



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université des Sciences et de la Technologie Houari Boumediene**

**Faculté d'Informatique  
Département d'Intelligence Artificielle  
et Sciences des Données**

Mémoire de Projet pluridisciplinaire

**Filière : Informatique**

**Spécialité : Informatique Ingénieur 2ème Année**

---

## **Thème**

**Conception et réalisation d'une application informatique pour  
la génération automatique des emplois du temps de la faculté informatique.**

---

**Sujet Proposé par :**

**Dr. DJABALLAH Kamel Ahsene**

**Devant le jury composé de :**

Dr. DJABALLAH Kamel Ahsene

Dr. Mekhaldi Fouzi

**Présenté par :**

LAOUDI Farid

GHAZI Oussama Sohieb

MEZGHENNA Mohanned

MAHMOUD Rayan

HESSAS Nassim

KHELIFI Souhil Idris

Groupe n° : 01 / 2024

# Table des matières

<b>Table des figures</b>	<b>i</b>
<b>Introduction Générale</b>	<b>1</b>
<b>1 Etude de l'existant</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Etude préalable . . . . .	2
1.2.1 Présentation de l'USTHB . . . . .	2
1.2.2 Organigramme de l'USTHB . . . . .	3
1.2.3 Présentation de la Faculté informatique à l'USTHB . . . . .	4
1.2.4 Organigramme de la faculté informatique . . . . .	4
1.3 Cadre d'étude . . . . .	5
1.3.1 Vice-rectorat chargé de la scolarité et de la pédagogie (VRSP) . . .	5
1.3.2 Centre des Ressources Informatiques (CRI) . . . . .	5
1.3.3 Les deux départements principaux de la Faculté d'Informatique . .	5
1.3.4 le Vice-doyenne chargée des études et des questions liées aux étudiants	5
1.3.5 Processus de planification des emplois du temps . . . . .	6
1.4 Les Ressources et les Contraintes . . . . .	7
1.4.1 Les Ressources Humaines . . . . .	7
1.4.2 Les Ressources Matérielles . . . . .	7
1.4.3 Les Ressources Technologiques . . . . .	7
1.4.4 Les Contraintes Temporelles . . . . .	7
1.4.5 Les Contraintes Logistiques . . . . .	7
1.4.6 Les Contraintes Pédagogique . . . . .	7
<b>2 Conception</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Analyse des besoins . . . . .	8
2.3 Choix technologiques . . . . .	8

2.4	Architecture du système . . . . .	10
2.4.1	Diagramme de l'architecture . . . . .	10
2.5	Conception des modules . . . . .	10
<b>3</b>	<b>Implementation</b>	<b>21</b>
3.1	Interface d'utilisateur . . . . .	21
3.1.1	Page de Connexion . . . . .	22
3.1.2	Tableau de Bord . . . . .	23
3.1.3	Page des Enseignants . . . . .	24
3.1.4	Formulaire d'Ajout d'un Enseignant . . . . .	25
3.1.5	Page des Salles . . . . .	26
3.1.6	Page de Génération des Emplois du Temps . . . . .	28
3.1.7	Page des Emplois du Temps . . . . .	29

# Table des figures

1.1	Organigramme de l'USTHB . . . . .	3
1.2	Organigramme de la faculté informatique . . . . .	4
1.3	Diagramme de flux de la processus de planification . . . . .	6
2.1	architecture de programme . . . . .	10
2.2	Diagramme d'activity . . . . .	10
2.3	Diagramme de code . . . . .	11
3.1	User flow . . . . .	21
3.2	Page de Connexion . . . . .	22
3.3	Page des Enseignants . . . . .	23
3.4	Page des Enseignants . . . . .	24
3.5	Formulaire d'Ajout d'un Enseignant . . . . .	25
3.6	Page des Salles . . . . .	26
3.7	Ajour des salles . . . . .	27
3.8	Page de Génération des Emplois du Temps . . . . .	28
3.9	Page des Emplois du Temps . . . . .	29

# Introduction Générale

La planification des emplois du temps est une tâche complexe et cruciale dans toute institution éducative, en particulier dans les facultés informatiques où les programmes d'études sont vastes et diversifiés. L'organisation des horaires pour les enseignants, les étudiants, les salles, et les ressources pédagogiques doit tenir compte de nombreuses contraintes, telles que la disponibilité des enseignants, les créneaux horaires, la capacité et la disponibilité des salles, et les programmes d'études des différentes filières.

Dans le contexte de la Faculté Informatique de l'USTHB, la gestion manuelle des emplois du temps peut entraîner des conflits de planification, une coordination inefficace, et des retards dans la communication des horaires aux étudiants et au personnel.

Ces problèmes peuvent avoir des répercussions sur la qualité de l'enseignement et la satisfaction des étudiants.

L'objectif de ce projet est de concevoir et de réaliser une application informatique qui permet de générer automatiquement des emplois du temps sans conflit, d'assurer une consultation rapide des horaires, et de garantir une meilleure coordination du travail entre les différents acteurs impliqués dans la planification. Pour y parvenir, il est nécessaire de réaliser une analyse détaillée du système pédagogique existant, afin de comprendre les processus actuels et d'identifier les opportunités d'amélioration.

Cette introduction donne un aperçu des problèmes liés à la planification des emplois du temps et souligne l'importance de concevoir un système informatique capable de répondre à ces défis. L'étape suivante consiste à effectuer une étude approfondie du système pédagogique de la faculté pour guider la conception de l'application proposée.

# Chapitre 1

## Etude de l'existant

### 1.1 Introduction

Avant de commencer la conception d'un système d'information, il est indispensable de réaliser une étude approfondie de l'environnement existant. Cette démarche vise à identifier les dysfonctionnements et à déterminer les causes des problèmes rencontrés, permettant ainsi de formuler des solutions adaptées. L'analyse du système existant doit fournir toutes les informations nécessaires pour garantir une conception solide, ainsi que des suggestions pour les améliorations à apporter. C'est une phase essentielle dans le processus de développement d'un système d'information.

Dans ce chapitre, nous allons présenter une étude globale du système pédagogique universitaire concerné par notre étude, qui est celui de la faculté d'informatique de l'USTHB. Cette étude permettra de mettre en lumière les enjeux majeurs, les processus opérationnels, ainsi que les contraintes à respecter. Elle servira de point de départ pour déterminer les besoins fonctionnels du nouveau système d'information, tout en identifiant les pistes d'amélioration pour optimiser l'efficacité du système existant.

### 1.2 Etude préalable

#### 1.2.1 Présentation de l'USTHB

L'Université des Sciences et de la Technologie Houari Boumediene est une université publique algérienne qui a été officiellement inaugurée par le décret n 74-50 du 25 avril 1974, et a été nommée Université des Sciences et de la Technologie Houari Boumediene en 1980 en hommage au président défunt Houari Boumediene. Elle est un pôle important de formation et de recherche dans les domaines des sciences et de la technologie en raison de

ses laboratoires équipés et de ses centres de recherche dans ses différentes facultés. Elle a également été conçue pour être une ville miniature offrant toutes les commodités dont un étudiant pourrait avoir besoin dans sa vie universitaire, garantissant ainsi une formation et une expérience de vie étudiante optimales, avec diverses installations à disposition.

