



UNIVERSITE DE FIANARANTSOA

ECOLE NATIONALE D'INFORMATIQUE

Rapport de stage en vue de passage en troisième année de Formation en Licence Professionnelle

Parcours : Informatique Générale

Intitulé :

**CONCEPTION ET REALISATION D'UN OUTIL DE GESTION
D'UNE CLINIQUE VETERINAIRE**

Présenté le :

Par :

- Monsieur RAJAONARISON Apollinaire Iabakidy Clairmont

CURRICULUM VITAE

Nom : RAJAONARISON

Prénoms : Apollinaire Iabakidy Clairmont

Date et lieu de naissance : 17 Août 1998 à Vangaindrano

Sexe : Masculin

Nationalité : Malagasy

Adresse : Soatsihadino Fianarantsoa 301

Contact : 034 68 047 17

Email : clairmont.rajaonarison@gmail.com

Développeur web junior



DIPLOMES ET FORMATIONS

2019 : 2ème année en Formation de Licence Professionnelle, Parcours Informatique Générale à l'Ecole Nationale d'Informatique

2018 : 1ère année de Formation en Licence Professionnelle, Parcours Informatique Générale à votre l'Ecole Nationale d'Informatique

2017 : Obtention du Baccalauréat, série D avec mention Bien, au Lycée Catholique Lucien BOTOVASOA Vangaindrano.

EXPERIENCES PROFESSIONNELS

Novembre – Janvier 2019 :

Lieu : Clinique Vétérinaire Boulevard, Ankazomanga Antananarivo 101

Thème de stage : Création d'un outil de gestion d'une clinique vétérinaire

Langages utilisés : PHP, Sql, HTML, CSS, JS

Frameworks : CodeIgniter3, Bootstrap

SGBD utilisé : MySql

2019 : Projet Python ; création d'un éditeur de texte avec PyQt5

2019 : Projet à l'Ecole portant sur la création d'une application web de gestion de magasin avec

le Framework PHP CodeIgniter 3.

2018 : Projet à l'école durant l'année, clonage de système d'exploitation Linux.

2018 : Projet portant sur la création d'un site web.

2018 : Projet portant sur l'animation de page avec JavaScript.

COMPETENCES EN INFORMATIQUES

-Langage de développement Web : JavaScript, HTML, PHP

-Langage de programmation : C, C++, Assembleur, Python, Visual Basic

-Frameworks : Qt, Codeigniter, Bootstrap

-Outils bureautiques : Microsoft Word, Excel, Powerpoint

COMPETENCES LINGUISTIQUES

| | Lecture | Compréhension à l'audition | Expression orale | Expression écrite |
|----------|-----------|-------------------------------|------------------|-------------------|
| FRANÇAIS | Très-Bien | Très-Bien | Très-Bien | Bien |
| ANGLAIS | Bien | Assez-Bien | Assez-Bien | Bien |

DIVERS

- Activités sportives pratiqués: Basket-ball, Football, Saut en hauteur
- Membre de l'Association FIkambanan'ny ZAnaray MAndrato FIanara à Fianarantsoa
- Vice-Président de l'Association des Jeunes Elites sortant du LCLB Vangaindrano
- Jeux vidéos

SOMMAIRE

| | |
|--|------|
| CURRICULUM VITAE | I |
| SOMMAIRE | III |
| LISTE DES FIGURES | VI |
| LISTE DES TABLEAUX | VIII |
| LISTE DES ABBREVIATIONS | IX |
| REMERCIEMENTS | XI |
| INTRODUCTION..... | 1 |
| PARTIE I : PRESENTATIONS | 2 |
| Chapitre 1 : Présentation de l'Ecole Nationale d'Informatique | 3 |
| 1.1 Informations d'ordre générale | 3 |
| 1.2 Missions et historiques | 3 |
| 1.3 Organigramme institutionnel de l'ENI | 4 |
| 1.4 Domaine de spécialisation | 6 |
| 1.5 Architecture des formations pédagogiques..... | 6 |
| 1.6 Relation de l'ENI avec les entreprises et les organismes | 9 |
| 1.7 Partenariat au niveau international | 10 |
| 1.8 Débouchés professionnels des diplômés | 11 |
| 1.9 Ressources humaines | 13 |
| Chapitre 2 : Présentation du Pharmacie et Cabinet Vétérinaire du Boulevard | 14 |
| 2.1 Information | 14 |
| 2.2 Historique | 14 |
| 2.3 Organigramme de la Pharmacie et CVB | 15 |
| 2.4 Missions et buts | 15 |
| 2.5 Soins offertes | 15 |
| 2.6 Partenaires et fournisseurs | 16 |
| Chapitre 3 : Description du projet | 17 |
| 3.1 Formulation du projet | 17 |
| 3.2 Objectif et besoin de l'utilisateur..... | 17 |

| | |
|--|-----------|
| 3.3 Moyens nécessaires à la réalisation du projet..... | 17 |
| 3.3.1 Moyens humains | 17 |
| 3.3.2 Moyens matériels | 18 |
| 3.3.3 Logiciels utilisés pour la mise en œuvre du projet..... | 19 |
| 3.4 Méthodologie pour la conduite du projet | 19 |
| 3.5 Résultat attendu | 20 |
| PARTIE II : ANALYSE ET CONCEPTION | 21 |
| Chapitre 4 : Analyse préalable | 22 |
| 4.1 Analyse de l'existant | 22 |
| 4.1.1 Organisation actuelle | 22 |
| 4.1.2 Inventaire des moyens logiciels | 22 |
| 4.2 Critique de l'existant | 23 |
| 4.3 Conception d'avant-projet | 23 |
| 4.3.1 Mise en avant du problème principal..... | 23 |
| 4.3.2 Scenario des solutions..... | 23 |
| 4.3.3 Ebauche de solution et justification du choix du moyen | 24 |
| 4.3.4 Justification du choix des outils | 24 |
| Chapitre 5 : Analyse conceptuelle..... | 28 |
| 5.1 Présentation de la méthode MERISE | 28 |
| 5.2 Dictionnaire des données | 29 |
| 5.3 Règles des gestions | 33 |
| 5.4 Diagramme des flux..... | 34 |
| 5.5 Modélisation des données et des traitements..... | 36 |
| 5.5.1 Le Modèle Conceptuel des Données..... | 36 |
| 5.5.2 Le Modèle Logique des Données | 36 |
| 5.5.3 Le Modèle Conceptuel des Traitements..... | 38 |
| 5.5.4 Le Modèle Organisationnel des Traitements | 40 |
| 5.5.5 Le Modèle Physique des Données | 42 |
| PARTIE III : REALISATION | 44 |
| Chapitre 6 : Installation et configuration des outils | 45 |

| | | |
|-------|--|------|
| 6.1 | Laragon et le SGBD MySQL | 45 |
| 6.2 | Visual Studio Code : notre outil de développement | 47 |
| | Chapitre 7 : Architecture de l'application | 49 |
| 7.1 | Téléchargement d'une application Codeigniter 3 | 49 |
| 7.2 | Description des dossiers et fichier au sein de notre application | 51 |
| 7.3 | Architecture de notre application | 52 |
| | Chapitre 8 : Développement de l'application..... | 53 |
| 8.1 | Création de la base de données..... | 53 |
| 8.2 | Codage de l'application | 56 |
| 8.2.1 | Création du dossier assets | 56 |
| 8.2.2 | Les controllers..... | 57 |
| 8.2.3 | Les Models | 58 |
| 8.2.4 | Les views | 59 |
| 8.3 | Présentation de l'application | 61 |
| 8.3.1 | Authentification | 61 |
| 8.3.2 | Page d'accueil | 61 |
| 8.3.3 | Consultation pour un patient..... | 62 |
| 8.3.4 | Profil d'un patient | 64 |
| 8.3.5 | Profil d'un client | 65 |
| 8.3.6 | Page finance | 67 |
| | CONCLUSION ET PERSPECTIVES | 68 |
| | BIBLIOGRAPHIE | XI |
| | WEBOGRAPHIE..... | XII |
| | GLOSSAIRE | XIII |
| | RESUME..... | XIV |
| | ABSTRACT | XIV |

LISTE DES FIGURES

| | |
|--|----|
| Figure 1: Organigramme de l'ENI..... | 5 |
| Figure 2: Organigramme de la Pharmacie et Cabinet Vétérinaire du Boulevard..... | 15 |
| Figure 3: Logos des fournisseurs et partenaires du CVB | 16 |
| Figure 4: Méthodologie de la conduite du projet | 20 |
| Figure 5: Diagramme de flux | 34 |
| Figure 6: Modèle Conceptuel de Données | 36 |
| Figure 7: Modèle Logique des Données | 38 |
| Figure 8:MCT de la réception au clinique | 39 |
| Figure 9:MCT pour une vaccination | 39 |
| Figure 10:MCD pour le traitement d'un patient..... | 40 |
| Figure 11: MCT sur les payements | 40 |
| Figure 12: Modèle Physique des Données | 43 |
| Figure 13:Téléchargement du fichier Zip de Laragon..... | 45 |
| Figure 14: Page d'accueil de l'application Laragon..... | 45 |
| Figure 15: Authentification pour la connexion à la BDD | 46 |
| Figure 16: Liste des bases des données et des tables | 46 |
| Figure 17: Site officiel pour le téléchargement de VS Code..... | 47 |
| Figure 18: Installation de VS Code | 48 |
| Figure 19: Page d'accueil..... | 48 |
| Figure 20 : Crédit d'une application CodeIgniter en ligne de commande avec composer..... | 49 |
| Figure 21: Structure par défaut de l'application..... | 50 |
| Figure 22: Clonage d'un projet sous GitHub | 50 |
| Figure 23: Contenu du dossier application..... | 51 |
| Figure 24: Génération du fichier SQL sous MySQL Workbench..... | 53 |
| Figure 25: Filtrage des objets à exporter | 54 |
| Figure 26: Création du fichier SQL..... | 54 |
| Figure 27: Importation d'un fichier SQL sous Laragon..... | 55 |
| Figure 28: Charger un fichier SQL..... | 55 |
| Figure 29: Exécution du fichier SQL | 56 |
| Figure 30: La base de données dans Laragon..... | 56 |
| Figure 31: Dossier assets | 57 |
| Figure 32: Controller vaccination | 58 |
| Figure 33 : ModelSoin..... | 59 |
| Figure 34: View index du controller Patient | 59 |
| Figure 35: Fichier patient.JS | 60 |
| Figure 36: Fichier JS pour les propriétaires. | 60 |
| Figure 37: Authentification | 61 |
| Figure 38: Page d'accueil de l'application | 62 |

| | |
|--|----|
| Figure 39: Traitement pour un patient..... | 63 |
| Figure 40: Aperçu de la facture | 63 |
| Figure 41: Profil d'Amonne..... | 64 |
| Figure 42 : Historique de ses soins et traitements | 64 |
| Figure 43 : Historiques des consultations au cabinet | 65 |
| Figure 44 : Médicaments reçus lors du traitement | 65 |
| Figure 45 : Profil pour un propriétaire | 66 |
| Figure 46: Liste des dépenses..... | 66 |
| Figure 47: Règlement des restes à payer | 67 |
| Figure 48 : Page finance | 67 |

LISTE DES TABLEAUX

| | |
|--|----|
| Tableau 1: Organisation du système de formation pédagogique de l'Ecole | 6 |
| Tableau 2: Architecture des études au système LMD..... | 7 |
| Tableau 3: liste des formations existantes..... | 8 |
| Tableau 4: Débouchés professionnels éventuels des diplômés | 12 |
| Tableau 5: Moyens humains pour la réalisation de l'application..... | 18 |
| Tableau 6: Serveur de CVB | 18 |
| Tableau 7: Matériels côtés clients pour le projet..... | 18 |
| Tableau 8: Logiciels utilisés pour la réalisation du projet..... | 19 |
| Tableau 9: Les moyens personnels dans la gestion du cabinet | 22 |
| Tableau 10: Les moyens matériels | 22 |
| Tableau 11: Les solutions proposées..... | 24 |
| Tableau 12: Dictionnaire de données | 30 |
| Tableau 13: Modèle Organisationnel des Traitements..... | 41 |

LISTE DES ABBREVIATIONS

AJAX: Asynchronous JavaScript And XML

ASP: Active Server Pages

BDD: Base De Données

CI: Codeigniter

CPU: Central Processing Unity

CSS: Cascading Style Sheets

CVB: Cabinet Vétérinaire Boulevard

DATA: Do All TAsk

GNU: GNU is Not Unix

GPL: General Public License

HTML: HyperText Markup Language

IDE: Integrated Development Environment

JS: JavaScript

LAN: Local Area Network

LTS: Long Term Support

MCD : Modèle Conceptuel de Données

MCT : Modèle Conceptuel des Traitements

MLD : Modèle Logique des Données

MOT : Modèle Organisationnel des Traitements

MOPT : Modèle Opérationnel et Physique des Traitements

MPD : Modèle Physique des Données

MVC : Model-View-Controller

ORM: Object Relational Mapping

OS: Operating System

PDG : Président et Directeur Général

PME: Petites et Moyennes Entreprises

RAM: Random Access Memory

RG: Règle de Gestion

SGBD: Système de Gestion de Base de Données

SQL: Structured Query Language

URL: Uniform Resource Locator

VS: Visual Studio

WAN: Wide Area Network

WAMPP: Windows Apache MySQL Php et Perl

XHTML: eXtensible HyperText Markup Language

XAMPP: X Apache MySQL Perl PHP

REMERCIEMENTS

En préambule à ce rapport, je tiens à adresser mes remerciements les plus sincères aux personnes ayant contribué à l'élaboration et la mise en œuvre de ce document ainsi qu'à la réussite de cette incroyable année universitaire.

Je tiens à remercier plus particulièrement :

- Monsieur Le Président de l'université de Fianarantsoa, le Professeur certifié RAFAMANTANANTSOA Fontaine qui a su mener cette année scolaire à son terme.
- Monsieur Le Directeur de l'Ecole Nationale d'Informatique, Professeur RAMAMONJISOA Bertin Olivier, sans qui ce stage n'aurait pas eu lieu.
- Monsieur Le Président et Directeur Générale du Cabinet Vétérinaire Boulevard ANKAZOMANGA, Monsieur RAKOTONIRINA Avana, qui m'a accueilli en tant que stagiaire et m'a encadré professionnellement, ainsi que tous les personnels de cette clinique, qui ont grandement et gentiment collaboré d'une manière ou d'une autre à la bonne réalisation de ce stage.
- Monsieur RAZAFINDRAMONJA Clément Aubert, en tant qu'encadreur pédagogique, pour l'assistance nécessaire et les précieux conseils qu'il a suggérés.

Je tiens également mes profondes gratitude envers mes parents qui ont été le fer de lance, pour leur soutien psychologiques et financiers.

Mes sincères remerciements sont aussi adressés à mes proches, qui ont été là à me soutenir durant de ce stage.

INTRODUCTION

Les Petites et Moyennes Entreprises commencent à vraiment se développer actuellement, au point de pouvoir devenir des références dans quelques années. Pouvoir rivaliser avec les autres deviennent alors une des marches à faire pour la réussite. Dans cette perspective, certains d'entre eux investissent dans des nouvelles systèmes d'informations pour pouvoir gagner du temps et plus encore, faire des suivis plus performants et ainsi satisfaire au mieux leurs clientèles. Pour accélérer ce développement, les PME se lancent aussi à la course aux systèmes informatiques, qui permettent aujourd'hui selon les besoins et les intentions exprimés, d'accompagner, d'automatiser et de dématérialiser quasiment toutes les opérations incluses dans les activités ou procédures d'entreprise.

Lorsque ces moyennes entreprises travaillent manuellement sur l'enregistrement des données, on n'a plus de suivis sur les activités et souvent, gérer les historiques deviennent un travail herculéen. Ces données, stockés manuellement peuvent être la cible des actes frauduleuses, erronées ou bien perdues.

La Clinique Vétérinaire Boulevard Ankazomanga est sur le point d'automatiser complètement son système, c'est pourquoi le PDG de ce PME a lancé une appel d'offre à des stagiaires informaticiens pour initier ce travail faramineux, pour pouvoir les aider à mettre en place les fondements de l'automatisation de leur service. La réalisation d'un tel projet est toujours très complexe car il faudra revoir toutes les paramètres qui entrent en jeu dans ce processus. Réordonner les données, supprimer et créer certaines tâches ainsi que des moyens humains qui n'est pas à négliger, le travail se présente plus rude puisque qu'on va aller étape par étape pour arriver à une fin qui satisfait utilisateur et main d'œuvre.

Avant d'entrer dans le vif du sujet, il est pratique de détailler ce qu'on a entrepris afin d'arriver à la fin de ce travail. Le stage s'est grandement focalisé sur la conception d'une base de données et la réalisation d'une application web permettant d'automatiser et regrouper les données de la clinique pour avoir un suivi plus poussé et plus avancées. On va dans un premier temps commencer avec les présentations, puis ensuite passer à l'analyse et conception du projet et finir avec la réalisation du projet.

PARTIE I : PRESENTATIONS

Chapitre 1 : Présentation de l’Ecole Nationale d’Informatique

1.1 Informations d’ordre générale

L’Ecole Nationale d’Informatique, en abrégé ENI, est un établissement d’enseignement supérieur rattaché académiquement et administrativement à l’Université de Fianarantsoa. Le siège de l’Ecole se trouve à Tanambao- Antaninarenina à Fianarantsoa. L’adresse pour la prise de contact avec l’Ecole est la suivante : Ecole Nationale d’Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 020 75 508 01. Son adresse électronique est la suivante : *eni@univ-fianar.mg*. Site Web : www.Univ-fianara.mg/eni.

1.2 Missions et historiques

L’ENI se positionne sur l’échiquier socio-éducatif malgache comme étant le plus puissant vecteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd’hui comme la vitrine et la pépinière des élites informaticiennes du pays.

L’Ecole s’est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l’ENI était constituée et créée au sein du CUR par le décret N° 83-185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d’Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L’ENI a par conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

- En fournissant à des étudiants des connaissances de base en informatique ;
- En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises.
- En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l’information et de la communication (TIC).

L’implantation de cette Ecole Supérieure de technologie de pointe dans un pays en développement et dans une Province (ou Faritany) à tissu économique et industriel faiblement développé ne l’a pourtant pas défavorisée, ni empêchée de former des spécialistes informaticiens de bon niveau, qui sont recherchés par les entreprises, les sociétés et les organismes publics et privés sur le marché de l’emploi.

La filière de formation d'Analystes Programmeurs a été mise en place à l'Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d'ingénieurs a été ouverte à l'Ecole en 1986.

Dans le cadre du Programme de renforcement en l'Enseignement Supérieur (PRESUP), la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes des informatiques a été mise en place en 1986 grâce à l'appui matériel et financier de la Mission Française de coopération auprès de l'Ambassade de France à Madagascar.

Une formation pour l'obtention de la certification CCNA et / ou NETWORK +. Appelée « CISCO Networking Academy » a été créée à l'Ecole en 2002-2003 grâce au partenariat avec CISCO SYSTEM et l'Ecole Supérieure Polytechnique d'Antananarivo (ESPA). Cependant, cette formation n'avait pas duré longtemps.

Une formation de troisième cycle a été ouverte à l'Ecole a été ouverte à l'Ecole depuis l'année 2003 – 2004 grâce à la coopération académique et scientifique entre l'Université de Fianarantsoa pour le compte de l'ENI et l'Université Paul Sabatier de Toulouse (UPST). Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l'Informatique, et notamment pour préparer la relève des Enseignants Chercheurs qui étaient en poste.

Pendant l'année 2007-2008, la formation en vue de l'obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l'ENI avec les deux options suivantes de formation :

- Génie Logiciel et base de Données.
- Administration des Système et Réseaux.

La mise en place à l'Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD). Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

En vue de surmonter les difficultés de limitation de l'effectif des étudiants accueillis à l'Ecole, notamment à cause du manque d'infrastructures, un système de « Formation Hybride » a été mis en place à partir de l'année 2010. Il s'agit en effet d'un système de formation semi présentielles et à distance avec l'utilisation de la visioconférence pour la formation à distance. Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu'Université de Toliara.

1.3 Organigramme institutionnel de l'ENI

Cet organigramme de l'Ecole est inspiré des dispositions du décret N° 83-185 du 23 Mai 1983. L'ENI est administrée par un conseil d'Ecole, et dirigée par un directeur nommé par un décret adopté en conseil des Ministres. Le Collège des enseignants regroupant tous les enseignants-chercheurs de l'Ecole est chargé de résoudre les problèmes liés à l'organisation pédagogique des enseignements ainsi que à l'élaboration des emplois du temps. Le Conseil Scientifique propose les

orientations pédagogiques et scientifiques de l'établissement, en tenant compte notamment de l'évolution du marché de travail et de l'adéquation des formations dispensées par rapport aux besoins des entreprises. Trois départements de formation caractérisent l'organigramme :

- Le département de formation théorique à l'intérieur de l'Ecole ;
- Le département de formation pratique pour la coordination et la supervision des stages en entreprise et des voyages d'études ;
- Le département de formation doctorale pour l'organisation de la formation de 3ème cycle.

La figure 1 présente l'organigramme actuel de l'Ecole.

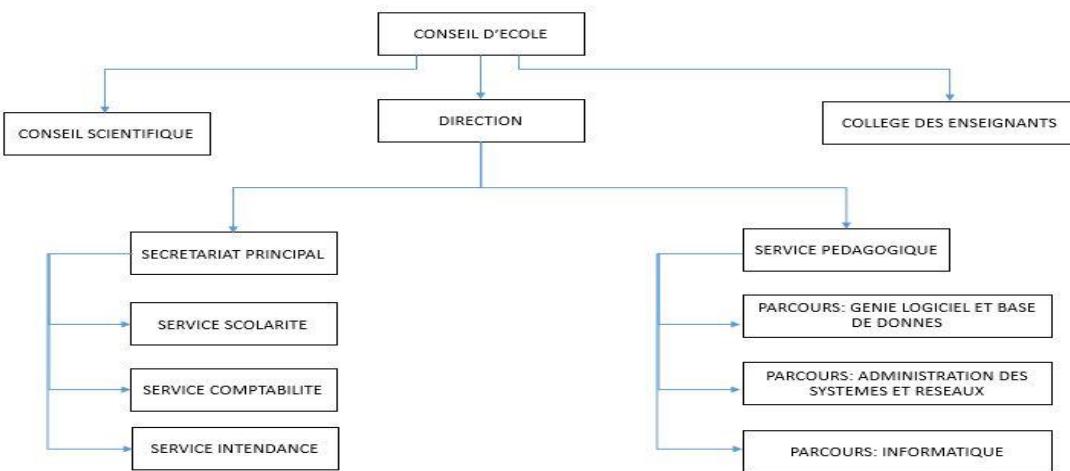


FIGURE 1: ORGANIGRAMME DE L'ENI

Sur cet organigramme, l'Ecole placée sous la tutelle académique et administrative de l'Université de Fianarantsoa, et dirigée par un Directeur élu par les Enseignants – Chercheurs permanents de l'Etablissement et nommé par un décret pris en Conseil des ministres pour un mandat de 3 ans.

Le Conseil d'école est l'organe délibérant de l'Ecole.

Le Collège des Enseignants propose et coordonne les programmes d'activités pédagogiques.

Le Conseil scientifique coordonne les programmes de recherche à mettre en œuvre à l'Ecole.

