|  |  |  |
| --- | --- | --- |
| **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

**Important : à partir de la première page de l’introduction et jusqu’à la dernière page de la conclusion de votre mémoire, le nombre de pages ne doit ni dépasser 30 pages et ni être au-dessous de 20 pages. Les annexes ne peuvent occuper plus de 10 pages supplémentaires. Vous devez respecter ce format de fichier, en l’occurrence,**

* **la taille de la police (12pts) et le type de police (times new roman) :**
* **Les marges doivent être 1cm (ne les modifiez pas) sauf pour la gauche (2cm)**
* **L’interligne est 1.5**
* **Le texte doit être justifié**
* **L’espacement des paragraphes est le suivant : 0pt avant, 6pt après**
* **Utilisez de préférence les styles déjà défini dans ce document : Normal, UneLégende, GrandTitre (pour les chapitre), Titre 1 et Titre 2.**
* **Dans la page de garde, les éléments en rouge doivent être remplacés par les éléments vous correspondant. Il ne faut pas modifier le reste**

## 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, 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 de ces offices .

## Problématique

Poser explicitement le problème

Dire en quoi les solutions actuelles ne sont pas satisfaisantes face à ce problème

La transaction informatisée entre les agriculteurs et les offices du blé en Algérie pose un problème ainsi que l’accès informatisé à la statistique des récoltes de ces derniers. Dans le but de remédier ce problèmes et de rapprocher les agriculteurs a les offices du blé nous souhaitons mettre en ouvre un system informatique qui géré tous ces problèmes

## Motivations

Citer les facteurs motivants qui nous poussent à développer une solution à la problématique

Vu le manque de la gestion informatique dans le domaine agricole, pour surmonter ce problème nous allons développer un system qui prendre en charge l'informatisation de ce domaine ce qui permettra de faciliter la transaction entres l'agriculteur et l'office du blé dans tous les activités et aussi pour rendre les statistique des rendement agricole accessible a travers tous les wilayas d’Algérie.

## Objectifs

l’objectif de ce projet est de faciliter la gestion des office du blé en Algérie dans lesquelles il permettra la gestion informatisé des rendez-vous du dépôt de récolte et aussi la classification des récoltes selon leurs qualité et de sauvegarder est ordonné les statistique de rendement de chaque agriculteur afin de rendre accessible et aussi la factorisation des rendement et permettra en outre la gestion du compte d’utilisateur grâce un système d’authentification qui permet d’assurer la sureté de ces ressources et leur fiabilité.

## Contenu du mémoire

Décrire le contenu de chaque chapitre

# Chapitre 1 : [Votre premier chapitre]

Les chapitres de la théorie ne doivent présenter que ce qui est nécessaire à la compréhension et la conception de votre application. Il faut privilégier la synthèse et surtout ne pas faire du copier/coller qui sera considéré comme un plagiat. Dans le cas où vous vous basez sur un travail d’autrui, il faut mettre la référence nécessaire.

Un chapitre commence par un résumé de trois à quatre lignes. Il se termine par une petite conclusion qui introduit et justifie la présence du prochain chapitre.

Enfin, il est préférable d’avoir un seul chapitre théorique vu la limite du nombre de pages.

## Généralité sur les systèmes d’exploitation mobiles

Un système d’exploitation mobile est…

### Aperçu sur les systèmes d’exploitation existants

Il n’est pas conseillé d’aller au-delà de deux niveaux de titres.

Toutes les figures doivent avoir une légende composée d’un numéro (unique) de figure et une description comme le montre la figure 1. Le numéro de la figure doit être cité dans le texte comme c’est le cas à la fin de la phrase précédente.

Mémoire

Unité de traitement

Entrées

Sorties

Processeur

Figure 1 : architecture d’un système

# 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 les diagrammes appropriés pour les différentes modélisations.

## 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 fonctionnalités permettre à l’utilisateur d’accéder à son compte pour utiliser les fonctionnalités offert.
* **La gestion des comptes des agriculteurs** : seul les administrateurs des offices permettront de valider des comptes des agriculteurs et aussi ajouter des nouvelles comptes.
* **La gestion des comptes des offices** : seul les administrateurs des offices responsables qui sont déjà des administrateurs d’office permettre de valider les comptes des offices et ajouter des offices responsables
* **La gestion Les rendez-vous**: cette fonctionnalité permettre les administrateurs d’offices d’ajouter des rendez-vous, supprimé des rendez-vous déjà existe et annuler des rendez-vous déjà prisse d’autre coté l’agriculteur peut prendre des rendez-vous offert par des offices de leur wilaya et aussi annuler des rendez-vous déjà prise.
* **La gestion des récoltes :** cette fonctionnalité permettre les administrateurs d’offices d’ajouter des récoltes suite à des rendez-vous prise par les agriculteurs et aussi établir des factures de vente.
* **consulter les statique des rendements**: cette fonctionnalité permettre aux utilisateurs de visualiser leurs rendement triés par un ordre sélectionner et aussi visualiser des graphes qui représentent la croissance des rendements par mois pendant chaque année.
* **La gestion du profil :** permettre aux utilisateurs de changer leurs informations personnelles dès qu’ils souhaitent.
* **La gestion des produits :** cette fonctionnalité permettre d’ajouter des nouveaux produits, modifier ou supprimé des produits déjà existeseule les administrateurs d’offices responsables accéder à cette fonctionnalité.
* **Envoyer des newsletters :** cette fonctionnalité permettre les administrateurs d’offices d’envoyer des lettres d’actualité et des nouveautés a les emails des personnes qui ont déjà abonnée à la newsletter.

