L1 Customer Support Ticketing & Chat System

Document Title
Business Requirements Document – L1 Customer Support Ticketing System (L1-CSTS)
Version: 1.0
Date: Jan 01, 2026
Prepared for: Careerzen Interns

Target Implementation Period: 30 days

1. Executive Summary

A web application that allows customers to raise support tickets with description + screenshots, then chat in real-time with L1 support agents. Agents have a backend dashboard to monitor, pick, and respond to tickets. Designed as a minimal helpdesk MVP.

2. Business Objectives

- Provide customers a structured way to report issues with visual proof (screenshots)
- Enable real-time chat communication tied to each ticket
- Give support agents visibility and control over incoming customer requests
- Demonstrate full-stack development + real-time features in 30 days

3. Scope

In Scope

- Customer registration/login
- Customer: Create ticket (subject, description, category, 1–3 screenshots)
- Ticket auto-generates unique ID and chat channel
- Real-time chat per ticket (WebSocket)
- Agent dashboard: list open tickets, claim ticket, view attachments, chat
- Ticket status flow (Open → In Progress → Resolved → Closed)
- Basic notification (unread messages)

Out of Scope (for 30-day project)

- Multi-agent assignment / teams / departments
- SLA timers / escalation
- Email integration / ticket export (PDF/CSV)
- Knowledge base / canned responses
- Customer satisfaction rating / survey
- Advanced reporting & analytics

4. Stakeholders & Users

- **Customer** – end user raising issue
- **L1 Support Agent** – company employee handling tickets
- **Project Evaluator** – assesses completeness & real-time functionality

5. Functional Requirements

ID	Requirement Description	Priority	Actor
F1	Customer can signup/login	High	Customer
F2	Customer creates ticket: subject, description, product/category, screenshots (1–3)	High	Customer
F3	System assigns ticket ID and creates dedicated chat room	High	System
F4	Real-time chat per ticket (WebSocket)	High	Both
F5	Agent dashboard shows open/new tickets (sortable by date/priority)	High	Agent
F6	Agent can claim ticket, view details + screenshots, reply in chat	High	Agent
F7	Both see message history and typing/online indicators	Medium	Both
F8	Agent can change ticket status; customer sees updated status	Medium	Agent
F9	Basic validation, file type check (png/jpg), error handling	High	System

6. Non-Functional Requirements

ID	Requirement	Target / Constraint
NF1	Technology stack	Backend: Python + FastAPI Frontend: HTML/CSS/JS Real-time: WebSockets
NF2	Authentication	JWT or cookie-based (separate roles)
NF3	File upload	Images only, max 4 MB/file, store locally or free service
NF4	Performance	Ticket list < 3 s, chat latency < 2 s
NF5	Responsiveness	Mobile-friendly layout

NF6	Security	Basic protection (XSS, file validation, auth checks)
NF7	Database	SQLite or PostgreSQL
NF8	Deployment	Local + optional free platform

7. Assumptions & Constraints

- 30 days → prioritize core ticket creation + per-ticket real-time chat
- Single company / no multi-brand support
- No need for very high concurrency
- Screenshots stored simply (no OCR or image analysis)
- Minimal styling (functional > beautiful)

8. High-Level Success Criteria (for project evaluation)

- Customer can create ticket with screenshot and get real-time response
- Agent can monitor tickets, view attachments, chat per ticket
- Status changes reflected to customer
- Stable WebSocket connection during active chat