

Projet de fin d'étude pour l'obtention de Licence Informatique

Site web avec système de
Gestion de Clinique

Encadré Par :

Mme Radia BELKZIZ

Réalisé Par :

ELHousseine Touhrache

Maha Adoul

Ahmed Kalaaji

Hicham el mallouki

Oussama Jaabari

Achraf Alfitouri

Dédicace

Nous tenons à dédier ce travail à nos chers parents, à nos familles et nos proches, à nos professeurs et encadrants, à nos ami(e)s et à toute personne qui, de près ou de loin, nous a donné un coup de main d'une façon ou d'une autre ou nous a apporté ses sollicitudes pour accomplir ce travail.

Les mots ne pourront pas exprimer nos profondes gratitude et reconnaissances pour votre soutien et votre

Encouragement. Nous avons fait de notre mieux pour présenter un travail qui va vous rendre fiers de nous, et c'est tout grâce à votre soutien et encouragement que nous sommes arrivées à la réalisation de ce travail que nous souhaitons que vous y trouviez le témoignage de tous nos sentiments de reconnaissance.

Nous vous remercions de tout notre cœur.

Remerciements

Nous tenons à exprimer notre profonde gratitude envers Mme Radia Belakziz pour ses précieux conseils et son encadrement tout au long de ce projet. Nous sommes également reconnaissants envers tous ceux qui ont contribué à sa réussite en nous apportant leur présence, leur soutien, leur disponibilité et leurs conseils, qui nous ont permis de trouver le courage nécessaire pour mener à bien ce projet.

Enfin, nous ne pouvons pas terminer cette reconnaissance sans exprimer notre gratitude envers tous les enseignants de l'Université Privée de Marrakech, qui a été dévoué et ont fourni une assistance précieuse tout au long de cette année. Nous leur sommes pour leur contribution à notre développement académique et professionnel.

Résumé

Notre projet consiste en la création d'un système de gestion de clinique, qui permettra d'optimiser les opérations administratives et médicales d'une clinique. Le système offrira des fonctionnalités de gestion des rendez-vous, de suivi des dossiers médicaux, de gestion des stocks de médicaments et de fournitures, ainsi que des outils de communication pour faciliter la coordination entre les différents membres du personnel.

Nous avons identifié les besoins fonctionnels et non fonctionnels de la clinique, ainsi que les exigences techniques nécessaires pour la mise en place du système. Nous avons également développé une architecture technique qui permettra d'intégrer le système dans l'infrastructure existante de la clinique et de garantir la sécurité des données des patients.

Navigation créé un écran d'application convivial et intuitif pour faciliter la navigation dans le système, ainsi qu'un système de gestion de base de données pour stocker et récupérer les informations des patients et des dossier médicaux.

Sommaire

Introduction Générale	7
I. Chapitre I	8
1. Objectifs	9
2. Planification	10
II. Chapitre II	11
Etude et spécification	11
1. Étude des systèmes de gestion de clinique existants	12
2. Les besoins fonctionnels et non fonctionnels	13
a. Besoins fonctionnels	13
b. Besoins non fonctionnels	13
III. Chapitre III	14
I. Conception	15
1. Les Acteurs	15
2. Diagramme cas d'utilisation	15
3. Diagramme de séquence	16
4. Diagramme de classe	17
IV. Chapitre IV	19
1. Introduction	20
2. L'environnement matériel	20
3. Environnement Logiciel	21
3.1 Les langages	21
4. Exemple de partie code	24
5. Ecran d'application	29
6. Description	30
7. Conclusion générale et propriétés	31
Bibliographie	32

Table de figure

Figure 1:Diagramme de gantt	10
Figure 2: Diagramme cas d'utilisation.....	15
Figure 3:Diagramme de séquence	16
Figure 4:structure des tables	17
Figure 5:Diagramme de classe	18
Figure 6:HTML CSS	21
Figure 7:MySQL.....	21
Figure 8:JavaScript	22
Figure 9:AJAX	22
Figure 10: PHP.....	22
Figure 11 : JQuery	23
Figure 12: Bootstrap	23
Figure 13 : JSon	23
Figure 14: reservation de rendez-vous	24
Figure 15:table d'board d'admin	27
Figure 16 : page login.....	28
Figure 17: Page d'accueil	29
Figure 18: Page d'accueil Partie 2	30

Introduction Générale

La gestion d'une clinique est un processus complexe qui requiert une attention particulière à de nombreux niveaux : de la prise de rendez-vous à la facturation en passant par la gestion des dossiers médicaux. De plus, avec l'évolution constante de la technologie, les cliniques doivent adapter leurs méthodes de travail pour rester compétitives et offrir un service de qualité à leurs patients. Dans ce contexte, nous avons réalisé un projet de fin d'études portant sur le développement d'un système de gestion de clinique informatisé.

L'objectif principal de ce projet est de créer un système de gestion de clinique efficace et facile à utiliser, qui permettra aux professionnels de santé de se concentrer sur

Leur cœur de métier : soigner leurs patients. Ce système offrira des fonctionnalités avancées telles que la gestion des rendez-vous, la consultation des dossiers médicaux, la gestion des factures, la gestion des stocks de médicaments, etc.

L'importance de ce projet réside dans le fait qu'il permettra aux cliniques d'optimiser leurs processus, de gagner en efficacité et d'améliorer la qualité des soins offerts à leurs patients. En outre, ce projet s'inscrit dans une perspective d'amélioration continue, en offrant la possibilité de mettre en place des mises à jour régulières et des améliorations fonctionnelles pour répondre aux besoins évolutifs des cliniques.

Dans la suite de ce rapport, nous allons présenter les différentes étapes de conception et de développement du système de gestion de clinique, ainsi que les résultats des tests réalisés pour évaluer son efficacité et son utilité. Nous allons également discuter des perspectives pour ce projet, en proposant des pistes pour de futures améliorations et évolutions.

