

# Email Order Processing Automation RPA

## TEAM MEMBERS

SAKKA Mohamad Mario

MAHMOUD MIRGHANI Abdelrahman

ZAFAR Azzam

AL KHALIDY Essam

## 1 EXECUTIVE SUMMARY

We designed and implemented an automation that processes online orders received by email and updates a local inventory restock spreadsheet. We built the solution with UiPath Studio Community and we integrated it with Outlook Classic and Microsoft Excel on Windows 11.

We improved the current manual workflow where staff read emails, extract product lines, download attachments, update Excel, and reply with an estimated delivery date. Our solution reads a configured inbox, detects order emails using subject keywords, parses order lines in three supported formats, computes totals and delivery estimates, saves attachments into structured folders, updates the restock spreadsheet, sends a confirmation email, moves the email to an outcome folder, and appends an audit record to a CSV log.

We designed the system with a clear boundary and configuration driven behavior. We used a work breakdown structure, a schedule plan, risk management, and a quality assurance plan with measurable metrics for extraction accuracy and NeedInfo rate. We also modeled the system using SysML structure and behavior diagrams and we traced requirements to the main components.

## 2 PROBLEM DEFINITION AND SCOPE

Many companies receive a mix of B2B and retail orders by email. Even when the order content is simple, the processing steps are repetitive. A staff member must identify order emails, copy the order lines into an internal file, download and store attachments, calculate totals, decide a delivery estimate, reply to the customer, and keep the inbox organized. This process consumes time and introduces avoidable errors.

Our project improves this existing process by automating the end to end email order processing workflow. We focus on orders that arrive in an Outlook mailbox and we update a local restock spreadsheet as the internal system of record. We also store attachments and we keep an audit log for traceability.

### System boundary

1. We include the following within the automated system:
2. Reading a configured Outlook inbox
3. Detecting candidate order emails using subject keywords and an unread filter
4. Extracting order lines from supported email body formats
5. Computing order total, item count, and delivery estimate
6. Saving attachments into a local folder structure
7. Updating the restock spreadsheet on a local PC
8. Sending a confirmation email with the estimated delivery date and a clear item summary
9. Moving emails into one of four outcome folders
10. Logging processing outcomes into a CSV file

We exclude the following from scope:

1. Payment processing and invoicing systems
2. Warehouse dispatch and shipping carrier integration

3. Customer account management and CRM integration
4. Automatic inventory decrement and real time stock validation

## Operating context

The solution runs on Windows 11 with Outlook Classic and Microsoft Excel installed. We implemented the automation in UiPath Studio Community 2026.0.182 STS. The automation processes the mailbox account orderemailsautomationrpa@outlook.com.

## 3 CURRENT APPROACHES, DRAWBACKS, AND TECHNOLOGIES

### 3.1 CURRENT MANUAL APPROACH

In the current process, a staff member monitors a shared or dedicated Outlook inbox that receives B2B purchase orders and retail order requests. The mailbox also contains non order messages such as questions, updates, and marketing emails.

The staff member opens each email, decides if it is an order, and then extracts the order details. In many cases the order lines appear as plain text rather than a structured attachment. The staff member copies product names, quantities, and sometimes prices into an internal file. In our case study the internal file is a restock spreadsheet with columns Product and TotalQty.

Attachments such as purchase orders or invoices are downloaded and stored in a local folder. Folder naming is often inconsistent and depends on who handled the email. The staff member then calculates totals and prepares a reply with an estimated delivery date. Finally the staff member marks the email as handled, or moves it into a folder. When the mailbox is processed by multiple people, the same email can be processed twice or remain unprocessed.

This approach works at low volume but it does not scale well. It relies on careful attention and repeatable behavior from humans, which becomes harder as volume and time pressure increase.

### 3.2 DRAWBACKS OF THE MANUAL APPROACH

We identified recurring issues in the manual workflow.

#### Time and effort

Manual processing requires reading, copying, formatting, and updating Excel. Each step is small but the total time adds up. The time cost is visible both in the initial processing and in later rework when data was captured incorrectly.

#### Human error and inconsistency

Manual data entry is vulnerable to mistakes such as misreading quantity, omitting a line item, using a slightly different product naming, or saving attachments in an incorrect place. When policies evolve, human operators may not apply the same rule consistently across orders.

## **Delays and missed emails**

When the inbox is busy, orders can be buried by other messages. If orders are handled in batches, customers may not receive timely confirmation or delivery estimates.

## **Duplicate handling**

When multiple staff members share responsibility, an email can be handled twice, which can create incorrect restock quantities and confusion in audit trails.

## **Weak traceability**

Without a consistent log, it is hard to answer basic questions such as which emails were processed, when they were processed, and what action was taken. This is especially important for B2B purchasing departments that expect traceable confirmation.

## **Limited resilience**

If the restock file is locked or Excel fails, a manual worker might postpone the update, leading to out of sync inventory information. The manual process provides no consistent way to isolate failures or to keep partial results out of the system.

### **3.3 EXISTING TECHNICAL APPROACHES**

Organizations commonly use one or more of the following approaches to reduce the manual effort.

#### **Email templates and structured order forms**

Some companies ask customers to use a standard template. This reduces parsing complexity but it is difficult to enforce across retail customers and external purchasing departments. Even when a template is recommended, real emails often contain variations.

#### **Portals and e commerce platforms**

Retail orders often flow through an online portal that directly captures structured data. However, many companies still receive orders by email, especially for B2B customers, custom orders, and urgent requests. Migrating all customers to a portal can be costly and slow.

#### **Electronic data interchange**

For some B2B relationships, electronic data interchange provides a standardized order format. This approach requires agreements and technical integration on both sides. It is not practical for smaller partners and it does not address the long tail of less standardized email orders.

#### **Customer relationship management and enterprise resource planning integration**

A fully integrated workflow can connect an inbox, a CRM, and an ERP system. These solutions are powerful, but they require cost, data modeling, and long integration timelines. For our

project, the internal system of record is a local spreadsheet, so a lighter automation approach is more appropriate.

## **Scripting and custom development**

Teams sometimes build scripts in Python, PowerShell, or other languages to read emails and update files. This can be effective but it requires more development and maintenance effort. It also depends on code changes for configuration updates, unless a configuration UI is built.

