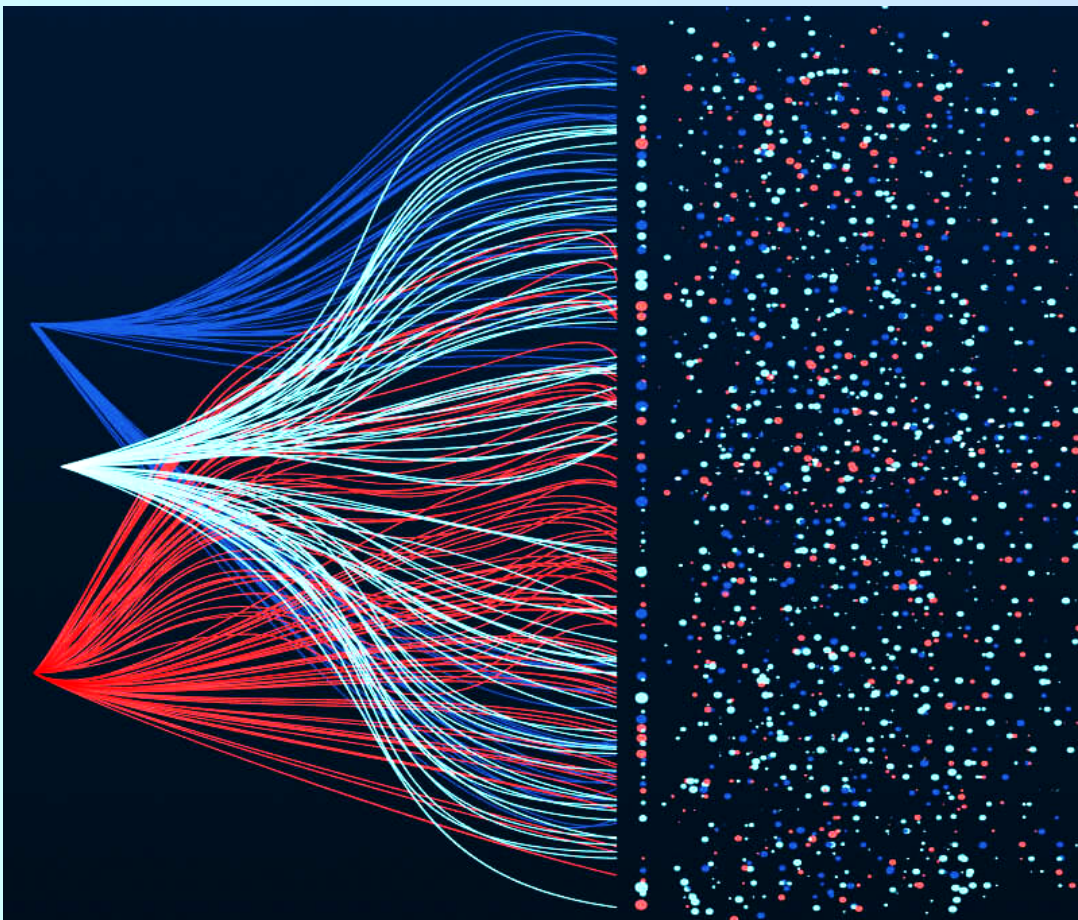


# CONCEPTION DÉTAILLÉE

LA DATA AUGMENTATION AU  
SERVICE DU MACHINE LEARNING  
**L3Q1-2022**



# CONCEPTION DÉTAILLÉE

## PROJET DATA AUGMENTATION

### Identification du document

<b>Référence du document</b> : L3Q1.C
<b>Version du document</b> : 1.0
<b>Date du document</b> : 20/02/2023
<b>Auteurs</b> : Abou Assaf Mahyr-Florian Daumont-Ouk Ilan' Ngo Alexandre Chemla Ethan

# Sommaire

<b>1. Introduction</b>	<b>4</b>
1.1 Objectifs	4
1.2 Documents de référence	5
1.3 Construction du projet informatique	5
1.3.1 Maître d'oeuvre (MOE)	5
1.3.2 Maître d'ouvrage (MOA)	6
<b>2. Architecture logicielle</b>	<b>6</b>
<b>3. Technologies et outils utilisés</b>	<b>8</b>
3.1 Back end	8
3.2 Front end	9
<b>4. Description des modules</b>	<b>11</b>
<b>5. Arborescence de l'application</b>	<b>12</b>
<b>6. Diagramme de cas d'utilisation</b>	<b>13</b>
<b>7. Diagramme de classes</b>	<b>14</b>
<b>8. Diagramme : cycle de vie</b>	<b>15</b>
<b>9. Organisation</b>	<b>16</b>
<b>10. Planification</b>	<b>17</b>
<b>11. Glossaire</b>	<b>18</b>
<b>12. Sources</b>	<b>19</b>

# 1. Introduction

La conception détaillée est un document essentiel dans la réalisation d'un projet, car il permet de garantir une bonne compréhension des aspects techniques et une description complète de l'architecture de l'application. Il vise à définir les différentes fonctions, classes, librairies et packages nécessaires pour traduire le cahier des charges et les recettes en une application web de qualité. En cela, il joue un rôle crucial pour assurer la réussite technique du projet.

Ce document constitue une première documentation de la réalisation technique du projet et sert de référence pour les développeurs tout au long de la réalisation du projet. Il est donc primordial que sa rédaction soit soignée et rigoureuse.

En général, la conception détaillée se divise en cinq grandes parties : la planification et l'organisation des tâches, la présentation des généralités de la méthode de développement utilisée, la décomposition du développement en phases successives, les méthodes et les outils de développement côté utilisateur, et les critères déterminant la qualité du logiciel. Cette structuration permet de faciliter la compréhension et la navigation dans le document.

