RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 55 02 80 70
http://www.ifri-uac.bj Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Systèmes informatiques et Réseaux informatiques

Présenté par :
Mahugnon Bénoît Romario **GANDONOU**

Developpement d'une application de gestion des données biomédicales dans un laboratoire

Sous la supervision :
Pierre Jérôme **ZOHOU**

Membres du jury :

AGOSSOU Carlos	Docteur	UCAO	Président
HOUZANME Vladimir	Ingénieur	IFRI	Examinateur
ZOHOU Pierre Jérôme	Docteur	IFRI	Rapporteur

Année Académique : **2022-2023**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Table des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire 1

Introduction 2

1 Revue de littérature 4

2 Analyse, Conception,Choix techniques 9

3 Réalisation de l'application 20

Conclusion 30

Table des matières 33

Dédicace

A

Mon père **Robert GANDONOU**

Ma mère **Pauline MILIDO**

Mes frères et soeurs

Remerciements

Nos sincères et profonds remerciements :

- Au Professeur **Eugène C. EZIN**, Directeur de l'Institut de Formation et de Recherche en Informatique (IFRI), pour son dévouement pour le développement de l'informatique dans notre pays.
- Au Professeur **Gaston EDAH**, Directeur Adjoint de l'IFRI, Chargé des affaires académiques, pour son sens de l'écoute et de la compréhension et à toute l'équipe dirigeante de IFRI.
- A Monsieur **Pierre Jérôme ZOHOU**, enseignant à l'IFRI, pour avoir librement et spontanément accepté de diriger ce projet de mémoire au cours duquel il a fait preuve d'une humilité inattendue et d'une disponibilité hors paire. Qu'il trouve à travers ce travail le fruit de ses innombrables disponibilités.
- A tous les enseignants de lIFRI, pour leur sens de partage de leurs connaissances avec les étudiants que nous sommes.
- A Monsieur **Josias ZOUNMA**, Directeur du laboratoire de la Clinique "Vie abondante" pour avoir mis à notre disposition les informations nécessaires à la réalisation de ce projet.
- A ma famille, pour son soutien et son amour indéfectible à mon égard.
- A tous mes camarades de promotion, pour leur sens de fraternité.
- A tous ceux qui, de prêt ou de loin, ont contribué à l'aboutissement du présent projet, qu'il trouve à travers celui un pas vers l'objectif très grand visé.

Résumé

Le laboratoire de la clinique "Vie Abondante" connaît quelques difficultés dans la gestion des données biomédicales telles la saisie des résultats avec erreur et la difficulté de retrouver un résultat d'examen sorti du laboratoire après un temps donné. La mise en place d'un système de gestion des données biomédicales en utilisant une application web implémentée avec Python (framework DJANGO),Jquery, HTMX et utilisant une base de donnée PostgreSQL sera profitable au laboratoire.

L'une des fonctionnalités clés de cette application réside dans la centralisation des données. Non seulement cela facilite l'accès aux informations critiques, mais cela garantit également la traçabilité complète des échantillons, offrant un niveau de confiance élevé dans la qualité des résultats. L'automatisation des processus comme la génération des rapports personnalisés permet d'alléger le travail des biologistes. Cette automatisation libère du temps et des ressources, favorisant une concentration accrue sur la manipulation des échantillons. L'application permettra de suivre les activités d'un personnel du laboratoire, d'enregistrer de nouveaux personnels, de nouveaux patients selon le cas. L'objectif de ce système est effet d'informatiser la gestion des données biomédicales du laboratoire.

Mots clés : gestion, accès aux informations, automatisation des processus.

Abstract

The laboratory of the "Vie Abondante" clinic experiences some difficulties in the management of biomedical data such as entering results with error and the difficulty of finding an examination result taken out of the laboratory after a given time. The establishment of a biomedical data management system using a web application implemented with Python (DJANGO framework), Jquery, HTMX and using a PostgreSQL database will be beneficial to the laboratory.

One of the key features of this application is the centralization of data. Not only does this facilitate access to critical information, but it also ensures full sample traceability, providing a high level of confidence in the quality of results. The automation of processes such as the generation of personalized reports makes it possible to reduce the work of biologists. This automation frees up time and resources, promoting increased focus on sample handling. The application will make it possible to follow the activities of laboratory staff and to register new staff and new patients as appropriate. The objective of this system is to computerize the management of biomedical laboratory data.

Key words: management, access to information, process automation

Table des figures

1.1 Labbook interface administrateur[3] 5
1.2 Labbook interface Biologiste 5
1.3 ProLab Lis Accueil [6] 6
1.4 GestLab : Enregistrement d'une demande d'examen [5] 7

2.1 Diagramme des cas d'utilisation du système 10
2.2 Diagramme de classe de l'application 11
2.3 Diagramme d'activité du cas d'utilisation : Ajouter une demande d'examens . 12
2.4 Diagramme d'activité du cas d'utilisation : Imprimer les résultats d'une fiche d'examens 13
2.5 Diagramme de séquence du cas d'utilisation : Ajouter une demande d'examens 14
2.6 Diagramme de séquence du cas d'utilisation : Imprimer les résultats d'une fiche d'examens 14
2.7 Diagramme de séquence du cas d'utilisation : Suivre les activités d'un personnel 15
2.8 L'architecture MVT [4] 16

3.1 Page d'authentification de notre application 20
3.2 Page d'accueil (administrateur) de notre application 21
3.3 Page d'accueil (biologiste) de notre application 21
3.4 Liste des demandes d'examens ordonnées par la date 22
3.5 Page d'enregistrement d'un patient 23
3.6 Liste des patients 23
3.7 Liste des fiches d'examens d'un patient 24
3.8 traitement d'une demande d'examens 24
3.9 La liste du personnel 25
3.10 Les activités d'un biologiste 25
3.11 Rapport des examens éffectués 26
3.12 impression d'un résultat sous forme pdf 27

Liste des tableaux

1.1 Tableau récapitulatif de quelques solutions existantes présentées 8

2.1 Synthèse des outils utilisés 19

Liste des acronymes

AFD :

Agence Française de Développement 5

CSS :

Cascading Style Sheet 15

HTML :

HyperText Markup Language 15

LIMS :

Laboratory Informations Management System 5, 7, *Glossaire :* LIMS

MVC :

Model-View-Controller 13, 26

MVT :

Model-View-Template 13, 14, 26

ORM :

Object Relational Mappping 16

UML :

Unified Modeling Language 11, 14

Glossaire

Framework : C'est un espace de travail, qui permet de mettre en place la base et l'architecture d'une application. 14, 16

LIMS : Système de Gestion des Information du Laboratoire vii

SGBD : Système de Gestion des Bases de Données 15

Introduction Générale

L'analyse biomédicale est une discipline qui étudie les échantillons biologiques, tels que le sang, l'urine ou les tissus, pour évaluer la santé d'un individu. Elle permet de diagnostiquer des maladies, surveiller le traitement médical et évaluer la fonction des organes. Ainsi la bonne gestion des données d'analyse médicale pourrait améliorer beaucoup de choses dans le domaine de la santé.

Problématique

Très souvent, le laboratoire d'analyse médicale de la clinique "La vie abondante" fait face à de nombreuses difficultés. Parmi ces difficultés on peut identifier la saisie avec erreur des résultats des patients après manipulation des échantillons, la difficulté de trouver après livraison certains résultats redemandés par les patients, une manière simple de faire le points des examens médicaux réalisés dans une période donnée. Plusieurs mesures ont été prises pour régler ces problèmes.

Les données concernant les patients, les examens et les résultats d'examens sont enregistrés dans des simples fichiers word ou dans des fichiers excel, ce qui ne permet pas une traçabilité. Il se pose la question de mise en place d'une application capable de gérer les problèmes courant dudit laboratoire.

Contexte

Parmi les sous domaines de la santé, nous avons la médecine clinique, la santé publique, les soins infirmiers,la pharmacie, la biologie médicale et autres. Tous ses sous-domaines sont reposés sur la biologie médicale. Ce secteur est l'un parmi les secteurs principaux d'un centre de santé, mieux gérer le laboratoire peut améliorer le domaine de la santé et même hausser le chiffre d'affaire dans un centre santé. Le laboratoire concerné est le laboratoire de la Clinique Vie abondante située à Agla.

Objectifs

Notre projet a pour objectif principal de mieux gérer les données dans un laboratoire d'analyse médicale. De façon spécifique les objectifs poursuivis en proposant cette solution sont les suivants :

- Gestion numérique des résultats des patients ;
- Génération des rapports des travaux effectués sur une durée donnée ;
- Centralisation des données du laboratoire.

Organisation du mémoire

Le présent mémoire, point de nos travaux, comporte trois chapitres. Le premier, présente une revue de littérature sur les outils de gestion de laboratoire existants. Le deuxième, présente les choix techniques opérés en vue de la conception et de la réalisation de la solution proposée. Quant au troisième, il fait une analyse critique des résultats issus de nos simulations.

Chapitre 1

Revue de littérature

Introduction

Pour la gestion des laboratoires d'analyses médicales, plusieurs applications ont été développées dans le monde. Ces applications sont conçues pour faciliter les activités dans le laboratoire, c'est-à-dire les prélèvements, le payement, la gestion des personnels et autres. Parmi ces applications on peut citer quelques uns et leurs limites.

1.1 Clarification conceptuelle

Avant de citer les applications existantes nous allons donner la définition de quelques mots utilisés dans un laboratoire

Echantillon : En analyse biomédicale, un échantillon fait référence à un petit fragment de tissu (la peau), de liquide biologique ou d'une substance biologique (le sang, l'urine) prélevé sur un patient ou un sujet d'étude. Ces échantillons sont utilisés pour des analyses en laboratoire afin de diagnostiquer des maladies, de surveiller l'état de santé d'un individu.

Examen : Un examen désigne une évaluation systématique et méthodique effectuée par un biologiste ou un technicien de laboratoire pour obtenir des informations sur l'état de santé d'un patient. Parmi ces examens nous pouvons citer : La Numération Formule Sanguine(NFS), la glycémie, la calcémie...

Automate : Un automate est un équipement ou un système automatisé utilisé pour effectuer des analyses et des tests sur des échantillons biologiques de manière efficace et précise. Nous avons les automates comme : les analyseurs de chimie clinique ; les systèmes d'hématologie...