I. Chapitre I

Contexte du projet

1. Objectifs

Dans cette partie, nous allons présenter le contexte spécifique du système de gestion de clinique que nous avons développé et les objectifs que nous avons fixés.

Le système de gestion de clinique que nous avons conçu vise à résoudre les problèmes spécifiques rencontrés par les cliniques de notre région en matière de gestion des dossiers médicaux et de la prise de rendez-vous.

Nous avons constaté que de nombreuses cliniques fonctionnent toujours avec des dossiers papier, ce qui peut entraîner des erreurs, des retards et des pertes de documents. De plus, les patients ont souvent des difficultés pour obtenir des rendez-vous en raison de la surcharge de travail des réceptionnistes et du manque de clarté des plannings.

Le système de gestion de clinique que nous avons développé permettra de résoudre ces problèmes en permettant aux cliniques de gérer les dossiers médicaux de manière électronique et de planifier les rendez-vous de manière plus efficace. Les professionnels de santé auront également accès à toutes les informations dont ils ont besoin en temps réel, ce qui réduira les erreurs et les retards dans les soins.

Les objectifs de notre projet sont donc de fournir un système de gestion de clinique qui soit facile à utiliser pour les professionnels de la santé et les patients, de permettre une gestion plus efficace des rendez-vous, de réduire les erreurs médicales et les retards dans les soins, et d'améliorer la satisfaction des patients et des professionnels de la santé.

2. Planification

Diagramme de gantt :

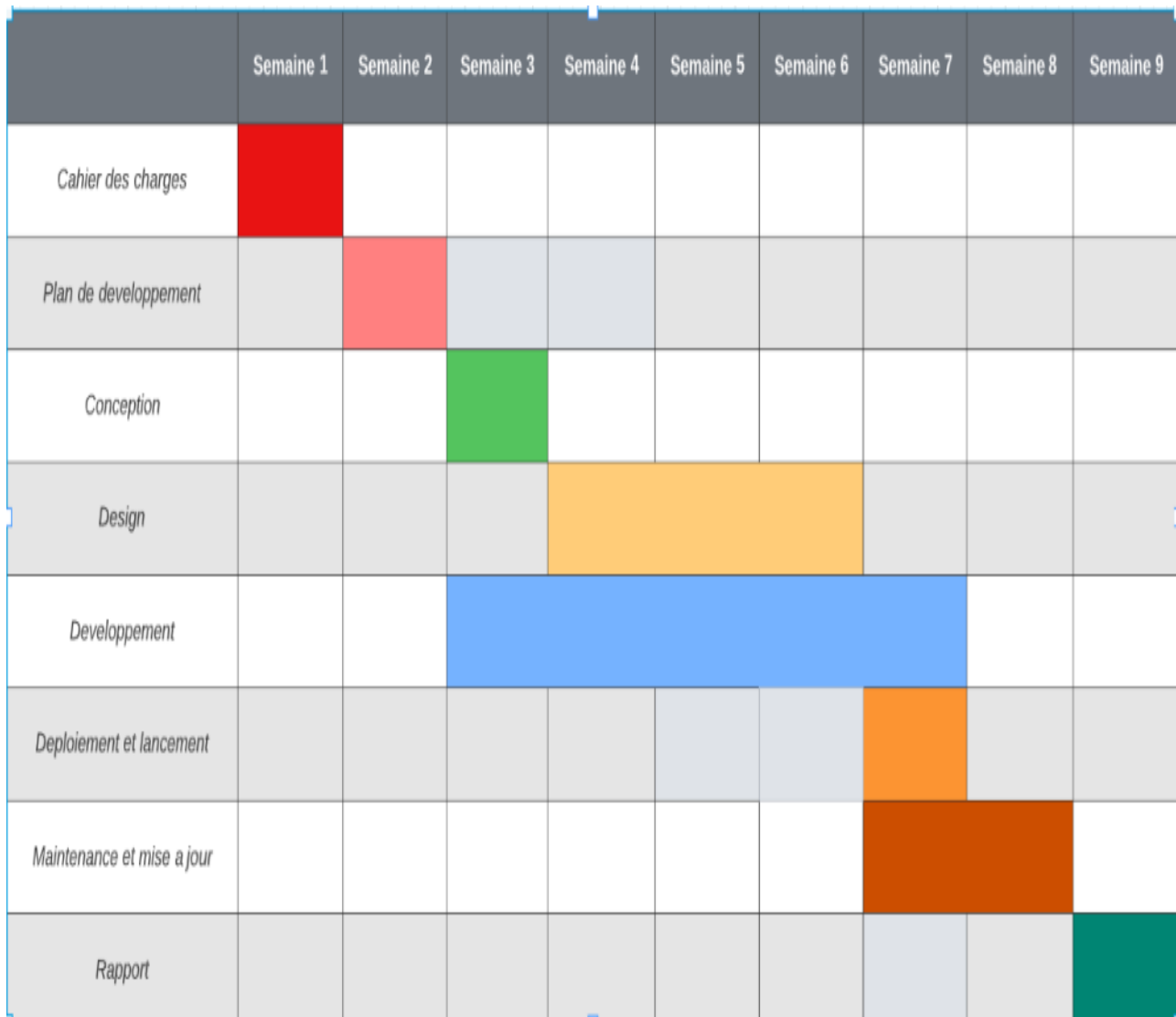


Figure 1:Diagramme de gantt

II. Chapitre II

Etude et spécification

1. Étude des systèmes de gestion de clinique existants

Avant de commencer le développement de notre système de gestion de clinique informatisé, nous avons mené une étude approfondie des systèmes existants sur le marché. Nous avons examiné différentes solutions, allant des logiciels open-source gratuits aux solutions commerciales haut de gamme. Nous avons également consulté des professionnels de santé et des administrateurs de cliniques pour connaître leur expérience avec les systèmes existants.

