

AI Customer Support Resolution System

Detailed Master Case Study | End-to-end support automation with AI classification, database verification, RAG, human review, SLA monitoring, follow-up, and closure

EXECUTIVE SUMMARY

Project 043 is a modular customer support resolution system built in n8n to automate the complete lifecycle of a support case. The solution receives customer emails, standardizes the request, uses AI to classify the issue, persists a structured ticket in PostgreSQL, verifies customer and order information when required, retrieves relevant support policy from a vector knowledge base, generates a grounded resolution, routes uncertain cases to human review, and continues monitoring the case through SLA escalation, customer follow-up, and final closure.

The design intentionally separates AI reasoning from business control. AI is used for language understanding and response generation, while deterministic rules, database verification, structured outputs, and human-review gates control whether the system is allowed to proceed. This makes the workflow more suitable for real support operations than a single autonomous chatbot.

Business Problem

- Support teams repeatedly read and categorize inbound emails, search customer/order records, locate policies, prepare responses, and remember which cases still require action.
- Manual handoffs can create inconsistent routing, slow response times, missed follow-ups, and limited visibility into unresolved cases.
- AI-only support creates additional risk when the model does not have verified customer context or attempts to infer refund, replacement, delivery, or account decisions without policy evidence.

Solution Goals

- Automate repetitive support work without removing human judgment from high-risk or uncertain cases.
- Create a shared, persistent ticket lifecycle across multiple specialized workflows.
- Ground customer-facing AI responses in retrieved business policy and verified operational data.
- Keep unresolved cases visible through SLA checks, escalation, follow-up, and automated closure rules.

Four-Workflow Architecture

Workflow	Role in the System	Key Controls
WF01 - Support Ticket Intake & Classification	Captures support email, normalizes data, generates ticket ID, classifies the request, assigns priority, stores the ticket, and routes the case.	Structured AI output, confidence checks, human-review gate, PostgreSQL persistence.
WF02 - Customer & Order Investigation	Retrieves customer/order records and verifies identity, order existence, and ownership before order-related automation.	Required-field validation, database lookup, ownership verification, exception routing.
WF03 - AI Resolution & Knowledge Engine	Retrieves relevant support policy, builds resolution context, generates a structured response, and decides between auto resolution and human review.	RAG, embeddings, structured parser, safety gate, Slack escalation, status updates.
WF04 - Follow-Up, SLA & Case Closure	Runs scheduled operational checks, escalates overdue tickets, sends follow-ups, records activity, and closes eligible cases.	Priority-based SLA logic, escalation throttling, follow-up state, scheduled closure.

Core Data Model & Status Lifecycle

PostgreSQL acts as the shared operational source of truth. Ticket records contain customer identity, email content, AI classification fields, category, priority, urgency, sentiment, confidence, human-attention flags, timestamps, and lifecycle status. Customer and order tables provide verification context for order-related cases.

Status	Operational Meaning
classified	Ticket has been classified and is ready for investigation, knowledge resolution, or review.
pending_human_review	Automation has intentionally paused for staff judgment.
response_sent	A customer-facing resolution has been sent and the case enters post-response monitoring.
follow_up_sent	A reminder has been sent after the configured waiting period.
closed	The support lifecycle has completed and no further automated action is required.

1. Intake, Classification & Routing

WF01 begins with a Gmail support trigger. Incoming message data is normalized so downstream nodes receive consistent fields regardless of the original email formatting. A ticket ID is generated, then the AI classifier converts the unstructured request into structured support attributes such as category, issue summary, order number, sentiment, urgency, classification confidence, and whether human attention is required. Business priority is calculated separately, preserving deterministic control over SLA behavior. The completed record is inserted into PostgreSQL before routing so the case is persisted even if later workflows fail.

2. Customer & Order Verification

Order-related cases are handed to WF02 rather than being resolved immediately. The workflow validates required investigation inputs, queries PostgreSQL for matching customer and order records, then builds a combined investigation context. It verifies the customer, the order, and the ownership relationship. Verified cases become `ready_for_resolution` and are handed to WF03; invalid, incomplete, or unverified cases are diverted to exception or human-review paths. This prevents the AI layer from acting on an order simply because a customer typed an order number in an email.

3. Knowledge-Grounded AI Resolution

WF03 validates the resolution context and selects a knowledge topic based on the support category. Relevant NovaCart support policy is retrieved from a vector knowledge base using embeddings. The retrieved policy is combined with verified case information to create the resolution context supplied to the LLM. A structured output parser ensures predictable response fields. The resulting output is validated again before a safety gate decides whether the case can proceed automatically or must be escalated.

4. Human Review & Auto Resolution

Cases that are uncertain, sensitive, insufficiently verified, or otherwise unsuitable for automation enter the human-review path. Staff receive a Slack alert and the ticket is marked `pending_human_review`. Eligible cases enter the auto-resolution path, where the customer response is prepared and sent through Gmail and PostgreSQL is updated to `response_sent`. The system is designed not to invent stock availability, refund approval, replacement eligibility, or other unsupported operational facts.

5. SLA Monitoring, Follow-Up & Closure

WF04 runs on a recurring schedule and retrieves active tickets from PostgreSQL. It calculates ticket age, hours since the last update, priority-based SLA target, breach state, and the required follow-up action. Unresolved and human-review cases can generate Slack escalations. Escalation activity is recorded so the same alert is not sent every hour. Tickets in `response_sent` can trigger a customer follow-up after the waiting period; the follow-up is recorded as `follow_up_sent`. After the subsequent waiting period, eligible cases are automatically marked closed.

Technology Stack

Technology	Implementation Role
n8n	Workflow orchestration, sub-workflow handoffs, scheduled monitoring, branching, validation, and integrations.
PostgreSQL	Persistent tickets, customers, orders, lifecycle state, and operational updates.
LLM + Structured Output	Support classification and grounded resolution generation with machine-readable outputs.
Vector Store + Gemini Embeddings	Semantic retrieval of relevant internal support policy for RAG.
Gmail	Inbound support intake, customer resolution emails, and follow-up messages.
Slack	Human-review alerts and SLA escalation notifications.

Testing & Validation

- Tested successful ticket classification and routing from WF01 into downstream investigation.
- Verified customer/order lookup and ownership checks before resolution handoff.
- Validated RAG retrieval, structured AI output, safety routing, Gmail delivery, and PostgreSQL status updates in WF03.
- Tested WF04 with controlled data to verify customer follow-up routing and persistent `follow_up_sent` state.
- During final testing, corrected legacy SQL references from an obsolete tickets table to `public.support_tickets`, confirming the production path against the actual schema.

Business Impact & Production Value

- Reduces manual triage, record lookup, routing, policy search, routine communication, and follow-up effort.
- Improves consistency because the same verification, priority, routing, and policy rules are applied to every case.
- Reduces unsupported AI behavior through RAG, structured outputs, verification gates, and explicit human escalation.
- Creates an auditable support lifecycle that can be extended into dashboards, CRM/help-desk integrations, analytics, and controlled action-taking agents.

Key takeaway: Project 043 demonstrates a practical pattern for AI automation in customer support - AI handles language and context, while verified data, business rules, persistent state, and human judgment control what the system is allowed to do.