté du côté serveur. Il n'y a pas besoin aux visiteurs d'avoir des logiciels ou plugins particulier.</p>
	<p>MySQL est une base de données relationnelle libre employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation. Le principe d'une base de données relationnelle est d'enregistrer les informations dans des tables, qui représentent des regroupements de données par sujets. Les tables sont reliées entre elles par des relations.</p>
	<p>Un front-end framework léger et modulaire pour développer des interfaces Web rapides et puissantes. Un framework est conçu en vue d'aider les programmeurs dans leur travail. L'organisation du framework vise la productivité maximale du programmeur qui va l'utiliser.</p>
	<p>Ajax est une méthode utilisant différentes technologies, et dont la particularité est de permettre d'effectuer des requêtes au serveur web et, en conséquence, de modifier partiellement la page web affichée sur le poste client sans avoir à afficher une nouvelle page complète.</p>

Figure 8: Table représentative des outils et langages utilisés

3-PRESENTATION DE L'APPLICATION

Notre plateforme se constitue de trois différentes interfaces selon nos 3 types d'utilisateur (étudiant, professeur, administrateur).

Ainsi qu'une page d'accueil commune qui se constitue d'un formulaire de connexion, d'une présentation de la plateforme, et d'un formulaire contact.

3.1 PAGE D'ACCUEIL

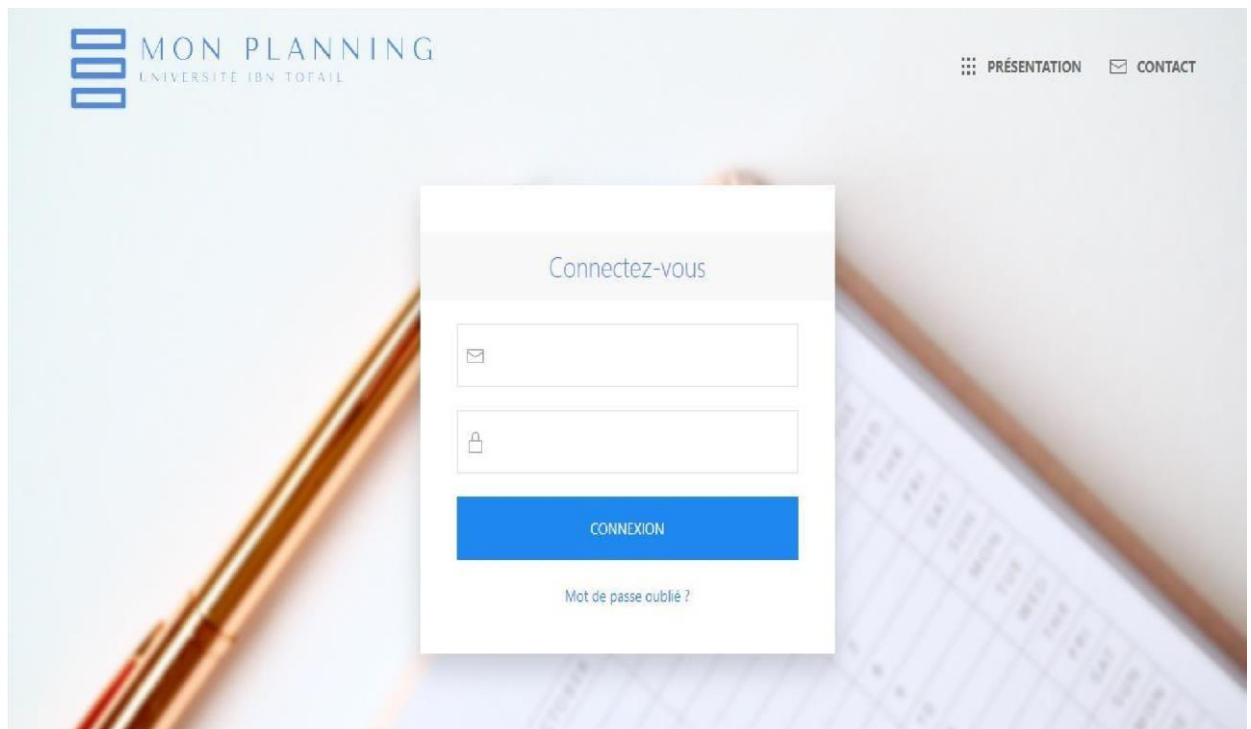


Figure 9: Page d'accueil

On a choisi la page de connexion comme page d'accueil pour faciliter l'accès aux utilisateurs, notre formulaire de connexion se compose du login et du mot de passe :

si ces deux sont acceptés le visiteur passe à son espace selon son type d'utilisateur.

