VC Outreach CRM - Technical Documentation

Project: AI-Powered Investor Outreach Platform
Client: Astant Global Management
Version: 1.0.0 (MVP)
Status: Complete, Ready for Deployment

Table of Contents

1. Executive Summary

Business Context

Astant Global Management operates across Madrid, Bangalore, and Luxembourg, providing quantitative finance, asset management, and trade finance services. Effective investor relations require personalized, high-quality outreach at scale—a challenge that traditional CRM systems fail to address.

Solution Delivered

This CRM system addresses the following operational challenges:

Challenge	Solution
Time-intensive personalized outreach	AI-powered email generation using Claude 3.5 Sonnet
Inconsistent email quality	Confidence scoring with red/yellow/green classification
Risk of sending inappropriate content	Human-in-the-loop approval workflow
Lack of relationship context	Persistent relationship memory database
No quality control mechanism	Tinder-style review queue with mandatory approval

Deliverables Summary

Component	Description
Backend Functions (3)	generate-draft, rebuttal, send-drip

Component	Description
Frontend Pages (4)	Dashboard, Queue, Campaigns, Contacts
Database Schema	PostgreSQL with 10 tables, 8 ENUM types
Demo Mode	Functional UI without backend dependencies

---

2. Project Scope

Included in MVP

Email Generation

- AI-powered personalization using contact and campaign data
- Structured output format (greeting, context, value proposition, CTA, signature)
- Confidence scoring based on data completeness
- Fallback strategy for incomplete contact records

Quality Control

- Mandatory human review before sending
- One-click refinement options (5 rebuttal types)
- Database-level constraints preventing red-confidence email delivery
- Full audit trail with original and modified versions preserved

Contact Management

- Contact database with firm, role, geography, and investment focus
- Campaign assignment and stage tracking
- Relationship memory for contextual AI generation

Campaign Management

- Campaign creation with tone, CTA, and fallback configuration
- Media asset linking (decks, one-pagers, tearsheets)
- Per-campaign metrics tracking

Excluded from MVP

- User authentication and role-based access
- Scheduled/automated sending
- Open and click tracking
- Email reply detection
- Bulk CSV import
- Analytics dashboard

---

3. Architecture Overview

Technology Stack

Layer	Technology	Rationale
Database	PostgreSQL (Supabase)	ACID compliance, ENUM support, RLS-ready
Backend	Supabase Edge Functions (Deno)	Low latency, same infrastructure as DB
AI Provider	Anthropic Claude 3.5 Sonnet	Superior instruction-following, consistent JSON output
Frontend	Next.js 14 (App Router)	Production-stable, TypeScript-native
Styling	Tailwind CSS	Rapid development, responsive by default
Email Delivery	Resend (pluggable)	Modern API, reliable delivery

System Diagram

```
                         ASTANT VC OUTREACH CRM

    ┌─────────────────────────────────────────────────────────────┐
    │                          FRONTEND                             │
    │                        (Next.js 14)                           │
    │  ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐        │
    │  │Dashboard │ │  Queue   │ │Campaigns │ │ Contacts │        │
    │  └──────────┘ └──────────┘ └──────────┘ └──────────┘        │
    └─────────────────────────────────────────────────────────────┘
                                 │
                                 ▼
    ┌─────────────────────────────────────────────────────────────┐
    │                       EDGE FUNCTIONS                          │
    │                     (Supabase / Deno)                        │
    │  ┌───────────────┐ ┌───────────────┐ ┌───────────────┐      │
    │  │generate-draft │ │   rebuttal    │ │   send-drip   │      │
    │  └───────┬───────┘ └───────┬───────┘ └───────┬───────┘      │
    └──────────┼─────────────────┼─────────────────┼──────────────┘
               │                 │                 │
               ▼                 ▼                 │
    ┌────────────────────────────────────┐        │
    │           ANTHROPIC API            │        │
    │        (Claude 3.5 Sonnet)         │        │
    └────────────────────────────────────┘        │
                                                   ▼
    ┌─────────────────────────────────────────────────────────────┐
    │                          DATABASE                             │
    │                   (PostgreSQL / Supabase)                    │
    │  ┌────────┐ ┌────────┐ ┌────────┐ ┌────────┐ ┌────────┐    │
    │  │contacts│ │campaigns│ │ emails │ │ memory │ │ events │    │
    │  └────────┘ └────────┘ └────────┘ └────────┘ └────────┘    │
    └─────────────────────────────────────────────────────────────┘
```

