

APPLICATION WEB

GESTION D’UN CABINET DENTAIRE

***Nom et Prénom de l’Etudiant :***

Ayanouz Soufyane

Chaara Abdelkarim

***Intitulé du sujet :***

Conception et réalisation d’une Application Web pour la gestion d’un cabinet dentaire

***Encadrant Pédagogique :***

M. EL AMRANI Chakir Professeur

à la FST de Tanger.

AVANT PROPOS

***Filière :***

1ère année Cycle Ingénieur -Logiciel et système informatique- (LSI - I)

**RESUME**

Dans le cadre du projet de fin du module « Modélisation orientée objet et programmation en C++ » assuré par Mr. El Amrani on a choisi de travailler sur la conception et la réalisation d’une Application Web de gestion d’un cabinet dentaire.

La réalisation de ce projet s’est allongée sur une période d’un mois sous l’encadrement de Mr El Amrani (Professeur au département informatique de la faculté des sciences et techniques de Tanger). Durant cette période nous avons pu améliorer nos connaissances, approfondir et mettre en exergue différentes notions, notamment la conception et modélisation orientée objet en UML et le développement Web.

Le rapport suivant resume les différentes étapes poursuivies pour développer les différentes fonctionnalités de l’application.

* *La gestion de l’application*
* *La gestion du cabinet*
* *La gestion des RDV*

Des outils divers ont été utilisés dans ce projet, tel que le Framework Django, dédié pour le langage Python, dans le but de la conception d’une Application Web moderne respectant le modèle MVT.

**ABSTRACT**

As part of the project to finish the module "Object-oriented modeling and programming in C ++" provided by Mr. El Amrani, we chose to work on the design and realization of a Web application for management of a dental office.

The project was extended over a period of one month under the supervision of Mr El Amrani (Professor in the IT department of the Faculty of Science and Technology of Tangier). During This period, we have been able to improve our knowledge, to deepen and to highlight various notions, in particular the conception and object-oriented modeling in UML and Web development.

The following report summarizes the different steps taken to develop the different functionalities of the application.

*Application management*

*Management of the firm*

*The management of appointments*

Various tools have been used in this project, such as the Django Framework, dedicated to the Python language, with the aim of designing a modern Web application respecting the MVT model.

**Table des matières**

**CHAPITRE 1 :**

[I. Introduction : 6](#_Toc484988230)

[II. Etude de l’existant : 6](#_Toc484988231)

[1. Présentation de l’application : 6](#_Toc484988232)

[2. Critique de l’existant : 7](#_Toc484988233)

[III. La spécification des besoins : 7](#_Toc484988234)

[1. Identification des besoins : 8](#_Toc484988235)

[ Les besoins fonctionnels : 8](#_Toc484988236)

[2. Identification des acteurs : 8](#_Toc484988237)

[3. Use cases générales de l’application : 10](#_Toc484988238)

[IV. Conclusion : 10](#_Toc484988239)

**CHAPITRE 2 :**

[I. Introduction : 11](#_Toc484988240)

[II. Conception : 11](#_Toc484988241)

[1. Diagramme de packages : 12](#_Toc484988242)

[2. Diagrammes de cas utilisation : 13](#_Toc484988243)

[a. Diagramme de cas d’utilisation « gestion de l’application » 13](#_Toc484988244)

[ Cas d’utilisation « authentification » 13](#_Toc484988245)

[ Cas d’utilisation « s’authentifier » : 14](#_Toc484988246)

[b. Diagramme de cas d’utilisation « gestion du cabinet » 15](#_Toc484988247)

[ Cas d’utilisation « Ajouter une fiche patient » : 16](#_Toc484988248)

[ Cas d’utilisation « Modifier une fiche patient » : 16](#_Toc484988249)

[ Cas d’utilisation « Ajouter un RDV » : 17](#_Toc484988250)

[ Cas d’utilisation « Modifier un RDV » : 17](#_Toc484988251)

[ Cas d’utilisation « Suivre un RDV » : 18](#_Toc484988252)

[ Cas d’utilisation « Gérer les consultations » : 18](#_Toc484988253)

[c. Diagramme de cas d’utilisation « gestion de l’application » 20](#_Toc484988254)

[ Cas d’utilisation « Ajouter un utilisateur » : 20](#_Toc484988255)

[ Cas d’utilisation « supprimer un utilisateur » : 21](#_Toc484988256)

[ Cas d’utilisation « Modifier les droits d’accès d’un utilisateur » : 21](#_Toc484988257)

[ Cas d’utilisation « Ajouter un cabinet » : 21](#_Toc484988258)

[ Cas d’utilisation « Supprimer un cabinet » : 22](#_Toc484988259)

[3. Diagrammes de séquences : 22](#_Toc484988260)

[a. DS : Ajouter un utilisateur : 22](#_Toc484988261)

[b. DS : Modifier ou supprimer un utilisateur : 24](#_Toc484988262)

[c. DS : Créer une fiche patient : 25](#_Toc484988263)

[d. DS : Modifier ou Supprimer une fiche patient : 26](#_Toc484988264)

[e. DS : Gestion des ordonnances : 27](#_Toc484988265)

[f. DS : Gestion des RDV : 28](#_Toc484988266)

[4. Diagrammes d’activités : 29](#_Toc484988267)

[a. Diagramme d’activités : gérer les RDV 29](#_Toc484988268)

[b. Diagramme d’activités : facture 30](#_Toc484988269)

[c. Diagramme d’activités : consultation 31](#_Toc484988270)

[5. Diagramme de classes : 32](#_Toc484988271)

[a. Les fonctions du diagramme de classes et leurs provenances : 33](#_Toc484988272)

[6. Model conceptuel de données MCD : 34](#_Toc484988273)

[7. Model Physique de données MPD : 35](#_Toc484988274)

[V. Conclusion : 35](#_Toc484988275)

**CHAPITRE 2 :**

[I. Introduction : 36](#_Toc484988276)

[II. Outils et langage utilisé : 36](#_Toc484988277)

[1. Spécifications techniques : 36](#_Toc484988278)

[a. Configuration matérielle : 36](#_Toc484988279)

[b. Configuration logiciel : 37](#_Toc484988280)

[III. Implémentation : 40](#_Toc484988281)

[1. Architecture générale du projet : 40](#_Toc484988282)

[2. Diagramme de déploiement : 41](#_Toc484988283)

[VI. Réalisation du projet : 42](#_Toc484988284)

[IV. Conclusion : 52](#_Toc484988285)

[DEMONSTRATION VIDEO 53](#_Toc484988286)

[Reference : 54](#_Toc484988287)

***CHAPITRE 1 : Spécifications du système***

# **Introduction :**

Dans ce chapitre on va présenter le cahier des charges du projet. En premier lieu, on commencera par l’étude de l’existant ainsi que l’étude du projet et les solutions proposées et en deuxième lieu, on précisera les besoins fonctionnels

Et enfin, je finirai par la présentation du diagramme de cas d’utilisation.

# **Etude de l’existant :**

Concernant la collecte d’information, On s’est basée sur des informations fournies par mon oncle qui veux informatiser le système informatique de son cabinet

Enfin il faut préciser que la documentation et la visualisation d’un ensemble des solutions web ont été considérées comme une autre source de données qui comprend notamment celles qui sont indispensables pour permettre la mise en place du système.

## **Présentation de l’application :**

Dans le cadre de notre formation, On a eu l’opportunité d’effectuer ce projet qui consiste à développer une application de gestion des cabinets médicaux et en particulier un cabinet de dentiste.

En effet, cette application permet de gérer les patients d’un cabinet, de rentrer les observations de la consultation en cours, de préparer les ordonnances, de les sauvegarder et de gagner le temps d’écriture en utilisant les listes prédéfinies de médicaments, de signes cliniques, d’antécédents familiaux et médicaux, etc.

## **Critique de l’existant :**

La gestion de chaque cabinet médical peut avoir quelques difficultés. Celles-ci peuvent se résumer dans :

* Des problèmes de gestion des rendez-vous avec les patients.
* La répétition des informations relatives aux patients.
* Le manque de confidentialité de l’information.

# **La spécification des besoins :**

Comme les bonnes questions représentent la moitié de la réponse dans la plupart des domaines, en informatique une bonne spécification des besoins est primordiale. En effet, elle représente le travail le plus délicat et le plus significatif, mais elle-même repose sur une bonne spécification des besoins qui n’est autre que la question que doit se poser tout ingénieur au début de son travail " Qu’est ce qu’on veut que nous fassions ? ". Dans cette section nous allons présenter notre réponse à cette question. Donc, cette phase consiste à mieux comprendre le contexte du système, il s’agit de déterminer les fonctionnalités et les acteurs et d’identifier les cas d’utilisation initiaux.

## **Identification des besoins :**

### **Les besoins fonctionnels :**

Les besoins fonctionnels représentent les principales fonctionnalités du système. Ces besoins proviennent généralement des utilisateurs du système. Cette application devra permettre :

**La gestion de l’application** : Cette partie permet l’ajout et la suppression des différents utilisateurs

**La gestion du cabinet** : Elle s’agit de la gestion des fiches des patients, la gestion des consultations, la gestion des ordonnances et la gestion de la caisse.

**La gestion des RDV** : Elle concerne la vérification des rendez-vous ainsi que la possibilité au patient de saisir un rendez-vous via le net directement.

