



Institut universitaire de technologie de Marne-la-Vallée
Département Informatique

RAPPORT D'APPRENTISSAGE

Présenté par Alex BAIET

Optimisation des logiciels de gestion de données

Référent universitaire : Tewfik ETTAYEB

Maître d'apprentissage : Cyrille LE FORESTIER, Jean-Baptiste BARREAU

Année universitaire 2021-2022

Sommaire

Sommaire	3
Remerciements	5
Résumé	6
Summary	7
Introduction	1
Présentation de l'association	2
Missions réalisés	4
I. Base de données Archéologie du handicap (Archeohandi)	4
II. Site ADN	14
Problématique	16
Conclusion	19
Webographie / Bibliographie / Citographie	21
Glossaire	22
Table des annexes	24

Remerciements

Avant de présenter mon alternance chez ADN, je tiens à remercier **Cyrille Le Forestier** et **Jean-Baptiste Barreau** de m'avoir pris comme alternant et ainsi permis d'acquérir une première expérience professionnelle avec une variété de missions que j'aurais eu beaucoup de mal à avoir ailleurs. Je tiens aussi à remercier **Amalaric Le Forestier** sans qui je n'aurais jamais trouvé cette alternance chez ADN. Je tiens aussi à remercier mon référent universitaire **Tewfik Ettayeb**, et la responsable de l'association **Micheline Kerien**.

Résumé

L'association ADN (Archéologie des nécropoles) est une association proposant des activités en lien avec l'archéologie. Cette association, n'ayant pas de développeur et ayant besoin de développer différents outils informatiques, m'a donc recruté pour satisfaire ces besoins sous la tutelle de Cyrille Le Forestier et Jean-Baptiste Barreau.

Mes missions consistait donc à développer des outils informatiques divers, dont les missions principales étant :

- La correction d'un site web nommé "*Archéologie du Handicap*" de gestion de données concernant les restes de sujets présentant des symptômes prouvant un handicap retrouvés lors des fouilles archéologiques;
- La reconstruction du site web présentant l'*Archéologie des Nécropoles* et ses différentes activités (<https://archeologie-des-necropoles.com/>).

Il y a autres tâches à venir telle que la création d'un plugin en Python pour QGIS¹, une base de données contenant tous les sujets (pas uniquement les sujets handicapés) sur le site d'ADN, des algorithmes à faire en langage **R**², et, si il y a assez de temps, un projet visite virtuel en **VR**³.

La plupart des tâches étant en lien étroit avec une base de données (Archéologie du Handicap, base de données d'ADN, plugin QGIS), le savoir concernant l'optimisation des bases de données étaient donc cruciales, car un mauvais savoir sur le sujet pouvait être responsable de nombreux problèmes d'optimisation sur les projets concernés. Il était donc important de savoir comment, dans le cadre de la production de logiciels dédiés au sein d'une entreprise, peut-on optimiser la gestion de grands volumes de données.

¹ Application professionnelle géographique utilisée pour créer et éditer des cartes

² Langage de programmation mathématiques utilisé pour la réalisation de statistiques et de graphiques

³ (*Virtual Reality*) : Équipements et logiciels permettant de s'immerger dans un univers fictif, transmettant les mouvements dans le monde dans cet univers virtuel

Summary

The ADN association (Archéologie des nécropoles) is an association offering activities related to archaeology. This association, having no developer and needing to develop various IT tools, recruited me to satisfy these needs under the supervision of **Cyrille Le Forestier** and **Jean-Baptiste Barreau**.

My missions therefore consisted in developing various IT tools, the main missions being :

- The correction of a website called “Archéologie du Handicap” for managing data concerning the remains of subjects with symptoms proving a disability found during archaeological excavations;
- The rebuild of a website presenting the *Archéologie des nécropoles* and its various activities (<https://archeologie-des-necropoles.com/>).

There are other tasks coming such as the creation of a plugin in Python for QGIS⁴, a database containing all the subjects (not only the disabled subjects) on the ADN’s web site, algorithms to be done in **R**⁵ language, and, if there is enough time, a virtual visit project in **VR**⁶.

Most of the tasks being closely related to a database (Archéologie du Handicap, ADN database, QGIS plugin), knowledge of database optimization was therefore crucial, because poor knowledge on the subject would be responsible for many optimization problems on the projects concerned. It was therefore important to know how, in the context of the production of dedicated software within a company, the management of large volumes of data could be optimized.

⁴ Professional geographic application used to create and edit map

⁵ Mathematical programming language used to create statistics and graphics

⁶ (Virtual Reality): Equipment and software allowing to immerse oneself in a fictitious universe, transmitting the movements in the world in this virtual universe

Introduction

L'association Archéologie des Nécropoles (ADN) a de nombreux besoins concernant l'informatique, les 2 thèmes principaux étant :

- La création et l'exploitation de diverses bases de données pour faciliter l'enregistrement de données diverses concernant les fouilles (ossements retrouvés, lieu, date, ...), et permettre la création simplifiée des statistiques;
- L'amélioration de la visibilité d'ADN via son site web en le rénovant complètement, (amélioration visuel et mise à jour des données), et en ajoutant du nouveau contenu divers, dont une carte exploitant une base de données.

Mais, n'ayant pas de développeur, l'association n'était pas en mesure de réaliser ces tâches. Il m'ont donc recruté pour les accomplir. J'ai donc réalisé ces missions sous la tutelle de Cyrille Le Forestier, travaillant chez ADN et à l'INRAP en tant qu'archéo-anthropologue, et Jean-Baptiste Barreau, informaticien et ingénieur d'étude au CNRS. Jean-Baptiste m'aide pour les tâches informatiques et pour les éléments plutôt techniques, tandis que Cyrille m'apprend des éléments plus globaux, ainsi que des connaissances concernant l'archéologie.

L'alternance s'est donc déroulée du décembre 2021 à août 2022, en raison des difficultés à trouver une alternance je n'ai pas pu commencer en septembre. Elle a eu lieu dans les locaux de l'Inrap à Croissy-Beaubourg, car ADN, étant une petite association, n'a pas de véritable locaux pour le travail. La santé de Jean-Baptiste Barreau ne lui permettant pas de venir en présentiel et Cyrille Le Forestier étant régulièrement sur le chantier, j'ai donc aussi travaillé en distanciel.

Mes missions étaient donc assez diverses, les plus importantes étant centrées sur l'exploitation de bases de données. Les bases de données sont dans les projets un élément capital, car une mauvaise base peut entraîner des problèmes dont la correction peut nécessiter de refaire la structure du projet. Il était donc important d'avoir une base de données solide, autrement c'est toute la structure du projet qui risquait d'en pâtir.

Il fallait donc savoir comment, dans le cadre de la production de logiciels dédiés au sein d'une entreprise, peut-on optimiser la gestion de grands volumes de données.

Présentation de l'association

Présentation

La société m'ayant recruté est **ADN** (*Archéologie des Nécropoles*). C'est une association travaillant sur des fouilles archéologiques, sur des présentations des résultats de ces fouilles ainsi que des activités grand public axées autour de l'archéologie.

Créé en 2014, ADN était à l'origine un regroupement de chercheurs travaillant sur le PCR⁷ **Archéologie des nécropoles mérovingiennes en Île-de-France**. Elle regroupe des personnels de l'Inrap, du CNRS, de l'INED, des collectivités territoriales, des opérateurs privés d'archéologie préventive, des étudiants de Paris I, Paris IV, Nanterre et Aix-Marseille.

L'association s'est ensuite diversifiée dans ces activités, notamment en créant de nombreux jeux pédagogiques permettant d'en apprendre plus sur l'archéologie, comme par exemple "l'Archéo'Scan", un jeu nécessitant de passer une truelle interactive sur un squelette, présentant ensuite les différentes démarches d'un archéo-anthropologue lors de la découverte d'un squelette. Il existe bien d'autres jeux actuellement, que vous pouvez découvrir sur la page suivante : <https://archeologie-des-necropoles.com/activites-archeo-scan.php>.