Le Secrétariat principal coordonne les activités des services administratifs (Scolarité, Comptabilité, et Intendance).

Conformément aux textes en vigueur régissant les Etablissements malgaches d'Enseignement Supérieur, qui sont barrés sur le système LMD, les Départements de Formation pédagogique ont été ainsi remplacés par des Mentions et des parcours. Et les chefs des Départements ont été ainsi remplacés par des responsables des mentions et les responsables des parcours.

Un administrateur des Réseaux et Systèmes gère le système d'information de l'Ecole et celui de l'Université.

1.4 Domaine de spécialisation

Les activités de formation, de recherche organisées à l'ENI portent sur les domaines suivants:

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale
- Modélisation informatique et mathématique des Systèmes complexes.

D'une manière plus générale, les programmes des formations sont basés sur l'informatique de gestion et sur l'informatique des Systèmes et Réseaux. Et les modules de formation intègrent aussi bien des éléments d'Informatique fondamentale que des éléments d'Informatique appliquée. Le tableau 1 décrit l'organisation du système de formation pédagogique de l'Ecole.

TABLEAU 1: ORGANISATION DU SYSTEME DE FORMATION PEDAGOGIQUE DE L'ECOLE

| Formation théorique | Formation pratique |
|--|--|
| <ul style="list-style-type: none">- Enseignement théorique- Travaux dirigés- Travaux pratiques | <ul style="list-style-type: none">- Etude des cas- Travail de réalisation- Projets / Projets tutorés- Voyages d'études- Stages |

1.5 Architecture des formations pédagogiques

Le recrutement des étudiants à l'ENI se fait uniquement par voie de concours d'envergure nationale en première année.

Les offres de formation organisées à l'Ecole ont été validées par la Commission Nationale d'Habilitation (CNH) auprès du Ministères de l'Enseignement Supérieur et de la Recherche

Scientifique selon les dispositions de l'Arrêté N°31.174/2012-MENS en date du 05 Décembre 2012.

Au sein de l'ENI, il existe une seule mention (INFORMATIQUE) et trois parcours:

- Génie Logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale

L'architecture des études à trois niveaux conformément au système Licence- Master-Doctoral (LMD) permet les comparaisons et les équivalences académiques des diplômes au niveau international.

- L = Licence (Bac + 3) = L1, L2, L3 = 6 semestres S1 à S6
- M = Master (Bac + 5) = M1, M2 = 4 semestres S7 à S10

Le diplôme de licence est obtenu en 3 années des études après Baccalauréat. Et le diplôme de Master est obtenu en 2 ans après obtenu du diplôme de LICENCE.
Le MASTER PROFESSIONNEL est un diplôme destiné à la recherche emploi au terme des études.

Le MASTER RECHERCHE est un diplôme qui remplace l'ancien Diplôme d'Etudes Approfondies (DEA), et qui permet de s'inscrire directement dans une Ecole Doctorale au terme des études.

- D = Doctorat (Bac + 8)

Le Doctorat est un diplôme qu'on peut obtenir en 3 ans après l'obtention du diplôme de MASTER RECHERCHE.

Le tableau 2 présente l'architecture des études correspondant au système LMD.

TABLEAU 2: ARCHITECTURE DES ETUDES AU SYSTEME LMD



DTS : Diplôme de Technicien Supérieur

BTS : Brevet de Technicien Supérieur

DUT : Diplôme Universitaire de Technicien

La licence peut avoir une vocation générale ou professionnelle.

Le master peut avoir une vocation professionnelle ou de recherche.

TABLEAU 3: LISTE DES FORMATIONS EXISTANTES

| | FORMATION EN | |
|-----------------------|--|--|
| | LICENCE PROFESSIONNELLE ET HYBRIDE | MASTER |
| Condition d'admission | Par voie de concours Formation Professionnelle : 100 candidats Formation hybride : 150 candidats | |
| Condition d'accès | Bac de série C, D ou Technique | Etre titulaire de licence professionnelle |
| Durée de formation | 3 années | 2 années |
| Diplôme à délivrer | Diplôme de Licence Professionnelle en Informatique | Diplôme de Master Professionnel Diplôme de Master |

L'accès en première année de MASTER se fait automatiquement pour les étudiants de l'Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s'inscrire directement dans une Ecole Doctorale. Les Ecoles Doctorales jouissent d'une autonomie de gestion par rapport aux Etablissements de formation universitaire. Il convient de signaler que par arrêté ministériel N° 21.626/2012 – MESupRES publié le 9 Août 2012 par la Commission National d'habilitation (CNH), l'Ecole Doctorale « Modélisation – Informatique » a été habilitée pour l'Université de Fianarantsoa.

Depuis l'année universitaire 2010-2011, l'ENI s'est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de l'insuffisance de la capacité d'accueil des infrastructures logistiques. En effet, le système de formation hybride semi - présentiel utilise la visioconférence pour la formation à distance.

Bien qu'il n'existe pas encore au niveau international de reconnaissance écrite et formelle des diplômes délivrés par l'ENI, les étudiants diplômés de l'Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (CANADA, Suisse, France...)

1.6 Relation de l'ENI avec les entreprises et les organismes

Les stages effectués chaque année par les étudiants mettent l'Ecole en rapport permanent avec plus de 300 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux.

L'Ecole dispose ainsi d'un réseau d'entreprises, de sociétés et d'organismes publics et privés qui sont des partenaires par l'accueil en stage de ses étudiants, et éventuellement pour le recrutement après l'obtention des diplômes par ces derniers.

Les compétences que l'Ecole cherche à développer chez ses étudiants sont l'adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l'expérimentation et l'innovation.

En effet, la vocation de l'ENI est de former des techniciens supérieurs de niveau LICENCE et des ingénieurs de type généraliste de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d'évoluer professionnellement dans des secteurs d'activité variés intégrant l'informatique.

Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l'Ecole et les besoins évolutifs du marché de l'emploi.

Les principaux débouchés professionnels des diplômés de l'Ecole concernent les domaines suivants :

- ✓ L'informatique de gestion d'entreprise
- ✓ Les technologies de l'information et de la communication (TIC)
- ✓ La sécurité informatique des réseaux
- ✓ L'administration des réseaux et des systèmes
- ✓ Les services bancaires et financiers, notamment le Mobile Banking
- ✓ Les télécommunications et la téléphonie mobile
- ✓ Les Big Data
- ✓ Le commerce, la vente et l'achat, le Marketing
- ✓ L'ingénierie informatique appliquée
- ✓ L'écologie et le développement durable

Parmi les sociétés, entreprises et organismes partenaires de l'Ecole, on peut citer :

ACCENTURE Mauritius, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie (AUF) , B2B, Banque Centrale, BFG-SG, BIANCO, BLUELINE, CNAPS, Bureau National de Gestion des Risques et des Catastrophes (BNNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, CHU, CNRIT, COLAS, Direction Générale des Douanes, DLC, DTS/Moov, FID, FTM, GNOSYS, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MEF, MEN, MESupRES, MFB, MIC, MNINTER, Min des postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SMMC, SNEDADRS Antsirabe, Sénat, Société

d'Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN'I BETSILEO, WWF ...

L'organisation de stage en entreprise continue non seulement à renforcer la professionnalisation des formations dispensées, mais elle continue surtout à accroître de façon exceptionnelle les opportunités d'embauche pour les diplômés de l'Ecole.

1.7 Partenariat au niveau international

Entre 1196 et 1999, l'ENI avait bénéficié de l'assistance technique et financière de la Mission Française de Coopération et d'action culturelle dans le cadre du Programme de Renforcement de l'Enseignement Supérieur (PRESUP) consacré à l'Ecole a notamment porté sur :

- Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques
- La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque
- L'appui à la formation des formateurs
- L'affectation à l'Ecole d'Assistants techniques français

De 2000 à 2004, l'ENI avait fait partie des membres du bureau de la Conférence Internationale des Ecoles de formation d'Ingénieurs et Technicien d'Expression Française (CITEF).

Les Enseignants-Chercheurs de l'Ecole participent régulièrement aux activités organisées dans le cadre du Colloque Africain sur la Recherche en Informatique (CARI).

L'ENI avait également signé un accord de coopération interuniversitaire avec l'Institut de Recherche en Mathématiques et Informatique Appliquées (IREMIA) de l'Université de la Réunion, l'Université de Rennes 1, l'INSA de Rennes, l'Institut National Polytechnique de Grenoble (INPG).

A partir du mois de Juillet 2001, l'ENI avait abrité le Centre de Réseau Opérationnel (Network Operating Center) du point d'accès à Internet de l'Ecole ainsi que de l'Université de Fianarantsoa. Grâce à ce projet américain qui a été financé par l'USAID Madagascar, l'ENI de l'Université de Fianarantsoa avait été dotées d'une ligne spécialisée d'accès permanent au réseau Internet.

L'ENI avait de même noué des relations de coopération avec l'Institut de Recherche pour le Développement (IRD).

L'objet du projet de coopération avait porté sur la modélisation environnementale du Corridor forestier de Fandriana jusqu'à Vondrozo (COFAV). Dans ce cadre, un atelier scientifique

international avait été organisé à l'ENI en Septembre 2008. Cet atelier scientifique avait eu pour thème de modélisation des paysages.

Et dans le cadre du programme scientifique PARRUR, l'IRD avait financé depuis 2010 le projet intitulé « Forêts, Parcs et Pauvreté dans le Sud de Madagascar (FPPSM). Des étudiants en DEA et des Doctorants issus de l'ENI avaient participé à ce Programme.

Par ailleurs, depuis toujours la même année 2010, l'ENI de Fianarantsoa avait été sélectionnée pour faire partie des organismes partenaires de l'Université de Savoie dans le cadre du projet TICEVAL relatif à la certification des compétences en TIC ;

Le projet TICEVAL avait été financé par le Fonds Francophone des Inforoutes pour la période allant de 2010 à 2012, et il avait eu pour objectif de généraliser la certification des compétences en Informatique et Internet du type C2i2e et C2imi.

Dans le cadre du projet TICEVAL, une convention de coopération avec l'Université de Savoie avait été signée par les deux parties concernées. La mise en œuvre de la Convention de Coopération avait permis d'envoyer des étudiants de l'ENI à Chambéry pour poursuivre des études supérieures en Informatique.

Enfin et non des moindres, l'ENI avait signé en Septembre 2009 un protocole de collaboration scientifique avec l'ESIROI – STIM de l'Université de la Réunion.

Comme l'ENI constitue une pépinière incubatrice de technologie de pointe, d'emplois et d'entreprises, elle peut très bien servir d'instrument efficace pour renforcer la croissance économique du pays, et pour lutter contre la Pauvreté.

De même que le statut de l'Ecole devrait permettre de renforcer la position concurrentielle de la Grande Ile sur l'orbite de la modélisation grâce au développement des nouvelles technologies.

1.8 Débouchés professionnels des diplômés

Le chômage des jeunes diplômés universitaires fait partie des maux qui gangrènent Madagascar. L'environnement socio-politique du pays depuis 2008 jusqu'à ce jour a fait que le chômage des diplômés est devenu massif par rapport aux établissements de formation supérieure existants.

Cependant, les formations proposées par l'Ecole permettent aux diplômés d'être immédiatement opérationnels sur le marché du travail avec la connaissance d'un métier complet lié à l'informatique aux TIC.

L'Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante.

Les diplômés en LICENCE et en MASTER issus de l'ENI peuvent faire carrière dans différents secteurs. L'Ecole bénéficie aujourd'hui de 34 années d'expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C'est une Ecole Supérieure de référence en matière informatique.

Par conséquent, en raison de fait que l'équipe pédagogique de l'Ecole est expérimentée, les enseignants-chercheurs et les autres formateurs de l'Ecole sont dotés d'une grande expérience dans l'enseignement et dans le milieu professionnel.

L'Ecole est fière de collaborer de façon régulière avec un nombre croissant d'entreprises, de sociétés et d'organismes publics et privés à travers les stages des étudiants. Les formations dispensées à l'Ecole sont ainsi orientées vers le besoin et les attentes des entreprises et des sociétés.

L'Ecole fournit à ses étudiants de niveau LICENCE et MASTER des compétences professionnelles et métiers indispensables pour les intégrer sur le marché du travail.

L'Ecole s'efforce de proposer à ses étudiants une double compétence à la fois technologique et managériale combinant l'informatique de gestion ainsi que l'administration des réseaux et systèmes.

D'une manière générale, les diplômés de l'ENI n'éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l'ENI recommande à ses diplômés de promouvoir l'entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d'études.

TABLEAU 4: DEBOUCHES PROFESSIONNELS EVENTUELS DES DIPLOMES

| | |
|---------|---|
| LICENCE | <ul style="list-style-type: none">- Analyste- Programmeur- Administrateur de site web/de portail web- Assistant Informatique et internet- Chef de projet web ou multimédia - Développeur Informatique ou multimédia- Intégrateur web ou web designer- Hot liner/Hébergeur Internet- Agent de référencement- Technicien/Supérieur de help desk sur Informatique- Responsable de sécurité web- Administrateur de réseau |
| MASTER | <ul style="list-style-type: none">- Administrateur de réseau et système- Architecture de système d'information- Développeur d'applications- Ingénieur réseau |

| | |
|--|---|
| | <ul style="list-style-type: none"> - Webmaster /web designer - Concepteur Réalisateur d'applications - Directeur du système de formation - Directeur de projet informatique - Chef de projet informatique - Responsable de sécurité informatique - Consultant fonctionnel ou freelance |
|--|---|

1.9 Ressources humaines

- Directeur de l'Ecole : Professeur Titulaire RAMAMONJISOA Bertin Olivier
- Responsable de Mention : RABETAFIKA Louis Haja
- Responsable de Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian
- Responsable de Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA
- Responsable de Parcours « Informatique Générale » : Monsieur Gilante GESAZAFY
- Nombre d'Enseignants permanents : 13 dont deux (02) Professeurs Titulaires, six (06) Maîtres de Conférences et cinq (05) Assistants d'Enseignement Supérieur et de Recherche
- Nombre d'Enseignants vacataires : 10

Personnel Administratif : 23

Chapitre 2 : Présentation du Pharmacie et Cabinet Vétérinaire du Boulevard

2.1 Information

La Pharmacie et Clinique Vétérinaire du Boulevard est une petite et moyenne entreprise située à Ankazomanga ANTANANARIVO, à 100m du Boulevard de l'Europe. Pour la prise de contact avec ce cabinet et pharmacie, contacter le numéro de téléphone mobile +261 33 64 369 12 et +261 34 04 374 74. On peut aussi les joindre par courrier électronique à l'adresse mail vetboulevard@yahoo.fr ou par Facebook : cabinet vétérinaire boulevard.

2.2 Historique

La Pharmacie et Clinique Vétérinaire du Boulevard, abrégée CVB, est en voie de devenir une référence dans le monde des animaux de compagnie. Née d'une initiative personnelle du Docteur RAMAHATAFANDRY Ilo Tsimok'Haja, diplômé d'Etat en Vétérinaire, branche Médecine, auprès de l'Université d'Antananarivo, ce cabinet pourrait, dans les années à venir, rivaliser avec les géants de la Grande Ile dans ce domaine.

En 2013, le Docteur Ilo s'est spécialisée dans le domaine des animaux de compagnie à l'Université Inter Etat des Sciences et Médecine Vétérinaire (IESMV) en Sénégal, avant de se lancer dans le monde professionnel.

Le 14 Avril 2014, avec un confrère, le Cabinet Vétérinaire de Boulevard d'Ankazomanga a ouvert pour la première fois ses portes au grand public. Ce jour-là, il n'y avait qu'une seule personnelle.

Début 2015, entre les fonctions de Dr Ilo à la clinique et ses fonctions auprès de la Ministère de l'Elevage, le Cabinet recrutait des stagiaires pour promouvoir ses images et offrir plus de service aux amoureux des animaux. Durant cette année, elle perfectionnait de plus en plus le Cabinet et commençait à former ses personnelles. En Avril 2015, soit un an après son ouverture, le Cabinet Vétérinaire du Boulevard étend ses domaines et devient par la suite Pharmacie et Cabinet Vétérinaire du Boulevard. Cependant, ce cabinet n'est plus géré que par Dr Ilo, son confrère ayant quitté ce PME en plein expansion.

En 2016, le Cabinet intègre la vente des alimentations pour les animaux, tout en recrutant des personnelles supplémentaire ; le Cabinet compte en tout : 2 personnelles dont un docteur-vétérinaire et un assistant de vente.

Début 2017, le Cabinet étend encore un peu plus ses domaines, en passant cette fois à la location de voiture.

L'année 2018, tout en améliorant les qualités de ses services, le Cabinet commence aussi à multiplier les services qu'il offre. Cette même année, la Pharmacie et Cabinet du Boulevard crée

sa première filiale basée à Antsirabe. Cette filiale, dans un endroit où le monde des animaux de compagnie prend de l'ampleur, commence à faire ses preuves.

En 2019, les personnelles passe à 3 docteurs, 2 assistants de vente et 1 autre personnel pour le CVB d'Ankazomanga.

2.3 Organigramme de la Pharmacie et CVB

La figure 2 représente l'organigramme actuelle de la Pharmacie et CVB.

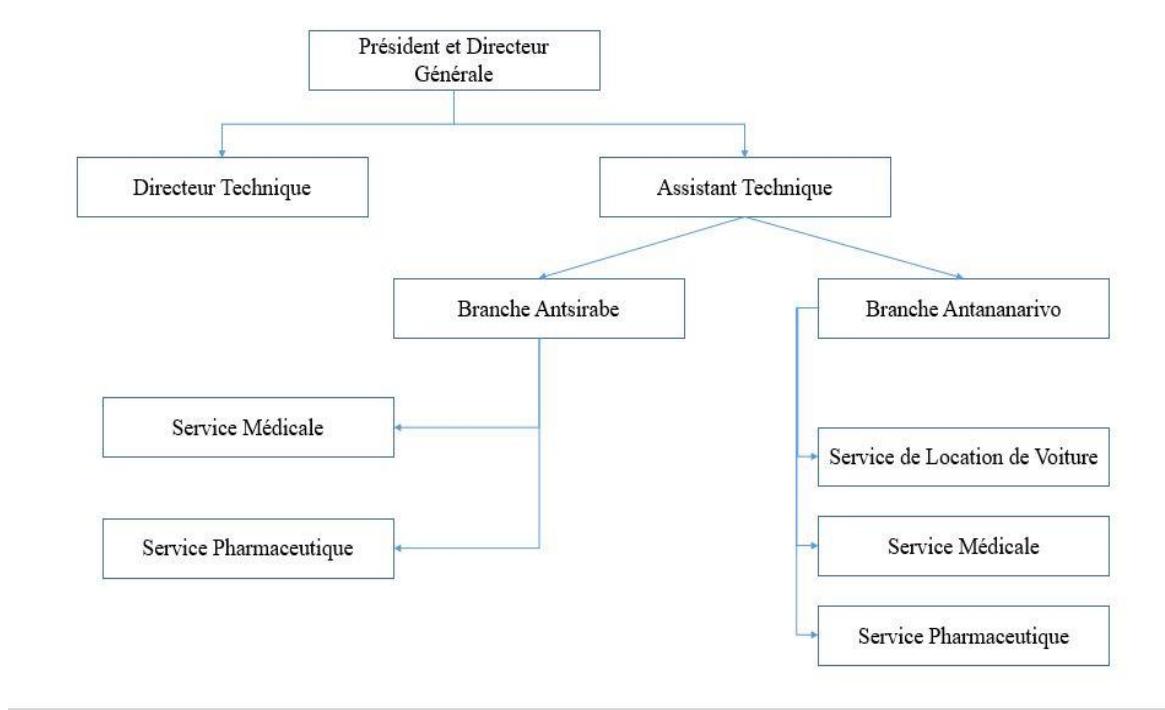


FIGURE 2: ORGANIGRAMME DE LA PHARMACIE ET CABINET VETERINAIRE DU BOULEVARD

2.4 Missions et buts

La Pharmacie et Cabinet Vétérinaire du Boulevard investit beaucoup dans l'expansion et la promotion du monde des animaux. Elle a pour mission de :

- Promouvoir le monde des animaux de compagnie ;
- Développer les qualités de services au meilleur prix et accessible à tous ;
- Répondre aux besoins de ses clients, surtout les éleveurs ;
- Offrir des soins et traitements pour les animaux ;

2.5 Soins offertes

La Pharmacie et CVB offrent des soins de très hautes qualités, comme :

- Vaccination
- Consultation
- Hospitalisation

- Alimentation
- Animalerie
- Consultation à domicile

2.6 Partenaires et fournisseurs

La figure 3 montre les logos des fournisseurs de la Pharmacie et Cabinet Vétérinaire du Boulevard.



FIGURE 3: LOGOS DES FOURNISSEURS ET PARTENAIRES DU CVB

Chapitre 3 : Description du projet

3.1 Formulation du projet

La Pharmacie et Cabinet Vétérinaire Boulevard Ankazomanga possède 3 services aux sein de son cabinet. Le service qui nous intéresse est le service médical qui s'occupe des traitements des animaux qui passe aux cabinet ainsi que la gestion de ses propres stocks.

La gestion des patients ainsi que la gestion des stocks sont occupés par les vétérinaires traitants au sein même de la clinique. La gestion monétaire est faite par ces mêmes docteurs, par exemple lors des mises à jour des restes à payer des clients, le paiement des commandes auprès des fournisseurs.

Le traitement de ces informations est un travail considérable car non seulement ils doivent parcourir un grand nombre de données avant de pouvoir rectifier certains paiements par exemple, mais aussi, ils doivent se référer à des données qui ne sont pas mises à jours.

Notre tâche a été alors de concevoir un nouvel outil de gestion complète du cabinet, qui permet à la fois d'avoir un suivi sur les patients, le stock ainsi que les échanges au sein de celle-ci.

3.2 Objectif et besoin de l'utilisateur

L'objectif principal de ce projet consiste avant tout de réorganiser cette masse de données du cabinet en une base de données répondant à leurs attentes ainsi que de concevoir et réaliser un outil de gestion du cabinet, d'un point financier et d'un point médical. Cet outil va alors régler les problèmes liés à :

- L'enregistrement des informations concernant un patient particulier
- L'historique des soins pour un patient particulier
- La gestion des flux monétaires au sein du cabinet
- La répartition par jour des soins pour pouvoir établir le jour de repos à attribuer à un vétérinaire
- Assurer l'historique de tout ce qui se passe au clinique

3.3 Moyens nécessaires à la réalisation du projet

3.3.1 Moyens humains

Pour la réalisation de ce projet, les moyens humains nécessaires sont listés dans le tableau 5.

| POSTE AU SEIN DU CABINET | NOMBRE | ROLE DANS LE PROJET |
|--------------------------|--------|--|
| Propriétaire de CVB | 1 | Encadreur professionnel et futur utilisateur |

| | | |
|--|---|-------------------------------------|
| Stagiaire étudiant en 2 ^{ème} année de la formation hybride en informatique à l'ENI | 1 | Concepteur et réalisateur du projet |
| Vétérinaire | 2 | Futurs utilisateurs |
| Professeur de l'ENI | 1 | Encadreur pédagogique |

TABLEAU 5: MOYENS HUMAINS POUR LA REALISATION DE L'APPLICATION

3.3.2 Moyens matériels

Dans un système en réseaux local avec une architecture Client-Serveur, on va considérer les deux côtés :

- a) Côté serveur :

Le tableau 6 montre les caractéristiques du serveur au sein de CVB.

