

# **Application web de gestion d'un Cabinet Médical**

Présenté par :

Fatma Beji & Yasmine Abdelhak

Groupe: FIA01-02

Sous Groupe: 1

Année Universitaire :2020/2021

## Contents

Introduction générale .....	3
Chapitre1 : Etude préalable.....	4
1. Contexte du projet .....	4
2. Problématique .....	4
3. Solution proposée .....	4
Chapitre 2 : Spécification des besoins .....	6
1. Identification des acteurs .....	6
2. Spécification des besoins fonctionnels .....	6
3. Spécification des besoins non fonctionnels .....	7
Chapitre 3 : Etude conceptuelle .....	8
1. Architecture logicielle.....	8
2. Diagramme de cas d'utilisation.....	9
3. Diagramme de classe.....	10
Chapitre 4 : Réalisation et implémentation .....	11
1. Environnement de développement .....	11
1.1. Adobe XD.....	11
1.2. StarUML .....	11
1.3. NetBeans .....	12
1.4. phpMyAdmin .....	12
2. Technologies utilisées .....	12
2.1. Java .....	12
2.2. MySQL .....	13
3. Interface de l'application .....	13
3.1. Interface d'authentification.....	13
3.2. Interfaces de la fiche patient.....	15
3.3. Interface de la fiche consultation .....	17
3.4. Interface de la fiche médicaments de la pharmacie .....	17
Conclusion générale .....	19

# Introduction générale

Étant donnée l'émergence de l'informatisation des systèmes de gestions des bases de données, beaucoup applications ont été développées dans divers domaines. Parmi ces domaines nous trouvons le domaine de la santé.

Ce domaine est très important et ses données relatives à lui sont super sensibles voilà pourquoi elles doivent être bien enregistrées et ordonnées dans une base de données pour pouvoir y accéder facilement en cas de besoin.

C'est dans ce cadre que s'impose l'idée de la planification des données d'un cabinet médical dans une base de données avec une petite application pour la gestion des informations.

# Chapitre1 : Etude préalable

## 1. Contexte du projet

Vu l'état sanitaire aujourd'hui, beaucoup de gens viennent consulter plus de médecins qu'avant. Les médecins ont vécu ces derniers temps une surcharge énorme.

## 2. Problématique

Etant donnée la surcharge des patients, les secrétaires et les médecins ne peuvent plus contrôler cet énorme nombre, les fiches patients sont de plus en plus dispersées ou mal organisées dans ce cas la secrétaire doit saisir de nouveau les informations relatives aux anciens patients sur un papier. Un autre problème rencontré est que les médecins retournent parfois pour vérifier quelques noms de médicaments ou leurs utilisations, ils doivent également à chaque fois prescrire une ordonnance pour chaque patient. Toutes ces informations concernant les patients, les médicaments et les ordonnances peuvent être informatisées dans une application.

## 3. Solution proposée

Notre solution consiste en une application web comportant quelques interfaces simples pour les médecins et les secrétaires selon le rôle de l'utilisateur, chaque information saisie est enregistrée dans la base de données de manière à retrouver toutes les données nécessaires en cas de besoin. Cette application

comporte les informations relatives aux patients, aux médicaments et aux consultations.

Notre application vise à :

- Ordonner les données médicales de chaque médecin de façon à mieux organiser le travail.
- Faciliter les tâches pour les médecins et les secrétaires.
- Gagner du temps et augmenter le nombre de consultations.

# Chapitre 2 : Spécification des besoins

## 1. Identification des acteurs

Notre application peut être utilisée par deux acteurs :

- Secrétaire : Qui se charge d'identifier les patients, elle peut gérer la liste des patients, les ajouter, les retirer ou modifier les informations d'un patient, elle peut aussi consulter la liste des consultations en apportant également quelques modifications.
- Médecin : Son rôle lui permet d'accéder à toutes les interfaces de l'application, il permet de consulter la liste des patients, des consultations et également l'interface consacrée à la création de l'ordonnance tout en consultant la liste à médicaments à prescrire.