1.2 Labbook

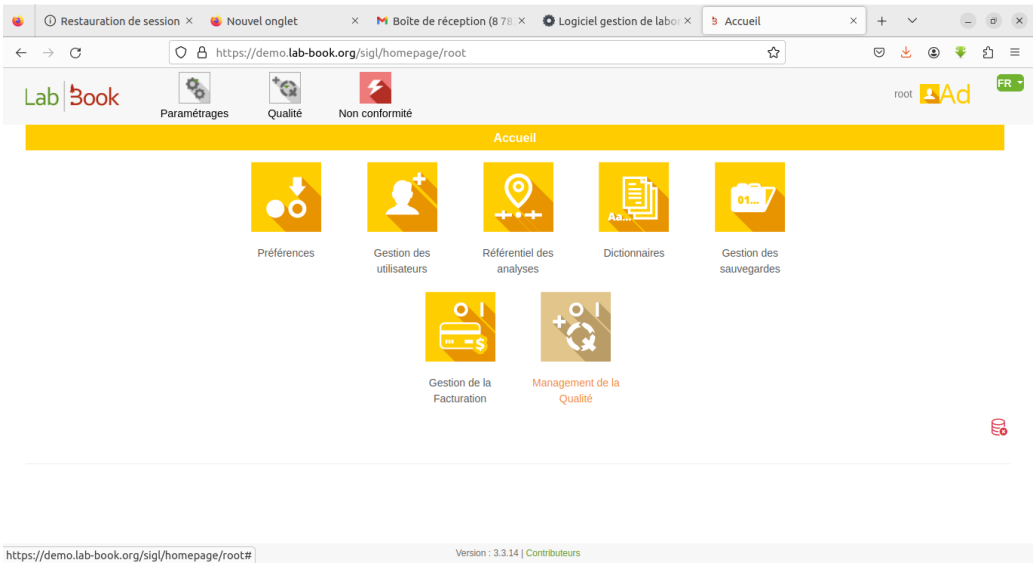


FIGURE 1.1 : Labbook interface administrateur[3]

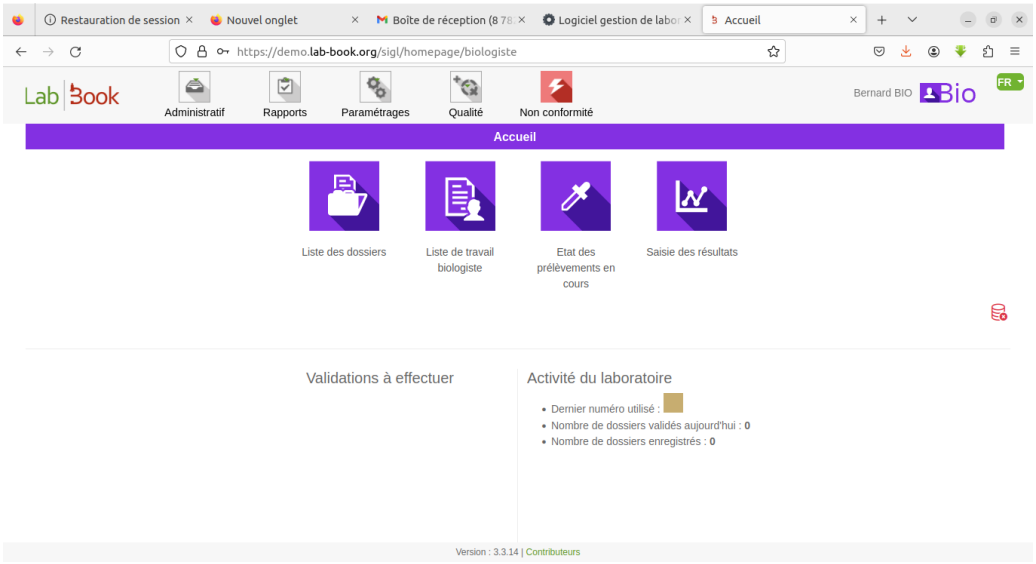


FIGURE 1.2 : Labbook interface Biologiste

Labbook est un LIMS créé par la Fondation MERIEUX en France, principalement pour les pays en voie de développement. Le logiciel a été cofinancé par l'Agence Française de Développement (AFD). Le logiciel aide les laboratoires de biologie médicale à ameliorer la gestion et le contrôle de leurs données, éléments indispensables pour garantir des analyses de qualité. Labbook a été testé, éprouvé et amélioré depuis la sortie de la première version en 2010[3].

Voic les fonctionnalités de Labbook :

- Recueil d'informations cliniques ;

- Saisie des données techniques et biologiques ;
- Validations des résultats ;
- Impression des feuilles des résultats ;
- Archivage des données ;
- Générations des rapports.

Labbook répond correctement aux attentes du laboratoire, mais Labbook est conçu seulement pour le système d'exploitation Ubuntu, installer Labbook c'est installer Ubuntu d'office. Les personnels du laboratoire ne sont pas habitués au système Ubuntu car ils auront besoin de Microsoft office pour effectuer d'autre travaux .

1.3 ProLab Lis

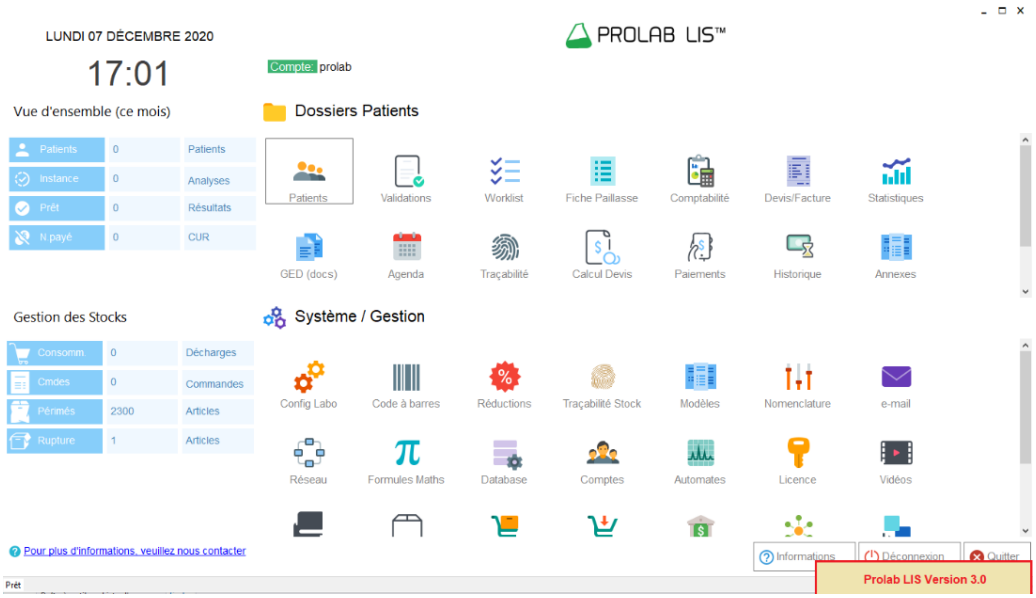


FIGURE 1.3 : ProLab Lis Accueil [6]

ProLab est un progiciel de gestion de laboratoire d'analyses médicales multi-modulaire (applicatif) contenant plus de 40 Applications dans sa version intégrale avec un système de données Microsoft SQL Server garantissant un niveau de sécurité élevé pour les données. ProLab a été conçu en Algérie. ProLab offre plusieurs fonctionnalités comme :

- de gestion de laboratoire d'analyses médicales multi-modulaire (applicatif) contenant plus de 40 Applications dans sa version intégrale avec un système de données Microsoft SQL Server vous garantissant un niveau de sécurité élevé pour vos données ;
- Gestion des dossiers patients, avec pointage QR (lecteur QR), Gestion de l'accueil et la file d'attente ;

- Intégration et Communication avec les Machines
- Échange de données électroniques (et réseaux), avec système sauvegardes/restaurations flexible ;
- Facturation, Devis, Relevés, Assurances, Système de tarification, Caisse intégrée avec les modèles de mise en page ouverts pour la personnalisation.

ProLab répond aussi aux attentes du laboratoire mais ProLab est un logiciel Payant qui offre beaucoup plus de fonctionnalités dont n'avait pas besoin le laboratoire. On peut avoir :

1. Connexion bidirectionnelle avec un HIS (hospital management system) via HL-7 (Health Level 7) ;
2. Étalonnage et maintenance des instruments ;
3. Conformité (Suivre les normes réglementaires qui affectent le laboratoire) ;

1.4 GestLab

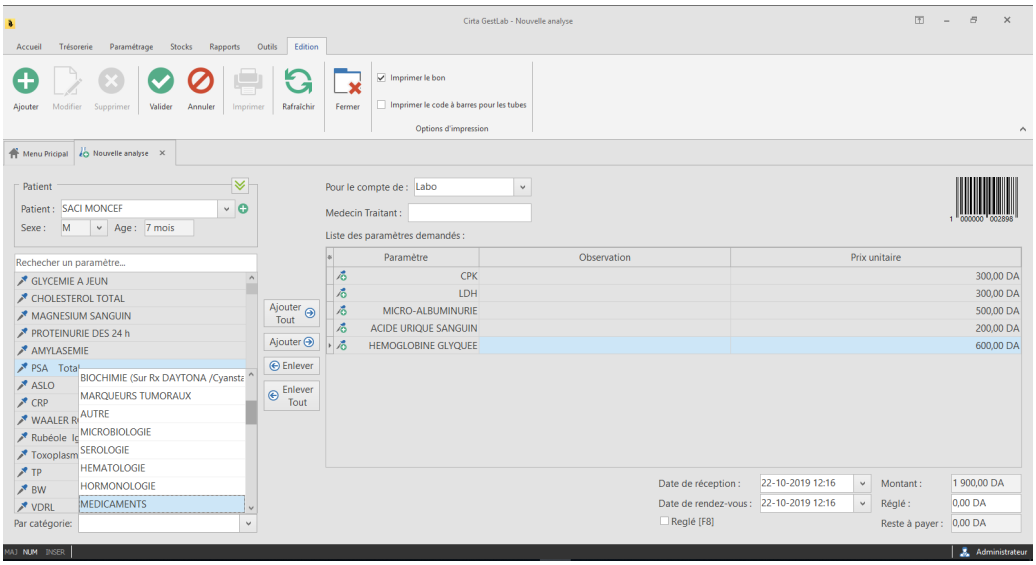


FIGURE 1.4 : GestLab : Enregistrement d'une demande d'examen [5]

GestLab est un LIMS développé par la société CIRTA IT basée en Algérie. GestLab tout comme Prolab est un logiciel avec une prise en main facile, il est personnalisable intégrant la gestion des profiles avec plus de 600 droits d'accès pour chaque utilisateur. Voici les fonctionnalités de GestLab :

- La gestion des échantillons par code à barres,
- Gestion du stocks
- Sécurité (chaque utilisateur aura son mot de passe)

- Gestions des tiers (Recevoir des analyses à sous-traiter pour d'autre laboratoire).
- prise en charge des automates (les automates de cytologie, de biochimie etc)
- personnalisation de l'impression

GestLab est aussi un logiciel payant offrant beaucoup de fonctionnalités qui ne sont pas obligatoires au laboratoire. GestLab comporte plus de 40000 paramètres, donc la personnalisation ne sera pas facile. Payer un logiciel comme GestLab ne sera pas bénéfique au laboratoire que ne recherche que quelques fonctionnalités.

