# **CHAPTER 1: INTRODUCTION**

## 1.1 Project Description

## Basic Introduction of the Project

The increasing popularity of outdoor activities such as trekking and expedition in remote regions has highlighted the critical need for reliable, real-time geolocation and emergency communication solutions. Traditional mobile and GPS tracking apps are fundamentally limited by the unavailability of cellular network infrastructure in such environments. This project pioneers the development of an offline, energy-efficient LoRa-based location and emergency tracking system designed specifically for trekkers and group safety in off-grid terrains.

The system comprises multiple portable nodes and a central gateway, utilizing the LoRa wireless communication protocol to enable long-range, low-power, and robust data exchange. Each node, carried by individual trekkers, incorporates an ESP32 microcontroller, NEO-6M GPS module, BMP280 sensor for environmental monitoring, LoRa transceiver, SD card storage, and an emergency alert button. These nodes periodically collect data points—longitude, latitude, altitude, temperature, pressure, battery percentage, RSSI (Received Signal Strength Indicator)—and transmit the information to a central node using JSON format.

The central node, equipped with its own GPS module, LoRa transceiver, SD card storage, and web server capability, acts as the system's aggregator and interface. All received data is logged locally in data.txt files and supports offline mapping by accessing stored map tiles to display dynamic location markers and environmental data. The entire system is accessible to users via any mobile device or laptop, simply by connecting to the ESP32 gateway's Wi-Fi access point—eliminating the need for external internet connectivity.

## Detailed Project Description

## Problem Statement

Safety risks for trekkers in remote environments stem from:

* Lack of real-time communication and location-sharing with fellow trekkers or guides
* Delayed response to emergencies due to communication blackouts
* Inability to continuously monitor group members' location, environmental data, and battery status
* Challenges in storing and visualizing geospatial data without cloud resources

## Project Objectives

* Design and implement sensor nodes capable of collecting accurate GPS, environmental, and battery data
* Enable reliable, low-power communication over several kilometers using LoRa technology
* Develop a central node (gateway) capable of aggregating, storing, and displaying data completely offline
* Integrate an emergency alert system for rapid, local response
* Build a user-friendly web interface accessible via local Wi-Fi for live monitoring and historic data review

## System Workflow

1. **Data Collection:**  
   Each node collects location, altitude, temperature, pressure, battery percent, RSSI, and emergency events.
2. **Packet Creation:**  
   Sensor data is structured as JSON packets for modularity and scalability.
3. **LoRa Communication:**  
   Packets are wirelessly transmitted using LoRa from all nodes to a single gateway node in a star topology.
4. **Data Logging:**  
   The gateway receives all packets and stores them in SD card (data.txt) files.
5. **Local Visualization:**  
   The gateway hosts a Wi-Fi access point and runs a local web server. When users connect via smartphones or laptops, they can:
   * View live map with markers for all tracked nodes (map tiles read from SD card)
   * Access historical logs and telemetry for all nodes
   * Monitor environmental trends and trigger/acknowledge alerts directly from their device
6. **Emergency Handling:**  
   Emergencies (triggered by the alert button or low battery) are immediately prioritized and displayed on the local web dashboard to all users connected to the gateway.

## Key Technologies and Components

* **ESP32 Microcontroller:**  
  Provides powerful processing, Wi-Fi AP mode, sleep modes for energy savings, and interfaces with SPI (SD card), UART (GPS), and I2C (BMP280).
* **LoRa Transceiver (e.g., SX1278):**  
  Enables secure, long-range wireless data transfer in the ISM band with low power draw.
* **NEO-6M GPS Module:**  
  Supplies real-time geospatial coordinates and altitude.
* **BMP280 Sensor:**  
  Measures ambient temperature and barometric pressure.
* **SD Card Storage:**  
  Facilitates local data logging and map tile hosting, ensuring full offline operation.
* **Emergency Button:**  
  Allows users to instantly raise alerts and mark packets as critical.
* **Web Dashboard (HTML/JavaScript):**  
  Presents data in an accessible map, tabular, and graphical format without any internet dependency.

## Theory and Concept Relevant to the Project

## LoRa Communication

LoRa (Long Range) is a physical radio communication technique based on spread spectrum modulation, suitable for low-power, long-distance wireless links. Its key features include:

* Operation in unlicensed ISM frequency bands (e.g., 868/915 MHz)
* Ability to communicate over several kilometers even with minimal infrastructure
* Adaptive data rates, robust error correction, and power-efficient operation make it ideal for sensor networks in remote environments

## GPS-Based Tracking

GPS modules provide global latitude, longitude, altitude, and precise time. When embedded in mobile sensor networks, GPS information allows for real-time location tracking, proximity detection among nodes, distance calculations (using the Haversine formula), and route mapping—even in offline scenarios.

## Environmental Sensing with BMP280

Barometric pressure and temperature data from the BMP280 sensor allow trekkers to monitor environmental trends such as weather changes or altitude profiles. This enriches situational awareness and supports real-world decision-making during expeditions.

## Local Web Server and Offline Visualization

Hosting a web server directly on the ESP32 gateway transforms the system into a fully self-contained solution. By using SPIFFS or microSD cards to store both incoming data and map tiles, the gateway can deliver real-time dashboards, graphs, and maps to any Wi-Fi-enabled device, maintaining full functionality even when completely disconnected from the internet.

## Power Management

To maximize autonomy, ESP32 nodes implement deep sleep between transmissions, dynamic polling intervals, and low power optimization throughout all hardware and firmware processes. This ensures multi-week field endurance on a single battery charge.

# CHAPTER 2:LITERATURE REVIEW

## 2.1 Literature Survey

This section presents a comprehensive review of 20 key references relevant to LoRa-based location tracking, IoT sensor networks, environmental telemetry, offline data visualization, and group safety systems for trekkers. Each reference is summarized to highlight its specific contribution, outcomes, and relationship to this project. The complete path or DOI is included for further consultation.

## *Table 2.1:Reference Survey Table*

|  |  |  |
| --- | --- | --- |
| **Ref. No.** | **Reference** | **Key Outcome / Relevance to Project** |
| 1 | L. Centenaro et al., IEEE Wireless Comm., 2016 | Establishes LoRa as ideal for long-range, low-power communication, validating its use in infrastructure-less tracking. |
| 2 | F. Adelantado et al., IEEE Comm. Mag., 2017 | Highlights LoRaWAN’s range, scalability limits; informs design constraints for optimal node deployment. |
| 3 | T. Banerjee, A. Nair, PR Letters, 2023 | Demonstrates GPS + LoRa mesh in outdoor tracking; useful for topology evaluation and system comparison. |
| 4 | S. Das, M. Sinha, J. Comm. Tech., 2022 | Explores hybrid LoRa network designs for remote sensing, supporting modular deployment strategies. |
| 5 | A. Sharma, K. Ramesh, J. Data Intelligence, 2023 | Validates energy-efficient practices like deep sleep in remote LoRa deployments; supports extended node uptime. |
| 6 | K. Singh, M. Joshi, Int. J. Web Appl., 2023 | Describes web dashboard for LoRa data visualization; guides user interface for offline map display. |
| 7 | N. Komninos et al., IEEE ICCSE, 2023 | Discusses LoRaWAN in emergency alerting; supports timely and energy-efficient notifications in remote settings. |
| 8 | Y. Mehrotra, H. R. Swamy, J. AI Data Sci., 2024 | Power optimization via sleep scheduling; confirms feasibility of long-duration battery-powered tracking. |
| 9 | R. Bhattacharya, D. Sen, IEEE ICCSE, 2023 | Compares mesh vs. star topology for alerts; supports the chosen star topology for simplicity and stability. |
| 10 | LoRa Alliance, LoRaWAN 1.1 Spec, 2017 | Technical base for LoRa configuration and compliance; essential for correct radio parameter settings. |
| 11 | Meshtastic Docs, 2024 | Real-world LoRa mesh protocol guide; useful for interface features like alerts and peer tracking. |
| 12 | O. Charif et al., ITM Web Conf., 2023 | Evaluates LoRa performance (range, delivery) in practice; guides expectations for field deployment. |
| 13 | M. A. Kamal et al., Comp. Intel. Neurosci., 2023 | Summarizes deployment challenges (e.g., antenna, NLOS); assists in hardware placement and tuning. |
| 14 | J. Haxhibeqiri et al., Sensors, 2018 | Covers broad LoRaWAN IoT applications; context for power-saving and scalable architectures. |
| 15 | P. Ferrari et al., Sensors, 2023 | Provides field data for LoRa RSSI and packet loss; benchmarks system performance in real-world scenarios. |
| 16 | L. M. Pires et al., Sensors, 2024 | Shows LoRa + SD card logging; supports design of local data storage for offline dashboard. |
| 17 | Y. T. Ting et al., Sensors, 2024 | Transmission parameter tuning improves performance and efficiency; applied in protocol optimization. |
| 18 | A. Augustin et al., Sensors, 2016 | Comparative study of LPWANs; reinforces LoRa as the best-fit for rural IoT tracking. |
| 19 | Arduino ESP32 Core & LoRa Library Docs | Implementation guide for ESP32 + LoRa coding; directly used for firmware development. |
| 20 | K. Sundaresan et al., Embedded Sys. Conf., 2023 | Offline map visualization techniques for SD card and microcontroller; aligns with the system’s dashboard design. |