## 2. Spécification des besoins fonctionnels

- ❖ Consulter la liste des patients
  - La secrétaire peut consulter cette liste, ajouter un patient, retirer un patient de la base de données ou modifier quelques informations relatives à un patient.
  - Le médecin peut consulter cette interface aussi juste pour s'informer de ces patients.
- ❖ Consulter la liste des consultations

- Le médecin peut consulter la liste des consultations pour gérer son temps et planifier son travail, il peut aussi prendre des notes pour chaque consultation.
- La secrétaire peut à son tour accéder à cette interface pour gérer les consultations en cas de besoin.
- ❖ Consulter la liste des médicaments
  - Seul le médecin son rôle lui permet d'accéder à cette interface pour s'informer sur les médicaments en cas de besoin.
- ❖ Prescrire l'ordonnance
  - Seul le médecin peut prescrire une ordonnance après chaque consultation.
- ❖ Authentification
  - Les deux utilisateurs doivent s'authentifier pour pouvoir accéder à l'application.

### 3. Spécification des besoins non fonctionnels

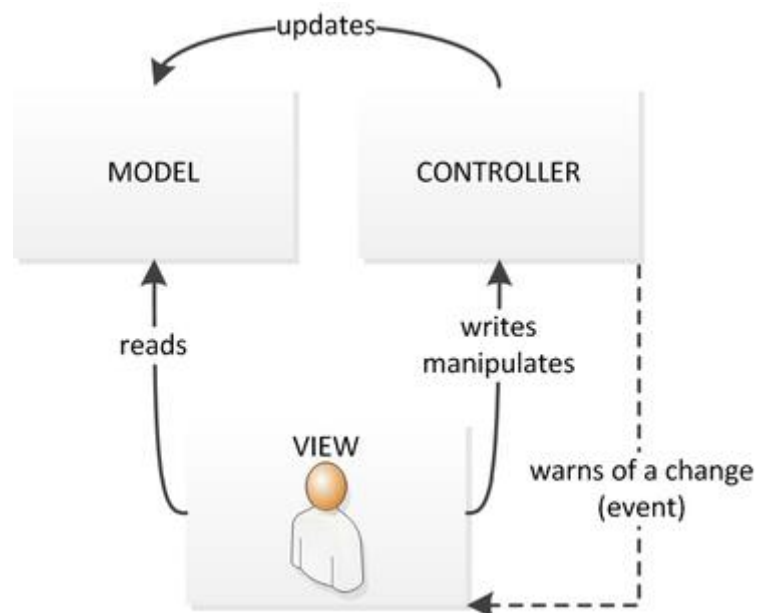
- ❖ Sécurité
  - Les données relatives à chaque patient sont bien sécurisées
  - Chacun des utilisateurs peut accéder aux interfaces dont il a le droit d'y accéder selon son rôle.
- ❖ Facilité d'emploi
  - Application facile à manipuler comportant des interfaces simples et claires.
- ❖ Ergonomie
  - Les interfaces de l'application sont simples et non encombrées, elles sont adaptables pour toute forme d'appareil.

# Chapitre 3 : Etude conceptuelle

## 1. Architecture logicielle

L'architecture de l'environnement de notre application est basée sur le modèle MVC qui permet de séparer les aspects traitements, données et présentations et de définir les interactions entre ces trois aspects. Les données sont gérées par le modèle, la présentation par la vue, les traitements par des actions et l'ensemble est coordonné par les contrôleurs.

Pour notre application, on a comme modèles ordonnance, patient et consultation qui contiennent les informations relatives à chaque entité, les contrôleurs de ces entités qui contiennent les opérations et les actions effectuées sur ces modèles et finalement les vues qui sont basées sur les interfaces de l'application, l'interaction de l'utilisateur avec la base de données.

