

# Specifications Document: AI-Powered Maintenance Technician Chat System

## 1. Introduction

### 1.1 Purpose

This document outlines specifications for an AI chat system that enables maintenance technicians to:

- Communicate with an AI assistant powered by DeepSeek API
- Share individual conversations with team members
- Participate in group conversations for collaborative problem-solving

### 1.2 Scope

The system will:

- Provide real-time AI chat interface for maintenance queries
- Support conversation sharing between users
- Enable group chat functionality for team collaboration
- Maintain conversation history with search capabilities

### 1.3 Key Benefits

- ✓Instant access to technical knowledge through AI
- ✓Improved team collaboration through shared conversations
- ✓Knowledge retention through conversation history
- ✓Reduced resolution time for complex maintenance issues

## 2. User Roles

Role	Responsibilities	System Needs
Maintenance Technician	Performs repairs, uses AI for support	AI chat, conversation sharing
Team Lead	Oversees multiple technicians	Group chat, conversation monitoring
Knowledge Manager	Maintains technical content	Conversation archiving, analytics
System Admin	Manages user access	User permissions, API configuration

## 3. Functional Requirements

### 3.1 Core Chat Functionality

- **FR1:** Real-time chat interface with DeepSeek API integration
- **FR2:** Context-aware conversation history
- **FR3:** Support for technical document uploads (PDFs, images)
- **FR4:** Markdown formatting in messages (code snippets, lists)

### 3.2 Conversation Sharing

- **FR5:** Ability to share individual conversations via link
- **FR6:** Permission controls for shared conversations
- **FR7:** Notification system when conversations are shared
- **FR8:** Visual indicators for shared conversation status

### 3.3 Group Conversations

- **FR9:** Create persistent group chats with multiple participants
- **FR10:** @mention functionality in group chats
- **FR11:** Group-specific conversation history
- **FR12:** Role-based permissions in groups (read/write/admin)

### 3.4 Advanced Features

- **FR13:** Conversation tagging and categorization
- **FR14:** Search across all conversations
- **FR15:** Export conversations (PDF, TXT formats)
- **FR16:** Integration with ticketing systems (optional)

## 4. Technical Specifications

### 4.1 System Architecture

1. **Frontend:** Web and mobile interfaces (React/React Native)
2. **Chat Service:** Node.js server handling real-time communication
3. **AI Integration:** DeepSeek API connection layer
4. **Database:** MongoDB for conversation storage
5. **Notification Service:** WebSocket/Push notifications

### 4.2 Data Model

Copy

```
Group {id, name, members[], conversationId}
```

### 4.3 API Integration

- DeepSeek API for AI responses
- Authentication via JWT tokens
- Rate limiting and usage monitoring

## 5. User Stories

### 5.1 Individual Technician Use

- "As a technician, I want to ask the AI about equipment specs so I can quickly solve problems"
- "As a field worker, I need to share my conversation with a supervisor for approval"

### 5.2 Team Collaboration

- "As a team lead, I want to create a group chat for complex equipment issues"

- "As a team member, I need to be notified when added to important conversations"

### **5.3 Knowledge Management**

- "As a knowledge manager, I want to search past conversations to build FAQs"
- "As an admin, I need to monitor API usage to control costs"