

Architected a Scalable System for Community Relationship Management

Requirements discovery, solution architecture, data strategy, workflow planning, and phased implementation developed for a proposed public-sector relationship management platform.

SYSTEMS STRATEGY · BUSINESS ANALYSIS · SOLUTION ARCHITECTURE · PROJECT PLANNING

KPI SNAPSHOT

A complex spreadsheet-based tracking process was translated into a structured, buildable cloud-platform strategy with defined data, workflow, reporting, and implementation requirements.

300+

Organizations Planned for Migration

50

Data Attributes per Organization

14

Week Implementation Plan

6

Defined Development Phases

ROLE

Solution Strategy & Project Architecture

PROJECT OVERVIEW

Evaluated a public-health team's spreadsheet-based organization tracking process and translated its operational needs into a complete system strategy. Defined the proposed database structure, user access, communication logging, reporting capabilities, development roles, implementation phases, technology requirements, data migration approach, testing, training, and handoff plan needed to move the concept from an internal spreadsheet workflow to a scalable cloud-based platform.

DATA NOTE

Metrics reflect the proposed system scope and implementation plan. The project was not executed after county-level budget cuts eliminated the funding allocated for the initiative.

CONTEXT

The challenge and the strategic response

The team relied on Excel to track hundreds of community organizations and detailed relationship information. The operational need extended beyond storing records: staff required shared visibility, communication history, reporting, secure access, and a structure that could grow with ongoing outreach.

PROBLEM

Spreadsheet-based relationship tracking

Hundreds of organization records and dozens of attributes were managed in a format not designed for scalable, collaborative relationship management.

Limited shared communication history

Outreach notes and relationship activity needed to remain visible across staff and departments without losing chronology or accountability.

Reporting required manual interpretation

The existing data needed to become filterable, visual, and exportable so staff could identify patterns and produce useful internal reporting.

Complex idea without a build roadmap

The organization knew what the system needed to accomplish, but the concept required a defined architecture, development sequence, roles, testing plan, and handoff process.

STRATEGIC RESPONSE

● Defined a scalable database model

Structured the proposed migration around 300+ organizations, up to 50 attributes per record, expandable organization entries, and future staff-user growth.

● Designed shared communication logging

Specified time-stamped notes, assignable points of contact, shared access, secure authentication, and a non-editable audit trail for continuity and review.

● Planned dashboards and reporting tools

Defined multi-criteria filters, three chart formats, and PDF and CSV exports to convert stored information into usable operational insight.

● Built a phased implementation strategy

Translated the concept into six development phases covering architecture, infrastructure, interface design, data handling, QA, training, documentation, and delivery.

WHAT WAS BUILT

A complete project strategy that translated operational requirements into a buildable technical plan

→ Requirements & Data Architecture

Defined migration requirements, organization records, attribute structure, user expansion, and the core database logic required to support the system.

→ Access & Accountability Model

Specified secure authentication, editable user profiles, assignable contacts, shared access, and audit-trail requirements.

→ Communication Workflow Design

Planned time-stamped organization notes and shared communication histories to support continuity across outreach roles and departments.

→ Reporting & Visualization Strategy

Defined filters, bar, pie, and line charts, plus PDF and CSV export requirements for internal reporting and presentations.

→ Phased Development Roadmap

Structured a 14-week plan across planning, backend infrastructure, frontend design, data handling, testing, optimization, onboarding, and delivery.

→ Delivery & Governance Framework

Defined project roles, QA responsibilities, data mapping, documentation, staff training, system handoff, and required client launch inputs.

What the proposed system was designed to support

The completed strategy converted a broad operational need into a defined system scope with measurable data requirements, reporting outputs, team responsibilities, and implementation readiness.

3

Defined Development Roles

3

Visual Reporting Formats

2

Report Export Formats

6

Required Launch Inputs

PROJECT DEFINITION OUTCOMES

01 Converted an operational need into system requirements

Defined what information the platform needed to store, how users would interact with it, how communications would be recorded, and how data would be surfaced for decision-making.

02 Created a development-ready implementation sequence

Organized the work into six interdependent phases so architecture, infrastructure, interface design, migration, testing, training, and delivery could be planned in the correct order.

03 Connected operational workflows to technical architecture

Mapped real outreach and collaboration needs to database structure, authentication, audit history, reporting, filtering, exports, and responsive interface requirements.

04 Defined ownership, roles, and handoff expectations

Established development responsibilities, required client inputs, quality assurance, training, documentation, and final handoff requirements before implementation began.

WHAT THIS PROJECT

DEMONSTRATES

Systems Strategy · Business Analysis · Solution Architecture · Requirements Discovery · Data Architecture · Workflow Design · Project Planning · Technical Scoping