**THEME** : Application web qui présente une clinique nommé « Cabinet Médical le Printemps » qui est une clinique de la place dont le médecin généraliste responsable préfère garder l’anonymat a son sujet. L’application web doit permettre au visiteur de pouvoir consulter tous les services proposés par le cabinet et lui offrant aussi la possibilité de pouvoir prendre rdv avec le médecin au sujet de l’un des services qu’il propose.

Pour ce faire j’ai commencé par une analyse UML du projet, ce dernier me permettra de mieux comprendre le fonctionnement de l’application et de pouvoir l’implémenter en respectant un organigramme bien structuré

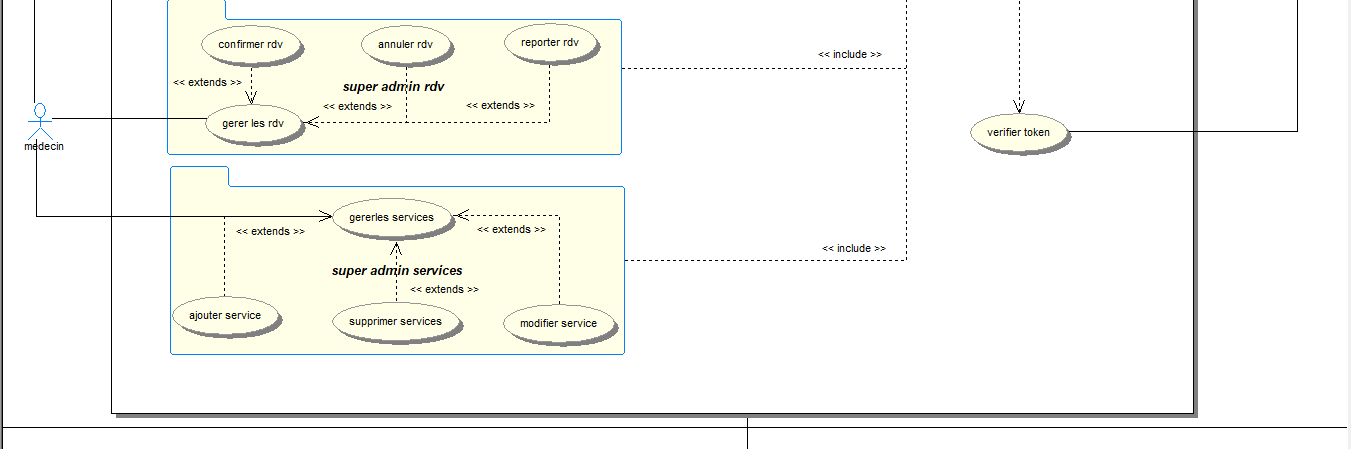
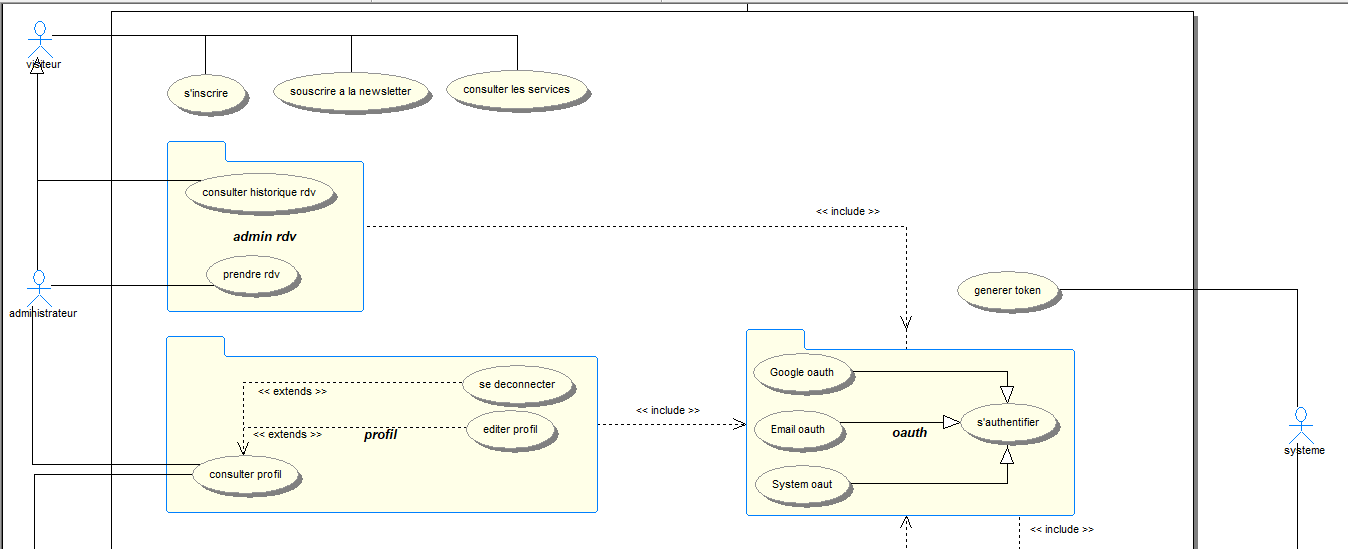
Dans mon analyse j’ai représenté 03 diagrammes UML à savoir le Diagramme des cas d’utilisation, le Diagramme de classe et le Diagramme de séquence.

