

# Conception

Malad'IF - Votre santé est notre priorité

Manuel Amouroux, Victor Bonin, Alexandre Carpentier, Yohan Gracia

10 Avril 2017

## Table des matières

<b>1</b>	<b>Historique des versions</b>	<b>1</b>
1.1	Version 0.1 . . . . .	1
1.2	Version 1.1 . . . . .	1
1.3	Version 1.1 . . . . .	1
<b>2</b>	<b>Introduction</b>	<b>2</b>
2.1	Présentation du document . . . . .	2
2.2	Terminologie et abréviations . . . . .	2
<b>3</b>	<b>Architecture</b>	<b>3</b>
3.1	Cas d'utilisation . . . . .	3
3.2	Vue logique . . . . .	4
<b>4</b>	<b>Diagrammes UML</b>	<b>5</b>
4.1	Classes . . . . .	5
4.2	Séquence . . . . .	6
4.3	Etats-transitions . . . . .	6
<b>5</b>	<b>Description des algorithmes importants</b>	<b>7</b>
<b>6</b>	<b>Structuration des données</b>	<b>7</b>

## 1 Historique des versions

### 1.1 Version 0.1

- **Date** : Mercredi 29 Mars 2017
- **Auteur** : Manuel Amouroux, Victor Bonin, Alexandre Carpentier, Yohan Gracia
- **Description** : Ajout du layout général du document

### 1.2 Version 1.1

- **Date** : Mercredi 05 Avril 2017
- **Auteur** : Manuel Amouroux, Victor Bonin, Alexandre Carpentier, Yohan Gracia
- **Description** : Ajout de l'architecture et des diagrammes UML

### 1.3 Version 1.1

- **Date** : Dimanche 09 Avril 2017
- **Auteur** : Manuel Amouroux, Victor Bonin, Alexandre Carpentier, Yohan Gracia
- **Description** : Ajout de l'algorithme et des structures de données

## **2 Introduction**

### **2.1 Présentation du document**

Après avoir analysé et spécifié les besoins dans le livrable précédent, il s'agira dans ce document d'expliquer notre démarche de conception et nos choix d'implémentation, ainsi que les solutions qui ont été retenues, au niveau architectural, organisationnel, algorithmique et de la structure de donnée.

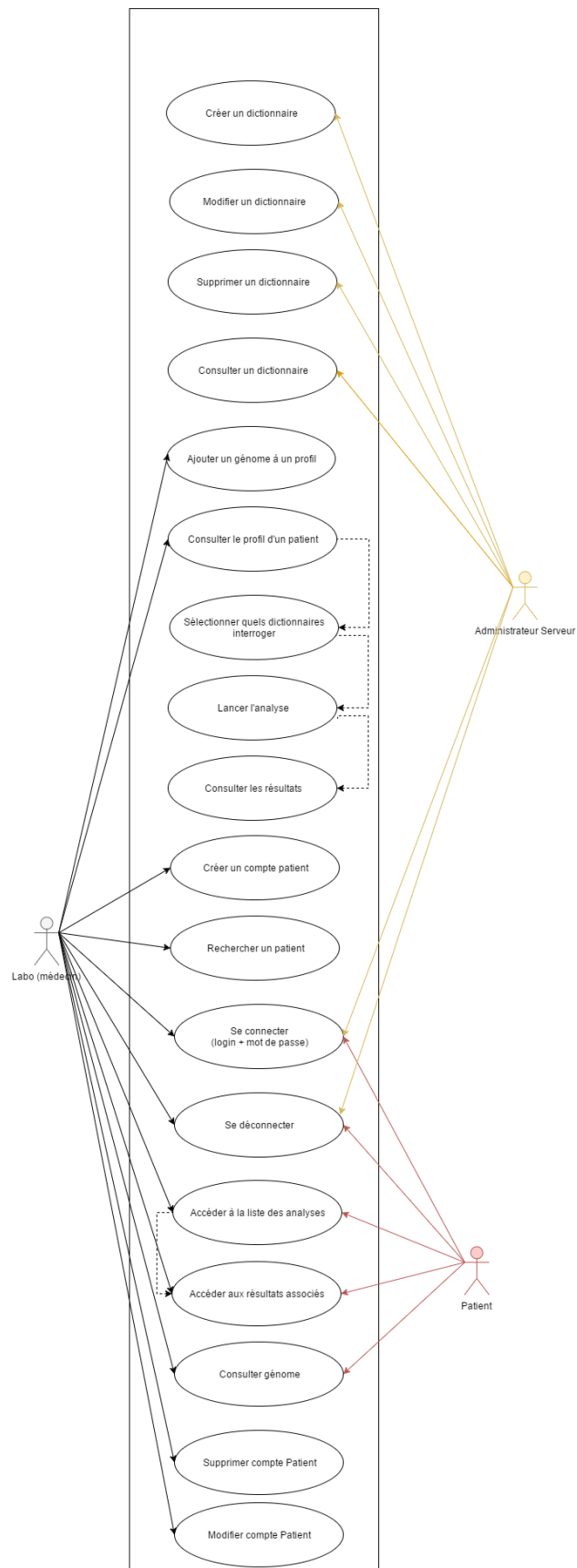
L'objectif de ce document est donc d'expliquer clairement au lecteur le fonctionnement de notre logiciel.

