Table des matières

[Liste des figures 3](#_Toc387053558)

[Liste des tableaux 5](#_Toc387053559)

[Introduction 6](#_Toc387053560)

[Chapitre 1. Conception 7](#_Toc387053561)

[1. Introduction 7](#_Toc387053562)

[2. Diagramme de Classe 7](#_Toc387053563)

[2.1. Description générale 7](#_Toc387053564)

[2.2. Dictionnaire de données 8](#_Toc387053565)

[2.3. Diagramme de cas d'utilisation 12](#_Toc387053566)

[2.3.1. Diagramme de cas d’utilisation générale 12](#_Toc387053567)

[2.3.2. Diagramme des cas d'utilisation relatif à la gestion des patients 13](#_Toc387053568)

[2.3.3. Diagramme des cas d'utilisation relatif à la gestion des Consultations 14](#_Toc387053569)

[2.3.4. Diagramme des cas d'utilisation relatif à la gestion des utilisateurs 15](#_Toc387053570)

[2.4. Diagramme de Séquence 15](#_Toc387053571)

[2.4.1. Gestion des patients 15](#_Toc387053572)

[2.4.2. Gestion des rendez-vous 19](#_Toc387053573)

[2.4.3. Gestion des consultations 21](#_Toc387053574)

[Chapitre 2. Choix techniques 24](#_Toc387053575)

[1. Introduction 24](#_Toc387053576)

[2. Choix techniques 24](#_Toc387053577)

[2.1. Choix du modèle MVC 24](#_Toc387053578)

[2.2. Eclipse 24](#_Toc387053579)

[2.3. Framework Hibernate 24](#_Toc387053580)

[2.4. Log4j 25](#_Toc387053581)

[2.5. JasperReports 25](#_Toc387053582)

[2.6. PostgreSQL 26](#_Toc387053583)

[2.7. JUnit 27](#_Toc387053584)

[2.8. Subversion (SVN) 27](#_Toc387053585)

[2.9. Serveur SVN utiliser dans le projet 28](#_Toc387053586)

[2.10. Balsamiq Mockups 28](#_Toc387053587)

[3. Maquette des écrans 28](#_Toc387053588)

[3.1. Maquette de l'interface d’authentification 28](#_Toc387053589)

[3.2. Maquette de l'interfaces principales 29](#_Toc387053590)

[3.3. Maquette de l'interface gestion patient 30](#_Toc387053591)

[3.4. Maquette de l'interface ajouter patient 30](#_Toc387053592)

[3.5. Maquette de l'interface de gestion des rendez-vous 31](#_Toc387053593)

[3.6. Maquette de l'interface ajouter un rendez-vous 31](#_Toc387053594)

[3.7. Maquette de l'interface ajouter consultation 32](#_Toc387053595)

[3.8. Maquette de l'interface ajouter ordonnance 32](#_Toc387053596)

[3.9. Maquette de l'interface de gestion des Médicament 33](#_Toc387053597)

[3.10. Maquette de l'interface ajouter Médicament 33](#_Toc387053598)

[3.11. Maquette de l'interface modifier dossier médicale 34](#_Toc387053599)

[Chapitre 3. Réalisation 35](#_Toc387053600)

[1. Introduction 35](#_Toc387053601)

[2. Implémentation de l'architecteur Modèle-Vue-Contrôleur (MVC) 35](#_Toc387053602)

[3. L’internationalisation de l'application (I18N) 36](#_Toc387053603)

[4. Création de base de données avec Hibernate 36](#_Toc387053604)

[5. Présentation des interfaces graphique 37](#_Toc387053605)

[5.1. Interface graphique d'authentification 37](#_Toc387053606)

[5.2. Interface graphique d'accueil 37](#_Toc387053607)

[5.3. Interface graphique de Gestion des patients 38](#_Toc387053608)

[5.4. Interface graphique ajouter patient 39](#_Toc387053609)

[5.5. Interface graphique gestion des rendez-vous 40](#_Toc387053610)

[5.6. Interface graphique ajouter rendez-vous 40](#_Toc387053611)

[5.7. Interface graphique de gestion des médicaments 41](#_Toc387053612)

[5.8. Interface graphique d'ajout d’un médicament 41](#_Toc387053613)

[Conclusion 42](#_Toc387053614)

[Bibliographie 43](#_Toc387053615)

# Liste des figures

[Figure 1 : Diagramme de classe 7](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053515)

[Figure 2 : Diagramme de cas d’utilisation générale de la gestion de cabinet de médicine 13](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053516)

[Figure 3 : Diagramme des cas d'utilisation relatif à la gestion des employés par patients 14](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053517)

[Figure 4 : Diagramme des cas d'utilisation relatif à la gestion des médicaments 14](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053518)

[Figure 5 : Diagramme des cas d'utilisation relatif à la gestion des utilisateurs 15](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053519)

[Figure 6 : Diagramme de séquence du scénario nominal du cas d’utilisation s’authentifier 16](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053520)

[Figure 7 : Diagramme de séquence du scénario nominal du cas d’utilisation Ajouter patient 16](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053521)

[Figure 8 : Diagramme de séquence du scénario nominal du cas d’utilisation modifier un patient 17](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053522)

[Figure 9 : Diagramme de séquence du scénario nominal du cas d’utilisation archivé patient 18](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053523)

[Figure 10 : Diagramme de séquence du scénario nominal du cas d’utilisation chercher patient 18](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053524)

[Figure 11 : Diagramme de séquence du scénario nominal du cas d’utilisation ajouter un rendez-vous 19](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053525)

[Figure 12 : Diagramme de séquence du scénario nominal du cas d’utilisation modifier rendez-vous 20](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053526)

[Figure 13 : Diagramme de séquence du scénario nominal du cas d’utilisation supprimer un rendez-vous 20](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053527)

[Figure 14 : Diagramme de séquence du scénario nominal du cas d’utilisation s’authentifier 21](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053528)

[Figure 15 : Diagramme de séquence du scénario nominal du cas d’utilisation ajouter consultation 22](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053529)

[Figure 16 : Diagramme de séquence du scénario nominal du cas d’utilisation modifier une consultation 22](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053530)

[Figure 17 : Diagramme de séquence du scénario nominal du cas d’utilisation chercher une consultation 23](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053531)

[Figure 18 : Maquette de l'interface d’authentification 29](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053532)

[Figure 19 : Maquette de l’interface principale 29](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053533)

[Figure 20 : Maquette de l’interface de gestion des patients 30](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053534)

[Figure 21 : Maquette de l'interface ajouter patient 30](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053535)

[Figure 22 : Maquette de l'interface de gestion des rendez-vous 31](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053536)

[Figure 23 : Maquette de l'interface ajouter un rendez-vous 31](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053537)

[Figure 24 : Maquette de l'interface ajouter consultation 32](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053538)

[Figure 25 : Maquette de l'interface ajouter ordonnance 32](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053539)

[Figure 26 : Maquette de l'interface de gestion des Médicament 33](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053540)

[Figure 27 : Maquette de l'interface ajouter Médicament 33](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053541)

[Figure 28 : Maquette de l'interface modifier dossier médicale 34](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053542)

[Figure 29 : Structure de notre projet avec MVC 35](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053543)

[Figure 30 : Menu de langue 36](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053544)

[Figure 31 : Fenêtre principale en français 36](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053545)

[Figure 32 : Fenêtre principale en anglais 36](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053546)

[Figure 33 : Classe java avec les annotations 36](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053547)

[Figure 34 : Interface graphique d'authentification 37](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053548)

[Figure 35 : Récupération de mot passe 37](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053549)

[Figure 36 : Interface graphique d'accueil 38](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053550)

[Figure 37 : Exemple de menu d'interface d'accueil 38](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053551)

[Figure 38 : Interface graphique de gestion des patients 39](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053552)

[Figure 39 : Interface graphique d'ajout un patient 39](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053553)

[Figure 40 : Interface graphique de gestion des rendez-vous 40](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053554)

[Figure 41 : Interface graphique d'ajout d'un rendez-vous 40](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053555)

[Figure 42 : Interface graphique de gestion des médicaments 41](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053556)

[Figure 43 : Interface graphique d'ajout un médicament 41](file:///E:\Enseignement%202013-2014\EPI\Zied%20ELLEUCH_PFA\Saif%20Said\Finale\Saifeddine%20said%20PFA.docx#_Toc387053557)

# Liste des tableaux

[Tableau 1 : Dictionnaire de données de classe Personne 8](#_Toc387050297)

[Tableau 2 : Dictionnaire de données de classe Médecin 8](#_Toc387050298)

[Tableau 3 : Dictionnaire de données de classe Secrétaire 8](#_Toc387050299)

[Tableau 4 : Dictionnaire de données de classe parent 8](#_Toc387050300)

[Tableau 5 : Dictionnaire de données de classe Enfant 9](#_Toc387050301)

[Tableau 6 : Dictionnaire de données de classe Compte 9](#_Toc387050302)

[Tableau 7 : Dictionnaire de données de classe Cnam 9](#_Toc387050303)

[Tableau 8 : Dictionnaire de données de classe Profession 9](#_Toc387050304)

[Tableau 9 : Dictionnaire de données de classe Vaccin 10](#_Toc387050305)

[Tableau 10 : Dictionnaire de données de classe Rendez-vous 10](#_Toc387050306)

[Tableau 11 : Dictionnaire de données de classe Dossier Médical 10](#_Toc387050307)

[Tableau 12 : Dictionnaire de données de classe Consultation 11](#_Toc387050308)

[Tableau 13 : Dictionnaire de données de classe Ordonnance 11](#_Toc387050309)

[Tableau 14 : Dictionnaire de données de classe prescription médicale 11](#_Toc387050310)

[Tableau 15 : Dictionnaire de données de classe Médicament 12](#_Toc387050311)

[Tableau 16 : Dictionnaire de données de classe Forme 12](#_Toc387050312)

[Tableau 17 : Dictionnaire de données de classe DCI 12](#_Toc387050313)

# Introduction

**Dans sa pratique, le médecin est confronté de manière répétitive à des problèmes de recueil, récupération, traitement et communication d’informations. Ses capacités ne sont pas suffisantes pour cela il est nécessaire d’informatiser les cabinets médicaux.**

Dans ce cadre, nous sommes appelés à concevoir, développer et mettre en place une application pour la gestion d'un cabinet médical pédiatrie.

