|  |  |  |
| --- | --- | --- |
| **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 :** Melle/M. Prénom Nom

**Encadré par :** Pr/Dr/M./Mme Prénom Nom

**Année : 2019-2020**

1. Remerciements

Avant tout, j’aimerai remercier vivement mon encadreur Madame Salima YAHIOUCHE qui m’a suivi et dirigé tout au long de ce projet.

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

# Dédicaces

Les plus brèves possibles

# Table des matières

Table des matières

**Tapez le titre du chapitre (niveau 1)** **1**

Tapez le titre du chapitre (niveau 2) 2

Tapez le titre du chapitre (niveau 3) 3

**Tapez le titre du chapitre (niveau 1)** **4**

Tapez le titre du chapitre (niveau 2) 5

Tapez le titre du chapitre (niveau 3) 6

# Tables des figures

A mettre ici

# 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 meilleure collaboration et transaction avec les agriculteurs, on s'intéresse donc en développement d’un système informatique qui va prendre en charge la gestion informatisée des récolte de ces offices .

## Problématique

La transaction informatisée entre les agriculteurs et les offices du blé en Algérie pose un problème ainsi que l’accès informatisé 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 a les offices du blé nous souhaitons arriver à la meilleure solution informatisé.

## Motivations

Vu le manque de la gestion informatique dans le domaine agricole en Algérie, et l’accès difficile aux statistique des récoltes comment remédier ce manque ? Pourquoi pas ne pas développer un system qui prendra en charge l'informatisation de ce domaine ce qui permettra de faciliter la transaction entre l'agriculteur et l'office du blé dans tous les activités et aussi pour rendre les statistique des rendements agricole accessible à travers tous 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 informatisé des rendez-vous du dépôt de la récolte et aussi la classification des ces dernières selon leurs qualité et de sauvegarder est ordonné les statistique du récolte de chaque agriculteur et office afin de les rendre accessible et aussi la factorisation de ces dernier.

## Contenu du mémoire

* **CHAPITRE 01 :** Dans le premier chapitre nous allons décrire quelques définition nécessaire à la compréhension et la conception de notre application.
* **CHAPITRE 02 :** Dans le deuxième chapitre nous allons présenter la conception générale du system.
* **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

## C’est quoi la 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 (3).

Il existe plusieurs Framework et bibliothèques JavaScript utiliser pour rendre le dévelepomment des interfaces utilisateur simple et plus rapide qu’avant parmi ces bibliothèques (Jquery, React js,..) et des Framework comme (Vue js, Angular js,..).



Figure 1 : les technologies populaires utiliser dans la front end

## C’est quoi le Back-end ?

Lorsque l'on parle de «Back-End», il s'agit la partie du site web qui l’e ne vois 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(3). Elle composé est composé essentiellement par le serveur qui représente l’hébergeur de site web et une base de donnés qui est composé par des tableaux qui contiens des donnés, pour pouvoir accéder et modifies ces données il faut qu’on utilise un langage de programmation comme (php, c#,Java…)

.

Figure : les technologies populaires utiliser dans le back end

## La technologie Ajax

AJAX *(*Asynchrones JavaScript and XML) est une technologie permet de communiquer avec le serveur à l'aide de 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 (7).

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

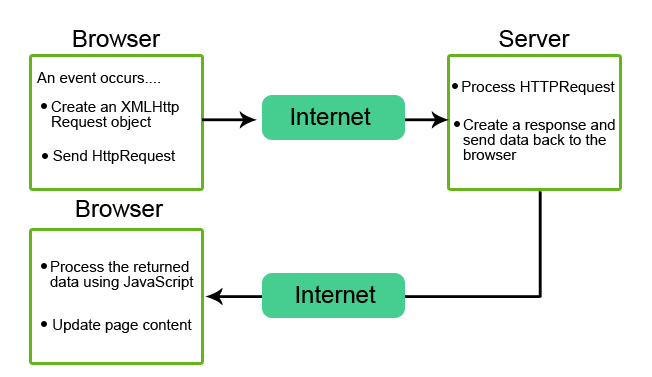


Figure : le mécanisme de travaille de la technologie Ajax

# 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 besoin fonctionnel qui représentent les fonctionnalités que le system doit fournir et les besoin non fonctionnelles qui définissent les caractéristiques du system.

### Besoins fonctionnelles

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 aux administrateurs d’offices d’ajouter des rendez-vous, supprimé des rendez-vous déjà existe et annuler des rendez-vous déjà pris, d’autre coté l’agriculteur peut prendre des rendez-vous offerts par des offices de leur wilaya et aussi annuler des rendez-vous qu’il a 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 établir 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 sélectionner et aussi visualiser 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 des offices responsables peuvent accéder àcette fonctionnalité, elle les 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 fonctionnelles

Notre système doit répondre aux exigences de qualité suivantes :

* **Conviviale :** le système 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 spécifie précédemment.
* **Sécurisé** : le système doit être sécurisé dans lesquelles chaque utilisateur identifié par un identifiant 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 system doit être accessible 24 heures sur 24, 7 jours sur 7et fonctionne dans toutes les cas normale et anormale (Cas d’erreur, entrée avec erreur)
* **Performances:** le system doit être performant dans laquelle tous les tache sont rapide
* **Maintenabilité**: le système doit être facile à modifier pour s'adapte aux nouvelles fonctionnalités (évolution).

