



REPUBLIQUE DU BENIN

**MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE (MESRS)**

DIRECTION GENERALE DE L'ENSEIGNEMENT SUPERIEUR (DGES)

HAUTE ECOLE DE COMMERCE ET DE MANAGEMENT (HECM)

**Mémoire de fin de formation pour l'obtention du diplôme de licence
professionnelle**

Option : Sciences et Techniques Filière : Système Informatique et Logiciel

Année académique : 2023-2024

THEME :

**CONCEPTION ET REALISATION D'UNE
APPLICATION DE SUIVI DE SANTE**

Réalisé par :

MALADE Zéidane & MOUTAIROU Imdade

Tuteur de stage :

M. Parfait FADONUGBO

Développeur FULL STACK,

Formateur à CITéDA

Sous la supervision de :

Maitre de mémoire :

M. Jérôme K. KINMISSÉDO

Ingénieur en Réseaux et SI,

Enseignant à HECM.

JUIN 2024

Dédicace

à :

- ◆ mon père MALADE Mohamed;
- ◆ ma mère Mouinath BAKARI.

MALADE Zéidane.

Dédicace

à :

- ◆ mon père Anziz MOUTAIROU ;
- ◆ ma mère Soubédatou SALOU.

MOUTAIROU Imdade.

REMERCIEMENTS

La réalisation de ce travail n'aurait pu être menée à bien sans la contribution essentielle d'un grand nombre de personnes. Nous exprimons notre gratitude à toutes ces personnes qui de près ou de loin ont participé à la réalisation de ce mémoire, en particulier :

- ❖ l'Honorable AKE Natondé, Promoteur de HECM pour avoir mis à notre disposition le cadre de formation ;
- ❖ M. Jérôme KINMISSEDO, notre directeur de mémoire pour la qualité de son enseignement, ses conseils et son intérêt qu'il nous a porté ;
- ❖ M. Parfait FADONUGBO, de nous avoir accordé un stage académique et formation au sein de son entreprise ;
- ❖ M. Parfait FADONUGBO notre maître de stage, pour son suivi ;
- ❖ Dr. Moutiou ADEBIYI, Directeur des Etudes à HECM Porto-Novo, pour ses nombreux conseils ;
- ❖ le corps professoral de HECM Porto-Novo pour la qualité de son enseignement, ce qui nous a donné les aptitudes pour réaliser ce projet ;
- ❖ tous nos proches, pour l'assistance morale, matérielle et financière qu'ils ne cessent de nous apporter et pour leurs différents conseils.

LISTE DES SIGLES ET ACRONYMES

A : Alphabétique

AN : Alphanumérique

CITéDA : Centre Informatique de Technologie et de Développement d'Application

CSS : Cascading Style Sheets

CV : Curriculum Vitae

D : Date

DG : Directeur Général

FTP : File Transfer Protocol

HECM : Haute Ecole de Commerce et de Management

HTML : HyperText Markup language

IDS : Intrusion Detection System

IPS : Intrusion Prevention System

LOO : Langage Orienté Objet

M : Monsieur

MVC : Model Vue Contrôleur

N : Numérique

PHP : HyperText Preprocessor

SIL : Système Informatique et Logiciel

SQL : Structured Query Language

SSL : Secure Sockets Layer

SGBD : Système de Gestion de Base de Données

TIC : Technologies de l'Information et de la Communication

UML : Unified Modeling Language

W3C : World Wide Web Consortium

WWW : World Wide Web

XML : eXtensible Markup Language

2FA : Two-Factor Authentication

XAMPP : X (cross) Apache MariaDB Perl PHP

LISTE DES FIGURES

Figure1 : Organigramme de CITéDA.....	7
Figure2 : Diagramme de contexte statique.....	18
Figure3 : Diagramme de cas d'utilisation.....	19
Figure4 : Diagramme de classes.....	23
Figure5 : Diagramme de séquence cas s'authentifier.....	27
Figure6 : Diagramme de séquence cas demande de consultation.....	29
Figure7 : Diagramme de séquence cas consulter dossiers médicaux.....	31
Figure8 : Diagramme d'activité cas éditer ordonnance.....	32
Figure9 : Diagramme d'activité cas enregistrer recommandations.....	33
Figure10 : Schéma de l'organisation du MVC.....	39
Figure11 : Page de connection.....	50
Figure12 : Page d'inscription.....	51
Figure13 : Page d'Accueil.....	52
Figure14 : Page d'Ajout d'un patient.....	52
Figure15 : Page liste des patients.....	53

LISTE DES TABLEAUX

Tableau1 : Dictionnaire des données.....	22
Tableau2 : Cas d'utilisation s'authentifier.....	26
Tableau3 : Cas d'utilisation demande de consultation.....	28
Tableau4 : Cas d'utilisation dossiers médicaux.....	30

RESUME

Notre application de suivi de santé est une plateforme complète qui permet utilisateurs de surveiller et d'améliorer leur bien-être physique au quotidien. En enregistrant facilement leur mesure de santé telles que le poids, la taille, la pression artérielle et la fréquence cardiaque, les utilisateurs peuvent suivre leur évolution au fil du temps et établir des objectifs personnalisés. Grace à des fonctionnalités avancées telles que les conseils personnalisés, les rappels et les notifications, nous encourageons les utilisateurs à adopter des habitudes de vie saines et à rester motivés dans leur parcours de bien-être. Avec un accent particulier sur la convivialité et la sécurité des données, notre application offre une expérience utilisateur optimale tout en garantissant la confidentialité des informations personnelles. En somme notre application de suivi de santé est bien plus qu'une plateforme de suivi de mesures de santé, c'est un partenaire indispensable pour aider les utilisateurs à atteindre leurs objectifs de bien-être physique et à vivre pleinement leur vie.

Mots clés : Santé , Suivi, Utilisateurs ;

ABSTRACT

Our health tracking application is a complete platform that allows users to monitor and improve their physical well-being on a daily basis. By easily recording their health measures such as weight, height, blood pressure and heart rate, users can track their evolution over time and set personalized health goals. With advanced features such as personalized advice, reminders and notifications, our application encourages users to adopt healthy lifestyles and stay motivated in their well-being journey. With a particular focus on usability and data security, our application offers an optimal user experience while guaranteeing the confidentiality of personal information. In short, our health monitoring application is much more than just a platform for monitoring health measures - it is an essential partner to help users achieve their physical well being goals and live their lives to the fullest.

Keywords: Health , Monitoring, Users ;

SOMMAIRE

Introduction :	2
Chapitre I : Présentation de la structure d'accueil	4
I. Présentation de CITéDA.....	5
II. Déroulement du stage.....	10
III. Présentation du thème.....	11
IV. Spécification fonctionnelle.....	12
V. Spécification non fonctionnelle.....	14
Chapitre II : Analyse et conception du système.....	15
I. Analyse fonctionnelle.....	16
II. Analyse statique.....	21
III. Analyse dynamique.....	25
Chapitre III : Réalisation du site web.....	35
I. Présentation des outils et des technologies utilisées.....	36
II. Architecture du système.....	39
III. Présentation de l'application.....	51
Conclusion :	56

INTRODUCTION

Dans un monde où la santé et le bien-être occupent une place de plus en plus importante, l'utilisation de la technologie pour surveiller et améliorer notre condition physique est devenue une pratique courante. En combinant des fonctionnalités avancées de suivi des mesures de santé, des conseils personnalisés, et une interface conviviale, notre application vise à offrir une expérience utilisateur enrichissante et efficace pour favoriser des modes de vie plus sains et équilibrés.

C'est dans ce contexte que nous nous engageons à faire notre étude autour du thème : CONCEPTION D'UNE APPLICATION DE SUIVI DE SANTE D'UN PATIENT ; une solution innovante conçue pour aider les utilisateurs à prendre en charge leur bien-être physique de manière personnalisée et sécurisée

Le présent travail s'articulera autour de trois grands chapitres :

- le premier chapitre intitulé présentation de la structure d'accueil et étude de l'existant comporte la présentation du lieu de stage et l'étude des sites existant ;
- le deuxième chapitre profère analyse et conception consiste à faire l'analyse fonctionnel et non fonctionnel en plus de la conception de l'application ;
- le troisième chapitre intitulé la réalisation de l'application présente les outils et technologies utilisés ainsi que l'architecture mis en place.

Chapitre I :

Présentation de la structure d'accueil

I. PRESENTATION DE CITéDA

1. Historique

CITéDA (Centre Informatique de Technologies et du Développement des Applications) est une entreprise de service en ingénierie informatique spécialisée en développement logiciel, installation réseaux et création de sites web. Elle fournit des services de maintenance informatique et assure la vente de produits informatiques. CITéDA, dont le site est situé à Porto-Novo, précisément à Iré Akari, organise des formations, des séminaires et des recrutements pour des entreprises. Elle est créée depuis 2020 et a développé ses objectifs autour des besoins qui touchent des institutions étatiques et des étudiants en matière de technologie et de gestion. Au sein de CITéDA, la curiosité, l'esprit d'équipes, l'innovation et le travail bien fait sont quelques secrets de réussite de cette entreprise.

2. Mission

L'entreprise s'est donnée comme mission de favoriser l'accès aux **TIC** et des outils de gestion des activités commerciales en assurant une formation des techniciens qualifiés. Afin de remplir efficacement cette mission, le directeur de l'entreprise CITéDA s'est associé avec des techniciens dévoués à la tâche pour assurer des informations de qualité aux jeunes résidents dans la ville capitale et ses environs.

L'objectif principale de CITéDA est de fournir des services de développement d'application d'offrir plus d'opportunités en termes de formation et de stage aux jeunes de la ville de Porto-Novo. Pour atteindre ce but, l'entreprise s'est fixé des objectifs spécifiques qui sont :

- Offrir un bon cadre de remise à niveau des professionnels en activités de même qu'aux Assistants de Directions ;
- Assurer des formations pour la conception graphique et la conception de sites internet ;

- Fournir des solutions technologiques et logiciel pour faciliter la gestion des entreprises ;
- Assurer des formations pour le dimensionnement, la configuration et l'installation des réseaux informatiques et téléphoniques ...

3. Objectifs

Les objectifs de CITéDA sont :

- ✓ installer un laboratoire électronique et informatique moderne, bien équipé où on pourra sortir des produits électroniques ou informatiques finis ;
- ✓ créer une confrérie de technologues pour une veille électronique, électrotechnique, informatique et de télécommunication permanente afin de réfléchir et de résoudre les différentes questions relatives à ces secteurs ;
- ✓ développer un esprit de solidarité et de coopération entre les ingénieurs béninois afin qu'ensemble, nous puissions participer au développement de notre cher pays ;
- ✓ mettre à la disposition de la population des consommables électroniques, informatiques et des énergies renouvelables.

4. Structure Organisationnelle

ORGANIGRAMME de CITéDA :

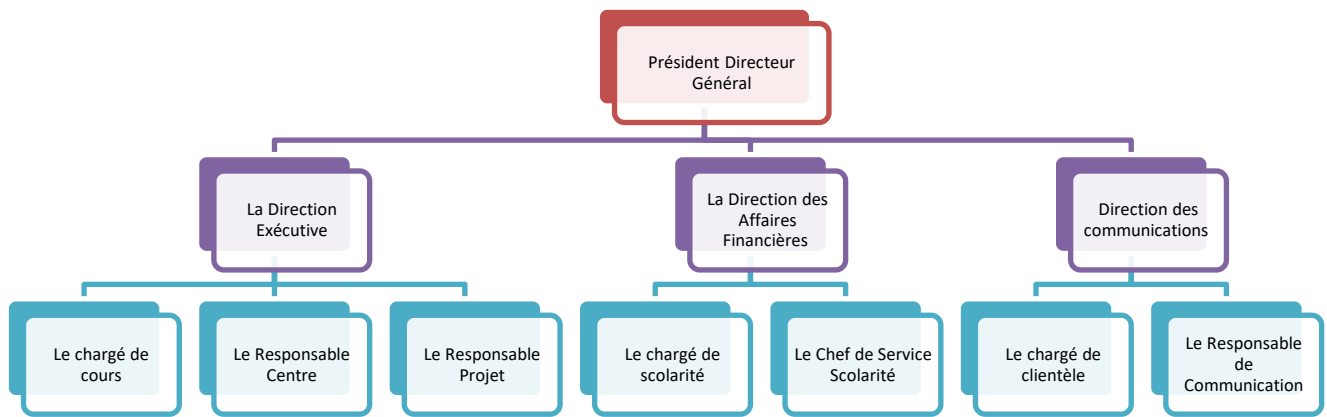


Figure 1 Organigramme de CITéDA

4.1 Président directeur général :

Organe décisionnel, elle organise et coordonne toutes les activités de l'entreprise. Elle prend les décisions en collaboration avec les autres directions et assure le suivi de leur exécution.

4.2 La Direction Exécutive :

Elle assure la mise en œuvre des décisions stratégiques prises par la direction Générale. Ils supervisent les opérations quotidiennes, coordonnent les différentes activités du centre de formation et veillent à ce que les objectifs fixés soient atteints.

