

RAPPORT D'ANALYSE INFORMATIQUE

**Solution Interactive pour
le Partage et Recherche des
Retours d'Expérience par
des Utilisateurs des
Données Géographiques**



Encadrants : Théo SZANTO, Bénédicte BUCHER, Mathieu BRÉDIF

Professeur référent : Victor COINDET

1. Contexte du Projet

1. Destinataires :

Le projet vise à répondre aux besoins des utilisateurs, en particulier des chercheurs issus de divers organismes, qui manipulent des données géographiques et qui souhaiteraient documenter leur donnée, déclarer un problème rencontré lors d'un traitement géographique ou explorer des données similaires ayant des points en commun.

2. Contexte général :

Lorsqu'un utilisateur est confronté à une question liée à un territoire, il doit identifier les données pertinentes, évaluer leur qualité et leur adéquation, comprendre leur utilisation optimale, et le cas échéant résoudre les anomalies détectées. Il doit également vérifier l'absence de décalage ou d'incohérence entre les différents jeux de données employés, avant d'exploiter les résultats pour répondre à sa problématique. À l'exception de l'exploitation finale des résultats, toutes ces étapes relèvent de ce qu'on désigne comme l'ingénierie de la donnée. Bien souvent, l'utilisateur possède des connaissances pointues dans son domaine d'application, mais manque d'expertise en sciences géographiques.

3. Objectif Principal :

Le but est de développer une application web pour permettre aux utilisateurs de renseigner des retours d'expériences à propos de leurs usages des données géographiques. La plateforme utilisera ces retours pour générer des métadonnées qui devront être exploitables en dehors de l'application (*l'option de l'export*) et explorables sous forme de "pages de profil" des différents jeux de données, navigables via des liens hypertextes.

La saisie de retours d'expériences se fera sous forme de texte libre que l'utilisateur peut annoter (i.e. attacher un label à un fragment du texte). Le système devra être capable d'accompagner l'utilisateur en complétant les labels à partir de ce qu'il connaît (à la manière de Teams qui auto-complète les noms d'utilisateur avec "@Nom..."). Pour ce faire, l'application recense les concepts qu'elle rencontre dans les textes des utilisateurs (à partir des annotations) et crée des liens entre ceux qu'elle rencontre au sein du même texte.

4. Enjeux globaux :

De manière plus étendue, le projet s'inscrit également dans le cadre du développement de la documentation de la donnée. De nombreuses initiatives en France mais surtout par l'Europe ont été faites afin de généraliser le partage de métadonnées géographiques, notamment à travers certaines normes ISO relatifs aux métadonnées mais en particulier par la création de la directive INSPIRE en 2007 qui met l'accent sur la production et le partage gratuit de catalogue de données.

2. Objectifs de l'étude

1. Objectifs au cours du projet :

De manière plus détaillé, le projet a pour objectif de fournir aux utilisateurs un ensemble de fonctionnalités facilitant l'exploitation et l'amélioration des données géographiques, permettant aux chercheurs et autres acteurs travaillant sur ces données de documenter leur travail en rédigeant des annotations détaillées sur les analyses effectuées. Ils pourront ainsi expliciter les démarches, les problématiques rencontrées ainsi que les solutions mises en place pour y remédier.

Par ailleurs, le projet doit offrir la possibilité d'explorer les retours d'expérience liés à une ressource, par exemple un jeu de données, et de naviguer au sein de ces contributions à l'aide de liens hypertextes. Cette approche favorise le partage de connaissances et l'enrichissement collectif des analyses.

Afin d'optimiser l'utilisation des données, la plateforme doit proposer également des jeux de données fréquemment combinés avec ceux consultés, facilitant la mise en relation de ressources pertinentes. De plus, les utilisateurs auront la possibilité d'identifier et d'examiner les problèmes relevés sur différents jeux de données, et de proposer des solutions adaptées pour améliorer leur qualité et leur fiabilité par le biais d'un commentaire qu'il peuvent publier sur un retour d'expérience d'un autre utilisateur.

Enfin, pour garantir une meilleure interopérabilité des informations, le projet devra intégrer un module d'exportation des métadonnées. Ces différentes fonctionnalités contribueront à structurer et à fiabiliser l'ingénierie des données géographiques, en offrant aux utilisateurs un cadre de travail plus efficace et collaboratif.

2. Objectifs Optionnels :

Mise à part les objectifs principaux, d'autres requêtes optionnelles ont été présentées afin de compléter le projet :

- Développement d'un plugin QGIS : Le but étant de proposer les mêmes fonctionnalités qu'en web mais cette fois à travers une extension QGIS, permettant de travailler au plus près des outils métiers.
- Élaboration d'un Chatbot : Le principe étant d'aider encore plus l'utilisateur à rédiger le retour d'expérience par exemple automatisant ainsi l'annotation du retour d'expérience, et éventuellement d'ajouter des précisions lorsqu'il le juge nécessaire pour améliorer davantage la qualité du retour d'expérience.
- Conteneurisation et Cloud : L'objectif est de parvenir à déployer la solution à partir de conteneurs dans un environnement cloud, plus précisément une infrastructure de cluster de dockers pour gérer tout un groupe d'utilisateurs.

3. Problématiques :

Différentes problématiques et obstacles se sont imposés au cours du projet. D'une part nos encadrants nous ont mis en avant divers problématiques relatifs à ce type de sujet :

- La documentation technique des jeux de données n'est pas toujours simple à trouver, ni à explorer / comprendre. C'est un travail de plus, notamment pour les chercheurs, qui peut alourdir leurs tâches quotidiennes.
- Trouver des exemples d'utilisation et des indicateurs de qualité sur les données n'est pas évident.
- Si on rencontre une anomalie ou une incohérence sur un ou plusieurs jeux de données, il y a de fortes chances qu'on ne puisse pas les régler si on ne dispose pas de connaissances techniques ou d'un moyen de contact avec les producteurs de données.
- Deux utilisateurs qui ont un besoin similaire ou rencontrent des problématiques techniques identiques se retrouvent souvent à faire la même chose individuellement, sans mutualisation. Alors qu'ils pouvaient collaborer entre eux et progresser plus.

- La plupart du temps, les producteurs de données n'ont pas ou peu de visibilité sur leurs usagers et ne savent pas comment leurs données sont utilisées.

4. Contraintes au cours du projet :

De plus, différentes contraintes se sont ajoutées au projet qui ont constitué un vrai challenge pour le développement de celui-ci.

