

Téléversement SINP



présenté pour l'obtention du

Diplôme de Master 2

Compétence Complémentaire en Informatique

Département informatique et Interactions

Année de soutenance : 2018

Remerciements

Je remercie Olivier Rovellotti et Christelle Khozian pour m'avoir donné la chance d'effectuer mon stage à Natural Solutions et pour le projet très intéressant et stimulant sur lequel j'ai travaillé pendant cette période. Je tiens à remercier Khaled Talbi pour son encadrement. Il a été d'une grande disponibilité tout au long du stage et a pris le temps de me suivre sur les différents aspects du développement de l'application et des méthodes de travail. Je remercie aussi Amine, Céline, Christophe, Cyril, El Makki, Fred, Naomi, Patrick, Romain, Tom, Vincent(s) pour leur disponibilité et leur aide lors du développement de l'application. Plus généralement un grand merci à toute l'équipe de Natural Solutions pour leur excellent accueil, le stage s'est déroulé dans une très bonne ambiance et c'était agréable de se sentir immédiatement intégré à l'équipe.

Je tiens à remercier Jean-Luc Mari et l'équipe pédagogique du Master 2 CCI pour m'avoir permis de suivre cette formation. La structure et la qualité des enseignements ont permis en quelques mois d'aborder des aspects fondamentaux nécessaires à nous mettre sur les rails du monde professionnel. Je voudrais également exprimer ma gratitude à Sylvie Risch pour son aide indispensable dans les méandres des financements de formations professionnelles.

Enfin, et surtout, j'aimerais remercier Léa qui me soutient pleinement dans ce processus de reconversion professionnelle, ce qui n'est pas toujours évident en cette année particulière et magnifique de la naissance de notre fils.

Résumé

J'ai réalisé mon stage à Natural Solutions, une entreprise spécialisée dans la gestion des données de biodiversité.

Depuis juin 2018, une nouvelle loi impose aux maîtres d'œuvre de projets publics ou privés impactant l'environnement de téléverser toutes les données brutes de biodiversité (collectées notamment lors d'études d'impact) sur une plate-forme web officielle. Les données brutes devant être formatées selon un standard précis, rigide et complexe (le standard « occurrences de taxons » du SINP ; Système d'Information sur la Nature et les Paysages) pour être acceptées par la plate-forme web, les maîtres d'œuvre vont avoir besoin d'outils et d'assistances pour standardiser leurs données parfois volumineuses. Par son expertise en gestion de données de biodiversité, Natural Solutions désire logiquement se positionner sur cette nouvelle activité.

L'objectif de mon stage était donc de **développer les outils et services nécessaires au téléversement des données brutes de biodiversité** collectées par les maîtres d'ouvrage. Pour cela, la majeure partie de mon stage a été consacrée au développement d'une application de téléversement permettant de convertir les données brutes de biodiversité au standard SINP. En parallèle au développement de l'application, j'ai également réalisé deux prestations de téléversement de données brutes de biodiversité et rencontré trois clients pour présenter nos développements en cours dans ce domaine. Enfin j'ai écrit une documentation sur le site web de Natural Solutions.

Glossaire des abréviations :

NS : Natural Solutions

GINCO : Gestion d'Information Naturaliste Collaborative et Ouverte

SINP : Système d'Information sur la Nature et les Paysages

UX : User eXperience

Table des matières

1.	Introduction.....	1
1.1.	Présentation de Natural Solutions	1
1.1.1.	Présentation générale.....	1
1.1.2.	Principaux outils à destination des professionnels de la nature.....	1
1.1.3.	Principaux outils à destination du grand public.....	2
1.1.4.	Méthode de travail	3
1.2.	Le téléversement des données brutes de biodiversité : de quoi s’agit-il ?.....	3
1.3.	Problématique : l’application concrète de la loi et ses difficultés	4
1.4.	Objectifs du stage.....	4
2.	Cahier des charges de l’application de téléversement	7
2.1.	Objectif principal	7
2.2.	Une application simple et rapide d’utilisation	7
2.3.	Quel niveau de détails du standard SINP pour l’application?	8
2.4.	La collecte d’informations utilisateurs	8
2.5.	Les informations taxonomiques	9
2.6.	Les informations chronologiques	9
2.7.	Les informations géographiques	9
2.8.	Les informations globales.....	10
2.9.	Les identifiants de l’observation	10
2.10.	Format du fichier final à fournir à l’utilisateur	11
2.11.	Gestion des erreurs utilisateurs	11
2.11.1.	Erreurs de format de fichier utilisateur	11
2.11.2.	Vérifications préliminaires.....	11
2.11.3.	Erreurs de conformité et cohérence	12
2.12.	Téléchargement du fichier final au standard SINP.....	13
3.	Travail effectué	15
3.1.	Methodologie de travail.....	15
3.2.	L’application de téléversement.....	15
3.2.1.	Documentation et définition des spécificités techniques du standard SINP	15
3.2.2.	Le choix des technologies	15
3.2.3.	Le choix de l’architecture et de la communication client/serveur	16
3.2.4.	L’application côté client.....	16
3.2.5.	Application coté serveur	17
3.2.6.	La correspondance des données utilisateurs avec les référentiels.....	19
3.2.7.	Design et intégration	20
3.3.	Autres activités.....	21
3.3.1.	Prestations de téléversement.....	21

3.3.2.	Rendez-vous avec clients potentiels	21
3.3.3.	Rédaction de documents	21
4.	Résultats	23
4.1.	Ecran d'accueil	23
4.2.	Ecran d'importation du fichier utilisateur	23
4.3.	Ecran d'identification taxonomique	26
4.4.	Ecran d'informations chronologiques	27
4.5.	Ecran d'informations géographiques	28
4.6.	Ecran d'informations globales sur le jeu de données	29
4.7.	Ecran de téléchargement du fichier standardisé	30
4.8.	Navigation	30
5.	Problèmes rencontrés et perspectives	33
5.1.	Compréhension du standard SINP	33
5.2.	Formation aux technologies abordées	33
5.3.	Respect des objectifs.....	33
5.4.	Développements futurs.....	33
6.	Conclusion	35
	Références bibliographiques	37
	Liste des légendes.....	37

1. Introduction

Mon stage s'est déroulé du 9 avril au 28 septembre 2018. J'ai été encadré par Khaled Talbi, chef de projet et développeur. J'ai fait le choix de réaliser mon stage chez Natural Solutions (NS) parce que l'entreprise est spécialisée dans le développement web dans le domaine de la biodiversité. Ceci alliait donc à la fois mes souhaits de reconversion professionnelle (développement) et la mise en œuvre d'une partie de mes compétences acquises antérieurement (enseignement et recherche) dans l'analyse de données de biodiversité.

1.1. Présentation de Natural Solutions

1.1.1. Présentation générale

NS est une entreprise marseillaise, spécialisée dans la gestion des données de biodiversité. Elle a été créée en 2008 par Olivier Rovellotti, ingénieur informatique passionné de nature. Elle conçoit des outils numériques permettant une prise en charge fiable et moderne des données issues d'observations biologiques. Ainsi, en réponse aux besoins des professionnels et amateurs de l'environnement, NS édite des solutions informatiques visant à **collecter, structurer, diffuser et analyser les données environnementales**. L'entreprise intervient sur quatre pôles de compétences dans le domaine de la biodiversité :

- Edition de logiciels sur mesure
- Conception de sites web
- Conseil
- Recherche et développement

De fait, les réalisations de NS se placent dans trois axes :

- Gestion des données environnementales. Les solutions accompagnent les acteurs de l'écologie dans leur prise de données sur le terrain, structurent et archivent les données d'observations (faune / flore, localisations géographique, etc ...)
- Visualisation et édition de données environnementales (interfaces graphiques, outils de systèmes d'information géographique - SIG – etc.).
- Acquisition de données d'observation sur environnement mobile (Smartphone, GPS, ordinateur de poche)

1.1.2. Principaux outils à destination des professionnels de la nature

EcoRelevé

EcoRelevé est un système de gestion de données environnementales. Conçu pour accompagner les professionnels de l'environnement dans la gestion des données d'observation, ecoRelevé stocke et structure ces données (taxonomie, localisations géographiques, protocoles...). Pour faciliter le partage et la diffusion de données stockées, plusieurs formats d'export sont proposés à l'utilisateur. C'est un outil open source proposant la visualisation des données sur un ensemble de cartes. L'interface permet de visualiser les données en suivant trois axes : la taxonomie, la chronologie et la provenance des données.

Cette application est conçue pour faciliter:

- La saisie de données au travers d'un ensemble de formulaires personnalisables.
- L'import/export de données.
- La communication entre les différents modules de la plateforme.
- La consultation des données saisies et importées.

Track

Track est une plateforme de gestion de suivi d'élevage conservatoire des outardes houbara. L'application est construite autour d'un protocole présentant des données sur un oiseau du groupe, ou sur l'ensemble du cheptel. L'application permet donc une gestion globale mais également extrêmement précise de l'élevage.

Kitizen

Kitizen est un kit de sciences participatives, qui regroupe plusieurs technologies développées par NS, aussi bien pour les novices que pour les amateurs avertis et les experts. Il permet de combiner différents modules (formulaire de saisie, géolocalisation, prise de vue, partage, visualisation des données, notifications et personnalisation) pour constituer la plateforme idéale d'un projet participatif.

1.1.3. Principaux outils à destination du grand public

Sauvages de ma rue

Sauvages de ma rue est un service de sciences participatives, créé par NS pour Tela Botanica. Il permet d'identifier et photographier les plantes sauvages de la rue afin de transmettre les informations à des scientifiques. Il donne tous les outils nécessaires à la reconnaissance des plantes par étape, permet de participer à l'inventaire des plantes en ville, d'envoyer des observations au monde scientifique, et d'être tenu informé de l'évolution du programme.

EcoBalade

Il s'agit d'un site web et d'une application mobile qui permet d'aider les promeneurs à découvrir la biodiversité qui les entoure, au travers de trois étapes :

- Proposer des circuits de randonnée, pour inciter les gens à venir découvrir la biodiversité.
- Une fois sur place, aider à identifier les espèces présentes. Cela se fait avec l'aide de très nombreuses photographies, et une clé de détermination très simple pour les plantes à fleurs. Les fiches espèces mises à disposition donnent les informations principales permettant d'identifier l'espèce, quelques informations sur son mode de vie, et une information marquante (usage médicinale ou culinaire par exemple).
- Le carnet de terrain dans l'application permet de garder en mémoire les observations pour y revenir ensuite.

