



République Tunisienne Ministère  
de l'Enseignement Supérieur  
et de la Recherche Scientifique

---

## PROJET DE FIN D'ÉTUDES

Presenté en vue de l'obtention de la  
License nationale en Informatique de gestion

---

# Modélisation et Réalisation d'une application web pour la gestion de la collecte des déchets

---

Organisation d'accueil :



Encadré par :

**Encadrant universitaire**  
Ayachi Raouia

**Encadrants professionnel**  
Mhiri Walid  
Louati Rami

Élaboré par :  
CHERNI Rihab  
BEN CHaabene Feryel

Année Universitaire 2021/2022

---

## DÉDICACES

Je voulais commencer par remercier mes parents **Lassaad** et **Hanen** pour leur foi en ma réussite et les sacrifices consentis pour mon avenir. Vous avez été toujours à mes côtés m'apportant tas de soutien et d'amour.

Que dieu vous procure une bonne santé et une longue vie.

À mes chères soeurs **Fatma** et **Yosra** d'avoir cru en moi et de m'avoir poussé à mieux faire tout au long de la réalisation de ce travail.

À ma grand-mère **Kalthoum** pour l'immense support qu'elle m'a présenté.

À ma tante **Nadia**, de ne jamais avoir cessé de me soutenir et de m'encourager.

À tous mes chères amis de m'avoir supporté et contribué à mon développement personnel.

À mon binôme **Rihab**, pour le support et la patience que tu m'as accordée durant cette période, merci infiniment. Dernièrement, je tiens à exprimer mes plus sincères gratitude pour toute personne qui m'a aidé même avec un mot, un conseil à la réalisation de ce projet.

Ben Chaabene Feryel

---

## DÉDICACES

Je tiens à adresser mes sincères remerciements à ma chére mère **Rebah** pour le courage et la patience illimitée qu'elle m'a accordé tout au long de ma vie surtout dans les moments difficiles.

Mon père **Salah** dont les sacrifices m'ont permis d'accomplir une éducation digne.

Mes soeurs **Arij** et **Rania** pour l'encouragement continue de leur part, elles ont su me soutenir en amont de ce stage.

À tous mes amis pour leur pleine disponibilité et l'immense aide et soutien qu'ils m'ont fourni avec.

Je dédie ce travail en espérant avoir répondu à les souhaits de mes proches de me voir réussir en exprimant plus sincères gratitude et respects.

Cherni Rihab

---

# Remerciements

Au terme de notre stage de fin d'études, c'est avec un grand plaisir que nous dédions ces lignes pour remercier toute personne ayant contribué, directement et indirectement, à l'aboutissement de ce modeste travail.

Nous tenons à remercier particulièrement et à témoigner toutes nos reconnaissances à *Mme Raouia Ayachi*, notre encadrante universitaire, d'avoir toujours cru en nous . Nous sommes également reconnaissants pour votre patience, votre disponibilité et les conseils accordés afin d'accomplir ce projet avec perfection.

Nous exprimons également nos profonds remerciements, avec toute gratitude, à toute l'équipe de la société **RE:SCHOOL EDUCATION** pour nous avoir accueilli si chaleureusement et de toute aide qu'ils nous ont pu prodiguer au cours de cette expérience professionnelle. Mr **Walid Mhiri** qui nous a accordé l'opportunité de faire partie de son honorable équipe et du soutien accordé de sa part et *Mr Rami Louati*, de nous avoir encouragé tout au long du stage et surtout de nous avoir guider dans la bonne direction avec l'aide, les renseignements et conseils fourni .

Nous tenons à remercier les membres du jury pour le grand honneur qu'ils nous rendent de pouvoir juger notre travail.

Nous voudrions également remercier tous les enseignants de l'**Ecole Supérieure des Sciences Économiques et Commerciales en Tunis** qui ont contribué à notre formation et nous ont bénéficié par leurs connaissances et expériences . Nos derniers remerciements, nous tenons à les exprimer à tous nos proches qui nous ont permis de réaliser ce travail.

# Table des matières

<b>Introduction générale</b>	<b>12</b>
<b>1 Étude préalable du projet</b>	<b>14</b>
<b>Introduction</b>	<b>14</b>
1.1 Contexte général . . . . .	14
1.2 Présentation de la société . . . . .	14
1.2.1 Fiche de présentation . . . . .	15
1.2.2 Organigramme de RE:SCHOOL . . . . .	15
1.3 Étude de l'existant et les solutions proposées . . . . .	15
1.3.1 Description et critique de l'existant . . . . .	16
1.3.2 Solution proposée . . . . .	16
1.4 Méthodologie adoptée . . . . .	17
1.4.1 Méthodologie Agile . . . . .	17
1.4.2 Langages de modélisation . . . . .	18
Conclusion . . . . .	18
<b>2 Spécification de besoins</b>	<b>19</b>
Introduction . . . . .	19
2.1 Capture des besoins . . . . .	19
2.1.1 Identification des acteurs . . . . .	19
2.1.2 Besoins fonctionnels . . . . .	20
2.1.3 Besoins non fonctionnels . . . . .	20
2.2 Diagramme de cas d'utilisation global . . . . .	21
2.3 Le Backlog du produit . . . . .	21
2.4 Planification des sprints . . . . .	22
Conclusion . . . . .	22
<b>3 Sprint 1 : Gestion de poubelles et camions et Visualisation de carte géographique</b>	<b>23</b>
Introduction . . . . .	23
3.1 Analyse . . . . .	24
3.1.1 Raffinement du cas d'utilisation Gestion des poubelles . . . . .	24
3.1.2 Raffinement du cas d'utilisation Gestion des camions . . . . .	26
3.1.3 Raffinement du cas d'utilisation Visualisation de la carte géographique .	29
3.1.4 Diagrammes de séquence système de la gestion des poubelles . . . . .	31
3.1.5 Diagrammes de séquence système de la visualisation de la carte géographique . . . . .	36
3.2 Conception . . . . .	38

3.2.1	Diagramme de séquence de conception . . . . .	38
3.2.2	Diagramme de classe global du premier sprint . . . . .	42
3.3	Codage . . . . .	43
3.3.1	Schéma de la base de données . . . . .	43
3.3.2	Extrait du code . . . . .	45
3.4	Test . . . . .	47
3.5	Sprint Review . . . . .	50
3.5.1	Livraison du sprint 1 . . . . .	50
3.5.2	Obstacles rencontrés et solutions proposées . . . . .	50
	Conclusion . . . . .	51
<b>4</b>	<b>Sprint 2 : Gestion de Ressources Humaines et Génération de statistiques</b>	<b>52</b>
	Introduction . . . . .	52
4.1	Analyse . . . . .	53
4.1.1	Diagrammes de séquence système du deuxième sprint . . . . .	58
4.2	Conception . . . . .	58
4.2.1	Diagramme de séquence de conception . . . . .	58
4.2.2	Diagramme de classe global du deuxième sprint . . . . .	64
4.3	Codage . . . . .	64
4.3.1	Schéma de la base de données . . . . .	64
4.3.2	Extraits du code . . . . .	66
4.4	Test . . . . .	67
4.5	Sprint Review . . . . .	71
4.5.1	Livraison du sprint 2 . . . . .	71
4.5.2	Obstacles rencontrés et solutions proposées . . . . .	71
	Conclusion . . . . .	71
<b>5</b>	<b>Sprint 3 : Authentification et Gestion de comptes</b>	<b>72</b>
	Introduction . . . . .	72
5.1	Analyse . . . . .	72
5.2	Conception . . . . .	73
5.2.1	Diagramme de séquence de conception . . . . .	73
5.2.2	Diagramme de classe global du troisième sprint . . . . .	76
5.3	Codage . . . . .	76
5.3.1	Schéma de la base de données . . . . .	76
5.3.2	Extraits du code . . . . .	77
5.4	Tests . . . . .	79
5.5	Sprint Review . . . . .	84
5.5.1	Livraison du sprint 3 . . . . .	84
5.5.2	Obstacles rencontrés et solutions proposées . . . . .	84
	Conclusion . . . . .	84
<b>6</b>	<b>Clôture du projet</b>	<b>85</b>
	Introduction . . . . .	85
6.1	Environnement de développement . . . . .	85
6.1.1	Environnement matériel . . . . .	85
6.1.2	Environnement logiciel . . . . .	86
6.1.3	Technologies et langages utilisées . . . . .	87
6.2	Architecture d'implémentation . . . . .	88
6.2.1	Architecture logicielle . . . . .	88
6.2.2	Gestion de projet . . . . .	89

Conclusion . . . . .	90
<b>Conclusion générale et Perspectives</b>	<b>91</b>
<b>Bibliographie</b>	<b>92</b>

# Liste des tableaux

1.1	Fiche de présentation . . . . .	15
2.1	Le Backlog du produit . . . . .	22
2.2	Planification des Sprints . . . . .	22
3.1	Backlog du sprint 1 . . . . .	23
3.2	Description textuelle du sous cas d'utilisation «Gérer commandes poubelles» . .	25
3.3	Description textuelle du sous cas d'utilisation «Consulter historique commandes»	25
3.4	Description textuelle du sous cas d'utilisation «Commander poubelles» . . . . .	25
3.5	Description textuelle du sous cas d'utilisation «Annuler commande poubelles» .	26
3.6	Description textuelle du sous cas d'utilisation «Modifier commande poubelles» .	26
3.7	Description textuelle du sous cas d'utilisation «Ajouter camion» . . . . .	27
3.8	Description textuelle du sous cas d'utilisation «Modifier camion» . . . . .	27
3.9	Description textuelle du sous cas d'utilisation «Supprimer camion» . . . . .	28
3.10	Description textuelle du sous cas d'utilisation «Rechercher camion» . . . . .	28
3.11	Description textuelle du sous cas d'utilisation «Afficher détails camion» . . . . .	28
3.12	Description textuelle du sous cas d'utilisation «Visualiser la carte géographique»	29
3.13	Description textuelle du sous cas d'utilisation «Consulter les détails des poubelles par établissements» . . . . .	30
3.14	Description textuelle du sous cas d'utilisation «Consulter les détails des camions»	30
3.15	Description textuelle du sous cas d'utilisation «Visualiser la carte géographique par établissement» . . . . .	31
3.16	table poubelles . . . . .	43
3.17	table bloc_poubelles . . . . .	44
3.18	table etage_etablissement . . . . .	44
3.19	table camions . . . . .	44
3.20	table commande_poubelles . . . . .	44
3.21	table detail_commande_poubelles . . . . .	45
4.1	Backlog du sprint 2 . . . . .	52
4.2	Description textuelle du sous cas d'utilisation «Ajouter fournisseur» . . . . .	54
4.3	Description textuelle du sous cas d'utilisation «Modifier Responsable Établissement» . . . . .	54
4.4	Description textuelle du sous cas d'utilisation «Supprimer Mécanicien» . . . . .	55
4.5	Description textuelle du sous cas d'utilisation «Rechercher Ouvrier» . . . . .	55
4.6	Description textuelle du sous cas d'utilisation «Afficher détails Réparateur poubelles» . . . . .	56
4.7	Description textuelle du sous cas d'utilisation «Générer les statistiques» . . . . .	56

4.8	Description textuelle du sous cas d'utilisation «Générer les statistiques par établissement» . . . . .	57
4.9	Description textuelle du sous cas d'utilisation «Visualiser les statistiques par zones» . . . . .	57
4.10	table fournisseurs . . . . .	64
4.11	table ouvriers . . . . .	65
4.12	table stock_poubelles . . . . .	65
4.13	table materiaux_primaire . . . . .	65
4.14	table reparateur_poubelles . . . . .	65
4.15	table reparation_poubelles . . . . .	66
5.1	Backlog du Sprint 3 . . . . .	72
5.2	Description textuelle du sous cas d'utilisation «S'authentifier» . . . . .	73
5.3	Description textuelle du sous cas d'utilisation «Gérer son compte» . . . . .	73
5.4	table gestionnaires . . . . .	76
5.5	table responsables_etablissements . . . . .	77
6.1	Description des machines de développement . . . . .	85

# Table des figures

1.1	Organigramme de la société . . . . .	15
1.2	La méthode Scrum <sup>4</sup> . . . . .	17
2.1	Diagramme de cas d'utilisation global . . . . .	21
3.1	Raffinement du cas d'utilisation "Gestion des poubelles" . . . . .	24
3.2	Raffinement du cas d'utilisation "Gérer les camions" . . . . .	26
3.3	Raffinement du cas d'utilisation "Visualiser carte géographique" . . . . .	29
3.4	Diagramme de séquence système du sous cas d'utilisation "Gérer commandes poubelles" pour le gestionnaire . . . . .	31
3.5	Diagramme de séquence système du sous cas d'utilisation "Gérer commandes poubelles" pour le responsable d'établissement . . . . .	32
3.6	Diagramme de séquence système du sous cas d'utilisation "Consulter historique commandes" pour le gestionnaire . . . . .	32
3.7	Diagramme de séquence système du sous cas d'utilisation "Consulter historique commandes" pour le responsable de l'établissement . . . . .	33
3.8	Diagramme de séquence système du sous cas d'utilisation "Commander poubelles" . . . . .	33
3.9	Diagramme de séquence système du sous cas d'utilisation "Annuler Commande poubelles" . . . . .	34
3.10	Diagramme de séquence système du sous cas d'utilisation "Modifier Commande poubelles" . . . . .	35
3.11	Diagramme de séquence système du sous cas d'utilisation «Visualiser la carte géographique» . . . . .	36
3.12	Diagramme de séquence système du sous cas d'utilisation «Consulter les détails des poubelles par établissements» . . . . .	36
3.13	Diagramme de séquence système du sous cas d'utilisation «Consulter la localisation des camions» . . . . .	37
3.14	Diagramme de séquence système du sous cas d'utilisation «Visualiser la carte géographique par établissement» . . . . .	37
3.15	Diagramme de séquence de conception du cas d'utilisation "Commander poubelles" . . . . .	38
3.16	Diagramme de séquence de conception du cas d'utilisation "Ajouter camion" . . . . .	39
3.17	Diagramme de séquence de conception du cas d'utilisation "Modifier camion" . . . . .	40
3.18	Diagramme de séquence de conception du cas d'utilisation "Supprimer camion" . . . . .	41
3.19	Diagramme de séquence de conception du cas d'utilisation "Rechercher camion" . . . . .	41
3.20	Diagramme de séquence de conception du cas d'utilisation "Afficher détails camion" . . . . .	42
3.21	Diagramme de classe global du sprint 1 . . . . .	43
3.22	Extrait du code Table.js . . . . .	45
3.23	Extrait du code MapGestionnaire.js . . . . .	46

3.24	Extrait du code MapGestionnaire.js . . . . .	46
3.25	Interface de la table Poubelles groupées par zones de travail . . . . .	47
3.26	Interface de la table Poubelles groupées par établissements . . . . .	47
3.27	Interface de la table des poubelles . . . . .	48
3.28	Test de suppression d'un camion . . . . .	48
3.29	Test de suppression d'un camion . . . . .	49
3.30	Interface de la carte géographique . . . . .	49
3.31	Interface détails d'un camion . . . . .	50
4.1	Raffinement du diagramme de cas d'utilisation du Sprint 2 . . . . .	53
4.2	Diagramme de séquence système du sous cas d'utilisation "Visualiser les statistiques par zones" . . . . .	58
4.3	Diagramme de séquence système de conception du sous cas d'utilisation "Ajouter Fournisseur" . . . . .	59
4.4	Diagramme de séquence système de conception du sous cas d'utilisation "Modifier Responsable Etablissement" . . . . .	60
4.5	Diagramme de séquence système de conception du sous cas d'utilisation "Supprimer Mécanicien" . . . . .	61
4.6	Diagramme de séquence système de conception du sous cas d'utilisation "Rechercher Ouvrier" . . . . .	62
4.7	Diagramme de séquence système de conception du sous cas d'utilisation "Afficher détails Réparateur Poubelle" . . . . .	62
4.8	Diagramme de séquence système de conception du sous cas d'utilisation "Générer les statistiques" . . . . .	63
4.9	Diagramme de séquence système de conception du sous cas d'utilisation "Générer les statistiques par établissement" . . . . .	63
4.10	Diagramme de classe du Sprint 2 . . . . .	64
4.11	Extrait du code DialogShow.js . . . . .	66
4.12	Extrait du code DialogAddUpdate.js . . . . .	67
4.13	Interface du Dashboard du gestionnaire . . . . .	67
4.14	Interface de la partie gestion de pannes poubelles . . . . .	68
4.15	Interface de la partie gestion de pannes camions . . . . .	68
4.16	Test de graphe à barres . . . . .	69
4.17	Test sur la carte de la Tunisie . . . . .	69
4.18	Interface de la table Ouvriers . . . . .	70
4.19	Test . . . . .	70
4.20	Test . . . . .	71
5.1	Raffinement du diagramme de cas d'utilisation du Sprint 3 . . . . .	72
5.2	Diagramme de séquence système de conception du cas d'utilisation «S'authentifier» . . . . .	74
5.3	Diagramme de séquence système de conception du cas d'utilisation «Gérer son compte» pour le gestionnaire . . . . .	75
5.4	Diagramme de classe du Sprint 3 . . . . .	76
5.5	Extrait du code de Login.js . . . . .	77
5.6	Extrait du code de ProfileGestionnaire.js . . . . .	78
5.7	Extrait du code de inputUpdate.js . . . . .	78
5.8	Interface de l'internaute . . . . .	79
5.9	Interface de l'authentification . . . . .	79
5.10	Test de l'authentification (champs obligatoires) . . . . .	80
5.11	Test d'entrée de données incorrects . . . . .	80

5.12	Affichage de message d'échec de connexion . . . . .	81
5.13	Affichage de message de succès de connexion . . . . .	81
5.14	Test de la 1ère méthode de gestion de compte . . . . .	82
5.15	Test de la 2ème méthode de gestion de compte -partie 1 . . . . .	82
5.16	Test de la 2ème méthode de gestion de compte -partie 2 . . . . .	82
5.17	Interface du Profil . . . . .	83
5.18	Interface de modification des données du compte . . . . .	83
5.19	Test de modification des données incorrectes du compte . . . . .	84
6.1	Architecture MVC . . . . .	89
6.2	Scrum Blocking Board du deuxième Sprint . . . . .	90
6.3	Burndown Chart . . . . .	90

# Introduction générale

Dans les pays en voie de développement, tel que notre pays la Tunisie, les outils des nouvelles technologies et les systèmes d'information ne sont pas assez exploité ou encore pire développer en interne soit à partir de 0, soit à partir de solution open source.

Dans le domaine de l'environnement et plus précisément dans le domaine de la gestion des déchets et du recyclage, il est sous exploité.

Nous pensons au contraire que ces outils peuvent servir non seulement au niveau professionnel mais aussi au niveau éducatif et de changer les mentalités, les habitudes des futurs générations par l'utilisation des nouvelles technologies et de rendre notre pays plus vert, plus propre et précurseur dans le domaine de l'ECO-TECH en Afrique et pourquoi pas dans le monde.

RE:SCHOOL Education a pour but d'innover en proposant sa solution technologique adaptée au besoin du marché local tunisien afin d'effectuer le tri, le recyclage des déchets d'une manière structurelle, professionnelle, dynamique et technologique afin de travailler sur une dimension macro-économique et d'optimiser le temps d'exploitation et de gestion des déchets poubelles.

Nous serons responsables de la construction d'une application web qui accorde aux différents utilisateurs la gestion de ce processus et des ressources employés à cette dernière et une deuxième équipe sera responsable de créer une application mobile pour faciliter la collecte des déchets par les ouvriers (agents de collecte) et une autre pour les clients voulant acheter des déchets recyclables.

C'est dans ce cadre que s'inscrit notre projet ambitieux de fin d'études intitulé "Modélisation et Réalisation d'une application web pour la gestion de la collecte des déchets" à l'Ecole Supérieure des Sciences Économiques et Commerciales en Tunis au sein de la société RE:SCHOOL Education.

Notre rapport va comporter 6 chapitres :

Le premier chapitre est intitulé "Étude préalable du projet" qui va mettre le projet dans son contexte général tout en présentant la société d'accueil. Le deuxième chapitre "Spécification des besoins" sera la première partie dans l'application du cadre méthodologique Scrum, il portera sur l'étude des besoins fonctionnels et non fonctionnels du client, la spécification des acteurs de la plate-forme et la composition de chaque Sprint. Puis, on procédera à présenter les Sprints du projet qui se composeront de quatre phases : Analyse, Conception, Codage et Tests.

Le troisième chapitre "Sprint 1 : Gestion de poubelles et camions et visualisation de carte géographique" est notre premier Sprint ensuite on trouvera le deuxième Sprint "Sprint 2 : Gestion des Ressources Humaines et Génération des statistiques" comme le quatrième chapitre.

Le cinquième chapitre "Sprint 3 : Authentification et Gestion de comptes" sera notre dernier Sprint dans ce projet.

Finalement, le sixième et dernier chapitre sera intitulé "Clôture du projet" qui listera les différents technologies et langages utilisés en plus de l'architecture employée tout au long de la réalisation de cette application.

Pour finir, on va clôturer ce rapport par une conclusion générale résumant le travail réalisé pour livrer ce projet et listant tout ce qu'on avait appris le long de ces trois mois au sein de la société RE:SCHOOL Education.

# Étude préalable du projet

## Introduction

Dans ce premier chapitre, nous allons commencer par mettre le projet dans son contexte général. Premièrement, nous commencerons par présenter la société qui développe le projet et son organigramme. Ensuite, nous allons procéder à l'étude du projet commercial, ses problématiques et les différentes solutions proposées afin de résoudre toutes les problématiques dû au projet.

Finalement, nous terminerons par la méthodologie utilisée tout au long du projet.

### 1.1 Contexte général

Dans le cadre de la formation de Licence Nationale Business Computing à l'Ecole Supérieure des Sciences Économiques et Commerciales en Tunis , on a eu l'opportunité de réaliser notre projet de fin d'études au sein de la société RE:SCHOOL EDUCATION TUNISIE pour une durée de trois mois du 02/02/2022 au 02/05/2022.

### 1.2 Présentation de la société

RE:SCHOOL EDUCATION TUNISIE est un projet dédié au soutien de l'éducation sous différentes formes aussi bien pour les écoles que les parents et surtout les enfants, en mettant à disposition soit à petit prix soit gratuitement du soutien scolaire soit sous forme digitale soit présentiel et une approche unique de développement d'un marché économique vertueux, le projet va pouvoir aider financièrement dans le soutien scolaire pour les enfants défavorisés en plus des autres enfants. RE:SCHOOL EDUCATION TUNISIE comporte plusieurs départements dont le département RE:SCHOOL EDUGAME, COFFEE BOOK, ATELIERS DE LANGUES et RE:SCHOOL ECOLOGY.

Tous les départements RE:SCHOOL EDUCATION TUNISIE ont pour objet d'offrir des activités à caractère éducatif et ludique dans le domaine de l'éducation soit à petit prix voir même gratuit et de pouvoir redistribuer jusqu'à 70% de ses revenus publicitaires pour soutenir des projets éducatifs. L'entreprise compte développer ses départements avec des sociétés partenaires qui désirent s'investir dans le *RSE* et consolider leurs noms et actions dans le domaine de l'éducation.

### 1.2.1 Fiche de présentation

<b>Nom de l'entreprise</b>	RE:SCHOOL EDUCATION
<b>Adresse</b>	40 Rue des Minéraux - 2035 Charguia 1
<b>Téléphone</b>	58080111
<b>Email entreprise</b>	info@reschool.education
<b>Site web</b>	reschoolwethink.education

TABLE 1.1 – Fiche de présentation

### 1.2.2 Organigramme de RE:SCHOOL

Ci-dessous est l'organigramme de la société RE:SCHOOL :

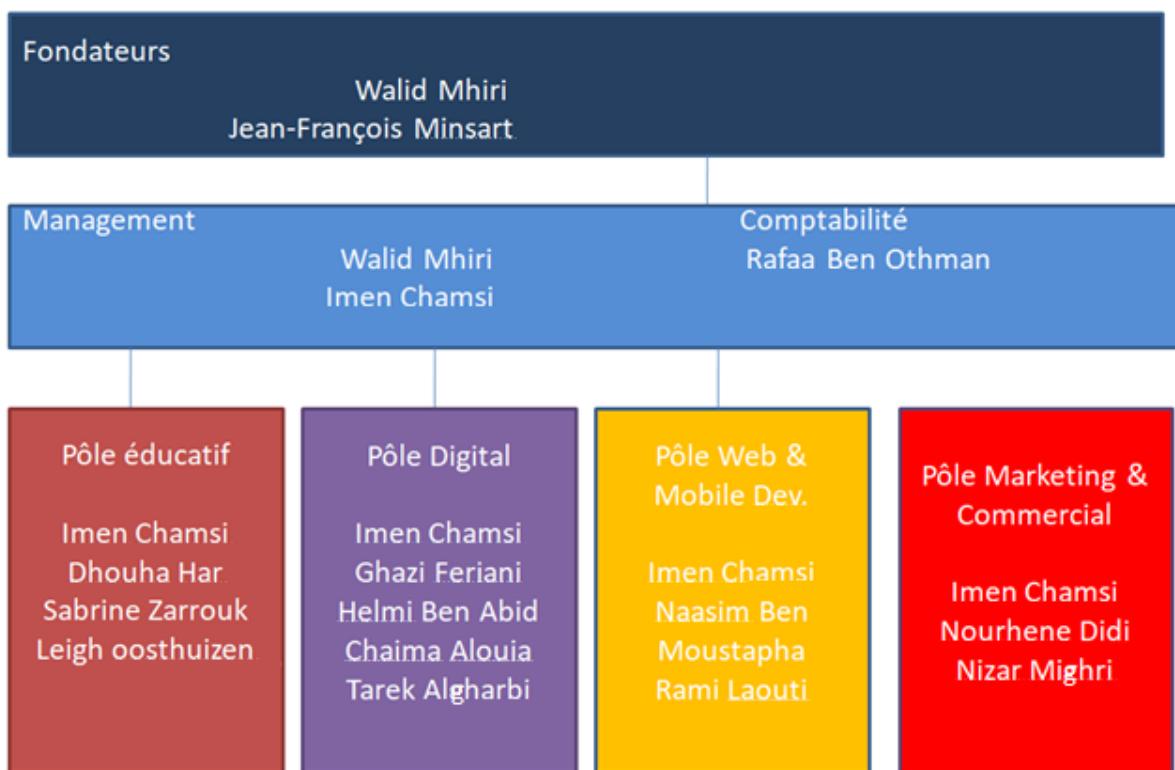


FIGURE 1.1 – Organigramme de la société

## 1.3 Étude de l'existant et les solutions proposées

L'analyse de la situation existante est une étape essentielle du projet. Cette étape permet une analyse de l'environnement, pour éviter les répétitions et avoir un produit/service unique.

### 1.3.1 Description et critique de l'existant

La société part du constat que dans le domaine de l'écologie, le secteur ne permet pas de remédier à la problématique de la gestion des déchets qui n'arrêtent pas de s'accumuler dans notre pays et qui devient un problème écologique, sanitaire et aussi économique. La société a aussi constaté que malgré les outils technologiques disponibles dans le pays, tel que l'internet, les smartphones, les développeurs, le cloud,..., sont nullement utilisés dans le secteur de la gestion des déchets.

Pour offrir des solutions structurelles, technologiques et innovantes dans le secteur, la société va commencer par développer son offre dans le domaine de l'éducation et va viser tous les établissements scolaires de la maternelle à l'université pour implanter une offre de gestion et recyclage des déchets par rapport à une communauté qui est malgré elle, un acteur polluant.

Pour y arriver, RE:SCHOOL développe un projet unique en son genre qui va allier les nouvelles technologies au domaine classique des poubelles et rendre ces dernières super intelligentes et connectées.

La solution permettra d'optimiser l'opération de ramassage et de recyclage des déchets, ce qui rendra notre environnement plus propre. Cette solution a pour but de sensibiliser les enfants et les jeunes gens à devenir plus vert.

En effet, cette idée n'a jamais été faite en Tunisie, et selon la RE:SCHOOL ni ailleurs en Afrique, dans le domaine de l'éducation, ce qui la rend non seulement pertinente et nécessaire à établir dans notre pays, mais surtout précurseur!! .

Pour cela, on va créer une application spécifique afin de réaliser la gestion de cette solution.

Après avoir constaté la situation de collecte et de recyclage des déchets en Tunisie et l'absence de solution technologique, nous avons relevé énormément d'insuffisance dont :

- Absence de système de gestion pour le ramassage et le tri des déchets.
- Absence d'un support automatisé pour la gestion des équipes de terrain et des ressources technologiques utilisées.
- Absence d'un support automatisé pour le traitement de la gestion, de l'organisation, de la vente, des réclamations et des pannes.
- Chaque jour, les travailleurs doivent faire plusieurs tournées de ramassage des ordures ce qui n'est pas facile à réaliser surtout avec les conditions de travail très difficiles telles que le matériel usagé, le manque de protection, la pollution, la circulation, les accidents et les heures de travail très long... Les ouvriers vont perdre beaucoup de temps à cause de la collecte des ordures dans les poubelles presque vides dans des dizaines d'établissements.
- Perte d'argent causée par le montant dépensé pour l'achat de carburant.
- Pollution de l'environnement causé par la consommation massive du carburant et la pollution provoquée par les déchets.
- Manque d'exploitation et perte potentielle de déchets : la quantité de déchets est énorme qui peut être recyclée et réutilisée.

### 1.3.2 Solution proposée

Après avoir discuté avec la société et avoir entendu leur attentes pour le projet, nous avons proposé la création d'une application web appelé "RE:SCHOOL ECOLOGY" qui permettra une gestion complète de A à Z de l'ensemble du processus qui va de la gestion du parc poubelle,

des camions, équipe de terrain et au bureau ainsi que la relation avec les différents intervenants (établissements scolaires, ministères, institutions) et au final les clients acheteurs.

## 1.4 Méthodologie adoptée

### 1.4.1 Méthodologie Agile

La méthode de gestion de projet agile est née suite à l'évolution des technologies récurrentes. Cette méthodologie est très flexible en comparaison aux méthodologies traditionnelles, elle permet le découpage en tâches, la diminution des risques et elle priorise la satisfaction complète du client.

#### Processus Scrum

”Un cadre de travail dans duquel les acteurs peuvent entamer des problèmes compliqués, en livrant des produits d'une manière efficace et créative.”<sup>2</sup>

Scrum est considéré la méthode la plus utilisée entre les méthodes Agile existantes, elle porte aide à la conduction des projets de façon répété en procédant à des cycles réguliers.

Cela augmente la productivité de l'équipe de développement concerné.

Dans le cas de notre projet de fin d'études, on a choisi de travailler avec cette méthodologie pour la raison que l'équipe de développement de la société d'accueil travaille en Scrum pour la plupart de leurs projets.

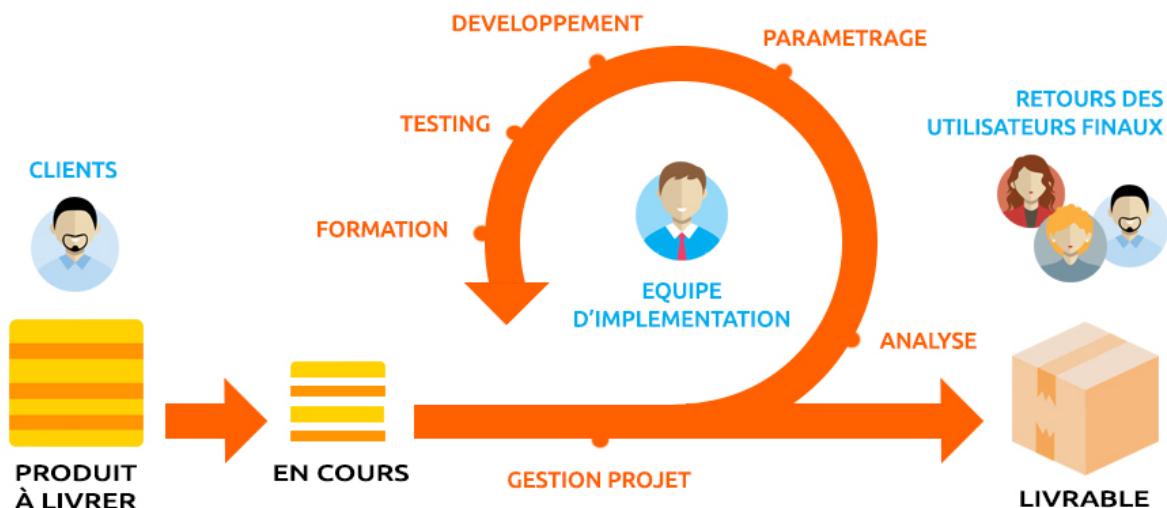


FIGURE 1.2 – La méthode Scrum<sup>4</sup>

#### Valeurs Scrum

”Lorsque les valeurs d'engagement, de courage et de concentration sont incarnées et pratiquées par l'équipe Scrum, les piliers Scrum de transparence, d'inspection et d'adaptation apparaissent et renforcent la confiance entre toute les membres de l'équipe. Ces membres d'équipe apprennent à explorer ces valeurs qu'ils travaillent avec les événements, les artefacts et les rôles de Scrum.”<sup>1</sup>

### Rôles de l'équipe Scrum<sup>3</sup>

- **Product Owner** : Le Product Owner est la personne responsable de l'augmentation de la valeur du produit final issu du travail de l'équipe Scrum. Il peut jouer de différents rôles selon les organisations, les équipes Scrum et les individus.

