

STAGE

**Année :2023/2024**

**Système de gestion du restaurant**

**Ingénierie Informatique et Réseaux**

Réalisé par : Elguessab Abderrafie

Encadré par :

Zakaria El Habousssi

# DEDICACE

***À Dieu***

*le tout puissant, nous rendons grâce pour nous avoir donné santé, patience, volonté et surtout raison.*

***À Mes parents***

*Je ne pourrais jamais assez-vous remercier pour tout ce que vous avez fait pour moi. Votre soutien inconditionnel et votre encouragement ont été la clé de ma réussite. Vous avez été là pour moi dans les moments difficiles et m'avez aidé à surmonter les obstacles. Votre présence dans ma vie est une bénédiction et je suis reconnaissant(e) de vous avoir à mes côtés.*

***À MES PROFESSEURS***

*Je tiens à exprimer ma gratitude envers tous mes professeurs qui ont contribué à mon éducation et à ma croissance personnelle. Votre dévouement et votre passion pour l'enseignement ont été une source d'inspiration pour moi. Vous m'avez appris à être curieux, à être persévérant(e) et à croire en moi-même. Je suis fier(e) de tout ce que j'ai accompli grâce à vos enseignements.*

***À MES COLLÈGUES***

*Je tiens à remercier tous mes collègues pour leur soutien et leur collaboration. Nous avons travaillé ensemble pour atteindre nos objectifs communs et nous avons réussi grâce à notre esprit d'équipe. Votre amitié et votre soutien ont été inestimables et je suis reconnaissant(e) de vous avoir comme collègues*

# REMERCIEMENTS

*Je souhaite exprimer ma sincère gratitude envers l'ensemble des enseignants de l'EMSI pour leur soutien constant et leur généreux partage de savoirs. Leur expertise et leur engagement envers l'éducation ont été une source d'inspiration ininterrompue. Je remercie l'EMSI de m'avoir offert cette précieuse opportunité de stage, qui a non seulement enrichi mes compétences académiques, mais a également élargi mes perspectives professionnelles. L'accueil chaleureux et l'accompagnement dont j'ai bénéficié tout au long de ce parcours ont joué un rôle essentiel dans mon développement.*

*Je désire transmettre mes remerciements les plus chaleureux à toute l'équipe de Hello World Agency pour la chance de collaborer avec vous durant mon stage. Cette expérience au sein de votre agence a été extrêmement formatrice et m'a permis d'appliquer mes connaissances dans un cadre dynamique et innovant. Votre encadrement et la confiance que vous m'avez accordée ont été cruciaux pour faire de ce stage un succès, et je suis reconnaissant pour tout ce que j'ai pu apprendre à vos côtés.*

*Je tiens à exprimer ma plus profonde reconnaissance à Zakaria El Haboussi et Anas Amejjad pour leur encadrement exceptionnel tout au long de mon stage. Leur direction, leurs conseils judicieux et leur disponibilité ont été d'une grande importance pour mon développement professionnel. Leur expertise et leur passion pour leur domaine ont non seulement enrichi mon apprentissage, mais ont également renforcé ma motivation à poursuivre cette voie. Je vous remercie pour votre soutien constant et pour avoir rendu cette expérience si précieuse.*

# RÉSUMÉ

Ce projet a pour objectif de concevoir un système de gestion complet destiné à un restaurant, organisé en trois segments distincts : un site web pour les administrateurs, un site web pour les clients, et une application mobile développée en Flutter pour les utilisateurs. Ce système a été élaboré afin de faciliter la gestion quotidienne du restaurant tout en garantissant une expérience utilisateur fluide et intuitive.

Le site web réservé aux administrateurs offre une analyse détaillée des activités du restaurant grâce à un tableau de bord interactif. Les administrateurs ont la possibilité de gérer les commandes, le menu, les stocks, le personnel, les utilisateurs, ainsi que des aspects financiers tels que la TVA. Cette interface robuste est conçue avec Spring Boot pour le back-end et React pour le front-end, assurant ainsi une solution performante et adaptable.

Le site web destiné aux clients leur permet de consulter le menu, de passer des commandes et de gérer leur profil, y compris leurs informations personnelles et leur historique de commandes. L'application mobile en Flutter propose une expérience similaire à celle du site web pour les utilisateurs, optimisée pour les appareils mobiles.

La gestion de la base de données du projet est assurée par MySQL, garantissant une administration sécurisée et efficace des données. L'architecture technique adoptée, qui combine Spring Boot, React, Flutter et MySQL, a été choisie pour sa robustesse, sa flexibilité et sa capacité à répondre aux exigences spécifiques du projet.

En conclusion, ce système de gestion de restaurant offre une solution intégrée et efficace pour la gestion des opérations internes et l'interaction avec les clients, en s'appuyant sur des technologies modernes et éprouvées.

**Mots clés :** Système de management de restaurant, Spring Boot (Back-end), React (Front-end), Flutter (Mobile), MySQL (Base de données), Développement web et mobile.

# ABSTRACT

This project aims to design a comprehensive management system for a restaurant, organized into three distinct segments: a website for administrators, a website for customers, and a mobile application developed in Flutter for users. This system has been designed to facilitate the daily management of the restaurant while ensuring a smooth and intuitive user experience.

The website for administrators provides detailed analysis of the restaurant's activities through an interactive dashboard. Administrators have the ability to manage orders, the menu, inventory, staff, users, and financial aspects such as VAT. This robust interface is built with Spring Boot for the back-end and React for the front-end, ensuring a high-performance and adaptable solution.

The customer website allows users to view the menu, place orders, and manage their profile, including personal information and order history. The mobile application in Flutter offers a similar experience to the website, optimized for mobile devices.

Database management for the project is handled by MySQL, ensuring secure and efficient data administration. The chosen technical architecture, combining Spring Boot, React, Flutter, and MySQL, was selected for its robustness, flexibility, and ability to meet the project's specific requirements.

In conclusion, this restaurant management system provides an integrated and effective solution for internal operations and customer interaction, leveraging modern and proven technologies.

Keywords: Restaurant management system, Spring Boot (Back-end), React (Front-end), Flutter (Mobile), MySQL (Database), Web and mobile development.

## Table des matières :

[DEDICACE II](#_bookmark0) [REMERCIEMENTS III](#_bookmark1) [RÉSUMÉ IV](#_bookmark2) [ABSTRACT V](#_bookmark3)

[Table des matières : VI](#_bookmark4) [Liste des figures VIII](#_bookmark5) [Glossaire IX](#_bookmark6) [Introduction Générale 1](#_bookmark7)

[Contexte General 2](#_bookmark8)

* 1. [Introduction 3](#_bookmark9)
  2. [Organisme d’accueil 3](#_bookmark10)
     1. [École Marocaine des Sciences de l'Ingénieur 3](#_bookmark11)
  3. [Présentation du Projet 3](#_bookmark13)
     1. [Cadre général du projet 3](#_bookmark14)
     2. [Problématique 4](#_bookmark15)
     3. [Objectif 4](#_bookmark16)
  4. [Conclusion 5](#_bookmark17)

[Analyse et conception 6](#_bookmark18)

* 1. [Introduction 7](#_bookmark19)
  2. [Étude d’existence 7](#_bookmark20)
     1. [Explication du fonctionnement actuel 7](#_bookmark21)
     2. [Critique de l'existant 7](#_bookmark22)
     3. [Solution proposée 8](#_bookmark23)
  3. [Spécification des besoins 8](#_bookmark24)
     1. [Les besoins fonctionnels 8](#_bookmark25)
     2. [Les besoins non fonctionnels 10](#_bookmark26)
  4. [Analyse et conception 10](#_bookmark27)
     1. [Introduction 10](#_bookmark28)
     2. [Diagramme de Gant : 11](#_bookmark29)
     3. [Diagramme de cas d’utilisation 12](#_bookmark32)
     4. [Diagramme de Class 13](#_bookmark34)
     5. [Diagramme de séquences : 14](#_bookmark36)

[2.4.5 Conclusion 16](#_bookmark40)

[Réalisation 17](#_bookmark41)

* 1. [Introduction 18](#_bookmark42)
  2. [Environnement du travail 18](#_bookmark43)
     1. [Environnement matériel 18](#_bookmark44)
     2. [Environnement Logiciel 18](#_bookmark45)
  3. [Les interfaces de l’application 20](#_bookmark46)
     1. [Log in Page 20](#_bookmark47)
     2. [Gestion des étudiants 20](#_bookmark49)
     3. [Gestion des stages 21](#_bookmark51)
     4. [Gestion de Conventions 24](#_bookmark56)
  4. [Conclusion 25](#_bookmark59)

[Conclusion générale 26](#_bookmark60)

[Références 29](#_bookmark61)

## Liste des figures

[Figure 1:logo EMSI 3](#_bookmark12)

[Figure 2:logo UML 10](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268243)

[Figure 3: Table de diagramme de gant 11](#_bookmark30)

[Figure 4: diagramme de gant 11](#_bookmark31)

[Figure 5: Logo de "app.instagantt" 11](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268246)

[Figure 6:Diagramme de cas 12](#_bookmark33)

[Figure 7: diagramme de class. 13](#_bookmark35)

[Figure 8: diagramme de séquence décrit la procédure d’ajout d’un étudiant 14](#_bookmark37)

[Figure 9: diagramme de séquence décrit la procédure de générer PDF automatiquement(convention). 15](#_bookmark38)

[Figure 10: diagramme de séquence décrit la procédure de filtrer la liste des étudiants en fonction de class 16](#_bookmark39)

[Figure 11: Logo WAMP 18](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268252)

[Figure 12: Logo Visual Studio Code 19](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268253)

[Figure 13: Logo PHP 19](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268254)

[Figure 14 : Logo JavaScript 19](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268255)

[Figure 15: Logo MySQL 19](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268256)

[Figure 16: Logo Tailwind CSS 19](file://localhost/C:/Users/Elgue/OneDrive/Desktop/PFA%20Final.docx%23_Toc152268257)

[Figure 17: Log in Page 20](#_bookmark48)

[Figure 18: Page des étudiants 20](#_bookmark50)

[Figure 19: Page des stages d'été 1 21](#_bookmark52)

[Figure 20: Page des stages d'été 2 22](#_bookmark53)

[Figure 21: Page des stage PFE 1 23](#_bookmark54)

[Figure 22: Page des stage PFE 2 23](#_bookmark55)

[Figure 23: Page de téléchargement des conventions 24](#_bookmark57)

[Figure 24:Resultat de PDF (exemplaire) 25](#_bookmark58)

## Glossaire

|  |  |
| --- | --- |
| EMSI | École marocaine des sciences d’ingénieure |
| PHP | **Hypertext Preprocessor** |
| PFE | **Projet fin d’étude** |
| SE | **Stage d’été** |
| UML | **Unified Modeling Langage** |
| CSS | **Cascading Style Sheets** |

**Introduction Générale**

L'avènement du numérique a radicalement modifié les approches de gestion dans de nombreux domaines, y compris celui de la restauration. Dans ce cadre en constante mutation, l'adoption de systèmes de gestion informatisés s'avère indispensable pour optimiser l'efficacité opérationnelle, assurer la traçabilité et renforcer la transparence des processus. Ce rapport de projet a pour objectif de présenter le développement d'un système de gestion intégral destiné à un restaurant, élaboré pour faire face aux défis spécifiques de ce secteur.