➤ Le chargé de cours :

Elle s'occupe de d'animation des cours, des ateliers ou des séminaires sur des sujets spécifiques. Leur rôle est d'enseigner, de guider et d'évaluer les étudiants afin de les aider à acquérir les compétences et les connaissances nécessaires dans leur domaine d'étude.

➤ Le Responsable Centre :

Elle est la personne chargée de superviser l'ensemble des opérations et activités du centre. Cela inclut la gestion administrative, financière, pédagogique et logistique.

➤ Le Responsable Projet :

Il est chargé de superviser la mise en œuvre et la gestion des projets spécifiques au sein de l'organisation. Cela peut inclure le développement de nouveaux programmes de formation, l'organisation d'événements, la mise en place de partenariats avec d'autres institutions, ou encore la gestion de projets de recherche.

4.3 La direction des affaires financières :

Elle assure la gestion des aspects administratifs, financiers et comptables de l'organisation. Cela inclut la gestion des budgets, la comptabilité, la gestion des ressources humaines, et parfois même les aspects liés aux achats.

➤ Le chargé de scolarité :

Elle est responsable de la gestion administrative des étudiants, de leur inscription jusqu'à leur diplomation. Elle assure la gestion des dossiers des étudiants, l'organisation des emplois du temps, la coordination des examens, la communication des informations administratives aux étudiants, ainsi que le suivi de leur progression académique.

➤ Le Chef de Service Scolarité :

Elle est responsable de la supervision et de la coordination de toutes les activités liées à la scolarité au sein du centre de formation. Cela comprend la

gestion administrative des étudiants, l'organisation des inscriptions et des réinscriptions, la coordination des emplois du temps, la gestion des examens et des résultats, ainsi que la coordination des équipes de chargés de scolarité.

4.4 Direction des communications :

Ce responsable est chargé de superviser toutes les activités de communication de l'organisation, y compris les relations publiques, la gestion des médias sociaux, les communications internes et externes, ainsi que la stratégie de communication globale du centre de formation.

➤ **Le chargé de clientèle :**

Elle est responsable de la gestion des relations avec les clients, qui peuvent être des entreprises, des organismes gouvernementaux ou des individus. Leur rôle consiste à comprendre les besoins de chaque client, à les conseiller sur les programmes de formation disponibles, à faciliter le processus d'inscription et à assurer un suivi personnalisé tout au long du parcours de formation.

➤ **Le Responsable de Communication :**

Elle est chargée de superviser toutes les activités de communication de l'organisation. Leur rôle consiste à élaborer et mettre en œuvre des stratégies de communication efficaces pour promouvoir les programmes de formation, renforcer l'image de marque du centre, et attirer de nouveaux étudiants. Cela peut inclure la gestion des médias sociaux, les relations publiques, la création de contenu, la coordination des événements, et la communication interne.

II. DEROULEMENT DU STAGE

Nous avons débuté les stages à CITéDA le 20 Février 2023. Nous avons eu à revoir des langages comme PHP, HTML, MYSQL, ce qui nous a aidé dans la réalisation de notre projet de fin d'année. Nous avons appris à travailler de manière organiser, bon et accélérer.

1. Activités effectuées :

Durant notre séjour à CITéDA, en vue de renforcer nos connaissances et réaliser notre projet nous avons menées plusieurs activités. Il s'agit :

✓ Création de notre application

Nous avons faire usage des langages tel que : PHP, CSS, HTML, JAVASCRIPT aussi la conception de notre application est faite avec l'éditeur de code Android studio, le serveur XAMPP et le Framewok BOOTSTRAP.

✓ Base de données

Nous avons implémenté notre base de données avec XAMPP, on a géré la récupération, la sauvegarde, la modification et la suppression etc. des données ont été gérées avec les requêtes SQL.

2. Les apports du stage :

Au cours de ces trois (03) mois de stage, nous avons acquis des connaissances pratiques puis développer des compétences techniques et aussi nous avons eu l'opportunité de travailler avec des professionnels afin de mieux servir dans le domaine professionnel.

III. PRESENTATION DU THEME

1. Explication

Notre thème porte sur le développement d'une application web et mobile permettant aux utilisateurs de prendre en main leur santé, de suivre leurs progrès et d'adopter un mode de vie plus sain.

2. Problématique

Avec l'évolution des technologies numériques et la nécessité croissante d'améliorer la gestion des soins de santé, le développement de plateformes de suivi de

santé des patients est devenu une priorité. Ces plateformes doivent non seulement permettre la collecte des données médicales en temps réel, mais aussi garantir la confidentialité et la sécurité des informations sensibles des patients. De plus, elles doivent être conçues de manière à faciliter l'accès aux dossiers médicaux pour le suivi des patients, tout en intégrant des capacités d'analyse avancées pour fournir des perspectives utiles à la prise de décision médicale.

3. Objectif général

L'objectif général de ce projet est de développer une application de suivi de la santé complète et conviviale, facilitant l'accès aux soins à la population et permettant de réduire le taux de mortalité et de surveiller et d'améliorer leur bien-être physique de manière personnalisée et sécurisée.

4. Objectifs spécifiques

Les objectifs de la conception de notre plateforme de suivi de santé d'un patient peuvent inclure :

- Concevoir une interface utilisateur intuitive et conviviale, facilitant l'enregistrement des mesures de santé et l'accès aux fonctionnalités de l'application.
- Implémenter un système de suivi avancé des mesures de santé, offrant aux utilisateurs des outils pour enregistrer leur poids, leur taille, leur tension artérielle, leur fréquence cardiaque, etc.
- Intégrer des fonctionnalités de personnalisation, permettant aux utilisateurs de définir des objectifs de santé individuels et de recevoir des conseils personnalisés en fonction de leurs préférences et de leur historique de santé.
- Assurer la sécurité et la confidentialité des données des utilisateurs en mettant en place des mesures de protection robustes, telles que le chiffrement des données et l'authentification sécurisée.
- Intégrer des rappels et des notifications pour encourager les utilisateurs à enregistrer régulièrement leurs mesures de santé, à suivre leurs programmes d'exercices et à adopter des habitudes de vie saines.

- En atteignant ces objectifs, l'application de suivi de la santé vise à fournir aux utilisateurs une solution pratique et efficace pour prendre en charge leur bien-être physique au quotidien, tout en respectant leurs besoins individuels et en assurant la sécurité de leurs données.

IV. SPECIFICATIONS FONCTIONNELLES

Les besoins des clients sont mis en exergue ici. Il s'agit de l'inscription, la gestion des clients, la gestion des commandes, l'authentification des clients. Ces différents cas seront expliqués à travers les besoins fonctionnel afin de proposer une plateforme qui répond aux besoins des utilisateurs.

1. Les besoins fonctionnels

Les besoins fonctionnels sont les solutions proposées par la plateforme, plus précisément les fonctionnalités du système. Ils se présentent comme suit :

Patient :

- Créer un compte ;
- Se connecter ;
- Echanger avec le médecin ;
- Suivre dossier médical ;
- Télécharger une ordonnance.

Médecin :

- Consulter les demandes du patient ;
- Entretenir avec le patient ;
- Créer dossier médical ;
- Effectuer des observations ;
- Faire des recommandations.

Responsable :

- S'authentifier ;

- Modifier les informations du patient ;
- Consulter les dossiers médicaux.

Administrateur :

- Gérer la maintenance.

2. L'inscription

Un utilisateur peut créer un compte en fournissant ses informations telles que :nom, prénom, téléphone, email et mot de passe.

3. L'authentification

L'utilisateur peut se connecter en renseignant son adresse email et mot de passe, il peut aussi réinitialiser son mot de passe en cas d'oubli.

Nous avons géré ces différentes fonctions en faisant appel à UML (UNIFIED MODELING LANGUAGE) qui consiste à répertorier et modéliser les différents progrès métiers afin de migrer facilement vers une architecture objet d'un point de vue statique.

V. SPECIFICATIONS NON FONCTIONNELLES

Les besoins non fonctionnels désignent tout simplement le comportement et la performance que notre plateforme doit avoir. Ils renvoient donc aux indicateurs de qualité et sont importants car ils agissent de façon indirecte sur les rendements de l'utilisateur et de l'outil. Pour cela, on a :

1. Fiabilité

Notre site web fonctionne de façon structurée, avec rapidité de traitement et

d'affichage sans problème

2. Performance

Notre application peut gérer une très grande quantité d'utilisateur simultanément sans dégradation des performances, et le temps de réponse pour les actions courantes (connexion, enregistrement de mesures, etc...) est très infime. L'utilisateur aura une navigation claire entre les différentes pages et un visuel de qualité basé sur les couleurs et les styles de textes.

3. Sécurité

Notre outil est basé sur la sécurité et l'intégrité des données qui y seront enregistrées. En ce qui concerne les utilisateurs, ils doivent se connecter à leur espace personnel avec un mot de passe propre à chacun. Notre application implémente des mécanismes de protection contre les attaques courantes (injection SQL, XSS, etc...).

4.Disponibilité

Notre application est disponible 99.9% du temps, avec des temps d'arrêt minimaux pour la maintenance.

5.Compatibilité

L'application est compatible avec les principaux navigateurs web (Chrome, Firefox, Safari, Edge). Une version de l'application sera disponible pour les systèmes Android et iOS.

Chapitre II :

Analyse et conception du système

I. ANALYSE FONCTIONNELLE

1. Choix du langage de modélisation

Le langage UML (Unified Modeling Language ou Langage de Modélisation Unifié en français) est un langage de modélisation visuelle commun et riche aussi sémantiquement que syntaxiquement. Il permet de faire la représentation des modèles sans pour autant définir le processus de leur élaboration. Il fonctionne comme une méthode normalisée de visualisation sans en être une. En effet, UML ne procure pas de solution pour la réalisation d'un projet. UML n'est pas un langage de programmation. Toutefois, il combine plusieurs notations orientées objet : la conception orientée objet, la technique de modélisation objet et le génie logiciel orienté objet. Il convient de noter que le langage UML a été élaboré par un groupe de trois penseurs Grady, Booch et Rumbaugh en 1996. Très vite, il a été adopté par des sociétés telles que Microsoft, IBM et Oracle, preuve de son efficacité.

2. Les diagrammes UML

Les diagrammes UML sont représentés sous deux types de vues du système à savoir les vues fonctionnelles et les vues dynamiques. Ainsi peuvent-ils être subdivisés en deux groupes. Nous avons d'une part les diagrammes UML structurels et d'autre part les diagrammes UML comportementaux. Les diagrammes UML structurels regroupent :

- le diagramme de classe ;
- le diagramme de composants ;
- le diagramme de structure composite ;
- le diagramme d'objets ;
- le diagramme de déploiement ;
- le diagramme de paquetage.

Les diagrammes UML comportementaux sont :

- le diagramme d'activités ;

- le diagramme de communication ;
- le diagramme de séquence ;
- le diagramme global d'interaction ;
- le diagramme de temps ;
- le diagramme de cas d'utilisation ;
- le diagramme d'états-transitions.

L'utilité de ces diagrammes est variable. UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun. Il n'est donc pas nécessaire de tous les reproduire pour une modélisation. Le diagramme de classe est considéré comme l'élément central en UML. À lui s'ajoutent les diagrammes de cas d'utilisation, d'objets, de séquences, d'états-transitions et d'activités pour donner les diagrammes les plus utiles pour une modélisation. Dans le cadre de notre travail, nous utiliserons Microsoft Visio. Ce logiciel de la suite Microsoft Office permet de faire l'analyse et la reproduction des diagrammes UML. La modélisation est généralement orientée autour de trois différents axes sur lesquels les différents diagrammes sont répartis. Il s'agit de :

- l'axe fonctionnel qui décrit les fonctionnalités à mettre en place dans le système ;
- l'axe statique ou structurel qui fait la description de la structure des objets du système donc des données ;
- l'axe dynamique ou opératoire qui décrit le fonctionnement du système.

2.1 Le diagramme de contexte statique

Chaque acteur est relié à une classe centrale unique qui représente le système. Il est aussi spécifié le nombre d'instances de chaque acteur connecté au système. La figure ci-dessous est le diagramme de contexte statique du système modélisé dans ce document.