### 1.2.2 Organigramme de l'USTHB

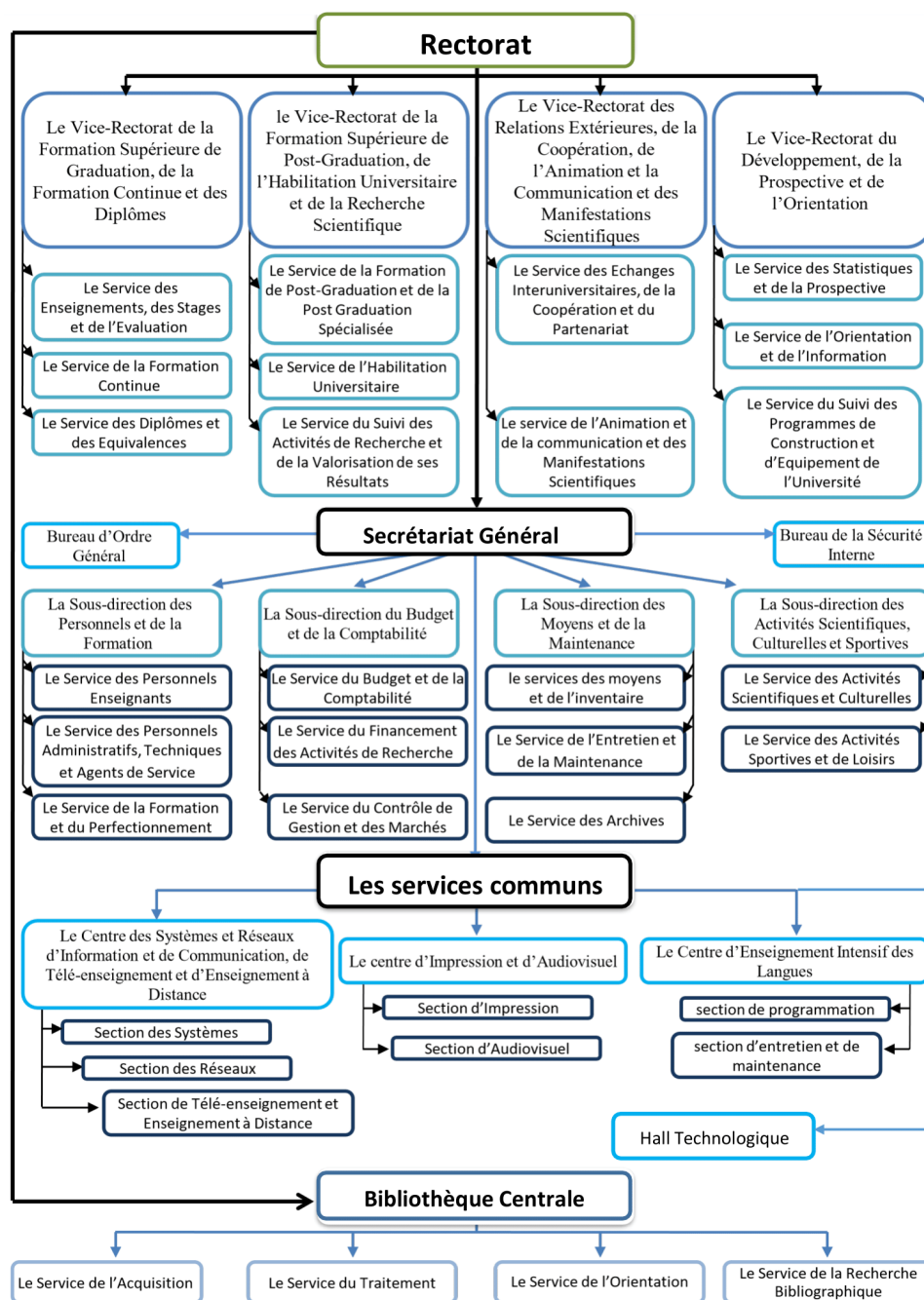


Figure 1.1: Organigramme de l'USTHB

### 1.2.3 Présentation de la Faculté informatique à l'USTHB

La Faculté d'Informatique de l'USTHB a été créée par décret exécutif au 25 novembre 2021. Elle est composée de deux départements principaux : le Département d'Intelligence Artificielle et Sciences des Données, et le Département des Systèmes Informatiques.

La faculté vise à former des informaticiens compétents, maîtrisant à la fois les techniques et les concepts fondamentaux. Cet enseignement est conçu pour permettre aux diplômés de poursuivre des études post-universitaires ou de s'insérer directement sur le marché du travail dans des domaines clés de l'informatique, où ils peuvent mettre en pratique leurs connaissances.

La formation offerte par la faculté est structurée autour de plusieurs axes :

- **Fondamental** : Les étudiants acquièrent les concepts de base et les méthodes de travail essentielles.
- **Appliqué** : Cet aspect facilite l'apprentissage des concepts et encourage le développement de compétences professionnelles.
- **Évolutif** : Les programmes de la faculté sont conçus pour intégrer les avancées technologiques et répondre aux exigences changeantes du milieu professionnel.
- **Ouvert** : La faculté favorise le développement des compétences de communication essentielles pour le travail en équipe et la collaboration professionnelle.

### 1.2.4 Organigramme de la faculté informatique

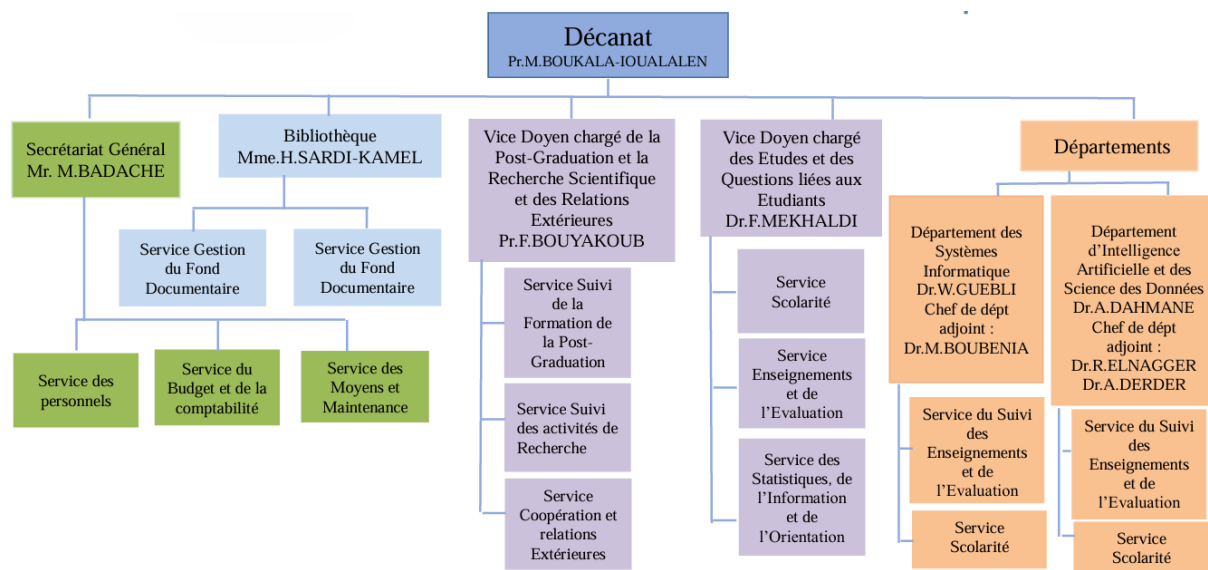


Figure 1.2: Organigramme de la faculté informatique



## 1.3 Cadre d'étude

Notre étude concerne le Vice-rectorat chargé de la scolarité et de la pédagogie (VRSP), le Centre des Ressources Informatiques (CRI), et les deux départements principaux de la Faculté d'Informatique avec le service de la Vice-doyenne chargée des études et des questions liées aux étudiants.

### 1.3.1 Vice-rectorat chargé de la scolarité et de la pédagogie (VRSP)

Le VRSP est responsable de la centralisation et de la gestion des données de disponibilité des salles. Ce service, représenté par Mme Samia et Mme Soumia, recueille et vérifie les informations relatives à l'occupation des salles et les disponibilités. Ces données sont ensuite transmises aux facultés pour l'élaboration des emplois du temps.

### 1.3.2 Centre des Ressources Informatiques (CRI)

Le CRI joue un rôle crucial dans l'intégration des données dans l'Espace Numérique de Travail (ENT) de l'USTHB. Ce centre est responsable de la maintenance et du bon fonctionnement de la plateforme ENT, assurant ainsi que les emplois du temps générés soient accessibles en ligne de manière fiable et sécurisée.

### 1.3.3 Les deux départements principaux de la Faculté d'Informatique

Les deux départements principaux de la Faculté d'Informatique sont :

- **Département d'Intelligence Artificielle et Sciences des Données**
- **Département des Systèmes Informatiques**

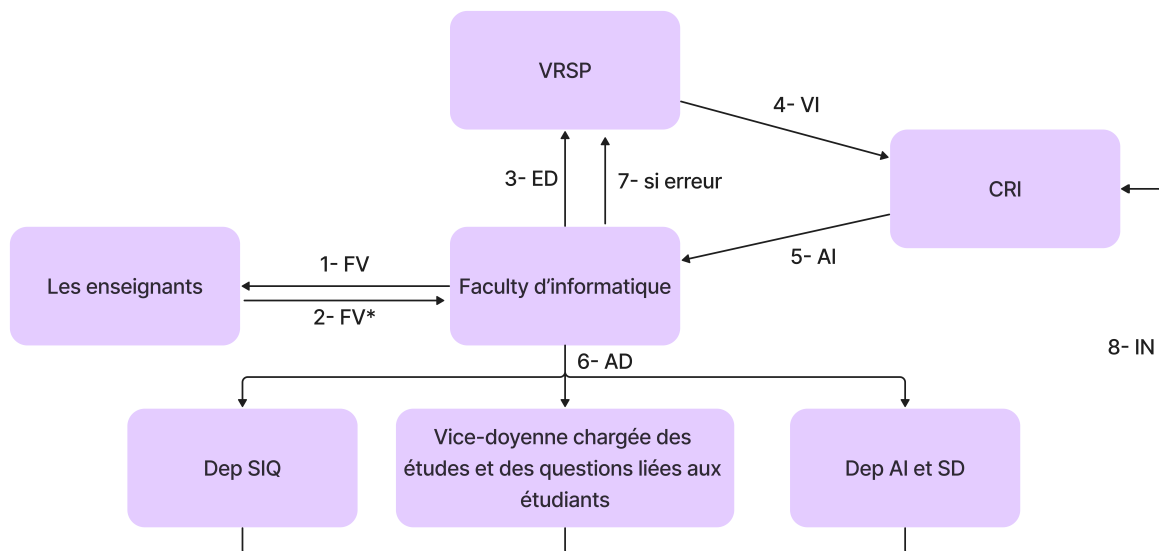
Chaque département a un responsables qui gère la collecte des données des enseignants via des outils comme Google Forms et Excel. Ces responsables sont en charge de l'organisation des horaires en fonction des disponibilités des enseignants et des salles.