1.1.4. Méthode de travail

La gestion des projets se fait par la méthode agile SCRUM. Cette organisation du travail a pour but de prioriser les retours clients sur les fonctionnalités à développer. Ceci permet de garder constamment une adéquation entre les besoins/souhaits du client et les activités de l'équipe de développeurs en permettant une gestion plus flexible, adaptable et personnalisée des projets. Cette méthode permet d'autre part aux personnes impliquées dans le développement d'un projet de s'organiser efficacement en étant constamment au courant de la nature et de l'état d'avancement du travail de leurs collaborateurs.

La méthode SCRUM est itérative et incrémentielle, c'est à dire que des objectifs à court terme sont fixés avec le client sur une période donnée, puis à la fin de cette période, les besoins prioritaires du client sont redéfinis et des objectifs à court terme sont de nouveau fixés. Ceci est répété jusqu'à ce que le client soit satisfait du produit fourni. Le client a donc une grande visibilité technique (et financière) sur l'évolution du projet.

Concrètement, Natural Solutions fonctionne sur des itérations de deux semaines (*sprint*). Le premier jour du *sprint*, les objectifs principaux sont fixés par le développeur et son chef de projet. Ces objectifs sont déclinés en tâches précises à effectuer durant le *sprint*. Pour chaque tâche, le développeur estime le nombre d'heures nécessaires pour effectuer cette tâche. La liste des tâches (*backlog* du *sprint*) est renseignée sur une application dédiée et permet de voir l'état d'avancement du travail d'un développeur, d'un projet ou de l'ensemble des équipes durant le *sprint*.

Au cours du *sprint*, le projet Scrum est rythmé par une réunion rapide quotidienne (*daily*), au début de la journée, dans laquelle les participants décrivent : les activités réalisées la veille, ce qu'ils vont faire au cours de la journée et les éventuels problèmes rencontrés. Enfin, la dernière journée du *sprint* a lieu la rétrospective dans laquelle l'équipe passe en revue les objectifs atteints ou non afin de déterminer l'état d'avancement des projets et les éventuels réajustements nécessaires.

1.2. Le téléversement des données brutes de biodiversité : de quoi s'agit-il ?

Dans le cadre de projets/chantiers publics ou privés ayant un impact environnemental, les maîtres d'ouvrage ou bureaux d'études doivent réaliser une étude d'impact. Pour cela, ils vont réaliser des observations (généralement de terrain) sur la biodiversité, c'est à dire **lister la présence (ou absence) de spécimens de telle ou telle espèce à une date et un lieu précis.**

Le téléversement est la transmission de ces données au SINP via la plate-forme web GINCO (Gestion d'Information Naturaliste Collaborative et Ouverte). Le SINP est la structure nationale développée sous l'égide du Ministère de l'Écologie, du Développement durable et de l'Énergie dont l'objectif principal est de recenser et rassembler les données de biodiversité nationale.

Si le principe est simple, la difficulté technique réside dans le fait que la plate-forme web GINCO n'accepte pas le versement de données brutes de terrain telles quelles. **Il faut que les données brutes soient au préalable transformées et structurées en un format précis et rigide pour que la plate-forme les accepte.** C'est ce qu'on appelle le « **standard occurrences des taxons** » du SINP (Jomier *et al.*, 2016, 2018).

A partir de juin 2018 (dans le cadre l'application de la loi du 8 août 2016 sur la reconquête de la biodiversité et des paysages), le téléversement des données brutes de biodiversité est une **obligation légale** pour tous

les acteurs de la biodiversité. Les maîtres d'ouvrage doivent donc obligatoirement soumettre leurs données à la plate-forme web du SINP afin d'obtenir (instantanément) un « **certificat de conformité des données** » qui leur permettra de poursuivre le projet/chantier.

Ce processus très contraignant pour les maîtres d'ouvrage va permettre à terme d'**homogénéiser la qualité des données de biodiversité**. La tâche est gigantesque et ambitieuse si on pense à la disparité et quantité astronomique d'observations passées et à venir, réalisées partout dans le monde. Et tout cela dans le cadre d'une connaissance scientifique toujours en mouvement, en particulier concernant la classification du vivant (nomenclature et taxonomie).

Au niveau national, la procédure de téléversement va maintenant permettre d'avoir **un point d'entrée unique** des données brutes de biodiversité : le SINP reçoit les données brutes et les filtre sur la base de leur qualité (vérification de la conformité, cohérence et validité scientifique des observations). Si la qualité des données brutes est validée par le SINP, celles-ci sont transmises à l'INPN (Inventaire National du Patrimoine Naturel) pour diffusion.

1.3. Problématique : l'application concrète de la loi et ses difficultés

L'application concrète/technique du dépôt légal des données brutes de biodiversité est complexe. Les maîtres d'ouvrage sont dans la quasi impossibilité de convertir seuls leurs données au format standard SINP. Sans entrer ici dans les détails (voir partie 2- Cahier des charges), un fichier respectant le standard SINP contiendra une vingtaine d'attributs obligatoires (et beaucoup d'autres optionnels) par observation. Ces informations obligatoires portent principalement sur l'observation elle-même (identifiant, dates, ...), l'identification du spécimen (par exemple la détermination de l'espèce par son nom scientifique) et sa localisation géographique. En tout, un maître d'ouvrage doit lire et retenir plus de 300 pages de documentations sur la méthodologie du standard de données SINP (voir par exemple Jomier 2017, Jomier *et al.*, 2016, 2018 ; Robert *et al.*, 2016) puis transformer les données brutes issues la plupart du temps d'un tableur classique (généré par le logiciel Excel notamment) fourni par des bureaux d'études.

En plus de ce fichier respectant le standard SINP, il faudra fournir d'autres informations sur le projet et le jeu de données en se connectant à d'autres plates-formes web pour renseigner différents types d'informations (informations sur le projet, métadonnées, jeu de données, ...). Le processus administratif complet est donc complexe et fastidieux pour le maître d'ouvrage.

Aucun outil n'est à ce jour proposé par le ministère ou le SINP pour aider les maîtres d'ouvrage / bureaux d'études à effectuer le téléversement et convertir leurs données au standard SINP.

1.4. Objectifs du stage

L'objectif de mon stage est de développer **les outils et services permettant le téléversement des données brutes de biodiversité** collectées par les maîtres d'ouvrage.

Le principal objectif est de réaliser entièrement une application web de téléversement : écriture du cahier des charges, développement côté client, développement côté serveur, réflexions sur l'UX (expérience utilisateur : ressenti de l'utilisateur lors de l'utilisation de l'application). Cette application devra permettre à

l'utilisateur de convertir rapidement et facilement des données brutes de biodiversité collectées sur le terrain au standard SINP.

Parallèlement, en attendant la mise en production de l'application de téléversement, il faudra être en mesure de fournir une **prestation pour les maîtres d'ouvrage** demandant une conversion de leurs données au format SINP. Il faudra non seulement fournir un service technique de transformation de données mais aussi accompagner le client durant tout le processus de téléversement à travers les diverses plates-formes.

De plus, il est important d'alimenter le blog du site web de NS (<http://www.natural-solutions.eu>) avec des **articles relatifs au téléversement** afin que les maîtres d'ouvrage et bureaux d'études identifient rapidement les outils que nous développons lors de leurs recherches sur le web. La rédaction de cette documentation est donc également une des missions de ce stage. Ceci permettra en outre de fournir une information simplifiée et clarifiée du processus de téléversement - ce dernier étant identifié comme un « casse-tête » par les maîtres d'ouvrage - en synthétisant les nombreuses informations dispersées, parfois contradictoires et/ou partielles (et/ou non mises à jours), disponibles sur le web.

2. Cahier des charges de l'application de téléversement

2.1. Objectif principal

Le SINP définit les données brutes issues des campagnes de terrain comme des données sources (DS). Les données converties aux normes du standard SINP sont appelées des données élémentaires d'échange (DEE). **L'objectif principal et général de l'application consiste donc à permettre à l'utilisateur (= maîtres d'ouvrage) de transformer leurs DS en DEE.**

Les DEE sont donc formatées/normalisées selon un **standard national précis, systématique et rigide**, défini et requis par le standard SINP. Chaque occurrence (observation) contenu dans le jeu de données brutes doit contenir des éléments obligatoires et facultatifs normalisés relatifs à l'observation et l'identification du taxon ainsi qu'à son regroupement et rattachement géographique.

Ces DEE doivent être contenues dans un fichier au format .csv. Chaque observations (= occurrence d'un taxon) en lignes et des attributs requis par le SINP en colonnes au format .csv. Les attributs à renseigner peuvent être classés en 2 grandes catégories : ceux liés à l'observation en elle-même et ceux liés au jeu de données en général.

D'un point de vue technique, l'application devra donc permettre au maître d'ouvrage d'obtenir un fichier csv valide, c'est-à-dire respectant les normes du standard SINP, prêt à soumis sur la plate-forme GINCO afin d'obtenir un **certificat de conformité** permettant de continuer leur projet.

Enfin, afin d'optimiser l'UX et alléger la collecte d'informations, il faudra disperser cette récolte d'informations sur 5 formulaires courts représentant **5 étapes** :

- 1- Envoi du fichier utilisateur contenant les données brutes
- 2- Informations sur l'identification taxonomique
- 3- Informations chronologiques
- 4- Informations géographiques
- 5- Informations globales

2.2. Une application simple et rapide d'utilisation

Afin de créer une application adaptée à l'utilisateur, nous avons défini un persona, c'est à dire les caractéristiques de l'utilisateur principal de l'application. En ce qui nous concerne, ce persona est un maître d'ouvrage. Il a comme caractéristiques d'être stressé par le téléversement qui représente une nouvelle contrainte importante dans son travail. Il est mal renseigné car la documentation est assez « sèche », en grande quantité et avec beaucoup de termes techniques. Il n'a pas beaucoup de temps à consacrer à cette nouvelle tâche obligatoire dans son emploi du temps et veut en être débarrassé au plus vite.

D'après notre persona, il faudra donc une application simple, c'est à dire d'une **grande facilité d'utilisation** afin d'optimiser l'UX. Ainsi, il faudra inclure uniquement des boutons/outils de navigation indispensables et **éviter la surcharge visuelle avec des informations superflues et éviter les termes techniques relatifs au standard SINP**. L'application devra donc consister en **quelques étapes rapides de collectes d'informations**

utilisateurs, avec à chaque étape un moyen de **rassurer l'utilisateur sur l'état d'avancement** de son processus de conversion de données.

2.3. *Quel niveau de détails du standard SINP pour l'application?*

Avec le standard SINP, une observation peut se caractériser à l'aide d'environ 80 attributs. Heureusement, la plupart d'entre eux sont facultatifs. Une vingtaine de champs sont obligatoires, un peu plus en fonction des attributs obligatoires conditionnels qui dépendent de la nature des données brutes. Ces attributs conditionnels sont des champs à renseigner pour une observation à condition qu'un autre champ soit présent.