La base de données Neo4J, étant un défi à utiliser, nous souhaitions que celle-ci soit une instance connectée en ligne, or, le réseau de l'école bloquait les ports en sortie utilisé par le SGBD (les ports 7474 et 7687 en particulier), nous avons dû alors travailler isoler les ports en passant par des conteneurs Docker. Mais avant, il fallait utiliser nos propres réseaux wifi pour se connecter à l'instance de la base de données Neo4J hébergée en cloud. Et quand nous avons contourné ces problèmes, le hôte de l'ENSG (localhost) bloquait le démarrage de notre application web. D'où une proposition d'utiliser des éditeurs de code en SaaS tel que Gitpod qui lançait l'application depuis une autre adresse IP. Cela d'une part, d'autre part il y avait une limitation puisque le compte gratuit ne permettait que la création d'une instance de base de données Neo4J expirant à l'issue d'un mois. Une approche pour remédier à ceci était de réaliser des backups fréquents et recréer l'instance avant l'expiration puis repasser les nouveaux credentials aux variables d'environnement consommés par la solution web.

Concernant l'aspect technique, nous devions aussi apprendre à utiliser des bases de données graphes qui nous étaient jusqu'ici inconnues (nous connaissions très bien les SGBD relationnelles et selon les membres du groupe les Bases de données NOSQL orientés clé-valeur). Cependant, nous n'étions pas familiarisés avec les bases de données à graphes, ni le code Cypher qui était en quelque sorte différent de SQL et SPARQL. Nous avons donc entrepris plusieurs jours à l'exploration du tutoriel dans l'académie Neo4J afin de s'acclimater avec le concept de bases de données graphes ainsi que le langage de requête associé Cypher.

En outre, il a fallu plonger et découvrir le monde de l'intelligence artificielle générative pour créer des chatbots (assistant intelligent qui facilite la tâche à l'utilisateur pour intégrer les retours d'expérience et générer des métadonnées). Les contraintes étaient de financer un API OpenAI. Plus le modèle est robuste et performant, plus il faudra payer, ou même plus la clé OpenAI reçoit de requêtes, plus elle consomme de tokens surtout pour la partie de l'interrogation de la base de données afin de répondre à une question utilisateur. Parmi les autres enjeux, il y avait le fait de garder l'historique des discussions par utilisateur dans une session donnée, et ce qu'on appelle le LLM grounding (faire comprendre au chatbot, notre

infrastructure de base de donnée, et le contexte de la discussion afin de rédiger les prompts nécessaires pour lancer des actions). Faute de disponibilité d'une base riche sur les retours d'expérience utilisateur sur les données géographiques, il n'y avait pas la possibilité de générer des graphes de connaissance et appliquer le RAG (Reality Augmented Generation) qui garantit de meilleurs résultats. En dépit de ces contraintes, nous avons réussi à concevoir un chatbot qui fonctionne.

Pareil pour le framework React, auxquels nous ne nous sommes pas familiarisés, il était d'une très grande utilité pour offrir une interface ergonomique. Donc finalement, nous avons décidé de l'utiliser au lieu de Vue.js même si ce dernier est plus simple.

5. Acteurs de la solution :

Différents acteurs/types d'utilisateurs ont été imaginés au cours du projet, chaque utilisateur est inscrit dans la Base de données, celui-ci peut avoir 3 rôles distincts lui octroyant différents droits :

- Rôle Visiteur/Anonyme : Permettant d'accéder uniquement à la partie recherche des métadonnées et toutes les fonctionnalités qui s'en suivent, sans pouvoir édifier.
- Rôle Éditeur : Qui comprend en plus la partie édition de métadonnées.
- Rôle Administrateur : Permettant de gérer les données et les utilisateurs. Il a la main sur la suppression et la modification des propriétés d'une manière très libre.

3. Analyse Fonctionnelle - Fonctionnalités proposées

1. Elaboration de la Base de Données :

1.1 Conception générale :

Après avoir pris en main l'outil Neo4J et le concept de graphes dans son ensemble, nous nous sommes intéressés à la schématisation de la base de données. Le concept des BDD graphe se repose sur 4 types d'entité :

- Le noeud : Elle englobe toute notre donnée et est définie par un nom
- La relation : Sert à faire les liens entre les noeuds
- Le label : Décrit à quoi correspond le noeud
- La propriété : Définit plus précisément un noeud

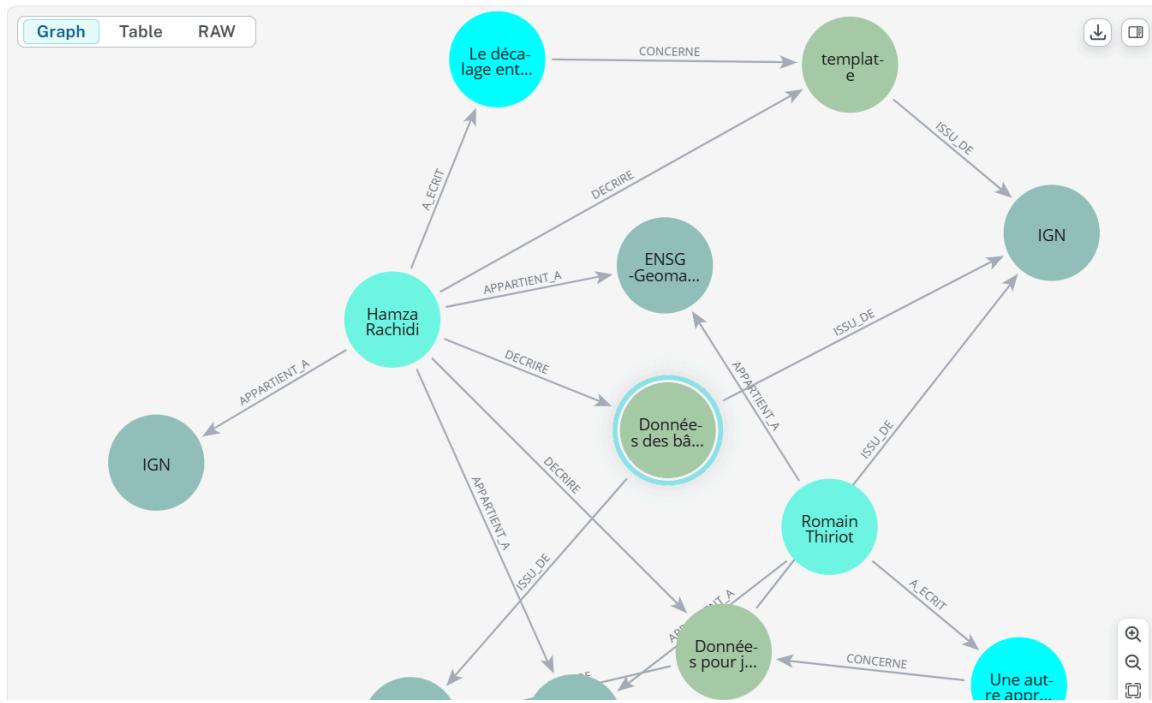
L'objectif in fine est de produire une base de données suffisamment intelligente, efficiente et performante.

