

**Université Assane SECK de  
Ziguinchor**  
**UFR Sciences et technologies**  
**Département Informatique**



## **Mémoire de fin d'études**

Pour l'obtention du diplôme de Master

Mention : Informatique

Spécialité : Informatique

**DEVELOPPEMENT D'UNE APPLICATION POUR LE  
PARTAGE DE CONNAISSANCES SUR LES PLANTES  
MEDICINALES AU SENEGAL**

Présenté par : Mlle Mame Diarra SY

Soutenu le 12 / 03 / 2022

Sous la direction de : Pr Marie NDIAYE

**Devant le jury composé de :**

Pr Ousmane DIALLO	Maître de conférences CAMES	<b>Président</b>
Pr Youssou FAYE	Maître de conférences CAMES	<b>Examineur</b>
Dr Mouhamadou GAYE	Maître de conférences CAMES	<b>Examineur</b>
Pr Marie NDIAYE	Maître de conférences CAMES	<b>Directrice de mémoire</b>

Année universitaire 2020-2021

## Résumé

La médecine par les plantes est l'utilisation de plantes médicinales pour se soigner. Cette pratique existe depuis des siècles et a toujours un impact fort dans la société. On l'appelle souvent les remèdes de grand-mère pour marquer l'ancienneté de ses usages.

Le Sénégal est un pays d'Afrique occidentale où la majorité des populations réside dans la pauvreté. Cette situation influe fortement dans le domaine de la santé car l'accès à la médecine dite moderne n'est pas toujours cadeau chez ces populations. D'ailleurs, dans la plupart des zones, l'accès même aux infrastructures de santé pose d'énormes problèmes. Ainsi, pour retrouver leur santé, la population fait souvent recours à de pratiques ancestrales soignantes.

Cependant, malgré la disponibilité des plantes médicinales, on observe beaucoup de conséquences néfastes dans l'utilisation de ces dernières. Face à cette situation, nous avons pensé à la réalisation d'un système de partage qui permettra aux personnes, en cas de besoin, d'avoir suffisamment d'informations concernant les plantes médicinales, les maladies et les traitements qu'il faut suivre pour chaque maladie afin de bien se soigner par les plantes médicinales. De plus, ce système permettra de faire des échanges entre médecins et patients dans le but de les orienter dans une meilleure pratique de la médecine par les plantes.

## Abstract

Herbal medicine is the use of medicinal plants to heal oneself. This practice dates back centuries and still has a strong impact on society. It is often referred to as grandmother's remedies to show how long it has been used.

Senegal is a West African country where the majority of the population lives in poverty. This situation has a strong influence on health care, as access to so-called modern medicine is not always available to these populations. Moreover, in most areas, access to health infrastructures poses enormous problems. Thus, to regain their health, the population often resorts to ancestral healing practices.

However, despite the availability of medicinal plants, there are many harmful consequences in their use. Faced with this situation, we thought of creating a sharing system that will allow people, in case of need, to have enough information about medicinal plants, diseases and treatments that must be followed for each disease in order to be well treated by medicinal plants. In addition, this system will allow exchanges between doctors and patients in order to guide them in a better practice of herbal medicine.

## Remerciements

*Nous rendons grâce à ALLAH de nous avoir donné le courage, la patience, la motivation et la capacité de pouvoir acheminer ce travail.*

*Nos profondes reconnaissances vont à l'endroit de notre guide spirituel Cheikh Ahmadou BAMBA, qu'ALLAH l'agrée pour ses enseignements de la quintessence de la vie et ses orientations dans l'obéissance au souverain JUGE.*

*Nous remercions solennellement notre chère professeur Mme DIOP née Marie NDIAYE pour sa disponibilité, son accompagnement, sa simplicité et sa bravoure. Nous vous témoignons notre profonde gratitude.*

*Nous remercions les membres de notre jury en commençant par le président Pr Ousmane DIALLO, les examinateurs Pr Youssou FAYE et Dr Mouhamadou GAYE. Merci d'avoir accepté d'évaluer notre travail.*

*Nous remercions parallèlement notre tuteur M. Alioune Badara DIENG chef de service pédagogique de l'UFR ST, M. Yaya DIALLO ainsi que tout le personnel de l'UFR ST et l'ensemble des enseignants et personnel administratif de l'université.*

*Nos remerciements vont à l'endroit du Professeur Muhamed CHARAKHABIL, Professeur Abdoulaye GASSAMA et Docteur Gorgoumack SAMBE pour leur disponibilité et leurs encouragements.*

*Nous remercions nos sœur Daba SY, Amy SY et Fatou SY ainsi que leurs maris pour leurs soutiens et leurs encouragements. Nous témoignons leur désir de nous voir réussir, que le bon DIEU nous l'accorde à tous.*

*Nous tenons également à remercier tous nos sœurs et frères de famille.*

*Nous remercions notre coach M. Mor MBOUP et M. Camir Augustin MALACK de nous avoir assisté techniquement. Merci pour votre disponibilité.*

*Nous remercions tous nos camarades de classe.*

## Dédicaces

*Je dédie ce travail :*

*A ma chère maman Oulimata GUEYE qui ne s'est jamais lassée et qui a toujours été là pour nous soutenir et nous encourager. Ma meilleure, je te souhaite longue vie, beaucoup de santé et de bonheur, merci pour tout ;*

*A mon père Mbaye SY, qui nous a toujours soutenu avec beaucoup d'amour ;*

*A nos camarades qui sont partis Alpha SANE et Mamour DIOUF et à mon ami et frère Saër THIANE, que DIEU les accueille dans son paradis éternel.*

## Table des matières

Résumé.....	i
Abstract.....	ii
Remerciements.....	iii
Dédicaces.....	iv
Liste des figures.....	vii
Liste des tableaux.....	ix
Abréviations.....	x
Introduction générale.....	1
Chapitre 1 : Présentation du sujet et de la méthodologie de travail.....	3
I. Contexte justificatif du sujet.....	3
I.1. La médecine par les plantes.....	3
I.2. Contexte justificatif.....	4
II. Problématique du sujet.....	7
II.1. Objectifs.....	8
II.2. Solution proposée.....	9
III. Méthodologie de travail.....	9
III.1. Le cycle en V.....	10
III.1.1. Définition et description.....	10
III.1.2. Le choix du cycle en V.....	12
III.1.3. Nos limites par rapport à cette méthode.....	12
III.2. La méthodologie Scrum.....	12
III.2.1. Le choix de Scrum.....	13
III.2.2. Adaptation de Scrum dans notre travail.....	13
IV. Déroulement du projet.....	13
Chapitre 2 : Spécification des besoins.....	15
I. Langage de modélisation UML.....	15
II. Spécification des besoins fonctionnels.....	17
II.1. Identification des acteurs.....	17
II.2. Identification des besoins fonctionnels.....	18
II.3. Diagramme de cas d'utilisation.....	19
Chapitre 3 : Analyse des besoins fonctionnels et conception de l'application.....	25
I. Analyse des besoins fonctionnels.....	25
I.1. Analyse de l'authentification.....	25

I.2.	Analyse de la gestion des plantes.....	28
I.3.	Analyse de la gestion des traitements.....	31
I.4.	Analyse de l'administration.....	34
II.	Conception.....	36
II.1.	Conception générale.....	36
II.1.1.	Architecture de l'application.....	36
II.1.2.	Diagramme de déploiement.....	38
II.2.	Conception détaillée.....	38
II.2.1.	Dictionnaire de données.....	38
II.2.2.	Diagramme de classes.....	39
Chapitre 4 :	Implémentation et présentation de l'application.....	41
I.	Outils et technologies utilisées.....	41
I.1.	Environnement de développement : PyCharm.....	41
I.2.	Langage de programmation : Python Django.....	42
I.3.	Langage de balisage : HTML.....	42
I.4.	Langage de requêtes : SQL.....	43
I.5.	Framework : Bootstrap.....	43
I.6.	SGBDR : PostgreSQL.....	44
II.	Création et structure du projet.....	44
II.1.	Implémentation de la base de données.....	46
II.1.1.	Modèle logique de données.....	47
II.1.2.	Création et configuration de la base de données.....	47
II.2.	Implémentation de l'application.....	49
II.2.1.	Code de la classe « Traitement » : modèle.....	49
II.2.2.	Codage du fichier urls.py de l'application.....	50
II.2.3.	Codage du fichier admin.py.....	51
II.3.	Présentation de quelques interfaces de l'application.....	51
II.3.1.	Accès publique.....	51
II.3.2.	Accès réservé.....	55
Conclusion générale et perspectives.....		58
Bibliographie.....		59
Annexe.....		61

## Liste des figures

Figure 1: modèle cycle en V.....	11
Figure 2: schéma d'UML.....	16
Figure 3: schéma de Visual Paradigm For UML.....	16
Figure 4: représentation d'un acteur.....	17
Figure 5: diagramme de cas d'utilisation de la gestion des plantes.....	21
Figure 6: diagramme de cas d'utilisation de la gestion des traitements.....	22
Figure 7: diagramme de cas d'utilisation de la gestion des maladies.....	23
Figure 8: diagramme de cas d'utilisation de la gestion des utilisateurs.....	24
Figure 9: diagramme d'activité de l'authentification.....	27
Figure 10: diagramme de séquence de l'authentification.....	28
Figure 11: diagramme d'activité du cas rechercher une plante.....	30
Figure 12: diagramme de séquence du cas rechercher une plante.....	31
Figure 13: diagramme d'activité du cas rechercher un traitement.....	33
Figure 14: diagramme de séquence de la recherche de traitement.....	34
Figure 15: diagramme d'activité du cas "ajouter un utilisateur".....	35
Figure 16: diagramme de séquence du cas "ajouter un utilisateur".....	36
Figure 17: présentation de l'architecture.....	37
Figure 18: diagramme de déploiement.....	38
Figure 19: diagramme de classe.....	40
Figure 20: logo de PyChram.....	41
Figure 21: logo de Django.....	42
Figure 22: schéma de HTML.....	43
Figure 23: schéma de SQL.....	43
Figure 24: schéma de Bootstrap.....	44
Figure 25: schéma de PostgreSQL.....	44
Figure 26: structure du projet à la création.....	45
Figure 27: résultat du serveur.....	46
Figure 28: structure de l'application « plante ».....	46
Figure 29: modèle logique de données.....	47
Figure 30: interface d'accès à pgAdmin.....	47
Figure 31: interface d'accès au serveur.....	48
Figure 32: configuration d'accès à la base de données.....	48

Figure 33: création de la classe «Plante».....	49
Figure 34: déclaration de la classe « Traitement ».....	50
Figure 35: fichier urls.py.....	50
Figure 36: fichier admin.py.....	51
Figure 37: page d'accueil de l'application.....	52
Figure 38: page d'accueil des plantes.....	53
Figure 39: détails des plantes.....	53
Figure 40: page d'accueil des maladies.....	54
Figure 41: page d'accueil des traitements.....	54
Figure 42: détails traitement.....	55
Figure 43: interface d'authentification.....	55
Figure 44: interface d'administration générale.....	56
Figure 45: insertion d'une plante.....	57

## Liste des tableaux

Tableau 1: quelques ressources web parlant des plantes médicinales.....	5
Tableau 2: identification des acteurs et des rôles.....	14
Tableau 3 : représentation des acteurs.....	22
Tableau 4: identification de fonctionnalités.....	23
Tableau 5: description de l'authentification.....	29
Tableau 6: description du cas rechercher une plante.....	32
Tableau 7: description du cas rechercher un traitement.....	35
Tableau 8: description de la gestion de l'administration.....	37

## Abréviations

**OMS** : Organisation Mondiale de la Santé

**UML** : Unified Modeling Language

**VCS** : Version Control System

**IDE** : Integrated Development Environment

**HTML** : HyperText Markup Language

**CSS** : Cascading Style Sheets

**SQL** : Structured Query Language

**MIT** : Massachusetts Institute of Technology

**SGBD** : Système de Gestion de Base de Données

**SGBDR** : Système de Gestion de Base de Données Relationnelles

**MVT** : Modèle Vue Template

**IHM** : Interface Home Machine

**MVC** : Modèle Vue Contrôleur

**ORM** : Objet Relationnel Mapping

**FAQ** : Foire Aux Questions

**HTTP**: Hypertext Transfer Protocol

## Introduction générale

La médecine par les plantes est l'utilisation des plantes médicinales pour se soigner. Cette pratique ancestrale réside toujours au Sénégal bien qu'une grande partie de ces pratiquants ignore les vraies conséquences qui pourraient survenir de la non-maitrise de la plante.

Jusqu'à nos jours, il n'existe aucun logiciel de partage ou de sensibilisation efficace sur l'utilisation des plantes médicinales au Sénégal. Toutefois, il existe des sites internet, des émissions télévisées, des émissions radio, des tradipraticiens etc. qui parlent des plantes médicinales dans son ensemble. Néanmoins, la véracité de leurs dires reste un consensus car d'après des recherches faites auprès des scientifiques (botanistes et chimistes), les plantes médicinales peuvent détenir des molécules nuisibles à la santé et ceci à long terme.

Fort de ces constats, afin d'améliorer les conditions sanitaires des populations et valoriser les plantes médicinales, nous présentons, à travers ce document, une solution informatique liée à l'utilisation des plantes médicinales. Il s'agit précisément d'une application web à travers laquelle toute personne désireuse peut trouver les plantes qui traitent une pathologie particulière ou les pathologies traitées par une plante avec toutes les informations associées (dosage, effets indésirables, etc.). Pour bien aborder ce travail, nous allons structurer notre document en cinq chapitres répartis comme suit :

- ❖ **Chapitre 1 : Présentation du sujet et de la méthodologie** dont l'objectif est de situer le domaine d'étude de notre sujet, le contextualiser à travers la présentation de la médecine par les plantes et du contexte justificatif. Ainsi le chapitre se poursuit en évoquant la problématique, en définissant nos objectifs et en présentant les solutions que nous voulons apporter. Pour terminer ce chapitre, nous avons mis au point les méthodologies de travaux que nous adoptons pour bien réaliser ce projet.
- ❖ **Chapitre 2 : Spécification des besoins** : dans ce chapitre, nous présentons le langage UML que nous avons adopté pour la conception de nos diagrammes. Dans la deuxième partie, nous faisons la spécification c'est-à-dire nous allons étudier les acteurs et quelques fonctionnalités de l'application.
- ❖ **Chapitre 3 : Analyse des besoins fonctionnels et conception de l'application** : ce chapitre se définit en deux grandes parties. Dans la première partie, nous faisons l'analyse de quelques cas d'utilisation. Dans la deuxième partie intitulée la conception qui se définit sur deux grandes parties à savoir la conception générale (architecture et diagramme de déploiement) et la conception détaillée (dictionnaire de données et diagramme de classe).

❖ **Chapitre 4 : Implémentation et présentation de l'application** : c'est le dernier chapitre du document. il consiste à l'implémentation de l'application, de la base de données et à la présentation de quelques interfaces de l'application. Mais nous l'avons précédé des outils et technologies qui nous accompagnent pour la réalisation de ce projet.

Après la présentation des quatre chapitres cités ci-haut, nous allons proposer une conclusion générale et des perspectives.

# Chapitre 1 : Présentation du sujet et de la méthodologie de travail

Situé sur la côte ouest de l’Afrique, le Sénégal a une position géographique stratégique qui lui confère une ouverture sur le monde et une facile accessibilité. Il couvre une population d’environ 17 215 433 habitants sur une étendue de 196 712 km<sup>2</sup>[1]. Cette population est répartie sur quatorze (14) régions, dont la capitale est DAKAR, soit 46 départements en 2021. La moyenne d’âge est estimée à 19 ans justifiant que la majorité de la population est jeune. Cette population présente des réalités socio-culturelles notamment des pratiques ancestrales dont la médecine traditionnelle.

La médecine traditionnelle est un concept très complexe qui relève d’un ensemble de caractères matériels, intellectuels, moraux et artistiques d’une société donnée. Ses pratiques varient grandement d’un pays à l’autre, d’une région à l’autre et sont fortement influencées par de nombreux facteurs comme la culture, l’histoire, les ethnies, la situation sociale, etc. Ainsi, la médecine traditionnelle sénégalaise n’en est pas une exception ; elle est issue des diverses influences culturelles et religieuses ayant marqué son histoire[2].

Cette médecine traditionnelle fait, de nos jours, objet de consensus dans son utilisation car on assiste à de nombreux événements causés par cette technique de traitement qui continue de séier dans le pays.

Ainsi, pour cadrer notre sujet, nous allons nous intéresser à une partie de la médecine traditionnelle qui est l’utilisation des plantes médicinales. Nous allons voir en détails les concepts fondamentaux de cette pratique, le contexte dans lequel le sujet a été défini et ainsi, dégager la problématique et proposer des solutions définies sur des méthodologies de travail.

## I. Contexte justificatif du sujet

### I.1. La médecine par les plantes

Une plante médicinale est une plante qui possède des capacités à guérir ou à soulager certains maux. Pour une définition plus simple, la médecine par les plantes (phytothérapie) est l’utilisation de plantes médicinales pour se soigner. Les plantes médicinales sont utilisées pour leurs propriétés thérapeutiques. Cela signifie qu’au moins une de leurs parties (feuille, tige, racine etc.) peut être employée dans le traitement d’une maladie.

En effet, pendant de nombreux siècles, les plantes médicinales ont constitué le principal élément thérapeutique à la disposition de l’homme. Leurs propriétés ont été mises en évidence par l’observation des effets qu’elles généraient sur l’organisme de ceux qui les absorbaient.

Souvent, quand on parle de médecine naturelle, la première chose qui vient à l'esprit est la plante médicinale. Mais, la médecine traditionnelle ne se résume pas uniquement aux plantes médicinales même si la majorité de ses thérapies sont principalement à base de plantes médicinales. Comme définie dans l'introduction de ce chapitre, la médecine traditionnelle, naturelle, douce, non conventionnelle, alternative ou complémentaire désignent tous la même technique qui consiste à guérir, soulager ou atténuer des maux, en ayant recourt à des méthodes et des substances 100% naturelles[3].

Notre intérêt porté sur la médecine par les plantes, nous amène à l'étude de différents aspects liés à son utilisation. Dans la suite, nous allons voir le contexte justifiant le choix de notre sujet.

