|  |
| --- |
| Documentation technique |
| Green IA |
| 1. Architecture du système et présentation du projet |
| 1. Développement et technologies utilisées |
| 1. IA et Data 2. Interface utilisateur et déploiement 3. Sécurité, maintenance et support 4. Tests, validation et améliorations 5. Annexes |

**Sommaire :**

[I. Architecture du système et présentation du projet 4](#_Toc172544198)

[1. Présentation du projet 4](#_Toc172544199)

[2. Qu’est-ce que l’éco-score ? 4](#_Toc172544200)

[3. Vue d’ensemble 4](#_Toc172544201)

[4. Acteurs du système et interactions (utilisateurs, admins, autres) 5](#_Toc172544202)

[5. Diagramme de sequence UML 5](#_Toc172544203)

[6. Use case des fonctionnalités 5](#_Toc172544204)

[code unique + use case + entrées et sortie 5](#_Toc172544205)

[II. Développement et technologies utilisées 5](#_Toc172544206)

[7. Langages de Programmation et normes de codage 5](#_Toc172544207)

[8. Frameworks et bibliothèques 5](#_Toc172544208)

[9. Systèmes d’exploitation et environnement de travail 5](#_Toc172544209)

[10. Déploiement de nos solutions 5](#_Toc172544210)

[Stratégies de déploiement (blue-green, canary) 5](#_Toc172544211)

[Environnement de déploiement (dev, test, prod) 5](#_Toc172544212)

[Outils de déploiement (CI/CD pipelines, versions) 5](#_Toc172544213)

[Stratégie de scalabilité (verticale et horizontale) 5](#_Toc172544214)

[11. Contenu du projet, fichiers et exécution 5](#_Toc172544215)

[III. IA et Data 5](#_Toc172544216)

[12. Collecte des données 5](#_Toc172544217)

[13. Préparation des données 6](#_Toc172544218)

[14. Entraînement et validation des modèles 6](#_Toc172544219)

[15. Stockage et mise à disposition des données 7](#_Toc172544220)

[16. Dashboard utilisateur 7](#_Toc172544221)

[17. Dashboard global 7](#_Toc172544222)

[18. Pipeline 7](#_Toc172544223)

[IV. Interface utilisateur 7](#_Toc172544224)

[19. Sécurité et authentification 7](#_Toc172544225)

[20. Intégration avec des services externes 8](#_Toc172544226)

[21. Présentation des maquettes 8](#_Toc172544227)

[Page principale et page d’informations 8](#_Toc172544228)

[Dashboards utilisateur et global 9](#_Toc172544229)

[Informations dépôts et collectes 9](#_Toc172544230)

[22. Accessibilité 9](#_Toc172544231)

[23. Chemins utilisateur 10](#_Toc172544232)

[24. Diagrammes de flux 10](#_Toc172544233)

[V. Sécurité, maintenance et support 10](#_Toc172544234)

[25. Contraintes de sécurité 10](#_Toc172544235)

[26. Gestion des données sensibles 10](#_Toc172544236)

[27. Conformité et réglementations 10](#_Toc172544237)

[28. Plan de maintenance et MCO 10](#_Toc172544238)

[29. Surveillance et monitoring 10](#_Toc172544239)

[Rapport d’état de santé du système 10](#_Toc172544240)

[Rapport d’évolution des données 10](#_Toc172544241)

[Notifications aux administrateurs 10](#_Toc172544242)

[30. Plan des mises à jour et support 10](#_Toc172544243)

[VI. Tests, validation et améliorations 10](#_Toc172544244)

[31. Critères d’acceptation pour chaque fonctionnalité 10](#_Toc172544245)

[32. Stratégie de tests 12](#_Toc172544246)

[33. Tests unitaires 12](#_Toc172544247)

[34. Tests d'intégration 12](#_Toc172544248)

[35. Tests de performance 12](#_Toc172544249)

[36. Améliorations envisagées et versioning 12](#_Toc172544250)

[37. Confidentialité et accès 12](#_Toc172544251)

[VII. Annexes 12](#_Toc172544252)

[38. Glossaire 12](#_Toc172544253)

[39. Documents applicables 13](#_Toc172544254)

[40. Diffusion du document 13](#_Toc172544255)

[41. Historique des modifications 13](#_Toc172544256)

**Table des figures :**

[Figure 1, vue d'ensemble du projet 4](#_Toc172544257)

[Figure 2, pipeline représentation graphique 7](#_Toc172544258)

# Architecture du système et présentation du projet

## Présentation du projet

## Qu’est-ce que l’éco-score ?

## Vue d’ensemble

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

Description générée automatiquement

Figure 1, vue d'ensemble du projet

## Acteurs du système et interactions (utilisateurs, admins, autres)

## Diagramme de sequence UML

## Use case des fonctionnalités

### code unique + use case + entrées et sortie

# Développement et technologies utilisées

## Langages de Programmation et normes de codage

## Frameworks et bibliothèques

## Systèmes d’exploitation et environnement de travail

Pour ce projet nous utilisons trois systèmes d’exploitation différents, nous permettant de tester le fonctionnement de nos infrastructures sur différentes machines ; une Ubuntu 24, un MacOs Sonoma ainsi que deux Windows 11. L’autre atout de travailler sur différents OS, est la diminution du risque de mises à jour comportant des failles, ce qui rendrait la machine concernée inutilisable quelques temps.