## Conception MERISE, UML

### Diagramme de cas d’utilisation

Un diagramme de cas d'utilisation est un diagramme UML utilisés pour donner une vision globale du comportement fonctionnel d’un system, il représente une unité discrète d'interaction entre un utilisateur (Humann ou Machine) et un system. (2)

Le diagramme de cas d’utilisation de notre application :

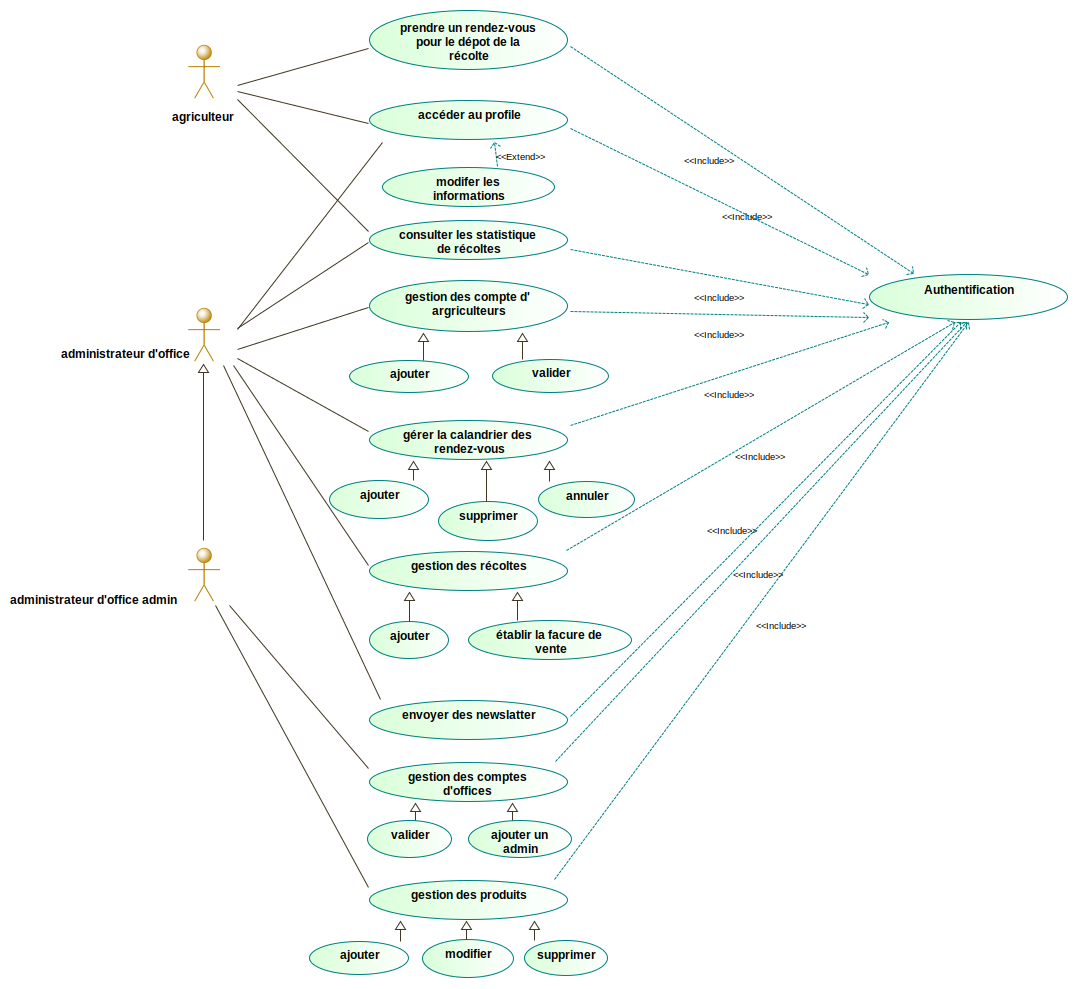


Figure : Diagramme de cas d'utilisation

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 des 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 :

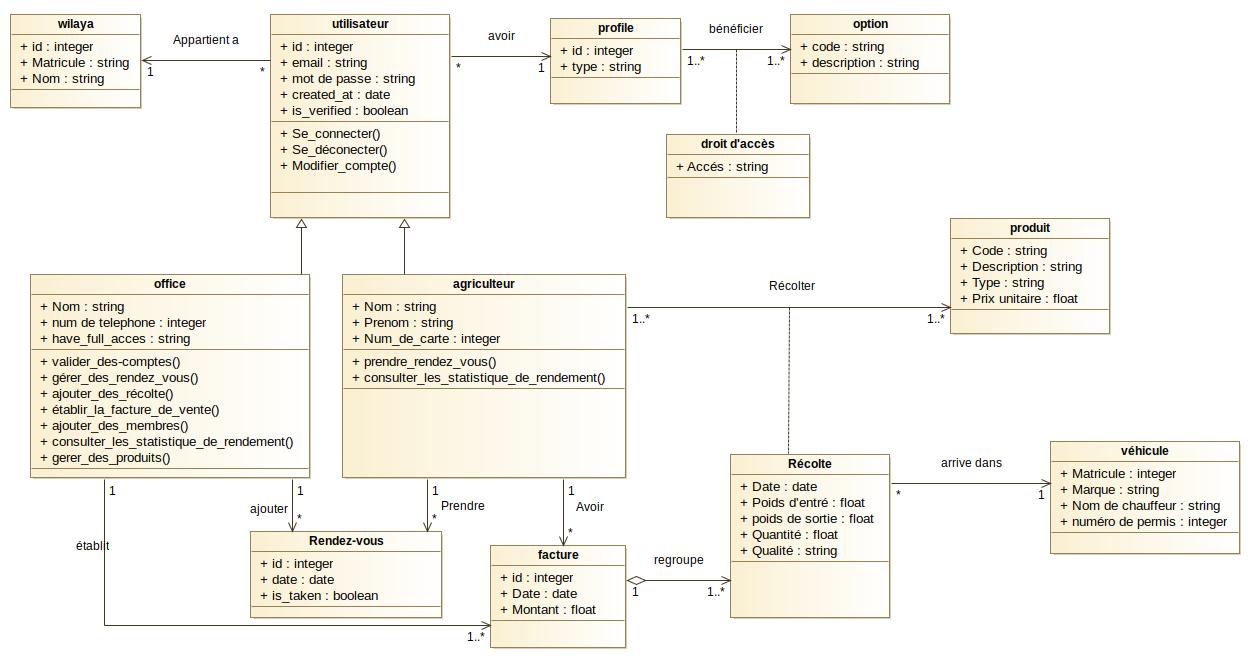


Figure : diagramme de classe

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 (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 au plusieurs produit et dans chaque récolte en garde les informations de ce dernier et aussi de la véhicule utiliser, chaque récolte peut faire partie d’une et une seule facture, cette dernière sont appartiennes 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

