SAFE Bluetooth Attendance

BTP II Project Report

submitted for the course

CS 496 (Spring'25)

By

Soupati Sri Nithya
210050152
210050152@iitb.ac.in

Palle Bhavana
210050111
210050111@iitb.ac.in

Guide: **Prof. Bhaskaran Raman**
br@cse.iitb.ac.in

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

April 2025

Contents

1 Introduction

SAFE (Smart Authenticated Fast Examination) is a mobile app designed to make online exams secure and auto graded in a controlled exam setting. Normally smartphones are not allowed in exam halls because they can be used for cheating. However SAFE takes a different approach by allowing students to use their own smartphones in a controlled way. This system makes exams easier, faster and cheating resistant

The SAFE system has two main parts:

- A mobile app that students use to take their exams
- A web server and WiFi infrastructure to enable app server communication

SAFE is built around four key goals:

Feature	**Description**
Easy Setup	The Bring Your Own Device model, app-based system make it easy to set up
Cheating-Free	The app locks itself during the exam, and any suspicious behavior is reported
Reliability	System syncs regularly to ensure it works smoothly, even in tricky situations
Scalability	works well for both small, large exams by optimizing Wi-Fi usage & scheduling

Table 1: Key Features of SAFE System

To make sure exams are fair, students must mark their attendance before starting an exam. Currently, SAFE offers three ways to mark attendance:

1. Wi-Fi Based Attendance
2. Bluetooth Based Attendance
3. Audio Based Attendance

Right now, Wi-Fi-based attendance is used, but it has a problem. The Wi-Fi signal can go beyond the classroom, which means students outside the room can mark attendance even if they aren't there.

2 Aim of BTP

In this project, we focus on enhancing the Bluetooth based attendance system in SAFE (Smart Authenticated Fast Examination). The goal is to create a more secure, location-sensitive way for students to mark their attendance, ensuring that only those physically present in the classroom can do so.
Currently, SAFE uses Wi-Fi scanning for attendance, which allows students outside the classroom to mark attendance due to the Wi-Fi signal's range. A previous R&D project by Manish Kumar and Omkar Kadam introduced Bluetooth-based attendance to address this issue, but it has yet to be integrated and tested in real classroom settings.

Work Involved:

- **UI Modifications:** Update the app interface for Bluetooth attendance.
- **Stress Testing Existing Code:** Ensure the system works as expected.
- **Backend Improvements:** Update both server-side and Android client components.
- **Field Evaluation:** Test the system in a real classroom and address any issues.

3 UI Modifications

3.1 Bluetooth Button Added in Attendance Mode Selection

Bluetooth button has been added and its functionality has been integrated into the app. Students can now mark their attendance using Bluetooth in addition to the existing methods.

3.1.1 Before:

- When the "Mark Attendance" button was clicked, a popup appeared-
  "Do you want Audio based or Wifi based Attendance?"
  with Audio, Wifi, Cancel buttons
- But the popup used to get disappeared in less than a second and got redirected automatically to mark attendance using WiFi, without giving users the time to choose

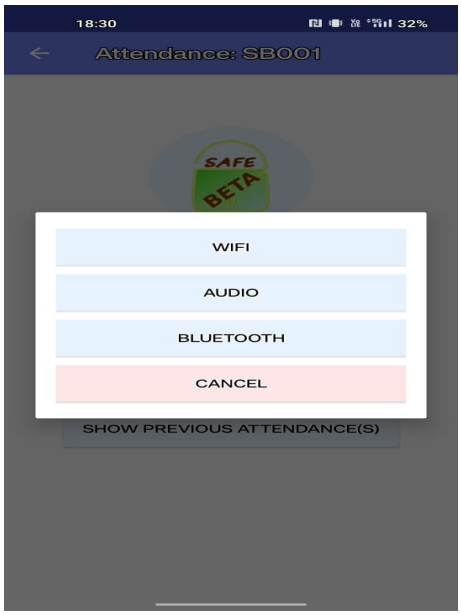


Figure 1: New "Mark Attendance" popup screen after UI changes

3.1.2 Now:

- Disabled the automatic redirection, and now when the popup appears, it remains on the screen until the user manually selects either Audio, WiFi, Bluetooth or Cancel
- The updated version shows a popup - **"Do you want Audio based or Wifi based or Bluetooth based Attendance?" with Audio, Wifi, Bluetooth buttons** when "Mark Attendance" is clicked
- The user can now manually select the method to mark attendance. The popup stays on the screen until the user makes a choice and manually presses any of the buttons