## **Robotic process automation**

Robotic process automation automates the user interface and application steps that humans normally do. This can be a good fit when the target systems are not exposed through stable APIs, or when the workflow crosses multiple desktop applications. RPA tools typically provide strong orchestration, built in activities for common applications, and easier maintenance by non specialist teams.

For this project we selected robotic process automation because it aligns with the existing Outlook and Excel desktop workflow, it supports configuration driven execution, and it allows us to implement a complete end to end flow within the project timeframe.

### **3.4 TECHNOLOGIES USED IN OUR SOLUTION**

#### **UiPath Studio Community and UiPath Robot**

We implemented the workflow in UiPath Studio Community 2026.0.182 STS. We structured the automation as modular sequences to separate configuration, email retrieval, parsing, file handling, Excel update, email response, outcome routing, and logging.

#### **Microsoft Outlook Classic integration**

We used Outlook integration to read unread emails from a configured mailbox, to access sender email and message content, to read attachments metadata, to send reply emails, and to move messages into outcome folders.

#### **Microsoft Excel integration**

We used Excel integration to read and update the restock spreadsheet. We update quantities by matching the Product column and we create new rows when a product does not exist. Excel is required to be installed on the host machine.

#### **Text parsing and normalization**

We support three body formats. We use deterministic parsing rules to extract ProductName, Quantity, and UnitPrice. We normalize whitespace and separators, and we validate that quantity and price are numeric values. We treat parsing ambiguity as a NeedInfo outcome.

#### **Local file system storage**

We save attachments in a configured root folder. For traceability we create one folder per processed email. We use the OrderID as part of folder naming to reduce collisions.

## **Configuration and settings form**

We separate configuration from logic. The settings include the inbox account, subject keywords, unread processing, max emails to read, schedule time, delivery thresholds, delivery days, and file paths. We provide a settings form so the operator can update these values without editing workflows.

## **Audit logging**

We maintain a CSV log to support traceability. Each processed email produces one log record with timestamp, generated order ID, sender email, email subject, status, attachment count, and attachment folder path.

## **4 STAKEHOLDERS AND THEIR NEEDS**

We considered both retail and B2B order flows.

### **Retail customer**

Retail customers expect fast confirmation, a clear delivery estimate, and accurate capture of items and quantities. They also benefit from quick follow up when information is missing.

### **B2B customer and purchasing department**

B2B customers expect a traceable acknowledgment, consistent handling of purchase orders, predictable delivery policies, and reliable storage of supporting attachments.

### **Sales and order processing clerk**

The processing clerk needs lower workload, fewer copy and paste steps, reduced error rates, and clear handling for exceptional cases. The clerk also needs a clear place to check messages that require manual review.

### **Warehouse and inventory team**

The inventory team needs correct restock totals in the spreadsheet, fast visibility for high priority orders, and protection against duplicate updates.

### **Operations manager**

The manager needs shorter cycle time, consistent process execution, and measurable performance indicators such as extraction accuracy and NeedInfo rate.

### **IT and system administrator**

The administrator needs secure access, stable configuration, predictable error handling, and minimal maintenance overhead. The administrator also needs the ability to change keywords, thresholds, and paths without modifying the workflow logic.

## 5 SYSTEM REQUIREMENTS

### 5.1 OPERATIONAL REQUIREMENTS

**OR1** The solution shall operate on a Windows 11 workstation with Outlook Classic and Microsoft Excel installed.

**OR2** The solution shall process a configured Outlook mailbox account and shall send follow up emails from the same account.

**OR3** The solution shall run on a daily schedule configured by the operator. The default run time is 23:45.

**OR4** The solution shall process at most 50 emails per run by default and shall allow this value to be configured.

**OR5** The solution shall process only unread emails by default and shall allow this behavior to be configured.

### 5.2 FUNCTIONAL REQUIREMENTS

**FR1** The system shall read emails from the configured Outlook inbox and identify candidate order emails when the subject contains any configured keyword.

**FR2** The system shall support keywords order, purchase, invoice, and PO by default and shall allow the operator to modify the list through the settings form.

**FR3** The system shall extract the sender email address from the email message metadata.

**FR4** The system shall extract product lines from the email body in three supported formats, labeled format, CSV style rows, and pipe separated rows.

**FR5** The system shall extract ProductName, Quantity, and UnitPrice for each order line.

**FR6** The system shall compute LineTotal for each line as Quantity times UnitPrice.

**FR7** The system shall compute TotalPrice as the sum of all LineTotal values for the email.

**FR8** The system shall compute ItemCount as the number of extracted order lines.

**FR9** The system shall generate OrderID from the processing timestamp to ensure uniqueness within the log and file storage.

**FR10** The system shall determine HighPriority when TotalPrice is greater than FastDeliveryMinPrice and Quantity is not greater than FastDeliveryMaxQuantity.

**FR11** The system shall compute the estimated delivery date based on the delivery policy rules. If HighPriority is true the system shall use FastDaysDeliveryTime. Else if Quantity is greater than SlowDeliveryMinQuantity the system shall use SlowDaysDeliveryTime. Otherwise the system shall use StandardDaysDelivery.

**FR12** The system shall save attachments into a configured root folder and shall create one subfolder per processed email.

**FR13** The system shall update the restock spreadsheet in XLSX format by adding quantities to the TotalQty column for matching Product names.

**FR14** The system shall create a new row in the restock spreadsheet when Product does not exist.

**FR15** The system shall send a confirmation email that includes the estimated delivery date, item count, order total, and a line summary that includes line totals.

**FR16** The system shall move processed emails into one of four Outlook folders named Processed, HighPriority, NeedInfo, and Failed.

**FR17** The system shall append one record to the CSV orders log for each processed email. The record shall include Timestamp, OrderID, SenderEmail, EmailSubject, Status, AttachmentCount, and AttachmentFolderPath.

**FR18** When the system cannot parse the order lines with confidence it shall move the email to NeedInfo and it shall send a reply asking for missing or unclear information.

**FR19** When an unexpected processing error occurs the system shall move the email to Failed.

### 5.3 NON FUNCTIONAL REQUIREMENTS

**NFR1** Performance The system shall process each email in 30 seconds or less under normal conditions.

**NFR2** Reliability The system shall achieve at least 95 percent successful runs over the measurement period where a successful run means the workflow completes without crashing.