## I.2. Contexte justificatif

Comme indiqué dans la première partie de ce chapitre, les plantes médicinales ont constitué, pendant de nombreux siècles, le principal élément thérapeutique à la disposition de l'homme. Avec les progrès de la chimie, les chercheurs ont été en mesure d'en isoler les principes actifs ; ces substances chimiques naturelles considérées comme responsables de leur action. Par la suite, ils les ont synthétisé artificiellement d'où la naissance de l'industrie pharmaceutique[4]. C'est ainsi que les plantes médicinales, jugées moins actives et fiables que le médicament de synthèse, furent alors écartées de la médecine. C'est dans ce contexte que l'OMS (Organisation Mondiale de la Santé) avait émis des réticences et a précisé qu'elle n'accepterait de recommander ce type de traitement qu'à la condition que ceux-ci reposent sur des éléments scientifiques probants. Mais cette mise à l'écart n'a duré qu'un temps ; les effets secondaires néfastes, la non efficacité, la non accessibilité de la plupart des médicaments de synthèse se sont vite révélés et on a recommencé à s'intéresser aux plantes. D'ailleurs, dans son annonce du 19 septembre en collaboration avec le Centre africain pour le contrôle et la prévention des maladies et la Commission des affaires sociales de l'Union africaine, l'OMS a mis en place et approuvé un protocole pour des essais cliniques de phase III concernant des traitements de phytothérapie pour la Covid-19[5].

Ces connaissances naturelles qui ont souvent été transmises de manière orale, de génération en génération, requièrent aujourd'hui l'attention des populations, en particulier les pays en voies de développement.

Le Sénégal, faisant partie des pays les moins avancés et aux moyens sanitaires restreints, sa population n'est pas indifférente à l'utilisation des plantes médicinales. En effet, le traitement

par les médicaments modernes n'est pas toujours accessible pour la majeure partie des populations. Nous constatons souvent que celle-ci fait recours à des moyens comme les cartes sanitaires (mutuelle de santé etc.) pour augmenter leur chance de se faire soigner dans les structures de santé modernes. A cela s'ajoute la difficulté liée à l'accès à ces structures de santé notamment dans les zones rurales.

Compte tenu de toutes ces circonstances, nous assistons, depuis quelques décennies, à la médiatisation de techniques naturelles de traitements. Nous constatons des émissions télévisées telle que « Sen wergi yaram » de Serigne Samba NDIAYE SenTv/Sénégal, des émissions radio, des chaînes de partages sur YouTube, des sites web, etc.

On peut décrire quelques sites web qui partagent des informations et des traitements à base de plantes médicinales en nous basant sur les propriétés suivantes :

- Lien : décrit le lien de la source ;
- site/auteur : le site où a été publié l'article ou l'auteur du livre (dans le cas où l'article est un livre) ;
- titre : décrit le titre du document ;
- description : présente un petit résumé de ce qui a été présenté dans le document ;
- date publication : la date de publication du document ;
- date de visite : la date à laquelle on a consulté le document ;
- accessibilité : permet d'identifier si la source en question est libre ou payante ;

Tableau 1: quelques ressources web parlant des plantes médicinales

<b>Lien</b>	<b>Site / Auteur</b>	<b>Titre</b>	<b>Accessi bilité</b>	<b>Description</b>	<b>Date de public ation</b>	<b>Date visité</b>
<a href="https://www.antes-medicinales-senegal.com/plantes-medicinales-">https://www.antes-medicinales-senegal.com/plantes-medicinales-</a>	AU- SENE GAL. COM	Plantes Médicina les du Sénégal	Libre	Traite des arbres retrouvés au Sénégal. Fait leurs descriptions (avec des images défilant) et donne leurs utilités dans la	06/04/ 2018	08/09/ 2020

du-senegal,3413.html				médecine traditionnelle (comment il décrit la plantes).		
<a href="https://www.doc-developpement-durable.org/fil e/Plantes-Medicinales-Aromatiques/plantesMedicina lesAfricaines.pdf">https://www.doc-developpement-durable.org/fil e/Plantes-Medicinales-Aromatiques/plantesMedicina lesAfricaines.pdf</a>	Doc-devel oppe ment-durabl e.org	A la découvrir te de quelques plantes africaine s médicin ales	Libre	Décrit des plantes médicinales utilisées au Bénin (avec des images à l'appui) en donnant leurs principes actifs, propriétés et leurs thérapeutiques traditionnelles au Bénin.		08/09/2020
<a href="https://www.afriquebio.com/pages/des-remedes-simples-et-pratiques/tout-sur-les-plantes-d-afrique.html">https://www.afriquebio.com/pages/des-remedes-simples-et-pratiques/tout-sur-les-plantes-d-afrique.html</a>	Afriquebio.com	Tout sur les plantes d'Afrique	Libre	Dans cet article, l'auteur présente des plantes médicinales (classées par catégorie selon les maladies qu'elles soignent) avec, pour chacune, son nom dans quelques langues africaines, sa description, ses actions curatives et son mode d'emploi dans chaque cas de maladie.		28/09/2020
<a href="https://naturoafricaine.com/onwebmedia/Plantes%20Medicinales%20Africaines.pdf">https://naturoafricaine.com/onwebmedia/Plantes%20Medicinales%20Africaines.pdf</a>	Jean-Louis POUSSET	Plantes médicinales africaines : utilisation pratique	Libre	J.L.P fait une étude l'histoire de l'homme avec les plantes médicinales africaines, leurs efficacités ainsi que leurs utilisations. Pour chaque plante, il donne l'historique et l'utilisation en médecine		11/09/2020

				traditionnelle, la description, les actions curatives et des références pour plus de recherches.		
<a href="https://www.santescience.fr/plantes-medicinales/">https://www.santescience.fr/plantes-medicinales/</a>	santescience.fr	Plantes médicinales	Libre	C'est un site français qui, d'après eux, décrit toutes les plantes médicinales de A à Z.		28/10/2020

L'utilisation croissante de la phytothérapie s'accompagne de quelques difficultés que nous allons voir dans la suite.

## II. Problématique du sujet

Dans nos pratiques de tous les jours, le mode d'utilisation des plantes médicinales n'est ni écrit, ni archivé quelque part, c'est juste un message transmis de bouche à oreille, de parents aux enfants, etc. Or, on sait bien qu'un message transmis de bouche à oreille perd sa valeur en tirant de longueur. Ce qui amène souvent que nous rencontrons des difficultés avec l'utilisation de ces plantes. Ces problèmes d'utilisation peuvent entraîner des conséquences douloureuses et quelques fois regrettables. Parmi ces craintes, nous pouvons en citer les plus fréquentes :

- **La connaissance de la plante** : beaucoup de gens utilisent des plantes pour se soigner sans qu'ils ne les connaissent vraiment. c'est juste un remède dicté par un ami ou un parent. Ce qui peut s'avérer très dangereux pour la santé. Il est également important de connaître l'utilité des différentes parties de la plante car, ces dernières peuvent avoir des propriétés thérapeutiques distinctes.  
La connaissance de la plante fait également intervenir d'autres aspects comme : la saison à laquelle la plante est récoltée, l'âge de la plante et l'heure de cueillette de la plante ;
- **le dosage** : le dosage des ingrédients pour un traitement peut varier selon l'âge, le sexe ou la situation du malade (si c'est une femme enceinte ou en allaitement) etc. Un excès de dosage peut entraîner des effets indésirables (problème de reins dans la plupart du temps) ou la non efficacité du traitement ;
- **l'appellation** : celle-ci reste un problème fondamental car les plantes peuvent être appelées différemment d'une localité à une autre (problème de langage) ;

- **Le traitement** : certains traitements ne sont pas adaptés à tous les types de personnes. Il peut y'avoir des différences de traitement entre une femme enceinte, un adulte, un adolescent, un nouveau-né, une personne présentant d'autres types de maladies ou des effets allergiques à un composant du traitement. La durée du traitement est également un aspect important dans l'utilisation des plantes ;
- **Le mode et la voix d'administration** : les modes et voix d'administration peuvent varier d'un traitement à un autre ;
- **L'absence de dispositif accessible et fiable de partage des connaissances** : ceci reste l'un des problèmes majeurs de l'utilisation des plantes médicinales. Il se trouve que les populations ont du mal à s'orienter pour utiliser les plantes sans risques de regrets futurs ;
- **La non fiabilité de l'information** : après des études faites par des scientifiques sur les plantes médicinales, il est avéré que des tradipraticiens dictent des plantes à leurs patients alors que ces dernières possèdent des substances désagréables qu'ils ne maîtrisent pas et qui peuvent avoir des conséquences à long terme.

Pour résoudre les problèmes cités ci-dessus, nous nous sommes fixés des objectifs que nous décrivons dans la section suivante.

## II.1. Objectifs

Dans le cadre de notre étude tournant autour d'un système de partage sur l'utilisation des plantes médicinales pour se soigner, notre principal objectif est de créer une application web pour le partage de connaissances fiables et sécurisées. Ce portail permettra aux populations d'avoir des informations fiables sur les plantes et sur leurs utilisations dans la médecine.

Sur cette démarche, nous envisageons de traiter les points suivants :

- permettre à la population Sénégalaise de trouver les informations importantes à connaître sur une plante médicinale ;
- permettre à la population Sénégalaise de trouver le traitement adéquat à son besoin ;
- sensibiliser les populations sur les effets indésirables des plantes médicinales.

Afin d'atteindre ces objectifs, nous proposons des solutions.

## II.2. Solution proposée

Afin de secourir les populations dans leurs problèmes au quotidien sur l'utilisation des plantes médicinales, nous avons pensé à la création d'une application web pour partager nos

recherches scientifiques sur les plantes médicinales. Cette application va servir d'orientation et de guide concernant l'utilisation des plantes médicinales avec tout ce qui peut en découler.

Sachant que le problème majeur de ce sujet reste le manque de dispositifs fiables, notre solution va porter principalement sur les points suivants :

- proposer une base de données des plantes médicinales : nous allons enregistrer un grand nombre de plantes médicinales accessibles au Sénégal ;
- proposer une page des plantes : qui présente dans une interface une liste de plantes qui permet aux visiteurs d'effectuer une recherche et de voir les détails de chaque plante et éventuellement voir d'autres sources ;
- nous allons lier chaque maladie aux traitements appropriés. Ces traitements sont marqués par des propriétés comme la recette, le dosage, les ingrédients etc. ;
- proposer un système de commentaire pour permettre aux visiteurs de partager leur point de vue sur l'information partagée ;
- des recettes et des dosages venant de sources fiables ;

Pour atteindre ces objectifs, nous proposons de concevoir des interfaces d'accès aux différentes fonctionnalités de l'application avec possibilité de faire des recherches sur chaque page et de naviguer d'une page à une autre.

Pour se faire, nous nous sommes basées sur des méthodologies efficaces de travail afin de nous assurer une bonne gestion de notre projet.

### III. Méthodologie de travail

La méthodologie désigne l'adoption d'un ensemble de méthodes, de règles et de démarches pour la réalisation d'un projet. C'est une façon différente d'aborder un sujet. Elle nous permet de définir, d'ordonner les tâches et de bien suivre le processus de développement de notre application.

Dans la réalisation de notre travail, nous utilisons un modèle simple, très pratique et qui prend en charge toutes les phases de réalisation de projet, il s'agit du cycle en V.

Par ailleurs, pour atténuer l'effet tunnel induit par le cycle en V, nous allons emprunter quelques pratiques issues de la méthodologie Scrum à partir de l'étape de conception du cycle en V.

### III.1. Le cycle en V

#### III.1.1. Définition et description

Comme nous l'avons défini plus haut, le cycle en V inclut toutes les phases du cycle de vie d'un projet : conception, réalisation et validation. C'est une ancienne méthode de gestion de projet qui reste à ce jour incontournable. Elle s'identifie par une phase descendante puis une phase ascendante, illustrées par les deux branches du V d'où son appellation.

Encore appelée modèle de gestion de projet en cascade, le cycle en V repose sur des étapes séquentielles et linéaires, allant de l'analyse des besoins au test d'acceptation. La partie descendante du « V » correspond aux quatre actions de conception et de développement du système, tandis que la partie ascendante reprend les quatre phases d'assurance qualité qui lui sont associées. En effet, la particularité de ce modèle de gestion de projet est de combiner une phase de validation pour chaque phase de développement. Le point de jonction, le bas du V, correspond quant à elle à l'étape de réalisation[6].

Ainsi, la Figure 1 suivante est un schéma illustratif de ce modèle :

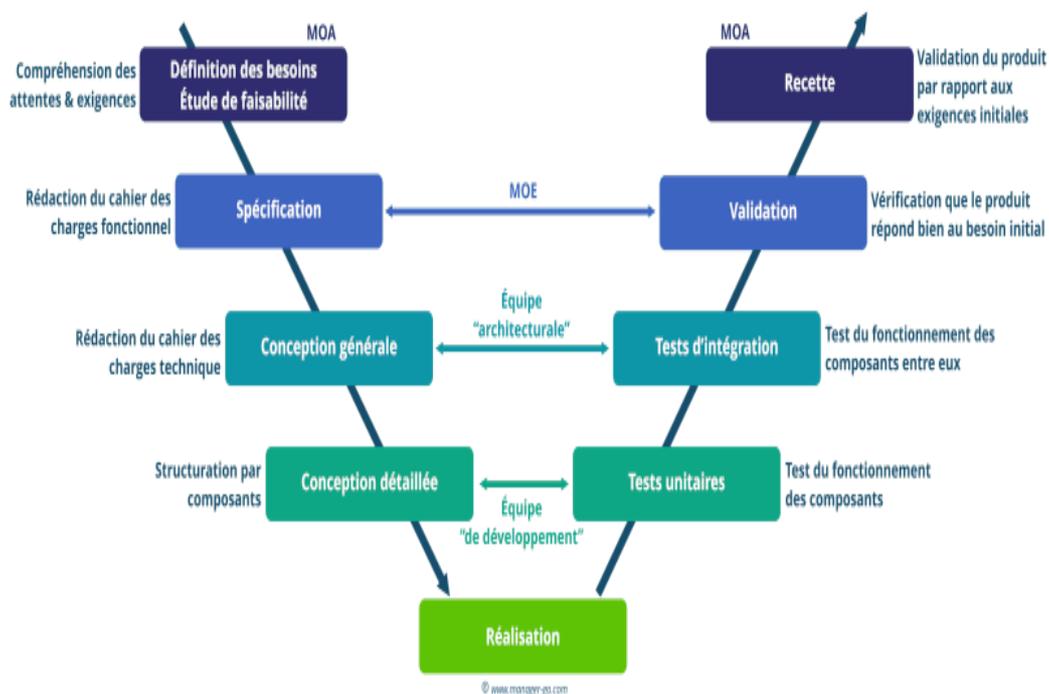


Figure 1: modèle cycle en V

Les phases de ce modèle sont définies sur neuf (9) étapes regroupées en trois grandes phases.

## 1. La conception

Dans un premier temps, la phase de conception permet de recueillir les besoins et spécifications nécessaires pour programmer de manière exacte l'ensemble du projet.

Elle fait intervenir :

- **l'analyse du besoin** : Il s'agit de comprendre les attentes et les exigences que nous devons prendre en compte dans notre solution. Nous avons fait une étude de la faisabilité et défini les besoins de nos clients (les visiteurs de notre application) ;
- **les spécifications** : ou définitions des spécifications fonctionnelles, elles définissent les demandes exactes de nos utilisateurs. Par contre, dans notre cas, un cahier de charges fonctionnelles ne sera pas défini car notre produit n'est pas à la demande d'un client ;
- **la conception générale** : ou conception architecturale, elle décline les spécifications fonctionnelles en spécifications techniques. Dans cette étape, nous avons récapitulé les technologies essentielles à la mise en œuvre du logiciel et leur architecture ;
- **la conception détaillée** : dans cette partie, nous avons précisé tous les composants indispensables à la fabrication de notre produit. Mais cette illustration va être réservée à la troisième partie de ce chapitre.

## 2. La réalisation

Après la conception vient l'étape de réalisation qui permet de créer et d'assembler tous les composants nécessaires pour arriver au produit final. Pour notre projet, nous avons choisi de travailler avec Django qui est un Framework python open source consacré au développement web.

## 3. La validation

La phase de validation comprend différents tests vérifiant chaque étape de la conception du produit. Ces mesures d'assurance qualité prennent la forme de :

- **tests unitaires** : Les tests unitaires correspondent à l'étape de conception détaillée. Ils permettent de détecter et de corriger d'éventuelles erreurs ;
- **tests d'intégration** : interviennent sur le produit fini. Regroupant des essais techniques et des essais fonctionnels, les tests d'intégration visent à s'assurer que le système fonctionne dans son ensemble ;
- **tests de validation** : correspondent à la phase des spécifications. Menés par les futurs utilisateurs du logiciel, les tests de validation sont basés sur des mises en situation

réelle pour vérifier la conformité fonctionnelle du logiciel par rapport aux spécifications ;

- **recette fonctionnelle** : appelée également test d'acceptation : c'est l'étape de validation du produit par rapport aux objectifs qu'on s'est fixés au niveau de la première phase, celle de l'analyse des besoins. La recette fonctionnelle est la dernière séquence avant la mise en production du logiciel.

Ainsi, dans la suite, nous allons expliquer nos motivations à adopter cette méthodologie.

### III.1.2. Le choix du cycle en V

Nous choisissons le cycle en V car il est assez intuitif et simple à mettre en œuvre. En effet, cette méthode reste la plus adaptée avec notre plan de travail. Elle ne nécessite pas de réunions quotidiennes. Les seules réunions de pilotage actent le passage d'une phase à l'autre et peuvent également se faire à distance (ce qui a facilité notre travail même pendant les vacances). La méthode nous évite de revenir en arrière incessamment pour redéfinir les spécifications initiales. Nous nous sommes penchés sur de bases solides pour assurer, sur chaque phase, les points fixés par le produit final avant de passer à la phase suivante. Ainsi, pour chaque phase de conception, nous avons une documentation précise et exhaustive.

Cependant, malgré ses avantages, la méthode cycle en V présente aussi des limites.

### III.1.3. Nos limites par rapport à cette méthode

Comme dans tous les autres projets utilisant le cycle en V, le modèle laisse un peu de place aux imprévus. Notre cadre ne répond pas à tous les concepts du cycle car nous ne formons pas une grande équipe où tous les rôles sont représentés. Ce qui nous a poussé à adopter quelques pratiques issues de l'approche agile, en particulier de la méthodologie Scrum.

## III.2. La méthodologie Scrum

Scrum est une méthode agile consacrée à la gestion de projets. C'est un cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. Elle utilise une approche itérative et incrémentale pour optimiser la prédictibilité et pour contrôler les risques.

Dans la section suivante, nous allons voir pourquoi le choix de Scrum dans notre projet.

### III.2.1. Le choix de Scrum

Notre choix d'adoption de la méthodologie Scrum dans notre travail repose principalement sur ces points :

- ♣ une vision centrée sur le produit plutôt que sur le projet ;
- ♣ transparence : visibilité et compréhension du travail ;
- ♣ inspection : détection des variations par rapports aux attentes ;
- ♣ adaptation : Ajustement rapide du processus pour minimiser les futures déviations ;
- ♣ etc.