TABLEAU 6: SERVEUR DE CVB

| Elément | Système | Mémoire | Disque dur |
|--|-----------|---------|------------|
| Ordinateur de bureau ; Fabriquant : ASUS | Windows 7 | 2Go | 250Gb |

- b) Côté client :

Côté client, le tableau 7 montre les caractéristiques des matériaux :

TABLEAU 7: MATERIELS COTES CLIENTS POUR LE PROJET

| Elément | Système | Mémoire | Disque dur |
|---|------------|------------|------------|
| Ordinateur portable ; Fabriquant : ASUS Modèle : X541NA Processeur : Intel Celeron CPU | Windows 10 | 4096Mb RAM | 500Go |

| | | | |
|----------------------------|--|--|--|
| N3350 @ 1.1Ghz (2 CPUs) | | | |
|----------------------------|--|--|--|

Unité :

Go : Gigaoctet

GHz : Gigahertz

Mb : Mégabit

3.3.3 Logiciels utilisés pour la mise en œuvre du projet

Le tableau 8 décrit les logiciels utilisés pour pouvoir réaliser ce projet.

TABLEAU 8: LOGICIELS UTILISES POUR LA REALISATION DU PROJET

| DESIGNATION | VERSION | UTILITE DANS LE PROJET |
|------------------------|---|--|
| Laragon Lite | 4.0.15 | Gestion et manipulation de la base de données durant le développement de l'application |
| Wampp Server | 2.5 | Gestion de la base de données en production |
| Visual Studio Code | 1.41 | Environnement de développement de l'application |
| Microsoft Office Excel | 2016 | Consultation du dictionnaire de données, du glossaire et de la nomenclature |
| Microsoft Office Word | 2016 | Rédaction du rapport de stage |
| Mozilla Firefox | 70.0.1 | Navigateur web des utilisateurs |
| Chrome | 79.0.3945.130 (Build officiel) (64 bits) | Navigateur web de test en mode développement |
| MySQL Workbench | 8 | Modélisation de la base de données et génération de la base de données |

3.4 Méthodologie pour la conduite du projet

Pour la réussite d'un projet, celle-ci doit se baser avant tout sur une planification bien régie parce que cela deviendra le facteur de l'aboutissement du travail. Cette démarche permet d'atteindre nos objectifs qui est de fournir une application performante et facile à comprendre et à apprendre, ainsi que la facilité de mettre à jour cette application même ultérieurement, le tout en mettant les données dans un espace de stockage sécurisé et fiable.

La figure 4 montre les démarches suivies pour la conduite du projet :

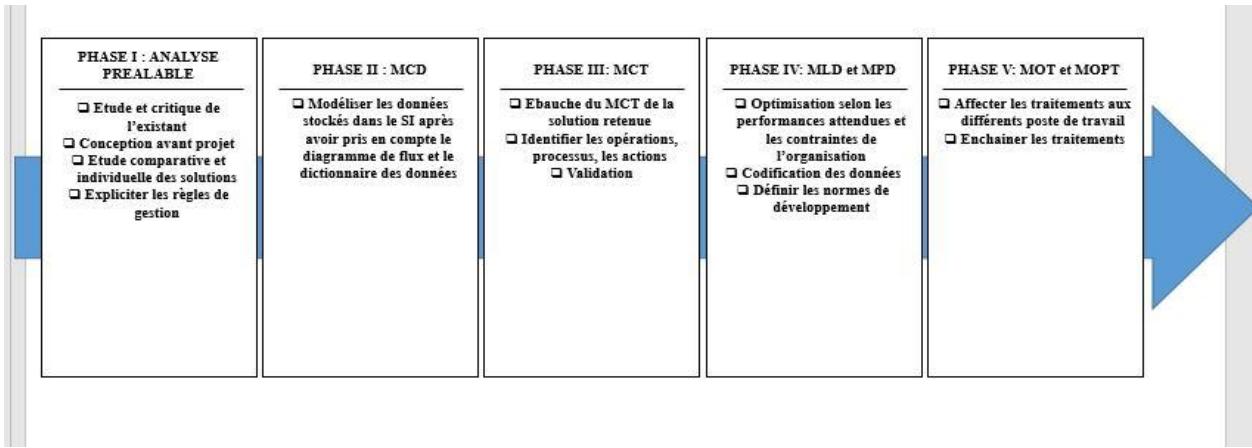


FIGURE 4: METHODOLOGIE DE LA CONDUITE DU PROJET

3.5 Résultat attendu

Un outil de gestion de cabinet vétérinaire qui s'occupera :

- Répertorier tous les clients du cabinet avec un système de profil
- Enregistrer les patients et leurs informations avec un système de profil
- Gérer l'état de stock des médicaments
- Afficher et notifier les médicaments en rupture de stock ou/et périmés
- Afficher l'historique des traitements reçus par un patient dans son profil
- Notifier l'utilisateur lors des échanges financiers incomplets (date limite de paiement côté fournisseur, paiement incomplet pour les clients...)
- Gérer la finance du cabinet

PARTIE II : ANALYSE ET CONCEPTION

Chapitre 4 : Analyse préalable

4.1 Analyse de l'existant

4.1.1 Organisation actuelle

Actuellement, les vétérinaires traitants de CVB enregistrent leurs données de stock sur une feuille de calque Excel, dont elles sont reparties par catégorie (appelé aussi traitement). Les fiches patientes sont quant à elles, enregistrés manuellement sur le carnet du patient. Les vétérinaires traitants, qui sont aussi les employés en charge du service financière s'occupent de la gestion des factures avec des données écrites, qu'on range ensuite un peu partout.

Le tableau 9 montre les moyens personnels impliqués dans la gestion de CVB :

TABLEAU 9: LES MOYENS PERSONNELS DANS LA GESTION DU CABINET

| Nombre | Rôle | Tâches |
|--------|----------------------|---|
| 2 | Vétérinaire traitant | Responsable des traitements des animaux |
| 1 | PDG | Chef hiérarchique dans le Cabinet |

Côté matériel, un ordinateur complet, représentées dans le tableau 10, est à la disposition des services médicaux et financiers.

TABLEAU 10: LES MOYENS MATERIELS

| Composants | Caractéristiques |
|------------------------|------------------|
| Constructeur | ASUS |
| CPU | Core2Duo |
| RAM | 2Gb |
| Disque dur | 160Go |
| Système d'exploitation | Windows7 |

Unité :

Go : Gigaoctet

Gb : Gigabit

4.1.2 Inventaire des moyens logiciels

Au niveau logistique, les services financières et médicaux travaillent sur des feuilles de calcul Excel pour gérer les stocks ainsi que les informations concernant un patient après qu'on lui ait attribué un soin. Ce logiciel est la façon de stockage de toutes les données concernant la clinique.

4.2 Critique de l'existant

D'après l'analyse, on a pu constater que les responsables chargés des traitements éprouvent beaucoup de difficulté au niveau de la gestion des informations des patients, ces derniers étant stockés sur des fichiers Excel. L'emploi du temps des patients, l'historique des consultations pour chaque patient n'étant pas bien réparti, il y a des difficultés pour le suivi des patients.

Les données étant encore manuellement effectués, on rencontre des problèmes concernant :

- Lenteur des traitements : comme tous les traitements sont effectués manuellement, leur accomplissement met beaucoup de temps à se faire, mettant ainsi en cause le bon fonctionnement du Service Financier.
- Difficulté à mettre à jour certains informations, par exemple la mise à jour des stocks
- Risque des pertes financières puisqu'on a pas le plein contrôle sur les informations, et parfois on ne peut pas déterminer à l'avance quels médicaments sont sur le point d'être périmé.

4.3 Conception d'avant-projet

4.3.1 Mise en avant du problème principal

Due à ce système d'information qu'on peut qualifier de risqué et très désordonné, des erreurs côté logistique peuvent subsister dans la gestion du cabinet. Les vétérinaires traitants auront du mal avec le traitement ainsi que l'établissement de la facture qui doit aussi passer par différentes paramètres et calculs. Ces failles engendrent bien évidemment un risque élevé de défaillance du système actuel, un risque auquel est exposé toutes les données concernant les flux d'échanges du cabinet.

4.3.2 Scenario des solutions

Pour optimiser la gestion du cabinet, on a proposé quelques solutions que l'utilisateur peut exploiter :

- **Solution 1 :** Utilisation des applications existants sur internet, comme [Vetup¹](#), qui sont des logiciels onlines.
- **Solution 2 :** Création et réalisation d'une application répondant aux besoins des utilisateurs, c'est-à-dire une application sur mesure

Le tableau 11 présente la comparaison des deux solutions éventuelles.

TABLEAU 11: LES SOLUTIONS PROPOSEES

| SOLUTION | INCONVENIENTS | AVANTAGES |
|---|---|---|
| Utilisation d'une application déjà existante | <ul style="list-style-type: none"> - Coût trop élevé puisqu'il faudrait une connexion internet dans l'enceinte du cabinet - Structure des données qui ne sera toujours pas en conformité avec la structure des données du cabinet donc une possibilité de perte des certains informations importants pour le cabinet. | <ul style="list-style-type: none"> - Gain de temps, l'application est disponible de suite avec une base de données incluse - IHM bonne et agréable, - Comme l'application est sur un serveur distant, elle est donc plus légère, c'est-à-dire avec un temps de réponse optimal |
| Création et réalisation d'une nouvelle application | <ul style="list-style-type: none"> - Nécessite un certain temps pour la conception et la réalisation de l'application. - Apprentissage un peu délicate pour les utilisateurs | <ul style="list-style-type: none"> - Faible coût d'investissement, - Facilité de maintenance pour les techniciens, car on peut accéder aux codes sources, - L'adaptation à l'application serait facile lors de son utilisation, |

4.3.3 Ebauche de solution et justification du choix du moyen

Après une étude approfondie des solutions, on réalise que la solution n° 2, qui sera la conception et la réalisation d'une application, serait la plus envisageable pour pallier aux failles. De cette manière, on répond exactement aux besoins des utilisateurs en assurant l'obtention d'une base de données plus structurée et bien ordonnée. Ce n'est qu'ainsi que les informations pourront être visualisées facilement, et c'est également le moyen le plus rapide et moins fastidieux.

4.3.4 Justification du choix des outils

Les principaux outils sont les suivants :

- Framework : CodeIgniter 3, Bootstrap 4

- SGBD : MySQL
- Outil de développement : Visual Studio Code
- Laragon Lite

a) Choix du Framework : Codeigniter 3

Dans l'ombre des géants de [Framework PHP²](#) comme Symphony, Laravel ou encore Phalcon, [CodeIgniter](#) [1] est sans doute l'un des Framework les plus faciles à apprendre. Très rapide et gère bien des projets un peu moins complexes que ceux des entreprises, CodeIgniter embarque déjà avec lui une structure MVC qui est très utile pour alléger notre application. Avec un système de gestion de base de données facile à mettre en place et ses systèmes de requêtes plus proches du SQL lui-même, ce Framework offre alors un avantage considérable pour ceux qui ne sont pas très à l'aise avec l'ORM de certains Framework comme celui de Laravel.

Des systèmes de librairies ou encore des systèmes d'Helpers sont là pour nous permettre d'interagir avec certains code source que les autres utilisent aussi.

b) Choix du SGBD : MySQL

A la place de MySQL, on aurait pu choisir MS Access ou Microsoft SQL Server pour gérer les bases de données. Mais les faits qui montrent que MySQL présente différents atouts, nous ont convaincu de le choisir.

MySQL est un système de gestion de base de données relationnelle (SGBDR) gratuit.

✓ Côté sécurité :

MySQL dispose d'un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.

✓ Portabilité :

MySQL fonctionne sur des systèmes d'exploitation différents, incluant Windows, Unix, Mac

✓ Mobilité/continuité :

L'application est accessible par tous les postes d'un réseau LAN et même WAN. Un incident technique sur un poste n'empêchera pas la personne de basculer immédiatement à une autre unité tout en étant opérationnel dans les minutes qui suivent. De même s'il est hors bureau ou sur un autre poste que le sien. Cela n'est pas possible en utilisant Access.

c) Choix de l'outil de développement : Visual Studio Code

Même si Dreamweaver fut l'un des premiers éditeurs HTML qui a intégré un gestionnaire de site, qui depuis la version MX, peut être utilisé avec des langages web dynamiques (ASP, PHP), nous avons eu un peu du mal à choisir entre PHPStorm qui est plutôt un IDE PHP et Visual Studio Code qui est lui multi langage. Cependant, Visual Studio Code s'avère être un choix intéressant car non seulement il est plus qu'un éditeur de code classique, c'est aussi un véritable IDE. Avec

des extensions plus ou moins complexes, VS Code devient un outil très puissant qui est de bon augure pour la mise en place de l'architecture de notre application. Un des extensions le plus populaires sur VS Code est par exemple Emmet.

d) Choix du langage de développement : PHP

A la place de PHP, on aurait pu choisir un autre langage pour développer l'application. Mais les faits qui montrent que PHP présente différents atouts, a convaincu de le choisir. PHP est un langage de script utilisé le plus souvent du côté serveur, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendues par un navigateur. La syntaxe du langage provient de celles du langage C, de Perl et de Java.

PHP offre notamment des atouts plus que considérable :

- La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL)
- La simplicité d'écriture de scripts ;
- La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGI, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML) ;
- La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés parmi lesquels : MySQL, Oracle, PostgreSQL ..., mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plateformes : Unix, Linux, Windows, Mac OS X, Solaris, etc....) ;
- La mise à jour affectant la structure de tables, formulaires, fonctionnalités ne nécessite pas de redéploiement, puisque c'est fait côté serveur. Contrairement à Access où il faudra recompiler l'applicatif et le redéployer sur tous les postes quel que soit la mise à jour.
- L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.) est un avantage.

e) L'outil pour la gestion de la base de données en développement : Laragon Lite

Peu connu du grand public que ses concurrents qui sont WAMPP et XAMPP, Laragon est aujourd'hui un des principaux logiciels pour la mise en place d'un serveur Apache avec un monté en charge très rapide. Le seul bémol pour ce logiciel est qu'il n'est disponible que pour Windows mais à part ça, il est très puissant. Finis la gestion de la base comme dans PhpMyAdmin de WAMPP en ouvrant un onglet sur le navigateur, Laragon permet de directement les données dans un onglet au sein même du logiciel.

Un des avantages le plus marquant pour Laragon est aussi sa capacité à directement créer des maquettes de site ou d'application déjà faite pour nous, comme des applications Laravel, Symphony, WordPress ou même Drupal. On gagne beaucoup de temps sur cet outil puisqu'il va même jusqu'à créer directement des hôtes virtuels pour nous et aussi créer notre base de données en fonction de cette nom d'hôte.

f) Outil de modélisation d'une base de données : MysqlWorkbench

MysqlWorkbench est un outil complet pour la conception et modélisation d'une base de données avec une possibilité d'exportation en SQL. Cet outil offre aussi une gestion des relations entre table et permet de modéliser les cardinalités utilisées dans ces relations.

Chapitre 5 : Analyse conceptuelle

5.1 Présentation de la méthode MERISE

La mise en place d'un système d'information n'est pas évidente car il faudrait avant tout réfléchir à l'ensemble des organisations concernées ainsi qu'au service auquel on fait l'étude.

La méthode **MERISE** [2], une des méthodes qui offre une flexibilité, est basé sur la séparation entre les données manipulées et les actions à entrevoir par le nouveau système. Cela offre une longévité au modèle et la production des résultats satisfaisants. En effet, les données en elles-mêmes ne seront pas souvent remaniés tandis que les traitements le sont.

Pour assurer la cohérence du système d'information, MERISE propose une démarche d'informatisation qui comprend plusieurs étapes :

- **Le schéma directeur** : qui a pour but, d'une manière globale, de définir la politique d'organisation et d'automatisation du système d'information. Le concept est de répertorier l'ensemble des applications informatiques déjà existantes à modifier et à développer. Cette étape est très importante pour la suite de l'informatisation car elle ouvre déjà une perspective de critique sur l'existant actuel.
- **L'étude préalable par domaine** : dont le but est d'aboutir à une présentation globale de notre futur système de gestion dont les modèles de données et des traitements, en mettant en lumière les principales innovations vis-à-vis du système actuellement en place –moyens matériels, moyens humains, coûts et avantages.

Cette étude est réalisée en 4 phases :

- Une **phase de recueil**, qui, par des échanges entre les acteurs actuels du système, d'analyser et de cerner les dysfonctionnements les plus frappants du système
- Une **phase de conception** : formaliser et hiérarchiser les orientations nouvelles ainsi issue des critiques sur l'existant mais aussi de prendre en considération les politiques et objectifs de la Direction Générale. Cela va donc dans le sens de modéliser le futur système avec déjà une vue pertinente de l'ensemble.
- Une **phase d'organisation** : qui est d'organiser le système actuel en fonction des organisations en place
- Une **phase d'appréciation** : établir les coûts et les délais des solutions définis pour pouvoir organiser la mise en œuvre la réalisation. Pour la réaliser, découper l'organisation en projets s'avère être nécessaire.
- L'**étude détaillé par projet** : qui est donc d'affiner les solutions conçues lors de l'étude préalable, rédiger un dossier de spécification détaillé ainsi qu'élaborer les algorithmes aux règles de gestion.

- La **réalisation** : à la suite d'un jeu d'essais approuvées par les utilisateurs, l'objectif est donc d'obtenir un programme fonctionnel
- La **mise en œuvre** : qui va transférer à l'utilisateur la responsabilité complète sur le produit, qui consiste à les former sur les modes d'emplois du programme.
- La **maintenance** : qui est une phase après la mise en production et qui consiste à faire évoluer l'application avec des nouvelles mises à jours en fonction des besoins de l'utilisateur ainsi que des progrès technologiques.

Le concept de MERISE se base sur la validation une à une des étapes tout en prenant compte des résultats de l'étape précédente et de vérifier que données et traitements se concordent correctement, s'assurer aussi que les données qu'on a besoin lors des traitements ne seront pas superflues.

La figure 6 représente la démarche globale de la méthode MERISE qui se détaille en 3 cycles de préoccupations différentes : le cycle d'abstraction, le cycle de décision et le cycle de vie.

Le cycle d'abstraction préconise 3 niveaux :

- * **Le niveau conceptuel** : décrit l'aspect statique et dynamique du système à mettre en place, mais ne se préoccupe que du niveau gestionnaire à travers les MCD et MCT.
- * **Les niveaux logiques et organisationnel** : où l'on parle des choix des ressources à utiliser pour supporter le système, matérielles/logicielles ou bien humaines, à effectuer avec le MLD et MOT
- * **Les niveaux physiques et opérationnels** qui sont plus côté terrains et dans lesquels on choisit les techniques de l'implémentation du système d'information, basés sur le MPD et MOPT.

5.2 Dictionnaire des données

Le dictionnaire des données est la ressource la plus importantes, nécessaires au système d'information de l'organisation, qui doivent être sans redondances ni synonymes.

Après une étude détaillée sur les documents au sein du cabinet, on a pu en extraire des données élémentaires qui ne peuvent plus être décomposés.

Le tableau 12 représente le dictionnaire des données concernant la gestion du cabinet.

TABLEAU 12: DICTIONNAIRE DE DONNEES

| Nom de la rubrique | Description de la rubrique | Structure | | Observation |
|---------------------------|---|------------------|-----------------|--|
| | | Type | Longueur | |
| NomPatient | Nom qui identifie un patient | AN | 60 | |
| Age | L'âge actuel du patient | N | 3 | En mois |
| DateNais | La date de naissance du patient | DATE | | jj-mm-aaaa (A laisser vide si le propriétaire ne s'en souvient pas) |
| Espèce | L'espèce auquel appartient le patient | A | 60 | |
| Sexe | Sexe du patient | AN | 60 | On a que 4 choix : mâle castré ou non castré, femelle stérilisé ou non stérilisé |
| Race | La race auquelle appartient le patient | A | 100 | |
| Couleur | La couleur du patient | A | 50 | |
| Corpulence | La forme du patient | A | 50 | On a que 4 choix : Géant, Moyen, Nain, Large |
| Description | Une description destinée au vétérinaire traitant | | | |
| createdAt | La date d'ajout du patient au sein du cabinet | DATE | | C'est la date auquelle est enregistré ce patient |
| nomProprio | Le nom complet ou le prénom du propriétaire de l'animal | AN | 255 | |
| contactProprio | Le numéro téléphone du propriétaire | AN | 255 | Avec plusieurs numéro, on les sépare avec un / |
| emailProprio | L'adresse mail d'un propriétaire | AN | 255 | |
| Statu | C'est le statut du client au sein du cabinet | A | 20 | On a deux choix : remisé, non-remisé |
| Organisation | L'organisation œuvrant pour les | AN | 80 | Le sigle si possible |

| | | | | |
|------------------|--|------|-----|---|
| | animaux auquelle appartient ce patient | | | |
| adresseProprio | L'adresse actuel du propriétaire | AN | 255 | Lot ou seulement ville |
| nomFrs | Nom du fournisseur d'article | AN | 255 | |
| responsable | Le responsable avec lequel on communique avec le fournisseur | AN | 255 | |
| contactFrs | Numéro téléphone du fournisseur | AN | 255 | |
| adresseFrs | L'adresse actuel du fournisseur | AN | 255 | |
| emailFrs | Adresse mail du fournisseur | AN | 255 | |
| rubrique | Rubrique pour une activité dans le clinique | AN | 255 | |
| prix | Prix de la rubrique | N | 11 | |
| type | Type de cette rubrique pour pouvoir la différencier | A | 20 | |
| descriptionSoin | Description de cette activité | TEXT | | |
| codeEspece | Code de l'espèce concerné pour l'activité | N | 11 | |
| numeroFactureCli | Numéro de la facture pour le client | N | 11 | |
| DateFact | Date de rédaction du facture | DATE | | |
| payeeCli | Somme déjà payée par un client sur une facture | N | | |
| remise | Remise attribué à un client | N | | En pourcentage, cette remise est attribué pour chaque facture |
| net | Net à payer par le client | | | |
| numeroFactureFrs | Numéro de la facture pour le fournisseur | N | 11 | |
| dateFactFrs | Date de rédaction du des dépenses | DATE | | |

| | | | | |
|--------------|---|-------------------------|-----|--------------------------------|
| payee | Somme déjà payée pour un fournisseur sur une dépense | N | | |
| dateEcheance | Date d'échéance pour le payement du net à payer | DATE | | |
| net | Net à payer par le client | | | |
| Suspicion | Suspicion sur le motif de consultation | AN | 255 | |
| Anamnèse | C'est l'historique détaillé sur ce motif de consultation, dans lequel on raconte tout | TEXT | | |
| codeCons | Code de la consultation | Code de la consultation | | |
| | | | | |
| codeParam | Code de paramètre du patient | N | | |
| examCompl | Examen complémentaire diagnostiqué sur le patient | AN | 255 | |
| decision | Décision prise pour cette consultation | A | 20 | |
| codeParam | Code de paramètre lié au patient | N | 11 | Composé de plusieurs paramètre |
| TRC | | A | 20 | <2s ou >= à 2s |
| FC | Fréquence cardiaque | N | 5 | Par minute |
| FreqResp | Fréquence respiratoire | N | | Par minute |
| Temperature | Température du patient | N | 3 | En Celsius |
| taille | Taille du patient sous la barre | N | 3 | En cm |
| poids | Poids du patient | N | 4 | En kg |
| traitement | Le type de traitement attribué au patient | AN | 255 | |
| codeMed | Code attribué à ce médicament | N | 255 | |
| libelleMed | Libellé du médicament | AN | 255 | |
| Unite | Unité de sortie du médicament | AN | 50 | |

| | | | | |
|------------------|---|------|----|------------|
| puDetail | Prix unitaire du médicament | N | 11 | En Ariary |
| prixPresentation | Prix d'achat par présentation du médicament | N | 11 | En Ariary |
| Presentation | Présentation du médicament au sein du cabinet | AN | 50 | |
| parPresentation | Effectif de l'unité dans la présentation | N | 6 | |
| stock | Stock du médicament actuellement | N | 6 | |
| datePeremption | Date de péremption du médicament | DATE | | jj-mm-aaaa |
| presentationGros | La présentation en gros du médicament | AN | 50 | |
| dateDebutHosp | Date de début d'hospitalisation pour un patient | DATE | | |
| dateFinHosp | Date de fin de l'hospitalisation | DATE | | |
| dateRappel | Date de rappel pour un traitement particulier | DATE | | |
| numCom | Numéro de commande auprès d'un fournisseur | N | | |

Types des données :

-**A** (Alphabétique) : Données uniquement composée de caractères alphabétiques (de ‘A’ à ‘Z’ et de ‘a’ à ‘z’)

-**AN** (Alphanumérique) : Données pouvant être composés à la fois de caractère alphabétiques et numériques.