Il convient donc de prendre le temps nécessaire pour rédiger la conception détaillée avec le plus grand soin afin de garantir la qualité technique du projet.

## 1.1 Objectifs

Le projet consiste en une application web proposant divers outils de traitement d'images permettant de générer des images légèrement modifiées. Cette application devra offrir la possibilité de créer un compte utilisateur, de s'authentifier et de déposer des ensembles de données.

L'interface de l'application web devra permettre à l'utilisateur de modifier les images en ajustant des paramètres tels que le bruit, la luminosité, le contraste, le recadrage, le flopping ou encore le zoom. Elle devra ensuite générer un nouvel ensemble de données à partir des images déposées par l'utilisateur. De plus, l'application devra permettre de consulter un historique des modifications d'images effectuées par l'utilisateur et de stocker les images dans une base de données. Enfin, il sera possible de télécharger les images depuis cet historique. Il convient de souligner que le développement de cette

application requiert une documentation détaillée qui permettra de comprendre les aspects techniques du projet, notamment l'architecture de l'application. Cette documentation doit fournir une description complète des fonctions, classes, librairies et packages nécessaires au développement de l'application. Elle doit également être en adéquation avec le cahier des charges et le document de recettes afin de garantir une réalisation technique de qualité. Cette documentation constitue la première étape de la réalisation technique du projet et sera une référence indispensable pour les développeurs tout au long du processus de développement.

## 1.2 Documents de référence

Les document suivants nous ont permis la réalisation de notre conception détaillée:

Référence du document	Type du document
L3Q1.A	Cahier des charges
L3Q1.R	Cahier de recettes

## 1.3 Construction du projet informatique

### 1.3.1 Maître d'oeuvre (MOE)

Responsables de la maîtrise d'œuvre:

- Abou Assaf Mahyr-Florian
- Daumont-Ouk Ilan'
- Ngo Alexandre
- Chemla Ethan

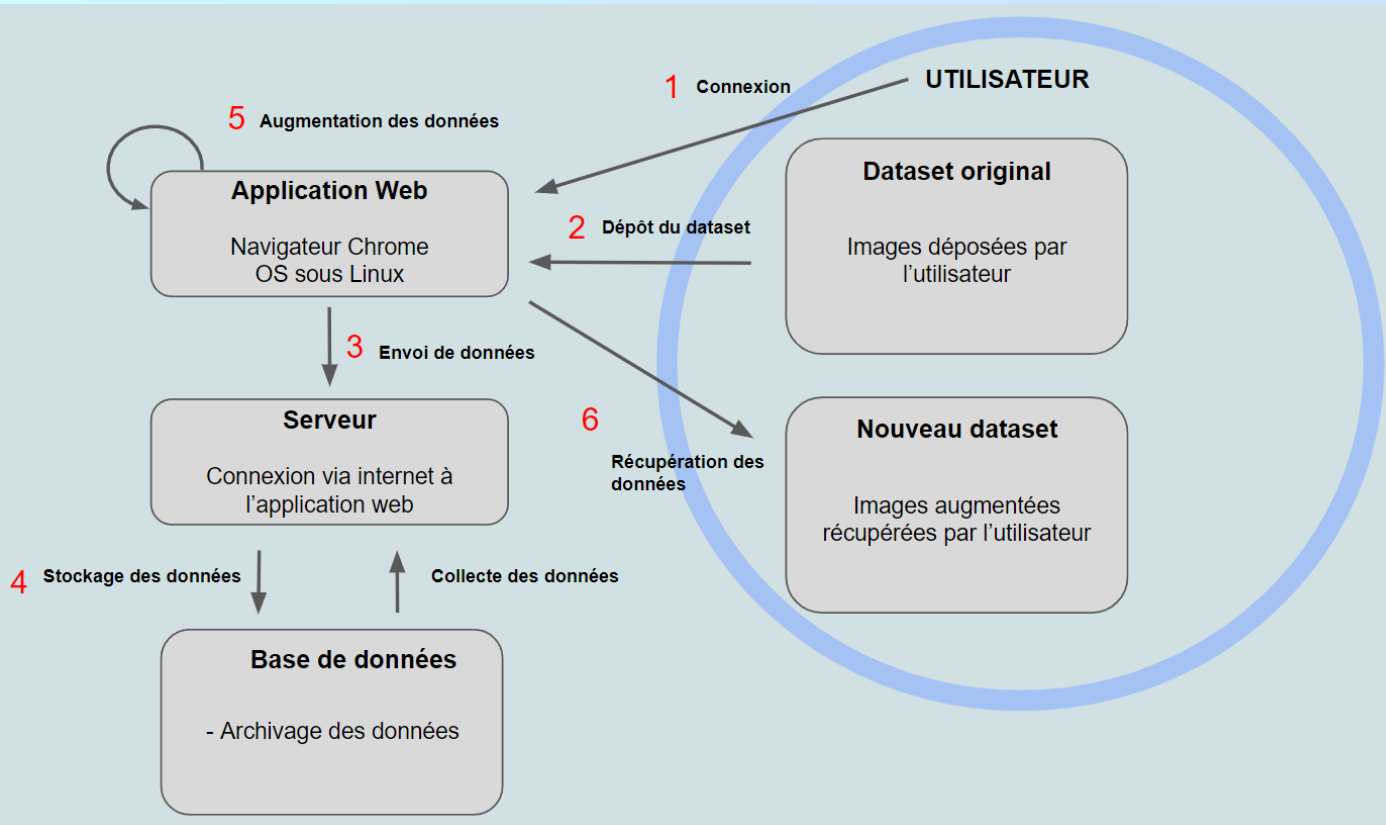
Un maître d'œuvre est chargé de déterminer les moyens techniques nécessaires pour concevoir le produit conformément aux spécifications définies par le maître d'ouvrage. Il est responsable de la mise en œuvre des différentes étapes du projet, de la coordination des différentes parties prenantes et de la supervision des travaux jusqu'à leur achèvement.