### 1.3.4 le Vice-doyenne chargée des études et des questions liées aux étudiants

Ce service supervise la planification des emplois du temps pour les étudiants du tronc commun de première année et du cycle d'ingéniorat. La Vice-doyenne, avec son équipe, coordonne les efforts entre les départements pour assurer une distribution équitable et sans conflit des créneaux horaires pour les cours et les examens.

### 1.3.5 Processus de planification des emplois du temps

- **Collecte des données** : Les départements recueillent les données sur les disponibilités des enseignants et des salles via Google Forms et Excel.
- **Vérification et validation** : Le VRSP vérifie les données de disponibilité des salles et les transmet aux départements.
- **Intégration des données** : Le CRI intègre les données validées dans l'ENT.
- **Génération des emplois du temps** : Les responsables de chaque département utilisent les données intégrées pour générer les emplois du temps manuellement.
- **Modification et adaptation** : Les responsables de la faculté peuvent apporter des modifications en fonction des imprévus et des disponibilités changeantes.
- **Publication** : Les emplois du temps finaux sont publiés sur l'ENT pour consultation par les étudiants et le personnel.



1. **FV** : Fiche de vœux
2. **FV\*** : Fiche de vœux signée
3. **ED** : Envoi des données de la division des classes pour chaque faculté
4. **VI** : Validation et intégration
5. **AI** : Accès insertion
6. **AD** : Accès administrateur
7. **SI** : Si manque des salles ou ajouter des enseignants
8. **IN** : Insertion des emplois du temps

**Figure 1.3:** Diagramme de flux de la processus de planification

## 1.4 Les Ressources et les Contraintes

### 1.4.1 Les Ressources Humaines

- **Enseignants** : Disponibilité et compétences des enseignants.
- **Personnel administratif** : Responsables de la collecte des données et de la coordination des horaires.

### 1.4.2 Les Ressources Matérielles

- **Salles de cours** : Capacités, équipements disponibles, et accès.
- **Équipements informatiques** : : Ordinateurs, projecteurs, et autres technologies nécessaires pour l'enseignement.

### 1.4.3 Les Ressources Technologiques

- **Espace Numérique de Travail (ENT)** : Plateforme utilisée pour la gestion et la publication des emplois du temps.
- **Outils de collecte de données** : Google Forms, Excel, etc.

### 1.4.4 Les Contraintes Temporelles

- **Disponibilité des enseignants** : Horaires de travail, congés, et autres engagements.
- **Disponibilité des salles** : Heures de disponibilité et conflits potentiels.

### 1.4.5 Les Contraintes Logistiques

- **Capacité des salles** : Assurer que le nombre d'étudiants ne dépasse pas la capacité des salles.
- **Accès aux ressources** : Disponibilité et accès aux équipements nécessaires pour les cours.

### 1.4.6 Les Contraintes Pédagogique

- **Programmes d'études** : Respect des cursus et des exigences pédagogiques.
- **Équilibre des horaires** : Assurer une répartition équilibrée des cours tout au long de la semaine pour éviter la surcharge.

# Chapitre 2

## Conception

### 2.1 Introduction

La phase de conception est cruciale dans le développement d'une application, car elle permet de définir les fonctionnalités, l'architecture, et les composants du système. Ce chapitre présente la conception de l'application pour la génération automatique des emplois du temps, en décrivant les choix technologiques, l'architecture logicielle, et les interfaces utilisateur.

### 2.2 Analyse des besoins

Avant de commencer la conception, une analyse approfondie des besoins fonctionnels et non-fonctionnels est nécessaire. Voici les principaux besoins identifiés :

- **Besoins fonctionnels :**
  - Génération automatique des emplois du temps sans conflit.
  - Consultation des emplois du temps par les enseignants et les étudiants.
  - Modification des emplois du temps par les administrateurs.
- **Besoins non-fonctionnels :**
  - Performance et rapidité de génération.
  - Sécurité des données
  - Interface utilisateur conviviale.

### 2.3 Choix technologiques

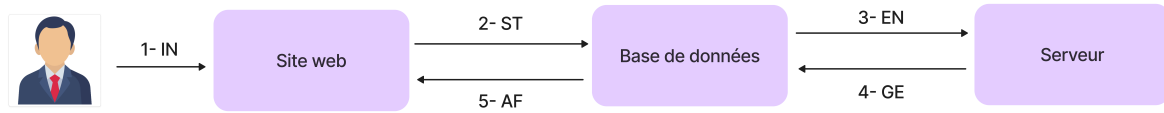
Le choix des technologies utilisées pour le développement de l'application (site web) est basé sur les besoins identifiés et les contraintes du projet. Les principales technologies

choisies sont :

1. **Frontend** : Nous avons utilisé **Next.js** et **React.js** pour construire l'interface utilisateur dynamique et réactive de notre plateforme. **Tailwind CSS** a été choisi pour la conception des styles en raison de sa flexibilité et de sa facilité d'utilisation, tandis que **TypeScript** nous a permis de bénéficier de la vérification statique des types et d'améliorer la qualité du code. Pour la sécurité, nous avons intégré **Bcrypt** pour le hachage des mots de passe.
2. **Backend** : Pour la logique métier et les réponses algorithmiques, nous avons utilisé **Flask**, un framework Python, connu pour sa simplicité et sa puissance. **Node.js** a été utilisé pour gérer les tâches côté serveur, offrant une performance élevée grâce à son modèle d'architecture basé sur les événements.
3. **Design** : Le Prototype de notre plateforme a été réalisée en utilisant **Figma**, **Illustrator** et **Photoshop** pour créer des maquettes et des éléments graphiques de haute qualité. **Flowbite** a été utilisé pour intégrer des composants d'interface utilisateur prêts à l'emploi.
4. **Base de données** : **MongoDB** a été choisi comme base de données en raison de sa flexibilité et de sa capacité à gérer de grandes quantités de données non structurées. **Prisma** a été utilisé comme ORM (Object-Relational Mapping) pour interagir avec la base de données de manière efficace.
5. **Collecte des données** : Nous avons utilisé **Python** pour effectuer du scraping, afin de collecter les données nécessaires et de les convertir selon notre schéma et model de base de données.
6. **Outils de développement** : Pour le développement et la gestion de notre projet, nous avons utilisé **Visual Studio Code (VSCode)** comme éditeur de code, **Postman** pour tester les API, **Docker** pour la conteneurisation et **GitHub** pour le contrôle de version et la collaboration.

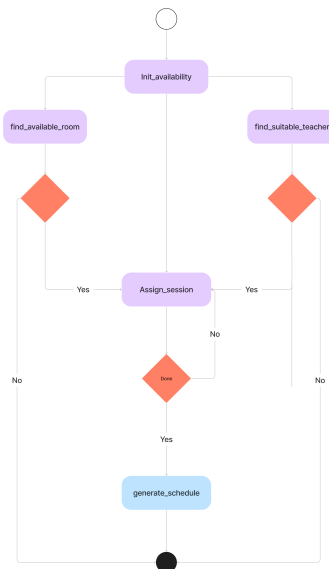
## 2.4 Architecture du système

### 2.4.1 Diagramme de l'architecture



1. **IN** : Insertion des donnees
2. **ST** : Stocker les donnee dans la base de donnees
3. **EN** : Envoi les données
4. **GE** : generation
5. **AF** : Affichage des emploi de temps

**Figure 2.1:** architecture de programme



**Figure 2.2:** Diagramme d'activity

## 2.5 Conception des modules

Le système de planification est conçu pour gérer et générer des emplois du temps pour diverses sections académiques au sein d'une institution. Il utilise plusieurs classes et méthodes interconnectées pour allouer efficacement des salles, des enseignants et des créneaux horaires pour les cours, tels que spécifiés dans la configuration JSON.