### III.2.2. Adaptation de Scrum dans notre travail

Nous tenons à rappeler que dans ce projet, la méthodologie Scrum n'a pas été utilisée dans son intégralité, mais nous avons seulement fait appel à ses pratiques. Dans le Tableau 2 qui suit, nous allons mettre en évidence ce qui pourrait constituer notre équipe :

Tableau 2: identification des acteurs et des rôles

<b>Personnages</b>	<b>Rôles</b>	<b>Fonctions</b>
<b>Dr. Marie NDIAYE</b>	Product Owner	Enseignant chercheuse au Département Informatique
<b>Dr. Marie NDIAYE</b>	Scrum Master	Enseignant chercheuse au Département Informatique
<b>Madame Mame Diarra SY</b>	Equipe	étudiante en Master 2 Génie logiciel, département Informatique

## IV. Déroulement du projet

Dans le déroulement de notre projet, nous avons essayé, dans la mesure du possible, de coupler le cycle en V avec la méthodologie Scrum. La raison principale qui nous a poussé à ne pas appliquer le cycle en V dans son intégralité, est que notre projet n'est pas un projet d'équipe ou tous les rôles doivent être représentés. Et aussi, notre projet n'est pas un besoin à la demande d'un client. C'est une solution que nous souhaitons apporter à la population suite aux nombreux problèmes qu'ils rencontrent dans l'utilisation des plantes médicinales.

Ainsi, nous pouvons dire qu'au début de notre travail nous avons adopté les étapes du cycle en V lors des trois premières étapes. Et plus tard, lors de la conception détaillée nous avons commencé à nous fixer des sprints afin d'assurer fonctionnalité par fonctionnalité. Nous fixons des réunions une fois que quelque chose doit être présentée. Nous faisons aussi, en cas de besoin, des échanges par appel téléphonique, par mail, etc.

Par ailleurs, l'adoption de méthodes n'est pas suffisante pour mener à bien notre projet car nous avons aussi besoins d'utiliser des outils et des technologies adaptés.

## **Conclusion**

Dans ce chapitre, nous avons défini le concept de la médecine par les plantes, ensuite nous avons présenté le contexte justificatif de notre sujet puis la problématique. Ce qui nous a permis de nous fixer des objectifs pour finir par proposer une solution.

Nous avons également emprunté des méthodologies de travaux pour le bon déroulement de notre projet.

Ainsi dans le chapitre suivant, nous allons voir la spécification des besoins.

## Chapitre 2 : Spécification des besoins

Le processus de développement que nous avons adopté avec les méthodologies de travail, recadre un ensemble de besoins que nous avons défini en des fonctionnalités dans notre application. C'est tout à fait l'objet de ce chapitre, où nous allons, en premier lieu, parler de l'outil UML qui nous a servi pour la réalisation de nos diagrammes et en second lieu, nous allons faire la spécification c'est-à-dire identifier les acteurs du système et leurs fonctionnalités.

Nous rappelons que notre objectif est de créer une plateforme web permettant aux visiteurs de s'orienter sur l'utilisation des plantes médicinales.

Cependant, dans les sections spécification (chapitre 2) et analyse (chapitre 3) des besoins fonctionnels, nous allons présenter quelques diagrammes UML en illustration.

### I. Langage de modélisation UML

Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer de l'information ou de la connaissance des systèmes dans une structure qui est définie par un ensemble cohérent de règles.

UML (Unified Modeling Language ou Langage de [modélisation](#) unifiée en français) est un langage de modélisation standardisé composé d'un ensemble intégré de diagrammes, développés pour aider les développeurs de systèmes et de logiciels à spécifier, visualiser, construire et documenter les artefacts des systèmes logiciels, ainsi que pour la modélisation commerciale et d'autres systèmes non logiciels[7]. Il est constitué de treize (13) diagrammes servant à visualiser et à décrire la structure et le comportement des objets qui se trouvent dans un système. Il permet de présenter des systèmes logiciels complexes de manière plus simple et plus compréhensible qu'avec du code informatique.

Il porte notre choix pour ce projet car il est la référence en modélisation objet, ou programmation orientée objet.

Les diagrammes UML se présentent sous deux (2) catégories :

- les diagrammes de structure qui représentent les éléments du système, leurs propriétés et les relations entre eux ;
- les diagrammes de comportement qui représentent les processus et les interactions entre les objets.

La Figure 2 suivante représente, en image, le logo d'UML :



Figure 2: schéma d'UML

Pour réaliser nos diagrammes UML, nous avons utilisé un outil de conception qui est Visual Paradigm for UML.

### **Outils de conception : Visual Paradigm for UML**

Visual Paradigm for UML est, comme son nom le laisse supposer, un logiciel permettant aux programmeurs de mettre en place des diagrammes UML. C'est une application qui est compatible avec de nombreuses applications standards et environnements. Son point fort est qu'il est facile à utiliser, clair et intuitive. Son interface graphique améliore l'expérience de l'utilisateur en lui permettant d'aller plus loin que ce qu'il peut croire. Visual Paradigm for UML permet de générer du code à partir de diagramme UML et possède de nombreuses fonctionnalités dans ses versions payantes.

Chaque fonctionnalité de Visual Paradigm for UML est accessible de plusieurs façons. En générale, celle-ci est accessible dans les menus déroulant se situant classiquement en haut de la fenêtre. On peut également y accéder par le clic droit qui a l'avantage d'être contextuel et de ne présenter que les fonctionnalités en rapport avec l'objet cliqué. Enfin on pourra se servir de l'interface qui apparait lorsque l'on sélectionne un élément du diagramme qui permet également d'accéder à ces fonctionnalités.

Nous avons, ci-dessous, une représentation de Visual Paradigm For UML en Figure 3 :



Figure 3: schéma de Visual Paradigm For UML

La version d'essai gratuite est disponible dans le site officiel de Visual Paradigm[8].

## V. Spécification des besoins fonctionnels

La spécification des besoins nous permet d'identifier les utilisateurs, les fonctionnalités et les relations entre eux. Nous commençons par identifier les acteurs du système, ensuite les fonctionnalités et enfin nous présentons les diagrammes de cas d'utilisation.

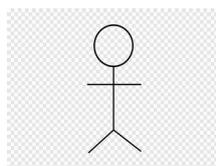
### V.1. Identification des acteurs

Un acteur est une entité extérieure au système modélisé, et qui interagit directement avec lui. En effet, il correspond à un rôle. Ainsi, une même personne physique peut être représentée par plusieurs acteurs si elle a plusieurs rôles mais aussi si plusieurs personnes jouent le même rôle vis-à-vis du système, elles seront représentées par un seul acteur. Ce qui fait qu'il reste le principal utilisateur du système.

En plus des utilisateurs, les acteurs peuvent être :

- des logiciels déjà disponibles à intégrer dans le projet ;
- des systèmes informatiques externes au système mais qui interagissent avec lui ;
- tout élément extérieur au système et avec lequel il interagit ;

Schématiquement, un acteur est représenté par un pictogramme humanoïde sous-titré par le nom de l'acteur (Figure 4).



« **Nom acteur** »

Figure 4: représentation d'un acteur

Le ci-dessous identifie les acteurs de notre système :

Tableau 3 : représentation des acteurs

Acteur	Définition	Action
Visiteur	Tout individu désirant visiter le système	Consulter toutes les plantes décrites dans le système ainsi que les détails et des sources de chacune d'elle, visualiser les traitements proposés pour chaque type de maladie, visualiser les maladies qui y sont traitées et éventuellement apporter des commentaires
Gestionnaire	Gestionnaire des informations du système	Tous les privilèges d'un visiteur et la gestion (plante, maladies, traitement, etc.) de l'application
Administrateur	Administrateur du système	Tous les privilèges d'un visiteur et aussi la gestion des utilisateurs

## V.2. Identification des besoins fonctionnels

Les besoins fonctionnels représentent l'ensemble des actions que doit exécuter un système. Ces derniers sont organisés en fonctionnalités auxquelles le système doit répondre.

Le Tableau 4 ci-après identifie les fonctionnalités de notre système :

Tableau 4: identification de fonctionnalités

Fonctionnalité	Acteur
Authentification	Réserver aux gestionnaires du système
Gérer les utilisateurs (ajouter, modifier, supprimer)	Réservé à l'administrateur du système

Gérer une plante (ajouter, modifier et supprimer)	Réserver aux gestionnaires du système
Gérer un traitement (ajouter, modifier et supprimer)	Réserver aux gestionnaires du système
Gérer une maladie (ajouter, modifier et supprimer)	Réserver aux gestionnaires du système
Consulter les plantes	Administrateur, gestionnaires et visiteurs
Rechercher une plante	Administrateur, gestionnaires et visiteurs
Afficher les détails d'une plante	Administrateur, gestionnaires et visiteurs
Consulter les maladies	Administrateur, gestionnaires et visiteurs
Rechercher une maladie	Administrateur, gestionnaires et visiteurs
Afficher les détails d'une maladie	Administrateur, gestionnaires et visiteurs
Consulter les traitements	Administrateur, gestionnaires et visiteurs
Rechercher un traitement	Administrateur, gestionnaires et visiteurs
Afficher le contenu d'un traitement	Administrateur, gestionnaires et visiteurs
Ajouter un commentaire	Administrateur, gestionnaires et visiteurs
Visualiser les commentaires	Administrateur, gestionnaires et visiteurs

Nous allons maintenant passer aux diagrammes de cas d'utilisation afin de bien comprendre les relations entre les acteurs et les fonctionnalités du système.

### V.3. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation nous permet de répondre à la question « à quoi va servir notre système ? ». Il est utilisé en UML pendant la phase d'analyse d'un projet pour identifier les fonctionnalités du système. Il décrit l'interaction des personnes ou du dispositif externe avec le système en cours de conception. Il ne montre pas beaucoup de détails, mais résume seulement certaines des relations entre les cas d'utilisation, les acteurs et le système.

Fondamentalement, quatre éléments doivent être inclus dans un diagramme de cas d'utilisation. Il s'agit des acteurs, du système, des cas d'utilisation et des relations. Les cas

d'utilisation représentent les actions qui sont effectuées par un ou plusieurs acteurs dans un but particulier. Le système est ce que nous sommes en train de développer.

Dans la suite de ce chapitre, nous allons présenter les diagrammes de cas d'utilisation pour les fonctionnalités suivantes : gestion des plantes, gestion des traitements, gestion des maladies et gestion des utilisateurs.

#### ▪ **Diagramme de cas d'utilisation de la gestion des plantes**

Deux acteurs interviennent dans la gestion des plantes à savoir : visiteur et gestionnaire.

L'acteur gestionnaire se généralise à l'acteur visiteur car ayant accès à toutes les fonctionnalités qui lui sont autorisées. En effet, la gestion des plantes fait intervenir les cas d'utilisations suivantes :

- afficher liste plantes : qui permet de visualiser la liste de toutes les plantes stockées dans la base de données du système. Ce cas étend « rechercher une plante » ;
- rechercher une plante : qui permet de faire une recherche dans le tableau présentant la liste de toutes les plantes. Ce cas étend « afficher détails plante » et « afficher sources plante » ;
- afficher détails plante : ce cas d'utilisation permet, aussi bien aux visiteurs qu'aux gestionnaires, de visualiser les détails concernant une plante donnée notamment, son nom scientifique, ses cas d'utilisation, sa description, ses appellations dans quelques langues et sa famille d'appartenance ;
- afficher sources plante : permet de parcourir des sources fiables qui parlent de la plante en question ;
- ajouter une plante : ce cas d'utilisation est réservé aux gestionnaires du système. Il lui donne la possibilité d'ajouter une plante dans la liste existante ;
- modifier une plante : permet de modifier les informations concernant une plante ;
- supprimer une plante : permet de retirer une plante dans la base de données des plantes ;

Ces trois derniers cas d'utilisations sont réservés uniquement aux gestionnaires du système et nécessitent une authentification au préalable.

Voici, en illustration, la Figure 5 qui est une représentation du diagramme de cas d'utilisation de la gestion des plantes :

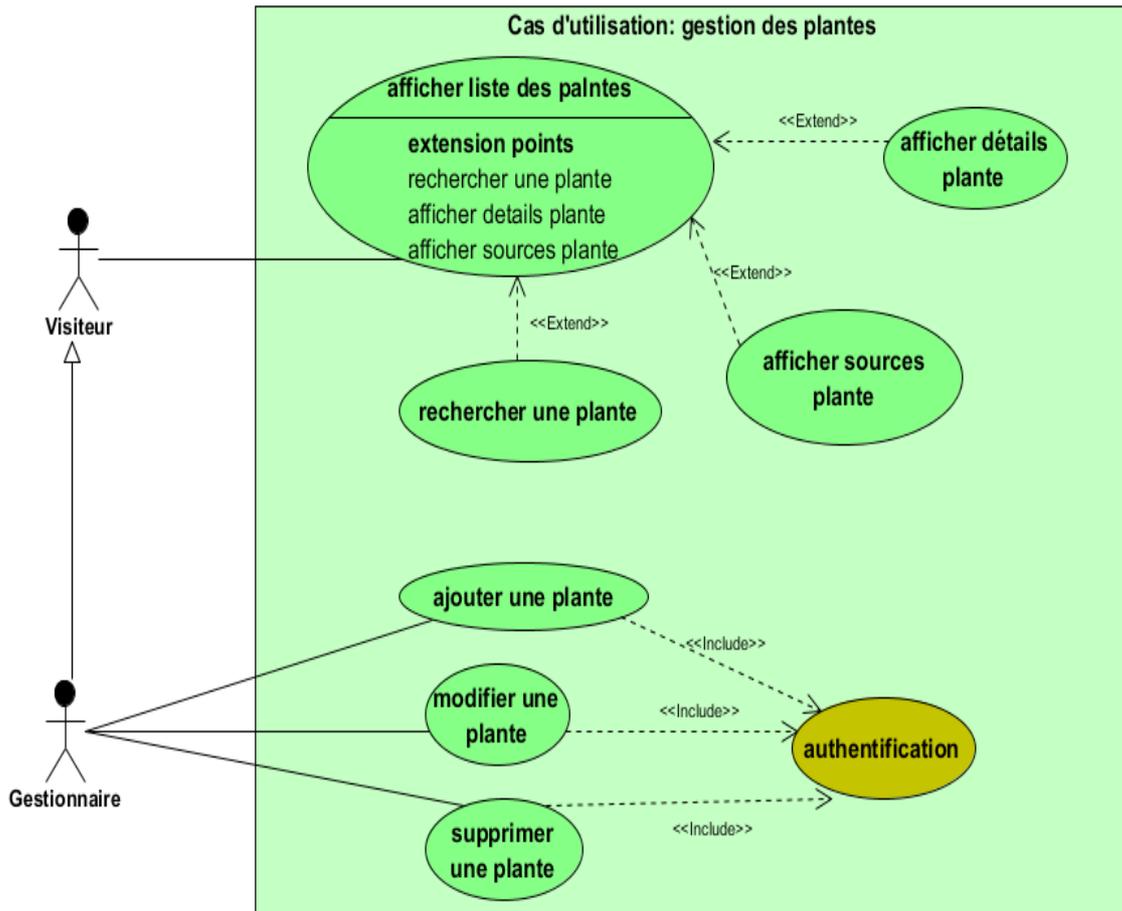


Figure 5: diagramme de cas d'utilisation de la gestion des plantes

#### ▪ Diagramme de cas d'utilisation de la gestion des traitements

Ce diagramme fait également intervenir les deux acteurs du système à savoir : le visiteur et le gestionnaire. L'acteur gestionnaire se généralise à l'acteur visiteur car ayant accès à toutes les fonctionnalités qui lui sont autorisées. Les cas d'utilisations suivants constituent ce diagramme :

- afficher liste traitements : permet d'afficher la liste de tous les traitements disponibles dans le système. Il étend le cas « rechercher traitement » ;
- rechercher un traitement : permet de faire une recherche sur les traitements disponibles. Il étend les cas « afficher détails traitement » ;
- afficher détails traitement : ce cas permet de visualiser les détails sur un traitement donné à savoir : la préparation, le mode d'emploi et toutes autres informations concernant un traitement (durée de traitement, dosage, effets indésirables, moyenne d'âge etc.). il étend le cas « rechercher maladie » ;
- rechercher maladie : permet de rechercher la maladie liée au traitement en question ;

- ajouter un traitement : qui permet d'ajouter un traitement à la liste existante ;
- modifier un traitement : permet d'apporter des modifications sur un traitement déjà enregistré ;
- supprimer un traitement : pour enlever un traitement dans la base de données des traitements du système ;

Ces trois derniers cas d'utilisations sont réservés uniquement aux gestionnaires et nécessitent une authentification.

Pour illustrer ces propos, nous allons passer la Figure 6 suivante en illustration :

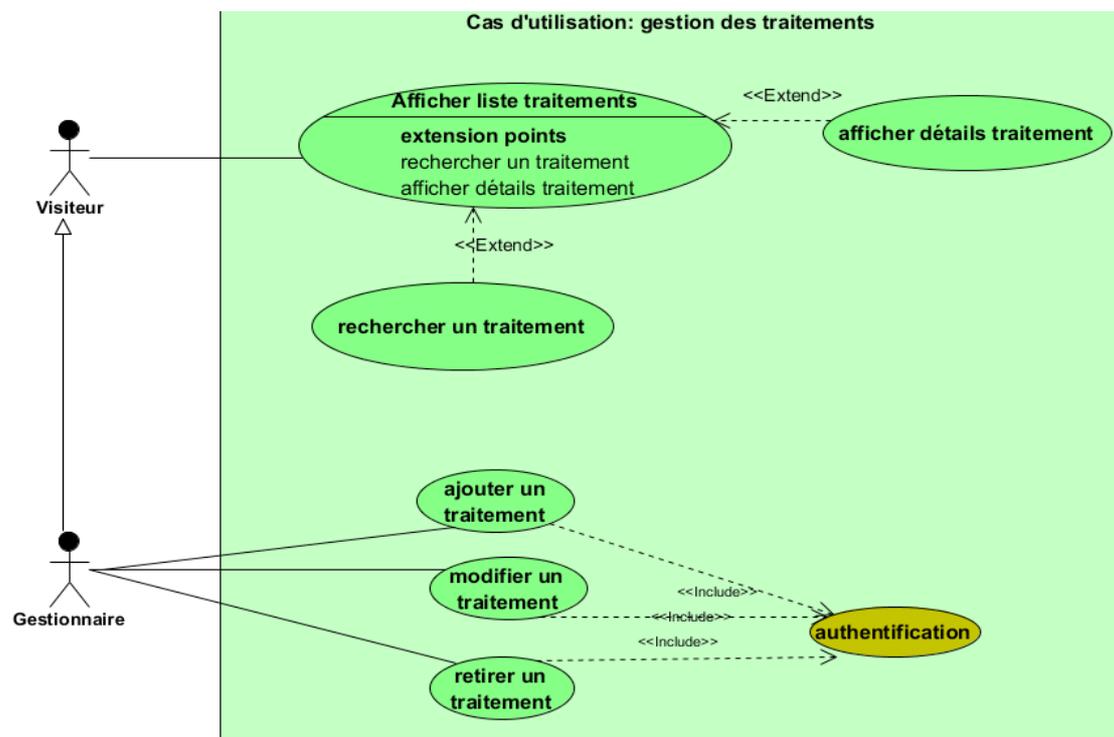


Figure 6: diagramme de cas d'utilisation de la gestion des traitements

#### ▪ Diagramme de cas d'utilisation de la gestion des maladies

Deux acteurs assurent les fonctionnalités de ce diagramme : le visiteur et le gestionnaire. L'acteur gestionnaire se généralise à l'acteur visiteur car ayant accès à toutes les fonctionnalités qui lui sont autorisées.