## Reference Outcomes and Project Relevance

Each reference provided new technical or practical insights crucial for the project:

* **LPWAN/LoRa Fundamentals:** establish the foundational theory and validate the choice of LoRa for robust, energy-efficient, long-range communication in low-infrastructure outdoor environments.
* **System Implementation Guides:** offer architectural, practical, and firmware support for multi-node deployment, GPS data handling, and mesh/star topology comparison, informing node and gateway design.
* **Energy and Power Management:** inform strategies for optimizing battery life using deep sleep, adaptive polling, and efficient radio tuning matching the multi-week autonomy goal.
* **Emergency Alert and Data Integrity:** highlight star topology efficacy, group alert systems, and the challenges of packet delivery in real-world conditions.
* **Offline Processing and Visualization:** contribute to methods for local storage on microSD (data logs, map tiles), and browser-based data display over local Wi-Fi, central to the project’s no-internet requirement.
* **IoT Application Surveys:** provide context and validation across environmental, tracking, and safety applications, fortifying design decisions.
* **Sensor and Hardware Utilization:** confirm sensor accuracy, wiring, hardware interface best practices, and real-world environmental resilience.

## Unresolved Issues and Emerging Opportunities

Despite significant research and practical deployments, certain challenges remain. Handling high-density node networks still poses issues in collision management and bandwidth, especially in infrastructure-free, offline systems. Environmental interference, multipath fading, and NLOS (non-line-of-sight) obstacles can still degrade communication reliability, particularly in dynamic trekker groups. Power budgeting for combined GPS, sensor, LoRa, and web server operation may be further optimized through hybrid harvesting (e.g., integrated solar) or more intelligent, event-triggered polling. The management of map tile storage, visualization efficiency on constrained hardware, and low-latency emergency delivery also present emerging opportunities for system refinement and research.

Recent advances in mesh routing, context-aware user interfaces, and lightweight embedding of AI for anomaly/event detection within nodes are promising areas to further enhance group safety, extend operational autonomy, and reduce false positives in alerting, representing key directions for future development.

## Conclusion and Motivation from Literature Survey

The surveyed literature consistently demonstrates that LoRa technology, combined with low-power microcontrollers, environmental sensors, and context-aware designs, is exceptionally well suited for offline group tracking and emergency alerting in remote environments. The recurring success of star and mesh topologies, validated in real deployments for trekker safety, environmental logging, and fleet management, affirms the reliability, autonomy, and flexibility of such systems. As user-centric needs for robust, internet-independent safety platforms increase, the motivation to integrate advanced offline mapping, local data logging, and autonomous alerting into a single, mobile gateway is both timely and impactful. This lays the foundation for the present project, which draws directly from these works to deliver a practical, field-ready, and uniquely local LoRa-based safety and telemetry platform for trekkers.

## 2.2 Existing and Proposed System

## Problem Statement

**"How can groups of trekkers in remote, non-connected regions achieve reliable real-time location sharing, environmental monitoring, and rapid emergency response without depending on cellular or cloud-based services?"**

## Scope of the Project

* Enable robust, local-only wireless communication and safety solutions for trekking groups beyond cellular service.
* Offer fully offline, real-time location tracking, environmental data logging, and group-wide alerting using energy-efficient, portable hardware.
* Provide intuitive, browser-based status views and mapping accessible on any mobile device without the internet.

## Methodology Adopted

1. **Star Topology Design:**  
   Four nodes (ESP32 + LoRa + BMP280 + GPS + button) transmit JSON-formatted telemetry to a single central gateway node (ESP32 + LoRa + SD card + GPS).
2. **Data Aggregation & Storage:**  
   The gateway receives, parses, and logs all data in SD card files. Map tiles are also stored locally for offline use.
3. **Visualization & Interface:**  
   ESP32 gateway runs a local Wi-Fi AP, providing a web server that displays real-time and historical data, location markers, and environmental readings on map tiles.
4. **Alert & Power Management:**  
   Emergency events (button/battery triggers) are prioritized and highlighted on the dashboard. All nodes use deep sleep and efficient transmission/reception intervals for longevity.

## Technical Features of the Proposed System

Four to five key unique features identified in the project include:

* **Pure Offline Operation:**  
  Complete local data display and alerting, no dependence on cloud or internet.
* **Real-Time Location & Environmental Telemetry:**  
  Each node provides detailed live sensor data in structured JSON, supporting granular group tracking.
* **Local Map Visualization:**  
  Gateway hosts map tiles from SD card, allowing live and historical positions to be shown on basemaps with high responsiveness.
* **SD Card Logging for Data Persistence:**  
  All telemetry is locally stored, supporting later retrieval and analysis even without connectivity.
* **Group Emergency Alert System:**  
  Includes hardware button and low-battery triggers for instant, collective notification.

## 2.3 Tools and Technologies Used

* **Platform / Tools:**
  + **Microcontrollers:** ESP32
  + **LoRa Transceiver Modules:** SX1278 433mhz
  + **Sensors:** BMP280 (temperature, pressure), NEO-6M GPS
  + **Storage:** microSD card (FAT32)
  + **Programming Languages:** C++ (Arduino IDE), JSON for data exchange
  + **Web Server/Visualization:** ESPAsyncWebServer, HTML5/CSS, JavaScript (Leaflet for visualization)
  + **Map Tiles:** OpenStreetMap (offline, pre-processed and stored)

**2.4 Hardware and software Requirement**

* **Hardware Requirements:**
  + 5 ESP32 boards (4 nodes, 1 gateway)
  + 5 LoRa modules (SX1278)
  + 5 BMP280 sensors
  + 5 NEO-6M GPS modules
  + 1 microSD card
  + Power sources: 18650 Li-ion battery
  + Emergency push buttons
* **Software Requirements:**
  + Arduino IDE
  + Web browser on client devices (smartphone/tablet/laptop)
  + SD card preparation tool

# CHAPTER 3: SOFTWARE REQUIREMENT SPECIFICATIONS

## 3.1 Introduction

This chapter defines the hardware and software requirements, platforms, and tools adopted for the implementation of a fully offline, LoRa-based location tracking and emergency alert system for trekkers. It describes terminology, an overview of system specifications, user interactions, and technical constraints essential for effective system engineering and operational clarity.

## *Table 3.1:Definitions, Acronyms and Abbreviations*

| Term | Full Form / Description |
| --- | --- |
| LoRa | Long Range; low-power wireless protocol for long-distance communication |
| ESP32 | Embedded microcontroller with Wi-Fi/BT/processing capabilities |
| GPS | Global Positioning System; satellite-based location data |
| BMP280 | Barometric Pressure & Temperature sensor |
| AP | Access Point; Wi-Fi server mode for user device connection |
| SD Card | Secure Digital card for storage of map tiles/log data |
| JSON | JavaScript Object Notation, human-readable data format |
| RSSI | Received Signal Strength Indicator |
| UI | User Interface |
| HTTP | Hypertext Transfer Protocol |
| SPI | Serial Peripheral Interface (hardware bus) |

## Overview

The system integrates embedded nodes equipped with LoRa, ESP32, GPS, BMP280, SD card, and an emergency button, forming a star topology to transmit structured data to a central ESP32 gateway. The gateway aggregates, logs, and visualizes telemetry and provides a local web server accessible over Wi-Fi.

## 3.2 General Description

## Product Perspective

The system is a real-time, offline location and emergency alert solution composed of several end-nodes and a central gateway, specifically crafted for operation in non-connected, outdoor environments. Unlike typical GPS tracking or IoT ecosystems dependent on cellular/cloud infrastructure, it performs all communication and visualization locally.

