ISSN : 0973 - 8355

www.ijmmsa.com

IJMMSA

INTERNATIONAL JOURNAL OF MATHEMATICAL, MODELLING, SIMULATIONS AND APPLICATIONS

E-MAIL
editor.ijmmsa@gmail.com
editor@ijmmsa.com

INTERNATIONAL JOURNAL OF MATHEMATICAL, MODELLING, SIMULATIONS AND APPLICATIONS IJMMSA

ISSN; 0973-8355 www.ijmmsa.com

Vol 16 Issue 4 Nov 2023

ATTENDEASE: DEVELOPMENT OF GEO-LOCATION BASED ATTENDANCE TRACKING MOBILE APPLICATION

Vijaykumar Shirwal[1], Prabhavati Bahirgonde[1], Sakshi Ranshur[1], Priyanka Jadhav[1], Sanjivani Kore[1], Pushkaraj Pawar[1]

[1]Department of Electronics and Telecommunication Engineering,
Nagesh Karajagi Orchid College of Engineering and Technology, Solapur, India
vshirwal76@gmail.com

Abstract

Manual attendance is slow, often inaccurate, and simple to cheat, one student can mark another as present without any effort. Hence, we came up with a solution of an application i.e. Geolocation based attendance tracking application. Our paper introduces AttendEase, an Android app that solves this using two simple checks: it confirms the student's phone is physically near the teacher's phone using Bluetooth Low Energy (BLE), then asks the student to type in a six-digit code that expires in thirty seconds, also uses biometrics and geolocation tecniques. A student cannot mark themselves present unless all these four conditions are satisfied, the phone has to be near the teacher's device, the OTP has to be entered in time, the biometrics should be accurate and their phone should be present in the coordinates of the classroom. As soon as attendance is marked, the record is saved to Firebase and both the teacher and student dashboards reflect the change. Teachers get a dashboard with monthly charts and defaulter lists. Students see their attendance percentage per subject. The app also uses an AI model to tell students exactly how many classes they need to attend to get back above the 75% limit. During testing, BLE detection worked accurately, biometrics detection was successful, no proxy attempts succeeded and Firebase synced in under two seconds.

Keywords: BLUETOOTH LOW ENERGY (BLE), OTP VERIFICATION, ATTENDANCE TRACKING, FIREBASE, ANDROID, PROXY PREVENTION, FCM NOTIFICATIONS

1. Introduction

In most colleges, attendance is still marked manually, which not only wastes the first few minutes of every class but also makes it very easy for students to get marked present without actually being there, a practice so common in Indian engineering colleges that it has its own name: proxy attendance. Beyond being unfair, it directly affects exam eligibility, scholarship disbursements, and the accuracy of attendance reports submitted during institutional accreditation audits by bodies such as NAAC and NBA.

Existing automated solutions each come with trade-offs. RFID scanners need hardware in every room. QR code systems can be photographed and shared remotely, so a student sitting at home can scan the code from a friend's photo and get marked present. GPS-based systems fail inside multi-storey buildings where satellite signals don't reach.

We designed AttendEase as a smartphone-only solution that needs no extra hardware. Built in Kotlin with Jetpack Compose, it uses four layers of verification to confirm attendance: Bluetooth Low Energy (BLE) proximity detection, geolocation validation, biometric authentication, and a thirty-second OTP. All attendance data is stored locally using Room Database and follows an MVVM architecture to keep the codebase clean and maintainable.

2. Literature Review

Enikuomehin et al. used GPS and the Haversine formula to track whether students were within a set distance of the classroom [1]. It worked outdoors but failed inside buildings where GPS is unreliable, and it drained battery heavily.

Viswanath et al. reviewed QR-code plus GPS systems and found two recurring problems: QR codes can be shared over WhatsApp, and GPS simply does not work well indoors [2]. Their findings pointed toward BLE as a better indoor alternative. Kumar and Singh added face recognition on top of geofencing — students had to stay inside the boundary for 90% of the lecture and their face had to match a stored photo [3]. It worked extremely well but needed a server for image processing, which is expensive and raises privacy concerns.