**L**e logiciel devrait mettre l'organisation et l'automatisation de la gestion d'un cabinet médical, afin d'augmenter la fiabilité, l'efficacité de l'effort humain et faciliter les tâches pénibles au sein du cabinet.

Notre application comprendra les fonctionnalités suivantes :

* + - * Gestion des patients;
      * Gestion des rendez-vous;
      * Gestion de la consultation;
      * Gestion de dossier médicale;
      * Gestion de médicament.

**Le présent projet s'articule autour de trois chapitres qui sont présentés comme suit :**

**Le premier chapitre est consacré à l’étude détaillée du projet qui contient essentiellement les modèles conceptuels (Diagramme de classe, diagrammes de cas d’utilisation, diagrammes de séquence).**

**Le deuxième chapitre définit les technologies choisi avec la présentation des quelques maquettes d'interfaces utilisateurs.**

**Enfin, le troisième chapitre décrit la partie réalisation du projet. On présente l’environnement de travail, quelques captures d’écrans des interfaces utilisateurs graphiques avec leurs descriptions et des exemples des états récapitulatifs à imprimer.**

# Conception

## Introduction

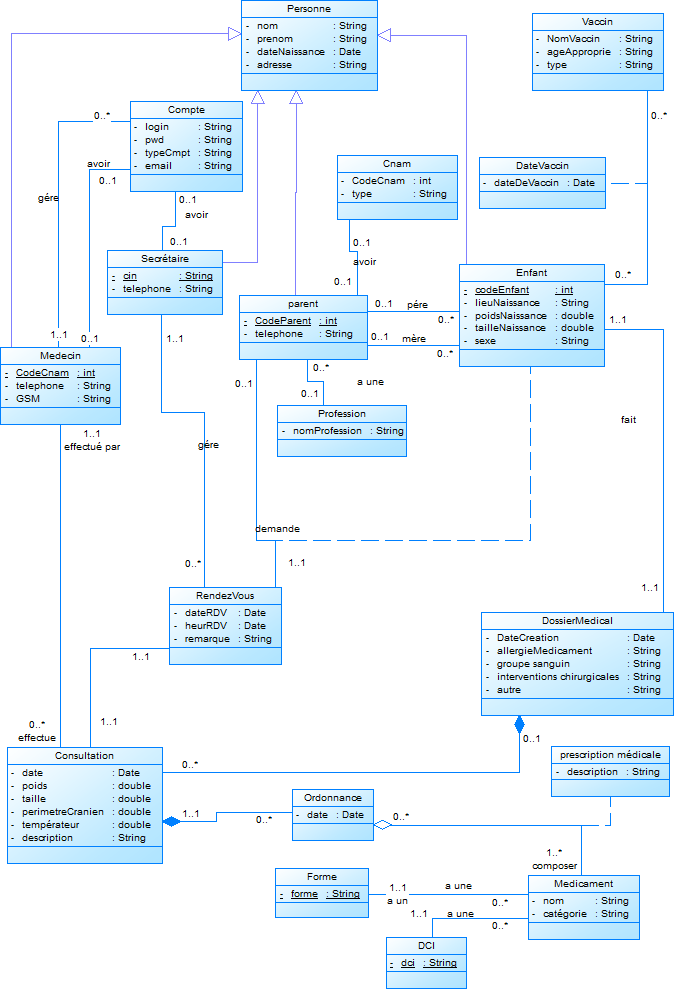
Dans ce chapitre nous nous intéressons à la conception statique, nous commençons par présenter le diagramme de classes, ensuite nous présentons les diagrammes de cas d'utilisation et enfin les diagrammes de séquences.

## Diagramme de Classe

### Description générale

Le diagramme de classes représente les classes et les interfaces du système ainsi que les différentes relations entre elles. La figure ci-dessous représente le diagramme de classe de notre application.

Figure : Diagramme de classe



### Dictionnaire de données

Le tableau 1 présente la liste des propriétés utilisées dans la classe personne. Chaque propriété représente une information que l'on souhaite stoker.

Tableau : Dictionnaire de données de classe Personne

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| nom | | Nom de personne | Texte |
| prénom | | Prénom de personne | Texte |
| date Naissance | | Date de naissance | Date |
| adresse | Adresse de personne | | Texte |

Le tableau ci-dessous présente les champs utilisés dans la classe médecin et le type de chaque champ.

Tableau : Dictionnaire de données de classe Médecin

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation** | **Type de données du champ** |
| Code Nam | Code Cam de médecin | numéro |
| téléphone | Numéro téléphone | Texte |
| GSM | Numéro GSM de médecin | Texte |

Le tableau suivant représente l'ensemble des propriétés utilisées dans la classe secrétaire et le type de chaque propriété.

Tableau : Dictionnaire de données de classe Secrétaire

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation** | **Type de données du champ** |
| ciné | carte d'identité nationale | Texte |
| telephone | Numéro de téléphone | Texte |

Le tableau suivante représente l'ensemble des propriétés utilisées dans la classe parent et le type de chaque propriétés.

Tableau : Dictionnaire de données de classe parent

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation** | **Type de données du champ** |
| Code Parent | Code parent | numéro |
| téléphone | Numéro de téléphone | Texte |

Le tableau 5présente la liste des propriétés utilisées dans la classe enfant et le type de données à stoker.

Tableau : Dictionnaire de données de classe Enfant

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| code Enfant | | Code de l'enfant | numéro |
| lieu Naissance | | Lieu de naissance | Texte |
| poids Naissance | | Poids de naissance | double |
| taille Naissance | Taille de naissance | | double |
| sexe | Sexe de l'enfant | | Texte |

Le tableau 6présente la liste des propriétés utilisées dans la classe compte et le type de données à stoker.

Tableau : Dictionnaire de données de classe Compte

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| login | | login | Texte |
| pwd | | Mot de passe | Texte |
| typeCmpt | | Type de compte | Texte |
| email | email | | Texte |

Le tableau ci-dessous présente les champs utilisés dans la classe Cnam et le type de chaque champ à stocker.

Tableau : Dictionnaire de données de classe Cnam

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation** | **Type de données du champ** |
| CodeCnam | Code de Cnam | numéro |
| type | Type (CNSS/CNRPS) | Texte |

Le tableau suivant représente les champs utilisés dans la classe profession et le type de chaque champ.

Tableau : Dictionnaire de données de classe Profession

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation** | **Type de données du champ** |
| nomProfession | Nom de Profession | Texte |

Le tableau ci-dessous présent les champs utilisés dans la classe vaccin et le type de chaque champ a stocké.

Tableau : Dictionnaire de données de classe Vaccin

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| NomVaccin | | Nom de Vaccin | Texte |
| ageApproprie | | âge approprié de vaccin | Texte |
| dateDeVaccin | | Date De Vaccin | Date |
| type | Type (optionnel/obligatoire) | | Texte |

Le tableau suivant représente l'ensemble des propriétés utilisées dans la classe rendez-vous et le type de chaque propriété.

Tableau : Dictionnaire de données de classe Rendez-vous

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| numRDV | | Numéro RDV | numéro |
| dateRDV | | Date de RDV | Date |
| heurRDV | | Heur de RDV | Date |
| remarque | remarque | | Texte |

Le tableau ci-dessous représente l'ensemble des propriétés utilisées dans la classe dossier médical et le type de chaque propriété.

Tableau : Dictionnaire de données de classe Dossier Médical

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| numDossier | | Numéro de Dossier | numéro |
| dateCreation | | Date de Création | Date |
| allergie | | Allergie médicament | Texte |
| groupeSanguin | groupe sanguin | | Texte |
| interventionsChirurgicales | Interventions chirurgicales | | Texte |
| autre | autre | | Texte |

Le tableau suivant représente l'ensemble des propriétés utilisées dans la classe consultation et le type de chaque propriété.

Tableau : Dictionnaire de données de classe Consultation

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| numConsult | | Numéro Consultation | numéro |
| date | | Date de Consultation | Date |
| poids | | poids pendant consultation | float |
| taille | Taille pendant consultation | | double |
| perimetreCranien | Périmètre Crânien pendant consultation | | double |
| températeur | Température pendant consultation | | double |
| description | description | | Texte |

Le tableau ci-dessous présente les champs utilisés dans la classe ordonnance et le type de chaque champ.

Tableau : Dictionnaire de données de classe Ordonnance

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation** | **Type de données du champ** |
| numOrdonn | Numéro Ordonnance | numéro |
| date | Date Ordonnance | Date |

Le tableau ci-dessous présente les champs utilisés dans la classe prescription médicale et le type de chaque champ.

Tableau : Dictionnaire de données de classe prescription médicale

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation** | **Type de données du champ** |
| description | Description prescription médicale | Texte |

Le tableau ci-dessous représente l'ensemble des propriétés utilisées dans la classe médicament et le type de chaque propriété.

Tableau : Dictionnaire de données de classe Médicament

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du champ** | | **Désignation** | **Type de données du champ** |
| numMedica | | Numéro Médicament | numéro |
| nom | | Nom Médicament | Texte |
| Forme | | Forme Médicament | Texte |
| catégorie | Catégorie Médicament | | Texte |

Le tableau ci-dessous représente l'ensemble des propriétés utilisées dans la classe forme et le type de chaque propriétés.

Tableau : Dictionnaire de données de classe Forme

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation du champ** | **Type de données du champ** |
| forme | Forme Médicament | Texte |

Le tableau ci-dessous représente l'ensemble des propriétés utilisées dans la classe dci et le type de chaque propriété.

