Personal Email Intelligent LLM Agent

24/7 Automated Email Manager

Team : Five Guys

1. Overview

Our project aims to develop an intelligent LLM agent that continuously monitors a user's emailbox 24/7. The agent will assess the urgency and importance of every incoming email and act accordingly. Depending on the content, it will automatically archive, reply, or alert the user via WhatsApp/WeChat. Additionally, if an email involves scheduling or meetings, it will automatically integrate with Outlook or Google Calendar. Every day at 8 PM, the agent will generate a summary report detailing the number of emails handled in each category (archive, auto-reply, and alert) and emphasize key events and itinerary arrangements. To enhance the quality of its responses, the agent will also leverage the user's entire historical email database (inbox and sent items) along with notes from OneNote/Notion using RAG (Retrieval Augmented Generation) techniques.

2. Background & Problem Statement

Email remains a primary communication tool for many, yet the volume and variety of emails can overwhelm users. Key challenges include:

- **Email Overload:** Users(especially professor, team leader, sales person) receive a mix of spam, non-critical, and urgent emails, making it difficult to prioritize.
- **Manual Processing:** Existing systems often require manual intervention for sorting, replying, and scheduling.
- **Lack of Contextual Response:** Without historical context, automated replies may miss nuances, leading to ineffective communication.

Our solution addresses these challenges by providing a fully automated, intelligent system that categorizes, responds, and schedules tasks—all while leveraging historical data to enhance response quality.

3. Proposed Solution

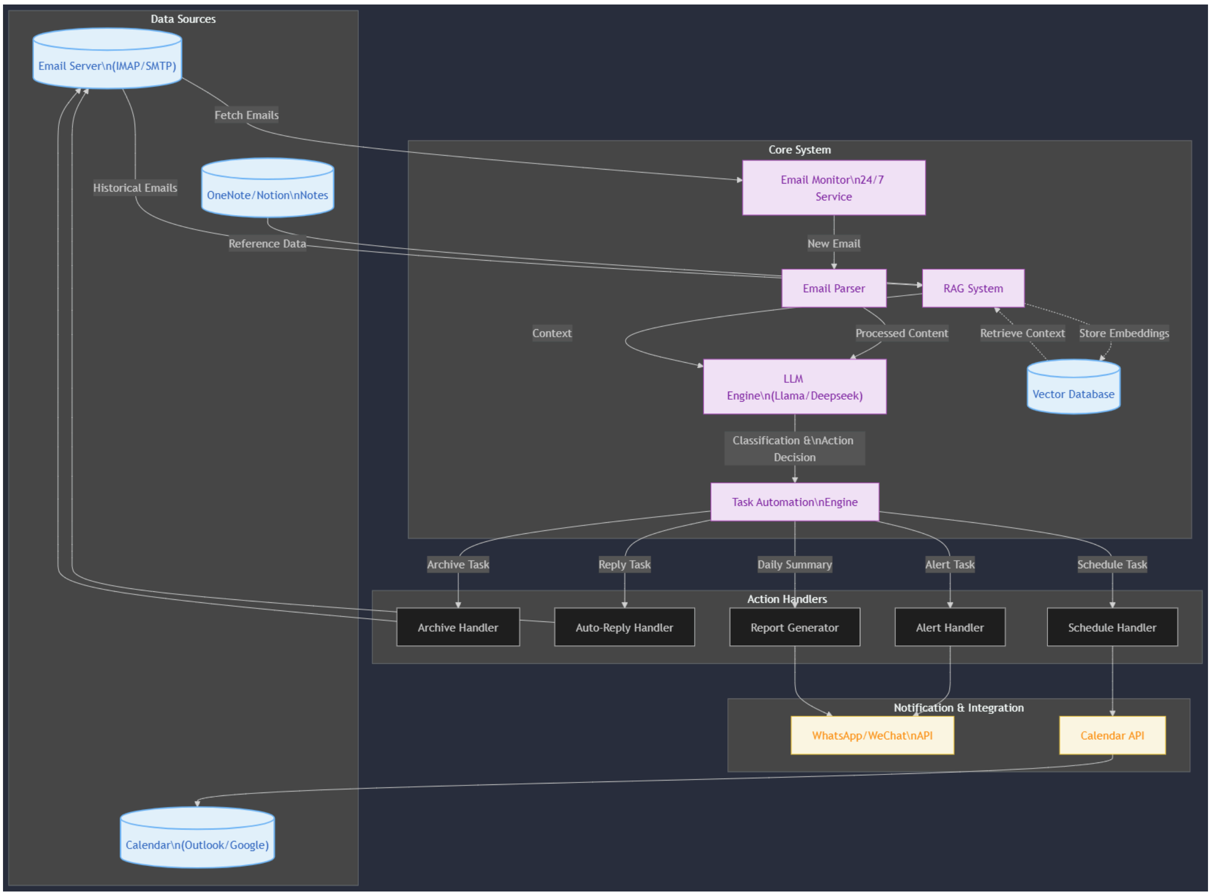
The system will implement the following core functionalities:

