



# Check-List Manager

Dossier de conception fonctionnelle

Version 1.0

**Auteur**

Le Magorou Jean-Martial

# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>2</b>
<b>2 - Introduction.....</b>	<b>3</b>
2.1 - Objet du document.....	3
2.2 - Références.....	3
2.3 - Besoin du client.....	3
2.3.1 - Contexte.....	3
2.3.2 - Enjeux et Objectifs.....	4
2.3.3 - Contraintes.....	4
<b>3 - Description générale de la solution.....</b>	<b>5</b>
3.1 - Les principes de fonctionnement.....	5
3.2 - Les acteurs.....	5
3.3 - Les cas d'utilisation généraux.....	6
3.3.1 - Le diagramme de contexte.....	7
3.3.2 - Le diagramme de packages.....	7
<b>4 - Les workflows.....</b>	<b>8</b>
4.1 - Le workflow.....	8
<b>5 - Application Web.....</b>	<b>9</b>
5.1 - Les acteurs.....	10
5.2 - Les cas d'utilisation.....	10
5.2.1 - Package "Gestion de commande".....	10
5.2.1.1 - Cas d'utilisation.....	10
5.2.2 - Package "Livrer".....	14
5.2.2.1 - Cas d'utilisation.....	14
5.2.3 - Package "Gestion administrative".....	16
5.2.3.1 - Cas d'utilisation.....	16
5.3 - Les règles de gestion générales.....	16
5.3.1 - Le client.....	16
5.3.2 - OC-Pizza.....	16
<b>6 - Glossaire.....</b>	<b>17</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
JMLM	25/09/20	Création du document	V0.1
JMLM	29/09/20	Ajout descriptions textuelles	V 0.2

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le volet « conception fonctionnelle » de la réponse au P13 du parcours Développeur d'Applications Python d'OPENCLASSROOMS.

Il a pour but de présenter les différents acteurs et fonctionnalités afin de :

- Définir des règles de gestion fonctionnelles
- Détailler les processus de création et de saisie des check-lists ainsi que la gestion des utilisateurs.
- Choisir une solution technique

Les éléments du présent dossiers découlent :

- de l'expression du besoin
- de l'analyse complète du besoin suite à mon expérience personnelle récente.

### 2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **P13 – Dossier Conception Technique** : Dossier de conception technique de l'application
2. **P13 – Dossier Exploitation**: Dossier d'exploitation de l'application

### 2.3 - Besoin

#### 2.3.1 - Contexte

Lors de ma précédente expérience professionnelle j'ai été confronté à des problèmes d'organisation sur des tâches répétitives à effectuer. Le travail était pour partie un travail de contrôles réglementaires d'appareils de levage, les outils disponibles pour cela étaient peu souples. Qu'est-ce qu'un contrôle sinon une liste de case à cocher, que l'on doit pouvoir ressortir en cas de soucis avec le matériel contrôlé.

Dans notre vie personnelle nous avons tous été confrontés à des problèmes de liste que nous n'avons pas tenue (un simple liste de courses par exemple). C'est pourquoi j'ai décidé de couvrir ce manque pour le P13 du parcours DA Python.

### 2.4 - Enjeux et Objectifs

CheckListManager doit pouvoir couvrir le besoin de check-lists de 2 types d'utilisateur différents :

**Le particulier :**

- Doit pouvoir créer son propre compte.
- Ne doit pas pouvoir visualiser/modifier les check-lists des autres utilisateurs.
- Doit pouvoir créer/saisir/visualiser ses propres check-lists pour lui seul (la notion de client ne doit pas être couverte).

#### **Le professionnel :**

- Doit pouvoir créer des comptes de techniciens qui ne pourront que saisir/visualiser les check-lists de ses collègues (notions de société).
- Doit pouvoir créer/modifier ses propres check-lists.
- Doit pouvoir créer ses propres clients (notion de client – check-list pour le compte de...)

## **2.5 - Contraintes**

Check-list Manager devra répondre aux besoins des particuliers et des professionnels sans réécriture de code.

La cohabitation entre les 2 types d'utilisateur devra être possible sur un même serveur sans qu'il n'y ait d'interaction entre eux.

Enfin, le fait que le système soit directement lié au client via la plateforme Internet impose au système d'être modulable afin de pouvoir rajouter facilement de nouvelles fonctionnalités si besoin sans avoir besoin de tout refaire.