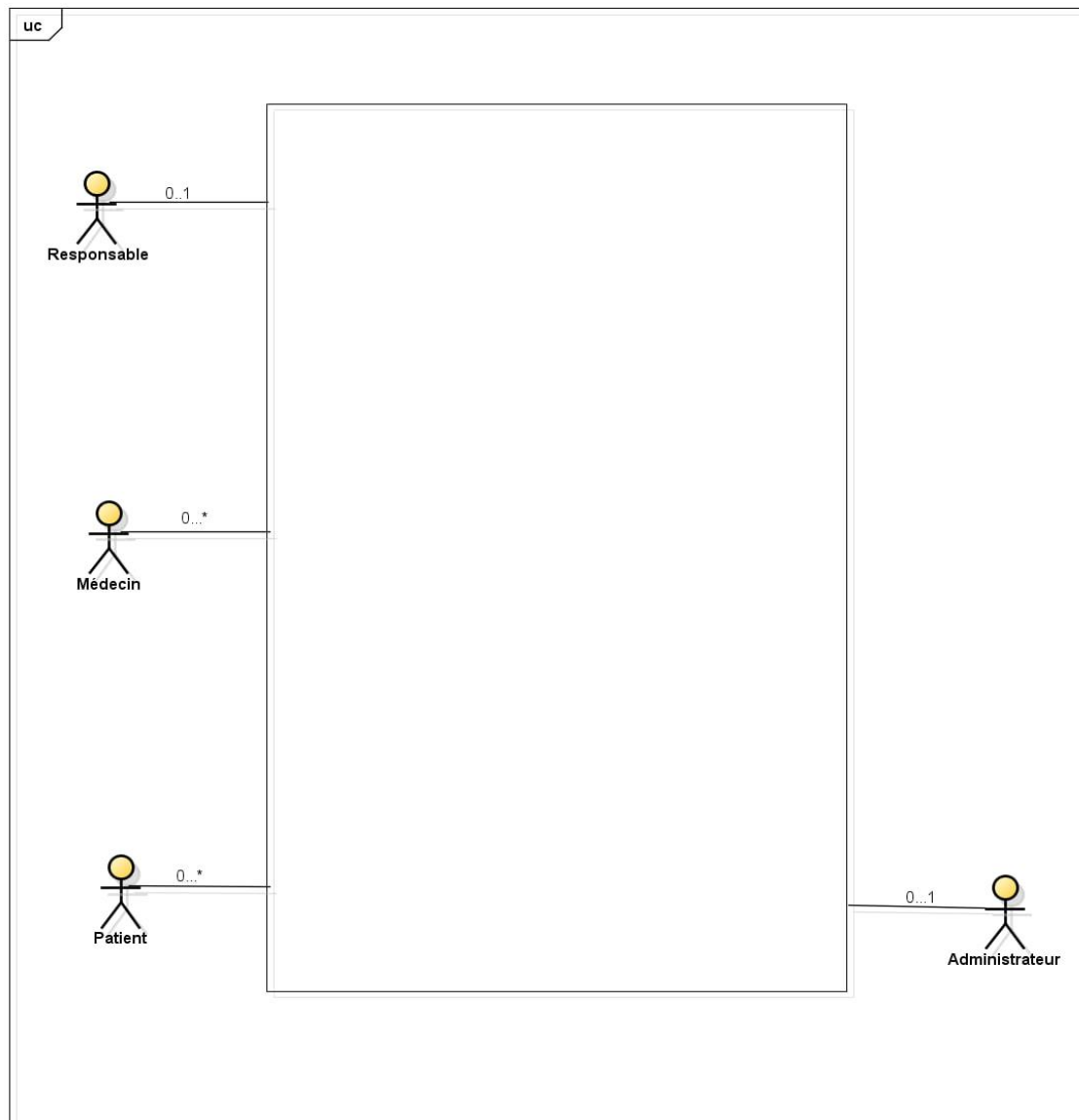


Figure 2 : Diagramme de contexte statique

2.2. Diagramme de cas d'utilisations

Un cas d'utilisation ou use case, est une unité logique d'une possibilité claire de l'extérieur. Il réalise un service de bout en bout avec un commencement, une évolution et une fin, pour l'acteur qui l'initie. On l'identifie en recherchant les différentes interactions avec lesquelles un acteur utilise le système ; en déterminant dans le cahier des charges les services fonctionnels attendus du système. Nous dirons donc que

chaque “cas d'utilisation” correspond à une fonction métier du système. Le diagramme de cas d'utilisation permet la mise en évidence des relations fonctionnelles entre le système et les acteurs. Voici le diagramme de cas d'utilisation de notre système :

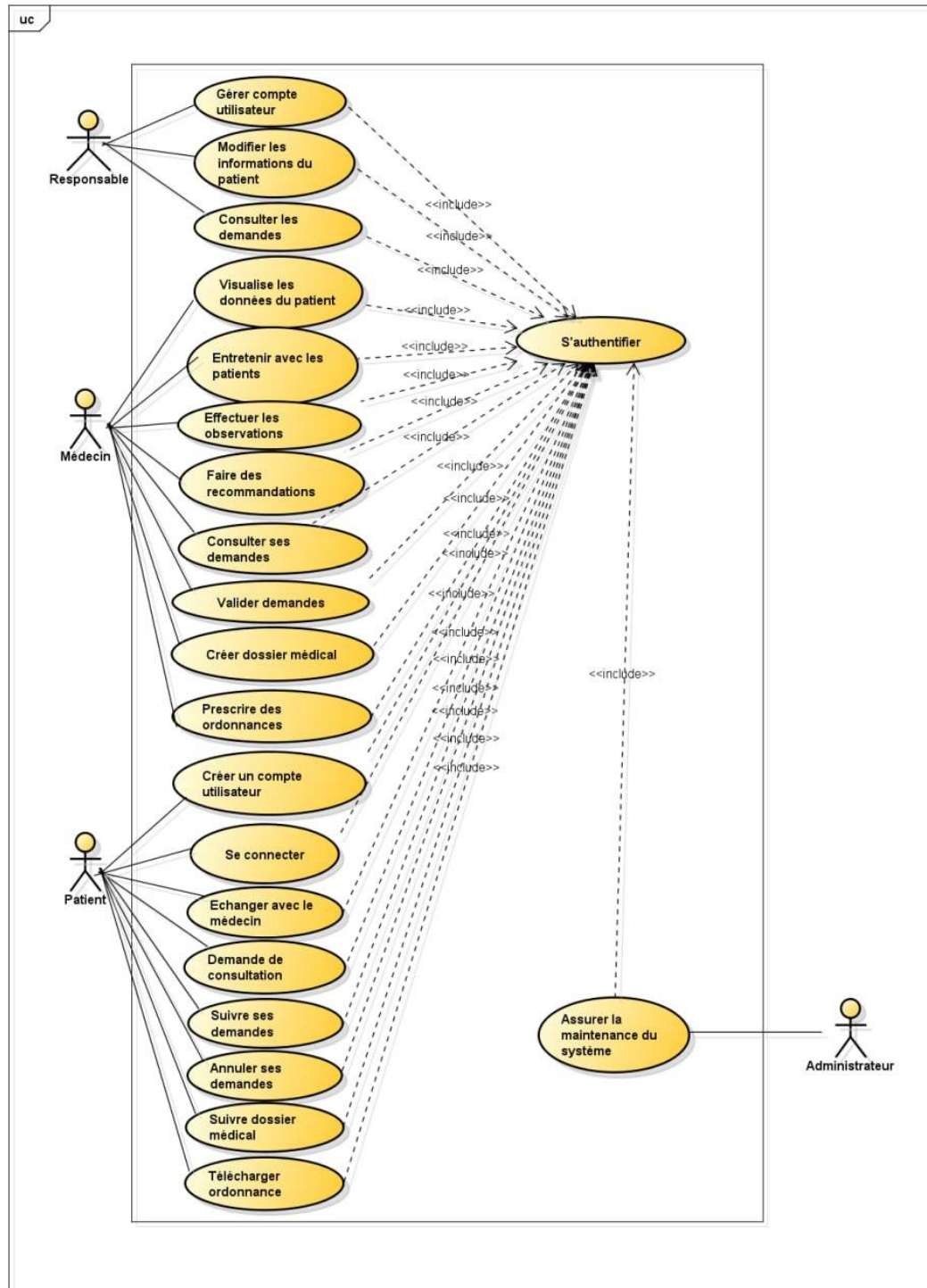


Figure 3 : Diagramme de cas d'utilisation

II. ANALYSE STATIQUE

L'analyse statique est une technique d'analyse de code source qui permet de détecter les erreurs, les bugs, les vulnérabilités de sécurité et autres problèmes potentiels avant même que le code soit exécuté. Elle peut être réalisée manuellement ou à l'aide d'outils automatisés, et elle présente de nombreux avantages en termes de coûts, de qualité et de sécurité du code.

1. Dictionnaire de données

Un dictionnaire des données est l'ensemble des données de référence indispensables à la mise en place d'une base de données relationnelle. C'est une collection des données, de leurs sémantiques, origines, format, utilisation et de leurs relations avec d'autres données. Il est représenté par un tableau et doit respecter les contraintes ci-dessous :

- il ne doit pas y avoir d'homonymes ni de synonymes ;
- les données sont regroupées par entité ;
- les noms des données doivent être monovalués et non décomposables ;
- les identifiants sont complètement précisés ;
- les commentaires doivent être pertinents.

Le dictionnaire des données de modélisation se présente comme suit :

Classes	Attributs	Significations	Types	Longueurs
Utilisateur	Id_utilisateur	Identifiant	Int	5
	Nom	Nom	Vachar	15
	Prenom	Prenom	Vachar	15
	Telephone	Telephone	Vachar	8
	Email	Email	Vachar	25
	Mot_de_passe	Mot de passe	Vachar	20
	Profil	Profil	Vachar	8

Patient	Id_Patient	Identifiant	Int	5
	Date_naissance	Date naissance	Date	8
	Age	Age	Int	5
	Sexe	Sexe	Vachar	8
	Adresse	Adresse	Vachar	25
	Taille	Taille	Int	20
	Groupe_sanguin	Groupe sanguin	Vachar	5
Medecin	Id_medecin	Identifiant	Int	5
	specialite	Specialite	Vachar	20
Dossier_medical	IdDossierMedical	Idendifiant	Int	5
	Poids	Poids	Int	5
	FrequenceCardiaque	Frequence cardiaque	int	5
	Tension_aterielle	Tension arterielle	Int	5
	Taux_glycemie	Taux glycemie	Int	5
	Temperature	Temperature	Int	5
Consultation	Id_consultation	Identifiant consultation	Int	5
	Date_consultation	date consultation	Date	8
Recommandation	Id_recommandation	Identifiant recommendation	Int	5
	Date	Date	Date	8
Observation	Id_observation	Identifiant	Int	5
	Date_observation	Date	Date	8
	Contenu	Contenu	Vachar	25
Ordonnance	Id_ordonnance	Identifiant ordonnance	Int	5

	Date_ordonnance	Date ordonnance	Date	8
	Contenu	Contenue	Vachar	25
Alimentation	Id_alimentation	Identifiant alimentation	Int	5
	Date_alimentation	Date alimentation	Date	8
Programme_exercice	Id_programme	Identifiant programme	Int	5
	Nom_programme	Nom programme	Vachar	25
	description	Description	Vachar	255

Tableau 1 : dictionnaire de données

2. Règles de gestions

Elle précise les contraintes qui doivent être respectées par le modèle. Elles aident à structurer et à réguler le fonctionnement de notre application de suivi de santé, en assurant une utilisation cohérente et sécurisée pour tous les utilisateurs. Dans le cadre de notre étude, les règles de gestion s'intitulent comme suit :

RG1: une adresse email correspond à un et un seul utilisateur.

RG2: chaque dossier médical est associé à un et un seul utilisateur.

RG3: un patient peut demander zéro ou plusieurs consultations.

RG4 : une observation peut être faite lors d'une consultation.

RG5 : une consultation peut donner lieu à une recommandation ou a une ordonnance.

3. Diagramme de classe

Le diagramme de classes est le diagramme le plus important de la modélisation orientée objet avec UML. Il est une représentation de la structure interne du système. Cette vue statique montre les différentes classes du système et les relations qui les lient entre elles. Voici le diagramme de classe du système en cours de modélisation :

