

Gestionnaire de couches

SPÉCIFICATIONS FONCTIONNELLES



Entreprise Maz&Mar - Toulouse 31000
Maureen MAZEN, Elia MARIE

Charte de projet écrite dans le cadre du TP sur la gestion de projet, la conception et la maintenance
d'applications et d'interfaces SIG

1. Présentation et Objectifs

a. Présentation

Le projet concerne la création d'un bouton de gestion des couches sur une interface web cartographique. Le gestionnaire de couches permet à l'utilisateur de visualiser les couches qu'il souhaite. L'intérêt de cette fonctionnalité est de permettre à l'utilisateur de paramétriser la carte de façon adaptée à ses besoins spécifiques.

b. Objectifs

Ce document explique en détail les attentes du futur utilisateur et du produit qu'il espère voir être construit. Les descriptions générales de l'outil, des descriptions détaillées des fonctionnalités ainsi qu'un prototype IHM sont donc exposés dans ce document.

2. Descriptions générales de l'outil

Dans un premier temps, l'utilisateur se connecte sur une interface web qui lui permet de visualiser différentes couches sur une web cartographie dynamique et interactive.

Il peut choisir la couche qu'il souhaite afficher, notamment en sélectionnant la couche active dans le gestionnaire. Aussi, plusieurs couches peuvent être sélectionnées et l'utilisateur a la possibilité de choisir un thème permettant l'affichage des couches par thématique (par exemple : Couche 1 et 2, Couche 1 et 4, etc).

Le gestionnaire de couche doit être déroulé avant que l'utilisateur puisse activer ou désactiver les couches. À chaque clic, l'utilisateur coche ou décoche les couches ce qui les active ou désactive.

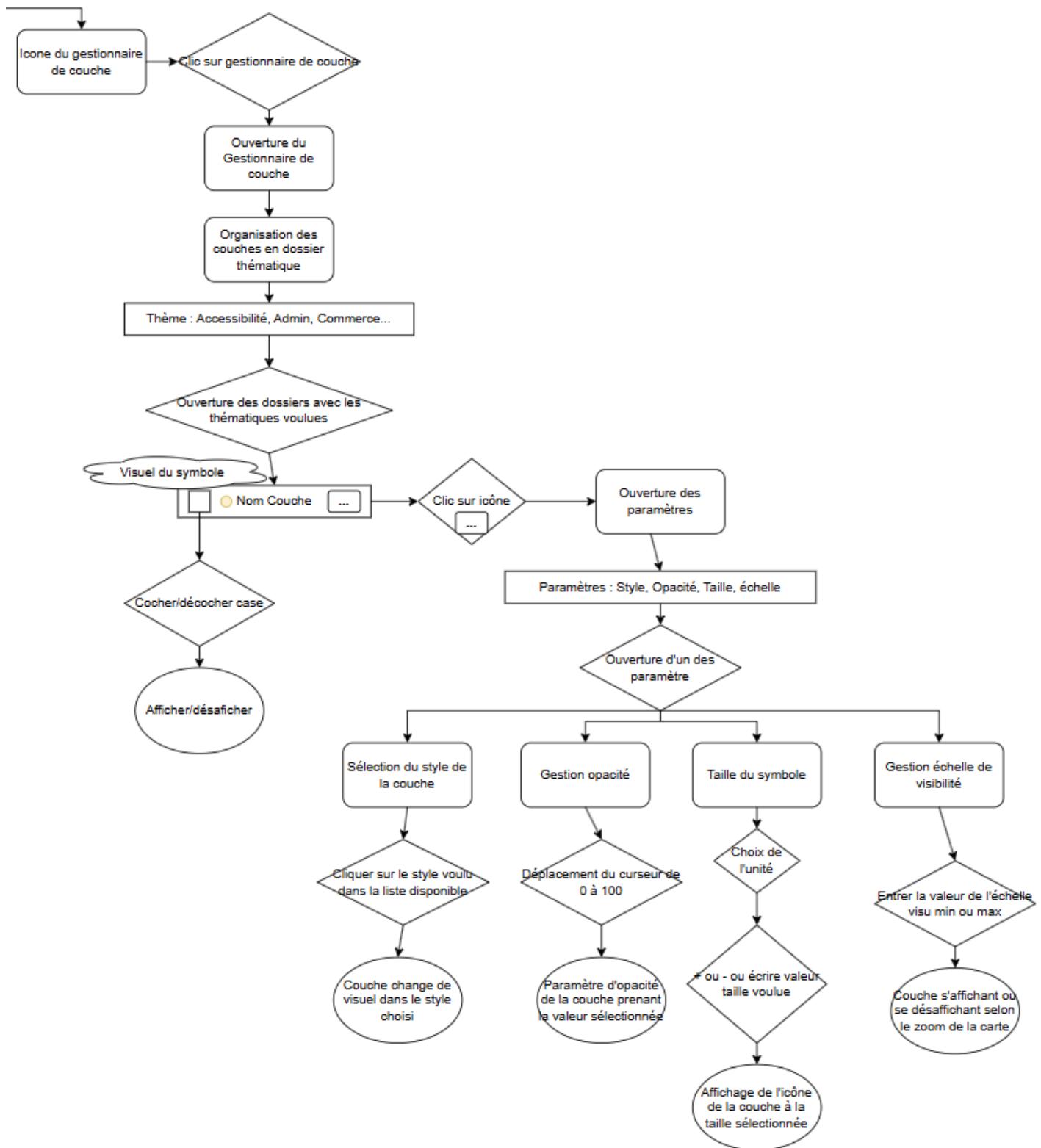
De la même façon, il peut choisir entre trois thèmes qui décoche ou coche automatiquement les couches du thème choisi.