En plus, il est responsable de la gestion du Backlog Produit.

- **Scrum Master** : Le Scrum Master est la personne chargé de faciliter et de soutenir l'équipe Scrum comme défini dans le Guide Scrum. Les Scrum Masters remplissent leur rôle en portant aide à tout le monde pour leurs faire comprendre la théorie, les pratiques, et les règles de Scrum. Le Scrum Master est le leader de l'équipe Scrum. Le Scrum Master aide les personnes extérieures à l'équipe Scrum à identifier les interactions bénéfiques pour elle. Le Scrum Master aide tout le monde à adapter leurs interactions avec l'équipe Scrum pour maximiser la valeur créée par cette équipe.

- **L'équipe de Développement** : Elle se compose des individus professionnels qui fournissent un incrément "Fini" potentiellement publiable (Releasable) à la fin de chaque Sprint. Cet incrément est requis à la revue de sprint. Seule l'équipe de développement puisse créer l'incrément. Les équipes de développement sont structurées par l'organisation à gérer leur propre travail sans ordre. La synergie qui résulte de cette méthode nous optimise l'efficacité et le rendement globale de l'équipe de travail.

#### 1.4.2 Langages de modélisation

On a eu recours au langage de modélisation *UML* (Unified Modeling Language) pour la conception du système.

Il est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système, c'est couramment utilisé en développement logiciel et en conception orientée objet.

## Conclusion

Dans ce chapitre, on a énoncé le contexte de notre projet, on a de plus dressé notre problématique et présenté la solution proposée à la société tout en justifiant notre choix de méthodologie de gestion de ce projet.

# Chapitre 2

## Spécification de besoins

### Introduction

Dans le premier chapitre, on a commencé par parler de notre problématique et la solution envisagée, nous allons adopter la méthode Scrum pour la conception.

Ce chapitre va porter sur l'extraction des besoins et l'identification des acteurs ensuite on va procéder à détailler notre Backlog du produit qui est une étape cruciale dans notre projet servant à déterminer les besoins à fournir pour les utilisateurs de l'application.

### 2.1 Capture des besoins

La capture des besoins est une phase importante qui permet au client de bien exprimer ses besoins.

Nous avons établi plusieurs réunions avec le Product Owner pour nous mettre d'accord sur les différents acteurs et leurs besoins.

#### 2.1.1 Identification des acteurs

- **Internaute :** ce sont les visiteurs qui peuvent être soit :

- De simple internaute qui désirent avoir des informations globales sur le projet.
- Des Institutions qui voudraient trouver des informations analytique par rapport au secteur.
- des Responsable d'établissements qui s'intéresse à notre solution et qui désirent en savoir plus.
- Des acheteurs du secteur qui désirent devenir client du projet.

- **Gestionnaire :** les différentes équipes qui vont gérer les différents départements de l'application (ramassage, gestion, commercial, suivi, technique,...) au sein de la société.

- **Responsable d'établissement :** c'est le client qui va acheter la solution complète RE :SCHOOL ECOLOGY pour la gestion et le recyclage de ses déchets scolaire, il va suivre le progrès de la collecte à partir de l'application.

### 2.1.2 Besoins fonctionnels

- **Authentification :**  
elle permet aux acteurs de s'identifier pour accéder à l'application.
- **Visualisation de carte géographique :**  
elle permet la visualisation concrète des positions et états du parc poubelle dans les différents établissements et secteur.
- **Gestion du parc poubelle :**  
permet aux gestionnaires de faire le suivi du parc dans les différents établissements.
- **Gestion des ouvriers :**  
permet aux gestionnaires de suivre l'évolution journalière de la collecte et de la gestion sur le terrain (routes ,équipes, ramassage, problème, ...)
- **Gestion des ressources humaines :**  
permet aux gestionnaires de suivre le planning des équipes.
- **Génération des statistiques :**  
permet aux acteurs d'avoir accès aux statistiques de RE :SCHOOL ECOLOGY.
- **Gestion des comptes :**  
c'est la gestion des comptes utilisateurs.
- **Gestion de stock :**  
aide les gestionnaires à savoir les quantités disponibles à l'instant T.

### 2.1.3 Besoins non fonctionnels

Les exigences non fonctionnelles sont des besoins qui caractérisent le système en matière de performance. Pour rendre le travail efficace aux utilisateurs, il est important de répondre aux exigences de qualités suivants :

- Le symbole d'une poubelle doit changer selon son statut
- Les mots de passe des comptes doit être de longueur minimale égale à 8 caractères et doit contenir au moins une lettre majuscule, une minuscule et un chiffre
- Visualisation du Dashboard en temps réel
- Visualisation de la carte en temps réel
- **Maintenabilité et scalabilité** : Le code de l'application doit être lisible et compréhensible afin d'assurer leur état évolutif et extensible par rapport aux besoins du marché.
- **La convivialité** : l'application doit être simple et facile à manipuler même par des non experts.
- **L'ergonomie** : le thème adopté par l'application doit être inspiré des couleurs et du logo type de l'entreprise d'accueil.
- **Responsive Design** : Le design de l'application web doit être adapté à toutes les tailles et résolutions d'écrans.

- **L'interface** : l'application respecte les principes des Interfaces Homme/Machine tels que l'ergonomie et la fiabilité
- **La sécurité** : l'application devra être sécurisée, les informations ne devront pas être accessibles à tout le monde. L'application est accessible par un identifiant et un mot de passe attribué à une personne physique. (Le mot de passe des comptes doit être de longueur x et doit contenir au moins une lettre majuscule, une minuscule et un chiffre)
- **L'extensibilité** : l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

## 2.2 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation, est un diagramme permettant la formalisation et la modélisation des besoins utilisateurs du système informatique en conception, c'est un outil de brainstorming pour l'équipe de travail. Il est un moyen de communication entre équipe de développement et client, et entre les membres de l'équipe de développement entre eux. L'identification des cas d'utilisation, nous donne un aperçu des fonctionnalités futures que nous devons implémenter dans le système.

Ci-dessous nous trouvons notre diagramme de cas d'utilisation global :

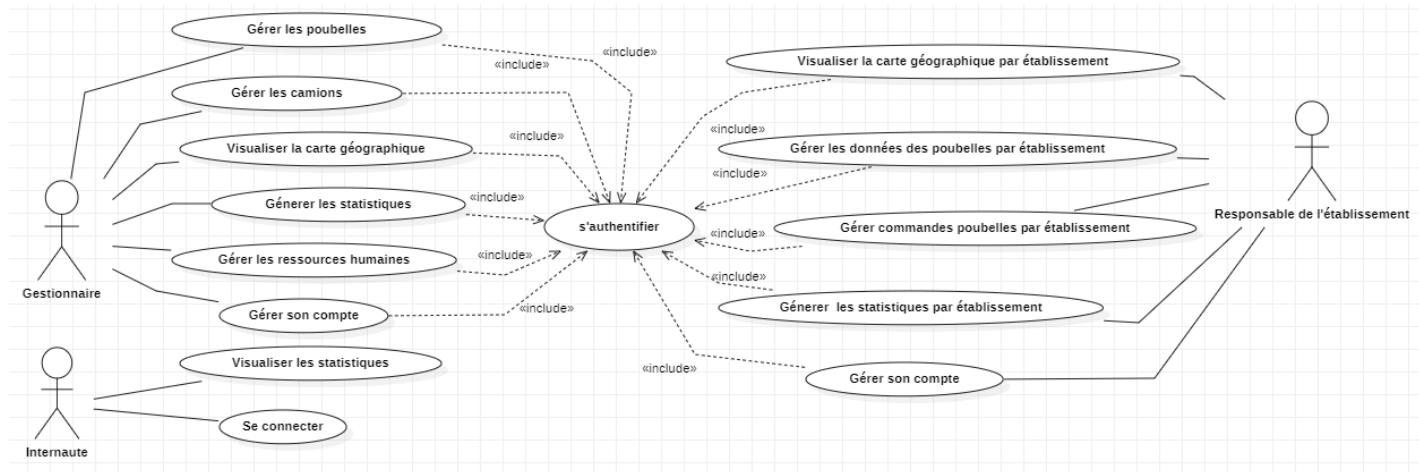


FIGURE 2.1 – Diagramme de cas d'utilisation global

## 2.3 Le Backlog du produit

Le backlog du produit est l'outil de travail principal du Product Owner et n'est autre qu'une liste ordonnée de choses à faire par l'équipe. Il recueille toutes les caractéristiques du produit et évolue avec celui-ci. Ci-dessous, le backlog du produit de notre projet :

U.S ID	User Stories	Priorité	Risques	Points (j)
[US1]	Authentification	moyenne	moyen	15
[US2]	Gestion des poubelles	haute	élevé	15
[US3]	Gestion des camions	haute	moyen	4
[US4]	Gestion des ressources humaines	moyenne	faible	9
[US5]	Gestion des comptes	faible	faible	15
[US6]	Visualisation de carte géographique	haute	élevé	12
[US7]	Génération des statistiques	haute	élevé	21
TOTAL				89 UE

TABLE 2.1 – Le Backlog du produit

## 2.4 Planification des sprints

Pour la division du projet en sprints, nous avons décidé après une longue discussion avec l'équipe de développement, qu'on divisera le travail à réaliser en 3 sprints et que leur répartition sera comme suit :

Sprint 1	Sprint 2	Sprint 3
Gestion de poubelles Gestion des camions Visualisation de carte géographique	Gestion des ressources humaines Génération des statistiques	Authentification Gestion des comptes

TABLE 2.2 – Planification des Sprints

## Conclusion

Dans ce chapitre, on a spécifié les besoins du client et des utilisateurs tout en indiquant les besoins non fonctionnels du futur système. Comme on a aussi planifié les sprints à traiter et établit notre Backlog de produit qui est un artefact principale dans la méthode Scrum. A ce niveau nous pouvons passer au prochain chapitre, qui contiendra le premier sprint et ses artefacts.

# Chapitre 3

## Sprint 1 : Gestion de poubelles et camions et Visualisation de carte géographique

### Introduction

Scrum contient une notion très importante qui désigne l'enchaînement de la réalisation du projet qui est les Sprint. Les Sprint sont des itérations d'un intervalle de temps pendant lequel l'équipe Scrum réalise un nombre de tâches en visant à atteindre un objectif convenu préalablement. Ils décomposent le projet pour le rendre plus simple et facile à réaliser.  
Pour ce sprint, on a choisi de travailler sur ces 3 cas d'utilisation. On a ci-dessous le backlog du sprint 1 en détails :

User Story	Tâches	Points (j)	Priorité
Gestion des poubelles	ajouter poubelle modifier poubelle supprimer poubelle rechercher poubelle afficher détails poubelle gérer commandes poubelles consulter historique commandes	2 1 1 2 1 2 2	haute moyenne faible moyenne faible haute faible
Gestion des camions	ajouter camion modifier camion supprimer camion rechercher camion afficher détails camion	8h 4h 4h 6h 3h	haute moyenne faible moyenne faible
Visualisation de la carte géographique	consulter les détails des poubelles par établissement consulter les détails des camions visualiser la carte géographique par établissement	5 3 2	haute moyenne haute

TABLE 3.1 – Backlog du sprint 1

### 3.1 Analyse

Cette étape a comme objectif la construction du modèle d'analyse pour préparer la phase de conception. Elle se caractérise par un diagramme de cas d'utilisation. Celui-ci permet de décrire les fonctionnalités attendues du système de point de vue utilisateur.

Pour ce sprint, on a divisé son contenu en 3 grands phases qui sont la gestion des poubelles, la gestion des camions et la visualisation de la carte géographique.

#### 3.1.1 Raffinement du cas d'utilisation Gestion des poubelles

Le cas d'utilisation "Gestion des poubelles" comporte la gestion du stock des poubelles et la gestion des données des poubelles d'un établissement qui sont assurés chacun séparément par des différents acteurs. La gestion du stock est assurée par le gestionnaire, quant à la gestion des données par établissement (comportant la partie d'achat des poubelles et la consultation de l'historique de ses achats) est assurée par le responsable de l'établissement. Ceci est mieux représenté dans la figure ci-dessous :

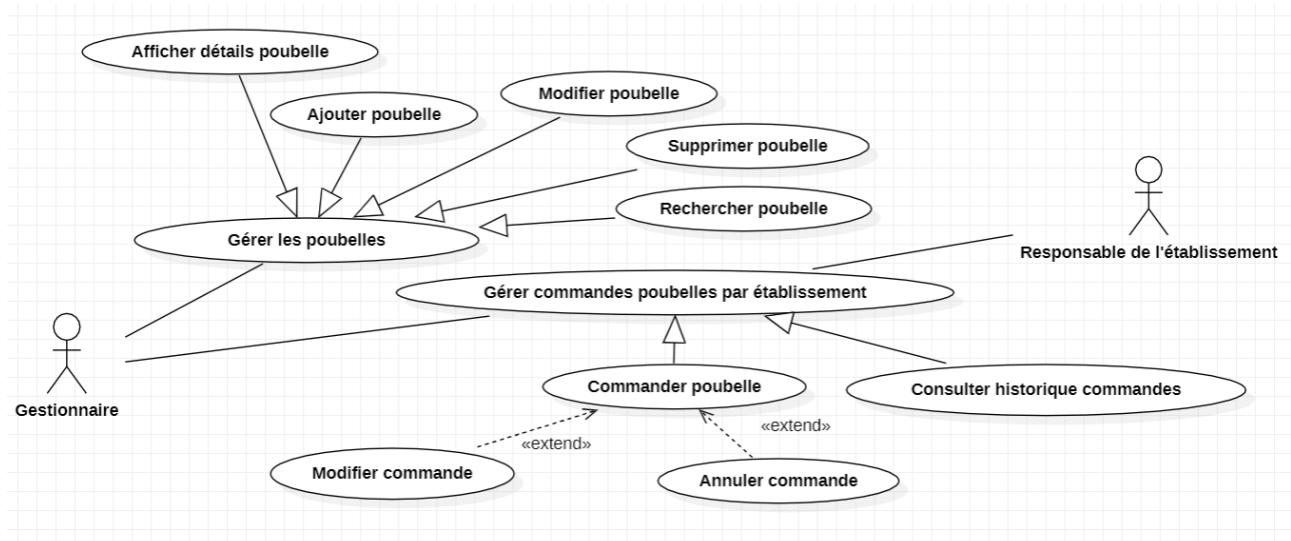


FIGURE 3.1 – Raffinement du cas d'utilisation "Gestion des poubelles"

La figure 3.1 montre que le gestionnaire peut gérer le stock des poubelles : il peut ajouter, modifier, supprimer des poubelles comme il peut réaliser une recherche et afficher tous les détails concernant la poubelle sélectionnée.

Le responsable d'établissement a le droit de gérer les données des poubelles se trouvant dans son établissement, commander des poubelles au choix, consulter l'historique des commandes établies auparavant, modifier une commande ou l'annuler tant que la commande n'a pas été confirmée par le gestionnaire.

**Description textuelle du sous cas d'utilisation «Gérer commandes poubelles»**

Cas d'utilisation	Gérer commandes poubelles
Acteur	Gestionnaire, Responsable de l'établissement
Pré-condition	L'acteur est connecté
Post-condition	L'acteur a géré les commandes
Scénario principal	1-L'acteur entre sur le dashboard 2-L'acteur sur le menu "Gérer commandes poubelles" 3-Le système affiche l'interface propre à lui
Scénario alternatif	Le système affiche qu'il n'a pas peu établit une connexion à la base de données.

TABLE 3.2 – Description textuelle du sous cas d'utilisation «Gérer commandes poubelles»

**Description textuelle du sous cas d'utilisation «Consulter historique commandes»**

Cas d'utilisation	Consulter historique commandes
Acteur	Responsable de l'établissement
Pré-condition	Le responsable est connecté et une commande a été établi auparavant
Post-condition	Le responsable a consulté son historique
Scénario principal	1- Le responsable entre sur le dashboard 2- Le responsable sur le menu "Gérer commandes poubelles" 3- Le système affiche l'interface propre à lui 4- Les commandes effectués sont affichés sur l'écran
Scénario alternatif	Le système affiche qu'il n'a établit aucune commande.

TABLE 3.3 – Description textuelle du sous cas d'utilisation «Consulter historique commandes»

**Description textuelle du sous cas d'utilisation «Commander poubelles»**

Cas d'utilisation	Commander poubelles
Acteur	Responsable de l'établissement
Pré-condition	Le Responsable de l'établissement est connecté
Post-condition	Le Responsable de l'établissement a effectué une commande
Scénario principal	1-Le responsable entre sur le dashboard 2-Le responsable clique sur le menu "Produits" 3-Le système affiche l'interface propre à lui 4-Les produits disponibles sont affichés sur l'écran
Scénario alternatif	Les produits affichés deviennent non cliquables et un message d'indisponibilité s'affiche commande.

TABLE 3.4 – Description textuelle du sous cas d'utilisation «Commander poubelles»

### Description textuelle du sous cas d'utilisation «Annuler commande poubelles»

Cas d'utilisation	Annuler commande poubelles
Acteur	Responsable de l'établissement
Pré-condition	Le responsable de l'établissement a passé une commande
Post-condition	Le responsable de l'établissement a annulé sa commande
Scénario principal	1- Le responsable entre sur le dashboard 2- Le responsable clique sur le menu "Gérer commandes" 3- Le système lui affiche l'interface "Gérer commandes" 4- Les commandes effectuées seront affichées 5- Le responsable choisit la commande à annuler 6- Le responsable clique sur le bouton "Annuler" de la commande 7- Le système lui affiche un message de confirmation d'annulation
Scénario alternatif	Le système affiche qu'il n'a établit aucune commande.

TABLE 3.5 – Description textuelle du sous cas d'utilisation «Annuler commande poubelles»

### Description textuelle du sous cas d'utilisation «Modifier commande poubelles»

Cas d'utilisation	Modifier commande poubelles
Acteur	Responsable de l'établissement
Pré-condition	Le responsable de l'établissement a passé une commande
Post-condition	Le responsable de l'établissement a modifié sa commande
Scénario principal	1- Le responsable entre sur le dashboard 2- Le responsable clique sur le menu "Gérer commandes" 3- Le système lui affiche l'interface "Gérer commandes" 4- Les commandes en cours de procès seront affichées 5- Le responsable choisit la commande à modifier 6- Le responsable clique sur le bouton "Modifier" de la commande 7- Le système lui affiche un message de confirmation de modification
Scénario alternatif	-Le système affiche qu'il n'a établit aucune commande.

TABLE 3.6 – Description textuelle du sous cas d'utilisation «Modifier commande poubelles»

### 3.1.2 Raffinement du cas d'utilisation Gestion des camions

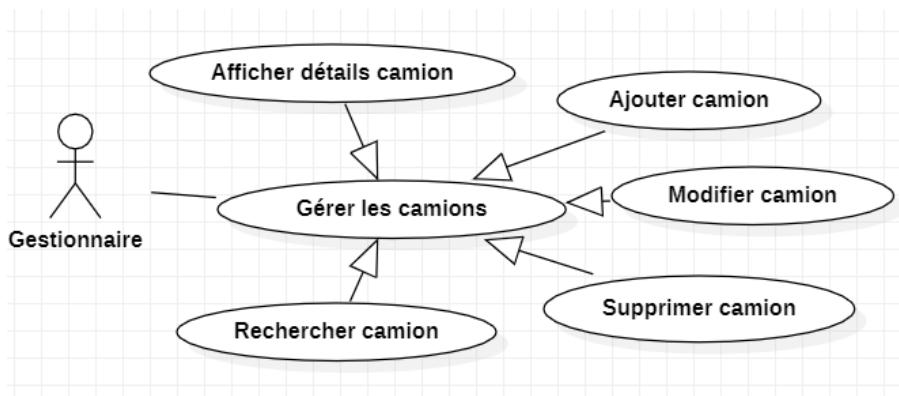


FIGURE 3.2 – Raffinement du cas d'utilisation "Gérer les camions"

## CHAPITRE 3. SPRINT 1 : GESTION DE POUBELLES ET CAMIONS ET VISUALISATION DE CARTE GÉOGRAPHIQUE

---

Les sous-cas d'utilisation d'ajout, modification, suppression, recherche et affichage sont les mêmes pour la gestion des poubelles et la gestion des camions donc on a décidé à travailler ces scénarios pour un seul raffinement de cas pour éviter de trop répéter les sous-cas.

Ainsi on a opté à les travailler seulement dans le raffinement du cas d'utilisation "Gérer les camions".

### **Description textuelle du sous cas d'utilisation «Ajouter camion»**

Cas d'utilisation	Ajouter camion
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a ajouté une camion
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Camions" 3- Le système affiche l'interface "Camions" propre à lui 4- Le gestionnaire clique sur le bouton "Ajouter" 5- Le système lui affiche une nouvelle interface d'ajout 6- Le gestionnaire entre les données nécessaires du camion 7- Le gestionnaire puis confirme l'ajout 8- Le système affiche un message de la réussite de l'ajout
Scénario alternatif	- Le système notifie que l'ajout n'a pas été établi avec succès

TABLE 3.7 – Description textuelle du sous cas d'utilisation «Ajouter camion»

### **Description textuelle du sous cas d'utilisation «Modifier camion»**

Cas d'utilisation	Modifier camion
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a modifié une camion
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Camions" 3- Le système affiche l'interface "Camions" propre à lui 4- Le gestionnaire clique sur le bouton "Modifier" 5- Le système lui affiche une nouvelle interface de modification 6- Le gestionnaire entre les nouvelles données du camion 7- Le gestionnaire puis la confirme 8- Le système affiche un message de la réussite de modification
Scénario alternatif	Le système affiche un message d'erreur de modification

TABLE 3.8 – Description textuelle du sous cas d'utilisation «Modifier camion»

### Description textuelle du sous cas d'utilisation «Supprimer camion»

Cas d'utilisation	Supprimer camion
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a supprimé le camion
Scénario principal	1- Le gestionnaire clique sur le menu "Camions" 2- Le système affiche l'interface "Camions" propre à lui 3- Le gestionnaire clique sur le bouton "Supprimer" 4- Le système lui affiche une nouvelle interface de confirmation de suppression 5- Le système affiche un message de la réussite de suppression
Scénario alternatif	Le système affiche un message d'erreur de suppression

TABLE 3.9 – Description textuelle du sous cas d'utilisation «Supprimer camion»

### Description textuelle du sous cas d'utilisation «Rechercher camion»

Cas d'utilisation	Rechercher camion
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a trouvé le camion qu'il cherche
Scénario principal	1- Le gestionnaire clique sur le menu "Camions" 2- Le système affiche l'interface "Camions" propre à lui 3- Le gestionnaire clique sur la zone de recherche 4- Le gestionnaire saisie l'identifiant du camion 5- Le système lui affiche le camion recherché dans la table 6- Le système affiche un message que la recherche est réussie
Scénario alternatif	Le système affiche un message "Camion introuvable"

TABLE 3.10 – Description textuelle du sous cas d'utilisation «Rechercher camion»

### Description textuelle du sous cas d'utilisation «Afficher détails camion»

Cas d'utilisation	Afficher détails camion
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a affiché les détails du camion
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Camions" 3- Le système affiche l'interface "Camions" propre à lui 4- Le gestionnaire clique sur le bouton "Afficher" 5- Le système lui affiche les détails du camion sélectionné 6- Le gestionnaire clique sur le bouton "Annuler" pour quitter la fenêtre
Scénario alternatif	Le système n'affiche pas le bouton "Afficher"

TABLE 3.11 – Description textuelle du sous cas d'utilisation «Afficher détails camion»

### 3.1.3 Raffinement du cas d'utilisation Visualisation de la carte géographique

Le cas d'utilisation "Visualisation de la carte géographique" comporte la visualisation de la carte par le gestionnaire qui contient la consultation des localisations des poubelles et des camions avec leurs détails, et la visualisation de la carte par le responsable de l'établissement qui pourra visualiser les emplacements de ses poubelles. On peut bien visualiser ceci dans la figure ci-dessous.

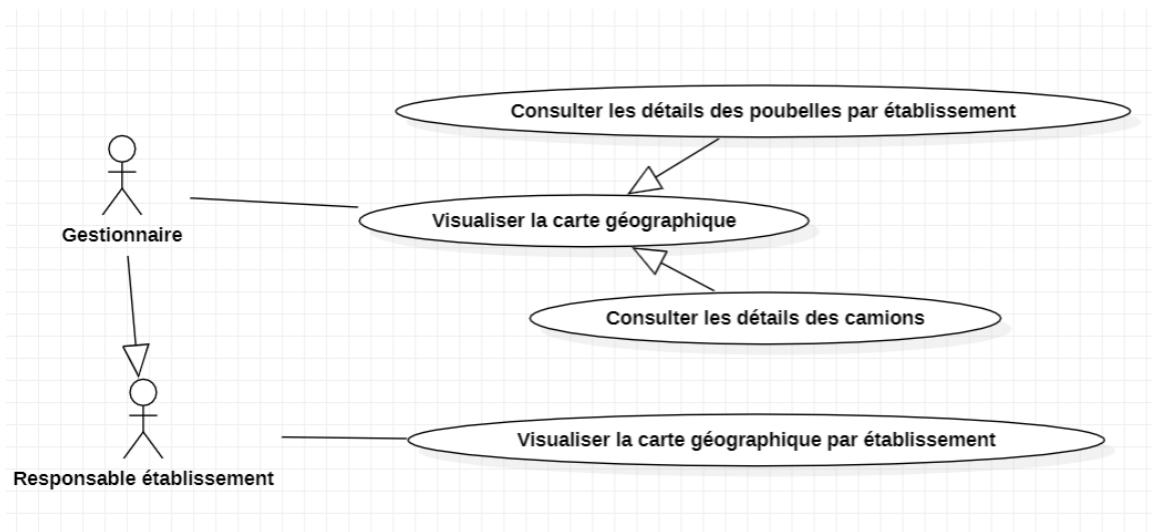


FIGURE 3.3 – Raffinement du cas d'utilisation "Visualiser carte géographique"

#### Description textuelle du sous cas d'utilisation «Visualiser la carte géographique»

Cas d'utilisation	Visualiser la carte géographique
Acteur	Gestionnaire
Pré-condition	Le gestionnaire est connecté
Post-condition	Le gestionnaire a visualisé la carte
Scénario principal	1-Le gestionnaire entre sur le dashboard 2-Le gestionnaire clique sur le menu "Map" 3-Le système affiche l'interface "Map" propre à lui 4-Les localisations des poubelles et camions sont affichés sur l'écran
Scénario alternatif	Le système affiche un message d'erreur d'affichage de la map

TABLE 3.12 – Description textuelle du sous cas d'utilisation «Visualiser la carte géographique»

**Description textuelle du sous cas d'utilisation «Consulter les détails des poubelles par établissement»**

Cas d'utilisation	Consulter les détails des poubelles par établissement
Acteur	Gestionnaire
Pré-condition	Le gestionnaire est connecté
Post-condition	Le système affiche les détails des poubelles par établissement
Scénario principal	1- L'acteur entre sur le dashboard 2- Le gestionnaire clique sur le menu "Map" 3- Le système affiche l'interface "Map" propre à lui 4- Les localisations des poubelles et camions sont affichés sur l'écran 5- Le gestionnaire clique sur un des marqueurs 6- Le système affiche les détails de l'établissement sélectionné à droite
Scénario alternatif	-Le système affiche un message d'erreur d'affichage de la map

TABLE 3.13 – Description textuelle du sous cas d'utilisation «Consulter les détails des poubelles par établissements»

**Description textuelle du sous cas d'utilisation «Consulter les détails des camions»**

Cas d'utilisation	Consulter les détails des camions
Acteur	Gestionnaire
Pré-condition	Le gestionnaire est connecté
Post-condition	Le gestionnaire a consulté les détails des camions
Scénario principal	1-Le gestionnaire entre sur le dashboard 2-Le gestionnaire clique sur le menu "Map" 3-Le système affiche l'interface "Map" propre à lui 4-Les localisations des poubelles et camions sont affichés sur l'écran 5-Le gestionnaire clique sur un des marqueurs des camions 6-Les détails des camions sont affichés sur l'écran
Scénario alternatif	Le système affiche un message d'erreur d'affichage de la carte

TABLE 3.14 – Description textuelle du sous cas d'utilisation «Consulter les détails des camions»

**Description textuelle du sous cas d'utilisation «Visualiser la carte géographique par établissement»**

Cas d'utilisation	Visualiser la carte géographique par établissement
Acteur	Responsable de l'établissement
Pré-condition	Le responsable est connecté
Post-condition	Le responsable a visualisé les localisations des poubelles
Scénario principal	1-Le responsable entre sur le dashboard 2-Le responsable clique sur le menu "Map" 3-Le système affiche l'interface "Map" propre à lui 4-Les localisations des poubelles sont affichés sur l'écran 5-Le responsable clique sur un des marqueurs de la carte 6-Le système affiche les détails des poubelles affiliées à droite
Scénario alternatif	Le système affiche un message d'erreur d'affichage de carte

TABLE 3.15 – Description textuelle du sous cas d'utilisation «Visualiser la carte géographique par établissement»

### 3.1.4 Diagrammes de séquence système de la gestion des poubelles

Diagramme de séquence système du sous cas d'utilisation «Gérer commandes poubelles »

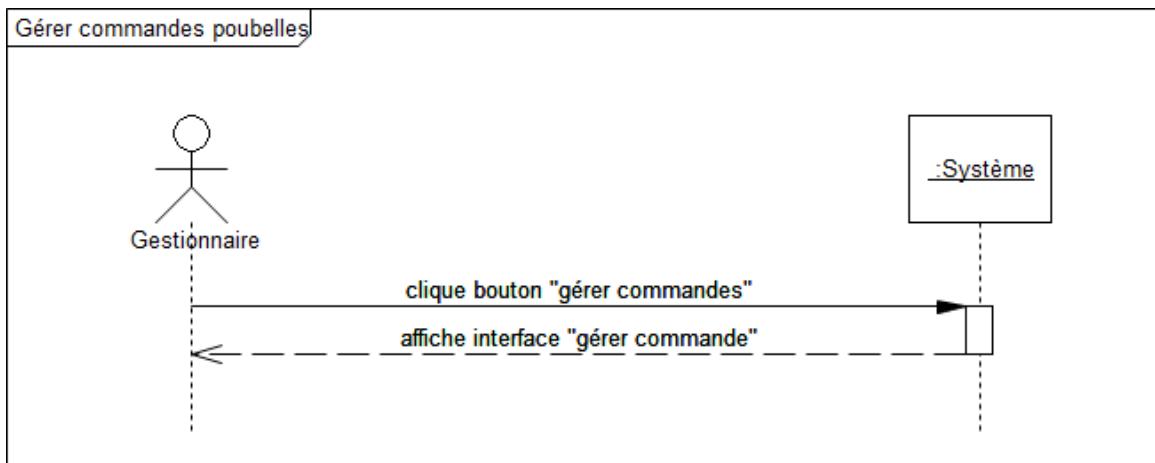


FIGURE 3.4 – Diagramme de séquence système du sous cas d'utilisation "Gérer commandes poubelles" pour le gestionnaire

Le diagramme de séquence système dans la figure 3.4 du cas d'utilisation «Gérer commandes poubelles» concerne deux acteurs : le gestionnaire et le responsable de l'établissement, conséquemment pour le responsable d'établissement c'est le même diagramme.