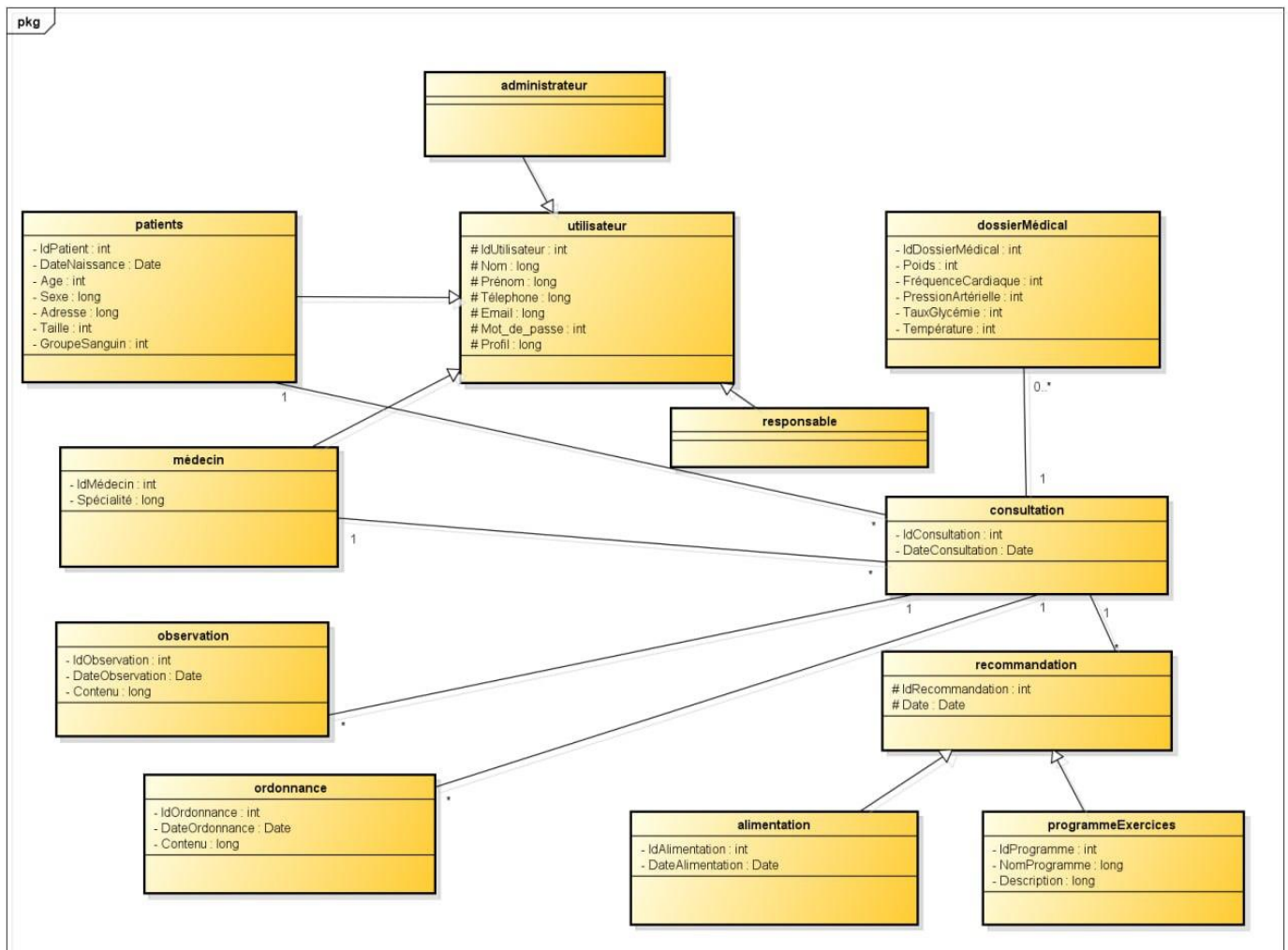


Figure 4: Diagramme de classes

4. Schéma logique relationnel

Utilisateur(id_utilisateur, nom, prenom, telephone, email, mot_de_passe, profil)

Medecin(id_medecin, specialite)

Observation(id_observation, date_observation, contenu, #id_consultation)

Responsable(id_utilisateur, nom, prenom, telephone, email, mot_de_passe, profil)

Dossier_Medical(id_utilisateur, poids, frequence_cardiaque, pression_arterielle, taux_glycemie, temperature, #id_consultation)

Consultation(id_consultation, date_consultation, #id_patient, #id_medecin)

Recommandation(id_recommandation, date, #id_consultation)

Alimentation(id_alimentation, date_alimentation, #id_recommandation)

Programme_exercice(id_programme, nom_programme, description, id_recommandation)

Patient(id_patient, date_naissance, age, sexe, adresse, taille, groupe_sanguin)

Administrateur(id_utilisateur, nom, prenom, telephone, email, mot_de_passe, profil)

III. ANALYSE DYNAMIQUE

1. Description des CAS d'utilisation

Pour réussir ce diagramme, il est primordial d'identifier les acteurs qui sont en contacts direct avec le système. En UML on n'utilise pas le thème d'utilisation mais plutôt d'acteurs. Un acteur d'un système est une entité externe qui interagit avec le système. Les acteurs intervenant dans notre système sont :

- l'administrateur ;
- le responsable ;
- les médecins ;

- les patients.

2. La description textuelle de quelque cas d'utilisations

La description textuelle consiste à détailler chaque cas d'utilisation avec ses scénarios : un scénario nominal (le cas le plus fréquent), les enchainements alternatifs (ou d'extension) et les enchainements d'erreurs (ou d'exception). Chaque scénario est composé d'étapes.

Une étape est un message de l'acteur vers le système, une validation ou changement d'état du système ou encore un message du système vers un acteur

• Cas d'utilisation « S'authentifier »

SOMMAIRE D'IDENTIFICATION

Titre : S'authentifier

Résumé : Ce cas permet à tout utilisateur de s'authentifier.

Acteur (s) : L'utilisateur

Date de création : 11/02/2024

Date de mise à jour : 14/02/2024 Version : 1.0

Responsables : Zéidane MALADE & Imdade MOUTAIROU

Description des scénarios

Préconditions

- l'utilisateur est inscrit ;
- l'utilisateur s'est authentifié.

Scénario nominal

Acteur	Système
1-L'utilisateur demande le formulaire d'inscription	2-le système affiche le formulaire d'inscription
3- L'utilisateur remplit le formulaire	4- le système contrôle et enregistre les informations

Tableau 2 : Cas d'utilisation s'authentifier

Scénario alternatif :

A1- Au point 4, l'identifiant et/ou le mot de passe est incorrect. Le système envoie un message en indiquant à l'utilisateur une erreur/incohérence dans les champs. L'utilisateur corrige les incohérences dans les champs et valide à nouveau. L'enchaînement reprend au point 3 du scénario nominal.

A2- Au point 4 toujours, les champs du formulaire ne sont pas renseignés. Le système envoie un message indiquant à l'utilisateur qu'il n'a pas saisi son email et/ou identifiant. L'utilisateur saisit les informations et valide. L'enchaînement reprend au point 3 du scénario normal.

Postcondition : L'utilisateur s'est enregistré au système.

• Diagramme de Séquence Cas « S'authentifier »

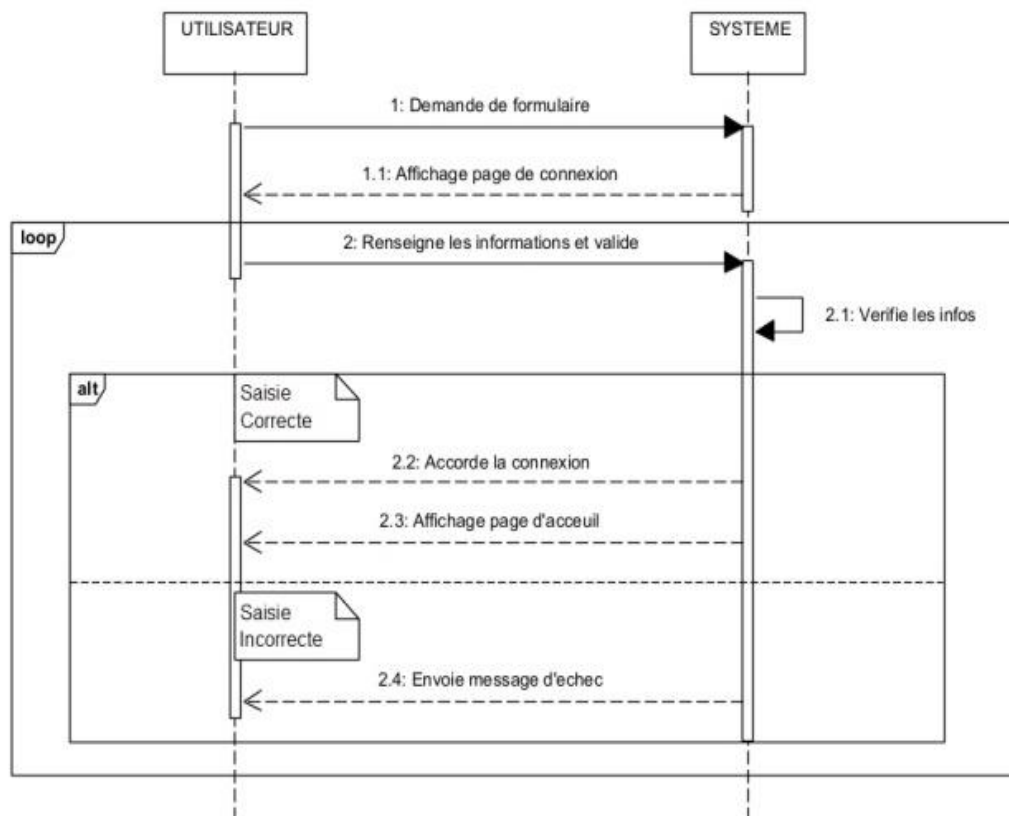


Figure 5 : Diagramme de séquence cas s'authentifier

• Cas d'utilisation « Demande de Consultation »

SOMMAIRE D'IDENTIFICATION

Titre : Demande de consultation

Résumé : Ce cas permet au patient de prendre un rendez-vous

Acteur (s) : Patient.

Description des scénarios

Préconditions

- Le patient doit être connecté au système

Scénario nominal :

Acteur	Système
1-L'utilisateur accède à la page de prise de rendez-vous	2-Le système affiche la liste des spécialités
3-L'utilisateur choisit sa spécialité	4- Le système affiche l'agenda des médecins disponibles
5- L'utilisateur choisit une date et soumet.	6- Le système enregistre et affiche la confirmation du rendez-vous

Tableau 3 : tableau du cas d'utilisation demande de consultation

• Diagramme de Séquence Cas « Demande de consultation »

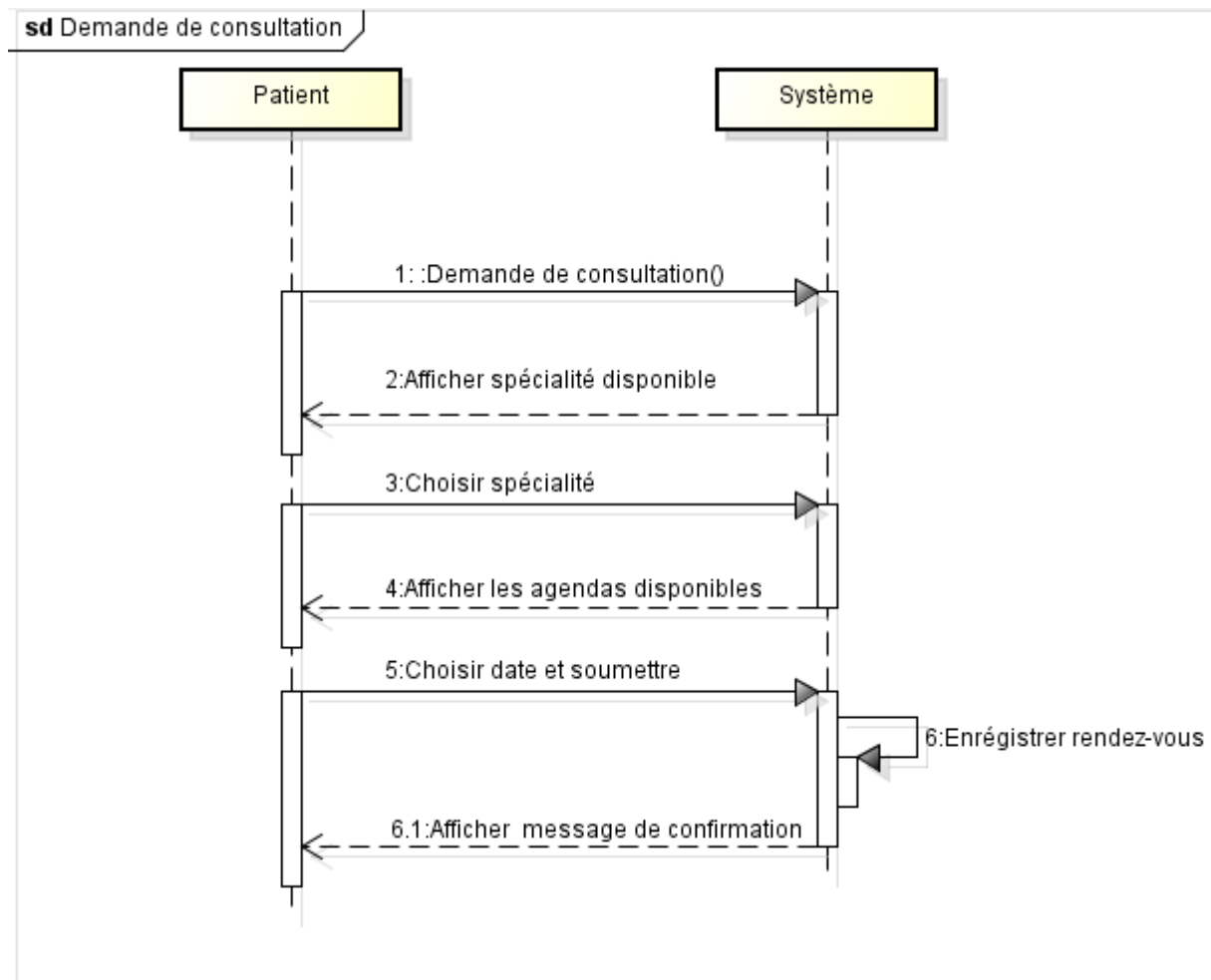


Figure 6 : diagramme de séquence du cas « Demande de consultation »

• Cas d'utilisation « Consulter dossiers médicaux »

Acteur : Médecin

Objectif : Permet au médecin de consulter les antécédents médicaux d'un patient.

