

PROJET Odoo

5ème Année en Ingénierie Informatique et Réseaux

Conception et mise en œuvre d'un module de gestion intelligente des courriels dans Odoo avec intégration de l'IA : MailCourriel



Réalisé par :

Mohammed ETTALBI

Tuteur :

M. Mohammed AITDAOUD

Année universitaire : 2025/2026

Remerciements

Avant d'entrer dans les détails de cette expérience professionnelle, il me semble important de commencer ce rapport de projet par des remerciements envers toutes les personnes qui m'ont apporté une aide précieuse tout au long de cette période, faisant de ce projet une expérience à la fois enrichissante et formatrice.

Tout d'abord, je tiens à exprimer ma profonde gratitude envers **ALLAH**, le Tout-Puissant, qui m'a inspiré et guidé sur le bon chemin. Je Lui suis redevable de ce que je suis devenu, et je Lui adresse mes louanges et mes remerciements.

Je souhaite également remercier chaleureusement M. **Mohammed AITDAOUD**, mon encadrant pédagogique, pour sa patience, son accompagnement, ses conseils avisés et sa confiance. Sa disponibilité et ses précieuses orientations ont largement contribué à la réussite de ce travail.

Enfin, je remercie l'ensemble de l'équipe pédagogique et administrative de l'EMSI pour les efforts déployés afin de m'offrir une formation de qualité et pour leur accompagnement tout au long de mon parcours académique.

Résumé

Ce projet porte sur la conception et la mise en œuvre d'un **module de gestion intelligente des courriels** au sein de l'ERP Odoo, enrichi de fonctionnalités d'intelligence artificielle. L'objectif principal est de fournir aux utilisateurs une interface moderne de type Outlook permettant de gérer efficacement leurs emails, avec l'assistance d'un modèle de langage local (LLaMA 3.2 via Ollama) pour **résumer les emails, suggérer des réponses et assister à la rédaction**.

Le module intègre une **architecture conteneurisée Docker** comprenant Odoo 17, PostgreSQL 16 et un service IA local. Il propose une **interface à trois panneaux** (dossiers, liste des emails, prévisualisation) inspirée des clients de messagerie modernes, ainsi qu'un **système de dossiers et d'étiquettes** pour organiser les courriels de manière efficace et personnalisée.

Ce rapport présente l'ensemble du processus de réalisation, depuis l'analyse des besoins métier jusqu'à la conception de l'architecture technique, le développement de la solution OWL (Odoo Web Library) et l'intégration du modèle LLaMA. Il détaille les choix techniques adoptés et les résultats obtenus, mettant en évidence **la pertinence et l'impact opérationnel** de la solution développée.

Abstract

This project focuses on the design and implementation of an **intelligent email management module** within the Odoo ERP system, enhanced with **artificial intelligence** capabilities. The main objective is to provide users with a modern Outlook-style interface for efficient email management, assisted by a local language model (LLaMA 3.2 via Ollama) for **email summarization, reply suggestions, and drafting assistance**.

The module integrates a **containerized Docker architecture** comprising Odoo 17, PostgreSQL 16, and a local AI service. It offers a **three-panel interface** (folders, email list, preview) inspired by modern email clients, along with a **folder and label system** for organizing emails efficiently and in a personalized manner.

This report presents the entire implementation process, from business requirements analysis to technical architecture design, OWL (Odoo Web Library) solution development, and LLaMA model integration. It details the technical choices adopted and the results obtained, highlighting the **relevance and operational impact** of the developed solution.

Liste des figures

| | |
|--|----|
| Figure 1 : Diagramme de cas d'utilisation - Gestion des Courriels | 18 |
| Figure 2 : Diagramme de cas d'utilisation - Fonctionnalités IA | 18 |
| Figure 3 : Diagramme de séquences - Réception d'un Email | 19 |
| Figure 4 : Diagramme de séquences - Résumé IA | 20 |
| Figure 5 : Diagramme d'activités - Email Entrant | 21 |
| Figure 6 : Diagramme d'activités- Rédaction Assistée IA | 22 |
| Figure 7 : Diagramme d'Architecture | 23 |
| Figure 8 : Module Gestion des Courriels dans la liste des applications Odoo..... | 26 |
| Figure 9 : Ollama avec modèle LLaMA 3.2 installé | 27 |
| Figure 10 : Mot de passe d'application Gmail | 27 |
| Figure 11 : Configuration serveur SMTP dans Odoo..... | 28 |
| Figure 12 : Activation de l'IMAP dans Gmail | 28 |
| Figure 13 : Configuration serveur IMAP dans Odoo | 29 |
| Figure 14 : Python Logo | 30 |
| Figure 15 : Odoo Logo..... | 30 |
| Figure 16 : Owl Logo..... | 30 |
| Figure 17 : Docker Logo | 31 |
| Figure 18 : PostgreSQL Logo | 31 |
| Figure 19 : Interface principale | 32 |
| Figure 20 : Panneau de navigation | 33 |
| Figure 21 : Liste des emails | 33 |
| Figure 22 : Panneau de prévisualisation avec boutons IA | 34 |
| Figure 23 : résumé IA | 35 |
| Figure 24 : Suggestion de réponse IA | 35 |
| Figure 25 : Rédaction assistée IA..... | 36 |
| Figure 26 : Résultat de la rédaction IA | 36 |
| Figure 27 : Gestion des étiquettes | 37 |

Liste des acronymes

- **AI / IA** : Artificial Intelligence / Intelligence Artificielle
- **API** : Application Programming Interface
- **CSS** : Cascading Style Sheets
- **Docker** : Plateforme de conteneurisation
- **ERP** : Enterprise Resource Planning
- **HTML** : HyperText Markup Language
- **IMAP** : Internet Message Access Protocol
- **JS** : JavaScript
- **LLM** : Large Language Model
- **ORM** : Object-Relational Mapping
- **OWL** : Odoo Web Library
- **PostgreSQL** : Système de gestion de base de données relationnelle
- **REST** : Representational State Transfer
- **SMTP** : Simple Mail Transfer Protocol
- **SSL** : Secure Sockets Layer
- **UI** : User Interface
- **UML** : Unified Modeling Language
- **XML** : eXtensible Markup Language
- **JSON** : JavaScript Object Notation

Table des matières

| | |
|--|----|
| <i>Remerciements.....</i> | 1 |
| Résumé | 2 |
| Abstract..... | 3 |
| Liste des figures | 4 |
| Liste des acronymes..... | 5 |
| Table des matières..... | 6 |
| Introduction générale | 9 |
| Chapitre 1 : <i>Contexte général du projet.....</i> | 10 |
| Introduction..... | 10 |
| 1. Présentation du projet | 10 |
| 1.1. Processus de gestion des courriels | 11 |
| 2. Cahier des charges | 12 |
| 2.1. Les règles de gestion | 12 |
| 3. Conduite du projet | 13 |
| Conclusion | 13 |
| Chapitre 2 : <i>Analyse et spécification des besoins</i> | 14 |
| Introduction..... | 14 |
| 1. Spécification des besoins :..... | 14 |
| 1.1. Identification des acteurs | 14 |
| 2. Spécification des besoins fonctionnels | 15 |
| 3. Spécification des besoins non fonctionnels | 15 |
| Conclusion | 16 |
| Chapitre 3 : <i>Conception de projet.....</i> | 17 |
| Introduction..... | 17 |
| 1. Méthode de conception : UML..... | 17 |
| 2. Diagrammes de cas d'utilisation : | 17 |
| 3. Diagramme de séquences | 19 |
| 3.1. Réception d'un Email : | 19 |
| 3.2. Résumé IA | 19 |
| 4. Diagramme d'activités : | 20 |

| | |
|---|----|
| 4.1. Gestion des utilisateurs | 20 |
| 4.2. Processus de prédition | 21 |
| 5. Architecture Globale du Système | 22 |
| Conclusion..... | 23 |
| Chapitre 4 : <i>La mise en place de la solution</i> | 24 |
| Introduction..... | 24 |
| 1. Configuration de l'environnement Docker :..... | 24 |
| 1.1. Structure du projet | 24 |
| 1.2. Fichier docker-compose.yml | 25 |
| 1.3. Fichier odoo.conf..... | 25 |
| 1.4. Démarrage des conteneurs..... | 25 |
| 2. Installation et configuration d'Ollama | 26 |
| 2.1. Installation d'Ollama | 26 |
| 2.2. Téléchargement du modèle LLaMA 3.2 | 26 |
| 2.3. Démarrage du serveur Ollama | 26 |
| 2.4. Test de l'API Ollama | 27 |
| 3. Configuration du serveur SMTP (Gmail)..... | 27 |
| 3.1. Création d'un mot de passe d'application Gmail | 27 |
| 3.2. Configuration dans Odoo..... | 28 |
| 3.3. Test d'envoi | 28 |
| 4. Configuration du serveur IMAP (Fetchmail) | 28 |
| 4.1. Activation de l'IMAP dans Gmail | 28 |
| 4.2. Configuration dans Odoo..... | 29 |
| 4.3. Configuration du Fetchmail..... | 29 |
| 5. Structure du module mail_courriel..... | 29 |
| 5.1. Fichier manifest.py | 29 |
| 5.2. Modèles de données..... | 30 |
| 6. Présentation des outils | 30 |
| 6.1. Python..... | 30 |
| 6.2. Odoo 17 | 30 |
| 6.3. OWL (Odoo Web Library)..... | 30 |
| 6.4. Docker | 31 |
| 6.5. Ollama | 31 |

| | |
|-------------------------------------|----|
| 6.6. PostgreSQL..... | 31 |
| 7. Présentation des résultats | 32 |
| Conclusion | 37 |
| Conclusion générale..... | 38 |
| Références..... | 39 |

Introduction générale

Ce projet s'inscrit dans le cadre de notre formation en ingénierie informatique à l'EMSI Casablanca. Il constitue l'aboutissement des compétences techniques et méthodologiques acquises tout au long de notre parcours académique, tout en répondant aux exigences et aux problématiques réelles du monde professionnel. Face à la transformation numérique des entreprises et à l'importance croissante d'une gestion efficace des communications électroniques, nous avons choisi de développer un module de gestion intelligente des courriels au sein de l'ERP Odoo, intégrant des fonctionnalités d'intelligence artificielle.

Le module conçu a pour objectif d'offrir aux utilisateurs une interface moderne et ergonomique de type Outlook, permettant de consulter, organiser et répondre aux emails de manière efficace. L'intégration d'un assistant IA local (LLaMA 3.2 via Ollama) permet de résumer automatiquement les emails, suggérer des réponses professionnelles et assister à la rédaction de nouveaux messages, facilitant ainsi la productivité quotidienne des utilisateurs.

À travers ce rapport, nous présentons les différentes étapes de réalisation du module. Nous commençons par exposer le contexte général et les objectifs du projet, suivis d'une analyse des besoins fonctionnels et techniques. Ensuite, nous détaillons la conception de l'architecture et les choix technologiques retenus, avant de décrire la phase de développement et de mise en œuvre des fonctionnalités. Enfin, nous présentons les résultats obtenus, ainsi que les perspectives d'amélioration de la solution développée.

Chapitre 1 : *Contexte général du projet*

Introduction

La transformation numérique et l'exploitation des communications électroniques sont devenues des leviers stratégiques majeurs pour les entreprises modernes. Dans un contexte professionnel de plus en plus exigeant, la capacité à gérer efficacement les emails, organiser les informations et répondre rapidement aux sollicitations constitue un avantage déterminant. Le volume croissant des courriels reçus quotidiennement par les collaborateurs nécessite des outils performants et intelligents. Ce chapitre présente le cadre général du projet, les enjeux liés à la gestion intelligente des courriels, les règles de gestion définies, ainsi que l'architecture technique retenue pour concevoir une solution répondant aux besoins fonctionnels et techniques exprimés dans le cahier des charges.

