

Makhi Application Specification Document

Date: March 07, 2025

Prepared by: Technical Specialist (on behalf of Salatiso, Client)

Application Name: Makhi (short for Makhelwane – "Neighbor" in Xhosa)

1. Overview

- 1.1 Purpose

Makhi is a community-driven application designed to enhance communication, cohesion, and crime-fighting efforts among neighbors in South African townships. It aims to replace ineffective tools like WhatsApp with a structured, secure, and localized platform that fosters neighborly cohesion, facilitates real-time emergency alerts, and provides critical updates on utilities and incidents.

- 1.2 Context and Justification

South African townships face rampant crime, unreliable utilities (e.g., load shedding, water outages), and fragmented communication. Current solutions like WhatsApp are chaotic—important messages get lost in noise, emergencies go unnoticed, and there's no verification of members, risking infiltration by outsiders. Makhi solves this by:

- Local Control: Runs peer-to-peer (P2P) without central administrators or external servers, keeping data within the community.
- Verification: Ensures only legitimate neighbors join via a 60% approval process, building trust.
- Purpose-Driven: Offers dedicated tabs for utilities, social activities, and incidents, plus a critical emergency alert feature.
- Crime Focus: Empowers residents to report incidents with multimedia and alert others instantly, combating slow police response times.

- 1.3 Value Proposition

- Safety: Real-time emergency alerts and incident reporting enable rapid community response to crime.
- Cohesion: Structured communication and social features strengthen neighborly bonds, countering isolation.
- Resilience: Offline functionality and utility updates help residents navigate infrastructure challenges.
- Privacy: Minimal personal data reduces risk if devices are compromised, a frequent issue in high-crime areas.
- Scalability: Designed for 50 households (up to 500 people), it's replicable across streets and townships.

- 1.4 Target Audience

- Residents of South African townships, with a focus on those in crime-affected areas.
- Users who have basic smartphone skills and are familiar with communication apps.
- Communities seeking to improve local communication, safety, and social cohesion.
- 1.5 Stakeholders
 - Primary Stakeholders:
 - Residents of target townships
 - Community leaders and organizations
 - Secondary Stakeholders:
 - Local law enforcement agencies
 - Municipal service providers
 - Investors and funders
 - Developers and technical team
 - Neighbors

2. Functional Requirements

- 2.1 Scope

Makhi will initially launch as a web application, later convertible to an Android app (e.g., via Apache Cordova). It supports up to 50 households per street, with each household accommodating up to 10 members. The app operates locally via P2P syncing, with no central server or administrators.
- 2.2 User Registration and Verification
 - 2.2.1 Starter Setup:
 - The first user (The Starter) downloads Makhi, enters their street name (e.g., “Mofolo Street, Soweto”), and specifies the total number of households in the street (max 50).
 - Required data: Name, Surname, House Number, Street Name.
 - The Starter is not auto-verified but initiates the network without voting privileges until others join.
 - 2.2.2 Household Representation:
 - The first member of a household (Household Starter) represents the household for street-level voting.
 - One household = one vote, regardless of how many members (1-10) join from that household.
 - 2.2.3 Street-Level Verification:
 - Second household member downloads Makhi, enters their details, and requests to join the street.

- Existing verified members vote (Approve/Reject). Requires 60% approval from all voting households (e.g., 6/10 if 10 households are registered).
 - Minimum 3 “yes” votes required when fewer than 5 households are active, scaling to 60% as more join.
 - 2.2.4 Household-Level Verification:
 - Subsequent members of the same household (e.g., No. 12) request to join via the Household Starter.
 - Approval requires 60% of existing household members (e.g., 3/5 if 5 are in), with a minimum of 1 “yes” (Household Starter) initially.
 - Max 10 members per household.
 - 2.2.5 Removal: Members/households can be removed by a 60% vote at their respective level (street or household).
- 2.3 Core Features
 - 2.3.1 Home Screen:
 - Displays street name, total households (e.g., “Mofolo Street – 12/50 Households”), and verified member count.
 - Navigation tabs: Utilities, Social, Incidents.
 - Prominent “EMERGENCY ALERT” button (red, always visible).
 - 2.3.2 Utilities Tab:
 - Purpose: Share updates on load shedding, water outages, and infrastructure issues.
 - Form: Dropdown (Power/Water/Other), text field for details, “Submit.”
 - Display: Timeline (e.g., “07/03/25 14:00 – Salatiso (No. 12): Water off, pipe burst.”).
 - Access: Any member can post or view.
 - 2.3.3 Social Tab:
 - Purpose: Foster community through announcements and activities.
 - Sections:
 - Announcements: Pinned posts (e.g., “Braai at No. 20, Sat 3pm”), editable only with 60% household vote.
 - Chat: Group chat for casual interaction.
 - Access: All members can post/chat; pinned items need approval.
 - 2.3.4 Incidents Tab:
 - Purpose: Report and track crime or suspicious activity.
 - Form:
 - Multimedia: Photo (camera access), Video (10-sec max), Audio (voice note).

