

Cahier des Charges Complet

Système d'Information Hospitalier

Application Web de Gestion Hospitalière

L3 MIAGE 2025-2026
Laura MRIDHA et Célia CHIOUKH

Table des matières

I Analyse du Besoin	5
1 Contexte	5
2 Problème à Résoudre	5
3 Objectifs du Système d'Information	5
4 Acteurs	5
5 Fonctionnalités Principales	6
6 Contraintes Techniques	6
II Étude d'Opportunité	6
7 Contexte et Enjeux	6
8 Intérêt du Projet	6
9 Analyse des Options	7
9.1 Option Retenue	7
10 Conclusion	7
III Étude de Faisabilité	7
11 Faisabilité Technique	7
12 Faisabilité Économique	7
13 Faisabilité Organisationnelle	7
14 Conclusion	8
IV Plan de Projet	8
15 Objectif	8
16 Étapes et Jalons	8
17 Ressources	8
17.1 Ressources Humaines	8
17.2 Ressources Matérielles	9
18 Coût Estimé	9

19 Risques et Mesures de Prévention	9
20 Suivi du Projet	9
21 Conclusion	9
V Spécifications Fonctionnelles	9
22 Spécifications Fonctionnelles Générales (SFG)	10
22.1 Gestión des Patients	10
22.2 Gestión des Médecins	10
22.3 Gestión des Rendez-vous	10
22.4 Gestión des Consultations	10
22.5 Authentification et Rôles	10
22.6 Recherche et Filtrage	10
22.7 Interface Utilisateur	10
23 Spécifications Fonctionnelles Détaillées (SFD)	11
23.1 Ajouter un Patient	11
23.1.1 Règles de gestion	11
23.1.2 Scénario principal	11
23.1.3 Scénario alternatif	11
23.2 Ajouter un Médecin	11
23.2.1 Règles de gestion	11
23.2.2 Scénario principal	11
23.3 Planifier un Rendez-vous	11
23.3.1 Règles de gestion	11
23.3.2 Scénario principal	11
23.3.3 Scénario alternatif	11
23.4 Ajouter une Consultation	12
23.4.1 Règles de gestion	12
23.4.2 Scénario principal	12
23.5 Authentification et Rôles	12
23.5.1 Règles de gestion	12
23.5.2 Scénario principal	12
VI Dossier Technique	12
24 Architecture du Système	12
24.1 Modules Principaux	12
25 Technologies Utilisées	13
26 Contraintes Techniques et Sécurité	13
26.1 Sécurité	13
26.2 Performance	13
26.3 Compatibilité	13

27 Architecture Logique	13
28 Arborescence du Projet	14
29 Diagrammes UML	15
29.1 Diagramme MCD (Modèle Conceptuel de Données)	15
29.2 Diagramme Use Case (Cas d'Utilisation)	16
29.3 Diagramme des Classes	17
30 Maquettes et Prototypes	18
30.1 Page de Connexion	18
30.2 Tableau de Bord	19
30.3 Gestion des Patients	20
30.4 Gestion des Rendez-vous	21
30.5 Nouvelle Consultation	21
VII Structure de la Base de Données	22
31 Crédation de la Base de Données	22
32 Tables Principales	22
32.1 Table : specialites	22
32.2 Table : medecins	23
32.3 Table : patients	23
32.4 Table : rendez_vous	23
32.5 Table : consultations	24
32.6 Table : disponibilites	24
32.7 Table : utilisateurs	25
33 Tables de Facturation	25
33.1 Table : factures	25
33.2 Table : ligne_facture	26
33.3 Table : paiements	26
34 Tables de Services et Hospitalisations	26
34.1 Table : services	26
34.2 Table : lits	26
34.3 Table : hospitalisations	27
35 Tables Tarifs et Assurances	27
35.1 Table : tarifs	27
35.2 Table : assurances	27
VIII Guide d'Installation	28
36 Prérequis	28
37 Installation du Projet	28

37.1 Télécharger le Projet	28
37.2 Configuration de la Base de Données	28
37.2.1 Créer la Base de Données	28
37.2.2 Importer la Structure et les Données	29
37.3 Configuration du Fichier database.php	29
38 Lancement de l'Application	29
38.1 Démarrer le Serveur	29
38.2 Ouvrir l'Application	29
39 Problèmes Fréquents et Solutions	29
IX Support et FAQ	29
40 Questions Fréquentes	30
40.1 Je n'arrive pas à me connecter	30
40.2 La page est blanche ou un bouton ne marche pas	30
40.3 Impossible d'ajouter un patient ou un rendez-vous	30
40.4 Les données ne s'enregistrent pas	30
41 Plan de Tests	30
41.1 Objectif	30
41.2 Types de Tests	30
41.3 Rapports de Tests	30
42 Site Web - Gestion Hospitalière	30

Première partie

Analyse du Besoin

1 Contexte

L'hôpital souhaite moderniser la gestion de ses informations médicales et administratives. Actuellement, les données relatives aux patients, médecins et rendez-vous sont gérées de manière manuelle (papier ou tableau Excel).

Ce mode de fonctionnement entraîne des pertes de temps, des risques d'erreurs et une difficulté d'accès aux informations. L'objectif est donc de concevoir une **application web de gestion hospitalière simplifiée**, permettant d'automatiser ces processus et de garantir la fiabilité et la centralisation des données.

2 Problème à Résoudre

La gestion manuelle des dossiers patients et des rendez-vous médicaux pose plusieurs problèmes :

- Difficulté à retrouver rapidement les informations d'un patient ou d'un médecin.
- Manque de coordination entre les services.
- Risques d'erreurs dans la planification des rendez-vous (doublons, oublis).
- Absence d'un historique des consultations.

Le futur système d'information devra automatiser ces tâches pour améliorer la productivité et fiabiliser la gestion.

3 Objectifs du Système d'Information

1. Enregistrer et gérer les informations des patients (identité, contact, informations médicales).
2. Gérer la base de données des médecins (nom, spécialité, horaires, statut).
3. Planifier et suivre les rendez-vous médicaux (date, heure, patient, médecin).
4. (*Optionnel*) Enregistrer les consultations effectuées.
5. Permettre la recherche, la modification et la suppression des données.
6. (*Optionnel*) Ajouter une authentification avec différents rôles (admin, secrétaire, médecin).
7. Gérer les rendez-vous.

4 Acteurs

Administrateur

Gère tout (patients, médecins, rendez-vous).

Secrétaire

(*optionnel*) Planifie les rendez-vous.

Médecin (*optionnel*) Consulte son planning.

5 Fonctionnalités Principales

- **CRUD patients** (Create, Read, Update, Delete)
- **CRUD médecins**
- **CRUD rendez-vous**
- Recherche / filtrage
- Interface simple et intuitive

6 Contraintes Techniques

- **Langages** : HTML, CSS, JavaScript, PHP, MySQL
- Application web simple, intuitive et responsive avec respect des rôles et des accès utilisateurs
- Base de données relationnelle
- Protection des données sensibles (patients, médecins). Authentification et hashage des mots de passe.

Deuxième partie Étude d'Opportunité

7 Contexte et Enjeux

L'hôpital fonctionne encore avec des documents papier ou des fichiers Excel pour gérer patients, médecins et rendez-vous.

Cette méthode entraîne des lenteurs administratives, des erreurs et une faible traçabilité des informations.

L'opportunité réside dans la **modernisation du système d'information**, afin d'améliorer la qualité du service et de réduire la charge de travail du personnel.