Il engendre les cas d'utilisations suivants :

- afficher liste maladies : pour visualiser la liste de toutes les maladies étudiées dans le système. Ce cas étend le cas « rechercher une maladie » ;

- rechercher une maladie : pour rechercher une maladie dans la liste. Il étend le cas « afficher détails maladie » ;
- afficher détails maladie : affiche les détails d'une maladie donnée (description, etc.). il étend le cas « afficher traitement » ;
- afficher traitement : permet d'afficher le(s) traitement(s) liés à la maladie ;
- ajouter une maladie : pour ajouter une maladie dans la liste des maladies existantes ;
- modifier détails maladie : pour modifier les informations concernant une maladie donnée ;
- retirer une maladie : supprimer une maladie dans la base de données des maladies ;

Ces trois derniers cas d'utilisations sont réservés uniquement aux gestionnaires et nécessitent une authentification.

Ci-dessous, la Figure 7 représentant le diagramme de cas d'utilisation de la gestion des maladies :

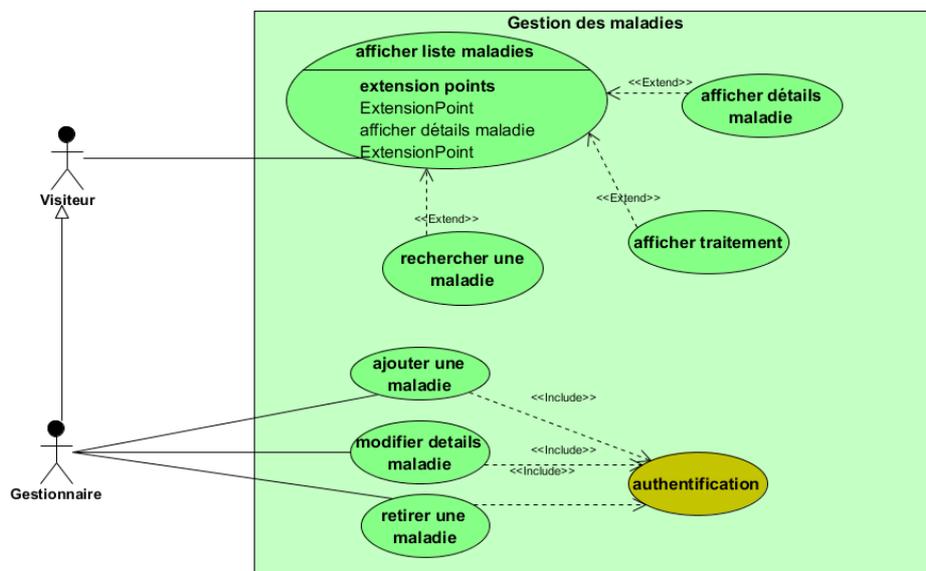


Figure 7: diagramme de cas d'utilisation de la gestion des maladies

#### ■ Diagramme de cas d'utilisation de la gestion des utilisateurs

Ce cas d'utilisation fait intervenir un seul acteur qui est l'administrateur du système. Il est constitué des cas d'utilisations suivantes :

- ajouter utilisateur : pour ajouter un nouveau utilisateur dans le système ;
- modifier utilisateur : pour modifier un utilisateur existant ;
- supprimer utilisateur : permet d'enlever un utilisateur déjà enregistré.

La Figure 8 est un schéma illustratif de ce diagramme :

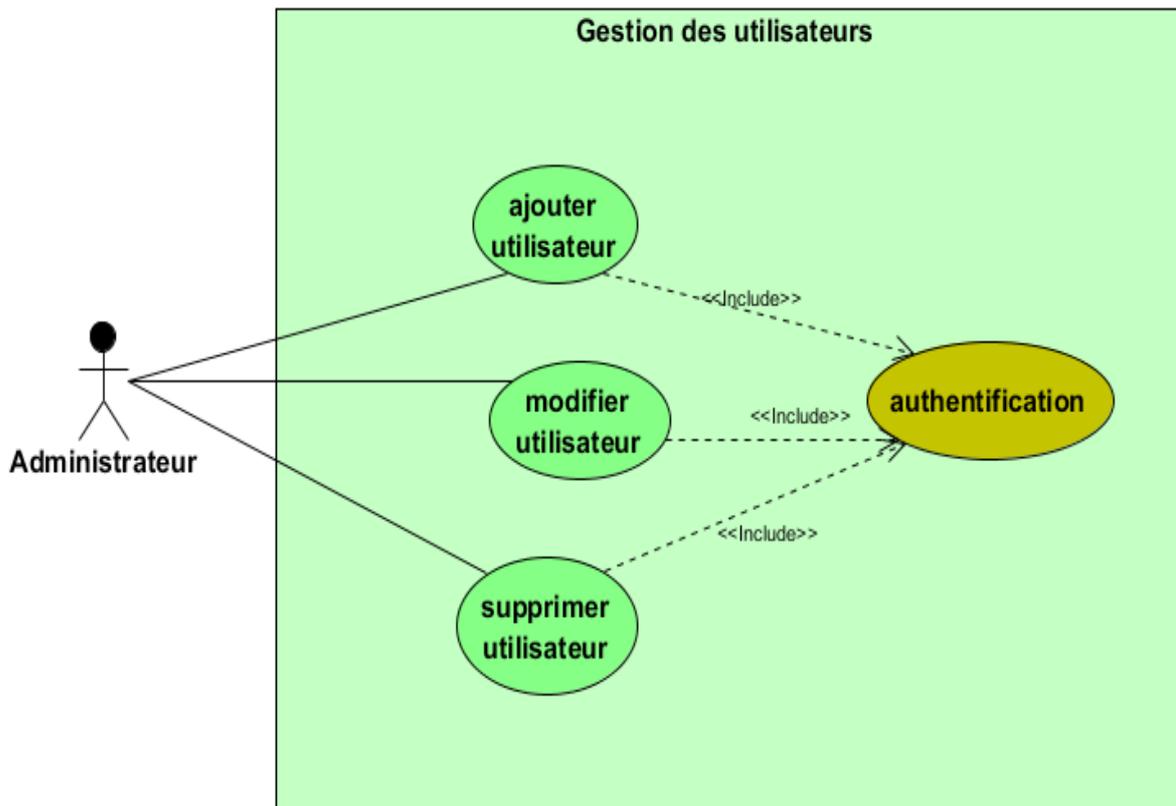


Figure 8: diagramme de cas d'utilisation de la gestion des utilisateurs

## Conclusion

Le passage de ce chapitre nous a permis d'avoir une vision plus claire des interactions qui animent notre système. En effet, la spécification des besoins fonctionnels nous montrent les intervenants du système ainsi que les privilèges de chaque type d'acteur.

Dans le chapitre suivant, nous allons voir l'analyse des besoins fonctionnels et la conception de notre application.

## Chapitre 3 : Analyse des besoins fonctionnels et conception de l'application

Le chapitre précédant nous a permis d'avoir un aperçu sur l'architecture logique de notre système ; c'est-à-dire comprendre les fonctionnalités ainsi que les interactions au sein du système.

Dans ce chapitre, nous allons étudier d'autres aspects de notre système : l'analyse des besoins fonctionnels (qui consiste en la description des fonctionnalités de l'application) et la conception de l'application. Cette dernière consiste à l'élaboration d'un ensemble d'activités bien organisées permettant de traduire un cahier de charge en un outil informatique capable de résoudre des problèmes.

Nous allons voir l'architecture de base de notre système en nous appuyant sur d'autres types de diagrammes. Pour ce faire, nous déroulerons ce chapitre sur deux grandes parties, à savoir : l'analyse de besoins fonctionnels et la conception.

### I. Analyse des besoins fonctionnels

#### V.4. Analyse de l'authentification

L'authentification est la première étape que le gestionnaire doit franchir avant de procéder à la gestion du système.

Dans la suite, nous allons faire sa description et élaborer son diagramme d'activité et de séquence.

Nous précisons que dans les diagrammes de séquences, nous avons mis les scénarios nominaux alors que dans les diagrammes d'activités, nous avons mis les scénarios nominaux et alternatifs.

##### **a) Description du cas d'utilisation « authentification »**

L'authentification est un processus dans lequel l'utilisateur doit renseigner, dans l'interface d'authentification, son identifiant et son mot de passe et cliquer sur le bouton « se connecter ». Si les informations renseignées sont correctes, l'utilisateur peut accéder à l'interface d'administration pour la gestion du système. Sinon, le système lui renvoie un message d'erreur.

Le Tableau 5 suivant nous fait une description détaillée du cas « authentification » :

Tableau 5: description de l'authentification

Description de l'authentification	
Titre	Authentification
Résumé	Permet d'avoir accès à la page d'administration du système
Acteurs	Gestionnaire, administrateur
Pré condition	Etre administrateur ou gestionnaire du système
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur saisit son identifiant et son mot de passe,</li> <li>2. Le système vérifie les informations saisies,</li> <li>3. Le système récupère le profil de l'utilisateur</li> </ol>
Post condition	Le système lui donne les privilèges de gestion
Exception	<ul style="list-style-type: none"> <li>• Champ vide</li> <li>• Identifiant ou mot de passe incorrecte</li> </ul>

### b) Diagramme d'activité du cas « authentification »

Le diagramme d'activité nous permettra de décrire le comportement dynamique du processus de l'authentification. Nous allons représenter les opérations qui s'y font et leurs conséquences sur les objets afin de décrire le déroulement de ce dernier.

Dans notre cas, notre diagramme est constitué des éléments suivants :

**nœud initial** : Un nœud initial est un nœud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Il est défini par un arc sortant et pas d'arc entrant. Graphiquement, un nœud initial est représenté par un petit cercle plein ;

**nœud d'action** : c'est un type d'élément abstrait permettant de représenter les étapes le long du flot d'une activité ;

**transitions** : Le passage d'une activité vers une autre est matérialisé par une transition. Graphiquement les transitions sont représentées par des flèches en traits pleins qui connectent les activités entre elles. Elles sont déclenchées dès que l'activité source est terminée et provoquent automatiquement et immédiatement le début de la prochaine activité à déclencher

(l'activité cible). Contrairement aux activités, les transitions sont franchies de manière atomique, en principe sans durée perceptible ;

**nœud de décision** : Un nœud de décision est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants. Ces derniers sont généralement accompagnés de conditions de garde pour conditionner le choix. Graphiquement, nous représentons un nœud de décision par un losange ;

**nœud final** : Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant ;

Ci-dessous, le schéma illustratif (Figure 9) :

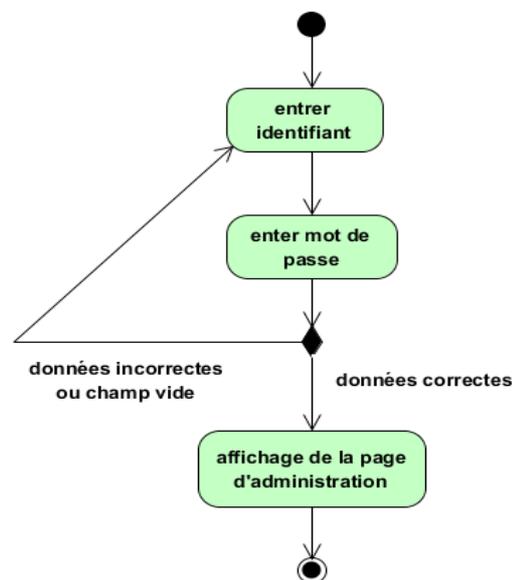


Figure 9: diagramme d'activité de l'authentification

### c) Diagramme de séquence du cas « authentification »

Le diagramme de séquences nous permet de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs. Les objets au cœur d'un système interagissent en s'échangeant des messages et les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

Dans un diagramme de séquence, une ligne de vie représente un participant à une interaction (objet ou acteur). Les principales informations contenues dans un diagramme de séquence

sont les messages échangés entre les lignes de vie : ils sont représentés par des flèches et sont présentés du haut vers le bas le long des lignes de vie, dans un ordre chronologique.

Un message définit une communication particulière entre des lignes de vie (objets ou acteurs).

Ci-après, le schéma illustratif (Figure 10) :

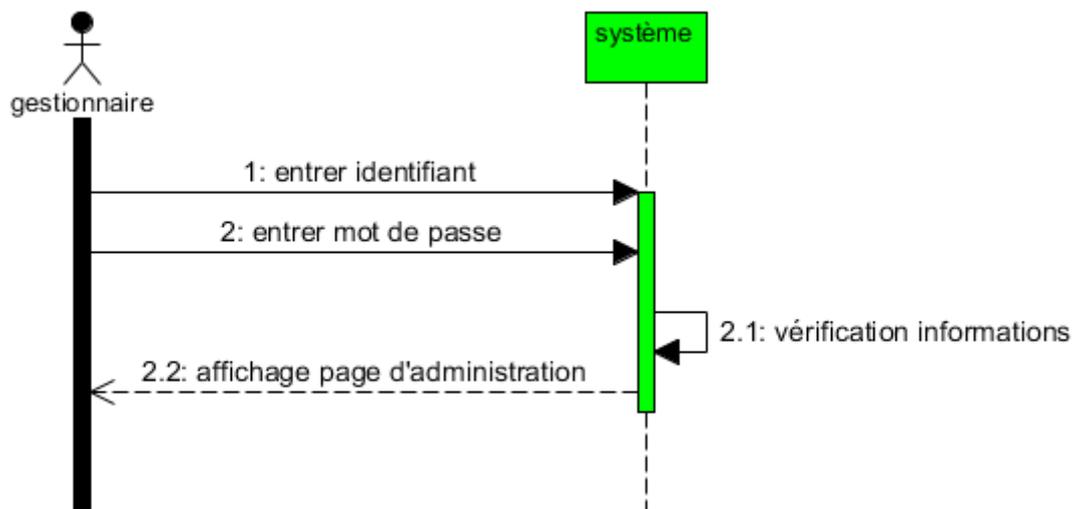


Figure 10: diagramme de séquence de l'authentification

## V.5. Analyse de la gestion des plantes

Dans cette partie, nous allons faire une description du cas d'utilisation « rechercher une plante », présenter son diagramme d'activité et son diagramme de séquence.

### a) Description du cas d'utilisation « rechercher une plante »

Cette action permet à un visiteur du système de rechercher une plante donnée dans la liste des plantes enregistrées dans le système. Si toutefois, cette plante ne figure pas dans la liste proposée à la page d'accueil, le visiteur peut entrer le nom scientifique de la plante sur le champ de recherche.

Le Tableau 6 suivant est une description détaillée de ce cas d'utilisation :

Tableau 6: description du cas rechercher une plante

<b>Description du cas d'utilisation « rechercher une plante »</b>
---

Titre	Rechercher une plante
Résumé	Permet aux visiteurs de rechercher une plante dans la liste des plantes disponibles
Acteurs	Gestionnaire, visiteur
Pré condition	Ouverture de la page d'accueil des plantes
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le système affiche une liste de plantes,</li> <li>2. La personne entre le nom de la plante dans la barre de recherche,</li> <li>3. Le système vérifie l'existence de la plante,</li> <li>4. Le système affiche la plante dans la page.</li> </ol>
Post condition	La plante est affichée
Exception	La plante n'existe pas : le système affiche un message d'erreur.

#### **b) Diagramme d'activité du cas d'utilisation « rechercher une plante »**

Ce diagramme d'activité nous permettra de comprendre le comportement du processus de recherche de plante dans notre système.

Nous allons présenter ce diagramme à la Figure 11 :

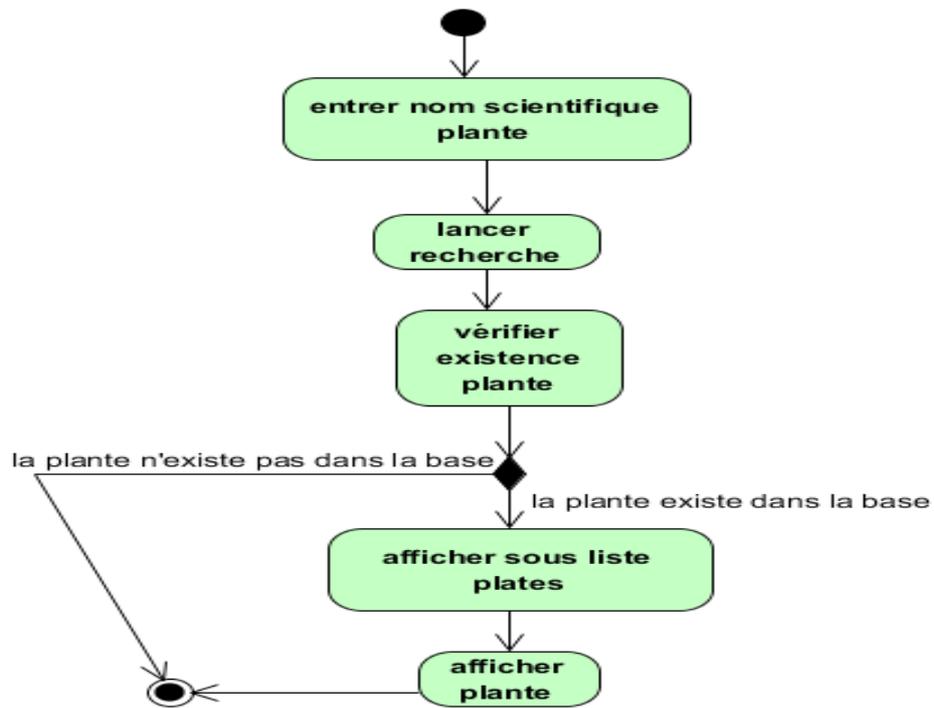


Figure 11: diagramme d'activité du cas rechercher une plante

### c) Diagramme de séquence du cas d'utilisation « rechercher une plante »

Ce diagramme nous permettra de bien définir la succession des faits déroulant dans la recherche d'une plante au sein du système après avoir fait sa présentation.

