



Documentation technique

Green IA

- 1) Architecture du système et présentation du projet
- 2) Développement et technologies utilisées
- 3) IA et Data
- 4) Interface utilisateur et déploiement
- 5) Sécurité, maintenance et support
- 6) Tests, validation et améliorations
- 7) Annexes

Sommaire :

I. ARCHITECTURE DU SYSTEME ET PRESENTATION DU PROJET	4
1. PRESENTATION DU PROJET	4
2. QU'EST-CE QUE L'ECO-SCORE ?	4
3. VUE D'ENSEMBLE.....	4
4. ACTEURS DU SYSTEME (UTILISATEURS, ADMINS, AUTRES)	5
5. INTERACTIONS ENTRE LES ACTEURS ET LE SYSTEME.....	5
6. DIAGRAMMES UML (SEQUENCE, CLASS)	5
7. USE CASE DES FONCTIONNALITES.....	5
<i>Fonctionnalité 1 (code unique + use case + entrées et sorties)</i>	<i>5</i>
II. DEVELOPPEMENT ET TECHNOLOGIES UTILISEES	5
8. LANGAGES DE PROGRAMMATION.....	5
9. FRAMEWORKS ET BIBLIOTHEQUES.....	5
10. OUTILS ET ENVIRONNEMENTS DE DEVELOPPEMENT	5
11. BASE DE DONNEES (SCHEMAS, RELATIONS)	5
12. SYSTEMES D'EXPLOITATION	5
III. IA ET DATA	5
13. COLLECTE DES DONNEES.....	5
14. PREPARATION DES DONNEES.....	5
15. PREDICTION DES INGREDIENTS	5
16. PREDICTION DES MOTS CLES DES PRODUITS.....	5
17. PREDICTION DE L'ECO-SCORE.....	5
18. ENTRAINEMENT DES MODELES.....	5
19. ÉVALUATION ET VALIDATION DES MODELE	5
20. DASHBOARD UTILISATEUR	5
21. DASHBOARD GLOBAL.....	6
22. DEPLOIEMENT DE NOS SOLUTIONS.....	6
<i>Stratégies de déploiement (blue-green, canary)</i>	<i>6</i>
<i>Environnement de déploiement (dev, test, prod).....</i>	<i>6</i>
<i>Outils de déploiement (CI/CD pipelines, versions).....</i>	<i>6</i>
23. PIPELINE.....	6
IV. INTERFACE UTILISATEUR ET DEPLOIEMENT	6
24. SECURITE ET AUTHENTIFICATION	6
25. INTEGRATION AVEC DES SERVICES EXTERNES.....	7
26. PRESENTATION DES MAQUETTES.....	7
<i>Page principale et page d'informations</i>	<i>7</i>
<i>Dashboards utilisateur et global</i>	<i>8</i>
<i>Informations dépôts et collectes</i>	<i>8</i>
27. ACCESSIBILITE.....	8
28. ENVIRONNEMENT DE DEPLOIEMENT	9
29. PROCESSUS DE DEPLOIEMENT	9
30. OUTILS DE DEPLOIEMENT (CI/CD).....	9
31. STRATEGIE DE SCALABILITE (HORIZONTALE ET VERTICALE)	9
V. SECURITE, MAINTENANCE ET SUPPORT	9

32.	CONTRAINTES DE SECURITE	9
33.	GESTION DES DONNEES SENSIBLES.....	9
34.	CONFORMITE ET REGLEMENTATIONS.....	9
35.	PLAN DE MAINTENANCE.....	9
36.	SURVEILLANCE ET MONITORING	9
	<i>Rapport d'état de santé du système</i>	<i>9</i>
	<i>Rapport d'évolution des données.....</i>	<i>9</i>
	<i>Notifications aux administrateurs.....</i>	<i>9</i>
37.	PLAN DES MISES A JOUR.....	9
38.	SUPPORT UTILISATEUR	9
VI. TESTS, VALIDATION ET AMELIORATIONS		9
39.	CRITERES D'ACCEPTATION POUR CHAQUE FONCTIONNALITE	9
40.	STRATEGIE DE TESTS.....	9
41.	TESTS UNITAIRES.....	9
42.	TESTS D'INTEGRATION.....	9
43.	TESTS DE PERFORMANCE	9
44.	AMELIORATIONS ENVISAGEES	9
VII. ANNEXES.....		10
45.	GLOSSAIRE.....	10
46.	DOCUMENTS APPLICABLES.....	10
47.	DIFFUSION DU DOCUMENT	10
48.	HISTORIQUE DES MODIFICATIONS	10

Table des figures :

Figure 1, vue d'ensemble du projet	4
Figure 2, pipeline représentation graphique	6