Le projet de gestion de restaurant se compose de trois éléments principaux : un site web dédié aux administrateurs, un site web pour les clients, et une application mobile développée en Flutter pour les utilisateurs. Ce système intégré a été conçu pour faciliter la gestion quotidienne du restaurant, en proposant des solutions pour la gestion des commandes, du menu, des stocks, du personnel, ainsi que pour les aspects financiers tels que la TVA. L'ambition est d'assurer une expérience utilisateur fluide et intuitive tout en améliorant les opérations internes.

En tant que développeur impliqué dans ce projet, j'ai eu la chance de collaborer étroitement avec une équipe de développement chevronnée pour concevoir et mettre en œuvre cette solution ambitieuse. Ce rapport a pour but de documenter mon parcours de développement, de détailler les différentes étapes de conception et de réalisation du système, de présenter les fonctionnalités essentielles et les technologies employées, ainsi que d'évaluer les bénéfices potentiels que ce système pourrait apporter au restaurant. Il offre également une perspective sur la manière dont ce projet s'intègre dans le paysage numérique contemporain, en mettant en lumière les enjeux et les défis liés à la gestion d'un restaurant moderne.

CHAPTRE 1 :

Contexte General

### Introduction

Ce chapitre situe le contexte général du projet de fin d’étude, en présentant dans un premier temps l’organisme d’accueil, puis il décrit le cadre du projet, ses objectifs ainsi que la démarche de sa gestion

* 1. Organisme d’accueil :
     1. Hello World Agency



Figure 1: Hello World Agency

L'agence Hello World, établie à Marrakech, se positionne comme un acteur dynamique dans le domaine du développement web et mobile, ainsi que dans la conception de solutions numériques novatrices. Située au cœur de cette ville animée, elle se distingue par son savoir-faire en matière de création et de mise en œuvre de projets technologiques sur mesure, répondant aux exigences spécifiques de sa clientèle.

Composée d'une équipe de professionnels dévoués et expérimentés, Hello World s'engage à fournir des solutions de haute qualité qui allient fonctionnalité, esthétique et performance. Que ce soit pour la conception de sites web interactifs, le développement d'applications mobiles conviviales ou l'implémentation de systèmes de gestion intégrés, l'agence utilise des technologies avancées pour assurer la satisfaction de ses clients.

L'agence est également appréciée pour sa méthode collaborative et son dévouement à cerner les besoins particuliers de chaque client. En collaborant étroitement avec ses partenaires, Hello World garantit une exécution précise et efficace des projets, tout en respectant les délais impartis et les budgets alloués.

### Présentation du Projet :

* + 1. Cadre général du projet

Ce rapport de projet décrit le développement d'un système de gestion intégral destiné à un restaurant, dans le cadre d'une initiative visant à optimiser les opérations et à enrichir l'expérience des clients. L'objectif de ce projet est de concevoir et d'implémenter une solution numérique cohérente, incluant un site web pour les administrateurs, un site web pour les clients, ainsi qu'une application mobile développée en Flutter pour les utilisateurs. Ce système a pour but de simplifier la gestion des commandes, du menu, des stocks, du personnel et des aspects financiers, tout en proposant une interface intuitive et performante pour améliorer les opérations internes du restaurant et rehausser l'expérience globale des clients.

* + 1. Problématique

Comment optimiser la gestion des opérations d'un restaurant en intégrant un système numérique complet, remplaçant les méthodes traditionnelles et les outils manuels, tout en facilitant la gestion des commandes, des stocks, des employés et des aspects financiers ? Cette solution doit non seulement améliorer l'efficacité et la précision des opérations internes, mais également offrir une expérience utilisateur fluide pour les clients, permettant ainsi de maximiser la satisfaction client et de rationaliser les processus de gestion du restaurant.

* + 1. Objectif

**Pour les clients (Mobile et Web) :**

* **Authentification :**
  + Créer un compte / Se connecter
* **Visualisation :**
  + Voir les repas recommandés
  + Afficher la carte pour les localisations
  + Consulter l'ensemble du menu
  + Visualiser les informations nutritionnelles des plats
  + Filtrer les plats du menu
  + Afficher le profil utilisateur
  + Accéder à l'historique des commandes
  + Visualiser les plats favoris
  + Consulter l'historique des ordres
* **Gestion des commandes :**
  + Placer une commande depuis la carte
  + Voir les articles dans la carte
  + Vider la carte
  + Passer les commandes

**Pour les administrateurs (Web Client) :**

* **Tableau de bord :**
  + Accéder au tableau de bord interactif
* **Gestion des commandes :**
  + Gérer les commandes en cours et passées
* **Gestion des repas :**
  + Gérer les plats du menu
* **Gestion des employés :**
  + Gérer les informations et les horaires des employés
* **Gestion des utilisateurs :**
  + Gérer les comptes et les informations des clients
* **Gestion du stock :**
  + Gérer les niveaux de stock des ingrédients et des fournitures
* **Analyse financière :**
  + Générer des indices financiers (TVA, etc.)
  1. Méthodologie et planification :
     1. Introduction :

Pour assurer le succès de notre projet de gestion de restaurant, il est essentiel d'adopter une méthodologie de travail structurée et flexible. Le travail en équipe doit être organisé avec rigueur, et un plan quotidien précis doit être suivi pour anticiper et surmonter les imprévus. Étant donné que les besoins du projet peuvent évoluer, nous avons opté pour une approche agile qui permet de s'adapter aux changements et aux nouvelles exigences tout en maintenant la cohérence du développement. Cette méthodologie garantit une gestion réactive et continue des ajustements nécessaires, assurant ainsi la pérennité et l'efficacité du système.

* + 1. La méthode agile SCRUM

Pour structurer le développement de notre projet de gestion de restaurant, nous avons choisi la méthode de gestion Agile SCRUM. Cette approche vise à optimiser la productivité de l’équipe, structurer le projet de manière efficace, et maximiser le temps disponible. SCRUM se base sur trois rôles clés :

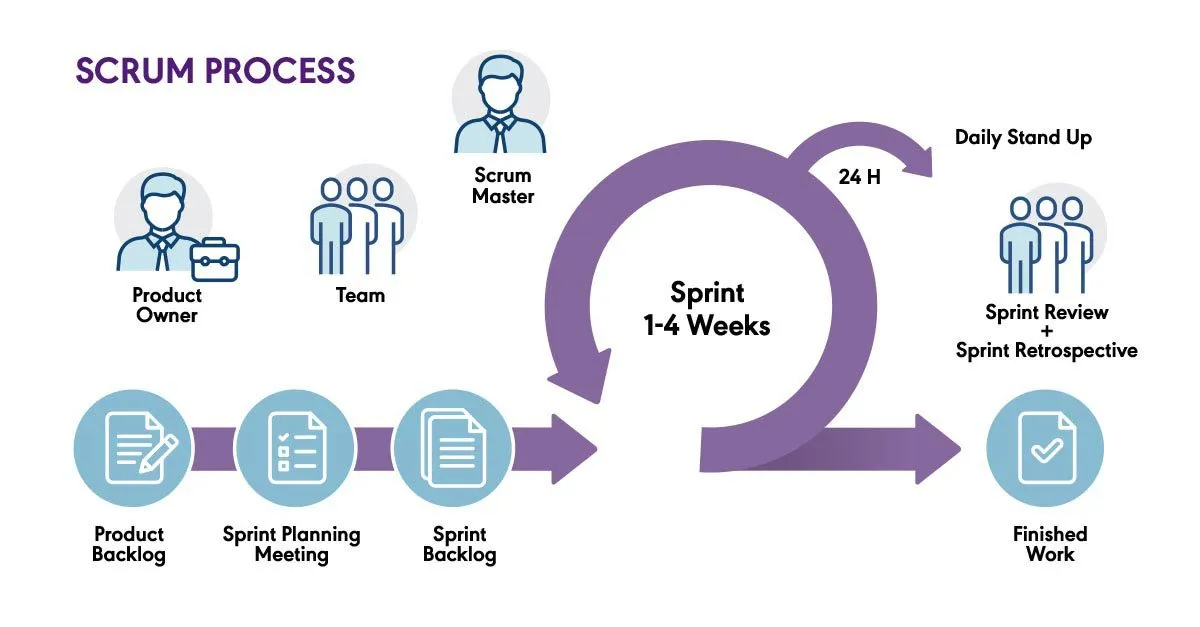
* Product Owner : Responsable de la définition des fonctionnalités du produit, il assure l’interface entre le client et l’équipe de développement, garantissant que les besoins du client sont clairement compris et intégrés.
* Scrum Master : Chargé de veiller à l'application des principes SCRUM, il facilite la communication au sein de l’équipe et intervient pour résoudre les conflits ou obstacles qui pourraient surgir.
* Équipe de développement : Composée des membres qui réalisent le développement des fonctionnalités du produit, cette équipe transforme les besoins en services ou produits concrets.

Figure 2 : Scrum Process

La méthode SCRUM est prisée pour sa simplicité, sa capacité à transformer le projet en un processus incrémental et itératif, et pour sa division du projet en cycles appelés "Sprints". Elle clarifie également les rôles et les responsabilités de chaque membre de l'équipe.