Nous avons constaté que la plupart des systèmes existants sont limités dans leurs fonctionnalités et ne répondent pas pleinement aux besoins des professionnels de santé et des patients. Les professionnels de santé ont besoin d'un système de gestion de clinique qui leur permet de gérer efficacement les rendez-vous, les dossiers médicaux, les prescriptions, les factures et les rapports. Les patients ont besoin d'un système qui leur permet de prendre rendez-vous facilement, de communiquer avec les professionnels de santé et d'accéder à leurs dossiers médicaux.

Nous avons également constaté que les systèmes existants sont souvent difficiles à utiliser et nécessitent une formation approfondie pour être maîtrisés. De plus, certains systèmes ont des problèmes de sécurité et de confidentialité des données des patients, ce qui peut compromettre leur utilisation.

En fin de compte, nous avons conclu qu'il est nécessaire de développer un système de gestion de clinique informatisé sur mesure pour répondre aux besoins spécifiques des professionnels de santé et des patients de notre région. Nous avons tenu compte des lacunes des systèmes existants et avons élaboré une liste de fonctionnalités essentielles pour notre système. Nous avons également mis en place des mesures de sécurité pour garantir la confidentialité et la sécurité des données des patients.

2. Les besoins fonctionnels et non fonctionnels

a. Besoins fonctionnels

- Gestion des patients : enregistrement des nouveaux patients, création et gestion des dossiers médicaux, suivi des rendez-vous et des traitements.
- Planification et gestion des rendez-vous : permettre aux patients de prendre des rendez-vous en ligne ou par téléphone, planification des rendez-vous pour les médecins, gestion de l'agenda des médecins.
- Gestion des ressources : gestion des salles d'examens, des équipements médicaux, des fournitures et du personnel médical.
- Gestion des médicaments : gestion des stocks de médicaments, des ordonnances, des interactions médicamenteuses et des rappels de renouvellement.

b. Besoins non fonctionnels

- Sécurité : le système doit être sécurisé pour garantir la confidentialité des données des patients, en particulier les informations de santé personnelles.
- Disponibilité : le système doit être disponible en tout temps, pour que les patients puissent prendre des rendez-vous, accéder à leur dossier médical et recevoir des soins de santé.
- Fiabilité : le système doit être fiable pour éviter les interruptions de service ou les erreurs qui pourraient avoir des conséquences graves pour les patients.
- Convivialité : le système doit être facile à utiliser pour les patients et les professionnels de la santé, afin de faciliter la prise de rendez-vous, la gestion des dossiers médicaux et la communication entre les patients et les professionnels de la santé.

III. Chapitre III

I. Conception

1. Les Acteurs

- **Un médecin** : cet acteur est ayant déjà créer un compte sur la plateforme, après l'authentification il peut gérer et choisir les rendez-vous, gérer la pharmacie, patients, l'inventaire et les dossiers médicaux.
- **Un Administrateur** : c'est un acteur responsable de la gestion des patients, médecins, l'inventaire, pharmacie...
- **Un Patient** : c'est un visiteur qui peut prendre un rendez-vous en saisir ses informations et choisir la date et l'heure selon la disponibilité du médecin

2. Diagramme cas d'utilisation

Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système.

Un médecin avant de commencer la gestion faut obligatoirement s'authentifier.
Un admin faut s'authentifier aussi pour continuer la gestion et mettre à jour le contenu de site web.

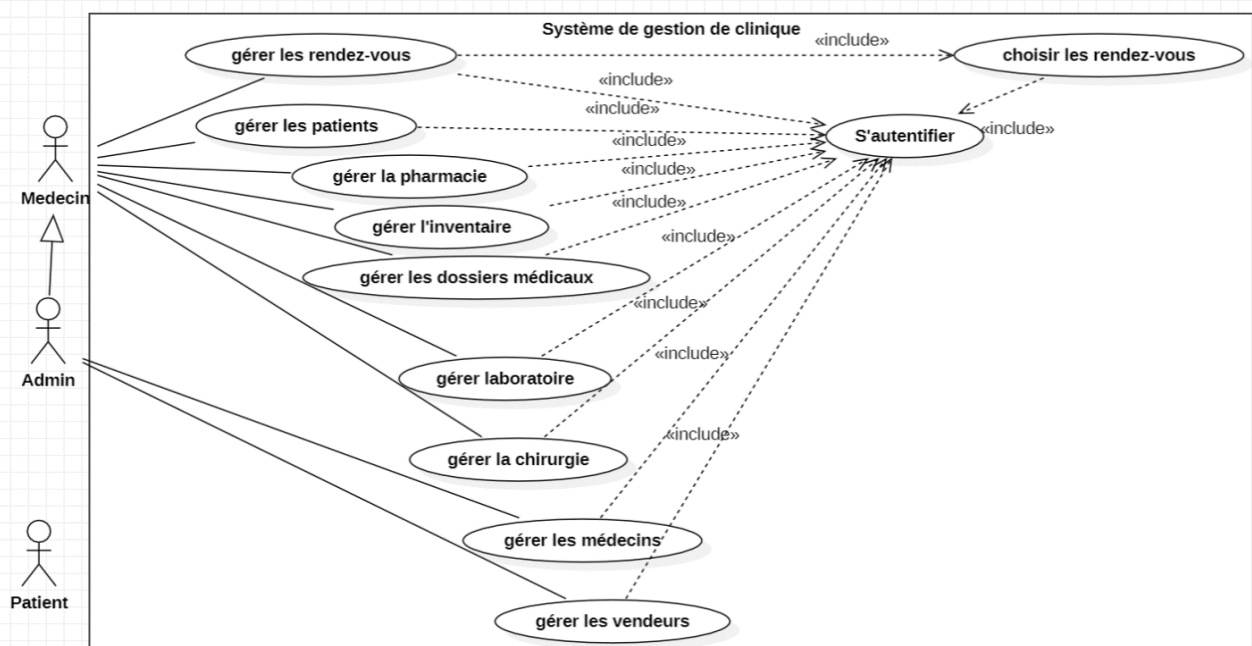


Figure 2: Diagramme cas d'utilisation

3. Diagramme de séquence

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquences sont organisés en fonction du temps qui s'écoule au fur et à mesure que nous parcourons la page.

Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence

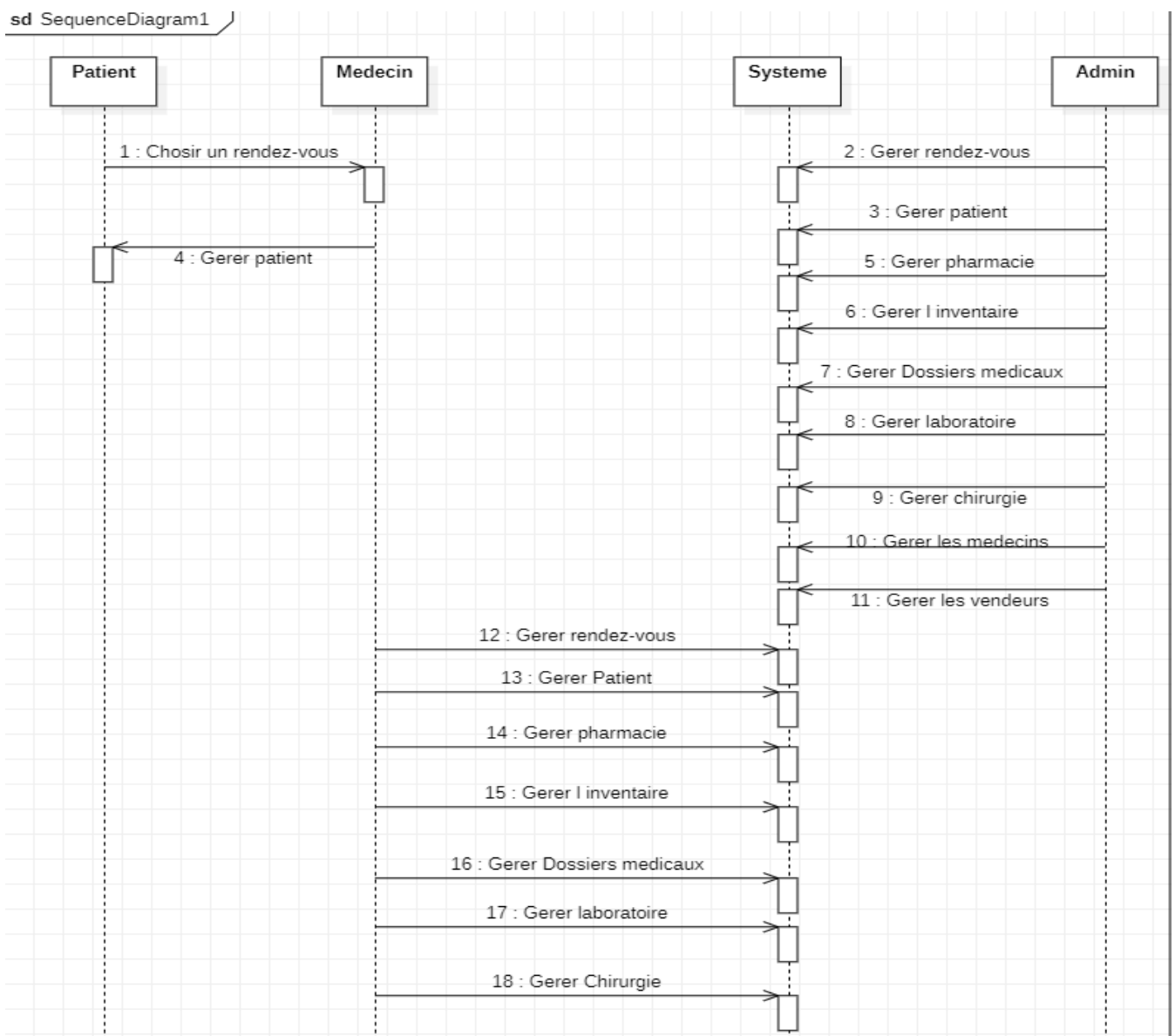


Figure 3:Diagramme de séquence

4. Diagramme de classe

En général un diagramme de classe peut contenir les éléments suivants :

- Les classes : une classe représente la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes. Elle est représentée en utilisant un rectangle divisé en trois sections.
- La section supérieure est le nom de la classe, la section centrale définit les propriétés de la classe alors que la section du bas énumère les méthodes de la classe

Table ▲
<input type="checkbox"/> appointment_list
<input type="checkbox"/> doctors_schedule
<input type="checkbox"/> his_admin
<input type="checkbox"/> his_docs
<input type="checkbox"/> his equipments
<input type="checkbox"/> his_laboratory
<input type="checkbox"/> his_medical_records
<input type="checkbox"/> his_patients
<input type="checkbox"/> his_patient_transfers
<input type="checkbox"/> his_pharmaceuticals
<input type="checkbox"/> his_pharmaceuticals_categories
<input type="checkbox"/> his_prescriptions
<input type="checkbox"/> his_surgery
<input type="checkbox"/> his_vendor
<input type="checkbox"/> his_vitals
15 tables