# **3 - DESCRIPTION GÉNÉRALE DE LA SOLUTION**

## **3.1 - Les principes de fonctionnement**

L'application sera une application web dite responsive. C'est-à-dire qu'elle s'adaptera automatiquement à la taille de l'écran de l'utilisateur.

Toutefois, la partie administrative ne sera utilisée que sur un écran de taille correcte (portable au moins) la création de check-list ne pouvant se faire sur un écran trop petit.

Le remplissage des check-lists pourra se faire sur un écran de smartphone sans poser de souci d'ergonomie.

Pour ce qui est de la création/gestion des utilisateurs et de droits, nous partons des principes suivants :

- Chaque environnement est cloisonné et indépendant des autres. Aucune information n'est partagée.
- Les professionnels sont liés à une société qui sera créée par l'Administrateur Site alors que lors de son « auto-enregistrement », la société liée à un particulier sera son automatiquement son id. Lors de l'enregistrement le flag « Pro » est mis a True ou False afin de facilement distinguer la catégorie d'utilisateur tout au long du parcours utilisateur.
- Un particulier ne peut pas gérer de client, car il ne peut faire de check-list pour le compte de (pas de notion de client).





Pour ce qui est des check-lists les principes suivants seront appliqués :

- Les check-lists saisies et validées seront stockées sur le serveur sous forme de fichier PDF. Elles ne seront donc plus modifiables ou supprimables.
- Les clients ne sont pas supprimables

## **3.2 - Les acteurs**

Les acteurs sont une entité qui interagit avec l'application. Les acteurs dits principaux interagissent directement avec elle. Il s'agit donc ici des utilisateurs. Les acteurs dits secondaires

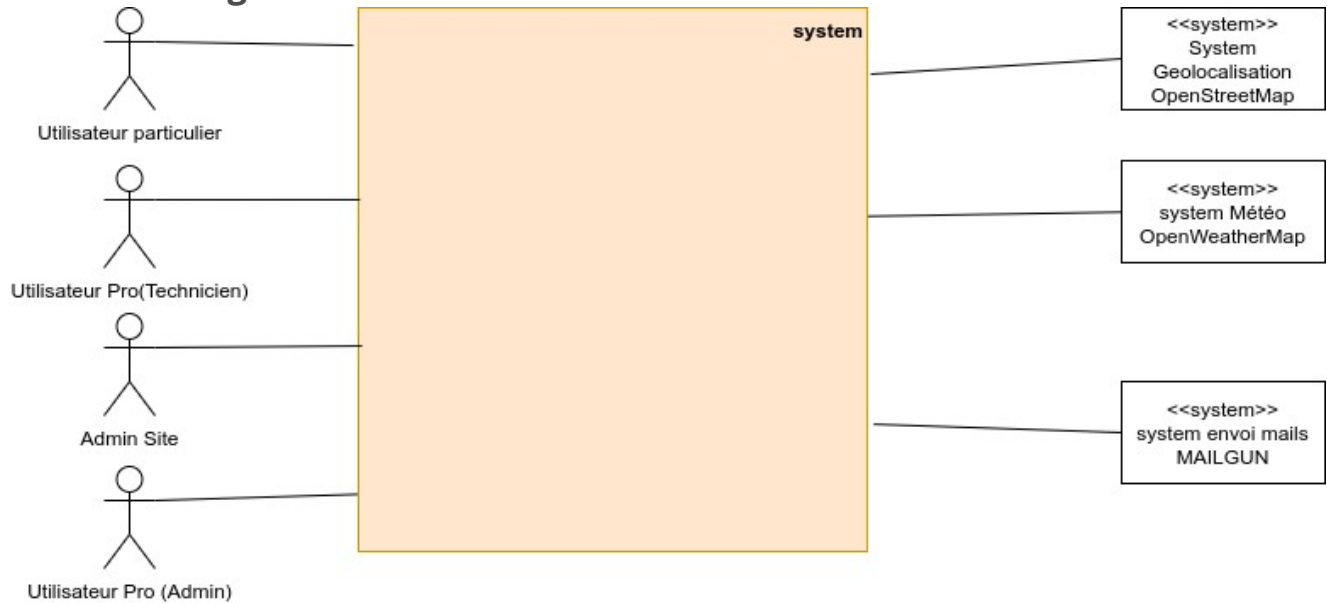
n'interagissent pas directement avec l'application, mais y sont liés au travers des données ou des fonctions... Il s'agira ici de MAILGUN pour l'envoi de mails, du système OpenStreetMap pour la géolocalisation et de OpenWeatherMap pour l'affichage de la météo.  
Le tableau ci-dessous nous montre rapidement les acteurs principaux de l'application avec leurs rôles respectifs.

