

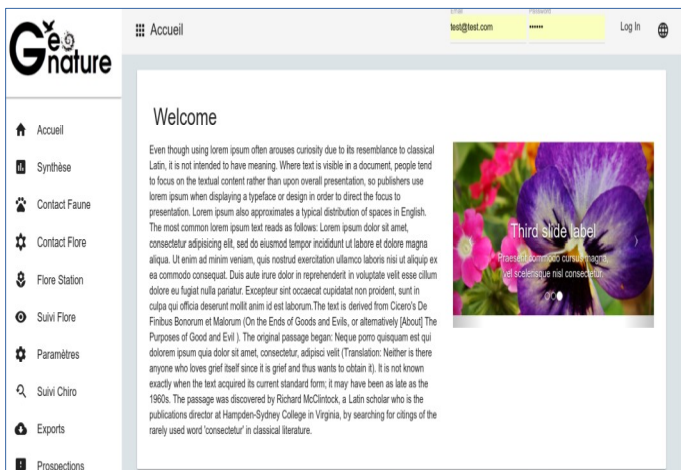
Université Grenoble-Alpes

Master 2 DCISS



Rapport de stage

Refonte de GeoNature pour moderniser et modulariser les applications web



Rapport présenté par :
Quang PHAM
Master 2 DCISS

Maître de stage :
Camille Monchicourt
Parc National des Écrins
Tuteur pédagogique :
Jean-Michel Adam
Responsable du Master DCISS

Remerciements

J'aimerais remercier chaleureusement mon maître de stage Camille Monchicourt, qui m'a aidé à me rendre au parc tous les jours, qui m'a donné la motivation pour aller de l'avant, à Gil Deluermoz, qui m'a transmis beaucoup de connaissances techniques et professionnelles. Merci d'avoir fait confiance à un « novice » en développement web pour commencer un projet important comme GeoNature.

Je remercie également l'ensemble des collègues du PNE pour votre accueil chaleureux pour un environnement amical, et sympathique !

Merci à mes examinateurs Jean-Michel Adam et Jérôme David à l'équipe pédagogique du master, Je suis vraiment impressionné par votre attention et votre aide pendant mes études.

Table des matières

Remerciements.....	2
I.Contexte.....	5
1.Présentation de la structure d'accueil.....	5
2.L'équipe de travail.....	6
3.Les missions du SI.....	6
II.Objectif général, Geonature, Taxhub.....	8
1.Objectif général.....	8
2.Contexte.....	8
3.Besions.....	8
III.Les applications développées par le Parc national des Ecrins.....	9
1.UsersHub.....	9
2.TaxHub.....	10
3.GeoNature-atlas.....	11
4.GeoSites.....	11
5.GeoNature.....	12
IV.Mon stage sur TaxHub.....	13
V.Mon stage sur GeoNature v2.....	13
1.La mission de stage.....	13
2.Inconvénient des technologies utilisées dans GeoNature 1.....	13
3.Renforcer le projet avec Python Flask et Angular 4.....	13
4.Structure du code GeoNature 2.....	15
5.Base de donnée.....	16
5.1Organisation de la base de données.....	16
5.2Modèle de données.....	17
6.Fontend.....	18
6.1Organisation frontend.....	18
6.2Frontend Core – Modules.....	19
6.3Frontend Interface.....	20
6.3.1Menu Bar - SideNav-Bar.....	21
6.3.2Accueil.....	22
6.3.3Module Contact.....	23
6.3.4Composant map.....	24
6.3.5Composant form contact.....	25
6.3.6Composant form contact avec les options.....	26
6.3.7Les technos frontend.....	27
7.Backend.....	28
7.1Organisation backend.....	28
7.2Backend API REST conception.....	29
7.3Les technologies backend.....	30
8.Dev-ops.....	31
8.1Docker.....	31
8.2Github.....	31
8.3Automatisation avec script SH.....	31
9.Toutes les technologies.....	32
VI.Travail collectif.....	33
1.Agile avec Github.....	33
2.Workshop de développement (Briançon – 21 aout 2017).....	35
VII.Calendrier de travail.....	36
1.Taxhub.....	36
2.GeoNature Frontend.....	36

3.GeoNature Backend.....	36
4.Docker.....	37
5.Workshop – Rapport – Transfert.....	37
VIII.Futur de GeoNature.....	38
1.Ionic.....	38
2.Progressive web apps.....	38
3.Docker.....	38
IX.Les difficultés.....	40
1.Difficulté de la choix de technologies.....	40
2.Difficulté de la discussion en groupe.....	40
X.La conclusion.....	41

I. Contexte

1. Présentation de la structure d'accueil

Créé en mars 1973, le Parc national des Écrins est un établissement public à caractère administratif placé sous la tutelle du ministère de l'Ecologie, du Développement durable et de l'Energie.

Ses personnels développent des missions de connaissance, de préservation des espèces et des milieux, d'accueil du public et d'accompagnement du développement du territoire.

Avec une petite centaine de salariés, le Parc national des Écrins est organisé autour d'équipes de terrain qui animent les sept secteurs géographiques du territoire.

Le Parc national des Écrins est un établissement public qui a pour première vocation la préservation de la biodiversité et la mise en place d'une politique de développement durable sur



Illustration 1: Carte de localisation du PNE et de ses secteurs

son territoire. Ce territoire de haute-montagne s'étend sur 53 communes entre les départements des Hautes-Alpes - région PACA - et de l'Isère - région Rhône-Alpes-Auvergne.

L'équipe permanente du PNE est constituée d'environ 100 personnes réparties entre le siège à Gap, et sept secteurs qui couvrent l'ensemble du territoire du parc (figure 1). Le siège est composé de quatre services : le service scientifique, le service aménagement, le service communication et le service général.

2. L'équipe de travail

Mon stage se déroule au sein du service scientifique du PNE. Ce service est lui même divisé en deux pôles:

- Le « pôle connaissance » qui travaille sur la mise en place de protocoles de suivi scientifique (faune, flore et mesures physiques)
- Le pôle système d'informations, dans lequel je me trouve, qui s'occupe de la géomatique et de l'informatique. Il est constitué d'un chargé de mission base de données et développement web, d'un chargé de mission administration réseau, téléphonie et informatique, et d'un géomaticien, chef du pôle.

3. Les missions du SI

Le pôle SI occupe une position transversale puisqu'il est amené à travailler avec tous les services du Parc. Il assiste aussi bien le service scientifique dans la mise en place de protocoles de suivi faune-flore, que le service aménagement dans le suivi du patrimoine bâti et de l'agriculture, ou encore le service communication dans le mise en place d'outils de mise en valeur des sentiers de randonnées et l'animation du site web.

De part ses missions de protection de la faune et de la flore, le parc national est amené à collecter des quantités importantes de données spatialisées. Le rôle du SI au sein du parc est donc d'organiser et de faciliter la collecte de ces données, de les gérer mais également de créer des outils pour les analyser. Une grande composante métier du SI tient donc dans l'administration de bases de données.

Le PNE a été novateur dans la mise en place de la collecte et le stockage des données sur informatique et possède aujourd'hui une architecture de base de données et des outils structurés. Le schéma ci-dessous (figure 2) résume la modernisation de la stratégie générale du SI, du recueil de la donnée jusqu'à son traitement et sa consultation. En fonction des protocoles et des besoins des agents, plusieurs chaînes de travail ont été mises en place. La collecte sur le terrain, anciennement effectuée sur papier (en blanc sur le schéma) est aujourd'hui saisie sur des outils nomades (applications mobiles sur tablette). Pour les protocoles importants, l'ensemble de ces données sont ensuite centralisées dans des bases de données PostgreSQL (avec extension PostGIS pour les besoins spatiaux). A partir de ces bases de données, des applications de consultation métier ou grand public sont développées soit par des prestataires extérieurs, soit directement par le SI.

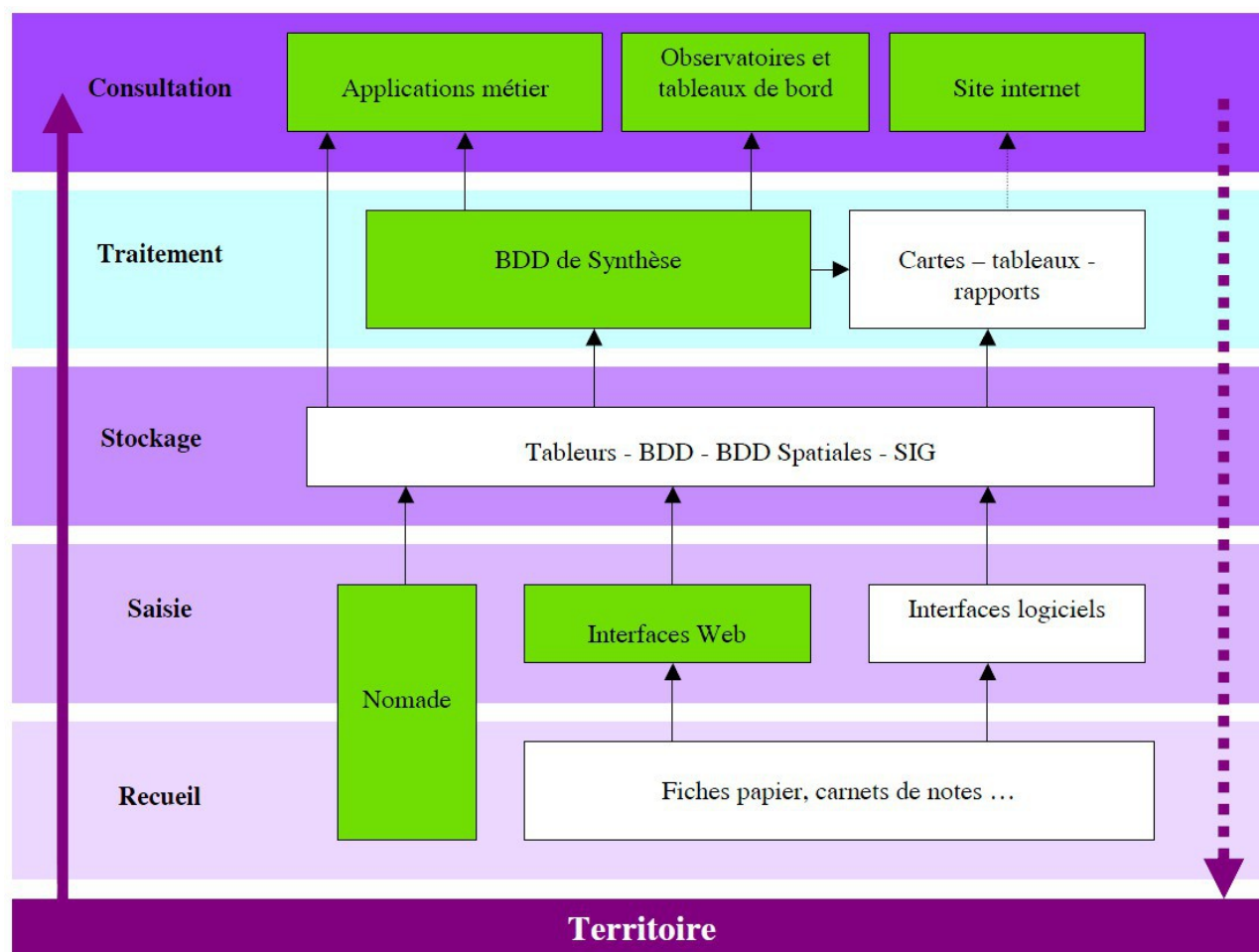


Illustration 2: La chaîne de travail du SI : du recueil à la consultation des données (document interne). On distingue en vert la chaîne de travail actuelle, et en blanc l'ancienne.

II . Objectif général, Geonature, Taxhub

1. Objectif général

Refondre les applications utilisées par les parcs nationaux (ObsOcc et GeoNature) pour en faire un outil open source plus modulaire, plus moderne, plus ouvert, qui intègre de nouvelles fonctionnalités et permet à chaque structure d'y intégrer des modules pour ses protocoles spécifiques

2. Contexte

Actuellement les parcs nationaux utilisent les outils ObsOcc (aka SICEN) et GeoNature (+ KaruNati au PNG et PNRun) pour la gestion de leurs protocoles et données Faune et Flore. Ces 2 outils sont développés en interne depuis 2010 et deviennent obsolètes techniquement (technologies vieillissantes, non maintenues, présentant donc des défauts de compatibilité, d'ergonomie et de performances). Ils sont partiellement redondants et ont été développés dans des contextes spécifiques. Le premier n'est pas compatible avec la stratégie scientifique inter-parc, ni la stratégie commune SI qui est en train d'être construite. Le second n'est pas assez modulaire pour permettre à chacun de l'adapter à ses besoins et protocoles. Par ailleurs les protocoles, le contexte et les besoins des structures ont évolué depuis la conception de ces outils.