3.2 Display Wait Time During Scanning

- In the original version of the Bluetooth Attendance feature, selecting the option would trigger a popup stating that scanning would continue for 24 seconds, along with a loading symbol. However, the user had no way to track the remaining wait time
- Now, a live countdown timer is displayed during the scanning phase, so the user can clearly see how much time is left. After scanning is complete, an additional popup appears, reminding the user not to turn off Bluetooth for the next 10 minutes

3.2.1 Before

The Bluetooth Attendance feature showed a popup with a static message:
"Scanning will run for 24 seconds"

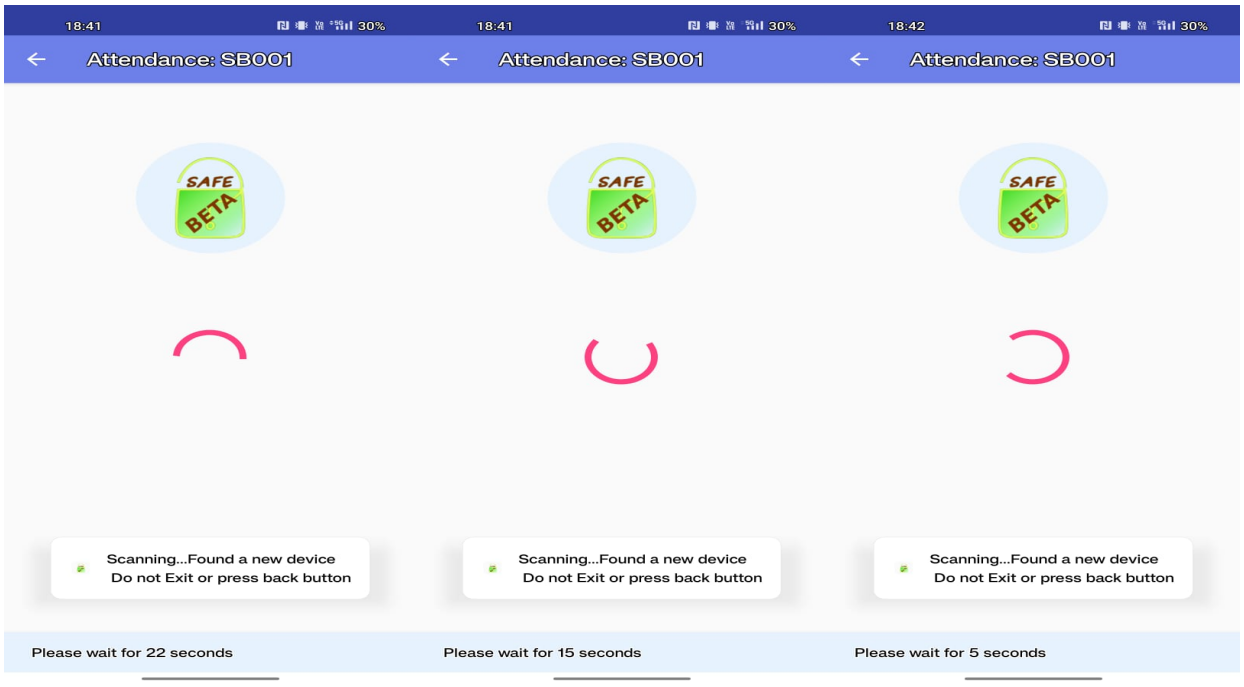


Figure 2: Live countdown timer displayed on screen, informing the user of the wait time

3.2.2 Now

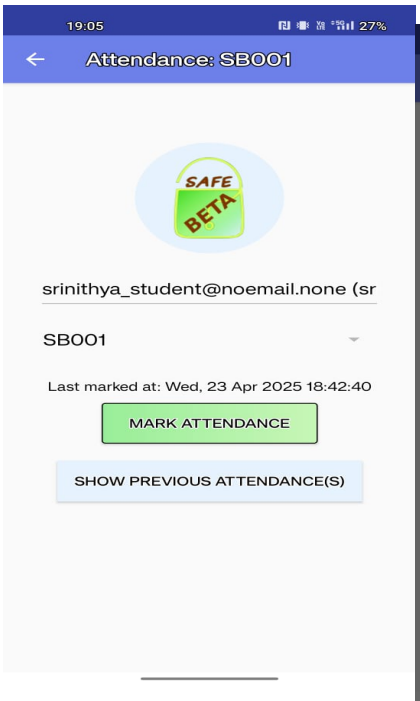
The **Live countdown timer** is shown **on the screen**, letting users know how much time is left and helps them track the wait time, keeping the experience smooth and predictable.