4. Design Rationale

Frozen Architecture Principle

The database schema (v1.2) was finalized before implementation began. This decision provides:

- Prevention of scope creep during development
- Consistent data contracts across all system components
- Simplified auditing and maintenance
- Early detection of data integrity issues via constraints

Deterministic AI Principle

AI is constrained to specific, bounded operations:

Permitted:

- Generate context paragraph (context_p1)
- Generate value proposition paragraph (value_p2)

Prohibited:

- Modify greeting (locked to contact name)
- Modify CTA (locked from campaign configuration)
- Modify signature (locked from request parameter)
- Fabricate information not present in database
- Speculate about relationship history

Human-in-the-Loop Principle

Every email follows a mandatory workflow:

```
Generate → Review → [Optional: Refine → Review] → Approve → Send
```

No email can be sent without explicit human approval. This is enforced at the database level via CHECK constraints.

Safety Rails Implementation

Database constraints prevent:

Constraint	Implementation
Sending unapproved emails	`approved_at IS NOT NULL` check before send
Sending red-confidence emails	CHECK constraint on stage transitions
Modifying sent emails	`sent_at IS NULL` check on updates
Invalid stage progression	ENUM type with allowed values only

5. Data Flow Specifications

Generate Draft Flow

Endpoint: `POST /functions/v1/generate-draft`

Request:

```
{
  "contact_id": "uuid",
  "campaign_id": "uuid",
  "signature": "Best regards,\nJohn Smith\nAstant Global Management"
}
```

Processing Steps:

1. Validate request parameters
2. Fetch campaign (tone, CTA, context, fallback strategy)
3. Fetch media assets linked to campaign
4. Fetch relationship memory for contact (if exists)
5. Fetch contact details (name, firm, role, investment focus)
6. Build composite context object (sanitized for AI consumption)
7. Construct prompt with strict output format requirements
8. Call Claude 3.5 Sonnet API
9. Parse JSON response and validate structure
10. Calculate confidence score based on data completeness
11. Persist to emails table (original_body = current_body)
12. Return email preview with confidence classification

Response:

```
{
  "email_id": "uuid",
  "subject": "string",
  "preview": {
    "greeting": "string",
    "context_p1": "string",
    "value_p2": "string",
    "cta": "string",
    "signature": "string"
  },
  "confidence": "green | yellow | red"
}
```

Rebuttal Flow

Endpoint: `POST /functions/v1/rebuttal`

Rebuttal Types:

Type	Instruction
SOFTER_TONE	Reduce assertiveness, add hedging language
MORE_TECHNICAL	Add quantitative details, technical terminology
SHORTER	Reduce word count by 30-40% while preserving key points
CLARIFY_VALUE_PROP	Strengthen unique value articulation
LESS_PITCHY	Remove sales language, adopt peer-to-peer tone

Field Locking:

- Greeting: Preserved from original
- CTA: Preserved from original
- Signature: Preserved from original
- context_p1: Modified per rebuttal instruction
- value_p2: Modified per rebuttal instruction

Send Flow

Endpoint: `POST /functions/v1/send-drip`

Validation Checks:

1. Email exists
2. Email not already sent (sent_at IS NULL)
3. Email is approved (approved_at IS NOT NULL)
4. Confidence is not red
5. Contact has valid email address

Processing:

1. Render JSON body to plain text and HTML formats
2. Send via configured email provider (Resend)
3. Update emails.sent_at timestamp
4. Update contact_campaigns.stage to 'sent'
5. Create engagement_events record

---

6. Component Reference

Backend: Edge Functions

generate-draft/index.ts

- Lines: ~350
- Purpose: AI-powered email generation
- Dependencies: Supabase client, Anthropic SDK

rebuttal/index.ts

- Lines: ~250
- Purpose: One-click email refinement
- Dependencies: Supabase client, Anthropic SDK

send-drip/index.ts

- Lines: ~280
- Purpose: Email delivery with validation
- Dependencies: Supabase client, Resend SDK

Frontend: Pages

page.tsx (Dashboard)