8 Intérêt du Projet

La mise en place d'un système informatisé permettrait :

- **Centralisation** des informations (patients, médecins, rendez-vous)
- **Meilleure traçabilité** et mise à jour en temps réel
- **Réduction des erreurs** humaines
- **Meilleure coordination** entre les acteurs (administration, secrétariat, médecins)

9 Analyse des Options

Option	Avantages	Inconvénients
Maintenir la gestion actuelle	Aucun coût	Inefficace, erreurs fréquentes
Acheter un logiciel commercial	Prêt à l'emploi	Coût élevé, adaptation limitée
Développer une application web interne	Adaptée aux besoins, faible coût	Temps de développement, maintenance nécessaire

TABLE 1 – Comparaison des options

9.1 Option Retenue

Développement d'une application web interne en PHP/MySQL, adaptée aux besoins réels de l'hôpital.

10 Conclusion

Le projet est opportun, car il répond à un besoin réel, améliore l'efficacité et la fiabilité, et reste réalisable avec les ressources disponibles.

Troisième partie Étude de Faisabilité

11 Faisabilité Technique

- **Technologies maîtrisées** : PHP, MySQL, HTML, CSS, JavaScript
- **Infrastructure disponible** : postes de travail et serveur local ou cloud
- Conception de la base de données relationnelle pour gérer patients, médecins et rendez-vous
- Interface responsive et accessible sur différents appareils

12 Faisabilité Économique

- Budget limité grâce aux technologies open source
- Pas de logiciel tiers à acheter

13 Faisabilité Organisationnelle

- Personnel hospitalier formé rapidement grâce à une interface intuitive

- Organisation interne respectée : administrateur, secrétaires, médecins
- Formation minimale nécessaire

14 Conclusion

Le projet est **réalisable** sur le plan technique, économique et organisationnel.

Le rapport de faisabilité confirme que la mise en œuvre est possible avec les ressources disponibles.

Quatrième partie Plan de Projet

15 Objectif

Le plan de projet permet d'**organiser et piloter** la réalisation de l'application web hospitalière.

Il définit les étapes, ressources, coûts, délais et risques.

16 Étapes et Jalons

Phase	Description	Durée	Livrable
Phase 1 : Analyse des besoins	Étude d'opportunité, faisabilité, cahier des charges, modèle métier	1 sem.	Cahier des charges validé, modèle métier
Phase 2 : Conception	Conception base de données (MCD/MLD), architecture système, maquettes interface	1 sem.	MCD, MLD, maquettes UI
Phase 3 : Développement	Codage des modules Patients, Médecins, Rendez-vous, Consultations	3 sem.	Application fonctionnelle
Phase 4 : Tests et validation	Tests unitaires et fonctionnels, corrections	1 sem.	Rapport de tests et version corrigée
Phase 5 : Déploiement	Mise en production, installation, formation	1 jour	Application déployée

TABLE 2 – Planning du projet

17 Ressources

17.1 Ressources Humaines

- 1 chef de projet

- 1 développeur
- 1 designer
- 1 utilisateur pilote

17.2 Ressources Matérielles

- Postes de travail
- Serveur local ou cloud
- Logiciels de développement
- Outils de suivi

18 Coût Estimé

- **Hébergement** : faible coût (open source ou interne)
- **Formation** : minimale, 1-2 sessions courtes
- **Maintenance** : planifiée sur le long terme

19 Risques et Mesures de Prévention

Risque	Solution
Retard de développement	Priorisation des fonctionnalités critiques
Difficulté d'adoption	Interface intuitive + formation
Bugs / anomalies	Tests fonctionnels et suivi des corrections
Problèmes techniques	Sauvegardes régulières, environnement de test

TABLE 3 – Gestion des risques

20 Suivi du Projet

- Réunions hebdomadaires
- Tableau de suivi des tâches (Trello / Gantt)
- Validation de chaque phase avant la suivante

21 Conclusion

Le plan de projet fournit une vision claire des étapes, ressources, délais, coûts et risques, servant de référence pour l'équipe et le client.

Cinquième partie

Spécifications Fonctionnelles

22 Spécifications Fonctionnelles Générales (SFG)

22.1 Gestion des Patients

- Ajouter, modifier, consulter et supprimer des patients.
- Stocker informations personnelles et médicales (nom, prénom, date de naissance, sexe, numéro de sécurité sociale, contact, adresse, groupe sanguin, allergies).

22.2 Gestion des Médecins

- Ajouter, modifier et consulter les informations des médecins.
- Gérer spécialité, téléphone, email et disponibilités.

22.3 Gestion des Rendez-vous

- Planifier, modifier et annuler les rendez-vous.
- Associer chaque rendez-vous à un patient et un médecin.
- Suivre le statut : planifié, confirmé, en cours, réalisé, annulé, absent.

22.4 Gestion des Consultations

- Saisir symptômes, diagnostics, prescriptions et observations.
- Relier chaque consultation à un rendez-vous existant.
- Enregistrer mesures médicales (poids, taille, tension, température).

22.5 Authentification et Rôles

- Accès sécurisé pour administrateur, secrétaire et médecin.
- Gestion des droits selon le rôle (CRUD selon permissions).

22.6 Recherche et Filtrage

- Rechercher patients, médecins ou rendez-vous par critères (nom, spécialité, date, statut).

22.7 Interface Utilisateur

- Interface web simple et intuitive, navigation claire entre modules.
- Compatible PC, tablette et smartphone (responsive design).

23 Spécifications Fonctionnelles Détaillées (SFD)

23.1 Ajouter un Patient

23.1.1 Règles de gestion

- Le numéro de sécurité sociale doit être unique.
- Les champs obligatoires : nom, prénom, date de naissance.

23.1.2 Scénario principal

1. L'utilisateur ouvre le formulaire « Ajouter un patient ».
2. Remplit les champs obligatoires et facultatifs.
3. Clique sur « Enregistrer ».
4. Le système vérifie les règles et enregistre le patient.

23.1.3 Scénario alternatif

- Si le numéro de sécurité sociale existe déjà, le système affiche un message d'erreur.

23.2 Ajouter un Médecin

23.2.1 Règles de gestion

- L'email doit être unique.
- Chaque médecin doit avoir une spécialité.

23.2.2 Scénario principal

1. L'utilisateur saisit les informations du médecin.
2. Le système vérifie les contraintes.
3. Le médecin est enregistré dans la base.

23.3 Planifier un Rendez-vous

23.3.1 Règles de gestion

- Le rendez-vous ne peut pas dépasser les disponibilités du médecin.
- Le patient et le médecin doivent exister dans la base.

23.3.2 Scénario principal

1. L'utilisateur sélectionne un patient et un médecin.
2. Choisit date et heure selon les disponibilités.
3. Le système enregistre le rendez-vous et affiche confirmation.

23.3.3 Scénario alternatif

- Si le créneau est occupé, le système propose des horaires disponibles.

23.4 Ajouter une Consultation

23.4.1 Règles de gestion

- Une consultation doit être liée à un rendez-vous existant.

23.4.2 Scénario principal

1. L'utilisateur sélectionne un rendez-vous.
2. Saisit symptômes, diagnostic, prescription et observations.
3. Enregistre la consultation.

23.5 Authentification et Rôles

23.5.1 Règles de gestion

- Les rôles définissent les droits d'accès aux modules.
- Mot de passe crypté dans la base.

23.5.2 Scénario principal

1. L'utilisateur saisit son identifiant et mot de passe.
2. Le système vérifie les informations et affiche l'interface adaptée au rôle.

Sixième partie

Dossier Technique

24 Architecture du Système

- **Architecture globale** : Application web avec séparation front-end / back-end.
- **Pattern utilisé** : MVC (Modèle-Vue-Contrôleur) pour organiser le code et faciliter la maintenance.