**NFR3** Traceability The system shall provide a CSV log that allows operators to trace each processed email to a timestamp, an order ID, a final folder status, and an attachment storage path.

**NFR4** Data minimization The system shall not store full email body content in any persistent file. The log shall store only the defined metadata fields.

**NFR5** Access control The attachments folder, restock spreadsheet, and orders log shall be stored in a restricted local directory accessible only to the Windows user account running the robot and an optional designated administrator.

**NFR6** Maintainability The system shall allow changes to keywords, thresholds, delivery days, file paths, and schedule time through the settings form without requiring workflow edits.

### 5.4 PHYSICAL REQUIREMENTS

**PR1** The host machine shall have stable access to the Outlook mailbox through the installed Outlook client.

**PR2** The host machine shall have permission to read and write to the configured local file paths for attachments, the restock spreadsheet, and the orders log.

**PR3** The host machine shall have Microsoft Excel installed to support spreadsheet updates.

## 6 PLANNED APPROACH

We followed a systems engineering workflow that starts with a clear definition of the problem, stakeholders, and requirements, and then proceeds to design, implementation, verification, and packaging.

### We used the following methods and tools:

1. Work breakdown structure to define and communicate work packages
2. Schedule plan with resource allocation to coordinate contributions across the team
3. SysML modeling to represent system structure, behavior, and requirements traceability
4. Risk management plan to identify and mitigate key risks early
5. Quality assurance plan with measurable metrics and test scenarios
6. Version control with shared project storage to maintain consistent artifacts

### 6.1 WORK BREAKDOWN STRUCTURE

We organized the project into four major workstreams. Solution management covers scope, configuration approach, and audit requirements. Platform and setup covers initialization, folder preparation, and error handling conventions. Order processing pipeline covers intake, parsing, business rules, attachments, Excel updates, follow up emails, and outcome routing. Quality, testing, and rollout covers test planning, validation, and delivery artifacts.

The complete work breakdown structure is included in Annex A.

### 6.2 SCHEDULE PLAN AND RESOURCE ALLOCATION

We planned work from 1 November 2025 to 24 December 2025. We aligned design and modeling early, then focused on workflow development, then dedicated time for testing, documentation, and presentation preparation.

We assigned roles based on strengths. Mahmoud handled project management and report and slides. Essam handled requirements and stakeholders. Zafar handled SysML modeling. Sakka handled UiPath development and testing. The Gantt chart and the resource allocation table are included in Annex B.

### 6.3 MODELING APPROACH

We modeled the system with SysML to ensure consistent understanding across stakeholders and to support traceability. We use two structure diagrams, a block definition diagram and an internal block diagram. We use two behavior diagrams, a use case diagram and a sequence diagram. We also provide a requirements diagram that traces key requirements to the main blocks. All diagrams are included in Annex D.

## 7 CHALLENGES, ISSUES, AND RISKS

Email orders are not guaranteed to be fully structured. Variations in how users write product lines are a major source of ambiguity. Another challenge is safe interaction with local files that can be locked by Excel or by other processes. We also need consistent folder routing to avoid duplicate work and to provide a clear operational view.

We created a risk management plan with eleven identified risks and defined response strategies. Major risks include false positives where non order emails match keywords, false negatives where real orders are missed due to limited keywords, parsing ambiguity that leads to incorrect quantities or prices, duplicate processing when unread state changes unexpectedly, file locking on the restock spreadsheet, and attachment save issues due to missing permissions or invalid paths.

We mitigate these risks by using configuration driven keyword lists, using OnlyProcessUnreadEmails by default, validating extracted numeric values, isolating each email so one failure does not break the whole run, and routing uncertain cases to NeedInfo. We also save attachments with deterministic folder naming based on OrderID and we maintain a consistent log record for each processed email.

The complete risk management plan is included in Annex C.

## 8 QUALITY ASSURANCE PLAN

We verify the solution with both functional tests and measurable quality metrics. Our testing covers the supported input formats, typical success paths, and expected failure paths such as NeedInfo routing and Failed routing. We also test file handling, Excel updates, and folder movements in Outlook.

We defined two quality metrics that we will measure during a two week observation period. During this period we will read about 50 emails and we expect about 40 of them to be real order emails.

### 8.1 METRIC 1 ORDER EXTRACTION ACCURACY

#### Measurement function

Accuracy equals the number of order emails where all extracted fields are correct for every line, divided by the total number of true order emails processed. An email is counted as correct only when ProductName, Quantity, and UnitPrice match the ground truth for every extracted line.

#### Stakeholder needs addressed

This metric supports the needs of the order processing clerk and the inventory team by reducing manual corrections and preventing incorrect restock updates. It supports the needs of customers by ensuring correct confirmation content.

## Quality attributes

Correctness, reliability, and maintainability. High accuracy also improves trust, which reduces manual oversight.

## Data collection scenario

In one run, we processed 50 emails and we manually label 40 of them as true order emails. We then compare the robot output for each order email against the labeled ground truth and compute the accuracy value.

### 8.2 METRIC 2 FAILED RATE

#### Measurement function

The restock update fail rate is defined as the number of restock file update attempts that **fail**, divided by the total number of restock file update attempts. A restock update attempt is considered **failed** if the automation cannot complete the update step (e.g., cannot open the file, cannot write changes, file locked, path invalid, permission issue, or an exception occurs during the update workflow).

#### Stakeholder needs addressed

This metric supports the order processing clerk and operations manager by showing how often the robot requires human intervention. It also supports customers by ensuring that unclear orders are detected and addressed quickly.

## Quality attributes

Usability, robustness, and operational efficiency.

## Data collection scenario

During the same evaluation window, we tracked each time the robot attempted to update the restock file. For every attempt, we logged whether the update **succeeded** or **failed**, then calculated the fail rate using the formula above.

### 9 IMPLEMENTATION

We implemented the full workflow end to end in UiPath. We designed the workflow to be configuration driven and to process each email independently so that one failure does not stop the entire run.