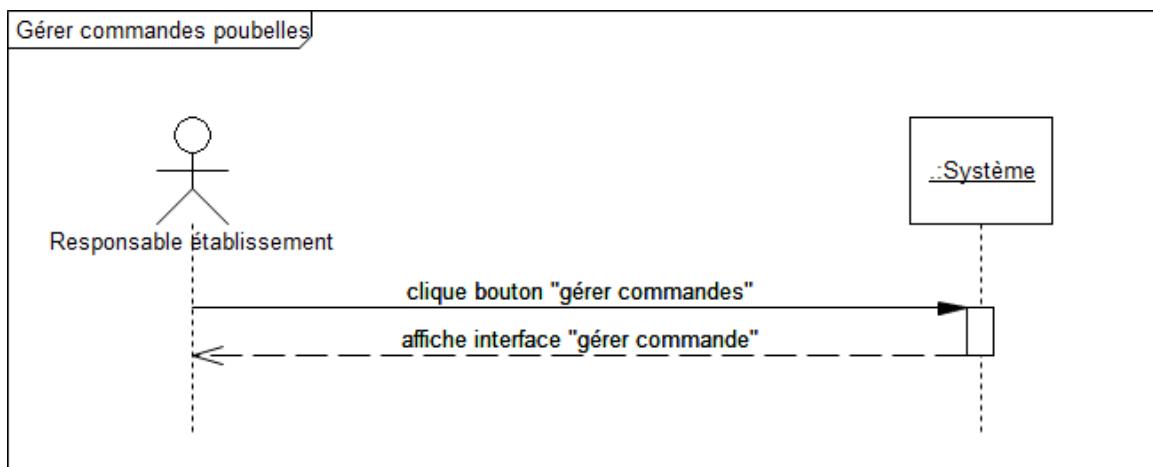


FIGURE 3.5 – Diagramme de séquence système du sous cas d'utilisation "Gérer commandes poubelles" pour le responsable d'établissement

#### Diagramme de séquence système du sous cas d'utilisation «Consulter historique commandes»

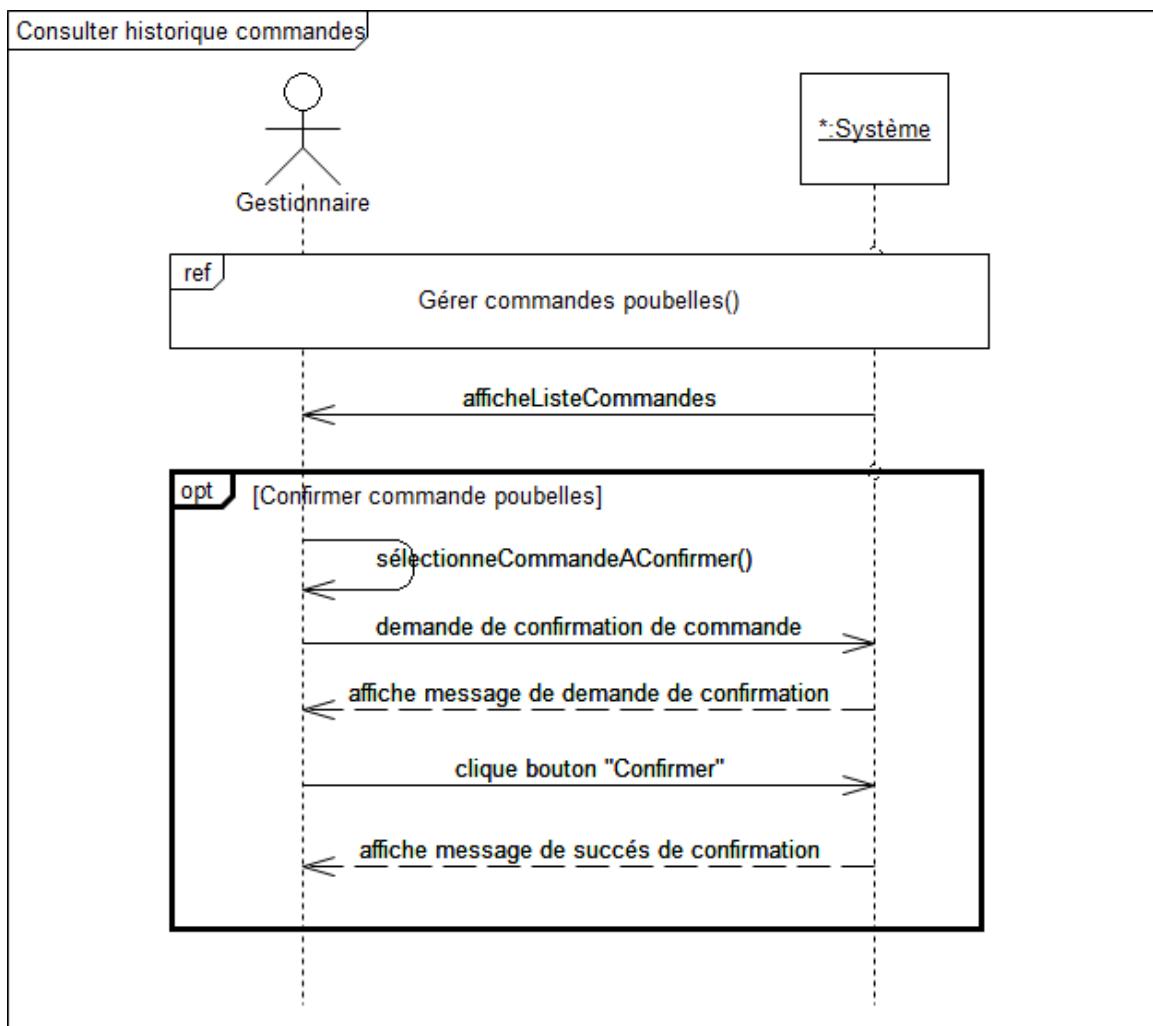


FIGURE 3.6 – Diagramme de séquence système du sous cas d'utilisation "Consulter historique commandes" pour le gestionnaire

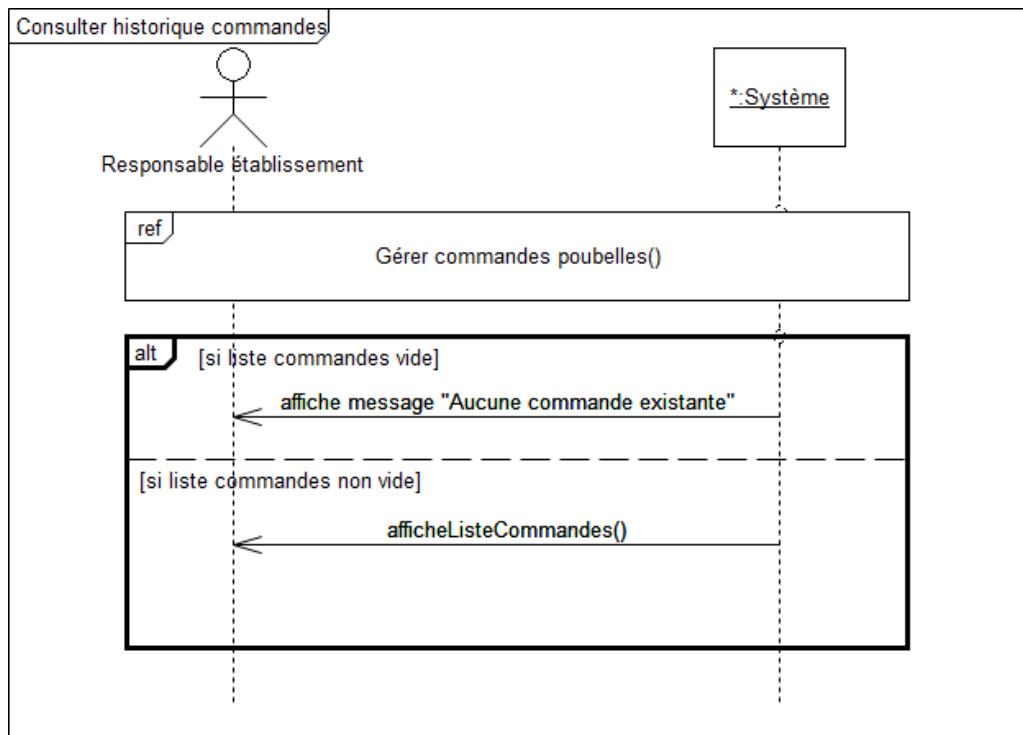


FIGURE 3.7 – Diagramme de séquence système du sous cas d'utilisation "Consulter historique commandes" pour le responsable de l'établissement

#### Diagramme de séquence système du sous cas d'utilisation «Commander poubelles»

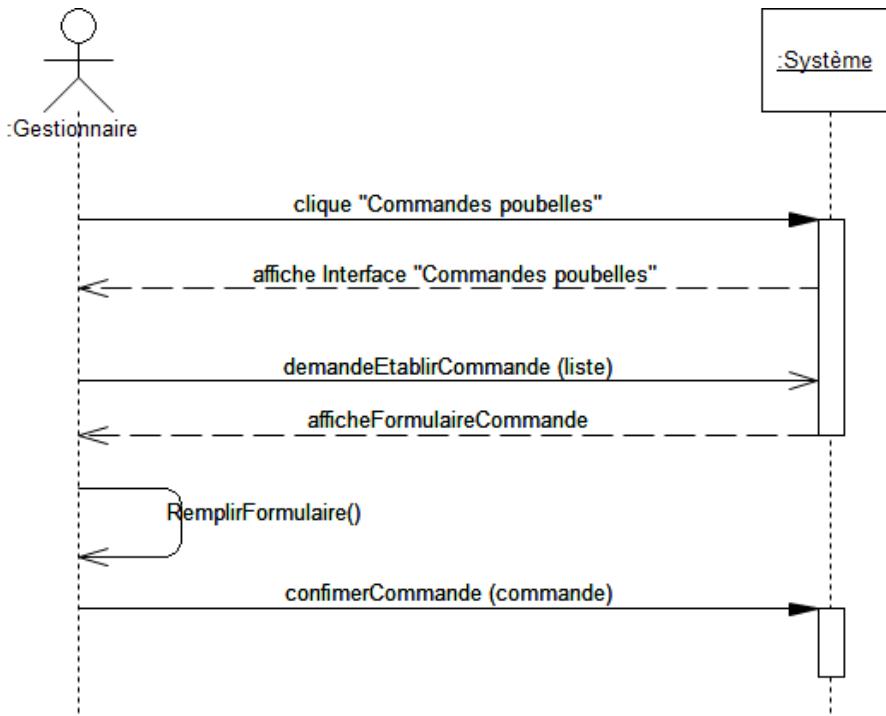


FIGURE 3.8 – Diagramme de séquence système du sous cas d'utilisation "Commander poubelles"

Diagramme de séquence système du sous cas d'utilisation « Annuler commande poubelles»

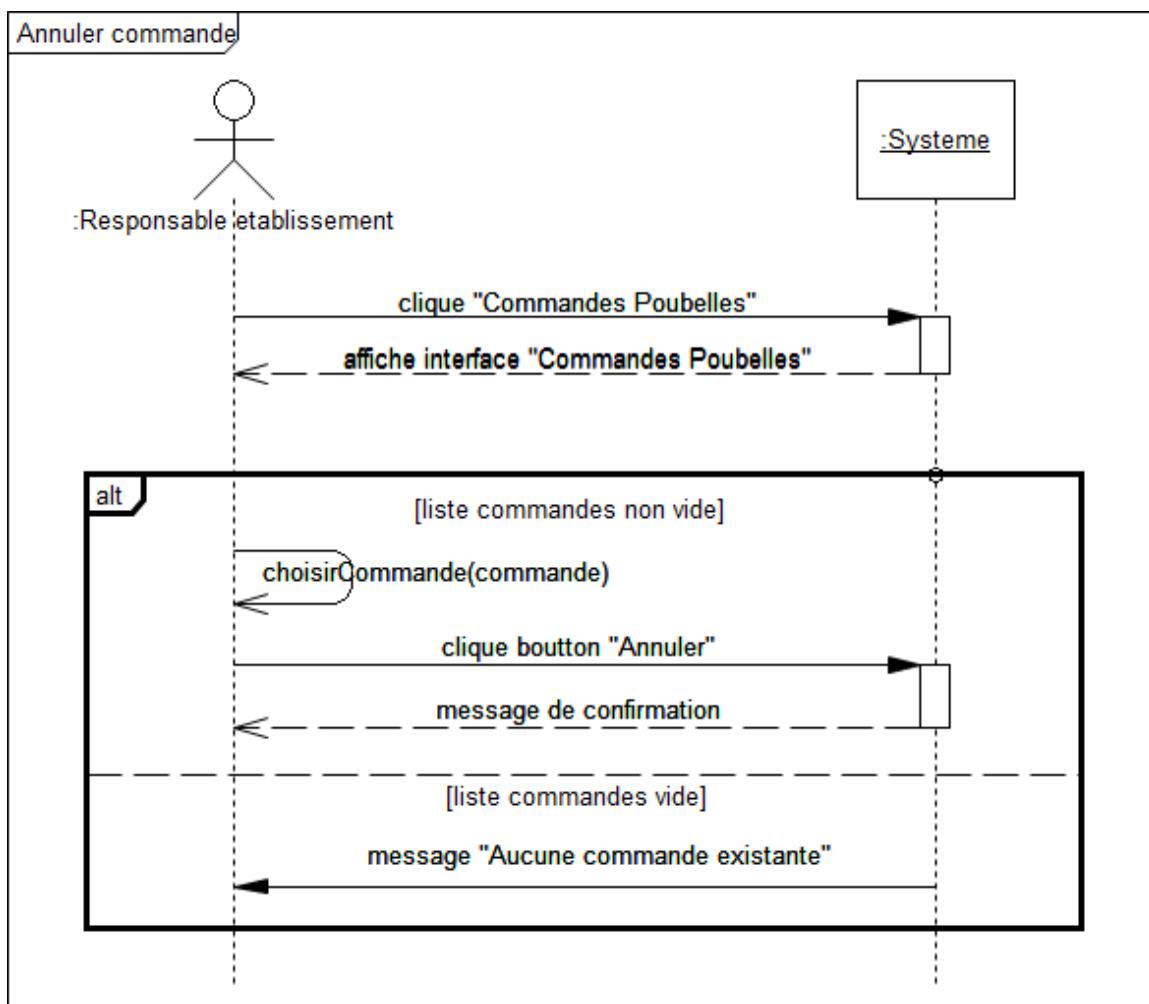


FIGURE 3.9 – Diagramme de séquence système du sous cas d'utilisation "Annuler Commande poubelles"

Le responsable ne peut annuler une commande seulement si elle en existe une. Si elle existe au moins une qui est en cours de procès, il peut donc l'annuler avant que le gestionnaire l'a confirme.

Le système lui affiche alors un message de confirmation d'annulation.

Diagramme de séquence système du sous cas d'utilisation « Modifier commande poubelles»

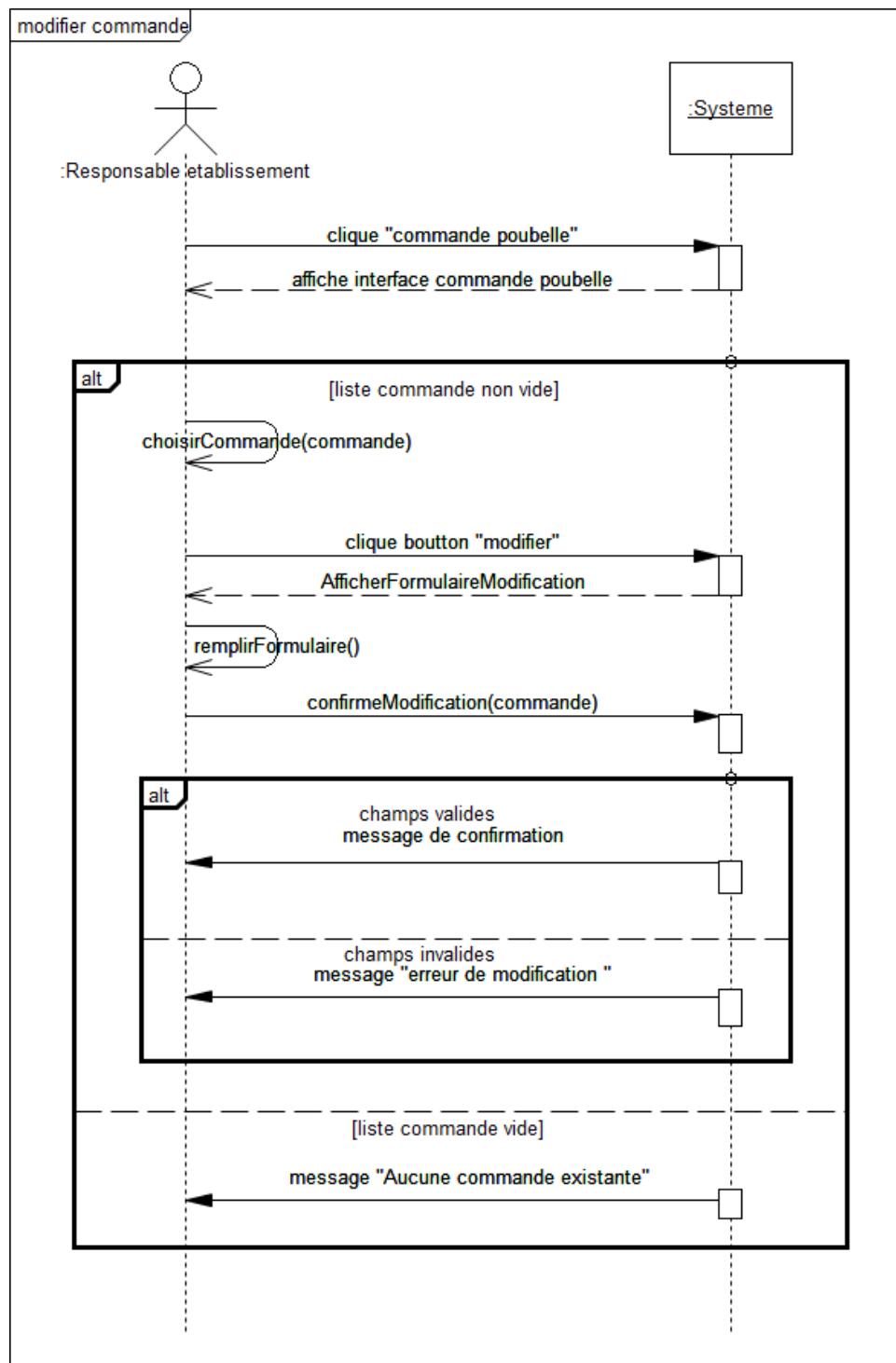


FIGURE 3.10 – Diagramme de séquence système du sous cas d'utilisation "Modifier Commande poubelles"

### 3.1.5 Diagrammes de séquence système de la visualisation de la carte géographique

Diagramme de séquence système du sous cas d'utilisation «Visualiser la carte géographique»

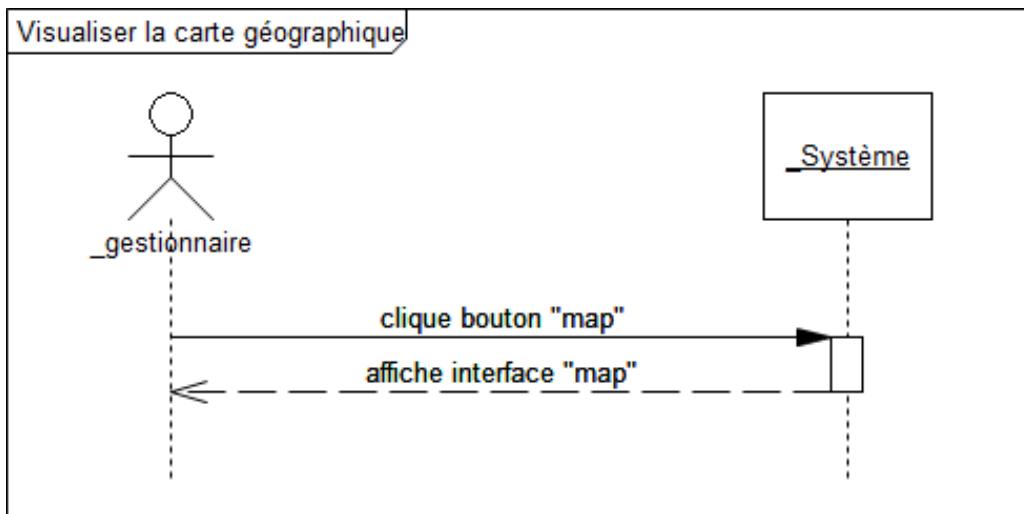


FIGURE 3.11 – Diagramme de séquence système du sous cas d'utilisation «Visualiser la carte géographique»

Diagramme de séquence système du sous cas d'utilisation «Consulter les détails des poubelles par établissement»

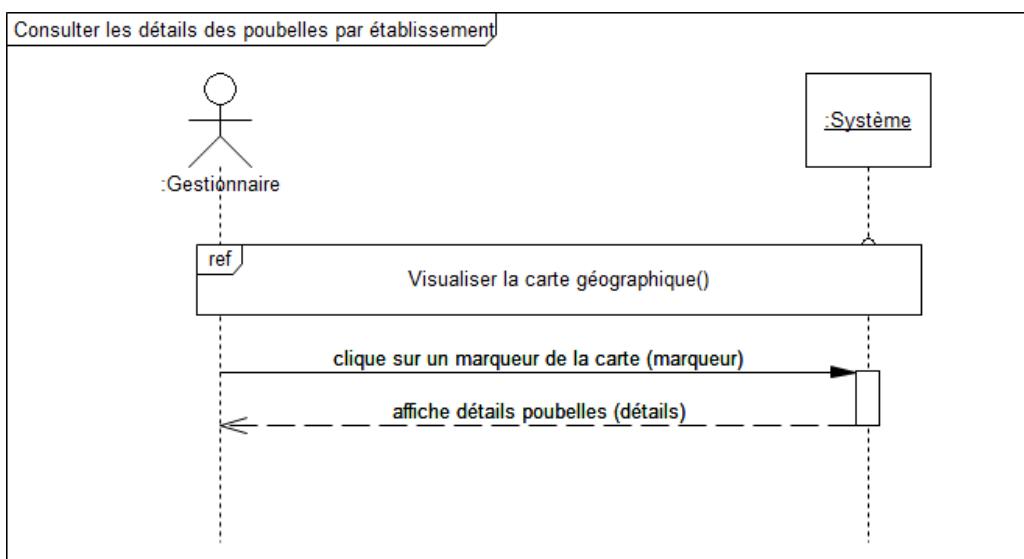


FIGURE 3.12 – Diagramme de séquence système du sous cas d'utilisation «Consulter les détails des poubelles par établissements»

**Diagramme de séquence système du sous cas d'utilisation «Consulter les détails des camions»**

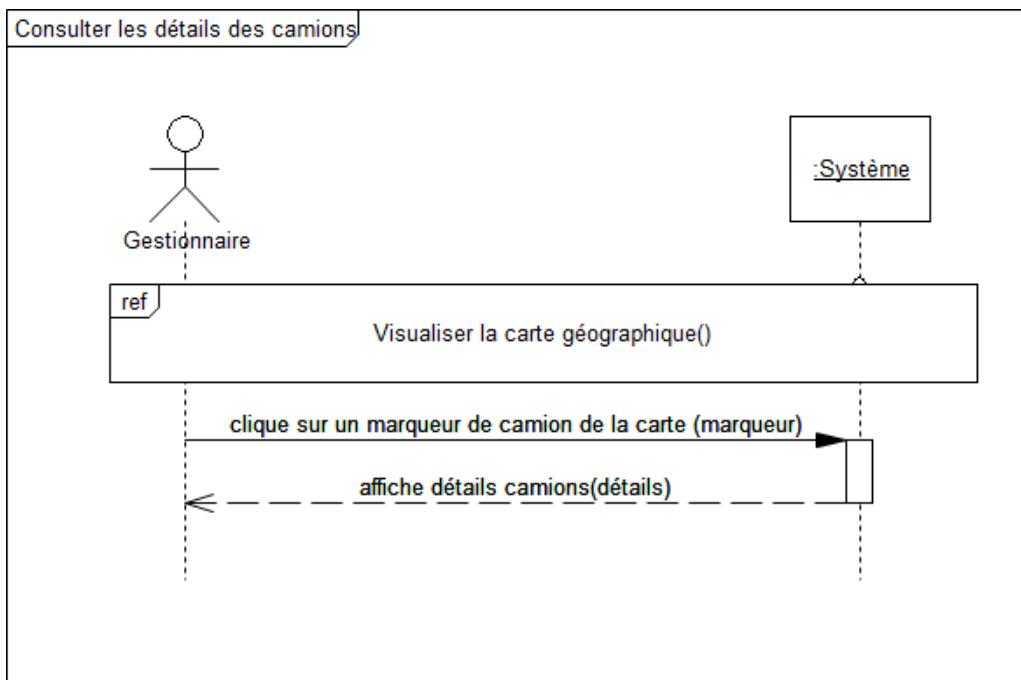


FIGURE 3.13 – Diagramme de séquence système du sous cas d'utilisation «Consulter la localisation des camions»

**Diagramme de séquence système du sous cas d'utilisation «Visualiser la carte géographique par établissement»**

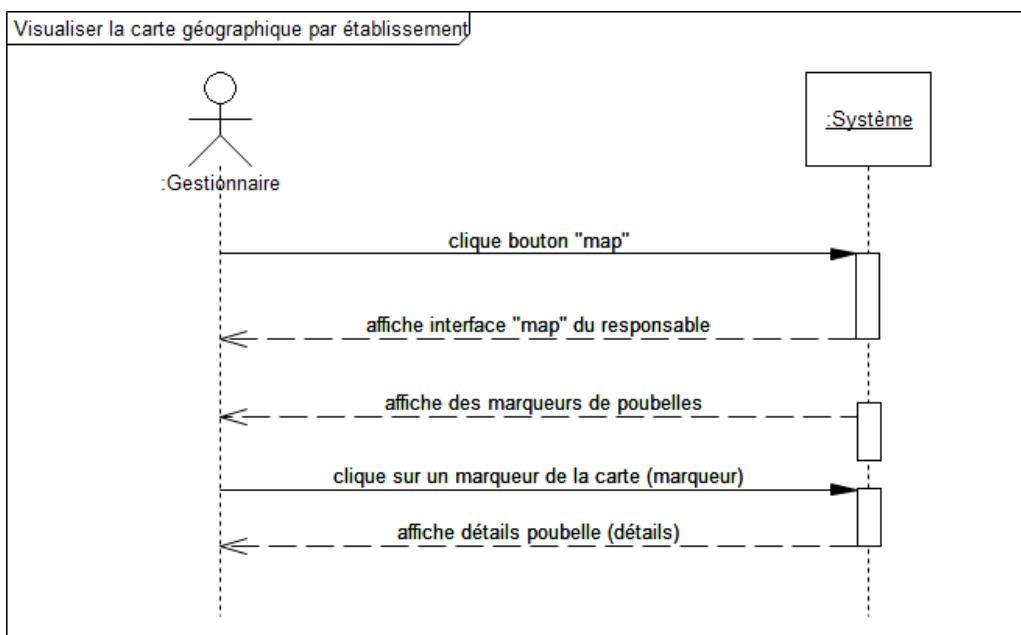


FIGURE 3.14 – Diagramme de séquence système du sous cas d'utilisation «Visualiser la carte géographique par établissement»

## 3.2 Conception

### 3.2.1 Diagramme de séquence de conception

Le diagramme de séquence système permet à l'équipe de développement de bien comprendre les échanges de messages entre les objets de classes. Il aide à avoir une idée claire de ce que le système accomplira finalement.

Dans la phase de conception, on débutera avec le cas d'utilisation "Commander poubelles".

**Diagramme de séquence de conception du sous cas d'utilisation «Commander poubelles»**

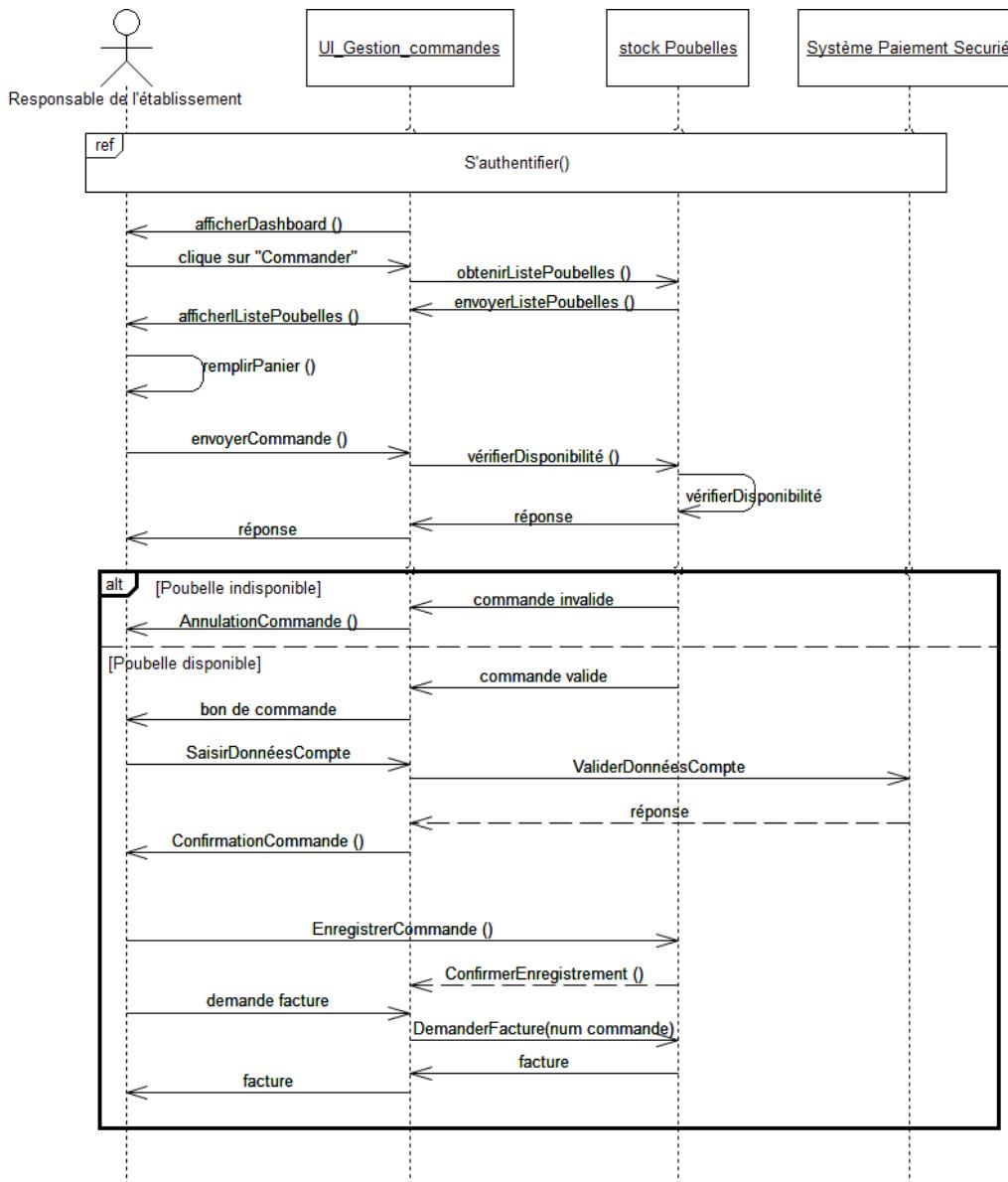


FIGURE 3.15 – Diagramme de séquence de conception du cas d'utilisation "Commander poubelles"

Ensuite on continuera vers les cas d'utilisation d'ajout, modification, suppression, recherche et affichage des camions.

Dans cette sous-section, on a réalisé ces cas pour une seule gestion or elle est applicable de

même pour la gestion des poubelles mais pour éviter la redondance, on a travaillé seulement sur la gestion des camions.

