Ecole nationale de la statistique
et de l'analyse de l'information

Projet informatique

Etudiants 2A

API cadastrale

Etudiants :
Marie Colin
Adrien Gôme
Lucas Laloue
Tanguy Legrand
Julien Sawadogo

Tuteur :
Mathe Thierry
Coach :
Rémi Pepin

2021 - 2022

30 septembre 2022

Table des matières

1 Introduction

-résumer sujet -but

2 Analyse fonctionnelle

2.1 Description générale du fonctionnement du programme

Dans cette partie, nous allons décrire plus précisément le fonctionnement générale de notre programme. Cela se fera nottament avec l'utilisation de diagramme tel qu'un diagramme d'utilisation et d'activité.

Le diagramme d'utilisation permet de visualiser mieux ce que notre programme permettra de faire pour un utilisateur. L'administrateur met à jours les données. Dans cette étape une base de donnée est créée et est stockée quelques soit le choix de l'utilisateur. Cette base de donnée sera présentée dans la partie 2.2 de ce rapport.

Ensuite, l'utilisateur choisis une des requêtes et se verra affichée le résultat souhaité dans la console et aura également un fichier dont le format reste à déterminer.Avec l'utilisation de ce programme l'utilisateur pourra répondre aux questions suivantes :

- Quelles sont les communes ayant des parcelles contiguës/limitrophe a une commune donnée ?

- Quelles sont les parcelles en limite d'une commune donnée ?

- Quelles sont les parcelles contiguës a une parcelle donnée ?

- Il y a t-il un permis de construire sur deux communes ?