### Besoins non fonctionnelles

Notre système doit répondre aux exigences de qualité suivantes :

* **Conviviale :** le system doit être facile à utiliser. En effet, les interfaces utilisateur doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur
* **Formulaires simples :** Les formulaires seront donc particulièrement soignés pour ne pas rebuter l’utilisateur
* **Valide :** le system doit être rempli exactement les besoin spécifie précédemment.
* **Sécurisé** : le system doit être sécurisé dans lesquelles chaque utilisateur identifié par un identifiant et un mot de passe, Chaque utilisateur ne peut pas voir les informations personnelles des autres utilisateurs, Chaque type d’utilisateur peut accès 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) au en cas de réparation des erreurs.

## 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. (wipikidia )

Le diagramme de cas d’utilisation de notre application est :

// picture

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

**L’agriculteur** il peut prendre un rendez-vous pour le dépôt de la récolte

**L’administrateur d’office et l’administrateur d’office responsable** permettront de la gestion des comptes des agriculteurs, la gestion des rendez-vous, la gestion des récoltes.

Seule l’administrateur d’office responsable permettre de la gestion des comptes des offices et aussi la gestion des produit.

Tous les utilisateurs permettront de modifier leurs profile et de consulter les statistique des rendements de façon que l’agriculteur peut voir seulement leur récoltes et l’administrateur de l’office peut voir les récoltes de tous les agriculteurs de sa wilaya.

### 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 :

La classe principale de notre diagramme est l’utilisateur, il peut être un agriculteur au un office, il appartient à une wilaya et avoir un profile, 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 ajouté par des offices, il peut récolter un au plusieurs produit et dans chaque récolte en garde les informations de ce récolte et aussi de la véhicule utiliser, chaque récolte peut fait partie d’un et un seule facture, cette dernière est appartenu 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 suivants représentent les diagrammes de séquence de notre application :