24.1 Modules Principaux

1. Gestion des patients
2. Gestion des médecins
3. Gestion des rendez-vous
4. Gestion des consultations
5. Authentification et gestion des rôles

25 Technologies Utilisées

Composant	Technologie	Raison
Front-end	HTML, CSS, JavaScript	Création d'interfaces interactives et responsive
Back-end	PHP	Langage serveur open source adapté à la gestion de formulaires et bases de données
Base de données	MySQL	Base relationnelle robuste, open source, facile à maintenir
Serveur web	Apache	Serveur stable et compatible PHP/-MySQL
Versioning	Git	Suivi des modifications et travail collaboratif

TABLE 4 – Stack technologique

26 Contraintes Techniques et Sécurité

26.1 Sécurité

- Authentification par mot de passe crypté (hash sécurisé).
- Gestion des rôles : accès limité selon profil (admin, secrétaire, médecin).

26.2 Performance

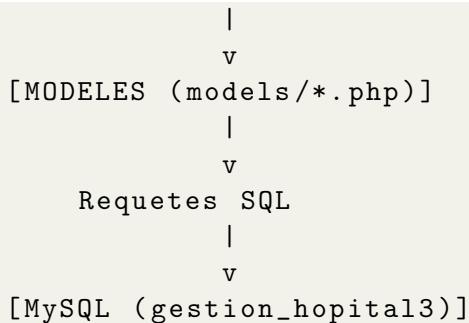
- Requêtes SQL optimisées avec index sur les colonnes utilisées pour recherche et filtrage.

26.3 Compatibilité

- Interface responsive compatible PC, tablette et smartphone.
- Tests sur navigateurs récents : Chrome, Safari.

27 Architecture Logique

```
[FRONTEND (HTML + CSS + JavaScript)]
  |
  v
Requetes HTTP (AJAX)
  |
  v
[API REST (api/*.php)]
  |
  v
Appels methodes
```



28 Arborescence du Projet

```

hospital-management-system/
|
+-- config.php                      # Configuration generale et connexion
|   BDD
+-- index.html                       # Page principale de l'application
|
+-- database/
|   +-- database.sql                 # Script de creation de la BDD
|
+-- models/
|   +-- Patient.php                # Modele Patient (CRUD)
|   +-- Medecin.php                # Modele Medecin
|   +-- RendezVous.php             # Modele Rendez-vous
|   +-- Consultation.php           # Modele Consultation
|   +-- Facturation.php            # Modele Facturation
|
+-- api/
|   +-- patients.php               # API REST Patients
|   +-- medecins.php               # API REST Medecins
|   +-- rendez-vous.php            # API REST Rendez-vous
|   +-- consultations.php          # API REST Consultations
|   +-- facturation.php            # API REST Facturation
|
+-- js/
|   +-- app.js                     # JavaScript principal
|   +-- patients.js                # Gestion des patients
|   +-- medecins.js                # Gestion des medecins
|   +-- rendez-vous.js              # Gestion des RDV
|   +-- facturation.js              # Gestion facturation
|
+-- css/
|   +-- style.css                  # Styles personnalises
|
+-- logs/
|   +-- error.log                  # Logs des erreurs
|
+-- README.md                        # Documentation

```

29 Diagrammes UML

Cette section présente les différents diagrammes de modélisation du système.

29.1 Diagramme MCD (Modèle Conceptuel de Données)

Le MCD représente la structure logique de la base de données avec les entités et leurs relations.

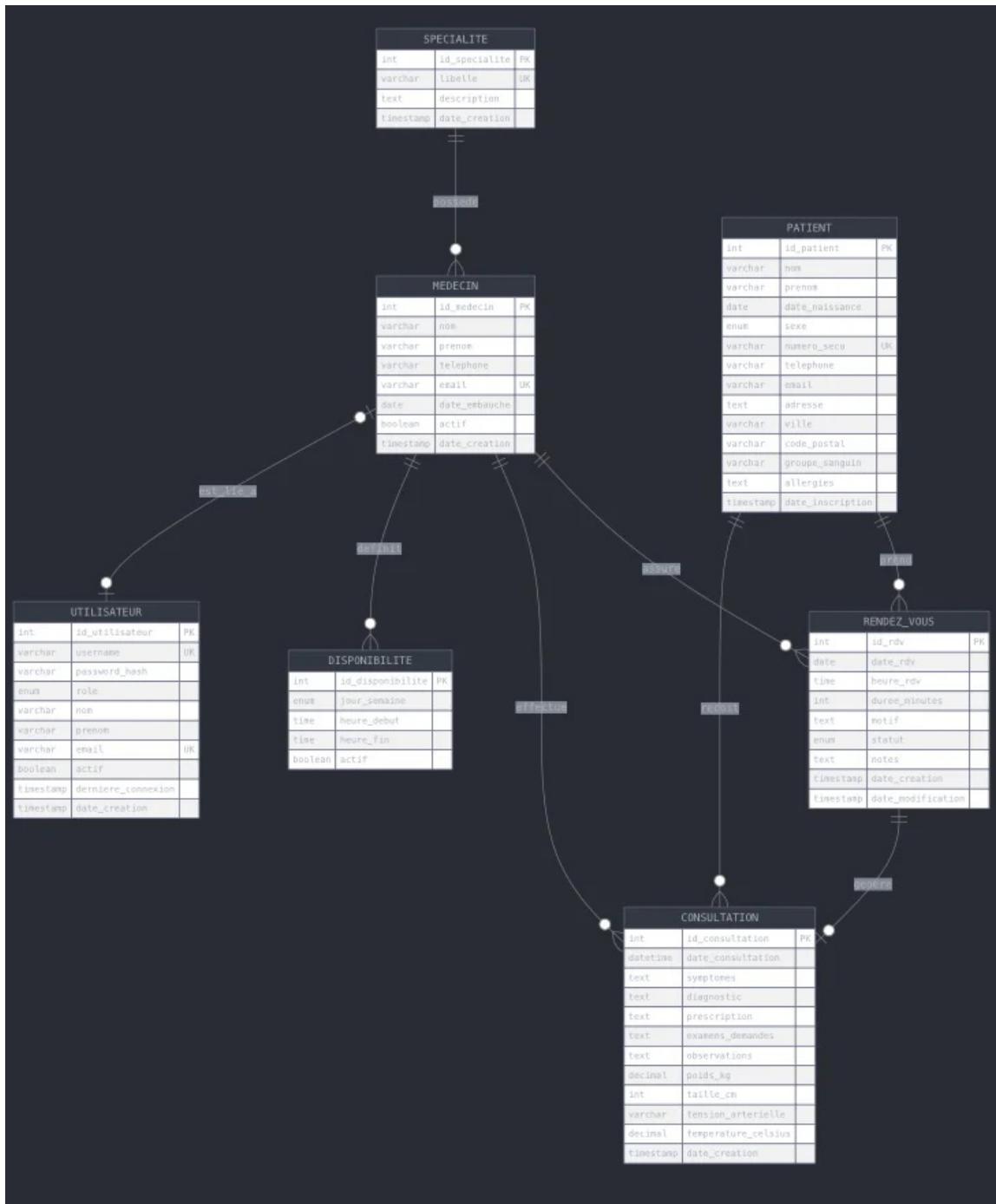


FIGURE 1 – Modèle Conceptuel de Données (MCD)

Le MCD montre les entités principales du système :

- **SPECIALITE** : Contient les spécialités médicales
- **MEDECIN** : Informations sur les médecins avec leur spécialité
- **PATIENT** : Données personnelles et médicales des patients
- **RENDEZ_VOUS** : Planification des rendez-vous entre patients et médecins
- **CONSULTATION** : Détails des consultations médicales
- **DISPONIBILITE** : Horaires de disponibilité des médecins
- **UTILISATEUR** : Comptes utilisateurs pour l'authentification

29.2 Diagramme Use Case (Cas d'Utilisation)

Le diagramme des cas d'utilisation présente les interactions entre les acteurs et le système.