Tableau récapitulatif

Le tableau 1.1 suivant montre le point fait sur les solutions existantes.

Logiciel	Licence	Force	Faiblesse
Labbook	Open source, Libre	Répond à nos attentes	Conçue pour le système de noyau Linux
ProLab LIS	payant(2000€)	Comporte plus de 40 applications	Configuration très complexe et utilise la connexion internet pour mieux fontionner
GestLab	payant	Répond à nos attentes	Configuration très complexe

TABLE 1.1 : Tableau récapitulatif de quelques solutions existantes présentées

Aucune des solutions existantes ne permet au laboratoire d'imprimer les résultats provenants des différentes institutions en collaborations avec le laboratoire avec leur entête.

Conclusion

Pour la gestion des informations d'un laboratoire médical, plusieurs applications ont été développées. Parmi ces applications, il y a des applications purement webs, le laboratoire doit avoir une connexion internet avant d'utiliser l'application, d'autres sont des applications complexes dont la paramétrisation et la personnalisation serait fastidieux.

Le projet actuel est une application qui sera utilisé en local, et qui réponds au attentes du laboratoire.

Chapitre 2

Analyse, Conception,Choix techniques

Introduction

Pour réaliser notre application, il est nécessaire de formaliser les étapes préliminaires de notre développement afin de rendre celle-ci plus fidèle aux besoins des utilisateurs. Dans ce chapitre, nous ferons l'analyse, la conception de notre plateforme et enfin nous expliquerons le choix des technologies utilisées pour la réalisation de notre plateforme.

2.1 Analyse des besoins

L'analyse des besoins est la première étape à franchir dans la conception de notre plateforme. Cette analyse définira de façon succincte les différentes fonctionnalités de notre application. L'application est conçue pour trois type d'utilisateur : l'administrateur, le biologiste et le secrétaire.

1. Le secrétaire peut :
    - Enregistrer, modifier et supprimer un patient
    - Ajouter une demande d'examens
    - imprimer une fiche d'examens sous un format pdf
    - Faire sortir un rapport mensuel ou annuel
2. Le biologiste peut faire tout ce que fait le secrétaire et en dehors de ça il peut :
    - traiter et supprimer une demande d'examens
3. L'administrateur peut faire tout ce que fait le biologiste et en dehors de ça il peut :
    - Il peut ajouter un personnel du laboratoire, désactiver et activer le compte d'un personnel
    - Suivre les travaux de chaque personnel

2.2 Conception

Cette phase permettra de définir de manière non ambiguë, en utilisant un langage de modélisation le fonctionnement futur du système, afin d'en faciliter la réalisation. Afin de modéliser les fonctionnalités de notre plateforme, nous avons porté notre choix sur le langage UML, un langage de modélisation visuelle standard destiné à être utilisé pour spécifier, visualiser et documenter des modèles de systèmes logiciels, y compris leur structure et leur conception de manière à répondre à toutes ses exigences.

2.2.1 Identification des acteurs

En UML, on entend par acteur toute idéalisation d'un rôle joué par une personne externe, un processus ou une entité qui interagit avec le système. Il existe deux types d'acteurs, à savoir les acteurs principaux, qui utilisent directement les fonctionnalités du système et les acteurs secondaires qui contribuent à la réalisation d'un cas d'utilisation. Dans le cadre de notre système nous avons retenu :

- La secrétaire
- Le biologiste
- l'administrateur

2.2.2 Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme qui résume la façon dont les utilisateurs interagissent avec un système. Il permet de recueillir, d'analyser et d'organiser les besoins et de recenser les grandes fonctionnalités d'un système. Grâce au diagramme de cas d'utilisation, nous aurons une vue globale du comportement fonctionnel du système.

La figure 2.1 présente le diagramme des cas d'utilisation du système.

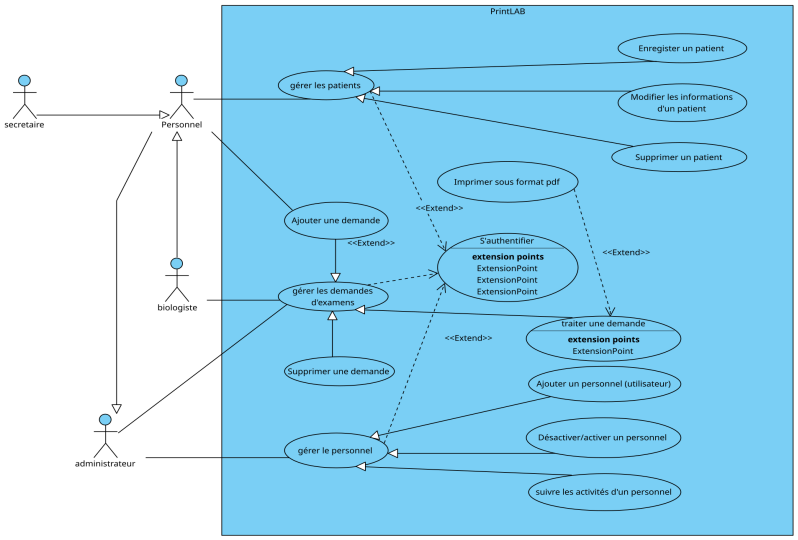


FIGURE 2.1 : Diagramme des cas d'utilisation du système

NB : Avant d'avoir accès au système les personnels doivent forcément s'authentifier

2.2.3 Le diagramme de classe

Le diagramme de classe est un diagramme UML utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Grâce au diagramme de classe, nous illustrerons les modèles de données pour notre système, quel que soit le degré de complexité.

la figure 2.2 nous présente le diagramme de classe du système.

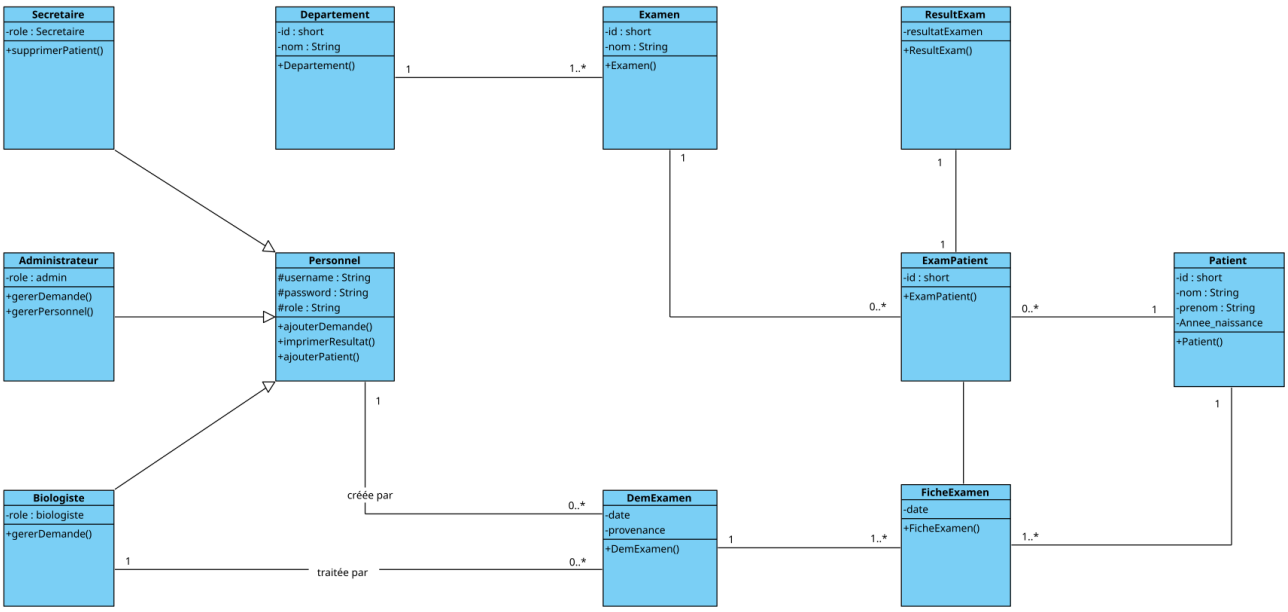


FIGURE 2.2 : Diagramme de classe de l'application

2.2.4 Diagramme d'activité

Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables. Le diagramme d'activité est également utilisé pour décrire un flux de travail.

la figure 2.3 nous présente le diagramme d'activité du cas d'utilisation : Ajouter Fiche examens

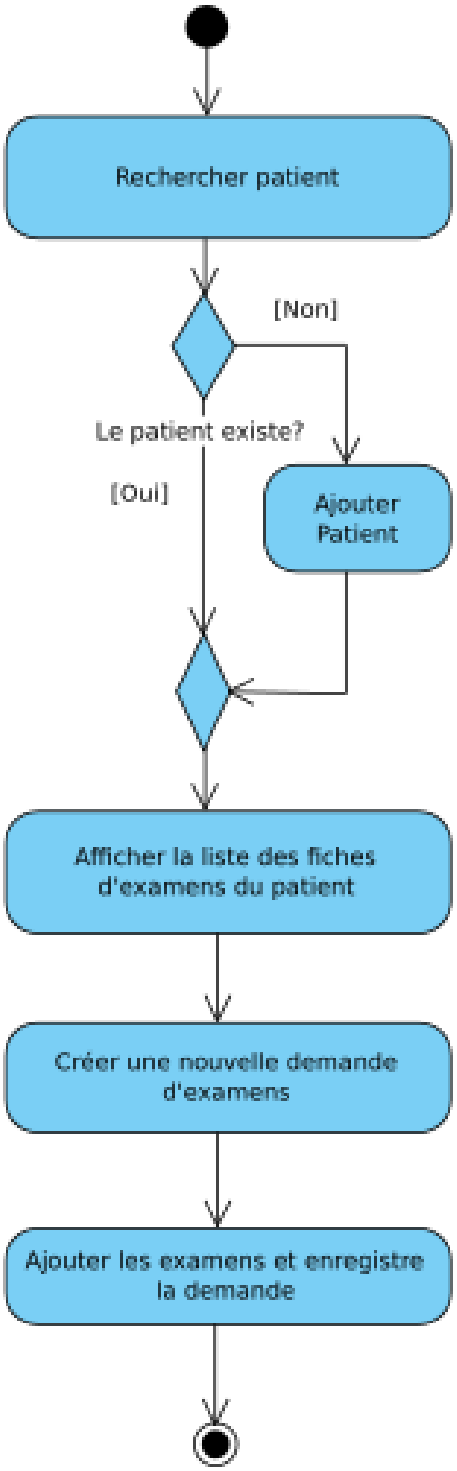


FIGURE 2.3 : Diagramme d'activité du cas d'utilisation : Ajouter une demande d'examens

la figure 2.4 nous présente le diagramme d'activité du cas d'utilisation : Imprimer les résultats d'une fiche d'examens