### 1.3.2 Maître d'ouvrage (MOA)

Chargé de la maîtrise d'ouvrage : Mr El Debs Elie

Le maître d'ouvrage est responsable de la gestion du projet tout au long de son cycle de vie. En tant que représentant du client, il est chargé de valider les différentes phases de développement du projet.

## 2. Architecture logicielle



Notre application web a été développée pour permettre aux utilisateurs de déposer leur propre dataset et d'appliquer des algorithmes d'augmentation de données dans le but d'améliorer leur modèle de machine learning.

L'architecture logicielle de l'application repose sur une architecture client-serveur basée sur les API, qui suit une structure en couches pour séparer les différentes responsabilités et faciliter la maintenance et l'évolution du système.



Le client est représenté par le navigateur web Google Chrome qui permet à l'utilisateur d'interagir avec l'application en naviguant sur les différentes pages et en effectuant des actions. Le serveur, quant à lui, est l'entité qui héberge l'application et fournit les ressources nécessaires pour son bon fonctionnement. Les requêtes HTTP sont utilisées pour communiquer entre le client et le serveur, et nous avons utilisé des requêtes POST et GET pour transmettre des données.

La couche de middleware sert d'interface entre les différentes couches de l'application, en s'assurant que les données sont transmises aux bonnes parties du système et en gérant les erreurs et les exceptions. L'API est la couche qui fournit une interface standardisée pour accéder aux différentes fonctionnalités de l'application. Elle est composée de différents modules qui contiennent des fonctions atomiques, c'est-à-dire des fonctions qui ne renvoient qu'un seul résultat pour une entrée donnée.

La séparation en couches permet de faciliter la maintenance et l'évolution de l'application, en permettant de modifier une couche sans avoir à toucher aux autres. Elle permet également de faciliter la collaboration entre les différents développeurs en charge des différentes parties de l'application. Les fonctions atomiques simplifient le code et facilitent la maintenance en permettant de réutiliser les fonctions dans différentes parties de l'application.

Le serveur gère les demandes des clients et fournit les réponses correspondantes. Chaque fois qu'un utilisateur souhaite effectuer une action, comme déposer un dataset ou appliquer un algorithme d'augmentation de données, une requête est envoyée au serveur qui effectue les actions nécessaires et renvoie une réponse. La base de données joue un rôle important dans l'application car elle permet de stocker les datasets des utilisateurs, ainsi que les informations d'authentification et de connexion.

Le processus de dépôt de dataset commence par l'authentification de l'utilisateur. Les informations d'authentification sont stockées dans la base de données et sont utilisées pour vérifier l'identité de l'utilisateur. Si l'utilisateur est authentifié avec succès, il est redirigé vers son espace personnel. Dans son espace personnel, l'utilisateur peut déposer son dataset en envoyant une requête HTTP au serveur. Le serveur reçoit les données et les stocke dans la base de données. Une fois le dataset stocké, l'utilisateur peut appliquer les algorithmes d'augmentation de données de son choix en sélectionnant les

options correspondantes. Le serveur reçoit les choix de l'utilisateur et applique les algorithmes sélectionnés.

Les algorithmes d'augmentation de données sont supervisés par des scripts Python qui communiquent avec le serveur pour fournir les données nécessaires à l'application et effectuer les traitements nécessaires. Une fois que les algorithmes d'augmentation de données ont été appliqués, l'utilisateur peut alors le télécharger depuis son espace personnel.

En résumé, l'application web est une solution conviviale qui permet aux utilisateurs de déposer leur propre dataset et d'appliquer des algorithmes d'augmentation de données à ce dernier pour améliorer leur modèle de machine learning. L'interaction entre le client et le serveur est facilitée par l'utilisation de requêtes HTTP et les données sont stockées dans une base de données structurée pour faciliter leur manipulation ultérieure.

### **3. Technologies et outils utilisés**

#### **3.1 Back end**

Le back-end d'un site web est la partie cruciale de l'architecture d'une application web, car elle est chargée de gérer les données et la logique de l'application. Cette partie de l'architecture est responsable de la gestion des bases de données, de l'authentification des utilisateurs, de la gestion des sessions, de la sécurité de l'application et de la manipulation des données qui transitent entre le client et le serveur. Dans ce contexte, il est essentiel de choisir les technologies les plus adaptées pour garantir des performances optimales et une expérience utilisateur satisfaisante.

Pour la gestion du serveur, le choix s'est porté sur le framework Flask, une solution open-source écrite en Python qui permet de construire des applications web. Flask fournit une structure de base pour le développement d'applications web en utilisant le protocole HTTP et permet de créer des serveurs web pour des applications web en utilisant le langage de programmation Python. Cette solution offre de nombreux avantages,



notamment sa flexibilité et sa facilité d'utilisation, qui en font une option de choix pour les petites et moyennes applications web.

Lorsqu'un utilisateur envoie une requête HTTP, le serveur Flask écoute cette requête sur une adresse IP et un port spécifiques, puis la traite et renvoie une réponse HTTP. Cette architecture permet ainsi de garantir une expérience utilisateur fluide et réactive tout en garantissant la sécurité des données qui transitent entre le client et le serveur.

En somme, le choix de Flask comme framework pour la gestion du serveur est un choix judicieux, car il offre une solution robuste, flexible et facile à utiliser pour le développement d'applications web. Cette technologie permettra d'assurer des performances optimales et une expérience utilisateur satisfaisante pour les utilisateurs du site web.

## 3.2 Front end

Le front-end d'un site web représente l'interface utilisateur visible et interactive. Il se compose principalement de code HTML, CSS et JavaScript, qui permettent de définir la structure, le style et le comportement d'une page web.