Enfin, il peut définir l'opacité des couches à partir de l'onglet de la couche. Une échelle d'opacité interactive lui permet d'affiner l'opacité de la couche.

3. Descriptions détaillées des fonctionnalités

Un diagramme UML est un moyen de visualiser des systèmes et des logiciels à l'aide du langage de modélisation uniifié (UML). Les ingénieurs logiciels créent des diagrammes UML pour comprendre les conceptions, l'architecture du code et la mise en œuvre proposée de systèmes logiciels complexes. Les diagrammes UML sont également utilisés pour modéliser les flux de travail et les processus d'entreprise.

Vous trouverez ci-après le diagramme UML associé à la fonctionnalité **Gestionnaire de couches**.



4. Prototype IHM

Maz&Mar Site web Évaluation Géocatalogue Contact

Simuler un commerce Accéder au géocatalogue

Échelle

Maz&Mar Site web Évaluation Géocatalogue Contact

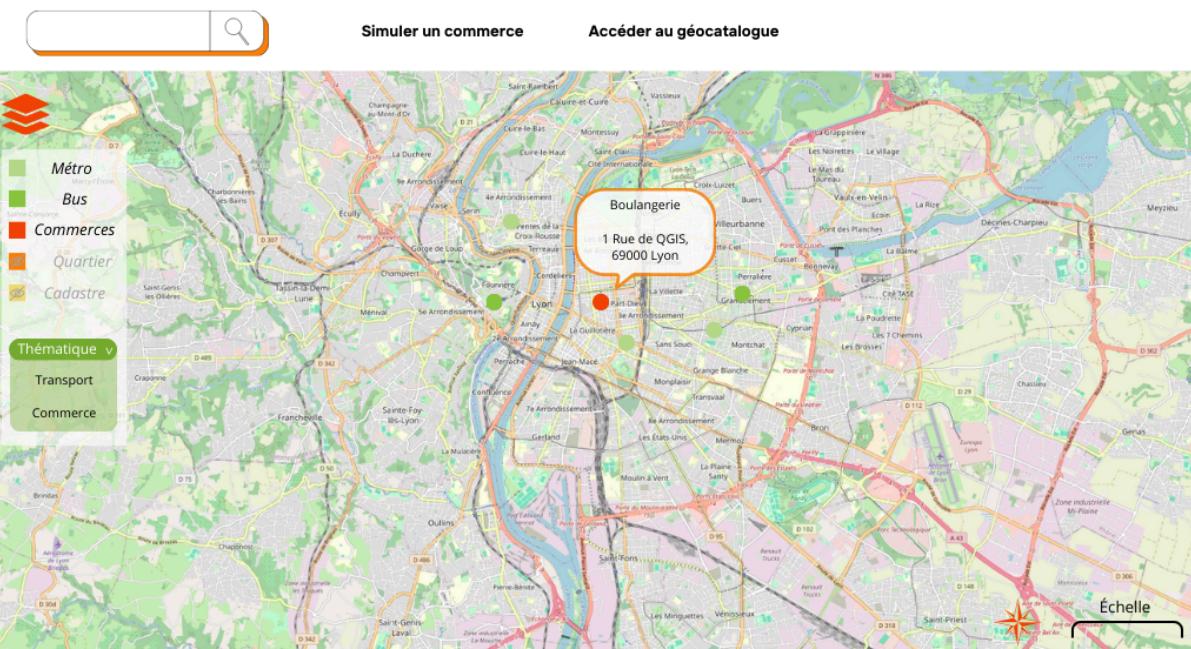
Simuler un commerce Accéder au géocatalogue

