Automated Personal Assistant for Educators Using n8n, Docker, and Ngrok

Nang Ying Lao Hsaing
dept.of Applied Digital Technology, Software Engineering

Mae Fah Luang University

ChiangRai, Thailand
6531503162

Nang Pyae Pyae Phyoe Aein
dept. of Applied Digital Technology, Software Engineering

Mae Fah Luang University

ChiangRai, Thailand
6531503161

Sai San Mine

dept.of Applied Digital Technology, Software Engineering

Mae Fah Luang University

ChiangRai, Thailand
6531503177

***Abstract*—** This paper proposes an automated personal assistant system designed to support educators by managing repetitive administrative tasks such as email summarization, signature-required document handling, and meeting scheduling. The assistant is developed using the open-source workflow automation tool **n8n**, deployed with **Docker**, and exposed via **ngrok** for webhook interactions and OAuth handling. Three main features are implemented: (1) scheduled daily email summarization delivered to Telegram, (2) document signature detection with automated signing based on approval, and (3) meeting invite processing that checks calendar availability before scheduling. This system reduces the workload for professors and teachers, allowing them to focus on core academic responsibilities.

***Keywords*—** n8n, Docker, ngrok, Workflow Automation, Email Summarization, Document Signature, Google Calendar, Telegram Bot.

I. Introduction

Educators handle large volumes of emails daily, many containing meeting invites or documents requiring their signature. This workload disrupts focus on academic tasks and introduces delays in responses. Manual processing of these tasks can lead to missed opportunities, inefficient scheduling, and administrative overhead.

Although tools like **Google Workspace** provide APIs for email and calendar management, they lack tailored automation for specific educator workflows. To address these challenges, we designed a personal assistant workflow using **n8n**, a visual workflow automation tool, containerized with **Docker** for consistent deployment, and integrated with **ngrok** for webhook and OAuth redirection. This assistant performs three key functions, each designed to minimize manual effort while keeping user control through approval-based decision points.

II. System Architecture

The system architecture follows a modular workflow approach. The **Main Workflow** handles the orchestration logic, while three **Child Workflows** manage specific tasks. The deployment uses Docker containers, ensuring environment consistency, and utilizes ngrok for external webhook accessibility.

A. Main Workflow

The main workflow is responsible for detecting new emails via the Gmail Trigger node and analyzing their content using AI-powered parsing. When a meeting invitation is detected, the main workflow calls the **Check Event Flow** (Child Workflow 3) to verify calendar availability.

B. Child Workflows

Child Workflow 1: Gmail Summary of 24 Hours:

Summarizes emails from the last 24 hours using an AI summarization model (Ollama) and sends the summary via Telegram. Filtering rules are dynamically loaded from Google Sheets.

Child Workflow 2: Attachment Signature Detection and Automation:

Detects attachments that require signatures using keyword analysis. Upon user approval via Telegram, the assistant applies for a digital signature or rejects the request with a polite message.

Child Workflow 3: Check Event Flow:

Extracts proposed meeting times from the email content and checks Google Calendar for scheduling conflicts. The assistant returns availability status to the main workflow for approval and scheduling.

III. Implementation Details

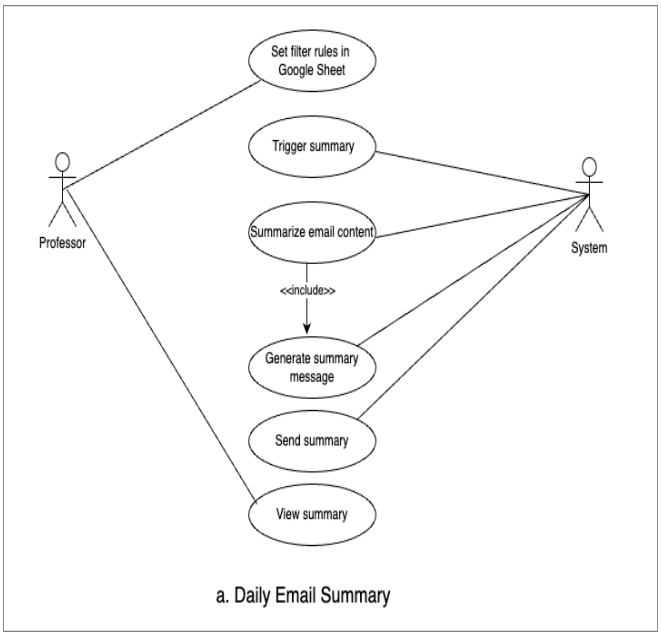
The system uses n8n's visual workflow editor to design each automation step. Communication between the main and child workflows is handled via the **tool Workflow** node. AI tasks such as summarization and meeting time extraction are performed using the **Ollama Chat Model**.

