

Travail de fin d’études

**InfraDraw : Architecture et dessin d’infrastructure**

Présenté par

**MAHAUX MATHIS**

En vue de l’obtention du grade de

**Bachelier en informatique de gestion**

**Année académique 2023 – 2024**





Travail de fin d’études

**InfraDraw : Architecture et dessin d’infrastructure**

Présenté par

**MAHAUX MATHIS**

En vue de l’obtention du grade de

**Bachelier en informatique de gestion**

**Année académique 2023 – 2024**



Tout d'abord, je tiens à remercier Mr. Frédéric Cammaratta, qui a accepté de m'accueillir au sein de l'entreprise, ainsi que Mr. Philippe De Ro qui a accepté de me superviser suite au souci de santé de Mr. Camaratta l'ayant empêché de continuer à me diriger.

Je tiens également à remercier Infrabel et son équipe, qui mettent tout en œuvre pour garder un espace de travail convivial et agréable, tout en restant sérieux.

Je remercie également Mr. Godefroid, mon maître assistant, pour son dévouement lors du suivi de la rédaction de ce travail de fin d'études. Ses conseils ont permis une bonne qualité de rédaction de ce travail.

Je remercie aussi Mr. Altares, qui a apporté son aide, à moi comme aux autres étudiants, dans la démarche de recherche de stage ainsi que dans les démarches administratives pour être intégré dans une entreprise en tant que stagiaire.

Je tiens a également remercier ma famille pour leur soutien durant toute la durée de mon stage chez Infrabel, ainsi que pour les conseils supplémentaires concernant la qualité de ce travail écrit.

Table des matières

[1. Introduction 9](#_Toc165108864)

[2. Description de l’entreprise 10](#_Toc165108865)

[2.1. Secteur d’activités et clientèle principale 10](#_Toc165108866)

[2.2. Évolution de l’entreprise 11](#_Toc165108867)

[2.3. Organigramme 12](#_Toc165108868)

[2.4. Environnement de travail 14](#_Toc165108869)

[2.5. Site Internet 14](#_Toc165108870)

[3. Position du problème 15](#_Toc165108871)

[3.1. Situation de départ 15](#_Toc165108872)

[3.2. Problème à résoudre 15](#_Toc165108873)

[3.3. Objectif du travail 15](#_Toc165108874)

[3.4. Services fournis par l’application 15](#_Toc165108875)

[3.5. Coûts en matériel, logiciel et ressources humaines 16](#_Toc165108876)

[4. Outils 17](#_Toc165108877)

[4.1. Logiciels utilisés 17](#_Toc165108878)

[4.1.1. Spyder 17](#_Toc165108879)

[4.1.2. MySQL Workbench 18](#_Toc165108880)

[4.1.3. Microsoft Visio 18](#_Toc165108881)

[4.2. Langages de programmation 20](#_Toc165108882)

[4.2.1. Python 20](#_Toc165108883)

[4.2.2. SQL 21](#_Toc165108884)

[4.2.3. HTML 21](#_Toc165108885)

[4.2.4. CSS 22](#_Toc165108886)

[4.3. L’API 23](#_Toc165108887)

[4.3.1. API 23](#_Toc165108888)

[4.3.2. REST 23](#_Toc165108889)

[4.3.3. JSON 24](#_Toc165108890)

[4.4. Bibliothèques Python utilisées 25](#_Toc165108891)

[4.4.1. requests 25](#_Toc165108892)

[4.4.2. pywin32 25](#_Toc165108893)

[4.4.3. mysql-connector-python 26](#_Toc165108894)

[4.5. Pip 27](#_Toc165108895)

[4.6. Conda 28](#_Toc165108896)

[4.7. Le Framework Django 30](#_Toc165108897)

[4.8. OpenShift 31](#_Toc165108898)

[5. Analyse et conception 32](#_Toc165108899)

[5.1. Diagramme de cas d’utilisation 32](#_Toc165108900)

[5.1.1. Enoncé 32](#_Toc165108901)

[5.1.2. Diagramme 32](#_Toc165108902)

[5.2. Diagramme Entité-Relation 33](#_Toc165108903)

[5.2.1. Enoncé 33](#_Toc165108904)

[5.2.2. Diagramme 36](#_Toc165108905)

[5.3. Diagrammes d'activité 37](#_Toc165108906)

[5.3.1. Programme général 37](#_Toc165108907)

[5.3.2. Premier programme 39](#_Toc165108908)

[5.3.3. Second programme 41](#_Toc165108909)

[6. Implémentation 43](#_Toc165108910)

[6.1. Premier programme 43](#_Toc165108911)

[6.1.1. Capture d’écran d’un extrait de code 43](#_Toc165108912)

[6.1.2. Explications 44](#_Toc165108913)

[6.2. Second programme 45](#_Toc165108914)

[6.2.1. Capture d’écran d’un extrait de code 45](#_Toc165108915)

[6.2.2. Explications 47](#_Toc165108916)

[6.2.3. Capture d’écran du résultat 48](#_Toc165108917)

[7. Conclusion 49](#_Toc165108918)

# Introduction

Lorsque l'on travaille dans le domaine de l'informatique pour une grande entreprise, on peut vite se rendre compte que celle-ci comporte dans son infrastructure une multitude de serveurs et autres appareils. De nos jours, ces machines sont devenues incontournables pour le bon fonctionnement d'une société, notamment car elles permettent un transport rapide et un stockage durable des informations au sein de celle-ci.

Même s'il existe des moyens de consulter différentes informations sur les serveurs présents dans l'infrastructure, comme leur nom, les logiciels hébergés par ceux-ci ou les applications les utilisant, cela peut parfois s'avérer fastidieux.

Infrabel, la société publique responsable de l'infrastructure ferroviaire belge s'est penchée sur cette problématique et a décidé de faire appel à un étudiant terminant son bachelier afin de la résoudre. Je suis celui qui a répondu à cet appel. En effet, j’ai décidé de poser ma candidature pour ce stage dans cette entreprise pour plusieurs raisons. Tout d'abord, le fait qu’Infrabel proposait de réaliser un programme générant un schéma de son infrastructure interne m’a attiré. Je trouvais intéressant d’en savoir plus sur la manière dont les informations circulent au sein d’une entreprise si conséquente. Ensuite, le langage de programmation utilisé lors du projet qui m’a été assigné est, pour la plus grande partie, Python, un langage fort et polyvalent qui m'intéresse beaucoup et que j'ai décidé d'apprendre parallèlement à mes études. Ce stage était donc une bonne opportunité pour moi d’améliorer mes connaissances et ma technique de ce langage.

Le projet réalisé lors de mes quinze semaines de stage consistait en plusieurs programmes informatiques : un programme de création et de remplissage de base de données, un programme de génération et de stockage de diagrammes sur un serveur utilisant les informations de la base de données et une application Web en à l’aide du Framework Django, qui s'occupe d'afficher le diagramme souhaité par l'utilisateur après avoir été le récupérer sur le serveur.

Ce travail de fin d’études reprend donc une description générale de l’entreprise Infrabel, une présentation des outils utilisés lors de la réalisation de ce projet et une analyse détaillée de l’application réalisée. Une illustration de l’implémentation et des solutions mises en œuvre y est également développée.

# Description de l’entreprise

## Secteur d’activités et clientèle principale



Figure 1 : Logo d'Infrabel

Infrabel est une société anonyme de droit public, appartenant au gouvernement belge. Elle est responsable de l’infrastructure ferroviaire de la Belgique, c’est-à-dire non seulement les rails, mais également les aiguillages, les caténaires (Câbles destinés à l’alimentation des moyens de transports électriques à captage du courant) et les passages à niveau. L’entreprise a réalisé un chiffre d’affaires de 24 117,95 millions d’euros à la fin de l’année 2022, contre 23 494,72 millions d’euros à la fin de l’année 2021, soit une augmentation de 623,23 millions d’euros, soit une augmentation du chiffre d’affaires de 2,7%. L’entreprise est composée de 9 557 employés en janvier 2024. Son siège social est basé à Saint-Gilles, une commune appartenant à la région Bruxelles-capitale.