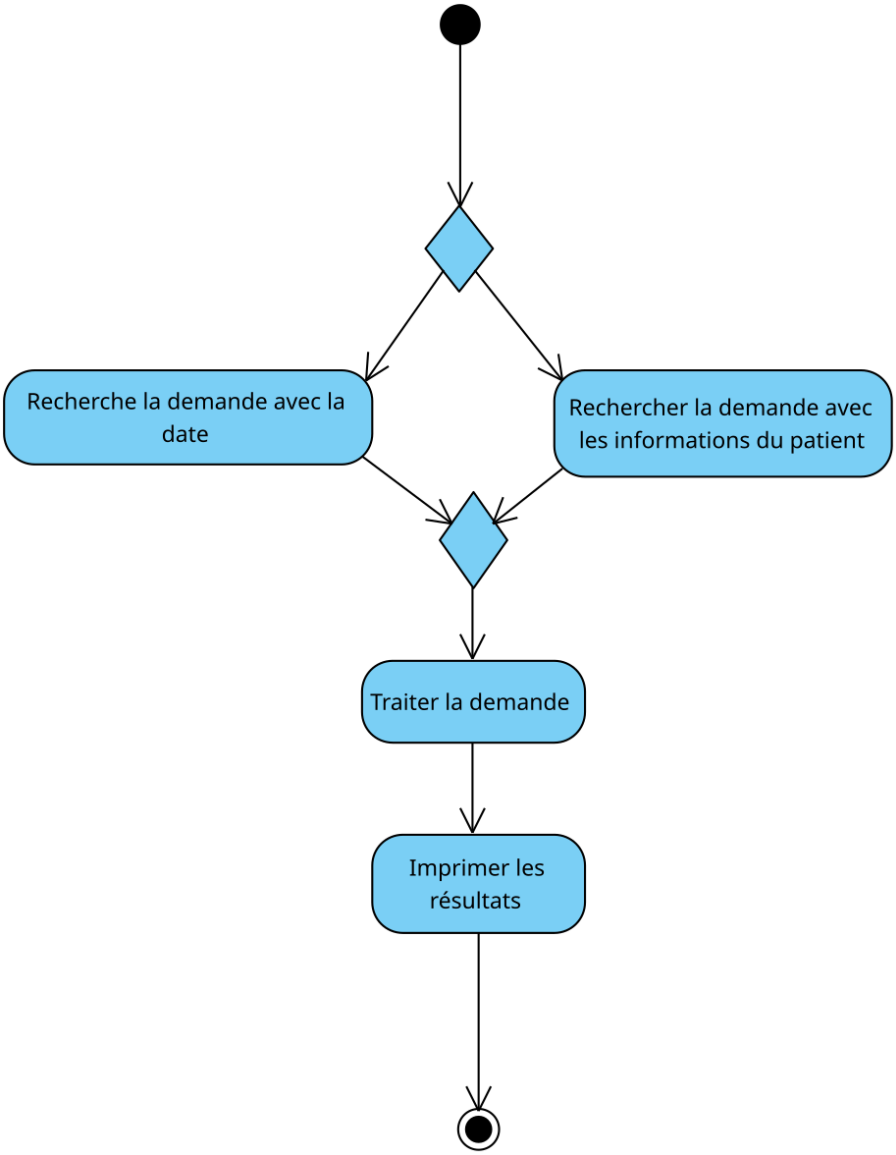


FIGURE 2.4 : Diagramme d'activité du cas d'utilisation : Imprimer les résultats d'une fiche d'examens

2.2.5 Diagramme de séquence

Le diagramme de séquence décrit comment les éléments du système interagissent entre eux et avec les acteurs. Il présente la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Il permet de modéliser le déroulement logique d'une fonctionnalité.

La figure 2.5 montre le diagramme de séquence du cas d'utilisation : Ajouter une demande d'examens

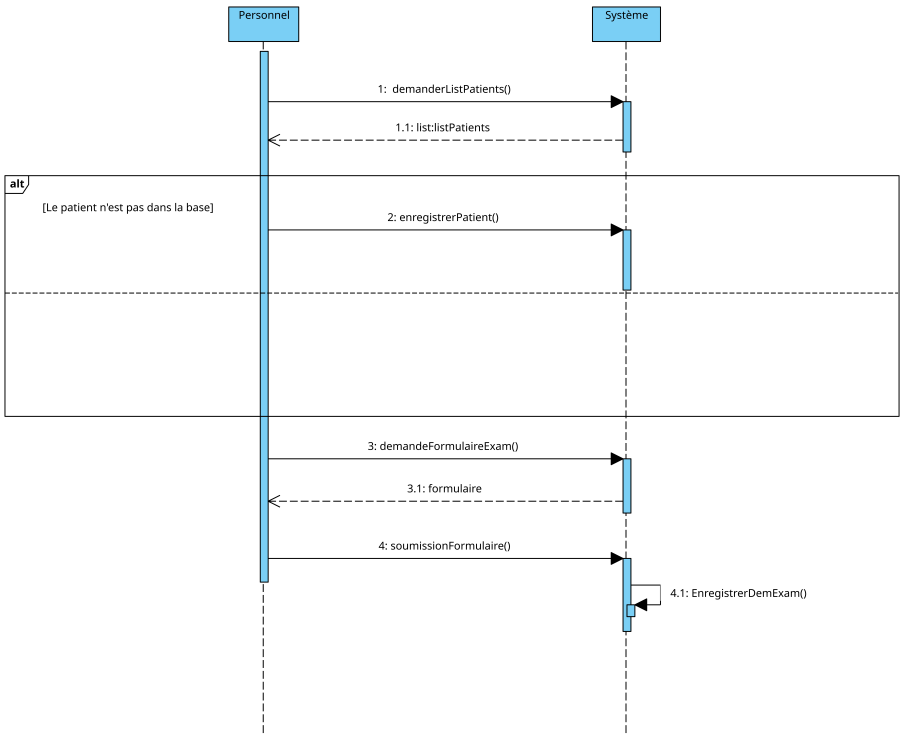


FIGURE 2.5 : Diagramme de séquence du cas d'utilisation : Ajouter une demande d'examens

La figure 2.6 montre le diagramme de séquence du cas d'utilisation : Imprimer un résultat

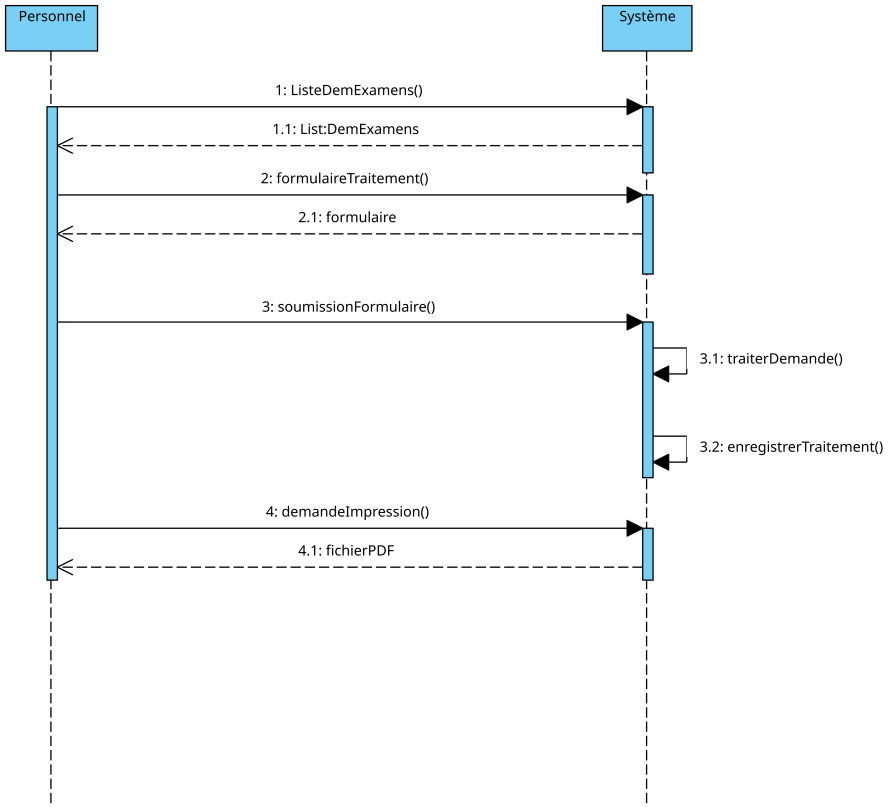


FIGURE 2.6 : Diagramme de séquence du cas d'utilisation : Imprimer les résultats d'une fiche d'examens

La figure 2.7 montre le diagramme de séquence du cas d'utilisation : Suivre les activités d'un personnel

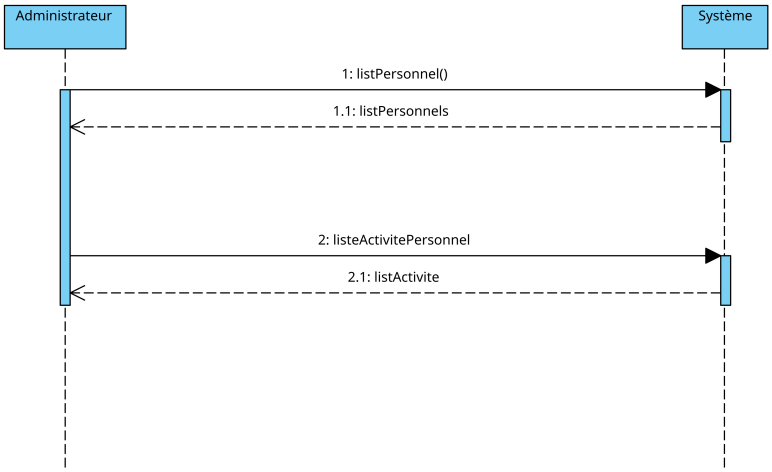


FIGURE 2.7 : Diagramme de séquence du cas d'utilisation : Suivre les activités d'un personnel

2.3 Choix techniques

Dans cette partie, nous présenterons les technologies utilisées ainsi que les outils qui nous ont permis de réaliser notre application.Aujourd'hui l'application est conçue pour une personne, mais les utilisateurs peuvent s'étendre, donc nous avons opté pour une application web qui sera utilisé comme un application desktop.

