****

Rapport du projet de web dynamique

**Encadré par:**

Mr. IDRAIS

**Réalisé par :**

Omar NASMI

Ilyas ATBIR

|  |
| --- |
| **Conception et Réalisation d’une application web de gestion des rendez-vous** |

**Membre(s) de jury:**

Mme. ANEJJAR

***Année* universitaire*:***

2024/2025

# Dédicace

C’est avec une grande fierté et une immense gratitude que nous adressons nos remerciements les plus sincères à notre encadrant, Monsieur Idrais, professeur du module Web Dynamique, pour son soutien précieux tout au long de notre projet intitulé "Health Bridge - Medical Appointment Management System". Ses orientations judicieuses, sa patience, et son accompagnement bienveillant ont été des piliers fondamentaux dans l’élaboration et la concrétisation de ce projet.

Nous tenons également à exprimer notre reconnaissance particulière à Madame Annejar et Monsieur Rida, dont l’encadrement et l’assistance, notamment durant les séances de travaux pratiques, ont été d’une aide inestimable. Leur pédagogie et leur disponibilité ont renforcé nos connaissances pratiques et permis d’approfondir nos compétences dans des domaines essentiels au projet.

Ce travail est dédié à nos familles et proches, dont le soutien indéfectible et les encouragements constants ont été une source de motivation et d’inspiration inestimable. À tous ceux qui, directement ou indirectement, ont contribué à la réussite de ce projet, nous exprimons notre gratitude infinie.

# Remerciement

Au terme de ce projet, nous souhaitons exprimer notre gratitude la plus sincère à Monsieur Idrais, notre encadrant pédagogique, pour son accompagnement constant et ses conseils éclairés. Son expertise, sa disponibilité, et ses orientations précieuses ont été d’une grande importance pour mener à bien ce travail.

Nous adressons également nos remerciements chaleureux à Madame Annejar et Monsieur Rida, pour leur assistance précieuse durant les séances de travaux pratiques. Leur engagement et leur pédagogie ont grandement contribué à notre maîtrise des outils et des concepts nécessaires à la réalisation de ce projet.

Nous tenons à remercier l’ensemble du corps enseignant du département d’ingénierie logicielle de la Faculté des Sciences d’Agadir pour la qualité de leur enseignement et leur soutien tout au long de notre parcours dans le cadre de la Licence d’Excellence en Ingénierie Logicielle.

Enfin, nos remerciements vont aux membres du jury pour le temps qu’ils consacrent à l’évaluation de ce projet et pour leurs remarques constructives, qui ne manqueront pas d’enrichir davantage notre travail.

À tous, nous exprimons notre gratitude la plus profonde.

# Table de matières

Contents

[Dédicace 2](#_Toc184237240)

[Remerciement 3](#_Toc184237241)

[Table de matières 4](#_Toc184237242)

[Table de Figures 6](#_Toc184237243)

[INTRODUCTION GENERALE 7](#_Toc184237244)

[CHAPITRE 1 : Spécification 8](#_Toc184237245)

[Cahier des charges 8](#_Toc184237246)

[**1. Contexte général :** 8](#_Toc184237247)

[**2. Problématique :** 8](#_Toc184237248)

[**3.** **Objectif :** 9](#_Toc184237249)

[**4.** **Acteurs :** 9](#_Toc184237250)

[**5.** **Etude des besoins :** 10](#_Toc184237251)

[Méthodologie suivie: 11](#_Toc184237252)

[**1.** **Logiciels et outils:** 11](#_Toc184237253)

[**2.** **Langages de programmation:** 12](#_Toc184237254)

[CHAPITRE 2 : ANALYSE ET CONCEPTION 13](#_Toc184237255)

[Introduction 13](#_Toc184237256)

[1. Conception générale : 13](#_Toc184237257)

[**1.1** **Cycle de vie :** 13](#_Toc184237258)

[1.**2** **Méthodologie de conception :** 14](#_Toc184237259)

[2. Conception détaillée : 14](#_Toc184237260)

[2.**1** **Diagramme de cas d’utilisation :** 14](#_Toc184237261)

[**2.** **Diagramme de classes :** 20](#_Toc184237262)

[Conclusion : 23](#_Toc184237263)

[CHAPITRE 3 : RÉALISATION ET MISE EN OEUVRE DU PROJET FINAL 23](#_Toc184237264)

[Introduction 23](#_Toc184237265)

[1. Les Interfaces: 24](#_Toc184237266)

[1. Landing Page: 24](#_Toc184237267)

[2. Sign Up: 24](#_Toc184237268)

[3. Account confirmation: 25](#_Toc184237269)

[4. Forget Password: 25](#_Toc184237270)

[5. Reset Password: 26](#_Toc184237271)

[6. Login: 26](#_Toc184237272)

[7. Profile: 27](#_Toc184237273)

[8. Side Bar: 27](#_Toc184237274)

[9. Assistants: 28](#_Toc184237275)

[10. Suppliers: 28](#_Toc184237276)

[11. Expenses: 29](#_Toc184237277)

[12. Employees: 29](#_Toc184237278)

[13. Cow: 30](#_Toc184237279)

[14. Stalls: 30](#_Toc184237280)

[Conclusion : 30](#_Toc184237281)

[Conclusion Générale 31](#_Toc184237282)

# Table de Figures

[Figure 1 : Visual Studio Code 11](#_Toc183798458)

[Figure 2 : XAMPP 11](#_Toc183798459)

[Figure 3 : MySQL 12](#_Toc183798460)

[Figure 4 : CodeIgniter 12](#_Toc183798461)

[Figure 5 : Bootstrap 12](#_Toc183798462)

[Figure 6 : Cycle de vie en V 13](#_Toc183798463)

[Figure 7 : Diagramme de cas d’utilisation primaire 15](#_Toc183798464)

[Figure 8 : Diagramme de classe 21](#_Toc183798465)

[Figure 9 : Landing Page 23](#_Toc183798466)

[Figure 10 : Sign Up 23](#_Toc183798467)

[Figure 11 : Account confirmation 24](#_Toc183798468)

[Figure 12 : Forget Password 24](#_Toc183798469)

[Figure 13 : Reset Password 25](#_Toc183798470)

[Figure 14 : Login 25](#_Toc183798471)

[Figure 15 : Profile 26](#_Toc183798472)

[Figure 16 : Side Bar 26](#_Toc183798473)

[Figure 17 : Assistants 27](#_Toc183798474)

[Figure 18 : Suppliers 27](#_Toc183798475)

[Figure 19 : Expenses 28](#_Toc183798476)

[Figure 20 : Employees 28](#_Toc183798477)

[Figure 21 : Cow 29](#_Toc183798478)

[Figure 22 : Stalls 29](#_Toc183798479)

# INTRODUCTION GENERALE

Le présent rapport met en lumière le développement de notre projet de fin d’études, intitulé Health Bridge - Medical Appointment Management System, une application web conçue pour simplifier et optimiser la gestion des rendez-vous médicaux. Ce projet, réalisé à l’aide du framework CodeIgniter, du système de gestion de bases de données MySQL, et du framework CSS Bootstrap, vise à offrir une plateforme conviviale et fonctionnelle répondant aux besoins des patients, des médecins, et des administrateurs.

Le domaine de la santé connaît une transformation numérique rapide, et la gestion des rendez-vous représente un défi majeur pour de nombreuses structures médicales. Les problèmes liés à la coordination des horaires, à l’efficacité du traitement des données, et à la communication entre les parties prenantes soulignent l’importance d’une solution centralisée. Health Bridge ambitionne de relever ces défis en proposant un outil intégré permettant de gérer les consultations, les disponibilités, et les préférences des utilisateurs avec simplicité et efficacité.

Ce projet s’articule autour de deux étapes principales : une phase d’analyse des besoins et une phase de développement technique. L’analyse des besoins a permis d’identifier les fonctionnalités indispensables, telles que la gestion des profils utilisateurs (patients, médecins, et administrateurs), la planification des rendez-vous, et la gestion des messages et des préférences utilisateurs. La phase de développement a consisté à transformer ces besoins en fonctionnalités concrètes et à implémenter une interface intuitive et accessible.

Ce rapport se structure en trois chapitres :

Le premier chapitre présente le cadre général du projet, y compris les objectifs, le cahier des charges, et les outils technologiques sélectionnés.

Le deuxième chapitre se consacre à la conception, avec des représentations UML telles que les diagrammes de cas d’utilisation, de classes, et de séquence.

Le troisième chapitre aborde la réalisation du projet, en décrivant les principales étapes de développement, les défis rencontrés, les solutions apportées, et une démonstration des interfaces clés de l’application.

L’objectif de ce document est de fournir une vue d’ensemble du processus de création de Health Bridge, tout en mettant en évidence les aspects techniques, les méthodologies adoptées, et l’impact attendu de cette solution sur la gestion des rendez-vous médicaux. Ce projet illustre notre volonté de proposer un outil moderne et performant pour répondre aux besoins croissants des structures médicales en matière de digitalisation.

Nous espérons que ce rapport saura refléter le travail accompli et constituer une ressource précieuse pour les lecteurs intéressés par les solutions numériques innovantes dans le domaine médical.

# CHAPITRE 1 : Spécification

## Cahier des charges

### **1. Contexte général :**

Le projet Health Bridge - Medical Appointment Management System vise à développer une application web intuitive et efficace pour la gestion des rendez-vous médicaux. Cette solution centralise les opérations liées à la planification des consultations, la gestion des profils utilisateurs (patients, médecins, et administrateurs), et la communication via un système de messagerie intégré. Le projet intègre également des outils avancés pour personnaliser l'expérience utilisateur et optimiser les interactions entre les différents acteurs du domaine médical.