3. Besoins

- Répondre aux besoins de connaissance et de gestion locaux en matière de faune et de flore
- Faciliter les échanges de données locaux et nationaux (avec instances et partenaires)
- Disposer d'une chaîne de travail et d'outils performants, modernes, modulaires et complets
- Disposer d'une organisation des données souple et solide, ouverte vers les partenaires, structurée à partir d'un socle et de référentiels communs tout en facilitant à chaque structure la possibilité d'y intégrer ses propres protocoles
- Agglomérer, valoriser et diffuser la connaissance naturaliste des parcs nationaux.

III. Les applications développées par le Parc national des Ecrins

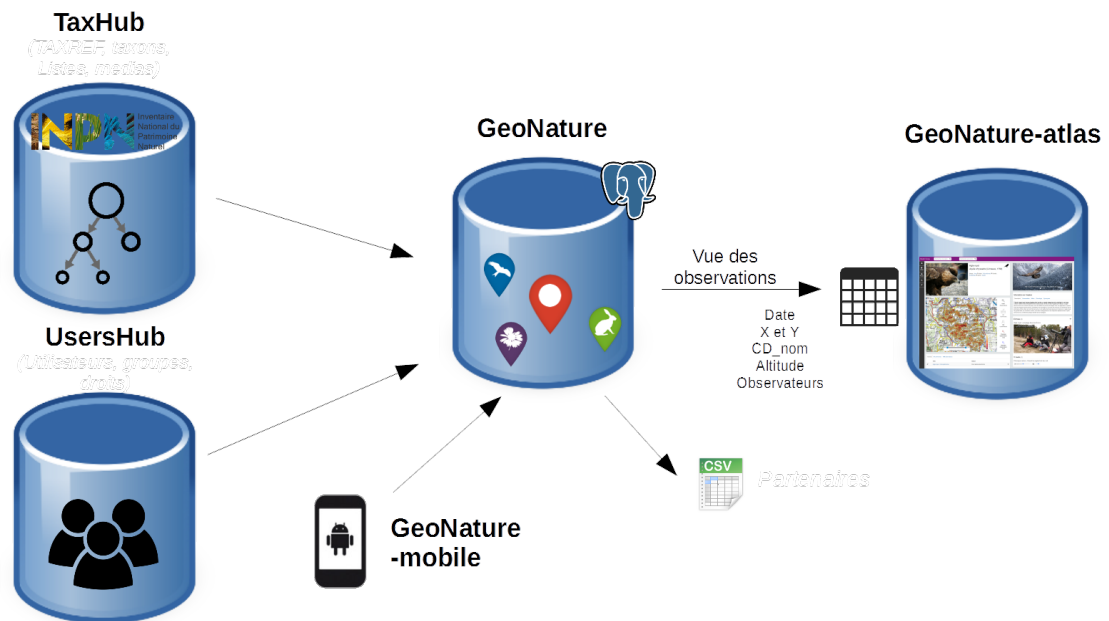


Illustration 3: Les projets existants – Parc national des Ecrins

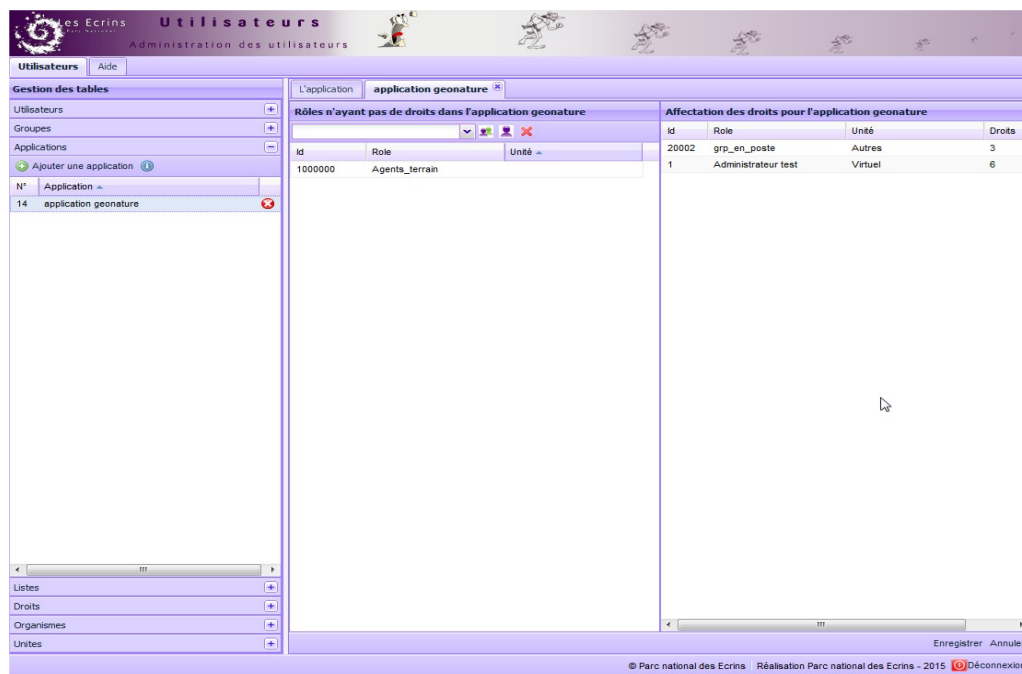


Illustration 4: UsersHub - Parc national des Ecrins

1. UsersHub

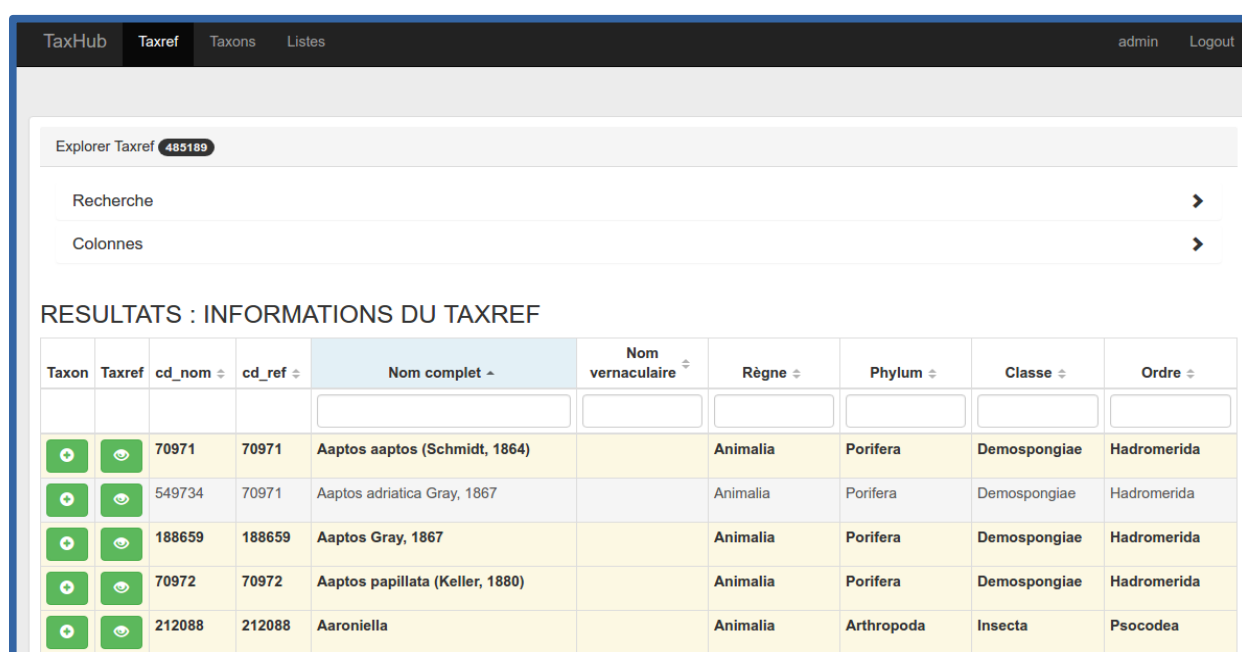
Application web de gestion centralisée des utilisateurs.

<https://github.com/PnEcrins/UsersHub>

UsersHub est une application web permettant de regrouper l'ensemble des utilisateurs d'applications web afin de gérer de manière différenciée et centralisée les droits d'accès à ces applications ainsi que le contenu des listes déroulantes d'observateurs.

Elle permet de gérer de manière centralisée des utilisateurs et de les placer dans des groupes ; de créer différents niveaux de droits et de les affecter aux utilisateurs et/ou aux groupes d'utilisateurs pour chacune de vos applications. Elle permet également de gérer des organismes, des unités et des listes déroulantes regroupant des utilisateurs ou des groupes d'utilisateurs.

2. TaxHub



The screenshot shows the TaxHub web application interface. At the top, there is a navigation bar with 'TaxHub', 'Taxref', 'Taxons', and 'Listes'. On the right, there are links for 'admin' and 'Logout'. Below the navigation bar, there is a search bar labeled 'Recherche' and a button 'Colonnes'. The main content area is titled 'RESULTATS : INFORMATIONS DU TAXREF'. It displays a table with columns: Taxon, Taxref, cd_nom, cd_ref, Nom complet, Nom vernaculaire, Règne, Phylum, Classe, and Ordre. The table contains five rows of data, each with a green plus icon in the first column.

Taxon	Taxref	cd_nom	cd_ref	Nom complet	Nom vernaculaire	Règne	Phylum	Classe	Ordre
		70971	70971	Aptos aaptos (Schmidt, 1864)		Animalia	Porifera	Demospongiae	Hadromerida
		549734	70971	Aptos adriatica Gray, 1867		Animalia	Porifera	Demospongiae	Hadromerida
		188659	188659	Aptos Gray, 1867		Animalia	Porifera	Demospongiae	Hadromerida
		70972	70972	Aptos papillata (Keller, 1880)		Animalia	Porifera	Demospongiae	Hadromerida
		212088	212088	Aaroniella		Animalia	Arthropoda	Insecta	Psocodea

Illustration 5: Taxhub - Parc national des Ecrins

Application web de gestion centralisée des taxons basée sur le référentiel TAXREF (<http://inpn.mnhn.fr/programme/referentiel-taxonomique-taxref>) du MNHN.

Elle permet de gérer la liste des taxons présents dans chaque structure, d'y greffer des informations spécifiques, de définir des listes de taxons et des filtres en fonction des besoins.

Elle permet aussi de gérer les descriptions et les médias des taxons pour leur affichage sur GeoNature-atlas.

<https://github.com/PnX-SI/TaxHub>

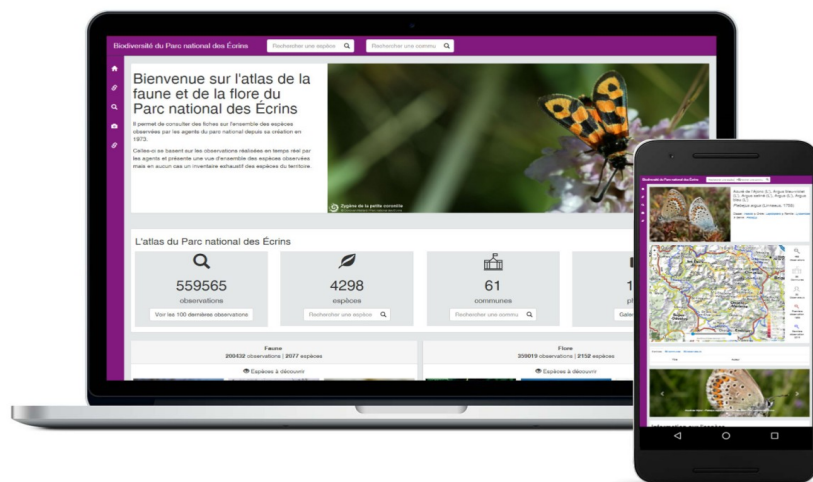


Illustration 6: GeoNature -atlas - Parc national des Ecrins

3. GeoNature-atlas

Atlas WEB dynamique Faune-Flore basé sur les données présentes dans la synthèse de GeoNature.