ADN propose aussi des fouilles ouvertes aux publics en ville chaque année depuis 3 ans, permettant à n'importe qui de découvrir le déroulement d'un chantier de fouille, et de discuter avec les anthropologues sur place. Lors de ces fouilles sont aussi proposées des animations pour attirer un plus large public, et donc permettre aux personnes ayant peu de connaissances sur le sujet de l'archéologie d'élargir leur savoir dans ce domaine.

ADN travaille maintenant sur la fouille programmée de la **nécropole des Mastraits** en collaboration avec la Mairie de *Noisy-le-Grand*, l'*Inrap* et la *Direction Régionale des Affaires Culturelles d'Île-de-France*.

L'association, composée de plus de 100 adhérents et 10 bénévoles, est actuellement dirigé par les personnes suivantes :

- Micheline Kérien, présidente de l'association;
- Elodie Perugini, secrétaire;
- Olivier Gérard, trésorier.

⁷ Projet Collectif de Recherche

L'association est en partenariat avec l'**Inrap** (*Institut national de recherches archéologiques préventives*), qui est un institut public s'occupant des fouilles préventives précédant les constructions des habitations.

Mon environnement

Je suis actuellement le seul développeur au sein de l'association, et comme mon tuteur Cyrille LE FORESTIER n'est pas informaticien, Jean-Baptiste BARREAU spécialisé en informatique et travaillant au CNRS (*Centre National de Recherche Scientifique*) est devenu mon second tuteur avec lequel je peux poser des questions techniques. L'association ADN n'ayant pas de bureau, je travaille actuellement dans les locaux de l'Inrap à Croissy Beaubourg, 56 boulevard Courcerin, où est regroupée une équipe de chercheurs travaillant pour l'INRAP. Cyrille étant parfois sur un chantier et Jean-Baptiste ayant des problèmes de santé l'empêchant de venir en présentiel, une partie de l'alternance s'est passée en distanciel.

Mes missions étant principalement le développement de sites web, j'ai donc dû utiliser des outils permettant la création de sites web, c'est-à-dire les langages *HTML*, *CSS*, *PHP* et *Javascript*, ainsi que les bibliothèques *Bootstrap* et *JQuery*. Lors du projet *Archéologie du handicap*, j'ai aussi utilisé le framework *FuelPHP*. N'ayant pas de restriction sur les outils à utiliser, j'ai choisi de développer avec l'éditeur de code *Visual Studio Code* sous le système d'exploitation *Linux (Ubuntu)*, tout en organisant mes tâches avec *Trello*.

Missions réalisés

I. Base de données *Archéologie du handicap* (Archéohandi)

1. Contexte

Pour comprendre l'intérêt de cette mission, il faut d'abord comprendre comment fonctionne une fouille archéologique. Chaque fouilles est séparé en différentes étapes, étant :

- La délimitation de la zone à fouiller;
- Le décapage permettant d'atteindre les niveaux intéressants pour les fouilles;
- Le recueil et le stockage des données du site de fouilles.

Les données récupérées sont ensuite utilisées pour faire des études.

Le problème venait de l'étape de stockage pour les sujets retrouvés présentant des signes de handicap. Chaque archéologue stockait les données de leurs sujets sur un tableau excel propre à chacun en utilisant un modèle commun. Les soucis étant que :

- Le modèle utilisé n'était pas idéal pour faciliter l'exploitation des données;
- Plutôt que de faire un tableau commun pour tous, chaque archéologue enregistrerait les données sur un tableau personnel hors-ligne : récupérer les données de tous les sujets handicapés nécessitaient donc de demander le tableau de chacun, puis de fusionner les tableaux en un unique tableau avant d'enfin pouvoir faire des études sur les données. Le processus de récupération des données était donc trop complexe et pouvaient prendre beaucoup de temps, par exemple si un des archéologue en question est en vacances;
- L'utilisation d'excel étant trop permissif, et malgré le modèle à suivre, les tableaux de chacun variaient légèrement (utilisation de colonnes différentes, ...), mais ces petites différences rendaient extrêmement compliqué la mise en commun automatique des différents tableaux ensemble pour permettre ensuite de faire des études sur tous les cas enregistrés.
- Une fois les tableaux fusionnés, il ne reste plus qu'à lire les données. Mais là encore, de nombreuses différences dans la notation des valeurs cause des problèmes. Chaque personne va noter un nom selon son opinion personnelle, pouvant par exemple écrire "Archéologie des Nécropoles", "archeologie des necropoles", "ADN" ou "adn" pour un même terme. Pour l'œil humain, il s'agit de la même chose, mais pour un tableau statistique il va s'agir de 4 organismes différents, faussant les résultats.

Il était donc nécessaire de trouver un moyen d'uniformiser et de mettre en commun toutes ces valeurs. C'est donc à ce moment que le projet "**Archéologie du handicap**" est né en 2020.

Voici l'équipe constitué en 2020 :

- Virgile Louin, stagiaire étudiant à l'IUT de Saint-Malo et développeur sur ce projet;
- Rozenne Colleter, archéo-anthropologue à l'Inrap chargé faire le schéma de la base de données;
- Jean-Baptiste Barreau, informaticien au CNRS et tuteur de Virgile Louin.

Cette équipe s'est donc formée pour créer le site web *Archéologie du handicap*, dont l'objectif était d'uniformiser les données en créant des formulaires pour permettre aux utilisateurs de stocker les données récupérées concernant les sujets handicapés retrouvés lors des fouilles dans une base de données. Ce site web devait donc permettre d'entrer les données tout en imposant des contraintes sur les types des données pouvant être entrés, pour faciliter ainsi l'exploitation de ces données à des fins scientifiques via des graphiques ou autres formes d'études.

Ce projet a donc duré 8 semaines (durée du stage de Virgile Louin), et le site web a été créé avec les frameworks **FuelPHP**⁸ et **Bootstrap**⁹ en utilisant les langages **HTML**¹⁰, **CSS**¹¹, **PHP**¹², **SQL**¹³ et **Javascript**¹⁴. Mais au bout de cette limite, le projet, n'étant pas fini et contenant trop de bugs, n'a pas pu être publié. La cause principale étant une mauvaise structure et une mauvaise manière de développer (voir partie "[Résolution des bugs](#)" pour plus de détails).

Ma mission était donc de réparer le site d'Archéologie du handicap avec l'aide de Cyrille Le Forestier, archéo-anthropologue, qui m'a aiguillé sur les tâches à faire en priorité et qui a testé le site, et de Jean-Baptiste Barreau m'ayant aidé pour les questions techniques.

2. Résolution des bugs

La première tâche a donc été de corriger tous les bugs existants et finir ce qu'avait commencé Virgile Louin, mais dû à la mauvaise structure du site, il a été nécessaire de refaire tout la partie back-end¹⁵ du site, ce qui a demandé beaucoup plus de temps que prévu. On peut voir par l'[annexe 1](#) tout de suite certains problèmes comme :

- L'absence de tabulation, rendant la lecture bien plus pénible;
- L'absence de fonctions et trop de copier-coller du code, une erreur classique à ne surtout pas faire en programmation car cela alourdit énormément le

⁸ Framework PHP facilitant la création d'un site en utilisant le modèle MVC

⁹ Bibliothèque de classes CSS

¹⁰ Langage de balisage utilisé pour structurer un site web

¹¹ langage permettant de changer la mise en forme d'un site web

¹² Langage utilisé pour exécuter des instructions du côté du serveur

¹³ Langage permettant d'exécuter des instructions sur une base de données

¹⁴ Langage permettant de rendre dynamique des pages web sur le poste du client

¹⁵ Partie logique cachée d'un projet informatique gérant les actions de l'utilisateur et les calculs

code. De plus, quand une modification est nécessaire, le développeur est obligé de modifier toutes les instances du code copié au lieu de simplement modifier le code à un seul endroit, ce qui fait perdre beaucoup de temps;

- Un manque de commentaires et de documentations, rendant pénible la compréhension du projet.