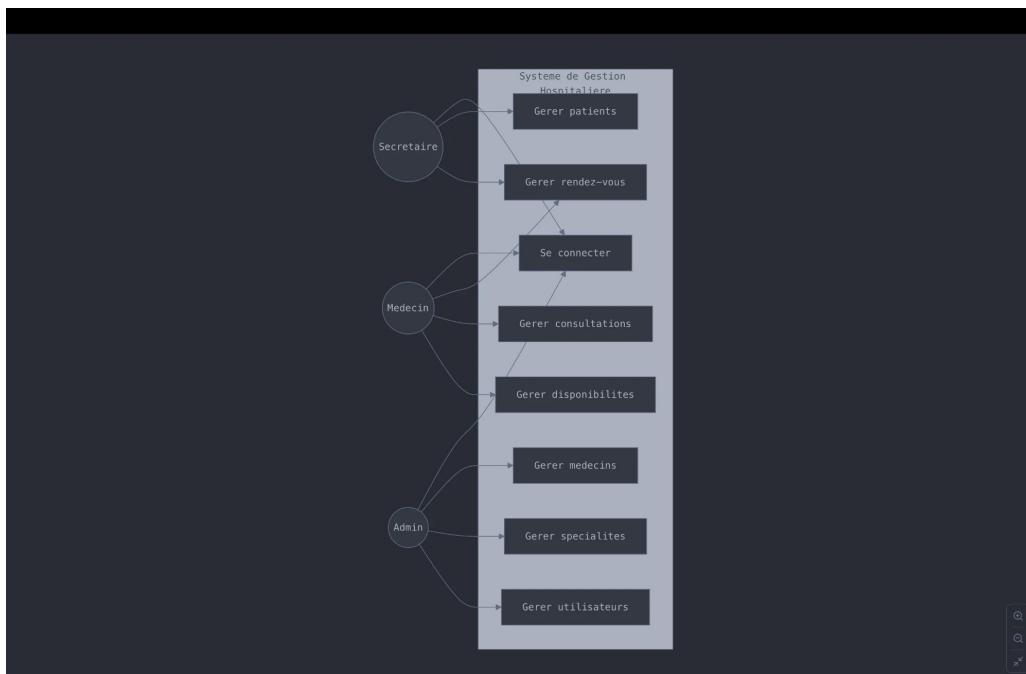


FIGURE 2 – Diagramme des Cas d'Utilisation

Les acteurs identifiés sont :

- **Secrétaire** : Gère les patients et les rendez-vous
- **Médecin** : Gère les consultations et ses disponibilités
- **Admin** : Gère les médecins, spécialités et utilisateurs

Les cas d'utilisation principaux incluent :

- Se connecter (tous les acteurs)
- Gérer patients
- Gérer rendez-vous
- Gérer consultations
- Gérer disponibilités
- Gérer médecins
- Gérer spécialités
- Gérer utilisateurs

29.3 Diagramme des Classes

Le diagramme des classes représente la structure objet du système avec les attributs et méthodes.

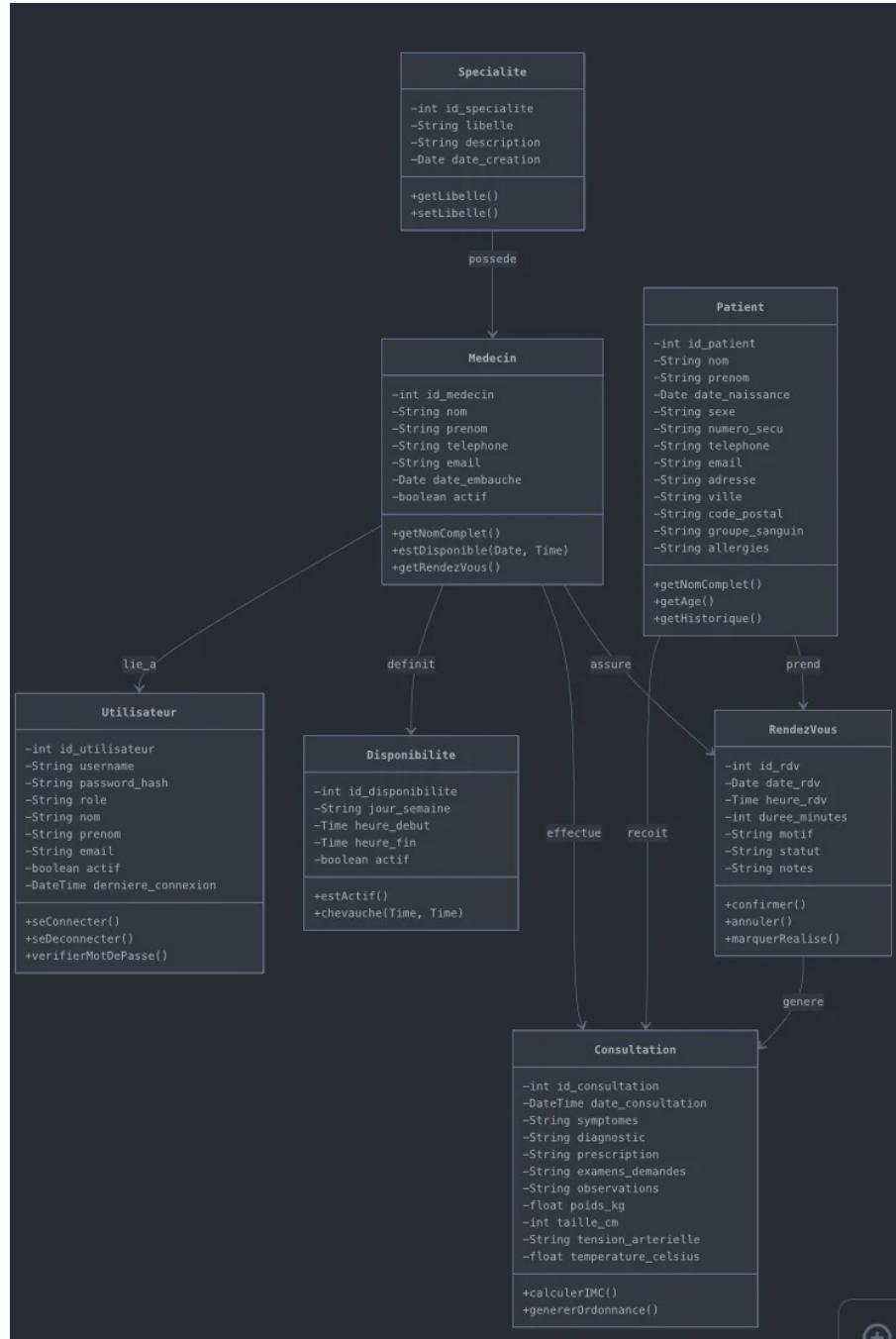


FIGURE 3 – Diagramme des Classes

Les classes principales et leurs responsabilités :

- **Specialite** : Gestion des spécialités médicales
 - Attributs : id_specialite, libelle, description, date_creation
 - Méthodes : getLibelle(), setLibelle()
- **Medecin** : Gestion des informations médecins

