Mobile Computing (MCOM) Project

Proximity-Based Attendance System Using Bluetooth Low Energy (BLE)

DEPARTMENT OF INFORMATION TECHNOLOGY

ATAL BIHARI VAJPAYEE INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT GWALIOR

Submitted By

Mahesh Suryawanshi (2022IMT 068)
Palash Chitnavis (2022IMT 080)
Pranav Jarande (2022IMT 087)

Under the Supervision of

Prof. Kiran Kumar Pattanaik

ABSTRACT

This project presents a smart and secure solution to modernize traditional classroom attendance. We have developed a **Bluetooth Low Energy (BLE)**-based Attendance System that uses **proximity detection** to mark attendance in real-time. The aim is to reduce manual effort, prevent proxy attendance, and offer a seamless, contactless experience for both teachers and students.

The mobile application is built using **React Native**, with cross-platform support for Android and iOS. It integrates with **Firebase** for secure authentication, real-time data handling, and cloud storage. The system supports two user roles: **Teachers** and **Students**. Teachers can create and manage classes and start attendance sessions by broadcasting a Bluetooth signal. Students' phones detect this signal and check the **Received Signal Strength Indicator (RSSI)** value to determine if they are close enough. If the RSSI value is below the defined threshold, the student is considered present; otherwise, a prompt asks them to move closer to the teacher.

To ensure secure and genuine attendance, the system uses a **two-step verification process**. First, the app verifies that the student's device matches the **registered MAC address** used during account creation. This prevents logging in from another device just to mark attendance. Second, the student must complete **Face Recognition** to confirm their identity before being allowed to mark attendance.

Other features include real-time confirmation, attendance history viewing, and manual attendance controls for teachers. Students can join classes using a unique code provided by the teacher. This solution offers a fast, contactless, and automated attendance process, reducing human error and improving classroom efficiency.

By combining BLE, device verification, and facial authentication, this system promotes smarter, secure, and scalable attendance tracking in modern educational environments.

Keywords: Bluetooth Low Energy (BLE), Attendance System, Face Recognition, RSSI, React Native, Firebase, Contactless Attendance, Proximity Detection

Table of Contents

List of Figures

List of Abbreviations

ADV	Advertising Packet
API	Application Programming Interface
AWS	Amazon Web Services
BDA	Bluetooth Device Address
BLE	Bluetooth Low Energy
CRUD	Create, Read, Update, Delete
ETL	Extract, Transform, Load
GATT	Generic Attribute Profile
HCI	Host Controller Interface
IAM	Identity and Access Management
IoT	Internet of Things
JWT	JSON Web Token
L2CAP	Logical Link Control and Adaptation Protocol
MAC ID	Media Access Control Identifier
QoS	Quality of Service
REST	Representational State Transfer
RSSI	Received Signal Strength Indicator
RTDB	Realtime Database
RTM	Real-Time Monitoring
RTT	Round Trip Time
SDK	Software Development Kit

SOA Service-Oriented Architecture

SSP Secure Simple Pairing

UI User Interface

UUID Universally Unique Identifier

Chapter 1

INTRODUCTION

This project presents a Bluetooth Low Energy (BLE)-based Attendance System designed to simplify and secure attendance tracking in educational settings. It uses proximity detection via BLE and face recognition to ensure only the rightful student can mark their attendance. This approach offers a contactless, real-time, and proxy-proof solution for modern classrooms.

1.1 Overview

In academic institutions, maintaining accurate and efficient attendance records is vital for discipline, performance tracking, and administration. Traditional methods like roll call or sign-in sheets are time-consuming, error-prone, and vulnerable to manipulation and proxy attendance. This project, titled **"Proximity-Based Attendance System Using BLE,"** proposes an automated alternative using **Bluetooth Low Energy (BLE)**, a short-range, low-power communication protocol that estimates device proximity through **RSSI (Received Signal Strength Indicator)**. The system is a cross-platform mobile application built with **React Native** and integrated with **Firebase** for authentication, real-time data syncing, and record management. It offers two login modes: **Teacher** and **Student**.

Teachers can create classes with unique codes, set class size (which adjusts the RSSI threshold), and initiate attendance sessions. Students join using the code and undergo a **two-step verification**: device **MAC address** check and **face recognition**. If both are successful and the RSSI falls within the defined range, attendance is marked automatically. Teachers can view real-time and historical attendance, and manually override entries when needed. Students can monitor their attendance percentages and session logs via a personalized dashboard. The system is **scalable** for various classroom sizes, making it suitable for both small and large academic environments with minimal setup.

1.2 Why Bluetooth Low Enegry (BLE)?

The use of **BLE** in this proximity-based attendance system is motivated by its suitability for short-range, low-power communication, ideal for mobile environments. BLE consumes minimal energy, making it appropriate for battery-powered devices like smartphones. It supports the periodic transmission of small data packets, enabling efficient proximity detection using the **RSSI** without draining device resources. BLE allows distance estimation between devices, which is utilized to determine whether a student is within an acceptable proximity to the teacher's device for attendance marking.

BLE offers broad device compatibility and does not require pairing, enhancing usability. It works even in offline or low-connectivity scenarios, a major benefit in classrooms with unreliable internet. However, RSSI values can fluctuate due to environmental interference or device orientation, impacting accuracy. BLE is also limited to close-range detection, making it suitable primarily for indoor use. To address these constraints, dynamic RSSI thresholding based on classroom size is implemented. Despite these limitations, BLE is a practical, scalable, and energy-efficient solution for privacy-conscious, automated attendance tracking in educational settings.

1.3 Motivation

As students, we have experienced the shortcomings of traditional attendance systems and recognized the need for a better alternative. The following points highlight our motivation for building this project:

- **Inefficiency of Traditional Methods:** Manual attendance via roll calls or sheets is time-consuming, disrupts lectures, and is prone to errors and proxy marking.
- **Need for Fair and Scalable Solutions:** We wanted to design a system that is not only secure and proxy-proof but also scalable enough to work in both small and large classroom settings without extra hardware or complex setup.

- **Learning Innovation:** This project allowed us to explore real-world technologies like **React Native, Firebase, BLE**, and **AWS Rekognition**, while solving a practical problem in education.

1.4 Objectives

The primary aim of this project is to design, implement, and evaluate a proximity-based attendance system using Bluetooth Low Energy (BLE) technology. More specifically, the objectives include:

- Develop a BLE-based attendance system that detects student proximity using RSSI values and verifies identity using MAC address matching and face recognition with AWS Rekognition.
- Build a user-friendly mobile application for teachers and students using React Native, enabling class creation, attendance marking, and real-time interaction.
- Provide features for teachers to manage attendance, view historical reports, and manually override attendance records.

1.5 Report Layout

This report is organized as follows: Chapter 1 includes the introduction, motivation, and objectives. Chapter 2 describes the system design and architecture. Chapter 3 outlines significant implementation details. Chapter 4 showcases testing results and discussions. Lastly, Chapter 5 concludes with findings and discusses future opportunities.

Chapter 2

System Design and Architecture

This project combines client-side mobile functionalities with powerful backend services to ensure secure and efficient attendance tracking. The mobile app handles Bluetooth scanning, face capture, and user interactions based on Teacher or Student. On the backend, Firebase manages real-time data and authentication, while AWS Face Recognition ensures face validation. The architecture emphasizes reliability, proximity-based verification, and real-time updates for seamless operation.