La société fait donc partie du secteur ternaire, c’est-à-dire le secteur économique basé sur la réalisation de services. Les services qu’Infrabel rend sont multiples : construction de nouveaux éléments d’infrastructure sur le réseau ferré belge, recherche de nouvelles technologies pour moderniser ceux-ci comme pour aider à les entretenir, la maintenance de l’infrastructure ferroviaire ainsi que la coordination du trafic sur l’ensemble du réseau.

La clientèle de cette entreprise regroupe les entreprises qui souhaitent utiliser les voies ferrées belges, que ce soit pour le transport de passagers comme pour le transport de marchandises. Parmi les clients, nous pouvons compter la SNCB (Société Nationale des Chemins de fer Belges), qui est l’entreprise se chargeant du transport de passagers sur l’ensemble du réseau ferroviaire de notre pays et l’exemple le plus notable, mais également des entreprises telles que CFL Cargo, Crossrail Benelux, Deutsche Bahn, Europorte, Eurostar, Fret SNCF, HSL Polska, Lineas, Railtraxx, RTB CARGO, et bien d’autres encore. Dans ces entreprises clientes, nous pouvons trouver des entreprises de transport de marchandises (CFL Cargo, Fret SNCF…) ainsi que des entreprises de transport de passagers (Eurostar).

## Évolution de l’entreprise

Infrabel est une société plutôt jeune, qui a été créée en 2005. Voici quelques étapes importantes, passées comme futures, dans l’histoire de l’entreprise :

* Création :

Infrabel a été créée en 2005 suite à la réforme des anciens chemins de fer belges. Cette réforme visait à séparer les activités de gestion de l’infrastructure ferroviaire (Infrabel) de celles de l’exploitation des trains (SNCB/NMBS). L’objectif était de permettre une libre concurrence et d’améliorer l’efficacité.

* Modernisation et Innovation :

Infrabel a entrepris des initiatives de modernisation pour optimiser le réseau ferroviaire. Parmi celles-ci, la concentration des cabines de signalisation a permis de centraliser la gestion des signaux et d’améliorer la sécurité. Le déploiement du système ETCS (European Train Control System) a également été crucial pour prévenir les dépassements de signaux et garantir une meilleure gestion du trafic.

* Transformation avec GoBlue :

Le programme GoBlue a été lancé par l'entreprise pour améliorer les conditions de travail des collaborateurs. Il met l’accent sur l’autonomie, le travail en équipe et le dialogue constructif.

* Investissements futurs :

Concernant les futures intentions de l'entreprise, Infrabel a annoncé un investissement de 130 millions d’euros dans l’infrastructure du port d’Anvers au cours de la prochaine décennie.

## Organigramme



Figure 2 : Organigramme des différents chefs à la tête d'Infrabel

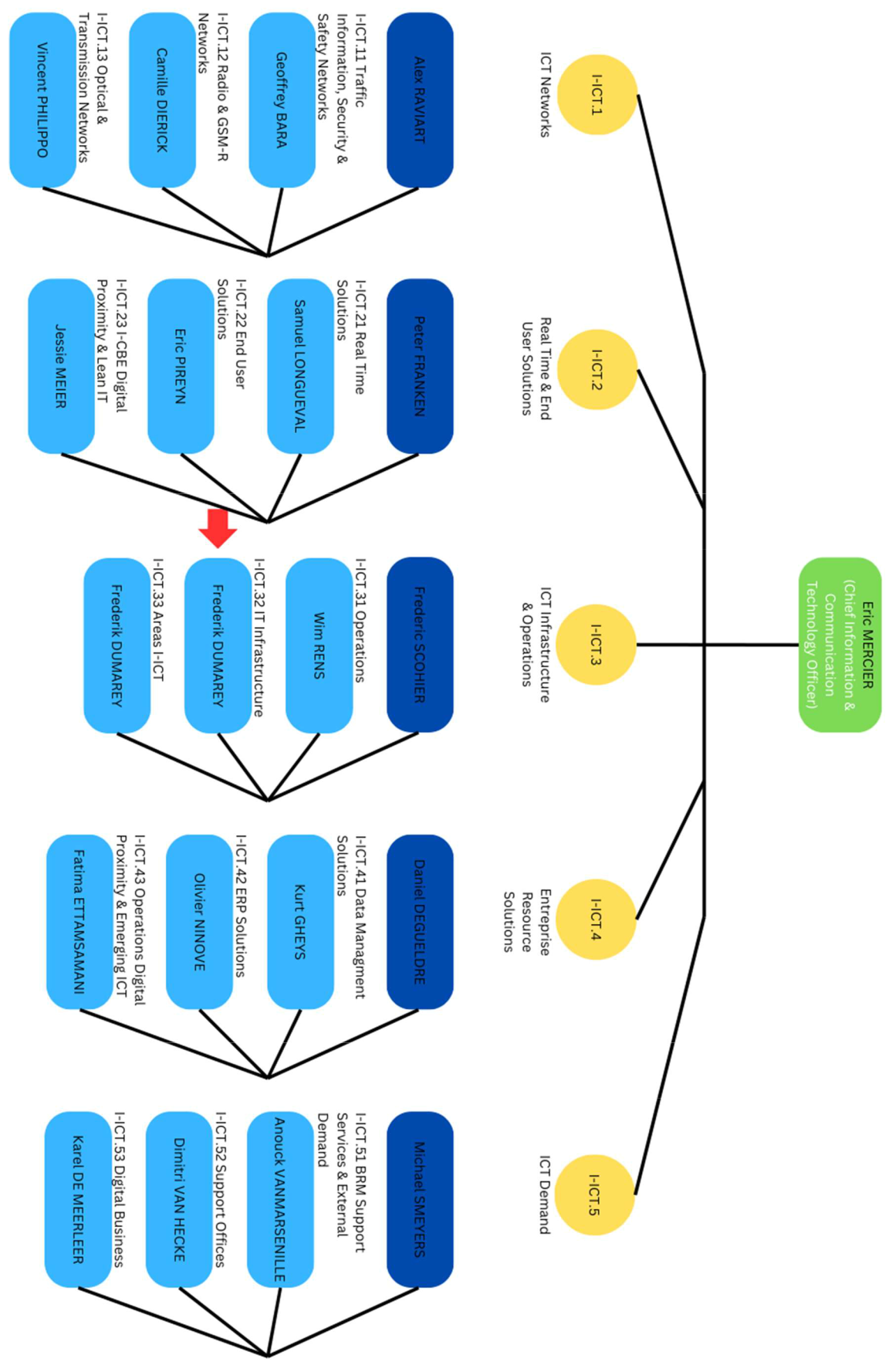


Figure 3 : Organigramme de la division sous la direction de Mr Eric MERCIER

Sur cet organigramme, nous pouvons voir que le CEO d’Infrabel Mr. Benoît GILSON est assisté par des Business Partners tels que Mme. Nathalie ROOSE et Mme. Katia BODART, des managers comme Mme. Aurélie TRAUBE et Mr. Carl DE SPIEGELEER ainsi que Mr. Serge DOZIN. Mr. GILSON dirige également plusieurs directeurs, chacun à la tête d’un domaine spécifique. Chaque directeur gère de plusieurs sections ayant un thème en lien avec le domaine dont le directeur est le responsable. Chaque section est dirigée par une personne attitrée et chaque section est divisée en plusieurs sous-sections, qui sont chacune dirigées par un manager. Dans notre cas, notre directeur est donc Mr. Eric MERCIER, la personne responsable de notre section est Mr. Frederic SCOHIER et le manager de notre sous-section est Mr. Frederik DUMAREY. La flèche rouge sur l’organigramme indique notre position dans l’entreprise.