Nous avons alors réfléchi aux différents labels attribués aux nœuds qui seraient présents pour le projet. Les 2 labels qui nous ont paru évident sont :

- **L'Utilisateur** qui se connecte, dispose d'un rôle lui permettant différentes fonctions.
- **Le Retour d'expérience**, qui est le cœur du projet et correspond à la donnée.

Au cours de nos réflexions, l'idée qu'un **Utilisateur** puisse commenter et apporter des solutions en répondant à un retour d'expérience semblait primordial, un label **Solution** a donc été imaginé et permet de faire le lien entre un **Utilisateur** et un **Retour d'expérience**. Enfin, il nous a semblé important d'ajouter un label **Source** de la donnée car une même source peut en effet être associée à plusieurs retours d'expérience, et ce label permet à l'utilisateur de faire le lien entre un jeu de données décrit dans un retour d'expérience et l'organisme dont il provient éventuellement.

Ci-dessous un schéma global type de notre conception de la base de données :



1.2 Liste des propriétés :

Pour que le retour d'expérience soit pertinent nous avons élaboré une liste de propriétés relatif à un retour d'expérience publié, cette liste s'inspire des normes relatifs aux métadonnées géospatiales diffusés par DataGouv: (<https://recherche.data.gouv.fr/fr/guide/guide-de-saisie-des-metadonnees-geospatiales?>)

Cette liste se compose donc de :

- **Title** : Correspond au titre mit avant la description
- **Date_creation** : Champ automatique qui retourne la date-heure de publication
- **Description** : Correspond à l'ensemble du texte brut
- **Généalogie** : Historique du traitement et/ou qualité générale de la donnée
- **Droits D'usage** : Niveau de partage de la donnée
- **Lieu** : Point inscrit par l'utilisateur si Zone Localisation non présent

- Unité Géo : Niveau de la localisation (ville,région,pays)
- Zone Localisation : Nom de la zone étudiée
- Mode Acquisition : Technologie d'acquisition de la donnée source
- Système De Coordonnées : Projection utilisé
- Format Fichier : Format utilisé (shp,tiff, etc)
- **Source : Provenance de la donnée**
- Problème : Problème relatif à la donnée étudiée
- Résolution Spatiale : Niveau de détail de la ressource
- (Catégorie Données)
- Degré : Niveau de conformité de la ressource
- Spécification : Référence des règles de mise en œuvre adoptées
- (Emprise)

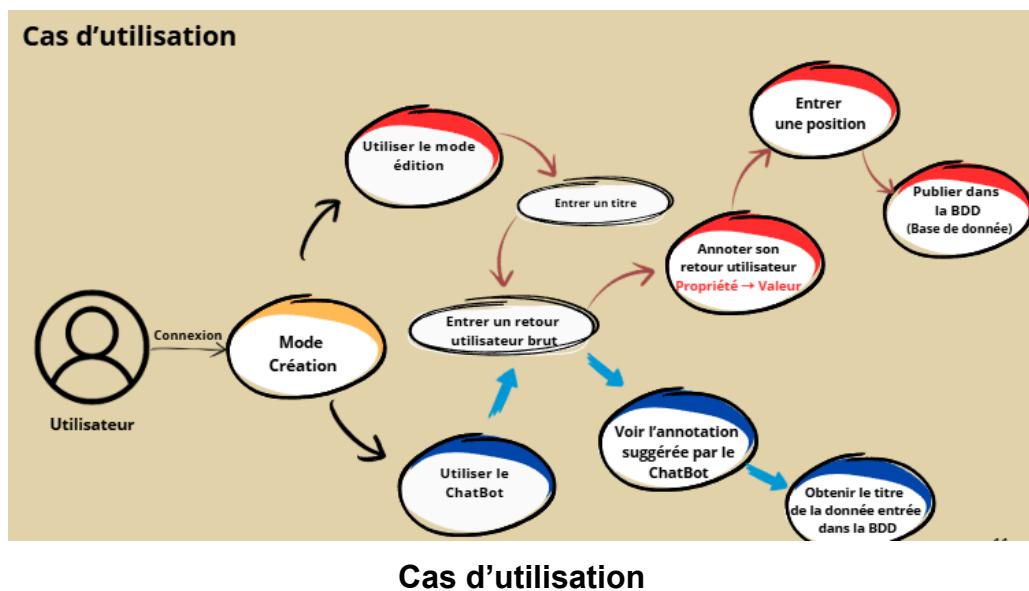
2. Identification en détail des fonctionnalités :

2.1.1 Crédit d'un retour d'expérience

Lorsque l'utilisateur se connecte, il peut accéder en tant qu'éditeur à l'interface de création de retour d'expérience, cette création se fait en plusieurs étapes distinctes :

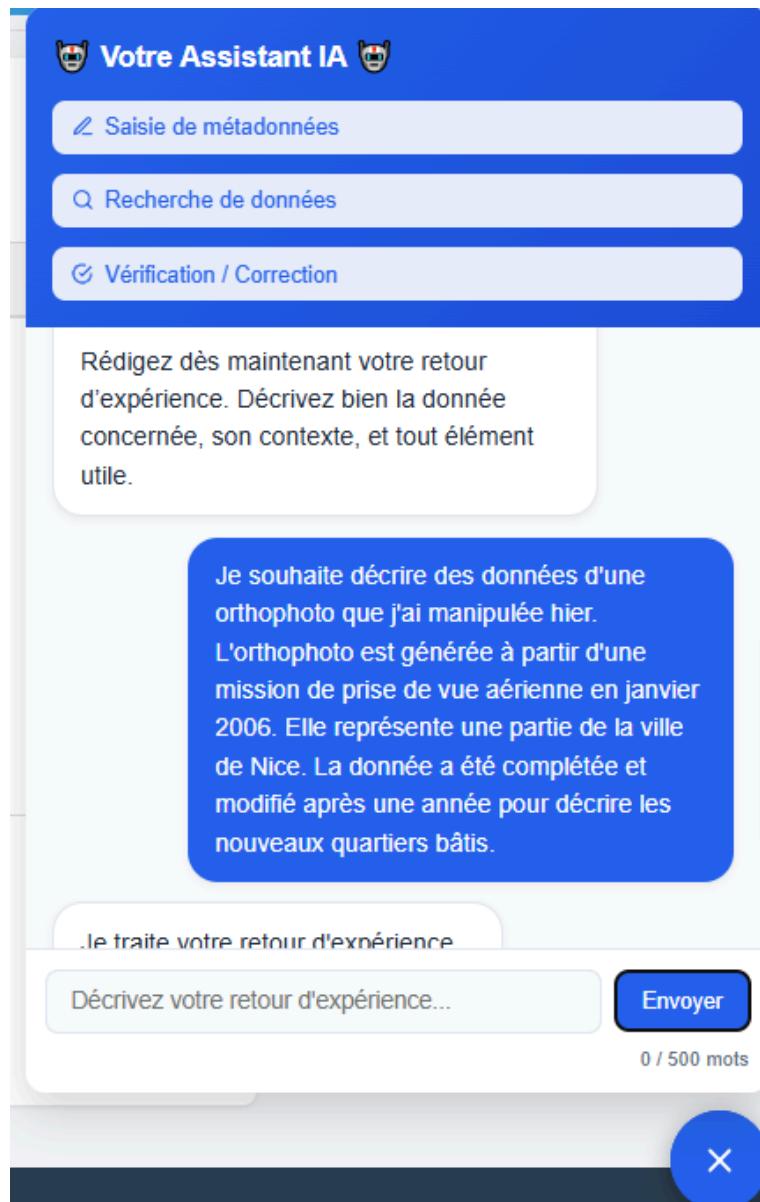
- L'utilisateur rentre un titre qui correspond à la donnée étudiée (ce champ est obligatoire et ne doit pas être déjà présent dans la base de données).
- L'utilisateur commence à écrire son texte puis via la touche # il accède d'une part à la liste des **propriétés**.

