Business Requirements Document (BRD)

CRM Portal

Developer: [Your Name] | **Timeline**: 9 Weeks

---

1. Introduction

1.1 Purpose

To build a **Technical Support CRM Portal** from scratch, designed for a technical company to manage issue resolution efficiently. The system aims to improve collaboration, streamline ticket workflows, and **reduce ticket resolution time from 48 hours to 30 hours** through structured tracking, SLA enforcement, and dynamic prioritization.

1.2 Scope

Core Features (MVP - Minimum Viable Product)

- **Role-based UI** for **Customers**, **Support Teams**, and **Technical Teams** (Team Lead and Team Member UIs are distinct).
- **Basic Ticket Lifecycle**: Create, Assign, Update, Resolve, Close.
- **Kanban Board** for real-time tracking.
- **SLA-driven Deadlines** with escalation alerts.
- **Multi-team Ticket Sharing** for joint issue resolution.
- **Live Chat for Internal Teams.**
- **Automated Priority Escalation** based on backlog and ticket age.
- **Ticket Reassignment** by Team Leads to correct misrouted tickets.
- **Member Dashboard** for tracking individual work.
- **Team Lead Dashboard** for monitoring team workload and performance.
- **Internal Ticket Comments**: A chronological log of updates made by each team member for transparency.
- **Custom Ticket Statuses & Workflows**: Allow admins to define additional statuses beyond *Open → In Progress → Resolved → Closed*.
- **Advanced Filtering & Search**: Filter by SLA breach, unresolved blockers, shared tickets, and priority.
- **Audit Logs & Change History**: Keep track of ticket modifications for accountability.
- **Real Time Notification**: Notification can be send in real time to team member regaring changes and work

Future Enhancements (If Time Permits)

- **Third-Party Integrations** (GitHub, Slack).
- **Advanced Analytics & Predictive Insights** (AI-based ticket routing).
- **Smart Ticket Assignment** using machine learning to route tickets efficiently.
- **Custom SLA Configurations** per ticket type.
- **Public Knowledge Base** for self-service solutions.
- **Recurring Tickets & Scheduled Reports**: Automate periodic tasks (e.g., *Monthly System Maintenance* tickets).
- **Performance Monitoring Dashboard**: Track real-time system health (API response times, ticket queue load).

---

2. Stakeholders & User Roles

Role	Responsibilities
Customer	Submit Issue, track status, and communicate updates.
Support Team	Triage tickets, assign them to technical teams, set priorities.
Technical Team	Work on assigned tickets, add comments, and attach files.
Team Lead	Manage team workloads, enforce SLAs, and configure system settings. Share tickets with multiple teams and reassign misrouted tickets.
Admin	Configure teams, permissions, and system rules.

3. Functional Requirements

- **Ticket Management & Workflows**
  - Full ticket lifecycle: *Open → In Progress → Resolved → Closed*.
  - Creation of **sub-tickets** to manage dependencies before closing a parent ticket.
  - **Multi-team ticket sharing** for collaborative issue resolution.
  - **Team Leads can reassign misrouted tickets** to the correct team.
  - **Custom ticket statuses** beyond default lifecycle for flexible workflows.
- **SLA Enforcement & Automation**
  - SLA-driven deadlines with **automated alerts at 75% expiry and breach**.
  - **Priority escalation mechanisms** based on backlog volume and ticket age.
  - **Auto-close parent tickets** when all subtasks are resolved.
- **Internal Collaboration & Communication**
  - **Chronologically ordered ticket comments** to track updates transparently.
  - **Audit logs and change history** to maintain accountability.
  - **Internal team discussions** separate from customer-visible updates.
  - **Threaded Comments in Tickets:** Organize discussions by topic/User
  - **In-app Messaging for Team Members:** Allow real-time communication within the CRM.
  - **Tagging & Mentions:** Notify relevant users with @mentions in ticket comments.
- **Customizable Workflows & Filters**
  - Team leads/admins can **configure custom ticket statuses**.
  - **Advanced filtering options**: SLA breaches, unresolved blockers, shared tickets, priority levels.
- **Notification System**
  - **Notification alerts for new ticket assignments, SLA breaches, and comments**.
  - **Real-time notifications** for critical updates.