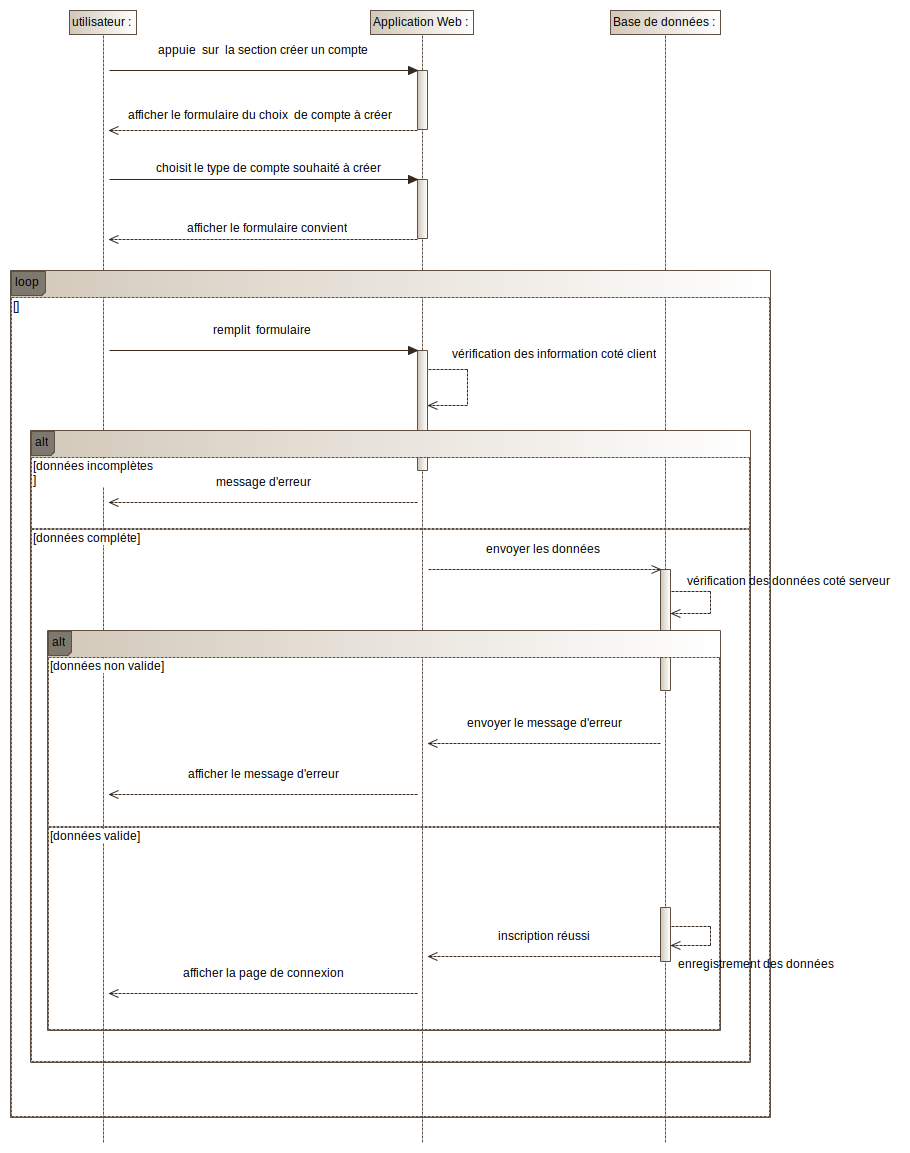


Figure 6 : diagramme de séquence d'inscription

Ce diagramme représente le diagramme de séquence de l’inscription d’un nouvelle utilisateur tous d’abord l’utilisateur choisit le type de compte l’application affiche le formulaire convient après avoir rempli le formulaire convient l’application lance deux vérification la première dans le coté client pour assurer que les informations entrés complète et l’autre dans le coté serveur pour vérifier ces dernières n’existe pas déjà dans notre base de donnés ensuite ses information serons envoyé vers la base de données.