We structured the automation into the following logical modules:

- Initialization and settings load
- Email retrieval and filtering
- Order parsing and validation
- Business rules for totals, priority, and delivery estimate
- Attachment storage

- Restock spreadsheet update
- Customer reply email
- Outcome routing and audit logging

## 9.1 INITIALIZATION AND SETTINGS

At the start of each run we load settings from the settings file edited by the settings form. Settings include AccountEmail, subject keywords, OnlyProcessUnreadEmails, MaxEmailsToRead, RunTime, delivery thresholds and days, and local file paths. We validate that the configured file paths exist or can be created, and we validate that the restock spreadsheet path points to an XLSX file.

We also ensure that the Outlook folders Processed, HighPriority, NeedInfo, and Failed exist. If a folder does not exist we create it before processing begins.

## 9.2 EMAIL RETRIEVAL AND DETECTION

We retrieve emails from the Outlook inbox of AccountEmail. By default we select only unread emails and we limit the number of emails processed in one run to MaxEmailsToRead.

For each email we check whether the subject contains any keyword from SubjectKeywords. If no keyword matches we skip the email. If a keyword matches we treat the email as a candidate order email and continue with extraction.

## 9.3 PARSING SUPPORTED FORMATS

We parse the email body to extract order lines. We support three input patterns.

### **Labeled format**

In the labeled format, each line contains explicit labels for the fields. We extract the product name, quantity, and price from patterns that include Name, Qty, and Price. We normalize separators such as semicolons and we remove extra whitespace.

### **CSV style rows**

In the CSV style, each order line is represented as three comma separated values in the order Product, Quantity, UnitPrice. We split lines by newline and then split each line by commas.

### **Pipe separated rows**

In the pipe style, each order line is represented as three values separated by a vertical bar. We split lines by newline and then split each line by the vertical bar.

For all formats we validate that quantity is an integer and that unit price is a decimal number. If any required value is missing or invalid we route the email to NeedInfo and we send a reply asking for a clearer format.

## 9.4 BUSINESS RULES AND DELIVERY ESTIMATE

After parsing, we compute LineTotal for each line as Quantity times UnitPrice. We compute TotalPrice as the sum of all line totals and we compute ItemCount as the number of lines.

We generate an OrderID from the current processing timestamp. We use this OrderID in the attachment folder name and in the log.

We classify HighPriority when TotalPrice is greater than FastDeliveryMinPrice and the total quantity is not greater than FastDeliveryMaxQuantity.

### **We compute the estimated delivery date as follows:**

If HighPriority is true we set delivery days to FastDaysDeliveryTime

Else if quantity is greater than SlowDeliveryMinQuantity we set delivery days to SlowDaysDeliveryTime

Else we set delivery days to StandardDaysDelivery

We then set ETA date as today plus delivery days.

## 9.5 ATTACHMENTS HANDLING

When the email contains attachments we create a dedicated folder under AttachmentsFolderPath for the current email and we save all attachments into this folder. We record the folder path and the number of saved files for logging.

If attachment save fails due to permission or path issues we route the email to Failed and we continue to the next email.

## 9.6 RESTOCK SPREADSHEET UPDATE

We open the restock XLSX file from RestockExcelPath and we read the existing rows with columns Product and TotalQty.

For each parsed order line we search for a matching Product. If a match exists we add the line quantity to TotalQty. If no match exists we append a new row with the new product name and the quantity.

If the spreadsheet is locked or cannot be updated we route the email to Failed to prevent inconsistent inventory totals.

## 9.7 CUSTOMER REPLY EMAIL

For processed orders we send a confirmation email back to the sender. The message includes the estimated delivery date, the delivery days used, item count, order total, and a line summary. The subject follows the format Order received - estimated delivery YYYY-MM-DD.

If the email is routed to NeedInfo we send a reply that requests the missing information and suggests using one of the supported formats.

## 9.8 OUTCOME ROUTING AND AUDIT LOG

After processing we move the email into one of the outcome folders.

**HighPriority** when the HighPriority rule is true and processing succeeded

**Processed** when processing succeeded and HighPriority is false

**NeedInfo** when parsing or validation fails

**Failed** when an unexpected error occurs

We then append one row to the CSV log at OrdersLogPath. The row includes Timestamp, OrderID, SenderEmail, EmailSubject, Status, AttachmentCount, and AttachmentFolderPath.

## 10 CONCLUSIONS

We delivered an end to end automation for email based order processing that improves an existing manual workflow. We integrated Outlook Classic, a local file system, and an Excel based restock spreadsheet using UiPath Studio Community. The solution detects order emails by subject keywords, extracts order lines from three common email body formats, computes totals and delivery estimates based on configurable policies, saves attachments in a structured way, updates inventory totals in Excel, sends a confirmation email to customers, routes messages into clear operational folders, and logs each processing outcome in a CSV file.

Our approach is better than the manual baseline because it is consistent, traceable, and configurable. It reduces repetitive work and limits copy and paste errors. It also provides a clear separation between normal processing, emails that require clarification, and emails that failed due to unexpected issues. The settings form enables operators to adjust keywords, thresholds, delivery policy values, and file paths without workflow edits. This reduces maintenance effort and makes the system suitable for both retail and B2B order flows.

We also planned the project using a work breakdown structure, a schedule plan with responsibilities, risk management, and a quality assurance plan with measurable metrics. These systems engineering practices help ensure that our implementation remains aligned with stakeholder needs and that the final system is verifiable and maintainable.