Pour cette première version de l'application développée pendant mon stage, le choix a été fait de se concentrer uniquement sur les attributs obligatoires et obligatoires conditionnels. En effet, nous pouvons estimer que dans leur grande majorité, nos clients potentiels voudront en priorité se débarrasser de cette tâche. Ils ne visent donc pas un téléversement qu'on pourrait qualifier de « qualitatif », c'est à dire composé d'observations « riches » caractérisées par un grand nombre d'attributs. Leur objectif est plutôt de fournir un fichier contenant le strict minimum d'informations qui permet d'obtenir le certificat de conformité. Dans le futur, à partir de ce noyau d'attributs obligatoires, nous pourrions éventuellement travailler sur des demandes spécifiques et concrètes de clients en développant certains attributs optionnels.

2.4. *La collecte d'informations utilisateurs*

La conversion des données brutes de biodiversité des utilisateurs vers le standard SINP nécessite la collecte d'informations provenant de l'utilisateur : par l'envoi (*upload*) du fichier utilisateur contenant les données brutes de biodiversité et par des formulaires.

Les formats acceptés pour le fichier utilisateur seront les formats les plus courants : .csv, .xls, .xlsx, .txt.

Les formulaires collecteront des informations de 2 types :

- **des informations globales sur le jeu de données**, en quelques sortes les métadonnées, qui seront remplies grâce à des questions auxquelles l'utilisateur devra répondre (ex : Quel est le statut de l'organisme ayant réalisé les observations ?).
- **des informations sur les données brutes elles-mêmes** pour lesquelles on demandera à l'utilisateur de les localiser dans son document (ex : quelle colonne renseigne l'identification taxonomique des espèces en nom latin).

Certains champs sont à remplir à partir d'une liste finie d'éléments définie par le SINP. Ceci dans le but d'homogénéiser les données en contraignant les choix. **Dans ce cas l'application devra afficher la liste des possibilités à l'utilisateur** (ex : le statut de l'organisme ayant réalisé les observations est public, privé, NSP (= ne sait pas)).

Certains attributs nécessitent l'utilisation de référentiels externes au SINP pour fournir des informations supplémentaires spécifiques. Par exemple, le nom d'espèce fourni par l'utilisateur devra se trouver dans le référentiel *TaxRef* (voir partie 2-5). Dans ce dernier, à chaque nom d'espèce correspond un numéro d'espèce unique qui est demandé par le SINP. **L'application devra donc lancer automatiquement les correspondances entre les informations fournies par l'utilisateur et les informations contenues dans ces référentiels et requis par le SINP.**

D'autre part, certains attributs sont à renseigner de façon libre (pas de contrainte de choix parmi une liste de valeurs) mais dans un format contraint. Par exemple l'identité de la personne ayant réalisé l'observation doit être sous la forme suivante : Prénom NOM. **L'application devra donc guider l'utilisateur sur le format à suivre et éventuellement appliquer des corrections automatiques.** Par exemple, si le NOM est écrit en minuscule par l'utilisateur, l'application le mettra automatiquement en majuscule.

2.5. Les informations taxonomiques

Selon le standard SINP, une observation doit contenir au moins un nom ('nomCite') permettant de renseigner l'identification taxonomique. Ce nom doit être présent dans le référentiel TaxRef.

TaxRef (Gargominy et al., 2017) est le référentiel taxonomique national pour la faune, la flore et la fonge de France, élaboré et géré par le Muséum national d'Histoire naturelle dans le cadre du SINP. TaxRef a pour but de fournir une liste exhaustive de tous les noms scientifiques de l'ensemble des êtres vivants recensés sur le territoire français. Il contient actuellement environ 550000 entrées (taxons) et 77 attributs permettant de caractériser le nom scientifique de chaque organisme sur l'ensemble des niveaux taxonomiques.

Dans le cadre du standard SINP, trois attributs de TaxRef sont obligatoires : LB_NOM, CD_NOM et CD_REF. Le LB_NOM est le nom scientifique (en latin) du taxon. Le CD_NOM est un champ de type entier qui correspond à un identifiant unique du nom scientifique. C'est la clé primaire de TaxRef. Un nom latin peut avoir plusieurs CD_NOM. CD_REF est un champ de type entier qui représente l'identifiant du taxon de référence, c'est à dire le nom scientifique valide actuellement. Un CD_REF doit obligatoirement correspondre à un CD_NOM et peut correspondre à plusieurs CD_NOM si le nom scientifique a évolué dans le temps par exemple (clé étrangère).

Pour notre application, l'utilisateur n'aura pas à fournir le CD_NOM et le CD_REF. Il faudra simplement qu'il indique le nom de la colonne contenant un nom latin valide, c'est à dire présent dans TaxRef. Si c'est le cas, l'application devra générer automatiquement le CD_NOM et le CD_REF associés au LB_NOM.

2.6. Les informations chronologiques

Le standard SINP requiert au moins une date de début ('dateDebut') et une date de fin ('dateFin') pour chaque observation au format 'jj/mm/aaaa'. **L'application devra donc permettre à l'utilisateur d'indiquer le nom des 2 colonnes contenant ces 2 dates.** Dans un premier temps, l'application devra fournir la possibilité à l'utilisateur de renseigner la date sous la forme jj/mm/aaaa et jj-mm-aaaa. Dans l'éventualité où les dates sont identiques pour toutes les observations, l'utilisateur pourra indiquer la date de début et la date de fin directement dans l'application avec un calendrier. Enfin, **si la date de fin est identique à la date de début, l'utilisateur pourra indiquer la date une seule fois.**

2.7. Les informations géographiques

Le standard SINP requiert au moins une information de géolocalisation pour chaque observation. Une information géographique SINP peut être de 2 types :

- des coordonnées géographiques : ces coordonnées doivent désigner soit le point précis de l'observation (un couple longitude/latitude), soit une zone géographique approximative dans laquelle le taxon observé se

situé (zone définie par plusieurs couples longitude/latitude). Ces coordonnées doivent être fournies au format WKT (*Well-Known Text*), standard courant permettant de représenter des objets géographiques (point, ligne, polygone).

- si aucune information géographique précise n'est disponible, il faut rattacher l'observation à une commune et/ou un département. Les informations concernant la commune (nom de commune et numéro de commune ; 'nomCom' et 'cdCom') et le numéro de département ('cdDept') doivent correspondre à des normes officielles disponibles dans le **référentiel de l'INSEE (Code Officiel Géographique au 1^{er} janvier 2018)**.

Pour l'application, l'utilisateur devra avoir la possibilité de fournir un des 2 types d'information géographique requis par le SINP ou les 2 types à la fois. Dans un premier temps, l'application ne gèrera que les coordonnées géographiques désignant un point précis avec des longitude/latitude en degré décimal. D'après l'expérience de Natural Solutions concernant les données de biodiversité, les clients potentiels fourniront pour la grande majorité ce type de coordonnées. Des développements plus spécifiques (gérer différents formats géographiques) seront faits en cas de demandes explicites de clients.

Pour les informations concernant les communes et départements, l'application devra permettre de croiser les données fournies par l'utilisateur avec le référentiel de l'INSEE. Ainsi, l'utilisateur aura 3 choix :

- fournir un nom de commune : l'application générera automatiquement le numéro de commune et le numéro de département si le nom de commune existe dans le référentiel INSEE.
- fournir un numéro de commune : l'application générera automatiquement le nom de commune et le département si le numéro de commune existe dans le référentiel INSEE.
- fournir le numéro de département : dans ce cas l'application vérifiera uniquement si le numéro de département existe dans le référentiel INSEE.

2.8. *Les informations globales*

Le standard SINP nécessite l'inclusion de plusieurs champs obligatoires renseignant le jeu de données en lui-même, les métadonnées. Pour la majeure partie d'entre eux, ces valeurs sont identiques pour toutes les observations.

Pour trois de ces attributs, l'utilisateur doit choisir une valeur parmi une liste finie d'éléments. Dans ce cas, il faudra que l'application fournisse la liste de choix. Il s'agit des champs renseignant le type de l'observation ('statObs' ; directe/indirecte), le statut de l'observation ('dSPublique' ; public/privé), le statut de la source ('statSource' ; donnée de terrain/collection/littérature).

Trois de ces attributs sont libres, c'est-à-dire que l'utilisateur doit taper lui-même la valeur. Il s'agit de l'organisme gestionnaire de données ('orgGestDat'), l'organisme ayant effectué les observations ('obsNomOrg'), l'identité de la personne ayant réalisé les observations ('obsId'). Le formulaire de l'application devra donc permettre d'entrer ces données.

2.9. *Les identifiants de l'observation*

Le standard SINP requiert l'inclusion d'un identifiant unique ('permId') pour chaque observation de type UUID (Universal Unique Identifier ; groupes de caractères hexadécimaux codés sur 128 bits dont le nombre de combinaison possible est 10^{38}). **L'application devra générer automatiquement ces UUID pour chaque**

observation. Elle devra quand-même fournir à l'utilisateur la possibilité de fournir ses propres UUID s'il les a générés au préalable.

D'autre part, il semble important d'offrir la possibilité à l'utilisateur d'**inclure un attribut facultatif pour le standard SINP qui est l'identifiant d'origine de l'observation attribué au moment de l'identification du spécimen** ('idOrigine'). L'application devra donc donner la possibilité à l'utilisateur de fournir cet identifiant s'il en possède un.

2.10. Format du fichier final à fournir à l'utilisateur

Le fichier final au standard SINP à fournir à l'utilisateur doit être au format **csv** avec comme séparateur le point-virgule « ; » et **encodage en UTF-8**. La première ligne comporte le nom des attributs attendus par le standard SINP. L'ordre des attributs (= colonnes) dans le fichier n'a pas d'importance. Les champs obligatoires doivent être présents et non-vides. Si un attribut contient plusieurs valeurs, il faut les séparer par des virgules. Les décimales sont représentées par un point.

2.11. Gestion des erreurs utilisateurs

L'application devra détecter les erreurs utilisateurs, l'en informer et lui donner la possibilité de les corriger.

2.11.1. Erreurs de format de fichier utilisateur

Si le format de fichier uploadé par l'utilisateur n'est pas .csv, .xlsx, .xls ou .txt, un message stipulera à l'utilisateur que le fichier uploadé n'a pas un format correct et qu'il doit soumettre à nouveau un fichier à un format valide pour l'application.

