

Cahier de charge:
Permis Vaccin

Pascal Bourgoin

Historique du document

Auteur(s)	Version	Date	Description
Pascal Bourgoïn	V01	24/02/2021	Début du cahier de charge et ajout des objectifs du projet, des diagrammes de classes et de Use cases.
Pascal Bourgoïn	V02	31/03/2021	Non complet – Ajout des services et des tests de l’application mais changement nécessaire au diagramme de usecases et de classes. Il reste aussi a ajouté un tableau des données insérées dans le ministèreWs.
Pascal Bourgoïn	V03	05/04/2021	Changement des diagrammes encore une fois et ajout de remarque.
Pascal Bourgoïn	V04	07/05/2021	Modification de tous les aspects du projet pour avoir un livrable final de qualité.
Pascal Bourgoïn	V05	02/06/2021	Ajout de modification ajouté à la version finale.

Table des matières

Finalité du projet	3
Besoin client :	3
Objectif :	3
Exigences techniques :	3
Diagrammes des Usecases :	4
Diagrammes de classes :	5

Finalité du projet

Besoin client :

Étant donné la situation sanitaire actuelle, un client, le ministère de la santé, vous sollicite pour le développement d'une application concernant la création et la diffusion d'un permis de santé COVID.

▪ C'est quoi un permis de santé COVID ?

C'est un code unique délivré et envoyé par le ministère de la santé aux personnes ayant :

- Soit reçu les 2 doses vaccin contre le virus : « permis vaccin ».
- Soit reçu un test négatif contre le virus durant les dernières 72 heures : « permis test ».

Objectif :

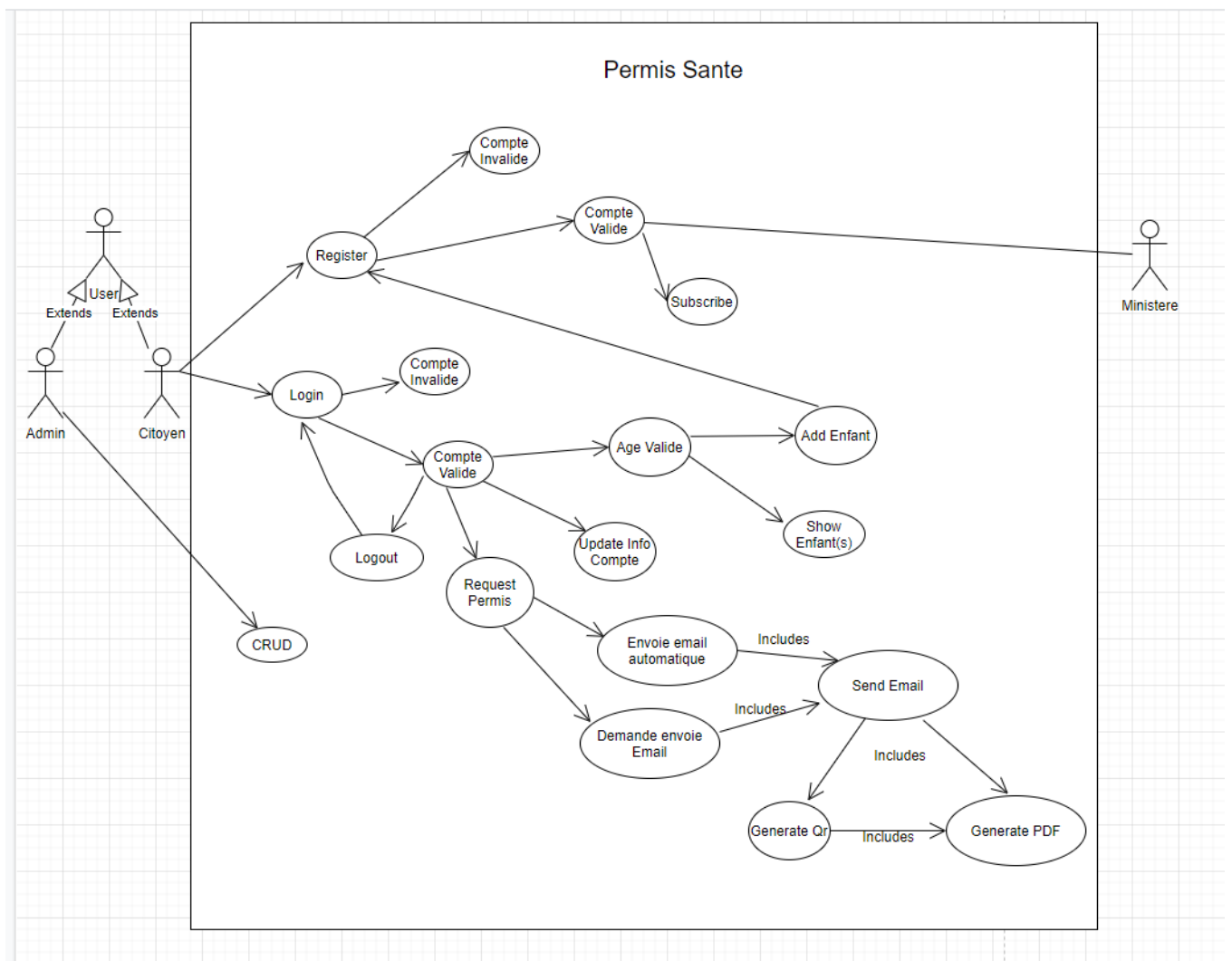
Créer un programme FullStack qui permettra de réussir les besoins du client.

