REACH OUT BOT

Software Development Plan (SDP)

Last updated on Feb 2, 2026

1. Introduction

1.1 Purpose

This Software Development Plan (SDP) defines the technical, organizational, and procedural approach for building the Reach Out Bot (NGO Outreach Automation Platform). It translates the Product Requirements Document (PRD) into an actionable engineering plan aligned with software engineering best practices found in peer-reviewed literature and industry standards (*e.g.* IEEE 1058, ISO/IEC/IEEE 12207, Sommerville, Pressman).

1.2 Scope

This SDP covers the full lifecycle of **v1** of the platform, from architecture design through development, testing, deployment, and maintenance, explicitly aligned with the PRD's *Definition of Done (v1)*.

Out-of-scope items mirror those defined in the PRD (*e.g.* advanced CRM automation, multi-language outreach).

1.3 Intended Audience

- Engineering team
- Product and impact stakeholders
- QA and operations
- Security and compliance reviewers

---

2. Development Methodology

2.1 Process Model

A **hybrid Agile–Incremental model** will be used:

- Iterative delivery of functional increments
- Continuous stakeholder feedback
- Strict scope control for v1

This model balances uncertainty in AI-driven components with the determinism required for CRM, email, and compliance workflows.

2.2 Sprint Structure

- Sprint length: 2 weeks
- Sprint planning, review, and retrospective mandatory
- Each sprint produces a deployable increment (even if feature-flagged)

2.3 Definition of Ready (DoR)

A feature enters development only if:

- Functional and non-functional requirements are explicit
- API dependencies are defined
- Acceptance criteria are testable

2.4 Definition of Done (DoD)

A feature is complete only if:

- Acceptance tests pass
- Security and privacy checks completed
- Deployed to staging
- Documented (README + inline comments)

2.5 Hard Constraints & Invariants

The following constraints apply across all components and override any conflicting functional or non-functional requirement.

---

3. System Architecture

3.1 Architectural Style

The system follows a **serverless, event-driven architecture**:

- Google Cloud Functions (2nd gen)
- Stateless compute
- Externalized state

This architecture is chosen to maximize scalability, fault isolation, and cost control.

3.2 High-Level Components

1. **Web Frontend**
   - Brief input, NGO review, email review
   - Authentication via Google OAuth
2. **Backend Services (Cloud Functions)**
   - Brief parsing (Gemini AI)
   - NGO discovery and ranking
   - Partner status classification
   - Email generation
   - Follow-up orchestration
3. **External Integrations**
   - HubSpot API
   - Gmail API
   - Google Drive API
4. **Persistence Layer**
   - Supabase (PostgreSQL)

    - Stores workflow state, timestamps, status flags, and structured AI outputs
5. **Scheduling & Events**
    - Cloud Scheduler
    - Pub/Sub (optional for decoupling)

---

4. Data Design

4.1 Data Persistence Model

The system uses Supabase (PostgreSQL) as the primary persistence layer. Supabase provides transactional guarantees, explicit relational modeling, and native support for semi-structured data (JSONB), making it well suited for managing outreach workflows, state transitions, and AI-generated outputs.

This persistence model must support reliable execution of scheduled follow-ups, idempotent integrations with external systems, and future extensibility for reporting and automation.

4.2 Core Data Entities

- Brief
    - Raw brief text
    - Parsed and normalized JSON output
- **NGO Candidate**
    - NGO name
    - Geography
    - Focus areas
    - Fit rationale
    - Partner status classification
    - Reputational risk score (advisory)
    - Reputational risk summary (auto-generated text referencing public sources)

- **Outreach Record**
    - NGO identifier
    - HubSpot IDs (company, contact, deal), when applicable
    - Gmail email thread ID
    - Outreach status *(Drafted / Sent / Replied / Follow-up Pending / Closed)*
    - Follow-up counters and relevant timestamps

4.3 Data Governance Principles

- Minimal data retention; only workflow-critical data must be persisted
- No modification, enrichment, or overwriting of existing HubSpot records
- Source data (user input, HubSpot records) and derived data (AI-generated outputs) must remain strictly separated
- Idempotent writes and transactional updates enforced at the database level
- Supabase is authoritative for workflow state; HubSpot is authoritative for CRM data. Conflicts must always resolve in favor of the authoritative system.