3.2 PRESENTATION DE LA PLATEFORME

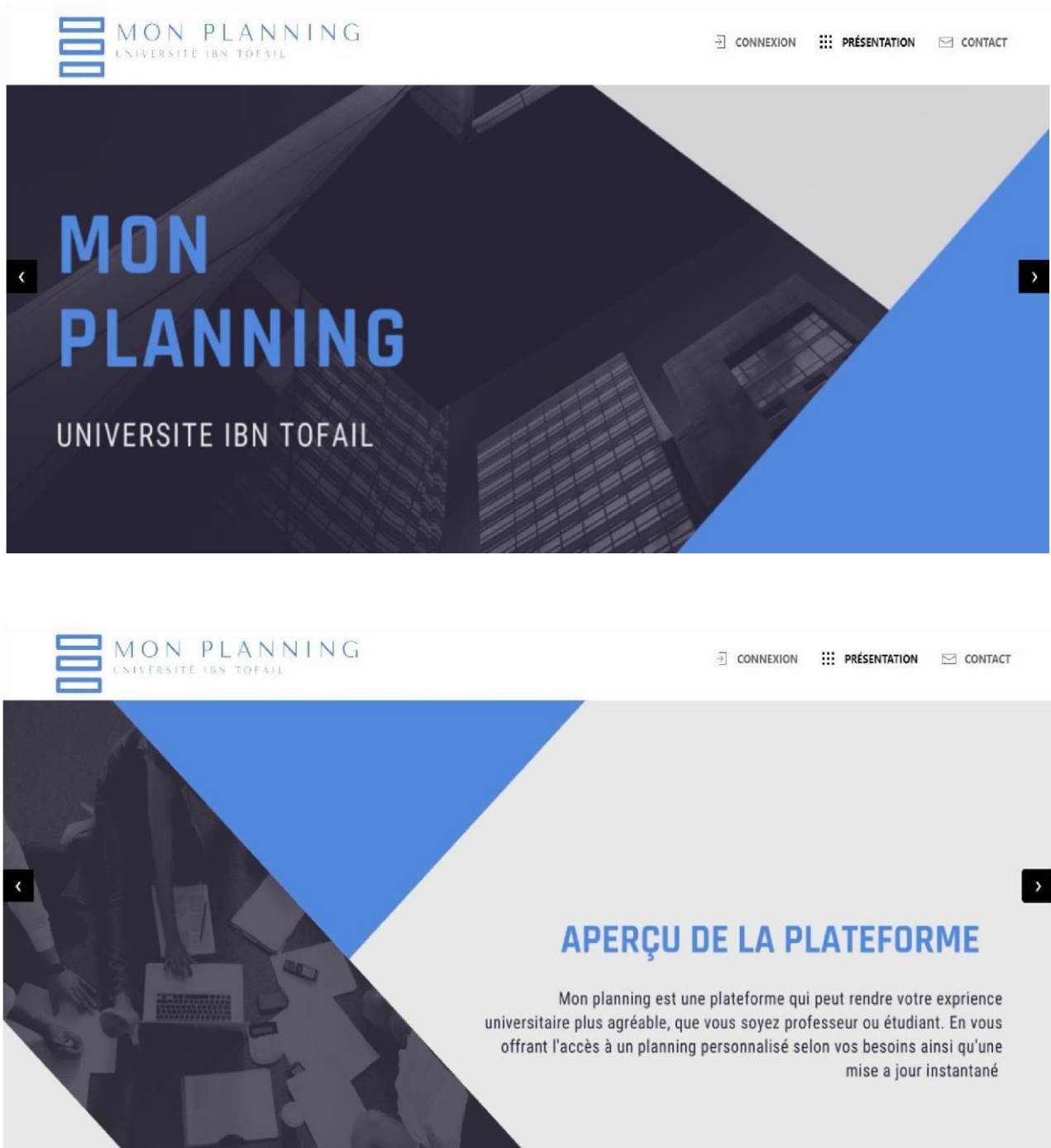


Figure 10: Page Présentation

Cette présentation donne au visiteur une idée générale sur le but de l'application, son utilité, ainsi que nos 3 types d'utilisateurs auxquels notre visiteur peut s'identifier.