Nom - Représentation	Acteur	Veut pouvoir faire
 Professionnel Technicien	Utilisateur Technicien	<ul style="list-style-type: none"> <li>- Se connecter.</li> <li>- Remplir une check-list de sa société.</li> <li>- Créer un client pour sa société.</li> <li>- Visualiser les check-lists de sa société.</li> </ul>
 Mehdi – Professionnel Admin	Professionnel Admin	<ul style="list-style-type: none"> <li>- Se connecter.</li> <li>- Créer/modifier les check-lists.</li> <li>- Remplir une check-list de sa société.</li> <li>- Créer les comptes Techniciens de sa société</li> <li>- Créer un client pour sa société.</li> <li>- Visualiser les check-lists de sa société</li> </ul>
 Utilisateur particulier	Utilisateur Particulier	<ul style="list-style-type: none"> <li>- Se connecter.</li> <li>- Créer/modifier ses check-lists.</li> <li>- Remplir/visualiser ses check-lists.</li> </ul>
 Admin Site	Admin Site	<ul style="list-style-type: none"> <li>- Tout ce que peuvent faire les autres.</li> <li>- Créer les comptes admin professionnels.</li> <li>- Créer les sociétés.</li> </ul>

### 3.3 - Les cas d'utilisation généraux

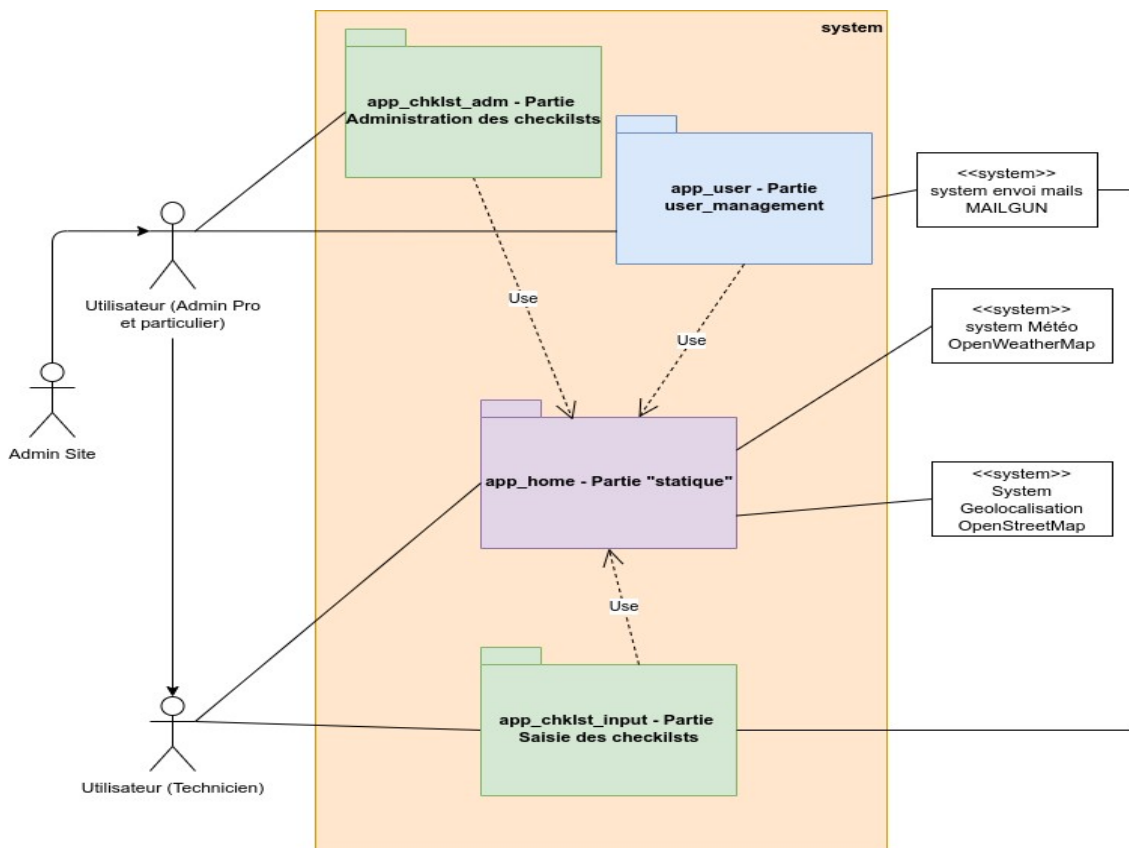
Les 2 diagrammes ci-dessous expliquent le contexte global de Check-list Manager ainsi que la découpe de ce même contexte en parties distinctes. Cela permet d'appréhender l'objectif global en plusieurs grandes parties qui seront-elles mêmes découpées en entités plus petites.

