

Cahier des charges

Chatbot interne IA

1. Cadrage du projet	2
Objectif général	3
Objectif du projet – POC (Proof of Concept)	3
Limites du POC.....	3
Étapes du projet.....	3
Contexte et enjeux	4
Parties prenantes.....	4
2. Méthodologie de travail.....	5
3. Spécifications fonctionnelles	6
Fonctionnalités principales	6
Critères d’acceptation	6
Détail des rôles utilisateurs	7
Interface utilisateur.....	7
Données utilisées	8
4. Spécifications techniques.....	8
Stack technique.....	8
Détail du choix des technologies.....	9
Diagramme BDD	9
Intégration du RAG (Retrieval-Augmented Generation).....	10
5. KPI de succès.....	11
6. Confidentialité des données.....	12
Chiffrement et stockage des données	12
Hébergement du POC	12
Gestion des rôles et authentification	12
Journaux d’activité (logs) et anonymisation	13
Conformité RGPD	13
Soutien au numérique responsable.....	13
7. Stratégie de validation et gestion des branches.....	14
8. Jalons & Livrables	14
9. Perspectives d’évolution et déploiement futur	15
Annexe : Questionnaire de satisfaction.....	15

1. Cadrage du projet

Objectif général

L'objectif du projet est de développer un chatbot interne, accessible via une interface web, pour accompagner les collaborateurs dans la compréhension des processus techniques complexes.

Le chatbot s'appuiera sur un modèle de langage (Mistral) et une architecture RAG (Retrieval-Augmented Generation).

Objectif du projet – POC (Proof of Concept)

L'objectif de ce projet est la réalisation d'un POC (prototype fonctionnel mais limité) visant à démontrer la faisabilité et la pertinence du chatbot .

Le projet s'inscrit dans une logique d'expérimentation rapide (2 à 3 semaines), avec une architecture modulaire et évolutive, permettant une montée en charge ultérieure vers une solution pleinement intégrée au système d'information.