- Il choisit ensuite la propriété et accède à la liste des valeurs déjà inscrit dans la base de données, il peut également taper une nouvelle valeur.
- Il recommence ce processus jusqu'à la fin de sa rédaction et clique sur publier.
- Si un lieu a été annoté, le texte est directement publié, autrement, on affiche la carte et on propose à l'utilisateur de choisir un point qui correspond à son retour d'expérience.



2.1.2 Utilisation de l'IA

Tout d'abord, l'utilisateur a une icône tout en bas, ce n'est ouvert que s'il a besoin de l'intelligence artificielle. Pour l'instant, la partie de génération métadonnées est celle qui est complètement opérationnelle. Il clique donc sur le premier bouton : "Saisie de métadonnées". Il reçoit un message explicatif qui lui explique la modalité. Ensuite saisit avec un langage humain naturel sans avoir besoin à annoter son retour d'expérience la description des données qu'il a manipulé.



Une fois qu'il a fini son message, il clique sur le bouton "Envoyer". Un message de traitement s'affiche, une fois le processus de l'annotation automatique, récupération de json et puis stockage dans la base de donnée avec un titre choisi par le chatbot, celui-ci envoie une réponse confirmant le succès de l'opération ou non. Si l'opération est réussie, il écrit le nom qu'il a affecté au retour d'expérience, si par contre l'utilisateur n'a pas saisi une description ou a demandé autre demande, il reçoit une alerte du chatbot affirmant que le json n'est pas valide. Donc les données ne sont pas stockées. Le géocodage est également pris en charge par le chatbot pour placer le marker dans l'endroit approximatif.

Je traite votre retour d'expérience...

Merci ! Votre retour "Données de l'orthophoto" a été enregistré avec succès.

Vous pouvez continuer à ajouter des détails ou passer à un autre mode.

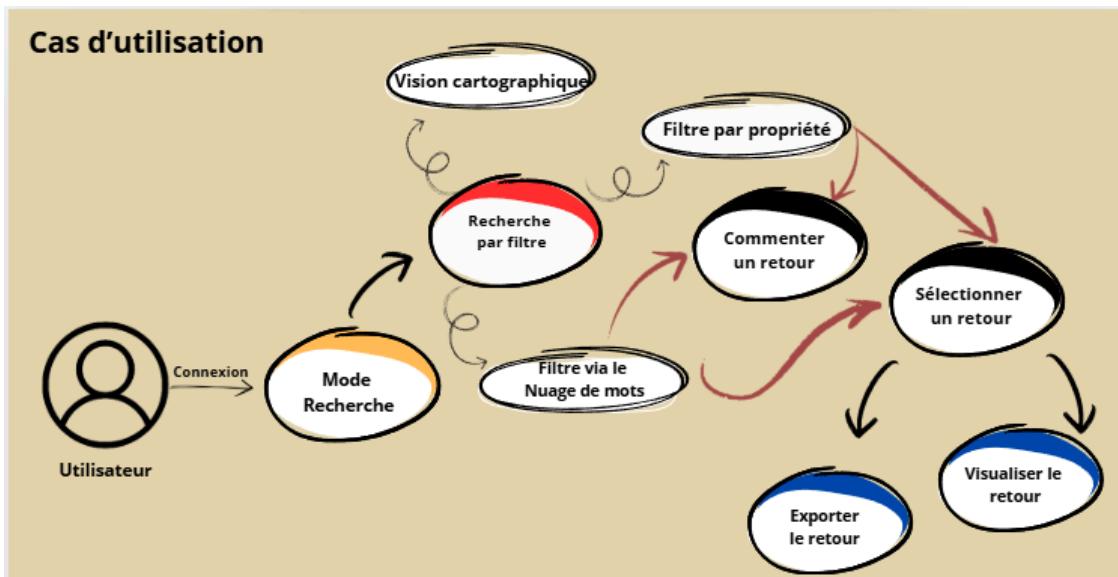
Décrivez votre retour d'expérience...

Envoyer

0 / 500 mots

2.2 Visualisation des retours d'expériences

3 compartiments ont été imaginés pour la partie recherche de retour d'expérience.

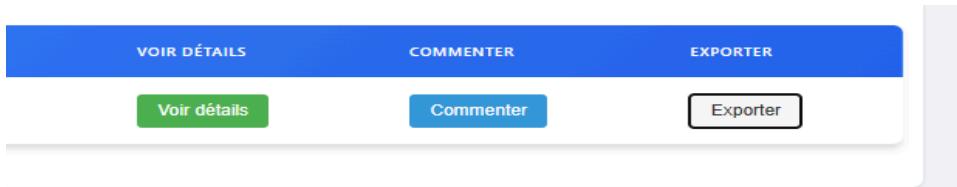


2.2.1 Recherche et Liste

L'utilisateur peut faire une recherche textuelle afin de trouver/filtrer le nom (titre) d'un retour d'expérience. Il peut également utiliser des filtres plus précis selon une propriété (exemple : rechercher les retours d'expérience ayant pour Source BDTOPO).

2.2.1.1 Visualisation du détail d'un retour

Une fois sa recherche appliquée, une liste de résultats s'affiche, en sélectionnant "Voir Détails" l'utilisateur accède aux données relatives au retour d'expérience. Il peut également **exporter** le retour d'expérience au format json.