<https://github.com/PnEcrins/GeoNature-atlas>

4. GeoSites

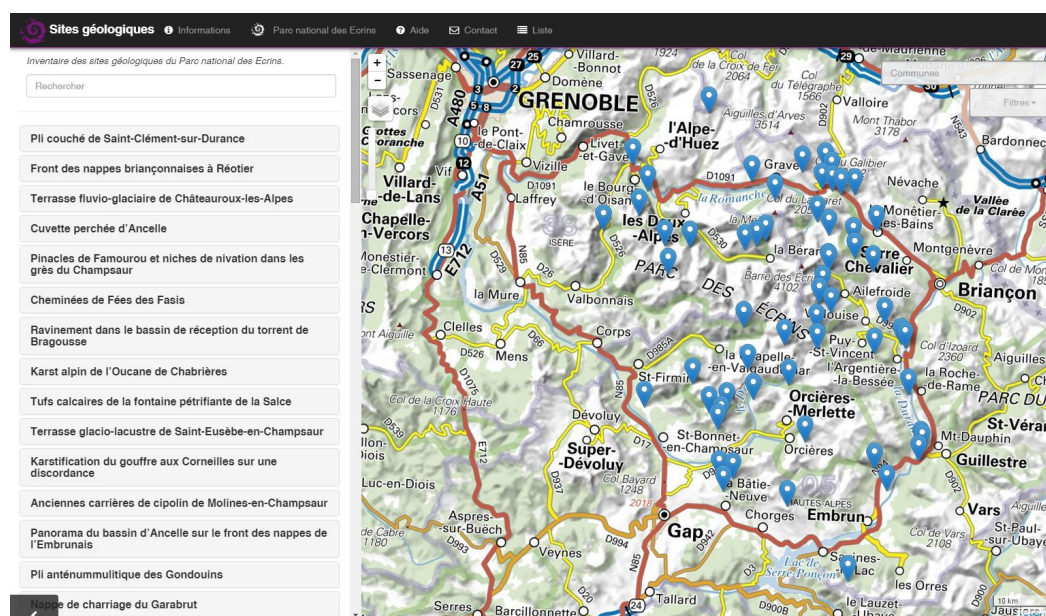


Illustration 7: GeoSites - Parc national des Ecrins

Application WEB permettant de consulter l'inventaire du patrimoine géologique.

GeoSites permet de mettre en ligne une application WEB grand public pour consulter l'inventaire du patrimoine géologique d'un territoire. Cet inventaire est coordonné au niveau national par la CPPG (Commission permanente du patrimoine géologique) et validé scientifiquement par le MNHN (Muséum national d'Histoire naturelle) et le BRGM (Bureau de Recherches Géologiques et Minières). <https://github.com/PnEcrins/GeoSites>

5. GeoNature

- <https://github.com/PnX-SI/GeoNature>
- GeoNature version 1.0 est un ensemble d'applications web et mobile utilisé par les parcs nationaux pour la gestion de protocoles et données Faune et Flore. Il est développé et maintenu par le collectif de développeurs du réseau des Parcs Nationaux depuis 2012
- GeoNature est aussi en train d'être déployé pour être utilisé comme l'outil national de saisie des données des études d'impact dans le cadre de l'article L 411-1 A de la loi pour la reconquête de la biodiversité, de la nature et des paysages n° 2016-1087 du 8 août 2016.
- GeoNature version 2 : Refondre GeoNature 1, pour en faire un outil open source plus modulaire, plus moderne, plus ouvert, qui intègre de nouvelles fonctionnalités et permet à chaque structure d'y intégrer des modules pour ses protocoles spécifiques.
- Cahier de charge V2 <http://geonature.fr/documents/2017-04-GN2-Fonctionnalites-0.1.pdf>



Illustration 8: GeoNature V1 et GeoNature V2 - page accueil

IV. Mon stage sur TaxHub

Rapport de mi-stage : <http://geonature.fr/documents/2017-05-QuangPham-Rapport-mi-stage.pdf>

V. Mon stage sur GeoNature v2

1. La mission de stage

Mes tâches:

- Tester les technologies.
- Initialiser l'environnement pour le projet GeoNature 2.

Avec la rôle de stagiaire, mes missions ont été la découverte des technologies et l'initialisation de l'environnement stable de développement pour un projet de développement qui aura une durée 3 ans minimum.

C'est une mission assez large mais pour ma part c'est la chance de découvrir tous les côté.aspects d'un projet qui démarre.

Je me suis fixé comme objectif initial d'acquérir le maximum connaissances sur les différents de la base de données, du backend ainsi que DevOps, Ux/Ui, API, tests, scripts SH de déploiement automatique, modularisation de l'application...

2. Inconvénient des technologies utilisées dans GeoNature 1

PHP symfony 1.4: vieille version d'un framework complexe, peu modulaire et impossible de le faire fonctionner sur php 7 dans sa version 1.4.

Extjs : ce framework javascript n'est quasiment plus utilisé ; les versions utilisées ne sont plus du tout maintenu et des incompatibilités avec l'implémentation javascript des navigateurs modernes sont déjà survenues (corrigées) mais pourrait devenir bloquantes. Extjs n'est plus réellement open source, son interface est vieillissante. Et surtout, c'est le framework js qui prend en charge totalement le DOM (c'est le JS qui construit tout le HTML). Il est donc difficile de personnaliser l'interface.

3. Renforcer le projet avec Python Flask et Angular 4

Python Flask: un micro-framework, il est facile d'apprendre à l'utiliser, c'est un petit framework, qui de plus consomme moins de ressources et impose souvent moins de contraintes et le moins de lignes de code possible.

La framework Flask a été utilisé dans le projet TaxHub et GeoNature Atlas. Dans GeoNature 2 Flask remplacera Symfony.

Angular 4: Mon stage a donc été l'occasion pour le parc national de faire de nouveaux choix en terme de technologie, en accord les SI des autres parcs nationaux.

On a discuté et fait plusieurs test pour choisir entre Angular 4 et VueJs.

VueJs est un framework intéressant qui monte fortement, il est très adapté à des projets léger ou pour des développeurs moins expérimentés.

Angular 4 est un framework récent, il est sorti en décembre 2016. Il fait suite à AngularJS et Angular 2. Il est soutenu par Google. C'est un framework lourd mais adapté à la taille du projet, avec une forte rigueur d'organisation de l'architecture du code, notamment pour obtenir un découpage pertinent et fonctionnel en modules suffisamment autonomes.

Finalement, l'équipe inter-parc a choisi Angular 4 parce qu'il est plus tôt adapté à un grand projet comme GeoNature 2.

	Langages	Bdd	Back-end	Front-end	Carto
GeoNature v1	PHP, Html, Javascript, Css	PostgreSQL, PostGIS	PHP : Symfony	Extjs	GeoExtjs, Mapfish Openlayers 2
GeoNature v2	Python, TypeScript, Html, Scss	PostgreSQL, PostGIS	Python : Flask	Angular 4 Bootstrap, Material Angular	Leaflet, Leaflet-Draw

Les technologies de GeoNature v1 vs GeoNature v2

4. Structure du code GeoNature 2

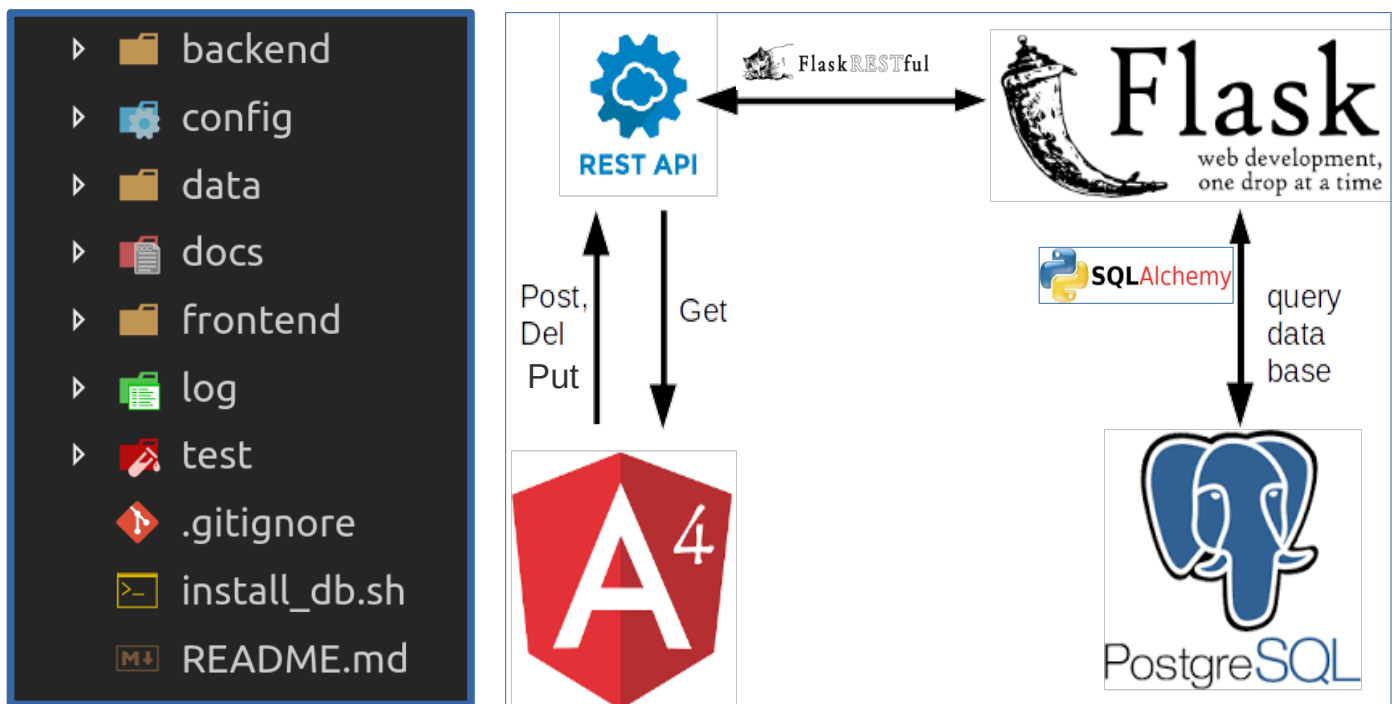


Illustration 9: Structure de GeoNature 2

En accord avec l'équipe de développement du Parc national des Ecrins et du Parc national des Cévennes, j'ai moi-même mis en place le code de la version 2 de GeoNature.

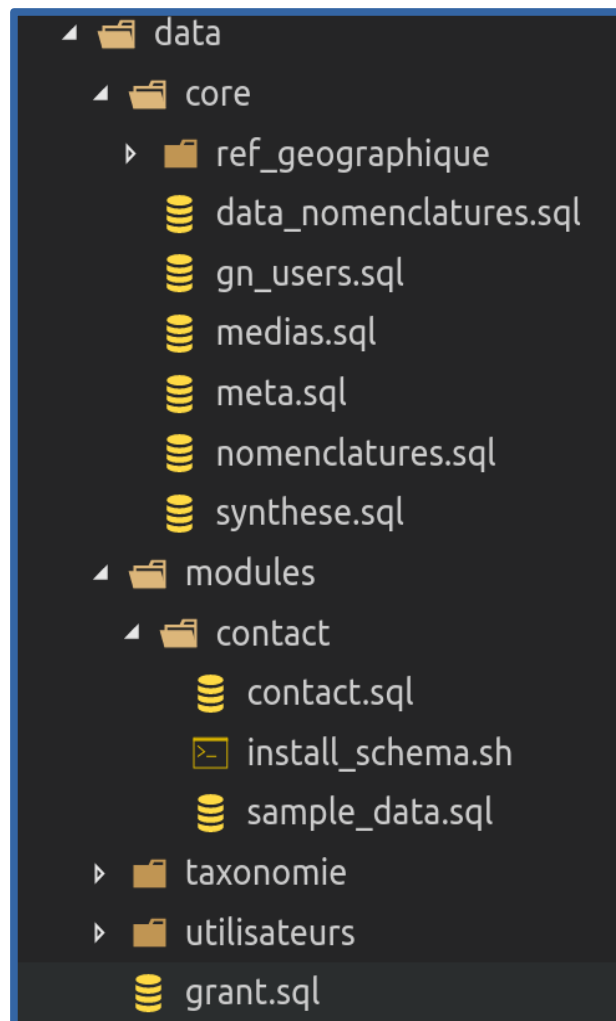
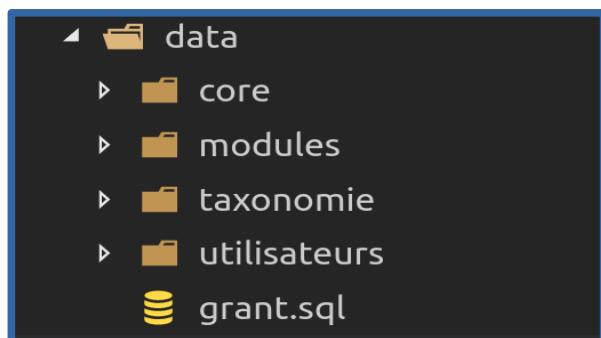
Organisation des répertoires : dans l'image de gauche, nous pouvons voir comment est organisée l'application avec les répertoires backend, frontend, data (base de données). Le projet contient des répertoires pour les tests, la documentation et la configuration globale de l'application.