I. Architecture du système et présentation du projet

1. Présentation du projet
2. Qu'est-ce que l'éco-score ?
3. Vue d'ensemble

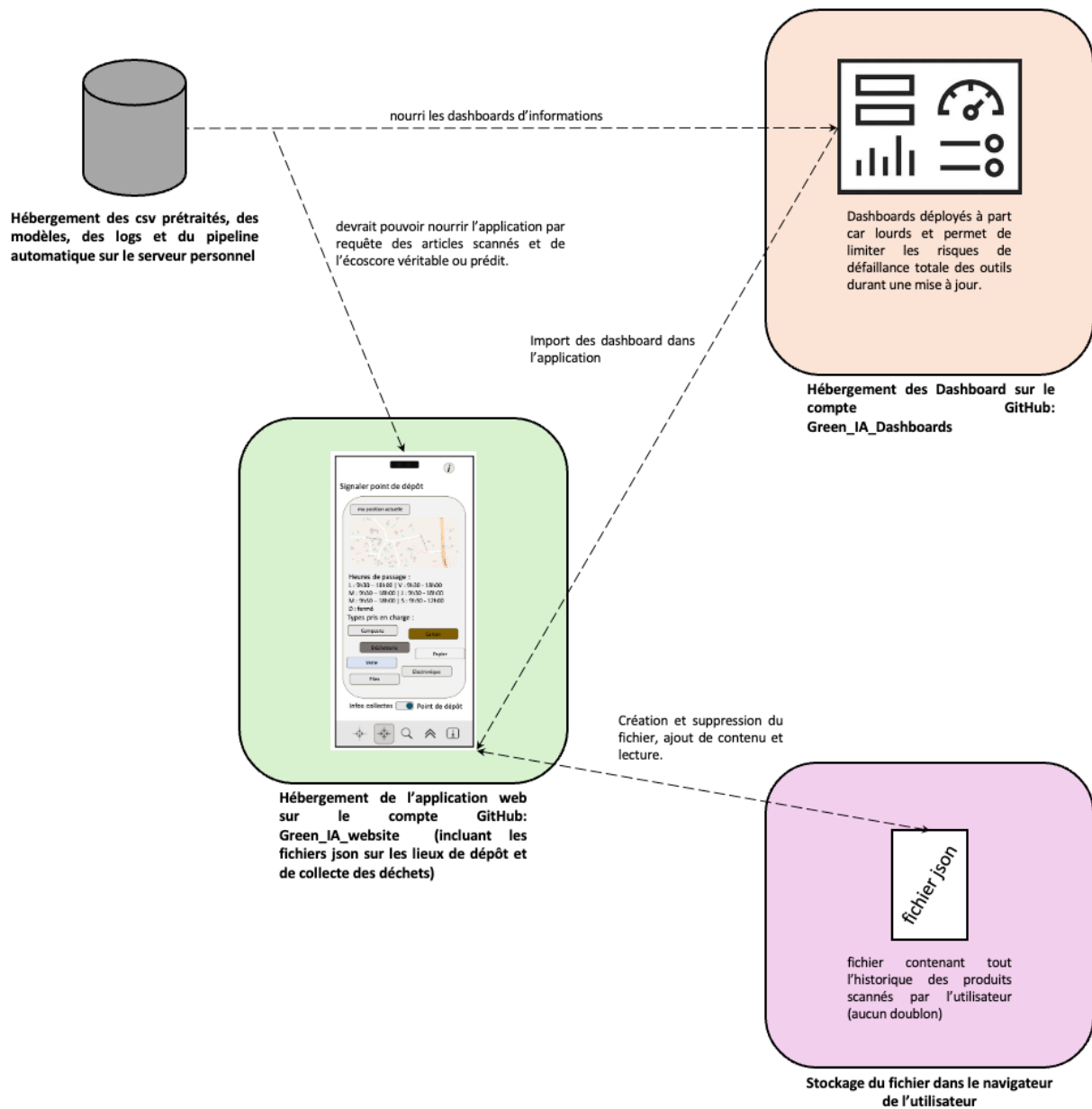


Figure 1, vue d'ensemble du projet

4. Acteurs du système (utilisateurs, admins, autres)
 5. Interactions entre les acteurs et le système
 6. Diagrammes UML (Sequence, Class)
 7. Use case des fonctionnalités
- Fonctionnalité 1 (code unique + use case + entrées et sorties)

II. Développement et technologies utilisées

8. Langages de Programmation
9. Frameworks et bibliothèques
10. Outils et environnements de développement
11. Base de données (schémas, relations)
12. Systèmes d'exploitation