## Environnement de travail

En temps normal, les employés de la section ICT de l’entreprise travaillent au bureau. Les bureaux du bâtiment sont agencés en open-space, c’est-à-dire que les différents postes de travail ne sont pas séparés par des cloisons, incitant ainsi les travailleurs à collaborer plus entre eux. Sur chaque bureau se trouve un ordinateur fixe et deux écrans. Les employés peuvent y connecter l’ordinateur portable qui leur a été fourni par l’entreprise, leur permettant ainsi d’avoir trois écrans au total.

De manière générale, les travailleurs travaillent de 8h à 16h, du lundi au vendredi. Mais il est possible qu’ils puissent assouplir leur horaire. Par exemple, ils peuvent sortir du bureau une heure plus tôt et travailler une heure plus tard un autre jour. Les heures d’entrée et de sortie des employés sont enregistrées par un système de pointage. Pour accéder ou pour partir de l’espace de travail, nous devons porter sur nous un badge à scanner. Cependant, nous ne devons pas seulement scanner notre badge à l’entrée du bâtiment mais également aux portes menant à différents endroits du bâtiment, comme les bureaux, la cafétéria, etc. Cela assure un environnement de travail plus sécurisé en évitant les intrusions.

## Site Internet

<https://infrabel.be/fr>

Nom du site : Infrabel

Titre du site : Home | Infrabel

Descriptif du site : Le site recense différentes informations sur l’entreprise (Jobs, business, à propos, chantier, actualités, écoles, …)

# Position du problème

## Situation de départ

Avant mon arrivée, notre équipe a finalisé le développement d’une API. Cette API a pour but de consulter l’UCMDB (Uniform Configuration Management Database) d’Infrabel, également existante au préalable, pour y exploiter des informations sur les différents éléments composant l’infrastructure informatique de l’entreprise (Serveurs, bases de données, etc.).

## Problème à résoudre

Le problème est que les organigrammes représentant les différents éléments de l’infrastructure et les relations entre eux sont, pour l’instant, fait à la main via le logiciel Microsoft Visio. Cela n’est donc pas entièrement fiable de par le facteur humain, car un humain, contrairement à une machine, est plus susceptible de se tromper de numéro de port, de nom de serveur, etc.

## Objectif du travail

Le but de l’application Web est de générer un diagramme représentant l’infrastructure informatique utilisée par une application appartenant à Infrabel dont le code applicatif a été fourni. L’application va communiquer avec l’UCMDB via une API créée au préalable, puis remplir une base de données spéciale avec les informations nécessaires à l'élaboration des diagrammes. L'application va ensuite consulter cette base de données pour générer les diagrammes de chaque environnement de l'application contenant les serveurs présent dans celui-ci ainsi que les Software Servers hébergés sur chacun des serveurs.

Le seul délai fixé pour la réalisation de ce projet est fin mai, date de la fin de ma période de stage et le seul moyen déployé est une base de données contenant les informations nécessaires à l’élaboration du diagramme.

Le projet a commencé lors de ma venue. Je suis la seule personne à qui l’on a confié cette tâche. Je dois donc réaliser cette application pour l’équipe "Architecture" du département ICT de l’entreprise.

## Services fournis par l’application

L’application rendra service aux employés souhaitant rapidement savoir quels sont les serveurs, bases de données, etc. utilisés par une certaine application dans un but de maintenance, d’ajout d’appareil dans l’infrastructure, ou autre, et ce de manière plus fiable. Elle permettra également un gain de temps considérable car il est beaucoup plus long de faire un diagramme d’infrastructure parfois fort complexe que de seulement demander à une machine de le générer.

## Coûts en matériel, logiciel et ressources humaines

Tout le matériel supplémentaire qui a été nécessaire à la réalisation de ce projet consiste en, simplement, un ordinateur portable et de ses accessoires rendant son utilisation plus confortable et ergonomique.

Les logiciels qui ont dû être installés sont :

* Spyder, un environnement de développement pour Python, spécialisé en analyse de données ;
* Visio, un outil de création de diagrammes développé par Microsoft;
* MySQL Workbench, un logiciel de gestion et de modélisation de bases de données.

Spyder et MySQL Workbench sont des logiciels open-source gratuits et Visio est un logiciel dont la licence était déjà existante dans l’entreprise. Une demande d'obtention du logiciel a dû être faite auprès de l'entreprise. Les seuls coûts engendrés sont donc les coûts de l’ordinateur portable et du matériel l’accompagnant (Chargeur, casque, souris et sac à dos).

J'ai été le seul responsable de la réalisation de ce projet. Bien entendu, je ne pouvais mener à bien ce projet seul. J'ai dû communiquer avec mes collègues afin d’obtenir les informations nécessaires au bon accomplissement de ma tâche.

# Outils

## Logiciels utilisés

### Spyder



Figure 4 : Logo de Spyder

Spyder est un environnement de développement intégré (IDE) open-source conçu principalement pour les scientifiques et les ingénieurs travaillant avec Python. Il offre des fonctionnalités telles que l'édition de code, la gestion des variables, le débogage, l'exécution de scripts et la consultation de données via des outils intégrés comme les consoles IPython et les explorateurs de variables. Spyder est souvent utilisé pour le développement et l'analyse de données grâce à son intégration des bibliothèques populaires telles que NumPy, SciPy et Matplotlib, facilitant ainsi le travail dans des domaines tels que la science des données.

Raison du choix : Spyder est un environnement de développement très complet de par son débogueur intégré et son éditeur de code puissant. En plus de cela, il facilite la tâche du développeur grâce à son interface intuitive et bien organisée, sa coloration syntaxique, son indentation automatique et ses suggestions de code. Il est également très personnalisable : le développeur peut configurer l'interface et les raccourcis de clavier comme bon lui semble.

Moment de l'utilisation pendant le projet : Spyder a été utilisé pour développer la totalité des programmes réalisés dans le cadre du projet. Aucun autre environnement de développement n'a été utilisé.

### MySQL Workbench



Figure 5 : Logo de MySQL Workbench

MySQL Workbench est un outil de conception, de développement et d'administration de bases de données MySQL. Il offre une interface graphique pour créer, modifier et gérer des schémas de base de données, des tables, des requêtes SQL et des utilisateurs. Il permet également de visualiser les relations entre les tables, d'exécuter des requêtes SQL, de surveiller les performances et de sauvegarder ou restaurer des bases de données.

Raison du choix : Grâce à son interface conviviale, ses fonctionnalités complètes et surtout son intégration avec MySQL, MySQL Workbench semblait le meilleur choix, compte tenu du fait que la base de données utilisée dans le cadre du projet est une base de données MySQL.

Moment de l'utilisation pendant le projet : MySQL Workbench a surtout été utilisé durant la réalisation du programme Python générant et remplissant la base de données. Afin de vérifier si tout se déroulait correctement.

### Microsoft Visio



Figure 6 : Logo de Visio

Microsoft Visio est un logiciel de création de diagrammes et de schémas techniques développé par Microsoft. Il permet aux utilisateurs de concevoir et de visualiser une variété de diagrammes, tels que des organigrammes, des plans de réseau, des diagrammes de flux, et plus encore. Visio offre une gamme d'outils pour dessiner des formes, connecter des éléments et ajouter du texte ou des données pour représenter visuellement des processus, des structures ou des relations. Il est largement utilisé dans les environnements professionnels pour la documentation, la planification de projets, la modélisation de données, et la communication visuelle des concepts complexes.