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

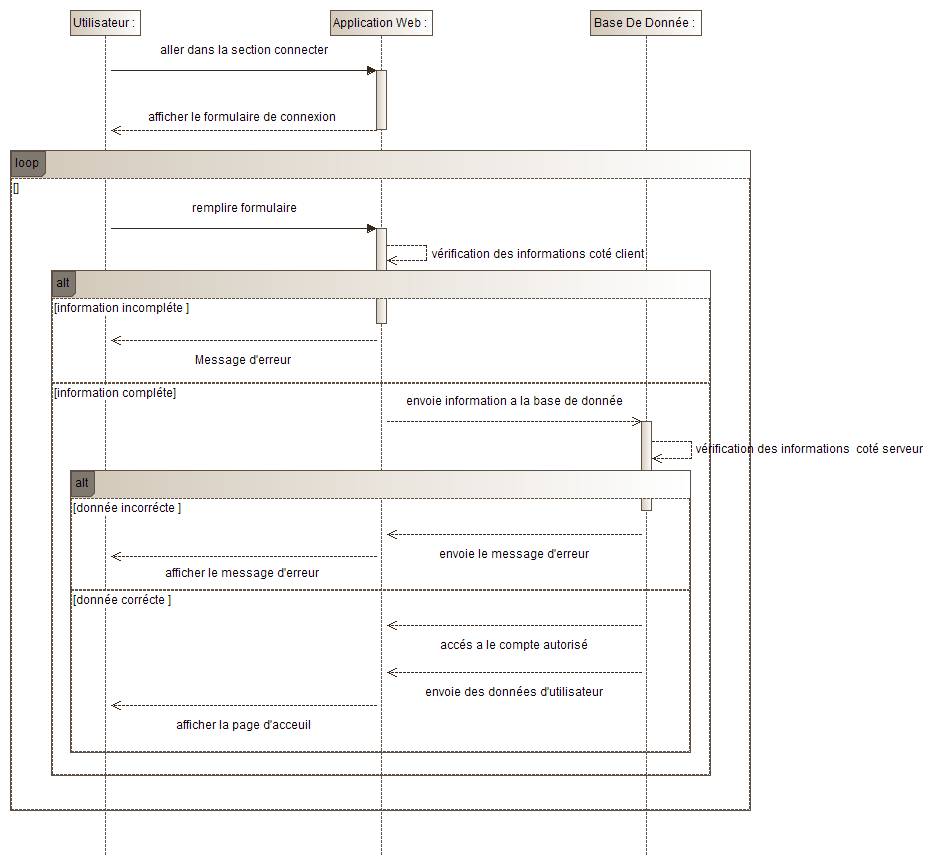
****

Figure 7 : diagramme de séquence de connexion

Dans la connexion l’utilisateur qui est déjà inscrit remplit le formulaire de connexion et comme dans l’inscription l’application lance 2 vérification sauf que la vérification en coté serveur réussi si et seulement si l’information entré sont identique à celle qui est dans la base de données et aussi que le compte de l’utilisateur doit être vérifié pour accéder à la page d’accueil.

### Le diagramme de séquence pour valider un compte

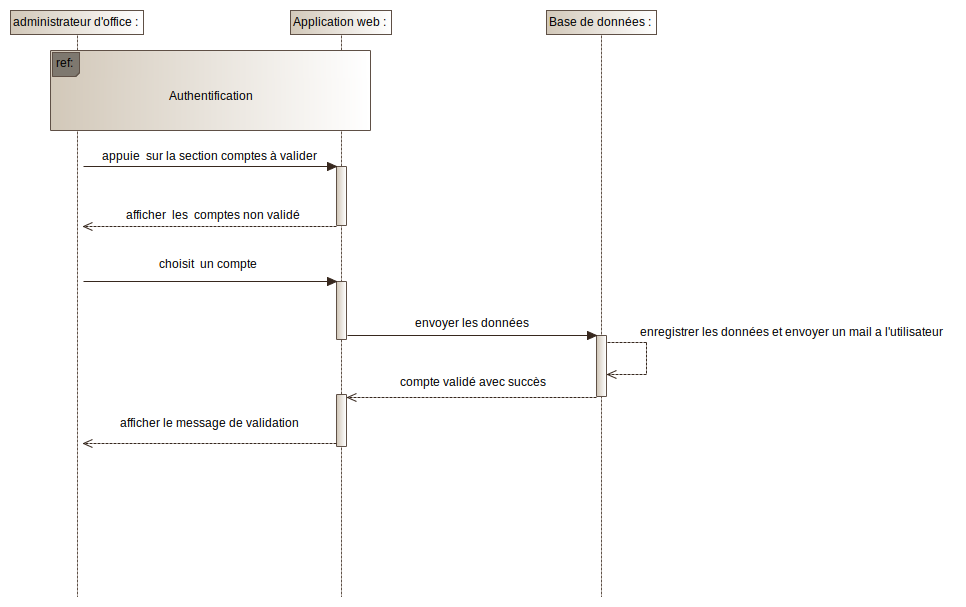
****

Figure 8 : diagramme de séquence de la validation d'un compte

Ce diagramme montre comment l’administrateur d’un office valide le compte d’un agriculteur, il choisit un compte pour le valider après tous l’information seront envoyer à la base de donnée pour modifier ce compte en suite et à l’aide de la bibliothèque PHP MAILER un mail sera envoyé à l’utilisateur concerner pour lui indiquer que son compte est validé.

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

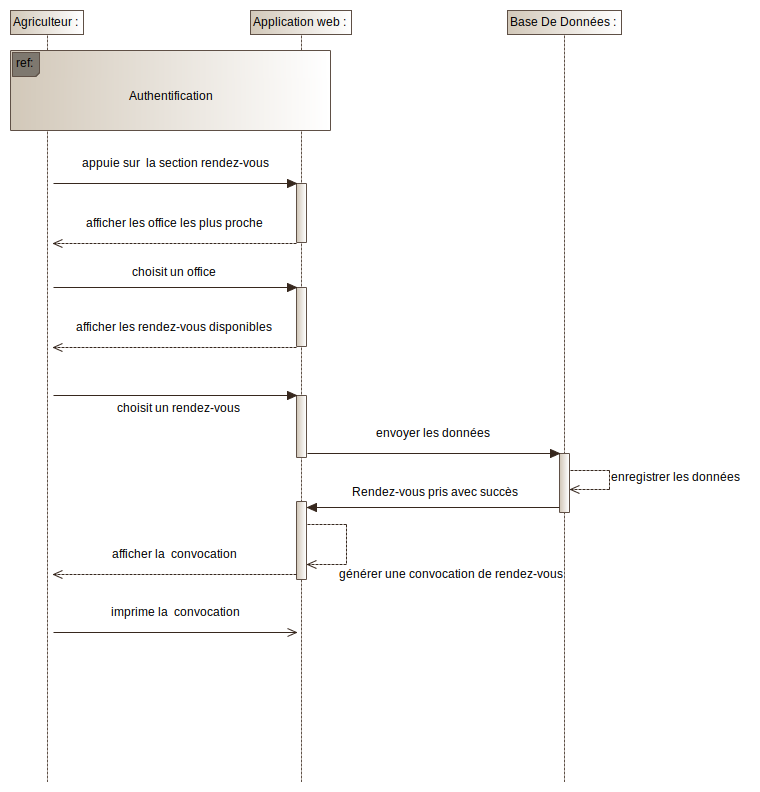


Figure 9 : diagramme de séquence pour prendre un rendez-vous de dépôt de la récolte

