



UNIVERSITÉ D'ANTANANARIVO
SCIENCES ET TECHNOLOGIES
PHYSIQUE ET APPLICATIONS



MÉMOIRE DE FIN D'ÉTUDE

Pour l'obtention du diplôme

Master d'Ingénierie en Systèmes Électroniques Informatiques

Intitulé

Conception d'une application web pour la gestion de plantes du
CNARP

Présentée par

RAJAONARIVELO Maminirina

Devant la commission d'examen

Président : RANDRIAMANANTANY Zely Arivelo

Rapporteur : RAZANAMANAMPISOA Harimalala

Examineur : RAKOTOARIMANANA Liva Graffin

Professeur titulaire

Maitre de conférences

Maitre de conférences



Le 21 Juillet 2021



UNIVERSITÉ D'ANTANANARIVO
SCIENCES ET TECHNOLOGIES
PHYSIQUE ET APPLICATIONS



MÉMOIRE DE FIN D'ÉTUDE

Pour l'obtention du diplôme

Master d'Ingénierie en Systèmes Électroniques Informatiques

Intitulé

Conception d'une application web pour la gestion de plantes du
CNARP

Présentée par

RAJAONARIVELO Maminirina



Devant la commission d'examen

Président : RANDRIAMANANTANY Zely Arivelo

Rapporteur : RAZANAMANAMPISOA Harimalala

Examineur : RAKOTOARIMANANA Liva Graffin

Professeur titulaire

Maitre de conférences

Maitre de conférences



Le 21 Juillet 2021

REMERCIEMENTS

Ce mémoire n'a pas pu être réalisé que grâce à la contribution et aux aides précieuses de plusieurs personnes qui nous ont soutenus sur les plans moral, intellectuel, matériel et financier. À tous, nous leur adressons nos reconnaissances les plus sincères.

À Monsieur RAMAHAZOSOA Irrish Parker, Maitre de Conférences, Responsable du Domaine Sciences et Technologie, pour avoir validé notre inscription d'être parmi les étudiants de ce Domaine ;

À Madame RAFANJANIRIANA Eulalie Odette, pour avoir facilité le bon déroulement de nos études au sein de la Mention Physique et Applications Responsable dont elle est la Responsable ;

À Madame RAZANAMANAMPISOA Harimalala, Maitre de Conférences, Responsable du Parcours Master d'Ingénierie en Systèmes électroniques et Informatiques (MISEI), pour avoir accepté notre inscription dans ledit Parcours et pour m'avoir guidé tout au long de ce mémoire ;

À Madame RAKOTOSAONA Rianaso Mbolanoro, Directeur du CNARP qui m'a accueilli chaleureusement dans sa société ;

À Madame RANDRIAMANANTANY Zely Arivelo, Professeur Titulaire, qui a bien voulu présider cette soutenance ;

À Monsieur RAKOTOARIMANANA Liva Graffin, Maître de conférences qui a bien voulu faire partie des membres du jury de ce mémoire ;

À Monsieur RAKOTONANDRASANA Stephan Richard, Maitre de recherches, responsable du département Ethnobotanique du CNARP, pour m'avoir encadré durant mon stage ;

À tous les membres du personnel du département Ethnobotanique et botanique du CNARP.

Nous adressons nos sincères remerciements à tous les membres de notre famille et à nos proches qui nous ont soutenus matériellement et moralement pendant les périodes difficiles. Nous sommes très reconnaissantes envers leur compréhension.

Enfin, nos reconnaissances vont aux endroits de ceux qui ont, de loin et de près, participé à la réalisation de ce mémoire de fin d'étude.

TABLE DE MATIÈRES

REMERCIEMENTS	i
TABLE DE MATIÈRES	ii
LISTE DES ACRONYMES / ABRÉVIATIONS	iv
LISTE DES FIGURES	v
LISTE DES TABLEAUX	vi
INTRODUCTION	1
CHAPITRE I: GÉNÉRALITÉS	3
I-1 Présentation du CNARP	3
I-1-1 Historique	3
I-1-2 Localisation	3
I-1-3 Missions du CNARP	3
I-1-4 Organigramme	4
I-2 Etude de l'existant	4
I-2-1 Inconvénients de l'existant	4
I-2-2 Objectif de l'application	5
I-3 Technologies du web	6
I-3-1 Définition d'une application web	6
I-3-2 Différence entre site web et application web	6
I-3-3 Structure d'une page web	7
I-3-4 L'architecture d'application web	7
I-3-5 Présentation de l'architecture d'un système client / serveur	8
I-4 Caractéristiques des plantes médicinales du CNARP	9
CHAPITRE II: ÉTUDE FONCTIONNELLE ET TECHNIQUE	11
II-1 Spécification des besoins	11
II-1-1 Spécification des besoins fonctionnels	11
II-1-2 Spécification des besoins non-fonctionnels	11
II-2 Choix méthodologique	12
II-2-1 Le langage UML	12
II-2-2 L'approche orientée objet	14
II-2-3 L'architecture MVC	15
II-2-4 Framework Laravel	17

II-2-5	Bootstrap	19
CHAPITRE III:	ANALYSE ET CONCEPTION.....	20
III-1	Modélisation en langage UML	20
III-1-1	Identification des acteurs	20
III-1-2	Diagramme de cas d'utilisation	20
III-1-3	Diagramme de séquence	23
III-2	Mise en place de la base de données.....	25
III-2-1	Diagramme de classe	25
III-2-2	Description des classes du système	26
CHAPITRE IV:	IMPLÉMENTATION ET RÉSULTATS	29
IV-1	Plateforme de développement.....	29
IV-1-1	Laragon.....	29
IV-1-2	PhpMyAdmin	29
IV-1-3	Langage de développement	30
IV-1-4	Visual Studio	32
IV-1-5	Node JS.....	32
IV-2	Interface de l'application	33
IV-2-1	Authentification	33
IV-2-2	Menu principal.....	34
IV-2-3	Affichage des informations dans la base de données	35
IV-2-4	Création de nouvelles données	36
IV-2-5	Suppression de données.....	37
IV-2-6	Mettre à jour	37
IV-2-7	Recherche dans la base de données	38
IV-3	Interprétations des résultats.....	38
CONCLUSION	39
RÉFÉRENCES	40
ANNEXES	I

LISTE DES ACRONYMES / ABRÉVIATIONS

API	Application Programming Interface
CNARP	Centre National d'Application de Recherche Pharmaceutiques
CSS	Cascading Style Sheets
E / S	Entrée et Sortie
FTP	File Transfer Protocol
HTML	HyperText Markup Language
MVC	Modèle - Vue – Contrôleur
OMS	Organisation Mondiale de la Santé
PHP	Hypertext Preprocessor
SFTP	Secure File Transfer Protocol
SSH	Secure Shell
UML	Unified Modeling Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WYSIWYG	What - You - See - Is - What - You – Get

LISTE DES FIGURES

Figure 1: Localisation du CNARP (source : Cnarp.recherche.gov.mg).....	3
Figure 2: Organigramme du CNARP.....	4
Figure 3: Les données sur Excel	5
Figure 4: Interactions entre le modèle, la vue et le contrôleur (source : baptiste-wicht.developpez.com).....	16
Figure 5: Logo Laravel.....	19
Figure 6: Diagramme de cas d'utilisation principal	20
Figure 7: Diagramme de l'authentification	21
Figure 8: Diagramme de création de nouvelles données.....	21
Figure 9: Diagramme de suppression de données	22
Figure 10: Diagramme de mise à jour des données	22
Figure 11: Diagramme de séquence d'authentification.....	23
Figure 12: Diagramme de séquence de création de nouvelles données	24
Figure 13: Diagramme de séquence de modification de données	24
Figure 14: Diagramme de séquence de suppression de données	25
Figure 15: : Diagramme de classe	26
Figure 16: Tables de la base de données	27
Figure 17: Interface Laragon.....	29
Figure 18: Interface PhpMyAdmin	30
Figure 19: Logo de Visual Studio	32
Figure 20: Logo de Node JS.....	32
Figure 21: Connexion.....	33
Figure 22: Inscription	33
Figure 23: Accueil avant authentification	34
Figure 24: Accueil après authentification	35
Figure 25: Liste des informations.....	35
Figure 26: Actions de l'application.....	36
Figure 27: Création de nouvelles données	37
Figure 28: Suppression de données	37
Figure 29: Mise à jour des données.....	37
Figure 30: Recherche dans la base de données	38

LISTE DES TABLEAUX

Tableau 1: Classification d' <i>Artemisia annua</i>	10
---	----

INTRODUCTION

De nos jours, devant le succès de l'avancée technologique, les entreprises cherchent à accroître leur potentiel dans le domaine informatique, notamment dans la gestion de base de données. L'informatique est de plus en plus utilisée dans tous les domaines d'activités y compris celui de la gestion, notre étude se rapporte ainsi à la conception d'une application web pour la gestion de plantes du Centre National d'Application de Recherche Pharmaceutique. L'objectif étant de concevoir une application permettant aux utilisateurs de gérer facilement leurs plantes médicinales.

Nous avons constaté pendant nos recherches que l'ensemble des traitements de données au sein du CNARP se fait sur différents fichiers Excel, répartis dans tous les départements, ce qui engendre un certain nombre de problèmes tel que la lenteur dans l'accès des données et un risque de perte d'information. La meilleure solution pour remédier aux problèmes est la conception d'une application qui assure l'accès instantané aux données et une sécurisation de ces dernières, ce qui simplifie le travail des chercheurs de l'entreprise.

Pour la réalisation de notre projet, notre choix s'est porté sur le Processus Unifié. Le processus unifié est une solution de développement logiciel adaptée à tout type de projet. Ses traits distinctifs tiennent en trois notions : piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental.

Le langage de modélisation qu'on a utilisé est UML (Unified Modeling Language), qui est une partie intégrante de la démarche UP. Ses diagrammes sont utilisés dans chaque étape et phases de ce processus de développement. Pour l'implémentation, le choix du langage de programmation a été dicté par le type de l'application qui devait être réalisée.

Pour mieux traiter et rapporter ce thème, nous structurons notre manuscrit de mémoire en quatre (04) principaux chapitres.

Dans le premier chapitre intitulé « Généralités » nous présentons le CNARP ainsi que quelques concepts de base ont servi pour l'accomplissement de notre projet.

Notre deuxième chapitre porte sur « l'Étude fonctionnelle et technique » où nous parlerons des études des besoins pour la réalisation du projet ainsi que les choix méthodologiques.

Le troisième chapitre se consacre à « l'Analyse et la conception », nous présenterons la modélisation en langage UML et la mise en place de la base de données.

Dans le dernier chapitre intitulé « Implémentation et résultats », nous présenterons la plateforme de développement et l'interface de l'application.

CHAPITRE I: GÉNÉRALITÉS

I-1 Présentation du CNARP

Le CNARP est un des centres de recherches du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique du Gouvernement Malgache. Collaborateur de l'OMS en médecine traditionnelle, les produits du centre sont testés cliniquement, et sans ajout chimiques.

I-1-1 Historique

Le Centre National d'Application de Recherches Pharmaceutiques a été créé le 1er Octobre 1976 par le décret n° 76.334, portant le nom de Centre National de Recherches Pharmaceutiques. Ce n'est qu'en 22 avril 1992 qu'il fut rebaptisé Centre National d'Application de Recherches Pharmaceutiques par le décret n°92.468.

Le 06 Octobre 2015, le statut régissant le Centre actuellement a été modifié et validé selon le Décret n°2015-1199.