### Diagramme de séquence de conception du sous cas d'utilisation «Ajouter camion»

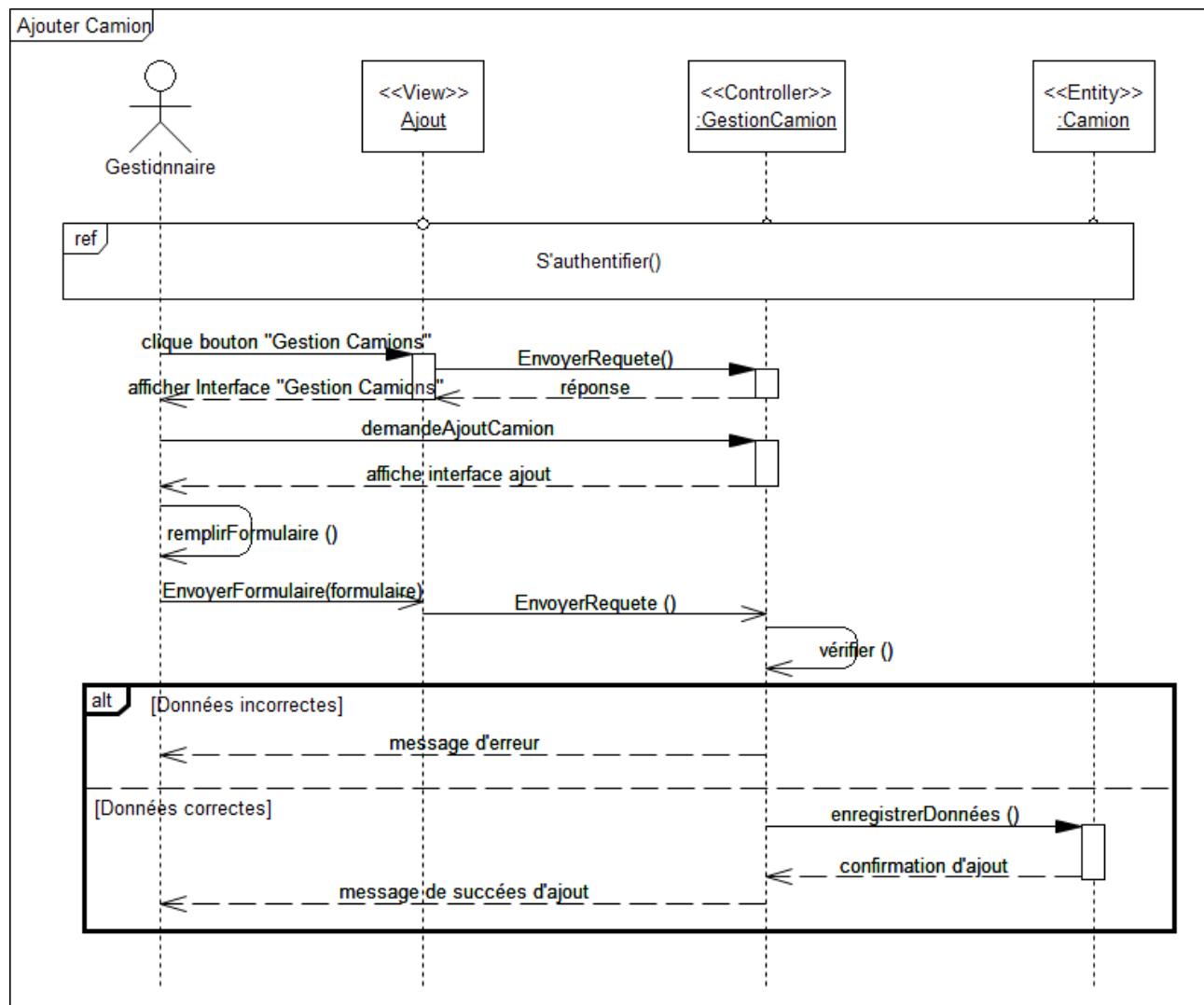


FIGURE 3.16 – Diagramme de séquence de conception du cas d'utilisation "Ajouter camion"

Diagramme de séquence de conception du sous cas d'utilisation «Modifier camion»

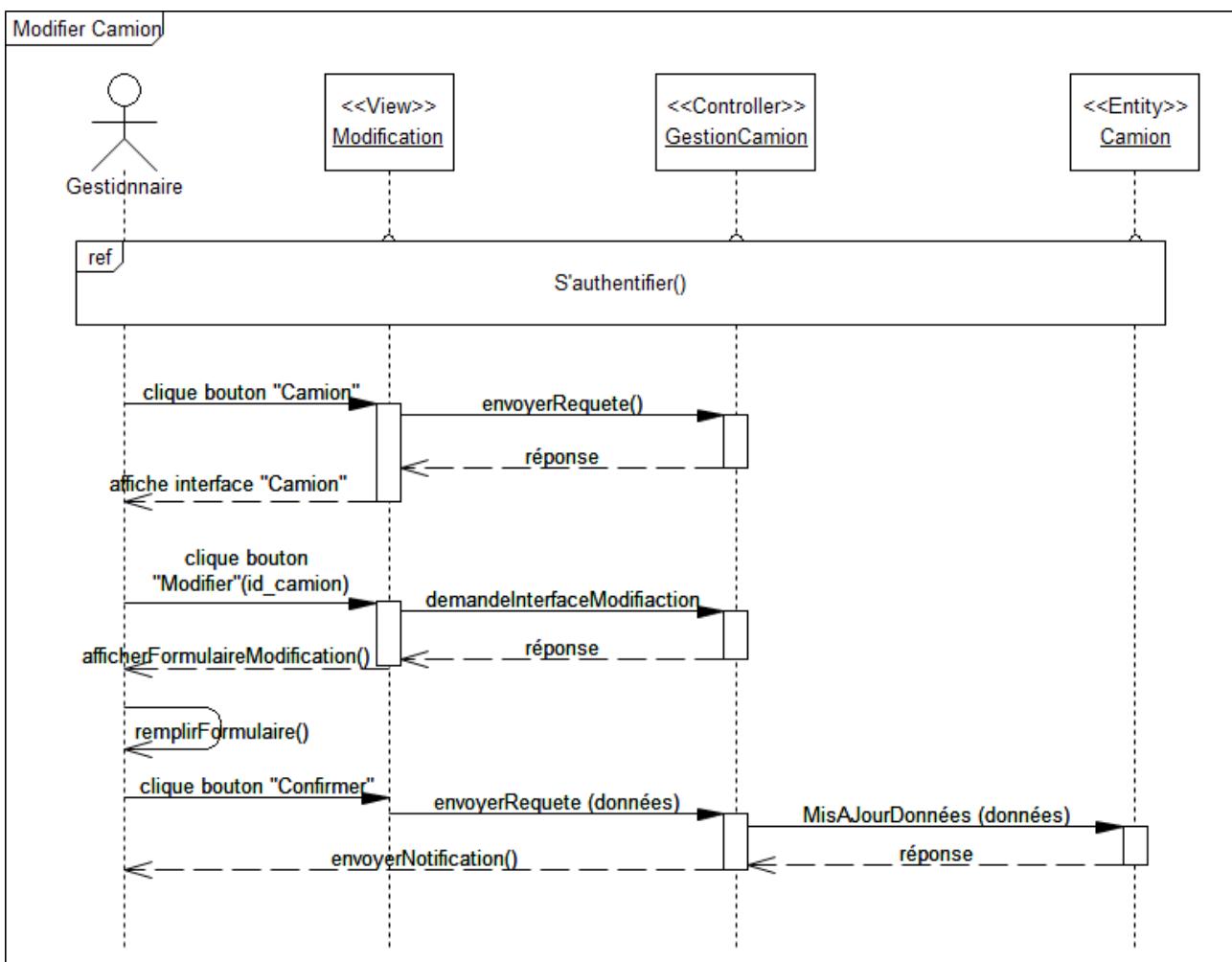


FIGURE 3.17 – Diagramme de séquence de conception du cas d'utilisation "Modifier camion"

Pour la modification des données d'un camion de la base de données, le gestionnaire doit accéder à l'interface de gestion des camions.

Un tableau lui s'affiche sur l'écran rempli des données des camions se trouvant dans notre base.

Après avoir sélectionner le camion à modifier, le gestionnaire doit cliquer sur le bouton de modification se trouvant dans la colonne "Actions" du tableau.

Une nouvelle interface de saisie de nouvelles données lui s'affiche. Il suffit maintenant de juste entrer ces nouvelles coordonnées.

Diagramme de séquence de conception du sous cas d'utilisation «Supprimer camion»

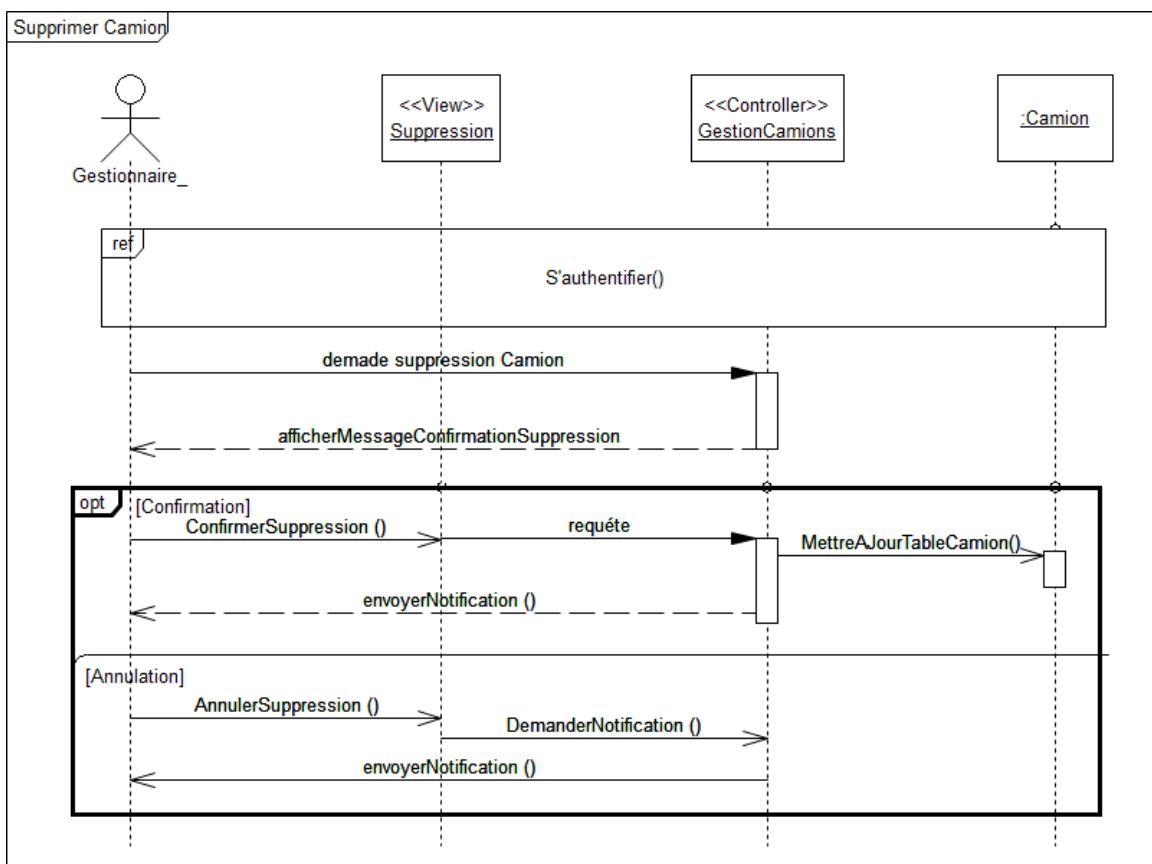


FIGURE 3.18 – Diagramme de séquence de conception du cas d'utilisation "Supprimer camion"

Diagramme de séquence de conception du sous cas d'utilisation «Rechercher camion»

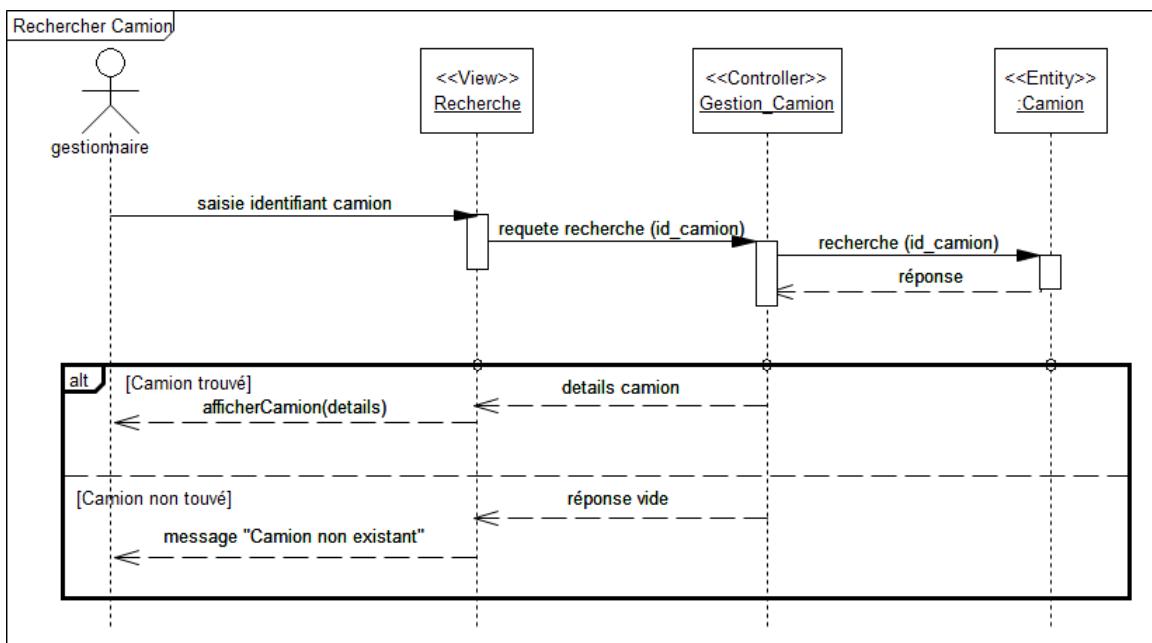


FIGURE 3.19 – Diagramme de séquence de conception du cas d'utilisation "Rechercher camion"

### Diagramme de séquence de conception du sous cas d'utilisation «Afficher détails camion»

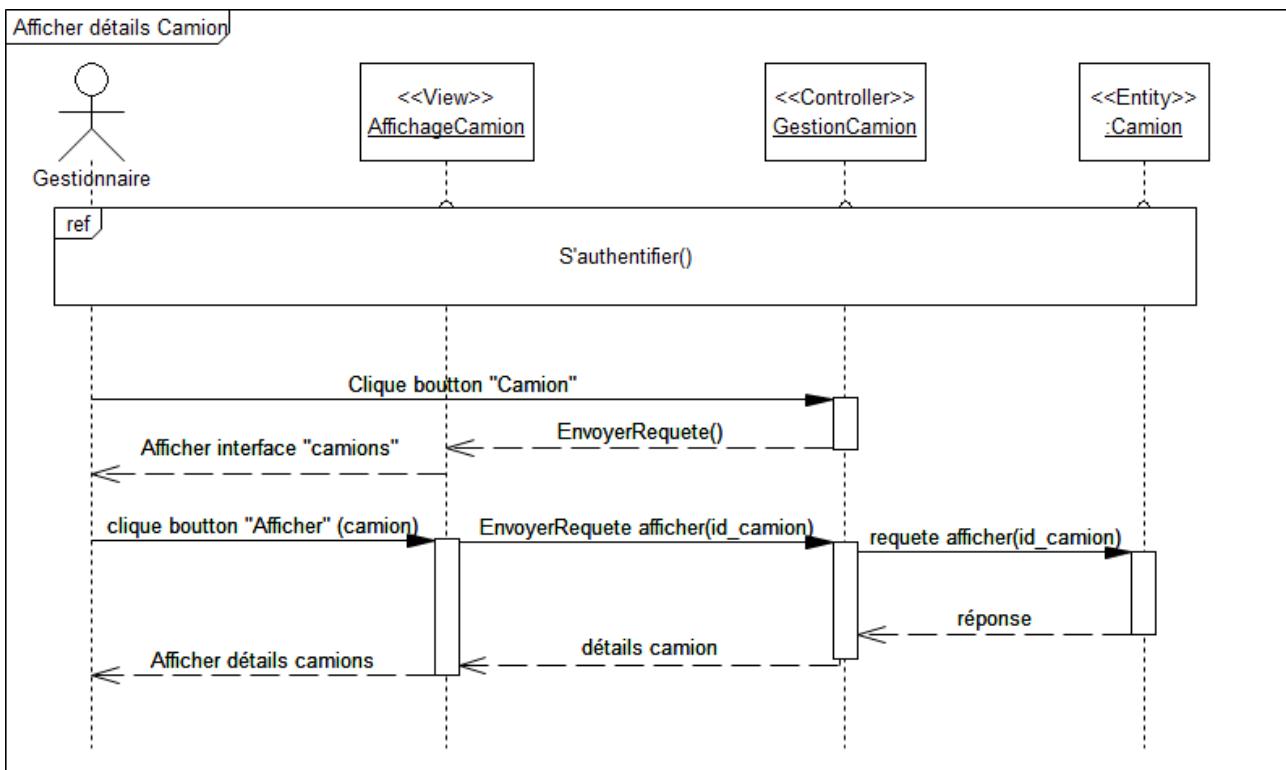


FIGURE 3.20 – Diagramme de séquence de conception du cas d'utilisation "Afficher détails camion"

### 3.2.2 Diagramme de classe global du premier sprint

Après avoir établi les diagrammes de cas d'utilisation raffinés et les diagrammes de séquence, on passe à la création de notre base de données propre à ce sprint mais pour cela on doit d'abord construire notre diagramme de classes.

Le diagramme de classes est un artefact cruciale dans l'étape de conception, il permet la modélisation de la structure statique du système.

La figure 3.21 si-dessous met en évidence les classes de notre sprint :

# CHAPITRE 3. SPRINT 1 : GESTION DE POUBELLES ET CAMIONS ET VISUALISATION DE CARTE GÉOGRAPHIQUE

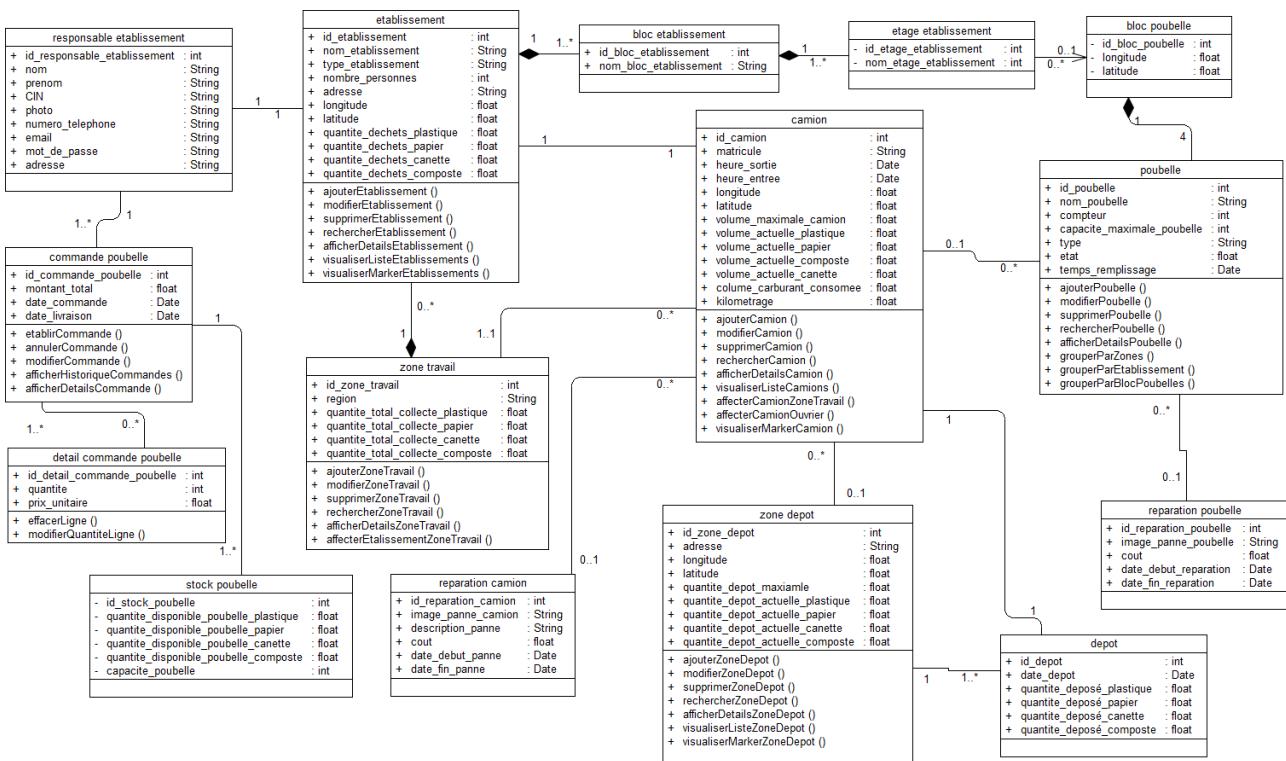


FIGURE 3.21 – Diagramme de classe global du sprint 1

## 3.3 Codage

### 3.3.1 Schéma de la base de données

Nous avons conçu le diagramme de classes de ce sprint d'où nous pouvons passer à la création de notre base de données. Nous allons commencer par convertir les classes établies dans la dernière section en tables de données. Le diagramme de classes du premier sprint est très développé donc nous choisirons quelques tables de la base réalisé :

Champs	Types	Contraintes
id	bigint(20)	Not NULL, Primary Key
bloc_poubelle_id	bigint(20)	Not NULL, Foreign Key
nom	varchar(255)	Not NULL
capacite_poubelle	int(11)	Not NULL
type	enum('plastique', 'composte', 'papier', 'canette')	Not NULL
Etat	double(8,2)	Not NULL
temps_remplissage	datetime	Not NULL

TABLE 3.16 – table poubelles

**CHAPITRE 3. SPRINT 1 : GESTION DE POUBELLES ET CAMIONS ET  
VISUALISATION DE CARTE GÉOGRAPHIQUE**

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
etage_etablissement_id	bigint(20)	Not NULL, Foreign Key

TABLE 3.17 – table bloc\_poubelles

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
bloc_etablissement_id	bigint(20)	Not NULL, Foreign Key
nom <sub>e</sub> tage <sub>e</sub> tablissement	bigint(20)	Not NULL

TABLE 3.18 – table etage\_etablissement

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
zone_travail_id	bigint(20)	Not NULL, Foreign Key
matricule	varchar(255)	Not NULL, Foreign Key
heure_sortie	datetime	Not NULL
heure_entree	datetime	Not NULL
longitude	double	Not NULL
latitude	double	Not NULL
volume_maximale_camion	double(8,2)	Not NULL
volume_actuelle_plastique	double(8,2)	Not NULL
volume_actuelle_papier	double(8,2)	Not NULL
volume_actuelle_composte	double(8,2)	Not NULL
volume_actuelle_canette	double(8,2)	Not NULL
volume_carburant_consomme	double(8,2)	Not NULL
Kilometrage	double(8,2)	Not NULL

TABLE 3.19 – table camions

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
responsable_etablissement_id	bigint(20)	Not NULL, Foreign Key
quantite	double(8,2)	Not NULL
montant_total	double(8,2)	Not NULL
date_commande	datetime	Not NULL
date_livraison	datetime	Not NULL

TABLE 3.20 – table commande\_poubelles

Champs	Types	Contraintes
id	bigint(20)	Not NULL, Primary Key
commande_poubelles_id	bigint(20)	Not NULL, Foreign Key
stock_poubelles_id	bigint(20)	Not NULL, Foreign Key
quantite	double(8,2)	Not NULL
prix_unitaires	double(8,2)	Not NULL

TABLE 3.21 – table detail\_commande\_poubelles

### 3.3.2 Extrait du code

Dans cette section nous allons essayer de montrer des parties du code développé tout au long de du travail pendant ce sprint. Nous commencerons par la partie d'affichage des données concernant les poubelles. Cette partie de code, nous avons voulu diminuer du gros nombre de lignes dans le code donc nous avons créer une composante pour les tables pour éviter la redondance du code. Cette composante est intitulé "Table.js", nous pouvons voir un extrait de son code dans la figure 3.22

```
> import { styled } from '@mui/material/styles';
>
> export const Item = styled(Paper)(({ theme }) => ...
> );
> export const defaultColDef = {resizable: true, sortable: true, flex: 1, filter: true}
> export const rowHeight = 60;
> export const columnTypes = { ... }
>
> export function Table({handleClickOpen, tableData, columnDefs}) {
>   const gridRef = useRef();
>   const [gridApi, setGridApi] = useState(null)
>   const onGridReady = (params) => { setGridApi(params) }
>   const onBtnExport = useCallback(() => { gridRef.current.api.exportDataAsCsv(); }, []);
>   const onQuickFilterChanged = useCallback(() => { ...
>     }, []);
>   const onPaginationChange = (pageSize) => {gridApi.api.paginationSetPageSize(Number(pageSize)) }
>
>   return (
>     <div>
>       <Grid container direction="row" justifyContent="space-between" alignItems="flex-start" >
>         <Item>
>           <ManageSearchIcon variant="contained" color="success" style={{marginBottom: "-5px"}} />
>           <input type="text" onInput={onQuickFilterChanged} id="quickFilter" placeholder="recherche..." />
>         </Item>
>         <Item>
>           <select onChange={(e)=>onPaginationChange(e.target.value)}>
>             <option value='5'>5</option>
>             <option value='25'>25</option>
>             <option value='50'>50</option>
>             <option value='100'>100</option>
>           </select>
>           <Button variant="contained" color="primary" onClick={onBtnExport}><FileDownloadIcon/></Button>
>           <Button variant="contained" color="success" onClick={handleClickOpen}><AddIcon/></Button>
>         </Item>
>       </Grid>
>       <div className="ag-theme-material">
>         <AgGridReact ref={gridRef} debounceVerticalScrollbar='true' rowData={tableData} columnDefs={columnDefs} defaultColDef={defaultColDef} onGridReady={onGridReady} columnTypes={columnTypes} rowHeight={rowHeight} pagination={true} paginationPageSize={5}/>
>       </div>
>     </div>
>   )
>
```

FIGURE 3.22 – Extrait du code Table.js

La figure 3.23 nous montre la partie de codage des marqueurs des poubelles, camions et zones de dépôts dont lequel on a réalisé le changement de la couleur du marqueur selon l'état de remplissage de la poubelle.

## CHAPITRE 3. SPRINT 1 : GESTION DE POUBELLES ET CAMIONS ET VISUALISATION DE CARTE GÉOGRAPHIQUE

```

if (!isLoading) return <div><Skeleton sx={{ height:500 }} animation="wave" variant="rectangular" /></div>;
return (
<div style={{display:"flex", justifyContent:"space-between"}>
  <GoogleMap mapContainerStyle={containerStyle} onMouseOver={() =>{ setActiveMarkerEtablissement(null);
    setUniqueEtablissements(null) ;setShowUniqueEtablissement(true);
    setActiveMarkerCamion(null); setUniqueCamions(null) ;setShowUniqueCamion(true);
    setActiveMarkerZoneDepot(null); setUniqueZoneDepot(null) ;setShowUniqueZoneDepot(true);}} center={center} zoom={11.2}>
    {etablissements.length==0?(etablissements.map((etb, id) =>
      <>
        <Marker key={id} onClick={() => handleActiveMarkerEtablissement(id)}
          icon={ EtablissementIcon} position={{ lat:etb.latitude, lng:etb.longitude} }> </Marker>
          {activeMarkerEtablissement === id ? [
            <InfoWindow onCloseClick={() => setActiveMarkerEtablissement(null)}
              | position={{ lat:etb.latitude+0.025, lng:etb.longitude} }>
              <div style={{divStyle}> <p>{etb.nom_etablissement}</p> </div>
            </InfoWindow>
          ] : null}
        </>):<div style={{color:"red"}}>pas des etablissements</div>)
      }
      {camions.length==0?(camions.map((camion, id) => ...
        </> ) :<div style={{color:"red"}}>pas des camions</div>
      )
      { zoneDepots.length==0?(zoneDepots.map((zone, id) => ...
        </> ) :<div style={{color:"red"}}>pas des zones</div>
      )
    </GoogleMap>

```

FIGURE 3.23 – Extrait du code MapGestionnaire.js

Une deuxième partie a été extraite du code de la composante "MapGestionnaire.js", c'est la partie de la récupération des données avec l'intermédiaire des APIs (Application Programming Interface) et elle est exposé dans la figure 3.23.

```

✓ export default function MapGestionnaire() {
  const { isLoading } = useLoadScript({ ... });
  const [etablissements, setEtablissements] = useState([]);
  const [camions, setCamions] = useState([]);
  const [zoneDepots, setZoneDepots] = useState([]);

  useEffect(() => {
    fetch("http://127.0.0.1:8000/api/google-map").then((e) => { return e.json(); }).then((data) => { setEtablissements(data); });
    fetch("http://127.0.0.1:8000/api/google-map-camion").then((e) => { return e.json(); }).then((data) => { setCamions(data); });
    fetch("http://127.0.0.1:8000/api/zone-depot").then((e) => { return e.json(); }).then((data) => { setZoneDepots(data.data); });
  },[])
  const [activeMarkerEtablissement, setActiveMarkerEtablissement] = useState(null);
  async function handleActiveMarkerEtablissement (marker) {
    if (marker === activeMarkerEtablissement) {
      return;
    }
    setActiveMarkerEtablissement(marker);
    if(marker!=null){
      await fetch(`http://127.0.0.1:8000/api/google-map/${marker+1}`)
      .then((e) => {
        return e.json();
      })
      .then((data) => {
        setUniqueEtablissements(data);
      })
    }
  };
}

```

FIGURE 3.24 – Extrait du code MapGestionnaire.js

## 3.4 Test

Le test est une partie primordiale dans la méthode Agile pour que le développement se déroule dans les normes et sans fautes. Donc il est très important de passer par cette partie au cours de notre processus.

Nous allons commencer à tester le code de la partie de la gestion des poubelles avec la totalité de ses fonctionnalités en utilisant l'acteur "gestionnaire" comme exemple.

The screenshot shows the RE:SCHOOL Ecology application interface. On the left, there is a sidebar menu with various icons and labels: Dashboard, Map, Gestion poubelles (highlighted with a red arrow), Gestion camions, Production poubelle, Personnel, Clients, Commandes, Pannes, Calendrier, Gestionnaire liste, and Contact-us. The main content area has a green header bar with tabs: GROUP, ZONE TRAVAIL, ETABLISSEMENT, BLOCS ETABLISSEMENT, ETAGES ETABLISSEMENT, BLOCS POUBELLES, and POUBELLES. Below the header is a search bar and a table titled "Details par zone travail". The table has columns: Id, Region, Quantite total collecte plastique, Quantite total collecte papier, Quantite total collecte composte, Quantite total collecte canette, created\_at, and updated\_at. There are 5 rows of data:

Id	Region	Quantite total collecte plastique	Quantite total collecte papier	Quantite total collecte composte	Quantite total collecte canette	created_at	updated_at
> 1	Tunis	10000	7000	5000	8000	17/05/2022	17/05/2022
> 2	Ariana	4000	5000	200	600	17/05/2022	17/05/2022
> 3	Ben Arous	700	5000	100	4000	17/05/2022	17/05/2022
> 4	Mannouba	5000	1500	500	500	17/05/2022	17/05/2022
> 5	Nabeul	0	0	0	0	17/05/2022	17/05/2022

At the bottom of the table, there are pagination controls: lignes par page: 5 lignes, 1-5 de 24, and navigation arrows.

FIGURE 3.25 – Interface de la table Poubelles groupées par zones de travail

Le tableau figurant dans la figure 3.26 est le premier qui s'affiche au gestionnaire lors de l'appui sur la section du menu "Gestion de poubelles". Ce tableau regroupe les poubelles par zones de travail c'est à dire par région.

The screenshot shows the RE:SCHOOL Ecology application interface. The sidebar menu is identical to Figure 3.25. The main content area has a green header bar with tabs: GROUP, ZONE TRAVAIL, ETABLISSEMENT, BLOCS ETABLISSEMENT, ETAGES ETABLISSEMENT, BLOCS POUBELLES, and POUBELLES. Below the header is a search bar and a table titled "Details par zone travail". The table has columns: Id, Region, Quantite total collecte plastique, Quantite total collecte papier, Quantite total collecte composte, Quantite total collecte canette, created\_at, and updated\_at. There is 1 row of data:

Id	Region	Quantite total collecte plastique	Quantite total collecte papier	Quantite total collecte composte	Quantite total collecte canette	created_at	updated_at
> 1	Tunis	10000	7000	5000	8000	17/05/2022	17/05/2022

Below this table is another table titled "Details établissement". It has columns: Id, zone travail, responsable établissement, nom établissement, type établissement, nbr personnes, adresse, longitude, latitude, quantite dechets plastique, and quantite dechets composte. There is 1 row of data:

Id	zone travail	responsable établissement	nom établissement	type établissement	nbr personnes	adresse	longitude	latitude	quantite dechets plastique	quantite dechets composte
> 1	1	1	ESSECT	universite	3500	4, Rue Abou Zakaria El Hafsi - 1089 Monflury - Tunis - 1089 Tunis	10.17475	36.7872	0	0

Below this table is a third table titled "Details Bloc etablissements". It has a single search bar at the top and no visible data.

FIGURE 3.26 – Interface de la table Poubelles groupées par établissements

## CHAPITRE 3. SPRINT 1 : GESTION DE POUBELLES ET CAMIONS ET VISUALISATION DE CARTE GÉOGRAPHIQUE

En appuyant sur une des lignes de la table, une autre table s'affiche en groupant les poubelles non pas seulement par zones mais par établissement aussi puis par bloc d'établissement puis par étage jusqu'à arriver aux blocs existants dans cet établissement.

The screenshot shows a web-based application interface for managing waste bins. The top navigation bar includes tabs for GROUP, ZONE TRAVAIL, ETABLISSEMENT, BLOCS ETABLISSEMENT, ETAGES ETABLISSEMENT, BLOCS POUBELLES, and POUBELLES. The current view is the 'Poubelle' table, which lists 292 items. Each row contains columns for ID, bloc (block), nom (name), capac... (capacity), capac... (capacity), type (type), etat de remplissage (filling status), temp... (temperature), date d'ajout (date added), date modification (date modified), and Actions (edit, delete). A progress bar indicates the filling level for each bin. A modal dialog box is visible at the top right, asking for confirmation to delete a record from 'localhost:3000'. The bottom of the screen shows a footer with copyright information and navigation links.

FIGURE 3.27 – Interface de la table des poubelles

La figure 3.27 représente le tableau de la gestion des données des poubelles. Nous avons créé une barre de progression pour le pourcentage de remplissage des poubelles en temps réel.

The screenshot shows the 'Camions' (Trucks) table interface. The top navigation bar has tabs for CAMION, ZONE DEPOTS, and DEPOT. The current view is the 'Camions' table, showing 8 records. The table structure is identical to the one in Figure 3.27. A modal dialog box is centered on the screen, asking for confirmation to delete a record from 'localhost:3000'. The bottom of the screen shows a footer with copyright information and navigation links.

FIGURE 3.28 – Test de suppression d'un camion

Nous pouvons visualiser de plus dans la figure 3.28, la table de la gestion des camions. Nous allons réaliser un test sur la fonctionnalité de suppression sur un camion aléatoire.

