

20/06/2016

# Rapport de stage

Stage de développement web &  
mobile



Tuteur enseignant :

Mme Hanene AZZAG

---



Maître de stage :

M. Matthieu ROGER

---

MARLHENS Samuel

LANETCIE



## TABLE DES MATIÈRES

TABLE DES MATIÈRES .....	2
REMERCIEMENTS .....	4
INTRODUCTION .....	6
L'ENTREPRISE.....	8
La société LaNetCie .....	8
Principales activités.....	9
Organigramme et différents pôles de l'entreprise.....	10
Évolution de l'entreprise .....	11
La concurrence .....	11
INTÉGRATION DANS L'ENTREPRISE .....	14
MISSIONS.....	16
SIAH – Description des missions.....	16
SIAH – Travail préparatoire .....	18
Symfony.....	18
Symfony – Les bundles .....	21
Symfony – Les entités.....	22
Symfony – Les contrôleurs .....	22
Symfony – Les vues .....	23
SIAH – Ajout de module sur l'Intranet.....	24
SIAH – Application mobile .....	30
BILANS .....	38
Bilan professionnel.....	38
Bilan personnel.....	40
CONCLUSION .....	42
LEXIQUE .....	44
RÉFÉRENCES .....	48
ANNEXES.....	50
Annexe 1 - Organigramme .....	51
Annexe 2 - Architecture de la base de données.....	52
Annexe 3 - Liste des tampons.....	53
Annexe 4 - Modification de l'ouvrage tampon Par.1 .....	54
Annexe 5 - Affichage des informations de l'ouvrage tampon Par.1.....	55

Annexe 6 - Affichage des photos des visites de l'ouvrage tampon Par.1.....	56
Annexe 7 - Affichage de l'historique des visites de l'ouvrage tampon Par.1 .....	57
Annexe 8 - Filtre sur la liste des visites.....	58
Annexe 9 - Activité présentant la carte.....	59
Annexe 10 - Activité présentant le formulaire de recherche d'ouvrage.....	60
Annexe 11 - Activité présentant les types d'ouvrage.....	61
Annexe 12 - Activité présentant les résultats de la recherche sur le code "Pa" .....	62
Annexe 13 - Activité présentant les informations d'un ouvrage.....	63
Annexe 14 - Activité présentant l'historique des visites d'un ouvrage .....	64
Annexe 15 - Activité création/modification de visite.....	65
Annexe 16 - Activité d'ajout de pollution(s).....	66
Annexe 17 - Activité nouvelle visite avec des photos .....	67
Annexe 18 - Activité d'édition du commentaire d'une photo.....	68
Annexe 19 - Activité de sélection des visites à envoyer au serveur.....	69

## REMERCIEMENTS

---

### **À Madame H. Azzag**

Pour m'avoir suivi tout au long de ce stage et m'avoir apporté ses conseils.

### **À l'ensemble de mes professeurs**

Pour m'avoir apporté les connaissances nécessaires au bon déroulement de ce stage et à la poursuite de mes études.

### **À Madame S. Cardoso**

Pour m'avoir aidé et accompagné dans ma recherche de stage.

### **À Monsieur M. ROGER**

Pour m'avoir permis d'effectuer ce stage de développement web & mobile au sein de LaNetCie.

### **À Monsieur M. KAISER**

Pour m'avoir guidé tout au long du stage et m'avoir apporté son savoir-faire et beaucoup de connaissances.

### **À l'ensemble des employés de LaNetCie**

Pour leur accueil et leur sympathie tout au long de ce stage.



## INTRODUCTION

---

Dans le cadre de ma deuxième année de DUT Informatique, j'effectue un stage de développement web & mobile au sein de l'entreprise LaNetCie. LaNetCie est une société de services en informatique, qui propose ses compétences dans l'infogérance\*, le développement applicatif, la téléphonie VOIP\* et la vidéo protection.

Mon maître de stage, M. Matthieu ROGER, directeur associé de la société, dirige l'équipe de développeurs composée d'un développeur senior web & mobile, M. Morgan KAISSE, d'un développeur & administrateur système M. Luiz DE ANDRADE, d'un apprenti, M. Valentin HUYENH et de deux stagiaires en développement web & mobile : Alex Couppoussamy et moi-même.

Mon stage a débuté par une phase d'intégration, présentation des employés de l'entreprise, présentation du projet auquel je participe actuellement, installation des logiciels nécessaires au développement et clonage du projet Git\* pour récupérer les fichiers sources.

Ma mission, réalisée pour le SIAH\* (Le Syndicat Intercommunal pour l'Aménagement Hydraulique des vallées du Croult et du Petit Rosne), a été de développer un nouveau module de gestion sur leur Intranet déjà existant.

Vous pourrez suivre, au travers de ce rapport, mon évolution tout au long du stage ainsi que les différentes missions que j'ai pu effectuer. Vous y trouverez également une description de l'entreprise, mon ressenti sur ce stage ainsi que ce qu'il m'a apporté personnellement et professionnellement.

Tous les mots suivis d'un \* sont définis dans le lexique page 42.



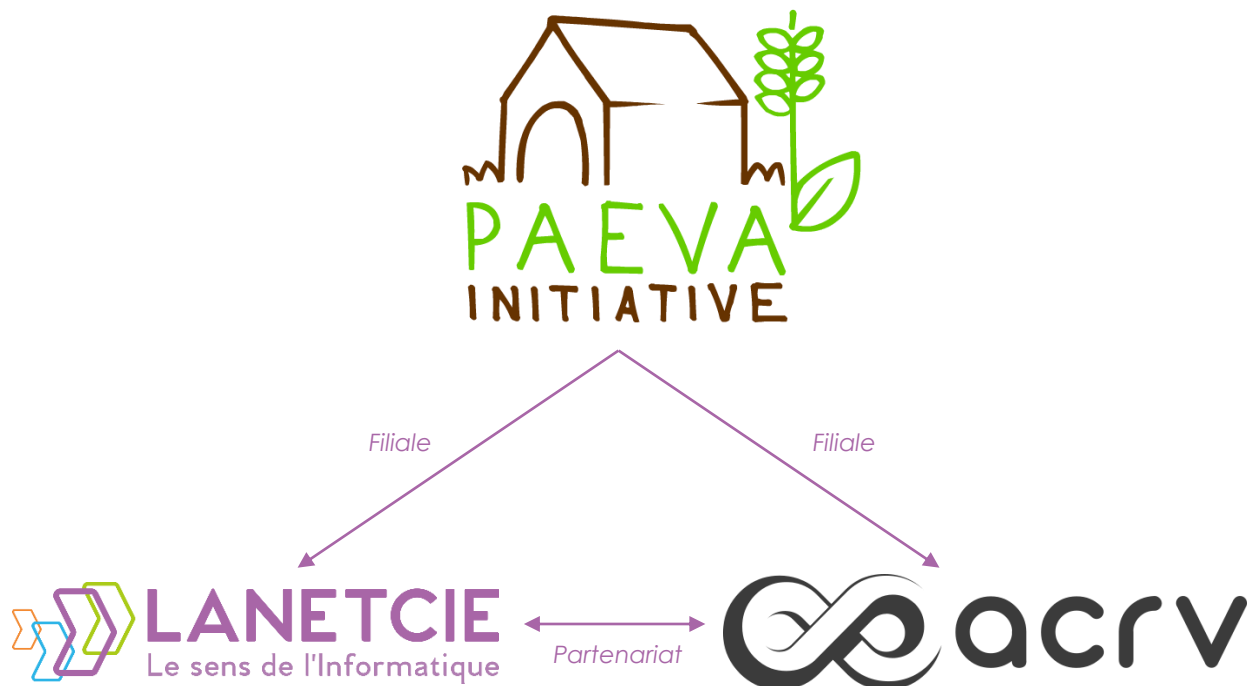


## L'ENTREPRISE

### La société LaNetCie

LaNetCie, créée en 2000 par M. Matthieu ROGER, est une société de service à taille humaine qui propose ses compétences dans l'infogérance, le développement applicatif, la téléphonie VOIP et la vidéoprotection.

LaNetCie est une filiale du groupe Paeva Initiative, dirigé par M. Matthieu ROGER et M. Stéphane SIMPER, et une entreprise partenaire de la société ACRV, agence de création graphique & de développement web, créée par M. Julien BODECHON.



LaNetCie avait été créée pour répondre à la forte demande de création de sites Internet et leur hébergement. Cependant, fin 2000, ce besoin s'estompe considérablement à cause de l'éclatement de la bulle Internet et des nouvelles technologies poussant alors l'entreprise à varier ses activités notamment dans le hardware.

LaNetCie a très vite évolué en même temps que les besoins de ses clients. Avec la généralisation de l'utilisation de la messagerie électronique, les sociétés clientes souhaitaient s'équiper d'une solution fiable et économique. Elles profitaient également de ce nouvel outil pour repenser en profondeur leur système d'information.

La compétence que LaNetCie avait acquise dans les technologies système et réseau lui a permis d'accompagner ses clients vers de nouveaux horizons informatiques.

Fort de la confiance établie avec ses clients, LaNetCie put leur proposer l'infogérance de leur parc informatique, la fourniture de matériels, de l'assistance aux utilisateurs, ainsi que de la maintenance.






En 2007, avec la généralisation des technologies IP, LaNetCie a étendu son panel d'offres avec les solutions de téléphonie sur IP : déploiement des téléphones, assistance et maintenance sur le système téléphonique.

Toujours 2007, la société fusionne avec l'entreprise Cyclope dirigée alors par M. Stéphane SIMPER qui deviendra cogérant de LaNetCie avec M. Matthieu ROGER. Cette fusion permettra d'élargir les horizons et les compétences de LaNetCie notamment avec la mise en place de nouvelles activités comme la vidéoprotection ou la téléphonie IP.

S'adaptant aux différents besoins qui ont pu survenir au fil des années, LaNetCie est alors devenue une entreprise polyvalente qui propose aujourd'hui des services informatiques très variés.

### Principales activités

On retrouve donc comme principales activités :

-  Le développement applicatif
-  La vente de matériel informatique
-  L'infogérance (du poste de travail et systèmes/réseaux)
-  La vidéo protection et Téléphonie IP
-  Le conseil en informatique

L'infogérance consiste à externaliser la gestion du secteur informatique d'une entreprise qui sera alors confiée à des prestataires extérieurs spécialisés dans ce domaine. Elle comprend la gestion du parc informatique, l'hébergement de sites Internet et messagerie d'entreprise, la sauvegarde de données et la maintenance permanente des serveurs.

LaNetCie est aussi sollicitée pour des développements web et logiciels répondant à un besoin spécifique. Fort de sa capacité à s'adapter aux différentes demandes, elle propose des solutions adaptées et personnalisées pour chaque client.




Elle propose aussi un support constant à distance ou sur place pour aider les utilisateurs sur les logiciels développés ou sur les problèmes techniques qui peuvent survenir dans une entreprise.

Enfin, la téléphonie sur IP et la vidéosurveillance IP permettent de proposer de nouveaux services comme l'envoi de SMS en tant qu'e-mail et la protection par vidéosurveillance, tout cela géré et maintenu par la société elle-même.

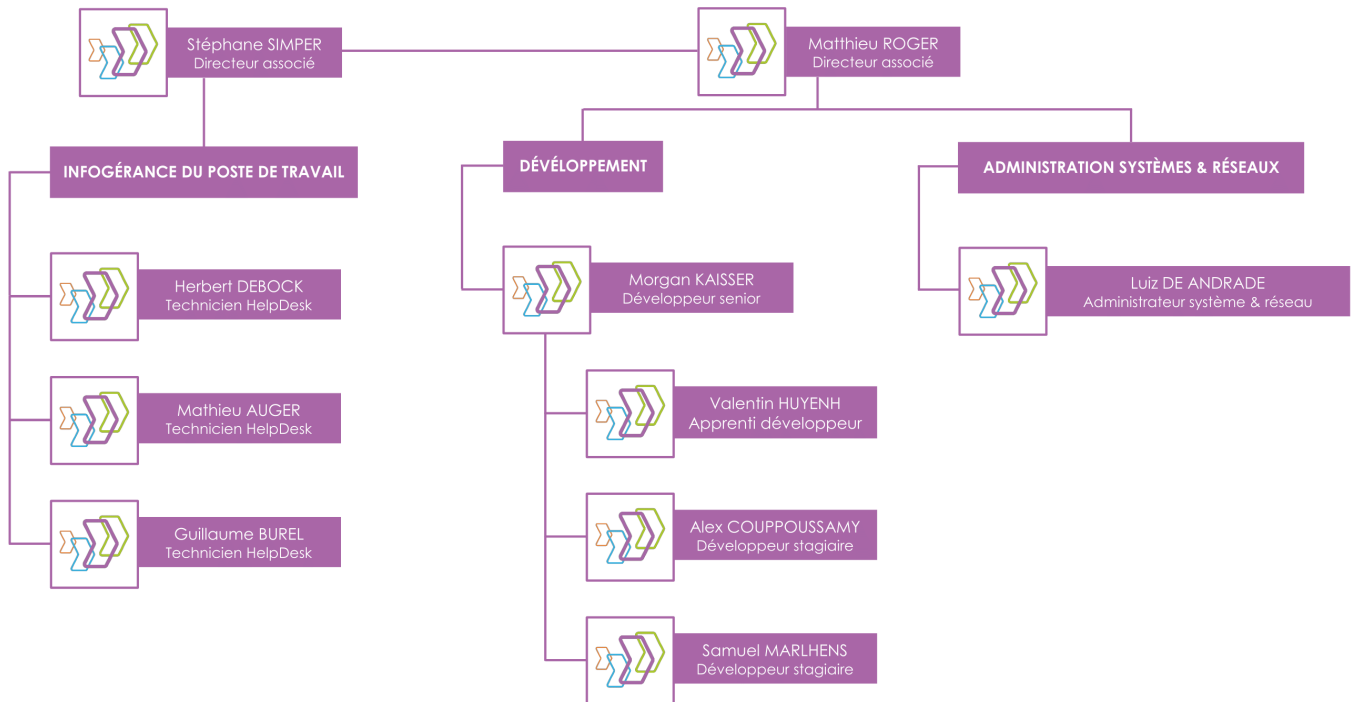
LaNetCie est donc une entreprise qui a la volonté d'accompagner ses clients en les guidant au mieux et en répondant aux différents besoins qui peuvent survenir avec l'évolution des technologies.