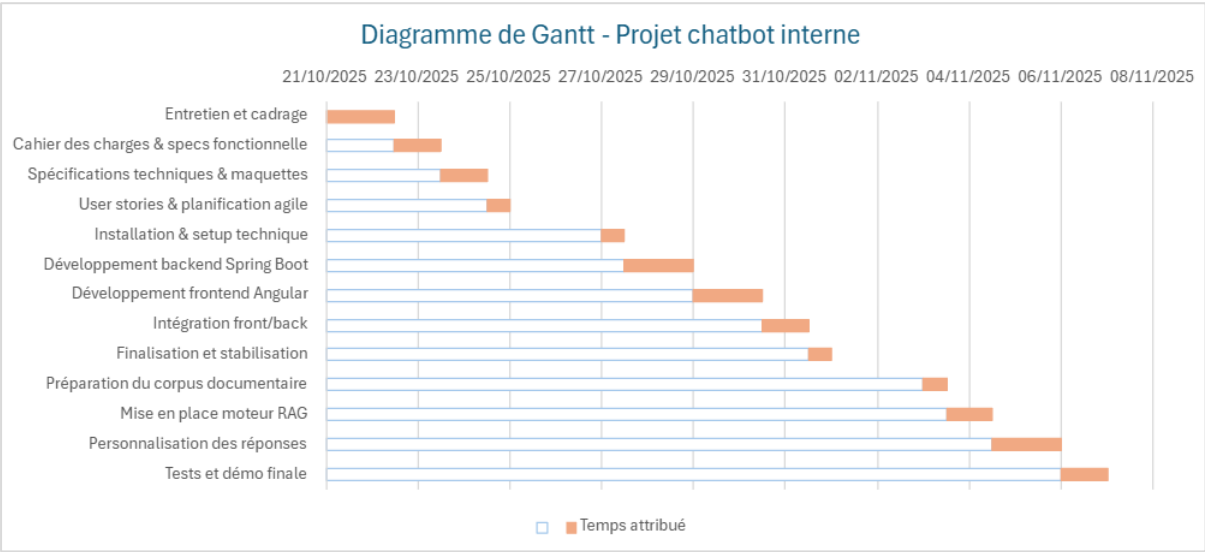
Limites du POC

Ce POC a pour objectif de démontrer la faisabilité technique et la pertinence fonctionnelle du chatbot interne, sans viser un déploiement en environnement de production. Il ne couvre donc pas les aspects liés à la montée en charge, à la sécurité avancée, à la supervision continue ni à l'intégration complète dans le système d'information de l'entreprise. Les tests seront réalisés sur un périmètre restreint, avec un volume limité de données. Les performances, la robustesse et la maintenabilité du prototype ne seront pas représentatives d'une version industrielle, mais serviront uniquement à valider les principes techniques et l'intérêt métier de la solution.

Étapes du projet

Le projet va se développer en 3 phases, en commençant par une phase de conception / cadrage qui va permettre de définir les besoins, processus, la planification et les spécifications fonctionnelles et techniques. Une deuxième phase portera sur le développement d'une application web avec le modèle conversationnel de Mistral AI intégré. La dernière phase reposera sur l'alimentation de l'agent conversationnel à travers un corpus documentaire comportant les processus métier / interne de l'entreprise concernée, dans le but d'obtenir un chatbot contextuel interne.

Le projet débute le 21 octobre avec une durée estimée à environ 3 semaines de travail (cf. Diagramme de Gantt pour le détail), pour 1 jour équivalent à 6h de travail.



Contexte et enjeux

Les processus de l’entreprise sont assez complexes, il est parfois difficile pour le collaborateur ou le client de s’y retrouver dans ces nombreuses phases. L’objectif du chatbot est d’avoir un moyen accessible (application web) pour monter rapidement en compétence sur le périmètre technique de l’entreprise à travers un échange fluide qui réduira les sollicitations des équipes techniques. Une conversation qui pourra aussi s’adapter en fonction d’un profil.

Dans ce contexte, de nombreux enjeux vont concerner la mise en place de la solution :

- Nécessité de sécurité et de cloisonnement des données internes
- Compréhension des processus internes complexes
- Prise en main technique du développeur (Spring AI, pgvector...)
- Intégration d’un chatbot avec une expertise moindre

En effet, le principal enjeu va être sécuritaire, touchant à des données internes et intégrant des solutions externes. Il est donc nécessaire de veiller à ce que l'accès aux données soit sécurisé et avoir une mise en place contrôlée en respectant les valeurs de l’entreprise. Le deuxième enjeu va porter sur la montée en compétence du développeur, n’ayant pas eu l’occasion d’expérimenter une telle solution.

Parties prenantes

Rôle	Acteur	Contribution
------	--------	--------------

AMOA	Loan	Cadrage, recueil besoin, spécifications
MOE	Loan	Développement de l'application
Sponsor	Direction métier	Validation du périmètre et du besoin
Utilisateurs	Clients/collaborateurs	Tests et feedbacks

2. Méthodologie de travail

Le projet sera mené selon une approche agile, en s'appuyant sur le framework Scrum, adaptée à un contexte d'équipe réduite (un seul développeur). Cette méthodologie s'impose ayant une vision réduite du périmètre du projet. Elle permettra une itération rapide, une meilleure visibilité sur l'avancement et une adaptation continue aux besoins fonctionnels et techniques.

L'organisation du travail se fera en deux sprints d'une semaine chacun :

- Sprint 1 : Développement de la première version du chatbot (avec Mistral et Spring AI).
- Sprint 2 : Intégration des processus métier de l'entreprise, adaptation du modèle RAG au corpus interne.

À la fin de chaque sprint, une Sprint Review sera organisée afin de présenter l'avancement, recueillir les retours et ajuster le backlog si nécessaire.

Les User Stories (US) seront rédigées selon les principes de priorisation MoSCoW (Must, Should, Could, Won't) et d'analyse INVEST (Indépendante, Négociable, Valeur, Estimable, Simple, Testable) avec des critères d'acceptation.

Un backlog produit sera tenu à jour et les US seront réparties sur les deux sprints selon leur priorité et leur complexité.

L'ensemble de la gestion de projet sera assuré via JIRA, qui permettra le suivi des tâches, du backlog et des sprints.