Tableau : Dictionnaire de données de classe DCI

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Désignation du champ** | **Type de données du champ** |
| dci | Dénomination commune international de médicament | Texte |

### Diagramme de cas d'utilisation

**Les cas d’utilisation décrivent, sous la forme d’actions et de réactions, le comportement d’un système de point de vue « Utilisateur ». Ils permettent de définir les exigences du système ainsi que ses relations avec l’environnement.**

#### Diagramme de cas d’utilisation générale

**Dans la Figure ci-dessous, le rôle de la secrétaire est de gérer la gestion de patient et la gestion de rendez-vous; et le rôle de médecin est de gérer la gestion de la consultation médicale, gestion des utilisateurs et gestion du médicament.**

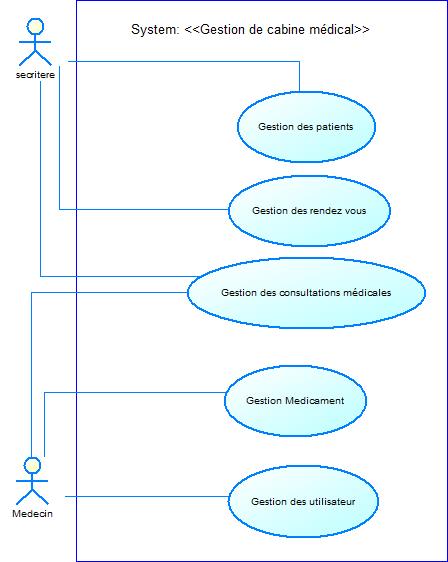
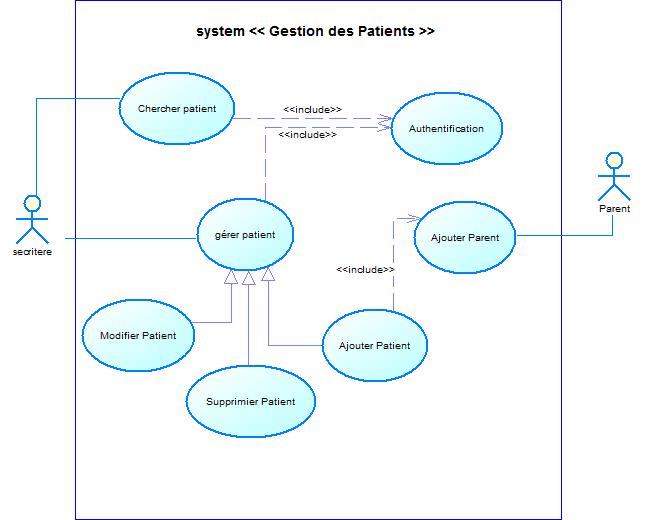


Figure : Diagramme de cas d’utilisation générale de la gestion de cabinet de médicine

#### Diagramme des cas d'utilisation relatif à la gestion des patients

**Dans la figure suivante, la secrétaire peut gérer la gestion de patient après son identification**

Figure : Diagramme des cas d'utilisation relatif à la gestion des employés par patients



#### Diagramme des cas d'utilisation relatif à la gestion des Consultations

Dans la Figure ci-dessous, le médecin peut gérer la gestion des Consultation (suppression, modification, et ajout) après qu'il s'identifier.

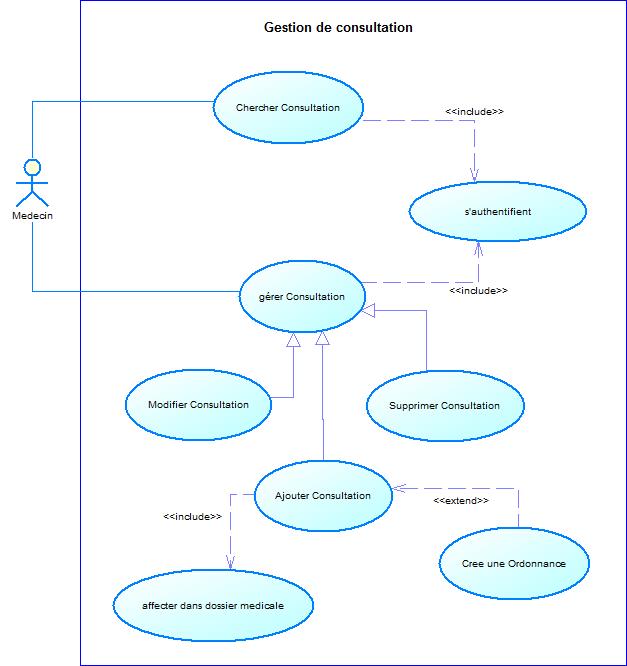
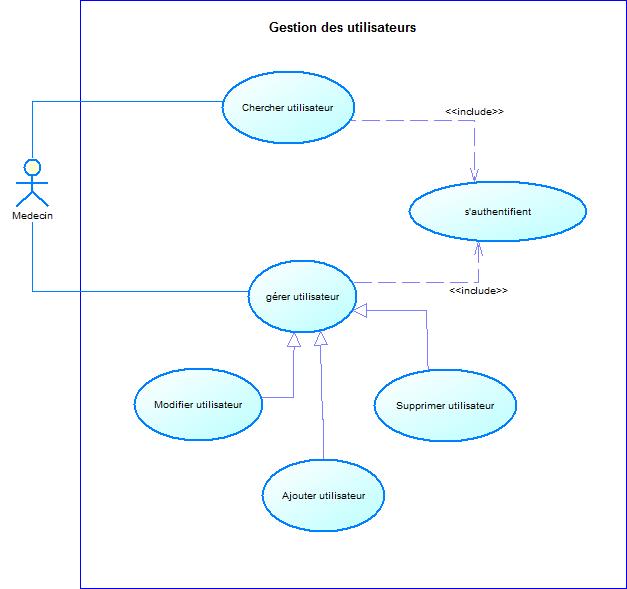


Figure : Diagramme des cas d'utilisation relatif à la gestion des médicaments

#### Diagramme des cas d'utilisation relatif à la gestion des utilisateurs

Dans la Figure suivante, le médecin peut gérer les utilisateurs de système (suppression, modification, et ajout) après qu'il s'identifier.

Figure : Diagramme des cas d'utilisation relatif à la gestion des utilisateurs



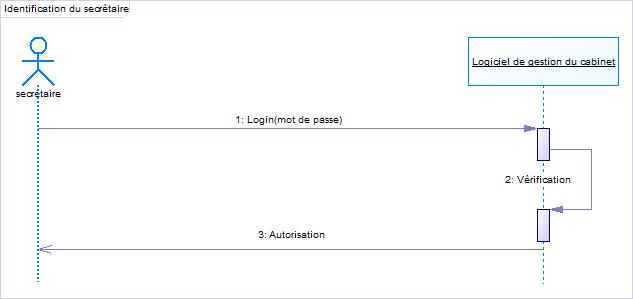
### Diagramme de Séquence

#### Gestion des patients

* + - ***Identification :***

Dans la Figure suivante la secrétaire se connecte au système et introduit son nom d'utilisateur et mot de passe. Le système vérifie l’identité de la secrétaire et autorise la connexion.

Figure : Diagramme de séquence du scénario nominal du cas d’utilisation s’authentifier

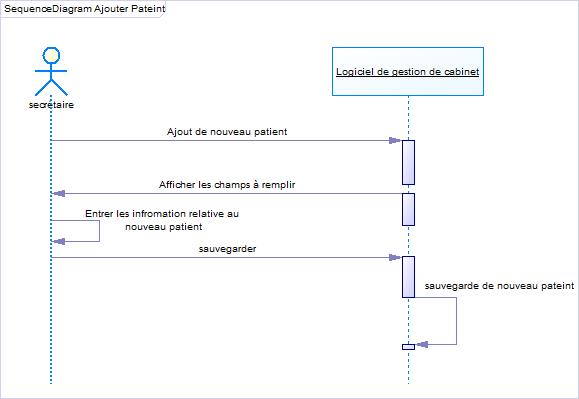


* + - ***Ajouter un patient***

Dans la figure ci-dessous, le parent fournit à la secrétaire les données relatives à son enfant et leurs données personnelles.

La secrétaire consulte le système et saisit les données relatives au nouveau patient; et le système enregistre les informations.

Figure : Diagramme de séquence du scénario nominal du cas d’utilisation Ajouter patient

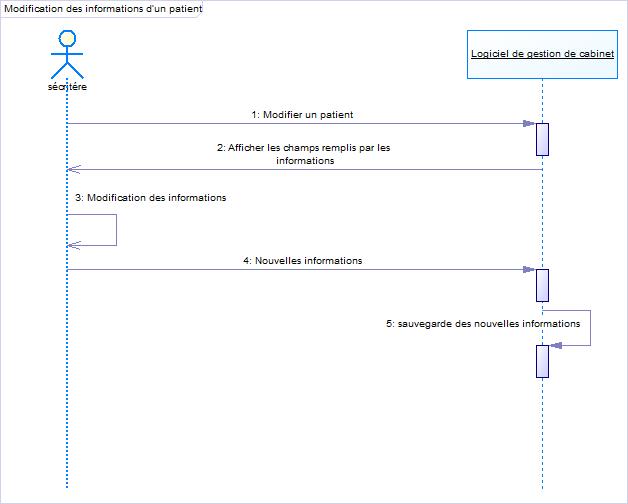


* + - ***Modification des informations d’un patient***

Dans la figure ci-dessous, la secrétaire sélectionne le patient à modifier et le système affiche tous les identifiants du patient : **nom, prénom, date de naissance, numéro de téléphone**, etc.

La secrétaire modifie les informations et le système les enregistre.

Figure : Diagramme de séquence du scénario nominal du cas d’utilisation modifier un patient



* + - **Archivé les informations d'un patient**

Dans la figure ci-dessous, la secrétaire sélectionne un patient et demande de archivé ses informations, le système demande la validation, après la validation de la secrétaire le système enregistre la modification.

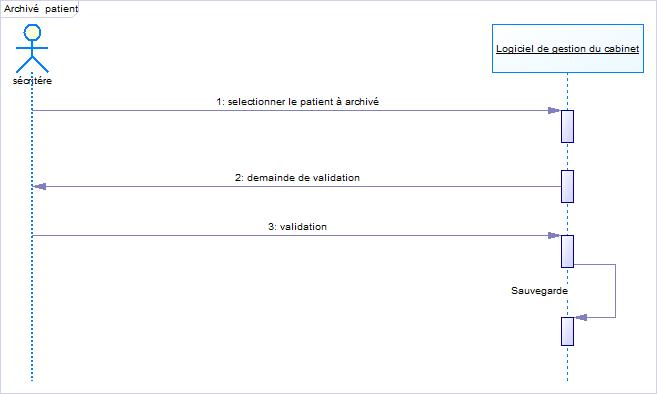


Figure : Diagramme de séquence du scénario nominal du cas d’utilisation archivé patient

* + - ***Rechercher des informations***

Dans la figure ci-dessous, la secrétaire entre les critères (**nom, prénom, etc.**) du patient à rechercher, en cliquant sur le bouton « rechercher », le système lui affiche tous les identifiants de ce patient.