La seule documentation disponible était donc le rapport de stage de Virgile Louin, ce qui est insuffisant car ce rapport n'a pas pour but de faire comprendre à un développeur l'organisation exacte d'un code (bien que cela m'a quand même été utile). En raison de tous ces problèmes venant du projet, mais aussi de la découverte de FuelPHP qui m'était inconnu, il m'a fallu plusieurs jours avant de bien comprendre le projet et avant de commencer à travailler efficacement.

Restructuration du projet

Comme dit précédemment, la structure du code PHP était fragile, mais la base de données n'était pas en reste en ayant elle aussi de nombreux problèmes. L'objectif était donc de repartir sur un projet bien structuré et clair.

J'ai donc commencé à corriger la base de données en créant un MCD¹⁶ (voir [annexe 2 et 3](#)) et en le faisant confirmer par Jean-Baptiste Barreau, étant donné qu'il avait déjà travaillé sur le projet. J'ai donc dans la base de données supprimé de nombreuses informations inutiles, comme des champs "id" servant de clé primaire¹⁷ n'étant pas toujours nécessaires, des tables inutiles, et des relations trop complexes pour la table des diagnostics.

L'étape suivante était de refaire toute la partie logique (code en PHP et Javascript), c'est-à-dire tout le code permettant de relier l'interface du site à la base de données, ainsi que mettre en commun toutes les fonctionnalités concernant plusieurs pages, et ajouter des commentaires et de la documentation pour les futurs développeurs. Cette correction s'est faite en profitant du modèle MVC proposé par FuelPHP. J'ai donc commencé par réorganiser les pages pour donner l'arborescence de navigation suivante.

¹⁶ Schéma d'une base de données suivant des règles précises

¹⁷ Champ(s) d'une table* servant à l'indexation des données

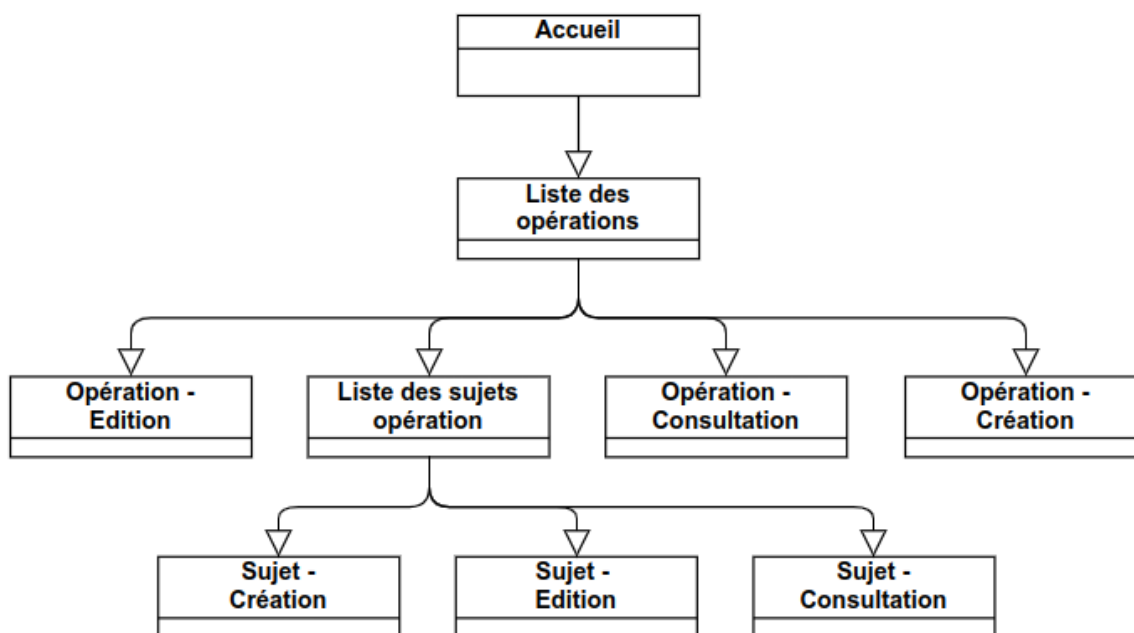


Schéma de la navigation d'*Archéologie des Nécropoles* après correction

La correction de la liaison à la base de données et la suppression du code dupliqué s'est faite en créant des modèles. D'un côté, des classes PHP contenant les fonctions en double ont été créées pour permettre l'accès simplifié aux fonctions communes entre les différentes pages du site. De l'autre côté, une liste de classes représentant chaque table de la base de données a été créée, facilitant la récupération des données, et offrant une autocomplétion¹⁸ sur les différents champs des tables, permettant de retrouver facilement le nom d'un champ.

Un système d'autocomplétion facilitant la recherche d'une commune pour une opération donné était déjà en place dans le site avec Javascript et l'outil Ajax¹⁹ de JQuery, créé par Virgile Louin. Ce système fonctionnait bien, mais là encore les fonctions concernés étaient dupliquées et il était nécessaire de mettre en commun le code. A ce stade du projet, je n'étais encore qu'un novice en Javascript et ne connaissais pas JQuery. Il m'a donc fallu apprendre comment cela fonctionnait avant de pouvoir mettre en commun les duplicatas de ces fonctions d'autocomplétion. Cette tâche a donc pris plus de temps que prévu, mais m'a permis de découvrir JQuery.

Seule la partie graphique a été conservée. Le fait de simplement restructurer le code a permis de régler la plupart des bugs sans même avoir essayé de les résoudre. De plus, avoir déjà une structure existante m'a permis de savoir quoi faire, m'épargnant ainsi le travail de la création d'un cahier des charges.

¹⁸ Système proposant de finir un mot / une expression automatiquement

¹⁹ Permet de faire des requêtes HTTP avec Javascript

Correction des bugs

La correction des bugs a suivi la correction de la structure. L'objectif ici était de rendre fonctionnel l'entrée de nouvelles valeurs, ainsi que l'affichage des valeurs existantes. Il existe 2 types de valeurs à rendre fonctionnel :

- Les sujets handicapés;
- Les opérations, contenant une liste de sujets handicapés.

Chacune de ces valeurs sont représentées en tant que table²⁰ dans la base de données. Mais ces tables sont reliées à de nombreuses valeurs ([voir annexe 2 et 3](#)), complexifiant grandement la correction de l'entrée des données.

L'utilisateur doit d'abord pouvoir choisir les données à entrer. Il faut donc pour chaque champ mettre des contraintes adaptés aux formats des champs dans la base de données (limite du nombre de caractères pour un champ, autorisation uniquement des nombres pour un champ). Certains champs ne peuvent avoir que certaines valeurs définies dans une table (exemple : le champ "sexe" du formulaire des sujets ne peut avoir comme valeur que "homme", "femme" et "indéterminé"). Cela nécessite de faire une requête sur la table en question, puis d'afficher la liste des valeurs disponible sur le formulaire à l'aide d'une balise "<select>".

Certains champs ne peuvent avoir une valeur d'une liste définie, mais cette liste peut-être conséquente, comme le champ de choix de la commune où l'affichage de tous ces éléments serait trop lourd (36701 éléments différents). Il y a donc eu nécessité de pouvoir chercher la valeur facilement. La solution choisie a été de faire un champ utilisant l'**autocomplétion**, proposant une liste de valeurs correspondant approximativement à la valeur saisie par l'utilisateur. Cette fonctionnalité avait déjà été faite par Virgile Louin en utilisant Ajax, je ne me suis que contenté d'améliorer cette fonctionnalité.

L'utilisateur doit ensuite pouvoir entrer les données dans la base de données et les modifier. Il fallait donc bien gérer les étapes suivantes :

- L'insertion des données
- La modification des données
- La suppression des données

Chacune de ces étapes, si mal maîtrisées, peuvent causer des pertes de données diverses (donnée utile effacée ou donnée attachée à aucune entité, le rendant inutile et remplissant la mémoire inutilement). En effet, il faut à chaque fois ne pas que penser à la modification des tables "operation" et "sujet", il faut aussi penser aux tables attachées et donc les éditer si besoin. C'est pourquoi ces étapes sont primordiaux dans l'édition des données par l'utilisateur.