I-1-2 Localisation

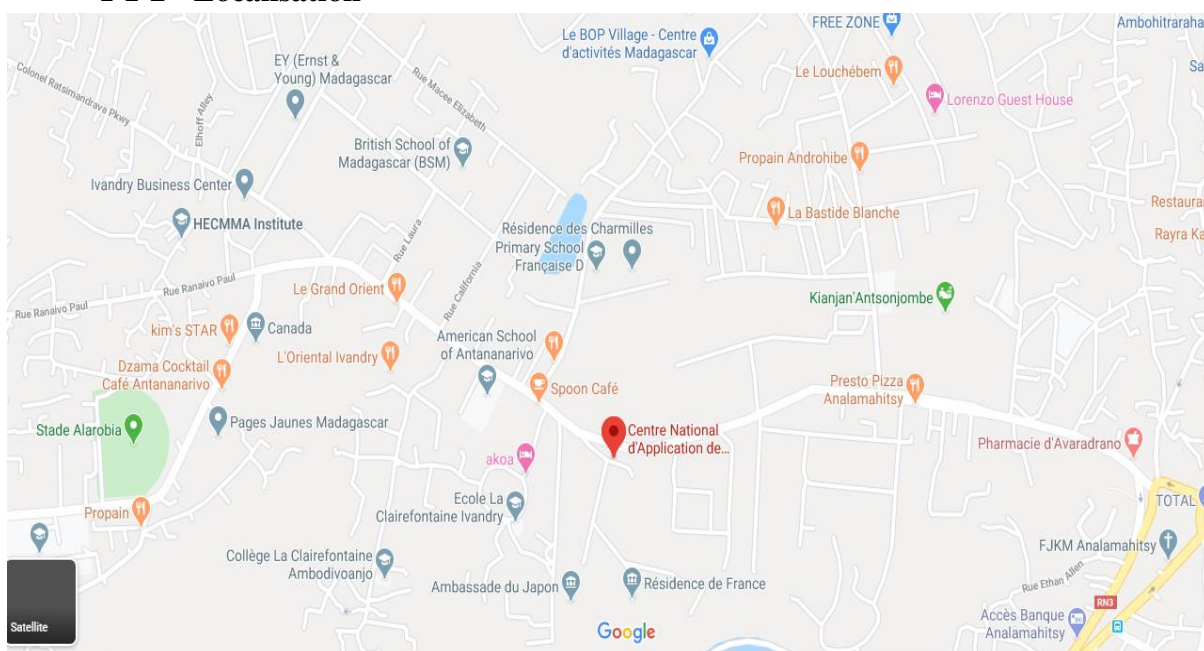


Figure 1: Localisation du CNARP (source : Cnarp.recherche.gov.mg)

I-1-3 Missions du CNARP

Leurs missions sont :

- De participer à l'élaboration et à la mise œuvre de la politique nationale en matière de recherches scientifiques et technique ;
- D'assurer dans le cadre de cette politique nationale, la définition, l'orientation, la promotion, l'innovation, la coordination et le contrôle de toutes les activités de recherches

concernant les plantes médicinales (études ethnobotaniques et botaniques, chimiques, etc. ...), les produits d'origine animale et minérale ayant des propriétés thérapeutiques. [1]

I-1-4 Organigramme

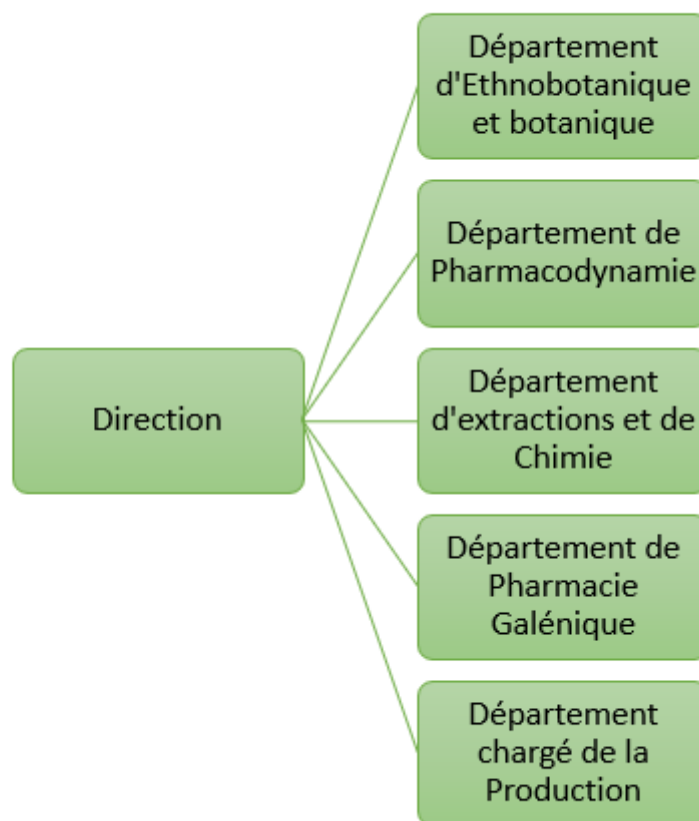


Figure 2: Organigramme du CNARP

I-2 Etude de l'existant

La base de données de plantes médicinales du CNARP est stockée dans plusieurs fichiers Excel lesquels contiennent les données sur les utilisations ethnobotaniques, les constituants chimiques (chimie) et les activités biologiques (pharmacologie).

I-2-1 Inconvénients de l'existant

La base de données stockées dans les fichiers Excel présente plusieurs inconvénients :

- Les données sont réparties dans plusieurs fichiers, ce qui ne rend facile la recherche d'une plante spécifique. Perte de temps de la recherche ;
- Il est difficile d'avoir une base de données mise à jour puisque les recherches sont faites régulièrement ;

- Personne ne détient l'intégralité des données. Possibilité de perdre des données en cours de route ;
- Pas de privilèges d'accès.

The figure shows two side-by-side Excel spreadsheets. The left spreadsheet is titled 'Xylopia beananensis' and the right is 'Strobilanthes madagascariensis'. Both spreadsheets have a similar structure with columns for Family, Genus, Species, Author, and Clade. The left spreadsheet has a tab labeled 'PLANTE' and the right has a tab labeled 'BDD'.

FAMILLE	GENRE ESPECE	GENRE	ESPECE	AUTEUR	CLADE 1
1	Acanthaceae	Avicennia marina	marina	Vierh.	Angiosper
2	Acanthaceae	Hypoestes serpens	serpens	Nees	Angiosper
3	Acanthaceae	Asystasia coromandelii	Asystasia coromandelii	Nees	Angiosper
4	Acanthaceae	Hypoestes trichochlamys	trichochlamys	Baker	Angiosper
5	Acanthaceae	Hypoestes maculosa	maculosa	Nees	Angiosper
6	Acanthaceae	Hypoestes glandulifera	glandulifera	Scott Elliot	Angiosper
7	Acanthaceae	Crossandra quadridentata	quadridentata	R. Benoist	Angiosper
8	Acanthaceae	Asystasia gangetica	gangetica	T. Anders	Angiosper
9	Acanthaceae	Thunbergia platyphylla	platyphylla	Baker	Angiosper
10	Acanthaceae	Strobilanthes madagascariensis	madagascariensis	Baker	Angiosper
11	Acanthaceae	Justicia sp	sp		Angiosper
12	Acanthaceae	Ruellia dequairei	dequairei	R. Benoist	Angiosper
13	Acanthaceae	Anisotes madagascariensis	madagascariensis	Benoist	Angiosper
14	Acanthaceae	Hypoestes comosa	comosa	R. Benoist	Angiosper
15	Acanthaceae	Justicia haplostachya	haplostachya	Anders	Angiosper
16	Acanthaceae	Justicia gendarussa	gendarussa	Burman	Angiosper
17	Acanthaceae	Dicliptera madagascariensis	madagascariensis	Nees	Angiosper
18	Acanthaceae	Dicliptera ocyroides	ocyroides	Jussieu	Angiosper
19	Acanthaceae	Thunbergia cyanea	cyanea	Bojer	Angiosper
20	Acanthaceae	Hypoestes decaryana	decaryana	R. Benoist	Angiosper
21	Acanthaceae	Hypoestes cernua	cernua	Nees	Angiosper
22	Acanthaceae	Thunbergia alata	alata	Hilsenber & Bojer	Angiosper
23	Acanthaceae				

Figure 3: Les données sur Excel

Les employés recensent à ce jour plus de sept milles (7000) plantes, et bien que toutes ces plantes soient sur Excel, il aurait fallu trier et fusionner pour arriver à quelque chose de plus facile à utiliser et qui ne fasse pas perdre du temps.

I-2-2 Objectif de l'application

Notre projet consiste ainsi à créer une application web capable de prendre en compte la gestion de toutes ces plantes sans distinction de département. L'application doit permettre de :

- Consulter la base de données ;
- Adjoindre ou supprimer une nouvelle information dans la base de données ;
- Rechercher dans la base de données par n'importe quel mot-clé ;
- Mettre à jour les données existantes.

La nouvelle base de données contiendra ainsi les noms des plantes médicinales ainsi que les informations sur l'ethnobotanique, chimique et pharmacologique. L'application est aussi protégée par un mot de passe pour que seuls les utilisateurs puissent y accéder. Il est possible de créer des identifiants à la guise de l'entreprise. Sur le domaine de la visibilité, l'interface devra être agréable, fluide et facile à utiliser.

I-3 Technologies du web

I-3-1 Définition d'une application web

Pour définir une application web, il faut d'abord pouvoir définir ce qu'est une application et ce qu'est le web. Avant l'arrivée d'Internet, il n'y avait pas d'applications web. Il n'y avait que des applications natives, appelées logiciel ou simplement application, pour Windows, MacOS et Linux. Bien qu'Internet ait totalement transformé le marché des applications, au début il n'y avait pas d'applications web. Des applications natives ont commencé à utiliser Internet pour échanger des informations entre plusieurs clients. Le courrier électronique en est un exemple et continu d'être au centre de nos communications dans la vie de tous les jours. [2]

Le web fût inventé au CERN (Organisation européenne pour la recherche nucléaire) à Genève par Tim Berners-Lee plusieurs années après Internet. À ses débuts, il n'était pas possible de développer des applications web. Il était seulement possible de créer de simples pages web pour afficher du texte. Le web était à l'origine fait pour distribuer de l'information, en créant des sites web, et non pour collaborer par le biais d'applications web. Il fallut attendre la création de nouvelles technologies web pour que les premières applications web voient le jour.

Contrairement aux applications natives, les applications web ne sont pas installées sur des ordinateurs mais se trouvent sur des serveurs web. Le seul prérequis pour les utiliser est d'avoir un navigateur web, logiciel que l'on trouve installé par défaut avec tous les ordinateurs, tablettes et mobiles.

I-3-2 Différence entre site web et application web

➤ Site web

Un site web est ce que l'on retrouve lorsqu'on entre une adresse et qu'on atterrit sur une page. Le site correspond à cette page et toute page reliée qui est gérée par la même entité (souvent sous le même nom de domaine). Le rôle principal d'un site web est de fournir et présenter de l'information aux visiteurs. Un blog, un site de nouvelles ou un site d'information sur un produit ou une compagnie sont de bons exemples de sites web.

➤ Application web

Une application web est tout site web qui permet à ses utilisateurs d'accomplir des tâches spécifiques. Une application gère donc généralement des utilisateurs et toutes sortes de données selon les requis spécifiques au projet.

I-3-3 Structure d'une page web

Une page Web peut être considérée comme une arborescence de documents pouvant contenir n'importe quel nombre de branches. Il existe des règles sur les éléments que chaque branche peut contenir. Pour comprendre le concept d'une arborescence de documents, il est utile de considérer une page Web simple avec des fonctionnalités de contenu typiques à côté de son arborescence.

Le premier élément à apparaître dans le code source d'une page Web est la déclaration doctype. Cela fournit au navigateur Web des informations sur le type de langage de balisage dans lequel la page est écrite, ce qui peut ou non affecter la façon dont le navigateur rend le contenu. La plupart des éditeurs Web WYSIWYG créeront automatiquement le doctype après avoir sélectionné dans une boîte de dialogue le type de document que vous créez.

Pour créer un vrai document HTML, il faut trois éléments dans le conteneur :

`<Html>`, `<Head>`, `<Body>`

Ces trois éléments se combinent pour décrire la structure de base de la page :

`<Html>` : cet élément encapsule tout le contenu de la page

`<Head>` : Cet élément désigne la partie d'en-tête du document. Vous pouvez inclure des informations facultatives sur la page Web, telles que le titre (le navigateur l'affiche dans la barre de titre), des mots-clés de recherche facultatifs et une feuille de style facultative

`<Body>` : Cet élément contient le contenu de votre page Web, c'est-à-dire ce que nous voulons voir apparaître dans la zone de navigation du navigateur.

I-3-4 L'architecture d'application web

L'architecture d'application Web est un cadre reliant différents éléments pour permettre une expérience Web. C'est l'épine dorsale de la navigation quotidienne sur Internet : saisir une URL et afficher et interagir avec le site Web pendant que le navigateur communique avec le serveur est l'un des moyens de décrire l'architecture des applications Web.

Les architectures d'applications Web se composent de composants d'application, de systèmes middleware et de bases de données. Ils peuvent être divisés en deux groupes :

Les composants UI / UX comprend des tableaux de bord, des données statistiques, des éléments de notification, des mises en page, le suivi des activités et d'autres éléments. Ces composants créent les visuels d'une page Web et jettent les bases de l'expérience utilisateur.

Les composants structurels incluent le serveur d'applications Web et le serveur de base de données. La connaissance de HTML, JavaScript et CSS, ainsi que de Python, PHP, Java, Ruby, .NET et Node.js est nécessaire pour les créer.

En ce qui concerne la construction des composants, nous avons le choix entre plusieurs modèles :

- 1 serveur Web et 1 base de données
- 2 serveurs Web et 2 bases de données

Un serveur Web avec une base de données est le modèle le plus simple. Avec cette architecture de serveur Web, le bon fonctionnement d'une application dépend de la stabilité du serveur. En d'autres termes, s'il y a un problème avec le serveur, l'application ne fonctionnera pas. Pourtant, le modèle est suffisant pour les tests et les sessions privées.

