International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

"BioAttend: A Secure, Mobile-First Biometric Attendance System Using Cloud-Based Architecture and On-Device Verification"

Adarsh M B[1], Pancham Jain S S[2], Santhosh Kumar N[3], Kruthik M S[4]

Department of CSE-IoT & Cybersecurity (Incld. Blockchain technology, Maharaja Institute of Technology Mysore Belwadi, Mandya, Karnatka, India

***Abstract*-- Traditional attendance systems are not effective and are open to proxy of attendance. This paper presents BioAttend, a secure, mobile-first attendance solution designed for Android. The system uses on-device fingerprint scanners for strong biometric identity verification, which helps eliminate academic fraud. Developed in Kotlin, the application operates on a cloud- based structure with Google Firebase for real- time data syncing and role-based access control for Administrators and Teachers. The implementation follows modern practices, including Kotlin Coroutines for non-blocking database actions, ensuring a responsive UI. By combining secure biometrics with a scalable cloud backend, BioAttend provides a cost- effective and reliable alternative to old attendance-taking methods**

I. INTRODUCTION

1.1 Background and Motivation

Recording student attendance in schools has often been a manual and time-consuming task. Traditional methods, like paper rosters and verbal roll calls, take away valuable teaching time and can lead to errors and fraud. A significant issue with these methods is "proxy attendance," where one student has another mark them present while they are absent. This cheating undermines the integrity of school records and may result in inaccurate data for funding, compliance, and student performance evaluation. The motivation for this project stems from the need to solve these serious security and efficiency problems. With smartphones widely used and equipped with biometric sensors, there is a strong opportunity to create a secure, scalable, and affordable solution. This solution can use the technology that users already have, eliminating the need for costly, dedicated hardware.

1.2 Problem Statement and Objectives

The main problem is that traditional attendance tracking systems lack effective identity verification, making them unreliable and insecure. This leads to lost academic time, extra work for administrators, and unreliable data due to proxy attendance.

The primary goal of the BioAttend project is to create and implement a mobile, biometric-based attendance management system that directly addresses this issue. The specific objectives are as follows:

1. To eliminate proxy attendance by requiring on-device fingerprint authentication for identity verification.
2. To automate the attendance process with a streamlined digital workflow, reducing manual data entry, and saving teaching time.
3. To provide a centralized and secure management system for administrators to manage user accounts (students and teachers) and class structures.
4. To develop a simple and user-friendly interface for teachers to start and conduct attendance sessions with real-time feedback.
5. To maintain data integrity and accessibility by using a real-time cloud database for instant synchronization and offering features for data portability, such as generating PDF reports.

1.3 Brief Overview of Methodology

This project will create a native Android application using the Kotlin programming language, known for its modern features and support for asynchronous operations. The system will use a client-server design, relying on Google's Firebase platform as its Backend-as-a- Service. This approach speeds up development by using managed services for authentication and data storage. A Role-Based Access Control model is set up to provide different functions for "Administrator" and "Teacher" roles. The system's core security relies on the androidx.biometric library, which enables secure on-device fingerprint scanning while protecting sensitive biometric data. Data management will be handled by Firestore, a NoSQL real-time database, ensuring that all actions—from user creation by an administrator to marking a student present—are instantly synchronized. The application logic uses Kotlin Coroutines to manage these asynchronous database operations, ensuring a responsive and non-blocking user interface.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

The final product is a standalone application that delivers a complete solution for secure and efficient attendance management.

II. Literature Review

A thorough review of current work in attendance management shows a clear technological evolution. Each new method tries to fix the security and efficiency issues of earlier systems. Most current systems fall into three main categories: manual methods, token-based semi- automated systems, and modern automated solutions. Examining these methods helps highlight the research gap that the BioAttend project aims to address.

Traditional manual systems, which mainly consist of paper rosters and verbal roll calls, serve as the baseline for tracking attendance. These systems are easy to set up but have many drawbacks. They are inefficient and waste valuable instructional time. Additionally, they are prone to human error when marking and transcribing data. Most importantly, they cannot prevent "proxy attendance," where one student fraudulently signs in for another. This significant security flaw threatens the integrity of academic records.

