



N° d'ordre : **00106/LP\_GL/2025**

**MÉMOIRE DE LICENCE PROFESSIONNELLE DE IPNET INSTITUTE OF  
TECHNOLOGY**

Spécialité : **Génie Logiciel**

présenté par

**Yao David SOUSSOUKPO**

Promotion : **2020**

Sujet de mémoire :

**APPLICATION WEB DE CARTOGRAPHIE DES OSC  
ACTIVES SUR LES QUESTIONS DE NUTRITION ET  
D'ALIMENTATION Saine AU TOGO**

Soutenue le **06 Juin 2025**,

devant le jury composé de Messieurs :

**Président : Dr. Justin AKORO**, Génie Électrique

**Examinateur : M. Yao Mokpokpo ASSOU**, Ingénieur Logiciel

**Directeur de mémoire : M. Alou Pougnozi BATANA**, Ingénieur Logiciel

*Je dédie ce mémoire à mes parents.*

*Aussi sont-ils pour moi, une source inépuisable de sagesse, de bienveillance, et  
les plus importants des pédagogues.*

## REMERCIEMENTS

---

« Aucun travail ne s'accomplit dans la solitude.

Aussi trouvé-je normal que figurent en début d'un ouvrage, notamment d'[un mémoire], des remerciements adressés à ceux qui ont aidé, concouru à sa réalisation (...) »<sup>1</sup>

Tout d'abord, je remercie mon directeur de mémoire, **M. Alou Pougnozi BATANA**, pour sa patience, sa bienveillance et sa confiance en moi. Sa disponibilité, sa rigueur et ses remarques constructives m'ont guidé à chaque étape de mon parcours d'étudiant et de ce mémoire.

Je souhaite également exprimer ma reconnaissance à l'**équipe de consultants**, dont **M. Yao Anani Paul-Martial AGBOBLI** et **M. Yonam Eliud AGBOGAN**, avec laquelle j'ai eu le privilège de collaborer, sur la mission dont le projet est l'objet de ce mémoire. Leur professionnalisme, leur enthousiasme et leur esprit d'équipe ont été essentiels pour créer un environnement dynamique et motivant, propice à l'épanouissement du projet.

Un grand merci à **mes professeurs et encadreurs académiques**, dont les enseignements ont enrichi mes connaissances et m'ont permis de développer les compétences nécessaires à l'élaboration de ce mémoire. À cet effet, je remercie particulièrement **M. Abdoulatif Nintché SHABAN**, **M. Yao Mokpokpo ASSOU**, **M. Folly Carlo GBOSSOU<sup>2</sup>** et **M. Dessabatouma DJAOURA** pour leur expertise approfondie, leurs critiques constructives et leur soutien constant. Leur accompagnement a été une véritable source d'enrichissement intellectuel et méthodologique, et leurs conseils éclairés m'ont permis de structurer mon travail de manière rigoureuse et cohérente.

Aux très respectés **membres du jury** ; **Dr Justin AKORO**, **M. Yao Mokpokpo ASSOU** et **M. Alou Pougnozi BATANA**, je n'oublie pas d'adresser mes remerciements pour les remarques constructives et points d'améliorations suggérés.

En ce qui concerne mes frères **Elpidio**, **Bénix** et ma sœur **Oxana**, je ne saurai assez exprimer mon immense reconnaissance à Dieu pour leur présence, leur amour et leur soutien indéfectible dans cette petite vie qui m'est offerte. Les moments, aussi courts ou longs qu'ils

---

<sup>1</sup> Beaud 1985, 2006, p. 5 ; référence complète en [2].

<sup>2</sup> Qui n'intervient plus à iPNet au moment où j'écris ce mémoire.

soient, que nous passons suffisent pour alléger les stress physiques et psychologiques provoqués par les différentes épreuves de ma vie en général, et en particulier ceux de cette épreuve de mémoire de fin de Licence.

Je n'oublie pas mes ami(e)s ; **Christelle, Espoir, Martial, Ahmed, Ulrich, Majoie**, et **camarades** ; **Inès, Mathis**, pour leur présence et leurs encouragements. Leur soutien moral m'a été d'une grande aide dans les moments de doute, et leur enthousiasme m'a permis de rester motivé jusqu'au bout de ce projet.

Enfin, je remercie l'agence **Manocom** qui, de tout cœur, a aidé à la reprographie de ce mémoire.

À toutes ces personnes, avec toute ma reconnaissance, j'adresse mes sincères remerciements.

*Yao David SOUSSOUKPO*

## AVANT - PROPOS

---

Ce mémoire s'inscrit dans le cadre de ma **Licence en Génie Logiciel à iPNet Institute of Technology (IIT), promotion 2020**, en vue de l'obtention du **grade de Licence**. Pour clôturer mon parcours académique, j'ai été amené à choisir un sujet de mémoire, validé par le **Département Scolarité et Affaires Académiques**. Ce travail devait permettre de mettre en lumière les connaissances et compétences techniques et transversales acquises durant mes trois années d'études.

J'ai longtemps hésité avant de choisir mon projet, tant en raison de l'incertitude quant au sujet inédit que j'allais aborder, que de la tentation de reprendre un projet déjà réalisé. Ce dilemme a prolongé mon parcours en Licence, mais il m'a également offert l'opportunité de réfléchir à ce qui me passionnait vraiment.

Heureusement, j'ai été approché par mon oncle, **M. Yao Anani Paul-Martial AGBOBLI**, qui m'avait demandé mon avis sur un projet qu'il menait avec **M. Yonam Eliud ABOGAN** ; je devais analyser la pertinence d'une « stack »<sup>1</sup> technique proposée pour développer une application web, destinée à compléter une base de données Excel qu'ils étaient en train de développer. Sans hésitation, j'ai apporté mon analyse, bien que je ne savais pas encore l'envergure de ce que je m'apprétais à entreprendre.

Ce projet est celui de l'**Application Web de Cartographie des OSC Actives sur les Questions de Nutrition et d'Alimentation Saine au Togo**, thème choisi pour mon mémoire. Après plusieurs mois<sup>2</sup> de travail passionnant, soutenu par mon oncle, j'ai décidé d'en faire mon sujet de mémoire.

Ce document a pour but de vous faire découvrir ce projet enrichissant et de vous transmettre la passion qui m'a animé tout au long de ce parcours. J'espère que ce mémoire pourra apporter une contribution utile aux initiatives technologiques en cours et inspirer d'autres projets pour le développement des OSC et l'amélioration des services communautaires au Togo.

---

<sup>1</sup> L'utilisation du « franglais », dans un travail de [mémoire] est défendu, par certains professeurs. Toutefois, par manque d'un mot en français qui traduirait l'exact sens de certains mots anglais, comme « stack » dans notre contexte, et d'autres qui apparaîtront dans la suite du mémoire pour les mêmes raisons ou parce que le jargon informatique le tolère ; je me permets de le maintenir. Voir Beaud 1985, p. 68, note 1 ; référence complète en [2].

<sup>2</sup> De Juillet 2024 à Octobre 2024.

## RÉSUMÉ

---

L'écriture de ce mémoire s'articule autour du défi que représente la mise en place d'une plateforme web pour cartographier les OSC actives sur les questions de Nutrition et d'Alimentation saine au Togo. Cette problématique s'inscrit dans le cadre du projet **EU4SUN<sup>1</sup>**, mené en partenariat entre **Expertise France**, le **Mouvement SUN** et plusieurs acteurs locaux.

Il met l'accent sur l'accessibilité des données et la coordination entre les acteurs, l'étude analyse les défis liés à la gestion et à la mise à jour des bases de données des OSC. Pour garantir la fiabilité et la pertinence des informations collectées, une méthodologie rigoureuse a été adoptée. Technique, la plateforme repose sur le framework **Django** et la base de données **PostgreSQL**, tout en intégrant **Leaflet.js** pour la cartographie interactive. Son développement a suivi plusieurs étapes clés, incluant la modélisation des données, l'implémentation des fonctionnalités essentielles et la mise en production sur un serveur.

Un soin particulier a été apporté à l'ergonomie et à l'expérience utilisateur afin de faciliter l'adoption de la plateforme. Le mémoire propose également un guide d'utilisation décrivant les différents profils utilisateurs et leurs droits d'accès, ainsi qu'un manuel d'exploitation destiné aux administrateurs. Pour l'avenir, une version améliorée reposant sur **React**, **Django REST Framework**, un **chatbot renforcé par l'IA<sup>2</sup>**, est envisagée afin d'optimiser la performance et la modularité de l'application. Ce projet incarne l'application des compétences acquises au cours de la formation en génie logiciel à iPNet et constitue une contribution majeure à une initiative nationale visant à renforcer la visibilité et l'impact des OSC dans ce domaine essentiel.

---

<sup>1</sup> EU4SUN : European Union for Scaling Up Nutrition

<sup>2</sup> De l'anglais, AI Powered Chatbot, il s'agit d'outil conversationnel textuel (basé sur l'IA : Intelligence Artificielle) pouvant répondre, de façon personnalisée aux visiteurs/utilisateurs et ce avec une disponibilité 24/7, aux questions relatives à la plateforme et au paysage de la nutrition et de l'alimentation saine au Togo.

## ABSTRACT

---

The writing of this thesis centers on the challenge of establishing a web platform to map Civil Society Organizations (CSOs) active in the fields of nutrition and healthy eating in Togo. This issue falls within the framework of the **EU4SUN project**, a partnership initiative involving **Expertise France, the SUN Movement**, and several local stakeholders.

Through a thorough contextual analysis, it examines challenges in managing and updating CSO databases and outlines methodological approaches to ensure data reliability. The technical architecture is built on **Django** and **PostgreSQL**, integrating **Leaflet.js** for interactive mapping. Development involved key phases, including data modeling, implementation of core functionalities, and deployment in a server environment, with a strong emphasis on usability and user experience to encourage adoption.

Additionally, the thesis provides a comprehensive user guide detailing roles, access rights, and an operational manual for platform administrators. Looking ahead, future enhancements include a transition to a **React + Django REST Framework** architecture for improved performance and modularity and the integration of an **AI Powered Chatbot** for user support. This project not only demonstrates the application of knowledge gained during software engineering studies at iPNet but also contributes significantly to a national initiative aimed at increasing CSO visibility and impact in nutrition and healthy eating.

## SOMMAIRE

---

<b>REMERCIEMENTS .....</b>	<b>ii</b>
<b>AVANT - PROPOS .....</b>	<b>iv</b>
<b>RÉSUMÉ .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>LISTE DES TABLEAUX .....</b>	<b>ix</b>
<b>LISTE DES FIGURES .....</b>	<b>x</b>
<b>GLOSSAIRE .....</b>	<b>xv</b>
<b>INTRODUCTION GÉNÉRALE .....</b>	<b>1</b>
<b>Chapitre 1 : PRÉSENTATION DES CADRES DE FORMATION ET DE TRAVAIL .....</b>	<b>3</b>
1.1. Cadre de formation : iPNet Institute of Technology (IIT) .....	4
1.2. Cadre professionnel : Projet EU4SUN - Développeur Web & Consultant Junior ....	11
Conclusion .....	15
<b>Chapitre 2 : ÉTUDE PRÉALABLE DU PROJET .....</b>	<b>17</b>
2.1. Présentation du projet .....	18
2.2. Problématique .....	19
2.3. Étude de l'existant .....	21
2.4. Critique de l'existant .....	22
2.5. Objectifs du projet et résultats attendus .....	22
2.6. Proposition de solutions .....	23
2.7. Évaluation financière du projet .....	26
2.8. Planning prévisionnel de réalisation du projet .....	28
2.9. Démarches méthodologiques .....	32
Conclusion .....	42
<b>Chapitre 3 : ANALYSE ET CONCEPTION .....</b>	<b>43</b>
3.1. Présentation de la méthode d'analyse .....	44
3.2. Présentation du langage de modélisation .....	46
3.3. Étude détaillée du projet .....	47
3.4. Choix techniques des solutions .....	67
Conclusion .....	79
<b>Chapitre 4 : RÉALISATION ET MISE EN ŒUVRE .....</b>	<b>80</b>

4.1. Mise en œuvre .....	81
4.2. Création de la base de données : Script de création ou ORM ? .....	95
4.3. Quelques captures des interfaces et codes sources des applications .....	96
Conclusion .....	102
<b>Chapitre 5 : GUIDE D'EXPLOITATION .....</b>	<b>103</b>
5.1. Configurations matérielles et logicielles .....	104
5.2. Déploiement et suivi .....	106
Conclusion .....	124
<b>Chapitre 6 : GUIDE D'UTILISATION .....</b>	<b>126</b>
6.1. Layout des pages de l'application web .....	127
6.2. Utilisateurs : Accès, Rôles & Responsabilités .....	131
Conclusion .....	153
<b>CONCLUSION GÉNÉRALE .....</b>	<b>154</b>
<b>BIBLIOGRAPHIE ET WEBOGRAPHIE .....</b>	<b>156</b>
1. Ressources relatives au travail de forme du mémoire .....	156
2. Ressources relatives au travail de fond du mémoire .....	156

## **LISTE DES TABLEAUX**

---

<b>Tableau 1</b> : Évaluation financière de développement de l'application web .....	26
<b>Tableau 2</b> : Planning prévisionnel de développement de l'application .....	30
<b>Tableau 3</b> : Récapitulatif des choix méthodologiques étudiés .....	40
<b>Tableau 4</b> : Description textuelle du diagramme de contexte .....	49
<b>Tableau 5</b> : Description textuelle du diagramme de package .....	51
<b>Tableau 6</b> : Description textuelle des cas d'utilisation de l'utilisateur par défaut .....	54
<b>Tableau 7</b> : Description textuelle des cas d'utilisation d'un manager d'OSC .....	56
<b>Tableau 8</b> : Description textuelle des cas d'utilisation de l'administrateur .....	58
<b>Tableau 9</b> : Récapitulatif des architectures logicielles de déploiement .....	74
<b>Tableau 10</b> : Liste des choix technologiques - Phases d'analyse, conception, développement et tests .....	83
<b>Tableau 11</b> : Liste des choix technologiques - Phase de déploiement .....	85
<b>Tableau 12</b> : Récapitulatif des dispositions de sécurité mises en place .....	93
<b>Tableau 13</b> : Caractéristiques minimales et recommandées pour le serveur .....	105

## **LISTE DES FIGURES**

---

<b>Figure 1</b> : Logo de iPNet Experts SA .....	5
<b>Figure 2</b> : Logo de iPNet Institute of Technology .....	6
<b>Figure 3</b> : Organigramme des départements académiques et administratifs de IIT .....	10
<b>Figure 4</b> : Logo d'Expertise France .....	12
<b>Figure 5</b> : L'équipe de consultants .....	13
<b>Figure 6</b> : Logo du Mouvement SUN .....	14
<b>Figure 7</b> : Logo de l'UONGTO .....	14
<b>Figure 8</b> : Logo de l'ANT .....	14
<b>Figure 9</b> : L'assistance au premier jour de l'atelier de restitution .....	16
<b>Figure 10</b> : Illustration de la démarche méthodologique en cascade .....	33
<b>Figure 11</b> : Illustration de la méthodologie agile .....	35
<b>Figure 12</b> : Implémentation de la Méthodologie Agile - Framework SCRUM .....	37
<b>Figure 13</b> : Illustration de l'approche DevOps .....	38
<b>Figure 14</b> : Diagramme de contexte de l'application web de cartographie des OSC .....	48
<b>Figure 15</b> : Diagramme de package de l'application web de cartographie des OSC .....	50
<b>Figure 16</b> : Diagramme de cas d'utilisation de l'utilisateur par défaut .....	53
<b>Figure 17</b> : Diagramme de cas d'utilisation d'un manager d'OSC .....	55
<b>Figure 18</b> : Diagramme de cas d'utilisation de l'administrateur .....	57
<b>Figure 19</b> : Diagramme d'activités illustrant la connexion d'un utilisateur .....	59
<b>Figure 20</b> : Diagramme d'activités illustrant l'ajout d'une nouvelle osc par l'administrateur ..	60
<b>Figure 21</b> : Diagramme d'activités illustrant la mise a jour d'un projet par un manager d'osc ..	61
<b>Figure 22</b> : Diagramme d'activités illustrant la visualisation des osc par un utilisateur par défaut .....	62

<b>Figure 23</b> : Diagramme d'activités illustrant la déconnexion d'un manager d'osc .....	63
<b>Figure 24</b> : Diagramme de classes de l'application web de cartographie des OSC .....	64
<b>Figure 25</b> : Modèle Relationnel de Données .....	66
<b>Figure 26</b> : Exemple d'architecture client–serveur : deux clients font leurs requêtes à un serveur via Internet .....	68
<b>Figure 27</b> : Architecture Monolithique .....	69
<b>Figure 28</b> : Architecture N-tiers .....	70
<b>Figure 29</b> : Architecture Microservices .....	71
<b>Figure 30</b> : Architecture Serverless .....	72
<b>Figure 31</b> : Architecture Orientée Services .....	73
<b>Figure 32</b> : Architecture monolithique modulaire de l'application web de cartographie .....	78
<b>Figure 33</b> : Schéma d'illustration des architectures matérielle et logicielle .....	88
<b>Figure 34</b> : Configuration de Nginx pour l'environnement de test .....	89
<b>Figure 35</b> : Exécution dans le terminal du serveur waitress .....	90
<b>Figure 36</b> : Accès à l'application à l'adresse http://localhost/ .....	90
<b>Figure 37</b> : Configuration globale de Nginx sur le VPS .....	91
<b>Figure 38</b> : Configuration spécifique à l'application web .....	92
<b>Figure 39</b> : Accès à l'application à l'adresse https://161.97.66.8/ .....	93
<b>Figure 40</b> : Page d'accueil de l'application .....	96
<b>Figure 41</b> : Page de la carte interactive .....	97
<b>Figure 42</b> : Page du tableau de bord .....	97
<b>Figure 43</b> : Page de gestion des informations d'une OSC par son manager .....	98
<b>Figure 44</b> : Page de connexion .....	98
<b>Figure 45</b> : Définition d'un modèle Django .....	99
<b>Figure 46</b> : Définition d'une vue Django .....	100

<b>Figure 47</b> : Définition d'un formulaire Django .....	100
<b>Figure 48</b> : Définition du template base.html .....	101
<b>Figure 49</b> : Intégration d'une bibliothèque JavaScript dans un template .....	102
<b>Figure 50</b> : Règles UFW .....	108
<b>Figure 51</b> : Transfert effectué entre les deux hôtes via WinSCP .....	110
<b>Figure 52</b> : Configuration globale de Nginx sur le VPS .....	114
<b>Figure 53</b> : Configuration spécifique à l'application web .....	115
<b>Figure 54</b> : Configuration du service système cartographie_osc.service .....	116
<b>Figure 55</b> : Fonctionnement des services système .....	117
<b>Figure 56</b> : À droite, les fichiers de logs (généraux et spécifiques) du serveur web .....	118
<b>Figure 57</b> : Fichier de logs Django .....	119
<b>Figure 58</b> : Aperçu du fichier de configuration pour logrotate .....	119
<b>Figure 59</b> : Aperçu de l'activité des services .....	120
<b>Figure 60</b> : Suivi de la consommation des ressources par les processus via la commande `htop` .....	121
<b>Figure 61</b> : Suivi des ports ouverts et connexions établies avec les commandes `netstat -tulnp` et `netstat -antp` .....	122
<b>Figure 62</b> : Activation de la Protection SSH avec Fail2ban .....	122
<b>Figure 63</b> : En-tête - Apparence 1 .....	128
<b>Figure 64</b> : En-tête - Apparence 2 - Utilisateur non-connecté .....	128
<b>Figure 65</b> : En-tête - Apparence 2 - Manager d'OSC connecté .....	128
<b>Figure 66</b> : En-tête - Apparence 2 - Administrateur connecté .....	129
<b>Figure 67</b> : Pied de page - Utilisateur non-connecté .....	129
<b>Figure 68</b> : Pied de page - Manager d'OSC connecté .....	130
<b>Figure 69</b> : Pied de page - Administrateur connecté .....	130

<b>Figure 70</b> : Aperçu de la page d'accueil .....	131
<b>Figure 71</b> : Aperçu du tableau de bord des statistiques .....	132
<b>Figure 72</b> : Aperçu de la carte .....	133
<b>Figure 73</b> : Aperçu du tableau des OSC .....	133
<b>Figure 74</b> : Visualisation des détails de l'OSC CREUSET TOGO .....	134
<b>Figure 75</b> : Aperçu du tableau des projets .....	135
<b>Figure 76</b> : Visualisation des détails d'un projet .....	135
<b>Figure 77</b> : Aperçu de la médiathèque .....	136
<b>Figure 78</b> : Visualisation des détails d'un média .....	136
<b>Figure 79</b> : Espace communautaire - À propos de nous .....	137
<b>Figure 80</b> : Espace communautaire - FAQs .....	138
<b>Figure 81</b> : Espace communautaire - Nous contacter .....	138
<b>Figure 82</b> : Page de connexion .....	139
<b>Figure 83</b> : Espace de gestion du Manager d'OSC - Gestion des informations de l'OSC .....	140
<b>Figure 84</b> : Espace de gestion du Manager d'OSC - Gestion des projets de l'OSC .....	140
<b>Figure 85</b> : Test d'ajout de nouveau projet - Remplissage du formulaire .....	141
<b>Figure 86</b> : Test d'ajout de nouveau projet - Nouveau projet ajouté avec succès à la liste des projets .....	141
<b>Figure 87</b> : Test de modification de projet - Remplissage du formulaire .....	142
<b>Figure 88</b> : Test de modification de projet - Projet modifié avec succès dans la liste des projets .....	143
<b>Figure 89</b> : Test de suppression de projet - Popup de confirmation .....	143
<b>Figure 90</b> : Test de suppression de projet - Projet supprimé avec succès de la liste des projets .....	144
<b>Figure 91</b> : Espace de gestion du Manager d'OSC - Gestion des média de l'OSC .....	144
<b>Figure 92</b> : Espace de gestion du Manager d'OSC - Gestion du logo de l'OSC .....	145

<b>Figure 93</b> : Espace de gestion d'Administrateur - Accueil .....	146
<b>Figure 94</b> : Interface d'administration - Liste des Organisations .....	147
<b>Figure 95</b> : Interface d'administration - Formulaire d'ajout d'une Organisation .....	148
<b>Figure 96</b> : Interface d'administration - Ajout avec succès d'une Organisation .....	149
<b>Figure 97</b> : Interface d'administration - Formulaire de modification d'une Organisation ....	150
<b>Figure 98</b> : Interface d'administration - Modification avec succès d'une Organisation .....	151
<b>Figure 99</b> : Interface d'administration - Sélection d'Organisation(s) à supprimer .....	151
<b>Figure 100</b> : Interface d'administration - Demande de confirmation de suppression d'Organisation(s) .....	152
<b>Figure 101</b> : Interface d'administration - Suppression avec succès d'Organisation(s) .....	152

## GLOSSAIRE

---

### A

- **API (Application Programming Interface)** : Interface logicielle permettant à différentes applications de communiquer entre elles.
- **API REST (Representational State Transfer)** : Style d'architecture pour les APIs basé sur les ressources, utilisant des méthodes HTTP standard (GET, POST, etc.), et privilégiant JSON ou XML pour l'échange de données. Elle est sans état et permet une interaction intuitive avec les services web.

### B

- **Back-End** : Partie d'une application web qui gère le traitement des données et la logique métier, souvent exécutée côté serveur.
- **BDD (Base de Données)** : Ensemble structuré de données stockées et accessibles par un système de gestion de base de données (SGBD).

### C

- **Cartographie des OSC** : Processus de recensement et de représentation des Organisations de la Société Civile (OSC) sur un support visuel, généralement une carte interactive.
- **Certificat SSL (Secure Sockets Layer)** : Fichier numérique qui assure une connexion sécurisée entre un navigateur et un serveur, certifiant l'identité du site web et chiffrant les données échangées.
- **Client** : Ordinateur ou programme transmettant, par réseau, des requêtes à un serveur auquel il est connecté.

### D

- **Déploiement** : Processus de mise en ligne et de mise à disposition d'une application pour les utilisateurs finaux.
- **Django** : Framework web open-source basé sur Python permettant le développement rapide d'applications sécurisées et évolutives.

- **DRF (Django REST Framework)** : Bibliothèque permettant de construire des API REST avec Django.

## E

- **EU4SUN (European Union for Scaling Up Nutrition)** : Programme de coopération entre l'Union européenne (UE) et des régions d'Amérique latine, des Caraïbes et d'Afrique de l'Ouest, y compris le Togo. Ce projet vise à lutter contre la faim et la malnutrition en mettant en œuvre des stratégies globales et durables. Il s'inscrit dans le cadre de la stratégie SUN 3.0 (Scaling Up Nutrition), qui met l'accent sur le leadership national et le soutien aux priorités locales pour réduire les taux de malnutrition de manière systémique.
- **ER (Entité-Relation, Entity-Relationship en anglais)** : un modèle utilisé pour conceptualiser et structurer les données dans une base de données.

## F

- **Front-End** : Partie visible d'une application web avec laquelle l'utilisateur interagit, généralement développée avec HTML, CSS et JavaScript.
- **FTP (File Transfer Protocol)** : Protocole réseau utilisé pour transférer des fichiers entre un client et un serveur sur Internet ou un réseau local, sans chiffrement par défaut.

## G

- **Git** : Système de gestion de versions utilisé pour le suivi et la collaboration sur le code source.
- **GitLab** : Plateforme web de gestion de dépôts Git intégrant des fonctionnalités de collaboration et d'intégration continue.

## L

- **Leaflet.js** : Bibliothèque JavaScript permettant d'intégrer des cartes interactives dans une application web.

## M

- **Mapping** : Utilisation de cartes interactives [en ligne] pour représenter et analyser des données géographiques.

- **Mouvement SUN (Scaling Up Nutrition)** : Initiative mondiale visant à lutter contre la malnutrition sous toutes ses formes. Lancé en 2010, ce mouvement rassemble des gouvernements, des organisations internationales, des entreprises, des chercheurs et des communautés pour travailler ensemble à améliorer la nutrition, en particulier dans les pays les plus touchés.

## N

- **Nginx** : Serveur web et proxy inverse (reverse proxy) utilisé pour héberger des applications web et gérer les requêtes.

## O

- **OSC (Organisation de la Société Civile)** : Structure non gouvernementale qui œuvre pour le développement et la défense des intérêts citoyens.

## P

- **PostgreSQL** : Système de gestion de base de données relationnelle open-source.
- **POO (Programmation Orientée Objet)** : Paradigme de programmation qui organise les logiciels autour d'objets plutôt que de fonctions ou de processus. Un objet est une structure contenant à la fois des données (appelées attributs) et des fonctions qui opèrent sur ces données (appelées méthodes). Elle permet de modéliser des concepts du monde réel en logiciels.
- **Python** : Langage de programmation utilisé notamment pour le développement web et l'intelligence artificielle.

## R

- **React.js** : Bibliothèque JavaScript développée par Meta pour la création d'interfaces utilisateur dynamiques et réactives.
- **Reverse proxy** : Un reverse proxy est un serveur intermédiaire qui se trouve entre les utilisateurs (clients) et un ou plusieurs serveurs backend. Sa principale fonction est de recevoir les requêtes des clients, de les transférer au serveur approprié, puis de retourner les réponses du serveur aux clients.
- **Rôle utilisateur** : Niveau d'accès attribué aux utilisateurs d'une application web (ex. public, manager OSC, administrateur).

## S

- **Serveur** : Ordinateur ou programme fournissant des services à d'autres applications ou utilisateurs.
- **SFTP (SSH File Transfer Protocol)** : Variante sécurisée de FTP, utilisant SSH pour garantir le chiffrement des données lors du transfert de fichiers.
- **SGBD (Système de Gestion de Base de Données)** : Logiciel permettant la gestion et l'organisation des données.
- **SSH (Secure Shell)** : Protocole réseau sécurisé permettant de se connecter à distance à un serveur ou un appareil, garantissant la confidentialité et l'intégrité des données grâce à un chiffrement.
- **Système d'Exploitation** : Logiciel essentiel qui agit comme une interface entre le matériel d'un ordinateur (ou appareil) et ses utilisateurs. Il gère les ressources matérielles, exécute les applications et coordonne les tâches pour offrir une expérience utilisateur fluide. Exemples courants : Windows, macOS, Linux et Android.

## T

- **Tableau de Bord** : Interface graphique affichant des indicateurs et statistiques pour le suivi des performances d'une application.

## U

- **UML (Unified Modeling Language)** : Langage visuel utilisé pour concevoir et documenter des systèmes, en particulier dans le domaine du développement de logiciels.

## V

- **VPS (Virtual Private Server)** : Un serveur privé virtuel, est une machine virtuelle qui fonctionne comme un serveur dédié dans un environnement partagé. Cela signifie que, bien qu'on partage un serveur physique avec d'autres utilisateurs, on dispose de son propre espace virtuel, complètement isolé des autres.

## W

- **WSGI (Web Server Gateway Interface)** : Standard permettant la communication entre un serveur web et une application Python.

# **INTRODUCTION GÉNÉRALE**

Dans un contexte où l'accès à l'information est un levier essentiel pour la prise de décision, les OSC jouent un rôle clé dans la mise en œuvre des politiques publiques et des projets de développement. C'est particulièrement le cas dans le domaine de **la nutrition et de l'alimentation saine**, où de nombreuses OSC interviennent au Togo pour sensibiliser, accompagner et améliorer l'état nutritionnel des populations. Cependant, l'absence d'un outil numérique structuré pour répertorier, localiser et mettre à jour leurs informations complique leur coordination et leur suivi.

C'est dans cette perspective qu'intervient le projet EU4SUN, qui vise à renforcer l'action des OSC engagées sur ces questions. Ainsi, une **base de données Excel** a été initialement commandée pour cartographier les OSC, mais elle s'est révélée **insuffisante**. Ainsi, le besoin d'une **application web dynamique et interactive** s'est imposé afin d'optimiser l'accès aux données, d'améliorer leur mise à jour et de faciliter leur exploitation par les parties prenantes.

Face à ces enjeux, la problématique centrale de ce travail est la suivante :

**Comment concevoir et mettre en œuvre une application web, efficace pour la cartographie des OSC actives sur les questions de nutrition et d'alimentation saine au Togo, en partant d'une base de données Excel ?**

Dans cette optique, plusieurs questions guideront la suite de ce mémoire :

- **Quelles sont les limites des outils existants pour la gestion des OSC ?**
- **Quels choix technologiques et méthodologiques permettent de répondre aux besoins des utilisateurs ?**
- **Comment structurer et modéliser les données pour une meilleure exploitation ?**
- **Quels mécanismes mettre en place pour assurer la pérennité et l'évolutivité de l'application ?**

Afin d'apporter des réponses à ces interrogations, à travers ce mémoire, nous commencerons par **une analyse du cadre général du projet**, avant d'explorer **les besoins fonctionnels et techniques** qui ont guidé la conception de la solution. Ensuite, nous détaillerons **les étapes du développement et de l'implémentation** de l'application, avant de conclure sur son **déploiement, ses perspectives d'amélioration** et son impact potentiel sur les OSC et les acteurs du projet EU4SUN.

# **Chapitre 1 : PRÉSENTATION DES CADRES DE FORMATION ET DE TRAVAIL**

Ce chapitre vise à présenter le contexte global de la formation suivie à **iPNet Institute of Technology** ainsi que le cadre professionnel dans lequel s'est inscrit le projet de mon mémoire. Il est essentiel de comprendre comment la formation et l'expérience pratique acquise lors de la mission ont été interconnectées, et comment elles ont contribué à l'acquisition et à l'application des compétences nécessaires à la réalisation du projet de **Cartographie des OSC actives sur les Questions de Nutrition et d'Alimentation Saine au Togo**.

Dans un premier temps, l'accent sera mis sur le cadre de formation offert par iPNet Institute of Technology, en précisant le programme de formation, les compétences enseignées, etc. Ensuite, le chapitre présentera le cadre professionnel de ma participation au projet **EU4SUN** au Togo, en décrivant le contexte de l'engagement, les acteurs impliqués et mon rôle en tant que **Développeur Web & Consultant Junior**.

## 1.1. Cadre de formation : iPNet Institute of Technology (IIT)

### 1.1.1. Présentation

**iPNet Institute of Technology** ; *la Grande École du Génie Informatique et du Génie Électrotechnique et Énergies Renouvelables*, fait partie d'un vaste ensemble d'infrastructures académiques et technologiques issues de l'évolution de **iPNet Experts SA**, une entreprise de réputation internationale.

Depuis **2017**, **IPNET EXPERTS SA** est connu comme un *groupe* comprenant iPNet Experts SA et iPNet Institute of Technology. Cette évolution a été possible grâce à *des partenariats stratégiques avec des firmes américaines et des universités d'Amérique du Nord*.

### 1.1.2. iPNet Experts SA

Fondée en **juin 2003**, elle devient une société anonyme au **capital de 100.000.000 francs CFA** depuis le **1er janvier 2009**, et est spécialisée dans *l'intégration de solutions informatiques et la sécurité informatique*. Son siège se trouve à **Lomé, au Togo**, mais son champ d'action s'étend au **Bénin, Niger, Burkina Faso, Mali, Tchad, Gabon, RCA, Congo-Brazzaville et Guinée Conakry**.



**Figure 1 : Logo de iPNet Experts SA**

Voici quelques-unes des activités menées envers ses cibles :

➤ **Intégration de solutions :**

- ✓ Sécurité Informatique
- ✓ Communication Unifiée et Mobilité
- ✓ Vidéo Communication et Vidéo Surveillance
- ✓ Maison Numérique
- ✓ Audit de performance réseau et tests de pénétration
- ✓ Câblage réseau et interconnexion LAN/WAN
- ✓ Fourniture de solutions Réseaux et Télécoms

➤ **Audit informatique :**

- ✓ Étude de projets informatiques
- ✓ Élaboration de schémas directeurs informatiques
- ✓ Audit de sécurité des systèmes d'information et du réseau
- ✓ Conception de l'architecture réseau sécurisée

➤ **Conseil Informatique**

➤ **Formations :**

- ✓ Formation en informatique pour collégiens, lycéens, étudiants et autodidactes
- ✓ CISCO
- ✓ MICROSOFT

- ✓ ORACLE
- ✓ VMWARE
- ✓ REDHAT ENTERPRISE LINUX

Ces activités, grâce à la collaboration entre IPNET EXPERTS SA et IIT, visent à doter les étudiants de compétences pratiques et pertinentes dans un contexte d'évolution technologique rapide.

### **1.1.3. iPNet Institute of Technology (IIT)**

IIT étant le prolongement académique d'iPNet Experts SA, à ce titre, elle bénéficie d'une riche expérience et expertise dans divers domaines technologiques et académiques et offre donc des programmes de formation en **Licence, Master et Certificats Professionnels** dans le domaine de l'informatique, un secteur en constante expansion et très recherché sur le marché du travail.



**Figure 2 : Logo de iPNet Institute of Technology**

L'institut propose diverses filières académiques, notamment en :

- **Licence Professionnelle en Informatique** (initiée en **octobre 2017**) accréditée par le ministère de l'Enseignement Supérieur, avec des spécialisations telles que :
  - ✓ **Génie Logiciel**
  - ✓ **Réseaux, Systèmes et Sécurité**
  - ✓ **Cybersécurité**
  - ✓ **Développement Web, Bases de Données et Big Data**
  - ✓ **Web Design, Infographie et Multimédia**

- **Master Professionnel en Informatique** (lancé en **octobre 2018**) avec des options similaires :
  - ✓ **Génie Logiciel**
  - ✓ **Réseaux, Systèmes et Sécurité**
  - ✓ **Cybersécurité**

Avec ses programmes diversifiés et la qualité de son enseignement, IIT contribue activement à la formation de professionnels compétents prêts à relever les défis technologiques contemporains.

J'ai, ainsi, fait partie de **la promotion d'étudiants de 2020** ; soit la 4ème promotion, à iPNet. Cinquante-trois (53) UE<sup>1</sup> étalées sur six (06) semestres ont façonné ma formation, en apportant chacune des connaissances théoriques et pratiques variées.

Durant ce parcours, on pouvait distinguer cinq (05) types d'UE : **fondamentale, transversale, complémentaire, spécialité, UE libre**. Toutes étant directement ou indirectement pertinentes par rapport au profil d'informaticien que nous<sup>2</sup> étions, dès lors, entraîn de développer et à la qualité d'entrepreneur pour certains visionnaires dont je peux prétendre être le premier. Voici (sans respecter l'ordre des semestres) quelques-unes de ces UE :

#### ➤ **Fondamentale**

- ✓ **CSC 241** : Algorithme et Programmation Structurée avec Python I
- ✓ **CSC 242** : Algorithme et Programmation Structurée avec Python II
- ✓ **IT 130** : Introduction aux technologies Web (HTML, CSS, JavaScript)
- ✓ **IT 240** : Introduction aux Bases de Données UML/SQL
- ✓ **NET 311** : Architecture des Ordinateurs & Systèmes d'Exploitation
- ✓ **IT 373** : System Concepts
- ✓ **CSC 373** : Programmation Systèmes I
- ✓ **CSC 374** : Programmation Systèmes II

---

<sup>1</sup> UE : Unité(s) d'Enseignement.

<sup>2</sup> Nous ; les étudiants de cette promotion.

- ✓ **MAT 140** : Logique & Mathématiques Discrètes I
- ✓ **MAT 141** : Logique & Mathématiques Discrètes II
- ✓ **IT 223** : Probabilités & Statistiques Descriptives
- ✓ **MAT 150** : Calculus I
- ✓ **MAT 151** : Calculus II
- ✓ **CAPM** : Formation & Certification CAPM (Management de Projets)
- ✓ **PPE-300** : Projet Professionnel Étudiant I
- ✓ **PPE-301** : Projet Professionnel Étudiant II
- ✓ **PPE-302** : Projet Professionnel Étudiant III

➤ **Transversale**

- ✓ **MGT 101** : Anglais Technique I
- ✓ **MGT 110** : Anglais Technique II
- ✓ **WRD 103** : Techniques d'expressions écrites et orales

➤ **Complémentaire**

- ✓ **ACC 101** : Comptabilité Générale
- ✓ **MGT 103** : Droit Général

➤ **Spécialité**

- ✓ **IT 211** : Introduction au Développement des Applications/Python
- ✓ **CSC 299** : Lab en Informatique Appliquée (Langages, API, etc.)
- ✓ **CSC 355** : DataBase Systems
- ✓ **CSC 321** : Conception et Analyse des Algorithmes
- ✓ **CSC 308** : Frameworks pour le Développement d'Application Web
- ✓ **SE 325** : Introduction au Génie Logiciel
- ✓ **CSC 394** : Management des Projets d'Ingénierie Logicielle

➤ **UE libre**

- ✓ **MGT 201** : Développement Personnel
- ✓ **MGT 301** : Entrepreneuriat
- ✓ **MGT 411** : Soft Skills, Leadership and Mindset

Ce programme n'est pas le fruit d'un hasard mais d'une réflexion commune et coordonnée des membres du personnel éducatif et administratif repartis en différents départements et cellules bien structurés. Chaque unité est dotée de spécialistes et de personnels académiques qualifiés assurant la coordination des programmes et la gestion des activités académiques.

Cette *organisation structurée*<sup>1</sup> assure un encadrement optimal des étudiants et garantit la bonne marche des activités académiques et administratives.

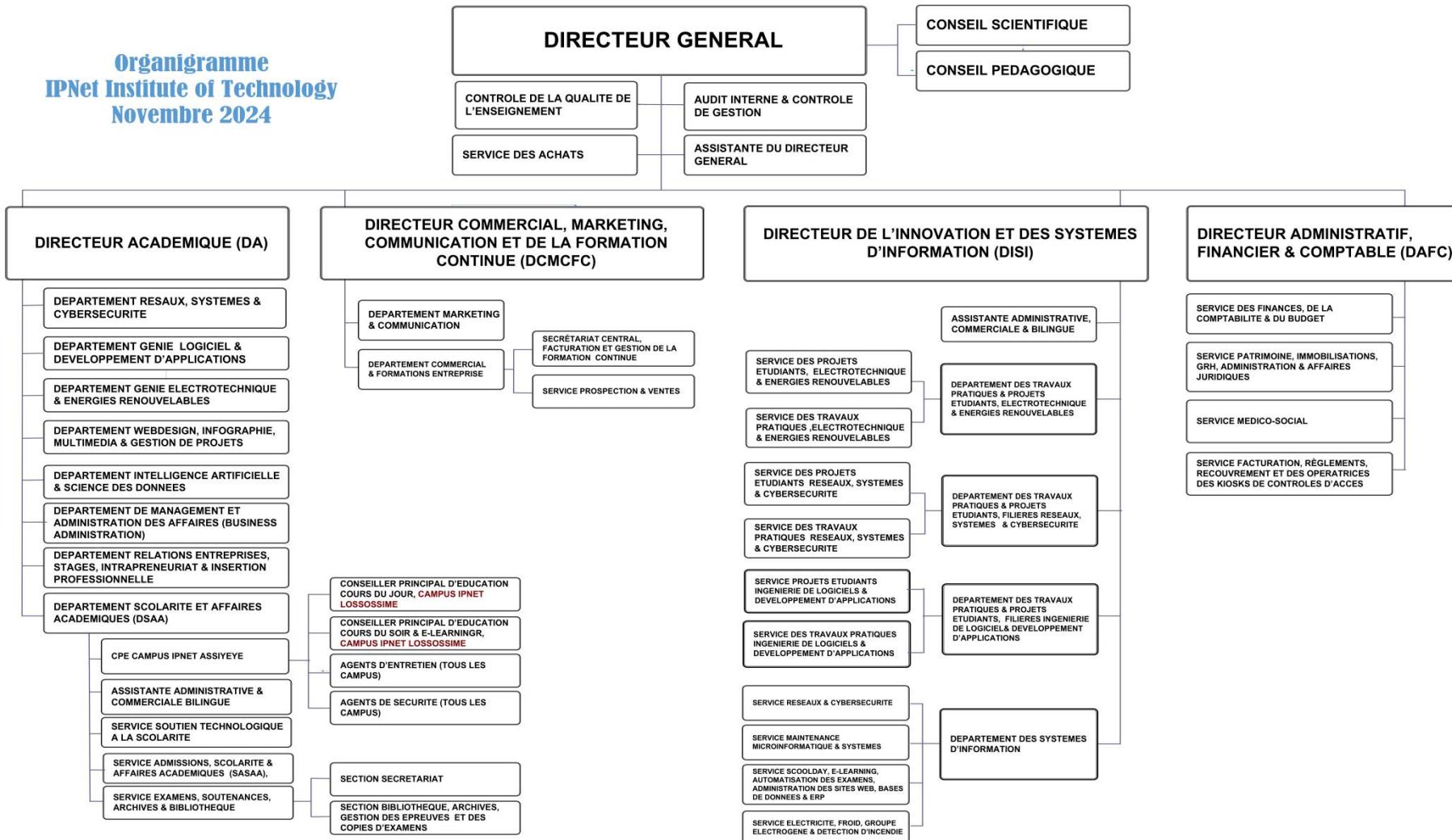
IIT offre donc un cadre unique sans commune mesure<sup>2</sup> et complet pour la formation des professionnels dans le domaine de l'informatique et de la sécurité des systèmes d'information. Ainsi, grâce à cette formation rigoureuse, j'ai pu mettre en pratique les compétences acquises dans le cadre de mon engagement professionnel dans le projet EU4SUN, décrit dans la section suivante.