Structure de l'application : le schéma de droite simplifie le processus interactif entre les éléments de l'application. Il présente aussi les technologies principales, même si le processus de travail contient beaucoup d'autres technologies auxiliaires.

L'interaction entre le backend et la base de données sont des interactions entre Flask et PostgreSQL via SQLAlchemy, l'interaction entre le frontend et le backend est l'interaction entre Angular et Flask via RestAPI.

5. Base de donnée

5.1 Organisation de la base de données



Organisation de la base de données

J'ai participé à l'important travail de redéfinition de la base de données de GeoNature.

A gauche on peut voir l'organisation des fichiers SQL qui permettent de créer la base de données Taxonomie (la base de TaxHub), Utilisateurs (la base de UsersHub). Il existe 2 applications supplémentaires (TaxHub, UserHub), qui fournissent des données de références pour GeoNature.

A droite - Core + Modules :

- Dans le répertoire core on retrouve des éléments de la base de données communs à tous les modules.
- Dans la répertoire module se trouvent de petits modules, qui correspondent à des protocoles distincts tels que Contact, Suivi chiro, Suivi Flore ...

Le core et ses dépendances sont installés via le script d'installation `install_db.sh` à la racine de l'application, les modules sont installés via `install_schema.sh` présents dans chaque module. Nous le faisons pour assurer la modularisation, l'indépendance et la réutilisation de chaque module à l'avenir.

5.2 Modèle de données

Pour ma part, je ne travaille pas directement sur la conception du MCD (Modèle Conceptuel des Données) de la base de données PostgreSQL mais interagit avec celle-ci dans les développements.

C'est Gil Deluermoz du Parc national des Ecrins, un expert sur PostgreSQL et MCD qui a suivi et développé le projet depuis les tous premiers jours, est le concepteur principal. Avec lui travaillent aussi mon maître de stage Camille Monchicourt et des agents des autres parcs nationaux.

Le développement de MCD est une tâche plus importante dans le processus de développement. C'est la première base, selon laquelle l'application sera développée. Avec un bon MCD, on simplifie le processus de développement du backend et du frontend.

J'ai ainsi participé et suivi le processus de conception MCD, l'ai déployé avec PostgreSQL et j'ai acquis beaucoup de connaissances, sur le fonctionnement, la création et la logique de la base de données.

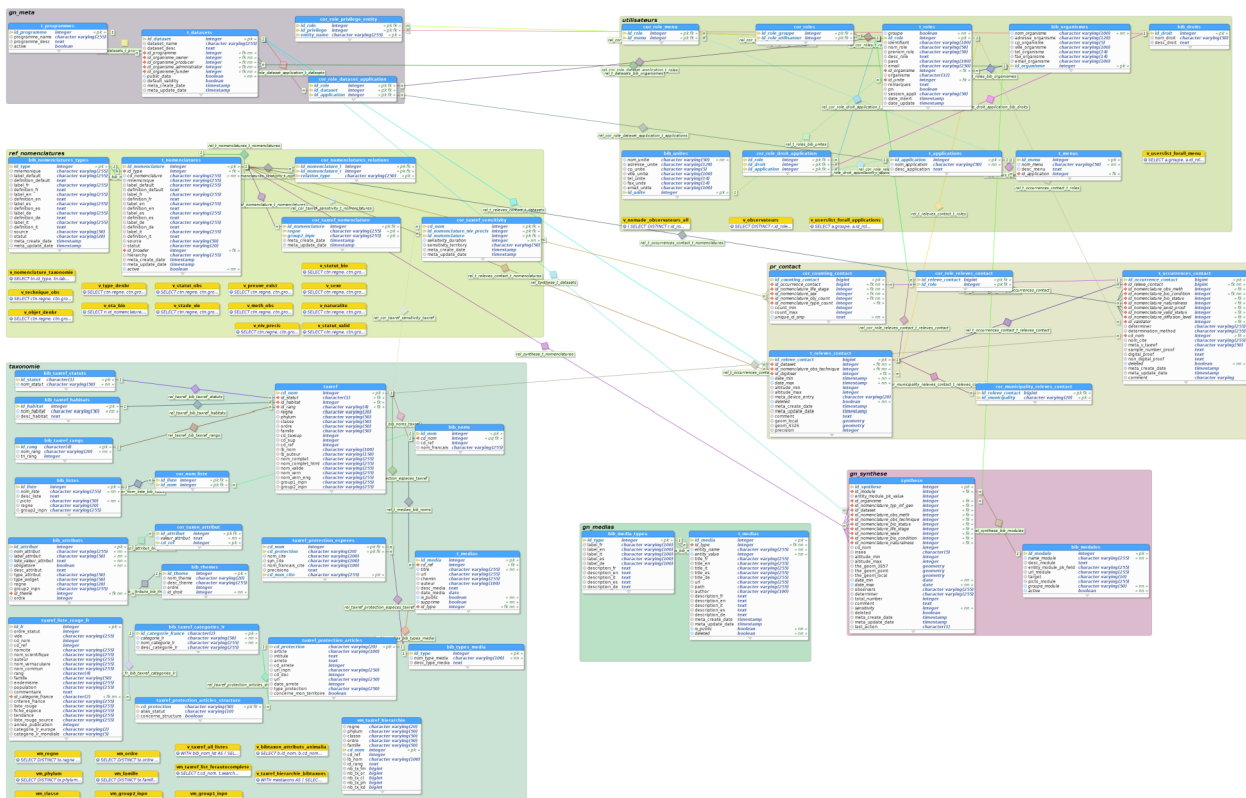


Illustration 10: Modèle Conceptuel des Données

6. Frontend

La conception et la mise en place du frontend est la partie que j'ai menée entièrement. C'est un travail où j'ai donc investi le plus de temps pour apprendre et approfondir. Les technologies du backend et de la base de données, ont été choisies à partir des projets précédents, mais pour la partie frontend, j'avais tout à réalisé avec des technologies nouvelles.

Angular 4 est un framework jeune, donc il existe également une limitation lorsqu'il s'agit de l'utiliser avec d'autres bibliothèques ou frameworks. Le test de la compatibilité d'Angular avec les bibliothèques, les frameworks ainsi que la création de démonstration a été complété après 2 mois.

6.1 Organisation frontend

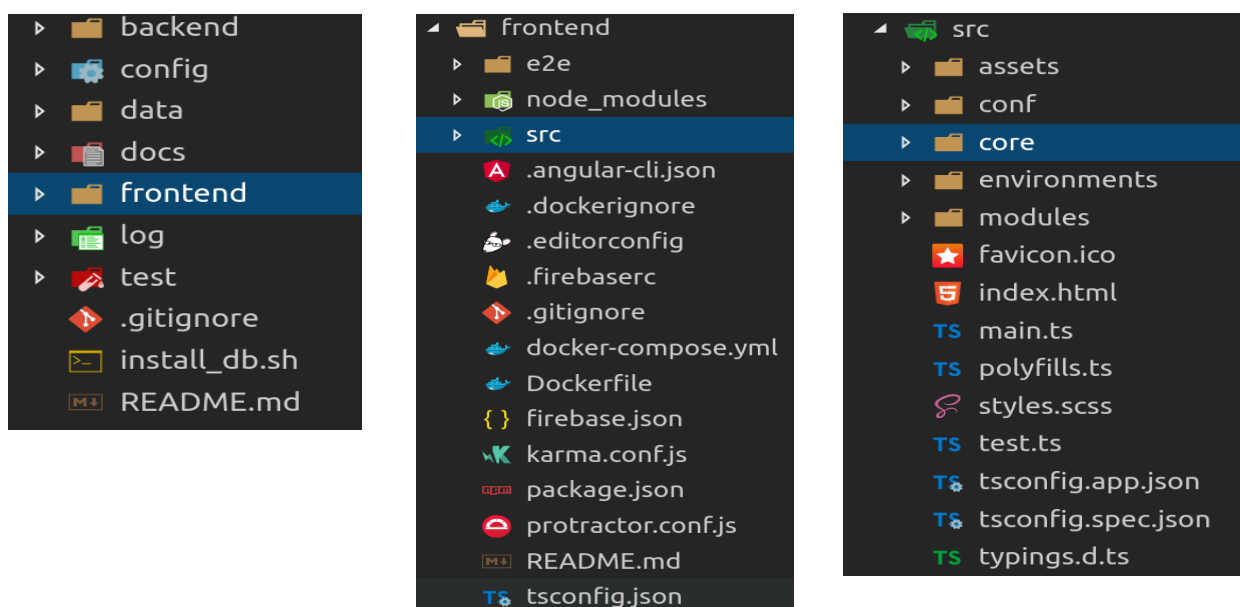


Illustration 11: Organisation frontend

- node_module, package.json : les packets et modules des dépendances des framework javascript pour le frontend.
- tslint.json, tsconfig.json : Fichiers de configuration pour TypeScript.
- angular-cli.json : configuration globale pour angular cli – une « command line interface » pour construire des applications Angular 4 en utilisant le style nodejs.
- Dockerfile, docker-compose : Configuration pour docker.
- src : comme le backend et la base de donnée, l'organisation des répertoires du frontend est structurée avec deux principaux répertoires, celui pour le core et celui pour les modules. Le core contient des composants communs à l'ensemble de GeoNature, et les modules contiennent des modules séparés correspondant à chaque protocole.

6.2 Frontend Core – Modules

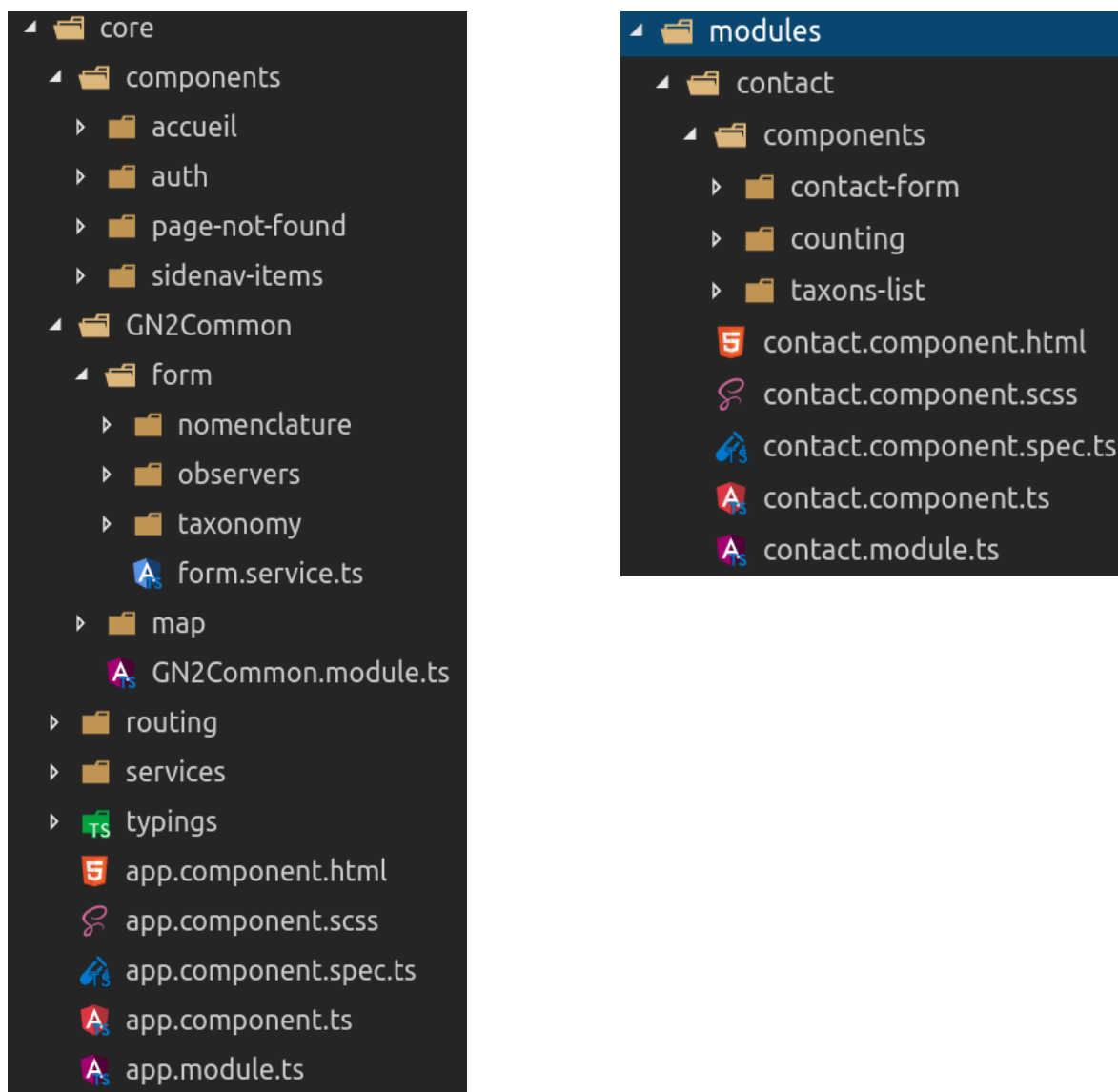


Illustration 12: Frontend core et modules

Dans l'image à gauche, dans composants du core, on retrouve des composants tels que auth (l'authentification), sidenav, taxonomy, observers et nomenclature. Ces composants ont été conçus individuellement et fonctionnent indépendamment d'autres composants pour assurer leur réutilisation, dans différents formulaires des protocoles.