Pour éviter les problèmes de déploiement et les conflits durant l’installation sur une nouvelle machine, nous avons exclusivement travaillé sur un environnement virtuel. Sa configuration aura été scrupuleusement mise à jour au fur et à mesure par chaque membre de l’équipe, grâce au fichier requirements.txt. Cet environnement virtuel est une reproduction à l’identique de notre environnement de déploiement.

## Déploiement de nos solutions

### Stratégies de déploiement (blue-green, canary)

### Environnement de déploiement (dev, test, prod)

### Outils de déploiement (CI/CD pipelines, versions)

### Stratégie de scalabilité (verticale et horizontale)

## Contenu du projet, fichiers et exécution

# IA et Data

## Collecte des données

Ici, trois sources de données ont été utilisées ; OpenFoodFacts pour la prédiction de l’éco score, le portail métropolitain ainsi que M data pour la gestion des points de collecte de déchets dans le département des Bouches du Rhône. Nous avons utilisé des techniques de scraping sur ces sites internet, notamment grâce à la librairie Beautiful Soup. Beautiful Soup permet de parser des documents HTML et XML, facilitant l'extraction de données à partir de pages web.

Les données pour la prédiction de l’éco score auront été plus complexes à traiter, en effet seulement deux fichiers compressés contiennent la totalité des articles d’Open Food Facts, un au format jsonl, l’autre au format Mongo DB Dump. Nous avons développé un script qui permet de télécharger le fichier compressé, avec la capacité de reprendre le téléchargement là où il s’était arrêté en cas de coupure réseau.

## Préparation des données

Une fois le fichier compressé téléchargé, notre script le décompressera en fichier jsonl, puis le parcourra en écrivant au fur et à mesure chaque ligne dans un fichier au format csv. Chacun de ces fichiers csv contiendra un maximum de 10 000 lignes, afin de les rendre exploitables sur des machines de particuliers. Par la suite, chaque csv sera ouvert et se verra amputer des colonnes jugées inutiles, il passera de 720 colonnes par ligne à une petite quinzaine.

Enfin, toutes les techniques de préparation classiques seront réalisées sur chacun des csv, avec une normalisation des valeurs numériques, une tokenisation du contenu textuel, ainsi que la fusion de plusieurs colonnes.

## Entraînement et validation des modèles

Régulièrement, un nouveau jeu de données provenant d’Open Food Facts sera téléchargé. Systématiquement un nouveau modèle sera entrainé sur ces nouvelles données, en parallèle du meilleur modèle que nous avons pu entrainer par le passé. Ainsi, un algorithme pourra comparer les résultats sur notre nouveau jeu de données de nos deux modèles, le meilleur sera mis de côté pour la prochaine mise à jour des données. Le csv avec le plus de bonnes réponses sera déployé.

Pour comparer les performances de nos réseaux de neurones, nous avons choisi les cinq métriques suivantes :

* L’erreur quadratique moyenne (MSE)
* L’erreur absolue moyenne (MAE) et médiane
* L’erreur quadratique moyenne racine (RMSE)
* Le coefficient de détermination (R2)

Ces métriques d’évaluation sont adaptées à notre volonté de prédire une variable continue à partir d’un ensemble de caractéristiques, c’est-à-dire une tache de régression.

## Stockage et mise à disposition des données

## Dashboard utilisateur

## Dashboard global

## Pipeline

Une image contenant diagramme, ligne, dessin, croquis

Description générée automatiquement

Figure 2, pipeline représentation graphique

# Interface utilisateur

## Sécurité et authentification

A ce stade du projet, aucune authentification n’est nécessaire pour utiliser l’application, il n’est d’ailleurs tout simplement pas possible de se créer un compte. Les données personnelles de l’utilisateur ainsi que se historique d’articles scannés sont sauvegardé dans des fichiers json en local dans son navigateur. Aucune donné ne nous est transmise, et stockée sur nos serveurs, le navigateur assurant de plus un niveau de sécurité que nous serions capable d’assurer sans un expert du domaine.