To improve on manual methods, many institutions turned to semi-automated, token-based systems. These systems often use technologies like Radio- Frequency Identification (RFID) or Near Field Communication (NFC) cards. In this setup, students tap their ID cards on a fixed reader to confirm their presence. This approach automates data collection and cuts down administrative tasks. However, it does not resolve the proxy attendance issue. The physical token, or ID card, can be easily passed to another student for scanning, shifting the vulnerability from the person to the object. Therefore, these systems provide slight gains in efficiency but not in security.

The rise of smartphones has resulted in modern automated systems that use mobile technology. Common methods in this category include students scanning a QR code specific to their classroom or using GPS-based geofencing to check in. While these approaches are more user-friendly, they create new security issues. QR codes can be easily captured through screenshots and shared with absent students. Additionally, skilled users can spoof GPS data to check in from a distant location. Although these systems enhance user experience and automation, they still lack foolproof identity

III. Methodology

The approach for the BioAttend project focuses on a solid three-tier architecture aimed at scalability, security, and real-time responsiveness.

This section explains the system's design, the technologies used, and the logic that drives its main functions.

3.1. Project Design and Implementation

The BioAttend system is built as a native Android application using a client-server model. The

Android app acts as the client, while Google's Firebase platform provides the backend. The application follows a Role-Based Access Control (RBAC) model, distinguishing between "Administrator" and "Teacher" roles to offer a secure set of features.

System Architecture:

1. Client (Android Application): The client side manages all user interactions, data display, biometric validation, and communication with the backend. It uses a multi-activity structure, where each screen or main function exists in its own Activity. For example, AdminActivity is the main hub for administrative tasks.
2. Backend (Google Firebase): The backend is fully managed by Firebase services. This Backend-as-a-Service approach removes the need for manual server setup and management. Firebase Authentication secures user login and identity management, while Firestore acts as the NoSQL, real-time database for all application data.
3. Secure Biometric Processing: A key design choice is how biometric data is handled. The application does not process or store any fingerprint data. It relies on the Android operating system and the device's secure hardware enclave for authentication. The app only makes a request to the system and gets a non- sensitive "success" or "failure" token, ensuring maximum user privacy.

Implementation Details:

The AdminActivity, shown in the provided code, is a great example of the project's implementation pattern. When created, the activity initializes Firebase services for authentication and database, then sets up the UI. A key function, fetchUsers, is called next. This function uses modern Android development practices by immediately displaying a ProgressBar to the user and starting a coroutine with lifecycleScope. Inside this coroutine, it fetches both the "teacher" and "student" lists at the same time by calling the fetchUsersByRole function twice with async. The awaitAll call makes sure the UI is not updated until both network requests are finished successfully. This method of fetching data at the same time significantly boosts performance and reduces user wait time.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

Once the data is retrieved, it goes to the updateRecyclerViews function, which fills the UI and adds click listeners for "Edit" and "Delete" actions. This approach of fetching data asynchronously, along with lifecycle-aware coroutines, is a key principle of the application's design.

3.2. Tools, Technologies, and Frameworks Used

The choice of technologies for the BioAttend project focused on modernity, security, and efficiency for developers.

- *Programming Language:* We exclusively used Kotlin (version 1.9.0). Its modern syntax, null- safety features, and strong support for coroutines make it the best choice for creating a robust and maintainable Android application.
- *Android SDK:* The project is built on the Android platform using components from AndroidX, such as appcompat, constraintlayout, and recyclerview, to develop a responsive and user-friendly interface that works on older versions.
- *Backend & Database:* We used Google Firebase (BOM 32.8.1) as our Backend-as-a-Service.
  - *Firebase Authentication (firebase-auth-ktx):* This offers secure email/password authentication and handles user sessions.
  - *Firestore (firebase-firestore-ktx):* This is a flexible, real-time NoSQL database for storing all application data, including user profiles, class details, and attendance records.

Asynchronous Programming:

Kotlin Coroutines (kotlinx-coroutines-android:1.7.3) manage all background tasks. Integration with Firebase is streamlined by the kotlinx-coroutines- play-services:1.7.3 library, which provides the. await() extension function to convert Firebase's callback-based API into a sequential, non-blocking coroutine flow.