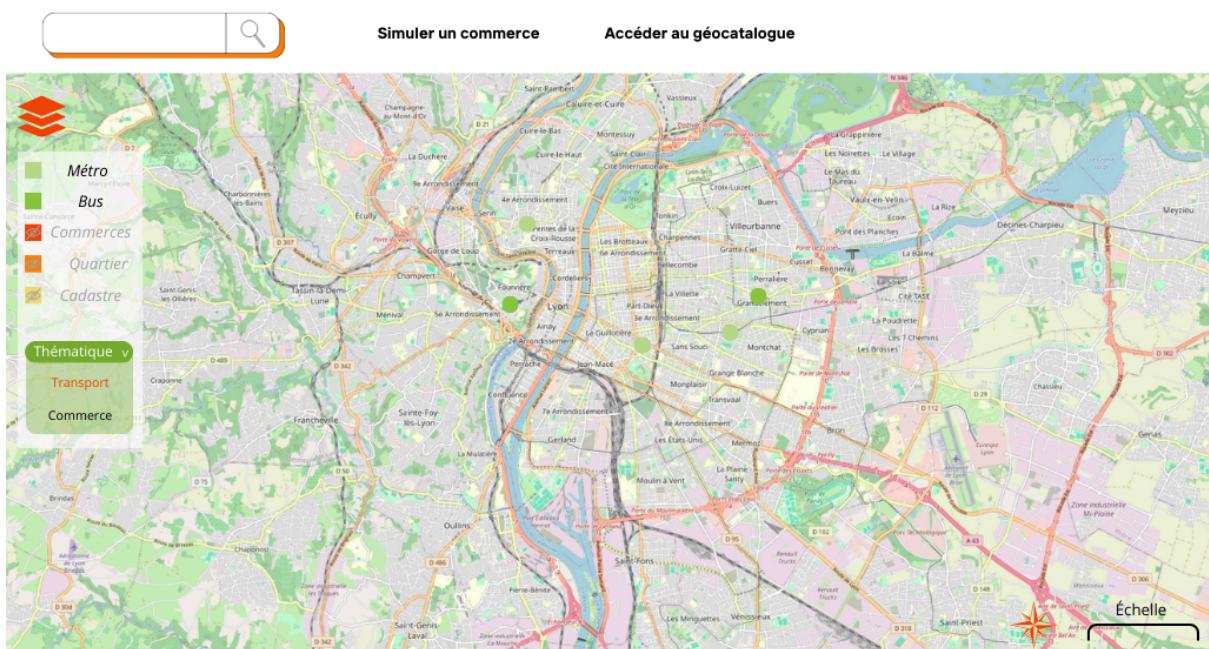
Boulangerie
1 Rue de QGIS,
69000 Lyon

Métro Bus Commerces Quartier Cadastre

Thématische ▾

Échelle





5. Cas particulier

Un utilisateur nous a fait remonter un dysfonctionnement de l'outil sur notre plateforme CRM. En effet, l'une des couches du gestionnaire de couches ne s'affiche pas.

L'utilisateur cherche à afficher une couche à partir d'un clic dans le gestionnaire de couches mais aucune donnée ne s'affiche.

Il se peut que ce soit lié au code source, c'est pourquoi notre expert SIG peut revoir au besoin le script utilisé pour la programmation de l'outil.

Il se peut également que la source de la donnée ait été modifiée, nous donc réalisons des veilles hebdomadaires pour assurer la qualité et le fonctionnement de notre outil.

SPÉCIFICATIONS TECHNIQUES



Entreprise Maz&Mar - Toulouse 31000
Maureen MAZEN, Elia MARIE

Charte de projet écrite dans le cadre du TP sur la gestion de projet, la conception et la maintenance
d'applications et d'interfaces SIG

1. Description des données

Afin de réaliser l'outil, il est nécessaire de connaître la réglementation. Le principal objectif est donc de diffuser les données du client, conformément à l'utilisation des données suivantes :

- Open Data Métropole de Lyon (urbanisme, mobilité, commerces) : nous utilisons des données issues de la plateforme [Open Data de la métropole de Lyon](#).
- INSEE (données démographiques et économiques) : données démographiques issues de la [plateforme de l'INSEE](#).
- OpenStreetMap (réseau routier, points d'intérêt, bâtiments).
- [IGN](#) (infrastructures, cadastre, réseaux de transport).
- API transport pour des données en temps réel : nous utilisons l'API des transports en commun lyonnais [TCL](#).
- Autres sources pertinentes identifiées au cours du projet.