A screenshot of a computer

Description automatically generatedPour assurer une gestion efficace du projet, nous utilisons des outils de gestion comme Azure DevOps Boards. Cet outil nous aide à suivre le développement des user stories, à gérer les deadlines, les tâches assignées à chaque membre de l’équipe, et à établir les priorités. Les tâches dans chaque Sprint sont classifiées comme suit :

Figure 3 :Azure Boards

- NEW : Tâches à venir.

- ACTIVE : Tâches en cours de développement.

- RESOLVED : Tâches modifiées et résolues.

- CLOSED : Tâches terminées.

* 1. A screenshot of a computer

     Description automatically generatedA screenshot of a computer

     Description automatically generatedRépartition et planning de GANTT

Figure 4 : Gant Diagramme

Figure 5 : Gant Tasks

* 1. Conclusion

Ce chapitre présente Hello World Agency à Marrakech et définit les objectifs d'un système de gestion intégral pour un restaurant, visant à moderniser la gestion des commandes, des stocks, des ressources humaines et des finances par des solutions numériques. En clarifiant les objectifs et la problématique, ce chapitre souligne les besoins spécifiques du restaurant et la manière dont le projet y répondra, notamment par l'intégration d'un site web pour les administrateurs, d'un site pour les clients, d'une application mobile et d'une gestion des données via MySQL. Cette approche méthodique prépare les prochaines étapes du projet, garantissant que chaque fonctionnalité répond aux besoins opérationnels et aux attentes des utilisateurs.

CHAPTRE 2 :

Analyse et conception

### Introduction

Dans ce chapitre nous se concentrerons sur l’analyse des besoins du projet, l’aspect de la gestion et la conception du projet adopté.

1. 1. Étude d’existence
      1. Explication du fonctionnement actuel :

À l'heure actuelle, la gestion des commandes, des menus et des profils clients dans les établissements de restauration se fait souvent de manière traditionnelle ou semi-automatisée. Dans de nombreux cas, les clients doivent se déplacer physiquement au restaurant pour passer leurs commandes, ce qui restreint leur accès aux services pendant les heures d'ouverture. Cette situation peut être particulièrement contraignante pour ceux qui ont des emplois du temps chargés ou qui préfèrent simplement profiter de la commodité de commander depuis chez eux.

Pour les clients qui souhaitent commander à distance, il est fréquent qu'ils aient recours à des applications tierces comme Glovo, Jumia Food ou d'autres plateformes de livraison. Bien que ces applications offrent une solution pratique pour passer des commandes en ligne, elles présentent également des inconvénients notables. En effet, ces plateformes ne sont pas intégrées directement au système interne du restaurant, ce qui complique la gestion des commandes et peut entraîner des erreurs dans le traitement des demandes. Les restaurants doivent alors compter sur des intermédiaires pour gérer une part significative de leur service client, ce qui peut engendrer des délais supplémentaires et une perte de contrôle sur l'expérience client.

De plus, cette dépendance vis-à-vis des applications tierces peut entraîner des répercussions financières pour les établissements de restauration. Les commissions prélevées par ces plateformes peuvent réduire les marges bénéficiaires, rendant la rentabilité plus difficile à atteindre. Par ailleurs, les restaurants perdent une partie de la relation directe avec leurs clients, ce qui limite leur capacité à fidéliser leur clientèle et à recueillir des données précieuses sur les préférences des consommateurs.

* + 1. Critique de l'existant :

À l'heure actuelle, la gestion des opérations dans le restaurant s'appuie sur des méthodes conventionnelles et des outils manuels, ce qui engendre de nombreuses inefficacités. Les processus liés aux commandes, à la gestion des stocks, au suivi des employés et aux aspects financiers sont souvent réalisés à l'aide de systèmes non connectés, de documents papier et de tableurs Excel. Cette méthode fragmentée et manuelle entraîne plusieurs problèmes significatifs.

* Gestion des Commandes : Les commandes sont souvent prises à la main ou via des systèmes peu intégrés, ce qui peut entraîner des erreurs de saisie, des doublons, et des délais dans la communication des informations entre la salle et la cuisine.
* Gestion des Stocks : La gestion des stocks est réalisée manuellement, ce qui peut mener à des erreurs de suivi, des ruptures de stock imprévues, et des difficultés dans la prévision des besoins.
* Suivi des Employés : Les horaires et les informations relatives aux employés sont souvent gérés via des documents papier ou des tableurs, augmentant le risque d'erreurs et compliquant le suivi des heures de travail et des performances.
* Aspects Financiers : Les aspects financiers, tels que la gestion de la TVA et autres indices financiers, sont souvent traités de manière ad hoc, rendant les calculs sujets à des erreurs humaines et compliquant la production de rapports financiers précis.
* Expérience Client : Actuellement, l'expérience client n'est pas optimisée par les méthodes utilisées par le restaurant. Les clients doivent soit se rendre physiquement au restaurant pour passer leurs commandes, soit utiliser des applications externes telles que « Glovo » ou Jumia Food. Cette situation entraîne des inconvénients tels que des délais de traitement plus longs, un manque de visibilité sur les options du menu directement depuis le site du restaurant, et une gestion limitée de leurs profils et historiques de commandes. Ces méthodes fragmentées ne permettent pas une intégration fluide et cohérente de l'expérience client, ce qui peut réduire la satisfaction et l'engagement des clients envers le restaurant.

Ces limitations résultent en une perte d'efficacité opérationnelle, une augmentation des erreurs et une gestion moins transparente des informations. La transition vers un système numérique intégré, combinant un site web pour les administrateurs et les clients, ainsi qu'une application mobile, vise à résoudre ces problèmes en offrant une solution centralisée et automatisée. Ce changement permettra d'améliorer la précision des opérations, de faciliter la gestion des ressources, et d'enrichir l'expérience client.

* + 1. Solution proposée

La Pour répondre aux défis de l'administration des restaurants, il est essentiel d'adopter une solution moderne et intégrée. La proposition consiste en un système de gestion informatisé et centralisé pour optimiser l'efficacité opérationnelle et l'expérience client.

* Amélioration de l'Expérience Client

Cette solution améliore l'expérience client en offrant une plateforme dédiée pour consulter le menu, passer des commandes en ligne, gérer son profil et suivre l'historique des commandes via une application mobile ou un site web. Cela favorise une interaction plus fluide et personnalisée entre le restaurant et sa clientèle.

* Gestion des Commandes et du Menu

Le système automatise la prise de commande, réduit les erreurs et accélère le service. Les administrateurs peuvent gérer le menu en temps réel, offrant flexibilité et réactivité face aux demandes des clients.

* Gestion des Stocks et du Personnel

La solution permet un suivi rigoureux des stocks avec alertes automatiques pour les réapprovisionnements, minimisant les ruptures. La gestion du personnel est centralisée, facilitant leur gestion

* 1. Spécification des besoins
     1. Les besoins fonctionnels

**Authentification :**

* **Connexion sécurisée :**  
  Les utilisateurs doivent pouvoir se connecter de manière sécurisée sur le site web et l'application mobile.
* **Création de compte :**  
  Les utilisateurs doivent pouvoir créer un compte en fournissant les informations nécessaires (nom, email, mot de passe, etc.).

**Gestion des Clients :**

* **Visualisation du menu :**  
  Les utilisateurs doivent pouvoir consulter le menu complet du restaurant, y compris les détails des plats (nom, description, prix, etc.).
* **Filtrage du menu :**  
  Les utilisateurs doivent avoir la possibilité de filtrer le menu en fonction de divers critères (catégorie, prix, préférences alimentaires, etc.).
* **Historique des commandes :**  
  Les utilisateurs doivent pouvoir accéder à l'historique de leurs commandes passées, avec des détails sur chaque commande.
* **Gestion du profil :**  
  Les utilisateurs doivent pouvoir visualiser et modifier leurs informations personnelles, telles que leur nom, email, adresse, etc.
* **Suivi des repas favoris :**  
  Les utilisateurs doivent pouvoir marquer certains plats comme favoris et les retrouver facilement.
* **Localisation sur une carte :**  
  Les utilisateurs doivent pouvoir visualiser l'emplacement du restaurant sur une carte intégrée.

**Gestion des Commandes :**

* **Placer une commande :**  
  Les utilisateurs doivent pouvoir sélectionner des plats et les ajouter à leur panier, puis passer une commande en ligne.
* **Gestion du panier :**  
  Les utilisateurs doivent pouvoir visualiser, modifier et vider leur panier avant de passer une commande.

**Gestion des Commandes et du Menu (Côté Administrateur) :**

* **Tableau de bord :**  
  Les administrateurs doivent avoir accès à un tableau de bord qui affiche les statistiques clés (nombre de commandes, chiffre d'affaires, etc.).
* **Gestion des commandes :**  
  Les administrateurs doivent pouvoir visualiser l'état des commandes reçues.
* **Gestion du menu :**  
  Les administrateurs doivent pouvoir ajouter, modifier ou supprimer des plats du menu.

**Gestion des Employés :**

* **Gestion du personnel :**  
  Les administrateurs doivent pouvoir gérer les informations des employés, telles que leur nom, poste, horaires, etc.

**Gestion des Stocks :**

* **Suivi des stocks :**  
  Les administrateurs doivent pouvoir suivre les niveaux de stock des ingrédients et recevoir des alertes pour les réapprovisionnements.

**Génération des Indices Financiers :**

* **Calcul des indices financiers :**  
  Les administrateurs doivent pouvoir générer automatiquement des rapports financiers (TVA, marges bénéficiaires, etc.).
* **Téléchargement de rapports :**  
  Les administrateurs doivent pouvoir télécharger ces rapports au format PDF.
  + 1. Les besoins non fonctionnels
* **Performance :**
* **Temps de réponse rapide :**
* **Sécurité :**
* **Protection des données :**
* **Scalabilité :**
* **Évolutivité :**  
  **Expérience utilisateur (UX) :**
* **Interface intuitive :**
* **Compatibilité multi-plateforme :**
* **Maintenance :**
* **Facilité de maintenance :**
* **Fiabilité :**
* **Testabilité :**
* **Gestion des erreurs :**  
  1. Analyse et conception :
     1. Introduction :

A logo of a company

Description automatically generatedL'analyse et la conception sont des étapes essentielles dans le développement de tout projet, permettant d'organiser et de modéliser les composants du système avant la mise en œuvre.