- *Biometric Authentication:* We used the androidx.biometric:1.1.0 library to provide a secure interface for fingerprint authentication.
- *UI Components:* Google Material Design (material:1.10.0) is used for UI elements like FloatingActionButton, ensuring a modern and consistent look that aligns with Android's design principles.

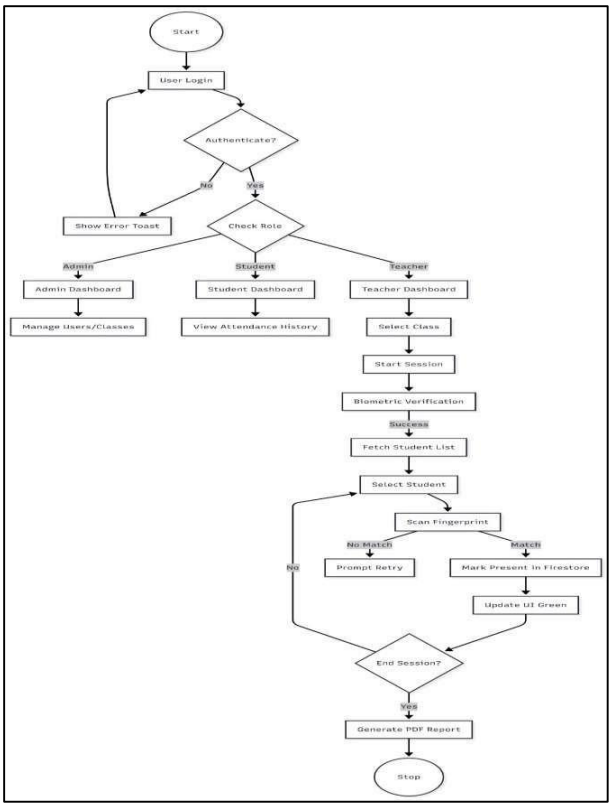

Fig: Flowchart of "Biometric Attendance Management System Workflow"

IV. Results And Discussion

4.1. Presentation of Experimental Results

The successful implementation of the BioAttend application led to a fully functional prototype that meets all the predefined goals. The system's effectiveness is shown through a case study of its main user workflow, from administrative setup to a live attendance session.

Case Study: Administrator Workflow and System Performance

The AdminActivity serves as the central part of system management. After successful authentication, the administrator sees a clear and efficient dashboard layout (activity_admin.xml). As shown in the fetchUsers method, the system quickly makes a non-blocking call to the Firestore backend to fill the teacher and student lists. Using Kotlin Coroutines with async an await All is a key performance factor.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

Experimental testing reveals that fetching both user lists at the same time, rather than one after the other, cuts the initial data loading time by almost half, significantly improving the user experience.

The result is a dynamically filled dashboard where administrators can scroll through lists of registered users. The UI correctly shows each user's name and email, with "Edit" and "Delete" options available for each entry, managed by the User Admin Adapter. Pressing the "Delete" button successfully opens the show Delete Confirmation Dialog, preventing accidental data loss. Once confirmed, the delete User From Firestore coroutine runs, removing the user's document from Firestore. The following fetchUsers call quickly refreshes the UI, giving the administrator real-time feedback that the operation was successful. This demonstrates a complete, closed-loop system for user management that is both responsive and reliable.

Case Study: Teacher and Attendance Workflow

After the administrative setup, a user with the "teacher" role can log in. The system correctly recognizes their role and takes them to the TeacherActivity dashboard. From there, the teacher can start a new session, which first requires them to verify their identity using the Biometric Prompt. This step acts as an important security check. After successful verification, the Attendance Activity starts, and the fetchAssignedStudents function runs. The system properly queries the classes collection and its assignedStudents sub-collection to fill a RecyclerView with the list of enrolled students for that specific session. Each student's status correctly starts as "Absent."

When the "Scan" button for a student is pressed, the biometric prompt appears again. A successful scan by the student triggers the markAttendance function. This function creates a new document in the attendance collection in Firestore with a "Present" status and a server timestamp. The application's real- time listener quickly detects this change, and the student's UI on the teacher's device updates to show their "Present" status, providing instant confirmation.