### **2. Problématique :**

Ce projet se propose de répondre aux problématiques suivantes :

2.1 Interface utilisateur et administrative

* Convivialité de l'interface : Comment concevoir une interface intuitive permettant aux utilisateurs de naviguer facilement entre les fonctionnalités, tout en assurant une expérience utilisateur fluide ?
* Gestion du contenu : Comment mettre en place des opérations CRUD (Création, Lecture, Mise à jour, Suppression) efficaces et sécurisées pour la gestion des utilisateurs, des rendez-vous, et des messages ?

2.2 Fonctionnalités spécifiques

* Administration avancée : Quelles options devraient être intégrées pour permettre une gestion simplifiée mais exhaustive des rôles des utilisateurs, des horaires des médecins, et des notifications ?
* Performance et évolutivité : Quels outils technologiques et pratiques garantiraient une application performante, capable de s’adapter aux besoins futurs d’un environnement médical en constante évolution ?

2.3 Sécurité et confidentialité

* Protection des données : Comment assurer un niveau de sécurité élevé pour protéger les données personnelles et médicales des utilisateurs contre tout accès non autorisé ?
* Contrôle des accès : Quels systèmes de permissions doivent être mis en place pour limiter l’accès aux fonctionnalités selon le rôle de chaque utilisateur et garantir une utilisation responsable ?

### **3.** **Objectif :**

L'objectif principal est de développer une plateforme web robuste et sécurisée, offrant :

* **Gestion centralisée des rendez-vous médicaux :** Permettre la gestion efficace des profils patients, médecins, et administrateurs, ainsi que des rendez-vous, en facilitant la prise, la modification et l’annulation des consultations.
* **Outils de suivi avancés :** Implémenter des fonctionnalités permettant de suivre l’historique médical des patients, leurs préférences de consultation, ainsi que les disponibilités des médecins, afin de garantir une gestion fluide et optimisée.
* **Interface utilisateur intuitive :** Offrir une interface moderne et ergonomique permettant une navigation simplifiée pour tous les types d’utilisateurs (patients, médecins, administrateurs), en veillant à la clarté des informations et à l’accessibilité des fonctionnalités.
* **Sécurisation des données:** Mettre en place des mécanismes de sécurité robustes pour protéger les informations sensibles des utilisateurs et garantir l’intégrité des données, tout en offrant une gestion fine des permissions d’accès.

### **4.** **Acteurs :**

* **Administrateurs :**

Les administrateurs gèrent les comptes des patients, configurent les paramètres du système, et supervisent les activités. Ils ont un accès complet aux fonctionnalités, y compris la gestion des rendez-vous et la consultation des messages des patients.

* **Patients :**

Les patients gèrent leurs rendez-vous, consultent leur historique médical, mettent à jour leurs informations, et personnalisent leurs préférences de notification. Ils peuvent également communiquer avec les administrateurs pour toute question.

### **5.** **Etude des besoins :**

#### 5.1 Besoins fonctionnels :

* **Gestion des utilisateurs :** Créer, modifier, suspendre ou supprimer des comptes patients et administrateurs.
* **Gestion des rendez-vous :** Prendre, modifier ou annuler des rendez-vous médicaux pour les patients.
* **Gestion des profils :** Mise à jour des informations personnelles des patients (adresse, téléphone, préférences).
* **Gestion des messages :** Permettre aux patients de poser des questions aux administrateurs et recevoir des réponses.
* **Gestion des paramètres :** Permettre aux patients de personnaliser leurs préférences (notifications, langue, thème).
* **Suivi des rendez-vous :** Suivi des rendez-vous passés et à venir pour chaque patient.

#### 5.2 Besoins non fonctionnels :

* **Fiabilité :** L'application doit fonctionner de manière fiable et sans erreurs, assurant une expérience utilisateur stable et satisfaisante.
* **Gestion des erreurs :** Les messages d'erreur doivent être clairs et informatifs, guidant l'utilisateur en cas d'ambiguïté et facilitant la résolution des problèmes.
* **Ergonomie et interface utilisateur :** L'interface utilisateur doit être intuitive, avec une navigation claire entre les différentes sections, des couleurs attrayantes et une mise en page conviviale.
* **Facilité d'utilisation :** L'application doit être adaptée à l'utilisateur, nécessitant un effort minimal pour la navigation et l'exécution des différentes tâches.
* **Aptitude à la maintenance et à la réutilisation :** Le système doit suivre une architecture claire et standard, permettant une maintenance facile et la réutilisation de composants pour de futures améliorations.
* **Performances :** Une réponse rapide de l'application, minimisant les temps de chargement et assurant une expérience utilisateur fluide.
* **Sécurité et confidentialité des données :** Assurer la confidentialité et la sécurité des données des utilisateurs, avec des mécanismes d'authentification et des autorisations d'accès appropriées.
* **Synchronisation des données :** Assurer une synchronisation efficace entre l'interface utilisateur et la base de données pour garantir la cohérence des informations.

## Méthodologie suivie:

### **1.** **Logiciels et outils:**

1. **Visual Studio Code** : Visual Studio est un environnement de développement intégré (IDE) polyvalent qui permet aux développeurs de créer des applications et des logiciels pour différentes plateformes et langages. Il fournit des outils de codage, de débogage et de déploiement pour faciliter le développement d'applications web, de bureau et mobiles. Sa prise en charge étendue de plusieurs langages de programmation et ses fonctionnalités avancées en font un choix populaire parmi les développeurs pour la réalisation de projets complexes.

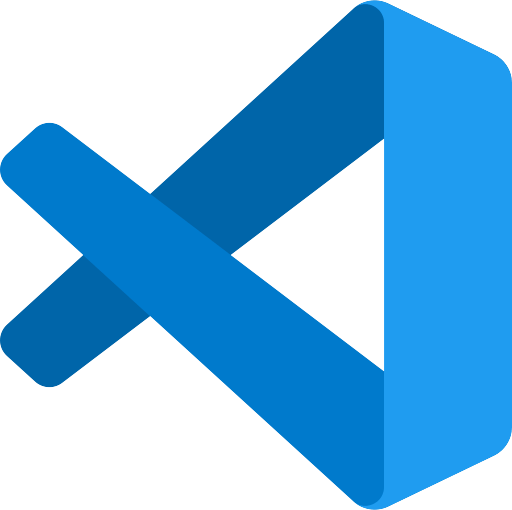


Figure 1 : Visual Studio Code

1. **Xampp:** XAMPP est un ensemble gratuit de programmes open-source, qui comprend Apache,MariaDB, PHP et Perl. Il permet aux développeurs de travailler sur un serveur Web local en testant leur code PHP et leur base de données MySQL. XAMPP est facile à installer et à utiliser, ce qui en fait un choix populaire pour les développeurs débutants. Cependant, il est important de noter que XAMPP n'est pas recommandé pour un environnement de production en raison de la faible sécurité par défaut.



Figure 2 : XAMPP

### **2.** **Langages de programmation:**

**MySQL :** Système de gestion de base de données relationnelle open-source, largement utilisé pour le stockage et la gestion des données. Offre une robuste prise en charge SQL et est compatible avec divers langages de programmation, y compris Java.



Figure 3 : MySQL

**CodeIgniter :** CodeIgniter est un framework PHP open-source léger et puissant, idéal pour le développement d'applications web robustes et performantes. Il offre une structure organisée, une sécurité renforcée, et des outils intégrés pour faciliter le développement. Grâce à CodeIgniter, notre projet bénéficie d'une organisation du code optimisée, d'une meilleure évolutivité et d'une base fiable pour les fonctionnalités de notre application.



Figure 4 : CodeIgniter

**Bootstrap :** est une bibliothèque CSS open-source populaire qui facilite la création d'interfaces utilisateur réactives et élégantes. Elle fournit une large gamme de composants prêts à l'emploi, de classes utilitaires, et d'outils de conception pour accélérer le développement front-end. L'utilisation de Bootstrap dans notre projet améliore considérablement la rapidité de développement et garantit une interface utilisateur moderne et attrayante.



Figure 5 : Bootstrap

# CHAPITRE 2 : ANALYSE ET CONCEPTION

## Introduction

Dans le développement de notre Dairy Farm, la phase de conception joue un rôle clé dans la réalisation d'une application performante et fiable. C’est durant cette étape que nous définissons d’abord une vue globale, en décrivant l’architecture générale qui guidera la réalisation du projet. Ensuite, nous approfondirons nos choix conceptuels à travers divers types de diagrammes.

## 1. Conception générale :

### **1.1** **Cycle de vie :**

Le cycle de vie d’une application englobe toutes les étapes, depuis la conception et la réalisation jusqu’à la mise en œuvre. Pour le développement de notre système de gestion de ferme, nous avons choisi d’adopter le modèle de cycle de vie en V. Ce modèle est reconnu pour son efficacité, grâce à son approche rigoureuse qui inclut : Chaque phase est contrôlée avant de passer à la suivante, minimisant ainsi les erreurs. Permet de détecter et corriger les problèmes dès les premières étapes, réduisant ainsi les coûts et les délais de développement.