### 3.3.1 - Le diagramme de contexte



Nous avons donc les 4 types d'utilisateurs (acteurs) ainsi que les 3 acteurs externes cités plus haut. Nous pouvons voir que les utilisateurs sont classés en 2 types distincts : les particuliers et les professionnels. En effet, la différence entre eux en termes de possibilité est trop importante pour ne pas les séparer.

### 3.3.2 - Le diagramme de packages



Le système est donc découpé en 4 packages :

- Package « statique » : La partie « statique » du site accessible par tout internaute désirant en savoir plus sur Check-list Manager.
- Package « User management » : Concerne toute la gestion des utilisateurs du site ainsi que les droits des uns et des autres.
- Package « Administration des check-lists » : Concerne la création/modification des check-lists ainsi que des clients.
- Package « Saisie des check-lists » : Concerne la saisie/visualisation des check-lists.

La partie « check-lists » a été découpée en 2 packages, car les droits concernant cette partie sont complètement différents : Un technicien (professionnel) ne peut pas créer ou modifier une check-list.

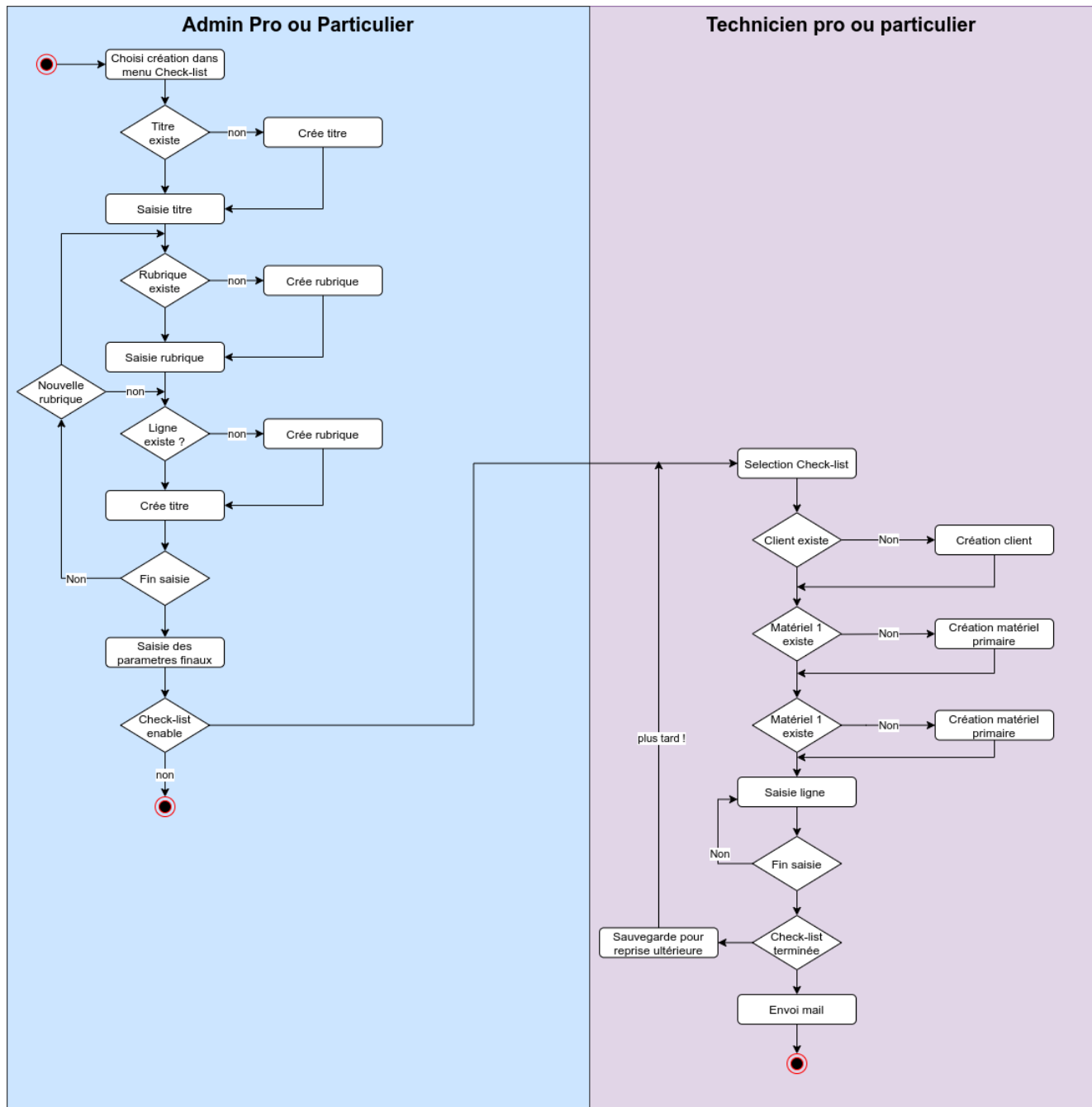
## 4 - LES WORKFLOWS

### 4.1 - Le workflow

Le diagramme d'activité ci-dessous permet d'avoir une vision d'ensemble du cycle de vie d'une



check-list. Nous traiterons la création et la saisie et partirons du postulat que chacun dispose d'un compte approprié et est connecté au système.



## 5 - APPLICATION WEB

L'application est donc une application web adaptative. C'est-à-dire qu'elle sera accessible et utilisable de n'importe où et de n'importe quel matériel. Un simple navigateur suffira. L'application restera donc la même quel que soit le matériel. Par contre, les packages administration des check-lists et user\_management ne sont pas optimaux sur un smartphone.

## 5.1 - Les acteurs

Les acteurs sont donc les mêmes que ceux présentés en 3.2.