## Intégration avec des services externes

Ne cherchant pas à systématiquement réinventer la roue, de nombreuses solutions développées par d’autres entreprises ont été importés dans notre application web. Pour la carte nous avons fait le choix d’utiliser [OpenStreetMap](https://www.openstreetmap.fr/). Gratuite, Open Source et facilement implémentable, cette solution nous a semblé être la meilleur pour afficher des cartes à nos utilisateurs, bien loin des prix exorbitants de Google. Pour scanner les articles de l’utilisateur par le biais de sa caméra, nous avons utilisé [Quagga](https://cdnjs.cloudflare.com/ajax/libs/quagga/0.12.1/quagga.min.js) dans sa version 0.12.1, une libraire Open Source écrite en Java Script. Ces solutions ne sont utilisables que pour des applications déployées en https, le cas échant, ces librairies sont inaccessibles.

## Présentation des maquettes

### Une image contenant texte, capture d’écran, Police, document Description générée automatiquementUne image contenant texte, capture d’écran, Téléphone mobile, Appareil de communication Description générée automatiquementPage principale et page d’informations

Permet à l’utilisateur de scanner des articles alimentaires, télécharger sa liste de produits scannés et la vider ainsi que visualiser le nombre d’articles, hors doublons comptabilisés par l’application. La page d’information sert de manuel utilisateur ligne, tous les conseils d’utilisation et toutes les informations sur le traitement des données utilisateur y figures.

**Input :** Images de la caméra de l’utilisateur.

**Output :** Liste d’historique des produits téléchargés, informations sur l’impact environnemental du produit scanné.

### Une image contenant texte, capture d’écran, Téléphone mobile, gadget Description générée automatiquementUne image contenant texte, capture d’écran, Téléphone mobile, gadget Description générée automatiquementDashboards utilisateur et global

Un bouton permet à l’utilisateur de passer de son dashboard personnel à un dashboard global, ces dashboards ne sont pas hébergés sur la même machine que l’application web elle-même. Les graphiques s’adaptent à la taille de l’écran.

**Input :** Position du bouton.

**Output :** Affiche le dashboard associé à la position du bouton.

### Informations dépôts et collectes

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

Description générée automatiquementUne image contenant texte, capture d’écran, Police, Téléphone mobile

Description générée automatiquementLa page de gauche affiche sur une carte les heures de passage ainsi que les déchets traités, une fois le code postal de la ville où se trouve l’utilisateur entré. Celle de droite affiche directement l’endroit précis où se trouve l’utilisateur, affichant les lieux recyclant les types de déchets sélectionnés par l’utilisateur.

**Input :** Code postal de l’utilisateur, activation du bouton de soumission, chaque type de déchet pris en charge par l’application.

**Output :** Affiche les informations sur les points de dépôts et les horaires de collecte des déchets.

## Accessibilité

Ce choix des technologies web, nous permet d’éviter le développement d’une application Android et IOS, environnements avec lesquels nous ne sommes absolument pas familiers, impliquant de nombreuses heures de formations et de développement, pour un résultat surement médiocre. Avec le web nous avons pu développer une application responsive, aussi bien disponible sur ordinateur, que sur mobile. Il suffit de cliquer se rendre sur le lien suivant : [Green Ia](https://caroluscharlemagne.github.io/Green_IA_website/). Il n’est pas non plus nécessaire de créer un compte pour utiliser notre application.

L’unique prérequis pour accéder à cette application web est un accès un internet (wifi, 4G, 5G, etc) ainsi qu’un navigateur web. Attention cependant à certains navigateurs trop restrictifs qui pourraient bloquer l’accès à la caméra où aux notifications.

## Chemins utilisateur

## Diagrammes de flux

# Sécurité, maintenance et support

## Contraintes de sécurité

## Gestion des données sensibles

Nous ne traitons pas de données sensibles dans ce projet, aucune donnée personnelle ou médicale. A terme, une base de données sera créée et hébergée chez AWS ou Google, la sécurité des données leur sera ainsi confiée. Nous ferons en revanche auditer par un expert de la sécurité, la robustesse de nos sites web et applications qui afficheront et réaccueillerons ces données, notamment des données utilisateur.