3.3 Successful Integration Without Issues

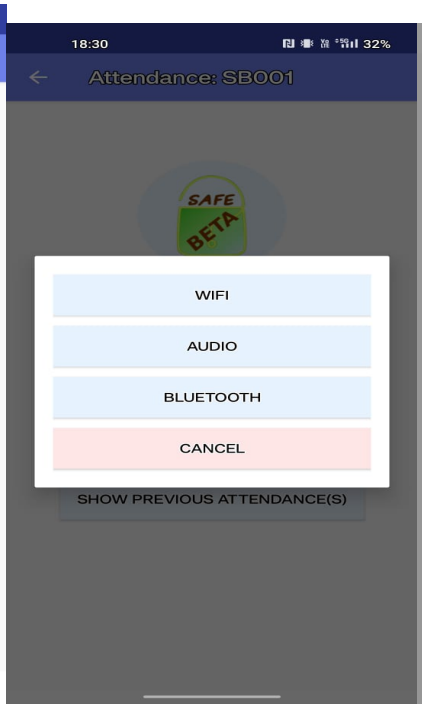
The **addition of the Bluetooth attendance option** has been successfully **integrated** without affecting the existing functionality of the app. **Procedure to Mark Attendance via Bluetooth:**

- Open the app and navigate to the attendance screen.
- Tap on the "MARK ATTENDANCE" button to begin the process.
- Select the "BLUETOOTH" option when prompted for the method to mark attendance.
- Grant necessary permissions (if prompted) by clicking on "Allow".
- Wait for the system to process your Bluetooth connection and countdown.
- After the countdown, follow the instructions to take your selfie, complete attendance.

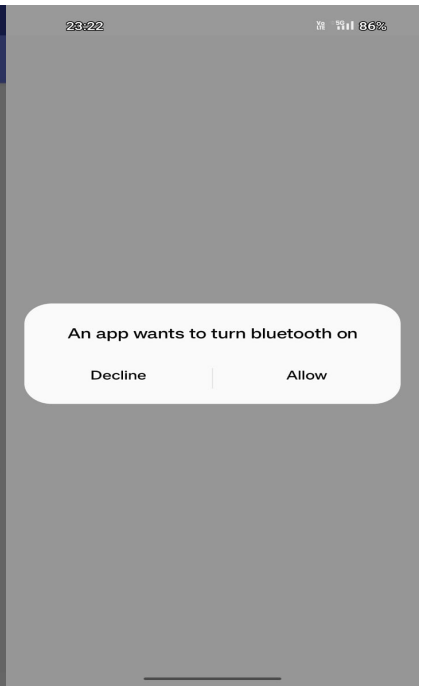
3.3.1 Screenshots taken while using the app to mark bluetooth attendance after updates in UI:



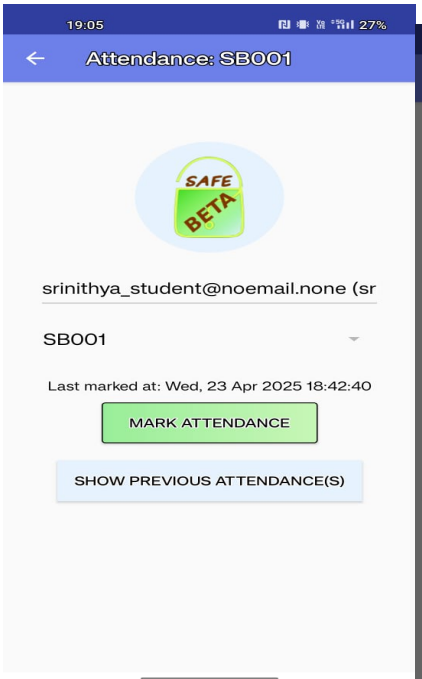
tap **"MARK ATTENDANCE"**



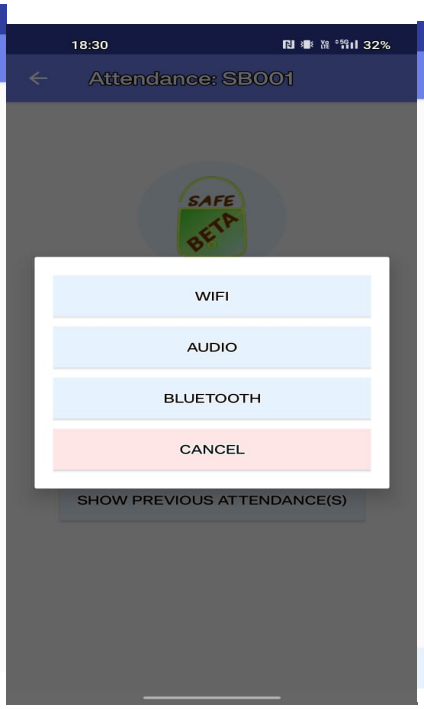
click on **"BLUETOOTH"**



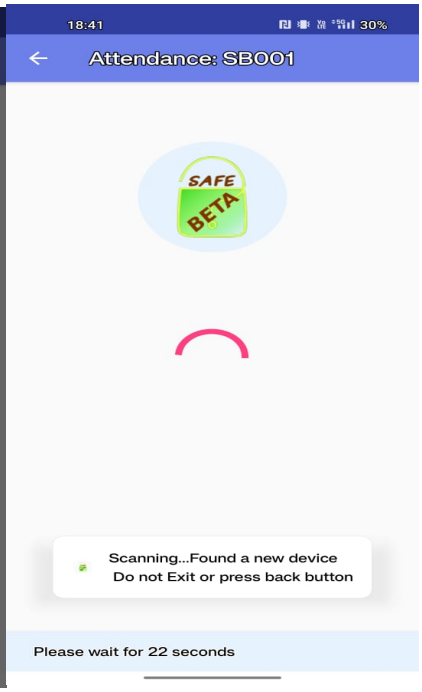
click on **"Allow"**



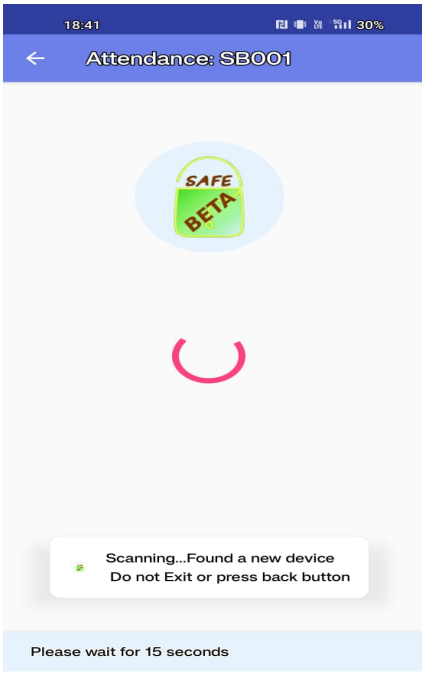
click on **"MARK ATTENDANCE"**



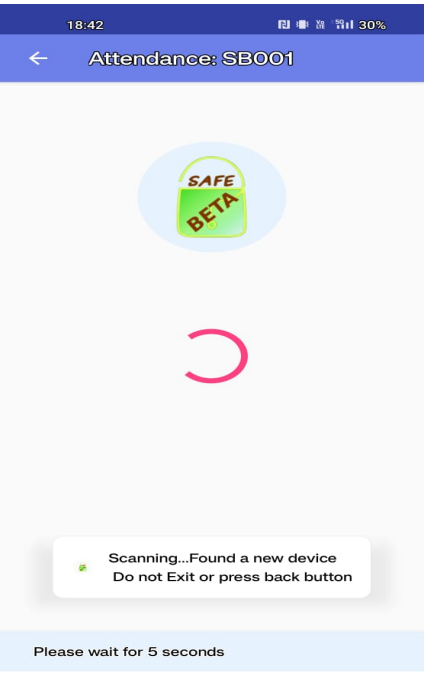
click on **"BLUETOOTH"**



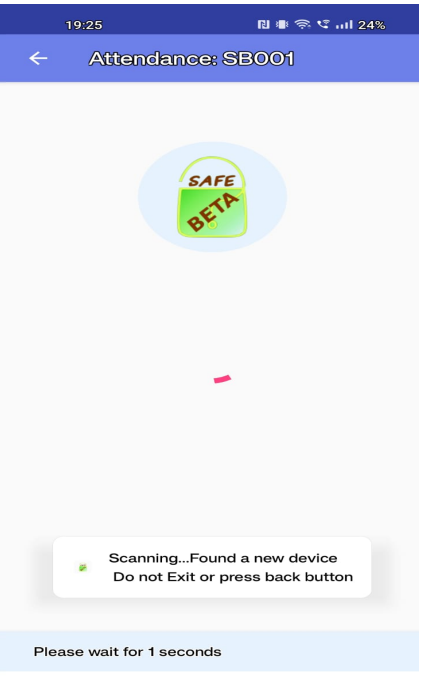
please wait for 22 seconds



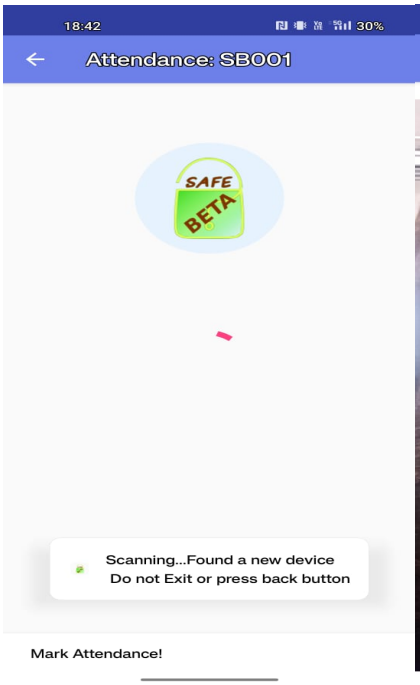
please wait for 15 seconds



please wait for 5 seconds

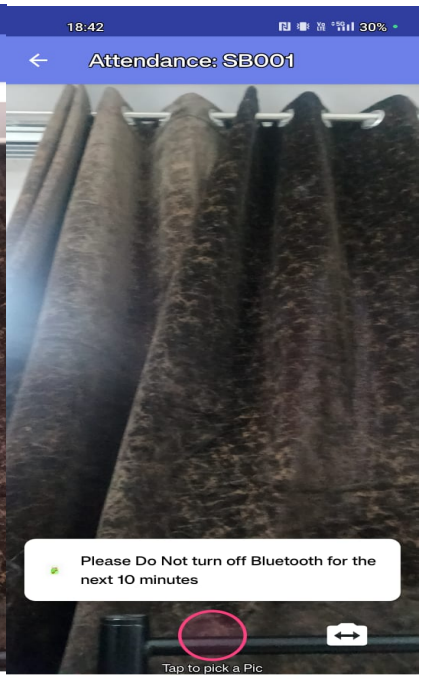


please wait for 1 second



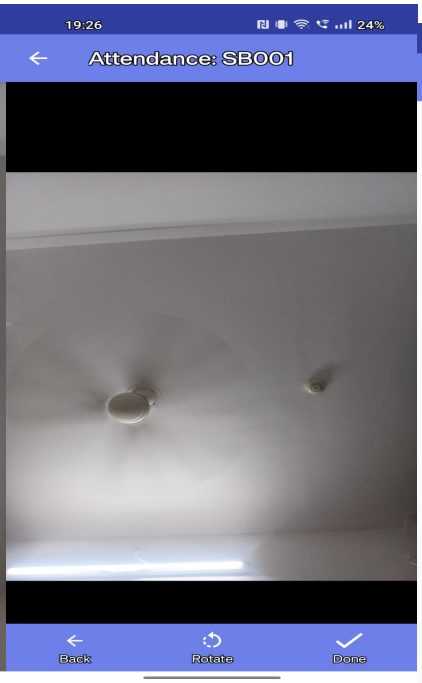
can mark attendance now

bluetooth scanning done

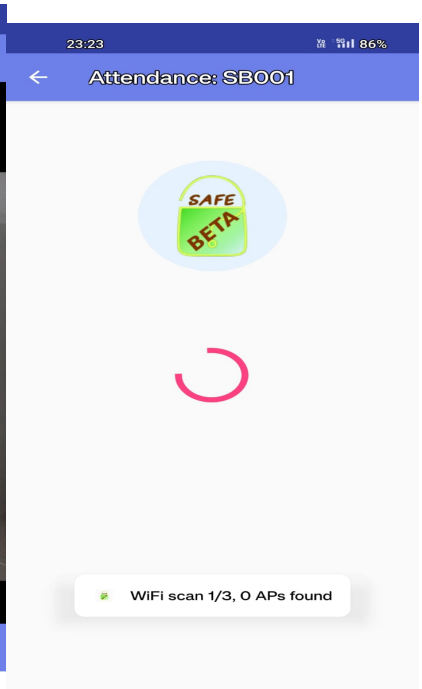


