|  |  |  |
| --- | --- | --- |
| **وزارة التعليم العالي و البحث العلمي** | | |
| **Universite Badji Mokhtar-Annaba**  Faculté des Sciences de l’Ingéniorat  Département d’Informatique |  | **جامعة باجي مختار – عنابـــــــــــــــة**  كلية علــــــــــــوم الهندســـــــــة  قسم الإعــــــــــــــلام الآلــــــــــــــي |

Mémoire présenté en vue d'obtenir le diplôme de   
**licence académique**

Intitulé :

**Une application web pour la gestion de la récolte des offices du blé en Algérie**

**Domaine :** Mathématiques-Informatique

**Filière :** Informatique

**Spécialité :** Informatique

**Préparé par :** M. Mohamed Hichem Laouar

**Encadré par :** M. Abdallah Benouareth

**Année : 2019-2020**

1. Remerciements

*Avant tout, j’aimerai remercier vivement mon encadreur Monsieur*

***Abdallah BENOUARETH*** *qui m’a suivi et dirigé tout au long de ce projet.*

*Un grand merci à mes parents pour tout leur soutien et dévouement.*

*Pour finir, je remercie toutes les personne ayant contribué de près ou de loin à la réalisation de ce projet.*

# Dédicaces

*Je dédie ce mémoire*

*À* *ma famille,*

*À* *mes Amis*

Table de matières

|  |  |  |
| --- | --- | --- |
| Remerciements | ………………... | 2 |
| Dédicaces | ………………... | 3 |
| Table des matières | ………………... | 4 |
| Table des figures ……………………………………………………………. | ………………... | 6 |
| Introduction ……………………………………………………………………. | ………………... | 9 |
| * 1. Contexte du projet | ………………... | 9 |
| * 1. Problématique | ………………... | 9 |
| * 1. Motivations | ………………... | 9 |
| * 1. Objectifs | ………………... | 9 |
| * 1. Contenu de mémoire | ………………... | 10 |
| Chapitre 1 : Description générale du projet | ………………... | 11 |
| 1. Introduction | ………………... | 11 |
| 1. C’est quoi le Front-end ? | ………………... | 11 |
| 1. C’est quoi le Back-end ? | ………………... | 11 |
| 1. La technologie Ajax | ………………... | 12 |
| 1. Conclusion | ………………... | 12 |
| Chapitre 2 : Conception de l’application | ………………... | 12 |
| 1. Introduction | ………………... | 13 |
| 1. Hypothèse de travaille | ………………... | 13 |
| 2.1. Besoin fonctionnelles | ………………... | 13 |
| 2.2. Besoin non-fonctionnelles | ………………... | 14 |
| 1. Conception UML | ………………... | 14 |
| 3.1. Diagramme de cas d’utilisation | ………………... | 15 |
| 3.2. Diagramme de classe | ………………... | 15 |
| 3.3. Diagrammes de séquence | ………………... | 16 |
| 3.3.1. Inscription d’un utilisateur | ………………... | 17 |
| 3.3.2. Connexion d’un utilisateur | ………………... | 18 |
| 3.3.3. Valider un compte | ………………... | 18 |
| 3.3.4. prendre un rendez-vous | ………………... | 19 |
| 3.3.5. Ajouter une récolte | ………………... | 20 |
| 3.3.6. Voir les informations d’une récolte | ………………... | 21 |
| 1. Architecture fonctionnelle de l’application | ………………... | 22 |
| 4.1. Introduction | ………………... | 22 |
| 4.2. La collection utilisateurs | ………………... | 22 |
| 4.3. La collection rendez-vous | ………………... | 23 |
| 4.4. La collection récoltes | ………………... | 23 |
| 1. Conclusion | ………………... | 24 |
| Chapitre 3 : implémentation et présentation de l’application | ………………... | 24 |
| 1. Les technologies utilisées | ………………... | 24 |
| 1.1. Technologies utilisées dans le Front-end | ………………... | 24 |
| 1.2. Technologies utilisées dans le Back-end | ………………... | 25 |
| 1.3. La configuration matérielle et logicielle nécessaire | ………………... | 25 |
| 1. Implémentation de l’application | ………………... | 25 |
| 2.1. Inscription | ………………... | 25 |
| 2.2. Connexion | ………………... | 26 |
| 2.3. Validation des comptes | ………………... | 27 |
| 2.4. Les rendez-vous | ………………... | 28 |
| 2.4.1. Ajouter un rendez-vous | ………………... | 28 |
| 2.4.2. Prendre un rendez-vous | ………………... | 28 |
| 2.5. Les récolte | ………………... | 29 |
| 2.5.1. Ajouter une récolte | ………………... | 29 |
| 2.5.2. Ajouter une facture | ………………... | 29 |
| 1. Présentation de l’application | ………………... | 30 |
| Conclusion et perspectives | ………………... | 31 |
| Références | ………………... | 32 |
| Annexe A | ………………... | 33 |
| 1. Les interfaces utilisateur | ………………... | 33 |
| 2.1. Les utilisateurs | ………………... | 33 |
| 2.1. Les rendez-vous | ………………... | 35 |
| 2.1. Les récoltes | ………………... | 36 |
| 2.1. Les produits | ………………... | 39 |
| Résumé | ………………... | 41 |

# Tables des figures

[*Figure 1 : les technologies populaires utilisées dans le front end 12*](#_Toc41847061)

[*Figure 2 : les technologies populaires utilisées dans le back end 12*](#_Toc41847062)

[*Figure 3 : le mécanisme de travaille de la technologie Ajax 13*](#_Toc41847063)

[*Figure 4 : Diagramme de cas d'utilisation 16*](#_Toc41847064)

[*Figure 5 : diagramme de classe 17*](#_Toc41847065)

[*Figure 6 : diagramme de séquence d'inscription 18*](#_Toc41847066)

[*Figure 7 : diagramme de séquence de connexion 19*](#_Toc41847067)

[*Figure 8 : diagramme de séquence de la validation d'un compte 20*](#_Toc41847068)

[*Figure 9 : diagramme de séquence pour prendre un rendez-vous de dépôt de la récolte 21*](#_Toc41847069)

[*Figure 10 : diagramme de séquence pour ajouter une récolte 22*](#_Toc41847070)

[*Figure 11 : diagramme de séquence pour voir les informations d’une récolte. 23*](#_Toc41847071)

[*Figure 12 : un exemple de la collection utilisateurs 24*](#_Toc41847072)

[*Figure 13 : un exemple de la collection rendez-vous 24*](#_Toc41847073)

[*Figure 14 : un exemple de la collection récoltes 24*](#_Toc41847074)

[*Figure 15 Code d'inscription d'un agriculteur 27*](#_Toc41847075)

[*Figure 16 Code d'inscription d'un office 27*](#_Toc41847076)

[*Figure 17 : code pour la connexion d'un agriculteur 28*](#_Toc41847077)

[*Figure 18 : code de validation d’un compte agriculteur 28*](#_Toc41847078)

[*Figure 19 : code de validation d’un compte office 28*](#_Toc41847079)

[*Figure 20 : code d’ajout d'un rendez-vous 29*](#_Toc41847080)

[*Figure 21 : code pour prendre un rendez-vous 29*](#_Toc41847081)

[*Figure 22 : code de l'ajout d’une récolte 30*](#_Toc41847082)

[*Figure 23 : code d'ajout d'une facture 30*](#_Toc41847083)

[*Figure 24 : le code pour récupérer les récoltes d'un office pendant une année donnée regroupées par mois 31*](#_Toc41847084)

[*Figure 25 : le code pour récupérer la quantité des récoltes de toutes les wilayas pendant une année donnée 31*](#_Toc41847085)

[*Figure 26 : interfaces d'inscription d’un agriculteur et d’un office du blé 34*](#_Toc41847086)

[*Figure 27 : interface de connexion 34*](#_Toc41847087)

[*Figure 28 : le profil d'un agriculteur 35*](#_Toc41847088)

[*Figure 29 : interface de la modification des informations du compte d'un agriculteur 35*](#_Toc41847089)

[*Figure 30 : le formulaire d'ajout d'un rendez-vous 36*](#_Toc41847090)

[*Figure 31 : les offices du blé les plus proches d'un agriculteur 36*](#_Toc41847091)

[*Figure 32 : la liste des rendez-vous disponibles proposés par un office 36*](#_Toc41847092)

[*Figure 33 : un rendez-vous pris par un agriculteur 37*](#_Toc41847093)

[*Figure 34 : liste des rendez-vous pris proposé par un office 37*](#_Toc41847094)

[*Figure 35 : le formulaire d'ajout d'une récolte 37*](#_Toc41847095)

[*Figure 36 : la liste des récoltes d'un agriculteur 38*](#_Toc41847096)

[*Figure 37 : les récoltes d'un office 38*](#_Toc41847097)

[*Figure 38 : les informations d’une récolte 39*](#_Toc41847098)

[*Figure 39 : la liste des récoltes par année d'un office 39*](#_Toc41847099)

[*Figure 40 : un graphique des récoltes d'un office par mois pendant l’année 2020 39*](#_Toc41847100)

[*Figure 41 : un graphique du revenu mensuel d’un office pendant l’année 2020 40*](#_Toc41847101)

[*Figure 42 : récoltes pour wilaya 40*](#_Toc41847102)

[*Figure 43 : le graphique de la récolte par wilaya 40*](#_Toc41847103)

[*Figure 44 : le formulaire d'ajout d'un produit 41*](#_Toc41847104)

[*Figure 45 : la liste des produits 41*](#_Toc41847105)

# Introduction

## Contexte du projet

L'agriculture est un facteur important de l'économie de l'Algérie, Elle génère elle-même, sans les industries agroalimentaires, près de 12,3% du produit intérieur brut (PIB) en 2017 **[1]**, depuis l'indépendance de l'Algérie de nombreux offices du blé sont créés, divisés, regroupés au fil du temps et des besoins pour soutenir les actions du ministère de l'agriculture dans l'organisation et le règlement des marchés, ces offices permettant une collaboration et transaction avec les agriculteurs à travers toutes les wilayas d’Algérie.

## Problématique

La transaction informatisée entre les agriculteurs et les offices du blé en Algérie pose un problème de communication ainsi que l’accès aux statistiques des récoltes de ces derniers et de chaque wilaya d’Algérie. Dans le but de remédier à ces problèmes et de rapprocher les agriculteurs aux offices du blé nous souhaitons donc arriver à la meilleure solution informatisée.

## Motivations

Vu le manque de la gestion informatique dans le domaine agricole en Algérie, et l’accès difficile aux statistiques des récoltes, comment remédier à ce manque ? Pourquoi ne pas développer un système qui prendra en charge l'informatisation de ce domaine ce qui permettra la fluidité de transaction entre l'agriculteur et l'office du blé dans toutes les activités et ainsi de rendre les statistiques des rendements agricoles accessibles à travers toutes les wilayas d’Algérie.