**La Gestion du stock (a développer prochainement)** : Elle sert à contrôler le stock d’un cabinet de dentiste.

## **Identification des acteurs :**

* *L’administrateur* : Il possède le droit d’accéder aux différentes rubriques de l’application afin d’ajouter des nouveaux cabinets au réseau et d’attribuer des niveaux d’accès pour chaque utilisateur au niveau de son profil afin de lui permettre ou pas d’agir sur certaines données au niveau d’un module donné dans un établissement donné.
* *Le médecin* : Le médecin peut être lui-même l’administrateur. Il peut gérer le planning de consultation quotidien. Il peut ajouter, modifier ou supprimer au niveau de la gestion des ordonnances.
* *La secrétaire* : La secrétaire consulte la disponibilité du médecin et réserve un RDV pour un patient. la gestion des fiches de patient Elle peut aussi mettre à jour les fiches des patients et elle gère la facturation (montant et date de chaque facture).
* *Le patient : Le patient peut accéder au planning des visites via internet afin de demander un RDV, il peut modifier ou annuler un RDV.*

***Le déroulement de l’application est représenté par les cas d’utilisation suivants :***

* *Authentification :*

*- Assure la sécurité d’accès à l’application et spécifie le type de l’utilisateur.*

* *Gestion de l’application :*

*- Ajouter, modifier et supprimer un utilisateur.*

* *Gestion du cabinet :*

*- Gérer les fiches de patients : ajouter, modifier et supprimer une ou plusieurs fiches de patients.*

*- Gérer les RDV : ajouter, modifier et supprimer un RDV.*

*- Gérer les factures.*

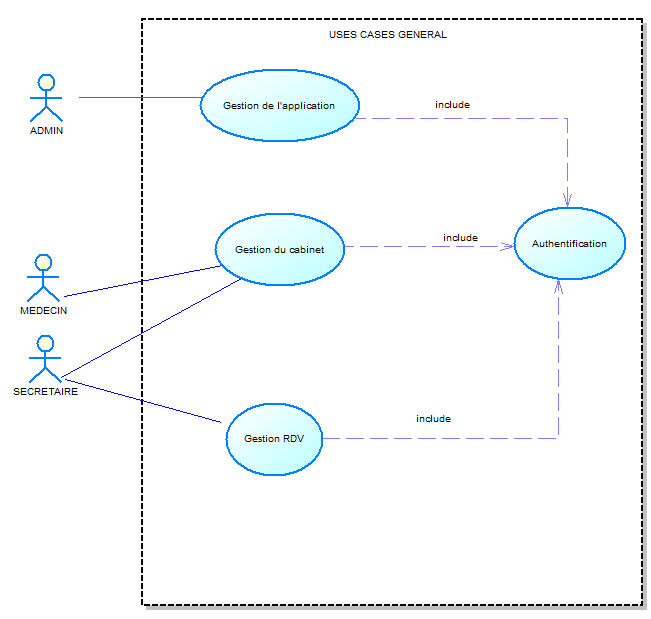
*- Gérer les ordonnances : ajouter, modifier et supprimer une ordonnance.*

*- Gérer les consultations : ajouter, modifier et supprimer une consultation.*

*.*

* *Gestion des RDV : Elle concerne la prise des rendez-vous, même les patients peuvent y accéder*

## **3. Use cases générales de l’application :**



**Figure 1 : Diagramme de cas d'utilisation général**

# **Conclusion :**

*Dans ce chapitre on a posé les questions fondamentales pour spécifier les tâches. Cette spécification va servir dans le prochain chapitre à trouver la réponse la plus convenable à ces questions et ainsi résoudre le problème en question.*

**CHAPITRE 2 : Conception**

# **Introduction :**

Partant du contexte technologique actuel, il semble évident que la conception de toute application informatique n’est autre que le fruit d’efforts conjugués pour avoir une bonne spécification.

Dans ce chapitre on va présenter la conception vis à vis du problème posé. Ce dernier n’est autre que la réalisation d’une application web services du système d’une gestion d’un cabinet de dentiste.

# **II. Conception :**

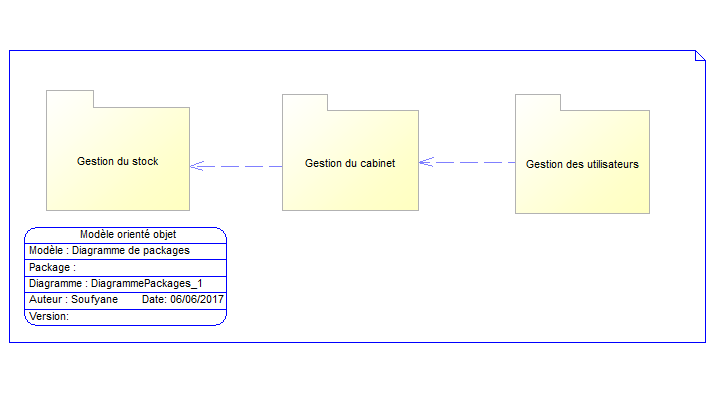
La conception est la phase qui répond à la question « que faut-il faire ? », elle a pour but de se doter d’une vision claire et rigoureuse du problème posé et du système à réaliser en déterminant ses éléments et leurs interactions. Nous allons commencer par une analyse du domaine puis une analyse de l’application.

## **Diagramme de packages :**

Cette application peut être divisé en 3 packages :

* Gestion de l’application OU gestion des comptes utilisateurs
* Gestion du cabinet
* Gestion du stock

Ci-dessous le diagramme de package :

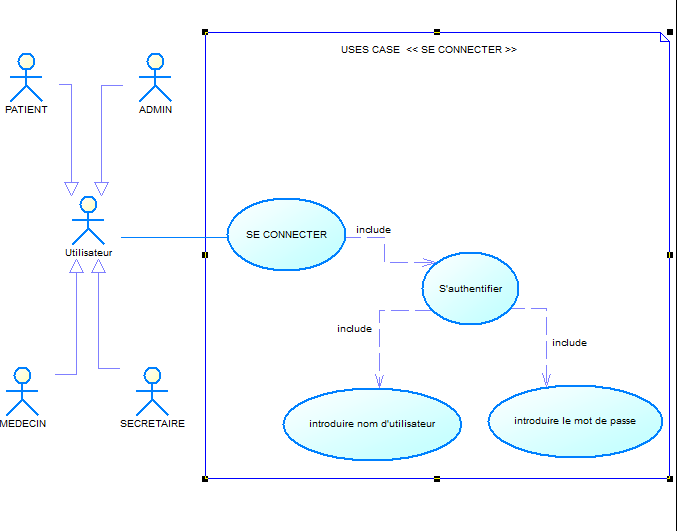


**Figure 2 : Diagramme de package**

## **Diagrammes de cas utilisation :**

## **Diagramme de cas d’utilisation « gestion de l’application »**

### **Cas d’utilisation « authentification »**



**Figure 3 : Diagramme de cas d'utilisation « authentification »**

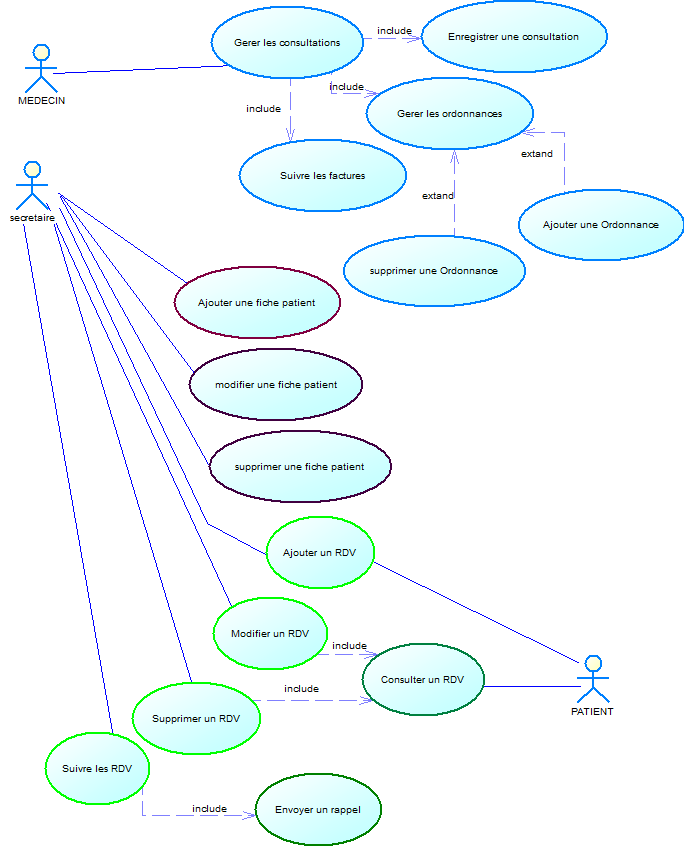
### **Cas d’utilisation « s’authentifier » :**

Titre : s’authentifier

Résumé : chaque utilisateur doit taper son propre nom d’utilisateur et son mot de passe pour accéder à l’interface qui le concerne.

Acteurs : administrateur, médecin, secrétaire et patient.

## **Diagramme de cas d’utilisation « gestion du cabinet »**



**Figure 4 : Diagramme de cas d'utilisation « Gestion du cabinet »**

## **Cas d’utilisation « Ajouter une fiche patient » :**

Titre : Ajouter une fiche patient.

Acteurs : secrétaire

Scénario :

1. L’utilisateur demande le formulaire d’ajout d’une fiche patient

2. Le système affiche le formulaire

3. L’utilisateur remplit le formulaire

4. L’utilisateur valide le formulaire

5. Le système enregistre la demande d’ajout

### **Cas d’utilisation « Modifier une fiche patient » :**

Titre : modifier une fiche patient.

Acteurs : secrétaire

Scénario :

1. L’utilisateur demande le formulaire de fiche patient

2. Le système affiche le formulaire

3. L’utilisateur demande de rechercher l’identifiant du patient

4. Le système effectue la recherche

5. Le système envoie les informations du patient concerné

6. L’utilisateur effectue les modifications nécessaires

7. L’utilisateur valide les modifications

8. Le système enregistre la mise à jour

### **Cas d’utilisation « Ajouter un RDV » :**

Titre : Ajouter un RDV

Acteurs : secrétaire

Résumé : C’est la secrétaire qui se charge de la gestion des RDV, elle consulte le planning des visites et selon la disponibilité du médecin elle ajoute un nouveau RDV. En outre, elle peut modifier ou annuler un RDV selon le choix du patient..

