|  |
| --- |
| 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](#_Toc172408633)

[1. Présentation du projet 4](#_Toc172408634)

[2. Qu’est-ce que l’éco-score ? 4](#_Toc172408635)

[3. Vue d’ensemble 4](#_Toc172408636)

[4. Acteurs du système (utilisateurs, admins, autres) 5](#_Toc172408637)

[5. Interactions entre les acteurs et le système 5](#_Toc172408638)

[6. Diagrammes UML (Sequence, Class) 5](#_Toc172408639)

[7. Use case des fonctionnalités 5](#_Toc172408640)

[Fonctionnalité 1 (code unique + use case + entrées et sorties) 5](#_Toc172408641)

[II. Développement et technologies utilisées 5](#_Toc172408642)

[8. Langages de Programmation 5](#_Toc172408643)

[9. Frameworks et bibliothèques 5](#_Toc172408644)

[10. Outils et environnements de développement 5](#_Toc172408645)

[11. Base de données (schémas, relations) 5](#_Toc172408646)

[12. Systèmes d’exploitation 5](#_Toc172408647)

[III. IA et Data 5](#_Toc172408648)

[13. Collecte des données 5](#_Toc172408649)

[14. Préparation des données 5](#_Toc172408650)

[15. Prédiction des ingrédients 5](#_Toc172408651)

[16. Prédiction des mots clés des produits 5](#_Toc172408652)

[17. Prédiction de l’éco-score 5](#_Toc172408653)

[18. Entraînement des modèles 5](#_Toc172408654)

[19. Évaluation et validation des modèle 5](#_Toc172408655)

[20. Dashboard utilisateur 5](#_Toc172408656)

[21. Dashboard global 6](#_Toc172408657)

[22. Déploiement de nos solutions 6](#_Toc172408658)

[Stratégies de déploiement (blue-green, canary) 6](#_Toc172408659)

[Environnement de déploiement (dev, test, prod) 6](#_Toc172408660)

[Outils de déploiement (CI/CD pipelines, versions) 6](#_Toc172408661)

[23. Pipeline 6](#_Toc172408662)

[IV. Interface utilisateur et déploiement 6](#_Toc172408663)

[24. Sécurité et authentification 6](#_Toc172408664)

[25. Intégration avec des services externes 7](#_Toc172408665)

[26. Présentation des maquettes 7](#_Toc172408666)

[Page principale et page d’informations 7](#_Toc172408667)

[Dashboards utilisateur et global 8](#_Toc172408668)

[Informations dépôts et collectes 8](#_Toc172408669)

[27. Accessibilité 8](#_Toc172408670)

[28. Environnement de déploiement 9](#_Toc172408671)

[29. Processus de déploiement 9](#_Toc172408672)

[30. Outils de déploiement (CI/CD) 9](#_Toc172408673)

[31. Stratégie de scalabilité (horizontale et verticale) 9](#_Toc172408674)

[V. Sécurité, maintenance et support 9](#_Toc172408675)

[32. Contraintes de sécurité 9](#_Toc172408676)

[33. Gestion des données sensibles 9](#_Toc172408677)

[34. Conformité et réglementations 9](#_Toc172408678)

[35. Plan de maintenance 9](#_Toc172408679)

[36. Surveillance et monitoring 9](#_Toc172408680)

[Rapport d’état de santé du système 9](#_Toc172408681)

[Rapport d’évolution des données 9](#_Toc172408682)

[Notifications aux administrateurs 9](#_Toc172408683)

[37. Plan des mises à jour 9](#_Toc172408684)

[38. Support utilisateur 9](#_Toc172408685)

[VI. Tests, validation et améliorations 9](#_Toc172408686)

[39. Critères d’acceptation pour chaque fonctionnalité 9](#_Toc172408687)

[40. Stratégie de tests 9](#_Toc172408688)

[41. Tests unitaires 9](#_Toc172408689)

[42. Tests d'intégration 9](#_Toc172408690)

[43. Tests de performance 9](#_Toc172408691)

[44. Améliorations envisagées 9](#_Toc172408692)

[VII. Annexes 10](#_Toc172408693)

[45. Glossaire 10](#_Toc172408694)

[46. Documents applicables 10](#_Toc172408695)

[47. Diffusion du document 10](#_Toc172408696)

[48. Historique des modifications 10](#_Toc172408697)

**Table des figures :**

[Figure 1, vue d'ensemble du projet 4](#_Toc172408698)

[Figure 2, pipeline représentation graphique 6](#_Toc172408699)

# 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 (utilisateurs, admins, autres)

## Interactions entre les acteurs et le système

## Diagrammes UML (Sequence, Class)

## Use case des fonctionnalités

### Fonctionnalité 1 (code unique + use case + entrées et sorties)

# Développement et technologies utilisées

## Langages de Programmation

## Frameworks et bibliothèques

## Outils et environnements de développement

## Base de données (schémas, relations)

## Systèmes d’exploitation

# IA et Data

## Collecte des données

## Préparation des données

## Prédiction des ingrédients

## Prédiction des mots clés des produits

## Prédiction de l’éco-score

## Entraînement des modèles

## Évaluation et validation des modèle

## Dashboard utilisateur

## Dashboard global

## Déploiement de nos solutions

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

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

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

## Pipeline

Une image contenant diagramme, ligne, Plan, Dessin technique

Description générée automatiquement

Figure 2, pipeline représentation graphique

# Interface utilisateur et déploiement

## 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

### Page principale et page d’informations

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 automatiquementPermet 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.

### Dashboards utilisateur et global

Une image contenant texte, capture d’écran, Téléphone mobile, gadget

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

Description générée automatiquementUn 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.

### 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.

## 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.

## Environnement de déploiement

## Processus de déploiement

## Outils de déploiement (CI/CD)

## Stratégie de scalabilité (horizontale et verticale)

# Sécurité, maintenance et support

## Contraintes de sécurité

## Gestion des données sensibles

## Conformité et réglementations

## Plan de maintenance

## Surveillance et monitoring

### Rapport d’état de santé du système

### Rapport d’évolution des données

### Notifications aux administrateurs

## Plan des mises à jour

## Support utilisateur

# Tests, validation et améliorations

## Critères d’acceptation pour chaque fonctionnalité

## Stratégie de tests

## Tests unitaires

## Tests d'intégration

## Tests de performance

## Améliorations envisagées

# 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 |
|  |  |  |  |
|  |  |  |  |