CHICAGO SEWER EXPERTS CRM

Technical Architecture & Stack Documentation

Version 1.0 | December 2024

A comprehensive field service management and CRM system

Executive Summary

Chicago Sewer Experts CRM is a full-stack web application designed for field service management in the plumbing and sewer industry. The system provides role-based access for administrators, dispatchers, field technicians, and salespeople.

Key Features:
- Lead management from multiple sources (eLocal, Networx, Angi, Thumbtack, Inquirly, Zapier)
- Real-time GPS tracking with interactive map views
- Quote generation with public shareable links
- Commission tracking based on NET profit calculations
- Automated notifications via email (Resend) and SMS (Twilio/Zapier)
- Pricebook management for standardized service pricing
- Marketing ROI tracking and analytics
- Call/SMS forwarding to office phone

Technology Stack Overview

1. FRONTEND LAYER

Core Framework

React 18 + TypeScript + Vite

Single-page application with type-safe components and hot module replacement for rapid development.

Files: client/src/App.tsx, client/src/main.tsx, vite.config.ts

UI Component System

Radix UI + shadcn/ui Pattern

Accessible, unstyled primitives wrapped in custom styled components. Includes dialogs, dropdowns, forms, tables, and navigation.

Files: client/src/components/ui/*

Styling

Tailwind CSS + Custom Design System

Utility-first CSS with custom color tokens, dark mode support, and elevation utilities for hover/active states.

Files: client/src/index.css, tailwind.config.ts, design_guidelines.md

State Management

TanStack Query (React Query)

Server state management with caching, automatic refetching, and optimistic updates for mutations.

Files: client/src/lib/queryClient.ts

Routing

Wouter

Lightweight client-side routing with role-based route protection.

Files: client/src/App.tsx, client/src/pages/*

Forms

React Hook Form + Zod Validation

Performant form handling with schema-based validation using shared Drizzle-Zod schemas.

Files: client/src/components/ui/form.tsx

Data Visualization

Recharts

Charts for analytics dashboards including bar charts, pie charts, area charts, and line charts.

Files: client/src/pages/admin/analytics.tsx

Maps

Leaflet + React-Leaflet

Interactive maps for real-time technician GPS tracking and job location visualization.

Files: client/src/pages/dispatcher/technician-map.tsx

2. BACKEND LAYER

Server Framework

Express.js + TypeScript

RESTful API server with middleware for logging, authentication, and request validation.

Files: server/index.ts, server/routes.ts

Authentication

Passport.js + Express Sessions

Session-based authentication with local strategy. Supports four user roles: admin, dispatcher, technician, salesperson.

Files: server/auth.ts, server/routes.ts

Database ORM

Drizzle ORM + PostgreSQL

Type-safe database operations with schema-first design. Supports complex queries, relations, and transactions.

Files: server/db.ts, shared/schema.ts

Storage Interface

Abstract Storage Pattern

IStorage interface abstracts data access, enabling easy switching between in-memory and database implementations.

Files: server/storage.ts

Email Service

Resend API

Transactional email delivery for lead notifications, job assignments, and appointment reminders.

Files: server/services/email.ts

SMS Service

Twilio + SignalWire + Zapier Integration

Multi-provider SMS with smart failover. Zapier endpoints bypass A2P verification requirements.

Files: server/services/sms.ts, server/routes.ts (Zapier webhooks)

Automation Service

Lead-to-Job Automation Pipeline

Automated workflows for lead intake, customer contact, job creation, technician assignment, and completion.

Files: server/services/automation.ts

3. DATABASE LAYER

PostgreSQL (Neon-backed)

The database uses PostgreSQL with the following core tables:

users

User accounts with role-based access (admin, dispatcher, technician, salesperson)