### **2.2 Terminologie et abréviations**

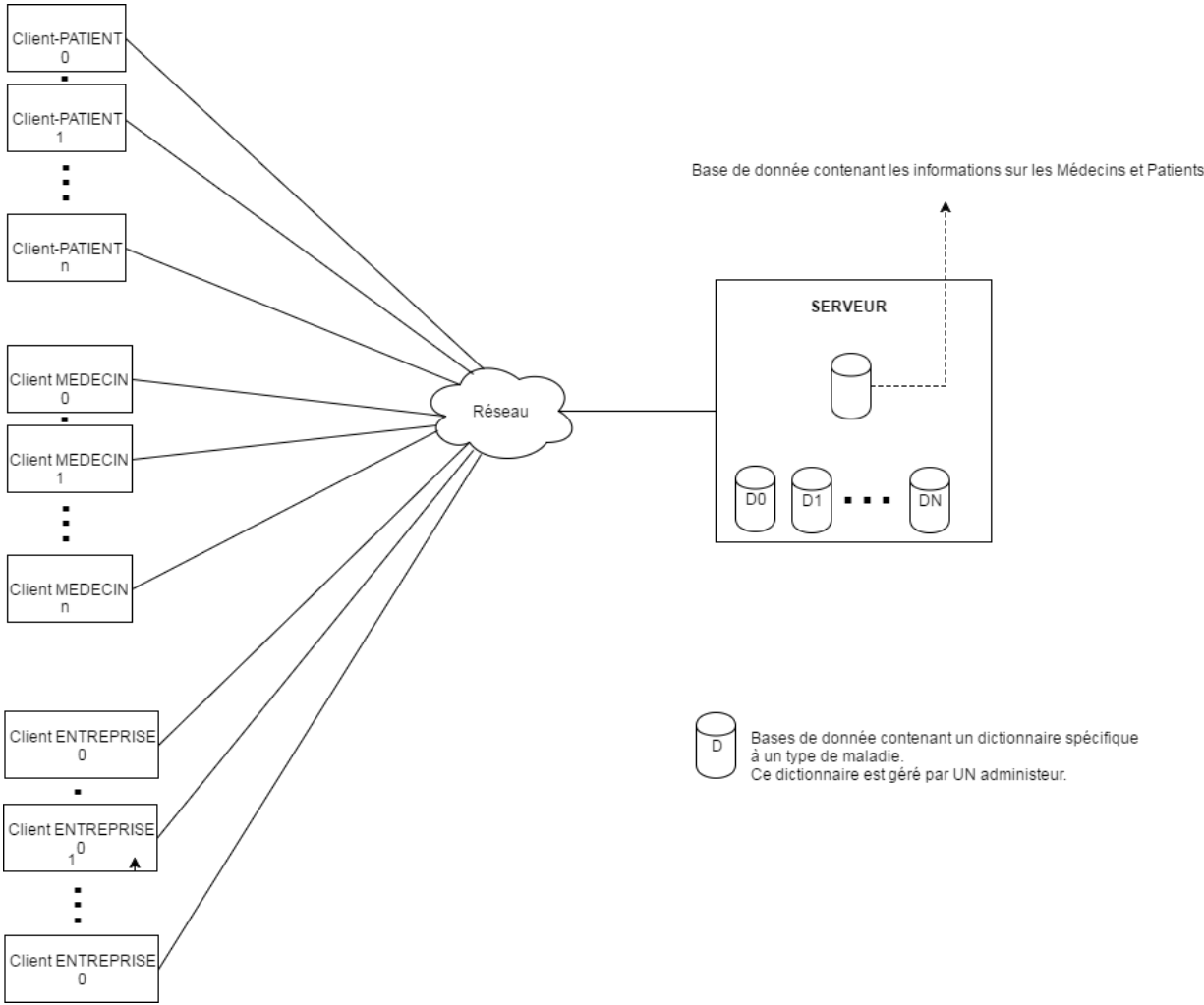
Nous n'utilisons pas d'abréviation particulière dans ce document.