## Product Functions

* Wireless acquisition of location, sensor, and status data from distributed nodes
* Local aggregation and logging to SD card at the gateway
* Dynamic map-based visualization using stored map tiles
* Emergency alert handling (button and battery triggers)
* User interaction via any web browser connected to the gateway’s Wi-Fi AP

## User Characteristics

* Trekkers, guides, or group leaders in remote/outdoor settings
* Users operate mobile phones, tablets, or laptops with Wi-Fi and browsers
* No need for technical skills or internet access; system is designed for plug-and-play usability

## General Constraints

* Operation limited by the maximum LoRa communication radius (typically 600–2,000 m depending on terrain)
* Web dashboard accessibility is constrained by Wi-Fi AP range (~100 m from gateway)
* Data storage limited by the SD card’s capacity
* Number of wireless clients limited by ESP32 AP (typically 8–10)

## Assumptions and Dependencies

* End-nodes and the gateway remain within mutual LoRa range for reliable communication
* User devices support modern web browsers (HTML5/JS)
* Map tiles are preloaded to SD card; no live internet-based tile fetching
* Battery maintenance and periodic SD card data management are observed by users

## 3.3 Functional Requirements

## Introduction

Functional requirements specify the key operations and data flows governing the behavior of each hardware/software module.

## Input

* Sensor node:
  + GPS module continuously provides latitude, longitude, altitude
  + BMP280: temperature and barometric pressure readings
  + Battery voltage measured via ESP32 ADC
  + Emergency trigger via button press or low battery
* Gateway:
  + Receives JSON packets via LoRa
  + User requests from browser (dashboard views, map, data export)

## Processing

* Sensor node:
  + Collects sensor data at set intervals
  + Encodes as JSON: {longitude, latitude, temperature, pressure, altitude, battery %, emergency, RSSI}
  + Transmits packet via LoRa
* Gateway node:
  + Decodes and parses received JSON
  + Logs entries to SD card (data.txt)
  + Updates real-time memory model for web interface
  + Loads and serves offline map tiles from SD card
  + Computes location markers and displays status/alerts

## Output

* Sensor node:
  + Transmits wireless payload to gateway
* Gateway node:
  + Responds to browser requests with dynamic dashboard (location, environmental history, alerts, logs)
  + Displays group positions on offline map
  + Provides alerts (visual banners) for emergency events
  + Allows data download/export

## 3.4 External Interface Requirements

## User Interfaces

* Responsive HTML5/CSS/JS web dashboard, accessible from any Wi-Fi-enabled device (no app installation required)
* Real-time map with node markers, sensor telemetry tables, alert banners, and simple controls for data/history viewing

## Hardware Interface

* SPI for SD card data storage
* UART/I2C for GPS and BMP280 sensor communication
* GPIO for emergency button input
* LoRa (SPI): long-range radio communication between all nodes and the gateway
* Wi-Fi (AP mode): provides local wireless network for users

## Software Interface

* Arduino/ESP-IDF APIs for hardware abstraction
* LoRaWAN/LoRa libraries for packet framing and RF control
* SPIFFS or SD card file systems for local file management
* Web server library (e.g., ESPAsyncWebServer) for HTTP serving and REST endpoints
* No external plugins; system is fully self-contained

## 3.5 Non-Functional Requirements

* **Reliability:** System maintains operation in intermittent RF/harsh environments; persistent logging to SD card ensures traceability
* **Performance:** Responsive dashboard (<1s load time), live data updates within 1–2 s from event to UI display
* **Scalability:** Up to 5 nodes (expandable with hardware/resource optimizations)
* **Fault Tolerance:** Graceful handling of missing node signals or power failures; data logs are robust to device resets
* **Security:** Wi-Fi AP can be password-protected; data is only exposed to local clients
* **Usability:** Browser-based interface with intuitive layout; map, tables, and alerts are clearly visible without training
* **Maintainability:** Modular firmware; SD card files are human-readable for diagnostics/debugging

## 3.6 Design Constraints

## Standard Compliance

* ISM band operation for LoRa (868/915 MHz, per region)
* Follows SPI, I2C, UART hardware specifications
* Local Wi-Fi AP uses WPA2 security
* JSON packet format for flexible data exchange

## Hardware Limitations

* ESP32 AP supports limited number of concurrent Wi-Fi clients
* SD card read/write endurance affects long-term logging
* Microcontroller and radio resources restrict support for advanced web UI features (e.g., heavy JavaScript libraries)
* Limited flash/RAM on ESP32 for map tiles and buffering

## Other Requirements

* All offline files (map tiles, static web assets) must fit on the SD card or SPIFFS partition
* Emergency button must be accessible and reliable under outdoor conditions
* Enclosures for nodes and gateway must be weather-resistant for field reliability
* Regular battery recharging is required; system must provide low battery alert both in UI and at node level

## Hardware and Software Specifications Table

## *Table 3.2:Hardware and Software Specifications Table*

|  |  |
| --- | --- |
| Category | Specification |
| Microcontroller | ESP32-WROOM-32 (dual-core, Wi-Fi, 520 KB RAM) |
| Radio Protocol | LoRa (SX1278 ) |
| Sensors | NEO-6M GPS module; BMP280 temperature/pressure sensor |
| Data Storage | microSD card (≥8 GB recommended) |
| Power Supply | Li-ion battery (4000 mAh typical per node) |
| Local Networking | ESP32 Wi-Fi AP mode, WPA2 security |
| Web Visualization | Embedded HTML5/CSS/JavaScript dashboard via ESPAsyncWebServer |
| File System | SPIFFS (internal) / SD card (external, FAT32) |
| Programming Tools | Arduino IDE / PlatformIO, C++, Arduino & ESP-IDF frameworks |
| Data Format | JSON (node→gateway), plain text logs (gateway→storage) |
| Map Visualization | Leaflet.js/OpenLayers with local map tiles |
| Supported User Devices | Any smartphone, tablet, or laptop with Wi-Fi + modern browser |

**CHAPTER 4: SYSTEM DESIGN**

## 4.1 System Perspective / Architectural Design

## Problem Specification

The project aims to provide trekkers with a reliable, real-time, and fully offline location tracking and emergency alert system in remote environments lacking cellular or internet connectivity. The system must be energy-efficient, robust against environmental factors, able to collect and transmit environmental and geospatial data, and offer a user-friendly interface for both real-time monitoring and historical data visualization.

**Key goals include:**

* Real-time, multi-node data aggregation and visualization using LoRa wireless technology
* Offline data logging and local map-based dashboard on an ESP32 gateway web server
* Immediate group-wide emergency alerting via hardware triggers
* Intuitive access for end-users via smartphones or laptops connected to local Wi-Fi
* **Nodes**: Multiple; transmit sensor and status data via LoRa
* **ESP32 Gateway**: Collects data, stores it, manages local Wi-Fi/AP, serves dashboard and maps
* **User Devices**: Connect to gateway’s Wi-Fi and view dashboard in browser