Figure 4:structure des tables

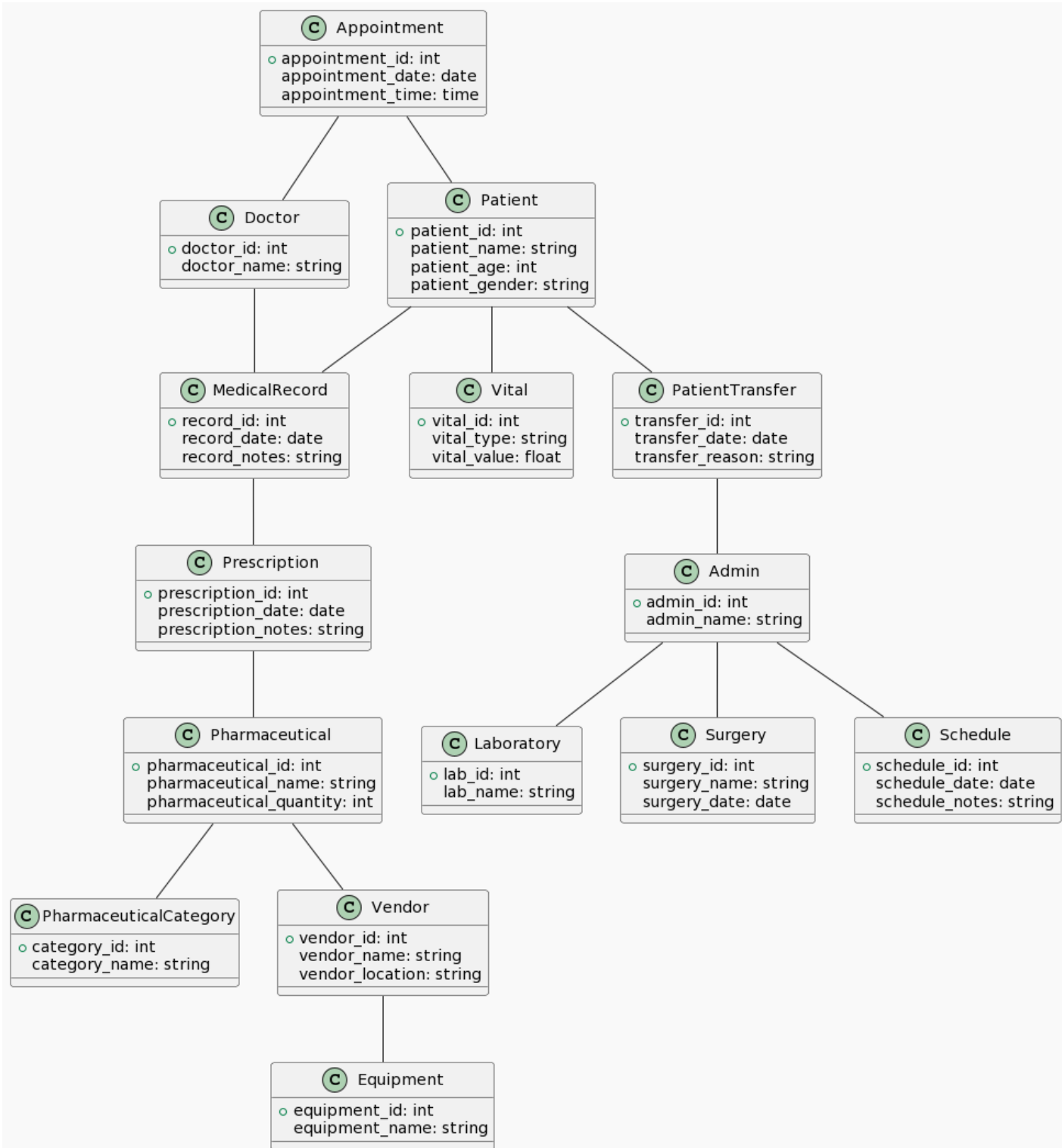


Figure 5: Diagramme de classe

IV. Chapitre IV

1. Introduction

Ce chapitre a pour objectif majeur de présenter le produit final. C'est la phase de réalisation de ce site web dynamique qui utilise des technologies spécifiques.

2. L'environnement matériel

Matériel informatique : un serveur de base de données pour stocker les informations des patients, un ou plusieurs serveurs d'applications pour exécuter le logiciel de gestion de clinique, des ordinateurs pour les utilisateurs (médecins, infirmières, personnel administratif) et des périphériques de stockage pour sauvegarder les données.

Réseau : un réseau local (LAN) pour connecter les ordinateurs de la clinique et les serveurs, ainsi qu'une connexion Internet pour accéder à des ressources externes.

Système d'exploitation : un système d'exploitation serveur, tel que Windows Server ou Linux, pour exécuter le logiciel de gestion de clinique, ainsi qu'un système d'exploitation client pour les ordinateurs des utilisateurs.

Base de données : une base de données relationnelle pour stocker les informations des patients, telle que MySQL ou Microsoft SQL Server.

Interfaces utilisateur : une interface utilisateur conviviale pour permettre aux utilisateurs de naviguer dans le système de gestion de clinique et d'effectuer des tâches telles que la prise de rendez-vous, la gestion des dossiers médicaux électroniques et la gestion des ordonnances.

Composants logiciels : des composants logiciels pour implémenter les différentes fonctionnalités du système, tels que des modules de gestion de rendez-vous, des modules de gestion des dossiers médicaux électroniques, des modules de gestion des prescriptions et des modules de gestion des factures et des paiements.

Protocoles de communication : des protocoles de communication tels que le protocole HTTPS pour la sécurité des données et le protocole HL7 pour l'échange de données avec d'autres systèmes de santé.

3. Environnement Logiciel

3.1 Les langages

➤ Html & CSS

HTML5 (HyperText Markup Language 5) est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014. - Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style

Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

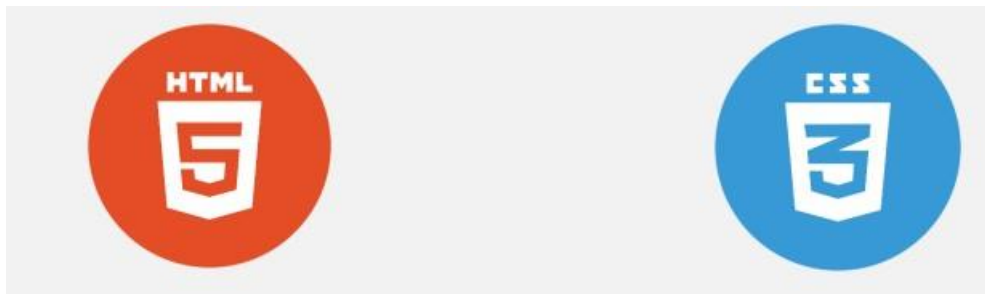


Figure 6:HTML CSS

➤ MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde³, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.



