Gestion des Rendez-vous d’un cabinet médical

**GesCabinet**

[**Cahier des charges**](https://drive.google.com/file/d/1c6fauMwI7O7fHbvAusvdXZ9KJTGQqJcn/view?usp=sharing)

* **Introduction**

Dans le cadre de fin de formation d’Orange Digital Kalanso qu’il nous ait demandé de réaliser une application web pour la validation du Certificat Développeur Fullstack.

Chaque matinles patients sont contraints de se présenter avant l’ouverture de la clinique pour obtenir un rendez-vous avec les médecins, il n’est pas toujours facile de gérer les différentes prises de rendez-vous. Répondre au téléphone, rappeler les heures de rendez-vous, fixer ou modifier des rendez-vous.

C’est pourquoi nous avons décidé de mettre en place une plateforme qui facilite la prise de rendez-vous en ligne d’un cabinet médical.

Avec ce projet, nous remettons la simplicité et la facilité dans la gestion médicale tout en permettant aux utilisateurs un gain de temps.

* **Etude de l’art et méthodologie**
* Solutions existantes et mode de fonctionnement en rapport avec le projet:

Au Mali, il existe deux projets similaires qui sont : Keneya-kura & Doctix. Ces deux plateformes ont pour objectif de mettre en relation un patient et le médecin de leur plateforme en planifiant le rendez-vous. nous par contre on voit les choses sous un autre angle,les cabinets peuvent avoir des comptes pour que les patients puissent les localiser pour planifier leurs rendez-vous, ils pourront aussi le modifier à tout moment avant le Jour-J. Le patient sera notifié par mail si son rendez-vous est validé ou pas.

Pour prendre un rendez-vous, le patient peut choisir un cabinet dans la localité de son choix avec un médecin de ce cabinet.

* Méthodologie de projet et outils à utiliser en rapport avec le projet

La méthodologie de projet utilisée pour la réalisation de **Gescabinet** est la méthode agile avec Trello.

* **Analyse et conception**

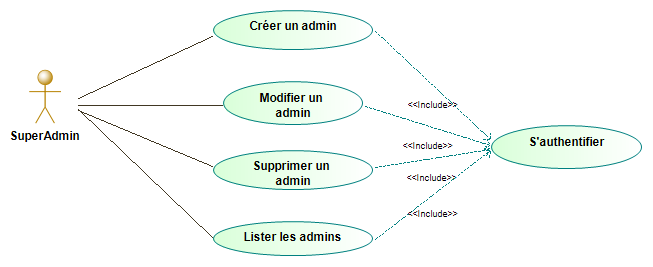
Nous allons voir les différentes fonctionnalités de l’application et les diagrammes de conception en utilisant le langage UML (Langage de modélisation). Comme diagrammes, on verra les cas d’utilisations, les classes et les séquences.

**Diagramme des cas d’utilisations**

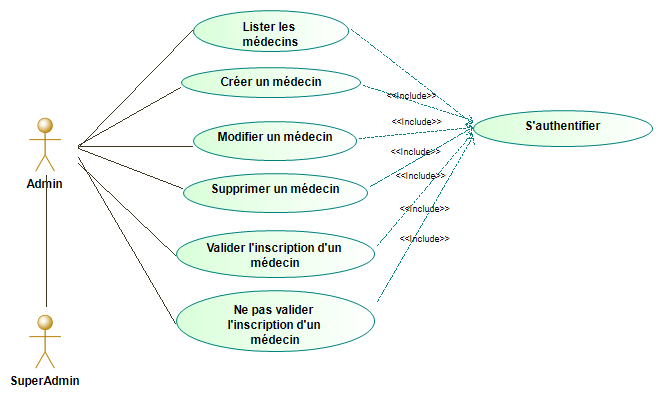
Il vous permet d’identifier les acteurs et leur interaction avec l’application.

Un acteur est toute entité qui réagit avec le système, comme acteur on peut trouver : Super Administrateur, Administrateur, Médecin, Patient

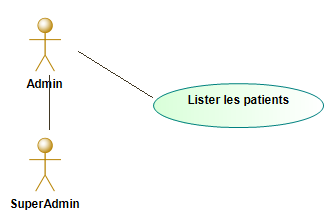
Cas d’un Super administrateur (Gestions des administrateurs)



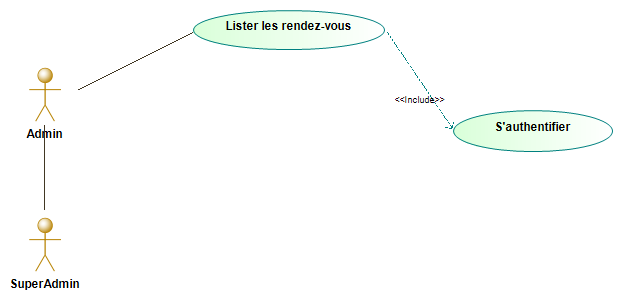
Cas des administrateurs et le super administrateur (Gestions des médecins)



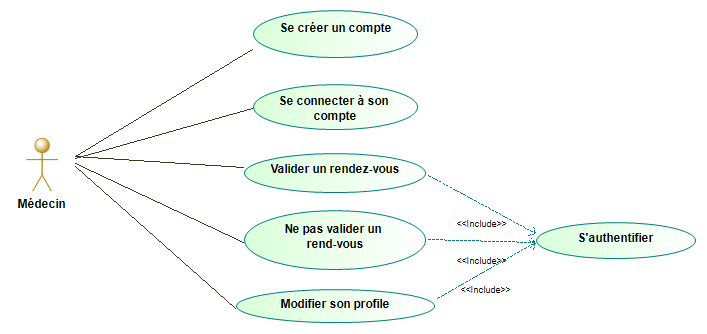
Cas des administrateurs et le super administrateur (Gestions des patients)