Seules les données Commerce, Stationnement et Arrêt de transport seront stockées dans la BDD.

Description de la couche [Arrêt de transport, Data Grand Lyon](#) :

Informations	Description	Remarque									
Description	Le point d'arrêt est caractérisé par des informations de gestion et de localisation : numéro du point d'arrêt, nom commercial, liste des lignes de desserte avec leur sens, accessibilité PMR, présence d'ascenseur ou d'escalier mécanique, adresse, côté de la chaussée vis-à-vis de l'adresse (en face de l'adresse ou à l'adresse), coordonnées x/y.										
Conditions d'utilisation	Licence Ouverte Open License										
Limite d'utilisation	/										
Informations géographiques	Type de représentation spatiale : Ponctuel Étendue géographique : 5.1296 est, 4.6406 ouest, 45.5678 sud, 45.9165 nord Système de coordonnées : EPSG:4171	cf. IHM pour la symbologie									
Attributs	<table border="1"><thead><tr><th>Nom</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>id (PK)</td><td>int4</td><td>Identifiant TITAN de l'arrêt</td></tr><tr><td>nom</td><td>varchar</td><td></td></tr></tbody></table>	Nom	Type	Description	id (PK)	int4	Identifiant TITAN de l'arrêt	nom	varchar		
Nom	Type	Description									
id (PK)	int4	Identifiant TITAN de l'arrêt									
nom	varchar										

	<table border="1"> <tr><td>desserte</td><td>varchar</td><td></td></tr> <tr><td>pmr</td><td>bool</td><td></td></tr> <tr><td>ascenseur</td><td>bool</td><td></td></tr> <tr><td>escalator</td><td>bool</td><td></td></tr> <tr><td>gid</td><td>int4</td><td></td></tr> <tr><td>last_update</td><td>timestamp</td><td></td></tr> <tr><td>last_update_fme</td><td>timestamp</td><td></td></tr> <tr><td>the_geom</td><td>geometry SRID=4171 GeomType=POINT</td><td></td></tr> <tr><td>adresse</td><td>text</td><td></td></tr> <tr><td>localise_face_a_address</td><td>bool</td><td></td></tr> <tr><td>commune</td><td>text</td><td></td></tr> <tr><td>insee</td><td>text</td><td></td></tr> </table>			desserte	varchar		pmr	bool		ascenseur	bool		escalator	bool		gid	int4		last_update	timestamp		last_update_fme	timestamp		the_geom	geometry SRID=4171 GeomType=POINT		adresse	text		localise_face_a_address	bool		commune	text		insee	text	
desserte	varchar																																						
pmr	bool																																						
ascenseur	bool																																						
escalator	bool																																						
gid	int4																																						
last_update	timestamp																																						
last_update_fme	timestamp																																						
the_geom	geometry SRID=4171 GeomType=POINT																																						
adresse	text																																						
localise_face_a_address	bool																																						
commune	text																																						
insee	text																																						
Lien vers la fiche INSPIRE	https://data.grandlyon.com/geonetwork/srv/fre/catalogsearch#/metadata/015b60fa-836f-4880-9ef2-10859e6e9c89																																						
Préconisation d'usage	<p>Contraintes légales : Pas de restriction d'accès public. Licence Ouverte.</p> <p>https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf.</p>																																						

Les données sont également sujettes à l'utilisation d'une architecture web (avec base de données spatiales et services de visualisation cartographique) telle que :

- **Base de données** : PostgreSQL/PostGIS.
- **Algorithmes de calcul et de routage** : pgRouting.
- **Services SIG** : GeoServer pour la diffusion des données spatiales.
- **Application cliente** : Mviewer pour la visualisation cartographique.
- **Back-end** : Python (Flask) pour le traitement des données et l'API.
- **Environnements** : Un environnement de développement et un environnement de production.
- **Outils de gestion de projet et CI/CD** : GitLab.