Le versioning du code sera réalisé sur GitHub, garantissant la traçabilité et la transparence du développement.

Les accès à JIRA et GitHub seront partagés avec le client pour assurer un suivi complet et collaboratif du projet.



3. Spécifications fonctionnelles

Fonctionnalités principales

1. **Chat conversationnel** (texte)
 - a. Questions libres ou suggérées
 - b. Réponses générées par Mistral / RAG
2. **Recherche documentaire intelligente**
 - a. Indexation de la documentation interne
 - b. Récupération des documents pertinents avant génération
3. **Profils et droits d'accès**
 - a. Utilisateur standard (lecture)
 - b. Expert technique (accès élargi)
 - c. Administrateur (gestion des documents)
4. **Interface web**
 - a. Connexion (compte interne)
 - b. Interface de chat (zone de saisie, affichage des réponses)
 - c. Historique des conversations (titre résumé, reprise de conversation)
 - d. Sidebar (historique, profil, paramètre du compte)
5. **Supervision / Feedback**
 - a. Feedback utilisateur (utile / non utile)
 - b. Mise à jour du corpus documentaire (admin)

Critères d'acceptation

Chaque fonctionnalité est décrite via des user stories comportant un critère d'acceptation. Celui-ci permet de déterminer si la fonctionnalité est correctement développée ou non.

Le tableau ci-dessous décrit les critères d'acceptation des fonctionnalités principales de l'application.

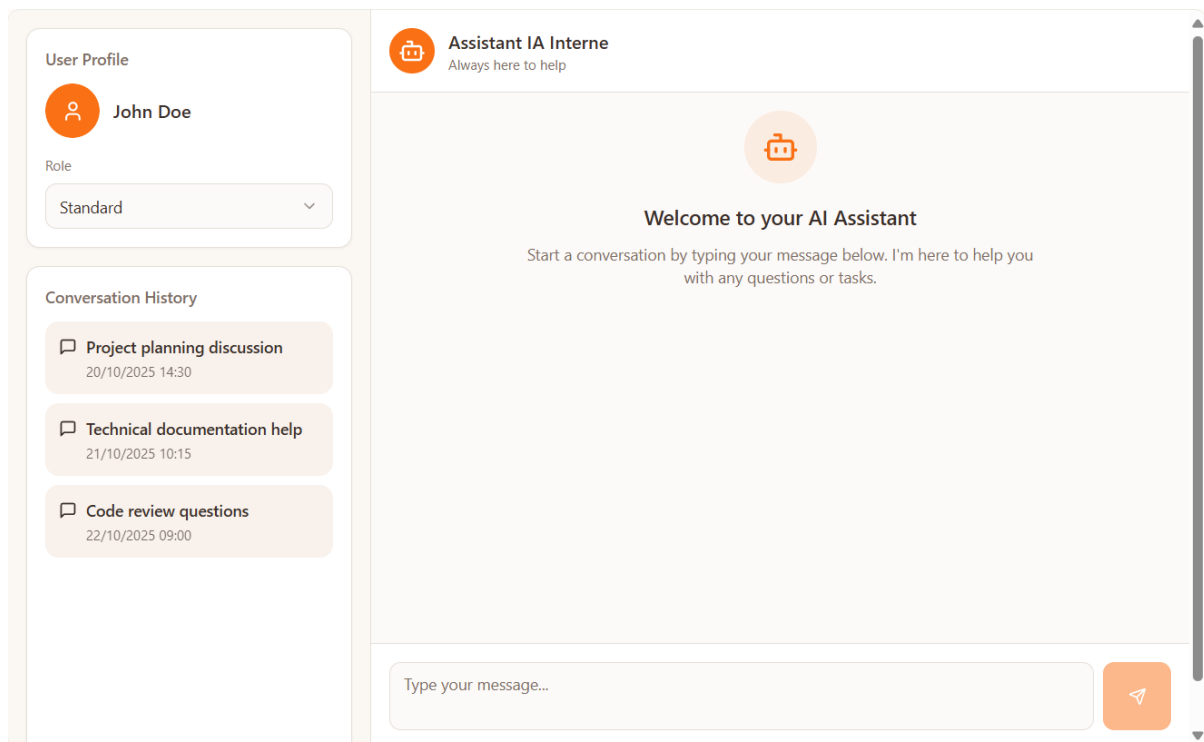
Fonctionnalité	Critères d'acceptation
Chat conversationnel	- L'utilisateur peut saisir une question libre et obtenir une réponse en moins de 3 secondes.

