Jérémy Salvadori
Bachelor Développeur PHP/Symfony
Session Juin 24

DOCUMENTATION TECHNIQUE

Tables des matières

1 - Réflexion préalable

a - Web App:

Symfony est un framework moderne intégrant nativement des fonctionnalités de sécurité avancées telles que l'authentification, l'autorisation, et la protection contre les attaques CSRF. Il comprend beaucoup de fonctionnalités natives et de dépendances compatibles dont notamment API Platform qui permet une mise en place simple et rapide d'une API qui sera nécessaire pour les parties Mobile et Desktop. Symfony utilise Doctrine comme ORM, facilitant grandement l'utilisation d'une base de données. Symfony dispose d'une communauté active et d'une documentation conséquente.
L'application est implémenté dans un container docker compose qui permet d'avoir une meilleure sécurité et une grande portabilité. Chaque conteneur fonctionne dans un environnement isolé, ce qui limite les dégâts qu'une application compromise peut causer à d'autres applications ou au système hôte.

b - Mobile App

Le but de l'application mobile étant de servir un maximum de médecins, il m'a semblé impératif que l'application soit compatible IOS et Android. Mon choix s'est porté sur Flutter qui utilise les mécanismes de sécurité native de ces 2 plateformes. Cela signifie que les applications Flutter bénéficient des mêmes protections que les applications natives écrites en Java (Android) ou Swift/Objective-C (iOS). Les applications Flutter sont compilées en code natif, ce qui leur permet d'offrir des performances élevées et une expérience utilisateur fluide. Il dispose également d'une large communauté et d'une bonne documentation

c - Desktop App

Python dispose d'une librairie customTkinter, basé sur tkinter, permettant d'obtenir une interface utilisateur graphiques simples et portables plus moderne que ce que propose tkinter. Les applications Python peuvent être exécutées sur différents systèmes d'exploitation, tels que Windows, macOS et Linux. Python dispose d'un large choix de bibliothèque, une large communauté et une documentation conséquente.

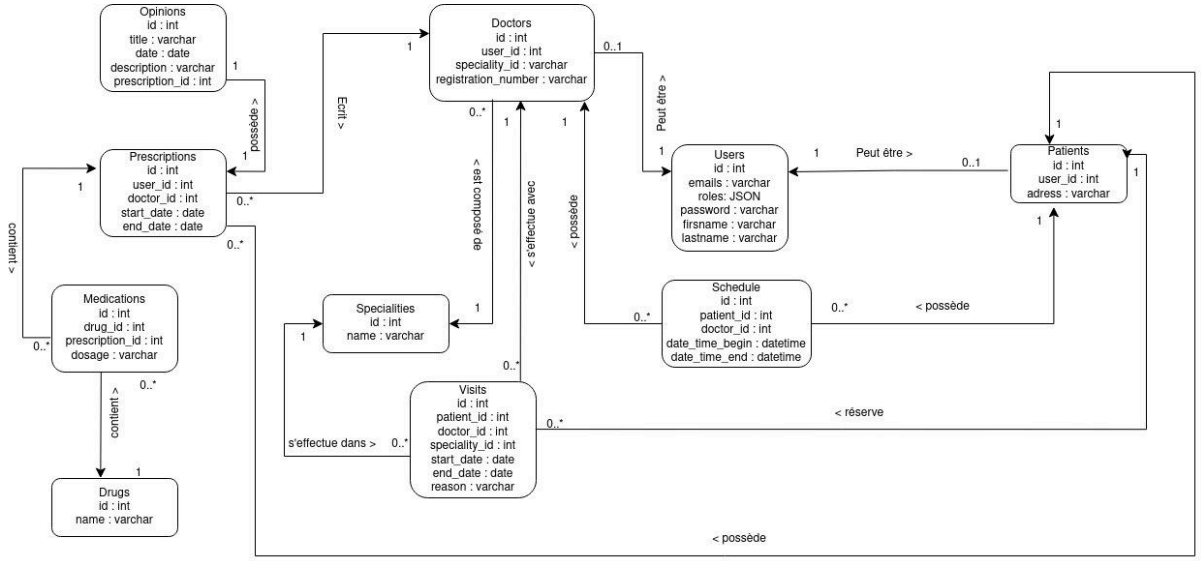
2 - Environnement de travail

Je travaille sur un système d'exploitation Linux, réputé pour sa stabilité et sa robustesse et disposant d'une large gamme d'outils de développement performant. L'installation d'outil supplémentaires peut se faire en grande majorité en simple ligne de commande, rapide et efficace.