Chacune des solutions utilisées sont documentées dans les annexes du dossier de Spécifications Techniques Détaillées (STD). Vous trouverez le détail des technologies utilisées dans la partie suivante.

2. Liste des technologies

La liste des technologies utilisées pour le développement ainsi que pour l'utilisation de l'outil est nécessaire à la bonne compréhension du document. Vous trouverez la liste des technologies utilisées ainsi que le détail de leur fonction.

- Mviewver :

Mviewer est une application cartographique initiée et développée par la Région Bretagne. Plusieurs modes de visualisation. Interopérable au travers de standards et API. Conçu sur des technologies libres.

- Pgsql, Postgis, Pgrouting :

PgSQL : PostgreSQL, aussi connu sous le nom de Postgres, est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD mettant l'accent sur l'extensibilité et la conformité avec SQL.

PostGIS : PostGIS est une extension (plugin) du SGBD PostgreSQL, qui active la manipulation d'informations géographiques (spatiales) sous forme de géométries (points, lignes, polygones), conformément aux standards établis par l'Open Geospatial Consortium (OGC). Il permet à PostgreSQL d'être un SGBD spatial (SGBDs) pour pouvoir être utilisé par les systèmes d'informations géographiques.

PgRouting : pgRouting est une extension qui ajoute le routage et d'autres fonctionnalités d'analyse de réseau aux bases de données PostGIS/PostgreSQL.

- HTML, CSS, PHP, JS :

HTML : langage de balisage conçu pour écrire les pages web. Il s'agit d'un format ouvert très utilisé en informatique.

CSS : langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C).

PHP : langage de programmation libre⁶, principalement utilisé pour produire des pages Web dynamiques via un serveur web

JS : JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web.

- Geoserver :

GeoServer est un serveur informatique⁴ open source et libre³ écrit en Java² qui permet aux utilisateurs de partager et modifier des données géographiques. Conçu pour l'interopérabilité, il publie les données de toutes les sources principales de données spatiales utilisant des normes ouvertes.

- Nginx :

Logiciel libre de serveur Web (ou HTTP) ainsi qu'un proxy inverse.

- Geonetwork :

GeoNetwork OpenSource est une application de catalogue pour la gestion des ressources référencées spatialement. Il offre des fonctions puissantes d'édition et de recherche de métadonnées, un visualiseur de cartes web interactif intégré et est basé sur des standards ouverts.

- Gateway :

Dispositif permettant de relier deux réseaux distincts présentant une topologie différente.

- Jira :

Jira est un système de suivi des bugs, de gestion des incidents et de gestion de projets. Il propose des solutions à la fois à destination des développeurs et des intervenants non développeurs.

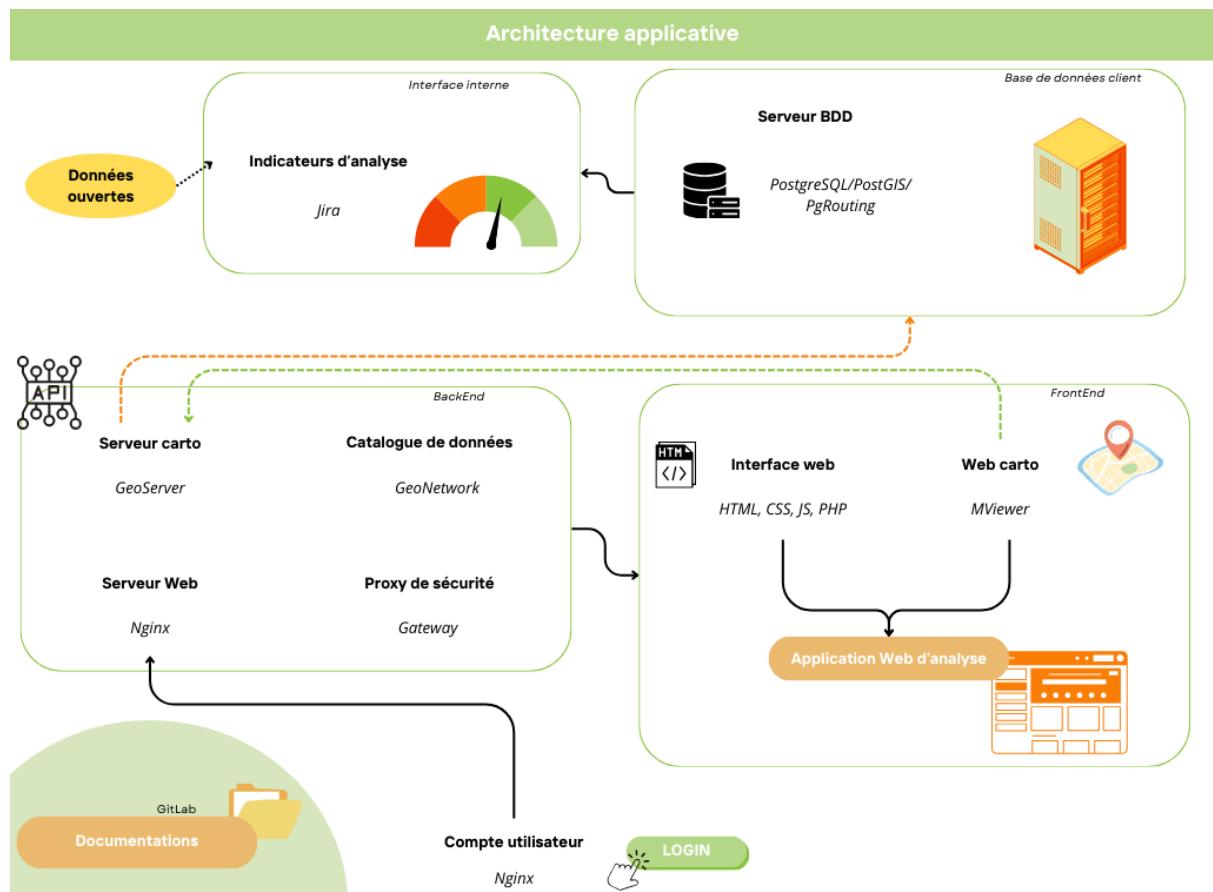
- Gitlab :