L'utilisation d'une base de données pour deux serveurs Web est un modèle plus fiable, car il existe un serveur de sauvegarde. D'un autre côté, il est important de s'assurer que la base de données est sécurisée et toujours en cours d'exécution.

Avoir plus de deux bases de données et serveurs Web est l'option la plus fiable. En raison de sa capacité à gérer et traiter de grandes quantités de données, ce modèle est une base solide pour une architecture d'application Web d'entreprise.

I-3-5 Présentation de l'architecture d'un système client / serveur

➤ Notion de client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur. Cela signifie que des machines clients (des machines faisant partie du réseau) se contactent à un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc. les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clients. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichier, tandis que pour le client de messagerie il s'agit de courrier électronique).

➤ **Fonctionnement du client/serveur**

Dans un réseau informatique, un serveur est à la fois un ensemble de logiciels et d'ordinateurs les hébergeant dont le rôle est de répondre de manière automatique à des demandes envoyées par des client par exemple ordinateur et logiciel et cela via le réseau. Les serveurs sont d'usage courant dans les centres de traitement de donnée, les entreprises, les institutions, et les réseaux internet, où ils sont souvent un point central et sont utilisés simultanément par de nombreux utilisateurs pour stocker, partager et échanger des informations. Les différents usagers opérant à partir d'un client.

- Il est initialement passif (ou esclave en attente d'une requête).
- Il est à l'écoute prête à répondre aux requêtes envoyées par des clients.
- Dès qu'une requête lui parvient, il la traite et envoie une réponse.

Dans un réseau informatique, un client est l'ordinateur et le logiciel qui envoient des demandes à un serveur. Il peut s'agir d'un logiciel manipulé par une personne. On appelle client aussi bien l'ordinateur depuis lequel les demandes sont envoyées mais aussi le logiciel qui contient les instructions relatives à la formulation des demandes et la personne qui opère les demandes. L'ordinateur client est généralement un ordinateur personnel ordinaire, équipé de logiciels relatifs aux différents types de demandes qui vont être envoyées, comme par exemple un navigateur web, un logiciel client pour le World Wide Web.

- Il est actif (ou maître)
- Il envoie des requêtes au serveur.
- Il attend et reçoit des réponses du serveur.

I-4 Caractéristiques des plantes médicinales du CNARP

Une plante médicinale est une plante utilisée pour ses propriétés thérapeutiques. Cela signifie qu'au moins une de ses parties (feuille, tige, racine etc.) peut être employée dans le but de se soigner. Elles sont utilisées depuis au moins 7.000 ans avant notre ère par les Hommes et sont à la base de la phytothérapie. [3]

Chaque plante est classifiée par son nom scientifique, famille, noms vernaculaire, groupe, espèce, règne, sous-règne, division, sous-embranchement, classe et ordre.

Par exemple pour l'*Artemisia annua*, plante célèbre à Madagascar depuis le lancement du « Covid-Organics » ou « Tambavy CVO ».

Tableau 1: Classification d'*Artemisia annua*

Famille	Asteraceae
Nom vernaculaire	Artemisia
Groupe	Dicotyledones
Genre	Artemisia
Especce	Annua
Regne	Plantae
Sous-regne	Tracheobionta
Division	Magnoliophyta
Classe	Magnoliopsida

Après la récolte de chaque plante et après recherche, les chercheurs par département prennent note des résultats. Il y a notamment les données pharmacologique, chimique, et ethnobotanique.

Nous avons présenté succinctement la structure du CNARP, ensuite le contexte général de l'application ainsi que ses objectifs, ce qui nous a permis de recenser toutes les informations nécessaires et indispensables dans sa réalisation. Passons maintenant à l'analyse et aux matériels utilisés.

CHAPITRE II: ÉTUDE FONCTIONNELLE ET TECHNIQUE

II-1 Spécification des besoins

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les différents besoins. Elle doit décrire sans ambiguïté l'application à développer.

II-1-1 Spécification des besoins fonctionnels

Les besoins fonctionnels ou besoin métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Ils expriment une action que doit effectuer le système en réponse à une demande. [4]

L'application que nous allons développer devra regrouper les fonctionnalités qui permettront à l'utilisateur de :

- Adjoindre ou supprimer des informations ;
- Rechercher une plante dans la base de données ;
- Mettre à jour les données ;
- Afficher la liste des plantes.

II-1-2 Spécification des besoins non-fonctionnels

Les besoins non fonctionnels sont des exigences qui ne concernent pas spécifiquement le comportement du système mais identifient plutôt les contraintes internes et externes du système. Les principaux besoins non-fonctionnels de notre application sont :

➤ Besoin de sécurité

L'application devra comporter un login et mot de passe que seuls les membres de chaque département possèdent. Les besoins de sécurité sont satisfaits par Laravel qui utilise un algorithme de cryptage évolué.

➤ Besoin de disponibilité

L'application devra être opérationnelle de telle sorte que 'elle soit disponible 24h/24 et 7j/ 7. Les utilisateurs pourront ainsi compter sur sa fiabilité. Ce besoin de disponibilité est satisfait grâce au serveur.

➤ **Besoins de performance**

Ce sont les performances d'exécution de l'application, généralement en matière de temps de réponse, temps de chargement d'une page. L'application devra ainsi répondre aux exigences des utilisateurs de façon optimale, c'est-à-dire effectuer les opérations dans un laps de temps plus ou moins court. Ce besoin est garanti par le Framework.

➤ **Une solution ouverte et évoluée**

L'application pourra être améliorée par l'ajout d'autres modules pour garantir sa souplesse et l'évolutivité. Notre application étant déjà un ensemble de plusieurs modules, on peut affirmer qu'elle répond au critère d'une solution ouverte et évoluée.

II-2 Choix méthodologique

Pour mener à bien notre projet, nos choix se sont portés sur les méthodologies et technologies suivantes :

- Le langage UML ;
- La technologie orienté objet ;
- Le modèle MVC ;
- Framework Laravel.

II-2-1 Le langage UML

Langage de modélisation objet dont le développement a commencé en 1994, l'Unified Modeling Language est une notation standardisée qui facilite la conception des logiciels, en amont du travail de programmation proprement dit. UML modélise les objets et leurs liens au moyen de vues constituées de diagrammes. [5] [6]

Avec une seule représentation subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce à ces différents diagrammes. En effet, il couvre l'aspect statique et dynamique d'un système selon ses différents diagrammes. [7]

➤ **Les vues statiques**

Elles permettent de représenter le système physiquement. On trouve alors les diagrammes suivants :

- Le Diagramme de classes : il représente la structure statique en termes de classes et de relations entre elles, il représente aussi un ensemble d'interface et de paquetages ainsi que leurs relations ;

- Le Diagramme d'objets ou le diagramme d'instances : représente une instance possible du diagramme de classes ;

- Le Diagramme de composants : représente les morceaux d'applications packagés sous la forme de composants disposants d'interfaces. Il permet de décrire ces composants qui sont : le sous-système, le module, le programme et le sous-programme, le processus et la tâche ;

- Le Diagramme de déploiement : complémentaire du diagramme de composants, il décrit la répartition physique des instances de composants, de processus et d'objets d'une application distribuée.

➤ **Les vues dynamiques**

Les cinq diagrammes comportementaux (ou dynamiques) représentent des vues dynamiques du système :

- Le Diagramme de cas d'utilisation : sont des vues qui décrivent les interactions entre les différents acteurs externes (utilisateurs du cas) et les fonctionnalités du système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur. Leur but est d'identifier les acteurs du domaine, leurs responsabilités respectives et de décrire leurs besoins ;

- Le Diagramme de collaboration : il décrit l'interaction modélisée par les échanges de messages entre objets ou entre acteurs et objets ;

- Le Diagramme de séquence : il diffère légèrement du diagramme de collaboration par l'ajout d'une dimension temporelle en précisant la chronologie des échanges de messages entre les objets ;

- Le Diagramme d'états transitions : il décrit l'ensemble des états des objets du système et les transitions qui déclenchent le passage d'un état donné vers un autre état ;

- Le Diagramme d'activités : est une variante des diagrammes d'états-transitions. Il décrit l'ensemble des activités effectuées par les acteurs du système en les décomposant en sous-activités et en spécifiant les contraintes relatives à l'enchaînement de ces dernières.

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à chaque modélisation. Pour le cas de notre application, les diagrammes utilisés sont les diagrammes de cas d'utilisation, de séquence et de classes.

➤ **Avantages de l'UML**

UML est un langage formel et normalisé qui facilite la compréhension de représentations abstraites complexes et le principal avantage d'UML est qu'il est devenu le standard en termes de modélisation objet, son caractère polyvalent et performant et sa souplesse en fait un langage universel.

Ses avantages sont multiples :

- UML est un langage formel et normalisé ;
- Gain de précision ;
- Gage de stabilité ;
- Encourage l'utilisation d'outils ;
- UML est un support de communication performant ;
- Il cadre l'analyse
- Il facilite la compréhension de représentations abstraites complexes.

Le principal avantage d'UML est qu'il est devenu le standard en termes de modélisation objet, son caractère polyvalent et performant et sa souplesse en fait un langage universel.

II-2-2 L'approche orientée objet

L'approche considère le logiciel comme une collection d'objets dissociés, identifiés et possédant des caractéristiques. Une caractéristique est soit un attribut (une donnée caractérisant l'état de l'objet), soit une entité comportementale de l'objet (une fonction). La fonctionnalité du logiciel émerge alors de l'interaction entre les différents objets qui le constituent. L'une des particularités de cette approche est qu'elle rapproche les données et leurs traitements associés au sein d'un unique objet.

Un objet est caractérisé par plusieurs notions :

➤ **L'identité**

L'objet possède une identité, qui permet de le distinguer des autres objets, indépendamment de son état. On construit généralement cette identité grâce à un identifiant découlant naturellement

du problème (par exemple un produit pourra être repéré par un code, une voiture par un numéro de série, etc.).

➤ **Les attributs**

Il s'agit des données caractérisant l'objet. Ce sont des variables stockant des informations sur l'état de l'objet.

➤ **Les méthodes**

Les méthodes d'un objet caractérisent son comportement, c'est-à-dire l'ensemble des actions (appelées opérations) que l'objet est à même de réaliser. Ces opérations permettent de faire réagir l'objet aux sollicitations extérieures (ou d'agir sur les autres objets). De plus, les opérations sont étroitement liées aux attributs, car leurs actions peuvent dépendre des valeurs des attributs, ou bien les modifier.

La difficulté de cette modélisation consiste à créer une représentation abstraite, sous forme d'objets, d'entités ayant une existence matérielle (chien, voiture, ampoule, personne...) ou bien virtuelle (client, temps...).

La Conception Orientée Objet (COO) est la méthode qui conduit à des architectures logicielles fondées sur les objets du système, plutôt que sur la fonction qu'il est censé réaliser.

En résumé, l'architecture du système est dictée par la structure du problème.

II-2-3 L'architecture MVC

L'architecture Modèle/View/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Dans l'architecture MVC, les rôles des trois entités sont les suivants :

➤ **Rôle du modèle**

Le modèle contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient.

Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ses données. Dans le cas de données importantes, le modèle peut autoriser plusieurs vues partielles des données.

➤ Rôle de la vue

La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, ...). Ses différents événements sont envoyés au contrôleur.

La vue peut aussi donner plusieurs vues, partielles ou non, des mêmes données.

➤ Rôle du contrôleur

Le contrôleur est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour. Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier.

