

**Ministère de l’enseignement supérieur et de la recherche scientifique**

**\*\*\***

**Université Libre de Tunis**

**\*\*\***

**Département Informatique**

**Cahier de charge**

**Conception et développement d’une application de gestion d’un cabinet de médecin**

**Elaboré par :**

**Hafedh Baazouzi**

**Charfeddine Ben Mohammed**

**Anouer Rouissi**

**Saida Oueslati**

**Encadré par :**

**Mr Nader Ayadi**

**Année Universitaire : 2017/ 2018**

Table de matière

[I. Présentation du projet 4](#_Toc496907148)

[1. Les objectifs : 4](#_Toc496907149)

[2. Description de l’existant : 4](#_Toc496907150)

[II. Spécification des besoins fonctionnels et non fonctionnels : 4](#_Toc496907151)

[1. Besoins fonctionnels : 4](#_Toc496907152)

[2. Besoins non fonctionnels : 5](#_Toc496907153)

[3. Identification des acteurs: 5](#_Toc496907154)

[4. Diagramme de cas d’utilisation : 5](#_Toc496907155)

[1. Maquettes 6](#_Toc496907156)

[Interface d’authentification 6](#_Toc496907157)

[Interface d’inscription : 7](#_Toc496907158)

[Interface agenda 8](#_Toc496907159)

[Interface de gestion des patients 8](#_Toc496907160)

[Interface d’ajout de patient 9](#_Toc496907161)

[Interface de gestion des dossiers des patients : 9](#_Toc496907162)

[Interface messages 10](#_Toc496907163)

[Interface de gestion des détails d’un patient : 10](#_Toc496907164)

[IV. Description techniques : 11](#_Toc496907165)

**Liste des figures**

[Figure 1:diagramme de cas d'utilisation général 5](#_Toc496907913)

[Figure 2: Interface d'authentification 6](#_Toc496907914)

[Figure 3: Interface d'inscription 6](#_Toc496907915)

[Figure 4: Interface Agenda 7](#_Toc496907916)

[Figure 5:Interface de gestion des patients 7](#_Toc496907917)

[Figure 6:Interface d'ajout d'un patient 8](#_Toc496907918)

[Figure 7:Interface de consultation la liste des dossiers 8](#_Toc496907919)

[Figure 8:Interface de gestion des messages 9](#_Toc496907920)

[Figure 9:Interface de détail d’un patient 9](#_Toc496907921)

**Cabinet de médecin**

# Présentation du projet

## Les objectifs :

Pour résoudre les problèmes liés à la procédure du travail d’un médecin nous proposons de concevoir et développer une application web qui permet d’informatiser les tâches de processus de gestion de patients et satisfaire le plus maximum possible les besoins des utilisateurs du système. Alors notre application doit répondre aux objectifs suivants :

* Alléger et faciliter le travail du médecin.
* Utiliser un système de gestion de bases de données performant afin de faciliter la gestion des données (ajout, recherche, suppression, modification).
* Permettre aux utilisateurs d’accéder aux données selon leurs droits d’accès
* Assurer la confidentialité des informations.

## Description de l’existant :

Généralement, la gestion des patients existe mais manuelle et elle est effectuée par le médecin et la secrétaire à travers des documents papier ce qui pose plusieurs problèmes dont on peut citer:

* L’encombrement de la quantité importante des documents papier.
* L’absence d’une base de données et le non archivage des documents rendre la mise à jour des données difficile (Ajout, modification, recherche,…).
* Perte de temps.
* Risque de perte des informations.
* L’accès aux données n’est pas sécurisé.

# Spécification des besoins fonctionnels et non fonctionnels :

## Besoins fonctionnels :

Les fonctionnalités visées suivantes :

* Authentification
* Inscription de patient
* Gestion des dossiers des patients
* Gestion des rendez-vous
* Gestion des patients
* Gestion des messages
* Consultation d’agenda des rendez-vous.

## Besoins non fonctionnels :

Ce sont les exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

Ce sont les exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

Les principaux besoins non fonctionnels de notre application sont :

* **La convivialité :** le système doit être facile à utiliser. En effet les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomique, et adaptées à l’utilisateur.
* **La performance :** le système à développer doit être avant tout performant c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d’une manière optimale.
* **La sécurité :** Le système doit assurer la sécurité et la confidentialité des données.
* **Portabilité :** l’application doit être portable, c'est-à-dire fonctionnelle sur n’importe quel système d’exploitation.