La Figure 12 suivante est une représentation du diagramme de séquence de ce cas d'utilisation :

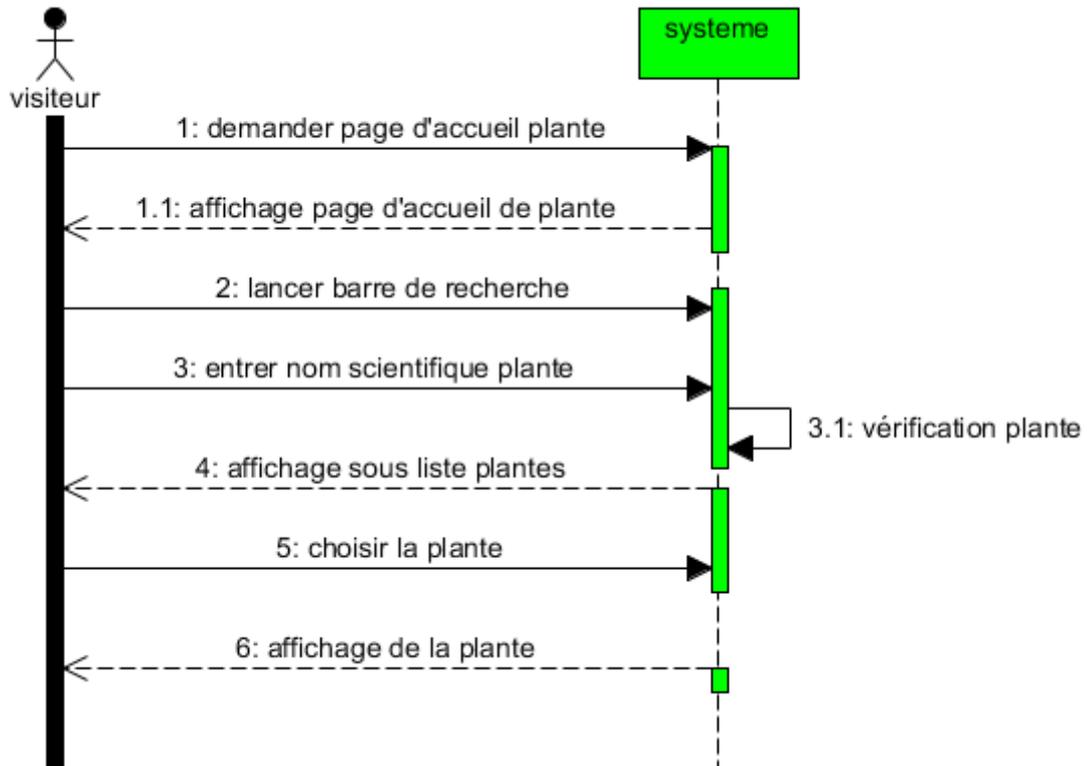


Figure 12: diagramme de séquence du cas rechercher une plante

## V.6. Analyse de la gestion des traitements

Dans cette section, nous allons présenter le diagramme de d'activité et le diagramme de séquence du cas « rechercher un traitement ».

### a) Description du cas d'utilisation « rechercher un traitement »

Cette action permet à l'utilisateur du système d'effectuer une recherche d'un traitement dans la base de données.

Le suivant est une description détaillée de ce cas d'utilisation :

Tableau 7: description du cas rechercher un traitement

Description du cas d'utilisation « rechercher un traitement »	
Titre	Rechercher un traitement
Résumé	Permet aux utilisateurs de rechercher un traitement dans la liste des traitements disponibles
Acteurs	Gestionnaire, visiteur
Pré condition	Ouverture de la page d'accueil des traitements ou affichage d'une maladie
Scénario nominal	<p>1-a. Le système affiche une liste de traitements,</p> <p>1-b. La personne entre l'objet du traitement dans la barre de recherche,</p> <p>1-c. Le système vérifie l'existence du traitement,</p> <p>1-d. Le système affiche le traitement.</p> <p>2-a. Le système affiche une maladie,</p> <p>2-b. La personne fait une recherche de traitement,</p> <p>2-c. Le système vérifie l'existence du traitement,</p> <p>2-d. Le système affiche le traitement.</p>
Post condition	Le traitement est affiché
Exception	Le traitement n'existe pas : le système affiche un message d'erreur.

**b) Diagramme d'activité du cas d'utilisation « rechercher un traitement »**

Ce diagramme d'activité nous permettra de comprendre les activités qui animent le processus de recherche d'un traitement.

A la Figure 13, nous présentons le diagramme d'activité illustrant ces propos :

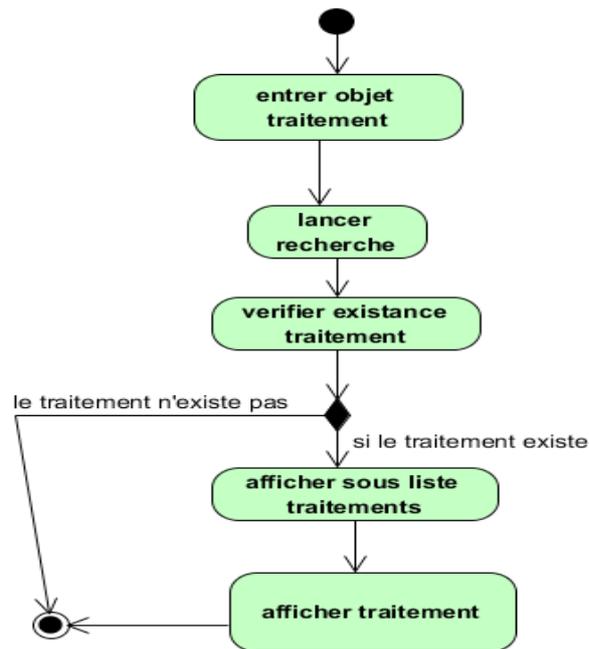


Figure 13: diagramme d'activité du cas rechercher un traitement

### c) Diagramme de séquence du cas « rechercher un traitement »

Cette représentation nous donne un aperçu sur le séquençement de la recherche d'un traitement dans le système.

A la Figure 14 ci-dessous, nous présentons le diagramme de séquence de ce cas d'utilisation :

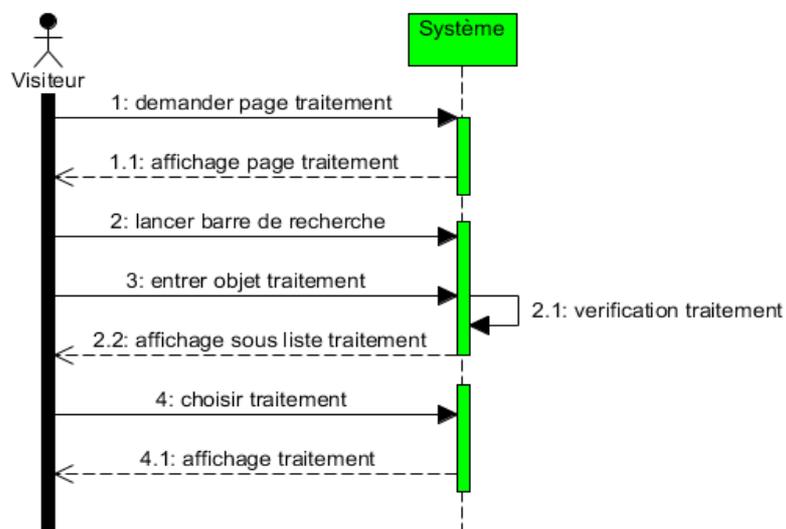


Figure 14: diagramme de séquence de la recherche de traitement

## V.7. Analyse de l'administration

L'administration fait partie des fonctionnalités les plus importantes de l'application. En effet, c'est l'administrateur qui est responsable de la gestion des utilisateurs (gestionnaires).

### a) Description de la gestion de l'administration

Le Tableau 8 suivant est une description du système de l'administration :

Tableau 8: description de la gestion de l'administration

Description de l'administration	
Titre	Gestion administration
Résumé	Permet la gestion des utilisateurs
Acteurs	Administrateur
Pré condition	Etre administrateur du système
Scénario nominal	1. L'administrateur s'authentifie, 2. Le système récupère son profil 3. L'administrateur passe à la gestion des utilisateurs
Post condition	Le système lui donne les privilèges d'administration
Exception	Il n'est pas administrateur

### b) Diagramme d'activité du cas « ajouter un utilisateur »

Cette action permet d'ajouter un utilisateur dans le système. Elle est réalisée par l'administrateur du système.

Le diagramme suivant nous montre les activités dans l'ajout d'un utilisateur (Figure 15) :

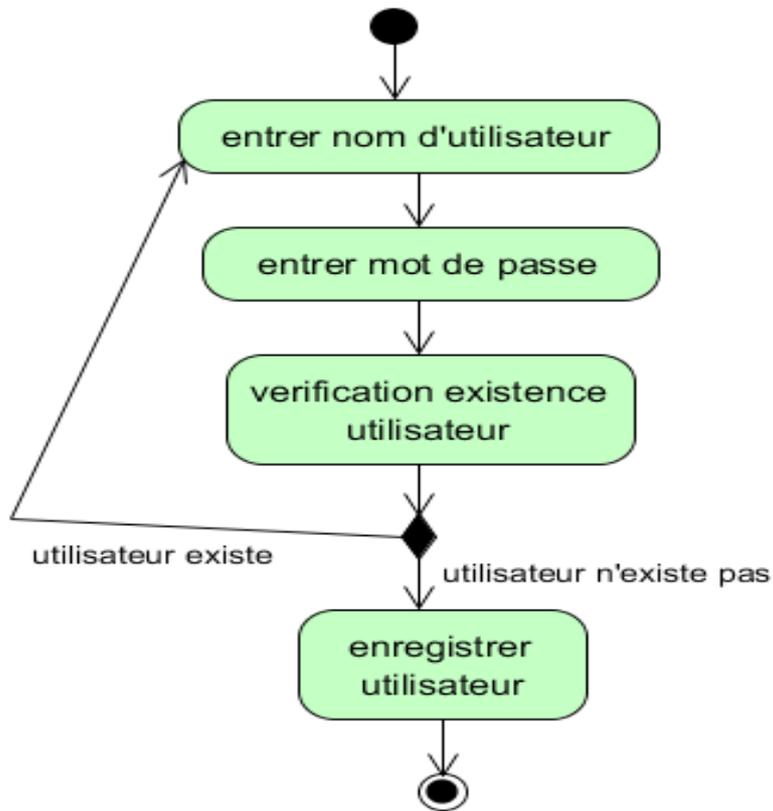


Figure 15: diagramme d'activité du cas "ajouter un utilisateur"

### c) Diagramme de séquence du cas « ajouter utilisateur »

Le diagramme de séquence définit le séquençage du processus de création d'un utilisateur.

La Figure 16 suivante est une représentation du diagramme de séquence pour ajouter un utilisateur :

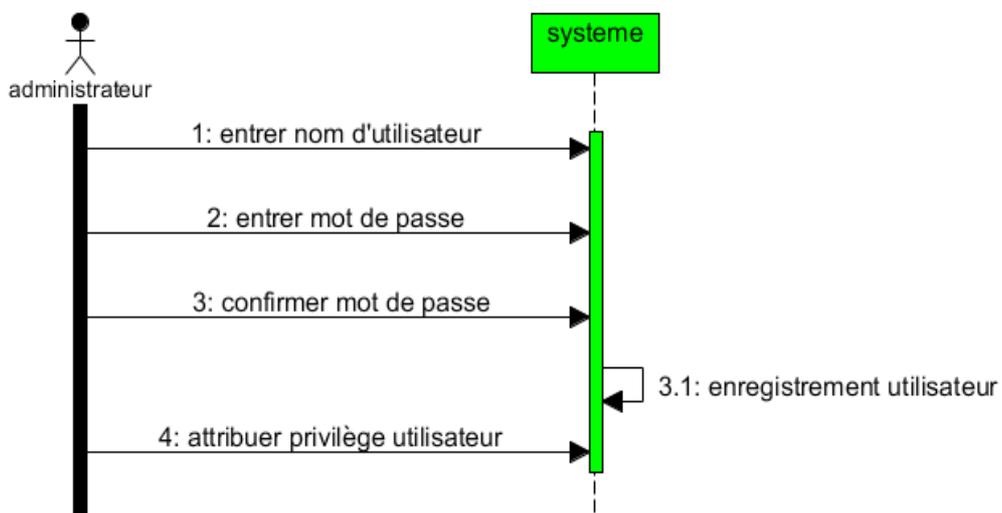


Figure 16: diagramme de séquence du cas "ajouter un utilisateur"

## VI. Conception

### VI.1. Conception générale

Dans cette première partie, nous élaborons le style d'architecture que nous allons adopter et présenter le diagramme de déploiement.

#### VI.1.1. Architecture de l'application

L'architecture d'un logiciel décrit la manière dont seront agencés les différents éléments d'une application et comment ils interagissent entre eux. Pour la conception de notre application, nous avons choisi d'adopter l'architecture 3-tiers (ou architecture à 3 niveaux). C'est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches dont le rôle est clairement défini.

Cette séparation a pour but de rendre indépendante chacune des couches afin de faciliter la maintenance et les évolutions futures de l'application. Elle assure une sécurité plus importante car l'accès à la base de données n'est autorisé que par la couche de traitements. Elle a également l'avantage d'optimiser le travail en équipe et le développement multi-cibles. Ces trois couches logicielles sont :

- la couche présentation (niveau 1) : représente la première étape sur l'architecture. Elle correspond à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur, on parle d'interface homme-machine. Elle peut être représentée en HTML, CSS, JavaScript, etc., pour être exploitée par un navigateur web ;
- la couche métier ou fonctionnelle (niveau 2) : représente le deuxième niveau de l'architecture. Elle correspond à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative. Elle est chargée de fournir la ressource en faisant appel à un autre serveur. Pour notre application, le langage serveur utilisé est python avec le Framework Django ;
- la couche d'accès aux données (niveau 3) : c'est le troisième niveau de notre architecture logicielle. Correspondant au serveur de base de données, elle fournit au serveur d'application les données dont elle a besoin. Sur ce troisième tiers, un SGBD est installé. Dans notre cas, nous travaillons avec PostgreSQL ;

La Figure 17 suivante représente une vue sur l'architecture logicielle 3-tiers :

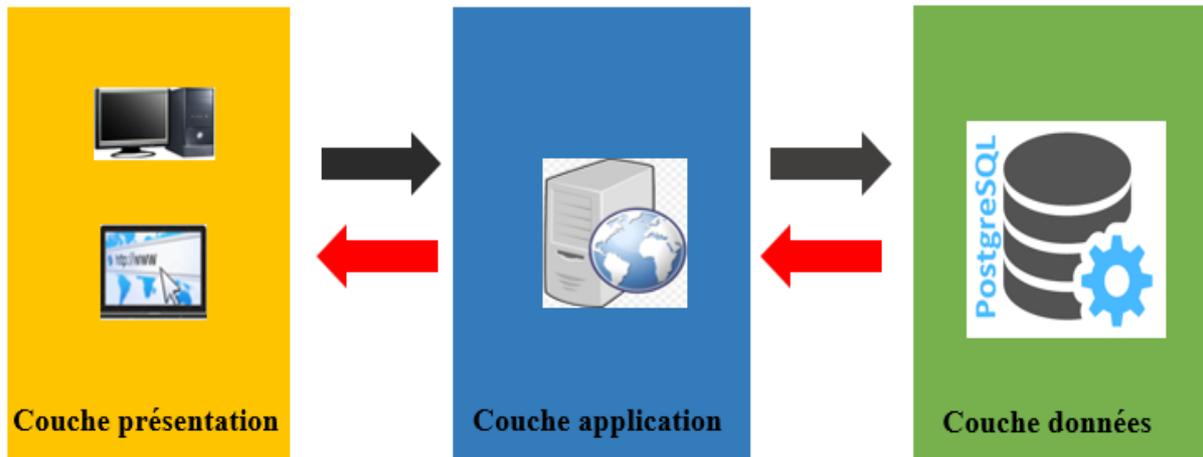


Figure 17: présentation de l'architecture

Le Framework Django est basé sur un modèle de conception de logiciel nommé le MVT (Modèle, Vue, Template) que nous avons appliqué au niveau 2 représentant la couche métier. Ce modèle est inspiré de l'architecture MVC (Modèle, Vue, Contrôleur) très répandu dans les Framework web. Son objectif est de séparer les responsabilités de chaque pôle afin que chacun se concentre sur ses tâches.

Nous présentons ci-dessous ces trois pôles en détails :

- **Le modèle** : il modèle interagit avec la base de données. Sa mission est de chercher dans une base de données les items correspondant à une requête et de renvoyer une réponse facilement exploitable par le programme.

Les modèles s'appuient sur un ORM (Object Relational Mapping, ou Mapping objet-relationnel en français).

- **La vue** : elle joue un rôle central dans un projet structuré en MVT : sa responsabilité est de recevoir une requête HTTP et d'y répondre de manière intelligible par le navigateur ;
- **Le template** : C'est un fichier HTML qui peut recevoir des objets Python et qui est lié à une vue. Il est placé dans le dossier templates.

## VI.1.2. Diagramme de déploiement

Cette modélisation décrit le déploiement physique des informations générées par le logiciel sur des composants matériels[17]. Généralement, le diagramme est constitué par les nœuds, les composants, les associations et les artefacts.

En essayant de les définir, nous pouvons dire qu'un artefact est l'information qui est générée par le logiciel, un nœud représente un composant du système, les lignes entre les nœuds indiquent les relations et un composant représente un élément du logiciel.

Voici en Figure 18 le schéma illustratif :

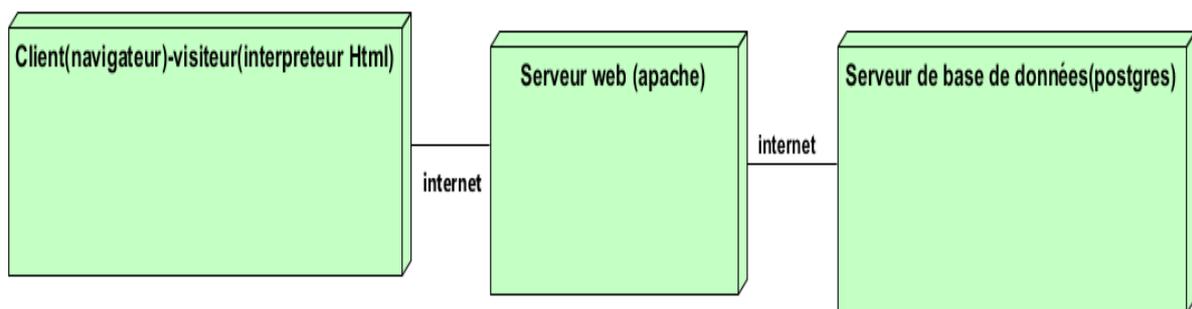


Figure 18: diagramme de déploiement

Dans la section suivante, nous présentons les résultats de conception détaillée.