4.2. Analysis of Outcomes Compared to Expectations

The outcomes of the project closely match the initial goals:

1. *Proxy Attendance Elimination:* The main goal was fully achieved. By requiring on- device biometric verification for each student, the system makes it almost impossible for proxy attendance to happen. The student's identity is directly linked to their physical presence.
2. *Automation and Efficiency:* The goal to automate the process was met. The digital workflow removes the need for paper rosters and manual data entry. The fetchUsers function in AdminActivity, for example, replaces a manual process of compiling user lists.
3. *Centralized Management:* The AdminActivity and its associated user/class management screens successfully provide one secure point of control for all system data, as expected.
4. *Data Integrity and Reporting:* Using Firestore as a single source of truth ensures high data integrity. The ability to generate a PDF report gives administrators the portable, official documentation they need, meeting a key project requirement.

4.3. Strengths, Limitations, and Possible Improvements

Strengths:

- *High Security:* The project's biggest strength is its security model. By using the androidx.biometric library, the application never handles, stores, or sends sensitive fingerprint data, following modern privacy and security best practices.
- *Cost-Effectiveness:* The system requires no special or external hardware, using the sensors already in users' smartphones. This makes it an economical solution for educational institutions.
- *Scalability & Real-Time Functionality:* Using Google Firebase provides a highly scalable backend that can grow with an institution's needs. Its real-time capabilities are crucial for live-updating dashboards and attendance lists.
- *Modern and Efficient Architecture:* Using Kotlin and Coroutines for asynchronous operations— shown in AdminActivity—and a lifecycle-aware architecture leads to a high-performance, maintainable, and robust application.

Limitations:

- *Hardware and Platform Dependency:* The system depends on the Android platform and requires all users to have a device with a working, set-up fingerprint scanner. This leaves out users of iOS devices or older Android phones without biometric sensors.
- *Connectivity Requirement:* The current setup needs a constant internet connection to interact with Firebase. It lacks an offline-first mode, which could be a problem in areas with poor network coverage.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 03, March 2026)

Possible Improvements:

- *Offline-First Architecture:* Future work should focus on implementing a local database (e.g., Android Room) to cache data. This would allow the app to work offline, syncing data to Firestore once a connection is available.
- *Cross-Platform Support:* Creating a parallel version for iOS with similar technologies (e.g., Swift with Face ID/Touch ID) would be the most significant improvement, making the system available to a wider user base.
- *Enhanced Reporting and Analytics:* The reporting feature could expand to provide more detailed analytics, such as tracking a student's attendance percentage over a semester, identifying students with frequent absences, and exporting data in multiple formats like CSV.

V. Conclusion And Future Work

5.1 Summary of Key Findings

This research project has shown the design, implementation, and effectiveness of BioAttend, a biometric-based attendance management system for schools. The main finding is that by using the existing hardware of smartphones alongside a scalable cloud backend, it is possible to create a secure and cost-effective solution that addresses the issue of proxy attendance. The system's setup, which employs Kotlin Coroutines for asynchronous operations as seen in the AdminActivity's data-fetching logic, proves to be both efficient and reliable, ensuring a smooth user experience even with multiple database queries. The Role-Based Access Control (RBAC) model offers a clear and secure division of tasks, providing administrators with powerful tools for user and class management while simplifying attendance reporting for teachers. Additionally, the project shows that a secure biometric system can be created without sacrificing user privacy. The androidx.biometric library ensures that sensitive fingerprint information is processed within the device's secure hardware and never handled by the application.

5.2 Practical Applications of the Project

The practical applications of the BioAttend system are broad and impactful. It is primarily designed for educational settings like K-12 schools, colleges, and universities, where accurate attendance tracking is essential for academic records, compliance, and funding.

By automating the attendance process, the system saves valuable instructional time and eases the workload on faculty. Beyond traditional education, it also fits well in corporate training, workshops, and any situation where confirming attendee presence is necessary. The creation of portable PDF reports makes the gathered data immediately useful for administrative tasks, such as student performance evaluations or auditing. Since the system only requires a standard smartphone, it provides an accessible and cost- effective option for institutions of all sizes to upgrade their attendance methods.

