High-Level Design Document

Generative AI Email Classification System

1. Overview

The system processes emails related to Commercial Bank Lending Services (CBLS) and classifies them based on request types (e.g., Fee Payment, Money Movement). It uses a Generative AI model to extract key attributes from emails and attachments. Classified emails are stored in a MongoDB database and assigned to appropriate teams or users.

2. System Architecture

Key Components

1. **Frontend (UI)**
    - Web interface for users to upload emails, view uploaded emails and configure request and sub-request types.
2. **Backend (Flask API)**
    - Handles email uploads, extraction and classification using LLM, storage, and retrieval.
    - API endpoints:
        - /api/email (CRUD operations for emails)
        - /api/request_types (Manage classification request types)
        - /api/email/add_new, /api/email/update_existing (Email processing)
3. **Database (MongoDB)**
    - Stores classified emails, assigned teams, urgency levels, request types, and users.
    - Collections:
        - email_classifications
        - assigned_to
        - assigned_team
        - urgency
        - request_types
4. **AI Processing**
    - **Email Extraction Module** (extract.py)
        - Extracts email content and attachments to structured format -

```
extracted_texts : Array (6)
  0: Object
  1: Object
     type : "email"
     from : "Gregor Ivanov <gregor.ivanov@frontofficebank.com>"
     to : "Anya Petrova <anya.petrova@frontofficebank.com>"
     subject : "Urgent - International Wire Transfer Request"
     time : "Tue, Oct 27, 2024 at 1:58 PM"
     body : "Hi Anya,

            Attached are the completed request form and the supplemental…"
  2: Object
     type : "email"
     from : "Dmitri Kuznetsov <dmitri.kuznetsov@kuznetsovimports.com>"
     to : "Gregor Ivanov <gregor.ivanov@frontofficebank.com>"
     subject : "Urgent Wire Transfer Request"
     time : "Tue, Oct 27, 2024 at 11:23 AM"
     body : "Dear Gregor,

            Following our conversation this morning, I'm sending the…"
  3: Object
     type : "attachment"
     attachment_name : "report_LN-517186.png"
     attachment_content : "Bank of America

                          Client Information Report:
```


- **AI Classification** (analyze_with_llm)
  - Uses a Generative AI model (gemini-2.0-flash) to classify emails based on the context and assigns the teams based on request and sub request types.
  - It also tries to find any other secondary intent the email is about and displays as a note.
- **Document Processing**
  - Saves classified emails, extracted content along with the plain text version of all the email chain and attachment to MongoDB.

---

3. Functional Flow

Step 1: Email Upload

- A user uploads an .eml / .pdf / .docx / .txt email file via the UI (/add).
- The file is read, and its content is extracted using LLM (extract_email_chain_and_attachments).
- Attachments (PDFs, images, DOCX) are processed with OCR/Tesseract.

Step 2: AI-Based Classification

- The extracted email text is passed to analyze_with_llm.
- AI returns a structured JSON with:

- request_type
- sub_request_type
- from, to, subject
- customer_name, urgency, confidence_score
- reason, secondary_intent

Step 3: Duplicate Check

- The system searches for similar emails in MongoDB (find_duplicate).
- If a duplicate exists:
  - The user can classify it as a new email
  - The user can discard it
  - The user can update the existing email chain

Step 4: Team Assignment

- The system assigns a team based on the request type (get_users_for_team).

Step 5: Storage & Retrieval

- Classified emails are stored in MongoDB (email_classifications).
- Users can:
  - **View emails** (/view/<email_id>)
  - **Edit classifications** (/edit/<email_id>)
  - **Manage request types** (/manage_requests)

Step 6: API Operations

- **GET** /api/email?id=<email_id> → Retrieve a classified email.
- **POST** /api/email → Upload a new email.
- **PUT** /api/email → Update an existing classification.
- **DELETE** /api/email?id=<email_id> → Remove an email.