

Detailed Workflow Notes

AI Customer Support Resolution System | Technical workflow reference

WORKFLOW 01 - SUPPORT TICKET INTAKE & CLASSIFICATION

Purpose

Convert an unstructured customer support email into a validated, prioritized, persistent support ticket and route it to the correct downstream process.

Node Flow & Logic

Stage	Key Nodes / Logic	Output
Intake & normalization	New Support Email -> Normalize Support Request -> Generate Ticket ID	Standardized customer/email payload with unique ticket ID.
AI classification	AI Ticket Classifier + LLM + Structured Output Parser	Category, issue summary, order number, sentiment, urgency, confidence, human-attention flag.
Enrichment & storage	Build Ticket Record -> Determine Ticket Priority -> Store Support Ticket	Canonical support record persisted in public.support_tickets.
Safety & routing	Human Review Gate -> Route Support Case	Human review, order investigation, or knowledge-resolution handoff.

Important Notes

- Classification and priority are separated: AI interprets the request while deterministic business logic controls operational priority.
- The ticket is stored before downstream handoff, preserving a record even if a later workflow fails.
- Flagged or low-confidence cases can be diverted before automated resolution.

Workflow 01 Representative Fields

ticket_id, customer_email, customer_name, email_subject, email_body, category, issue_summary, order_number, sentiment, urgency, classification_confidence, requires_human_attention, status, priority, priority_reason.

WORKFLOW 02 - CUSTOMER & ORDER INVESTIGATION

Purpose

Verify the operational facts required for order-related support before the AI resolution layer is allowed to act.

Node Flow & Logic

Stage	Logic
Request validation	Receive Investigation Request -> Validate Investigation Input -> Investigation Input Gate. Missing or invalid context is stopped before database investigation.
Database lookup	Lookup Customer Order queries PostgreSQL using the support case identifiers and retrieves matching customer/order data.
Verification	Build Investigation Context combines ticket and database information. Evaluate Investigation Result checks customer, order, and ownership match.
Decision routing	Investigation Decision Gate routes READY FOR RESOLUTION, HUMAN REVIEW, or INVALID / UNVERIFIED.
Handoff	Verified cases are prepared for WF03. Human-review and exception branches preserve structured case context.

Verification Controls

- customer_verified confirms a valid customer match.
- order_verified confirms the referenced order exists.
- ownership_verified confirms the order belongs to the customer before automated order-specific resolution.
- Verified cases use investigation_result = verified_match and case_status = ready_for_resolution.

WORKFLOW 03 - AI RESOLUTION & KNOWLEDGE ENGINE

Purpose

Generate a customer-support resolution using verified case context and retrieved business policy, then decide whether the response is safe for automation.

Node Flow & Logic

Stage	Key Logic	Control
Context validation	Receive Resolution Request -> Validate Resolution Context -> Resolution Context Gate.	Checks required fields and verification state.
Knowledge retrieval	Select Knowledge Topic -> Retrieve Relevant Policy using vector search + Gemini embeddings.	Grounds the AI in relevant NovaCart support policy.
Resolution engine	Build Resolution Context -> AI Resolution Engine + LLM + Structured Output Parser.	Produces predictable structured resolution output.
Safety routing	Validate Resolution Output -> Resolution Safety Gate.	Determines auto resolution vs human review.
Human path	Prepare Human Review -> Slack alert -> Mark Ticket Pending Review.	Automation pauses for staff judgment.
Auto path	Prepare Auto Resolution -> Gmail customer response -> Mark Ticket Response Sent.	Updates public.support_tickets to response_sent.

Knowledge & Safety Notes

- Knowledge topics cover common support scenarios such as damaged items, wrong items, refunds, cancellations, delivery issues, payment issues, account support, product questions, and general inquiries.
- The AI must not invent stock availability, refund approval, replacement approval, or other unsupported operational facts.
- Incomplete verification, suspicious cases, sensitive decisions, or insufficient context are appropriate reasons for human review.

WORKFLOW 04 - FOLLOW-UP, SLA & CASE CLOSURE

Purpose

Provide the operational layer after initial resolution by monitoring active tickets, enforcing SLA rules, following up with customers, and closing completed cases.

Node Flow & Logic

Stage	Behavior
Scheduled monitoring	Scheduled Support Case Check runs hourly and loads active classified, pending_human_review, response_sent, and follow_up_sent tickets.
SLA evaluation	Evaluate Ticket SLA & Follow-Up Status calculates ticket age, hours since update, SLA target, breach state, and follow_up_action.
Routing	Route Follow-Up Action directs cases to escalation, customer follow-up, monitoring, or closure.
Escalation	Slack alerts support staff for overdue unresolved or human-review cases. PostgreSQL records the escalation to reduce repeated hourly alerts.
Customer follow-up	response_sent cases can trigger a follow-up after the waiting period. Gmail sends the reminder and status becomes follow_up_sent.
Closure	After the follow-up waiting period, eligible follow_up_sent tickets are updated to closed.

Shared Status Lifecycle

Status	Meaning
classified	Classified and awaiting operational action.
pending_human_review	Paused for staff review.
response_sent	Initial customer resolution delivered.
follow_up_sent	Post-resolution follow-up delivered.
closed	Support lifecycle complete.

Cross-Workflow Design Principles

- PostgreSQL provides persistent state and connects workflows without requiring one giant workflow.
- Deterministic validation and routing surround AI nodes so the model is not responsible for authorization or lifecycle control.
- Human review is a designed business path, not simply an error condition.
- Immediate resolution (WF03) is separated from scheduled operations (WF04), making the system easier to maintain and scale.

Technical pattern: Intake -> Persist -> Verify -> Retrieve -> Generate -> Validate -> Act / Escalate -> Monitor -> Follow Up -> Close.