---

<sup>1</sup> Voir sur la page suivante, l'organigramme permettant de visualiser ladite organisation.

<sup>2</sup> Ceci s'applique dans le contexte géographique concurrentiel régional d'iPNet et précisément au moment où j'écris ce mémoire.

**Organigramme  
IPNet Institute of Technology  
Novembre 2024**



**Figure 3 : Organigramme des départements académiques et administratifs de IIT**

## **1.2. Cadre professionnel : Projet EU4SUN - Développeur Web & Consultant Junior**

L'expérience professionnelle dont je m'en vais présenter le cadre, s'inscrit dans le contexte de l'exécution du projet EU4SUN au Togo. Il s'agit d'un projet de trois ans (2023-2025), financé par la **Commission Européenne**, mené conjointement par la **FIIAPP<sup>1</sup>** et **Expertise France**, de soutien à l'opérationnalisation de la **Stratégie SUN 3.0** (2021-2025) du **Mouvement SUN<sup>2</sup>**.

Ledit projet vise quatorze (14) pays dans deux zones géographiques :

- ✓ **l'Amérique Latine et les Caraïbes** (coordination réalisée par la **FIIAPP**)
- ✓ **l'Afrique de l'Ouest** (coordination réalisée par **Expertise France**) ; où le projet accorde une attention particulière aux pays suivants : **Bénin, Burkina Faso, Côte d'Ivoire, Guinée-Bissau, Niger, Togo et Sénégal.**

L'**objectif principal** du projet est de *contribuer à l'augmentation et à l'amélioration des interventions en matière de nutrition dans les pays du Mouvement SUN*.

Parler donc du projet EU4SUN au Togo n'est donc pas une chose anodine. Face au défis de la nutrition au Togo, notamment l'absence d'une cartographie qui regroupe les Organisations de la Société Civile (OSC) en Nutrition et Alimentation Saine (pour apprécier leurs déploiements sur l'étendue du pays, synchroniser leurs actions, faire un suivi de leurs projets et leurs permettre de bénéficier de financements en toute transparence) ; l'amélioration de leur visibilité et la création d'un réseau global s'est donc imposée. L'**Application Web de Cartographie des OSC Actives sur les Questions de Nutrition et d'Alimentation Saine au Togo** s'est donc imposée comme solution à cette difficulté.

À cet effet, **Expertise France** a recruté une équipe de consultants initialement composée de deux membres : **M. Yao Anani Paul-Martial AGBOBLI**, chef de mission, et **M. Yonam Eliud ABOGAN**, consultant associé. Par la suite, cette équipe a été renforcée par mon recrutement en tant que **Développeur Web & Consultant Junior**, un recrutement effectué par le chef de mission.

---

<sup>1</sup> FIIAPP : Fondation Internationale et Ibéro - Américaine pour l'Administration et les Politiques Publiques.

<sup>2</sup> SUN : Scaling Up Nutrition.

**Expertise France** est une agence française d’assistance technique internationale qui soutient le développement durable et contribue à la mise en place de projets innovants dans divers secteurs, notamment la santé, l’éducation et la gouvernance.



Figure 4 : Logo d’Expertise France

**M. Yao Anani Paul-Martial AGBOBLI** est juriste, économiste, gestionnaire et formateur, avec plus de **25 années d’expérience professionnelle internationale en consultance**. Il a collaboré avec des organisations prestigieuses telles que l'**International Trade Centre** (en tant que **Consultant associé**), **Secours Catholique-Caritas France** (où il était **Consultant en renforcement de capacités**), **Caritas Africa** (il y était **Accompagnateur en renforcement de capacités des Caritas**), **Solthis** (étant **Consultant principal**), et bien d’autres. Il est actuellement **Enseignant** de deux (02) unités d’enseignement à l'**Université de Kara** sur l'**Entrepreneuriat et l’Insertion Professionnelle des Jeunes**, et **Consultant Senior** sur le projet **EU4SUN au Togo**. Sur cette mission, il a dirigé la collecte des données auprès des OSC et a été responsable de la supervision globale du projet (du début jusqu’à la restitution en passant par la phase de développement), assurant une bonne dynamique de travail et partageant régulièrement des leçons de consultance pour notre développement professionnel. Il a également présidé les réunions de présentation des livrables avec les commanditaires.

**M. Yonam Eliud AGBOGAN**, **ingénieur cloud, développeur web et analyste projet**, était mon supérieur direct au sein de l’équipe. Il était responsable de **la mise en place de la base de données Excel de cartographie des OSC**, qui regroupait les données collectées sur **74 organisations** œuvrant dans le domaine de la nutrition et de l’alimentation saine à travers tout le territoire togolais, de la région Maritime à la région des Savanes ; celle-ci étant le livrable initialement commandé par les commanditaires et attendu par les bénéficiaires, avant qu’ils ne

valident la proposition complémentaire concernant l'application web. Il assurait la supervision et le suivi de mon travail tout au long de la mission ; car, je devais, en effet, fournir une application web avec des données initiales identiques à celles de la base de données Excel.



**Figure 5 : L'équipe de consultants<sup>1</sup>**

Les échanges professionnels concernant l'avancée de l'élaboration des livrables se faisaient régulièrement, à raison de réunions hebdomadaires entre les consultants et mensuelles avec les commanditaires et bénéficiaires. Ces réunions en ligne impliquaient la présence des consultants, du représentant d'Expertise France, de la représentante du Mouvement SUN, des représentants des faîtières d'OSC, notamment l'**UONGTO<sup>2</sup>** et l'**ANT<sup>3</sup>**, ainsi que celui de la **Division Nutrition du Ministère de la Santé**.

<sup>1</sup> De gauche à droite : Je, Yao David SOUSSOUKPO, Développeur Web & Consultant junior ; M. Yao Anani Paul-Martial AGBOBLI, Juriste Économiste-Gestionnaire & Consultant en chef; M. Yonam Eliud AGBOGAN, Ingénieur Cloud & Consultant associé.

<sup>2</sup> UONGTO : Union des ONG du Togo.

<sup>3</sup> ANT : Alliance pour la Nutrition au Togo.



**Figure 6 : Logo du Mouvement SUN**



**Figure 7 : Logo de l'UONGTO**



**Figure 8 : Logo de l'ANT**

Un atelier de restitution<sup>1</sup> a eu lieu du **28 au 29 Octobre 2024**, à l'hôtel Novela Star. Cet atelier visait à rassembler les représentants des OSC bénéficiaires du projet, le représentant de la Division Nutrition du Ministère de la Santé, les représentants d'Expertise France, SUN, et des faîtières d'OSC autour de la même thématique de **Nutrition et Alimentation Saine au Togo**. L'objectif était de présenter le travail accompli afin que ces derniers puissent

<sup>1</sup> Réunion ou événement organisé pour présenter et partager les résultats ou les conclusions d'un travail, d'un projet, ou d'une étude avec un public cible. Une photo d'ensemble des membres de l'atelier est disponible après la conclusion de ce chapitre.

l'apprécier et prendre en main la première version de l'application web. C'était un atelier qui a mis à l'épreuve non seulement notre capacité à rendre compte d'une mission et restituer les résultats qui en sortent, mais aussi à encadrer et former un public composé de hauts responsables<sup>1</sup>. J'étais personnellement chargé de présenter l'application web et de former l'assistance pour la prise en main de celle-ci. Le chef de mission, était, durant les deux (02) jours, le modérateur principal de cet atelier et nous n'avons guère manqué d'apprendre de son savoir et de son savoir-faire.

L'ambiance de l'atelier était très positive, avec une grande appréciation du travail accompli. Les participants ont exprimé leur satisfaction face à la première version de l'application web, soulignant son utilité et sa pertinence pour la mise en place d'une **Société Civile Unifiée**<sup>2</sup>. Toutefois, après avoir pris connaissance des défis rencontrés lors du développement, de nouvelles perspectives d'amélioration ont émergé. Les échanges ont permis de mettre en lumière certaines limites de l'outil, et plusieurs suggestions ont été formulées pour affiner et adapter l'application aux réalités du terrain, ouvrant ainsi la voie à de futures évolutions.

## Conclusion

Se former à **iPNet** m'a permis d'acquérir des compétences solides tant sur le plan technique que méthodologique, essentielles dans le cadre de projets d'envergure comme celui du projet **EU4SUN**. Mon implication dans ce projet m'a offert l'opportunité d'appliquer et de valoriser ces acquis, notamment dans la gestion technique d'une application web et la collaboration avec diverses parties prenantes.

Cette expérience a mis en lumière des **compétences techniques** telles que le développement web, la mise en production et la rédaction documentaire, mais également des **compétences transversales** indispensables : écoute active, communication continue, gestion du temps, gestion du stress, discipline et capacité d'adaptation.

Le chapitre suivant présentera **l'étude préalable du projet**, avec une analyse approfondie du contexte, de la problématique, des objectifs visés ainsi que des aspects techniques liés à la réalisation de l'application.

---

<sup>1</sup> Tels que des directeurs et chefs exécutifs. Il était essentiel de prendre en compte leur statut ainsi que les dynamiques propres aux personnes occupant des postes de responsabilité.

<sup>2</sup> Travaillant en Nutrition et Alimentation Saine.



**Figure 9 : L'assistance au premier jour de l'atelier de restitution**

# **Chapitre 2 : ÉTUDE PRÉALABLE DU PROJET**

Ce chapitre propose une analyse préliminaire du projet de **Cartographie des OSC de Nutrition et d'Alimentation Saine au Togo**, en mettant en exergue son contexte, ses objectifs et ses enjeux. L'étude préalable est une étape cruciale pour comprendre les bases de la réalisation du projet, car elle permet de cerner les besoins, d'évaluer les problématiques existantes et d'établir une feuille de route pour la conception de l'application web.

La section commencera par présenter **le contexte du projet**, les raisons pour lesquelles il a été initié, et les attentes des parties prenantes, en particulier celles des OSC impliquées dans la promotion de la nutrition et de l'alimentation saine. Nous aborderons ensuite **la problématique** soulevée par la gestion traditionnelle des données et la justification de la nécessité d'une application web. Ensuite, seront détaillés **les objectifs spécifiques** du projet et **l'évaluation financière** prévus pour la réalisation de l'application.

Enfin, les choix méthodologiques retenus pour le développement sont expliqués, en comparant les approches classiques telles que **le modèle en cascade** avec des méthodologies plus **agiles** comme **Scrum** ou **DevOps**. Nous justifions ainsi notre décision d'adopter **une méthodologie Hybride**, adaptée à la nature évolutive du projet et aux attentes des parties prenantes.

## 2.1. Présentation du projet

L'application Web de Cartographie des OSC Actives sur les Questions de Nutrition et d'Alimentation Saine au Togo, s'inscrit dans le contexte précis de l'exécution de la **Feuille de Route pour la Mise en Place d'une Société Civile Unifiée au Togo**, travaillant pour la Nutrition et l'Alimentation Saine. En effet, afin que la société civile participe activement aux initiatives en cours en matière de nutrition, il a été nécessaire de restructurer et de redynamiser ses acteurs sur la thématique nutrition au Togo. Pour cela, une nouvelle feuille de route avait été définie avec l'accompagnement du Mouvement SUN et d'Expertise France en juin 2023. L'objectif de cette feuille de route est de consolider davantage la coordination unifiée de la société civile. La mise en œuvre de cette feuille de route requérerait un appui technique notamment pour la réalisation de la cartographie des OSC qui allait permettre d'apprécier la couverture des interventions de ces OSC en matière d'offre de services relatives à la nutrition et d'alimentation saine au Togo. L'élaboration de cette cartographie a fait l'objet d'une requête d'appui technique adressée à Expertise France par la société civile togolaise.

Dans un premier temps, il était prévu d'élaborer une cartographie des OSC à l'aide d'une base de données Excel. Cependant, après avoir analysé les dimensions de la problématique, *l'équipe initiale de consultants*<sup>1</sup>, a proposé une solution supplémentaire : au lieu de se limiter à une simple BDD Excel, ils ont suggéré le développement d'une application web. Cette proposition a été validée.

Le développement de l'application web a donc, dans un deuxième temps, succédé à la création de la BDD Excel, qui avait été constituée à partir des données collectées auprès des OSC. Ces données ont été soigneusement nettoyées puis structurées, permettant ainsi de lancer le développement de l'application web une fois la BDD prête.

## 2.2. Problématique

Le besoin d'une « application web » de cartographie des OSC travaillant sur la nutrition et l'alimentation saine au Togo découle d'une double problématique : d'une part, les défis liés à l'absence de coordination et de visibilité des OSC dans ce domaine, et d'autre part, les limites potentielles de l'outil initial envisagé<sup>2</sup> pour répondre aux ambitions du projet.

### 2.2.1. Une absence de cartographie et de coordination entre les OSC

Malgré leur contribution régulière aux efforts de la plateforme multisectorielle pour le renforcement de la nutrition, les OSC togolaises opèrent souvent de manière individuelle et dispersée. Les lacunes suivantes ont été identifiées :

- **Manque de visibilité des acteurs** : Il n'existe pas de cartographie claire permettant d'identifier les OSC intervenant sur les thématiques de nutrition et d'alimentation saine. Cela limite la compréhension de leur répartition géographique et thématique.
- **Faible collaboration entre les OSC** : Les interactions entre les organisations travaillant sur des interventions sensibles et spécifiques à la nutrition restent rares, ce qui freine l'efficacité collective.

---

<sup>1</sup> Elle était constituée de Monsieur Y. A. P-M. AGBOBLI et Monsieur Y. E. AGBOGAN.

<sup>2</sup> La BDD Excel initialement commandée.

- **Absence de gouvernance structurée** : Aucun mécanisme formel ne permet d'assurer une coordination efficace ou une gouvernance commune entre ces acteurs.

Ces constats ont motivé la société civile togolaise à solliciter un appui technique pour créer une cartographie initiale, qui a pris la forme d'une BDD Excel.

## 2.2.2. Les limites d'une base de données Excel comme solution unique

Bien qu'une BDD Excel puisse constituer une première étape pour répondre à ce besoin, elle présenterait des limites majeures si elle était retenue comme seule solution :

- **Suivi des modifications et des versions<sup>1</sup>** : Dans un fichier Excel, il peut être difficile de suivre les modifications ou d'assurer un contrôle de version efficace. Plusieurs utilisateurs travaillant sur le même fichier peuvent rendre le suivi des changements laborieux et augmenter les risques d'incohérences et d'erreurs dans les données.
- **Accès restreint** : L'utilisation d'un fichier statique limiterait la consultation simultanée et compliquerait la collaboration entre les acteurs. Cela entraînerait des délais supplémentaires et des difficultés pour gérer les informations en temps réel.
- **Manque d'interactivité** : Les analyses dynamiques et visuelles (cartes, graphiques, rapports) seraient limitées ou impossibles avec un outil Excel. Cette absence d'interactivité réduirait l'efficacité de l'analyse et de la présentation des données.
- **Sécurité des données** : La gestion de la sécurité des données serait insuffisante avec un fichier Excel. L'absence de protocoles de sécurité robustes pourrait rendre les informations sensibles vulnérables à des accès non autorisés.
- **Scalabilité** : Au fur et à mesure de l'augmentation du volume des données, Excel deviendrait difficile à gérer, avec des ralentissements de performance. Cette limitation affecterait la capacité du système à évoluer avec le projet.
- **Automatisation limitée** : Les possibilités d'automatisation des processus (génération de rapports, notifications automatiques) seraient réduites dans un fichier Excel. Cela entraînerait une charge de travail manuelle élevée et un risque d'erreurs humaines.

---

<sup>1</sup> L'anglicisme « versioning » est généralement employé pour parler du suivi des versions [du code source]. En français, ça se dit « versionnage ».

- **Dépendance à une seule personne** : Si une seule personne est responsable de la mise à jour du fichier Excel, cela crée une dépendance excessive. En cas d'absence ou de départ, cela compliquerait la gestion des données et risquerait de ralentir le projet.
- **Compatibilité et portabilité** : Bien qu'Excel soit largement utilisé, il peut y avoir des problèmes de compatibilité sur différentes plateformes. De plus, l'accès aux données est limité par la nécessité d'avoir le fichier localement, contrairement à une solution web plus flexible.

Ces limites potentielles, identifiées dès la phase de réflexion, ont conduit les consultants<sup>1</sup> à proposer une solution complémentaire<sup>2</sup> : le développement d'une application web. Cette plateforme irait au-delà des capacités de la BDD Excel, offrant une meilleure interactivité, une gestion centralisée des données et des fonctionnalités avancées, tout en répondant aux ambitions stratégiques du projet.

### 2.3. Étude de l'existant

Actuellement, la société civile togolaise impliquée dans les thématiques de nutrition et d'alimentation saine fonctionne de manière disparate. Les OSC travaillent généralement de manière isolée, sans coordination ni partage structuré des informations. Les rares initiatives existantes de collecte de données sont réalisées ponctuellement, souvent dans un cadre projet ou pour répondre à des besoins spécifiques.

Les outils utilisés pour ces collectes de données sont majoritairement rudimentaires, tels que des formulaires papier ou des fichiers Excel non centralisés. De plus, aucun mécanisme institutionnel ne garantit l'actualisation régulière des données collectées. Par conséquent, il est difficile d'obtenir une vue d'ensemble actualisée de la couverture des OSC et de leurs interventions dans le domaine de la nutrition au Togo.

La **Task Force**<sup>3</sup> joue un rôle clé dans la coordination des efforts, mais elle ne dispose pas de données consolidées et actualisées sur les OSC. Ainsi, bien qu'elle facilite les discussions

<sup>1</sup> Bien entendu, l'équipe initiale des consultants.

<sup>2</sup> Dans ce sens où, elle ne vient pas supprimer la commande de la BDD Excel, mais elle vient la compléter. Désormais, deux (02) livrables étaient essentiellement attendus.

<sup>3</sup> La plateforme multisectorielle de coordination de la Nutrition.

stratégiques, son efficacité est limitée par l'absence d'une base d'information fiable et dynamique. Cela crée un besoin criant de solutions techniques pour pallier ces insuffisances.

## 2.4. Critique de l'existant

L'analyse de l'existant révèle plusieurs insuffisances majeures :

- **Fragmentation des données** : Les informations sur les OSC sont dispersées entre différents acteurs et outils, ce qui complique leur gestion et leur exploitation.
- **Manque de mise à jour** : Les outils comme Excel, souvent utilisés de manière non collaborative, ne permettent pas une actualisation simultanée des données par plusieurs utilisateurs.
- **Absence de centralisation** : Aucun outil unique ne regroupe les données sur les OSC, ce qui entraîne des doublons et des incohérences dans les informations disponibles.
- **Limitation dans la prise de décision** : L'indisponibilité de données fiables et actualisées affecte la capacité des parties prenantes à prendre des décisions basées sur des analyses complètes et pertinentes, pour assurer un déploiement et une synchronisation uniforme des OSC sur le territoire.

Ces insuffisances rendent difficile la coordination entre les OSC et limitent leur impact collectif dans le domaine de la nutrition et de l'alimentation saine.

## 2.5. Objectifs du projet et résultats attendus

**L'objectif général** de l'application web est de permettre de cartographier efficacement les OSC travaillant sur les thématiques de nutrition et d'alimentation saine au Togo, afin de renforcer la coordination et d'améliorer la prise de décision stratégique.

Au delà de ça, elle vise à satisfaire certains **objectifs spécifiques** que sont :

- Centraliser les données des OSC en une plateforme accessible et dynamique,
- Permettre la visualisation géographique des OSC et de leurs interventions,
- Faciliter la collaboration entre les OSC et les autres parties prenantes,
- Fournir des outils d'analyse pour évaluer la couverture des interventions des OSC,

- Mettre en place un système évolutif et durable, adapté aux besoins futurs de gestion des données.

L'atteinte de ces objectifs permettrait de surmonter les défis existants et d'améliorer la coordination des OSC travaillant sur la *nutrition et alimentation saine au Togo*<sup>1</sup>. À cet effet, quelques résultats concrets sont attendus.

**Le résultat principal** fait objet de la formulation suivante : La mise à disposition d'une plateforme web opérationnelle et accessible, facilitant la gestion et la cartographie des OSC (<https://oscnutritionsainetogo.com>). Et **les résultats spécifiques** qui le composent sont :

- Une base de données relationnelle robuste, garantissant l'intégrité et l'accessibilité des informations des OSC,
- Une interface utilisateur intuitive et fluide, permettant l'affichage des données sur une carte dynamique et la consultation des fiches détaillées des OSC,
- Des fonctionnalités avancées de recherche et de filtrage, facilitant l'extraction d'informations pertinentes,
- Un système d'administration sécurisé, doté de rôles et permissions définis, assurant une gestion optimisée des accès et des modifications.

Avec ces résultats, la plateforme deviendrait un outil essentiel pour renforcer la coordination et optimiser la prise de décision stratégique dans le domaine de la nutrition et de l'alimentation saine au Togo.

## 2.6. Proposition de solutions

En réponse aux réels besoins des commanditaires et bénéficiaires, dans le contexte de la coordination des OSC, l'application web proposée devra répondre aux spécifications suivantes.

---

<sup>1</sup> Désormais, il faudra retenir que les OSC, dans le contexte de ce projet et donc de ce mémoire, sont uniquement celles œuvrant dans les thématiques de Nutrition et Alimentation Saine au Togo. Il sera mentionné « OSC » dans le reste du document, sauf en cas d'une situation requérant la terminologie complète.

## **2.6.1. Spécifications fonctionnelles**

Il s'agit des fonctionnalités techniques tangibles offertes par l'application web.

➤ **Gestion des utilisateurs**

- ✓ Création, modification et suppression d'utilisateurs
- ✓ Attribution de rôles et de permissions
- ✓ Authentification et sécurité

➤ **Gestion des organisations**

- ✓ Ajout, modification et suppression d'organisations
- ✓ Importation et exportation de données d'organisations
- ✓ Recherche et filtrage d'organisations
- ✓ Consultation des informations d'une organisation (nom, adresse, contact, etc.)
- ✓ Ajout de notes et de commentaires

➤ **Gestion des projets**

- ✓ Ajout, modification et suppression de projets
- ✓ Importation et exportation de données de projets
- ✓ Recherche et filtrage de projets
- ✓ Consultation des informations d'un projet (nom, description, objectifs, etc.)
- ✓ Ajout de notes et de commentaires

➤ **Visualisation et Exportation des données**

- ✓ Représentations graphiques des données
- ✓ Visualisation de statistiques sur les organisations et les projets
- ✓ Exportation des données en format XLS

➤ **Cartographie**

- ✓ Affichage d'une carte interactive avec les organisations géolocalisées
- ✓ Filtrage des organisations par zone géographique

- ✓ Ajout de points d'intérêt et de couches de données
- ✓ Exportation de la carte

➤ **Fonctionnalités collaboratives**

- ✓ Partage de documents et de notes
- **Aide et documentation**
- ✓ Aide en ligne et documentation utilisateur
  - ✓ Tutoriels et guides d'utilisation

Ces fonctionnalités permettront des manipulations plus ou moins avancées selon le type d'utilisateur connecté. Voici la liste des types d'utilisateurs ou rôles :

- ✓ **Administrateur** : Il aura la main mise sur le système dans son entièreté. Mais ses actions seront dirigées par les propriétaires du système.
- ✓ **Manager d'OSC** : Chaque OSC, dispose d'un manager. Ce dernier a la possibilité de gérer toutes les informations concernant son OSC, tout en restant dans le périmètre d'actions défini.
- ✓ **Utilisateur par défaut** : Il s'agit de tout utilisateur non authentifié, il a un accès uniquement en lecture sur les données que présente le système.

Au delà de ces fonctionnalités, l'application devra présenter certaines dispositions non-négligeables.

## 2.6.2. Spécifications non-fonctionnelles

Par spécifications non-fonctionnelles, on sous-entend fonctionnalités transversales. C'est un ensemble de prédispositions que doit garantir l'application pour assurer une bonne expérience utilisateur.

On distingue :

- ✓ **Scalabilité** : L'application devra être conçue de manière à pouvoir évoluer en fonction de la croissance du nombre d'utilisateurs, d'organisations et de projets.

- ✓ **Évolutivité** : L'application devra être construite sur une architecture modulaire permettant d'ajouter de nouvelles fonctionnalités à l'avenir sans réécrire le code de base.
- ✓ **Responsivité** : Elle devra offrir une expérience optimale sur tout type de dispositif, qu'il s'agisse d'un ordinateur de bureau, d'une tablette ou d'un smartphone.
- ✓ **Accessibilité** : Elle devra être accessible à tous les utilisateurs, y compris ceux en situation de handicap, conformément aux normes d'accessibilité WCAG<sup>1</sup>.
- ✓ **Rapidité** : L'application doit être conçue pour garantir une rapidité de traitement et de réponse optimale, même avec un grand nombre de données ou d'utilisateurs simultanés.

Ces spécifications non-fonctionnelles garantiront non seulement la stabilité, la performance et l'évolutivité de l'application, mais aussi une expérience utilisateur fluide et inclusive. En plus des fonctionnalités décrites dans la sous-section précédente, ces critères permettront de répondre aux besoins des commanditaires et des bénéficiaires tout en assurant un usage à long terme et une adaptabilité face aux changements technologiques et aux évolutions des besoins.

## 2.7. Évaluation financière du projet

L'évaluation financière d'un projet est une étape essentielle permettant d'analyser les ressources économiques mobilisées ainsi que les coûts associés à son développement et à sa mise en œuvre. Dans le cadre du projet EU4SUN, et plus particulièrement de l'application web de cartographie des OSC, l'estimation financière repose sur plusieurs postes de dépenses, incluant les coûts techniques, humains et infrastructurels.

L'objectif de cette section est de détailler **les principaux éléments budgétaires** nécessaires au bon déroulement du projet, en prenant en compte :

**Tableau 1 : Évaluation financière de développement de l'application web**

---

<sup>1</sup> WCAG : Web Content Accessibility Guidelines. Ce sont des directives établies par le W3C pour rendre les contenus web accessibles à tous, y compris aux personnes en situation de handicap. Elles couvrent plusieurs aspects de l'accessibilité, comme la **perception**, l'**utilisabilité**, et la **compréhension des informations**. Ces directives sont essentielles pour garantir une expérience utilisateur inclusive et sont divisées en niveaux de conformité : **A**, **AA** et **AAA**, afin de répondre aux besoins de diversité des utilisateurs, y compris ceux ayant des déficiences visuelles, auditives ou motrices.

<b>Poste de dépenses</b>	<b>Description</b>	<b>Coût estimé</b>
<b>Développement de l'application</b>	Honoraires pour la conception et le développement.	<b>800 000 XOF</b>
<b>Hébergement initial, infrastructure, sécurité et sauvegardes</b>	Gestion d'un VPS, gestion des serveurs web et d'application, nom de domaine et certificat SSL, pare-feu et solutions de sauvegarde.	<b>100 000 XOF</b>
<b>Maintenance et mises à jour</b>	Support technique, corrections de bugs et évolutions de l'application.	<b>50 000 XOF / an</b>
<b>Formation des utilisateurs</b>	Sessions de formation pour les administrateurs, managers et utilisateurs.	<b>100 000 XOF</b>
<b>Documentation et support</b>	Rédaction des manuels d'utilisation et guides techniques.	<b>50 000 XOF</b>
<b>Total estimé</b>	Coût global du projet pour la première année.	<b>1 100 000 XOF</b>

Cette estimation prend en compte les coûts de mise en place ainsi que les frais de maintenance pour assurer la durabilité du projet. Elle vise à garantir que les ressources disponibles permettent un déploiement efficace et une exploitation pérenne de l'application.

## 2.8. Planning prévisionnel de réalisation du projet

Le projet a été structuré en **4 phases** principales, définies dans le cadre du planning initial proposé dans les **TDR<sup>1</sup>**. Toutefois, une révision du phasage a été effectuée après la prise en main de la mission par l'équipe de consultants. Cette révision a été réalisée lors de la phase de **cadrage de la mission et validation de la méthodologie**, en conformité avec le premier phasage.

### 2.8.1. Phasage initial proposé dans les TDR

Le phasage initial proposé dans les TDR comprenait **4 phases** principales, réparties comme suit :

- **Phase 1** : Cadrage de la mission et validation de la méthodologie.
- **Phase 2** : Développement de la cartographie Excel : collecte, analyse et structuration des données.
- **Phase 3** : Conception et mise en place de l'interface web pour exploiter la base de données Excel.
- **Phase 4** : Rédaction et valorisation des livrables finaux.

La durée estimée pour l'exécution des trois premières phases était de **40 jours d'activités**, répartis sur la période de **mars à juillet**. Les livrables attendus à ces phases (notamment la base de données Excel et d'autres documents) devaient permettre de lancer la **4e phase** dédiée au développement de l'application web, qui devait s'étaler sur **27 jours**, entre **juillet et décembre**.

### 2.8.2. Révision du phasage après le cadrage

Après l'arrivée de l'équipe de consultants, et conformément à la phase de **cadrage de la mission et validation de la méthodologie** (Phase 1), une **note de cadrage méthodologique** a été remise. Celle-ci a permis de réajuster le planning initial afin de mieux répondre aux

---

<sup>1</sup> TDR : Termes de Référence. Ils contenaient déjà les spécifications techniques attendues pour le système. Ces spécifications ont servi de base pour guider le développement de l'application web, en assurant que chaque fonctionnalité développée réponde aux exigences définies.

besoins du projet. Après validation de cette nouvelle proposition par le **comité de pilotage**, un nouveau phasage en **5 phases** a été retenu, structuré de la manière suivante :

- **Phase 1** : Cadrage de la mission et validation de la méthodologie.
- **Phase 2** : Développement de la cartographie Excel : collecte, analyse et structuration des données.
- **Phase 3** : Développement de l'application web.
- **Phase 4** : Rédaction et valorisation des livrables finaux.
- **Phase 5** : Suivi, maintenance et mise à jour de l'application.

Cette révision a permis d'ajouter une phase de suivi post-livraison.

### 2.8.3. Retards et défis rencontrés

Le développement de la **cartographie Excel** (Phase 2) était initialement prévu pour la période du **03 juin au 06 juillet 2024**. Cependant, des défis liés à la collecte des données sur le terrain ont entraîné des **retards significatifs**. Ces imprévus ont impacté la **phase de développement de l'application web** (Phase 3), qui a donc débuté en **juillet 2024** au lieu de juin et a été prolongée jusqu'en **octobre 2024**.

Un **atelier de restitution des livrables** a eu lieu en **octobre 2024**, conformément aux spécifications des **Termes de Référence**, pour présenter les résultats obtenus et valider l'application web développée avec les commanditaires.

### 2.8.4. Planning de développement de l'application

Le développement de l'application web, prévu entre juillet et octobre 2024, a suivi une approche progressive et itérative, avec un développement en chaîne des fonctionnalités attendues. Chaque fonctionnalité a été implémentée, testée et validée avant de passer à la suivante, assurant ainsi un suivi rigoureux de l'avancement et la mise en place d'une version fonctionnelle et stable à chaque étape. Tout cela conformément au planning prévisionnel suivant :

**Tableau 2 : Planning prévisionnel de développement de l'application**

Fonctionnalité	Période de développement	Détails
<b>Gestion des utilisateurs</b>	Juillet : Semaine 1 - 2	<ul style="list-style-type: none"> <li>• Création, modification, suppression des utilisateurs</li> <li>• Attribution de rôles et de permissions</li> <li>• Authentification et sécurité</li> </ul>
<b>Gestion des organisations</b>	Juillet : Semaine 3 - 4	<ul style="list-style-type: none"> <li>• Ajout, modification, suppression d'organisations</li> <li>• Importation et exportation de données d'organisations</li> <li>• Recherche et filtrage d'organisations</li> <li>• Consultation des informations d'une organisation</li> <li>• Ajout de notes et de commentaires</li> </ul>
<b>Gestion des projets</b>	Août : Semaine 1 - 2	<ul style="list-style-type: none"> <li>• Ajout, modification, suppression de projets</li> <li>• Importation et exportation de données de projets</li> </ul>

		<ul style="list-style-type: none"> <li>• Recherche et filtrage des projets</li> <li>• Consultation des informations d'un projet</li> <li>• Ajout de notes et de commentaires</li> </ul>
<b>Visualisation et Exportation des données</b>	Août : Semaine 3 - 4	<ul style="list-style-type: none"> <li>• Représentations graphiques des données</li> <li>• Visualisation de statistiques sur les organisations et les projets</li> <li>• Génération de rapports sur les organisations et les projets</li> <li>• Exportation des rapports en format PDF ou CSV</li> </ul>
<b>Cartographie</b>	Septembre : Semaine 1 - 2	<ul style="list-style-type: none"> <li>• Affichage d'une carte interactive avec les organisations géolocalisées</li> <li>• Filtrage des organisations par zone géographique</li> <li>• Ajout de points d'intérêt et de couches de données</li> <li>• Exportation de la carte</li> </ul>
<b>Fonctionnalités collaboratives</b>	Septembre : Semaine 3 - 4	<ul style="list-style-type: none"> <li>• Partage de documents et de notes</li> </ul>

<b>Aide et documentation</b>	Octobre : Semaine 1 - 2	<ul style="list-style-type: none"> <li>• Aide en ligne et documentation utilisateur</li> <li>• Tutoriels et guides d'utilisation</li> </ul>
------------------------------	-------------------------	---

### **Expllications :**

- **Juillet** : Le début du développement se concentre sur la gestion des utilisateurs et des organisations, deux modules essentiels pour assurer la gestion des données de base.
- **Août** : La priorité est donnée à la gestion des projets et à la manipulation des données (visualisation, recherche, génération de rapports et des statistiques).
- **Septembre** : Le mois de septembre se concentre sur la mise en place de la cartographie, ainsi que des fonctionnalités collaborative.
- **Octobre** : Ce mois est dédié à la mise en place et la finalisation de l'aide et documentation.

### **Notes :**

- Le développement s'est effectué de manière progressive et incrémentielle, permettant de tester chaque fonctionnalité avant d'en commencer une autre.
- Des ajustements avaient été effectués en fonction des priorités ou des retards survenus dans le développement.

Ce planning a non seulement permis de suivre le développement de manière structurée mais aussi d'assurer que l'application réponde aux besoins des utilisateurs finaux et aux spécifications des Termes de Référence. Toutefois, quelle démarche méthodologique a soutenu la mise en application de ce planning ?

## **2.9. Démarches méthodologiques**

Les démarches méthodologiques constituent une étape essentielle dans la réalisation d'un projet, permettant de structurer les activités de développement, de coordonner les équipes et de garantir une livraison efficace et conforme aux attentes. Dans le cadre de ce projet,

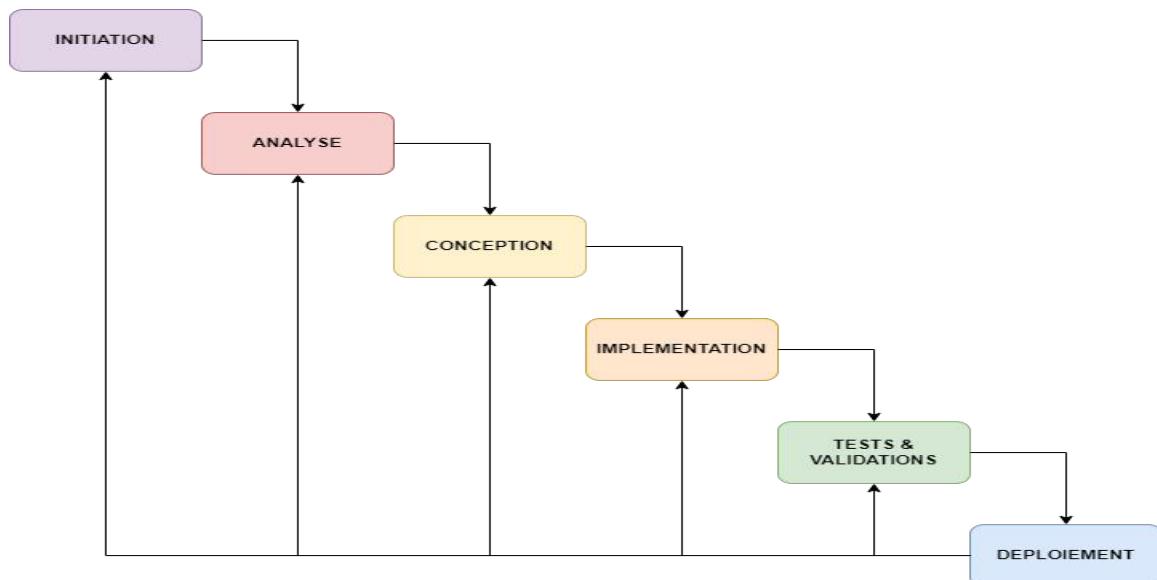
plusieurs approches méthodologiques ont été étudiées pour choisir celle qui s'adapte le mieux aux spécificités de la mission.

### 2.9.1. Étude des choix méthodologiques

Pour répondre aux besoins du projet de développement de l'application web pour la cartographie des OSC, plusieurs approches méthodologiques couramment utilisées dans le domaine du développement logiciel ont été envisagées.

#### ➤ Méthode en cascade : Processus sequentiel [6]

**Le modèle en cascade**, ou « waterfall<sup>1</sup> » en anglais, est une organisation des activités d'un projet sous forme de phases linéaires et séquentielles, où chaque phase correspond à une spécialisation des tâches et dépend des résultats de la phase précédente. Il comprend les phases d'**expression des exigences, d'analyse, de conception, de mise en œuvre et de mise en service**.



**Figure 10 : Illustration de la démarche méthodologique en cascade**

Le modèle en cascade comprend les phases et les livrables suivants :

<sup>1</sup> Appellation anglaise de la méthodologie en cascade.

- a. **Initiation:** Elle fait l'objet d'une **expression des besoins**,
- b. **Analyse:** les exigences sont analysées pour établir un **cahier des charges fonctionnel**,
- c. **Conception:** le produit est *conçu*<sup>1</sup> et spécifié de sorte à pouvoir être réalisé,
- d. **Implémentation** (Mise en œuvre): le produit est *réalisé* sur la base des spécifications,
- e. **Tests & Validations:** le produit est *testé* et *vérifié* et **sa conformité aux exigences est validée**,
- f. **Déploiement** (Mise en service): le produit est *installé* ou *déployé*, les préparatifs pour sa mise en service sont organisés, puis le produit est *utilisé*.

**Chaque phase ne commence qu'une fois les résultats de la phase précédente validés.**

Dans le domaine du développement logiciel, la phase de conception détermine l'architecture du système, la mise en œuvre correspond principalement aux activités de programmation, et la phase de validation comprend pour une grande part des tests.

## **CRITIQUES**

Le modèle en cascade se base sur **des exigences exprimées en début de projet**. Toutefois les exigences et besoins peuvent se montrer incomplets ou de qualité insuffisante (ambiguïté, incohérence, etc.).

De plus, le client peut ne pas être pleinement conscient de ses exigences avant d'avoir vu le logiciel fonctionner. Ceci peut conduire à **revoir** la conception, **redévelopper** une partie du logiciel, et **retester** le produit et donc **augmenter les coûts**. C'est pourquoi le modèle en cascade est particulièrement *adapté* à des projets dont les exigences sont bien comprises et réalisés avec une technologie bien maîtrisée.

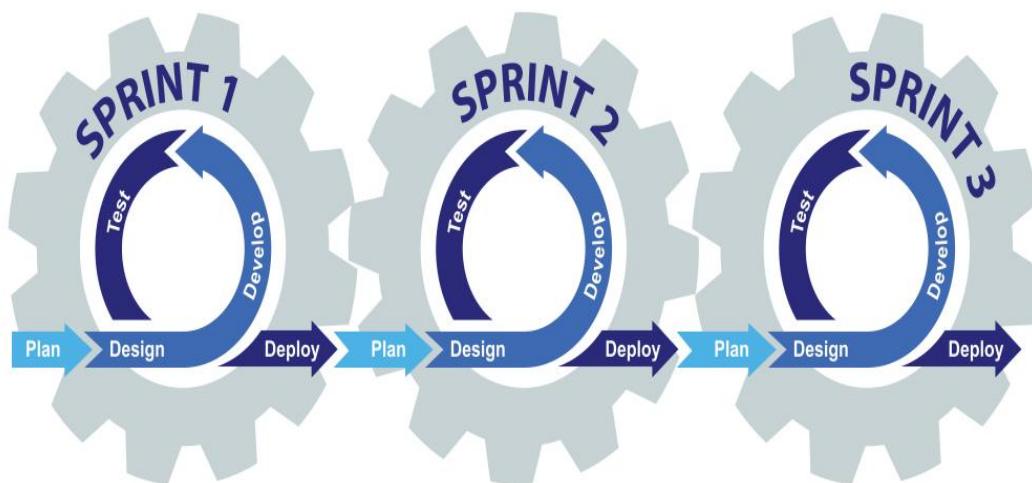
**La structuration des phases par spécialisation d'activité** préconisé par le modèle en cascade est source de rigidité dans l'organisation des travaux, **ne favorise pas suffisamment l'implication du client tout au long du projet**, et décourage la prise en compte des changements.