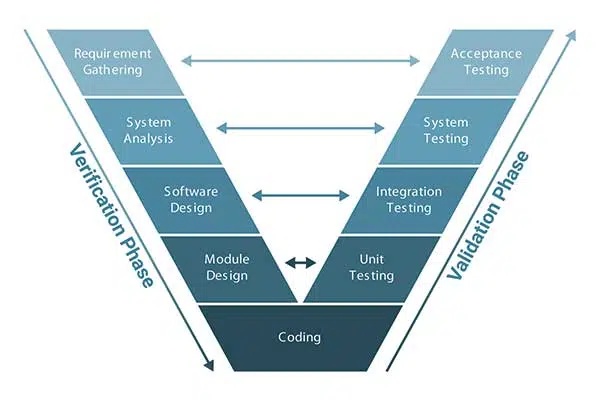


Figure 6 : Cycle de vie en V

### 1.**2** **Méthodologie de conception :**

Pour structurer et organiser notre conception, nous utilisons le Langage de Modélisation Unifié (UML). UML est une norme qui facilite la modélisation d'un problème de manière visuelle et standardisée. Ce langage, né de la fusion de plusieurs méthodes antérieures, est devenu incontournable dans la conception orientée objet. Les principaux avantages de l’UML pour notre projet incluent : Visualisation claire des composants et des interactions. Documentation standardisée facilitant la compréhension et la communication entre les membres de l’équipe. Modélisation structurée pour garantir une application robuste et évolutive. Grâce à cette méthodologie, nous assurons une conception bien structurée qui guidera efficacement le développement de notre application de gestion de ferme.

## 2. Conception détaillée :

### 2.**1** **Diagramme de cas d’utilisation :**

Les rôles des diagrammes de cas d’utilisation sont de recueillir, d’analyser et d’organiser les besoins, ainsi que de recenser les grandes fonctionnalités d’un système. Il s’agit donc de la première étape UML pour la conception d’un système.

Le diagramme de cas se compose de trois éléments principaux :

· **Un Acteur** : c’est l’idéalisation d’un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

· **Un cas d’utilisation :** c’est une unité cohérente représentant une fonctionnalité visible de s d’utilisation l’extérieur.

· **Les relations :** Trois types de relations sont pris en charge par la norme [UML](http://fr.wikipedia.org/wiki/Unified_modeling_language) et sont graphiquement représentées par des types particuliers de ces relations.

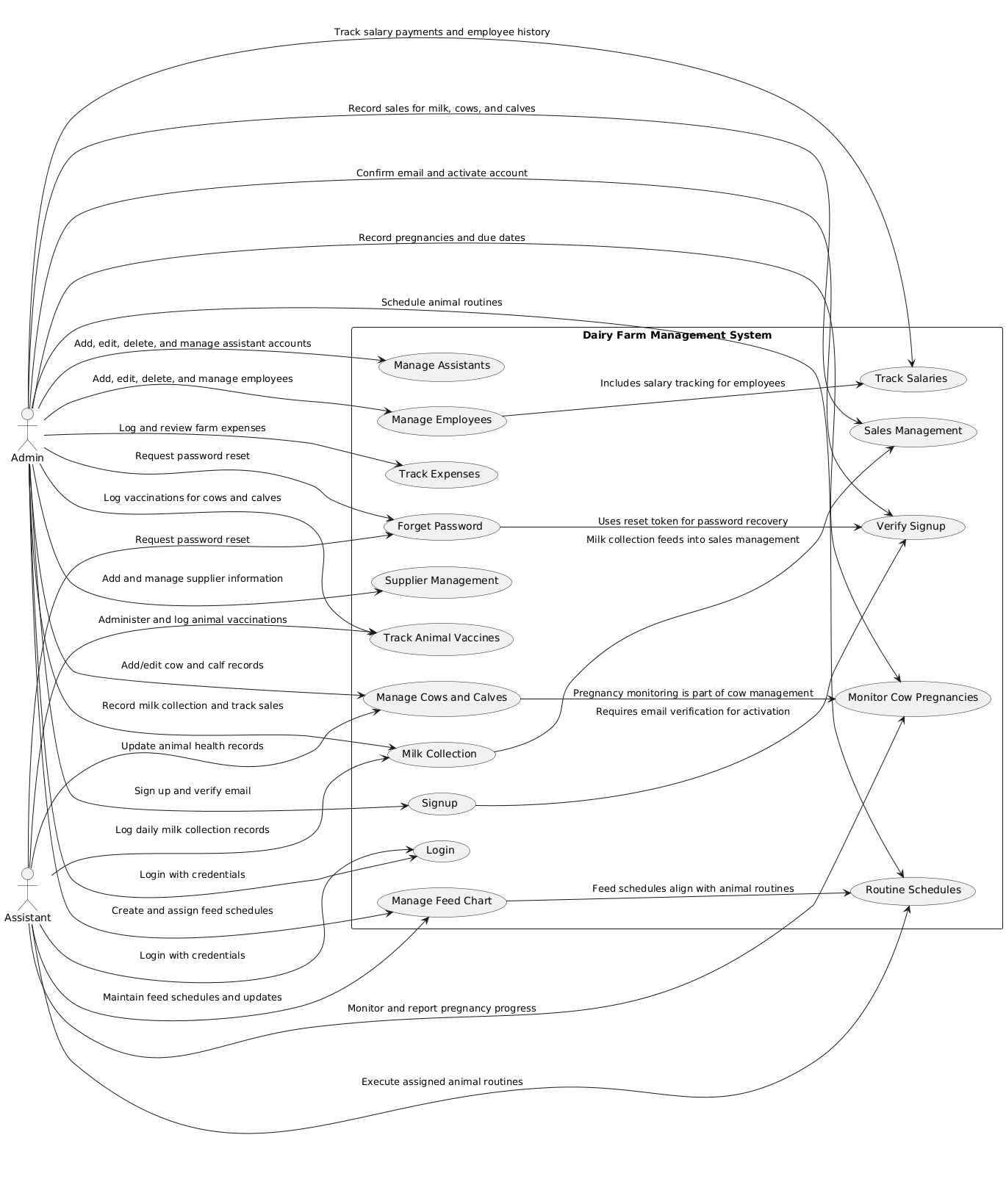
. 

Figure 7 : Diagramme de cas d’utilisation primaire

* **Accéder au profil** : L'utilisateur peut accéder à son profil pour voir et modifier ses informations personnelles, telles que son nom, son adresse e-mail et sa photo de profil.
* **Se connecter (Login) :** Les utilisateurs, qu'ils soient administrateurs ou assistants, peuvent se connecter au système en utilisant leurs identifiants (nom d'utilisateur et mot de passe). Cela permet d'accéder à leurs fonctions spécifiques en fonction de leur rôle.
* **S'inscrire (Signup) :** L'utilisateur peut créer un compte dans le système en fournissant des informations nécessaires. Un e-mail de confirmation est envoyé, et l'utilisateur doit vérifier son e-mail pour activer son compte.
* **Mot de passe oublié (Forget Password) :** Si un utilisateur oublie son mot de passe, il peut demander une réinitialisation. Un lien de réinitialisation est envoyé à son adresse e-mail pour lui permettre de choisir un nouveau mot de passe.
* **Vérifier l'inscription (Verify Signup) :** Cette fonction est activée lorsqu'un utilisateur confirme son inscription en vérifiant son adresse e-mail. Cela permet de garantir l'authenticité des informations fournies.

**Cas d’utilisation pour les Administrateurs :**

* **Gérer les assistants (Manage Assistants) :** L'administrateur peut ajouter, modifier ou supprimer des comptes d'assistants, leur attribuant ainsi des responsabilités spécifiques au sein du système.
* **Gérer les employés (Manage Employees) :** L'administrateur peut gérer les informations des employés, telles que leurs profils, leurs rôles et leurs détails de contact.
* **Suivi des salaires (Track Salaries) :** L'administrateur suit les paiements des salaires des employés et consulte l’historique des paiements effectués.
* **Collecte de lait (Milk Collection) :** L'administrateur peut enregistrer les collectes de lait quotidiennes et suivre leur quantité. Cette information est liée à la gestion des ventes de lait.
* **Gérer le plan de nourrissage (Manage Feed Chart) :** L'administrateur est responsable de la création et de l'attribution des plannings de nourrissage pour les animaux, afin d'assurer une alimentation adéquate.
* **Suivi des vaccins des animaux (Track Animal Vaccines) :** L'administrateur logue les vaccinations des vaches et des veaux pour garantir leur bien-être et leur santé.
* **Gérer les vaches et les veaux (Manage Cows and Calves) :** L'administrateur peut ajouter et modifier les informations relatives aux vaches et aux veaux, comme leur santé, leur âge, et leur suivi.
* **Suivi des grossesses des vaches (Monitor Cow Pregnancies) :** L'administrateur suit l'état de grossesse des vaches, en enregistrant les dates d'accouchement prévues.
* **Suivi des dépenses (Track Expenses) :** Cette fonctionnalité permet à l'administrateur de suivre les dépenses de la ferme, incluant les achats d'équipement et autres frais.
* **Gestion des fournisseurs (Supplier Management) :** L'administrateur gère les informations des fournisseurs, comme les coordonnées et les produits fournis.
* **Gestion des ventes (Sales Management) :** L'administrateur enregistre les ventes de lait, de vaches, et de veaux. Les informations relatives aux ventes sont également reliées à la gestion des stocks et des finances.
* **Planification des routines (Routine Schedules) :** L'administrateur peut planifier les routines quotidiennes des animaux (nourrissage, nettoyage, etc.) et s'assurer qu'elles sont suivies correctement.