1. **Real-Time Email Monitoring & Intelligent Handling:**
    - **a. Archiving:** Automatically archive emails identified as spam or non-important.
    - **b. Auto-Reply:** For emails that can be safely answered automatically, generate a reply and notify the user via WhatsApp/WeChat.

    - **c. Urgent Alerts:** For emails that require immediate human intervention, send an instant notification via WhatsApp/WeChat with a summarized email content and suggested reply.
    - **d. Scheduling Integration:** Automatically create calendar events in Outlook or Google Calendar for emails involving meetings or scheduling.
2. **Daily Summary Report:**
    - At 8 PM every day, the agent compiles a summary of all emails processed that day, detailing the counts for each action (archive, auto-reply, and urgent alerts).
3. **Enhanced Contextual Replies via RAG:**
    - Utilize the user's entire historical email corpus (inbox and sent emails) and notes (from OneNote/Notion) as a reference database.
    - Apply Retrieval Augmented Generation (RAG) techniques to provide the LLM with relevant context, ensuring more accurate and context-aware email replies.

4. Technical Implementation

- **Platform & Environment:**
    - Developed on a Windows platform with cloud integration.
    - We plan to use open-source models from Llama or Deepseek and deploy them locally through LM Studio
    - Utilize standard email protocols (IMAP/SMTP) for real-time monitoring and email operations.
- **LLM & RAG Integration:**
    - Incorporate a Llama or Deepseek model enhanced with RAG to access and process historical email and note data.
    - To use email files as a database with RAG for an LLM, we first parse and convert the emails into structured text, then split them into manageable chunks. Next, generate embeddings for each chunk using a pretrained embedding model and store these vectors in a vector database. During inference, you retrieve the most similar chunks based on the user's query and supply them as context along with the query to the LLM.
    - The RAG module will query the reference database to fetch context, which the LLM uses to generate accurate responses.
- **Task Automation & Scheduling:**
    - Develop a robust automation engine to ensure 24/7 monitoring and to trigger daily summary reports at 8 PM.
    - Integrate APIs for WhatsApp/WeChat for immediate user notifications.
- **Calendar Integration:**
    - Use Outlook and Google Calendar APIs to automatically create events based on email content.
- **Data Security & Privacy:**
    - All user data will be handled in a secure, encrypted environment, ensuring compliance with relevant privacy regulations.



Graph1: system architecture diagram

5. Technology Stack

Our project is primarily developed in Python, leveraging a wide range of libraries and frameworks to ensure efficient development, robust performance, and scalable deployment. The key components of our technology stack include:

Programming Language:

- **Python 3.11:** The core language used for model development, service integration, and automation tasks.

Libraries & Frameworks:

- **Flask/FastAPI:** For building RESTful APIs and backend services.
- **Celery:** To manage background tasks and asynchronous processing.
- **IMAPlib & SMTPlib:** For email retrieval and sending functionalities.
- **Transformers / Sentence-Transformers:** For natural language processing, generating embeddings, and LLM integration.
- **Pandas / NumPy:** For data manipulation and processing.

Model & AI Frameworks:

- **PyTorch / TensorFlow:** For deploying, fine-tuning, and running our LLM models.

- **LM Studio:** For containerized deployment and scalable model serving, specifically for our Deekseek model.

DevOps & Deployment:

- **Docker:** For containerizing our Python applications and models.
- **Kubernetes:** For orchestrating containers, ensuring high availability and auto-scaling.
- **LM Studio's Deployment Services:** For seamless integration, monitoring, and management of deployed models.
- **CI/CD Pipelines (e.g., GitHub Actions):** For continuous integration, automated testing, and deployment.