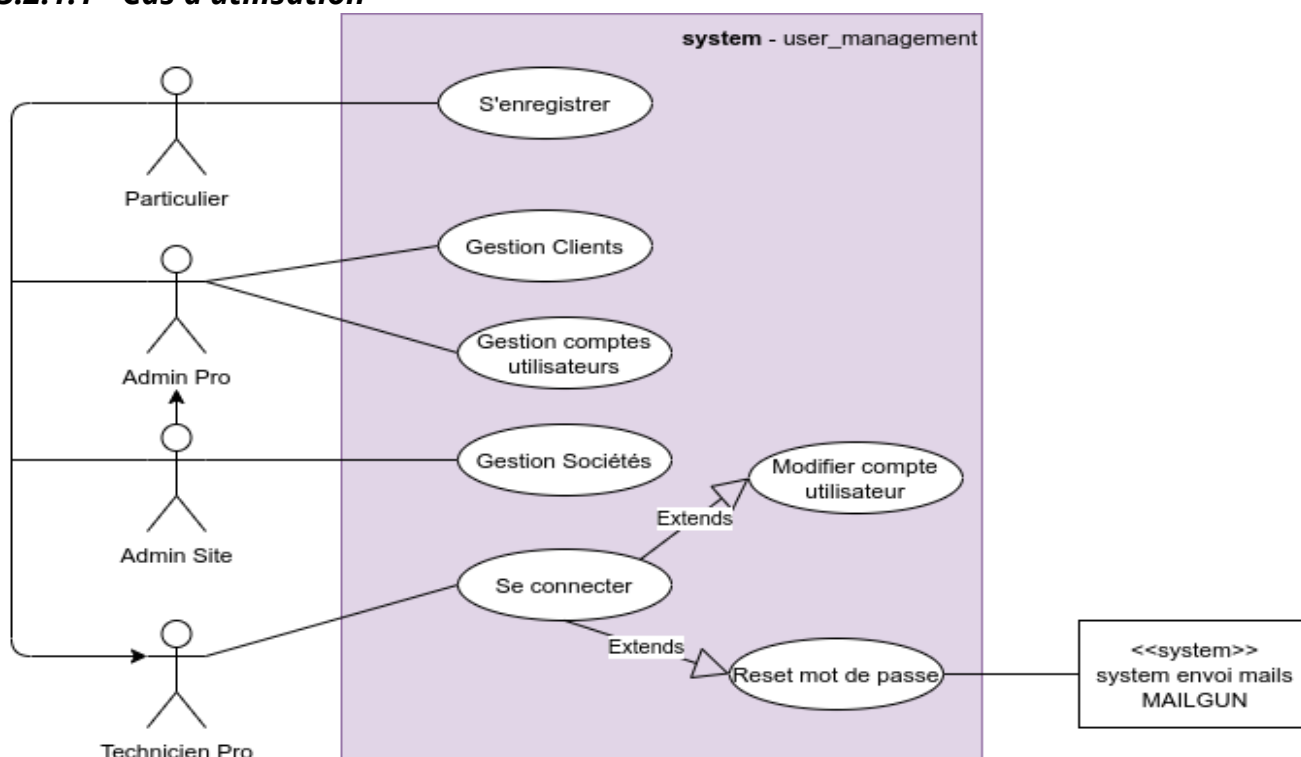
## 5.2 - Les cas d'utilisation

Les cas d'utilisation permettent pour chaque package de répondre à la question QUI devra faire QUOI pour s'acquitter de sa tâche. À noter qu'à aucun moment, n'est abordé la connexion, car le système d'authentification ne possédera aucune complexité. Nous partirons donc à chaque fois du prédicat suivant : l'utilisateur est connecté.

Les descriptions textuelles des cas d'utilisation reprennent les schémas des cas d'utilisation en décrivant la chronologie des actions qui devront être réalisées par les acteurs et par le système. Cela s'approche d'un scénario et permet aussi de poser des questions et de clarifier certains points d'ergonomie, d'échecs d'actions...