## Organigramme et différents pôles de l'entreprise

LaNetCie se divise en plusieurs pôles :

-  L'infogérance du poste de travail
-  Le développement
-  L'administration système & réseau

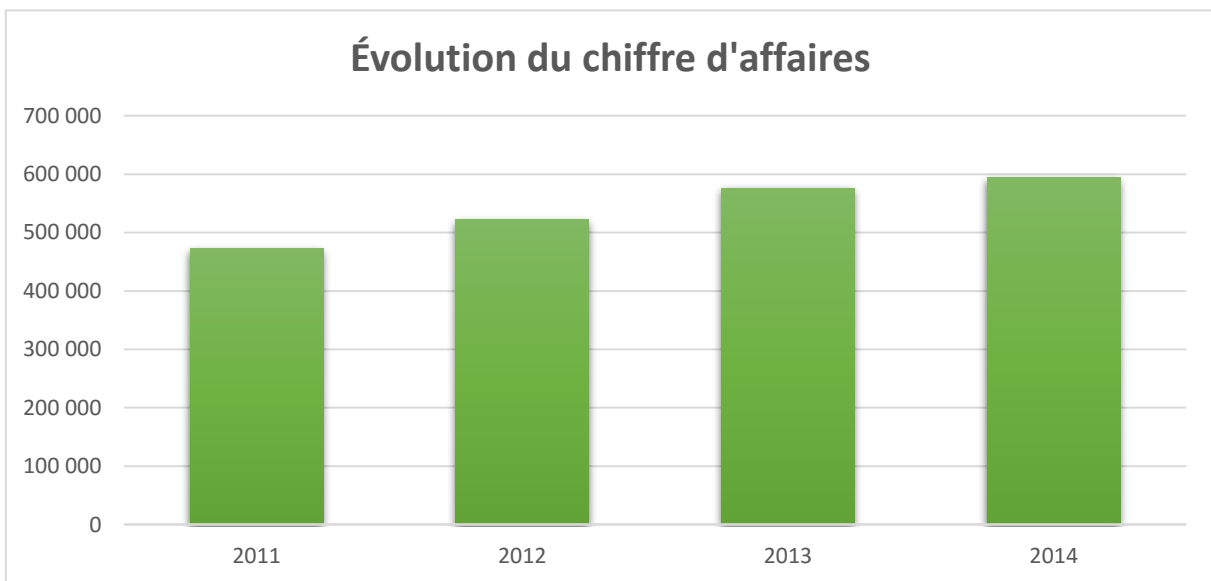
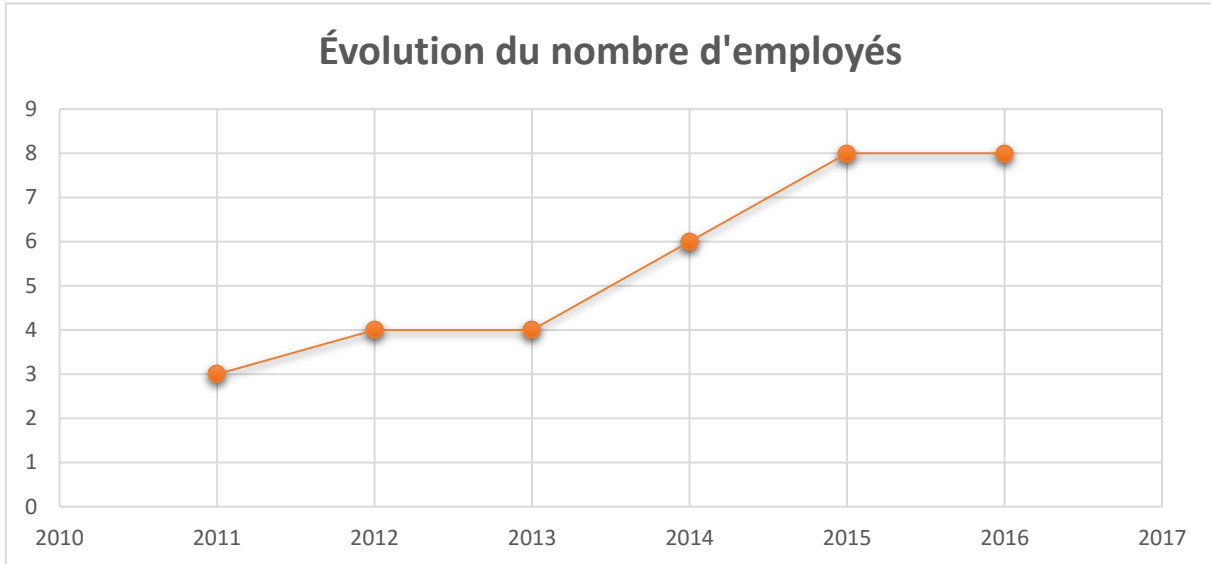
Voici donc l'organigramme de l'entreprise :



(Ci-dessus – Organigramme, Cf. Annexe 1 – page 51)

## Évolution de l'entreprise

LaNetCie est une société à taille humaine en constante évolution. Grâce à la fusion de Cyclope et de LaNetCie et l'évolution des technologies, LaNetCie a vu son nombre d'employés évoluer ainsi que son chiffre d'affaires.



## La concurrence

LaNetCie se démarque de sa concurrence par sa taille et ses méthodes de travail. En effet LaNetCie n'emploie pas de commercial pour démarcher de nouveau client, mais est force de proposition auprès de ses fidèles clients. Autrement dit, la réputation de LaNetCie augmente avec la réussite de ses projets et la satisfaction de ses clients. La fidélisation des clients est importante pour LaNetCie et n'est pas en concurrence avec d'autres sociétés de services, car les clients de LaNetCie ne font pas d'appel d'offres pour leurs différents projets et lui font confiance.

Voici une liste d'entreprises potentiellement concurrentes dans différents secteurs de l'informatique se situant dans un rayon de quelques kilomètres.

<b>Nom de l'entreprise</b>	<b>Principale(s) activité(s)</b>
ACJ-ECO	Déploiement de serveurs Infogérance Maintenance & Dépannage
BSA-Info	Développement web Infogérance Maintenance & Dépannage Vidéo surveillance
WSI	Infogérance Maintenance Sauvegarde en ligne
France Computer Support	Développement & Ingénierie Formation & Insertion Maintenance & Dépannage



## INTÉGRATION DANS L'ENTREPRISE

J'ai effectué mon stage dans les locaux de LaNetCie dans l'open-space\* dédié aux développeurs. Il se présente comme un open-space composé de six bureaux. L'open-space est un avantage certain, car il facilite la communication entre les membres de l'équipe et permet de poser plus aisément ses questions.

Pendant la durée de mon stage, j'ai utilisé un ordinateur sous Windows 7 ayant des accès à deux serveurs distants nommés « Bitzer » & « Shaun ». Le premier a servi au développement de la mission sur l'Intranet, et le deuxième pour le développement de l'application mobile.

De plus, j'avais un répertoire personnel sur le serveur « Bitzer » me permettant d'héberger le projet pour l'Intranet et de voir le résultat en ligne grâce à une URL\* spécifique accessible seulement depuis le réseau de l'entreprise. L'avantage de cette méthode est de pouvoir avoir un aperçu du résultat rapidement.

Durant mon stage j'ai utilisé différents logiciels et outils permettant le développement d'applications web et mobiles et je voulais vous les présenter.



**Android Studio** est un environnement de développement pour développer des applications Android\*.

Il m'a permis d'accéder aux codes sources de la première application Android développée pour SIAH l'an passé.



**MySQL Workbench** (anciennement MySQL administrator) est un logiciel de gestion et d'administration de bases de données MySQL\* créé en 2004. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données.

Il m'a permis de modéliser la structure de la base de données que nous avons utilisée.



**SQLite Browser** est un outil open source qui crée, met en forme, et modifie des fichiers de type base de données compatible avec SQLite\*. C'est un logiciel pour les utilisateurs et les développeurs qui veulent créer des bases de données, rechercher ou encore modifier des données dans ces dernières.

Il m'a permis de vérifier la structure et les données de bases de données SQLite stockées par l'application mobile que nous avons développée.



**Team Viewer** est un logiciel de téléassistance\* et d'accès à distance mondialement connu.

Il m'a permis de prendre la main sur mon poste de travail depuis la salle de réunion et de présenter l'évolution de mon travail.



phpPgAdmin

**PhpPgAdmin** est une application web réalisée en langage PHP\* destinée à faciliter la gestion du SGBD\* PostgreSQL\*. Ce logiciel libre est distribué sous licence GNU GPL\*.

Il m'a permis de consulter la base de données de l'Intranet que nous avons développé.



**PhpMyAdmin** (PMA) est une application web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

Il m'a permis au début de ma mission sur l'Intranet de consulter la base de données puis nous avons migré sur une base PostgreSQL.



**Microsoft Visual Studio** est une suite de logiciels de développement pour Windows conçue par Microsoft. Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET\*, des services web XML\*, des applications bureautiques et des applications mobiles.

Il m'a permis de développer l'application mobile pour le SIAH.



**PuTTY** est un émulateur de terminal doublé d'un client pour les protocoles SSH\*.

Il m'a permis de me connecter aux serveurs distants « Bitzer » & « Shaun ».



**Git** est un logiciel de gestion de versions décentralisé.

Il m'a permis de travailler tout en gardant une trace des modifications apportées successivement. Il permet également de travailler à plusieurs sur un même projet, ce qui est le cas durant mon stage.



**PhpStorm** est un éditeur pour PHP, HTML et JavaScript, édité par JetBrains. Il contient une correction automatique des erreurs.

Il m'a permis de développer le nouveau module pour l'Intranet.



**Xamarin** est un outil de développement cross-platform\* qui permet de créer des versions Windows, iOS et Android d'une même application en utilisant un seul langage.

Il m'a permis de générer et de développer l'application mobile sous Android.



## MISSIONS

---



### SIAH – Description des missions

Pendant mon stage, deux principales missions m'ont été confiées pour un même client, le Syndicat Intercommunal pour l'Aménagement Hydraulique (SIAH) des vallées du Croult et du Petit Rosne.

Le SIAH, créé en 1945, regroupe 33 communes et une Communauté d'Agglomération situées dans le Val-d'Oise dont le secteur est défini par les bassins versants des cours d'eau du Croult et du Petit Rosne (situé au sud-est du Val d'Oise). Les objectifs du SIAH sont de lutter contre la pollution des eaux et contre les inondations. Pour cela, ils ont pour tâches l'assainissement des eaux usées, l'entretien des cours d'eau et réseaux ainsi que la mise en place de nouvelles structures.

Le SIAH est un client de longue date de LaNetCie qui avait déjà bénéficié de leurs services lors de la création de l'Intranet. LaNetCie a par la suite détecté un nouveau besoin pour les techniciens du SIAH et a décidé de développer un PoC\* (Proof of Concept) pour un nouveau module sur l'Intranet et une nouvelle application mobile.

Une preuve de concept ou POC (acronyme de l'anglais « proof of concept »), ou encore démonstration de faisabilité, est une réalisation courte ou incomplète d'une certaine méthode ou idée pour démontrer sa faisabilité. La preuve de concept est habituellement considérée comme une étape importante sur la voie d'un prototype pleinement fonctionnel.

L'objectif de ce nouveau module est de pouvoir gérer les visites sur les ouvrages. Un ouvrage est un élément qui constitue le réseau de traitement des eaux. Par exemple, un ouvrage peut être une bouche d'égout, un fossé, un cours d'eau ou encore une canalisation d'eau. Une visite sera ici une inspection de l'état d'un ouvrage en particulier. La visite résulte de l'inspection effectuée par un agent du SIAH. Des pollutions peuvent être liées à un ouvrage si elles sont détectées lors d'une visite. Les pollutions peuvent être gérées sur ce nouveau module également.

L'application mobile a pour objectif de faciliter le travail des agents du SIAH équipés de tablettes en leur permettant, une fois sur un ouvrage, de créer une visite, de saisir les informations correspondantes, et de l'envoyer au serveur.

En parallèle de la partie développement, LaNetCie voudrait présenter un outil au SIAH qui serait lié au nouveau module de l'Intranet : QGIS\*. QGIS est un logiciel SIG (système d'information géographique) libre multiplate-forme publié sous licence GPL. Le développement a débuté en mai 2002 et est sorti en tant que projet sur SourceForge en juin 2002. Il était également appelé Quantum GIS jusqu'à la version 1.9. La version 2.14 (Essen) est sortie fin février 2016.

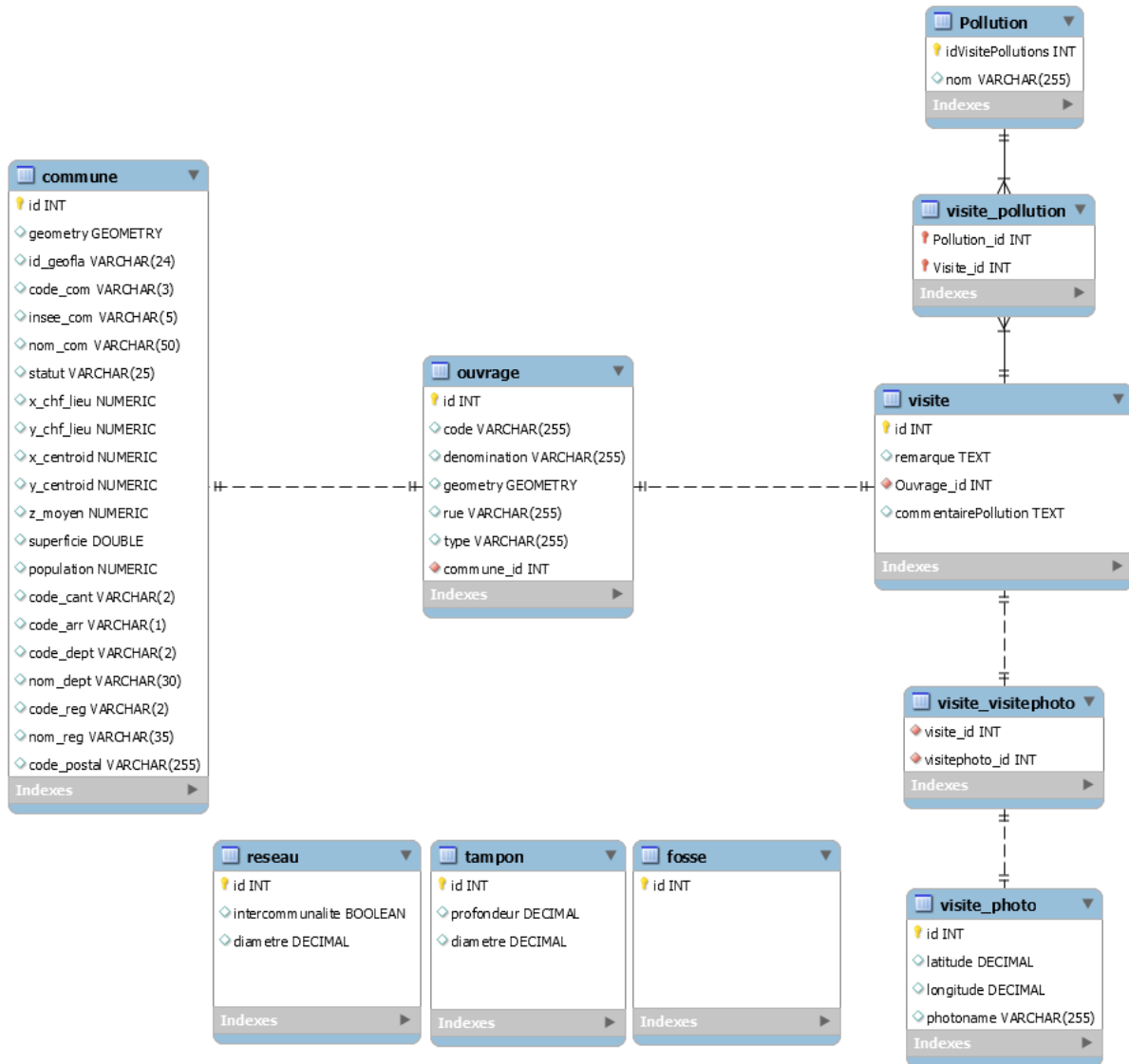
Logiciel que le SIAH n'utilise pas actuellement et qui serait lié à la base de données de l'Intranet, ce qui faciliterait le travail des agents du SIAH.

Ayant participé au développement du nouveau module web et de l'application mobile, je vais donc présenter ce que j'ai développé au cours de ce stage.



SIAH – Travail préparatoire

Avant de commencer le développement sous Symfony2\*, nous avons commencé par modéliser les différentes tables de base de données et les liens entre elles.



(Ci-dessus – Architecture de la base de données, Cf. Annexe 2 – page 52)

Sur ce schéma, les tables fossé, réseau et tampon ne sont pas liées à la table ouvrage, car elles héritent de cette classe et c'est Doctrine\*, inclus dans Symfony, qui fera le lien entre ces classes.

Symfony

Symfony est un framework MVC\* (Modèle, Vue, Contrôleur) libre écrit en PHP. Comme tout framework, il facilite et accélère le développement de sites et d'applications Internet et Intranet.

