



Cégep **André-Laurendeau**

Cégep andré-laurendeau

# TP4

Cahier des charges

Émerick Poulin  
29/05/2021

## Historique du document

Auteur(s)	Version	Date	Description
Émerick Poulin	V01	24/02/2021	Système de persistance des données
Émerick Poulin	V02	31/03/2021	Création des services
Émerick Poulin	V02.1	06/04/2021	Restructuration complète
Émerick Poulin	V3.1	07/05/2021	Site-web et contrôleur côté serveur
Émerick Poulin	V4.0	29/05/2021	Application administrative

## Table des matières

Historique du document.....	2
Concept et objectifs du projet .....	2
1. Finalité du projet.....	2
1.a. Contexte métier du projet .....	Error! Bookmark not defined. 1.b.
Objectifs du projet .....	Error! Bookmark not defined.
Glossaire de tous les noms, acronymes et abréviations utilisés dans le document. ....	Error! Bookmark not defined.

## Concept et objectifs du projet

### 1. Finalité du projet

Le projet consiste au développement d'une application de gestion des permis de vaccination et de tests dans le cadre la crise sanitaire globale connu sous le nom de COVID-19.

Cette application gère la distribution numérique, l'autorisation d'obtention et la validité des permis en communiquant avec le système du ministère de la santé

### 2. Exigences techniques

- Base de données H2
- Framework SpringBoot
- SpringBootTest Data Jpa

- SpringBootStarter Data Rest
- SpringBootStarter Validation
- SpringBootStarter mail
- SpringBootStarter web
- SpringBootStarter Thymeleaf (Thymeleaf framework)
- Junit 4.12
- Zxing 3.4.1
- Itextpdf 7.1.7
- Zxing-javase 3.3.0
- Librairie Lombok
- Java 11
- Angular Framework
- Bootstrap (webjar)
- font-awesome (webjar)

### 3. Diagramme de use case

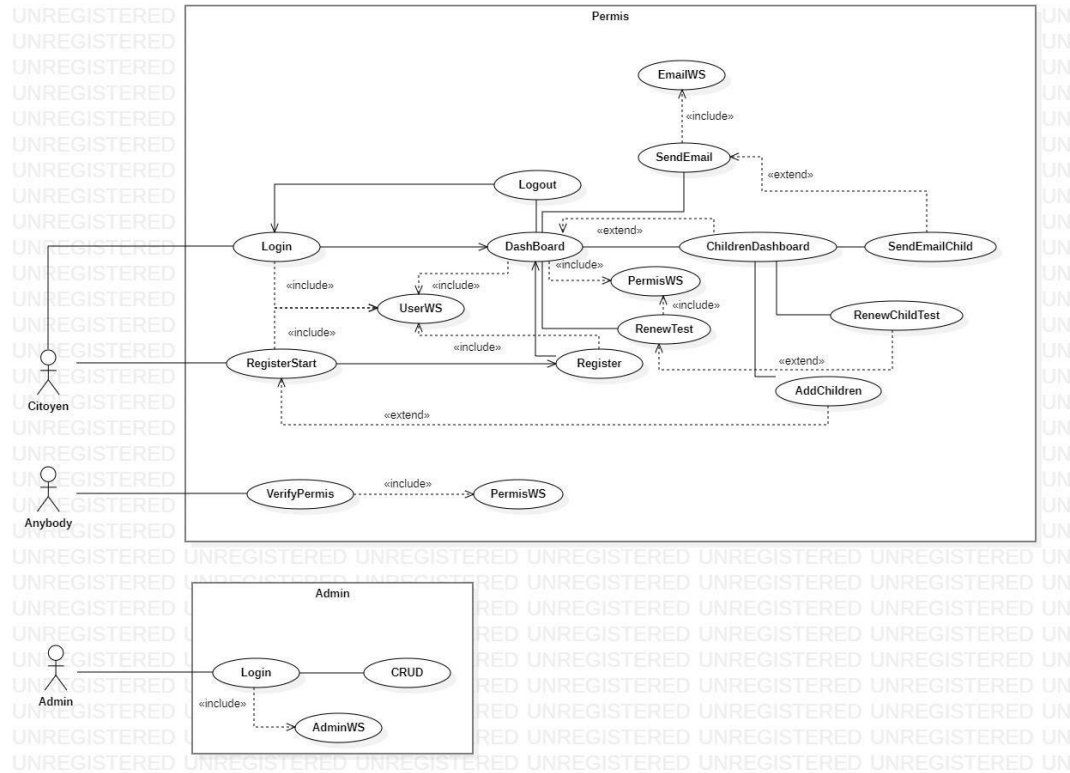
Les méthodes spécifiques utilisées par les uses case incluant les web services ont été omises à des fins de clarté

Explication de la vérification de validité des permis sous forme QR :

Les codes QR générés par l'application contiennent tous un URL sous la forme <http://webiste.abc/permis/verify/hash>

Le « hash » contenu dans l'url est un SHA-256 unique à chaque permis généré. Cela a pour avantage de permettre à n'importe qui doté d'un appareil mobile avec des capacités de lecture pour les codes QR de vérifier la validité d'un permis.