Le langage HTML permet de structurer le contenu de la page en définissant la hiérarchie des différents éléments tels que les titres, les paragraphes, les images, les vidéos, les formulaires, les tableaux, etc.

Le CSS, quant à lui, permet de définir le style visuel de la page web en utilisant des règles de style qui s'appliquent à des éléments spécifiques, en contrôlant l'apparence et la mise en page d'une page web. Sans le CSS, la page HTML n'aurait qu'un contenu brut et aucune mise en page.

Enfin, JavaScript est principalement utilisé pour ajouter de l'interactivité et de la dynamique à une page web en créant des scripts qui peuvent être exécutés côté client pour manipuler le contenu et le comportement de la page web. JavaScript permet également de capturer et de gérer les événements de l'utilisateur, tels que les clics de souris, les pressions de touche et les mouvements de souris, créant ainsi du dynamisme dans une page statique HTML.

Le code front-end est exécuté côté client, c'est-à-dire dans le navigateur de l'utilisateur. Il peut contenir une variété d'éléments interactifs tels que des formulaires, des boutons, des menus, des images, des vidéos, des animations, etc. Le langage CSS est crucial pour donner de l'attrait et de l'esthétisme à ces éléments en permettant un contrôle total sur la présentation de la page web. En effet, sans CSS, la page HTML serait simplement une collection de texte et d'images sans aucune forme de mise en page.

Pour développer la partie front-end d'un site web, il est essentiel de maîtriser ces trois langages. De plus, il existe des frameworks tels que Bootstrap qui facilitent le développement de sites web en proposant des modèles de conception et des éléments prédéfinis pour construire des pages web rapidement et facilement. Le front-end est crucial pour la conception de sites web modernes, car il permet aux utilisateurs d'interagir avec les informations présentées de manière attrayante, facile à utiliser et accessible.

## 4. Description des modules

Notre application web d'augmentation de la donnée est constituée de plusieurs modules qui ont chacun une fonction spécifique.

La **page d'accueil** sert de présentation de notre application et permet aux utilisateurs d'accéder facilement à différentes pages telles que la page de connexion, la page d'inscription, la page de contact et la page de présentation.

La **page d'inscription** comprend un formulaire permettant la création d'un espace personnalisé pour chaque utilisateur. Les données saisies dans ce formulaire seront envoyées au serveur, qui les traitera et mettra à jour la base de données en cas de nouveau compte utilisateur.

La **page de connexion** est destinée aux utilisateurs existants et leur permet de se connecter à leur compte en entrant leur adresse e-mail, leur nom, leur prénom et leur mot de passe. La page de réinitialisation du mot de passe permet de réinitialiser le mot de passe d'un utilisateur en cas d'oubli.

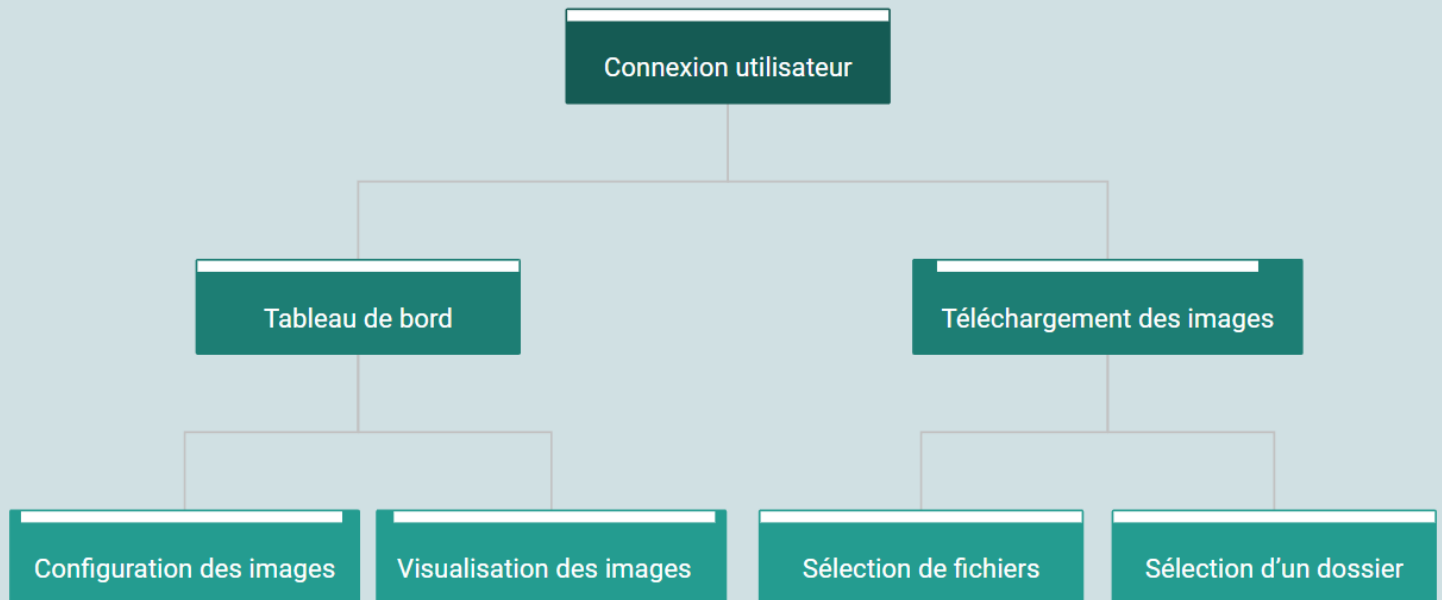
La **page d'espace personnel** est l'endroit où les utilisateurs peuvent déposer des images et configurer leur apparence. Cette page comprend plusieurs boutons permettant de modifier différents aspects de l'image tels que la luminosité, le contraste, le bruit, le zoom et le recadrage.