Sur l'image de droite, nous nous voyons le détail du module Contact Faune-Flore, avec ses composants contact-form, counting, taxons-list qui utilisent des mini-composants intégrés dans le core (taxonomy, observers, nomenclature...).

6.3 Frontend Interface

L'interface utilisateur est conçue comme un tableau de bord. A gauche on peut accéder aux différents modules de l'application, à droite on affiche le contenu de chaque module. Pour concevoir cette interface, j'ai utilisé Angular Material Design - un framework qui respectait la conception de Google Material.

Dans le processus de conception de l'interface, j'ai également rencontré de nombreux difficultés de conception ux/ui, ainsi que la façon de mettre en œuvre le framework css frontend pour bien organiser les composants de l'interface et essayer de créer une interface « human friendly ».

Pour tester la demo :

<https://geonature-a568d.firebaseio.com/>

Compte : test@test.com

Mot de passe : 123456

6.3.1 Menu Bar - SideNav-Bar

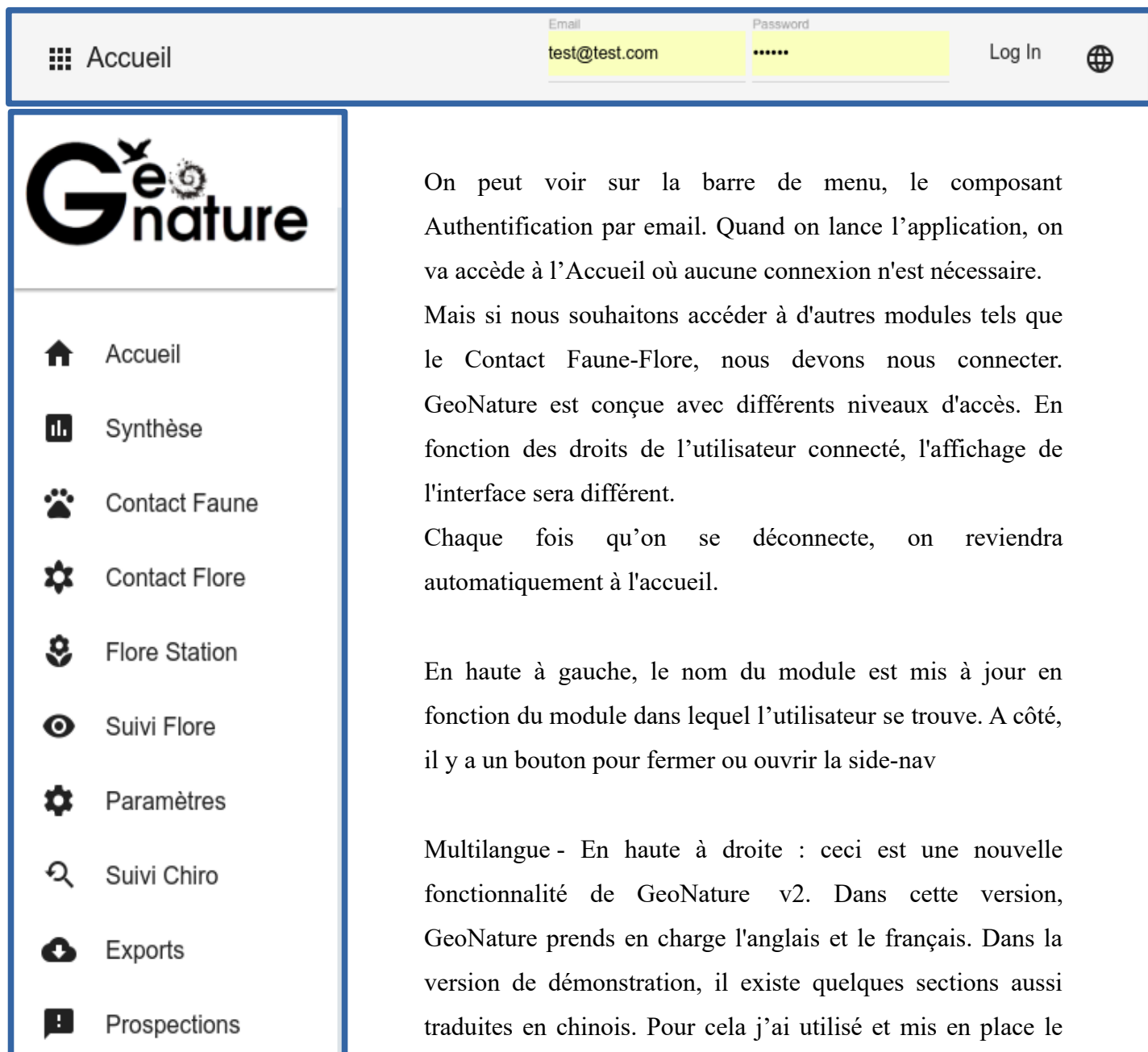


Illustration 13: Menu Bar - SideNav-Bar

On peut voir sur la barre de menu, le composant Authentification par email. Quand on lance l'application, on va accéder à l'Accueil où aucune connexion n'est nécessaire.

Mais si nous souhaitons accéder à d'autres modules tels que le Contact Faune-Flore, nous devons nous connecter. GeoNature est conçue avec différents niveaux d'accès. En fonction des droits de l'utilisateur connecté, l'affichage de l'interface sera différent.

Chaque fois qu'on se déconnecte, on reviendra automatiquement à l'accueil.

En haut à gauche, le nom du module est mis à jour en fonction du module dans lequel l'utilisateur se trouve. À côté, il y a un bouton pour fermer ou ouvrir la side-nav.

Multilangue - En haut à droite : ceci est une nouvelle fonctionnalité de GeoNature v2. Dans cette version, GeoNature prends en charge l'anglais et le français. Dans la version de démonstration, il existe quelques sections aussi traduites en chinois. Pour cela j'ai utilisé et mis en place le module ngx-translate.

À gauche on affiche la liste des modules de l'application. C'est le module side-navigation. Chaque module correspond à un protocole. En haut de la side-nav on retrouve le logo de GeoNature que j'ai réalisé.

6.3.2 Accueil

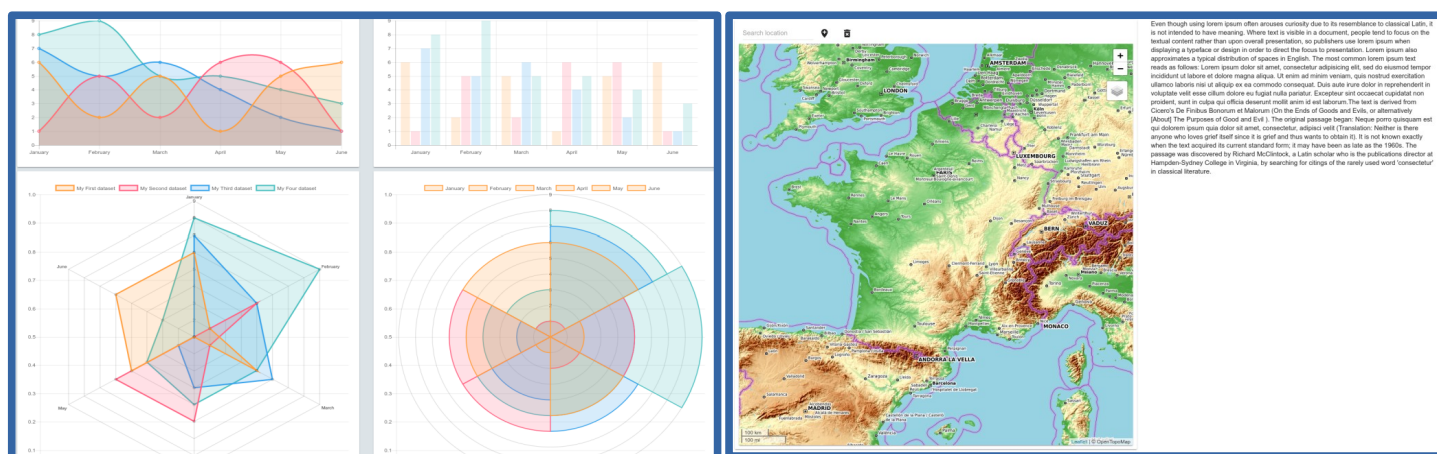
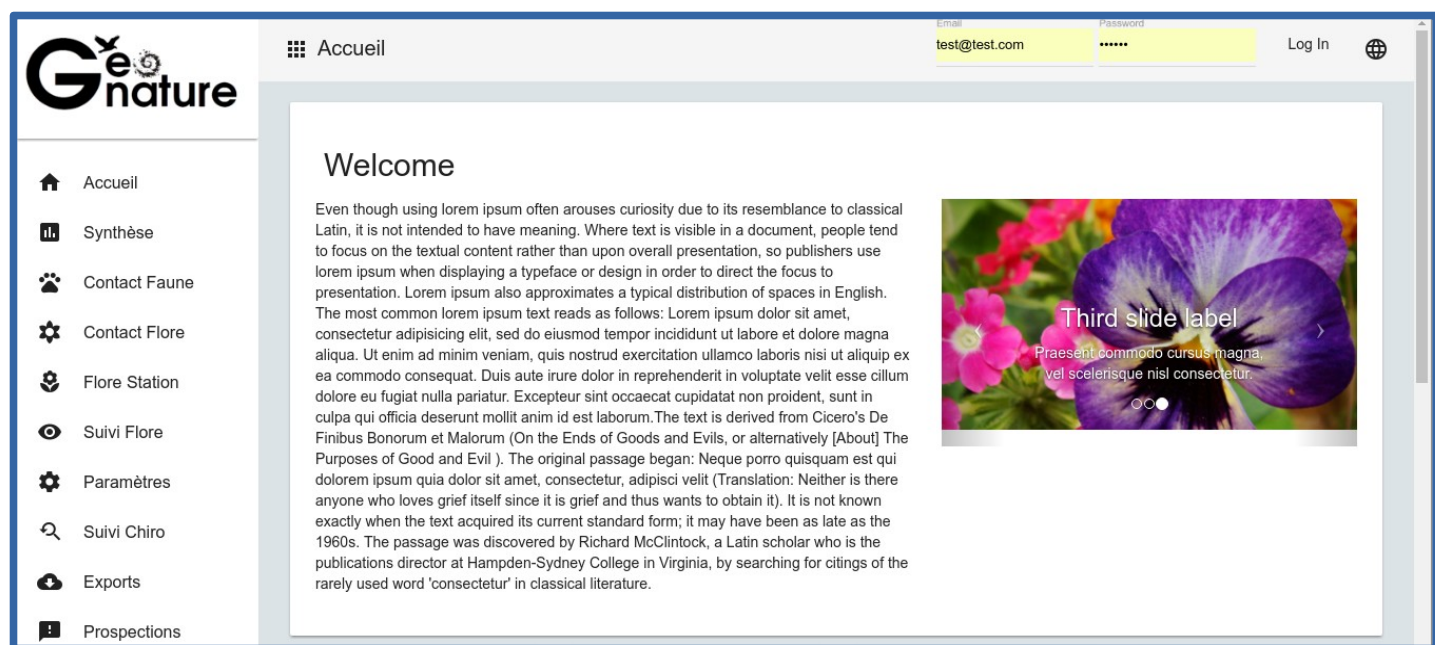


Illustration 14: Page Accueil GeoNature 2

L'image ci-dessus correspond à l'interface de la page d'accueil. Pour accéder à cette page, aucune connexion n'est nécessaire.

