

EisenFlow : Gestion Intelligente d'Emails

Projet DevOps 1 - Documentation

Alex BRINDUSOIU

Arthur CHAUVEAU

Swetha SARAVANAN

Franck ZHENG

Février 2026

Table des matières

1	Présentation Générale	2
1.1	1. Qui ?	2
1.2	2. Quoi ? Pourquoi ?	2
1.3	3. Description de l'application	2
1.3.1	Fonctionnalités clés	2
1.4	4. Analyse de Concurrence	2
1.5	5. Éléments de gestion de projet	2
1.6	6. Diagramme de classes	3
1.6.1	6.1 But des features (Release 1)	3
1.6.2	6.2 Scénarios Utilisateur	3
1.6.3	6.3 Wireframe et screenshots	3
1.7	7. Résumé des features	4
1.8	8. Annexe API REST	4

1 Présentation Générale

1.1 1. Qui ?

EisenFlow a été créé pour aider les personnes qui reçoivent beaucoup d'emails et qui ont du mal à tout gérer. Le but est de réduire le stress et de mieux organiser le travail grâce à l'Intelligence Artificielle.

1.2 2. Quoi ? Pourquoi ?

L'objectif est d'appliquer la méthode de productivité *Eisenhower* directement dans le flux de travail des emails, en utilisant une IA locale pour garantir la **confidentialité** des données utilisateur.

1.3 3. Description de l'application

Notre application permet à l'utilisateur de charger ses mails dans notre application. Cette application va ensuite donner la possibilité de soit classer automatiquement les mails dans la Matrice Eisenhower pour que les mails se fassent taguer, soit de laisser classer et taguer les mails manuellement par l'utilisateur. Parmi les fonctionnalités présentes dans notre application il y a la possibilité d'envoyer les résultats dans Gmail directement, classer et taguer les mails. Dans les tags disponibles, il y a le DO, PLAN, DELEGATE et DELETE. Le DO est le tag qui regroupe les tâches urgentes et importantes nécessitant une action immédiate.

- **Classement automatique :** L'IA analyse le contenu et applique les tags de la **Matrice Eisenhower**.
- **Classement manuel :** L'utilisateur garde le contrôle total sur le tagage.

1.3.1 Fonctionnalités clés

L'application permet d'envoyer les résultats directement dans Gmail. Les tags disponibles sont : 1. **DO** : Tâches urgentes et importantes. 2. **PLAN** : Tâches importantes mais non urgentes. 3. **DELEGATE** : Tâches urgentes mais non importantes. 4. **DELETE** : Tâches ni urgentes ni importantes.

1.4 4. Analyse de Concurrence

Parmi les concurrents de notre application, on retrouve plusieurs applications largement utilisées qui proposent des fonctionnalités similaires. Par exemple, les boîtes mails classiques comme Gmail, Outlook ou AppleMail, qui permettent la gestion centralisée des mails, d'afficher des messages, de créer des dossiers ou encore de créer des règles manuelles. Contrairement à notre application, Gmail vise à faciliter la communication et l'organisation de l'information pour les utilisateurs mais elle ne permet pas le tri de mails automatique (pour l'instant).

Un autre type d'applications parmi les boîtes mails sont celles utilisant l'IA comme SuperHuman. L'IA peut résumer un mail et écrire une réponse. SuperHuman a donc quelques ressemblances par rapport à notre application (utilisation de l'IA ainsi que réponse qui ressemble à notre DELEGATE automatique) mais contrairement à nous, SuperHuman n'utilise pas une IA locale mais envoie les mails sur des serveurs en ligne. Si l'utilisateur souhaite de la confidentialité, ça serait une problématique.

Il y a aussi les applications de gestion de projet comme Jira qui proposent des tableaux Kanban, permettent de suivre les tâches et sont donc très utiles pour le travail en équipe. Notre application permet de suivre les tâches lues dans les mails, contrairement à Jira qui nécessite de mettre manuellement chaque tâche.

Note sur la confidentialité : Contrairement à SuperHuman, EisenFlow privilégie une **IA locale** pour protéger la vie privée des utilisateurs.

1.5 5. Éléments de gestion de projet

Dans le cadre de ce projet DevOps, nous avons structuré notre cycle de développement autour des pratiques agiles et des outils suivants : * **Organisation & Suivi** : Méthodologie Agile avec suivi des tickets via un tableau Kanban. * **Versionning & CI/CD** : Git avec hébergement sur GitHub. Mise en place de pipelines

GitHub Actions pour l'intégration continue et la génération automatique de la documentation technique (via Pandoc/LaTeX). * **Architecture** : Conteneurisation de l'application via **Docker**.

