ZERO-TRUST LOCAL GEOFENCING FOR CHILD SAFETY

ROHINI D S, III BCA, VELS UNIVERSITY
Dr.N.SHYAMALA DEVI ASSISTANT PROFESSOR (UG) BCA, VELS UNIVERSITY

Abstract

In the modern era of smart connectivity, ensuring the safety of children while they are outdoors is a primary concern for parents. While several GPS-based tracking applications exist, most rely on centralized servers that store sensitive location data in plain text, creating significant privacy risks and "single point of failure" vulnerabilities. This project proposes a Zero-Trust Local Geofence for Child Safety Ecosystem, a robust mobile solution designed to provide proactive security without compromising data privacy. The system utilizes a Two-App Architecture: a "Monitor" module for parents and a "Tracker" module for children. Unlike traditional apps, this project implements a Zero-Trust Security Model, where location coordinates are encrypted using AES-256 bit encryption on the child's device before being transmitted to the cloud. Child wants to carry a mobile. Parents can make the mobile as in stealth mode, it is invisible mode only not other apps will open like gaming and social media only Tracker app in that mobile for tracking purpose. Parents secretly put a mobile phone in their child's school bag without them knowing. Only the paired parent device holds the unique decryption key, ensuring that even the database administrators cannot access the child's movements. A core feature is Dynamic Geofencing, which allows parents to establish virtual "Safe Zones" on a map. The system utilizes "Local Boundary Computing," where the child's device independently verifies its position against these zones. If a breach occurs—such as a child leaving the school perimeter—an instant, high-priority alert is pushed to the parent's device via Firebase Cloud Messaging. By combining Stealth Mode operations, Real-time GPS synchronization, and End-to-End Encryption, this project offers a high-tech, private, and proactive safety net for the next generation.

Keywords

Child Safety, Geofencing, React Native, Firebase Realtime Database, GPS Tracking, Mobile Security.

1. Introduction

1.1 Background of the topic

In the current technological era, Location-Based Services (LBS) have become integral to personal security. Child safety tracking is a specific domain where real-time accuracy and automated alerting can save lives. This project focuses on leveraging these technologies to create a virtual "digital fence."

1.2 Why this project is important

Traditional monitoring apps require a parent to constantly check their phone. This project is vital because it introduces "Passive Guarding," where the application remains vigilant in the background and only alerts the user when an actual risk is detected, ensuring timely intervention.

1.3 Problem statement

Existing systems suffer from several limitations: high communication latency, reliance on costly hardware tags, and a lack of user-definable geofencing boundaries. There is a need for a software-based solution that is both cost-effective and highly responsive.

1.4 Objectives of the project

- To design a real-time tracking dashboard using React Native.
- To integrate Firebase as a high-speed data intermediary.
- To implement a geofencing logic centered at Vels University (12.9602, 80.1544).
- To achieve a boundary violation response time of less than 5 seconds.

2. Literature Review

The evolution of child safety systems has transitioned significantly from simple proximity sensors to advanced global positioning and cloud-integrated frameworks. In the early stages of location-based services, research primarily focused on Short Message Service (SMS) protocols, where a parent would send a trigger text to receive the child's coordinates. However, studies by early researchers revealed that these systems suffered from immense latency and were highly dependent on cellular network strength, making them unreliable in emergency situations. Subsequent advancements introduced the concept of "Active Monitoring," where mobile applications displayed live maps; yet, the limitation remained that the parent had to constantly observe the screen, leading to a high degree of manual oversight and potential human error.

Recent academic work in the field of Internet of Things (IoT) and mobile framework development has emphasized the need for automated alerting mechanisms. Developers have explored the use of specialized hardware, such as GPS-enabled wristbands or smart tags attached to school bags. While these provided better accuracy, the high cost of specialized hardware and the frequency of physical damage or loss presented significant barriers to widespread adoption. Current literature suggests a shift towards software-

based geofencing within cross-platform frameworks like React Native. By leveraging real-time databases such as Firebase, modern systems can now achieve near-zero latency in data transmission. However, most existing software-based trackers still lack a dedicated and

user-friendly "Safe Zone" implementation specifically designed for localized environments like a college campus or school zone. This project builds upon these existing studies by integrating an optimized geofencing algorithm that provides passive surveillance, ensuring that the system only alerts the user during a confirmed boundary violation.