3.3 FORMULAIRE DE CONTACT :

Figure 11: Formulaire de contact

Indispensable à chaque site Internet, le formulaire de contact est un moyen simple de créer un lien avec les utilisateurs, avec très peu de contraintes.

Une fois remplies, les informations saisies sont directement envoyées au mail de l'administrateur. Ce qui permet un traitement rapide des messages envoyés par nos utilisateurs.

3.4 FOOTER :

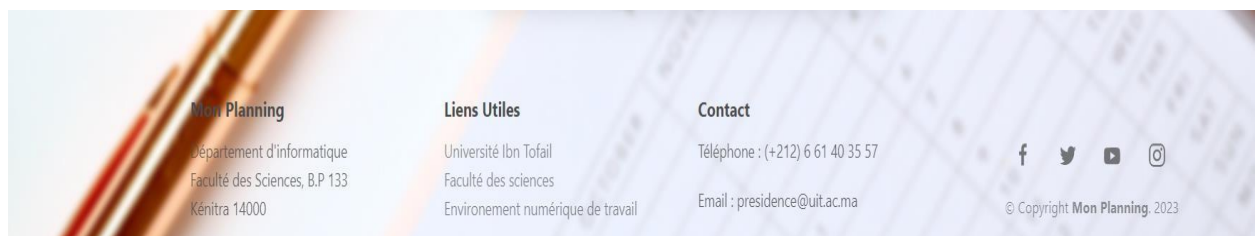
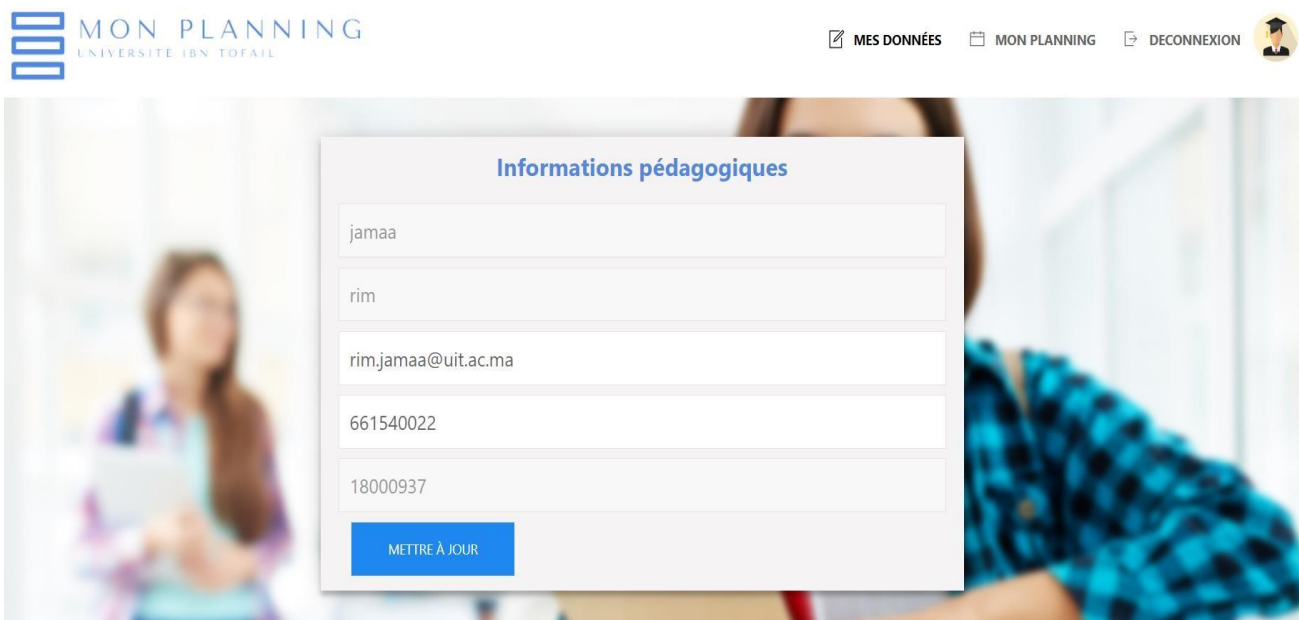


Figure 12: Footer

Le « Footer » désigne le bas d'une page, il s'agit d'un élément qui doit être étudié attentivement car le bas d'une page doit pouvoir permettre au visiteur de retrouver toutes les informations principales relative à l'utilité de notre plateforme.