Raison du choix : Visio est le logiciel utilisé pour le dessin des diagramme pour la simple raison qu'il s'agit du logiciel de ce genre le plus facile à automatiser en Python, à l'aide de la bibliothèque "pywin32". Le logiciel propose une multitude de fichiers "pochoir" contenant plusieurs formes d'un certain type en fonction du genre de diagramme que l'on veut créer (Plan d'une infrastructure réseau, plan d'un bâtiment, …). Avec Visio, il est donc possible de créer tout un éventail de diagrammes remplissant des buts différents, automatiquement.

Moment de l'utilisation pendant le projet : Visio est naturellement intervenu lors de l'implémentation du programme Python s'occupant de générer les diagrammes d'infrastructure à partir des informations contenue dans la base de données créée et remplie par le premier programme, c'est-à-dire après la réalisation de ce dernier.

## Langages de programmation

### Python



Figure 7 : Logo de Python

Python est un langage de programmation polyvalent, open source, interprété et de haut niveau. Créé par Guido van Rossum et publié pour la première fois en 1991, il est réputé pour sa syntaxe claire et concise, ce qui le rend facile à apprendre et à lire. Python prend en charge différents paradigmes de programmation, notamment la programmation orientée objet, impérative et fonctionnelle. Il est utilisé dans un large éventail de domaines très variés tels que le développement Web, la science des données, l'intelligence artificielle, l'automatisation de tâches et bien d'autres. Python bénéficie d'une vaste communauté de développeurs qui contribuent à son écosystème en constante expansion grâce à une grande variété de bibliothèques et de Frameworks.

Raison du choix : En raison de sa facilité d'utilisation et de sa polyvalence, Python est un excellent choix pour réaliser différents programmes informatiques remplissant une fonction différente les uns par rapport aux autres, le tout sans devoir changer de langage de programmation.

Moment de l'utilisation pendant le projet : Python a été utilisé pour développer la majeure partie de l'ensemble des programmes créés pour le projet, y compris l'application Web, créée avec le Framework Django, utilisant Python. Par conséquent, il a été utilisé pendant toute la durée du stage.

### SQL



Figure 8 : Logo de SQL

SQL, acronyme de "Structured Query Language", inventé par Donald D. Chamberlin et Raymond F. Boyce au début des années 70, est un langage de programmation conçu pour gérer et manipuler des bases de données relationnelles. Il permet d'effectuer diverses opérations telles que l'insertion, la mise à jour, la suppression et la récupération de données à partir de ces bases de données. SQL utilise des commandes telles que SELECT, INSERT, UPDATE et DELETE pour interagir avec les données stockées dans les tables. Il offre également des fonctionnalités avancées, notamment les jointures pour combiner les données de plusieurs tables, la clause WHERE pour filtrer les résultats et les fonctions de regroupement pour effectuer des calculs sur des ensembles de données.

Raison du choix : En raison de sa performance permettant une réactivité élevée des applications l'utilisant, ses fonctionnalités de sécurité avancées assurant l'intégrité et la confidentialité des données, sa simplicité et son évolutivité, SQL est choisi par les entreprises dans la grande majorité des cas.

Moment de l'utilisation pendant le projet : SQL est intervenu au moment de la réalisation des deux scripts Python générant la base de données et dessinant les diagrammes qui seront consultés par l'application Web. Son utilisation a été donc parallèle à celle de Python durant le développement de ces deux programmes.

### HTML



Figure 9 : Logo de HTML

HTML, ou "Hypertext Markup Language", est un langage de balisage utilisé pour créer des pages Web. Il définit la structure et le contenu d'une page en utilisant des balises pour indiquer les différents éléments tels que les titres, les paragraphes, les images et les liens. Ces balises sont interprétées par les navigateurs Web pour afficher le contenu de manière appropriée. HTML permet donc de créer des sites Web statiques et dynamiques en se combinant avec d'autres langages comme CSS pour le style et JavaScript pour l'interactivité.

Raison du choix : HTML a été utilisé non seulement car Django utilise des templates HTML pour générer le contenu des pages Web dynamiquement, mais surtout en raison de sa compatibilité avec de nombreux navigateurs Web, son intégration avec d'autre technologies comme CSS et Javascript, et la facilité de maintenance des sites réalisés à l'aide ce langage. HTML permet donc de créer de sites ou applications Web dynamiques et accessibles par un large public.

Moment de l'utilisation pendant le projet : HTML a été utilisé lors du développement de l'application Web, après la finalisation des deux programmes Python nécessaires à la création des éléments qu'elle doit consulter.

### CSS



Figure 10 : Logo de CSS

CSS, ou "Cascading Style Sheets", est un langage de programmation utilisé pour styliser et mettre en forme les éléments HTML d'une page Web. Il permet de contrôler l'apparence visuelle des éléments tels que le texte, les images et les mises en page. En utilisant des sélecteurs et des propriétés, CSS permet aux développeurs de définir des styles comme la couleur, la taille, la police, les marges et les espacements entre les éléments. Grâce à la séparation des contenus et de la présentation, CSS facilite la maintenance et la gestion des styles sur un site Web.

Raison du choix : Ce langage doit naturellement être utilisé en parallèle avec HTML, afin de définir l'apparence du site ou, dans le cas du projet, de l'application Web.

Moment de l'utilisation pendant le projet : CSS à été naturellement utilisé en parallèle à HTML, durant le développement de l'application Web.

## L’API

L’API à laquelle le programme de création de la base de données fait appel est une API REST, c’est à dire qui a été développée en suivant les conventions et les principes REST. Passons en revue ces différentes technologies et techniques intervenant dans le fonctionnement de cette API. Les réponses prodiguées par ce type d’API sont le plus souvent sous forme de JSON.

### API

Avant tout, une API (Pour "Application Programming Interface") est un ensemble de règles et de protocoles à suivre afin que deux programmes informatiques puissent communiquer entre eux de manière standardisée et structurée, sous la forme de demandes et de réponses. Elle définit le format des données et les méthodes à utiliser afin que les deux programmes puissent s’échanger des ressources entre eux correctement et efficacement.

Les API sont fortement utilisées dans le cadre du développement de logiciels. Elles sont le plus souvent utilisées pour intégrer des fonctionnalités d’un programme tiers à un autre programme. Par exemple, lorsqu’un site Web contient un objet contenant une carte Google Maps, celui-ci utilise une API effectuant des requêtes vers Google Maps.

### REST

#### Description

REST, ou "Representational State Transfer", est un style d'architecture pour les systèmes distribués, souvent utilisé dans le développement Web. Fondamentalement, il repose sur l'idée de traiter les ressources (comme des pages Web ou des données) comme des objets accessibles via des URLs. Avec REST, les clients (comme les navigateurs Web ou les applications mobiles) peuvent effectuer des opérations CRUD (Create, Read, Update, Delete) sur ces ressources en utilisant les méthodes HTTP standards : GET pour récupérer des données, POST pour créer de nouvelles ressources, PUT ou PATCH pour mettre à jour des ressources existantes, et DELETE pour les supprimer. En évitant de conserver l'état côté serveur et en mettant l'accent sur l'utilisation efficace des requêtes HTTP, REST simplifie la communication entre les composants d'une application distribuée, favorisant ainsi la scalabilité et la fiabilité.

#### Pourquoi REST?

Dans le cadre du projet, REST a été choisi car, parmi ses façons d'agencer les données, nous retrouvons JSON, un format de données léger et facile à utiliser avec Python, qui comporte plusieurs packages ou bibliothèques permettant de traiter ce type de données. De plus, REST est plus facile à utiliser lorsqu'il est associé à JSON que les autres alternatives, comme SOAP.