### 5.2.1 - Package "user\_management"

#### 5.2.1.1 - Cas d'utilisation



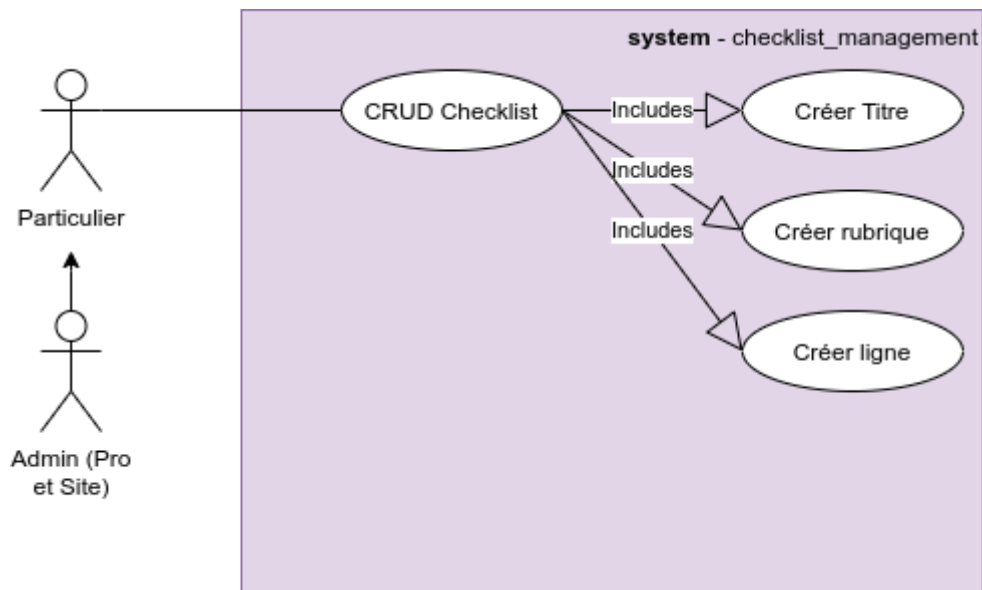
Nous voyons donc que :

- L'admin Site peut faire tout ce que peuvent faire les autres acteurs.
- Tous les acteurs peuvent se connecter et réinitialiser leur mot de passe.
- Le fait de s'enregistrer sur le site donne automatiquement le niveau « particulier ».
- Seul l'acteur « Admin Pro » peut créer des « clients » pour le compte de sa société.

La logique de chacune de ces cas d'utilisation étant somme toute basique, il n'a pas été fait de description textuelle.