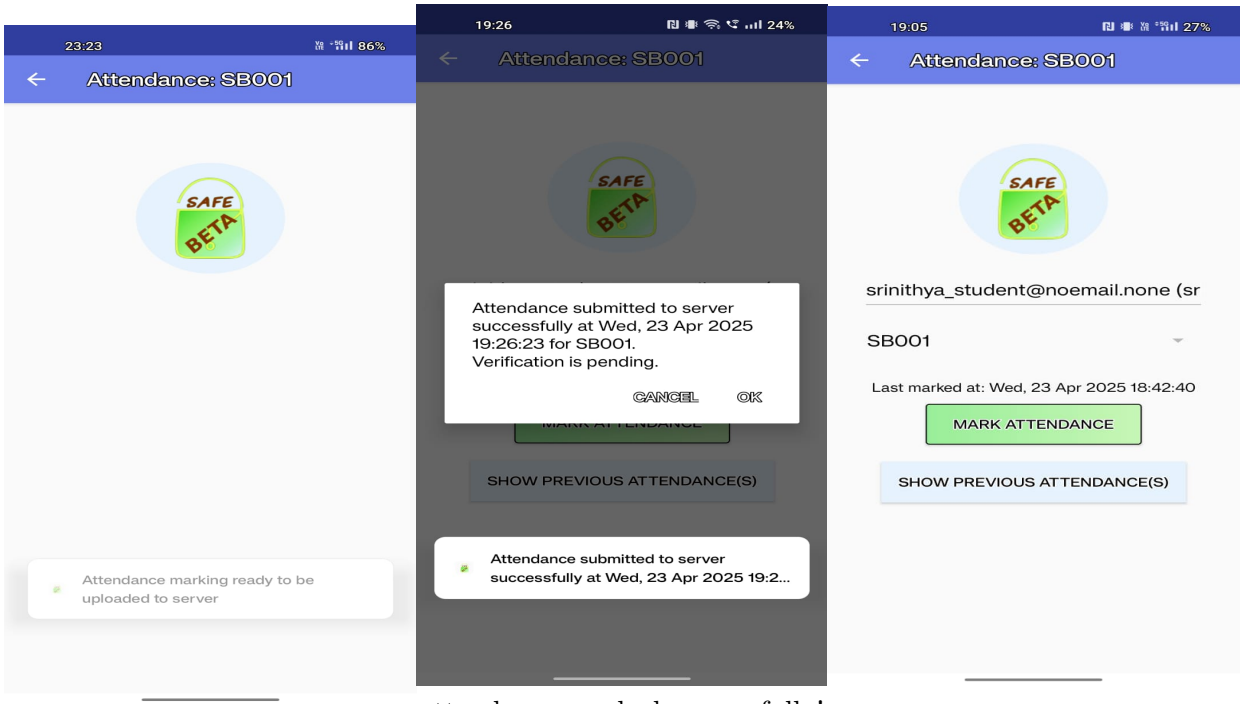
don't turn off bluetooth

click a **selfie**



click on "done"





attendance marked successfully! back to markattendance screen

4 Backend Processing for Attendance Validation

4.1 Current Attendance System Design and Methodology

Data Collection from Student Devices

After students mark their attendance, their devices send data to the server.
That data includes:

- Devices with a list of nearby Bluetooth devices detected during scanning
- RSSI values (Received Signal Strength Indicator), which indicates the signal strength of nearby devices

By analyzing this data, the server determines a student's prescence in the class.

Principles Behind the Current Attendance Algorithm

- A stronger RSSI value suggests that a detected device is physically closer. Only devices within close range are considered for forming edges in the graph
- Formation of Connected Components: Students who are physically near each other in the classroom will form a connected graph component
- Identifying Students present in the class: The largest connected component in the graph represents the actual group of students present inside the classroom

Present Attendance Validation Algorithm

- Constructing a graph representation of detected devices and applying Depth-First Search (DFS) to identify Connected Components.
- The largest connected component is considered as the group of present users, while others are marked absent.