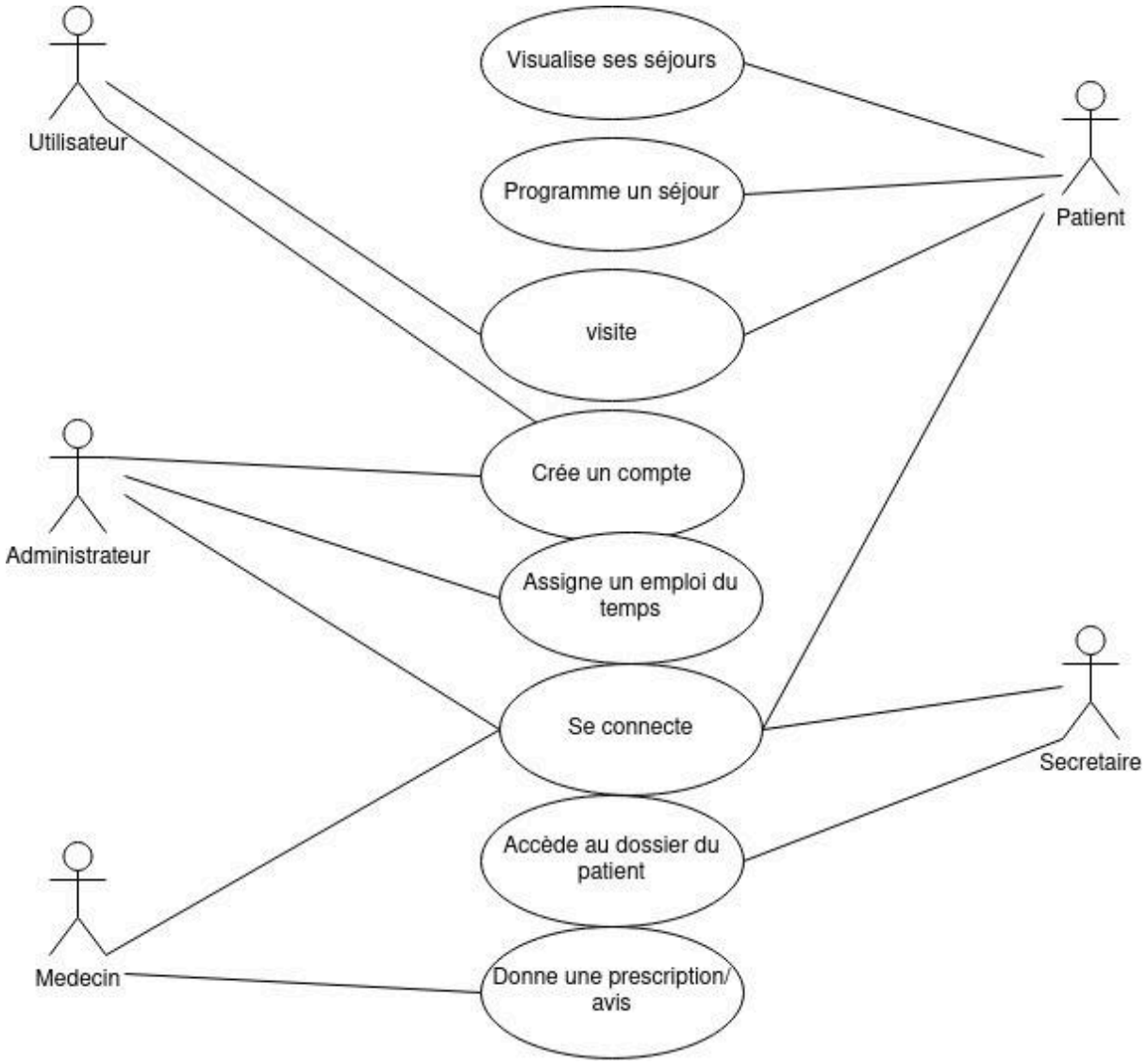
Je code avec les outils de JetBrains (PhpStorm, PyCharm) et Visual Code Studio pour la partie mobile. Ces outils disposent d'un large panel de fonctionnalités tel que l'aide à la saisie automatique, coloration syntaxique, refactorisation de code, etc. Ils disposent d'une grand nombre de plugins installables permettant d'ajouter des fonctionnalités supplémentaires à l'éditeur de code. Toutes ces fonctionnalités permettent de coder de manière plus rapide et efficace.