Doctrine est un ORM\* (couche d'abstraction à la base de données) pour PHP. Il s'agit d'un logiciel libre sous licence GNU LGPL. Doctrine est l'ORM par défaut du framework Symfony (depuis la version 1.3 de ce framework).

Un mapping objet-relationnel (en anglais object-relational mapping ou ORM) est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. On pourrait le désigner par « correspondance entre monde objet et monde relationnel ».

Avec Doctrine, la relation d'héritage entre les classes est possible. Les classes fossé, réseau et tampon héritent de la classe ouvrage. Ces trois classes possèdent donc les mêmes attributs que la classe ouvrage.

```
abstract class Ouvrage
{
    const TYPE_TAMPON = 'Tampon';
    const TYPE_RESEAU = 'Réseau';
    const TYPE_FOSSE = 'Fossé';
    /**
     * @var int
     * @ORM\Column(name="id", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="IDENTITY")
     */
    protected $id;

    /**
     * @var string
     * @ORM\Column(name="code", type="string", length=255)
     * @Assert\NotBlank(message="Le champ ne peut pas être vide")
     * @Gedmo\Versioned
     */
    protected $code;

    /**
     * @var string
     * @ORM\Column(name="denomination", type="text")
     * @Gedmo\Versioned
     * @Assert\NotBlank(message="Le champ ne peut pas être vide")
     */
    protected $denomination;

    /**
     * @ORM\Column(type="geometry", options={"srid":4326})
     * @Gedmo\Versioned
     */
    protected $geometry;

    /**
     * @var string
     * @ORM\Column(name="rue", nullable=true)
     * @Gedmo\Versioned
     */
    protected $rue;

    /**
     * @var \DateTime $deletedAt
     * @ORM\Column(name="deleted_at", type="datetime", nullable=true)
     * @Gedmo\Versioned
     */
    protected $deletedAt;
}
```

L'entité ouvrage dans Symfony est donc implémentée comme ci-dessus

Par exemple la ligne `@var string` correspond à la définition du type du champ qui sert lors du développement lors de l'affectation de valeur, la ligne `@ORM\Column` correspond à la définition du champ dans doctrine. La ligne `@Assert\NotBlank` correspond au message qui sera affiché en erreur si lors de la validation, le champ est vide.

Et l'entité Tampon comme ceci :




```
class Tampon extends Ouvrage
{
    /**
     * @var string
     * @ORM\Column(name="profondeur", type="string", nullable=true)
     * @Gedmo\Versioned
     * @Assert\NotBlank(message="Le champ ne peut pas être vide")
     */
    private $profondeur;

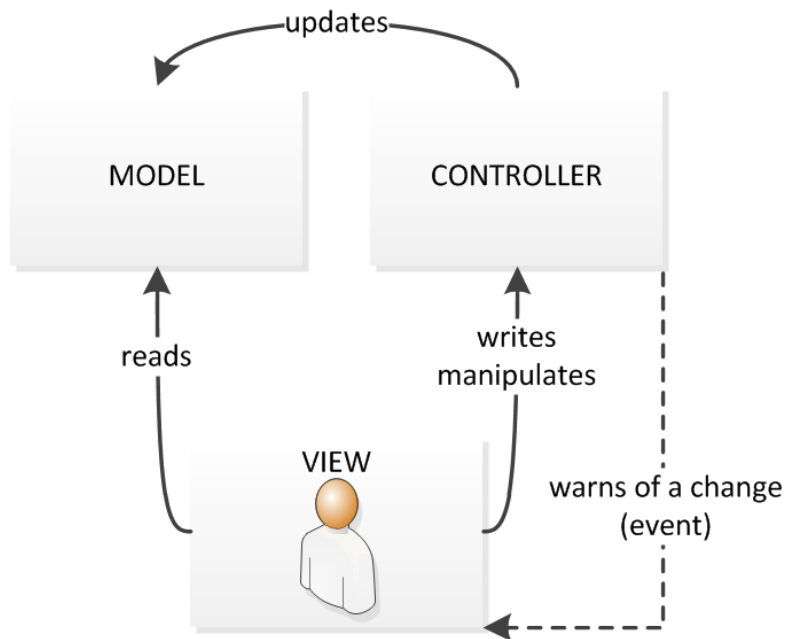
    /**
     * @var string
     * @ORM\Column(name="diametre", type="string", nullable=true)
     * @Gedmo\Versioned
     * @Assert\NotBlank(message="Le champ ne peut pas être vide")
     */
    private $diametre;
}
```

Une fois ce travail de préparation réalisé nous avons commencé le développement du module sous Symfony2.

Symfony2 est un framework MVC. Le patron d'architecture logicielle modèle-vue-contrôleur (en abrégé MVC, en anglais model-view-controller), tout comme les patrons modèle-vue-présentation ou présentation, abstraction, contrôle, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

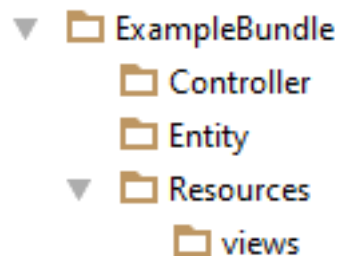
Cette architecture regroupe les fonctions nécessaires en trois catégories :

-  Un modèle (modèle de données)
-  Une vue (présentation, interface utilisateur)
-  Un contrôleur (logique de contrôle, gestion des événements, synchronisation).



Sous Symfony, les trois catégories sont représentées comme ceci :

- 📁 Le modèle est l'entité (par exemple l'entité ouvrage)
- 📁 La vue est le template Twig
- 📁 Le contrôleur est le controller



Cette architecture est reproductible dans n'importe quel bundle\*.

Le dossier *Controller* contient tous les contrôleurs, le dossier *Entity* contient les entités, et le dossier *views* contient les vues de notre bundle c'est-à-dire les templates Twig\*. Dans le dossier *Resources*, on peut y trouver d'autres dossiers, notamment le dossier *public* qui contient les fichiers publics du bundle (fichiers CSS\* et JavaScript\*, images, etc.), ainsi que le dossier *config* qui contient les fichiers de configuration du bundle.

### Symfony – Les bundles

Un bundle est une brique de l'application. Symfony2 utilise ce concept novateur qui consiste à regrouper dans un même endroit, le bundle, tout ce qui concerne une même fonctionnalité.

Nous avons donc créé plusieurs bundles, car nous avons implémenté plusieurs fonctionnalités dans notre application web.

## Symfony – Les entités

Une entité est une classe qui contient des données, qui sera manipulée et enregistrée dans la base de données par l'ORM.

Après avoir créé le bundle *OuvrageBundle* et avoir créé les entités *Ouvrage* et *Tampon* (présenté précédemment), j'ai créé le premier contrôleur qui allait servir de page d'accueil, c'est-à-dire contenant le menu du site et le menu du module.

## Symfony – Les contrôleurs

Un contrôleur est une fonction PHP que l'on crée pour prendre les informations provenant de la requête HTTP et qui construit puis retourne une réponse HTTP (sous forme d'un objet *Symfony2 Response*). La réponse peut être une page HTML, un document XML, un tableau JSON sérialisé, une image, une redirection, une erreur 404 ou tout ce que vous pouvez imaginer.

Par exemple un contrôleur pour avoir la structure suivante :

```
1 // src/AppBundle/Controller/HelloController.php
2 namespace AppBundle\Controller;
3
4 use Symfony\Component\HttpFoundation\Response;
5 use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
6
7 class HelloController
8 {
9     /**
10      * @Route("/hello/{name}", name="hello")
11      */
12     public function indexAction($name)
13     {
14         return new Response('<html><body>Hello '.$name.'!</body></html>');
15     }
16 }
```

Le contrôleur ci-dessus, nommé *HelloController*, possède une route nommée *hello* et une méthode *indexAction*. La ligne `@Route("/hello/{name}", name="hello")` définit la route par laquelle la méthode sera accessible sur l'application web. Ici l'URL sera par exemple <http://www.example.com/hello/LaNetCie>. La méthode retournera alors une *Response* et sera affichée à l'URL précédente. L'on pourra voir sur la page affichée « Hello LaNetCie ».

Un contrôleur peut aussi retourner une vue. Par exemple :

```
// renders app/Resources/views/hello/index.html.twig
return $this->render('hello/index.html.twig', array('name' => $name));
```

Dans l'exemple ci-dessus, la méthode retourne la vue *index.html.twig*, stockée dans le dossier *hello* que contient le dossier *Resources/views*, avec en paramètre '*name*' qui contient *LaNetCie* si on reprend l'exemple ci-dessus.

## Symfony – Les vues

Les vues dans Symfony sont des templates écrites en Twig. Twig est un moteur de templates PHP. Les templates permettent d'afficher du contenu HTML dynamiquement grâce aux paramètres transmis depuis le contrôleur.

Voici un exemple de template Twig :

```
1      {# Resources/views/hello/index.html.twig #}
2      <!DOCTYPE html>
3      <html>
4          <body>
5              <h1>Hello {{ name }} !</h1>
6          </body>
7      </html>
```

Ce template affichera la valeur du paramètre '*name*' s'il est existant, sinon il provoquera une erreur. Si l'on reprend la valeur de l'exemple ci-dessus, il sera affiché sur la page « Hello LaNetCie » sous forme de titre.





## SIAH – Ajout de module sur l'Intranet

Comme Symfony est un framework respectant le pattern MVC et que le modèle correspond à l'entité, j'ai travaillé sur l'entité ouvrage. J'ai donc implémenté des méthodes dans le controller du bundle dédié à la gestion des ouvrages nommé *OuvrageBundle*.

La première méthode que j'ai développée est l'affichage de la liste des tampons présents dans la base de données.

Voir	Code	Denomination	Adresse	Profondeur	Diamètre	Modifier	Supprimer
	Par.1	AVENUE DENIS PAPIN	hj 95400 ARNOUVILLE	3.25	1500		
	Par.2	AVENUE DENIS PAPIN	95400 ARNOUVILLE	3.35	1500		
	Par.3	AVENUE DE CHOISEUL	95400 ARNOUVILLE	0.85/3.85	1500		
	Par.4	AVENUE DE CHOISEUL	95400 ARNOUVILLE	1.70/3.80	1500		
	Par.5	AVENUE DENIS PAPIN	95400 ARNOUVILLE	2.90	1500		
	Par.6	PLACE DU GENERAL LECLERC	95400 ARNOUVILLE	2.40	1500		
	Par.7	PLACE DU GENERAL LECLERC	95400 ARNOUVILLE	1.78	500		
	Par.8	PLACE DU GENERAL LECLERC	95400 ARNOUVILLE	2.10	800		
	Par.9	SUD LYCEE HORTICOLE	95400 ARNOUVILLE	2.35	dalot		
	Par.10	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.11	SUD LYCEE HORTICOLE	95140 GARGES-LES-GONESSE		dalot		
	Par.12	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.13	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.14	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot 2.00x2.15		
	Par.15	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.16	ROND POINT DU CHRIST	95400 ARNOUVILLE		dalot		
	Par.17	AVENUE DE STALINGRAD	95140 GARGES-LES-GONESSE	3.45	dalot		
	Pbf.5165	RUE LOUISE MICHEL	95330 DOMONT	2.05	900		

(Ci-dessus – Liste des tampons, Cf. Annexe 3 – page 53)

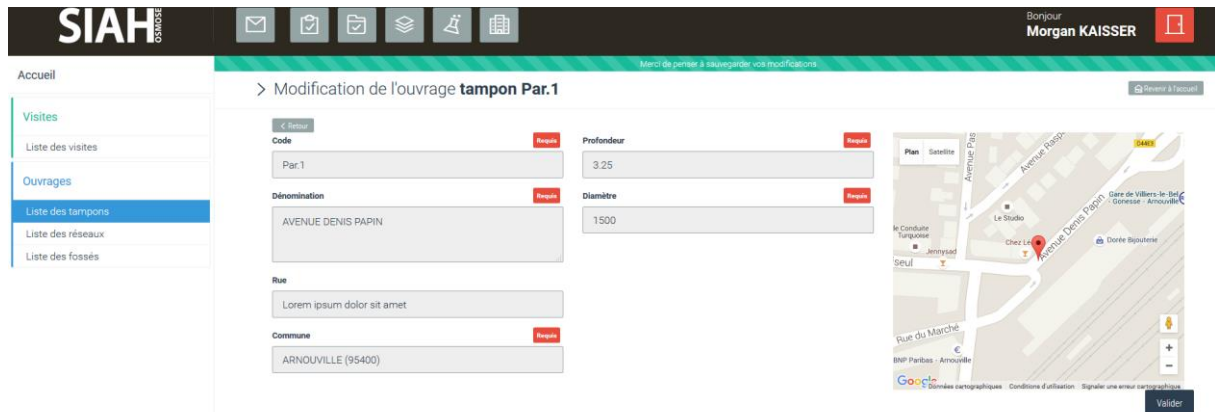
À partir de cette liste, plusieurs actions sont disponibles sur chaque ouvrage :

	Accéder aux détails d'un ouvrage
	Modifier un ouvrage
	Supprimer un ouvrage

Ensuite j'ai développé la méthode permettant la modification d'un ouvrage en base de données. La méthode retourne le template Twig nommé *form.html.twig*. Symfony permet de créer un formulaire à partir des champs d'une entité et de l'implémenter en Twig. C'est donc avec cette méthode que j'ai implémenté le formulaire twig. J'ai aussi dû utiliser l'API Google Map\* sur cette vue ainsi que sur la vue permettant de voir les détails d'un ouvrage, pour pouvoir situer l'ouvrage sur une carte.

L'API Google Map pour le web en JavaScript permet d'ajouter et de personnaliser les cartes avec des contenus et des images, et de les intégrer sur un site web.

Une fois le type des champs paramétré (champ texte, autocomplétion), voici le résultat :

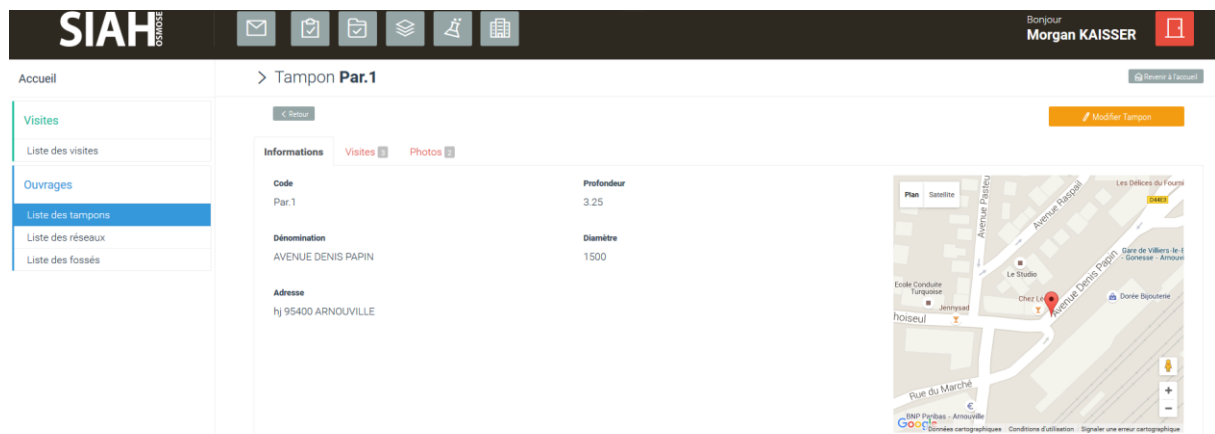


(Ci-dessus – Modification de l'ouvrage tampon Par.1, Cf. Annexe 4 – page 54)

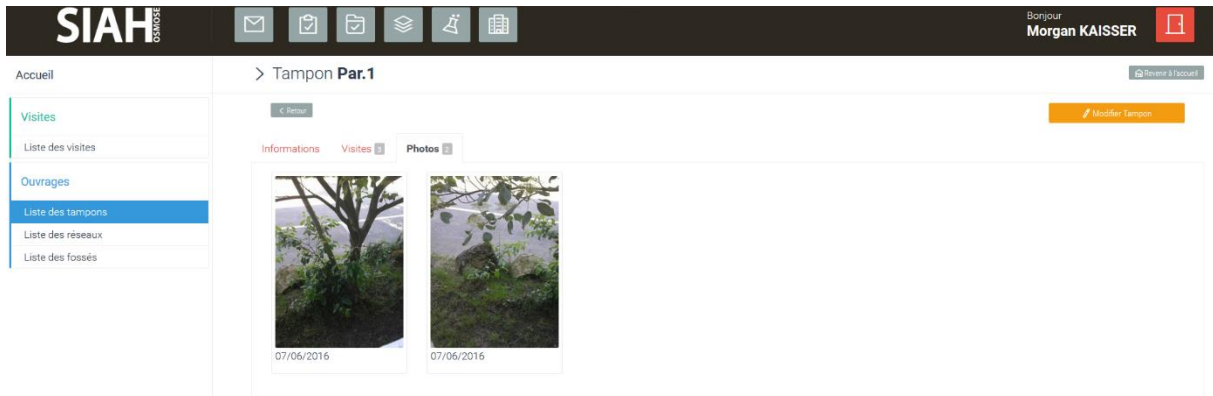
Pour remplir les champs avec les bonnes valeurs, j'ai utilisé *EntityRepository*, c'est-à-dire que j'ai créé un dossier *Repository* dans le bundle *OuvrageBundle* et créé une classe héritant d'*EntityRepository*. Une fois notre classe liée à notre entité, elle permet d'effectuer des requêtes à notre base de données via Doctrine. Par exemple, récupérer les informations d'un ouvrage par son id, est une méthode que j'ai utilisée pour pouvoir remplir les champs dans la vue précédente.