## Objectifs

L’objectif de ce projet est de faciliter la gestion de la récolte des offices du blé en Algérie dans lesquelles il permettra la gestion informatisée des rendez-vous du dépôt de la récolte et aussi la classification de ces dernières selon leurs qualités ,la sauvegarde et l’ordonnancement , la facturation des statistiques de la récolte de chaque agriculteur et office pour chaque année afin de les rendre accessibles.

## Contenu du mémoire

* **CHAPITRE 01 :** Dans le premier chapitre nous allons décrire quelques définitions nécessaire pour la compréhension de notre application.
* **CHAPITRE 02 :** Dans le deuxième chapitre nous allons présenter la conception générale du système.
* **CHAPITRE 03 :** Dans letroisièmechapitre nous allons expliquer les technologies et les environnements utilisés pour la réalisation de notre application.

Et enfin, nous terminerons par une conclusion générale et des perspectives qui pourraient être envisageables.

# Chapitre 1 : Description générale du projet

## Introduction

\*Dans ce chapitre nous allons présenter des définitions nécessaires et indispensables pour la compréhension de notre application, ces définitions sont notamment des concepts techniques que nous allons les utiliser pour la réalisation de cette dernière.

## C’est quoi le Front-end ?

Lorsque l'on parle de «Front-End», il s'agit des éléments du site web que l'on voit à l'écran et avec lesquels on peut interagir. Ces éléments sont composés de HTML, CSS et de JavaScript contrôlés par le navigateur web de l'utilisateur **[2].**

Il existe plusieurs Framework et bibliothèques « JavaScript » utilisés pour rendre le développement des interfaces utilisateur simple et plus rapide qu’avant parmi ces bibliothèques (jquery, react js,..) et des Framework comme (vue js, angular js,..).



Figure : les technologies populaires utilisées dans le front end

## C’est quoi le Back-end ?

Lorsque l'on parle de «Back-End», il s'agit de la partie du site web qu’on ne la voit pas, elle est invisible pour les visiteurs mais représente une grande partie du développement d'un projet web. Sans elle, le site web reste une coquille vide **[2]**.

Il est composé essentiellement par le serveur qui représente l’hébergeur de site web et une base de données qui est composée par des tableaux qui contient des données, pour pouvoir accéder et modifier ces données il faut qu’on utilise un langage de programmation comme (Php, C#, Java…).



Figure : les technologies populaires utilisées dans le back end

## La technologie Ajax

AJAX *(*Asynchrones JavaScript and XML) est une technologie permet de communiquer avec le serveur à l'aide du code « JavaScript » en arrière-plan pendant que la page est affichée à l'écran sans qu'elles nécessitent le moindre rechargement visible par l'utilisateur de la page Web **[3]**.

Parmi les avantages de cette technologie est de réduire le trafic entre le client et le serveur donc le temps de réponse devient rapide ce qui augmente les performances.

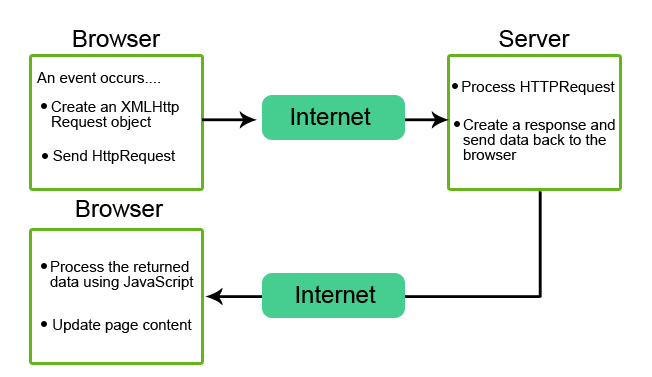


Figure : le mécanisme de travaille de la technologie Ajax

## Conclusion

Dans ce chapitre nous avons présenté quelques définitions techniques indispensables pour la compréhension de notre projet que nous allons utiliser pour la réalisation de ce projet. Le chapitre prochain sera dédié à la conception générale de notre application.

# Chapitre 2 : conception de l’application

## Introduction

Dans ce chapitre nous allons présenter la conception générale du projet et aussi les fonctionnalités et les services que l’application doit fournir en utilisant des diagrammes « UML »pour modéliser les besoins d’utilisateurs.

## Hypothèses du travail

L’objectif de cette partie est de définir les besoins fonctionnels qui représentent les fonctionnalités que le système doit fournir et les besoins non fonctionnels qui définissent les caractéristiques du système.

### Besoins fonctionnels

Les fonctionnalités que nous system doit fournir est :

* **l’authentification** : après avoir créé un compte cette fonctionnalité permet à l’utilisateur d’accéder à son compte pour utiliser les fonctionnalités offertes.
* **La gestion des comptes des agriculteurs** : seuls les administrateurs des offices peuvent valider des comptes d’agriculteurs et aussi ajouter des nouveaux comptes.
* **La gestion des comptes des offices** : seuls les administrateurs des offices responsables qui sont bien sur des administrateurs d’office peuvent valider les comptes des offices et ajouter des offices responsables.
* **La gestion des rendez-vous**: cette fonctionnalité permet les administrateurs d’offices d’ajouter des rendez-vous, supprimé des rendez-vous déjà existe et annuler des rendez-vous déjà pris par des agriculteurs, d’autre coté les agriculteurs peuvent prendre des rendez-vous offerts par des offices de leurs wilayas et aussi annuler des rendez-vous qu’ils ont déjà pris.
* **La gestion des récoltes :** cette fonctionnalité permet aux administrateurs d’offices d’ajouter des récoltes suite à des rendez-vous pris par des agriculteurs et aussi établit des factures de vente qui regroupe une ou plusieurs récoltes.
* **La consultation des statiques des récoltes**: cette fonctionnalité permet aux utilisateurs de visualiser leurs récoltes triées par un ordre sélectionné et aussi visualisé des graphes qui représentent la croissance des récoltes par mois pendant chaque année.
* **La gestion du profil :** cette fonctionnalité permet aux utilisateurs de changer leurs informations personnelles dès qu’ils souhaitent.
* **La gestion des produits :** seuls les administrateurs d’offices responsables peuvent accéder àcette fonctionnalité, elle permet d’ajouter des nouveaux produits, modifier ou supprimer des produits déjà existe.
* **Envoyer des newsletters :** cette fonctionnalité permet les administrateurs d’offices d’envoyer des mails d’actualité et des nouveautés aux personnes qui ont abonnée à la newsletter.

### Besoins non fonctionnels

Notre système doit répondre aux exigences de qualité suivantes :

* **Conviviale :** notre application doit être facile à utiliser, dans lesquelles les interfaces utilisateur doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.
* **Valide :** le système doit remplir exactement les besoins spécifiés précédemment.
* **Sécurisé** : le système doit être sécurisé dans lesquelles chaque utilisateur identifié par un email et un mot de passe et il ne peut pas voir les informations personnelles des autres utilisateurs, chaque type d’utilisateur peut accéder à des taches particulières.
* **Fiable :** le système doit être accessible 24 heures sur 24, 7 jours sur 7et fonctionne dans tous les cas normaux et anormaux (cas d’erreur, entrée avec erreur).
* **Performances:** le système doit être performant dans laquelle tous les taches sont rapides.
* **Maintenabilité**: le système doit être facile à modifier pour s'adapter aux nouvelles fonctionnalités (évolution).

## Conception UML

### Diagramme de cas d’utilisation

Un diagramme de cas d'utilisation est un diagramme « UML » utilisé pour donner une vision globale du comportement fonctionnel d’un système, il représente une unité discrète d'interaction entre un utilisateur et un système. **[2]**

Le diagramme de cas d’utilisation de notre application :

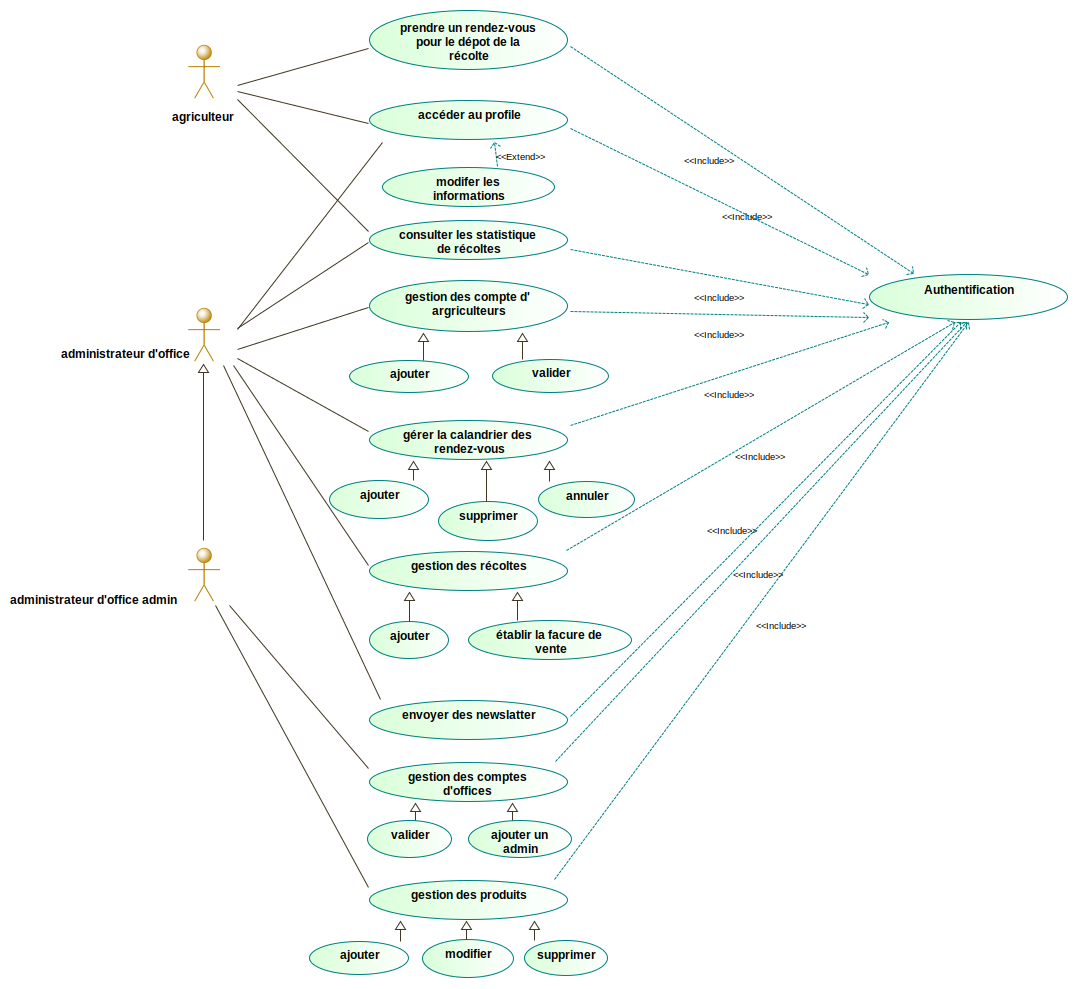


Figure : Diagramme de cas d'utilisation

Les acteurs de notre diagramme sont : l’agriculteur, l’administrateur d’office et l’administrateur d’office responsable.