1.6 6. Diagramme de classes

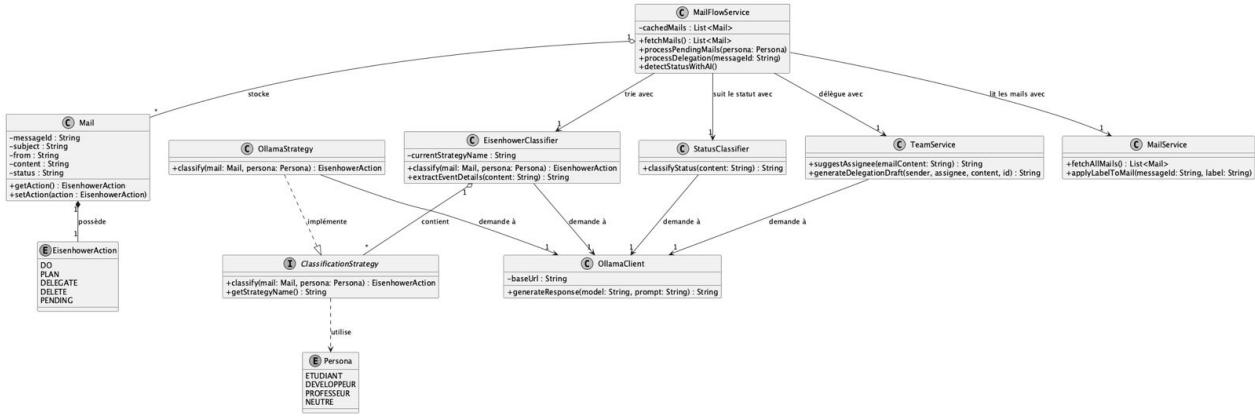


Figure 1: Diagramme de classes

1.6.1 6.1 But des features (Release 1)

Pour notre première version (MVP), nous nous concentrerons sur 6 petites features à forte valeur ajoutée :

- F1 : Interface de lecture unifiée** Interface permettant de lire les mails et les notes au même endroit, avec des sélecteurs de priorité manuels (DO, PLAN, DELEGATE, DELETE).
- F2 : Bascule de Persona IA** Système permettant de changer de profil (Étudiant, Développeur, Professeur). Cela modifie les prompts et le comportement de l'IA locale lors de l'analyse automatique.
- F3 : Actions rapides Gmail** Boutons d'action intégrés sur le Kanban pour “Archiver”, “Supprimer” ou “Marquer comme lu”, synchronisés en direct avec la boîte Gmail.
- F4 : Indicateurs visuels d'échéance** Affichage de badges de couleur (vert, orange, rouge) sur les cartes du Kanban pour identifier rapidement l'urgence des tâches.
- F5 : Recherche et filtrage en temps réel** Barre de recherche pour filtrer instantanément les cartes du Kanban par expéditeur, mots-clés ou par date.
- F6 : Export local sécurisé** Possibilité d'exporter l'état actuel de son tableau (tâches et notes) au format CSV ou Markdown pour garder une trace locale sans passer par le cloud.

1.6.2 6.2 Scénarios Utilisateur

- Scénario 1 (Persona & Tri Automatique)** : L'utilisateur sélectionne le persona “Étudiant”. Lorsqu'il synchronise sa boîte mail, l'IA locale comprend que les messages de l'administration scolaire sont urgents et les tag automatiquement en “DO”, tandis que les offres d'emploi étudiant vont dans “PLAN”.
- Scénario 2 (Interface unifiée & Actions rapides)** : L'utilisateur lit un mail depuis l'interface unifiée. Il décide que c'est une tâche inutile, clique sur le sélecteur “DELETE”, puis utilise le bouton d'action rapide pour l'archiver directement sur son vrai compte Gmail.
- Scénario 3 (Recherche & Export)** : En fin de semaine, l'utilisateur utilise la barre de recherche pour filtrer toutes les tâches liées au mot “Projet”. Il clique ensuite sur le bouton d'export pour récupérer ces informations dans un fichier Markdown local.

1.6.3 6.3 Wireframe et screenshots

(À compléter)

1.7 7. Résumé des features

L'application centralise et automatise la gestion des tâches via 6 fonctionnalités clés de la Release 1 :

- **Interface unifiée** : Mails et notes regroupés avec assignation manuelle ou automatique.
- **Persona IA** : Adaptation du tri automatique au profil de l'utilisateur (Étudiant, Dev, etc.).
- **Actions directes** : Contrôle de la boîte Gmail (Archiver/Supprimer) depuis le Kanban.
- **Badges d'urgence** : Repères visuels pour ne rater aucune échéance.
- **Filtres temps réel** : Retrouver instantanément une information ciblée.
- **Export local** : Sauvegarde des données en un clic (CSV/MD) pour une maîtrise totale de ses données.

1.8 8. Annexe API REST

Endpoint	Méthode	Description
/api/v1/emails/sync	POST	Déclenche la synchronisation avec l'API Gmail de l'utilisateur.
/api/v1/emails/classify	POST	Lance l'analyse par l'IA locale (selon le Persona actif).
/api/v1/tasks/	GET	Récupère la liste des tâches triées pour affichage dans le Kanban.
/api/v1/export/	GET	Génère le fichier d'export du Kanban (Feature F6).