### JSON

#### Description

A black and grey logo

Description automatically generated

Figure 11 : Logo de JSON

JSON, ou "JavaScript Object Notation" est un format de données léger et facile à lire utilisé pour échanger des informations entre les applications. Il est basé sur une structure clé-valeur, où les données sont organisées en paires clé-valeur. Les données peuvent être représentées sous forme de chaînes de caractères, de nombres, de tableaux ou d'objets imbriqués. JSON est largement utilisé dans le développement Web pour la communication entre serveur et client, ainsi que pour le stockage et le transfert de données structurées.

#### Pourquoi JSON?

Dans l'élaboration de l'application, JSON a été préféré aux autres formats de données tels que XML, car il est plus rapide et plus simple à utiliser, ce qui est utile lors d'un projet à durée limitée, ou lorsque la productivité est un objectif, comme c'est le cas au sein d'une entreprise.

## Bibliothèques Python utilisées

Durant la réalisation du projet, plusieurs modules, packages et librairies de Python ont dû être importés afin d'implémenter les fonctionnalités souhaitées. Voici les plus importants.

### requests

La bibliothèque Python "requests" permet d'envoyer facilement des requêtes HTTP. Il offre une interface permettant d'effectuer des requêtes HTTP de manière simple, que ce soit pour récupérer des données depuis un serveur Web ou pour interagir avec une API. Il permet d'envoyer des requêtes GET, POST, PUT, DELETE, entre autres, avec seulement quelques lignes de code Python. Par exemple, pour effectuer une simple requête GET, il suffit simplement d'utiliser la fonction "requests.get()" en passant l'URL cible comme argument.

Utilité dans le projet : Cette bibliothèque est intervenue dans la cadre de la communication entre l'UCMDB et le programme de création et d'insertion dans la base de données, plus précisément pour communiquer avec l'API permettant d'aller chercher les informations nécessaires sur l'UCMDB.

### pywin32

La bibliothèque Python "pywin32" fournit un accès complet aux fonctionnalités de l'API Windows et permet donc aux développeurs Python d'interagir avec le système d'exploitation de Microsoft en utilisant des fonctions natives telles que la manipulation de fichiers, la création de processus, la gestion des fenêtres, l'accès au registre, et bien d'autres. Elle offre une interface de programmation puissante pour automatiser des tâches système, créer des applications Windows et manipuler des ressources du système d'exploitation. Cet outil facilite également l'intégration de Python aux applications Windows existantes, notamment les logiciels Microsoft Office tels que Excel, Word, Outlook, et bien d'autres. Elle permet aux développeurs d'automatiser ces applications, de manipuler leurs données et d'étendre leurs fonctionnalités à l'aide de scripts Python. Cela offre aux utilisateurs la flexibilité d'utiliser la langage pour des tâches telles que le traitement de données, la génération de rapports ou l'automatisation de tâches répétitives dans des applications Office.

Utilité dans le projet : Cette bibliothèque a été utilisée dans le but d'avoir accès aux fonctionnalités de Windows, dont l'automatisation de l'application Visio, un logiciel Microsoft Office, via Python.

### mysql-connector-python

"mysql-connector-python" est une bibliothèque développée et entretenue par Oracle qui permet aux développeurs Python d'interagir avec une base de données MySQL à partir de leurs programmes. Ce connecteur offre une interface pour exécuter des requêtes SQL telles qu'insérer des données, mettre à jour des enregistrements et effectuer d'autres opérations de gestion de base de données en écrivant du code Python. Il est conçu pour être compatible avec les versions récentes de Python et de MySQL, offrant ainsi une solution fiable et performante pour l'accès aux bases de données MySQL depuis Python. "mysql-connector-python" utilise le protocole MySQL natif pour communiquer avec le serveur de base de données, ce qui garantit une compatibilité élevée avec les fonctionnalités de MySQL. Il prend également en charge les fonctionnalités avancées telles que les transactions, les curseurs et les gestionnaires de contexte.

Utilité dans le projet : Cette bibliothèque a été utilisée pour remplir et pour interagir avec la base de données contenant les informations nécessaires à l'élaboration des diagrammes. La base de données étant une base de données MySQL, cette librairie est un bon choix car elle est conçue pour fonctionner avec ce système de gestion de base de données, permettant ainsi un communication fluide entre le code Python et le serveur. De plus, sa fiabilité est élevée puisqu'elle est conçue par Oracle, la compagnie derrière MySQL, signifiant qu'elle est à jour au niveau des fonctionnalités et des patches de sécurité en tout temps.

## Pip

Une image contenant cube, conception

Description générée automatiquement

Figure 12 : Logo de pip

Pip est un gestionnaire de paquets pour Python, permettant d'installer, mettre à jour et gérer des bibliothèques et des outils Python. Il simplifie le processus en téléchargeant automatiquement les dépendances nécessaires depuis le Python Package Index (PyPI) et en les installant dans l'environnement Python local. En utilisant une ligne de commande, les utilisateurs peuvent facilement installer des packages Python en exécutant des commandes simples comme "pip install <nom\_du\_package>". Cela facilite grandement le développement et la gestion des projets Python en automatisant la gestion des dépendances.

Raison du choix : En raison de son intégration étroite avec Python, sa facilité d'utilisation, son large éventail de packages, sa capacité à gérer automatiquement les dépendances et le fait qu'il est possible d'isoler les dépendances des projets et de maintenir la cohérence des environnements de développement avec cet outil, pip est un choix judicieux car, grâce à ces fonctionnalités, il facilite le développement de produits digitaux et donc augmente la productivité de l'entreprise.

Moment de l'utilisation pendant le projet : pip est intervenu surtout durant l'élaboration des deux programmes de départ, afin d'installer les packages Python nécessaires à leur fonctionnement.

## Conda

Une image contenant Graphique, Police, graphisme, logo

Description générée automatiquement

Figure 13 : Logo de Conda

Conda est un gestionnaire de paquets et un environnement de gestion pour plusieurs langages de programmation, notamment Python. Il permet d'installer, mettre à jour et gérer des packages, ainsi que de créer et gérer des environnements virtuels. Contrairement à pip, un autre gestionnaire de paquets Python, Conda est plus polyvalent car il peut installer des paquets non-Python et gérer des dépendances hors de l'écosystème Python. Conda est souvent utilisé dans le domaine de la science des données et de l'apprentissage automatique pour gérer les dépendances de projets complexes. Il existe deux versions améliorées de Conda. Englobant Conda, la version de base, nous avons Miniconda, puis Anaconda, qui englobe le tout, comme l'explique le schéma ci-dessous.

Une image contenant cercle, texte, capture d’écran, Police

Description générée automatiquement

Figure 14 : Schéma représentant les versions de Conda et leurs particularités

Les deux versions ont chacune des caractéristiques qui leur sont propres. Passons-les en revue :

* Anaconda :

Anaconda est une distribution Python complète qui comprend Python lui-même, ainsi qu'un grand nombre de packages scientifiques et d'outils populaires utilisés dans des domaines tels que de la science des données, le machine learning, l'analyse de données, entre autres. En plus de Python, Anaconda inclut des packages comme NumPy, pandas, matplotlib, scikit-learn, Jupyter.

* Miniconda :

Miniconda, d'autre part, est une version plus légère et minimaliste d'Anaconda. Il comprend uniquement Python et conda, le gestionnaire de packages et d'environnements de Anaconda. Contrairement à Anaconda qui vient avec une grande collection de packages préinstallés, Miniconda permet de sélectionner et d'installer uniquement les packages dont le développeur a besoin, ce qui rend l'installation de packages plus légère et plus personnalisée.

Raison du choix : Si un développeur travaillant pour une entreprise veut simplifier la gestion des dépendances, garantir la portabilité des applications, avoir à disposition un large écosystème de packages, faciliter le contrôle des versions de ses applications et bénéficier d'un support communautaire et professionnel, Django est un choix très judicieux.