## 5.2.2 - Package "Administration des check-lists"

### 5.2.2.1 - Cas d'utilisation



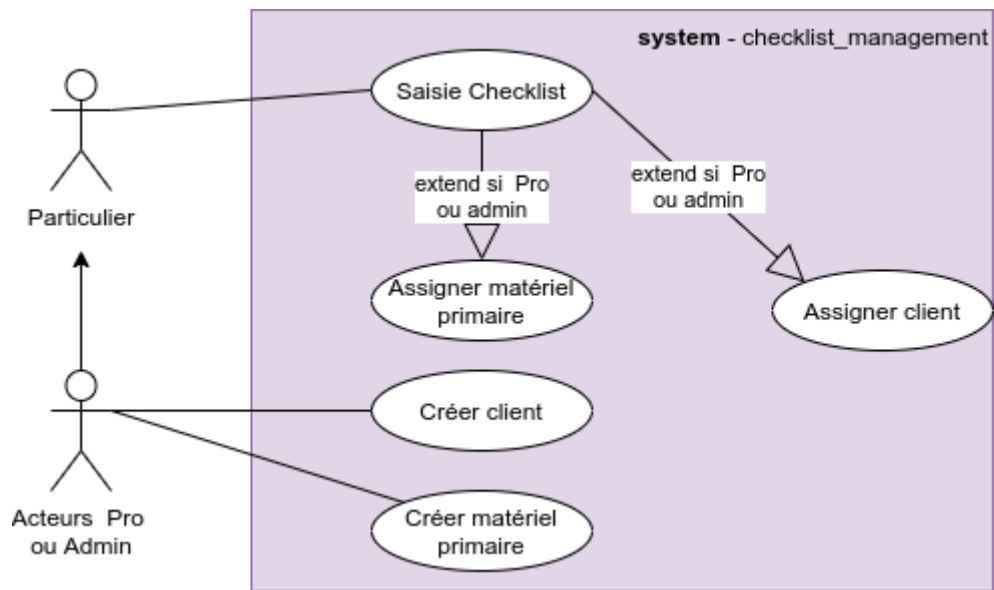
Ici CRUD signifie (Create-Read-Update-Delete – Créer-Lire-Modifier-Supprimer).

À noter que aucune check-list ne sera supprimée, car il faut pouvoir y revenir ultérieurement, il s'agit d'un flag qui met la check-list à « disable ». Le même principe est utilisé (pour les mêmes raisons) pour la modification. L'original est conservé mais mis à « disable » et une nouvelle check-list ayant le même nom est créée (elles sont différenciées par leur id et la date de création).

	PACKAGE	Création check-list
	N° CAS D'UTILISATION	1
	NOM	Création (Cas d'utilisation « Administration des check-lists »)
IDENTIFICATION	ACTEURS	Admin Pro et Site – Particulier
	DESCRIPTION	L'acteur crée une check-list
	AUTEUR	Jean-Martial Le Magorou
	DATE	28/09/2020
	PRÉCONDITION	L'utilisateur est connecté au système
	DÉMARRAGE	L'acteur souhaite créer une nouvelle check-list
DESCRIPTION	SCENARIO NOMINAL	1. – <b>Le système</b> fait appel au cas d'utilisation interne « Créer check-list » 2. – <b>Le système</b> fait appel au cas d'utilisation interne « Créer titre » 3. – <b>le système</b> fait appel au cas d'utilisation interne « Créer rubrique » 4 – <b>Le système</b> fait appel au cas d'utilisation interne « Créer ligne » 5 – <b>Le système</b> fait appel au cas d'utilisation interne « Saisie paramètres check-list » 6 – <b>Le système</b> fait appel au cas d'utilisation interne « Sauvegarde Check-List » puis sort
	SCENARIO(S) ALTERNATIF(S)	5a. – Tant que saisie non terminée retourne en 3 ou en 4
	SCENARIO(S) D'EXCEPTION	6a. – Échec enregistrement. Le système affiche un message d'erreur et sort
FIN et POST-CONDITIONS	SCENARIO NOMINAL	– Fin du processus de commande
	SCENARIO ALTERNATIF	
	SCENARIO D'EXCEPTION	– Fin du processus de commande
COMPLÉMENTS	ERGONOMIE	– Le système doit revenir à la liste des check-lists (mode création) à la fin triée par date inverse (dernière saisie en tête de liste).
	PROBLÈMES NON RÉSOLUS	
	PERFORMANCES ATTENDUES	