La **page de visualisation** des images affiche tous les datasets d'images déposés, qu'ils aient été augmentés ou non.

La **page de documentation** fournit des informations sur les fonctionnalités de notre site et les traitements qui peuvent être appliqués sur les datasets.

Enfin, la **page de présentation** de l'équipe de développeurs permet aux utilisateurs de mieux connaître les personnes derrière le développement de notre application. Pour se déconnecter, les utilisateurs peuvent simplement cliquer sur le bouton "Se déconnecter" situé sur la **page de déconnexion**.

## 5. Arborescence de l'application



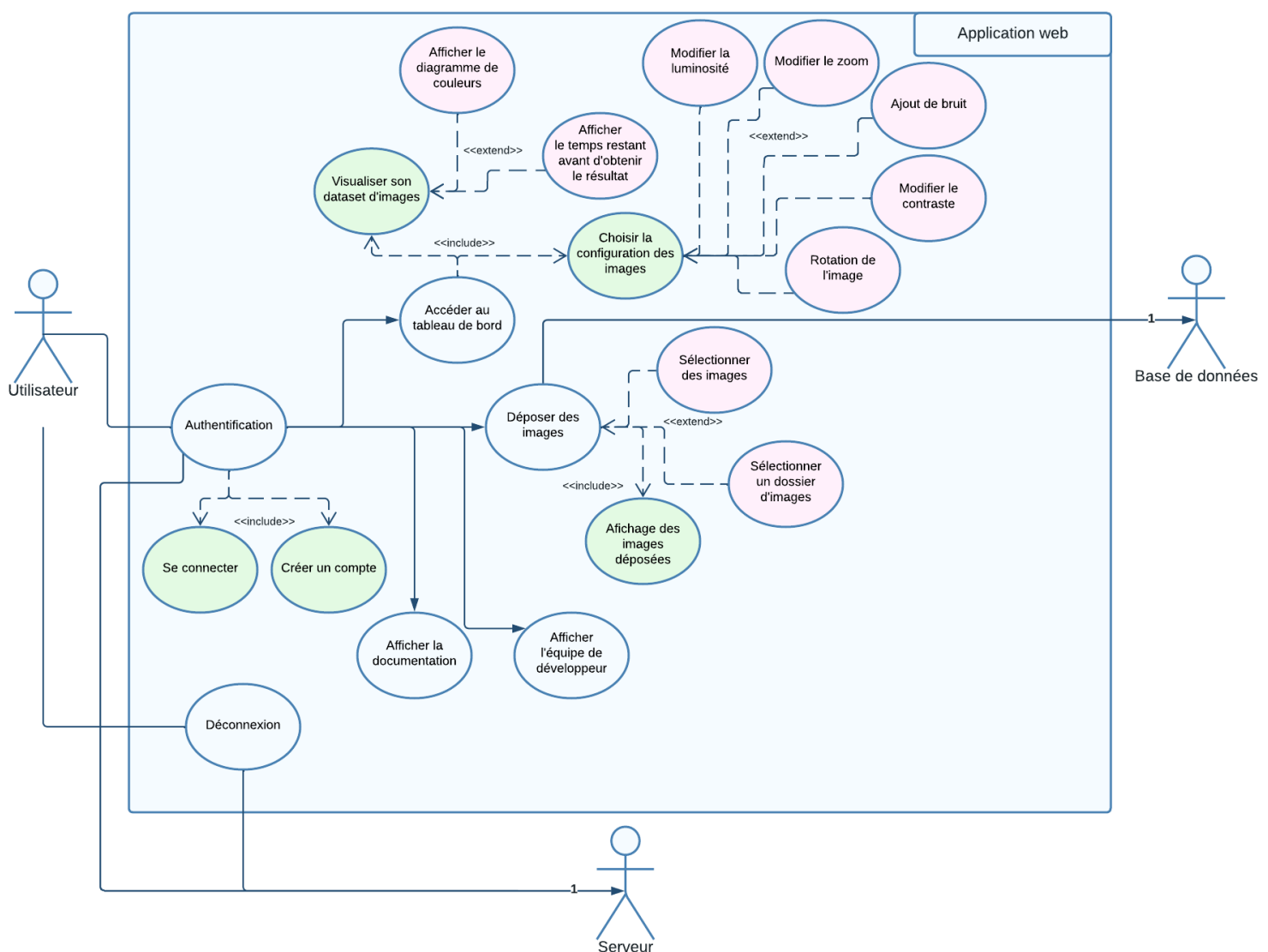
Le module de configuration d'images de l'application web d'augmentation de données permet à l'utilisateur de personnaliser les modifications qu'il souhaite apporter aux images déposées. Les options de configuration proposées comprennent la modification de la luminosité, du bruit, du contraste, du zoom et de la rotation des images. Cela offre à l'utilisateur une grande flexibilité pour adapter ses images selon ses besoins.

En outre, l'utilisateur peut visualiser les images augmentées et afficher un diagramme de couleurs pour mieux comprendre les modifications apportées. Un décompte du temps restant avant d'obtenir les images augmentées est également disponible, ce qui permet à l'utilisateur de mieux planifier son temps.