	<ul style="list-style-type: none"> - Au moins 80 % des réponses sont jugées “utiles” par les testeurs.
Historique des conversations	<ul style="list-style-type: none"> - Chaque conversation est sauvegardée et accessible depuis la sidebar. - L'utilisateur peut relancer une conversation antérieure et reprendre le contexte. - L'historique est lié au compte utilisateur.
Feedback utilisateur	<ul style="list-style-type: none"> - Un bouton “utile / non utile” est visible sous chaque réponse du chatbot. - Le clic est enregistré dans la base de données. - L'administrateur peut consulter le score moyen d'utilité.
Accès aux documents internes (RAG)	<ul style="list-style-type: none"> - Lorsqu'une question correspond à un document métier, la réponse inclut la référence du document source. - Le RAG retourne uniquement des contenus issus du corpus interne.

Détail des rôles utilisateurs

Rôle	Description	Droits d'accès
Utilisateur standard	Collaborateur accédant au chatbot pour poser des questions générales.	<ul style="list-style-type: none"> - Consulter et interagir avec le chatbot. - Accéder à son historique. - Donner un feedback sur une réponse.
Expert technique	Collaborateur disposant d'un accès élargi au corpus documentaire.	<ul style="list-style-type: none"> - Accès au chatbot avec données techniques plus détaillées. - Peut signaler une erreur dans le contenu d'un document.
Administrateur	Responsable du corpus et de la maintenance du chatbot.	<ul style="list-style-type: none"> - Gérer les utilisateurs et leurs rôles. - Ajouter / supprimer des documents du corpus. - Consulter les logs et statistiques d'usage.

Interface utilisateur



L'interface utilisateur cherche à être simple et sobre. Il suit le design de la majorité des chatbots connus (ChatGPT, Mistral ...) pour faciliter la compréhension de l'utilisateur. Il est composé d'une sidebar à gauche de l'écran divisée en deux parties : une sur le profil utilisateur et l'autre sur son historique de conversation. La partie principale de l'interface est la zone dédiée à la conversation.

Données utilisées

- Documentation interne (pages internes, périmètre technique)
- Indexation via moteur sémantique

4. Spécifications techniques

Stack technique

Composant	Technologie
Backend	Java Spring Boot (API REST) + Spring AI
Frontend	React

Base de données	PostgreSQL
API IA	Mistral API
Moteur sémantique	Mistral API & pgvector

Détail du choix des technologies

Le choix des technologies retenues pour le projet repose sur plusieurs critères : la maîtrise technique du développeur, la pertinence des outils pour la solution IA, ainsi que la cohérence avec les pratiques actuelles en entreprise.

Backend – Spring Boot et Spring AI

- Le choix du framework Spring Boot s’est imposé car il est maîtrisé par le développeur et facilite la mise en place d’API REST.
- L’intégration du module Spring AI offre une solution simple à mettre en œuvre pour la communication avec le modèle d’intelligence artificielle.

Frontend – React

- Le choix du framework React répond à un objectif d’utilisation d’une technologie moderne et largement utilisée en entreprise.

Base de données – PostgreSQL

- La base de données PostgreSQL a été retenue pour sa capacité à stocker des données structurées liée au chatbot et parce qu’elle est déjà bien maîtrisée par le développeur.

Intelligence artificielle – Mistral

- Le modèle Mistral a été choisi étant l’un des modèles open sources les plus performant et ayant une intégration possible via Spring AI.
- Il présente aussi un atout stratégique étant une solution européenne (française), garantissant une meilleure conformité avec les exigences de souveraineté des données pour une entreprise française.