## High-Level Block Diagram

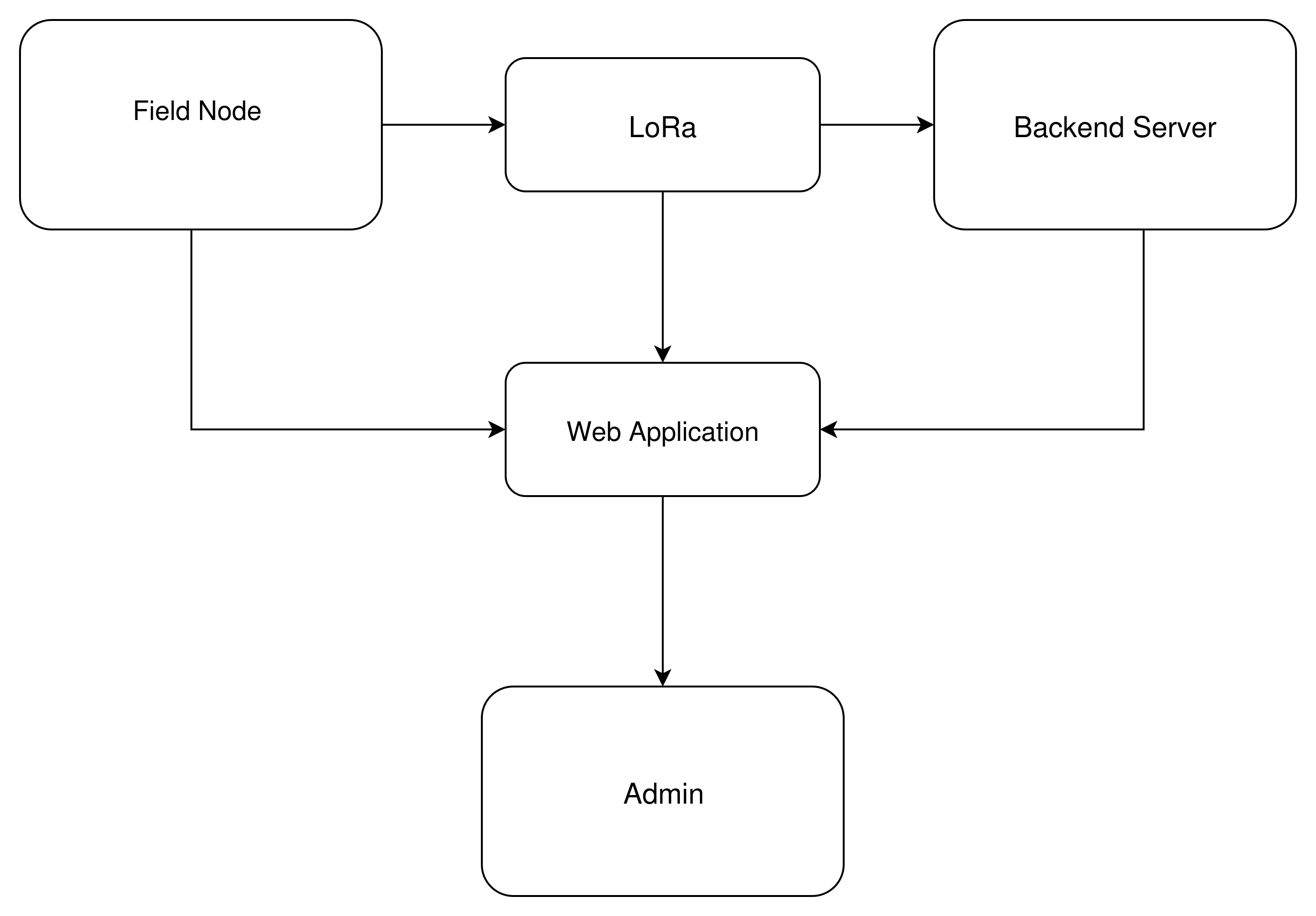


Fig4.1: Architectural diagram

## *Table 4.1:sensor data type and description*

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | Data Type | Source | Description |
| **node\_id** | String | Node | Unique identifier for each node |
| **timestamp** | DateTime | Node | Time of data transmission (ISO format) |
| **latitude** | Float | GPS Module | Latitude coordinate in decimal degrees |
| **longitude** | Float | GPS Module | Longitude coordinate in decimal degrees |
| **altitude** | Float | GPS Module | Altitude in meters |
| **temperature** | Float | BMP280 | Ambient temperature in °C |
| **pressure** | Float | BMP280 | Barometric pressure in hPa |
| **battery** | Integer(%) | Node | Battery charge level in percent |
| **rssi** | Integer | LoRa RX | Received Signal Strength Indicator (dBm) |
| **emergency\_flag** | Boolean | Button/SW | "True" if emergency button pressed or battery low, else "False" |

## **Module Specification**

**1. Node Module**

* **Inputs:** GPS data, temperature/pressure sensor, emergency button, battery ADC reading
* **Processing:** Collects sensor data at set intervals, builds JSON packet, transmits via LoRa
* **Outputs:** Wireless JSON data packet

**2. LoRa Communication Module**

* **Handles**: Long-range, low-power wireless packet transmission from nodes to gateway on ISM band

**3. Gateway Module (ESP32)**

* **Inputs:** LoRa packets from nodes, user HTTP/WebSocket requests
* **Processing:** Parses, logs, and indexes data to SD card; loads and stores map tiles; calculates distances; builds web dashboard from local data; processes and visualizes emergency states
* **Outputs:** Serves dashboard and map interface via local Wi-Fi AP; files for download

**4. Web Interface Module**

* **Inputs:** User device browser requests
* **Processing:** Renders live dashboard, plots markers, Tabular/graphical sensor data, displays alerts; supports data/history queries
* **Outputs:** Interactive HTML/JS/CSS dashboard viewable on any modern browser

**5. Emergency Handling Module**

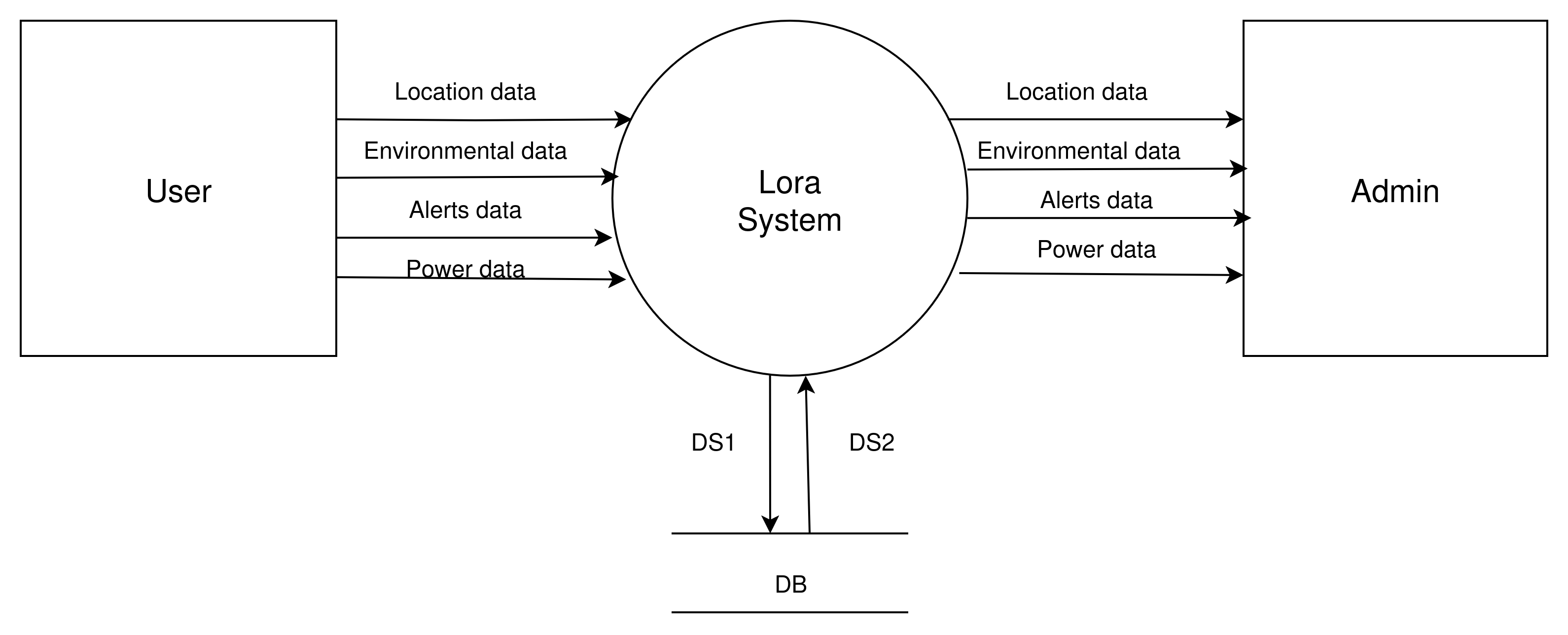
* **Inputs:** Emergency flag in node packet
* **Processing: Prioritizes, records, and visually highlights emergency events on dashboard; enables acknowledgment**

## Assumptions Made

* All nodes remain within LoRa range of the central gateway node during operation.
* ESP32 gateway Wi-Fi AP signal covers the trekking group vicinity (up to approx. 100 meters).
* All user devices support modern web browsers and Wi-Fi connection.
* Map tile data is preloaded on SD card and does not require online fetching.
* Emergency button and related logic are hardware-debounced and tested for reliability.
* Battery management is sufficient for expected trip duration; users are trained to recharge as needed.
* Data packets are timestamped consistently for accurate temporal logging and visualization.
* The ESP32 gateway remains operational (battery or external power) for the intended deployment duration.

## **4.2 Context Diagram**

The context diagram below provides a high-level, process-centric view of the system, depicting all primary external entities, input/output flows, and the system boundary.



*Fig 4.2:context diagram*

**Inputs:**

* Sensor nodes periodically transmit JSON data packets (location, environmental, alert info) to the gateway.

**Processing:**

* Gateway logs/updates data, stores to SD card, hosts map tiles and dashboard, processes emergencies, and answers user dashboard queries.

**Outputs:**

* Live map and dashboard with all telemetry, status, and alerts, displayed in the user’s browser when connected via Wi-Fi AP.

# **CHAPTER 5: DETAILED DESIGN**

## 5.1 System Design Overview

The proposed LoRa-based Offline Location Tracking System employs an object-oriented approach due to its modularity, extensibility, and maintainability. Object orientation allows encapsulation of sensor node behavior, gateway functions, and user interactions, which makes feature additions, code reuse, and future scalability efficient. The design is structured across three stages: object modeling, dynamic modeling, and functional modeling to comprehensively describe the system’s core architecture.