-**N** (Numérique) : Données composées uniquement de nombres (entiers ou réels).

-**DATE** : Données de type date (Au format AAAA-MM- JJ).

5.3 Règles des gestions

Les règles de gestion sont là pour relever les contraintes que la représentation du système actuel devrait respectés. Ces règles peuvent être la forme, la valeur admissible ou possible des données.

- RG1 : dans les décisions prises, on n'a que 3 possibilités : hospitalisé avec traitement, traité seulement, traité et rappelé
- RG2 : une personne n'est enregistrée en tant que client que lorsqu'elle passe avec un patient au cabinet
- RG3 : tous les prix sont en Ariary
- RG4 : lors de l'établissement d'une facture, les quantités sont majorés.
- RG5 : les dates de rappel sont notifiées une journée avant sa date
- RG6 : la passage d'un patient au clinique est soit : pour les soins, soit pour les traitements soit pour les vaccinations
- RG7 : les médicaments utilisés en soin sont déjà inclus dans le tarif du soin et ne s'affichent plus dans cette facture
- RG8 : les tarifs doivent être : soit pour vaccination, soit pour traitement soit pour soin

5.4 Diagramme des flux

Les flux d'échanges d'information dans une organisation sont rédigés en tant que diagramme de flux pour la Méthode MERISE.

La figure 5 représente le diagramme de flux au sein du CVB :

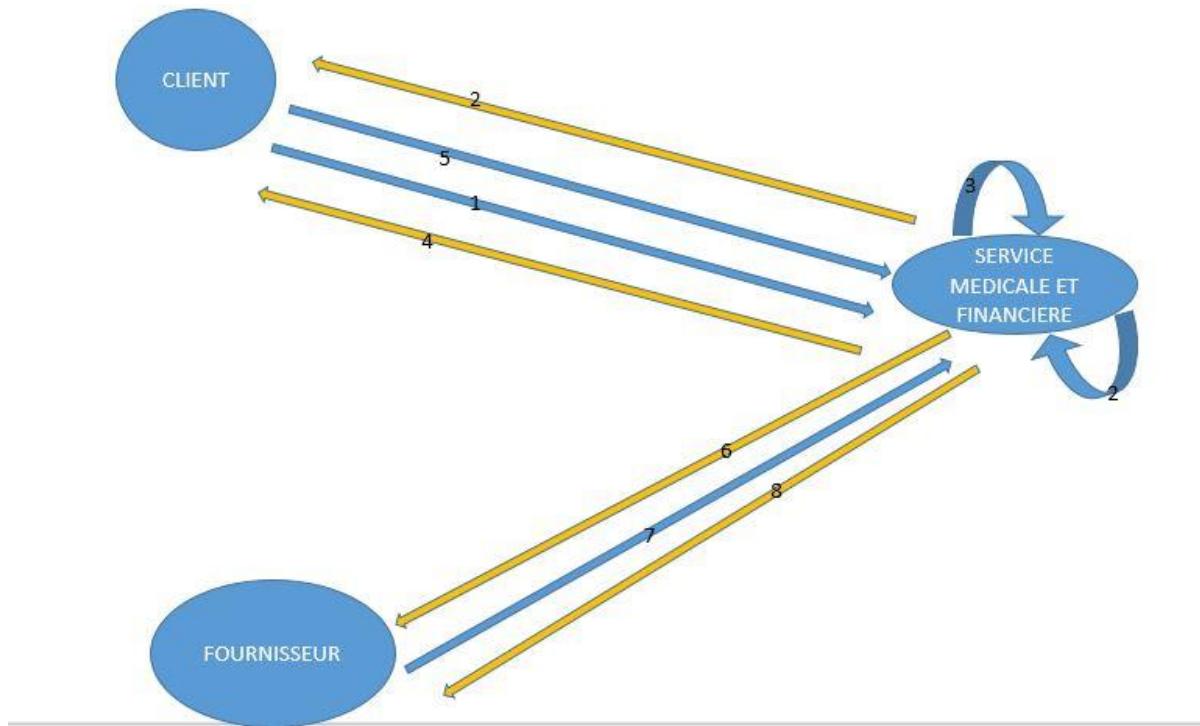


FIGURE 5: DIAGRAMME DE FLUX

- 1: Présentation du client et son animal pour un passage au clinique
- 2 : Notification du client sur des possibles examens complémentaires
- 2 : Traitement du patient avec ses motifs de visite
- 3 : Etablissement du facture
- 4 : Facturation du client
- 5 : Payement de la facture en entier ou une part
- 6 : Demande d'approvisionnement
- 7 : Envoi des commandes avec un possible date d'échéance de payement
- 8 : Payement de la dépense en entier ou avec des restes

5.5 Modélisation des données et des traitements

5.5.1 Le Modèle Conceptuel des Données

Le MCD constitue la description globale des données manipulées. C'est un aspect des données qui ne se tient pas compte des techniques ni de l'organisation sur tel ou tel traitement sur le support de ces données.

Formalisme :

La figure 6 représente le Modèle Conceptuel des Données liées à la gestion de la clinique

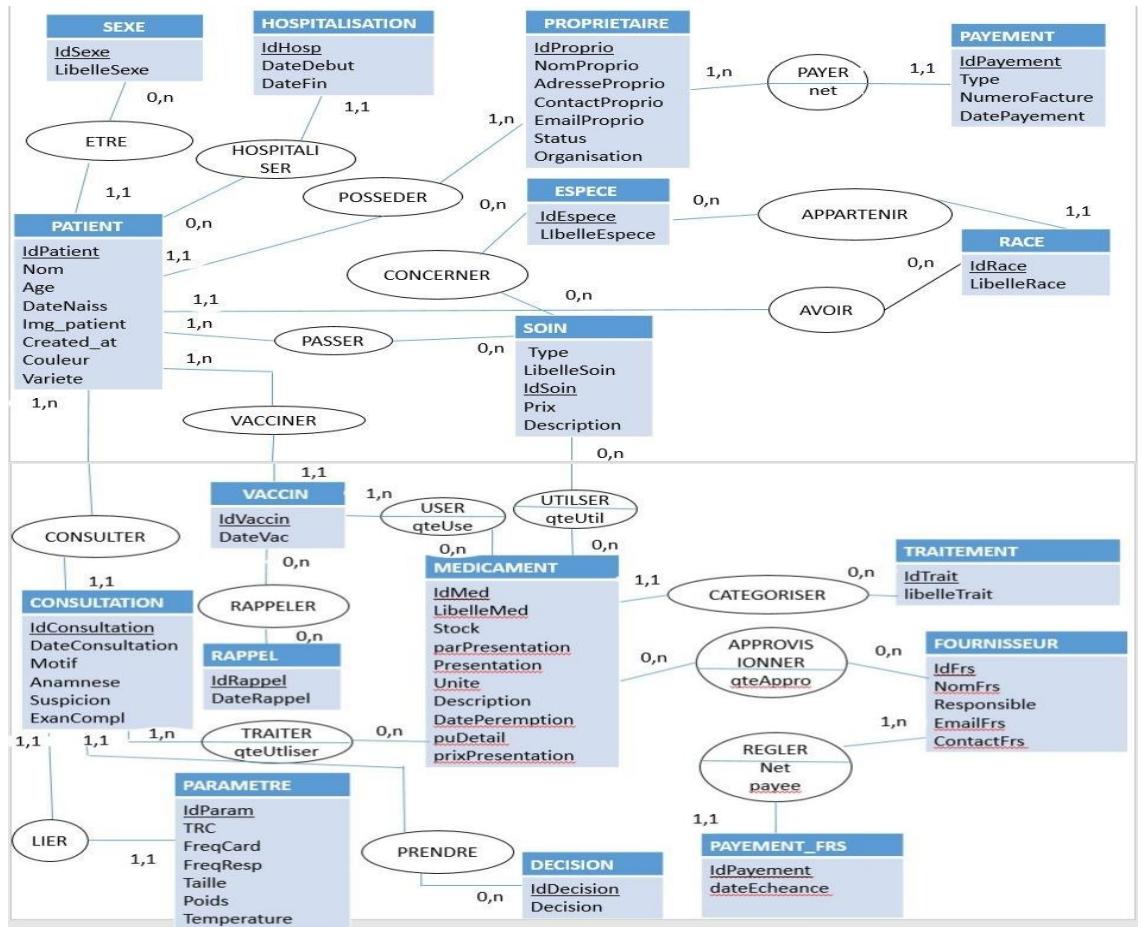


FIGURE 6: MODELE CONCEPTUEL DE DONNEES

5.5.2 Le Modèle Logique des Données

Le Modèle Logique des Données est la transformation du MCD en tant qu'objet pour une base de données. Il y a plusieurs règles à suivre sur ce [passage](#) [3]:

- ❖ Pour les entités :

- Toute entité se transforme en table
- L'identifiant de l'entité devient sa clé primaire
- Ses propriétés deviennent ses attributs

❖ Pour les associations :

- **Associations père (0,n ou 1,n) – fils (0,1 ou 1,1)**
 - « L'entité père » devient une table.
 - « L'entité fils » devient une table.
 - Les propriétés éventuelles de l'association deviennent des attributs pour la table fils.
 - L'identifiant de la table père devient la clé étrangère dans la table fils, c'est-à-dire un identifiant d'une autre table qui est un attribut d'une table.
- **Cardinalité des entités associés 0,n ou 1,n**
 - Chaque entité concernée se transforme en table, l'identifiant de cette entité devient la clé primaire de la table.
 - L'association devient une table dont son identifiant serait sa clé primaire.
 - Les propriétés éventuelles de l'association passent pour attributs de la table « fils ».
- **Associations dont la cardinalité est réflexive**
 - Dans le cas où une association lie des occurrences d'une même entité.
- **Associations dont les identifications sont relatives**
C'est dans les cas où une occurrence d'une entité n'a pas d'existence propre par rapport à une entité. Donc cette occurrence dépend de l'entité principale, alors l'identifiant de l'entité principale s'ajoute à celui de l'entité dépendante.

La figure 7 représente le Modèle Logique des Données.

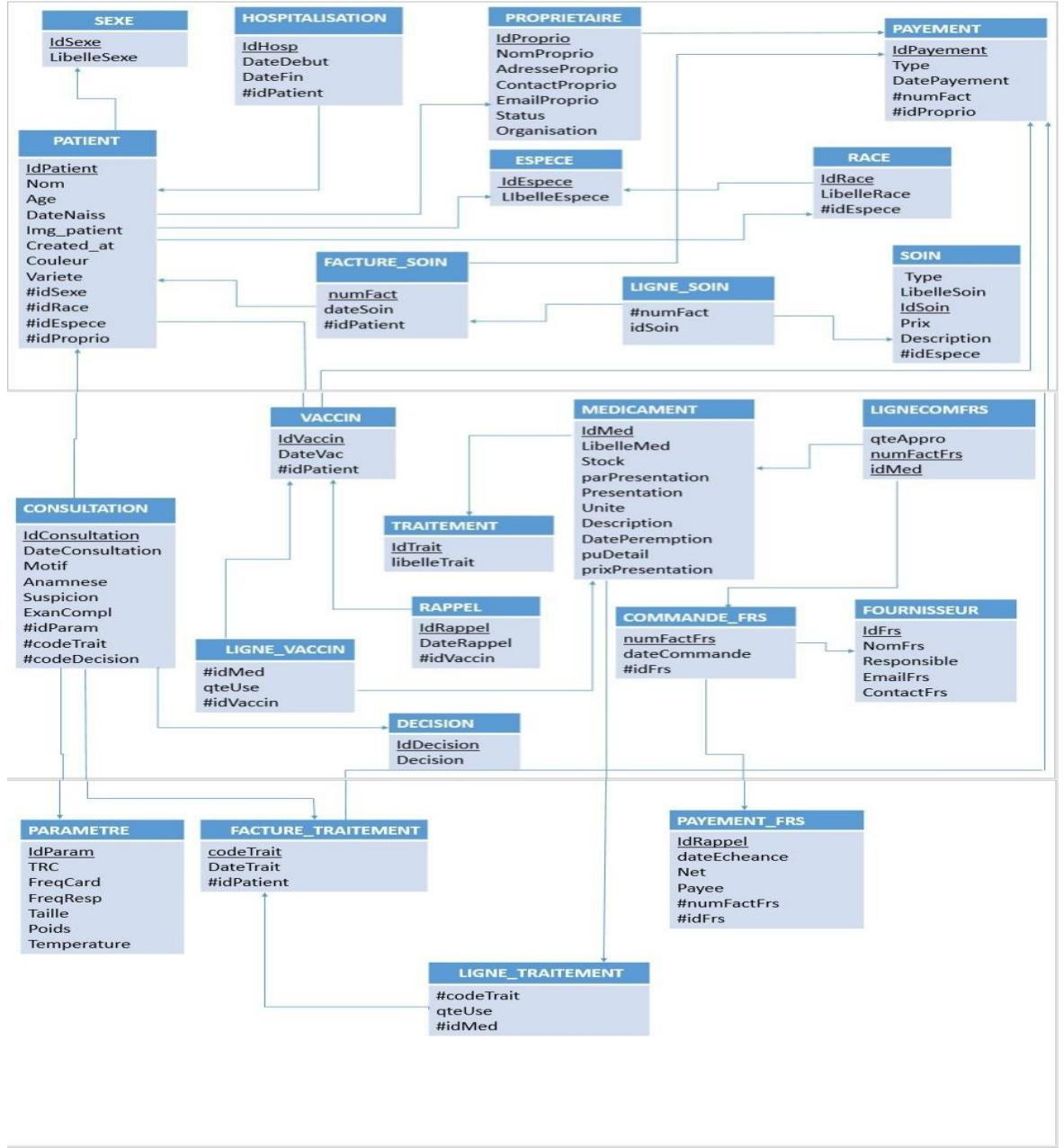
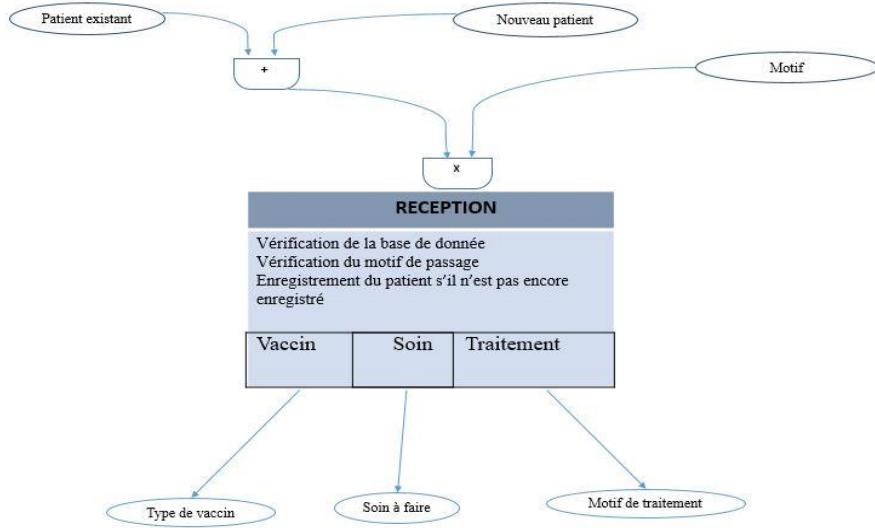


FIGURE 7: MODELE LOGIQUE DES DONNEES

5.5.3 Le Modèle Conceptuel des Traitements

Le Modèle Conceptuel des Traitements définit l'activité d'une entreprise indépendamment des organisations et moyens d'exécution choisis. Ces sont les résultats et la traduction en action des règles des gestions instaurées.

La figure 8 montre le Modèle Conceptuel de Traitement concernant la réception d'un client au sein de la clinique Boulevard.



+ : OU

FIGURE 8:MCT DE LA RECEPTION AU CLINIQUE

La figure 9 montre le MCT concernant une vaccination

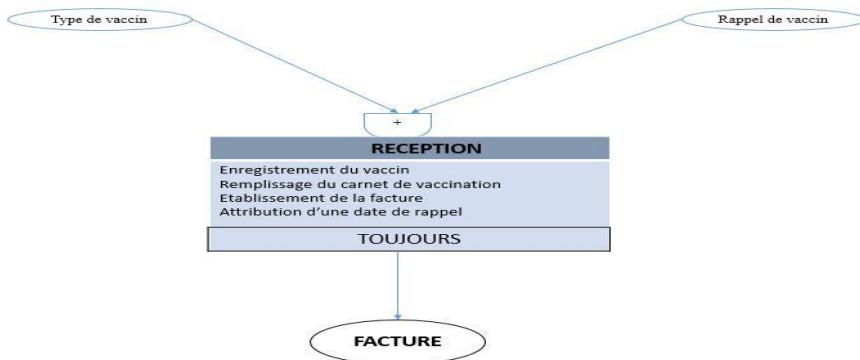


FIGURE 9:MCT POUR UNE VACCINATION

La figure 10 représente une consultation pour un patient suivi de ses traitements.

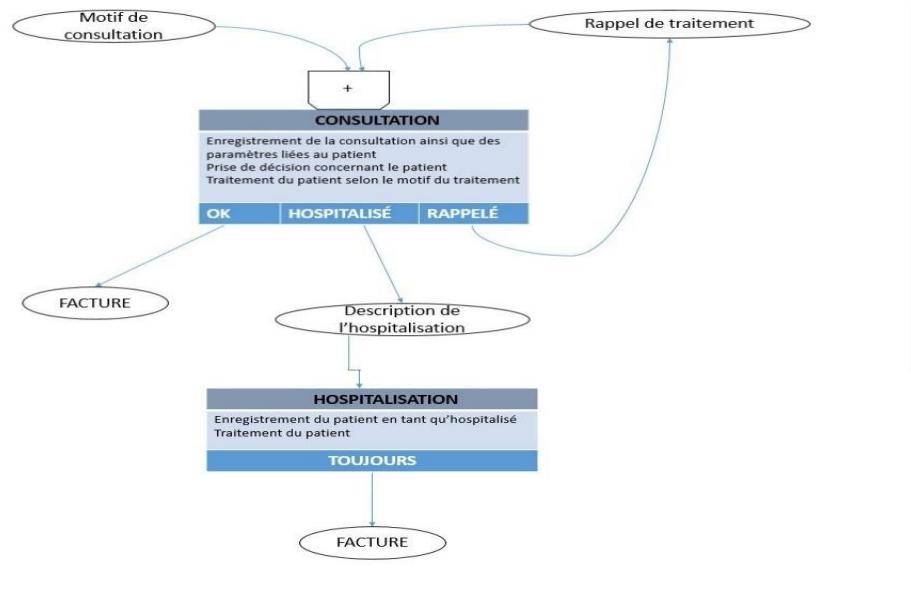


FIGURE 10:MCD POUR LE TRAITEMENT D'UN PATIENT

La figure 11 représente le MCT sur le paiement client/fournisseur qui suit la même méthode de traitement.

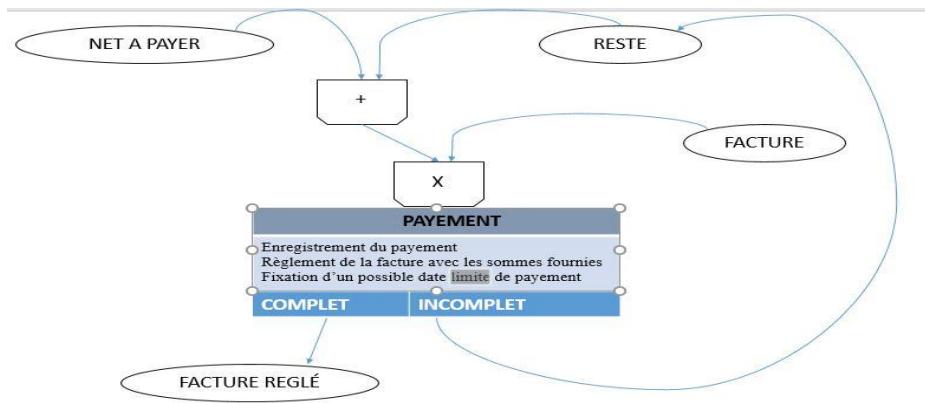


FIGURE 11: MCT SUR LES PAYEMENTS

+ : OU

X : ET

5.5.4 Le Modèle Organisationnel des Traitements

Le Modèle Organisationnel de Traitement spécifie l'organisation sur le traitement des données et des opérations : il répond aux questions QUI, QUAND et Où qui donneront une vue sur l'organisation à mettre en place lors de l'implémentation du nouveau système.

Le tableau 13 montre le Modèle Organisationnel des Traitements

TABLEAU 13: MODELE ORGANISATIONNEL DES TRAITEMENTS

| PERIODE | TRAITEMENT | ACTEURS | TYPE |
|-------------------------|--|------------------|------------|
| Réception d'un patient | <ul style="list-style-type: none"> • Vérification de la base de données • Enregistrement du patient en tant que patient de la clinique et son propriétaire en tant que client s'ils sont nouveaux • Traitement du patient en fonction du motif de son visite | Service Médicaux | Manuelle |
| Lors d'une consultation | <ul style="list-style-type: none"> • Vérification du patient et enregistrement s'il est nouveau • Enregistrement de la consultation • Prise de décision correspondant à l'état du patient • Traitement du patient • Etablissement de la facture | Service Médicaux | Automatisé |
| Lors d'une vaccination | <ul style="list-style-type: none"> • Vérification du patient et enregistrement s'il est nouveau • Enregistrement de la vaccination • Etablissement d'une date de rappel s'il faut le rappeler • Etablissement de la facture des dépenses | Service Médicaux | Automatisé |
| Lors d'une soin | <ul style="list-style-type: none"> • Vérification du patient et enregistrement s'il est nouveau • Enregistrement des soins • Etablissement de la facture des dépenses | Service Médicaux | Automatisé |

| | | | |
|-----------------------------|--|--------------------|------------|
| Lors d'un approvisionnement | <ul style="list-style-type: none"> • Enregistrement des achats • Demande de crédit si l'on ne paye pas le tout directement • Etablissement des dépenses | Service Financière | Automatisé |
| Lors d'un paiement | <ul style="list-style-type: none"> • Enregistrement de la facture • Détermination du type de facture : pour fournisseur ou issue d'un client • Règlement d'un tiers ou de tous les dépenses | Service Financière | Automatisé |

5.5.5 Le Modèle Physique des Données

Le Modèle Physique des Données est plus concret que le MCD ou MLD, donc c'est le modèle qui s'approche le mieux des données utilisées. Il aide ainsi à analyser les tables, les vues ou encore les objets nécessaires pour être utilisé par l'application par la suite.