1. Présentation du projet

L'ERP Odoo offre de nombreuses fonctionnalités métier intégrées, mais sa gestion native des emails peut être améliorée pour offrir une expérience utilisateur plus moderne et productive. Une interface de messagerie peu ergonomique peut entraîner une perte de temps, des emails non traités ou une difficulté à retrouver les informations importantes. Afin de répondre à ces problématiques, nous avons entrepris la réalisation d'un module de gestion intelligente des courriels basé sur l'intelligence artificielle.

Ce module permet aux utilisateurs d'exploiter une interface à trois panneaux inspirée des clients de messagerie modernes comme Outlook, afin de consulter, organiser et traiter leurs emails de manière intuitive et efficace. Il repose sur des techniques avancées de traitement du langage naturel via le modèle LLaMA 3.2, permettant de résumer les emails, suggérer des réponses professionnelles et assister à la rédaction de nouveaux messages.

Les objectifs principaux du projet sont les suivants :

- Offrir une interface moderne et ergonomique de type Outlook pour la gestion des emails ;
- Permettre l'organisation des courriels via un système de dossiers et d'étiquettes ;
- Fournir une interface intuitive et interactive pour l'analyse des données ;
- Assister les utilisateurs dans la rédaction de nouveaux messages grâce à l'IA.

1.1. Processus de gestion des courriels

Le module a été conçu pour refléter fidèlement le processus classique de gestion des emails, décliné selon les étapes suivantes :

i. Réception et affichage des courriels

- Synchronisation des emails via IMAP (Fetchmail) ou le système de messagerie Odoo.
- Affichage dans une liste avec indicateurs de lecture et métadonnées.
- Prévisualisation complète du contenu dans le panneau droit.

ii. Organisation et classement

- Gestion des dossiers système : Boîte de réception, Envoyés, Brouillons, Archives, Spam.
- Attribution d'étiquettes personnalisées avec codes couleur.
- Marquage comme lu/non lu et archivage.

iii. Envoi et réponse

- Composition de nouveaux emails avec éditeur HTML.
- Envoi via SMTP sécurisé (Gmail, Outlook, etc.).
- Réponse et transfert de messages existants.

iv. Assistance IA et rédaction

- Résumé automatique du contenu d'un email en 2-3 phrases.
- Suggestion de réponse professionnelle adaptée au contexte.
- Rédaction assistée de nouveaux emails à partir d'une instruction en langage naturel.

2. Cahier des charges

Les exigences fonctionnelles et techniques du module ont été définies à partir du cahier des charges initial, puis traduites en composants applicatifs cohérents.

2.1 Les règles de gestion

ii. Gestion des courriels

- Chaque email appartient à un dossier unique (inbox, sent, draft, archive, spam).
- Un email peut avoir plusieurs étiquettes personnalisées.
- Les emails ont un statut : brouillon, envoyé, lu, non lu, archivé.
- L'utilisateur ne peut accéder qu'à ses propres emails.

iii. Gestion des dossiers et étiquettes

- Les dossiers système sont créés automatiquement à l'installation.
- Les utilisateurs peuvent créer des dossiers personnalisés.
- Les étiquettes sont personnalisables avec des couleurs.

iv. Intégration IA

- Le service IA est accessible via Ollama sur le port 11434.
- Le modèle LLaMA 3.2 est utilisé pour le traitement du langage naturel.
- Les prompts sont construits dynamiquement selon le contexte de l'email.
- Les réponses IA sont générées en français.

v. Configuration email (SMTP/IMAP)

- Serveur SMTP pour l'envoi des emails (Gmail, Outlook, etc.).
- Serveur IMAP pour la réception des emails via Fetchmail.
- Authentification sécurisée SSL/TLS.

3. Conduite du projet

La réalisation de ce projet a été organisée en plusieurs phases successives, permettant une progression structurée et efficace. La première phase a consisté à mettre en place l'environnement de développement avec Docker (Odoo 17 + PostgreSQL 16). La deuxième phase a porté sur la création du module mail_courriel avec les modèles de données (courriels, dossiers, étiquettes). La troisième phase a été dédiée au développement de l'interface utilisateur OWL avec le layout à trois panneaux de style Outlook. La quatrième phase a permis l'intégration du service d'intelligence artificielle (Ollama + LLaMA 3.2) avec les fonctionnalités de résumé, suggestion de réponse et rédaction assistée. Enfin, la phase finale a été consacrée aux tests, à l'optimisation des performances et à la documentation du module. Cette approche par étapes a permis de valider progressivement chaque composant avant de passer au suivant, garantissant ainsi la qualité et la cohérence de la solution développée.

Conclusion

Ce premier chapitre a permis de présenter le contexte général du projet, ses objectifs et les principales règles de gestion qui encadrent le fonctionnement du module. Le développement d'une solution intelligente de gestion des courriels dans Odoo répond à un besoin réel des entreprises en matière de productivité et d'efficacité communicationnelle. En définissant clairement les processus de gestion des emails, les configurations SMTP et IMAP nécessaires, les fonctionnalités d'assistance IA, ainsi que les exigences fonctionnelles, nous avons posé les bases nécessaires à une implémentation structurée et efficace, qui sera détaillée dans les chapitres suivants.

Chapitre 2 : Analyse et spécification des besoins

Introduction

Ce chapitre a pour objectif de définir avec précision les besoins du système à concevoir. Il s'agit d'identifier les différents utilisateurs du module de gestion des courriels, les services attendus, ainsi que les exigences fonctionnelles et non fonctionnelles qui guideront la conception et l'implémentation de la solution. Cette analyse constitue une étape essentielle pour garantir que le module réponde efficacement aux besoins métiers de l'entreprise et aux attentes des utilisateurs finaux. Nous présenterons les acteurs impliqués, leurs interactions avec le système, les fonctionnalités attendues concernant la gestion des emails et l'intégration de l'intelligence artificielle.

1. Spécification des besoins :

Le module intelligent doit permettre à plusieurs types d'utilisateurs d'interagir avec le système à travers une interface intuitive, sécurisée et performant.

1.1 Identification des acteurs

Utilisateur standard

- Consulte ses emails dans l'interface Outlook-style à trois panneaux.
- Navigue entre les différents dossiers (Boîte de réception, Envoyés, Brouillons, etc.).
- Lit, compose et répond aux emails.
- Utilise les fonctionnalités IA (résumé, suggestion de réponse, rédaction assistée).
- Organise ses emails via les étiquettes personnalisées.

Administrateur système

- Gère les dossiers système et les étiquettes globales.
- Configure les paramètres SMTP et IMAP.
- Supervise le fonctionnement global du module via le backend Odoo.
- Gère les paramètres de connexion au service Ollama.

2. Spécification des besoins fonctionnels

Utilisateur standard

- Afficher les emails dans une interface à trois panneaux de style Outlook.
- Naviguer entre les dossiers (Inbox, Sent, Draft, Archive, Spam).
- Prévisualiser le contenu complet d'un email sélectionné.
- Composer, modifier et envoyer de nouveaux emails.
- Résumer un email via l'IA (bouton "Résumer").
- Obtenir une suggestion de réponse professionnelle via l'IA.
- Rédiger un email avec assistance IA à partir d'une instruction.

Administrateur

- Configurer les serveurs SMTP et IMAP.
- Gérer les paramètres de connexion Ollama (URL, modèle, timeout).
- Consulter les logs des appels IA.

3. Spécification des besoins non fonctionnels

- **Ergonomie** : Interface claire, intuitive et moderne de type Outlook.
- **Performance** : Temps de réponse optimisé pour le chargement des emails.

- **Sécurité** : Authentification sécurisée, modèle IA local garantissant la confidentialité.
- **Portabilité** : Déploiement via Docker pour une installation simplifiée.

Conclusion

L'analyse des besoins fonctionnels et non fonctionnels a permis de définir clairement les attentes des différents utilisateurs du module de gestion des courriels avec IA. En identifiant précisément les rôles (utilisateur standard, administrateur) et leurs interactions avec le système, cette étape a posé les bases d'une conception technique cohérente et évolutive. Le respect de ces spécifications guidera le développement de l'architecture logicielle et garantira une solution performante, sécurisée et adaptée aux exigences métier, tout en offrant une expérience utilisateur optimale.

Chapitre 3 : *Conception de projet*

Introduction

La phase de conception représente une étape critique dans le développement d'un module de gestion intelligente des courriels avec intégration IA. Elle consiste à transformer les besoins fonctionnels et métier exprimés précédemment en une architecture technique solide, reproductible et évolutive. Ce chapitre présente les modèles UML retenus pour représenter clairement les interactions entre les différents acteurs, les processus métier clés tels que la réception des emails, le résumé IA et la rédaction assistée, ainsi que la structure logique du système. Ces diagrammes serviront de référence tout au long du développement et faciliteront la maintenabilité à long terme.

1. Méthode de conception : UML

Nous avons choisi le langage UML (Unified Modeling Language) pour modéliser l'ensemble de la solution. UML offre une représentation visuelle normalisée permettant de formaliser les interactions homme-système, les flux de données et l'architecture logique des composants.

2. Diagrammes de cas d'utilisation :

Les diagrammes de cas d'utilisation décrivent les interactions possibles entre les différents acteurs du module et les fonctionnalités mises à leur disposition.

Ces diagrammes permettent ainsi de représenter clairement les rôles de chaque acteur et les fonctionnalités associées, offrant une vision globale et structurée du fonctionnement de la plateforme.