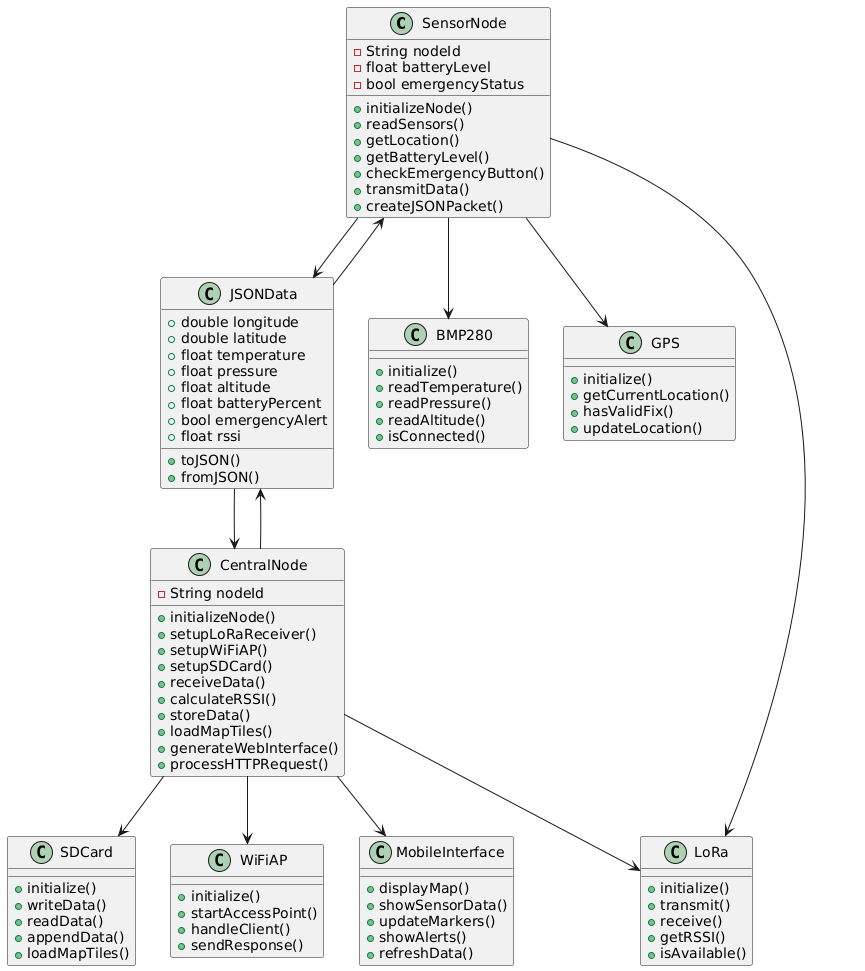
## 5.2 Object Modeling

## Brief Description

Object modeling defines the static structure of the system by identifying the key objects, their attributes, and the relationships among them. In the proposed LoRa-based Offline Location Tracking System, object modeling mirrors real-world components and interactions using an object-oriented paradigm, which significantly enhances modularity, code reuse, and future scalability. The system comprises several core classes, each representing a crucial component in the tracking and monitoring workflow.

The **Node** object represents an individual wearable device attached to a trekker or personnel. Each node includes sensors such as GPS and BME280, a LoRa transceiver, and an ESP32 microcontroller. It has attributes like nodeID, location, temperature, humidity, batteryStatus, and signalStrength. Its methods include collecting sensor data, sending data to the gateway, and triggering an emergency alert. The **Gateway** class represents the central LoRa receiver node that manages data collection from all deployed nodes. It maintains attributes such as gatewayID, receivedPackets, connectedNodes, and dataBuffer, and provides methods to receive, parse, store data, and broadcast alerts in case of emergencies.Its methods include data validation and formatting for storage. In critical situations, the **EmergencyAlert** class is used to manage alerts generated by the nodes, such as a fall or manual .