Exigences techniques :

- SpringBoot et le JPA
- JUnit 5
- SQL (insertion)
- Java 8

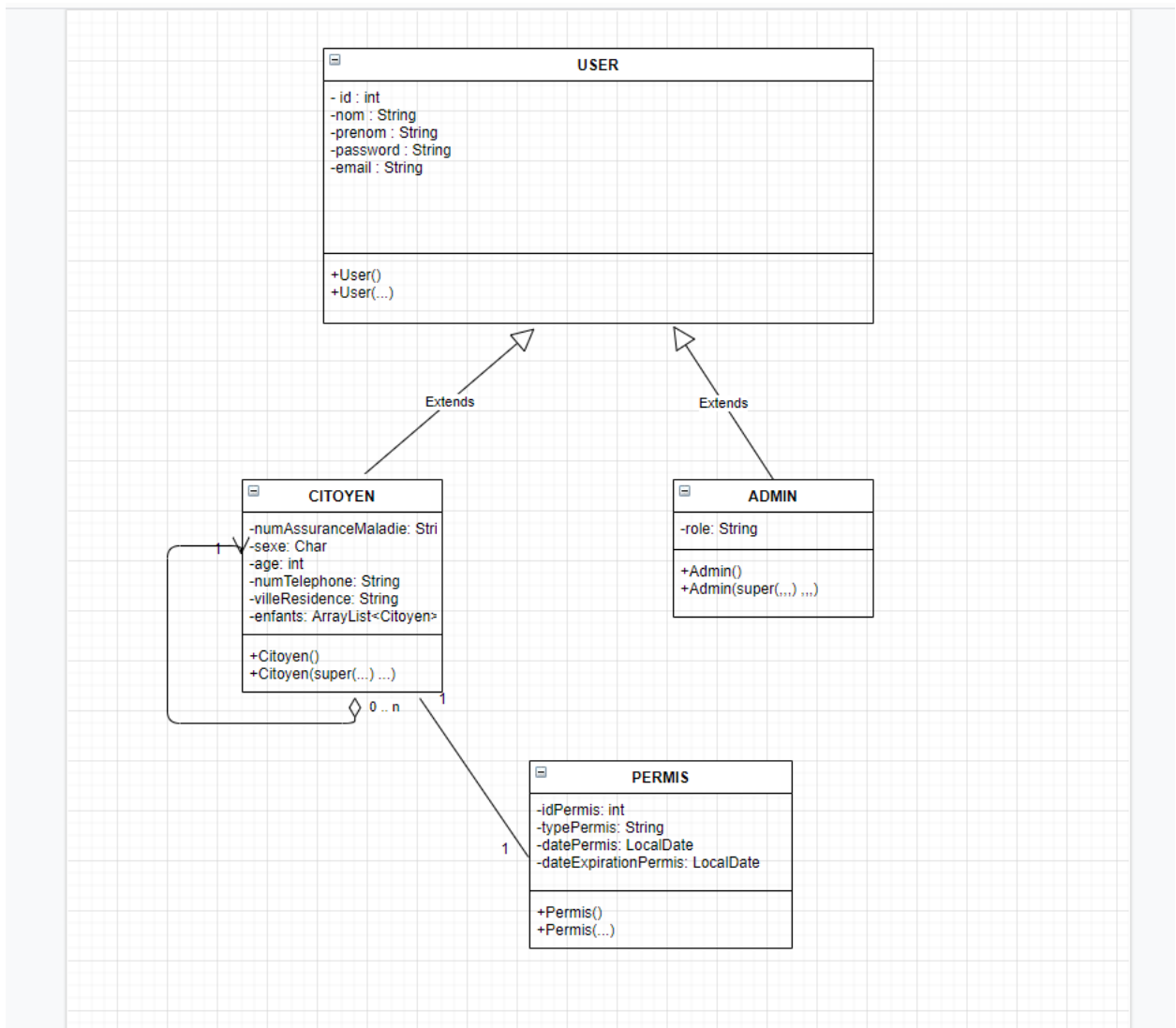
Remarque : Les exigences sont limitées pour le moment mais il y aura des ajouts à travers les étapes du projet.