La gestion de version s'est faite sous Github et git, permettant de coder sous plusieurs branches différentes, et de fusionner celles-ci sur une branche de développement une fois les tests effectués.

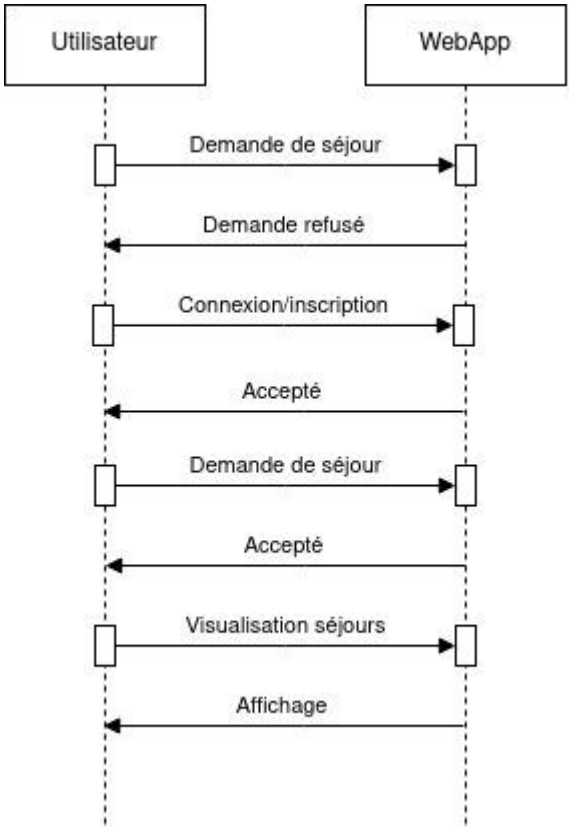
3 - MCD

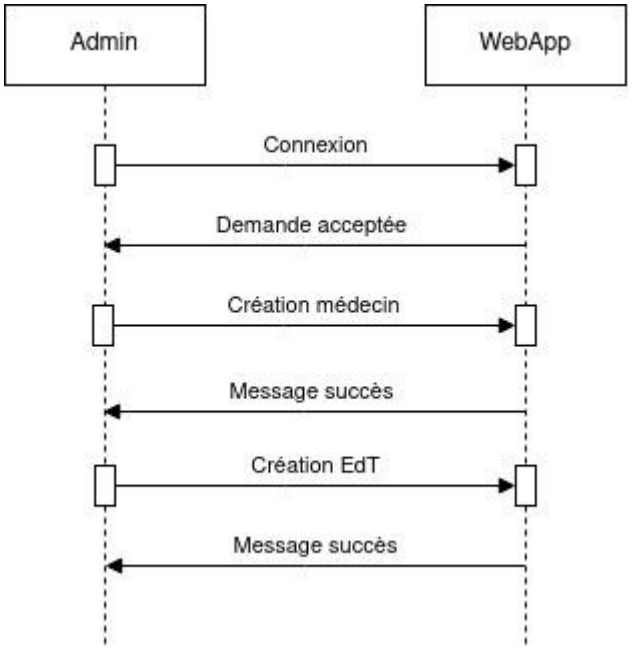

Opinions
id : int
title : varchar
date : date
description : varchar
prescription_id : int
Doctors
id : int
user_id : int
speciality_id : varchar
registration_number : varchar
Prescriptions
id : int
user_id : int
doctor_id : int
start_date : date
end_date : date
Users
id : int
emails : varchar
roles: JSON
password : varchar
firsname : varchar
lastname : varchar
Patients
id : int
user_id : int
adress : varchar
Medications
id : int
drug_id : int
prescription_id : int
dosage : varchar
Specialities
id : int
name : varchar
Schedule
id : int
patient_id : int
doctor_id : int
date_time_begin : datetime
date_time_end : datetime
Visits
id : int
patient_id : int
doctor_id : int
speciality_id : int
start_date : date
end_date : date
reason : varchar
Drugs
id : int
name : varchar
possède >
Ecrit >
Peut être >
Peut être >
contient <
< est composé de
< s'effectue avec
< possède
< possède
< possède
contient >
s'effectue dans >
< réserve
< possède