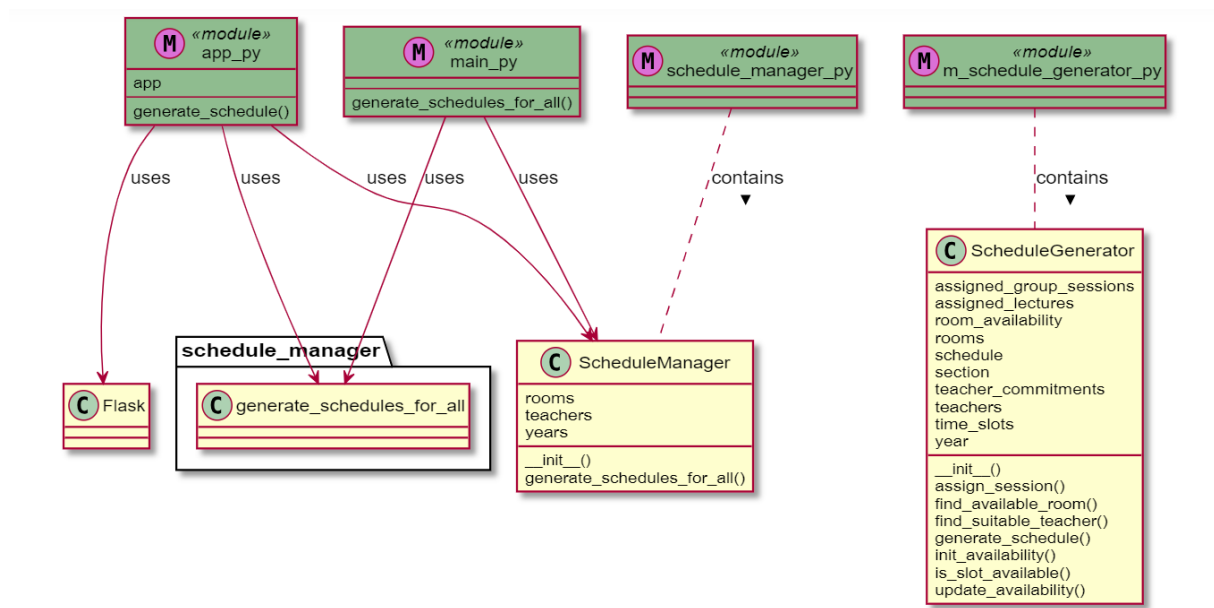


Figure 2.3: Diagramme de code

## Module : `m_schedule_generator.py`

### Classe : `ScheduleGenerator`

Cette classe encapsule la logique de génération des emplois du temps pour une section spécifique.

#### Attributs

- `year` : L'année académique pour laquelle l'emploi du temps est généré.
- `section` : La section académique pour laquelle l'emploi du temps est généré.
- `rooms` : Liste des salles disponibles pour la planification.
- `teachers` : Liste des enseignants disponibles, y compris leurs modules d'enseignement et leur disponibilité.
- `schedule` : Liste qui stocke les entrées de l'emploi du temps généré.
- `assigned_lectures` : Ensemble qui garde une trace des conférences déjà attribuées pour éviter les doublons.
- `assigned_group_sessions` : Ensemble pour suivre les sessions attribuées à des groupes spécifiques afin de prévenir les chevauchements.
- `teacher_commitments` : Dictionnaire pour suivre quand et où chaque enseignant est déjà planifié.
- `room_availability` : Dictionnaire pour suivre la disponibilité des salles.
- `time_slots` : Dictionnaire des créneaux horaires disponibles pour la planification.

des sessions.

## Méthodes

- `init_availability()`
  - Objectif : Initialise la disponibilité des salles et des enseignants.
  - Explication : Configure la disponibilité de chaque salle et enseignant pour tous les jours de la semaine.
- `find_suitable_teacher(module_name, day)`
  - Objectif : Trouve un enseignant approprié pour un module donné un jour spécifique.
  - Paramètres :
    - `module_name` : Le nom du module.
    - `day` : Le jour pour lequel un enseignant est nécessaire.
  - Retourne : Le nom d'un enseignant approprié, ou `None` si aucun enseignant n'est disponible.
- `find_available_room(session_type, day)`
  - Objectif : Trouve une salle disponible pour un type de session un jour spécifique.
  - Paramètres :
    - `session_type` : Le type de session (Conférence, TD, TP).
    - `day` : Le jour pour lequel une salle est nécessaire.
  - Retourne : Le nom d'une salle disponible, ou `None` si aucune salle n'est disponible.
- `assign_session(group, module, session_type, day)`
  - Objectif : Assigne une session à un groupe, un module et un jour, en assurant la disponibilité des salles et des enseignants.
  - Paramètres :
    - `group` : Le groupe pour lequel la session est attribuée.
    - `module` : Le module pour lequel la session est attribuée.
    - `session_type` : Le type de session (Conférence, TD, TP).
    - `day` : Le jour de la session.
  - Explication : Vérifie la disponibilité des salles et des enseignants et assigne la session si possible.
- `is_slot_available(slot, day)`
  - Objectif : Vérifie si un créneau spécifique est disponible pour une salle et un enseignant le jour donné.
  - Paramètres :



- slot : Le créneau horaire vérifié.
- day : Le jour vérifié.
- Retourne : True si le créneau est disponible, sinon False.
- update\_availability(room\_name, teacher\_name, day, slot)
  - Objectif : Met à jour la disponibilité d'une salle et d'un enseignant pour un créneau et un jour spécifiques.
  - Paramètres :
    - room\_name : Le nom de la salle.
    - teacher\_name : Le nom de l'enseignant.
    - day : Le jour de la session.
    - slot : Le créneau horaire de la session.
- generate\_schedule()
  - Objectif : Génère l'emploi du temps complet pour tous les modules de la section.
  - Retourne : L'emploi du temps généré.
  - Explication : Itère à travers tous les groupes de modules dans la section et planifie les conférences, TD et TP pour chaque module en utilisant les méthodes de planification respectives.

## Module : `schedule_manager.py`

### Classe : `ScheduleManager`

Cette classe gère la génération des emplois du temps pour plusieurs sections.

#### Attributs

- years : Contient des informations sur toutes les années académiques et leurs sections respectives.
- rooms : Contient des données sur les salles disponibles où les sessions peuvent être planifiées.
- teachers : Stocke les informations sur les enseignants disponibles pour mener des sessions.

#### Méthodes

- init(self, years, rooms, teachers)
  - Objectif : Initialise le `ScheduleManager` avec les années, les salles et les enseignants.

- Paramètres :
  - years : Dictionnaire contenant les années académiques et leurs sections respectives.
  - rooms : Liste des salles disponibles pour la planification.
  - teachers : Liste des enseignants disponibles, y compris leurs modules d'enseignement et leur disponibilité.
- generate\_schedules\_for\_all(self)
  - Objectif : Génère les emplois du temps pour toutes les sections en créant une instance de ScheduleGenerator pour chaque section et en générant des emplois du temps en utilisant les salles et les enseignants fournis.
  - Retourne : Une liste des emplois du temps générés pour toutes les années et sections.
  - Explication :
    - Itère à travers chaque année et section, génère des emplois du temps en utilisant ScheduleGenerator, et les compile dans un emploi du temps final.
    - Pour chaque année, itère à travers les spécialités et leurs sections, créant une instance de ScheduleGenerator pour chaque section.
    - Génère l'emploi du temps pour chaque section, ajoutant l'emploi du temps généré aux informations de la section respective.
    - Compile les emplois du temps pour toutes les sections dans l'emploi du temps final pour chaque année.

## Interaction des Composants

- **Configuration JSON** : Contient toutes les données nécessaires sur les sections, les modules, les enseignants et les salles. Ces données servent de fondation pour les opérations de planification.
- **Classe ScheduleManager** : Agit comme l'unité de contrôle central qui gère la planification pour toutes les sections en utilisant les instances de ScheduleGenerator. Elle accède aux ressources globales telles que la disponibilité des salles et des enseignants et coordonne les activités de planification entre les différentes sections pour éviter les conflits.
- **Classe ScheduleGenerator** : Responsable de la génération de l'emploi du temps pour une section spécifique. Elle interagit avec les couches de données pour récupérer la disponibilité des enseignants et les horaires des salles. Cette classe contient plusieurs méthodes pour gérer les différentes tâches de planification.
- **Interaction avec la Couche de Données** : ScheduleManager et ScheduleGene-

rator interagissent fortement avec les données configurées en JSON pour récupérer et mettre à jour les informations de planification. Cela inclut la vérification de la disponibilité des salles et des enseignants et la mise à jour des emplois du temps au fur et à mesure que les sessions sont attribuées.

- **Méthodes Utilitaires** : Des méthodes comme `find_suitable_teacher` et `find_available_room` fournissent un support utilitaire en vérifiant des conditions spécifiques qui influencent les décisions de planification, assurant qu'il n'y a pas de chevauchements ou de conflits dans les emplois du temps.
- **Flux d'Exécution** : Le système démarre en créant une instance de `ScheduleManager`. `ScheduleManager` itère ensuite à travers chaque section et crée une instance de `ScheduleGenerator` pour chacune d'elles. `ScheduleGenerator` utilise ses méthodes pour générer un emploi du temps complet pour sa section en fonction des données disponibles et des contraintes. Les résultats sont compilés dans l'emploi du temps final pour toutes les sections et peuvent être traités ou affichés.