Scénario :

1. L’utilisateur demande la liste des RDV

2. Le système affiche la liste

3. L’utilisateur choisit une date et heure selon la disponibilité

4. Le système vérifie la disponibilité

5. Le système confirme l’ajout

6. Le système enregistre l’ajout

### **Cas d’utilisation « Modifier un RDV » :**

Titre : Modifier un RDV

Acteurs : secrétaire.

Scénario :

1. La secrétaire demande la liste des RDV.

2. Le système affiche la liste.

3. La secrétaire cherche le RDV demandé (par date, Num fiche patient).

4. Le système effectue la recherche

5. Le système affiche le RDV demandé.

6. La secrétaire modifie le RDV.

7. Le système confirme la modification.

8. Le système valide la modification.

### **Cas d’utilisation « Suivre un RDV » :**

Titre : Suivre un RDV

Acteurs : secrétaire.

Résumé : La secrétaire se charge d’envoyer un appel téléphonique, SMS ou email au patient pour suivre son RDV avec le médecin.

Scénario :

1. La secrétaire demande de moyen de suivi de RDV.

2. Le système choisit un moyen (email, SMS)

3. En cas de RDV, la secrétaire envoie manuellement un mail ou un SMS.

### **Cas d’utilisation « Gérer les consultations » :**

* **Sous cas** **« Ajouter une Ordonnance » :**

Acteurs : médecin.

Scénario :

1. Le médecin demande le modèle d’ordonnance

2. Le système injecte les infos (patient, actes) dans le modèle

3. Le médecin ajoute les informations nécessaires.

4. Le système valide l’ajout

5. Le système effectue l’enregistrement

* **Sous cas** **« Supprimer une Ordonnance » :**

Acteurs : médecin.

Scénario :

1. Le médecin demande la liste des ordonnances

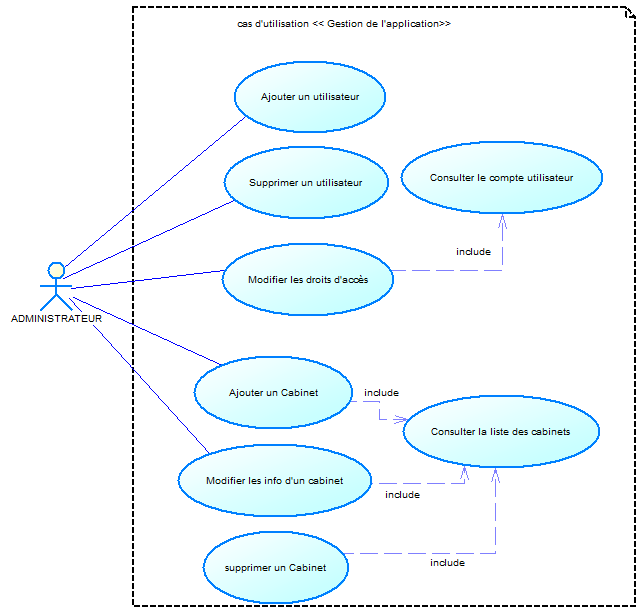
2. Le système affiche la liste

3. Le médecin supprime l’ordonnance demandé

4. Le système valide la suppression

5. Le système effectue l’enregistrement

## **Diagramme de cas d’utilisation « gestion de l’application »**



**Figure 5 : Diagramme de cas d'utilisation « Gestion de l’application »**

### **Cas d’utilisation « Ajouter un utilisateur » :**

Acteurs : Administrateur.

Scénario :

1. L’administrateur tape l’identifiant de l’utilisateur.

2. Le système vérifie l’existence de l’identifiant.

3. L’administrateur lui attribue un mot de passe et un nom d’utilisateur

4. Le système enregistre le nouvel ajout.

### **Cas d’utilisation « supprimer un utilisateur » :**

Acteurs : Administrateur.

Scénario :

1. L’administrateur tape l’identifiant de l’utilisateur.

2. Le système vérifie l’existence de l’identifiant.

3. L’administrateur supprime l’utilisateur ainsi que les coordonnées qui

le concerne.

4. Le système enregistre la suppression.

### **Cas d’utilisation « Modifier les droits d’accès d’un utilisateur » :**

Acteurs : Administrateur.

Scénario :

1. L’administrateur tape l’identifiant de l’utilisateur.

2. Le système vérifie l’existence de l’identifiant.

3. L’administrateur modifie les droits d’accès de l’utilisateur

4. Le système enregistre la modification.

### **Cas d’utilisation « Ajouter un cabinet » :**

Acteurs : Administrateur.

Scénario :

1. L’administrateur tape l’identifiant du cabinet.

2. Le système vérifie l’existence de l’identifiant.

3. L’administrateur introduit les informations du cabinet.

4. Le système enregistre les informations.

5. Le système confirme l’ajout.

### **Cas d’utilisation « Supprimer un cabinet » :**

Acteurs : Administrateur.

Scénario :

1. L’administrateur tape l’identifiant du cabinet.

2. Le système vérifie l’existence de l’identifiant.

3. L’administrateur supprime le cabinet demandé.

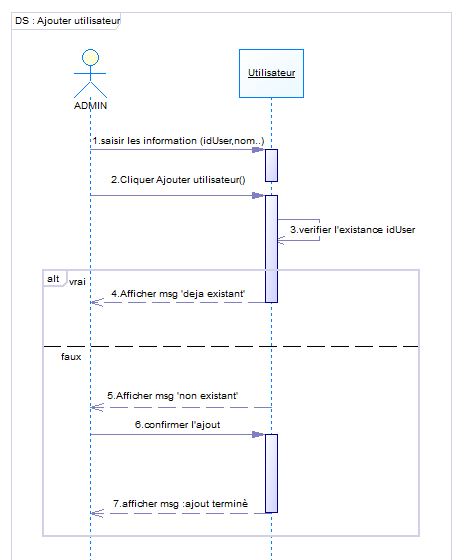
4. Le système enregistre la suppression.

5. Le système confirme la suppression.

## **Diagrammes de séquences :**

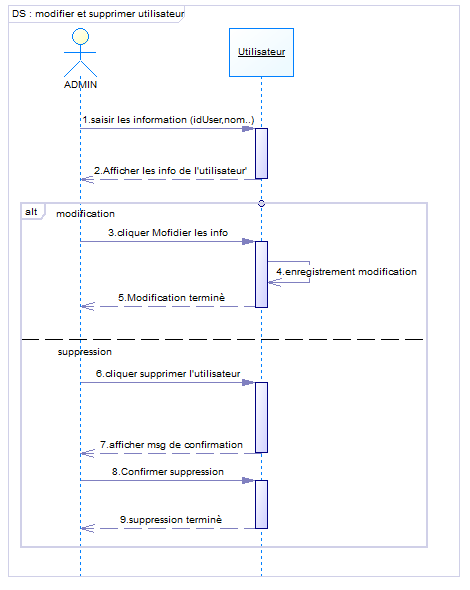
### **DS : Ajouter un utilisateur :**

C’est l’administrateur qui se charge de l’injection d’un nouvel utilisateur dans la base de données. Pour effectuer cette tâche, il accède à son interface, il clique sur le bouton « ajouter un utilisateur » .Le SGBD doit vérifier l’existence de cet utilisateur dans la base et suivant le résultat de la requête, l’utilisateur sera ajouté ou non.



**Figure 6: Diagramme de séquence « Ajouter un utilisateur »**

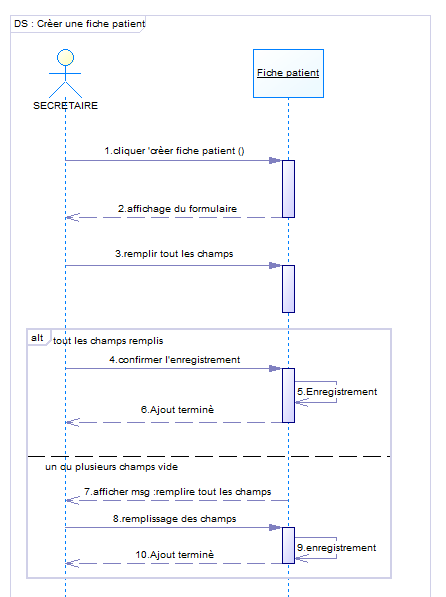
### **DS : Modifier ou supprimer un utilisateur :**



**Figure 7 : Diagramme de séquence « supprimer ou modifier un utilisateur »**

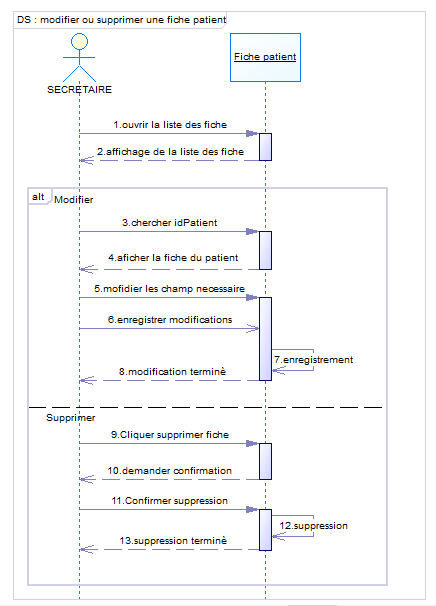
### **DS : Créer une fiche patient :**

Pour ajouter une fiche patient, la secrétaire remplit le formulaire affiché sur l’écran. Le système de gestion de base de données (SGBD), se charge de la vérification de l’existence de la fiche patient dans la base de données, et si cette dernière n’existe pas, une fiche patient sera alors créé et sauvegardé.