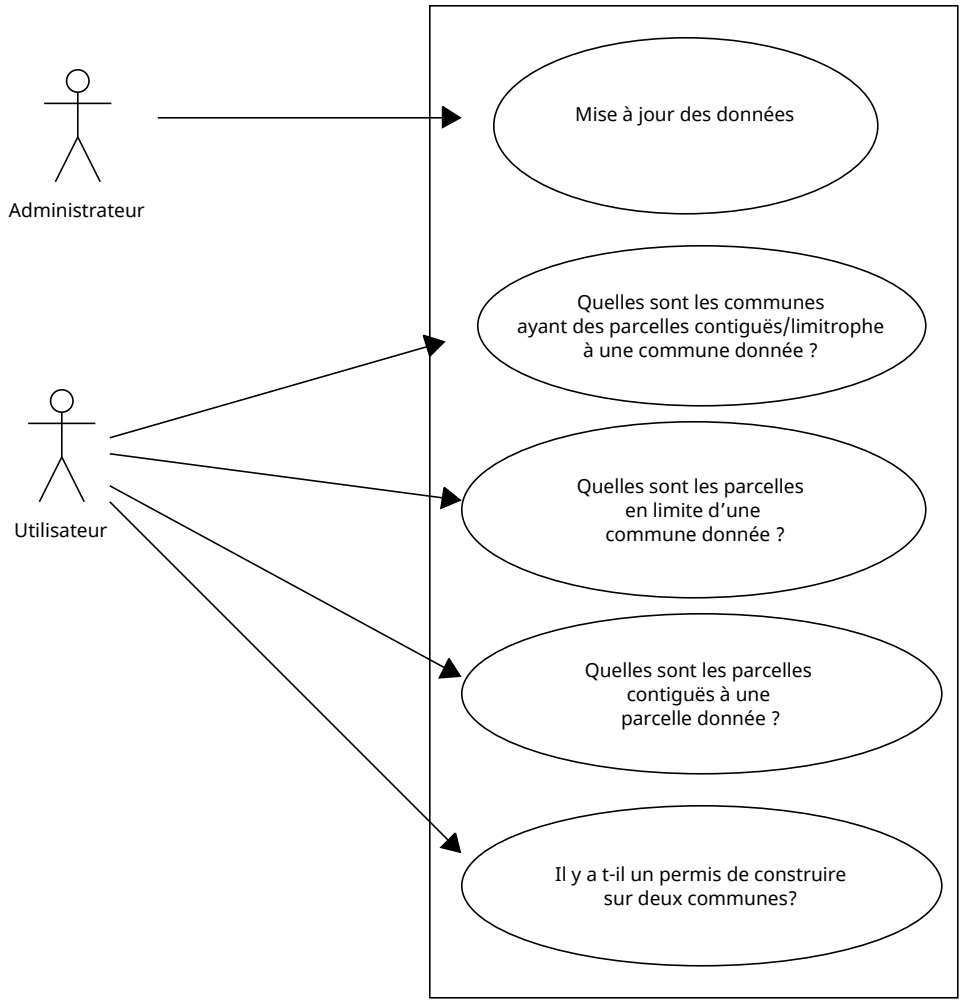


Diagramme de cas d'utilisation.

Le diagramme d'état ou d'activité qui va suivre va permettre de détailler les différentes étapes possibles du programme.

Les parties en vertes représentent les actions de l'admistrateur tandis que la partie en bleu représentent les actions de l'utilisateur. Nous avons choisis de faire un unique diagramme d'activité car le début de chaque reqûete est identique et que certaines reqûetes partagent les mêmes actions.