Files: shared/schema.ts

leads

Customer leads with source tracking, status, and contact info

Files: shared/schema.ts

technicians

Employee records with hourly rates, skills, GPS location

Files: shared/schema.ts

salespersons

Sales team with commission rates and Twilio routing

Files: shared/schema.ts

jobs

Full job lifecycle with cost tracking and profit calculations

Files: shared/schema.ts

quotes

Customer quotes with line items and public access links

Files: shared/schema.ts

sales_commissions

Commission records with NET profit calculations

Files: shared/schema.ts

pricebook_items

Service catalog with standardized pricing

Files: shared/schema.ts

marketing_campaigns

Marketing campaign tracking and ROI

Files: shared/schema.ts

webhook_logs

Audit trail for all external webhook integrations

Files: shared/schema.ts

4. INTEGRATION LAYER

Lead Source Webhooks

External lead providers send data via HTTP webhooks:

eLocal Webhook

POST /api/webhooks/elocal - Regional lead aggregator

Files: server/routes.ts

Networx Webhook

POST /api/webhooks/networx - Home service leads

Files: server/routes.ts

Angi Webhook

POST /api/webhooks/angi - Angi lead integration

Files: server/routes.ts

Thumbtack Webhook

POST /api/webhooks/thumbtack - Basic auth protected

Files: server/routes.ts

Inquirly Webhook

POST /api/webhooks/inquirly - Form submissions

Files: server/routes.ts

Zapier Webhook

POST /api/webhooks/zapier/lead - Zapier automation

Files: server/routes.ts

Twilio Communication

Voice Forwarding

POST /api/webhooks/twilio/voice - Forwards calls to (630) 251-5628

Files: server/routes.ts

SMS Forwarding

POST /api/webhooks/twilio/sms - Forwards texts + email notification

Files: server/routes.ts

Voicemail

POST /api/webhooks/twilio/voicemail - Records and emails voicemails

Files: server/routes.ts

Zapier SMS Automation

Forward SMS

POST /api/webhooks/zapier/forward-sms - Bypass A2P verification

Files: server/routes.ts

Send SMS

POST /api/webhooks/zapier/send-sms - Template-based messaging

Files: server/routes.ts

Auto-Reply

POST /api/webhooks/zapier/auto-reply - Smart keyword detection

Files: server/routes.ts

5. BUILD & DEPLOYMENT

Development

Vite Dev Server + tsx

Hot module replacement for frontend, TypeScript execution for backend with live reload.

Files: package.json, server/vite.ts

Production Build

Vite + esbuild

Frontend bundled with Vite, backend bundled with esbuild for optimized cold starts.

Files: scripts/build.ts

Deployment

Replit Hosting

Automated deployments with health checks, TLS, and custom domain support.

Files: .replit, replit.nix

Code Organization

Directory Structure

```
project-root/
% % %  client/                    # Frontend React application
%    % % %  src/
%    %    % % %  components/        # Reusable UI components
%    %    %    % % %  ui/           # shadcn/ui component library
%    %    % % %  hooks/            # Custom React hooks
%    %    % % %  lib/              # Utility functions
%    %    % % %  pages/            # Route components
%    %    %    % % %  admin/        # Admin-only pages
%    %    %    % % %  dispatcher/   # Dispatcher pages
%    %    %    % % %  technician/   # Technician pages
%    %    %    % % %  salesperson/  # Salesperson pages
%    %    % % %  App.tsx           # Main app with routing
%    %    % % %  main.tsx          # Entry point
%    % % %  index.html            # HTML template
% % %  server/                    # Backend Express application
%    % % %  services/             # Business logic services
%    %    % % %  automation.ts     # Lead-to-job automation
%    %    % % %  email.ts          # Resend email service
%    %    % % %  sms.ts            # Twilio/SignalWire SMS
%    % % %  auth.ts               # Passport authentication
%    % % %  db.ts                 # Database connection
%    % % %  routes.ts             # API route handlers
%    % % %  storage.ts            # Data access layer
%    % % %  index.ts              # Server entry point
% % %  shared/                    # Shared between client/server
%    % % %  schema.ts             # Drizzle ORM schemas + Zod
% % %  public/                    # Static assets
% % %  migrations/               # Database migrations
```

Code Classification

By Function Type

1. SCHEMA DEFINITIONS

Define data structures shared between frontend and backend.

Drizzle Tables

PostgreSQL table definitions with types, constraints, and relations

Files: shared/schema.ts

Zod Schemas

Runtime validation schemas generated from Drizzle tables

Files: shared/schema.ts

TypeScript Types

Inferred types for insert and select operations

Files: shared/schema.ts

2. API HANDLERS

Process HTTP requests and return responses.