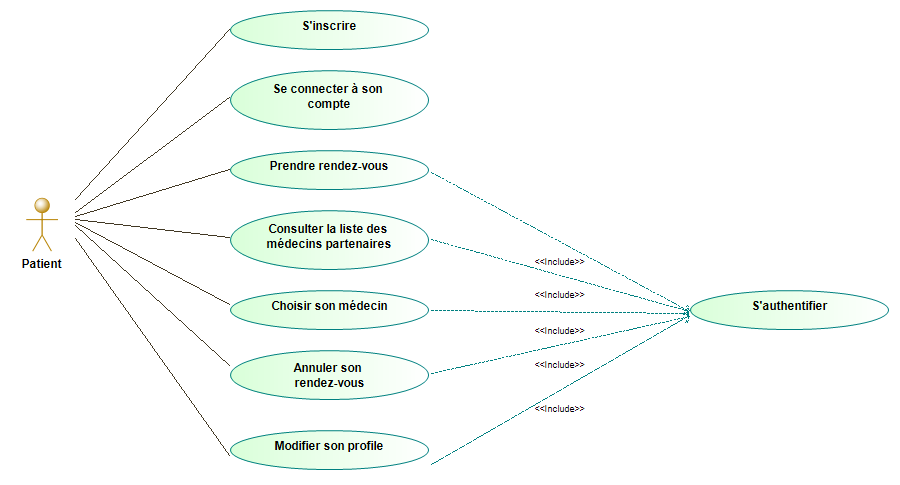
Cas des administrateurs et le super administrateur (Les rendez-vous)



Cas d’un médecin

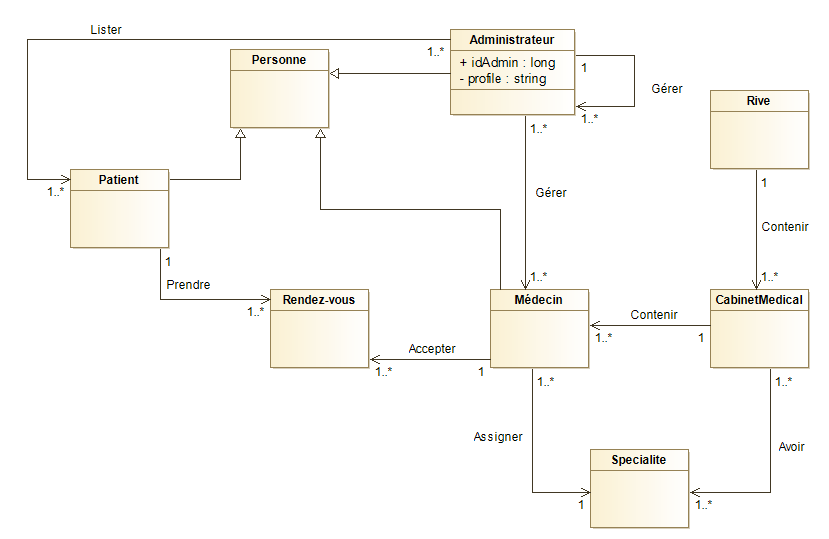


Cas d’un patient



**Diagramme de classe**

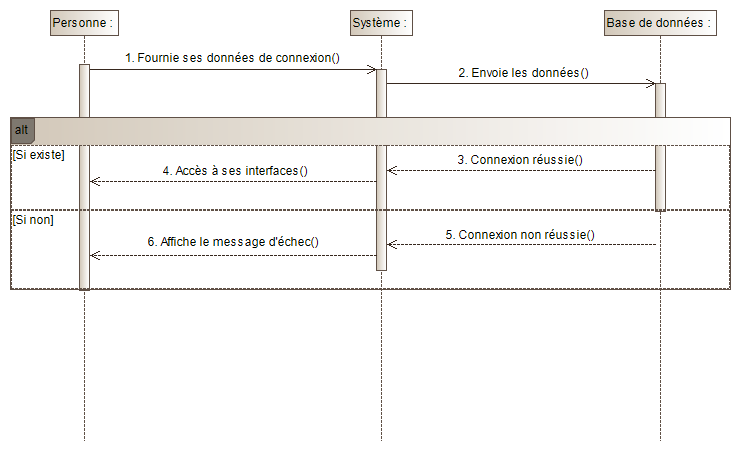
Ce diagramme est utilisé pour développer la structure des entités manipulées par les utilisateurs c’est-à-dire la structure du système tout en modélisant ses classes et les relations entre eux.



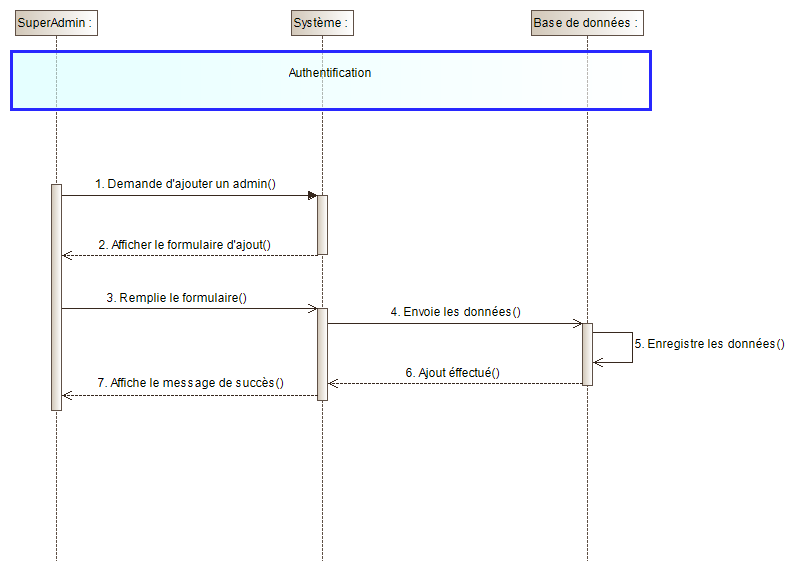
**Diagramme de séquence**

Il représente l’interaction entre les acteurs et le système.