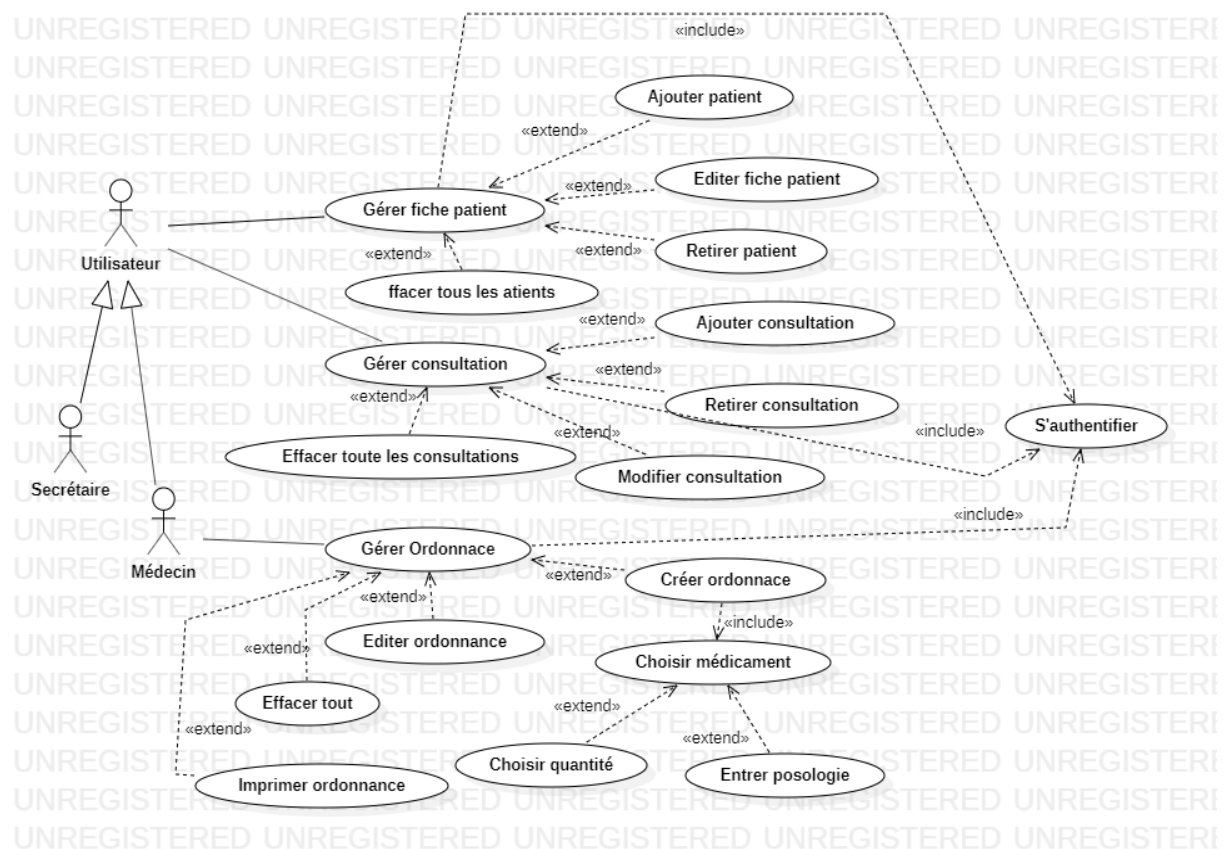




## 2. Diagramme de cas d'utilisation

Le diagramme de Cas d'Utilisation (Use Case) n'est qu'un des moyens d'offrir une représentation du comportement fonctionnel d'un système. Ce diagramme constitue le moyen essentiel pour citer les fonctionnalités d'un système du point de vue de l'utilisateur. Il permet d'assurer l'ensemble des relations entre l'utilisateur et les objets mis en œuvre par le système.