- Text field: Incident details (e.g., “Two guys at No. 8 gate, 9pm”).
 - “Submit” to Incidents chat.
 - Display: Timeline (e.g., “07/03/25 21:03 – Themba (No. 12): [Photo] Suspicious car.”).
 - Access: All members can report/view.
 - 2.3.5 Emergency Alert:
 - Button: Red, labeled “EMERGENCY ALERT,” on every screen.
 - Action:
 - Triggers loud sound/vibration on all devices.
 - Sends alert with sender’s name/house (e.g., “Salatiso (No. 12): Emergency!”).
 - Limit: One alert per household every 10 minutes to prevent spam.
 - Access: All members can trigger; all receive.
 - 2.3.6 Group Chats
 - Tabs: Separate chats for Utilities, Social, Incidents.
 - Format: Messages tagged with nickname (e.g., “Salatiso12: Gate’s open again.”).
 - Access: All members can post/view within respective tabs.
- 2.4 Privacy and Data
 - Required Data: Name, Surname, House Number, Street Name (stored locally on devices).
 - Nicknames: Used in chats (e.g., “Themba12”) to minimize exposure.
 - No Extras: No emails, phone numbers, or IDs collected.
 - Storage: Data cached locally, synced P2P when devices are in proximity (e.g., via Bluetooth/Wi-Fi).
- 2.5 Offline Functionality
 - Requirement: Fully functional during loadshedding/no internet.
 - Implementation: Cache all data (members, chats, alerts) locally; sync when connectivity resumes or via P2P.
 - Priority: Alerts and incident reports must save and queue offline, broadcasting once synced.
- 2.6 Mapping and Location
 - Integration with Google Maps for display of household locations.
 - Support for what3words addresses for precise location identification of individual units within a household.

3. Technical Specifications

- 3.1 Platform
 - Phase 1: Web application (HTML, CSS, JavaScript).

- Phase 2: Android app (convertible via Apache Cordova or similar).
- 3.2 Architecture
 - P2P Model: No central server; data syncs between devices using WebRTC or similar P2P tech.
 - Local Storage: HTML5 LocalStorage or IndexedDB for offline caching.
 - Sync: Triggered by proximity (Bluetooth/Wi-Fi) or internet availability.
- 3.3 User Interface (UI)
 - Design: Simple, bold, intuitive.
 - Green theme (e.g., background-color: lightgreen).
 - Large buttons (e.g., Emergency Alert: red, 20px font, full-width).
 - Tabs clearly labeled (Utilities, Social, Incidents).
 - Responsive: Must work on small screens (e.g., 320px width) for budget smartphones.
 - Mapping: Display household locations on a map.
- 3.4 Sample Web Structure
 - index.html: Home screen with tabs, Emergency button, and map display.
 - verify.html: Member verification page (list pending requests, Approve/Reject buttons).
 - utilities.html: Utilities updates and chat.
 - social.html: Social announcements and chat.
 - incidents.html: Incident reporting form and chat.
 - profile.html: User profile management
- 3.5 Development Tools
 - Frontend: HTML5, CSS3, JavaScript (Vanilla or lightweight framework like Vue.js).
 - Storage: LocalStorage for simplicity. IndexedDB for larger data storage needs.
 - P2P: WebRTC for syncing (fallback to manual sync if complex).
 - App Conversion: Apache Cordova for Android APK.
 - Mapping: Google Maps API, what3words API

4. Constraints and Assumptions

- 4.1 Constraints
 - User Limit: 50 households, 10 members each (500 total).
 - No Internet Reliance: Must function offline with periodic syncing.
 - Budget Devices: Target low-end Android phones (e.g., 2GB RAM, small screens).
- 4.2 Assumptions
 - Residents have basic smartphones with camera/audio capabilities.
 - At least some members meet physically for P2P syncing.

- Community trusts the 60% voting system for verification.
- Users will enable location services for mapping functionality.

5. Deliverables

- Web Application: Fully functional site with all features listed.
- Documentation: User guide for residents (e.g., “How to join Makhi”).
- Android APK: Phase 2 deliverable, post-web completion.
- Source Code: Open for client review and future expansion.

6. Success Metrics

- Adoption: 80% of targeted households (40/50) join within 3 months.
- Usage: At least 10 incident reports or utility updates weekly.
- Safety: One successful emergency alert response (e.g., neighbors intervene or call police).
- Feedback: Positive community response on usability and trust.

7. Next Steps

- Design Phase: Wireframes/UI mockups for client approval, including map integration.
- Development: Build web app, test locally with simulated 50 users.
- Deployment: Distribute to initial street for pilot testing.
- App Conversion: Post-web success, develop Android version.