- Attributs : id_medecin, nom, prenom, telephone, email, date_embauche, actif
- Méthodes : getNomComplet(), estDisponible(), getRendezVous()
- **Patient** : Gestion des dossiers patients
 - Attributs : id_patient, nom, prenom, date_naissance, sexe, numero_secu, etc.
 - Méthodes : getNomComplet(), getAge(), getHistorique()
- **RendezVous** : Gestion des rendez-vous
 - Attributs : id_rdv, date_rdv, heure_rdv, duree_minutes, motif, statut, notes
 - Méthodes : confirmer(), annuler(), marquerRealise()
- **Consultation** : Gestion des consultations médicales
 - Attributs : id_consultation, date_consultation, symptomes, diagnostic, prescription, etc.
 - Méthodes : calculerIMC(), genererOrdonnance()
- **Disponibilite** : Gestion des créneaux disponibles
 - Attributs : id_disponibilite, jour_semaine, heure_debut, heure_fin, actif
 - Méthodes : estActif(), chevauche()
- **Utilisateur** : Gestion de l'authentification
 - Attributs : id_utilisateur, username, password_hash, role, nom, prenom, email, actif
 - Méthodes : seConnecter(), seDeconnecter(), verifierMotDePasse()

30 Maquettes et Prototypes

Cette section présente les maquettes de l'interface utilisateur de l'application.

30.1 Page de Connexion

La page de connexion permet aux utilisateurs de s'authentifier selon leur rôle.



FIGURE 4 – Maquette - Page de Connexion

Caractéristiques de l'interface de connexion :

- Champ identifiant (nom d'utilisateur)
- Champ mot de passe (masqué)
- Sélection du rôle (Admin, Secrétaire, Médecin)
- Bouton de connexion
- Design épuré et professionnel

30.2 Tableau de Bord

Le tableau de bord offre une vue d'ensemble de l'activité hospitalière.



FIGURE 5 – Maquette - Tableau de Bord

Éléments du tableau de bord :

- **Statistiques** : Nombre de patients, médecins, RDV du jour, consultations
- **Prochains RDV** : Liste des rendez-vous à venir avec statut
- **Menu latéral** : Navigation vers les différents modules
- **En-tête** : Informations utilisateur et date

30.3 Gestion des Patients

L'interface de gestion des patients permet le CRUD complet.



FIGURE 6 – Maquette - Gestion des Patients

Fonctionnalités de l'interface patients :

- **Barre de recherche** : Filtrage par nom, prénom
- **Bouton Nouveau** : Ajout d'un nouveau patient
- **Tableau des patients** : Affichage des informations principales
- **Actions** : Voir, Modifier, Supprimer pour chaque patient

30.4 Gestion des Rendez-vous

L'interface de gestion des rendez-vous permet la planification et le suivi.

La maquette montre une interface web avec un menu latéral sur la gauche et une liste de rendez-vous sur la droite.

Menu latéral (Hôpital Manager):

- Contenu Claude
- Tableau de bord
- Patients
- Medecins
- Rendez-vous (sélectionné)
- Consultations
- Facturation

Gestion des Rendez-vous :

- Date : 15/01/2025
- Filtre par médecin : Tous les medecins
- Bouton : + Nouveau RDV

HEURE	PATIENT	MEDECIN	MOTIF	STATUT
09:00	Dupont Marie	Dr. Martin	Consultation	Confirmé
10:30	Bernard Jean	Dr. Dubois	Suivi cardio	Planifié
14:00	Petit Sophie	Dr. Martin	Ordonnance	Confirmé
15:30	Moreau Luc	Dr. Leroy	Pediatrie	Annulé

FIGURE 7 – Maquette - Gestion des Rendez-vous

Fonctionnalités de l'interface rendez-vous :

- **Filtre par date** : Sélection de la date
- **Filtre par médecin** : Liste déroulante des médecins
- **Bouton Nouveau RDV** : Création d'un nouveau rendez-vous
- **Tableau des RDV** : Heure, patient, médecin, motif, statut
- **Statuts colorés** : Confirmé (vert), Planifié (jaune), Annulé (rouge)

30.5 Nouvelle Consultation

L'interface de consultation permet la saisie des informations médicales.



FIGURE 8 – Maquette - Nouvelle Consultation

Éléments de l'interface consultation :

- **Informations patient** : Nom, âge, sexe, groupe sanguin
- **Symptômes** : Zone de texte pour la description
- **Diagnostic** : Zone de texte pour le diagnostic médical
- **Prescription** : Zone de texte pour les médicaments prescrits
- **Constantes** : Poids, taille, tension, température
- **Actions** : Enregistrer, Générer ordonnance

Septième partie

Structure de la Base de Données

31 Création de la Base de Données

```

1 -- Creation de la base de donnees
2 CREATE DATABASE IF NOT EXISTS gestion_hopital3
3 CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
4
5 USE gestion_hopital3;
```

32 Tables Principales

32.1 Table : specialites

```

1 CREATE TABLE specialites (
2     id_specialite INT AUTO_INCREMENT PRIMARY KEY ,
3     libelle VARCHAR(100) NOT NULL UNIQUE ,
4     description TEXT ,
5     date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP
6 ) ENGINE=InnoDB ;

```

32.2 Table : medecins

```

1 CREATE TABLE medecins (
2     id_medecin INT AUTO_INCREMENT PRIMARY KEY ,
3     nom VARCHAR(100) NOT NULL ,
4     prenom VARCHAR(100) NOT NULL ,
5     id_specialite INT NOT NULL ,
6     telephone VARCHAR(20) ,
7     email VARCHAR(150) UNIQUE ,
8     date_embauche DATE ,
9     actif BOOLEAN DEFAULT TRUE ,
10    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
11    FOREIGN KEY (id_specialite)
12        REFERENCES specialites(id_specialite) ON DELETE RESTRICT ,
13    INDEX idx_medecin_nom (nom, prenom) ,
14    INDEX idx_medecin_specialite (id_specialite)
15 ) ENGINE=InnoDB ;

```

32.3 Table : patients

```

1 CREATE TABLE patients (
2     id_patient INT AUTO_INCREMENT PRIMARY KEY ,
3     nom VARCHAR(100) NOT NULL ,
4     prenom VARCHAR(100) NOT NULL ,
5     date_naissance DATE NOT NULL ,
6     sexe ENUM('M', 'F', 'Autre') ,
7     numero_secu VARCHAR(15) UNIQUE ,
8     telephone VARCHAR(20) ,
9     email VARCHAR(150) ,
10    adresse TEXT ,
11    ville VARCHAR(100) ,
12    code_postal VARCHAR(10) ,
13    groupe_sanguin VARCHAR(5) ,
14    allergies TEXT ,
15    date_inscription TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
16    INDEX idx_patient_nom (nom, prenom) ,
17    INDEX idx_patient_secu (numero_secu)
18 ) ENGINE=InnoDB ;

```

32.4 Table : rendez_vous

```

1 CREATE TABLE rendez_vous (
2     id_rdv INT AUTO_INCREMENT PRIMARY KEY ,
3     id_patient INT NOT NULL ,
4     id_medecin INT NOT NULL ,
5     date_rdv DATE NOT NULL ,

```

```

6    heure_rdv TIME NOT NULL ,
7    duree_minutes INT DEFAULT 30 ,
8    motif TEXT NOT NULL ,
9    statut ENUM('planifie', 'confirme', 'en_cours',
10              'realise', 'annule', 'absent') DEFAULT 'planifie',
11    notes TEXT ,
12    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
13    date_modification TIMESTAMP DEFAULT CURRENT_TIMESTAMP
14          ON UPDATE CURRENT_TIMESTAMP ,
15    FOREIGN KEY (id_patient)
16        REFERENCES patients(id_patient) ON DELETE CASCADE ,
17    FOREIGN KEY (id_medecin)
18        REFERENCES medecins(id_medecin) ON DELETE RESTRICT ,
19    INDEX idx_rdv_date (date_rdv, heure_rdv),
20    INDEX idx_rdv_patient (id_patient),
21    INDEX idx_rdv_medecin (id_medecin),
22    INDEX idx_rdv_statut (statut)
23 ) ENGINE=InnoDB ;