2.11.2. Vérifications préliminaires

Si le format du fichier uploadé par l'utilisateur est correct, une série de vérifications préliminaires sera exécutée :

- **Détection de fichier vide** ou ne possédant que les titres de colonnes. Si c'est le cas, un message indiquera à l'utilisateur que son fichier est vide et qu'on ne peut pas continuer le processus.
- **Détection de titres de colonnes** : si l'application ne détecte pas de titres de colonnes, l'utilisateur sera invité à corriger son fichier en ajoutant les titres et recommencer le processus.
- **Détection et correction de doublon de ligne** : si l'application détecte un doublon de ligne, celui-ci sera automatiquement corrigé en effaçant en ne gardant que le premier. L'erreur ne sera pas bloquante et un message indiquera à l'utilisateur que des doublons ont été détectés et corrigés.
- **Détection et correction d'espaces éventuels** au début et à la fin de chaque champ du fichier. L'utilisateur n'est pas informé de cette correction. Cette correction permettra de réduire les possibilités d'erreurs utilisateurs, notamment pour les champs à choisir parmi une liste finie (ex : si le champ attend une valeur égale à 'pr', une entrée utilisateur de type ' pr' ne générera pas d'erreur).

2.11.3. Erreurs de conformité et cohérence

Définitions

Les erreurs de conformité correspondent aux :

- champs pour lesquels la valeur n'appartient pas à une liste de valeurs attendues par le standard SINP, que ce soit une liste définie par le SINP ou un référentiel.
- champs libres pour lesquels le format n'est pas respecté par l'utilisateur. Par exemple, l'identité de l'observateur doit être formée du prénom et du nom mais l'utilisateur n'a fourni qu'un seul des deux.
- valeurs manquantes alors que le champ est obligatoire pour le SINP.

Les erreurs de cohérence correspondent à des valeurs impossibles. Par exemple, une longitude ne peut pas avoir une valeur supérieure à 180.

Pour chaque étape (= chaque formulaire), les erreurs seront listées à l'utilisateur et il sera invité à les corriger directement avec l'application, sans retourner à son fichier original. La détection des erreurs est lancée pour chaque observation (= chaque ligne) indépendamment.

Erreurs liées à l'identification taxonomique

Une erreur de conformité sera signalée à l'utilisateur si l'appellation taxonomique est non valable, c'est à dire manquante, pas en nom latin ou absent de TaxRef.

Erreurs liées aux informations chronologiques

Une erreur de conformité sera signalée si une date de début ou de fin est manquante ou n'est pas au format attendu.

Une erreur de cohérence sera détectée si la date de fin est inférieure à la date de début.

Erreurs liées aux informations géographiques

Une erreur de conformité sera signalée à l'utilisateur si aucune information géographique n'est disponible (coordonnées géographique ou nom/numéro de commune ou numéro de département). Les erreurs de conformité et de cohérence seront qualifiées de 'WARNING' si d'autres données géographiques sont manquantes et 'ERROR' si aucune autre donnée n'est fournie.

Pour les coordonnées géographiques :

- Une erreur de conformité sera indiquée si la longitude ou latitude est manquante (si pas d'information de commune ou département) ou si la valeur est non-numérique (type caractère).
- Une erreur de cohérence sera indiquée si la valeur de latitude est inférieure à -90 et supérieure à 90, ou si la valeur longitude est inférieure à 180 ou supérieure à 180.

Pour les valeurs du référentiels INSEE :

- nom de commune : une erreur de conformité sera pointée si le nom de commune n'existe pas (absent du référentiel INSEE), s'il n'est pas de type caractère. L'utilisateur sera informé que le champ est manquant et

qu'il doit obligatoirement le corriger si les coordonnées, le numéro de commune et le département sont également manquants.

- numéro de commune : même vérifications que pour nom de commune mis à part que l'erreur se déclenche si la valeur n'est pas de type nombre entier. La donnée sera qualifiée de manquante si aucune autre donnée géographique n'est présente.

- numéro de département : même vérifications que pour le numéro de commune.

Erreurs liées aux informations globales :

Une erreur de conformité sera envoyée à l'utilisateur si un des attributs liés aux métadonnées est manquant. Pour l'identité de la personne ayant réalisé l'observation, il faudra veiller à ce qu'elle soit au format requis par le SINP (Prénom NOM).

2.12. Téléchargement du fichier final au standard SINP

L'application devra permettre à l'utilisateur de télécharger son fichier final au standard SINP à l'issue du processus.

3. Travail effectué

3.1. Méthodologie de travail

Le développement de l'application de téléversement était un peu particulier par rapport aux autres projets de Natural Solutions dans le sens où il n'émanait pas d'une commande. L'objectif final étant de vendre l'application, les *sprints* n'étaient donc pas rythmés par une interaction avec le client. L'objectif du stage était d'avoir une application prête à mettre en production sur un des serveurs loués par Natural Solutions pour proposer une version de démo en ligne à des entreprises intéressées. Il fallait donc créer une application de téléversement opérationnelle de bout en bout (de l'importation du fichier utilisateur au téléchargement du fichier standard SINP par l'utilisateur) le plus rapidement possible, en s'offrant la possibilité d'ajouter des fonctionnalités supplémentaires ensuite. Nous avons donc fixé nous-même entièrement le cahier des charges dicté en grande partie par les spécificités techniques à suivre pour obtenir un standard SINP valide à partir des attributs obligatoires et obligatoires conditionnels uniquement. Après avoir fixé les grandes lignes techniques du développement (technologies utilisées, architecture de l'application, stockage des données) avec mon chef de projet, nous avons défini des objectifs prioritaires à atteindre et les avons déclinés au sein de chaque *sprint*.

3.2. L'application de téléversement

3.2.1. Documentation et définition des spécificités techniques du standard SINP

Une première partie importante de mon travail a été la **documentation et compréhension du standard de données attendu par le SINP**. La documentation ayant la caractéristique d'être à la fois très volumineuse (plus de 300 pages), assez technique, dispersée (multitude de sources) et contradictoire (notamment en raison de versions différentes du standard). De plus, en raison de la rapidité d'application de la loi sur le téléversement, les aspects théoriques du standard ne sont pas au même stade d'avancement que l'implémentation de la plate-forme GINCO gérant la vérification de la validité des fichiers envoyés par les utilisateurs. Certaines caractéristiques de la version la plus récente du standard SINP ne sont donc pas acceptées par la plate-forme GINCO.

J'ai donc rédigé dans un premier temps une documentation technique, sorte de cahier des charges, la plus précise possible des éléments indispensables pour constituer un document valide au standard SINP. Lorsque nous avons eu accès en juin à une plate-forme 'test' en ligne, j'ai pu réaliser les ajustements entre théorie et pratique du standard SINP. J'ai également assisté à une journée de formation/information sur le standard SINP organisée par le Muséum national d'Histoire naturelle à Paris. Au final, **j'ai pu isoler les éléments obligatoires à fournir par les clients pour pouvoir convertir leurs données brutes de biodiversité et lister les règles de conformité et cohérence à respecter pour chaque attribut**.

3.2.2. Le choix des technologies

Le choix des technologies adéquates pour le développement d'une application est crucial afin d'assurer la stabilité et l'évolutivité de l'application. Le choix du langage **Python côté serveur**, un langage interprété

orienté objet créé en 1989, s'est fait naturellement puisque c'est un des langages de prédilection des développeurs et *data scientists* pour manipuler/modifier/nettoyer des données. Du fait de sa large diffusion chez les développeurs et scientifiques et de son ancienneté, Python possède un grand nombre de bibliothèques et de très bonnes performances pour la manipulation de données massives. Nous avons particulièrement utilisé la **bibliothèque 'Pandas'** pour la manipulation des données, la lecture et l'écriture de données vers différents formats et la jointure d'un large volume de données. De plus, Python est le langage privilégié au sein de Natural Solutions, notamment pour l'application *ecoRelevé* qui gère des données naturalistes relevées sur le terrain. C'est un langage extensible pouvant s'interfacer à l'aide de modules spécifiques à des librairies R. Python possède également des frameworks permettant le développement web. Ici, nous utiliserons le **micro-framework web Flask** qui fournit un ensemble d'outils web de base (gestion des routes, contrôleur, ...) et des extensions à activer et installer au fur et à mesure des besoins. Ce framework Python est léger et parfaitement adapté aux « petites » applications.

Côté client, nous avons choisi le **framework Angular** car l'application de téléversement est grandement susceptible d'évoluer. Angular fonctionne en effet sur un système de briques (= les composants : assemblage de fichiers HTML/CSS et langage TypeScript gérant l'affichage du HTML) agencées de façon très flexible et réutilisables. L'autre avantage d'Angular aux vues de notre application est le module 'service' qui permet notamment de récupérer certaines informations envoyées par le serveur et les garder en stock tout au long de la navigation sur les différentes pages de l'application. Autre atout d'Angular, il adapte et étend le HTML traditionnel pour servir le contenu dynamique de façon améliorée grâce à un data-binding bidirectionnel permettant la synchronisation automatique des modèles et des vues. En conséquence, Angular diminue l'importance des manipulations DOM et améliore la testabilité du code.

3.2.3. Le choix de l'architecture et de la communication client/serveur

Pour l'architecture de l'application nous nous sommes inspirés de la structure classique en **Model/View/Controller**. Les vues sont générées côté client avec Angular : les composants Angular, par l'intermédiaire du langage TypeScript gèrent l'interface utilisateur (HTML et CSS) qui consiste notamment à remplir plusieurs formulaires et envoyer les données issues des formulaires vers le côté serveur et recevoir les réponses du serveur. Les requêtes HTTP envoyées côté client par l'intermédiaire d'une URI dynamique (la ressource est identifiée par le nom du fichier utilisateur) sont reçues côté serveur par les routes correspondantes prédéfinies dans Flask. Le contrôleur associé à la route envoie le traitement des données à effectuer au modèle.

La communication client/serveur se base sur les URI (Uniform Resource Identifier) afin d'identifier la ressource (fichier client). Les verbes HTTP sont utilisés comme identifiant des opérations. GET et POST sont utilisés dans l'ensemble de l'application (ex : GET pour la requête des noms de colonne du fichier client, POST pour l'envoi d'un formulaire). Pour les corrections des erreurs utilisateurs par l'utilisateur, PUT/PATCH et DELETE seront utilisés pour modifier ou supprimer des ressources. La communication entre côté client et serveur se fait par l'intermédiaire du format JSON.

3.2.4. L'application côté client

Côté client, **nous avons organisé l'application en 7 écrans/vues dont l'affichage est géré par 7 composants Angular** : un écran d'accueil, puis 5 étapes de collecte des données utilisateurs, et enfin une page permettant au client de télécharger son fichier. La première étape permet au client d'envoyer son fichier de données brutes et les 4 suivantes sont des formulaires. Etant donné que nous allons offrir à l'utilisateur la possibilité

de corriger les erreurs de son fichier directement dans l'application au moment de chaque formulaire, nous avons ordonné les formulaires de façon à ce que ceux qui génèrent potentiellement le plus d'erreurs utilisateurs soient en premier. Ceci permet de limiter la frustration utilisateur en évitant au maximum qu'il se retrouve dans la situation d'avoir passé du temps à remplir des formulaires et de se rendre compte à la fin du processus qu'il doit recommencer tout depuis le départ.