Moment de l'utilisation pendant le projet : Similairement à pip, cet outil a été utilisé pour installer les packages Python nécessaires, mais est intervenu dès le début du projet pour mettre en place un environnement virtuel de développement.

## Le Framework Django

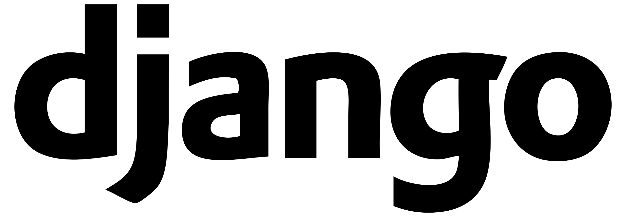


Figure 15 : Logo de Django

Django est un Framework Web open-source, écrit en Python, qui facilite le développement rapide et efficace d'applications Web complexes. Il suit le principe du modèle-vue-contrôleur (MVC), où les données de l'application sont manipulées par le modèle, présentées à l'utilisateur par la vue, et les actions de l'utilisateur sont gérées par le contrôleur. Il possède des composants prêts à l'emploi pour gérer les tâches courantes telles que la gestion des URL, les formulaires de saisie, l'authentification utilisateur, et bien plus encore. Le Framework permet également de mapper facilement les URL aux vues correspondantes. De plus, son ORM (Object-Relational Mapping) intégré simplifie l'interaction avec la base de données, en permettant aux développeurs d'utiliser des objets Python plutôt que des requêtes SQL. De plus, Django intègre des mécanismes de défense contre les attaques telles que les injections SQL, les attaques CSRF (Cross-Site Request Forgery) et les attaques XSS (Cross-Site Scripting).

Raison du choix : Grâce à son ORM, son système de routage et sa robustesse aux attaques, Django permet de développer des applications Web complexes et sûres rapidement, et plusieurs grandes entreprises telles que Instagram, Pinterest et bien d'autres, ont placé leur confiance en ce Framework.

Moment de l'utilisation pendant le projet : Django étant un Framework conçu pour développer des applications Web, son utilisation a commencé après que les deux premiers scripts Python furent finalisés.

## OpenShift

Une image contenant Graphique, Police, cercle, graphisme

Description générée automatiquement

Figure 16 : Logo de OpenShift

OpenShift est une plateforme de conteneurisation et de déploiement d'applications, développée par Red Hat. Elle permet aux développeurs de créer, déployer et gérer des applications de manière efficace et évolutive. OpenShift utilise la technologie des conteneurs pour encapsuler les applications et leurs dépendances, assurant ainsi une portabilité et une cohérence entre les environnements de développement, de test et de production. Il intègre également Kubernetes, un système d'orchestration de conteneurs, pour automatiser le déploiement, la mise à l'échelle et la gestion des applications. La plateforme offre également des fonctionnalités telles que la gestion des droits d'accès, la surveillance et les outils de développement collaboratif, ce qui en fait une solution complète pour le développement d'applications cloud-native.

Raison du choix : OpenShift permet d'augmenter l'agilité et l'efficacité du développement d'applications, de contrôler la cadence de l'innovation, de simplifier le déploiement, de simplifier la mise à l'échelle et la gestion des applications conteneurisées, et de donner aux équipes au sein d'une société une plateforme commune pour déployer et mettre à l'échelle leurs travaux sur le Cloud. C'est pour ces capacités que l'entreprise a choisi OpenShift pour déployer les différents travaux réalisés durant le stage sur leurs serveurs.

Moment de l'utilisation pendant le projet : OpenShift a été utilisé en fin de stage, afin de déployer l'ensemble des travaux réalisés durant le stage sur un serveur de l'entreprise pour utilisation.

# Analyse et conception

## Diagramme de cas d’utilisation

### Enoncé

Le diagramme des cas d'utilisation de l'application est assez simple puisque ses fonctionnalités ne se limitent qu'à consulter le stockage du serveur sur lequel sont stockés les diagrammes de tous les environnements, générés grâce aux deux programmes Python hébergés sur ce même serveur. Il n'y a qu'un seul type d'acteur : l'employé. Celui-ci devra avoir le possibilité d'effectuer les actions suivantes:

* Encoder le code applicatif de l'application pour laquelle il souhaite consulter l'infrastructure utilisée;
* Choisir l'environnement de développement pour lequel il souhaite voir le diagramme représentant l'infrastructure utilisée par celui-ci;
* Consulter le diagramme souhaité.

### Diagramme

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Figure 17 : Diagramme de cas d'utilisation de l'application web

## Diagramme Entité-Relation

### Enoncé

La base de données (appelée "MY1788") est agencée d’une manière similaire à un arbre. Un type d’entité peut en englober plusieurs d’un sous-type, et ainsi de suite. Le diagramme Entité-Relation de cette base de données se trouve à la page 36. Voici une description détaillée des attributs de chaque entité présente dans la base de données :

Tout d’abord, nous avons l’application, qui englobe toutes les entités décrites plus bas. Les informations contenues dans la table sont comme suit :



Figure 18 : Tableau descriptif des colonnes de la table "A1788\_application"

Ensuite, nous avons les environnements. Un environnement appartient à une application. Les informations les concernant qui seront retenues sont :



Figure 19 : Tableau descriptif des colonnes de la table "A1788\_environment"

Après, nous avons les Computer Systems, qui sont les différents éléments de l’infrastructure. Ils appartiennent à un environnement. Les informations sur les Computer Systems qui seront stockées sont :



Figure 20 : Tableau descriptif des colonnes de la table "A1788\_computer\_system"

Enfin, nous avons les Software Servers, c’est-à-dire des logiciels permettant à l’appareil sur lequel ils sont installés de se comporter comme un serveur, peuvent être hébergés sur un Computer System. Les informations sur les Software Servers qui seront retenues en base de données sont :



Figure 21 : Tableau descriptif des colonnes de la table "A1788\_software\_server"

Il y a également une table séparée pour les adresses IP, liée à celle des Computer Systems. Les informations que l’on peut trouver dans cette table sont :



Figure 22 : Tableau descriptif des colonnes de la table "A1788\_ip"

La raison de la création de cette table est que la valeur de la colonne de l’adresse IP peut avoir une longueur très importante, car il peut y avoir plusieurs adresses IP, parfois un nombre très important, séparées par des points-virgules. Une table dédiée seulement à celles-ci semblait donc un choix judicieux.

Il existe aussi dans la base de données deux tables de liaison: l'une entre les environnements et les Computer Systems, et l'autre entre les environnements et les Software Servers. La raison de leur existence est que un Computer System peux appartenir à plusieurs environnements, et que un environnement peut contenir plusieurs Computer Systems. Il en va de même pour les Software Servers. Ces tables contiennent les informations suivantes :

* Table "A1788\_computer\_system\_environment" :



Figure 23 : Tableau descriptif des colonnes de la table "A1788\_computer\_system\_environment"

* Table "A1788\_software\_server\_environment" :



Figure 24 : Tableau descriptif des colonnes de la table "A1788\_software\_server\_environment"

Passons en revue les relations entre les différentes relations entre les différentes entités présentes dans la base de données :

* Entre les tables "A1788\_application" et "A1788\_environment" :

Relation non-identifiante, 1:n (Une application peut posséder plusieurs environnements de développement)

* Entre les tables "A1788\_environment" et "A1788\_software\_server" :

Relation non-identifiante, m:n à l'aide de la table "A1788\_software\_server\_environment" (Un environnement de développement peut posséder plusieurs Software Servers, et vice versa).

* Entre les tables " A1788\_software\_server " et "A1788\_computer\_system" :

Relation non-identifiante, 1:n (Un Computer System peut posséder plusieurs Software Servers).

