Template rapport technique – Assistant intelligent de recommandation d'événements culturels

1. Objectifs du projet

- **Contexte** : Puls-Events souhaite proposer un assistant intelligent capable de recommander des événements culturels à partir d'une base de données publique issue de l'API OpenAgenda.

  L'objectif est de faciliter l'accès à l'information événementielle via une interface conversationnelle permettant aux utilisateurs de poser des questions en langage naturel.

- **Problématique** : Les données issues de l'API OpenAgenda sont volumineuses, hétérogènes et difficilement exploitables par simple recherche textuelle.

  Un système basé sur une architecture RAG (Retrieval-Augmented Generation) permet de :
  - rechercher des événements pertinents via similarité sémantique,
  - contextualiser la réponse grâce à une base vectorielle,
  - générer une réponse structurée et synthétique à partir d'un LLM

  Cette approche combine recherche d'information et génération de texte, répondant ainsi aux besoins métier de recommandation intelligente.

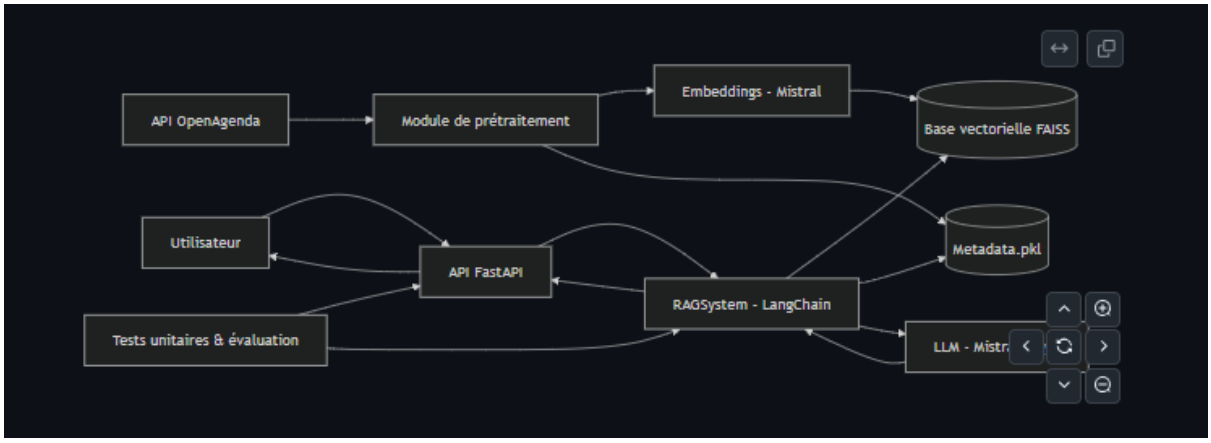
- **Objectif du POC** : Le Proof of Concept vise à démontrer :
  - la faisabilité technique d'un assistant basé sur RAG,
  - la pertinence des recommandations générées,
  - la performance du système (temps de réponse, qualité des réponses),
  - l'intégration dans une architecture API conteneurisée (Docker).