Nous avons mis en place des **services Angular** pour accéder depuis n'importe quel composant aux différentes fonctions définissant les requêtes HTTP et aux observables liées aux requêtes HTTP. Ces observables (package RxJS) sont des objets qui vont émettre des événements auxquels on va pouvoir souscrire au sein de chaque composant. Par exemple, la requête HTTP pour obtenir les titres des colonnes du fichier utilisateur va créer un observable auquel on va souscrire dans chaque composant. Ceci évite notamment d'envoyer une requête HTTP pour chaque formulaire pour lesquels les titres de colonnes sont nécessaires. L'affichage des différents composants est assuré par un routing géré également par l'intermédiaire d'un service. Nous avons défini un routing par défaut permettant à l'utilisateur d'être redirigé vers la page d'accueil s'il tape manuellement une route non définie par les composants.

Les formulaires sont construits avec la **méthode 'reactive form'** de Angular. Cette méthode utilise également les observables. Dans ce cas, nous créons un observateur pour chaque valeur du formulaire (= souscription à la valeur). Chaque fois que l'utilisateur entre ou modifie une valeur du formulaire, celui-ci émet un événement intercepté directement par l'observateur, ce qui provoque une notification immédiate de tous les changements de valeur. Cette méthode permet également de définir des règles de validation côté client. Les erreurs clients sont ainsi limitées en le guidant « en direct » sur le respect du format voulu pour une valeur donnée du formulaire.

L'affichage des erreurs utilisateurs de conformité et de cohérence se fait par l'intermédiaire d'une grille HTTP générée par l'outil **ag-Grid**. Ag-Grid est une grille JavaScript utilisable dans n'importe quel framework et possédant de nombreuses options : celles qui nous intéressent ici sont les possibilités de sélectionner les lignes et modifier des champs. Cette partie est encore en cours de développement lors de l'écriture de ce rapport mais sera terminée à la fin du stage.

3.2.5. Application côté serveur

Le traitement des requêtes HTTP

Nous avons structuré l'application côté serveur de façon à séparer au maximum les tâches. Ainsi les routes côté serveur (contrôleurs) réceptionnent les requêtes et envoient les manipulations de données au modèle. Le modèle est formé de plusieurs scripts qui séparent les différents types de traitement des données : ceux permettant de définir des « setters » qui initialisent des variables à partir des données envoyées par le formulaire, ceux initialisant les variables issues de jointures avec les référentiels, ceux vérifiant la conformité et cohérence des valeurs utilisateurs. Cette organisation permet de faciliter la mise à jour de l'application si une modification externe a lieu (changements/mises à jour dans un des référentiels ou dans le standard SINP). Ainsi, si certaines règles de conformité et cohérences sont modifiées dans de futures versions du standard SINP, seule la partie du modèle gérant la définition des erreurs sera impactée et devra être modifiée. De plus, la séparation des types de traitement permet de gagner en clarté concernant le code, à la fois en cas de bug (plus grande facilité à repérer les sources d'erreurs) et en cas de développements ultérieurs de l'application par d'autres développeurs.

Écriture du code Python et bonnes pratiques

Dans un souci de lisibilité et compréhension du code, j'ai adopté certaines règles de bonnes pratiques dont la plupart sont listées dans le PEP8 (guide de style pour le code Python). Ceci consiste notamment à :

- séparer par des tirets bas ('_') les mots dans les noms des fonctions et variables
- faire commencer par une majuscule les noms de classe
- faire débiter les noms de fonction par des verbes (ex : « get_... », « set_... », « correct_... ») pour signifier l'action
- faire débiter par « is_... » les noms de fonctions ou variables de types booléen (ex : is_latitude)
- respect des espaces/non-espaces selon les divers cas listés par le PEP8

J'ai essayé au maximum de séparer toutes les actions en fonctions distinctes pour améliorer la compréhension du code et repérer plus facilement les sources d'erreur.

Gestion des données

Les données brutes de biodiversité sont acquises par l'envoi du fichier utilisateur (HTTP POST formé d'un objet FormData contenant le type File) au départ du processus de l'application. A partir de là, deux solutions étaient possibles : créer une base de données et y insérer les données utilisateur ou **traiter directement les fichiers utilisateurs en les stockant temporairement dans un répertoire dédié côté serveur**. Notre choix s'est porté sur la deuxième solution pour plusieurs raisons :

- pour le moment, il n'y a pas d'enjeu de sauvegarde des données utilisateurs.
- pour le moment, il n'y a pas d'enjeu de traitement de données massives, le nombre de données brutes de biodiversité contenues dans chaque fichier étant de l'ordre d'une dizaine de milliers maximum. Les bibliothèques Python dédiés à la manipulation de données, notamment Pandas, garderont donc une très bonne performance pour le traitement des données.
- pour le moment, le nombre d'utilisateurs de l'application sera assez limité (en comparaison avec les applications grand public) puisqu'il s'agira des maîtres d'ouvrage travaillant sur des projets impactant l'environnement ou des bureaux d'études dans le cadre d'études d'impact.
- l'objectif étant de produire rapidement une version fonctionnelle (de démonstration) de l'application, il était plus rapide et facile de travailler directement avec les fichiers utilisateurs.

Le fichier utilisateur est donc enregistré côté serveur dans un répertoire './upload'. Afin d'éviter les doublons si deux utilisateurs travaillent en même temps sur un fichier portant le même nom, un **UUID est généré et accolé en suffixe au nom original du fichier** ('originalName_UUID'). A la suite de la série de vérifications préliminaires et prétraitements des données originales, un fichier csv est enregistré contenant les éventuelles modifications en accolant le suffixe '_pretreated' ('originalName_UUID_pretreated'). Le travail de conversion des données se fera à partir des données contenues dans ce second fichier. Ensuite un troisième fichier csv contenant les données transformées au standard SINP sera enregistré en accolant le suffixe ('_SINP'). Les colonnes seront ajoutées dans ce fichier au fur et à mesure de l'avancée de l'utilisateur sur les différents formulaires. Les noms des fichiers clients sont passés comme paramètre via des URL dynamiques entre le

côté client et le côté serveur. A la fin du processus ou si l'utilisateur décide de recommencer le processus au cours des étapes que nous avons définies, **les données du fichier client sont effacées du serveur.**

Gestion des erreurs

Les erreurs sont gérées de 3 façons côté serveur :

- Pour l'import des données, **une exception est levée et retournée avec le code d'état 400 si le fichier utilisateur ne respecte pas le format attendu ou comporte des erreurs bloquantes** (fichier vide, absence de titres de colonnes, absence de structuration correcte). Un message est visible pour l'utilisateur sur l'interface graphique lui signifiant que son fichier n'est pas valide (accompagné d'une courte description de la source de l'erreur) et qu'il doit le corriger puis recommencer le processus.

- **Pour les erreurs de conformité et de cohérence du fichier utilisateur (non-respect des normes du standard SINP), les erreurs sont remontées côté client dans l'interface graphique** et présentée avec ag-Grid (pour informer l'utilisateur et lui donner la possibilité de corriger) mais sans lever d'exception. Pour permettre cela, lors du prétraitement du fichier et de l'initialisation des variables utilisateur à partir des données de formulaire, les variables sont systématiquement converties en type caractère. Par exemple, dans le cas d'un champ utilisé pour une jointure, si l'utilisateur fournit une variable originale de type entier, la jointure (qui attend une valeur de type caractère) ne va pas générer d'erreur interne du serveur (car l'entier aura préalablement été converti en caractère). C'est ensuite dans un second temps, lorsque la jointure aura été exécutée que la partie du modèle gérant la détection des erreurs utilisateurs va indiquer que le champ contient une erreur car il est formé d'une variable de type entier et non d'une variable de type caractère. Cette approche permet de simplifier considérablement la gestion des erreurs côté serveur en ne se souciant plus des différents types de variables fournis par l'utilisateur au moment des divers traitements des données.

- **Une erreur est levée et retournée avec le code d'état 500 en cas d'erreur interne du serveur.**

3.2.6. La correspondance des données utilisateurs avec les référentiels

Pour établir les correspondances entre les données utilisateurs et les informations contenues dans les référentiels, j'ai utilisé des **jointures externes**. Ces jointures externes permettent de retourner tous les enregistrements du fichier utilisateur avec leur correspondance dans le référentiel quand elle existe, sinon la valeur nulle est retournée.

La performance en temps d'exécution des jointures est un élément important à prendre en compte pour la satisfaction utilisateur. Python et R proposent 2 outils, respectivement la bibliothèque 'Pandas' et le package 'dplyr', dans lesquels les temps de jointures sont présentés comme excellents, comparables à ceux de Spark ou PostgreSQL. J'ai donc comparé la performance de ces 2 outils en exécutant une jointure entre un fichier de données brutes (environ 1135 lignes et 77 colonnes) et le référentiel TaxRef allégé (environ 550000 entrées et 3 colonnes). Les performances du package 'dplyr' sont meilleures, avec une jointure quasi-instantanée. Avec Pandas, la jointure se déroule pendant 1 à 2 secondes. Cependant, l'installation de Python et R côté serveur, avec l'utilisation d'un module Python permettant de lancer des fonctions R depuis le contrôleur rajoutent de la complexité, et font diminuer le gain de temps des jointures sous 'dplyr'. J'ai donc décidé d'utiliser la **bibliothèque 'Pandas' pour effectuer les jointures.**

Le référentiel TaxRef

Pour rechercher les identifiants TaxRef (colonnes 'CD_NOM' et 'CD_REF'), les jointures sont exécutées sur la colonne du fichier utilisateur décrivant les noms latins des taxons et la colonne 'LB_NOM' de TaxRef. Lorsque la jointure est lancée, 3 cas se présentent :

- la correspondance du nom latin entre les 2 fichiers existe, nous pouvons enregistrer les identifiants TaxRef ('cdNom' et 'cdRef').
- la correspondance n'existe pas, une erreur de cohérence est renvoyée à l'utilisateur pour qu'il corrige le nom latin.
- le nom latin indiqué par l'utilisateur correspond à plusieurs entrées dans TaxRef provoquant en conséquence des doublons d'observations dans le fichier utilisateur. Dans ce cas, nous gardons uniquement la ligne pour laquelle le 'CD_NOM' est égal au 'CD_REF' (c'est-à-dire le nom de taxon faisant référence actuellement).