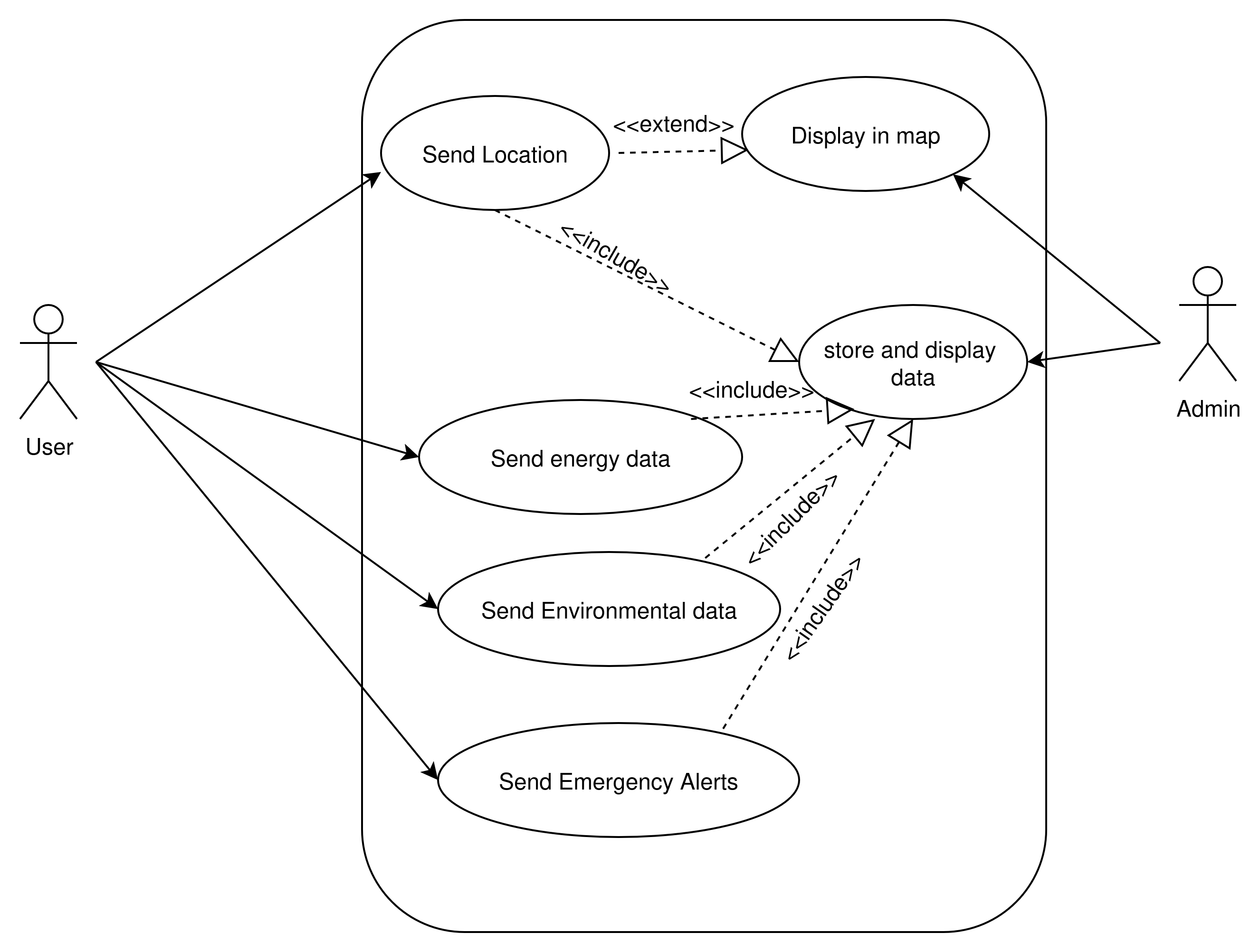
## **5.2.1 Class Diagram**

*Figure 5.1: Class Diagram for the Offline LoRa Location Tracking System*

## **5.3 Dynamic Modeling**

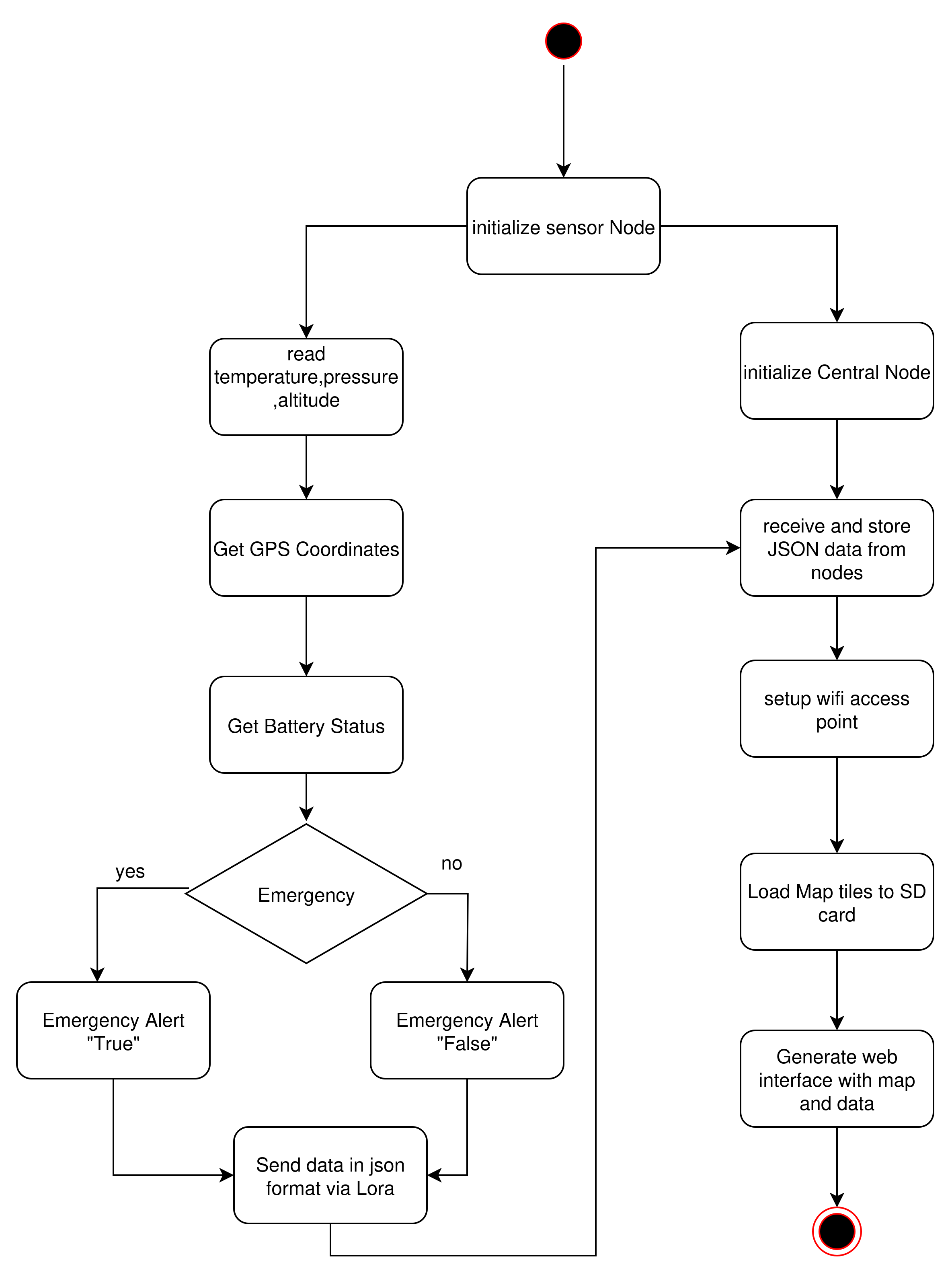
Dynamic modeling illustrates how objects interact and change over time through system use-cases.

## 5.3.1 Use Case Diagram

*Figure 5.2: Use Case diagram*

This diagram visualizes interactions between principal actors (User, Node, Gateway) and their roles in data transmission, emergency management, and system usage.

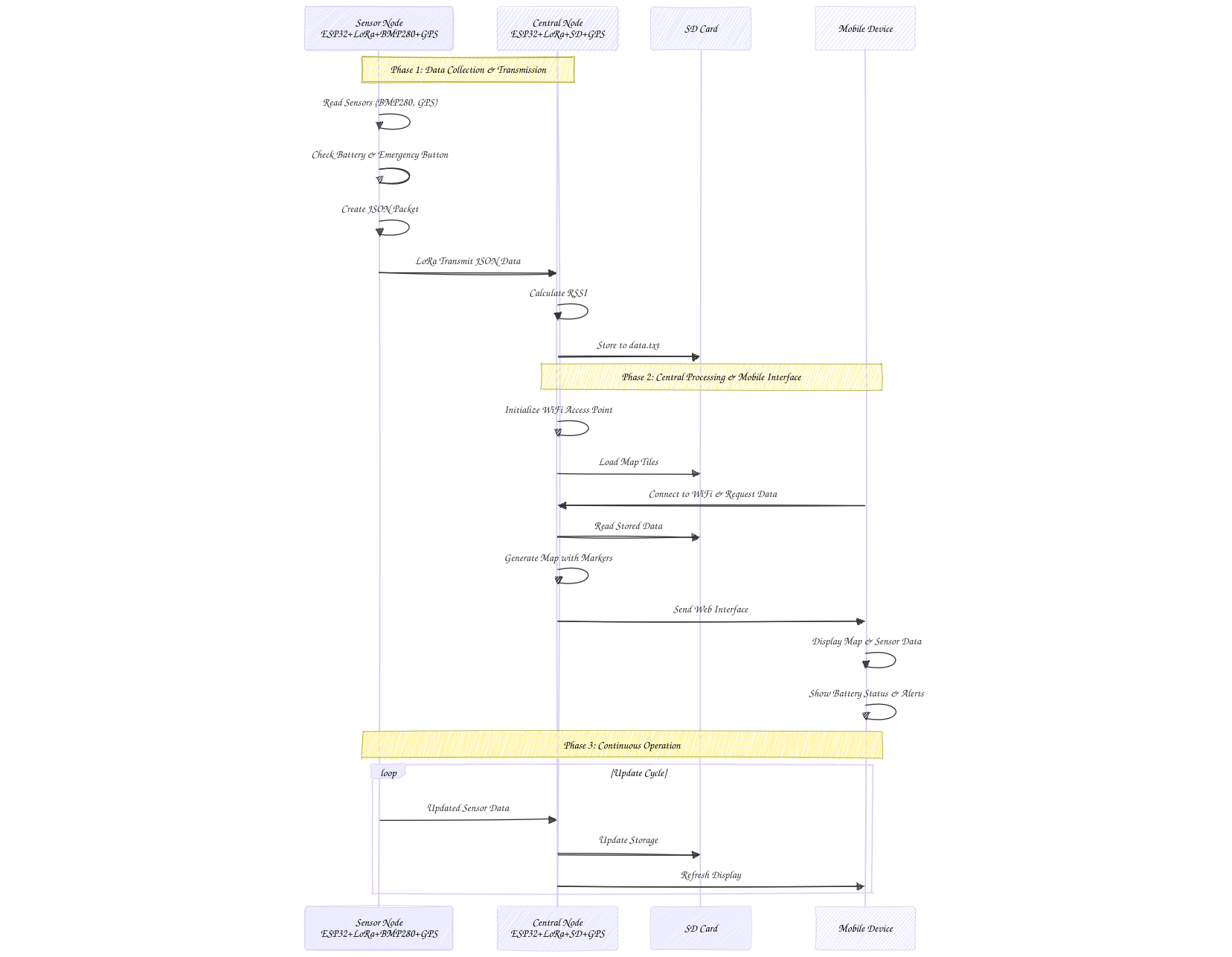
## **5.3.2 Activity Diagram**

****

*Figure 5.3: Activity Diagram for Data Acquisition to Dashboard*

Describes the end-to-end process from sensor data collection at the node to user visualization on the local dashboard.

## **5.3.3 Sequence Diagram**

****

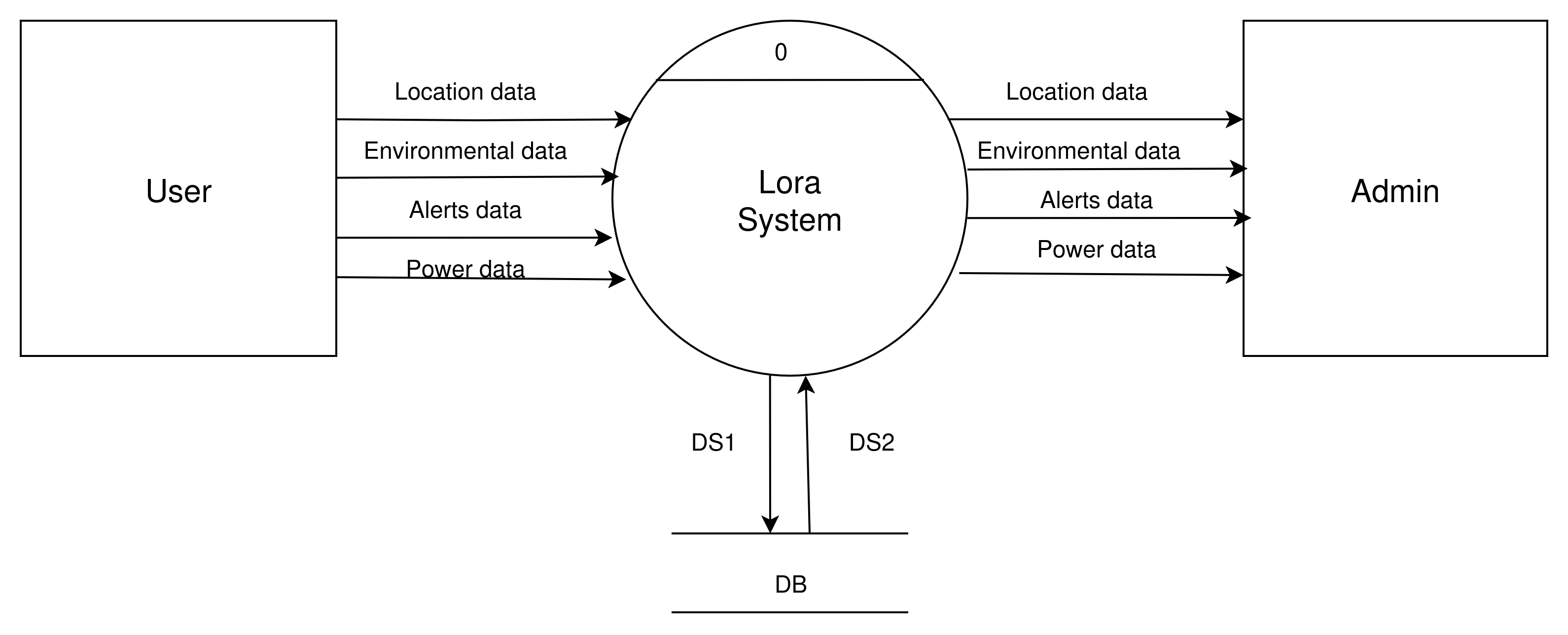
*Figure 5.4: Sequence Diagram for Sensor Data Transmission*