---

<sup>1</sup> À cette étape, sont produites, les maquettes et modélisations.

➤ **Agile** : Développement itératif avec cycles courts [5]

En ingénierie logicielle, les pratiques agiles mettent en avant **la collaboration entre des équipes auto-organisées et pluridisciplinaires et leurs clients**. Elles s'appuient sur l'utilisation d'un cadre méthodologique léger mais suffisamment centré sur l'humain et la communication. Elles préconisent **une planification adaptative, un développement évolutif, une livraison précoce et une amélioration continue**, et elles encouragent **des réponses flexibles au changement**.



**Figure 11 : Illustration de la méthodologie agile**

Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles (**Waterfall**, etc.), impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles reposent sur **un cycle de développement itératif, incrémental et adaptatif**.

Le *Manifeste pour le développement agile de logiciels* est un texte rédigé aux États-Unis en 2001 par dix-sept experts du développement logiciels. Ils ont proclamé qu'ils attachaient de l'importance:

- ✓ « aux individus et leurs interactions plutôt qu'aux processus et aux outils »,
- ✓ « à un logiciel fonctionnel plutôt qu'à une documentation exhaustive »,
- ✓ « à la collaboration avec les clients plutôt qu'à la négociation contractuelle »,

- ✓ « à l'adaptation au changement plutôt qu'à l'exécution d'un plan ».

Ces quatre citations sont appelées *les quatre « valeurs » du manifeste agile*.

## **CRITIQUES**

Parmi les 17 signataires du *manifeste agile*, certains ont depuis émis des critiques non sur les principes Agiles, mais sur « leur mise en pratique ».

- ✓ **Ron Jeffries** montre plusieurs problèmes :

- le dévoiement des approches agiles (notamment à des fins commerciales) ;
- le rejet des approches agiles par les développeurs ;
- l'imposition de ces approches aux équipes (notamment lors d'une mauvaise utilisation des méthodes SAFe, Scaled Scrum, LeSS, etc.).

Il conseille aux développeurs, non pas d'abandonner l'agile comme le titre de sa note laisse penser, mais de détacher sa réflexion des méthodes agiles imposées et de rester concentré sur le développement d'un logiciel qui marche, par exemple en mettant en pratique **l'eXtreme Programming (XP)**.

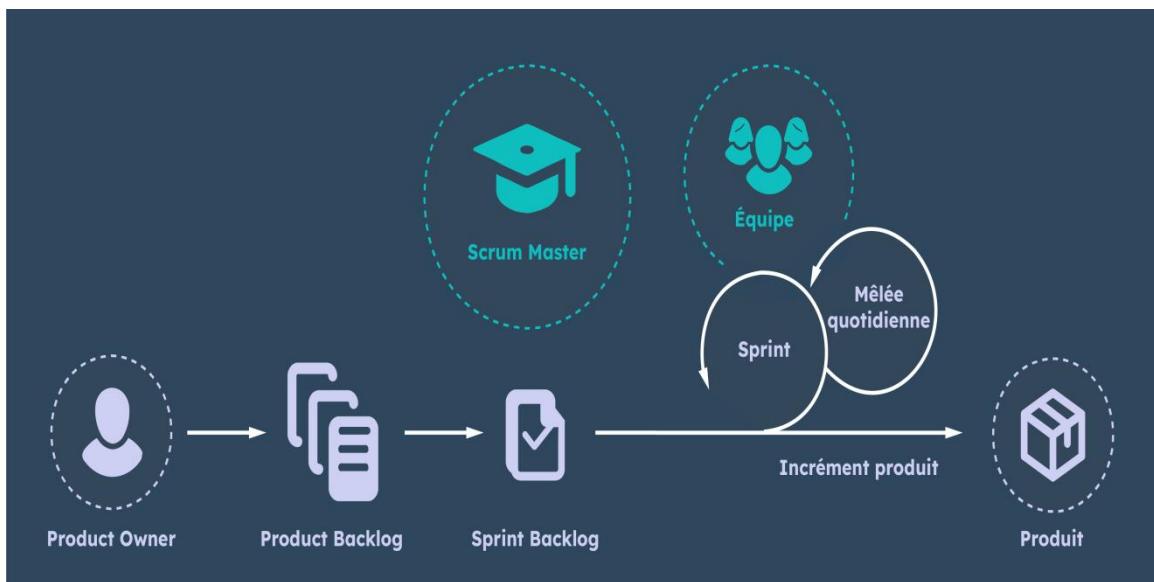
- ✓ **Martin Fowler** indique que la plupart des projets agiles en 2018 sont « faux-agile » et ne respectent pas ses valeurs et ses principes. Ainsi, les trois principaux défis sont de :

- se battre contre l'industrie agile complexe et sa tendance à imposer les processus aux équipes ;
- insister sur l'importance de *l'excellence technique* ;
- organiser les équipes autour d'un *produit* et non d'un *projet*.

### ➤ **Scrum : Framework Agile avec sprints et réunions régulières [7]**

**Scrum** est un framework ou cadre de développement de produits complexes. Il est défini par ses créateurs comme un « **cadre de travail holistique itératif qui se concentre sur les buts communs en livrant de manière productive et créative des produits de la plus grande valeur possible** ». Scrum est considéré comme un groupe de pratiques répondant pour la plupart aux préconisations du *manifeste agile*.

**Scrum** s'appuie sur le découpage d'un projet en « boîtes de temps », nommées **sprints** (pointes de vitesse). Les sprints peuvent durer entre quelques heures et un mois (avec *un sprint médian* à deux semaines). **Chaque sprint commence** par une estimation suivie d'une planification opérationnelle. **Le sprint se termine** par une démonstration de ce qui a été achevé. **Avant de démarrer un nouveau sprint**, l'équipe réalise **une rétrospective**. Cette technique analyse le déroulement du sprint achevé, afin d'améliorer ses pratiques. Le flux de travail de l'équipe de développement est facilité par son auto-organisation, il n'y aura donc pas de gestionnaire de projet.



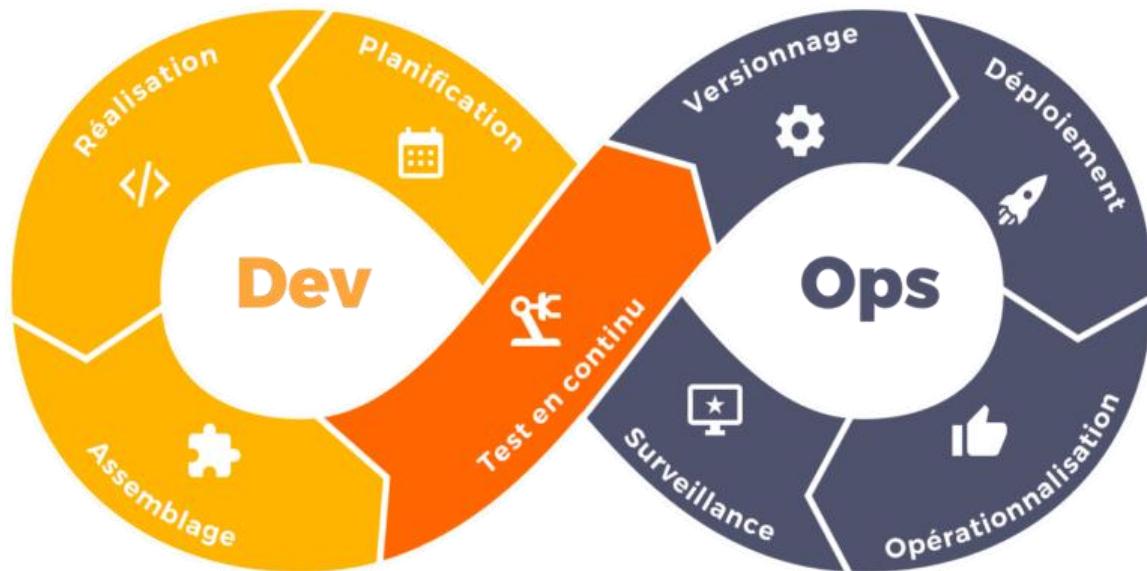
**Figure 12 : Implémentation de la Méthodologie Agile - Framework SCRUM**

**Scrum** est un processus empirique qui s'appuie sur trois piliers : **la transparence, l'inspection et l'adaptation**. Il suit également les principes de *la culture agile*. Scrum met l'accent sur le fait d'avoir **un langage commun entre tous les acteurs liés au produit**. Ce langage commun doit permettre à tout observateur d'obtenir rapidement une bonne compréhension du projet et de son état d'avancement. À intervalles réguliers, Scrum propose de faire le point sur les différents *artéfacts* produits, afin de détecter toute variation indésirable. Si une dérive est constatée pendant l'inspection, le processus doit alors être adapté. Scrum fournit des « événements », durant lesquels cette adaptation est possible. Il s'agit de **la réunion de planification de sprint, de la mêlée quotidienne, de la revue de sprint ainsi que de la rétrospective du sprint**.

➤ **DevOps** : Intégration continue et déploiement continu [4]

**DevOps** est une culture et un ensemble de pratiques qui visent à intégrer **le développement<sup>1</sup>** et **les opérations<sup>2</sup>** pour accélérer les cycles de livraison. Apparu autour de 2007 en Belgique, le mouvement « DevOps » se caractérise principalement par la promotion de l'**automatisation** et du suivi (**monitoring**) de toutes les étapes de la création d'un logiciel, depuis le développement, l'intégration, les tests, la livraison jusqu'au déploiement, l'exploitation et la maintenance des infrastructures.

« Les principes DevOps » soutiennent **des cycles de développement plus courts, une augmentation de la fréquence des déploiements et des livraisons continues**, pour une meilleure atteinte des objectifs économiques d'une entreprise.



**Figure 13** : Illustration de l'approche DevOps

Sanjeev Sharma et Bernie Coyne recommandent :

- ✓ **un déploiement régulier des applications**, la seule répétition contribuant à fiabiliser le processus ;

<sup>1</sup> Dev.

<sup>2</sup> Ops.

- ✓ **un décalage des tests « vers la gauche »,** autrement dit de tester au plus tôt ;
- ✓ **une pratique des tests dans un environnement similaire à celui de production ;**
- ✓ **une intégration continue** incluant des tests continus ;
- ✓ **une boucle d'amélioration courte** (c'est-à-dire un feed-back rapide des utilisateurs) ;
- ✓ **une surveillance étroite de l'exploitation et de la qualité de production** factualisée par des métriques et indicateurs clés.

### **Récapitulatif des méthodologies abordées :**

Après avoir exploré les différentes méthodologies de développement logiciel, il est important de les résumer de manière concise pour mieux comprendre leurs caractéristiques, avantages et inconvénients. Le tableau suivant permet de mettre donc en évidence les principales différences entre les méthodologies abordées, offrant ainsi une vue d'ensemble claire des choix possibles en fonction des besoins spécifiques du projet.

Tableau 3 : Récapitulatif des choix méthodologiques étudiés

Méthodologie	Description	Avantages	Inconvénients
<b>DevOps</b>	Culture et pratiques intégrant développement et opérations pour accélérer la livraison continue.	<ul style="list-style-type: none"> <li>• Automatisation des processus</li> <li>• Amélioration de la collaboration</li> <li>• Livraison continue</li> </ul>	<ul style="list-style-type: none"> <li>• Nécessite des outils spécialisés</li> <li>• Intégration complexe</li> </ul>
<b>Méthode Agile</b>	Approche itérative et incrémentale centrée sur la collaboration et l'adaptabilité, avec des cycles courts (sprints).	<ul style="list-style-type: none"> <li>• Forte implication des parties prenantes</li> <li>• Adaptation rapide aux changements</li> </ul>	<ul style="list-style-type: none"> <li>• Moins structuré</li> <li>• Peut entraîner des dérives sans gestion rigoureuse</li> </ul>
<b>Méthode en cascade</b>	Méthode linéaire et séquentielle où chaque phase est terminée avant de passer à la suivante.	<ul style="list-style-type: none"> <li>• Clarté des étapes</li> <li>• Documentation complète et facile à suivre</li> </ul>	<ul style="list-style-type: none"> <li>• Manque de flexibilité</li> <li>• Difficulté à s'adapter aux changements en cours de développement</li> </ul>
<b>Méthodologie Scrum</b>	Cadre Agile avec des rôles et événements spécifiques, s'appuyant sur des sprints (2-4 semaines).	<ul style="list-style-type: none"> <li>• Organisation claire des équipes</li> <li>• Suivi rigoureux des tâches</li> </ul>	<ul style="list-style-type: none"> <li>• Nécessite une équipe expérimentée</li> <li>• Peut être difficile à gérer sans un Scrum Master dédié</li> </ul>

## 2.9.2. Choix méthodologique

Pour ce projet, le choix méthodologique s'est porté sur **une approche hybride** ; mélangeant **Cascade et Agilité (sans cadre particulier)**.

Ce choix de couplage s'explique par plusieurs raisons :

- **Linéarité** : **Cascade** allait permettre de rendre la gestion globale du projet linéaire en faisant progressivement passer de l'expression des besoins à la mise en production tout en passant par la phase d'analyse, de conception, de développement, et de tests & validations.
- **Flexibilité** : **Agile** permettrait d'adapter facilement le travail [de développement] en fonction des retours des commanditaires (Expertise France, SUN et les faîtières d'OSC) ainsi que des contraintes techniques rencontrées lors du développement.
- **Collaborations fréquentes** : Avec une forte implication des parties prenantes, il est essentiel de pouvoir intégrer les retours au fur et à mesure du développement.
- **Focus sur les livrables prioritaires** : L'approche Agile favorise l'appréciation<sup>1</sup> progressive des fonctionnalités clés (comme les filtres de recherche, la cartographie interactive, etc.) pour permettre des tests et des ajustements rapides.

Contrairement à un cadre strict comme Scrum, l'utilisation d'Agile sans cadre spécifique permet de simplifier la gestion et d'éviter des contraintes organisationnelles inutiles. Avec le reste de l'équipe, je pouvais ainsi me concentrer sur les besoins du projet tout en restant agile et réactif.

En optant pour **une méthodologie Hybride**, le projet bénéficie d'une approche adaptable et centrée sur les besoins des utilisateurs finaux. Ce choix permet non seulement de livrer un produit de qualité répondant aux attentes des commanditaires, mais également de faciliter les interactions entre les équipes techniques et les parties prenantes tout au long du cycle de développement. Cette méthodologie garantit une linéarité, une exécution efficace et une flexibilité essentielle dans un contexte où les exigences et les priorités peuvent évoluer rapidement.

---

<sup>1</sup> En Agilité, on parle, généralement, de livraison, car on livre progressivement les fonctionnalités. Mais dans notre contexte, il n'y a qu'un seul déploiement attendu. Les rencontres en ligne [et l'atelier de restitution] permettaient donc d'apprécier les fonctionnalités afin de les valider ou les invalider.

## **Conclusion**

L'étude préalable du projet a permis de mettre en lumière les besoins, les enjeux et les objectifs liés à la mise en œuvre de **la cartographie des Organisations de la Société Civile (OSC) actives sur les questions de Nutrition et Alimentation Saine au Togo**. Nous avons examiné **le contexte global du projet**, identifié **les problématiques** qui ont motivé la demande initiale, et analysé **les limites potentielles d'une gestion exclusivement basée sur une base de données Excel**. Cette analyse a justifié la nécessité de développer une application web capable de dépasser ces limitations tout en répondant aux attentes des parties prenantes.

Dans ce chapitre, nous avons également étudié **l'existant**, critiqué ses faiblesses, défini **les objectifs du projet**, et proposé des solutions adaptées, tout en tenant compte des ressources et contraintes identifiées. **Une évaluation financière et un planning prévisionnel** ont été établis pour encadrer la réalisation du projet dans un cadre réaliste et pragmatique. Enfin, le choix d'**une méthodologie Hybride**, en réponse à la flexibilité requise par le projet, s'est également avéré pertinent pour répondre aux besoins évolutifs des utilisateurs et intégrer efficacement leurs retours au cours du développement.

Ces éléments constituent une base solide pour aborder les étapes suivantes, notamment l'analyse détaillée et la conception technique du système, qui seront développées dans le chapitre suivant.

# **Chapitre 3 : ANALYSE ET**

## **CONCEPTION**

**L'analyse et la conception** jouent un rôle central dans le cycle de vie d'un projet informatique, car elles permettent de passer de la formulation des besoins fonctionnels à une structure formelle et cohérente du système à développer. Ces étapes garantissent non seulement la faisabilité technique du projet, mais aussi sa capacité à évoluer pour répondre aux attentes futures des utilisateurs.

Dans ce chapitre, nous abordons en détail les approches d'analyse adoptées, à savoir la **POO**<sup>1</sup> et la modélisation **ER**<sup>2</sup>, ainsi que le langage **UML**<sup>3</sup> qui constitue l'outil principal de représentation graphique des modèles.

Ensuite, une étude détaillée du projet est réalisée à travers une série de **diagrammes UML**, allant du diagramme de contexte au modèle Entité-Relation. Ces représentations offrent une vue d'ensemble complète des aspects fonctionnels, structurels et dynamiques du système, servant de référence pour le développement technique.

Enfin, les choix techniques retenus pour le développement et le déploiement sur le réseau sont expliqués, en présentant **le modèle de communication**, **l'architecture de déploiement** et **l'architecture de code**. Ce chapitre constitue ainsi une étape clé dans la transition vers la réalisation concrète du projet.

### 3.1. Présentation de la méthode d'analyse

La méthode d'analyse adoptée pour ce projet repose sur deux approches complémentaires : la **Programmation Orientée Objet** et la **modélisation Entité-Relation**. Ces deux approches permettent de structurer les données et les fonctionnalités de manière cohérente, tout en assurant une adaptabilité aux besoins évolutifs du projet.

#### 3.1.1. Programmation Orientée Objet (POO)

La POO est une approche d'analyse et de développement qui se concentre sur *la modélisation des concepts du système sous forme d'objets*. Un **objet** est une entité qui combine des

---

<sup>1</sup> POO : Programmation Orientée Objet. Elle est utilisée pour la modélisation de l'application.

<sup>2</sup> ER : Entité-Relation. Elle est utilisée dans le cadre de la modélisation de la base de données.

<sup>3</sup> UML : Unified Modeling Language.

**attributs**<sup>1</sup> et des **méthodes**<sup>2</sup>, reflétant les éléments du monde réel ou du système à concevoir. Cette méthode est particulièrement adaptée aux projets nécessitant une structuration claire des données et une forte modularité.

Dans le cadre du projet de cartographie des OSC, la POO permet de représenter les entités du système, telles que les OSC, les utilisateurs, et les projets, sous forme d'objets. Chaque entité possède des attributs spécifiques (par exemple, nom, localisation, domaine d'intervention pour les OSC) et des méthodes qui définissent ses interactions (par exemple, ajouter des données, consulter des rapports).

Les avantages de la POO incluent :

- **Réutilisabilité** : Les objets créés peuvent être utilisés dans plusieurs parties du système ou dans d'autres projets similaires.
- **Modularité** : Chaque objet est indépendant, ce qui facilite les modifications et l'ajout de nouvelles fonctionnalités.
- **Extensibilité** : Le système peut évoluer facilement en ajoutant de nouveaux objets ou en modifiant les objets existants.

### 3.1.2. Modélisation Entité-Relation (ER)

La modélisation Entité-Relation est une méthode d'analyse axée sur **la structuration des données**. Elle repose sur la représentation des entités (objets du système) et des relations entre ces entités. Un **modèle ER** est généralement constitué de :

- **Entités** : Les éléments principaux du système (par exemple, les OSC ou les utilisateurs).
- **Attributs** : Les caractéristiques ou propriétés des entités (par exemple, le nom, l'adresse, ou le secteur d'intervention).
- **Relations** : Les liens entre les entités (par exemple, une OSC peut avoir des projets associés ou être localisée dans une région donnée).

---

<sup>1</sup> Caractéristiques.

<sup>2</sup> Comportements.

Cette approche est essentielle pour concevoir la base de données du projet, car elle permet de visualiser et de structurer les données de manière logique avant leur implémentation. Dans le contexte de ce projet, la modélisation ER facilite l'identification des données critiques et les relations complexes entre elles, comme les collaborations entre OSC ou la couverture géographique de leurs interventions.

En combinant la POO et le modèle ER, cette méthode d'analyse offre une vision complète et cohérente du système, à la fois en termes de structure des données et de leur interaction dynamique.

### 3.2. Présentation du langage de modélisation

Le **langage UML** a été choisi comme outil principal pour la modélisation des concepts du système. Il s'agit d'un langage standardisé utilisé pour modéliser les systèmes complexes, particulièrement dans le cadre de *projets orientés objet*. Il offre une série de diagrammes qui permettent de représenter les différents aspects du système, facilitant ainsi la communication entre les parties prenantes et la validation des spécifications.

#### 3.2.1. Pourquoi UML ?

UML est un langage polyvalent qui offre plusieurs avantages :

- **Standardisation** : UML est largement utilisé et reconnu dans le domaine de l'ingénierie logicielle, ce qui garantit une compréhension commune entre les développeurs, les analystes et les autres parties prenantes.
- **Polyvalence** : Il permet de représenter aussi bien *les aspects fonctionnels* (diagrammes de cas d'utilisation) que *structurels* (diagrammes de classes) et *dynamiques* (diagrammes de séquences ou d'activités).
- **Adaptabilité** : Les diagrammes UML peuvent être adaptés aux besoins spécifiques du projet, qu'il s'agisse d'une petite application ou d'un système complexe.

### 3.2.2. Application d'UML dans le projet

Pour ce projet, UML a été utilisé dès les premières étapes de l'analyse pour clarifier les exigences fonctionnelles et identifier les entités clés du système. Au stade de la conception, UML a permis de formaliser la structure et le comportement du système à travers des diagrammes détaillés, assurant ainsi une transition fluide vers le développement.

Ainsi, dans le cadre de la conception de l'application web de cartographie des OSC, plusieurs types de diagrammes UML ont été utilisés pour capturer les différentes dimensions du système :

- **Diagramme de contexte** : Présente une vue d'ensemble des interactions entre le système et ses acteurs.
- **Diagramme de package** : Organise les composants du système en sous-systèmes ou modules.
- **Diagramme de cas d'utilisation** : Décrit les fonctionnalités principales du système du point de vue des utilisateurs.
- **Diagrammes d'activités** : Illustrent les workflows<sup>1</sup> ou processus au sein du système.
- **Diagramme de classes** : Représente la structure statique des objets et leurs relations.

En résumé, l'utilisation d'UML comme langage d'analyse garantit une modélisation claire, compréhensible et exhaustive du système, tout en facilitant la collaboration entre les différentes parties prenantes du projet.

### 3.3. Étude détaillée du projet

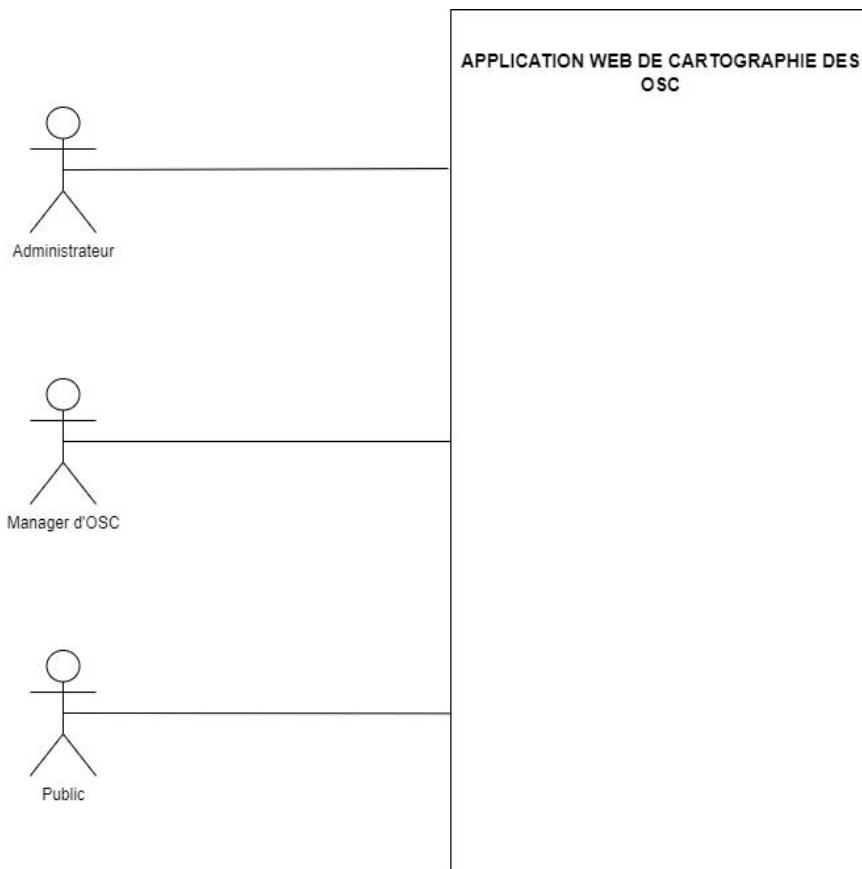
Dans cette section, nous présentons les différentes modélisations UML utilisées pour structurer et analyser le projet. Ces modèles visent à offrir une représentation claire et visuelle des aspects fonctionnels et structurels du système, facilitant ainsi la compréhension et la communication des besoins, ainsi que la conception du système.

---

<sup>1</sup> Anglicisme généralement utilisé en informatique, faisant référence au(x) « flux de travail ».

### 3.3.1. Diagramme de contexte

Le **diagramme de contexte** permet de définir **les frontières du système** et de visualiser **les interactions principales entre le système et ses environnements externes**. Il présente les acteurs externes (utilisateurs, autres systèmes) qui interagissent avec le système sans entrer dans les détails de la manière dont ces interactions sont réalisées.



**Figure 14 :** Diagramme de contexte de l'application web de cartographie des OSC

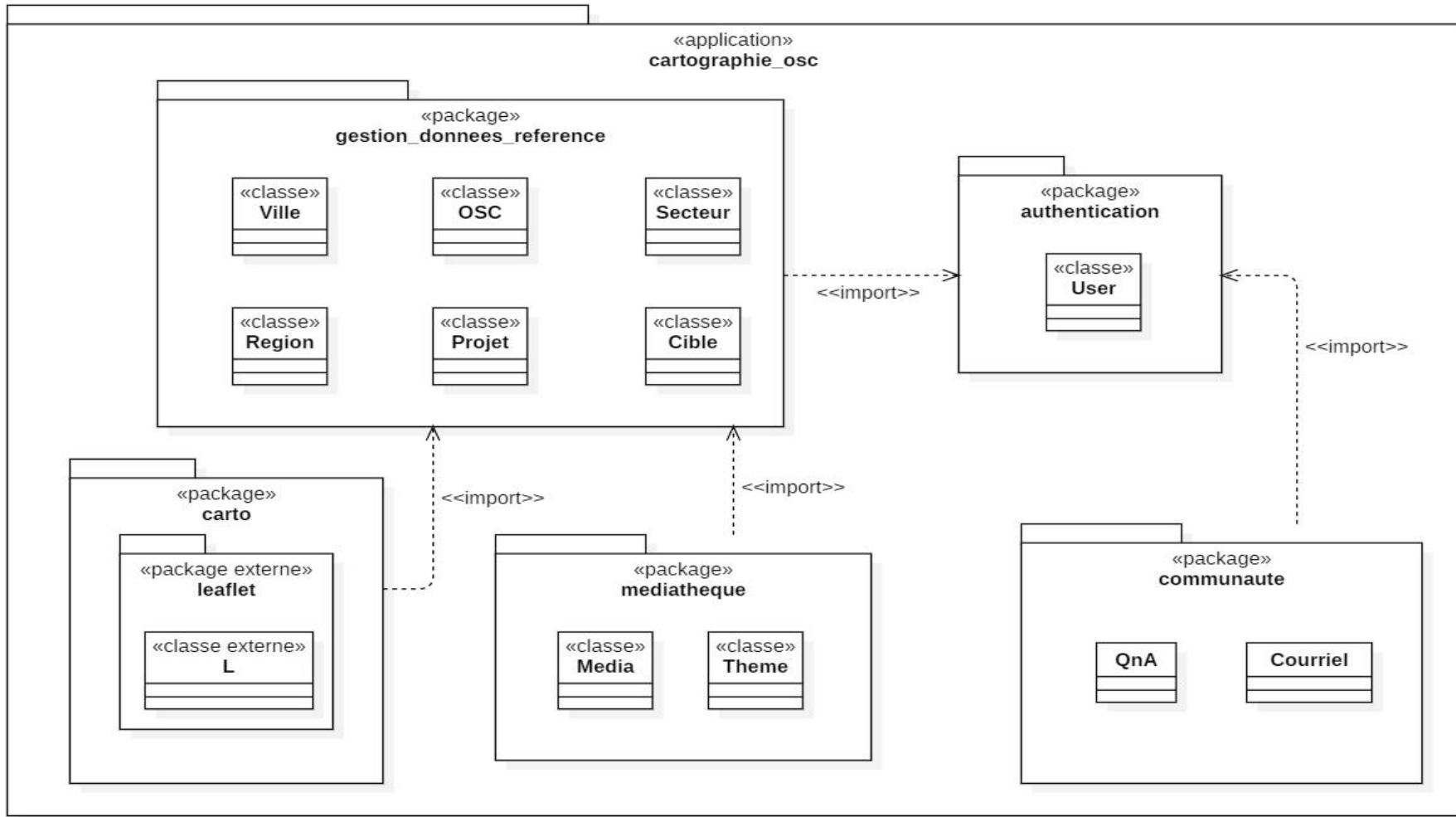
Chaque acteur du système incarne un rôle distinct avec des responsabilités spécifiques. Les interactions possibles entre un utilisateur et le système varient en fonction de son profil, définissant ainsi les actions et fonctionnalités qui lui sont accessibles.

**Tableau 4 : Description textuelle du diagramme de contexte**

Acteur	Description
<b>Public</b>	Les utilisateurs par-défaut qui consultent les informations publiques sur les OSC et les projets.
<b>Manager d'OSC</b>	Ils gèrent les données des OSC et des projets spécifiques.
<b>Administrateur</b>	Il gère l'application, les utilisateurs, et les données.

### 3.3.2. Diagramme de package

Le **diagramme de package** est utilisé pour structurer le système en regroupant des classes ou des composants qui partagent des responsabilités similaires. Il permet de visualiser **l'organisation du système en sous-systèmes**, ce qui facilite la gestion de la complexité et l'évolution du projet.



**Figure 15 :** Diagramme de package de l'application web de cartographie des OSC

Chaque package regroupe des éléments partageant des responsabilités communes, reflétant une organisation logique du système. Cette structure permet de simplifier la gestion de la complexité en isolant les fonctionnalités principales, facilitant ainsi leur compréhension et leur évolution.

**Tableau 5 : Description textuelle du diagramme de package**

Package	Description
<b>authentication</b>	Package contenant l'ensemble des modules liés à l'authentification
<b>carto</b>	Package contenant l'ensemble des modules liés à la cartographie
<b>cartographie_osc<sup>1</sup></b>	Package représentant l'application dans sa globalité et contenant tous les autres packages utiles à l'exécution de l'application
<b>communaute</b>	Package contenant l'ensemble des modules liés à l'espace communautaire (discussions, FAQs <sup>2</sup> , événements, contacter la communauté)
<b>gestion_donnees_reference</b>	Package contenant l'ensemble des modules liés à la gestion des données de référence ; à savoir les osc, projet, région, ville, secteur, cible, etc.
<b>leaflet</b>	Package JavaScript externe contenant les modules et fonctions permettant de créer une carte géographique du monde et de la manipuler grâce à des données de géolocalisation
<b>mediatheque</b>	Package contenant l'ensemble des modules liés à

<sup>1</sup> On notera ici, l'utilisation de la convention « snake case » pour nommer les packages. Elle consiste à écrire des noms de variables, de fonctions, ou d'autres éléments avec des mots séparés par des underscores (\_), tout en utilisant uniquement des lettres minuscules.

<sup>2</sup> FAQs : Foire Aux Questions. De l'anglais : Frequently Asked Questions.

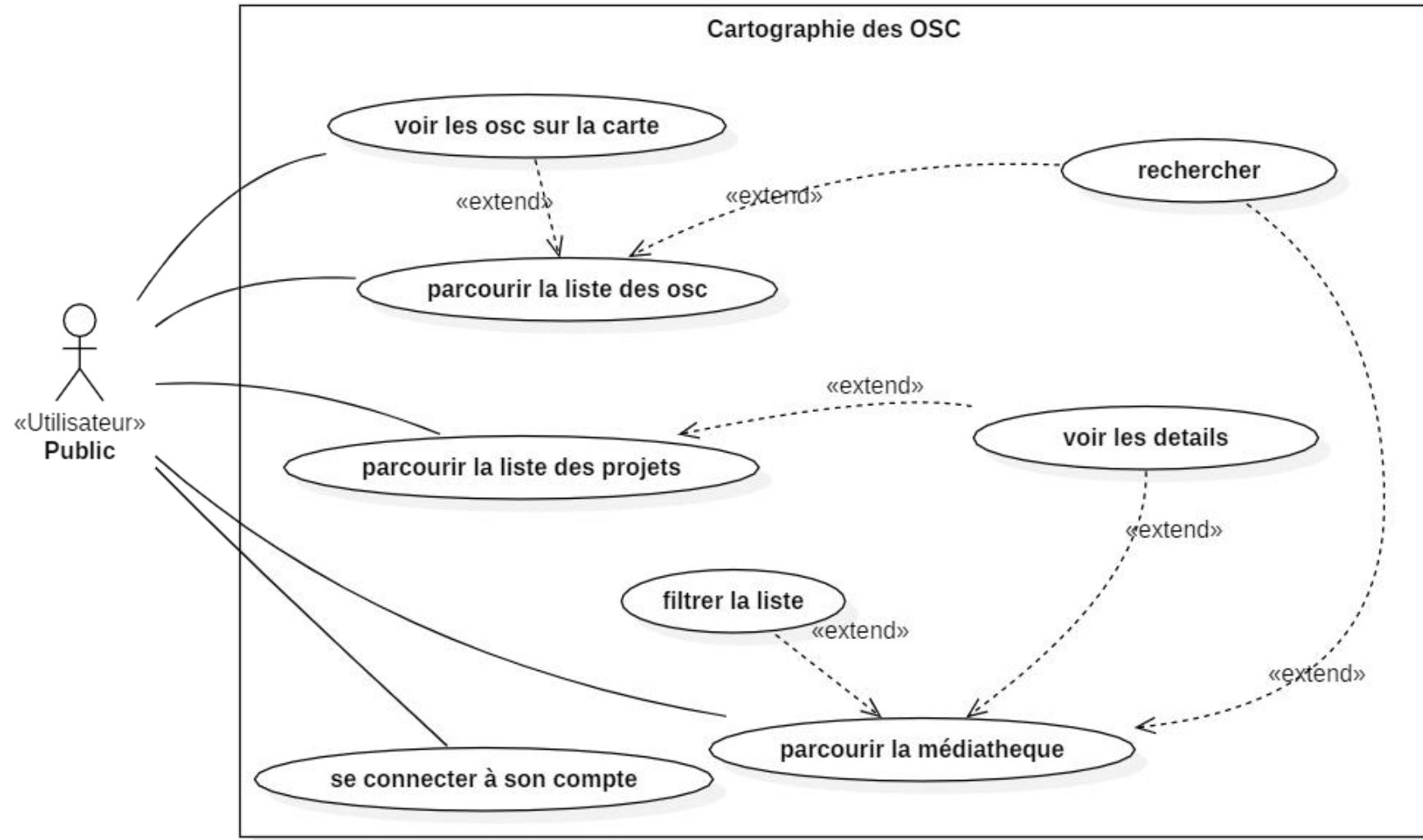
la gestion des média (texte, image, audio, vidéo)  
de documentation sur les question de Nutrition  
et Alimentation Saine

### 3.3.3. Diagrammes de cas d'utilisation

Les **diagrammes de cas d'utilisation** sont utilisés pour décrire **les fonctionnalités** que le système doit offrir **du point de vue des utilisateurs** (acteurs). Chaque cas d'utilisation représente une interaction spécifique entre un acteur et le système, et vise à montrer comment les utilisateurs interagiront avec le système.

Voici les cas d'utilisation en fonction des acteurs :

- **Public** : Consulter les informations des OSC, explorer les projets et accéder à la médiathèque.



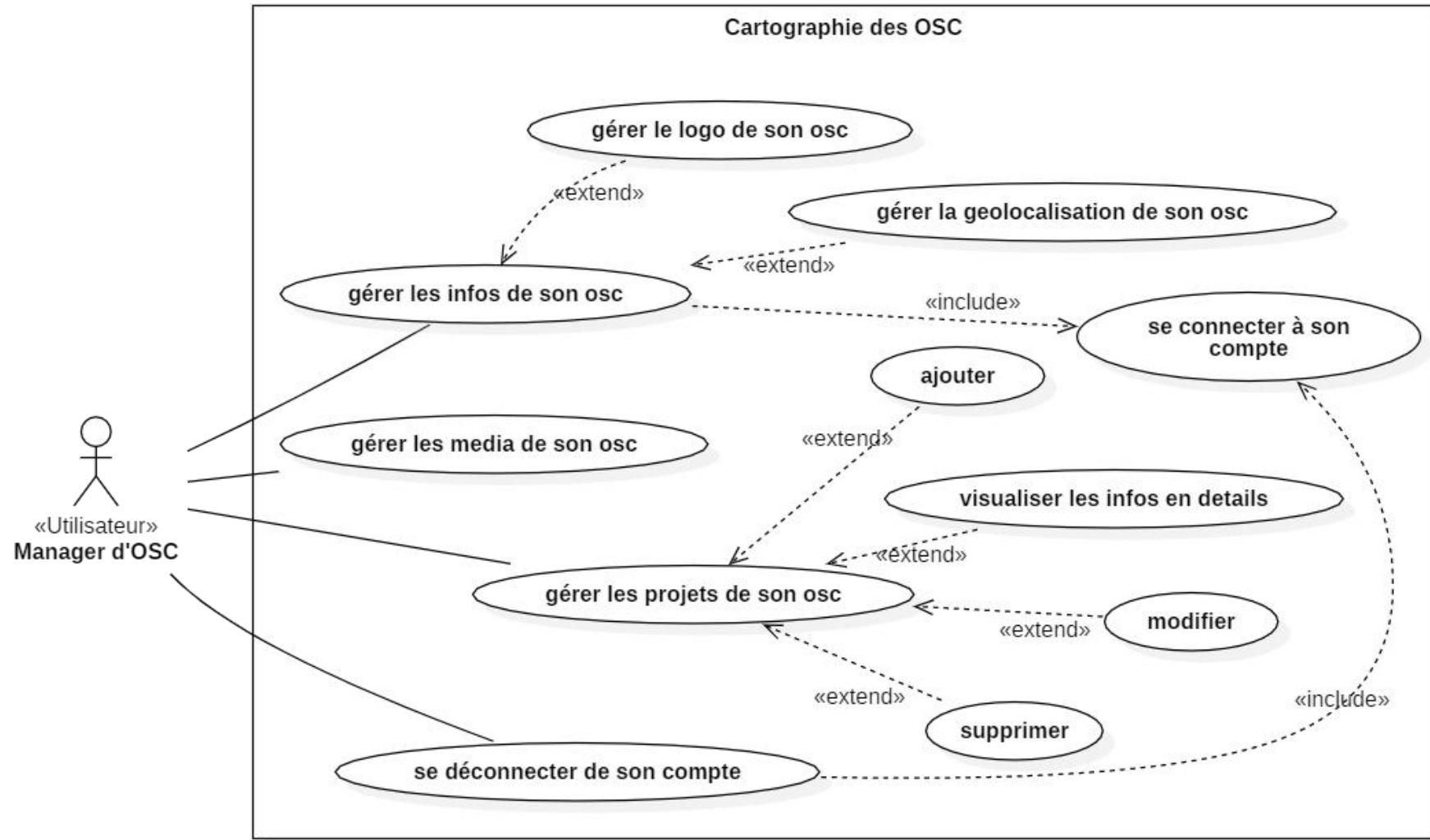
**Figure 16 :** Diagramme de cas d'utilisation de l'utilisateur par défaut

Les interactions possibles pour un utilisateur par défaut se concentrent sur l'accès aux informations publiques des OSC, des projets, et de la médiathèque. Ces cas d'utilisation mettent en évidence les fonctionnalités accessibles sans authentification, permettant une exploration ouverte du système.

**Tableau 6 : Description textuelle des cas d'utilisation de l'utilisateur par défaut**

Cas d'utilisation	Scénario principal	Scénario alternatif
<b>Parcourir la liste des osc</b>	L'utilisateur accède à la liste des OSC et peut consulter les détails d'une OSC spécifique.	Les informations d'une OSC sont temporairement indisponibles ou non mises à jour.
<b>Parcourir la liste des projets</b>	L'utilisateur explore les projets en cours ou réalisés par les OSC, avec des options de tri et de recherche.	Certains projets ne sont pas visibles en raison de restrictions ou de données manquantes.
<b>Parcourir la médiathèque</b>	L'utilisateur accède à des ressources multimédias partagées, telles que des images, vidéos, ou documents relatifs aux OSC et à leurs projets.	Certaines ressources ne se chargent pas correctement en raison d'une erreur réseau ou d'un format non pris en charge.
<b>Se connecter à son compte</b>	L'utilisateur saisit ses identifiants pour accéder à un espace personnel.	Échec de la connexion en cas de mot de passe incorrect ou de compte inexistant.
<b>Voir les OSC sur la carte</b>	L'utilisateur visualise les OSC affichées sur la carte interactive en fonction des filtres appliqués.	Aucun résultat trouvé si les filtres appliqués ne correspondent à aucune OSC.