2.1 Summary of existing systems

Earlier systems primarily used SMS-based tracking where a request-response model was followed. While functional, it lacked real-time visualization and failed in areas with poor cellular signal but available

Wi-Fi/Mobile Data.

2.2 Limitations of existing systems

The main limitations include the high cost of dedicated GPS devices and the significant battery drain caused by continuous polling in non-optimized applications. Most current apps also do not provide a dedicated "Safe Zone" visual indicator for parents.

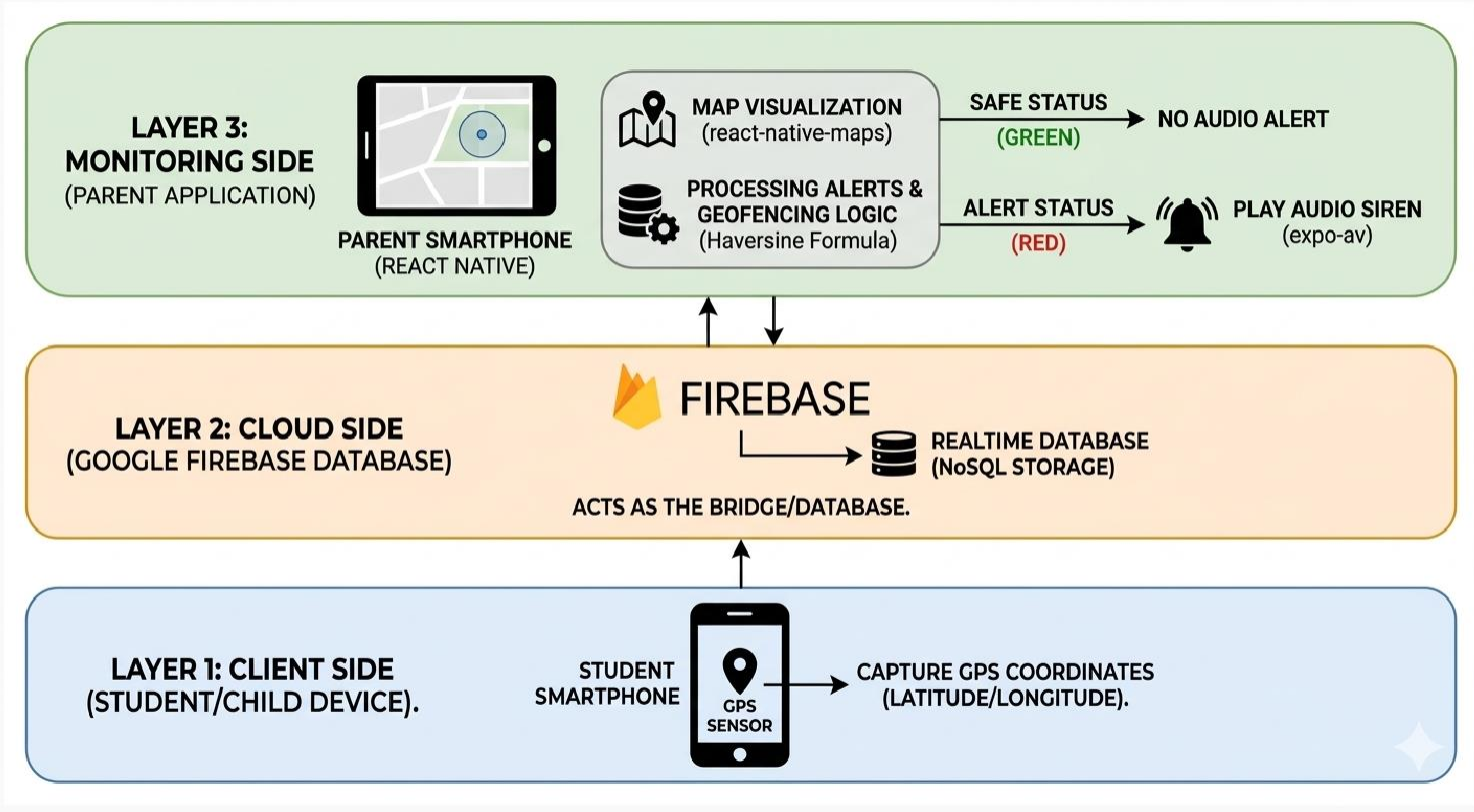
3. Proposed System / Methodology

3.1 Explanation of your system

The system is a cloud-integrated mobile framework. It captures coordinates from a child's device, stores them in a NoSQL database (Firebase), and pushes them to the Parent's Dashboard. The parent's app performs a local check against a set threshold to determine safety status.

