Email classification system for Service Providers (SP) to improve and simplify the email communication with customers, suppliers, authorities, etc.

Introduction

They project should have several development steps:

1. Classify email (category) and provide statistics about how many emails exists per category, per day. It should be tracked, if emails are already manually replied/processed etc.
   Classify emails priority: NORMAL, HIGH
   If its a customers issue, determine, if its an already known customer or a new customer. For this an access to the customers database is needed. Check, how this can be done more or less generic (e.g. access an exported customers csv file, which is provided by SP)

2. Define event tickets for each event and try to determine, if a new event is generated or if this specific email belongs to an already existing event/ticket. Emails should be assigned to an event/ticket. Provide a list of all events and their duration.

3. Recognising and summarising the processing status of an event. The AI-LLM component should track the email threads and give a brief summary of what happened up to now, how long the current process took already etc.
   E.g.: A client sent a complaint email, SP replies and ask some questions about it, Client replies again, specifies the problem etc.

4. Implement an automation feature. Email should be partly processed (replied) automatically based on there category, i.e. the system will generate drafts, which will be approved manually, i.e. user is able to edit this drafts, reject the draft, or sent the draft as it is.

5. Fully automation. Admin configures/defines some procedures, which can be handled fully automated by the system. System will generate brief summaries of what is happened.

Main functional requirements Step 1:

ID	Description	
01	Fetch Emails from IMAP mailboxes (user could have several email addresses) and pass them to the classifier module (ID 02). Check: 1. How are emails unique identified, do they have IDs? If not, how can we identify them to reference them (e.g. to a database table which holds the statistics) 2. How to handle email threads? 3. Is it possible to assign category labels or tags to mails of the IMAP mailbox, so this could be visualized in or from the email client which is used by the user (e.g. outlook, Thunderbird, etc.) If yes, does this makes sense? Or is it better to visualize these informations by a separate dashboard and consider the IMAP inbox just as read-only?	
02	Classify each emails using AI-LLM technology: 1. categories should be configurable, i.e. read from config file. Category tags/labels looks like: CUSTOMER_REQUEST, CUSTOMER_COMPLAINT, CUSTOMER_COMPLAINT, … 2. The LLM systemprompt does the main task, should be read from config file (see python prototype) 3. Classify priority NORMAL, HIGH 4. Generate a brief summary of the content, and extract some important keywords (or key informations) from the email like order numbers, invoice numbers, customer numbers, etc. Use an open source LLM (e.g. gemini or llama3.1 or newer) and local LLM (ollama) or Groq-API.	

ID	Description	
03	Provide a statistic, how many emails per category exists, how many are unprocessed (unread, read, replied, etc.), sorted by date/time categories. E.g. today, last 2/3/4-6 days, last week…	
04	If its a customers issue, determine, if its an already known customer or a new customer. For this an an exported customers csv file is used, which must be provided by SP with a fixed format: ID, name, address, email, …	
06	TODO: define, how results (statistics) could be stored and tracked etc. and how this data could be visualised using a dash board.	
07	TODO	

No functional requirements (TODO):

- Check: Which software development architecture is needed? Python should be used as programming language, but front end should be a pure web application. Should be able to integrate into the SP evidio dashboard somehow.
  Can we run python on the web server? Do we need a different python application server and define a REST-API to communicate with the web server?
- Which tools/libs should be used?