* L’agriculteur : il peut prendre des rendez-vous pour le dépôt de la récolte.
* L’administrateur d’office et l’administrateur d’office responsable peuvent gérer des comptes d’agriculteurs, les rendez-vous de dépôt de la récolte, les récoltes.
* Seul l’administrateur d’office responsable peut gérer les comptes d’offices et les produits.
* Tous les utilisateurs peuvent modifier leurs profiles et consulter les statistiques des récoltes de façon que l’agriculteur peut voir seulement ses récoltes et l’administrateur de l’office peut voir tous les récoltes qui étaient récoltée dans son office.

### Diagramme de classe

Le diagramme de classe est un diagramme « UML » utilisé pour représenter la structure d’une application orientée objet en montrant les classes et les relations qui s’établissent entre elles.

Le diagramme de clases de notre application :

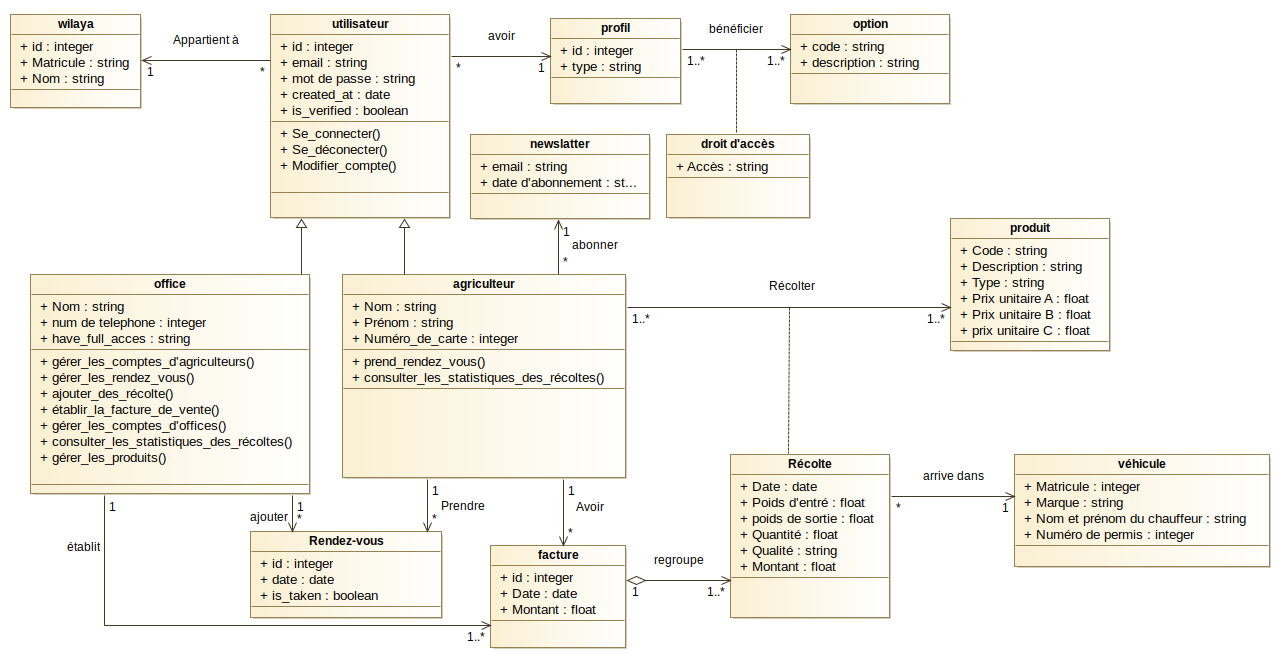


Figure : diagramme de classe

La classe principale de notre diagramme est l’utilisateur, il peut être un agriculteur ou un office, il appartient à une wilaya et possède un profil, chaque profil bénéfice des options et dans chaque option il y a un droit d’accès (accès complète, insérer, …).

L’agriculteur peut prendre des rendez-vous pour le dépôt de la récolte, ces rendez-vous sont ajoutés par des offices de sa wilaya, il peut aussi récolter un ou plusieurs produits et dans chaque récolte on garde les informations de ce dernier et aussi du véhicule utilisée. Chaque récolte peut faire partie d’une et une seule facture, cette dernière appartienne à un agriculteur et établit par un office.

### Diagrammes de séquence

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs, ils sont organisés en fonction du temps.

Les diagrammes de séquences de notre application sont :

### Le diagramme de séquence pour l’inscription d’un utilisateur

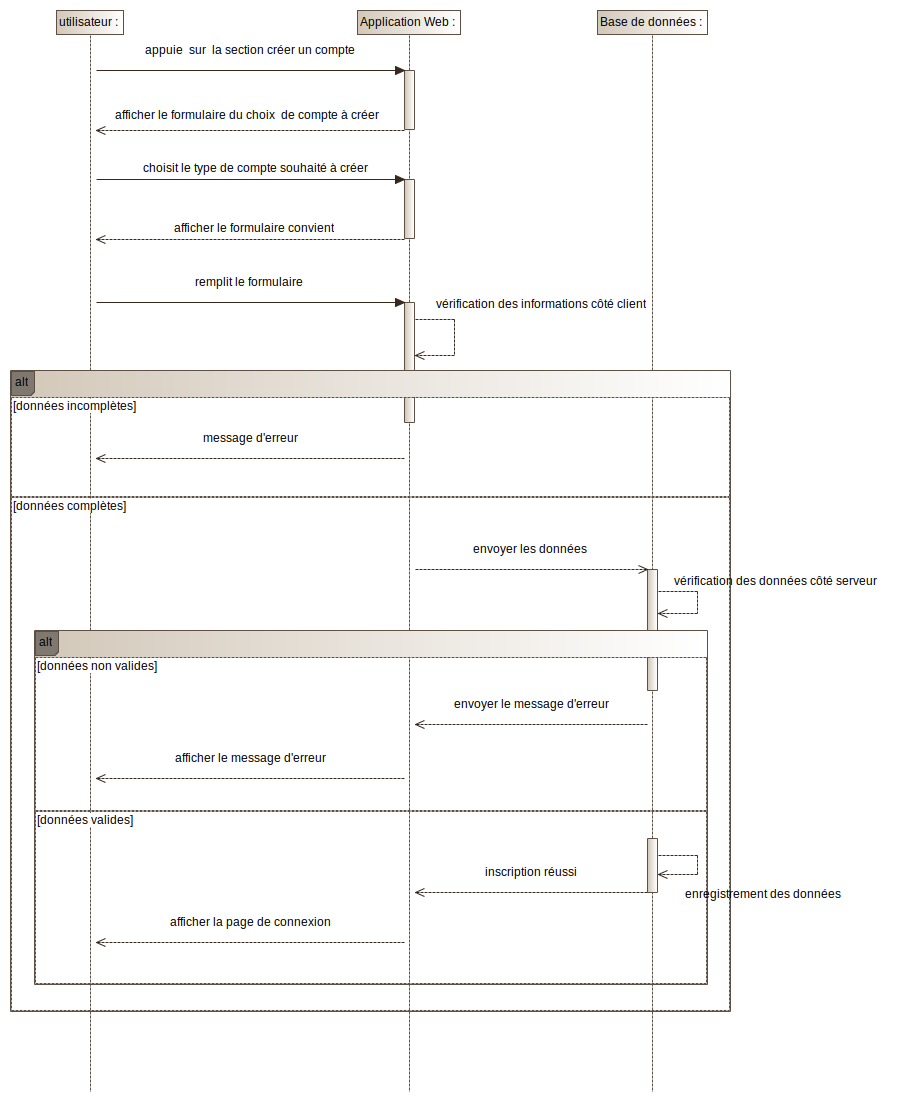


Figure : diagramme de séquence d'inscription

Ce diagramme représente le diagramme de séquence de l’inscription d’un nouvel utilisateur tous d’abord l’utilisateur choisit le type de compte à créer l’application ensuite affiche le formulaire convient, après avoir rempli ce formulaire l’application lance deux vérifications la première dans le côtéclient pour assurer que les informations entrées sont complètes et l’autre dans le côtéserveur pour vérifier que ces dernières n’existent pas déjà ensuite ces information serons envoyé vers la base de données.

### Le diagramme de séquence pour la connexion d’un utilisateur

Dans la connexion l’utilisateur qui est déjà inscrit remplit le formulaire de connexion et comme dans l’inscription l’application lance deux vérifications sauf que la vérification en côtéserveur réussite si et seulement si les informations entrées sont identiques à celles qui sont dans la base de données et aussi que le compte de l’utilisateur doit être vérifié pour accéder à la page d’accueil.