## 3 Architecture

### 3.1 Cas d'utilisation



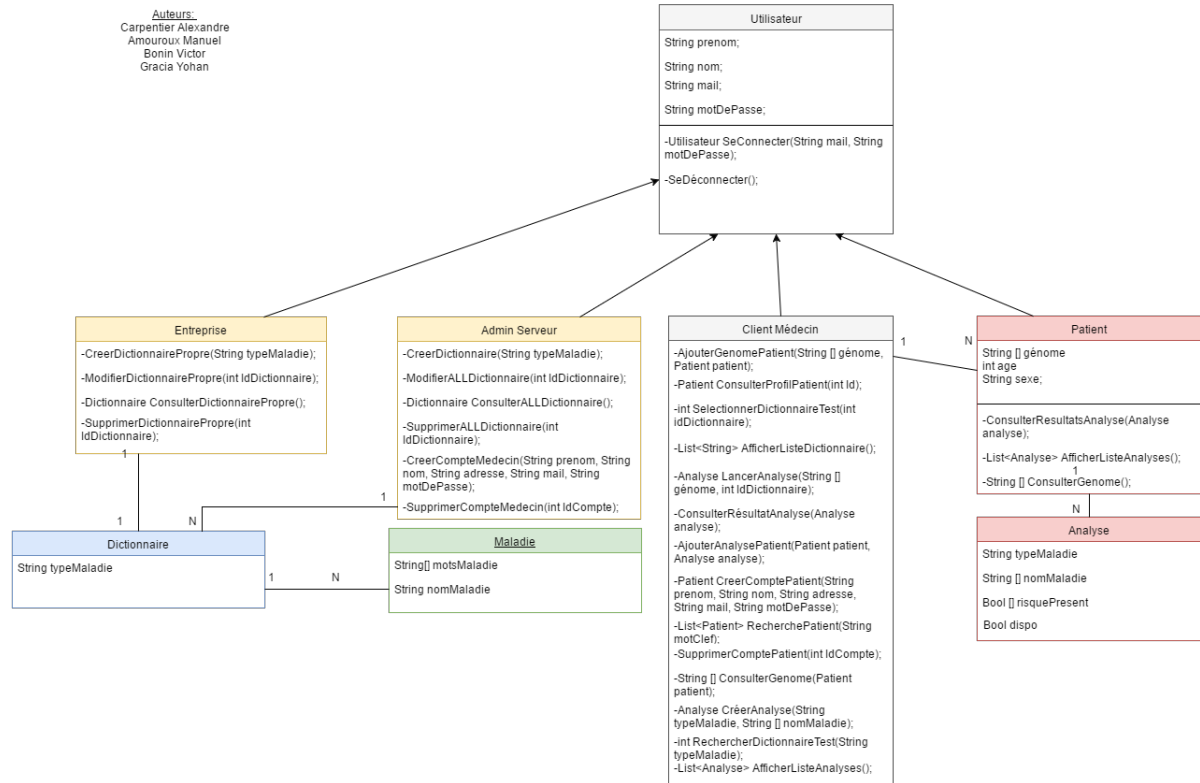
### 3.2 Vue logique



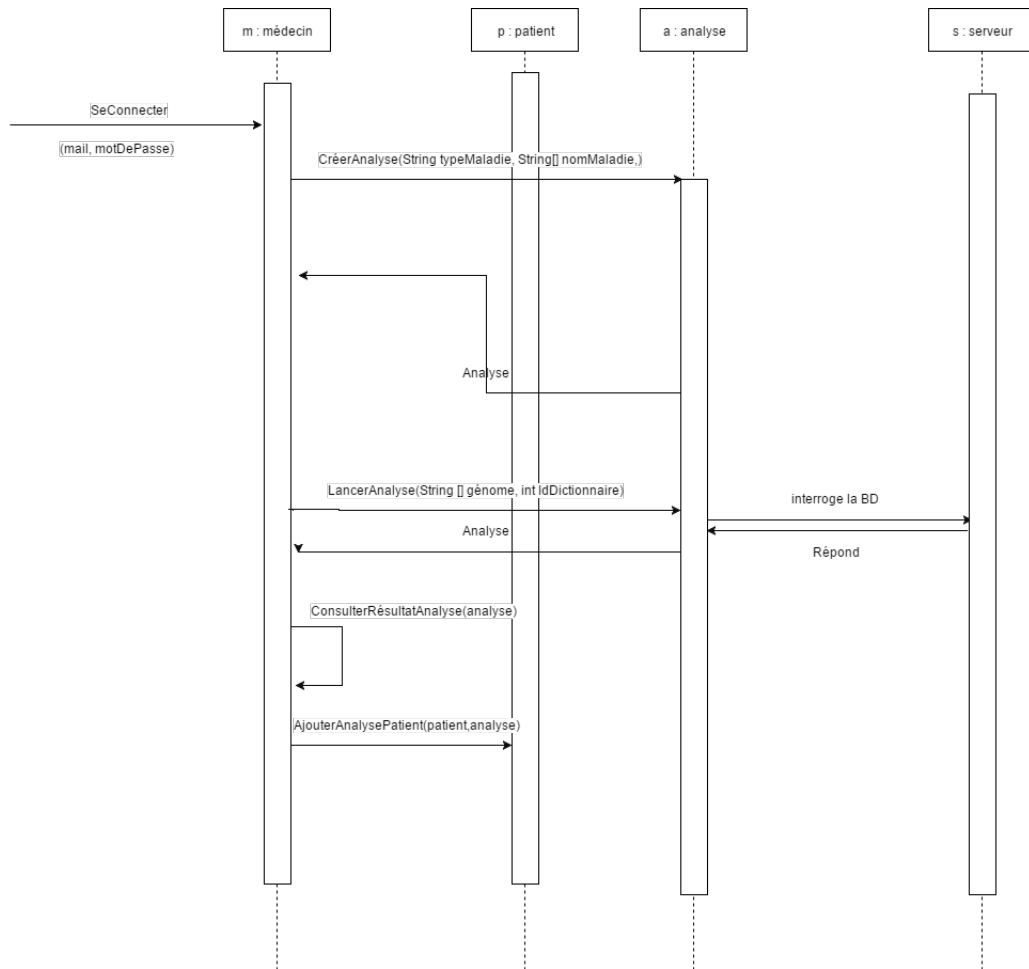
## 4 Diagrammes UML

### 4.1 Classes

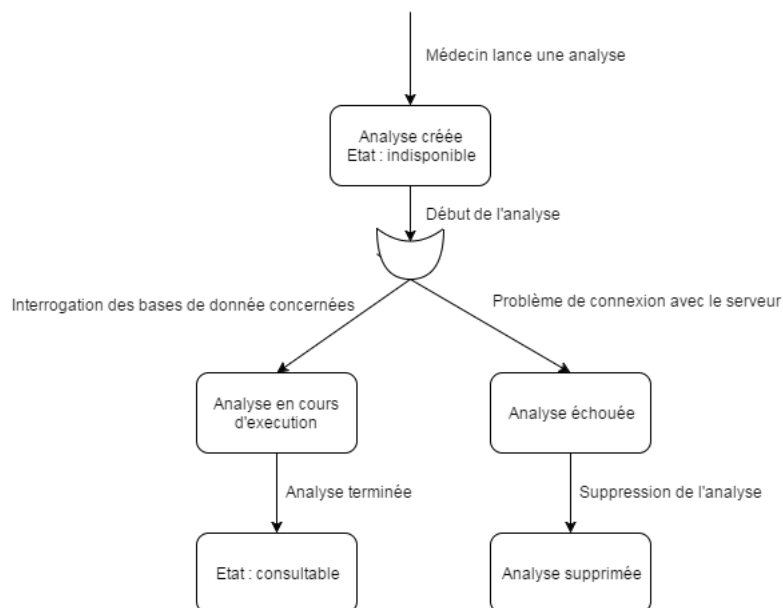
Diagramme de classe Malad'IF:



## 4.2 Séquence



## 4.3 Etats-transitions



## 5 Description des algorithmes importants

Le seul algorithme complexe de notre solution est la fonction d'analyse cherchant les correspondances entre le génome d'un patient et les dictionnaires sélectionnés.

### Algorithme de recherche de maladie:

Données : Génome du patient (mots = 1 000 000, mot = 256 caractères)

Liste de mots symptomatiques de la maladie (mots « génome »)

But : Rechercher l'apparition des mots symptomatiques de la maladie dans le génome du patient. L'apparition de l'ensemble de la liste spécifique à une maladie indique que le patient souffre possiblement de la maladie et provoquera des examens complémentaires.

Algorithme :

```
Boolean positif = true;

Pour tous (les mots de la liste symptomatique && positif)
{
    positif = false;
    Pour tous (les mots du génome)
    {
        Si (mot identique)
        {
            positif = true;
            break;
        }
    }
}

return positif;
```

Nous pensons faire la recherche des mots identiques à l'aide d'une requête SQL, ce qui réduirait grandement le temps d'exécution de l'algorithme car le second "Pour tous" serait "caché" (SELECT mot FROM génome WHERE mot = motSymptomatique).

## 6 Structuration des données

Pour réaliser ce projet nous avons dû décider quelles structures de données nous utiliseront pour nos différents éléments.

Le premier élément est le génome, que nous avons décidé de représenter sous la forme d'un tableau de chaînes de caractères.

Le second élément est le dictionnaire. Lorsqu'un dictionnaire est chargé depuis la base de données sur le logiciel, il est représenté sous la forme d'un arbre binaire de recherche qui prend pour clef le nom de chaque maladie composant le dictionnaire, et qui associe à cette clef le génome de la maladie. Ce choix de structure nous permettra de réaliser l'algorithme de manière efficace.

Nous n'avons pas d'autre structure de données notable pour réaliser notre projet.