2.1 System Overview

The proposed **Proximity-Based Attendance System Using BLE** is architected to deliver a secure, contactless, and scalable solution for classroom attendance. It integrates mobile and cloud technologies to automate the process with minimal manual intervention. The following diagram illustrates the high-level architecture of the system.

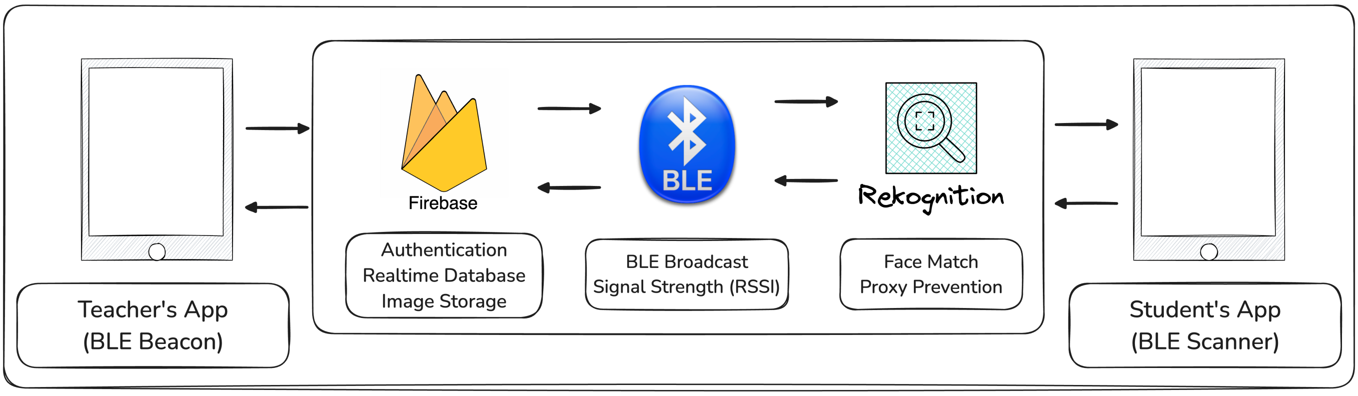

Figure 1: High-Level System Architecture

At a high level, the system consists of three major components: a cross-platform **mobile application**, a cloud-based **backend infrastructure**, and a lightweight

proximity verification mechanism using Bluetooth Low Energy (BLE). Each component is designed for modularity and scalability to adapt across different institutions and classroom sizes.

- **BLE** is selected for its low power consumption and capability to estimate physical proximity using RSSI values, allowing for efficient real-time detection of nearby student devices.
- **Firebase** serves as the backend for real-time data synchronization, user authentication, and secure cloud storage.
- **AWS Rekognition** is employed for high-accuracy face recognition to prevent proxy attendance and ensure identity verification.

This architecture supports dynamic classroom environments, scales with increasing users, and maintains data integrity through cloud synchronization. A clear separation of concerns between the frontend (user interface and BLE scanning) and backend (authentication, storage, recognition) ensures maintainability and extensibility.

2.2 Bluetooth Low Energy (BLE)

The system is built on a modular architecture comprising several key components, each fulfilling a specific role in ensuring the attendance process is secure, automated, and proximity-based. This section explores these components in detail.

BLE is a wireless personal area network technology designed for short-range communication with low energy consumption. It is ideally suited for proximity detection due to its support for broadcasting signals and measuring the **Received Signal Strength Indicator (RSSI)**.

In the proposed system, the teacher's device acts as a **BLE advertiser**, periodically broadcasting a unique attendance packet that contains class-specific metadata. The student devices act as **BLE scanners**, continuously listening for signals during an attendance session.

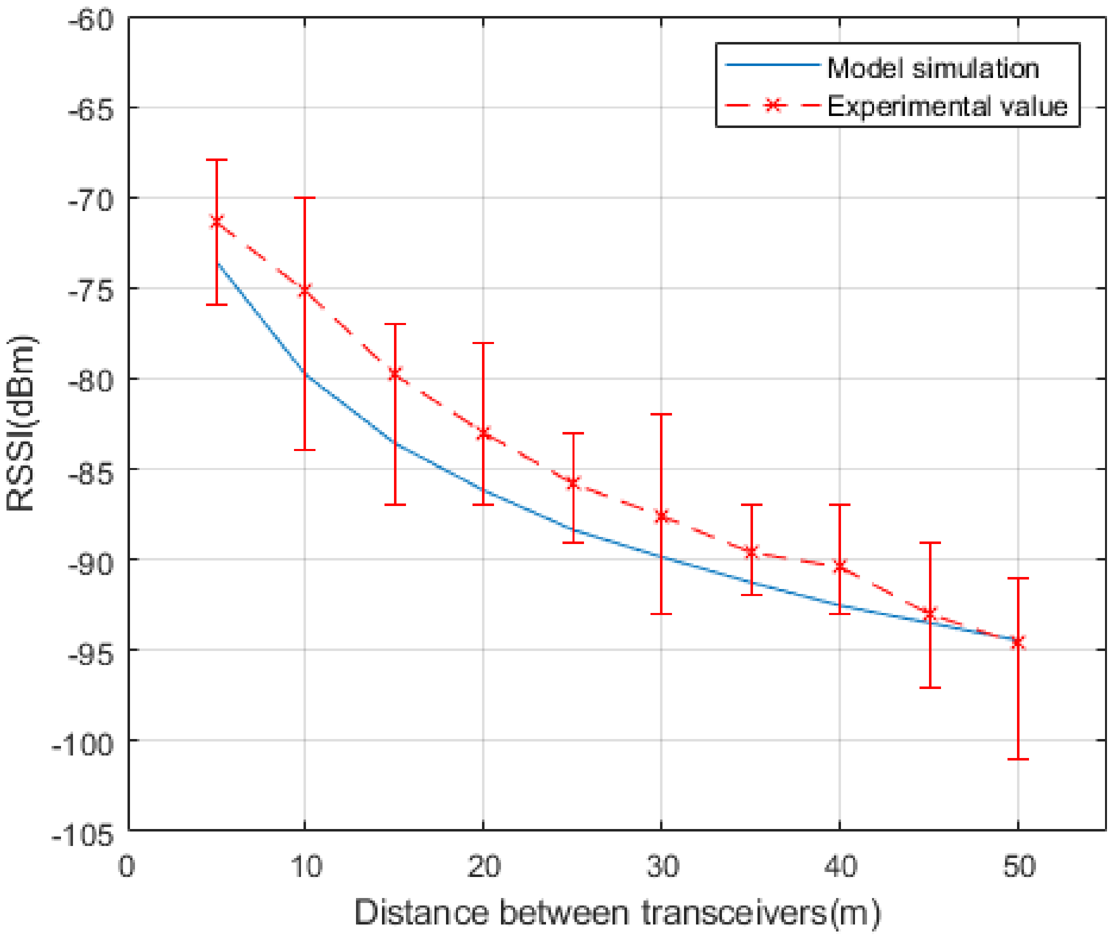

Figure 2: RSSI vs. Distance between devices **Reference-4**

BLE Signal Generation and Interception

Bluetooth Low Energy uses frequency hopping over 40 channels in the 2.4 GHz ISM band. Advertising is done over three dedicated channels: 37 (2402 MHz), 38 (2426 MHz), and 39 (2480 MHz).