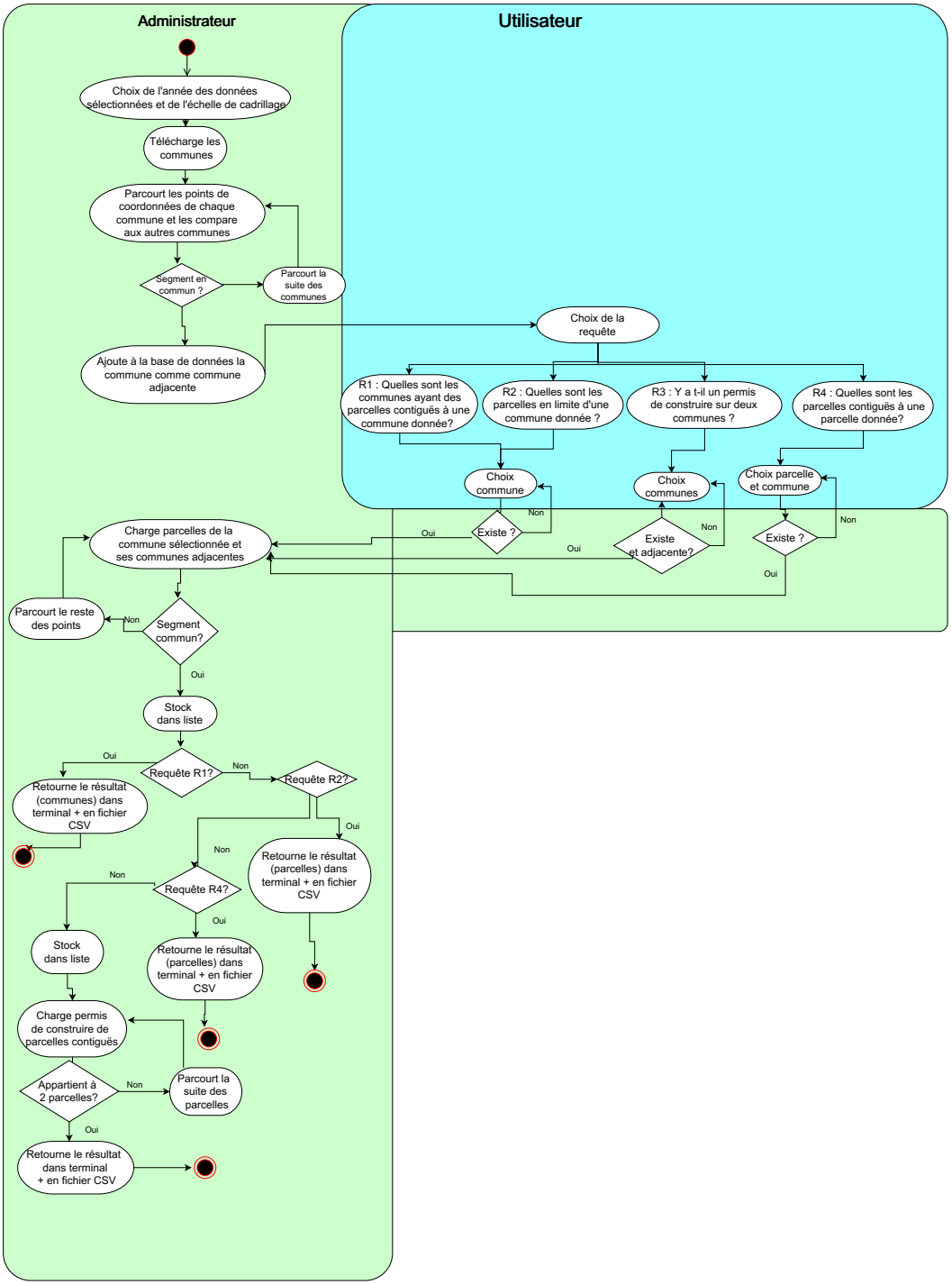


Diagramme d'activitée.

Les premières étapes de chaque réponse à une requête sont identiques. Elles correspondent sur

le diagramme d'utilisation au fait que l'administrateur mettent à jour et créer la base de donnée.
Pour cela, l'administrateur choisis l'année des données qui veut utiliser ainsi qu'une échelle de cadrillage. Cette échelle permet de faciliter l'importation des données : la France est découpé en carré de 10*10 par exemple, et chaque commune est importée dans ce carré avec ces coordonnées. Cela permet de gérer le problème de stockage de donnée.
Ensuite, le programme permet de trouver pour chaques communes ces communes adjacentes en comparant les coordonnées de celle-ci. Une commune est dit adjacente à une autre si elles partagent un segment en commun. On peut donc stocker dans la base de donnée, pour chaque commune ces communes adjacentes.

L'utilisateur intervient ensuite pour choisir une des requêtes, et suivant celles-ci il devra choisir une ou plusieurs communes et/ou une parcelle. Avant de continuer dans l'excution des reqûetes, le programme s'assurera que les communes et parcelles existent et dans le cas de la requête 3 (il y a-t-il un permis de constuire sur deux communes ?) que les communes sélectionnées sont adjacentes. Cela permet de renvoyer directement une erreur à l'utilisateur dû à une impossibilité d'exécution.