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

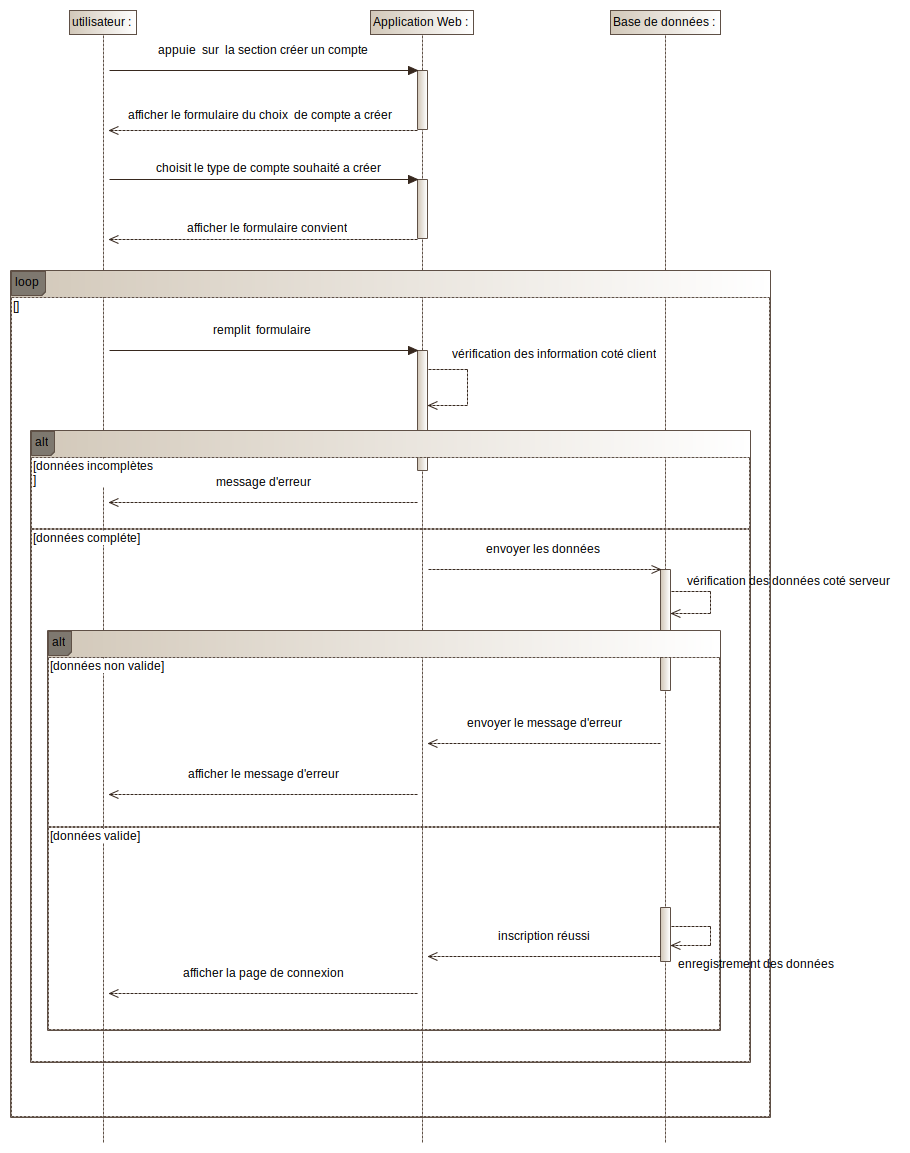
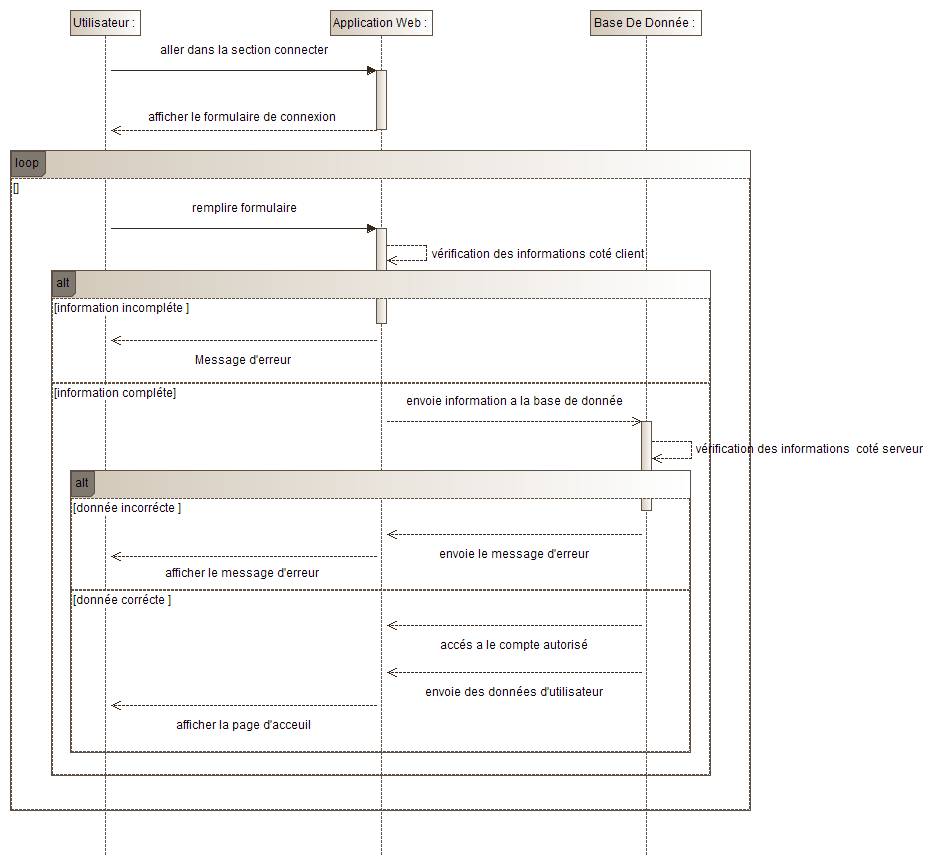
****

Figure : diagramme de séquence d'inscription

Ce diagramme représente le diagramme de séquence de l’inscription d’un nouvelle utilisateur tous d’abord l’utilisateur choisi le type de compte l’application affiche le formulaire convient après le remplissions de cette formulaire l’application lance deux vérification la première dans le coté client pour assurer que les information entré cohérent et complète et l’autre dans le coté serveur pour vérifier que les information entré 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

****

Dans la connexion l’utilisateur qui est déjà inscrit remplier 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.