Sur la page actuelle, il existe trois composants principaux: un texte introduction et de présentation de l'application, suivi d'une carte dynamique avec des fonctionnalités simple comme la recherche, l'affichage de coordonnées GPS chaque fois que l'on clique sur la carte, et des graphiques d'analyse automatique des données.

Sur la page d'accueil, j'ai testé la compatibilité des bibliothèques Leaflet et Chartjs avec Angular 4.

6.3.3 Module Contact

Search location



Observations:

Observateurs

Date min

Date max

Alt min

Alt max

Commune

Lot de données

Technique observation

Contexte & Commentaire

Taxon observé:

Noms de taxon

Methode observation

Avancé

Stade vie

Sexe

Type

Obf

Denom

Ajouter

Adulte

Inconnu

Type

Obf

1

+

Commentaire:

Valider taxon

Taxons saisis:

Num	Observateur	Taxon	Modif	Suppr
1	admin	Bouquetin des Alpes		
2	geonatuser	Campagnol amphibie		

Ajouter taxons

Illustration 15: Module Contact

Le contact faune flore est un processus d’observation de la faune ou de la flore. Il est caractérisé par l’absence de protocole conduisant à l’observation. L’observation se fait de manière aléatoire dans le temps et dans l’espace avec différentes techniques d’observation et des méthodes scientifiques spécifiques au différents groupes d’espèces.

C'est l'un des modules les plus importants de l'application car c'est celui utilisé le plus fréquemment. Le module de contact se compose de deux composants principaux: la carte et le formulaire. Il y a d'autres petits composants intégrés dans la formulaire comme taxonomy, nomenclature et observers.

6.3.4 Composant map

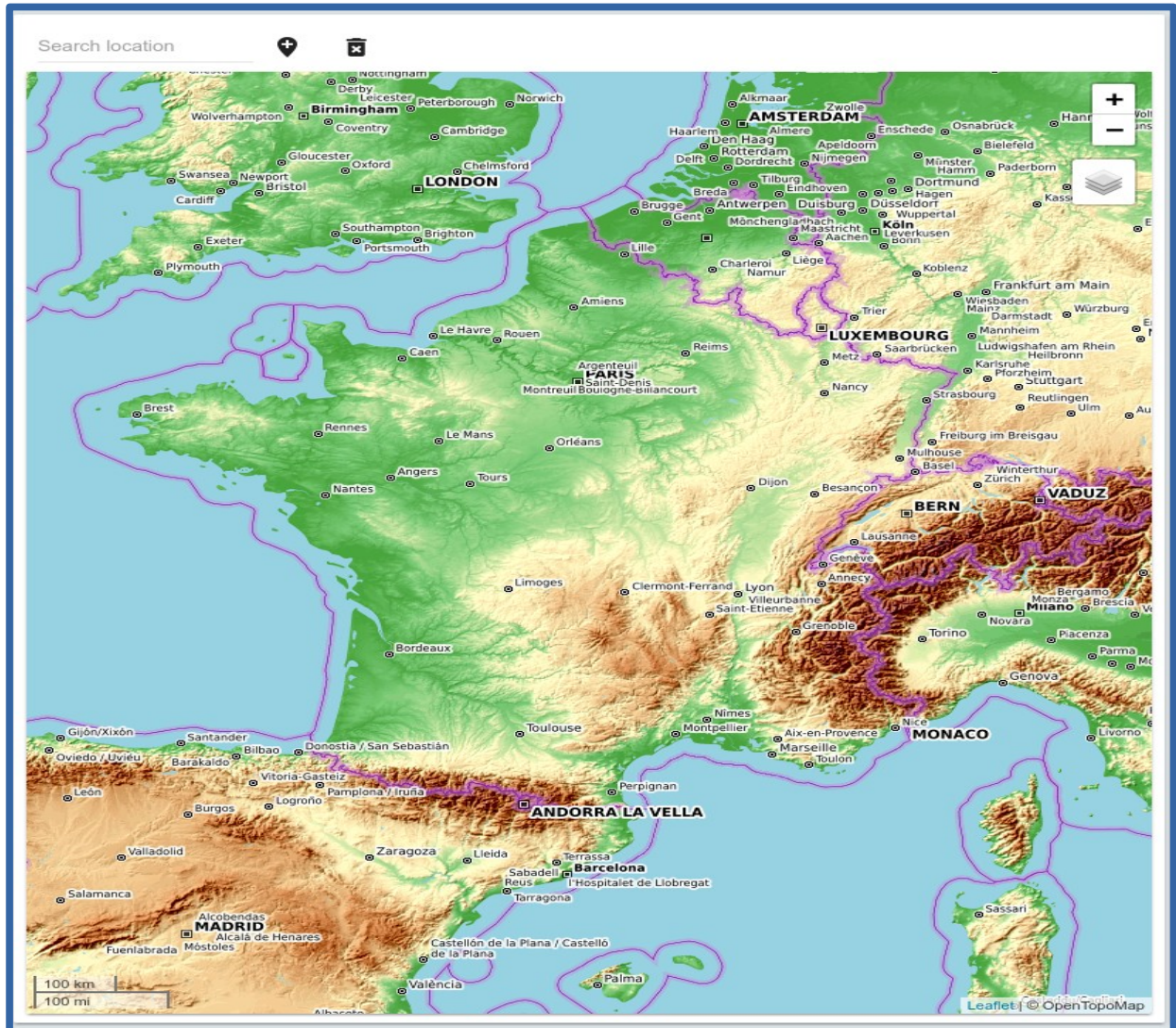


Illustration 16: Composant map

C'est également l'un des composants les plus importants de l'application, qui sera réutilisé à plusieurs reprises dans différents modules qui sont basés sur la bibliothèque javascript Leaflet de cartographie.

Le coin supérieur gauche est une fonction de recherche de nom de lieux sur la carte du monde. Cette fonction est réalisée sur la base de l'API OpenStreetMap.

D'autres fonctions comme le dessin de polygone, polyligne sont affichées en fonction du module. Par exemple sur Accueil, cette fonction sera masquée, mais sur le Contact, elle sera affichée.

Chaque clic sur la carte affichera les coordonnées GPS de ce point.

Il est possible de naviguer, zoomer, se déplacer dans la carte et de changer le fond cartographique (ex: OpenStreetMap, OpenTopoMap, Google satellite).

6.3.5 Composant form contact:

The form is divided into three main sections:

- Observations:** Includes fields for Observateurs, Date min, Date max, Alt min, Alt max, Commune, Lot de données, Technique observation, and Contexte & Commentaire.
- Taxon observé:** Includes fields for Noms de taxon, Methode observation, Avancé, Stade vie, Sexe, Type, Obf, Denom, and Ajouter. Below these is a Commentaire field.
- Taxons saisis:** A table with columns Num, Observateur, Taxon, Modif, and Suppr. It contains two rows of data.

Buttons: Valider taxon, Ajouter taxons.

Illustration 17: Composant form contact

Le formulaire est divisé en trois sections principales: Relevé, Taxon observé et Taxons saisis. Après avoir entré les données requises sur Relevé et Taxon observé, en cliquant sur le bouton "Valider taxon", les nouvelles données seront ajoutées à la table Taxons saisis. Le bouton "Ajouter taxons" permet d'ajouter les nouvelles données dans la base de données.

Dans le formulaire Contact, il existe d'autres petits composants tels que taxonomy, nomenclature, observers qui sont coupés en composants séparés pour permettre leur réutilisation dans d'autres modules.

Dans la conception de ux / ui de ce formulaire, j'ai également eu de nombreuses difficultés car il y a beaucoup de champs de saisie dans le même formulaire. Cela sera évoqué dans la prochaine section "Formulaire avec options".

6.3.6 Composant form contact avec les options

Observations:

Observateurs

Date min

Date max

Alt min

Alt max

Commune

Lot de données

Technique observation

Contexte & Commentaire

Contexte:

Commentaire:

Taxon observé:

Groupe: None

☐ Patrimoniaux
 ☐ Priorité
 ☐ Nouveau
 ☐ Facultatif

Noms de taxon

Méthode observation

Avancé

Etat Bio

Statut Bio

Naturalite

Déterminateur

Method déter

Num prelev

Preuve num

PreuveNom n

Stade vie

Sexe

Type

Obf

Denom

Ajouter

Commentaire:

Valider taxon

Taxons saisis:

Num	Observateur	Taxon	Modif	Suppr
1	admin	Bouquetin des Alpes		
2	geonatuser	Campagnol amphibie		

Illustration 18: Composant form contact avec les options

Voici le formulaire après avoir ouvert toutes les fonctionnalités avancées. On peut voir qu'il y a plus de 30 champs de saisie dans le même formulaire. Le travail d'organisation des champs de saisie sans casser le formulaire et ne pas affecter à la « human-friendly » interface est une tâche importante.

Il existe deux possibilités de conception qui ont été prises en compte, soit en ajoutant un nouvel onglet avec des fonctionnalités avancées, soit en masquant par défaut les fonctionnalités avancées avec des boutons. Après avoir fait référence aux différents designs sur le web Google, Wiki, Youtube, j'ai choisi la deuxième option.

6.3.7 Les technos frontend

FrameWorks



Libraries



Languages



Others



Illustration 19: Les technos frontend

7. Backend

J'ai commencé à développer le backend pendant la période où le reset de l'équipe était en vacances d'été, pendant environ 2 semaines. Le développement du backend était relativement simple car Flask est un bon microframework, facile à utiliser et que j'avais bien pris en main lors de mon travail de début de stage sur TaxHub.

À ce moment, la base de données GeoNature 2 était encore dans l'étape d'initialisation. Les évolutions fréquentes de la base de données entraînant également une modification du backend. Donc ce que j'ai fait sur le backend est principalement le travail de création d'une structure de code permettant l'héritage et la réutilisabilité.

Pendant le développement du backend, j'ai ajouté la bibliothèque Flask_RestfulAPI, qui est une bibliothèque utilisée pour simplifier le processus de conception de l'API. J'ai également ajouté Postman comme un outil pour simplifier les tests de l'API, la documentation de l'API et le partage de cas de test entre les membres. Après un workshop de développement à Briançon fin août nous avons décidé de ne plus utiliser Flask_RestfulAPI.