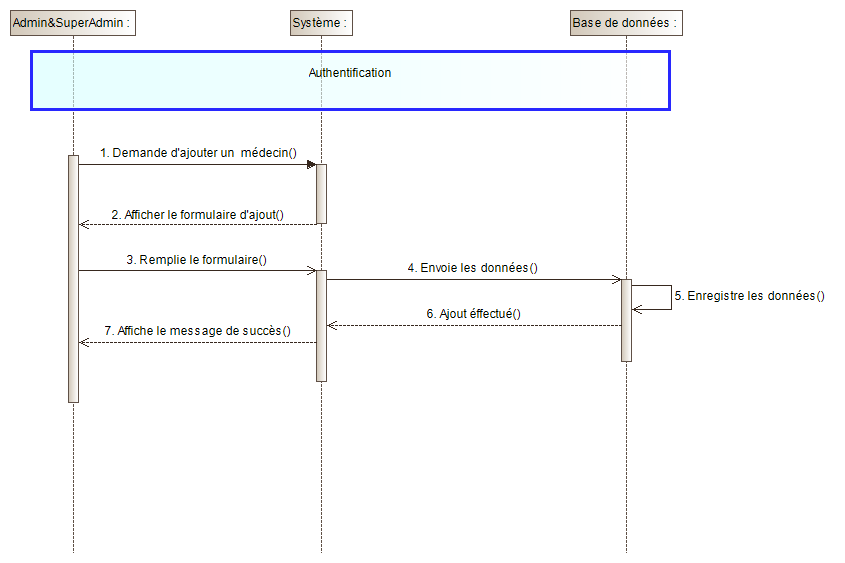
Séquence authentification



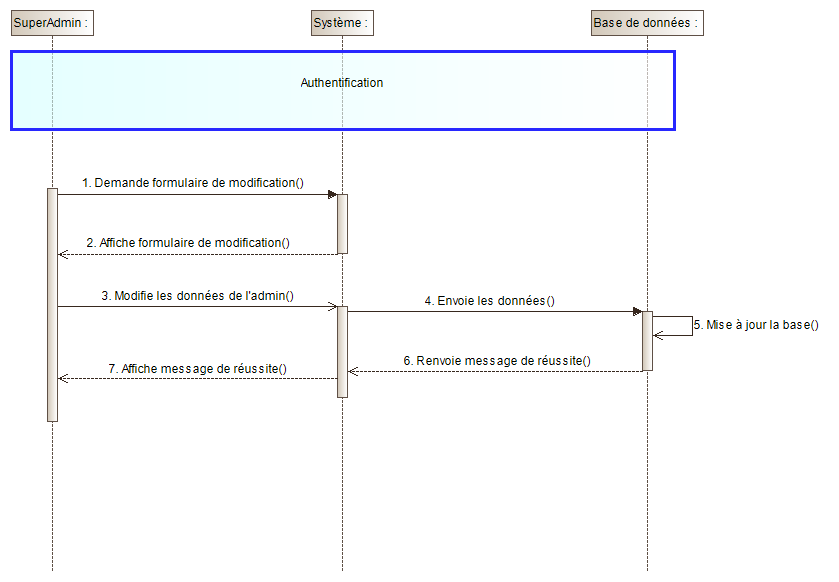
Séquence créer un admin



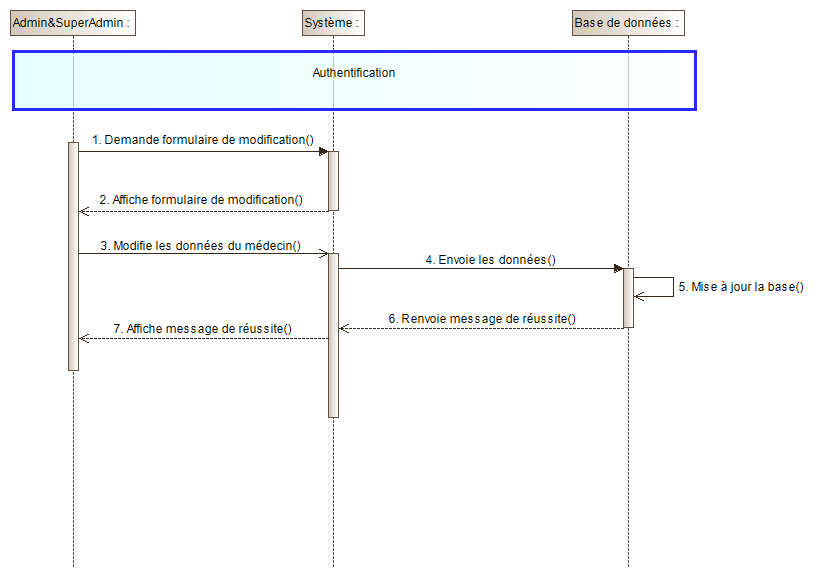
Séquence créer un médecin



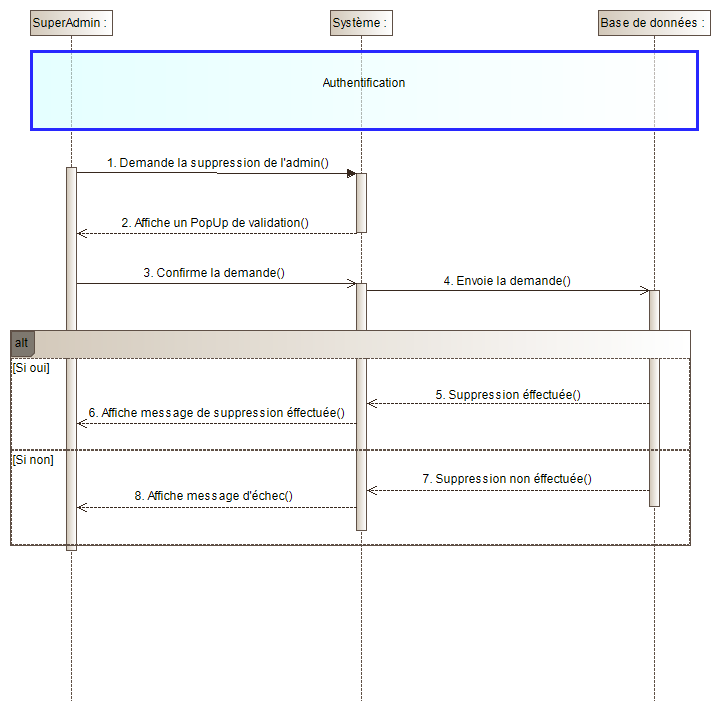
Séquence Modifier un admin



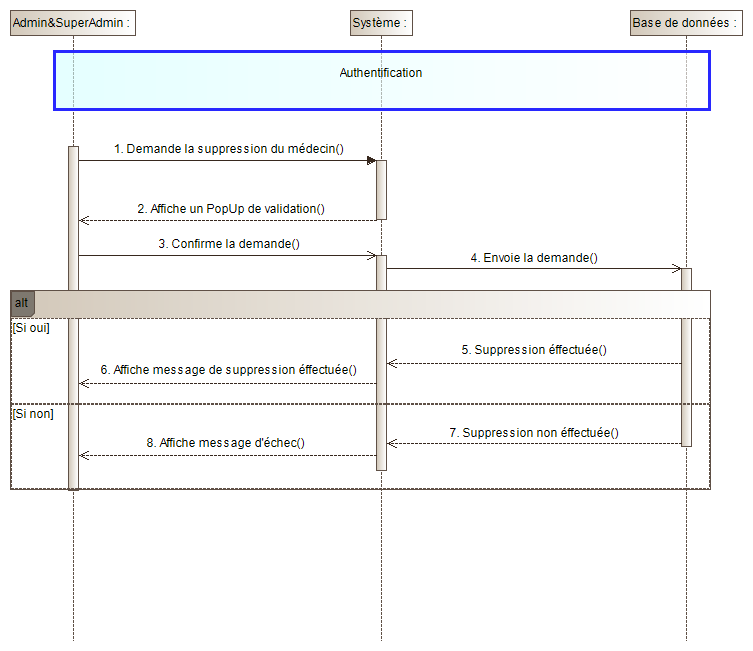
Séquence modifier un médecin



Séquence supprimer un admin



Séquence supprimer un médecin



* **Mise en place de la solution**
* Outils utilisés dans le cadre du projet