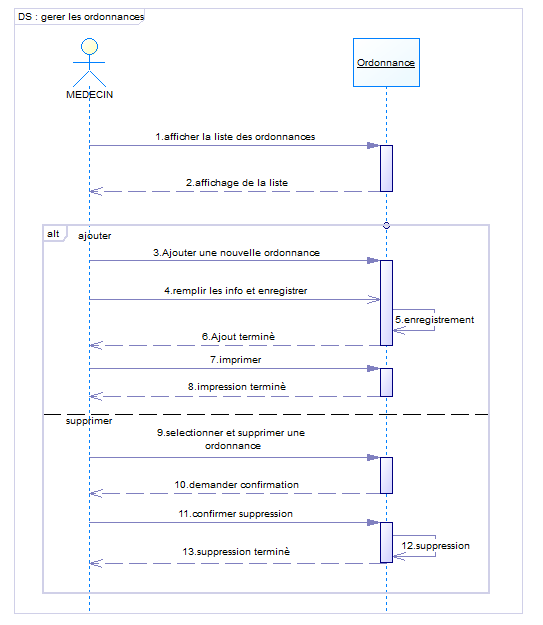
**Figure 7 : Diagramme de séquence « Créer une fiche patient »**

### **DS : Modifier ou Supprimer une fiche patient :**



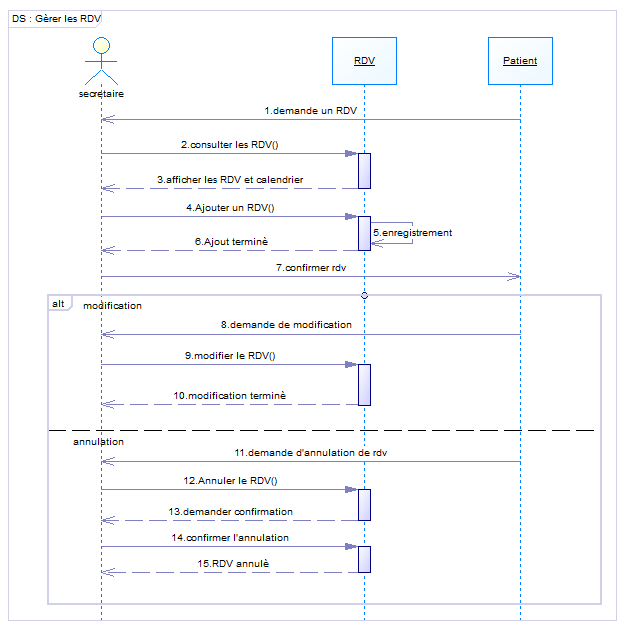
**Figure 8 : Diagramme de séquence « Modifier ou supprimer une fiche patient »**

### **DS : Gestion des ordonnances :**



**Figure 9 : Diagramme de séquence « Gestion des ordonnances »**

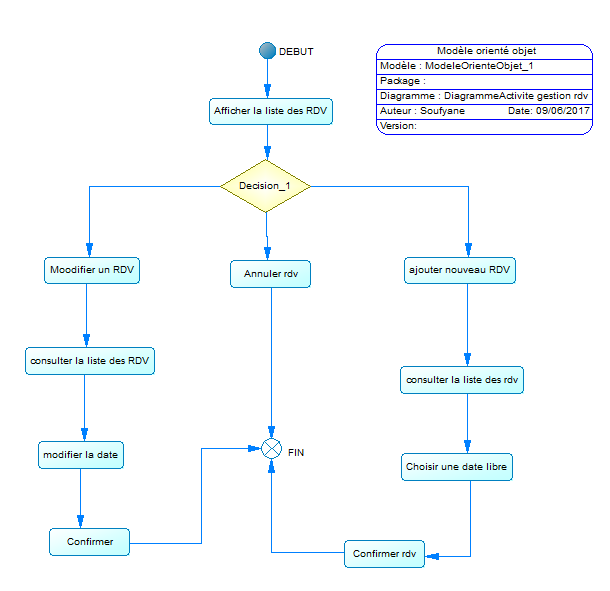
### **DS : Gestion des RDV :**



**Figure 10 : Diagramme de séquence « Gestion des RDV »**

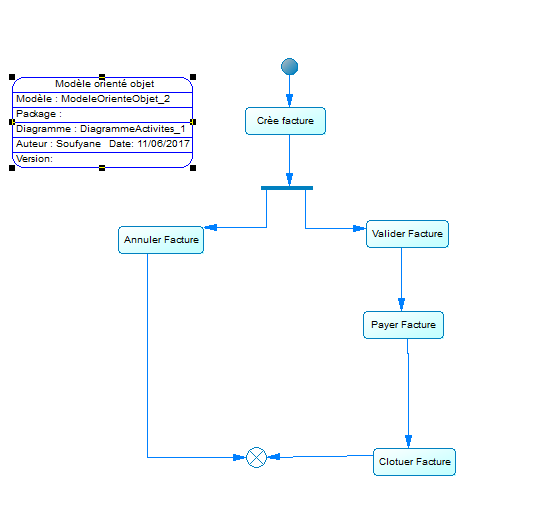
## **Diagrammes d’activités :**

### **Diagramme d’activités : gérer les RDV**



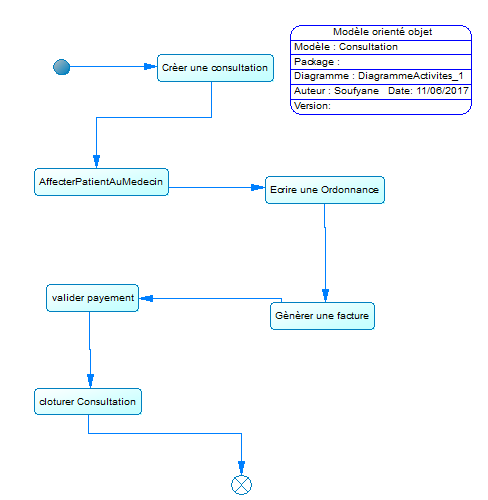
**Figure 11 : Diagramme d’activités « Gestion des RDV »**

### **Diagramme d’activités : facture**



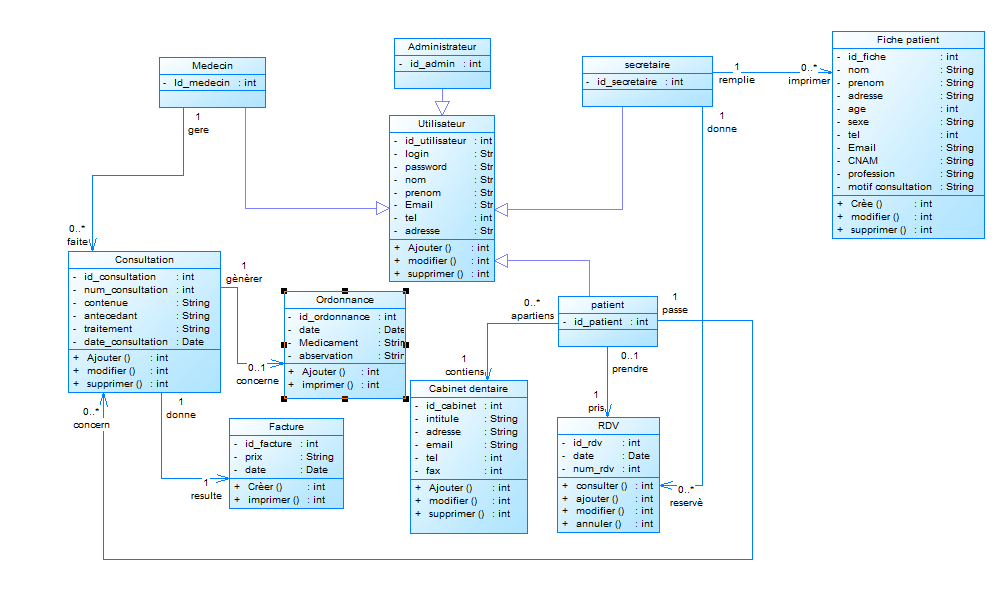
**Figure 12 : Diagramme d’activités « Facture »**

### **Diagramme d’activités : consultation**



**Figure 13 : Diagramme d’activités « Consultation »**

## **Diagramme de classes :**

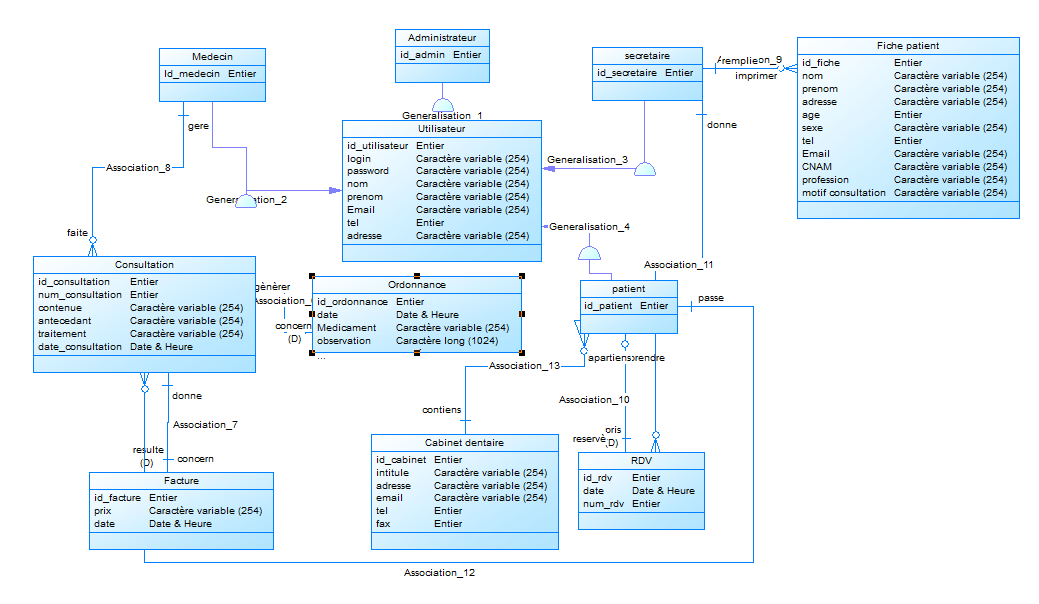
 Les diagrammes de classes représentent bien la structure statique du code, par le biais des attributs et des relations entre classes, mais ils contiennent également les opérations (aussi appelées méthodes) qui décrivent les responsabilités dynamiques des classes logicielles. L’attribution des bonnes responsabilités aux bonnes classes est l’un des problèmes les plus délicats de la conception orientée objet. Pour chaque service ou fonction, il faut décider quelle est la classe qui va le contenir.