- Route: /
- Purpose: Overview metrics and navigation

queue/page.tsx

- Route: /queue
- Purpose: Tinder-style email review interface
- Features: Swipe animations, rebuttal menu, approval workflow

campaigns/page.tsx

- Route: /campaigns
- Purpose: Campaign creation and management

contacts/page.tsx

- Route: /contacts
- Purpose: Contact database management

Shared: Library

lib/types.ts

- TypeScript interfaces matching database schema
- ENUM types as union types

lib/utils.ts

- Utility functions for formatting and classification

lib/supabase/client.ts

- Browser-side Supabase client initialization

lib/supabase/server.ts

- Server-side Supabase client initialization

7. Technical FAQ

Operational Questions

Q: Is Supabase required to run the application?

A: The UI functions in demo mode without Supabase, displaying sample data for design review. Full functionality requires Supabase configuration.

Q: Is an Anthropic API key required?

A: Only for generate-draft and rebuttal functions. The UI and send-drip function operate independently.

Q: Can alternative email providers be used?

A: Yes. The send-drip function implements a pluggable sendEmail() function. Resend is configured; SendGrid, AWS SES, Postmark, and SMTP can be added.

Data Questions

Q: What happens to historical emails?

A: Full audit trail preserved:

- original_body: Immutable after generation
- current_body: Final version sent
- sent_at: Delivery timestamp
- engagement_events: Open, click, reply tracking

Q: How is GDPR compliance addressed?

A: Schema supports GDPR requirements:

- CASCADE delete removes all related records
- No unnecessary data retention
- Consent field can be added to contacts table

Architecture Questions

Q: Why Edge Functions instead of Next.js API Routes?

A: Edge Functions provide lower latency (same infrastructure as database), automatic scaling, native cron/queue support, and pay-per-invocation cost model.

Q: Why Claude 3.5 Sonnet instead of GPT-4?

A: Claude demonstrates superior instruction-following, more consistent JSON output formatting, nuanced tone control, and lower cost per token.

Q: Why PostgreSQL ENUMs instead of flexible strings?

A: ENUMs provide database-level validation, faster query performance (integer comparison), self-documenting schema, and prevention of invalid states.

8. Future Roadmap

Feature	Priority	Effort	Notes
User authentication	High	Medium	Supabase Auth integration
Scheduled sending	Medium	Low	Cron job implementation
Open/click tracking	Medium	Medium	Webhook endpoint, pixel tracking
Reply detection	Medium	High	Inbox integration required
Campaign detail page	Medium	Low	Batch draft generation
Contact detail page	Medium	Low	Relationship history view
Bulk CSV import	Low	Low	Parse and insert logic
Email templates	Low	Medium	Template library
Analytics dashboard	Low	Medium	Conversion funnel visualization

9. Extension Guide

Adding a Rebuttal Type

1. Update schema ENUM:

```
ALTER TYPE rebuttal_enum ADD VALUE 'NEW_TYPE';
```

2. Add instruction in rebuttal/index.ts:

```
const REBUTTAL_INSTRUCTIONS = {
  NEW_TYPE: `Instruction text here...`,
}
```

3. Add option in UI component.

Adding a Contact Field

1. Add column:

```
ALTER TABLE contacts ADD COLUMN linkedin_url TEXT;
```

2. Update TypeScript interface in lib/types.ts

3. Add form field in contacts/page.tsx

4. Optionally include in composite context for AI

Adding an Email Provider

Modify sendEmail() in send-drip/index.ts:

```
async function sendEmail(payload: EmailPayload) {
  const PROVIDER_API_KEY = Deno.env.get("PROVIDER_API_KEY");

  if (PROVIDER_API_KEY) {
    // Provider-specific implementation
  }

  // Fallback chain
}
```

Summary

Delivered: A complete AI-powered CRM for investor outreach with card-based approval queue.

Design Principles: Frozen architecture, deterministic AI, human-in-the-loop, database-level safety rails.

Operational Capabilities:

- AI email generation with confidence scoring
- One-click refinement (5 rebuttal types)
- Red-confidence blocking at database level
- Card-based review queue
- Campaign and contact management
- Pluggable email delivery

Next Steps: Configure Supabase credentials, implement authentication, deploy to production environment.

Technical documentation for Astant Global Management VC Outreach CRM.
Built with Supabase, Next.js 14, Claude 3.5 Sonnet, and Tailwind CSS.