4. Non-Functional Requirements

- **Performance**
  - The system should load dashboards in under 2 seconds even with 100+ concurrent users.
- **Security**
  - Implement Role-Based Access Control (RBAC) to ensure only authorized users can access certain features.
  - AES-256 encryption for storing sensitive data.
- **Usability**

    - The UI should be intuitive and responsive, designed with collapsible admin tools for efficiency.
- **Scalability**
    - The system must support 5,000+ tickets per month and allow for the dynamic addition of new teams without performance degradation.

---

4. Additional Functional (For Future)

- **Incident & Problem Management Module**
    - Link related tickets under a major incident category.
    - Track recurring problems and apply preventive measures.
- **Enhanced Knowledge Base & Self-Service**
    - Recommend relevant knowledge base articles based on ticket content.
    - Customers can find solutions before submitting tickets.
- **Automated Response Suggestions**
    - Provide pre-written responses based on ticket type and urgency.
    - Reduces response time for common issues.
- **Integration with Calendar & Scheduling**
    - Sync ticket deadlines with Google Calendar/Outlook.
    - Team members can schedule maintenance or follow-ups.
- **Dynamic Workload Distribution**
    - Automatically assign tickets based on the current workload of team members.
    - Prevents overload and ensures even distribution.
- **Advanced Reporting & Analytics**
    - **Customizable Performance Dashboards:** Allow users to set up personalized analytics.
    - **Heatmaps & Bottleneck Analysis:** Identify areas where tickets get stuck.
    - **Customer Satisfaction Tracking:** Collect feedback on resolved tickets.
- **Enhanced UI**
    - **Drag & Drop Ticket Prioritization:** Reorder tasks easily for improved workflow management.
    - **Quick Action Shortcuts:** Implement keyboard shortcuts for common actions.
    - **Dark Mode Support:** Provide a UI theme toggle for accessibility.

---

5. Technology Stack

Component	Tools	Rationale
Frontend	React + Tailwind CSS	Lightweight, responsive design.
Backend	Node.js + Express.js	REST API, scalable microservices architecture.
Database	MongoDB (NoSQL)	Flexible schema for dynamic tickets/teams.
Deployment	Vercel (Frontend) + Render (Backend)	Cloud hosting with free-tier support.

---

6. Workflow Diagram

```
1. Customer Submits Ticket → 2. Support Team Assigns → 3. Technical Team
Resolves → 4. Admin Closes Ticket
```

---

7. Timeline (9 Weeks)

Phase	Weeks	Deliverables
Core Setup	1-2	Authentication, database setup, basic UI.
Ticket Management	3-4	Ticket lifecycle, sub-tickets, escalations.
Team Features	5-6	Role-based UI, workload monitoring, dashboards.
Collaboration Tools	7	In-app messaging, ticket comments, notifications.
Testing & Compliance	8	Load testing, GDPR compliance, bug fixes.
Deployment & Review	9	Final optimizations, documentation, deployment.

---

8. Appendix

- **SLA (Service Level Agreement)**: A predefined set of rules dictating response and resolution times for tickets based on priority.
- **RBAC (Role-Based Access Control)**: A security model that restricts system access based on predefined roles and permissions.
- **Kanban Board**: A visual workflow management tool that helps track ticket progress in real time.
- **Sub-Tickets**: Smaller tasks linked to a primary (parent) ticket to break down complex issues into manageable steps.
- **Audit Logs**: A detailed record of all changes made to a ticket, including updates, comments, and reassignments, ensuring transparency and accountability.
- **Incident & Problem Management:** A process to manage recurring or major incidents by linking related tickets and applying preventive measures.
- **Dynamic Workload Distribution:** An automated method to assign tickets based on team availability to balance workload effectively.