

# CRM Development Documentation Suite

This document contains three sections: **PRD**, **MVP Plan**, and **Development Helper / Architecture Guide** for building a production-grade CRM system based on our previous discussions.

---

## 1 Product Requirements Document (PRD)

### Objective:

Build a **scalable, configurable, multi-tenant CRM** that supports Leads, Contacts, Accounts, Deals, Pipelines, Activities, Automation, Reporting, and Integrations.

### Target Users:

- Sales teams (B2B/B2C)
- Marketing teams
- Customer success teams

### Features:

#### Dashboard

- Aggregated KPIs
- Sales pipeline overview
- Activity summary

#### Leads

- Source tracking
- Configurable status/stage workflow
- Lead assignment & ownership
- Events: LeadCreated, LeadQualified, LeadConverted

#### Contacts

- Linked to Accounts
- Roles: Decision Maker, Influencer, Technical, Finance
- Multi-contact deal support

#### Accounts

- Type: B2B / B2C
- Industry & size metadata
- Lifecycle status

## **Deals**

- Linked to Accounts & Contacts
- Multi-contact support
- Pipeline & stage tracking
- Amount, Probability, CloseDate

## **Pipelines & Forecasts**

- Configurable pipeline stages per tenant
- Probability per stage
- Forecast aggregation

## **Activities & Calendar**

- Immutable activity logs
- Task / meeting / email tracking
- Integration with Task Service

## **Automation & Templates**

- Workflow engine: Trigger → Condition → Action
- Email / task / activity templates
- Tenant-configurable automation rules

## **Reports & Dashboards**

- Sales KPIs
- Stage progress
- Custom dashboards

## **Products, Quotes, Invoices**

- Product catalog linked to deals
- Quote generation from deals
- Invoice generation from deals/accounts

## **Integrations**

- Task backend
- Email / SMS notifications
- Payment / accounting integrations

## **Settings & Help**

- Roles, permissions, RBAC
- Tenant-specific configuration
- Knowledge base / help section

## Non-Functional Requirements

- Multi-tenancy with row-level isolation
  - Event-driven architecture
  - Scalable to 100K+ users
  - 24/7 uptime, production-grade reliability
  - Audit logging and immutable history
  - Security: JWT, OAuth, encryption, OWASP protections
  - Extensible for future microservices
- 

## 2 MVP Plan

### Goal:

Build the **minimum viable CRM** that is fully functional, testable, and extensible.

### MVP Scope:

- Dashboard: Basic metrics
- Leads: CRUD, status change, lead conversion
- Accounts: CRUD
- Contacts: CRUD, link to accounts
- Deals: CRUD, pipeline stages
- Activities: Log calls, emails, meetings
- Automation: Basic trigger → action workflow
- Multi-tenancy: Row-level data isolation
- Role-based access: Admin, Sales User

### Excluded in MVP:

- Advanced reports/dashboards
- Full integration with external Task Service (simplified mock)
- Quotes, invoices, product catalog
- Complex automation templates
- Advanced RBAC / feature flags

### MVP Flow Example:

1. Create Lead → qualify → convert → create Account & Contact → create Deal → move through pipeline → log activity → trigger workflow
2. Dashboard shows summary of pipeline stages and total deals
3. Basic task reminders simulated locally

### MVP Tech Stack:

- Backend: .NET 8, EF Core, MSSQL
- Frontend: React, React Query / Zustand for state

- Auth: Central system JWT + RBAC
  - Event system: Domain events + outbox pattern for async
- 

## 3 Development Helper / Architecture Guide

### Core Principles

- Domain-driven design (DDD)
- Modular Monolith (future microservices possible)
- Event-driven: domain events, outbox pattern, async processing
- Tenant-configurable pipelines, stages, automation
- Immutable activity logs and history
- Multi-contact deal support
- Security: RBAC, tenant isolation, OWASP best practices