- #### ➤ Manager d'OSC : Gérer son OSC et ses projets.



**Figure 17 : Diagramme de cas d'utilisation d'un manager d'OSC**

Le manager d'OSC dispose de priviléges spécifiques pour gérer les informations liées à son organisation. Ces cas d'utilisation décrivent les principales fonctionnalités auxquelles il a accès pour assurer la mise à jour et la visibilité des données associées à son OSC.

**Tableau 7 : Description textuelle des cas d'utilisation d'un manager d'OSC**

Cas d'utilisation	Scénario principal	Scénario alternatif
<b>Gérer les infos de son OSC</b>	Le manager met à jour les informations de son OSC (nom, adresse, contact, etc.) via une interface dédiée.	Certaines modifications échouent en raison de validations ou d'un manque d'autorisation.
<b>Gérer les médias de son OSC</b>	Le manager ajoute, modifie ou supprime des fichiers multimédias associés à son OSC (photos, documents).	Téléchargement ou suppression échoués à cause d'un fichier corrompu ou d'une limite de stockage atteinte.
<b>Gérer les projets de son OSC</b>	Le manager crée, modifie ou archive des projets réalisés par son OSC, avec des descriptions détaillées et des documents joints.	Des erreurs surviennent lors de la création ou de la modification en raison de données manquantes ou d'un bug du système.
<b>Se déconnecter de son compte</b>	Le manager ferme sa session en toute sécurité pour revenir à l'interface publique.	La session reste ouverte en cas de déconnexion non réussie, nécessitant une fermeture forcée de l'application.

- **Administrateur** : Gérer les OSC, les projets, les utilisateurs, et toutes les données du système.

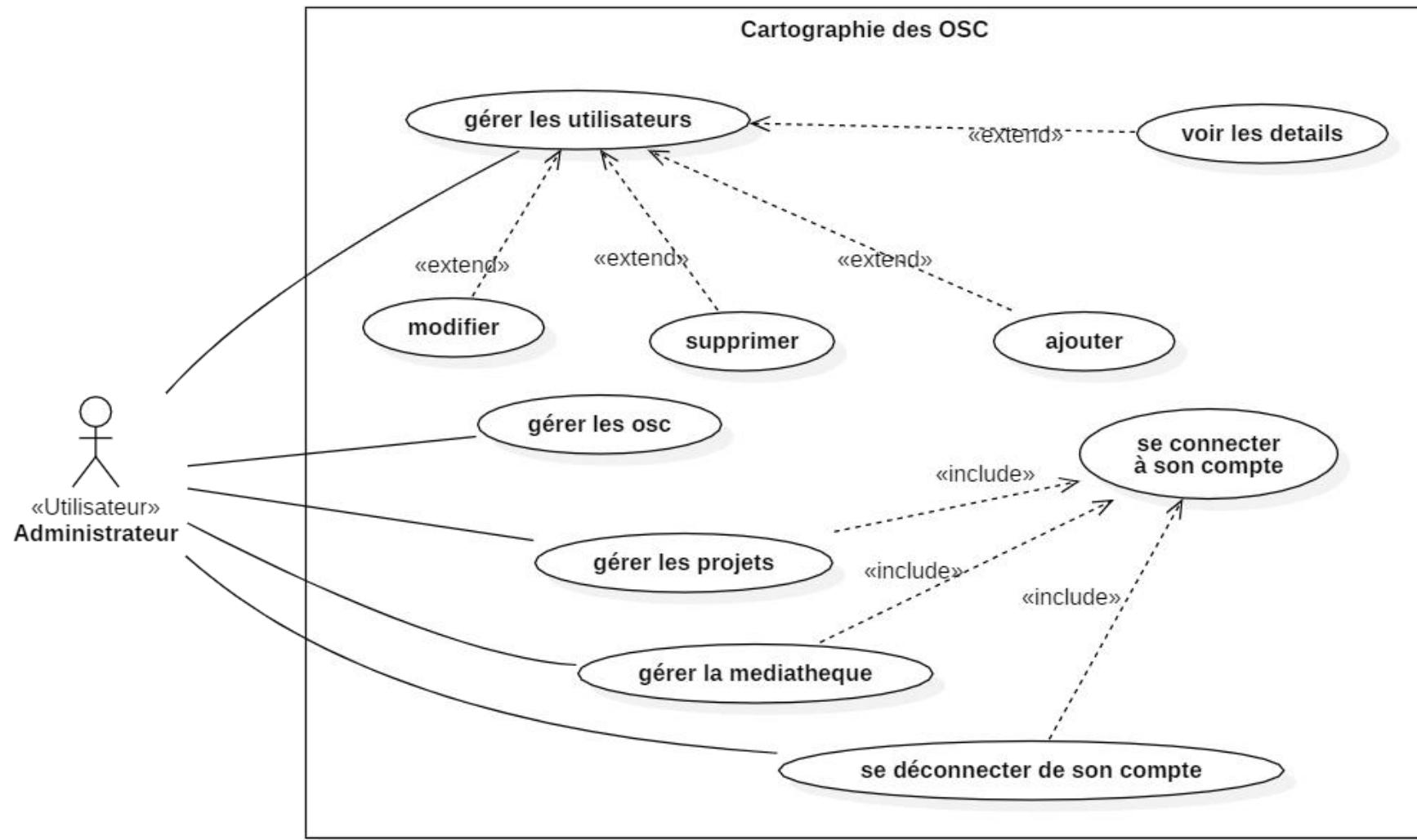


Figure 18 : Diagramme de cas d'utilisation de l'administrateur

L'administrateur possède les droits les plus étendus dans le système, lui permettant de gérer les données critiques et les utilisateurs. Ces cas d'utilisation détaillent les responsabilités liées à son rôle, visant à garantir le bon fonctionnement et la cohérence globale de l'application.

**Tableau 8 : Description textuelle des cas d'utilisation de l'administrateur**

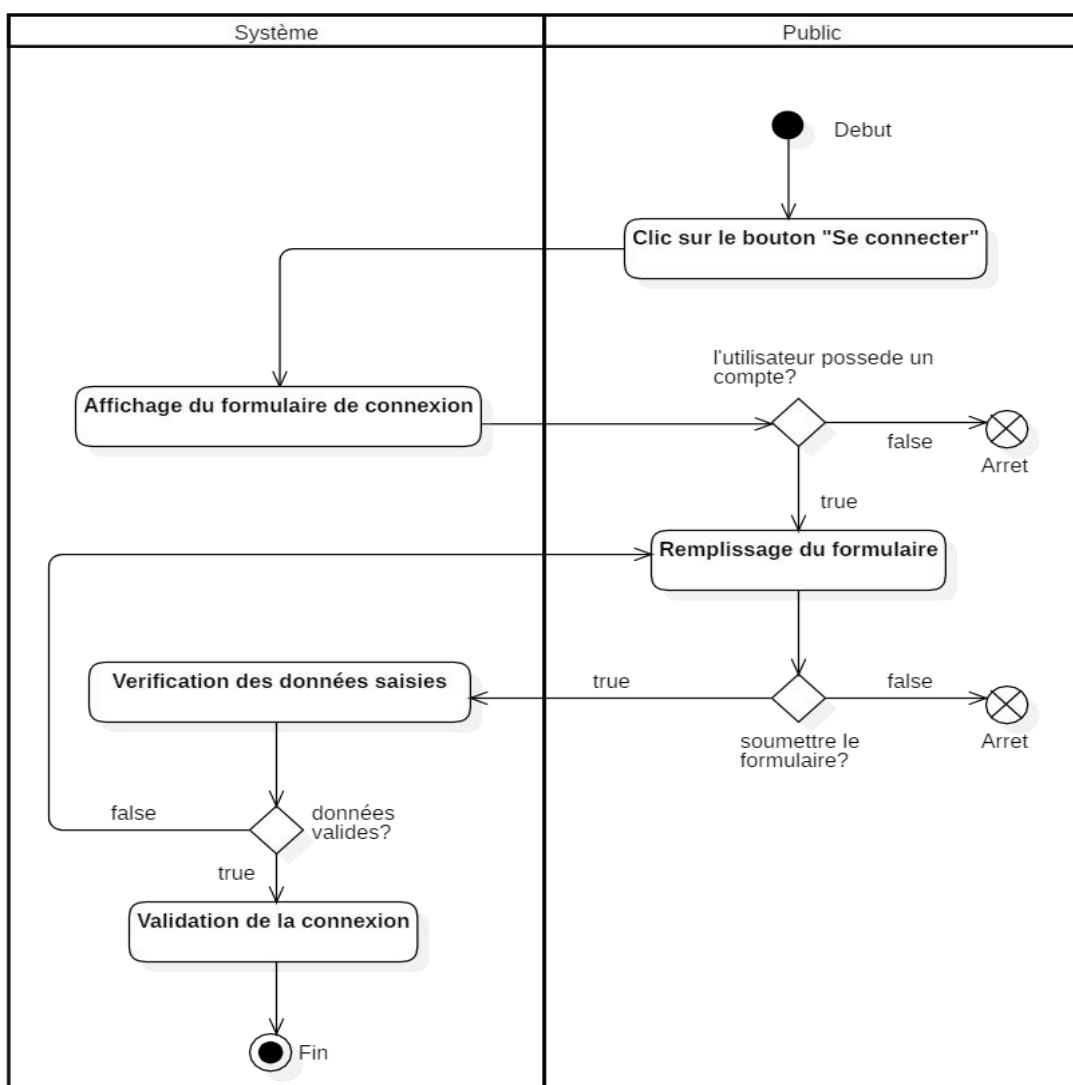
Cas d'utilisation	Scénario principal	Scénario alternatif
<b>Gérer la médiathèque</b>	L'administrateur contrôle et organise les ressources multimédias (ajout, modification, suppression) accessibles sur la plateforme.	Certaines ressources ne peuvent pas être supprimées ou modifiées en raison de dépendances ou de restrictions techniques.
<b>Gérer les osc</b>	L'administrateur vérifie, approuve ou modifie les informations soumises par les managers d'OSC pour garantir leur conformité.	Les informations d'une OSC ne sont pas validées en raison de données incomplètes ou incohérentes.
<b>Gérer les projets</b>	L'administrateur supervise et approuve les projets soumis par les OSC, assurant leur alignement avec les objectifs de la plateforme.	Les projets sont rejetés ou marqués comme incomplets en cas de description insuffisante ou de documents manquants.
<b>Gérer les utilisateurs</b>	L'administrateur ajoute, modifie ou désactive les comptes utilisateur dans le système.	Échec d'ajout ou de modification en cas de duplication ou de non-respect des règles de sécurité des mots de passe.
<b>Se déconnecter de son compte</b>	L'administrateur quitte sa session après avoir effectué les opérations nécessaires.	Une déconnexion forcée peut être requise en cas d'erreur ou d'inactivité prolongée.

### 3.3.4. Diagrammes d'activités

Les diagrammes d'activités représentent les processus internes du système, montrant les différentes étapes qu'un flux d'activité peut traverser. Ces diagrammes sont particulièrement utiles pour modéliser les processus métiers, les flux de travail et les transitions d'état dans le système.

Voici quelques diagrammes d'activités :

➤ **Se connecter :**



**Figure 19 : Diagramme d'activités illustrant la connexion d'un utilisateur**

➤ Ajouter une OSC :

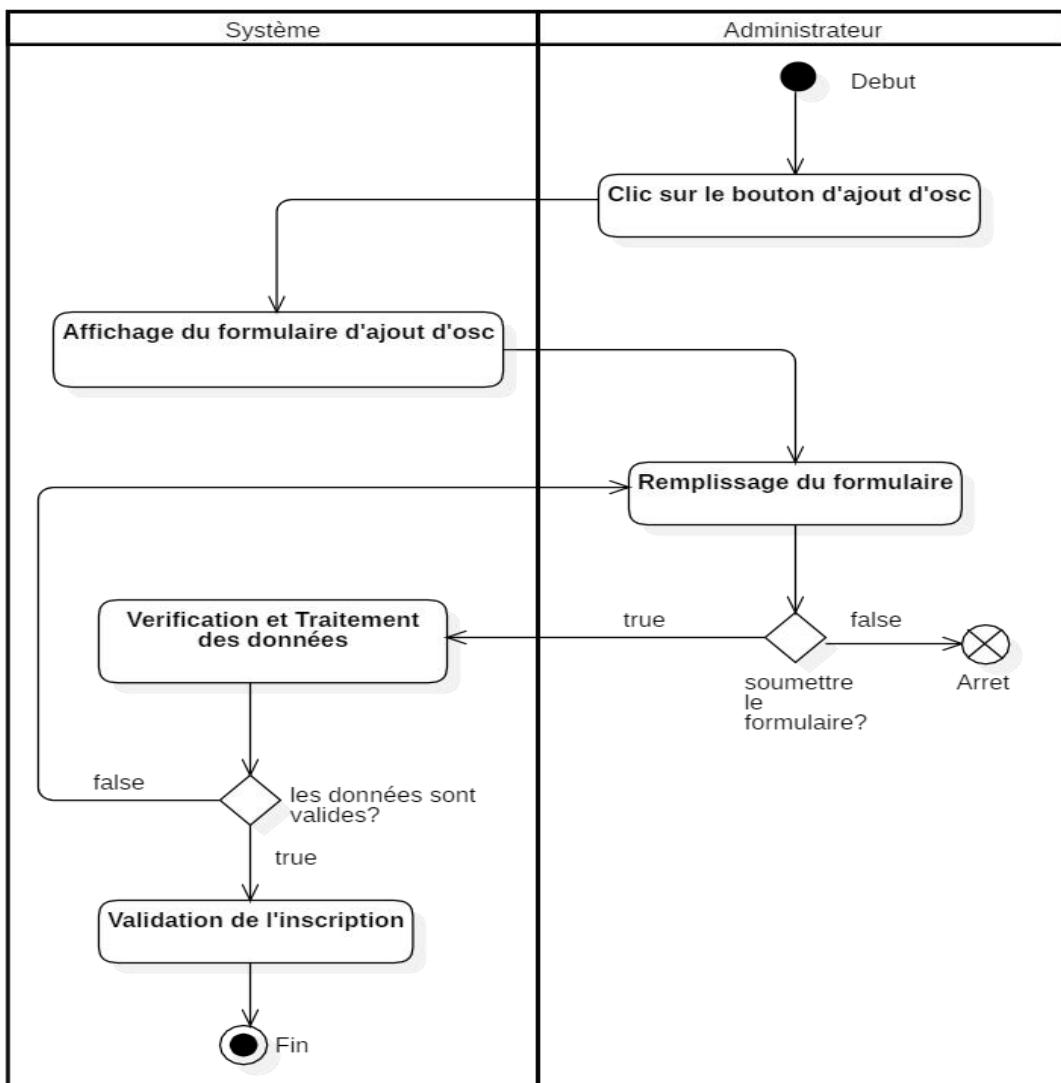
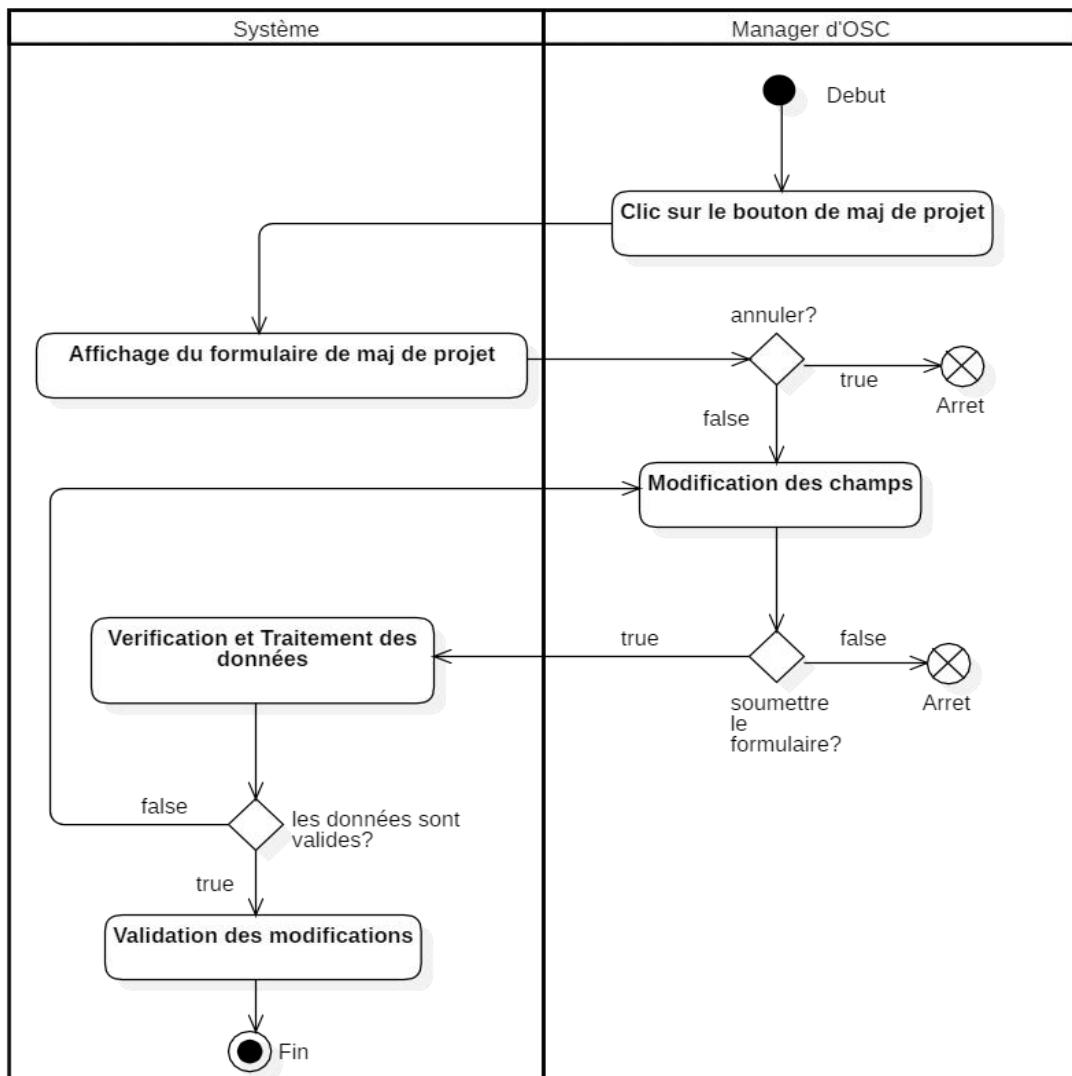


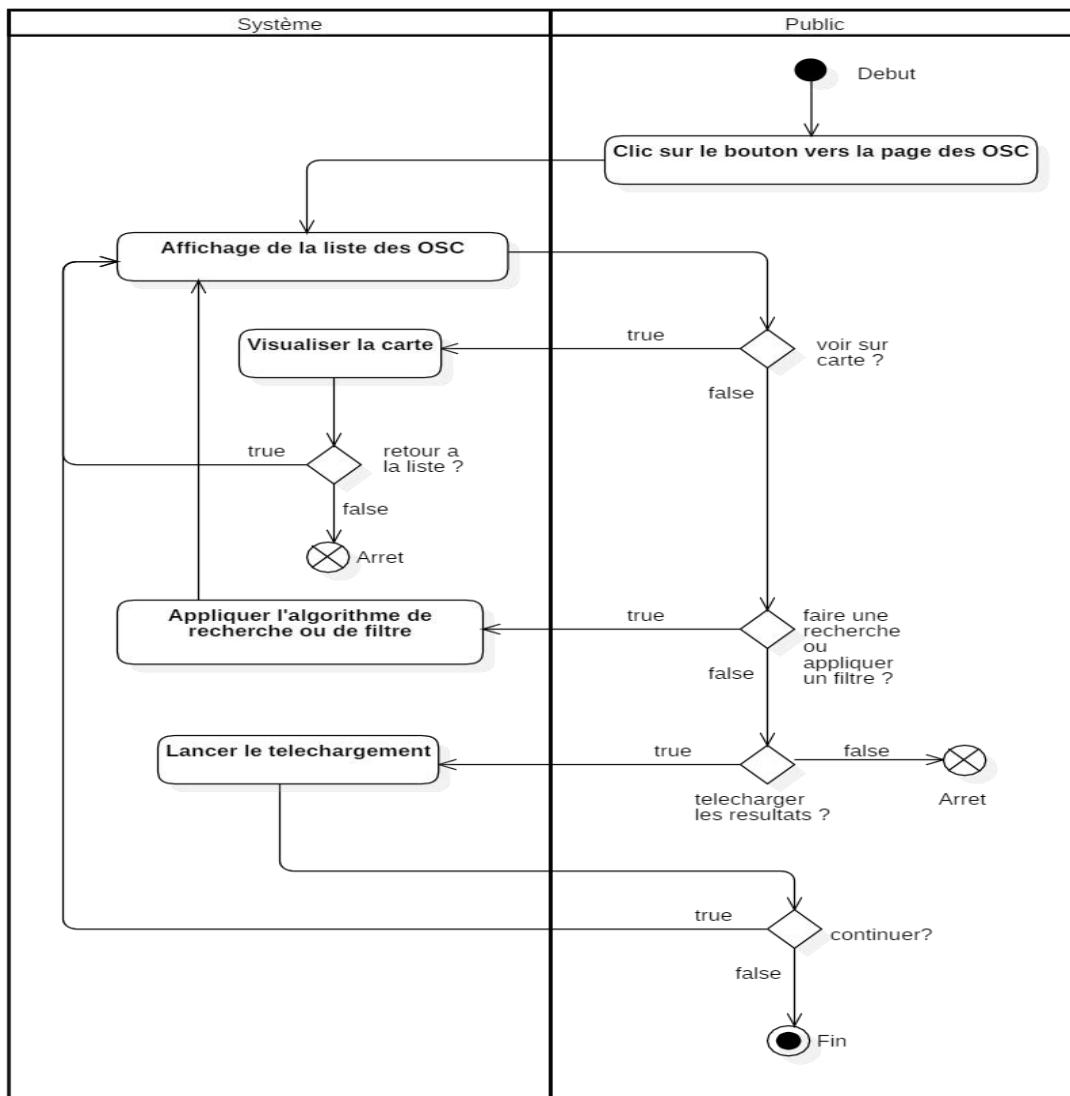
Figure 20 : Diagramme d'activités illustrant l'ajout d'une nouvelle osc par l'administrateur

➤ Mettre à jour un projet :



**Figure 21 : Diagramme d'activités illustrant la mise à jour d'un projet par un manager d'osc**

➤ Visualiser les OSC :



**Figure 22 : Diagramme d'activités illustrant la visualisation des osc par un utilisateur par défaut**

➤ Se déconnecter :

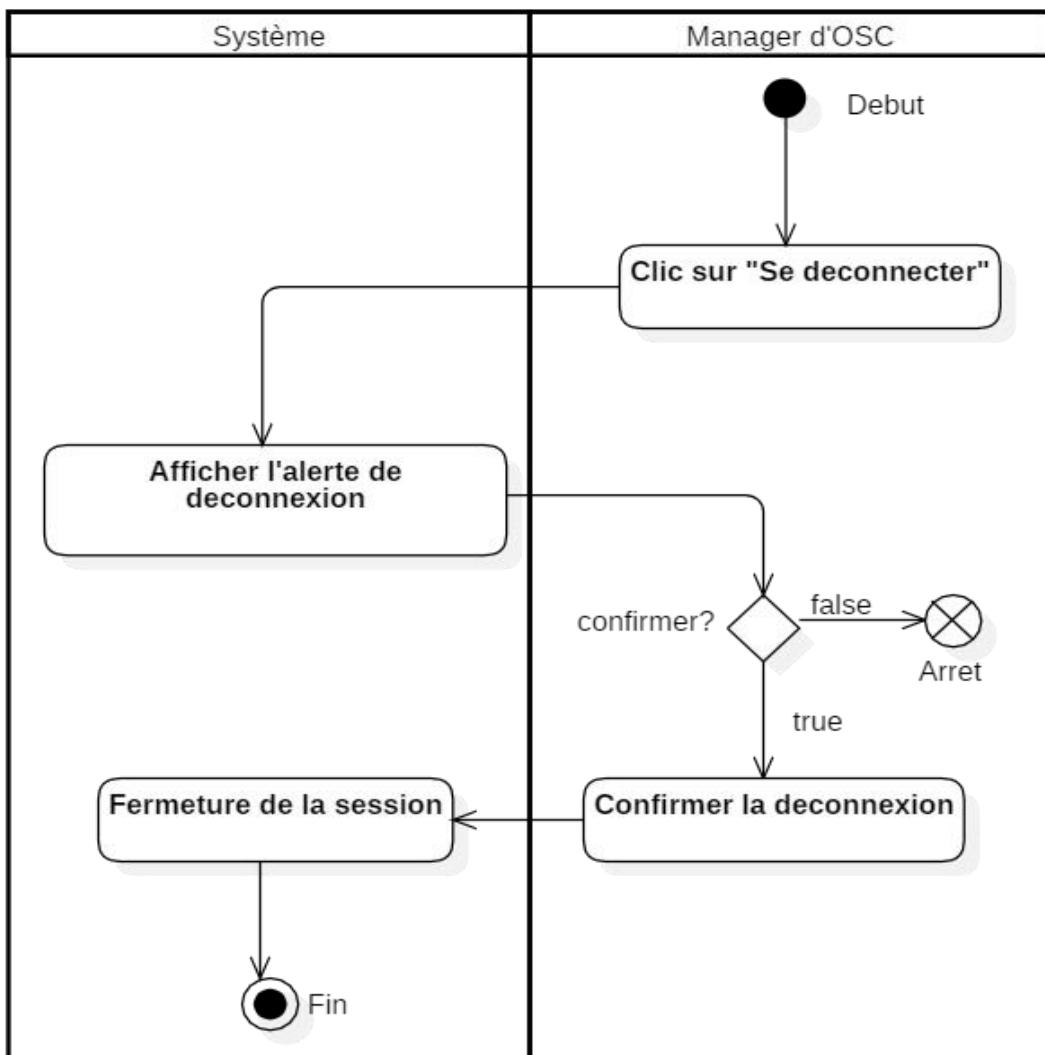
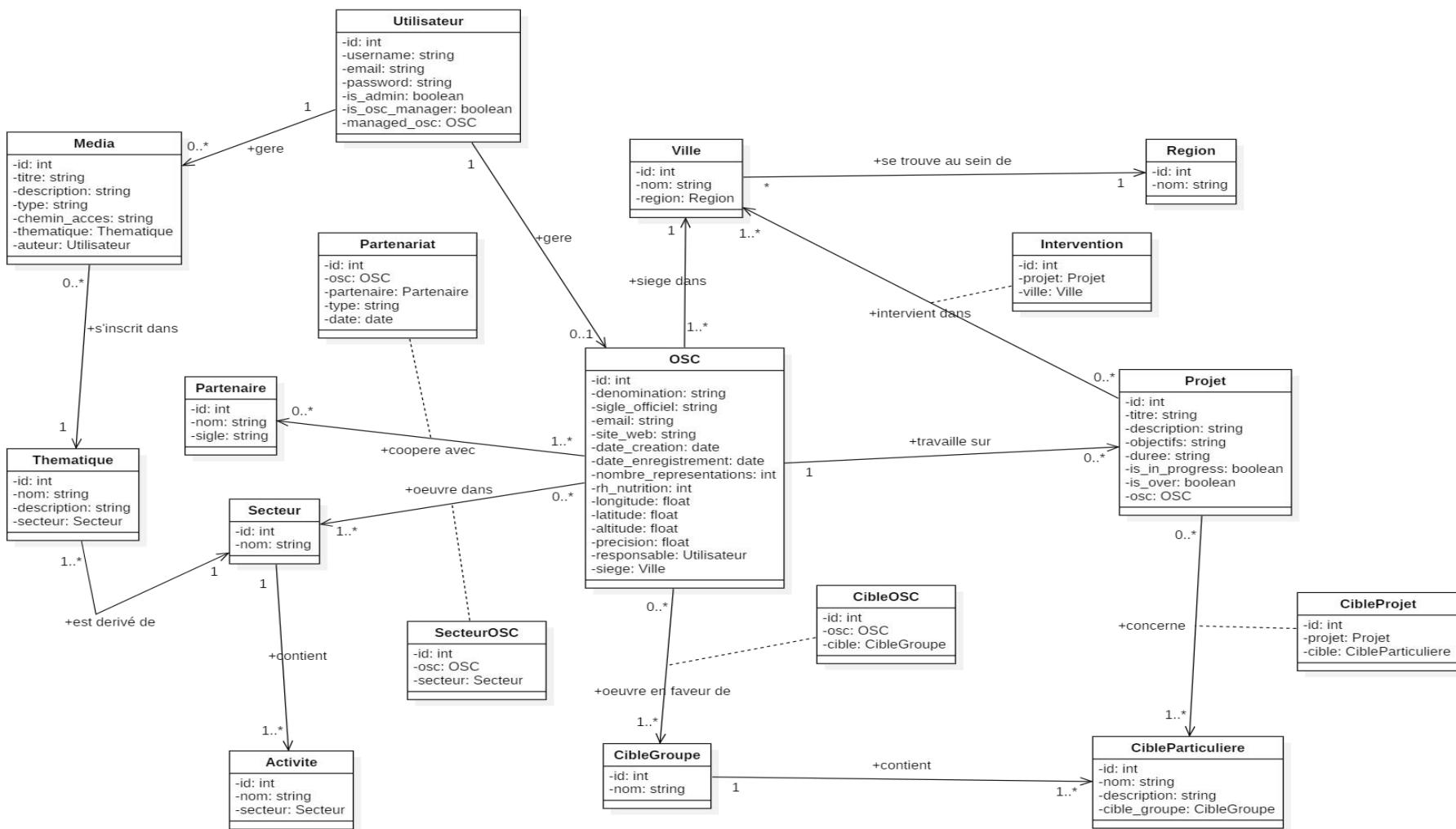


Figure 23 : Diagramme d'activités illustrant la déconnexion d'un manager d'osc

### 3.3.5. Diagramme de classes

Le **diagramme de classes** est utilisé pour représenter la **structure statique du système**. Il montre les différentes **classes de l'application**, leurs **attributs**, leurs **méthodes**, ainsi que les relations entre elles (par exemple, les héritages et les associations).



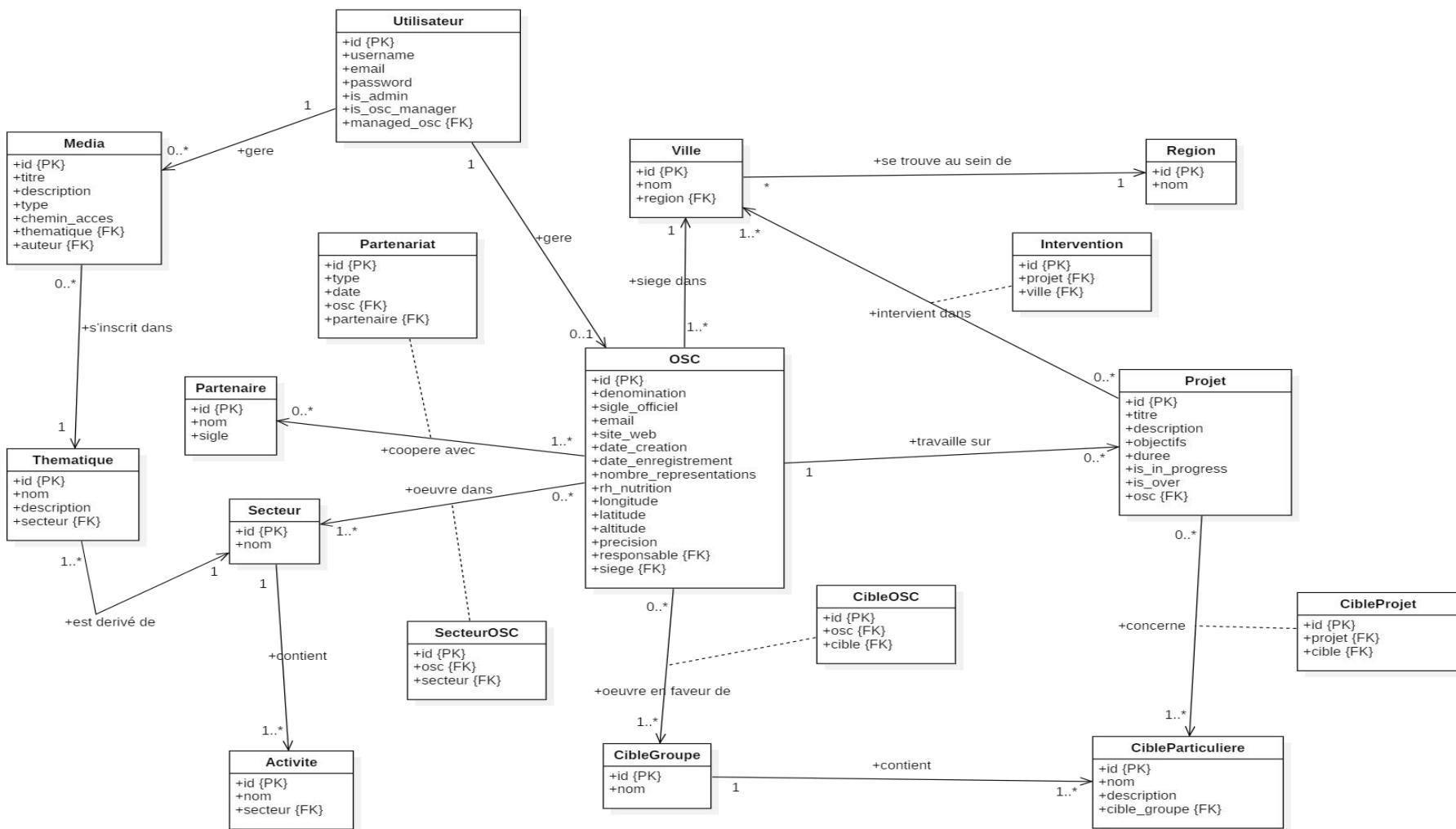
**Figure 24 : Diagramme de classes de l'application web de cartographie des OSC**

### **3.3.6. Modèle Entité – Relationnel**

Le **modèle Entité-Relation (ER)** est utilisé pour représenter la **structure des données**, du système, à *persister*. Il permet de définir les entités (objets ou concepts) de l'application, leurs attributs et les relations entre elles, servant ainsi de base pour la conception de la base de données.

Dans cette application, le formalisme ER a été utilisé pour construire un **Modèle Logique de Données (MLD)** en suivant une approche relationnelle. Le type de MLD adopté est un **Modèle Relationnel de Données (MRD)**, avec la notation UML comme standard graphique pour faciliter la compréhension et la communication des concepts.

Contrairement au **diagramme de classes**, qui se concentre sur la modélisation des composants métiers du système en mettant en avant leurs attributs, leurs méthodes et leurs relations (associations, généralisations), le **modèle Entité-Relation** se focalise exclusivement sur les **données à persister**.



**Figure 25 : Modèle Relationnel de Données**

Ces diagrammes et modèles sont essentiels pour la conception détaillée du système, car ils fournissent une vision complète de son architecture, de son fonctionnement et de la gestion de ses données. Ils serviront également de guide pour le développement et la mise en œuvre du système dans les phases suivantes du projet.

## 3.4. Choix techniques des solutions

### 3.4.1. Étude des choix techniques

Le choix de l'architecture logicielle est un facteur clé pour assurer la performance, la scalabilité et la résilience d'une application web. Dans le cadre de l'application web de cartographie des OSC, cette règle ne fait pas exception. Une étude préalable comparative doit être réalisée afin de déterminer l'architecture la plus adaptée en fonction des besoins spécifiques des commanditaires et bénéficiaires. Dans cette optique, il convient de distinguer les **architectures de déploiement**, qui concernent la répartition physique des composants sur l'infrastructure, et les **patterns architecturaux**, qui concernent l'organisation du code au sein de l'application.

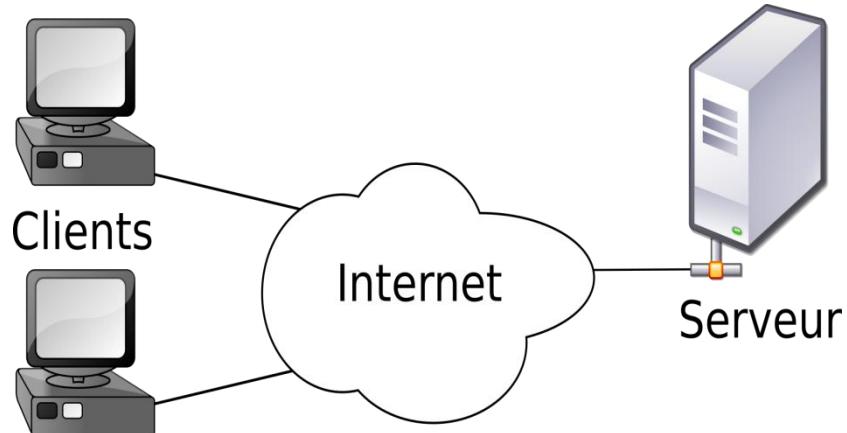
#### 3.4.1.1. Modèle de communication : Client - Serveur

Le modèle **client-serveur** décrit la relation fondamentale entre un **client<sup>1</sup>** et un **serveur<sup>2</sup>**. Ce modèle est présent dans **toutes les architectures modernes**, car il définit la **communication entre les composants** d'une application, qu'il s'agisse de composants distribués ou d'une application monolithique [14].

---

<sup>1</sup> Qui demande des services ou des ressources.

<sup>2</sup> Qui fournit ces services ou ressources.



**Figure 26 : Exemple d'architecture client–serveur : deux clients font leurs requêtes à un serveur via Internet**

L'architecture client-serveur repose sur le principe de la communication réseau, où les clients envoient des requêtes au serveur, qui répond à ces requêtes en fournissant des données ou en exécutant des actions.

Bien que le modèle client-serveur soit une base commune, il peut être intégré de manière différente selon l'architecture de déploiement choisie. Chaque type d'architecture met en œuvre cette relation client-serveur d'une manière spécifique, selon l'organisation des composants et la manière dont ils sont déployés.

### 3.4.1.2. Architecture de déploiement

Voici les principales architectures de déploiement, chacune intégrant le modèle client-serveur de manière spécifique :

- **Architecture Monolithique** [11] : Dans une architecture monolithique, l'application est déployée en tant qu'unité unique, généralement sur un seul serveur. Le **client**<sup>1</sup> envoie des requêtes au **serveur**, qui gère la logique métier, les données et l'interface, souvent dans un même environnement d'exécution. Le modèle client-serveur est intégré de manière simplifiée, où le serveur exécute tout et renvoie les résultats directement au client.

---

<sup>1</sup> Interface utilisateur.

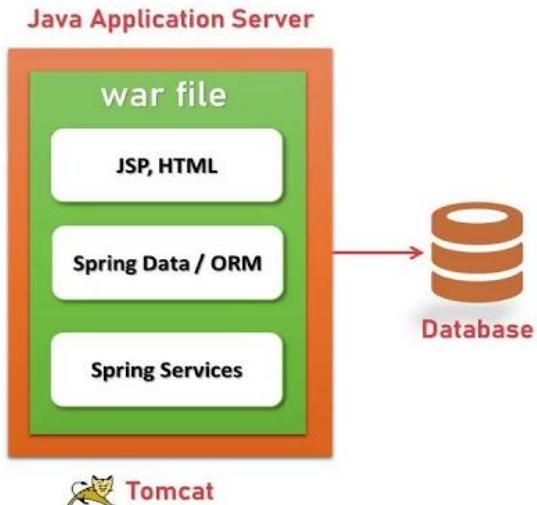


Figure 27 : Architecture Monolithique

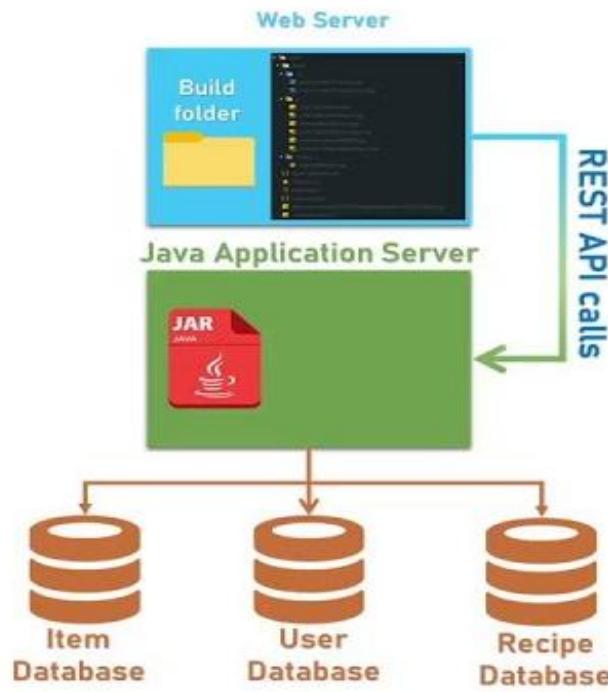
Il existe une variante ; l'**architecture monolithique modulaire**, qui consiste à organiser le code source en modules distincts au sein du monolithe. Chaque module peut avoir ses propres responsabilités et peut être développé de manière indépendante, tout en restant intégré dans un seul projet. Ce modèle permet de mieux structurer le code sans introduire la complexité d'un modèle de microservices.

- **Architecture N-tiers<sup>1</sup>** [13] : Cette architecture sépare les responsabilités en plusieurs couches<sup>2</sup>. Le modèle client-serveur est encore utilisé, mais cette fois-ci, le **client** communique avec un serveur de présentation, qui lui-même interagit avec un serveur de logique métier et un autre pour l'accès aux données. Le modèle client-serveur est donc multiplié entre les différentes couches de l'application.

---

<sup>1</sup> Ou architecture multi-couches.

<sup>2</sup> Par exemple, présentation, logique métier, accès aux données.



**Figure 28 : Architecture N-tiers**

- **Architecture Microservices** [13] : Dans une architecture microservices, l'application est composée de services indépendants, chacun étant responsable d'une fonctionnalité spécifique. Le modèle client-serveur est appliqué entre les **microservices**<sup>1</sup> et les **clients**<sup>2</sup>. Chaque microservice expose une API<sup>3</sup> que les autres microservices ou clients utilisent pour communiquer.

