SOFTWARE REQUIREMENT SPECIFICATION

AI-Powered Customer Support Platform

(Clothing Brand Edition)

Table of Contents

---

1. Introduction

1.1 Purpose

The purpose of this project is to develop a privacy-first, fully local Customer Support Automation Platform tailored for a clothing brand. The system acts as an intelligent first-line support agent, capable of resolving customer queries through natural conversation, visual defect verification, and automated backend workflows.

1.2 Scope

The platform is designed to operate entirely on consumer-grade hardware (Single GPU) without reliance on third-party cloud APIs (e.g., OpenAI or Google Cloud). It integrates three distinct AI modalities—**Generative Text (LLM)**, **Computer Vision (CNN)**, and **Speech Recognition (STT)**—to provide a seamless "Self-Service" experience.

---

2. System Architecture

The system follows a **Multi-Agent Modular Architecture**, optimized for resource-constrained environments (8GB VRAM). A central controller orchestrates data flow between specialized AI agents to ensure memory efficiency.

- **Controller: FastAPI** (Python) manages API endpoints and agent orchestration.
- **Cognitive Agent (Brain): Llama 3 (8B)** handles conversation logic and intent recognition.

- **Auditory Agent (Ear): Faster-Whisper** handles real-time speech transcription (CPU-optimized).
- **Visual Agent (Eye): ResNet18** handles image classification for fabric defects.

---

3. Functional Requirements

3.1 Core Platform Features

ID	Feature Name	Description
REQ-01	**Authentication**	Secure user login using **OAuth2** standards with **JWT Access Tokens**. Passwords must be hashed using **Bcrypt**.
REQ-02	**Chat History**	Persistent conversation storage in SQL. The system must retrieve the last **20 messages** upon user login to maintain continuity.
REQ-03	**Multilingual Support**	Automatic detection and response in **English** and **Hindi**. No manual language toggle is required.
REQ-04	**Ticket Automation**	The system must automatically generate a support ticket if **Sentiment** is negative or the issue remains unresolved after 3 turns.

3.2 AI Intelligence Modules

ID	Feature Name	Technical Specification
REQ-05	**Contextual NLU**	Powered by **Llama 3 (Quantized)**. The system must retain a context window of the last **10 message pairs**.
REQ-06	**RAG (FAQ)**	Powered by **FAISS (Vector DB)**. The system must retrieve relevant policy documents (e.g., Return Policy) before generating an answer.
REQ-07	**Voice Input**	Powered by **Faster-Whisper**. The system must transcribe microphone input in real-time with **automatic silence detection**.
REQ-08	**Defect Detection**	Powered by **ResNet18**. The system must analyze uploaded images for fabric defects (holes, stains). If confidence is **< 70%**, it must escalate to a human.
REQ-09	**Sentiment Analysis**	Powered by **DistilBERT**. The system must classify every user message as *Positive*, *Neutral*, or *Negative*.

3.3 Business Automation Tools

ID	Feature Name	Description
REQ-10	**Order Status**	**SQL Tool:** The AI shall execute database queries to fetch live status and delivery dates based on the provided Order ID.
REQ-11	**Refund Processing**	**Function Call:** The AI shall trigger the initiate_refund workflow automatically only after the Visual Agent verifies a defect.
REQ-12	**Human Escalation**	**Logic Switch:** The session must transfer to a human agent queue if the user explicitly requests it or if the AI fails to generate a valid tool call.

---

4. Non-Functional Requirements

- NFR-01: Resource Optimization

Total Video RAM (VRAM) usage must remain strictly below 7.8GB to ensure system stability on the RTX 3070 Ti.

- **NFR-02: Latency Targets**
    - Text Generation: **< 3 seconds** per chunk.
    - Speech Transcription: **< 1 second** for short queries.
- NFR-03: Data Privacy

All processing must occur On-Premise (Local). No customer audio, images, or text data shall exit the local network.

- NFR-04: Reliability

The system must prioritize Retrieved Context (RAG) over the LLM's internal knowledge to prevent hallucinations regarding store policies.

---

5. Technical Stack

Component	Technology	Role
Backend	Python 3.10, FastAPI	API & Orchestration
Database	PostgreSQL (Prod)	Relational Data Storage
AI Inference	Llama.cpp (Python Bindings)	Running Quantized LLMs
Vision Model	PyTorch / Tensorflow (ResNet18)	Image Classification
NLP Model	Hugging Face Transformers	Sentiment Analysis

Component	Technology	Role
Vector DB	FAISS (CPU Version)	FAQ Similarity Search
Frontend	React / HTML5 Dashboard	User Interface

---

6. Database Schema Design

The database requires a relational structure to manage users, orders, and support tickets efficiently.

- **users**: Stores authentication details (id, email, password_hash, role).
- **orders**: Stores product transactions (id, user_id, status, delivery_date, image_url).
- **tickets**: Stores support cases (id, user_id, priority, status, sentiment_score).
- **messages**: Stores chat logs (id, ticket_id, sender_role, content, timestamp).