* Entre les tables "A1788\_environment" et "A1788\_computer\_system" :

Relation non-identifiante, m:n à l'aide de la table "A1788\_computer\_system\_environment" (Un environnement de développement peut posséder plusieurs Computer Systems, et vice versa).

* Entre les tables " A1788\_computer\_system " et "A1788\_ip" :

Relation non-identifiante, 1:n (Un Computer System peut posséder plusieurs adresses IP).

### Diagramme

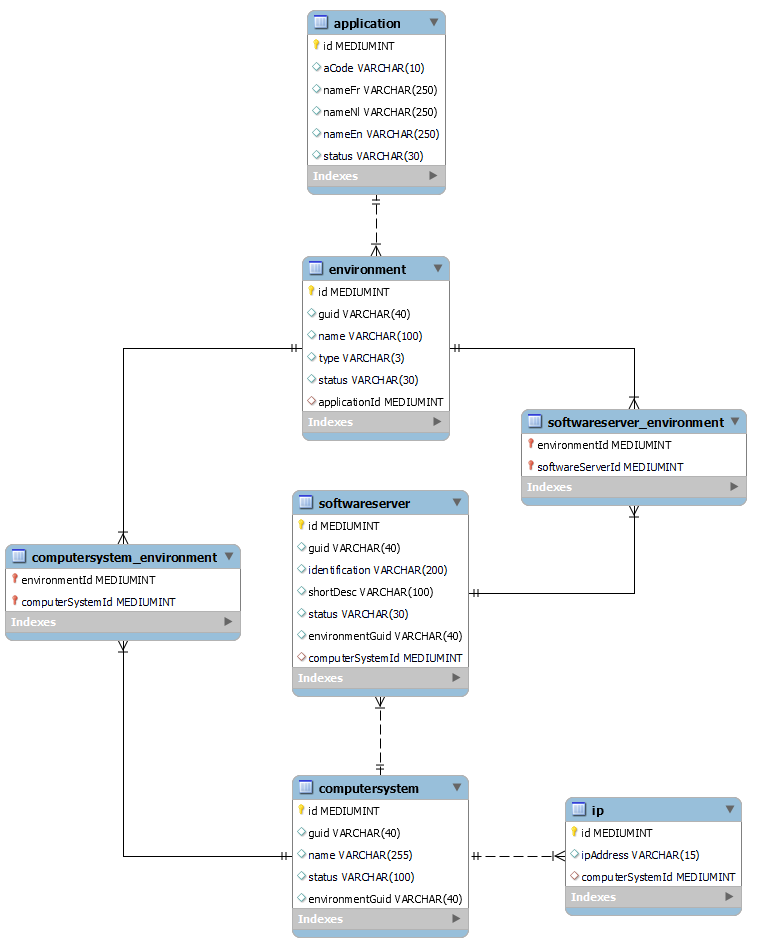


Figure 25 : Diagramme Entité-Relation de la base de données

## Diagrammes d'activité

### Programme général

#### Énoncé

Tout d'abord, il faut savoir que les deux autres programmes, celui de remplissage et de mise à jour de la base de données et celui de génération de diagrammes, seront exécuté à intervalles réguliers sur un serveur, sans interface graphique, donc sans intervention de l'utilisateur.

Le fonctionnement du programme principal est simple. Il se contente simplement de consulter les fichiers .svg générés et stockés sur le serveur. Il se déroule comme suit :

* Le code applicatif d'une application est demandé à l'utilisateur;
* Si le code encodé n'existe pas, l'utilisateur en est prévenu et il peut recommencer. Sinon, une liste déroulante affiche les différents environnements correspondant à l'application dont le code applicatif a été renseigné.
* Quand l'utilisateur a choisi un environnement dans la liste, le programme va chercher le fichier .svg correspondant sur le serveur et l'affiche sur l'ordinateur de l'utilisateur.

Les fonctionnement des programmes de gestion de la bases de données et de génération de diagrammes Visio sont expliqués plus en profondeur dans les sections qui suivent.

#### Diagramme

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

Figure 26 : Flow Chart du programme principal

### Premier programme

#### Énoncé

Le programme va d’abord réinitialiser la base de données, en supprimant le contenu de toutes les tables puis en réinitialisant l'auto-incrément des clés primaires à 1, avant de se connecter à l'UCMDB afin d'aller chercher le code applicatif de toutes les applications recensées dans celle-ci, et en faire une liste. Les éléments de cette liste seront utilisés en tant qu’argument dans les requêtes vers l’API.

Pour rappel, une requête vers l’API est une fonction, avec comme arguments l’URL de l’API, des paramètres et des en-têtes. Par exemple, une requête retournant les informations d’une application avec, en prime, la liste des Computer Systems associés à celle-ci ressemble à ceci :

params = {"RequestFor": "Application", "Detail": "D", "Identifiers": <Code applicatif de l'application>, "ComputerSystems": "All"}

headers = {"content-type": "application/json"}

response\_data = api\_request(<URL de l’API>, params, headers)

La valeur étant associée à la clé "Identifiers" dans le dictionnaire "params" sera celle du code applicatif courant.

Il va ensuite boucler dans la liste des codes applicatifs pour récupérer les données de l’application liée au code courant, puis les données sur les environnements, les Computer Systems et les Software Servers liés à l’application, ainsi que les informations sur les adresses IP de chaque Computer System.

Le programme va ensuite insérer les données récupérées dans la base de données destinée à être consultée par le second programme qui va générer des diagrammes sur Visio à partir des informations qu’elle trouvera dans cette base de données.

Les actions suivantes doivent également être effectuées, afin d’éviter des problèmes d’insertion dans la base de données :

* Avant l’insertion, il faut retirer les Software Servers qui sont hébergés sur un Computer System ne se trouvant pas dans la liste des Computer Systems qui a été retournée par l’API ;
* Il faut également vérifier si la liste des Computer Systems retournée par l’API n’est pas vide. Si c’est le cas, les Software Servers ne doivent pas être insérés.

#### Diagramme

A diagram of a process

Description automatically generated

Figure 27 : Flow Chart du premier programme

### Second programme

#### Énoncé

Le programme ouvre tout d’abord l’application Microsoft Visio à l’aide de la bibliothèque Python "PyWin32" et crée ensuite un nouveau document.

Le code et le nom de chaque application ainsi que les environnements contenus dans celles-ci sont récupérés dans la base de données "MY1788" (Celle qui a été remplie par le premier programme).

Pour chaque application, le programme va boucler à travers chacun de ses environnements. À chaque environnement, les actions suivantes sont effectuées :

* Les noms des Computer Systems et des Software Servers liés à l’environnement courant sont récupérés en base de données et stockés dans deux listes distinctes ;
* Si il existe au moins un Computer System au sein de l’environnement, une nouvelle page est créée, portant le nom de l’environnement ;
* Deux ensembles, l’un destiné à contenir les Computer Systems, l’autre ayant pour but d’englober les Software Servers, sont créés ;
* Pour chaque nom présent dans la liste des Computer Systems, un serveur portant le nom courant sera dessiné dans l’ensemble "Computer Systems" ;
* Pour chaque nom présent dans la liste des Software Servers, un rectangle portant le nom courant sera dessiné dans l’ensemble "Software Servers" ;
* À chaque diagramme d'environnement terminé, le programme le sauve sur le serveur en tant que fichier .svg, ayant pour nom le nom de l'environnement.