2.2.1.2 Commenter un retour d'expérience

L'utilisateur en cliquant sur Commenter peut également apporter un message au retour d'expérience, pouvant apporter une solution à un problème, donner des conseils, divulguer des informations. Ces commentaires sont visibles via l'interface de la vision des détails ("Voir Détails") du retour d'expérience.

Détails pour : Données pour jumeau numérique de la ville de Loos-en-Gohelle

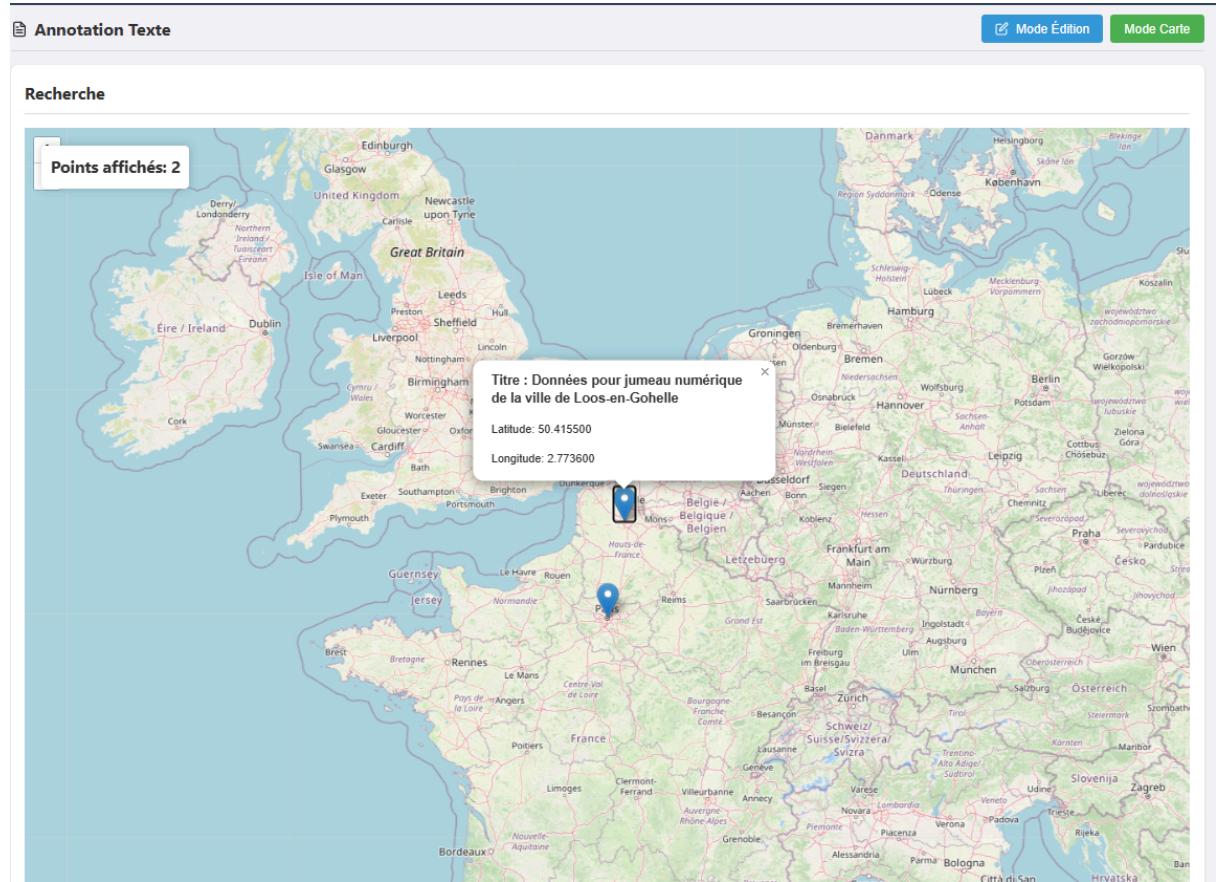
Retour aux résultats

Données complètes :

ATTRIBUT	VALEUR	Ouvrir
Commentaires :	<p>Romain Thiriot - Une autre approche possible serait d'utiliser un algorithme de snapping ou de recalage automatique dans un SIG pour aligner les emprises au plus proche des toits détectés. On peut aussi envisager une normalisation des sources en amont (résolution, projection, seuils de tolérance) pour limiter les écarts avant modélisation</p> <p>romain coin - Pour résoudre les problèmes d'alignement entre emprises et toits des bâtiments, l'utilisation d'outils comme Geoflow-Bundle est en effet pertinente. Il peut aussi être utile de vérifier les écarts de hauteur ou d'emprise en amont via des jointures spatiales dans QGIS ou PostGIS pour ajuster les sources avant fusion.</p>	Ouvrir

2.2.2 Vision cartographique

Comme vu précédemment, les retours d'expérience sont à chaque fois géo-référencés par une **position**. Il est possible d'accéder à une carte et de voir l'ensemble des positions affectées aux retours d'expérience.



2.2.3 Nuage de mots

Avoir un outil de visualisation de statistique était également pertinent dans l'objectif du projet, celui-ci se fait par un “nuage de mots” qui permet de visualiser l'ensemble des termes annotés avec un texte plus ou moins grand selon la présence du terme dans la base de données, un filtre est également possible par propriété. En sélectionnant un terme dans le nuage de mots, la liste des résultats est filtrée.