Le référentiel INSEE

Lorsque l'utilisateur sélectionne le nom de commune dans le formulaire géographique, la jointure est effectuée avec la colonne 'NCCENR' du référentiel. Ceci permet à la fois de vérifier la conformité des noms de commune de l'utilisateur et d'ajouter les numéros de communes et les numéros de département correspondant dans le fichier utilisateur standardisé. Lorsque l'utilisateur sélectionne le numéro de commune, la jointure est effectuée sur la colonne correspondant à la concaténation des valeurs de la colonne 'DEP' et 'COM' du référentiel INSEE. Les noms de commune et les numéros de département correspondants sont ajoutés au fichier utilisateur. Lorsque l'utilisateur sélectionne le numéro de département dans le formulaire géographique, aucune jointure n'est effectuée, seule une vérification de l'existence du numéro de département dans le référentiel est lancée.

Les erreurs de jointure éventuelles correspondent à un nom de commune, numéro de commune ou numéro de département de l'observation fournit par l'utilisateur qui n'existe pas. Dans ce cas, une erreur bloquante est remontée à l'utilisateur pour qu'il la corrige si l'observation ne comporte pas également de coordonnées géographiques. Si l'observation possède des coordonnées géographiques, l'erreur est seulement signalée ('warning') à l'utilisateur.

3.2.7. Design et intégration

Naomi Fischer, web designer et intégratrice, a réalisé les maquettes pour le design de l'application et nous avons travaillé ensemble sur l'intégration, avec le framework Bootstrap, pour me former à cet aspect du développement.

Le design de l'application devait respecter trois contraintes :

- en ce qui concerne l'UX, l'application devait suivre un design simple et épuré pour que l'utilisateur identifie et comprenne immédiatement les éléments à renseigner et profite d'une navigation simple (minimiser les options et nombre de boutons).
- suivre dans les grandes lignes le design de l'application ecoRelevé. Ce dernier permet de relever des données de biodiversité sur le terrain et notamment d'exporter un fichier csv au format SINP en un clic. Notre application est donc un complément d'ecoRelevé en ce qui concerne le téléversement puisqu'elle permet de convertir des données utilisateurs générées sur le terrain sans l'aide d'ecoRelevé (donc brutes et sans aucun

formatage). Pour une meilleure identification de la gamme des outils proposés par NS, un design proche de l'application de téléversement et d'ecoRelevé était donc nécessaire.

- faciliter un design *responsive*, c'est à dire adapté à plusieurs formats d'écrans (dont téléphone mobile).

3.3. Autres activités

3.3.1. Prestations de téléversement

Parallèlement au développement de l'application, l'entreprise Neoen nous a contactés pour effectuer le téléversement de données brutes de biodiversité issues d'une étude d'impact réalisée pour un de leur projet d'implantation d'éoliennes. Cette prestation comportait 2 aspects, d'une part la conversion de 5 fichiers de données brutes (environ 1500 observations) au format SINP et d'autre part un accompagnement à travers l'ensemble du processus administratif complexe de téléversement sur les différentes plates-formes Web. Etant donné que notre application était en cours de développement, je ne pouvais pas l'utiliser pour cette prestation. Les délais étant courts pour cette mission, j'ai écrit des scripts en langage R pour convertir les données au format SINP parce que c'était le langage que je maîtrisais le mieux à ce moment-là.

Neoen nous a contactés quelques semaines plus tard pour une seconde prestation pour laquelle, après avoir réalisé le devis, j'ai appliqué la même procédure que pour le premier contrat.

3.3.2. Rendez-vous avec clients potentiels

Pendant mon stage, j'ai participé à 3 rendez-vous avec des clients potentiellement intéressés par une application de téléversement : le groupe Eiffage à Paris, Biotope à Montpellier et Audiccé que nous avons rencontré à Marseille dans nos locaux. Au cours de ces rendez-vous, j'ai présenté le principe général du téléversement, notre projet d'application et une démonstration d'ecoRelevé en ligne.

3.3.3. Rédaction de documents

J'ai réalisé 2 articles sur le blog du site web de Natural Solutions. Un article visait à expliquer rapidement, en quelques phrases simples, le principe du téléversement et les services que nous proposons dans ce domaine. L'objectif était que les entreprises intéressées par le téléversement identifient clairement et rapidement ce qu'on pouvait leur apporter. Un autre article, incluant plus de détails techniques, avait pour objectif de synthétiser, grouper et simplifier les informations disponibles sur différentes sources web. Ces deux articles permettent aussi un référencement Google de Natural Solutions pour les entreprises cherchant des informations sur le téléversement. Plusieurs entreprises nous ont d'ailleurs contactés après avoir lu les articles.

J'ai écrit également un document à usage interne sur les spécificités techniques du standard SINP.

4. Résultats

4.1. Ecran d'accueil

La page d'accueil résume les étapes que l'utilisateur va devoir suivre pour convertir ses données (Figure 1). Pour économiser au maximum le temps de l'utilisateur, chaque étape est résumée en un mot. L'utilisateur n'a pas d'autre action à effectuer que de cliquer sur le bouton « Commencer ».



Figure 1 : Ecran d'accueil

4.2. Ecran d'importation du fichier utilisateur

Cette page permet à l'utilisateur d'envoyer (*upload*) son fichier de données brutes de biodiversité. L'application ne donne que deux choix à l'utilisateur lors de son arrivée sur cet écran : le bouton « choisir un fichier » ou le bouton « retour à l'accueil » (Figure 2). Si l'utilisateur survole la lettre i dans un cercle gris à côté du bouton « choisir un fichier » il obtient des informations sur les formats valides pour le téléchargement de son fichier.

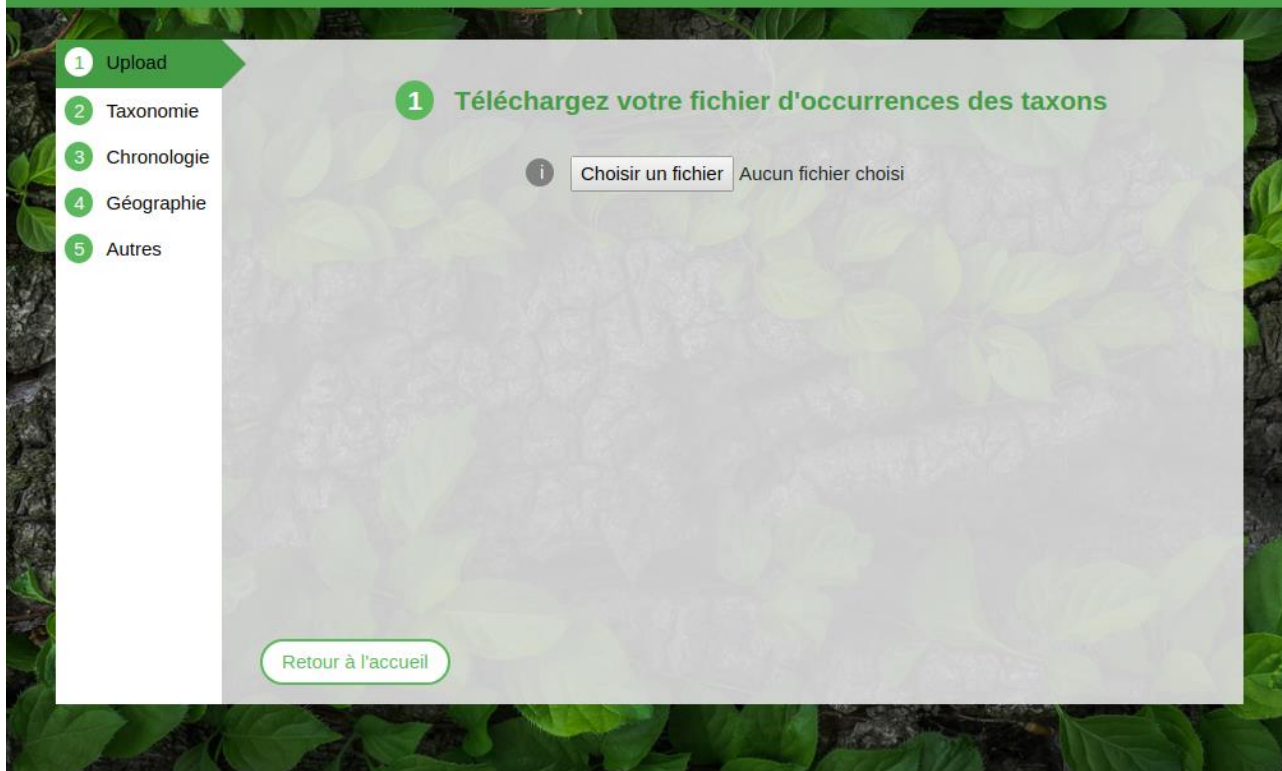


Figure 2 : Importation du fichier utilisateur

Une fois le fichier sélectionné, l'utilisateur peut cliquer sur le bouton « lancer les vérifications préliminaires » (Figure 3) qui va entraîner les actions suivantes :

- vérification du format du fichier client
- enregistrement du fichier client sur le serveur
- lecture des données clients (sous la forme d'un *dataframe* Pandas)
- encodage de tous les champs du fichier en utf-8
- vérification que le fichier client n'est pas vide
- vérification que le fichier client possède des titres de colonnes
- vérification et éventuelles corrections de doublons de lignes
- éventuelles suppressions et corrections d'espaces superflus en début et fin de chaque champ



Figure 3 : Vérifications préliminaires du fichier utilisateur



Figure 4 : Affichage d'une erreur utilisateur lors de l'envoi du fichier

Les vérifications sont listées côté client avec un point vert pour chaque action vérifiée et validée. Si tous les points vérifiés sont validés, le bouton « étape suivante » apparaît.

Si une erreur au cours de ces vérifications, la description de l'erreur est affichée côté client. L'application lui indique de corriger l'erreur et de recommencer avec le bouton « recommencer le téléchargement » (Figure 4).

4.3. Ecran d'identification taxonomique

L'utilisateur atteint sur cette page le premier formulaire dont le but est d'informer sur la détermination taxonomique et générer les identifiants TaxRef (Figure 5). Par l'intermédiaire d'un menu déroulant, l'utilisateur doit sélectionner le titre de la colonne de son fichier indiquant le nom latin des taxons. Le bouton « étape suivante » est désactivé tant que l'utilisateur n'a pas renseigné les noms latins car c'est une information indispensable pour générer les identifiants SINP.

L'utilisateur peut indiquer éventuellement les colonnes de son fichier contenant les identifiants TaxRef s'il les a déjà générés lui-même. A priori, ce cas va être très minoritaire donc le formulaire est pré-rempli avec 'non' à la question de savoir si l'utilisateur possède déjà des identifiants TaxRef.