Figure 6: UML logo

Cette approche méthodique assure une bonne compréhension et planification des éléments en fonction des exigences fonctionnelles et non fonctionnelles. Ce chapitre explore les méthodologies et outils utilisés pour l'analyse et la conception du projet de gestion de restaurant, en se basant sur l'UML (Unified Modeling Language), qui offre des outils efficaces pour la modélisation orientée objet.

L'UML permet de représenter visuellement les différents aspects du système, facilitant la transition vers la mise en œuvre. Grâce à des diagrammes tels que les cas d'utilisation, de séquences et de classes, nous pouvons capturer et organiser clairement les exigences, les interactions entre acteurs, ainsi que la structure des données et des processus métiers.

Nous mettrons en avant ces représentations et expliquerons leur mise en œuvre pour concevoir une solution robuste et évolutive pour le projet de gestion de restaurant.

* + 1. Diagramme de cas d’utilisation :
       - Client :

*A diagram of a company

Description automatically generated*

Figure 7 : Use Case (Client)

Le premier schéma illustre le cas d'utilisation d'un client interagissant avec une application dédiée à la commande de repas. Ce client a la possibilité d'effectuer plusieurs actions, telles que rechercher un plat, examiner les informations relatives à un repas, et l'ajouter à son panier. De plus, il peut gérer son profil, consulter l'historique de ses commandes, ses plats préférés, et finaliser le paiement de ses commandes, que ce soit par voie électronique ou en espèces

|  |  |  |
| --- | --- | --- |
| Cas d'utilisation | Acteur Principal | Détails Supplémentaires |
| S'authentifier | Client | La connexion permet au client d'accéder aux fonctionnalités de l'application réservées aux utilisateurs authentifiés tels que passer une commande . |
| Créer un compte | Client | Cette étape est nécessaire pour les nouveaux utilisateurs qui souhaitent utiliser l'application. Ils devront fournir des informations personnelles pour l'inscription. |
| Accéder à la Page d'Accueil | Client | La page d'accueil présente un aperçu des fonctionnalités principales de l'application, y compris l'accès aux menus |
| Consulter le Menu | Client | Le menu présente une liste des repas disponibles, avec la possibilité de filtrer ou de rechercher des plats spécifiques. |
| Consulter les détails du repas | Client | Les détails du repas incluent la description, les images , le prix, et d'autres informations nutritionnelles pertinentes. |
| Ajouter une Quantité au panier | Client | Après avoir consulté les détails du repas, le client peut choisir la quantité désirée et l'ajouter à son panier pour passer commande ultérieurement. |
| Consulter le Panier | Client | Le panier affiche tous les repas ajoutés, avec la possibilité de de supprimer des articles avant de passer à la validation de la commande. |
| Valider la Commande | Client | Cette étape finalise la sélection des repas et prépare le client à choisir une méthode de paiement. |
| Payer en Ligne | Client | Le paiement en ligne se fait via des options telles que les cartes de crédit, PayPal, ou d'autres services de paiement en ligne. |
| Payer en Espèces | Client | Cette option est pour les clients qui préfèrent payer en espèces au moment de la réception de la commande. |
| Consulter le Profil | Client | Le profil inclut les informations personnelles, les adresses de livraison, et les préférences de commande. |
| Consulter l'Historique | Client | L'historique affiche une liste des commandes passées, avec des détails tels que les dates, les repas commandés, et les montants payés. |
| Consulter les Informations de Compte | Client | Cette section permet au client de modifier ses informations personnelles, telles que l'email, le numéro de téléphone, ou le mot de passe. |
| Consulter les Plats Favoris | Client | Les plats favoris permettent au client de retrouver rapidement et de commander à nouveau ses repas préférés. |

* + - * A diagram of a company

        Description automatically generated with medium confidenceAdmin :

Figure 8 : Use Case (Admin)

Le deuxième diagramme représente le cas d'utilisation pour un administrateur qui gère le back-end d'un système de commande de repas. L'administrateur peut s'authentifier, accéder au tableau de bord, gérer les repas, les employés, les commandes, les utilisateurs, et le stock, ainsi qu'exporter des données sous différents formats. L'administrateur a également la possibilité de consulter les indices financiers et de gérer la galerie du système.

|  |  |  |
| --- | --- | --- |
| **Cas d'utilisation** | **Acteur Principal** | **Détails Supplémentaires** |
| S'authentifier | Administrateur | L'authentification permet à l'administrateur d'accéder aux fonctionnalités de gestion réservées. |
| Accéder au Tableau de Bord | Administrateur | Le tableau de bord offre une vue d'ensemble sur les activités, les commandes, et les statistiques financières. |
| Consulter les indices financiers | Administrateur | Ces indices fournissent des informations critiques sur les performances financières et les revenus générés par l'application. |
| Consulter les commandes | Administrateur | Cette fonctionnalité permet de gérer les commandes, de suivre leur statut, et de résoudre les problèmes éventuels. |
| Gérer les employés | Administrateur | Inclut l'ajout, la modification, et la suppression des informations des employés. |
| Gérer les commandes | Administrateur | L'administrateur peut modifier ou annuler des commandes en fonction des demandes des clients ou des circonstances. |
| Gérer les utilisateurs | Administrateur | Cela inclut la gestion des rôles, des permissions, et la résolution des problèmes liés aux comptes utilisateurs. |
| Gérer le stock | Administrateur | Cette fonction est cruciale pour s'assurer que les produits nécessaires sont disponibles pour la préparation des repas. |
| Exporter les données | Administrateur | Cette fonctionnalité est essentielle pour le « reporting », l'analyse des données, et l'archivage. |
| Ajouter un repas | Administrateur | Les repas peuvent être ajoutés avec toutes les informations nécessaires, y compris les ingrédients, les prix, et les images. |
| Modifier le Menu | Administrateur | Permet de mettre à jour les informations des repas, tels que les prix ou la disponibilité. |
| Supprimer un repas | Administrateur | Cette fonction est utilisée pour retirer les repas qui ne sont plus disponibles ou qui ne sont plus offerts. |
| Ajouter des détails sur les repas | Administrateur | Ces détails peuvent inclure des descriptions, des valeurs nutritionnelles, et des suggestions d'accompagnement. |
| Ajouter dans la Galerie | Administrateur | La galerie est utilisée pour promouvoir les repas ou les événements spéciaux dans l'application. |
| Modifier les informations | Administrateur | Inclut la modification des descriptions, des ingrédients, et d'autres attributs des repas. |
| Modifier la Galerie | Administrateur | Permet de remplacer ou de retirer les images existantes de la galerie. |

* + 1. Diagramme de Class

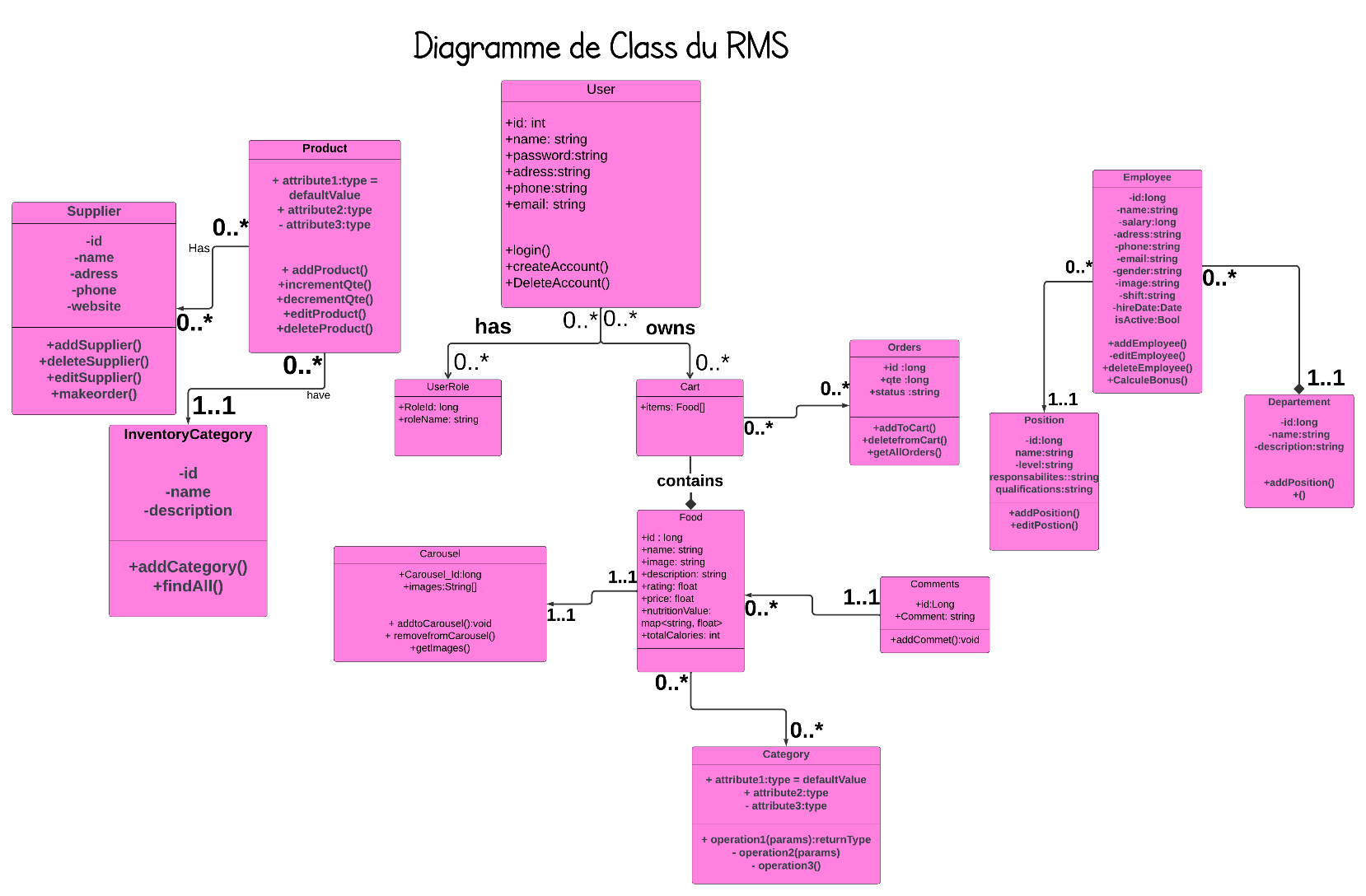
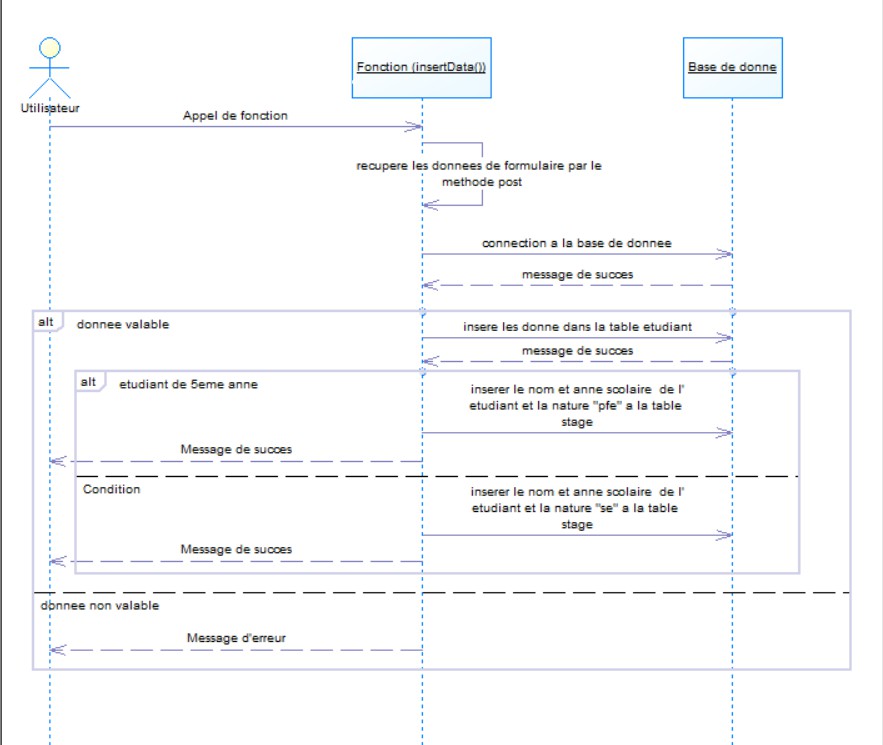