²⁰ Dans une base de données, correspond à un tableau de valeur

La plupart des champs sont assez banales et ne vont pas être traités dans ce rapport, mais certains champs ont demandé une plus longue réflexion. En voici la liste :

- Le champ des organismes est un champ utilisant l'autocomplétion, mais la liste des valeurs n'est pas fixe : Chaque personne devait pouvoir indiquer un nouvel organisme dans le cas où celui-ci ne serait pas stocké dans la base de données.
- La sélection de la commune est composée de 2 champs, un pour le département et le deuxième pour le nom de la commune. Il s'agit donc d'une paire de champ utilisant l'autocomplétion, dont les valeurs dépendent l'une de l'autre. Par exemple, la sélection de la commune "Montfermeil" étant située en Seine-Saint-Denis, ne peut se faire que si le champ "département" a comme valeur "Seine-St-Denis" ou est vide.

La gestion des diagnostics des sujets handicapés est la partie la plus complexe, car il est relié au sujet via une relation ternaire avec la localisation du diagnostic ([voir annexe 3](#)). La sélection des diagnostics se fait donc via un tableau à 2 entrées (nom du diagnostic pour les lignes, localisation sur le sujet pour les colonnes).







		D	G		D	G
						
<input type="checkbox"/> Trépanation	<input type="checkbox"/>					
<input type="checkbox"/> Édentement complet	<input type="checkbox"/>					
<input type="checkbox"/> Atteinte neurale	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Scoliose sévère				<input type="checkbox"/>		
<input type="checkbox"/> Maladie de Paget ou ostéite déformante	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> DISH	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Rachitisme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Nanisme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Fracture non réduite	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Amputation		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Pathologie dégénérative sévère	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Autre	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tableau de sélection des diagnostics

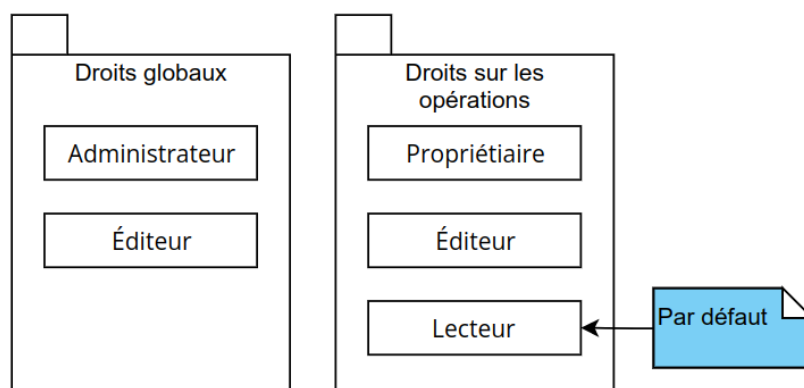
Après avoir résolu tous ces bugs et fini le travail de Virgile Louin, le site *Archéologie du handicap* était enfin prêt à être utilisé et rempli de données.

3. Création des comptes

Cependant, n'ayant pas de système de gestion des droits, ce qui signifie que n'importe quel utilisateur connaissant l'url²¹ du site pouvait facilement ajouter ou supprimer des données, sans même avoir besoin de connaissance en cybersécurité. Ma seconde tâche a donc été d'ajouter un système de compte avec différents niveaux de droits pour sécuriser le site et l'accès aux données.

²¹ Adresse permettant d'accéder à un site web

Les comptes sont séparés selon leur droit, avec les comptes “éditeur” ayant le droit d’ajouter et de modifier certaines opérations et sujets existants, et les comptes “administrateurs”, ayant tous les droits, y compris créer d’autres comptes. Les comptes éditeur peuvent donc créer une nouvelle opération, et faire ce qu’il veut avec. En revanche, il ne peut que lire les informations des autres opérations, sauf si le propriétaire de l’opération l’a autorisé à éditer l’opération en renseignant le nom du compte dans l’opération en question, faisant du compte un “éditeur”. (voir le champ dans l’[annexe 4](#)).



Représentation des différents droits des comptes

Il a d’abord fallu créer les comptes sur la base de données et les relier aux opérations pour permettre aux différents comptes d’avoir des droits spécifiques pour chaque opération (voir [annexe 3](#)). Pour des raisons de sécurité, les mots de passe sont encodés avec l’algorithme **MD5** sur la base de données.

Ensuite, il a fallu automatiser la création de comptes en envoyant des mails automatiques, en suivant ces étapes :

- L’utilisateur sans compte demande une création de compte en remplissant la page de création de compte, envoyant automatiquement un mail de confirmation aux administrateurs;
- Un administrateur valide la création du compte, envoyant par mail les identifiants du compte
- L’utilisateur peut ensuite se connecter avec les identifiants et commencer à profiter du site.

Le site, devenu exploitable et sécurisé, pouvait maintenant être publié.

4. Exploitation

Après avoir réparé le site puis l’avoir sécurisé, il était temps d’exploiter les données. L’objectif était d’utiliser les données pour faire des statistiques scientifiques concernant les sujets handicapés, comme par exemple le taux de sujets ayant un diagnostic précis dans une zone, etc, mais aussi de pouvoir exporter les données pour ne pas se limiter qu’aux graphiques.

L'idée pour résoudre ce problème était de faire une page de recherche, pratique pour récupérer tous les sujets ayant un trait commun, et de d'utiliser ces résultats de recherche pour en faire des graphiques (voir l'[annexe 7](#)).

Recherche

La page de recherche s'est construite en reprenant des éléments déjà connus par les utilisateurs, c'est-à-dire en reprenant la même forme que les formulaires de création / édition des sujets et opérations. De même pour la page de résultat, qui affiche les résultats sous forme d'un tableau ayant le même format que le tableau listant toutes les opérations disponibles.

En raison des nombreuses données existantes, il a fallu faire un effort d'optimisation des requêtes pour éviter un chargement trop lent des résultats de la recherche, étant donné qu'il était obligatoire de pouvoir récupérer toutes les données d'un coup pour créer les graphiques concernant la globalité des données (donc pas de limite du nombre de résultats).

Graphiques

La création des différents graphiques s'est faite grâce à la Highchart, une bibliothèque Javascript permettant de faire différentes sortes de graphiques personnalisables. Les données ont donc été représentées via :

- Des camemberts pour connaître le ratio de de sujets ayant un trait spécifique dans une population;
- Des graphiques en colonnes pour connaître le nombre de sujet concerné par une caractéristique.

De plus, l'affichage des résultats de recherche a demandé d'utiliser PHP, tandis que l'utilisation des graphs a nécessité l'utilisation de Javascript, ce qui a posé un second problème : comment relier des données pour des langages totalement différents. J'ai pour cela récupérer les données sous forme de JSON²² via PHP et SQL et stocké ce JSON dans la page HTML, pour permettre à Javascript d'accéder et d'exploiter ces données sans avoir besoin de refaire la requête de recherche.

Export des données

Pour l'export des données, le choix du format a été le format CSV²³. Ce format permet une lecture par un tableau excel pour les utilisateurs lambdas, et peut être exploité par un script pour des utilisations plus avancées grâce à son format de données clair.

²² Format d'écriture de données standardisé

²³ Format de tableau de données textuel

Cet export peut se faire via un bouton utilisant une fonction Javascript, exploitant les données JSON déjà utilisées pour créer les graphiques citées précédemment.

5. Problèmes rencontrés

Durant la création du site web, j'ai dû faire face à plusieurs problèmes :

- Le problème de structure au début du projet (voir le 1er paragraphe dans la partie "tâche réalisé")
- La découverte du framework²⁴ **FuelPHP**, que je ne connaissais pas avant cela. A l'apprentissage de ce framework s'est ajouté le problème d'une documentation très légère : FuelPHP étant un framework peu connu, peu de questions sont posées en ligne et donc peu de réponses existent, contrairement à n'importe quel autre framework ou langage plus connu.
- L'utilisation de **Javascript** et **JQuery**, bien que mineur dans ce projet, a été un souci car mes connaissances en Javascript étaient limitées, et JQuery m'était inconnu. Il a donc fallu que je me forme sur ces aspects.
- Bien que connaissant le SQL avant de commencer cette mission, la complexité de la base de données et la gestion de très nombreux champs dans une même table (voir [annexe 3](#)) était une nouveauté pour moi, et cela m'a contraint à bien structurer le projet pour éviter de perdre du temps à écrire des requêtes similaires.