Les outils utilisés pour la réalisation de ce projet sont :

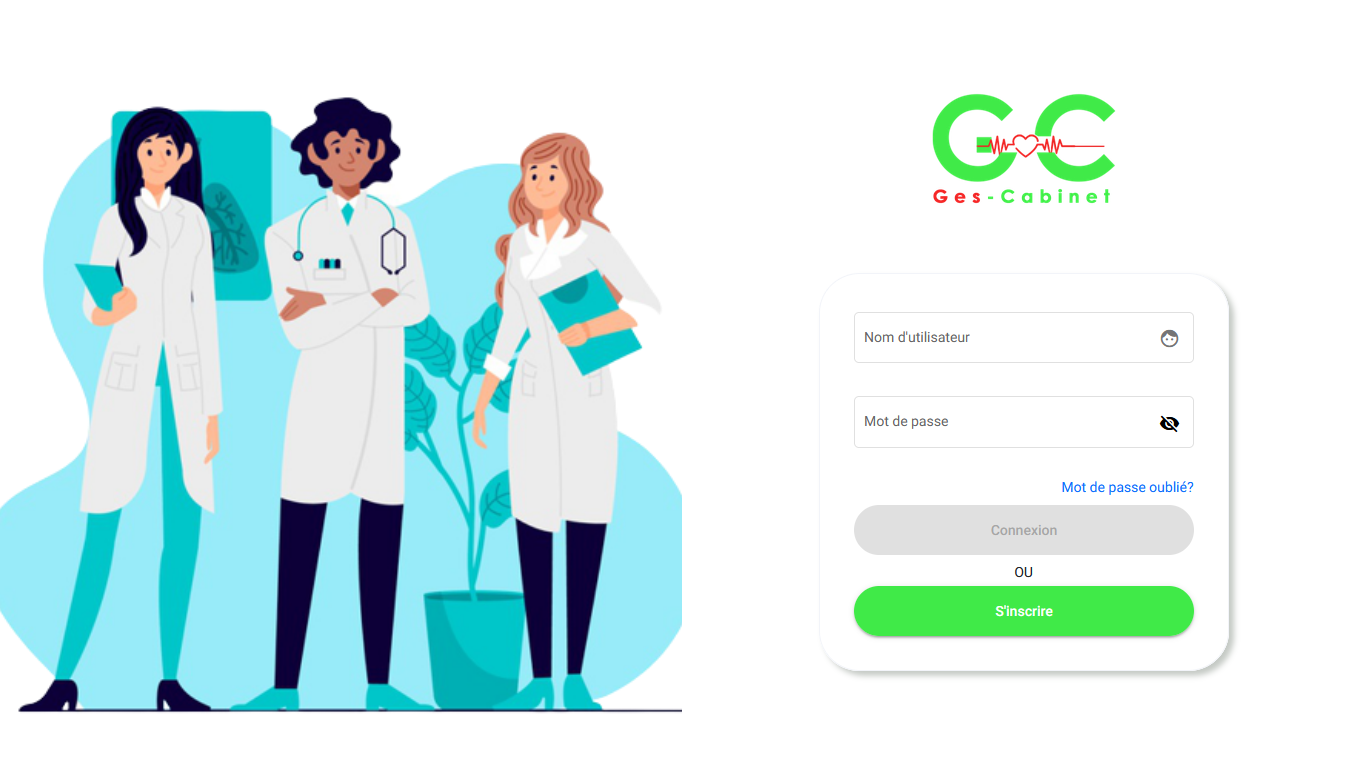
* **Spring Boot**, **Spring security** et **MySQL** pour le développement de l’API;
* **Angular** pour l’interface utilisateur;
* **WordPress** pour le landing page;
* **IntelliJ IDEA, Vs Code** Comme IDE de développement;
* **Git/Github** pour le versionning;
* **Trello** pour la gestion des tâches;
* **Discord** pour les réunions et espaces de partage;
* Maquettes