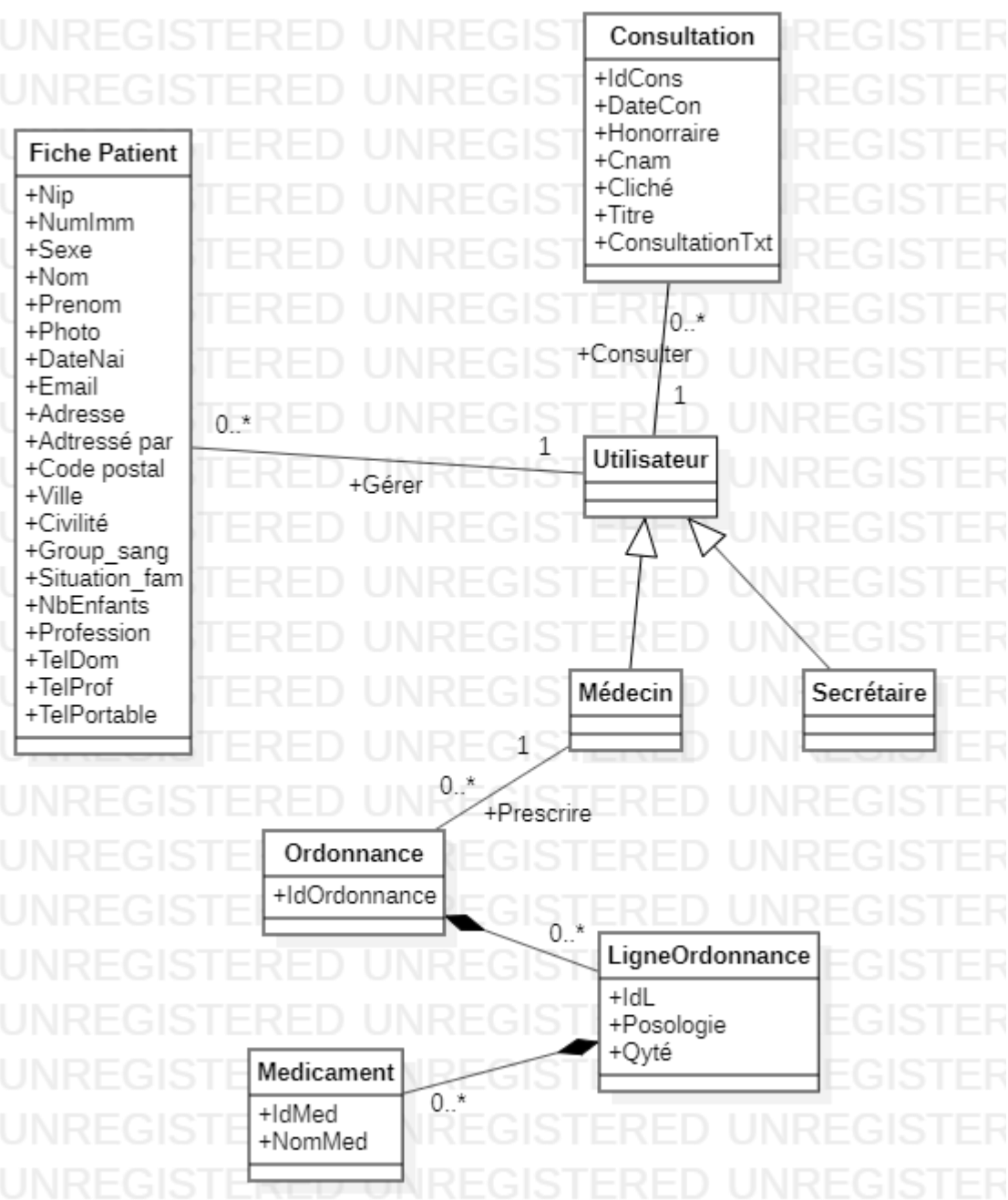
Dans notre cas, le diagramme de cas d'utilisation nous donne une vue globale sur les acteurs de notre application et leurs fonctionnalités.



### 3. Diagramme de classe

Le diagramme de classe représente les classes intervenant dans le système.

Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations.