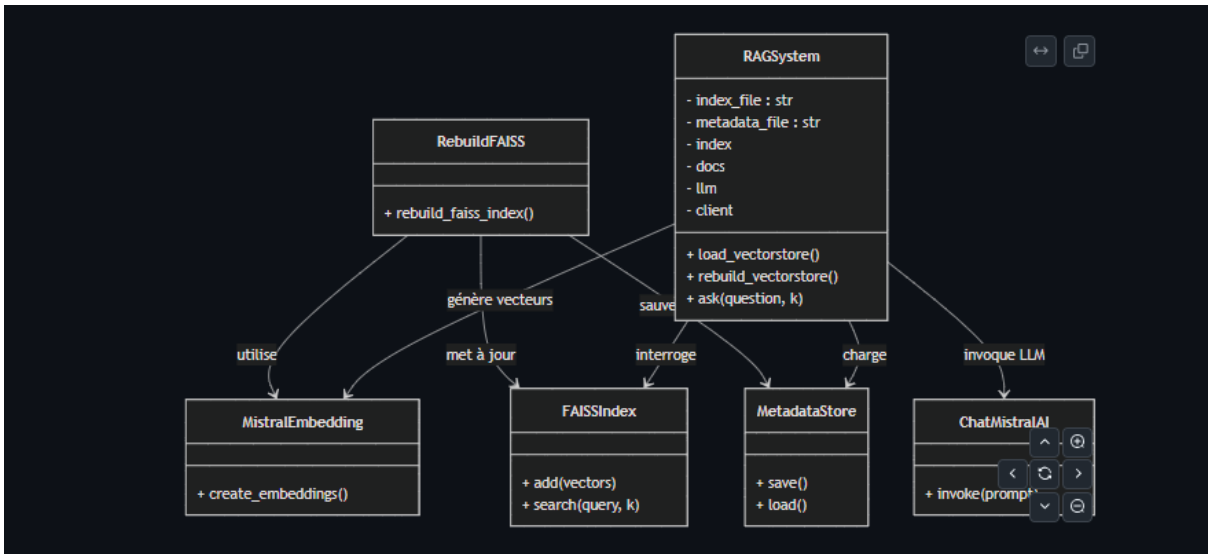
- **Périmètre** :
  - Zone géographique : Département Nord (France)
  - Période : événements des 12 derniers mois

- Source de données : API OpenAgenda
- Langue : Français
- Volume : événements culturels disponibles via API

2. Architecture du système

- **Schéma global** (schéma UML) :





- Données entrantes (API Open Agenda)
- Prétraitement / embeddings / base vectorielle
- Intégration LLM avec LangChain

- Exposition via API

Le système repose sur :

- Extraction des données via l'API OpenAgenda
- Nettoyage et transformation
- Génération d'embeddings via l'API de Mistral AI
- Stockage dans une base vectorielle FAISS
- Interrogation via un modèle LLM
- Exposition via API REST

- **Technologies utilisées** :

- Python 3.12
- FastAPI (API REST)
- LangChain (orchestration LLM)
- FAISS (base vectorielle)
- Mistral AI (embeddings + génération)
- Docker (conteneurisation)
- Uvicorn (serveur ASGI)

3. Préparation et vectorisation des données

- **Source de données** :

  Les données proviennent de l'API OpenAgenda avec les filtres suivants :

  - Département du Nord
  - Période des 12 derniers mois

- **Nettoyage** :
  Les étapes de nettoyage comprennent :
  - suppression des champs vides
  - normalisation des dates
  - suppression des caractères spéciaux
  - concaténation des champs pour créer un texte contextuel riche

- **Chunking** :
  Chaque événement est traité comme un document unique.
  Le chunking n'est pas nécessaire car la taille moyenne d'un événement est inférieure à la limite du modèle.

- **Embedding** :

  - Modèle utilisé : mistral-embed

- Fournisseur : Mistral AI
- Dimension : 1024 dimensions
- Traitement en batch pour optimiser les appels API
- Format : vecteurs float32 compatibles FAISS

4. Choix du modèle NLP

- **Modèle sélectionné** :

  Génération : mistral-large-latest

  Embeddings : mistral-embed

- **Pourquoi ce modèle ?**
  Critères :

  - Bonne compréhension du français
  - Qualité de génération structurée
  - Faible latence
  - Compatible LangChain
  - API stable et documentée

  Ce choix permet un équilibre entre qualité de réponse et performance.

- **Prompting (si utilisé)** :
  Le prompt structure :
  - Contexte récupéré (top-k documents)
  - Question utilisateur
  - Instructions de réponse claire et synthétique

  Exemple :
  Réponds uniquement à partir du contexte fourni.
  Si l'information n'est pas présente, indique que tu ne sais pas.