Figure 9 : Diagramme de Class

Le troisième diagramme est un diagramme de classes UML montrant la structure du modèle de données du système. Il inclut diverses entités comme Utilisateur, Produit, Commandes, Nourriture, Panier, et des classes associées telles que Fournisseur, Catégorie, et Employé. Les relations entre ces classes sont également représentées, indiquant comment les différents composants du système interagissent les uns avec les autres.

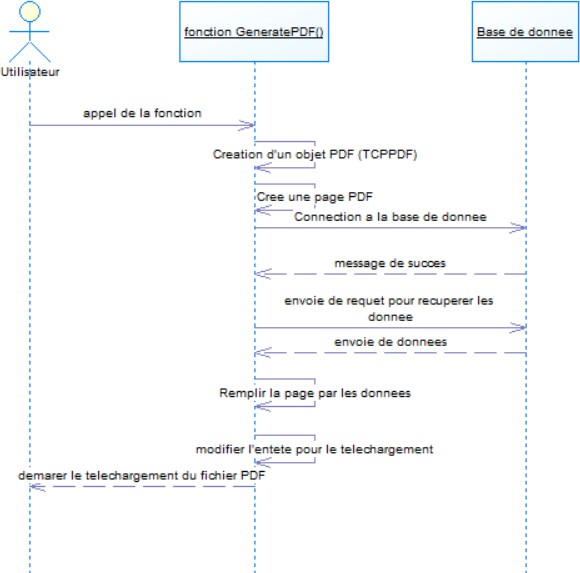
* + 1. Diagramme de séquences :
       1. *Ajouter un étudiant :*



*Figure 8: diagramme de séquence décrit la procédure d’ajout d’un étudiant.*

Ce schéma illustre la démarche d'inscription d'un étudiant dans la base de données en remplissant le formulaire approprié, ainsi que les étapes nécessaires pour créer un stage correspondant à son niveau académique.

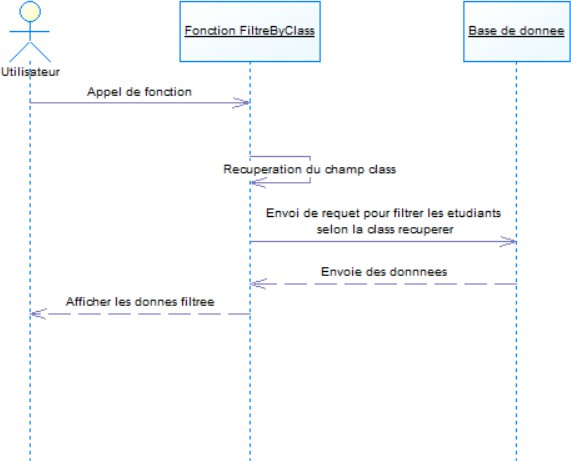
* + 1. 2.4.3.3 Générer PDF



*Figure 9: diagramme de séquence décrit la procédure de générer PDF automatiquement(convention).*

La méthode illustrée par ce schéma nous permet de générer automatiquement des conventions en utilisant un objet TCPPDF.

* + 1. Filtre (class)



*Figure 10: diagramme de séquence décrit la procédure de filtrer la liste des étudiants en fonction de class.*

Le schéma ci-dessus illustre le mode de fonctionnement de l'affichage des étudiants triés par classe en se servant du champ "classe".

* 1. Conclusion

Au cours de ce chapitre, nous avons exploré en profondeur la phase de conception du projet en mettant en lumière les principaux aspects de la modélisation à l'aide de diagrammes UML. Nous avons détaillé le diagramme de cas d'utilisation, qui a permis de définir les interactions entre les différents acteurs et le système. Nous avons également présenté les diagrammes de séquences, illustrant les échanges d'informations et les processus dynamiques nécessaires pour accomplir les tâches spécifiques au sein du système. Enfin, le diagramme de classes a été utilisé pour structurer les entités du système et leurs relations, fournissant une vue d'ensemble claire de l'architecture du projet.

Ces outils de modélisation jouent un rôle crucial en offrant une représentation visuelle et structurée du système, facilitant ainsi la compréhension et la communication des exigences fonctionnelles et techniques. Ils constituent des bases solides pour la phase de développement en aidant à anticiper et résoudre les éventuels problèmes avant la mise en œuvre. En synthèse, ce chapitre a jeté les bases nécessaires pour une réalisation efficace du projet, en assurant une cohérence entre la conception théorique et la réalisation pratique.

# CHAPITRE 3 :

## Réalisation

### Introduction

Dans ce chapitre, nous discuterons le cadre technique de notre application, nous explorerons les outils utilisés pour la réalisation de l'application, et nous explorerons les résultats obtenus sous forme des captures d'écran des interfaces les plus pertinentes.

### Environnement du travail

Pour la réalisation de ce projet, nous avons mis en place un environnement de travail complet et adapté aux besoins spécifiques du développement de l'application de gestion de restaurant, tant pour les utilisateurs que pour les administrateurs. Cet environnement inclut des éléments matériels et logiciels essentiels pour la conception, le développement, le déploiement, ainsi que la gestion des ressources.

* + 1. Environnement matériel

Pour le développement de l'application, les éléments matériels suivants ont été utilisés :

* + - A cartoon of a cat

      Description automatically generated**Ordinateur Personnel** : Un ordinateur personnel sous Windows a été employé pour la programmation, la conception des interfaces utilisateur, ainsi que pour les tests. Cet ordinateur a servi de poste principal pour le développement en React (web) et Flutter (mobile).
    - A pink and white logo

      Description automatically generated**Serveur Tomcat** : Le serveur Tomcat a été utilisé pour déployer et tester l'application Java en environnement de développement. Ce serveur a été essentiel pour héberger les applications web développées en Java et pour tester l'intégration avec d'autres composants du système

Figure 10 : Tomcat

* + - **Serveur Local (WAMP)** : Un serveur local a été configuré en utilisant WAMP (Windows, Apache, MySQL, PHP). Ce serveur a permis de développer et tester l'application à une échelle locale, en simulant l'environnement de production.

Figure 11 : WAMP

**Serveur de Stockage d'Images : Un serveur dédié a été mis en place pour héberger les images utilisées dans l'application, telles que les photos des plats et les avatars des utilisateurs. Ce serveur permet de gérer efficacement le stockage, l'accès rapide, et la mise à jour des images nécessaires pour les différentes interfaces de l'application. Il est essentiel pour assurer une expérience utilisateur fluide, notamment pour la gestion des menus et des commandes.**

Figure 12 : postimages

* + 1. Environnement Logiciel :
* **IDE (Environnement de Développement Intégré) :**

**Visual Studio Code : Utilisé pour écrire et gérer le code source de l'application, en particulier pour le développement front-end avec React et Flutter.**

**IntelliJ IDEA : IDE principal utilisé pour le développement backend en Java, notamment avec Spring Boot.**

* **Systèmes de Gestion de Version :**

**Git : Utilisé pour le contrôle de version, permettant la gestion des versions du code, la collaboration entre les membres de l'équipe, et la gestion des branches de développement.**

* **Frameworks et Bibliothèques :**

**React avec TypeScript : Utilisé pour le développement de l'interface utilisateur de l'application web, avec TypeScript pour un typage statique améliorant la qualité du code.**

**Flutter : Utilisé pour le développement d'applications mobiles multiplateformes (iOS et Android).**

**Spring Boot : Employé pour le développement des services backend en Java, facilitant la création d'API RESTful.**

**Material-UI (MUI) : Utilisé pour créer des composants React avec une apparence cohérente et moderne, assurant une meilleure expérience utilisateur.**

* **Base de Données :**

**MySQL : Base de données relationnelle utilisée pour gérer les données de l'application, incluant les informations des utilisateurs, commandes, stocks, etc.**

* **Outils de Gestion de Projet :**

**Azure DevOps : Utilisé pour la gestion des tâches, le suivi des user stories, la planification des sprints, et la collaboration au sein de l'équipe de développement.**

**Notion : Utilisé pour la documentation du projet, la gestion des connaissances et la planification.**

* 1. Architecture technique et design pattern
     1. Architecture technique

L'architecture technique de notre projet est conçue pour garantir une séparation claire des responsabilités, une modularité, et une flexibilité permettant de faire évoluer le système selon les besoins futurs.