```

32.5 Table : consultations

```

1 CREATE TABLE consultations (
2     id_consultation INT AUTO_INCREMENT PRIMARY KEY ,
3     id_rdv INT NOT NULL UNIQUE ,
4     id_patient INT NOT NULL ,
5     id_medecin INT NOT NULL ,
6     date_consultation DATETIME NOT NULL ,
7     symptomes TEXT ,
8     diagnostic TEXT ,
9     prescription TEXT ,
10    examens_demandes TEXT ,
11    observations TEXT ,
12    poids_kg DECIMAL(5,2) ,
13    taille_cm INT ,
14    tension_arterielle VARCHAR(20) ,
15    temperature_celsius DECIMAL(4,2) ,
16    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
17    FOREIGN KEY (id_rdv)
18        REFERENCES rendez_vous(id_rdv) ON DELETE CASCADE ,
19    FOREIGN KEY (id_patient)
20        REFERENCES patients(id_patient) ON DELETE CASCADE ,
21    FOREIGN KEY (id_medecin)
22        REFERENCES medecins(id_medecin) ON DELETE RESTRICT ,
23    INDEX idx_consultation_patient (id_patient),
24    INDEX idx_consultation_date (date_consultation)
25 ) ENGINE=InnoDB ;

```

32.6 Table : disponibilites

```

1 CREATE TABLE disponibilites (
2     id_disponibilite INT AUTO_INCREMENT PRIMARY KEY ,
3     id_medecin INT NOT NULL ,
4     jour_semaine ENUM('Lundi', 'Mardi', 'Mercredi',
5                      'Jeudi', 'Vendredi', 'Samedi', 'Dimanche') NOT
NULL ,

```

```

6    heure_debut TIME NOT NULL,
7    heure_fin TIME NOT NULL,
8    actif BOOLEAN DEFAULT TRUE,
9    FOREIGN KEY (id_medecin)
10       REFERENCES medecins(id_medecin) ON DELETE CASCADE,
11       INDEX idx_dispo_medecin (id_medecin),
12       UNIQUE KEY unique_disponibilite (id_medecin, jour_semaine,
13         heure_debut)
13 ) ENGINE=InnoDB;

```

32.7 Table : utilisateurs

```

1 CREATE TABLE utilisateurs (
2     id_utilisateur INT AUTO_INCREMENT PRIMARY KEY,
3     username VARCHAR(50) NOT NULL UNIQUE,
4     password_hash VARCHAR(255) NOT NULL,
5     role ENUM('admin', 'secretaire', 'medecin') NOT NULL,
6     id_medecin INT NULL,
7     nom VARCHAR(100),
8     prenom VARCHAR(100),
9     email VARCHAR(150) UNIQUE,
10    actif BOOLEAN DEFAULT TRUE,
11    derniere_connexion TIMESTAMP NULL,
12    date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
13    FOREIGN KEY (id_medecin)
14       REFERENCES medecins(id_medecin) ON DELETE SET NULL,
15       INDEX idx_user_role (role)
16 ) ENGINE=InnoDB;

```

33 Tables de Facturation

33.1 Table : factures

```

1 CREATE TABLE IF NOT EXISTS factures (
2     id_facture INT AUTO_INCREMENT PRIMARY KEY,
3     id_patient INT NOT NULL,
4     id_utilisateur INT NULL,
5     date_emission DATETIME DEFAULT CURRENT_TIMESTAMP,
6     date_debut DATE,
7     date_fin DATE,
8     total_ht DECIMAL(10,2) DEFAULT 0.00,
9     tva_rate DECIMAL(5,2) DEFAULT 0.00,
10    total_ttc DECIMAL(10,2) DEFAULT 0.00,
11    statut ENUM('brouillon', 'emise', 'reglee', 'annulee') DEFAULT 'brouillon',
12    notes TEXT,
13    FOREIGN KEY (id_patient)
14       REFERENCES patients(id_patient) ON DELETE CASCADE,
15    FOREIGN KEY (id_utilisateur)
16       REFERENCES utilisateurs(id_utilisateur) ON DELETE SET NULL,
17       INDEX idx_facture_patient (id_patient),
18       INDEX idx_facture_statut (statut)
19 ) ENGINE=InnoDB;

```

33.2 Table : ligne_facture

```

1 CREATE TABLE IF NOT EXISTS ligne_facture (
2     id_ligne INT AUTO_INCREMENT PRIMARY KEY ,
3     id_facture INT NOT NULL ,
4     description VARCHAR(255) NOT NULL ,
5     quantite INT DEFAULT 1 ,
6     prix_unitaire DECIMAL(10,2) NOT NULL ,
7     total_ligne DECIMAL(10,2) AS (quantite * prix_unitaire) STORED ,
8     type_ligne ENUM('consultation','examen','medicament',
9                     'hospitalisation','autre') DEFAULT 'autre',
10    FOREIGN KEY (id_facture)
11        REFERENCES factures(id_facture) ON DELETE CASCADE ,
12    INDEX idx_ligne_facture (id_facture)
13 ) ENGINE=InnoDB ;

```

33.3 Table : paiements

```

1 CREATE TABLE IF NOT EXISTS paiements (
2     id_paiement INT AUTO_INCREMENT PRIMARY KEY ,
3     id_facture INT NOT NULL ,
4     date_paiement DATETIME DEFAULT CURRENT_TIMESTAMP ,
5     montant DECIMAL(10,2) NOT NULL ,
6     methode ENUM('espece','carte','cheque','virement','assurance')
7         DEFAULT 'espece' ,
8     reference VARCHAR(255) ,
9     notes TEXT ,
10    FOREIGN KEY (id_facture)
11        REFERENCES factures(id_facture) ON DELETE CASCADE ,
12    INDEX idx_paiement_facture (id_facture)
13 ) ENGINE=InnoDB ;

```

34 Tables de Services et Hospitalisations

34.1 Table : services

```

1 CREATE TABLE IF NOT EXISTS services (
2     id_service INT AUTO_INCREMENT PRIMARY KEY ,
3     libelle VARCHAR(100) NOT NULL UNIQUE ,
4     description TEXT ,
5     responsable VARCHAR(100) ,
6     telephone VARCHAR(20) ,
7     email VARCHAR(150) ,
8     date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
9     INDEX idx_service_libelle (libelle)
10 ) ENGINE=InnoDB ;

```

34.2 Table : lits

```

1 CREATE TABLE IF NOT EXISTS lits (
2     id_lit INT AUTO_INCREMENT PRIMARY KEY ,
3     id_service INT NOT NULL ,

```

```

4     numero_lit VARCHAR(20) NOT NULL,
5     etat ENUM('libre', 'occupe', 'maintenance') DEFAULT 'libre',
6     date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
7     FOREIGN KEY (id_service)
8         REFERENCES services(id_service) ON DELETE CASCADE,
9     UNIQUE KEY unique_lit (id_service, numero_lit),
10    INDEX idx_lit_service (id_service),
11    INDEX idx_lit_etat (etat)
12 ) ENGINE=InnoDB;