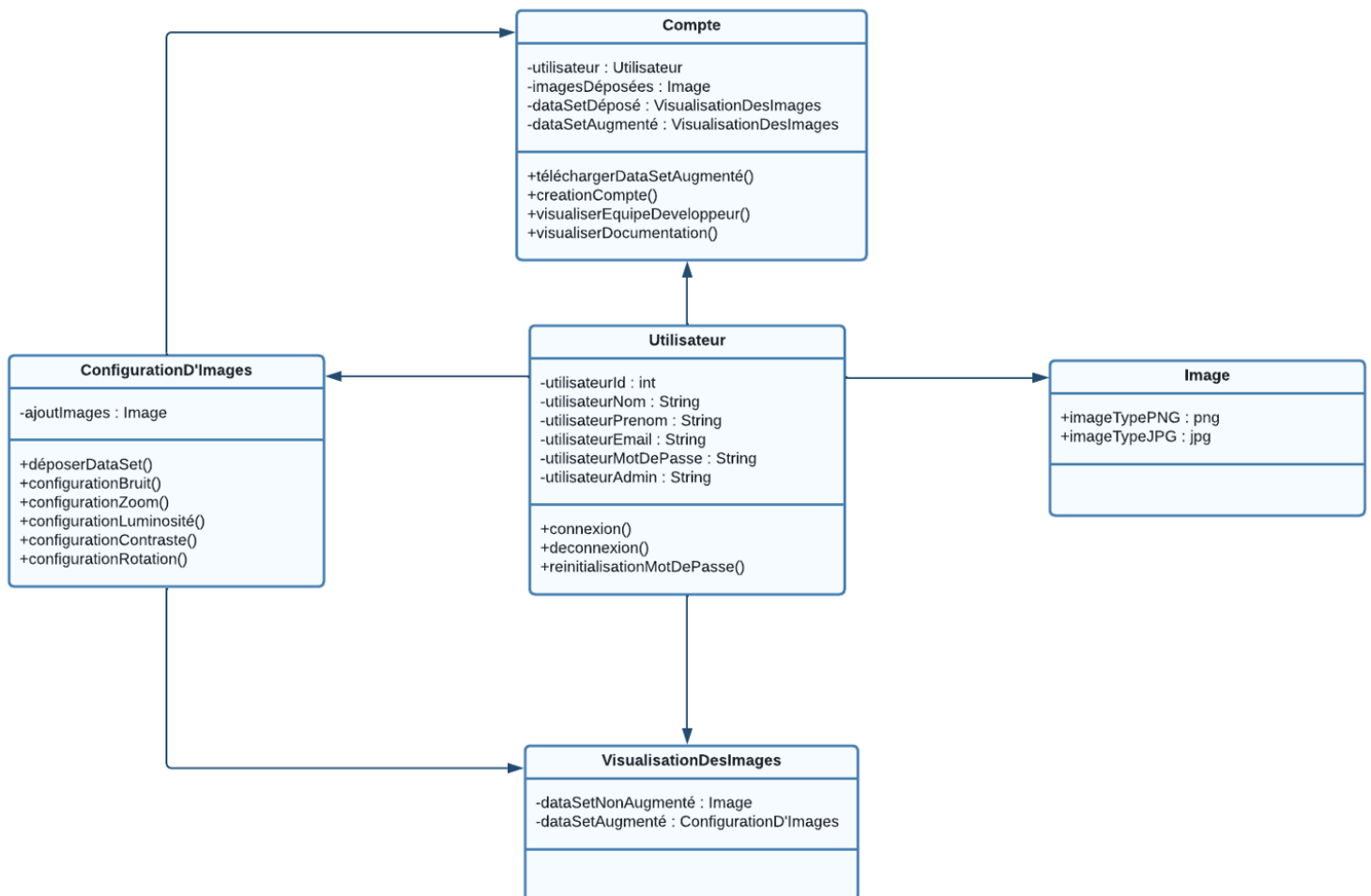
Le module de téléchargement d'images est également très pratique, car il permet à l'utilisateur de déposer facilement un dataset d'images en format PNG ou JPG depuis son appareil et de les envoyer à l'application web. L'option de sélection d'un dossier permet également à l'utilisateur de choisir

En somme, ces fonctionnalités offrent à l'utilisateur une expérience conviviale et personnalisée pour traiter ses images et les télécharger sur l'application web d'augmentation de données.