Précondition : Le médecin est connecté à son compte sur la plateforme

Tableau 5 : Description textuelle du cas d'utilisation d'accès aux dossiers médicaux des patients

Scénario nominal :

Acteur	Système
1- Appui sur le bouton « consulter dossiers médicaux »	2- Le système affiche un formulaire de recherche de patient
3- Choisir le patient concerné grâce à son identifiant	4- Vérifier si l'identifiant existe et affiche les informations du patient

Tableau 4 : tableau du cas d'utilisation dossiers médicaux

Scénario alternatif :

A1- Au point 4 l'identifiant n'est pas valide, le système envoie un message indiquant au médecin que le champ est mal rempli. Le médecin saisie les informations et valide. L'enchaînement reprend au point 3

A2- Au point 4 l'identifiant n'est pas renseigné, le système envoie un message indiquant au médecin que le champ est vide. Le médecin saisie les informations et valide. L'enchaînement reprend au point 3.

• Diagramme de Séquence Cas « Consulter dossiers médicaux »

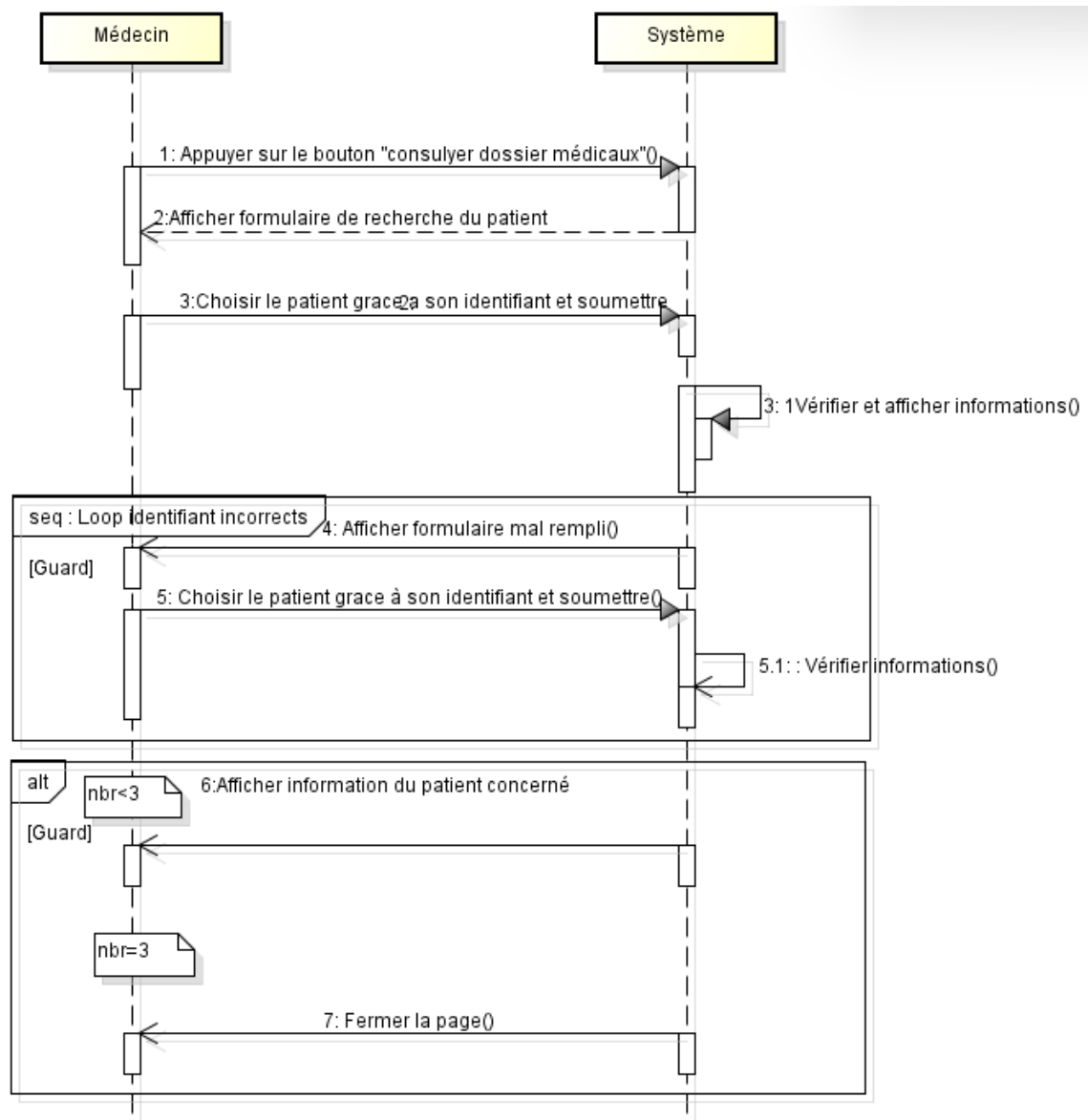


Figure7: diagramme de séquence du cas «Consulter dossiers médicaux »

3. Diagramme d'activité

Le diagramme d'activité représente l'ensemble des actions réalisé par le système, avec tous les branchements conditionnels et toutes les boucles possibles. C'est un graphe orienté d'actions et de transitions. Les transitions sont franchies lors de la fin des actions. Elles représentent le passage d'une activité vers une autre. Un diagramme d'activité est modélisé par des nœuds et des transitions les diagrammes d'activité de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la

modélisation du cheminement de flots de contrôle et de flots de données. Ainsi ils permettent de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas utilisation.

• Cas « éditer ordonnance »

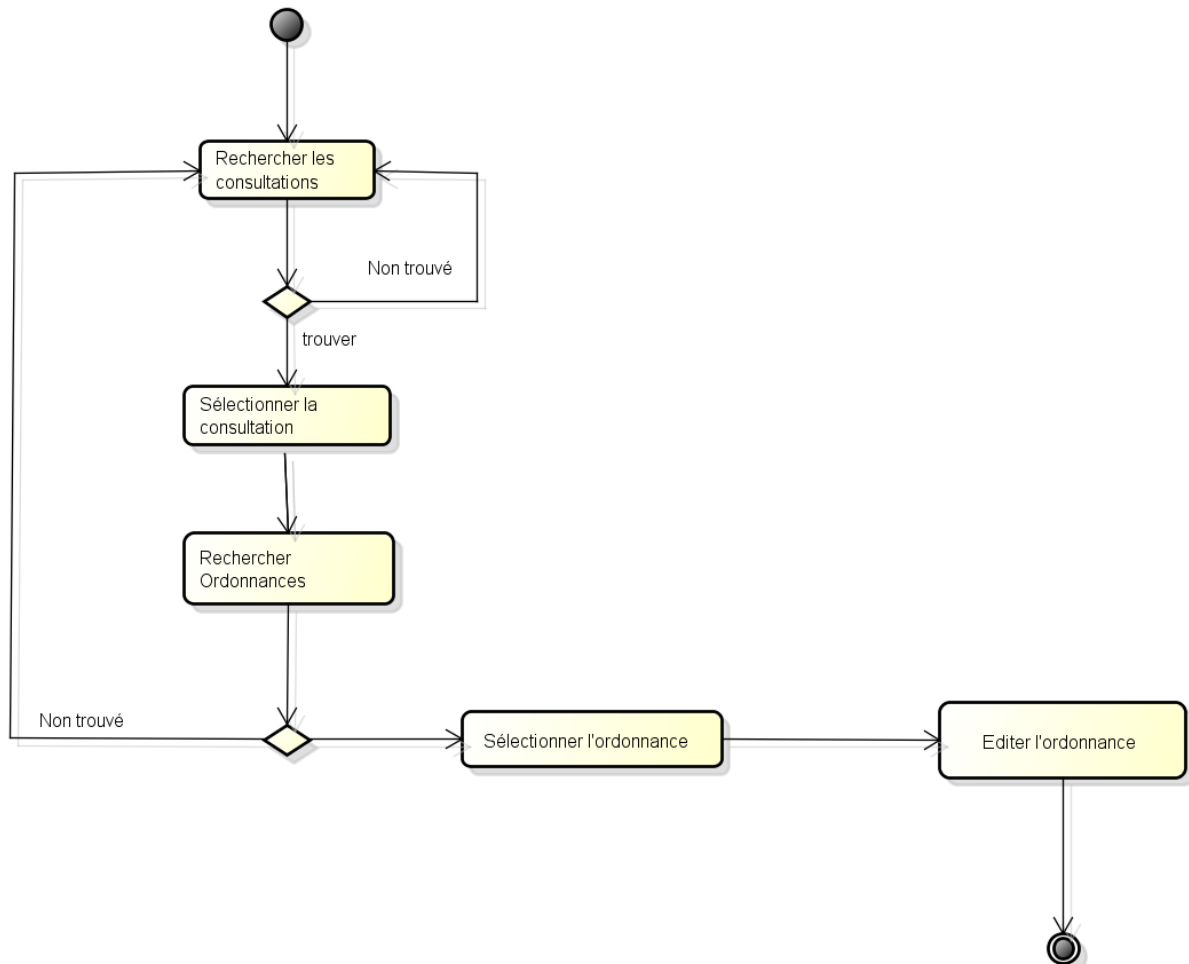


Figure 8: Diagramme d'activité du cas « éditer ordonnance »

• Cas « Enregistrer recommandation »

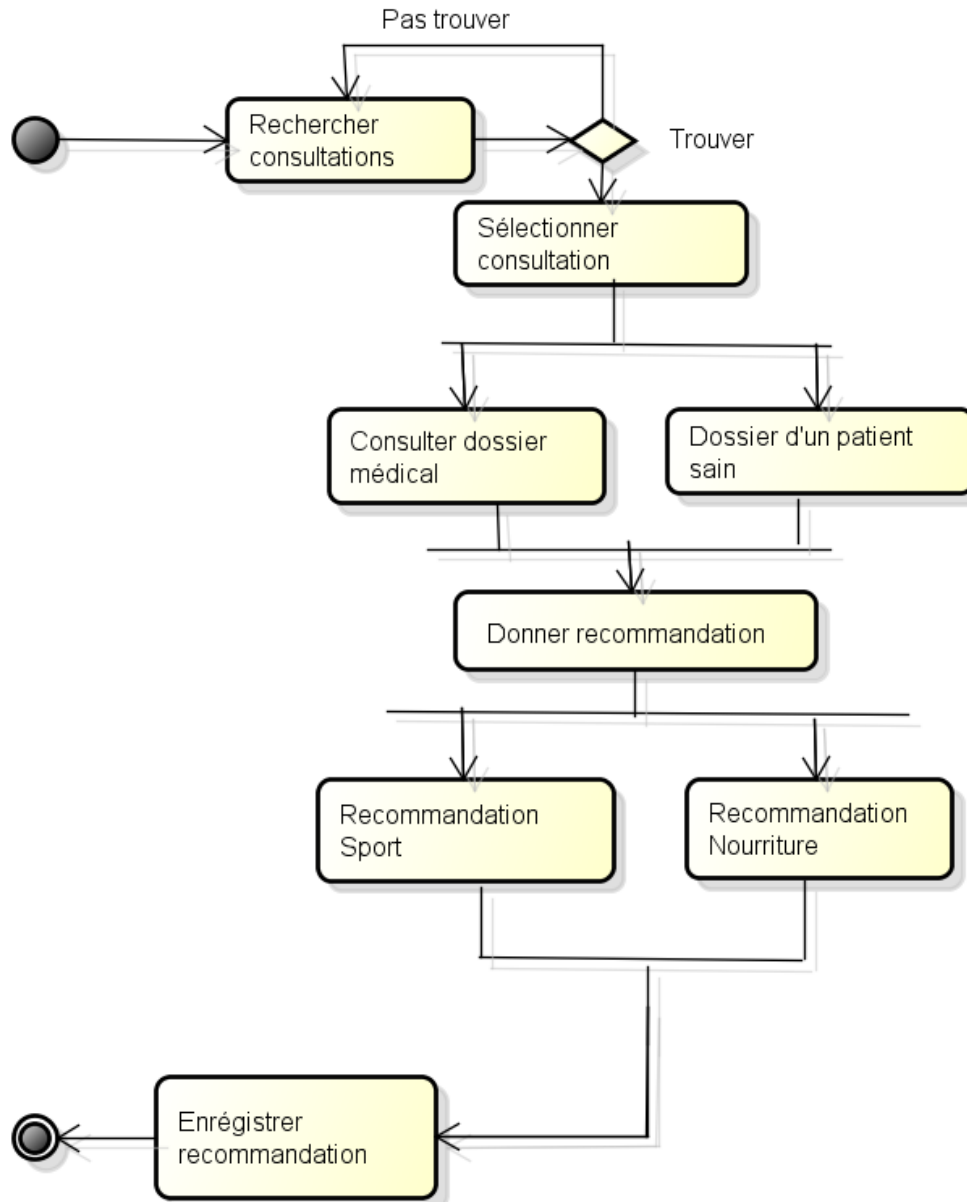


Figure 9 : Diagramme d'activité du cas « Enregistrer recommandation »

Chapitre III :

Réalisation du site web

I. PRESENTATION DES OUTILS ET DES TECHNOLOGIES UTILISEES

1. Présentation des outils SGBD

Le système de gestion de base de données relationnelle utilisée est MySQL (My Structured Query Language). C'est un logiciel libre et open source, distribué sous une double licence GPL et propriétaire. Grace à ses performances élevées, sa fiabilité, sa facilité d'utilisation MySQL s'est imposé comme l'une des solutions favorites pour le développement d'applications web et mobiles nécessitant une base de données. Il utilise le langage SQL pour manipuler les données stockées dans des tables et assurer leur intégrité, avec un support étendu de fonctionnalités avancées.