<sup>1</sup> Qui peuvent fonctionner en tant que serveurs.

<sup>2</sup> Qui peuvent être d'autres services ou des utilisateurs finaux.

<sup>3</sup> API : Application Programming Interface. Il s'agit souvent d'une API REST (Representational State Transfer) ou gRPC (Google Remote Procedure Call).

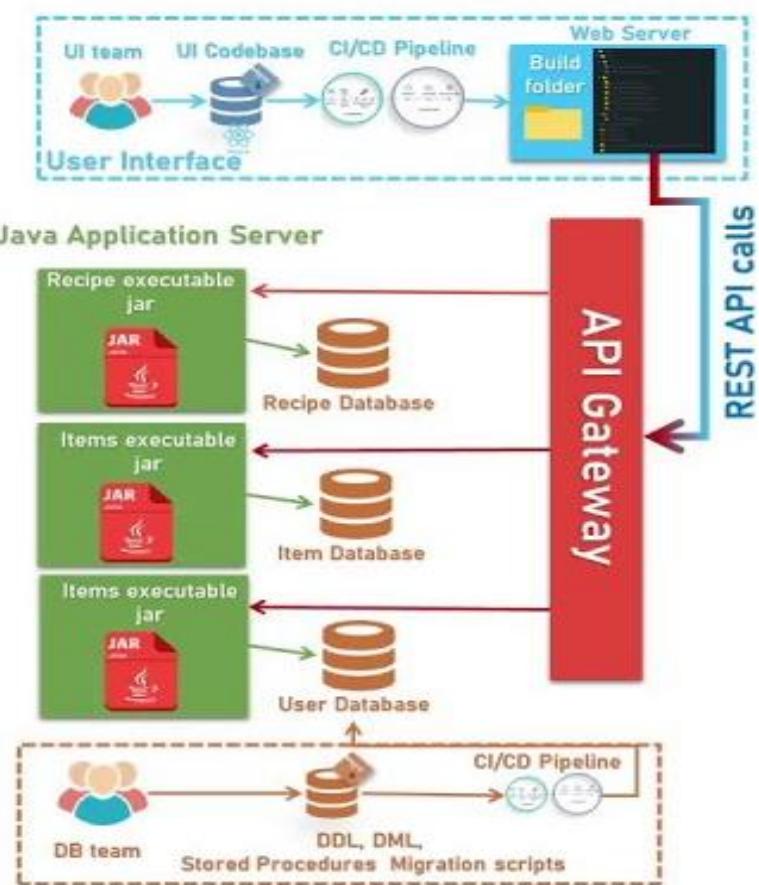
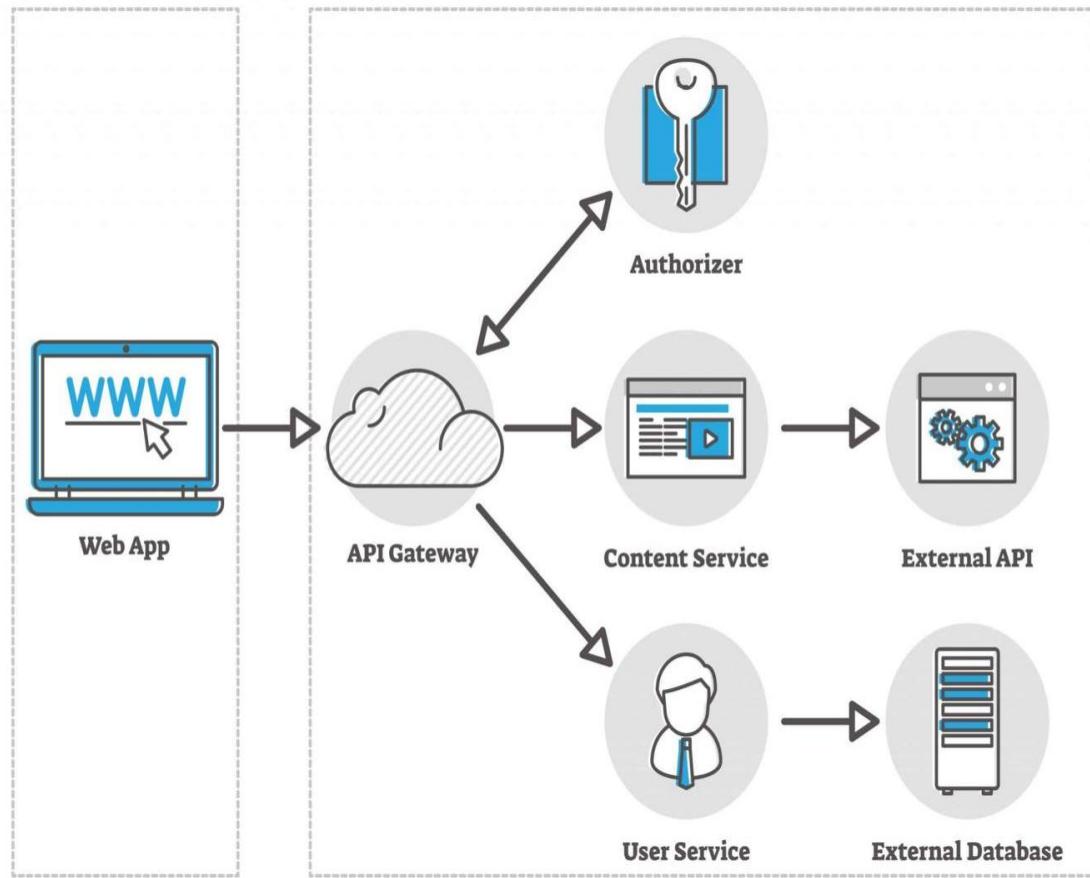


Figure 29 : Architecture Microservices

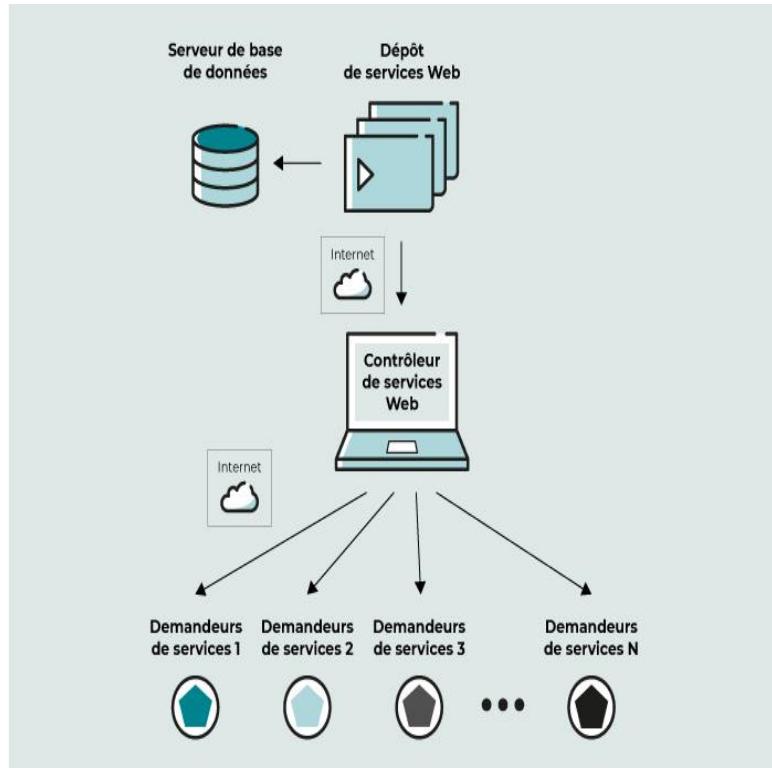
- **Architecture Serverless [12]** : Dans une architecture serverless, les applications sont déployées sous forme de fonctions individuelles exécutées sur une infrastructure cloud. Le modèle client-serveur est toujours présent, mais les **clients** envoient des requêtes aux fonctions hébergées dans le cloud, qui agissent comme des **serveurs**, fournissant des réponses après avoir exécuté des actions sur demande. L'infrastructure est gérée dynamiquement par le fournisseur de services cloud.



**Figure 30 : Architecture Serverless**

- **Architecture SOA (Service-Oriented Architecture)** [13] : L'architecture SOA repose sur l'idée de décomposer l'application en services autonomes, chacun offrant une fonctionnalité bien définie. Les **clients** communiquent avec ces services via des protocoles réseau standard<sup>1</sup>. Le modèle client-serveur est ici largement utilisé pour permettre aux services de fonctionner indépendamment tout en étant accessibles par le client.

<sup>1</sup> Il s'agit souvent de HTTP : HyperText Transfer Protocol (protocole de communication léger et sans état, utilisé principalement pour la transmission de documents et de données sur le Web, notamment entre les clients ; navigateurs, et les serveurs) ou SOAP : Simple Object Access Protocol (protocole de communication basé sur XML, conçu pour permettre l'échange de messages structurés entre les applications via des réseaux, souvent utilisé dans des architectures orientées services (SOA) pour des communications plus complexes et sécurisées).



**Figure 31 : Architecture Orientée Services**

Afin de mieux comprendre et comparer les différentes architectures abordées précédemment, il est utile de les synthétiser dans un tableau récapitulatif. Ce tableau présentera de manière claire et structurée les caractéristiques essentielles de chaque architecture, en mettant en évidence leurs spécificités, avantages et inconvénients, ainsi que les contextes dans lesquels elles sont le plus souvent utilisées. Cela permettra de mieux visualiser les choix possibles en matière d'architecture logicielle, en fonction des besoins spécifiques d'un projet, de ses exigences en termes de scalabilité, de performance, ou de modularité.

Tableau 9 : Récapitulatif des architectures logicielles de déploiement

Architecture	Définition	Description	Avantages	Inconvénients	Exemples
<b>Microservices</b>	Application décomposée en services indépendants	Chaque fonctionnalité est un service autonome, communiquant via des APIs. Chaque service est développé, déployé et mis à jour indépendamment.	<ul style="list-style-type: none"> <li>Scalabilité et flexibilité élevées</li> <li>Déploiement indépendant des services</li> <li>Facilité d'intégration continue</li> </ul>	<ul style="list-style-type: none"> <li>Complexité accrue</li> <li>Difficulté de gestion de la communication entre services</li> <li>Gestion de la sécurité complexe</li> </ul>	Netflix, Uber, Amazon
<b>Monolithique</b>	Application unique avec tous les composants intégrés	Une architecture où toutes les fonctions de l'application sont regroupées dans un seul bloc de code. Les différentes fonctionnalités sont	<ul style="list-style-type: none"> <li>Facilité de développement initial</li> <li>Moins de complexité pour les petites</li> </ul>	<ul style="list-style-type: none"> <li>Faible scalabilité</li> <li>Difficulté à maintenir à mesure que l'application grandit</li> </ul>	Applications bureautiques classiques, systèmes de gestion simples

		fortement couplées.	applications	• Risque de défaillance totale	
N-tiers	Architecture à plusieurs couches distinctes	L'application est divisée en plusieurs couches (par exemple, présentation, logique métier, données). Chaque couche est indépendante des autres.	<ul style="list-style-type: none"> <li>• Déploiement simple</li> </ul> <ul style="list-style-type: none"> <li>• Séparation claire des responsabilités</li> <li>• Facilité de gestion de la complexité</li> <li>• Scalabilité modérée</li> </ul>	<ul style="list-style-type: none"> <li>• Peut devenir complexe à maintenir avec de nombreuses couches</li> <li>• Problèmes de performance entre les couches</li> </ul>	Applications d'entreprise, systèmes ERP
Serverless	Exécution sans gestion explicite des serveurs	L'application s'exécute dans un environnement cloud où l'infrastructure est gérée par le fournisseur (par exemple, AWS Lambda).	<ul style="list-style-type: none"> <li>• Pas besoin de gérer l'infrastructure</li> <li>• Paiement à l'usage</li> <li>• Haute scalabilité</li> </ul>	<ul style="list-style-type: none"> <li>• Dépendance au fournisseur cloud</li> <li>• Latence potentielle pour le démarrage des fonctions</li> <li>• Moins de contrôle sur les</li> </ul>	AWS Lambda, Google Cloud Functions

				ressources	
SOA	Architecture centrée sur les services interconnectés	Un ensemble de services interopérables qui échangent des informations pour accomplir des tâches complexes. Chaque service a une fonctionnalité bien définie.	<ul style="list-style-type: none"> <li>• Flexibilité et réutilisation des services</li> <li>• Intégration facile de systèmes hétérogènes</li> </ul>	<ul style="list-style-type: none"> <li>• Complexité dans l'intégration des services</li> <li>• Nécessite une gestion rigoureuse de la communication entre services</li> </ul>	Services bancaires, intégration d'applications d'entreprise

### 3.4.1.3. Architecture de code

Les **patterns architecturaux** désignent les **modèles de conception** utilisés pour structurer le code d'une application, en séparant les différentes responsabilités et en facilitant sa maintenance et son évolution. Ces patterns influencent la façon dont le code est organisé et comment les différentes parties de l'application interagissent.

Voici quelques patterns architecturaux courants dans le développement d'applications web :

- **MVC<sup>1</sup>** [9] : Ce **pattern** sépare l'application en trois composants principaux :
  - ✓ **Modèle (Model)** : La gestion des données et de la logique métier.
  - ✓ **Vue (View)** : La présentation de l'interface utilisateur.
  - ✓ **Contrôleur (Controller)** : La gestion des interactions de l'utilisateur et la coordination entre le modèle et la vue. Ce pattern est largement utilisé dans les applications web pour organiser les responsabilités et favoriser la séparation des préoccupations.
- **MVT<sup>2</sup>** [10] : C'est une variation du MVC, principalement utilisée dans des frameworks comme **Django**. La différence réside dans la vue, qui est remplacée par le **template**.
  - ✓ **Modèle (Model)** : Comme dans le MVC, il représente la logique métier et les données.
  - ✓ **Vue (View)** : La logique qui contrôle l'affichage des données (généralement les vues dynamiques dans une application web).
  - ✓ **Template (Template)** : Un modèle de code HTML avec des balises spéciales qui sont remplies avec des données dynamiques. Ce pattern est populaire dans les frameworks web Python comme Django, où la vue est en fait une fonction qui génère un template à afficher.
- **MVVM<sup>3</sup>** [8] : C'est un pattern utilisé principalement dans les applications **de bureau** et **mobiles**, mais aussi dans certains environnements web. La **VueModèle** joue ici le rôle d'un intermédiaire entre la **Vue** et le **Modèle**. Il facilite la gestion de la logique de

<sup>1</sup> MVC : Modèle (Model) - Vue (View) - Contrôleur (Controller).

<sup>2</sup> MVT : Modèle (Model) - Vue (View) - Template (Controller).

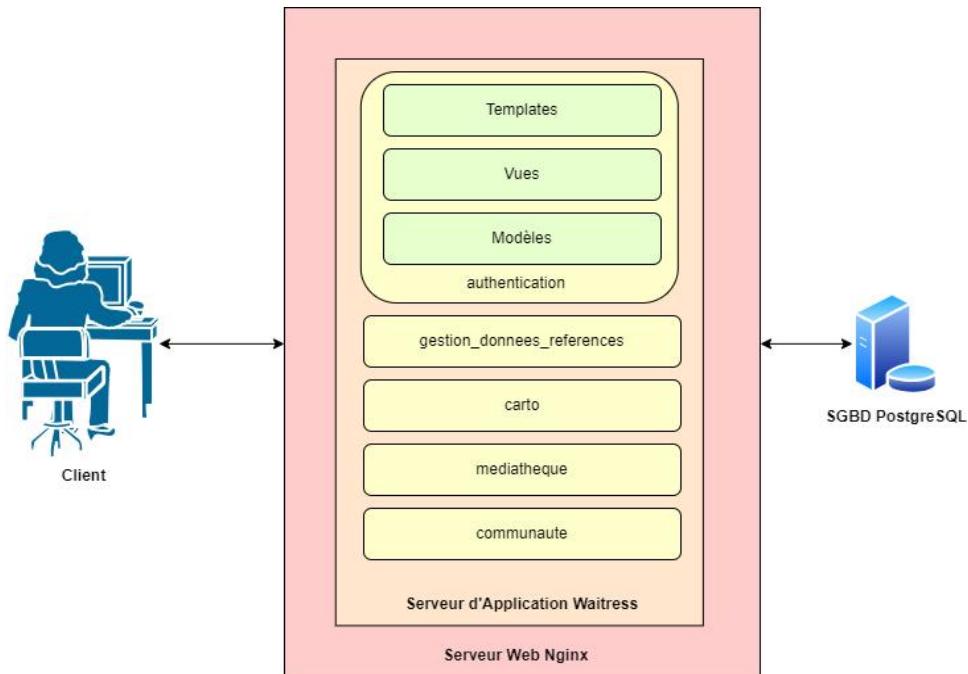
<sup>3</sup> MVVM : Modèle (Model) - Vue (View) - VueModèle (ViewModel).

présentation, en permettant de lier directement les données entre la vue et le modèle via des mécanismes de binding (comme dans les frameworks tel que **Angular** ou **Knockout.js**).

En résumé, l'architecture client-serveur est le modèle de communication fondamental sur lequel reposent de nombreuses architectures modernes. Il est intégré de manière différente dans chaque architecture de déploiement selon les besoins spécifiques de répartition des composants, de scalabilité et de performance. Parallèlement, les patterns architecturaux sont utilisés pour structurer le code de l'application de manière efficace et maintenir une séparation des préoccupations au niveau de la logique métier, de l'interface utilisateur et des données.

### 3.4.2. Choix techniques retenus

Après une analyse approfondie des architectures de déploiement et des patterns architecturaux, le choix s'est porté sur une **architecture monolithique modulaire combinée au pattern MVT**. Cette solution permet de répondre efficacement aux exigences du projet tout en offrant une base flexible pour les évolutions futures.



**Figure 32 : Architecture monolithique modulaire de l'application web de cartographie**

L'**architecture monolithique modulaire** permet de structurer l'application en modules (authentication, gestion \_donnees \_references, carto, mediatheque, communaute) indépendants au sein du monolithe, facilitant ainsi son développement et sa maintenance tout en conservant la simplicité du déploiement d'un monolithe. Ce choix est particulièrement adapté pour des applications de taille moyenne, où la gestion de la complexité ne justifie pas encore la mise en place d'une architecture plus complexe comme les microservices.

Le **pattern MVT** a été choisi pour la structuration du code, car il offre une séparation claire des préoccupations entre la gestion des données (Modèle), la logique de présentation (Template), et l'affichage dynamique des informations (Vue). Ce modèle, particulièrement adapté au framework Django, favorise une bonne organisation du code tout en permettant une évolution rapide de l'application sans sacrifier la maintenabilité.

## Conclusion

Ce chapitre a permis de structurer et de formaliser l'ensemble des aspects conceptuels et techniques du projet. Grâce à l'approche méthodologique adoptée, basée sur **l'analyse orientée objet** et **la modélisation UML**, nous avons pu documenter les spécifications du système et établir une architecture claire et évolutive.

Les différents diagrammes présentés, tels que le diagramme de contexte, les diagrammes de cas d'utilisation, les diagrammes d'activités, le diagramme de classes et le modèle Entité-Relation, offrent une vision complète du système et de la base de données, couvrant à la fois ses fonctionnalités principales et son organisation structurelle. Ces représentations permettent de garantir **une communication efficace** entre les différentes parties prenantes du projet tout en assurant **la cohérence du système**. Les différentes alternatives techniques ont été étudiées et comparées, et un ensemble de choix techniques a été validé pour guider la mise en œuvre du projet.

En conclusion, l'analyse et la conception réalisées dans ce chapitre établissent des bases solides pour le développement de l'application web. Elles assurent une transition fluide vers la phase de réalisation technique, tout en minimisant les risques liés à des imprécisions ou à des incohérences dans les spécifications initiales. Le prochain chapitre se consacrera à la mise en œuvre concrète de cette conception, en abordant les aspects de développement et de test.

# **Chapitre 4 : RÉALISATION ET MISE EN ŒUVRE**

Le chapitre 4 présente les différentes étapes de la réalisation et de la mise en œuvre du projet de développement de l'application web pour la cartographie des Organisations de la Société Civile (OSC) en Nutrition et Alimentation Saine au Togo. Cette phase marque la transition entre la phase d'analyse et de conception et la livraison du produit final. Elle a impliqué **la mise en place des composants matériels et logiciels** nécessaires à la création de l'application, ainsi que **la configuration des environnements de développement, de test et de production**. Ce chapitre détaille également **les outils utilisés, les architectures matérielles et logicielles** adoptées, ainsi que **les processus de déploiement et les mesures de sécurité** mises en place pour assurer la protection des données et des utilisateurs. L'objectif principal étant de garantir une solution stable, performante, et sécurisée qui répond aux attentes des utilisateurs tout en respectant les contraintes du projet.

Il offre également une vue d'ensemble sur les choix technologiques qui ont été faits pour répondre aux besoins spécifiques du projet, ainsi que les défis rencontrés lors de l'implémentation. En outre, elle présente les captures d'interface et de code source pour illustrer les principales fonctionnalités développées, permettant ainsi une meilleure compréhension du fonctionnement de l'application.

## 4.1. Mise en œuvre

La phase de mise en œuvre du projet a consisté à définir et à configurer les ressources nécessaires au développement et au déploiement de l'application web. Cette étape a impliqué la sélection du matériel et des logiciels adéquats pour assurer une performance optimale, une gestion sécurisée des données et une interaction fluide entre les différentes parties du système. L'infrastructure utilisée a été soigneusement choisie en fonction des besoins du projet, en tenant compte des spécifications techniques, de la scalabilité, ainsi que des contraintes liées au déploiement et à la maintenance. Nous détaillerons ci-dessous les caractéristiques des matériels et logiciels utilisés pour cette phase.

### 4.1.1. Les matériels utilisés

Afin d'assurer la bonne exécution des différentes phases du projet, de l'analyse à la mise en production, des équipements performants ont été sélectionnés. Ces matériels ont permis

d'optimiser chaque étape, en garantissant des conditions de travail adaptées aux exigences du projet.

➤ **Phases d'analyse, de conception, de développement et de test :**

La machine (**HP Envy M6**) a été choisie en fonction de ses performances et de sa capacité à exécuter les outils nécessaires à l'analyse et aux tests en passant par la conception et le développement. Les caractéristiques principales sont :

- ✓ **Processeur** : Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50 GHz
- ✓ **Mémoire RAM** : 8.00 Go (7.90 Go utilisable)
- ✓ **Mémoire de stockage** : 119 Go SSD
- ✓ **Type du système** : Système d'exploitation 64 bits, processeur x64
- ✓ **Fonctionnalité tactile/stylet** : Non disponible

Il a donc servi à la fois de machine de développement et de serveur physique de test. Ses spécifications ont permis de mener à bien les tâches de modélisation tout en garantissant une gestion fluide des ressources et des outils nécessaires à la réalisation du projet.

➤ **Phase de mise en production :**

- ✓ **VPS<sup>1</sup>** : Le serveur de déploiement est un VPS, acheté sur **Contabo<sup>2</sup>**, avec les spécifications suivantes :
  - **CPU** : 8 cœurs
  - **Mémoire RAM** : 32 Go
  - **Stockage** : 500 Go SSD
- ✓ **Base de données** : PostgreSQL avec 8 Go de RAM dédié pour le SGBD<sup>3</sup>
- ✓ **Nom de domaine** : Inclus avec un certificat SSL pour garantir la sécurité des échanges.

---

<sup>1</sup> VPS : Virtual Private Server (Serveur Privé Virtuel). C'est un environnement d'hébergement virtuel qui partage les ressources d'un serveur physique tout en offrant une isolation et un contrôle sur le système d'exploitation et les applications. Il combine la flexibilité d'un serveur dédié avec la rentabilité de l'hébergement mutualisé.

<sup>2</sup> Il s'agit d'un fournisseur de services d'hébergement web.

<sup>3</sup> SGBD : Système de Gestion de Bases de Données.

Ces choix matériels ont ainsi permis d'assurer une gestion fluide des ressources tout au long du cycle de développement et de garantir des performances optimales lors de la mise en production de l'application.

#### 4.1.2. Les logiciels utilisés

L'analyse, la conception, le développement et les tests de l'application ont impliqué l'utilisation d'un ensemble de technologies et d'outils adaptés aux besoins du projet. Voici la liste des outils utilisés :

**Tableau 10 : Liste des choix technologiques - Phases d'analyse, conception, développement et tests**

Technologie	Version	Rôle
Bootstrap	5.3.2	Librairie CSS de mise en forme
Chart.js	4.4.7	Librairie JavaScript pour la création des graphs
Django	5.0.7	Framework de développement de l'application
draw.io	24.7.17.0	Création de figures (architecture logicielle monolithique, diagramme de contexte)
Git	2.42.0.windows.2	Environnement de versioning de fichiers
GitLab	-	Plateforme web, de gestion collaborative de code, basée sur Git
jQuery	3.7.1	Librairie JavaScript de développement front-end ; il permet de dynamiser l'interface grâce à ses fonctions
Leaflet.js	1.9.4	Librairie JavaScript pour la carte

		interactive
<b>Nginx</b>	1.26.2	Serveur web de l'environnement de test
<b>PostgreSQL</b>	16	Système de gestion de base de données relationnelles
<b>Python</b>	3.12.6	Langage de programmation et base de Django
<b>StarUML</b>	3.0.2.0	Création des modélisations UML
<b>Visual Studio Code</b>	1.94.2	Environnement de développement intégré
<b>Waitress</b>	3.0.0	Serveur d'application WSGI <sup>1</sup>
<b>Windows</b>	10 Professionnel 22H2	Système d'exploitation
<b>WPS Office</b>	12.2.0.19307	Écriture des manuel et guide, création des documents de présentation

Ces outils ont permis de développer une application robuste, responsive<sup>2</sup> et sécurisée, tout en facilitant la gestion du code et des versions et la production de livrables de présentation et documentation.

Au delà de ça, il s'agissait choisir un ensemble de logiciels adéquats pour la mise en production sur le VPS afin de garantir de bonnes performances, une haute disponibilité et réactivité. Pour ce faire, les logiciels utilisés sont :

---

<sup>1</sup> WSGI : Web Server Gateway Interface. C'est un standard Python pour les serveurs web et les applications Python. Il permet de faire l'interface entre un serveur web (comme Nginx ou Apache) et une application Python (comme Django ou Flask). WSGI assure que les requêtes HTTP sont envoyées du serveur vers l'application et que les réponses sont renvoyées du serveur vers le client. Il permet de déployer efficacement des applications Python sur le web en respectant un format standard.

<sup>2</sup> Se dit d'un logiciel qui s'adapte automatiquement aux dimensions de l'écran de l'utilisateur que ce soit une ordinateur, un smartphone ou encore une tablette.

**Tableau 11 : Liste des choix technologiques - Phase de déploiement**

Technologie	Version	Rôle	Matériel hôte
<b>Docker</b>	27.4.0	Plateforme de conteneurisation qui permet de déployer des applications dans des conteneurs légers et isolés	<b>VPS</b>
<b>Git</b>	2.25.1	Environnement de versioning de fichiers	<b>VPS</b>
<b>Image PostgreSQL</b>	16.6	Image Docker qui contient l'installation et la configuration de PostgreSQL, prête à être déployée dans un conteneur Docker	<b>VPS</b>
<b>Nginx</b>	1.18.0	Serveur web	<b>VPS</b>
<b>PuTTY</b>	0.82.0.0	Client SSH <sup>1</sup> et Telnet. Il permet de se connecter au VPS de manière sécurisée pour administrer ou interagir avec lui via la ligne de commande	<b>HP Envy M6</b>
<b>Python</b>	3.12.8	Langage de programmation et base de Django	<b>VPS</b>
<b>Ubuntu</b>	20.04.6 LTS	Système d'exploitation du VPS	<b>VPS</b>
<b>WinSCP</b>	6.3.6.0	Client graphique permettant de transférer des fichiers	<b>HP Envy M6</b>

<sup>1</sup> SSH : Secure Shell.

		entre la machine de développement et le VPS distant via les protocoles <b>FTP<sup>1</sup></b> et <b>SFTP<sup>2</sup></b>	
--	--	--	--

#### 4.1.3. Architectures matérielles et logicielles

Dans cette section, nous détaillerons l'architecture matérielle et logicielle mise en œuvre pour le projet, en expliquant comment les différents composants matériels et logiciels ont été orchestrés pour garantir une exécution fluide du processus de développement, de test et de production :

##### ➤ Architecture matérielle

Le projet repose sur une architecture qui combine des ressources locales pour le développement et un hébergement sur un **VPS** distant pour la mise en production. La machine de développement, un **HP Envy M6**, a joué un rôle central tout au long des phases d'analyse, de conception, de développement et de test et même de mise en production. Elle a été utilisée pour héberger les outils de modélisation et de développement, ainsi que pour exécuter les tests unitaires et d'intégration puis a été utilisée pour administrer le VPS à distance.

Le serveur VPS utilisé pour la mise en production offre des performances robustes et scalables, rappelons le, avec un CPU 8 cœurs et 32 Go de RAM. Il permet de garantir une gestion optimale des ressources pour l'application web une fois mise en ligne, tout en offrant la flexibilité nécessaire à son évolutivité future. Les données sont hébergées sur un stockage SSD de 500 Go, assurant des temps de réponse rapides et une bonne disponibilité des services.

##### ➤ Architecture logicielle

Le projet repose sur plusieurs couches logicielles et technologiques complémentaires. **Visual Studio Code** a été choisi pour le développement, notamment pour l'écriture, la correction, et l'exécution du code source de l'application web. Ce dernier est couplé avec le SGBD **PostgreSQL**, qui gère efficacement les données liées aux OSC, aux projets et aux utilisateurs.

---

<sup>1</sup> FTP : File Transfer Protocol.

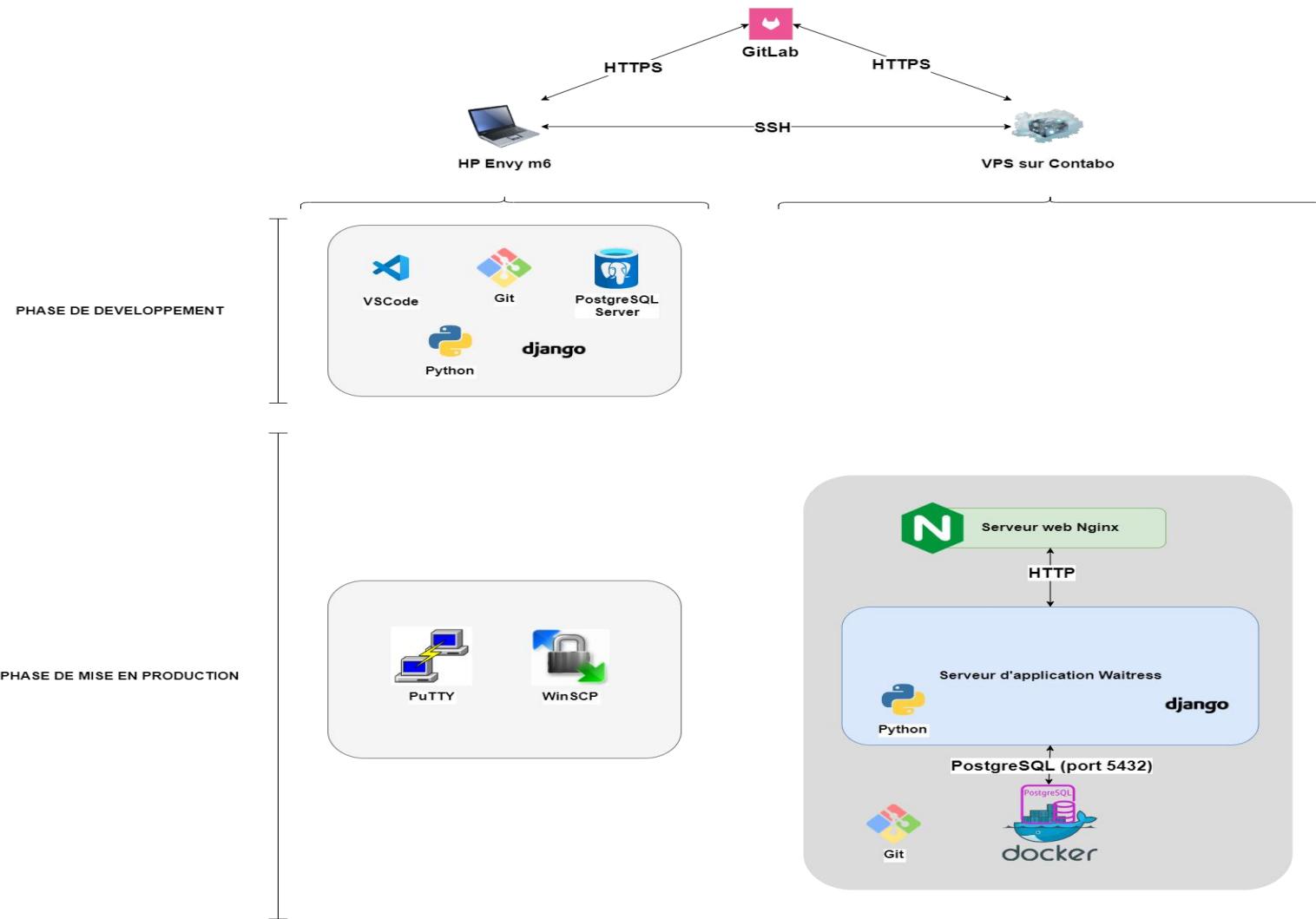
<sup>2</sup> SFTP : SSH File Transfer Protocol.

**Git** et **GitLab** ont été utilisés pour le versioning et la gestion collaborative du code source, assurant un suivi rigoureux des différentes versions du projet.

L'intégration entre le serveur de développement local et le VPS se fait par le biais de **PuTTY**, pour les connexions **SSH**, et **WinSCP**, pour le transfert sécurisé des fichiers. Ces outils facilitent une administration fluide et sécurisée entre les environnements local et distant.

Pour la mise en production, **Nginx** est utilisé comme **serveur web** pour gérer les requêtes entrantes, assurer l'équilibrage de charge et distribuer efficacement les ressources du serveur. En parallèle, **Waitress**, un serveur WSGI, assure la gestion des applications **Python/Django** en production, garantissant une exécution stable et performante.

En résumé, cette combinaison d'outils assure une gestion optimale des processus de développement, de test, et de production, tout en offrant une architecture logicielle robuste et évolutive.



**Figure 33 : Schéma d'illustration des architectures matérielle et logicielle**

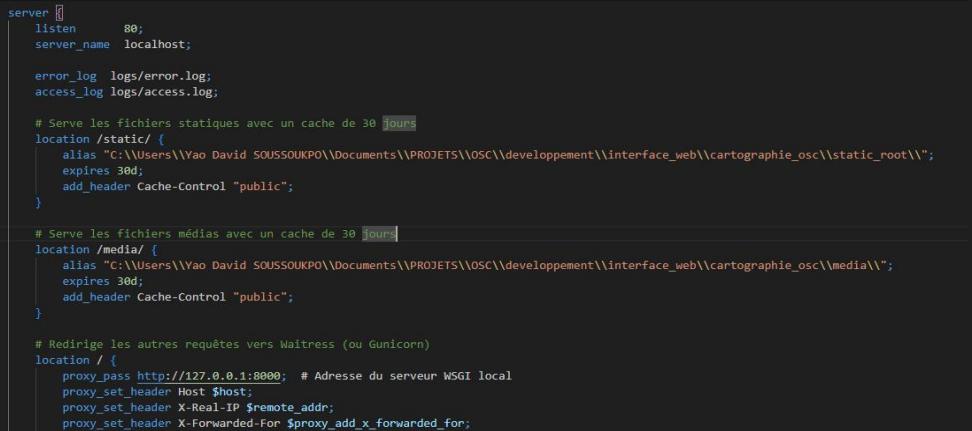
#### 4.1.4. Déploiement des applications

L'application a été déployée en local pour les phases de développement et de test, à l'aide de **Nginx** comme serveur web et **Waitress** pour servir l'application Django. Ce déploiement a permis de simuler un environnement de production à petite échelle, avec une gestion efficace des requêtes et des ressources.

#### ➤ Environnement de test sur la machine locale

Lors des phases de développement et de test, l'application a été déployée localement sur un environnement contrôlé afin de simuler un contexte de production. La configuration s'est déroulée comme suit :

- ✓ **Configuration de Nginx** : Il a été configuré en tant que serveur web inverse (reverse proxy) pour gérer les requêtes HTTP



The screenshot shows a code editor window with the file "nginx.conf" open. The code defines an Nginx configuration for a local host. It includes sections for static files (aliasing "static/" and "media/" to local paths) and proxying requests to a local WSGI server (port 8000). The configuration uses standard Nginx syntax with directives like listen, server\_name, error\_log, access\_log, alias, expires, add\_header, proxy\_pass, proxy\_set\_header, and proxy\_add\_x\_forwarded\_for.

```
File Edit Selection View Go Run ... ↺ → 🔍 nginx-1.26.2 08 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41

nginx.conf ×
conf > nginx.conf
7 http {
11   server {
12     listen      80;
13     server_name localhost;
14
15     error_log  logs/error.log;
16     access_log logs/access.log;
17
18     # Serve les fichiers statiques avec un cache de 30 jours
19     location /static/ {
20       alias "C:\\Users\\Yao David SOUSSOUKPO\\Documents\\PROJETS\\OSC\\developpement\\interface_web\\cartographie_osc\\static_root\\";
21       expires 30d;
22       add_header Cache-Control "public";
23     }
24
25     # Serve les fichiers médias avec un cache de 30 jours
26     location /media/ {
27       alias "C:\\Users\\Yao David SOUSSOUKPO\\Documents\\PROJETS\\OSC\\developpement\\interface_web\\cartographie_osc\\media\\";
28       expires 30d;
29       add_header Cache-Control "public";
30     }
31
32     # Redirige les autres requêtes vers Waitress (ou Gunicorn)
33     location / {
34       proxy_pass http://127.0.0.1:8000; # Adresse du serveur WSGI local
35       proxy_set_header Host $host;
36       proxy_set_header X-Real-IP $remote_addr;
37       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
38       proxy_set_header X-Forwarded-Proto $scheme;
39     }
40   }
41 }
```

**Figure 34 : Configuration de Nginx pour l'environnement de test**

- **Serveur Waitress** : Il est utilisé depuis le terminal pour exécuter l'application Django.

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "INTERFACE\_WEB" with files like "cartographie\_osc", "manage.py", ".gitignore", ".gitlab-ci.yml", "note.txt", "README.md", and "requirements.txt".
- Terminal View:** Displays the command "waitress-serve --port=8000 cartographie\_osc.wsgi:application" being run, with the output "INFO:waitress:Serving on http://0.0.0.0:8000".
- Status Bar:** Shows the current file is "settings.py", the path is "C:\Users\Yao David SOUSSOUKPO\Documents\PROJETS\OSC\developpement\interface\_web", and the Python version is 3.12.6 ("env: venv").

Figure 35 : Exécution dans le terminal du serveur waitress

✓ **Accès via le navigateur :**

- Une fois configuré, l'application a été accessible localement via l'URL '<http://localhost>'.



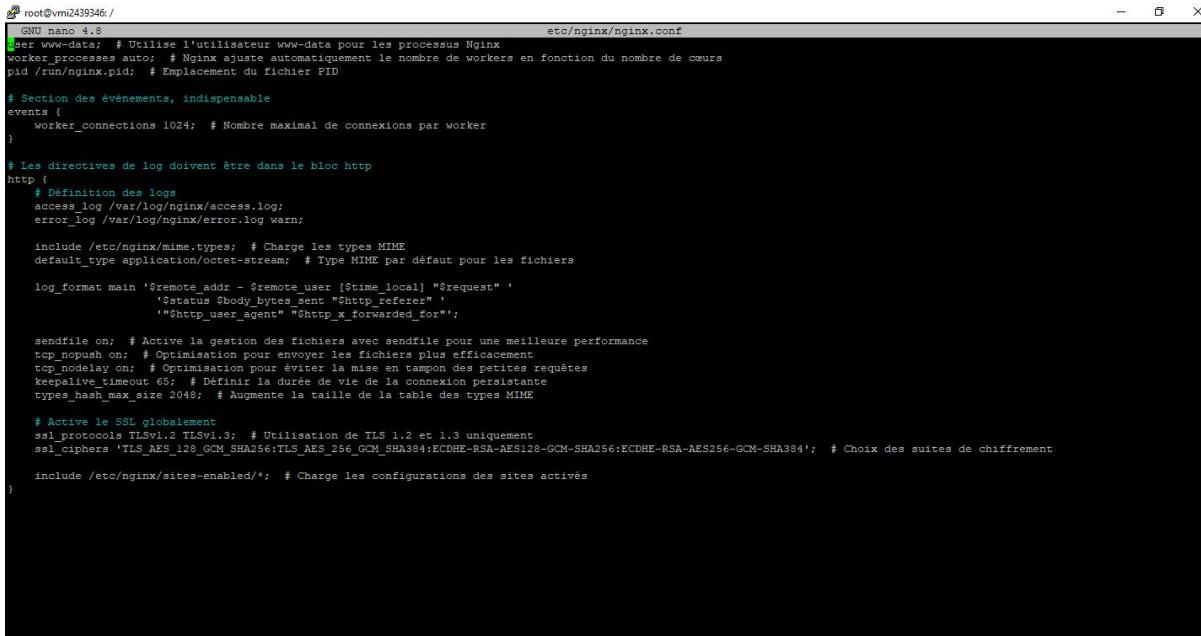
Figure 36 : Accès à l'application à l'adresse <http://localhost>

- Des tests ont été effectués pour valider le bon fonctionnement des fonctionnalités clés.

## ➤ Mise en production sur le VPS

Pour la mise en production, l'application a été déployée sur le VPS avec des configurations presque identiques.