# Chapitre 4 : Réalisation et implémentation

## 1. Environnement de développement

### 1.1. Adobe XD

**Adobe XD** est un outil de conception d'expérience utilisateur basé sur le vecteur pour les applications Web et les applications mobiles. Il permet de concevoir les maquettes d'une application avant que les interfaces soient implémentées.



### 1.2. StarUML

**StarUML** est un logiciel open source Il s'agit d'une plateforme de modélisation avec le langage UML. Cet outil propose tous les diagrammes nécessaires à une bonne modélisation.



### 1.3. NetBeans

**NetBeans** est un environnement de développement intégré (EDI), placé en *open source* . Il permet la prise en charge de divers langages tel que java. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web.



### 1.4. phpMyAdmin

**phpMyAdmin** est une application Web de gestion pour les systèmes de gestion de base de données MySQL.



## 2. Technologies utilisées

### 2.1. Java

**Java** est un langage de programmation orienté objet. Il contient des API pour les interfaces graphiques tel que awt, swing des API pour la gestion de la base de données tel que SQL.



## 2.2. MySQL

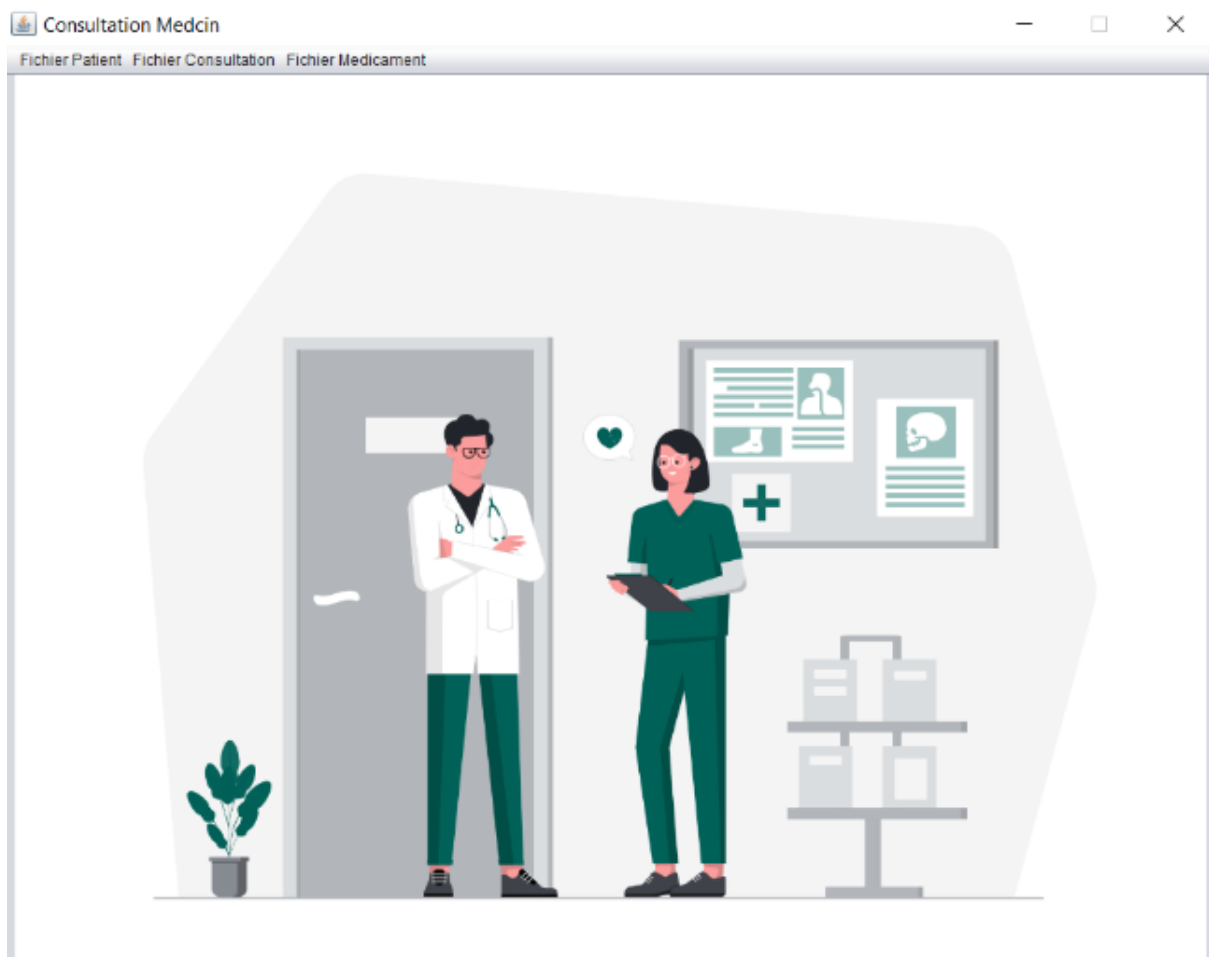
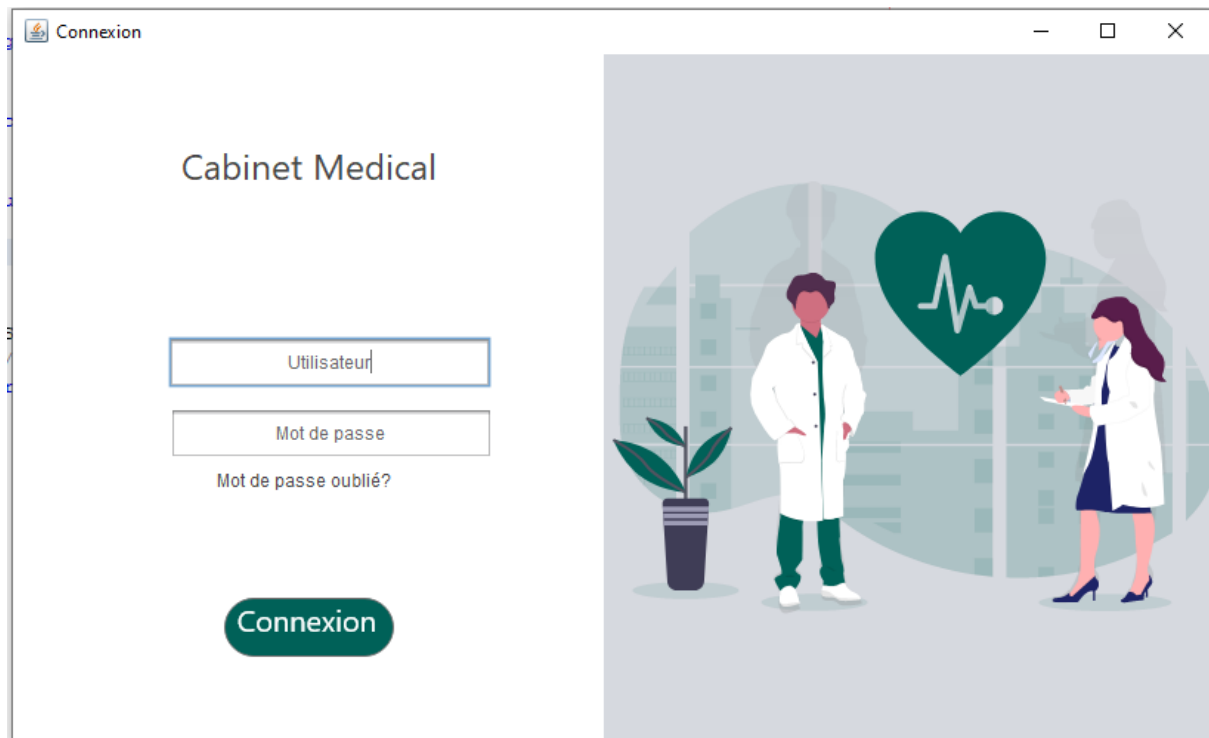
**MySQL Database Service** est un service de base de données entièrement géré pour déployer des applications natives du cloud en utilisant la base de données.



