Minimizing Power consumption of GPS Module

This solution minimizes GPS power consumption using an IoT device (ESP and Neo-6M GPS module) combined with a backend service and Firebase RTDB. The backend service classifies zones as safe or unsafe based on coordinates from the IoT device, manages zone data, expands safe zones when conditions are met, and runs periodic cleanups.

Components:

1. **IoT Device (A)**:
    - Sends GPS coordinates to the backend service.
    - Executes sleep cycle according to the value received from backend service
2. **Backend Service (B)**:
    - Receives and processes coordinates from the IoT device.
    - Runs scripts for cleanup and analysis
3. **Firebase RTDB (C)**:
    - Stores classifications and attributes of zones.

Workflow:

1. IoT Device Operations

- **Input**: GPS coordinates sent to the backend service.
- **Output**: Executes sleep cycle accordingly

2. Backend Service Operations

- **Coordinate Processing**:
    - Takes coordinates from the IoT device.
    - Classifies them into Safe or Unsafe zones.

3. Zone Classification

- **Safe Zone (C)**:
    - If the zone has been visited 5 times or is part of an existing geofenced safe zone:
        - Add the zone to the Safe Zone list in Firebase RTDB.
        - Check if new coordinates fall completely under the existing safe zone or fall within for more than or equal to 70%:
            - If yes, expand the safe zone.
        - Device turns off the GPS with a 15-minute sleep cycle (F).

- **Unsafe Zone (L)**:
    - If the zone does not meet the safe criteria:
        - Update the zone data in the Temp Zone in Firebase RTDB.
        - Send notification to the user: "You are in an unsafe zone."
        - Device stays on and sends coordinates every 2 minutes (G).
        - Include hooks for frequency management (O).

4. Frequency Management (O)

- Increment frequency only if there is a 24-hour gap since the last visit (P).
- No increment if the zone is visited more than once in a day (Q).

5. Cleanup Process (R)

- Periodically removes zones not visited for over a month from the Safe Zone list in Firebase RTDB (S).

6. Firebase RTDB Schema (T)

- **Safe Zone Data Structure (U1)**:
    - last_visited
    - coordinates
- **Temp Zone Data Structure (U2)**:
    - coordinates
    - lastVisited
    - frequency