## Intégration avec l'Application Flask

Le système de planification utilise une application web Flask pour fournir une interface conviviale permettant de gérer et de visualiser les emplois du temps générés. Flask est un framework web léger particulièrement bien adapté aux petites et moyennes applications web comme ce système de planification.

### Structure de l'Application Flask

- **app.py** : C'est le fichier principal qui contient la configuration de l'application Flask, les routes et la logique du serveur.

### Fonctionnalités Clés

- **Interface Utilisateur** : Fournit une interface interactive et facile à naviguer pour les utilisateurs afin de visualiser et de gérer les emplois du temps.
- **Points de Terminaison API** : Les routes Flask gèrent les requêtes pour générer, mettre à jour ou récupérer des emplois du temps, en interfaçant avec le backend du système de planification.
- **Visualisation des Données** : Intègre des outils pour visualiser les emplois du temps sous forme de calendrier afin de faciliter la compréhension et la gestion.

## Mise en Route

Pour configurer et exécuter le système de planification :

- **Installation** : Assurez-vous que Python 3.x est installé avec Flask. Installez les autres dépendances listées dans `requirements.txt`.
- **Exécution** : Exécutez l'application Flask avec `python app.py` et utilisez les points de terminaison API fournis pour interagir avec le système.

## Description du Schéma de la Base de Données

### Modèle Account

Le modèle **Account** représente les informations de compte des utilisateurs, y compris les détails d'authentification et les tokens d'accès.

- `id` : Identifiant unique du compte, généré automatiquement.
- `userId` : Référence à l'utilisateur associé au compte.
- `providerType` : Type de fournisseur d'authentification (par exemple, OAuth, LDAP).
- `providerId` : Identifiant du fournisseur d'authentification.
- `providerAccountId` : Identifiant du compte auprès du fournisseur.
- `refreshToken` : Token de rafraîchissement, utilisé pour obtenir un nouveau token d'accès.
- `accessToken` : Token d'accès pour les sessions authentifiées.
- `accessTokenExpires` : Date et heure d'expiration du token d'accès.
- `createdAt` : Date et heure de création du compte.
- `updatedAt` : Date et heure de la dernière mise à jour du compte.
- `user` : Relation avec le modèle **User**.

### Modèle Session

Le modèle **Session** représente les sessions d'authentification des utilisateurs.

- `id` : Identifiant unique de la session, généré automatiquement.
- `userId` : Référence à l'utilisateur associé à la session.
- `expires` : Date et heure d'expiration de la session.
- `sessionToken` : Token unique de la session.
- `accessToken` : Token d'accès unique de la session.
- `createdAt` : Date et heure de création de la session.
- `updatedAt` : Date et heure de la dernière mise à jour de la session.
- `user` : Relation avec le modèle **User**.

## Modèle User

Le modèle **User** représente les utilisateurs de l'application.

- **id** : Identifiant unique de l'utilisateur, généré automatiquement.
- **nom** : Nom de l'utilisateur.
- **email** : Adresse email de l'utilisateur, unique.
- **password** : Mot de passe de l'utilisateur.
- **createdAt** : Date et heure de création de l'utilisateur.
- **updatedAt** : Date et heure de la dernière mise à jour de l'utilisateur.
- **accounts** : Relation avec le modèle **Account**.
- **sessions** : Relation avec le modèle **Session**.

## Modèle VerificationRequest

Le modèle **VerificationRequest** représente les requêtes de vérification pour des opérations comme la récupération de mot de passe.

- **id** : Identifiant unique de la requête, généré automatiquement.
- **identifiant** : Identifiant de la requête (par exemple, email).
- **token** : Token unique pour la vérification.
- **expires** : Date et heure d'expiration de la requête.
- **createdAt** : Date et heure de création de la requête.
- **updatedAt** : Date et heure de la dernière mise à jour de la requête.

## Modèle Annee

Le modèle **Annee** représente les années académiques.

- **id** : Identifiant unique de l'année, généré automatiquement.
- **annee** : Année académique.
- **specialites** : Relation avec le modèle **Specialite**.

## Modèle Specialite

Le modèle **Specialite** représente les spécialités académiques.

- **id** : Identifiant unique de la spécialité, généré automatiquement.
- **nom** : Nom de la spécialité.
- **anneeId** : Référence à l'année académique associée.
- **annee** : Relation avec le modèle **Annee**.
- **sections** : Relation avec le modèle **Section**.

## Modèle Section

Le modèle **Section** représente les sections académiques.

- **id** : Identifiant unique de la section, généré automatiquement.
- **nom** : Nom de la section.
- **specialite\_name** : Nom de la spécialité.
- **specialite** : Relation avec le modèle **Specialite**.
- **specialiteId** : Référence à la spécialité associée.
- **annee** : Année académique de la section.
- **groupes** : Relation avec le modèle **Groupe**.
- **modules** : Relation avec le modèle **Module**.
- **capacite** : Capacité de la section.

## Modèle Groupe

Le modèle **Groupe** représente les groupes au sein des sections.

- **id** : Identifiant unique du groupe, généré automatiquement.
- **nom** : Nom du groupe.
- **section** : Relation avec le modèle **Section**.
- **sectionId** : Référence à la section associée.

## Modèle Module

Le modèle **Module** représente les modules d'enseignement.

- **id** : Identifiant unique du module, généré automatiquement.
- **nom\_module** : Nom du module.
- **nb\_cours** : Nombre de cours du module.
- **td** : Indique si le module comprend des travaux dirigés (TD).
- **tp** : Indique si le module comprend des travaux pratiques (TP).
- **section** : Relation avec le modèle **Section**.
- **sectionId** : Référence à la section associée.

## Modèle Room

Le modèle **Room** représente les salles disponibles pour les cours.

- **id** : Identifiant unique de la salle, généré automatiquement.
- **nom** : Nom de la salle.
- **type** : Type de la salle (par exemple, TP, Amphi).

- `capacite` : Capacité de la salle.
- `disponibilite` : Disponibilité de la salle.

## Modèle Professor

Le modèle `Professor` représente les enseignants de l'institution.

- `id` : Identifiant unique de l'enseignant, généré automatiquement.
- `nom` : Nom de l'enseignant.
- `prenom` : Prénom de l'enseignant.
- `gender` : Genre de l'enseignant.
- `email` : Adresse email de l'enseignant, unique.
- `numero_de_telephone` : Numéro de téléphone de l'enseignant.
- `date_de_naissance` : Date de naissance de l'enseignant.
- `grade` : Grade de l'enseignant.
- `availability_prof` : Disponibilité de l'enseignant.
- `modules` : Relation avec le modèle `ModuleAssignment`.

## Modèle ModuleAssignment

Le modèle `ModuleAssignment` représente l'affectation des modules aux enseignants.

- `id` : Identifiant unique de l'affectation, généré automatiquement.
- `professorId` : Référence à l'enseignant associé.
- `professor` : Relation avec le modèle `Professor`.
- `priority` : Priorité de l'affectation du module.
- `nom_module` : Nom du module.

## Modèle Configuration

Le modèle `Configuration` représente les configurations générales du système.

- `id` : Identifiant unique de la configuration, généré automatiquement.
- `key` : Clé de la configuration.
- `value` : Valeur de la configuration.

## Modèle AcademicYear

Le modèle `AcademicYear` représente les années académiques.

- `id` : Identifiant unique de l'année académique, généré automatiquement.
- `year` : Année académique.

- `specialities` : Relation avec le modèle `Speciality`.

## Modèle `Speciality`

Le modèle `Speciality` représente les spécialités académiques.

- `id` : Identifiant unique de la spécialité, généré automatiquement.
- `name` : Nom de la spécialité.
- `academicYearId` : Référence à l'année académique associée.
- `academicYear` : Relation avec le modèle `AcademicYear`.
- `sections` : Relation avec le modèle `SpecialitySection`.

## Modèle `SpecialitySection`

Le modèle `SpecialitySection` représente les sections au sein des spécialités.

- `id` : Identifiant unique de la section, généré automatiquement.
- `name` : Nom de la section.
- `specialityId` : Référence à la spécialité associée.
- `speciality` : Relation avec le modèle `Speciality`.
- `schedule` : Relation avec le modèle `ScheduleEntry`.

## Modèle `ScheduleEntry`

Le modèle `ScheduleEntry` représente les entrées de l'emploi du temps.