The screenshot shows a web application interface for 'Téléversement SINP'. The main content area is titled '2 Identification taxonomique'. It contains the following elements:

- A sidebar on the left with five steps: 1 Upload, 2 Taxonomie (highlighted), 3 Chronologie, 4 Géographie, and 5 Autres.
- A heading: 'Veillez indiquer la colonne d'identification taxonomique en noms latins'.
- A dropdown menu labeled 'Nom latin'.
- A question: 'Avez-vous une colonne correspondant au 'CDNOM' du référentiel TaxRef?' with radio buttons for 'Oui' and 'Non' (selected).
- A second question: 'Avez-vous une colonne correspondant au 'CDREF' du référentiel TaxRef?' with radio buttons for 'Oui' and 'Non' (selected).
- At the bottom, there are two buttons: 'Recommencer le processus' and 'Étape suivante'.

Figure 5 : Formulaire d'identification taxonomique

Lorsque l'utilisateur clique sur « étape suivante », les informations du formulaire sont envoyées au serveur. Si l'utilisateur a indiqué qu'il ne possède pas d'identifiant TaxRef, ceux-ci sont générés à partir des noms latins. La cohérence et conformité des informations (voir Partie 2 – Cahier des charges) est ensuite évaluée

et les erreurs définies et repérées précisément. Les lignes du fichier utilisateur comportant des erreurs sont renvoyées côté client sous forme d'un tableau ag-Grid (cette dernière étape est encore en cours de développement et sera terminée d'ici la fin du stage ; pour le moment, les erreurs utilisateur sont reportées dans le fichier SINP). L'utilisateur va pouvoir corriger l'erreur (modifier le champ ou supprimer l'observation entière) directement dans ag-Grid puis soumettre une nouvelle fois le formulaire en cliquant sur « étape suivante ». Côté serveur, le processus recommence jusqu'à ce que le fichier du client ne contienne plus d'erreur.

Lorsque le fichier utilisateur ne contient plus d'erreur concernant la détermination taxonomique, il est automatiquement dirigé vers la page suivante lorsqu'il clique sur « étape suivante ».

4.4. Ecran d'informations chronologiques

L'utilisateur renseigne les dates de chaque observation (Figure 6). Avec un menu déroulant, il doit sélectionner le nom de la colonne de son fichier contenant la date de début de l'observation. Etant donné que dans la grande majorité des cas une observation est réalisée en une journée, le formulaire est pré-rempli à « oui » à la question « est-ce que la date de fin est la même que la date de début ? ». Si ce n'est pas le cas, l'utilisateur doit renseigner la date de fin.

Le bouton « étape suivante » est désactivé tant que l'utilisateur n'a pas renseigné la date de début car les dates de début et de fin sont des éléments obligatoires du SINP.

Lorsque le formulaire est envoyé côté serveur, la conformité et cohérence des données (voir Partie 2 – Cahier des charges) est vérifiée.

The screenshot shows a web interface for 'Téléversement SINP'. On the left is a sidebar with five menu items: 1 Upload, 2 Taxonomie, 3 Chronologie (highlighted), 4 Géographie, and 5 Autres. The main content area is titled '3 INFORMATIONS CHRONOLOGIQUES'. It contains two sections: 'Date de début de l'observation' with a dropdown menu labeled 'Date' and the instruction 'Choisir la colonne renseignant la date de début de chaque observation (format date requis)'; and 'Date de fin de l'observation' with the question 'Est-ce que la date de fin de chaque observation est la même que la date de début ?' and radio buttons for 'Oui' (selected) and 'Non'. At the bottom, there are two buttons: 'Recommencer le processus' and 'Etape suivante'.

Figure 6 : Formulaire d'informations chronologiques

4.5. Ecran d'informations géographiques

Ce formulaire permet à l'utilisateur d'indiquer les différents types de données géographiques dont il dispose. Il peut indiquer des coordonnées géographiques et/ou le nom ou le numéro de la commune ou le département. Le bouton « étape suivante » est désactivé tant qu'il n'a pas indiqué un type de donnée géographique.

Si l'utilisateur indique qu'il possède des coordonnées géographiques, il doit ensuite indiquer les noms de colonnes contenant la latitude et la longitude pour chaque observation. Le principe est le même pour le nom/numéro de commune et numéro de département.

Lorsque l'utilisateur clique sur le bouton « étape suivante », les informations du formulaire sont envoyées côté serveur, et leur conformité et cohérence vérifiées :

- pour les observations dont les coordonnées sont manquantes, une erreur est émise si elles ne possèdent pas d'autres informations géographiques.
- pour les observations dont le nom/numéro de commune et numéro de département sont manquantes, une erreur est émise si elles ne possèdent pas de coordonnées géographiques.
- les valeurs longitude et latitude doivent être comprise dans des intervalles de respectivement -180 à 180 et -90 à 90.
- par une jointure avec les données de l'INSEE, une vérification est lancée pour savoir si le nom/numéro de commune et le numéro de département existent. Sinon une erreur est émise.

The screenshot shows the 'Téléversement SINP' interface. On the left is a sidebar with five steps: 1 Upload, 2 Taxonomie, 3 Chronologie, 4 Géographie (highlighted), and 5 Autres. The main content area is titled '4 Informations géographiques'. It contains the following elements:

- A message: 'Vous devez posséder au moins un type d'information géographique pour chaque observation :'
- Two radio buttons: 'Coordonnées géographiques' (selected) and 'Commune et/ou département'.
- A section titled 'Vos données contiennent-elles des coordonnées géographiques ?' with a radio button for 'Oui' selected and 'Non' unselected.
- Input fields for 'Latitude' (containing 'Lat (WGS84)') and 'Longitude' (containing 'Lon (WGS84)').
- A section titled 'Commune et/ou département :' with three radio buttons: 'Commune' (checked), 'Numero de Commune', and 'Département'.
- At the bottom, there are two buttons: 'Recommencer le processus' and 'Etape suivante'.

Figure 7 : Formulaire d'informations géographiques

Lorsque l'utilisateur sélectionne les noms de commune, le numéro de commune et de département sont automatiquement générés. Lorsqu'il sélectionne le numéro de commune, ce sont le nom de commune et les numéros départements qui sont générés.

D'ici la fin du stage, les erreurs seront remontées directement à l'utilisateur pour qu'il puisse corriger les erreurs géographiques directement dans l'application.

Lorsque les informations géographiques de l'utilisateur ne contiennent plus d'erreur, l'utilisateur navigue automatiquement vers le dernier formulaire concernant les informations globales lorsqu'il clique sur « étape suivante ».

4.6. Ecran d'informations globales sur le jeu de données

Pour finir, l'utilisateur rencontre sur cette page une série de 8 questions correspondant pour la plupart d'entre-elles aux métadonnées. Toutes les réponses aux questions sont pré-remplies par défaut. Nous avons essayé au maximum de ne pas utiliser les termes techniques du SINP afin de rendre plus compréhensive et naturelles les réponses attendues.

Les 3 premiers champs sont libres et concernent les informations sur les noms des entreprises et personnes impliquées sur les différentes étapes de la création et gestion des données brutes de biodiversité. Par défaut, les champs sont remplis avec 'INCONNU' car c'est la valeur attendue par le standard SINP lorsqu'aucune info n'est disponible pour ces attributs. Si l'utilisateur efface ce champ et le laisse vide, il ne peut pas cliquer sur le bouton « Télécharger mon fichier SINP ».

Téléversement SINP

1 Upload
2 Taxonomie
3 Chronologie
4 Géographie
5 **Autres**

5 Informations globales

Quel est le nom de l'organisme qui détient les données que vous allez téléversées (et en a la responsabilité) ?

INCONNU *

Quel est le nom de l'organisme qui a réalisé les observations ?

INCONNU *

Connaissez-vous le nom de la ou des personnes qui a/ont réalisé les observations ? (format: Prénom NOM)

INCONNU *

Est-ce que certaines de vos observations sont des "non-observations"?

Oui, au moins une des observations est une 'non-observation'

Non, chaque taxon a été observé directement et/ou des indices témoignant du taxon ont été relevés

Je ne sais pas

Figure 8 : Formulaire d'informations globales sur le jeu de données

Les 3 autres questions du formulaire concernent 3 attributs du standard SINP relatifs au statut, à la nature et à la source de l'observation. Les valeurs attendues par le standard SINP se font parmi une liste finie de valeurs.

Les 2 derniers éléments du formulaire concernent 2 attributs du SINP relatifs à l'identification de l'observation. Pour le premier, l'utilisateur peut indiquer le numéro d'identifiant qu'il a attribué pour chacune de ses observations s'il en possède un. Pour le second, l'utilisateur indique s'il possède un identifiant UUID pour chaque observation. S'il n'en a pas, un UUID sera généré pour chaque observation lorsque le formulaire sera envoyé.

4.7. Ecran de téléchargement du fichier standardisé

Lorsque l'utilisateur arrive sur cette page, le processus de conversion des données est terminé. Un message le félicite pour sa conversion et il peut cliquer sur le bouton « Téléchargez votre fichier au standard SINP » pour obtenir son fichier csv respectant le standard SINP. Ce fichier est prêt à être soumis sur la plate-forme web GINCO pour obtenir le certificat de conformité.