La figure 12 représente le Modèle Physique des Données.

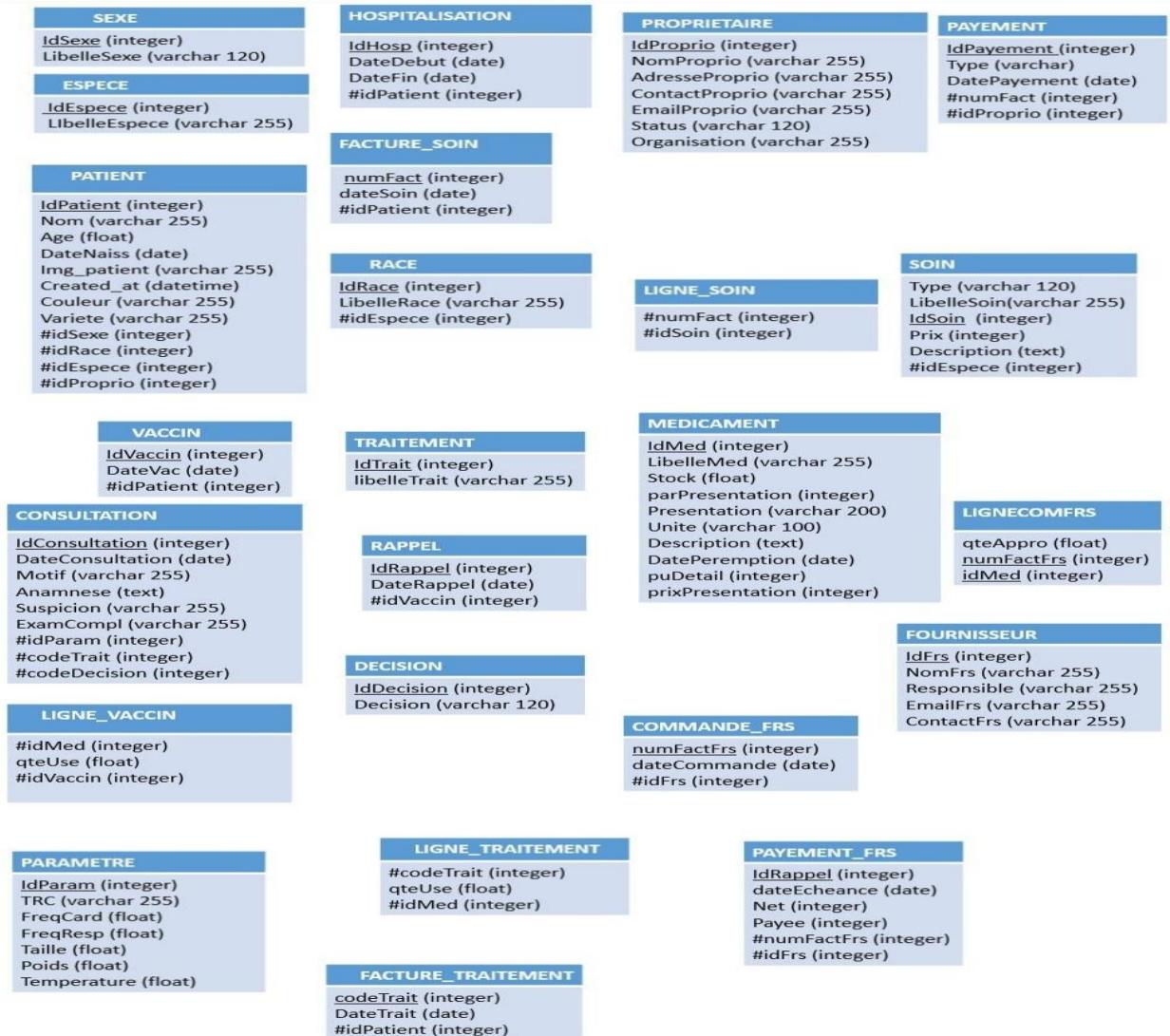


FIGURE 12: MODELE PHYSIQUE DES DONNEES

PARTIE III : REALISATION

Chapitre 6 : Installation et configuration des outils

6.1 Laragon et le SGBD MySQL

Le SGBD MySQL est souvent embarqué sur un logiciel comme les modules Apache et PHP aussi, il offre une flexibilité avec un gain de temps considérable sur la gestion de notre BDD. Apache et PHP déjà inclus, l'installation est aussi simplifiée comme tout logiciel. Il est disponible sur le site officiel de [Laragon](#)³ avec des versions pro et Lite. Notons qu'on peut directement télécharger Laragon avec un fichier Zip qui contient tous les dossiers.

La figure 13 montre la page de téléchargement de Laragon Lite 4 qui va télécharger un fichier Zip contenant un exécutable et les dossiers de Laragon.

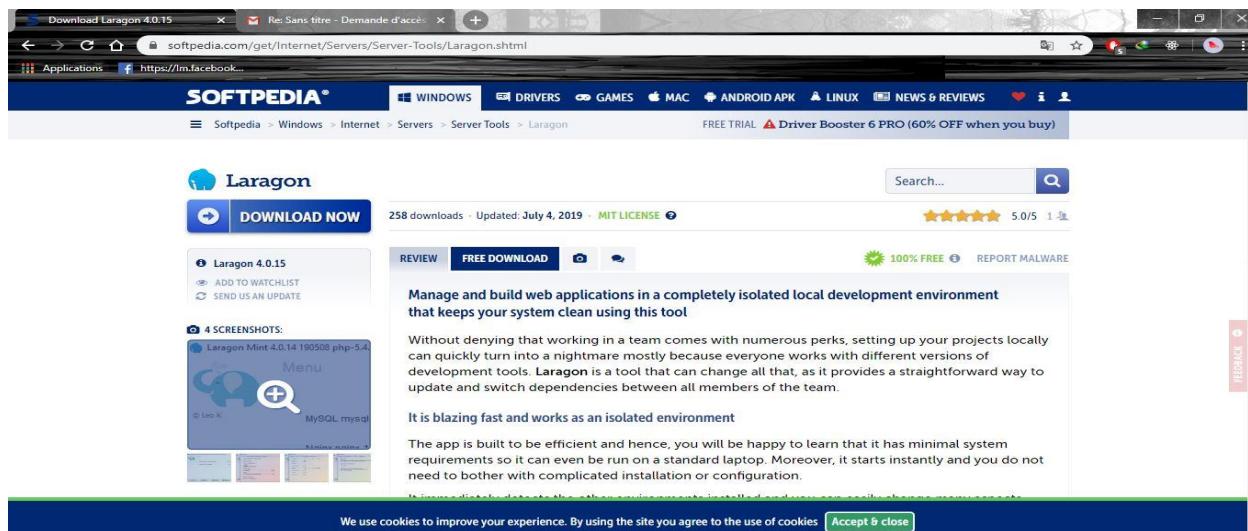


FIGURE 13:TELECHARGEMENT DU FICHIER ZIP DE LARAGON

Une fois installée et lancé, l'interface de Laragon s'affiche avec un dicton différent à chaque fois qu'on l'ouvre. La figure 14 montre la page d'accueil de Laragon.



FIGURE 14: PAGE D'ACCUEIL DE L'APPLICATION LARAGON

On voit déjà que PHP 7.2.19 est installé. On peut directement démarrer notre serveur en cliquant sur démarrer qui enclenchera le lancement d'Apache et MySQL. Dans Base de données, on accèdera directement à notre base de données comme le montre la figure 15 et 16 ;

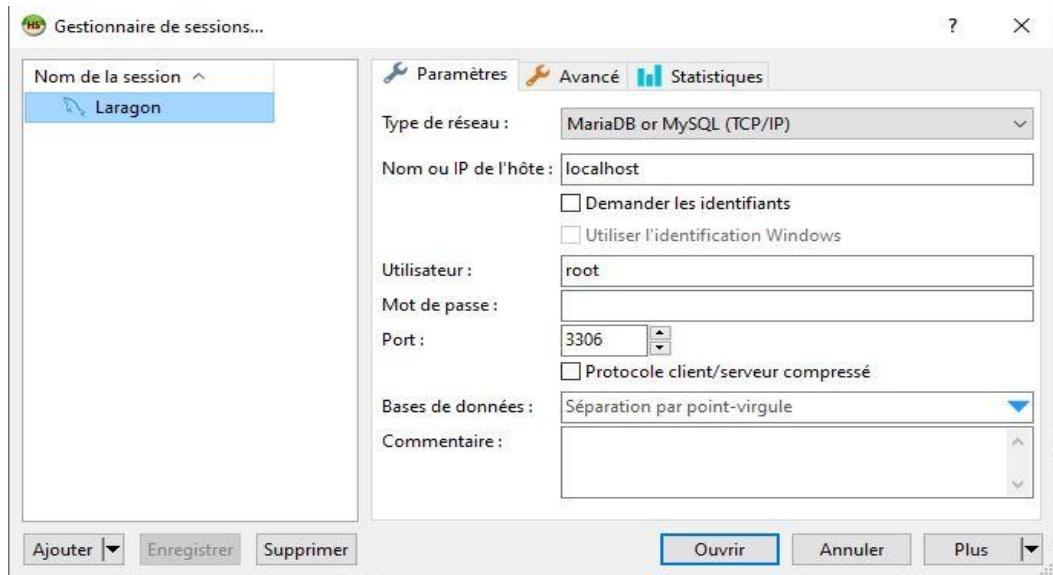


FIGURE 15: AUTHENTICATION POUR LA CONNEXION A LA BDD

The screenshot shows the HeidiSQL Portable interface connected to 'localhost' with the database 'pharmacie'. The left sidebar shows the schema structure:

- cybb
- gestion
- instat
- klay
- mysql
- payment
- performance_schema
- pharmacie** (selected, 400,0 KiB)
 - Tables: consultation, decision, espece, facture_com... (2), facture_soin (39), facture_trait... (9), fournisseur, hospitalisation, lignecomfrs (2), ligne_soin (21), ligne_traitement (15), ligne_vaccin (8)
 - Vues
 - Procédures
 - Fonctions
 - Déclencheurs
 - Événements

The main pane displays a table of tables with columns: Nom, Lignes, Taille, Crée le, Mis à jour, Moteur, Commentaire, and Typ. The table data is as follows:

| Nom | Lignes | Taille | Crée le | Mis à jour | Moteur | Commentaire | Typ |
|------------------|--------|----------|---------------------|------------|--------|-------------|-----|
| consultation | 9 | 16,0 KiB | 2020-02-01 15:12:09 | | InnoDB | | |
| decision | 4 | 16,0 KiB | 2020-02-01 15:12:10 | | InnoDB | | |
| espece | 2 | 16,0 KiB | 2020-02-01 15:12:10 | | InnoDB | | |
| facture_com... | 2 | 16,0 KiB | 2020-02-01 15:12:11 | | InnoDB | | |
| facture_soin | 39 | 16,0 KiB | 2020-02-01 15:12:11 | | InnoDB | | |
| facture_trait... | 9 | 16,0 KiB | 2020-02-01 15:12:12 | | InnoDB | | |
| fournisseur | 0 | 16,0 KiB | 2020-02-01 15:12:12 | | InnoDB | | |
| hospitalisation | 4 | 16,0 KiB | 2020-02-01 15:12:12 | | InnoDB | | |
| lignecomfrs | 2 | 16,0 KiB | 2020-02-01 15:12:13 | | InnoDB | | |
| ligne_soin | 21 | 16,0 KiB | 2020-02-01 15:12:13 | | InnoDB | | |
| ligne_traitement | 15 | 16,0 KiB | 2020-02-01 15:12:14 | | InnoDB | | |
| ligne_vaccin | 8 | 16,0 KiB | 2020-02-01 15:12:14 | | InnoDB | | |

The bottom pane shows the MySQL command history and connection status:

```

21 SHOW FUNCTION STATUS WHERE `Db`='information_schema';
22 SHOW PROCEDURE STATUS WHERE `Db`='information_schema';
23 SHOW TRIGGERS FROM `information_schema`;
24 SHOW EVENTS FROM `information_schema`;
25 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.EVENTS WHERE `EVENT_SCHEMA`='pharmacie';

```

pharmacie: 25 tables Connecté : 0:0:0 MariaDB or MySQL 5.7.2 Disponibilité : 00:00 h Temps sur le serveur : En attente

FIGURE 16: LISTE DES BASES DES DONNEES ET DES TABLES

6.2 Visual Studio Code : notre outil de développement

L'outil phare des développeurs, VS Code est très puissant pour la gestion des grands comme des petits projets. L'installation de cet outil est plutôt facile, en plus il est disponible sur d'autre OS comme Linux sans passer par d'autre outil intermédiaire. Son site officiel explique comment le télécharger puis l'installer sur notre machine.

La figure 17 montre le [site officiel](https://code.visualstudio.com/download)⁴ de VS Code avec le zone téléchargement

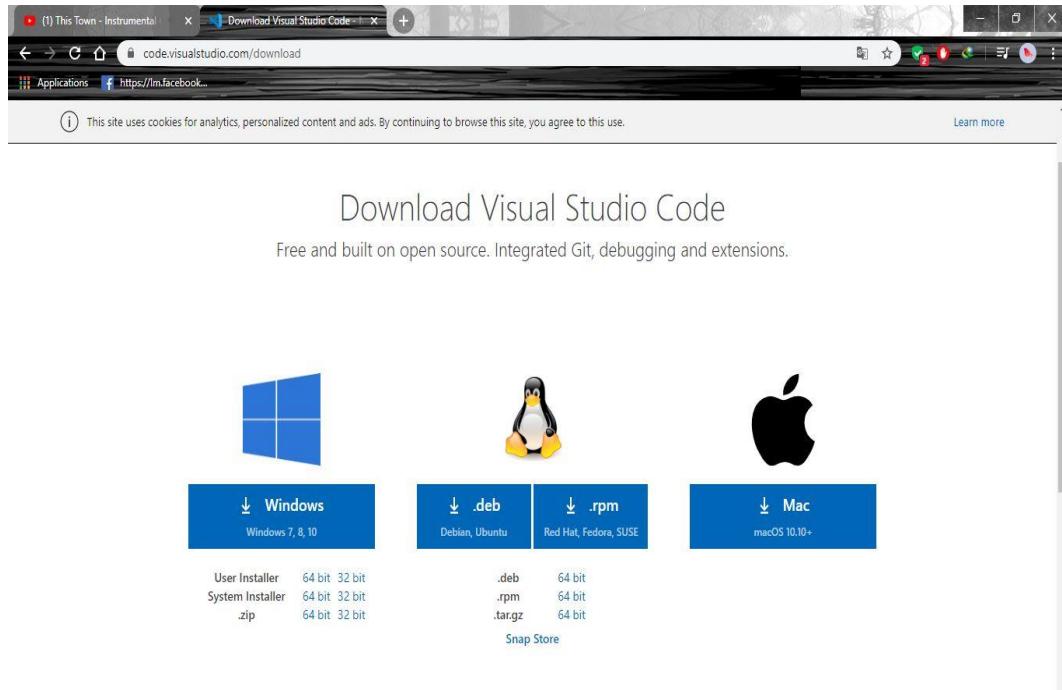


FIGURE 17: SITE OFFICIEL POUR LE TELECHARGEMENT DE VS CODE

La figure 18 montre l'installation de VS Code sur la machine.

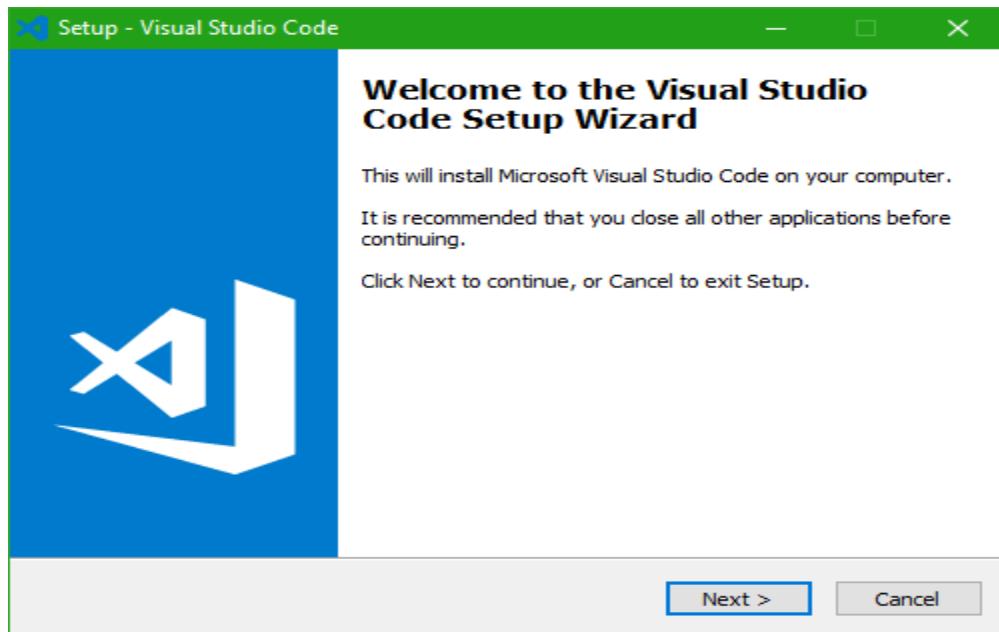


FIGURE 18: INSTALLATION DE VS CODE

Après l'installation, on lance VS Code et par la suite, on vient sur l'espace de travail où l'on pourra ajouter le dossier de notre projet. La figure 19 montre cet espace de travail lors de la première ouverture de VS Code.

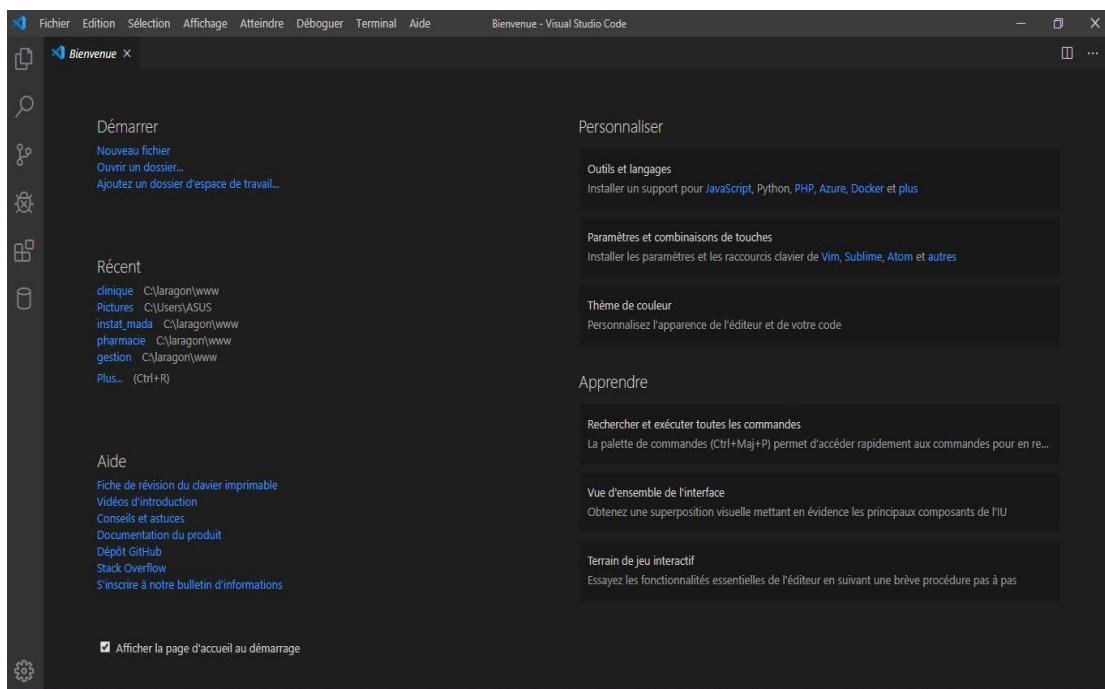


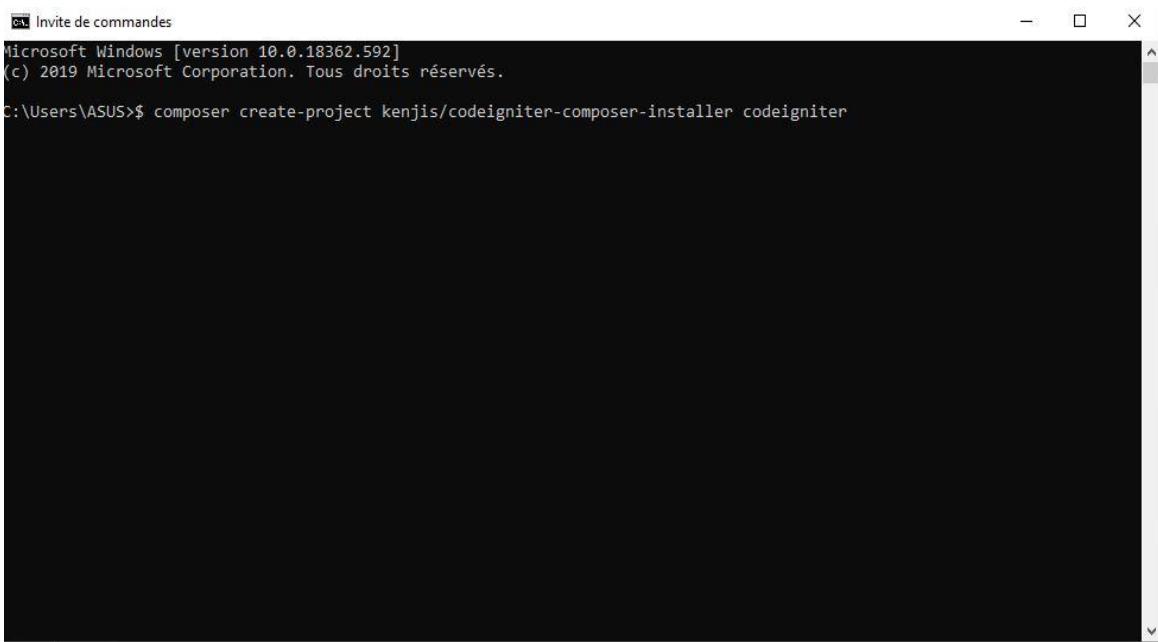
FIGURE 19: PAGE D'ACCUEIL

Chapitre 7 : Architecture de l'application

7.1 Téléchargement d'une application Codeigniter 3

- Via Composer :

Composer [4] est le gestionnaire de dépendance de PHP. Toute application avec un Framework PHP peut passer par celui-ci s'il dépend par exemple d'un autre Framework comme c'est le cas de Laravel et Symfony. Composer peut donc directement installer la maquette de l'application dans un répertoire qu'on choisit lors du lancement de la commande sur une console. Composer s'obtient de deux façons, soit en installant le logiciel elle-même par le biais d'un exécutable soit par ligne de commande. La méthode par ligne de commande est très simple et aussi très rapide. Après avoir installé Composer, on peut ouvrir notre invite de commande et lancer la commande sur la figure 20.



```
Windows [version 10.0.18362.592]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\ASUS>$ composer create-project kenjis/codeigniter-composer-installer codeigniter
```

FIGURE 20 : CREATION D'UNE APPLICATION CODEIGNITER EN LIGNE DE COMMANDE AVEC COMPOSER

Après avoir lancé cette commande, Composer va créer dans le dossier codeigniter inclus dans laragon/www une application qui sera nommé codeigniter.