À l'intérieur la page web de vérification envoie les hash à un web service qui récupère le permis avec le hash donné et retourne l'état de validité à la page web pour un affichage en conséquence.



#### 4. Diagramme de classes primaire

Les getters/setters ont été omis pour des raisons de clarté dans le diagramme

Les getters/setters sont générés par la librairie lombok : ils ne sont pas visibles dans le code source

Les exceptions « custom » ont été omises pour ne pas encombrer le diagramme

Une image au format JPEG est fournie avec ce document pour permettre une visualisation plus claire



SendMail		EmailService PDFService QRService	testQRCodeEmail()
----------	--	---	-------------------

## 6. Système de gestion des erreurs

Du côté « front-end » un service a été mis en place pour faire la gestion automatique des erreurs retournés par le serveur.

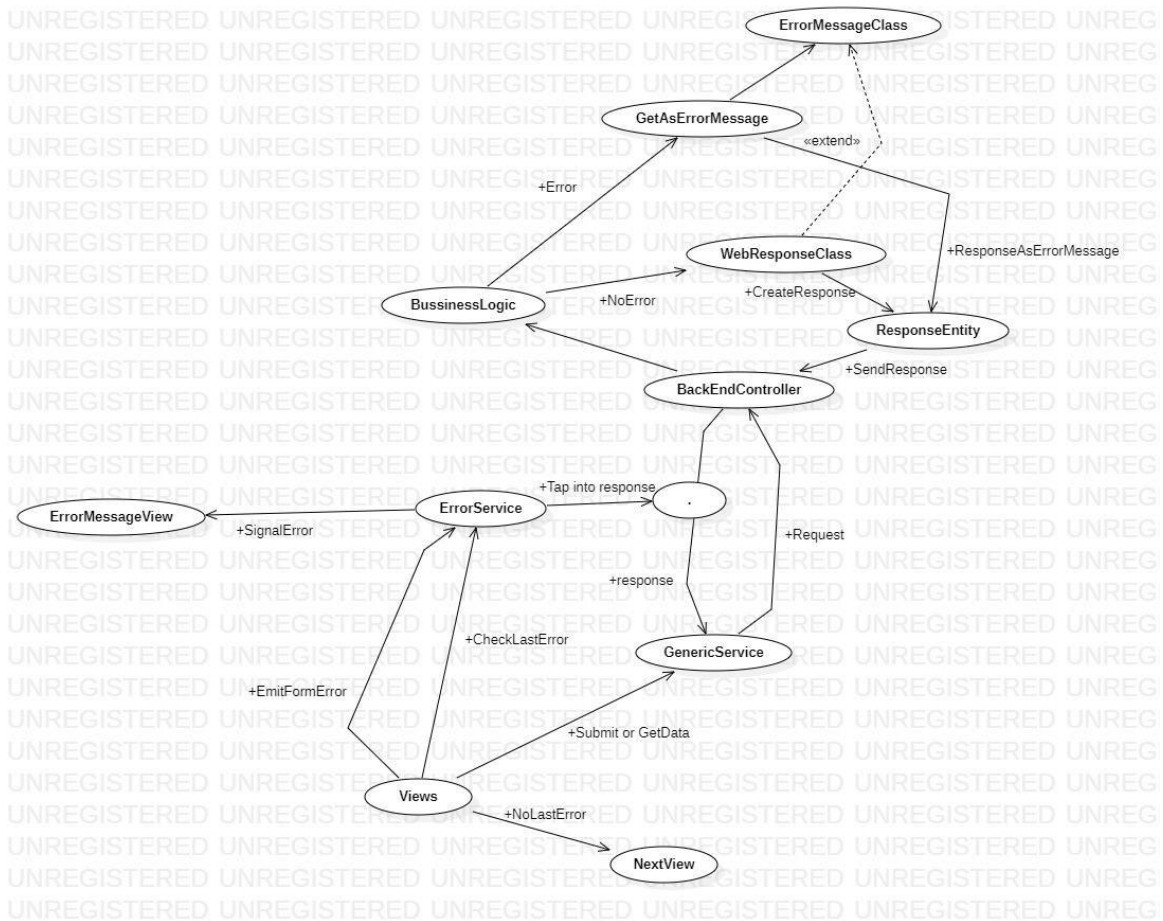
Un service gère les erreurs en surveillant l'état des réponses reçu du serveur.