**Cas d’utilisation pour les Assistants :**

* **Se connecter (Login) :** Les assistants peuvent se connecter au système avec leurs identifiants pour accéder à leurs fonctionnalités spécifiques.
* **Mot de passe oublié (Forget Password) :** Si un assistant oublie son mot de passe, il peut demander une réinitialisation via un lien envoyé par e-mail.
* **Enregistrer la collecte de lait (Log daily milk collection records) :** Les assistants enregistrent quotidiennement la quantité de lait collectée.
* **Maintenir les plannings de nourrissage (Maintain feed schedules) :** Les assistants aident à maintenir à jour les plannings de nourrissage des animaux.
* **Administrer et enregistrer les vaccinations des animaux (Administer and log animal vaccinations) :** Les assistants participent à l'administration des vaccinations et à l'enregistrement de ces informations dans le système.
* **Mettre à jour les informations de santé des animaux (Update animal health records) :** Les assistants mettent à jour les dossiers de santé des animaux en fonction des observations quotidiennes.
* **Suivi des grossesses (Monitor and report pregnancy progress) :** Les assistants suivent les progrès des grossesses des vaches et signalent toute anomalie.
* **Exécuter les routines assignées (Execute assigned animal routines) :** Les assistants effectuent les tâches quotidiennes assignées pour garantir le bon fonctionnement des routines des animaux.

**Relations entre les cas d’utilisation :**

* **Vérification de l'inscription :** L'activation du compte nécessite une vérification de l'adresse e-mail.
* **Réinitialisation du mot de passe :** L'utilisateur peut récupérer son mot de passe en utilisant un lien de réinitialisation envoyé à son e-mail.
* **Suivi des salaires :** La gestion des salaires est incluse dans la gestion des employés.
* **Collecte de lait et ventes :** Les informations de collecte de lait alimentent la gestion des ventes de lait.
* **Gestion des vaches et suivi des grossesses :** Le suivi des grossesses fait partie de la gestion des vaches, permettant une gestion complète de leur état de santé.
* **Planification des routines et gestion de nourrissage :** Les plannings de nourrissage sont associés aux routines animales pour une gestion synchronisée.

### **2.** **Diagramme de classes :**

Un diagramme de classes est un type de diagramme de modélisation utilisé en génie logiciel pour représenter la structure d'un système en termes de ses classes et des relations entre elles. Il fait partie des diagrammes de modélisation orientée objet et est utilisé principalement pour concevoir la structure interne d'un logiciel. Chaque classe dans un diagramme de classes représente un concept, une entité ou un objet ayant des attributs (propriétés) et des méthodes (fonctions ou comportements).

Voici l'explication détaillée du diagramme de classes que vous avez partagé pour un système de gestion de ferme laitière :

**Classes principales :**

* **User (Administrateur) :** Représente un utilisateur du système (par exemple, un administrateur). Il contient des informations comme l'identifiant, le nom d'utilisateur, l'email, le mot de passe, la photo, le rôle (administrateur, assistant, etc.), et d'autres attributs liés à la gestion du compte utilisateur.
* **Assistant :** Une classe représentant un assistant dans le système. Il hérite des attributs de User, mais a aussi des informations spécifiques comme qui l'a créé et des informations de gestion du mot de passe.
* **Employee (Employé) :** Représente un employé travaillant à la ferme. Il possède des informations comme le nom, le poste, le salaire et la date d'embauche.
* **Cow (Vache) et Calf (Veau) :** Ces classes représentent les animaux de la ferme, leurs informations telles que le numéro d'identification, leur état de santé, leur emplacement dans les stalles, et leur statut de vente.
* **MilkCollection (Collecte de lait) :** Cette classe gère les enregistrements de lait collecté, y compris la quantité et la date de collecte.
* **FeedChart (Plan de nourrissage) :** Représente les plannings de nourrissage pour les animaux.
* **AnimalVaccine (Vaccination des animaux) :** Représente les vaccins administrés aux animaux, y compris la date et les informations liées à la vaccination.
* **PregnancyRecord (Enregistrement des grossesses) :** Cette classe suit l'état de grossesse des vaches, incluant la date de début et la date prévue de livraison.
* **Expense (Dépense) :** Représente les dépenses de la ferme (comme les achats d'aliments pour animaux ou autres frais).
* **Sale (Vente) :** Représente les ventes des produits de la ferme (lait, vaches, etc.), y compris les informations sur le prix et l'acheteur.
* **Routine (Routine) :** Cette classe représente les tâches régulières à effectuer sur les animaux (par exemple, nourrissage, nettoyage des stalles).

**Relations entre les classes :**

* User crée des Assistants, Employés, enregistre des Collectes de lait, des FeedCharts, des Ventes, des Routines, des Enregistrements de grossesse, etc.
* Assistant gère et met à jour les informations relatives à la collecte de lait, les plans de nourrissage, et les routines des animaux, tout en surveillant les grossesses.
* Employee est payé, ce qui lie la classe Employee à EmployeeSalary (enregistrements de salaires).
* Cow produit des Calfs et des MilkCollection. Chaque vache peut avoir plusieurs enregistrements de grossesse et peut être associée à des ventes.
* Stall héberge des Cows et des Calfs, avec une capacité donnée pour chaque stalle.
* Calf suit des FeedCharts et des Routines.
* AnimalVaccine est associé aux Cows et Calfs, représentant les vaccins administrés à ces animaux.

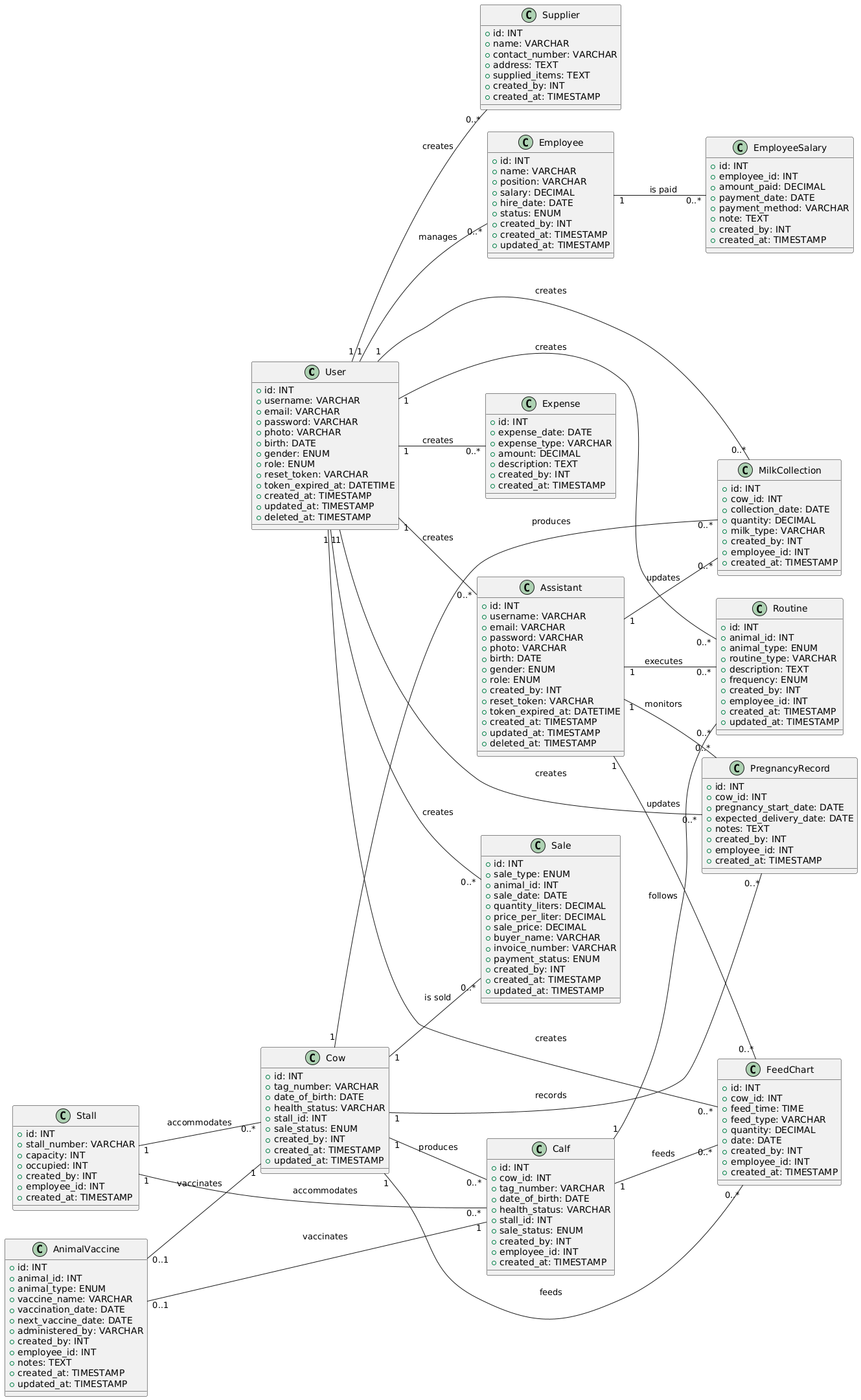


Figure 8 : Diagramme de classe

## Conclusion :

La phase conceptuelle est une étape fondamentale pour la réalisation de n’importe quel projet. Elle permet de faciliter le système d’information et réaliser l’implémentation de la base de données et le traitement. Par la suite, nous devons chercher les moyens et les outils possibles pour développer l’application, ce que nous allons présenter dans le chapitre suivant.