### 5.2.3 - Package "Saisie des check-lists"

#### 5.2.3.1 - Cas d'utilisation



	PACKAGE	Saisie check-list
	N° CAS D'UTILISATION	2
	NOM	Saisie (Cas d'utilisation « Saisie des check-lists »)
IDENTIFICATION	ACTEURS	Tech Pro et Particulier
	DESCRIPTION	L'acteur remplit une check-list (Saisie)
	AUTEUR	Jean-Martial Le Magorou
	DATE	29/09/2020
	PRÉCONDITION	L'utilisateur est connecté au système
	DÉMARRAGE	L'acteur souhaite remplir une check-list (Saisie)
DESCRIPTION	SCENARIO NOMINAL	<p>1. – <b>Le système</b> fait appel au cas d'utilisation interne « Liste des check-lists.</p> <p>2 – L'utilisateur sélectionne la check-list à remplir</p> <p>3. – <b>Le système</b> fait appel au cas d'utilisation interne « Saisie check-list »</p> <p>4 – L'utilisateur sélectionne un client</p> <p>5 – L'utilisateur remplit la check-list.</p> <p>6 – L'utilisateur valide la check-list</p> <p>7 – <b>Le système</b> fait appel au cas d'utilisation interne « Enregistrement check-list saisie »</p> <p>8 – <b>Le système</b> fait appel au cas d'utilisation interne « Envoi check-list mail »</p> <p>9 – <b>Le système</b> fait appel au cas d'utilisation interne « Liste des check-lists.</p>
	SCENARIO(S) ALTERNATIF(S)	<p>4a – L'utilisateur est un particulier → pas de client. Passage en 5</p> <p>4b – Le client est un nouveau client. <b>Le système</b> fait appel au cas d'utilisation interne « création client » et passe en 5</p> <p>6a. – L'utilisateur met la check-list en attente pour terminer la saisie ultérieurement → statut en cours</p> <p>8a – Pas d'envoi de mail si statut « en cours »</p>
	SCENARIO(S) D'EXCEPTION	<p>7a. – Échec enregistrement. Le système affiche un message d'erreur et sort.</p> <p>8a – Échec envoi de mail. Le système affiche un message et continue.</p>
FIN et POST-CONDITIONS	SCENARIO NOMINAL	– Fin de la saisie.
	SCENARIO ALTERNATIF	
	SCENARIO D'EXCEPTION	– Fin de la saisie
	ERGONOMIE	– Le système doit revenir à la liste des check-lists (mode saisie) à la fin triée par date inverse (dernière saisie en tête de liste).
	PROBLÈMES NON	

ÉMENTS	RÉSOLUS	
	PERFORMANCES ATTENDUES	

## 5.3 - Les règles de gestion générales

Nous allons séparer les règles de gestion en 3 grandes parties : Les cas généraux, le Particulier et le Professionnel. La partie « Professionnel » se découpant elle-même en 2 sous-parties (Administrateur et Techniciens).

### 5.3.1 - Cas généraux

- Une check-list saisie et validé ne peut plus être modifiée.
- Un client ne peut pas être supprimé.

### 5.3.2 - Le Particulier

- Créé lui-même son compte (S'enregistrer sur home page).
- Créé/modifie/supprime/saisie/visualise ses propres check-list.

### 5.3.3 - Le Professionnel

- Notion de société. Le professionnel est rattaché à une société. Tout ce qu'il fait est au nom de la société.
- Notion de client. Le professionnel peut saisir des check-lists pour le compte de.
- Notion de matériel primaire et secondaire. Il est possible de saisir une check-list pour un matériel attaché à un autre matériel (Un brûleur dans une chaudière, un groupe froid sur un camion...).

#### 5.3.3.1 - L'administrateur

- Compte et société créés par l'administrateur Site.
- Créé les comptes des techniciens.
- Créé/modifie/supprime les check-lists.
- Créé/modifie les clients.
- Saisie les check-lists.

#### 5.3.3.2 - Le technicien

- Saisie les check-lists.
- Créé/modifie les clients.

## 6 - GLOSSAIRE

<b>UML</b>	Langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système.
<b>Package UML</b>	Sert à grouper des éléments en un ensemble cohérent
<b>Cas d'utilisation</b>	Définit une manière d'utiliser un système qui a une valeur ou une utilité pour les acteurs impliqués