

BATTERY HEALTH REMAINDER SYSTEM

TECH STACK, ARCHITECTURE & TOOLS

1. Project Overview

This project is a complete **end-to-end battery reminder system** that identifies locks whose battery levels have not been checked in over **1 month**, sends **FCM notifications** to their assigned users, and tracks **user clicks** to measure the campaign's effectiveness.

The process runs **weekly**, fully **locally**, with **no AWS required**.

2. Tech Stack Used

Core Technologies

Component	Technology Used	Purpose
Programming Language	Python 3.10	Main project logic
Database 1	SQLite (local file database)	Simulated PostgreSQL user-lock mapping
Database 2	Local JSON files	Simulated DynamoDB + analytics storage
Notifications	Firebase Cloud Messaging (FCM)	Send push notifications
Tracking	Local JSON events store	Track campaign clicks
Scheduling	Manual or Task Scheduler	Run weekly

4. System Architecture (Local Version)

Components

1. **locks.json**

- Simulates DynamoDB
- Contains lock IDs + last battery check timestamp

2. **SQLite Database (users.db)**

- Simulates external PostgreSQL

- Stores: lock_id → user_id → fcm_token

3. **main.py**

- Weekly job
- Reads stale locks
- Sends notifications
- Logs “sent” events

4. **analytics.py**

- Tracks click events
- Generates weekly CTR report

Findings & Recommendations

Findings

- The Battery Reminder System successfully tracks notification campaigns and user interactions (opens, clicks).
- The local setup works fully without AWS or cloud services.
- Analytics module accurately logs user behavior for each campaign with unique campaign IDs.
- The modular structure (main.py, analytics.py, firebase_service.py) makes the system easy to maintain and extend.
- testing with sample campaigns shows the system correctly sends reminders and records user engagement.

Recommendations

- Add a GUI dashboard to view real-time analytics (opens, clicks, reminder status).
- Enable scheduling so notifications can be sent automatically at fixed intervals.
- Add battery-percentage monitoring integration for real phone data.
- Store analytics in SQLite for better history tracking instead of text/log files.
- Add email/SMS support using a local SMTP server for multi-channel reminders.

GitHub link

GitHub: https://github.com/Anisha-1805/Atomberg_battery-reminder-/tree/main