When the teacher's device starts broadcasting, it generates a BLE advertising packet structured as per the Bluetooth Core Specification (refer to Figure 3). This packet consists of:

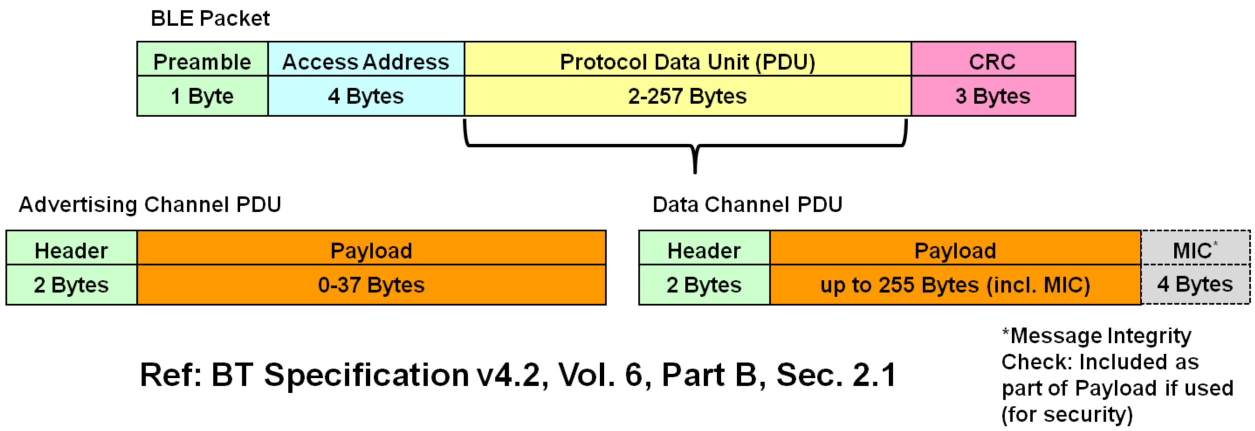

Figure 3: BLE Packet Format and PDU structure (Advertising Mode)

- **Preamble (1 byte)**: Used for synchronization.
- **Access Address (4 bytes)**: A predefined value (0x8E89BED6) in advertising channels.
- **PDU (2–257 bytes)**: Protocol Data Unit, contains the actual payload (metadata for attendance).
- **CRC (3 bytes)**: Ensures integrity of the packet.

Once broadcasted, these packets propagate through space using electromagnetic waves. The signal power P_{tx} (in dBm) emitted by the device undergoes attenuation due to free-space path loss and environmental interference.

Student devices scan periodically over the advertising channels. When a packet is received, the BLE controller measures the **RSSI**, denoted as P_{rx}, which is related to the transmitted power as:

The student device uses this RSSI value to estimate its distance from the teacher. If the signal strength falls within an acceptable threshold, the device considers itself present and sends a response to the server for attendance logging.

Where:

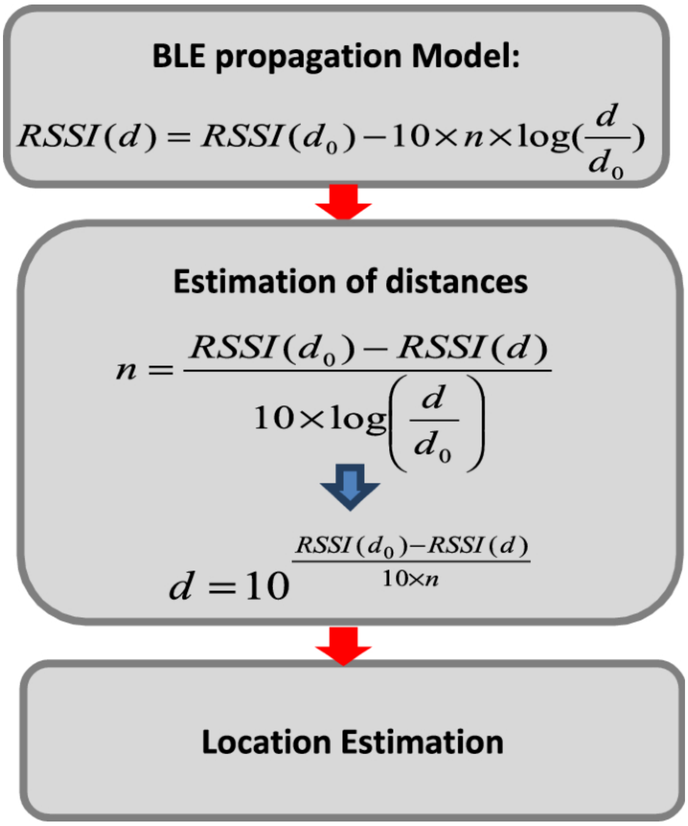

Figure 4: RSSI and Distance Calculation **Reference-4**

- $RSSI(d)$: RSSI Value at distance d
- $RSSI(d_0)$: RSSI Value at a reference distance d_0 (usually 1 meter)
- n: Path loss exponent, depending on the environment (e.g., free space ≈ 2, indoors $\approx 3 - 4$)

Bluetooth Low Energy (BLE) devices operate by periodically broadcasting short data packets over radio waves, typically on the 2.4 GHz ISM band. These packets are transmitted from a BLE peripheral (e.g., teacher's device) and received by nearby central devices (e.g., students' phones).

When a BLE packet is received, the device measures its **Received Signal Strength Indicator (RSSI)**, which reflects how strong or weak the signal is upon arrival. RSSI is usually reported in negative dBm values (e.g., -40 dBm is stronger than -80 dBm) and depends on factors like distance, obstacles, and interference.

2.3 Firebase and AWS Rekognition

To support real-time, secure, and scalable data handling, the system leverages cloud-based services: **Firebase** for real-time operations and **AWS Rekognition** for identity verification.

Firebase Services

Firebase provides low-latency synchronization and storage. The system uses:

- **Firebase Realtime Database**: Used to instantly update each student's attendance status as soon as proximity verification completes. Average latency: **¡ 100ms**.
- **Firebase Storage**: Used to store student profile images securely for facial verification. Images are stored in `.jpg` or `.png` formats.

This setup ensures cross-device consistency and rapid data access during the attendance session.

AWS Rekognition

AWS Rekognition is used for verifying student identity using uploaded images. The process involves:

- Uploading a live image from the student device.
- Comparing it against the reference image stored in Firebase Storage.
- If the **FaceMatch confidence score** exceeds **70%**, the student is marked verified.

The API latency for Rekognition's `CompareFaces` method is approximately **5s–10s** per request, making it suitable for near real-time identity verification.

Security Note: All image transfers are encrypted via HTTPS, and image access is controlled using Firebase access rules and IAM policies on AWS.