3.5 PREMIER PROFIL : PROFIL ETUDIANT

PAGE D'ACCUEIL « ETUDIANT » :



The screenshot shows the 'MON PLANNING' interface for a student. The header includes the logo 'MON PLANNING UNIVERSITE IBN TOFAIL' and navigation links: 'MES DONNÉES', 'MON PLANNING', 'DECONNEXION', and a user profile icon. A modal window titled 'Informations pédagogiques' is open, containing a form with the following fields: 'jamaa', 'rim', 'rim.jamaa@uit.ac.ma', '661540022', and '18000937'. A blue button labeled 'METTRE À JOUR' is at the bottom of the form.

Figure 13: Page d'accueil "ETUDIANT"

Notre principal utilisateur est l'étudiant, après l'authentification son interface lui permet de revoir ces informations personnelles, ainsi que de modifier des champs particulier (l'email, et le numéro de téléphone) et les mettre à jour.

Dans le cas où l'étudiant saisit une modification. Un « pop-up » apparaît pour assurer l'utilisateur de la réussite de sa saisie.



Figure 14: "Pop up"

PAGE PLANNING « ETUDIANT »

Sciences mathématiques et informatiques

Modules :

- ☐ Analyse1
- ☐ Mécanique du point
- ☐ LT1
- ☐ Thermodynamique
- ☐ Algèbre1
- ☐ Algèbre2
- ☐ Introduction à l'informatique
- ☐ LT2
- ☐ Optique
- ☐ Électricité
- ☐ Analyse2
- ☐ Analyse3
- ☐ Algèbre3
- ☐ Algorithmique1
- ☐ Electronique
- ☐ Langage C
- ☐ Algorithmique2
- ☐ Probabilité et statistiques

- ☐ Technologie du web
- ☐ Système d'exploitation1
- ☐ Analyse numérique
- ☐ Electromagnétisme
- ☐ Structure de données
- ☐ Architecture des ordinateurs
- ☐ Langage C suite
- ☐ UML
- ☐ Compilation
- ☐ C++
- ☐ Réseaux
- ☐ Recherche opérationnelle
- ☐ Base de données
- ☐ Base de données 2
- ☐ Java
- ☐ Réseaux2
- ☐ Programmation système

AFFICHER PLANNING

SUPPRIMER LA SELECTION

<
>
today

Jun 19 – 25, 2023

month
week
day

	Mon 6/19	Tue 6/20	Wed 6/21	Thu 6/22	Fri 6/23	Sat 6/24	Sun 6/25
all-day							
6am							
7am							
8am							
9am					9:00 - 10:30 Java (M4)		
10am							
11am			10:45 - 12:15 Programmation système (Amphi5)	10:45 - 12:15 Java (3)			
12pm							
1pm							
2pm							
3pm							
4pm							

Figure 15: Page planning ETUDIANT

Dans cette interface, l'étudiant peut consulter son planning selon la filière dans laquelle il est inscrit. En lui donnant la main de choisir les modules qu'il souhaite afficher.

Selon les modules sélectionnés, un planning va s'afficher précisant les horaires des séances et le locale, offrant plusieurs format (mois, semaine, jour).

3.6 DEUXIEME PROFIL : PROFIL PROFESSEUR

PAGE D'ACCUEIL « PROFESSEUR » :

The screenshot shows the 'MON PLANNING' website interface for a professor. At the top, there is a navigation bar with the logo 'MON PLANNING UNIVERSITE IBN TOFAIL' on the left and three links: 'MES DONNÉES', 'MON PLANNING', and 'DECONNEXION' on the right, followed by a user profile icon. Below the navigation bar, a modal form titled 'Informations personnelles' is displayed. The form contains four input fields: 'khalid', 'benlhachmi', 'khalid.benlhachmi@uit.ac.ma', and '610935956'. A blue button labeled 'METTRE À JOUR' is at the bottom of the form. The background of the page is a blurred image of a smiling woman.

Figure 16: Page d'accueil PROFESSEUR

Tout comme le profil étudiant, le professeur peut consulter ces données personnelles sur sa page d'accueil, ainsi qu'une modification des champs précis.