7.1 Organisation backend

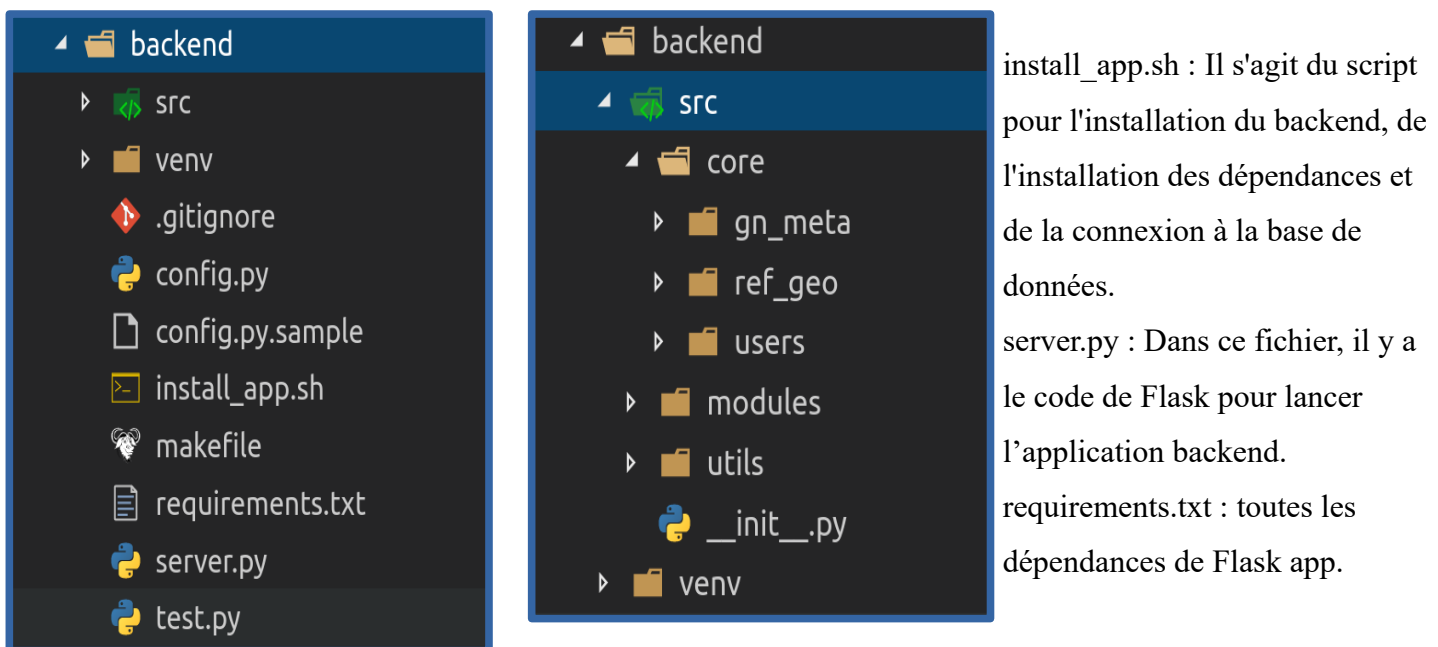


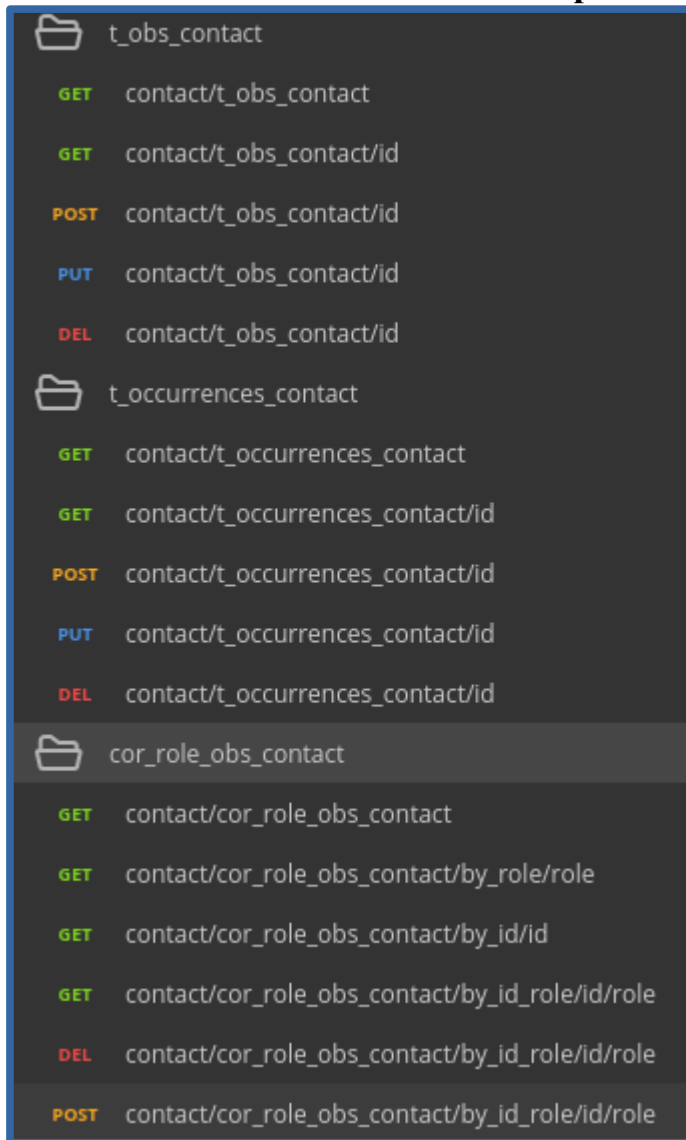
Illustration 20: Organisation backend

venv : Virtualenv pour python – une environnement indépendant où on installe toutes les dépendances du backend – python.

utils : les classes, les fonctions outils, le code répété.

Src : il y a un répertoire module et un core (structure similaire à la base de données et à la base de données)

7.2 Backend API REST conception



	t_obs_contact
GET	contact/t_obs_contact
GET	contact/t_obs_contact/id
POST	contact/t_obs_contact/id
PUT	contact/t_obs_contact/id
DEL	contact/t_obs_contact/id
	t_occurrences_contact
GET	contact/t_occurrences_contact
GET	contact/t_occurrences_contact/id
POST	contact/t_occurrences_contact/id
PUT	contact/t_occurrences_contact/id
DEL	contact/t_occurrences_contact/id
	cor_role_obs_contact
GET	contact/cor_role_obs_contact
GET	contact/cor_role_obs_contact/by_role/role
GET	contact/cor_role_obs_contact/by_id/id
GET	contact/cor_role_obs_contact/by_id_role/id/role
DEL	contact/cor_role_obs_contact/by_id_role/id/role
POST	contact/cor_role_obs_contact/by_id_role/id/role

Illustration 21: Backend API REST conception

J'ai travaillé sur la conception de l'API lorsque la base de données et le MCD étaient dans la phase d'initialisation. Les routes de l'api ont ensuite évoluées, mais le travail de backend m'a permis d'élargir mes connaissances sur le développement web lorsque je créais mon propre environnement complet et que je travaillais avec le backend et le frontend.

L'image à côté est la conception de l'api que j'ai implémentée,. Chaque répertoire correspondant à chaque table dans la base. Pour chaque table dans la base, on a une api pour récupérer toutes les données dans le schéma, et une api pour Get, Post, Delete, Put par id de table.

Au niveau du code, la tâche de créer une api pour chaque table de la base est répété alors j'ai appliqué 'singleton pattern' pour simplifier cette tâche.

Il s'agit d'une api simple car il n'y a pas de relations entre les tables, dans cette partie. J'aurai souhaité en développer davantage, mais en raison des limites du temps de stage, de l'importance du test des technologies ainsi que l'initialisation de l'environnement, je n'ai pas pu aller plus loin.

Au cours de la conception de l'api, j'ai utilisé Postman, un outil utile pour partager des cas de test d'une API, ainsi que l'auto-documentation d'une API et pour partager la conception de l'API entre les développeurs backend, les développeurs frontend ou d'autres utilisateurs. Cet outil a considérablement simplifié le travail en équipe.

7.3 Les technologies backend



Illustration 22: Les technologies backend

8. Dev-ops

8.1 Docker

- Docker est une technologie qui simplifie et accélère le processus de déploiement de logiciels en créant un environnement de travail unifié et un emballage dans un conteneur à partir duquel les développeurs peuvent travailler sans perdre de temps. Il permet d'installer les logiciels, les dépendances, l'environnement de configuration des bibliothèques. Et le conteneur prend aussi en charge le déploiement sur le serveur.
- Au cours des tests des technologies, j'ai également passé une semaine et demi à mettre en place le docker et le docker-compose. Actuellement j'ai initialisé un conteneur pour déployer le frontend. Pour créer un environnement docker complet pour le backend et la base de données, il faudra plus de temps, car GeoNature est un projet relativement complexe. La complexité de GeoNature est due aux applications dont il dépend tel que TaxHub et UsersHub. Afin de créer un environnement pour le docker il faut donc beaucoup de temps et il est nécessaire de savoir comment configurer l'environnement pour chaque application.

8.2 Github

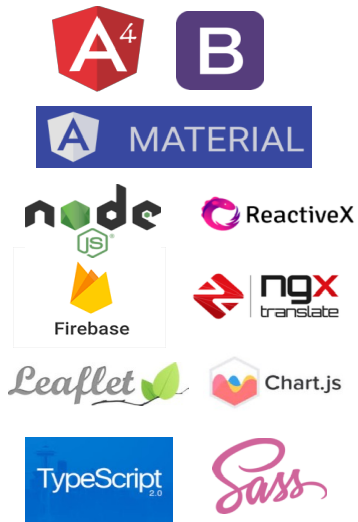
- Voir ma présentation de la façon dont le PNE utilise github sur mon rapport de mi-stage, page 11 <http://geonature.fr/documents/2017-05-QuangPham-Rapport-mi-stage.pdf>
- Pendant le temps que j'ai travaillé ici, Camille a organisé le travail de l'équipe selon les méthodes Agiles avec Github. C'est aussi une chose très intéressante que j'ai appris. La méthode Agile avec github sera présentée plus précisément dans la section sur le travail de groupe.

8.3 Automatisation avec script SH

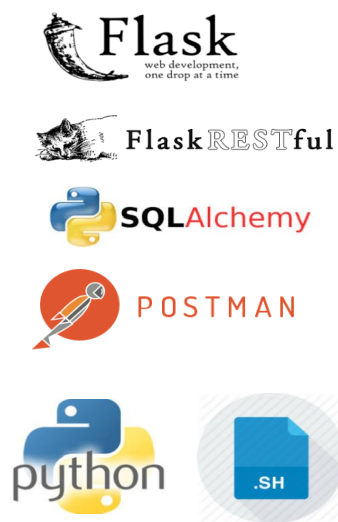
- Au niveau de la base de données et du Backend, on a utilisé des scripts SH pour simplifier l'installation et la configuration du backend ainsi que la base de données. Avec les conseils de Gil, j'ai créé un script pour installer et configurer l'environnement du backend, ce qui est une connaissance très intéressante.

9. Toutes les technologies utilisées

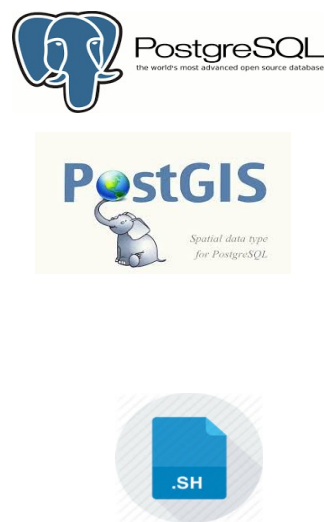
Front-end



Back-end



Database



Dev-Ops

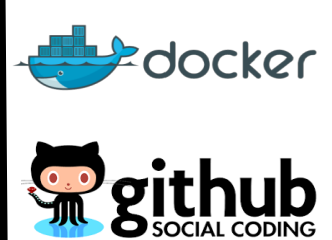


Illustration 23: Toutes les technologies utilisées

VI . Travail collectif

1. Agile avec Github

<https://github.com/PnX-SI/GeoNature/projects/1>

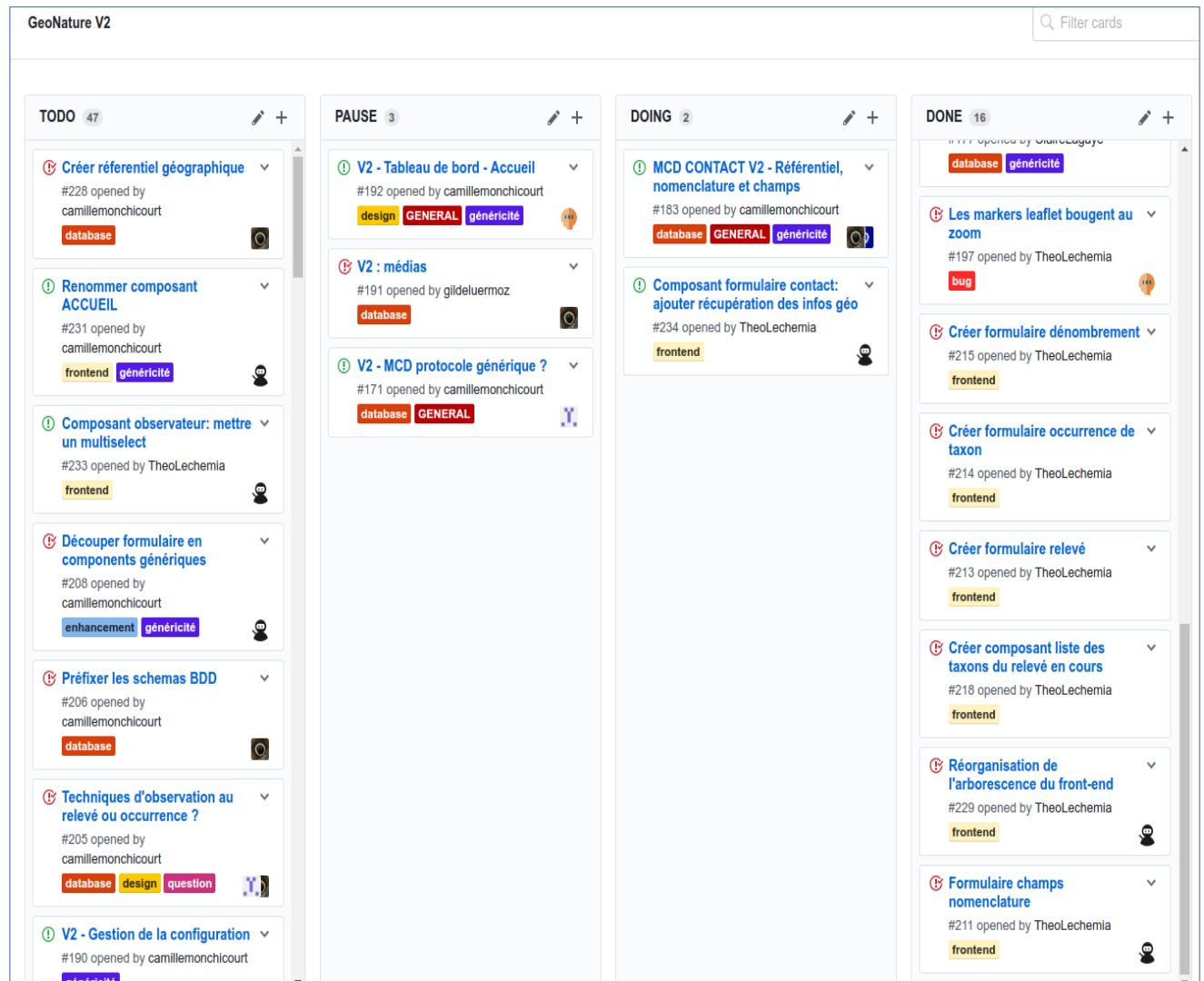


Illustration 24: Method Agile de GeoNature avec Github

Dans l'image ci-dessus, on voit le projet Github, avec 4 colonnes (TODO / PAUSE / DOING / DONE) :

- TODO : les tâches sont estimées pour être complétées dans un sprint mensuel.
- PAUSE : les tâches étaient auparavant dans DOING, mais en raison de certaines limitations, ces tâches sont en pause temporairement.
- DOING : les tâches sont en cours de développement.
- DONE : les tâches complétées.