Cas d'Utilisation : Gestion des Courriels

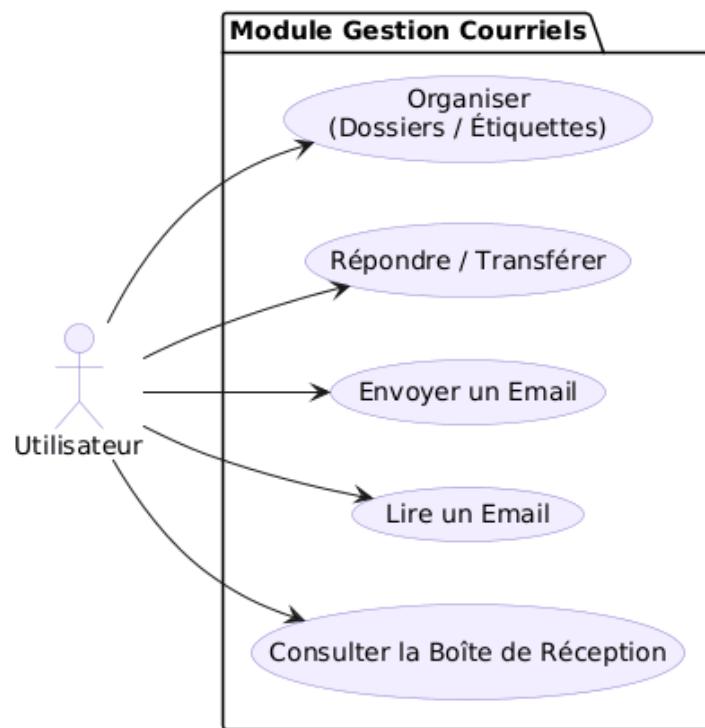


Figure 1 : Diagramme de cas d'utilisation - Gestion des Courriels

Cas d'Utilisation : Fonctionnalités IA

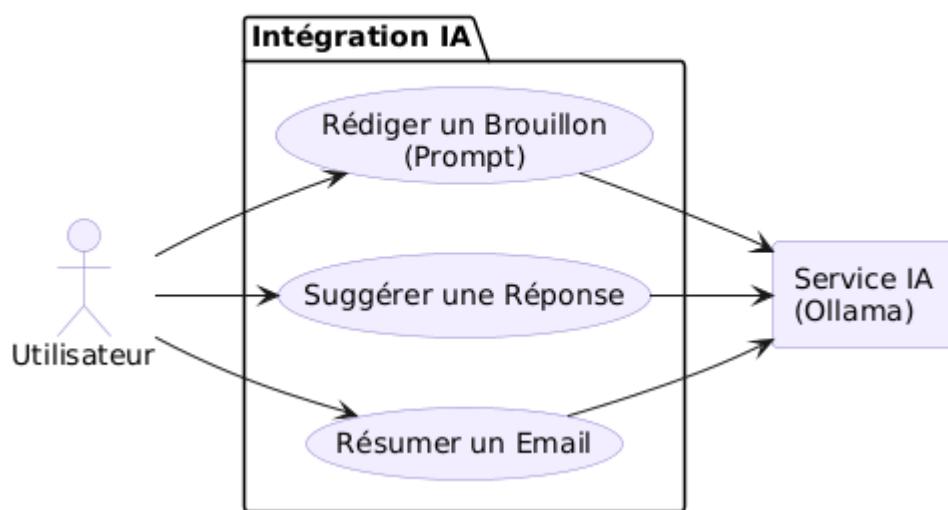


Figure 2 : Diagramme de cas d'utilisation - Fonctionnalités IA

3. Diagramme de séquences

Les diagrammes de séquences permettent de décrire, de manière chronologique, les échanges entre les différentes entités du système lors de l'exécution d'une fonctionnalité. Ces diagrammes détaillent les messages transmis entre les objets et définissent la logique d'interaction.

3.1. Réception d'un Email :

Lorsqu'un email arrive sur le serveur de messagerie, le service Fetchmail d'Odoo récupère le message via IMAP. Le message est ensuite traité par le module mail_thread d'Odoo qui l'analyse et transmet les données au modèle mail.courriel.

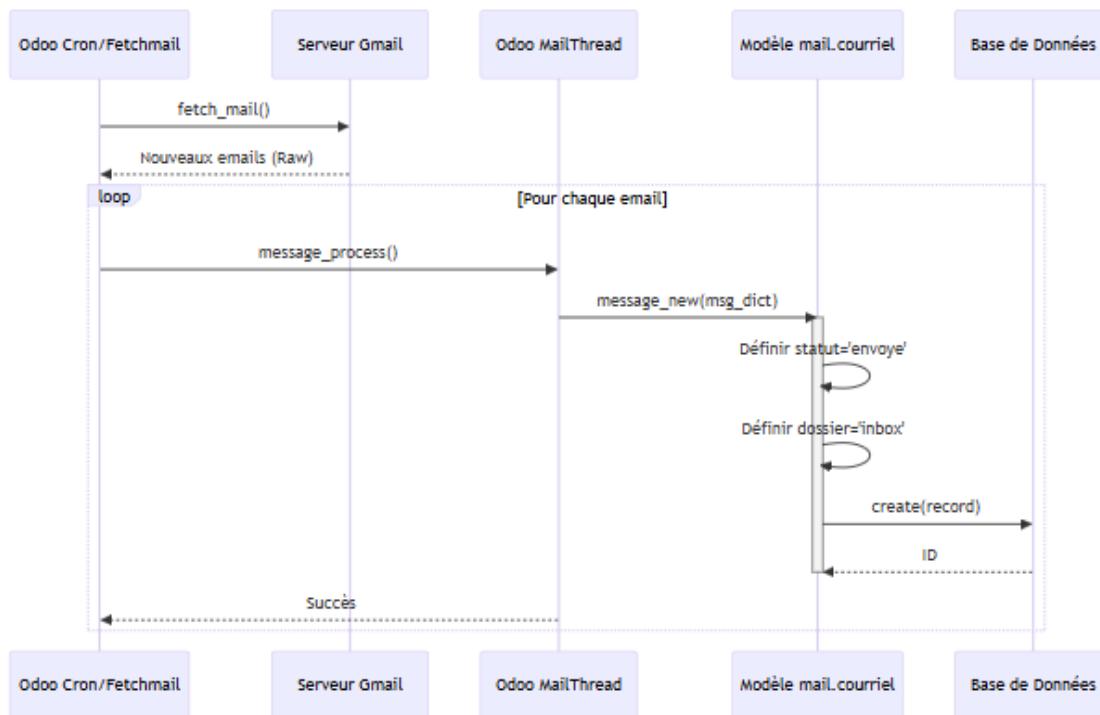


Figure 3 : Diagramme de séquences - Réception d'un Email

3.2. Résumé IA

L'utilisateur clique sur "Résumer", l'interface OWL envoie une requête au backend Odoo. Le service mail.ai construit un prompt et envoie une requête POST à l'API Ollama. Le modèle LLaMA 3.2 génère un résumé qui est retourné à l'interface.

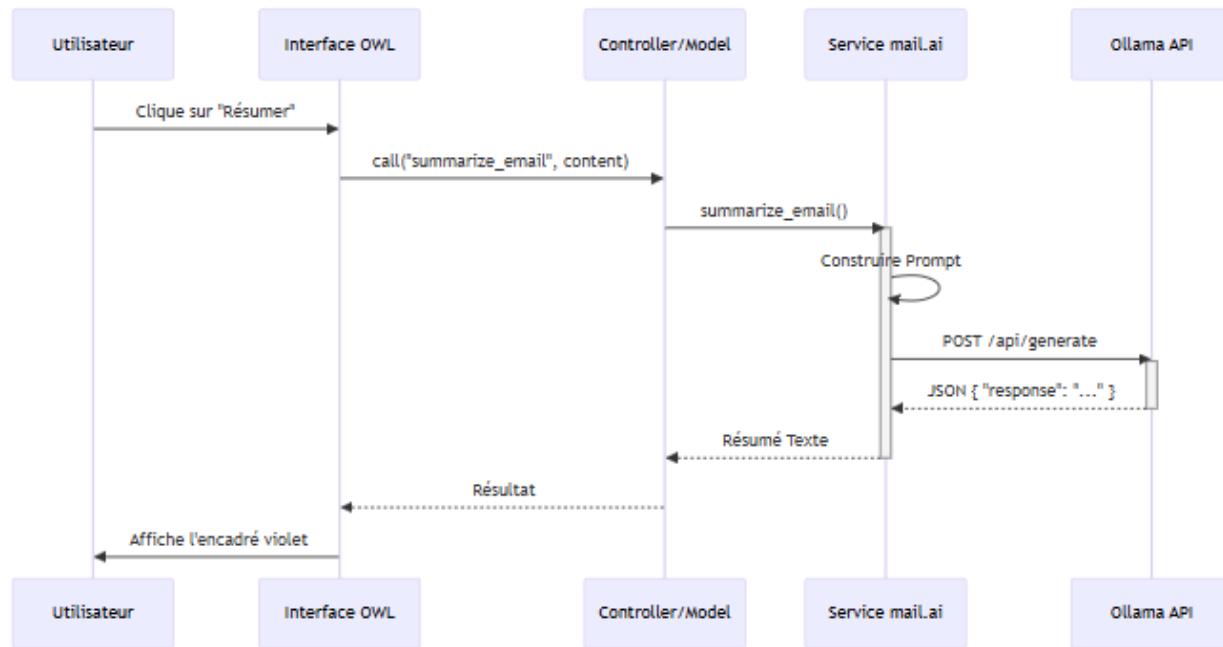


Figure 4 : Diagramme de séquences - Résumé IA

4. Diagramme d'activités :

Les diagrammes d'activités permettent de représenter, de manière claire et structurée, le déroulement complet des processus métier et techniques au sein du module. Ils décrivent la succession des actions, les points de décision et les flux de traitement, depuis l'initiation d'une opération par l'utilisateur jusqu'à la production du résultat final. Ces diagrammes offrent une vision globale du fonctionnement du système, en mettant en évidence l'enchaînement des étapes automatisées et les interactions nécessaires.

4.1. Gestion des utilisateurs

Ce diagramme d'activités présente le flux de gestion des utilisateurs par l'administrateur, depuis l'authentification jusqu'à l'ajout, la modification ou la suppression des comptes, tout en appliquant les règles de sécurité et de validation avant la mise à jour de l'affichage.

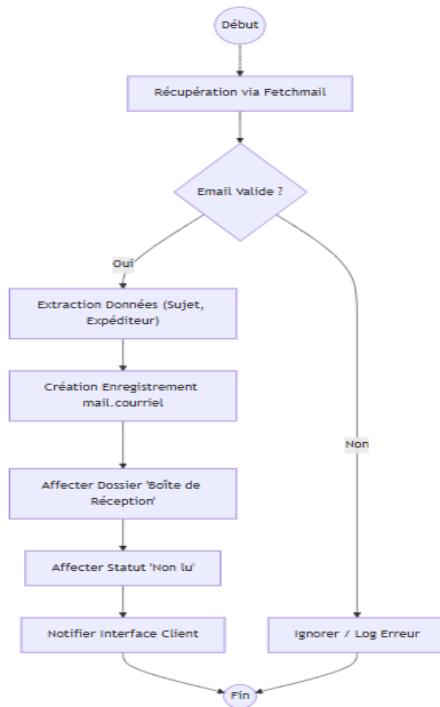


Figure 5 : Diagramme d'activités - Email Entrant

4.2. Processus de prédiction

Ce diagramme d'activités décrit le déroulement complet du processus de prédiction, depuis l'ouverture de l'application et le chargement des données (via une source financière ou un fichier CSV), jusqu'à l'entraînement du modèle LSTM, le calcul des métriques, la génération des prédictions et l'affichage des résultats sous forme de tableaux de bord interactifs.

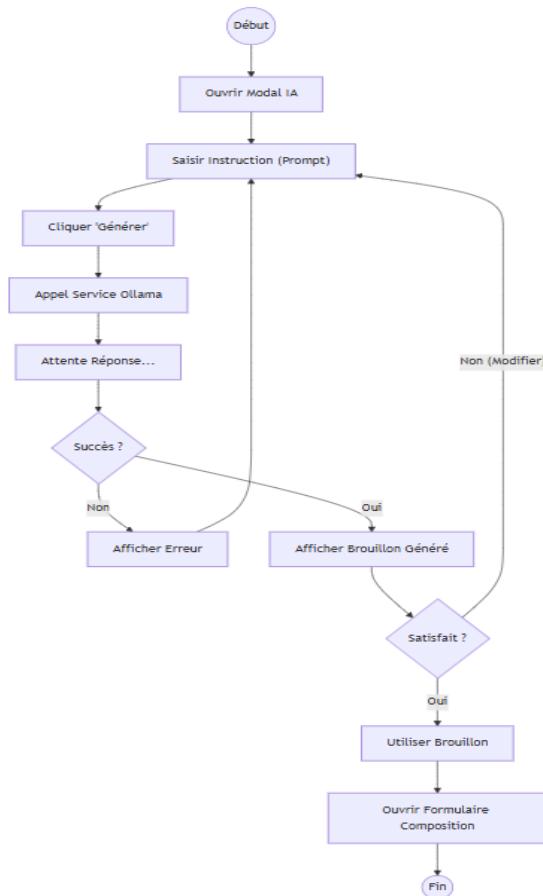


Figure 6 : Diagramme d'activités- Rédaction Assistée IA

5. Architecture Globale du Système

Le système interagit avec plusieurs acteurs et systèmes externes :

- **L'utilisateur**, qui consulte et gère ses emails via l'interface OWL.
- **L'administrateur**, qui configure le module et supervise son fonctionnement.
- **Le serveur SMTP**, qui envoie les emails.
- **Le serveur IMAP**, qui reçoit les emails.
- **Le service IA Ollama**, qui traite les requêtes de résumé et suggestion.