PAGE PLANNING « PROFESSEUR »

The screenshot shows the 'PAGE PLANNING' for a professor. At the top, there is a navigation bar with the logo 'MON PLANNING UNIVERSITE IBN TOFAIL' on the left and three links: 'MES DONNÉES', 'MON PLANNING', and 'DECONNEXION' on the right, followed by a user profile icon. Below the navigation bar, a calendar interface is displayed. The calendar shows the dates 'Jun 19 – 25, 2023'. The calendar has a header with 'Mon 6/19', 'Tue 6/20', 'Wed 6/21', 'Thu 6/22', 'Fri 6/23', 'Sat 6/24', and 'Sun 6/25'. The calendar grid shows the following schedule:

	Mon 6/19	Tue 6/20	Wed 6/21	Thu 6/22	Fri 6/23	Sat 6/24	Sun 6/25
all-day							
6am							
7am							
8am							
9am					9:00 - 10:30 Java (M4)		
10am							
11am				10:45 - 12:15 Java (3)			
12pm							
1pm							
2pm							
3pm							
4pm							

Figure 17: Page planning PROFESSEUR

Dans cette interface, le professeur peut consulter le planning des modules qu'il enseigne selon la filière qu'il souhaite afficher.

Selon la ou les filières sélectionnées, un planning va s'afficher précisant les horaires des séances et le locale des modules enseignait par ce professeur, offrant plusieurs format (mois, semaine, jour).

3.7 TROISIEME PROFILE : PROFIL ADMINISTRATEUR

Notre plateforme offre une interface administrateur, qu'il lui permet d'effectuer diverses opérations : Ajouter Utilisateur (étudiant ou professeur), Ajouter Module, et Modification des séances.

PAGE D'ACCUEIL « ADMINISTRATEUR » :

The screenshot displays the administrator's dashboard. At the top, the navigation bar includes the 'MON PLANNING' logo and links for 'MES DONNÉES', 'UTILISATEUR', 'MODULE', 'PLANNING', 'DECONNEXION', and a user profile icon. The main content area features a form titled 'Informations administrateur' with the following fields:

- Username: ata
- Password: abdellah
- Email: abdellah.ata@uit.ac.ma
- Phone number: 688765433

A blue button labeled 'METTRE À JOUR' is located at the bottom of the form.

Figure 18: Page d'accueil ADMIN

Tout comme les autres utilisateurs, l'administrateur bénéficie aussi d'une interface lui permettant de consulter ces données personnelles et l'es modifiés s'il le souhaite.

MODIFICATION DES SEANCES :

[MES DONNÉES](#)[UTILISATEUR](#)[MODULE](#)[SEANCE](#)[DECONNEXION](#)

Critère de recherche :

--- Toutes les filières ---

Tous les modules

GÉNÉRER PLANNING

VIDER LA SÉLECTION

SEANCE

JOUR

HEURE DÉBUT

HEURE FIN

SALLE

MODIFIER

Critère de recherche :

Sciences mathématiques et informatique

--- Tous les modules ---

GÉNÉRER PLANNING

VIDER LA SÉLECTION

SEANCE

JOUR

HEURE DÉBUT

HEURE FIN

SALLE

MODIFIER

Mécanique du point

Mardi

11:00:00

12:00:00

C1



Réseaux2

Mercredi

09:00:00

10:30:00

C1



Java

Jeudi

10:45:00

12:15:00

Amphi5



Programmation système

Mercredi

10:45:00

12:15:00

Amphi5



Java

Vendredi

09:00:00

10:30:00

M4



Thermodynamique

Mardi

11:00:00

12:00:00

Amphi2





The screenshot shows a web form titled "Modifier la Seance : Réseaux2" with a close button (X) in the top right corner. The form contains four input fields: "Jour" with the value "Mercredi", "Heure Debut" with the value "09:00:00", "Heure Fin" with the value "10:30:00", and "Salle" with the value "2". At the bottom right, there are two buttons: "ANNULER" (light gray) and "ENREGISTRER" (blue).

Figure 19: Modification des séances

L'interface offre à l'administrateur la possibilité de modifier les séances du planning.

Si l'administrateur sélectionne le choix « toutes les filières » « tous les modules ».