### Le diagramme de séquence pour valider un compte

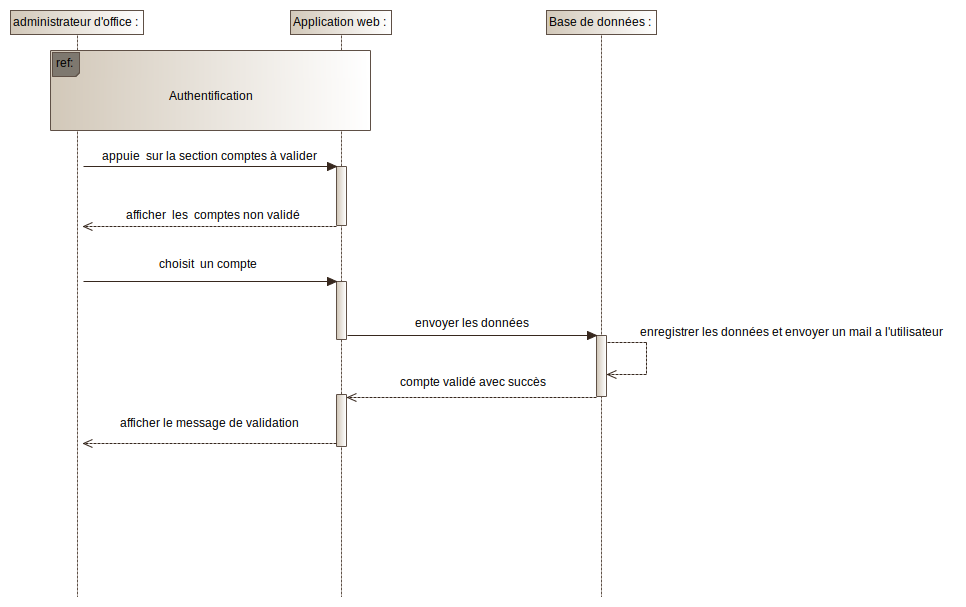
****

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

Ce diagramme montre comment l’administrateur d’un office valider 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 envoyer à l’utilisateur concerner pour lui indiquer que son compte est valider.

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

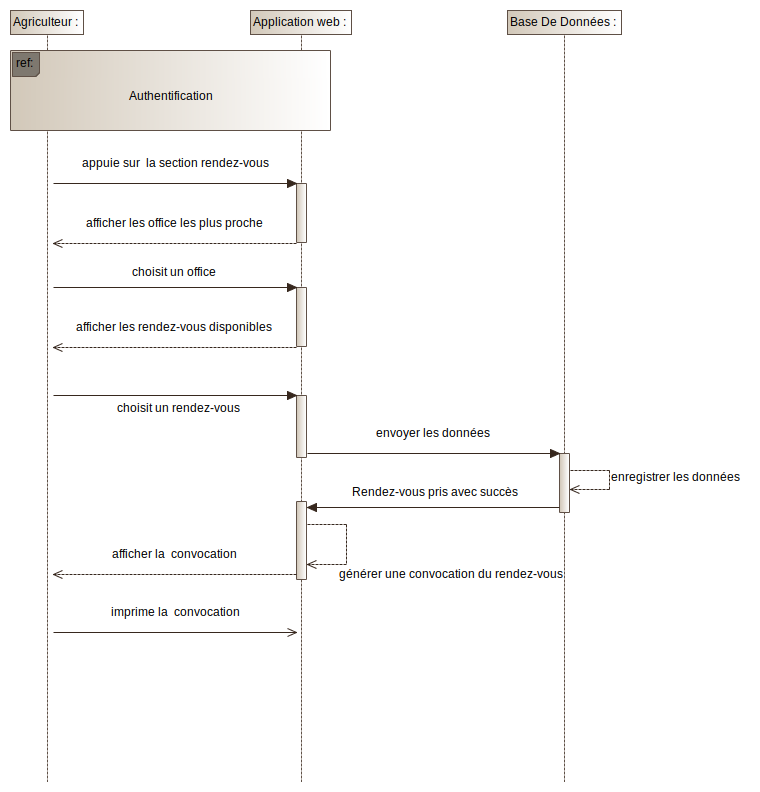
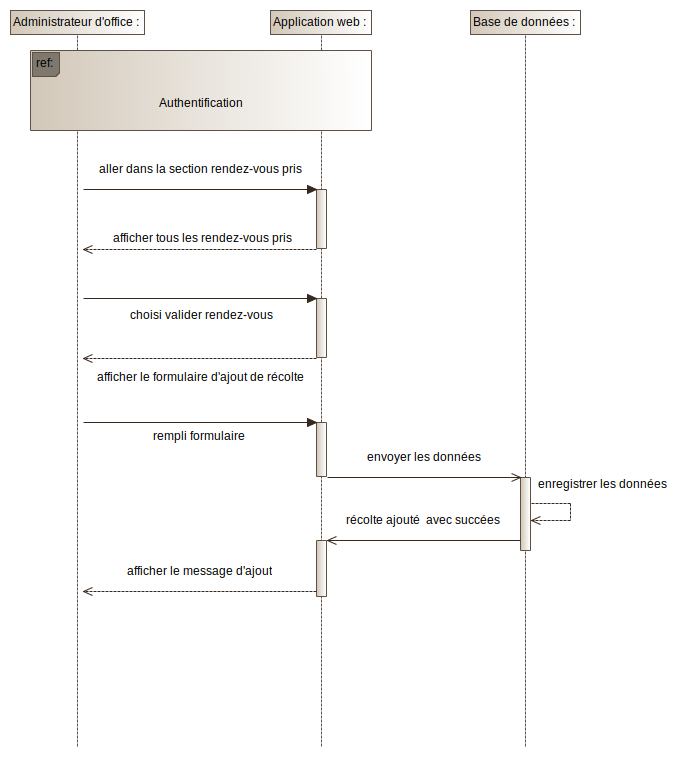