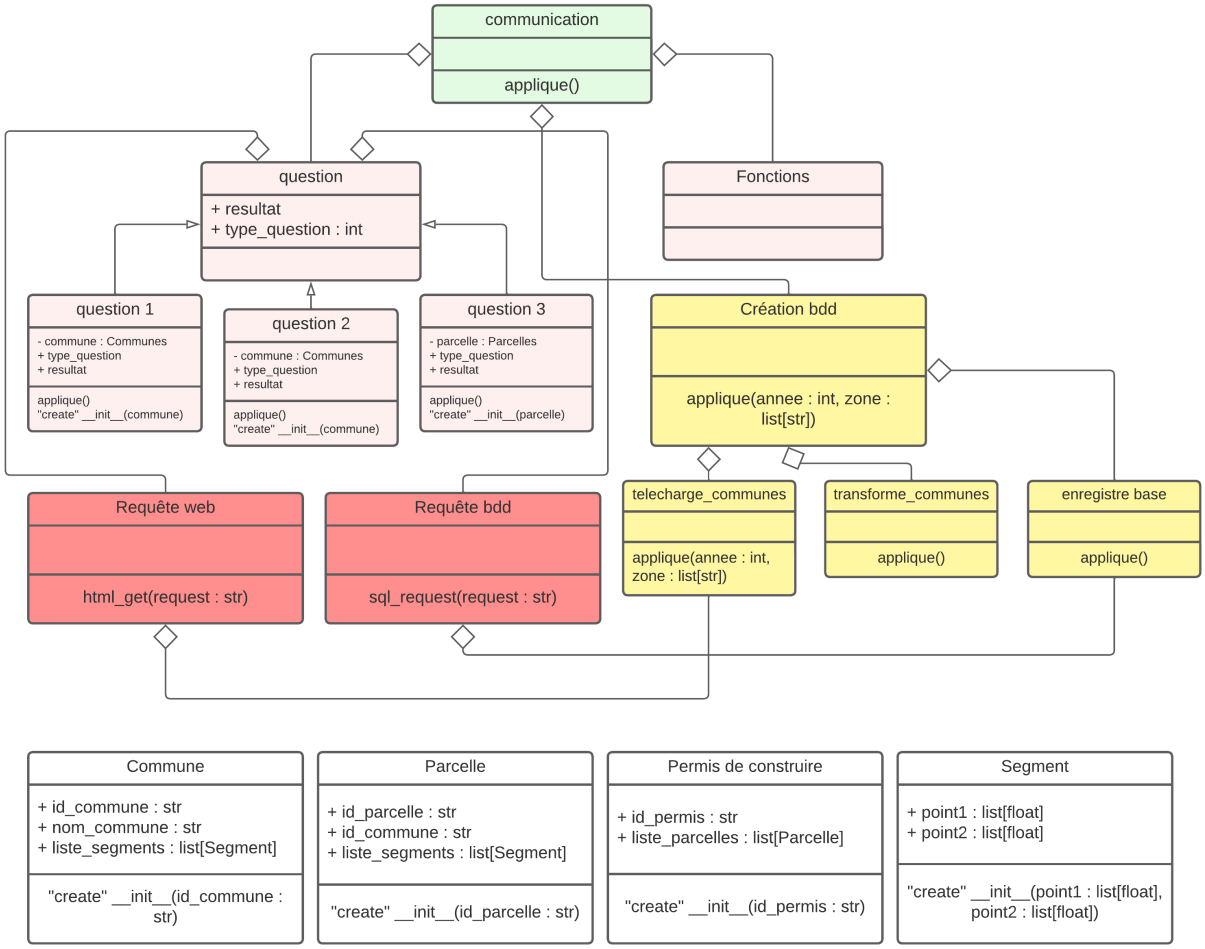
Pour répondre aux différentes reqûetes, le programme va récuperer les données sur internet et charger les parcelles de la commune (ou des communes) sélectionnée(s) et de ces communes adjacentes. Pour répondre aux deux premières requêtes le programme va comparer les coordonées des parcelles et voir s'ils partagent un segment en commun, si c'est le cas cela va être stockée. Si l'utilisateur a choisis la requête 1, cela va lui retourner dans son terminal et dans un fichier CSV les communes ayant des parcelles contigües à la commune choisis, et c'est la fin du programme. S'il a choisis la requête 2 cela va lui retourner dans son terminal et dans un fichier CSV les parcelles en limite de la commune choisis, et c'est la fin du programme. S'il a choisis la requête 4 cela va lui retoruner dans son terminal et dans un fichier CSV les parcelles contigües à la parcelle choisie, et c'est la fin du programme. Si ce n'est toujours pas le cas, les résultats des questions précédentes sont stockées dans une liste, et le programme va maintenant récupéré les données relatif au permis de construire présent dans les parcelles contigües stockées. Si ceux-ci appartiennent à 2 parcelles différentes, le programme retourne le résultat dans le terminal et dans un fichier CSV, et c'est la fin du programme, sinon le programme s'arrête en affichant qu'il n'y en a pas.

2.2 La base de donnée

- modèle physique de donnée - description

2.3 Conception

2.3.1 Diagramme de classes



2.3.2 Description des classes et de leurs méthodes

Le but de notre architecture de code est de distinguer 5 types de classes : La classe servant à communiquer avec l'utilisateur (vert), les classes contenant les calculs internes permettant de répondre aux questions (rose), celles qui permettent de communiquer avec le site web ou la base de donnée en ligne (rouge), les classes de création de la base de données (jaune) et enfin les classes objets métier (blanc).