## Identification des acteurs:

Suivant les besoins de notre application on peut présenter trois acteurs :

* **Le Médecin :** l’application doit permettre au médecin de :
* S’authentifier
* Gérer ses messages de communication avec la secrétaire.
* Consulter l’agenda de rend-vous.
* **La secrétaire :** l’application doit permettre à la secrétaire de :
* S’authentifier.
* Gérer les patients.
* Gérer les rendez-vous des patients.
* Gérer les dossiers des patients.
* Gérer ses messages de communication avec le médecin.
* **Le patient :** l’application doit permettre au patient de :
* S’inscrire
* S’authentifier
* Gérer ses rendez-vous

## Diagramme de cas d’utilisation :

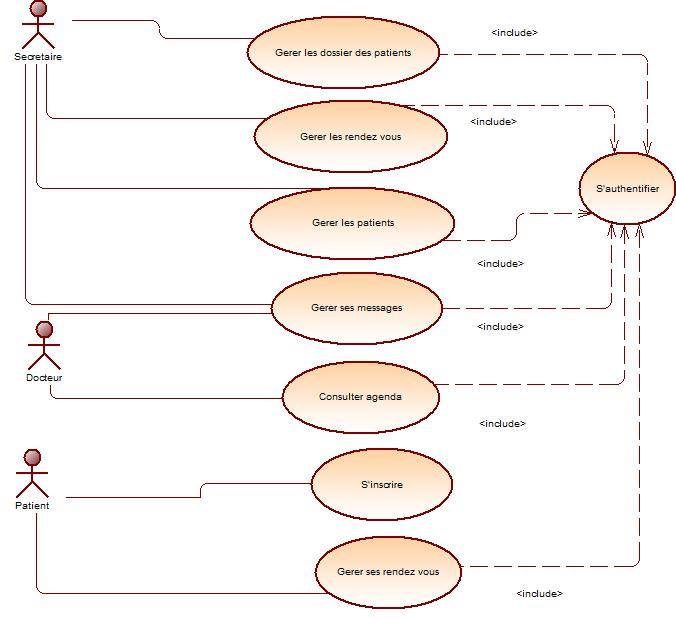


Figure 1:diagramme de cas d'utilisation général

1. **Description graphique**

## Maquettes

### Interface d’authentification

L’authentification c’est la première interface qui s’affiche au lancement de notre application. Un utilisateur veut accéder à notre application doit saisir son login et son mot de passe.

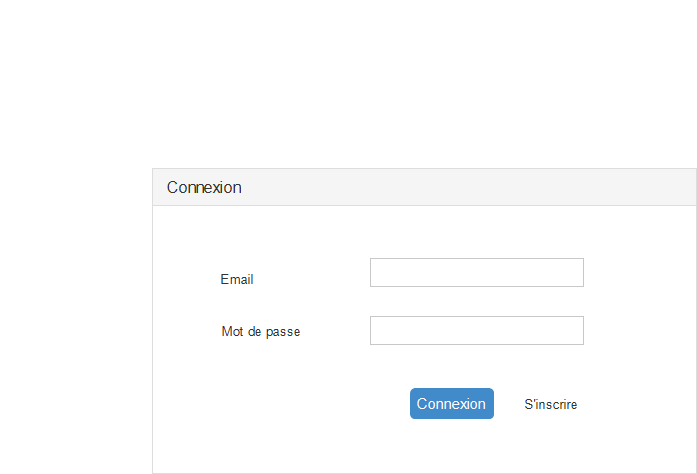
****

Figure 2: Interface d'authentification

### Interface d’inscription :

C’est l’interface qui va permettre à un utilisateur simple de s’inscrire.