* Diagramme des cas d’utilisations : ce diagramme nous permettra d’avoir une vue d’ensemble sur les différentes fonctionnalités de l’application et les différentes éventualités de chaque fonctionnalité si nécessaire tout en mettant en évidence les différents acteurs qui y interviendront.
* Diagramme de classe : ce diagramme quant a lui nous donne une idée de la structure de notre base de données et des migrations qui se feront sans oublier qu’il nous permet aussi d’avoir une idée sur les différents attributs de chaque entité de notre base de données et les différentes méthode ou fonctions auxquelles ces dernières sont rattachés s’il y’en a.
* Diagramme de séquence : ce diagramme comme sont nom l’indique nous donne les différentes séquences d’opérations sur une action bien précise ; dans notre cas on va s’attarder sur notre action principale qui est la prise de rendez-vous par l’utilisateur et nous allons détailler son processus au sein de l’application.

Maintenant passons à la présentation de nos diagrammes.

# Diagrammes

## Diagramme des cas d’utilisation



*Figures1 : Diagramme des cas d’utilisations (uses Cases)*

Explications : ce diagramme de cas d’utilisation nous présente tout d’abord un système *(représenté par le grand rectangle)* dans le quel interviennent quatre acteurs *(représentés par les bonhommes bleu)* à savoir :

* Le visiteur : qui est celui qui arrive sur le site et ne fait que visiter le site peut être pour sa première fois ou par simple curiosité alors ce dernier n’a la possibilité d’effectuer que trois actions dans le site à savoir :

1. S’inscrire : cette action lui permettra d’avoir un compte au sein de l’application et ce compte lui donnera accès a l’ensemble des autres fonctionnalités du sites.
2. Souscrire à une newsletter : cette action invitera le visiteur à envoyer son adresse email au system afin de rester informé de toutes les nouvelles actualités à propos de la clinique.
3. Consulter les services : cette action envoie l’utilisateur a la section services de l’application afin qu’ils puissent voir chaque service proposé par la clinique.

* L’administrateur : dans ce diagramme l’administrateur est tout utilisateur/visiteur possédant un compte dans le system ; ce compte la qui lui donne accès a l’ensemble des fonctionnalités de l’application en plus de celles auxquelles le simple visiteur a accès. Dans ce diagramme nous disons que :

1. Un administrateur peut consulter l’historique de ses rendez-vous et il peut aussi prendre un rendez-vous, ces deux actions incluent obligatoirement une authentification de l’administrateur dans le system *(c’est la relation « include »)*.
2. Un administrateur peut aussi consulter son profil une fois authentifier dans le system. Pendant que l’administrateur consulte son profil, selon sa volonté il peut soit éditer son profil ou se déconnecter *(c’est la relation « extend »)*.

* Le super administrateur/le médecin : ici lui il est beaucoup plus axé sur les taches de gestion du site à savoir :

1. Tout comme l’administrateur, il peut consulter son profil et l’éditer ou se déconnecter selon ses besoins.
2. Il gère les rendez-vous c’est-à-dire qu’après avoir accéder à ses rendez-vous, selon sa disponibilité et ses besoins il peut soit annuler un rendez-vous, soit reporter un rendez-vous ou soit confirmer un rendez-vous.
3. Il gère aussi les services c’est-à-dire qu’il peut ajouter, modifier ou supprimer un service selon sa volonté.