- Via son site officiel : la plus simple des solutions est de télécharger directement l'application sur son [site web](#)⁵. Après l'avoir téléchargé, on n'a plus qu'à écrire nos programmes. La figure 21 montre la racine de l'application CodeIgniter ;

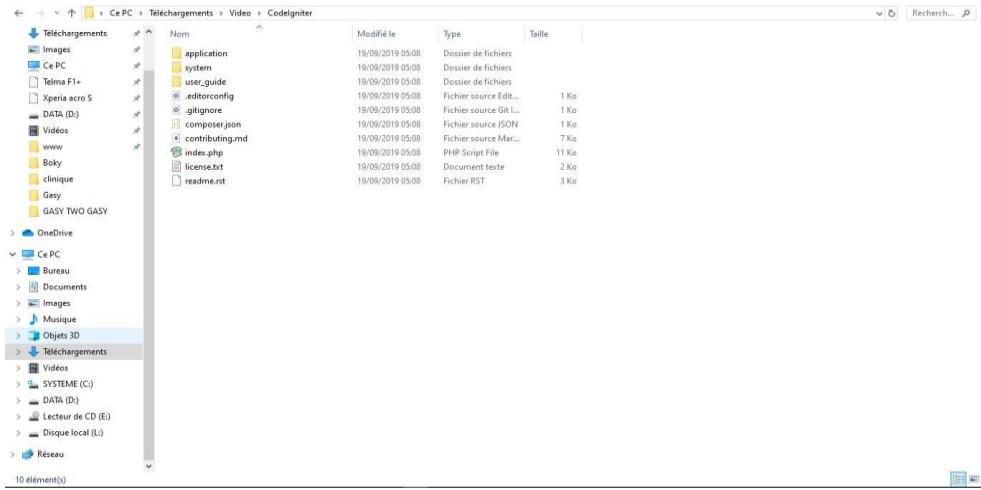


FIGURE 21: STRUCTURE PAR DEFAUT DE L'APPLICATION

- Via GitHub : pour la télécharger sous Git, on peut lancer la commande sur la figure 22 sous notre console Git, préalablement installé⁶.

```
MINGW64:/c/Users/ASUS
ASUS@RAJAONARISON-CLAIRMONT MINGW64 ~
$ git clone https://github.com/bcit-ci/CodeIgniter.git
```

FIGURE 22: CLONAGE D'UN PROJET SOUS GITHUB

7.2 Description des dossiers et fichier au sein de notre application

- **Application** : c'est là qu'on va mettre tous les codes de notre application, les vues, modèles, les librairies...
- **System** : ce dossier s'occupe de la configuration globale de notre application vis-à-vis de son environnement actuel.
- **User_guide** : c'est la documentation embarquée de Codeigniter. On peut y accéder directement même en local, en tapant seulement **root/user_guide** où root désigne la racine de notre application.

Les autres fichiers tels que composer.json ou encore .gitignore sont là pour permettre à notre application de s'intégrer facilement lors de sa mise en production.

Dans le dossier **application**, on peut constater aussi plusieurs dossiers présentés sur la figure 23.

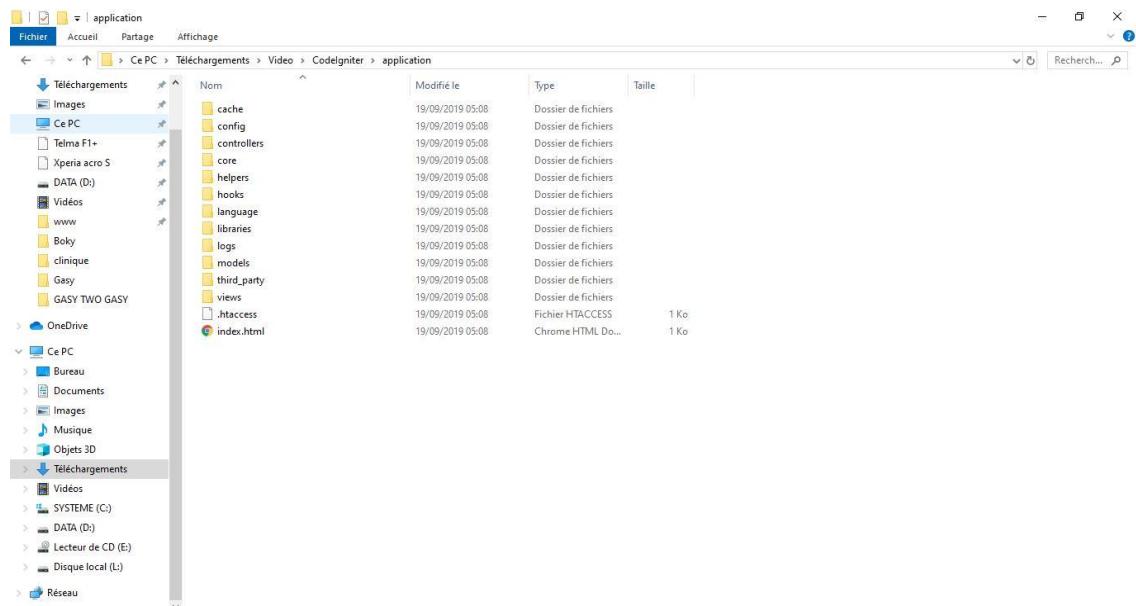


FIGURE 23: CONTENU DU DOSSIER APPLICATION

On va décortiquer une à une les contenus de ces dossiers :

- **Cache** : c'est le dossier en charge de notre système de caching si on en veut mettre en place une application avec des systèmes de cache. Très utile dans les applications comme GoogleMaps pour une utilisation hors-ligne.
- **Config** : c'est le dossier de configuration générale de notre application, comme son nom, la racine ou encore l'index file, son url de base...
- **Controllers** : c'est là qu'on va mettre nos **controllers**, dont les méthodes deviendront nos **url**.
- **Core** : c'est le dossier où l'on met nos propres classes si on veut modifier les comportements natives de Codeigniter.

- **Helpers** : c'est le dossier des fichiers helper, des fichiers composés des fonctions groupées par thèmes qu'on a créés nous-même et qu'on doit utiliser dans notre code.
- **Hooks** : c'est le dossier des hooks
- **Language** : le dossier des langages utilisés par CI. Par défaut, CodeIgniter utilise l'anglais mais on peut le configurer en français ou dans un autre langue grâce à [des packs de langue sur internet](#)⁷.
- **Librairies** : comme son nom l'indique, c'est là-dedans qu'on va mettre les librairies externes qu'on utilisera dans le projet. On va par exemple utiliser [html2pdf](#)⁸ et on va donc le mettre dans ce dossier.
- **Logs** : si on souhaite mettre des logs lors des erreurs par exemple, on crée le fichier de log dans ce dossier.
- **Models** : c'est le dossier des modèles, qui sera en charge de la communication à la base de données.
- **Third_party** : un des dossiers pour les fichiers apart.

Views : c'est le dossier où l'on va mettre les vues, c'est-à-dire les fichiers qui vont s'occuper des affichages côtés clients.

7.3 Architecture de notre application

Comme on l'a déjà vue, CodeIgniter intègre directement au sein de son application l'architecture MVC [4]. Cette architecture se présente en 3 parties avec des classes mères pour chacune d'elle :

- ❖ **Model** : tout modèle au sein de CodeIgniter doit hériter de la classe CI_Model. Les modèles s'occupent de la communication avec la base, ils exécutent les requêtes, récupèrent les valeurs, insérer des données ou encore les modifier. Bien sûr, les modèles peuvent aussi afficher des résultats. Le nom du fichier doit correspondre au nom de la classe et vice-versa.
- ❖ **Controller** : c'est lui qui s'occupe du routage de l'application. A chaque fois que le client fait une requête sur le site, c'est le controller qui est appelé en premier. Il va analyser la requête et donner un résultat en fonction de celui-ci. C'est lui qui renvoi les résultats d'un model et les injectent dans la view. Il hérite de la classe CI_Controller. On peut cependant créer un controller qui n'hérite pas de cette classe et ainsi configurer nous-même le routage. Toutes méthodes dans le controller deviennent automatiquement des url et peuvent être appelé comme suit : **racine/controller/methode/** avec éventuellement un ou plusieurs paramètres. Le nom du fichier doit aussi correspondre au nom de la classe comme avec les models.
- ❖ **View** : c'est lui qui s'occupe de l'affichage, de l'interface graphique. Il sera donc composé la plupart des codes Html et CSS et des codes PHP aussi. Le view ne s'occupe pas des traitements des données, il ne fait qu'interagir avec les données qu'il reçoit du controller.

Chapitre 8 : Développement de l'application

8.1 Crédit de la base de données

La base de données de l'application est issue du MPD qui va donc représenter les tables de cette base. Comme [MySQLWorkbench](#) [5] peut exporter la structure de la base de données pour en faire un fichier SQL, on n'a plus qu'à importer ce fichier pour notre base. La figure 24 montre comment générer ce fichier sous MySQLWorkbench. Il faut cliquer sur Fichier > Exporter et exporter en tant Forward Engineering SQL, ou alors cliquer sur la combinaison Ctrl+Maj+G.

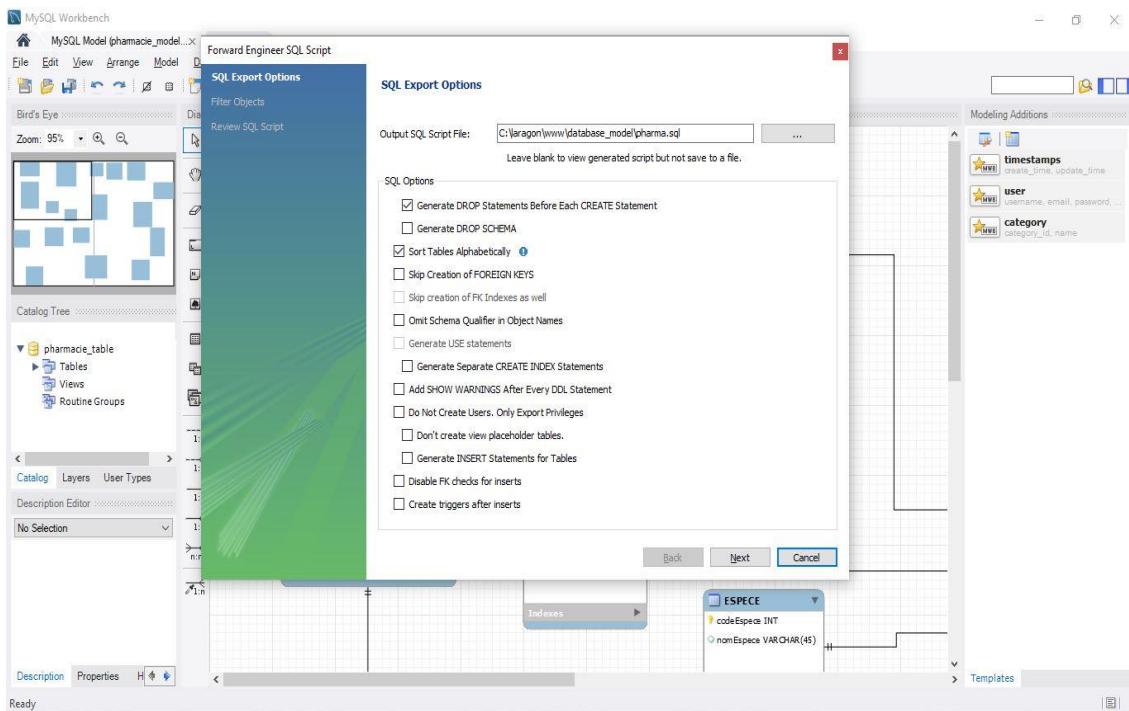


FIGURE 24: GENERATION DU FICHIER SQL SOUS MySQL WORKBENCH

On choisit le nom et la location du fichier et on clique sur NEXT. La figure 25 configure les structures qu'on veut exporter.

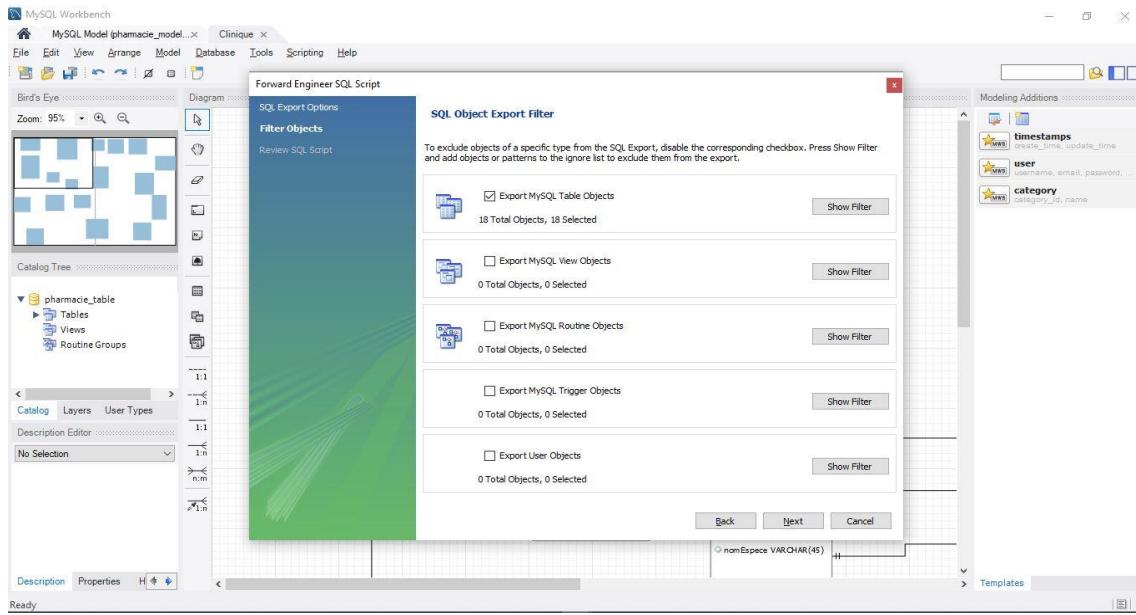


FIGURE 25: FILTRAGE DES OBJETS A EXPORTER

On clique sur NEXT quand on a filtré tous ce qu'on veut exporter, et la figure 26 montre l'étape finale de la génération du fichier SQL.

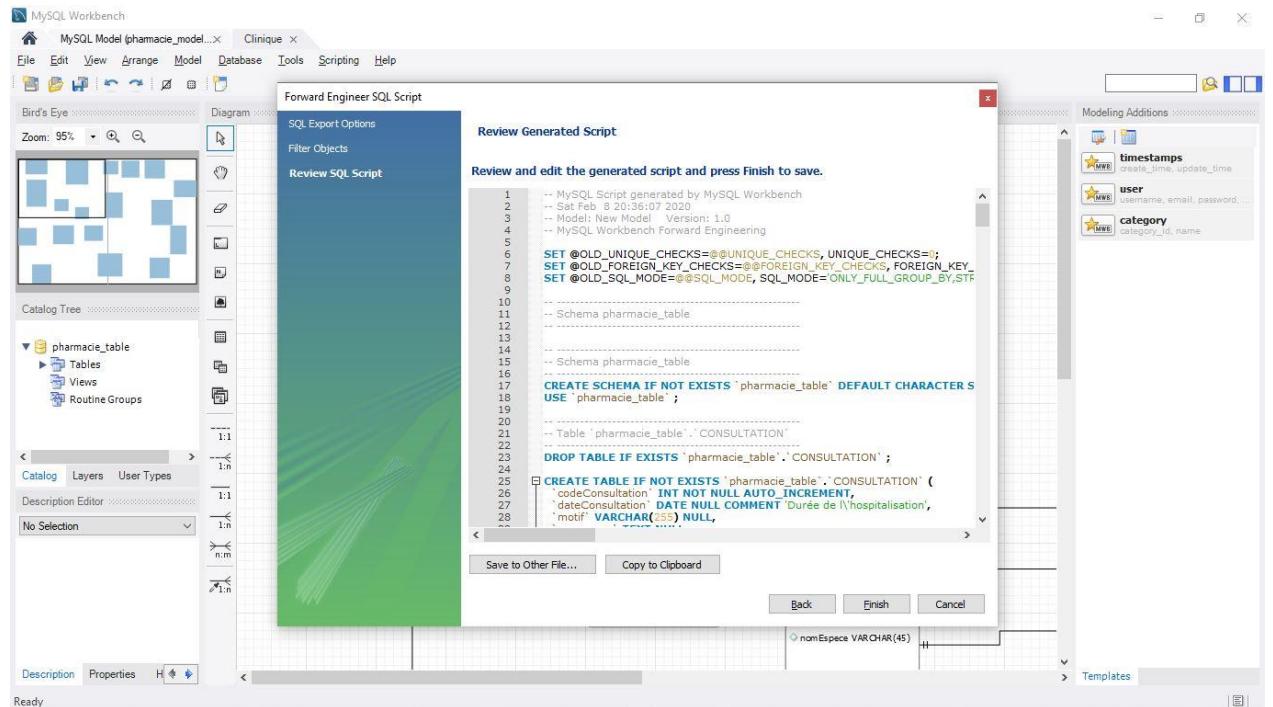


FIGURE 26: CREATION DU FICHIER SQL

Après avoir cliqué sur FINISH, on obtient un fichier SQL qui contient la création de la base de données ainsi que sa structure avec les tables et les colonnes.

Laragon supporte l'importation des fichiers SQL qui facilite et accélère la création de notre base de données. La figure 27 montre l'importation d'un fichier SQL sous Laragon.

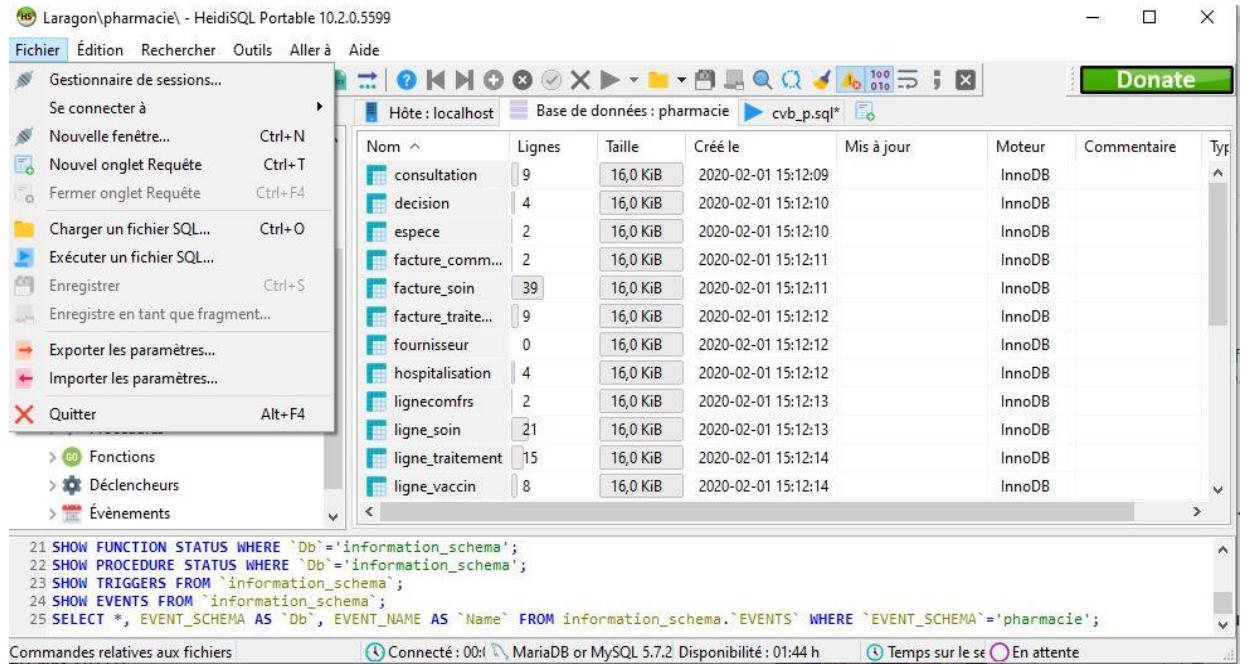


FIGURE 27: IMPORTATION D'UN FICHIER SQL SOUS LARAGON

On peut ainsi choisir entre exécuter un fichier SQL ou d'abord le charger. On va ici charger notre fichier SQL. La figure 28 montre comment choisir notre fichier sur notre disque dur.

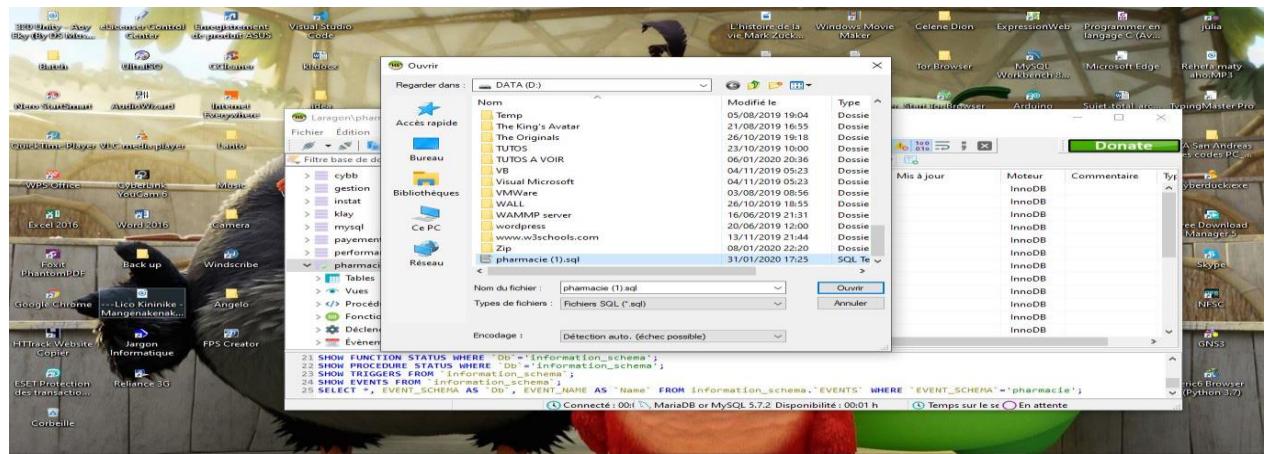


FIGURE 28: CHARGER UN FICHIER SQL

Après avoir choisi notre fichier et cliqué sur ouvrir, on n'a plus qu'à l'exécuter sur Laragon qui va entraîner la création de notre base de données ainsi que ses structures.