²⁴ Pack d'outils informatique facilitant la création d'un logiciel

II. Site web ADN

1. Contexte

L'association ADN utilise différents moyens de communication pour se faire connaître, dont un site web présentant leurs activités et leurs réalisations (<https://archeonec.hypotheses.org/>). Cependant, ce site web n'était plus à jour et l'aspect visuel du site était améliorable. Ma mission était donc de refaire le site, de l'améliorer et de le mettre à jour. Voici le lien vers la nouvelle version du site web : <https://archeologie-des-necropoles.com/>

2. Structuration du site web

Avant de commencer à recréer le site, il a d'abord fallu mettre en place le projet et choisir les outils à utiliser. Étant donné qu'il était probable qu'un futur développeur travaille sur le site web, j'ai opté pour des outils simples que tous les développeurs connaissent et aucun framework, c'est-à-dire **PHP**, **HTML**, **CSS** et **Javascript**.

La difficulté du projet était de remettre en forme une énorme quantité de pages, de textes et de régler la qualité des nombreuses images. Corriger les pages une par une étant extrêmement lent (et ennuyeux), il a fallu trouver un moyen de modifier un maximum d'éléments en un minimum d'instructions de ma part, et cela s'est fait via des scripts automatisant la modification du site. J'ai donc créé différents scripts, dont un script **Bash**²⁵ pour changer la taille des images rapidement, et un script **Python**²⁶ pour mettre en forme les différentes pages du catalogue des sites de fouilles ([Lien vers le catalogue](#)).

En effet, il était donc question d'automatiser le traitement d'une grande quantité de données lors de cette tâche. Pourtant, il ne s'agissait pas d'un format de données habituelles, car ce n'était pas des données contenues dans une base de données ou écrites dans un fichier dans un format exploitable (CSV, JSON, ...), mais de données relatives à la structure d'un site. Il s'agissait donc d'une forme de traitement bien différente de d'habitude.

3. Carte des sites de fouilles

Une des raisons de la restructuration du projet était de permettre l'ajout des éléments plus divers tel qu'une roue de la fortune en javascript et une carte des différents sites fouillés, ce qui était compliqué sur l'ancienne version du site.

²⁵ Langage de programmation uniquement sur les systèmes Unix, souvent utilisé pour faire des scripts touchant au système d'exploitation

²⁶ Langage de programmation interprété polyvalent

J'ai donc commencé à faire la carte des sites de fouilles, donnant le résultat de l'[annexe 8](#). Cette carte, étant la tâche actuellement en cours, est encore incomplète. Cette mission s'est basée sur un fichier CSV contenant toutes les données des sites à afficher. Je me suis donc basé sur ce fichier de données pour créer un schéma de la base de données ([annexe 9](#)), puis créer la base de données. J'ai ensuite importé les données du fichier vers la base de données en utilisant PHP, ce qui a été problématique pour certaines raisons :

- L'encodage du texte utilisé par le fichier, PHP et la base de données étant différents, l'import des valeurs ayant des caractères spéciaux a causé certains problèmes;
- Les données dont la valeur est inconnue sont représentées par la valeur "0". Par conséquent, il est impossible de savoir si un nombre étant à 0 représente la valeur "0" ou la valeur "inconnu".

La base de données maintenant remplie, il fallait afficher tous les sites sur une carte, et pouvoir récupérer les informations d'un site en cliquant sur un marqueur sur la carte. La carte étant construite à l'aide de la bibliothèque Leaflet et Javascript, la difficulté était d'accéder aux données via Javascript. Pour cela, j'ai donc créé une API permettant de récupérer les informations de tous les sites existant dans la base de données au format JSON, un format facilement exploitable via Javascript. Une fois les données récupérées, j'ai ensuite pu facilement afficher les sites sur la carte et afficher les informations d'un site précis lors de la sélection sur la carte.

La carte sera complétée avec d'autres éléments, tels que la complétion de la base de données étant actuellement incomplet lors de la réalisation de ce rapport, ainsi que par l'ajout d'un formulaire permettant d'ajouter des nouveaux sites sur la carte.

4. Problèmes rencontrés

Ayant dû réaliser le projet de A à Z, il a fallu que je m'occupe de créer le site, mais aussi de le publier et le rendre visible aux navigateurs web, ce que je ne savais pas faire avant. Cela m'a permis de découvrir comment utiliser les balises "<meta>", le "robots.txt" et le "sitemap", qui permettent/améliorent la visibilité du site sur les navigateurs.

Problématique

I. Création de la base de donnée

La création d'une base de données est toujours une phase très importante d'un projet, car si cette étape est mal faite, elle peut ralentir l'application l'utilisant, et dans le cas où la base de données doit être refaite, l'application l'exploitant devra aussi être refaite. C'est pourquoi il faut prendre le temps de respecter les quelques éléments qui vont suivre.

a) Clarté des noms des tables et champs

Une bonne clarté dans le nommage des champs aide dans l'utilisation de la base de données et évite de faire des erreurs de confusion entre différents champs. Il existe d'ailleurs différentes conventions pour ça, libre à vous de choisir la votre, du moment que la convention de nommage utilisée est homogène au sein de la base.

b) Problèmes récurrents des bases de données

En éditant une base de données, il peut y avoir des problèmes lors de l'édition des données. Il en existe 3 types :

- **L'anomalie d'insertion** : Le problème d'une insertion impossible peut exister si cette insertion demande l'existence d'une donnée sur une autre table. Par exemple, un employé existe mais ne travaille pour aucun service, mais la BDD nécessite d'avoir l'employé relié à 1 service. Il faut alors créer un service "nul" avant d'entrer les données de l'employé.
- **L'anomalie de suppression** : La suppression d'une donnée peut supprimer une autre donnée reliée, mais que l'on ne veut pas supprimer. Par exemple, si les informations de l'employé et du service sont dans la même table, la suppression de l'employé supprime aussi le service.
- **L'anomalie de modification** : La modification d'une table nécessite la modification d'une ou plusieurs autres tables pour garder la base cohérente. Par exemple, modifier un service peut demander de mettre à jour tous les employés concernés.

Il faut donc prendre en compte toutes ces anomalies lors de la création de la base de données, sinon l'édition de la base en question peut devenir inutilement complexe.

c) Les valeurs nulles

Les valeurs dites "nulles" sont souvent dérangeantes pour plusieurs raisons :

- La présence de valeurs nulles augmentent l'utilisation de ressources, notamment lors de jointure;
- Ces valeurs gaspille l'espace disponible;
- L'interprétation des valeurs nulles peut varier (la valeur est inconnue, la valeur n'est pas applicable pour le cas actuel, ...).

L'existence des valeurs nulles étant dérangeantes, il faut donc penser la base de données en évitant si possible l'utilisation des valeurs nulles.

d) Normalisation

Une fois la 1ère maquette de la base de données terminée, il existe une méthode qui permet de corriger et optimiser la base de données : la **normalisation**. Il existe 7 étapes de normalisations, mais dans la plupart des cas les 3 premières sont suffisantes :

- La 1ère forme normale consiste à rendre tous les champs atomiques (chaque élément est assigné à un champ distinct), et rendre les valeurs d'une même colonne similaire
- La 2ème forme normale retire de la table toutes les données dépendantes d'une clé primaire
- La 3ème forme normale retire toutes les données dépendantes d'une autre colonne qui n'est pas une clé primaire.

Après cette transformation, la base sera moins gourmande sur l'espace de stockage.