2.3.1 L'architecture MVC et MVT

Le MVC est un modèle de développement utilisé par les webmaster pour créer des applications web. Il consiste en la séparation des tâches à effectuer selon les différentes couches de développement.

Le MVC est donc décomposé en trois couches :

- Modèle : le modèle représente une information stockée quelque part, dans une base de données ou simplement dans un fichier comme dans le cas de SQlite.
- Vue : La vue est la présentation de cette information à l'utilisateur. Il s'agit donc de ce que l'utilisateur voit, quelles actions effectue-t-il sur l'interface ?
- Contrôleur : Il prend en charge tout les éléments de l'utilisateur comme une requête, interroge le modèle et puis se charge de renvoyer le résultat à la vue correspondante afin que l'utilisateur aperçoive le résultat.

Le MVT est un modèle de développement créé par DJANGO un Framework web du langage de programmation Python. Il n'y a pas beaucoup de différences entre le MVC et le MVT.

Pour le système MVT il n'y a pas de contrôleur et la *Vue* ne représente pas la présentation des données affichées. En d'autres termes, elle s'occupe des données à afficher et non de quelle façon les affichées.

La *Vue* est simplement une fonction appelée par une adresse URL particulière. Alors que la présentation ou l'affichage concrète se fait dans une couche à part appelée *Template*. Ainsi, le contenu est séparé de la présentation.

Dans le MVT on a alors :

- Modèle : le modèle décrit la structure de données (base de données)
- Vue : la *Vue* contrôle ce que l'utilisateur voit ;
- Template s'occupe de la présentation de l'information.

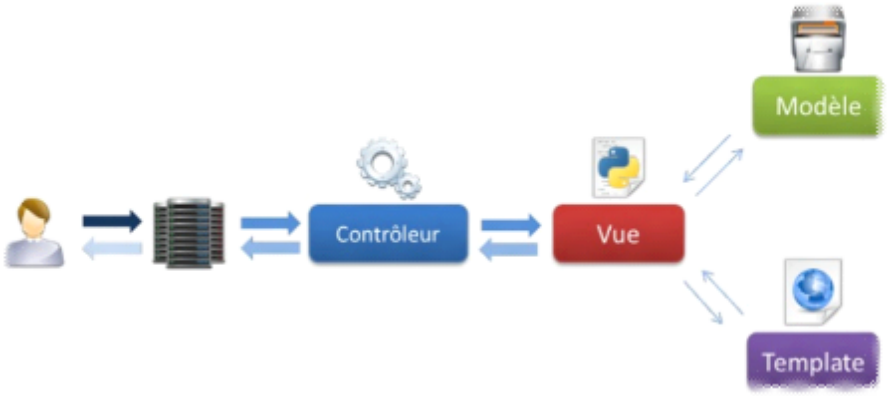


FIGURE 2.8 : L'architecture MVT [4]

Nous avons choisi cette l'architecture MVT parce qu'elle permet une meilleure organisation du code, offre une possibilité de réutilisation et beaucoup plus de faciliter pour les tests unitaires.

2.3.2 Le langage de modélisation UML

Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer de l'information ou de la connaissance ou des systèmes dans une structure qui est définie par un ensemble cohérent de règles.
Le langage UML est un langage de modélisation de type graphique constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. Il sert de modèle pour un projet et garantit une architecture d'information structurée. UML est principalement utilisé dans le développement de logiciels orientés objet. Nous avons opté pour ce langage car il permet d'obtenir une modélisation de très haut niveau, indépendante des langages et des environnements.

2.3.3 SGBD :PostgreSQL

Un système de gestion de base de données (SGBD) est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Il permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données.

PostgreSQL est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, même dans des situations critiques. Il est de type SQL et extrêmement respectueux des standards, se conformant au plus près à la norme ANSI-SQL 2008.

Nous avons choisi PostgreSQL car il est un SGBD relationnel, gratuit, conforme avec SQL, robuste ayant une communauté active.

2.4 Technologies utilisées

2.4.1 Frontend

La partie frontend est la partie avec laquelle un utilisateur interagit directement. Il s'agit notamment de polices, de menus déroulants, de boutons, de transitions, de curseurs, de formulaires de contact, etc. Pour l'application web nous avons utilisé plusieurs technologies à savoir :

- HTML5 est un langage de description de pages web. Il est très flexible, mobile permet d'intégrer du contenu multimédia. HTML permet de construire des pages web robustes adaptées à tous les navigateurs.
- CSS3 est un langage informatique qui permet de mettre en forme des pages web (HTML ou XML). Ainsi, CSS permet d'avoir une cohérence dans l'affichage de nos pages web en gérant la couleur, les tailles, l'animation, la typographie des éléments grâce aux fichiers css.
- Javascript est un langage de programmation de script souvent utilisé pour rendre interactive les pages web. Il nous permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, etc.
- Bootstrap est un framework frontend qui utilise les langages HTML, CSS et JavaScript, ce qui offre aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran et en priorité pour les smartphones. Bootstrap dispose d'un système de grille permettant de positionner des éléments HTML sur une page web. Il fournit également des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation,etc accessible grâce aux classes. Nous avons choisi Bootstrap parce qu'il offre une grande compatibilité avec tous les navigateurs et est facile d'utilisation.
- Jquery est une bibliothèque javascript, simple d'utilisation nous permettant de créer des effets dynamiques complexes avec moins de code javascript.Aussi, Jquery dispose d'une grande communauté très active.
- HTMX est un bibliothèque JavaScript qui simplifie le développement web en permettant d'ajouter des fonctionnalités dynamiques aux pages HTML sans avoir à écrire

beaucoup de code JavaScript. Il se concentre sur l'amélioration progressive des pages web en utilisant les capacités existantes du navigateur. Les fonctionnalités de HTMX sont déclenchées à l'aide des attribut HTML standard, ce qui le code HTML plus expressif et lisible.

2.4.2 Backend

Derrière l'interface avec lequel un utilisateur interagit se cache un autre monde, invisible mais essentiel : le backend. Pour notre application web nous avons utilisé plusieurs technologies à savoir :

- Python est un langage de programmation de haut niveau interprété, orienté objet dont la syntaxe est très lisible et possédant une vaste bibliothèque standard. Python dynamiquement typé ayant une communauté activé et utilisé presque dans tous les domaines.
- DJANGO : est un Framework open source, écrit en Python, qui suit le modèle MVT et encourage le développement rapide et propre d'application web. DJANGO offre un ORM, l'administration automatique et une communauté active.
- REPORTLAB : est une bibliothèque open source pour la création de documents PDF en Python. Elle offre des fonctionnalités puissantes permettant de générer des fichiers PDF dynamique et personnalisés. REPORTLAB possède une documentation complète.

2.4.3 Les logiciels utilisés

VScode : Pour la creation de l'application, nous avons utilisé Visual Studio Code (VScode). C'est un éditeur de texte gratuit et open source développé par Microsoft, disponible sur Windows, Linux et macOS. Malgré sa légèreté il offre des fonctionnalités puissantes qui ont fait de lui l'un des environnements de développement les plus appréciés récemment. Il prend en une multitude de langages : JAVA, C++, Python, HTML, CSS, GO ... VScode offre également la possibilité d'ajouter et même de créer de nouvelles extensions, y compris les débogueurs et le support pour le développement cloud et web. Son interface utilisateur est très interactive par rapport à d'autre éditeurs de texte.

Visual Paradigm est une société de logiciels qui propose une gamme d'outils et de solutions pour la modélisation visuelle et la création de diagrammes. Ils offrent divers produits adaptés à différents aspects du développement logiciel, de la conception système et de la modélisation des processus métier.

2.5 Synthèse des outils utilisés

L'ensemble des outils utilisés sont dans le tableau 2.1 suivant :

Outil		Version
Système d'exploitation		
Ubuntu		$22.04LTS$
Langages		
UML		2.0
HTML		5
CSS		3
Python		3.10
Framework		
Bootstrap		5.4
DJANGO		4.2.3
HTMX		1.0.0
REPORTLAB		4.0.4
SGBD		
PostgreSQL		15.6
Autres logiciels		
VSCode		1.85
Visual Paradigm		17.0

TABLE 2.1 : Synthèse des outils utilisés

Conclusion

Dans ce chapitre, nous avons fait une analyse et réalisé les diagrammes de cas d'utilisation et d'activités. Ceci nous a permis de définir l'architecture de notre système et de choisir convenablement les outils techniques.

Chapitre 3

Réalisation de l'application

Introduction

Dans ce chapitre, nous présenterons dans un premier temps le prototype de notre plateforme à travers ses différentes fonctionnalités. Ensuite nous exposerons quelques limites de notre solution.

3.1 Présentation du prototype de la plateforme

3.1.1 Page d'authentification

Avant d'accéder au système, tout utilisateur doit s'authentifier.

