Section 2: Functional Specifications

This section defines the core functionalities of Orgo, detailing how each feature operates to fulfill its purpose. It provides precise descriptions of system capabilities, workflows, and modular functionalities tailored for diverse organizational needs.

---

2.1 Purpose of Functional Specifications

Objective: To define the features and operations of Orgo, ensuring that all components and workflows align with organizational goals.

Outcome:

Clear understanding of system capabilities.

Detailed workflow processes for key use cases.

Modular flexibility for customization and scalability.

---

2.2 Core Functionalities

1. Email Parsing:

Extracts key information from incoming emails, including:

Subject line.

Sender/recipient email addresses.

Body content.

Attachments (if any).

Example:

Email Subject: "Urgent: Water Leak in Room 101."

Extracted Metadata:

Sender: secretary@organization.com

Recipient: emergency@organization.com

Keywords: ["urgent", "leak", "room 101"]

2. Role-Based Routing:

Assigns messages to appropriate roles or departments based on:

Keywords in the email subject or body.

Predefined routing rules in YAML/JSON.

Example:

Keywords "leak" and "urgent" route the email to maintenance@organization.com.

3. Dynamic Attachment and Template Management:

Automatically attaches relevant files and templates to routed emails.

Example:

Attach:

Location\_Map.pdf (specific to "Room 101").

Leak\_Protocol.pdf (standard leak response guidelines).

Preformatted Templates:

Auto-generate task-specific response forms.

4. Feedback Loop Integration:

Parses replies to update workflows dynamically.

Example:

Technician replies: "ETA 30 minutes."

Orgo logs the response and updates the database.

5. Workflow Escalation:

Triggers escalations for unresolved tasks based on time limits or priority levels.

Example:

If no action is taken on a maintenance request in 2 hours, escalate to supervisor@organization.com.

6. Sensitive Data Anonymization:

Removes identifying metadata for sensitive reports (e.g., harassment cases).

Example:

Original Sender: employee@company.com

Routed Email: Replaces sender with "Anon1."

7. Offline Processing:

Handles emails locally when internet connectivity is unavailable.

Syncs data to the main database when connectivity is restored.

8. Logging and Audit Trails:

Maintains a log of all routing decisions, escalations, and workflows for transparency and compliance.

Example:

Log Entry:

Email ID: 12345

Sender: secretary@organization.com

Keywords: [urgent, leak]

Routed To: maintenance@organization.com

Status: Completed

---

2.3 Workflow Examples

1. Maintenance Request Workflow:

Trigger: Secretary emails emergency@organization.com about a water leak.

Steps:

1. Email is parsed for metadata and keywords.

2. Rule engine routes the email to maintenance@organization.com.

3. Orgo attaches a map of the location and a leak protocol document.

4. Technician replies with ETA and status updates.

5. If unresolved in 2 hours, Orgo escalates to the supervisor.

2. Harassment Reporting Workflow:

Trigger: Employee emails report@organization.com about harassment.

Steps:

1. Orgo anonymizes the sender and strips identifying metadata.

2. Email is routed to HR and legal advisors.

3. Attachments include the company’s harassment policy and investigation guidelines.

4. HR logs investigation steps and updates.

5. Case status is logged in the system for compliance tracking.

3. Offline Operations Workflow:

Trigger: Organization operates without internet connectivity.

Steps:

1. Incoming emails are stored locally in .mbox or .pst files.

2. Orgo processes emails using SQLite for local storage.

3. Once connectivity is restored, data syncs with the PostgreSQL database.

---

2.4 Modular Functionalities

Orgo’s functionality is modular, allowing organizations to implement only the features they need:

Maintenance Module:

Focused on incident reporting, dynamic document attachment, and task escalation.

HR Module:

Handles sensitive workflows like harassment reports or policy communication.

Education Module:

Manages teacher-parent communication, scheduling, and incident reporting.

Government Module:

Supports crisis management workflows, resource allocation, and inter-departmental communication.

---

2.5 Key Benefits

1. Efficiency:

Automates manual tasks like sorting and routing emails.

Reduces delays with escalations and dynamic updates.

2. Reliability:

Ensures consistent performance through offline processing.

Logs all actions for traceability.

3. Scalability:

Adapts to different organizational sizes and industries.

4. Privacy:

Protects sensitive workflows with anonymization and RBAC.

---

2.6 Deliverables

1. Functional Flow Diagrams:

Visual representation of workflows (e.g., maintenance, harassment reporting).

2. Rule Definitions:

Sample YAML/JSON files for routing and escalation rules.

3. Workflow Templates:

Preformatted examples for task-specific responses.

4. Anonymization Protocols:

Guidelines for stripping sensitive metadata.

---

Summary

This section defines Orgo’s core and modular functionalities, focusing on efficient email parsing, dynamic routing, and workflow management. By integrating sensitive data handling, offline processing, and scalability, Orgo is tailored to diverse organizational needs while maintaining privacy and operational continuity.