II. Utilisation optimisée

Une fois la base de données créée, on peut maintenant l'exploiter. Mais là encore, il y a de nombreuses règles à respecter pour optimiser les requêtes et éviter des ralentissements sur l'application exploitant la base de données.

a) Optimisation des requêtes

La première chose à faire est d'optimiser les requêtes. Il faut en général penser à de nombreux détails pour bien optimiser une requête, dont voici les plus importants :

- Éviter les jointures autant que possible, car elle multiplie la complexité de la requête : chaque table ajoutée de cette manière multiplie les ressources demandées par le nombre de champs de la table;
- Éviter les "SELECT *" si seul une partie des champs est nécessaire : là encore, les ressources demandées sont multipliées par le nombre de champs des tables, ce qui augmente lors de jointures;
- Éviter les clauses WHERE avec un LIKE commençant par "%". Il s'agit d'un test gourmand pour le processeur.

Optimiser les requêtes est donc important, mais cela ne suffit toujours pas.

b) Nombre de requêtes

Le nombre de requêtes à exécuter impacte grandement la vitesse d'exécution. En effet, il vaut mieux faire une requête contenant beaucoup de conditions, plutôt que de nombreuses petites requêtes, car les bases de données sont optimisées pour traiter de grands volumes de données. De plus, faire de nombreuses requêtes SQL signifie aussi faire un grand nombre de requêtes HTTP²⁷ (si la base de données n'est pas sur le même support que la machine), ce qui est bien plus lent qu'une seule requête HTTP contenant tous les résultats. Il faut donc éviter les requêtes multiples si elles se ressemblent.

c) La clause EXPLAIN

Si, malgré la prise en compte des éléments cités précédemment, un problème de lenteur semble venir des requêtes SQL, il est possible d'avoir des informations sur la requête grâce à la clause **"EXPLAIN"**. Ces informations peuvent aider à comprendre l'origine du problème et à le régler.

Il est donc important de bien optimiser ses requêtes pour éviter toute lenteur sur le logiciel exploitant une base de données.

d) Position géographique de la requête

Du point de vue de l'optimisation, il vaut mieux en général faire les requêtes SQL dans le code PHP que dans le code Javascript (sauf si utilisation de Node.js). Le code PHP est exécuté côté serveur, alors que le code Javascript est exécuté côté client. Cela signifie que les requêtes SQL provenant de PHP sont envoyées directement dans la base de données qui est souvent sur le même support que le serveur web ou dans un réseau proche, signifiant que la réponse de la requête peut être reçue rapidement. Faire une requête similaire en Javascript demande d'envoyer la requête à un serveur lointain, et la réception des données peut être lourde et demander un gros débit internet dans certain cas, rendant l'application lente à répondre.

²⁷ Protocole de transmission permettant à l'utilisateur d'accéder à des pages web par l'intermédiaire d'un navigateur

Conclusion

Une bonne gestion de la base de données est donc un point crucial lors du développement d'un logiciel : une base de données mal structurée entraîne une succession de problèmes d'optimisation du logiciel exploitant et peut complexifier son architecture, et la correction de l'architecture de la base de données est souvent très coûteuse car elle demande souvent de réadapter aussi le logiciel, ce qui crée une charge de travail conséquente. De même, une exploitation mal optimisée de la base de données, bien que plus facile à corriger, peut entraîner de gros ralentissement et des temps de chargement trop long pour l'utilisateur.

Bien que l'année de DUT précédent m'ai appris à créer et exploiter les bases de données MySQL²⁸ avec SQL et PHP, les différentes missions chez ADN m'ont permis d'améliorer mes compétences dans le développement et l'exploitation des bases de données via les différents projets, le principal étant *Archéologie des Nécropoles*, un site web ayant justement pour but d'exploiter une base de données. Mais ayant travailler pour chaque projet sur toutes les parties (front-end²⁹ et back-end³⁰), j'ai pu m'améliorer sur de nombreux aspects, comme :

- L'utilisation d'un framework (dans ce cas FuelPHP);
- La (re)structuration de projet en partant d'une base déjà existante;
- L'utilisation de Javascript et ses bibliothèques (jQuery, Leaflet), que je ne connaissais que très mal avant cette alternance. Ces outils ont aussi été enseignés en cours en parallèle de l'alternance;
- La mise en page **responsive**³¹ à l'aide de CSS et Bootstrap : je maîtrisais déjà CSS, mais n'avais pas fait de projet pouvant s'adapter à différent format d'écran;

Je me suis aussi beaucoup amélioré sur les compétences organisationnelles en raison du contexte professionnel des missions. Ayant l'habitude des projets scolaires où les tâches à faire sont précisément définies, ou des projets personnels où les tâches à faire sont choisies selon mes envies et mon humeur, il m'a fallu pour l'alternance devenir capable de gérer et noter les différentes tâches à faire en allant demander les besoins si nécessaire. Contrairement au cas scolaire où toutes les tâches données sont à réaliser, il a fallu dans le cadre de l'alternance trier les tâches par priorité en fonction de son utilité et des besoins des utilisateurs. J'ai donc appris à utiliser **Trello**, une application collaborative (que j'ai utilisé seul) qui permet de gérer les tâches à l'aide de tableaux. Cette compétence organisationnelle s'est aussi développée dans ma vie personnelle, car en raison du rythme des cours bien chargés pour les alternants et des projets scolaires à faire sur le temps libre, J'ai dû

²⁸ Service de base de données

²⁹ Partie visible d'un projet informatique pour l'utilisateur, comprenant l'affichage des informations à afficher provenant du back-end, et la mise en page

³⁰ Partie logique cachée d'un projet informatique gérant les actions de l'utilisateur et les calculs

³¹ Capacité pour une application de s'adapter visuellement à différent format d'écran correctement

mieux m'organiser pour accomplir les projets scolaires et personnels pour me libérer du temps libre.

Cette année a donc été très enrichissante pour moi, principalement grâce à cette première expérience professionnelle qui, étant dans une association travaillant dans l'archéologie, avec une équipe de personnes passionnées, est à mon avis une expérience unique. Cette expérience m'a convaincu que le développement était bien ma voie. J'ai pu apprendre et améliorer mon savoir concernant les langages informatiques, et j'espère dans une prochaine alternance apprendre à mieux utiliser les frameworks et à travailler dans une équipe de développeurs, et sur plus long terme, devenir développeur à temps plein.

Webographie / Bibliographie / Citographie

Conception et architecture des bases de données 4e édition, Ramez Elmasri et Shamkant Navathe, 2007

Ionos.fr :

<https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/normalisation-bas-e-de-donnees/>

Rapport de stage de Virgile Louin :

<https://drive.google.com/file/d/1m5cilsuxsxLZv1jeCFLbLv34w8YFNpQ7/view?usp=sharing>

IRCF : <https://ircf.fr/actualites/optimiser-votre-base-de-donnees-mysql/>

SQL.sh : <https://sql.sh/optimisation>

Site d'ADN : <https://archeonec.hypotheses.org/>

DELL :

https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/fr/fr/top_10_tips_for_optimizing-final_fr.pdf

Glossaire

Ajax (*Asynchronous JavaScript And XML*) : Permet de faire des requêtes HTTP avec Javascript*.

Autocomplétion : Système proposant de finir un mot / une expression automatiquement, permet de gagner du temps sur l'écriture.

Back-end : Partie logique cachée d'un projet informatique gérant les actions de l'utilisateur et les calculs.

Bash (*Bourne Again Shell*) : Langage de programmation uniquement sur les systèmes Unix*, souvent utilisé pour faire des scripts touchant au système d'exploitation.

Bootstrap : Bibliothèque de classes CSS.

Clé primaire : Champ(s) d'une table* servant à l'indexation des données.

Complexité : Mesure permettant de connaître l'optimisation d'une requête ou d'un script

CSS (*Cascading Style Sheets*) : Langage permettant de changer la mise en forme d'un site web.

CSV (*Comma Separated Values*) : Format de tableau de données lisible au format textuel et par les éditeurs de tableaux excel.