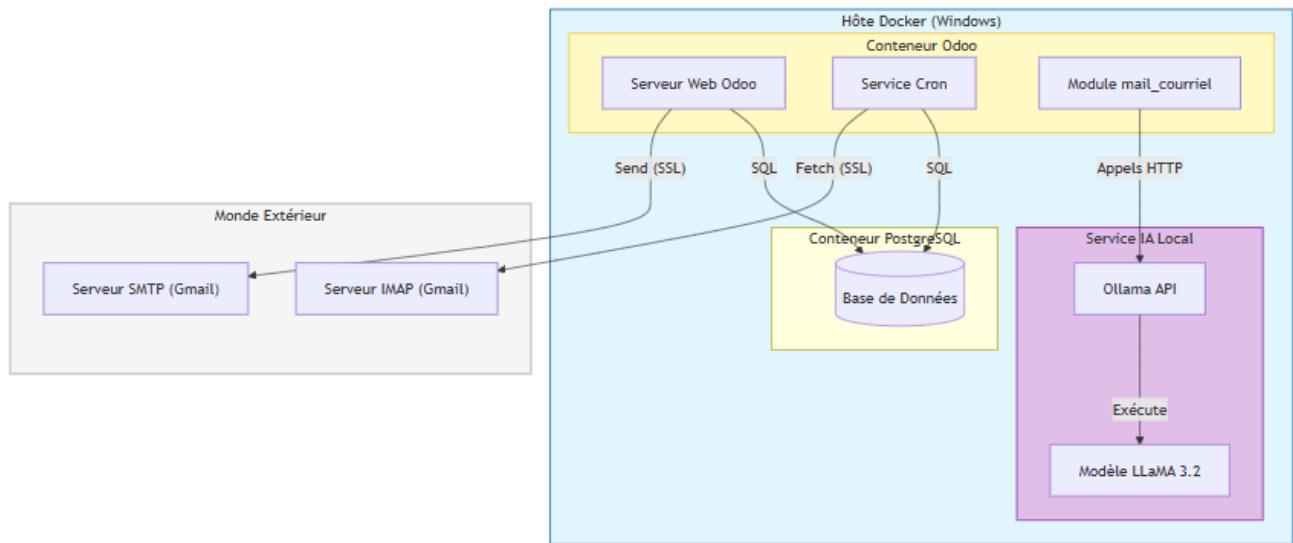


Figure 7 : Diagramme d'Architecture

Conclusion

La phase de conception a permis de structurer de manière rigoureuse l'ensemble des fonctionnalités du module de gestion intelligente des courriels à travers une modélisation UML claire et cohérente. L'utilisation des diagrammes de cas d'utilisation, de séquences et d'activités a facilité la compréhension des interactions entre les différents acteurs du système (utilisateur, administrateur) ainsi que les échanges entre les composants applicatifs, le service d'intelligence artificielle Ollama et la base de données PostgreSQL. Cette modélisation a également permis d'anticiper les principaux processus métier, notamment la réception et l'affichage des emails, le résumé automatique, la suggestion de réponse et l'assistance à la rédaction via le modèle LLaMA 3.2. Cette étape constitue une base solide pour l'implémentation technique de la solution..

Chapitre 4 : *La mise en place de la solution*

Introduction

Après avoir défini les besoins fonctionnels et réalisé une modélisation détaillée du système, cette phase de mise en œuvre marque le passage à la concrétisation du projet. Elle consiste à développer l'application en utilisant les technologies sélectionnées, à configurer son architecture, et à préparer l'environnement pour son exécution et son exploitation par les utilisateurs finaux. Ce chapitre présente en détail la mise en œuvre technique du projet, depuis la configuration de l'environnement Docker (Odoo 17 + PostgreSQL 16) jusqu'à l'intégration des serveurs de messagerie SMTP/IMAP et du service d'intelligence artificielle Ollama. Nous détaillerons également la structure du module mail_courriel, les étapes d'installation et les interfaces réalisées.

1. Configuration de l'environnement Docker :

1.1. Structure du projet

Le projet est organisé selon la structure suivante :

```
odoo-docker/
  ├── docker-compose.yml          # Configuration des conteneurs
  └── config/
      └── odooredit.conf          # Configuration Odoo
  ├── addons/
      └── mail_courriel/
          ├── __init__.py           # Module personnalisé
          ├── __manifest__.py
          ├── models/
          ├── views/
          ├── static/
          └── security/
  └── data/                         # Données PostgreSQL
```

1.2. Fichier docker-compose.yml

Le fichier docker-compose.yml configure les services Odoo et PostgreSQL :

```
version: '3.8'
services:
  odoo:
    image: odoo:17.0
    container_name: odoo17
    depends_on:
      - db
    ports:
      - "8069:8069"
    volumes:
      - ./addons:/mnt/extra-addons
      - ./config:/etc/odoo
    environment:
      - HOST=db
      - USER=odoo
      - PASSWORD=odoo
    extra_hosts:
      - "host.docker.internal:host-gateway"

  db:
    image: postgres:16
    container_name: postgres16
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=odoo
      - POSTGRES_PASSWORD=odoo
    volumes:
      - ./data:/var/lib/postgresql/data
```

1.3. Fichier odoo.conf

Le fichier de configuration Odoo spécifie les paramètres de connexion à la base de données et le chemin des addons :

```
[options]
addons_path = /mnt/extra-addons
admin_passwd = admin
db_host = db
db_port = 5432
db_user = odoo
db_password = odoo
```

1.4. Démarrage des conteneurs

Pour démarrer l'environnement, exécuter la commande suivante :

`docker-compose up -d`

Odoo sera accessible à l'adresse : <http://localhost:8069>

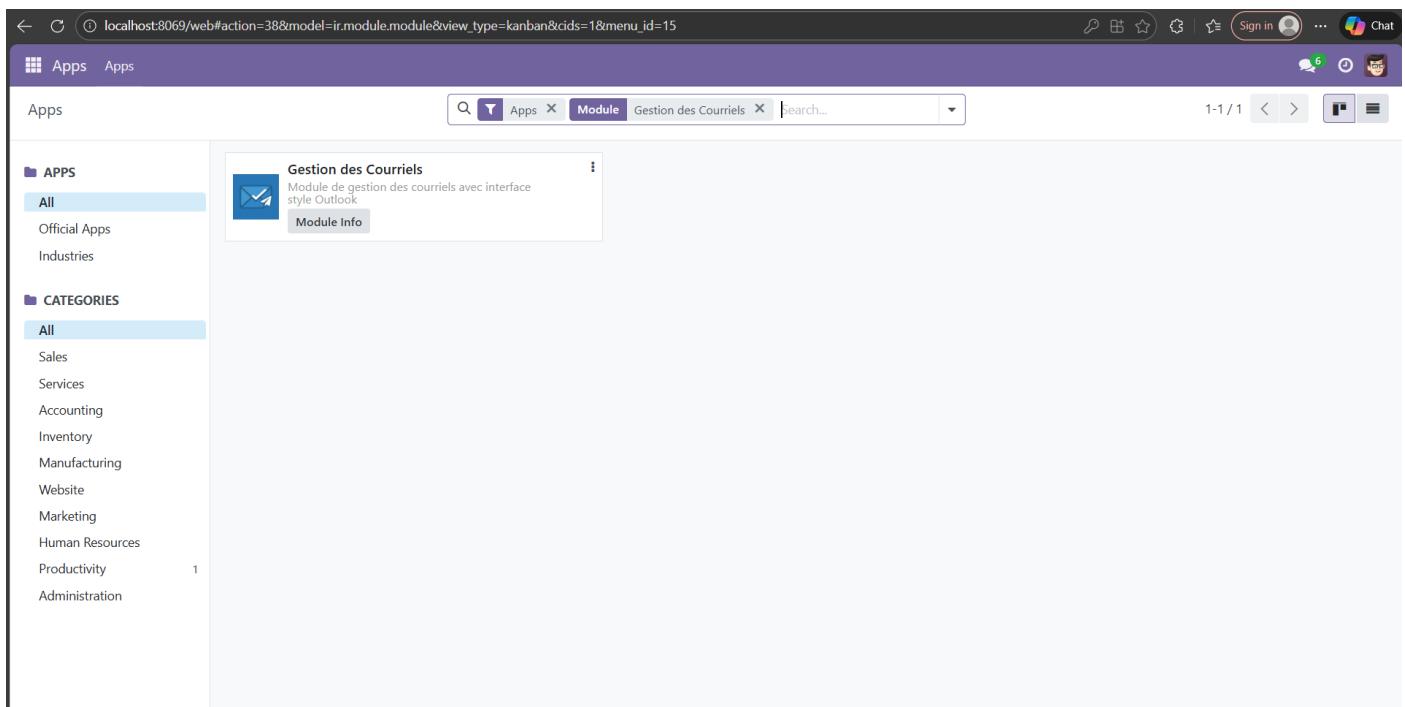


Figure 8 : Module Gestion des Courriels dans la liste des applications Odoo

2. Installation et configuration d'Ollama

2.1. Installation d'Ollama

Ollama doit être installé sur la machine hôte (Windows, macOS ou Linux) :

```
# Windows (PowerShell)
winget install Ollama.Ollama

# macOS
brew install ollama

# Linux
curl -fsSL https://ollama.com/install.sh | sh
```

2.2. Téléchargement du modèle LLaMA 3.2

Une fois Ollama installé, télécharger le modèle LLaMA 3.2 :

```
ollama pull llama3.2:latest
```

2.3. Démarrage du serveur Ollama

Le serveur Ollama doit être démarré avant d'utiliser les fonctionnalités IA :

```
ollama serve
```

Le service sera accessible sur le port 11434. Depuis le conteneur Odoo, l'URL est :

<http://host.docker.internal:11434>

```
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>ollama list
NAME           ID      SIZE    MODIFIED
Fine-tune-llm:latest 548edef9ba1c  2.3 GB   3 weeks ago
homic-embed-text:latest 0a109f422b47  274 MB   4 weeks ago
llama3.2:latest    a80c4f17acd5  2.0 GB   6 weeks ago

C:\Users\user>
```

Figure 9 : Ollama avec modèle LLaMA 3.2 installé

2.4. Test de l'API Ollama

Pour vérifier que le service fonctionne correctement :

```
curl http://localhost:11434/api/generate -d '{  
  "model": "llama3.2:latest",  
  "prompt": "Bonjour, comment ça va?",  
  "stream": false  
'
```

3. Configuration du serveur SMTP (Gmail)

3.1. Crédit d'un mot de passe d'application Gmail

Pour utiliser Gmail comme serveur SMTP, il faut créer un mot de passe d'application :

1. Accéder à <https://myaccount.google.com/apppasswords>
2. Sélectionner “Mail” et “Ordinateur Windows”
3. Générer le mot de passe et le noter