➤ Interactions

Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure suivante :

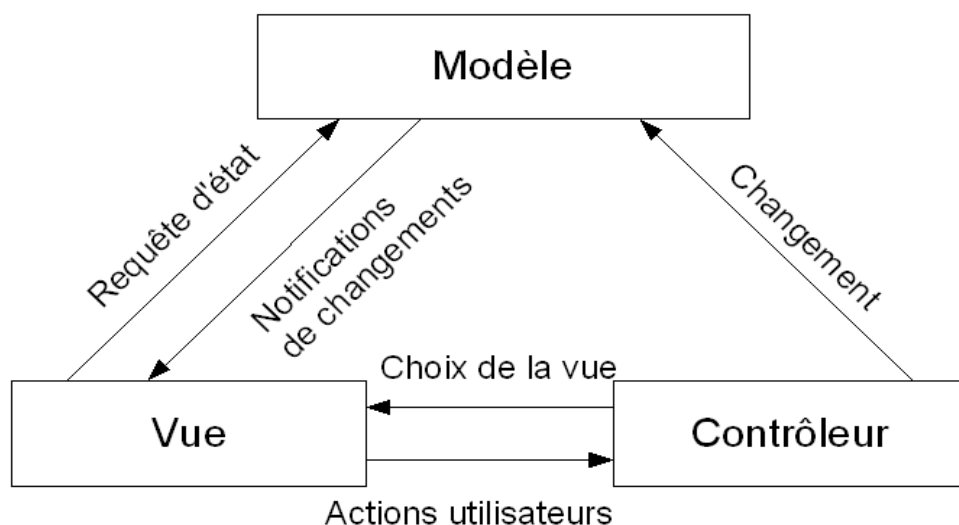


Figure 4: Interactions entre le modèle, la vue et le contrôleur (source : baptistewicht.developpez.com)

II-2-4 Framework Laravel

Un Framework n'est pas indispensable pour la création de notre application web. Cependant, pour que l'application soit robuste, facile à faire évoluer et réalisable en un temps minimum, un Framework représente un outil idéal. [8] [9]

Laravel est un Framework web open-source écrit en PHP. Il respecte le modèle MVC (modèle-vue-contrôleur) . C'est un Framework orienté objet distribué sous licence MIT. Laravel est un outil qui, dans sa conception, se base sur le meilleur de plusieurs autres Frameworks pour développer son propre système et être plus efficace. Il possède également des composants qui lui sont propres.

➤ Traitement des routes

Laravel offre une gestion des routes simple et intuitive, en utilisant des noms simples pour identifier les routes plutôt que de longs noms de chemin. L'utilisation d'identificateurs de route facilite également la maintenance des applications, car le nom de la route peut être modifié à un seul endroit plutôt que de devoir le changer partout. Toutes les routes de l'interface web d'une application Laravel sont enregistrées dans le fichier routes/web.php.

➤ Sécurité

Laravel comprend un certain nombre de fonctions de sécurité, notamment l'authentification des utilisateurs, l'autorisation des rôles des utilisateurs, la vérification des e-mails, les services de cryptage, le hachage des mots de passe et les fonctions de réinitialisation des mots de passe.

➤ Migration

Laravel fournit un contrôle de version pour les bases de données d'applications en utilisant les migrations. Les migrations permettent de suivre la façon dont une base de données a été modifiée au fil du temps, ce qui facilite la destruction ou la recréation de la base de données si nécessaire.

➤ Templating

Blade est le moteur de templating PHP de Laravel. Les moteurs de modèles PHP permet de séparer la logique métier des modèles HTML, ce qui permet d'obtenir une base de code plus facile à maintenir. De nombreuses fonctionnalités de Laravel reposent sur les modèles Blade.

Blade offre plus de fonctionnalités que les autres moteurs de création de modèles car il permet d'utiliser du code PHP simple, ce que les autres ne font pas.

➤ **Sessions**

Laravel utilise des sessions pour stocker des informations sur l'utilisateur à travers plusieurs requêtes. Les cookies sont un exemple de pilote de session intégré à Laravel.

➤ **Validation des données**

Laravel simplifie la validation des données utilisateurs entrants. Laravel comprend un certain nombre de règles de validation des données, avec des messages d'erreur personnalisables.

➤ **Traitement du cache**

Laravel prend en charge la mise en cache des données afin de minimiser les temps de traitement des tâches applicatives. L'API de mise en cache de Laravel prend en charge une variété de backends de cache tiers tels que Memcached et Redis.

➤ **Traitement des erreurs**

La gestion des erreurs est automatiquement configurée au démarrage d'un nouveau projet Laravel. Les applications Laravel peuvent être exécutées en mode débogage, générant des messages d'erreur détaillés pour toutes les erreurs qui se produisent.

➤ **Tests**

Laravel offre d'emblée d'importantes fonctionnalités de test. Laravel prend en charge les tests unitaires, qui testent de petites sections isolées du code de l'application, ainsi que les tests de fonctionnalités, qui testent des sections plus importantes du code et des fonctionnalités de plus haut niveau.

➤ **Stockage et gestion des fichiers**

Laravel utilise le paquetage PHP Flysystem pour fournir des pilotes permettant de travailler avec une variété de systèmes de fichiers, depuis les systèmes de fichiers locaux jusqu'au stockage sur le cloud comme Amazon S3. Laravel permet également le transfert de fichiers avec le protocole de transfert de fichiers SSH (SFTP).

➤ **E-mail**

Laravel comprend une API d'e-mail basée sur la bibliothèque SwiftMailer, qui permet d'envoyer des e-mails via le service de son choix. Laravel prend en charge les pièces jointes et la mise en file d'attente des e-mails.

➤ **Notifications**

Laravel prend en charge l'envoi de notifications via un certain nombre de canaux, qu'il s'agisse de canaux connus tels que SMS ou Slack, ou de canaux développés par la communauté Laravel.



Figure 5: Logo Laravel

II-2-5 Bootstrap

C'est un Framework front-end gratuit pour le développement web. Il contient plusieurs outils utiles à la réalisation de sites interactifs. Il permet de concevoir plus facilement un design responsif qui va permettre d'adapter l'affichage de l'application à tout type d'écran. Il fait partie des Frameworks les plus populaires. Il a été conçu en 2010 par deux développeurs de chez Twitter : Mark Otto et Jacob Thornton. Son but était de diminuer les coûts de maintenances dus aux incohérences entre les différentes bibliothèques existantes. [10] [11]

Il est conçu pour être compatible avec tous les navigateurs majeurs tels que Google Chrome, Firefox, Safari ou encore Opera. Il fonctionne également en mode dégradé sur des navigateurs plus anciens.

Le concept de Bootstrap est basé sur les grilles. Chaque élément de l'interface se situe à l'intersection d'une ligne et d'une colonne. Cette grille sert d'armature à la totalité de l'interface et est également très pratique en termes de design responsive.

Bootstrap est le frontend utilisé avec Laravel. Il est publié sous licence MIT.

L'étude technique et fonctionnelle nous a permis de passer en revue les besoins de notre application ainsi que les méthodes choisies pour l'élaboration de notre application. Passons maintenant à l'analyse et conception de notre projet.

CHAPITRE III: ANALYSE ET CONCEPTION

III-1 Modélisation en langage UML

L'UML est un langage de modélisation objet. Il permet de modéliser les objets et ainsi représenter l'application sous forme de diagramme.

III-1-1 Identification des acteurs

Les acteurs d'un système sont les entités externes à ce système qui interagissent (saisie de données, réception d'information...) avec lui. Les acteurs sont donc à l'extérieur du système et dialoguent avec lui. Dans notre application, nous avons comme acteur les chercheurs du CNARP. Ils gèrent ainsi le fruit de leur recherche dans l'application.

III-1-2 Diagramme de cas d'utilisation

La connaissance des fonctionnalités à implémenter est essentielle pour établir le diagramme de cas d'utilisations de l'application. L'acteur est donc ici un utilisateur qui communique et interagit avec les cas d'utilisation du système. [12]

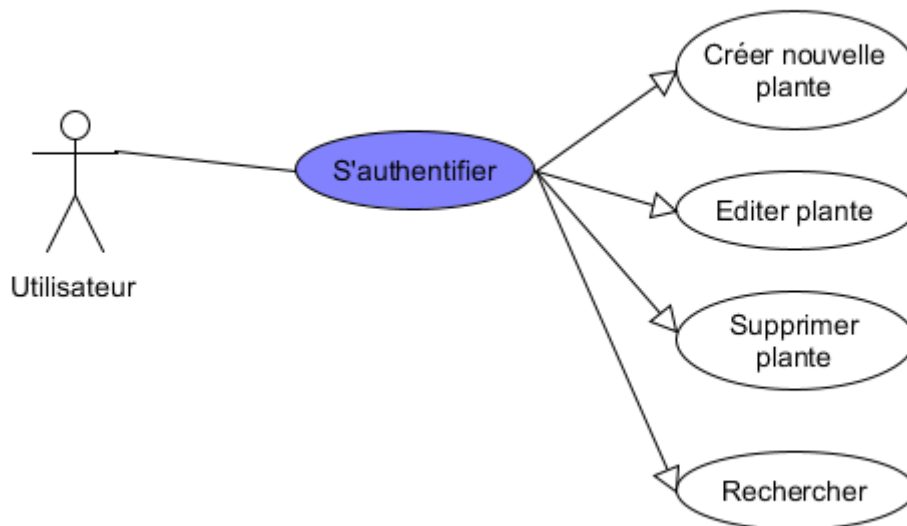


Figure 6: Diagramme de cas d'utilisation principal

➤ **Diagramme de l'authentification**

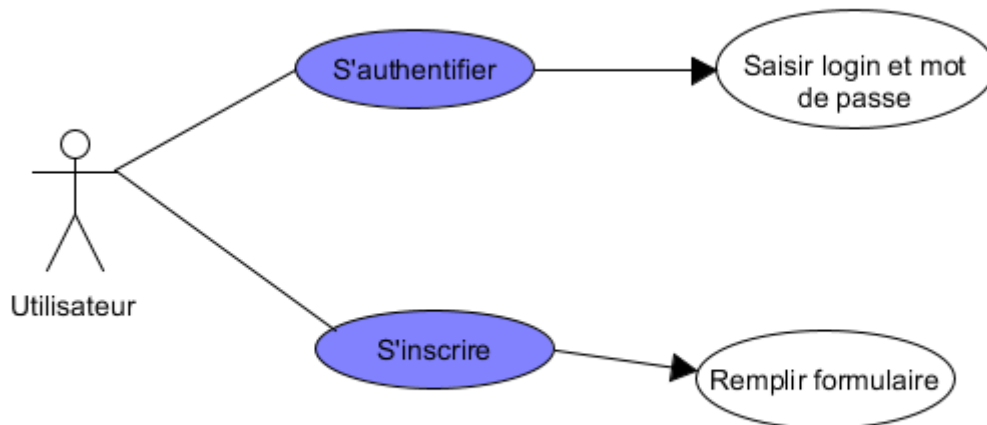


Figure 7: Diagramme de l'authentification

➤ **Diagramme de création de nouvelles données**

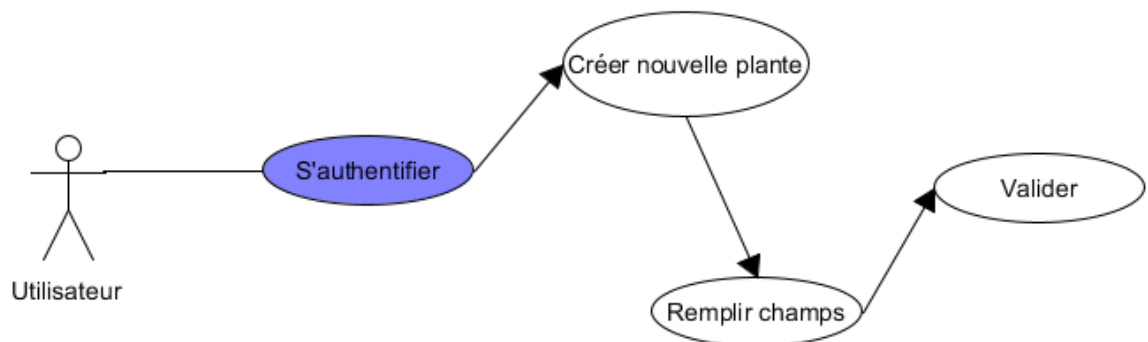


Figure 8: Diagramme de création de nouvelles données

➤ **Diagramme de suppression de données**

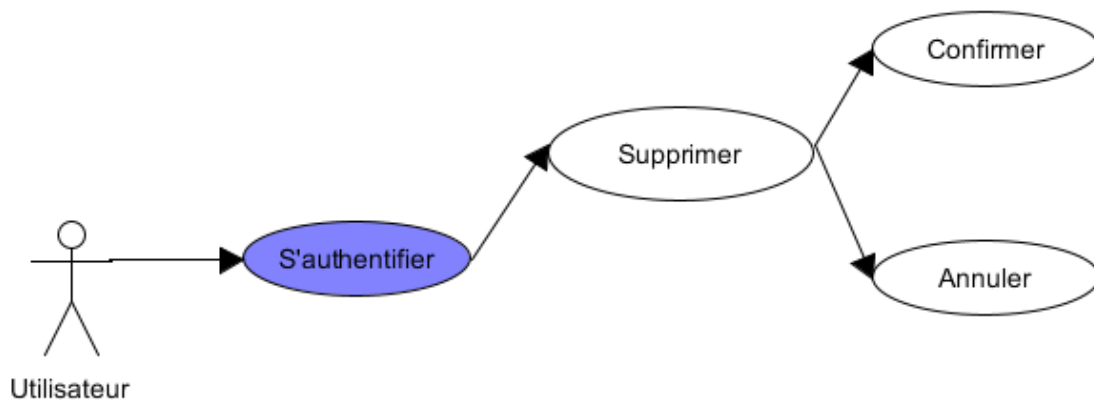


Figure 9: Diagramme de suppression de données

➤ **Diagramme de mise à jour des données**

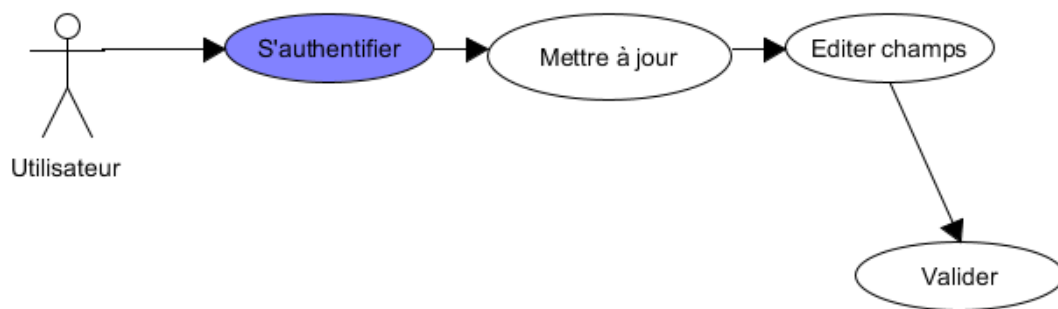


Figure 10: Diagramme de mise à jour des données

III-1-3 Diagramme de séquence

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de requêtes.