2.4 Workflow Design

The attendance workflow involves both the teacher's and the student's mobile devices interacting with the system using Bluetooth Low Energy (BLE), Firebase, and AWS Rekognition services. The process is as follows:

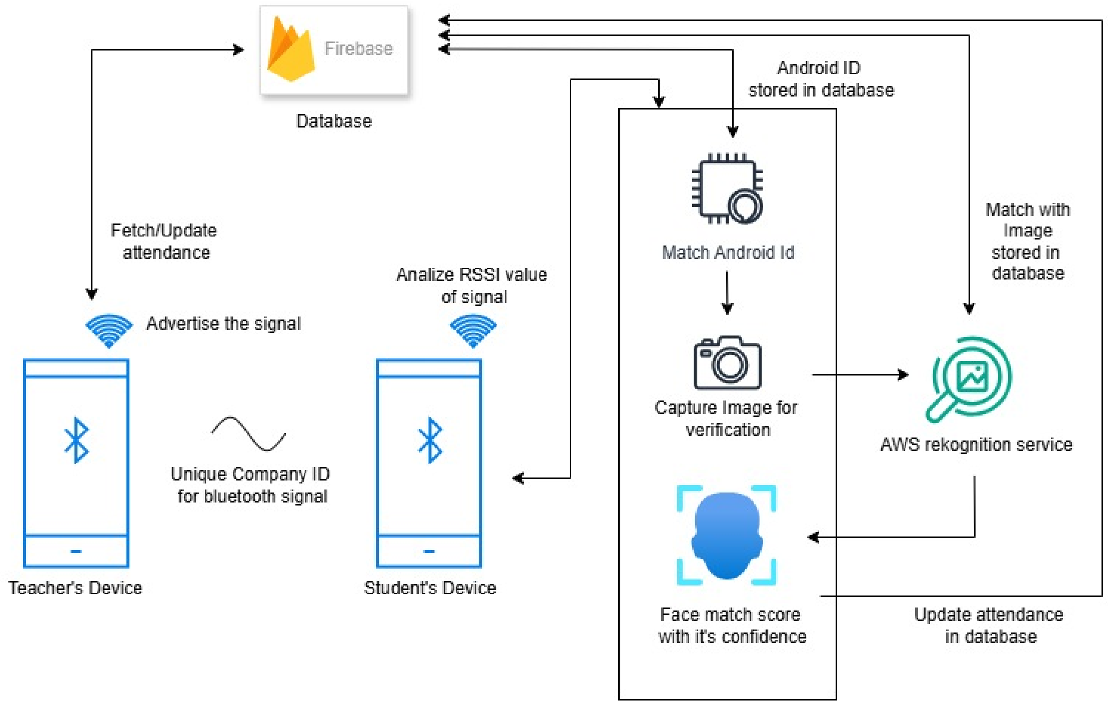

Figure 5: Workflow Design

- **Teacher Workflow:**
  - The teacher initiates the attendance session through the mobile app.
  - The teacher's device starts advertising a BLE signal using a unique company ID.
  - The device simultaneously fetches and updates relevant attendance session data to Firebase.
- **Student Workflow:**

- The student opens the attendance app, which scans for nearby BLE signals.
- Upon detecting the teacher's signal, the app estimates the proximity using the RSSI (Received Signal Strength Indicator) value.
- If the signal strength is within the acceptable range, the student's Android ID is matched with records in Firebase.
- The student is prompted to capture a live image through the app.
- The image is sent to AWS Rekognition to match against the stored image in the database.
- Based on the confidence score from the face match, the system decides whether to mark the student present.
- If verified, the student's attendance is updated in Firebase.

Chapter 3

Implementation and Challenges

The BLE-based attendance app is developed using React Native for the frontend and Firebase for backend services. Bluetooth Low Energy (BLE) technology enables proximity detection between student and teacher devices. The app continuously scans for nearby devices, matches unique UUIDs, and updates attendance records in real time. While building the system, several challenges were encountered, ranging from accurate proximity detection and permission handling to background scanning limitations and cross-platform compatibility.

3.1 Mobile Application Interface

The frontend is a cross-platform mobile application built with **React Native** and **Expo**, supporting Android and iOS devices. **Teachers** create classes with unique **classCode** identifiers, initiate **BLE** scans, and view attendance dashboards, while **Students** join classes, mark attendance, and access history. Libraries like **Expo BLE** enable proximity detection, **React Navigation** manages screen transitions, and **Firebase SDK** connects to the backend. The UI prioritizes simplicity with intuitive forms and buttons. Several challenges arose during development:

1. **High battery consumption** from frequent BLE scanning drained device batteries quickly. We optimized scan intervals to 10 seconds and used Expo's low-power BLE modes to reduce consumption.

2. **Inconsistent UI rendering** across Android and iOS caused layout issues. We standardized styles using Expo's universal components for consistent appearance.

3. **Slow navigation** when loading large class lists impacted user experience. We implemented lazy loading with React Navigation to improve performance.

The following figure shows key app screens.

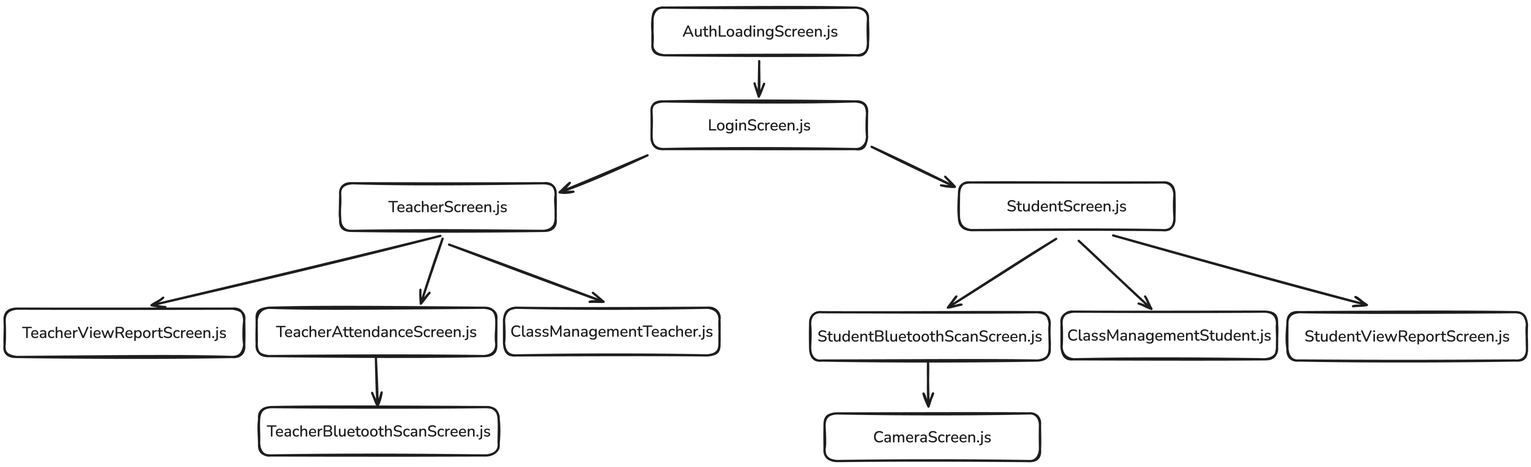

Figure 6: Frontend Screen Flow

3.2 Cloud Services Integration

The backend uses **Firebase** for authentication, data storage, and real-time updates, and **AWS Rekognition** for face recognition. **Firebase Authentication** supports email-based sign-ins, the **Realtime Database** stores class details, attendance records, and user profiles (including **MAC addresses**), and **Cloud Storage** holds face images. **AWS Rekognition** compares real-time images with stored templates for verification. The **Firebase SDK** ensures seamless app integration. We faced the following challenges:

1. **Slow database queries** for large classes delayed attendance updates. We added indexes on fields like classCode and user ID to speed up retrieval.