## CHAPITRE 3. SPRINT 1 : GESTION DE POUBELLES ET CAMIONS ET VISUALISATION DE CARTE GÉOGRAPHIQUE

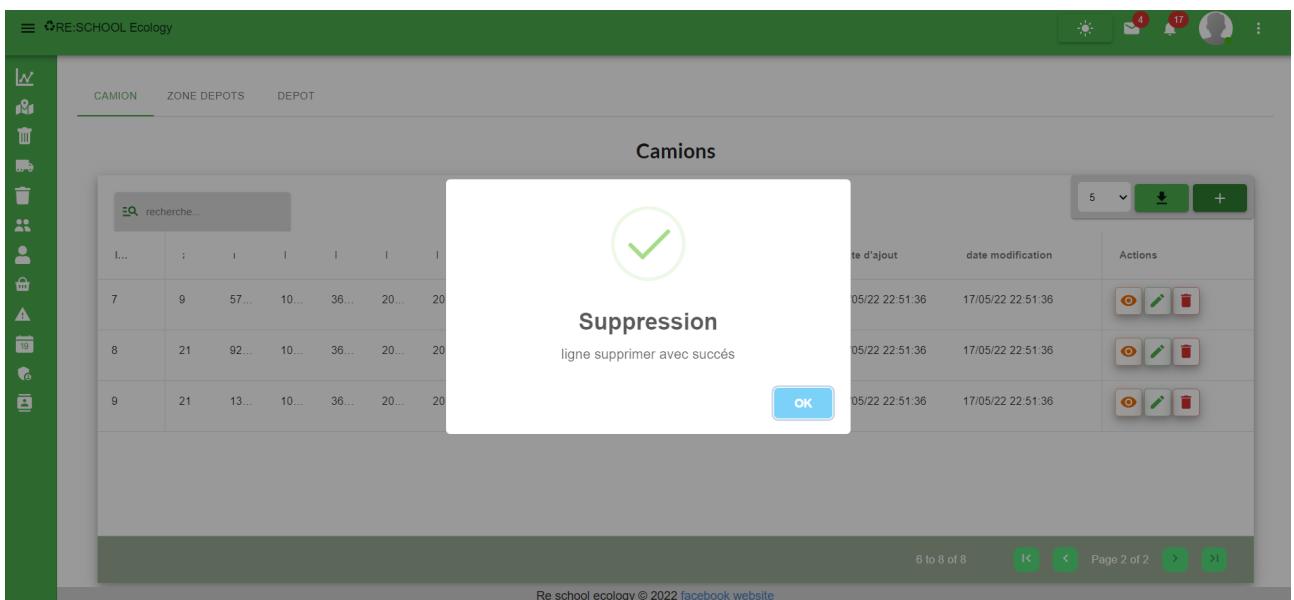


FIGURE 3.29 – Test de suppression d'un camion

Un message de succès de l'opération de suppression s'affiche au gestionnaire comme montré dans la figure 3.29.

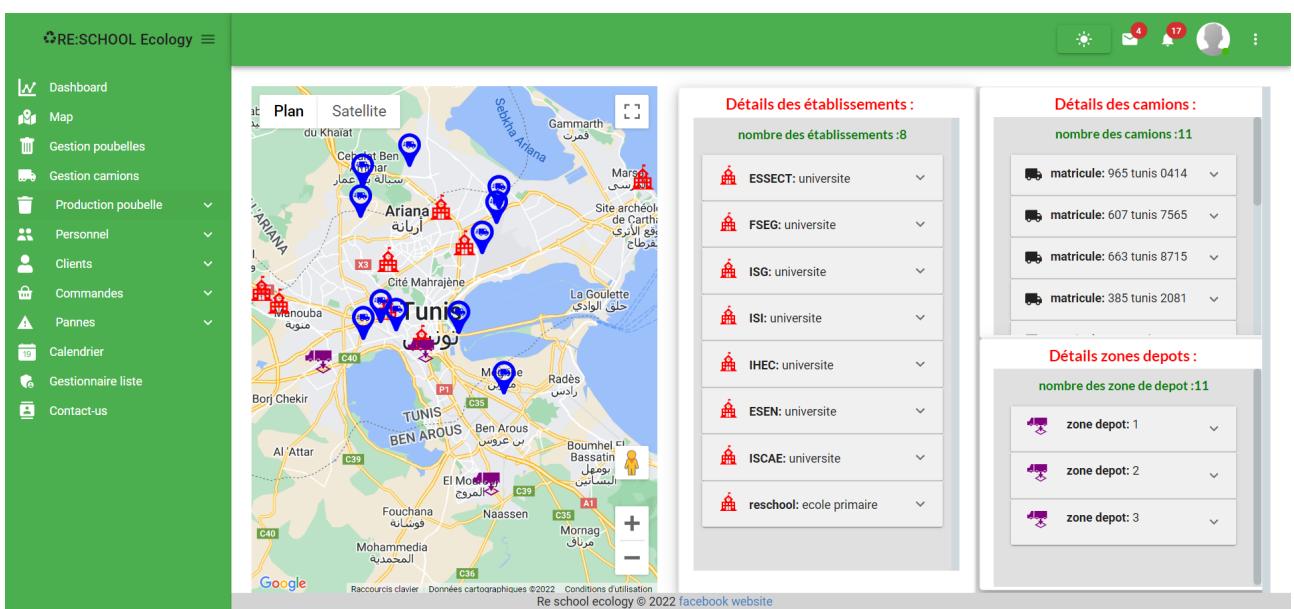


FIGURE 3.30 – Interface de la carte géographique

Après avoir appuyer sur la section "Map" du menu se trouvant à gauche, le système redirige notre gestionnaire vers l'interface de la carte propre ( figure 3.30 ) à lui du fait que le responsable d'établissement possède une carte personnalisé puisqu'il n'a pas accès à toutes les informations de la base de données.

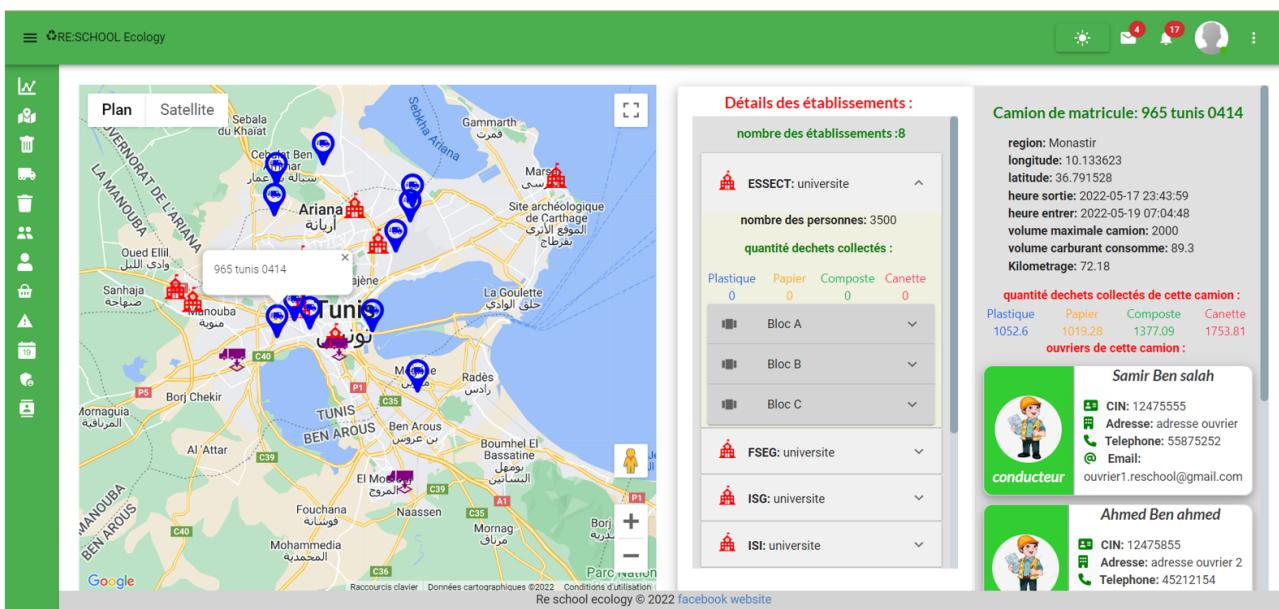


FIGURE 3.31 – Interface détails d'un camion

Lors du clic sur une rubrique d'un camion de la liste "Détails des camions", nous aurons comme affichage les détails du camion cliqué au-dessus et des cartes des ouvriers travaillant sur ce camion. Nous pouvons voir cela dans la figure 3.31.

## 3.5 Sprint Review

### 3.5.1 Livraison du sprint 1

Dans ce sprint, on a pu permettre à les utilisateurs de l'application de manipuler les données des poubelles par la création d'une rubrique dans le menu à cliquer pour accéder à l'interface de leur gestion, et de visualiser la carte géographique pour bien comprendre l'emplacement des poubelles. Comme on a terminé la partie de gestion des camions pour le gestionnaire.

Dans ce livrable, nous pouvons donc ajouter, modifier, supprimer et afficher les données des poubelles et camions, la possibilité de réaliser des recherches en plus de la visualisation de la carte personnalisée selon l'acteur. Toutes ces tâches sont fonctionnelles comme on nous pu voir dans les tests effectués.

### 3.5.2 Obstacles rencontrés et solutions proposées

De nombreux obstacles ont été rencontrés afin d'achever ce sprint surtout de côté technique parmi lesquels :

- La perte de temps dû à les nouvelles technologies qu'on devaient maîtriser pour la réalisation de ce projet.
- La recherche d'une carte géographique à intégrer dans l'application puisque il existe peu de solutions de cartes open-source.

A la fin, nous avons réussi à implémenter une carte de Google Maps pour en travailler au-dessus après avoir discuter avec le Product Owner. De cette manière, nous avons terminé notre premier sprint sans de grosses difficultés.

## Conclusion

Tout au long de ce chapitre, nous avons présenté les différents activités du premier sprint : analyse, conception, codage et test de l'application en détails tout en les définissant ensuite nous avons achever par la livraison du sprint et les . Nous avions donc terminé la première itération de notre projet.

# Chapitre 4

## Sprint 2 : Gestion de Ressources Humaines et Génération de statistiques

### Introduction

Dans le chapitre précédent, nous avons présenté le premier sprint et réussi à avoir une première version de l'application. A présent, nous commencerons à travailler sur le deuxième incrément qui se focalisera principalement sur la génération des statistiques et la gestion des ressources humaines de l'entreprise. Les ressources humaines comprennent :

- les fournisseurs des matières premières pour la fabrication des poubelles.
- les responsables des établissements.
- les mécaniciens se sont les agents de réparation des camions
- les ouvriers qui sont les agents de collecte des déchets des poubelles se trouvant dans les différents établissements.
- les réparateurs des poubelles

Puisque les ressources sont nombreux, nous pourrons pas réaliser à chacun d'eux des descriptions et diagrammes qui seront identiques l'un à l'autre. D'où nous travaillerons sur une seule ressource pour chaque cas d'utilisation . On peut voir ceci en détails dans le backlog du deuxième sprint ce trouvant en dessous :

User Story	Tâches	Points (j)	Priorité
Gestion des Ressources humaines	ajouter fournisseur	2	haute
	modifier responsable établissement	2	moyenne
	supprimer mécanicien	2	moyenne
	rechercher ouvrier	2	moyenne
	afficher détails réparateur poubelle	1	faible
Génération des statistiques	générer les statistiques	10	haute
	générer les statistiques par établissement	7	haute
	visualiser les statistiques par zones	3	moyenne

TABLE 4.1 – Backlog du sprint 2

## 4.1 Analyse

La première étape dans chaque sprint est de commencer par établir l'analyse des cas d'utilisation convenues par l'équipe de développement lors de la création du diagramme de cas d'utilisation global dans le deuxième chapitre. Cette étape est considérée cruciale d'où nous devons être prudents.

Nous pouvons voir le diagramme de cas d'utilisation raffiné de notre deuxième sprint en dessous :

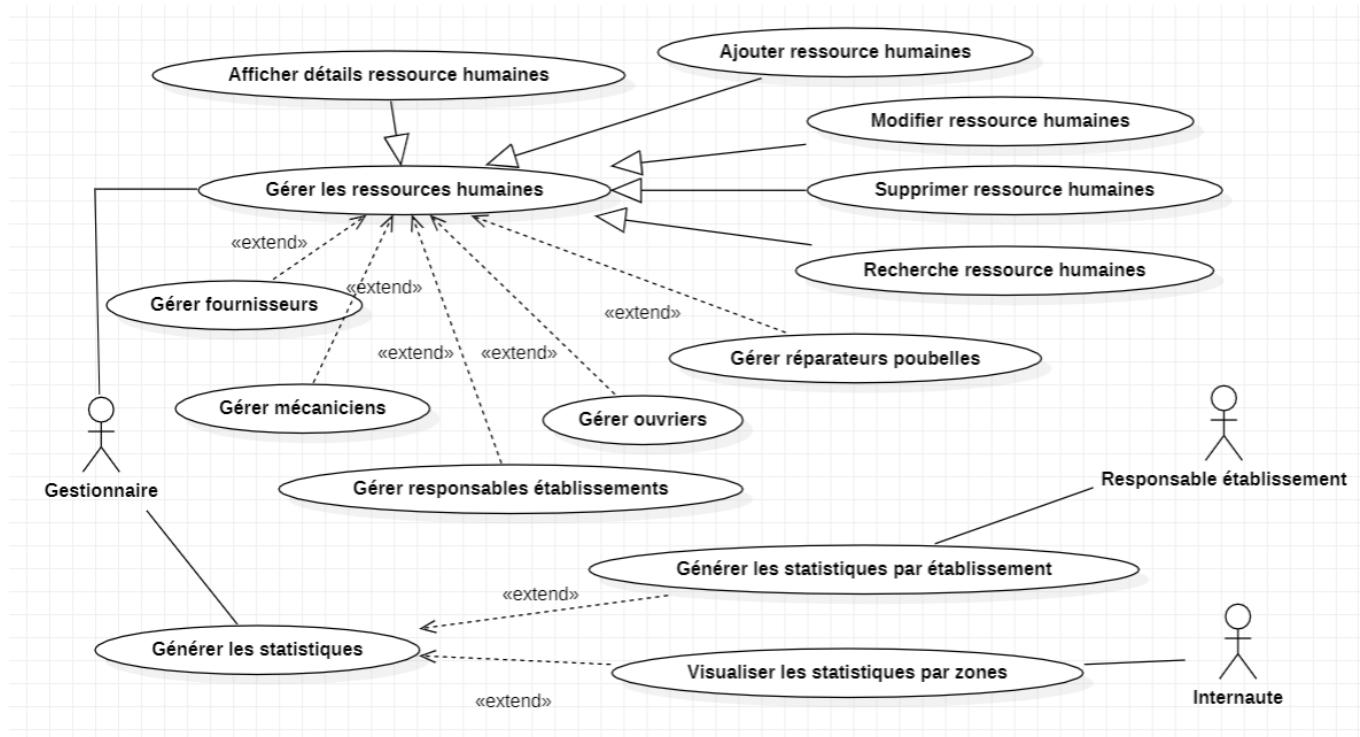


FIGURE 4.1 – Raffinement du diagramme de cas d'utilisation du Sprint 2

Le gestionnaire a l'accès de gérer les différents ressources humaines de la solution ; il peut gérer les données des fournisseurs, mécaniciens, responsables d'établissements, ouvriers et réparateurs des poubelles. Il peut ajouter, modifier, supprimer, afficher les données des divers ressources humaines comme il peut en rechercher.

Il faut noter que chaque ressource possède son propre espace de gestion.

Après avoir établi le diagramme de cas d'utilisation du deuxième Sprint, on commence à décrire ses cas par détails en spécifiant l'acteur concerné, les pré-conditions, les post-conditions et ses scénarios comme on a précédemment fait dans le premier Sprint.

### Description textuelle du sous cas d'utilisation «Ajouter fournisseur»

Cas d'utilisation	Ajouter fournisseur
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a ajouté un fournisseur
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Production poubelle" 3- Le gestionnaire clique sur le sous-menu "Fournisseurs" 4- Le système affiche l'interface "Fournisseurs" propre à lui 5- Le gestionnaire clique sur le bouton "Ajouter" 6- Le système lui affiche une nouvelle interface d'ajout 7- Le gestionnaire entre les données nécessaires du fournisseur 8- Le gestionnaire puis confirme l'ajout 9- Le système affiche un message de la réussite de l'ajout
Scénario alternatif	- Le système informe que l'ajout n'a pas été établi avec succès

TABLE 4.2 – Description textuelle du sous cas d'utilisation «Ajouter fournisseur»

### Description textuelle du sous cas d'utilisation «Modifier Responsable Établissement»

Cas d'utilisation	Modifier Responsable Établissement
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a modifié les données du responsable d'établissement
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Clients" 3- Le gestionnaire clique sur le sous-menu "Responsables Établissements" 4- Le système lui affiche l'interface "Responsables Établissements" 5- Le gestionnaire clique sur le bouton "Modifier" 6- Le système lui affiche une nouvelle interface de modification 7- Le gestionnaire entre les nouvelles données du responsable 8- Le gestionnaire clique sur "Confirmer" 9- Le système affiche un message de la réussite de modification
Scénario alternatif	Le système affiche un message d'erreur de modification

TABLE 4.3 – Description textuelle du sous cas d'utilisation «Modifier Responsable Établissement»

**Description textuelle du sous cas d'utilisation «Supprimer Mécanicien»**

Cas d'utilisation	Supprimer Mécanicien
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a supprimé le mécanicien
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Personnel" 3- Le gestionnaire clique sur le sous-menu "Mécaniciens" 4- Le système affiche l'interface "Mécaniciens" propre à lui 5- Le gestionnaire clique sur le bouton "Supprimer" 6- Le système lui affiche une nouvelle interface de confirmation de suppression 7- Le système affiche un message du succès de suppression
Scénario alternatif	Le système affiche un message d'erreur de suppression

TABLE 4.4 – Description textuelle du sous cas d'utilisation «Supprimer Mécanicien»

**Description textuelle du sous cas d'utilisation «Rechercher Ouvrier»**

Cas d'utilisation	Rechercher Ouvrier
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a trouvé l'ouvrier qu'il cherche
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Personnel" 2- Puis, il clique sur le sous-menu "Ouvriers" 3- Le système lui affiche l'interface "Ouvriers" 4- Le gestionnaire clique sur la zone de recherche 4- Le gestionnaire saisie le nom de l'ouvrier 5- Le système lui affiche l'ouvrier recherchée dans la table
Scénario alternatif	Le système affiche un message "Ouvrier introuvable"

TABLE 4.5 – Description textuelle du sous cas d'utilisation «Rechercher Ouvrier»

**Description textuelle du sous cas d'utilisation «Afficher détails Réparateur poubelles»**

Cas d'utilisation	Afficher détails Réparateur poubelles
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a affiché les détails du réparateur poubelles désiré
Scénario principal	1- Le gestionnaire entre sur le dashboard 2- Le gestionnaire clique sur le menu "Personnel" 3- Le gestionnaire clique sur le sous-menu "Réparateurs poubelles" 4- Le système affiche l'interface "Réparateurs poubelles" propre à lui 5- Le gestionnaire clique sur le bouton "Afficher" d'un réparateur désiré 6- Le système lui affiche les détails du réparateur sélectionné 7- Le gestionnaire clique sur le bouton "Annuler" pour quitter la fenêtre
Scénario alternatif	Le système n'affiche pas le bouton "Afficher"

TABLE 4.6 – Description textuelle du sous cas d'utilisation «Afficher détails Réparateur poubelles»

**Description textuelle du sous cas d'utilisation «Générer les statistiques»**

Cas d'utilisation	Générer les statistiques
Acteur	Gestionnaire
Pré-condition	Le gestionnaire s'est connecté
Post-condition	Le gestionnaire a consulté les statistiques
Scénario principal	1-Le gestionnaire entre sur le dashboard 2-Le système affiche l'interface propre à lui 3-Toutes les statistiques sont affichés sur l'écran
Scénario alternatif	-Aucune valeur ne s'affiche dans les statistiques

TABLE 4.7 – Description textuelle du sous cas d'utilisation «Générer les statistiques»

**Description textuelle du sous cas d'utilisation «Générer les statistiques par établissement»**

Cas d'utilisation	Générer les statistiques par établissement
Acteur	Responsable de l'établissement
Pré-condition	Le responsable s'est connecté
Post-condition	Le responsable a consulté les statistiques
Scénario principal	1-Le responsable entre sur le dashboard 2-Le système affiche l'interface propre à lui 3-Les statistiques globales et propres à lui sont affichés sur l'écran
Scénario alternatif	-Aucune valeur ne s'affiche dans les statistiques

TABLE 4.8 – Description textuelle du sous cas d'utilisation «Générer les statistiques par établissement»

**Description textuelle du sous cas d'utilisation «Visualiser les statistiques par zones»**

Cas d'utilisation	Visualiser les statistiques par zones
Acteur	Internaute
Pré-condition	L'internaute s'est connecté
Post-condition	L'internaute a consulté les statistiques camions
Scénario principal	1-L'internaute entre sur la page d'accueil 2-Le système lui affiche l'interface internaute 3-Des statistiques globales par zones sont affichés sur la page
Scénario alternatif	Le système affiche des espaces vides au lieu des statistiques

TABLE 4.9 – Description textuelle du sous cas d'utilisation «Visualiser les statistiques par zones»

#### 4.1.1 Diagrammes de séquence système du deuxième sprint

Diagramme de séquence système du sous cas d'utilisation «Visualiser les statistiques par zones»

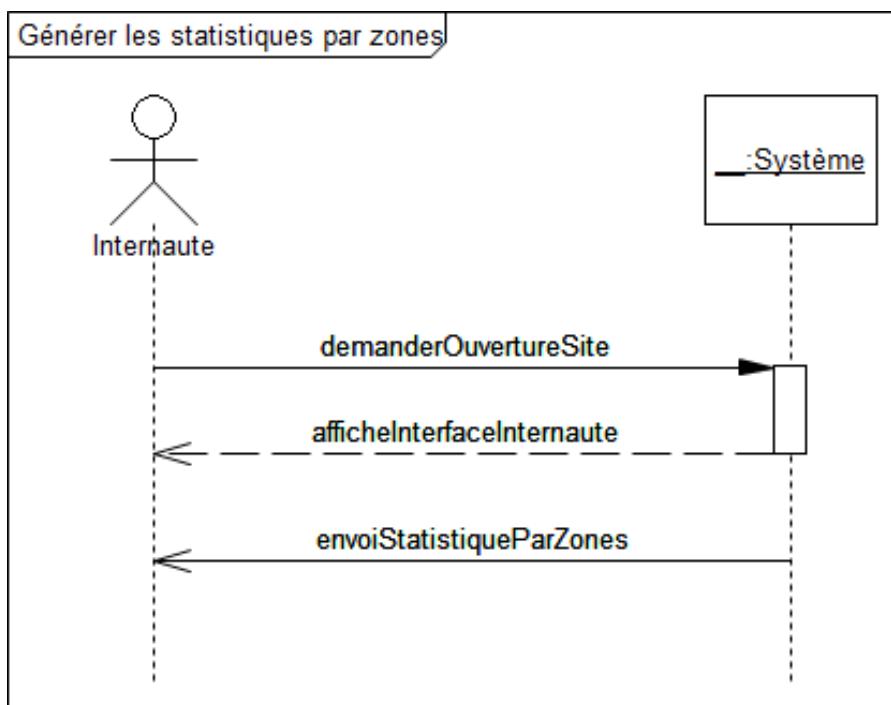


FIGURE 4.2 – Diagramme de séquence système du sous cas d'utilisation "Visualiser les statistiques par zones"

## 4.2 Conception

### 4.2.1 Diagramme de séquence de conception

La figure 4.3 nous explique comment l'ajout d'une nouvelle ressource humaine se réalise dans notre application.

Désidément, le gestionnaire doit cliquer sur le bouton des ressources humaines qu'il veut y ajouter une nouvelle ressource, pour accéder à les tableaux de data et clique sur le bouton "+" à coté de lui. Le système lui affiche une nouvelle fenêtre d'ajout. Après avoir remplis les données nécessaires, il doit confirmer l'ajout et un message de succès d'ajout d'une nouvelle ressource.

Diagramme de séquence système de conception du sous cas d'utilisation «Ajouter Fournisseur»

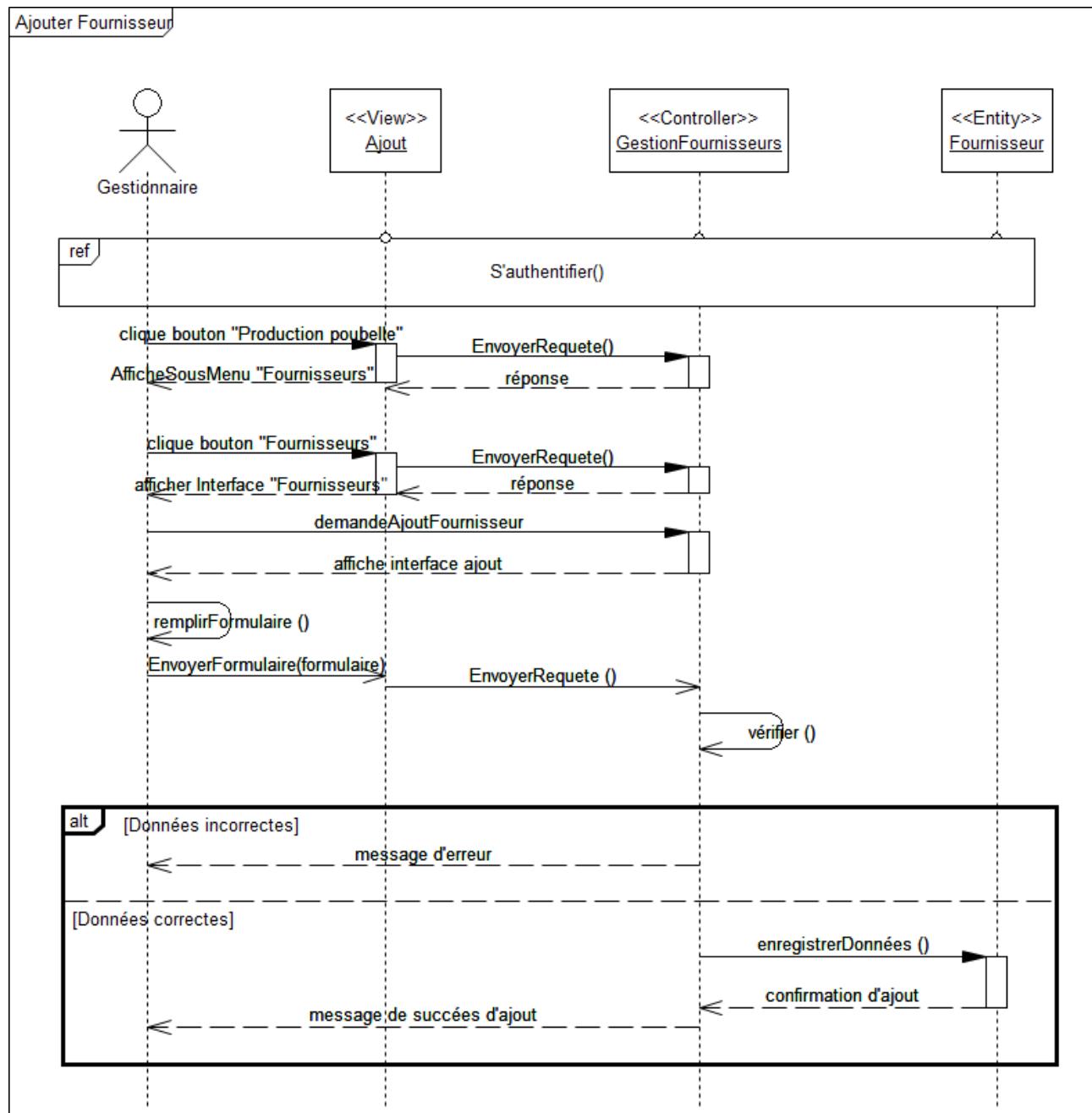


FIGURE 4.3 – Diagramme de séquence système de conception du sous cas d'utilisation "Ajouter Fournisseur"

**Diagramme de séquence système de conception du sous cas d'utilisation «Modifier Responsable Établissement»**

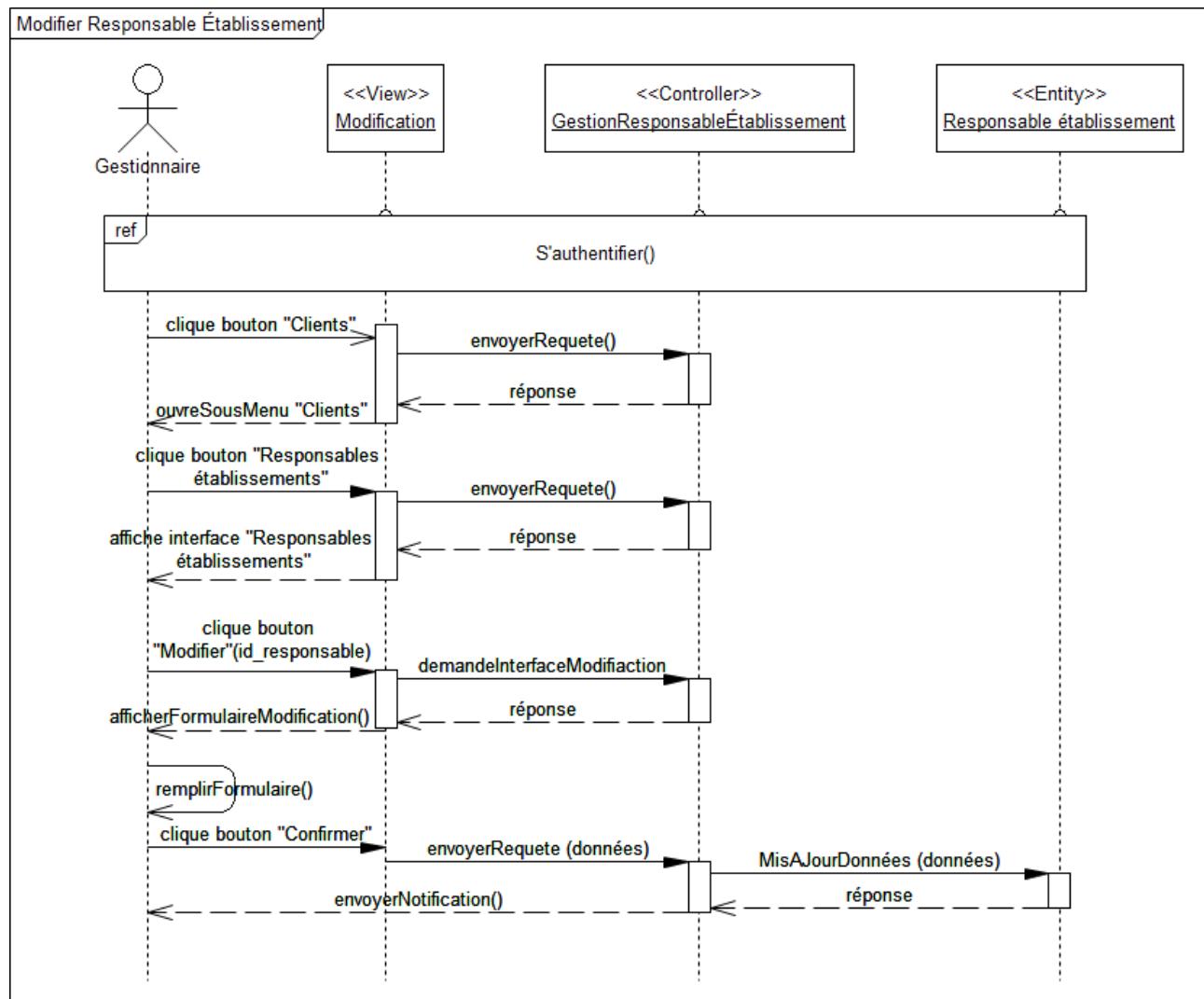


FIGURE 4.4 – Diagramme de séquence système de conception du sous cas d'utilisation "Modifier Responsable Établissement"