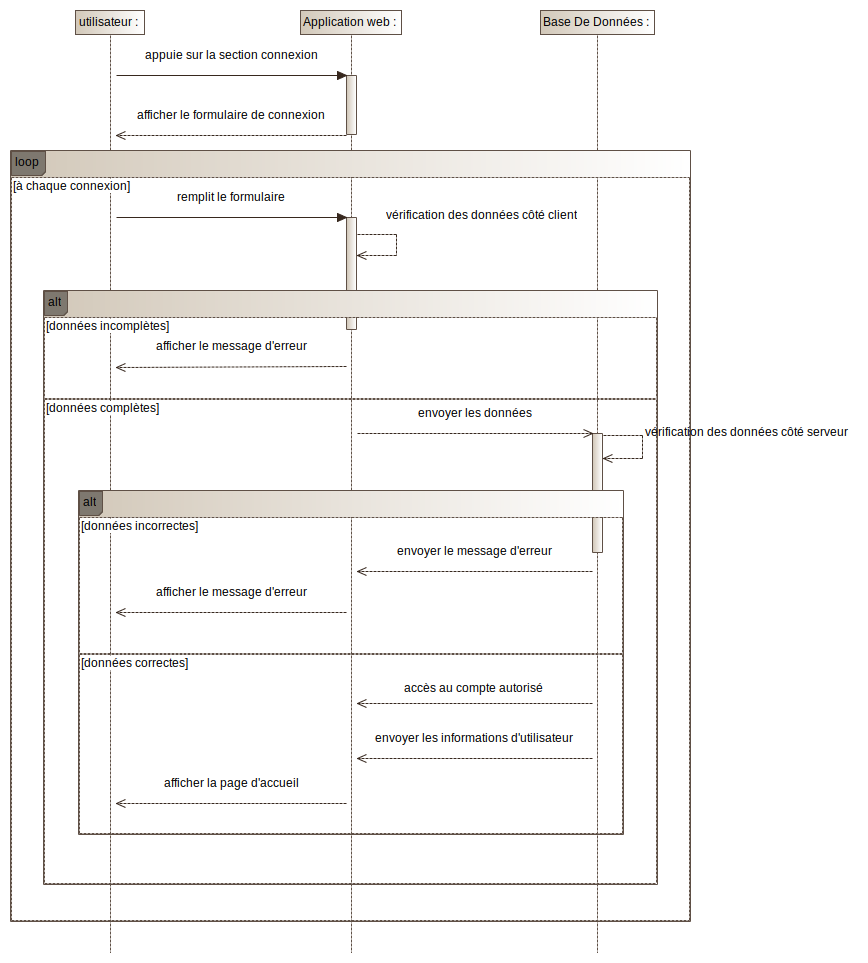


Figure : diagramme de séquence de connexion

### Le diagramme de séquence pour valider un compte

Ce diagramme montre comment l’administrateur d’un office valide le compte d’un utilisateur, tous d’abord il faut que cet administrateur s’authentifie, après avoir choisi un compte pour le valider tous les informations seront envoyées à la base de données pour modifier ce compte, enfin et à l’aide de la bibliothèque « PhpMailer » un mail sera envoyé à l’utilisateur concerner pour lui indiquer que son compte a été validé.

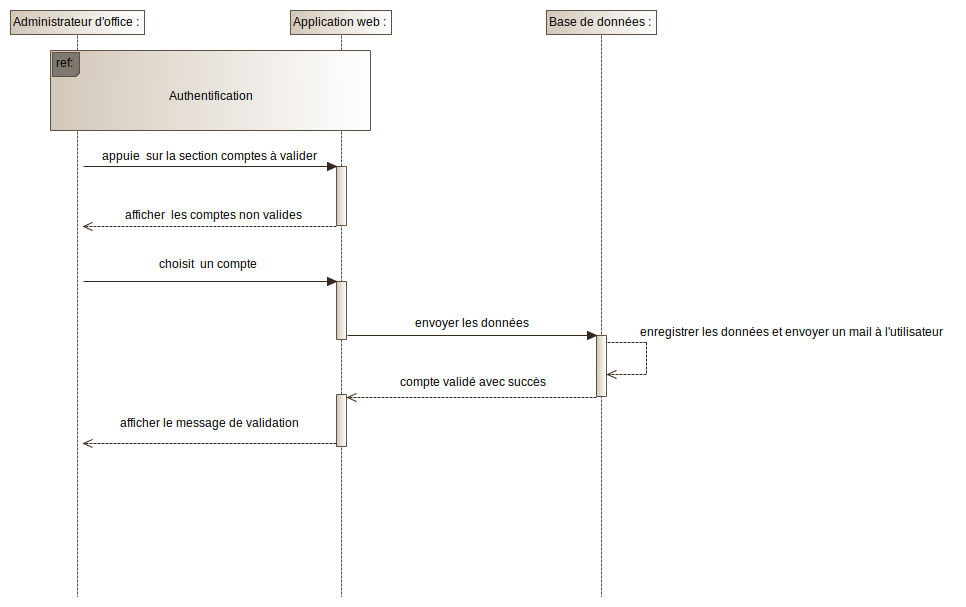


Figure : diagramme de séquence de la validation d'un compte

### Le diagramme de séquence pour prendre un rendez-vous

Ce diagramme montre comment l’agriculteur prend un rendez-vous pour la dépôt de la récolte, après avoir authentifier, il appuie sur la section « rendez-vous » l’application affiche tous les offices les plus proches il choisit un, ensuite l’application va afficher les rendez-vous disponibles proposés par cet office ,après avoir choisi un rendez-vous toutes les information de ce rendez-vous seront enregistrés dans la base de données ,enfin l’application génère une convocation du type PDF contient toutes les information à propos de ce rendez-vous.

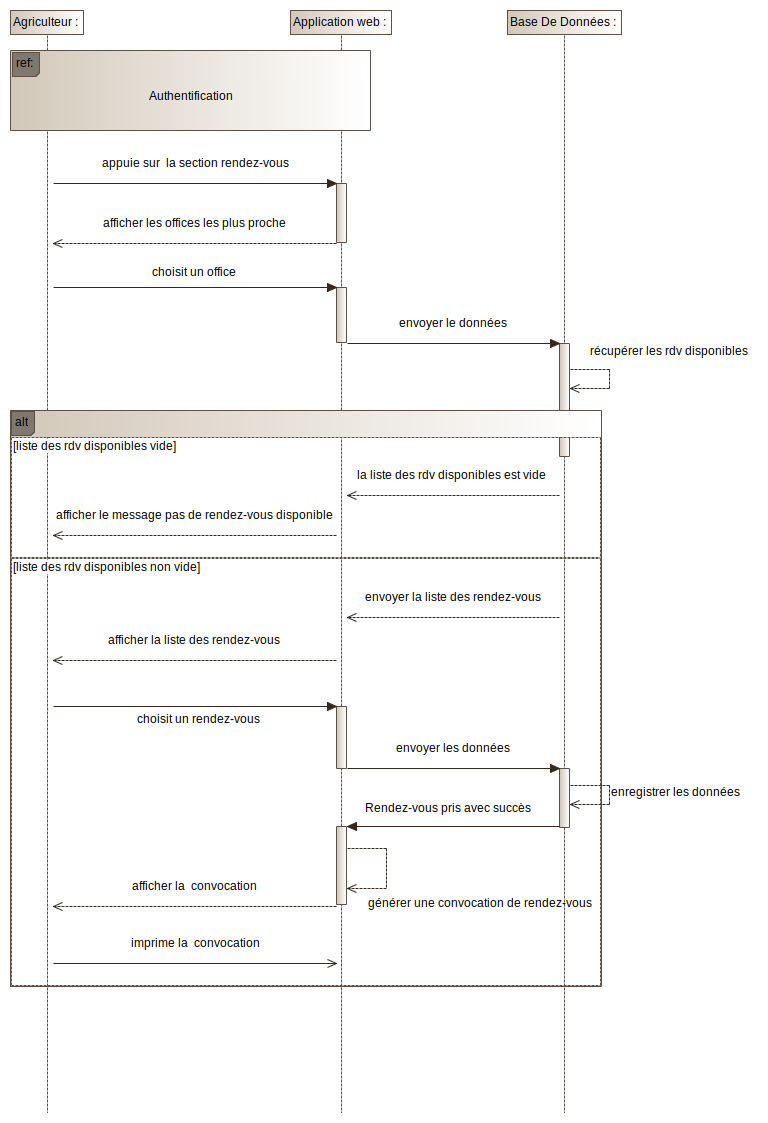


Figure : diagramme de séquence pour prendre un rendez-vous de dépôt de la récolte

### Le diagramme de séquence pour ajouter une récolte

Ce diagramme montre comment l’administrateur d’office ajout une récolte, lors d’un rendez-vous de dépôt de la récolte d’un agriculteur, l’administrateur appuie sur la section rendez-vous pris pour ajouter la récolte, après avoir rempli le formulaire d’ajout et la vérification des informations entrées, ces dernières seront enregistrées dans notre base de données.

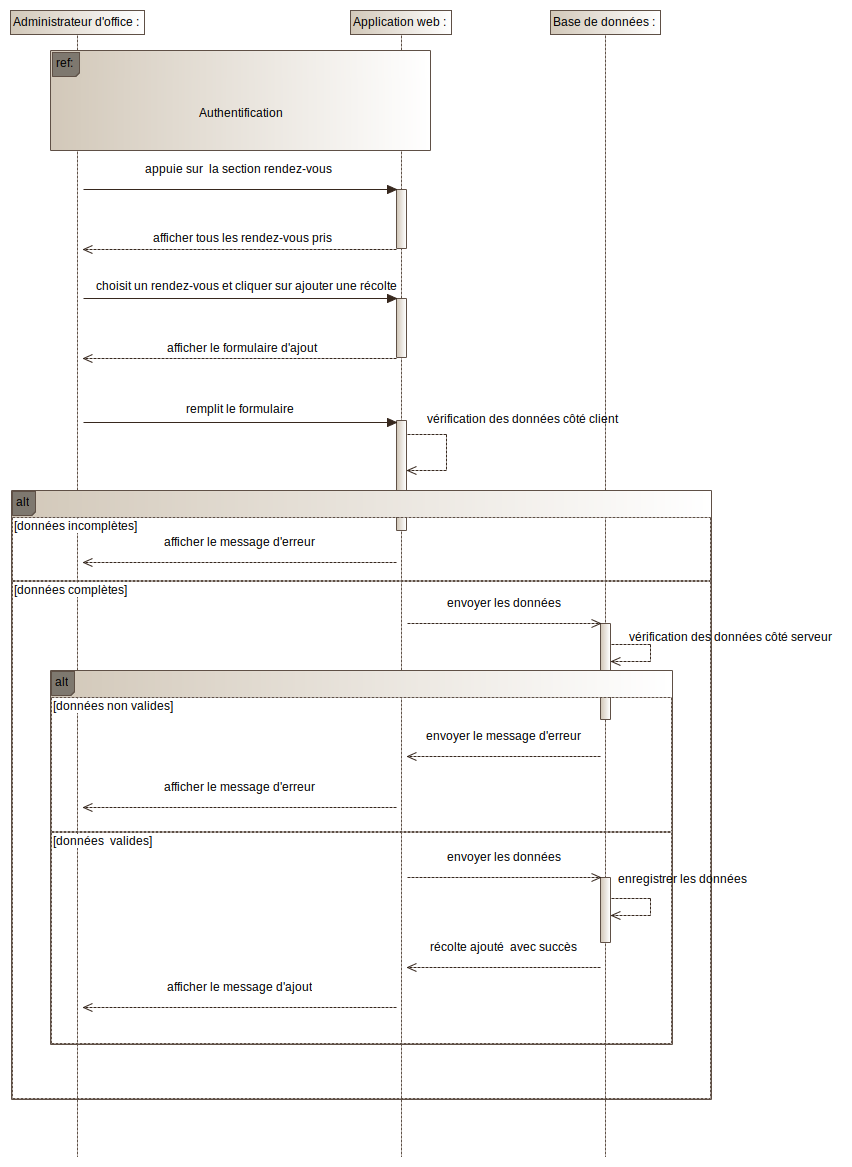


Figure : diagramme de séquence pour ajouter une récolte

### Le diagramme de séquence pour voir les informations d’un récolte

Ce diagramme montre comment chaque utilisateur voit les informations d’une récolte qui lui appartienne, après avoir consulté la section récoltes et choisit une récolte toutes les informations de cette dernière seront affichées.