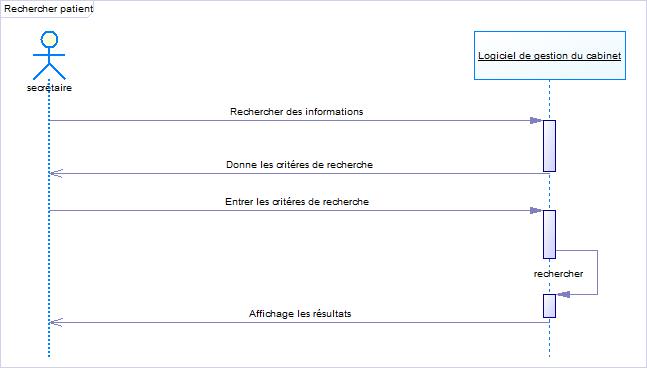


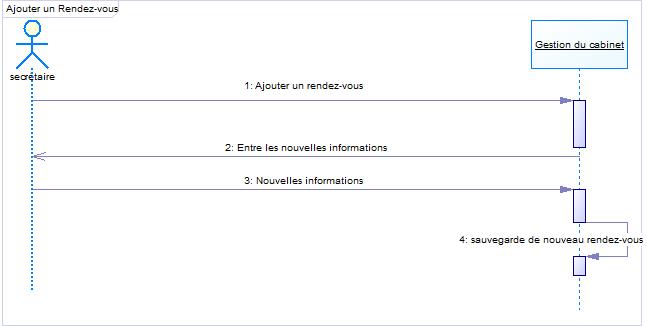
Figure : Diagramme de séquence du scénario nominal du cas d’utilisation chercher patient

#### Gestion des rendez-vous

* + - ***Ajout d’un rendez-vous :***

Dans la figure suivante, la secrétaire demande d’ajouter un nouveau rendez-vous identifié par sa date, le nom, le prénom d’un patient ainsi que l'heure de rendez-vous et le système ajoute ce rendez-vous à la liste.

Figure : Diagramme de séquence du scénario nominal du cas d’utilisation ajouter un rendez-vous

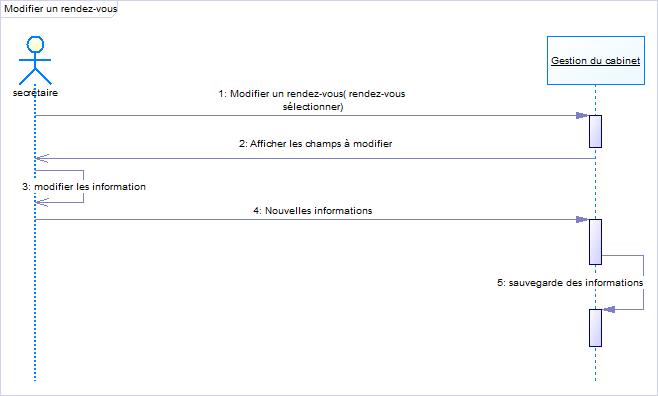


* + - ***Modification des informations d’un rendez-vous :***

Dans la figure ci-dessous, la secrétaire sélectionne le rendez-vous à modifier et le système affiche tous les identifiants du rendez-vous.

La secrétaire modifie les informations et le système les enregistre.

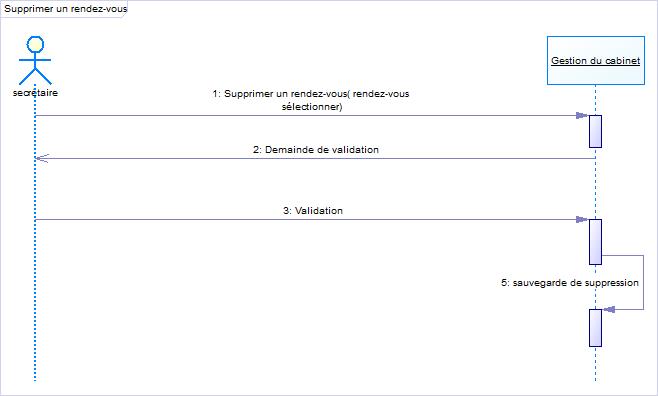
Figure : Diagramme de séquence du scénario nominal du cas d’utilisation modifier rendez-vous



* + - **Supprimer les informations d’un rendez-vous**

Dans la figure suivante, la secrétaire sélectionne un rendez-vous et demande de supprimer ses informations, le système demande la validation, après la validation de la secrétaire le système enregistre la suppression.

Figure : Diagramme de séquence du scénario nominal du cas d’utilisation supprimer un rendez-vous

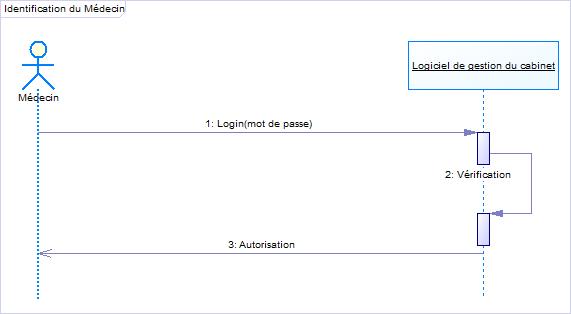


#### Gestion des consultations

* + - **Identification :**

Dans la figure suivante, le médecin se connecte au système à travers son login et mot de passe, le système vérifie l’identité de médecin et autorise la connexion.

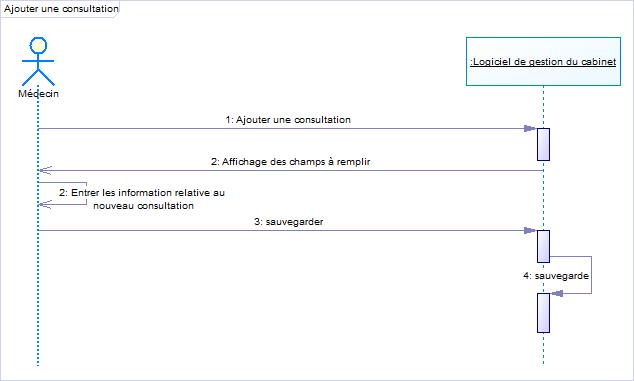
Figure : Diagramme de séquence du scénario nominal du cas d’utilisation s’authentifier



* + - **Ajouter une consultation médicale :**

Dans la figure suivante, le médecin demande d'ajouter une consultation, le système affiche les champs à remplir. Le médecin saisie les informations relative à la nouvelle consultation, puis il demande d’enregistrer ses informations et le système les enregistre.

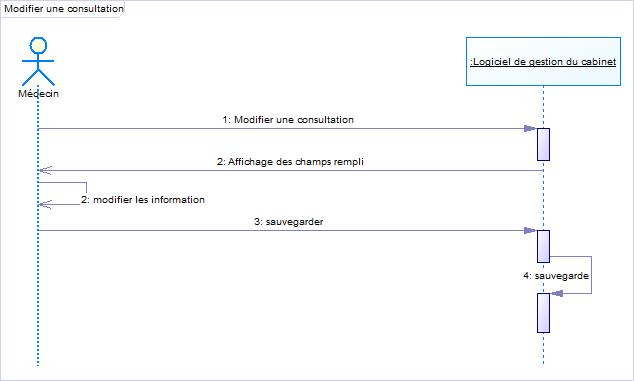
Figure : Diagramme de séquence du scénario nominal du cas d’utilisation ajouter consultation



* + - **Modification des informations d’une consultation médicale :**

Dans la figure ci-dessous, le médecin demande de modifier les informations concernant une consultation médicale identifiée par un numéro, le système affiche les champs contenant les informations. Le médecin change les informations d’une consultation et le système enregistre la modification.

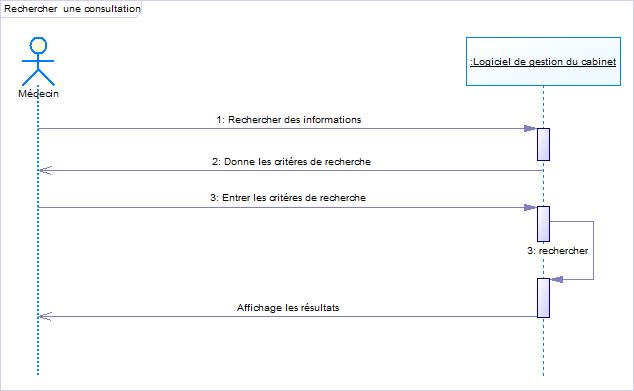
Figure : Diagramme de séquence du scénario nominal du cas d’utilisation modifier une consultation



* + - **Recherche des informations sur la consultation :**

Dans la figure ci-dessous, le médecin choisit les critères de recherche (le numéro de consultation); et le système lui affiche toutes les informations (diagnostic, traitement, date consultation).

Figure : Diagramme de séquence du scénario nominal du cas d’utilisation chercher une consultation



# Choix techniques

## Introduction

Dans ce chapitre, nous décrivons les choix techniques ainsi que la technologie adoptée. Ensuite nous présentons quelques maquettes de capture d’écran relatives à notre application.

## Choix techniques

### Choix du modèle MVC

Le Modèle-Vue-Contrôleur (en abrégé MVC, de l'anglais Model-View-Controller) est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle. Ce paradigme divise l'IHM en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface.

### Eclipse

Eclipse[[1]](#footnote-2) est un IDE, (environnement de développement intégré), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

### Framework Hibernate

Hibernate [1] est un Framework de mapping Objet/Relationnel ou de **persistance objet des données**, recommandé pour assurer la rapidité, l’évolutivité et la maintenabilité des développements.