Monitoring & Logging:

- **Prometheus & Grafana:** For real-time monitoring and performance metrics.
- **ELK/EFK Stack:** For centralized logging and analysis.

Integration APIs:

- **WhatsApp/WeChat APIs:** For sending notifications and alerts.
- **Outlook/Google Calendar APIs:** For automatic calendar event creation based on email content.

6. Project Plan & Timeline (48-Hour)

Phase 1 (0–12 hours):

- Set up LM Studio and large language model environment on Qualcomm AI laptop
- Finalize requirements and design the system architecture.
- Set up the email monitoring framework and integrate essential APIs.
- Configure the LLM and RAG modules with sample historical data.

Phase 2 (12–30 hours):

- Develop the core email processing modules:
  - Automatic archiving, auto-reply generation, and urgent alert notification via WhatsApp/WeChat.
  - Integrate calendar creation functionality.
- Begin initial testing of individual modules.

Phase 3 (30–42 hours):

- Integrate the daily summary report generation module.
- Perform end-to-end testing, ensuring seamless communication between modules and overall system stability.
- Optimize performance and fine-tune the RAG integration for context-aware replies.

Phase 4 (42–48 hours):

- Final testing and bug fixing.
- Prepare comprehensive documentation and a demo video.
- Final review and submission of the project.

7. Team Members

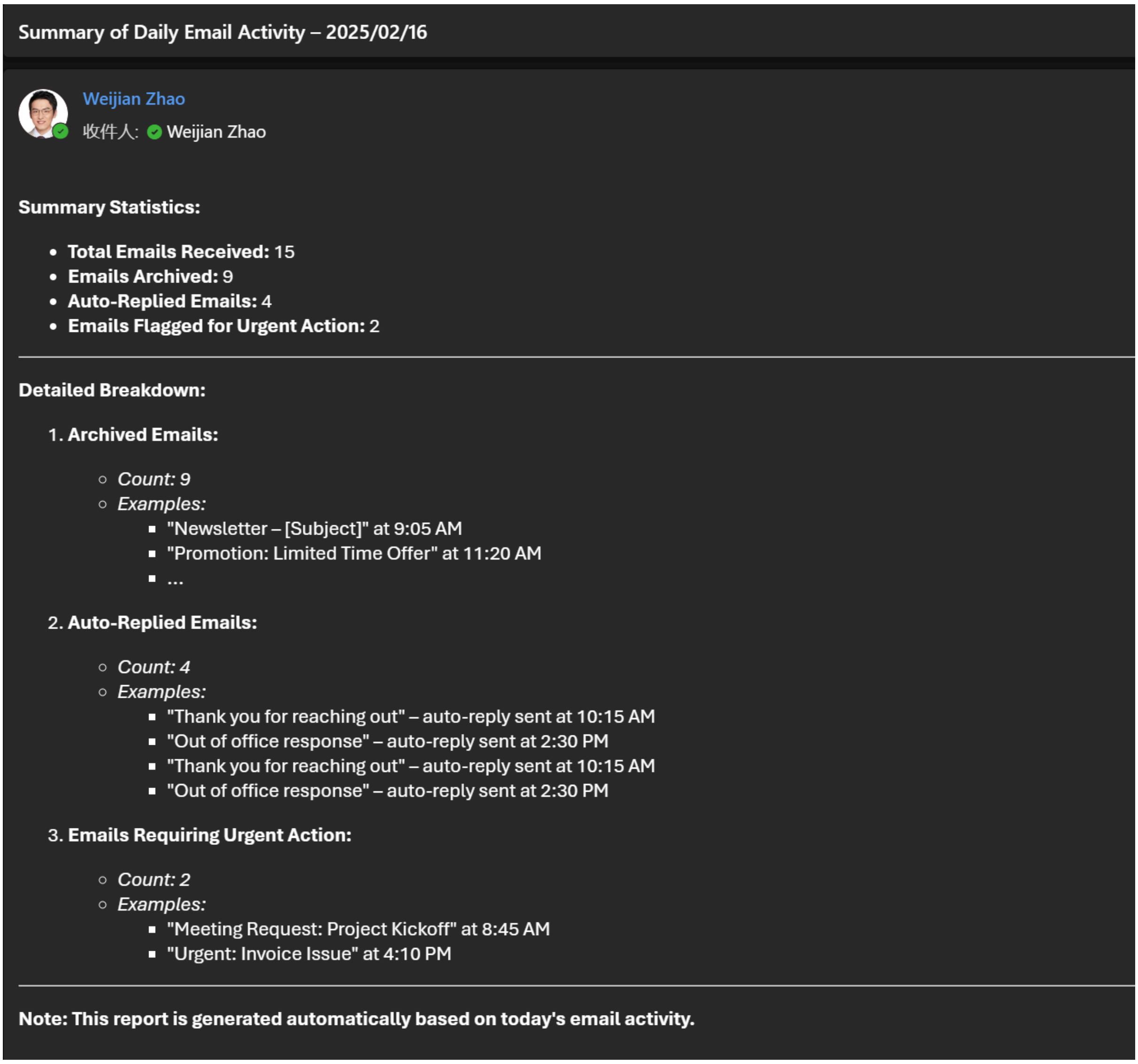
1. **Dayu Wang**
    - **Role:** Data Embedding, including Email parsing & structuring and embedding generation & vector storage
    - **Experience:** Python and text processing libraries, application of pretrained embedding models for NLP
2. **Erli Zhou (Francis)**
    - **Role:** Backend development support and implementation of NLP components
    - **Experience:** Python with a focus on text processing and NLP applications.
3. **Gengyuan Zhang**
    - **Role**：Optimized LLM inference by implementing RAG
    - **Experience:**Web Scraping, Data Processing and Machine Learning
4. **Weijian Zhao (David)**
    - **Role:** project design and project management, LLM implement and RAG database build up
    - **Experience:** Extensive background in AI product design and manager.
5. **Xuetao Li (Lionel)**
    - **Role**:Designing and implementing intuitive, responsive user interfaces and ensuring smooth integration with backend services.
    - **Experience:**Proficient in HTML, CSS, JavaScript, and modern frameworks