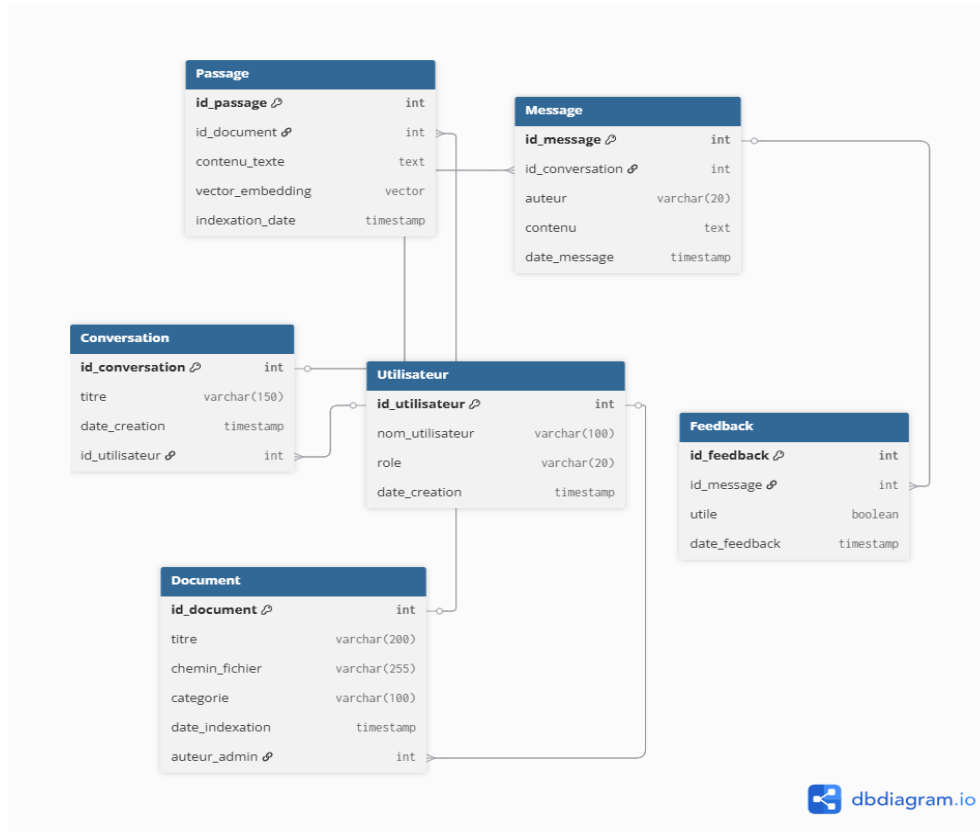
Mistral AI



PostgreSQL

Diagramme BDD

Le modèle logique de données définit la structure de la base de données (PostgreSQL) par le chatbot interne. Il repose sur six entités principales : Utilisateur, Conversation, Message, Document, Passage et Feedback.



Modèle Logique de Données

Chaque utilisateur peut initier plusieurs conversations, elles-mêmes composées de messages échangés entre l'utilisateur et le chatbot. Les documents internes représentent la base de connaissances métier, et sont découpés en passages dont le contenu textuel est vectorisé grâce à l'extension pgvector de PostgreSQL, afin d'alimenter le moteur RAG (Retrieval-Augmented Generation) pour la recherche sémantique.

Enfin, le feedback utilisateur permet d'évaluer la pertinence des réponses générées, favorisant l'amélioration continue du corpus documentaire. Ce modèle assure ainsi une structuration cohérente des données entre les utilisateurs, les échanges conversationnels et la base documentaire technique du chatbot.

Intégration du RAG (Retrieval-Augmented Generation)

Le modèle de langage Mistral ne dispose pas, à lui seul, des connaissances propres aux processus internes et à la documentation métier. Pour contextualiser les réponses du chatbot, l'intégration d'un RAG (Retrieval-Augmented Generation) est donc

essentielle. Cette approche permet d’aller chercher les documents pertinents dans une base documentaire, puis de combiner ces informations avec la question de l’utilisateur afin de générer une réponse précise et cohérente.