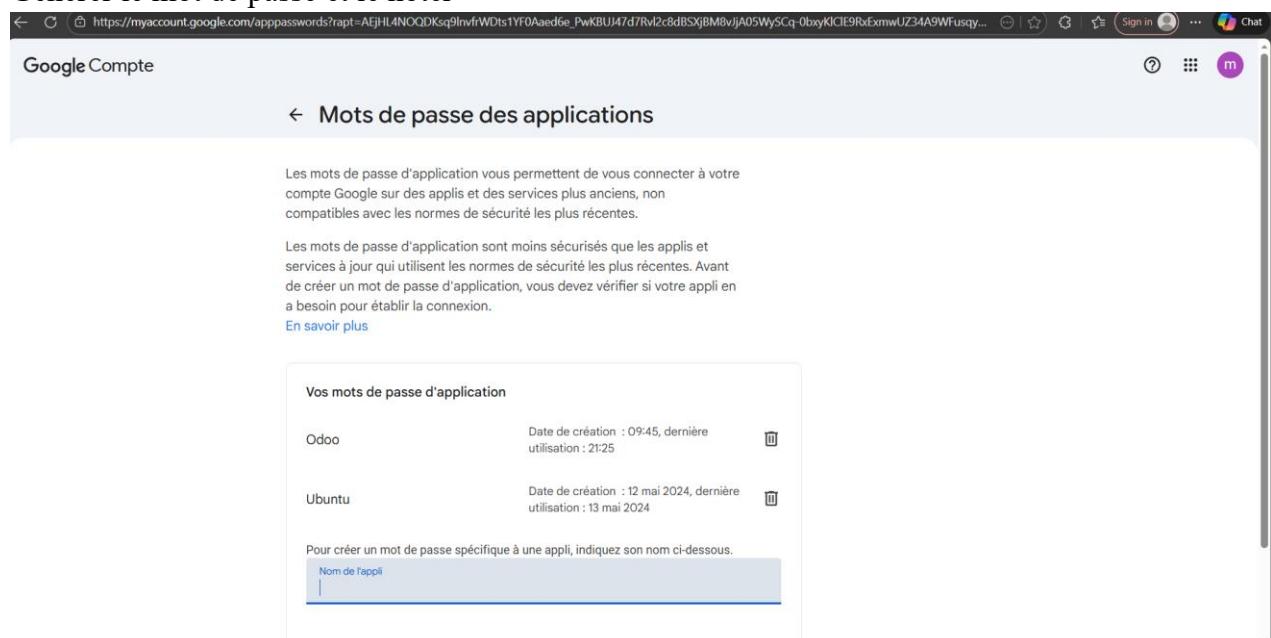
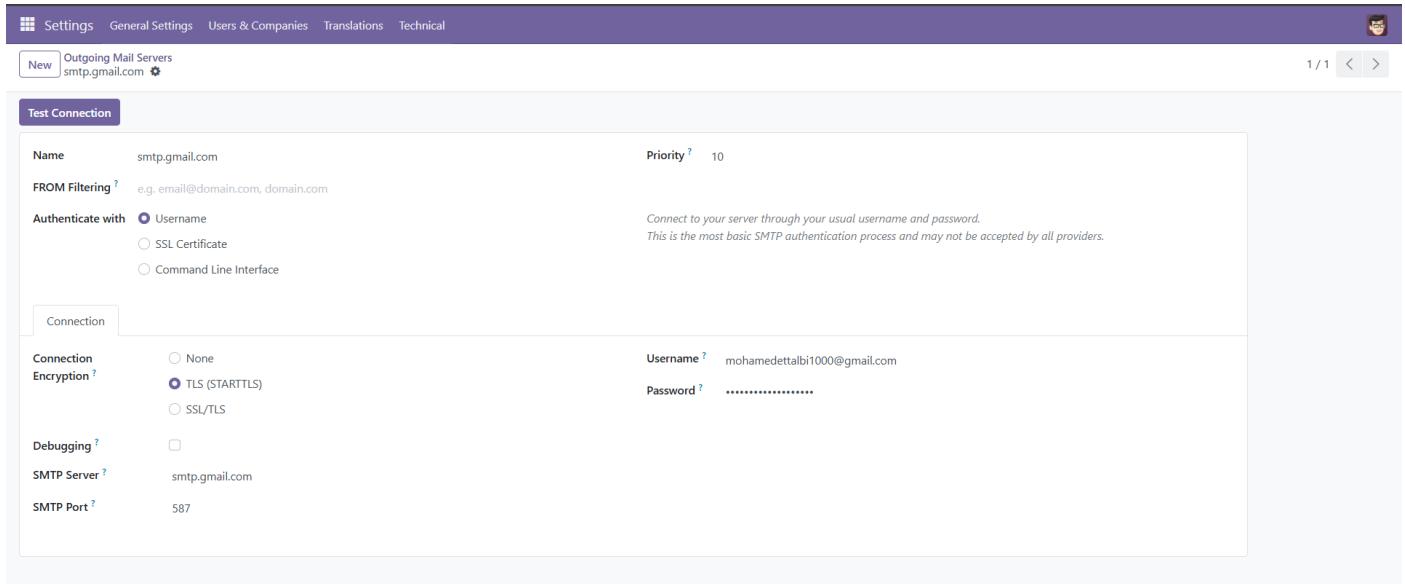


Figure 10 : Mot de passe d'application Gmail

3.2. Configuration dans Odoo

Dans Odoo, aller dans **Paramètres > Technique > Serveurs de messagerie sortants :**



The screenshot shows the Odoo Settings interface under 'Outgoing Mail Servers'. A new server 'smtp.gmail.com' is being configured. The 'Test Connection' tab is selected. The configuration includes:

- Name:** smtp.gmail.com
- Priority:** 10
- FROM Filtering:** e.g. email@domain.com, domain.com
- Authenticate with:** Username (selected)
- Connection Encryption:** TLS (STARTTLS) (selected)
- Username:** mohamedettalbi1000@gmail.com
- Password:** (redacted)
- Debugging:** Off
- SMTP Server:** smtp.gmail.com
- SMTP Port:** 587

Figure 11 : Configuration serveur SMTP dans Odoo

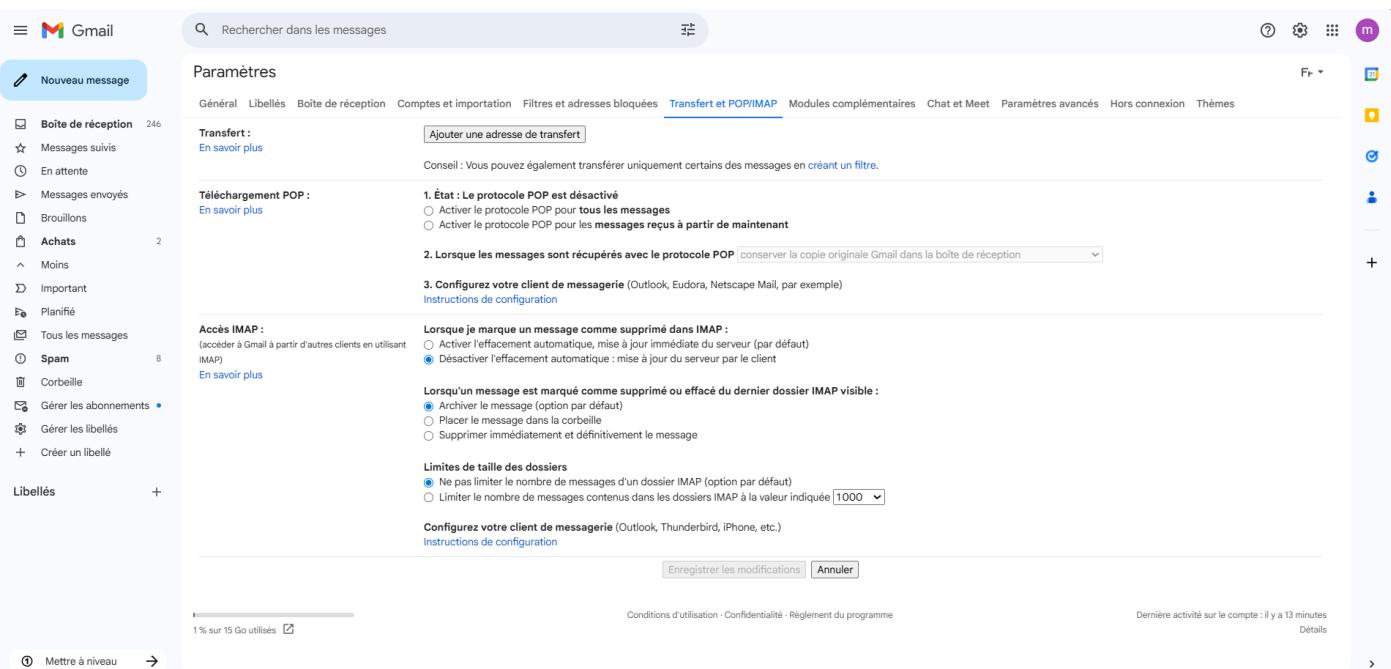
3.3. Test d'envoi

Cliquer sur “Tester la connexion” pour vérifier que la configuration est correcte.

4. Configuration du serveur IMAP (Fetchmail)

4.1. Activation de l'IMAP dans Gmail

Dans les paramètres Gmail, activer l'accès IMAP : **Paramètres > Voir tous les paramètres > Transfert et POP/IMAP > Activer IMAP**



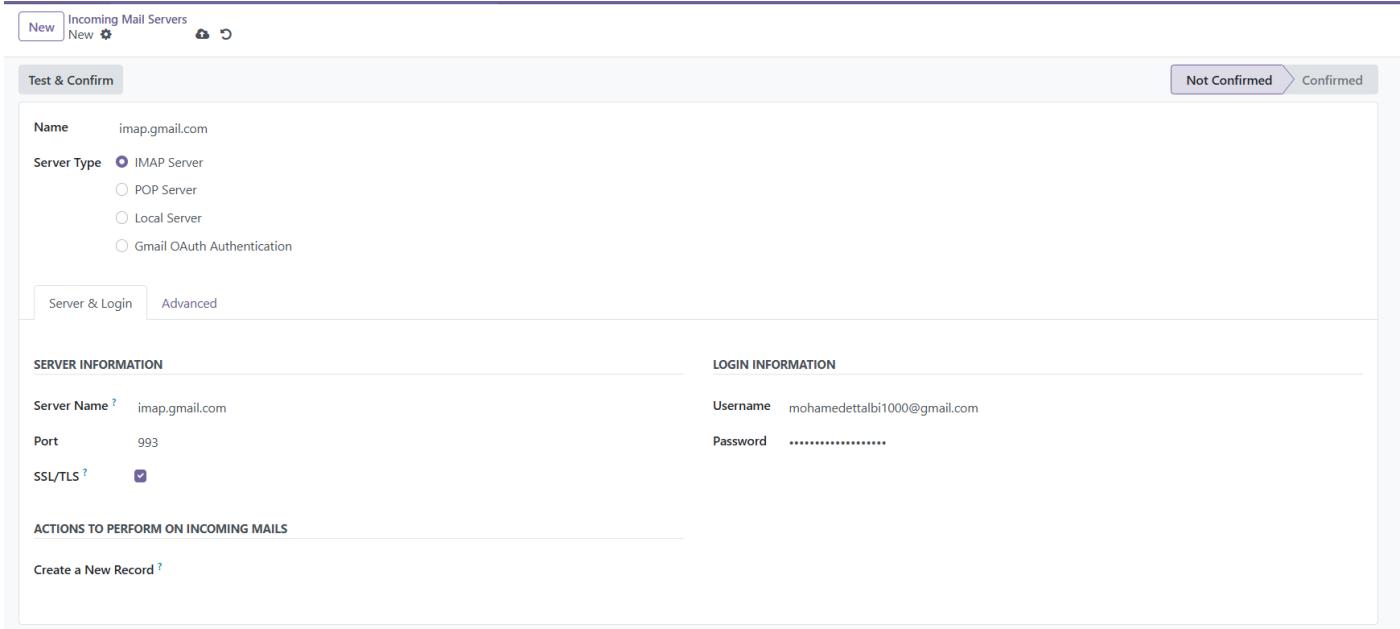
The screenshot shows the Gmail settings interface under 'Transfert et POP/IMAP'. The 'Accès IMAP' section is active, showing the following configuration options:

- Téléchargement POP :** En savoir plus
- État :** Le protocole POP est désactivé
 - Activer le protocole POP pour tous les messages
 - Activer le protocole POP pour les messages reçus à partir de maintenant
- Lorsque les messages sont récupérés avec le protocole POP :** conserver la copie originale Gmail dans la boîte de réception
- Configurez votre client de messagerie :** (Outlook, Eudora, Netscape Mail, par exemple)
- Lorsque je marque un message comme supprimé dans IMAP :**
 - Activer l'effacement automatique, mise à jour immédiate du serveur (par défaut)
 - Désactiver l'effacement automatique : mise à jour du serveur par le client
- Lorsqu'un message est marqué comme supprimé ou effacé du dernier dossier IMAP visible :**
 - Archiver le message (option par défaut)
 - Placer le message dans la corbeille
 - Supprimer immédiatement et définitivement le message
- Limites de taille des dossiers :**
 - Ne pas limiter le nombre de messages d'un dossier IMAP (option par défaut)
 - Limiter le nombre de messages contenus dans les dossiers IMAP à la valeur indiquée: 1000

Figure 12 : Activation de l'IMAP dans Gmail

4.2. Configuration dans Odoo

Dans Odoo, aller dans **Paramètres > Technique > Serveurs de messagerie entrants** :



The screenshot shows the 'Incoming Mail Servers' configuration screen in Odoo. A new server is being created for 'imap.gmail.com'. The 'Server Type' is set to 'IMAP Server'. The 'Name' field contains 'imap.gmail.com'. Under 'SERVER INFORMATION', the 'Server Name' is 'imap.gmail.com', 'Port' is '993', and 'SSL/TLS' is checked. Under 'LOGIN INFORMATION', the 'Username' is 'mohamedettalbi1000@gmail.com' and the 'Password' is masked. At the bottom, there is a link to 'Create a New Record'.

Figure 13 : Configuration serveur IMAP dans Odoo

4.3. Configuration du Fetchmail

Configurer le modèle de destination pour les emails entrants et définir la fréquence de synchronisation.

5. Structure du module mail_courriel

5.1. Fichier manifest.py

```
{
    'name': 'Mail Courriel - Gestion Intelligente',
    'version': '17.0.1.0.0',
    'category': 'Mail',
    'summary': 'Module de gestion des emails avec IA',
    'depends': ['mail', 'fetchmail'],
    'data': [
        'security/ir.model.access.csv',
        'data/mail_dossier_data.xml',
        'views/mail_courriel_views.xml',
        'views/mail_dossier_views.xml',
        'views/mail_etiquette_views.xml',
        'views/mail_client_action.xml',
        'views/menu_views.xml',
    ],
    'assets': {
        'web.assets_backend': [
            'mail_courriel/static/src/js/mail_client.js',
            'mail_courriel/static/src/xml/mail_client.xml',
            'mail_courriel/static/src/css/mail_client.css',
        ],
    },
}
```

```
'installable': True,
'application': True,
}
```

5.2. Modèles de données

Le module contient quatre modèles principaux :

- **mail.courriel** : Modèle principal des emails
- **mail.dossier** : Gestion des dossiers (Inbox, Sent, Draft, etc.)
- **mail.etiquette** : Gestion des étiquettes personnalisées
- **mail.ai** : Service d'intelligence artificielle (Ollama)

6. Présentation des outils

6.1. Python



Figure 14 : Python Logo

Le langage **Python** constitue le socle principal du développement backend. Il est utilisé pour l'implémentation des modèles Odoo, la logique métier et la communication avec le service IA Ollama. Grâce à son intégration native avec le framework Odoo et ses nombreuses bibliothèques, Python permet un développement rapide et une maintenance aisée du code..

6.2. Odoo 17



Figure 15 : Odoo Logo

Odoo 17 est l'ERP open-source qui sert de base au module. Il offre un framework complet pour le développement de modules métier, incluant un ORM puissant, un système de vues XML, et le framework OWL pour le frontend. Odoo assure également la gestion de l'authentification, des droits d'accès et de la persistance des données.

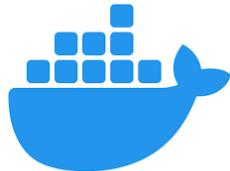
6.3. OWL (Odoo Web Library)



Figure 16 : Owl Logo

OWL est le framework JavaScript moderne d'Odoo, basé sur des composants réactifs. Il permet de créer des interfaces utilisateur dynamiques et performantes. Le composant MailCourrielClient a été développé entièrement en OWL, offrant une expérience de messagerie fluide et interactive.

6.4. Docker



Docker et Docker Compose sont utilisés pour la conteneurisation et le déploiement de l'environnement. Un fichier docker-compose.yml configure les conteneurs Odoo et PostgreSQL, assurant une installation simplifiée et une portabilité maximale de la solution..

Figure 17 : Docker Logo

6.5. Ollama



Ollama est le serveur d'inférence local permettant d'exécuter des modèles de langage. Il expose une API REST sur le port 11434, utilisée par le service mail.ai pour envoyer les prompts et recevoir les réponses générées par le modèle.

Figure 14 : ollama Logo

6.6. PostgreSQL



PostgreSQL 16 est le système de gestion de base de données relationnelle utilisé par Odoo. Il assure le stockage persistant de toutes les données du module (courriels, dossiers, étiquettes, paramètres) avec des performances optimales et une sécurité robuste..

Figure 18 : PostgreSQL Logo

7. Présentation des résultats

Dans cette partie, nous présentons notre travail en exposant des captures d'écran des interfaces les plus importantes de notre application.

Interface principale

La figure ci-dessous présente l'interface principale du module de gestion des courriels. Cette page constitue le cœur de l'application et affiche un layout à trois panneaux inspiré d'Outlook : le panneau gauche pour la navigation (dossiers, étiquettes, paramètres), le panneau central pour la liste des emails avec aperçu, et le panneau droit pour la prévisualisation complète de l'email sélectionné. Cette disposition permet une consultation efficace et une navigation intuitive entre les différents messages.

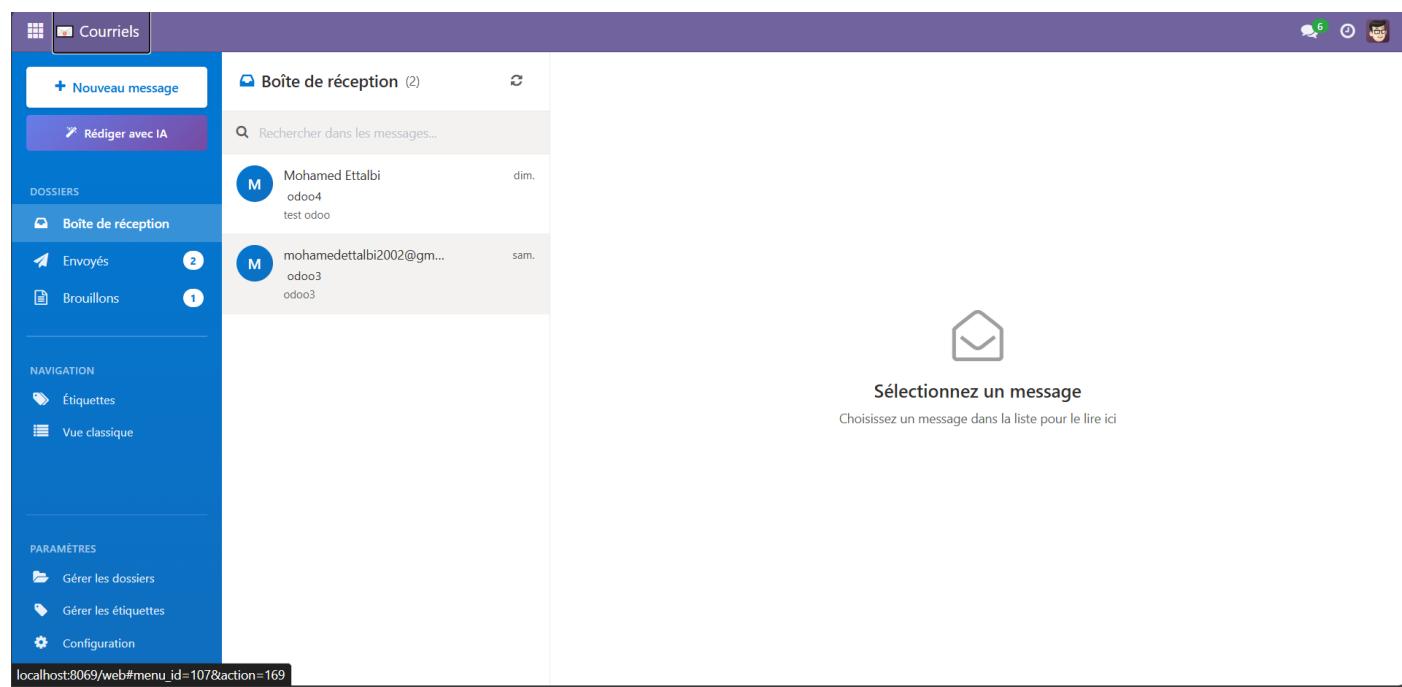


Figure 19 : Interface principale

Panneau de navigation avec dossiers

La figure ci-dessous présente le panneau de navigation situé à gauche de l'interface. Il affiche les dossiers système (Boîte de réception, Envoyés, Brouillons, Archives, Spam) avec leurs compteurs d'emails non lus. Des liens de navigation permettent d'accéder à la gestion des étiquettes, aux paramètres et à la vue classique Odoo. Le bouton “Rédiger avec IA” ouvre le modal d'assistance à la rédaction.

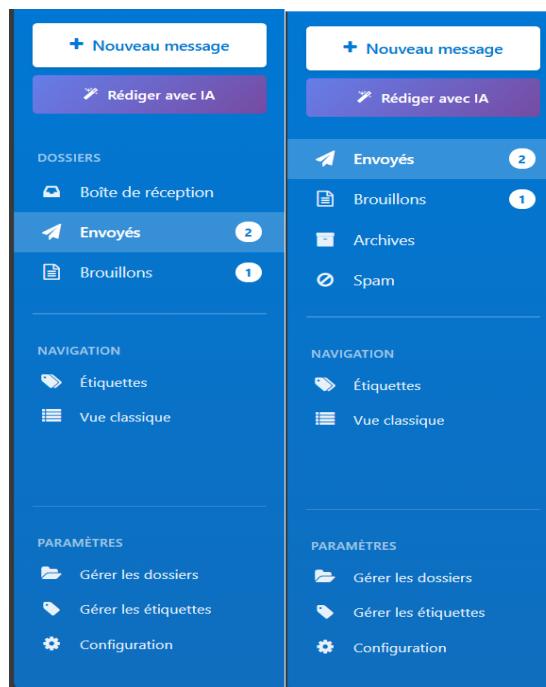


Figure 20 : Panneau de navigation

Liste des emails

La figure ci-dessous présente le panneau central affichant la liste des emails du dossier sélectionné. Chaque ligne montre l'expéditeur, le sujet, un aperçu du contenu et la date. Les emails non lus sont mis en évidence par un style distinctif. Un clic sur un email le sélectionne et affiche son contenu complet dans le panneau de prévisualisation.

