



Rapport de projet d’IAE



|  |  |
| --- | --- |
| Projet réalisé par  Jonas ELYSÉE et Paul FERRARON | Pour la date du  31/03/2017 |

**Table des matières**

[1. Rappel du sujet 1](#_Toc478737984)

[2. Réalisation du projet 2](#_Toc478737985)

[A. Outils utilisés 2](#_Toc478737986)

[B. Modélisation du problème 2](#_Toc478737987)

[C. EJBs 2](#_Toc478737988)

[D. Webservice REST 2](#_Toc478737989)

[3. Tests et résultats 3](#_Toc478737990)

## Rappel du sujet

Ce projet d’IAE (Intégration d’Applications pour l’Entreprise) devait être réalisé en binôme pour le 31/03/2017. Il consiste en la réalisation d’une application J2EE permettant la gestion d’un cabinet médical. En voici un schéma conceptuel :

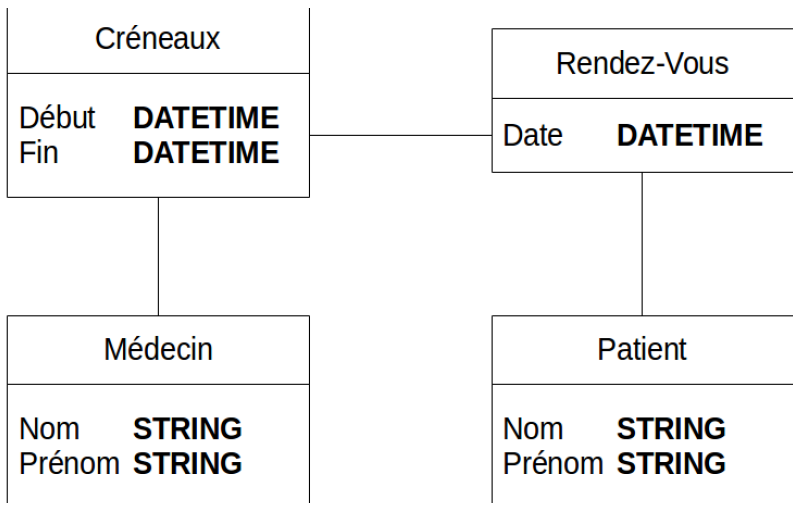


Figure 1 : Schéma conceptuel d'un cabinet médical

Pour le mener à bien, il fallait :

1. Proposer une modélisation sous la forme d’un ensemble d’entité JPA (Médecin, Créneau, RDV et Patient) ;
2. Créer des EJB s’appuyant sur l’EntityManager permettant de gérer le CRUD (Create, Read, Update & Delete) sur les entités Médecin et Patient ;
3. Créer des EJB permettant de lister les créneaux libres, prendre un rendez-vous, d’annuler et de modifier un rendez-vous ;
4. Exposer les différents EJB dans des Web Service REST (Comme dans les TPs précédents).

## Réalisation du projet

### Outils utilisés

Nous avons utilisé plusieurs outils pour mener à bien ce projet :

* NetBeans, le célèbre IDE (Integrated Development Environment) ;
* Payara, un server de type GlassFish sur lequel l’application se lance ;
* phpMyAdmin, le SGBD permettant d’interagir avec la BDD de l’application.

### Modélisation du problème

Pour modéliser le problème, nous avons utilisé les 4 classes suggérées : Médecin, Créneau, RDV et Patient. De plus, nous nous sommes fixés quelques conventions :

* Chaque Médecin possède ses propres créneaux ;
* Chaque Créneau a la même durée (exemple 1h) ;
* Un Créneau ne peut accueillir qu’un RDV au maximum ;
* Un RDV peut s’effectuer sur plusieurs créneaux ;
* Un RDV consiste en la rencontre d’un seul Médecin avec un seul Patient ;
* Un Patient fait référence à une personne et/ou une famille désirant avoir un RDV avec un Médecin ;

### EJBs

L’application comporte plusieurs EJBs (Enterprise Java Bean), 3 pour être exact, chacun destiné à des tâches bien précises :

* L’EJB gérant les CRUD de Médecin
* L’EJB gérant les CRUD de Patient
* Un EJB de gestion d’un RDV (le prendre, le modifier, l’annuler)

### Webservice REST

## Tests et résultats