Serveur

Nous nous sommes servis de XAMPP



Qui est un ensemble de logiciels permettant de mettre en place un serveur web local, un serveur FTP ainsi qu'un serveur de messagerie électronique. Cette plateforme est utilisée pour créer un environnement de développement PHP/MySQL sur sa propre machine, en évitant les complexités de configuration d'un serveur web.

Environnement logiciel



Visual Studio Code: est un éditeur de code source gratuit et open source développé par Microsoft. Il est disponible pour Windows, macOS et Linux. Visual Studio Code est un éditeur léger et puissant qui offre une prise en charge de nombreux langages de programmation. Il comprend également des fonctionnalités avancées telles que le débogage, la mise en évidence de la syntaxe.

2. Présentations des technologies

2.1 Langages utilisés

HTML : Il se définit comme HyperText Markup Langage, ce qui en français signifie « langage de balisage d'hypertexte ». En plus de permettre d'écrire de l'hypertexte, ce langage permet de structurer et de mettre en forme le contenu des pages web. Il est aussi utilisé pour inclure des ressources multimédia telles que des images, des formulaires de saisie ou encore des programmes informatiques. Il faut noter qu'un hypertexte est un document ou un ensemble de documents numériques de type littéraires liés les uns aux autres avec des hyperliens.



CSS (Cascading Style Sheets): c'est un langage informatique qui décrit la présentation des documents HTML et XML. Il permet de contrôler le style et le rendu des éléments contenus dans une page web. En d'autres termes, il est utilisé pour définir l'esthétique des pages web. Les standards qui définissent ce langage sont publiés par le World Wide Web Consortium (W3C). Sa version initiale a été introduite en 1996. CSS devient couramment utilisé pour la conception des sites internet et est bien pris en charge par les navigateurs web dans les années 2000.



PHP : Il se définit comme HyperText Preprocessor. C'est un langage de script utilisé pour la conception de pages web dynamiques via un serveur HTTP. Le code PHP est exécuté côté serveur et permet de générer du HTML et d'interagir avec des bases de données. Le PHP peut également générer des images ou des fichiers PDF. Depuis sa version 5, ce langage dispose de fonctionnalités de modèle objet complètes.



JavaScript: c'est un langage de programmation de scripts principalement

utilisé pour ajouter de l'interactivité et des fonctionnalités avancées aux pages web. Grace à des bibliothèques telles que jQuery, JavaScript rend possible facilement des effets de transition, d'animation. Couplé à HTML et CSS, JavaScript est devenu essentiel dans la conception de sites web modernes et d'applications web réactives. Il est considéré comme l'une des principales technologies du World Wide Web2 aux côtés du HTML et du CSS. JavaScript est utilisé par la majorité des sites web et la plupart des navigateurs web sont en mesure de l'interpréter.



Xml: XML (eXtensible Markup Language) est un langage de balisage polyvalent utilisé pour structurer et stocker des données de manière lisible par les humains et les machines. Conçu pour être extensible, XML offre la possibilité de définir des balises personnalisées et des structures de données spécifiques à un domaine. Cette flexibilité en fait un choix populaire dans le développement logiciel pour représenter des informations structurées dans divers contextes.

2.2. Framework utilisés

Un Framework est une structure logicielle réutilisable qui sert de base pour organiser et développer rapidement des projets en fournissant un ensemble de fonctionnalités et de composants déjà préconstruits. Dans ce projet, nous avons utilisé comme framework :



Bootstrap: C'est la bibliothèque de composants frontend la plus populaire pour développer des projets web responsives et mobiles-first. Open-source et gratuite, elle fournit des outils pour créer rapidement des designs web réactifs, flexibles et compatibles avec les navigateurs modernes. Basée sur HTML, CSS et JavaScript, Bootstrap inclut une collection de composants prêts à l'emploi tels que des boutons, des formulaires, des menus de navigation, des alertes, des carrousels, des modales et bien d'autres. Les développeurs peuvent utiliser les variables de style prédéfinies pour modifier les couleurs, les polices et autres propriétés des composants. Ils peuvent également créer des thèmes personnalisés en modifiant les fichiers CSS

fournis avec Bootstrap. Bootstrap est largement utilisé par les développeurs web pour créer des designs modernes et réactifs. Il est également utilisé dans de nombreuses applications web populaires telles que Twitter, Airbnb et Lyft. Bootstrap est distribué sous la licence MIT, ce qui signifie qu'il peut être utilisé gratuitement pour les projets personnels et commerciaux.

II. ARCHITECTURE DU SYSTEME

1. Présentation de l'architecture

Nous avons utilisé l'architecture MVC (Modèle-Vue-Contrôleur) qui est un modèle de conception qui divise une application en trois parties distinctes : le modèle, la vue et le contrôleur. Le modèle représente les données et la logique métier de l'application. Cela peut inclure des fonctions pour récupérer, stocker, modifier et supprimer des données, ainsi que des règles de validation et de traitement des données. La vue est la représentation visuelle des données pour l'utilisateur. Cela peut être une page web, une interface utilisateur graphique ou tout autre élément qui permet à l'utilisateur de voir et d'interagir avec les données. Le contrôleur est responsable de la gestion des actions de l'utilisateur et de l'interaction entre le modèle et la vue. Il reçoit les entrées de l'utilisateur, effectue les traitements nécessaires sur les données et envoie les résultats à la vue. Enfin, le modèle contient les données et la logique de l'application, la vue est la présentation visuelle de ces données, et le contrôleur est le lien entre les deux, qui gère les actions de l'utilisateur et les modifications apportées aux données.

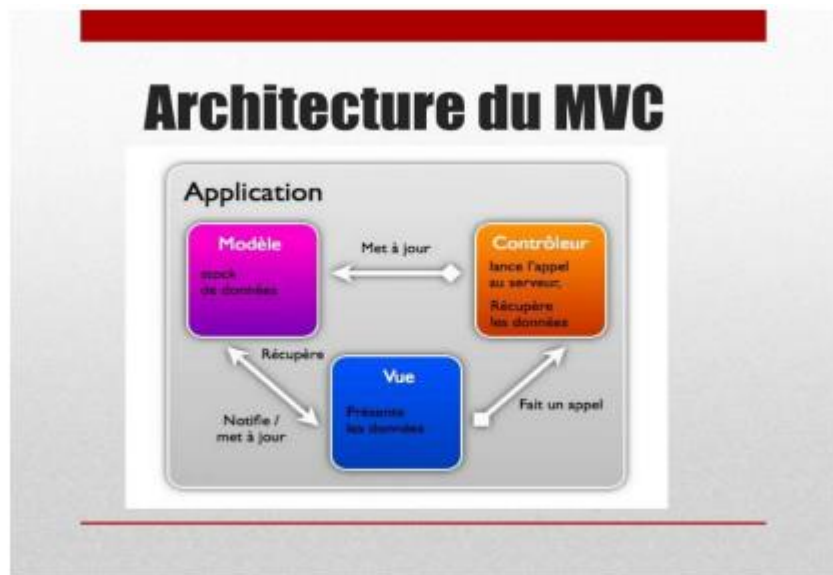


Figure 10 : Schéma de l'architecture de l'MVC

2. Principe de fonctionnement

Le principe de fonctionnement de l'architecture MVC repose sur la séparation des responsabilités de l'application en trois parties distinctes : le modèle, la vue et le contrôleur. Le processus de fonctionnement se déroule généralement comme suit : L'utilisateur interagit avec la vue de l'application (par exemple en cliquant sur un bouton ou en saisissant des données dans un formulaire). La vue envoie les informations d'interaction à travers le contrôleur. Le contrôleur traite les informations reçues et met à jour le modèle en conséquence. Le modèle met à jour ses données en fonction de l'interaction de l'utilisateur. Le contrôleur récupère les données mises à jour à partir du modèle et les renvoie à la vue. La vue est alors mise à jour pour refléter les modifications apportées aux données. Ce découplage permet également de faciliter la maintenance et l'évolutivité de l'application, car chaque partie peut être modifiée ou remplacée sans affecter les autres parties.

3. Avantages et limites

3.1. Avantages de l'architecture MVC (Modèle-Vue-Contrôleur)

L'architecture MVC (Modèle-Vue-Contrôleur) est un modèle de conception de logiciel

qui permet de séparer les différents composants d'une application en trois parties distinctes : le modèle, la vue et le contrôleur. Chaque partie a un rôle spécifique et indépendant, ce qui présente plusieurs avantages :

Séparation des préoccupations : Le modèle, la vue et le contrôleur ont chacun leur propre responsabilité. Le modèle représente la logique de l'application et la gestion des données, la vue gère l'interface utilisateur et le contrôleur assure la coordination entre le modèle et la vue. Cela permet une meilleure organisation et une répartition claire des responsabilités entre les différentes parties.

Maintenance facile : En raison de la séparation claire des différentes parties de l'application, la maintenance et la modification de l'application est plus facile. Si vous avez besoin de modifier la logique de l'application, vous pouvez modifier le modèle sans avoir à toucher à la vue ou au contrôleur.

Meilleure évolutivité : L'architecture MVC facilite l'ajout de nouvelles fonctionnalités à l'application. Vous pouvez ajouter un nouveau modèle, une nouvelle vue ou un nouveau contrôleur sans affecter le reste de l'application.

Collaboration efficace : L'architecture MVC permet une collaboration efficace entre les développeurs. Chaque développeur peut travailler sur une partie spécifique de l'application sans avoir à se soucier du reste.

En résumé, l'architecture MVC permet une meilleure organisation et une meilleure répartition des responsabilités dans une application, facilite la maintenance et la modification de l'application, et permet une collaboration efficace entre les développeurs.

3.2. Limite de l'architecture MVC (Modèle-Vue-Contrôleur)

Bien que l'architecture MVC présente de nombreux avantages, elle présente également quelques limites :

Complexité : L'architecture MVC peut être complexe à mettre en œuvre, en

particulier pour les développeurs débutants. La séparation des préoccupations nécessite une compréhension claire de chaque partie de l'application, ce qui peut être difficile à réaliser pour les débutants.

Couplage : Bien que la séparation des préoccupations soit un avantage, elle peut également entraîner un couplage accru entre les différentes parties de l'application. Le modèle, la vue et le contrôleur doivent travailler ensemble pour que l'application fonctionne, ce qui peut rendre la modification ou la maintenance de l'application plus difficile.

Taille des fichiers : En raison de la séparation des préoccupations, l'architecture MVC peut conduire à la création de nombreux fichiers différents, ce qui peut rendre la gestion de l'application plus difficile.

Inefficacité : Dans certaines situations, l'architecture MVC peut être moins efficace que d'autres architectures, en particulier pour les applications simples qui ne nécessitent pas de séparation claire des préoccupations.

Difficulté à trouver un équilibre : Il peut être difficile de trouver un équilibre entre la séparation des préoccupations et la surcharge de l'application. Si les différentes parties de l'application sont trop séparées, cela peut entraîner une surcharge qui peut ralentir l'application.

En résumé, l'architecture MVC présente des limites telles que la complexité, le couplage, la taille des fichiers, l'inefficacité et la difficulté de trouver un équilibre entre la séparation des préoccupations et la surcharge de l'application. Il est important de peser les avantages et les limites de l'architecture MVC pour déterminer si elle est la meilleure solution pour votre application.