## Conformité et réglementations

## Plan de maintenance et MCO

## Surveillance et monitoring

### Rapport d’état de santé du système

### Rapport d’évolution des données

### Notifications aux administrateurs

## Plan des mises à jour et support

# Tests, validation et améliorations

## Critères d’acceptation pour chaque fonctionnalité

C : conforme, EC : en cours, NC : non conforme.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifiant fonctionnalité** | **Résultat attendu** | **Constatations** | **Validation** |
| **Site web** | | | |
| 01102023SW05 | Le menu tous les boutons nécessaires, tous cliquables et redirigeant vers la bonne page | Ok | C |
| 01102023SW06\_00 | Possibilité de scanner en boucle le code barre de produits alimentaires. Il faut afficher un message d’erreur dans le cas où le produit est inconnu. Pour ne pas empiéter sur la vie privée de nos utilisateurs, il sera également obligatoire de valider l’utilisation de sa caméra à chaque ouverture d’application | ~~Impossible de scanner plusieurs articles sans fermer et ouvrir l’application de nouveau~~ | C |
| 01102023SW06\_01 | Le bouton doit permettre le téléchargement de la liste d’historique des produits scannés par l’utilisateur | Ok | C |
| 01102023SW06\_02 | Bouton permettant à l’utilisateur de vider son historique | Ok | C |
| 01102023SW07\_00 | L’utilisateur doit avoir un champ pour entrer son adresse postale, ainsi qu’un bouton | Ok | C |
| 01102023SW07\_01 | Le code postal renseigné doit permettre d’afficher la position de la ville sur la carte, avec les informations de collecte des déchets associés, stockés dans le fichier json | Ok | C |
| 01102023SW08\_00 | L’utilisateur doit pouvoir filtrer les types de déchets qu’il souhaite jeter | Ok | C |
| 01102023SW08\_01 | Seuls les points de dépôt appartenant à une catégorie choisie par l’utilisateur et autour de lui, doivent apparaitre sur une carte. De plus, au clic sur l’un de ses points, toutes les informations sur le lieu doivent apparaitre (horaires, types, jours d’ouverture) | ~~Différencier la couleur du pin de l’utilisateur de celui des boutiques~~ | C |
| 01102023SW11\_00 | Cette page doit contenir toutes les informations utiles à l’utilisateur pour comprendre le projet, ses objectifs et répondre à toutes ces questions courantes | Ok | C |
| 01102023SW12 | Aucun code inutile ne doit être présent, il doit être compréhensible par tous, respecter la PEP-8 | Aucun soucis | C |
| 01102023SW13 | Le site web doit être déployer en https pour faire fonctionner tous les composants externes, disponible depuis n’importe quel navigateur | Ok | C |
| **Dashboard** | | | |
| 01102023DBrd02 |  |  |  |
| 01102023DBrd03 |  |  |  |
| 01102023DBrd04 |  |  |  |
| 01102023DBrd05 |  |  |  |
| 01102023DBrd06 |  |  |  |
| **Prédiction éco score** | | | |
| 01102023PES01 | Ne produisant pas de données et n’ayant pas la possibilité de payer, nous devons nous tourner vers une ou plusieurs sources gratuites | Ok | C |
| 01102023PES07 | Étant donné la complexité de prédire l’éco score sur des produits alimentaires, un objectif de 80% de positifs sur des données de validation serait un bon résultat |  | EC |

## Stratégie de tests

## Tests unitaires

## Tests d'intégration

## Tests de performance

## Améliorations envisagées et versioning

## Confidentialité et accès

Dans notre groupe de travail restreint, chacun d’entre nous a les mêmes responsabilités. La perte de fichiers, la fuite de données ou une éventuelle régression des performances sera directement imputée au collaborateur n’ayant pas validé lui-même, puis fait valider son travail par un pair avant de le mettre en production.

# Annexes

## Glossaire

|  |  |
| --- | --- |
| Abréviation | Signification |
| IA et Data Science | |
| IA | Intelligence Artificielle |
| Développement | |
| Framework | Environnement de travail facilitant le développement d’une solution technique. |
| Responsivité | Possibilité d’adapter la taille du logiciel à la taille de l’écran de l’utilisateur |
| Front | Développement des aspects visuels du logiciel |
| Back | Développement de la partie logique du logiciel (caché à l’utilisateur) |
| API REST | API Representational State Transfer Application Program Interface est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil. Le plus souvent les développeurs utilisent des API REST pour créer des services web. Souvent appelés services web RESTful, REST utilise des méthodes HTTP pour récupérer et publier des données entre un périphérique client et un serveur. |

## Documents applicables

|  |  |
| --- | --- |
| Description | Identification |
| 23-24 Modalités Évaluations Titre EISI N7 Étudiants – AYC Pour les M2 | REF [0] |
| Dépôt Moodle filière informatique M2 | REF [1] |

## Diffusion du document

|  |  |  |  |
| --- | --- | --- | --- |
| Diffusion | Statut | Nom | Emis le |
| Edition | Charlemagne | 05/07/2024 |
|  |  |  |
|  |  |  |

## Historique des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Pages | Description de la modification - Auteur | Date |
| 0.3 |  | Architecture globale du document | 05/07/2024 |
|  |  |  |  |
|  |  |  |  |