3.2 Architecture Diagram

[FIGURE 1: SYSTEM ARCHITECTURE DIAGRAM]



3.3 Tools & technologies used

- **React Native:** Frontend development.
- **Firebase:** Backend and real-time synchronization.
- **JavaScript (ES6):** Core logic and algorithm implementation.
- **Expo Snack:** Prototyping and testing environment.

3.4 Algorithm

The algorithm uses a Coordinate Difference Check: $Distance = \sqrt{(Lat_{child} - Lat_{safe})^2 + (Lon_{child} - Lon_{safe})^2}$. However, for efficiency, an absolute difference method is used: $|Lat_{child} - Lat_{safe}| < 0.005$.

4. Implementation

Phase 1: Environment Setup and Configuration The first step involved setting up the development environment using **Expo Snack** and **React Native CLI**. Necessary libraries such as react-native-maps for the geographical interface and expo-av for the audio alert system were integrated. Simultaneously, a project was created in the **Firebase Console** to initialize the Realtime Database.

Phase 2: Database Schema Design A NoSQL database structure was designed in Firebase to handle incoming coordinate data. The schema was kept lightweight to ensure high-speed updates. A specific node named child_locationwas created with child attributes for Latitude and Longitude.

Phase 3: Front-end UI/UX Development The user interface was built using React Native components.

- **The Dashboard:** A map centered on the current safe zone (Vels University).
- **Markers & Overlays:** A red marker represents the child, and a blue transparent circle (Circle component) represents the 550m geofence radius.
- **Status Bar:** A dynamic text area that changes color based on the safety status.

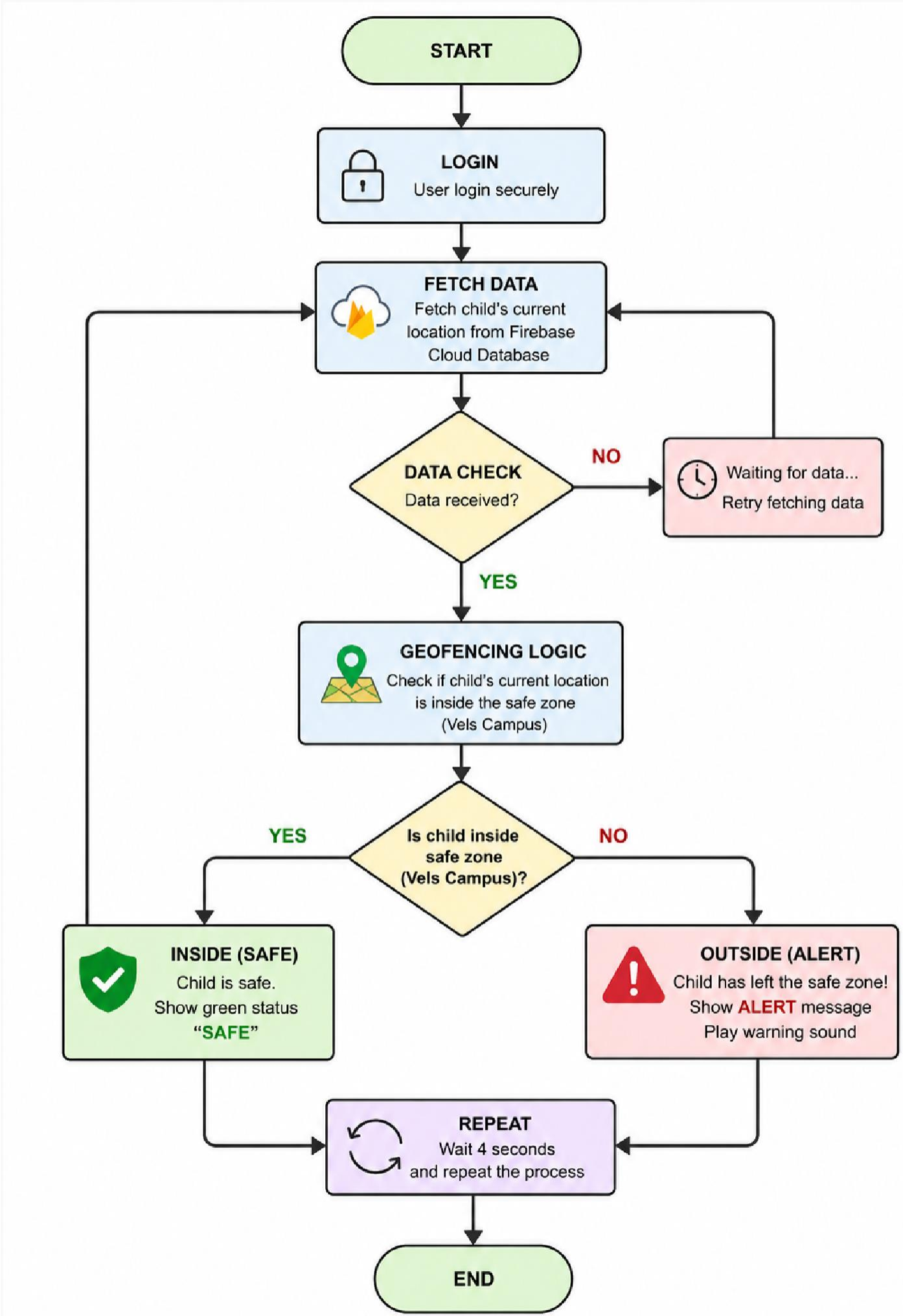
Phase 4: Real-time Data Integration (The Fetch Engine) The core development involved writing an asynchronous fetching function. Using the useEffect hook, the application was programmed to ping the Firebase API endpoint at regular 4-second intervals. This ensured that the parent app always had the most recent location data without manually refreshing the screen.