Displays event ordering for a typical data update cycle from sensor node to persistent storage and then to the web dashboard.

## **5.4 Functional Modeling**

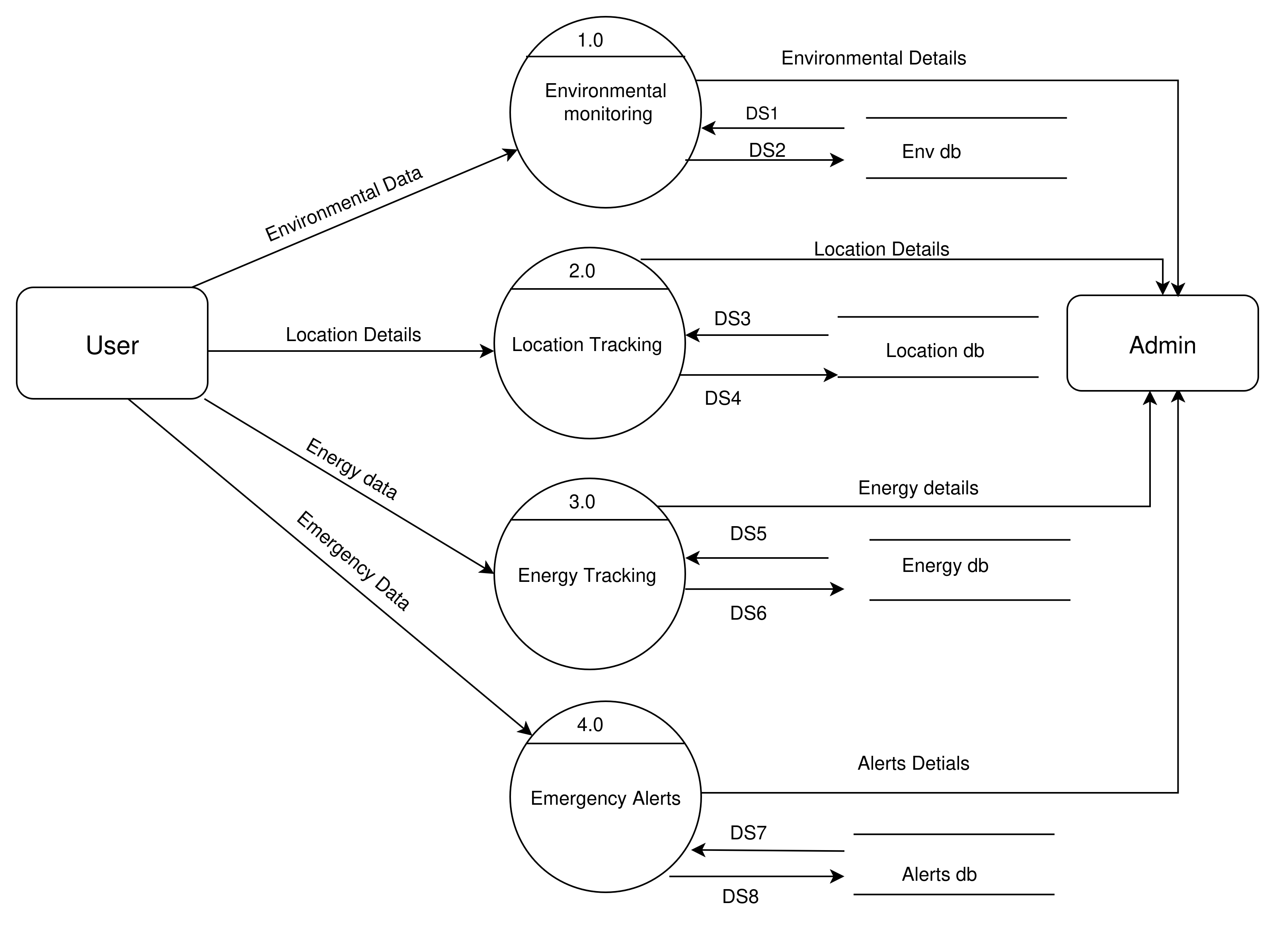
## Data Flow Diagrams (DFD)