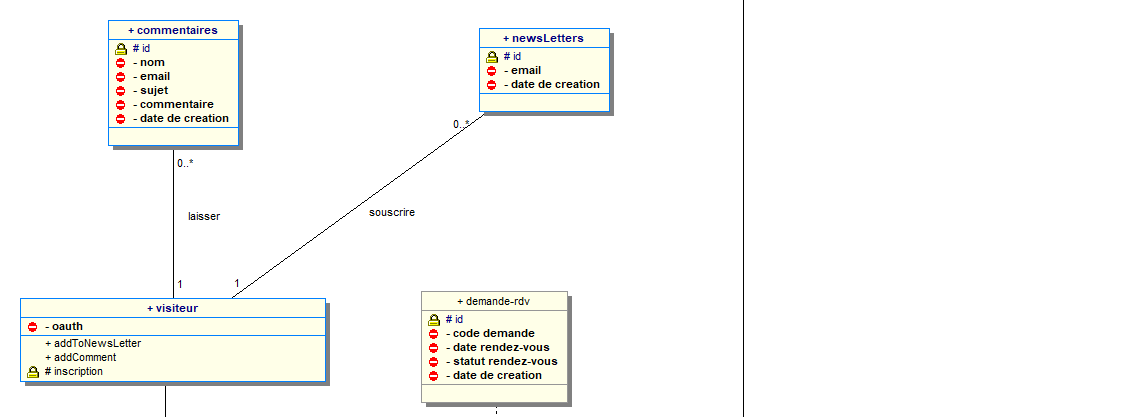
* Le système/l’API : contrairement aux autres acteurs, le system dans notre analyse n’est pas un acteur primaire mais plutôt un acteur secondaire car il n’intervient pas directement dans le système d’analyse. Ce dernier a comme action à effectuer :

1. La génération des jetons d’accès/tokens
2. La vérification des tokens/jetons d’accès.

Il est a noté toutes les authentifications effectuées dans le system d’analyse nécessitent obligatoirement une vérification des jetons d’accès par le système.

**NB :**  Toutes les relations portantes le marqueurs « include » sont des actions qui ne peuvent pas être effectué tant que l’action vers la quelle leur flèche pointe n’est pas vérifié ; Toutes les relations portantes le marqueurs « extends » sont des actions dont l’élément entrant par la flèche est facultatif pour cette action-là. Sans oublier que toutes les actions de l’administrateur incluent une authentification par le system.

## Diagramme de Classes



*Figures2 : Diagramme de classes*

Explication : ce diagramme est un diagramme de classe qui repressente les différentes entités de notre application. Pour être plus explicite, les entités ici sont les éléments que nous allons stockés dans notre base de données.

Ici nous dénombrons huit entités dont deux d’entres elles sont filles a une entités mère donc on se résume à six entité qui sont :

* L’entité Utilisateur : celle-ci est l’entité mère qui donne naissance a 2 entités filles qui sont visiteur et administrateur *(nos acteurs principaux du diagramme de cas d’utilisation)*. Dans cette entité, nous allons stocker les informations de tous nos utilisateurs de façon sécurisé et c’est grâce à ces informations là qu’ils se connecteront à l’application.
* L’entité Newsletter où les emails de tout ceux qui auront souscris seront stockés.
* L’entité Commentaire où les commentaires des différents visiteurs seront stockés.
* L’entité Services où seront stocker les différents services proposés par la clinique.
* L’entité Rendez-vous où seront stockées de façon sécurisé les informations de nos patients et la description de leur symptôme.
* L’entité Demande-Rdv où nous allons stocker les chaque rendez-vous et suivre le statut du rendez-vous.

Il est à noter que ces entités ne sont pas juste la comme ça, elles sont toutes en relations :

* Visiteur et Commentaires

*Un visiteur peut laisser un ou plusieurs commentaires*

*Un commentaire n’est laissé que par un et un seul visiteur*

* Visiteur et Newsletter

*Un visiteur peut souscrire à une ou plusieurs newsletters*

*Une newsletter ne peut être souscrite que par un et un seul visiteur*

* Administrateur et Services

*Un administrateur peut créer un ou plusieurs services*

*Un service ne peut être créer que par un seul administrateur*

* Utilisateur, Rendez-vous et Demande-Rdv

*Un utilisateur peut demander plusieurs rendez-vous et*

*Un Rendez-vous peut être demander par un ou plusieurs utilisateurs*

*Une fois les deux réunis ceci génère une demande de rendez-vous avec un statut.*

* Rendez-vous et Services

*Un rendez-vous ne concerne qu’un et un seul service de la clinique*

*Un service peut être concerné par plusieurs rendez-vous*

## Diagramme de séquences