- ✓ **Configuration de Nginx :** Deux fichiers de configuration [complémentaires] similaires à celui de l'environnement local ont été utilisés, adaptés pour une exposition publique sur le port **443**<sup>1</sup>.



```

root@vmi2439346: ~
GNU nano 4.8
/etc/nginx/nginx.conf
user www-data; # Utilise l'utilisateur www-data pour les processus Nginx
worker_processes auto; # Nginx ajuste automatiquement le nombre de workers en fonction du nombre de coeurs
pid /run/nginx.pid; # Emplacement du fichier PID

# Section des événements, indispensable
events {
    worker_connections 1024; # Nombre maximal de connexions par worker
}

# Les directives de log doivent être dans le bloc http
http {
    # Définition des logs
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log warn;

    include /etc/nginx/mime.types; # Charge les types MIME
    default_type application/octet-stream; # Type MIME par défaut pour les fichiers

    log_format main '$remote_addr - $remote_user [$time_local] "$request"
                      $status $body_bytes_sent "$http_referer"
                      "$http_user_agent" "$http_x_forwarded_for"';

    sendfile on; # Active la gestion des fichiers avec sendfile pour une meilleure performance
    tcp_nopush on; # Optimisation pour envoyer les fichiers plus efficacement
    tcp_nodelay on; # Optimisation pour éviter la mise en tampon des petites requêtes
    keepalive_timeout 65; # Définir la durée de vie de la connexion persistante
    types_hash_max_size 2048; # Augmente la taille de la table des types MIME

    # Active le SSL globalement
    ssl_protocols TLSv1.2 TLSv1.3; # Utilisation de TLS 1.2 et 1.3 uniquement
    ssl_ciphers 'TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:ECDHE-RSA-AES256-GCM-SHA384'; # Choix des suites de chiffrement
    include /etc/nginx/sites-enabled/*; # Charge les configurations des sites activés
}

```

**Figure 37 : Configuration globale<sup>2</sup> de Nginx sur le VPS**

<sup>1</sup> C'est le port par défaut utilisé pour les connexions HTTPS. Le serveur web Nginx a été configuré pour permettre les échanges sécurisés via HTTPS.

<sup>2</sup> Ce fichier est la configuration principale de Nginx. Il définit les paramètres globaux comme le nombre de connexions, les logs, les workers, etc. Il inclut souvent d'autres fichiers pour la gestion de sites spécifiques.

```

root@vmi2439346: ~
GNU nano 4.8
server {
    listen 80;
    server_name 161.97.66.8;

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl; # Utilise le port 443 pour HTTPS
    server_name 161.97.66.8;

    # Emplacement des logs
    error_log /var/log/nginx/cartographie_osc_error.log;
    access_log /var/log/nginx/cartographie_osc_access.log;

    # Définitions SSL
    ssl_certificate /etc/nginx/ssl/161.97.66.8.crt;
    ssl_certificate_key /etc/nginx/ssl/161.97.66.8.key;

    # Serve les fichiers statiques avec un cache de 30 jours
    location /static/ {
        alias /var/www/cartographie_osc/static_root/;
        expires 30d;
        add_header Cache-Control "public";
    }

    # Serve les fichiers médias avec un cache de 30 jours
    location /media/ {
        alias /var/www/cartographie_osc/media/;
        expires 30d;
        add_header Cache-Control "public";
    }

    # Redirige les autres requêtes vers Waitress
    location / {
        proxy_pass http://127.0.0.1:8000; # Adresse du serveur waitress WSGI local
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

```

**Figure 38 : Configuration spécifique<sup>1</sup> à l'application web**

✓ **Services de gestion des processus :**

- **waitress-cartographie.service** : Service pour lancer automatiquement le serveur **Waitress** lors du démarrage du VPS.
  - **nginx.service** : Service pour lancer automatiquement le serveur **Nginx** lors du démarrage du VPS.
- ✓ **Accès via le navigateur** : L'application a été rendue accessible publiquement via l'adresse IP du VPS.

---

<sup>1</sup> Ce fichier contient la configuration spécifique à l'application Django. Il est inclus dans nginx.conf, [généralement] via une directive comme "**include /etc/nginx/sites-enabled/\***". Il définit comment Nginx doit gérer les requêtes pour l'application, en précisant par exemple : le domaine ou l'IP associée, le proxy vers [Gunicorn ou] Waitress (WSGI), les réglages des certificats SSL, les règles pour servir les fichiers statiques.



**Figure 39 :** Accès à l'application à l'adresse <https://161.97.66.8/><sup>1</sup>

En résumé, la stratégie de déploiement a permis de maintenir un environnement de test réaliste tout en assurant une transition fluide vers un environnement de production stable et sécurisé. L'utilisation conjointe de Nginx et Waitress a contribué à une gestion optimale des requêtes et des performances.

#### 4.1.5. Sécurité des applications

La sécurité des applications est un enjeu majeur tout au long du cycle de vie du développement. Afin de protéger les données sensibles, assurer la confidentialité des utilisateurs et prévenir les attaques potentielles, des mesures de sécurité adaptées ont été mises en place à chaque étape du projet, de la phase de conception à la mise en production.

Les dispositions prises sont présentées dans le tableau suivant :

**Tableau 12 : Récapitulatif des dispositions de sécurité mises en place**

<sup>1</sup> L'acquisition du nom de domaine n'étant pas terminée au moment de la rédaction de ce mémoire, l'application est accessible via l'adresse IP du VPS sur le port 443. Toutefois, le navigateur affiche une alerte de sécurité malgré l'utilisation de HTTPS. Cela est dû à l'absence d'un certificat SSL valide émis pour un nom de domaine. En effet, un certificat SSL est généralement associé à un domaine et non à une adresse IP. De plus, si un certificat auto-signé est utilisé, le navigateur ne peut pas en vérifier l'authenticité, ce qui entraîne l'affichage de l'avertissement.

Phase/Niveau de sécurité	Matériel	Logiciel
Analyse, Conception, Développement, Test	<ul style="list-style-type: none"> <li>Mot de passe d'accès à l'ordinateur de développement</li> <li>Verrouillage automatique de l'écran après un temps d'inactivité</li> </ul>	<ul style="list-style-type: none"> <li>Dépôt GitHub privé</li> <li>Mise à jour régulière du système et des logiciels</li> <li>Chiffrement des données sensibles</li> </ul>
Production	<ul style="list-style-type: none"> <li>Authentification via SSH sur le VPS</li> <li>Pare-feu <b>UFW</b> bloquant les <b>trafics entrants</b> vers tous les ports sauf <b>443, 22 (SSH)</b> et d'autres ports utiles au système.</li> <li>Configuration de <b>Fail2ban</b> pour bannir temporairement les IPs associées à des tentatives d'authentification infructueuses et successives</li> </ul>	<ul style="list-style-type: none"> <li>Gestion des rôles et permissions</li> <li>Journalisation des accès</li> <li>Sauvegardes régulières</li> <li>Mise à jour régulière du système et des logiciels</li> </ul>

En résumé, des mesures de sécurité robustes ont été mises en place à chaque phase du projet, de la conception à la production, afin de garantir la confidentialité, l'intégrité et la disponibilité des données, ainsi que la sécurité des échanges via le réseau.

## 4.2. Création de la base de données : Script de création ou ORM ?

Dans le cadre de cette application développée avec Django, bien qu'un **Modèle Relationnel de Données (MRD)**<sup>1</sup> ait été conçu lors des phases d'analyse et de conception, la création de la base de données n'a pas nécessité la classique écriture manuelle d'un script SQL.

En effet, Django utilise un **ORM**<sup>2</sup> qui permet de définir les modèles de données directement en Python. La base de données a été générée automatiquement lors de l'exécution de la commande suivante :

```
```bash
```

```
python manage.py migrate
```

```
``
```

Cette commande a :

- ✓ créé automatiquement les tables correspondant aux modèles définis dans le code Python ;
- ✓ appliqué les contraintes et relations définies dans les modèles Django ;
- ✓ généré une structure de base de données conforme au MRD prévu, sans nécessiter de script SQL explicite.

La prise en compte des modifications ultérieures des modèles a été gérée par :

```
```bash
```

```
python manage.py makemigrations
```

```
python manage.py migrate
```

```
``
```

Cette approche a garanti la synchronisation entre le code applicatif et la structure de la base de données tout en facilitant les évolutions et la maintenance du schéma.

---

<sup>1</sup> Voir *supra*, Chapitre III, Section 3.4.6, Page 65.

<sup>2</sup> ORM : Object-Relational Mapping. C'est un concept en informatique qui permet de relier les objets d'un langage de programmation orientée objet (par exemple, ici **Python**, Java, C#) aux tables d'une base de données relationnelle (comme MySQL, ici **PostgreSQL**, SQLite).

## 4.3. Quelques captures des interfaces et codes sources des applications

Cette section présente des extraits visuels de l'application web afin d'illustrer les différentes interfaces et parties du code source utilisées lors du développement.

### 4.3.1. Captures des interfaces

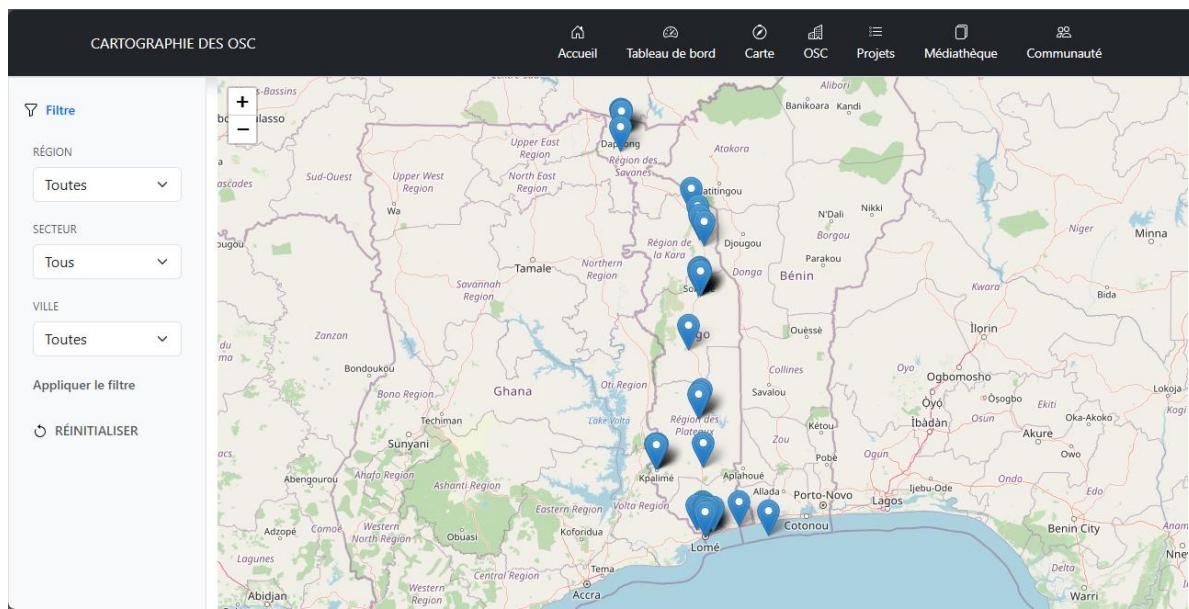
Les captures suivantes montrent les principales interfaces de l'application web, reflétant l'expérience utilisateur et l'ergonomie mise en place :

- ✓ **Page d'accueil** : Présentation générale du projet et accès rapide aux fonctionnalités principales.



Figure 40 : Page d'accueil de l'application

- ✓ **Carte interactive des OSC** : Visualisation géographique des OSC avec filtres dynamiques.



**Figure 41 : Page de la carte interactive**

- ✓ **Tableau de bord** : Interface de visualisation des graphiques liés à la répartition des OSC, projets et personnels des OSC.



**Figure 42 : Page du tableau de bord**

- ✓ **Formulaire de gestion des OSC** : Interface permettant aux managers d'OSC de gérer leurs informations.

The screenshot shows the 'CARTOGRAPHIE DES OSC' platform. At the top, there is a navigation bar with icons for Accueil, Tableau de bord, Carte, OSC, Projets, Médiathèque, Communauté, and a user profile icon. Below the navigation bar, the title 'AREJ' is displayed, followed by the subtitle 'Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse'. On the left side, there is a sidebar titled 'PARAMETRES' containing links for 'AREJ', 'Projets', and 'Média'. It also includes a 'Changer le logo' button and a 'Se déconnecter' link. The main content area is titled 'Informations' and contains fields for 'Sigle' (AREJ), 'RH nutrition' (0), 'Email' (None), 'Site web' (None), 'Siège' (Cinkassé), 'Représentations' (1), 'Dénomination' (Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse), and a 'Responsable' section. A yellow 'Mettre à jour' button is located at the top right of the information form.

**Figure 43 : Page de gestion des informations d'une OSC<sup>1</sup> par son manager**

- ✓ **Page de connexion :** Interfaces d'authentification.

The screenshot shows a login page titled 'Identifiez-vous'. It features two input fields: 'Nom d'utilisateur' containing 'arej\_manager' and 'Mot de passe' containing '\*\*\*\*\*'. Below the inputs is a blue 'Connexion' button. Underneath the button, there is a link 'Pas encore membre ? [Nous rejoindre maintenant !](#)'. At the bottom of the page, there is a copyright notice '© EU4SUN - Togo 2024'.

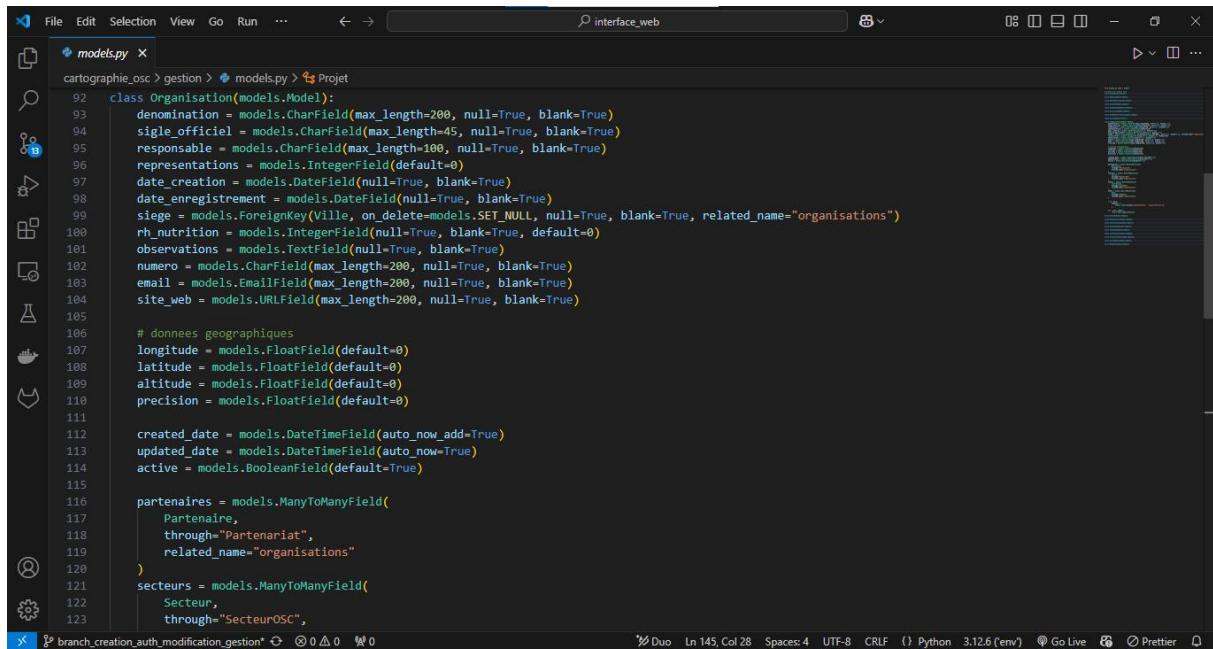
**Figure 44 : Page de connexion**

<sup>1</sup> Dans ce cas, AREJ : Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse.

### 4.3.2. Captures de codes sources

Les extraits de code suivants mettent en avant la structure et l'organisation du code source de l'application :

- ✓ **Un Modèle Django (models.py) :** Définition de l'entité Organisation (il s'agit du modèle qui représente une OSC).



The screenshot shows a code editor window with the file 'models.py' open. The code defines a class 'Organisation' that inherits from 'models.Model'. It contains various fields such as 'denomination', 'sigle\_officiel', 'responsable', 'representations', 'date\_creation', 'date\_enregistrement', 'siege', 'rh\_nutrition', 'observation', 'numero', 'email', 'site\_web', and geographical coordinates ('longitude', 'latitude', 'altitude', 'precision'). It also includes fields for timestamps ('created\_date', 'updated\_date') and a boolean field 'active'. Two many-to-many relationships are defined: 'partenaires' (through 'Partenariat', related\_name 'organisations') and 'secteurs' (through 'SecteurOSC', related\_name 'organisations'). The code is color-coded for syntax highlighting.

Figure 45 : Définition d'un modèle Django

- ✓ **Une Vue Django (views.py) :** Définition de la vue de présentation des OSC.

The screenshot shows a Microsoft Visual Studio Code window with the following details:

- File Explorer:** On the left, it shows a tree view of files and folders, with 'views.py' selected.
- Code Editor:** The main area contains Python code for a Django application. The code defines several functions: `root_entry`, `summary`, `get_organisations`, `get_projects`, and `is_osc_manager`. It also includes a `manager_interface` function which is partially visible at the bottom.
- Status Bar:** At the bottom, it shows the file name 'branch\_creation\_auth\_modification\_gestion.py', line numbers (Ln 15, Col 109), spaces count (Spaces: 4), encoding (UTF-8), and other status indicators.

**Figure 46 :** Définition d'une vue Django

- ✓ **Un Formulaire Django (forms.py) :** Définition des formulaires relatifs à l'authentification d'un utilisateur.

**Figure 47 :** Définition d'un formulaire Django

- ✓ **Un Template Django (base.html) :** Définition de *la structure de base des templates*<sup>1</sup> et intégration des fichiers statiques.

```

File Edit Selection View Go Run ... interface_web
base.html M x
cartographie_osc > templates > base.html > html > head
3 <html lang="fr">
4   <head>
5     {% block csslinks %}
6       {% endblock csslinks %}
7       <link rel="stylesheet" href="{% static 'css/styles.css' %}">
8       <link rel="stylesheet" href="{% static 'css/headers.css' %}">
9
10      <!-- CDN -->
11      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
12        integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjCDSCGIMXxSR1GAsXEV/DwyKc2MPK8M2HN" crossorigin="anonymous">
13
14      <title>Cartographie des OSC | {% block pagetitle %}{% endblock pagetitle %}</title>
15
16   </head>
17   <body data-bs-spy="scroll" data-bs-target="#TableOfContent">
18     <!-- HEADER -->
19     {% include 'partials/header.html' %}
20
21     <!-- MAIN -->
22     <main>
23       <div class="container-xxl container-fluid bd-gutter bd-layout">
24         {% block content %}
25           {% block content %}
26         </div>
27       </main>
28
29     <!-- FOOTER -->
30     {% include 'partials/footer.html' %}
31     <script src="{% static 'libraries/jquery/jquery-3.7.1.js' %}"></script>
32
33     <!-- CDN -->
34     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
35       integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjCDSCGIMXxSR1GAsXEV/DwyKc2MPK8M2HN" crossorigin="anonymous"></script>
36     {% block scripts %}
37     {% endblock scripts %}

```

Duo Ln 8, Col 32 Spaces: 4 UTF-8 CRLF HTML Go Live Prettier

**Figure 48 : Définition du template base.html**

- ✓ **Utilisation d'une bibliothèque JavaScript :** Intégration de Leaflet.js dans le template ‘map.html’ pour la visualisation de la carte interactive.

---

<sup>1</sup> Django permet de définir une structure de base des templates de l’application. Cela permet d’éviter de répéter du code et de pouvoir étendre directement cette structure à travers les autres templates.

```
File Edit Selection View Go Run ... ↵ ↶ interface_web
map.html x
cartographie_osc > carto > templates > carto > map.html > ...
3
4  {% block othercsslinks %}
5  <link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css" />
6
7  > <style>...
8  </style>
9  {% endblock othercsslinks %}
10
11  {% block pagetitle %} Carte {% endblock pagetitle %}
12
13  [% block sidebar %]
14  [% include 'carto/partials/map_sidebar.html' %]
15  [% endblock sidebar %]
16
17  [% block main_content %]
18  <!-- <div class="col-md-9 ms-sm-auto col-lg-10 px-md-4"> -->
19  <div class="ms-sm-auto col-lg-10">
20
21      <!-- Insérer les données JSON -->
22      {{ organisations|json_script:"organisations-data" }}
23
24      <div id="map"></div>
25
26  </div>
27  {% endblock main_content %}
28
29  {% block otherscripts %}
30  <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
31
32  > <script>...
33  </script>
34  {% endblock otherscripts %}
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
99
100
101
102
103
104
105
106
107
108
109
110
111
112
```

**Figure 49 : Intégration d'une bibliothèque JavaScript dans un template**

## Conclusion

La réalisation et la mise en œuvre de l'application web ont constitué une étape décisive du projet, permettant de concrétiser les spécifications techniques et les besoins définis lors des phases précédentes. Grâce à des choix judicieux de technologies et à une architecture claire, l'application est désormais opérationnelle et accessible aux utilisateurs finaux. Les efforts pour sécuriser l'accès aux données et les interactions des utilisateurs garantissent que le système est à la fois performant et protégé.

Cette phase a également permis de mettre en place les bases nécessaires pour assurer une maintenance continue du système et son évolution future. À partir de ce point, le système pouvait désormais être testé sur le terrain, avec un suivi constant pour identifier et corriger les éventuelles anomalies. Le projet se trouve maintenant dans une phase de consolidation, avec des perspectives d'amélioration continue basées sur les retours d'expérience des utilisateurs finaux.

# Chapitre 5 : GUIDE

## D'EXPLOITATION

Ce chapitre présente **un guide détaillé sur l'exploitation de l'application**, couvrant les étapes essentielles pour assurer son bon fonctionnement et sa pérennité. Il aborde les configurations matérielles et logicielles requises pour garantir un environnement stable et performant, que ce soit pour le développement, les tests ou la production.

Nous expliquerons les procédures de récupération du code source depuis le dépôt privé GitHub, suivies des étapes de mise en production, aussi bien en environnement local que sur un serveur distant (VPS). Un accent particulier sera mis sur le suivi de l'application à travers l'analyse des journaux de logs, permettant une supervision proactive des performances et la détection rapide des anomalies.

Enfin, la maintenance sera abordée avec la gestion des mises à jour du code et des dépendances, assurant ainsi la sécurité, la correction des éventuels bugs et l'amélioration continue du système. Ce guide vise à offrir une méthodologie claire et complète pour exploiter l'application de manière fiable et sécurisée.

## 5.1. Configurations matérielles et logicielles

Dans cette section, nous détaillerons les configurations matérielles et logicielles nécessaires pour garantir un fonctionnement optimal de l'application. Ces configurations sont essentielles pour assurer la stabilité du système, sa performance et sa capacité à évoluer avec les besoins des utilisateurs. Elles incluent les paramètres techniques nécessaires sur le matériel (serveur, machines locales) et sur les logiciels utilisés (environnement de développement, logiciels tiers, etc.).

### 5.1.1. Configuration matérielle

Pour l'exécution optimale de l'application, il est nécessaire de disposer d'une infrastructure matérielle adéquate comprenant un serveur capable d'héberger et de servir l'application, ainsi que des machines clientes permettant aux utilisateurs d'y accéder de manière fluide et sécurisée.

#### ➤ Serveur

Le serveur, qu'il soit local ou distant, doit posséder des ressources matérielles suffisantes pour assurer la disponibilité, la performance et la gestion des requêtes simultanées. Les caractéristiques matérielles minimales et celles recommandées sont :

**Tableau 13 : Caractéristiques minimales et recommandées pour le serveur**

Caractéristique	Minimale	Recommandée
<b>Processeur</b>	<ul style="list-style-type: none"> <li>• 1-2 cœurs</li> <li>• fréquence de 2 GHz</li> <li>• architecture x86_64</li> </ul>	<ul style="list-style-type: none"> <li>• 4-8 cœurs</li> <li>• fréquence de 2.5 GHz ou plus</li> <li>• architecture x86_64</li> </ul>
<b>Mémoire vive (RAM)</b>	4-8 Go	16 Go ou plus
<b>Stockage</b>	100 Go (SSD)	250 Go ou plus (SSD NVMe recommandé)
<b>Carte réseau</b>	1 Gbps	10 Gbps
<b>Système d'exploitation</b>	<ul style="list-style-type: none"> <li>• Linux (Ubuntu, Debian, CentOS)</li> <li>• Windows</li> </ul>	<ul style="list-style-type: none"> <li>• Linux (Ubuntu Server, Debian, CentOS)</li> <li>• Windows Server</li> </ul>

Ce serveur devra également prendre en charge des outils de gestion comme Nginx et Waitress, facilitant la distribution des ressources et la gestion des connexions entrantes.

## ➤ Client

Les machines clientes, utilisées par les utilisateurs finaux pour accéder à l'application via un navigateur, doivent également respecter certaines exigences minimales afin de garantir une expérience fluide :

- ✓ **Processeur** : Dual-core 2 GHz minimum.
- ✓ **Mémoire vive (RAM)** : 4 Go ou plus.
- ✓ **Navigateur compatible** : Versions récentes de Chrome, Firefox, Edge, etc.
- ✓ **Connexion Internet** : Stabilité et débit suffisants pour charger les données dynamiques de l'application.

Ces prérequis assurent que l'application est accessible et réactive pour l'ensemble des utilisateurs, quel que soit leur rôle (public, gestionnaire, administrateur).

### 5.1.2. Configuration logicielle

La configuration logicielle repose sur un ensemble d'outils et de technologies nécessaires pour garantir une mise en production effective de l'application.

- **Serveur** : On admettra que le serveur exécute **Ubuntu** comme système d'exploitation. Ainsi, il faudra installer les logiciels présentés *plus haut*<sup>1</sup>.
- **Client** : Un simple **navigateur web récent** (Chrome, Firefox, Edge) est suffisant pour accéder à l'application.

## 5.2. Déploiement et suivi

Dans cette section, nous examinerons les étapes du déploiement de l'application ainsi que les stratégies de suivi et de maintenance après sa mise en production. L'objectif est de garantir que le système fonctionne efficacement, en suivant les bonnes pratiques pour le déploiement initial et la gestion à long terme.

### 5.2.1. Déploiement

Cette sous-section détaillera les étapes de mise en ligne de l'application, depuis la préparation des serveurs de production jusqu'à la configuration finale. Le processus de déploiement sur le VPS sera expliqué, ainsi que la configuration des services du système nécessaires pour l'exécution automatique du serveur web et du serveur d'application.

#### 5.2.1.1. Mise à jour du système (Ubuntu 20.04)

Pour mettre à jour le système d'exploitation et ses différentes dépendances :

```
```bash
```

```
sudo apt update && sudo apt upgrade -y
```

---

<sup>1</sup> Voir *supra*, Chapitre IV, Section 4.1.2, Page 85.

```
'''
```

### 5.2.1.2. Installation de Python 3.12, pip et venv

Une fois l'OS<sup>1</sup> mis à jour, on procède à l'installation de la version 3.12 de Python, son gestionnaire de dépendances et le paquet de gestion des environnements virtuels :

```
'''bash
```

```
sudo apt install -y python3.12 python3.12-venv python3.12-dev python3-pip
```

```
'''
```

### 5.2.1.3. Installation de Docker et Téléchargement de l'image PostgreSQL 16

N'étant pas possible d'installer PostgreSQL 16 directement sur Ubuntu 20.04, on passe par l'installation de Docker et l'utilisation d'une image PostgreSQL 16 :

```
'''bash
```

```
sudo apt install -y docker.io
```

```
sudo systemctl enable --now docker
```

```
docker pull postgres:16
```

```
'''
```

### 5.2.1.4. Installation de Nginx

On installe le serveur web Nginx et on active son service :

```
'''bash
```

```
sudo apt install -y nginx
```

```
sudo systemctl enable --now nginx
```

```
'''
```

---

<sup>1</sup> OS : Operating System. C'est la traduction anglaise de : Système d'Exploitation.

### 5.2.1.5. Installation et configuration de UFW

Pour contrôler et restreindre les trafic entrants et sortants, on installe UFW et on définit les différentes règles au besoin :

```
```bash
```

```
sudo apt install -y ufw
```

```
sudo ufw allow OpenSSH
```

```
sudo ufw allow 443/tcp    # HTTPS
```

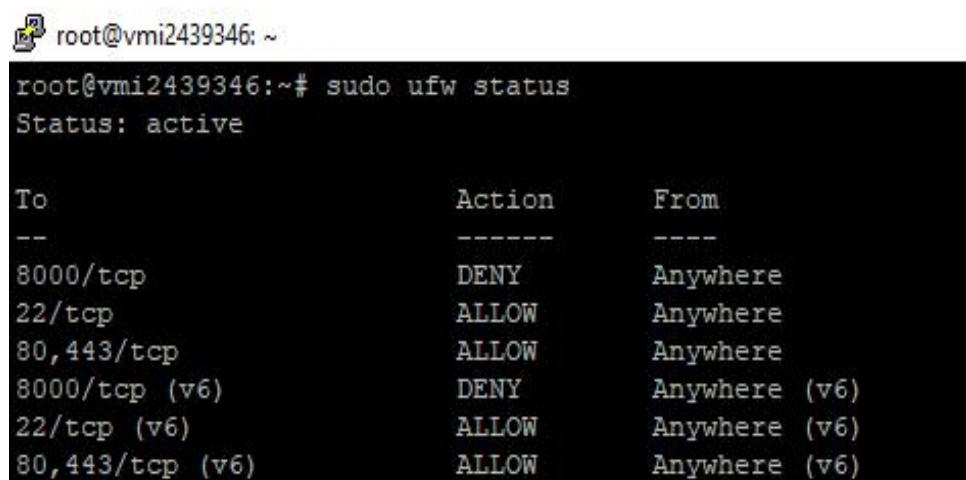
```
Sudo ufw allow 80/tcp    #HTTP
```

```
sudo ufw deny 8000/tcp    # Bloquer accès public à l'app Django en local
```

```
sudo ufw enable
```

```
```
```

Voici un aperçu des règles définies :



```
root@vmi2439346: ~
root@vmi2439346:~# sudo ufw status
Status: active

To                         Action      From
--                         --         --
8000/tcp                   DENY       Anywhere
22/tcp                      ALLOW      Anywhere
80,443/tcp                 ALLOW      Anywhere
8000/tcp (v6)               DENY       Anywhere (v6)
22/tcp (v6)                 ALLOW      Anywhere (v6)
80,443/tcp (v6)             ALLOW      Anywhere (v6)
```

Figure 50 : Règles UFW

### 5.2.1.6. Exportation de la base de données depuis la machine locale

Les données existent en local. Il faut donc les exporter vers la machine de mise en production.

Ainsi sur l'hôte local, on exécute :

```
```bash
```

```
pg_dump -h 127.0.0.1 -U postgres -F c -b -v -f carto_osc_backup.dump osc_db
```

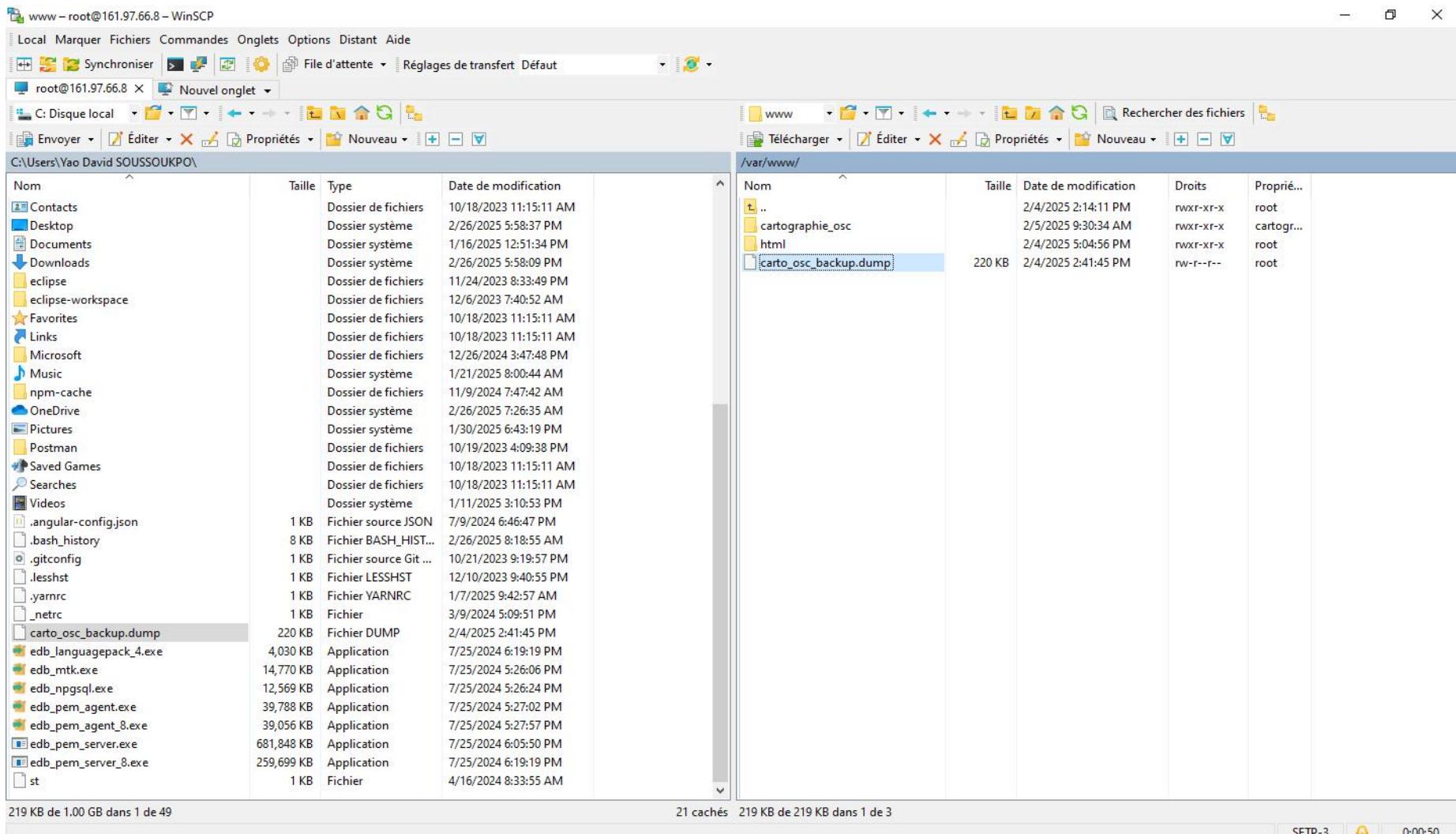
```
```
```

### 5.2.1.7. Envoi de la base de données exportée vers le VPS

Avec l'utilisation de WinSCP,<sup>1</sup> on envoie la base de données exportée vers l'hôte distant :

---

<sup>1</sup> WinSCP (Windows Secure Copy), [rappelons le], est un client SFTP (SSH File Transfer Protocol) et FTP (File Transfer Protocol) [à interface graphique] pour Windows. Il permet donc, ici, de transférer la base de données exportée ; **carto\_osc\_backup.dump**, depuis l'hôte local ; **Windows 10 Pro**, vers l'hôte distant ; **Ubuntu 20.04** en toute sécurité.



**Figure 51 :** Transfert effectué entre les deux hôtes via WinSCP

Une alternative aurait été, en ayant au préalable **OpenSSH**<sup>1</sup> installé, d'exécuter directement dans l'invite de commande de la **machine locale**, la commande suivante :

```
```bash
```

```
scp      C:\Users\Yao      David      SOUSSOUKPO\carto_osc backup.dump  
root@161.97.66.8:/var/www/
```

```
```
```

### 5.2.1.8. Création d'un conteneur Docker PostgreSQL 16 et restauration de la base de données

Une fois l'envoi effectué, sur le **VPS**, on exécute<sup>2</sup> :

```
```bash
```

```
docker      run      --name      postgres16-container      -e  
POSTGRES_PASSWORD="MOT_DE_PASSE" -d -p 5432:5432 postgres:16
```

```
```
```

Ensuite, restaurer la base de données avec :

```
```bash
```

```
docker exec -i postgres16-container pg_restore -U postgres -d postgres <  
/var/www/cartoon_osc_backup.dump
```

```
```
```

### 5.2.1.9. Configuration du répertoire du code source

On récupère maintenant le code source de l'application sur le VPS. Pour ce faire, on suit les étapes suivantes :

---

<sup>1</sup> OpenSSH (Open Secure Shell) est une suite d'outils pour accéder en toute sécurité à des systèmes distants sur des réseaux [potentiellement] non sécurisés. OpenSSH met en œuvre le protocole SSH (Secure Shell), qui permet de chiffrer les connexions réseau afin de protéger les données en transit. SCP (Secure Copy) est l'un de ses outils principaux.

<sup>2</sup> En remplaçant « MOT\_DE\_PASSE » par sa vraie valeur.

## ➤ Cr ation du r epertoire

```
```bash
```

```
sudo mkdir -p /var/www/cartographie_osc
```

```
cd /var/www/cartographie_osc
```

```
...
```

## ➤ Clonage du code source depuis GitLab

```
```bash
```

```
git clone -b branch_creation_auth_modification_gestion
```

```
https://gitlab.com/PsYoni/cartographie_osc-app.git
```

```
/var/www/cartographie_osc
```

```
...
```

## ➤ Cr ation et activation d'un environnement virtuel

```
```bash
```

```
python3.12 -m venv env
```

```
source env/bin/activate
```

```
...
```

## ➤ Installation des d ependances

```
```bash
```

```
pip install -r requirements.txt
```

```
...
```

## ➤ Application des migrations

```
```bash
```

```
python manage.py migrate
```

```
...
```

## ➤ Collecte des fichiers statiques

```
```bash
```

```
python manage.py collectstatic
```

```
```
```

## ➤ Configuration de Django

- ✓ Modifier `DEBUG=False` dans `settings.py`
- ✓ Ajouter l'IP du serveur dans `CSRF\_TRUSTED\_ORIGINS` toujours dans `settings.py`

```
```python
```

```
CSRF_TRUSTED_ORIGINS = ["https://161.97.66.8"]
```

```
```
```

### 5.2.1.10. Exécution de l'application avec Waitress

Pour lancer le serveur WSIG Waitress, on exécute la commande suivante :

```
```bash
```

```
waitress-serve --port=8000 cartographie_osc.wsgi:application
```

```
```
```

### 5.2.1.11. Acquisition d'un certificat SSL auto-signé

Le nom de domaine, n'étant pas encore acquis, l'utilisation du protocole de communication HTTPS, qui nécessite un certificat SSL authentique, se fait ici, via un certificat SSL auto-signé. Pour acquérir ce dernier, on exécute la commande suivante :

```
```bash
```

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/nginx/ssl/161.97.66.8.key -out /etc/nginx/ssl/161.97.66.8.crt
```

```
```
```

### 5.2.1.12. Configuration de Nginx

Afin de respectivement configurer les paramètres généraux de Nginx en tant que *reverse proxy*<sup>1</sup> et les paramètres spécifiques à l’application, on procède aux ajustements suivants :

- Configuration globale `/etc/nginx/nginx.conf` :

```
```bash
```

```
sudo nano /etc/nginx/nginx.conf
```

```
root@vmi2439346:/
```

```
GNU nano 4.8
```

```
user www-data; # Utilise l'utilisateur www-data pour les processus Nginx
worker_processes auto; # Nginx ajuste automatiquement le nombre de workers en fonction du nombre de coeurs
pid /run/nginx.pid; # Emplacement du fichier PID
```

```
# Section des événements, indispensable
events {
    worker_connections 1024; # Nombre maximal de connexions par worker
}

# Les directives de log doivent être dans le bloc http
http {
    # Définition des logs
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log warn;

    include /etc/nginx/mime.types; # Charge les types MIME
    default_type application/octet-stream; # Type MIME par défaut pour les fichiers

    log_format main '$remote_addr - $remote_user [$time_local] "$request"
$status $body_bytes_sent "$http_referer"
"$http_user_agent" "$http_x_forwarded_for"';

    sendfile on; # Active la gestion des fichiers avec sendfile pour une meilleure performance
    tcp_nopush on; # Optimisation pour envoyer les fichiers plus efficacement
    tcp_nodelay on; # Optimisation pour éviter la mise en tampon des petites requêtes
    keepalive_timeout 65; # Définir la durée de vie de la connexion persistante
    types_hash_max_size 2048; # Augmente la taille de la table des types MIME

    # Active le SSL globalement
    ssl_protocols TLSv1.2 TLSv1.3; # Utilisation de TLS 1.2 et 1.3 uniquement
    ssl_ciphers 'TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384';
    include /etc/nginx/sites-enabled/*; # Charge les configurations des sites activés
}
```

Figure 52 : Configuration globale de Nginx sur le VPS

- Crédit de la configuration spécifique `/etc/nginx/sites-available/cartographie\_osc` :

```
```bash
```

```
sudo nano /etc/nginx/sites-available/cartographie_osc
```

```
```
```

---

<sup>1</sup> Un reverse proxy est un serveur intermédiaire qui se trouve entre les utilisateurs (clients) et un ou plusieurs serveurs back-end.

```

root@vmi2439346: ~
GNU nano 4.8
server {
    listen 80;
    server_name 161.97.66.8;

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl; # Utilise le port 443 pour HTTPS
    server_name 161.97.66.8;

    # Emplacement des logs
    error_log /var/log/nginx/cartographie_osc_error.log;
    access_log /var/log/nginx/cartographie_osc_access.log;

    # Définitions SSL
    ssl_certificate /etc/nginx/ssl/161.97.66.8.crt;
    ssl_certificate_key /etc/nginx/ssl/161.97.66.8.key;

    # Serve les fichiers statiques avec un cache de 30 jours
    location /static/ {
        alias /var/www/cartographie_osc/static_root/;
        expires 30d;
        add_header Cache-Control "public";
    }

    # Serve les fichiers médias avec un cache de 30 jours
    location /media/ {
        alias /var/www/cartographie_osc/media/;
        expires 30d;
        add_header Cache-Control "public";
    }

    # Redirige les autres requêtes vers Waitress
    location / {
        proxy_pass http://127.0.0.1:8000; # Adresse du serveur waitress WSGI local
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

```

**Figure 53 : Configuration spécifique à l'application web**

## ➤ Activation de la configuration et redémarrage de Nginx

````bash`

```
sudo ln -s /etc/nginx/sites-available/cartographie_osc /etc/nginx/sites-enabled/
```

```
sudo systemctl restart nginx
```

`````

### 5.2.1.13. Utilisation de systemd pour automatiser le lancement de l'application

Afin d'assurer une disponibilité continue, on devra s'assurer que les différents serveurs notamment le serveur d'application soient en cours d'exécution aussi longtemps que le VPS est actif. Ainsi, tout comme avec Nginx, on crée un service système.