On présentera quelques diagrammes de séquences relatifs aux cas d'utilisations

➤ Diagramme de séquence d'authentification

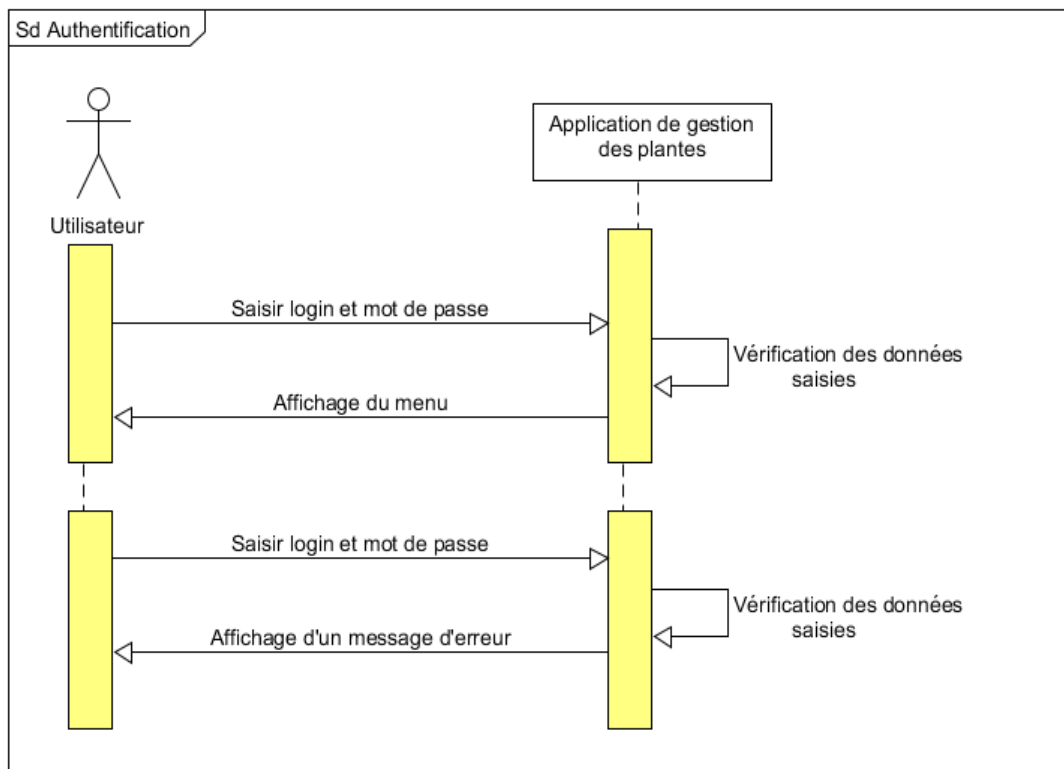


Figure 11: Diagramme de séquence d'authentification

➤ Diagramme de séquence de création de nouvelles données

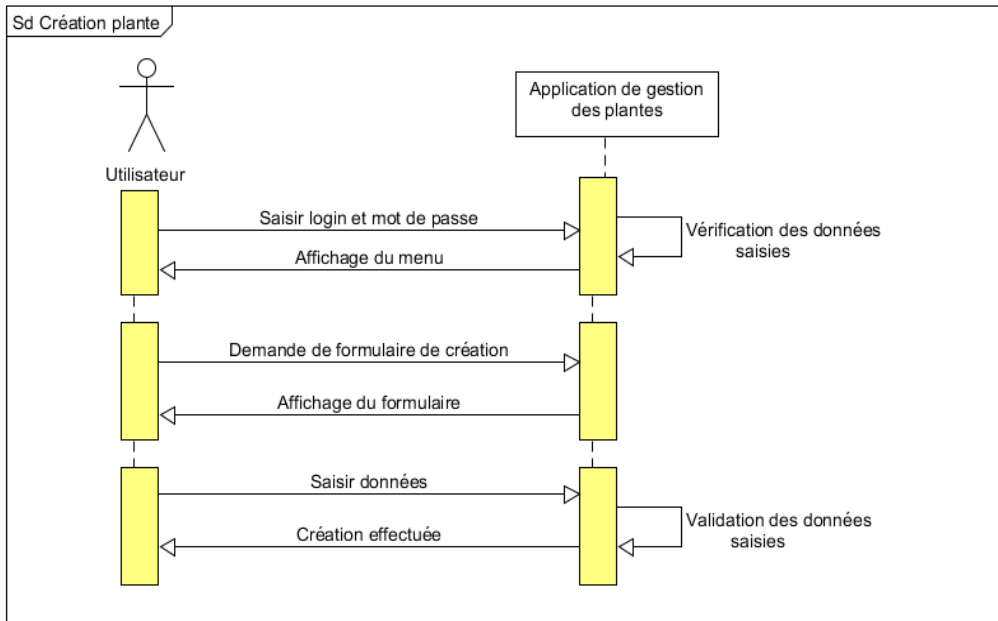


Figure 12: Diagramme de séquence de création de nouvelles données

➤ Diagramme de séquence de modification des données

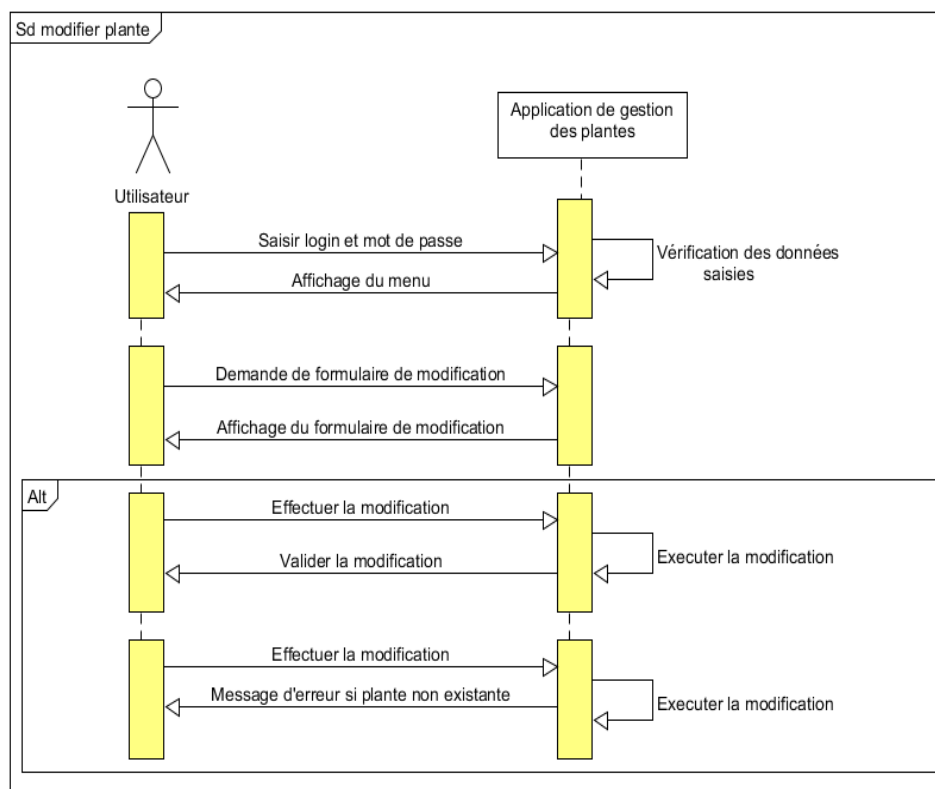


Figure 13: Diagramme de séquence de modification de données

➤ *Diagramme de séquence de suppression de données*

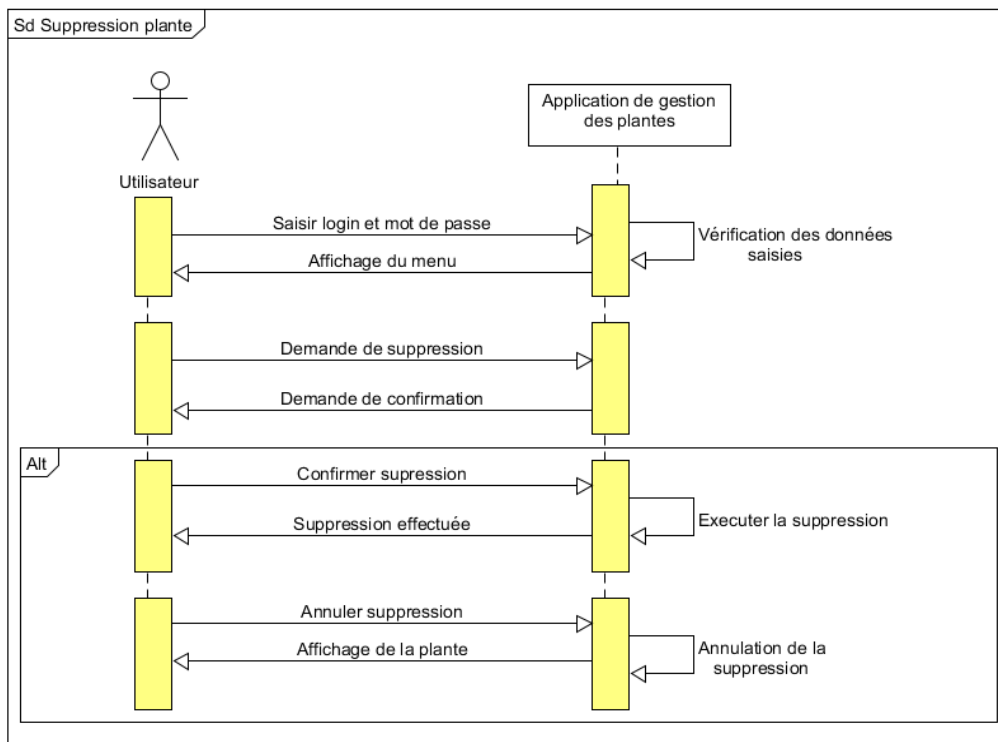


Figure 14: *Diagramme de séquence de suppression de données*

III-2 Mise en place de la base de données

III-2-1 Diagramme de classe

Dans la phase d'analyse, ce diagramme représente les entités (des informations) manipulées par les utilisateurs. Dans la phase de conception, il représente la structure objet d'un développement orienté objet. [15]