# CHAPITRE 3 : RÉALISATION ET MISE EN OEUVRE DU PROJET FINAL

## Introduction

L'objectif principal de ce chapitre est de vous montrer ce à quoi ressemble le projet terminé, pour que vous puissiez mieux comprendre comment avance l'application web. Nous allons vous montrer quelques exemples de pages web pour que vous puissiez voir concrètement ce sur quoi nous avons travaillé jusqu'à présent

## 1. Les Interfaces:

### 1. Landing Page:

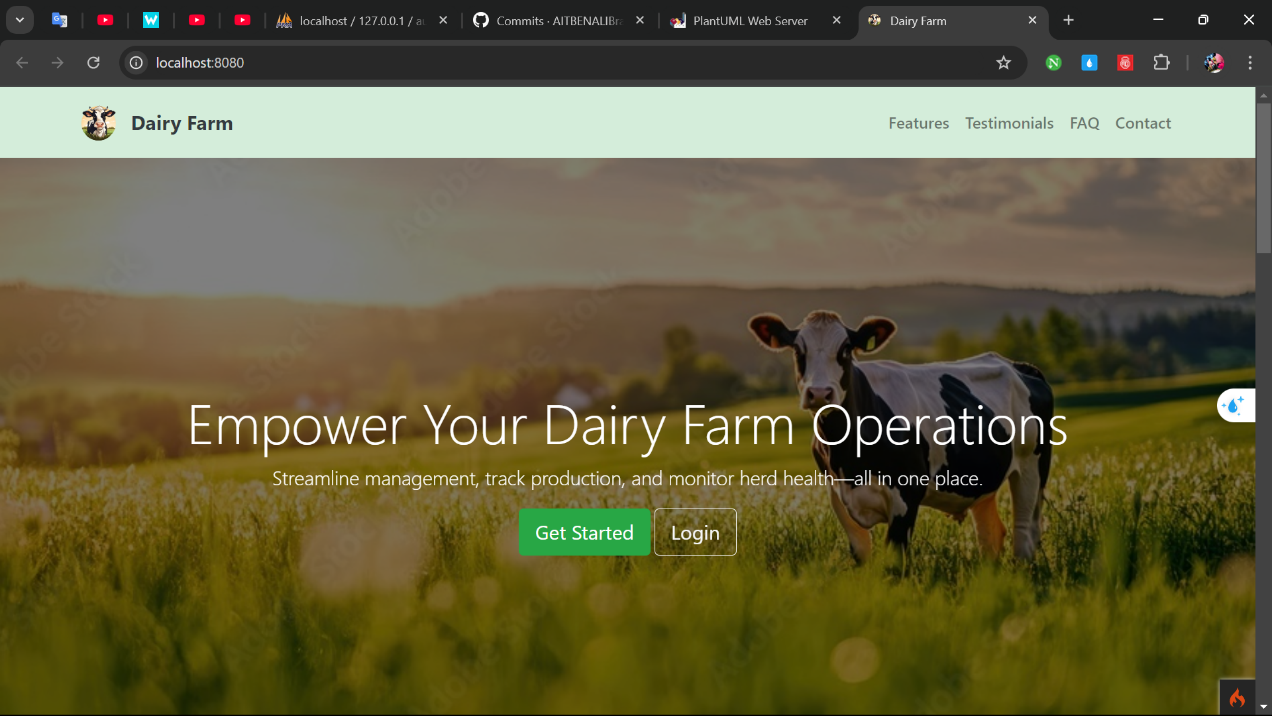


Figure 9 : Landing Page

### 2. Sign Up:

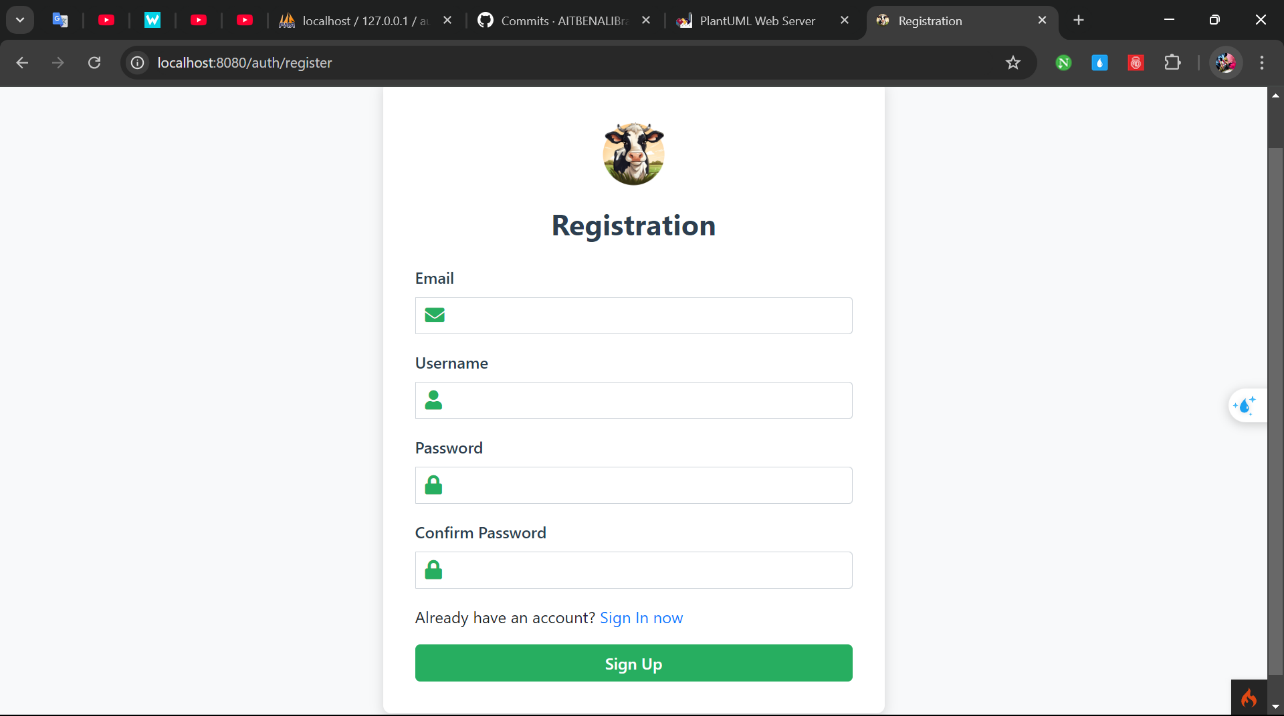


Figure 10 : Sign Up

### 3. Account confirmation:

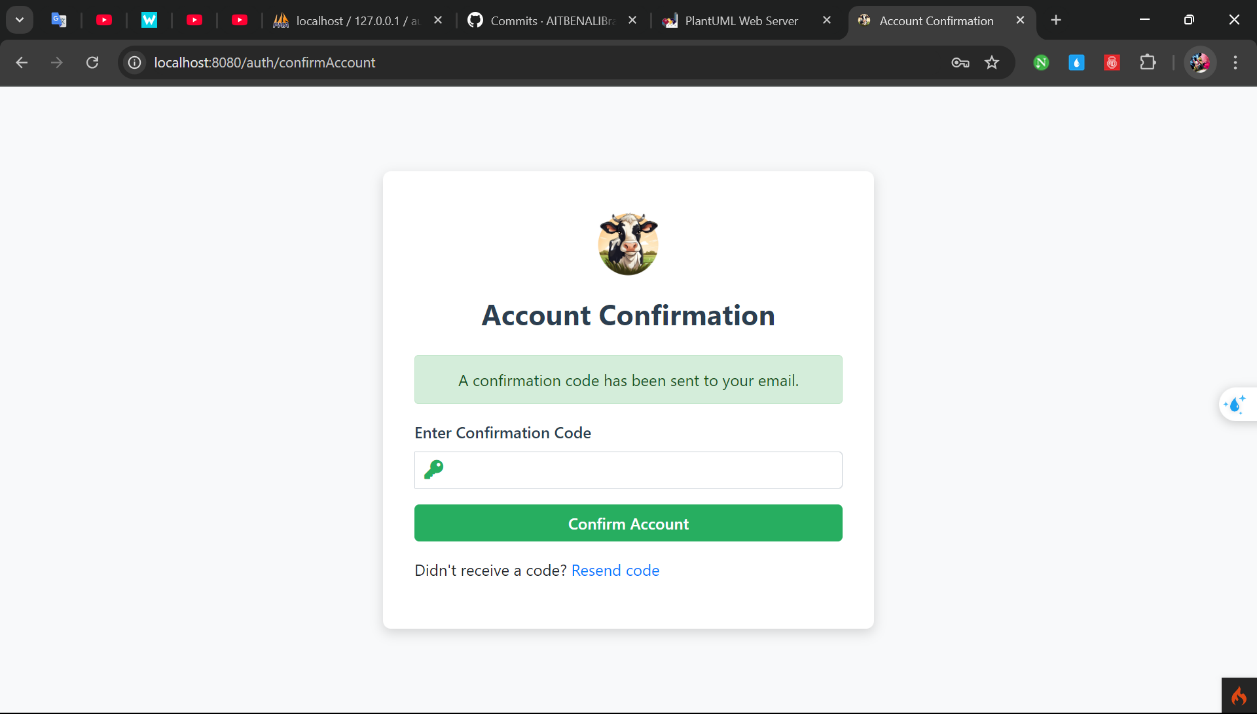


Figure 11 : Account confirmation

### 4. Forget Password:

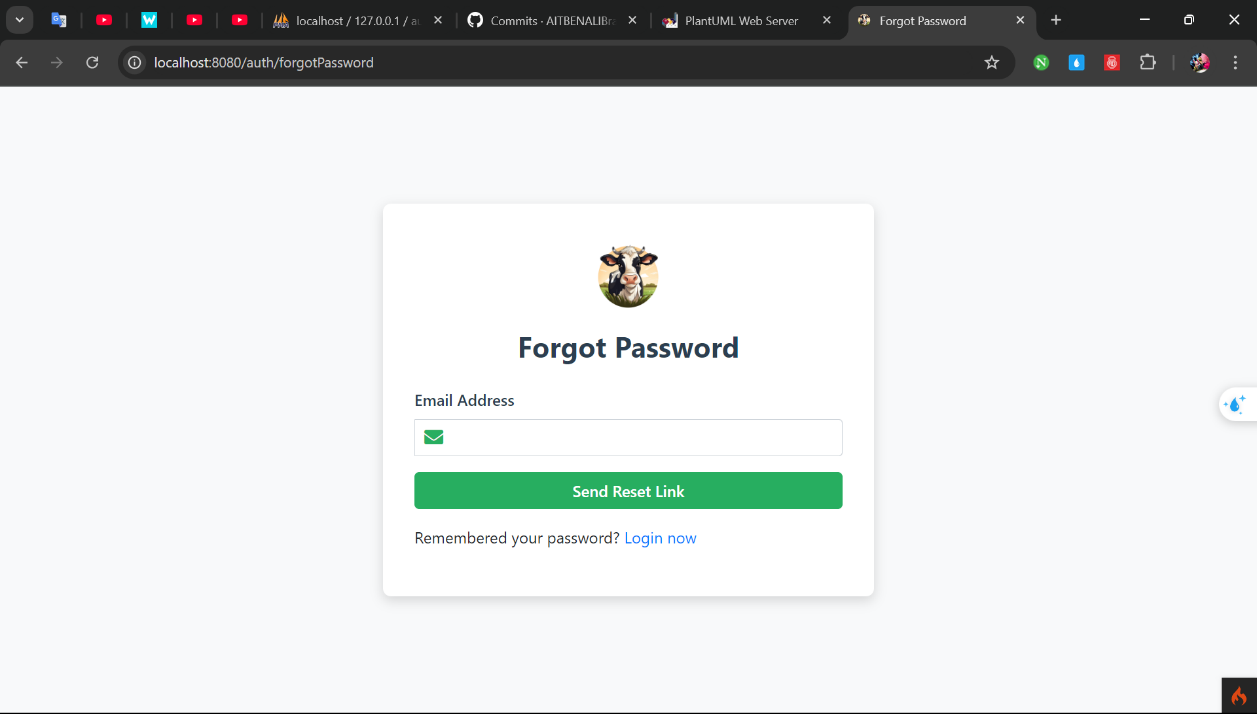


Figure 12 : Forget Password

### 5. Reset Password:

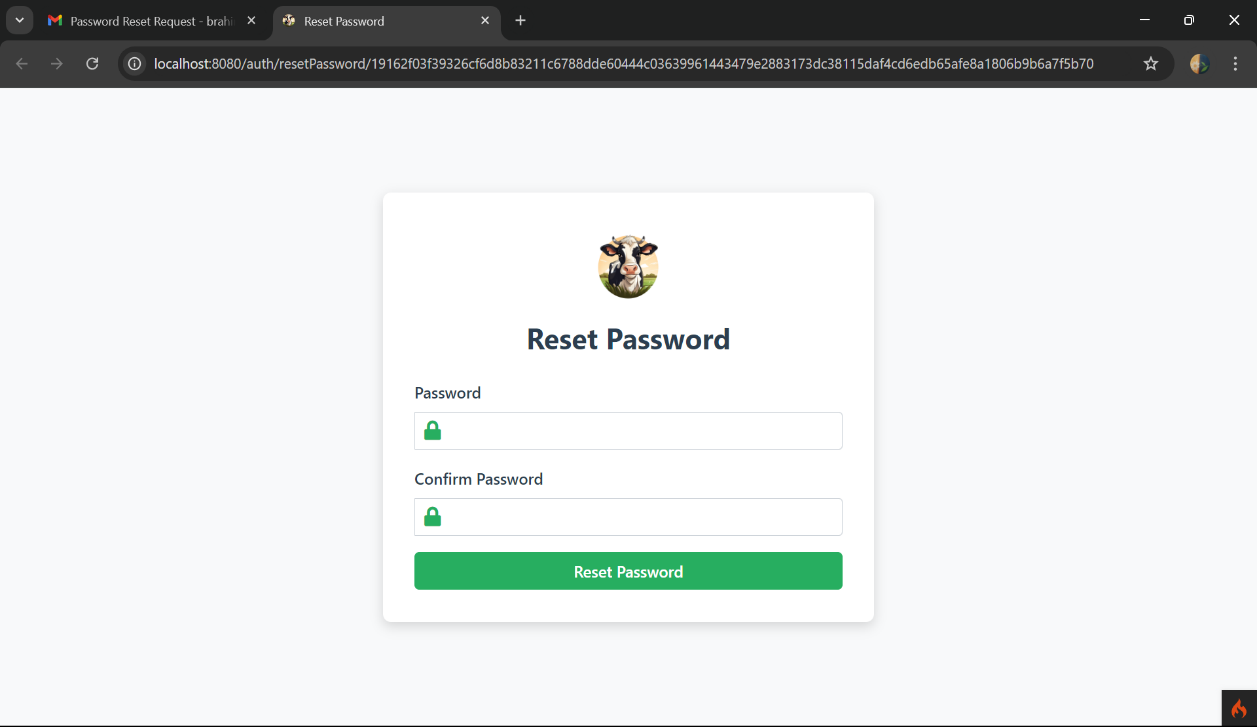


Figure 13 : Reset Password

### 6. Login:



Figure 14 : Login