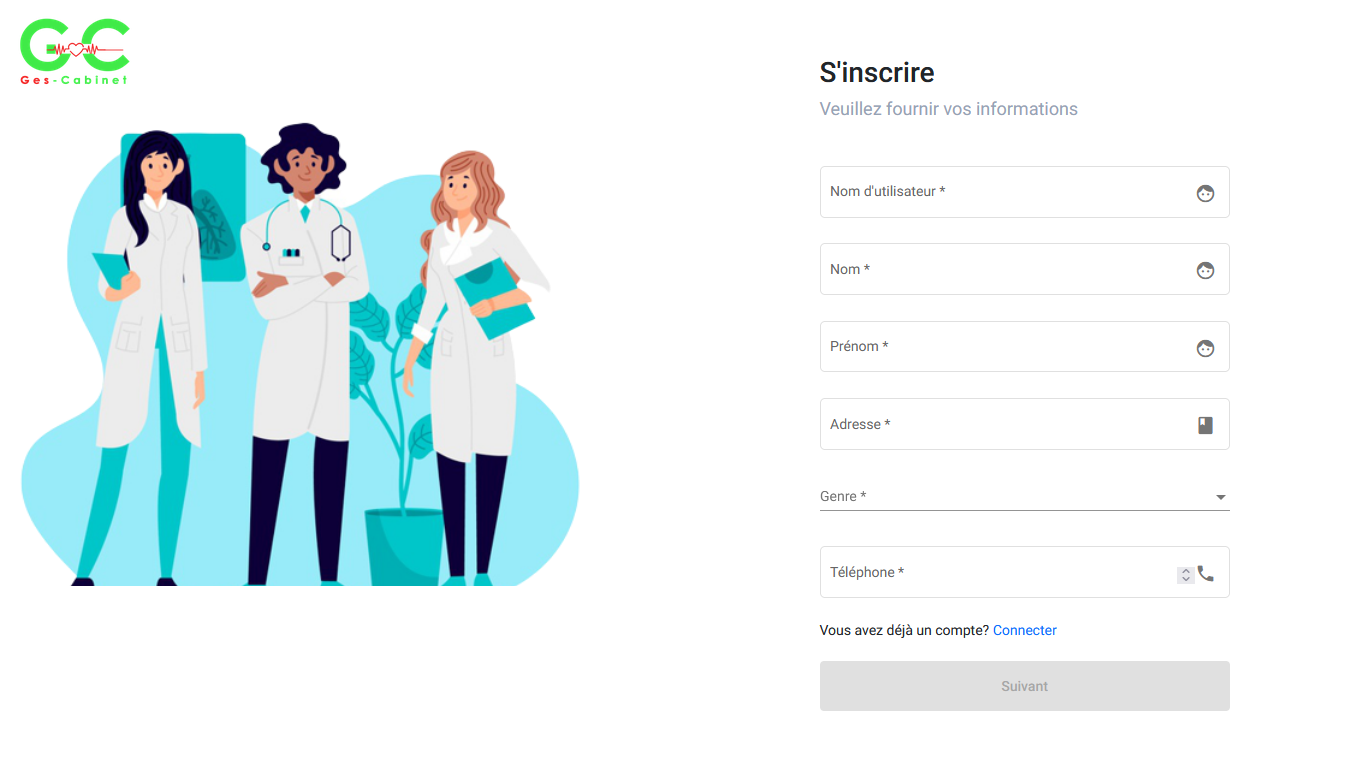
Les maquettes sont faites avec **Adobe XD**