Hibernate est une surcouche de JDBC qui lui ajoute une dimension objet. Le terme mapping objet/relationnel (ORM) décrit la technique consistant à faire le lien entre la représentation objet des données et sa représentation relationnelle, basée sur un schéma SQL.

Hibernate permet de lier/mapper un objet défini en Java avec une table dans une base de données, via un fichier déclaratif de mapping. Le système peut s’occuper de la création des tables en fonction des fichiers de configuration et mettre aussi à jour les tables si nécessaire lors d’un changement dans l’un des fichiers de configuration.

### Log4j

Log4J simplifie les gestions des logs et le débogage des applications Java en fournissant des classes et des méthodes pour l’enregistrement de ces informations.

Les fichiers journaux d’une application représentent la mémoire d’une application, un historique permanent de la vie de celle-ci, il est donc important de correctement enregistrer ces messages.

Ces différents types d’affichage des messages peuvent être configurés facilement au runtime de log4j par fichier XML ou par fichier de propriétés, donc de façon totalement externe au code.

### JasperReports

JasperReports[[2]](#footnote-3) est une librairie Java open source permet la création des états, elle permet la représentation de données sous forme textuelle, mais aussi la génération de graphiques divers. JasperReports propose l’export de ces rapports dans de nombreux formats :

* PDF;
* XLS (Excel);
* RTF;
* ODF;
* HTML;
* XML
* CSV;
* Texte brut.

Les données peuvent provenir de sources diverses :

* JDBC;
* JavaBeans;
* EJBQL;
* XML;
* Hibernate;
* Fichiers texte;
* CSV.

### PostgreSQL

PostgreSQL [2] est un système de gestion de bases de données ([SGBD](http://doc.ubuntu-fr.org/sgbd)) très performant dont les performances sont comparables à Oracle 9. Il propose de très nombreuses fonctionnalités, tout en respectant les standards SQL et en partie la norme SQL2003. En outre, il intègre plusieurs langages embarqués (Perl, Python, Java) depuis de nombreuses années. L'une des principales qualités de PostgreSQL est d'être un logiciel Open source.

**Caractéristique de PostgreSQL** :

PostgreSQL[[3]](#footnote-4) possède de nombreuses caractéristiques en faisant un SGBD robuste et puissant digne des SGBD commerciaux :

* des interfaces graphiques pour gérer les tables; des bibliothèques pour de nombreux langages (appelés frontaux) afin d'accéder aux enregistrements à partir de programmes écrits en : [Java](http://www.commentcamarche.net/contents/java/javaintro.php3), [langage C](http://www.commentcamarche.net/contents/c/cintro.php3)/[C++](http://www.commentcamarche.net/contents/cpp/cppintro.php3), [Perl](http://www.commentcamarche.net/contents/perl/perlintro.php3), etc. ;
* une API [ODBC](http://www.commentcamarche.net/contents/bdd/odbc.php3) permettant à n'importe quelle application supportant ce type d'interface d'accéder à des bases de données de type PostgreSQL;

PostgreSQL fonctionne selon une architecture [client/serveur](http://www.commentcamarche.net/contents/initiation/client.php3), il est ainsi constitué d'une partie serveur, c'est-à-dire une application fonctionnant sur la machine hébergeant la base de données (le serveur de bases de données) capable de traiter les requêtes des clients. Il s'agit dans le cas de PostgreSQL d'un programme résident en mémoire appelé postmaster d'une partie client devant être installée sur toutes les machines nécessitant d'accéder au serveur de base de données (un client peut éventuellement fonctionner sur le serveur lui-même) Les clients (les machines sur lesquelles le client PostgreSQL est installé) peuvent interroger le serveur de bases de données à l'aide de [requêtes SQL](http://www.commentcamarche.net/contents/sql/sqlintro.php3).

### JUnit

JUnit[[4]](#footnote-5) est un framework open source pour le développement et l'exécution de tests unitaires automatisables. Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires de non régression.

JUnit propose :

* Un framework pour le développement des tests unitaires reposant sur des assertions qui testent les résultats attendus;
* Des applications pour permettre l'exécution des tests et afficher les résultats.

Le but est d'automatiser les tests. Ceux-ci sont exprimés dans des classes sous la forme de cas de tests avec leurs résultats attendus. JUnit exécute ces tests et les comparent avec ces résultats.

Cela permet de séparer le code de la classe, du code qui permet de la tester. Souvent pour tester une classe, il est facile de créer une méthode main () qui va contenir les traitements de tests. L'inconvénient est que ce code "superflu" est inclus dans la classe et son exécution se fait manuellement.

### Subversion (SVN)

SVN[[5]](#footnote-6) [3] est un logiciel de gestion de versions permettant de configurer, de stocker des informations pour une ou plusieurs ressources informatiques et de récupérer toutes les versions intermédiaires des ressources, ainsi que les différences entre les versions.

Un logiciel de gestion de versions agit sur une arborescence de fichiers afin de conserver toutes les versions des fichiers, ainsi que les différences entre les fichiers. Ce système permet par exemple de mutualiser un développement. Un groupe de développeurs autour d'un même développement se servira de l'outil pour stocker toute évolution du code source.

SVN fonctionne sur le mode client-serveur avec :

* un serveur informatique et unique où se situent :
  + les fichiers constituant la référence (le «dépôt » ou « référentiel », ou « repository » en anglais),
  + un logiciel serveur Subversion tournant en « tâche de fond » ;
* des postes clients sur lesquels se trouvent :
  + - les fichiers recopiés depuis le serveur, éventuellement modifiés localement depuis;
    - un logiciel client sous forme d'exécutable (exemple : SmartSVN) ou de plug-in (exemple : TortoiseSVN, Eclipse subversive) permettant la synchronisation, manuelle et/ou automatisée, entre chaque client et le serveur de référence.

### Serveur SVN utiliser dans le projet

Dans notre projet nous avons utilisé comme serveur SVN assembla, c'est un serveur gratuit pour héberger notre projet, c'est un système de subversion privé gratuit permet de garder notre propriété intellectuelle à l'abri des regards indiscrets; seulement ceux que nous inviter à être un membre de notre espace peut voir notre code.

Assembla nous propose, gratuitement :

* hébergement gratuite (subversion privé);
* un espace pour vos sources et fichiers de projet de 200 Mo;
* un serveur SVN/CVS;
* un bugtracker basé sur Trac;
* des outils de collaborations : un wiki, un chat;
* des outils de management de l’équipe et du projet.

### Balsamiq Mockups

C'est un outil pour faire des maquettes rapides d'écrans applicatifs, il permet de :

* créer une maquette d'interface logicielle.
* créer une maquette d'interface web.
* créer une maquette d'interface application mobile.

## Maquette des écrans

### Maquette de l'interface d’authentification

La figure ci-dessous présente la maquette de l'interface d'authentification proposé qui permet à l'utilisateur de se connecter au système de gestion de cabinet.

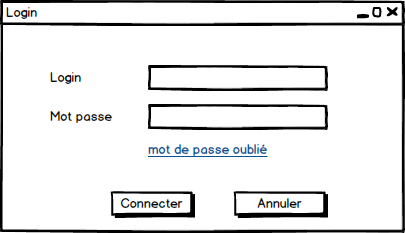


Figure : Maquette de l'interface d’authentification

### Maquette de l'interfaces principales

La figure suivante représente présente la maquette de l'interfaces principales qui offre à l'utilisateur d'accéder a toutes les gestions et les services de l'application.

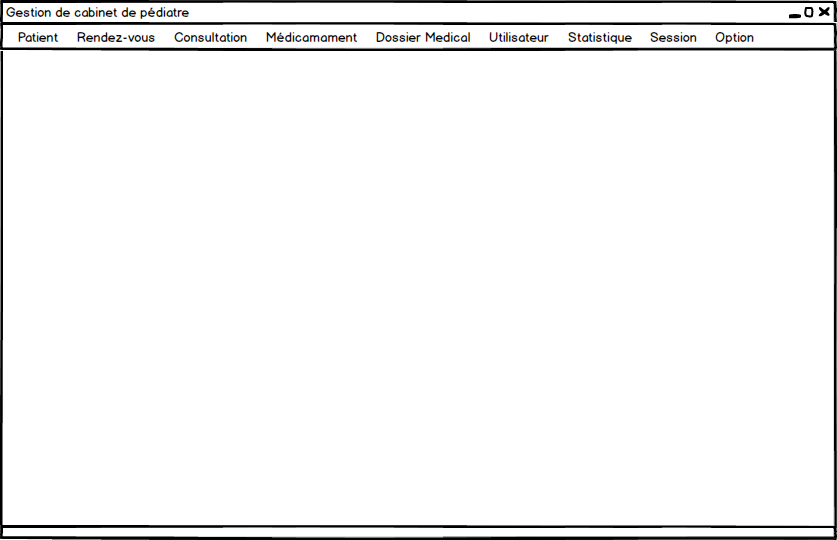


Figure : Maquette de l’interface principale

### Maquette de l'interface gestion patient

La figure ci-dessous présente la maquette de l'interface de gestion des patients qui permet à l'utilisateur de gérer les patients.

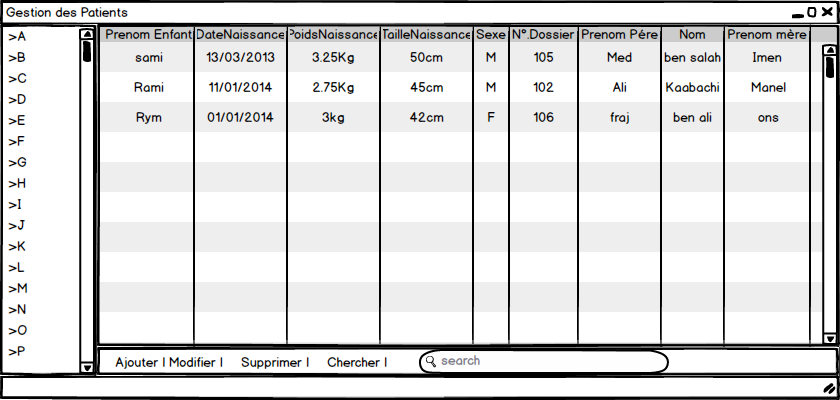


Figure : Maquette de l’interface de gestion des patients

### Maquette de l'interface ajouter patient

La figure ci-dessous présente la maquette de l'interface ajouter un patients qui permet à la secrétaire de gérer les patients.