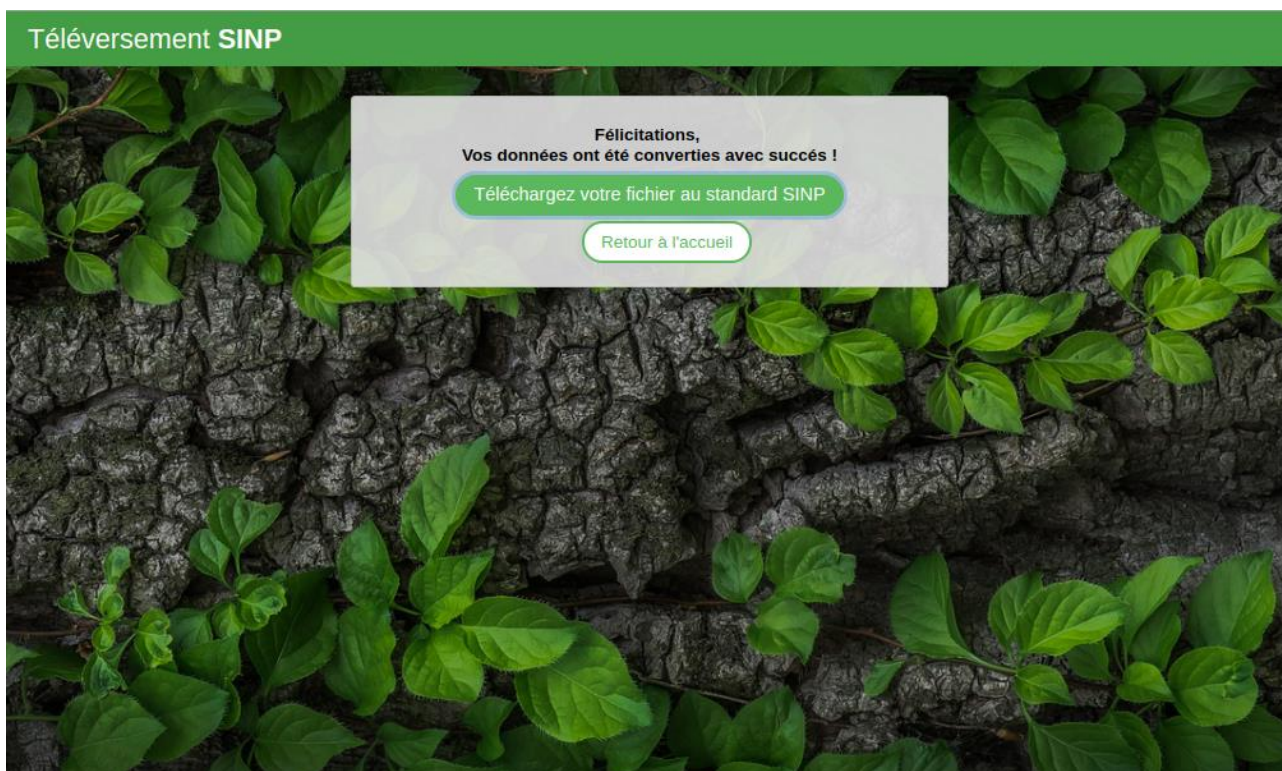


Figure 9: Téléchargement du fichier utilisateur au format standard SINP

4.8. Navigation

Sur toutes les vues, l'application est composée d'un menu sur le côté gauche de l'écran pour que l'utilisateur puisse constamment se repérer dans le processus de conversion de ses données.

Sur toutes les pages, on retrouve un bouton 'retour à l'accueil' (sauf pour la première page) et un bouton 'étape suivante' (sauf pour la dernière page). Lorsque l'utilisateur recommence le processus, ses fichiers sont effacés côté serveur.

5. Problèmes rencontrés et perspectives

5.1. Compréhension du standard SINP

L'aspect théorique du standard « occurrences de taxons » du SINP a été une difficulté au début de mon stage parce que la documentation est volumineuse, précise et parfois contradictoire (versions différentes). De plus, certaines incohérences sont encore présentes entre la documentation du standard SINP et la vérification de la validité du fichier sur la plate-forme web GINCO. Ceci s'explique par la rapidité de mise en place de la loi de téléversement des données de biodiversité, ce qui a provoqué un décalage entre le développement web des plates-formes et les aspects théoriques.

5.2. Formation aux technologies abordées

Etant donné que l'objectif de mon stage était de développer entièrement l'application de téléversement, le champ des compétences à acquérir était assez large, à la fois côtés client et serveur. Je me suis formé à Python que nous n'avions pas vu au cours du Master CCI. Même si un grand nombre de notions étaient similaires à d'autres langages que nous avons abordés, j'ai passé beaucoup de temps à cette tâche.

Les spécificités du développement web, avec Angular côté client et le framework Flask côté serveur, ont également été une difficulté, notamment l'architecture de l'application et la communication client/serveur. Lors du projet industriel au cours du Master CCI, j'avais utilisé Angular mais surtout pour le design et l'intégration, ce qui correspond aux deux points que je n'ai pratiquement pas abordés pour ce stage. J'ai donc dû me former à de nouveaux aspects de ce framework, tels que les requêtes HTTP et les observables.

5.3. Respect des objectifs

Je pense globalement avoir atteint les objectifs de mon stage, dans le sens où l'application est pratiquement opérationnelle de bout en bout, même si elle ne peut pas encore être mise en production. Ma formation aux divers outils ayant été chronophage, je suis encore en cours de développement pour les corrections utilisateurs par l'intermédiaire d'ag-Grid. Je dois améliorer certaines parties de la navigation et j'aimerais développer certaines options supplémentaires pour l'utilisateur en ce qui concerne les informations chronologiques et géographiques. Je n'ai pour l'instant pas rempli entièrement le cahier des charges concernant les coordonnées géographiques puisqu'un seul format de données géographiques (le plus courant) n'est accepté.

5.4. Développements futurs

D'après les retours récents de clients potentiels, il semble que le principal point à aborder dans le futur sera l'intégration d'une base de données. Il semblerait en effet intéressant pour les grandes entreprises de pouvoir bénéficier d'un système de connexion d'utilisateurs correspondant à plusieurs projets/chantiers simultanés pour téléverser les données et les centraliser en interne. De plus, un système de base de données offrirait la possibilité à l'utilisateur d'enregistrer une session au cours du processus de conversion et de reprendre plus tard.

Il serait également intéressant d'améliorer/complexifier les possibilités de correspondances entre les données utilisateur et le référentiel TaxRef. Pour l'instant, notre application permet la détection des erreurs utilisateur concernant le nom scientifique en latin. Il serait intéressant d'intégrer des corrections automatiques d'erreurs utilisateurs. Une première piste serait le *fuzzy matching*, ce qui permettrait de détecter et corriger les éventuelles erreurs de frappes pour les noms latins. Dans l'idéal, on pourrait même imaginer un système permettant de corriger la validité scientifique des observations. Par exemple en définissant des caractéristiques pour les taxons, tels que des zones d'habitats et analyser leur position géographique. Ainsi un taxon dont la zone d'habitat se situe à plus de 1000m d'altitude ne pourrait pas être rencontré en plaine.

6. Conclusion

Le téléversement, consistant à verser sur la plate-forme web GINCO des données brutes de biodiversité respectant les normes du standard « occurrences de taxons » du SINP, est une contrainte légale pour les maîtres d'ouvrage depuis juin 2018. En raison des difficultés techniques des maîtres d'ouvrage à conformer leurs données à ce standard, il est nécessaire de créer les outils facilitant cette tâche. C'était l'objectif de mon stage à Natural Solutions pendant lequel j'ai développé une application permettant la conversion de données utilisateurs au format SINP.

La version de l'application développée pendant mon stage permet d'effectuer le processus de conversion de données de bout en bout, c'est à dire de l'envoi du fichier utilisateur contenant les données brutes au téléversement par l'utilisateur de son fichier conforme au standard SINP. C'est une première version assez rigide, dans le sens où elle ne contient pas encore de nombreuses options pour l'utilisateur. L'utilisateur ne peut pas encore corriger ses erreurs en direct par l'intermédiaire de l'application (j'espère finir cette partie du développement avant la fin de mon stage). Les formats des données chronologiques et géographiques acceptés par l'application sont encore limités. Ce sont des perspectives d'améliorations à court terme, probablement lors de commandes et demandes concrètes de clients.

D'un point de vue professionnel, j'ai énormément appris au cours de ce stage, tant au niveau du développement web que des méthodologies de travail. Mon travail ayant concerné aussi bien le côté client que le côté serveur, j'ai beaucoup appris sur le développement web, mes compétences dans le domaine étant très limitées au début du stage. Je me suis également formé à la méthode de travail SCRUM qui est appliquée fréquemment dans le domaine du développement web. J'ai pu aussi me former à des outils tels que *git* et *github* pour gérer et partager le code. J'ai beaucoup apprécié travailler sur ce projet de téléversement parce que c'est un projet nouveau et important de NS et qu'il correspond à un besoin concret des maîtres d'ouvrage. De plus, NS et plus particulièrement Olivier Rovellotti et mon maître de stage, Khaled Talbi, m'ont fait confiance pour ce projet (développement de l'application, rendez-vous clients, prestations de téléversement, articles publiés sur le blog) ce qui m'a permis d'avoir une assez grande liberté et autonomie, même si mon travail a bien sûr fait l'objet de nombreux réajustements par les personnes impliquées dans le projet au fur et à mesure des avancées.

Références bibliographiques

Gargominy, O., Terceirie, S., Régnier, C., Ramage, T., Dupont, P., Daszkiewicz, P. & Poncet, L. (2017). *TAXREF v11, référentiel taxonomique pour la France : méthodologie, mise en œuvre et diffusion*. Muséum national d'Histoire naturelle, Paris. Rapport Patrinat 2017-116. 152 pp

Jomier R. (2017). *Métadonnées du SINP v1.3.8*, UMS 2006 Patrimoine Naturel, Paris, 51 pp

Jomier R., Chataigner J., Poncet L., Lebeau Y., Robert S., Bourgoïn T., Chagnoux S., Saltré A., Borremans C., Archambeau A-S., Lecoq M-E, Pamerlon S., Just A., Milon T., Cousin J-L., Viel N., Barreau S. (2016). *Standard de données SINP Occurrences de taxons*, version 1.2, MEDDE/MNHN, Paris, 93 pp

Jomier R., Poncet L., Robert S., Milon T., Archambeau A-S., Pamerlon S., Bourgoïn T., De Monicault L., Barneix M., Fromage P., Carpy B., Callou C., Candelier S., Huguet A., Meunier D., Wódka-Gosse A., Grossiord F., Guichard B., Landrieu G., Lafage B. (2018). *Standard de données SINP Occurrences de taxons*, version 2.0, UMS 2006 "Patrimoine naturel", Paris, 100 pp

Robert S., Barneix M., Body G., Castanet J., Caze G., Cellier P., Desse A., de Mazières J., Fromage P., Gourvil J., Jomier R., Juste A., Landry P., Lebeau Y., Lecoq M.E., Lescure J., Marage D., Meyer D., Pamerlon S., Papacotsia A., Poncet L., Quintenne G., Saltré A. & Touroult J. (2016). *Guide méthodologique pour la conformité, la cohérence et la validation scientifique des données et des métadonnées du SINP – Volet 1 : occurrences de taxons*, Version 1. Rapport pour le SINP, rapport MNHN-SPN 2016-77, 63 pp

Liste des légendes

Figure 1 : Ecran d'accueil	23
Figure 2 : Importation du fichier utilisateur	24
Figure 3 : Vérifications préliminaires du fichier utilisateur	25
Figure 4 : Affichage d'une erreur utilisateur lors de l'envoi du fichier	25
Figure 5 : Formulaire d'identification taxonomique	26
Figure 6 : Formulaire d'informations chronologiques	27
Figure 7 : Formulaire d'informations géographiques	28
Figure 8 : Formulaire d'informations globales sur le jeu de données	29
Figure 9: Téléchargement du fichier utilisateur au format standard SINP	30