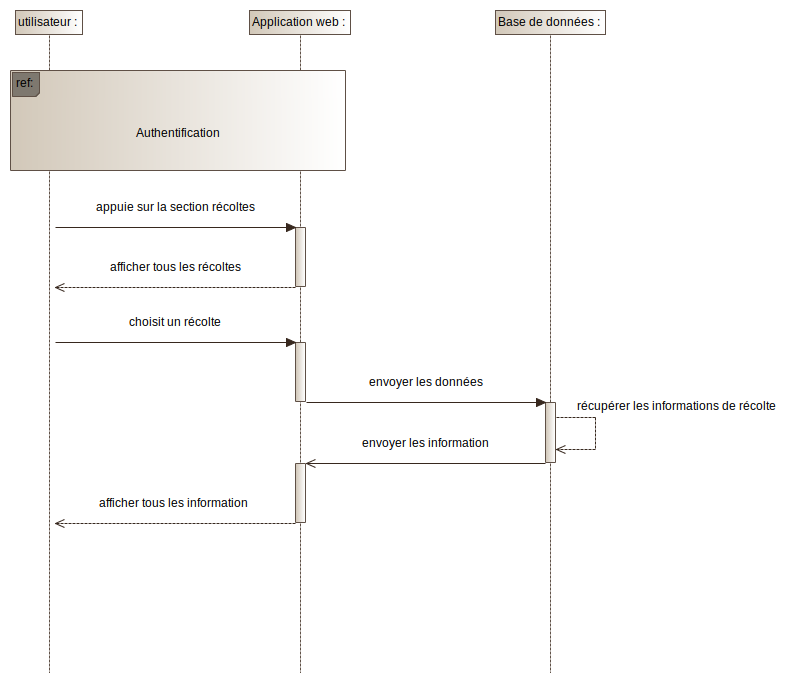


Figure : diagramme de séquence pour voir les informations d’une récolte.

## Architecture fonctionnelle de l’application

### Introduction

Dans cette partie nous allons présenter les données qui sont exploitées par notre application en montrant du fragment de notre base de données à l’aide de notre SGBD « PhpMyAdmin » qui nous permet d’administrer notre base de données facilement.

### La collection utilisateurs

Cette collection regroupe les informations d’utilisateurs qui sont notamment des agriculteurs et des offices du blé, ces deux dernières ont des informations partagées (id, l’email, le mot de passe, id de la wilaya qui la appartienne...) qui sont regroupés dans la table utilisateurs et d’autres informations spécifiques pour chaque un : pour l’agriculteur (le nom, le prénom, le numéro de carte, l’avatar) et pour l’office (le nom, le numéro de téléphone, l’id de la wilaya qui la appartienne).

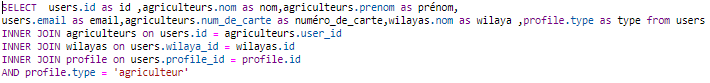




Figure : un exemple de la collection utilisateurs

### La collection rendez-vous

Cette collection regroupe les informations des rendez-vous de dépôt de la récolte qui sont ajoutées par des offices dans laquelle chaque rendez-vous composé par (un identifiant, sa date, l’identifiant de l’office qui l’ajouté, l’identifiant de l’agriculteur qui la prit s’il est déjà pris).

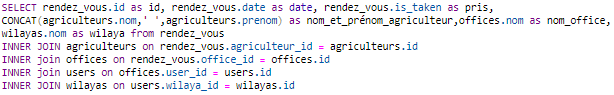
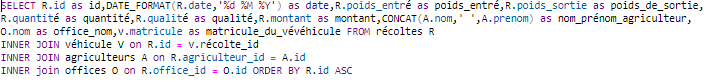




Figure : un exemple de la collection rendez-vous

### La collection récoltes

Cette collection présente les récoltes des agriculteurs qui sont ajoutées par des offices, chaque une comporte (un identifiant, la date, la qualité, la quantité, le montant, l’id de l’office qu'il a ajouté, l’id de l’agriculteur qui il a récolté, le code du produit récolté dans cette récolte), chaque récolte avoir aussi un véhicule dans laquelle il vient, et il fait partie d’une seule facture, cette dernière comporte au moins une récolte.



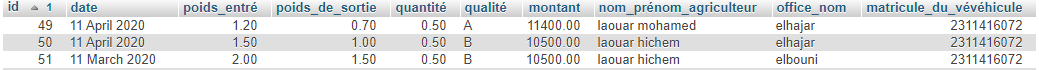


Figure : exemple de la collection récoltes

## Conclusion

Dans ce chapitre nous avons présenté les fonctionnalités que l’application doit fournir en utilisant des diagrammes « UML », nous avons enfin présenté les données qui sont exploitées par notre application.

Dans le prochain chapitre nous allons aborder l’implémentation de notre application.

# Chapitre 3 : Implémentation et présentation de l’application

Dans ce chapitre nous allons expliquer les technologies et les environnements utilisés pour la réalisation de ce projet et aussi donner le code source de certaines fonctionnalités et des captures de l’interface utilisateur.

## Les technologies utilisées

Dans cette partie nous allons expliquer les différents technologies utilisées pour développer notre application et la configuration matérielle et logicielle nécessaire pour l‘exécution de cette application.

### 1.1. Technologies utilisées dans le front-end

Dans le front-end de notre application nous avons utilisé :

**Html :** (HyperText Markup Language)  C’est un langage de balisage qui est utilisé pour créer des pages web qui va nous permettre de définir les différents contenus d’une page web

**Css :** (Cascading Style Sheets) c’est un langage de style qui va nous permettre de modifier l’apparence des contenues de la page en appliquant des styles

**JavaScript** : un langage de script exécuté sur l'[o](https://www.futura-sciences.com/tech/definitions/informatique-ordinateur-586/)rdinateur de l’utilisateur par le navigateur il permet une interaction avec l'utilisateur en fonction de ses actions (passage de souris, click...)

**Jquery** : c'est une bibliothèque JavaScript qui permet de coder plus vite et plus simplement. **[4]**

**JsPDF** : c’est une bibliothèque JavaScript qui génère des documents « PDF » en utilisant des données stockées dans la base de données (dans notre cas on l’utilise pour générer des convocations des rendez-vous et des factures). **[5]**

**ChartJs** : c’est une bibliothèque JavaScript qui génère des déférents types des graphes (barre, circulaire, anneau..) en basant sur des données stockées dans la base des données. **[6]**

**Swwetalert2** : c’est une bibliothèque JavaScript qui permet de créer des «Pop-up  » pour les alertes facilement. **[7]**

### 1.2. Technologies utiliser dans le Back-end

Dans notre application nous avons utilisé « XAMPP » qui est une plateforme de développement web permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts « PHP », elle est composée essentiellement  par un serveur Apache « http », la base de données « MariaDB » et « PhpMyAdmin » pour l’administration des bases de données. **[8]**

Nous avons également utilisé la technologie Ajax (voir la page 12) qui nous a aidés à réduire le temps de réponse et augmenter les performances de notre application. Enfin la bibliothèque « PhpMailer » qui nous a permet d’envoyer d'e-mails facilement et aussi pour éviter tous genres de vulnérabilités d'injection pour spammer. **[9]**

### 1.3. La configuration matérielle et logicielle nécessaire

Les logiciels et les environnements nécessaires pour l’exécution de notre application :

* Un ordinateur doté d’un system d’exploitation (Windows, linux, mac).
* Un navigateur web.
* Une plateforme de développement web (Xamp, Wampserver, easyPHP).

### Implémentation de l’application

Cette partie présente une description des fonctions qu’offre notre application avec des explications de quelques parties de codes importants.

### 2.1. Inscription

L’utilisateur doit d’abord sélectionne le type de compte qui souhaite le créer, après avoir entré ses informations l’application va lancer deux vérifications l’une en côté client et l’autre dans le côté serveur, si ces deux vérifications ont réussi les informations de l’utilisateur seront enregistrées dans la base de données.

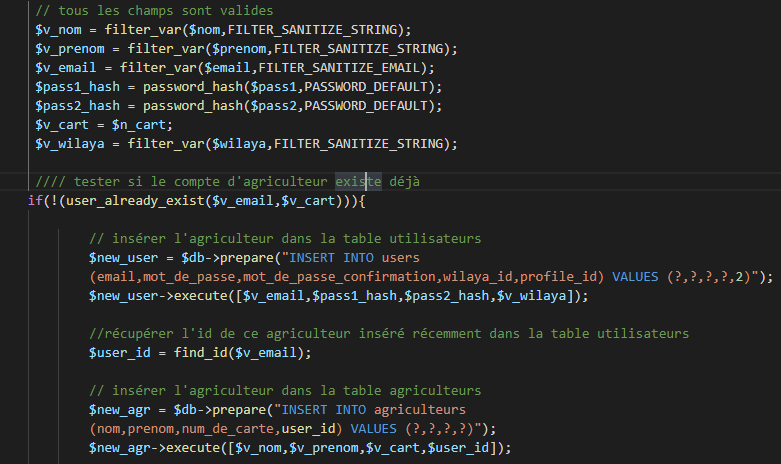


Figure Code d'inscription d'un agriculteur



Figure Code d'inscription d'un office

### 2.2. Connexion

Dans la connexion l’utilisateur qui a déjà inscrit entre ses informations, après la vérification de ces dernières en côté client et serveur l’application va le diriger vers la page convient à leur type (agriculteur ou office).

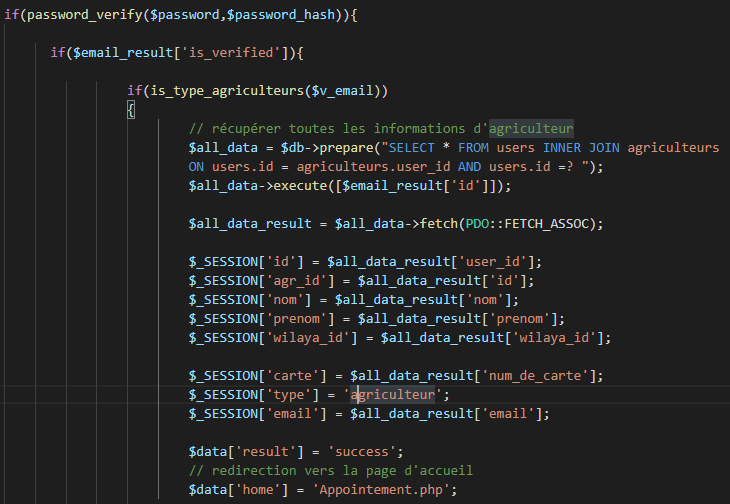


Figure  : code pour la connexion d'un agriculteur

### 2.3. Validation des comptes

Chaque administrateur d’office peut valider les comptes de tous les agriculteurs de sa wilaya et pour les comptes d’offices ils seront validés par des offices responsables.