Une liste comportant toutes les séances existantes dans la base de données s'affiche, et il pourrait choisir laquelle il souhaite modifier.

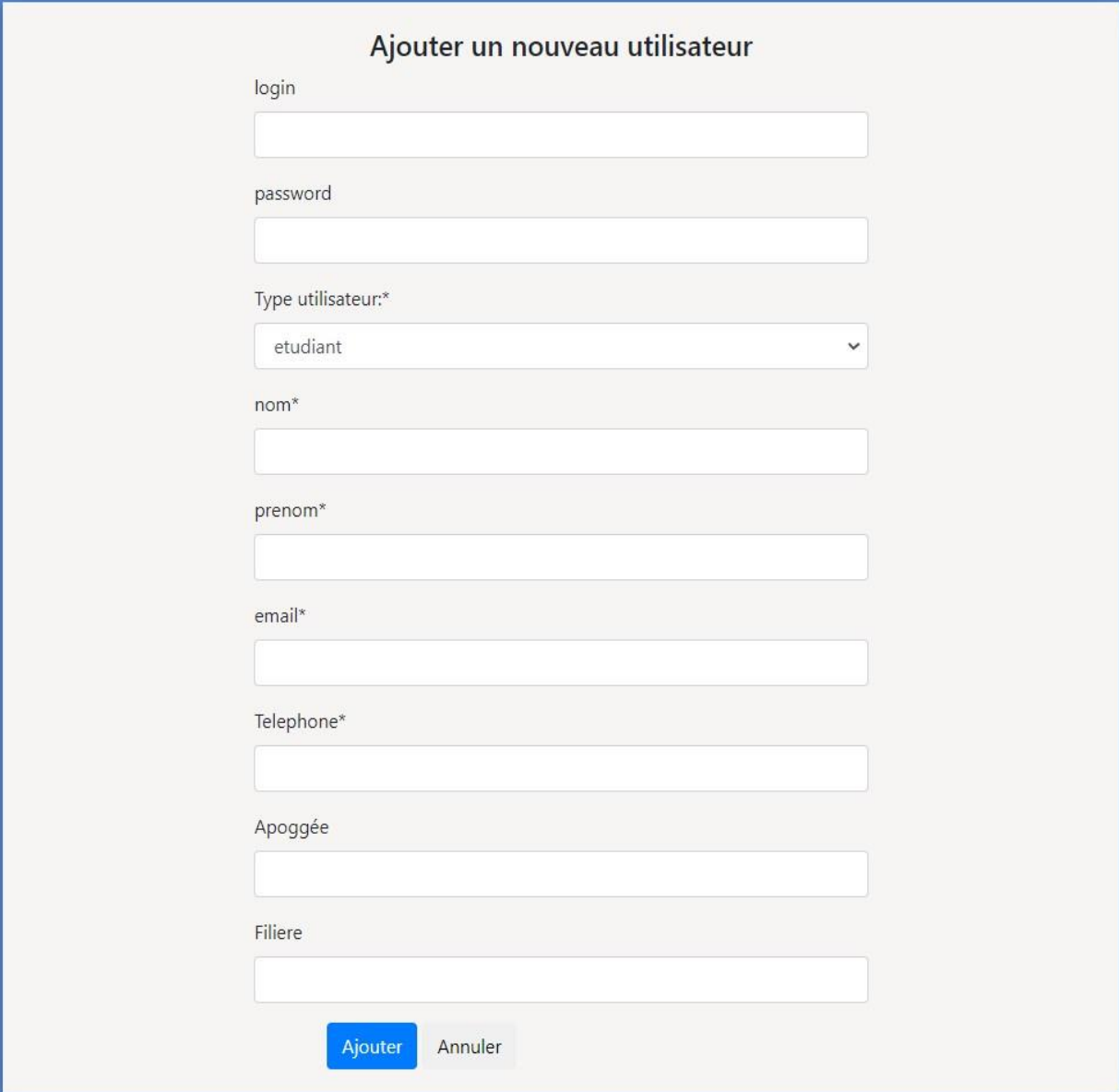
Si l'administrateur précise la filière qu'il veut, la liste des modules qui va apparaître ne contient que les modules de la filière sélectionnée.

Après que l'administrateur choisit un des modules à afficher. Une liste comportant les séances relatives à ce module va apparaître.

Ensuite, il suffit de cliquer sur l'icône modifier, pour qu'un formulaire de modification apparaisse.