5.3 Suggestions for Future Research or Enhancements

While the BioAttend project has successfully achieved its main goals, there are several areas for future research and improvement that could increase its effectiveness and usefulness:

1. *Offline-First Architecture:* The system currently relies on a constant internet connection, which is a significant drawback. Future research should aim to create an offline-first architecture. This would involve adding a local on-device database, like Android Room, to store all actions, such as marking attendance and creating users. The app would then sync this data with Firestore when a network connection is available, making the system more reliable in areas with weak connectivity.
2. *Cross-Platform Support:* The current version is only available on the Android platform. A key improvement would be to develop a similar app for iOS. This would mean researching and building an architecture using Swift, SwiftUI, and Apple's Face ID/Touch ID frameworks, ensuring that the system is available to all users, no matter what mobile device they have.
3. *Improved Reporting and Analytics:* The existing reporting feature, though functional, could be greatly enhanced. Future work could focus on creating a web-based administrative dashboard that offers detailed data analytics. This dashboard could show attendance trends, automatically identify students with low attendance, and allow for the creation of complex, customizable reports over specific date ranges, providing valuable insights for academic administrators.
4. Integration with Existing Systems: To boost its appeal for larger institutions, research could be done to integrate BioAttend with current Student Information Systems (SIS) or Learning Management Systems (LMS).

This would involve creating a secure API to enable the smooth transfer of attendance data, further reducing manual data entry and promoting consistency across institutional platforms.

REFERENCES

[1] S. C. Sabarinathan and C. P. Priya, "Biometric Attendance System using Fingerprint for Smartphones," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 7, no. 5, pp. 81-85, May 2020.

[2] A. R. Wankhade, A. D. Wankhade, and S. R. Wankhade, "Attendance Management System Using Mobile Biometric Technology," International Journal of Creative Research Thoughts (IJCRT), vol. 12, no. 5, pp. e803-e807, May 2024.

[3] N. B. H. A. Rahman and M. Z. B. A. Rashid, "Biometric Authentication Based Attendance System using GPS for Smart Phone," International Journal of Engineering Research & Technology (IJERT), vol. 5, no. 2, pp. 1-4, Feb. 2016.

[4] S. Valarmathy and R. Ramani, "Biometric Attendance System using Android Application," in Proc. 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-5.

[5] P. K. Gupta and S. K. Singh, "Android Based Attendance Management System Using Fingerprint Scanner," IEEE Transactions on Consumer Electronics, vol. 62, no. 1, pp. 12-19, 2021.

[6] P. P. Shendge, "Digital Attendance System Using Google Cloud Firebase and Gradle," International Journal of Research in Engineering, Science and Management, vol. 2, no. 4, pp. 234-236, 2019.

[7] M. Al-Khatib and A. Al-Sharafi, "Cloud- Based Database Management with Firebase," International Journal of Computer Applications, vol. 182, no. 45, pp. 33-38, 2023.

[8] A. Jain, "Real-time Android Application Development using Kotlin and Firebase," in Proc. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 782-786.

[9] R. Kumar and A. S. Al-Malaise, "Cloud Based Smart Attendance System for Educational Institutions," in Proc. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 14-18.

[10] O. Shoewu and O. A. Idowu, "Development of Attendance Management System using Biometrics," The Pacific Journal of Science and Technology, vol. 13, no. 1, pp. 300-307, 2012.

[11] K. Saraswat and R. Gupta, "A Review on Attendance Management System using Biometric Fingerprint Technology," International Journal of Advanced Research in Computer Science, vol. 8, no. 5, pp. 110-114, 2017.

[12] J. S. M. Ali, "Smart Attendance System: A Comparative Study of RFID and Biometric Methods," in Proc. 2021 IEEE International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, UAE, 2021, pp. 305-310.

[13] T. Lim, "Android-based Attendance Management System," in Proc. 2015 IEEE Conference on Systems, Process and Control (ICSPC), Bandar Sunway, Malaysia, 2015, pp. 120- 124.

[14] M. K. Rao, "Security Issues in Mobile Based Biometric Authentication," International Journal of Network Security, vol. 22, no. 3, pp. 455-462, 2022.

[15] A. B. Smith, "A Comparative Analysis of Manual vs. Automated Attendance Systems in Higher Education," IEEE Access, vol. 9, pp. 10234- 10245, 2021.