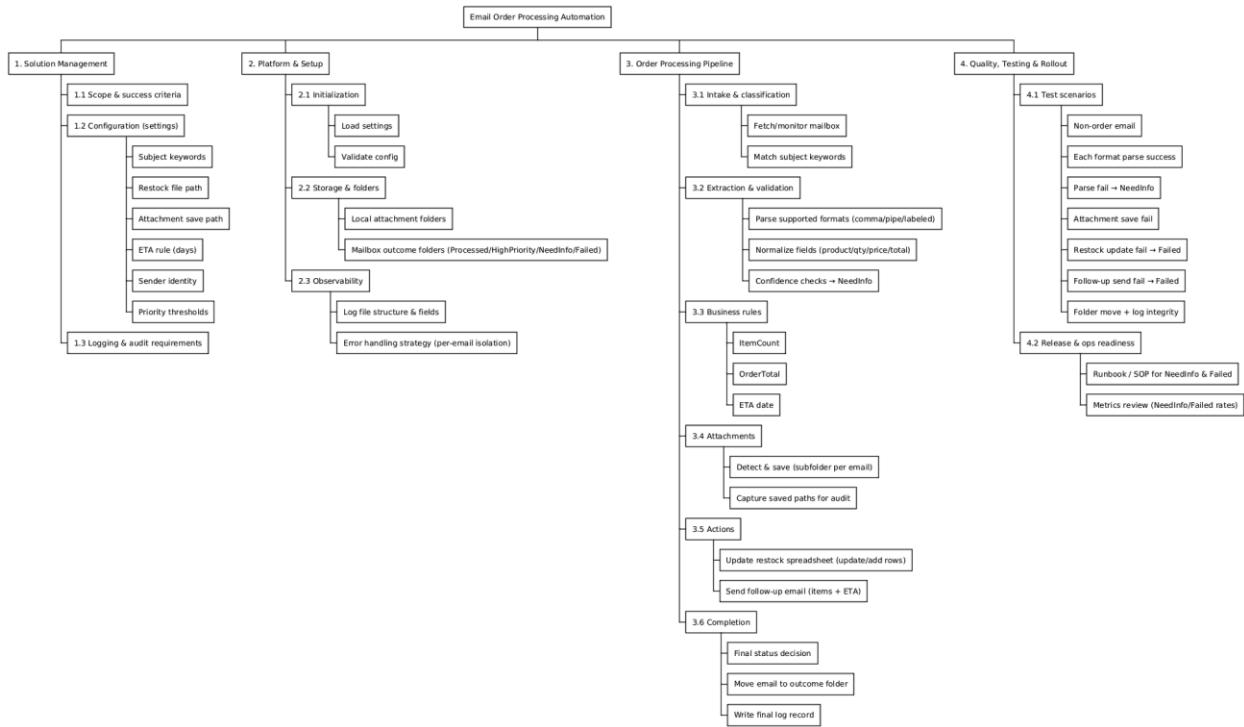
## REFERENCES

- Moreira, S., Mamede, H. S., and Santos, A. 2023. Process automation using robotic process automation A literature review. Procedia Computer Science, 219, 202–211.  
<https://doi.org/10.1016/j.procs.2023.01.287>
- Robotic process automation. n.d. Wikipedia.  
[https://en.wikipedia.org/wiki/Robotic\\_process\\_automation](https://en.wikipedia.org/wiki/Robotic_process_automation)

- Ivancic, L., Susa Vugec, D., and Bosilj Vuksic, V. 2019. Robotic process automation Systematic literature review. In Business Process Management Blockchain and Central and Eastern Europe Forum. Springer. [https://doi.org/10.1007/978-3-030-30429-4\\_19](https://doi.org/10.1007/978-3-030-30429-4_19)
- Microsoft. n.d. Security behavior of the Outlook object model. Microsoft Learn. Retrieved January 2, 2026, from <https://learn.microsoft.com/en-us/office/vba/outlook/how-to/security/security-behavior-of-the-outlook-object-model>
- National Institute of Standards and Technology. 2020. NIST Privacy Framework A tool for improving privacy through enterprise risk management Version 1.0. [https://www.nist.gov/system/files/documents/2020/01/16/NIST%20Privacy%20Framework\\_V1.0.pdf](https://www.nist.gov/system/files/documents/2020/01/16/NIST%20Privacy%20Framework_V1.0.pdf)
- List of system quality attributes. n.d. Wikipedia. [https://en.wikipedia.org/wiki/List\\_of\\_system\\_quality\\_attributes](https://en.wikipedia.org/wiki/List_of_system_quality_attributes)
- Ribeiro, J., Lima, R. B., Eckhardt, T., and Paiva, S. 2021. Robotic process automation and artificial intelligence in Industry 4.0 A literature review. Procedia Computer Science, 181, 51-58. <https://doi.org/10.1016/j.procs.2021.01.104>
- UiPath. 2025 December 9. About the Mail activities package. UiPath Documentation Portal. <https://docs.uipath.com/activities/other/latest/productivity/about-the-mail-activities-pack>
- UiPath. 2025 December 9. Get Outlook Mail Messages. UiPath Documentation Portal. <https://docs.uipath.com/activities/other/latest/productivity/get-outlook-mail-messages>
- UiPath. 2025 December 9. Move Outlook Mail Message. UiPath Documentation Portal. <https://docs.uipath.com/activities/other/latest/productivity/move-outlook-message>
- UiPath. 2025 December 9. Read Range. UiPath Documentation Portal. <https://docs.uipath.com/activities/other/latest/productivity/excel-read-range>
- van der Aalst, W. M. P., Bichler, M., and Heinzl, A. 2018. Robotic process automation. Business and Information Systems Engineering, 60 4, 269-272. <https://doi.org/10.1007/s12599-018-0542-4>

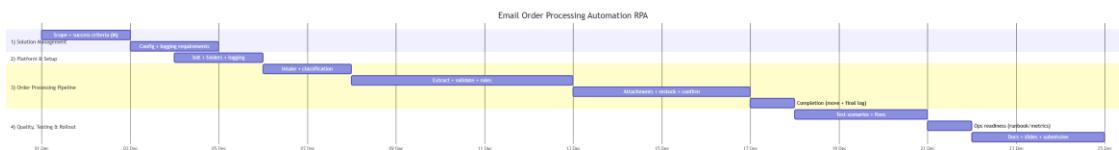
## ANNEX A WORK BREAKDOWN STRUCTURE

The following figure shows the work breakdown structure used to plan the project.



## ANNEX B GANTT CHART AND RESOURCE ALLOCATION

The following figure shows the planned schedule from 1 Dec 2025 to 24 December 2025.



## RESOURCE ALLOCATION

MAHMOUD MIRGHANI Abdelrahman Project manager and coordinator. UiPath development.

ESSAM AL KHALIDY Business analyst. Requirements and stakeholder analysis. Report writing.

ZAFAR Azzam SysML modeling and diagram production.

SAKKA Mohamad Mario UiPath development and quality assurance. PPT.