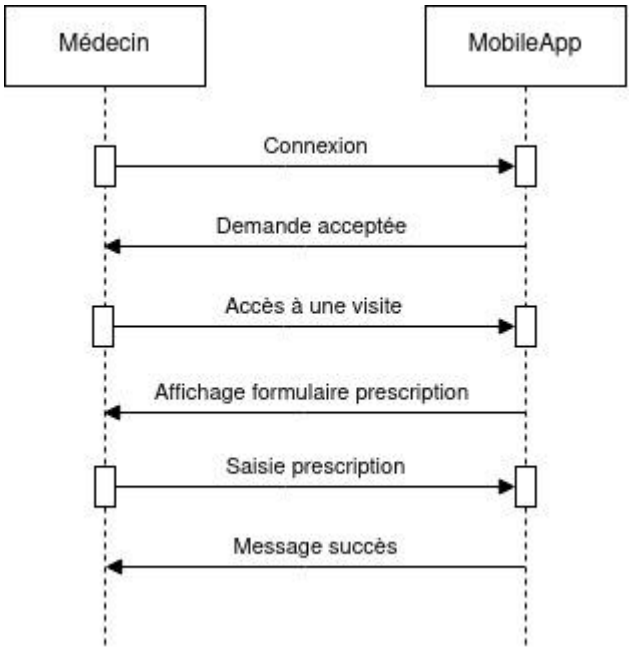
4 - Cas d’utilisations

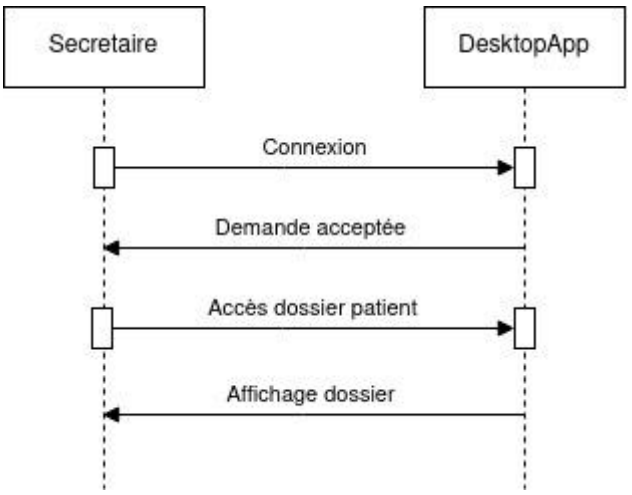


5 - Diagramme de séquences









6 - WireFrame et Mockup

Les fichiers sont visibles dans le dossier docs de la web app.

7 - Transaction SQL

Le fichier se trouve dans le dossier docs du projet web.

```
BEGIN;

-- Création de l'utilisateur et récupération de son id
INSERT INTO `users`(`email`, `roles`, `password`, `firstname`, `lastname`) VALUES ('doctor@soignemoi.com','["ROLE_DOCTOR"]','password','Pascal','Perrod');
SET @id_user := LAST_INSERT_ID();

-- Création d'une spécialité et récupération de son id
INSERT INTO `specialities`(`name`) VALUES ('Psychologie');
SET @id_speciality := LAST_INSERT_ID();

-- Création du docteur associé à l'utilisateur et à la spécialité précédemment crée
INSERT INTO `doctors`(`user_id`, `speciality_id`, `registration_number`) VALUES (@id_user, @id_speciality,'123456789');

COMMIT;
```

But de cette transaction : créer un docteur et une spécialité associée

On crée un utilisateur, on récupère son id que l'on set dans une variable. On fait de même avec une spécialité. On crée ensuite un docteur que l'on associe au user_id et speciality_id via les variable précédemment déclaré.

8 - Charte graphique

#007bff

#2b2b2b

9 - Reste à développer et bug

L'un des gros points restant à traiter est la validation des données, en particulier au niveau de leur cohérence. Par exemple, un séjour peut-être saisi par un patient pour une date déjà passée ou avec une date de fin antérieure à la date de début. Le choix du médecin doit être filtré par la spécialité dans le formulaire de programmation de séjour.

Sur la partie desktop, le fonction de récupération des données du patient ne permet pas de récupérer d'informations si le patient n'a pas de prescription/avis. Un correctif devra être apporté.

Sur la webApp, en parcours patient, avec un compte où aucun séjour n'a été saisi, on tombe sur une page blanche. Il faudra afficher un message pour plus de clarté.

L'app mobile à un bug d'affichage du logo selon la configuration de lancement.

L'environnement de test n'a pas encore été développé.