**Diagramme de séquence système de conception du sous cas d'utilisation «Supprimer Mécanicien»**

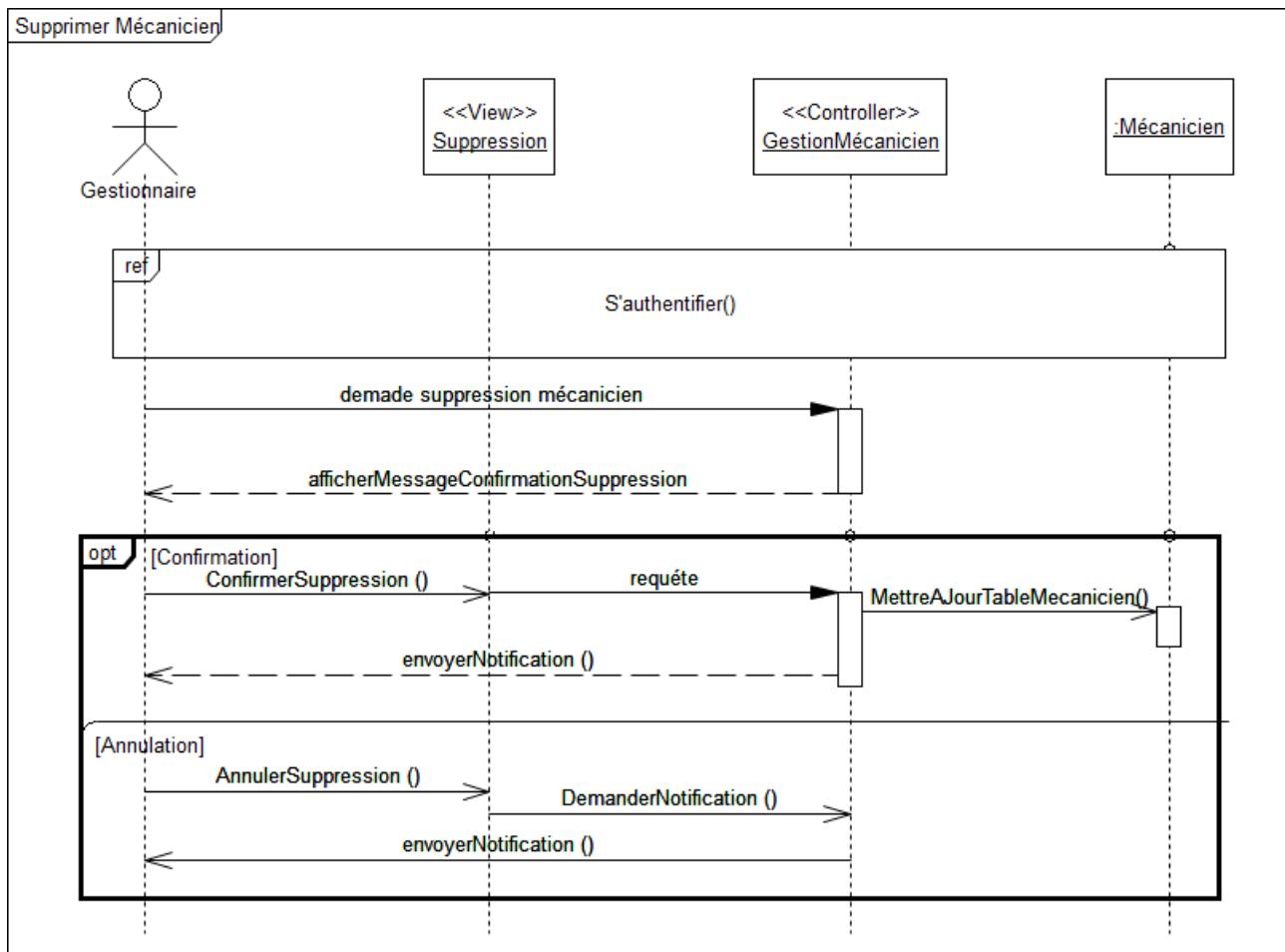


FIGURE 4.5 – Diagramme de séquence système de conception du sous cas d'utilisation "Supprimer Mécanicien"

**Diagramme de séquence système de conception du sous cas d'utilisation «Rechercher Ouvrier»**

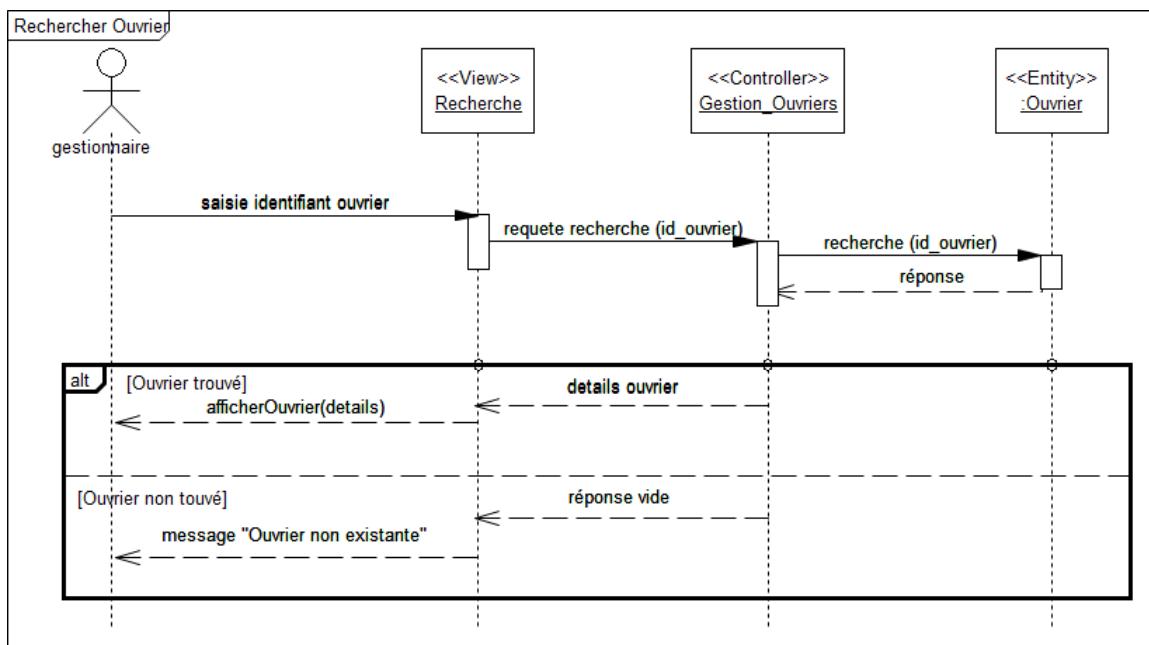


FIGURE 4.6 – Diagramme de séquence système de conception du sous cas d'utilisation "Rechercher Ouvrier"

**Diagramme de séquence système de conception du sous cas d'utilisation «Afficher détails Réparateur Poubelle»**

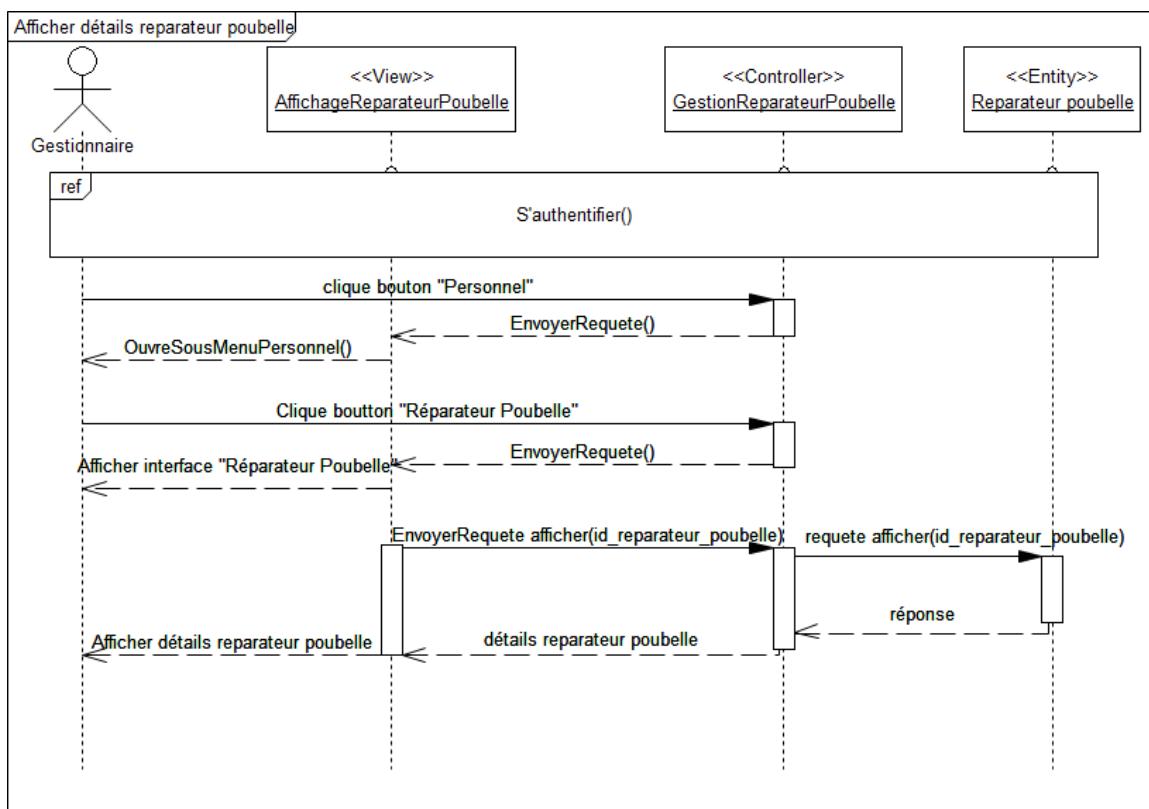


FIGURE 4.7 – Diagramme de séquence système de conception du sous cas d'utilisation "Afficher détails Réparateur Poubelle"

**Diagramme de séquence système de conception du sous cas d'utilisation «Générer les statistiques»**

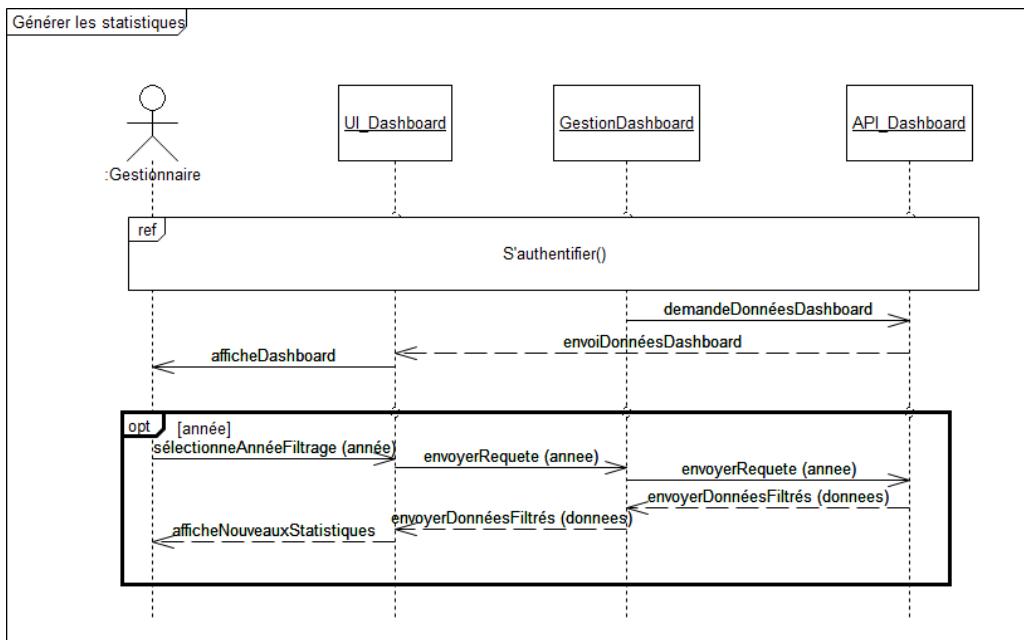


FIGURE 4.8 – Diagramme de séquence système de conception du sous cas d'utilisation "Générer les statistiques"

**Diagramme de séquence système de conception du sous cas d'utilisation «Générer les statistiques par établissement»**

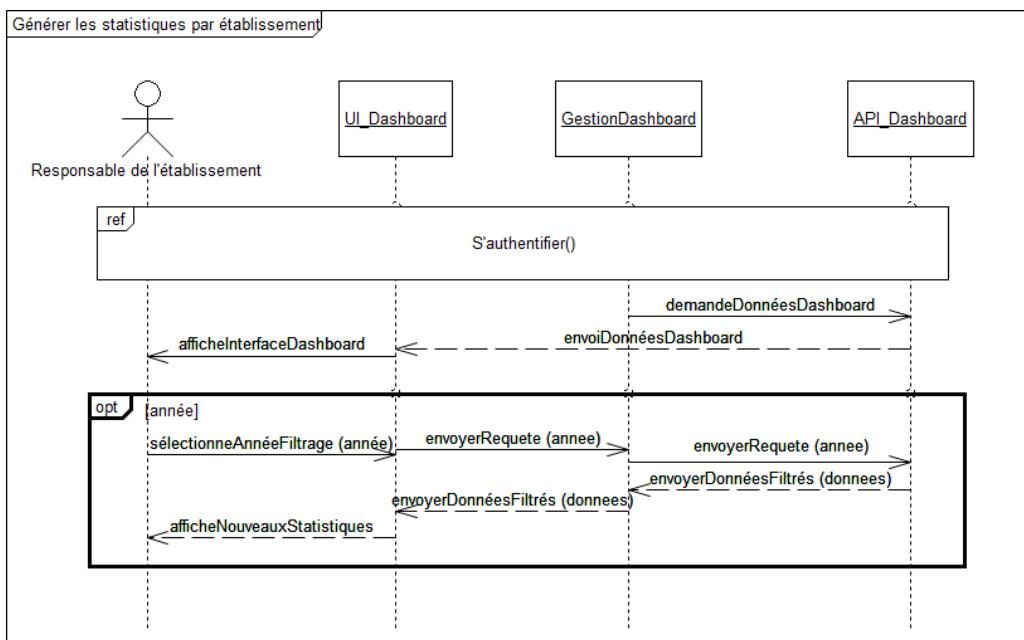


FIGURE 4.9 – Diagramme de séquence système de conception du sous cas d'utilisation "Générer les statistiques par établissement"

## 4.2.2 Diagramme de classe global du deuxième sprint

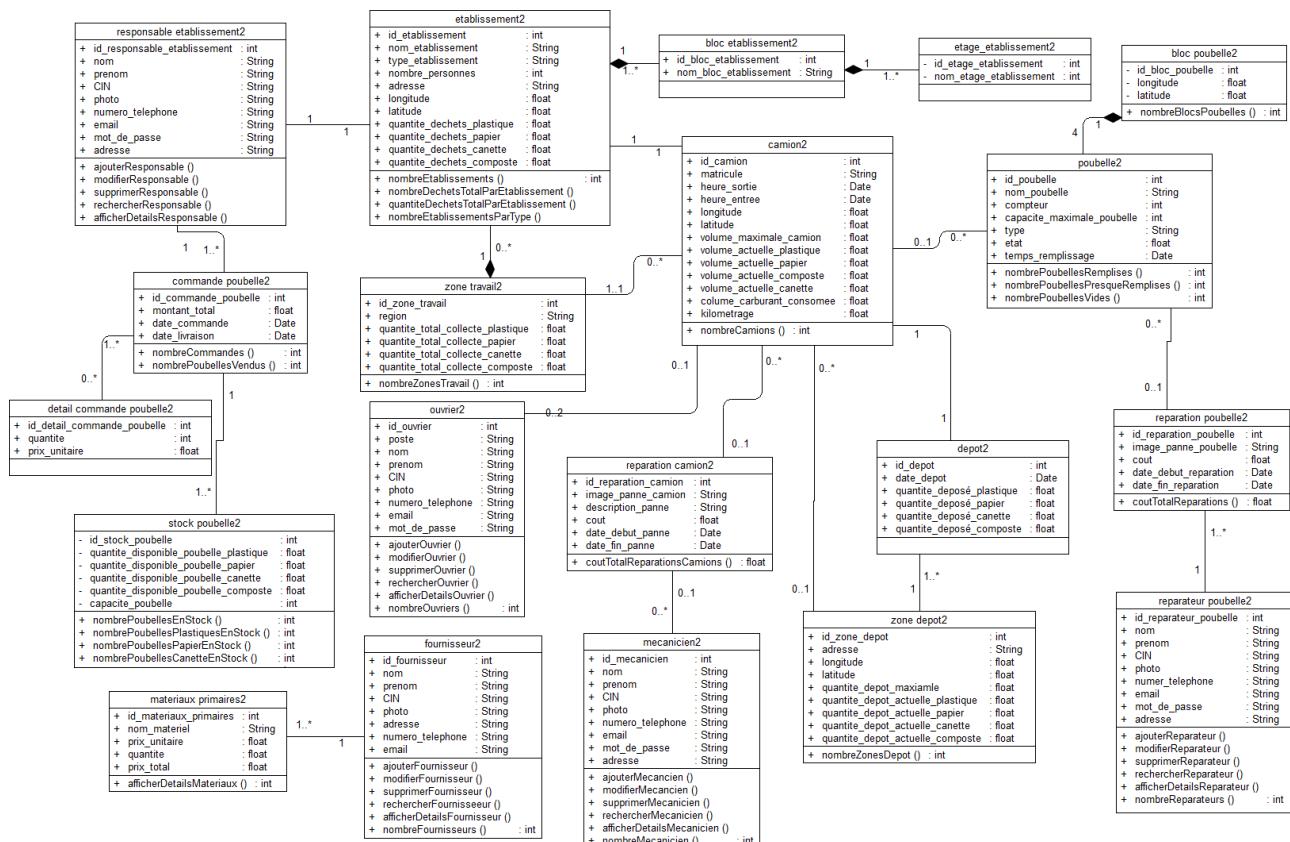


FIGURE 4.10 – Diagramme de classe du Sprint 2

## 4.3 Codage

### 4.3.1 Schéma de la base de données

Le diagramme de classes du premier sprint est très développé donc nous choisissons quelques tables de la base réalisé : Maintenant, nous passerons à la partie de codage du système. Nous allons ajouter toutes les nouvelles classes ajoutées au diagramme de classes de notre deuxième sprint.

Nous allons choisir quelques tables des tables ajoutées récemment à la base existante :

Champs	Types	Contraintes
id	bigint(20)	Not NULL, Primary Key
CIN	varchar(8)	Not NULL
nom	varchar(20)	Not NULL
prenom	varchar(20)	Not NULL
photo	varchar(255)	-
adresse	varchar(255)	Not NULL
numero_telephone	varchar(12)	Not NULL
email	varchar(40)	Not NULL

TABLE 4.10 – table fournisseurs

**CHAPITRE 4. SPRINT 2 : GESTION DE RESSOURCES HUMAINES ET GÉNÉRATION  
DE STATISTIQUES**

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
zone_travail_id	bigint(20)	Not NULL, Foreign Key
camion_id	bigint(20)	Not NULL, Foreign Key
poste	enum('conducteur', 'agent')	Not NULL
nom	varchar(20)	Not NULL
prenom	varchar(20)	Not NULL
cin	varchar(8)	Not NULL
photo	varchar(255)	-
numero_telephone	varchar(12)	Not NULL
email	varchar(40)	Not NULL
adresse	varchar(255)	Not NULL

TABLE 4.11 – table ouvriers

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
quantite_disponible_plastique	int(11)	Not NULL
quantite_disponible_canette	int(11)	Not NULL
quantite_disponible_composte	int(11)	Not NULL
quantite_disponible_papier	int(11)	Not NULL
capacite_poubelle	int(11)	Not NULL

TABLE 4.12 – table stock\_poubelles

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
fournisseur_id	bigint(20)	Not NULL
nom_materiel	varchar(255)	Not NULL
prix_unitaire	double(8,2)	Not NULL
quantite	int(11)	Not NULL
prix_total	double(8,2)	Not NULL

TABLE 4.13 – table materiaux\_primaire

<b>Champs</b>	<b>Types</b>	<b>Contraintes</b>
id	bigint(20)	Not NULL, Primary Key
CIN	varchar(8)	Not NULL
nom	varchar(20)	Not NULL
prenom	varchar(20)	Not NULL
photo	varchar(255)	-
adresse	varchar(255)	Not NULL
numero_telephone	varchar(12)	Not NULL
email	varchar(40)	Not NULL

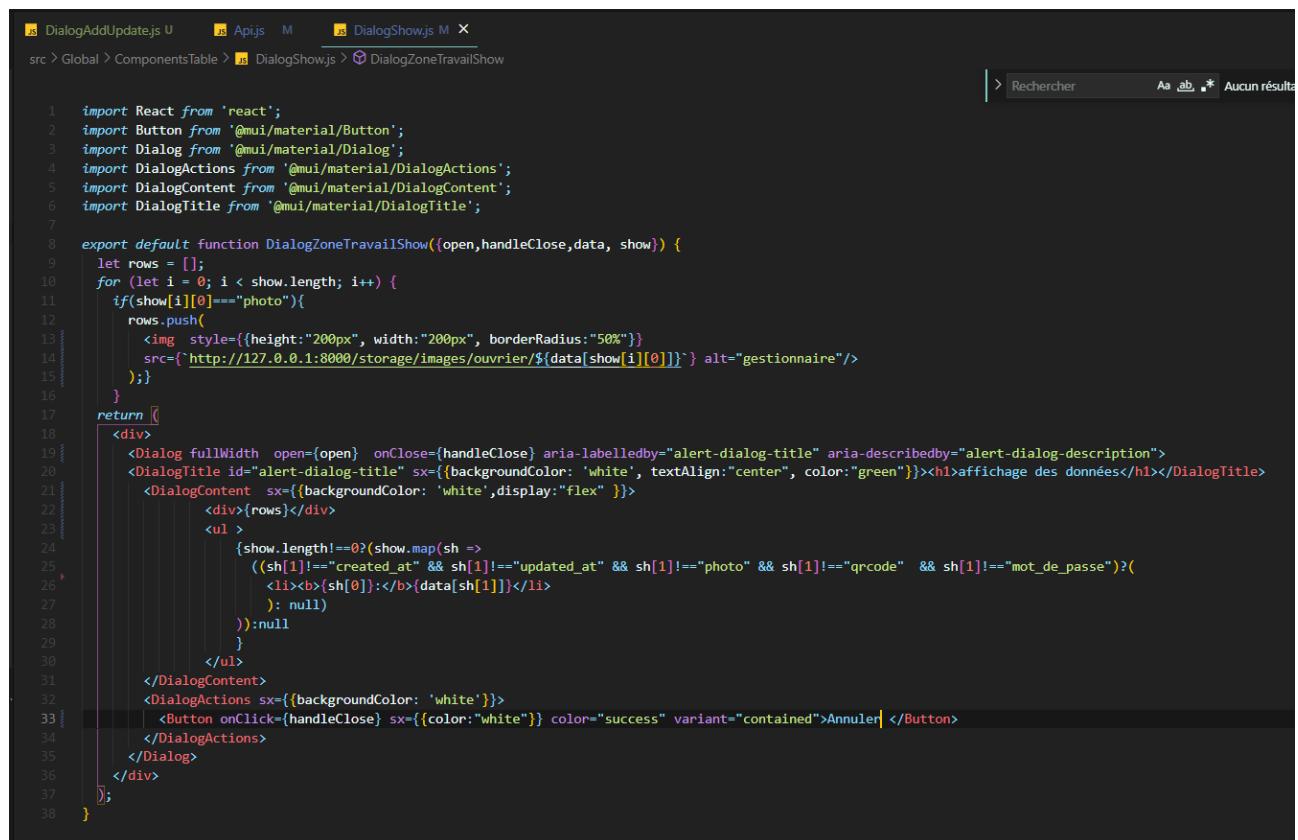
TABLE 4.14 – table reparateur\_poubelles

Champs	Types	Contraintes
id	bigint(20)	Not NULL, Primary Key
poubelle_id	bigint(20)	Not NULL, Foreign Key
reparateur_poubelle_id	bigint(20)	Not NULL, Foreign Key
image_panne_poubelle	varchar(255)	Not NULL
description_panne	text	Not NULL
cout	double(8,2)	Not NULL
date_debut_reparation	datetime	Not NULL
date_fin_reparation	datetime	-

TABLE 4.15 – table reparation\_poubelles

### 4.3.2 Extraits du code

Nous sommes arrivés à la partie de codage de notre application de gestion des déchets. Pour ce sprint, on va commencer par la partie d'affichage des détails de toute instance d'une table existante dans nos interfaces créées tout au long du premier et deuxième sprints.



```

1 import React from 'react';
2 import Button from '@mui/material/Button';
3 import Dialog from '@mui/material/Dialog';
4 import DialogActions from '@mui/material/DialogActions';
5 import DialogContent from '@mui/material/DialogContent';
6 importDialogTitle from '@mui/material/DialogTitle';
7
8 export default function DialogZoneTravailShow({open, handleClose, data, show}) {
9   let rows = [];
10  for (let i = 0; i < show.length; i++) {
11    if(show[i][0]==="photo"){
12      rows.push(
13        <img style={{height:"200px", width:"200px", borderRadius:"50%"}}
14          src={'http://127.0.0.1:8000/storage/images/ouvrier/${data[show[i][0]]}' alt="gestionnaire"/>
15      );
16    }
17  }
18  return (
19    <div>
20      <Dialog fullWidth open={open} onClose={handleClose} aria-labelledby="alert-dialog-title" aria-describedby="alert-dialog-description">
21        <DialogTitle id="alert-dialog-title" sx={{backgroundColor: 'white', textAlign:"center", color:"green"}}><h1>affichage des données</h1></DialogTitle>
22        <DialogContent sx={{backgroundColor: 'white',display:"flex" }}>
23          <div>{rows}</div>
24          <ul>
25            {show.length!==0?(show.map(sh =>
26              ((sh[1]!=="created_at" && sh[1]!=="updated_at" && sh[1]!=="photo" && sh[1]!=="qrCode" && sh[1]!=="mot_de_passe")?(
27                <li><b>{sh[0]}:</b></li><li>{data[sh[1]]}</li>
28              ): null)
29            )):null}
30          </ul>
31        </DialogContent>
32        <DialogActions sx={{backgroundColor: 'white'}}>
33          <Button onClick={handleClose} sx={{color:"white"}} color="success" variant="contained">Annuler</Button>
34        </DialogActions>
35      </Dialog>
36    </div>
37  );
38}

```

FIGURE 4.11 – Extrait du code DialogShow.js

Cet extrait de code de la composante "DialogShow.js" illustre le code effectué pour faire fonctionner la possibilité d'afficher les détails d'une ligne des tables réalisées.

## CHAPITRE 4. SPRINT 2 : GESTION DE RESSOURCES HUMAINES ET GÉNÉRATION DE STATISTIQUES

```

1 > import React from 'react';
2
3 export default function DialogAddUpdate({open, handleClose, data, onChange, handleFormSubmit, validation, show}) {
4
5 const [id]=data;
6 let rows = [];
7 for (let i = 0; i < show.length; i++) {
8   if(show[i][0]==="photo"){
9     rows.push(
10       <>
11         <input type="file" name={show[i][0]} id={show[i][0]} onChange={e=>onChange(e)} value={data[show[i][0]]} />
12         <FormHelperText error={true}> {validation[show[i][0]]} </FormHelperText>
13       );
14   }
15 }
16
17 return (
18   <div>
19     <Dialog open={open} onClose={handleClose} aria-labelledby="alert-dialog-title" aria-describedby="alert-dialog-description" fullWidth>
20       <DialogTitle id="alert-dialog-title" sx={{backgroundColor: 'white', color:'green' , textAlign:'center', fontWeight:'bold'}}>{id?"modifier des données ":"créer un nouveau "}</DialogTitle>
21       <DialogContent sx={{backgroundColor: 'white'}}>
22         <form>
23           {rows}
24           {show.length!=0?show.map(sh =>
25             { (sh[1]!=="id" && sh[1]!=="created_at" && sh[1]!=="updated_at" && sh[1]!=="photo" )?(
26               <TextField id={sh[1]} value={data[sh[1]]} onChange={e=>onChange(e)} placeholder={sh[1]} error={!validation[sh[1]]} label={sh[1]} variant="outlined" margin="dense" fullWidth />
27               <FormHelperText error={true}> {validation[sh[1]]} </FormHelperText>
28             ): null
29           )}:null}
30         </form>
31       </DialogContent>
32       <DialogActions sx={{backgroundColor: 'white'}}>
33         <Button sx={{color:"green"}} color="primary" onClick={handleClose} variant="outlined">
34           Annuler
35         </Button>
36         <Button sx={{color:"white"}} color="success" onClick={()>handleFormSubmit()} variant="contained">
37           {id?"modifier":"envoyer"}
38         </Button>
39       </DialogActions>
40     </Dialog>
41   </div>
42 );
43
44
45
46
47
48
49
50
51
}

```

FIGURE 4.12 – Extrait du code DialogAddUpdate.js

Puis on trouve dans la figure 4.12, un petit extrait du code l'ajout et de modification des multiples instances de la base de données.

## 4.4 Test

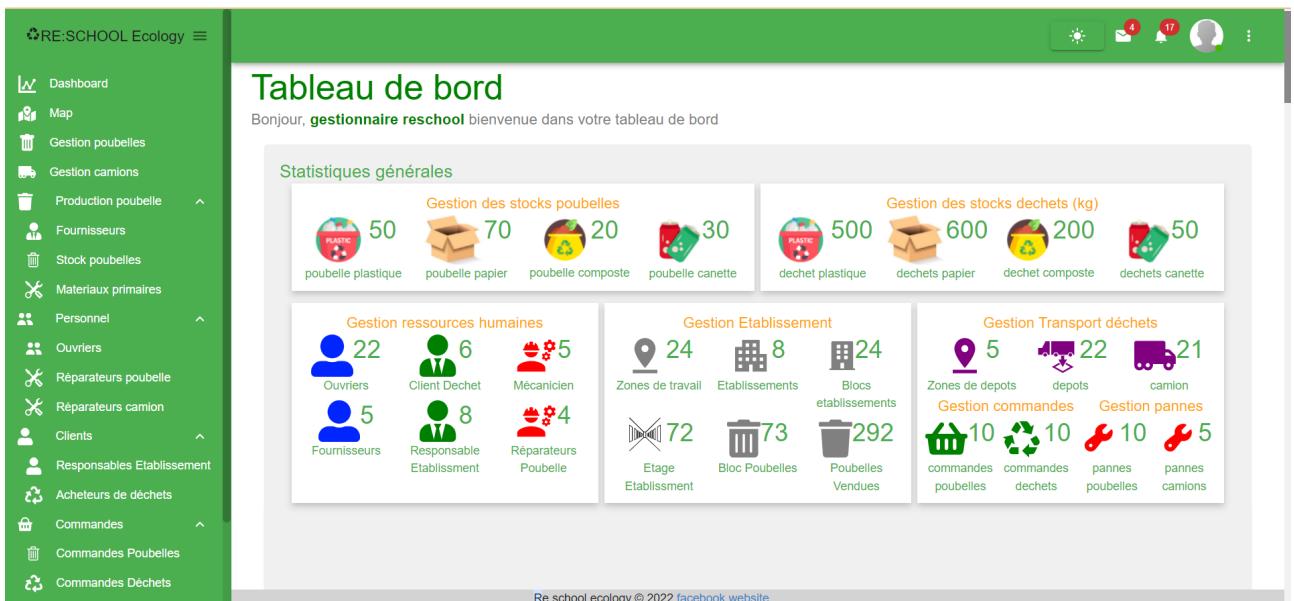


FIGURE 4.13 – Interface du Dashboard du gestionnaire

On trouve dans la figure 4.13 ci-contre la 1ère page qui s'affiche au gestionnaire après avoir entrer ses coordonnées de connexion correctement.

## CHAPITRE 4. SPRINT 2 : GESTION DE RESSOURCES HUMAINES ET GÉNÉRATION DE STATISTIQUES

Des statistiques portant sur les nombres totaux de chaque instance groupés par type de gestion s'affiche à notre gestionnaire. Ses données sont reliées à une base de données hébergée.

En descendant dans la même page, nous trouverons d'autres statistiques sous forme de plusieurs types de graphiques.

Nous pouvons visualiser les statistiques reliées à les pannes des poubelles et camion comme la durée maximale, minimale et moyenne des réparation effectués et plusieurs autres statistiques.

Les figures 4.14 et 4.15 nous permettent de mieux comprendre ceci.

