

# AI Enabled Helpdesk System - Requirements

---

## 1. Purpose

Define product requirements for **Helpdesk-Light**, an MSP-ready helpdesk system that supports:

- Web and email ticket lodgement.
- Ticket resolution workflows.
- A GPT-5.2 powered AI agent (via Semantic Kernel) that helps resolve tickets, communicates with end users, and writes knowledge articles.
- Multi-customer support using email domain based tenant mapping.

## 2. Product Goals

- Reduce first response and resolution times through AI-assisted triage and response drafting.
- Provide a modern and visually polished helpdesk experience for end users and technicians.
- Enable one MSP team to serve multiple customers from one platform.
- Keep deployment simple with ASP.NET Core + Blazor WebAssembly + SQLite.

## 3. Personas

- **End User:** Logs tickets, receives updates, confirms resolution.
- **Support Technician:** Triage and resolves tickets, reviews AI actions.
- **MSP Admin:** Manages customers, domains, SLAs, and system settings.

## 4. Scope

### 4.1 In Scope (MVP)

- Customer and domain setup.
- Ticket creation from:
  - Web portal.
  - Inbound email.
- Ticket lifecycle management (new, triaged, in progress, waiting on customer, resolved, closed).
- AI agent capabilities:
  - Ticket triage and categorization.
  - Suggested and optional auto-generated customer replies.
  - Suggested resolution steps.
  - Knowledge article draft generation from resolved tickets.
- Multi-tenant segmentation by customer.
- Role-based access and audit trail.
- Responsive Blazor WebAssembly frontend.

### 4.2 Out of Scope (MVP)

- Billing and invoicing.
- Full ITSM change management modules.

- On-prem AD/SSO federation (can be phase 2).
- High-scale distributed database design.

## 5. Functional Requirements

### 5.1 Customer and Tenant Management

- **FR-001:** System must allow MSP Admins to create and manage customers.
- **FR-002:** Each customer must support one-to-many mapped email domains (for example `contoso.com`, `support.contoso.com`).
- **FR-003:** Incoming email tickets must be auto-assigned to a customer using sender domain mapping.
- **FR-004:** If domain mapping fails, ticket must be routed to an "Unmapped Queue" for manual assignment.
- **FR-005:** All ticket data must be partitioned by `CustomerId` with role-based visibility.

### 5.2 Ticket Lodgement and Lifecycle

- **FR-006:** End users must be able to submit tickets through a web form with subject, description, priority, and attachments.
- **FR-007:** System must ingest inbound emails and create or update tickets.
- **FR-008:** Email replies containing a ticket reference must append as ticket conversation entries.
- **FR-009:** Ticket states must support: `New`, `Triaged`, `InProgress`, `WaitingCustomer`, `Resolved`, `Closed`.
- **FR-010:** Technicians must be able to assign, reassign, and reprioritize tickets.
- **FR-011:** End users must see ticket history and status updates via portal and email notifications.

### 5.3 AI Agent (Semantic Kernel + GPT-5.2)

- **FR-012:** AI agent must run on new or updated tickets and produce:
  - Category and urgency prediction.
  - Resolution suggestions.
  - Draft user-facing response.
- **FR-013:** AI-generated outputs must be tagged as AI content and stored with provenance metadata.
- **FR-014:** Admins must configure AI mode by policy:
  - `SuggestOnly` (human approval required).
  - `AutoRespondLowRisk` (automatic reply for low-risk categories).
- **FR-015:** Agent must generate draft knowledge articles from resolved tickets when confidence threshold is met.
- **FR-016:** AI actions must be logged with prompt/response metadata and token usage.

### 5.4 Knowledge Base

- **FR-017:** System must store knowledge articles with version history and publication status (`Draft`, `Published`, `Archived`).
- **FR-018:** Technicians must edit AI-drafted articles before publishing.
- **FR-019:** AI agent should use published articles as retrieval context during response generation.

### 5.5 Communication

- **FR-020:** System must send outbound email updates for new comments, status changes, and resolution notices.
- **FR-021:** Users must be able to reply by email and have content synced to the same ticket.
- **FR-022:** System must prevent email loop storms through message-id tracking and deduplication.

## 5.6 Security and Access

- **FR-023:** Role support: `EndUser`, `Technician`, `MspAdmin`.
- **FR-024:** API and UI must enforce customer boundary checks on all ticket and article operations.
- **FR-025:** Sensitive secrets (mail credentials, model keys) must be externalized to secure configuration.

## 6. Non-Functional Requirements

- **NFR-001:** Web UI should achieve Lighthouse accessibility score  $\geq 90$  for key pages.
- **NFR-002:** P95 ticket list page load time  $\leq 2.5$  seconds with 10k tickets in SQLite.
- **NFR-003:** P95 API response time  $\leq 500$  ms for standard CRUD operations.
- **NFR-004:** Background email and AI processing should be resilient with retry and dead-letter handling.
- **NFR-005:** Audit logs must be immutable and retained for at least 12 months.
- **NFR-006:** System should support backup/restore of SQLite and file attachments.
- **NFR-007:** UI must be responsive and visually polished on desktop and mobile.

## 7. UX and Visual Requirements ("Beautiful" Experience)

- Clean, modern design language with strong typography hierarchy, generous spacing, and high contrast.
- Branded color system with light neutrals and accent colors for state signaling.
- Dashboard cards for queue health, SLA risk, and AI activity.
- Smooth but minimal motion for page transitions, loading states, and timeline updates.
- Mobile-first layouts for ticket creation and conversation views.
- Consistent component library usage across Blazor pages.

## 8. Reporting and Metrics

- Ticket volume by customer, category, and channel (web vs email).
- First response time and resolution time per customer.
- AI assist metrics:
  - Suggestion acceptance rate.
  - Auto-response rate.
  - Article draft acceptance rate.
  - Deflection estimate.

## 9. Acceptance Criteria (MVP)

- Admin can create two customers with distinct domains and tickets are correctly auto-mapped by sender domain.
- End user can submit tickets from web and by email.
- Technician can process tickets through full lifecycle.
- AI agent produces triage and response drafts for new tickets.
- AI-generated knowledge article draft can be reviewed and published.

- End users receive and can reply to ticket emails, preserving thread continuity.
- UI is responsive, accessible, and visually cohesive across core screens.