* + - * Architecture à Trois Couches (Three-Tier Architecture)

L'application est structurée selon une architecture à trois couches, qui se compose des éléments suivants :

* **Couche de Présentation (Front-End) :**

Cette couche est responsable de l'interface utilisateur, à travers laquelle les utilisateurs interagissent avec le système. Elle est développée en utilisant React pour le web et Flutter pour les applications mobiles. Cette couche consomme les API fournies par la couche métier pour afficher les données et permettre l'interaction avec le système.

* **Couche Métier (Back-End) :**

La couche métier contient toute la logique de l'application, traitant les règles de gestion et orchestrant les interactions entre les différentes couches. Spring Boot est utilisé pour implémenter cette couche, assurant la gestion des transactions, la validation des données, et l'exécution des processus métiers.

* **Couche de Données (Data Layer) :**

Cette couche est dédiée à la gestion de la persistance des données. Elle utilise MySQL comme base de données relationnelle, gérée par l'ORM Hibernate, qui facilite les opérations CRUD (Create, Read, Update, Delete) en mappant les objets métiers aux tables de la base de données.

* + - * API RESTful

L'application adopte une architecture REST (Representational State Transfer) pour la communication entre le front-end et le back-end. Les API RESTful permettent une communication stateless entre le client et le serveur, garantissant une meilleure scalabilité et une interopérabilité avec d'autres systèmes

* + - * Hébergement et Déploiement

L'application est hébergée sur un serveur **Tomcat**, qui gère les requêtes du serveur backend. Pour les tests et le développement local, un serveur **WAMP** est utilisé. Cette configuration permet de tester l'application dans un environnement similaire à celui de la production avant le déploiement final.

* + 1. Design Patterns :

Les design patterns, ou patrons de conception, sont des solutions éprouvées pour résoudre des problèmes de conception communs. Ils permettent de structurer le code de manière claire et réutilisable. Voici les principaux patterns utilisés dans notre projet, en se concentrant sur ceux spécifiques à Spring Boot, Flutter, et React avec TypeScript.

* **Patterns dans Spring Boot**
  + **• Pattern MVC (Model-View-Controller):**

Sépare la logique métier (Model), la présentation (View), et le contrôle des flux de données (Controller). Spring Boot suit ce modèle pour organiser le code de manière modulaire et cohérente.

* + **• Pattern Singleton :**

Utilisé pour garantir qu'une seule instance d'une classe est créée. Spring Boot l'utilise pour les Beans gérés par le conteneur IoC (Inversion of Control).

* + **Pattern Builder :**

Implémenté via l'annotation @Builder dans Spring.

* + **• Pattern Factory :**

Implémenté via l'annotation @Bean dans Spring, permettant de définir des méthodes de création pour les objets nécessaires.

* + **• Pattern Dependency Injection (DI) :**

Au cœur de Spring Boot, DI permet l'injection de dépendances dans les composants sans les instancier manuellement, favorisant ainsi la testabilité et la modularité.

* + **• Pattern DAO (Data Access Object):**

Encapsule l'accès aux données. Utilisé en combinaison avec Spring Data JPA pour isoler la logique de persistance des données.

* + **• Pattern Observer :**

Implémenté par le système d'événements de Spring, où les classes peuvent s'abonner aux événements publiés dans l'application.

* **Patterns dans Flutter**
  + **Pattern MVVM (Model-View-ViewModel):**

Flutter favorise ce modèle via des bibliothèques comme provider ou riverpod, où le ViewModel contient la logique de présentation et la View (l'interface utilisateur) observe le ViewModel.

* + **Pattern Builder :**

Utilisé pour créer des widgets de manière flexible, en particulier pour les constructions complexes d'interfaces utilisateurs.

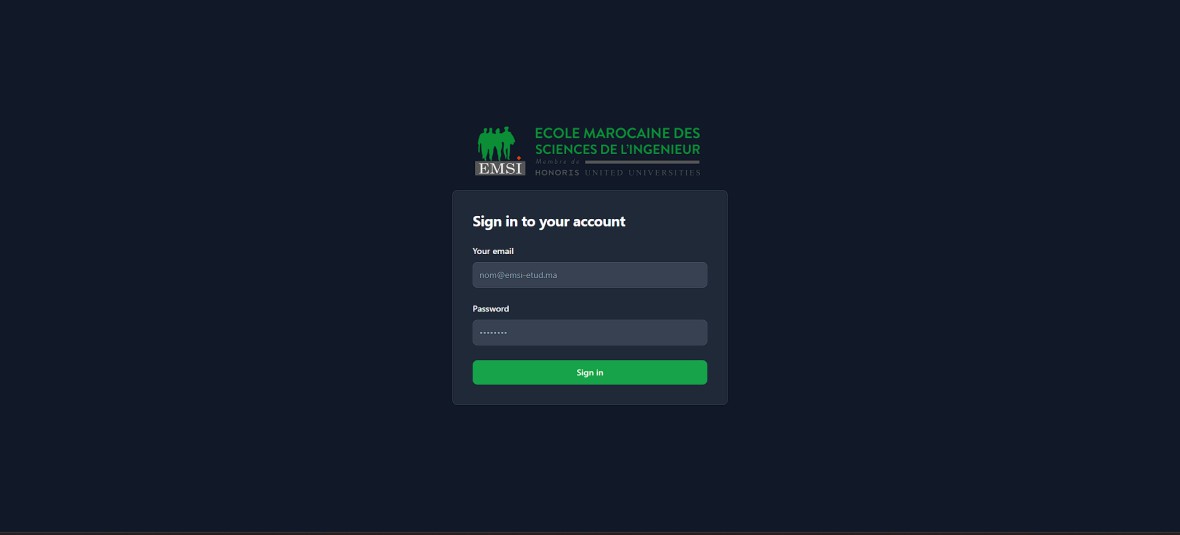
* + **Pattern Singleton :**

Utilisé pour créer des services partagés, comme la gestion de l'état global de l'application ou l'accès à une API.

* + **Pattern Observer :**

Implémenté par les Stream et ChangeNotifier, permettant aux widgets d'écouter et de réagir aux changements de données.

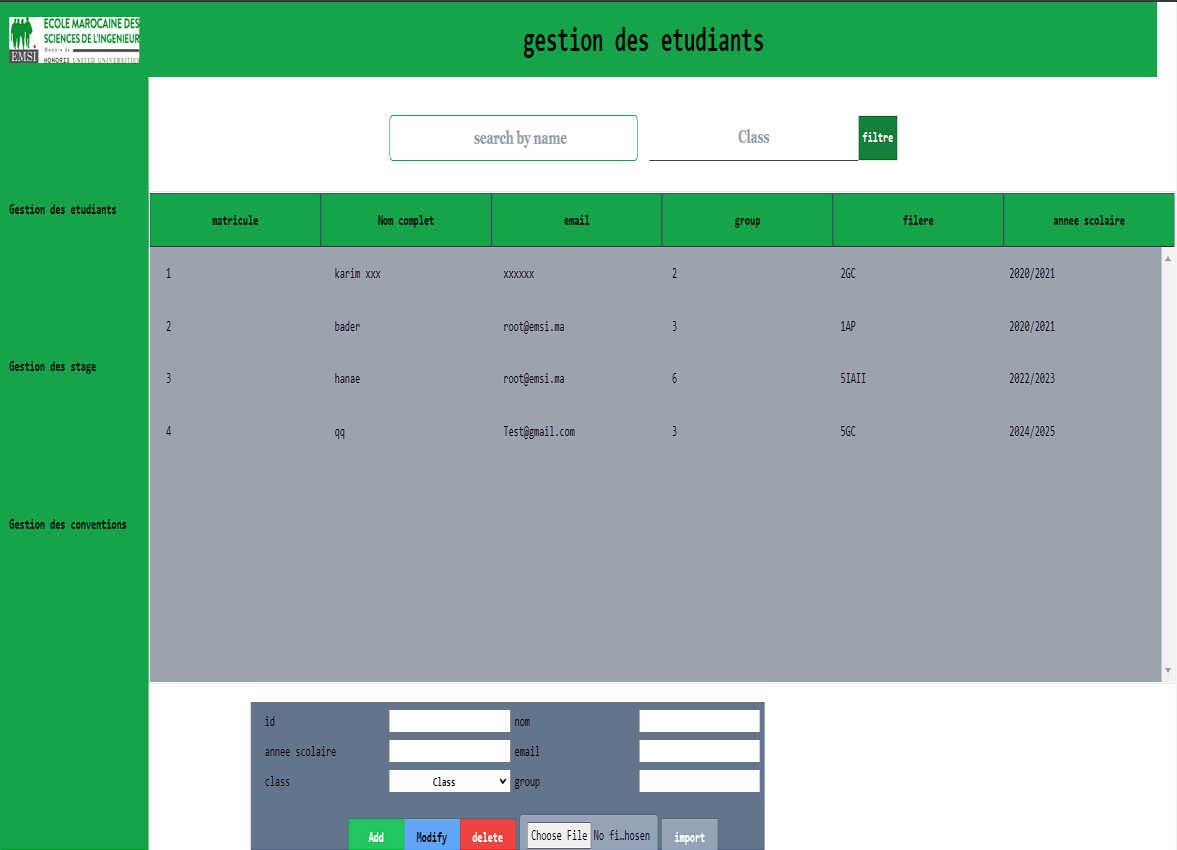
* **Patterns dans React avec TypeScript**
  1. Les interfaces de l’application
     1. Log in Page :



*Figure 17: Log in Page*

Le gérant doit d'abord s'authentifier en utilisant un compte offert par L'EMSI (email de la forme \*\*\*@emsi-edu.ma) et un mot de passe pour accéder au site.