Figure 7:MySQL

➤ JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript5 pour l'interpréter.



Figure 8:JavaScript

➤ Ajax

Ajax est une méthode utilisant différentes technologies ajoutées aux navigateurs web entre 1995 et 2005, et dont la particularité est de permettre d'effectuer des requêtes au serveur web et, en conséquence, de modifier partiellement la page web affichée sur le poste client sans avoir à afficher une nouvelle page complète. Cette architecture informatique permet de construire des applications Web et des sites web dynamiques interactifs.



Figure 9:AJAX

➤ PHP

Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.



Figure 10: PHP

➤ JQuery

JQuery CDN :est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. La première version est lancée en janvier 2006



Figure 11 : JQuery

➤ Bootstrap UI

Bootstrap UI Frameworks : Bootstrap est un framework de développement Web open-source, qui permet de créer des sites Web et des applications Web réactives et mobiles. Il a été créé par Twitter et est maintenant maintenu par la communauté open-source.



Figure 12: Bootstrap

➤ JSon

JSon : JavaScript Object Notation est un format de données textuel dérivé de la notation des objets du langage JavaScript.



Figure 13 : JSon

4. Exemple de partie code

Dans la page d'accueil on a cette forme d'reservation qui contrôler par un ensemble d'fichier, chacun utiliser une méthode pour effectuer la reservation

Rendez-Vous

Choisir ▼

Votre nom

Votre Email

mm/dd/yyyy 

Horaire

Prendre rendez-vous

Figure 14: reservation de rendez-vous

Get-doctors.php : ce fichier récupère des données de la base de données MySQL et les renvoie sous forme d'objet JSON, contenant une liste de médecins et de leurs horaires disponibles, utilisables pour remplir le menu déroulant du formulaire HTML

`\assets\php\get-doctors.php`

```
// Récupérer la liste des médecins dans la base de données
$doctors = $mysqli
    ->query("SELECT doc_id as id, CONCAT(doc_fname, ' ', doc_lname) AS
name FROM his_docs")
    ->fetch_all(MYSQLI_ASSOC);
// Créer un tableau de données
$data = array();
$data["doctors"] = $doctors;
$available = $mysqli
    ->query("SELECT doc_id, day, time_from, time_to FROM
doctors_schedule")
    ->fetch_all(MYSQLI_ASSOC);
$data["available"] = $available;
echo json_encode($data);
$mysqli->close();
```


Get-schedule.php : ce fichier-là est un script PHP qui récupère les horaires disponibles d'un médecin à partir de la base de données MySQL en fonction de l'ID du médecin sélectionné dans un formulaire HTML

\assets\php\get_schedule.php

```
// Obtenir l'horaire du médecin de la table
    $schedule_query = "SELECT * FROM doctors_schedule WHERE doc_id =
".$doctor_id;
    $schedule_result = $mysqli->query($schedule_query);
    $schedule = array();
    while($schedule_row = $schedule_result->fetch_assoc()) {
        $schedule[] = $schedule_row;
    }
```

Process-appointment.php : ce code PHP récupère les données du formulaire envoyé en POST (méthode utilisée pour envoyer des données à un script PHP). Les données récupérées comprennent le nom et l'ID du médecin sélectionné, le nom et l'e-mail du patient, la date et l'heure souhaitées pour le rendez-vous.

\assets\php\process-appointment.php

```
// Insérer les données dans la base de données
$sql = "INSERT INTO appointment_list (doc_id, doctor_name, name, email,
schedule) VALUES ('$doctor', '$doctorName', '$name', '$email', '$date
$time')";
```

Submit.js : ce fichier contient plusieurs fonctions :

1. Contient du code JavaScript qui gère la soumission d'un formulaire en envoyant une requête AJAX au serveur. Il utilise la méthode \$. ajax() pour envoyer les données du formulaire au serveur et traiter la réponse

\assets\js\submit.js

```
$(document).ready(function(){
    // Récupérer la liste des médecins dans la base de données
    $.ajax({
        type: 'GET',
        url: 'assets/php/get-doctors.php',
        dataType: 'json',
        success: function(data){
            const doctors = data.doctors;
            available = [...data.available];
            // Remplissez l'élément sélectionné avec la liste des
```

```

médecins
    $.each(doctors, function(index, {id, name}){

        $('#doctor').append('<option
value=${id}>${name}</option>');
    });
}
});

```

2. Récupère les données du formulaire et les transmet au script serveur via une demande AJAX. Le code ajoute également une valeur pour identifier le médecin sélectionné, puis soumet les données du formulaire au fichier `assets/PHP/process-appointment.php` sur le serveur

```

$('#appointment-form').submit(function(event){
    // Empêcher la soumission normale du formulaire
    event.preventDefault();

    // Obtenir les données du formulaire
    var formData = $(this).serialize();
    var doctorName = $('#doctor option:selected').text();
    formData += '&doctor_name=' + doctorName;

    // Envoyez le formulaire à l'aide d'AJAX.
    $.ajax({
        type: 'POST',
        url: 'assets/php/process-appointment.php',
        data: formData,
        success: function(response){
            // Afficher le message de réussite.
            $('#submit-button').after('<p>Rendez-vous créé avec
succès</p>');
        }
    });
});

```

Et d'autre codes pour autre fonctionnaliste.

Après en a une page pour les deux rôles sur le system « Admin » et « Doctor » c'est la page d'réserveation.

Pour l'admin il s'affiche toutes les réservations pour tous les docteurs.

Et dans l'interface Doctor il s'affiche juste les réservations du docteur connecter.