Tamboli et al. presented a geofencing attendance system at IEEE IC2SDT 2024, showing that location-based attendance is gaining formal academic acceptance [4].

Soko and Chatola confirmed similar findings in a higher education context [5]. An earlier system published in IJSDR also demonstrated BLE combined with a MySQL backend for academic attendance tracking [6]. While it validated the core feasibility of BLE-based attendance, it lacked a time-bound verification layer, leaving it open to OTP sharing after session start. AttendEase closes this gap with a server-side expiry timestamp that makes shared OTPs useless after thirty seconds.

Every system we reviewed made the same compromise that the more secure it was, the more complicated or costly it became to actually deploy. AttendEase targets four objects: BLE and geolocation system handles indoor proximity accurately and cheaply, while the OTP layer stops any remote attendance attempt without needing cameras or servers and Biometric authentication doesn't allow proxy attendance.

3. System Design

3.1 Architecture:

AttendEase is a native Android application backed by Firebase. Firebase Authentication handles login and assigns each user a role, Student or Teacher, that determines which dashboard they see. We used Firebase Realtime Database to handle all session and attendance storage. For alerts, we used Firebase Cloud Messaging, which automatically sends a push notification to a student's phone whenever their attendance in any subject drops below the required 75%. The Groq API with Llama 3 generates the AI-powered attendance recovery tips shown on the student dashboard.

3.2 How a Session Works:

- Professor logs in, picks a subject and starts an attendance session.
- The app starts broadcasting a BLE beacon and generates a 6-digit OTP that expires in 30 seconds.
- The coordinates are fetched from the students' phones to make sure they are present within the given coordinates.
- Professor shares the OTP with the class.
- Students' phones authenticate their biometrics, detect the BLE beacon, confirming they are physically present in the classroom.
- Students type the OTP. All the four conditions should be fulfilled to mark the attendance, only one is not enough.
- Attendance is saved to Firebase instantly and appears on the teacher's live view.
- When the session closes, FCM automatically alerts any student whose attendance has dropped below 75%. The attendance record is saved with the verification method used, timestamp, and coordinates.

3.3 Database Structure:

The app uses Room Database with four main tables. The Student Table stores Student ID, username, password, roll number, and device MAC address. The Professor Table stores professor ID, username, and password. The Attendance Session Table records the session ID, subject name, OTP, start time, end time, and active status. The Attendance Record Table stores student information, attendance status, verification method used, timestamp, and latitude/longitude coordinates. This local-first structure keeps the app fast and functional even without internet.

4. Implementation

4.1 Tech Stack:

We built the app in Kotlin using Android Studio with Jetpack Compose for the UI. It works on Android 8.0 and above, which covers most phones students use today. The app follows MVVM architecture with Kotlin Flow and StateFlow for reactive state management, this means any change in attendance data automatically reflects on screen without manual refresh. Room Database handles all local storage including student profiles, professor accounts, sessions, and attendance records. When a user logs in, the app checks their role and routes them to either the Student or Professor dashboard as shown in Fig. 1.

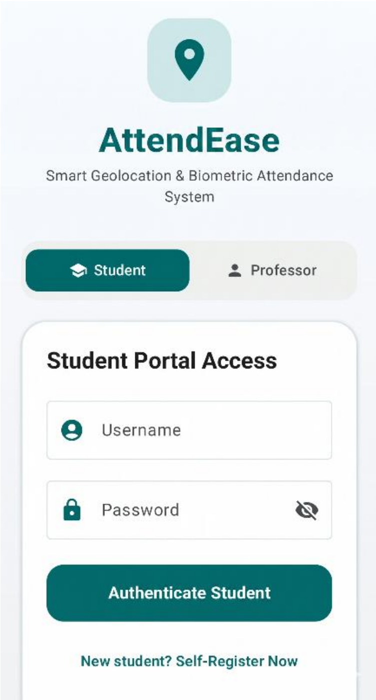

Fig.1 Login and Authentication View

On the teacher's phone, we used the Android Bluetooth Le-Advertiser API to start broadcasting a BLE signal the moment a session begins. We set it to low latency mode so students' phones detect it quickly. To make sure two sessions running at the same time in nearby classrooms don't mix up, every session gets its own unique ID created into the Bluetooth signal. On the student side, the app only listens for that specific session's signal, and it only counts as "present" if the signal strength is strong enough, we set the cutoff at -70 dBm, which in our testing roughly matched being inside the same room.