GitLab est un logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et la livraison continue.

- Python Flask :

Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger². Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités.

3. Architecture technique



L'architecture technique est une vue tournée vers les différents éléments matériels et l'infrastructure dans laquelle le système informatique s'inscrit, les liaisons physiques et logiques entre ces éléments et les informations qui y circulent.

Ci-dessus se trouve l'architecture technique du projet.

4. Performance

Les performances définissent les exigences fonctionnelles de l'élément, l'environnement dans lequel il doit fonctionner et les caractéristiques d'interface et d'interchangeabilité.

Pour ce projet, l'outil d'analyse spatiale doit être conforme à l'environnement existant du client. Si le client possède un serveur web carto, l'outil doit s'implémenter directement dessus.

L'outil d'analyse prend en entrée la position de l'utilisateur en récupérant le clic ou l'adresse entrée par celui-ci. Grâce au point créé, une liste de commerces, d'arrêts de transport ainsi que de stationnements est générée. Celle-ci apparaît directement sur l'interface web carto utilisateur.

5. Sécurité

Décrire Techniquement le fonctionnement de l'analyse spatiale (donnée et rayon) à partir d'un point
-> qu'est ce qui se passe derrière ?

Bouton d'analyse spatiale : lance une fonction JS permettant à l'utilisateur de cliquer sur la carte ou d'entrer une adresse.

Au clic sur la carte un point est enregistré temporairement dans une vue matérialisée SQL aux coordonnées indiquées et cette couche d'un point s'affiche. Si l'utilisateur clique plusieurs fois, le point change ses coordonnées mais le point est créé seulement au premier clic.

Pour les paramètres, l'utilisateur peut cocher ou décocher les couches de transport ou commerce selon les informations qu'il désire. Il peut également régler le rayon en mètres avec une barre coulissante, la valeur de base est de 300m, et les valeurs bougent de 50m en 50m.

Lorsque l'utilisateur clic sur le bouton valider, si une adresse est entrée dans le texte le point est géocodé à partir de l'adresse et enregistré temporairement, si l'utilisateur à cliqué sur la carte le point est déjà créé. Une vérification est effectuée : il faut qu'un point soit dessiné ou qu'une adresse soit entrée et qu'au moins une couche à analyser soit sélectionnée. Si ce n'est pas le cas, un message d'erreur est renvoyé à l'utilisateur. Une fois toutes les informations paramétrées correctement, le traitement peut donc commencer.

en SQL une vue avec le buffer du nombre de mètres demandé autour du point sélectionné sera créé. Ensuite, pour chaque table sélectionnée, une sélection est faite sur les entités étants en intersection avec le buffer. Ces sélections sont enregistrées dans une vue matérialisée.

La ou les vue matérialisées créées sont ajoutées à la carte.