Admin :

[\backend\admin\his_appointment.php](#)

```
// Récupérer les données de la table appointment_list
$result = $mysqli->query("SELECT * FROM appointment_list ORDER BY
date_created DESC LIMIT $offset, $records_per_page");
?>
```

Doctor :

[\backend\doc\his_appointment.php](#)

```
// Récupérer les données de la table appointment_list
$result = $mysqli->query("SELECT *FROM appointment_list WHERE
doc_id = {$_SESSION['doc_id']} ORDER BY date_created DESC LIMIT
$offset, $records_per_page");
```

Dans le tabl  d'board d'admin on peut ajouter un planning pour chacun Medicine:

Remplir tous les champs

S lectionnez un m decin ▼

Les Jours

Choisir ▼

temps de

Jusqu' 

--:--



--:--



Ajouter un Horaire

Figure 15:tabl  d'board d'admin

Il y a une table sur la base donne pour le planning sous le nom : doctors schedule qui va recupere toutes les informations de la page du planning.

[\assets\php\get_schedule.php](#)

```
if(isset($_POST['add_doc']))
{
    $doctor = $_POST['doc_id'];
    $time_from = $_POST['time_from'];
    $time_to = $_POST['time_to'];
    $day = $_POST['day'];

    $stmt = $mysqli->prepare("INSERT INTO doctors_schedule
(doc_id, day, time from, time to) VALUES (?, ?, ?, ?)");
```

```

$stmt->bind_param('ssss', $doctor, $day, $time_from,
$time_to);
$stmt->execute();
$stmt->close();
}

```

- Login page

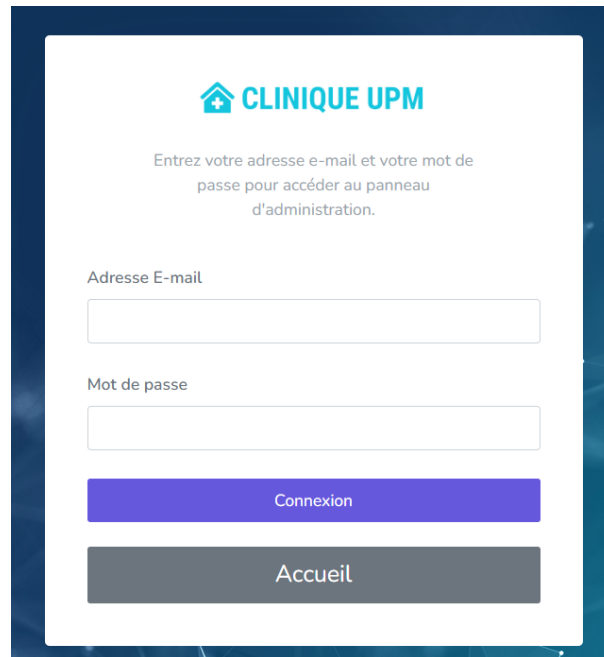


Figure 16 : page login

<backend/admin/index.php>

```

$ad_email=$_POST['ad_email'];
$ad_pwd=sha1(md5($_POST['ad_pwd'])); //double encrypt to increase
security
$stmt=$mysqli->prepare("SELECT ad_email ,ad_pwd , ad_id FROM
his_admin WHERE ad_email=? AND ad_pwd=? "); //sql to log in user
$stmt->bind_param('ss',$ad_email,$ad_pwd); //bind fetched parameters
$stmt->execute(); //execute bind
$stmt -> bind_result($ad_email,$ad_pwd,$ad_id); //bind result
$rs=$stmt->fetch();
$_SESSION['ad_id']=$ad_id; //Assign session to admin id

```

Ce code effectue l'authentification d'un administrateur (admin) sur un site web en PHP. Les étapes sont les suivantes :

- Les variables `$ad_email` et `$ad_pwd` sont initialisées avec les valeurs des champs du formulaire envoyé en POST.
- Le mot de passe est crypté deux fois avec sha1 et md5 pour augmenter la sécurité.
- Une requête SQL est préparée pour sélectionner les champs `ad_email`, `ad_pwd` et `ad_id` de la table `his_admin` où les champs `ad_email` et `ad_pwd` correspondent aux valeurs fournies par l'utilisateur.

- Les paramètres de la requête sont liés aux valeurs des variables \$ad_email et \$ad_pwd.
- La requête est exécutée et le résultat est lié aux variables \$ad_email, \$ad_pwd et \$ad_id.
- Les valeurs de ces variables sont stockées dans la session en utilisant la variable superglobale \$_SESSION['ad_id'], qui stocke l'ID de l'administrateur. Si la requête renvoie une ligne, cela signifie que l'authentification a réussi et que l'utilisateur est redirigé vers la page d'accueil pour les administrateurs.

Le même code est appliqué pour Doctor login

5. Ecran d'application

- Page d'accueil contenant les informations des médecines :