- `id` : Identifiant unique de l'entrée, généré automatiquement.
- `day` : Jour de la session.
- `group` : Groupe concerné par la session.
- `moduleName` : Nom du module.
- `room` : Salle où se déroule la session.
- `sessionType` : Type de session (par exemple, conférence, TD, TP).
- `slot` : Créneau horaire de la session.
- `teacher` : Enseignant responsable de la session.
- `time` : Heure de la session.
- `sectionId` : Référence à la section associée.
- `section` : Relation avec le modèle `SpecialitySection`.



# Chapitre 3

## Implementation

### 3.1 Interface d'utilisateur

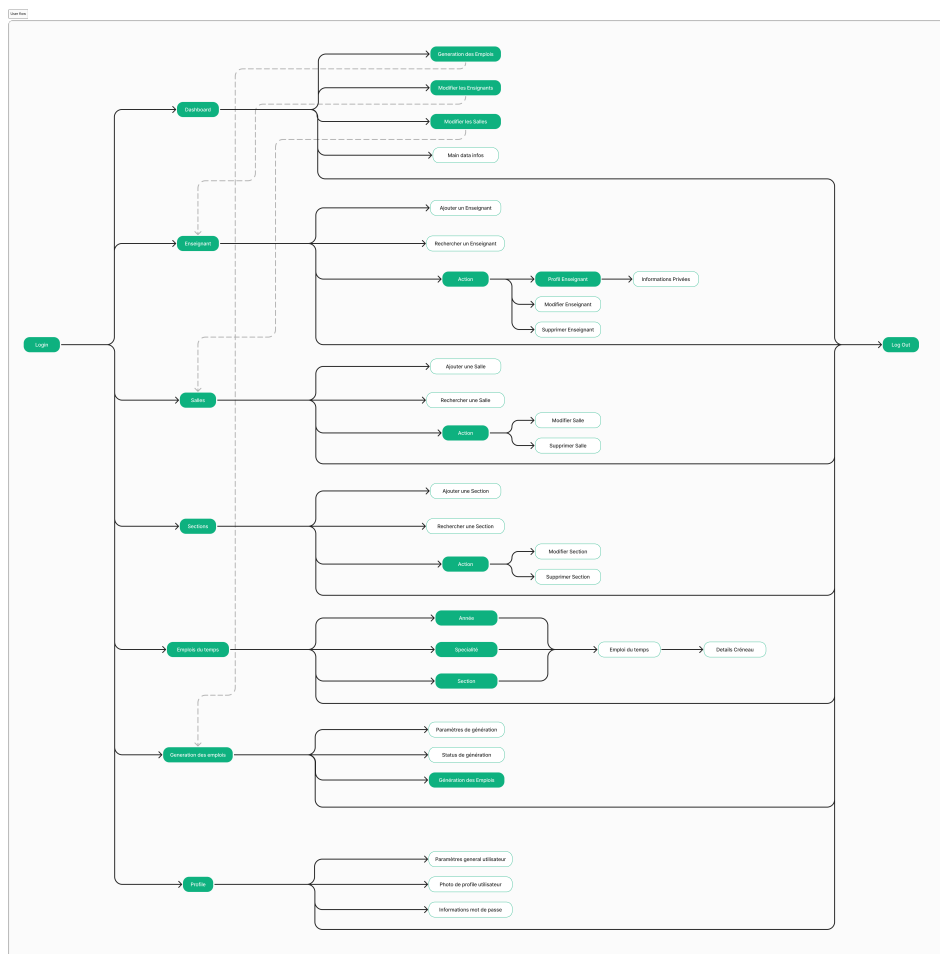
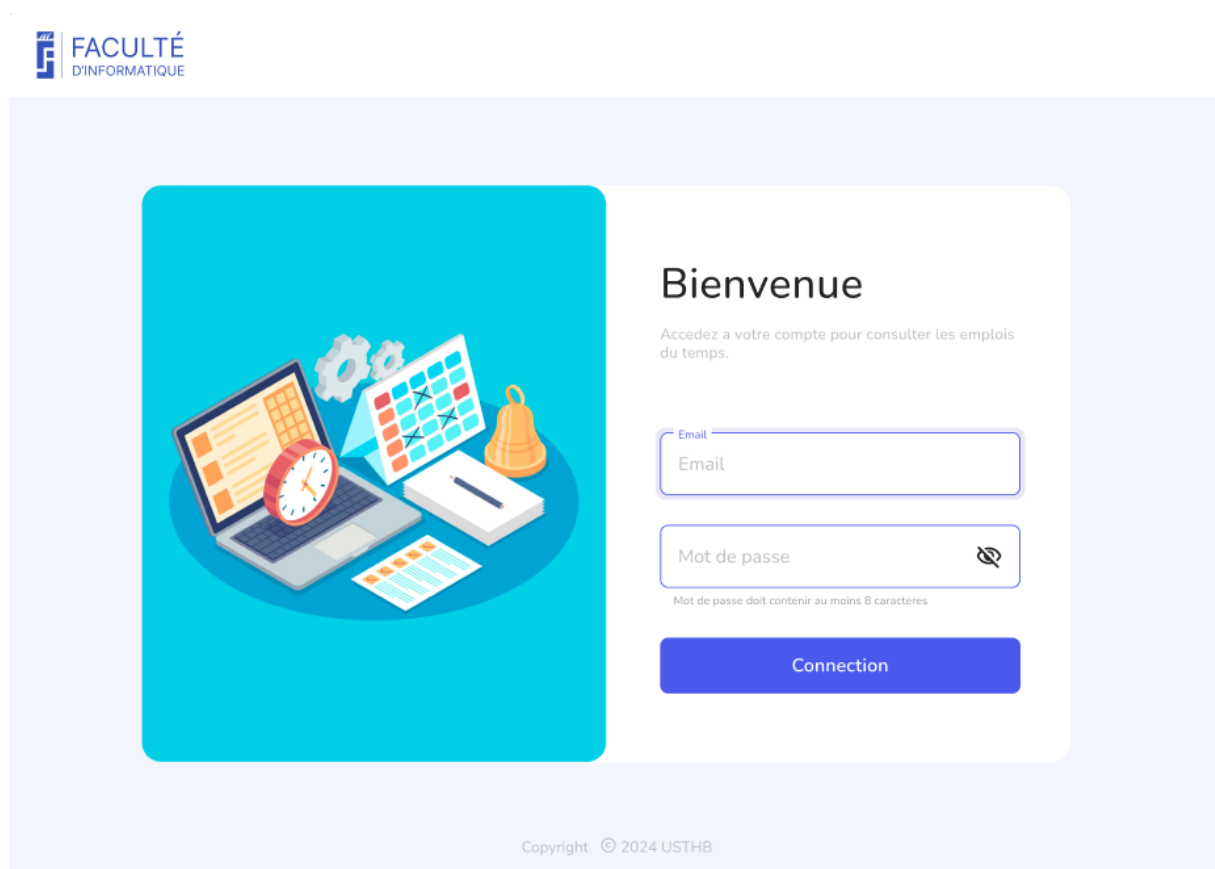


Figure 3.1: User flow

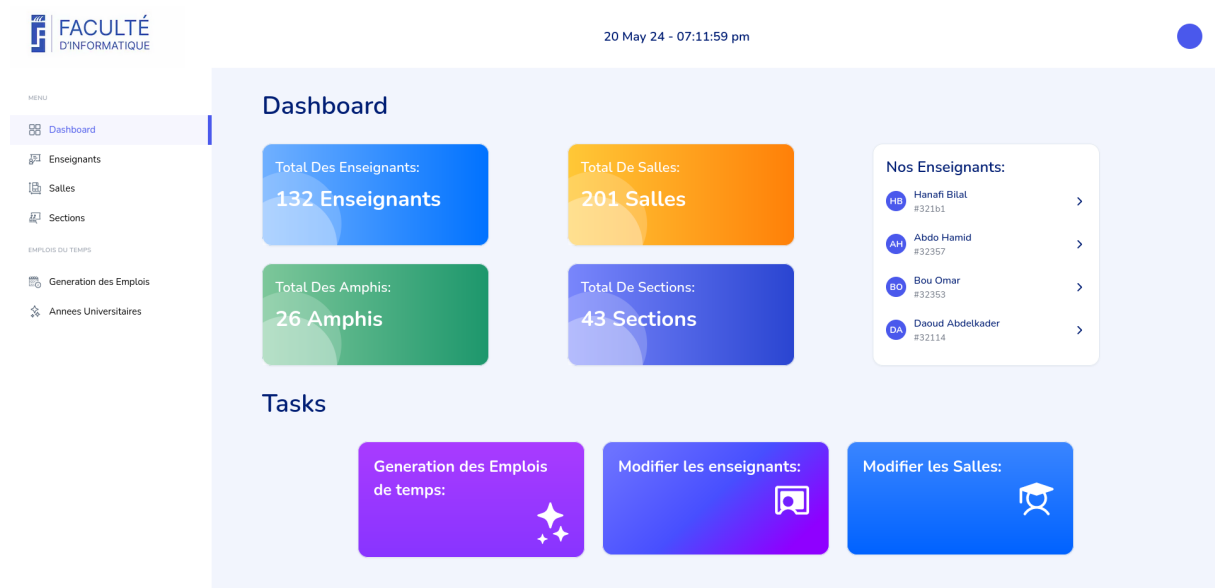
### 3.1.1 Page de Connexion



**Figure 3.2:** Page de Connexion

- La page de connexion de notre application est divisée en deux sections principales :
- Gauche : Illustration représentant la gestion du temps et la planification.
  - Droite : Formulaire d'authentification avec champs pour l'email et le mot de passe, et bouton "Connexion".