4.2 Stress Testing and App Stability

Bug Reported in Previous Work: The app crashes when a user opens the app but has not yet marked attendance. This is due to an attempt to access a non-existent key in the adjacency list during a loop, without checking if it exists.
Solution: Add the missing users to the adjacency list before proceeding with loop operations. The following code snippet addresses the issue:

```
missing_keys: dict[str, set[str]] = {}

for user, neighbors in adj_list.items():
    for adj_user in neighbors:
        if adj_user not in adj_list:
            missing_keys.setdefault(adj_user, set()).add(user)

adj_list.update(missing_keys)
```

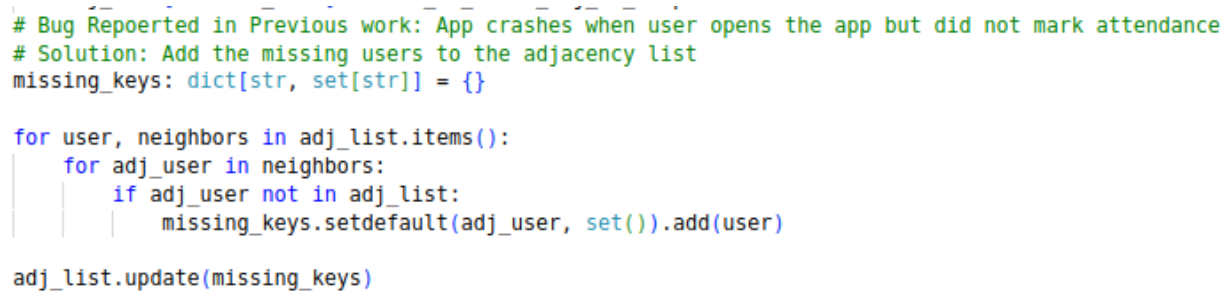


Figure 3: Crash location during attendance access loop

4.3 Scope for Algorithm Improvement

4.3.1 Re-evaluating the Signal Strength Threshold (-55 dBm)

A fixed signal strength threshold of **-55 dBm** is currently used to ignore distant Bluetooth devices. However, this static value may not suit all classroom settings.

We suspect that the threshold might be either too strict—excluding nearby valid devices—or too lenient—capturing signals from far-off devices. To address this, we propose an empirical study measuring RSSI values at different distances to determine a more adaptive, environment-specific threshold.

Algorithm 1: Attendance Marking via Bluetooth Proximity (initial algorithm)

Input: List of devices, each with discovered Bluetooth devices and their RSSI values
Output: Each device is assigned a flag: 'F' (Present) or 'T' (Absent)
Initialize:
visited_devices $\leftarrow \emptyset$ ▷ Tracks checked devices
graph_components $\leftarrow \emptyset$ ▷ Stores connected components
proximity_map $\leftarrow \emptyset$ ▷ Stores adjacency relationships
Construct Adjacency List: for *each device D in device list* **do**
    each detected device D' in scan results of D **if** $RSSI(D') > -55$ **then**
    High signal strength indicates proximity
proximity_map$[D] \leftarrow$ proximity_map$[D] \cup \{D'\}$

Define DFS Function:
function DFS$(D, component)$
    visited_devices $\leftarrow$ visited_devices $\cup \{D\}$
    component $\leftarrow$ component $\cup \{D\}$ **for** *each D' in proximity_map*$[D]$ **do**
    $D' \notin$ visited_devices
    DFS$(D', component)$

end function
Identify Connected Components: for *each device D in device list* **do**
    $D \notin$ visited_devices
component $\leftarrow \emptyset$
DFS$(D, component)$
graph_components $\leftarrow$ graph_components $\cup \{component\}$

Find Largest Connected Component (LCC):
$LCC \leftarrow \arg\max_{component \in \text{graph\_components}} |component|$
Assign Attendance Status: for *each device D in device list* **do**
    $D \in LCC$
Mark D as 'F' (Present) **else**
    Mark D as 'T' (Absent)

4.3.2 Device Identification Using MAC Addresses

In some classroom layouts, students seated in clusters may form disconnected components (or "islands") in the Bluetooth graph. To address this, we propose using the MAC address of devices along with the user-defined device name.

- Each student's device is renamed in the format: `SAFE_rollNumber_SAFE`.
- Although Google restricts access to a device's own MAC address for privacy reasons, MAC addresses of surrounding devices are still accessible.
- For each node in the graph, we associate both the MAC address and the renamed device identifier (if available).
- If a device has the name `SAFE_rollNumber_SAFE`, it is considered a student's device (SAFE device), and its name is used as an identifier.
- Devices without this special name are treated as non-SAFE devices (extra devices), but may still serve as bridges to connect otherwise isolated student clusters; in such cases, their MAC address is used as the identifier.