## ANNEX C RISK MANAGEMENT PLAN SUMMARY

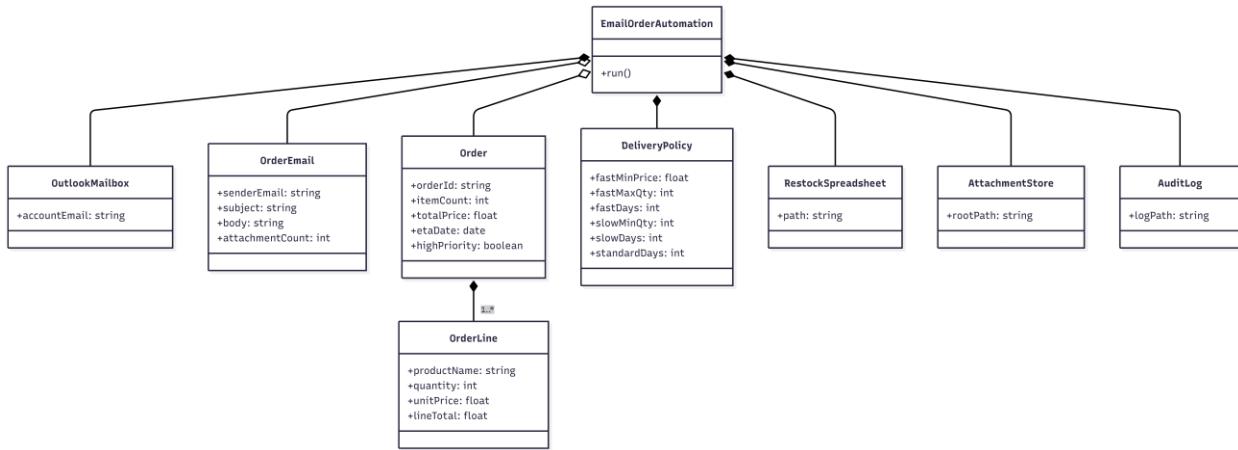
This table summarizes the main project risks and response strategies. The detailed plan is maintained in the risk management spreadsheet.

| ID      | ThreatEvent   | Trigger  | Probability | Impact    | Strategy               | Owner                     |
|---------|---|--|-------------|-----------|------------------------|---------------------------|
| OA-R001 | Non-order email matches subject keywords and gets processed as an order.                          | Increase in Processed emails with empty/odd item extraction or customer complaints.              | Moderate    | Moderate  | Mitigation             | Mailbox/Admin + Developer |
| OA-R002 | Real orders are missed due to incomplete keyword rules (false negatives).                         | Ops reports missing expected orders; order volume mismatch vs manual checks.                     | Moderate    | High      | Mitigation             | Ops Owner + Mailbox/Admin |
| OA-R003 | Email body format differs from supported patterns; extraction fails and NeedInfo volume spikes.   | NeedInfo rate increases; repeated parse failures from same sender.                               | High        | Moderate  | Mitigation             | Developer + QA            |
| OA-R004 | Incorrect parsing (wrong qty/price) silently produces wrong totals/ETA and wrong restock updates. | Totals not matching line items; unusual quantities/prices; customer disputes.                    | Low         | Very High | Avoidance / Mitigation | Developer                 |
| OA-R005 | Duplicate processing of the same email causes double increments in the restock spreadsheet.       | Same sender/subject processed multiple times; repeated internal IDs; spreadsheet jump anomalies. | Moderate    | Very High | Avoidance / Mitigation | Developer + Ops Owner     |
| OA-R006 | Attachments fail to save or overwrite due to filename collisions.                                 | Missing attachments on disk; identical filenames from different emails.                          | Moderate    | High      | Mitigation             | Developer                 |
| OA-R007 | Restock spreadsheet is locked, corrupted, or its schema changes, causing update failures.         | Write errors; missing columns; file lock conflicts.  | Moderate    | Very High | Mitigation             | Developer + Ops Owner     |
| OA-R008 | Misconfigured delivery-days rule generates incorrect ETA in follow-up                             | ETAs inconsistent with policy; unusually far/near dates.   | Moderate    | Moderate  | Mitigation             | Project Owner + Developer |

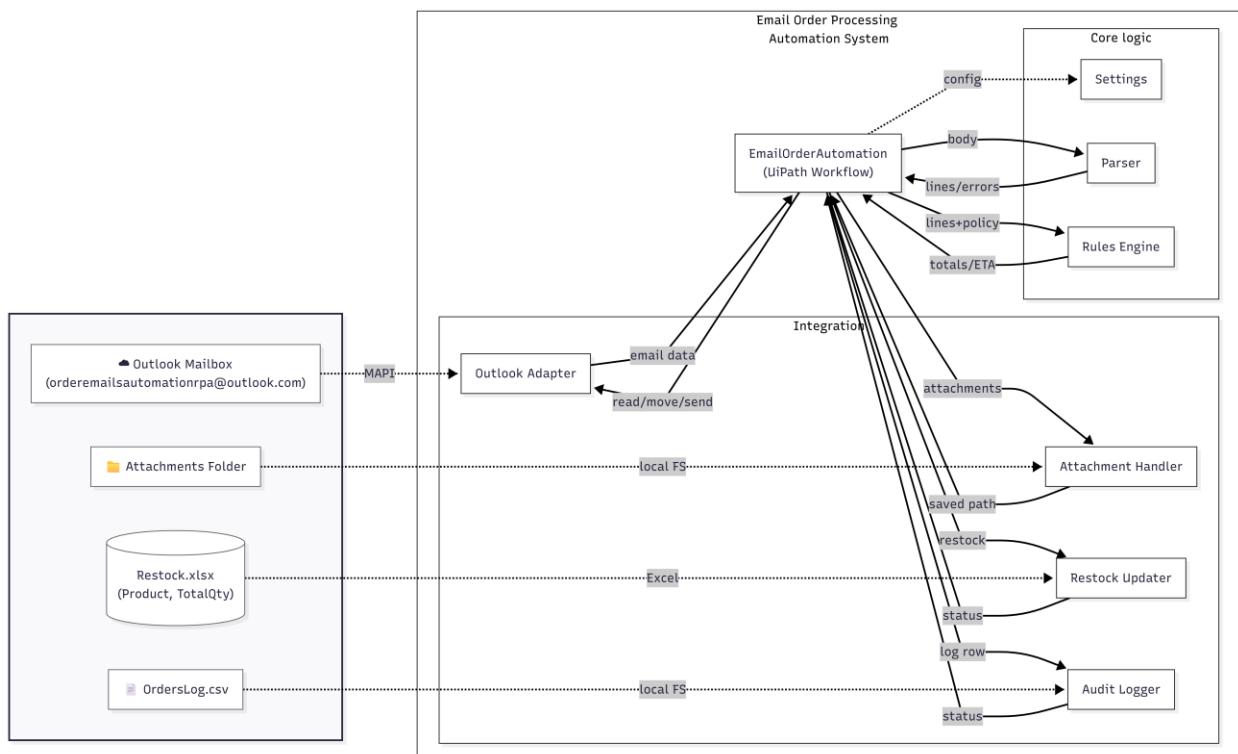
|         | messages.  |  |     |           |                        |                               |
|---------|--|--|-----|-----------|------------------------|-------------------------------|
| OA-R009 | Mailbox foldering fails (missing folders/permissions) leaving status unclear and inbox cluttered.            | Emails remain in inbox; move operation errors.                           | Low | Moderate  | Mitigation             | Mailbox/Admin                 |
| OA-R010 | Log record is missing or does not reflect final status after downstream actions (audit gap).                 | Mismatch between folder status and log; missing attachment paths in log. | Low | Very High | Avoidance / Mitigation | Developer                     |
| OA-R011 | Security/privacy issue: attachments or logs stored in an insecure location or with overly broad permissions. | Unauthorized access report; storage path not approved.                   | Low | Very High | Avoidance / Mitigation | Project Owner + Mailbox/Admin |