Par la suite j'ai implémenté la méthode permettant de voir les informations d'un ouvrage. Cette méthode prend en paramètre un id d'ouvrage et affiche les informations de celui-ci. Ces informations sont obtenues grâce à une requête vers la base de données via Doctrine.

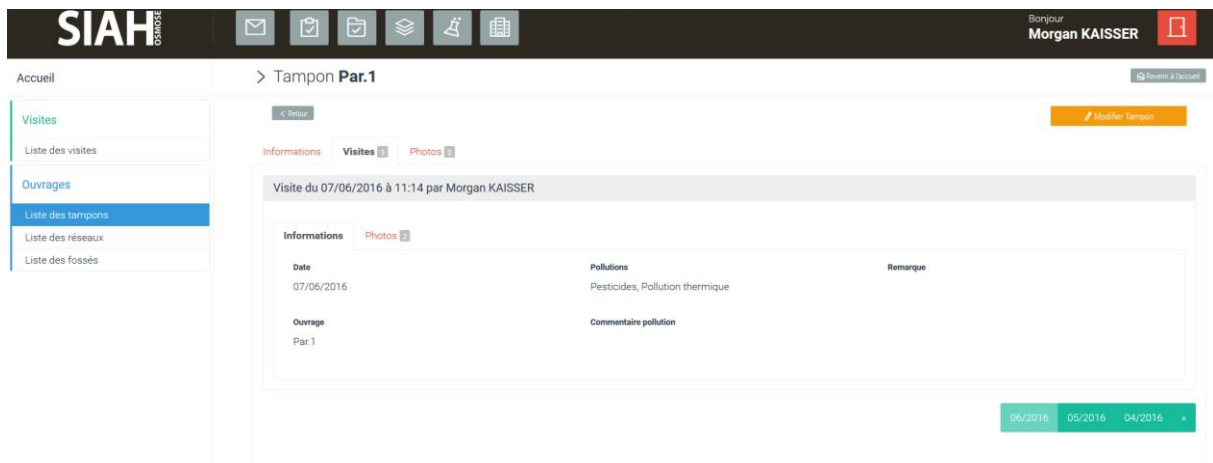
(Ci-dessous – Affichage des informations de l'ouvrage tampon Par.1, Cf. Annexe 5 – page 55)



(Ci-dessous – Affichage des photos des visites de l'ouvrage tampon Par.1, Cf. Annexe 6 – page 56)



(Ci-dessous – Affichage de l'historique des visites de l'ouvrage tampon Par.1, Cf. Annexe 7 – page 57)



Les photos affichées sur la vue précédente sont les photos qui ont été prises lors des visites sur l'ouvrage tampon Par.1. Cette vue affichera toutes les photos de toutes les visites ayant eu lieu sur cet ouvrage.

Sur la vue ci-dessus, les visites sont classées par mois, c'est-à-dire que j'ai créé une méthode pour paginer les résultats de la requête me retournant les visites, en faisant attention de n'afficher seulement les mois pendant lesquels s'étaient déroulées des visites.

Pour afficher le contenu de la visite, j'utilise une vue existante, et utilisée pour afficher les détails des visites depuis la liste des visites développée par M. Alex COUPOUSSAMY. Il est donc possible de retrouver les photos de la visite en cliquant sur l'onglet *Photos*.

J'ai développé la modal\* de suppression d'élément sur le site, modal que l'on retrouvera sur toutes les listes du nouveau module, c'est-à-dire les listes des ouvrages (tampon, fossé, réseau), la liste des visites et la liste des pollutions.

Cette modal s'affichera lors du clic de l'utilisateur sur le bouton rouge de suppression.

Voici un exemple de modal lors du clic sur le bouton de suppression qui concerne le tampon Par.1.

### Confirmer la suppression

✕

Êtes-vous sûr(e) de vouloir supprimer **l'ouvrage tampon Par.1** ?  
**Attention, la suppression de l'ouvrage tampon Par.1 entrainera la suppression des visites associées.**

Annuler
Confirmer

La suppression d'un ouvrage ne supprime pas l'ouvrage de la base de données, mais des yeux de l'utilisateur. Lors de cette suppression, la date de l'action par l'utilisateur est enregistrée dans la base de données. Ce qui permet aux administrateurs de restaurer l'ouvrage en cas de fausse manipulation. Si l'on supprime un ouvrage, toutes les visites qui concernent cet ouvrage seront aussi supprimées, et auront la même date de suppression. Par contre si l'on supprime une visite, seule la visite sélectionnée est supprimée.

Une fois ce travail réalisé, j'ai travaillé sur le filtre présent sur la liste des visites, liste réalisée par M. Alex COUPPOUSSAMY. Ce filtre permet de trier les visites affichées par le code de l'ouvrage concerné par cette visite, le type de l'ouvrage, l'agent qui a créé la visite, par date ou par période (par exemple par mois), par pollution(s), par ville et par remarque.

Filtrer les visites

Code ouvrage <input style="width: 90%;" type="text"/>	Type <input style="width: 90%;" type="text" value="Aucune sélection"/>	Agent <input style="width: 90%;" type="text" value="Aucune sélection"/>	Remarque <input style="width: 90%;" type="text"/>
Date du <input style="width: 90%;" type="text" value=""/>	Pollutions <input style="width: 90%;" type="text"/>	Ville <input style="width: 90%;" type="text"/>	
au <input style="width: 90%;" type="text" value=""/>			

Filter

(Ci-dessus – Filtre sur la liste des visites, Cf. Annexe 8 – page 58)

## RAPPORT DE STAGE

Les champs type et Agent peuvent être remplis en cliquant sur un élément dans les listes respectives :

### Type

Aucune sélection ▼

- Tampon
- Réseau
- Fossé

### Agent

Aucune sélection ▼

- Morgan KAISSER
- Grigore SCOTU
- Lynette BUCHONVILLE

La date ou la période sont saisies sur un calendrier en cliquant sur l'année, le mois et le jour :

### Date du

|

juin 2016

L	M	M	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Le champ ville est un champ d'autocomplétion, l'utilisateur peut saisir une ville parmi une liste de propositions dès la saisie d'une lettre ou plus :

### Ville

A|

- ARNOUVILLE
- BONNEUIL-EN-FRANCE
- BOUQUEVAL
- EPIAIS-LES-LOUVRES
- EZANVILLE

Une ou plusieurs pollutions parmi une liste de pollutions existante :



Une fois les champs remplis, il peut filtrer les visites. Pour cela j'ai développé une méthode qui différencie deux méthodes d'affichage, avec et sans paramètre. Sans paramètre, toutes les visites sont affichées, avec paramètres, cela signifie que l'utilisateur filtre les visites sur au moins une caractéristique. Pour cela, j'ai créé une méthode dans la classe *OuvrageRepository*, pour pouvoir effectuer des requêtes sur la base de données via Doctrine, permettant de ne renvoyer que les visites correspondantes aux filtres. Cette méthode utilise les jointures avec la table ouvrage pour pouvoir effectuer un tri sur les ouvrages liés aux visites et avec la table commune pour trier par nom de commune.

Pour toutes les vues que j'ai présentées précédemment, nous avons réutilisé le design de l'Intranet qui utilise le framework CSS et jQuery Bootstrap\*.

Pour conclure sur ce nouveau module, nous avons réalisé ce qui était demandé dans le cadre de ce PoC. Cette mission m'a permis de développer des compétences dans le domaine du développement web, notamment en PHP, jQuery, MySQL, PostgreSQL, et m'a permis de découvrir de nouvelles technologies telles que le framework PHP Symfony2 et le moteur de template Twig.



## SIAH – Application mobile

Toujours pour le SIAH, nous avons développé une application en C#\* en lien avec le nouveau module. Cette application doit permettre aux agents du SIAH, lors d'une visite sur un ouvrage, de pouvoir enregistrer les informations concernant sa visite dans la base de données du SIAH.

Dans un premier temps, j'ai développé une activité\* qui renvoie la position actuelle sur une carte avec les ouvrages situés autour de cette position.

Une activité correspond à une partie de l'application présentant une vue à l'utilisateur.

Pour développer cette activité, j'ai dû me former à l'utilisation d'OpenLayers 3\*. OpenLayers est un logiciel libre, publié sous licence BSD. Il constitue une bibliothèque de fonctions JavaScript assurant un noyau de fonctionnalités orienté vers la mise en place d'applications clientes web cartographiques fluides. OpenLayers permet d'afficher des fonds cartographiques tuilés ainsi que des marqueurs provenant d'une grande variété de sources de données.

Je me suis documenté sur cet outil et sa prise en main a nécessité une semaine environ. Une fois l'implémentation terminée, j'ai dû récupérer les coordonnées de l'appareil puis les transmettre à la carte pour qu'il affiche notre position sur la carte. Ensuite il a fallu récupérer les coordonnées des ouvrages dans la base de données locale et calculer la distance entre la position actuelle et chaque ouvrage pour déterminer si on devait les afficher sur la carte. Pour cela j'ai utilisé SpatialLite.

SpatialLite\* est une extension spatiale de SQLite disposant des fonctionnalités de base de données géographiques gérant les données de type vectoriel. Elle m'a donc permis de calculer la distance lors de la requête à la base de données et de ne pas la calculer après. La requête retourne seulement les ouvrages étant situés à une distance inférieure ou égale à la distance que l'on aura définie auparavant.

Après avoir eu des problèmes provenant des bibliothèques de SpatialLite, je me suis tourné vers une autre méthode : grâce à la position, il est possible de récupérer la ville où se situe l'appareil. J'ai donc créé une requête qui ne renvoie que les ouvrages étant situés dans la ville, ce qui réduit grandement le nombre d'ouvrages à traiter. Après avoir récupéré les ouvrages, je calcule la distance entre la position actuelle et la position de l'ouvrage. L'ouvrage ne sera affiché sur la carte que si la distance entre lui et la position actuelle de l'appareil est inférieure à la distance que l'on aura fixée auparavant.

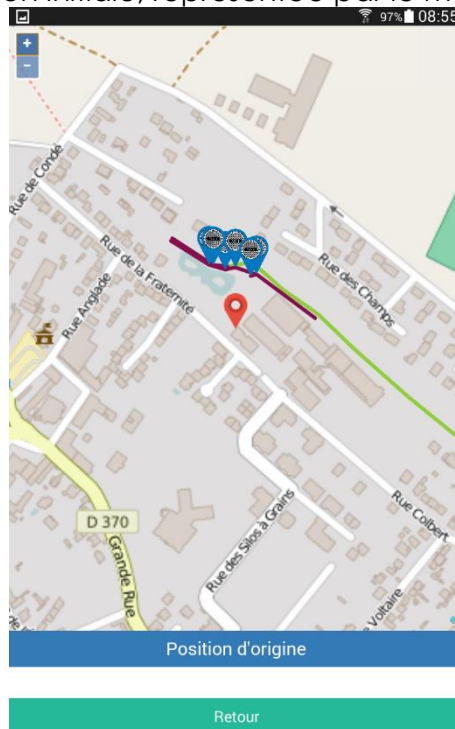
Une fois les problèmes provenant de l'implémentation de SpatialLite résolus, j'ai donc implémenté une nouvelle fois la fonction développée au départ : récupérer les coordonnées des ouvrages dans la base de données locale et calculer la distance entre la position actuelle et chaque ouvrage pour déterminer si on doit les afficher sur la carte.

## RAPPORT DE STAGE

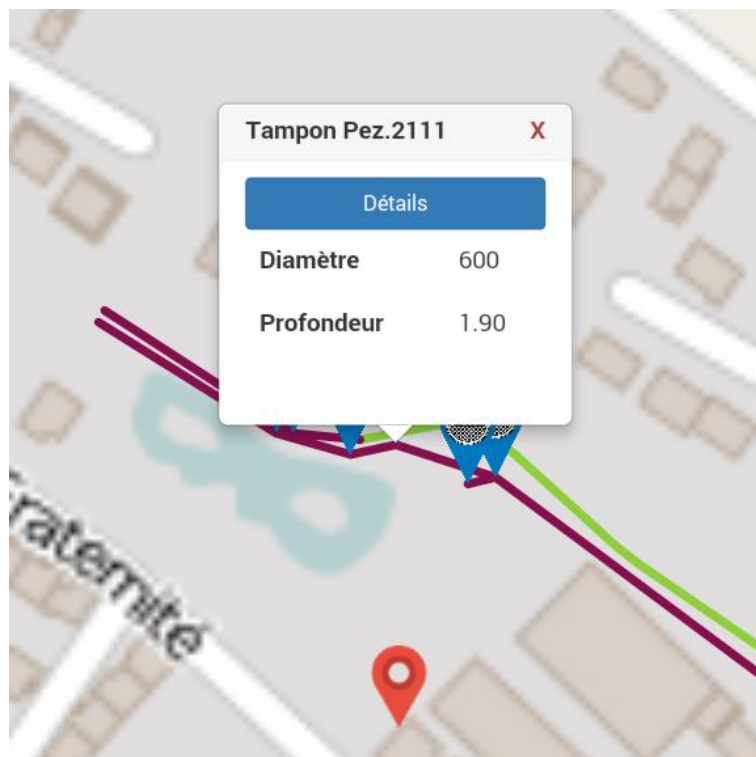
La distance doit être de moins de 100 mètres pour qu'un ouvrage soit affiché sur la carte. Cette activité est accessible par le menu principal et permet après clic sur un ouvrage de la carte d'obtenir ses informations par le biais d'une nouvelle activité.

J'ai aussi implémenté un écouteur sur la carte, lorsque l'utilisateur se déplace sur cette carte, on charge les ouvrages au fur et à mesure. C'est-à-dire que l'on charge les ouvrages qui sont à moi de 100 mètres du centre de la carte à l'écran.

J'ai implémenté également un bouton de retour à la position d'origine, qui permet donc de revenir à la position initiale, représentée par le marqueur rouge sur la carte.