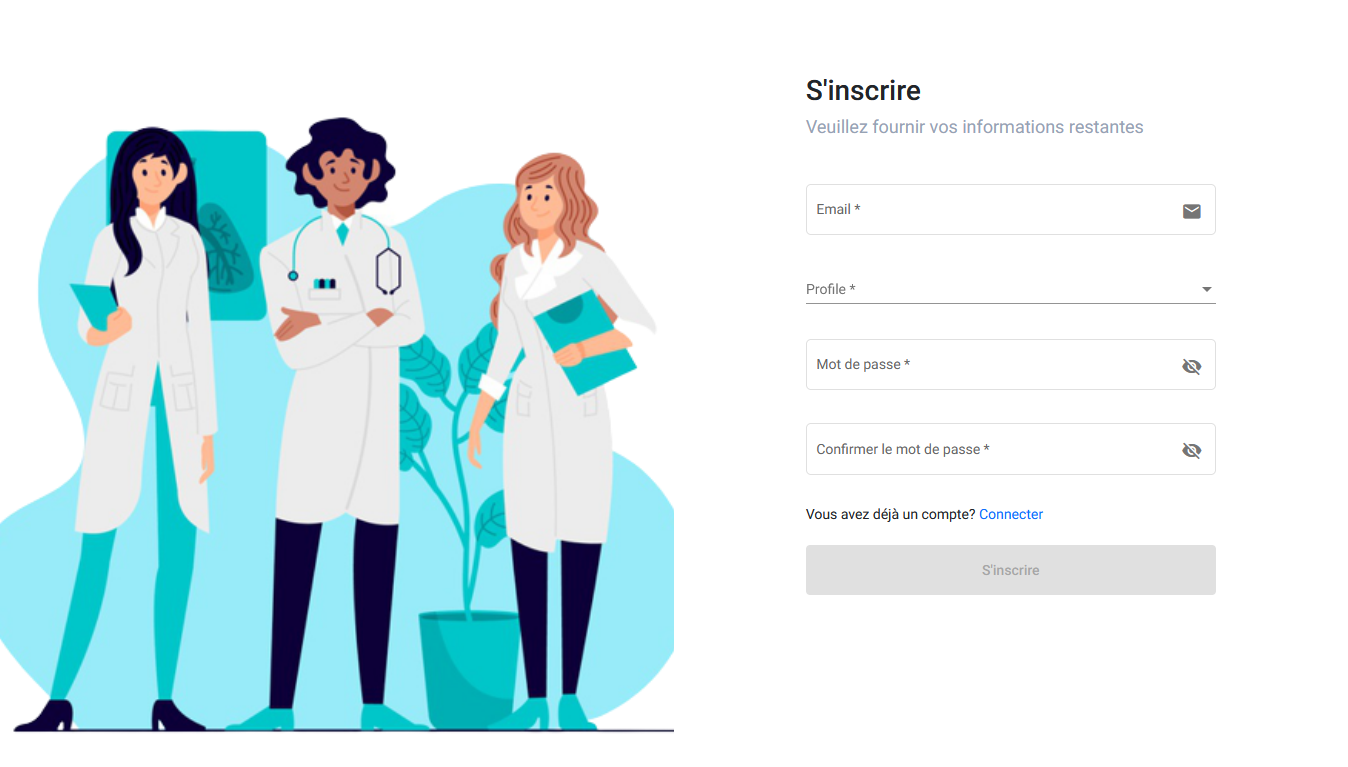
* Présentation de la solution

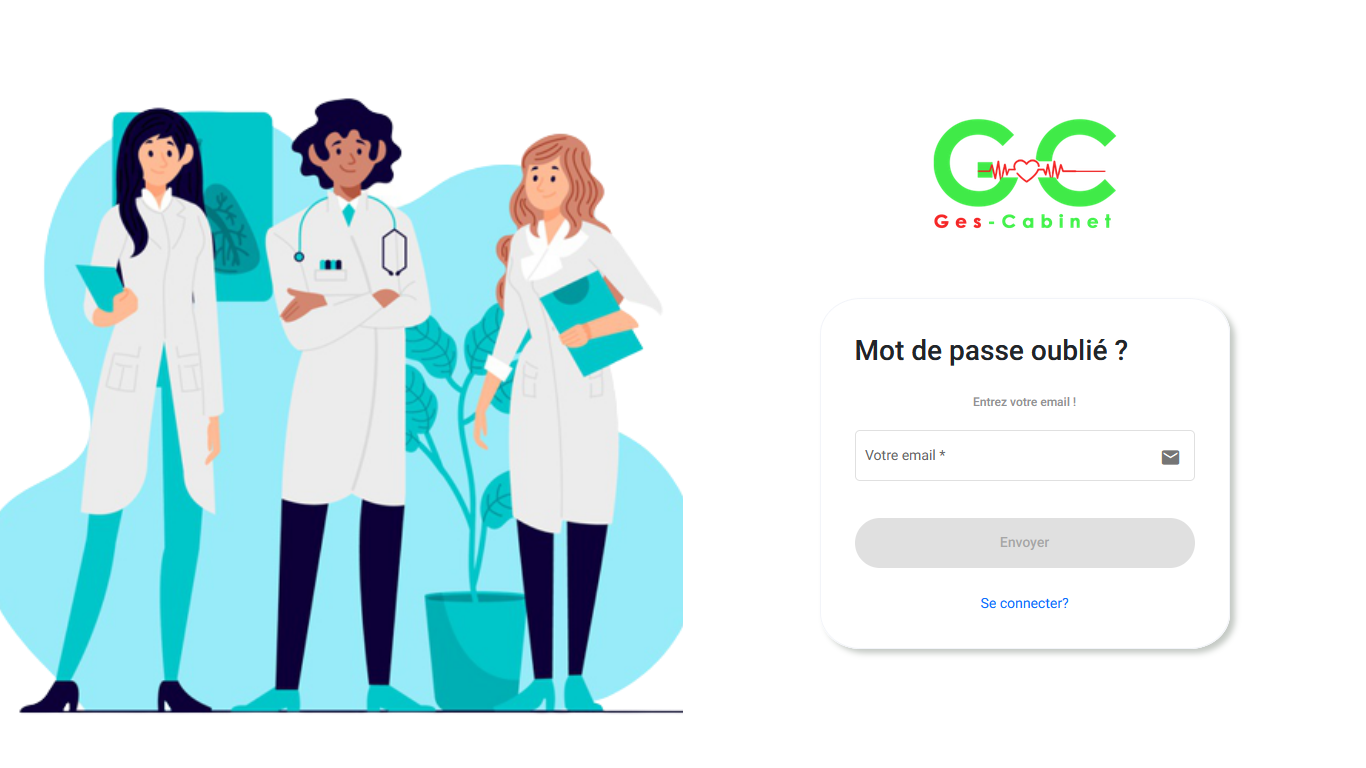
**GesCabinet** est une plateforme web pour les prises des rendez-vous en ligne. Avec cette plateforme, vous pouvez prendre des rendez-vous plus facilement que d’habitudes c’est-à-dire le processus de prise de rendez-vous entre un corps médical et ses patients. Chaque patient sera notifié par mail une fois que son rendez-vous sera accepté. Les médecins ou comme les patients peuvent aussi créer un compte plus facilement.