## 6. Diagramme de cas d'utilisation



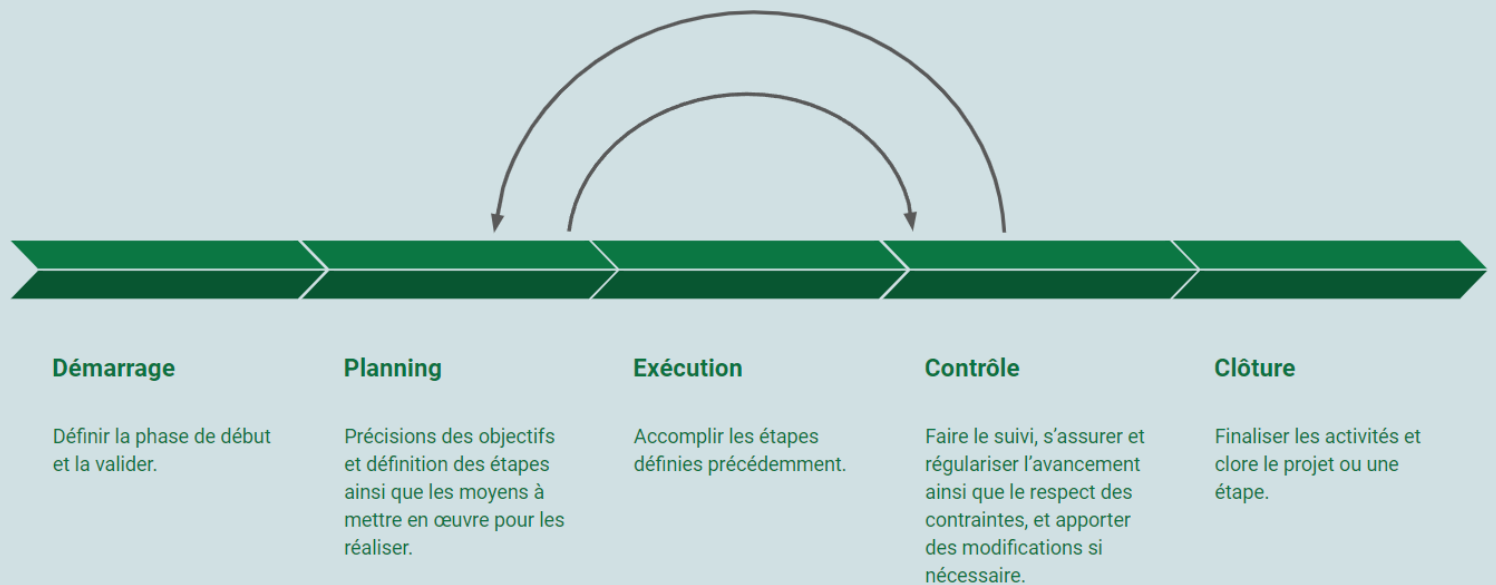
## 7. Diagramme de classes





## 8. Diagramme : cycle de vie

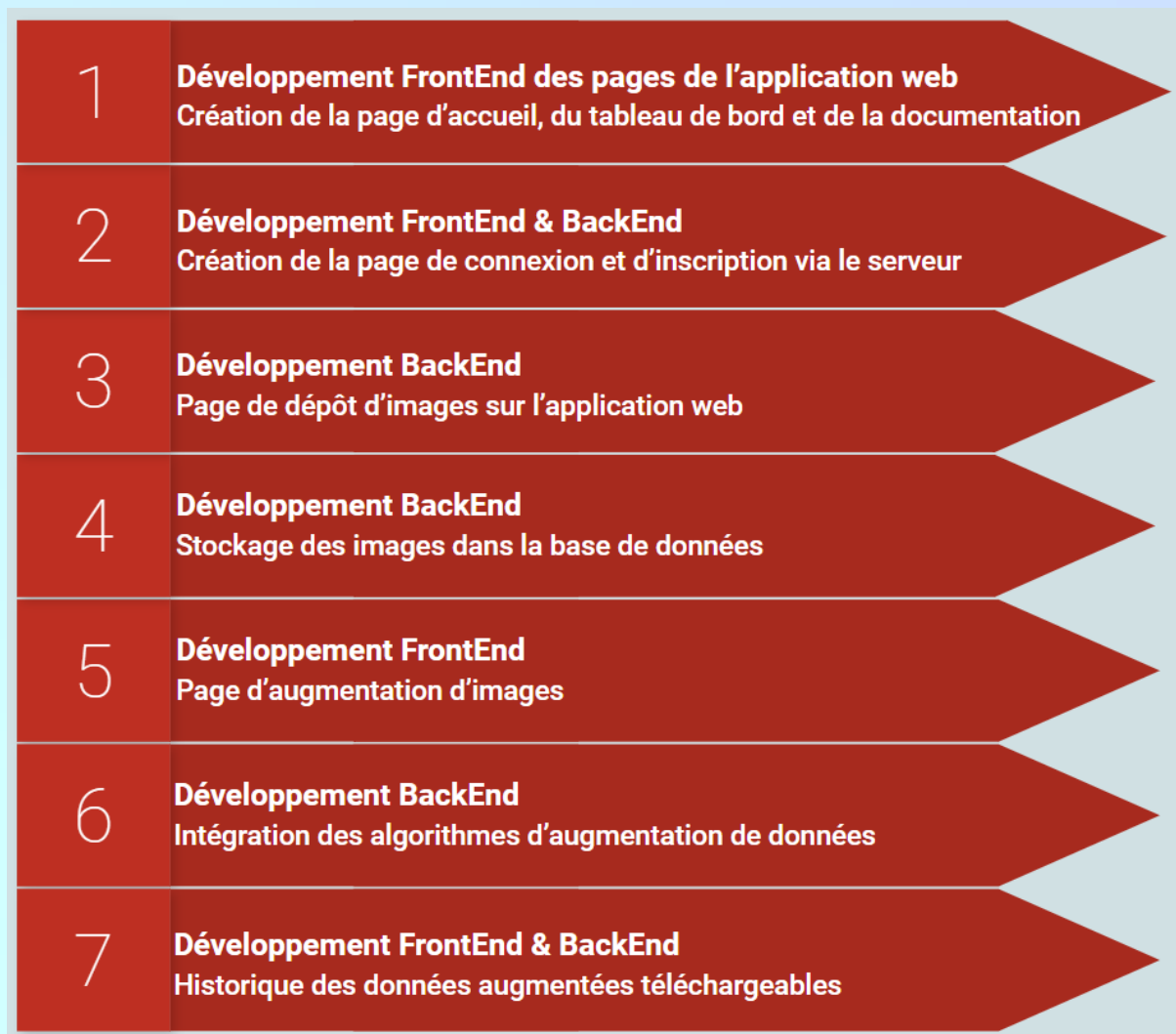
Dans un projet, le cycle de vie est un ensemble d'étapes de la vie du projet. En effet, nous nous sommes basés sur un le modèle représenté ci-dessous par le schéma :



La phase de contrôle permet de détecter si certaines décisions ou choix pris lors de la planification initiale n'étaient pas les bons et donc nécessitent d'être corrigés. D'où le processus de repasser par la planification, l'exécution et le contrôle.