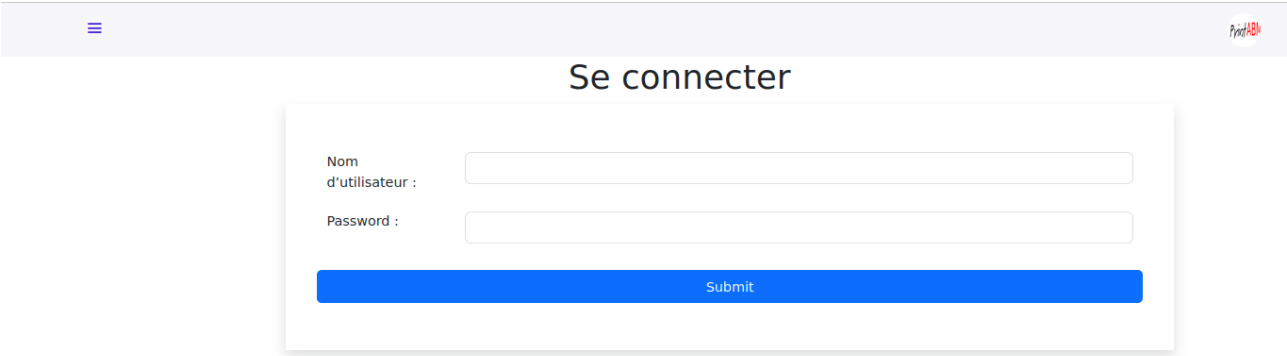


FIGURE 3.1 : Page d'authentification de notre application

3.1.2 Page d'accueil de l'application

Après avoir être authentifié, l'utilisateur sera dirigé vers la page d'accueil de l'application. Sur la page d'accueil et chacune des autres pages nous avons un panneau latéral qui indique le menu principal de l'application.

Depuis le menu principal on peut accéder à la liste des patients, la gestion des utilisateurs (si l'utilisateur connecté est un administrateur) et le menu des statistique et en haut de chaque page il y a le nom d'utilisateur et le rôle de d'utilisateur connecté

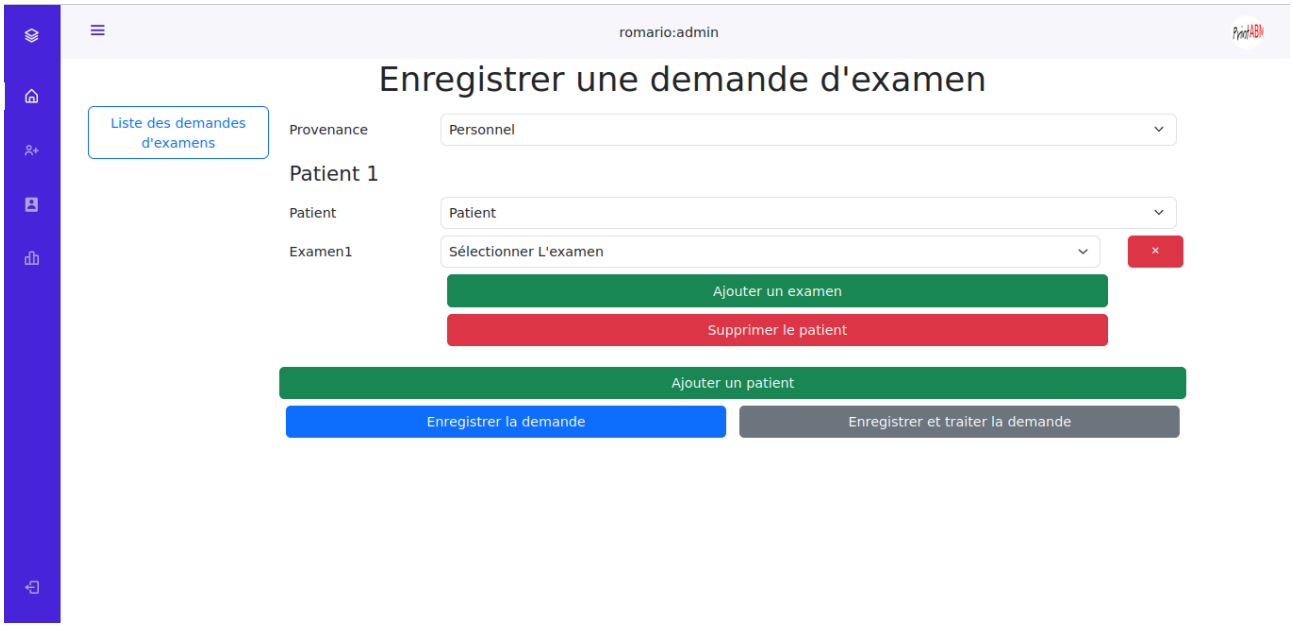


FIGURE 3.2 : Page d'accueil (administrateur) de notre application

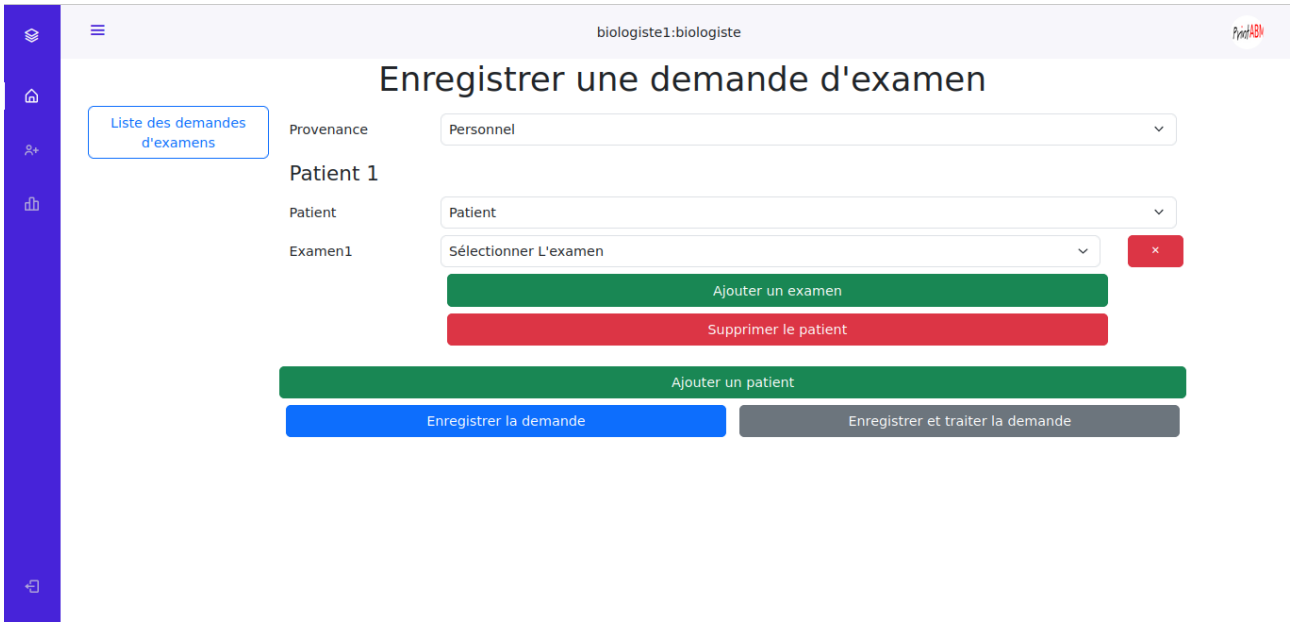


FIGURE 3.3 : Page d'accueil (biologiste) de notre application

Sur la page d'accueil :

- l'utilisateur peut ajouter une demande d'examens ;
- l'utilisateur peut consulter la liste des demandes d'examens existantes.

3.1.3 Liste des demandes d'examens

Sur cette page nous avons la liste des demandes d'examens ordonnée par la date. L'état d'une demande d'examens permet de savoir si les résultats de cette demande sont disponibles ou non et par quel utilisateur cette demande a été traitée.

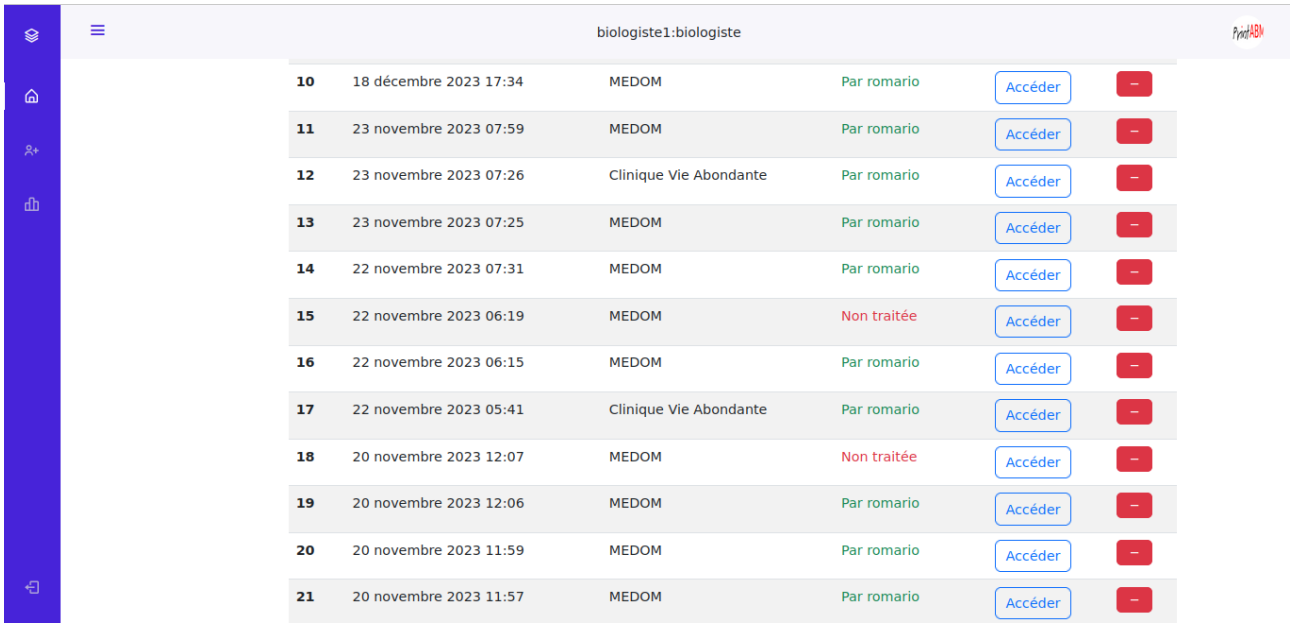


FIGURE 3.4 : Liste des demandes d'examens ordonnées par la date

Sur cette page l'utilisateur peut :

- Accéder à la demande, c'est à dire, voir chaque fiches d'examens qui sont en relations avec cette demande d'examens ;
- Supprimer une demande d'examens.