Framework : Pack d'outils informatique facilitant la création d'un logiciel. Un framework peut être une simple bibliothèque d'outils comme il peut être beaucoup plus complet et avoir un éditeur propre à lui.

Front-end : Partie visible d'une projet informatique pour l'utilisateur, comprenant l'affichage des informations à afficher provenant du back-end, et la mise en page.

FuelPHP : Framework PHP facilitant la création d'un site en utilisant le modèle MVC*.

HTML (*Hypertext Markup Language*) : Langage de balisage utilisé pour structurer un site web.

HTTP (*HyperText Transfert Protocol*) : Protocole de transmission permettant à l'utilisateur d'accéder à des pages web par l'intermédiaire d'un navigateur

Javascript : Langage permettant de rendre dynamique des pages web sur le poste du client.

JQuery : Bibliothèque Javascript.

JSON (*JavaScript Object Notation*) : Format d'écriture de données standardisé, facilement exploitable depuis la plupart des logiciels et lisible pour l'humain.

MCD (*Modèle Conceptuel de Données*) : Schéma d'une base de données suivant des règles précises.

MVC (*Model View Controller*) : Méthode d'organisation d'un projet.

MySQL : Service de base de données

PCR (*Projet Collectif de Recherche*) : Forme d'organisation de la recherche archéologique

PHP (*Hypertext Preprocessor*) : Langage utilisé pour exécuter des instructions du côté du serveur.

Processeur : Composant d'ordinateurs interprétant les instructions.

Python : Langage de programmation interprété polyvalent.

QGIS : Application professionnelle géographique utilisée pour créer et éditer des cartes.

R : Langage de programmation mathématiques utilisé pour la réalisation de statistiques et de graphiques.

Responsive : Capacité pour une application de s'adapter visuellement à différent format d'écran correctement.

SQL (*Structured Query Language*) : Langage permettant d'exécuter des instructions sur une base de données.

Table : Dans une base de données, correspond à un tableau de valeur.

UNIX : Système d'exploitation.

URL (*Uniform Resource Locator*) : Adresse permettant d'accéder à un site web

VR (*Virtual Reality*) : Traduit "Réalité Virtuelle" en français, concerne les équipements, capteurs et logiciels permettant de s'immerger dans un univers fictif, transmettant les mouvements dans le monde dans cet univers virtuel.

Table des annexes

Exemple du code de Virgile Louin	26
CMD de la base de données avant les modifications	27
CMD final de la base de données	27
Formulaire de création d'une opération	28
Formulaire de création d'un sujet handicapé	29
Schéma de navigation final	30
Page de résultat de recherche	30
Carte des sites de fouilles	31
Schéma base de données de la carte des sites	31

1. Exemple du code de Virgile Louin

```
if(Input::post('attelle') != NULL):
foreach(Input::post('attelle') as $key => $val):
foreach ($val as $key2 => $val2) {
$valeurs.='&attelle_'.$key.'_'.$key2;
$appareil[$key][$key2]+=array('attelle'=>1);
}
endforeach;
endif;

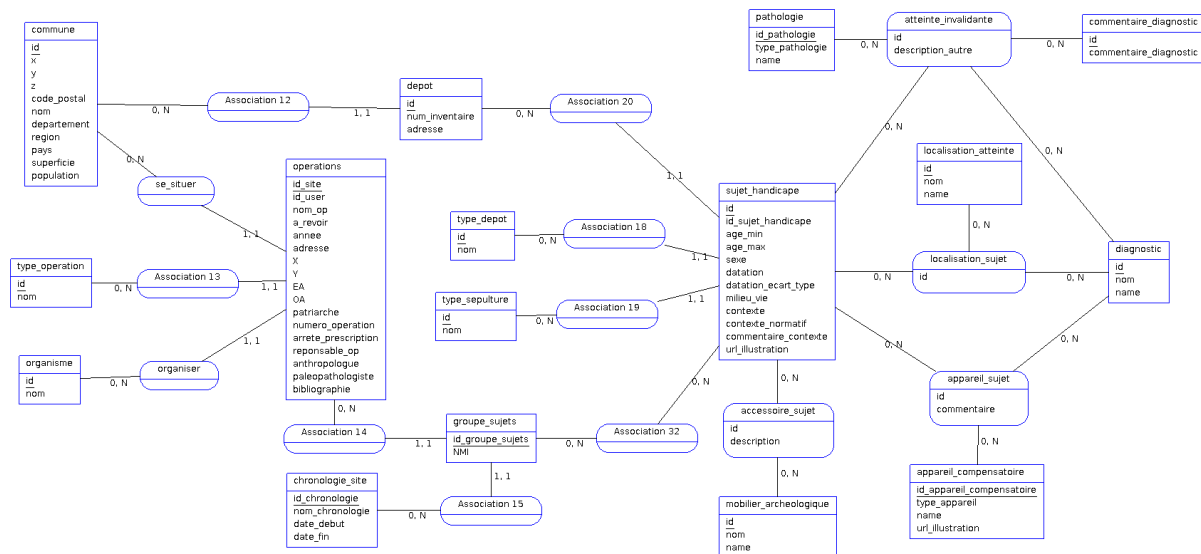
if(Input::post('prothese') != NULL):
foreach(Input::post('prothese') as $key => $val):
foreach ($val as $key2 => $val2) {
$valeurs.='&prothese_'.$key.'_'.$key2;
$appareil[$key][$key2]+=array('prothese'=>2);
}
endforeach;
endif;

if(Input::post('orthese') != NULL):
foreach(Input::post('orthese') as $key => $val):
foreach ($val as $key2 => $val2) {
$valeurs.='&orthese_'.$key.'_'.$key2;
$appareil[$key][$key2]+=array('orthese'=>3);
}
endforeach;
endif;

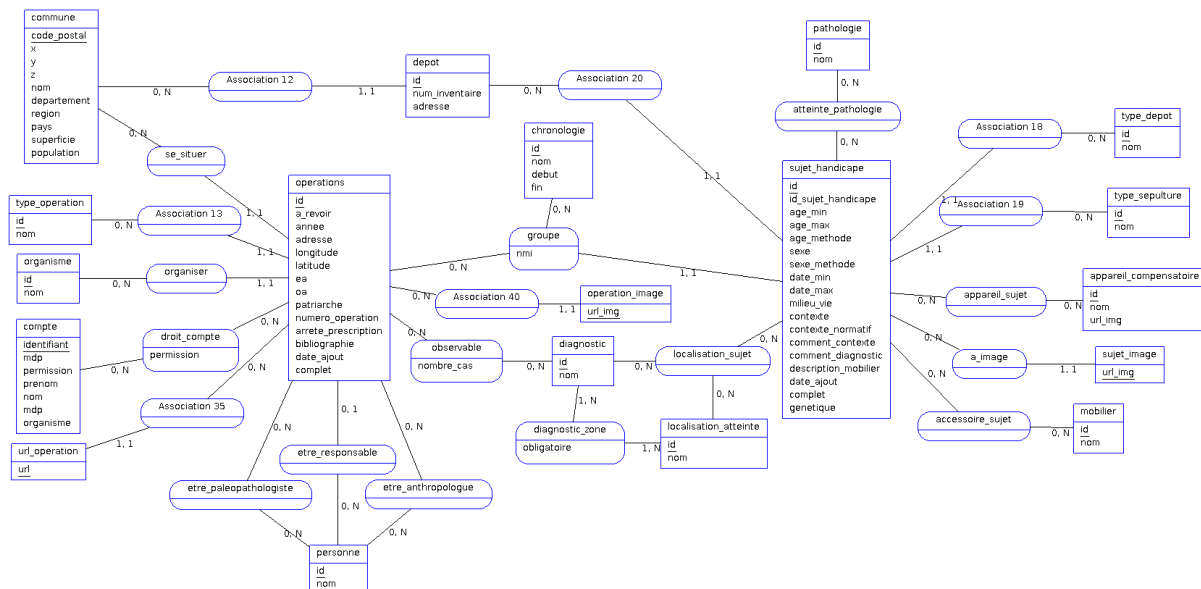
if(Input::post('bequillage') != NULL):
foreach(Input::post('bequillage') as $key => $val):
foreach ($val as $key2 => $val2) {
$valeurs.='&bequillage_'.$key.'_'.$key2;
$appareil[$key][$key2]+=array('bequillage'=>4);
}
endforeach;
endif;

if(Input::post('commentaire_appareil') !=NULL):
foreach(Input::post('commentaire_appareil') as $key => $val):
$val=trim(strip_tags($val));
$valeurs.='&commentaire_appareil_'.$key.'='.$val;
$commentaire_appareil[$key]+=array('commentaire_appareil'=> $val);
endforeach;
endif;
```

2. CMD de la base de données avant les modifications



3. CMD final de la base de données



4. Formulaire de création d'une opération

Opération

Sélectionner une position sur la carte pour récupérer les coordonnées.