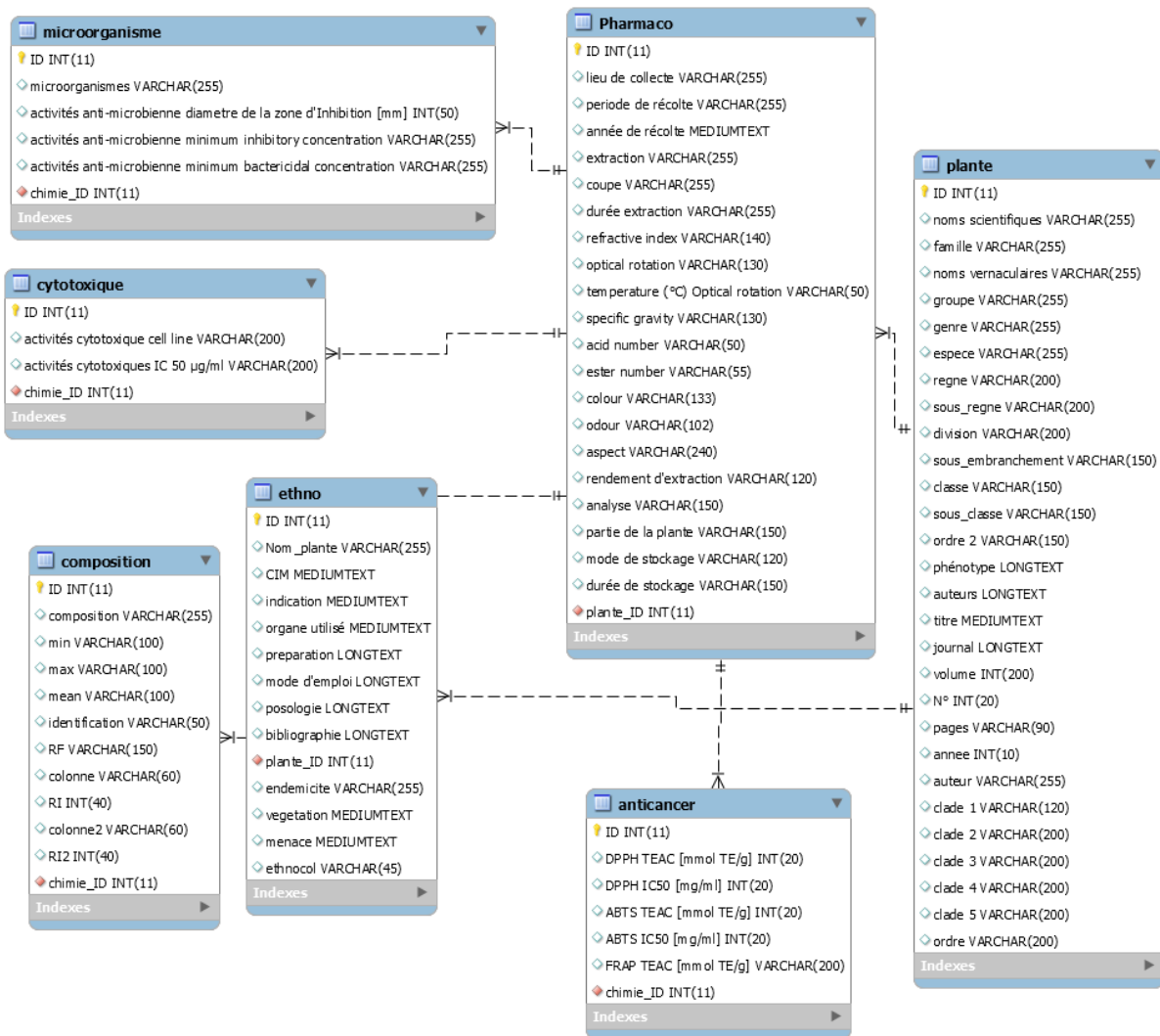


Figure 15: : Diagramme de classe

III-2-2 Description des classes du système

La base de données est réalisée avec phpMyAdmin, nous proposons une architecture composée de sept (07) tables.

Table	Action	Rows	Type	Collation	Size	Overhead
anticancers	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	32 K18	-
compositions	★ Browse Structure Search Insert Empty Drop	8,578	InnoDB	utf8mb4_unicode_ci	1.7 M18	-
cytotoxiques	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_unicode_ci	32 K18	-
ethnos	★ Browse Structure Search Insert Empty Drop	27,423	InnoDB	utf8mb4_unicode_ci	8.5 M18	-
failed_jobs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 K18	-
microorganismes	★ Browse Structure Search Insert Empty Drop	75	InnoDB	utf8mb4_unicode_ci	32 K18	-
migrations	★ Browse Structure Search Insert Empty Drop	18	InnoDB	utf8mb4_unicode_ci	16 K18	-
password_resets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 K18	-
pharmacos	★ Browse Structure Search Insert Empty Drop	265	InnoDB	utf8mb4_unicode_ci	144 K18	-
plantes	★ Browse Structure Search Insert Empty Drop	7,143	InnoDB	utf8mb4_unicode_ci	1.5 M18	-
users	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 K18	-
11 tables	Sum	43,499	InnoDB	latin1_swedish_ci	12 M18	0 B

Figure 16: Tables de la base de données

Nous pouvons voir ci-dessus l'ensemble des tables de la base de données. Nous allons maintenant les présenter sous forme détaillée :

- Plantes : Cette table servira pour le dépôt de plantes. Chaque document possédera un identifiant unique (clé primaire) ainsi qu'un nom. La date de son dépôt, et sa provenance ;
- Pharmaco : Cette table regroupe les données des plantes du pharmacologie ;
- Ethnos : Cette table regroupe les données des plantes du département ethnobotanique ;
- Compositions : Cette table regroupe les données des compositions des plantes ;
- Microorganismes : Cette table regroupe les données des microorganismes des plantes ;
- Anticancers : Cette table regroupe les données des plantes anticancers ;
- Cytotoxiques : Cette table regroupe les données des plantes ayant des propriétés cytotoxiques.

Ces tables et leurs structures nous semblent le minimum requis pour répondre aux besoins des utilisateurs.

Il est à noter que quatre (04) tables ont été ajoutés automatiquement grâce à la migration de Laravel :

- Users : Cette table regroupe les données des utilisateurs ;
- Failed_jobs : C'est une table qui concerne les queues ;
- Password_reset : Cette table liée à la Users qui permet de gérer le renouvellement des mots de passe en toute sécurité ;
- Migrations : C'est la table où sont listés toutes les migrations par Laravel.

Nous avons présenté les différents diagrammes définis qui ont permis de bien comprendre les besoins de l'application à développer ainsi que les différentes interactions, nous allons à présent passer à la phase d'implémentation.

CHAPITRE IV: IMPLÉMENTATION ET RÉSULTATS

IV-1 Plateforme de développement

IV-1-1 Laragon

Laragon, est un serveur de développement web pour Windows. Une fois installé il propose nativement et par défaut les outils suivants : Apache, Nginx, PHP, MySQL, Cmdr, HeidiSQL, Notepad++, Netstat, WinSCP, Putty.

L'énorme avantage de Laragon c'est qu'il est extensible, portable, c'est-à-dire que son environnement est isolé.

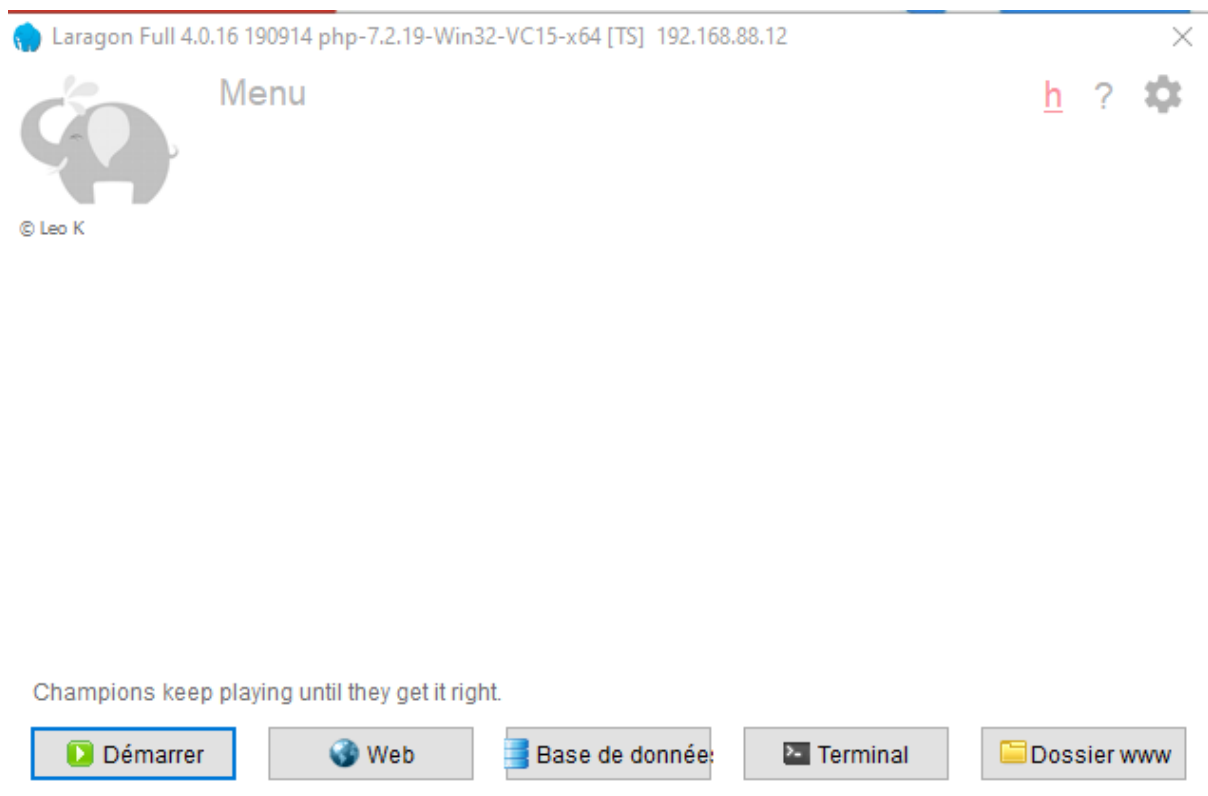


Figure 17: Interface Laragon

IV-1-2 PhpMyAdmin

Laragon étant fourni de base avec HeidiSQL, il a fallu installer manuellement PhpMyAdmin.

PhpMyAdmin est un logiciel libre écrit en PHP qui a pour mission de s'occuper de l'administration d'un serveur de base de données MySQL ou MariaDB. Il peut être utilisé pour

réaliser la plupart des tâches d'administration, ceci incluant la création de base de données, l'exécution de demandes, et l'ajout de comptes utilisateur.

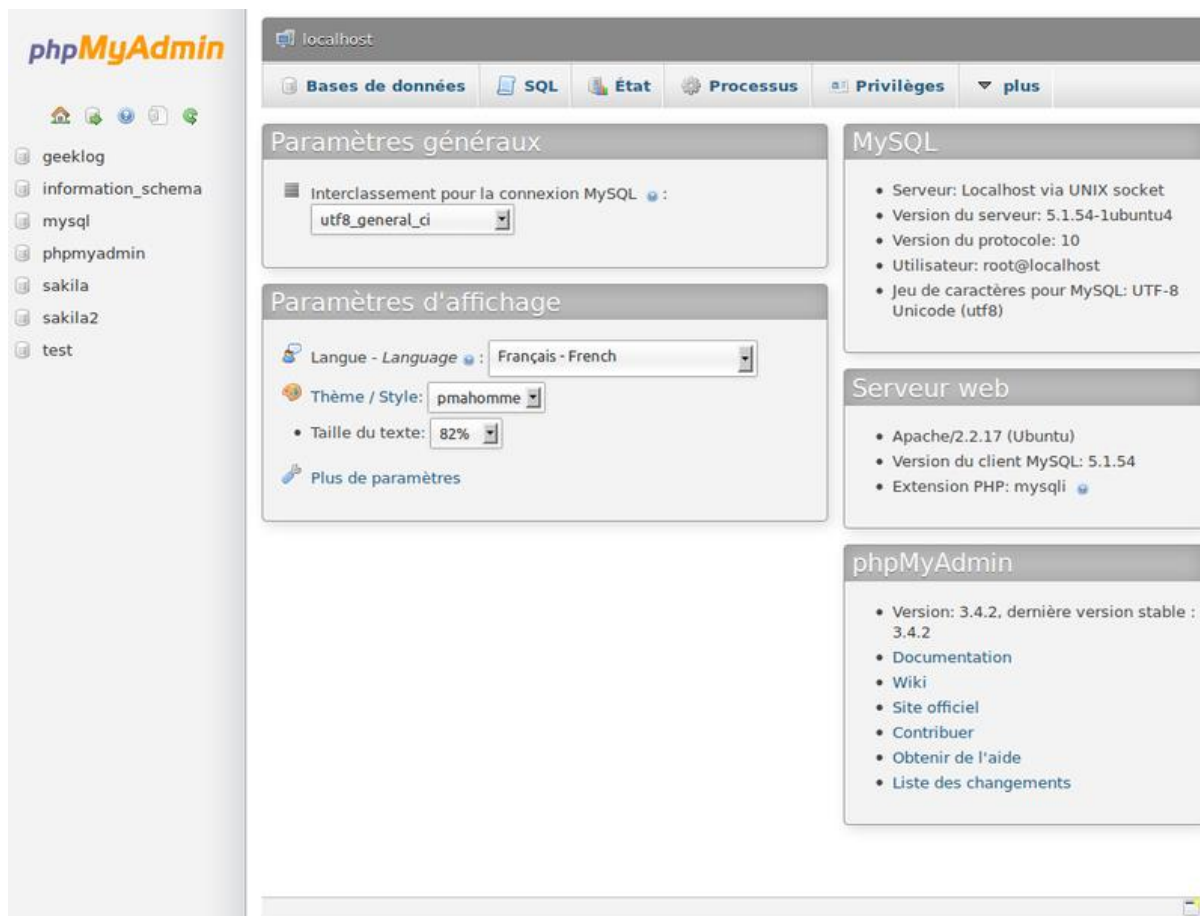


Figure 18: Interface PhpMyAdmin

IV-1-3 Langage de développement

➤ HTML

Développé pour la première fois par Tim Berners-Lee en 1990, HTML est l'abréviation de Hypertext Markup Language. HTML est utilisé pour créer des documents électroniques (appelés pages) qui sont affichés sur le World Wide Web. Chaque page contient une série de connexions à d'autres pages appelées hyperliens. Chaque page Web que vous voyez sur Internet est écrite en utilisant une version de code HTML ou une autre.