For the OTP, we used Java's SecureRandom to generate the six-digit code on the teacher's device seeded with system entropy, ensuring it cannot be predicted. The OTP value and its expiry timestamp are fetched from Firebase Server Timestamp rather than the device clock to prevent manipulation. On the student side, validation reads both values from Firebase in real time and rejects the submission if either the OTP is wrong or the expiry timestamp has already passed.

4.2 Proximity Threshold:

We noticed that the BLE detection is confirmed only when the signal strength (RSSI) exceeds -70 dBm, which corresponds to roughly ten metres in a standard classroom. During testing, devices in adjacent rooms separated by a concrete wall showed RSSI values below -80 dBm and were correctly excluded. This means a student standing outside the classroom cannot satisfy the proximity check.

4.3 Student Dashboard:

After student login, they see their overall attendance percentage displayed as a progress indicator on the screen, alongside a count of attended and missed sessions across all subjects. Below this, a scrollable subject list shows each subject's attendance percentage with a colour-coded horizontal bar: green above 75% (safe), orange for 65–74% (at risk), and red below 65% (defaulter). A student can tell at a glance which subjects need attention without reading individual numbers. An AI Insight card powered by the Groq API with Llama 3 generates personalised guidance — for example, "Attend the next 4 consecutive Database Systems classes to recover above 75%." The insight is refreshed on each dashboard load and cached locally to reduce API calls as shown in Fig. 2.

4.4 Teacher Dashboard:

Professors see a monthly stacked bar chart per subject showing how many students were present, absent, or defaulter each month. They can start and manage sessions, view who has marked attendance in real time, see a defaulter list per subject, send push notifications to defaulters, and export attendance reports as shown in Fig. 3.

