Technical Assessment

Multi-Agent Voice System for Electronic Consumer Store

Position: Senior GenAI Engineer
Duration: 3-5 Day
Submission: GitHub Repository + Demo Recording

Overview

Build a multi-agent system that intelligently routes and handles customer inquiries for an electronic consumer store. The system should simulate receiving voice/text inquiries and delegate them to specialized sub-agents based on the nature of the request.

Business Context

ElectroMart is an electronic consumer store that receives hundreds of daily customer inquiries across various departments. Currently, customers experience long wait times due to manual routing. Your task is to build an intelligent multi-agent system that:

1. Understands customer intent from their initial inquiry
2. Routes to the appropriate specialized agent
3. Provides contextual, accurate responses
4. Handles handoffs between agents when needed

Core Requirements

1. Orchestrator Agent (Router)

The main agent that receives all incoming inquiries and determines routing.

Responsibilities:

- Analyze incoming customer message/query
- Classify intent and determine appropriate sub-agent
- Route conversation to the correct specialist
- Handle cases where multiple agents may be needed
- Manage conversation handoffs between agents

Expected Capabilities:

- Intent classification (minimum 85% accuracy on test cases)
- Context retention across the conversation
- Graceful handling of ambiguous queries
- Escalation logic for complex scenarios

---

2. Sub-Agents (Implement at least 4)

Each sub-agent must have its own **Knowledge Base** containing domain-specific information.

A. Sales Agent

Domain: Product inquiries, pricing, availability, recommendations

Should Handle:

- Product specifications and comparisons
- Pricing inquiries
- Stock availability
- Product recommendations based on needs
- Bundle deals and offers

Sample Queries:

- "What's the price of iPhone 15 Pro?"
- "Do you have any gaming laptops under $1500?"
- "Compare Samsung S24 vs iPhone 15"
- "What's the best TV for a small apartment?"

B. Marketing/Promotions Agent

Domain: Deals, discounts, loyalty programs, campaigns

Should Handle:

- Current promotions and sales
- Discount codes and coupons
- Loyalty program information
- Upcoming sales events
- Newsletter subscriptions

Sample Queries:

- "Are there any Black Friday deals?"
- "How do I join the loyalty program?"
- "Do you have any student discounts?"
- "When is your next sale?"

C. Technical Support Agent

Domain: Troubleshooting, warranty, repairs, technical specifications

Should Handle:

- Product troubleshooting
- Warranty information and claims
- Repair services
- Technical specifications
- Setup and installation guidance

Sample Queries:

- "My laptop won't turn on"
- "Is my TV still under warranty?"
- "How do I connect my soundbar to my TV?"
- "The screen on my phone is cracked, can you fix it?"

D. Order & Logistics Agent

Domain: Order status, shipping, returns, refunds

Should Handle:

- Order tracking
- Shipping information and options
- Return and exchange policies
- Refund status
- Delivery scheduling

Sample Queries:

- "Where is my order #12345?"
- "How do I return a product?"
- "Can I change my delivery address?"
- "When will my refund be processed?"

Database Operations Required:

- READ from database (e.g., order status, tracking info)
- WRITE to database (e.g., create support tickets, return requests)

Technical Specifications

Mandatory Requirements

Requirement	Details
Framework	Use any agentic framework (LangGraph, CrewAI, AutoGen, Swarm, or custom)
LLM Provider	Any (OpenAI, Anthropic, Gemini, etc.) - document your choice
Language	Python (preferred) or TypeScript
Interface	React /Next Web UI, with API - minimum viable interface required
Documentation	README with setup instructions and architecture explanation

Optional Enhancements (Bonus Points)

- [ ] Persistent conversation memory
- [ ] Agent performance analytics/logging
- [ ] Sentiment analysis integration
- [ ] Human handoff capability

Mock Data Requirements

Create realistic mock data for:

1. **Product Catalog** (minimum 20 products across categories)
   - Electronics: Phones, Laptops, TVs, Audio
   - Include: name, price, specs, stock status
2. **Promotions Database**
   - At least 5 active promotions
   - Loyalty program details
3. **Order Database**
   - Sample orders with various statuses
   - Tracking information
4. **FAQ/Knowledge Base**
   - Common troubleshooting steps
   - Return/warranty policies

Evaluation Criteria

Criteria	Weight
Architecture Design (Agent + Software)	40%
Code Quality	20%
Creativity & Innovation	20%
Agent Response Quality	20%

Submission Requirements

GitHub Repository

- README.md with setup instructions
- ARCHITECTURE.md explaining your design decisions
- .env.example with required environment variables
- Source code (backend + frontend)

Demo Video (3-5 minutes)

- Demonstrate voice and text input
- Show database read and write operations
- Show LangSmith/Langfuse traces
- Explain your architecture and creative decisions

Questions?

If you have any clarifying questions about the requirements, please email dinal@aigp.health before starting the assessment.

Good luck! We're excited to see your implementation.