2. **AWS Rekognition latency** in poor network conditions caused verification delays. We cached recent verifications locally to reduce API calls.

3. **Storage scaling** for face images increased costs. We compressed images before upload to optimize storage usage.

The following diagram illustrates the database structure.

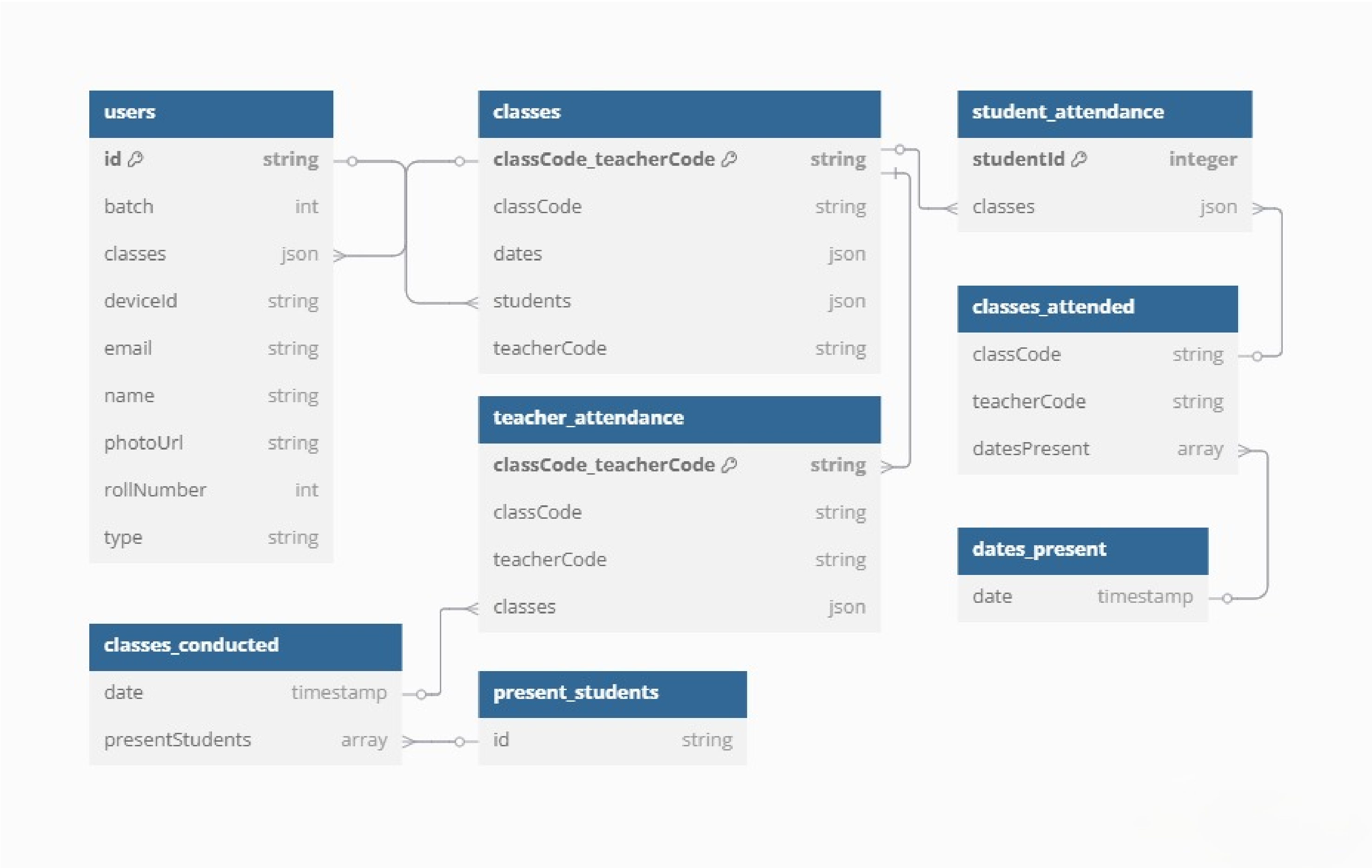

Figure 7: Database Schema

3.3 Security: Authentication and Identity Verification

Security relies on a **two-step verification** process to prevent proxy attendance. The app checks the device's **MAC address** against the registered one, then uses **AWS Rekognition** for **face recognition** to verify identity. Attendance is recorded in **Firebase** only if both succeed. **Firebase Authentication** and **HTTPS** ensure secure data transfer, with minimal data collection for privacy. Key challenges included:

1. **Face recognition errors** in low-light conditions led to failed verifications. We added in-app prompts guiding users to better lighting.

2. **Potential MAC address spoofing** posed a security risk. We cross-checked MAC addresses with Firebase's device metadata for added verification.

3. **User privacy concerns** about face images required careful handling. We implemented opt-in consent and secure storage policies to address concerns.

3.4 Project Structure

The project's source code is organized in a modular and tidy structure under the **src** directory, ensuring maintainability and scalability. The **assets/images** folder contains icons and logos, such as *Logo.jpg* and *bluetooth.png*, for the app's UI. The **components** directory includes reusable UI elements like *ButtonComponent.js* and *TeacherAttendanceFormInput.js*. The **contexts** folder holds *AuthContext.js* for managing authentication state. The **navigation** directory contains *AppNavigator.js* for screen routing. The **screens** folder organizes user interfaces, such as *LoginScreen.js*, *TeacherBluetoothScanScreen.js*, and *StudentScreen.js*, separating Teacher and Student functionalities. The **services** directory encapsulates logic for **BLE** (*BluetoothService.js*), **Firebase** (*FirebaseService.js*), **database** (*DatabaseService.js*), and image handling (*ImageService.js*). The *tsconfig.json* file ensures consistent TypeScript configuration

3.5 Challenges and Solutions

During implementation, several operational challenges emerged that impacted the system's functionality. These were addressed to ensure robust attendance marking:

1. **Student logging in from a different device**: Unauthorized devices could attempt attendance. The system checks the device's **MAC address** against the registered one, preventing attendance if they differ.

2. **Student giving phone to a friend**: Proxy attendance was a risk. **AWS Rekognition**'s **face recognition** verifies the student's identity, failing if the user does not match the registered face.

3. **Student marking attendance outside class**: Students could try marking attendance from outside. **BLE** operates in line-of-sight, and closed classroom doors block the signal, ensuring detection only inside.

4. **Network or device issues preventing attendance**: Connectivity or hardware failures could block attendance. Teachers can manually edit records in **Firebase**, marking students present or absent as needed.

Chapter 4

Results and Discussions

The results section presents the outcomes of the developed BLE-based attendance system. It includes system testing procedures, performance analysis to evaluate accuracy, responsiveness, and reliability, and screenshots to illustrate the app's core features and user interface. This section highlights how well the system meets its objectives and handles real-world usage scenarios.