#### Gestion des étudiants:



*Figure 18: Page des étudiants*

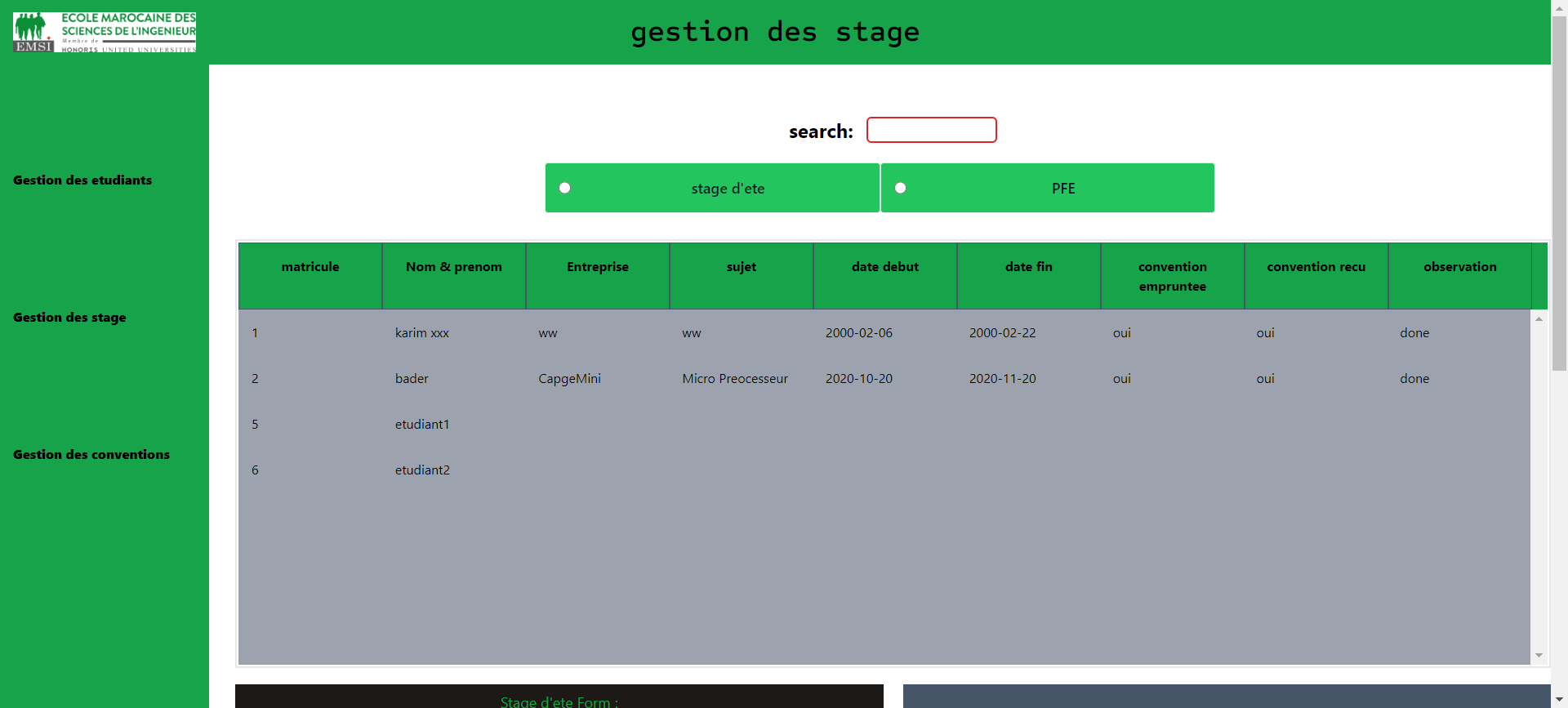
Sur cette page, le gestionnaire a la possibilité de consulter la liste des étudiants dans le système. Il peut rechercher un étudiant par son nom ou filtrer la liste des étudiants par filière.

De plus, il a la possibilité de créer un étudiant de deux manières différentes : soit en remplissant le formulaire contenant les informations requises, soit en important un fichier Excel contenant une liste d'étudiants.

Et il a l’accès ainsi de modifier les informations d’étudiant ou le supprimer.

#### Gestion des stages :

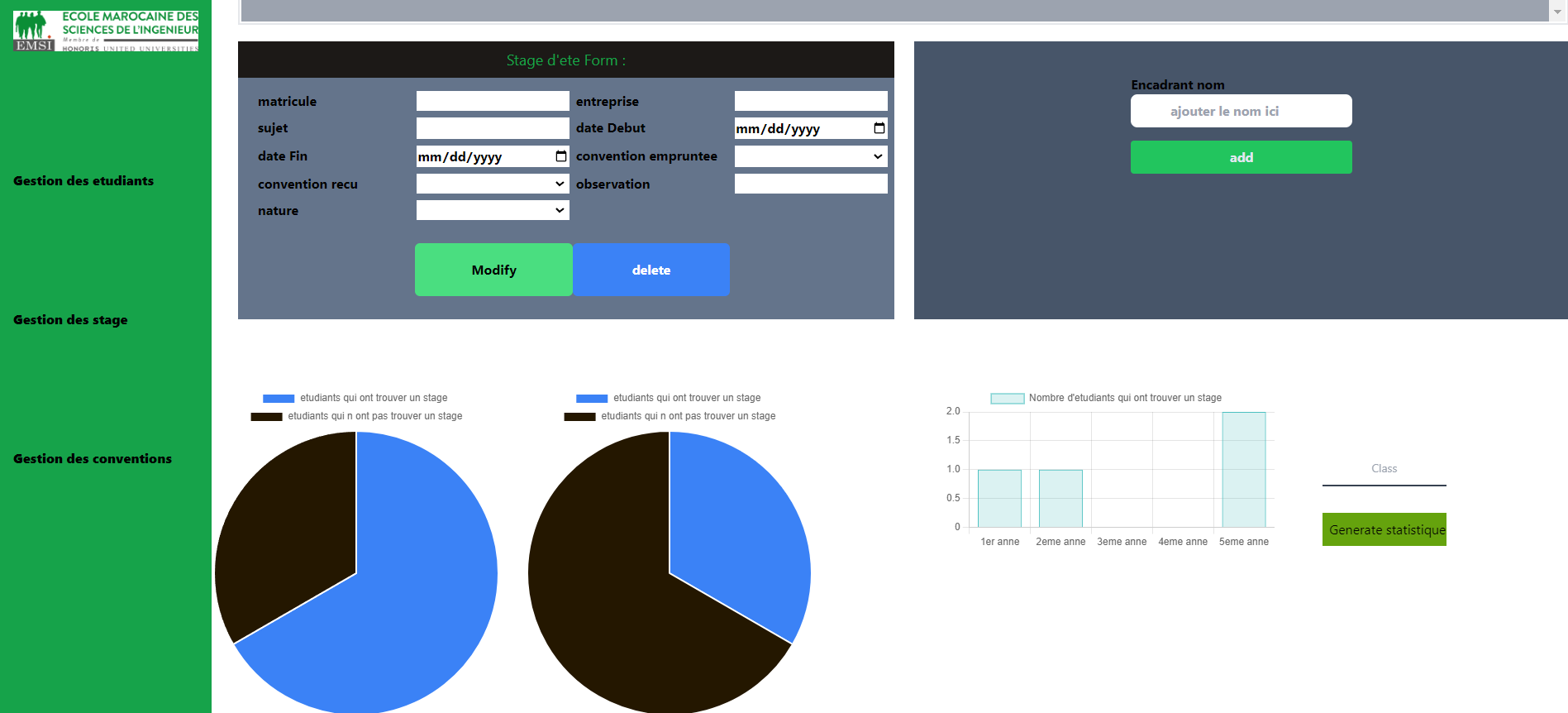
Interface Stage d’été



*Figure 19: Page des stages d'été 1*

En accédant cette page, le responsable a la possibilité de sélectionner la liste de stagiaires à traiter (Stage PFE ou stage d'été). Il peut alors effectuer une recherche de stage par nom.

Les stages d'été sont présentés par défaut.

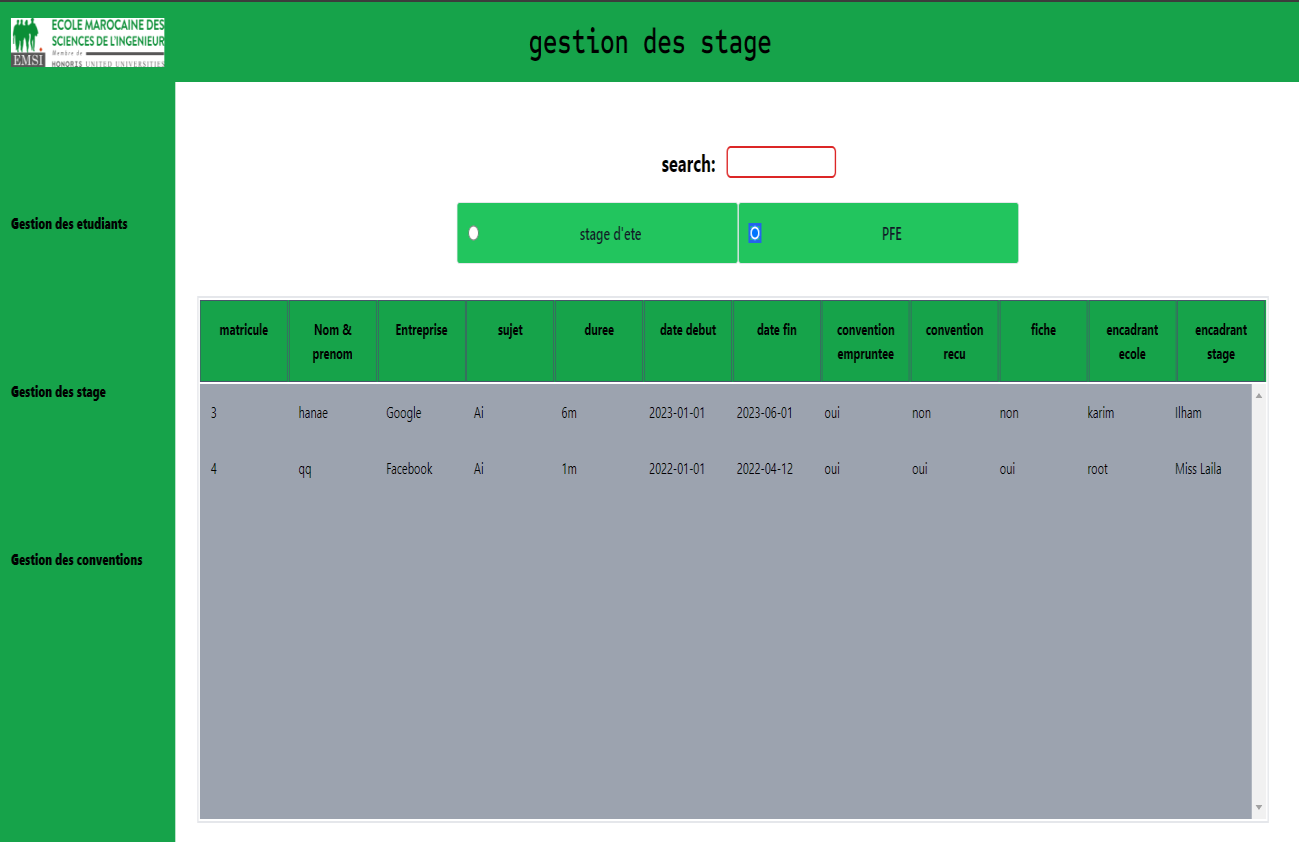


*Figure 20: Page des stages d'été 2*

Au-dessous on trouve 3 parties :