## ANNEX D SYSML DIAGRAMS

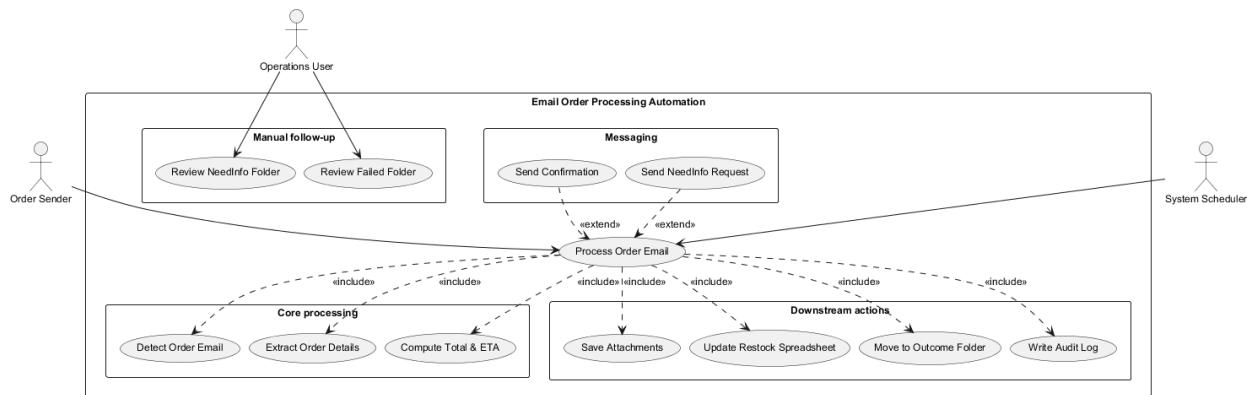
### BLOCK DEFINITION DIAGRAM



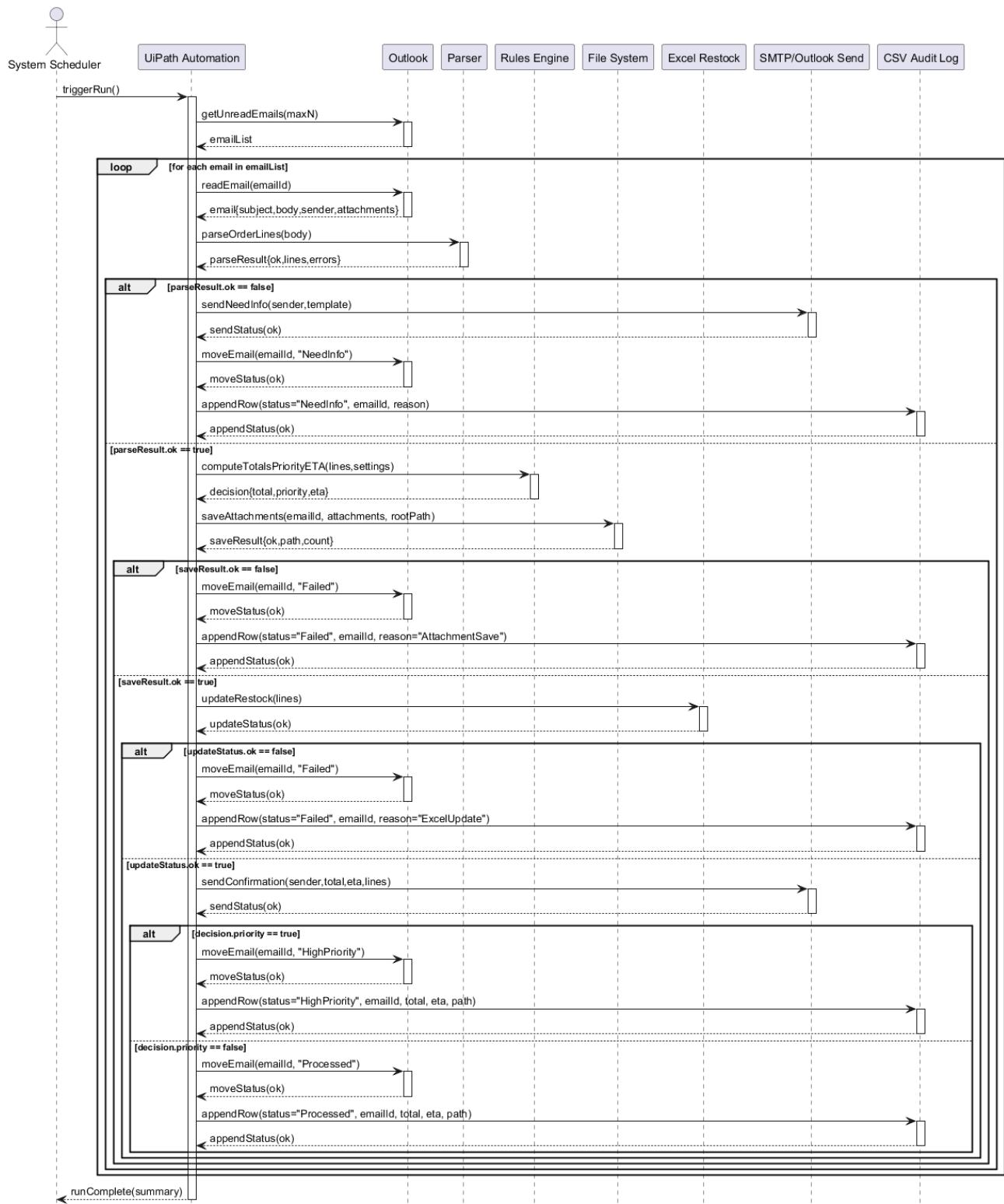
### INTERNAL BLOCK DIAGRAM



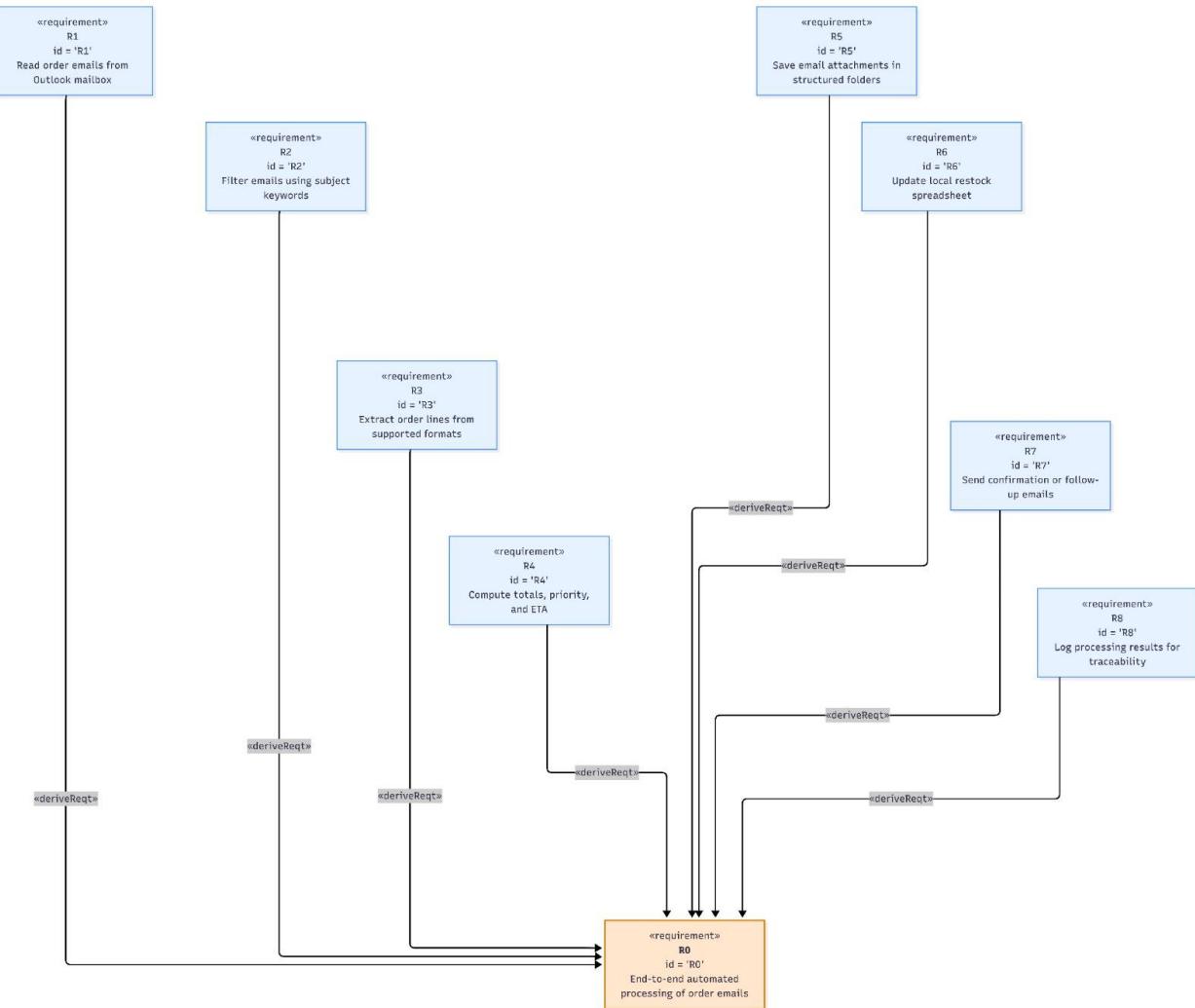
## USE CASE DIAGRAM



## SEQUENCE DIAGRAM



## REQUIREMENTS DIAGRAM



## ANNEX E CONFIGURATION VALUES AND SAMPLE EMAILS

### E1 DEFAULT CONFIGURATION VALUES

#### Emails

AccountEmail orderemailsautomationrpa@outlook.com  
RunTime 23:45  
MaxEmailsToRead 50  
OnlyProcessUnreadEmails true  
SubjectKeywords order, purchase, invoice, PO

#### Policy

FastDeliveryMinPrice 200  
FastDeliveryMaxQuantity 10  
FastDaysDeliveryTime 2  
SlowDeliveryMinQuantity 20  
SlowDaysDeliveryTime 5  
StandardDaysDelivery 3

#### File paths

AttachmentsFolderPath configured by operator  
RestockExcelPath configured by operator  
OrdersLogPath configured by operator

### E2 SAMPLE LABELED EMAIL BODY

Hello  
Objective  
Replace a critical part and minimize downtime  
Fast delivery preferred if eligible

Order items

Name Printer Drum Unit Qty 1 Price 245.00

Thanks

Mario

#### E3 SAMPLE CSV STYLE EMAIL BODY

Hi

Objective

Restock IT accessories

Confirm ETA and attach an order confirmation

Items

USB Cable,4,3.99

HDMI Cable,2,9.50

Mouse Pad,6,2.25

#### E4 SAMPLE PIPE SEPARATED EMAIL BODY

Hello

Objective

Restock small items for the office

Reply with ETA

Items

Paper A4 | 5 | 6.25

Staples Box | 2 | 3.40

Sticky Notes | 6 | 1.15