Annotation Texte

Mode Édition Mode Carte

Recherche

Nuage de Mots

Données de l'orthophoto Données des bâtiments par utilisateur

Données pour jumeau numérique de la ville de Loos-en-Gohelle template

Choisir une propriété : Title

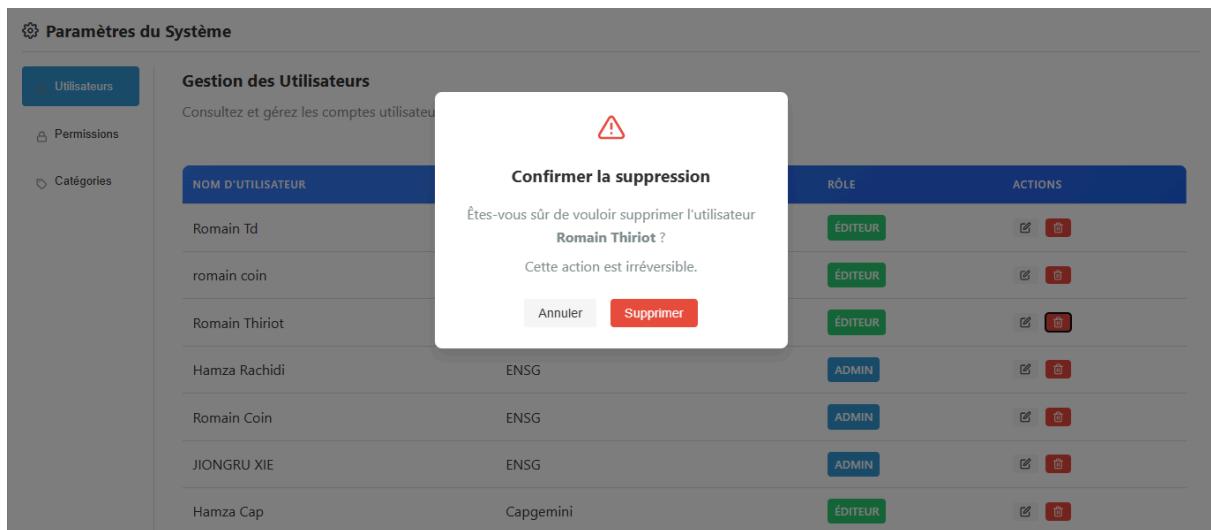
Propriété sélectionnée : Title

Rechercher : Tapez ici (min. 3 caractères).

2.3 Gestion Admin

Regroupe l'ensemble des fonctionnalités destinées aux administrateurs de la plateforme. Elle permet de superviser et d'intervenir sur les éléments clés du système, notamment :

- La gestion des mises à jour des rôles : les administrateurs peuvent modifier les droits d'accès des utilisateurs, en attribuant ou retirant des rôles spécifiques en fonction de leurs besoins ou responsabilités.
- La gestion des nouveaux comptes : les nouveaux utilisateurs peuvent être validés, modifiés ou supprimés par les admins.
- L'accès à l'export global des données : permettant d'exporter l'ensemble des données des retours d'expérience.
- La gestion des types d'annotations : cette fonctionnalité permet de définir, modifier ou supprimer les différentes catégories d'annotations disponibles sur la plateforme, assurant une cohérence dans les types de métadonnées utilisées par les utilisateurs.



3. Généralités sur l'Interface et l'Expérience Utilisateur :

L'interface utilisateur propose un design épuré avec une alternance entre thème clair et thème sombre, permettant une adaptation aux préférences visuelles de l'utilisateur. Un tutoriel intégré facilite la prise en main de l'application dès la première utilisation, assurant une meilleure compréhension des fonctionnalités. Par ailleurs, l'approche minimaliste sans défilement (no scroll), privilégiant une navigation avec peu d'onglets, renforce la lisibilité et améliore l'expérience utilisateur en évitant la surcharge visuelle. En tant qu'éditeur les fonctionnalités se retrouvent dans un

même onglet, on utilise un bouton “Mode édition/recherche” pour changer de module.

Bienvenue, Hamza Rachidi de ENSG-Geomatique

Exporter Complet

Annotation Texte Mode Recherche

Créer un retour d'expérience

Entrez le titre du texte

Tapez votre texte ici (utilisez # pour ajouter une annotation et [] pour encapsuler l'annotation)...

Publier Réinitialiser

Annotation Texte Mode Édition Mode Carte

Recherche

NUAGE DE MOTS

Données de l'orthophoto template Données pour jumeau numérique de la ville de Loos-en-Gohelle

Données des bâtiments par utilisateur

Choisir une propriété : Title

Propriété sélectionnée : Title

Rechercher : Tapez ici (min. 3 caractères)

4. Analyse Technique - Choix des technologies et Architecture

1. Identification des outils proposés et envisagés

Nos commanditaires nous ont dès le début proposé des outils et langages de programmation pour développer la solution :

1.1 Front-End :

Utilisation de React, cette technologie nous permet de travailler en javascript que nous avons tous utilisé bien que nous l'ayons fait via d'autres framework (Vue ou Angular). React permet cependant de mélanger intelligemment le HTML et JS avec le JSX et le framework permet de créer des applications performantes. De plus, nous avons tous vu l'intérêt de travailler et de développer nos connaissances sur un framework aussi professionnel.

1.2 Back-End :

Utilisation d'API Rest avec Express, les API sont également de plus en plus utilisés pour créer les liens entre la Bases de Données et l'application. Les informations de l'utilisateur ne sont pas stockées directement sur le serveur et la structure efficiente qui aide à bien distinguer la partie client-front de la partie serveur-back nous aide également en tant que concepteurs de la solution. Nous avons plus particulièrement divisé le back-end en 2 sous-parties: d'un côté les fonctions de requête et de l'autre les routes api qui appellent ces fonctions.

1.3 Base de Données :

Utilisation de base de données graphe hébergée en cloud via le service SaaS Neo4J auraDB (c'est un , primordial dans le projet, permettant de faire des liens plus rapides entre les données et d'aboutir au système de recherche et de recommandation voulu. Neo4J dispose de son propre langage de requête, exclusif à celui-ci : Cypher. Elle est basée sur des nœuds ayant des propriétés, des relations entre ces nœuds avec leurs propres propriétés. En l'occurrence, ça nous servira pour créer les liens entre utilisateurs qui se connectent dans notre application, les organismes auxquels ils appartiennent et les données qu'ils ont manipulé, ainsi que les solutions aux problèmes éprouvés et qui sont apportés en commentaire.

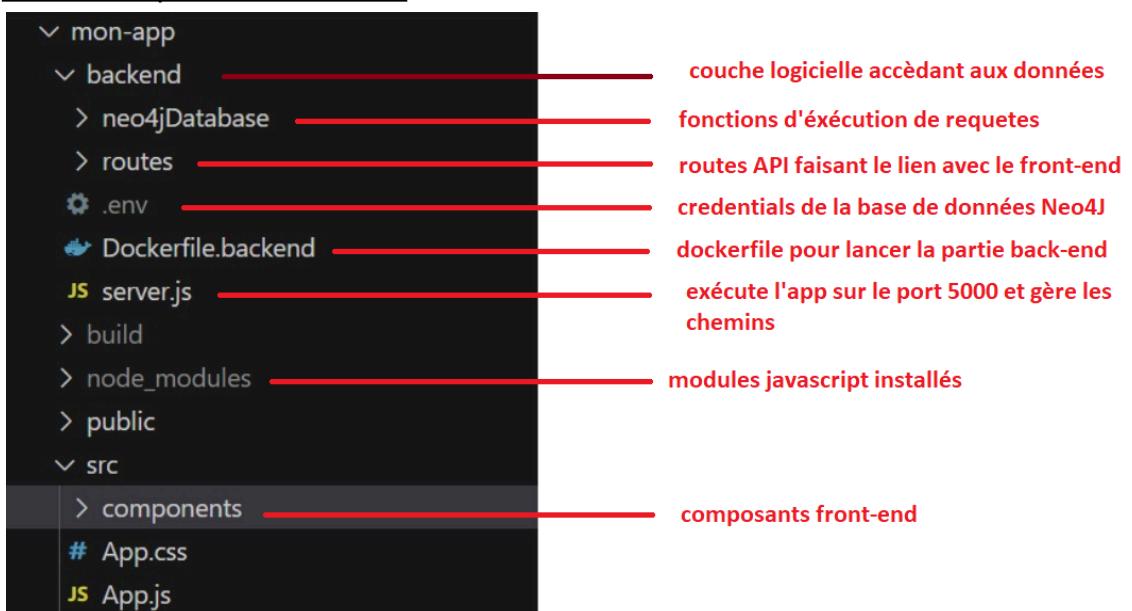
1.4 Autres idées et validation des technologies :