Ce diagramme montre comment l’agriculteur prendre un rendez-vous tout d’abord l’agriculteur qui est déjà authentifier appuie dur la section rendez-vous après l’application affiche tous les offices les plus proche il choisit un ensuite l’application affiche les rendez-vous disponible proposé par se office ,après avoir choisi un rendez-vous tous les information de ce rendez-vous seront enregistré dans la base de données , après elle génère un convocation de type PDF contient tous les information à propos de ce rendez-vous.

### Le diagramme de séquence pour ajouter une récolte

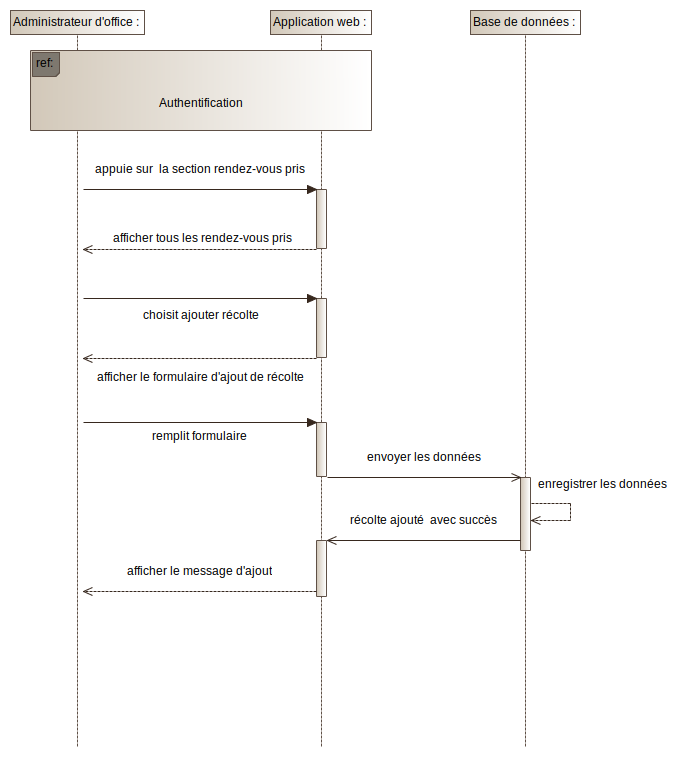


Figure 10 : diagramme de séquence pour ajouter une récolte

Ce diagramme montre comment l’administrateur d’office ajout un récoltes, leur d’un rendez-vous de dépôt de la récolte d’un agriculteur, l’administrateur appuie sur la section rendez-vous pris pour ajouter le récolte, après avoir remplit le formulaire d’ajout de récolte et la vérification des informations, ces information seront enregistré dans notre base de données.

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

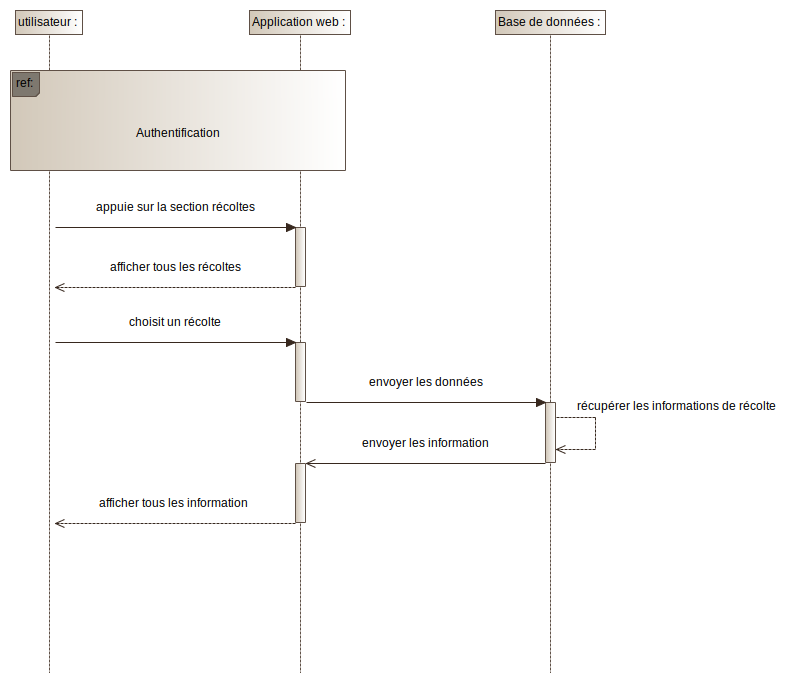


Figure 11 : diagramme de séquence pour voir les informations d’une récolte

Ce diagramme montre comment chaque utilisateur peut voir les informations d’une récolte qui lui appartienne après avoir consulté la section récoltes et choisit un récolte tous les informations de ce récolte seront afficher.

## Architecture fonctionnelle de l’application

Expliquer dans un schéma toutes les fonctionnalités offertes par votre application en montrant le flux de contrôle et le flux de données et en insistant sur les entrées et les sorties.

# Chapitre 3 : Implémentation et présentation de votre 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 différent technologies utilisé pour développer notre application et la configuration matérielle et logicielle nécessaire pour l ‘exécution de cette application.