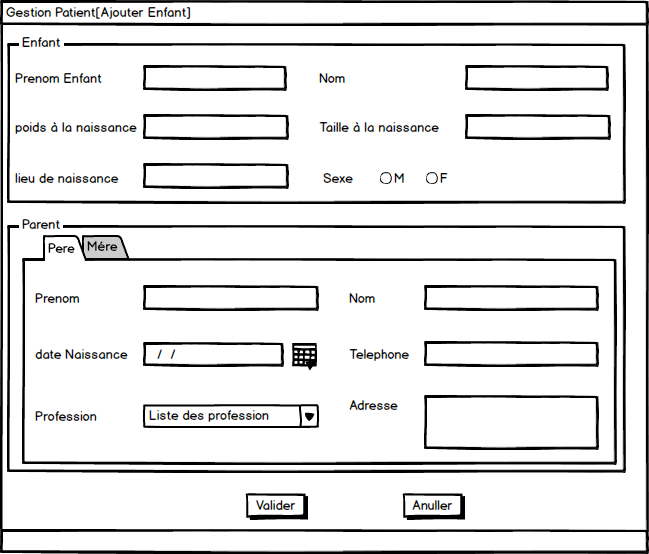
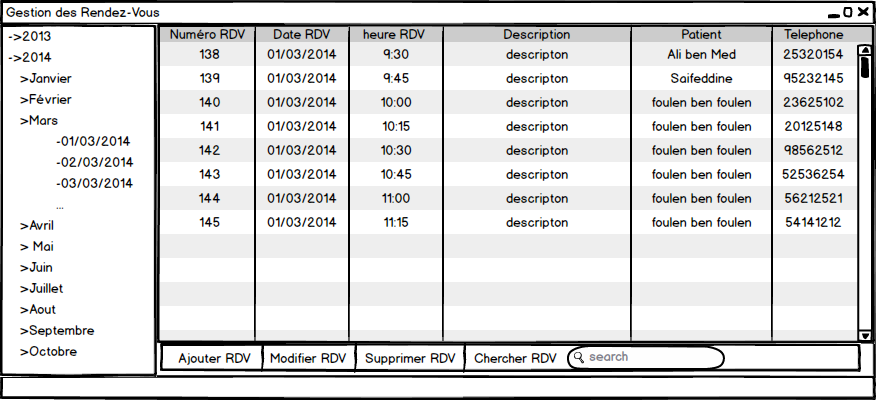


Figure : Maquette de l'interface ajouter patient

### Maquette de l'interface de gestion des rendez-vous

La figure ci-dessous présente la maquette de l'interface de gestion des rendez-vous qui permet à la secrétaire de gérer les rendez-vous.

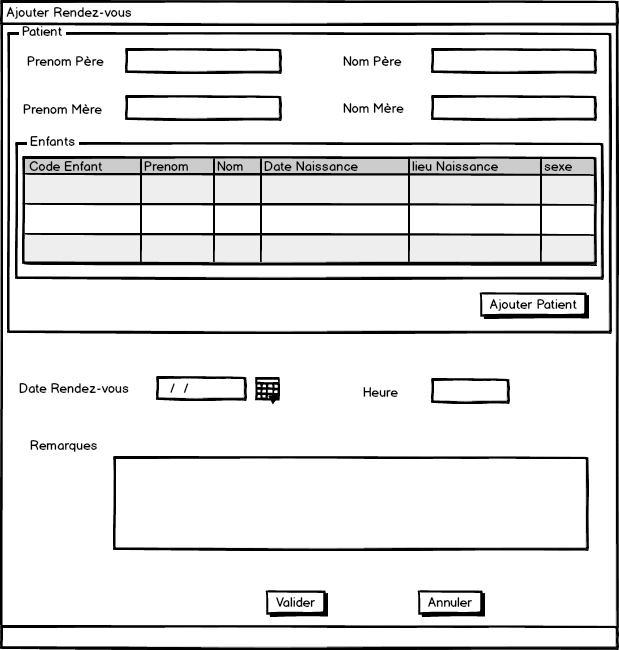
Figure : Maquette de l'interface de gestion des rendez-vous



### Maquette de l'interface ajouter un rendez-vous

La figure ci-dessous présente la maquette de l'interface ajouter un rendez vous qui permet à la secrétaire d'ajouter un rendez-vous.

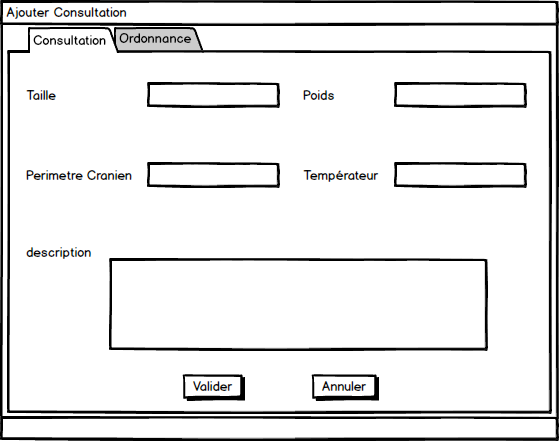
Figure : Maquette de l'interface ajouter un rendez-vous



### Maquette de l'interface ajouter consultation

La figure ci-dessous présente la maquette de l'interface ajoutée consultation qui permet au médecin d'ajouter une consultation.

Figure : Maquette de l'interface ajouter consultation



### Maquette de l'interface ajouter ordonnance

La figure ci-dessous présente la maquette de l'interface ajoutée ordonnance qui permet au médecin d'ajouter une ordonnance à une consultation.

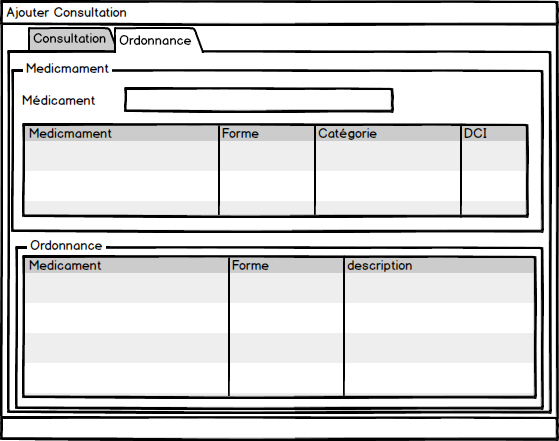


Figure : Maquette de l'interface ajouter ordonnance

### Maquette de l'interface de gestion des Médicament

La figure ci-dessous présente la maquette de l'interface de gestion des médicaments qui permet au médecin la gestion des médicaments.

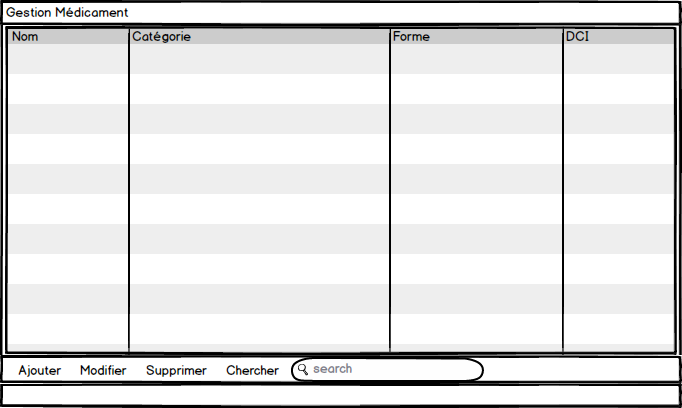
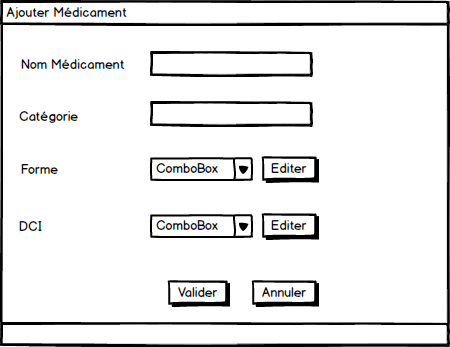


Figure : Maquette de l'interface de gestion des Médicament

### Maquette de l'interface ajouter Médicament

La figure ci-dessous présente la maquette de l'interface ajoutée médicament qui permet au médecin d'ajouter un médicament.

Figure : Maquette de l'interface ajouter Médicament



### Maquette de l'interface modifier dossier médicale

La figure ci-dessous présente la maquette de l'interface modifiée dossier médicale qui permet au médecin de modifier un dossier médical d'un patient.

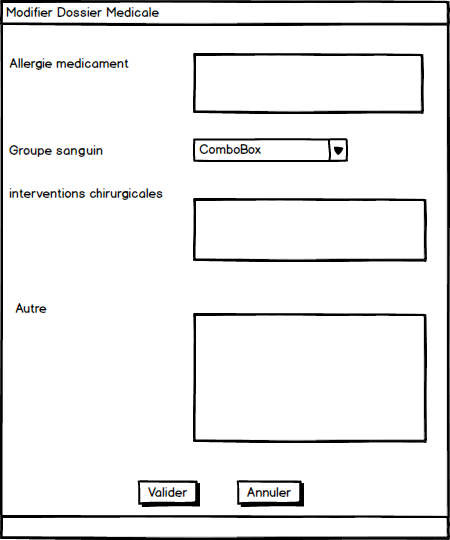


Figure : Maquette de l'interface modifier dossier médicale

# Réalisation

## Introduction

Après avoir défini ce que nous entendons par analyse et conception, nous décrivons la réalisation qui est la concrétisation des réflexions conceptuelles élaboré.

La phase de « la réalisation et le développement » consiste à construire et à documenter les interfaces et les méthodes.

## Implémentation de l'architecteur Modèle-Vue-Contrôleur (MVC)

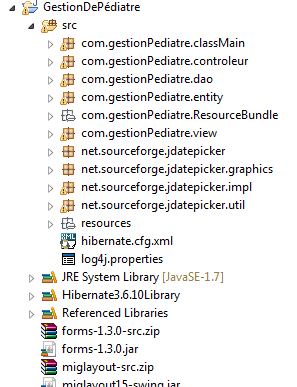
Nous avons découpé notre codes en trois couches : modèle, vue et contrôleur, pour chaque partie nous avons crée un package.