## ➤ Création du fichier de service `'/etc/systemd/system/cartographie\_osc.service'

````bash`

```
sudo nano /etc/systemd/system/cartographie_osc.service
```

`````

```

system - root@161.97.66.8 - WinSCP
Loc /etc/systemd/system/cartographie_osc.service - root@161.97.66.8 - Éditeur - WinSCP
[Unit]
Description=Waitress Application Server for Cartographie OSC
After=network.target

[Service]
ExecStart=/var/www/cartographie_osc/env/bin/waitress-serve --port=8000 cartographie_osc.wsgi:application
WorkingDirectory=/var/www/cartographie_osc
User=cartographie_osc
Group=cartographie_osc
Environment=PATH=/var/www/cartographie_osc/env/bin
Environment=VIRTUAL_ENV=/var/www/cartographie_osc/env
Environment=PYTHONUNBUFFERED=1
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target

```

**Figure 54 : Configuration du service système cartographie\_osc.service**

## ➤ Activation du service et démarrage automatique

*```bash*

*sudo systemctl daemon-reload*

*sudo systemctl enable cartographie\_osc*

*sudo systemctl start cartographie\_osc*

*```*

Une fois les services configurés, on peut maintenant vérifier le fonctionnement de ceux-ci, avec les commandes :

*```bash*

*sudo systemctl status cartographie\_osc.service*

*sudo systemctl status nginx*

*```*

```

root@vmi2439346:~# sudo systemctl status cartographie_osc.service
● cartographie_osc.service - Waitress Application Server for Cartographie OSC
   Loaded: loaded (/etc/systemd/system/cartographie_osc.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-02-13 12:06:31 CET; 1 weeks 6 days ago
     Main PID: 276899 (waitress-serve)
        Tasks: 16 (limit: 57756)
       Memory: 64.5M
      CGroup: /system.slice/cartographie_osc.service
              └─276899 /var/www/cartographie_osc/env/bin/python3.12 /var/www/cartographie_osc/env/bin/wa

Feb 26 18:21:21 vmi2439346.contaboserver.net waitress-serve[276899]: Not Found: /public/.git/config
Feb 26 18:21:21 vmi2439346.contaboserver.net waitress-serve[276899]: WARNING:django.request:Not Found:
Feb 26 18:21:22 vmi2439346.contaboserver.net waitress-serve[276899]: Not Found: /logs/archived/.git/con
Feb 26 18:21:22 vmi2439346.contaboserver.net waitress-serve[276899]: WARNING:django.request:Not Found:
Feb 26 18:34:33 vmi2439346.contaboserver.net waitress-serve[276899]: Not Found: /cgi-bin/luci/;stok=/lo
Feb 26 18:34:33 vmi2439346.contaboserver.net waitress-serve[276899]: WARNING:django.request:Not Found:
Feb 26 18:51:32 vmi2439346.contaboserver.net waitress-serve[276899]: Not Found: /favicon.ico
Feb 26 18:51:32 vmi2439346.contaboserver.net waitress-serve[276899]: WARNING:django.request:Not Found:
Feb 26 18:52:19 vmi2439346.contaboserver.net waitress-serve[276899]: Not Found: /favicon.ico
Feb 26 18:52:19 vmi2439346.contaboserver.net waitress-serve[276899]: WARNING:django.request:Not Found:
root@vmi2439346:~# sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-02-13 12:06:38 CET; 1 weeks 6 days ago
     Docs: man:nginx(8)
  Process: 276914 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, s
  Process: 276922 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/S
   Main PID: 276923 (nginx)
      Tasks: 13 (limit: 57756)
     Memory: 17.1M
      CGroup: /system.slice/nginx.service
              ├─276923 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              ├─276924 nginx: worker process
              ├─276925 nginx: worker process
              ├─276926 nginx: worker process

```

**Figure 55 : Fonctionnement<sup>1</sup> des services système**

### 5.2.1.14. Déploiement terminé !

L’application est ainsi accessible via <https://161.97.66.8>

## 5.2.2. Suivi et maintenance

Une fois l’application déployée, il est essentiel de mettre en place un plan de suivi et de maintenance pour garantir sa stabilité, ses performances et sa sécurité à long terme. Cette section détaille les différentes actions mises en œuvre pour assurer un fonctionnement optimal du système.

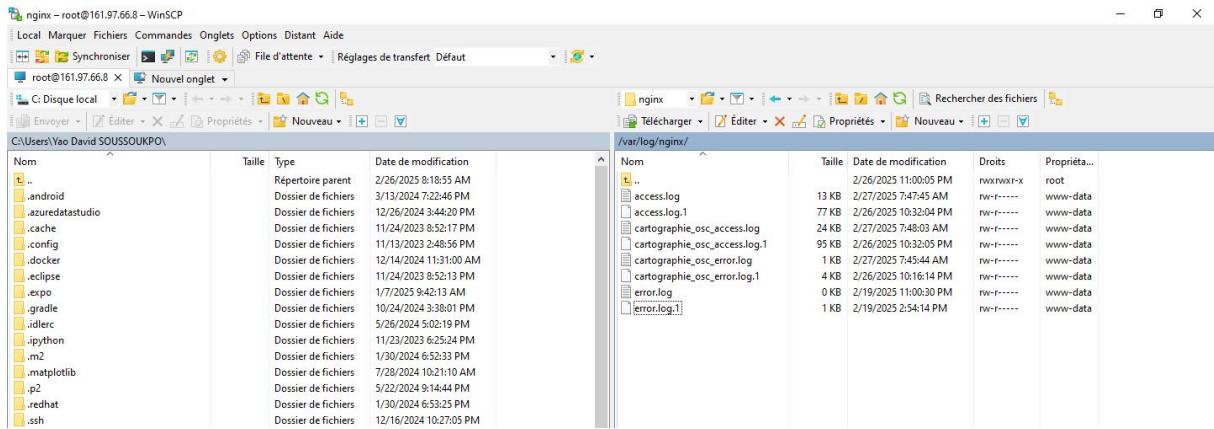
### 5.2.2.1. Surveillance et monitoring

La supervision de l’application repose sur plusieurs outils permettant d’assurer un suivi en temps réel et d’anticiper d’éventuels problèmes :

- **Logs Nginx :**

---

<sup>1</sup> Les signaux verts `active (running)` sur l’image indiquent que les services s’exécutent avec succès.



**Figure 56 : À droite, les fichiers de logs (généraux et spécifiques) du serveur web**

On distingue différents fichiers tels que :

- ✓ **Logs d'accès globaux `'/var/log/nginx/access.log'** : enregistre toutes les requêtes traitées par Nginx.
- ✓ **Logs d'erreurs globaux `'/var/log/nginx/error.log'** : enregistre les erreurs globales de Nginx, y compris celles non liées à ton application.
- ✓ **Logs d'accès spécifiques à l'application web `'/var/log/nginx/cartographie\_osc\_access.log'** : enregistre uniquement les requêtes liées à l'application cartographie\_osc.
- ✓ **Logs d'erreurs spécifiques à l'application web `'/var/log/nginx/cartographie\_osc\_error.log'** : enregistre uniquement les erreurs générées par cartographie\_osc.

#### ➤ Logs d'application Django :

- ✓ Suivi des erreurs via le fichier `logs/django.log`, configuré dans `settings.py` pour enregistrer les erreurs en production.

| /var/www/cartographie_osc/logs/ |        |                      |           |                  |  |
|---------------------------------|--------|----------------------|-----------|------------------|--|
| Nom                             | Taille | Date de modification | Droits    | Propriétaire     |  |
| ..                              |        |                      |           |                  |  |
| django.log                      | 1 KB   | 2/27/2025 8:08:42 AM | rwxr-xr-x | cartographie_osc |  |

Figure 57 : Fichier de logs Django

- ✓ Mise en place d'une rotation des logs, pour éviter que le fichier devienne trop grand, avec la configuration de **logrotate**<sup>1</sup>.

```
root@vmi2439346: ~
GNU nano 4.8
/var/www/cartographie_osc/logs/django.log {
    daily          # Pivoter les journaux quotidiennement
    missingok      # Ne pas générer d'erreur si le fichier n'existe pas
    rotate 14       # Conserver 14 fichiers journaux avant de supprimer
    compress        # Compresser les fichiers journaux lors de la rotation
    delaycompress   # Ne pas compresser le dernier fichier journal immédiatement
    notifempty      # Ne pas pivoter si le fichier journal est vide
    create 664 cartographie_osc cartographie_osc # Créer un nouveau fichier journal avec les bonnes permissions
    sharedscripts    # Exécuter les scripts après chaque rotation
    postrotate
        systemctl reload cartographie_osc.service > /dev/null 2>&1 || true
    endscript
}
```

Figure 58 : Aperçu du fichier de configuration pour logrotate

- **Suivi des services** : Vérification de l'état des services avec `systemctl status nginx`, `systemctl status cartographie\_osc` et `docker ps` pour la base de données PostgreSQL.

---

<sup>1</sup> Logrotate est un outil pratique pour gérer les fichiers journaux, permettant de les faire pivoter, compresser, et supprimer les anciens journaux afin de conserver une utilisation efficace de l'espace disque.

```

root@vmi2439346:~#
root@vmi2439346:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-02-13 12:06:38 CET; 1 weeks 6 days ago
     Docs: man/nginx(8)
 Process: 276914 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 276922 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 276923 (nginx)
   Tasks: 13 (limit: 57756)
  Memory: 18.4M
    CGroup: /system.slice/nginx.service
            └─276923 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              ├─276924 nginx: worker process
              ├─276925 nginx: worker process
              ├─276926 nginx: worker process
              ├─276927 nginx: worker process
              ├─276928 nginx: worker process
              ├─276929 nginx: worker process
              ├─276930 nginx: worker process
              ├─276931 nginx: worker process
              ├─276932 nginx: worker process
              ├─276933 nginx: worker process
              ├─276934 nginx: worker process
              └─276935 nginx: worker process

Warning: journal has been rotated since unit was started, output may be incomplete.
root@vmi2439346:~# systemctl status cartographie_osc
● cartographie_osc.service - Waitress Application Server for Cartographie OSC
   Loaded: loaded (/etc/systemd/system/cartographie_osc.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-02-27 10:02:42 CET; 8min ago
     Main PID: 558860 (waitress-serve)
       Tasks: 16 (limit: 57756)
      Memory: 65.9M
        CGroup: /system.slice/cartographie_osc.service
                  └─558860 /var/www/cartographie_osc/env/bin/python3.12 /var/www/cartographie_osc/env/bin/waitress-serve --po

Feb 27 10:02:42 vmi2439346.contaboserver.net systemd[1]: Started Waitress Application Server for Cartographie OSC.
Feb 27 10:02:43 vmi2439346.contaboserver.net waitress-serve[558860]: INFO:waitress:Serving on http://0.0.0.0:8000
root@vmi2439346:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c56033be2887 postgres:16 "docke... 3 weeks ago Up 3 weeks 0.0.0.0:5432->5432/tcp postgres16
root@vmi2439346:~#
root@vmi2439346:~#

```

**Figure 59 : Aperçu de l'activité des services**

## ➤ Outils de monitoring :

- ✓ Mise en place de **htop** et **netstat** pour surveiller la consommation des ressources (CPU, RAM, ports ouverts) [essentiellement] par les services **nginx**, **cartographie\_osc** et **docker**.

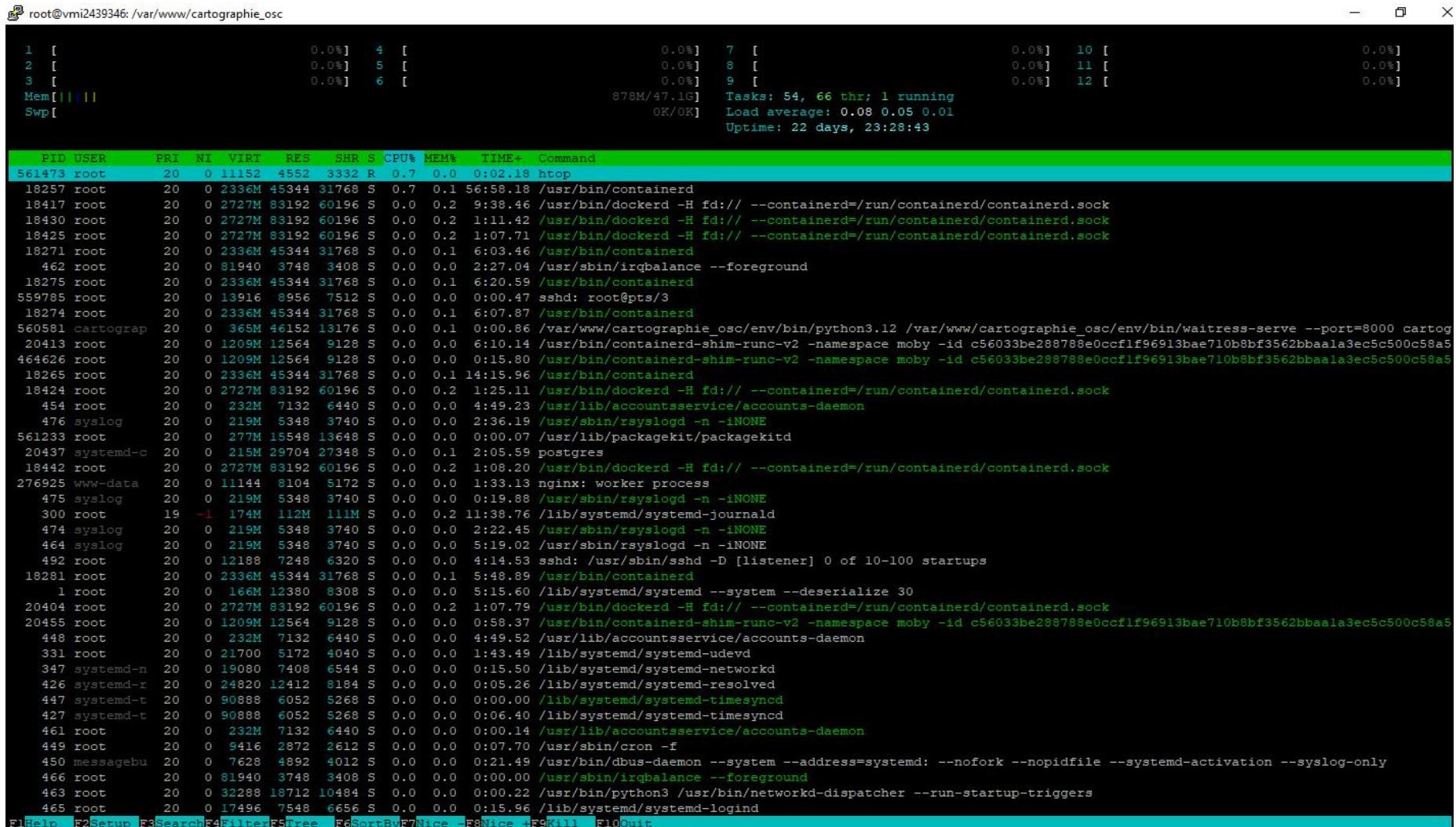


Figure 60 : Suivi de la consommation des ressources par les processus via la commande 'htop'

```

root@vmi2439346: ~
This server is hosted by Contabo. If you have any questions or need help,
please don't hesitate to contact us at support@contabo.com.

Last login: Thu Feb 27 11:28:17 2025 from 196.168.221.104
root@vmi2439346:~# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
tcp      0      0 0.0.0.0:80                0.0.0.0:*              LISTEN    276923/nginx: maste
tcp      0      0 127.0.0.53:53             0.0.0.0:*              LISTEN    426/systemd-resolve
tcp      0      0 0.0.0.0:22                0.0.0.0:*              LISTEN    492/sshd: /usr/sbin
tcp      0      0 0.0.0.0:5432             0.0.0.0:*              LISTEN    20386/docker-proxy
tcp      0      0 0.0.0.0:443               0.0.0.0:*              LISTEN    276923/nginx: maste
tcp      0      0 0.0.0.0:8000             0.0.0.0:*              LISTEN    560581/python3.12
tcp6     0      0 :::80                  :::*                  LISTEN    276923/nginx: maste
tcp6     0      0 ::::22                 ::::*                 LISTEN    492/sshd: /usr/sbin
udp      0      0 127.0.0.53:53             0.0.0.0:*              LISTEN    426/systemd-resolve
root@vmi2439346:~# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
tcp      0      0 0.0.0.0:80                0.0.0.0:*              LISTEN    276923/nginx: maste
tcp      0      0 127.0.0.53:53             0.0.0.0:*              LISTEN    426/systemd-resolve
tcp      0      0 0.0.0.0:22                0.0.0.0:*              LISTEN    492/sshd: /usr/sbin
tcp      0      0 0.0.0.0:5432             0.0.0.0:*              LISTEN    20386/docker-proxy
tcp      0      0 0.0.0.0:443               0.0.0.0:*              LISTEN    276923/nginx: maste
tcp      0      0 0.0.0.0:8000             0.0.0.0:*              LISTEN    560581/python3.12
tcp      0      0 161.97.66.8:80              43.156.168.214:56358 TIME_WAIT   -
tcp      0      272 161.97.66.8:22            196.168.221.104:47261 ESTABLISHED 561669/sshd: root@p
tcp      0      0 161.97.66.8:22              196.168.221.104:47094 ESTABLISHED 561671/sshd: root@n
tcp      0      0 172.17.0.1:53234            172.17.0.2:5432          TIME_WAIT   -
tcp      0      0 127.0.0.1:45054            127.0.0.1:8000          TIME_WAIT   -
tcp      0      0 161.97.66.8:22              196.168.221.104:47090 ESTABLISHED 560672/sshd: root@n
tcp      0      0 161.97.66.8:22              196.168.221.104:47105 ESTABLISHED 559785/sshd: root@p
tcp      0      0 161.97.66.8:22              196.168.221.104:47272 ESTABLISHED 560294/sshd: root@n
tcp      0      0 161.97.66.8:22              196.168.221.104:47264 ESTABLISHED 559741/sshd: root@n
tcp      0      0 127.0.0.1:46188            127.0.0.1:5432          TIME_WAIT   -
tcp6     0      0 ::::80                  ::::*                 LISTEN    276923/nginx: maste
tcp6     0      0 ::::22                 ::::*                 LISTEN    492/sshd: /usr/sbin

```

**Figure 61 :** Suivi des ports ouverts et connexions établies avec les commandes `netstat -tulnp` et `netstat -antp`

- ✓ Utilisation de **Fail2ban** pour détecter et bloquer les tentatives d'accès non autorisées.

```

[sshd]
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
enabled = true
port    = ssh
filter  = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 3600
backend = auto

```

**Figure 62 :** Activation de la Protection SSH avec Fail2ban

### 5.2.2.2. Gestion des mises à jour

Les mises à jour du système et de l'application sont réalisées à intervalles réguliers pour assurer la sécurité et l'amélioration continue du service :

- **Mises à jour du système** : Une mise à jour mensuelle est effectuée pour maintenir les performances et appliquer les correctifs de sécurité du serveur VPS.
- **Mises à jour des données** : L'administrateur et les gestionnaires de l'application mettent à jour les données du système chaque mois, via les interfaces d'administration.
- **Mises à jour fonctionnelles** :
  - ✓ Tous les trois mois, une analyse des besoins est effectuée pour intégrer de nouvelles fonctionnalités ou améliorer l'existant sur la base d'éventuels bugs.
  - ✓ Le processus de mise à jour suit un cycle **développement → test → validation → déploiement** via GitLab.

#### 5.2.2.3. Gestion des erreurs et optimisation

Pour minimiser les interruptions de service, des actions correctives sont prévues :

- **Identification et correction des bugs** :
  - ✓ Surveillance régulière des logs pour détecter les anomalies.
  - ✓ Correction et déploiement des correctifs via GitLab et redémarrage du service en cas de besoin :

```
```bash
```

```
sudo systemctl restart cartographie_osc
```

```
```
```

- **Optimisation des performances** : un nettoyage périodique des logs pour éviter une surcharge du stockage :

```
```bash
```

```
sudo truncate -s 0 /var/log/nginx/access.log
```

```
sudo truncate -s 0 /var/log/nginx/error.log
```

```
sudo truncate -s 0 /var/log/nginx/cartographie_osc_access.log
```

```
sudo truncate -s 0 /var/log/nginx/cartographie_osc_error.log
```

```

#### 5.2.2.4. Sauvegarde et récupération des données

La mise en place d'un plan de sauvegarde est essentielle pour éviter toute perte de données en cas d'incident.

➤ **Sauvegarde automatique de la base de données :**

- ✓ Un script cron est configuré pour exporter la base de données PostgreSQL quotidiennement à 02h du matin :

```bash

```
0 2 * * * pg_dump -U postgres -Fc -f /var/backups/carto_osc_$(date  
+%\F).dump
```

```

- ✓ Les sauvegardes sont stockées sur le serveur et une copie est envoyée sur un espace de stockage distant sécurisé.

➤ **Restauration en cas de problème :** En cas de corruption ou de perte de données, la restauration peut être effectuée via :

```bash

```
pg_restore -U postgres -d postgres /var/backups/carto_osc_backup.dump
```

```

Grâce à ces mesures de suivi et de maintenance, l'application garantit un fonctionnement optimal, une sécurité renforcée et une amélioration continue pour répondre aux besoins des utilisateurs.

## Conclusion

Le déploiement et la gestion efficace de l'application nécessitent une attention particulière à la configuration matérielle et logicielle, ainsi qu'à la mise en place de stratégies de suivi et de

maintenance. Ce chapitre a permis de détailler les différentes étapes impliquées dans l'exploitation de l'application ; **la configuration de l'environnement de production et la gestion continue du système une fois déployé.**

Nous avons vu que le succès de l'application repose sur une configuration matérielle robuste, adaptée à l'échelle du projet, ainsi qu'une architecture logicielle bien définie pour garantir sa performance et sa sécurité. La maintenance continue et les mises à jour régulières garantiront la pérennité du système et sa capacité à évoluer avec le temps.

Ainsi, ce chapitre met en lumière l'importance de l'exploitation dans la gestion d'un projet logiciel, et fournit une base solide pour garantir son bon fonctionnement à long terme. En somme, l'exploitation de l'application ne se limite pas seulement à sa mise en production, mais englobe un suivi constant pour optimiser son efficacité et sa durabilité.

## **Chapitre 6 : GUIDE D'UTILISATION**

Ce chapitre se consacre à la présentation détaillée de l'utilisation de l'application web développée dans le cadre du projet EU4SUN. L'objectif est de fournir aux différents utilisateurs un accompagnement structuré pour une prise en main optimale de la plateforme, tout en garantissant la sécurité et la bonne gestion des données.

La diversité des profils impliqués dans l'exploitation de l'application, tels que les membres du public, les gestionnaires d'organisations et les administrateurs, nécessite une répartition claire des rôles et responsabilités. Ce guide vise donc à expliquer l'organisation des interfaces, les accès aux différentes sections ainsi que les actions autorisées pour chaque profil utilisateur.

L'accent sera mis sur :

- **La présentation des interfaces** : Une description du layout et des principales sections de l'application.
- **Les rôles et accès** : La distinction des niveaux d'accès (public, manager OSC, administrateur).
- **Les bonnes pratiques d'utilisation** : Pour maintenir l'intégrité des données et assurer une gestion collaborative efficace.

Ce guide constitue une référence essentielle pour les parties prenantes, permettant à la fois une utilisation simplifiée et la préservation de la qualité des données collectées au sein de la cartographie des OSC.

## 6.1. Layout des pages de l'application web

L'application web est conçue autour d'une interface intuitive et accessible, organisée comme suit :

### 6.1.1. En-tête

L'en-tête a deux apparences principales :

- **La première apparence** se retrouve sur la page d'accueil : On y trouve, juste en dessous de la bannière, les liens vers les principales pages de l'application.



Figure 63 : En-tête - Apparence 1

- **La deuxième apparence** est [une bande noire] présente dans les autres pages de l'application avec la particularité de s'adapter selon que l'utilisateur est connecté ou non :
  - ✓ **Utilisateur non-connecté** : C'est la configuration par défaut de l'en-tête.



Figure 64 : En-tête - Apparence 2 - Utilisateur non-connecté

- ✓ **Manager d'OSC connecté** : Des boutons supplémentaires ; en cliquant sur le logo de l'OSC, sont présents pour permettre au manager de naviguer dans son espace de gestion.



Figure 65 : En-tête - Apparence 2 - Manager d'OSC connecté

- ✓ **Administrateur connecté** : En extension à l'apparence par défaut, il y a un bouton « Administration » qui permet d'aller directement sur l'interface d'administration avancée Django.



Figure 66 : En-tête - Apparence 2 - Administrateur connecté

### 6.1.2. Menu de gauche et corps d'une page

Absent sur la page d'accueil, le menu de gauche permet de manipuler les informations d'une page, de naviguer entre différentes sections d'une même page ou encore de passer d'une page à l'autre dans certains cas. Le corps d'une page, quant à lui, contient l'essentiel de la page ; son contenu.

Ils seront présentés et développés pour chaque page dans la section 6.2.

### 6.1.3. Pied de page

Le pied de page contient les informations sur les auteurs, commanditaires de l'application, quelques liens vers diverses pages et la même bannière de l'en-tête de la page d'accueil. Au delà de permettre une navigation, il permet aussi d'accéder à la documentation.

Le pied de page a principalement une seule apparence mais s'adapte en fonction de l'état de l'utilisateur qu'il soit connecté ou pas et en fonction du type d'utilisateur connecté :

- **Utilisateur non-connecté** : Étant non-connecté, il peut se connecter en tant que « Manager d'OSC » ou « Administrateur ».

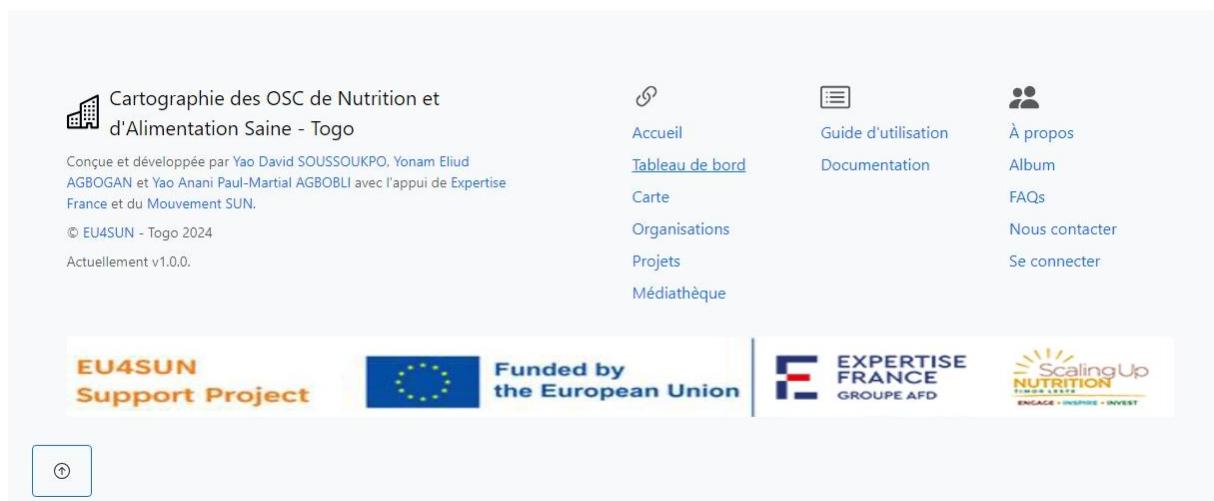
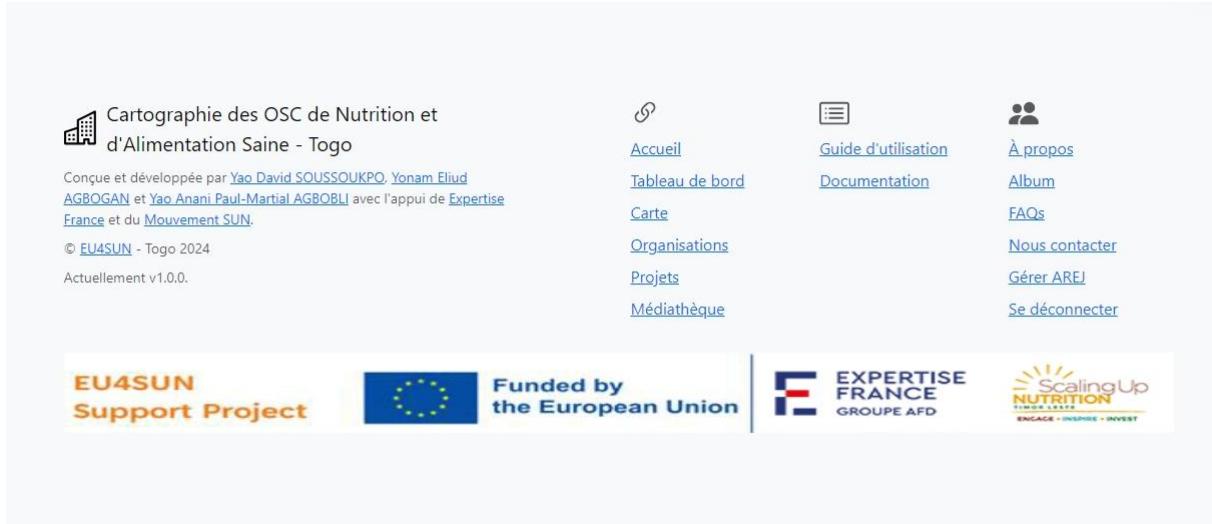


Figure 67 : Pied de page - Utilisateur non-connecté

➤ **Utilisateur connecté :**

- ✓ **Manager d'OSC** : Il peut naviguer vers son espace de gestion et peut se déconnecter.



**Figure 68 : Pied de page - Manager d'OSC connecté**

- ✓ **Administrateur** : À l'instar du « Manager d'OSC », il peut naviguer vers son espace d'administration et se déconnecter.



**Figure 69 : Pied de page - Administrateur connecté**

## 6.2. Utilisateurs : Accès, Rôles & Responsabilités

Dans cette section, nous aborderons les pages et fonctionnalités auxquelles ont accès les différents *rôles utilisateurs*<sup>1</sup>.

### 6.2.1. Public

Ce premier rôle est l'utilisateur par défaut (non-authentifié). Il a accès à un nombre limité de pages et ne dispose que d'un accès en « lecture » sur les données de l'application.

- **Page d'accueil** : Elle présente la mission et le projet EU4SUN en général et le contexte précis de développement de l'application web de la cartographie.



Figure 70 : Aperçu de la page d'accueil

- **Tableau de bord** : Cette page présente les statistiques liées à la répartition des OSC, leurs projets et la répartition du personnel en nutrition de ces OSC sur le territoire. Ces statistiques présentées en graphiques interactifs permettent de mieux apprécier la couverture du territoire par les OSC et de mieux orienter les actions à mener pour atteindre les cibles et les objectifs définis. Le **menu de gauche** permet de naviguer entre

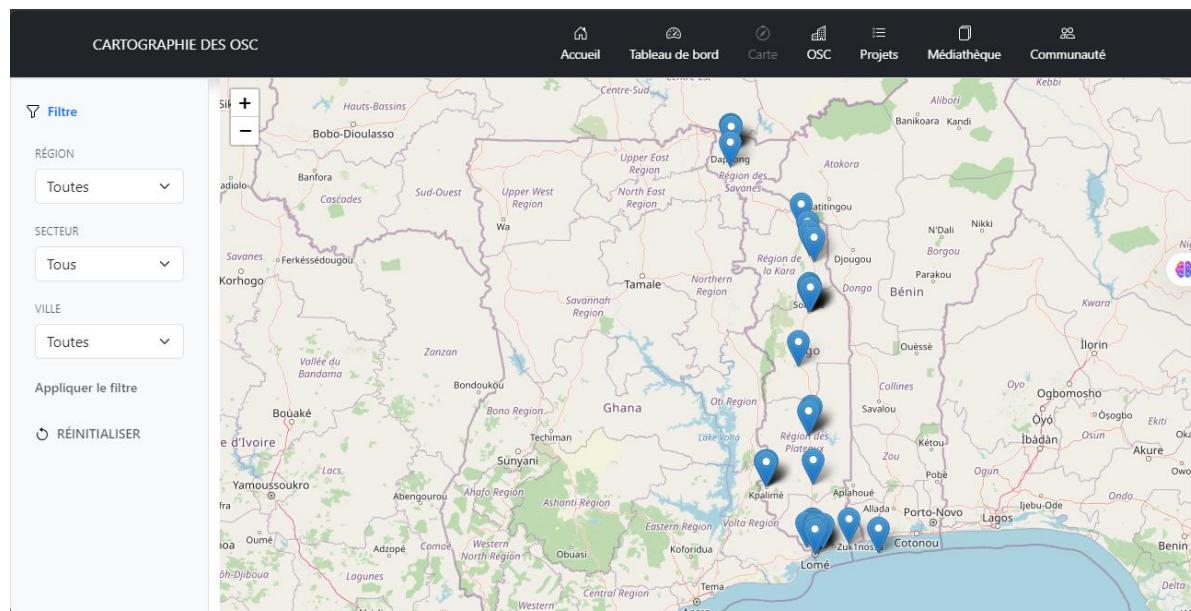
<sup>1</sup> Ceux-ci ont déjà été introduits plus haut dans le document. Voir *supra*, Chapitre III, Section 3.4.1, Page 49.

les différentes sections de la page et un **bouton flottant** permet de remonter directement en haut une fois en bas de la page.



**Figure 71 : Aperçu du tableau de bord des statistiques**

- **Carte** : L'essence même de cette application était de disposer d'une cartographie. Celle-ci pouvait être interprétée de plusieurs façons. Mais au sens propre du terme, une cartographie reviendrait à disposer d'une carte. Et là, j'ai mis en place une carte interactive qui permet d'avoir la répartition visuelle des sièges des OSC sur le territoire togolais. Chaque bulle de position localise une OSC et en cliquant dessus, on peut obtenir quelques informations de l'OSC. Le menu de gauche permet de filtrer la liste de ces OSC en fonction des régions, secteurs d'activité et villes.



**Figure 72 : Aperçu de la carte**

- **OSC :** Cette page permet de consulter la liste des « OSC » sous forme de tableau. Ces informations « publiques » peuvent être téléchargées au delà d'être lues. À gauche, le menu permet de filtrer le tableau interactif des OSC en fonction de leurs secteurs d'activité. La recherche et d'autre option de filtrage sont déjà possible via la saisie de caractères dans le champ de recherche au dessus du tableau.

| Organisations de la Société Civile                                                            |                     |                           |           |            |         |
|-----------------------------------------------------------------------------------------------|---------------------|---------------------------|-----------|------------|---------|
| <a href="#">Visualiser sur la carte</a> <a href="#">Télécharger la liste complète des OSC</a> |                     |                           |           |            |         |
| Afficher 10 entrées                                                                           | Rechercher :        |                           |           |            |         |
| Dénomination                                                                                  | Sigle               | Responsable               | Siège     | Création   | Actions |
| CENTRALE                                                                                      |                     |                           |           |            |         |
| Regroupement des Associations pour le Développement Appliqué des Ruraux                       | RADAR               | PANA Tétoutokina Justin   | Sotouboua | 20-07-2001 |         |
| Appui au Développement et à la Santé Communautaire                                            | ADESCO              | M. DATAGNI Nadzombé       | Sokodé    | 16-12-1995 |         |
| Construire Ensemble                                                                           | CONSTRUIRE ENSEMBLE | KUMADUGUNA Katanama       | Sokodé    | 25-08-2002 |         |
| Club des Amis du Village                                                                      | CAV                 | GBELE-GUEWE Djobo Ousmane | Sokodé    | 01-01-1970 |         |

**Figure 73 : Aperçu du tableau des OSC**

En cliquant sur l'œil bleu de la colonne action de chaque OSC, on peut visualiser les détails de celle-ci :



Figure 74 : Visualisation des détails de l'OSC CREUSET TOGO

- **Projets** : Cette page permet de consulter la liste des « projets » sous forme de tableau. Tout comme avec les OSC, ces informations sont « publiques » et peuvent être téléchargées au delà d'être lues. À gauche, le menu permet de filtrer le tableau interactif des projets en fonction des secteurs d'activité qui les concernent. La recherche et d'autre option de filtrage sont déjà possible via la saisie de caractères dans le champ de recherche au dessus du tableau.

**Figure 75 : Aperçu du tableau des projets**

Tout comme pour une OSC, en cliquant sur l'œil sur la ligne d'un projet, on peut visualiser les détails d'un projet :

**Figure 76 : Visualisation des détails d'un projet**

- **Médiathèque** : Cette page permet de consulter la liste des « média » sous forme de tableau. À gauche, le menu permet de filtrer le tableau interactif des média en fonction des thématiques qu'ils développent. La recherche et d'autre option de filtrage sont déjà possible via la saisie de caractères dans le champ de recherche au dessus du tableau.

The screenshot shows the 'Médiathèque' section of the platform. At the top, there is a navigation bar with links: Accueil, Tableau de bord, Carte, OSC, Projets, Médiathèque, and Communauté. On the left, a sidebar has a 'Filtrer par thématique' dropdown, an 'Appliquer le filtre' button, and a 'REINITIALISER' button. The main area is titled 'Médiathèque' and displays a table of media entries. The table columns are: Titre, Thème, Auteur, Publication, and Détails. The first entry is 'Rapport sur la nutrition mondiale' by 'Global Nutrition Report' on 11-11-2024. The second entry is 'FEUILLE DE ROUTE POUR LA MISE EN PLACE D'UNE SOCIETE CIVILE UNIFIEE AU TOGO' by 'SUN' on 11-11-2024. Below the table, it says 'Affichage de 1 à 2 sur 2 entrées' and includes navigation buttons: Première, Précédente, 1, Suivante, and Dernière. There is also a 'Rechercher:' input field at the top right.

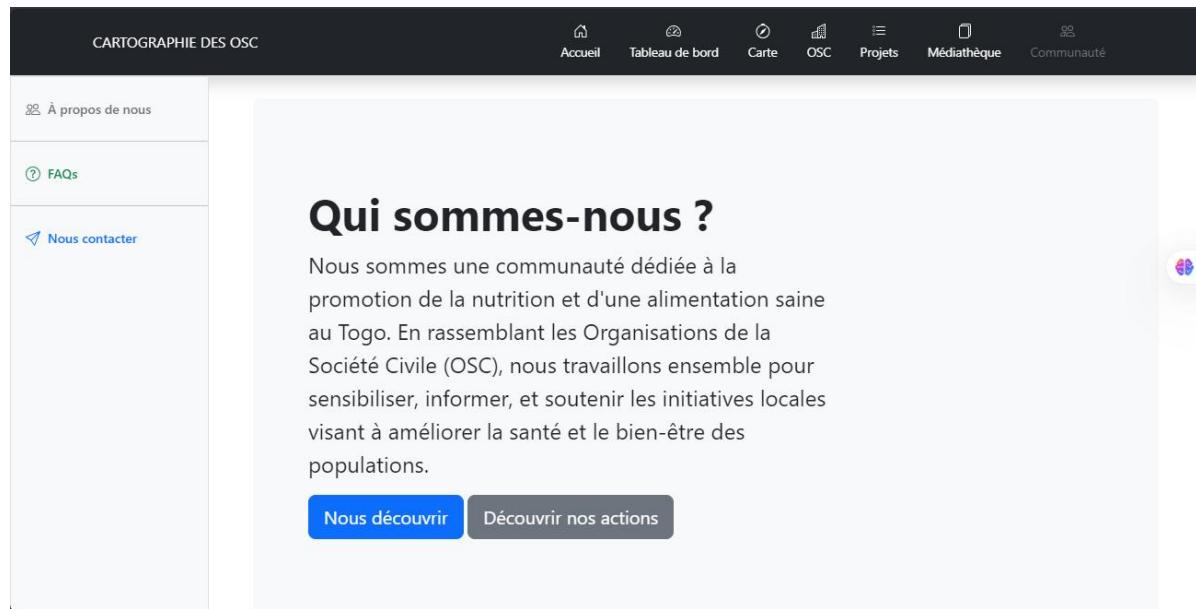
**Figure 77 : Aperçu de la médiathèque**

Comme dit précédemment pour la liste des OSC et celle des projets, ici aussi, l'œil bleu sert à visualiser les détails d'un média :

The screenshot shows a detailed view of the 'Rapport sur la nutrition mondiale' report. The title is at the top. Below it, there is a 'Description' section with the author listed as 'Global Nutrition Report' and the theme as 'Nutrition et Alimentation Saine'. The text describes the report's purpose and content. In the 'Fichier' section, it says 'Cliquez [ici](#) pour télécharger le document.' At the bottom right of the detail view is a 'Fermer' button. The background shows the same 'Médiathèque' interface as Figure 77.

**Figure 78 : Visualisation des détails d'un média**

- **À propos** : Étant avant tout une application web devant servir à la société civile œuvrant sur les questions de **Nutrition et Alimentation Saine**, cette page de l'espace communautaire se veut donc de présenter la société civile. Le menu de gauche permet de naviguer entre les pages de l'espace communautaire.



**Figure 79 : Espace communautaire - À propos de nous**

- **FAQs** : Toujours dans l'espace communautaire, cette page permet d'avoir des réponses à des questions fréquentes.