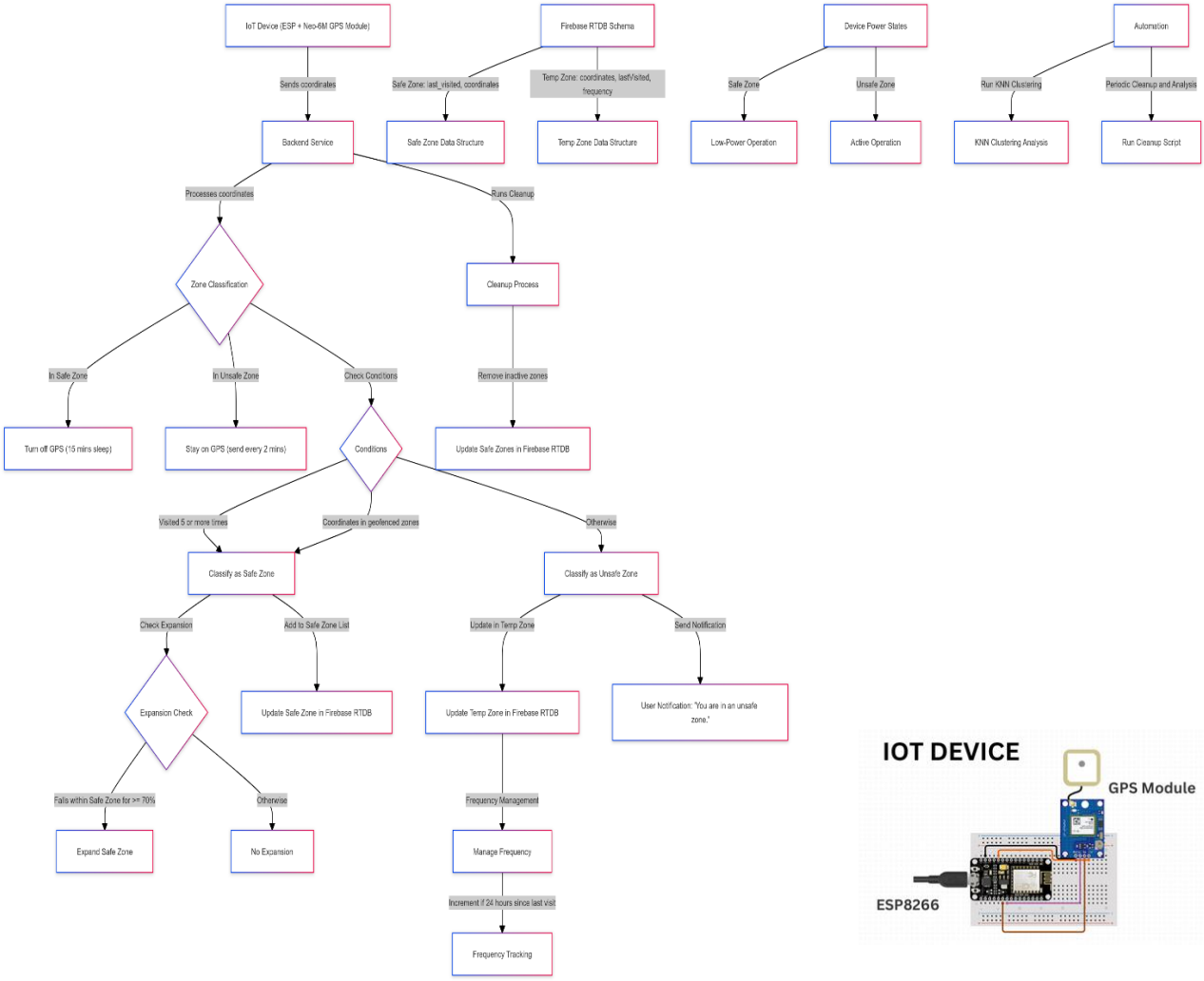
7. Device Power States

- **Safe Zone**: Device enters low-power operation mode (V).
- **Unsafe Zone**: Device remains in active operation mode (W).

8. Automation

- The backend service runs the cleanup script and analysis periodically (weekly or monthly) (X).

DIAGRAM


IoT Device (ESP + Neo-6M GPS Module)
Sends coordinates
Backend Service
Processes coordinates
Zone Classification
In Safe Zone
In Unsafe Zone
Check Conditions
Turn off GPS (15 mins sleep)
Stay on GPS (send every 2 mins)
Conditions
Visited 5 or more times
Coordinates in geofenced zones
Otherwise
Classify as Safe Zone
Classify as Unsafe Zone
Check Expansion
Add to Safe Zone List
Update in Temp Zone
Send Notification
Expansion Check
Update Safe Zone in Firebase RTDB
Update Temp Zone in Firebase RTDB
User Notification: 'You are in an unsafe zone.'
Falls within Safe Zone for >= 70%
Otherwise
Expand Safe Zone
No Expansion
Frequency Management
Manage Frequency
Increment if 24 hours since last visit
Frequency Tracking
Runs Cleanup
Cleanup Process
Remove inactive zones
Update Safe Zones in Firebase RTDB
Firebase RTDB Schema
Safe Zone: last_visited, coordinates
Temp Zone: coordinates, lastVisited, frequency
Safe Zone Data Structure
Temp Zone Data Structure
Device Power States
Safe Zone
Unsafe Zone
Low-Power Operation
Active Operation
Automation
Run KNN Clustering
Periodic Cleanup and Analysis
KNN Clustering Analysis
Run Cleanup Script

IOT DEVICE
GPS Module
ESP8266