3.1.4 Formulaire d'enregistrement d'un patient

Sur cette page, le biologiste peut remplir le formulaire permettant d'enregistrer un patient



FIGURE 3.5 : Page d'enregistrement d'un patient

3.1.5 Liste de patients



FIGURE 3.6 : Liste des patients

Sur cette page l'utilisateur peut :

- consulter la liste de tous les patients ;
- Supprimer un patient ;
- Aller sur la page listant les fiches d'examens d'un patient donné

3.1.6 Liste des fiche d'examens d'un patient

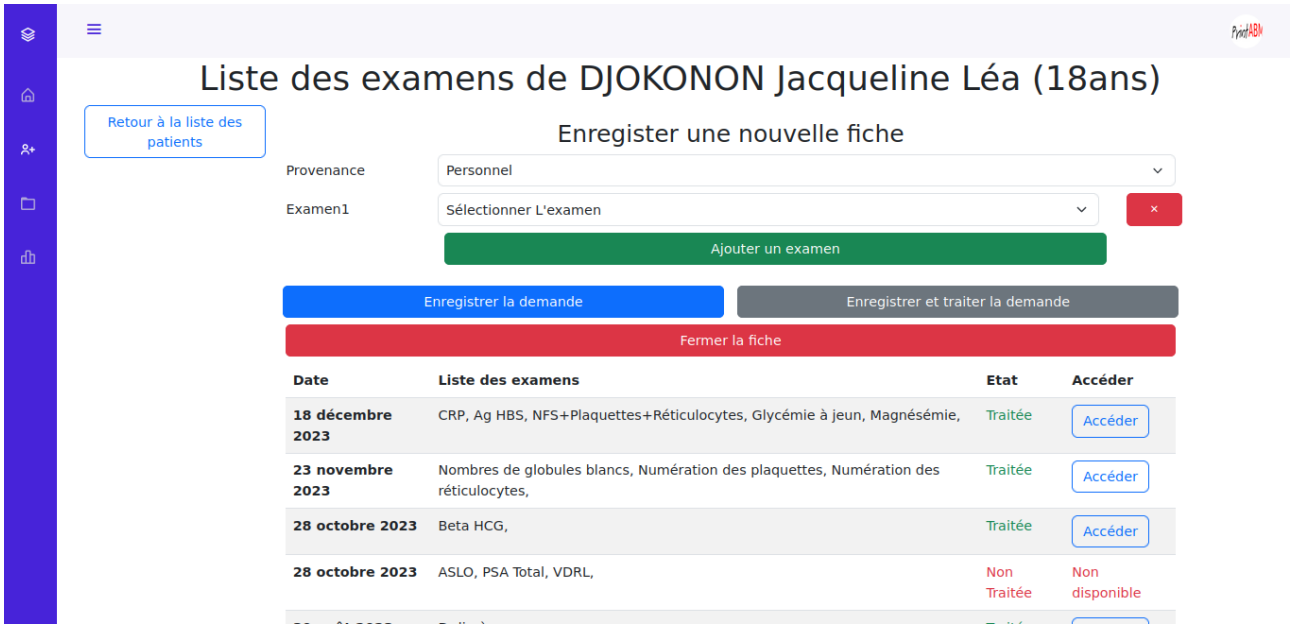


FIGURE 3.7 : Liste des fiches d'examens d'un patient

Sur cette page l'utilisateur peut :

- Consulter la liste des fiches d'examens d'un patient ;
- Ajouter une autre fiche d'examens ;
- Consulter les résultats d'une fiche d'examens si ces résultats sont disponibles.

3.1.7 Traitement d'une demande d'examens

L'utilisateur peut consulter les résultats passés d'une demande d'examens



FIGURE 3.8 : traitement d'une demande d'examens

3.1.8 La liste du personnel

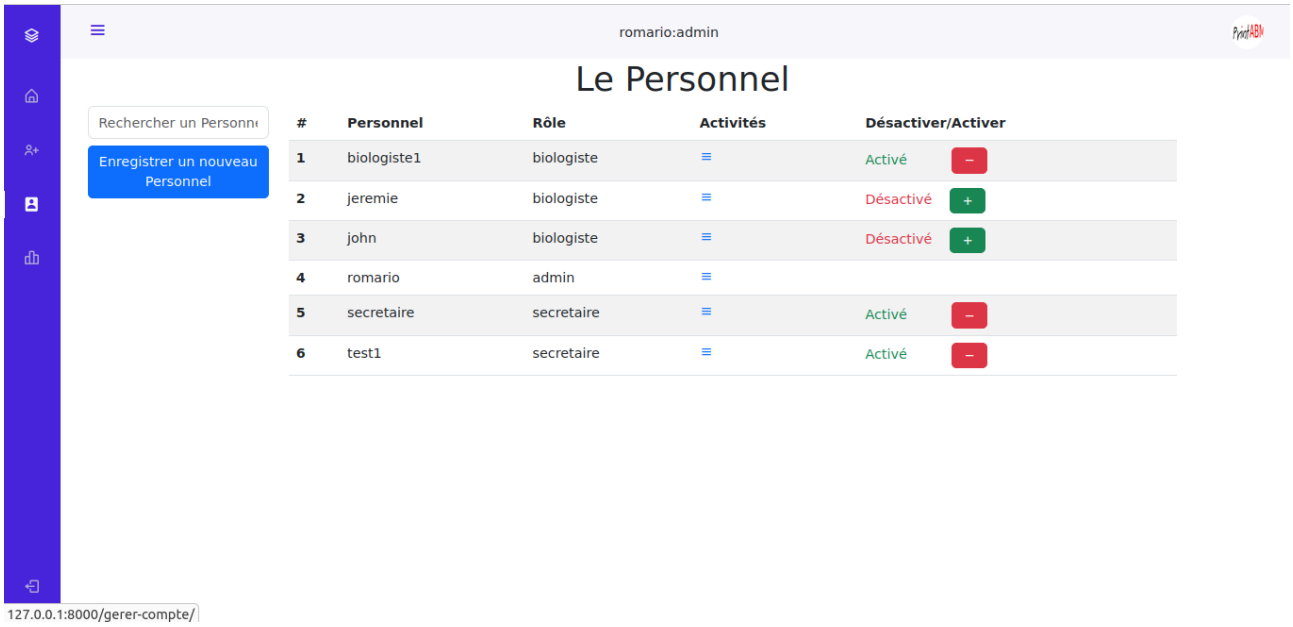


FIGURE 3.9 : La liste du personnel

Sur cette page :

- l'administrateur peut consulter la liste du personnel ;
- désactiver ou activer un personnel ;
- Accéder aux activités d'un personnel donné.

3.1.9 Les activités d'un biologiste



FIGURE 3.10 : Les activités d'un biologiste

Sur cette page :

- l'administrateur peut consulter la liste des demandes crées ou traitées par le personnel ;
- Accéder à une demande d'examens.

3.1.10 Rapport des examens effectués

Sur cette page nous avons la liste des examens effectués dans le mois d'octobre 2023 avec la recette

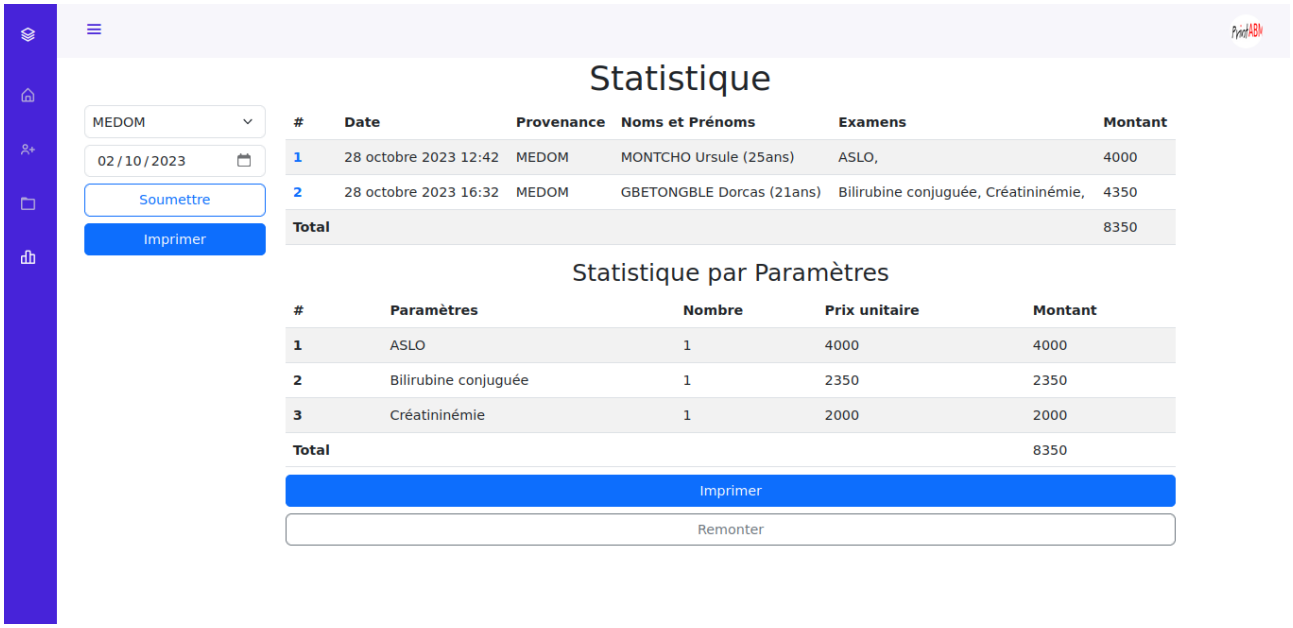
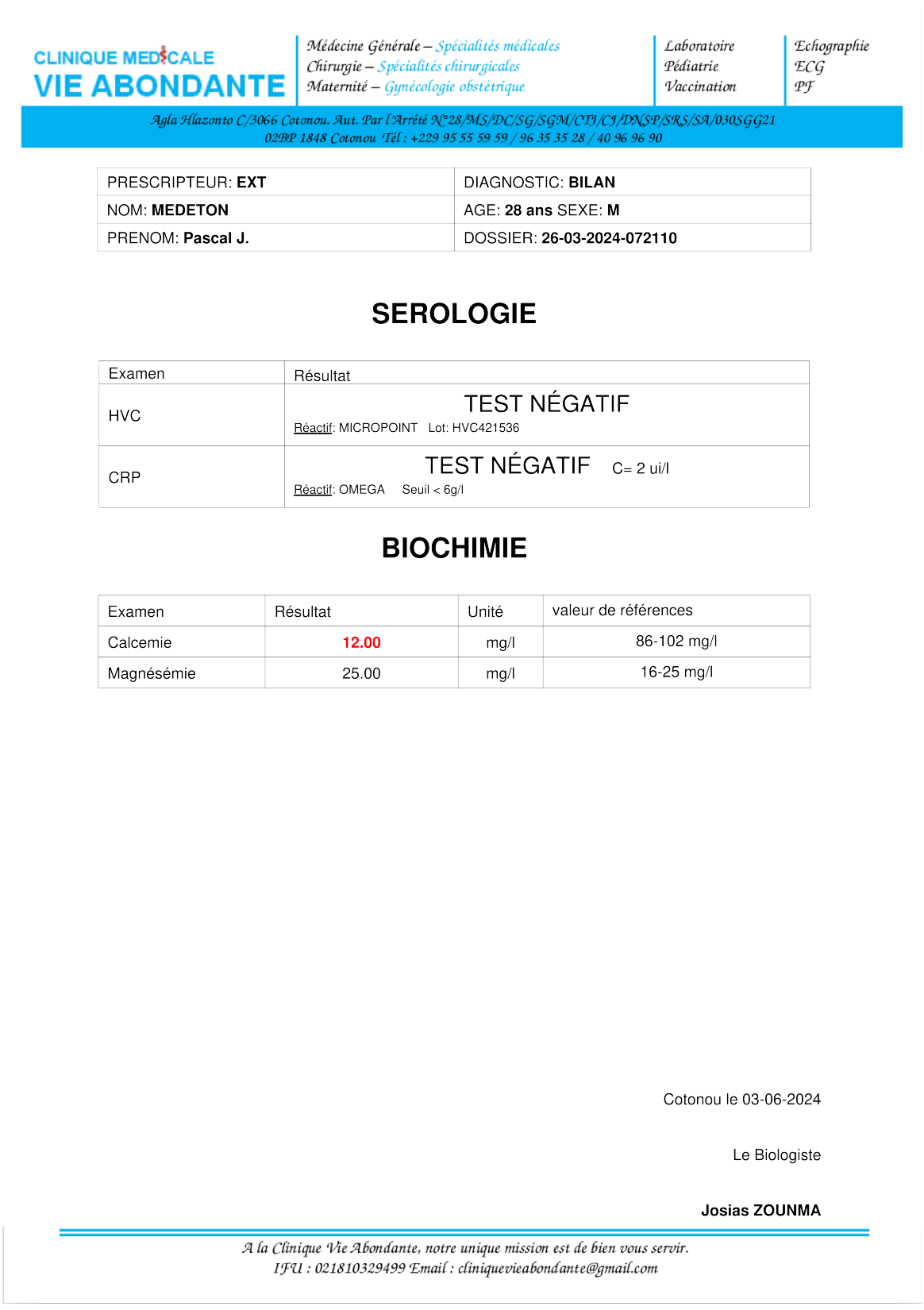