```

34.3 Table : hospitalisations

```

1 CREATE TABLE IF NOT EXISTS hospitalisations (
2     id_hospitalisation INT AUTO_INCREMENT PRIMARY KEY,
3     id_patient INT NOT NULL,
4     id_lit INT NOT NULL,
5     date_entree DATETIME NOT NULL,
6     date_sortie DATETIME NULL,
7     motif TEXT NOT NULL,
8     statut ENUM('en_cours', 'terminee', 'annulee') DEFAULT 'en_cours',
9     date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
10    FOREIGN KEY (id_patient)
11        REFERENCES patients(id_patient) ON DELETE CASCADE,
12    FOREIGN KEY (id_lit)
13        REFERENCES lits(id_lit) ON DELETE RESTRICT,
14    INDEX idx_hospitalisation_patient (id_patient),
15    INDEX idx_hospitalisation_lit (id_lit),
16    INDEX idx_hospitalisation_statut (statut)
17 ) ENGINE=InnoDB;

```

35 Tables Tarifs et Assurances

35.1 Table : tarifs

```

1 CREATE TABLE IF NOT EXISTS tarifs (
2     id_tarif INT AUTO_INCREMENT PRIMARY KEY,
3     libelle VARCHAR(100) NOT NULL,
4     type ENUM('consultation', 'examen', 'hospitalisation',
5               'medicament', 'autre') NOT NULL,
6     prix_unitaire DECIMAL(10,2) NOT NULL,
7     description TEXT,
8     date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
9     INDEX idx_tarif_type (type)
10 ) ENGINE=InnoDB;

```

35.2 Table : assurances

```

1 CREATE TABLE IF NOT EXISTS assurances (
2     id_assurance INT AUTO_INCREMENT PRIMARY KEY,
3     nom VARCHAR(100) NOT NULL,
4     contact VARCHAR(100),
5     telephone VARCHAR(20),

```

```

6   email VARCHAR(150),
7   adresse TEXT,
8   taux_remboursement DECIMAL(5,2) DEFAULT 0.00,
9   date_creation TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
10  INDEX idx_assurance_nom (nom)
11 ) ENGINE=InnoDB;

```

Huitième partie

Guide d'Installation

36 Prérequis

Avant de commencer, les éléments suivants doivent être installés :

- **Serveur web** : Apache ou Nginx
- **PHP** : version 8.x ou supérieure
- **Extensions PHP nécessaires** : pdo_mysql, mysqli, session
- **Serveur de base de données** : MySQL / MariaDB
- **Navigateur web** : Chrome, Firefox, etc.
- Dossier du projet récupéré (ZIP ou dépôt Git)

37 Installation du Projet

37.1 Télécharger le Projet

Récupérer l'archive ZIP ou cloner le dépôt Git :

```
git clone https://lien-du-projet.git
```

Extraire le dossier dans le répertoire web du serveur :

- Sous XAMPP : htdocs/
- Sous WAMP : www/
- Sous Linux/Apache : /var/www/html/

37.2 Configuration de la Base de Données

37.2.1 Créer la Base de Données

1. Ouvrir phpMyAdmin ou un client MySQL.
2. Créer une base nommée :

```
1 CREATE DATABASE gestion_hopital;
```

37.2.2 Importer la Structure et les Données

1. Dans phpMyAdmin → Importer.
2. Sélectionner le fichier SQL fourni (`database.sql`).
3. Valider l'importation.

37.3 Configuration du Fichier database.php

Dans le projet, ouvrir le fichier : `/config/database.php`
 Modifier avec les bonnes informations :

```
$host = "localhost";
$dbname = "gestion_hopital";
$username = "root";           // ou autre utilisateur MySQL
$password = "";               // ou mot de passe si nécessaire
```

38 Lancement de l'Application

38.1 Démarrer le Serveur

Lancer Apache et MySQL via XAMPP/WAMP ou démarrer les services sous Linux :

```
sudo service apache2 start
sudo service mysql start
```

38.2 Ouvrir l'Application

Accéder à : `http://localhost/nom_du_projet/`
 La page d'accueil ou la page de connexion doit s'afficher.

39 Problèmes Fréquents et Solutions

Problème	Cause possible	Solution
Page blanche	Erreur PHP non affichée	Activer <code>display_errors</code> dans <code>php.ini</code>
Connexion impossible	Mauvaises infos DB	Vérifier <code>database.php</code>
Tables manquantes	Import SQL incomplet	Réimporter le fichier SQL

TABLE 5 – Dépannage

Neuvième partie

Support et FAQ

40 Questions Fréquentes

40.1 Je n'arrive pas à me connecter

- Vérifier identifiant/mot de passe.
- Vérifier que Apache et MySQL sont démarrés.

40.2 La page est blanche ou un bouton ne marche pas

- Rafraîchir la page.
- Redémarrer les services (Apache/MySQL).

40.3 Impossible d'ajouter un patient ou un rendez-vous

- Vérifier que tous les champs sont remplis.
- Réessayer après avoir rechargé la page.

40.4 Les données ne s'enregistrent pas

- Vérifier la connexion à la base de données.
- Contacter l'administrateur en cas de persistance.

41 Plan de Tests

41.1 Objectif

Valider le bon fonctionnement de l'application avant mise en production.

41.2 Types de Tests

- **Tests unitaires** : Vérification des fonctions individuelles
- **Tests fonctionnels** : Validation des scénarios utilisateur
- **Tests d'intégration** : Vérification des interactions entre modules
- **Tests de performance** : Mesure des temps de réponse

41.3 Rapports de Tests

Présenter les résultats des tests et les anomalies corrigées.

Livrable : Rapport de validation

Explication : Fournit un bilan sur la qualité du système testé avant sa mise en production.