**Figure 14 : Diagramme de classes général**

### **Les fonctions du diagramme de classes et leurs provenances :**

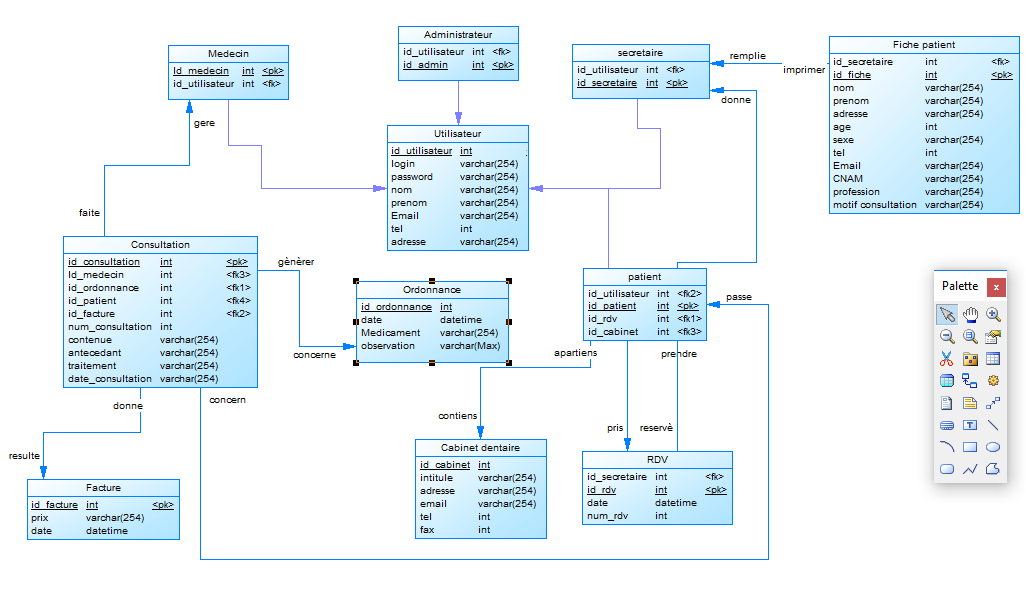
|  |  |
| --- | --- |
| Fonction | Provenance |
| Créer fiche patient | DS : Ajouter fiche patient |
| Modifier fiche patient | DS : modifier ou supprimer fiche patient |
| Supprimer fiche patient | DS : modifier ou supprimer fiche patient |
| Ajouter utilisateur | DS : Ajouter utilisateur |
| Modifier utilisateur | DS : modifier ou supprimer utilisateur |
| Supprimer utilisateur | DS : modifier ou supprimer utilisateur |
| Ajouter une ordonnance | DS : gérer les ordonnances |
| Supprimer ordonnance | DS : gérer les ordonnances |
| Imprimer une ordonnance | DS : gérer les ordonnances |
| Consulter RDV | DS : gérer les RDV |
| Ajouter RDV | DS : gérer les RDV |
| Modifier RDV | DS : gérer les RDV |
| Annuler RDV | DS : gérer les RDV |
|  |  |

## **Model conceptuel de données MCD :**



**Figure 15 : Model conceptuel de donnés**

## **Model Physique de données MPD :**



**Figure 16 : Model physique de donnés**

# **Conclusion :**

Ce chapitre m’a permis de présenter la démarche que j’ai suivie pour résoudre le problème en question, que ce soit par la description de la hiérarchie du site ou même par le diagramme des classes présentant le squelette de la base de données. Mais, il reste tout de même la partie concrète ; la réalisation qui sera le sujet du chapitre suivant.

**CHAPITRE 3 : Réalisation**

# **Introduction :**

La réalisation vient couronner le travail de l’étude préalable et de l’étude conceptuelle. Elle présente la dernière étape et elle est très importante puisque grâce à elle le projet informatique va exister réellement, sa réussite est conditionnée par une multitude de choix essentiellement d’ordre technique concernant l’exécution de ce qui a été conçu et proposé comme solution afin de répondre aux besoins des utilisateurs et de remédier aux insuffisances perçues.

Ce chapitre est donc consacré à la présentation de l’environnement matériel et logiciel nécessaire pour implémenter cette structure

# **Outils et langage utilisé :**

## **Spécifications techniques :**

Une étape intéressante de ce projet était la mise en place de l’environnement matériel et logiciel nécessaire pour la conception et le développement de l’application. Dans ce qui suit, nous présenterons l’environnement logiciel et matériel exploité dans notre projet.

### **Configuration matérielle :**

Ordinateur portable : MacBook PRO

Système d’exploitation : MAC

Processeur : Core I7

Mémoire : 8 G RAM

Disque dur : 320 GO

### **Configuration logiciel :**

* **Phyton 2.7**

Le Python est un langage de programmation orienté Object similaire au Perl et Ruby. Sa force Majeure est qu'il propose des milliers de modules qui vous feront gagner un temps précieux durant le développement de vos applications.

Pour ce qui est de l'apprentissage, le Python est très intuitif grâce à son typage dynamique fort et a sa communauté présente sur les forums. Entre autres, des sites sont disponibles pour apprendre Rapidement ce langage

* **Django 1.11.2**

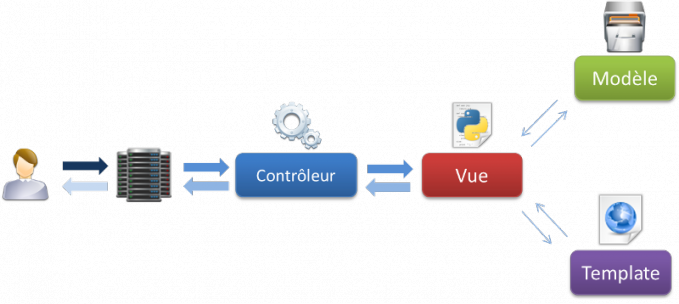
Django est un Framework web Python de haut niveau qui encourage le développement rapide et propre. Gratuit et open source, Django vous permet d'éviter de réinventer la roue grâce à toutes les libraires disponibles en python, mais aussi de tout ce que ce Framework offre dès son installation.

Le fonctionnement d'une application Django se divise en 2 parties.

La première section a pour but de préparer l'étape entre l'utilisateur et l'application en elle-même.

Cette section gère la relation entre les bases de données et les applications Django, mais aussi s’occupe du routage via des règles URL.

La seconde section est l'application. Organisée en modèles, vues et Template, l'application se trouve au cœur du projet Django.

* **Model MVT (model view Template)** :

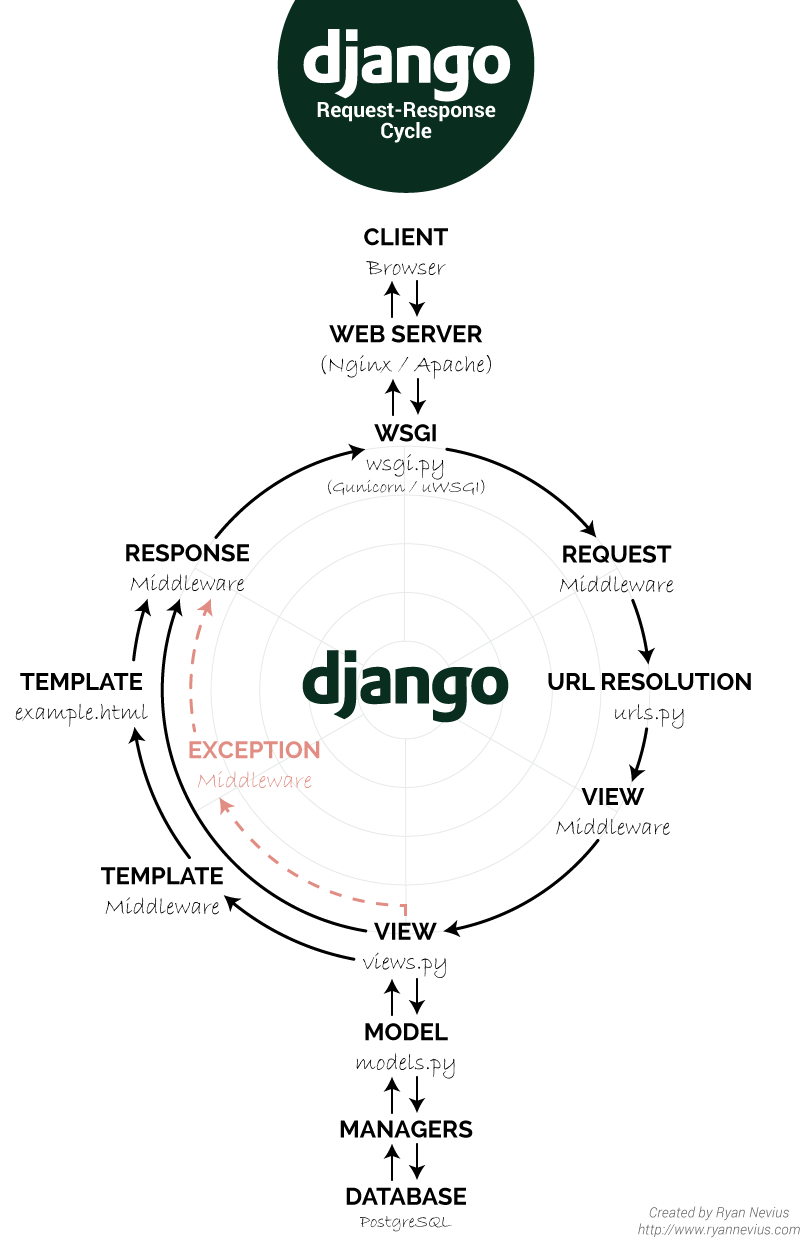
Django se base sur le modèle MVT, légèrement différent du modèle MVC, le Framework gère lui-même le contrôleur et laisse place au Template.