Figure : diagramme séquence pour prendre un rendez-vous

Ce diagramme montre comment l’agriculteur puise prendre un rendez-vous tout d’abord l’agriculteur qui est déjà authentifier aller à 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 à la base de données ,pour l’enregistré après elle génère un convocation de type PDF contient tous les information à propos de ce rendez-vous.

### Le diagramme de séquence pour ajouter un récolte



Ce diagramme montre comment l’administrateur ajouter un récoltes , leurs du rendez-vous de dépôt de la récolte d’un agriculteur , l’administrateur aller à la section rendez-vous prise pour ajouter le récolte , après avoir rempli la formulaire d’ajout du récolte et la vérification des information, ces information seront enregistré dans notre base de données.

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

///screnn

Ce diagramme montre comment chaque utilisateur peut voir les information d’un récolte qui lui appartienne

## 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

Lorsque l'on parle de «Front-End», il s'agit finalement des éléments du site 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 lien(<https://www.alticreation.com/difference-developpeur-front-end-et-developpeur-back-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. (<https://jquery.com/>)

**Jspdf** : c’est une bibliothèque JavaScript qui généré des documents PDF en utilisant des information, donner par l’utilisateur (dans notre cas on l’utilise pour génère des convocation des rendez-vous et des facture)[https](https://www.chartjs.org/) <https://parall.ax/products/jspdf>

**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 <https://www.chartjs.org/>

### 1.2. Technologies utiliser dans la front-end

Le Back-End, c'est la partie 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.

Dans notre application nous avons utilisé l’environnement XAMPP(**X A**pache **M**ySQL **P**erl **P**HP) qui est une plateforme de développement Web permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP et Perl, 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 aussi utiliser AJAX *(*Asynchrones JavaScript and XML) 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. <https://openclassrooms.com/fr/courses/1916641-dynamisez-vos-sites-web-avec-javascript/1920925-quest-ce-que-lajax>

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.

