

# Rawasy WhatsApp CRM SaaS - Product Requirements Document (PRD)

---

## 1. Project Overview

Rawasy WhatsApp CRM is a multi-tenant SaaS platform for customer relationship management, marketing automation, and customer support via WhatsApp Cloud API. It provides robust customer, ticket, campaign, and user management, with strong data isolation and extensibility.

---

## 2. Architecture & Technology

- Backend: Java 17, Spring Boot, JPA/Hibernate, MySQL
  - API: RESTful, secured with JWT and custom app key middleware
  - Frontend: (Not included in this repo, but API-ready for SPA/React/Angular)
  - Multi-tenancy: Data isolation at the entity level (Tenant, User, Customer, etc.)
  - Extensible: Modular codebase with DTOs, services, and repositories
- 

## 3. Key Features

- Multi-tenant customer, ticket, and campaign management
  - WhatsApp Cloud API integration for messaging and automation
  - Role-based access: Admin, Marketing, Sales, Support, Bot, etc.
  - Marketing campaigns with contact upload, consent, and template management
  - Automated bot replies using tenant-specific lookup tables
  - Subscription and billing management (planned)
  - Comprehensive reporting and analytics
  - Secure authentication and application-level middleware (X-App-Key)
- 

## 4. Data Model Overview

- **Tenant:** Organization, with WhatsApp API config, users, customers, etc.
- **User:** Linked to tenant, with roles and department/supervisor structure
- **Customer:** Profile, consent, and marketing group membership
- **Campaign:** WhatsApp marketing, linked to contacts and templates
- **Ticket/Quotation:** Support and sales flows, per tenant
- **LookupTable:** Dynamic, per-tenant, for bot categorization

- **MessageTemplate:** WhatsApp Meta-approved, per tenant
  - **MarketingGroup:** Segments for campaign targeting
  - **Subscription:** Tenant plan and billing (future)
- 

## 5. Security & Middleware

- JWT-based authentication for users
  - Custom **X-App-Key** middleware (Spring Interceptor) for application-level access control
  - Profile-based configuration (application-development/production.properties)
  - Validation on DTOs (e.g., password length, email format)
  - Role checks and tenant isolation in all service layers
- 

## 6. Functional Requirements

1. Multi-tenant WhatsApp messaging and campaign management
  2. Automated bot replies using per-tenant lookup tables
  3. Ticket and quotation management, segregated by tenant and role
  4. User, department, and role management (including supervisor chains)
  5. Marketing contact upload (CSV), consent management, and group targeting
  6. WhatsApp Meta message template management
  7. Subscription and billing lifecycle (future)
  8. Comprehensive reporting and analytics
  9. Feedback and review collection per tenant
  10. Notifications for assignments and campaign status
  11. Secure onboarding/offboarding and GDPR compliance
- 

## 7. Non-Functional Requirements

- Scalable, cloud-ready, and modular backend
  - Strong data isolation and security for all tenants
  - Performance: Fast bot replies, reliable campaign delivery
  - Rate limiting/throttling for WhatsApp API compliance
  - Audit logs and error handling (middleware/global exception handler)
  - Profile-based configuration for dev/prod environments
- 

## 8. User Stories

- As a marketing user, I can upload contacts and run WhatsApp campaigns to consented customers

- As a tenant admin, I can manage users, roles, and message templates
  - As a support/sales user, I can manage tickets and quotations for my tenant
  - As a platform admin, I can monitor tenants, subscriptions, and system health
  - As a customer, I receive relevant WhatsApp messages and can opt in/out of marketing
  - As a bot, I reply to customer messages using tenant-specific lookup tables
- 

## 9. Glossary

- **Tenant:** A client organization using the CRM platform
  - **Campaign:** A WhatsApp marketing effort targeting a group of contacts
  - **Message Template:** Pre-approved WhatsApp message format
  - **Lookup Table:** Dynamic, per-tenant data for bot categorization
  - **Marketing Group:** Segment of customers for targeted campaigns
  - **X-App-Key:** Custom header for application-level API access
  - **Role:** User's permission level (Admin, Marketing, Sales, Support, Bot)
- 

## 10. Notes & Future Work

- Subscription and billing integration is planned but not yet implemented
- Frontend SPA and mobile apps can be built on top of the REST API
- Further extensibility for integrations (e.g., payment gateways, analytics)
- More granular audit logging and compliance features

Document generated/refined on May 15, 2025. For latest updates, see the README.md and docs/ folder.

---