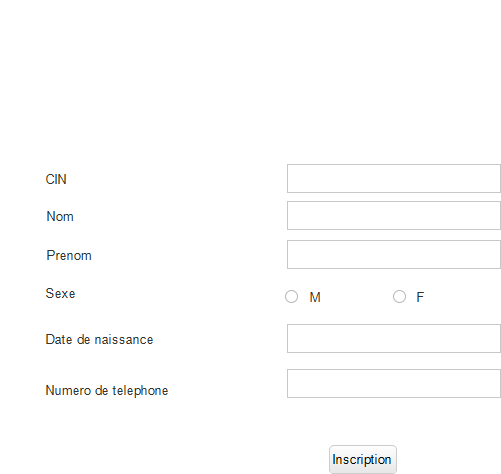
****

Figure 3: Interface d'inscription

### Interface agenda

A partir de cette interface le médecin peut consulter son planning.

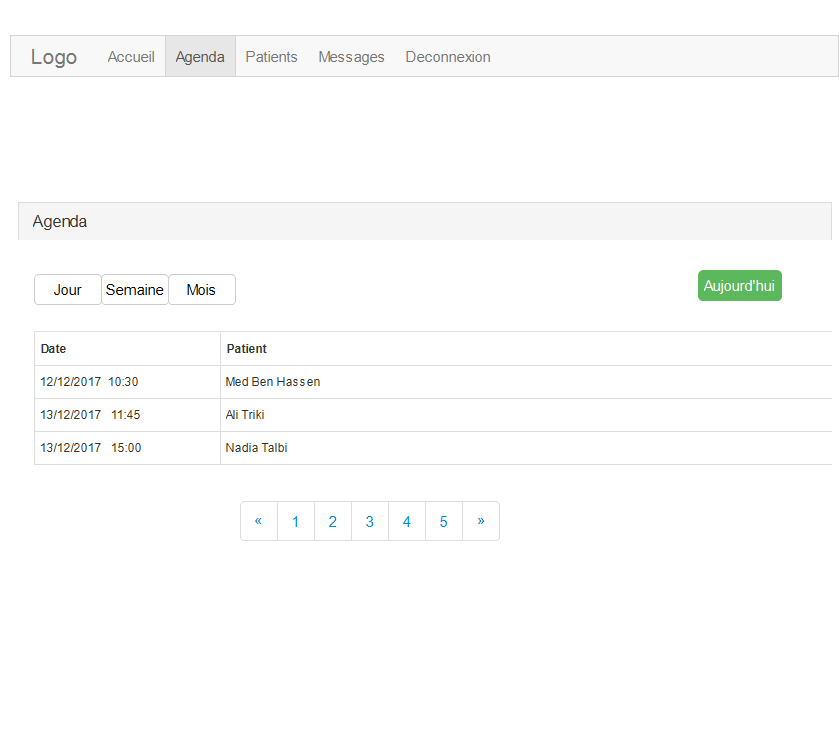


Figure 4: Interface Agenda

### Interface de gestion des patients

L’interface présentée dans la **Figure 5**  la secrétaire peut gérer les patients : elle a le droit de consulter les dossiers et ajouter, modifier, supprimer et consulter un patient, etc.