## VI.2. Conception détaillée

La conception détaillée constitue la dernière phase de la conception de notre système. Dans cette partie, nous allons parler en premier lieu du dictionnaire de données et en second lieu du diagramme de classes.

### VI.2.1. Dictionnaire de données

Le dictionnaire de données va être pour nous une représentation détaillée de toutes les données traitées dans notre base. Il est défini sur les propriétés suivantes :

- 📁 table : définit le nom de la table dans notre base de données ;
- 📁 champs : désigne l'intitulé du champ dans la table ;
- 📁 description : décrit ce que représente le champ dans la table ;
- 📁 type : la valeur du champ dans la table ;
- 📁 liaison : définit les autres tables qui entretiennent une liaison avec la table en question ;

L'Annexe définit le dictionnaire de données.

## VI.2.2. Diagramme de classes

Le diagramme de classes est un diagramme structurel (statique) utilisé pour représenter les classes et les interfaces d'un système ainsi que les différentes relations entre celles-ci.

Par définition, une classe est une représentation abstraite d'un ensemble d'objets et contient les informations nécessaires à la construction de l'objet (c'est-à-dire la définition des attributs et des méthodes)[18].

Schématiquement, le diagramme de classe est constitué des éléments suivants :

- les classes (attributs + méthodes) ;
- les associations (relations) entre les classes.

Il est considéré comme le plus important des diagrammes car il définit la structure de notre base de données.

A la Figure 19 suivante, nous présentons le diagramme de classes globales de notre système :

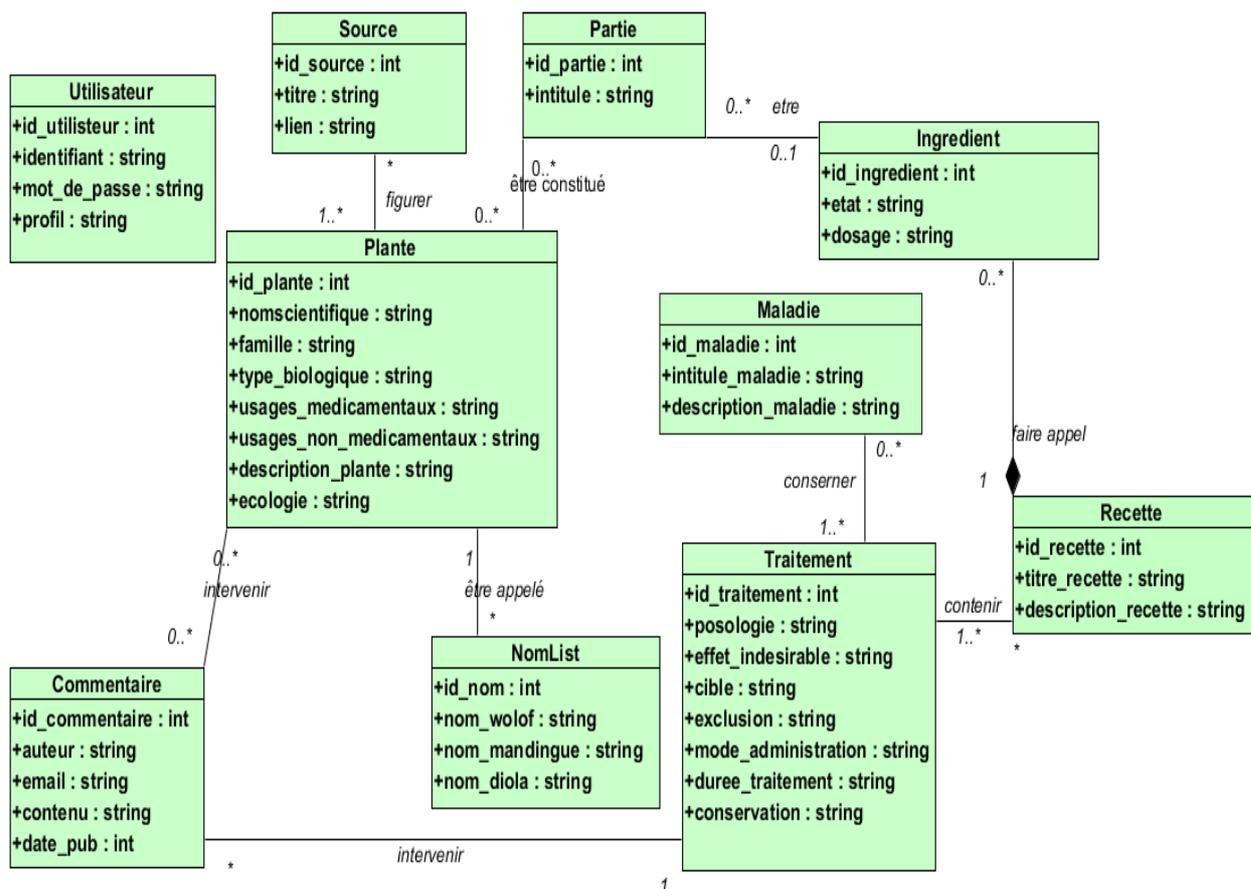


Figure 19: diagramme de classe

## Conclusion

En résumé, le chapitre 3 nous a permis de comprendre le déroulement des fonctionnalités de l'application. Par la suite, nous avons étudié l'architecture de notre solution informatique ainsi que les outils de déploiement nécessaires. En outre, nous avons aussi présenté le diagramme de classes qui constitue une représentation statique des classes de notre application.

Ainsi, nous allons passer à l'implémentation et à la présentation de quelques interfaces de l'application dans le chapitre suivant.

## Chapitre 4 : Implémentation et présentation de l'application

Ce dernier chapitre de notre document est consacré entièrement à l'implémentation et à la présentation de notre application. Pour se faire, nous allons passer au processus de mise en place de notre base de données et présenter quelques interfaces de notre solution.

Toutefois, la réalisation de ce projet a nécessité l'utilisation de certains outils et technologies que nous allons présenter dans la première partie de ce chapitre.

### I. Outils et technologies utilisées

La réalisation de notre projet de gestion des plantes médicinales, a nécessité l'utilisation de plusieurs outils et technologies. Ces derniers se présentent sous différentes natures et nous ont rendu la tâche plus aisée. Sur cette lancée, nous allons passer à la présentation des différents outils et technologies utilisés dans le travail.

#### VI.3. Environnement de développement : PyCharm

En programmation informatique, L'environnement de développement regroupe l'ensemble des outils de gestion informatique qui participent à la conception et la création de systèmes informatiques[9]. Dans notre projet nous choisissons PyCharm.

PyCharm est un outil développé par la société JetBrains, spécialiste du développement d'IDEs pour différents langages de programmation. Il présente des outils comme : un débogueur et un testeur intégrés, un profileur Python, un terminal intégré, une intégration avec les systèmes de contrôles de versions (VCS) majeurs et les outils de base de données intégrées, des capacités de développement à distance avec des interpréteurs distants, un terminal ssh intégré et une intégration avec Docker et Vagrant. Il fonctionne sur Windows, MacOS ou Linux et se présente sous deux versions : la version Communautaire (et gratuite) et la version Professionnelle (qui est payante et soumise à licence commerciale).

Dans notre travail, nous utilisons la version communautaire.



Figure 20: logo de PyCharm

## VI.4. Langage de programmation : Python Django

Un langage de programmation est un langage informatique permettant à un être humain d'écrire un code source qui sera analysé par une machine, généralement un ordinateur[10].

Django est un Framework Web Python de haut niveau, gratuit et open source, qui encourage un développement rapide et une conception propre et pragmatique. Construit par des développeurs expérimentés, il prend en charge une grande partie des tracas du développement Web. Il présente une documentation riche et organisée et également une large communauté.

En outre, sa particularité est qu'il est :

- **ridiculement rapide** : il est conçu pour aider les développeurs à faire passer les applications du concept à la réalisation le plus rapidement possible ;
- **d'une sécurité rassurante** : Django prend la sécurité au sérieux et aide les développeurs à éviter de nombreuses erreurs de sécurité courantes ;
- **excessivement évolutif** : certains des sites les plus fréquentés du Web tirent parti de la capacité de Django à évoluer rapidement et de manière flexible ;
- **d'détenteur d'une interface d'administration** ;



Figure 21: logo de Django

## VI.5. Langage de balisage : HTML

En informatique, un langage de balisage est un langage servant à définir et à structurer les informations contenues dans un document[11]. Nous avons choisi de travailler avec HTML (HyperText Markup Language en anglais) qui reste sans doute le langage de balisage le plus connu.

HTML est un langage informatique de balisage standard pour les documents conçus pour être affichés dans un navigateur web. Il décrit la structure d'une page web de manière sémantique et inclut à l'origine des indices pour l'apparence du document. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page (CSS) et/ou ses fonctionnalités interactives (JavaScript).

Le langage HTML utilise des « balises » pour annoter du texte, des images et d'autres contenus afin de les afficher dans un navigateur web. Ces balises sont insensibles à la casse et peuvent aussi bien être écrites en minuscules, en majuscules voir avec un mélange des deux[12].



Figure 22: schéma de HTML

## VI.6. Langage de requêtes : SQL

Un langage de requête est un langage informatique utilisé pour accéder aux données d'une base de données ou d'autres systèmes d'informations. Nous avons choisi SQL qui est un langage standard pour stocker, manipuler et récupérer des données dans des bases de données[13]. Il peut être utilisé dans : MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, PostgreSQL et d'autres systèmes de bases de données. Dans notre projet, nous utilisons Postgres.



Figure 23: schéma de SQL

## VI.7. Framework : Bootstrap

Un Framework (ou infrastructure logicielle en français) est un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application permettant de simplifier et d'uniformiser le travail des développeurs.

Bootstrap est un Framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce Framework utilise les langages HTML, CSS, JavaScript et fournit aux développeurs des outils pour créer un site facilement. Ce Framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de Framework un "Front-End Framework"[14].



Figure 24: schéma de Bootstrap

## VI.8. SGBDR : PostgreSQL

Un système de gestion de base de données relationnelles (SGBDR) est un logiciel standard qui repose sur les principes du modèle relationnel[15].

PostgreSQL est un puissant système de base de données relationnelle objet open source avec plus de 30 ans de développement actif qui lui a valu une solide réputation de fiabilité, de robustesse des fonctionnalités et de performances[16]. Il propose une documentation et est soutenu par une communauté active. Il est disponible pour Windows , Linux, FreeBSD et Open BSD.



Figure 25: schéma de PostgreSQL

## VII. Création et structure du projet

La création de notre projet commence par l'installation de Django dans notre ordinateur, ensuite, nous avons lancé l'outil Pycharm et nous y avons créé un nouveau projet nommé **monMemoire**. Après avoir créé le projet, nous avons lancé le terminal de Pycharm, situé en bas à gauche. C'est dans ce terminal où nous avons exécuté les commandes.

Pour démarrer le projet, nous avons saisi la commande `django-admin startproject monMemoire`; ce qui a généré automatiquement dans le répertoire de notre projet les fichiers et les dossiers suivants :

- Le sous répertoire **monMemoire/** : correspond au package python effectif. C'est le nom que l'on va utiliser pour importer ce qu'il contient ;
- **manage.py** : qui se trouve dans la racine du projet, c'est un utilitaire en ligne de commande qui permet d'interagir avec le projet ;
- **monMemoire/\_init\_.py** : c'est un fichier vide qui indique à python que le répertoire doit être considéré comme un package ;

- **monMemoire/settings.py** : fichier de réglages et de la configuration du projet ;
- **monMemoire/urls.py** : il contient les déclarations des URL, une sorte de table des matières de l'application ;
- **monMemoire/asgi.py** : un point d'entrée pour les serveurs Web compatibles aSGI pour le déploiement de l'application ;
- **monMemoire/wsgi.py** : définit un point d'entrée pour tous les serveurs web compatibles WSGI pour servir le projet.

La Figure 26 montre la structure de notre projet après sa création.

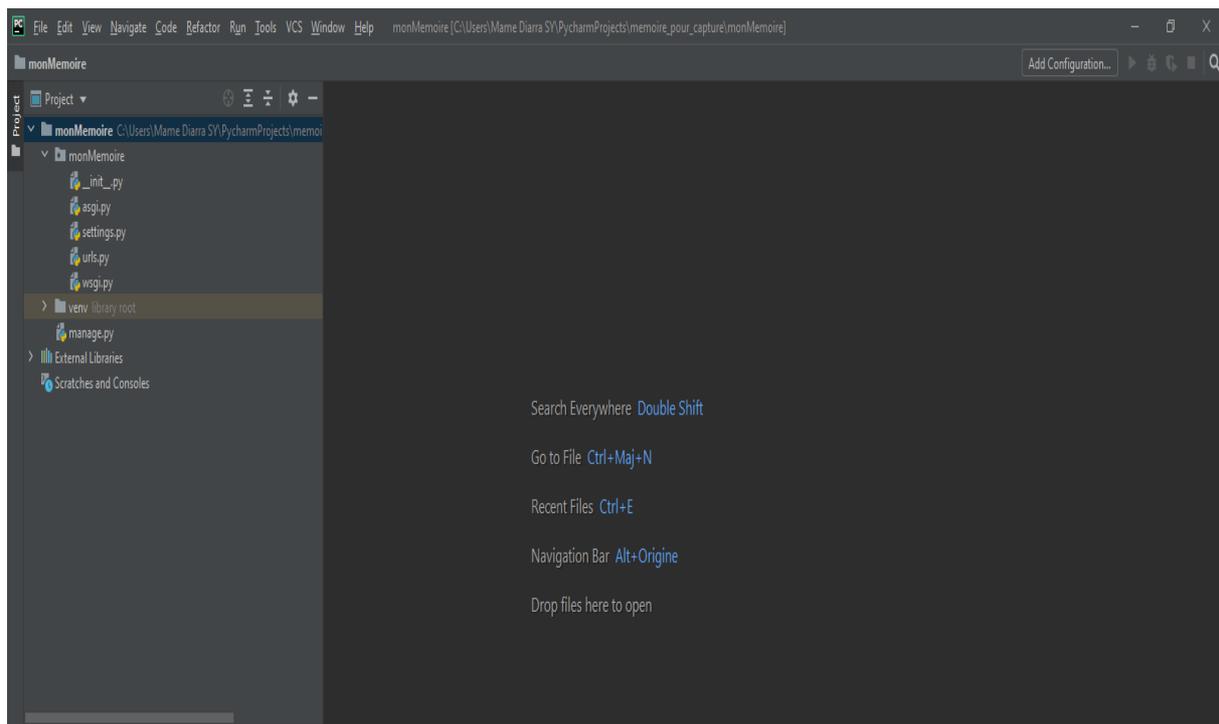


Figure 26: structure du projet à la création

Pour vérifier que notre projet fonctionne, nous nous sommes déplacés dans le répertoire monMemoire/ et nous avons lancé le serveur avec la commande `python manage.py runserver`.