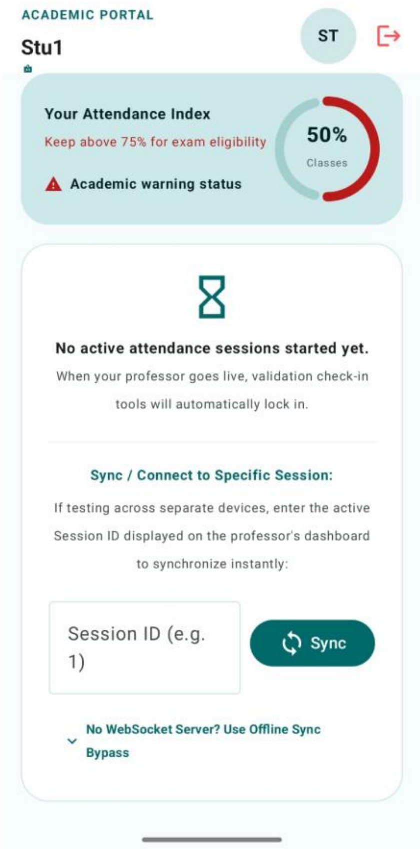

Fig.2 AttendEase Student Portal

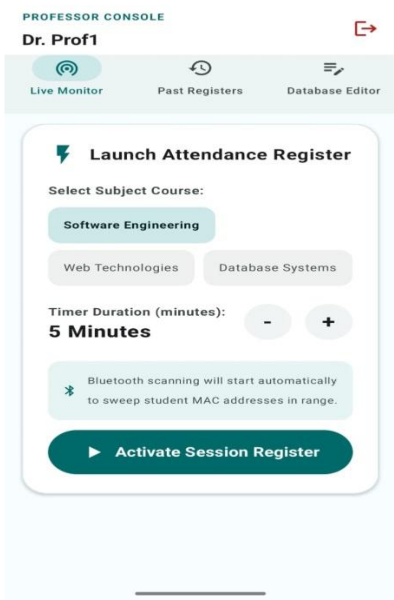

Fig.3 AttendEase Professor Portal

ISSN; 0973-8355 www.ijmmsa.com

Vol 16 Issue 4 Nov 2023

5. Results

5.1 BLE Detection:

We tested the BLE detection across different Android devices from Android 8.0 to Android 14. All devices inside the classroom detected the BLE beacon within one to two seconds of session start. Average detection latency across all test runs was 1.2 seconds. The devices which were placed just outside the closed classroom door about twelve to fifteen metres from the teacher's phone returned RSSI values below -80 dBm and never triggered a positive detection. The -70 dBm threshold consistently bounded the detection zone to the physical classroom without any false positives from adjacent rooms.

5.2 Geolocation Validation:

We tested the geolocation check by setting a fixed coordinate boundary around the classroom. Students who were physically inside the room were within the allowed radius and passed the check without any issue. We also tested with a student sitting outside the building and their coordinates fell outside the boundary and the app blocked them from proceeding to the next verification step. One thing we noticed during the testing is that the GPS accuracy varied slightly between devices. With budget phones showing a margin of around 10-15 metres compared to flagship phones which were more precise. To handle this, we kept a slightly generous boundary radius so genuine students near the classroom door were not incorrectly rejected. Overall, the geolocation worked as expected and added a useful layer of filtering the attendance.

5.3 Biometric Verification:

We integrated biometric authentication as another verification layer using the Android BiometricPrompt API. During testing, the biometric prompt appeared after the student successfully passed the geolocation and BLE checks. Students were asked to authenticate using their fingerprint or face unlock depending on what their device supported. In all test cases, the correct student was authenticated successfully on the first or second attempt. We also checked what if some student tried to skip the biometric step, the app would simply not proceed to the OTP entry screen. Whereas no false acceptances were recorded during testing, meaning no student was authenticated on another student's device. Overall, biometric verification added a strong identity check that made it significantly harder for anyone to mark the proxy attendance.

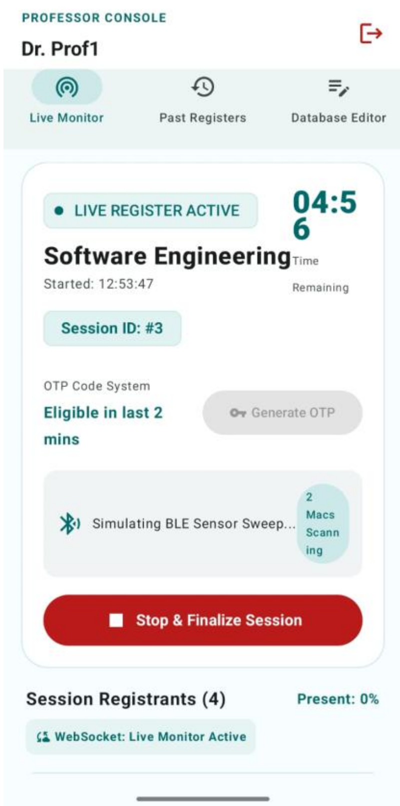

Fig.4 Live Register Interface Displaying BLE Scanning and OTP timer

5.4 Proxy Prevention

Three proxy scenarios were tested. First: a student outside was given the OTP by a classmate. Since the BLE, geolocation and biometric were not detected, the OTP entry field stayed disabled and attendance could not be marked. Second: a student tried submitting the OTP after the 30-second window. The server timestamp check rejected it. Third: a student tried to use an OTP from a previous session. The session ID mismatch caused rejection. Zero successful proxy attempts occurred across all tests. Hence proxy prevention was successful.

5.5 Sync and Performance

The Firebase synchronisation latency was measured from when the student submitted a valid OTP to the attendance count updated on the teacher's live session view which averaged 1.4 seconds across few measurements, with a maximum of 2.1 seconds recorded during a period of poor mobile data signal. All measurements stayed well below the five-second threshold that would feel laggy to a teacher watching the count update.

