**Email Classification and OCR Tool - Documentation**

**Team Name**: Future\_Evangelists  
**Team Members**: Venkata Evani, Srinivasu Choppa, Anil Devarala

**Overview**

The **Email Classification and OCR Tool** is a command-line Python application designed to read and interpret multiple file formats (.eml, .pdf, .docx, .txt). The tool extracts text content, including from email attachments, and uses the Google Gemini model to classify the document content into request types and subtypes relevant to finance operations.

**Key Functionalities**

**1. Interactive CLI Loop**

* The tool runs in a loop, asking users to input a file path.
* Accepts "exit" as input to terminate the program.

**2. File Type Detection**

* Detects file type based on extension:
  + .eml: Email with optional attachments
  + .pdf: PDF documents
  + .docx: Word documents
  + .txt: Plain text files

**3. Content Extraction by Type**

* **TXT**: Opens and reads plain text.
* **DOCX**: Uses python-docx to extract text from Word files.
* **PDF**: Uses PyPDF2 to extract text from all pages.
* **EML**: Uses email.parser.BytesParser to extract the body and parse attachments.

**4. Attachment Handling (EML only)**

* Lists attachments found in .eml files.
* Allows user to:
  + View content of any or all attachments
  + Choose whether to include attachment content in classification

**5. Google Gemini API Integration**

* API key is configured and the Gemini model is invoked via google.generativeai.
* A prompt is constructed combining the email/document content and a fixed header + tail format.

**6. Classification Output**

* The Gemini model returns structured output:
  + Request Type
  + Request Subtype
  + Sender
  + Receiver (To)
  + Date
  + Amount
  + Currency
  + Reasoning

**Architecture Design**

**Please find the architechtural design for this effort. This contains layered components that are used to build this program and various checks thar are implemented.**

**Layered Components**

User Input Handler

File Type Detection

File Type Check

Content Readers:

* 1. PDF Reader (pyPDF2)
  2. DOCX Reader
  3. EML Reader & Attachment Handler

Content Processing & Combining, Prompt Builder for

Gemini and Gemini API Integration

Output with Classification and Reasoning

**Technology Stack**

* **Python 3**
* **Libraries Used**:
  + email, email.parser, os, docx, PyPDF2, extract\_msg
  + google.generativeai (Gemini API)
* **Input**: File path via terminal
* **Output**: Printed classification result

**Future Enhancements**

* GUI (e.g., Tkinter or Streamlit)
* Batch processing support
* Export classification results to CSV
* Add support for .msg or .xlsx files
* Add logging and error reporting

**Usage Example**

$ python GenAI\_Email\_Classify\_and\_OCR.py

Enter the path to your file (.eml, .pdf, .docx, .txt), (type exit to quit): sample.eml

📂 Detected file type: Email File (.eml)

📎 Found 1 attachment(s):

1. invoice.pdf

Enter the number of the attachment to view (or 'all'): 1

Do you want to include this attachment content in the email body? (y/n): y

Classifying the extracted content:

...