**Level 0 (Context):**

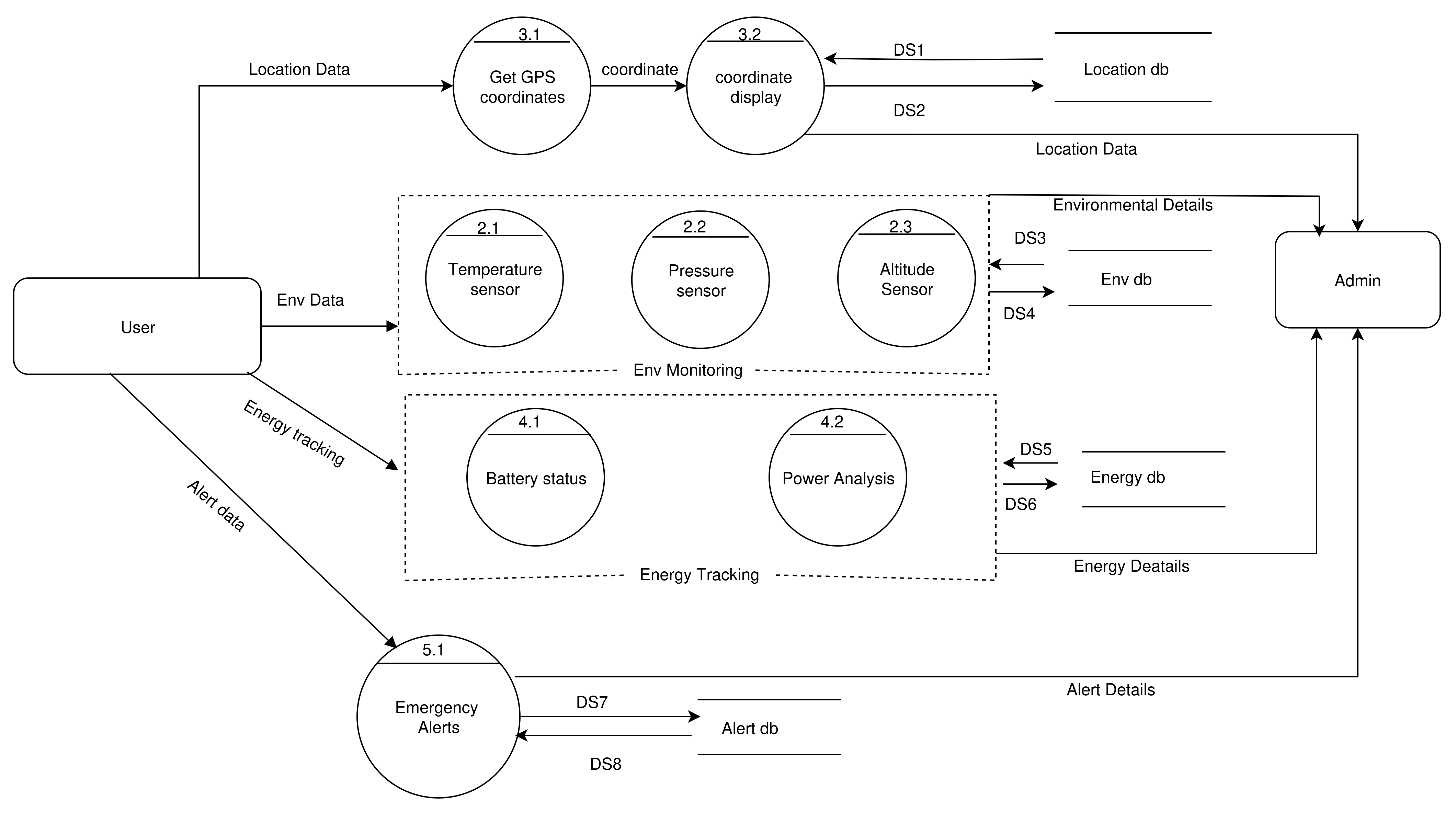
****

***Figure 5.5: Level 0 Data Flow Diagram***

**Data Flow Diagram Level 1**

***Figure 5.6: Level 1 Data Flow Diagram***

**Level 2 (Gateway Detail):**



***Figure 5.6: Level 2 Data Flow Diagram***

## Assumptions Made

* Each sensor node produces a unique node\_id.
* Time synchronization (timestamp) is based on the node’s GPS time.
* Emergency events are distinctly logged and have precedence in displays.
* Data files are stored locally on the SD card using simple tabular (CSV/JSON) formats for robustness and compatibility.
* The ESP32 gateway supports up to 8–10 concurrent Wi-Fi user sessions due to hardware limits.

## 5.5 Detailed Design

## Design Decisions

* **Data Structures:** Sensor data is encapsulated in JSON objects for communication and in rows (CSV/JSON) for persistent storage. Each entity (node, alert, user session) is represented by corresponding C++ structs or classes, improving clarity.
* **Modularity:** Functional decomposition into classes (Node, Gateway, SensorData, EmergencyAlert, UserSession, DataStorage) ensures separation of concerns, which simplifies testing and future upgrades.
* **Efficiency:** Minimal use of dynamic memory; use of rings/circular buffers for recent data; pre-indexed SD card access for rapid lookup.

## Logic Design (PDL for Each Module)

## 1. Node Data Acquisition Module (PDL)

* **Begin**
  + Read all sensor values (GPS, temperature, pressure, battery)
  + If emergency button pressed or battery < threshold, set alert flag
  + Package all data as JSON
  + Transmit JSON packet via LoRa
  + Sleep until next interval
* **End**

## 2. Gateway Data Reception Module (PDL)

* **Begin**
  + Continuously listen for LoRa packets
  + Upon receiving packet, parse JSON
  + Validate packet integrity
  + Log data to SD card (append to file)
  + Update shared memory/datastructure for dashboard
  + If alert\_flag detected, update alert status
* **End**

## 3. Web Dashboard Module (PDL)

* **Begin**
  + On user connect, authenticate session if needed
  + Serve main HTML dashboard and map tiles
  + On data/API request, return most recent and historical telemetry
  + On alert acknowledgment, update state and refresh UI
* **End**

**Bibliography**

[[1] W. Chen, L. Zhang, and S. Kumar, "IoT-based emergency communication systems for remote areas: A comprehensive review,"](https://www.techscience.com/cmc/v82n2/59507/html?utm_source=chatgpt.com) IEEE Internet of Things Journal, vol. 10, no. 8, pp. 2156-2170, Aug. 2023.

[2] M. Rodriguez, A. Thompson, and N. Patel, "LoRa technology for long-range IoT applications: Performance analysis and implementation strategies," Wireless Communications and Mobile Computing, vol. 2022, Article ID 8974523, 2022.

[3] J. Kim, H. Lee, and R. Anderson, "Offline web server implementation on ESP32 for remote monitoring applications," Journal of Embedded Systems and Applications, vol. 15, no. 3, pp. 45-58, Mar. 2023.

[4] P. Singh, D. Williams, and K. Brown, "Power management techniques for battery-operated LoRa sensor networks in harsh environments," Sensors and Actuators A: Physical, vol. 341, pp. 113567, Jun. 2022.

[5] T. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology," in Proc. 14th Int. Conf. ITS Telecommunications, Copenhagen, Denmark, Dec. 2015, pp. 55-59.

[6] R. Sanchez-Iborra and M. D. Cano, "State of the art in LP-WAN solutions for industrial IoT services," Sensors, vol. 16, no. 5, pp. 708, May 2016.

[7] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," IEEE Communications Magazine, vol. 55, no. 9, pp. 34-40, Sep. 2017.

[8] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the internet of things," Sensors, vol. 16, no. 9, pp. 1466, Sep. 2016.

[9] S.Okubo, S. Horihata, T. Takahashi, and K. Ohta, "LoRa-based tracking system for search and rescue operations in mountainous regions," in Proc. IEEE Int. Conf. Consumer Electronics, Las Vegas, NV, USA, Jan. 2019, pp. 1-4.

[10] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN specification v1.0.2," LoRa Alliance, Tech. Rep., Jul. 2016.

[11] B. Martinez, M. Monton, I. Vilajosana, and J. D. Prades, "The power of models: Modeling power consumption for IoT devices," IEEE Sensors Journal, vol. 15, no. 10, pp. 5777-5789, Oct. 2015.

[12] K. Mikhaylov, J. Petäjäjärvi, and T. Hänninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in Proc. European Wireless Conf., Oulu, Finland, May 2016, pp. 1-6.

[13] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," Sensors, vol. 18, no. 11, pp. 3995, Nov. 2018.

[14] D. Bankov, E. Khorov, and A. Lyakhov, "Mathematical model of LoRaWAN channel access with capture effect," in Proc. IEEE 28th Int. Symp. Personal, Indoor, and Mobile Radio Communications, Montreal, QC, Canada, Oct. 2017, pp. 1-5.

[15] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?," in Proc. 19th ACM Int. Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems, Malta, Nov. 2016, pp. 59-67.

[16] G. Ferrari, P. Medagliani, S. Di Piazza, and M. Martalo, "Wireless sensor networks: Performance analysis in indoor scenarios," EURASIP Journal on Wireless Communications and Networking, vol. 2007, Article ID 81864, 2007.

[17] Y. Mo, C. Goursaud, and J. M. Gorce, "Theoretical analysis of UNB-based IoT networks with path loss and random spectrum access," in Proc. IEEE 25th Annual Int. Symp. Personal, Indoor, and Mobile Radio Communications, Washington, DC, USA, Sep. 2014, pp. 1667-1672.

[18] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?," IEEE Wireless Communications Letters, vol. 6, no. 2, pp. 162-165, Apr. 2017.

[19] A. Lavric and V. Popa, "Internet of things and LoRa™ low-power wide-area networks: A survey," in Proc. Int. Symp. Signals, Circuits and Systems, Iasi, Romania, Jul. 2017, pp. 1-5.

[20] T. Voigt and M. Bor, "Mitigating inter-network interference in LoRa networks," in Proc. 2nd Int. Conf. Embedded Wireless Systems and Networks, Junction Publishing, 2017, pp. 323-328.

[21] J. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa: Experiences from a large-scale measurement study," ACM Transactions on Sensor Networks, vol. 15, no. 2, pp. 1-35, Feb. 2019.

[22] R. P. Centelles, F. Freitag, X. Vilajosana, R. Meseguer, and S. F. Ochoa, "A LoRa-based communication system for coordinated response in an earthquake aftermath," Proceedings of the Computer Science, vol. 151, pp. 427-434, 2019.

[23] S. Chaudhari, R. Kotak, and F. T. Z. Khatri, "Low power wide area networks: A survey of enabling technologies, applications and interoperability needs," IEEE Access, vol. 8, pp. 66066-66093, 2020.

[24] M. Bor and U. Roedig, "LoRa transmission parameter selection," in Proc. 13th Int. Conf. Distributed Computing in Sensor Systems, Ottawa, ON, Canada, Jun. 2017, pp. 27-34.

[25] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in NS-3," IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2186-2198, Dec. 2017.

[26] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair adaptive data rate allocation and power control in LoRaWAN," in Proc. IEEE 19th Int. Symp. World of Wireless, Mobile and Multimedia Networks, Chania, Greece, Jun. 2018, pp. 1-9.

[27] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa imperfect orthogonality: Analysis of link-level performance," IEEE Communications Letters, vol. 22, no. 4, pp. 796-799, Apr. 2018.

[28] J. P. Bardyn, T. Melly, O. Seller, and N. Sornin, "IoT: The era of LPWAN is starting now," in Proc. 42nd European Solid-State Circuits Conf., Lausanne, Switzerland, Sep. 2016, pp. 25-30.

[29] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 855-873, 2nd Quart. 2017.

[30] A. Eldefrawy, A. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of LoRaWAN," Computer Networks, vol. 148, pp. 328-339, Jan. 2019.