Projet 9 Mediscreen

## Retro sprint 1 :

### Ce qui s’est bien passé :

Une fois la problématique dans son ensemble compris j’ai construit assez aisément mon model,

Et la structure du projet 1 Repository Service Controller puis j’ai valider mon projet avec postman

Choix du framework front : angular et l’acheminement des donnéés

### Ce qui aurait pu aller mieux :

Gestion des erreurs, et Swagger

Une nouvelle version était sortie mais tous les tutos et ressource présentait encore l’ancienne façon de l’utiliser sa ma pris du temps.

Mais surtout docker un outil formidable mais un manque de maitrise de l’outils et manque de temps

### Ce que j’aimerais faire différemment :

Vraiment blinder au niveau de la gestion des erreurs la vérification des données entrées

Reliées mon dossier source au dossier docker afin de ne pas avoir besoin de régénéré le jar à chaque fois. La maitrise de la création de volume et de la configuration de MySQL je voulais qu’il puisse lire directement les données déjà enregistrer dans mon fichier SQL mais ce n’est pas le cas

## Retro sprint 2 :

### Ce qui s’est bien passé :

La mise en place du back était sensiblement la même

### Ce qui aurait pu aller mieux :

Découverte de mongo des base de donnés Nosql à approfondir

La gestion du cors Policy

### Ce que j’aimerais faire différemment :

Vraiment blinder au niveau de la gestion des erreurs la vérification des données entrées

Pareil la maitrise de docker sur ce projet, factoriser un peu plus mon code

## Retro sprint 3 :

### Ce qui s’est bien passé :

La structure back de l’application

### Ce qui aurait pu aller mieux :

La gestion du DTO et la vérification sur les données métiers

Docker compose et c’est nombreuse possibilités

### Ce que j’aimerais faire différemment :

Peaufiner bien plus la présentation du front, en ajoutant un système d’authentification

Docker compose a été une vrai problématique