CRUD Endpoints

Standard create, read, update, delete for all entities

Files: server/routes.ts

Webhook Handlers

External integrations from lead sources and Twilio

Files: server/routes.ts

Authentication Routes

Login, logout, session management

Files: server/auth.ts

3. BUSINESS LOGIC

Core application functionality and automation.

Automation Pipeline

Lead intake, auto-contact, job creation, tech assignment

Files: server/services/automation.ts

Commission Calculator

NET profit calculation: Revenue - (Labor + Materials + Travel + Equipment + Other)

Files: server/services/automation.ts

Email Templates

Lead notifications, job assignments, reminders

Files: server/services/email.ts

SMS Templates

Acknowledgment, reminders, completion notices

Files: server/services/sms.ts

4. DATA ACCESS

Abstracts database operations behind interfaces.

IStorage Interface

Defines all CRUD operations available to routes

Files: server/storage.ts

DbStorage Class

PostgreSQL implementation using Drizzle ORM

Files: server/storage.ts

5. UI COMPONENTS

Reusable interface elements following shadcn/ui patterns.

Primitives

Button, Input, Card, Badge, Dialog, etc.
Files: client/src/components/ui/*

Composite

DataTable, Form with validation, Sidebar navigation
Files: client/src/components/ui/*

Domain-specific

LeadCard, JobTimeline, QuoteBuilder, TechnicianMap
Files: client/src/pages/*

6. PAGE COMPONENTS

Full page layouts for each user role and function.

Admin Pages

Dashboard, Leads, Jobs, Analytics, Pricebook, Marketing ROI
Files: client/src/pages/admin/*

Dispatcher Pages

Dispatch Center, Technician Map, Staffing Pool
Files: client/src/pages/dispatcher/*

Technician Pages

My Jobs, Quote Builder, Earnings
Files: client/src/pages/technician/*

Salesperson Pages

Sales Dashboard, Commission Tracker, Quote Tool
Files: client/src/pages/salesperson/*

7. HOOKS & UTILITIES

Reusable logic extracted into hooks and helper functions.

useAuth

Authentication state and user info
Files: client/src/hooks/use-auth.tsx

useToast

Toast notification system
Files: client/src/hooks/use-toast.ts

useMobile

Responsive design detection
Files: client/src/hooks/use-mobile.tsx

apiRequest

Typed fetch wrapper for API calls
Files: client/src/lib/queryClient.ts

API Endpoint Reference

Authentication

POST /api/login - User login
POST /api/logout - User logout
GET /api/user - Current user info

Leads

GET /api/leads - List all leads
POST /api/leads - Create lead
GET /api/leads/:id - Get lead
PATCH /api/leads/:id - Update lead
POST /api/leads/:id/contact - Log contact attempt

Jobs

GET /api/jobs - List all jobs
POST /api/jobs - Create job
GET /api/jobs/:id - Get job
PATCH /api/jobs/:id - Update job
POST /api/jobs/:id/assign - Assign technician
POST /api/jobs/:id/complete - Complete job

Quotes

GET /api/quotes - List quotes
POST /api/quotes - Create quote
GET /api/quotes/:id - Get quote
GET /api/quotes/public/:publicId - Public quote view

Technicians

GET /api/technicians - List technicians
GET /api/technicians/available - Available techs
POST /api/technicians/:id/location - Update GPS

Webhooks

POST /api/webhooks/elocal - eLocal leads
POST /api/webhooks/networx - Networx leads
POST /api/webhooks/angi - Angi leads
POST /api/webhooks/thumbtack - Thumbtack leads
POST /api/webhooks/zapier/lead - Zapier leads
POST /api/webhooks/zapier/forward-sms - SMS forwarding
POST /api/webhooks/twilio/voice - Call forwarding

Document Information

Generated: December 26, 2025 at 05:52 PM

Application: Chicago Sewer Experts CRM
Version: 1.0

For technical support or questions, refer to:
- OPERATIONS_MENU.md - Complete operations guide
- replit.md - Development documentation
- design_guidelines.md - UI/UX standards