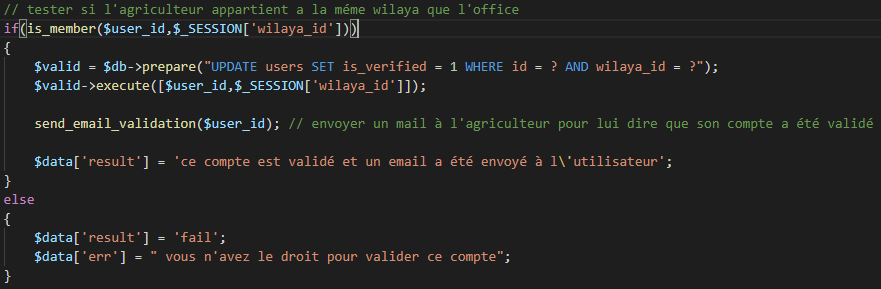


Figure  : code de validation d’un compte agriculteur

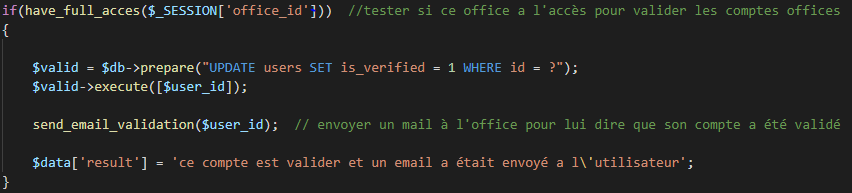


Figure  : code de validation d’un compte office

### 2.4. Les rendez-vous

### 2.4.1. Ajouter un rendez-vous

L’administrateur d’office peut ajouter des rendez-vous pour le dépôt de la récolte, il choisit la date et l’heure du rendez-vous ensuite ces informations seront validées puis ce rendez-vous sera ajouté à notre base de données et désormais les agriculteurs qui appartiennent à la même wilaya peuvent le prendre.

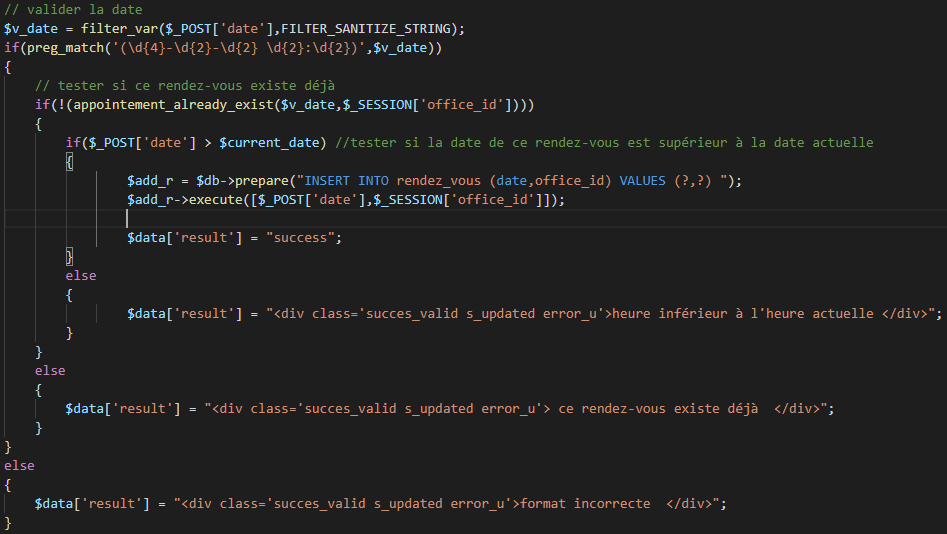


Figure : code d’ajout d'un rendez-vous

### 2.4.2. Prendre un rendez-vous

Chaque agriculteur peut prendre un rendez-vous pour le dépôt de la récolte à condition que ce rendez-vous n’est pas encore pris et ajouté par un office de sa wilaya.

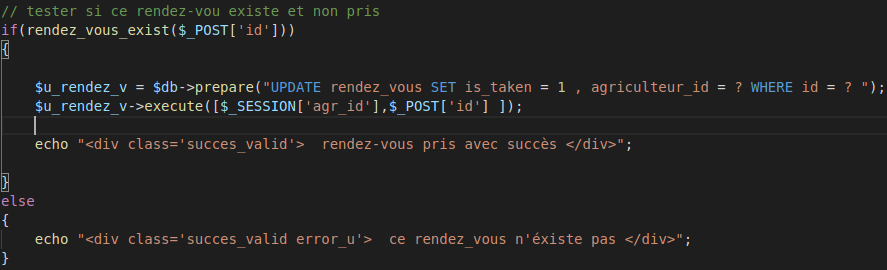


Figure : code pour prendre un rendez-vous

### 2.5. Les récoltes

### 2.5.1 Ajouter une récolte

Chaque administrateur d’office peut ajouter des récoltes lors d’un rendez-vous de dépôt de la récolte d’un agriculteur, il entre toutes les informations concernent cette récolte (poids d’entré, poids de sortie, quantité, qualité..), ces informations seront enregistrées à notre base de données.

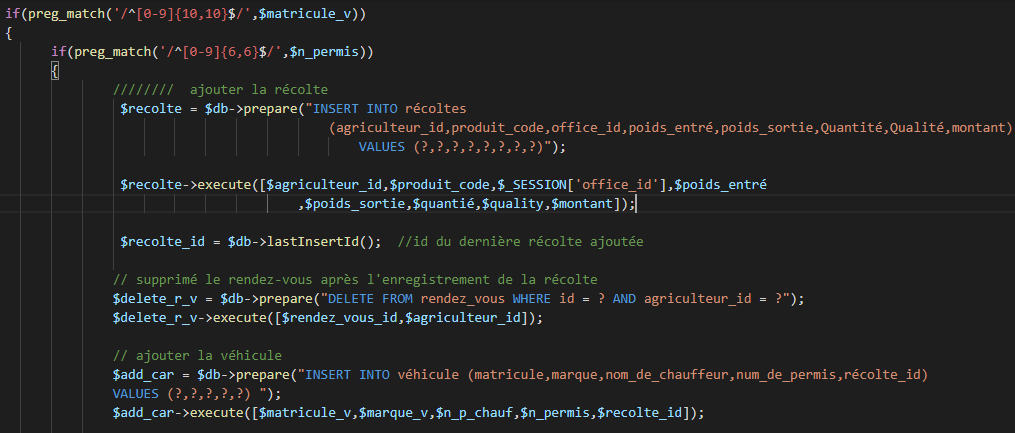


Figure : code de l'ajout d’une récolte

### 2.5.2. Ajouter une facture

Chaque administrateur d’office peut ajouter des factures de vente des récoltes des agriculteurs qui ont au minimum fait un dépôt de la récolte dans cet office, chaque facture regroupe une ou plusieurs récoltes qui n’ont pas fait partie d’une autre facture.

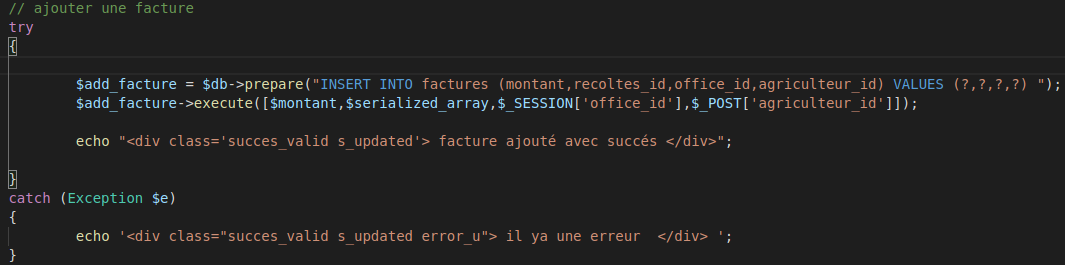


Figure : code d'ajout d'une facture

### 2.6. Statistiques des rendements

Chaque agriculteur ou administrateur d’office peut visualiser toutes ses récoltes soit dans des tableaux qui représentent les informations de chaque récolte ou d’une forme des graphesqui représentent la croissance des récoltes regroupées par mois pendant une année donnée en fonction de la quantité ou du montant.

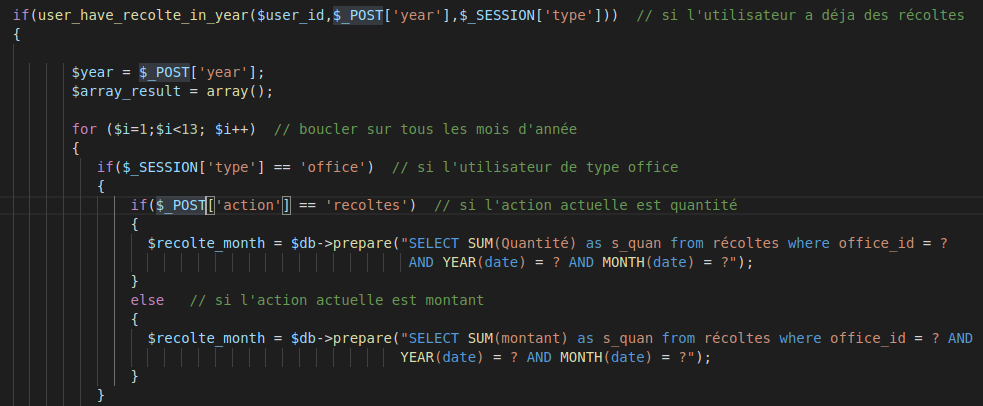


Figure : le code pour récupérer les récoltes d'un office pendant une année donnée regroupées par mois

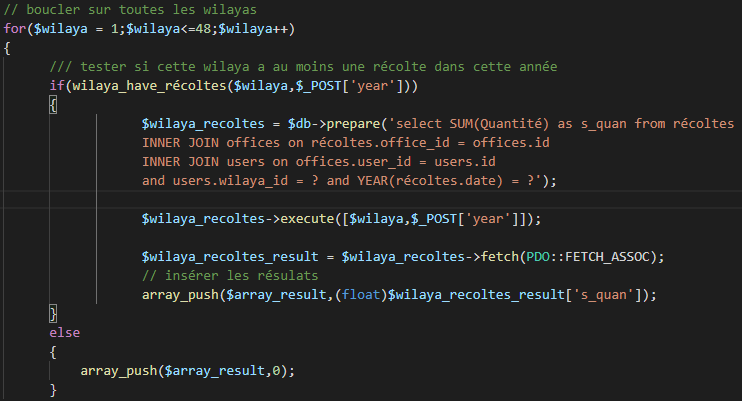


Figure  : le code pour récupérer la quantité des récoltes de toutes les wilayas pendant une année donnée

## Présentation de l’application :

Dans cette partie nous allons présenter les interfaces utilisateur de certaines fonctionnalités de notre application, « allez à l’annexe A pour voir ces interfaces ».