Conversion coordonnées GPS

Longitude
44,6484375

Latitude
30,9323874351851

Année de l'opération

Organisme
Indéterminé

Numéro de l'opération

[Informations sur les champs EA et OA](#)

EA

OA

Type d'opération
Indéterminé

Patriarche

Arrêté de prescription

Responsable de l'opération

Anthropologue

+

Paleopathologiste

+

Bibliographie ([recommandations sur le format de GALLIA](#))

URL

+

Comptes autorisés

Nom du compte

+

Iconographie

Aller sur Nakala

Aide

Lien URL de l'image

+

☐ Les informations de l'opération sont complètes. (vous pourrez toujours modifier l'opération)

Confirmer

Alex BAIET

Page 27

5. Formulaire de création d'un sujet handicapé

Groupe du sujet

NMI

Chronologie Indéterminée

Sujet handicapé

Identifiant du sujet*

Datation minimale

Datation maximale

Sexe Indéterminé

Méthode de détermination du sexe

Âge minimum au décès

Âge maximum au décès

Méthode de détermination de l'âge

Type de dépôt Indéterminé

Type de sépulture Indéterminé

Milieu de vie Indéterminé

Contexte de la tombe Indéterminé

Contexte de la tombe Indéterminé

Commentaire

Atteinte invalidante

D : Partie droite, G : Partie gauche

	D	G	D	G	Nb cas concerné	Nb cas observable	Prévalence
<input type="checkbox"/> Trépanation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Édentement complet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Atteinte neurale	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Scoliose sévère	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Maladie de Paget ou ostéite déformante	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> DISH	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Rachitisme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Nanisme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Fracture non réduite	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Amputation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Pathologie dégénérative sévère	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
<input type="checkbox"/> Autre	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0

☐ Lèpre
 ☐ Syphilis
 ☐ Variole
 ☐ Tuberculose
 ☐ Peste
 ☐ Autre pathologie infectieuse

*La prévalence est ici une estimation minimale de la prévalence réelle.

Commentaire du diagnostic

☐ Données génétiques

Dépôt

Numéro de dépôt

Commune

Adresse du dépôt

Accessoire

☐ Accessoire vestimentaire et parure
 ☐ Armement, objet de prestige
 ☐ Dépôt de récipient
 ☐ Autre mobilier
 ☐ Mobilier atypique

Description du mobilier

Appareil compensatoire

☐ Appareillage de transport
 ☐ Prothèse
 ☐ Orthèse
 ☐ Béquillage

Iconographie

Aller sur Nakala

Aide ?

Lien URL de l'image

+

☐ Les informations du sujet sont complètes. (vous pourrez toujours modifier le sujet)

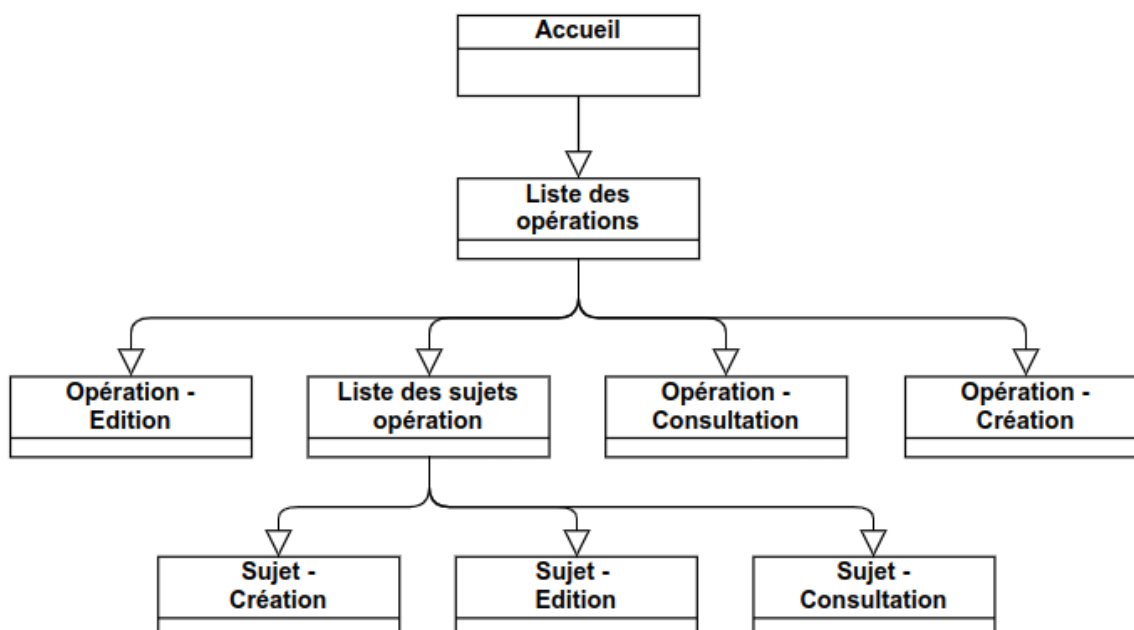
Confirmer et sortir

Confirmer et dupliquer la fiche

Alex BAIET

Page 28

6. Schéma de navigation final



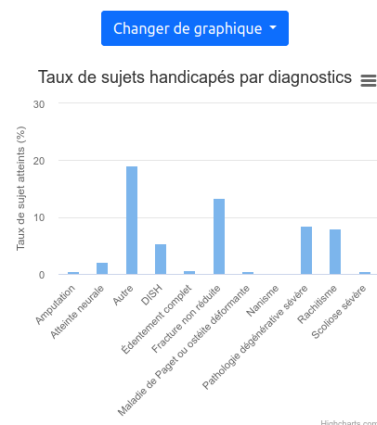
7. Page de résultat de recherche

Résultats de la recherche [Exporter en CSV](#)

146 opérations et 425 sujets handicapés correspondent à votre recherche.

>	⚠	2685	Mikael Rouzic	Saint-Pierre-en-Faucigny, Les Molettes	2021	Consulter
>	⚠	2684	Mikael Rouzic	Annecy, 39 Avenue des Romains	2016	Consulter
>	⚠	2683	Mikael Rouzic	Annecy, 41 Avenue des Romains	2014	Consulter
✓	✓	2676	Nathalie Ameye	Saint-André-les-Vergers, ZAC d'Echenilly	2010	Consulter

Sujets handicapés de l'opération									
État	Id	Nom	Sexe	Datation	Milieu de vie	Type de dépôt	Type de sépulture	Âge	Actions
⚠	65882	127	Homme	-50 - 500	inconnu	Primaire	Individuelle	30 - 99	Consulter
⚠	65881	133	Indéterminé	-50 - 500	inconnu	Primaire	Individuelle	20 - 99	Consulter
⚠	65880	139	Indéterminé	-50 - 500	inconnu	Primaire	Individuelle	20 - 99	Consulter
⚠	65879	154	Homme	-50 - 500	inconnu	Primaire	Individuelle	20 - 99	Consulter
>	✓	2675	Nathalie Ameye	Buchères, Le Clos				2009	Consulter
>	✓	2674	Nathalie Ameye	Le Clos, Le Clos de l'Église				2008	Consulter



8. Carte des sites de fouilles



9. Schéma base de données de la carte des sites