8. Expected Outcomes & Future Prospects

- **Expected Outcomes:**
    - A fully operational automated email management system capable of real-time email categorization, auto-reply, urgent notifications, and automatic calendar scheduling.
    - A daily summary report that provides insights into email handling performance.
    - Enhanced email reply quality through RAG-enhanced contextual understanding.
- **Future Prospects:**
    - Expansion to support additional communication platforms (Whatsapp/Facebook/Tiktok).
    - Continuous LLM model and RAG refinement to improve accuracy and efficiency.
    - Potential adaptation for enterprise-level email management and automated office workflows.

9. Prototype

1. Daily Summary Report



Summary of Daily Email Activity – 2025/02/16

Weijian Zhao

收件人: Weijian Zhao

Summary Statistics:

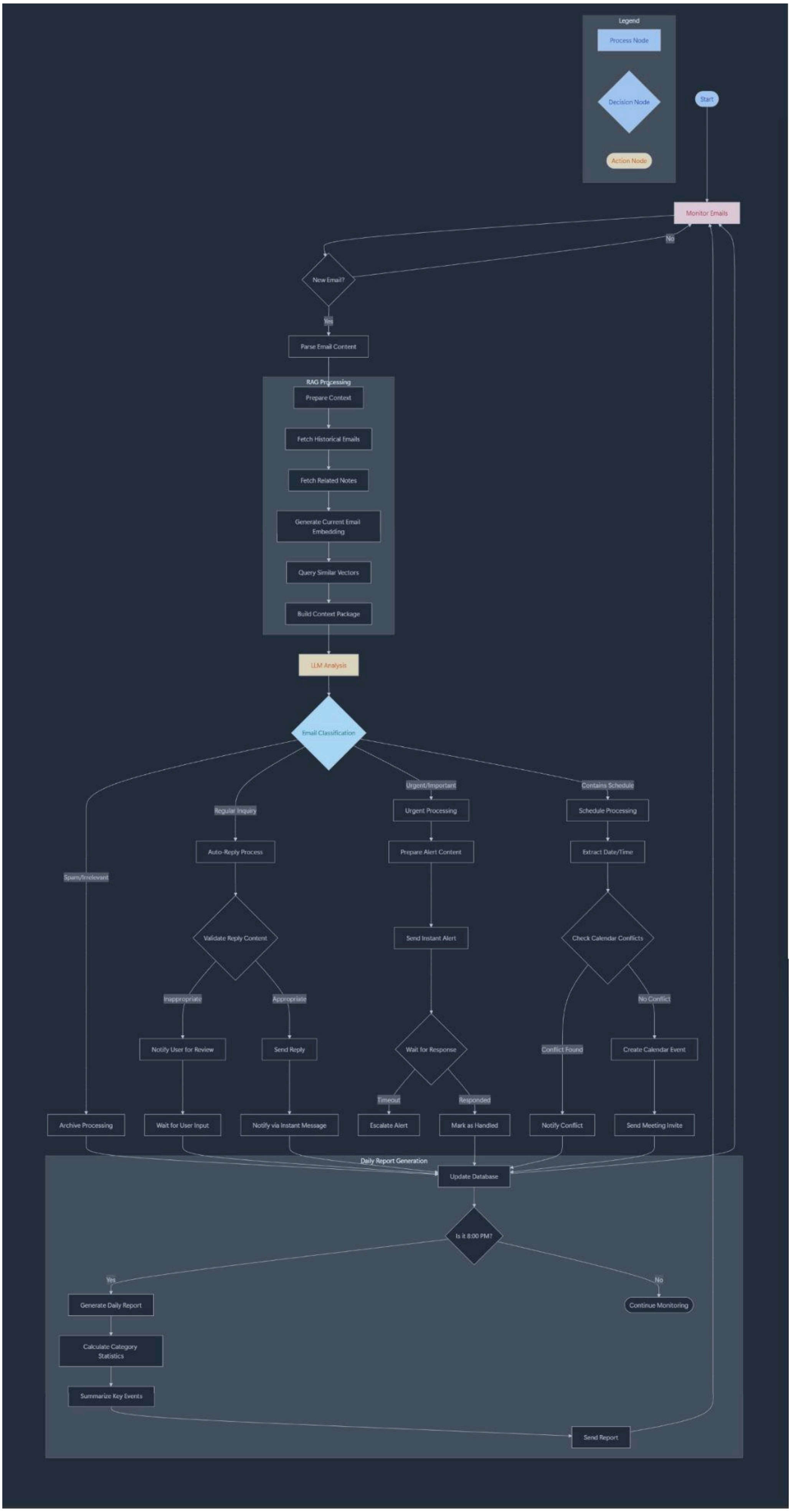
- **Total Emails Received:** 15
- **Emails Archived:** 9
- **Auto-Replied Emails:** 4
- **Emails Flagged for Urgent Action:** 2

Detailed Breakdown:

1. **Archived Emails:**
    - *Count: 9*
    - *Examples:*
        - "Newsletter – [Subject]" at 9:05 AM
        - "Promotion: Limited Time Offer" at 11:20 AM
        - ...
2. **Auto-Replied Emails:**
    - *Count: 4*
    - *Examples:*
        - "Thank you for reaching out" – auto-reply sent at 10:15 AM
        - "Out of office response" – auto-reply sent at 2:30 PM