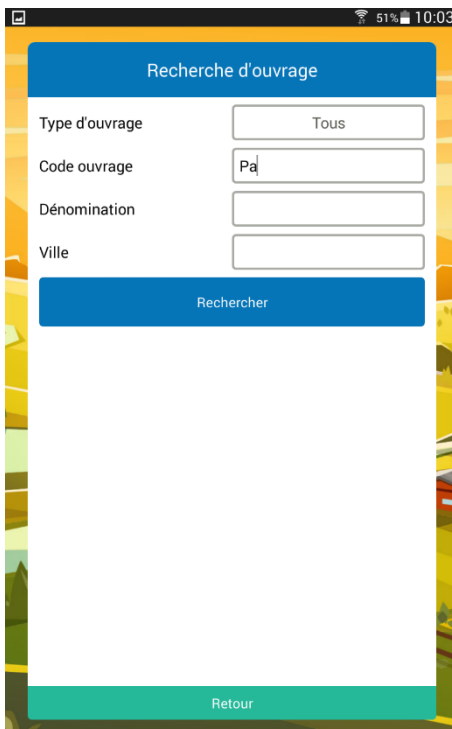
(Ci-contre – Activité présentant la carte, Cf. Annexe 9 – page 59)



(Ci-dessus exemple de popup s'affichant au clic sur un ouvrage)

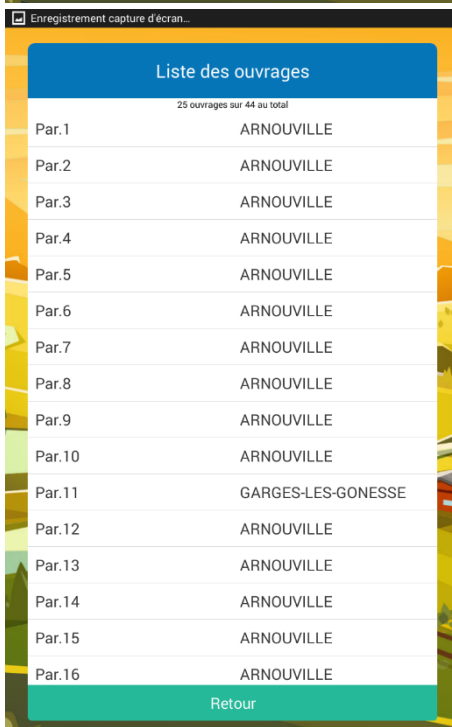
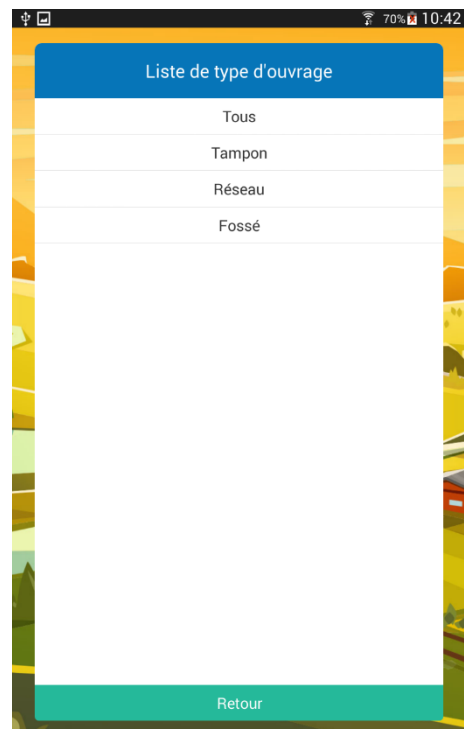


J'ai ensuite développé une activité permettant de faire une recherche multicritère sur les ouvrages existants en base de données. Recherche pouvant s'effectuer par la sélection du type de l'ouvrage, la saisie de son code et/ou de la ville où il est situé. Les champs de saisi du code, de la dénomination et de la ville possèdent l'autocomplétion, ce qui permet à l'utilisateur d'avoir une liste de propositions dès la saisie de plus de deux caractères. L'utilisateur peut sélectionner le type de l'ouvrage parmi une liste de types existants ou effectuer sa recherche sans tenir compte du type. Une fois les champs remplis, l'utilisateur obtient une liste d'ouvrages correspondant aux critères qu'il a saisis précédemment. Cette liste est le résultat d'une requête effectuée sur la base de données locale. Puis, en cliquant sur un ouvrage de cette liste, il obtiendra ses informations par le biais d'une nouvelle activité.



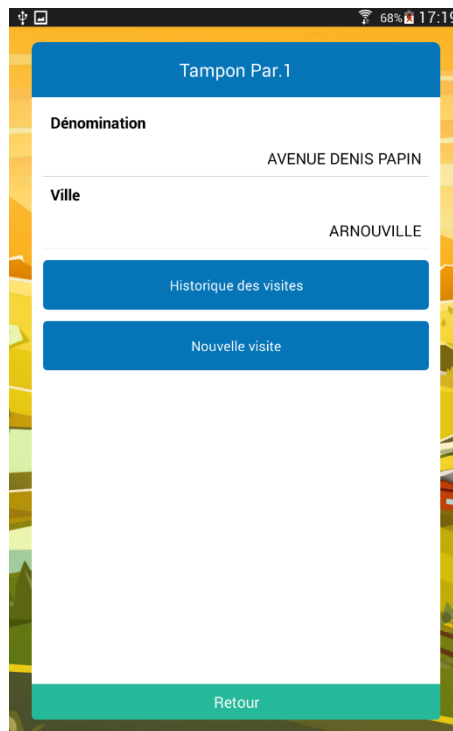
(Ci-contre – Activité présentant le formulaire de recherche d'ouvrage, Cf. Annexe 10 – page 60)

(Ci-dessous – Activité présentant les types d'ouvrage, Cf. Annexe 11 – page 61)



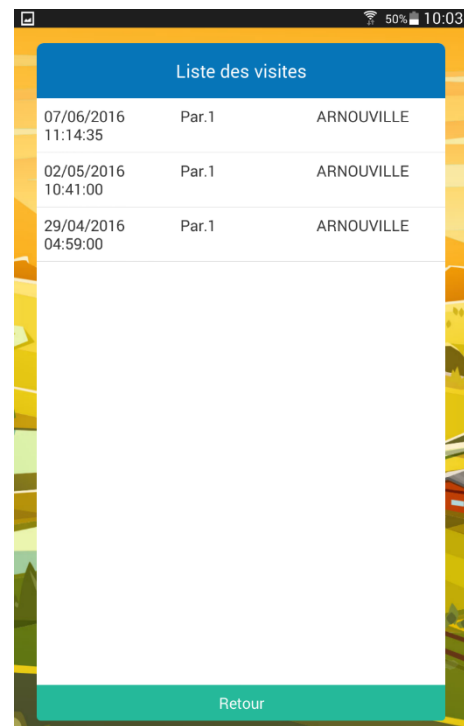
(Ci-contre – Activité présentant les résultats de la recherche sur le code "Pa", Cf. Annexe 12 – page 62)

J'ai par la suite développé une activité permettant d'obtenir les informations d'un ouvrage et de créer une visite concernant cet ouvrage. Sur cette activité l'on peut aussi accéder à l'historique des visites qui ont été effectuées sur cet ouvrage. Cette activité peut être appelée depuis l'une des deux activités précédentes.



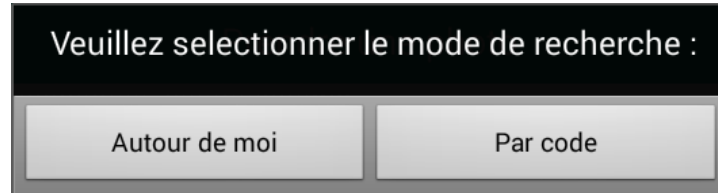
*(Ci-contre – Activité présentant les informations d'un ouvrage, Cf. Annexe 13 – page 63)*

J'ai développé une activité permettant de connaître l'historique des visites ayant été effectuées sur un ouvrage. Cette activité est disponible en cliquant sur le bouton historique présent sur l'activité affichant tous les détails d'un ouvrage. Les visites sont récupérées directement en allant requêter une URL sur l'Intranet qui renvoie la liste des visites au format JSON\*. Après avoir traité les données, on affiche donc l'historique des visites sous forme de liste. Lors du clic sur une visite, une nouvelle activité, que j'ai aussi développée, se lance et l'on obtient les détails de cette visite.

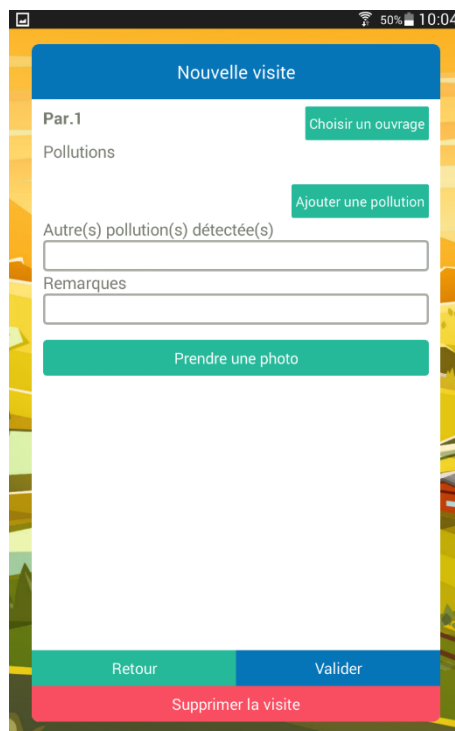


*(Ci-dessus – Activité présentant l'historique des visites d'un ouvrage, Cf. Annexe 14 – page 64)*

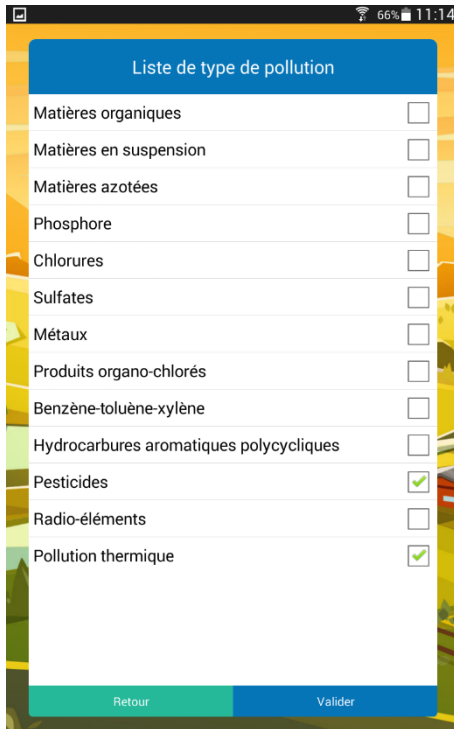
J'ai participé au développement de l'activité permettant la création/modification de visite. Sur cette activité, l'utilisateur a la possibilité de saisir un ouvrage, si cela n'est pas déjà fait, ou de le modifier. La modification de cet ouvrage peut s'effectuer de deux manières : en allant sur la carte ou en effectuant une recherche et en cliquant sur un ouvrage.



Il peut également saisir une ou plusieurs remarques, saisir une ou plusieurs pollutions parmi une liste de pollutions existantes, prendre une ou plusieurs photos et ajouter un commentaire à ces photos. Une fois terminé, il peut soit valider la visite soit la supprimer. S'il valide la visite et qu'il est connecté à Internet, il a le choix d'envoyer directement la visite dans la base de données du serveur ou de la stocker en base de données locale. S'il n'est pas connecté à Internet, elle sera automatiquement sauvegardée dans la base de données locale.

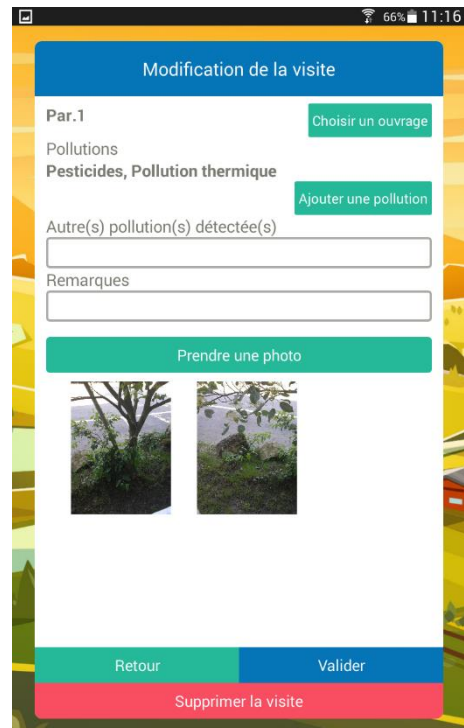


(Ci-dessus – Activité création/modification de visite,  
Cf. Annexe 15 – page 65)



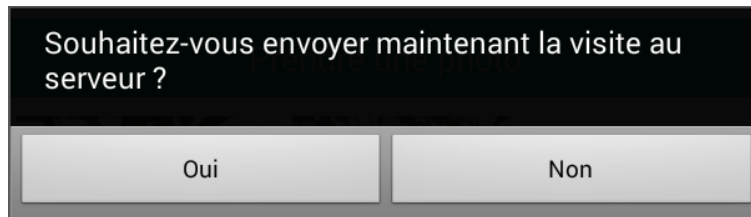
(Ci-contre – Activité d'ajout de pollution(s), Cf. Annexe 16 – page 66)

(Ci-dessous – Activité nouvelle visite avec des photos, Cf. Annexe 17 – page 67)



(Ci-contre – Activité d'édition du commentaire d'une photo, Cf. Annexe 18 – page 68)

Message de confirmation qui apparaîtra si l'utilisateur est connecté à Internet lors de la validation de la visite.

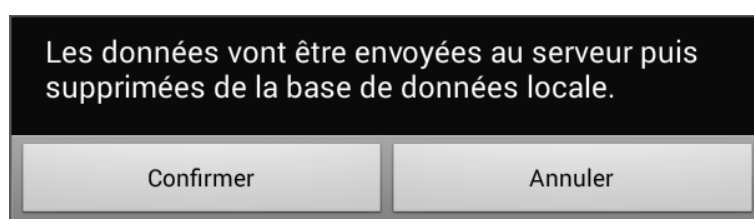


Les visites étant stockées en base de données locale, il faut les envoyer au serveur. J'ai donc développé une activité permettant de sélectionner une ou plusieurs visites parmi une liste de visites valides, c'est-à-dire étant liées à un ouvrage. Cette activité n'est accessible que si l'utilisateur est connecté à Internet et s'il y a des visites en base de données. Une fois que les visites sont sélectionnées et que l'utilisateur confirme l'envoi des données, les informations en base de données concernant ces visites sont converties au format JSON.



*(Ci-contre – Activité de sélection des visites à envoyer au serveur, Cf. Annexe 19 – page 69)*

Une fois que l'utilisateur clique sur le bouton *Valider*, le message suivant apparaît à l'écran.



JSON (JavaScript Object Notation) est un format de données textuelles, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter de l'information structurée.

Si des photos ont été prises lors des visites sélectionnées, elles sont stockées dans une archive au format ZIP\*.

ZIP est un format de fichier permettant l'archivage (utilisation d'un seul fichier pour stocker plusieurs fichiers) et la compression de données (diminution de l'espace occupé sur le support numérique) sans perte de qualité.

Puis on ajoute le fichier JSON contenant les informations concernant les visites et l'archive contenant les photos dans une archive que l'on envoie sur le serveur. Une fois envoyées sur le serveur et enregistrées en base de données, les visites sélectionnées sont supprimées de la base de données locale et les photos correspondantes sont supprimées du stockage de l'appareil.

J'ai aussi développé un script côté serveur permettant de traiter les données présentes dans l'archive ZIP envoyée par l'application. Le script commence par décompresser l'archive et boucle sur les visites présentes dans le fichier JSON pour les ajouter en base de données sur le serveur. Puis il décompresse l'archive contenant les photos, les copie dans les dossiers sur le serveur et ajoute à chaque visite les références des photos correspondantes.