Dans le model nous avons développé les fonctions nécessaire pour chaque vue.

Dans la vue nous avons construire nos interface IHM avec l'implémentation des accesseurs et mutateurs.

Dans le contrôleur nous avons établi le lien entre la vue et le modèle lorsqu'une action utilisateur est intervenue sur la vue et aussi nous avons l'utiliser pour contrôler les données saisir par l'utilisateur.

Figure : Structure de notre projet avec MVC



## L’internationalisation de l'application (I18N)

Nous avons essayé développez à développer une application multilingue pour que l'utilisateur puise modifier la langue de l'application, on simple clic.

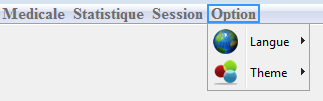


Figure : Menu de langue

Lors de son démarrage l'application est capable de détecter la langue par défaut utiliser par l'ordinateur est de changer automatiquement sa langue.

Figure : Fenêtre principale en français

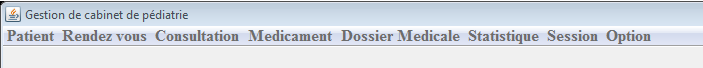
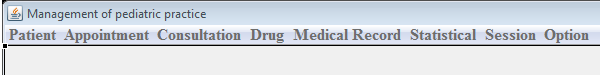


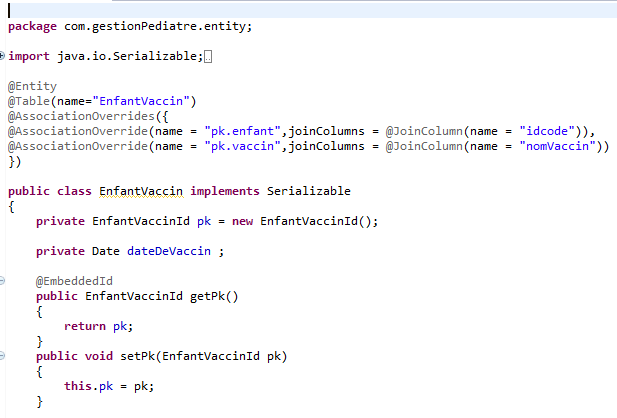
Figure : Fenêtre principale en anglais



## Création de base de données avec Hibernate

La base de données a été générée par le framewok hibernate. Nous avons configuré d’abord le fichier « hibernate.cfg.xml » (figure) qui permet de définir les paramètres de connexion à notre base de données PostgreSQL (nom de driver, nom ,……). Ensuite, nous avons introduire des annotations dans les classes java afin de définir les tables à construire, les attribues, et les relations entre les tables et aussi les classes d'association.

Figure : Classe java avec les annotations

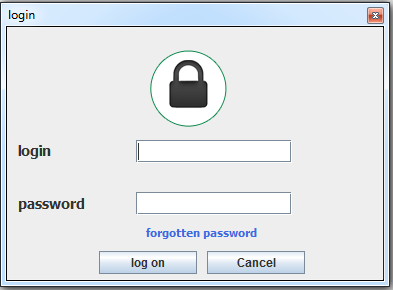


## Présentation des interfaces graphique

### Interface graphique d'authentification

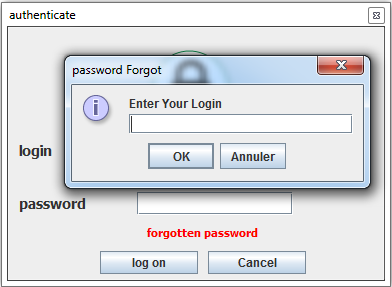
Cette fenêtre (Figure 29), apparaît au moment de l’appel de l’application, pour interdire tout accès non permis. Elle comporte deux champs, le premier est réservé pour l’identification de l’utilisateur et le deuxième pour le mot de passe.

Figure : Interface graphique d'authentification



En cas où l'utilisateur a oublié son mot de passe il peut le récupérer on cliquant sur mot de passe oublié, l'application demande son nom d’utilisateur (figure 30), ensuite le mot de passe sera envoie à l'adresse mail de l'administrateur (médecin).

Figure : Récupération de mot passe



### Interface graphique d'accueil

Cette interface (figure 36) représente la fenêtre principale de l’application, elle apparaît après la validation du nom d’utilisateur et de son mot de passe. Elle permet l’accès à la fenêtre de gestion des patients, ainsi qu’à la fenêtre de gestion de rendez-vous, de gestion des consultations, de gestion de médicament, de gestion des dossiers médical, de gestion des utilisateurs, menu des statistique, menu des sessions et enfin aux menu d'option.

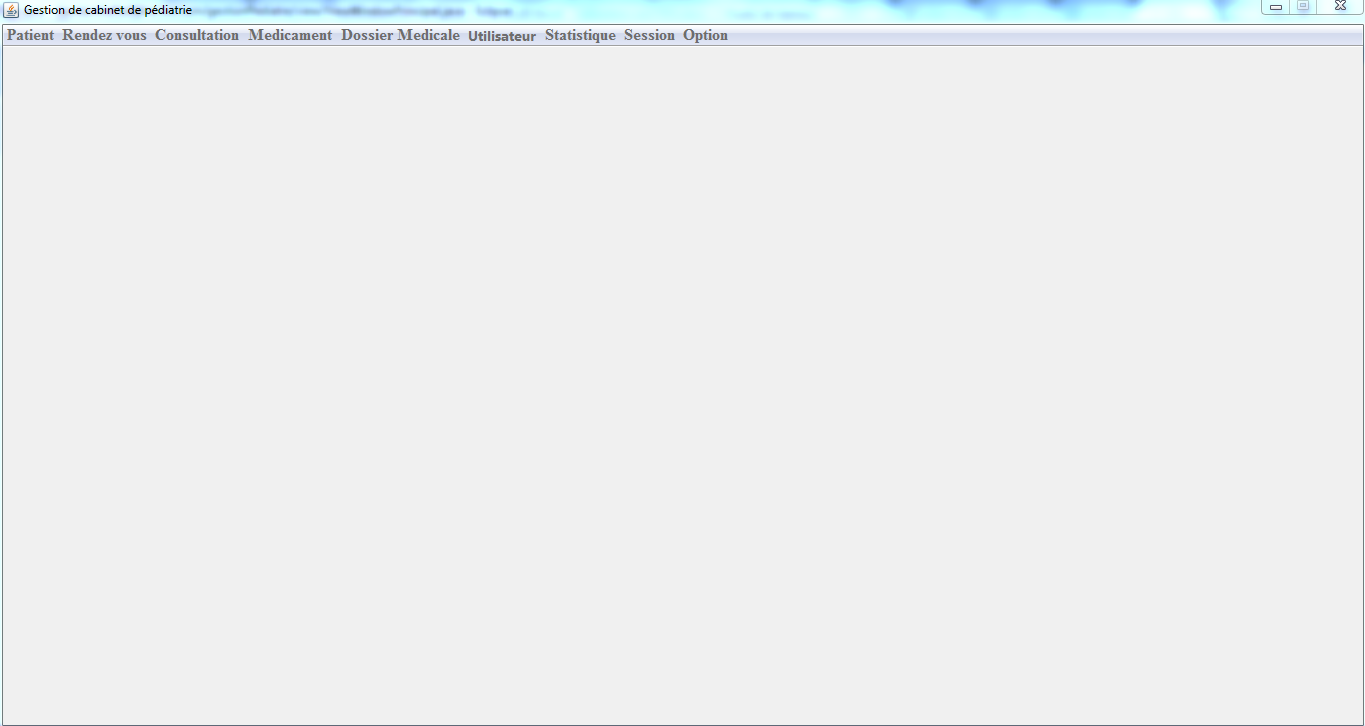
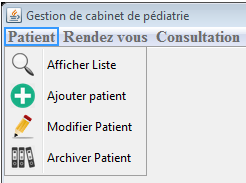


Figure : Interface graphique d'accueil

Exemple de menu de l'interface d'accueil (figure 37).

Figure : Exemple de menu d'interface d'accueil



### Interface graphique de Gestion des patients

Dans l'interface ci-dessous la secrétaire pourra consulter la liste des patient ainsi les gérer, cette interface est composer de trois zone principale :

* la zone numéroté 1 dans la figure 38, permet à la secrétaire de filtré la liste des patients selon l'alphabet sélectionner.
* la zone numéroté 2 permet de consulter tous les patients enregistré dans la base de données.
* la zone numéroté 3 permet d'ajouter, modifier et de supprimer un patient.

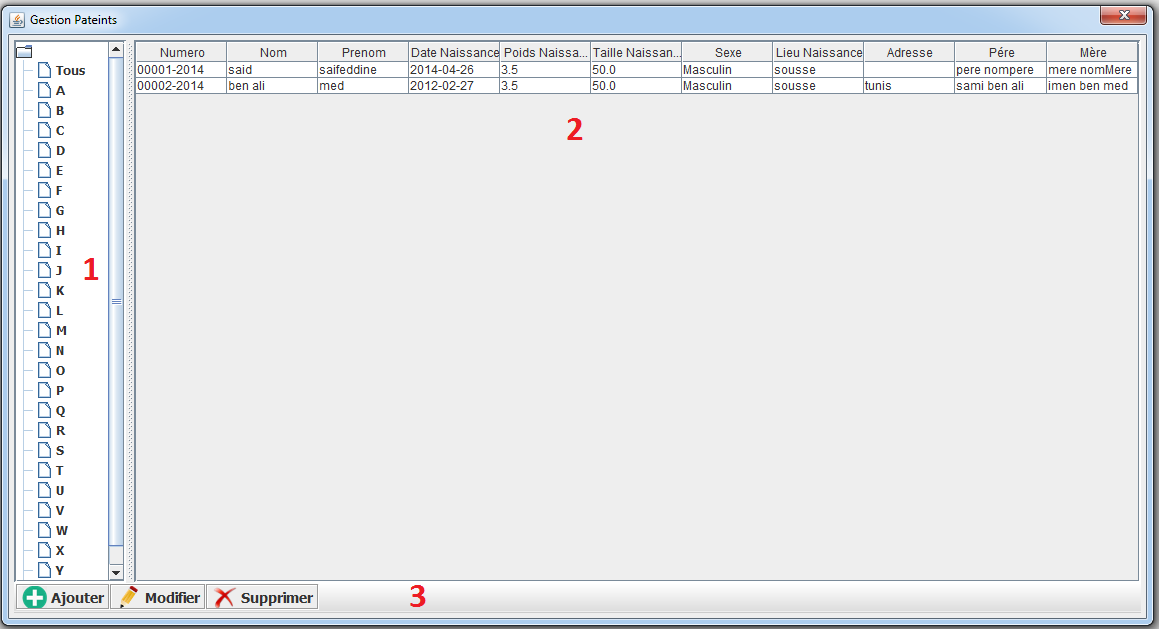


Figure : Interface graphique de gestion des patients

### Interface graphique ajouter patient

Grâce à cette interface (figure 39), la secrétaire pourra ajouter un patient.