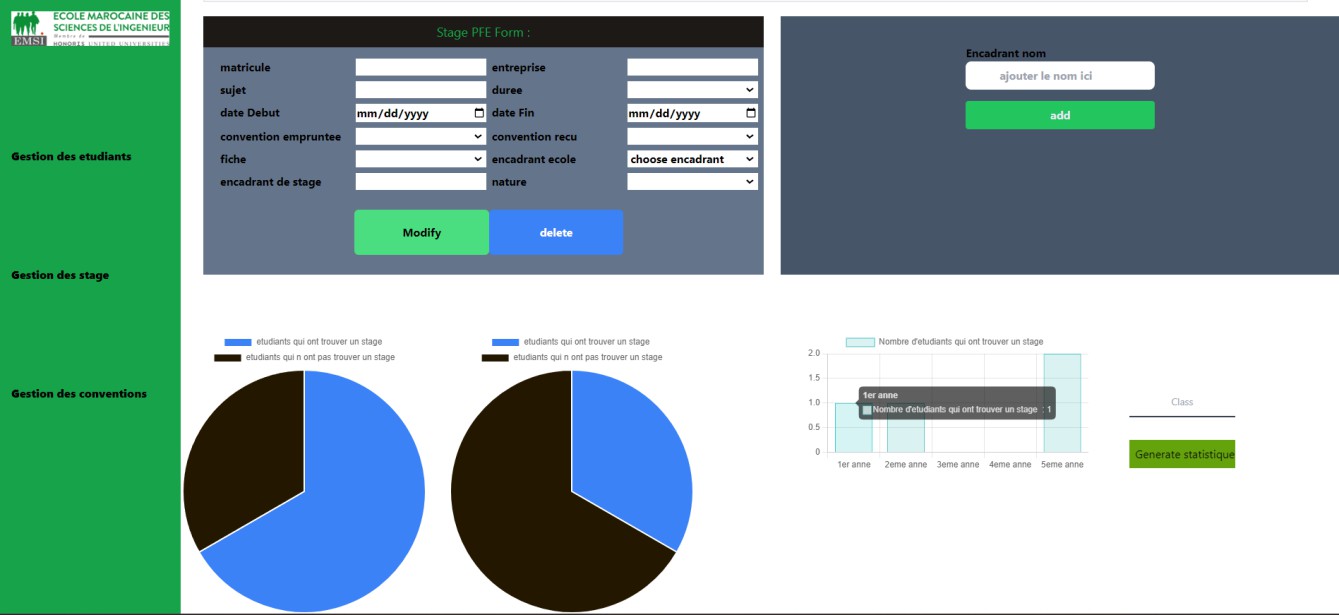
* Il y a tout d'abord un formulaire comprenant les renseignements relatifs au stage d'été, ainsi que deux boutons permettant de modifier les informations liées au stage ou de les supprimer.
* La deuxième partie offre la possibilité d’ajouter un encadrant a la liste des encadrant disponible
* Le troisième et la dernière concernent des graphs qui donnent une idée sur le nombre des étudiants qui ont réalisé des stages.

Interface de gestion de stage



*Figure 21: Page des stage PFE 1*

Dans cette page le gérant des filières peut visualiser la liste des étudiants qui ont réalisé un stage PFE « étudiants de 5eme année » après avoir choisir la liste des stages PFE.

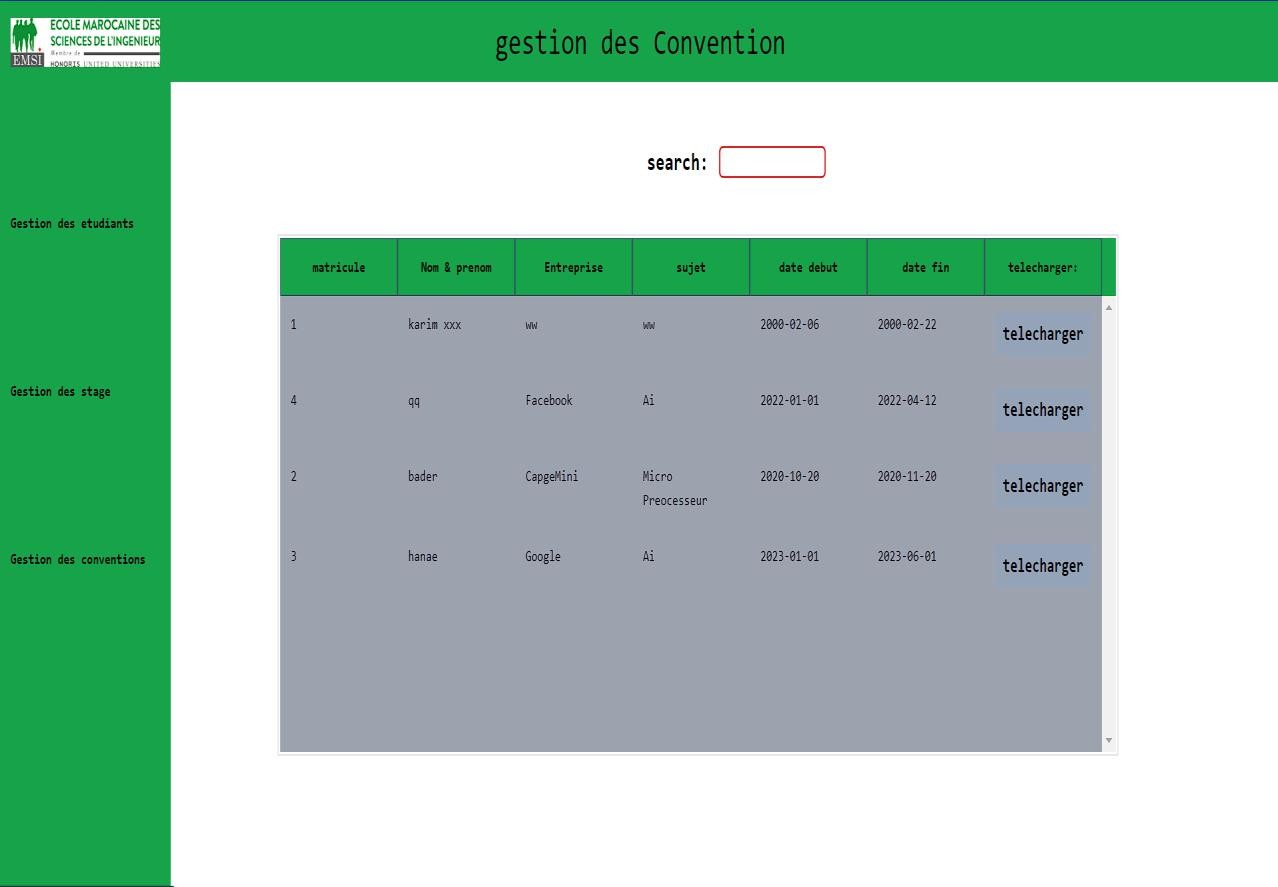


*Figure 22: Page des stage PFE 2*

Au-dessous on peut remarquer les mêmes trois parties avec une différence dans les champs à remplir (les champs des stages PFE sont différents

aux champs des stages d’été

#### Gestion de Conventions



*Figure 23: Page de téléchargement des conventions*

Dans cette page on a une liste de tous les stages avec un buton qui offre la possibilité de télécharger la convention correspondant au stage choisit.

La page offre ainsi la possibilité de chercher un étudiant par son nom

*Figure 24:Resultat de PDF (exemplaire)*

### Conclusion

Ce cadre matériel et logiciel a créé un environnement favorable à l'épanouissement individuel, facilitant la conception, la programmation et la mise en œuvre de l'application de gestion des étudiants et des stages, Et l’ensemble des interfaces réaliser dans le projet

**CHAPTRE 4 :**

## Conclusion générale

En conclusion, le développement de l'application de gestion des étudiants et des stages pour l'École Marocaine des Sciences de l'Ingénieur (EMSI) représente une étape significative dans l'amélioration des processus administratifs et de suivi au sein de l'institution. À travers ce projet, nous avons cherché à résoudre les inefficacités liées à la gestion manuelle des données, à la redondance des informations, et aux défis rencontrés dans le suivi des stages.

L'analyse approfondie de l'existant a permis de mettre en évidence les lacunes du système actuel, justifiant ainsi la nécessité d'une solution informatisée. La proposition d'une application web dédiée à la gestion des étudiants et des stages a été guidée par le souci d'automatisation, de centralisation des données, et d'amélioration de la transparence dans le processus.

La conception du projet, illustrée à travers les diagrammes UML, a fourni une base solide pour le développement. Les choix technologiques, du langage de programmation au système de gestion de base de données, ont été soigneusement pensés pour assurer la robustesse, la sécurité, et la convivialité de l'application.

En plus de son impact tangible sur l'amélioration des processus administratifs à l'EMSI, ce projet a été une véritable opportunité d'apprentissage pour moi sur le plan des compétences techniques et humaines. Sur le plan des compétences techniques, j'ai consolidé ma maîtrise des langages de programmation tels que PHP et JavaScript, ainsi que l'utilisation de technologies comme MySQL et Tailwind CSS.

La conception et la réalisation de l'application m'ont permis de renforcer mes compétences en développement web, de comprendre la logique derrière la modélisation UML, et d'acquérir une expérience pratique dans la résolution de problèmes liés à la gestion de données complexes.

En somme, ce projet a constitué une expérience riche en enseignements, me permettant d'acquérir et de perfectionner un éventail diversifié de compétences techniques et humaines, essentielles pour ma croissance professionnelle future.

## Références

<https://devdocs.io/javascript/> <https://www.chartjs.org/>

<https://github.com/PHPOffice/PhpSpreadsheet> <https://tailwindcss.com/docs> <https://www.php.net/docs.php>