Lors d'une requête venant de l'utilisateur, le Framework Django gère lui-même, via les règles de routage défini par le développeur, de charger la bonne vue correspondante au résultat voulu.

Une Template est un fichier HTML qui sera récupéré par la vue pour être envoyer à l'utilisateur, Mais entre cette étape, Django va exécuter la Template comme si c'était un fichier de code.

Inclus dans les Template, le Framework propose l'utilisation des structures conditionnelles, des boucles, des variables... permet d'avoir une grande liberté de développement.

* **Présentation du cycle de Django :**



**Figure 17 : Présentation du cycle de Django**

# **Implémentation :**

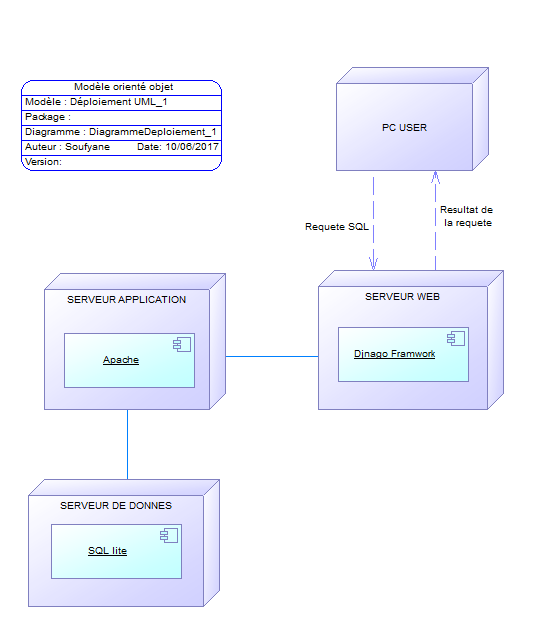
## **Architecture générale du projet :**

Cette figure représente le fonctionnement de notre système ; les utilisateurs sont interconnectés au serveur d’application, qui permet d’exécuter les requêtes envoyées par ces derniers en se connectant à la base de données qui traite la requête, extrait les données et envoi le résultat de la requête au serveur.



**Figure 18 : Architecture générale du projet**

## **Diagramme de déploiement :**



**Figure 19 : Diagramme de déploiement**

# **Réalisation du projet :**

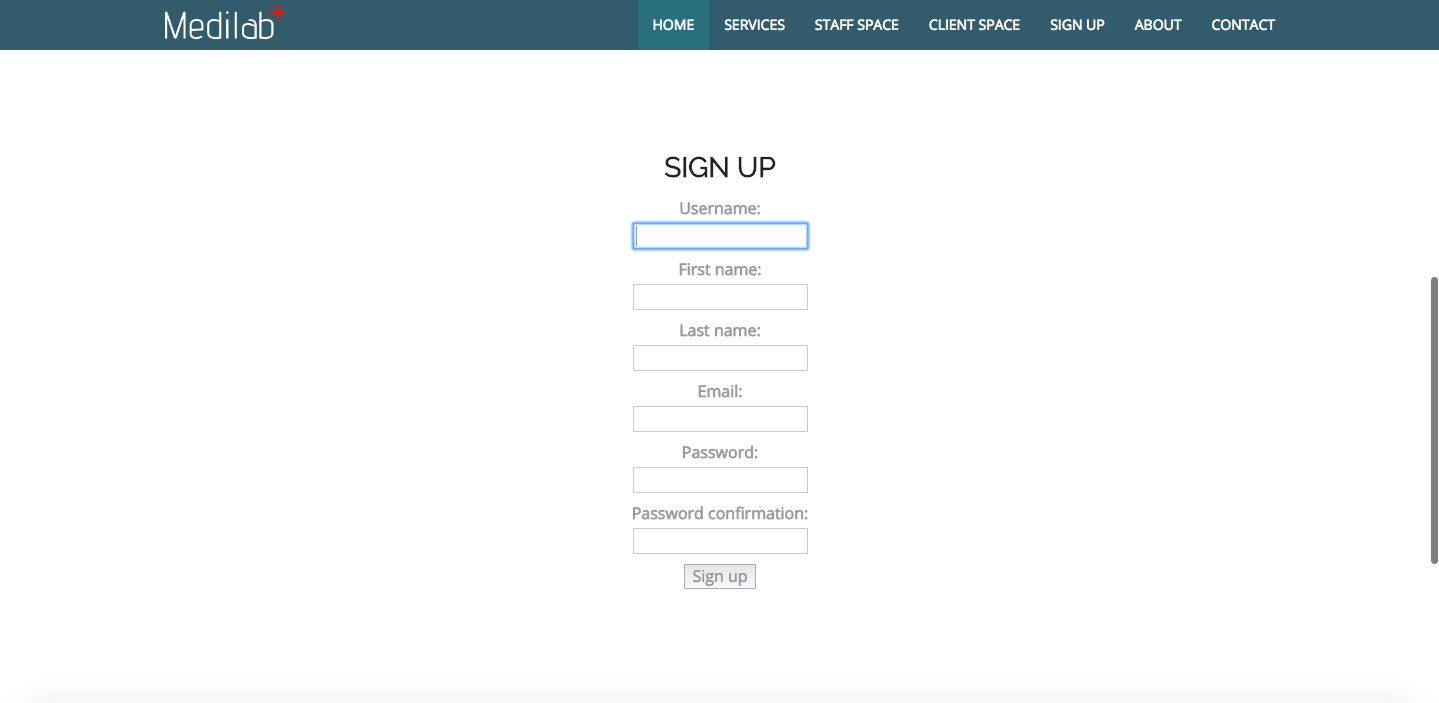
On va présenter dans cette partie quelques interfaces utilisateurs de notre application.

* **La page d’accueil :**



**Figure 20 : page d’accueil de l’application**

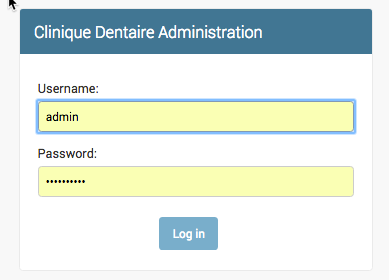
* **Interface d’inscription** :



**Figure 21 : interface d’inscription**

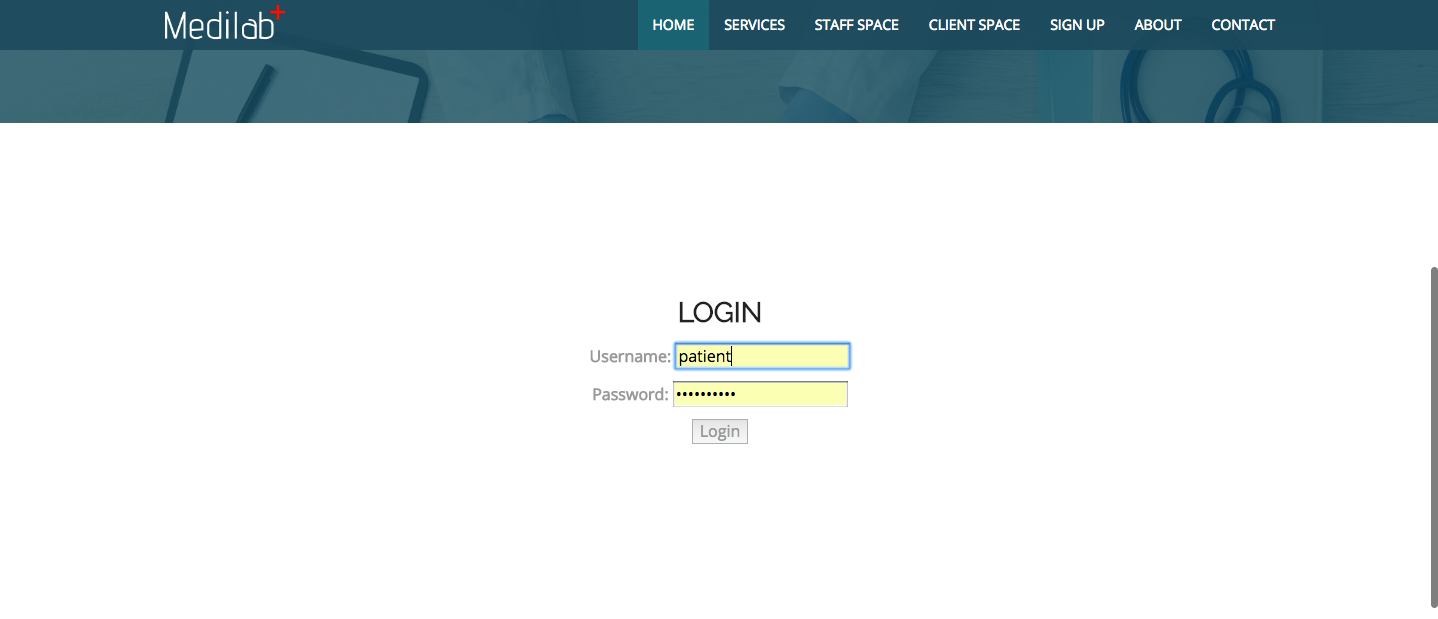
* **Interface d’Authentification**

L’authentification est un certificat de sécurité du système, et à partir d’elle chaque utilisateur authentifié peut accéder à notre application. Pour se faire l’utilisateur introduit son nom d’utilisateur, son mot de passe comme le montre la figure ci-dessous.