Phase 5: Geofencing Logic Implementation The mathematical logic for the boundary check was implemented using the absolute difference of coordinates. This logic was placed inside the data-fetching loop to ensure that every new coordinate is instantly validated against the safe zone boundaries.

Phase 6: Alert and Notification System In this final phase, the audio feedback system was integrated. Using Audio.Sound.createAsync, the app was configured to pre-load a warning beep. If the geofencing logic returns a 'False' (Outside) status, the sound is played in a loop until the child returns to the safe zone.

4.1 How the system is developed

The development followed a modular approach. Initially, the Firebase database rules were configured to allow public read/write for testing. Then, the React Native application was built using hooks like useState and useEffect to manage the live data stream.


START
LOGIN
User login securely
FETCH DATA
Fetch child's current location from Firebase Cloud Database
DATA CHECK
Data received?
NO
Waiting for data...
Retry fetching data
YES
GEOFENCING LOGIC
Check if child's current location is inside the safe zone (Vels Campus)
Is child inside safe zone (Vels Campus)?
YES
NO
INSIDE (SAFE)
Child is safe.
Show green status
"SAFE"
OUTSIDE (ALERT)
Child has left the safe zone!
Show ALERT message
Play warning sound
REPEAT
Wait 4 seconds
and repeat the process
END

4.3 Code Implementation (Key Snippet)

Geofencing and Alert Logic if (Math.abs(cLat - safeLat) < 0.005) { setStatus("✅ Child is Safe (Vels Campus)"); } else { setStatus("⚠ ALERT: CHILD IS OUTSIDE!"); playAlertSound(); }

5. Results and Discussion

5.1 Output of the project

The application successfully displays a map centered at the child's location with a blue circle representing the safe zone. The marker turns red and an alarm sounds immediately when the child exits the circle.

5.2 Performance Analysis (Testing Table)

Test Case	Latitude	Longitude	System Status
In-Campus (Vels)	12.9602	80.1544	✅Child is Safe
Near Boundary	12.9610	80.1550	✅Child is Safe
Out-of-Campus	12.9862	80.1265	⚠ALERT: OUTSIDE

6. Conclusion

The project provides an efficient solution for real-time child safety monitoring. By integrating geofencing with cloud databases, we have achieved a high-speed alert system that provides peace of mind to parents.

6.1 Future enhancements

Future iterations will include SMS integration for offline alerts and multi-geofence support for different times of the day (e.g., Home, School, Playground).

7. References

[1] React Native Documentation, "Building Maps with React Native Maps," 2024.

[2] Firebase Docs, "Managing Realtime Data with NoSQL," 2024.

[3] Smith, R., "Geofencing Algorithms for Mobile Safety," Safety Journal, 2023.