Le code HTML garantit le bon formatage du texte et des images pour votre navigateur Internet. Sans HTML, un navigateur ne saurait afficher du texte sous forme d'éléments ou charger des images ou d'autres éléments. HTML fournit également une structure de base de la page, sur laquelle les feuilles de style en cascade sont superposées pour changer son apparence.

On pourrait penser au HTML comme aux os (structure) d'une page Web et au CSS comme à sa peau (apparence).

➤ **CSS**

Cascading Style Sheets est un langage utilisé pour décrire les styles réutilisables pour la présentation de documents écrits dans un langage de balisage. Son concept a été créé par Håkon Wium Lie en 1994. En décembre 1996, CSS a été créé par le W3C et permet aujourd'hui aux développeurs Web de modifier la mise en page et l'apparence de leurs pages Web. Par exemple, CSS peut être utilisé pour changer la police utilisée dans certains éléments HTML, ainsi que sa taille et sa couleur. Un seul fichier CSS peut être lié à plusieurs pages, ce qui permet à un développeur de modifier l'apparence de toutes les pages en même temps.

➤ **PHP**

Créé par Rasmus Lerdorf en 1994 et rendu public le 8 juin 1995, PHP, qui est l'abréviation de PHP : Hypertext Preprocessor, est un langage de script interprété côté serveur. Il a été conçu pour créer des pages Web dynamiques et des pages Web qui fonctionnent efficacement avec des bases de données. [13]

➤ **MySQL**

Introduit en 1998, MySQL est un SGBDR open source (système de gestion de base de données relationnelle). MySQL est largement utilisé et constitue une alternative populaire aux autres solutions SQL car il est open source et peut être téléchargé gratuitement.

Les fichiers qui incluent du code PHP sur un serveur Web peuvent avoir n'importe quelle extension de fichier, mais le plus souvent, ils se terminent par .PHP, .PHP3 ou PHTML.

Un fichier PHP peut être créé et son contenu peut être visualisé à l'aide d'un éditeur de code source, tel que Dreamweaver ou Notepad ++.

➤ **JQuery**

En JavaScript, JQuery est un package de script JavaScript destiné à effectuer des tâches courantes et répétitives en JavaScript avec un code HTML court et compréhensible. La première version stable de jQuery a été publiée par John Resig le 26 janvier 2006, avec plusieurs versions supplémentaires depuis.

IV-1-4 Visual Studio

Visual Studio, également connu sous le nom de Microsoft Visual Studio et VS, est un environnement de développement intégré pour Microsoft Windows. C'est un outil pour écrire des programmes informatiques, des sites Web, des applications Web et des services Web. Il comprend un éditeur de code, un débogueur, un outil de conception graphique et un concepteur de schéma de base de données, et prend en charge la plupart des principaux systèmes de contrôle de révision. Il est disponible à la fois en version gratuite "Community" et en version commerciale payante.

Langages de programmation pris en charge par Visual Studio :

C, C ++, C #, Visual Basic .NET, F#, Fossil, Python, HTML / XHTML / CSS, JavaScript.



Figure 19: Logo de Visual Studio

IV-1-5 Node JS

Node.js, également appelé Node, est un environnement d'exécution pour les applications côté serveur, lancé pour la première fois en 2009. Il s'agit d'une architecture logicielle open source orientée événement, elle détecte et réagit aux choses qui se produisent, au fur et à mesure qu'elles se produisent.

Node fournit une API d'entrée / sortie non bloquante, ce qui signifie que les programmes de Node peuvent effectuer des demandes d'E / S et continuer sans attendre la réponse. Cette approche a des utilisations importantes dans les applications Web, où l'utilisateur peut utiliser votre site, page ou application même lorsque les E / S se produisent en arrière-plan. Ce type d'application est appelé asynchrone car les opérations individuelles peuvent se dérouler indépendamment du déroulement du programme principal.

Les programmes de Node sont écrits en JavaScript. Son moteur JavaScript est V8, le moteur open source qui a été développé chez Google pour son navigateur Web Chrome. [14]



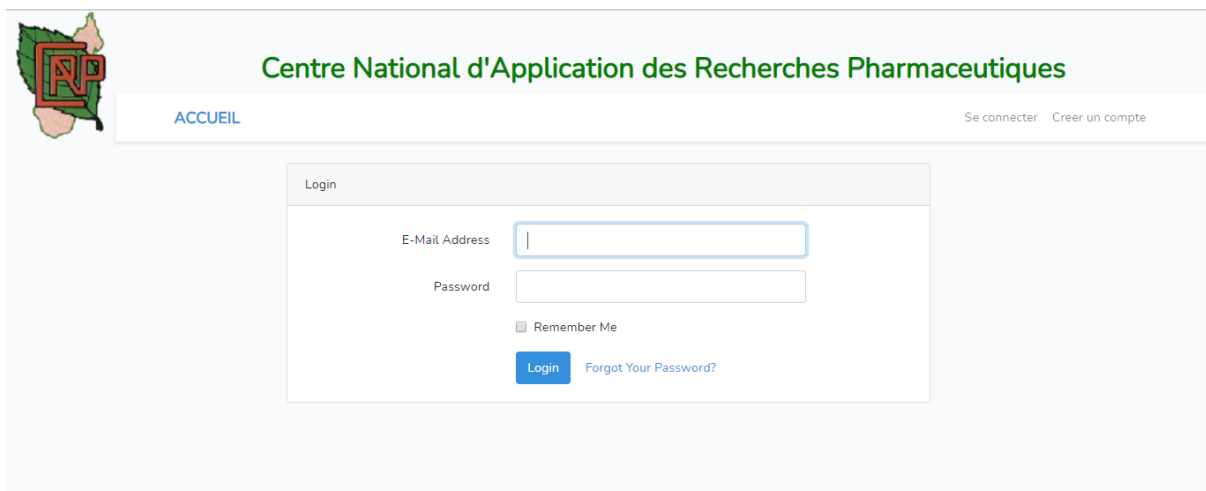
Figure 20: Logo de Node JS

IV-2 Interface de l'application

Comme toute application, notre application doit être facile à utiliser et compréhensive par tout type de cible.

IV-2-1 Authentification

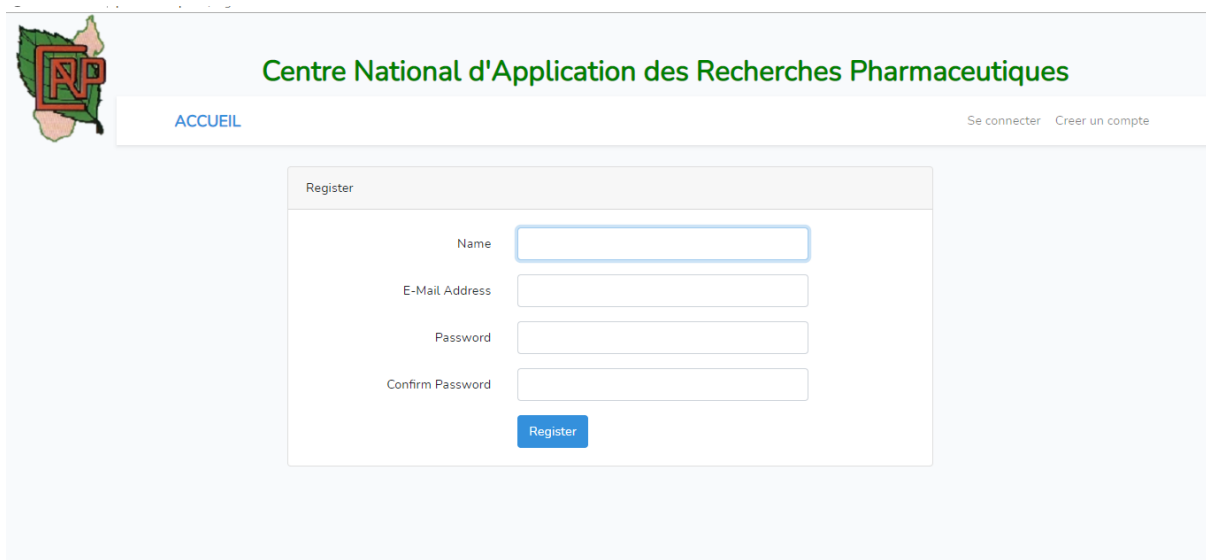
L'authentification permet à l'utilisateur d'accéder à l'application en saisissant le login et le mot de passe, si tous les champs sont bien remplis et les informations saisies sont correcte, elle se redirige vers la page d'accueil, sinon il est redirigé vers la page d'authentification.



The screenshot shows the login interface of the 'Centre National d'Application des Recherches Pharmaceutiques'. The header includes a logo on the left, the site name in green, and a blue 'ACCUEIL' link. On the right, there are links for 'Se connecter' and 'Créer un compte'. The main content area is titled 'Login' and contains a form with fields for 'E-Mail Address' and 'Password'. Below these fields is a 'Remember Me' checkbox and a blue 'Login' button. A link for 'Forgot Your Password?' is also present.

Figure 21: Connexion

La figure ci-dessous montre la page d'inscription de l'application :



The screenshot shows the registration interface of the 'Centre National d'Application des Recherches Pharmaceutiques'. The header is identical to the login page. The main content area is titled 'Register' and contains a form with fields for 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. A blue 'Register' button is located at the bottom of the form.

Figure 22: Inscription

IV-2-2 Menu principal

L'application se compose de plusieurs interfaces qui guident l'utilisateur vers différents modules. En fonction du choix de l'utilisateur, d'autres interfaces seront affichées par l'application.

➤ **Accueil avant authentification**

La figure ci-après présente l'accueil de l'application au démarrage.

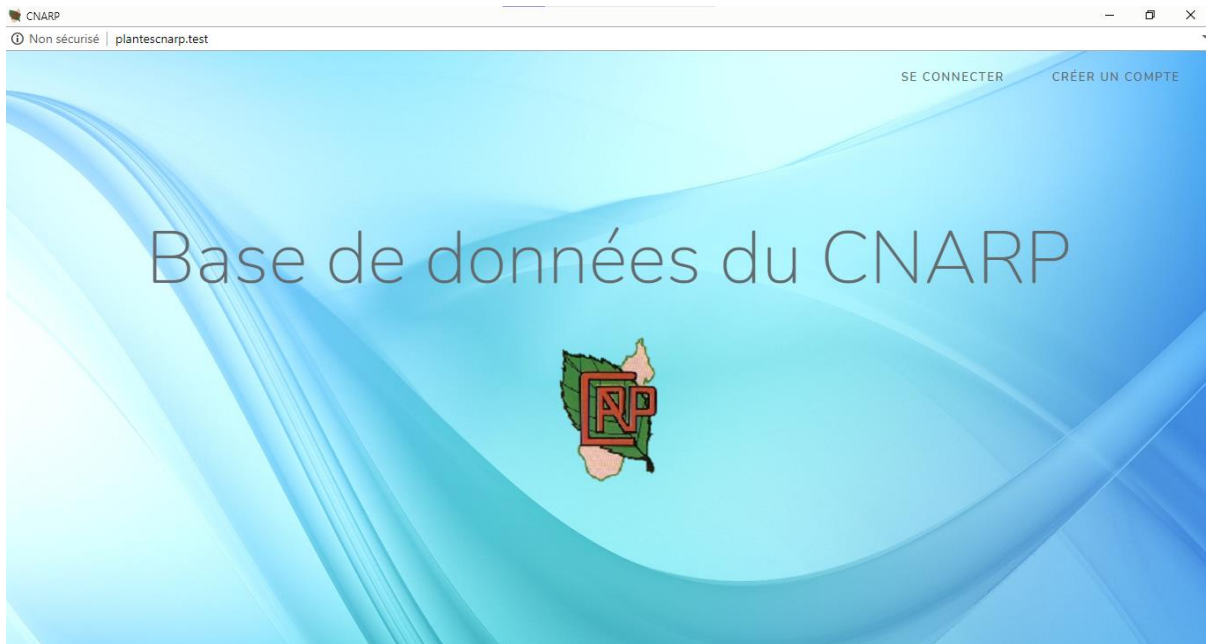


Figure 23: Accueil avant authentification

➤ **Accueil après authentification**

Sur la figure ci-dessous, il y a la barre de navigation qui permet de naviguer entre les différentes données de chaque département.