### 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

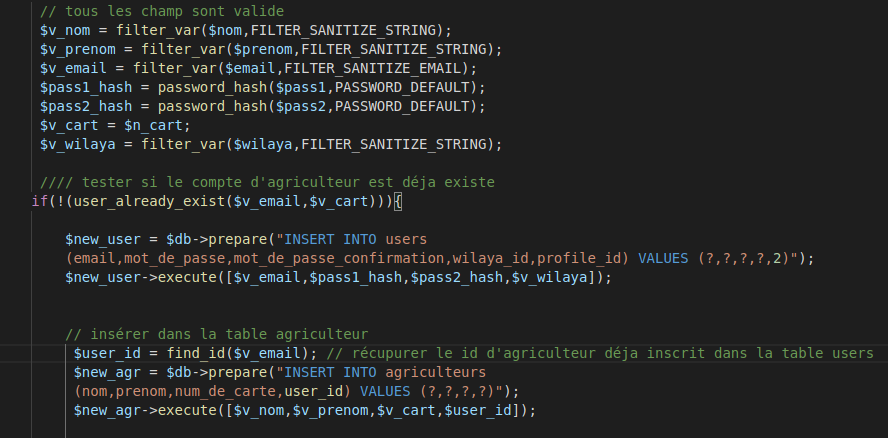


Figure Code d'inscription d'un agriculteur

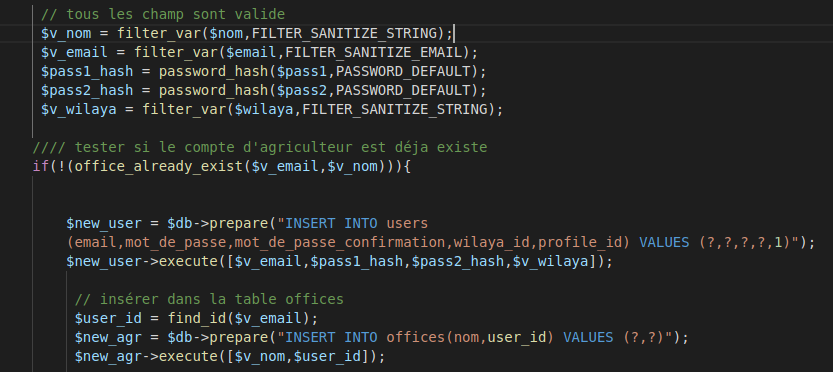


Figure Code d'inscription d'un office

### 2.2. Connexion

L’utilisateur entre ces information puits en vérifier les informations en coté client et serveur et enfin lorsque les informations entré sont correctes et ce compte est déjà vérifier en le dirige vers la page qui convient leur type (agriculteur ou office).

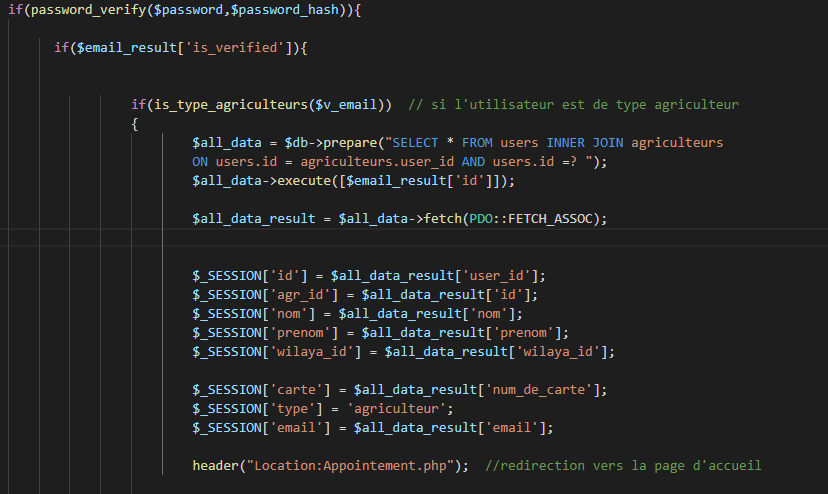


Figure  : code pour la connexion d'un agriculteur

### 2.3. Validation des comptes

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

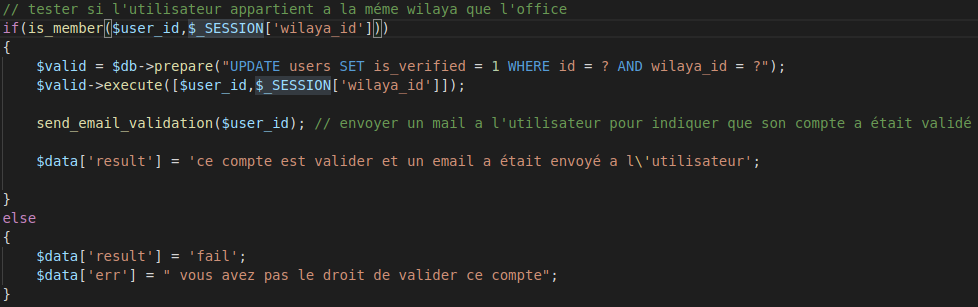


Figure  : code de validation d’un compte agriculteur

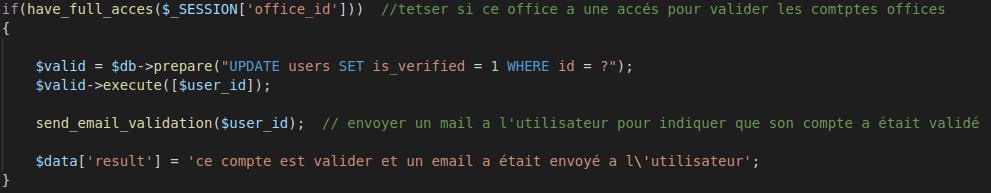


Figure  : code de validation d’un compte agriculteur

### 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 sauront validé puis on ajoute ce rendez-vous et désormais les agriculteurs de la même wilaya que ce office peuvent le perdre