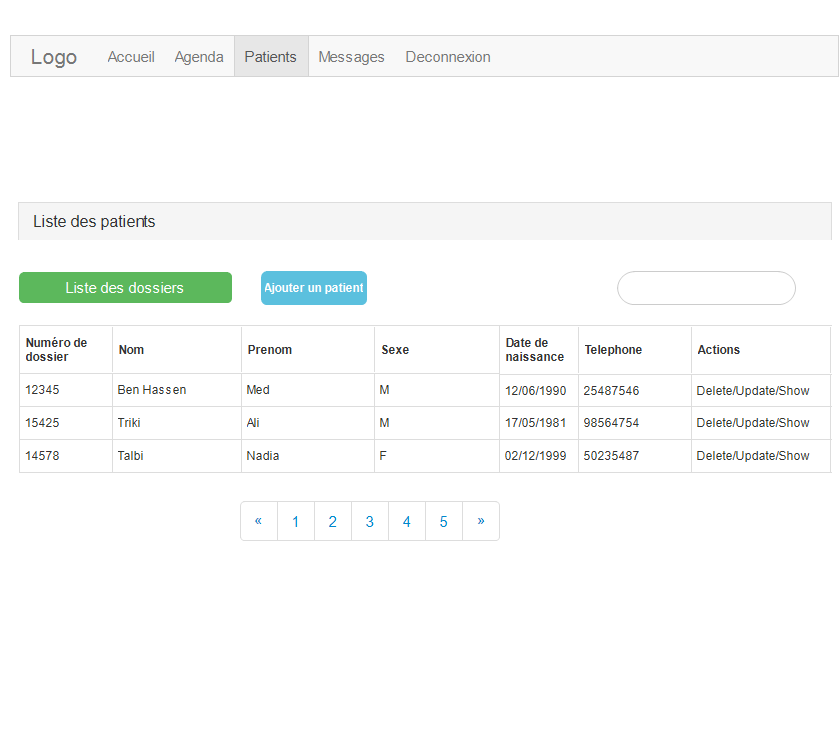


Figure 5:Interface de gestion des patients

### Interface d’ajout de patient

Cette interface présente le formulaire d’ajout d’un patient en remplissant les différents champs de formulaire tels que le CIN, le nom, le prénom etc. puis valider par un clic sur le bouton ajouter.

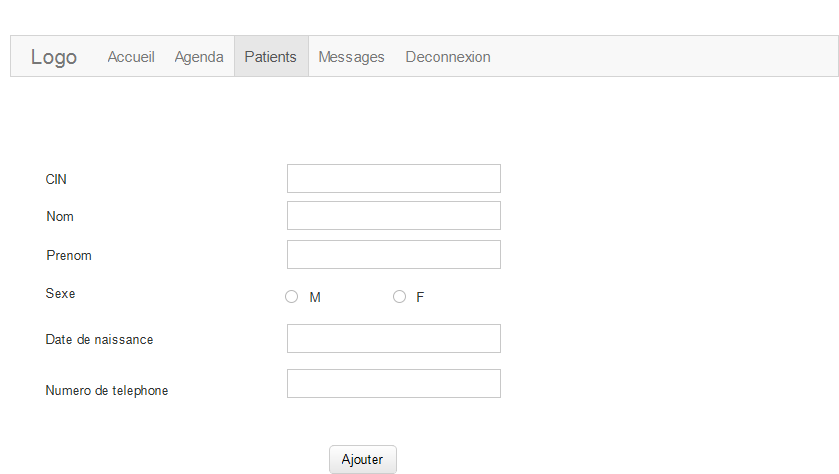


Figure 6:Interface d'ajout d'un patient

### Interface de gestion des dossiers des patients :

A travers cette interface la secrétaire peut chercher et consulter les détails de dossier des patients.

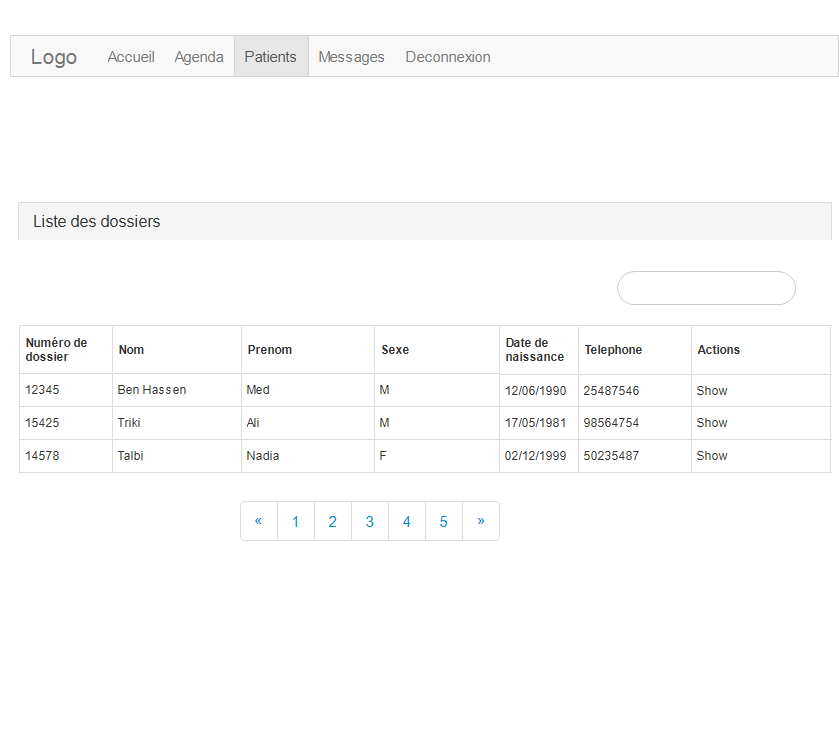
****

Figure 7:Interface de consultation la liste des dossiers

### Interface messages

Cette interface présente une boite de dialogue qui peut faciliter la communication entre le médecin et la secrétaire.