Le RAG s’appuie sur la création de vector embeddings, représentations numériques des textes permettant de mesurer leur similarité sémantique. Les vecteurs sont stockés et exploités dans PostgreSQL via l’extension pgvector, qui permet d’effectuer des recherches de similarité efficaces. Sur le plan technique, la solution repose sur trois briques principales : Spring AI comme orchestrateur, Mistral API pour la génération de texte, et PostgreSQL + pgvector pour la gestion et la recherche des embeddings.

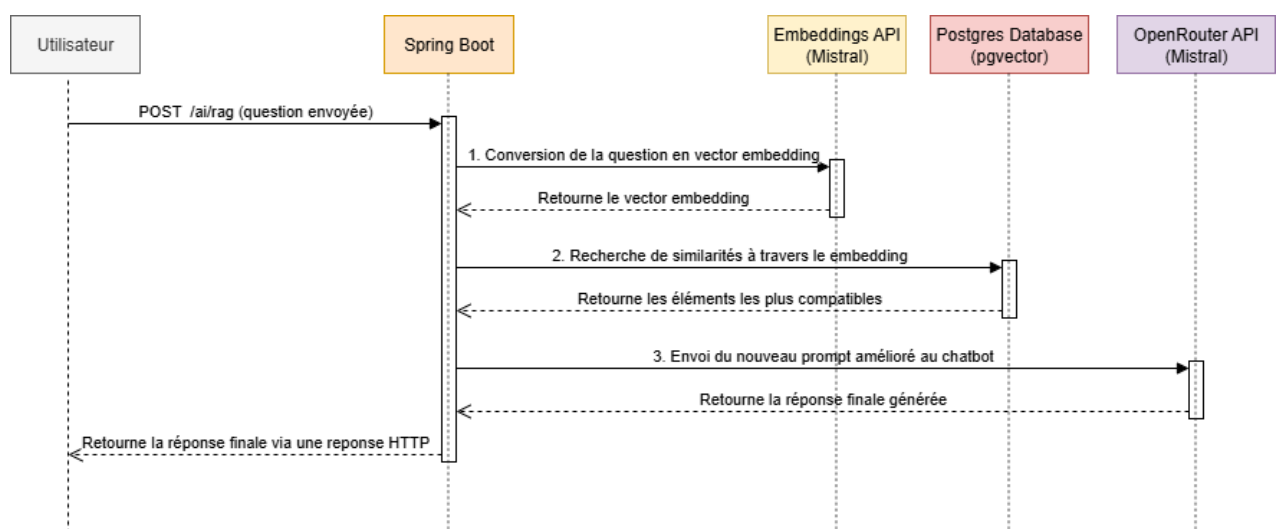


Diagramme de séquence UML – Construction d’une réponse contextuelle

5. KPI de succès

Pour déterminer si le POC est concluant voici des objectifs mesurables qui permettent de déterminer si la solution est faisable et utile.

Objectifs techniques

Objectif	Indicateur / Cible
Intégrer correctement le modèle Mistral via Spring AI	100 % des requêtes traitées sans erreur d’appel API
Implémenter la recherche RAG fonctionnelle	Temps de recherche documentaire < 2 s
Indexation réussie du corpus interne	≥ 90 % des documents chargés sans erreur
Stabilité du prototype	Aucun crash/erreur sur 10 sessions consécutives de test

Objectifs fonctionnels

Objectif	Indicateur / Cible
Fournir des réponses pertinentes et contextualisées	≥ 80 % de réponses jugées “utiles” par les testeurs
Capacité à relancer une conversation antérieure	Fonction testée avec succès dans ≥ 3 scénarios métier

Objectifs de performance et d'adoption

Objectif	Indicateur / Cible
Temps moyen de réponse global	< 3 secondes
Taux de satisfaction global (cf. Questionnaire)	≥ 75 %
Taux de testeurs estimant le chatbot “pertinent pour l'entreprise”	≥ 70 %

6. Confidentialité des données

La mise en place de ce POC s'accompagne d'une attention particulière à la protection des données personnelles et sensibles, conformément au Règlement Général sur la Protection des Données (RGPD) et aux bonnes pratiques de sécurité informatique.