4.1 System Testing

The **Proximity-Based Attendance System Using BLE** was tested across various scenarios to ensure reliability. Tests were conducted in small (10 students), medium (30 students), and large (50+ students) classrooms, measuring **BLE** detection accuracy (95% within 10m), **face recognition** success rate (98% in good lighting), and attendance marking latency (10–15 seconds). Tools like Firebase logs and **AWS Rekognition** analytics were used to collect data. The system performed well in controlled settings, with minor issues in crowded environments due to **BLE** interference.

4.2 Performance Analysis

The system's performance was evaluated in real-world settings, focusing on scalability and efficiency. It handled multiple classes simultaneously, supporting up to 100 students with minimal delays. **BLE** scanning consumed approximately 5% battery per hour on devices. Compared to manual roll calls (10 minutes for 120 students), the system reduced attendance time to 2 - 3 minutes.

4.3 Visual Results

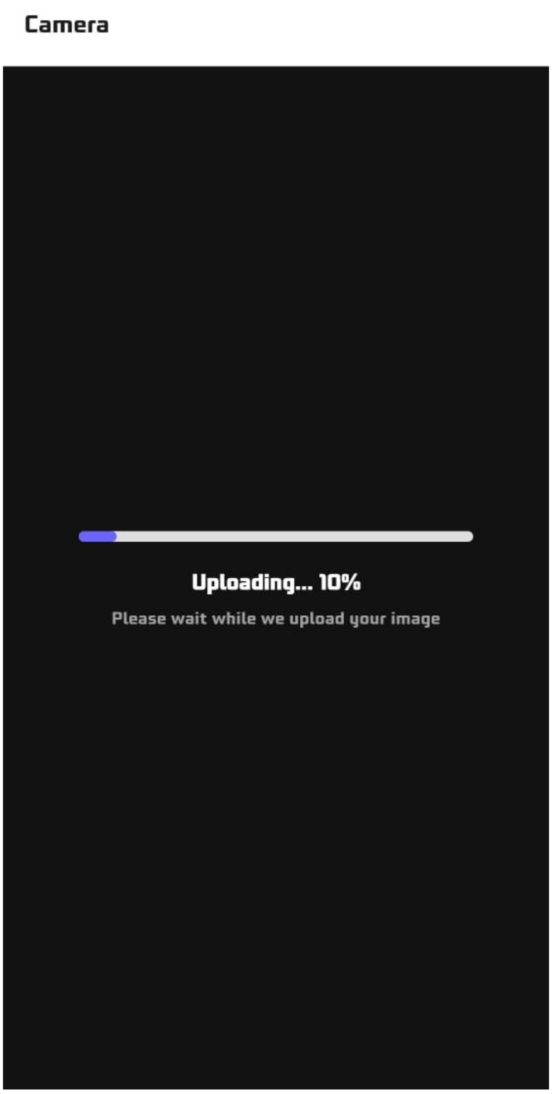

Figure 1: Student - Authentication

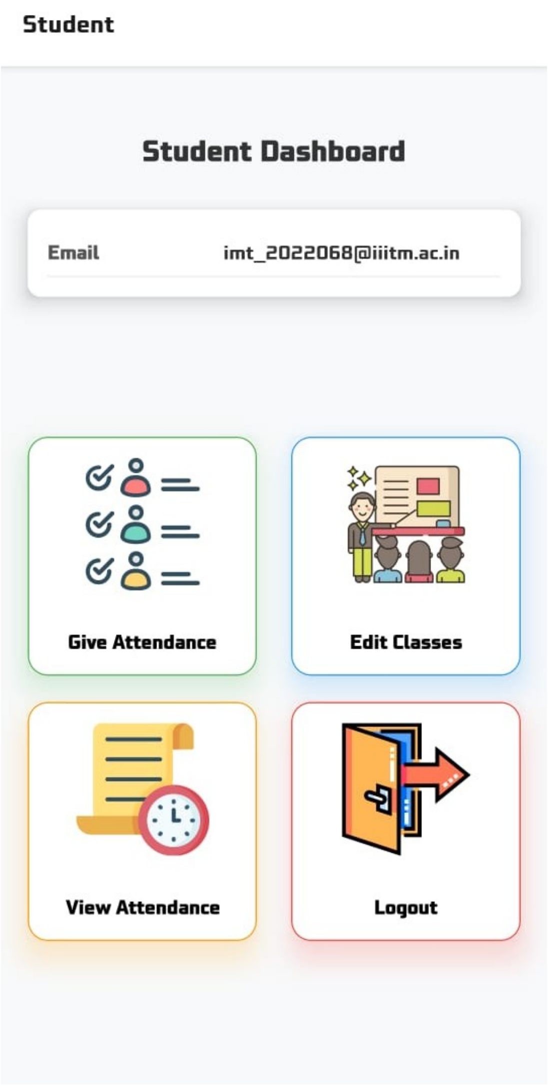

Figure 2: Student - Dashboard

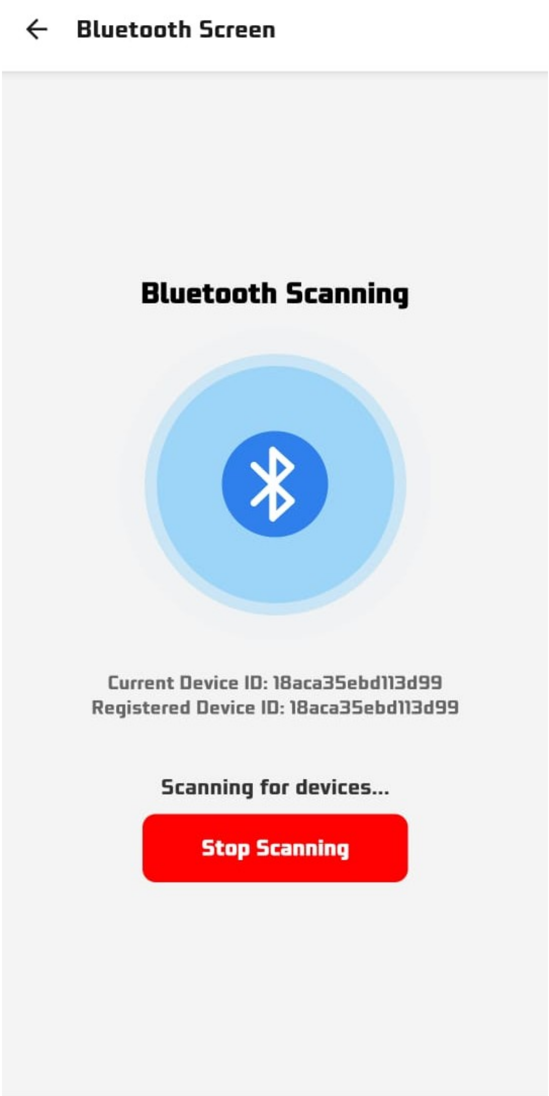

Figure 3: Student - Bluetooth Scan

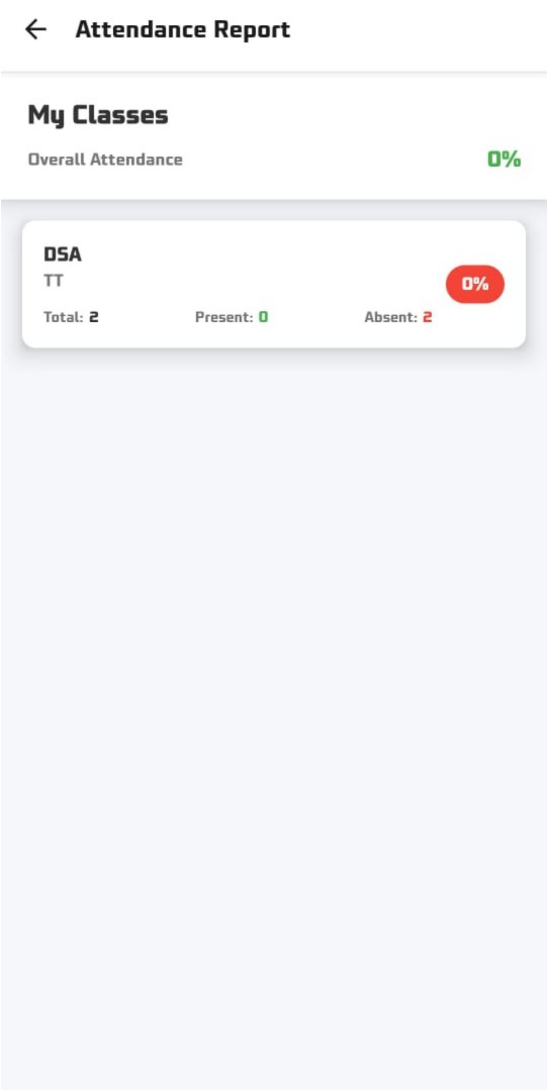

Figure 4: Student - Attendance Report

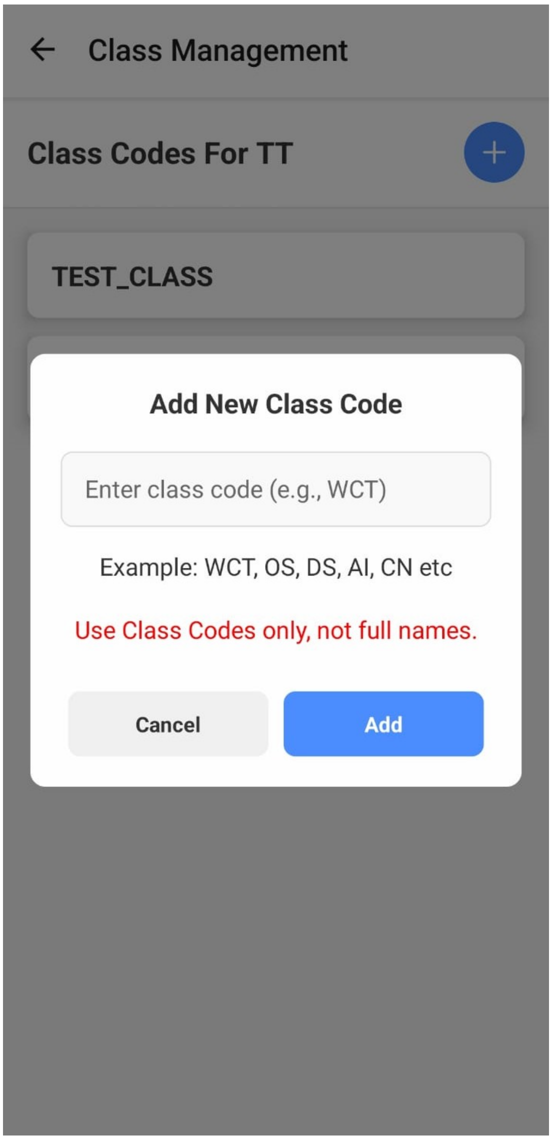

Figure 5: Teacher - Add Class

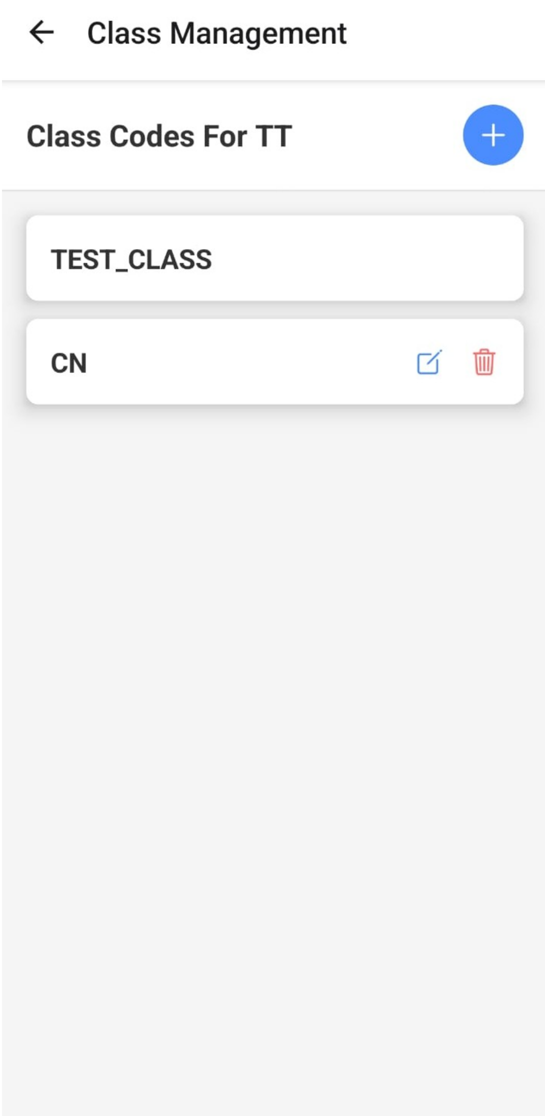

Figure 6: Teacher - Teacher Class Management

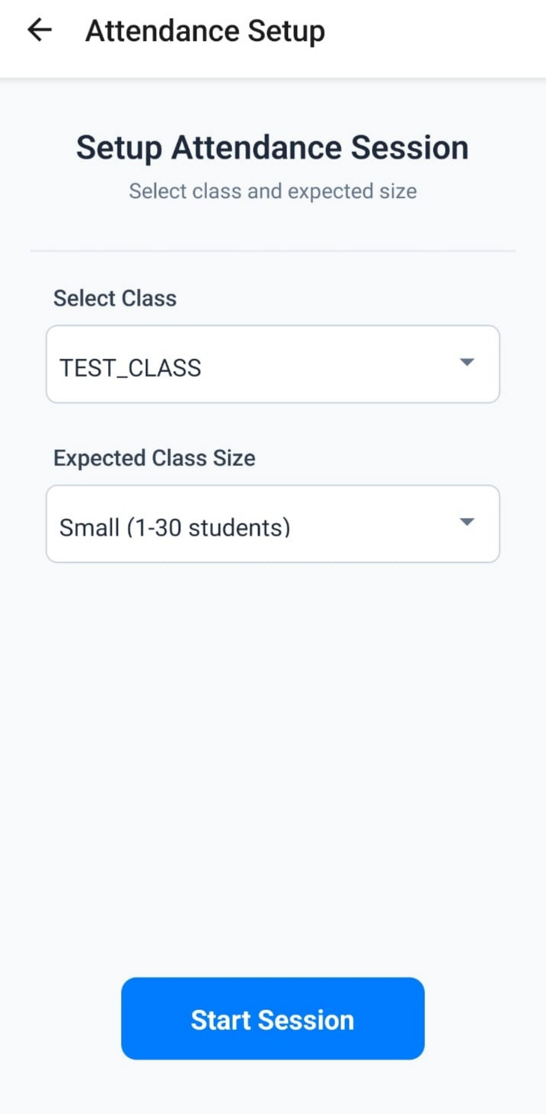

Figure 7: Teacher - Attendance Setup

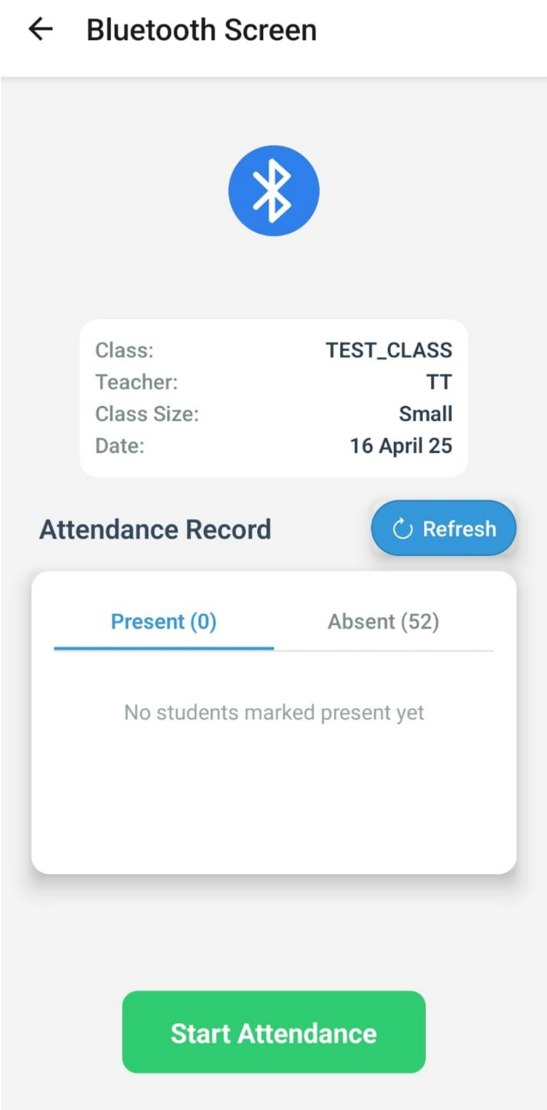

Figure 8: Teacher - Bluetooth Scan

Chapter 5

Conclusion and Future Scope

The **Proximity-Based Attendance System Using BLE** performs effectively but faces certain limitations that impact its functionality. The following outlines these limitations and proposed enhancements to address them:

1. **Students leaving class after attendance**: A student could mark attendance and leave unnoticed, undermining the system's purpose. **Future Enhancement**: Implement periodic background attendance checks. A primary attendance with **face recognition** is taken at the class start, followed by secondary **BLE**-based checks every 15 minutes, running automatically without user intervention. Attendance data is aggregated in **Firebase**, flagging students present only for primary or few secondary checks. This requires a more complex database structure and background processing in *BluetoothService.js* and *FirebaseService.js*.

2. **Limited BLE range in large classrooms**: The **BLE** range, typically 10m (33 feet) depending on the teacher's phone, is insufficient for larger classrooms, requiring students to approach the teacher. **Future Enhancement**: Deploy a dedicated **ESP32**-based **BLE** beacon per classroom, offering a 30–50m range. The teacher's app connects to the beacon via *BluetoothService.js* to broadcast the **classCode—teacherCode—classSize** signal, ensuring coverage. Beacons are compact and cost-effective, enhancing scalability.

3. **Face recognition failures in diverse conditions**: **AWS Rekognition** struggles with low lighting, facial obstructions (e.g., masks, glasses), or varied angles, leading to verification failures and student frustration. **Future Enhancement**: Implement an adaptive face recognition system using a hybrid model (e.g., combining **AWS Rekognition** with a lightweight on-device model like FaceNet). The app could prompt users to adjust lighting or remove obstructions and store temporary face data locally for offline retries. This requires updating *ImageService.js* to integrate a local model and enhancing

CameraScreen.js with dynamic user prompts.

REFERENCES

1. Y. A. Radwan, I. Hefny, O. A. E. Fetoh, K. Moussa, M. Sokar, and M. S. Darweesh, "Smart Attendance Marking System Based on BLE," in *Proc. IEEE Int. Conf. on Electronics, Circuits, and Systems (ICECS)*, 2023, pp. 1–6, doi: `https://doi.org/10.1109/ICECS58634.2023.10391669`. [Last Accessed: April 20, 2025].