### 1.1. Technologies utiliser 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 Stylisées) 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'internaute par le navigateur 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 généré des documents PDF en utilisant des informations, donner par l’utilisateur (dans notre cas on l’utilise pour générer des convocations des rendez-vous et des facture) (5)

**ChartJs** : c’est une bibliothèque JavaScript qui génère des défirent type des graphes (line, battent, circulaire..) en basent sur des donnés stocké dans la base des données (6)

### 1.2. Technologies utiliser dans la front-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 composé essentiellement  par un serveur Apache HTTP, la base de données MariaDB et PhpMyadmin pour l’administration des bases de données.

Nous avons aussi utilisé.

Nous avons également utilisé la bibliothèque PHP php mailer qui nous permettre d’envoyer d'e-mails facilement et aussi pour éviter tous genre de vulnérabilités d'injection pour spammer.(8)

### 1.3. La configuration matérielle et logicielle nécessaire

Les logiciel est les environnements nécessaire 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 représente une description des fonctions qu’offre notre application avec des explications de quelques fragments de codes importants.

### 2.1. Inscription

L’utilisateur doit d’abord sélectionner le type de compte qui souhaite le créer (agriculteur ou office)

Puis il entre ces informations nécessaire après l’application va faire une vérification en coté client pour assurer que les informations entré respecte les normes et une vérification coté serveur pour vérifier si le compte est déjà existe

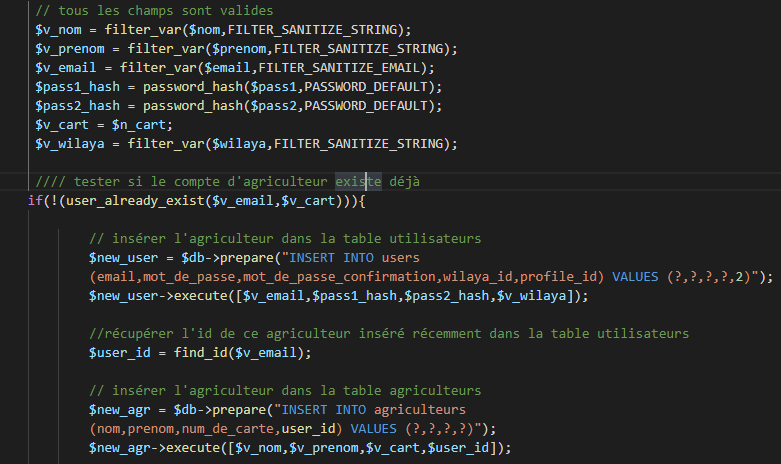


Figure 12 Code d'inscription d'un agriculteur



Figure 13 Code d'inscription d'un office

### 2.2. Connexion

L’utilisateur qui est déjà un compte validé entre ces information, après la vérification de ces dernières en coté client et serveur l’application va le diriger vers le page convient a leur type (agriculteur ou office).

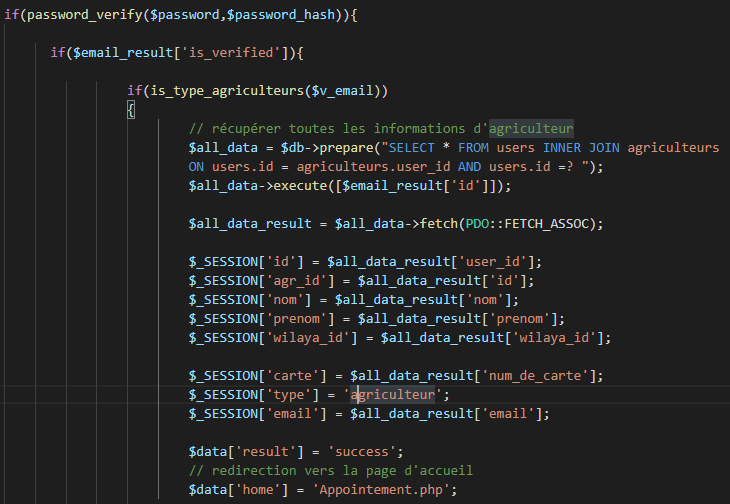


Figure 14 : code pour la connexion d'un agriculteur

### 2.3. Validation des comptes

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

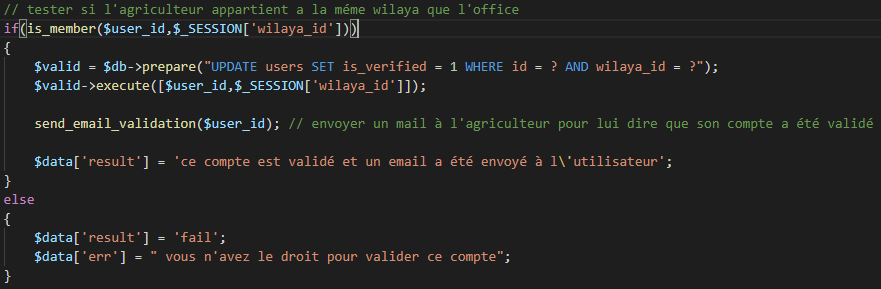


Figure 15 : code de validation d’un compte agriculteur

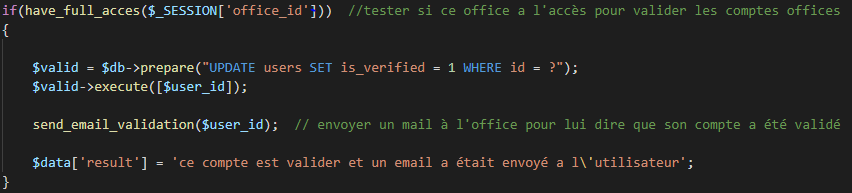


Figure 16 : code de validation d’un compte office

### 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 information seront valider puis ce rendez-vous sera ajouté à notre base de et désormais les agriculteurs de la même wilaya que ce office peuvent le perdre