La Figure 27 nous montre que notre projet fonctionne bien :

```
Run 'python manage.py migrate' to apply them.
February 19, 2022 - 10:38:25
Django version 4.0.2, using settings 'monMemoire.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figure 27: résultat du serveur

Après la vérification, nous sommes passés à la création de notre application « plante » avec la commande : `django-admin startApp plante`. Cette application a généré tout notre projet et génère initialement les fichiers suivants :

- `models.py` : fichier destiné à accueillir les modèles de l'application ;
- `views.py` : fichier destiné à accueillir les contrôleurs de l'application ;

Ainsi, toutes les applications créées dans le projet doivent être déclarées dans la section `INSTALLED_APPS` de `settings.py` et les views dans les fichiers `urls.py`.

La Figure 28 montre l'aspect de notre application après sa création :

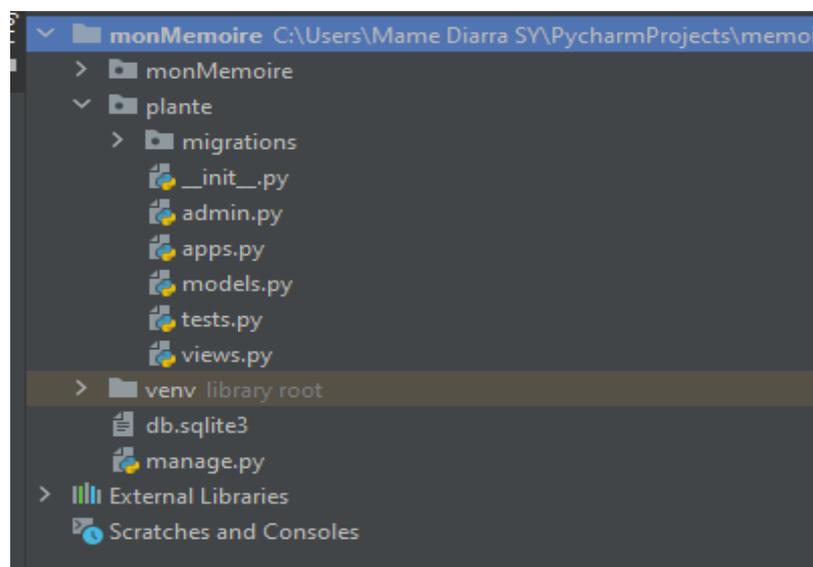


Figure 28: structure de l'application « plante »

La création de notre application marque la dernière étape des configurations de base. Nous sommes ensuite passés à l'implémentation de notre base de données.

## VII.1. Implémentation de la base de données

Pour faire l'implémentation de notre base de données, nous allons passer au modèle logique de données et à la création de la base de données dans notre système.

### VII.1.1. Modèle logique de données

Dans cette section nous décrivons la structure des données que nous traitons dans notre application. Ces données sont implémentées dans le SGBDR PostgreSQL. Dans notre cas, les données sont manipulées avec le langage serveur SQL.

La Figure 29 est une représentation de notre modèle logique de données (MLD) :

Plante (id\_plante, nomscientifique, famille, utilisation, description, planteImage, #source, #list\_nom, #commentPlante)

Source (id\_source, titre, lien)

NomList (id\_nomList, entite, langue, nom\_langue)

Maladie (id\_maladie, description\_maladie, #list\_nom)

Ingredient (id\_ingredient, etat, dosage, #list\_partie)

Partie (id\_partie, intitule, #partie\_plante)

Recette (id\_recette, description\_recette)

Traitement (id\_traitement, posologie, effet\_indesirable, cible, exclusion, mode\_administration, duree\_traitement, conservation, #objet\_maladie, #commentTraitement, #recette)

Commentaire (id\_commentaire, auteur, email, contenu, date\_pub)

Figure 29: modèle logique de données

### VII.1.2. Création et configuration de la base de données

Pour la création de la base de données, nous avons lancé pgAdmin 4 qui est le serveur de postgresSQL avec notre mot de passe « root » (voir Figure 30) et ensuite nous avons accédé au serveur avec un autre mot de passe « root » (voir Figure 31) pour y créer notre base de données.

Les figures ci-dessous justifient ces propos :

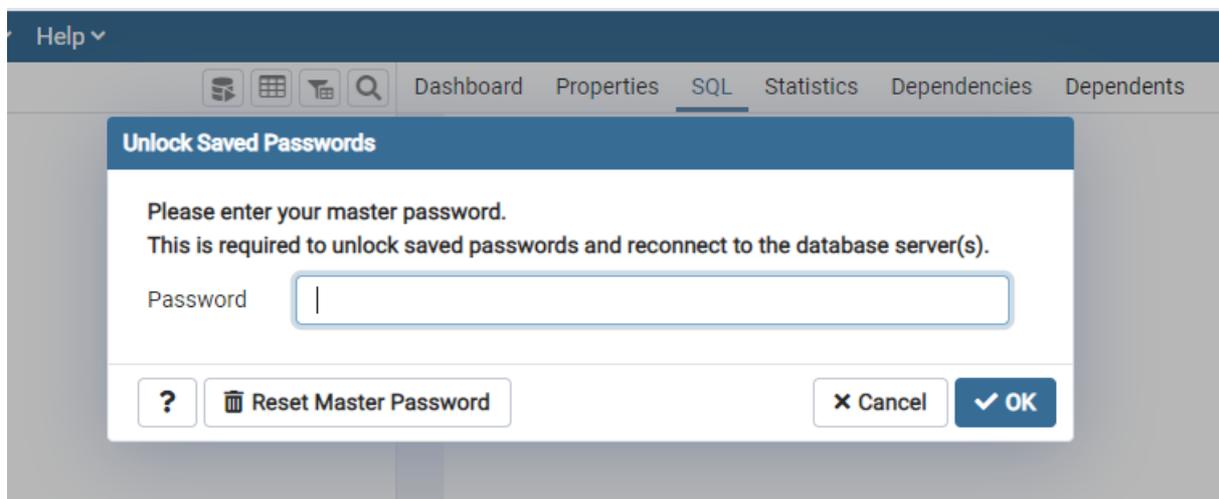


Figure 30: interface d'accès à pgAdmin

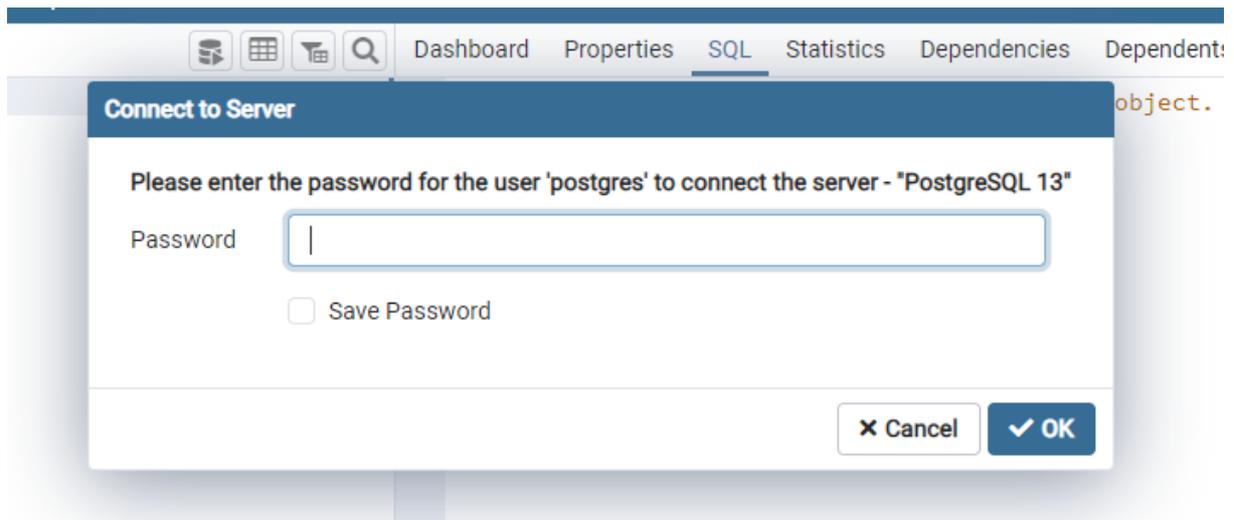


Figure 31: interface d'accès au serveur

Toutefois dans la configuration au niveau de l'IDE PyCharm nous avons renseigné la section « DATABASES », se trouvant dans le fichier « settings.py », en donnant :

- plantebd : comme nom de la base de données ;
- Mame\_Bouso : comme nom d'utilisateur ;
- bouso : comme mot de passe de l'utilisateur qui s'est connecté ;
- 5432 : comme le numéro de port d'accès au serveur ;

Ce qui nous a permis de nous connecter à notre base de données.

La Figure 32 suivante nous montre la configuration de la base de données :

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'plantebd',  
        'USER': 'bouso',  
        'PASSWORD': 'boussopw',  
        'HOST': 'localhost',  
        'PORT': '5432',  
    }  
}
```

Figure 32: configuration d'accès à la base de données

Après avoir établi la connexion à notre base de données, nous sommes passés à la création des tables. Dans la suite, nous présentons le processus de création d'une table.

Toute classe créée dans notre modèle correspond à une table dans la base de données et les attributs de la classe correspondent aux champs de la table. En outre, la migration des clés se définit également à l'intérieur de la classe. Les classes se définissent dans le fichier `models.py`.

La Figure 33 correspond au script de création de la classe « Plante » :

```
#classe plante.
class Plante(models.Model):
    nomscientifique = models.CharField(max_length=100, null=False, default='')
    famille = models.CharField(max_length=50, null=True)
    utilisation = models.TextField(max_length=500)
    description_plante = models.TextField(max_length=500)
    planteImage = models.ImageField(upload_to='images/', null=True)

    source = models.ForeignKey('SourceList', null=False, on_delete=models.CASCADE)
    list_nom = models.ForeignKey('NomList', null=False, on_delete=models.CASCADE)
    commentPlante = models.ForeignKey('Commentaire', on_delete=models.CASCADE, null=True)
```

Figure 33: création de la classe «Plante»

Après la création de la classe, celle-ci doit être exporté vers la base de données **plantebd**.

La commande `python manage.py mikemigrations` examine tous les modèles disponibles et crée des migrations pour les tables qui n'existent pas déjà. Ainsi, la commande `python manage.py migrate` exécute ces migrations et crée les tables dans la base de données, tout en fournissant éventuellement un contrôle de schéma beaucoup plus riche.

Dans la section suivante, nous présentons l'implémentation de l'application.

## VII.2. Implémentation de l'application

Dans cette partie, nous allons faire la présentation de quelques sections de codes en capture.

### VII.2.1. Code de la classe « Traitement » : modèle

Cette classe permet de créer la table « traitement » dans la base avec toutes ses liaisons. Ici, nous voyons que cette classe établit des relations manyToMany avec les classes « Maladie », « Commentaire » et « Recette ».

La Figure 34 nous montre sa structure :

```
class Traitement(models.Model):
    effet_indesirable = models.CharField(max_length=300, null=True)
    posologie = models.CharField(max_length=150)
    cible = models.CharField(max_length=200, null=True)
    exclusion = models.CharField(max_length=200, null=True)
    mode_administration = models.CharField(max_length=200, null=True)
    duree_traitement = models.CharField(max_length=100, null=True)
    conservation = models.CharField(max_length=200, null=True)

    objet_traitement = models.ForeignKey('Maladie', null=True, on_delete=models.CASCADE)
    recette = models.ForeignKey('Recette', null=True, on_delete=models.CASCADE)

    def __str__(self):
        return self.objet_traitement.intitule_maladie
```

Figure 34: déclaration de la classe « Traitement »

## VII.2.2. Codage du fichier urls.py de l'application

C'est dans ce fichier que se déclarent tous les liens vers nos différentes interfaces de l'application.

La Figure 35 est une représentation de ce fichier :

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
    path('plante', views.plante),
    path('source', views.sourceList),
    path('nom', views.nomList),
    path('maladie', views.maladie),
    path('ingredient', views.ingredient),
    path('partie', views.partie),
    path('recette', views.recette),
    path('traitement', views.traitement),
    path('traitementDetails', views.traitementDetails),
    path('comment', views.commentaire),
]
```

Figure 35: fichier urls.py

### VII.2.3. Codage du fichier admin.py

Dans ce fichier, nous déclarons toutes les classes que nous souhaitons rendre visible dans l'interface d'administration. Dans cette section, nous y avons managé beaucoup de fonctionnalités notamment l'affichage, la recherche, le tri, etc.

La Figure 36 montre la définition de quelques classes dans le fichier admin.py :

```
@admin.register(Recette)
class Recette(admin.ModelAdmin):
    list_display = ('id', 'description_recette',)

@admin.register(Traitement)
class Traitement(admin.ModelAdmin):
    list_display = ('id', 'posologie', 'effet_indesirable', 'cible', 'exclusion', 'mode_administration', 'duree_traitement', 'conservation')

@admin.register(Commentaire)
class Commentaire(admin.ModelAdmin):
    list_display = ('id', 'auteur', 'contenu', 'email', 'date_pub', 'active')
    list_filter = ('active', 'date_pub')
    search_fields = ('auteur', 'email', 'contenu')
    actions = ['approve_comment']