# Conclusion et perspectives

L’objectif de ce projet est de concevoir une application web pour faciliter la gestion de la récolte des offices du blé en Algérie dans lequel il permettra la gestion informatisée de ces derniers et une meilleure transaction entre les offices du blé et les agriculteurs grâce à une interface conviviale et une base de données qui va lier tous les offices de toutes les wilayas d’Algérie et sauvegarder tous les données concernent les récoltes des agriculteurs afin de les rendre accessibles.

Notre application pour le moment n’est disponible que sur la plateforme Web, on pourrait la rendre disponible sur d’autres plateformes (Android, IOS).

Des perspectives d’amélioration sont envisageables, élargir la fonctionnalité de visualisation des statistiques des récoltes et ajouter des nouvelles fonctionnalités (le classement des offices du blé selon leurs récoltes pendant une année d’une wilaya ou de toutes les wilayas d’Algérie).

Des améliorations au niveau de design par ce que notre application à ce moment n’est pas responsive avec les petits écrans (mobile, tablette).

# Références

[1] [*https://fr.wikipedia.org/wiki/Agriculture\_en\_Algérie*](https://fr.wikipedia.org/wiki/Agriculture_en_Algérie)

[2] [*https://www.alticreation.com/difference-developpeur-front-end-et-developpeur-back-end/*](https://www.alticreation.com/difference-developpeur-front-end-et-developpeur-back-end/)

[3] [*https://openclassrooms.com/fr/courses/1916641-dynamisez-vos-sites-web-avec-javascript/1920925-quest-ce-que-lajax*](https://openclassrooms.com/fr/courses/1916641-dynamisez-vos-sites-web-avec-javascript/1920925-quest-ce-que-lajax)

[4] [*https://jquery.com/*](https://jquery.com/)

[5] [*https://parall.ax/products/jspdf*](https://parall.ax/products/jspdf)

[6] [*https://www.chartjs.org/*](https://www.chartjs.org/)

[7] [*https://sweetalert2.github.io/*](https://sweetalert2.github.io/)

[8] [*https://en.wikipedia.org/wiki/XAMPP*](https://en.wikipedia.org/wiki/XAMPP)

[9] [*https://github.com/PHPMailer/PHPMailer*](https://github.com/PHPMailer/PHPMailer)

# Annexe A

## Les interfaces utilisateur

### Les utilisateurs

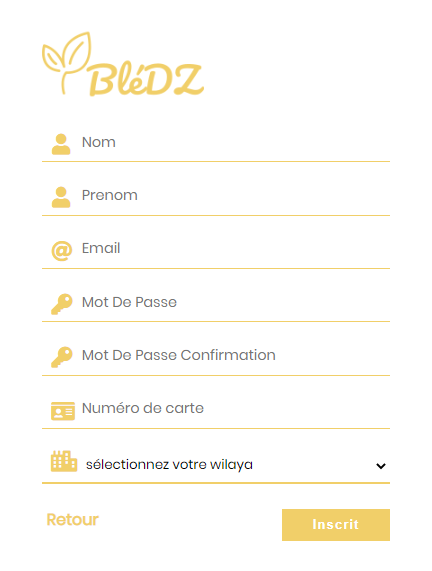
 

Figure : interfaces d'inscription d’un agriculteur et d’un office du blé



Figure : interface de connexion

Ces trois interfaces présentent l’inscription d’un nouvel utilisateur ainsi que la connexion d’un utilisateur qui est déjà inscrit comme décrit dans le chapitre 3(2.1).

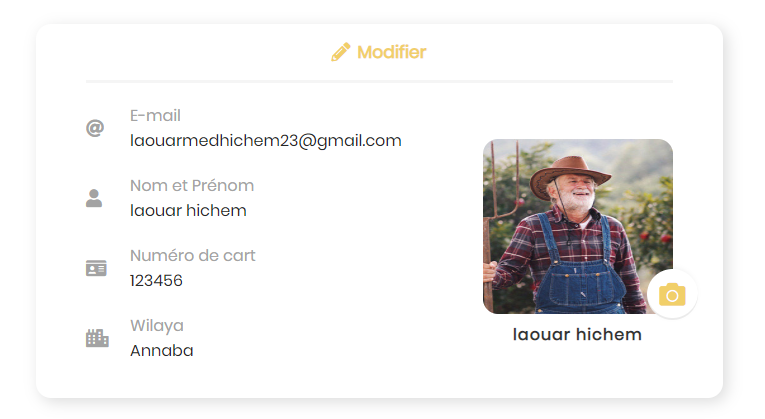


Figure : le profil d'un agriculteur

Cette interface présente le profil d’un agriculteur

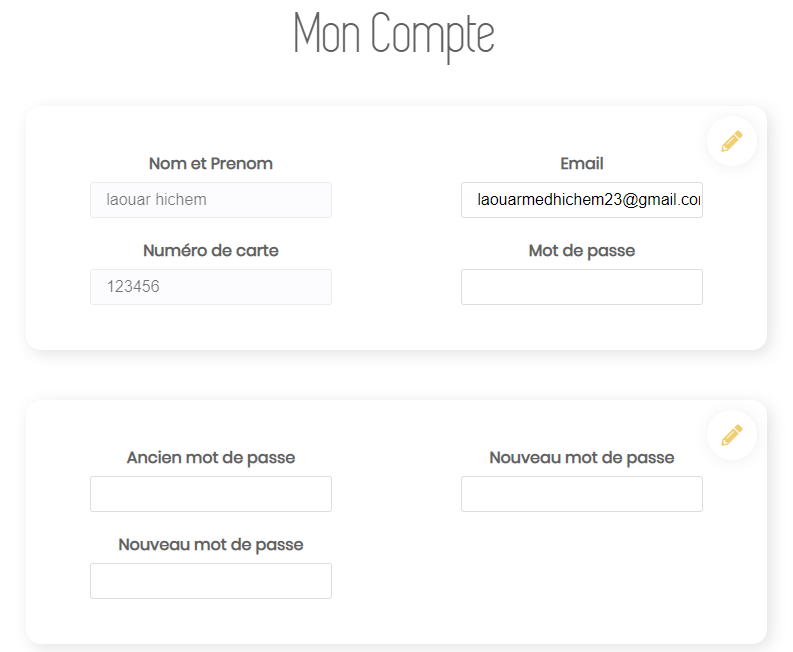


Figure : interface de la modification des informations du compte d'un agriculteur

Cette interface présente la page de modification d’informations du compte d’un agriculteur

### Les rendez-vous

1. 
2. Figure : le formulaire d'ajout d'un rendez-vous

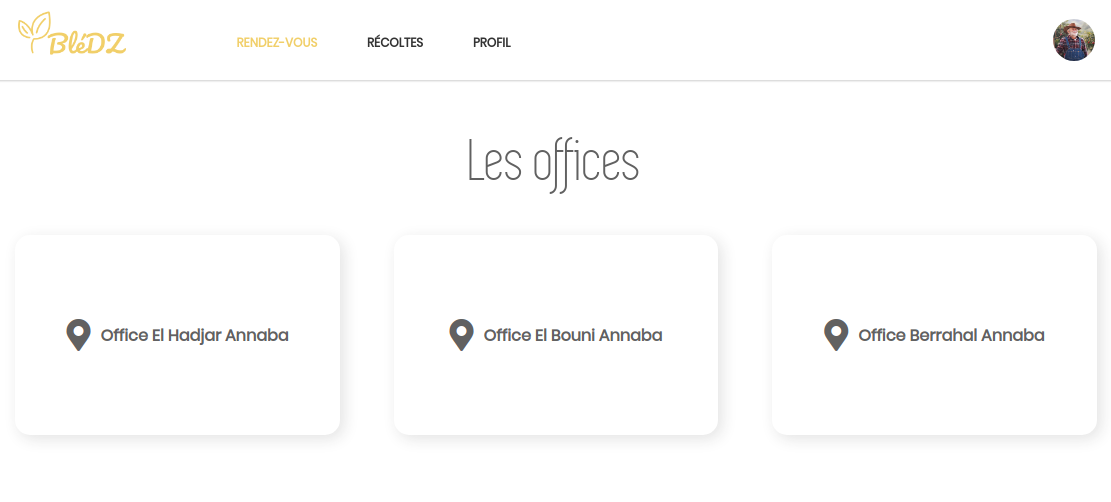


Figure : les offices du blé les plus proches d'un agriculteur

Cette interface présente les offices du blé les plus proches d’un agriculteur.

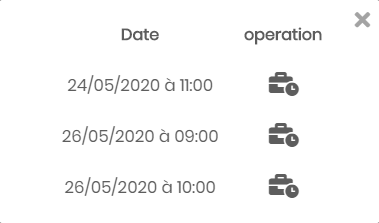


Figure : la liste des rendez-vous disponibles proposés par un office

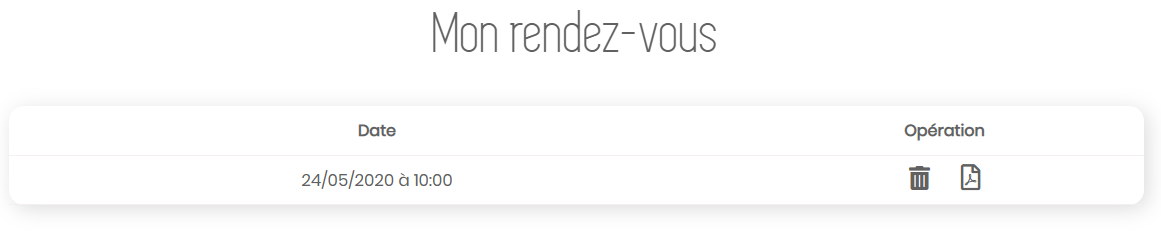


Figure : un rendez-vous pris par un agriculteur

Cette interface présente un rendez-vous de dépôt de la récolte proposé par un office et pris par un agriculteur.