#### Diagramme

Une image contenant diagramme, Caractère coloré, capture d’écran, conception

Description générée automatiquement

Figure 28 : Flow Chart du second programme

# Implémentation

## Premier programme

### Capture d’écran d’un extrait de code

A screenshot of a computer code

Description automatically generated

### Explications

Cette fonction de récupération de données des Software Servers dans l’UCMDB est différente des autres. Il était impossible de récupérer les informations des Computer Systems hôtes de ces Software Servers à partir d’une seule requête vers l’API. Pour contourner ce problème, la fonction va effectuer différentes instructions :

1. La fonction va faire une première requête vers l’API (De type "Application") afin de récupérer les identifiants des Software Servers utilisés par l’application dont le code applicatif a été passé en argument. Ses identifiants seront stockés dans la liste "ss\_identifiers". Il faut noter qu’une requête de type "Application" ne permet pas de récupérer directement les informations sur les Computer Systems hébergeant les Software Servers. Il s’agit là de la source du problème.
2. La fonction va ensuite faire une requête vers l’API pour chaque identifiant retourné par la première requête. Ces requêtes sont cette fois de type "SoftwareServer". Définir le type de la requête sur "SoftwareServer" permet de récupérer les informations sur les Computer Systems hébergeant les Software Servers, contrairement aux requêtes de type "Application". Les données retournées seront stockées dans la liste de listes "all\_ss".
3. Enfin, les données de chaque entrée dans la liste "all\_ss" qui nous intéressent seront stockées dans la base de données qui sera utilisée par le second programme qui génèrera un diagramme Visio à partir des informations contenues dans celle-ci.

## Second programme

### Capture d’écran d’un extrait de code

A text on a white background

Description automatically generated

A text on a white background

Description automatically generated

### Explications

Le but de cette fonction est, comme son nom l’indique, de créer le document Visio. C’est dans celle-ci que se trouve la majorité de la logique du programme. Passons en revue les différentes actions qu’elle effectue :

1. La fonction va ouvrir l’application Visio, puis créer un document, puis y créer une page vierge.
2. Elle va chercher les informations nécessaires pour commencer à dessiner dans le document Visio, telles que le nom et le code de l’application, ainsi que les environnement liés à celle-ci.
3. Pour chaque environnement, elle va :

* Récupérer les noms des Computer Systems contenus dans l’environnement courant ;
* Récupérer les noms des Software Servers contenus dans l’environnement courant ;
* Créer une nouvelle page portant le nom de l’environnement courant, avec une étiquette mentionnant le nom de l’application ainsi que son code applicatif ;
* Définir une position initiale pour les rectangles correspondant aux ensembles "Computer Systems" et "Software Servers" qui seront dessinés lors de l’étape suivante ;
* Dessiner les deux ensembles : l’un pour les Computer Systems et l’autre pour les Software Servers ;
* Définir une position initiale qui sera mise à jour après le dessin de chaque Computer System ;
* Définir une position initiale qui sera mise à jour après le dessin de chaque Software Server ;
* Ensuite pour chaque Computer System :
  + Dessiner un rectangle dans l’ensemble "Computer Systems" portant le nom du Computer System courant ;
  + Mettre à jour la position avant de dessiner le rectangle suivant ;
* Enfin, pour chaque Software Server, répéter les mêmes étapes que pour les Computer Systems.

### Capture d’écran du résultat

Le résultat du programme sera donc un document avec une page par environnement avec, sur chaque page, deux ensembles, dont l’un contenant chacun des Computer Systems. Chaque Computer System dans le premier ensemble est relié à une liste de Software Servers lui correspondant qui se trouve dans le second ensemble.

Par exemple, voici le diagramme pour l’environnement portant le nom "A1788 A02" pour l’application "Discovery Tool BMC Atrium (BMC)" ayant pour code applicatif le code "A1788":

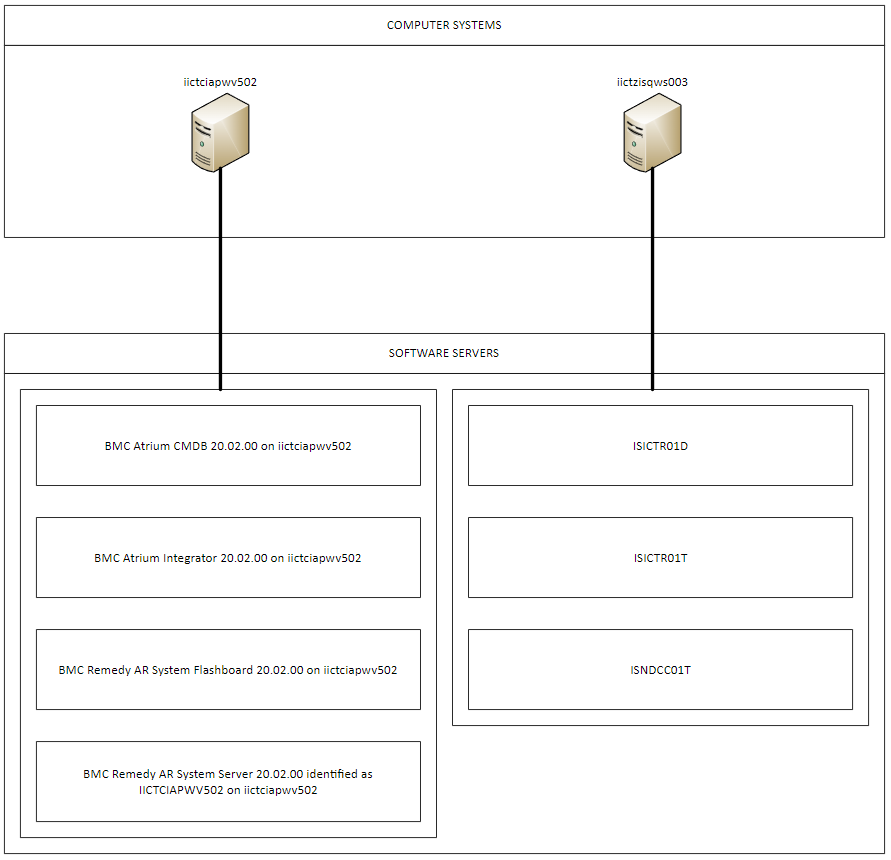


Figure 29 : Capture d'écran d'un fichier .svg résultat

# Conclusion

Durant mon stage chez Infrabel, j'ai eu l'occasion d'apprendre à utiliser le langage Python dans tout un éventail de situations, telles que l'utilisation d'une API, la création et l'interaction avec une base de données, l'automatisation d'une application installée sur Windows ou le développement d'une application Web. Je me suis également amélioré au niveau de la gestion des erreurs dans un programme Python, ainsi que l'optimisation de celui-ci afin qu'il n'utilise pas les ressources de l'appareil sur lequel il est exécuté dans l'excès. J'éprouve une grande satisfaction de cela car l'une des raisons pour laquelle j'ai décidé de me lancer dans ce stage est que je souhaitais approfondir mes compétences en ce langage.

Ce stage m'a également fait découvrir et maîtriser des outils très intéressants tels que Spyder, MySQL Workbench, Conda, Django et OpenShift. Ces outils vont être utiles pour améliorer ma productivité quant au développement de programmes ou d'applications écrites en Python.

En ce qui concerne l'adaptation de mes compétences acquises lors de mon cursus au milieu professionnel, j'ai pu faire cela avec mes connaissances en développement d'applications Web avec un Framework du cours d'Application Entreprise Multi-Tiers, de manipulation de bases de données du cours de Fichiers et Bases de Données et de conteneurisation de produits digitaux du cours d'Analyse et Conduite de projet avec OpenShift. Lors du cours d'Application Entreprise Multi-Tiers, nous avons utilisé Java comme langage. Dans le cas de mon stage, j'ai dû réaliser quelque chose de similaire, excepté que le langage utilisé est Python. Pour communiquer avec la base de données MySQL, des commandes en SQL ont été encodées. Ces commandes ont dû être incorporées au code Python.