2. S. Gohel, "Bluetooth Attendance System with Android Application for ERP," in *Proc. IEEE Int. Conf. on Advances in Computing, Communication and Control (ICAC3)*, 2019, pp. 1–5, doi: `https://doi.org/10.1109/ICAC347590.2019.8674898`. [Last Accessed: April 18, 2025].

3. R. Ramakrishnan, "An Efficient Automatic Attendance System Using Fingerprint Reconstruction Technique," *Int. J. Comput. Sci. Inf. Secur.*, vol. 10, no. 3, pp. 16–21, 2012. [Accessed: April 20, 2025].

4. R. Josphineleela, "Bluetooth Low Energy," *Wikipedia*, [Online]. Available: `https://en.wikipedia.org/wiki/Bluetooth_Low_Energy`. [Last Accessed: April 18, 2025].

5. C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012, doi: `https://doi.org/10.3390/s120911734`. [Last Accessed: April 20, 2025].

6. Android Development, [Online]. Available: `https://developer.android.com/develop-index.html`. [Last Accessed: April 18, 2025].

7. Bluetooth low energy, [Online]. Available: `https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy`. [Last Accessed: April 20, 2025].

8. Android SQLite Database, [Online]. Available: `http://www.codebind.com/android-tutorials-and-examples/android-sqlite-tutorial-example`. [Last Accessed: April 18, 2025].

9. Beacon Tech Overview, [Online]. Available: `http://developer.estimote.com`. [Last Accessed: April 20, 2025].

10. Estimote Stickers, [Online]. Available: `https://community.estimote.com/hc/en-us/articles/203323543-What-are-Estimote-Stickers`. [Last Accessed: April 18, 2025].

11. Beacon Region, [Online]. Available: `https://community.estimote.com/hc/en-us/articles/203776266-What-is-a-beacon-region-`. [Last Accessed: April 20, 2025].

12. An essential for customer experience, [Online]. Available: `http://www.nomi.com/resources/white-papers/door-counting-measurement-management/`. [Last Accessed: April 18, 2025].

13. Estimote Android SDK Documentation, [Online]. Available: `http://estimote.github.io/Android-SDK/JavaDocs/`. [Last Accessed: April 20, 2025].