J'ai participé à la mise en place du responsive design de l'application ainsi qu'aux différents tests de débogage.




Pour conclure sur cette application mobile, nous avons réalisé ce qui était demandé dans le cadre de ce PoC. Cette mission m'a permis de découvrir des technologies comme Xamarin et le C# que je n'ai pas eu l'occasion d'aborder pendant mon DUT. Elle m'a permis aussi d'utiliser SQLite, SpatiaLite ainsi qu'OpenLayers. Ces compétences et connaissances que j'ai acquises lors de cette mission sont un avantage dans le domaine du web car le développement mobile et web sont souvent liés.

## BILANS

---

### Bilan professionnel

Ce stage m'a permis d'atteindre les objectifs que je m'étais fixés au départ :

-  Approfondir mes connaissances
-  Développer de nouvelles compétences
-  Appréhender le travail en entreprise

Pendant ce stage, je me suis aperçu que les connaissances et les compétences que j'avais acquises à l'IUT m'avaient permis de comprendre les différentes missions qui m'ont été confiées et de comprendre les besoins du client. Les projets à l'IUT m'ont aussi aidé à travailler en équipe avec des développeurs qui n'ont pas forcément la même vision du projet et des besoins du client.

Lors de ces différentes missions, j'ai su me documenter sur Internet, mais aussi demander conseil au développeur senior, la communication dans l'équipe est importante au bon déroulement du projet.

Ma première mission sur l'Intranet, concernant le développement du nouveau module m'a permis de développer des compétences dans le domaine du développement web et m'a permis de découvrir de nouvelles technologies telles que le framework PHP Symfony2 et le moteur de template Twig. Compétences qui me permettront de continuer de travailler et d'approfondir mes connaissances dans ce domaine.

Ma deuxième mission, le développement de l'application mobile, m'a permis de découvrir des technologies comme Xamarin et le C# que je n'ai pas eu l'occasion d'aborder pendant mon DUT. En effet avec la présence des deux parcours de DUT Informatique possibles, j'ai choisi de me diriger vers une formation d'ingénieur, au regret de ne pas aborder le développement Android en Java avec Android Studio. Ce stage m'a donc permis de découvrir le développement mobile, intimement lié au développement web, un atout pour travailler dans ce domaine.

Ce stage m'a permis aussi de mieux appréhender les phases d'un projet et d'un PoC. Ce stage fait découvrir une autre vision du développement, de ne plus penser que les utilisateurs vont s'adapter au produit, mais que le produit doit s'adapter à l'utilisateur. Ce que j'ai développé, en plus d'être fonctionnel et optimisé, doit être maintenable et évolutif, c'est-à-dire qu'il faut penser son code de manière à ce qu'il soit lisible par n'importe quel autre développeur.

Ce stage a conforté mon envie de continuer et de découvrir de nouvelles technologies qui font la richesse du domaine du développement web.





### Bilan personnel

Les différences missions qui m'ont été confiées m'ont permis de devenir de plus en plus autonome, mais aussi de savoir poser des questions quand je suis dans une impasse. Pour pouvoir poser les questions, il faut pouvoir prendre du recul sur ce que l'on est en train de faire pour pouvoir cibler les recherches et poser les bonnes questions.

La communication est primordiale pour le bon déroulement d'un projet, et cela commence par une bonne intégration au sein d'une entreprise. La taille de l'entreprise étant favorable, connaître tout le monde est plus facile et plus agréable au bon fonctionnement de l'entreprise et au bon déroulement du stage. En plus de cela, nous étions deux stagiaires, de la même promotion, nous nous connaissions depuis la première année de notre DUT Informatique et avons déjà travaillé ensemble sur des projets, ce qui a facilité la communication entre nous et l'entraide.

Je me suis rendu compte que travailler dans une entreprise à taille humaine est fait pour moi, mais voulant faire de l'alternance en parcours ingénieurs il faut que la structure de l'entreprise et son budget le permettent, et le nombre de petites structures le permettant est restreint.

Cette expérience professionnelle m'a conforté dans mon choix de me spécialiser dans le développement web bien que le développement logiciel me plaise aussi. Ma formation de DUT Informatique m'ayant formé dans les deux domaines, si un jour je veux me réorienter, j'aurais toujours la possibilité de le faire.



## CONCLUSION

---

Pour conclure, ce stage a été une expérience professionnelle significative, car elle m'a permis de choisir la voie dans laquelle je voulais me spécialiser et m'a conforté dans mon choix de vouloir faire de l'alternance pendant ma poursuite d'études. À travers les missions qui m'ont été confiées, j'ai pu développer mon autonomie, mes connaissances, de nouvelles compétences et découvrir de nouvelles technologies réputées dans le domaine du web. Grâce à ces nouvelles compétences et cette expérience professionnelle en développement web et mobile, l'envie de découvrir davantage ce domaine et les multiples technologies a été confortée.

La formation de DUT Informatique, qui a pour but initial de former des techniciens opérationnels, a su m'apporter une base théorique et technique, m'a permis de réaliser ce stage, et me permet aujourd'hui de candidater en cycle d'ingénieur. Une suite que je trouve logique et qui va me permettre, tout au long de ma future carrière de pouvoir évoluer.



## LEXIQUE

<b>Activité</b>	Une activité correspond à une partie de l'application présentant une vue à l'utilisateur.
<b>Android</b>	Android est un système d'exploitation mobile, basé sur le noyau Linux et développé actuellement par Google.
<b>API Google Map</b>	L'API Google Map pour le web en JavaScript permet d'ajouter et de personnaliser les cartes avec des contenus et des images, et de les intégrer sur un site web.
<b>ASP.NET</b>	ASP.NET est une technique pour générer à la demande des pages web, lancée par Microsoft en juillet 2000, et utilisée pour mettre en œuvre des applications web.
<b>Bootstrap</b>	Bootstrap est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.
<b>Bundle</b>	Un bundle est une brique de l'application. Symfony2 utilise ce concept novateur qui consiste à regrouper dans un même endroit, le bundle, tout ce qui concerne une même fonctionnalité.
<b>C#</b>	Le C# est un langage de programmation orienté objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plate-forme Microsoft .NET.
<b>Cross-platform</b>	Un logiciel multi-plate-forme ou multiplate-forme est un logiciel conçu pour fonctionner sur plusieurs plates-formes.
<b>CSS</b>	Le CSS est un langage informatique utilisé sur l'Internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.
<b>Design Pattern</b>	Un patron de conception (plus souvent appelé design pattern) est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels.
<b>Doctrine</b>	Doctrine est un ORM (couche d'abstraction à la base de données) pour PHP. Doctrine est l'ORM par défaut du framework Symfony (depuis la version 1.3 de ce framework).
<b>Framework</b>	Un framework est un ensemble de classes généralement regroupées sous la forme de bibliothèques pour proposer des services plus ou moins sophistiqués. Un cadre de conception reposant sur les design patterns pour proposer tout ou partie d'un squelette d'application.
<b>Git</b>	Git est un logiciel de gestion de versions décentralisé.
<b>GNU GPL</b>	Le terme GNU/GPL désigne le type de licence d'utilisation le plus en vigueur dans le monde du logiciel libre (open source).

<b>Hardware</b>	Le matériel informatique est l'ensemble des pièces détachées des appareils informatiques.
<b>Infogérance</b>	L'infogérance consiste à externaliser la gestion du secteur informatique d'une entreprise qui sera alors confiée à des prestataires extérieurs spécialisés dans ce domaine. Elle comprend la gestion du parc informatique, l'hébergement de sites Internet et messagerie d'entreprise, la sauvegarde de données et la maintenance permanente des serveurs.
<b>JavaScript</b>	Le JavaScript est un langage informatique utilisé sur les pages web. Ce langage a la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activés côté serveur.
<b>JSON</b>	JSON (JavaScript Object Notation) est un format de données textuelles, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter de l'information structurée.
<b>Modal</b>	Un modal est une classe CSS du Framework Bootstrap. C'est un popup personnalisable.
<b>MVC</b>	<p>Le patron d'architecture logicielle modèle-vue-contrôleur (en abrégé MVC) est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.</p> <p>Ce patron regroupe les fonctions nécessaires en trois catégories : un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation).</p>
<b>MySQL</b>	MySQL est une base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).
<b>OpenLayers 3</b>	OpenLayers est un logiciel libre, publié sous licence BSD. Il constitue une bibliothèque de fonctions JavaScript assurant un noyau de fonctionnalités orienté vers la mise en place d'applications clientes web cartographiques fluides. OpenLayers permet d'afficher des fonds cartographiques tuilés ainsi que des marqueurs provenant d'une grande variété de sources de données.
<b>Open-space</b>	Les open-spaces sont des bureaux sans cloison. À l'inverse des bureaux individuels, tous les individus sont dans une même pièce.
<b>ORM</b>	Un mapping objet-relationnel (en anglais object-relational mapping ou ORM) est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. On pourrait le désigner par « correspondance entre monde objet et monde relationnel ».

<b>Ouvrage</b>	Ici un ouvrage peut être une bouche d'égout, un fossé, un cours d'eau ou encore une canalisation d'eau.
<b>PHP</b>	Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.
<b>PoC</b>	Une preuve de concept ou POC (de l'anglais : proof of concept), ou encore démonstration de faisabilité, est une réalisation courte ou incomplète d'une certaine méthode ou idée pour démontrer sa faisabilité. La preuve de concept est habituellement considérée comme une étape importante sur la voie d'un prototype pleinement fonctionnel.
<b>PostgreSQL</b>	PostgreSQL est un système de gestion de base de données relationnelle et objet.
<b>QGIS</b>	QGIS est un logiciel SIG (système d'information géographique) libre multiplate-forme publié sous licence GPL. Le développement a débuté en mai 2002 et est sorti en tant que projet sur SourceForge en juin 2002. Il était également appelé Quantum GIS jusqu'à la version 1.9. La version 2.14 (Essen) est sortie fin février 2016.
<b>SGBD</b>	Un système de gestion de base de données (SGBD) est un logiciel système conçu pour créer et gérer des bases de données.
<b>SIAH</b>	Syndicat Intercommunal pour l'Aménagement Hydraulique des vallées du Croult et du Petit Rosne.
<b>Spatialite</b>	Spatialite est une extension spatiale de SQLite disposant des fonctionnalités de base de données géographiques gérant les données de type vectoriel.
<b>SQLite</b>	SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL.
<b>SSH</b>	Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés.
<b>Symfony2</b>	Symfony est un framework MVC (Modèle, Vue, Contrôleur) libre écrit en PHP. Comme tout framework, il facilite et accélère le développement de sites et d'applications Internet et Intranet.
<b>Téléassistance</b>	La téléassistance, téléassistance, assistance à distance ou encore assistance en ligne, comme son nom l'indique, est l'action ou le service qui permet d'aider à distance les utilisateurs, d'un système ou d'un produit, à l'aide d'un moyen de télécommunication.
<b>Téléphonie VOIP</b>	Voice over Internet Protocol. VoIP : Voice over Internet Protocol, ou Voice over IP, ou Voix Par IP, vous permet par exemple d'appeler d'un PC vers une ligne de téléphone fixe ou vers un PC gratuitement ou moins cher vers le monde entier.

<b>Template</b>	Un template (parfois appelé layout) est une page web sans contenu dont les éléments statiques sont déjà positionnés et mis en forme.
<b>Twig</b>	Twig est un moteur de templates PHP.
<b>URL</b>	Le sigle URL (de l'anglais Uniform Resource Locator, littéralement « localisateur uniforme de ressource »), auquel se substitue informellement l'expression adresse web, désigne une chaîne de caractères utilisée pour adresser les ressources du web.
<b>XML</b>	XML ou eXtensible Markup Language est un langage informatique de balisage générique.
<b>ZIP</b>	ZIP est un format de fichier permettant l'archivage (utilisation d'un seul fichier pour stocker plusieurs fichiers) et la compression de données (diminution de l'espace occupé sur le support numérique) sans perte de qualité.



## RÉFÉRENCES

---

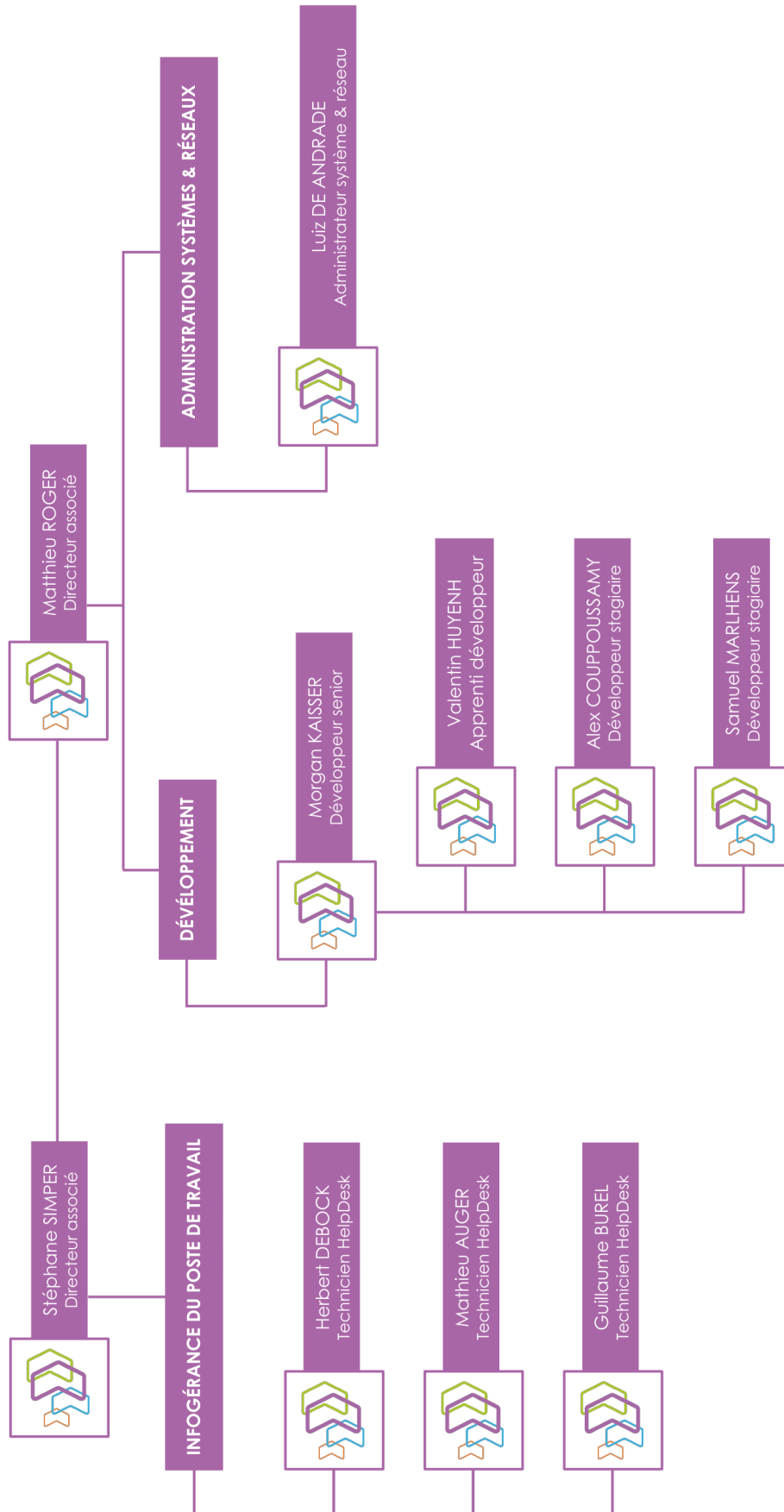
- ❖ La documentation de Xamarin : <https://developer.xamarin.com/>
- ❖ Tutoriels Symfony2 & autres : <https://openclassrooms.com/>
- ❖ StackOverflow pour l'aide en C# & Symfony2 : <http://stackoverflow.com/>
- ❖ Google pour les recherches sur Internet : <http://www.google.fr>
- ❖ La documentation de Twig : <http://twig.sensiolabs.org/>
- ❖ Pour les définitions : <https://fr.wikipedia.org/>
- ❖ Les bundles Symfony : <http://knpbundles.com/>
- ❖ La documentation de bundles : <https://github.com/>
- ❖ La documentation d'OpenLayers 3 : <http://openlayers.org/>
- ❖ La documentation de Google pour l'API Google Maps :  
<https://developers.google.com/>
- ❖ La documentation de Git : <https://git-scm.com/>
- ❖ La documentation de SpatiaLite : <http://www.gaia-gis.it/gaia-sins/spatialite-sql-4.2.0.html>
- ❖ La documentation de SQLite : <https://www.sqlite.org/docs.html>