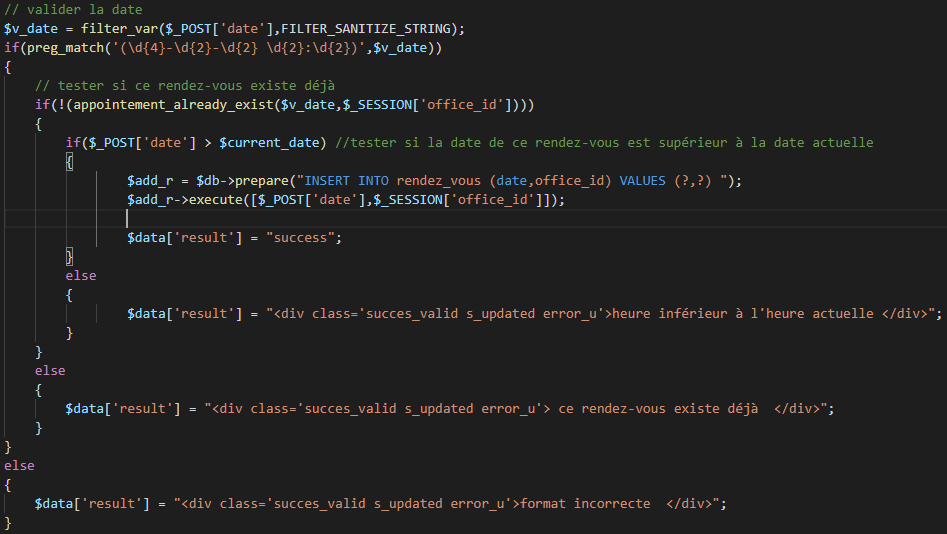


Figure 17 : code pour l'ajout d'un rendez-vous

### 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 est n’pas encore pris et ajouté par un office de la même wilaya que lui.

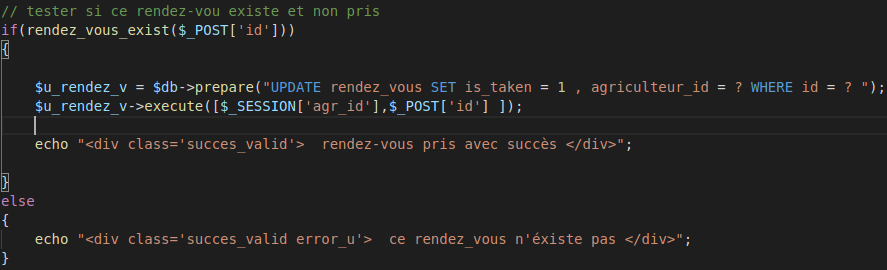


Figure 18 : code pour prendre un rendez-vous

### 2.5. Les récoltes

### 2.5.1 Ajouter une récolte

Chaque administrateur d’office peut ajouter des récoltes leur de rendez-vous de dépôt de la récolte d’un agriculteur, il entre tous l’information concerne ce récolte (poids d’entré, poids de sortie, quantité, qualité..) puis en vérifie la cohérence de ces information et on l’ajout a notre base de données

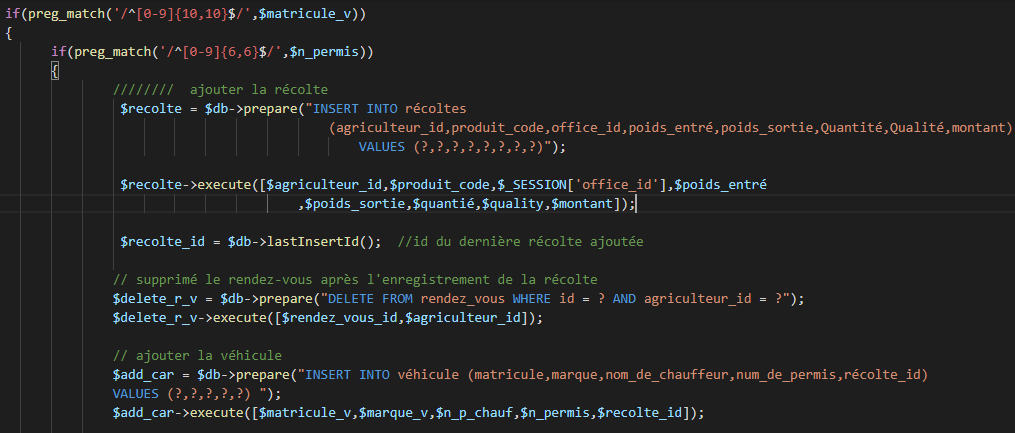


Figure 19 : code de l'ajout d’une récolte

### 2.5.2. Ajouter une facture

Chaque administrateur d’office peut ajouter des factures de rendement des agriculteurs qui ont au minimum fait un dépôt de récoltes dans se office chaque facture regroupe un ou plusieurs récoltes qui sont pas fait partie d’une autre facture

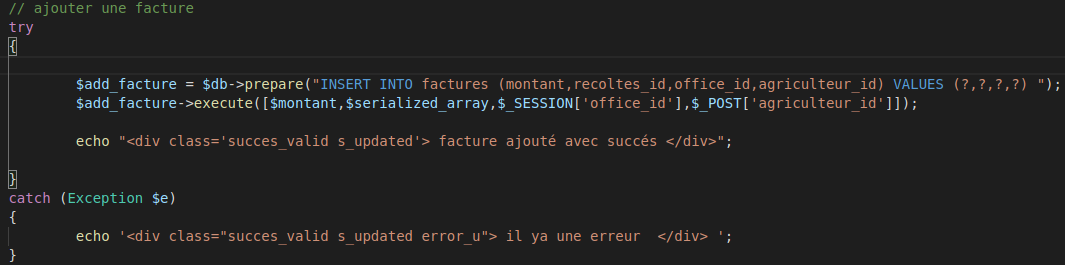


Figure 20 : code d'ajout d'une facture

### 2.6. Statistiques des rendements

Chaque agriculteur ou administrateur d’office peut visualiser tous les récoltes qui ont fait partie soit dans des tableaux qui représente tous les information de chaque récolte ou d’une forme des graphesqui représentent la croissance de ces récoltes regrouper par mois pendant une année donné en fonction du quantité au du montant .