3.3. Script d'implémentation de la base de données

```
CREATE DATABASE hopital ;
```

Structure de la table `alimentation`

```
CREATE TABLE `alimentation` (
  `id_alimentation` int(11) NOT NULL,
  `id_patient` int(11) NOT NULL,
  `date` date NOT NULL,
  `repas` text DEFAULT NULL,
  `calories` int(11) DEFAULT NULL,
  `proteines` float DEFAULT NULL,
  `lipides` float DEFAULT NULL,
  `glucides` float DEFAULT NULL
)ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `consultation`

```
CREATE TABLE `consultation` (
  `id_consultation` int(11) NOT NULL,
  `id_patient` int(11) NOT NULL,
  `id_medecin` int(11) NOT NULL,
  `date_consultation` date NOT NULL,
  `note` text DEFAULT NULL
)ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `dossiermedical`

```
CREATE TABLE `dossiermedical` (
  `id_dossier` int(11) NOT NULL,
```

```
`id_patient` int(11) NOT NULL,
`poids` float DEFAULT NULL,
`frequence_cardiaque` int(11) DEFAULT NULL,
`pression_arterielle` varchar(7) DEFAULT NULL,
`taux_glycemie` float DEFAULT NULL,
`temperature` float DEFAULT NULL,
`date_mesure` date NOT NULL
```

```
)ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `medecin`

```
CREATE TABLE `medecin` (
  `id_medecin` int(11) NOT NULL,
  `nom` varchar(50) NOT NULL,
  `prenom` varchar(50) NOT NULL,
  `specialite` varchar(100) DEFAULT NULL
)ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `observation`

```
CREATE TABLE `observation` (
  `id_observation` int(11) NOT NULL,
  `id_consultation` int(11) NOT NULL,
  `contenu` text DEFAULT NULL,
  `date_observation` date NOT NULL
```

```
) ENGINE=InnoDBDEFAULTCHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `ordonnance`

```
CREATE TABLE `ordonnance` (
```

```
  `id_ordonnance` int(11) NOT NULL,
```

```
  `id_consultation` int(11) NOT NULL,
```

```
  `medicaments` text DEFAULT NULL,
```

```
  `date_ordonnance` date NOT NULL
```

```
) ENGINE=InnoDBDEFAULTCHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `patient`

```
CREATE TABLE `patient` (
```

```
  `id_patient` int(11) NOT NULL,
```

```
  `id_utilisateur` int(11) NOT NULL,
```

```
  `date_naissance` date DEFAULT NULL,
```

```
  `age` int(11) DEFAULT NULL,
```

```
  `sexe` char(1) DEFAULT NULL,
```

```
  `adresse` varchar(255) DEFAULT NULL,
```

```
  `taille` float DEFAULT NULL,
```

```
  `groupe_sanguin` varchar(3) DEFAULT NULL
```

```
) ENGINE=InnoDBDEFAULTCHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

- Structure de la table `programmeexercices`

```
CREATE TABLE `programmeexercices` (
```

```
`id_programme` int(11) NOT NULL,
`nom_programme` varchar(100) NOT NULL,
`description` text DEFAULT NULL,
`duree` int(11) DEFAULT NULL,
`niveau_difficulte` varchar(50) DEFAULT NULL
```

```
)ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `recommandation`

```
CREATE TABLE `recommandation` (
```

```
`id_recommandation` int(11) NOT NULL,
`id_consultation` int(11) NOT NULL,
`recommandation` text DEFAULT NULL,
`date_recommandation` date NOT NULL
```

```
)ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Structure de la table `utilisateur`

```
CREATE TABLE `utilisateur` (
```

```
`id` int(11) NOT NULL,
`nom` varchar(50) NOT NULL,
`prenom` varchar(50) NOT NULL,
`telephone` varchar(15) DEFAULT NULL,
`email` varchar(255) NOT NULL,
`mot_de_passe` varchar(100) NOT NULL,
```

```
`profil` varchar(50) NOT NULL
```

```
)ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Index pour la table `alimentation`

```
ALTER TABLE `alimentation`
```

```
ADD PRIMARY KEY (`id_alimentation`),
```

```
ADD KEY `id_patient` (`id_patient`);
```

Index pour la table `consultation`

```
ALTER TABLE `consultation`
```

```
ADD PRIMARY KEY (`id_consultation`),
```

```
ADD KEY `id_patient` (`id_patient`),
```

```
ADD KEY `id_medecin` (`id_medecin`);
```

Index pour la table `dossiermedical`

```
ALTER TABLE `dossiermedical`
```

```
ADD PRIMARY KEY (`id_dossier`),
```

```
ADD KEY `id_patient` (`id_patient`);
```

Index pour la table `medecin`

```
ALTER TABLE `medecin`
```

```
ADD PRIMARY KEY (`id_medecin`);
```

Index pour la table `observation`

```
ALTER TABLE `observation`
```

```
ADD PRIMARY KEY (`id_observation`),
```

```
ADD KEY `id_consultation` (`id_consultation`);
```

Index pour la table `ordonnance`

```
ALTER TABLE `ordonnance`
```

```
ADD PRIMARY KEY (`id_ordonnance`),
```

```
ADD KEY `id_consultation` (`id_consultation`);
```

Index pour la table `patient`

```
ALTER TABLE `patient`
```

```
ADD PRIMARY KEY (`id_patient`),
```

```
ADD KEY `id_utilisateur` (`id_utilisateur`);
```

Index pour la table `programmeexercices`

```
ALTER TABLE `programmeexercices`
```

```
ADD PRIMARY KEY (`id_programme`);
```

Index pour la table `recommandation`

```
ALTER TABLE `recommandation`
```

```
ADD PRIMARY KEY (`id_recommandation`),
```

```
ADD KEY `id_consultation` (`id_consultation`);
```

Index pour la table `utilisateur`

```
ALTER TABLE `utilisateur`
```

```
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `alimentation`
```

```
MODIFY `id_alimentation` int(11) NOT NULL AUTO_INCREMENT;
```



```
ALTER TABLE `consultation`
```

```
MODIFY `id_consultation` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `dossiermedical`
```

```
MODIFY `id_dossier` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `medecin`
```

```
MODIFY `id_medecin` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `observation`
```

```
MODIFY `id_observation` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `ordonnance`
```

```
MODIFY `id_ordonnance` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `patient`
```

```
MODIFY `id_patient` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=6;
```

```
ALTER TABLE `programmeexercices`
```

```
MODIFY `id_programme` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `recommandation`
```

```
MODIFY `id_recommandation` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `utilisateur`
```

```
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```

```
ALTER TABLE `alimentation`
```

```
ADD CONSTRAINT `alimentation_ibfk_1` FOREIGN KEY (`id_patient`)
```

```
REFERENCES `patient` (`id_patient`);
```

```
ALTER TABLE `consultation`
```

```
ADD CONSTRAINT `consultation_ibfk_1` FOREIGN KEY (`id_patient`)
REFERENCES `patient` (`id_patient`),
```

```
ADD CONSTRAINT `consultation_ibfk_2` FOREIGN KEY (`id_medecin`)
REFERENCES `medecin` (`id_medecin`);
```

```
ALTER TABLE `dossiermedical`
```

```
ADD CONSTRAINT `dossiermedical_ibfk_1` FOREIGN KEY (`id_patient`)
REFERENCES `patient` (`id_patient`);
```

```
ALTER TABLE `observation`
```

```
ADD CONSTRAINT `observation_ibfk_1` FOREIGN KEY (`id_consultation`)
REFERENCES `consultation` (`id_consultation`);
```

```
ALTER TABLE `ordonnance`
```

```
ADD CONSTRAINT `ordonnance_ibfk_1` FOREIGN KEY (`id_consultation`)
REFERENCES `consultation` (`id_consultation`);
```

```
ALTER TABLE `patient`
```

```
ADD CONSTRAINT `patient_ibfk_1` FOREIGN KEY (`id_utilisateur`)
REFERENCES `utilisateur` (`id`);
```

```
ALTER TABLE `recommandation`
```

```
ADD CONSTRAINT `recommandation_ibfk_1` FOREIGN KEY (`id_consultation`)
REFERENCES `consultation` (`id_consultation`);
```

```
COMMIT;
```

II. PRESENTATION DE L'APPLICATION

1. Les captures d'écran

Nous présentons les différentes interfaces de l'application :

- **La page de connexion**

Sur cette page l'utilisateur doit renseigner son e-mail et le mot de passe si son compte existe déjà. Dans le cas contraire, une création de compte est nécessaire et l'utilisateur clique donc sur « S'enregistrer ici »

Se connecter

Entrez votre email and mot de passe pour se connecter

Email

Mot de passe

☐ Rappelle-moi

Se connecter

N'avez-vous pas un compte? [Se connecter](#)

Figure 11 : Page de connexion

• La page d'inscription

Sur cette page, l'utilisateur s'enregistre au cas où il n'aurait pas de compte disponible sur la plateforme et renseigne ses informations dans le formulaire ensuite il clique sur « S'inscrire » et le système de vérification effectue une vérification d'e-mail automatiquement à travers un message mail envoyé par l'application.

The image shows a mobile app registration screen titled "Inscription". At the top, there are three social login buttons: Facebook, Apple, and Google. Below these is the word "ou" (or). The main form consists of several input fields: "Nom" (Name), "Prenom" (First Name), "Telephone" (with a dropdown arrow), "Email", "Confirmation Email", "Mot de passe" (Password), and "Confirmation Mot de passe" (Confirm Password). Below the password fields is a checkbox labeled "J'accepte les conditions et les termes". At the bottom of the form is a dark button labeled "S'inscrire". Below the button, there is a link that says "Avez-vous déjà un compte? S'inscrire".

Figure 12 : Page d'inscription

• La page d'accueil

Que vous soyez un habitué ou un nouveau venu, notre page d'accueil est votre passerelle vers un monde hospitalier.

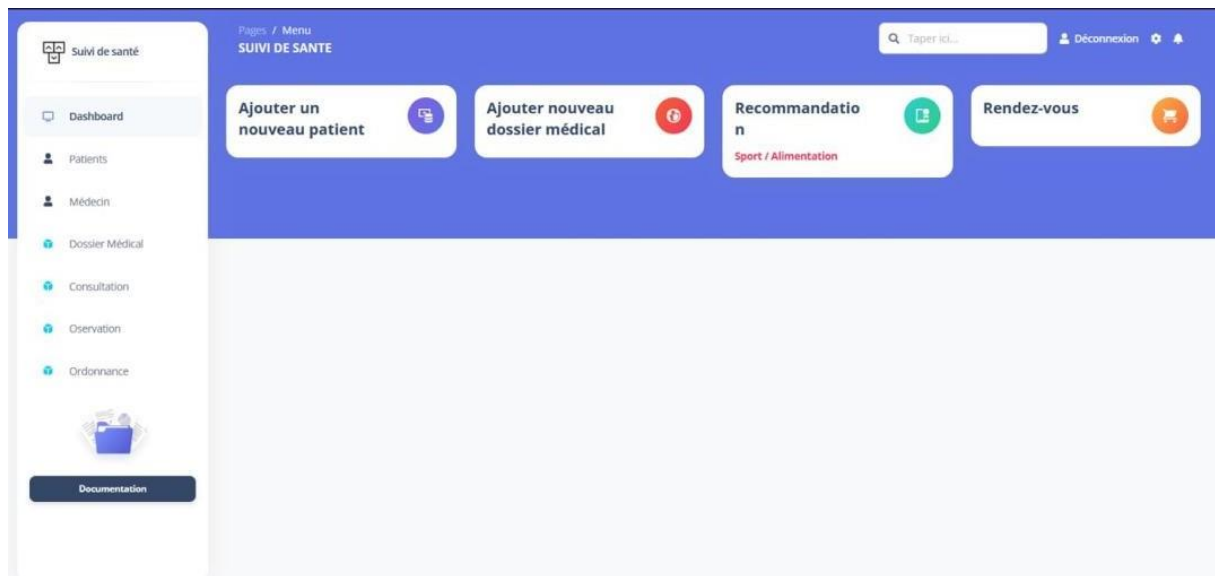


Figure 13 : Page d'Accueil

• La page d'ajout d'un patient

Sur cette page, l'administrateur ajoute des patients.

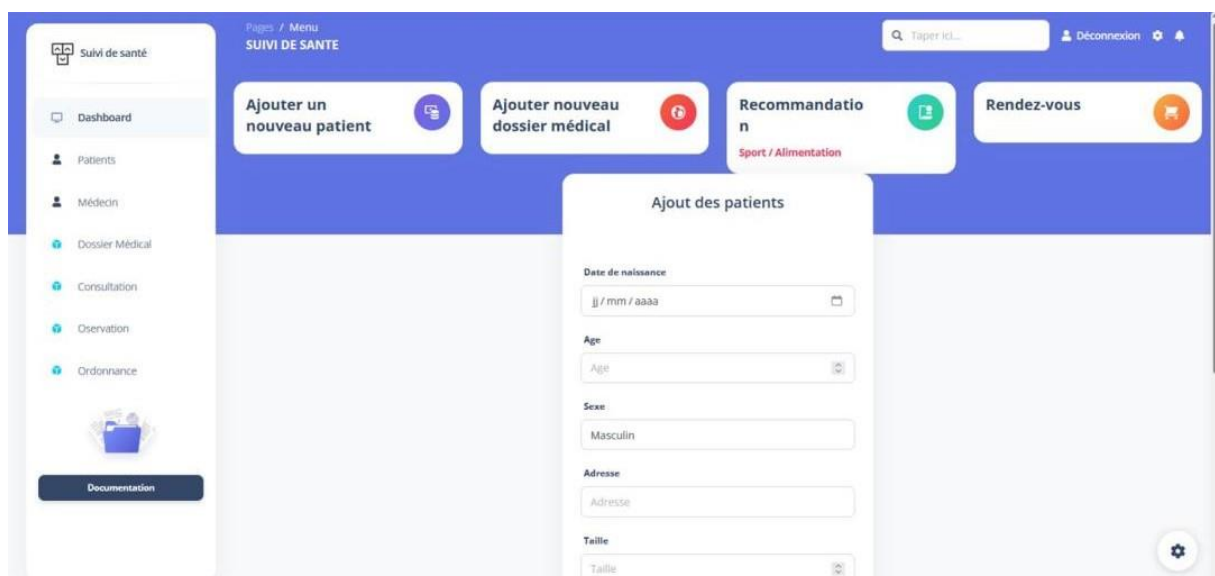


Figure 14 : Page d'Ajout d'un patient

• La page Listes Des patients

Vous pouvez parcourir visuellement la liste des patients, ce qui permis à l'admin

d'ajouter, de modifier et de supprimer des patients.

ID	Date	Age	Sexe	Adresse	Taille	Groupe Sanguin	Nom	Prénom
6	2002-08-15	21	M	Tchiwé	1.69	O	GANIOU	Rouchdan
5	2003-12-05	21	M	Hounsa	1.80	A	AMOOUSSOU	Moufihou
4	2003-03-02	20	M	Avakpa	1.68	B	AHOUANJINO	Hamid
3	2002-06-18	21	M	houlinmé	1.80	A	YACOU	Nassirou
2	2003-07-03	20	M	Cotonou	1.75	AB	MALADE	Zeidane
1	2002-07-05	21	M	Davé	1.88	A	MOUTAIROU	Imdade

Figure 15 : Page Liste des patients

2. Mesure de sécurité et perspective

2.1. Mesure de sécurité

La sécurité informatique est une discipline qui se veut de protéger l'authenticité et l'intégrité des informations stockées dans un système informatique. Quoiqu'il en soit, il n'existe aucune technique capable d'assurer l'invulnérabilité d'un système. La sécurité informatique vise généralement cinq principaux objectifs :

Authenticité : les utilisateurs doivent prouver leur identité par l'usage de code d'accès. Mettre en correspondance un identifiant public avec un secret est le mécanisme permettant de garantir l'authenticité de l'identifiant. Cela permet de gérer les droits d'accès aux ressources concernées et maintenir la confiance dans les relations d'échange.

Intégrité : les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets. Cet objectif utilise généralement des méthodes de calcul de checksum ou de hachage.

Confidentialité : seules les personnes autorisées peuvent avoir accès aux informations qui leur sont destinées (notions de droits ou permissions). Tout accès indésirable doit être empêché.

La non-répudiation et l'imputation : aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées et aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.

Protocole SSL/TLS : Utilisé pour sécuriser les communications sur Internet en chiffrant les données échangées entre un navigateur web et un serveur web. Il assure la confidentialité et l'intégrité des données.

Pare-feu (Firewall) : Un système de sécurité réseau qui contrôle le trafic entrant et sortant en fonction de règles définies, aidant à prévenir les attaques externes et internes.

Systèmes de détection et de prévention des intrusions (IDS/IPS) : Surveillent le réseau ou le système pour détecter et prévenir les activités malveillantes ou les violations de sécurité.

Authentification à deux facteurs (2FA) : Ajoute une couche de sécurité supplémentaire en demandant aux utilisateurs de fournir deux formes d'identification avant de leur permettre l'accès, généralement un mot de passe et un code unique envoyé sur un appareil mobile.

Cryptographie : Utilisée pour chiffrer les données sensibles, garantir l'intégrité des données et authentifier les utilisateurs, contribuant ainsi à assurer la confidentialité et la sécurité des informations échangées sur le web.

2.2. Perspective

Dans le but de satisfaire la clientèle nous nous engageons à améliorer continuellement notre plateforme de livraison des commandes provenant des restaurants locaux a domicile. C'est dans cet optique que nous avons pensés pour les prochaines améliorations à :

- intégré un système de gestion de comptes de fidélité ;

- un système de discussion (chat) entre client et responsable ;
- suivie des commandes après chaque livraison ;
- détermination du coût de livraison en fonction de la localisation ;
- suivie efficaces des restaurants.

CONCLUSION

Le stage académique que nous avons effectué au Centre Informatique de Technologie et de Développement d'Application nous a permis d'acquérir de précieuses compétences dans le domaine du développement de plateformes numériques. Nous avons particulièrement été préparés à relever les défis pratiques de la conception et de la réalisation d'une application dédiée au suivi de santé. Ces trois mois de stage nous ont permis de nous immerger dans la programmation et de concrétiser le site web modélisé dans ce mémoire. Notre objectif principal était de créer un système facilitant le suivi rapide de la santé des utilisateurs et de les satisfaire. Ce mémoire nous a permis d'explorer les défis et les opportunités liés à la conception et à la réalisation d'une application de suivi de santé. Notre projet a démontré la faisabilité technique d'une telle plateforme et a souligné l'importance de la collaboration entre les différents acteurs impliqués, des médecins aux utilisateurs finaux. Avec un engagement continu à l'amélioration et à l'adaptation aux besoins des utilisateurs, cette plateforme a le potentiel de stimuler la visibilité et le bien-être des utilisateurs, contribuant ainsi à dynamiser et à enrichir l'expérience des utilisateurs. Nous espérons que cette plateforme sera un outil précieux pour les médecins désireux de promouvoir leurs établissements et pour les utilisateurs souhaitant découvrir les trésors hospitaliers de leur région.

ANNEXES