---

5. Component-Level Design

All components must enforce the system-wide "Hard Constraints & Invariants" defined in Section 2.5.

5.1 Intelligent Brief Parsing

- Gemini AI prompt templates version-controlled
- Deterministic JSON schema validation
- Fail-safe defaults must be applied for missing entities

5.2 NGO Identification & Research

- The system must produce a ranked scoring function with explainable signals
- Deduplication logic mandatory
- Generates a reputational risk score and controversy summary based on public information

- Elevated reputational risk must be flagged for user awareness only
- User retains full discretion to proceed regardless of risk indicators
- Reputational risk signals must never block NGO selection or outreach actions
- Explicit confidence thresholds must be enforced

Reputational risk outputs are advisory signals only and must not block downstream selection or outreach actions.

(Illustrative scoring framework defined in PRD Appendix A)

5.3 Partner Status Classification

- Single source of truth: HubSpot Environmental Pipeline
- Mutually exclusive state machine

5.4 Email Generation

- Email templates must be retrieved from Google Drive at runtime
- All placeholders must be validated before sending
- Email regeneration must not produce side effects
- Email drafts must not be sent, scheduled, or auto-dispatched without an explicit user approval event

5.5 Follow-Up Engine

- Time-based finite state machine
- Human confirmation must act as a hard gate for all follow-up actions
- Business-day calendar logic must be enforced
- State transitions must be idempotent and must never result in duplicate follow-up emails

---

6. Security & Privacy

6.1 Authentication & Authorization

- OAuth 2.0 for Gmail, HubSpot, Google Drive
- Principle of least privilege

6.2 Secrets Management

- GCP Secret Manager
- No secrets in code or logs

6.3 Data Privacy

- No scraping of private data
- Only publicly available sources must be used for NGO research
- The system must comply with Google and HubSpot API policies

---

7. Quality Assurance Strategy

7.1 Testing Levels

- **Unit Tests**
    - Parsing logic
    - Status classification
- **Integration Tests**
    - HubSpot record creation
    - Gmail sending and logging
- **End-to-End Tests**
    - Brief → NGO list → Email draft

7.2 AI Output Validation

- Schema validation
- Confidence scoring
- Human review checkpoints must be enforced before downstream actions

---

8. Deployment Plan

8.1 Environments

- Development
- Staging
- Production

8.2 CI/CD

- Automated build and test pipeline
- Manual approval for production deploy

8.3 Rollback Strategy

- Versioned Cloud Functions
- Feature flags for risky components

---

9. Monitoring & Maintenance

9.1 Observability

- Structured logging
- Error rate and latency tracking

9.2 Operational Alerts

- API quota exhaustion
- Failed follow-up workflows

9.3 Maintenance Policy

- Dependencies must be reviewed and updated monthly
- Prompt and template reviews

10. Risks & Mitigation (Engineering Perspective)

Risk	Mitigation
AI hallucination	Schema validation + human review
API instability	Retries + graceful degradation
Duplicate CRM records	Idempotent logic
Silent email failures	Hard blocks + user warnings
Over-automation by AI agent	Hard Constraints & Invariants enforced at runtime and test level

11. Acceptance Criteria (v1)

The system is accepted when:

- Users must be able to complete the full outreach flow without manual research
- NGO recommendations are transparent and defensible
- All sent emails are logged correctly in HubSpot
- Follow-ups never occur without explicit confirmation

12. References (Methodological)

- IEEE Std 1058 – Software Project Management Plans

- ISO/IEC/IEEE 12207 – Software Life Cycle Processes
- Sommerville, *Software Engineering*
- Pressman & Maxim, *Software Engineering*: *A Practitioner’s Approach*