Figure 80 : Espace communautaire - FAQs

- **Nous contacter** : Il s'agit uniquement d'un formulaire. Il permet soit de poser une question, soit de soumettre une demande d'adhésion à la communauté des OSC bénéficiaires de cette application web de cartographie ou soit encore de signaler un problème lié à l'utilisation de l'application.

Figure 81 : Espace communautaire - Nous contacter

- **Se connecter** : C'est le formulaire de connexion. Il permet aux « Managers d'OSC » et à l'« Administrateur » de se connecter à l'application et disposer de leurs priviléges respectifs. Quiconque n'ayant pas de compte cliquera sur le lien sous-ligné bleu. Celui-ci le redirigera vers la page « Nous contacter » où il pourra soumettre sa demande d'adhésion.

Nous rejoindre maintenant !' and at the bottom right is the copyright notice '© EU4SUN - Togo 2024'."/>

The screenshot shows a light gray login form with a white background. At the top center, it says "Identifiez-vous". Below that is a light blue input field labeled "Nom d'utilisateur" with the value "arej\_manager". Underneath it is another light blue input field labeled "Mot de passe" with masked input. At the bottom of this section is a large blue rectangular button with the word "Connexion" in white. To the left of the bottom section, there is a link in blue text: "Pas encore membre ? [Nous rejoindre maintenant !](#)". At the very bottom of the page, centered, is the copyright notice "© EU4SUN - Togo 2024".

**Figure 82 : Page de connexion**

### 6.2.2. Manager d'OSC

En plus des pages et fonctionnalités disponibles pour l'utilisateur « Public », il dispose d'accès lui permettant de gérer les infos de son OSC, leurs projets, leurs média et leur logo.

- **Gérer « OSC<sup>1</sup> »** : Il s'agit d'un formulaire pré-rempli avec les informations de l'OSC. Le formulaire est par défaut verrouillé pour éviter les erreurs de manipulation. Ainsi, si le Manager nourrit l'ambition de mettre à jour les informations de son OSC, il n'aura qu'à cliquer sur le bouton jaune « Mettre à jour » qui est au dessus à droite du formulaire. Cette action déverrouillera le formulaire et le Manager pourra apporter les modifications nécessaires. Il pourra soit invalider ses modifications en cliquant sur le bouton rouge « Annuler » ; ceci reverrouillera le formulaire, ou soit il pourra les valider en cliquant sur le bouton bleu « Valider les modifications ».

---

<sup>1</sup> Pour chaque Manager d'OSC, « OSC » est remplacé par le sigle de son OSC.

CARTOGRAPHIE DES OSC

- [Accueil](#)
- [Tableau de bord](#)
- [Carte](#)
- [OSC](#)
- [Projets](#)
- [Médiathèque](#)
- [Communauté](#)

**PARAMETRES**

- [AREJ](#)
- [Projets](#)
- [Média](#)

[Changer le logo](#)

[Se déconnecter](#)

## AREJ

Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse

### Informations

[Mettre à jour](#)

|              |                                                                        |
|--------------|------------------------------------------------------------------------|
| Sigle        | RH nutrition                                                           |
| AREJ         | 0                                                                      |
| Email        | Site web                                                               |
|              |                                                                        |
| Siège        | Représentations                                                        |
| Cinkassé     | 1                                                                      |
| Dénomination | Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse |
| Responsable  | BATIYENPENI NAMETOUGLI                                                 |

[Obtenir ma position \[Uniquement, si vous êtes au siège !\]](#)

|                   |                    |          |           |
|-------------------|--------------------|----------|-----------|
| Latitude          | Longitude          | Altitude | Precision |
| 6,202628243633084 | 1,1942485010101713 | 0,0      | 81,0      |

[Annuler](#) [Valider les modifications](#)

**Figure 83 : Espace de gestion du Manager d'OSC - Gestion des informations de l'OSC**

➤ **Gérer projets** : Cette interface permet d'ajouter, modifier et supprimer des projets.

CARTOGRAPHIE DES OSC

- [Accueil](#)
- [Tableau de bord](#)
- [Carte](#)
- [OSC](#)
- [Projets](#)
- [Médiathèque](#)
- [Communauté](#)

**PARAMETRES**

- [AREJ](#)
- [Projets](#)
- [Média](#)

[Changer le logo](#)

[Se déconnecter](#)

## AREJ

Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse

### Projets (0)

[Ajouter](#)

Afficher  entrées

| Titre                                    | Durée | Statut | Actions |
|------------------------------------------|-------|--------|---------|
| Aucune donnée disponible dans le tableau |       |        |         |

Affichage de 0 à 0 sur 0 entrées [Première](#) [Précédente](#) [Suivante](#) [Dernière](#)

**Figure 84 : Espace de gestion du Manager d'OSC - Gestion des projets de l'OSC**

**a. Ajouter un projet :** Pour ce faire, on clique sur le bouton jaune « Ajouter » au dessus à droite du tableau des projets. Une formulaire d'ajout apparaît et le Manager le rempli.

The screenshot shows the AREJ application's project management interface. On the left, a sidebar with 'PARAMETRES' (AREJ, Projets, Média), 'Changer le logo', and 'Se déconnecter'. The main area displays 'Projets (0)' with an 'Ajouter' button. A central modal window titled 'Nouveau projet' is open, containing the following fields:

- Titre:** Test d'ajout projet
- Durée:** 1 jour
- Description:** Il permet de tester la fonction d'ajout de projet
- Secteurs:**
  - Auto. femme
  - Élevage
  - Environnement
  - Gouv. nutri.
  - Séc. alim.
  - Santé
  - EHA
  - Agroalimentaire
  - Pro. sociale
  - Agriculture
  - CC
  - Edu. nutri.
- Objectifs:** Ce projet devra permettre d'apprécier le fonctionnement de l'ajout d'un nouveau projet. Afin de voir au cas échéant, les
- Résultats (Optional):** Comme résultat, on attend que le projet soit ajouté à la liste des projets.
- Statut:** En cours

At the bottom right of the modal are 'Annuler' (Cancel) and 'Valider' (Validate) buttons. The background of the application shows a dark theme with a navigation bar at the top and a footer with links like 'Cartographie des OSC de Nutritio', 'À propos', 'Album', 'FAQs', and 'Nous contacter'.

**Figure 85 : Test d'ajout de nouveau projet - Remplissage du formulaire**

Une fois rempli, le Manager valide l'enregistrement en cliquant sur le bouton vert « Valider ». Le projet est ainsi ajouté à la liste des projets :

The screenshot shows the 'Projets (1)' list after a new project has been added. The table has columns: Titre, Durée, Statut, and Actions. The first row represents the newly added project:

| Titre               | Durée  | Statut   | Actions |
|---------------------|--------|----------|---------|
| Test d'ajout projet | 1 jour | En cours |         |

At the bottom of the list, there are navigation buttons: Première, Précédente, 1, Suivante, and Dernière.

**Figure 86 : Test d'ajout de nouveau projet - Nouveau projet ajouté avec succès à la liste des projets**

**b. Modifier un projet :** Pour modifier un projet, on clique sur le stylo bleu dans la colonne « Actions » du projet concerné. Le formulaire<sup>1</sup> de modification apparaît et le Manager apporte les modifications désirées.

The screenshot displays the 'Cartographie des OSC' application interface. On the left, a sidebar shows navigation links like 'PARAMETRES', 'AREJ', 'Projets', ' Média', 'Changer le logo', and 'Se déconnecter'. The main area shows a list of 'Projets (1)' with a single entry: 'Test d'ajout projet'. The 'Titre' field contains 'Test de modification projet', and the 'Duree' field shows '1 jour'. The 'Description' field contains the text 'Il permet de tester la fonction de modification de projet'. Under 'Secteurs', several checkboxes are checked: 'Pro. sociale', 'Agriculture', 'CC', and 'Agroalimentaire'. In the 'Objectifs' section, there is a note: 'Ce projet devra permettre d'apprécier le fonctionnement de la modification d'un projet. Afin de voir au cas échéant, les'. The 'Results (Optional)' section contains the text 'Comme résultat, on attend que les mises à jour proposées soient persistées.' The 'Statut' section shows 'Terminé' selected. At the bottom, there are 'Annuler' and 'Valider' buttons. To the right, a preview window shows the updated project details with a dark background.

**Figure 87 : Test de modification de projet - Remplissage du formulaire**

Une fois le formulaire validé, le projet est mis à jour ; comme le démontre son titre dans le tableau.

<sup>1</sup> Notons que ce formulaire peut permettre aussi de juste visualiser les détails d'un projet.

The screenshot shows the 'Projet' (Project) page of the 'CARTOGRAPHIE DES OSC' application. The top navigation bar includes links for Accueil (Home), Tableau de bord (Dashboard), Carte (Map), OSC, Projets (Projects), Médiathèque (Library), Communauté (Community), and a user icon. On the left, a sidebar titled 'PARAMETRES' contains links for AREJ, Projets, Média, Changer le logo (Change logo), and Se déconnecter (Logout). The main content area is titled 'AREJ' and displays the association's name: 'Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse'. Below this, a table titled 'Projets (1)' lists one project: 'Test de modification projet' with a duration of '1 jour' and status 'Terminé'. The table includes columns for Titre, Durée, Statut, and Actions (with edit and delete icons). Navigation buttons at the bottom show 'Affichage de 1 à 1 sur 1 entrées', 'Première', 'Précédente', a page number '1', 'Suivante', and 'Dernière'.

**Figure 88 : Test de modification de projet - Projet modifié avec succès dans la liste des projets**

- c. Supprimer un projet : Dans ce cas d'utilisation, on clique sur la poubelle rouge de la colonne « Actions ». Un popup de confirmation apparaît :

This screenshot is similar to Figure 88, showing the 'Projet' page. A modal dialog box is overlaid on the screen, asking '127.0.0.1:8000 indique Êtes-vous sûr de vouloir supprimer ce projet ?' (Are you sure you want to delete this project?). The dialog has 'OK' and 'Annuler' (Cancel) buttons. The background table and sidebar are visible but dimmed.

**Figure 89 : Test de suppression de projet - Popup de confirmation**

Après la confirmation, le projet est supprimé de la liste :

The screenshot shows the 'Projets' section of the 'AREJ' organization in the 'CARTOGRAPHIE DES OSC' application. The left sidebar includes links for 'PARAMETRES' (AREJ, Projets, Média), 'Changer le logo', and 'Se déconnecter'. The main content area displays the title 'AREJ' and its description 'Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse'. A table titled 'Projets (0)' shows no data with the message 'Aucune donnée disponible dans le tableau'. Below the table, there are buttons for 'Afficher 10 entrées', 'Rechercher', and 'Ajouter'. Navigation links at the bottom include 'Première', 'Précédente', 'Suivante', and 'Dernière'.

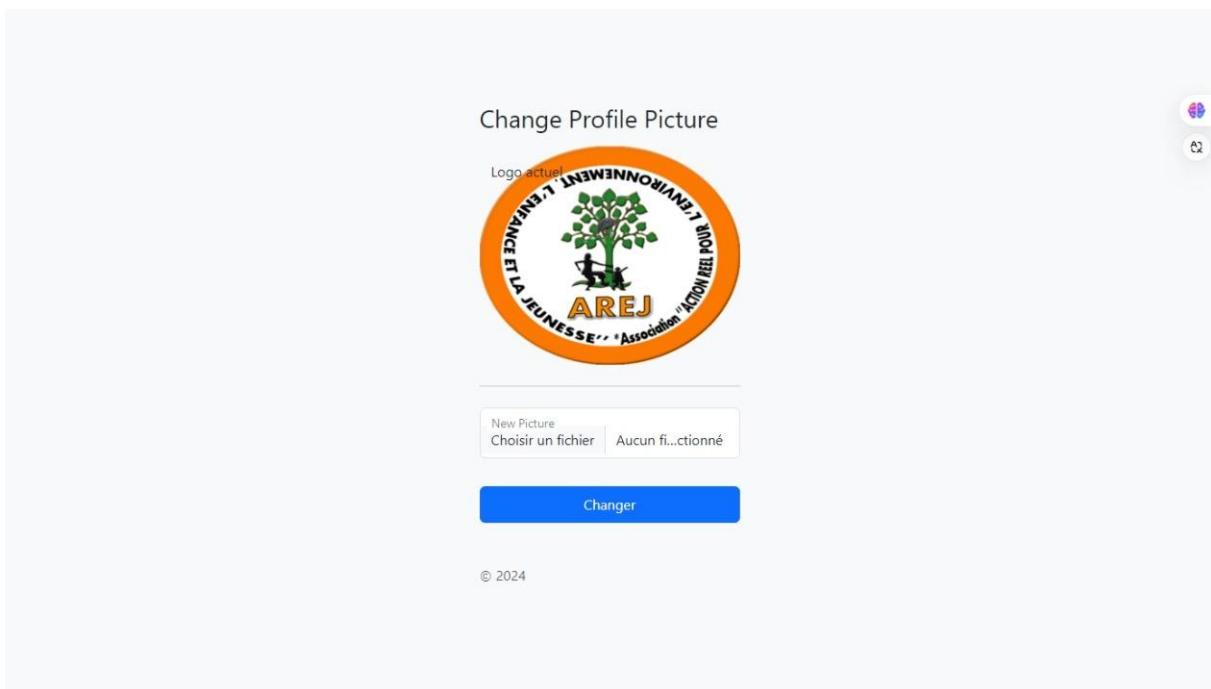
**Figure 90 : Test de suppression de projet - Projet supprimé avec succès de la liste des projets**

- **Gérer média :** Cette interface permet d'ajouter, modifier et supprimer des média. Sans se montrer inutilement redondant, on précisera que l'ajout, la modification et la suppression d'un média suit la même logique que celle précédemment présentée pour les projets. Donc il suffira de se référer aux descriptions précédentes.

The screenshot shows the 'Média' section of the 'AREJ' organization in the 'CARTOGRAPHIE DES OSC' application. The left sidebar includes links for 'PARAMETRES' (AREJ, Projets, Média), 'Changer le logo', and 'Se déconnecter'. The main content area displays the title 'AREJ' and its description 'Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse'. A table titled 'Média (0)' shows no data with the message 'Aucune donnée disponible dans le tableau'. Below the table, there are buttons for 'Afficher 10 entrées', 'Rechercher', and 'Ajouter'. Navigation links at the bottom include 'Première', 'Précédente', 'Suivante', and 'Dernière'.

**Figure 91 : Espace de gestion du Manager d'OSC - Gestion des média de l'OSC**

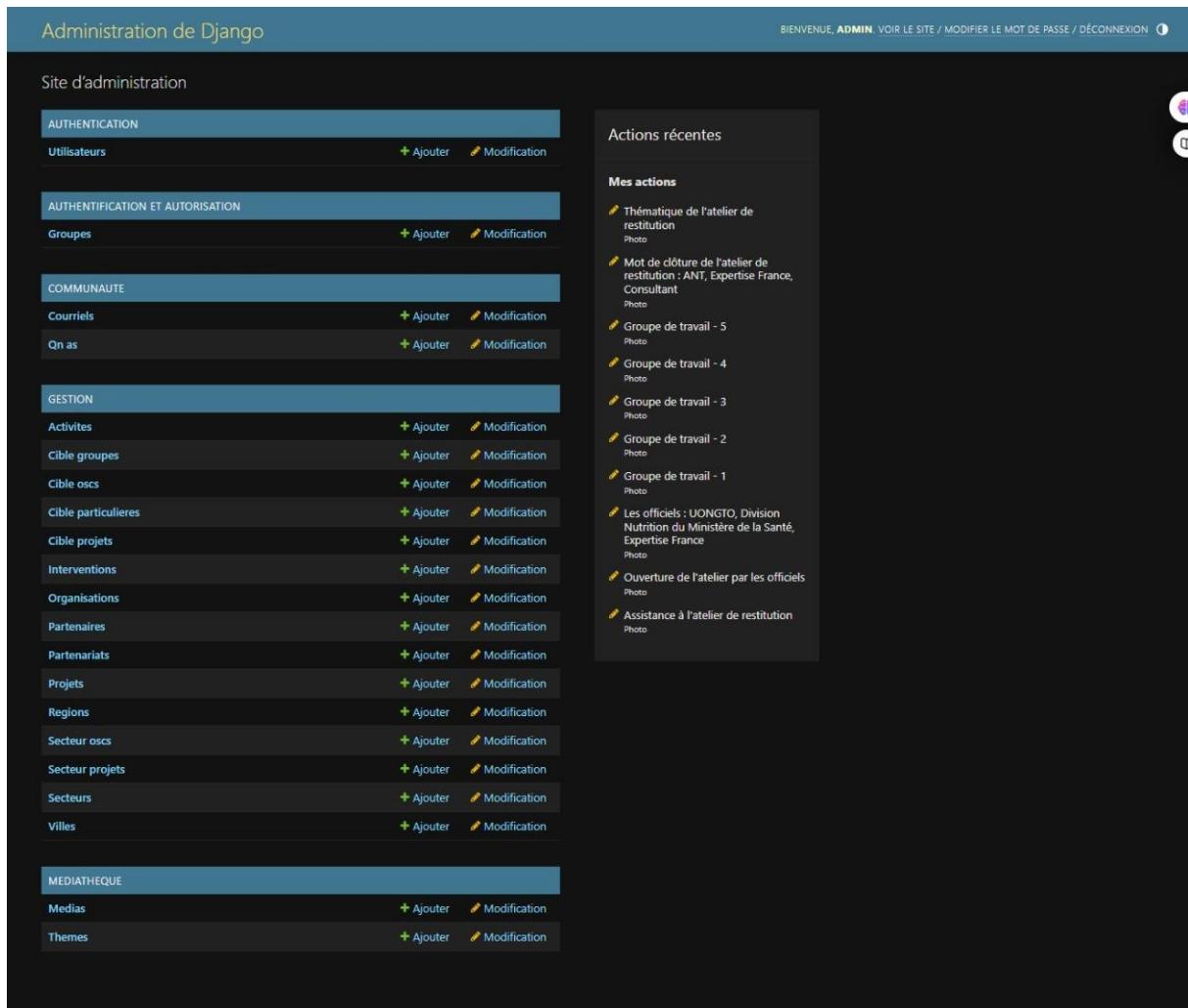
- **Changer le logo :** Cette interface permet de charger un nouveau logo au cas où l'OSC n'avait pas son logo enregistré ou que son logo actuel [enregistré] est obsolète. On clique donc sur le bouton « Choisir un fichier », on choisit une image (.jpg, .jpeg, .png) et on valide en cliquant sur le bouton bleu « Changer ».



**Figure 92 : Espace de gestion du Manager d'OSC - Gestion du logo de l'OSC**

### **6.2.3. Administrateur**

Tout comme le « Manager d'OSC », il bénéficie de tous les accès de l'utilisateur « Public ». En plus de ça, il dispose pour lui seul, d'une interface d'administration avancée fournie par Django.



**Figure 93 : Espace de gestion d'Administrateur - Accueil**

Ladite interface permet d'ajouter de nouvelles données, les modifier, et éventuellement les supprimer : elle permet de gérer la globalité des données du système ; des OSC aux Utilisateurs en passant par les projets, média, etc.

En cliquant sur un type de données (« Organisations » par exemple), on peut visualiser la liste complète des occurrences de ce type de données (dans notre exemple, la liste complète des « Organisations ») :

| Sélectionnez l'objet organisation à changer |    |                                                                      |                                                                        |                              |                                                           |
|---------------------------------------------|----|----------------------------------------------------------------------|------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------|
| Action :                                    |    | DENOUMENEMENT                                                        | SIGLE OFFICIEL                                                         | EMAIL                        | NUMERO                                                    |
| <input type="checkbox"/>                    | ID | 1                                                                    | Association Action Réelle sur l'Environnement l'Enfance et la Jeunesse | AREJ                         | -                                                         |
| <input type="checkbox"/>                    | 2  | Comité d'Action pour la Recherche et le Développement                | CARD                                                                   | mutecard@yahoo.fr            | 90145473/90330520 IL_MANQUE_LE_SITE_WEB                   |
| <input type="checkbox"/>                    | 3  | Développement- Initiative- Solidarité- Action Concertée- Coopération | DISACC                                                                 | disaccb@gmail.com            | +228 91519065 IL_MANQUE_LE_SITE_WEB                       |
| <input type="checkbox"/>                    | 4  | Programme d'Appui à la Femme et à l'Enfance Déshéritée               | PAFED                                                                  | pafed_togo@yahoo.com         | 25510251 https://www.pafedtogo.org                        |
| <input type="checkbox"/>                    | 5  | Action Base de Développement                                         | ABAD                                                                   | togoabad@gmail.com           | +228 90 95 02 55 / *228 99 45 08 79 IL_MANQUE_LE_SITE_WEB |
| <input type="checkbox"/>                    | 6  | Construire Ensemble                                                  | CONSTRUIRE ENSEMBLE                                                    | construire.ensemble@yahoo.fr | 90342129/ 90349838/ 70170427 IL_MANQUE_LE_SITE_WEB        |
| <input type="checkbox"/>                    | 7  | Association EZAME                                                    | E2AME                                                                  | e2ame.wsame@gmail.com        | 27 70 21 31 / 70773737 IL_MANQUE_LE_SITE_WEB              |
| <input type="checkbox"/>                    | 8  | Action pour la Jeunesse                                              | AJA                                                                    | ajatg@yahoo.fr               | Cel: 90724245/ 25 50 66 64 https://WWW.ajatg.org          |

**Figure 94 : Interface d'administration - Liste des Organisations**

Nous allons maintenant voir un aperçu de chacune des actions à savoir l'ajout, la modification et la suppression de donnée(s). On travaillera avec une « Organisation » mais le processus est le même pour tous les autres types de données.

- **Ajout :** On clique sur le bouton « AJOUTER ORGANISATION<sup>1</sup> » au dessus à droite du tableau. L'administrateur se retrouve sur un formulaire et il devra renseigner au moins les informations obligatoires ; à savoir celles dont les libellés sont en gras.

<sup>1</sup> Pour chaque type de données, on aura « AJOUTER TYPE\_DE\_DONNÉES ».

Administration de Django

Bienvenue, ADMIN. VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION

Accueil > Gestion > Organisations > Ajouter organisation

Ecrivez ici pour filtrer...

**AUTHENTICATION**

Utilisateurs + Ajouter

**AUTHENTIFICATION ET AUTORISATION**

Groupes + Ajouter

**COMMUNAUTE**

Courriels + Ajouter  
Qn as + Ajouter

**GESTION**

Activites + Ajouter  
Cible groupes + Ajouter  
Cible oscs + Ajouter  
Cible particulières + Ajouter  
Cible projets + Ajouter  
Interventions + Ajouter  
Organisations + Ajouter  
Partenaires + Ajouter  
Partenariats + Ajouter  
Projets + Ajouter  
Regions + Ajouter  
Secteur oscs + Ajouter

Ajout de organisation

Dénomination : OSC\_TEST

Sigle officiel : OSC de test

Responsable : Admin

Représentations : 0

Date création : 31/01/2025 Aujourd'hui |

Note : votre heure précède l'heure du serveur de 1 heure.

Date enregistrement : 31/01/2025 Aujourd'hui |

Note : votre heure précède l'heure du serveur de 1 heure.

Siège :

Rh nutrition : 0

Observations :

Numéro :

Email : email@email.com

Site web :

Longitude : 0

Latitude : 0

Altitude : 0

Precision : 0

Active

**ENREGISTRER** Enregistrer et ajouter un nouveau Enregistrer et continuer les modifications

**Figure 95 : Interface d'administration - Formulaire d'ajout d'une Organisation**

Une fois le formulaire rempli, l'administrateur valide en cliquant sur le bouton bleu « ENREGISTRER » en bas à gauche. Il est ensuite redirigé vers la page contenant la liste des OSC. Une bande transversale verte apparaît pour confirmer l'ajout effectif de la nouvelle organisation.

The screenshot shows the Django Admin interface for the 'Organisations' model. On the left, there's a sidebar with navigation links: 'Accueil', 'Gestion', 'Organisations', 'AUTHENTICATION' (with 'Utilisateurs' and '+ Ajouter'), 'AUTHENTIFICATION ET AUTORISATION' (with 'Groupes' and '+ Ajouter'), 'COMMUNAUTE' (with 'Courriels' and '+ Ajouter', 'Qn as' and '+ Ajouter'), and 'GESTION' (with 'Activites' and '+ Ajouter', 'Cible groupes' and '+ Ajouter', 'Cible oscs' and '+ Ajouter'). The main content area has a green success message: 'L'objet organisation « OSC de test » a été ajouté avec succès.' Below it, a table lists organizations. The table has columns: ID, DENOMINATION, SIGLE OFFICIEL, EMAIL, NUMERO, and SITE WEB. The rows show:

| ID | DENOMINATION                                              | SIGLE OFFICIEL | EMAIL                    | NUMERO              | SITE WEB              |
|----|-----------------------------------------------------------|----------------|--------------------------|---------------------|-----------------------|
| 77 | OSC_TEST                                                  | OSC de test    | email@email.com          | -                   | -                     |
| 74 | Cellule d'Appui des Producteurs Agricoles des Savanes     | CAPAS          | -                        | -                   | -                     |
| 73 | Programme d'Aide Humanitaire pour Cas Sociaux             | PAHCS          | -                        | -                   | -                     |
| 72 | Coopérative de Transformateur de Produit Agricole du Togo | COTPAT         | IL_MANQUE_L'ADRESSE_MAIL | IL_MANQUE_LE_NUMERO | IL_MANQUE_LE_SITE_WEB |

At the top right of the main content area, there's a button labeled 'AJOUTER ORGANISATION +'. The top right corner of the page shows a user profile and links for 'BIENVENUE', 'ADMIN', 'VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION'.

**Figure 96 : Interface d'administration - Ajout avec succès d'une Organisation**

- **Modification :** L'administrateur retrouve la position de l'OSC qu'il veut modifier et clique sur son numéro. Une fois fait, il est redirigé vers un formulaire identique au formulaire d'ajout. Sauf que cette fois-ci, le formulaire est pré-rempli avec les informations de l'OSC et attend des modifications de la part de l'utilisateur.

Administration de Django

Bienvenue, ADMIN. VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION

Accueil > Gestion > Organisations > OSC de test

Ecrivez ici pour filtrer...

**AUTHENTICATION**

- Utilisateurs [+ Ajouter](#)

**AUTHENTIFICATION ET AUTORISATION**

- Groupes [+ Ajouter](#)

**COMMUNAUTE**

- Courriels [+ Ajouter](#)
- Qn as [+ Ajouter](#)

**GESTION**

- Activites [+ Ajouter](#)
- Cible groupes [+ Ajouter](#)
- Cible oscs [+ Ajouter](#)
- Cible particulières [+ Ajouter](#)
- Cible projets [+ Ajouter](#)
- Interventions [+ Ajouter](#)
- Organisations [+ Ajouter](#)
- Partenaires [+ Ajouter](#)
- Partenariats [+ Ajouter](#)
- Projets [+ Ajouter](#)
- Regions [+ Ajouter](#)
- Secteur oscs [+ Ajouter](#)

**Modification de organisation**

**OSC de test**

**Dénomination :** OSC\_TEST\_MODIFICATION

**Sigle officiel :** OSC de test modifiée

**Responsable :** Admin

**Représentations :** 0

**Date création :** 31/01/2025 Aujourd'hui |

Note : votre heure précède l'heure du serveur de 1 heure.

**Date enregistrement :** 31/01/2025 Aujourd'hui |

Note : votre heure précède l'heure du serveur de 1 heure.

**Siege :** .....

**Rh nutrition :** 0

**Observations :** On met à jour les infos de l'OSC.

**Numéro :** +2280000000

**Email :** email@email.com

**Site web :**

**Longitude :** 0,0

**Latitude :** 0,0

**Altitude :** 0,0

**Precision :** 0,0

Active

**ENREGISTRER** **Enregistrer et ajouter un nouveau** **Enregistrer et continuer les modifications** **Supprimer**

**Figure 97 : Interface d'administration - Formulaire de modification d'une Organisation**

Après saisie des modifications, l'administrateur clique sur le bouton « ENREGISTRER ». Une fois de plus, il est redirigé vers la page des OSC avec une bande verte de confirmation.

The screenshot shows the Django admin interface for 'Organisations'. A success message at the top right says 'L'objet organisation « OSC de test modifiée » a été modifié avec succès.' Below it, a table lists several organizations. One row is highlighted in green, indicating the modified object. The table columns are ID, DENOMINATION, SIGLE OFFICIEL, EMAIL, NUMERO, and SITE WEB. The highlighted row shows ID 77, DENOMINATION 'OSC\_TEST\_MODIFICATION', SIGLE OFFICIEL 'OSC de test modifiée', EMAIL 'email@email.com', NUMERO '+2280000000', and SITE WEB '-'. Other rows include 'Cellule d'Appui des Producteurs Agricoles des Savanes' (ID 74), 'Programme d'Aide Humanitaire pour Cas Sociaux' (ID 73), 'Coopérative de Transformateur de Produit Agricole du Togo' (ID 72), and 'OEUVRE KOLPING DU TOGO' (ID 71). A sidebar on the left shows navigation links for AUTHENTICATION, COMMUNAUTE, and GESTION.

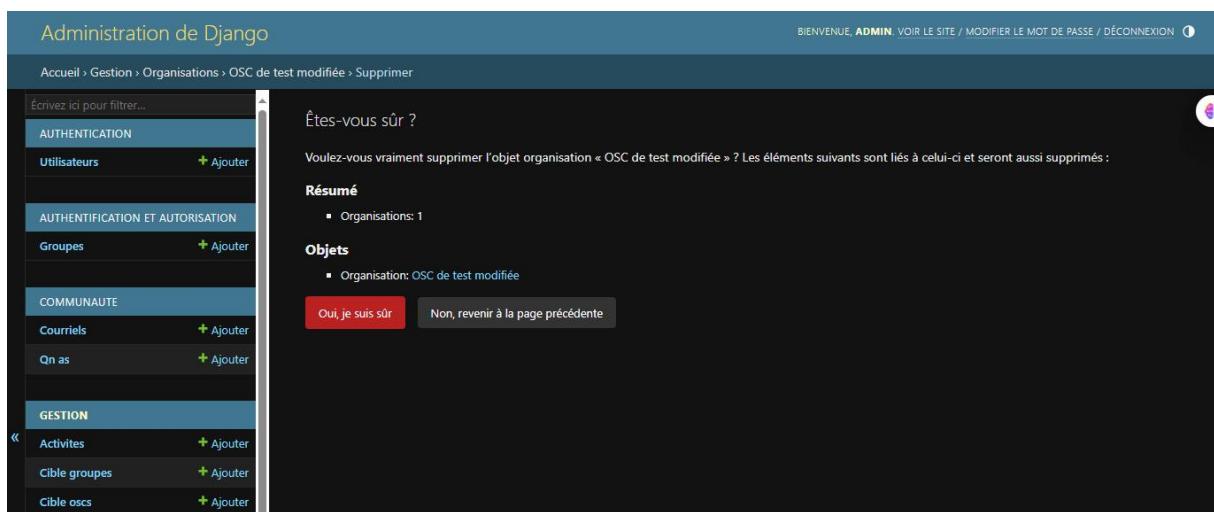
**Figure 98 : Interface d'administration - Modification avec succès d'une Organisation**

- **Suppression** : Pour supprimer une OSC, on peut soit utiliser le bouton rouge « Supprimer » en bas à droite du formulaire de modification, ou soit à partir du tableau des OSC, cocher la case au début de la ligne de l'OSC ou des OSC à supprimer, puis choisir « Supprimer les organisations sélectionnés » dans la liste déroulante du champ « Action » au dessus du tableau et là cliquer sur le bouton « Envoyer » à droite du champ.

The screenshot shows the Django admin interface for 'Organisations'. A dropdown menu 'Action' is set to 'Supprimer les organisations sélectionnés'. The table lists the same organizations as Figure 98. The first row, 'OSC\_TEST\_MODIFICATION', has a checked checkbox in the first column. The other rows have unchecked checkboxes. The table columns are ID, DENOMINATION, SIGLE OFFICIEL, EMAIL, NUMERO, and SITE WEB. The highlighted row shows ID 77, DENOMINATION 'OSC\_TEST\_MODIFICATION', SIGLE OFFICIEL 'OSC de test modifiée', EMAIL 'email@email.com', NUMERO '+2280000000', and SITE WEB '-'. Other rows include 'Cellule d'Appui des Producteurs Agricoles des Savanes' (ID 74), 'Programme d'Aide Humanitaire pour Cas Sociaux' (ID 73), 'Coopérative de Transformateur de Produit Agricole du Togo' (ID 72), and 'OEUVRE KOLPING DU TOGO' (ID 71). A sidebar on the left shows navigation links for AUTHENTICATION, COMMUNAUTE, and GESTION.

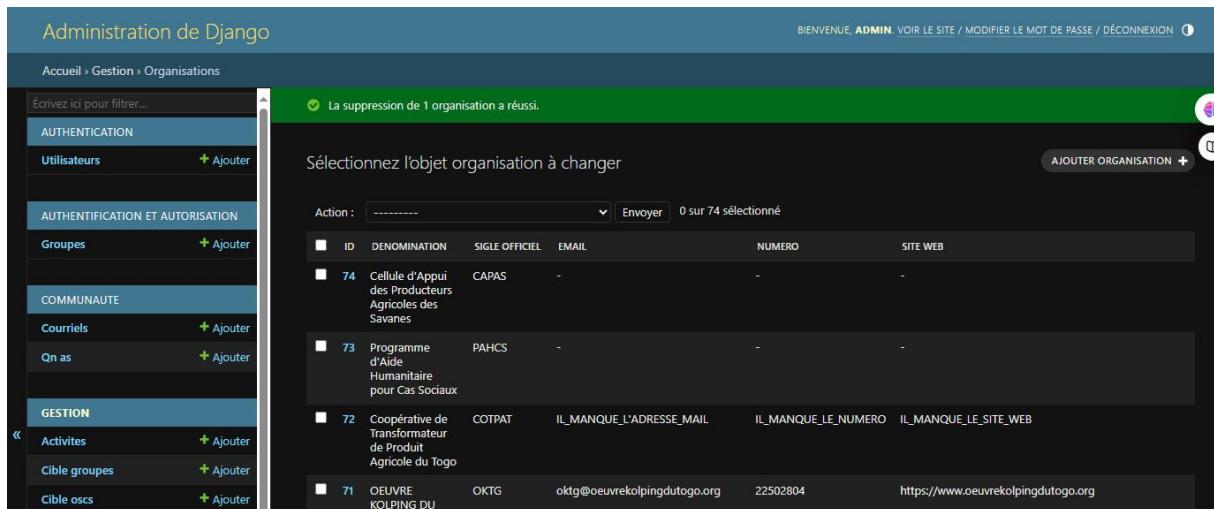
**Figure 99 : Interface d'administration - Sélection d'Organisation(s) à supprimer**

Dans les deux cas, l'administrateur est dirigé vers une page pour confirmer l'envie de suppression.



**Figure 100 : Interface d'administration - Demande de confirmation de suppression d'Organisation(s)**

Une fois avoir cliqué sur le bouton rouge « Oui, j'en suis sûr », il est redirigé vers la page des Organisations avec la bande verte de succès.



**Figure 101 : Interface d'administration - Suppression avec succès d'Organisation(s)**

Les principales manipulations ont été présentées. Les autres manipulations possibles sont basées sur le même squelette que celles-ci et ne souffrent d'aucune ambiguïté.

## Conclusion

Le guide d'utilisation présenté dans ce chapitre offre une vue d'ensemble complète sur l'organisation de l'application web et les modalités d'accès adaptées à chaque type d'utilisateur. En définissant précisément les rôles et responsabilités de chacun, il garantit une gestion fluide et sécurisée de la plateforme.

L'approche multi-niveaux avec des accès différenciés entre le « Public », les « Managers d'OSC » et l'« Administrateur » permet :

- **Une accessibilité étendue** pour la consultation des données.
- **Un contrôle rigoureux des informations** avec validation avant publication.
- **Une gestion collaborative des ressources** en favorisant l'implication directe des OSC.

En facilitant la compréhension de l'outil et en accompagnant chaque utilisateur dans ses interactions avec la plateforme, ce guide contribue à l'efficacité globale de la prise en main de l'**application web**. Il soutient ainsi les objectifs de transparence et d'amélioration continue dans la gestion des données des organisations de la société civile œuvrant pour la nutrition et l'alimentation saine au Togo.

# **CONCLUSION GÉNÉRALE**

Ce mémoire marque l'aboutissement de mes trois années de formation en génie logiciel à **iPNet Institute of Technology**. Durant ce parcours, j'ai eu l'opportunité de développer des compétences solides en ingénierie logicielle, de la conception à la mise en production d'applications, en passant par l'analyse des besoins et la gestion de projets.

Ma participation au projet **EU4SUN** au Togo, et plus précisément **le développement de la cartographie des Organisations de la Société Civile (OSC) dans le domaine de la nutrition et de l'alimentation saine**, a été une expérience déterminante. Ce projet m'a permis d'appliquer concrètement mes compétences techniques et de collaborer avec des acteurs clés du secteur, tout en contribuant à un projet d'impact social significatif.

L'élaboration de ce mémoire a représenté une véritable aventure, alliant rigueur technique et réflexion approfondie. J'espère qu'il pourra être une ressource utile et inspirante pour toute personne souhaitant s'engager dans des projets similaires ou approfondir ses connaissances dans le domaine du développement logiciel et de la gestion de données.

Enfin, avec la volonté de toujours progresser et d'apporter des solutions plus modernes et performantes, une évolution vers une « **version 2 de l'application** » est envisagée. Celle-ci intégrera **React** pour la partie **front-end**, **Django Rest Framework (DRF)** pour **l'API** et un **chatbot renforcé par l'IA**, afin de rendre l'application plus dynamique, maintenable, évolutive et d'apporter un support utilisateur de qualité.

Ce mémoire clôt une étape importante de mon parcours, tout en ouvrant de nouvelles perspectives de perfectionnement et d'innovation dans le domaine du développement logiciel.

## BIBLIOGRAPHIE ET WEBOGRAPHIE

Cette bibliographie et webographie regroupe les sources utilisées dans le cadre de ce mémoire, organisées en deux grandes catégories : les ressources relatives au travail de forme du mémoire et celles relatives à son travail de fond. Les dernières sont classées par thèmes selon les aspects abordés dans l'étude et le développement du projet.

### 1. Ressources relatives au travail de forme du mémoire

[1] Aline, 2024. "Comment rédiger le glossaire d'un mémoire ? Glossaire d'un mémoire : ce qu'il

faut y mettre et où le placer", *Prorédaction*, <https://proredaction.com/memoire/glossaire/> (page consultée le 13 mars 2025).

[2] BEAUD Michel, 1985. *L'Art de la thèse*, La Découverte, Paris, 160 p ; 5e éd., 2006, 202 p.

[3] DESMET Amélie, 2024. "Sommaire de mémoire : Comment faire un bon sommaire ?", *MemoRedaction*, <https://memoredaction.com/memoire/sommaire-memoire/> (page consultée le 20 fevrier 2025).

### 2. Ressources relatives au travail de fond du mémoire

Ordonnées par thèmes, on distingue :

#### ➤ Approches méthodologiques

[4] Contributeurs de Wikipédia, 2024. "Devops", *Wikipédia, l'encyclopédie libre*, <https://fr.wikipedia.org/wiki/Devops> (page consultée le 03 novembre 2024).

[5] Contributeurs de Wikipédia, 2024. "Méthode agile", *Wikipédia, l'encyclopédie libre*, [https://fr.wikipedia.org/wiki/M%C3%A9thode\\_agile](https://fr.wikipedia.org/wiki/M%C3%A9thode_agile) (page consultée le 03 novembre 2024).

[6] Contributeurs de Wikipédia, 2024. "Modèle en cascade", *Wikipédia, l'encyclopédie libre*, [https://fr.wikipedia.org/wiki/Mod%C3%A8le\\_en\\_cascade](https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade) (page consultée le 03 novembre 2024).

- [7] Contributeurs de Wikipédia, 2024. "Scrum (développement)", *Wikipédia, l'encyclopédie libre*, [https://fr.wikipedia.org/wiki/Scrum\\_\(d%C3%A9veloppement\)](https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement)) (page consultée le 03 novembre 2024).

## ➤ Architectures de code

- [8] Contributeurs de Wikipédia, 2023. "Modèle-vue-vue modèle", *Wikipédia, l'encyclopédie libre*, [https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-vue\\_mod%C3%A8le](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-vue_mod%C3%A8le) (page consultée le 29 novembre 2024).
- [9] Contributeurs de Wikipédia, 2024. "Modèle-vue-contrôleur", *Wikipédia, l'encyclopédie libre*, <https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur> (page consultée le 29 novembre 2024).
- [10] Vinayak Nishant, 2020. "Django MVT Architecture: A Fresh Take on Classic MVC". *AskPython*, <https://www.askpython.com/django/django-mvt-architecture> (page consultée le 29 novembre 2024).

## ➤ Architectures de déploiement

- [11] InvolveInInnovation, 2022. "Monolithic, N-Tier, Microservices explained with comparison and example - Software Architecture". *YouTube*, <https://www.youtube.com/watch?v=qbDkBPpmjJM> (page consultée le 29 octobre 2024).
- [12] KofiGroup, 2024. "Serverless Architecture Explained In 10 Minutes". *KofiGroup*, <https://www.kofi-group.com/serverless-architecture-explained-in-10-minutes/> (page consultée le 29 octobre 2024).
- [13] OpenClassrooms, 2024. "Définissez votre architecture logicielle grâce aux standards reconnus - Apprenez l'architecture orientée services". *OpenClassrooms*, <https://openclassrooms.com/fr/courses/7210131-definissez-votre-architecture-logicielle-grace-aux-standards-reconnus/7371141-apprenez-larchitecture-orientee-services> (page consultée le 29 octobre 2024).

➤ **Modèle de communication**

- [14] Contributeurs de Wikipédia, 2024. "Client-serveur", *Wikipédia, l'encyclopédie libre*, <https://fr.wikipedia.org/wiki/Client-serveur> (page consultée le 28 octobre 2024).