### 3.1.2 Tableau de Bord



**Figure 3.3:** Page des Enseignants

Le tableau de bord de notre application est conçu pour fournir une vue d'ensemble claire et concise des informations clés et des tâches disponibles pour la gestion des emplois du temps. Il est divisé en plusieurs sections principales pour une navigation facile et une gestion efficace des ressources académiques.

### 3.1.3 Page des Enseignants

The screenshot displays the 'Enseignants' page of a web application. The page features a sidebar on the left with navigation links: Dashboard, Enseignants (selected), Salles, Sections, EMPLOIS DU TEMPS, Generation des Emplois, and Annees Universitaires. The main content area is titled 'Enseignants' and includes a search bar labeled 'Rechercher Enseignant.'. Below the search bar is a table listing teachers with columns: Nom, Email, Grade, Phone, and Actions. The table contains 14 entries, each with a checkbox, a circular icon with initials, the teacher's name, email, grade (indicated by a star), phone number, and a three-dot menu for actions. At the bottom of the table, it says 'Page 1 of 14' and has 'Previous' and 'Next' buttons. A button labeled 'Ajouter un enseignant' is located in the top right corner of the main content area.

	Nom	Email	Grade	Phone	Actions
<input type="checkbox"/>	BM Bouhaddi Mariam	bouhaddi.mariam@hotmail.com	★ Maître Assistant Classe A	0715137453	...
<input type="checkbox"/>	BM Benrahou Mahdi	benrahou.mahdi@yahoo.com	★ Professeur	0698624378	...
<input type="checkbox"/>	DA Daoud Abdelkader	daoud.abdelkader@yahoo.com	★ Maître de Conférences Classe B	0743730472	...
<input type="checkbox"/>	CN Chalhoub Nadia	chalhoub.nadia@hotmail.com	★ Maître de Conférences Classe B	0613726916	...
<input type="checkbox"/>	HA Hamdi Amina	hamdi.amina@hotmail.com	★ Maître de Conférences Classe B	0557382563	...
<input type="checkbox"/>	HF Hamouda Farah	hamouda.farah@yahoo.com	★ Maître Assistant Classe A	0675895807	...
<input type="checkbox"/>	AA Arar Aicha	arar.aicha@yahoo.com	★ Professeur	0692169083	...
<input type="checkbox"/>	CB Chahine Bilal	chahine.bilal@yahoo.com	★ Professeur	0717086750	...
<input type="checkbox"/>	HI Hmida Idriss	hmida.idriss@outlook.com	★ Maître Assistant Classe A	0575584987	...
<input type="checkbox"/>	CA Chakir Abderrahmane	chakir.abderrahmane@gmail.com	★ Maître de Conférences Classe B	0755999513	...

Figure 3.4: Page des Enseignants

La page de gestion des enseignants permet aux utilisateurs de visualiser et de gérer les informations relatives aux enseignants.

- Liste des Enseignants : Affichage en tableau des informations des enseignants, y compris leur nom, email, grade, et numéro de téléphone.
- Recherche et Filtrage : Barre de recherche en haut de la liste pour trouver rapidement un enseignant spécifique en fonction de son nom ou d'autres critères.
- Ajout d'un Enseignant : Bouton "Ajouter un enseignant" en haut à droite permettant d'accéder à un formulaire pour ajouter un nouvel enseignant dans le système.
- Informations Affichées : Nom, Email, Grade (par exemple, Maître Assistant Classe A, Professeur), Téléphone, Actions (pour modifier ou supprimer les informations de l'enseignant.)

### 3.1.4 Formulaire d'Ajout d'un Enseignant

**Figure 3.5:** Formulaire d'Ajout d'un Enseignant

Le formulaire d'ajout d'un enseignant permet aux utilisateurs d'ajouter de nouvelles informations sur les enseignants de manière simple et organisée. Il est conçu pour être clair et facile à utiliser, en regroupant toutes les informations nécessaires dans une seule fenêtre modale. Sections Principales

- Fenêtre Modale : La fenêtre modale s'ouvre lorsque l'utilisateur clique sur le bouton "Ajouter un enseignant" dans la page de gestion des enseignants.
- Champs de Saisie :
  - Prénom : Champ pour saisir le prénom de l'enseignant.
  - Nom : Champ pour saisir le nom de l'enseignant.
  - Email : Champ pour saisir l'adresse email de l'enseignant.
  - Numéro de téléphone : Champ pour saisir le numéro de téléphone de l'enseignant.
  - Date de Naissance : Champ pour saisir la date de naissance de l'enseignant.
  - Grade : Champ pour sélectionner ou saisir le grade de l'enseignant.
  - Jours de disponibilité : Champ pour indiquer les jours de disponibilité de l'enseignant.
  - Modules prioritaires : Champ pour indiquer les modules prioritaires de l'enseignant.
- Sauvegarder : Bouton en bas à droite de la fenêtre modale permettant de sauvegarder les informations saisies. Ce bouton envoie les données au backend pour être

enregistrées dans la base de données.

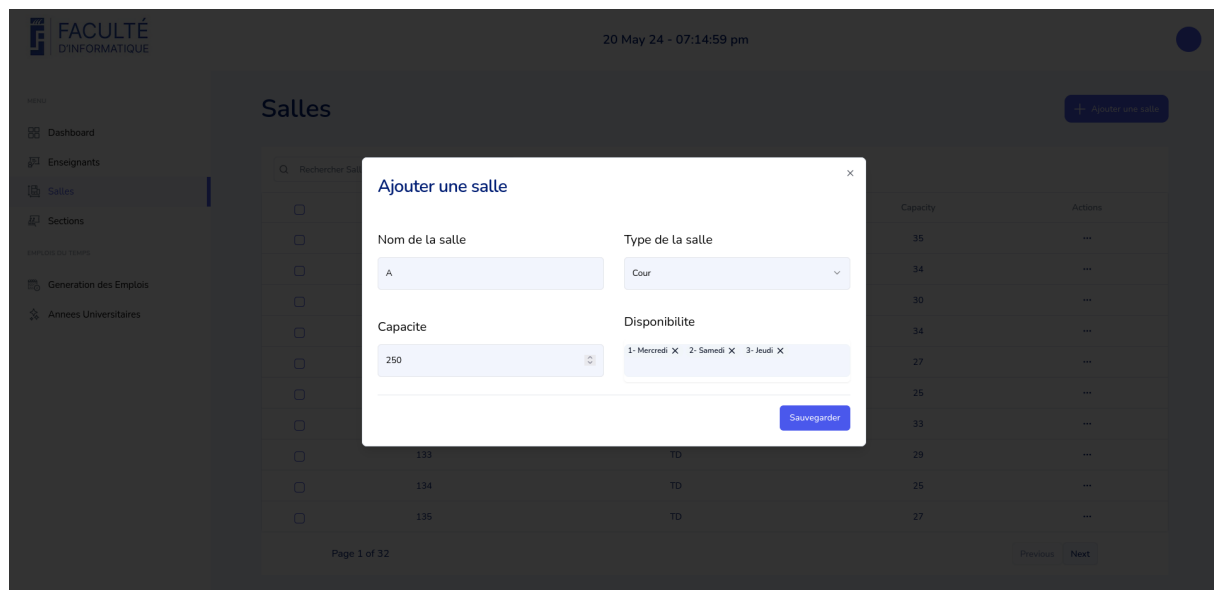
### 3.1.5 Page des Salles

The screenshot displays the 'Salles' page of a web application. The page has a light blue header with the date and time '25 Avril 2024 - 08:58:00 Pm' and a user profile 'Tarek Garici'. On the left, there is a sidebar menu with options: Dashboard, Enseignants, Salles (selected), Sections, Emplois du temps, Génération des emplois, and Profile. The main content area is titled 'Salles' and includes a search bar 'Rechercher Salle' and a filter button. Below this is a table with the following data:

	Nom Salle	Type Salle	Capacité	Action
<input type="checkbox"/>	126	Salle TP	45	...
<input type="checkbox"/>	D	Amphi	326	...
<input type="checkbox"/>	126	Salle TP	45	...
<input type="checkbox"/>	D	Amphi	326	...
<input type="checkbox"/>	357	Salle TD	52	...
<input type="checkbox"/>	357	Salle TD	52	...
<input type="checkbox"/>	D	Amphi	326	...
<input type="checkbox"/>	357	Salle TD	52	...
<input type="checkbox"/>	D	Amphi	326	...
<input type="checkbox"/>	126	Salle TP	45	...