La figure 29 montre la dernière étape à faire pour la création de notre base de données.

```

1 -- phpMyAdmin SQL Dump
2 -- version 4.1.14
3 -- http://www.phpmyadmin.net
4 --
5 -- Client : 127.0.0.1
6 -- Généré Le : Ven 31 Janvier 2020 à 1
7 -- Version du serveur : 5.6.17
8 -- Version de PHP : 5.5.12
9 -- 
10 SHOW PROCEDURE STATUS WHERE `Db`='information_schema';
11 SHOW TRIGGERS FROM `information_schema`;
12 SHOW EVENTS FROM `information_schema`;
13 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.EVENTS WHERE `EVENT_SCHEMA`='pharmacie';
14 /* Loading file "D:\pharmacie (1).sql" (24,9 KiB) into query tab #2 ... */

```

Connecté: 00:01 MariaDB or MySQL 5.7.2 Disponibilité: 00:08 h Temps sur le serveur: En attente

FIGURE 29: EXECUTION DU FICHIER SQL

Notre base de données est fin prêt, on peut alors commencer à coder notre application. La figure 30 montre notre base de données au sein de Laragon.

| Nom | Lignes | Taille | Créé le | Mis à jour | Moteur | Commentaire | Type |
|-------------------|--------|----------|---------------------|------------|--------|-------------|-------|
| consommation | 4 | 16,0 KiB | 2020-02-01 15:12:10 | | InnoDB | | Table |
| decision | 4 | 16,0 KiB | 2020-02-01 15:12:10 | | InnoDB | | Table |
| espece | 2 | 16,0 KiB | 2020-02-01 15:12:10 | | InnoDB | | Table |
| facture_commun... | 2 | 16,0 KiB | 2020-02-01 15:12:11 | | InnoDB | | Table |
| facture_soin | 39 | 16,0 KiB | 2020-02-01 15:12:11 | | InnoDB | | Table |
| facture_trait... | 9 | 16,0 KiB | 2020-02-01 15:12:11 | | InnoDB | | Table |
| fournisseur | 0 | 16,0 KiB | 2020-02-01 15:12:12 | | InnoDB | | Table |
| hospitalisation | 4 | 16,0 KiB | 2020-02-01 15:12:12 | | InnoDB | | Table |
| ligneconfr... | 2 | 16,0 KiB | 2020-02-01 15:12:13 | | InnoDB | | Table |
| ligne_soin | 21 | 16,0 KiB | 2020-02-01 15:12:13 | | InnoDB | | Table |
| ligne_traitement | 15 | 16,0 KiB | 2020-02-01 15:12:14 | | InnoDB | | Table |
| ligne_vaccin | 8 | 16,0 KiB | 2020-02-01 15:12:14 | | InnoDB | | Table |
| medicament | 101 | 16,0 KiB | 2020-02-01 15:12:14 | | InnoDB | | Table |
| parametre | 0 | 16,0 KiB | 2020-02-01 15:12:15 | | InnoDB | | Table |
| patient | 10 | 16,0 KiB | 2020-02-01 15:12:15 | | InnoDB | | Table |
| paiement_client | 12 | 16,0 KiB | 2020-02-01 15:12:16 | | InnoDB | | Table |
| paiement_frs | 2 | 16,0 KiB | 2020-02-01 15:12:16 | | InnoDB | | Table |
| proprietaire | 8 | 16,0 KiB | 2020-02-01 15:12:16 | | InnoDB | | Table |
| race | 5 | 16,0 KiB | 2020-02-01 15:12:17 | | InnoDB | | Table |
| rappel | 0 | 16,0 KiB | 2020-02-01 15:12:17 | | InnoDB | | Table |
| sexe | 4 | 16,0 KiB | 2020-02-01 15:12:17 | | InnoDB | | Table |
| soin | 27 | 16,0 KiB | 2020-02-01 15:12:18 | | InnoDB | | Table |
| traitement | 22 | 16,0 KiB | 2020-02-01 15:12:18 | | InnoDB | | Table |
| user | 3 | 16,0 KiB | 2020-02-01 15:12:19 | | InnoDB | | Table |
| vaccin | 8 | 16,0 KiB | 2020-02-01 15:12:19 | | InnoDB | | Table |

```

22 SHOW PROCEDURE STATUS WHERE `Db`='information_schema';
23 SHOW TRIGGERS FROM `information_schema`;
24 SHOW EVENTS FROM `information_schema`;
25 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.EVENTS WHERE `EVENT_SCHEMA`='pharmacie';
26 /* Loading file "D:\pharmacie (1).sql" (24,9 KiB) into query tab #2 ... */

```

Peut contenir des expressions régulières, par ex. « phpb:\d » Connecté: 00:16 h MariaDB or MySQL 5.7.24 Disponibilité: 00:16 h Temps sur le serveur: En attente

FIGURE 30: LA BASE DE DONNEES DANS LARAGON

8.2 Codage de l'application

8.2.1 Crédit du dossier assets

Le dossier assets ou ressources est le dossier où l'on va mettre toutes les ressources qu'on aura besoin, que ce soit les feuilles de styles ou encore des ressources visuelles comme image ou

vidéo. Bref, ce dossier contient les ressources pour notre interface graphique. On n'impose pas le nom, on aurait pu la nommer **ressources, public**, tout ce qu'on veut. La figure 31 montre le contenu du dossier assets. Ici, on peut ajouter des ressources que l'on veut.

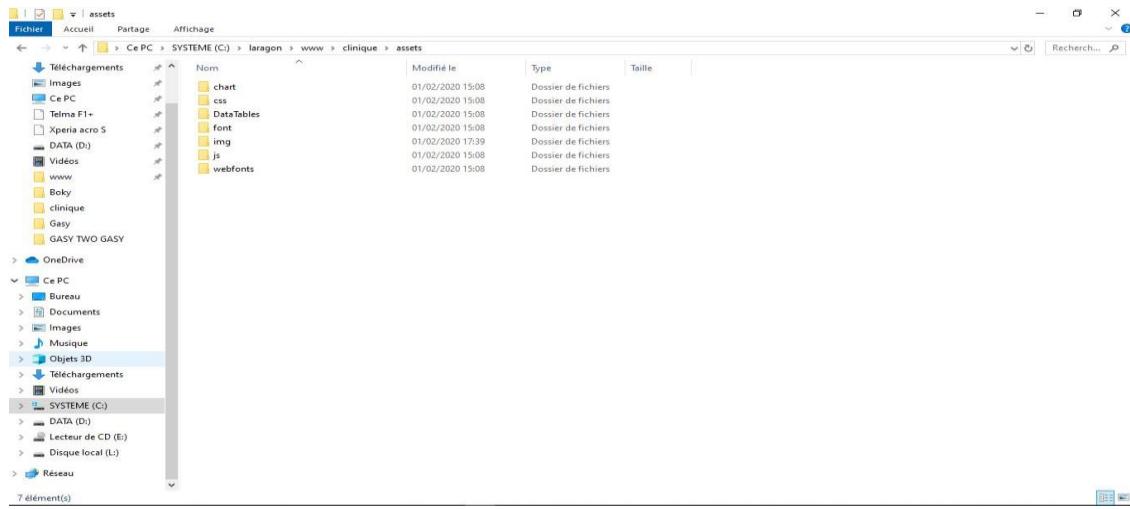


FIGURE 31: DOSSIER ASSETS

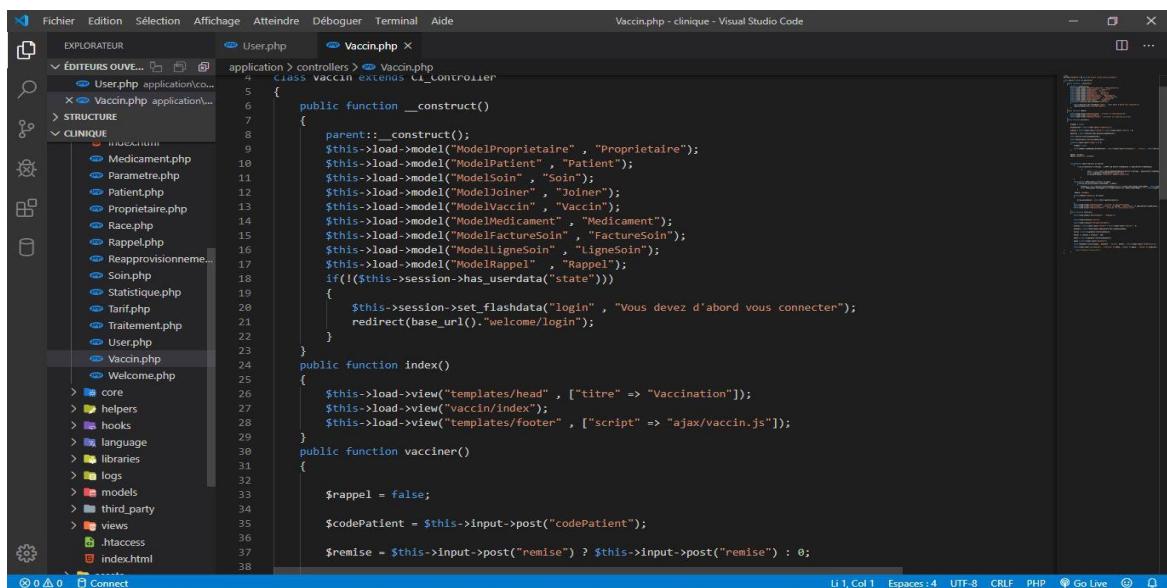
8.2.2 Les controllers

Ces controllers s'occupent des url et la liaison entre vue et modèle.

- ✓ **Consultation** : controller pour une consultation passée par le patient.
- ✓ **Especie** : controller sur la gestion des espèces des animaux au sein du cabinet.
- ✓ **Finance** : c'est le controller qui gère les finances du cabinet.
- ✓ **Fournisseur** : la gestion des fournisseurs de médicament est faite par ce Controller
- ✓ **History** : ceci gère l'historique, l'affichage ainsi que ce qui se rapporte aux historiques.
- ✓ **Hospitalisation** : c'est le controller sur la gestion d'une hospitalisation
- ✓ **Medicamement** : le controller pour la gestion des médicaments, des stocks et des CRUD sur les médicaments.
- ✓ **Parametre** : c'est le paramètre associé à un traitement reçu par un patient.
- ✓ **Patient** : c'est pour les patients répertoriés au sein du cabinet, leur profil...
- ✓ **Proprietaire** : controller pour la gestion des clients
- ✓ **Race** : qui s'occupe de la CRUD des races
- ✓ **Rappel** : le controller qui s'occupe de tous les rappels, que ce soit financièrement ou médicalement.
- ✓ **Reapprovisionnement** : le controller pour le réapprovisionnement au sein d'un fournisseur.
- ✓ **Soin** : c'est lui qui s'occupe de répertorier les soins subis par un patient ainsi que la gestion des activités au sein de la clinique.
- ✓ **Statistique** : c'est le controller de la page statistique. Il va gérer les charts de cette page.
- ✓ **Tarif** : s'occupe de la gestion des tarifs au sein du cabinet.

- ✓ **Traitement** : c'est le controller des catégories. On les nomme traitement puisqu'ils vont s'afficher sur la facture en tant que traitement.
- ✓ **User** : il gère les utilisateurs qui ont accès à l'application, crée, supprime ou modifie des comptes.
- ✓ **Vaccin** : ce controller est chargé de la gestion des vaccinations.
- ✓ **Welcome** : c'est notre controller par défaut. On peut le modifier en un autre controller mais on préfère le garder comme il est. Il va s'occuper de la page de connexion, d'enregistrement ainsi que de la page d'accueil.

La figure 32 montre une portion de code dans le controller Vaccination ;



```

Fichier Edition Sélection Affichage Atteindre Déboguer Terminal Aide
EXPLORATEUR Vaccin.php < User.php > application > controllers > Vaccin.php
    < User.php application/cont...
    < Vaccin.php application/cont...
> STRUCTURE
> CLINIQUE
    < Medicament.php
    < Parametre.php
    < Patient.php
    < Proprietaire.php
    < Race.php
    < Rappel.php
    < Reapprovisionneme...
    < Soin.php
    < Statistique.php
    < Tarif.php
    < Traitement.php
    < User.php
    < Vaccin.php
    < Welcome.php
> core
> helpers
> hooks
> language
> libraries
> logs
> models
> third_party
> views
    < .htaccess
    < index.html
0 0 ▲ 0 Connect
Vaccin.php - clinique - Visual Studio Code
4 class Vaccin extends CI_Controller
5 {
6     public function __construct()
7     {
8         parent::__construct();
9         $this->load->model("ModelProprietaire", "Proprietaire");
10        $this->load->model("ModelPatient", "Patient");
11        $this->load->model("ModelSoin", "Soin");
12        $this->load->model("ModelJoiner", "Joiner");
13        $this->load->model("ModelVaccin", "Vaccin");
14        $this->load->model("ModelMedicament", "Medicament");
15        $this->load->model("ModelFactureSoin", "FactureSoin");
16        $this->load->model("ModelLigneSoin", "LigneSoin");
17        $this->load->model("ModelRappel", "Rappel");
18        if(!($this->>session->has_userdata("state")))
19        {
20            $this->>session->set_flashdata("login", "Vous devez d'abord vous connecter");
21            redirect(base_url()."welcome/login");
22        }
23    }
24    public function index()
25    {
26        $this->load->view("templates/head", ["titre" => "Vaccination"]);
27        $this->load->view("vaccin/index");
28        $this->load->view("templates/footer", ["script" => "ajax/vaccin.js"]);
29    }
30    public function vacciner()
31    {
32        $rappel = false;
33
34        $codePatient = $this->input->post("codePatient");
35
36        $remise = $this->input->post("remise") ? $this->input->post("remise") : 0;
37
38

```

Li 1, Col 1 Espaces : 4 UTF-8 CRLF PHP Go Live

FIGURE 32: CONTROLLER VACCINATION

8.2.3 Les Models

Les models gèrent la communication avec la base de données donc on met nos requêtes dans les models. La figure 33 montre le model ModelSoin qui gère les soins.

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class ModelSoin extends CI_Model
{
    protected $table = "soin";

    public function tarif()
    {
        return $this->db->order_by("prix", "DESC")
            ->where("espece.codeEspece = soin.codeEspece")
            ->get($this->table, "espece");
    }

    public function getAll($type)
    {
        return $this->db->select("rubrique")
            ->distinct()
            ->where("espece.codeEspece = soin.codeEspece")
            ->where("type", $type)
            ->get($this->table, "espece");
    }

    public function getPriceByName($nomSoin)
    {
        return $this->db->where("rubrique", $nomSoin)
            ->get($this->table)
            ->result_array()[0]["prix"];
    }

    public function getSoInByName($nomSoin)
    {
        return $this->db->where("rubrique", $nomSoin)
            ->get($this->table);
    }
}

```

FIGURE 33 : MODELSOIN

8.2.4 Les views

Les views sont des fichiers qui s'occupent de l'affichage des données issues des controllers/models. Ces sont donc eux qui s'occupent de l'interface graphique de l'application, comme l'affichage de la liste des patients. C'est aussi les views qui affichent les formulaires et les envoient aux controllers. Ici, on a privilégié la solution de traitement en AJAX puisque cette technologie a l'avantage d'éviter un rechargeement de la page actuelle et aussi de traiter ces informations sous JavaScript.

La figure 34 montre le view patient qui affiche la liste des patients de la clinique.

```

<div id="search-result" class="mb-6" style="height:200px;overflow-y:scroll;display:none;">

```

```

<div class="accordion mt-4" id="accordionExample">
    <div class="card">
        <div class="card-header" id="headingOne">
            <h2 class="mb-0">
                <button class="btn btn-sm btn-outline-light" type="button" data-toggle="collapse" data-target="#col1"
                    aria-expanded="true" aria-controls="col1">
                    Liste des patients
                
            
        
```

FIGURE 34: VIEW INDEX DU CONTROLLER PATIENT

La figure 35 montre le fichier JS associé à ce view, qui s'occupe de récupérer les données via **AJAX** [6], ou encore de les envoyer.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** Fichier, Edition, Sélection, Affichage, Atteindre, Déboguer, Terminal, Aide.
- Toolbar:** EXPLORATEUR, User.php, Vaccin.php, ModelFactureTraitement.php, profil.php, index.php, patient.js (highlighted).
- Left Sidebar:** ÉDITEURS OUVERTS (User.php, Vaccin.php, ModelFactureTraitement.php, profil.php, index.php) and CLINIQUE (index.html, assets (chart, css, DataTables, font), js (comfrjs, consjs, financejs, fournisseurjs, history.js, medicament.js, patient.js, profilpatient.js, proprietaire.js, rappels.js, scriptProprio.js, soin.js)).
- Right Panel:** The patient.js file content is displayed, showing JavaScript code for handling patient lists and profiles.

```
assets > js > ajax > patient.js > ...  
var XHR = getHttp();  
var XHR2 = getHttp();  
  
XHR.open("GET", "http://"+url+"patient/liste");  
  
//XHR2.open("GET", "http://"+url+"patient/showChatList");  
  
XHR.onreadystatechange = function () {  
    if (XHR.status == 200 && XHR.readyState == 4) {  
        var response = JSON.parse(XHR.responseText);  
        var html = response.map(function (reponse) {  
            return '|'  
                + '<td style="text-align: center">' + reponse.nomPatient + '</td>'  
                + '<td style="text-align: center">' + reponse.libelle_espece + '</td>'  
                + '<td style="text-align: center">' + reponse.created_at + '</td>'  
                + '<td style="text-align: center"><a href="http://'+url+'patient/profil/' + reponse.codePatient + '" class="button">Voir Profil</a></td>'  
            + '
';  
        }).join("");  
        document.getElementById("liste_patient").innerHTML = html;  
    }  
};  
XHR.send();  
  
listPatient();
```

FIGURE 35: FICHIER PATIENT.JS

La figure 36 représente le fichier contenant les scripts JS avec des requêtes AJAX pour les propriétaires des animaux.