FIGURE 3.11 : Rapport des examens éffectués

3.1.11 Impression d'un résultats sous forme PDF

Voici un résultat d'examens imprimer en PDF



CLINIQUE MEDICALE
VIE ABONDANTE

Médecine Générale – Spécialités médicales
Chirurgie – Spécialités chirurgicales
Maternité – Gynécologie obstétrique

Laboratoire
Pédiatrie
Vaccination

Echographie
ECG
PF

Agla Hlazonto C/3066 Cotonou. Aut. Par l'Arrêté N°28/MS/DC/SG/SGM/CTJ/CJ/DNSP/SRS/SA/030SGG21
02BP 1848 Cotonou Tél : +229 95 55 59 59 / 96 35 35 28 / 40 96 96 90

PRESCRIPTEUR: **EXT**	DIAGNOSTIC: **BILAN**
NOM: **MEDETON**	AGE: **28 ans** SEXE: **M**
PRENOM: **Pascal J.**	DOSSIER: **26-03-2024-072110**

SEROLOGIE

Examen	Résultat
HVC	TEST NÉGATIF Réactif: MICROPOINT Lot: HVC421536
CRP	TEST NÉGATIF C= 2 ui/l Réactif: OMEGA Seuil < 6g/l

BIOCHIMIE

Examen	Résultat	Unité	valeur de références
Calcemie	12.00	mg/l	86-102 mg/l
Magnésémie	25.00	mg/l	16-25 mg/l

Cotonou le 03-06-2024

Le Biologiste

Josias ZOUNMA

A la Clinique Vie Abondante, notre unique mission est de bien vous servir.
IFU : 021810329499 Email : cliniquevieabondante@gmail.com

FIGURE 3.12 : impression d'un résultat sous forme pdf

3.2 Sécurité

Nous travaillons avec les résultats d'examens des patients. certains résultats sont confidentiels car ils peuvent avoir des effets sur la vie sociale des patients, ainsi la sécurité est une propriété indispensable pour notre système.

Pour garantir la sécurité des données gérées par le système, nous avons le système d'authentification de DJANGO : *django.contrib.auth*. C'est un module intégré dans DJANGO qui permet de gérer l'authentification des utilisateurs, l'inscription, la connexion, la déconnexion et la réinitialisation du mot de passe.

3.3 Discussion

La solution développée nous a permis de :

1. Centraliser les données : Les données sont organisées dans une base ce qui facilite leur accès et leur l'analyse ;
2. Sécuriser les données : Nous avons utilisé le système d'authentification de DJANGO pour garantir la confidentialité des données ;
3. Générer automatiquement un rapport : Le système permet de générer automatiquement un rapport des travaux effectués dans le laboratoire sur une durée donnée ;
4. Réduire le temps de saisir et d'impression : le système permet d'insérer rapidement avec moins d'erreur les résultats des examens et réduire le temps de saisir-impression.

Elle connaît aussi certaines insuffisances aux nombres desquelles nous avons :

1. L'ajout des paramètres est fait par le développeur. On peut permettre à l'utilisateur d'ajouter des nouveaux paramètres (examens) et définir la manière dont le résultat de chaque paramètre sera présenté.
2. Intégrer aussi la sous-traitance (préciser si l'examen a été fait dans le laboratoire ou confié à un autre laboratoire).
3. Permettre à l'application de communiquer avec d'autres applications de santé, cela peut permettre de bien suivre et bien traiter les patients.
4. Introduire les antécédents, c'est-à-dire sur chaque nouveau résultat d'examen, il aura le plus récent résultat de cet examen réalisé par le patient dans le laboratoire, cela peut permettre au médecin de rapidement prendre des décisions.

Conclusion

Cette étape nous a permis de présenter le prototype développé en montrant chaque page, leur liaison avec les autres pages et en montrant comment fonctionne le système. Nous avons également soulevé dans ce chapitre les insuffisances de notre prototype.

Conclusion Générale

Le présent rapport est le résultat de notre stage effectué dans le cadre de la réalisation de notre projet de fin d'études de Licence au sein du laboratoire biomédicale de la Clinique "La Vie abondante". Au cours de ce stage de 3 mois, nous avons pu mettre en pratique nos connaissances théoriques acquises durant notre formation.

Le laboratoire avait besoin d'un système de gestion des données, qui va permettre d'avoir la traçabilité des données d'un patient, la présentation d'un résultat de qualité tout en éliminant les erreurs de saisie et réduire le temps de saisie des documents, et présenter un rapport des travaux effectués dans une durée donnée.

Nous avons, tout d'abord, entamé notre étude par l'analyse qui est une étape cruciale et nécessaire pour mieux assimiler le système déjà existant, puis l'identification des besoins. Enfin, nous avons procédé au choix justifié des technologies avant de passer à l'implémentation, qui nous a permis de développer une application répondant aux attentes de l'utilisateur.

De ce qui précède, il est difficile de prétendre avoir eu une solution idéale. Toutefois, nous espérons avoir répondu tant soi peu à notre problématique. Cependant, nous sommes conscientes des améliorations à faire. Nous avons prévu d'intégrer un système qui peut permettre à l'utilisateur de mettre les paramètres qu'il voudra.

Bibliographie

[1] Ninon KPOSSOU et Ratheil HOUNDJI. “Développement d’une plateforme de gestion des rendez-vous et des dossiers médicaux dans les établissements de santé”. In : (2017).

[2] Audrey METOEVI et Pierre Jerôme ZOHOU. “Aide à la gestion numérique du flux d’accès d’un site touristique”. In : (2021).

Webbographie

[3] *Labbook*. https ://www.lab-book.org. Consulté le 20 février 2024 à 16h52.

[4] *les modèles MVC et MVT*. https ://www.memoireonline.com/11/22/13439/mEssai-de-conception-et-dimplementation-dune-plateforme-web-daide-a-lorient27.html. Consulté le 25 février 2024 à 14h11.

[5] *GestLab*. https ://www.cirtait.com/gestlab/. Consulté le 23 février 2024 à 09h35.

[6] *ProLab LIS*. https ://www.prolablis.com. Consulté le 20 février 2024 à 16h52.

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

. iv

Abstract v

. v

Table des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire 1

Introduction 2

1 Revue de littérature 4

Introduction 4

1.1 Clarification conceptuelle 4

1.2 Labbook 5

1.3 ProLab Lis 6

1.4 GestLab 7

Conclusion 8

2 Analyse, Conception,Choix techniques 9

Introduction 9

2.1 Analyse des besoins 9

2.2 Conception 10

2.2.1 Identification des acteurs 10

2.2.2 Le diagramme de cas d'utilisation 10

2.2.3 Le diagramme de classe 11

2.2.4 Diagramme d'activité 11

2.2.5 Diagramme de séquence 13

2.3 Choix techniques 15

2.3.1 L'architecture MVC et MVT 15

2.3.2 Le langage de modélisation UML 16
2.3.3 SGBD :PostgreSQL 16
2.4 Technologies utilisées 17
2.4.1 Frontend 17
2.4.2 Backend 18
2.4.3 Les logiciels utilisés 18
2.5 Synthèse des outils utilisés 18
Conclusion 19

3 Réalisation de l'application **20**
Introduction 20
3.1 Présentation du prototype de la plateforme 20
3.1.1 Page d'authentification 20
3.1.2 Page d'accueil de l'application 21
3.1.3 Liste des demandes d'examens 22
3.1.4 Formulaire d'enregistrement d'un patient 22
3.1.5 Liste de patients 23
3.1.6 Liste des fiche d'examens d'un patient 24
3.1.7 Traitement d'une demande d'examens 24
3.1.8 La liste du personnel 25
3.1.9 Les activités d'un biologiste 25
3.1.10 Rapport des examens effectués 26
3.1.11 Impression d'un résultats sous forme PDF 26
3.2 Sécurité 28
3.3 Discussion 28
Conclusion 28

Conclusion **30**

Table des matières **33**