Le formulaire permet de remplir les informations suivantes : le jour de la séance, l'heure de début, l'heure de fin, et la salle.

AJOUTER UTILISATEUR :

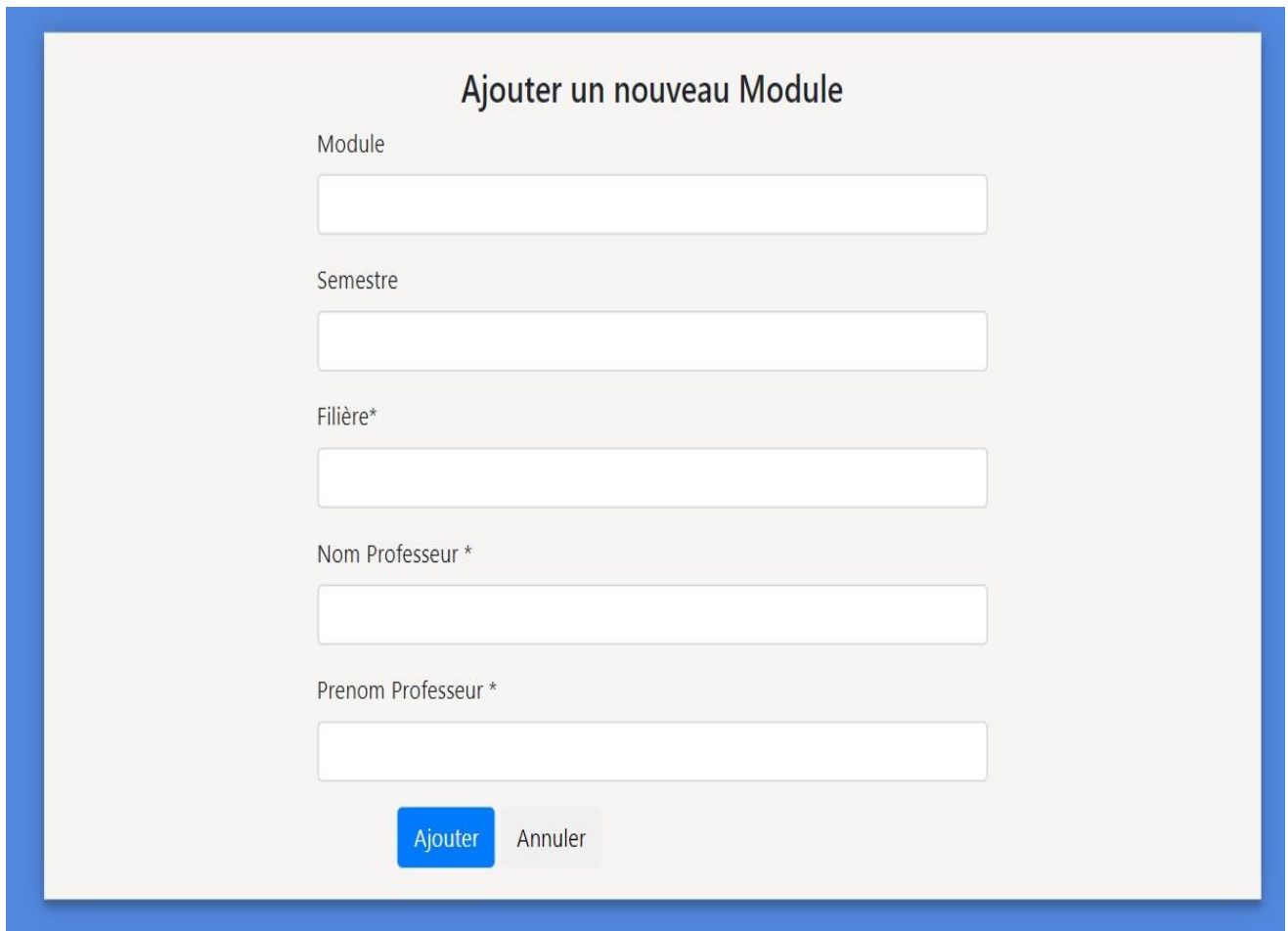


The screenshot shows a web form titled "Ajouter un nouveau utilisateur" (Add a new user). The form is set against a light gray background with a blue border. It contains several input fields and a dropdown menu, all with light gray borders. The fields are labeled as follows: "login", "password", "Type utilisateur:*" (with a dropdown menu showing "etudiant"), "nom*", "prenom*", "email*", "Telephone*", "Apoggée", and "Filiere". At the bottom of the form, there are two buttons: a blue "Ajouter" button and a gray "Annuler" button.