Diagrammes des Usecases :



Modification du V04 : Diagramme beaucoup plus détaillé et qui décrit beaucoup mieux les usecases. Celui-ci, est un mapping générique du programme et de son fonctionnement actuel.

Diagrammes de classes :



Remarque : Les getter et setter ne sont pas dans le UML car ils sont visibles dans le code sources qui est fourni avec la remise.

Modification du V04 : Enlever la l'attribut login puisque l'email agit comme le username de login. En plus, l'ajout de l'attribut enfants dans la classe citoyen, celle-ci est une liste de citoyen qui contiendrait tous les enfants d'un citoyen.

Usecase	Repository	Service	Test Unitaire
Login	CitoyenRepository	AppService	testLogin
Créer compte enfant	CitoyenRepository	AppService	testAddEnfant
Send email	CitoyenRepository	AppService	
Generate QR		AppService	testGenerateQr
Generate PDF		AppService	testGeneratePdf

Modification majeure du V4 :

- Ajout de plusieurs valeurs de retour sur des méthodes principales pour s'assurer du bon fonctionnement.
- Changement des paramètres de certaines fonctions pour mieux compléter le programme et rendre le programme plus efficace et avoir du low binding.
- Réparation de certains bugs ou certains éléments inutile pour le mieux du programme.
- Ajout du controller pour faire appelle au service à partir du frontend, modification de certaines méthodes pour faciliter leur utilisation avec le front end.

Conclusion sur le développement pour la phase 03 :

Conception bonne, il y avait certaine chose à modifier pour le mieux de l'application mais la conception primaire des idées et du fonctionnement de l'application était toutefois bonne et a permis un développement plus facile, car tout ne devait pas être changer.

Méthode du rest controller et leur utilisation en usecase :

Méthode rest controller	Usecase
Login()	login
LoginExist()	-----
Subscribe()	Register, subscribe
requestPermis()	requestPermis, send email
sendEmail()	Send email
addEnfant()	Ajouter Enfant
qrCode	Va chercher le qrcode par le controller

Ajout dans version finale :

-Ajout d'une méthode dans le rest controller pour aller chercher le code QR par l'appelle du service.

-Changement de la façon pour ajouter des enfants au parent.

-Ajout du admin qui permet de voir la liste de permis et la liste de citoyen et de modifier et supprimer un citoyen ainsi que de supprimer un permis, tout à l'aide de thymeleaf.

-Changement dans le repository de citoyen dans ministère web service pour que la recherche de numAssMal ne soit pas case sensitive.