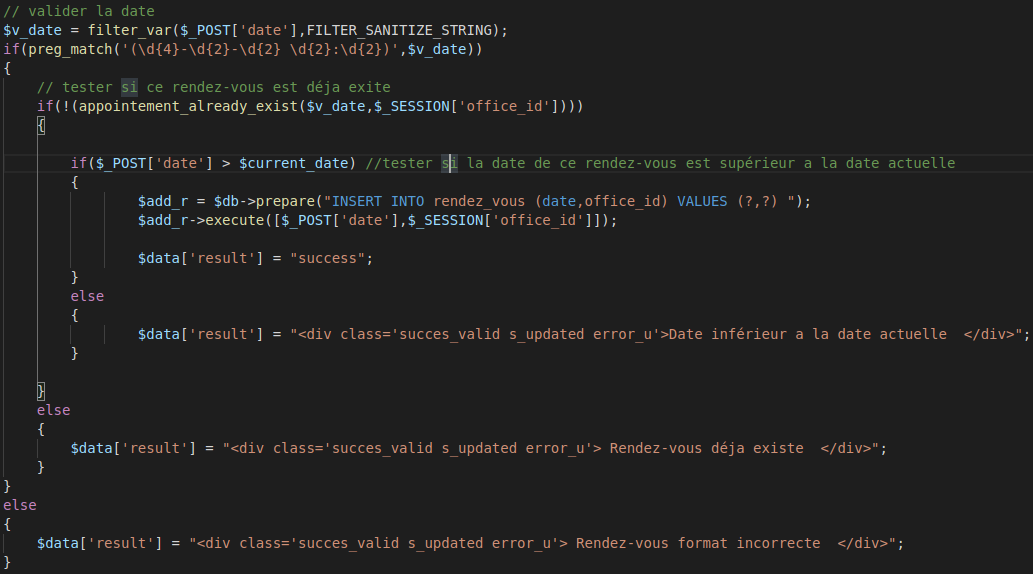


Figure 9 : code pour l'ajout d'un rendez-vous

### 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

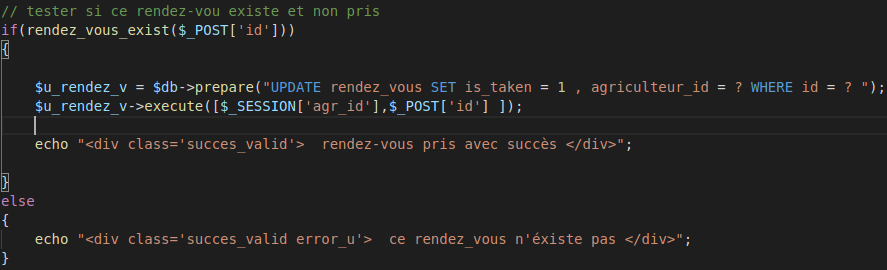


Figure 10 : code pour prendre un rendez-vous

### 2.5. Les récoltes

### 2.5.1 Ajouter un récolte

Chaque administrateur d’office peut ajouter des récoltes leur de rendez-vous

de dépôt de la récolte d’un agriculteur, il entre tous les 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