- **Limites du modèle** :

  - Dépendance API externe
  - Coût lié aux appels LLM

- Possibles hallucinations si contexte insuffisant

5. Construction de la base vectorielle

- **Faiss utilisé** :

  IndexFlatL2 pour recherche par distance euclidienne.

- **Stratégie de persistance** :
  - Index sauvegardé : faiss_index_openagenda.idx
  - Métadonnées : metadata_openagenda.pkl
  - Stockage local dans le conteneur

- **Métadonnées associées** :
  Pour chaque événement :
  - Titre
  - Description
  - Date
  - Lieu
  - keyword
  - catégorie

6. API et endpoints exposés

- **Framework utilisé** : FastAPI

- **Endpoints clés** :

  - `/ask` : question utilisateur, réponse du système
    - Entrée : {
      "question": "Quels événements culturels à Lille ce week-end ?"
      }
    - Sortie : {
      "answer": "...",
      "events": [...]
      }

  - `/rebuild` : Permet de reconstruire l'index FAISS.
    - Entrée : {
      " x-api-key": "clé secrete
      }

        - Sortie : {
          "status": "...",
          "message": [...]
          }
    - `/health` : Vérifie l'état du service et retourne le vrai uptime.
        - Entrée : {
          }
        - Sortie : {
          "status": "...",
          "uptime_seconds": "...",
          "uptime_human": "...",
          }

    - `/Metadata` : Retourne les informations générales sur l'API.
        - Entrée : {
          }
        - Sortie : {
          "name": "...",
          "version": "...",
          "description": "...",
          "status": "...",
          }

- **Exemple d'appel API** : avec `curl`

  curl -X POST http://127.0.0.1:8000/ask \
  -H "Content-Type: application/json" \
  -d '{"question": "Concerts à Lille"}'

- **Tests effectués et documentés**
    - Évaluation automatique via RAGAS (~~Exact Match,~~ Similarité Levenshtein, Coverage)

    - Vérification des scores compris entre 0 et 1

  Les tests sont exécutables via pytest et garantissent la reproductibilité des résultats.

- **Gestion des erreurs / limitations**
    - Validation des entrées
    - Gestion des erreurs API
    - Handler Exceptions

7. Évaluation du système

- **Jeu de test annoté** :
    - 20 questions représentatives
    - Annotation manuelle des réponses attendues
- **Métriques d'évaluation** :
    - Similarité sémantique
    - Pertinence perçue
    - Taux de réponse correcte
- **Résultats obtenus** :
    - Analyse quantitative (scores globaux) :
      test_rag.py::test_similarity_scores PASSED
      [33%]
      test_rag.py::test_coverage_scores PASSED
      [66%]
      test_rag.py::test_answers_not_empty PASSED
      [100%]
    Analyse qualitative (exemples de bonnes/mauvaises réponses)

Bonne réponse :

Question:

Y a-t-il des ateliers pour les enfants à Tourcoing en avril ?,

ground_truth

Atelier Stop-motion | Du noir et blanc à la couleur - 11 April 2025 - Tourcoing

answer

"Atelier Stop-motion | Du noir et blanc à la couleur - 11 April 2025 – Tourcoing "

Mauvaise réponse :

Question :

Expositions à Tourcoing cet automne ?

ground_truth

AUCUN Evenement,

answer

"Exposition Le Cercle des femmes - 01 October 2025 - Hazebrouck *(Note : Hors Tourcoing, exclu)*

8. Recommandations et perspectives

- Ce qui fonctionne bien
    - Architecture modulaire
    - Bonne précision sémantique
    - API facilement déployable
- **Limites du POC** :
    - Dépendance API externe
    - Volume de données limité
    - Pas de filtrage avancé (dates dynamiques)
- **Améliorations possibles** :
    - Ajout d'un filtre temporel dynamique
    - Mise en cache des embeddings
    - Passage à un index FAISS plus performant (HNSW)
    - Déploiement cloud (AWS / Azure)

