AI Orchestrator for Email and Document Triage & Routing

Overview

The AI orchestrator is designed to process emails and documents, classify their content, and create service requests with key metadata. The solution ensures efficient triage and routing for enhanced workflow automation.

Solution Architecture

We are using Spring boot and Java libraries to solve this problem statement

The architecture consists of the following components:

1. Email Integration

- **Technologies**: JavaMail, POP3
- **Functionality**:
    - Fetches emails from the email server.
    - Supports email parsing for both plain text and HTML content.
    - Retrieves attachments (e.g., documents, images).

2. Content Analysis

- **Technologies**: Apache Tika, OCR (Tesseract)
- **Functionality**:
    - Extracts meaningful text content from email bodies and attachments.
    - Processes images within attachments using OCR to extract text.
    - Handles multiple file types (PDF, DOCX, images).

3. AI Classification

- **Technologies**: Ollama with Phi4 Mini
- **Functionality**:
    - Uses AI models to classify emails into request types and subtypes.
    - Leverages Natural Language Processing (NLP) for text comprehension and categorization.

4. Service Request Management

- **Technologies**: Spring Boot, MySQL
- **Functionality**:
    - Generates service requests with the following details:
        - Sender information.

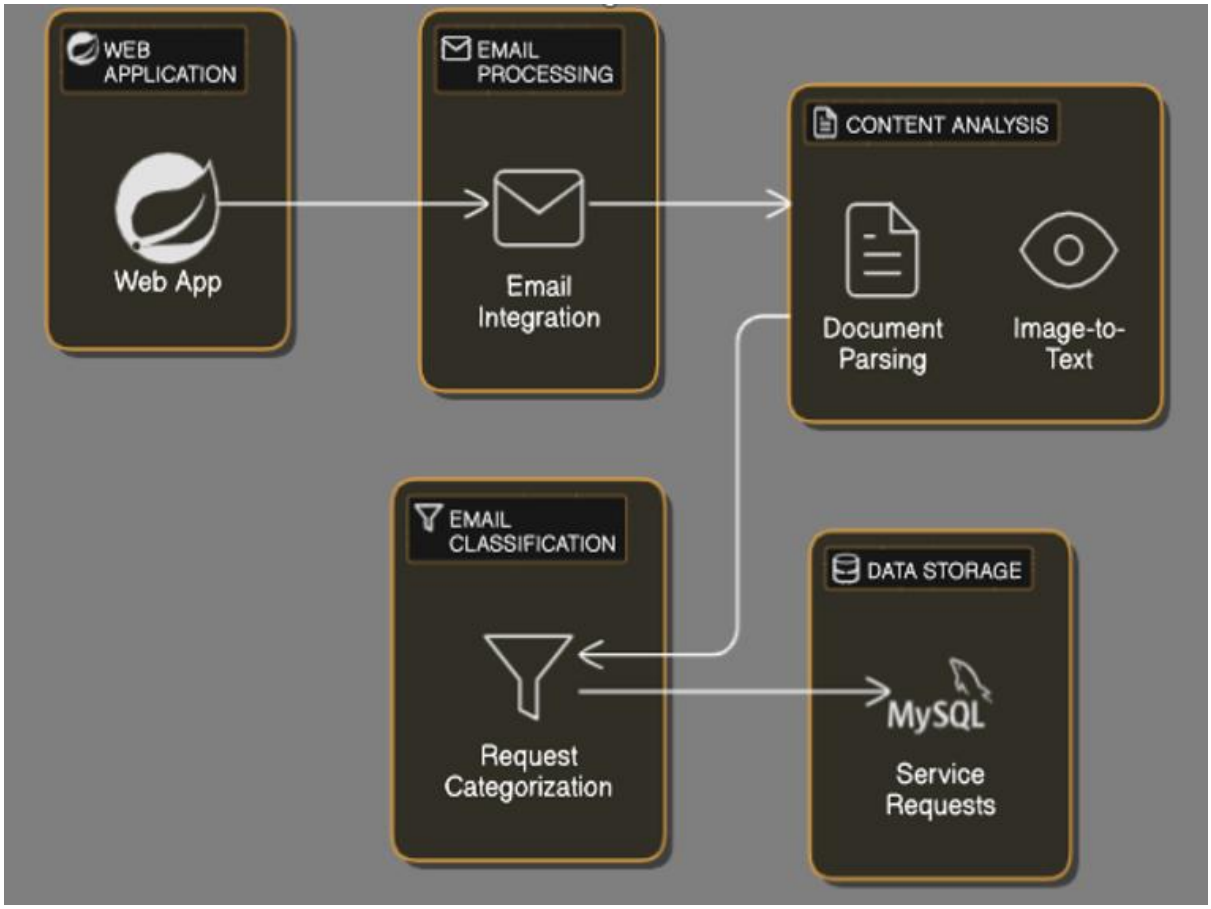
        - Extracted and classified content.
        - Request type and subtype.
    - Stores service requests in a relational database for further action.

5. Web Application

- **Technologies**: Java Spring Boot (RESTful services)
- **Functionality**:
    - Provides a front-end interface for tracking service requests.
    - Exposes APIs for external integrations.

Architecture Diagram

Below is the high-level solution architecture:



Description of the diagram:

1. **Email Server**: Source of incoming emails.
2. **Email Fetcher (JavaMail)**: Retrieves emails using POP3/IMAP.
3. **Content Extractor (Apache Tika)**: Analyzes email bodies and attachments.
4. **AI Classifier (Ollama)**: Processes content to identify request type/subtype.

5. **Database**: Stores service request details.
6. **Web Application**: Enables users to view, manage, and query service requests.

Sequence of Operations

1. **Step 1: Email Fetching**
    - The JavaMail API fetches incoming emails and their attachments.
2. **Step 2: Content Extraction**
    - Apache Tika parses the email body and attachments.
    - OCR extracts text from image attachments.
3. **Step 3: AI Classification**
    - Ollama (with Phi4 Mini) processes the content to classify it into predefined categories.
4. **Step 4: Service Request Creation**
    - Extracted information (sender details, request type, subtype, content) is stored as a service request.
5. **Step 5: User Interaction**
    - Users interact with the web application to view and manage service requests.

Benefits of the Solution

- Automated email triage and classification.
- Efficient handling of large volumes of emails and documents.
- Reduced manual intervention, saving time and effort.
- Improved routing accuracy and faster resolution.