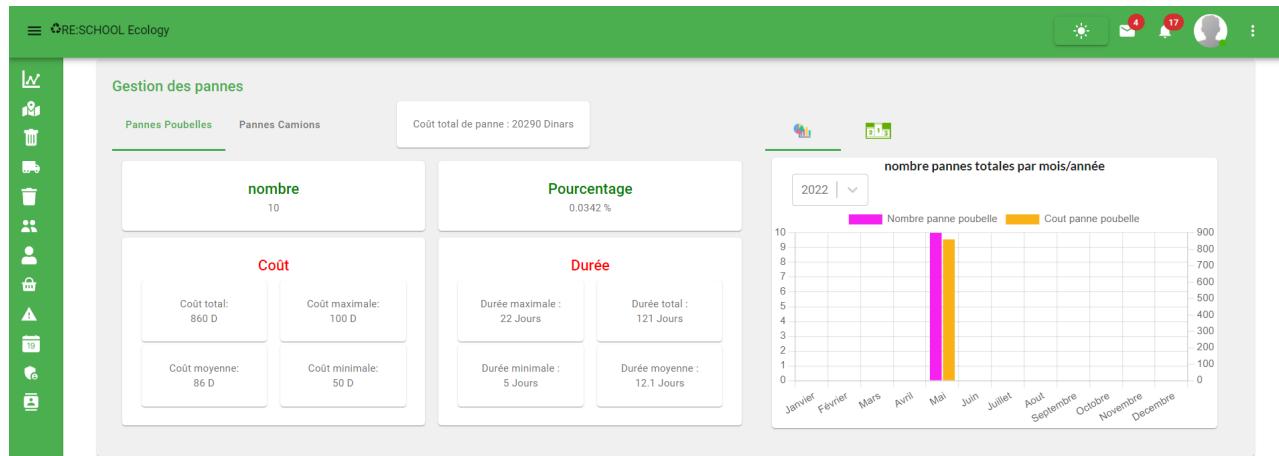


FIGURE 4.14 – Interface de la partie gestion de pannes poubelles

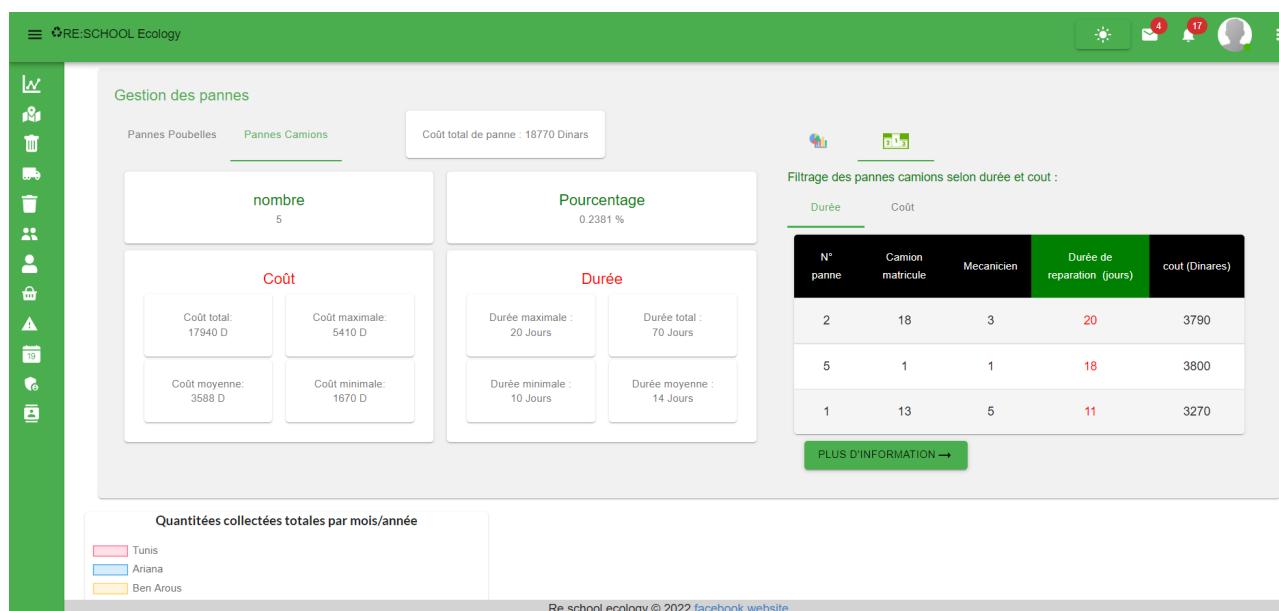


FIGURE 4.15 – Interface de la partie gestion de pannes camions

Comme on a pu créer des graphes à barres pour afficher la somme des quantités des déchets collectés par mois et par année.

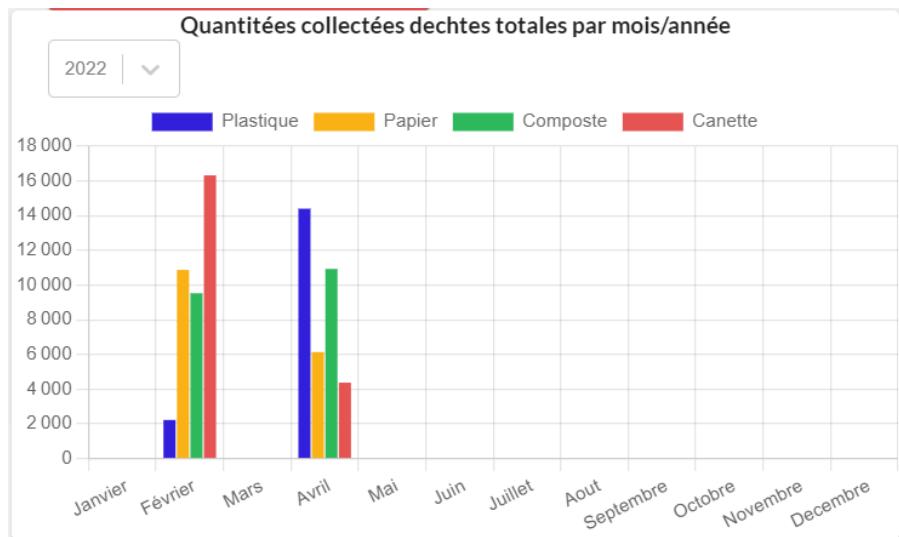


FIGURE 4.16 – Test de graphe à barres

La figure 4.17 montre la carte de la Tunisie qu'on a créé pour afficher au gestionnaire les quantités des déchets collectés sur tout le pays, par région et par établissement.

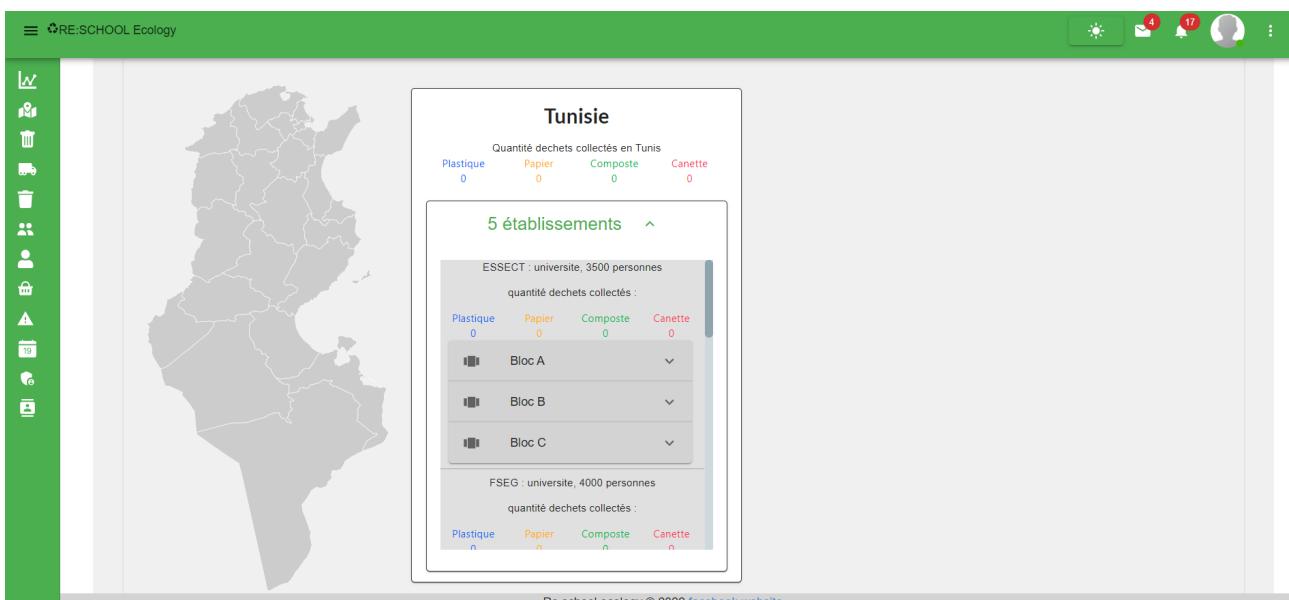


FIGURE 4.17 – Test sur la carte de la Tunisie

Maintenant, nous procéderons à l'illustration du test effectué pour la fonctionnalité d'ajout d'une nouvelle ressource humaine. On choisiras l'ouvrier comme exemple pour ce test.

## CHAPITRE 4. SPRINT 2 : GESTION DE RESSOURCES HUMAINES ET GÉNÉRATION DE STATISTIQUES

I...	camion...	photo	poste	nom	prenom	CIN	numero_te...	email	date c...	Actions
1	1		conducteur	Samir	Ben salah	12475555	55875252	ouvrier1.reschool@gm...	17/05/2022	
2	1		agent	Ahmed	Ben ahmed	12475855	45212154	ouvrier2.reschool@gm...	17/05/2022	
3	2		conducteur	Roslyn	Smitham	09246963	206482328	webster17@example.net	17/05/2022	
4	2		agent	Guy	Hagenes	50846992	49169900	fisher.destini@example...	17/05/2022	
5	9		conducteur	Geovanni	Schmitt	89611475	28405158	ona.farrell@example.net	17/05/2022	

Re school ecology © 2022 facebook website

FIGURE 4.18 – Interface de la table Ouvriers

créer un nouveau

Choisir un fichier Aucun fichier choisi

camion\_id

poste

nom

prenom

CIN

numero\_telephone

email

adresse

1 : il peut confirmer l'ajout du nouveau camion

2 : ou il peut annuler sa création

ANNULER ENVOYER

Re school ecology © 2022 facebook website

FIGURE 4.19 – Test

Finalement, nous allons présenter l'interface de l'affichage des données d'un ouvrier sélectionnée au préalable.

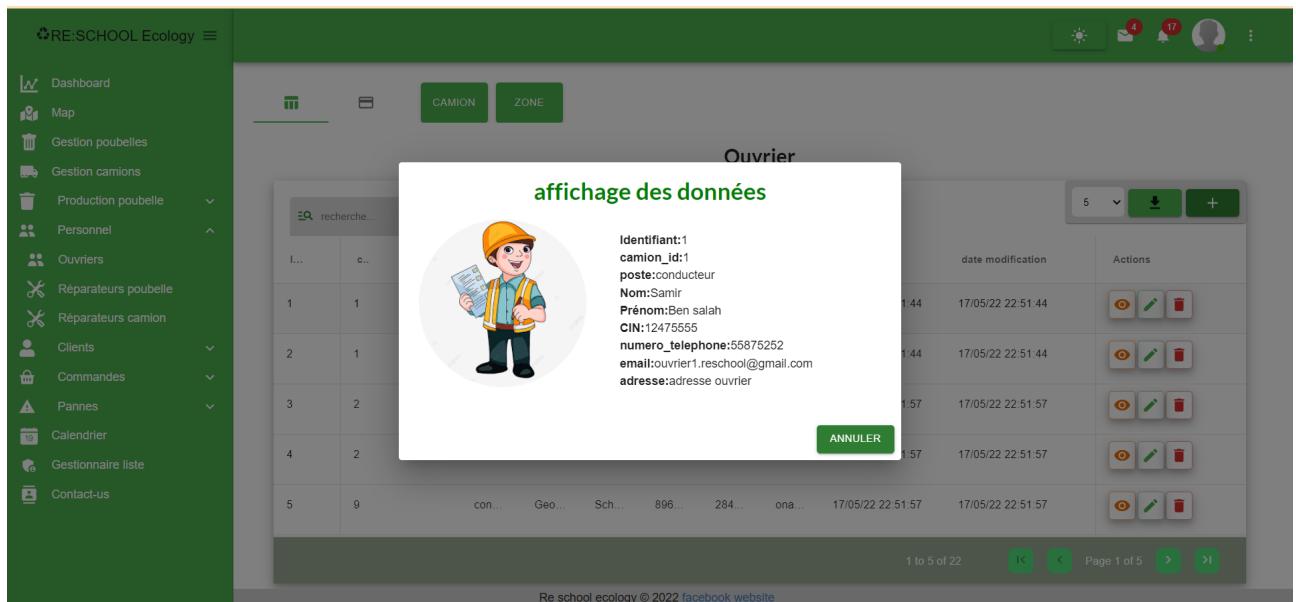


FIGURE 4.20 – Test

## 4.5 Sprint Review

### 4.5.1 Livraison du sprint 2

À la suite de ce que nous avions pu livrer du premier sprint, nous avons ajouté les parties présentées dans la section des tests réalisés au cours de notre sprint actuel. Nous avons terminé le développement du dashboard pour les deux acteurs de plus nous avons mis en place les tables des multiples ressources humaines (ouvriers, fournisseurs, mécanciens,etc).

### 4.5.2 Obstacles rencontrés et solutions proposées

Pour ce sprint, nous avons rencontrés qu'un seul obstacle celui étant le choix des statistiques à afficher étant donnée que la société n'a pas exigé de certains statistiques et nous a laissé la liberté d'en choisir.

## Conclusion

A présent on pourrait dire qu'on a terminé le deuxième sprint sans rencontrer de gros problèmes. Nous avons présenté la gestion des ressources humaines et les différents actions que le gestionnaire peut effectuer en plus de la partie dashboard et statistiques. Il nous reste maintenant seulement le troisième sprint portant sur l'authentification et la gestion de comptes.

# Sprint 3 : Authentification et Gestion de comptes

## Introduction

En partant sur le même principe que les chapitres précédents, ce chapitre portera sur l'authentification des utilisateurs (gestionnaire et responsable d'établissement) et la gestion de leurs comptes.

Ceci est bien représenté dans le backlog de notre troisième Sprint :

User Story	Tâches	Points (j)	Priorité
Authentification	S'authentifier	11	haute
Gestion de comptes	Gérer compte gestionnaire Gérer compte responsable de l'établissement	6 5	moyen moyen

TABLE 5.1 – Backlog du Sprint 3

## 5.1 Analyse

Pour la phase d'analyse de ce Sprint, nous établirons les descriptions textuelles ainsi que les diagrammes de séquence système des cas d'utilisation identifiées dans le backlog définie au-dessous. Voici le diagramme de cas d'utilisation raffiné du troisième sprint :

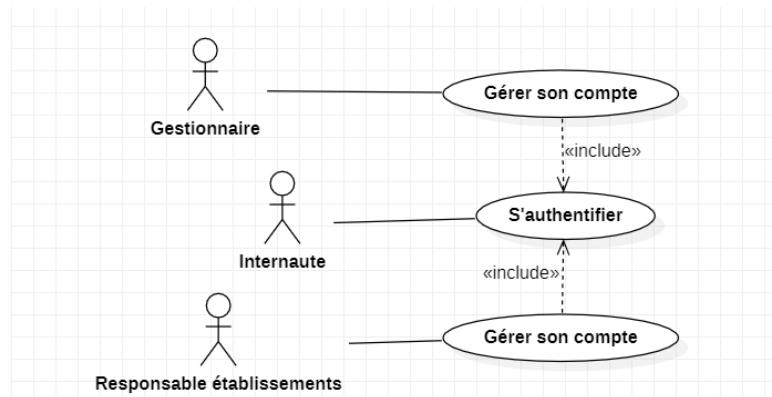


FIGURE 5.1 – Raffinement du diagramme de cas d'utilisation du Sprint 3

### Description textuelle du sous cas d'utilisation «S'authentifier»

Cas d'utilisation	S'authentifier
Acteur	Internaute
Pré-condition	L'internaute doit avoir un compte
Post-condition	L'internaute est connecté
Scénario principal	1- L'internaute entre sur la page d'accueil du site web 2- L'internaute clique sur le bouton "Se Connecter" en haut à droite 3- Le système affiche l'interface de connexion à l'internaute 4- L'internaute entre ses données de connexion dans les champs affichés 5- L'internaute clique sur "Se Connecter" 6- Le système vérifie les données envoyées 7- Le système affiche une fenêtre de succès de connexion 8- Le système lui affiche le Dashboard de son espace
Scénario alternatif	- Le système informe que les données saisies ne sont pas correctes et qu'il doit réessayer de se connecter

TABLE 5.2 – Description textuelle du sous cas d'utilisation «S'authentifier»

### Description textuelle du sous cas d'utilisation «Gérer son compte»

Cas d'utilisation	Gérer son compte
Acteur	Gestionnaire - Responsable de l'établissement
Pré-condition	L'utilisateur est connecté
Post-condition	L'utilisateur a modifié ses informations personnelles
Scénario principal	1- L'utilisateur clique sur la photo de son profil en haut à droite 3- Ses informations personnelles sont affichées dans une fenêtre à droite 4- L'utilisateur clique sur le bouton "Modifier" 5- Le système lui redirige vers une nouvelle page intitulée "Modifier Profil" 6- Les informations de son compte lui sont affichées 7- Après avoir modifié ses informations, il clique sur le bouton "Confirmer" 8- Un message de succès de modification lui s'affiche
Scénario alternatif	Un message d'échec de modification lui s'affiche

TABLE 5.3 – Description textuelle du sous cas d'utilisation «Gérer son compte»

## 5.2 Conception

### 5.2.1 Diagramme de séquence de conception

Pour les diagrammes de séquence système de conception, on débutera par bien expliquer le déroulement du cas d'authentification de l'internaute soit comme gestionnaire soit comme

responsable d'établissement.

Nous pouvons trouver ci-dessous le diagramme de séquence système de conception du cas d'utilisation «S'authentifier» :

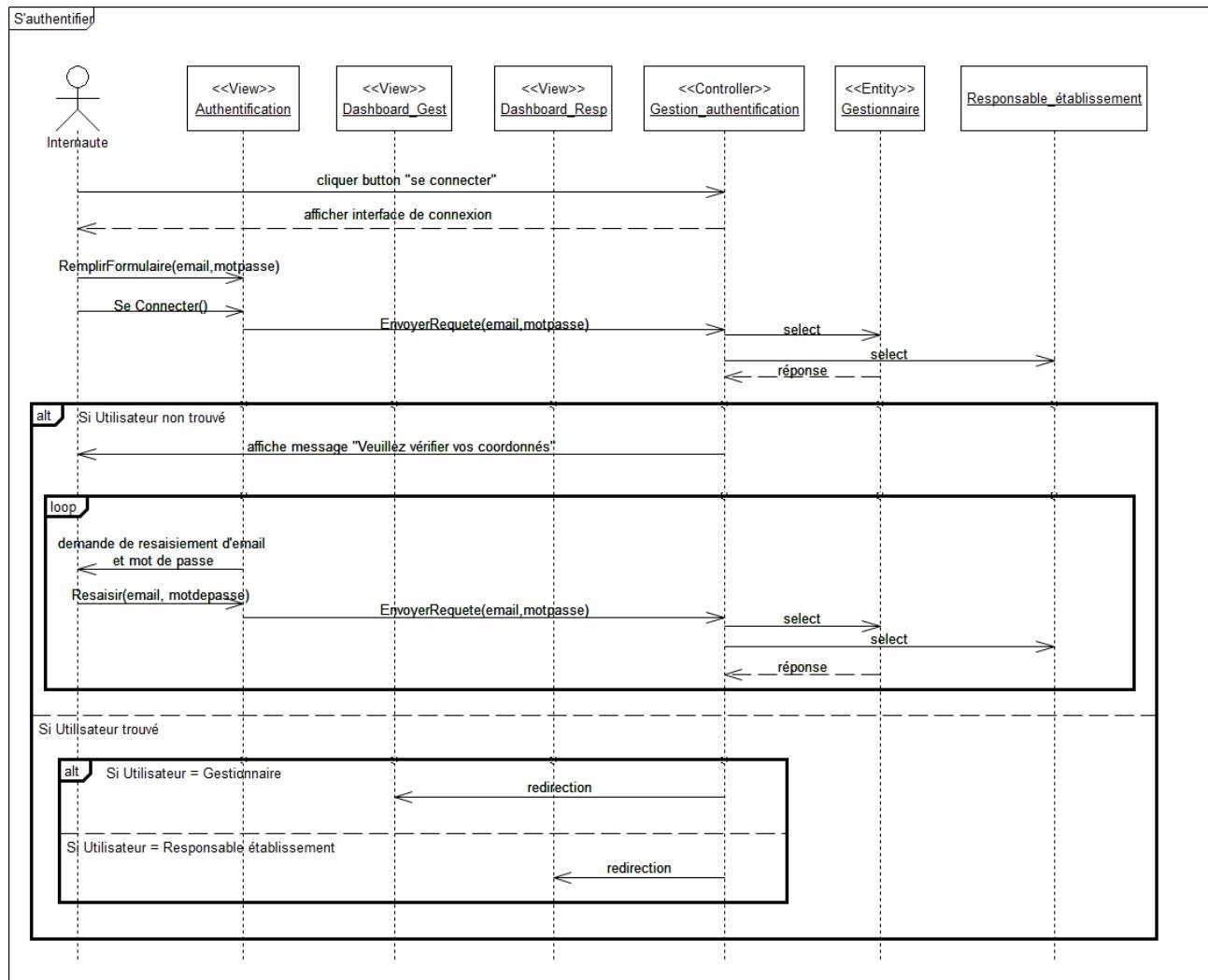


FIGURE 5.2 – Diagramme de séquence système de conception du cas d'utilisation «S'authentifier»

Le diagramme trouvé dans la figure 5.2 détaille l'ensemble d'étapes à effectuer pour que l'internaute puisse se connecter comme étant gestionnaire ou bien responsable d'établissement. L'internaute doit cliquer sur le bouton "Se Connecter" pour que le système lui affiche l'interface de connexion. Il doit

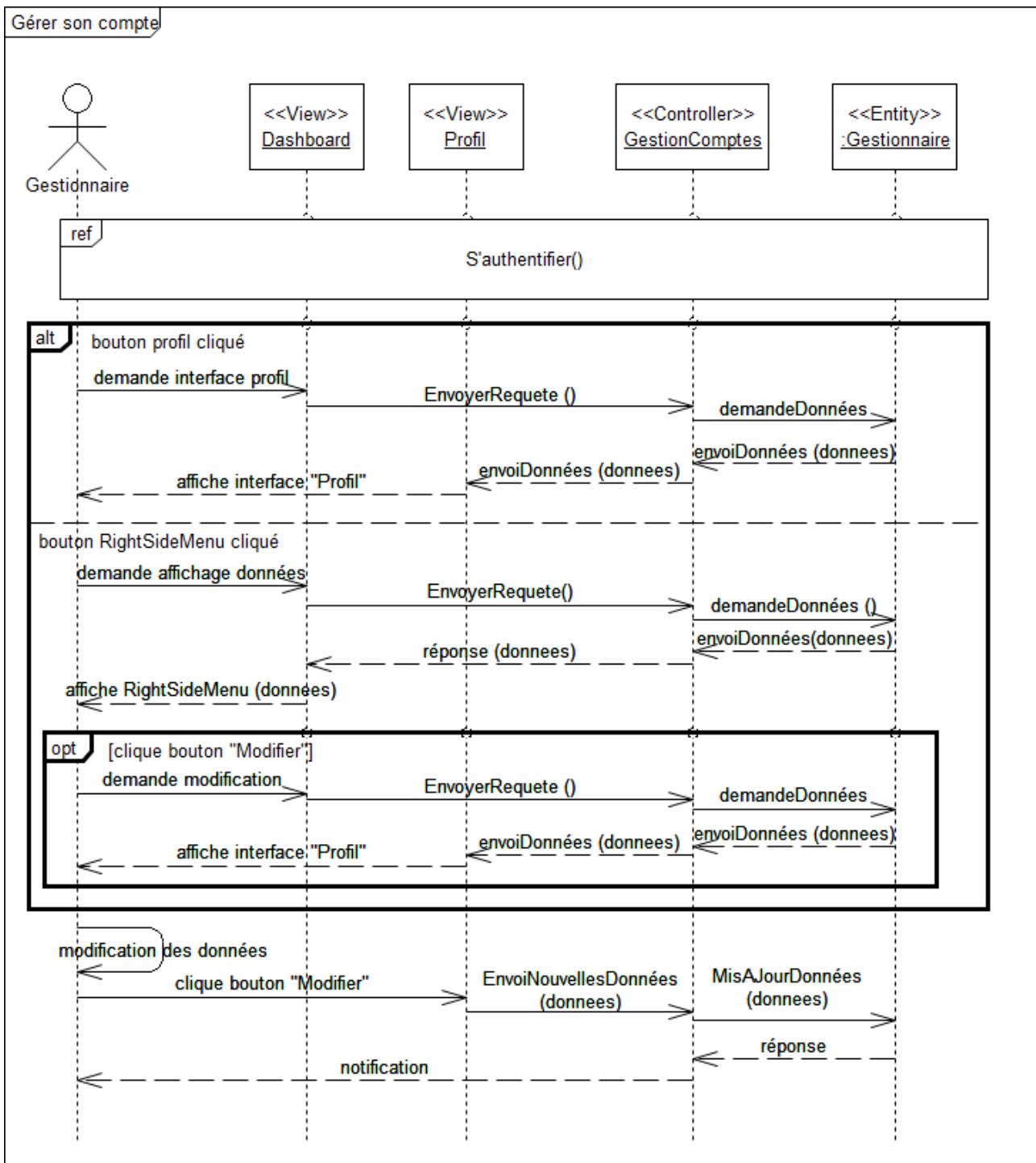


FIGURE 5.3 – Diagramme de séquence système de conception du cas d'utilisation «Gérer son compte» pour le gestionnaire

La figure 5.3 présente le diagramme de séquence système de conception du cas d'utilisation «Gérer son compte» pour l'acteur gestionnaire.

De même pour le responsable d'établissement, il peut modifier ses données de son profil

### 5.2.2 Diagramme de classe global du troisième sprint

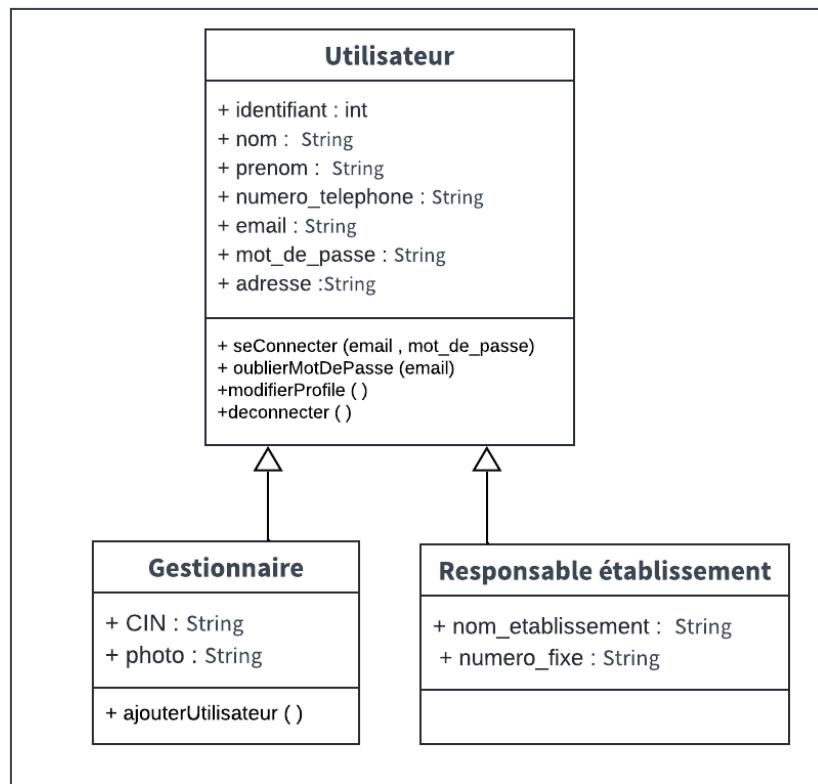


FIGURE 5.4 – Diagramme de classe du Sprint 3

## 5.3 Codage

### 5.3.1 Schéma de la base de données

Champs	Types	Contraintes
id	bigint(20)	Not NULL, Primary Key
nom	varchar(20)	Not NULL
prenom	varchar(20)	Not NULL
CIN	varchar(255)	Not NULL
numero_telephone	varchar(255)	Not NULL
email	varchar(40)	Not NULL
photo	varchar(255)	-
mot_de_passe	varchar(255)	Not NULL
adresse	varchar(255)	Not NULL
remember_token	varchar(100)	-

TABLE 5.4 – table gestionnaires

Champs	Types	Contraintes
id	bigint(20)	Not NULL, Primary Key
nom	varchar(20)	Not NULL
prenom	varchar(20)	Not NULL
CIN	varchar(255)	Not NULL
numero_telephone	varchar(255)	Not NULL
email	varchar(40)	Not NULL
photo	varchar(255)	-
mot_de_passe	varchar(255)	Not NULL
adresse	varchar(255)	Not NULL
remember_token	varchar(100)	-

TABLE 5.5 – table responsables\_etablissements

### 5.3.2 Extraits du code

Dans cette partie on va essayer de montrer une partie de code développé.

Cette figure montre le code nécessaire pour faire travailler la partie d'authentification des différents acteurs.

```

const navigate = useNavigate();
const [loginInput, setLoginInput] = React.useState({email: '', mot_de_passe: '', showPassword: false, error_list: []});
const handleInput = (e) => {
  e.persist();
  setLoginInput({ ...loginInput, [e.target.name]: e.target.value });
};
const loginSubmit = (e) => {
  e.preventDefault();
  const data = {
    email: loginInput.email,
    mot_de_passe: loginInput.mot_de_passe,
  }
  axios.get('sanctum/csrf-cookie').then(response => {
    axios.post(`api/auth-gestionnaire/login`, data).then(res =>{
      if(res.data.status === 200){
        localStorage.setItem('auth_token', res.data.token);
        localStorage.setItem('auth_email', res.data.gestionnaire.email);
        window.location.reload();
        navigate("/gestionnaire");
        Swal('Success', res.data.message, "success");
      }else if(res.data.status === 401){
        console.log(res.data.validation_errors)
        Swal ('Oops' , res.data.validation_errors , "error" )
        setLoginInput({...loginInput,error_list:res.data.validation_errors});
      }
    })
  });
};

```

FIGURE 5.5 – Extrait du code de Login.js

Cette partie du code nous permet d'afficher les données d'utilisateur connecté à droite de l'écran comme on peut le voir dans la figure 5.16.

```

c > interface > gestionnaire > components > RightSidebar > ProfileGestionnaire.js > ProfileGestionnaire
1  import React, {useState, useEffect} from 'react'
2  import Box from '@mui/material/Box';
3  import Skeleton from '@mui/material/Skeleton';
4  import ProfilePhoto from '../../../../../Global/images/default_profile_image.jpg'
5  import { Button } from '@mui/material';
6  import { Link } from 'react-router-dom';
7  export default function ProfileGestionnaire({toggleDrawer}) {
8      var myHeaders = new Headers();
9      myHeaders.append("Authorization", `Bearer ${localStorage.getItem('auth_token')}`);
10     var requestOptions = { method: 'GET', headers: myHeaders };
11     const [profile, setProfile] = useState(null);
12     useEffect(() => {
13         const getData = () => {
14             fetch("http://127.0.0.1:8000/api/auth-gestionnaire/profile", requestOptions)
15                 .then(response => response.json())
16                 .then(response => setProfile(response))
17                 .catch(error => console.log('error', error));
18         }
19         if(profile==null){
20             getData();
21         }
22     }, []);
23     return (
24         <>
25             {profile.photo==null?<img src={ProfilePhoto} alt="profile"/>:<img src={"http://127.0.0.1:8000/storage/images/gestionnaire/${profile.photo}"} alt="gestionnaire image"/>}
26             <div style={{fontSize:'16px', margin:'20px'}}>
27                 <li><b>Nom :</b> {profile.nom}</li>
28                 <li><b>Prénom :</b> {profile.prenom}</li>
29                 <li><b>CIN :</b> {profile.CIN}</li>
30                 <li><b>Numéro téléphone :</b> {profile.numero_telephone}</li>
31                 <li><b>Email :</b> {profile.email}</li>
32                 <li><b>Adresse :</b> {profile.adresse}</li>
33             </div>
34             <Link to="/gestionnaire/profile"> <Button onClick={toggleDrawer} color="primary" variant="contained">Modifier profile</Button></Link>
35         </>
36     )>;
37     else {
38         return (
39             <Box sx={{padding:'1px 30px'}}>
40                 <Skeleton sx={{height:'250px', width:'250px'}} variant="circular"/>
41                 <Skeleton sx={{height:60}} />
42             </Box>
43         );
44     }
45 }
46

```

FIGURE 5.6 – Extrait du code de ProfileGestionnaire.js

Le code trouvé dans la figure 2 5.7 porte sur la modification des données de l'utilisateur.

```

import React, { useState } from 'react';
import { Button, FormHelperText, TextField } from '@mui/material';
export default function InputUpdate() {
    const initialValue = { nom: "", prenom: "", CIN: "", numero_telephone: "", email: "", adresse:"",created_at:"", updated_at:"", error_list:[], validation:[], validation_error:[], data:[], setData:[], show:[], handleFormSubmit:(e) => {
        fetch("http://127.0.0.1:8000/api/auth-gestionnaire/modifier-profile-gestionnaire", {
            method: 'PUT',
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer ${localStorage.getItem('auth_token')}`
            },
            body: JSON.stringify(e.target.value)
        })
        .then(resp => resp.json())
        .then(resp => {
            if(resp.validation_error){
                setValidation(resp.validation_error)
            }else{
                setData(resp.gestionnaire)
                window.location.reload()
                console.log('Success',resp.message,"success")
                localStorage.setItem('auth_email',resp.gestionnaire.email);
            }
        })
        .catch(err => {
            console.log("Error Reading data " + err);
        });
    },
    const onChange = (e) => {
        const { value, id } = e.target
        setData({ ...data, [id]: value })
        console.log(data)
    }
    return (
        <>
            {show.length!=0 ?(show.map(sh =>
                (sh[1]==="id" && sh[1]==="created_at" && sh[1]==="updated_at" && sh[1]==="photo")?( 
                    <TextField id={sh[1]} onChange={(e=>onChange(e)} focused placeholder={localStorage.getItem("auth_"+$[sh[1])}} error={!!validation[sh[1]]} label={sh[0]} variant="outlined" margin="dense" fullWidth />
                    <FormHelperText error={true}>
                        {validation[sh[1]]}
                    </FormHelperText>
                )>: null
            )):null
        )
        <Button variant="contained" className='tableIcon' color="primary" onClick={()=>handleFormSubmit()}>modifier</Button>
    </>
)
}

```

FIGURE 5.7 – Extrait du code de inputUpdate.js

## 5.4 Tests

Nous présentons dans cette partie quelques tests réalisés pendant la réalisation des tâches de ce Sprint.