### Backend Architecture

```
CRM.Api
├── Controllers (v1, v2)
├── Application Layer (Commands / Queries / Handlers)
├── Domain Layer (Aggregates / Events / Policies)
├── Infrastructure (EF Core, MSSQL, Outbox, Messaging)
└── Shared (DTOs, Errors, Enums)
```

### Aggregates / Entities

- LeadAggregate: Source, Status, Owner
- AccountAggregate: Type, Industry, LifecycleStatus
- ContactAggregate: Role, AccountId
- DealAggregate: AccountId, PipelineStage, Amount, Probability, Multi-contact mapping
- PipelineAggregate: Stages, Tenant config
- ActivityAggregate: Immutable logs
- WorkflowAggregate: Triggers, Conditions, Actions

### Event Flow

1. User action → Domain command
2. Domain logic → Domain event
3. Event stored in Outbox
4. Background worker → async processing → external services

### Multi-Tenancy

- Shared DB + row-level isolation
- TenantId filter in EF Core global query

- Custom fields stored in Key-Value / JSON table

## Frontend Architecture

```
/src
  /app
    /features
      /leads
      /contacts
      /deals
      /activities
      /pipelines
      /automation
    /entities
    /hooks
    /services
    /ui
    /state
    /auth
```

- React Query for server state - Zustand for UI state - Role-based UI rendering - Sidebar mirrors bounded contexts

## Security & Best Practices

- JWT / OAuth auth
- RBAC for menus and actions
- Row-level tenant filtering enforced in backend
- HTTPS everywhere, TLS 1.2+
- Secrets stored in KeyVault / Env
- Audit logs for all entity changes

## Event-Driven Automation

- Trigger → Condition → Action
- Tenant-configurable via DB
- Async, retryable
- Never mutate domain directly

## Scaling & Performance

- Horizontal scaling for API + workers
- Redis cache for permissions & lookups
- CQRS for reports / dashboards
- Partition large tables (Activities, Audit, Outbox)
- Background workers handle tasks, reminders, notifications

## DevOps & Deployment

- CI/CD with PR validation, tests, security scans
- Rolling deployments + feature flags
- Observability: logs, metrics, tracing
- Environment separation: Dev → QA → Staging → Prod

## Future-Proofing

- Modular monolith → microservices extraction
- Workflow, Task, Notification services separate
- Event contracts versioned
- Custom fields & tenant configuration remain flexible

---

## Sidebar Module Map to Backend

Sidebar	Backend Aggregate	Notes
Dashboard	Reporting projections	Read-only metrics
Leads	LeadAggregate	Events: LeadCreated, LeadQualified, LeadConverted
Contacts	ContactAggregate	Multi-contact deals
Accounts	AccountAggregate	B2B/B2C, Industry, LifecycleStatus
Deals	DealAggregate	Multi-contact, Pipeline stage tracking
Pipelines	PipelineAggregate	Tenant configurable, stage rules
Forecasts	Reporting projections	Stage probability aggregation
Activities	ActivityAggregate	Immutable logs, linked to Lead/Deal/Contact
Calendar	ActivityAggregate	Scheduling, integration with tasks
Emails	ActivityAggregate	Email logs & triggers
Automation	WorkflowAggregate	Trigger → Condition → Action engine
Templates	WorkflowAggregate	Email / Task templates
Reports	Reporting projections	Aggregated read-side queries
Dashboards	Reporting projections	Visual KPI & charts
Products	ProductAggregate	Linked to Deals (future)
Quotes	QuoteAggregate	Linked to Deals / Products
Invoices	InvoiceAggregate	Linked to Deals / Accounts
Integrations	External integration adapters	Task, Email, Payments

Sidebar	Backend Aggregate	Notes
Settings	TenantConfig / Role	Roles, permissions, RBAC
Help	KnowledgeBase	Static or dynamic content

**This three-part document acts as:** 1. PRD for business understanding 2. MVP plan for dev scope 3. Architecture + Development Helper for implementation guidance

It ensures developers, architects, and QA have a **single source of truth** for building, scaling, and maintaining the CRM system.