```
assets > js > ajax > proprietaire.js > getHttp
1  function getHttp() {
2    if (window.XMLHttpRequest) {
3      // code for modern browsers
4      return new XMLHttpRequest();
5    } else {
6      // code for old IE browsers
7      return new ActiveXObject("Microsoft.XMLHTTP");
8    }
9  }
10 function getProprio() {
11   var XML = new getHttp();
12   XML.open("GET", "http://"+url+"proprietaire/show");
13   XML.onreadystatechange = function () {
14     if (XML.readyState == 4 && XML.status == 200) {
15       if ((XML.responseText) == "[ ]") {
16         document.getElementById("tbody").innerHTML = "<td colspan='6' style='text-align:center'>Aucun proprietaire trouv</td>";
17       } else {
18         reponse = JSON.parse(XML.responseText);
19
20         html = reponse.map(function (proprio) {
21           if (proprio.organisation == "") {
22             proprio.organisation = "-";
23           }
24
25           return `<tr>
26             <td><a class='btn btn-outline-info' href='http://${url}proprietaire/profil/${proprio.codeProprio}'>${proprio.adresseProprio}</a></td>
27             <td>${proprio.contactProprio}</td>
28             <td>${proprio.emailProprio}</td>
29             <td>${proprio.status}</td>
30             <td>${proprio.organisation}</td>
31             <td><a href='http://${url}proprietaire/profil/${proprio.codeProprio}' onclick='affect(${proprio.codeProprio})'>Modifier</a></td>
32           </tr>`;
33         });
34       }
35     }
36   }
37 }
```

FIGURE 36: FICHIER JS POUR LES PROPRIETAIRES.

8.3 Présentation de l'application

8.3.1 Authentification

Pour pouvoir accéder à l'application, il faut un login et un mot de passe. Il y a deux types de comptes, un administrateur et un simple utilisateur. Ces comptes vont avoir des rôles différents et peuvent faire ou pas des actions dans l'application.

La figure 37 montre la page d'authentification pour les utilisateurs.

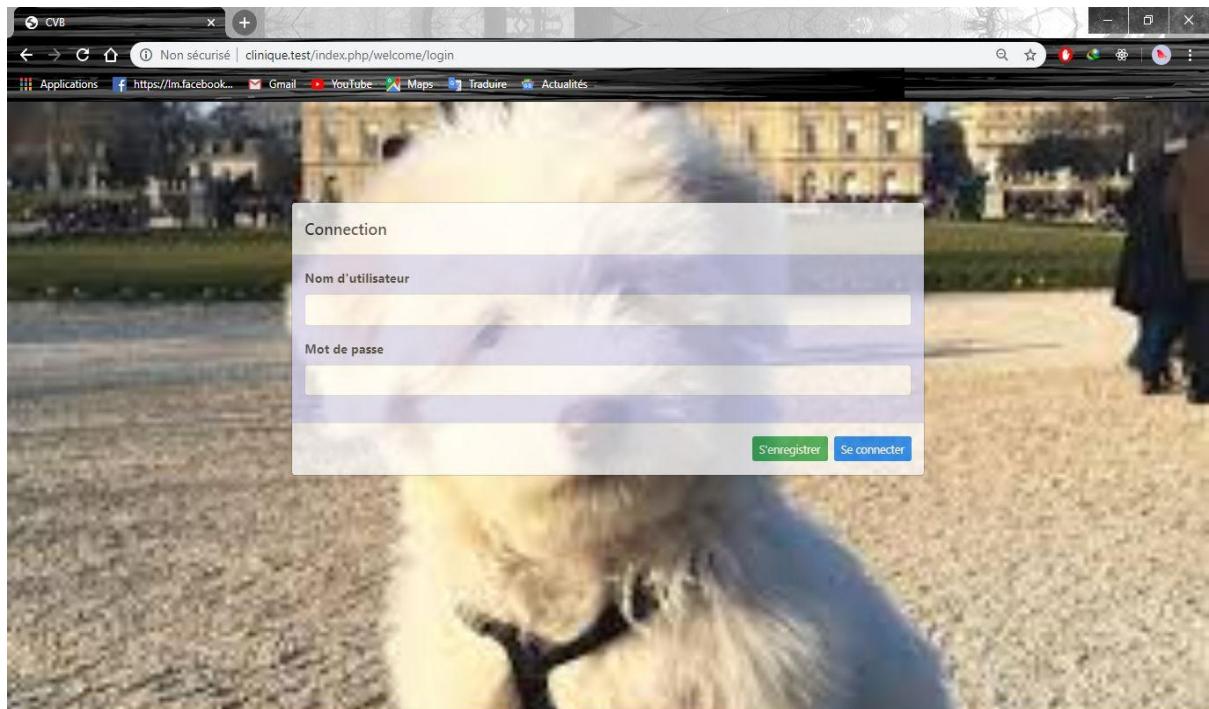


FIGURE 37: AUTHENTIFICATION

8.3.2 Page d'accueil

Après que l'utilisateur ait pu se connecter, il arrive alors sur la page d'accueil de l'application où des choix sur les tâches à faire s'affichent. La figure 38 montre la page d'accueil de l'application.

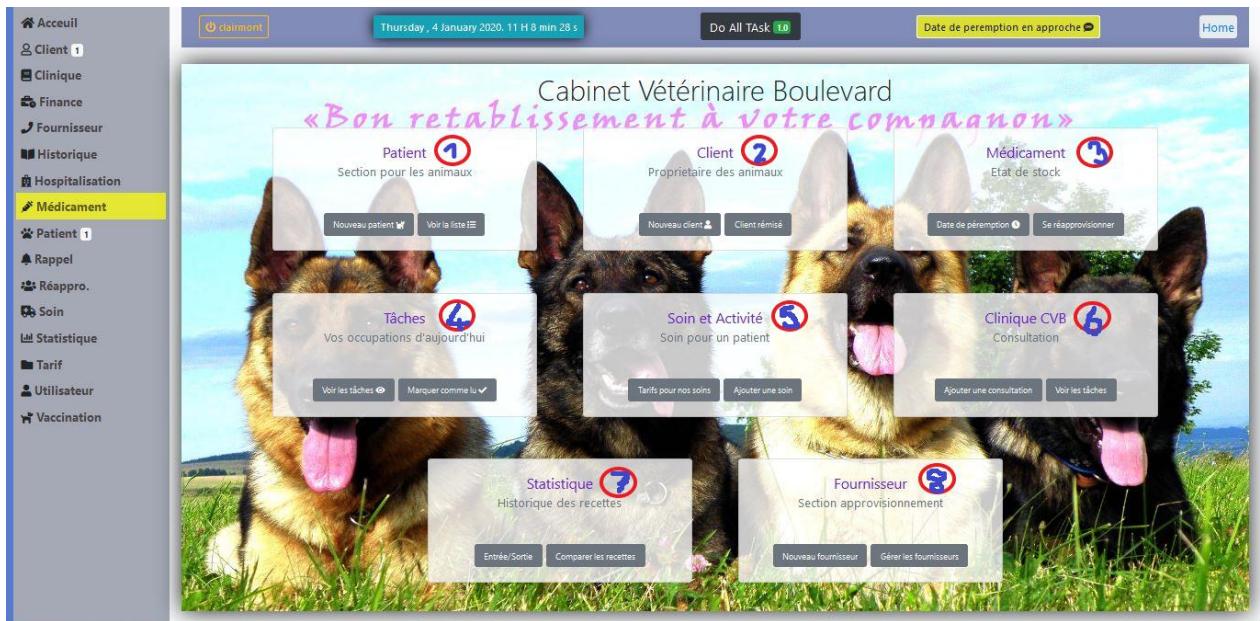


FIGURE 38: PAGE D'ACCUEIL DE L'APPLICATION

- * **1** : Entré vers les listes des patients ainsi que leur profil
- * **2** : Entré vers les propriétaires et leur profil
- * **3** : Entré vers l'état de stock
- * **4** : Entré vers les tâches à venir, donc des rappels
- * **5** : Entré vers les soins pour un patient
- * **6** : Entré vers une consultation dans la clinique pour le client
- * **7** : Entré vers les statistiques concernant la clinique
- * **8** : Entré vers les fournisseurs

Notifions que sur la gauche, on a un volet de menu qui nous permet de naviguer librement n'importe où sur l'application.

8.3.3 Consultation pour un patient

En cliquant sur le lien clinique CVB, on accède à la page de traitement pour un patient. La figure 39 montre la page de traitement pour un patient.

The screenshot shows a web-based medical consultation application. At the top, there's a header bar with the logo 'clairmont', the date 'Sunday, 9 February 2020, 8 H 57 min 10 s', and a button 'Do All Task 1.0'. Below the header, a blue bar displays the text 'PASSE UNE NOUVEAU' and 'Notification'. On the right, there are 'Home / Consultation' links.

The main form area has several input fields:

- A dropdown menu titled 'Qu'est ce que vous voulez faire ?' containing options: 'Consultation au Cabinet', 'Consultation à domicile', 'Hospitalisation', and 'Chirurgie'. The 'Chirurgie' option is highlighted with a blue number '1'.
- A 'Patient:' dropdown showing 'Amonne' with a blue number '2' next to it.
- A 'Motif:' field.
- A 'Suspicion:' field.
- An 'Anamnèse:' field.
- An 'Examen complémentaire' field.
- A dropdown menu 'Décision à prendre' with a blue number '3' next to it.
- Buttons at the bottom: 'Remplir la fiche du patient', 'Remise' (with a text input field), and 'CONSULTER' with a blue number '5' next to it.

FIGURE 39: TRAITEMENT POUR UN PATIENT

- 1 : c'est qu'on va faire, on le sélectionne sur cet onglet
- 2 : liste déroulante contenant les patients
- 3 : on sélectionne une décision puis les listes des médicaments vont alors s'afficher
- 4 : si on attribue une remise au client, c'est là qu'on le met
- 5 : passer la consultation et voir l'aperçu de la facture

Après avoir cliqué sur consulter, un aperçu de la facture s'affiche. La figure 40 montre cette étape.

The screenshot shows the invoice preview screen after clicking 'CONSULTER' in Figure 39. The top bar includes the date 'Sunday, 9 February 2020, 9 H 7 min 48 s', a 'Do All Task 1.0' button, and a 'Home / Consultation' link. A 'Date : 2020-02-09' is also visible.

The main area displays a table of treatments with their descriptions, quantities, unit prices, and totals:

| Description : | | | | |
|-------------------------|----------|-------|---------------|-----------|
| Traitements | Quantité | Unité | Prix unitaire | Total |
| Anestésies | 2 | cc | 10000 | 20000 |
| Anestésies | 1 | cc | 10000 | 10000 |
| Calcium | 3 | cc | 500 | 1500 |
| Chirurgie | | | | 250000Ar |
| Consultation au Cabinet | | | | 15000Ar |
| Somme | | | | 296500 Ar |

At the bottom, there are two buttons: '1 A payer maintenant' and 'Facturer chez nous 2'.

FIGURE 40: APERÇU DE LA FACTURE

- ✓ 1 : La somme payée à ce jour par le client

✓ 2 : Générer la facture

En cliquant sur facturer chez nous, on génère un PDF sur lequel est imprimé la facture.

8.3.4 Profil d'un patient

On peut alors consulter l'historique du patient et de son propriétaire pour voir ses dépenses. Les figures 41 et 42 montrent le profil pour la patiente Amonne qu'on vient de consulter.

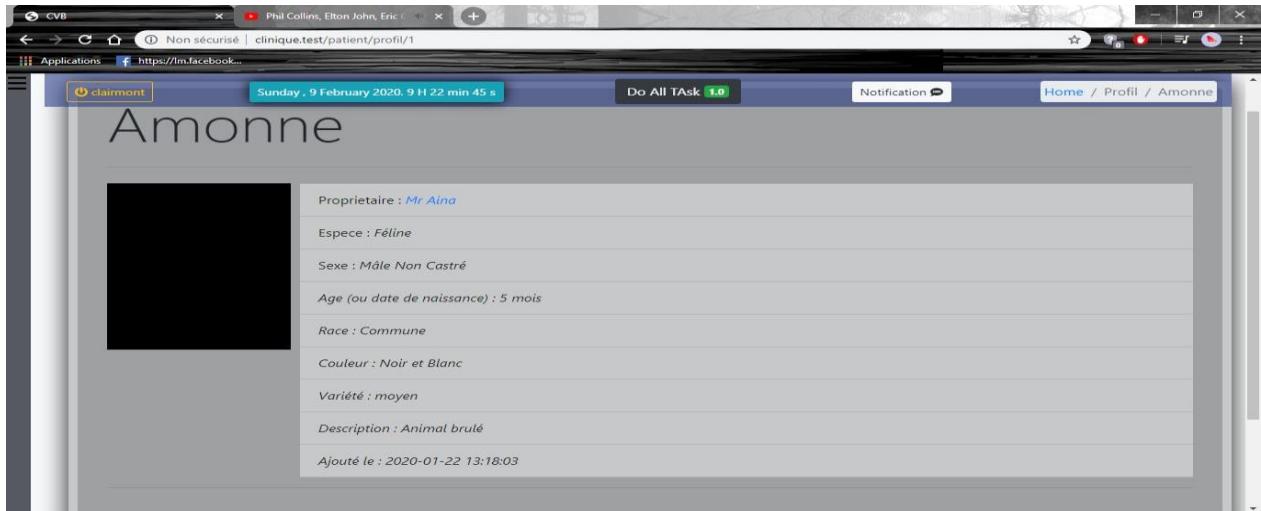


FIGURE 41: PROFIL D'AMONNE

A screenshot of a web-based application showing the medical history of 'Amonne'. The interface includes a header with a logo, date ('Sunday, 9 February 2020, 9 H 23 min 40 s'), and navigation links. The main content area shows a table of recent visits and treatments:

| Ses historiques en clinique | | | |
|-----------------------------|-------------------------|--|------------|
| Ses historiques de soin | | | |
| 110 | Consultation au Cabinet | | 2020-02-02 |
| 111 | Hospitalisation | | 2020-02-02 |
| 112 | Consultation au Cabinet | | 2020-02-02 |
| 113 | Toilettage simple | | 2020-02-02 |
| 114 | Chirurgie | | 2020-02-09 |
| 114 | Consultation au Cabinet | | 2020-02-09 |

FIGURE 42 : HISTORIQUE DE SES SOINS ET TRAITEMENTS

La figure 42 détaille ses récentes passages au cabinet. On y voit notamment sa consultation ainsi que sa chirurgie. En cliquant sur **Ses historiques en clinique**, on peut voir les traitements qu'il a reçu ce jour-là.

La figure 43 montre ces historiques.

| Ses historiques en clinique | | | | |
|-----------------------------|--------|--|---------------------|-----------------------------------|
| C111 | Amonne | | | |
| C112 | Amonne | | Voir le paramètre | HOSPITALISÉ TRAITEMENT 2020-02-02 |
| C113 | Amonne | | Voir le paramètre | TRAITÉ TRAITEMENT 2020-02-02 |
| C114 | Amonne | | 1 Voir le paramètre | TRAITÉ 2 TRAITEMENT 3 2020-02-09 |

Ses historiques de soin

FIGURE 43 : HISTORIQUES DES CONSULTATIONS AU CABINET

- 1 : Les paramètres du patient associées à cette consultation
- 2 : La décision prise pour cette consultation
- 3 : Les traitements reçus

En cliquant sur **Traitements**, on voit les médicaments reçus lors de ce passage comme représenté sur la figure 44.

| Traitement reçu | | | | |
|-----------------|------|--|--|--|
| Médicament | Dose | | | |
| Calmivet | 2 cc | | | |
| Castran | 1 cc | | | |
| Théracalcium | 3 cc | | | |

| Ses historiques de soin | | | | |
|-------------------------|--------|--|-------------------|-----------------------------------|
| C111 | Amonne | | | |
| C112 | Amonne | | Voir le paramètre | HOSPITALISÉ TRAITEMENT 2020-02-02 |
| C113 | Amonne | | Voir le paramètre | TRAITÉ TRAITEMENT 2020-02-02 |
| C114 | Amonne | | Voir le paramètre | TRAITÉ TRAITEMENT 2020-02-09 |

FIGURE 44 : MEDICAMENTS REÇUS LORS DU TRAITEMENT

8.3.5 Profil d'un client

Si on regarde ensuite le profil du propriétaire d'Amonne, qui est ici Mr Aina, on arrive sur la figure 45.

Nom : Mr Aina
Adresse :
Contact : 0337054298
Email :
Statut : Non remisé
Organisation :
Liste de ses patients : Amonne Pixi 1
Liste des dépenses : METTRE A JOUR CE CLIENT 2
DÉPENSE TOTALE FAITE AU SEIN DE NOTRE CLINIQUE : 586.500 Ar

FIGURE 45 : PROFIL POUR UN PROPRIÉTAIRE

- ❖ 1 : Listes des patients pour ce client
- ❖ 2 : Liste de ses dépenses
- ❖ 3 : Mise à jour du client

En cliquant sur **Liste des dépenses**, on voit tous les dépenses effectuées par le client, de ses restes à payer. La figure 46 montre cette liste.

| Numéro de la facture : | Dépense total : | Payée | Reste à payer : | Type : |
|---|-----------------|-----------|-----------------|--------------|
| 115 | 296.500 Ar | 0 Ar | 296.500 Ar | traitement 1 |
| 107 | 30.000 Ar | 0 Ar | 30.000 Ar | soin 2 |
| 113 | 30.000 Ar | 0 Ar | 30.000 Ar | soin 3 |
| 113 | 30.000 Ar | 15.000 Ar | 15.000 Ar | traitement 4 |
| 104 | 25.000 Ar | 25.000 Ar | 0 Ar | traitement 5 |
| 109 | 45.000 Ar | 45.000 Ar | 0 Ar | traitement 6 |
| 110 | 25.000 Ar | 25.000 Ar | 0 Ar | traitement 7 |
| DÉPENSE TOTALE FAITE AU SEIN DE NOTRE CLINIQUE : 586.500 Ar | | | | |

FIGURE 46: LISTE DES DÉPENSES

- 1 : s'il y a un reste à payer sur une éventuelle facture, on observe une icône sur cette partie. En cliquant sur cette icône, on est redirigé vers une page pouvant régler cette reste.

La figure 47 montre cette page.

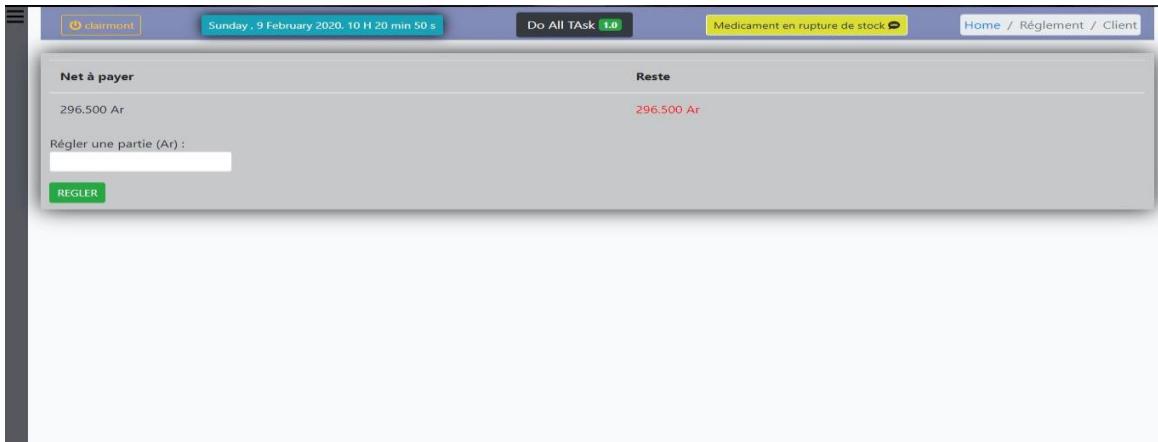


FIGURE 47: REGLEMENT DES RESTES A PAYER

8.3.6 Page finance

En réglant cette partie, on est redirigé vers la page finance qui affiche les entrées et les sorties au sein du cabinet. La figure 48 montre la page des finances.

| Finance | | | | | | | |
|-------------------------|--|------------------|---------------|-------------|------|---------------|-------|
| Côté fournisseur : | | N° de la facture | Fournisseur | Net à payer | Payé | Reste à payer | Réglé |
| Date limite de paiement | | 1 | Pharmacie CVB | 0 Ar | 0 Ar | 0 Ar | ✓ |
| 2020-02-02 | | 2 | Pharmacie CVB | 0 Ar | 0 Ar | 0 Ar | ✓ |
| 2020-02-02 | | 3 | Pharmacie CVB | 0 Ar | 0 Ar | 0 Ar | ✓ |

| Côté client : | | Client | Total | Payé | Reste à payer | Type | Date de paiement | Réglé |
|----------------------|---------|------------|-----------|------------|---------------|------------|------------------|-------|
| Numéro de la facture | | | | | | | | |
| 115 | Mr Aina | 296.500 Ar | 0 Ar | 296.500 Ar | traitement | 2020-02-09 | ✓ | |
| 103 | Mr Tely | 60.000 Ar | 0 Ar | 60.000 Ar | soin | 2020-02-02 | ✓ | |
| 107 | Mr Aina | 30.000 Ar | 0 Ar | 30.000 Ar | soin | 2020-02-02 | ✓ | |
| 113 | Mr Aina | 30.000 Ar | 0 Ar | 30.000 Ar | soin | 2020-02-02 | ✓ | |
| 106 | Mr Tely | 35.000 Ar | 35.000 Ar | 0 Ar | traitement | 2020-02-01 | ✓ | |
| 110 | Mr Aina | 25.000 Ar | 25.000 Ar | 0 Ar | traitement | 2020-02-02 | ✓ | |
| 104 | Mr Aina | 25.000 Ar | 25.000 Ar | 0 Ar | traitement | 2020-02-01 | ✓ | |

FIGURE 48 : PAGE FINANCE

Cette page s’occupe de la gestion des entrées/sorties monétaires du cabinet. On a deux sections, section Client et section fournisseur dont on peut aussi régler les dépenses inachevées.

L’application DATA est donc un outil complet pour la gestion d’une clinique vétérinaire puisqu’elle intègre directement la gestion des traitements et des flux monétaires. Cette application gère aussi les chiffres d’affaires d’un client en particulier, ce qui rend l’historique d’un client plus facile à gérer. L’approvisionnement des médicaments est aussi facilité et cette section permet de montrer aussi les étendues domaines de l’outil.

CONCLUSION ET PERSPECTIVES

La Clinique Vétérinaire Boulevard est une moyenne entreprise en pleine développement, avec des gros chantiers à faire au niveau du système d'information. Plus encore, avec l'arrivée des nouvelles technologies, la concurrence sera encore plus rude et le moindre avantage sur un système à un autre n'est pas à négliger. En passant par les recueils des informations jusqu'à sa modélisation en tant que donnée concret et utile à la clinique, nos méthodologies apprises durant les formations pédagogiques sont mises en maintes épreuves pour satisfaire les besoins très détaillés de l'utilisateur.

Durant le stage, une informatisation au niveau de la branche médicale et financière a été réalisé, permettant ainsi à ces deux branches d'éviter la perte d'une partie de leur information mais aussi de pouvoir compter sur une application pouvant à la fois organiser et traiter leurs données tout en concevant une nouvelle base de données qui répond grandement à leur besoin.

Pour la réalisation de ce projet, on s'est basé sur la méthode MERISE, une méthode très complet et répondant aux normes de gestion d'un projet. On a opté l'outil de développement VS Code qui supporte plusieurs types de langage, du Framework PHP « Codeigniter » qui intègre en elle une architecture MVC au sein de son application.

Les déficits rencontrés avant l'implantation du système ont été palliés au fur et à mesure que le projet avançait et la formation des utilisateurs ainsi que les phases de test, dans le processus de changement, ont été essentiel puisque ça nous a permis de cerner les failles et bugs du logiciel. Les problèmes liés à l'implémentation de ce logiciel sur un réseau local aient pu survenir, la résolution de ces derniers se sont faits étapes par étapes pour arriver à un résultat qui est une application qui marche et utilisable.

L'application est maintenant opérationnel et est déjà utilisé aux sein de la CVB, qui n'ont pas attendu longtemps pour porter ses fruits puisque les clientèles ont fait part de leur contentement et ont souhaités la possibilité d'une mise à jour de cette application pour encore l'améliorer un peu plus encore et ainsi atteindre la perfection.

Néanmoins, dans la perspective, on envisage de l'améliorer en intégrant un système de payement en ligne des dépenses ainsi qu'un système de caisse qui va permettre de savoir exactement l'argent actuellement en caisse mais aussi d'étendre le réseau concerné aux niveaux des filiales de CVB.

Le stage effectué nous a permis de nous familiariser avec le monde professionnel, des stress générés par les deadlines ainsi que les besoins incessants des utilisateurs, mais aussi d'avoir pu approfondir nos connaissances sur le développement web.

BIBLIOGRAPHIE

- 1 : Michel DIVINÉ, Parlez-vous MERISE ?, 1994
- 2 : Contributeur de Stack Overflow, Apprenez composer-php
- 3 : Serge TAHÉ, Méthodologie de développement MVC d'une application PHP, Janvier 2004
- 4 : Dominique DOLÉ, Bases de données avec MySQLWorkbench, 10 Décembre 2014

WEBOGRAPHIE

- 1 : Site officiel de Vetup : « vetup.com »
- 2 : Comparaison des Framework PHP « <http://socialcompare.com/fr/comparison/php-frameworks-comparison> »
- 3 : Site web de Laragon : « <https://laragon.org> »
- 4 : Visual Studio Code : «<https://code.visualstudio.com/>»
- 5 : Codeigniter, site officiel : «<https://codeigniter.com>»
- 6 : Téléchargement de la console de git : «<https://git-scm.com/downloads>»
- 7 : Codeigniter 3 translations : « <https://github.com/bcit-ci/codeigniter3-translations> »
- 8 : Html2PDF : « <https://html2pdf.fr/> »

GLOSSAIRE

Cardinalité : nombre minimum et maximum de fois qu'une occurrence d'entité participe aux occurrences de l'association

Composer : c'est un gestionnaire de dépendance entre application et librairies, mis en place avec l'arrivée de PHP5 pour gérer les composants, bibliothèques et modules pour chaque projet.

Dépendances : une dépendance exprime les relations entre paquets : de prérequis d'inclusion conditionnelle, des post-requis d'inclusion conditionnelle...

Déploiement : concept pour désigner la mise en production d'un système ou application bien testé et approuvé.

Entité : objet, une chose concrète ou abstraite qui peut être reconnue distinctement et qui est reconnue par son unicité.

Framework : c'est un cadre de travail qui a pour objectif de simplifier le travail des développeurs informatiques en leur offrant une architecture prête à l'emploi pour ne plus repartir à zéro à chaque nouveau projet.

Online : désigne une application qui requiert une connexion internet pour fonctionner.

Paramètre : ensemble des paramètres concernant un patient, composé de son TRC, Fréquence Cardiaque, Poids, Taille, Fréquence Respiratoire...

Payé/Impayé : montre l'état d'une facture après que celle-ci a été générée.

Soin : ensemble des traitements faits à un client, un soin se caractérise par le fait que les médicaments utilisés seront facturés ou pas.

RESUME

Dans ce projet, on a conçu une application web nommée DATA, en partant de la conception de la base de données pour aboutir à un outil de gestion d'une clinique vétérinaire. Cette base de données a été conçue conformément aux demandes des services dans la clinique, et combinée à l'application, facilite la gestion complète des flux d'information du CVB. On a pu pallier les problèmes liés au suivi des informations, les historiques de traitement reçus par un patient, l'historique des payements, aux statistiques sur les jours les plus productives de la clinique.

Ce stage a été une bonne occasion de se familiariser avec tous ce qui est du monde professionnel et des demandes clientèles.

Mot clé : clinique, cabinet vétérinaire, consultation, statistique journalière

ABSTRACT

In the present project, we design and implement a database for CVB Ankazomanga, SMEs whose main occupation is to satisfy their customer base by offering services worthy of their reputation. We have implemented a new automated information system, which overcomes the problems associated with the current system in place. Then, we created a new web app according to the needs of the users on the recording as well as viewing patient data but also the state of stock and monetary flow of the clinic.

This internship therefore allowed us to realize the needs of users and the professional world, to apply the knowledge acquired during pedagogical studies, to acquire more experience on web development and already to prepare to deal with these requests once our circus at ENI finished.

Keywords: clinic, veterinary practice, consultation, daily statistics.