L'accès à l'application est protégé par un système d'authentification permettant la connexion des utilisateurs.

En entrant sur le site, l'internaute se trouve sur la page se trouvant au-dessous dans la figure 5.8.



FIGURE 5.8 – Interface de l'internaute

Lors du clique de l'internaute sur le bouton "Se Connecter" en haut de la page d'accueil, le système le redirige vers la page de connexion illustré dans la figure 5.9 pour entrer ses coordonnées d'authentification.

FIGURE 5.9 – Interface de l'authentification

L'internaute doit entrer ses données de connexion : email et mot de passe pour accéder à

l'application de gestion sans oublier de cliquer sur la case à cocher du test CAPTCHA.

On suppose que l'internaute clique sur le bouton "Se Connecter" sans avoir saisi aucune coordonnée comme figurant dans la figure 5.10, le système lui affiche des messages d'erreur pour lui informant de l'obligation de leur saisi.

The screenshot shows the 'Espace Gestionnaire' login interface. At the top is a green circular icon with a lock symbol. Below it, the title 'Espace Gestionnaire' is centered. The form consists of several input fields and labels:

- Adresse Email**: An input field with a placeholder 'Entrer votre email'. A red border surrounds the field, and a red error message 'Le champ email est obligatoire.' is displayed below it.
- Mot de passe**: An input field with a placeholder 'Entrer votre mot de passe'. A red border surrounds the field, and a red error message 'Le champ mot de passe est obligatoire.' is displayed below it.
- Souvenez de moi**: A checkbox labeled 'Souvenez de moi'.
- reCAPTCHA**: A reCAPTCHA verification box containing a checkbox 'Je ne suis pas un robot' and the reCAPTCHA logo.
- SE CONNECTER**: A large green rectangular button with white text.
- Avez-vous oublier votre mot de passe ?**: A blue link at the bottom right of the form.

FIGURE 5.10 – Test de l'authentification (champs obligatoires)

The screenshot shows the same 'Espace Gestionnaire' login interface as Figure 5.10, but with different error messages due to incorrect input:

- Adresse Email**: The input field contains 'test@gmail.com', which is invalid, resulting in the error message 'Le champ email sélectionné est invalide.'
- Mot de passe**: The input field contains a series of dots ('.....'), which is invalid, resulting in the error message 'Le champ mot de passe est invalide.'
- Souvenez de moi**: A checkbox labeled 'Souvenez de moi'.
- reCAPTCHA**: A reCAPTCHA verification box containing a checkbox 'Je ne suis pas un robot' and the reCAPTCHA logo.
- SE CONNECTER**: A large green rectangular button with white text.
- Avez-vous oublier votre mot de passe ?**: A blue link at the bottom right of the form.

FIGURE 5.11 – Test d'entrée de données incorrects

Lorsqu'il entre les mauvaises données (figure 5.11), email et/mot de passe un message d'erreur lui s'affiche sur l'écran comme indiqué dans la figure 5.13.

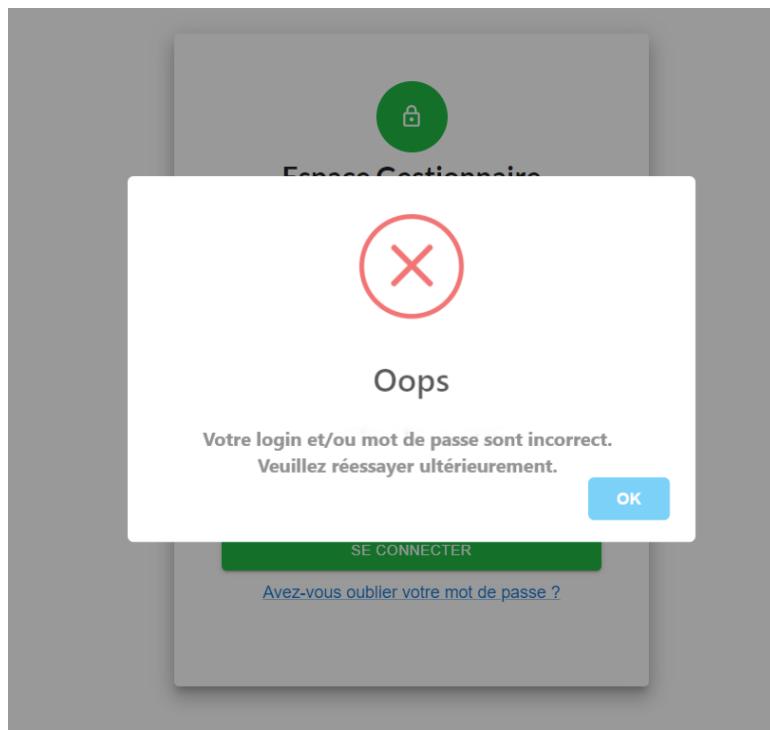


FIGURE 5.12 – Affichage de message d'échec de connexion

Sinon s'il entre les bonnes données, un message de succès lui s'affiche sur l'écran.

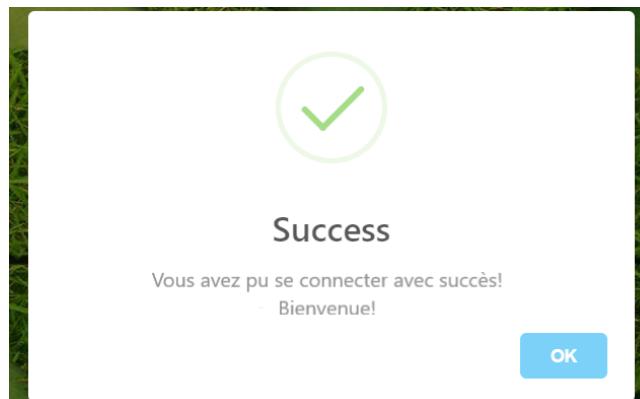


FIGURE 5.13 – Affichage de message de succès de connexion

Après avoir saisi ses coordonnées correctement, le système lui redirige vers son propre espace que soit le dashboard du gestionnaire ou du responsable d'établissement.

Nous trouvons dans la barre en haut à droite deux méthodes pour accéder aux informations du compte de l'utilisateur :

- **1ère Méthode** : en cliquant sur l'icône du photo de profil, puis en appuyant sur "Mon Profil".
- **2ème Méthode** : en cliquant sur les trois points et un menu à droite lui s'affiche pour l'affichage de ses données ensuite il clique sur le bouton "Modifier Profil".

Nous prendrions le gestionnaire comme un exemple pour voir l'interface de l'affichage des données de son compte.

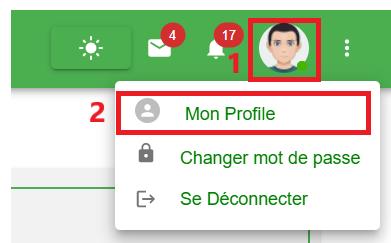


FIGURE 5.14 – Test de la 1ère méthode de gestion de compte

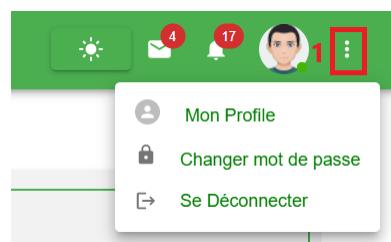


FIGURE 5.15 – Test de la 2ème méthode de gestion de compte -partie 1

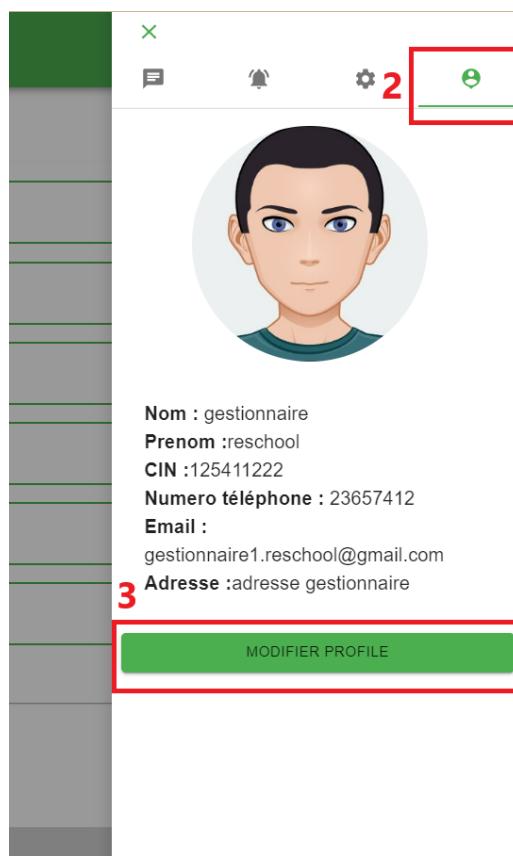


FIGURE 5.16 – Test de la 2ème méthode de gestion de compte -partie 2

Les 2 méthodes illustrées dans les figures 5.14, 5.15 et 5.16 nous mènerons à la page de son profil se trouvant dans la figure 5.17 .

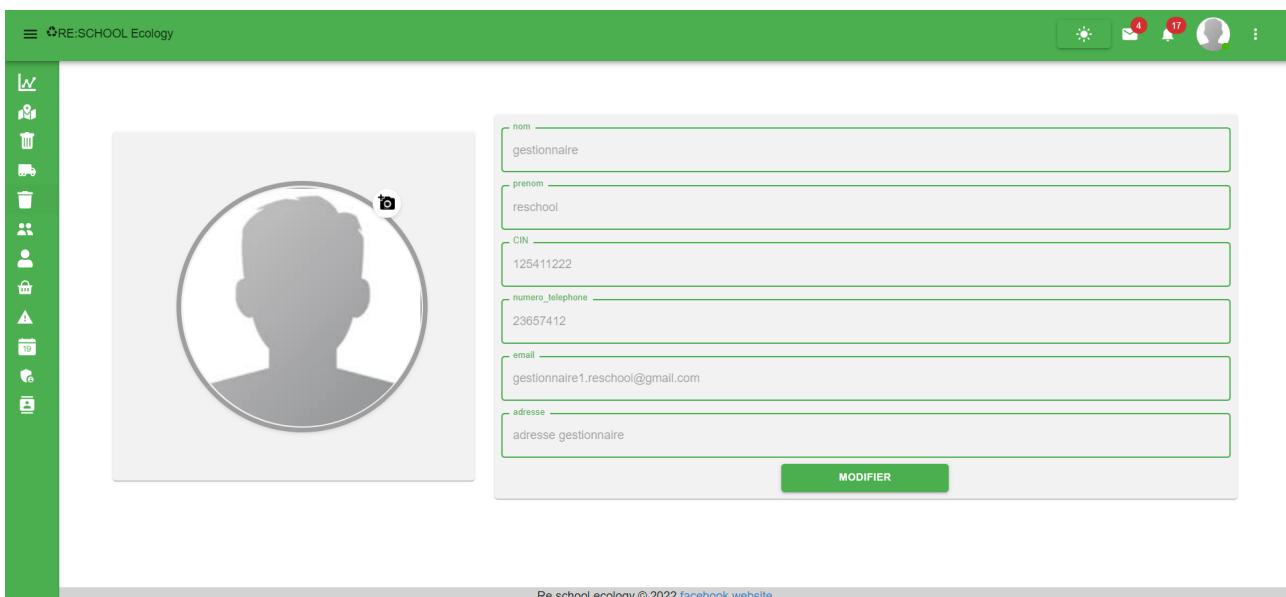


FIGURE 5.17 – Interface du Profil

L'utilisateur (gestionnaire ou responsable d'établissement) a l'accès de modifier ses informations personnelles en entrant sur l'interface de son profil.

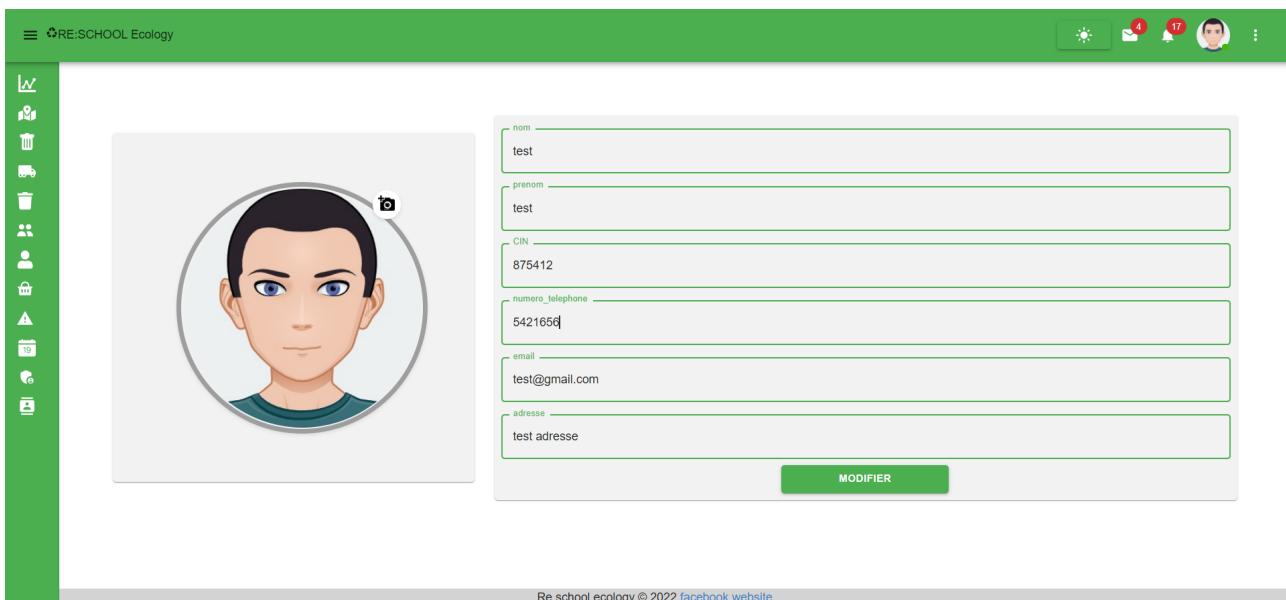


FIGURE 5.18 – Interface de modification des données du compte

Nous présumons que le gestionnaire entre des données de forme incorrecte, le système donc lui affichera un message en rouge en-dessous de chaque champ de saisie incorrect comme illustré dans la figure 5.19.

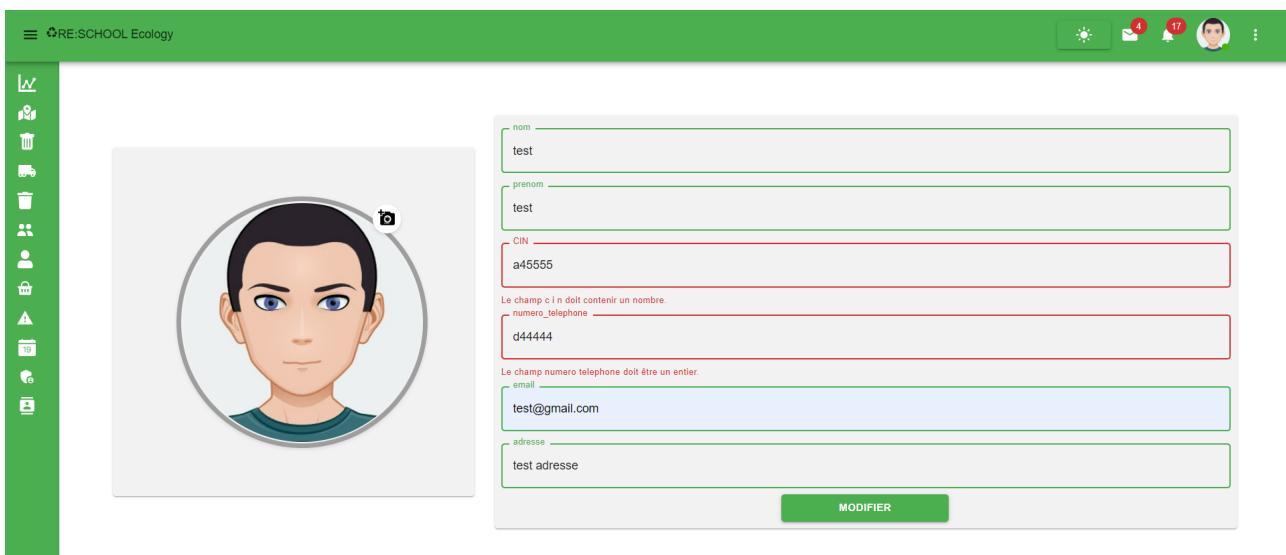


FIGURE 5.19 – Test de modification des données incorrectes du compte

## 5.5 Sprint Review

### 5.5.1 Livraison du sprint 3

A la fin de notre troisième sprint, nous avons achevé avec un livrable complet de l’application attendue de nous.

Nous avons ajouté la partie de l’authentification des utilisateurs avec sécurisation comme nous avons également terminé la partie de l’affichage et modification de leurs coordonnées.

Ainsi, nous pouvons dire que l’application a été finalisé avec succès.

### 5.5.2 Obstacles rencontrés et solutions proposées

Pour les obstacles rencontrés au cours de la réalisation de ce sprint, on peut citer :

- Nous avons beaucoup réfléchi au choix de la méthode de stockage des jetons de l’authentification des acteurs puisqu’il en existe plusieurs, mais finalement nous avons eu recours à la méthode de stockage local.
- Nous avons discuté à plusieurs reprises la création d’une seule ou de deux interfaces d’authentification : une pour le gestionnaire et une pour les clients (responsables des établissements). En fin de compte, nous avions choisi de travailler sur un seul espace de connexion.

## Conclusion

On se retrouve actuellement à la fin de ce chapitre et avec un produit livrable. On a accomplis le dernier Sprint de notre application.

Dans le chapitre suivant, nous allons clôturer notre travail en exposant les logiciels et les langages utilisés, l’environnement de développement de cette application et l’architecture employée.

# Clôture du projet

## Introduction

La clôture du projet est la dernière phase du processus Scrum marquant la fin du projet. Au cours de cette phase, on présentera l'environnement de travail matériel et logiciel de ce projet ainsi que les technologies qu'on avait recours à sans oublier de présenter l'architecture choisie.

### 6.1 Environnement de développement

#### 6.1.1 Environnement matériel

Ce tableau ci-dessous présente les caractéristiques des machines de développement que nous avons travaillés dessus :

<b>Propriétaire</b>	Ben Chaabene Feryel	Cherni Rihab
<b>Nom de l'ordinateur</b>	DESKTOP-BGEJ2T4	DESKTOP-61T0R0P
<b>Processeur</b>	Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz, 2112 MHz	Intel(R) Core(TM) i5-1135G7 CPU @ 2.40GHz
<b>RAM</b>	12 Go	16 Go
<b>Disque dur</b>	930 Go	512 Go
<b>Système d'exploitation</b>	Microsoft Windows 10 Professionnel	Microsoft Windows 10 Professionnel

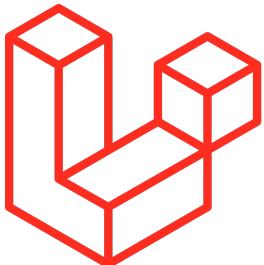
TABLE 6.1 – Description des machines de développement

### 6.1.2 Environnement logiciel



Le Visual Studio Code est un environnement open-source multi langages léger publié sous une licence de produit Microsoft. Il est compatible avec Windows, Linux et Mac. VS Code prend en charge les langages JavaScript, Java, Node.js, PHP, Python, C++ et .Net. Il fournit aux développeurs un outil de débogage, une coloration syntaxique intelligente et l'auto-complétion.

Overleaf est un éditeur en ligne permettant de faire l'édition du texte en LATEX de manière collaboratif en temps réel. Il permet la création des fichiers LATEX en se référant à des templates ainsi que l'importation des fichiers LATEX compressés.



Publié en Laravel est un framework open-source PHP permettant de développer des applications web respectant le principe modèle-vue-contrôleur. Laravel permet la création d'authentification, routage, une cartographie objet-relationnelle, migration de base de données ...

PowerAMC (SAP PowerDesigner) est un logiciel de modélisation de traitements informatiques créé par SDP. Ce logiciel prend en charge des formats variés comme par exemple BPM, CDM et PDM. Il stocke ces fichiers dans un Référentiel.



StarUML est un logiciel gratuit de modélisation Agile facile à utiliser en plus il est flexible, rapide et intégré avec de divers fonctionnalités et fonctions efficaces. Ce logiciel permet la création des diagrammes UML de manière simple.

Git est un système de contrôle de version décentralisé développé en 2005 par Linus Torvalds. Git a pour but d'être le meilleur de côté performance, sécurité et flexibilité.



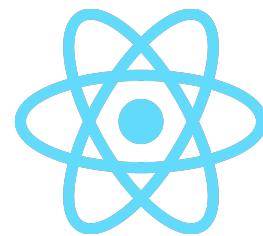
Publié en 2008, Github est un site web qui utilise le logiciel de gestion Git permettant le partage du code source d'un projet mis en œuvre par des multiples développeurs.

Trello est un outil d'aide et de collaboration gratuit pour la gestion des projets plus précisément l'organisation de tâches. Il est inspiré des méthodes Agiles.



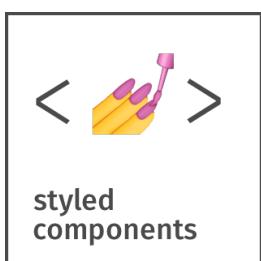
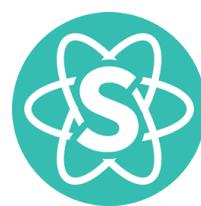
### 6.1.3 Technologies et langages utilisées

React est une bibliothèque open-source JavaScript développée pour faciliter la création des interfaces utilisateurs établies et maintenue par Facebook depuis 2013. ReactJS est basé sur les composants travaillant avec un DOM virtuel (Document Object Model).



Material UI est un framework frontend React UI, fournissant des composants React autonomes pour un développement Web simple et rapide.

Semantic-UI React est un framework de développement JavaScript libre supporté par tous les systèmes d'exploitation. Il fournit des composants React utilisant des APIs déclaratives qui fournissent des fonctionnalités robustes et une validation des accessoires.



Styled Components est une bibliothèque de CSS-in-JS. Elle nous permet de styler les composantes React en toute liberté à l'aide de littéraux de modèles étiquetés.

JavaScript est un langage de programmation de script léger permettant l'implémentation d'un contenu mis à jour dynamiquement. En plus, JS est un langage orienté objet.



## HTML



HTML est un langage de balisage hypertexte standard pour les pages Web exploité pour structurer leur contenu. Il utilise des « balises » pour représenter ses éléments. Il est indispensable à la création d'une page.

## CSS

CSS, abrégée de «Cascading StyleSheets», met en forme les pages HTML ou XML également il est basé sur des règles de style. Avec CSS, on peut rendre le site responsive.



## PHP



PHP est un langage de programmation libre orienté objet, conçu pour la création des pages Web dynamiques via un serveur HTTP. Il est considéré parmi l'un des bases de la création de sites web dynamiques.



Développé par Isaac Z. Schlueter, npm est le gestionnaire de paquets officiel de Node.js. Il permet d'interagir avec les serveurs et les navigateurs à l'aide des lignes de commande.

## 6.2 Architecture d'implémentation

L'architecture est la structure générale d'un système qui peut renfermer des sous-systèmes qui ont des liaisons avec d'autres sous-systèmes. Elle est une sorte d'engagement à la solution qui interviendra avant son développement.

### 6.2.1 Architecture logicielle

L'architecture logicielle est la structure de niveau supérieur d'un système logiciel. Elle décrit d'une manière symbolique et schématique les éléments d'un ou de plusieurs systèmes informatiques et leurs interactions.

L'une des plus utilisées des architectures logicielles est l'architecture MVC (Modèle-Vue-Contrôleur) qui celle employée dans notre projet.  
Cette architecture est composée de 3 entités, comme le montre la figure ci-dessous :

- Modèle** : gère les données de votre site, permet l'accès à la base de données pour la récupération des informations.
- Vue** : réalise l'interface avec l'utilisateur. Elle assure essentiellement l'affichage des données.
- Contrôleur** : gère la logique du code qui est responsable de la prise de décision et assure la synchronisation entre le modèle et la vue.

Dans cette architecture, le contrôleur demande au modèle les données, le modèle les renvoie au contrôleur qui réalisera l'analyse et le traitement de ces données. Après il les transmet à la vue qui sera responsable de l'affichage.

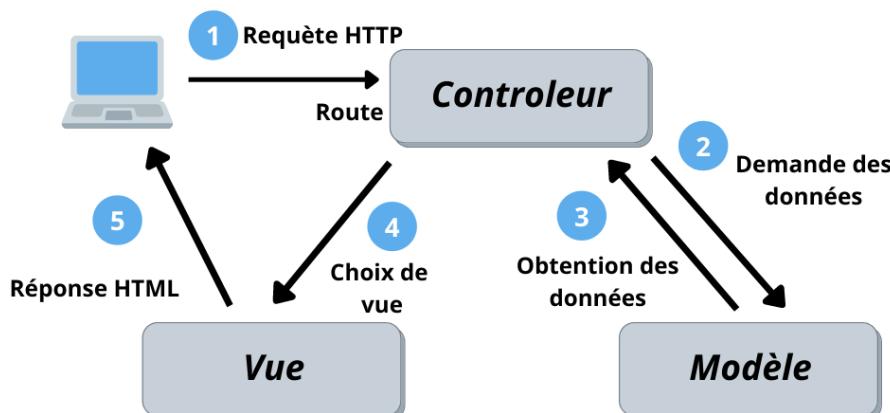


FIGURE 6.1 – Architecture MVC

### 6.2.2 Gestion de projet

Dans cette section nous allons présenter les différents artefacts Scrum que nous avions utilisé pour accomplir notre projet afin de nous faciliter la gestion des tâches.

#### Scrum blocking board

Maintenant, nous allons présenter le Scrum Blocking Board de notre deuxième sprint. Un Scrum Board est un outil permettant la suivi visuelle des tâches réalisées et leur statut dans chaque sprint puisque nous ne disposons pas d'un repère en travaillant Scrum.

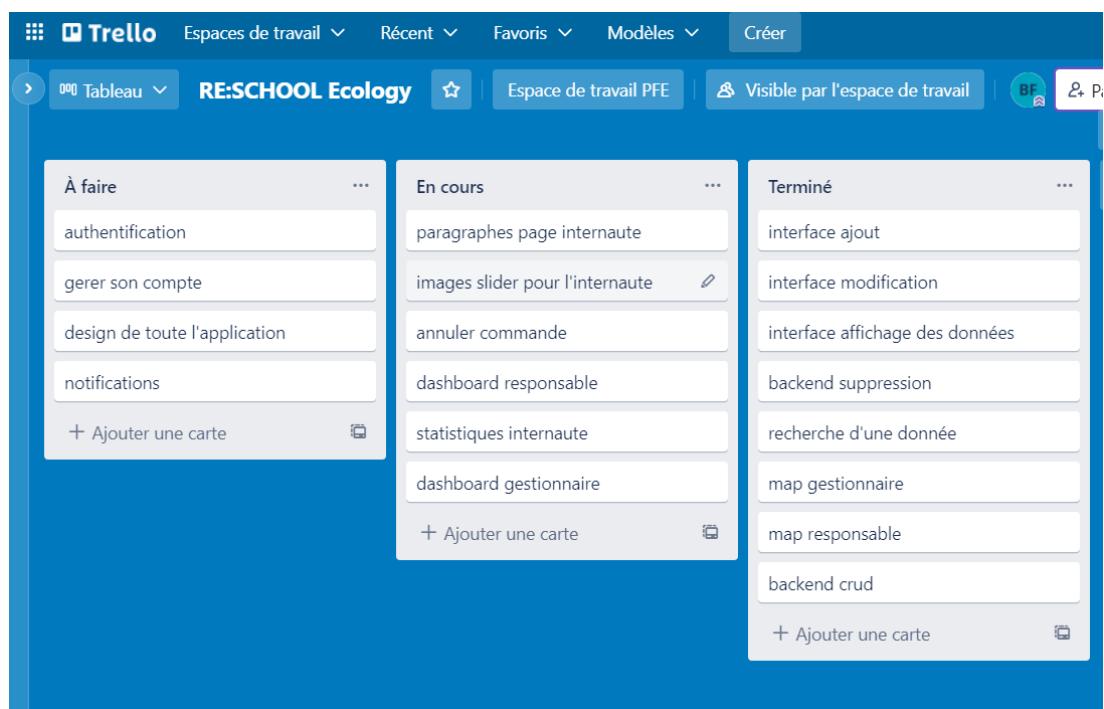


FIGURE 6.2 – Scrum Blocking Board du deuxième Sprint

### Burndown Chart

Un Burndown chart est un graphe représentant le flux du travail réalisé tout au long du projet en fonction de temps. On trouve dans la figure 6.3 notre Burndown chart :

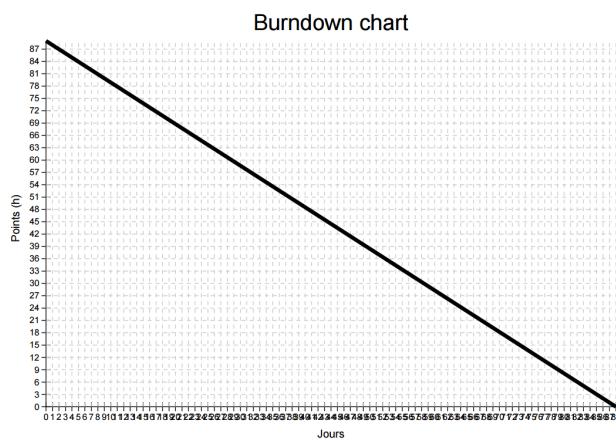


FIGURE 6.3 – Burndown Chart

## Conclusion

Nous avons invoqué dans ce chapitre l'architecture de l'application créé tout en présentant les différents technologies appliquées dans sa réalisation sans oublier de présenter les artefacts Scrum.

# Conclusion générale et Perspectives

Durant ce stage de trois mois, nous avons eu la chance de modéliser et concevoir une application web chargée de la gestion des poubelles.

Le projet a comme objectif la création d'une solution technologique pour la collecte et le tri des déchets afin de rendre le travail des agents de collecte (ouvriers) plus facile et fluide.

Tout au long de la réalisation de ce projet, nous avons eu recours à la méthodologie Scrum en divisant les tâches à réaliser en sprints qui sont composés de quatre phases : analyse, conception, codage et test.

En premier lieu, nous avons commencé par mettre le projet dans son contexte général et énoncé notre problématique et la solution proposée à la société RE:SCHOOL. Puis nous avons passé à lister les différentes exigences et besoins du Product Owner et la planification de notre plan de travail.

En deuxième lieu, nous avons divisé les tâches en trois Sprints individuels en détaillant les quatre phases mentionnées au-dessus. Pour finir, nous avons clôturé par la mention des technologies et langages utilisées et l'architecture choisie pour la création de cette application.

Lors de la réalisation de ce projet, nous avons appris à travailler avec de nouvelles technologies tout en exploitant les connaissances acquises durant notre formation.

Enfin, bien que le système soit approuvé par la société, il reste encore ouvert à plusieurs perspectives à envisager pour l'améliorer tel que l'ajout de plus de statistiques pour rendre le dashboard plus détaillé, un chatbot pour rendre la communication entre le responsable d'établissement et le gestionnaire plus facile et de rendre l'application bilingue.

# Bibliographie

- [1] Ken Schwaber and Jeff Sutherland. Le Guide de Référence de Scrum : Les Règles de Jeu. <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-French.pdf>, 2017. page 5, Novembre 2017, France.
- [2] Ken Schwaber and Jeff Sutherland. Le guide scrum. page 3. France, 2017.
- [3] Ken Schwaber and Jeff Sutherland. Le guide scrum. pages 6–9. France, 2017.
- [4] Mercator. La méthode agile comme méthode de travail chez mercator : explications. <https://www.mercator.eu/en/la-methode-agile-comme-methode-de-travail-chez-mercator-explications.chtml>, Belgique, 2018.