Figure 24: Accueil après authentification

IV-2-3 Affichage des informations dans la base de données




La figure ci-dessous affiche la page où est listée toutes les plantes du CNARP
















Listes des Plantes

Noms Scientifiques	Famille	Noms Vernaculaires	Groupe	Genre	Espèce	Regne	Sous Règne	Division	Sous Embranchement
Aloe deltoideodonta	Asphodelaceae	vahona	Monocotyledones	Aloe	deltoideodonta	-	-	-	-
Astrocassine sp	Celastraceae	-	-	Astrocassine	sp	-	-	-	-
Brochoneura sp	Myristicaceae	-	-	Brochoneura	sp	-	-	-	-
Capurodendron tampinense	Sapotaceae	-	-	Capurodendron	tampinense	-	-	-	-
Cladogelonium madagascariense	Euphorbiaceae	tsontso	Dicotyledones	Cladogelonium	madagascariense	-	-	-	-

Figure 25: Liste des informations

Différentes options sont proposées par l'application, notamment :

- Éditer 
- Afficher 
- Supprimer 

Composition	Min	Max	Mean	Identification	RF	Colonne	RI	Colonne	RI	Référence de la plante	Plante récoltée	Actions
2-melhoxy -4-methy lp henol	-	-	0	MS, RI	-	HP1-MS	1169	-	-	Cananga odorata 3	Cananga odorata	  
5-indanol	-	-	0,02	MS, STD	-	HP1-MS	1308	-	-	Cananga odorata 6	Cananga odorata	  
Fenchyl acetate	-	-	0	-	-	SPB-29	1225	Supelcowax	1443	Cedrelopsis grevei 1	Cedrelopsis grevei	  
(E)-"beta"-Farnesene	-	-	0,7	RI, MS, 13C NMR	-	Rtx-1	1447	Rt-wax	1656	Cedrelopsis grevei 9	Cedrelopsis grevei	  
"alpha"-Pinene	-	-	0,9	-	-	Rtx-1	931	Rt-wax	1025	Cedrelopsis grevei 14	Cedrelopsis grevei	  

«
1
2
3
4
5
6
7
8
9
10
...
1713
1714
»

Figure 26: Actions de l'application

IV-2-4 Création de nouvelles données

La figure ci-dessous est la page d'ajout d'une nouvelle information. Tous les champs doivent être remplis et le champ ne doit pas être déjà existante, sinon il est impossible d'ajouter les informations à la base de données.

IV-2-7 Recherche dans la base de données

L'application est pourvue d'un module de recherche qui permet de rechercher à travers toute la base de données.

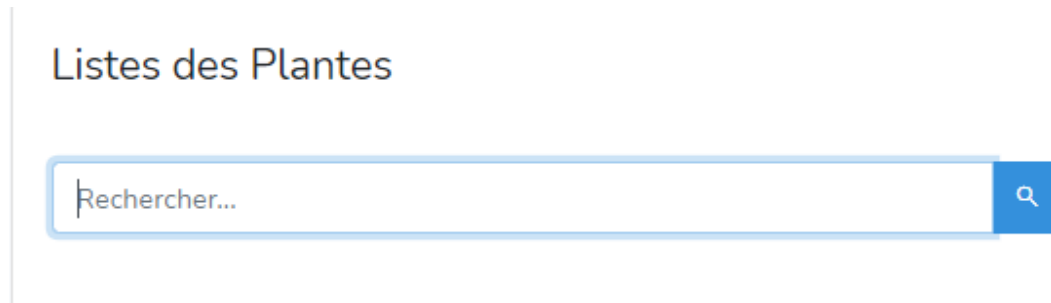


Figure 30: Recherche dans la base de données

IV-3 Interprétations des résultats

La réalisation de l'application web de gestion des plantes du CNARP nous a permis d'apporter une large amélioration dans la gestion de leurs plantes ainsi que la satisfaction du membre du personnel de chaque département. En effet, d'une part, elle a facilité le triage des données et la classification de chaque espèce auprès des chercheurs avec une durée bien réduite dans le processus de recherche.

Elle répond rapidement à la demande du personnel pour connaître après mise à jour des effectifs de chaque espèce. De même, on peut aussi modifier les données d'une plante existante.

Après quelques tests, on a pu constater que notre application est actuellement opérationnelle ; en ce sens que le temps de recherche dans la base de données par les utilisateurs a été réduit, loin des fichiers d'Excel séparés d'auparavant.

L'utilisation de l'application ne nécessite aucune notion en informatique. Ainsi, l'accès aux données est simple et pratique pour les utilisateurs.

L'application de gestion de plante répond à la majorité des besoins du CNARP, les utilisateurs des bureaux où le programme est installé sont satisfaits puisqu'on a respecté le principe « la simplicité fait la beauté ». L'application est simple et performante. Parmi les points forts de l'application, on trouve ainsi :

- Interface simple et facile à utiliser ;
- Restauration de la base de données à tout moment ;
- Utilisation de Laravel pour économiser le temps de rechargement des pages ;
- Recherche simplifiée et rapide.

CONCLUSION

Au terme de ce travail, on a pu constater que la gestion des données est primordiale pour chaque entreprise. Une bonne gestion des données facilite le travail des employés, d'où ce mémoire dont l'objectif était la mise en place d'une application web conçu pour gérer les données sur les plantes du CNARP.

L'ensemble du processus de la création de l'application a été décrite à partir de la motivation, montrant la raison pour laquelle il a été décidé de mettre en œuvre cette application, en poursuivant avec une analyse approfondie d'où les principes de base et les exigences fonctionnelles de l'application a émergé.

Cette application Web utilise un Framework Opensource Laravel qui est basé sur des technologies tels que HTML, PHP, et MySQL. Pour développer cette application web, il fallait utiliser un backend qui permet au web administrateur de gérer facilement et rapidement les informations qui était essentiel. Divers open source de système de gestion ont été utilisés pour la première fois, mais après avoir comparé la flexibilité et la facilité d'utilisation, Laravel a été choisi comme backend pour gérer l'application Web et Bootstrap a été choisi pour le frontend. Des procédures bien défini ont aussi été suivi pour la conception de la base de données à partir de l'architecture MVC (Modèle Vue Contrôleur).

Notre stage a été enrichissant avec l'analyse des besoins fonctionnels et non-fonctionnels de l'application ainsi que l'élaboration de divers diagrammes qui nous engageaient dans un scénario précis [16]. Nous avons pu acquérir aussi une expérience sur le développement et la fonctionnalité de Laravel avec son extrême serviabilité pour ce qui est de la conception.

L'application réalisée permet d'assurer plusieurs fonctionnalités de base telles que l'affichage des plantes, la création de nouvelle plante, la suppression d'une plante existante, la mise à jour ainsi que la recherche dans la base de données.

Néanmoins, l'application n'est pas encore au meilleure de ses performances, beaucoup reste à améliorer, comme les interfaces pour qu'elles répondent aux critères ergonomiques, l'hébergement sur un serveur ou encore l'utilisation d'Intranet, ainsi qu'une meilleure sécurité pour les données.

RÉFÉRENCES

- [1] [<https://cnarp.recherche.gov.mg>] (Consulté le 28 Mai 2021)
- [2] [<https://www.ideematic.com/dictionnaireweb/applicationweb>] (Consulté le 08 Mars 2021)
- [3] [<https://www.futura-sciences.com/sante>] (Consulté le 28 Mai 2021)
- [4] [<https://baptiste-wicht.developpez.com/tutoriels/conception/>](Consulté le 17 Mai 2021)
- [5] J. CONALLEN. Concevoir des applications web avec UML. Eyrolles, 2000
- [6] [<https://www.computerhope.com>] (Consulté le 15 Mars 2021)
- [7] [<http://wikipedia.com>] Les vues UML (Consulté le 08 Mai 2021)
- [8] [<https://fr.wikipedia.org/wiki/Bootstrap>] (Consulté le 08 Mars 2021)
- [9] [<http://www.pcinpact.com>] Bootstrap et JQuery (Consulté le 15 Mars 2021)
- [10] [<https://laravel.com/docs/8.x/r>] (Consulté le 08 Mars 2021)
- [11] [<https://laurent-audibert.developpez.com>] (Consulté le 17 Mai 2021)
- [12] [<https://fr.wikipedia.org/wiki/PHP>] (Consulté le 01 Avril 2021)
- [13] Maurice Chavelli, Découvrez le Framework PHP Laravel, 2016
- [14] [<http://wikipedia.com/node.js>] (Consulté le 08 Mars 2021)
- [15] Concepts et langages des Base de Données Relationnelles, support de cours, IUT de Nice-
Département INFORMATIQUE
- [16] [<http://www.uml-sysml.org>] Diagramme UML et SysML (Consulté le 17 Mai 2021)

ANNEXES

Extrait de code de l'application de gestion de plantes

```
<!doctype html>

<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="icon" href="{{ asset('img/logo.png') }}">

    <p style="float:left"></p>

    <h1 align=center><font size=6.5 color=green>

    <br /><b>Centre National d'Application des Recherches Pharmaceutiques<br
/></b></font></h1>

    <!-- CSRF Token -->

    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>CNARP</title>

    <!-- Scripts -->

    <script src="{{ asset('js/app.js') }}" defer></script>


    <!-- Fonts -->

    <link rel="dns-prefetch" href="//fonts.gstatic.com">

    <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">


    <!-- Styles -->

    <link href="{{ asset('css/app.css') }}" rel="stylesheet">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/notyf@3/notyf.min.css">
```

```

<!-- Icons -->

<link                href="https://unpkg.com/ionicons@4.5.10-0/dist/css/ionicons.min.css"
rel="stylesheet">

<script              src="https://cdnjs.cloudflare.com/ajax/libs/turbolinks/5.2.0/turbolinks.js"
integrity="sha512-
G3jAqT2eM4MMkLMyQR5YBhvN5/Da3IG6kqgYqU9zIIH4+2a+GuMdLb5Kpxy6ItMdCfg
aKlo2XFhI0dHtMJjoRw==" crossorigin="anonymous"></script>

<script              src="https://cdn.jsdelivr.net/gh/alpinejs/alpine@v2.x.x/dist/alpine.min.js"
defer></script>

</head>

<body>

<div id="app">

    @include('layouts.nav')

    <main class="py-4">

        @yield('content')

    </main>

</div>

@stack('scripts')

<script src="https://cdn.jsdelivr.net/npm/notyf@3/notyf.min.js"></script>

@if (session()->has('success'))

<script>

```

```

    const notyf = new Notyf({ dismissible: true })

    notyf.success('{{ session('success') }}')

</script>

@endif


<script>

    /* Simple function to retrieve data url from file */

    function fileToDataURL(event, callback) {

        if (! event.target.files.length) return

        let file = event.target.files[0],

            reader = new FileReader()

        reader.readAsDataURL(file)

        reader.onload = e => callback(e.target.result)

    }

</script>

</body>

</html>

```

Conception d'une application web pour la gestion de plantes du CNARP

Résumé

Notre travail consiste à concevoir et réaliser d'une application web de gestion de plantes au Centre National d'Application de Recherches Pharmaceutiques. Cette application a été conçue dans le dessein de gérer les plantes du centre afin de faciliter l'accès aux données et de les sécuriser. Ainsi, son principal objectif est de pouvoir afficher, créer, éditer, mettre à jour les données et de pouvoir faire des recherches. Nous avons créé une application web, modélisée à partir du langage UML (langage de modélisation unifié). Le langage de programmation choisi est le langage PHP (Hypertext Preprocessor) et le système de gestion de base de données (SGBD) est MySQL. Ainsi, après conception, l'application a été fonctionnelle mais nécessite encore quelques ajustements.

Mots-clés : UML, base de données, PHP, Application web, plantes

Abstract

Our job is to design and build a plant management web application at the National Center for Pharmaceutical Research Application. This application has been designed with the intention of managing the plants in the center in order to facilitate access to and secure data. Thus, its main objective is to be able to display, create, edit, update data and to be able to do research. We have created a web application, modeled from UML (unified modeling language). The chosen programming language is PHP (Hypertext Preprocessor) and the database management system (DBMS) is MySQL. After design, the application was functional but still requires some adjustments.

Keywords : UML, database, PHP, Web application, plants

Rapporteur :

RAZANAMANAMPISOA Harimalala,

Maitre de conférences

Lieu de stage :

CNARP Ivandry

Impétrante :

RAJONARIVELO Maminirina

0348634159

Lot IVH6 Andravoahangy

ninierj@gmail.com