Key integrations include:

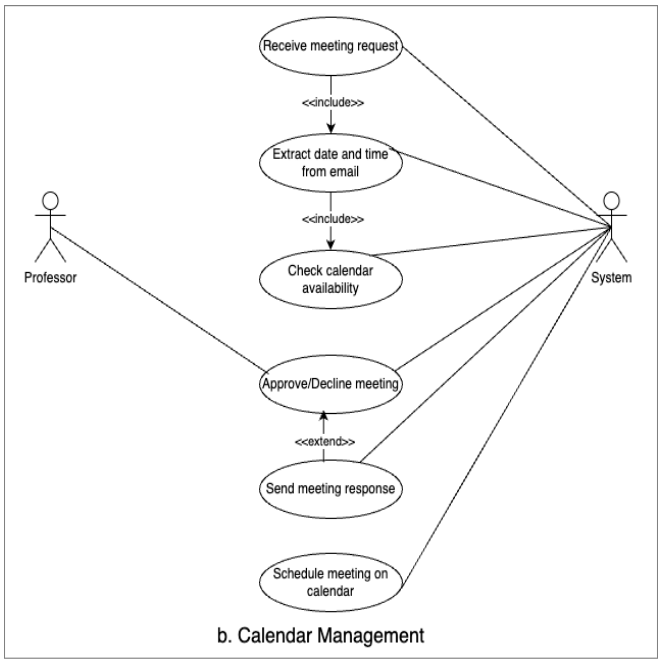
1. **Gmail API** for email retrieval
2. **Google Sheets API** for filter rules
3. **Telegram Bot API** for user interaction
4. **Google Calendar API** for event creation and conflict checking.

All services are secured using OAuth 2.0, and ngrok exposes the localhost webhook URLs required for these authentications.

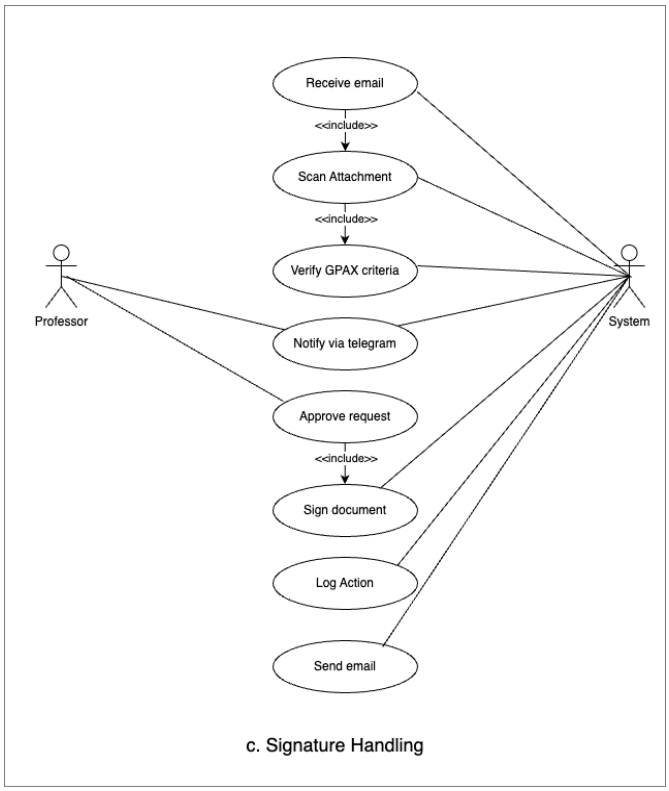
IV. Use Case Diagrams



a. Daily Email Summary Use Case



b. Calendar Management Use Case



c. Signature Handling Use Case

V. Technical Challenges and Solutions

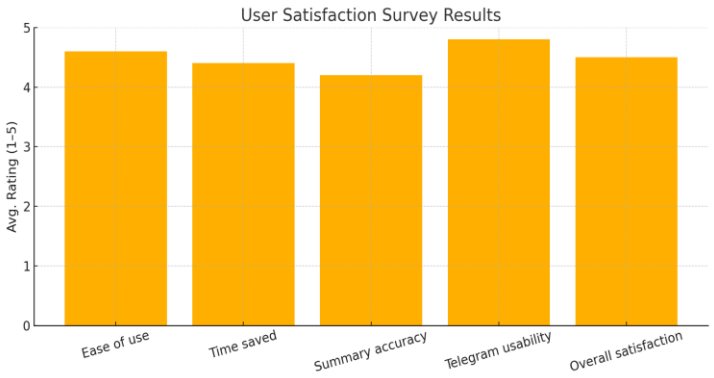
The development of the personal assistant system presented several technical challenges. Managing OAuth callbacks and webhook routing for Gmail and Google Calendar was addressed using **ngrok** for secure URL exposure. To enable flexible email filtering, **Google Sheets** was integrated as a dynamic rule source, allowing easy adjustments without modifying the workflows. Detecting signature-required documents involved PDF text extraction combined with keyword scanning and AI-based analysis to ensure accurate

detection. Coordinating between the main workflow and child workflows required reliable orchestration, which was achieved using n8n's **tool Workflow** nodes. These solutions ensured effective automation while maintaining user control and decision points throughout the process.

VI. User Satisfaction Evolution

Category	**Avg. Rating (1–5)**
Ease of use	4.6
Time saved	4.4
Summary accuracy	4.2
Telegram usability	4.8
Overall satisfaction	4.5

Participants appreciated the speed of daily summaries, clarity of Telegram prompts, and ability to approve without switching apps. The assistant helped reduce their screen time and manual effort.



d. User Satisfaction Graph

VII. Conclusion

The proposed personal assistant successfully automates repetitive tasks for educators, improving productivity and reducing response times for administrative requests. Through the integration of n8n workflows, Docker deployment, and external API services, the system ensures flexibility and scalability for future expansions.

Future improvements may include:

- Adding support for Slack alongside Telegram.
- Implementing natural language command processing.
- Expanding integrations with LMS platforms such as Moodle or Microsoft Teams.