III. IA et Data

13. Collecte des données
14. Préparation des données
15. Prédiction des ingrédients
16. Prédiction des mots clés des produits
17. Prédiction de l'éco-score
18. Entraînement des modèles
19. Évaluation et validation des modèle
20. Dashboard utilisateur

21. Dashboard global

22. Déploiement de nos solutions

Stratégies de déploiement (blue-green, canary)

Environnement de déploiement (dev, test, prod)

Outils de déploiement (CI/CD pipelines, versions)

23. Pipeline

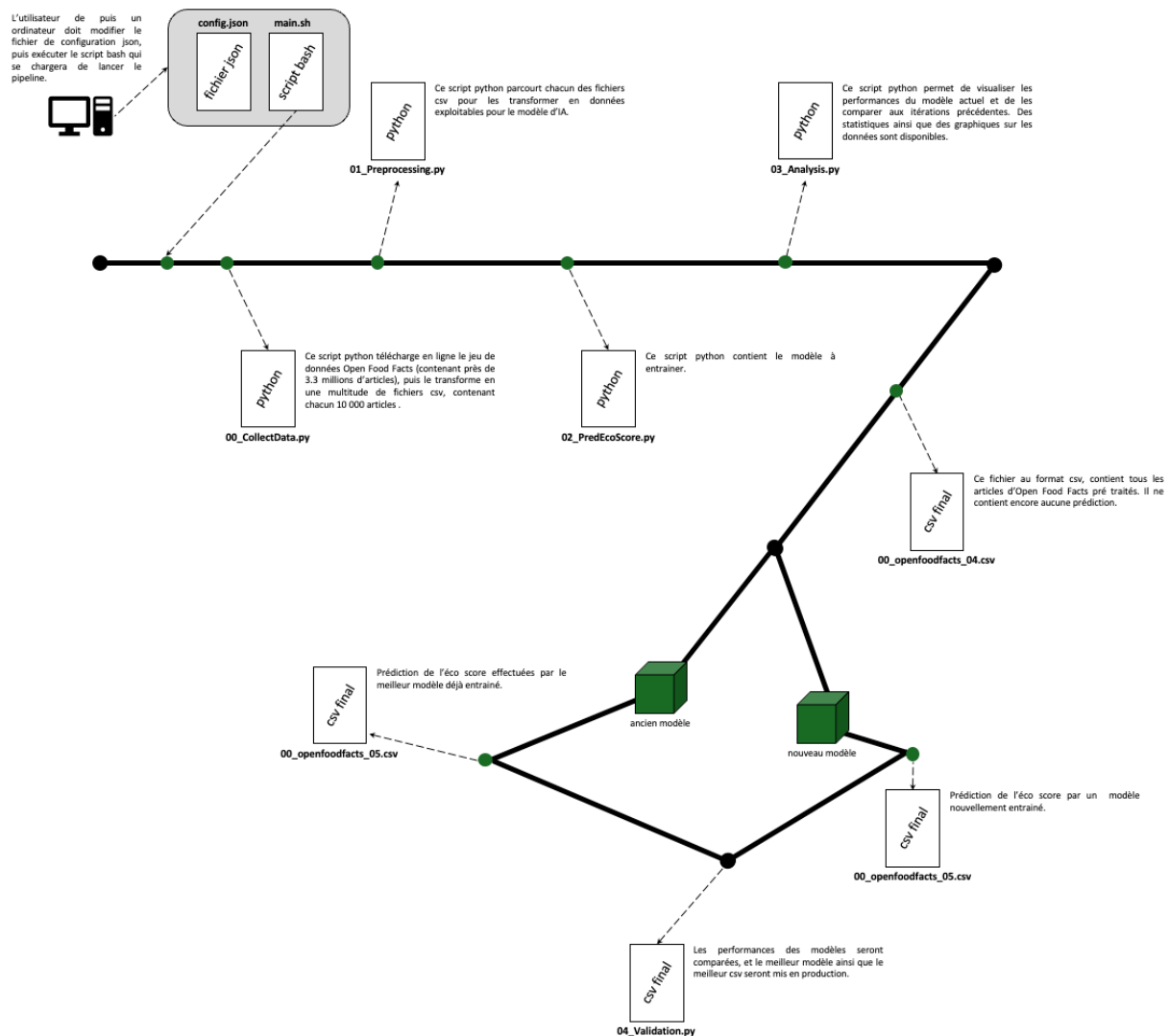


Figure 2, pipeline représentation graphique

IV. Interface utilisateur et déploiement

24. 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 donnée 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.