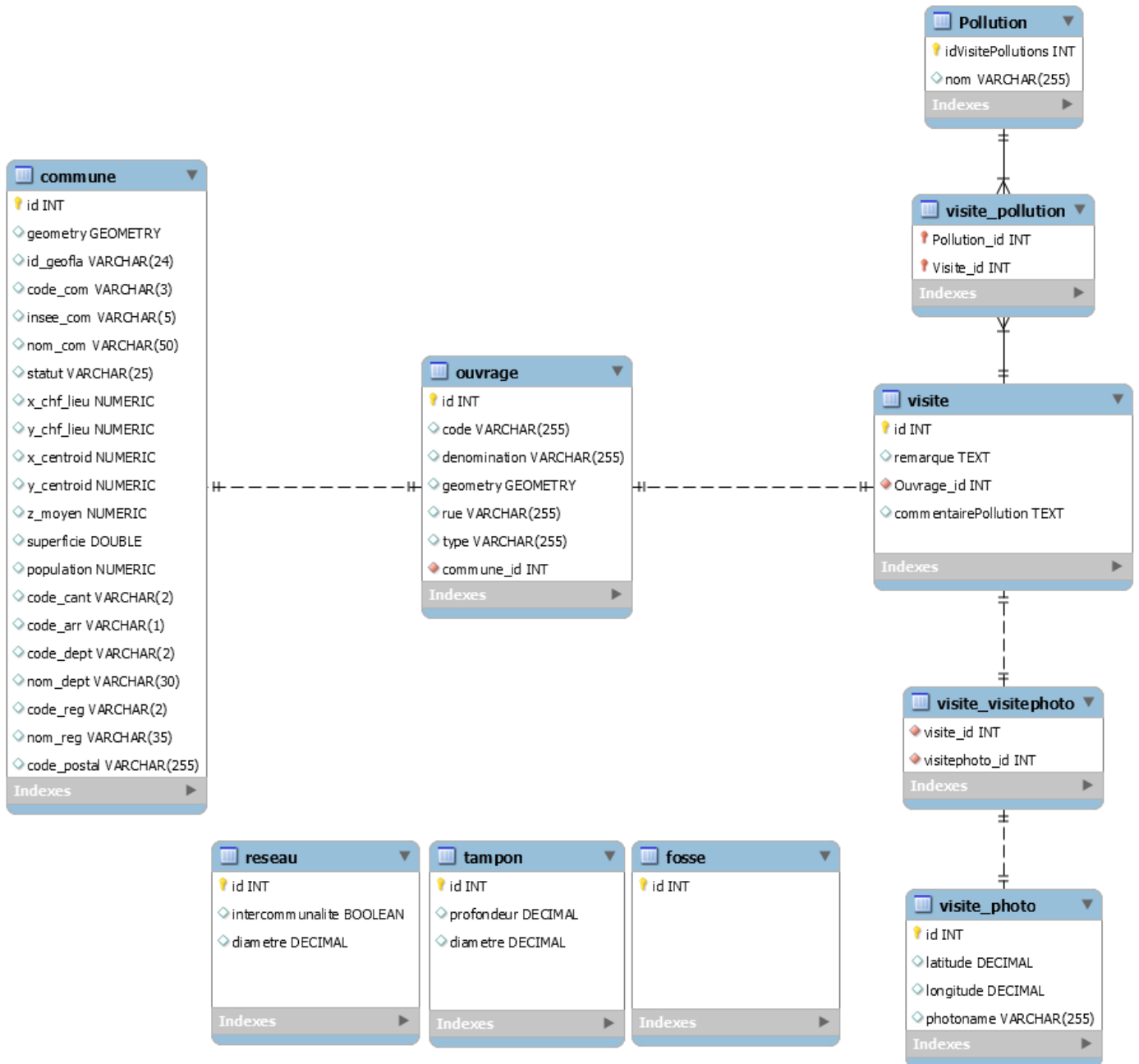
## ANNEXES

Annexe 1 - Organigramme .....	51
Annexe 2 - Architecture de la base de données .....	52
Annexe 3 - Liste des tampons .....	53
Annexe 4 - Modification de l'ouvrage tampon Par.1 .....	54
Annexe 5 - Affichage des informations de l'ouvrage tampon Par.1 .....	55
Annexe 6 - Affichage des photos des visites de l'ouvrage tampon Par.1 .....	56
Annexe 7 - Affichage de l'historique des visites de l'ouvrage tampon Par.1 .....	57
Annexe 8 - Filtre sur la liste des visites .....	58
Annexe 9 - Activité présentant la carte .....	59
Annexe 10 - Activité présentant le formulaire de recherche d'ouvrage .....	60
Annexe 11 - Activité présentant les types d'ouvrage.....	61
Annexe 12 - Activité présentant les résultats de la recherche sur le code "Pa" .....	62
Annexe 13 - Activité présentant les informations d'un ouvrage .....	63
Annexe 14 - Activité présentant l'historique des visites d'un ouvrage.....	64
Annexe 15 - Activité création/modification de visite .....	65
Annexe 16 - Activité d'ajout de pollution(s) .....	66
Annexe 17 - Activité nouvelle visite avec des photos.....	67
Annexe 18 - Activité d'édition du commentaire d'une photo .....	68
Annexe 19 - Activité de sélection des visites à envoyer au serveur.....	69

# Annexe 1 - Organigramme



## Annexe 2 – Architecture de la base de données



## Annexe 3 - Liste des tampons

**SIAH**  
ESOMSO

Bonjour  
**Morgan KAISSER**

Accueil

- Visites
- Liste des visites
- Ouvrages
- Liste des tampons
- Liste des réseaux
- Liste des fossés

> Gestion des tampons

3449 tampons affichés

Filter

[Revenir à l'accueil](#)

Voir	Code	Denomination	Adresse	Profondeur	Diamètre	Modifier	Supprimer
	Par.1	AVENUE DENIS PAPIN	hj 95400 ARNOUVILLE	3.25	1500		
	Par.2	AVENUE DENIS PAPIN	95400 ARNOUVILLE	3.35	1500		
	Par.3	AVENUE DE CHOISEUL	95400 ARNOUVILLE	0.85/3.85	1500		
	Par.4	AVENUE DE CHOISEUL	95400 ARNOUVILLE	1.70/3.80	1500		
	Par.5	AVENUE DENIS PAPIN	95400 ARNOUVILLE	2.90	1500		
	Par.6	PLACE DU GENERAL LECLERC	95400 ARNOUVILLE	2.40	1500		
	Par.7	PLACE DU GENERAL LECLERC	95400 ARNOUVILLE	1.78	500		
	Par.8	PLACE DU GENERAL LECLERC	95400 ARNOUVILLE	2.10	800		
	Par.9	SUD LYCEE HORTICOLE	95400 ARNOUVILLE	2.35	dalot		
	Par.10	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.11	SUD LYCEE HORTICOLE	95140 GARGES-LES-GONESSE		dalot		
	Par.12	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.13	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.14	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot 2.00x2.15		
	Par.15	SUD LYCEE HORTICOLE	95400 ARNOUVILLE		dalot		
	Par.16	ROND POINT DU CHRIST	95400 ARNOUVILLE		dalot		
	Par.17	AVENUE DE STALINGRAD	95140 GARGES-LES-GONESSE	3.45	dalot		
	Pbf.5165	RUE LOUISE MICHEL	95330 DOMONT	2.05	900		

# Annexe 4 - Modification de l'ouvrage tampon Par.1

**SIAH** 5SOWSO

Accueil

- Visites
- Liste des visites
- Ouvrages
- Liste des tampons**
- Liste des réseaux
- Liste des fossés

Bonjour **Morgan KAISER**

Merci de penser à sauvegarder vos modifications

< Retour

## > Modification de l'ouvrage tampon Par.1

Requis

**Code**  
Par.1

**Dénomination**  
AVENUE DENIS PAPIN

**Rue**  
Lorem ipsum dolor sit amet

**Commune**  
ARNOUVILLE (95400)

Requis

**Profondeur**  
3.25

**Diamètre**  
1500

Requis


Plan | Satellite

Revenir à l'accueil

Google | Données cartographiques | Conditions d'utilisation | Signalez une erreur cartographique


Valider







# Annexe 5 - Affichage des informations de l'ouvrage tampon Par.1



OSMOSE

Bonjour  
**Morgan KAISER**



> Tampon **Par.1**

< Retour

Informations

Visites 1

Photos 2

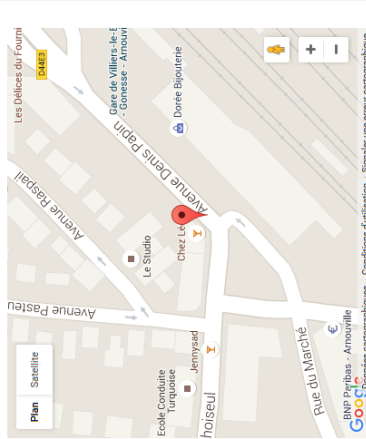
<b>Code</b>	Par.1	<b>Profondeur</b>	3,25
<b>Dénomination</b>	AVENUE DENIS PAPIN	<b>Diamètre</b>	1500
<b>Adresse</b>	hj 95400 ARNOUVILLE		

Modifier Tampon

Retour à l'accueil

Accueil

- Visites
- Liste des visites
- Ouvrages
- Liste des tampons
- Liste des réseaux
- Liste des fossés





## Annexe 6 - Affichage des photos des visites de l'ouvrage tampon Par.1

The screenshot displays a mobile application interface. On the left is a vertical navigation bar with the SIAH logo (350W50) at the top, followed by the text 'Accueil' and a series of icons: a mail icon, a folder icon, a document icon, a magnifying glass icon, a person icon, and a building icon. Below these icons is the text '> Tampon Par.1'. At the top of the main content area, there is a header with 'Bonjour Morgan KAISER' and a home icon, and a 'Retour à l'accueil' button. Below the header is a 'Modifier Tampon' button. The main content area is divided into two sections. The top section shows a list of photos under the heading 'Photos'. There are two photo thumbnails, each with a date '07/06/2016'. The bottom section shows a menu with the following items: 'Visites', 'Liste des visites', 'Ouvrages', 'Liste des tampons', 'Liste des réseaux', and 'Liste des fossés'. A 'Retour' button is located at the top left of this menu section.

# Annexe 7 - Affichage de l'historique des visites de l'ouvrage tampon Par.1

**SIAH**  
OSMOSO

Bonjour  
**Morgan KAISSER**

Accueil
Retour

> Tampon **Par.1**

Revenir à l'accueil

Visites
Infos
Photos

Modifier tampon

**Visites**

Liste des visites

**Ouvrages**

Liste des tampons

Liste des réseaux

Liste des fossés

Visite du 07/06/2016 à 11:14 par Morgan KAISSER

**Informations**

**Date**  
07/06/2016

**Ouvrage**  
Par.1

**Pollutions**  
Pesticides, Pollution thermique

**Remarque**  
Commentaire pollution

06/2016

05/2016

04/2016

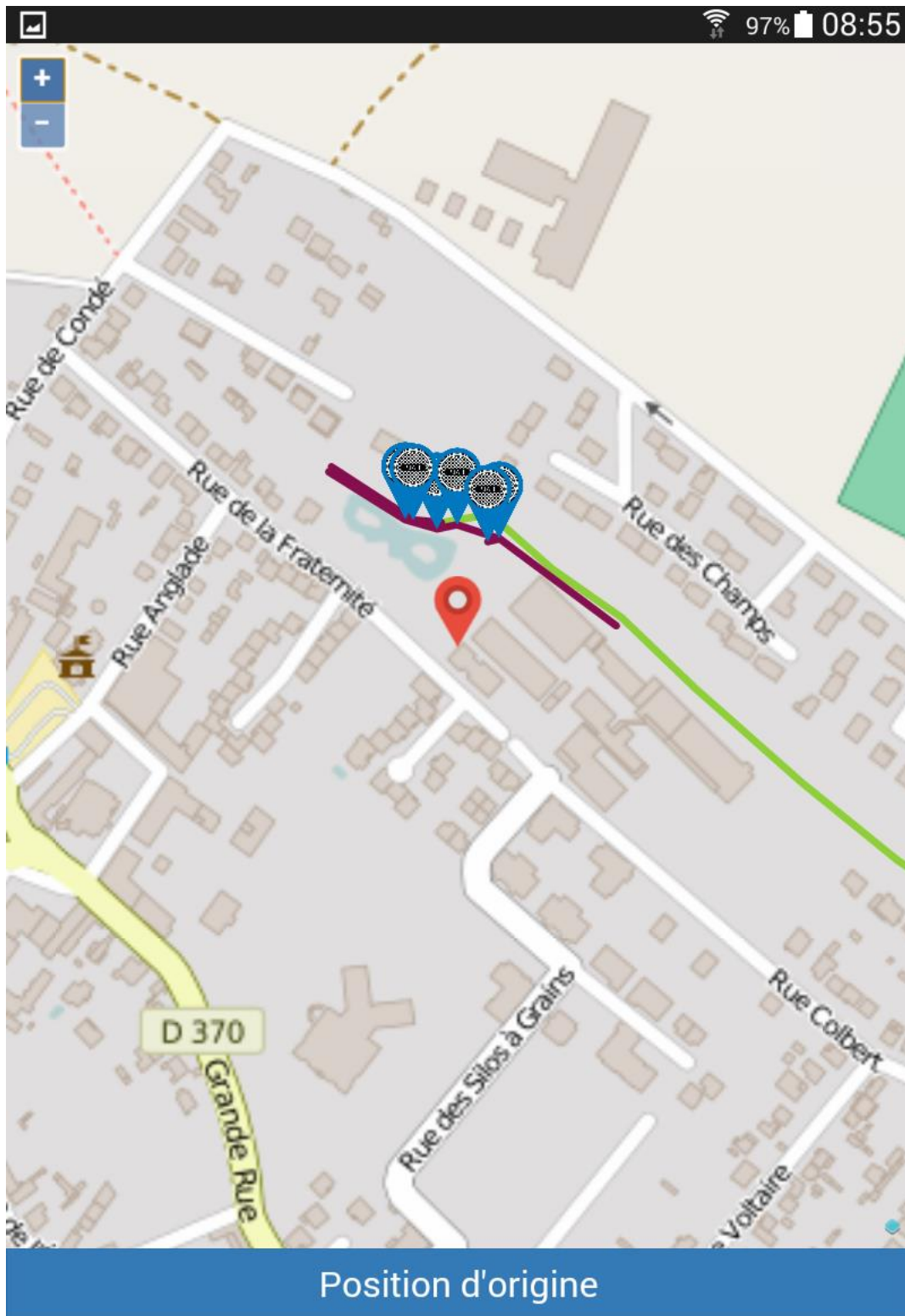
## Annexe 8 - Filtre sur la liste des visites