| Courriels | | | | | | | | | Configuration |
|--------------------------|---------------------|----------------------|------------------------------|--------------------------------|--------------------|-----------|------------|--------|---------------|
| New | Mes Courriels | | Tous les courriels | | | Search... | | | |
| <input type="checkbox"/> | Date d'envoi | Objet | Expéditeur | Destinataires | Dossier | Priorité | Étiquettes | Statut | |
| <input type="checkbox"/> | 12/29/2025 16:14:48 | Re: odoo4 | YourCompany, Mitchell Admin | mohamedettalbi2002@gmail.co... | Envoyés | Normale | | 0 | |
| <input type="checkbox"/> | 12/28/2025 09:34:19 | from odoo | Mohamed Ettalbi | mohamedettalbi2002@gmail.co... | Envoyés | Urgente | | | 0 |
| <input type="checkbox"/> | 12/28/2025 09:31:21 | odoo4 | Mohamed Ettalbi | | Boîte de réception | Normale | | 0 | |
| <input type="checkbox"/> | 12/28/2025 00:15:55 | odoo3 | mohamedettalbi2002@gmail.com | | Boîte de réception | Normale | | 0 | |
| <input type="checkbox"/> | 12/27/2025 23:58:37 | odoo2 | YourCompany, Mitchell Admin | | Brouillons | Normale | | 0 | |
| <input type="checkbox"/> | 12/27/2025 23:22:19 | Test Failure Message | Azure Interior | mohamedettalbi2002@gmail.co... | Envoyés | Normale | | 0 | |
| <input type="checkbox"/> | 12/27/2025 23:13:04 | test Odoo | Azure Interior | mohamedettalbi2002@gmail.co... | Archivé | Normale | | | 0 |

Figure 21 : Liste des emails

Panneau de prévisualisation avec boutons IA

La figure ci-dessous présente le panneau de prévisualisation affiché à droite de l'interface. Il montre le contenu complet de l'email sélectionné avec l'expéditeur, le sujet, la date et le corps du message. Deux boutons d'action IA sont disponibles : “Résumer” pour générer un résumé automatique et “Suggérer réponse” pour obtenir une proposition de réponse professionnelle. Des actions standards (Répondre, Transférer, Archiver) sont également accessibles.

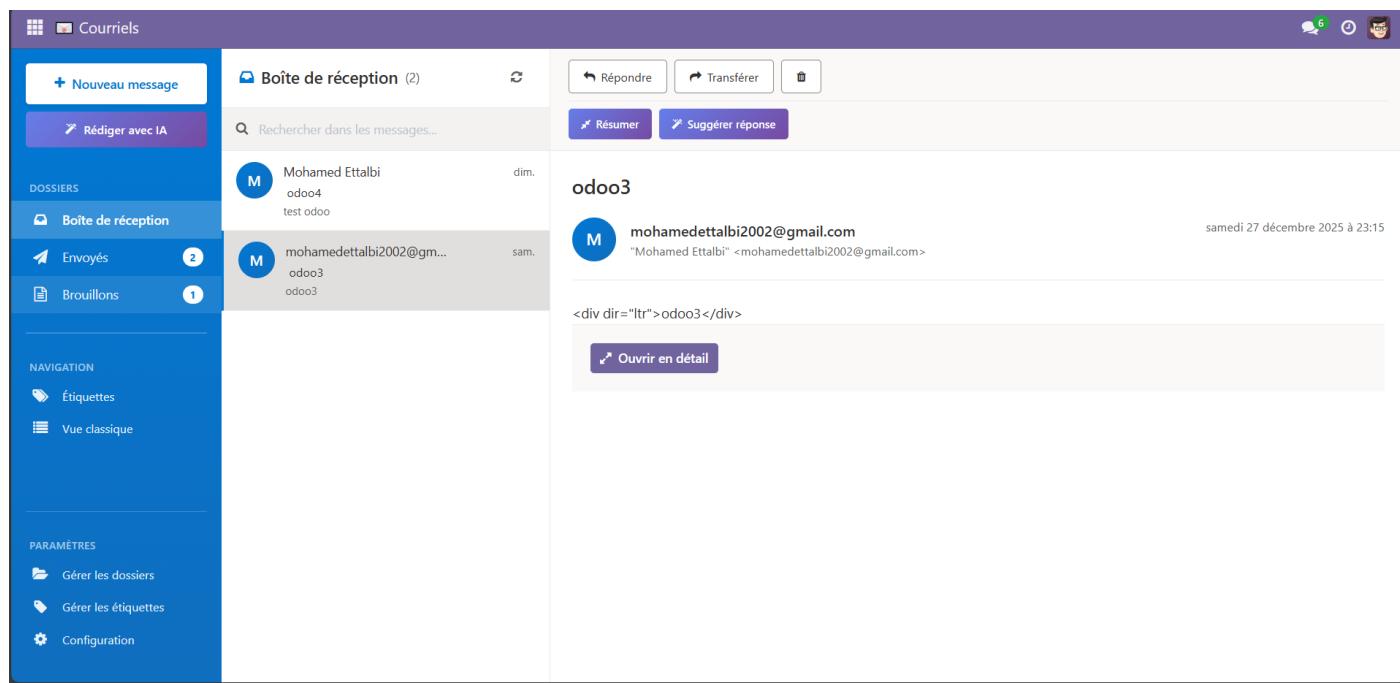


Figure 22 : Panneau de prévisualisation avec boutons IA

Résultat du résumé IA

La figure ci-dessous illustre l'affichage du résumé généré par l'IA. Lorsque l'utilisateur clique sur “Résumer”, le modèle LLaMA 3.2 analyse le contenu de l'email et génère un résumé concis en 2-3 phrases. Le résultat s'affiche dans un encadré violet distinct, permettant à l'utilisateur de comprendre rapidement l'essentiel du message sans lire l'intégralité du contenu.

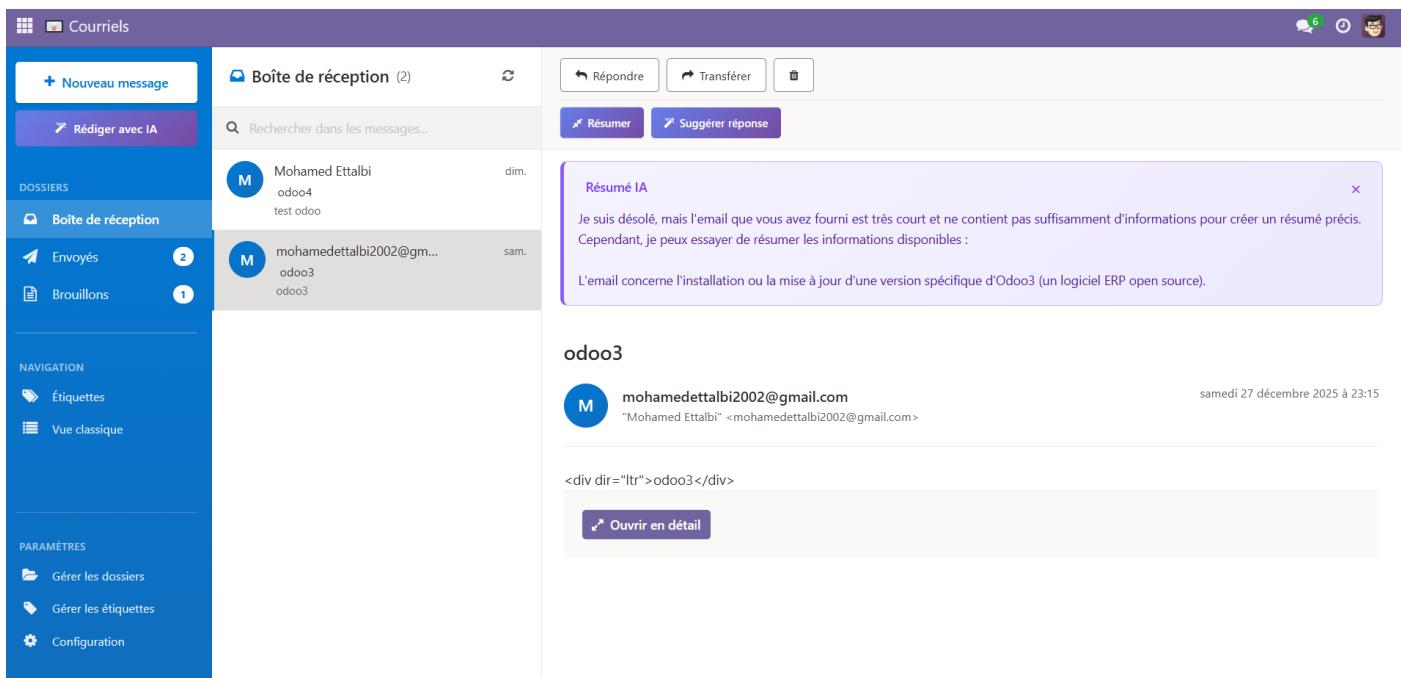


Figure 23 : résumé IA

Suggestion de réponse IA

La figure ci-dessous présente la fonctionnalité de suggestion de réponse. Lorsque l'utilisateur clique sur “Suggérer réponse”, le service IA analyse le contenu de l'email et génère une proposition de réponse professionnelle et courtoise. L'utilisateur peut ensuite utiliser cette suggestion pour composer sa réponse ou la modifier selon ses besoins.

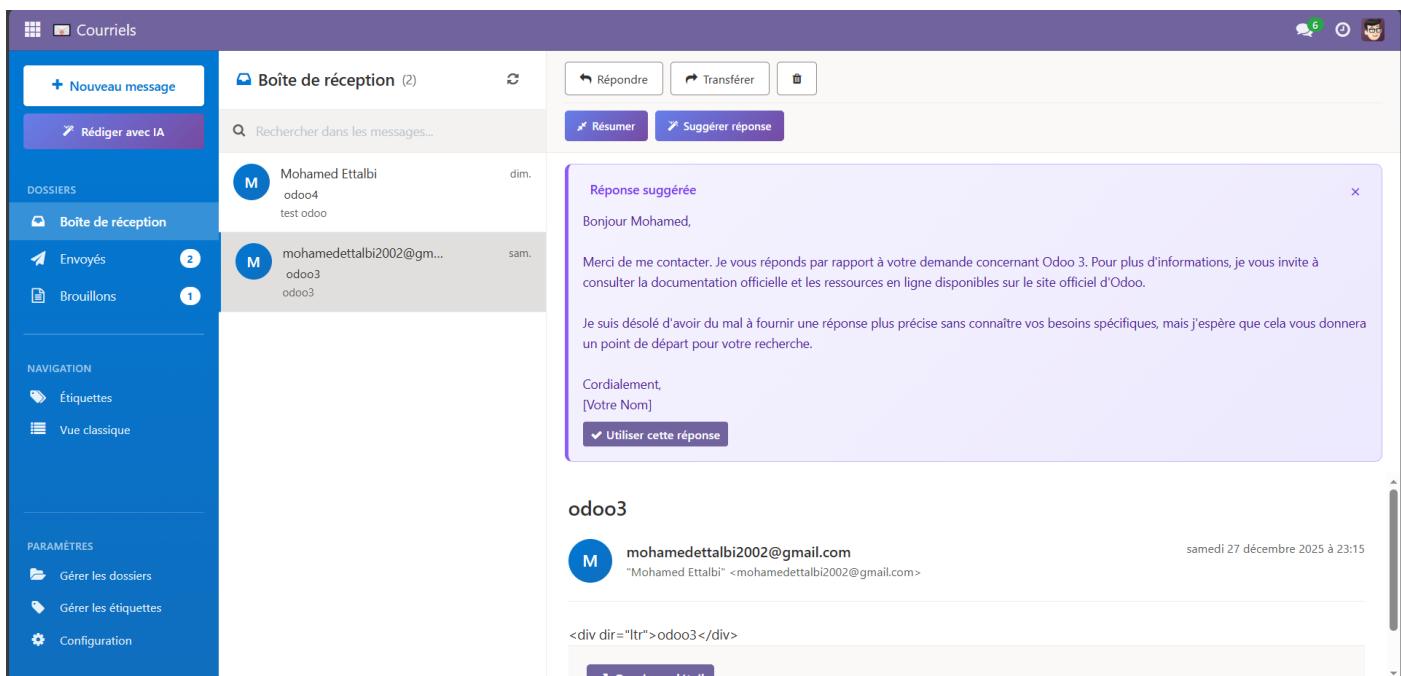


Figure 24 : Suggestion de réponse IA

Modal de rédaction assistée par IA

La figure ci-dessous présente le modal de rédaction assistée qui s'ouvre lorsque l'utilisateur clique sur “Rédiger avec IA”. Cette interface permet de saisir une instruction en langage naturel décrivant l'email souhaité (par exemple : “Rédige un email de remerciement pour une collaboration réussie”). L'utilisateur clique ensuite sur “Générer” pour obtenir un brouillon complet généré par le modèle LLaMA.