## 3. Interface de l'application

### 3.1. Interface d'authentification

Chaque utilisateur doit s'authentifier pour pouvoir y accéder à l'application



### 3.2. Interfaces de la fiche patient

Cette interface peut être consultée soit par le médecin soit par la secrétaire mais elle est plus utile pour la secrétaire pour qu'elle enregistre les données des patients tout en pouvant les modifier et les consulter.

Créer Fichier Patient X

Ajouter PatientListe des PatientsModifier Patient



NIP...  
Num Immatricule...  
Sexe...  
Groupe Sanguin...  
Civilité...

Adressé par Docteur: Docteur...

Nom...Prénom...

Email...Date de Naissance...

Profession...Situations Familiale

Code Postale...Ville...Adresse...


Nombre Enfant...Téléphone Portable...

Téléphone Professionnelle...Téléphone Domicile...

AnnulerAjouter

Créer Fichier Patient X

Ajouter Patient Liste des Patients Modifier Patient



Adressé par Docteur: Docteur...

Nom... Prénom...

Email... Date de Naissance...

Profession... Situations Familiale...

Code Postale... Ville... Adresse...

Nombre Enfant... Téléphone Portable...

Téléphone Professionnelle... Téléphone Domicille...

NIP... Num Immatricule... Sexe... Groupe Sanguin... Civilité...

Annuler Modifier

Créer Fichier Patient X

Ajouter Patient Liste des Patients Modifier Patient

Rechercher Patient...
Rechercher

nip	num_imm	nom	prenom	adressé_par	tel_portable
326687	140018	fff	fff	maha ben amor	12545
8555555	7777777	z	szsz	maha ben amor	1111
123	12456	a	a	Docteur...	11
90	2	z	z	ashref gadour	77
356	1248	gg	gg	Kaouther Belaalia	147
777	785	a	aa	wlem chebaane	111
17777	77777	lkjk	kl	Kaouther Belaalia	545
7555	75555555	k	s	Docteur...	7878
753333	59987	wqswsqw	xqsqsqx	Kaouther Belaalia	6464



### 3.3. Interface de la fiche consultation

Cette interface peut être consultée soit par le médecin, soit par la secrétaire ou par le médecin, cette interface permet d’afficher la liste des consultations, de retirer une consultation et de créer une consultation tout en entrant les données relatives au patient et des notes durant la consultation.

**Fiche Consultation**

Gestion Des patients

NIP  Civilité  Prénom  Nom  Sit. Familiale

Liste des Consultations

id_consultation	id_patient	dateCons
11	777	01-01-2020

Nouvelle Consultation

Date  Honoraires  CNAM ☐ Cliché ☒

Titre

Il aura une autre consultation dans deux semaine avec Dr. maha ben amor

### 3.4. Interface de la fiche médicaments de la pharmacie

Cette interface peut être consultée par le médecin afin de prescrire les médicaments et créer l'ordonnance après chaque consultation.

Medicaments de la pharmacie

Recherche Rapide

Medicaments

Doliprane

Panadole

Aspirine

Spasmocalme

Voltarène

Ajouter Medicament

Modifier Medicament

Retirer Medicament

Medicament choisi

Panadole

Posologie

Deux fois par jour

Ajouter

Saut de lig...

☐ Imprimer duplicata

☐ Imprimer triplicata

☐ Ordonnance ALD

Panadole 2 boites(s)

Quantite 2

Ordo memorisee

Creer ordonnance

Tout effacer

Annuler

# Conclusion générale

Ce projet a pour objectif de réaliser un système de gestion de base de données d'un cabinet médical permettant aux médecins de gérer les fiches patientes, les fiches consultations et les ordonnances après chaque consultation.

Notre application peut être améliorée en intégrant l'entité client en lui favorisant un espace pour qu'il puisse consulter la disponibilité des médecins pour prendre un rendez-vous, ceci permet aussi de faciliter la gestion des rendez-vous pour les médecins.