42 Site Web - Gestion Hospitalière

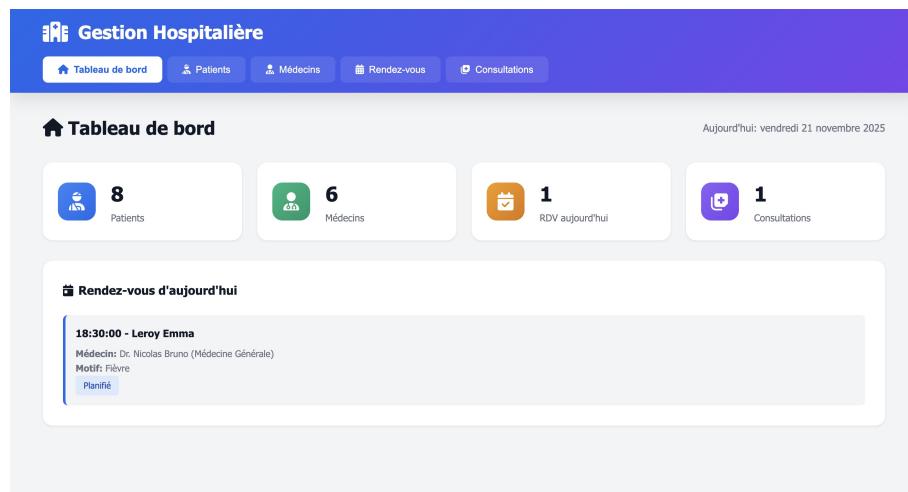


FIGURE 9 – Page d'accueil

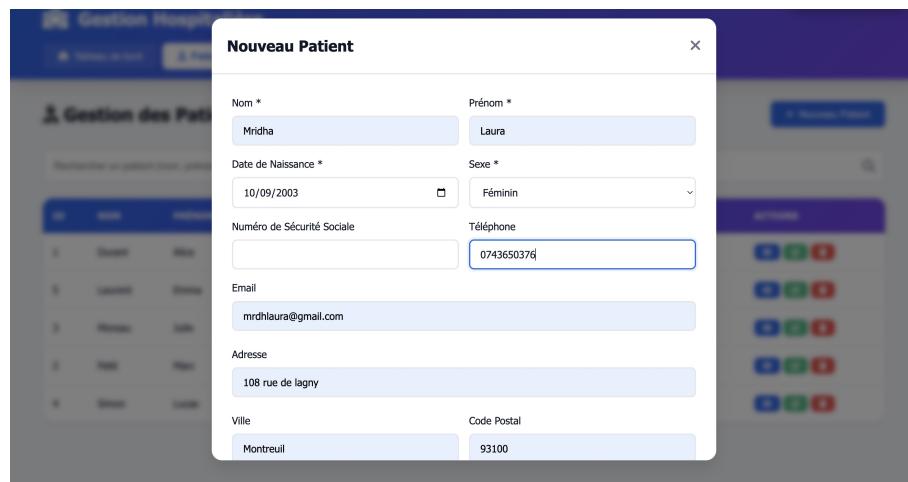


FIGURE 10 – Ajout d'un nouveau Patient

ID	NOM	PRÉNOM	DATE NAISSANCE	SEXE	TELÉPHONE	EMAIL	ACTIONS
1	Durant	Alice	15/05/1990	F	0612345678	alice.durant@email.fr	
5	Laurent	Emma	30/12/1978	F	0612345682	emma.laurent@email.fr	
3	Moreau	Julie	08/03/1995	F	0612345680	julie.moreau@email.fr	
6	Mridha	Laura	10/09/2003	F	0783650348	mrdhlaura@gmail.com	
2	Petit	Marc	22/11/1985	M	0612345679	marc.petit@email.fr	
4	Simon	Lucas	20/07/2015	M	0612345681	contact.simon@email.fr	

FIGURE 11 – Liste des patients

Nouveau Médecin

Nom * Prénom *

Spécialité *
Sélectionner une spécialité...

Téléphone Email

Date d'Embauche Statut *

22/11/2025 Actif

Annuler Enregistrer

FIGURE 12 – Ajout des médecins

Gestion des Médecins

ID	NOM	PRÉNOM	SPÉCIALITÉ	TÉLÉPHONE	EMAIL	STATUT	ACTIONS
3	Bernard	Pierre	Pédiatrie	0123456791	p.bernard@hôpital.fr	Actif	
4	Dubois	Marie	Médecine générale	0123456792	m.dubois@hôpital.fr	Actif	
1	Dupont	Jean	Cardiologie	0123456789	j.dupont@hôpital.fr	Actif	
6	Hugo	Durand	Cardiologie	0787658349	h.durand@gmail.com	Actif	
5	Leroy	Thomas	Orthopédie	0123456793	t.leroy@hôpital.fr	Actif	
2	Martin	Sophie	Dermatologie	0123456790	s.martin@hôpital.fr	Actif	

FIGURE 13 – Liste des médecins

Nouveau Rendez-vous

Patient *
Laurent Emma

Médecin *
Dr. Dupont Jean - Cardiologie

Date * Heure *

24/11/2025 14:30

Durée (minutes) Statut *

30 Planifié

Motif *

FIGURE 14 – Nouveau Rendez-vous

ID	DATE	HEURE	PATIENT	MÉDECIN	MOTIF	STATUT	ACTIONS
5	28/11/2025	15:30:00	Laurent Emma	Dr. Leroy Thomas	Douleur au genou	Planifié	
4	27/11/2025	11:00:00	Simon Lucas	Dr. Bernard Pierre	Vaccination	Planifié	
3	26/11/2025	14:00:00	Moreau Julie	Dr. Dubois Marie	Consultation générale	Planifié	
2	25/11/2025	10:30:00	Petit Marc	Dr. Martin Sophie	Problème de peau	Confirmé	
1	25/11/2025	09:00:00	Durant Alice	Dr. Dupont Jean	Consultation cardiologie	Planifié	
7	24/11/2025	14:30:00	Laurent Emma	Dr. Dupont Jean	fievre	Planifié	

FIGURE 15 – Liste des Rendez-vous

Grille Tarifaire			
Consultations			
CODE	LIBELLÉ	PRIX	REM.
underrined	Consultation générale	50,00 €	70,00%
underrined	Consultation spécialisée	80,00 €	70,00%
Examens			
CODE	LIBELLÉ	PRIX	REM.
underrined	Analyse sanguine	60,00 €	70,00%
underrined	Radiographie	120,00 €	70,00%
Hospitalisation			
CODE	LIBELLÉ	PRIX	REM.

Assurances & Mutuelles Partenaires			
NOM	TYPE	TAUX COUVERTURE	CONTACT
Assurance Santé Plus	Mutuelle	70,00%	0123456780 contact@santeplus.fr
Mutuelle Générale	Mutuelle	80,00%	0123456781 contact@mutuelgen.fr

FIGURE 16 – Grille des tarifs

NOM	TYPE	TAUX COUVERTURE	CONTACT
Assurance Santé Plus	Mutuelle	70,00%	0123456780 contact@santeplus.fr
Mutuelle Générale	Mutuelle	80,00%	0123456781 contact@mutuelgen.fr

FIGURE 17 – Assurance et Mutuelle

The screenshot shows a modal window for creating a patient invoice. The top section includes fields for 'Patient * (6 disponibles)' (selected: Laurent Emma), 'Date de facture *' (22/11/2025), 'Assurance / Mutuelle (2 disponibles)' (selected: Assurance Santé Plus (70.00%)), and 'N° Prise en charge' (Optionnel). Below this is a table for 'Lignes de facturation (6 tarifs disponibles)' with one row added: 'Consultation spécialisée (80,00 €)' with a quantity of 1 and a unit price of 80,00. A button '+ Ajouter une ligne' is visible. At the bottom, summary fields show 'Sous-total HT' (80,00 €), 'TVA' (0,00 €), 'Total TTC' (80,00 €), 'Prise en charge assurance' (56,00 € (70%)), and 'Reste à charge patient' (24,00 €).

FIGURE 18 – Facture d'un patient

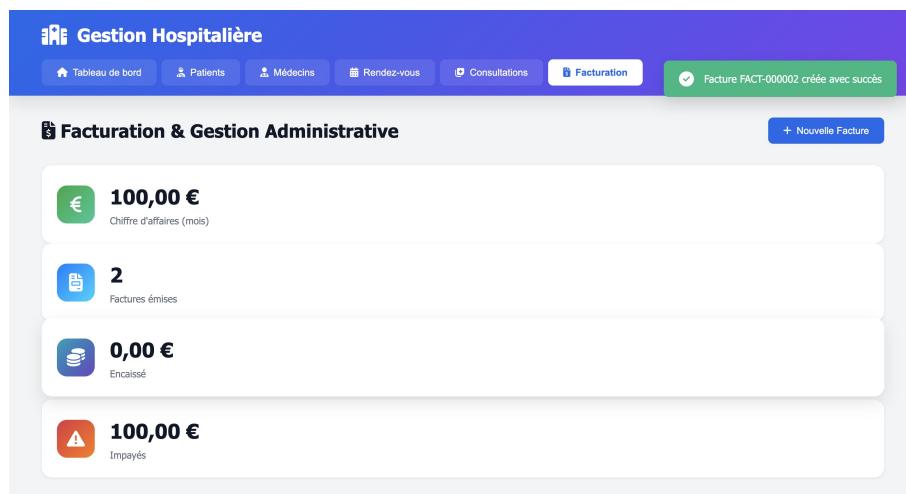


FIGURE 19 – Facture