L'affichage se fait par le biais d'un composant. Celui-ci reçoit un événement lors de la détection d'une erreur dans la dernière réponse reçue d'un des services. Cela permet de faire l'affichage des erreurs qu'en insérant la balise du « selector » à l'endroit prévu pour l'affichage.

Ensuite les composantes peuvent vérifier s'il y a un état d'erreur pour décider s'il est OK de poursuivre à la prochaine page/étape.

Pour le « back-end » la gestion des erreurs se fait sous la forme d'héritage d'une classe nommée « ErrorMessage » en héritant de celle-ci, il est possible de retourner au web un objet contenant un message d'erreur à être afficher à l'utilisateur sans avoir à changer le type de l'objet.

Le diagramme suivant n'est pas un diagramme de use case. L'absence de diagramme général a fait en sorte que ce type de diagramme était le plus facile à utiliser dans ce contexte



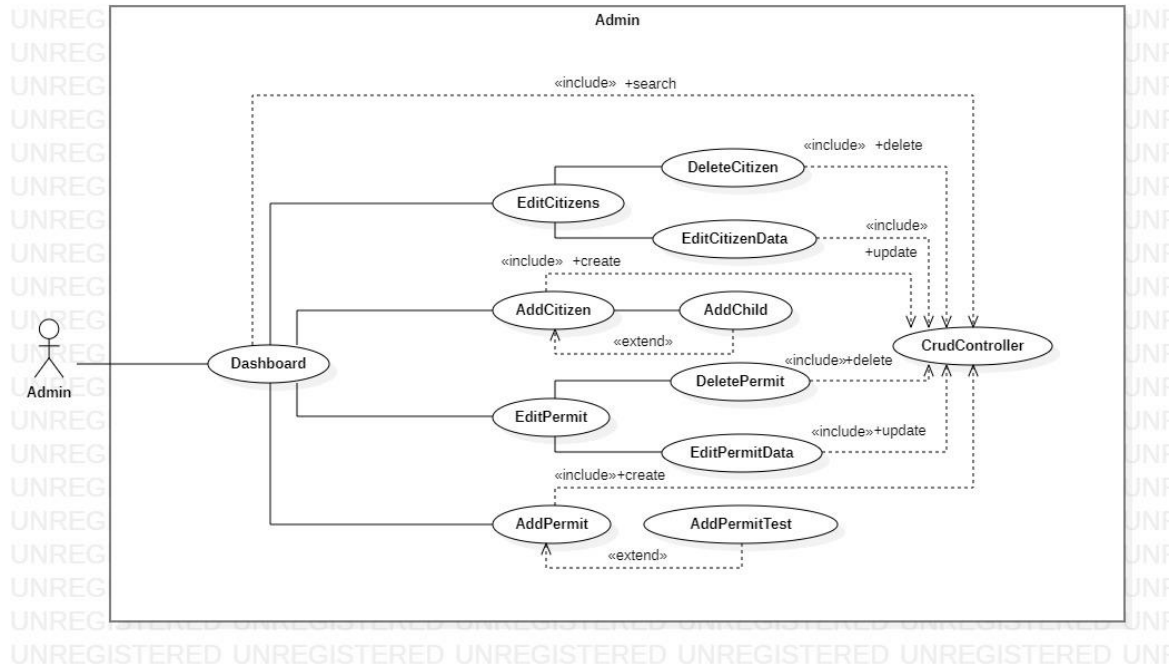
## 6. Application administrative

Une application administrative a été développée pour répondre au besoin de gestion.

Cette application repose sur le principe « CRUD ». Cet acronyme est représenté les opérations communes dans le contexte d'interaction avec une base de donnée.

Présentant une interface web au moyen du framework Thymeleaf, l'application permet la recherche, modification, l'ajout et la suppression des permis ainsi que des comptes pour l'application de distributions des permis.

Suivant est le diagramme de cas d'utilisation pour cet application administrative.



À noter que certaines méthodes « héritées » du parent, par exemple « create » dans le cas d'utilisation « AddPermitTest », n'ont pas la même implémentation en raison d'un problème de gestion des objets polymorphiques.