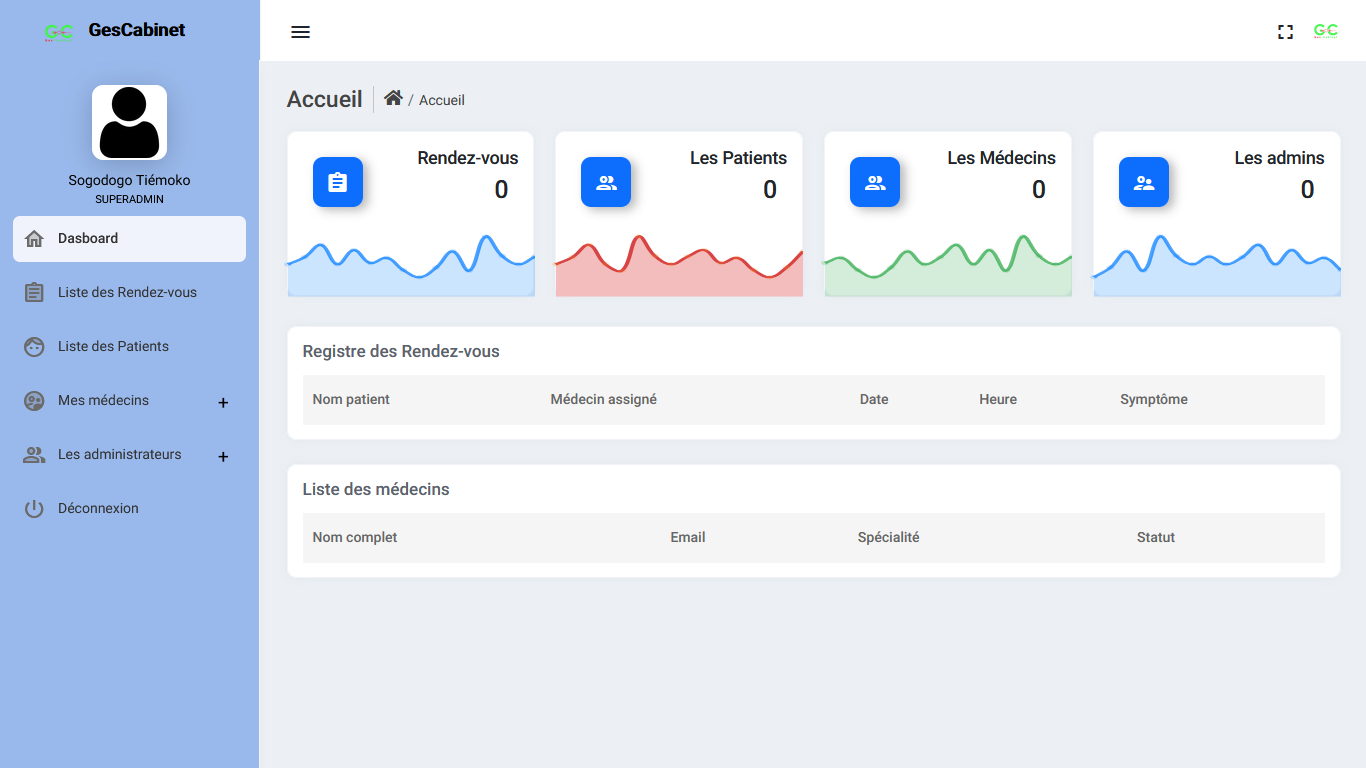
* Capture d’écran

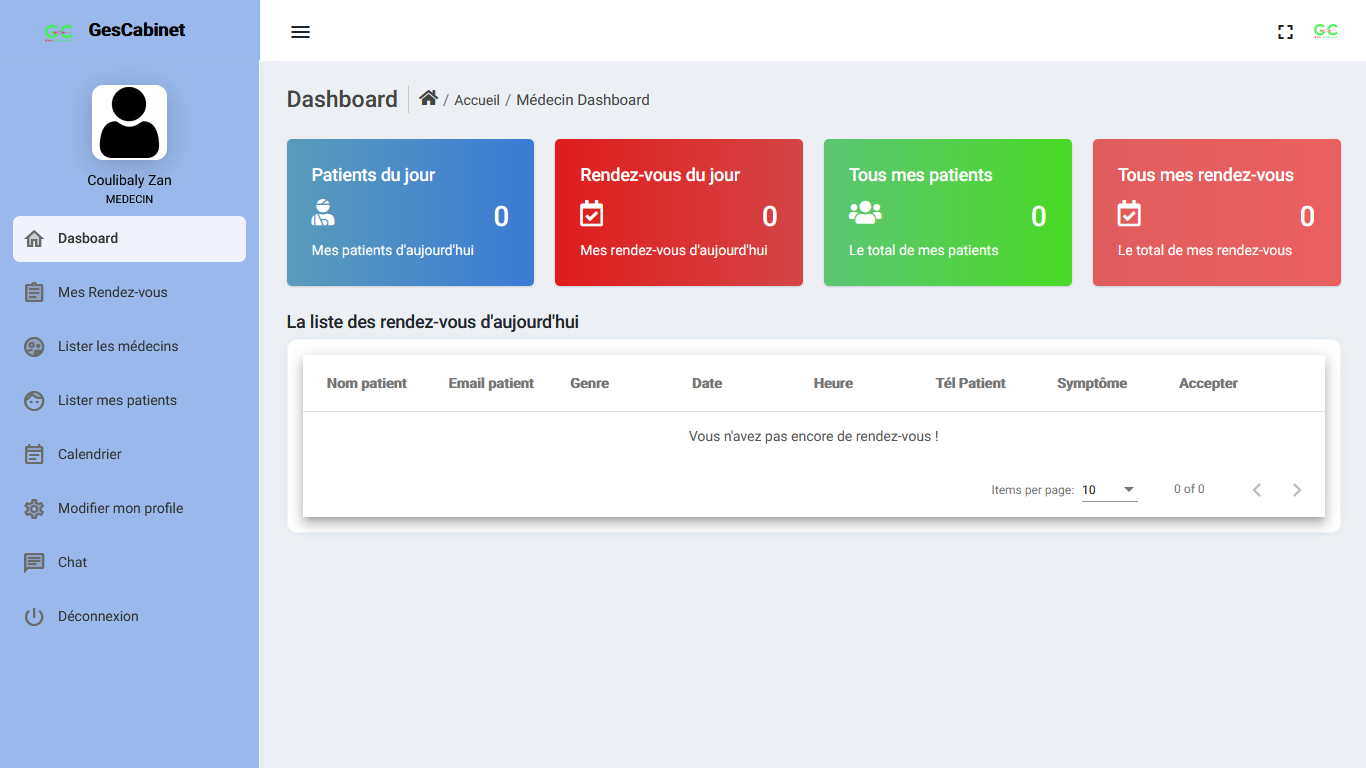


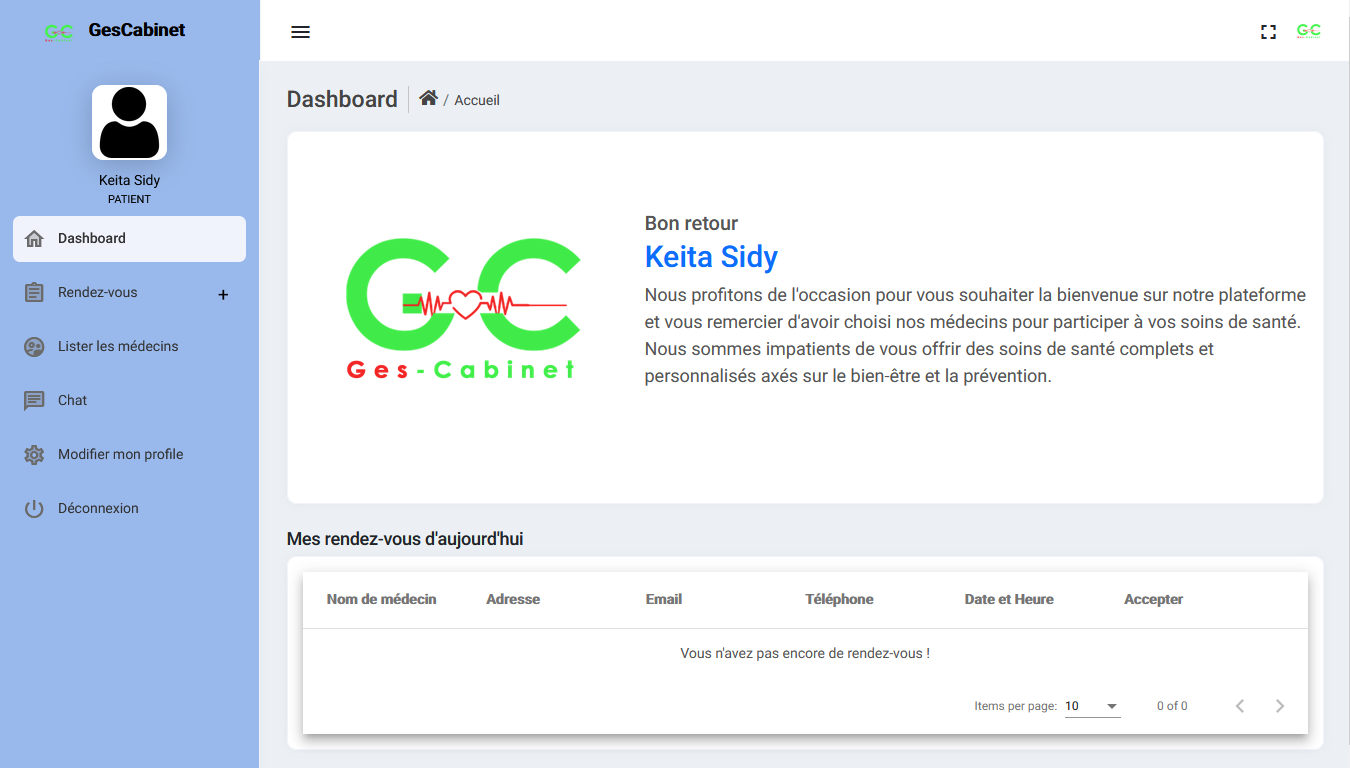










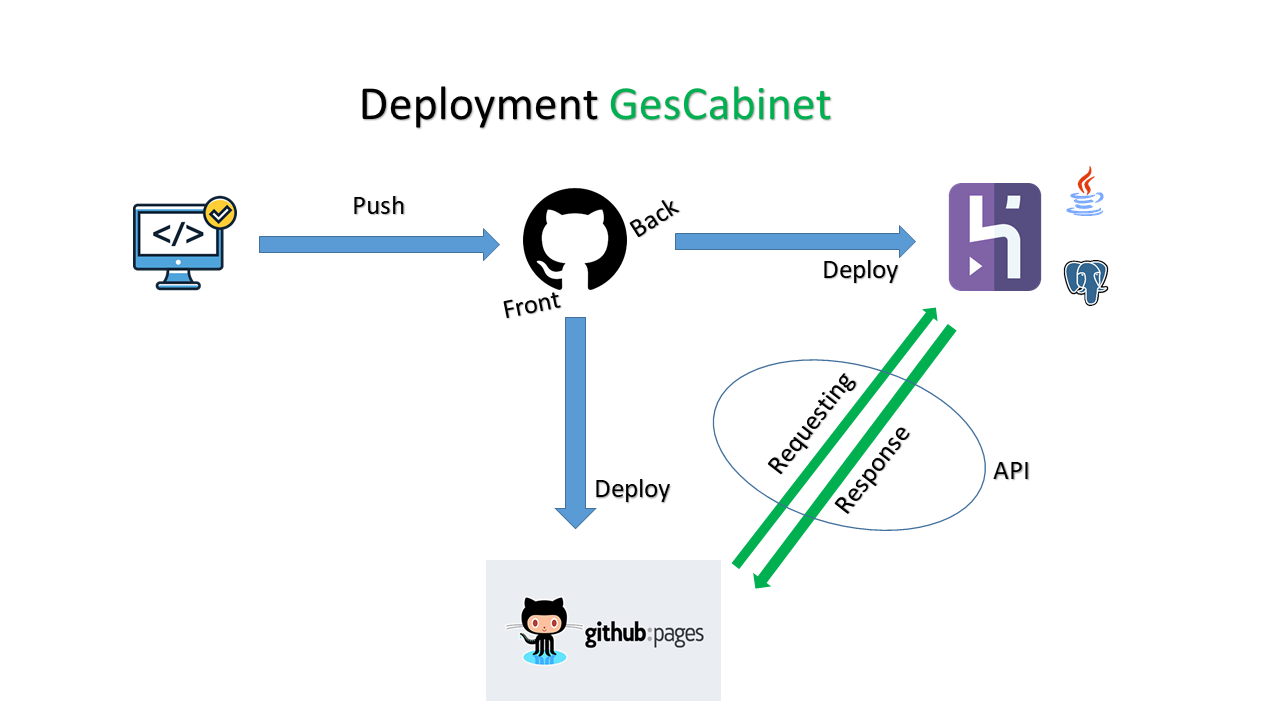


* Plan de déploiement

Comme moyen de déploiement, nous avons utilisé **Github Page** pour le front end et **Heroku** pour le Back end.

**Github Page** est un service de Github qui vous permet de publier un site web ou une application web à partir d’un projet GitHub. [gh-pages](https://www.npmjs.com/package/angular-cli-ghpages)

**Heroku** est une plate-forme en tant que service (PaaS) qui permet aux développeurs de créer, d'exécuter et d'exploiter des applications entièrement dans le cloud. [heroku](https://www.heroku.com)



* **Conclusion**
* Résumé du travail (apport personnel, utilité de l’application) et Perspectives

En conclusion, nous dirons qu’avec notre plateforme, la gestion des rendez-vous d’un cabinet médical serait de plus en plus facile.

Comme suggestion, il serait préférable de développer la partie mobile pour les patients et les médecins pour plus de faciliter.