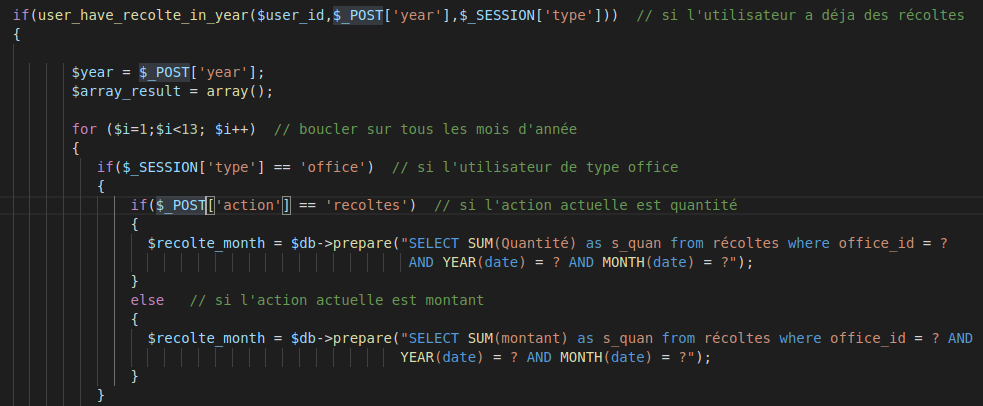


Figure 21 : le code pour récupérer les récoltes d'un office pendant une année donné regroupé par mois

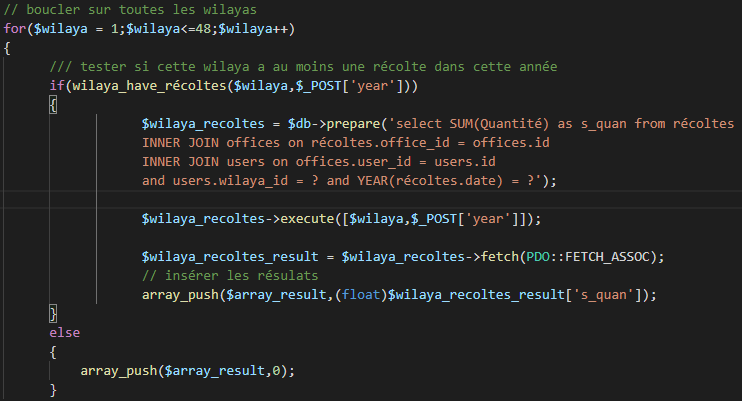


Figure 22 : le code pour récupérer la quantité des récoltes de tous les wilayas pendant une année donnée

## Présentation de l’application :

Cette partie peut être déplacée en annexe si elle nécessite beaucoup d’espace. Elle comporte des copies d’écran décrivant quelques scénarii d’exécution de votre application accompagnées d’explications. /// en annexe screenshot of all ui

# Conclusion et perspectives

L’objectif de ce projet est de concevoir, une application web pour faciliter la gestion de la récoltes des offices du blé en Algérie dans lequel il permettra la gestion informatisée de ces derniers et permettra une meilleure transaction entre les offices 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 statistique des récoltes et ajouter des nouvelles fonctions (classement des récoltes des offices pendant une année d’une wilaya ou de tous 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

Chaque référence doit avoir un numéro unique qui est utilisé chaque fois que la référence est citée dans le texte. Par exemple, « le système Android [5] a été conçu… ».

Pour un livre, vous devez donner les noms des auteurs, le titre, l’éditeur, le numéro d’édition (si disponible) et l’année d’édition. Exemple :

[1] A. Aho, R. Sethi, J.Ullman, « Compilateurs : principes, techniques et outils », édition Dunod, 2000.

Pour un article, vous devez donner les auteurs, le titre, le nom du journal, le volume, le numéro, les pages (si disponible) et l’année. Exemple :

[2] C. A. R. Hoare, « Communicating sequential processes », Communications of the ACM 21 (8): 666–677, 1978.

Pour une thèse de doctorat ou de magister, vous devez donner l’auteur, le titre, l’université de soutenance et l’année de soutenance. Il n’est pas conseillé de mettre un mémoire de licence ou de master comme référence à moins que la référence ne soit disponible sur le net, le cas échéant vous donnez l’adresse de téléchargement. Exemple,

[3] J. Dupont, « Sur les systèmes multi-agents……. », thèse soutenue à l’université X, 2007.

Pour les sites Web, il faut, de préférence, préciser le nom de l’auteur si c’est indiqué ou à défaut le nom de l’organisation. Il faut indiquer le titre de la page ainsi que son adresse et la date de consultation. Il ne faut pas mettre une référence vers un forum, une page facebook, etc. Exemple :

[4] Documentation JavaFX, Oracle, http://www.oracle.com/...., date de consultation : 13 mai 2013.

**Résumé**

[Le résumé ne doit pas excéder une douzaine de lignes. Il doit être écrit de sorte à permettre aux lecteurs de déterminer rapidement si le mémoire l’intéresse ou pas. Pour cela, il doit donner un bref aperçu sur le sujet, la problématique ainsi qu’un aperçu de ce qui a été réalisé. Le résumé ne comporte pas de sous-titre.]

Le résumé figure au verso de la dernière page

**Mots clés**. [Mot clé 1, mot clé 2, …]