🚨 Hal0 Emergency Alert System – Project Planning & Phases

📌 Overview

This document outlines the project plan and phased development approach for the **Hal0 Emergency Alert System**, a wearable-triggered SOS platform that delivers real-time alerts to contacts and authorities via mobile integration.

🔹 Phase 1: Requirement Analysis & Planning

Goals:

- Understand all functional and non-functional requirements.
- Identify stakeholders (users, emergency contacts, police API).

Tasks:

- Break down the system into modules:
  - Ring communication
  - Mobile app interaction
  - Backend alert processing
  - Notification dispatch
- Define use-case diagrams and success metrics.

🔹 Phase 2: Architecture & Design

Goals:

- Define system architecture and technology choices.

Tasks:

- **Data Flow:** Ring → Mobile App → Backend → Emergency Recipients
- **Key Components:**
  - Alert Dispatcher
  - Location Service
  - Notification Queue

    - Offline Fallback Manager
- **Tech Stack:**
    - Mobile App: React Native
    - Backend: Node.js + Express
    - Queue: Kafka
    - DB: PostgreSQL, Redis, MongoDB
    - Notification: Firebase, Twilio
    - Maps: Google Maps / Mapbox

🔹 Phase 3: MVP Development (Proof of Concept)

Goals:

- Create a minimal functional prototype.

Tasks:

- **Mobile App:**
    - SOS UI (simulate triple tap, biometric alert)
    - Location access
    - Background sync & offline storage
- **Backend:**
    - REST API for alert reception
    - Dispatcher logic
    - Notification engine (mock/simulated)

🔹 Phase 4: Edge Case Handling & Offline Support

Goals:

- Handle weak or no network conditions.

Tasks:

- Store alerts offline in app
- Retry with exponential backoff
- Backend to use Kafka queues for reliability

🔹 Phase 5: Testing & QA

Goals:

- Ensure system reliability and stability.

Tasks:

- Unit & Integration Testing
- Network failure simulations
- Geo-location accuracy tests
- Load testing (simulate 1M users)

🔹 Phase 6: Deployment & Monitoring

Goals:

- Deploy the system and monitor operations.

Tasks:

- Use Docker + Kubernetes
- Cloud deployment (AWS/GCP)
- Monitoring: Prometheus, Grafana, NewRelic
- Logging: ELK Stack or Loki

🔹 Phase 7: Scaling & Future Proofing

Goals:

- Prepare for 1M+ users and global scale.

Tasks:

- Introduce sharded DBs
- Use edge CDN for low latency
- Firebase topic-based alerting
- Research mesh networking (future)

📊 Final Deliverable

Prepare a 1–2 page **System Design Document** with:

- System Architecture Diagram
- Component Overview
- Data Flow
- Tech Stack Justification
- Edge Case Handling Strategy