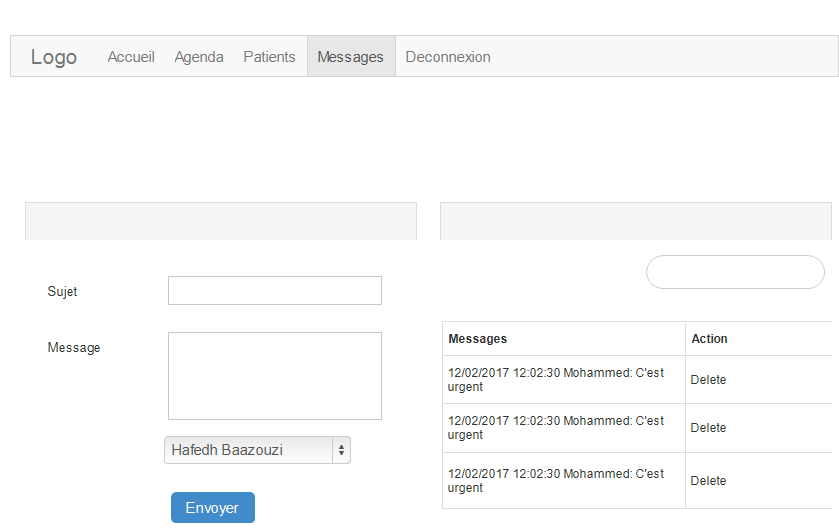
****

Figure 8:Interface de gestion des messages

### Interface de gestion des détails d’un patient :

La suivante présente les détails d’un patient.

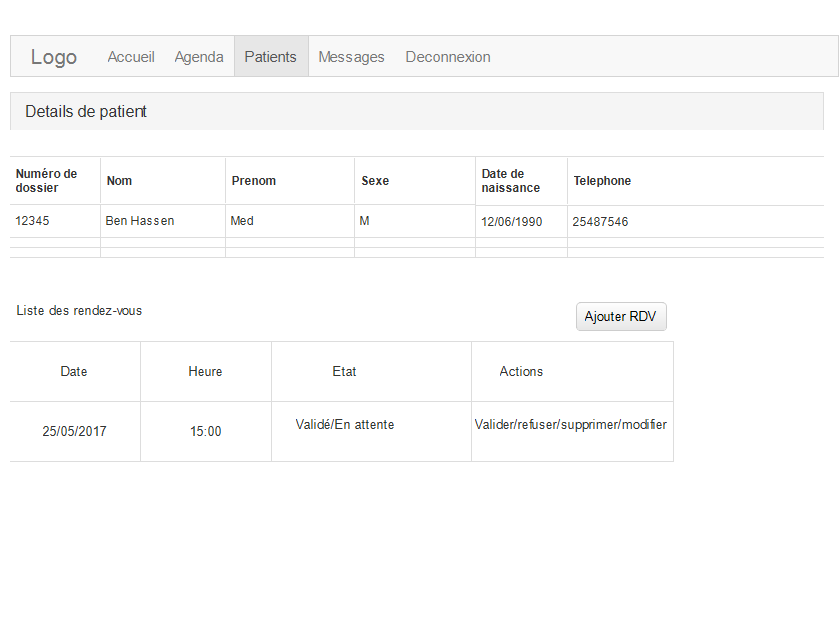
****

Figure 9:Interface de détail d’un patient

# Description techniques :

Les langages imposés : **Java, Html, JavaScript, CSS, AJAX** ...

L’environnement de développement : **Eclipce IDE**.

Pour la gestion de la base des données : **MySQL**

Le test : **Postman**.

🡺L’application doit être compatible avec tous les navigateurs (**chrome, Firefox, IE, Opera**,..)**.**

Préparation des maquettes**: Justin\_mind**.

Conception**: PowerDesigner**