SAFE-LINK AFRICA

Technical Documentation Manual

1. SYSTEM OVERVIEW

SafeLink Africa is a continent-wide safety ecosystem that connects individuals, families, transport users, communities, and emergency responders through a unified platform combining:

- Emergency Alert & Rapid Response System
- Community Safety Reporting
- Safe Transport Monitoring
- Women & Family Safety Layer
- Smart Home Security (IoT)
- Admin Dashboard for Organizations & Agencies

The system supports Android, iOS, and web. Built with a microservices architecture, real■time communication engine, encrypted databases, and optional IoT hardware.

2. SYSTEM ARCHITECTURE

2.1 High-Level Architecture Components

- Mobile App (React Native / Flutter)
- Backend (Node.js + Python microservices; Firebase optional)
- Real-Time Engine (Socket.IO / WebRTC)
- Video Streaming Server
- Encrypted Cloud Database (PostgreSQL / Firestore)
- IoT Gateway & MQTT Broker
- Admin Console (React / Next.js)
- Government & Security Partner API

2.2 Architecture Style

- Distributed microservices

• Event-driven real-time messaging

• End-to-end encrypted communication

2.3 Core Services

• User Authentication Service

• Emergency Trigger Service

• Video/Audio Streaming Service

• Location Tracking Service

• Reporting & Evidence Storage Service

• Trip Monitoring Service

• Women Safety Service

• IoT Security Service

• Notification & Alert Engine

3. SECURITY DESIGN

3.1 Encryption Model

• AES■256 encryption for emergency data

• RSA■2048 for key exchange

• Encrypted video stream handling

• Secure WebRTC transport

3.2 Privacy Controls

• User manages trusted contacts

• Auto-delete sensitive media (optional)

• Anonymous reporting mode

• Zero-knowledge storage for evidence

3.3 Fraud & Abuse Prevention

• IP/device fingerprinting

• AI-based false alert detection

• Rate-limiting & abuse monitoring

• Community moderator verification

4. MOBILE APP MODULES

4.1 Emergency Alert Module

Features:

• One-tap SOS

• Live video/audio stream

• Auto location tracking

• Panic timer with auto-activation

• Fake gesture SOS

• Offline SMS/USSD fallback

4.2 Community Safety Reporting

• Upload images/video

• Incident tagging

• Anonymous reporting

• Community moderators

• Evidence log

4.3 Safe Transport Monitoring

• Start-trip → End-trip tracking

• Unexpected stop alerts

• Driver ID verification

• Share trip with trusted contact

4.4 Women & Family Safety

• Hidden emergency trigger

• Safe walk map

• Domestic violence silent reporting

4.5 Smart Home IoT

• Door/Window sensors

• Motion detection

• Camera integrations (API)

• Smart alarm push alerts

5. BACKEND SERVICES & API

5.1 Authentication API

• JWT-based authentication

• OAuth for partners

• Device verification

5.2 Emergency Alert API

• POST /emergency/trigger

• POST /emergency/video-stream

• POST /emergency/location

5.3 Reporting API

• POST /report/incident

• GET /report/history

5.4 IoT API

• MQTT messaging

• Sensor event push

5.5 Notification API

• Push notifications

• SMS fallback

6. DATABASE DESIGN

Tables include:

• users

• trusted_contacts

• emergency_alerts

• emergency_media

• location_logs

• transport_trips

• community_reports

• iot_devices

• iot_events

• organizations

• moderators

7. ADMIN DASHBOARD (WEB)

Features:

• Manage reports

• View emergency alerts

• User & organization profiles

• Moderator management

• Heat-map view

• Analytics

8. DEVOPS & INFRASTRUCTURE

• CI/CD pipeline (GitHub Actions)

• Cloud: AWS / GCP

• Auto-scaling for real-time services

• Data backup & disaster recovery

• Monitoring (Grafana/Prometheus)

9. TESTING STRATEGY

9.1 Test Types

• Unit tests

• Integration tests

• Load & stress tests

• Security penetration testing

• IoT hardware testing

9.2 Test Groups

• Individuals

• Families

• Transport users

• Security partners

• Community groups

10. MAINTENANCE & UPDATES

• Monthly updates

• Quarterly feature rollouts

• Security patches

• AI enhancements

• IoT compatibility upgrades

END OF MANUAL