Figure 20: Ajouter utilisateur

Cette interface permet à l'administrateur d'ajouter un utilisateur via un formulaire, tout en précisant son type (étudiant ou professeur). Après avoir ajouter un utilisateur un pop s'affiche pour mentionner la réussite de l'opération.

AJOUTER MODULE :

The image shows a web form titled "Ajouter un nouveau Module" (Add a new Module). The form is contained within a light gray box with a blue border. It has five text input fields: "Module", "Semestre", "Filière*" (with an asterisk), "Nom Professeur *" (with an asterisk), and "Prenom Professeur *" (with an asterisk). At the bottom of the form, there are two buttons: a blue button labeled "Ajouter" and a light gray button labeled "Annuler".

Ajouter un nouveau Module

Module

Semestre

Filière*

Nom Professeur *

Prenom Professeur *

Ajouter Annuler

Figure 21: Ajouter module

Cette interface permet à l'administrateur d'ajouter un module via un formulaire, tout en précisant la filière et le semestre auquel il appartient.

4- CONCLUSION

La dernière partie était dédiée à la navigation dans notre application. Elle constitue le dernier volet de ce rapport, elle a pour objet de présenter l'environnement logiciel et matériel de la réalisation, et de décrire les principales fonctionnalités implémentées suite au développement par des captures écrans présentant les interfaces de l'application.

CONCLUSION ET PERSPECTIVES

L'objectif principal de ce Rapport était de représenter la conception et le développement une application Web profilée pour affichage et modification des emplois du temps universitaire.

En effet, cette application permet de :

- Faciliter l'accès à un planning selon le type de profil utiliser, et les données propres à chacun :
 - Pour l'étudiant : un planning suivant les modules qu'il a choisi.
 - Pour le professeur : un planning suivant la ou les filières qu'il a choisies.
- La possibilité de modifier les séances par l'administrateur.

Cette application peut être améliorée par divers fonctionnalités :

- En ajoutant un système de notification instantané, qui permet d'informer les étudiants ainsi que les professeurs des dernières modifications du planning.
- En permettant au professeur de rectifier son planning, et d'informer ses étudiants.
- Il est aussi possible de créer un système qui analyse les données des emplois et propose les salles et les horaires libres lors d'une modification.

L'élaboration de notre application web nous a permis de nous confronter à une expérience professionnelle dans le domaine de développement des applications Web que nous ne pouvons qualifier que d'enrichissante. En outre, elle nous a offert l'opportunité de concrétiser nos connaissances théoriques grâce à un cas réel, approfondir nos connaissances dans plusieurs langages de programmation qu'on a découvert et pratiqué, ainsi que savoir utiliser un Framework et un Plugin, et comment savoir lire et utiliser une documentation.

BIBLIOGRAPHIE

COURS : (FILIERE SMI - FACULTE DES SCIENCES DE KENITRA) :

- [1] Pr. Abdelalim Sadiq. « Cours Technologie du Web » SMI-S3.
- [2] Pr. Jalal Laassiri. « Cours C.O.O (UML) » SMI-S5.
- [3] Pr. Akram.Azami « Cours Bases de données » SMI-S5.

AUTRE COURS :

- [4] Apprendre le PHP-2019, GrafikArt.
- [5] Programmez en orienté objet en PHP, OpenClassroom. « Victor Thuillier ».
- [6] Adoptez une architecture MVC en PHP. « Mathieu Nebra ».

LIVRES :

- [7] Mathieu Nebra. « Concevez votre site web avec PHP et MySQL ». Édition Simple IT, 2013.
- [8] Pascal Roques. « UML 2 par la pratique : Etudes de cas et exercices corrigés ». 5^e édition, 2006.
- [9] Jean Engels. « PHP 5 : Cours et Exercices ». 2^e édition, 2009.

SITES INTERNET :

[10] <https://www.php.net/>

[11] <https://openclassrooms.com/>

[12] <https://www.w3schools.com/>

[13] <https://stackoverflow.com/>

[14] <https://github.com/>

[15] <https://developer.mozilla.org/>

