Synchro PM

Multi-Tenant SaaS Project Management System

Complete Architecture Documentation

Version 1.0

2024

1. Executive Summary

Synchro PM is a comprehensive, production-ready multi-tenant SaaS project management system designed as a modern alternative to Jira, Redmine, and Asana. Built on Next.js 15 with App Router, it provides a complete solution for organizations to manage projects, tickets, team collaboration, and time tracking with enterprise-grade features.

The system features a modern, responsive UI with dark/light theme support, real-time collaboration capabilities, and comprehensive API endpoints for all operations. It is optimized for deployment on Vercel with serverless architecture and edge function support.

2. Technology Stack

The system is built on a modern, Vercel-compatible technology stack:

Layer	Technology	Purpose
Frontend	Next.js 15 + React 19	App Router, Server Components
Styling	TailwindCSS 4 + ShadCN UI	Modern design system
State	Zustand + TanStack Query	Client & server state management
Backend	Next.js API Routes	Serverless functions
Database	PostgreSQL (Neon/Supabase)	Primary data store
ORM	Prisma	Type-safe database access
Auth	NextAuth.js	JWT sessions, OAuth
Charts	Recharts	Data visualization
DnD	@dnd-kit	Drag and drop

Table 1. Technology Stack Overview

3. Multi-Tenant Architecture

The system implements organization-based multi-tenancy where each organization has complete data isolation. Users can belong to multiple organizations with different roles in each. This design enables true SaaS functionality with subscription-based access control.

3.1 Data Isolation Model

All tenant data is scoped at the organization level through foreign key relationships. The Prisma schema enforces cascade deletes to maintain referential integrity while allowing organizations to manage their own data lifecycle. Each organization has isolated projects, tickets, members, and settings.

3.2 Role-Based Access Control

The system implements a 5-tier role hierarchy for fine-grained permissions:

Role	Permissions	Use Case
OWNER	Full organization control, billing, delete org	Organization founder
ADMIN	Manage members, create/delete projects, approve time	Team administrator
MANAGER	Create projects, manage team assignments	Project manager
MEMBER	Create/edit tickets, log time, comment	Team member
GUEST	Read-only access to assigned projects	External stakeholder

Table 2. Role-Based Access Control Hierarchy

4. Database Schema

The Prisma schema defines 30+ models covering all aspects of project management. The schema is designed for PostgreSQL with proper indexing for performance and cascade delete rules for data integrity.

4.1 Core Models

Category	Models	Description
User Management	User, Account, Session	Authentication and session management

Multi-Tenancy	Organization, OrganizationMember, OrganizationInvitation	Tenant isolation and membership
Projects	Project, ProjectMember, Milestone, Module	Project hierarchy and organization
Tickets	Ticket, TicketDependency, KanbanColumn, Sprint	Work items and workflow
Collaboration	Comment, Attachment, ChatChannel, ChatMessage	Team communication
Time Tracking	TimeLog	Hours logging and approval
Automation	CustomField, CustomFieldValue, Workflow, WorkflowStep	Extensibility and automation
Billing	Subscription, Invoice	Stripe-ready billing

Table 3. Database Model Categories

5. API Endpoints

The system provides RESTful API endpoints for all operations, implemented as Next.js API routes. All endpoints require authentication via NextAuth.js JWT sessions and enforce role-based access control.

Endpoint	Methods	Description
/api/organizations	GET, POST	List and create organizations
/api/projects	GET, POST, PUT, DELETE	Full project CRUD operations
/api/tickets	GET, POST, PUT, DELETE	Ticket management with filters
/api/comments	GET, POST, DELETE	Ticket comments and replies
/api/time-logs	GET, POST, PUT	Time tracking and approval
/api/notifications	GET, POST, PUT	User notifications
/api/attachments	GET, POST, DELETE	File uploads and management

Table 4. API Endpoints Summary

6. Frontend Views

The user interface is built as a single-page application with multiple views accessible through sidebar navigation. Each view is optimized for its specific use case while maintaining consistent design language.

View	Features
Dashboard	Statistics cards, productivity charts, sprint burndown, activity feed, priority issues
Projects	Project list with progress, status indicators, member counts, due date tracking
Kanban Board	Drag-and-drop tickets, multiple columns, priority indicators, assignee avatars
Gantt Chart	Timeline visualization, project/task hierarchy, progress bars, zoom controls
Calendar	Monthly calendar, event types, upcoming events sidebar, date selection
Team Chat	Channel list, direct messages, message threading, user presence, file attachments

Settings	Organization settings, billing management, integration configuration

Table 5. Frontend Views and Features

7. Deployment Configuration

The system is optimized for Vercel deployment with zero-configuration setup. The following environment variables are required for production deployment:

Variable	Required	Description
DATABASE_URL	Yes	PostgreSQL connection string
NEXTAUTH_SECRET	Yes	JWT secret (32+ chars)
NEXTAUTH_URL	Yes	Production URL
GOOGLE_CLIENT_ID	Optional	Google OAuth client ID
GITHUB_CLIENT_ID	Optional	GitHub OAuth client ID
PUSHER_KEY	Optional	Real-time notifications
STRIPE_SECRET_KEY	Optional	Billing integration

Table 6. Environment Variables

8. Conclusion

Synchro PM provides a complete, production-ready solution for organizations seeking a modern project management platform. With its multi-tenant architecture, comprehensive feature set, and Vercel-optimized deployment, it offers an enterprise-grade alternative to established platforms while maintaining flexibility for customization and growth.

The system is designed to scale with organizational needs, supporting unlimited organizations through its subscription-ready architecture. Future enhancements can include additional integrations, advanced reporting, and AI-powered features.