### 7. Profile:

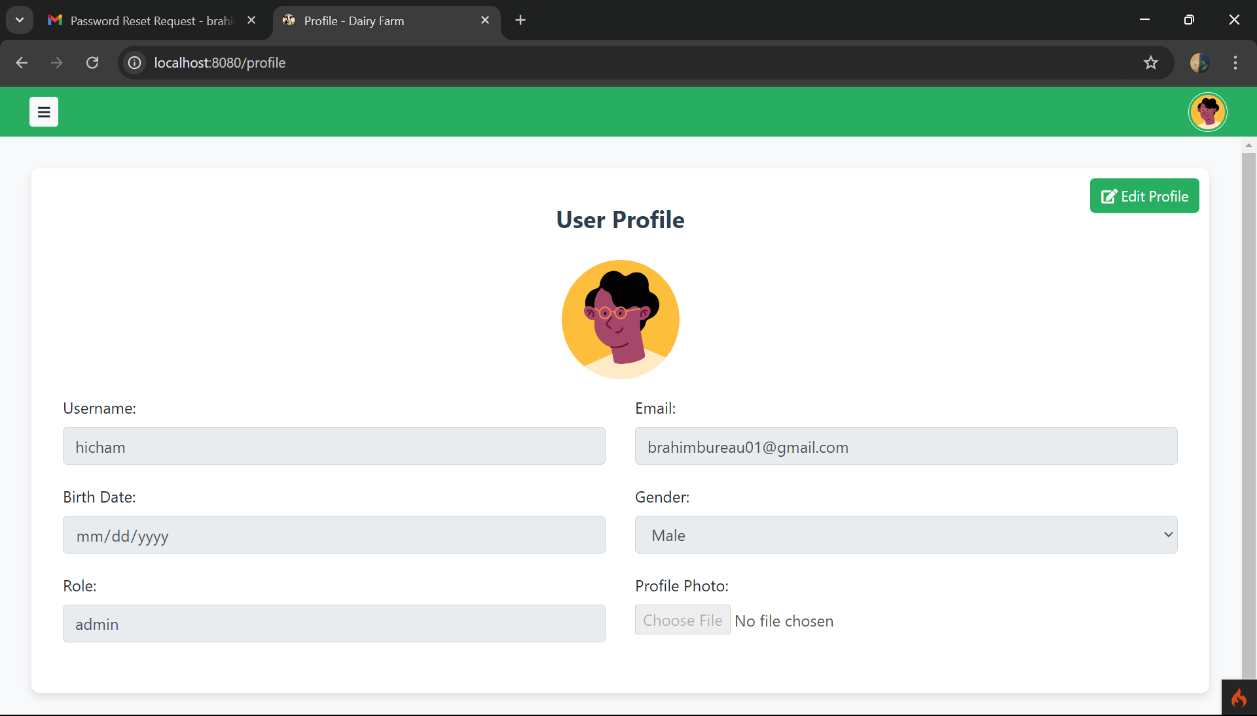


Figure 15 : Profile

### 8. Side Bar:

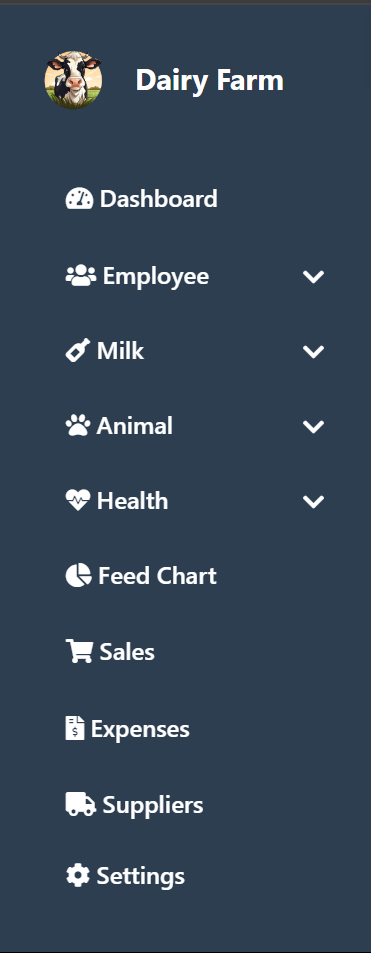


Figure 16 : Side Bar

### 9. Assistants:

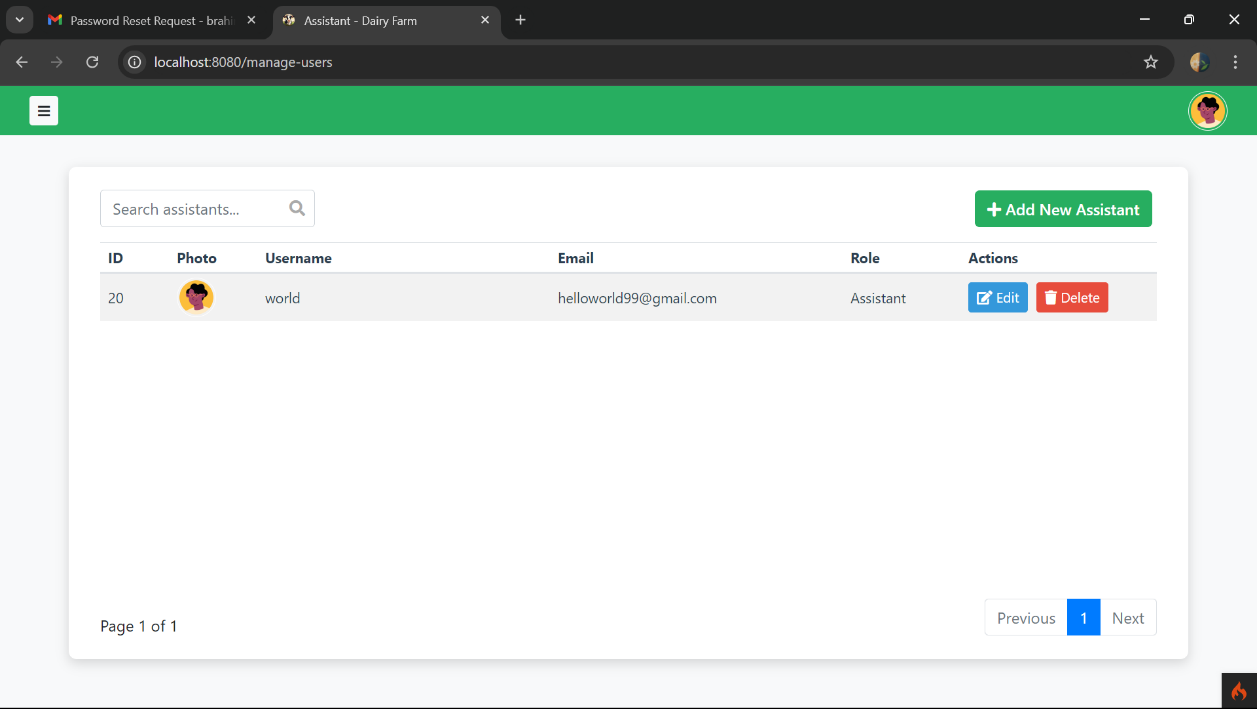


Figure 17 : Assistants

### 10. Suppliers:

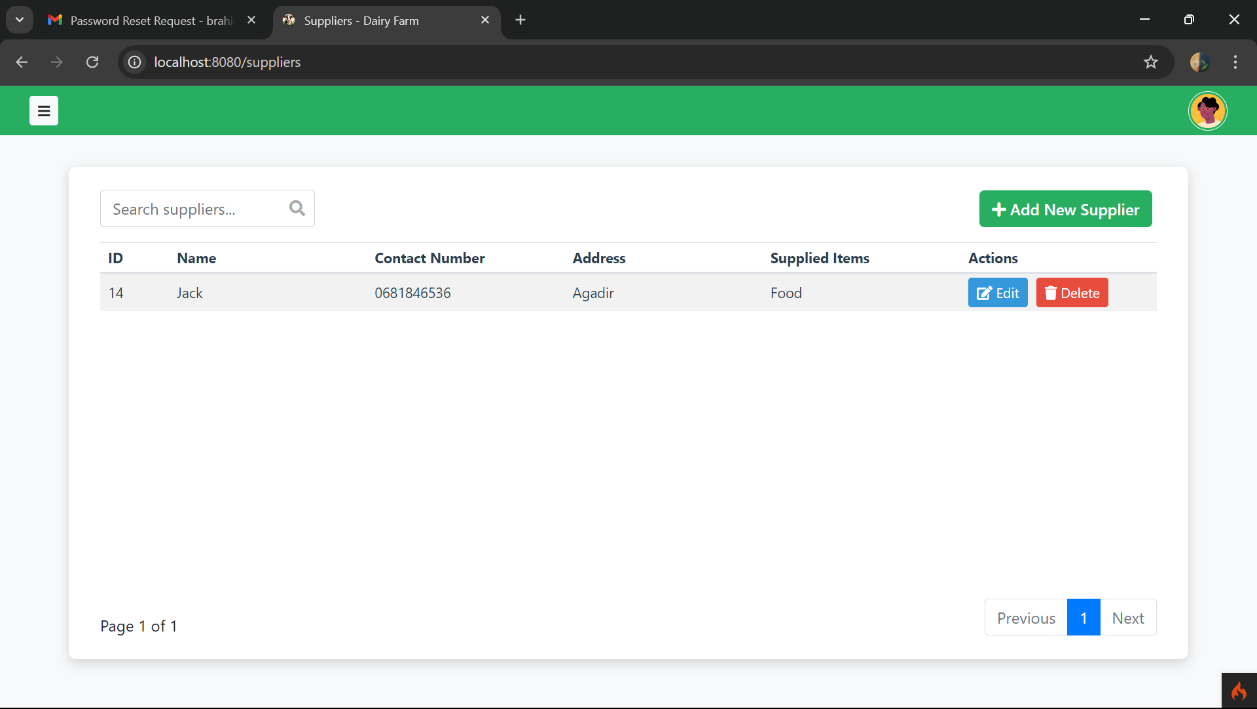


Figure 18 : Suppliers

### 11. Expenses:

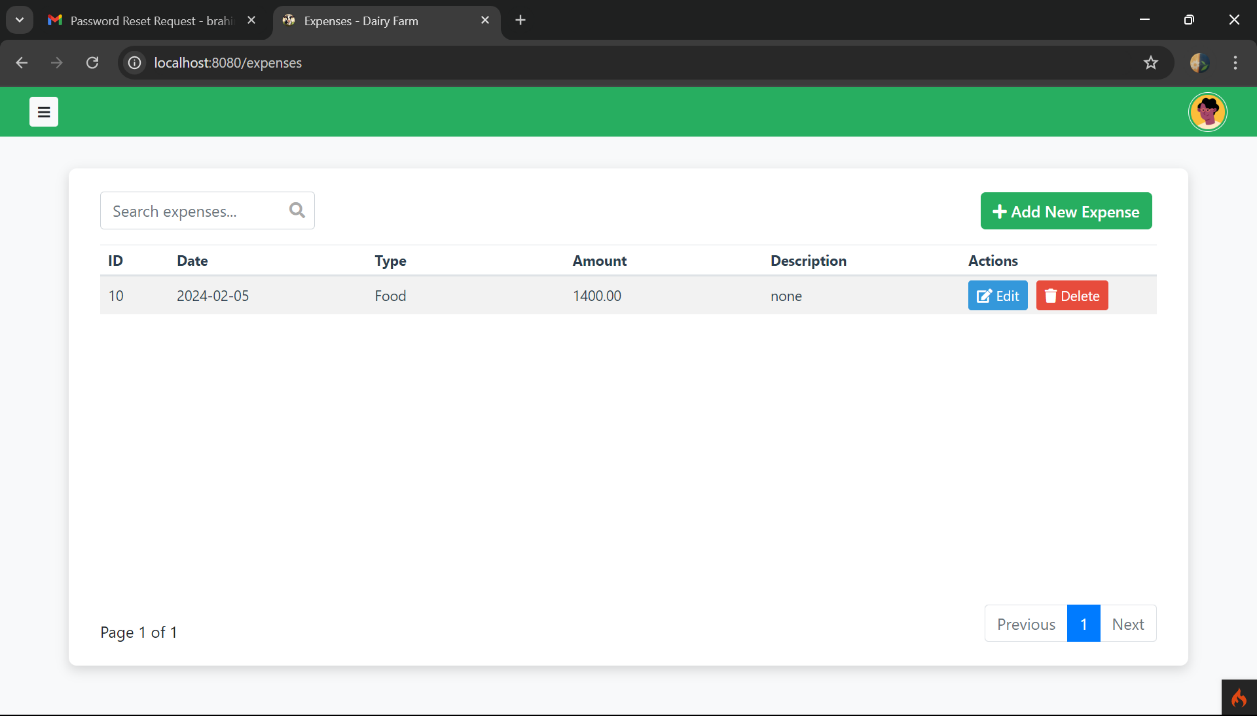


Figure 19 : Expenses

### 12. Employees:

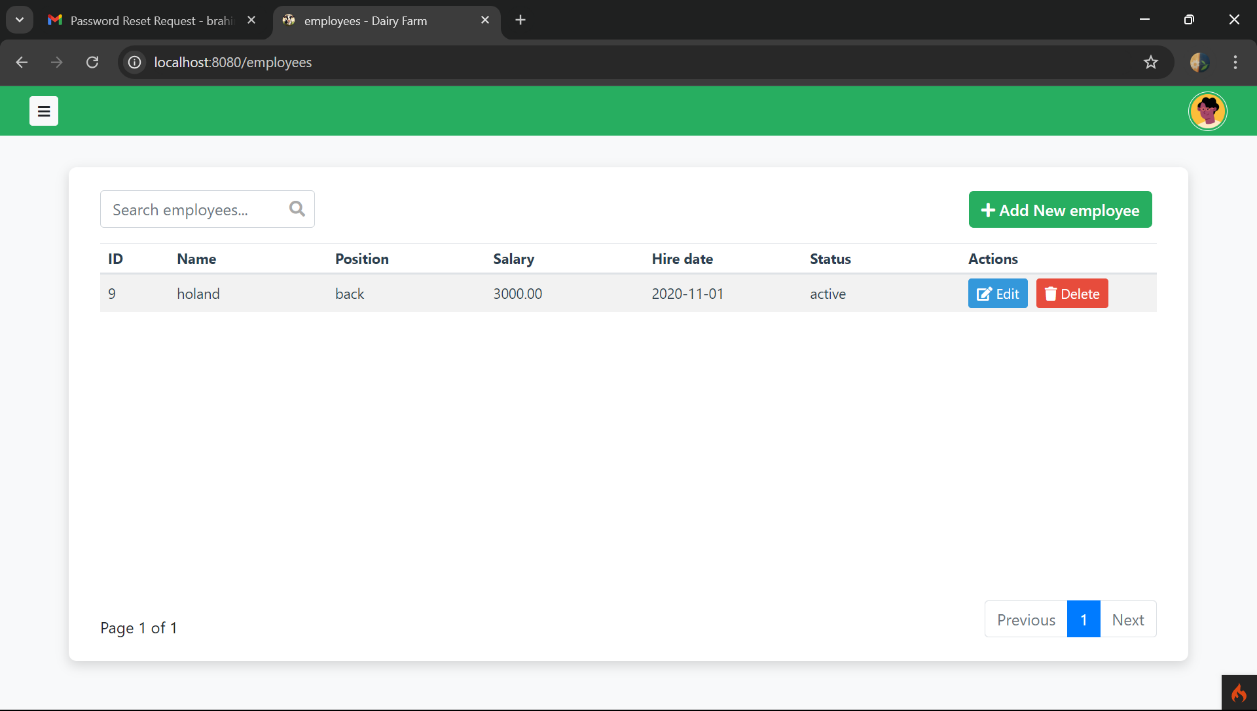


Figure 20 : Employees

### 13. Cow:

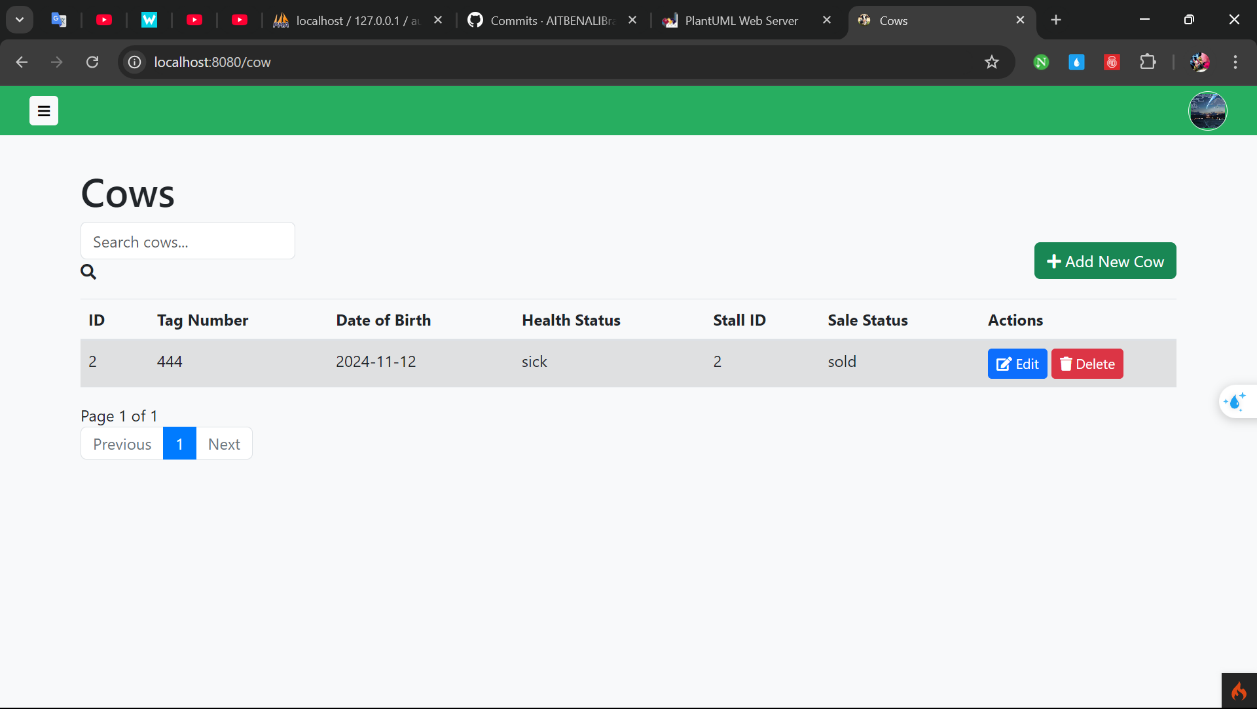


Figure 21 : Cow

### 14. Stalls:

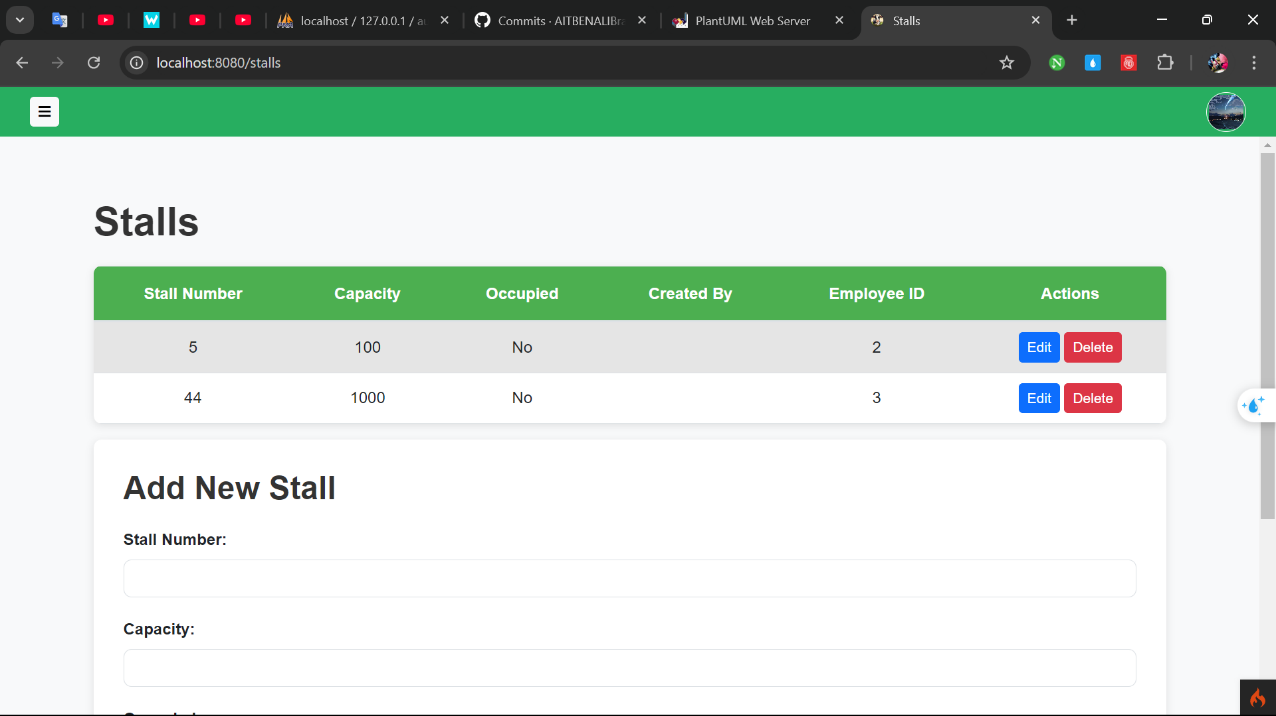


Figure 22 : Stalls

## Conclusion :

Dans la section "Réalisations", nous avons pour objectif de présenter les interfaces réalisées dans notre application de bureau. Cela permettra de clarifier les étapes d'utilisation de notre application, en mettant en évidence à la fois ses parties statiques et dynamiques. Nous