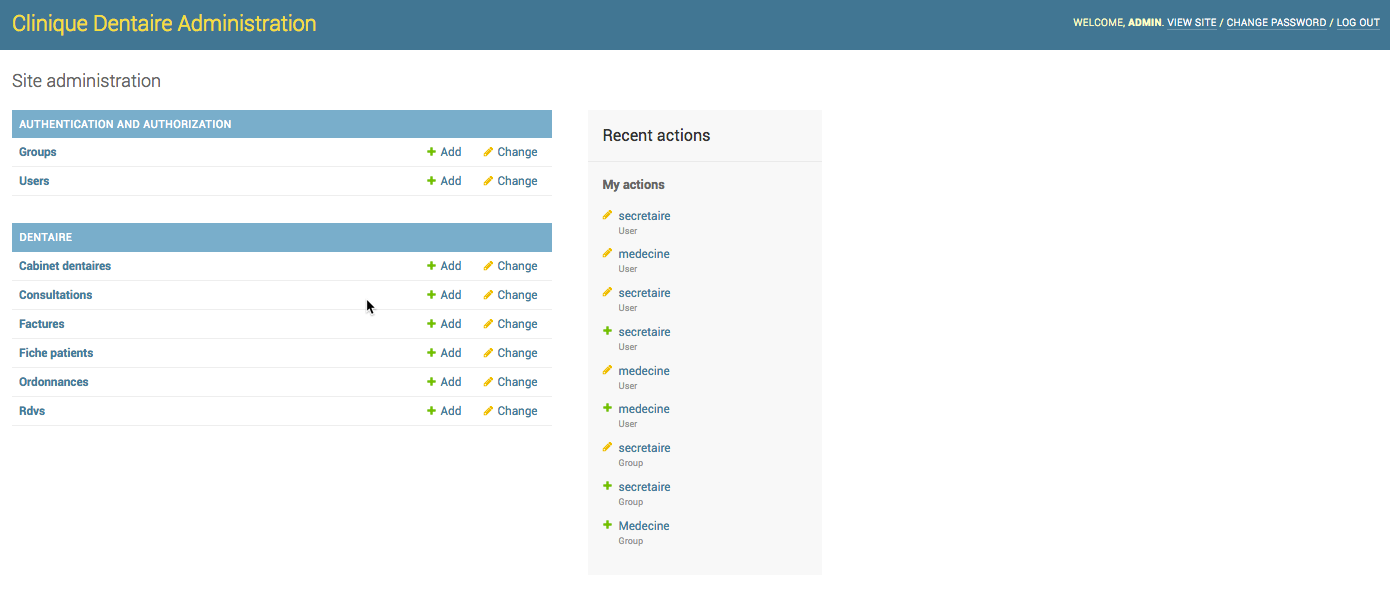
**Figure 22 : interface d’authentification des utilisateurs de l’application**

