EPILYRA: TECHNICAL BLUEPRINT & DEVELOPER BRIEF

A Simplified AI Framework for Zero-Latency Syndromic Surveillance
Based on the White Paper by Sohaib Khan (March 2026)

---

1. Project Concept & Innovation: Why EpiLyra is Different

In humanitarian crises, traditional electronic disease surveillance platforms act merely as static digital record keepers. Data is collected by Community Health Workers (CHWs), but it decays in spreadsheets while waiting for human epidemiologists to interpret it.

The Core Innovation: EpiLyra shifts disaster epidemiology from *delayed, manual data interpretation* to **zero-latency, AI-driven syndromic surveillance**. It removes friction for the CHW in the field and uses a two-layered AI architecture to instantly standardize messy field data into ICD-11 codes and automatically trigger geospatial anomaly alerts.

---

2. System Architecture (The 3-Tier Blueprint)

EpiLyra is designed for high-stress, resource-lacking environments. The architecture is divided into three primary tiers:

Tier 1: The Intake Layer (Mobile Client)

- **Users:** Community Health Workers (CHWs).
- **Function:** Frictionless, rapid symptom-data entry.
- **Input Modalities:** Text, Voice Notes, and Images.

Tier 2: The Simplified AI Layer (Core Engine)

- **Sub-Layer A (Semantic Parser):** An LLM-driven microservice that ingests unstructured field data (voice, text, images) and standardizes it into WHO ICD-11 classifications.
- **Sub-Layer B (Spatial Anomaly Trigger):** A geospatial algorithm that continuously monitors baseline health data in a specific radius and detects statistical anomalies.

Tier 3: The Epidemiologist Dashboard (Alerting & Analytics)

- **Users:** Human Epidemiologists and Decision Makers.
- **Function:** Receives automated alarms generated by the Spatial Anomaly Trigger. Displays clean, standardized data mapped to emerging outbreak patterns (e.g., Cholera).

---

3. Design Instructions for Developer: The Intake Logic (UI/UX)

Current digital tools fail because they assume the user has time. A CHW in a disaster zone is depleted and overloaded. **Friction must be zero.**

- **Adaptive Syndromic Branching:** The UI must *not* be a long, static form. It must be built as a dynamic decision tree.
- **Top-Threat Matrix:** The home screen should display a simple matrix of primary syndromes based on the **Sphere Project's minimal health action benchmarks**.
- **Context-Aware UI:** When a user clicks a primary syndrome, the app should instantly open specific secondary options tailored to the *known epidemiology of that specific population and area*.
- **Multimodal Input:** The UI must prominently feature a "Voice Record" and "Camera" button to allow CHWs to bypass typing entirely.

---

4. Core AI Algorithms: The "Clarity Engine"

Developers must implement the following two AI layers to process the data flowing from the Intake Layer.

Layer 1: The Semantic Parser (Standardization)

- **Objective:** Convert messy, unstructured field data into clean, standardized data.
- **Mechanism:** Deploy a Large Language Model (LLM) fine-tuned on the **WHO's International Classification of Diseases (ICD-11)**.

- **Workflow:**
    1. CHW uploads a voice note describing symptoms.
    2. Speech-to-Text (STT) converts audio to text.
    3. The LLM parses the text, identifies the most plausible disease, and assigns the exact ICD-11 code.
    4. The data is instantly aligned and saved to the database.

Layer 2: The Spatial Anomaly Trigger (Alerting)

- **Objective:** Detect local outbreaks before they spread.
- **Mechanism:** A continuous geospatial monitoring algorithm.
- **Workflow:**
    1. The system establishes a "baseline" of normal symptom reports within a specific geospatial radius.
    2. The algorithm continuously calculates the variance from this baseline.
    3. If a sudden spike in specific ICD-11 codes (e.g., acute watery diarrhea) occurs within a tight radius, the system generates an immediate, automated alarm for the epidemiologist.

---

5. Data Sovereignty & Ethical Framework

Vulnerable populations in humanitarian crises face severe risks regarding digital privacy. EpiLyra must be engineered with strict data sovereignty protocols.

- **Local PII Lockdown:** Personally Identifiable Information (PII) must be locked down at the local device level.
- **No Centralized PII Storage:** The system must **never** centralize or store PII in the cloud.
- **Edge Processing:** Any data sent from the CHW's device to the AI layers or the Epidemiologist Dashboard must be strictly anonymized, containing only syndromic data, timestamps, and generalized geospatial coordinates.

---

6. Recommended Tech Stack (Developer Addendum)

(Note: Based on the white paper's requirements, the following modern stack is recommended to achieve EpiLyra's goals)

- **Frontend / Mobile (Intake Layer):** Flutter or React Native (for cross-platform deployment). Must include robust offline-first capabilities (e.g., WatermelonDB or SQLite) to store data when connectivity drops.
- **Multimodal Processing:**
    - *Voice:* OpenAI Whisper API (or a lightweight on-device STT model) for transcribing voice notes.
    - *Images:* Vision-capable LLM (e.g., GPT-4o or open-source equivalent like LLaVA) for analyzing visual field data.
- **AI Semantic Parser (Backend):** Python (FastAPI). Integration with an LLM (via LangChain or LlamaIndex) utilizing Retrieval-Augmented Generation (RAG) over the official WHO ICD-11 API/database to ensure accurate coding.
- **Geospatial Database & Anomaly Detection:** PostgreSQL with **PostGIS** for radius-based querying. Python (SciPy/Scikit-learn) for time-series anomaly detection (e.g., ARIMA, Isolation Forests, or DBSCAN for spatial clustering).
- **Security:** On-device encryption for PII. Strict payload stripping before syncing to the cloud to ensure compliance with the IASC Operational Guidance on Data Responsibility.