Figure : liste des rendez-vous pris proposé par un office

Cette interface présente la liste des rendez-vous de dépôt de la récolte proposés par un office du blé et pris par des agriculteurs.

### Les récoltes

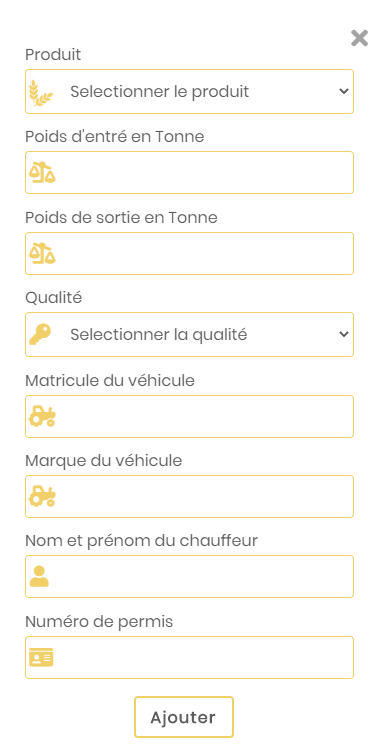


Figure 35 : le formulaire d'ajout d'une récolte



Figure : la liste des récoltes d'un agriculteur

Cette interface présente la liste des récoltes d’un agriculteur



Figure : les récoltes d'un office

Cette interface présente la liste des récoltes d’un office du blé

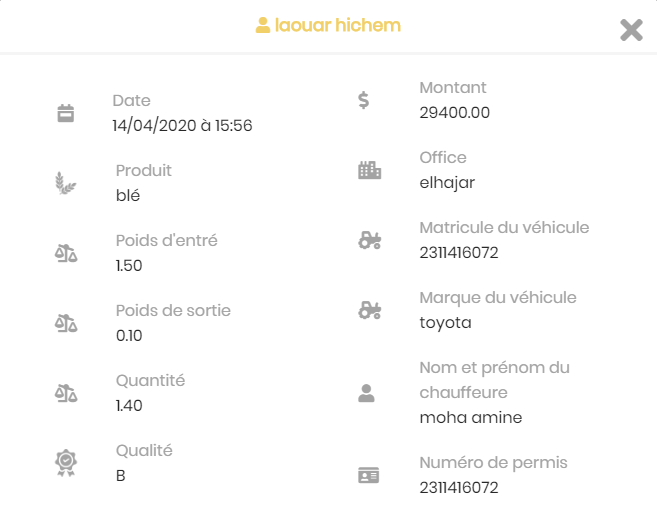


Figure : les informations d’une récolte

Cette interface présente toutes les informations d’une récolte d’un agriculteur



Figure : la liste des récoltes par année d'un office

Cette interface présente la liste des récoltes par année d’un office (ou agriculteur)

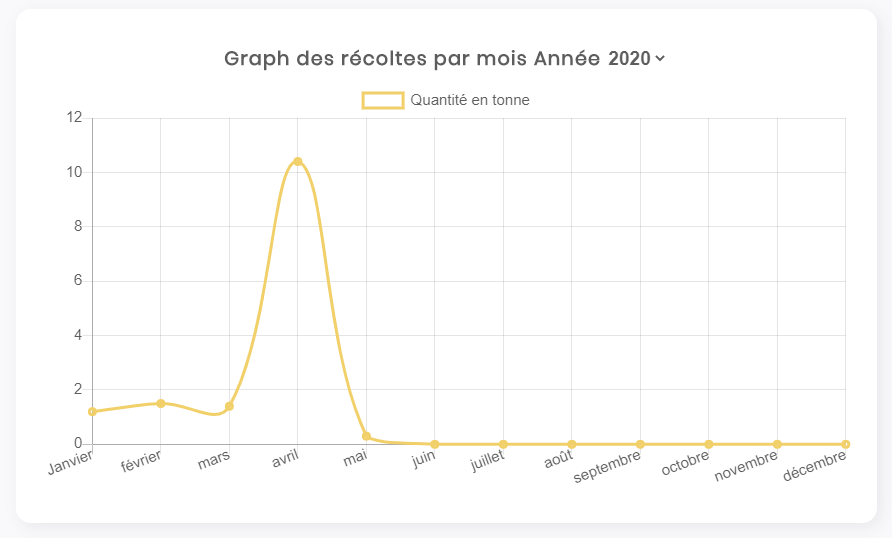


Figure : un graphique des récoltes d'un office par mois pendant l’année 2020

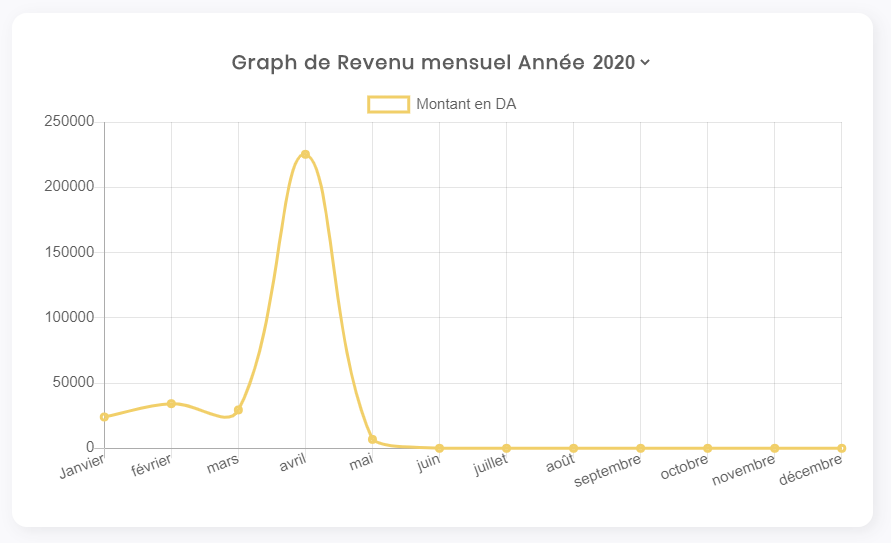


Figure : un graphique du revenu mensuel d’un office pendant l’année 2020

Ces deux dernières interfaces présentent des graphiques de la croissance de la quantité et des revenus mensuelles des récoltes d’un office du blé (ou agriculteur).



Figure : récoltes pour wilaya

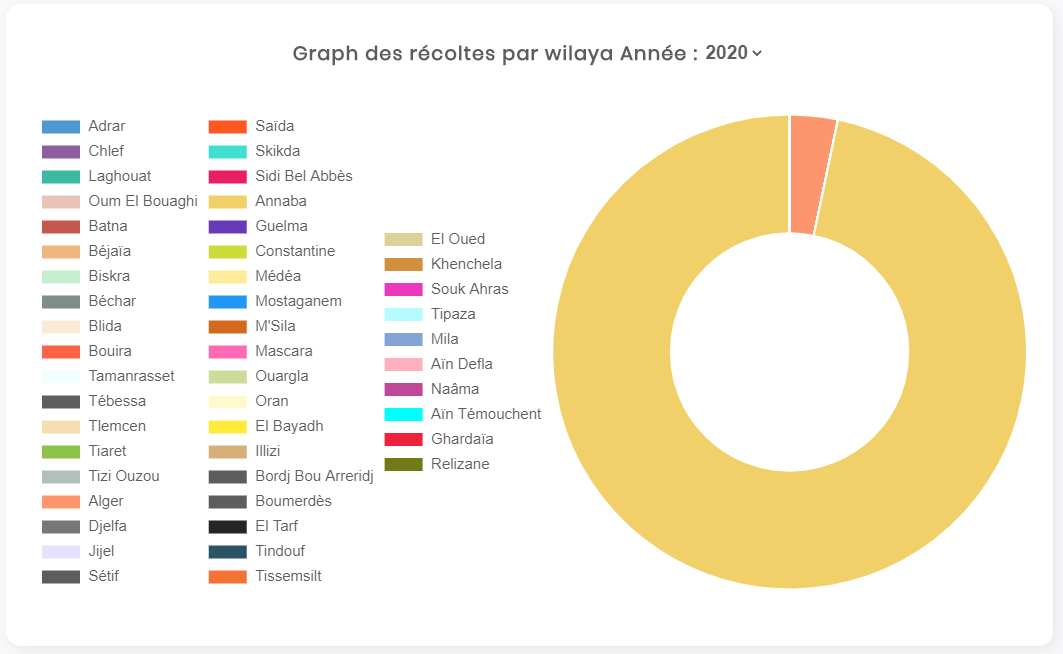


Figure : le graphique de la récolte par wilaya

Cette interface présente la quantité des récoltes de toutes les wilayas d’Algérie en forme d’un graph en anneau.

### Les produits



Figure : le formulaire d'ajout d'un produit

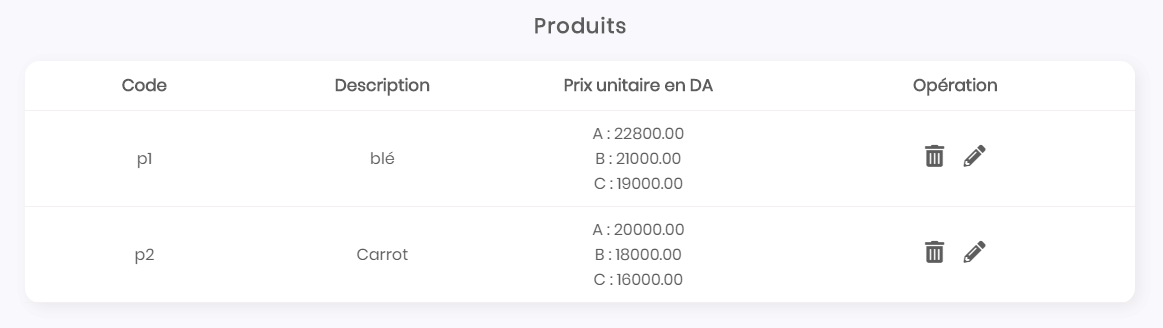


Figure : la liste des produits

Cette interface présente tous les produits ajoutés pour le récolter.

**Résumé**

La transaction entre les agriculteurs et les offices du blé en Algérie pose un problème de communication ainsi que l’accès informatisé aux statistiques des récoltes de ces derniers.

Ce mémoire présente un moyen de rapprocher les agriculteurs aux offices du blé de leurs wilayas.

L’application que nous avons développée permettre de lier tous les offices du blé à travers toutes les wilayas d’Algérie pour faciliter le dépôt de la récolte et de sauvegarder et ordonner tous les données concernant les récoltes des agriculteurs afin de les rendre accessibles.

**Mots clés**. Front End, Back End, Html, CSS, JavaScript, Ajax, Jquery, JsPDF, ChartJs, Php, PhpMailer