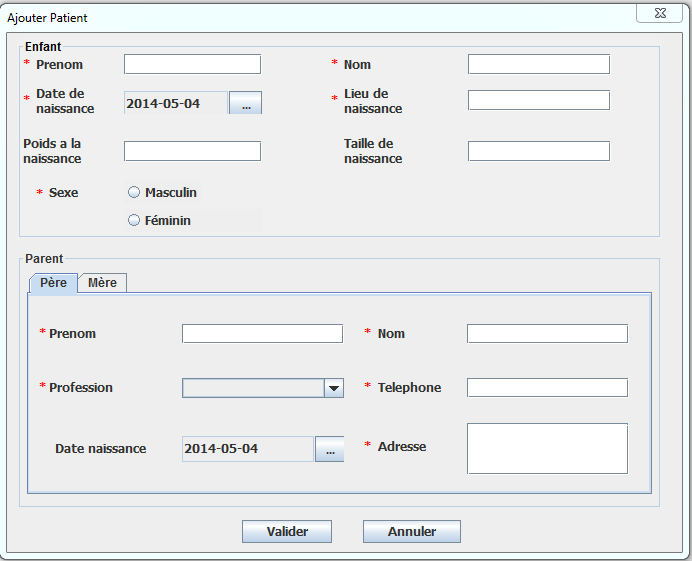


Figure : Interface graphique d'ajout un patient

### Interface graphique gestion des rendez-vous

Dans l'interface ci-dessous, permet de gérer et d'afficher la liste des rendez-vous.

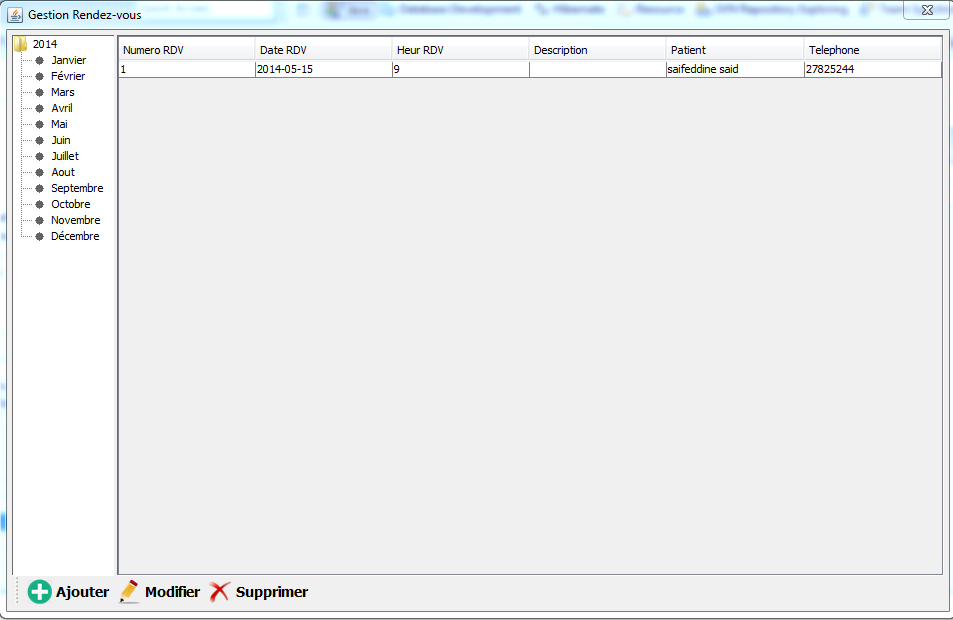
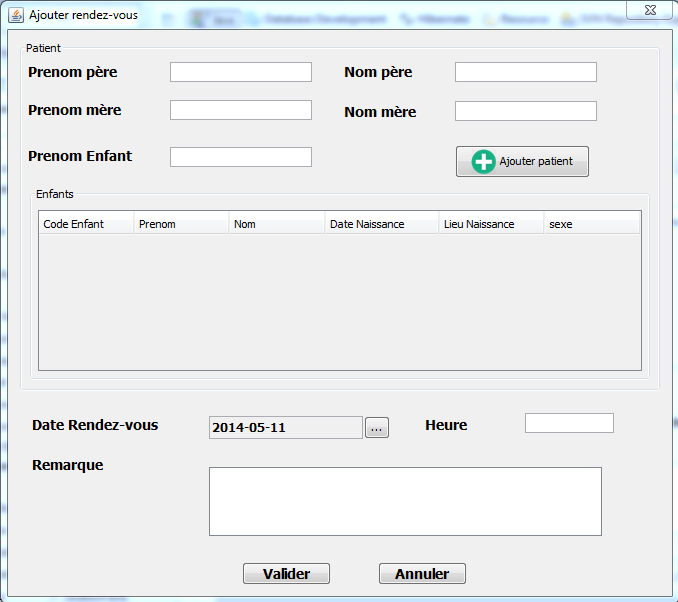


Figure : Interface graphique de gestion des rendez-vous

### Interface graphique ajouter rendez-vous

Grâce à cette interface (figure 41), la secrétaire pourra ajouter un rendez-vous.

Figure : Interface graphique d'ajout d'un rendez-vous



### Interface graphique de gestion des médicaments

L’interface ci-dessous, permet de gérer et d'afficher la liste des médicalement.

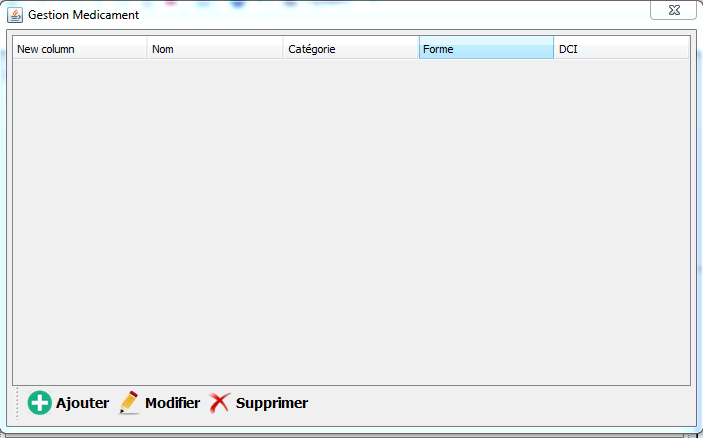
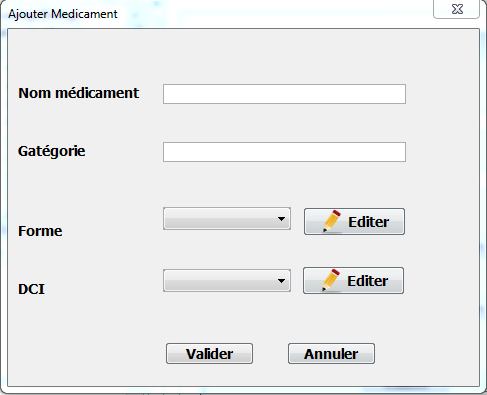


Figure : Interface graphique de gestion des médicaments

### Interface graphique d'ajout d’un médicament

Grâce à cette interface (figure 43), l'utilisateur pourra ajouter un médicament.

Figure : Interface graphique d'ajout un médicament



Dans l'interface ajouter médicament l'utilisateur de l'application peut gérer la forme et la dci de médicament grâce au bouton éditer.

# Conclusion

Notre projet consiste à analyser, concevoir et développer une application informatique facilitant la gestion d’un cabinet médical de pédiatres.

L’absence complète de l’outil informatique dans un cabinet médical rend la gestion des informations médicales assez difficile et le travail du personnel de ce cabinet fatiguant, exhaustif, imprécis et généralement mal géré.

Toutes les raisons précitées rendent indispensable la mise en place d’un système de gestion de cabinet, en effet, il répond aux exigences de la spécification et de la conception et aide à la résolution de ces divers problèmes de gestion.

Cette application informatique est une solution complète pour les cabinets individuels et collectifs. En toute convivialité. En outre, ce logiciel évoluera avec les besoins de tous les cabinets. Grâce à un enregistrement précis, ce logiciel offre automatiquement un soutien décisionnel éprouvé. De cette manière, le suivi des patients devient encore plus facile.

# Bibliographie

[1] CHRISTIAN BAUER AND GAVIN KING (2007) Java Persistence With Hibernate

[2] Gregory Smith (2011) Bases de données PostgreSQL 9.0

[3]  [Mike Mason](http://www.eyrolles.com/Accueil/Auteur/mike-mason-66618) (2006) Subversion: Pratique du développement collaboratif avec SVN

1. http://www.enseignement.polytechnique.fr/profs/informatique/Julien.Cervelle/eclipse/ [↑](#footnote-ref-2)
2. http://www.ouestdecision.fr/contenu/les\_editeurs/fiches/fiche-produit-jasper.htm [↑](#footnote-ref-3)
3. http://fr.wikipedia.org/wiki/PostgreSQL [↑](#footnote-ref-4)
4. http://www.jmdoudoux.fr/java/dej/chap-junit.htm [↑](#footnote-ref-5)
5. http://fr.wikipedia.org/wiki/Apache\_Subversion [↑](#footnote-ref-6)