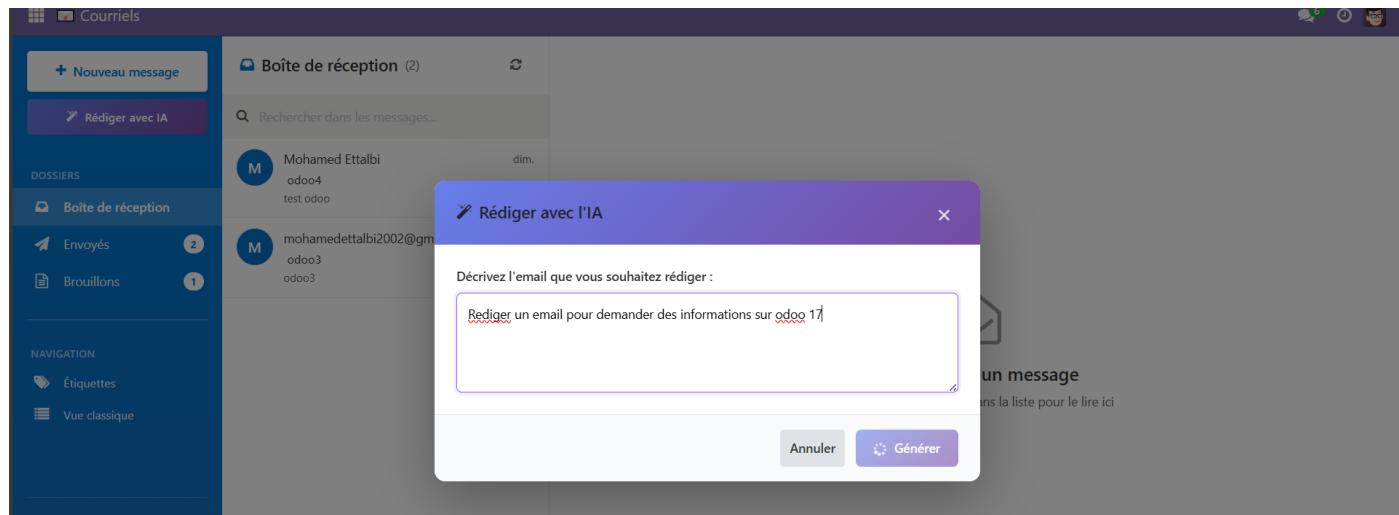


Figure 25 : Rédaction assistée IA

Résultat de la rédaction IA

La figure ci-dessous illustre le résultat de la génération d'email par l'IA. Le brouillon généré s'affiche dans le modal avec un bouton “Utiliser” permettant de créer un nouvel email avec ce contenu. L'utilisateur peut également modifier son instruction et régénérer s'il n'est pas satisfait du résultat.

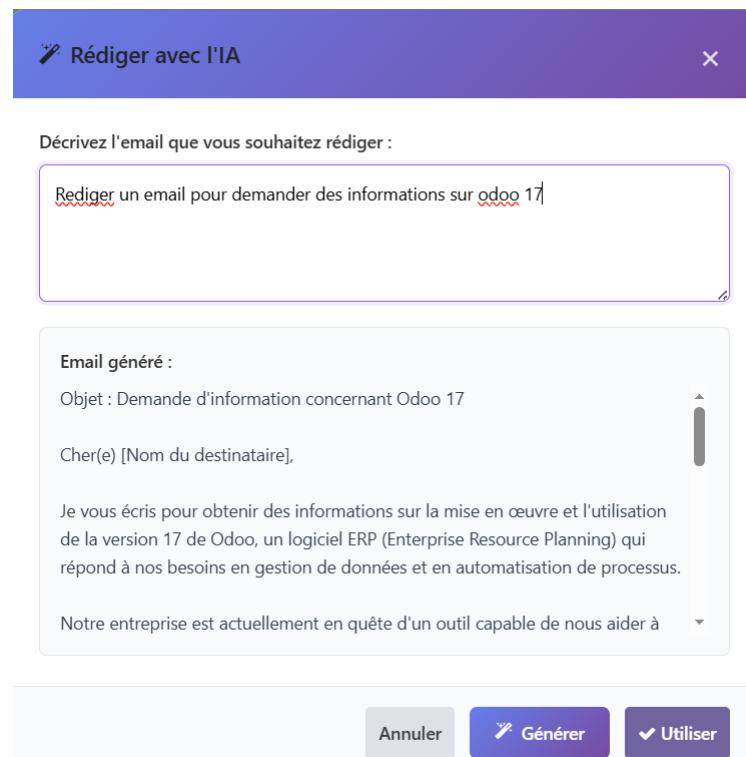
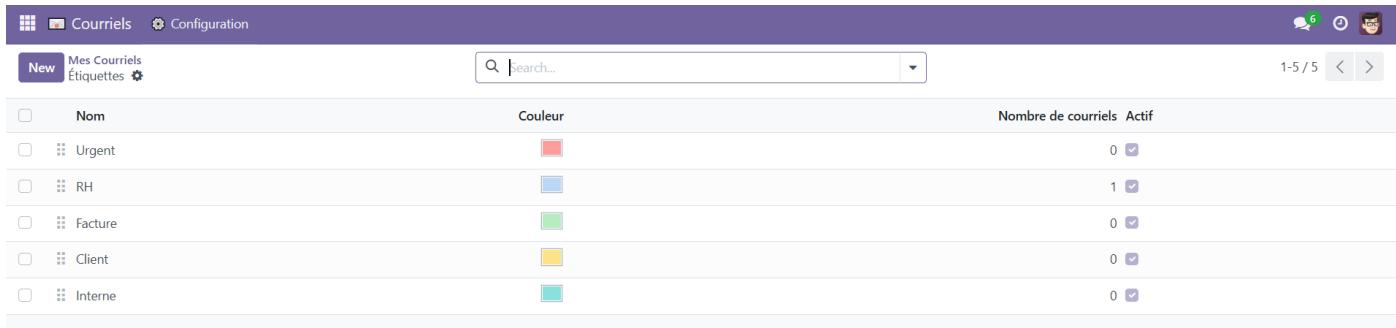


Figure 26 : Résultat de la rédaction IA

Gestion des étiquettes

La figure ci-dessous présente l'interface de gestion des étiquettes accessible depuis le panneau de navigation. Les utilisateurs peuvent créer, modifier et supprimer des étiquettes personnalisées avec des couleurs distinctives pour organiser leurs emails selon leurs propres critères.



| New | Nom | Couleur | Nombre de courriels | Actif |
|--------------------------|---------|---------|---------------------|-------------------------------------|
| <input type="checkbox"/> | Urgent | Red | 0 | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> | RH | Blue | 1 | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> | Facture | Green | 0 | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> | Client | Yellow | 0 | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> | Interne | Cyan | 0 | <input checked="" type="checkbox"/> |

Figure 27 : Gestion des étiquettes

Conclusion

La phase de mise en œuvre a permis de concrétiser le projet MailCourriel en un module fonctionnel de gestion intelligente des courriels, reposant sur une architecture cohérente et modulaire. L'utilisation d'Odoo 17 et du framework OWL pour l'interface utilisateur, de PostgreSQL pour la gestion des données, de Docker pour la conteneurisation, ainsi que l'intégration d'un assistant intelligent basé sur Ollama et LLaMA 3.2, ont permis de proposer une solution complète, interactive et orientée productivité. Les différentes interfaces développées couvrent l'ensemble du processus de gestion des emails, depuis la consultation et l'organisation jusqu'aux fonctionnalités d'assistance IA. Les résultats obtenus démontrent la stabilité, la performance et la pertinence du module, qui est prêt à être exploité dans un contexte réel tout en offrant des perspectives d'évolution vers des fonctionnalités avancées.

Conclusion générale

La réalisation de ce module de gestion intelligente des courriels s'est inscrite dans une démarche à la fois professionnelle et pédagogique visant à répondre à un besoin réel des entreprises : améliorer la productivité dans la gestion des emails grâce à une interface moderne et à l'assistance de l'intelligence artificielle.

Tout au long de ce projet, nous avons appliqué les principes de l'ingénierie logicielle pour concevoir une application robuste reposant sur une architecture moderne et modulaire. Le choix de technologies telles qu'Odoo 17 pour l'ERP, le framework OWL pour l'interface utilisateur, PostgreSQL pour la gestion des données, Docker pour la conteneurisation et Ollama/LLaMA 3.2 pour l'intégration IA, nous a permis de proposer un module fluide, évolutif et adapté aux besoins des utilisateurs professionnels.

Ce projet nous a permis de mettre en pratique de nombreuses compétences acquises durant notre formation, notamment la modélisation UML, la conception d'interfaces utilisateur modernes, le développement de modules Odoo (Python, OWL, XML), l'intégration de modèles de langage, la conteneurisation Docker, ainsi que l'adoption d'une méthodologie Agile (Scrum) pour la gestion du cycle de vie du projet.

Au-delà de l'aspect technique, ce travail a renforcé notre capacité à collaborer efficacement, à structurer un projet complexe et à respecter un cahier des charges exigeant. Il constitue une expérience concrète et enrichissante qui nous prépare activement au monde professionnel, en particulier dans les domaines de l'intelligence artificielle et du développement d'applications d'entreprise.

Perspectives :

Plusieurs pistes d'amélioration peuvent être envisagées, notamment l'intégration native IMAP/SMTP pour la synchronisation avec des serveurs de messagerie externes, l'ajout de fonctionnalités de recherche avancée et de filtres, le support multilingue pour les suggestions IA, l'amélioration des performances avec mise en cache des résultats, ou encore le développement d'une version mobile responsive. Ce projet représente ainsi une base solide pour une exploitation à plus grande échelle et une évolution future dans un contexte professionnel.

Références

1. **Documentation officielle Odoo 17**
<https://www.odoo.com/documentation/17.0/>
2. **Odoo OWL Framework**
<https://github.com/odoo/owl>
3. **Ollama – Run LLMs locally**
<https://ollama.ai/>
4. **LLaMA 3.2 Model – Meta AI**
<https://ai.meta.com/llama/>
5. **Docker Documentation**
<https://docs.docker.com/>
6. **PostgreSQL 16 Documentation**
<https://www.postgresql.org/docs/16/>
7. **Python Documentation**
<https://docs.python.org/3/>
8. **UML – Unified Modeling Language**
<https://www.omg.org/spec/UML/>
9. **Agile & Scrum Guide**
<https://scrumguides.org/>