visons à offrir une expérience utilisateur fluide et intuitive, en illustrant les fonctionnalités et les interactions disponibles à travers nos interfaces.

# Conclusion Générale

En conclusion, ce projet vise à offrir aux utilisateurs une plateforme complète permettant de gérer efficacement les employés, les animaux (vaches et veaux), la collecte et la vente de lait, les routines d’alimentation, les vaccinations, les grossesses, ainsi que les dépenses et les fournisseurs.

Tout au long de ce projet, nous avons acquis une expérience enrichissante en développement logiciel, en utilisant des technologies modernes telles que CodeIgniter 4, MySQL, Bootstrap, et des méthodologies agiles. Ces technologies ont permis de développer une solution modulaire, flexible et évolutive, répondant aux besoins spécifiques des fermes laitières. La collaboration avec notre encadrant et la prise en compte des retours des parties prenantes ont joué un rôle crucial dans l’amélioration continue de l’application.

Le système de gestion de la ferme Dairy Farm propose une solution fiable pour centraliser et automatiser les tâches quotidiennes. Ses fonctionnalités principales incluent la gestion des utilisateurs, des assistants et des employés, le suivi des performances animales (santé, reproduction, alimentation), la gestion des ventes et des dépenses, ainsi que la génération de rapports. Grâce à sa conception intuitive, cette plateforme simplifie les processus complexes et offre une vue globale des activités de la ferme.

Enfin, nous souhaitons reconnaître que, comme tout projet informatique, notre système n’est pas exempt de limitations ou de défis techniques. Nous restons ouverts aux retours constructifs et aux suggestions d’amélioration, dans l’objectif de continuer à perfectionner cette application et à mieux répondre aux besoins des utilisateurs à l’avenir.