Filtrer les visites

Code ouvrage		Type	Aucune sélection	Agent	Aucune sélection	Remarque	
Date du		Pollutions		Ville			
au							

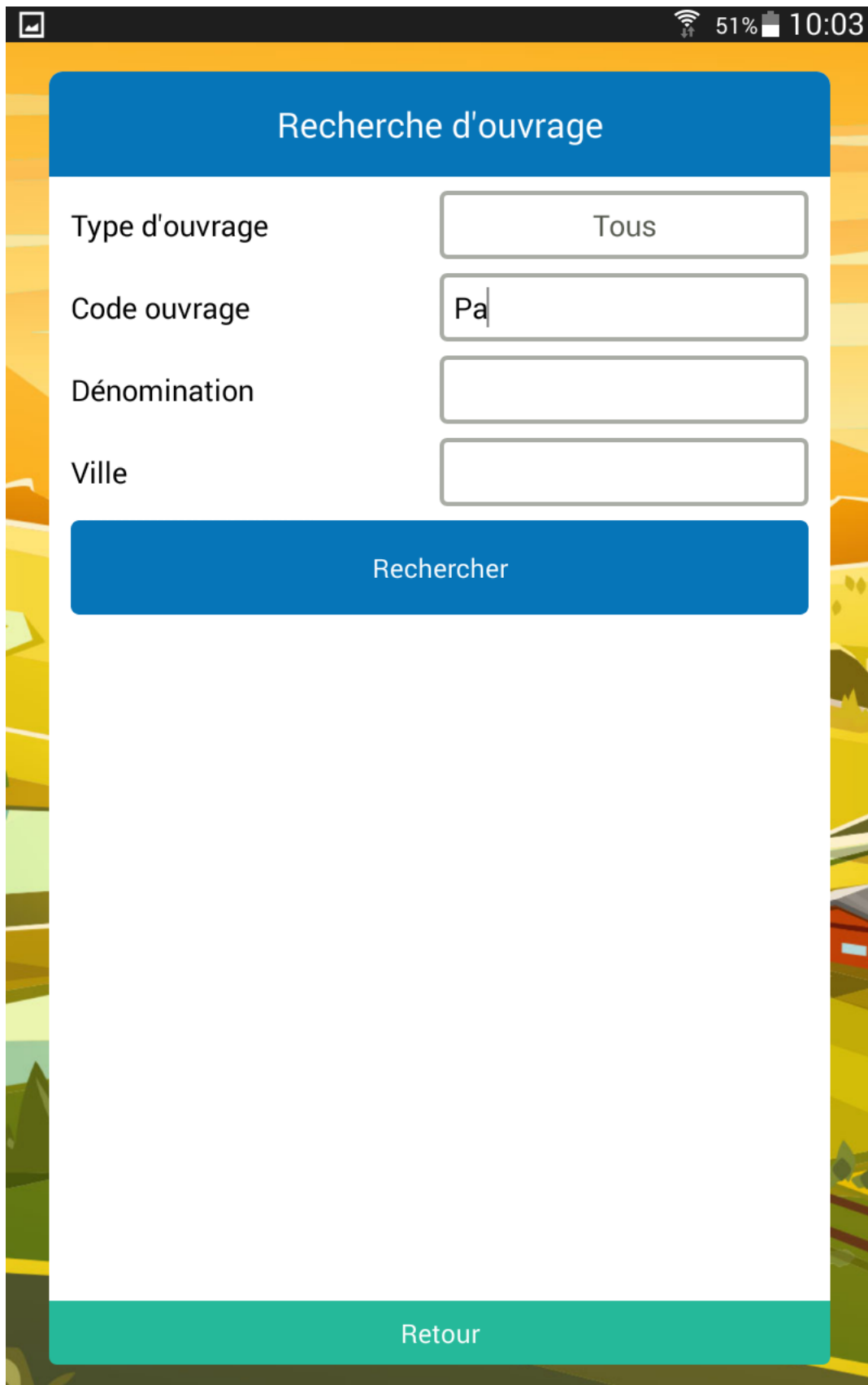
Filter

## Annexe 9 – Activité présentant la carte



Retour

## Annexe 10 - Activité présentant le formulaire de recherche d'ouvrage



The image shows a mobile application interface for searching for works. At the top, there is a status bar with a signal icon, 51% battery, and the time 10:03. Below this is a blue header with the text 'Recherche d'ouvrage'. The main form area contains four input fields: 'Type d'ouvrage' with a dropdown menu showing 'Tous', 'Code ouvrage' with the text 'Pa' entered, 'Dénomination', and 'Ville'. Below these fields is a large blue button labeled 'Rechercher'. At the bottom of the form is a teal button labeled 'Retour'.

Recherche d'ouvrage

Type d'ouvrage

Code ouvrage

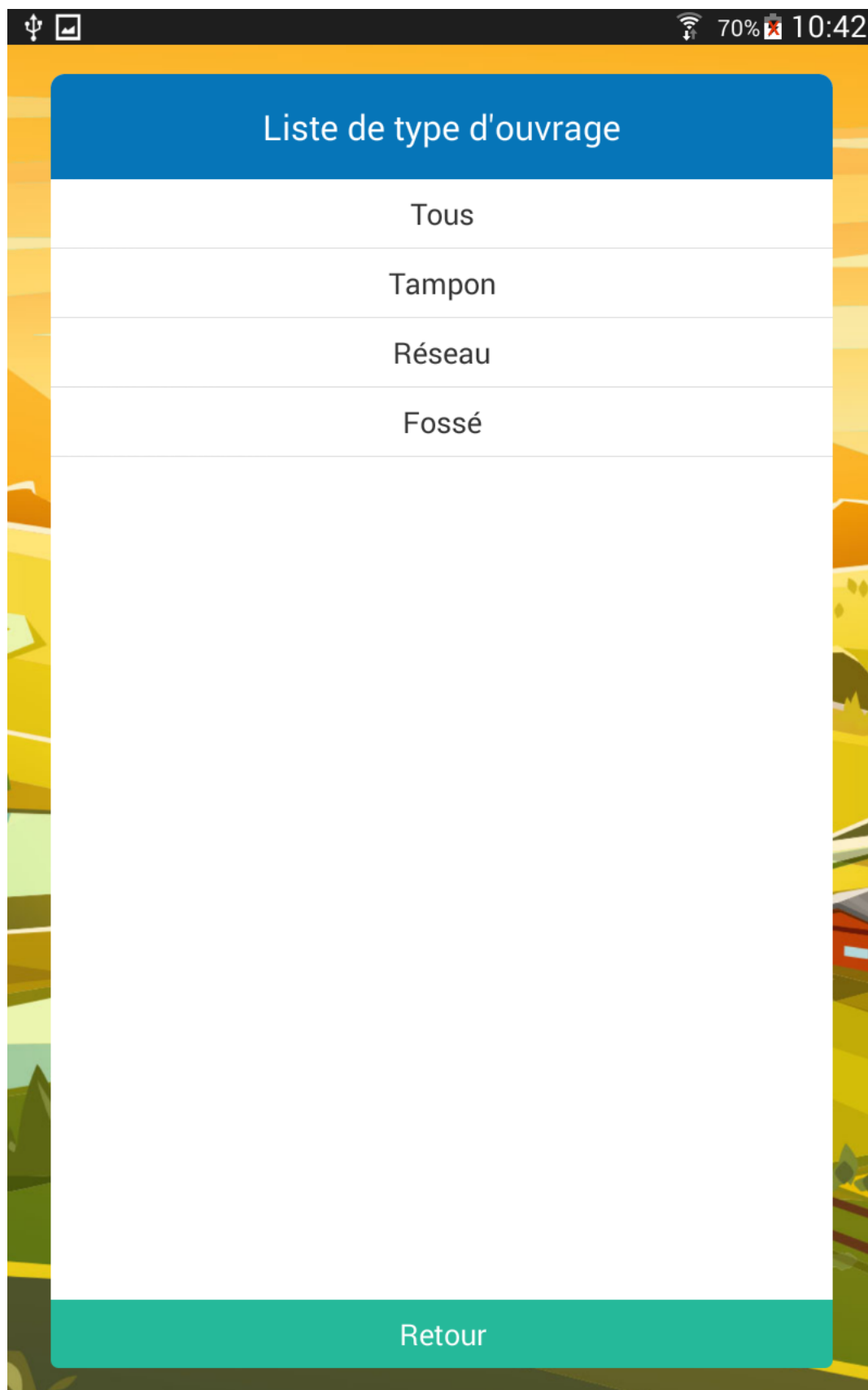
Dénomination

Ville

Rechercher

Retour

## Annexe 11 - Activité présentant les types d'ouvrage



## Annexe 12 - Activité présentant les résultats de la recherche sur le code "Pa"

Enregistrement capture d'écran...

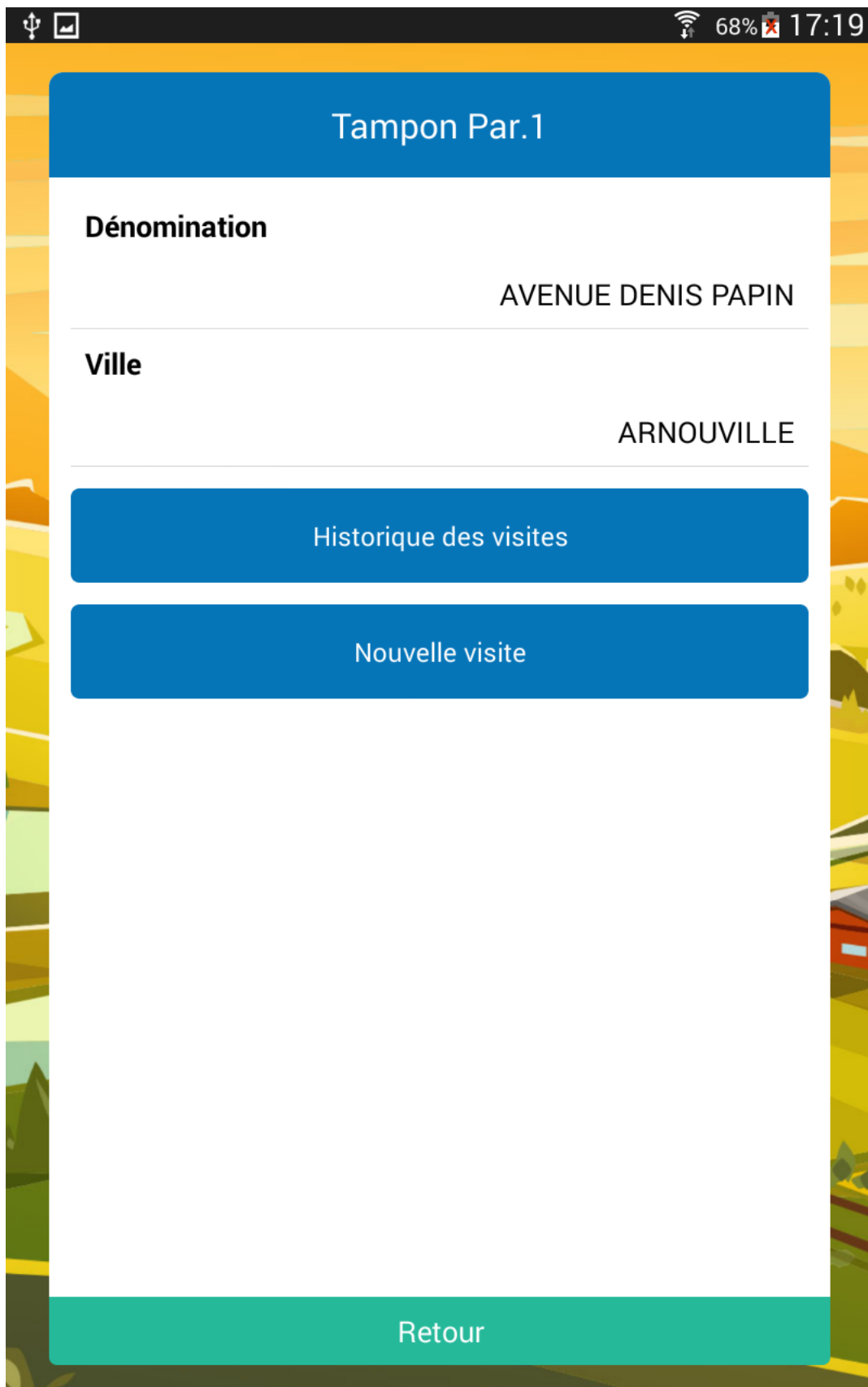
### Liste des ouvrages

25 ouvrages sur 44 au total

Par.1	ARNOUVILLE
Par.2	ARNOUVILLE
Par.3	ARNOUVILLE
Par.4	ARNOUVILLE
Par.5	ARNOUVILLE
Par.6	ARNOUVILLE
Par.7	ARNOUVILLE
Par.8	ARNOUVILLE
Par.9	ARNOUVILLE
Par.10	ARNOUVILLE
Par.11	GARGES-LES-GONESSE
Par.12	ARNOUVILLE
Par.13	ARNOUVILLE
Par.14	ARNOUVILLE
Par.15	ARNOUVILLE
Par.16	ARNOUVILLE

Retour

## Annexe 13 - Activité présentant les informations d'un ouvrage





## Annexe 14 - Activité présentant l'historique des visites d'un ouvrage

Enregistrement capture d'écran...

Liste des visites		
07/06/2016 11:14:35	Par.1	ARNOUVILLE
02/05/2016 10:41:00	Par.1	ARNOUVILLE
29/04/2016 04:59:00	Par.1	ARNOUVILLE

Retour

## Annexe 15 - Activité création/modification de visite

**Nouvelle visite**

**Par.1** Choisir un ouvrage

Pollutions Ajouter une pollution

Autre(s) pollution(s) détectée(s)

Remarques

Prendre une photo

Retour Valider

Supprimer la visite

## Annexe 16 - Activité d'ajout de pollution(s)

66% 11:14

### Liste de type de pollution

Matières organiques	<input type="checkbox"/>
Matières en suspension	<input type="checkbox"/>
Matières azotées	<input type="checkbox"/>
Phosphore	<input type="checkbox"/>
Chlorures	<input type="checkbox"/>
Sulfates	<input type="checkbox"/>
Métaux	<input type="checkbox"/>
Produits organo-chlorés	<input type="checkbox"/>
Benzène-toluène-xylène	<input type="checkbox"/>
Hydrocarbures aromatiques polycycliques	<input type="checkbox"/>
Pesticides	<input checked="" type="checkbox"/>
Radio-éléments	<input type="checkbox"/>
Pollution thermique	<input checked="" type="checkbox"/>

Retour Valider

## Annexe 17 - Activité nouvelle visite avec des photos

The screenshot shows a mobile application interface for editing a visit. At the top, there is a blue header with the text "Modification de la visite". Below this, the form is organized into several sections:

- Par.1**: A green button labeled "Choisir un ouvrage" is positioned to the right of the text "Par.1".
- Pollutions**: The text "Pollutions" is followed by "Pesticides, Pollution thermique". To the right of this text is a green button labeled "Ajouter une pollution".
- Autre(s) pollution(s) détectée(s)**: Below this text is a white rectangular input field.
- Remarques**: Below the input field is another white rectangular input field.
- Prendre une photo**: A large green button with this text is located below the input fields.
- Photos**: Two square images showing trees and a landscape are displayed below the "Prendre une photo" button.
- Navigation**: At the bottom, there are three buttons: a green button labeled "Retour", a blue button labeled "Valider", and a red button labeled "Supprimer la visite".

The top status bar of the phone shows a signal strength icon, a Wi-Fi icon, 66% battery, and the time 11:16.

## Annexe 18 - Activité d'édition du commentaire d'une photo



## Annexe 19 - Activité de sélection des visites à envoyer au serveur