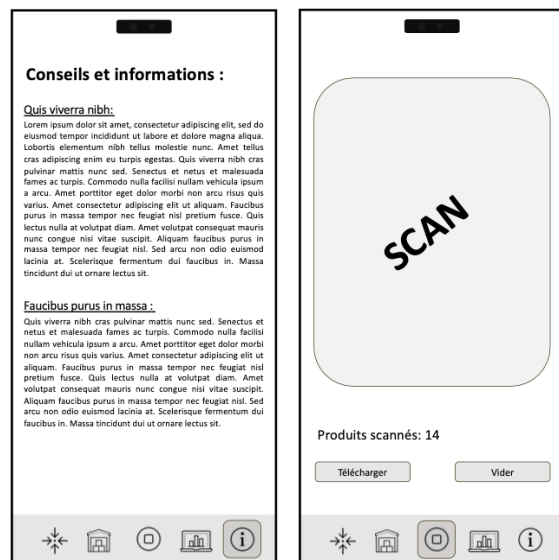
25. Intégration avec des services externes

Ne cherchant pas à systématiquement réinventer la roue, de nombreuses solutions développées par d'autres entreprises ont été importées dans notre application web. Pour la carte nous avons fait le choix d'utiliser [OpenStreetMap](#). Gratuite, Open Source et facilement implémentable, cette solution nous a semblé être la meilleure 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](#) dans sa version 0.12.1, une librairie Open Source écrite en Java Script. Ces solutions ne sont utilisables que pour des applications déployées en https, le cas échéant, ces librairies sont inaccessibles.

26. Présentation des maquettes

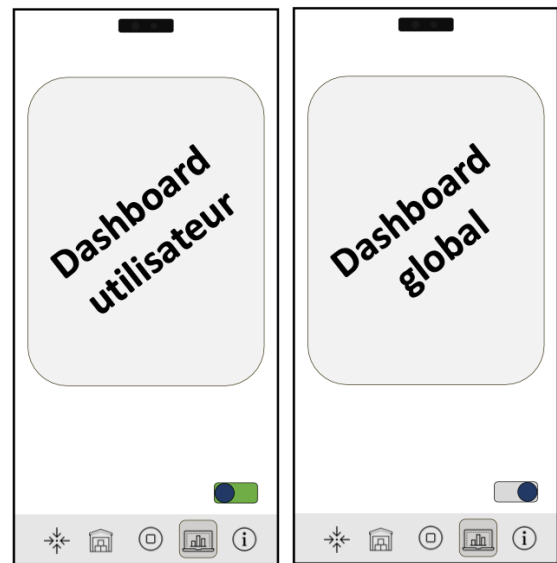
Page principale et page d'informations

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 figurent.



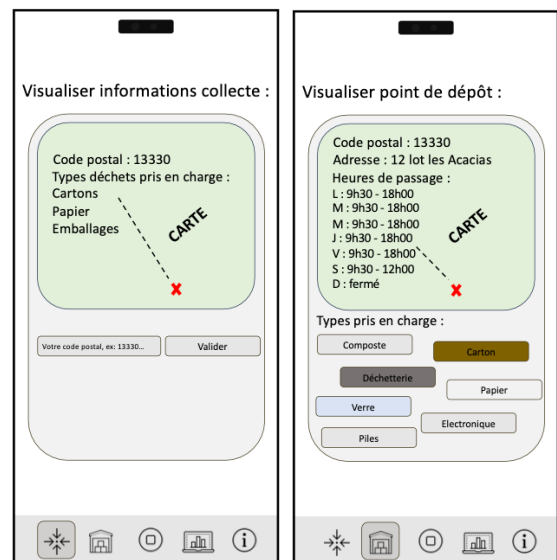
Dashboards utilisateur et global

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.



Informations dépôts et collectes

La 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.



27. 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://green-ia.com). Il n'est pas non plus nécessaire de créer un compte pour utiliser notre application.

28. Environnement de déploiement

29. Processus de déploiement

30. Outils de déploiement (CI/CD)

31. Stratégie de scalabilité (horizontale et verticale)

V. Sécurité, maintenance et support

32. Contraintes de sécurité

33. Gestion des données sensibles

34. Conformité et réglementations

35. Plan de maintenance

36. Surveillance et monitoring

Rapport d'état de santé du système

Rapport d'évolution des données

Notifications aux administrateurs

37. Plan des mises à jour

38. Support utilisateur

VI. Tests, validation et améliorations

39. Critères d'acceptation pour chaque fonctionnalité

40. Stratégie de tests

41. Tests unitaires

42. Tests d'intégration

43. Tests de performance

44. Améliorations envisagées

VII. Annexes

45. 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.

46. 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]

47. Diffusion du document

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

48. Historique des modifications

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