9. Organisation du dépôt GitHub

- /app :
    - Main.py point d'entrée de l'application (Handler et le routeur)
    - Exceptions.py : gestion centralisée des erreurs
- /rag

    - Rag_system.py : gestion du retrieval + génération
    - Rebuild_faiss.py : reconstruction de l'index vectoriel
- /api
    - endpoints.py : Definition des routes FastAPI
- /tests
    - Contient les tests
- Dockerfile : decrit environnement d'execution
- Requirements.txt : liste des dépendances pythons
- Faiss_index_openagenda.idx : index vectorial généré a partir des embeddings
- Metadata_openagenda.pkl : stockage des metadonnées associées aux vecteur

10. Annexes

- Extraits du jeu de test annoté

  Question :
  Y a-t-il des ateliers pour les enfants à Tourcoing en avril ?
  Gold Answer :

  Atelier Stop-motion | Du noir et blanc à la couleur - 11 April 2025 – Tourcoing

- Prompt utilisé :

  Réponds uniquement avec les événements pertinents du contexte fourni.

  CONTEXTE :
  {context_text}

  QUESTION : {question}

  Chaque événement doit être sur une ligne : Titre - Date - Lieu
  Si aucun événement ne correspond à la demande, répond AUCUN Evenement

  Aujourd'hui : {today.strftime('%d %B %Y')}

  Instructions supplémentaires :

- Si la question mentionne "à venir", inclut uniquement les événements dont la date est >= aujourd'hui.
- Si la question mentionne "il y a", inclut uniquement les événements dont la date est < aujourd'hui
- Ne réécris pas les titres, ne change pas les dates ni les lieux.
- Réponse précise et concise.

Réponse :

- Extraits de logs ou exemples de réponse JSON


```
{
  "answer": "Teen Orchestra - 13 February 2026 - Lille\nSoirée karaoké - 20 February 2026 - Lille\nLettres d'amour - 26 February 2026 - Lille\nConcert : Co-plateau Sheng + 0 Degré - 28 February 2026 - Lille\nConcert des élèves - Stage de musique de chambre - 01 March 2026 - Lille\nSitar connection - 03 March 2026 - Lille\nOFENBACH en concert - 04 March 2026 - Lille\nFemmes, Femmes, Femmes ! - 07 March 2026 - Lille\nLes échos du silence - 07 March 2026 - Lille\nConcert : 2L - 07 March 2026 - Lille\nPVA en concert - 08 March 2026 - Lille\nCOMPLET / Nos 50 ans avec Renaud Capuçon - Orchestre National de Lille - 11 March 2026 - Lille\nCardinals en concert - 19 March 2026 - Lille\nLes Enfants terribles - 20 March 2026 - Lille\nStereolab en concert - 20 March 2026 - Lille\nConcert piano violon Sugita - 20 March 2026 - Lille\nYOA en concert - 21 March 2026 - Lille\nConcert Orchestre d'Harmonie Robert Lannoy - 24 March 2026 - Lille\nBiga*Ranx en concert - 26 March 2026 - Lille\nVisite de l'Opéra de Lille - 28 March 2026 - Lille\nDIDIER BARBELIVIEN - 29 March 2026 - Lille\nconcert : Stony Stone - 02 April 2026 - Lille\nPulse ! - 14 April 2026 - Lille\nConcert : TH - 29 April 2026 - Lille\nArchets et banderilles - 17 May 2026 - Lille\nConcert - 31 May 2026 - Lille",
  "retrieved_events": [
    {
      "title": "Festival du court-métrage",
      "city": "Lille",
      "start": "22 March 2025",
      "end": "22 March 2025"
    },
    {
      "title": "\"Pas à Pas\", concert entre les livres !",
      "city": "Lille",
      "start": "14 May 2025",
      "end": "14 May 2025"
    },
[….]
```