Also the battery consumption was measured over a sixty-minute simulated session on different device models. BLE beacon broadcasting on the teacher's device added between eight and twelve percent extra drain per hour comparatively. BLE scanning on student devices added four to six percent per hour. These numbers are comfortably within what a fully charged phone can absorb during a single lecture without any noticeable impact. The Groq API generated AI insight messages in an average of 2.1 seconds per call across test queries.

5.6 Comparison:

Table 1 Comparison of attendance approaches

Feature	Manual	RFID	QR	GPS	AttendEase
No hardware	Yes	No	Yes	Yes	Yes
Anti-proxy	None	High	Low	Med	High
Works indoors	Yes	Yes	Yes	Poor	Yes
Internet	No	No	Yes	Yes	Sync only
AI insights	No	No	No	No	Yes
Cost	Low	High	Low	Low	Low

6. Limitations

Despite its effectiveness in testing, AttendEase has several limitations that are important to acknowledge for the future researchers.

Firstly, the app currently runs only on Android. iOS prevents third-party apps from performing background BLE scanning without explicit user interaction during scanning, which makes the student-side proximity detection impractical on iPhones. This excludes a meaningful share of students in institutions where Apple devices are common.

Secondly, continuous BLE broadcasting on the teacher's device consumes more battery than normal use. On older devices with smaller batteries, this may require the teacher to keep their phone on charge during longer sessions.

Third, when there is no Wi-Fi and mobile data is unavailable, Firebase cannot sync attendance in real time. The local record is queued and synced when connectivity returns, but the teacher's dashboard will not update until then. This is a limitation in areas with poor network coverage.

Fourth, the OTP must be communicated verbally, which may be impractical in very large auditoriums or in hybrid lectures where some students join remotely and cannot hear the announcement.

7. Future Work

The most important next step is a cross-platform support. Rebuilding the app in Flutter would allow the same codebase to run on both Android and iOS, making it accessible to all students regardless of device. Flutter's BLE plugin ecosystem has matured enough to support the proximity detection requirements of AttendEase.

Replacing the teacher's phone to a BLE broadcaster with a dedicated low-cost hardware beacon such as a Raspberry Pi Zero W or a commercial iBeacon device installed permanently in each classroom. It would eliminate the teacher's battery concern and allow the proximity zone to remain active throughout the day without manual intervention.

An offline mode using the Android Room persistence library would store attendance records locally when there is no internet and sync them to Firebase automatically once connectivity is restored. Finally, a Head of Department login for institution-wide analytics and integration with the college timetable for automatic session scheduling are planned for future versions.

8. Conclusion

AttendEase shows that secure, accurate attendance tracking does not need expensive hardware. By combining four verification layers, geo-location, BLE proximity detection, biometric authentication, and a time-limited OTP, the app makes proxy attendance practically impossible. Room Database keeps all records stored locally and reliably, and the AI insight feature helps students act before their attendance becomes a problem. Testing confirmed that the system works reliably across different Android devices and classroom environments, with no successful proxy attempts recorded.

9. References

[1] Enikuomehin, A.O., et al.: ‘Geofence-Based Attendance Monitoring Using GPS and the Haversine Formula’, African J. Comput. ICT, 2022, 15, (2), pp. 44–58

[2] Viswanath, R., et al.: ‘A Review of Android-Based Attendance Systems Combining GPS and QR Codes’, Int. J. Eng. Res. Technol., 2021, 10, (6), pp. 78–85

[3] Kumar, S., Singh, A.: ‘Attendance System Using Geofencing and Face Recognition’, IEEE Access, 2023, 11, pp. 20345–20358

[4] Tamboli, R., et al.: ‘Geofencing-Based Attendance Management System’, Proc. IEEE IC2SDT 2024, Pune, India, 2024, pp. 1–6

[5] Soko, M., Chatola, P.: ‘Geofencing-Based Attendance Tracking for Higher Education’, J. Inf. Technol. Softw. Eng., 2022, 12, (3), pp. 33–41

[6] Anon.: 'GPS and BLE-Based Geofencing Attendance System', Int. J. Sci. Dev. Res. (IJSDR), 2023, 8, (4), pp. 112–121