## 9. Organisation

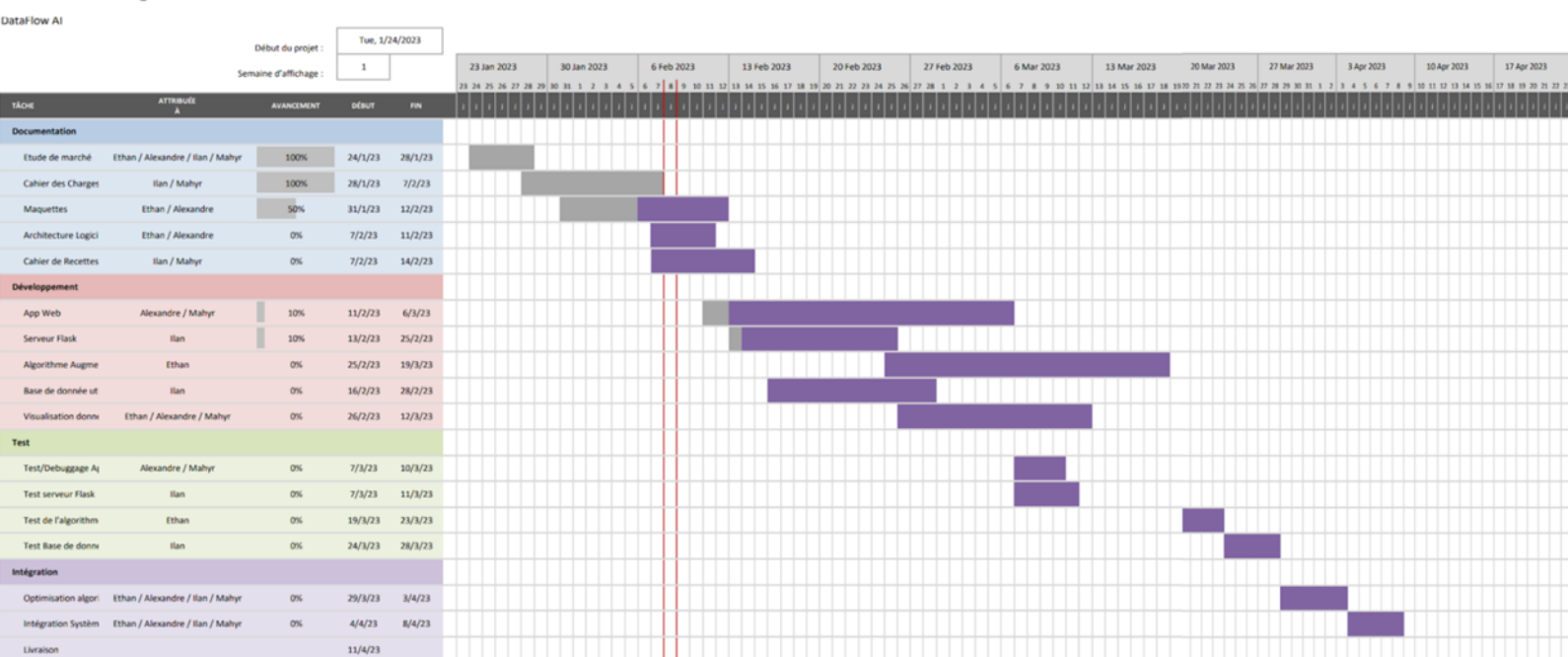
L'organisation du projet se décompose en plusieurs tâches :



## 10. Planification

Le projet est planifié sur une durée de 12 semaines, comprenant des réunions hebdomadaires ainsi que des soutenances et la rédaction d'un rapport final de projet. Les quatre premières semaines sont consacrées à la réalisation d'une étude de marché, la conception du cahier des charges, l'élaboration des recettes et des tests. Les six semaines suivantes sont dédiées à la phase de développement, tandis que les deux dernières semaines sont allouées à la rédaction du rapport de projet et à la préparation de la soutenance.

L3Q1 - Outil d'augmentation de la Donnée



A noter les dates importantes indiquées par M. Janiszek :

- Les *pré-soutenances* entre le 17/04/2023 et le 21/04/2023
- Les *soutenances* entre le 15/05/2023 et le 21/05/2023
- Le *dépôt du rapport* le 16/05/2023 à 18h sur la forge
- La **fermeture de la forge** le 16/05/2023 à 18h sur la forge

# 11. Glossaire

**Application web**: Programme ou logiciel qui est accessible depuis différents navigateurs web.

**Data Cleaning** : Ensemble de processus ayant pour but d'améliorer la qualité des données en corrigeant les données inexactes dans un ensemble de données.

**Dataset** : Ensemble de données cohérentes qui peuvent se présenter sous différents formats (textes, chiffres, images, audios, vidéos...).

**Données non structurées** : Données non organisées en base de données, c'est-à-dire la messagerie, les images, les vidéos, etc...

**Machine learning** : Le Machine Learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement **une sous-catégorie de l'intelligence artificielle**. Elle consiste à laisser des algorithmes découvrir des » patterns « , à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques...

**Deep learning** : Il s'agit d'un type de machine learning, version plus améliorée. L'apprentissage profond utilise une technique lui conférant une aptitude supérieure à détecter les patterns même les plus subtiles.

**Python** : Langage de programmation orienté objet, notamment très utile dans le domaine de l'intelligence artificielle.

**Flask** : Permet de créer des applications web en Python. Flask est un framework web léger qui offre une grande flexibilité et une facilité de développement pour la création d'applications web.

## 12. Sources

Introduction générale à la data augmentation :

<https://neovision.fr/data-augmentation-solutions-manque-donnees/>

[https://en.wikipedia.org/wiki/Data\\_augmentation](https://en.wikipedia.org/wiki/Data_augmentation)

<https://blog.datumize.com/the-five-most-common-data-quality-issues-and-how-to-overcome-them>

<https://www.cnil.fr/fr/definition/augmentation-de-donnees>

<https://lbourdois.github.io/blog/nlp/Data-augmentation-in-NLP/>

Présentation machine learning et deep learning :

<https://datascientest.com/machine-learning-tout-savoir>

[https://inside-machinelearning.com/pourquoi-et-comment-normaliser-ces-donnees-pytorch-une-etape-essentielle-du-deep-learning-partie-1/#Normaliser\\_les\\_donnees](https://inside-machinelearning.com/pourquoi-et-comment-normaliser-ces-donnees-pytorch-une-etape-essentielle-du-deep-learning-partie-1/#Normaliser_les_donnees)