

# PRD: Customer Relationship Management (CRM) software

## Team

Member	ID
Tasneem Abdullah Aurnobb	2111752642
Fariza Tasnia Ahmed	2232072042
Md. shazzad Haque	2131750042

## 1.0 Problem Statement

Problems surrounding CRM software may be divided into two broad categories: those faced by companies **not using** them, and those faced by companies **who do**.

1. **Non-users:** Many small businesses (e.g., a local poultry supplier) lack the technical training to operate existing CRMs. They rely on Excel sheets and manual notes, which are disorganized and lead to lost leads and poor ticket follow-ups.

2. **Existing users:** For companies which do adopt a CRM, their teams often find the process tedious, having to make constant manual updates through complex interfaces.

We have also read testimonials from regular users of existing CRM software, so that we may aim to tackle them:

- “Getting reliable exports of all CRM data for backups is nearly impossible. I would like an option to download my activity/contact lists.”
- “Way too complex.”

## 2.0 Proposed Solution

The proposed solution is a **multi-tenant CRM** aimed at balancing **simplicity and intuitive features for new users** with **efficiency and automation for experienced teams**. It will enable small, non-technical businesses, such as poultry farms, to manage customers without training or complex configuration.

The system consolidates all customer interactions including emails, tickets, and calls, into one platform, supports **open-source ticketing for customer support**, and allows users to make **updates directly through chat-based interfaces (Telegram)**. Together, these features reduce manual workload, ensure proper record-keeping, and make the CRM accessible to organizations of any scale.

### 2.1 Goals

- **Unified Platform:** Consolidate customer data, sales, and support under one system.
- **Ease of Use:** Offer intuitive features and simple navigation requiring **little to no formal training**.
- **Data Isolation:** Provide robust **multi-tenant data isolation** between organizations.
- **Integration:** Integrate seamlessly with everyday tools (e.g., Gmail, LinkedIn, Calendar).
- **Intelligent Control:** Allow complete conversational control through a **Telegram bot**.

### 3.0 Users

- **Sales Representatives:** manage leads, log activities, follow up on deals.
- **Marketing Representatives:** view contact history and lead sources.
- **Team Heads / Managers:** oversee pipelines, assign tasks, view analytics.
- **Admin / Owner:** manage users, permissions, and configurations.
- **Customers (End Users):** submit and track support requests through the company portal or app.

## 3.1 User Stories

### 1. Sales Rep

As a sales rep, I want to see all emails, calls and meetings **under one customer timeline** so that I always know the latest status before the next meeting. I also want to **update contact notes via Telegram**, so that I **don't need to open the CRM web or mobile app** for small updates.

### 2. Support Agent

As a Support Agent, I want to **view all support tickets related to a customer**, so that I can respond with full context. I also want to **see expected resolution times**, so that I can prioritize urgent cases.

### 3. Admin

As an Admin, I want to **create and manage user accounts with roles**, so that I can control who can view or edit customer data.

### 4. Customer (End-Client)

As a Customer, I want to **submit a complaint or support request easily through the website or app**, so that I can get help.

## 4.0 Requirements

*\*follows the MoSCoW method of prioritization – Must Have, Should Have, Could Have, Won't Have*

### **A. Core CRM functions - contact management, lead management, sales pipeline management, activity tracking (Must Have)**

- Users should be able to create, **add, update, delete, search and sort** contacts (persons and organisations) leads, and deals (pipeline items).
- Each record must belong to a tenant and must not be accessible to other tenants.
- The system shall log Activities (e.g. calls, follow-up emails) linked to these records.

### **B. Authentication (Must Have)**

- The system must authenticate users securely (e.g. username/password, token-based sessions).

### **C. Authorization (Must Have)**

- The system shall support **Role-Based Access Control (RBAC)**.
- Roles include - Admin, User (Head, Rep, Support).
- The roles dictate what actions (e.g. add/update/delete) are permitted for that user.