Au-delà de ce qui a été proposé par les encadrants, nous avons également exploré d'autres technologies qui pouvaient également correspondre au projet. L'utilisation de Python a notamment été envisagée pour le développement, pour le back-end via le framework Django. Pour le frontend, nous avons décidé entre Vue.js et React.js. Et pour l'API de l'IA générative, nous avons aussi pensé à exploiter l'API de DeepSeek au début. Cela prouve que le choix de technologies ne s'avère point comme un choix simple.

Finalement, nous avons suivi les conseils de nos encadrants, nos connaissances avec Javascript concernant le web sont plus avancées qu'en Python, de plus, Neo4J paraissait plus facilement intégrable avec Javascript qu'avec Python.

2. Architecture du projet

2.1 Décomposition Générale



2.2 Liste des composants pour l'interface (frontend) de l'application web

La décomposition sous forme de composants React nous permet de structurer le front-end en briques fonctionnelles utiles au projet. Bien qu'un fichier **App.css** recense les éléments utilisés fréquemment, chaque composant dispose de son propre fichier css en cas d'éléments exclusifs à ces derniers.

2.2.1 Partie Connexion

- **LoginForm** : Gère le formulaire de connexion des utilisateurs en vérifiant que les informations tel que le nom, prénom, l'organisation et le rôle dans l'application existent ou non
- **RegisterForm** : Gère l'inscription des utilisateurs via un formulaire dédié en s'assurant qu'aucune information est omise.
- **OnBoardingTutorial** : Propose un tutoriel interactif au premier accès de l'utilisateur lorsqu'il descend sur la page d'accueil de notre application web.

2.2.2 Partie Création de retours d'expérience

- **TextAnnotator** : Permet l'annotation manuelle lors de la saisie du retour d'expérience, elle offre l'auto-complétion et fetch en temps réel: les valeurs existantes des autres données au fur et à mesure de la saisie.
- **AIChatbot** : Gère l'ensemble de la bulle IA en particulier lié à l'aide à la création de retour d'expérience.
- **MapAdd** : Gère l'ajout de points ou zones géographiques sur la carte. Une fois le retour d'expérience validé et publié dans la base de donnée, l'utilisateur place le marker. Une fois les bonnes coordonnées géographiques affichées, il les valide.

2.2.3 Partie Recherche de retours d'expérience

- **BarreRechercheBDD** : gère l'essentiel du module, notamment le formulaire de filtre, le nuage de mots, la liste des résultats du filtre appliqué, l'export des données et la visibilité du jeu de données et des commentaires.
- **Commentaire** : gère l'écriture des commentaires sur un retour d'expérience publié par un utilisateur quelconque.
- **MapRecherche** : Permet de localiser les données via une carte interactive.

2.2.4 Partie Admin :

- **AdminPanel** : Moteur frontend central qui bascule entre mode édition et de recherche, et qui est l'hôte du chatbot.
- **PermissionRequestForm** : Interface pour gérer les demandes d'autorisations de création de comptes, ou de suppression des comptes existants.
- **SettingsPage** : Rôle pas très essentiel dans notre code.
- **UserPermissions** : Gestion des droits utilisateurs.
- **MyButtons** : Ensemble de boutons réutilisables dans l'interface et qui souvent fait appel a la requête HTTP **POST** pour appeler serveur qui emploie les fonctionnalités du backend.

- **ImportExport** : Gestion de l'export de données sous format JSON pour le moment.
- **AboutPage** : Page d'information qui explique l'utilité de notre application.

2.3 Liste des composants pour l'API Rest (backend) de l'application web

La partie du backend a été également structurée en routes, fonctions qui gèrent la base de données Neo4J, prompts pour alimenter le chatbot. L'architecture back a été réalisée en forme d'un API Rest express de Node.js qui reçoit les demandes utilisateur, les traite soit avec LLM, soit sans LLM et puis les renvoie à l'interface.

- **DataRoutes** : Insère le retour d'expérience annoté dans la base de données
- **ListeRoutes** : Renvoie les différentes données relatives à la liste des propriétés, la liste des valeurs, des commentaires, des titres des retours d'expérience et leurs positions. Utilisé dans toutes les briques de l'application
- **OrganizationRoutes** : Gère la partie des organismes (création, affichage, suppression).
- **SolutionRoutes** : Permet de gérer l'ajout de commentaire/solution.
- **UserRoutes** : Gère la partie utilisateur (création, affichage, vérification, suppression, mise à jour)
- **LLMRoutes** : Similaire à **DataRoutes** mais via l'IA.

5. Réalisation du Projet

1. Planning Prévisionnel :

Nous avons 5 semaines pour réaliser le projet. Nous avons imaginé d'une part un planning général comme ceci :

1ère semaine : Analyse du sujet, Création du backlog et des user stories, réflexions sur l'UI, apprentissage et réflexions sur la BDD Neo4J, initialisation du projet React.

2ème semaine : Création de premières fonctionnalités majeures en Front-End concernant la carte, l'annotation et le login, création de la Base de Données et connexion, commencer la configuration des routes API et requêtes cypher.

3ème semaine : Commencer le développement de la partie "Recherche des retours d'expérience", mettre à jour la configuration des routes API et requêtes cypher, commencer à rédiger le rapport.

4ème semaine : Terminer le développement principal du projet et choisir un objectif optionnel à réaliser, continuer à rédiger le rapport.

5ème semaine : Affinements et débugs du projet principal, développement de l'objectif optionnel, finir de rédiger le rapport et préparer la soutenance

2. Outils de collaboration :

Plusieurs outils sont utilisés afin de coopérer au mieux entre les membres du groupe :

- D'un point de vue purement technique, Git/GitHub est utilisé afin de travailler sur le code en collaboration. L'utilisation de l'outil de gestion de projet nous a permis de mieux connaître les tâches et leur statut (à faire, en cours, fini).