Chiffrement et stockage des données

Les mots de passe des utilisateurs seront stockés de manière sécurisée en base de données à l'aide d'un hachage fort (bcrypt) afin d'éviter toute récupération exploitable en cas de fuite.

Dans le cadre du POC, les échanges front-back s'effectueront en HTTP sur environnement local. Lors d'un futur déploiement en production, la communication sera entièrement sécurisée via HTTPS (TLS) avec certificat valide

Aucune donnée personnelle sensible (nom, email, etc.) ne sera transmise à des services externes ou tiers dans le cadre du POC.

Hébergement du POC

Ce POC ne prévoit aucun hébergement en production. Le déploiement se fera en local, à des fins exclusivement de démonstration et de tests fonctionnels.

En conséquence, aucune donnée réelle de collaborateurs ne sera utilisée : les tests s'effectueront à partir de données fictives ou anonymisées.

Gestion des rôles et authentification

L'accès à l'application est restreint et repose sur une authentification par token JWT (JSON Web Token) gérée via Spring Security.

Trois rôles sont définis :

- **Utilisateur standard** : consultation limitée du chatbot et de son historique.
- **Expert technique** : accès élargi à certaines sources documentaires.
- **Administrateur** : gestion du corpus documentaire et des utilisateurs.

Les permissions sont vérifiées à chaque requête via le token JWT afin d'assurer une séparation stricte des droits d'accès.

La mise en place d'un LDAP pourra appuyer cette sécurité et restriction d'accès suivant le profil interne de l'utilisateur.

Journaux d'activité (logs) et anonymisation

Des logs d'activité seront conservés pour assurer un suivi technique et fonctionnel du POC (erreurs, temps de réponse, interactions).

Ces logs ne contiendront aucune donnée nominative et seront anonymisés : seules des informations techniques (horodatage, identifiant de session, requêtes types) seront enregistrées.

Les journaux seront stockés localement, sans transfert vers des systèmes externes.

Conformité RGPD

Le respect du RGPD est assuré selon les principes suivants :

- **Collecte minimale** : seules les données strictement nécessaires au fonctionnement du POC sont collectées (identifiants techniques et journaux anonymisés).
- **Droits des utilisateurs** : chaque utilisateur peut demander l'accès, la rectification ou la suppression de ses données d'identification.
- **Limitation de conservation** : les données seront supprimées à la fin du POC ou anonymisées pour des analyses techniques futures.
- **Sécurité et confidentialité** : toutes les données stockées localement sont chiffrées et accessibles uniquement par le développeur en charge du POC.

Soutien au numérique responsable

Dans une logique de sobriété numérique, le projet privilégie :

- l'usage de frameworks modernes et optimisés (Spring Boot, React) limitant la charge serveur,
- une architecture réactive favorisant la réduction des traitements inutiles,
- des composants front-end légers et réutilisables, réduisant la consommation énergétique côté client.

7. Stratégie de validation et gestion des branches

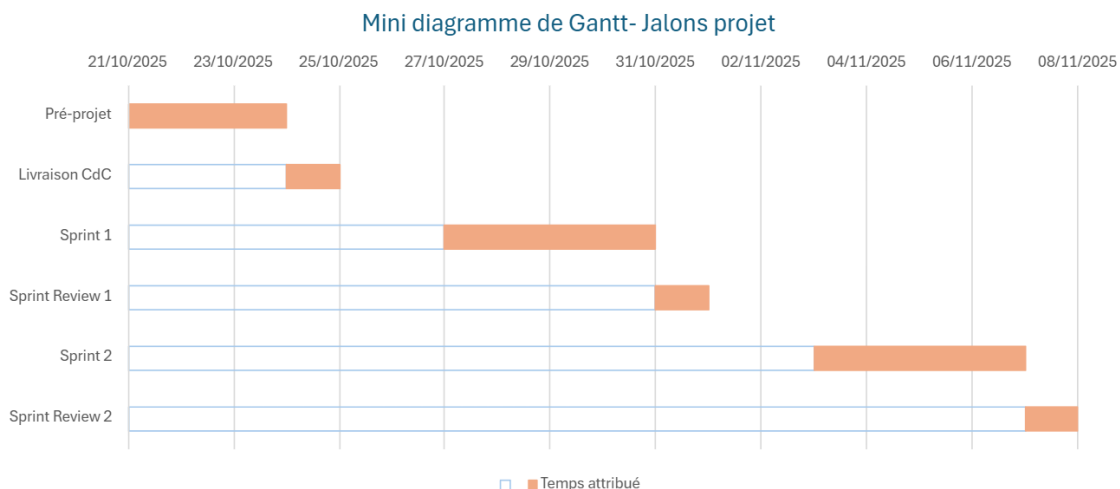
Même dans le cadre d'un POC, une stratégie de validation rigoureuse est mise en place afin d'assurer la qualité et la traçabilité du code. Le projet sera structuré autour de plusieurs branches Git :