```

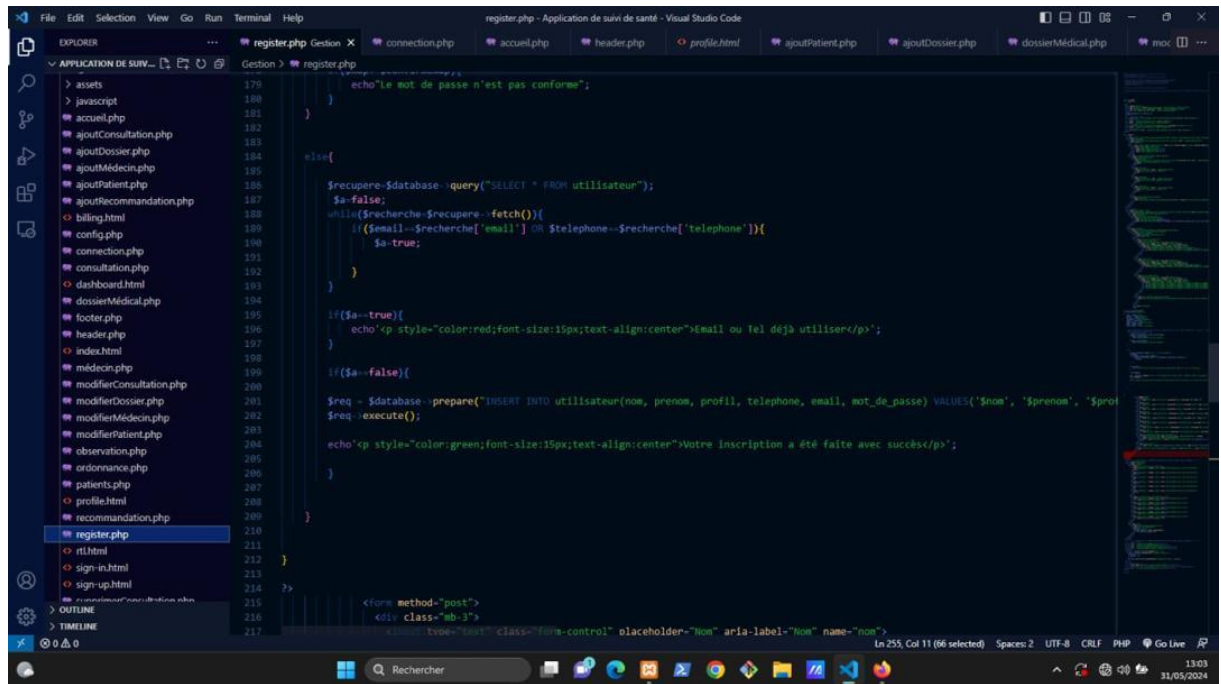
100  $age = $_POST['age'];
101  $sexe = $_POST['sexe'];
102  $adresse = $_POST['adresse'];
103  $taille = $_POST['taille'];
104  $groupe_sanguin = $_POST['groupe_sanguin'];
105  $nom = $_POST['nom'];
106  $prenom = $_POST['prenom'];
107
108
109
110  $req = $database->prepare("INSERT INTO patient( date_naissance, age, sexe, adresse, taille, groupe_sanguin, nom, prenom) VALUES ( '$date
111  $req->execute();
112
113  if($req){
114      echo '<p style="color:green;font-size:15px;text-align:center">Ajout effectué avec succès</p>';
115  }
116  }
117
118  <form method="post">
119
120      <div class="mb-3">
121          <label for="">Date de naissance</label>
122          <input type="date" class="form-control" placeholder="Date de naissance" aria-label="Date de naissance" name="date_naiss">
123      </div>
124      <div class="mb-3">
125          <label for="">Age</label>
126          <input type="number" class="form-control" placeholder="Age" aria-label="Age" name="age">
127      </div>
128      <div class="mb-3">
129          <label for="">Sexe</label>
130

```

```

31  $age = $_POST['age'];
32  $sexe = $_POST['sexe'];
33  $adresse = $_POST['adresse'];
34  $taille = $_POST['taille'];
35  $groupe_sanguin = $_POST['groupe_sanguin'];
36  $nom = $_POST['nom'];
37  $prenom = $_POST['prenom'];
38
39
40  $req = $database->prepare("INSERT INTO patient( date_naissance, age, sexe, adresse, taille, groupe_sanguin, nom, prenom) VALUES ( '$date
41  $req->execute();
42
43  if($req){
44      echo '<p style="color:green;font-size:15px;text-align:center">Ajout effectué avec succès</p>';
45  }
46  }
47
48  <form method="post">
49
50      <div class="mb-3">
51          <label for="">Date de naissance</label>
52          <input type="date" class="form-control" placeholder="Date de naissance" aria-label="Date de naissance" name="date_naiss">
53      </div>
54      <div class="mb-3">
55          <label for="">Age</label>
56          <input type="number" class="form-control" placeholder="Age" aria-label="Age" name="age">
57      </div>
58      <div class="mb-3">
59          <label for="">Sexe</label>
60

```



Bibliographie et Webographie

WEBOGRAPHIE

- <http://bootstrap5.com/> consulté le Mercredi 17 Avril 2024 à 10h ;
- <https://www.monmedenligne.com/> consulté le Jeudi 18 Avril 2024 à 09h ;
- <https://www.w3schools.com/> consulté le Samedi 20 Avril 2024 à 10h ;
- <https://w3layouts.com/> consulté le Samedi 27 Avril 2024 à 15h ;

BIBLIOGRAPHIE

ALAPINI ZOULAIHA & VIDEGLA SIDEGLA Symplice (2023) : « Mémoire de Licence Professionnelle : CONCEPTION ET REALISATION D'UNE PLATEFORME DE PRESTATIONS DE COURS A DOMICILE », HECM.

KEDAGNI Judicaël & GBANHOUME Emmanuel (2022) : « Mémoire de Licence Professionnelle : Développement d'une plateforme de E-Learning : Cas de CFA Empire Computing », UCAO.

LE LIVRE PRENEZ EN MAIN BOOTSTRAP de Maurice CHAVELLI.

Le livre Apprendre Java Script de Enita BORDEAUX UF Informatique.

Table des matières

INTRODUCTION	1
I. PRESENTATION DE CITÉDA	5
1. Historique	5
2. Mission	5
3. Objectifs.....	6
4. Structure Organisationnelle	6
4.1 Président directeur général :	7
4.2 La Direction Exécutive :	7
4.3 La direction des affaires financières :	8
4.4 Direction des communications :	9
II. DEROULEMENT DU STAGE	9
1. Activités effectuées :	10
2. Les apports du stage :	10
III. PRESENTATION DU THEME	10
1. Explication	10
2. Problématique	10
3. Objectif général	11
4. Objectifs spécifiques.....	11
IV. SPECIFICATIONS FONCTIONNELLES.....	12
1. Les besoins fonctionnels	12
2. L'inscription	13
3. L'authentification	13
V. SPECIFICATIONS NON FONCTIONNELLES	13
1. Fiabilité	13
2. Performance	14
3. Sécurité.....	14
4.Disponibilité.....	14
5.Compatibilité	14

I.	ANALYSE FONCTIONNELLE	16
1.	Choix du langage de modélisation	16
2.	Les diagrammes UML	16
2.1	Le diagramme de contexte statique.....	17
2.2.	Diagramme de cas d'utilisations.....	18
II.	ANALYSE STATIQUE	20
1.	Dictionnaire de données	20
2.	Règles de gestions	22
3.	Diagramme de classe.....	23
III.	ANALYSE DYNAMIQUE.....	24
1.	Description des CAS d'utilisation	24
2.	La description textuelle de quelque cas d'utilisations	25
3.	Diagramme d'activité	31
I.	PRESENTATION DES OUTILS ET DES TECHNOLOGIES UTILISEES.....	35
1.	Présentation des outils SGBD	35
2.	Présentations des technologies.....	36
2.1	Langages utilisés	36
2.2.	Framework utilisés	37
II.	ARCHITECTURE DU SYSTEME.....	38
1.	Présentation de l'architecture.....	38
2.	Principe de fonctionnement.....	39
3.	Avantages et limites	39
3.1.	Avantages de l'architecture MVC (Modèle-Vue-Contrôleur)	39
3.2.	Limite de l'architecture MVC (Modèle-Vue-Contrôleur).....	40
3.3.	Script d'implémentation de la base de données.....	44
III.	PRESENTATION DE L'APPLICATION	50
2.	Mesure de sécurité et perspective.....	53
2.1.	Mesure de sécurité.....	53
2.2.	Perspective	54
	CONCLUSION	36
	Bibliographie et Webographie	59