At the bottom of the table, it says '02 page of 21'. There is a pagination bar with buttons for '<', '01', '02' (selected), '03', '...', '21', and '>'. A 'Log Out' button is located at the bottom left of the sidebar.

Figure 3.6: Page des Salles



**Figure 3.7:** Ajout des salles

La page de gestion des salles permet aux utilisateurs de visualiser et de gérer les informations relatives aux salles de la faculté.

- Liste des Salles : Affichage en tableau des informations des salles, y compris le nom de la salle, le type de salle (e.g., Salle TP, Amphi), la capacité, et les actions disponibles.
- Recherche et Filtrage : Barre de recherche en haut de la liste pour trouver rapidement une salle spécifique en fonction de son nom ou d'autres critères.
- Ajout d'une Salle : Bouton "Ajouter Salle" en haut à droite permettant d'accéder à un formulaire pour ajouter une nouvelle salle dans le système.

### 3.1.6 Page de Génération des Emplois du Temps

The screenshot displays the 'Génération Emploi du Temps' interface. On the left is a sidebar menu with options: Dashboard, Enseignants, Salles, Sections, Emplois du temps, Génération des emplois, and Profile. The main content area is titled 'Génération Emploi du Temps' and contains a 'Paramètres de génération' section with four input fields: 'Nombre de salles' (287), 'Nombre de sections' (31), 'Nombre d'enseignants' (130), and 'Nombre de spécialités' (12). Below these is a 'Status' section with a green bar indicating 'Généré avec succès'. At the bottom are two buttons: 'Générer' (blue) and 'Télécharger' (green). The top of the page shows the date '25 Avril 2024 - 08:58:00 Pm' and the user 'Tarek Garici'.

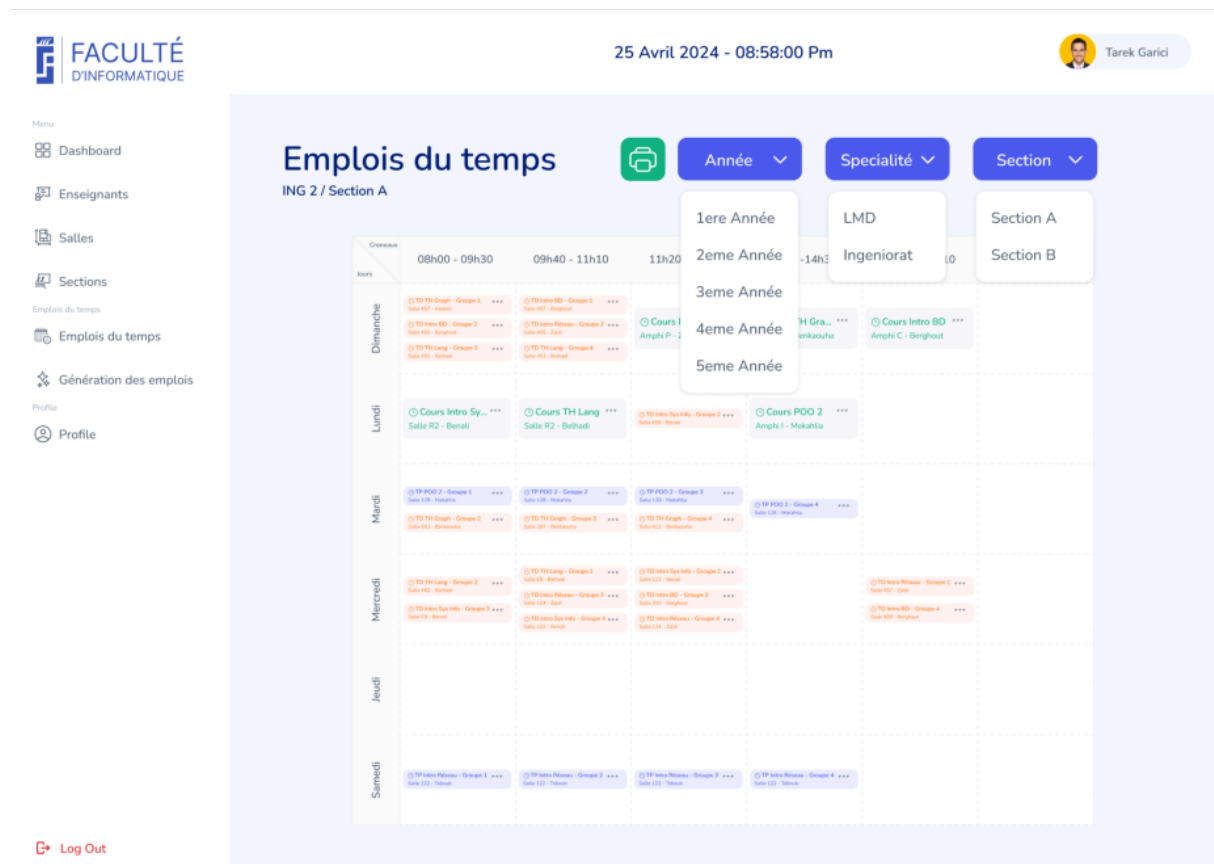
**Figure 3.8:** Page de Génération des Emplois du Temps

La page de génération des emplois du temps permet aux utilisateurs de créer automatiquement des emplois du temps basés sur les paramètres fournis. Elle est conçue pour être simple, facilitant la génération et le téléchargement des emplois du temps.

- Paramètres de Génération :
  - Nombre de Salles
  - Nombre d'Enseignants
  - Nombre de Sections
  - Nombre de Spécialités
- Statut : Généré avec Succès ou necessite regeneration ou necessite generation : Indicateur de statut indiquant que la génération des emplois du temps s'est effectuée avec succès ou necessite regeneration ou necessite generation.



### 3.1.7 Page des Emplois du Temps



**Figure 3.9:** Page des Emplois du Temps

La page des emplois du temps permet aux utilisateurs de visualiser les emplois du temps hebdomadaires pour différentes années, spécialités et sections.

— **Filtres de Sélection :**

- Année : Filtre permettant de sélectionner l'année d'étude (e.g., 1ère Année, 2ème Année).
- Spécialité : Filtre permettant de sélectionner la spécialité (e.g., LMD, Ingéniorat).
- Section : Filtre permettant de sélectionner la section (e.g., Section A, Section B).

— **Emploi du Temps :**

- Affichage.
- Détails des Cours.
- Couleurs et Codes : Utilisation de codes couleur pour distinguer différents types de cours (e.g., Cours, Travaux dirigés, Travaux pratiques).

# Conclusion Générale

La mise en œuvre de notre projet de génération automatique des emplois du temps pour la Faculté d'Informatique de l'USTHB, nous avons créé une solution qui répond aux besoins spécifiques de planification des cours, des salles et des enseignants.

Le projet a permis de résoudre plusieurs problématiques majeures, notamment les conflits d'horaires, la coordination inefficace, et le manque de flexibilité des systèmes existants. Grâce à l'automatisation et à l'utilisation d'algorithmes avancés, nous avons pu optimiser l'allocation des ressources et améliorer la qualité de la gestion des emplois du temps.

Les principaux avantages de notre solution incluent :

- **Efficacité** : Réduction des conflits d'horaires et optimisation de l'utilisation des salles et des enseignants.
- **Flexibilité** : Adaptation rapide aux changements et mises à jour faciles grâce à une architecture modulaire.
- **Accessibilité** : Interface utilisateur intuitive facilitant la consultation et la gestion des emplois du temps pour les étudiants et le personnel.

L'intégration avec une application web Flask offre une interface conviviale pour les utilisateurs, permettant une gestion simplifiée et une visualisation claire des emplois du temps. De plus, l'utilisation de technologies comme Next.js, React.js, MongoDB, et Prisma garantit la performance et la fiabilité de notre application.

En conclusion, ce projet a non seulement atteint ses objectifs initiaux mais a également ouvert la voie à de futures améliorations et extensions. Les défis rencontrés ont été surmontés grâce à une approche collaborative et innovante, et les solutions mises en place ont démontré leur efficacité et leur pertinence. Nous espérons que cette application contribuera significativement à améliorer la gestion des emplois du temps à la Faculté d'Informatique de l'USTHB, et servira de modèle pour d'autres projets similaires dans le domaine académique.

ce projet pourrait inclure l'intégration de fonctionnalités supplémentaires telles que l'optimisation basée sur l'intelligence artificielle.