- **feature/** : chaque nouvelle fonctionnalité est développée dans une branche dédiée, permettant un travail isolé et un suivi clair des évolutions.
- **dev** : cette branche regroupe l'ensemble des développements en cours. Les fonctionnalités y sont intégrées progressivement pour effectuer des tests manuels basés sur les critères d'acceptation définis dans les user stories.
- **master** : une fois les tests validés, la fonctionnalité est fusionnée dans la branche principale. Celle-ci correspond à la version stable utilisée pour les démonstrations et présentations au client.

Cette organisation favorise une gestion agile du code, tout en préparant la solution à une éventuelle intégration future dans un pipeline CI/CD complet.

8. Jalons & Livrables

Pour ce projet le planning suivi contient 3 jalons : Livraison cahier des charges, Sprint Review 1 et Sprint Review 2.



En termes de documentation, il est prévu de livrer en cette fin de second Sprint : un code source versionné grâce à Git, un accès au backlog via JIRA, une documentation technique et fonctionnelle.

9. Perspectives d'évolution et déploiement futur

Dans le cas où la solution ferait l'objet d'un déploiement en environnement de production, plusieurs améliorations techniques seraient à envisager. Tout d'abord, la mise en place d'un pipeline CI/CD complet (via GitHub Actions) permettrait d'automatiser le processus de build, de test et de déploiement de l'application. L'utilisation du conteneur Docker devient aujourd'hui indispensable dans les entreprises souhaitant déployer des applications de manière agile et reproductible.

Enfin, afin d'assurer la continuité de service et un usage illimité du moteur de génération, il serait nécessaire de souscrire à une offre payante Mistral API. Cette version « production » permettrait d'augmenter le volume de requêtes autorisées, d'obtenir un support technique dédié et d'améliorer la stabilité du service dans un contexte d'utilisation réelle.

Annexe : Questionnaire de satisfaction

Questionnaire évaluant la satisfaction client

Cette évaluation est mise en place dans le but d'évaluer quantitativement l'application suivant les indicateurs suivants.

Indicateur de satisfaction (12pts)

1. Qualité du design : A quel point trouvez-vous l'apparence générale de l'application agréable et professionnelle ?
0 pt ☐ 1 pt ☐ 2 pt ☐ 3 pt ☐
2. Intuitivité : trouvez-vous l'application facile à utiliser et intuitive ?
0 pt ☐ 1 pt ☐ 2 pt ☐ 3 pt ☐
3. Réactivité : Etes-vous satisfait de la rapidité de l'application lors de la navigation et des temps de réponse ?
0 pt ☐ 1 pt ☐ 2 pt ☐ 3 pt ☐
4. Utilité des fonctionnalités : Les fonctionnalités proposées répondent-elles à vos attentes ou besoins ?
0 pt ☐ 1 pt ☐ 2 pt ☐ 3 pt ☐

Indicateur temporel (8pts)

Détail : test d'un scénario de questionnement sur les processus internes via ancien moyen (communication, base documentaire...) puis nouvel outil (chatbot).

Temps de réponse :

- Ancien moyen = T_a =
- ChatBot = T_c =

Notation : $\text{Note} = 4 \left(1 + \frac{(T_a - T_c)}{(T_a + T_c)} \right) =$

Note finale : /20