ReachOut Platform - System Documentation & Process Manual

Version: 1.0 (Beta)
Date: November 2025
Platform Type: Humanitarian / Emergency Last-Mile Communication

1. Executive Summary

ReachOut is a crisis response platform designed to bridge the "digital gap" during emergencies. When standard communication channels (calls/SMS) fail due to dead batteries, lost phones, or network congestion, ReachOut activates a network of verified volunteers to physically locate individuals and relay messages.

The system operates on a Human Relay concept:
Requester (Digital) -> Admin (Verification) -> Volunteer (Physical Action) -> Admin (Validation) -> Requester (Result)

2. System Architecture

- **Frontend:** React (TypeScript) with Tailwind CSS for responsive, high-contrast "Emergency UI".
- **Backend:** Firebase Firestore (Real-time NoSQL Database).
- **Authentication:** Firebase Auth (Anonymous & Custom Tokens).
- **Localization:** Built-in translation engine supporting **English, Sinhala, and Tamil**.

3. User Roles & Capabilities

The platform serves three distinct user roles, each with specific permissions and workflows.

A. The Requester (Public User)

- **Goal:** Find a missing loved one or deliver an urgent message.
- **Key Features:**
  - **Multi-lingual Form:** Submit details (Target Name, Address, Phone, Message).
  - **Live Dashboard:** Track the status of the request in real-time.
  - **Case Closure:** Confirm receipt of information to close the loop.

B. The Administrator (Moderator)

- **Goal:** Ensure safety, prevent spam, and verify volunteer findings.
- **Key Features:**
  - **Secure Login:** Requires password authentication.
  - **Request Verification:** "Calls" the requester to ensure the emergency is real before publishing.
  - **Report Validation:** Reviews volunteer reports before releasing them to the family.
  - **Volunteer Oversight:** Has access to Volunteer phone numbers for accountability.

C. The Volunteer (Field Agent)

- **Goal:** Locate the target and report back.
- **Key Features:**
  - **Guest/Login Modes:** Can view tasks as a guest or login for detailed performance stats.
  - **Task Acceptance:** Must provide a verified **Phone Number** to accept a mission.
  - **Navigation:** One-click integration with Google Maps.
  - **Insights Dashboard:** Visualizes "People Reached" vs. "Platform Needs" (Donut & Bar charts).

4. The Core Process Workflow (The "Happy Path")

This section details the exact lifecycle of a request from start to finish.

Phase 1: Initiation

1. **Submission:** Requester fills out the "New Emergency Request" form.
2. **State:** Request enters the system with status **Pending Verification** (Yellow).
3. **Visibility:** Visible ONLY to the Requester and Admin. Hidden from Volunteers to prevent spam.

Phase 2: Verification

1. **Admin Action:** Admin sees the request in the "New Requests Pending" list.
2. **Check:** Admin calls the Requester's phone number to confirm identity.
3. **Action:** Admin clicks **VERIFY**.
4. **State:** Status changes to **Verified - Needs Volunteer** (Blue).
5. **Visibility:** Now visible to all Volunteers in the "Available Near You" list.

Phase 3: Field Action

1. **Volunteer Action:** A Volunteer sees the request and clicks **ACCEPT MISSION**.
2. **Safety Gate:** System prompts Volunteer to enter their **Phone Number**.
3. **Action:** Volunteer confirms.
4. **State:** Status changes to **Volunteer En Route** (Orange).

5. **Navigation:** Volunteer travels to the targetAddress.

Phase 4: Reporting & Validation (Safety Layer)

1. **Volunteer Action:** Volunteer locates the target (or the house).
2. **Reporting:** Volunteer types the outcome (e.g., "Found her, she is safe") and clicks **SUBMIT FOR REVIEW**.
3. **State:** Status changes to **Report Under Review** (Pink).
4. **Privacy Lock:** The Requester sees "Report Sent" but **cannot read the details yet**.
5. **Admin Action:** Admin sees "Reports Pending Review".
6. **Verification:** Admin calls the Volunteer (using the number provided in Phase 3) to ensure the report is truthful.
7. **Action:** Admin clicks **APPROVE & NOTIFY**.

Phase 5: Closure

1. **State:** Status changes to **Report Approved** (Green/Purple).
2. **Requester Action:** Requester can now read the Volunteer's message.
3. **Action:** Requester clicks **CONFIRM DETAILS RECEIVED**.
4. **State:** Status changes to **Case Closed** (Green Checkmark).

5. Database Schema (Firestore)

The data is stored in the emergency_requests collection.

Field Name	**Type**	**Description**
id	String	Unique Document ID.
targetName	String	Name of the person to find.
targetAddress	String	Physical location for the volunteer.
targetPhone	String	(Optional) Phone of the missing person.
requesterName	String	Name of the user asking for help.
requesterPhone	String	Phone of the user asking for help.
message	String	The message to be delivered.

status	String	Lifecycle state (pending, verified, accepted, report_review, found, completed).
volunteerId	String	UID of the volunteer who accepted the task.
volunteerPhone	String	**Crucial:** Phone number of the volunteer for accountability.
volunteerReport	String	The findings submitted by the volunteer.
createdAt	Timestamp	Server time for sorting.

6. Security & Safety Protocols

1. **Double-Blind Privacy:**
   - Volunteers do not see Requester details until they accept the mission.
   - Requesters do not see Volunteer details (only status updates).
2. **Misinformation Prevention:**
   - Volunteers cannot post findings directly to the public. All "Found" reports go into a holding pattern (report_review) until an Admin validates them. This prevents trolls from giving families false hope.
3. **Volunteer Accountability:**
   - Volunteers cannot accept a task anonymously. They must provide a phone number at the moment of acceptance, creating a digital trail.

7. Future Roadmap (Technical Suggestions)

- **GPS Integration:** Automatically verify Volunteer location using the device's GPS coordinates before allowing them to mark "Arrived".
- **Push Notifications:** Send SMS alerts to Volunteers when a request appears in their zip code.
- **Offline First:** Use PWA (Progressive Web App) caching to allow Volunteers to view address details even if they lose internet connection while traveling.