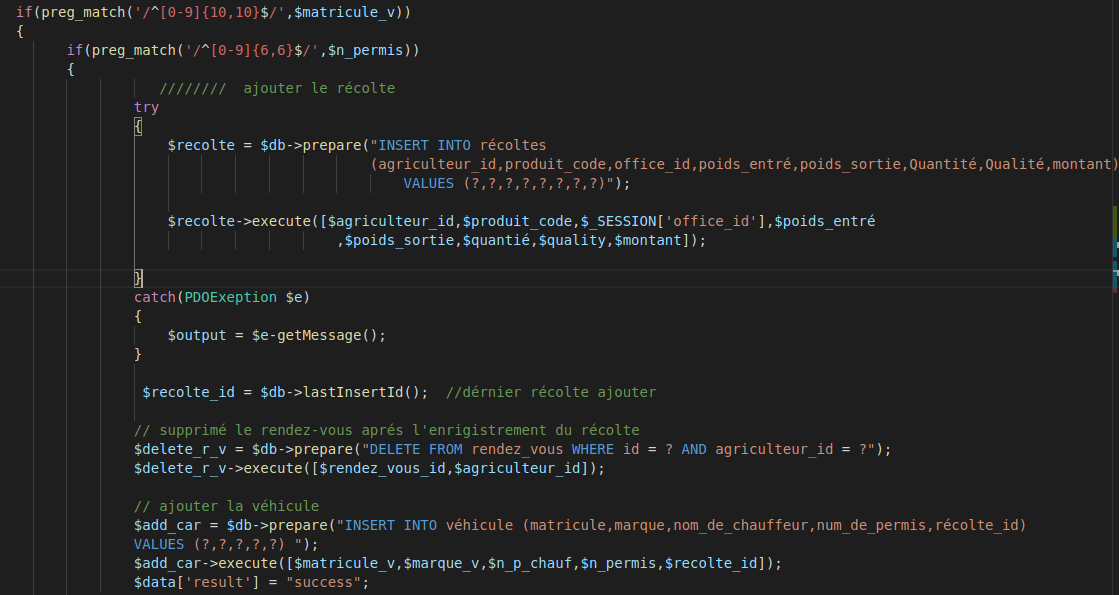


Figure : code de l'ajout d’une récolte

### 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

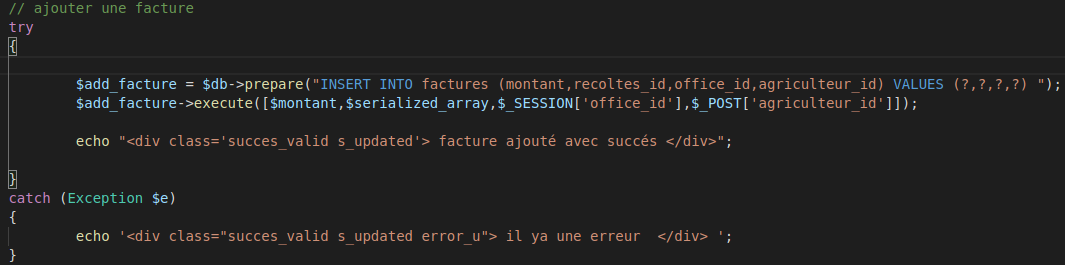


Figure : code d'ajout d'une voiture de la récolte

### 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 .

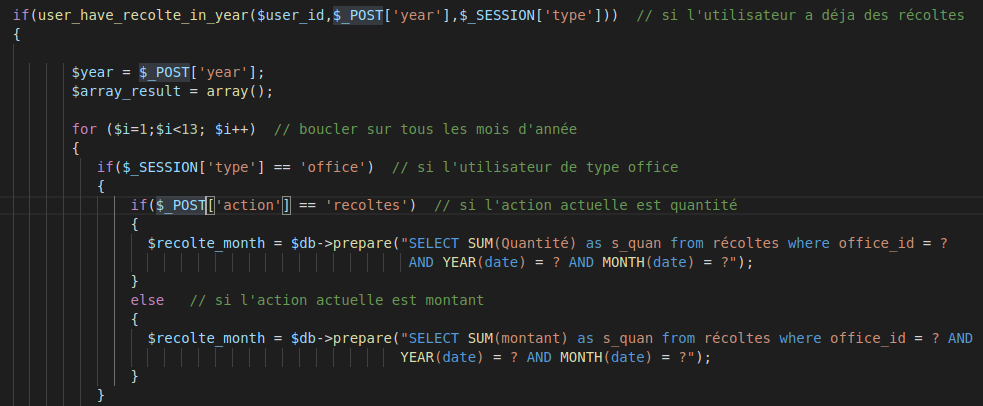


Figure : le code pour récupérer les récoltes d'un office pendant une année donné regroupé par mois

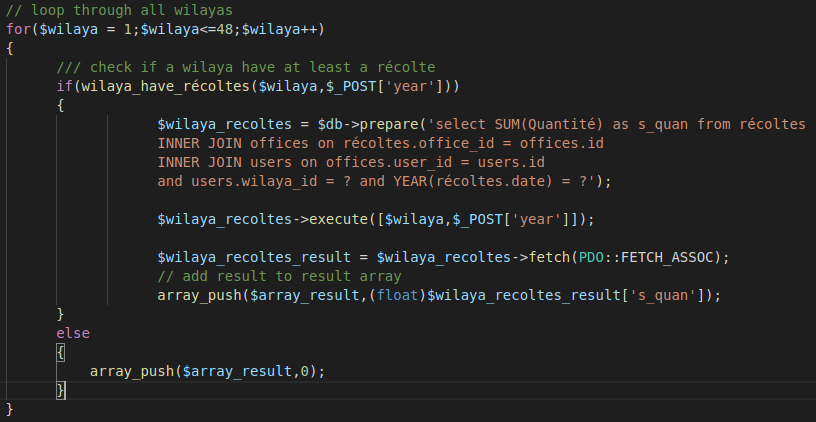


Figure  : le code pour récupérer la quantité des récoltes de tous wilayas pendant une année donnée

## 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

Rappeler le sujet et la problématique en allant du plus général au plus spécifique. Parler des difficultés ayant été surmontées pour réaliser le travail (application et/ou mémoire), de votre expérience, de ce que vous appris.

Il faut ensuite rappeler la solution réalisée tout en abordant les résultats obtenus. Il faut présenter les atouts du travail réalisé et parler, sans être pessimiste ou donner l’impression que le travail comporte des lacunes, des limites de ce qui a été fait.

La partie perspective (extension future du travail, amélioration de certains points…) démarre des limites de votre travail et propose des améliorations possibles. Il faut discuter minutieusement vos propositions ainsi que leurs intérêts. A noter que les perspectives ne doivent pas dépasser le travail lui-même.

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.

# Annexe A

A priori, le nombre des annexes n’est pas limité mais vous devez respecter la limite de pages imposée. Une annexe doit toujours être référencée dans le texte du mémoire, par exemple, « voir l’annexe X pour tel ou tel ».

# Annexe B

Une deuxième annexe…

**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, …]