### **D. Multi-Tenant Architecture (Must Have)**

- The system should securely support multiple **tenants**, ensuring **proper isolation** of tenant data.
- Every data record (Contact, Lead, Activity) must have its tenant ID, and there will be no leakage in UI, API, or logs.
- A greater security measure enforced by having **one-schema-per-tenant** on the same database, ensuring schema level isolation (should have).

#### **E. Audit Logging (Should Have)**

- The system should log critical actions (create, update, delete) on key entities (Contacts, Deals, Tickets), including who performed the action and when.

#### **F. Customisability (Should Have)**

- The system should allow each tenant to define and store custom fields (e.g. custom text) on core CRM objects (Contacts, Leads, Deals) without code changes.

#### **G. Data Export (Could Have)**

- The system shall allow a tenant admin to export their CRM data including Contacts, Leads, Deals, and Activities into a human-readable PDF report. The export should include summary statistics (e.g., number of contacts, open deals, unresolved tickets) and time of generation. Each export must be limited to the authenticated tenant's data only.
- *\*This requirement has been included as a Could-Have because it has been listed as a nice-to-have feature (not offered by existing software) by a CRM user on Reddit.*

#### **H. Mobile and Web platforms as a Unified System**

- The system shall provide both a web client and a mobile client that use the same backend API and share the same authentication model.
- Data created/updated on one client must be visible to the same tenant's users on the other client after refresh.
- Session/auth must work consistently across platforms.

#### **I. Ticketing System (Must Have)**

- The system must integrate with an open-source ticketing solution, and allow ticket creation from within the CRM.

- Each ticket will be linked to an existing Contact/Lead.
- Ticket information such as status, priority, assigned agent, timestamps should be stored in the CRM.
- The system *should* provide an expected resolution time to clients.
- Tickets should be linked to specific contacts.

#### **J. In-app calls (Must Have)**

- The system shall allow users to initiate calls from within the CRM using an open-source VoIP integration.
- Call recordings must be stored and linked to the relevant Contact.
- The system *should* ask for permission (e.g. using a Push notification) before recording calls (Should Have).

#### **K. Gmail/LinkedIn Integration (Should Have)**

- The system should allow a user to connect external services (e.g. Gmail, LinkedIn) to enrich CRM data.

#### **L. Records of all Customer and Lead Interactions (Must Have)**

- The system must maintain a unified record for each Contact/Lead. This should include logged calls, emails support tickets, Telegram actions, and meeting summaries.
- Meeting summaries *may* be generated using an AI notetaker based on uploaded recordings or transcripts.

#### **M. Automated tasks in a conversational format - MCP server and Telegram bot integration (Must Have)**

- The system should support conversational updates via a Telegram bot.
- Users should be able to perform common actions (e.g. add a contact, log a note, update a deal stage) without opening the web or mobile app.

- For sensitive actions (e.g. delete account, change ownership), the bot must ask for confirmation.

## 5.0 Success Metrics

The CRM will be considered successful if:

- Core CRUD operations are functional and responsive across both web and mobile.
- New (including non-technical) users are able to create their first contact without guidance.
- Role-based permissions correctly restrict unauthorized actions per role.
- No data leakage occurs across tenants in isolation testing.
- VoIP and Gmail integrations complete at least one successful end-to-end operation with correct data reflection in CRM.
- Telegram bot handles supported commands reliably.
- Phase-2 automation executes chat-to-update commands in <10 seconds.

## 5.1 Phases

- Phase 1 deliverables: CRUD, multitenancy, ticketing, VoIP demo, Gmail sync.
- Phase 2 deliverables: MCP API, Telegram bot, automated updates.

## 6.0 Open Questions

1. Which ticketing and VoIP APIs to use to ensure API stability and documentation quality?
2. How to design the database to ensure data isolation between tenants – e.g. single schema with tenant ID OR one schema per tenant?
3. How to avoid privacy breaches while recording customer calls?