

---

# Spécifications Fonctionnelles — Système Conversationnel RAG

## 1. Objectif du système

Créer une interface conversationnelle avec :

- Gestion de conversations multi-utilisateurs.
- Intégration dynamique de modèles LLM via configuration.
- Utilisation de corpus RAG pour enrichir les réponses (documents PDF ou notes internes).
- Interface d'administration pour gérer les RAG, les modèles, et les paramètres de conversation.

---

## 2. Acteurs

Acteur	Description
Utilisateur	Interagit avec l'interface de chat.
Utilisateur Gère	Gère les modèles, RAG, notes, et paramètres.

---

## 3. Fonctionnalités principales

### 3.1 Page d'accueil — Liste des conversations

- **Afficher toutes les conversations.**
- **Créer une nouvelle conversation :**
  - Insertion en base.
  - Redirection immédiate vers la conversation.
- **Supprimer une conversation :**
  - Suppression en cascade (messages).
  - Mise à jour immédiate de la liste.
- **Clic sur une conversation :** ouvre l'historique.

### 3.2 Page de gestion des RAG

- **Liste des RAG :**
  - Affiche nom, nombre de documents, date de création.
- **Créer un RAG :**
  1. Sélection d'un dossier local.
  2. Extraction et découpage des PDF.
  3. Génération des embeddings.

4. Stockage dans la base vectorielle.
  5. Insertion dans la table `rag_corpus`.
- **Supprimer un RAG :**
    - Supprime le corpus, les documents, les chunks et embeddings associés.
  - **Mettre à jour un RAG :**
    - Ajouter ou retirer des PDF.

### 3.3 Page de gestion des modèles LLM

- **Créer une configuration :**
  - Choix du provider (OpenAI, Claude, Cohere, Local).
  - Saisie : nom, modèle, clé API, URL, paramètres.

### 3.4 Page de conversation

- **Sélection du modèle LLM** via dropdown.
- **Activation de RAG et Notes** via cases à cocher.
- **Zone de chat :**
  - Sauvegarde du message.
  - Recherche sémantique dans les RAG/Notes activés.
  - Envoi du prompt + contexte au LLM.
  - Affichage + sauvegarde de la réponse.

---

## 4. Fonctionnalité complémentaire : Notes

- Création/suppression de **notes textuelles internes**.
- Les notes peuvent être :
  - Chunkées et vectorisées comme les documents PDF.
  - Activées dans une conversation comme des RAG.
- **Utilisation possible dans les prompts.**

---

## 5. Workflows utilisateur

### ► Création de conversation

1. Clic "Nouvelle conversation".
2. Conversation insérée.
3. Page de chat ouverte.

### ► Création d'un RAG

1. Clic "Créer un RAG".
2. Sélection d'un dossier contenant des PDF.
3. Traitement automatique + confirmation.

## ► Envoi d'un message

1. L'utilisateur tape son message.
  2. Recherche sémantique dans les RAG/notes activés.
  3. Contexte construit + prompt envoyé.
  4. Réponse reçue + affichée.
- 

## 6. Cas d'utilisation (extraits)

Cas	Description
UC1 : Créer une conversation	L'utilisateur démarre une nouvelle discussion.
UC2 : Ajouter un RAG	L'utilisateur ajoute un nouveau corpus de PDF.
UC3 : Activer une note	L'utilisateur sélectionne une note pour la session.
UC4 : Changer de modèle LLM	L'utilisateur bascule entre plusieurs configurations.

---

## 7. Critères d'acceptation (exemples)

Feature	Critère d'acceptation
Création de conversation	Elle s'affiche immédiatement dans la liste.
Recherche RAG	Seuls les corpus cochés sont utilisés.
Réponse LLM	La réponse est cohérente et enregistrée.
Création de note	Une note est immédiatement utilisable comme RAG.

---

## 8. Contraintes techniques

- **Base de données relationnelle** (ex: PostgreSQL, SQLite).
  - **Stockage vectoriel** (ChromaDB, Weaviate, etc.).
  - **Respect du RGPD** si déploiement public.
  - **Extensibilité** : possibilité d'ajouter d'autres types de sources.
-