    def approve_comment(self, request, queryset):
        queryset.update(active=_True)
```

Figure 36: fichier admin.py

Maintenant, nous allons passer à la présentation de quelques interfaces de l'application.

## VII.3. Présentation de quelques interfaces de l'application

Les interfaces graphiques forment une partie très importante pour notre application. C'est à travers les interfaces que les utilisateurs pourront interagir avec notre système et exploiter toutes les fonctionnalités selon les privilèges de chacun d'eux.

### VII.3.1. Accès public

Dans cette partie, nous allons présenter les interfaces qui sont libres d'accès pour tous les visiteurs de notre application.

#### VII.3.1.1 Page d'accueil

La page d'accueil est la première interface à laquelle nous accédons quand on lance l'application. Elle est assez simple et lisible et permet ainsi de naviguer entre les autres pages de l'application.

Elle est présentée par la Figure 37.

## Les plantes médicinales

Bienvenue dans votre espace de documentation et de partage sur l'utilisation des plantes médicinales.

Les plantes médicinales sont des plantes utilisées pour leurs propriétés particulièrement bénéfiques pour la santé humaine voire animale. Elles se distinguent aujourd'hui par une variété de leurs secteurs d'utilisation : cosmétiques, compléments alimentaires, alimentation animale ou encore médecine vétérinaire. A travers cette application, nous allons partager, sur différentes interfaces, des connaissances sur l'utilisation des plantes médicinales (utilisations thérapeutiques).



**BAOBAB**  
ADANSONIA DIGITATA

**BIENFAITS**

- riche en calcium
- riche en potassium
- riche en vitamines A, B, C
- riche en Fer
- antioxydant
- riche en fibre
- anti-inflammatoire
- fibre
- dysenterie
- diarrhée
- etc...

**UTILISATION**

- Environ 200 g de graines dans 1 litre d'eau. Laisser macérer 3 heures, détacher la pulpe des graines.

**Filter**  
Rajouter 1 boîte de lait concentré non sucré, sucre et arôme selon le goût. Boire très frais.

- Sucrer directement les graines



Figure 37: page d'accueil de l'application

### VII.3.1.2 Page d'accueil des plantes

La page d'accueil des plantes est la première interface qui est accessible sur les plantes médicinales. Nous pouvons la joindre par la page d'accueil de l'application ou par les autres pages par l'intermédiaire de la barre de navigation. Elle présente trois boutons à savoir : « détails », « sources » et « ajouter commentaire » dont chacun d'eux présente des informations comme le prétend leur nom.

Nous voulons ainsi présenter à la Figure 38 la page d'accueil des plantes :

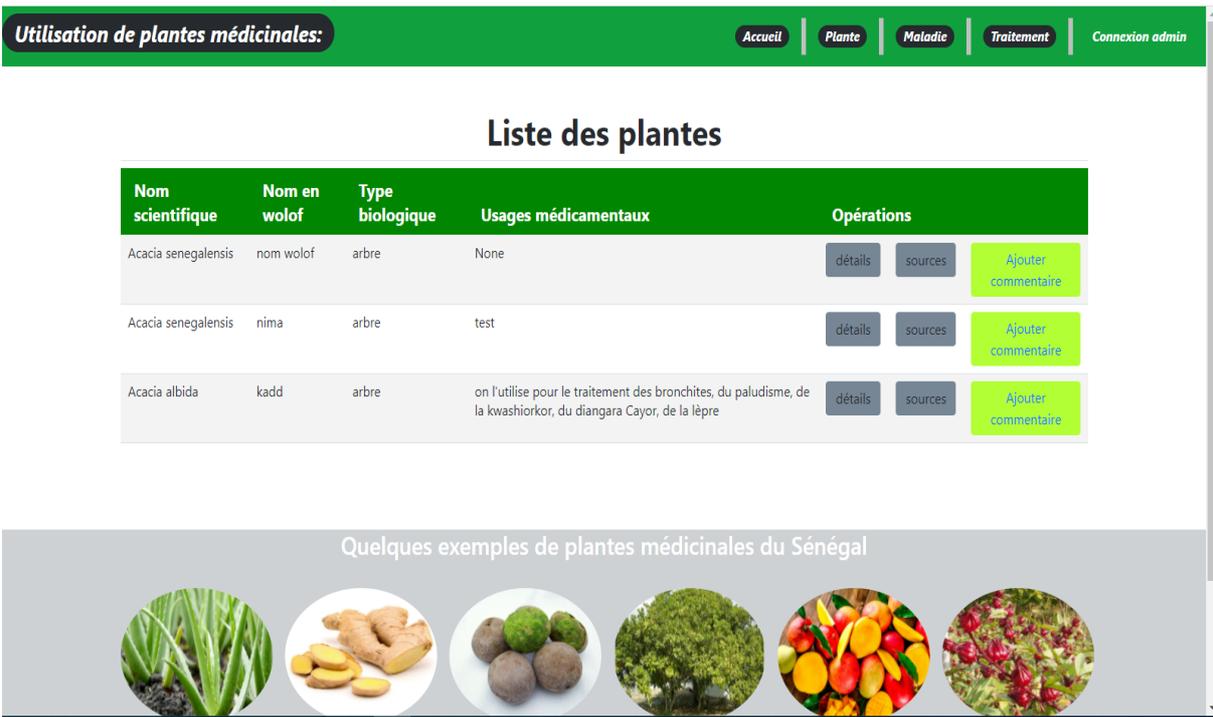


Figure 38: page d'accueil des plantes

### VII.3.1.3 Détails des plantes

Cette partie permet de visualiser les détails sur une plante c'est-à-dire afficher toutes les autres informations de la plante. Ainsi, les détails sont accessibles à partir d'un bouton sur la page d'accueil des plantes.

Cette interface est représentée par la Figure 39 :

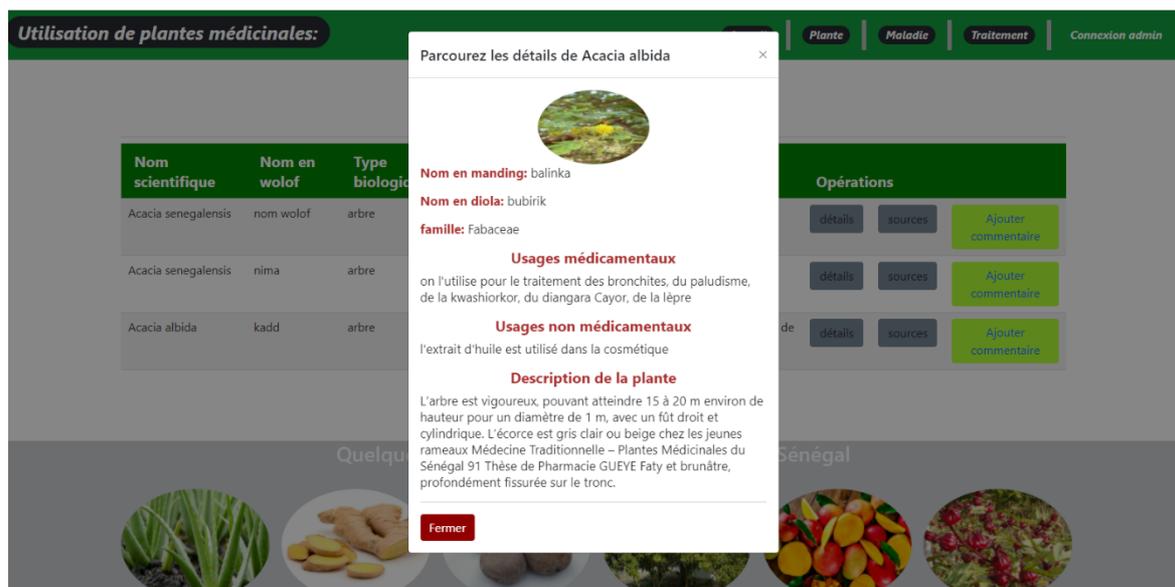


Figure 39: détails des plantes

### VII.3.1.4 Page d'accueil des maladies

Cette interface permet de visualiser les maladies qui sont disponibles dans l'application. Cette interface contient un bouton au clic duquel nous pouvons accéder à tous les traitements de cette maladie.

La Figure 40 présente l'interface de la page d'accueil des maladies :

Maladie	Description de la maladie	Opérations
paludisme	maux de tete, vomissement	<a href="#">Voir les traitement</a>
diabète	amaigrissement, prise de poids	<a href="#">Voir les traitement</a>
douleur de dos	douleur de dos	<a href="#">Voir les traitement</a>
cancer	maladie cutané	<a href="#">Voir les traitement</a>
diarrhée	maladie cutané	<a href="#">Voir les traitement</a>

Quelques exemples de plantes médicinales du Sénégal

- L'Aloe vera
- Le gingembre
- Le detarium
- Le baobab
- La mangue
- L'Hibiscus ou bissab

Figure 40: page d'accueil des maladies

### VII.3.1.5 Page d'accueil des traitements

Cette interface présente les traitements qui sont proposés dans le système. On peut y effectuer une recherche et éventuellement afficher les détails de chaque traitement.

Dans la Figure 41, nous présentons l'interface d'accueil des traitements :

Objets	Cible	posologie	Opération
cancer	enfant	utilisé pour traitement de stress durée: deux mois ...	<a href="#">détails</a>
cancer	plus de 30 ans	traitement de la peau	<a href="#">détails</a>
diabète	adulte	posologie	<a href="#">détails</a>

Showing 1 to 3 of 3 entries

Previous 1 Next

Quelques exemples de plantes médicinales du Sénégal

- L'Aloe vera [En savoir plus](#)
- Le gingembre [En savoir plus](#)
- Le detarium senealense ou ditax
- Le baobab [En savoir plus](#)
- La mangue [En savoir plus](#)
- L'Hibiscus ou bissab [En savoir plus](#)

Figure 41: page d'accueil des traitements

### VII.3.1.6 Détails traitement

Cette interface est accessible par un clic sur le bouton « détails » dans l'interface traitement.

La Figure 42 nous montre sa structure.

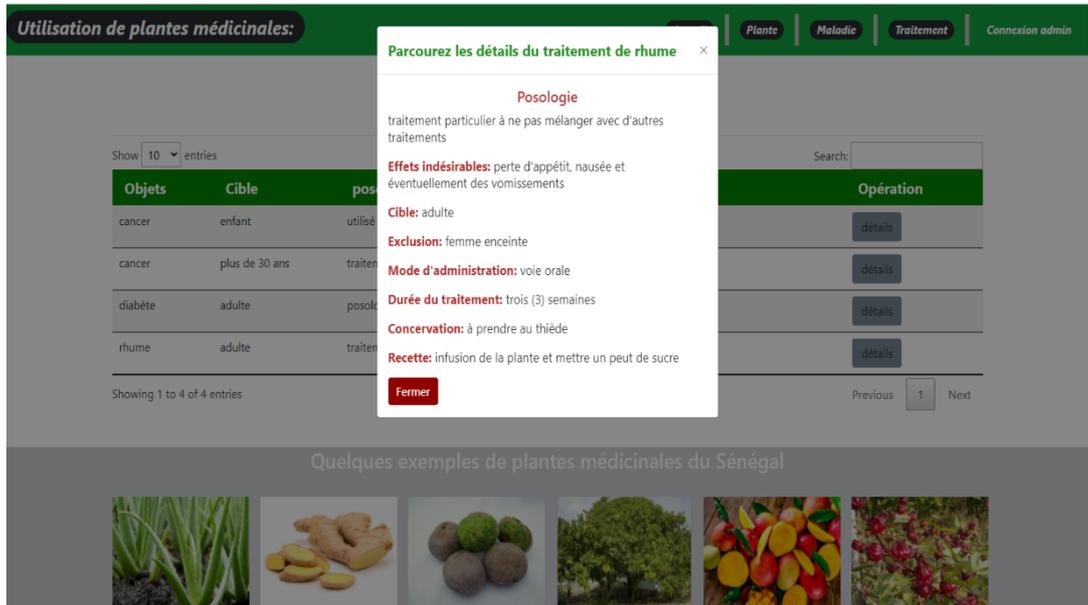


Figure 42: détails traitement

### VII.3.2. Accès réservé

#### VII.3.2.1 Interface d'authentification

L'authentification est la première étape que l'utilisateur doit franchir pour accéder à l'interface d'administration et procéder à la gestion des différentes fonctionnalités du système. A travers l'interface d'authentification, il doit renseigner son nom d'utilisateur et son mot de passe pour pouvoir se connecter.

En Figure 43, nous vous présentons l'interface d'authentification :



Figure 43: interface d'authentification

### VII.3.2.2 Interface d'administration générale

C'est l'interface de gestion globale de l'application. Dans cette partie nous avons le contrôle sur toutes les fonctionnalités du système.

A côté du tableau de menu, nous avons les actions récentes réalisées sur la page et la partie d'en haut à droite, concerne l'utilisateur qui s'est connecté.

La structure de cette interface est représentée à la Figure 44 :

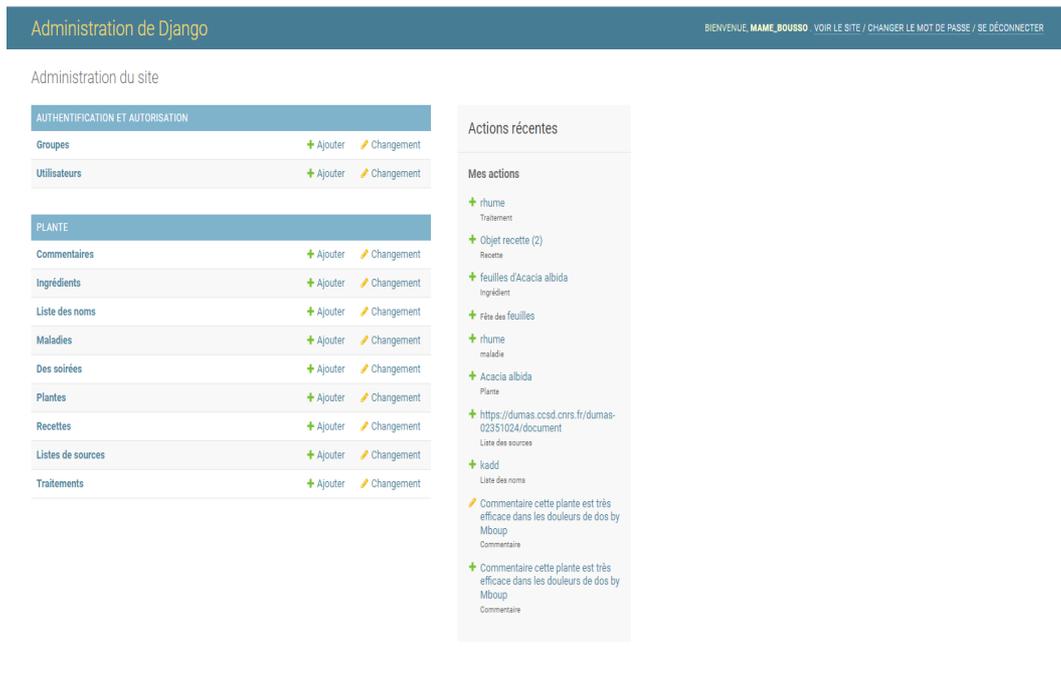


Figure 44: interface d'administration générale

### VII.3.2.3 Ajouter une plante

Dans cette partie, nous pouvons ajouter des plantes. Nous devons renseigner les informations demandées : nom scientifique, famille, type biologique (arbre, herbe ou liane), usages médicamenteux, usages non médicamenteux, description de la plante, écologie, source et nom (pour les autres langues) et après valider l'enregistrement.

La Figure 45 est une représentation de cette interface.

Administration de Django BIENVENUE, MAME BOUSSO · VOIR LE SITE / CHANGER LE MOT DE PASSE / SE DÉCONNECTER

Accueil · Plante · Plantes · Ajouter une plante

Commencez à taper pour filtrer...

**AUTHENTIFICATION ET AUTORISATION**

Groupes + Ajouter

Utilisateurs + Ajouter

**PLANTE**

Commentaires + Ajouter

Ingrédients + Ajouter

Liste des noms + Ajouter

Maladies + Ajouter

Des soirées + Ajouter

**Plantes** + Ajouter

Recettes + Ajouter

Listes de sources + Ajouter

Traitements + Ajouter

### Ajouter plante

Noms scientifique :

Famille :

Type biologique :

Usages médicamenteux :

Usages non médicamenteux :

Descriptif plante :

Écologie :

PlantelImage :  Aucun fichier choisi

La source:  ✎ +

Nom de la liste :  ✎ +

Figure 45: insertion d'une plante

## Conclusion

En définitive, le chapitre 4 nous a permis d'avoir une vision globale de l'application. Après avoir présenté les outils et technologies que nous avons utilisés, nous avons commencé par présenter le projet depuis sa création, ensuite, nous avons présenté quelques parties de l'implémentation et en fin nous avons montré quelques interfaces de l'application.

Ce chapitre, marquant le dernier de notre document de mémoire, nous amène à passer à la conclusion générale et à dégager nos perspectives sur ce projet.

## Conclusion générale et perspectives

Notre travail de mémoire de fin d'étude s'est bien terminé avec l'obtention d'une partie des objectifs que nous nous sommes fixés au début du travail (car l'application n'est pas encore, mis en ligne, etc.). Pour rappel, le travail consistait à la création d'une application permettant aux populations de bien s'orienter sur l'utilisation des plantes médicinales au Sénégal. Cette orientation va dans le sens de les permettre de connaître les plantes dont la véracité de leurs effets thérapeutiques a été prouvée, de trouver les traitements valables à leurs maux avec des indications précises.

Toutefois, la réalisation de ce mémoire nous a été d'un grand avantage car elle nous a permis d'avoir plus d'expériences dans le domaine des plantes médicinales qui fait intervenir beaucoup de secteur à savoir : la santé, l'agroforesterie, la chimie, etc. elle nous a également permis d'apprendre et d'utiliser une nouvelle technologie Django qui est très riche et simple d'utilisation, d'avoir de l'expérience dans le domaine de la recherche scientifique, etc.

Néanmoins, nous avons rencontré pas mal de soucis durant ce long travail puisque le sujet touche beaucoup de domaines qui nécessitent une certaine connaissance avant de pouvoir cerner la problématique et d'apporter des solutions. Parmi ces domaines, on peut citer la santé qui est un domaine très sensible et dans lequel l'utilisation des plantes médicinales est indispensable.

En perspectives, nous envisageons :

- d'ajouter les fonctionnalités qui n'ont pas été codées ;
- de déployer l'application en ligne ;
- de collaborer avec les experts du domaine pour pouvoir alimenter notre application ;
- de concevoir une application mobile USSD ;
- de rechercher de sources fiables pour trouver les bonnes informations ;
- de proposer des contacts de médecins phytothérapeutes qualifiés afin d'orienter les patients aux bons endroits c'est-à-dire les orienter vers des experts du domaine ;
- de faire une étude plus détaillée sur les plantes afin de déterminer les zones d'évolution dans les périodes les plus importantes de leur production ;
- de proposer des conseils sanitaires sur les plantes médicinales dans le but de maintenir la stabilité de leur état de santé ;
- d'intégrer plusieurs langues dans le système ;
- de proposer un FAQ entre les visiteurs et les médecins.

## Bibliographie

- [1] « Agence Nationale de Statistique et de la Démographie ». <https://www.ansd.sn/> (consulté le 11 novembre 2021).
- [2] F. Gueye, « Médecine traditionnelle du Sénégal : exemples de quelques plantes médicinales de la pharmacopée sénégalaise traditionnelle », p. 175, sept. 2019.
- [3] massotherapiedesmoulins, « Les plantes et la medecine traditionnelle », Massotherapiedesmoulins, 13 octobre 2021. <https://www.massotherapiedesmoulins.com/les-plantes-et-la-medecine-traditionnelle-188/> (consulté le 12 novembre 2021).
- [4] « Phytothérapie : la médecine par les plantes », Pharmacien Giphar. <https://www.pharmaciengiphar.com/medecines-naturelles/conseils-phytotherapie/phytotherapie-medecine-par-plantes> (consulté le 16 novembre 2021).
- [5] « Un panel d'experts approuve le protocole des essais cliniques de phytothérapie pour la COVID-19 », OMS | Bureau régional pour l'Afrique. <https://www.afro.who.int/fr/news/un-panel-dexperts-approuve-le-protocole-des-essais-cliniques-de-phytotherapie-pour-la-covid-19> (consulté le 17 novembre 2021).
- [6] I. Sancelot, « Cycle en V en gestion de projet : définition, méthode et intérêt ». <https://blog.hubspot.fr/marketing/cycle-en-v> (consulté le 13 décembre 2021).
- [7] « What is Unified Modeling Language (UML)? » <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> (consulté le 31 mars 2022).
- [8] « Téléchargez Visual Paradigm. Essai gratuit ». <https://www.visual-paradigm.com/download/> (consulté le 16 décembre 2021).
- [9] S. User, « Environnements de dev », Pulsar - Formations et créations sites, extranets et intranets. <https://www.pulsar-agency.com/creation-site-internet/comment-creer-un-site-internet/comment-gerer-son-projet-de-site-web/outils-de-gestion-de-projet-web/envIRONNEMENTS-DE-DEV> (consulté le 31 mars 2022).
- [10] « □ Langage de programmation - Définition et Explications », Techno-Science.net. <https://www.techno-science.net/glossaire-definition/Langage-de-programmation.html> (consulté le 17 décembre 2021).
- [11] « Langage de balisage - Définition de Langage de balisage (Markup language) ». <http://www.dicodunet.com/definitions/normes/langage-de-balisage.htm> (consulté le 24 mars 2022).

- [12] « HTML (HyperText Markup Language) | MDN ». <https://developer.mozilla.org/fr/docs/Web/HTML> (consulté le 28 mars 2022).
- [13] « Tutoriel SQL ». <https://www.w3schools.com/sql/> (consulté le 31 mars 2022).
- [14] « Bootstrap : définition, tutoriels, astuces, pratiques ». <https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/> (consulté le 16 décembre 2021).
- [15] « qu'est-ce qu'un SGBDR - Saytu ci Google ». [https://www.google.com/search?q=qu%27est+ce+qu%27un+SGBDR&ei=ubG8YZuxC5GNjLsPjJipqA0&ved=0ahUKEwjb8d\\_2l-v0AhWRBmMBHQxMCtUQ4dUDCA0&uact=5&oq=qu%27est+ce+qu%27un+SGBDR&gs\\_lcp=Cgdnd3Mtd2l6EAMyBggAEAoQEzIICAAQFhAeEBMyCAgAEBYQHhATOgcIABBHELADogQIABATsgUIPBIBMUoECEEYAEoECEYYAFC9DljBJGD\\_K2gBcAJ4AIABpAKIAcMHkgEFMC40LjGYAQCgAQHIAQjAAQE&client=gws-wiz](https://www.google.com/search?q=qu%27est+ce+qu%27un+SGBDR&ei=ubG8YZuxC5GNjLsPjJipqA0&ved=0ahUKEwjb8d_2l-v0AhWRBmMBHQxMCtUQ4dUDCA0&uact=5&oq=qu%27est+ce+qu%27un+SGBDR&gs_lcp=Cgdnd3Mtd2l6EAMyBggAEAoQEzIICAAQFhAeEBMyCAgAEBYQHhATOgcIABBHELADogQIABATsgUIPBIBMUoECEEYAEoECEYYAFC9DljBJGD_K2gBcAJ4AIABpAKIAcMHkgEFMC40LjGYAQCgAQHIAQjAAQE&client=gws-wiz) (consulté le 17 décembre 2021).
- [16] P. G. D. Group, « PostgreSQL », *PostgreSQL*, 28 mars 2022. <https://www.postgresql.org/> (consulté le 28 mars 2022).
- [17] « Qu'est-ce qu'un diagramme de déploiement UML ? », *Lucidchart*. <https://www.lucidchart.com/pages/fr/diagramme-de-deploiement-uml> (consulté le 21 décembre 2021).
- [18] « coursUML3.pdf ». Consulté le: 22 décembre 2021. [En ligne]. Disponible sur: <http://remy-manu.no-ip.biz/UML/Cours/coursUML3.pdf>

## Annexe

Table	Champs	Description	Type	Liaison
Plante	id_plante	identifiant de la plante qui est unique.	entier	Source, NomList, Partie, Commentaire
	nomscientifique	nom scientifique de la plante	chaîne de caractères	
	famille	famille de la plante	chaîne de caractères	
	planteImage	photo de la plante	image	
	type_biologique	arbre, herbe ou liane	chaîne de caractères	
	usages_medicamenteux	les usages médicamenteux de la plante	chaîne de caractères	
	usages_non_medicamenteux	les usages non médicamenteux de la plante	chaîne de caractères	
	description	un peu de description sur la plante pour permettre aux visiteurs d'avoir un aperçu sur celle-ci.	chaîne de caractères	
	ecologie	zones fréquentes d'évolution de la plante	chaîne de caractères	
Source	id_source	identifiant de la source qui est unique.	entier	Plante
	titre	dans cette section, on mettra le titre de la source.	chaîne de caractères	
	lien	le lien vers la source.	lien url	
Traitement	id_traitement	identifiant du traitement qui est unique.	entier	Commentaire, Maladie, Recette
	posologie	les indications	chaîne de	

		à prendre en compte pour le traitement c'est-à-dire comment on doit s'y prendre.	caractères	
	effet_indesirable	les effets indésirables susceptibles d'être causés par la prise du traitement.	chaîne de caractères	
	cible	le groupe de personne destinée à prendre le traitement.	chaîne de caractères	
	exclusion	le groupe de personne qui ne doit pas prendre le traitement.	chaîne de caractères	
	mode_administration	comment on doit prendre le traitement (voie orale, etc...).	chaîne de caractères	
	duree_traitement	la durée du traitement	chaîne de caractères	
	conservation	apporte des indications sur le mode de conservation du traitement.	chaîne de caractères	
Maladie	id_maladie	identifiant de la maladie qui est unique.	entier	Traitement
	intitule_maladie	« nom » de la maladie	chaîne de caractères	
	description	la description de la maladie	chaîne de caractères	
Nom	id_nom	identifiant du nom de l'entité (plante ou maladie) qui est unique.	entier	Plante
	Nom_wolof	nom de la plante en langue wolof	chaîne de caractères	

	Nom_mandingue	nom de la plante en langue mandingue	chaîne de caractères	
	Nom_diola	nom de la plante en langue diola	chaîne de caractères	
Ingredient	id_ingredient	identifiant de l'ingrédient qui est unique.	entier	Partie, Recette
	etat	indique dans quel état doit être l'ingrédient.	chaîne de caractères	
	dosage	indique à quel rythme on doit un utiliser l'ingrédient.	chaîne de caractères	
Partie	id_partie	identifiant de la partie qui est unique.	entier	Plante, Ingredient
	intitule	le nom de la partie de la plante qui est utilisée (feuilles, tige, etc...).	chaîne de caractères	
Recette	id_recette	identifiant de la recette qui est unique.	entier	Traitement, Ingredient
	titre_recette	titre de la recette	chaîne de caractères	
	description_recette	permet de décrire la recette c'est-à-dire ses composants.	chaîne de caractères	
Commentaire	id_commentaire	identifiant du commentaire qui est unique.	entier	Plante, Traitement
	contenu	le contenu du commentaire.	chaîne de caractères	
	auteur	l'auteur du commentaire.	chaîne de caractères	
	email	Email de l'auteur du commentaire	email	
	date_pub	la date à laquelle le	date	

		commentaire a été publié.		
Utilisateur	id_utilisateur	identifiant de l'utilisateur	entier	
	identifiant	nom d'utilisateur	chaîne de caractères	
	mot_de_passe	mot de passe	chaîne de caractères	
	profil	profil (gestionnaire pou administrateur	chaîne de caractères	