        - "Thank you for reaching out" – auto-reply sent at 10:15 AM
        - "Out of office response" – auto-reply sent at 2:30 PM
3. **Emails Requiring Urgent Action:**
    - *Count: 2*
    - *Examples:*
        - "Meeting Request: Project Kickoff" at 8:45 AM
        - "Urgent: Invoice Issue" at 4:10 PM

Note: This report is generated automatically based on today's email activity.

10. Appendix Flowchart


Legend
Process Node
Decision Node
Action Node
Start
Monitor Emails
No
New Email?
Yes
Parse Email Content
RAG Processing
Prepare Context
Fetch Historical Emails
Fetch Related Notes
Generate Current Email Embedding
Query Similar Vectors
Build Context Package
LLM Analysis
Email Classification
Urgent/Important
Contains Schedule
Regular Inquiry
Urgent Processing
Schedule Processing
Auto-Reply Process
Prepare Alert Content
Extract Date/Time
Spam/Irrelevant
Validate Reply Content
Send Instant Alert
Check Calendar Conflicts
Inappropriate
Appropriate
No Conflict
Notify User for Review
Send Reply
Wait for Response
Conflict Found
Create Calendar Event
Timeout
Responded
Archive Processing
Wait for User Input
Notify via Instant Message
Escalate Alert
Mark as Handled
Notify Conflict
Send Meeting Invite
Daily Report Generation
Update Database
Is it 8:00 PM?
Yes
No
Generate Daily Report
Continue Monitoring
Calculate Category Statistics
Summarize Key Events
Send Report