V2 - Tableau de bord - Accueil #192

[Open](#)
camillemonchicourt opened this issue on 27 Jun · 12 comments

camillemonchicourt commented on 27 Jun • edited

Owner

Dans la V2, il est souhaité améliorer la page d'accueil de l'outil pour en faire un vrai tableau de bord avec un aperçu des données existantes dans la BDD, quelques graphiques sur la répartition des données, un aperçu des dernières données, des liens directs vers les différents modules.

Une première maquette pour illustrer cela avait été réalisée :

Il est défini qu'une sidebar est présente sur la page d'accueil mais est aussi intégrée dans les différents modules. Il est donc important d'en faire un composant/template générique que l'on peut intégrer dans

Assignees

QuangPhamII

Labels

design

GENERAL

généricité

Projects

PAUSE in GeoNature V2

Milestone

V2 - Sprint Septe...

Notifications

Subscribe

You're ignoring this thread.

4 participants

Lock conversation

Illustration 25: La tâche dans la method Agile avec Github

Chaque tâche est une "issue" sur Github, chaque tâche a une description de travail spécifique et a des discussions avec les autres membres. Chaque tâche a deux états: Open (en vert) ou Closed (en rouge)

Assignees

QuangPhamII

Labels

design

GENERAL

généricité

Projects

PAUSE in GeoNature V2

Milestone

V2 - Sprint Septe...

Notifications

Subscribe

You're ignoring this thread.

4 participants

Chaque tâche est assignée à un ou plusieurs membres.

Elle a des labels pour spécifier les champs d'activité.

Elle spécifie aussi les projets, milestone et les participants.

2. Workshop de développement (Briançon – 21 aout 2017)



Illustration 26: Workshop de développement

Dans la phase finale du stage, j'ai eu l'occasion de passer une semaine de Workshop passionnante à Briançon. Dans ce Workshop, nous avons travaillé collectivement sur de nombreux problèmes qui n'étaient pas réglés avant. Ce sont des problèmes et choix qui vont de la base de données à l'interface, du choix des technologies aux outils utilisés. Durant ce travail collectif, de nombreux problèmes ont été résolus. Le projet GeoNature v2 a beaucoup avancé après cette semaine de travail.

A côté du travail collectif sur Github ou dans le Workshop, on a aussi organisé des réunions téléphoniques régulières.

VII. Calendrier de travail

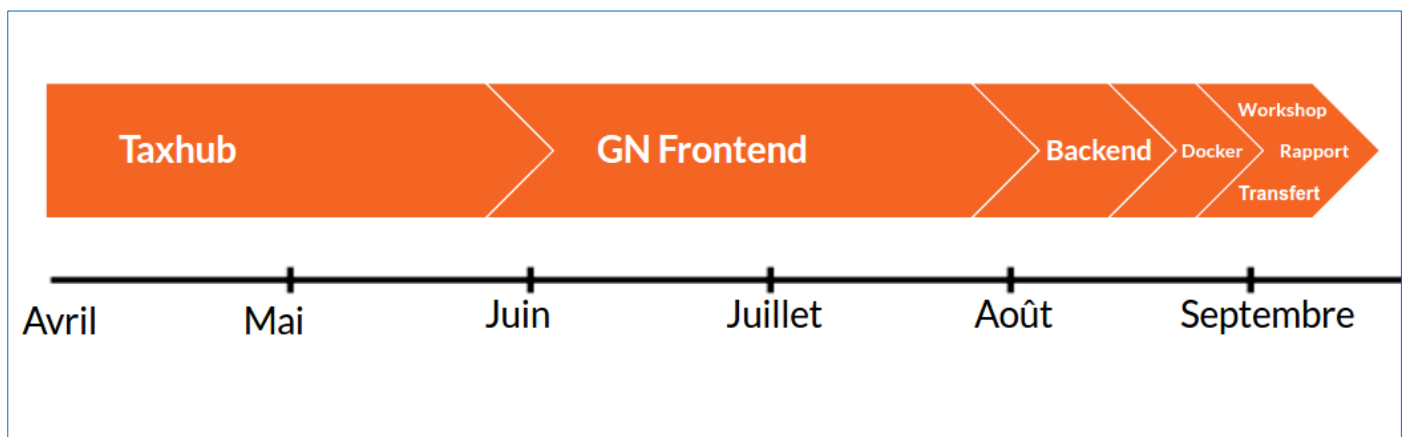


Illustration 27: Calendrier de travail

1. Taxhub

Rapport de mi stage <http://geonature.fr/documents/2017-05-QuangPham-Rapport-mi-stage.pdf>

2. GeoNature Frontend

Cette étape de test du frontend, de choix des technologies, d'auto-apprentissage technologies, des tests de compatibilité avec les frameworks et les bibliothèques, de développement d'une démo a pris environ 1,5 mois. Puis l'étape de conception de l'interface avec Materiel Design et Bootstrap a pris 2 semaines. Une démonstration a été réalisée après 2 mois de travail.

Pour tester la démonstration sur : <https://geonature-a568d.firebaseio.com/>

Compte : test@test.com

Mot de passe : 123456

3. GeoNature Backend

Le travail de création d'un environnement du backend a pris deux semaines. Pour moi, c'était un travail vraiment intéressant, bien que l'environnement backend n'ait pas été utilisé dans le projet, mais j'ai acquis beaucoup de connaissance intéressantes et très utiles. Avant cela il n'y avait pas d'outil pour tester les APIs, partager les APIs, et faire la documentation des APIs. Postman – l'outil que j'utilise pendant le développement du backend, a été ensuite déployé et utilisé par l'ensemble de l'équipe.

4. Docker

J'ai passé une semaine à explorer cette technologie et l'ai appliqué avec succès au développement de l'environnement du frontend.

5. Workshop – Rapport – Transfert

Pendant les 3 dernières semaines de mon stage, il y a eu une semaine pour le Workshop à Briançon. Puis les deux dernières semaines, j'ai transféré le projet à Théo Lechémia qui va désormais développer GeoNature pendant 3 ans. J'ai aussi pris du temps pour faire mon rapport de stage.

VIII. Futur de GeoNature

GeoNature est une application qui augmente fortement. A l'avenir, elle offre beaucoup de potentiel à explorer.

Ci-dessous, je vais énumérer certaines des technologies qui pourraient être appliquées au projet GeoNature et celles qui pourraient faire évoluer cette application à l'avenir.

1. Ionic

Ionic Framework est un mélange d'outils et de technologies pour développer des applications mobiles hybrides rapidement et facilement. Il s'appuie sur Angular pour la partie application web du framework et sur Cordova pour la partie construction des applications natives. Ce framework open source permet de développer une application déployable sur plusieurs environnements tel qu'un site web ou une application mobile pour des systèmes tel que Android ou iOS.

Avec la GeoNature 1, une entreprise a été engagée pour créer une application mobile basée sur les fonctionnalités de GeoNature Web. Ce travail est compliqué à amener en terme de ressources personnelles du PNE pour pouvoir suivre le projet mobile.

IONIC : serait une bonne solution à ce problème, et il sera encore plus pratique lorsque GeoNature sera entièrement développé avec Angular 4. Il y aura des adaptations à faire dans le CSS et HTML pour à taille de l'application mobile notamment, mais la logique du core de TypeScript et Angular ne changera pas par rapport à une application pour le Web.

2. Progressive web apps

Une PWA est un site internet développé spécifiquement pour les mobiles. Pour cela, elle repose sur une architecture de type Application Shell, ce qui rend possible son utilisation en mode hors-ligne.

Si l'on applique la technologie de PWA dans GeoNature, puis que l'on encapsule le HTML, CSS, TypeScript de GeoNature avec IONIC, alors on aura une application mobile qui marche hors-ligne. Les données saisies seront stockées et sauvegardées hors-ligne puis synchronisées avec la base de données sur le serveur quand on se reconnectera à internet.

Il y a aussi la mécanique permettant de pousser des notifications à l'utilisateur.

3. Docker

L'installation d'un environnement de travail pour une application comme GeoNature est toujours une tâche complexe. Le problème de « It works on my machine » existe toujours.

GeoNature est un projet open source, lorsqu'il sera largement développé, il y aura plus de développeurs qui souhaiteront participer au développement, apporter des modifications à GeoNature comme ils veulent et souhaitent déployer sur leur serveur. C'est beaucoup plus simple si tout cela se fait avec une seule ligne de commande de docker.

IX. Les difficultés

1. Difficulté du choix des technologies

Angular 4 vs Vuejs

J'ai travaillé de manière importante sur ce sujet, pendant plus d'un mois, c'est le sujet le plus considéré. Nous avons d'abord comparé les frameworks javascript, testé Angular et Vuejs. En plus des discussions dans le groupe, Camille a également demandé des conseils aux autres développeurs qu'il connaissait. Cependant il était difficile d'arriver à un consensus final, et pendant plus d'un mois, le groupe n'avait pas de position claire sur le sujet. Camille a décidé de commencer à développer une démonstration avec Angular. *Voir la partie Renforcer le projet avec Python Flask et Angular 4 pour plus de détail.*

Bootstrap vs Materiel Design Angular

Bootstrap est le Framework Css standard le plus populaire aujourd'hui. Les avantages de Bootstrap sont qu'il est facile à utiliser et il existe de nombreux exemples et documentations. Materiel Design Angular est un nouveau Framework Css basé sur la norme de conception de Materiel Design de Google et est le Framework pour Angular, avec des designs modernes et pratiques.

Il y a eu une réflexion dans le choix entre l'un de ces 2 Frameworks Css, avec le consensus de commencer à concevoir avec Bootstrap. Mais après quelques jours de test de Bootstrap pour réaliser le design du tableau de bord, j'ai rencontré de nombreuses difficultés. Celles-ci ont pu être résolues plus facilement avec Materiel Design.

Enfin, pour résoudre ce problème, j'ai intégré les deux Frameworks. Bootstrap et ses composants sont la structure principale de l'application combinée à un certain nombre de composants avec des designs modernes et pratiques de Materiel Design.

2. Difficulté de la discussion en groupe

Il y a toujours des difficultés dans les discussions de groupe en raison de ma contrainte linguistique. Pour surmonter cette limitation, Camille et Gil posent toujours des questions pour confirmer et m'expliquer les idées qui doivent être faites en détail.

X. La conclusion

Après six mois, ce stage m'a permis de développer de nombreuses compétences nouvelles. J'ai énormément appris, autant sur la partie architecture et développement web, que sur la partie base de données, mais aussi sur les nouvelles technologies : Python Flask, Angularjs, Angular 4, PostgreSQL, REST API, Docker ...

J'ai eu la chance de suivre et participer à la conception du projet TaxHub, GeoNature 2 (définition du cahier des charges, MCD), à sa réalisation (choix des technologies et de l'architecture, développement), ce qui fut très enrichissant.

J'ai également eu la chance de rencontrer des collègues intéressants tels que Camille, Gil, Vincent, Théo et beaucoup d'autres collègues. Je travaille dans un environnement convivial. Pour un étranger comme moi, c'est une expérience extrêmement précieuse.

Les six derniers mois ont été les six meilleurs mois et les plus spéciaux que j'ai jamais eu dans mon séjour à l'étranger.