* **Interface d’Authentification**

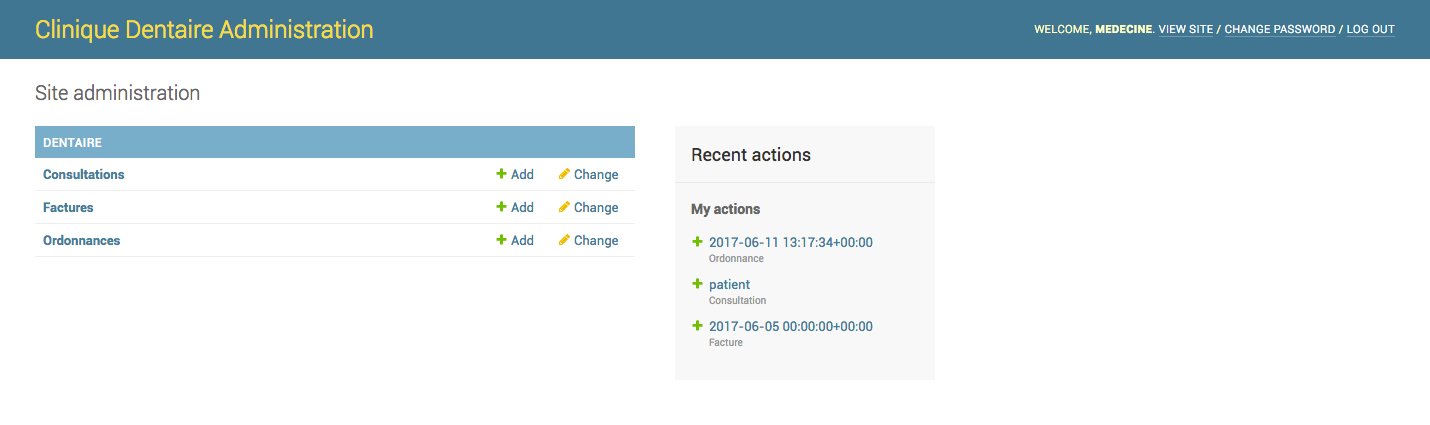


**Figure 23 : interface d’authentification des patients**

* **Interface de l’administrateur de l’application**

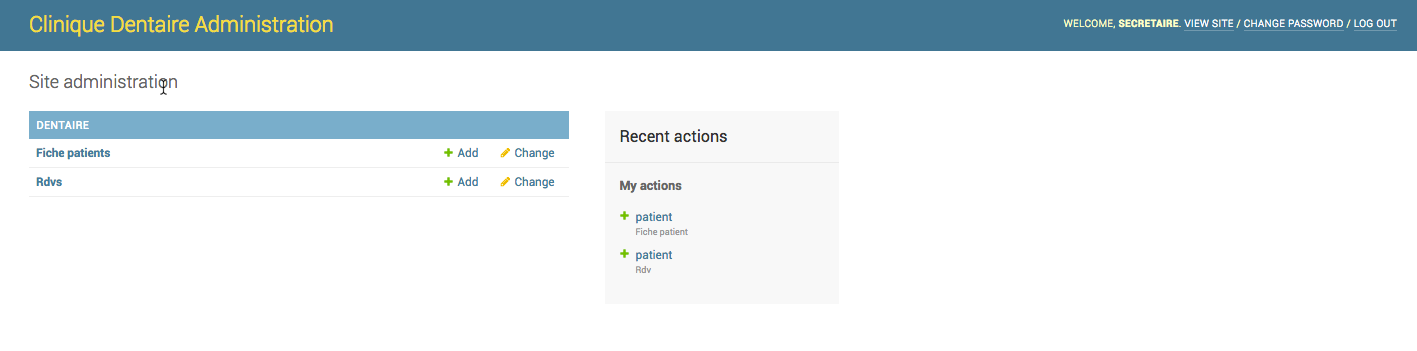


**Figure 24 : interface d’administration de l’application**

* **Interface du médecin :**

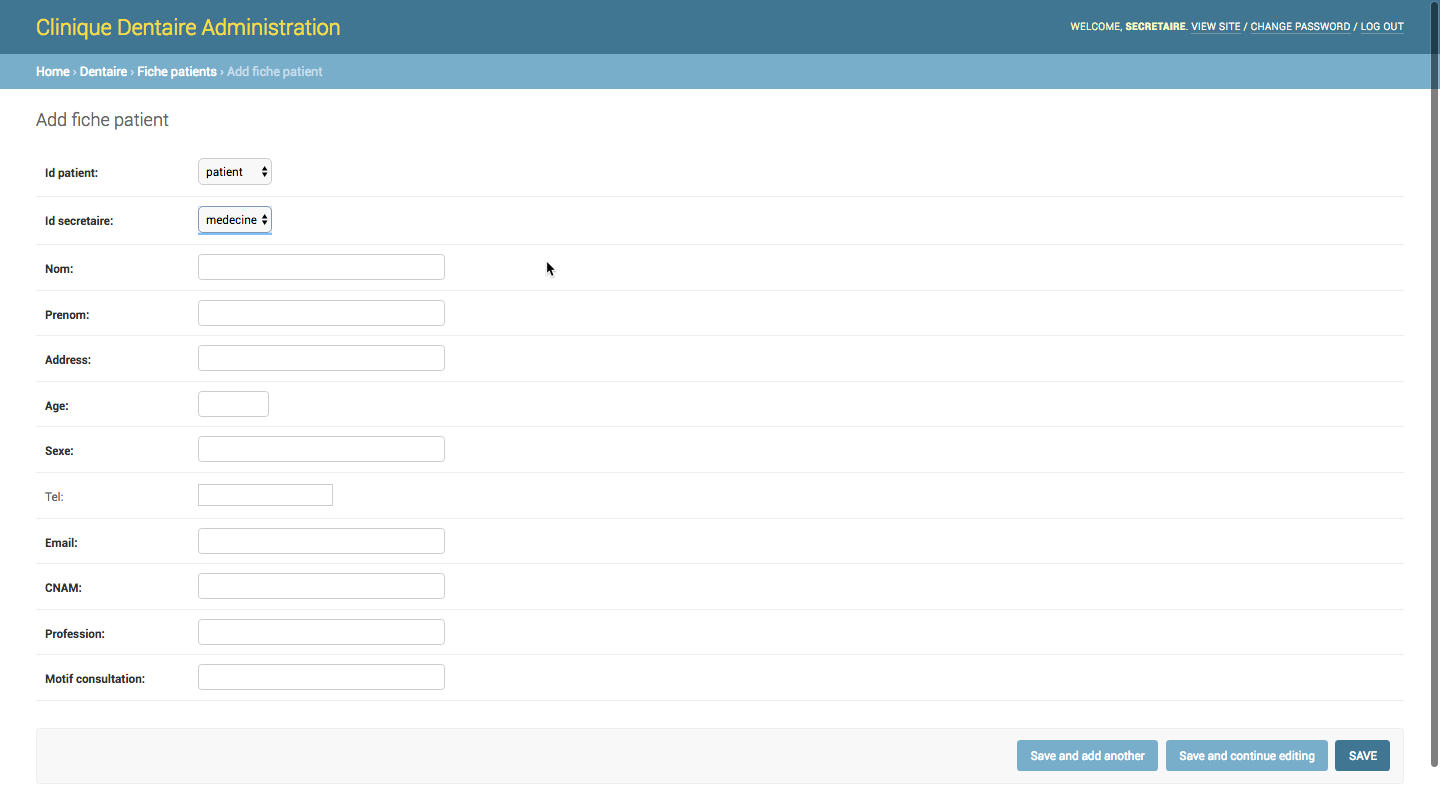
**Figure 25 : interface du médecin**

* **Interface de la secrétaire :**



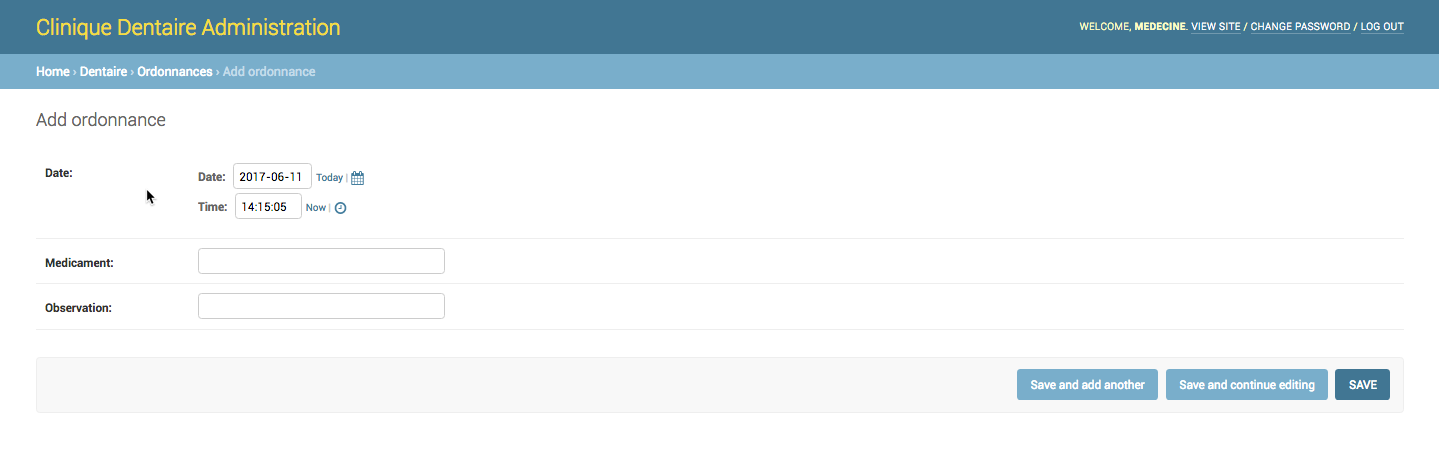
**Figure 26 : interface de la secrétaire**

* **Interface pour créer une fiche patient :**



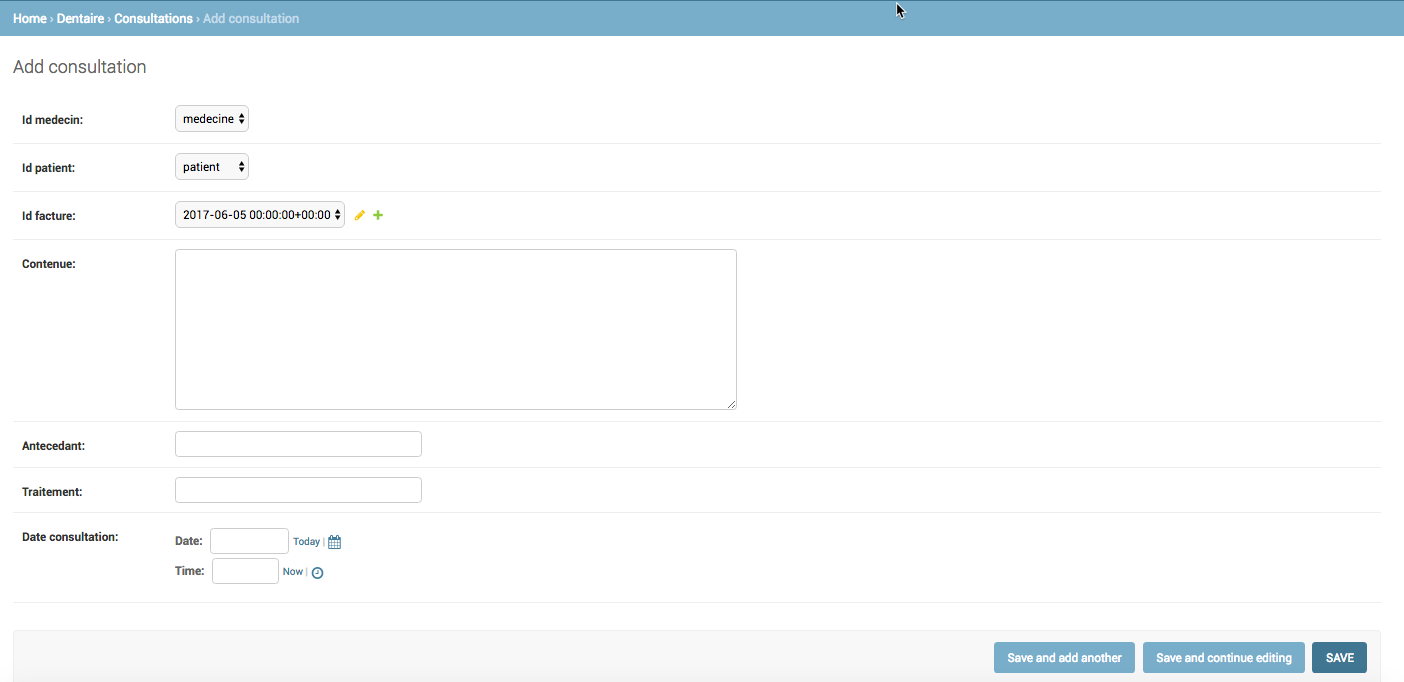
**Figure 27 : « Créer une fiche patient »**

* **Interface pour créer une Ordonnance :**



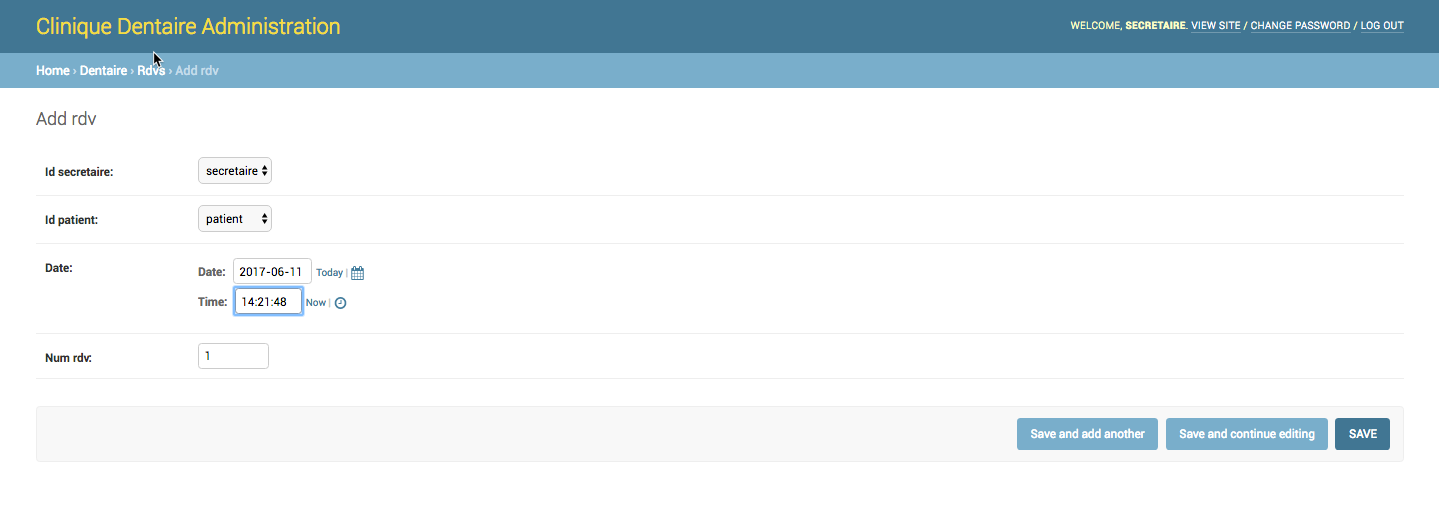
**Figure 28 : « Créer une ordonnance »**

* **Interface pour créer une consultation :**



**Figure 29 : « Créer une consultation »**

* **Interface pour prendre un RDV :**



**Figure 30 : « prendre un RDV »**

# **Conclusion :**

Dans ce chapitre, j’ai présenté les différentes techniques utilisées pour implémenter le système ainsi que, les principes généraux d’ergonomie qui nous ont aidés à fournir des interfaces conviviales et qui permettent aux utilisateurs de réaliser leurs tâches d’une manière plus efficace.

# **DEMONSTRATION VIDEO**

Double clique pour voir la vidéo



# **Reference :**

* **https://docs.djangoproject.com**
* **https://stackoverflow.com**
* [**https://stackexchange.com**](https://stackexchange.com)
* **www.Youtube.com**