Communication avec l'utilisateur (vert) :
La classe communication contient la fonction applique() qui lance un programme de discussion avec l'utilisateur par l'intermédiaire de questions / réponses sur la console. Celui ci peut choisir la question à laquelle il souhaite répondre ainsi que l'année de la base de données qu'il souhaite utiliser et la région qui doit être observée. C'est la seule classe qui entre directement en communication avec l'utilisateur. Les autres classes fourniront un travail interne au programme et automatique.

Classes de requêtes (rouge) :
Les classes de requêtes sont les classes qui permettent de communiquer soit avec le site web https ://cadastre.data.gouv.fr/ soit avec la base de données du serveur de l'école. La classe Requête web contient la méthode html_get() qui prend en entrée une requête html et l'envoie à l'API du site web. Cela peut avoir pour but de télécharger les données d'une année/région particulière ou d'obtenir des informations que nous avons choisi de ne pas stocker dans la base de données du serveur de l'école. La classe Requête bdd quant à elle permet de communiquer avec la base de données de l'école par l'intermédiaire de la méthode sql_request qui prend en entrée une requête sql et l'envoie au serveur de l'école. Cette classe a essentiellement pour but de lire des informations sur la base de données et de changer de base de données si l'utilisateur souhaite changer d'année.

Classes de création de la base de données (jaune) :
Quatre classes participent à la création de la base de données : La classe Création bdd contient la méthode applique qui prend en entrée l'année et la zone géographique voulue et va coordonner les autres classes en fonction de ces deux informations pour créer la base de données. Elle fera d'abord appel à la classe télécharge_communes et à sa méthode applique() qui en fonction de l'année et de la zone géographique va créer la requête html adéquate et la transmettre à la classe requête web qui l'enverra à son tour au site web. La base de données va alors être téléchargée depuis le site web et la classe transforme_communes prendra ce fichier contenant des données brutes pour en faire des données plus intelligibles. La fonction applique() de la classe permet ainsi de faire tous les calculs nécéssaires sur les données brutes pour les rendre plus facilement utilisables. C'est notamment cette fonction qui cherche les communes qui sont adjacentes. Une fois cette étape terminée, les données sont transmises à la classe enregistre_base dont la méthode applique() va permettre de transformer les données en une requête sql puis communiquer cette requête à la classe requête bdd qui la transmettra à son tour au serveur sql.

Classes de réponses aux questions (rose) :
Le programme a pour le moment récupéré les demandes de l'utilisateur et créé la bonne base de données en fonction de ces informations. L'utilisateur a également entré la question à laquelle il souhaite obtenir une réponse. Il faut maintenant répondre à cette question en faisant les bonnes lectures/calculs et les classes de réponses aux questions servent à cela. La classe question est une classe abstraite regroupant toutes les questions que l'utilisateur peut poser. Dans les classes questions 1, 2, 3, on déroule en fonction des attributs commune ou parcelle + numéro de question le programme de calcul de la réponse. Lorsqu'une fonction est nécéssaire, elle est récupérée dans les classes filles de la classe abstraite Fonctions. Une fois le résultat obtenu il est stocké dans l'attribut résultat (par défaut égal à None). La classe question termine par renvoyer le résultat à l'utilisateur sous forme de texte sur la console et/ou un csv.

Classes objets métiers (blanc) :
Pendant l'execution du programme, il pourra être utile de manipuler des objets simples et les classes objets métier servent à cela. Les objets sont Commune, Parcelle, Permis de construire et Segment. La Commune a pour attributs un identifiant, un nom et la liste des segments qui la composent. La parcelle a pour attributs un identifiant, l'identifiant de la commune dans laquelle elle se trouve et la liste de segments qui la composent. Le Permis de construire a un identifiant et la liste des parcelles sur lesquelles il se trouve et pour finir, le Segment a pour attributs les deux points qui le définissent. On peut également noter que nous définiront la notion d'égalité au sein de la classe Segment pour accepter le fait que [a,b] = [b,a].

2.4 Organisation des sous-systèmes techniques

3 Organisation et planning

3.1 Description des tâches

3.2 Diagramme de Gantt