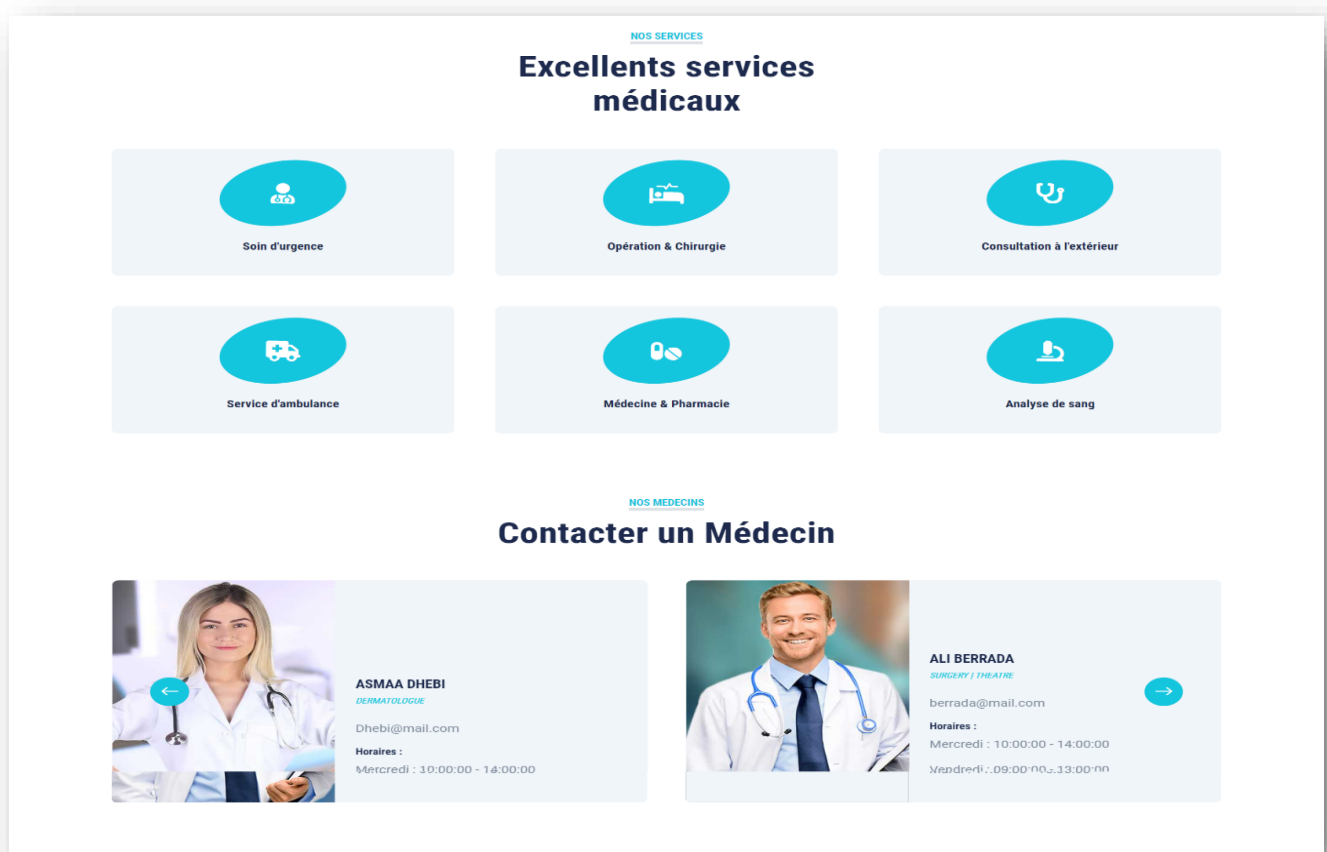


Figure 17: Page d'accueil

RENDEZ-VOUS

Prendre rendez-vous pour un membre de votre famille.

Vous pouvez prendre rendez-vous avec un professionnel de la santé (médecin de famille, infirmière praticienne spécialisée ou médecin président)

[Trouver un Médecin](#)

Rendez-Vous

Choisir Votre nom

Votre Email mm/dd/yyyy

Temp

[Prendre rendez-vous](#)

ENTRER EN CONTACT

Nous sommes là pour vous aider n'hésitez pas à nous contacter afin que nous puissions répondre à votre question ou préoccupation le plus rapidement possible.

[Marrakech Guiliz](#)
[info@example.com](#)
[+212524141400](#)

NEWSLETTER

Votre email ici [s'inscrire](#)

FOLLOW US

[Twitter](#) [Facebook](#) [LinkedIn](#) [Instagram](#)

Figure 18: Page d'accueil Partie 2

6. Description

Notre projet, est un système de gestion des patients cliniques. Il s'agit d'un projet d'application Web développé en PHP et MySQL Database. L'objectif principal de ce projet est de fournir une plateforme en ligne et automatiser pour la gestion des données de certaines cliniques médicales. Ce système permet à la clinique de stocker, de récupérer et de gérer facilement les dossiers de visite et de prescription de ses patients. Il dispose d'une interface utilisateur agréable à l'aide du Bootstrap Framework et du modèle AdminLTE qui offre une meilleure expérience aux utilisateurs finaux. Il se compose également de multiples fonctionnalités conviviales.

7. Conclusion générale et propriétés

En conclusion, un système de gestion de clinique est un outil indispensable pour toute clinique moderne qui souhaite améliorer l'efficacité de ses opérations et la qualité de ses soins. Cependant, il est important de noter que chaque clinique a des besoins spécifiques et uniques, qui peuvent nécessiter des fonctionnalités personnalisées dans leur système de gestion de clinique.

Les propriétés d'un système de gestion de clinique efficace incluent la facilité d'utilisation, la personnalisation, la sécurité des données des patients, la compatibilité avec d'autres systèmes de santé, l'interopérabilité et la mise à jour régulière pour inclure les dernières technologies et améliorations.

En outre, un système de gestion de clinique bien conçu peut offrir des avantages supplémentaires tels que la réduction des coûts administratifs, l'optimisation de la planification des rendez-vous, l'amélioration de la coordination et de la communication entre les différents membres du personnel, ainsi que la réduction des erreurs médicales et des problèmes de confidentialité des données.

Il est donc essentiel pour les cliniques de choisir un système de gestion de clinique qui répond à leurs besoins spécifiques et de s'assurer que ce système est bien conçu, sécurisé et facile à utiliser pour les membres du personnel et les patients. En fin de compte, un système de gestion de clinique efficace peut aider à améliorer l'efficacité et la qualité des soins offerts aux patients et à assurer le succès à long terme de la clinique.

Bibliographie

<https://www.youtube.com/@freecodecamp>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://github.com/>

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

<https://www.php.net/manual/en/>

<https://sql.sh/>

https://www.youtube.com/playlist?list=PLbhHxQMrsDNjgwkb51IKA_BXoy4u5H3RH

<https://visualstudio.microsoft.com/>

<https://www.apachefriends.org/>