***Team Name**: The Spark*
***Challenge Name**: Gen AI-Based Email Classification and OCR*

Solution Design Document

Prepared by:

- Thiru
- Praveen
- Magdaleen
- Selvasathya

AI Engine Name: Finspark Insights

---

1. Overview

The Email Analyzer is a Streamlit-based AI-powered tool designed to analyze commercial banking emails, categorize them, extract key attributes, and generate structured responses. The solution leverages OpenAI's GPT model for text processing and categorization.

---

2. Models Explored

Models were explored to identify the most efficient AI framework for accurate email categorization, key attribute extraction, and confidence scoring, ensuring optimal performance and reliability.

- LLAMA – downloaded format for llama-2-7b-chat.Q4_K_M.gguf
    - Had initial success with smaller sampled
    - Due to computation limitation on personal system unable to proceed further
- Microsoft/ phi-2
- TinyLlama-1.1B-Chat-v1.0

---

3. Architecture

Components

1. **User Interface**
    - Built using Streamlit.
    - Accepts email content via text input or file upload (Single / Multiple)
    - Output generation, instantly visible on UI and download option in csv format
2. **Preprocessing Layer**
    - Cleans email content by removing redundant spaces and formatting inconsistencies.
    - Extracts an existing Service Request (SR) number, if present.
    - Generates a new SR number if no existing one is found.

- Extracting the attachment content from the email along with key fields (Sender, From, Subject, Body)

3. **AI Model Layer**
    - Uses OpenAI's GPT model (gpt-4o-mini) for analyzing email content.
    - Extracts structured information such as request type, sub-request type, key attributes, intent, and confidence score.
4. **Confidence Scoring Mechanism**
    - Calculates a confidence score based on lexical match, key attribute presence, intent clarity, and ambiguity.
5. **Storage & Processing**
    - Stores input emails and outputs structured data.
    - Processes uploaded .txt, .eml, .msg, and .pdf files.
6. **Configuration & Extensibility**
    - Uses config.json for predefined request types and attributes.
    - Supports future API integrations.

---

4. Workflow

Step 1: Input Handling

- User enters email content manually or uploads a file.
- If a file is uploaded, the system extracts text from it.
- Email content is displayed for review.

Step 2: Email Preprocessing

- Standardizes text format.
- Checks for an existing SR number.
- Generates a new SR number if none is found.

Step 3: AI Processing & Categorization

- Constructs a prompt for OpenAI's GPT model.
- GPT returns a JSON response with:
    - request_type
    - sub_request_type
    - key_attributes
    - main_intent
    - confidence_score
    - confidence_explanation

Step 4: Confidence Scoring

- **Confidence is calculated using weighted parameters:**
    - Lexical match: 25%

    - Key attributes: 30%
    - Intent match: 20%
    - Ambiguity penalty: 25%

Step 5: Output & Display

- JSON output is displayed.
- SR number is generated and shown.
- Confidence score is displayed as a metric.

Step 6: File Processing

- Processes files stored in the input folder.
- Extracts text, analyzes content, and presents results.

---

5. Key Features & Enhancements

- Multi-file support (TXT, EML, MSG, PDF)
- Automated categorization based on AI analysis
- SR number tracking for follow-ups
- Confidence scoring for reliable outputs
- Extensible architecture for API integrations

---

6. Technologies Used

- Programming Language: Python
- Framework: Streamlit
- AI Model: OpenAI GPT-4o-mini
- Data Processing: Regex, JSON
- Storage: Local File System (with potential for API integration)
- Environment Management: dotenv (for API keys)

---

7. Future Enhancements

- Integration with commercial banking systems via API.
- Improved SR tracking with a database backend.
- Enhanced confidence scoring with machine learning models.
- Web-based UI for email management.

---

8. Conclusion

This solution provides a streamlined approach to commercial banking email analysis, automating categorization and data extraction while ensuring accurate SR tracking. The architecture allows for scalability, making it a future-ready tool for banking operations.