- Nous utilisons également un google sheets pour partager toutes les infos utiles entourant le projet, que ce soit les schémas, les débriefs de réunion et l'ébauche du rapport.

- D'un point de vue plus informel, nous utilisons whatsapp afin d'échanger rapidement et faisons régulièrement des points oraux pour décrire ce qu'on a fait, ce qu'on fait et ce qu'on compte faire.

6. Ouverture

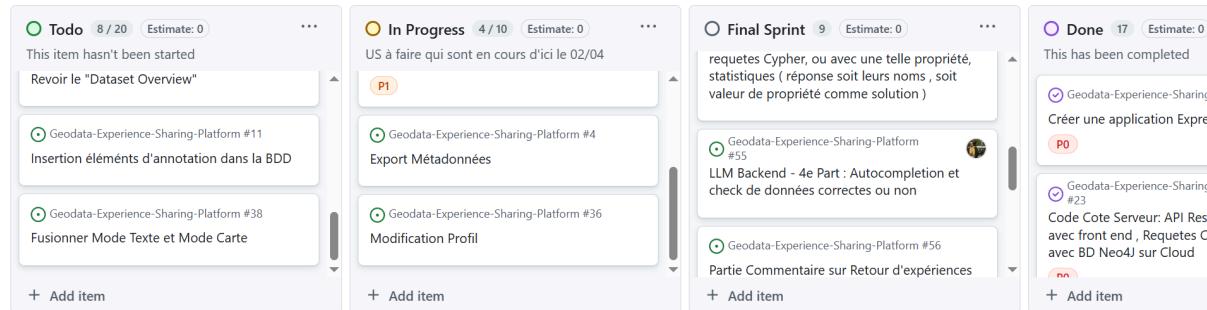
Dans un perspective d'amélioration continue et future qui pourra être envisagée, nous recommandons ce qui suit :

- Ajouter d'autres formats d'export de métadonnées.
- Autocomplétion plus avancée en tenant compte des relations entre propriétés lors de la description de données.
- Recherche multicritère de retours d'expérience par plusieurs filtres.
- Utiliser Framework RAG (Langchain) pour une meilleure performance LLM.

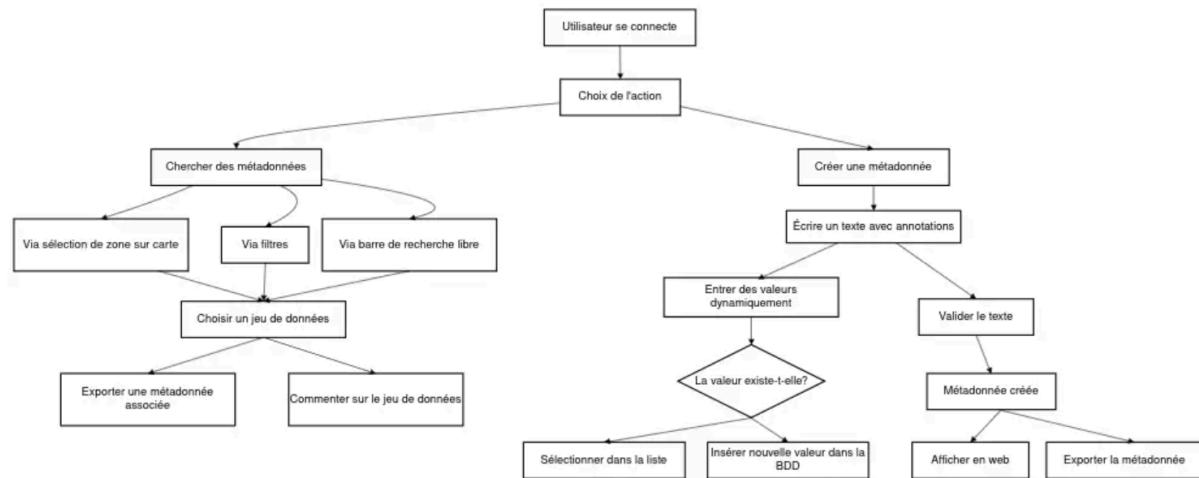
7. Conclusion

En dépit des contraintes que nous avons rencontrées et de la durée proposée pour entamer le présent projet académique, nous sommes arrivés à développer une application web qui fonctionne bien avec des fonctionnalités assez riches. Le besoin principal éprouvé par les commanditaires est satisfait, les objectifs optionnels ont été également abordés. En somme, notre application offre une interface pour l'utilisateur de décrire plus efficacement ses retours d'expérience, les stocker dans un emplacement sécurisé (géré par le SaaS de Neo4J), mais aussi de parcourir les anciens retours d'expérience rédigés par d'autres utilisateurs (et qui peuvent apporter une solution à un problème éprouvé). L'intégration de l'intelligence artificielle générative sous forme d'un chatbot représente aussi un aspect important puisqu'il vaut mieux être d'actualité avec les frameworks et technologies récentes.

8. Annexes



Annexe 1 : Backlog du projet



Annexe 2 : Diagramme de séquence résumant les 2 cas d'utilisation principaux

Pour créer les noeuds

Donnée

- cette partie la elle intervient à quel moment?
- Organisme
- Auteur du retour d'expérience
- Xie propose d'avoir possibilité d'ajouter une propriété?
- pour en avoir partie c'est une bonne idée
- Catégorie
- Bouton Valider les annotations
- Hamza's 2nd suggestion** (ajouter bouton pour déclencher le stockage de la requête cypher dans la base de donnée) c là où se joue le liaison avec backend

j'hésite encore sur les # labels possibles pour les noeuds à part les données ??????

Lié entre eux par code

Donnée

Lié entre eux par code

Catégorie

Système de coordonnées

Emprise géographique

Résolution spatiale

Type de donnée

Droits d'usage

Date de création

Il faut que la 2e liste déroulante encadre l'utilisateur (pour saisir la valeur de la propriété du noeud en question) grâce à la liste déroulante

Utilisateur :

Présentation utilisateur [7], son nom[obligatoire 1/2], organisme[obligatoire 1/2], fonction[npas obligatoire]

et puis **Description** (retour d'expérience), temps

nom du noeud

Titre :

Description :

Ajouter

Annuler

barre de recherche pour trouver des métadonnées, connecté au graphe

barre de recherche ici

Filtres

nom du jeu de donnée

test1

Action

Bouton Détails

détails = voir la MD en détail + commentaires

Bouton Commenter

Hamza's 1st suggestion ! ajouter bouton pour commenter un jeu de donnée comme ça si le commentaire peut aider l'auteur de la retour d'exo

Bouton Exporter

Annexe 3 : Brouillon de design UI via l'outil Miro