5 Experiments

5.1 RSSI threshold

Assumptions

For all experiments, we are assuming a standard classroom layout:

- The server (Bluetooth scanner) is placed near the blackboard
- Students sit in rows starting from the blackboard towards the back of the classroom
- The only entry/exit door is located at the back (farthest from the server)
- This ensures that RSSI values directly reflect physical proximity within the classroom

Purpose

To determine an appropriate RSSI threshold for attendance marking based on classroom size

Methodology

To measure RSSI value variations with distance, we took help of Bluetooth RSSI Meter app. This app allowed real-time scanning of nearby Bluetooth devices, displaying the corresponding RSSI values.

Experiment 1: RSSI Values Recorded at Different Distances

Distance (meters)	RSSI values (dBm)
≤ 1	-24
5	-46
10	-53
25	-64
50	-68
60	-84
70	-78
80	-69
80–100	-78

Table 2: Experiment 1: RSSI Values at Different Distances

Experiment 2: RSSI Values at Different Distances with Large Obstacles

Scenario	Distance (meters)	RSSI (dBm)
Case 0: Open Space	0.1	-42
Case 1: Door Open	0.1 (adjacent to wall)	-44
Case 2: Door Closed	0.1 (adjacent to wall)	-47
Case 0: Open Space	5	-50
Case 1: Door Open	5 (along the wall)	-51
Case 2: Door Closed	5 (along the wall)	-59

Table 3: Experiment 2: RSSI Values at Different Distances with Large Obstacles

Observations

- Devices are being detected till to a distance of 180m
- RSSI value is getting weaken as the distance increases, there are fluctuations beyond 50 meters
- RSSI values are not always decreasing linearly, there are unexpected variations occuring which may be due to factors like reflections and interference
- Large obstacles like closed doors are significantly reducing RSSI strength compared to open spaces

Conclusions

- **The existing code uses a threshold of -55 dBm. This may be too restrictive or lenient depending on the room size**.
  Based on observations, suggested RSSI thresholds are:

ClassType	Suggested RSSI Threshold
CC	-55
LC	-65
LHC	-80
LA	-96

Table 4: Conclusion from Experiments 1.1 & 1.2: RSSI Threshold vs Class Type

Problem

- The experiments above were based on a standard classroom layout where the entry door is at the back. However, in real classrooms, doors are often located near the blackboard or along the side walls. This means that these doors can be closer to the server than the last row of students.
- Relying only on the RSSI can lead to proxies. Because a student standing just outside the classroom near the door may have a stronger RSSI than a student sitting inside in the last row. Even if the side doors are closed, walls and closed doors aren't weakening the signal enough, so the RSSI from outside the class behing closed door can still be high.
- To handle this issue, the existing algorithm treats each device as a node and connects nearby nodes with edges. It then forms components of connected students. Only students who are part of the largest component (the main group inside the class) are marked present and not just those with strong signals.

5.2 Device Identification for Attendance Marking

Objective

To ensure that attendance is marked on the server under the correct student name, regardless of the device used.

Methodology

- Devices are dynamically renamed to the format `SAFE_studentRollNumber_SAFE` when the student clicks the "Mark Attendance" button.
- Only devices with names in this format are recognized as valid student nodes in the system.
- Several experiments were conducted to evaluate the effectiveness of the renaming mechanism and its consistency across devices.

Current Implementation

Currently, devices renamed in the format `SAFE_studentRollNumber_SAFE` are the only ones considered for node creation and component building. The renaming occurs when the student

clicks the "Mark Attendance" button in the app, and the renamed format remains active only while the app is open.

Experiment: Single Device Renaming Validation

- Student `studentRollNumber1` used Device1 (MAC1).
- The device was successfully renamed to `SAFE_studentRollNumber1_SAFE`.
- The device node was correctly integrated into the system and appeared in the largest component, allowing the student to mark their attendance successfully.

Experiment: Cross-Device Renaming Consistency Check

- The same student `studentRollNumber1` switched to a different device (Device2 with MAC2).
- The new device was also successfully renamed to `SAFE_studentRollNumber1_SAFE`.
- Attendance was marked correctly, confirming that students can switch devices and still mark attendance under their correct identity.

Device Switching and Attendance Marking

Students can switch devices, and the system will continue to support flexible attendance marking without any issues.

- No inconsistencies, delays, or issues in attendance marking were observed after renaming the device, confirming the reliability of the system.

Conclusion

- The dynamic renaming mechanism works effectively, even when students change devices. However, this flexibility introduces a risk: a single device could be renamed multiple times to mark attendance for multiple students.
- To mitigate this risk, the system should enforce a "one device, one attendance per session" rule. No MAC address should be reused for marking attendance for multiple students in the same session, even though different devices can be used by the same student. This would help prevent proxy attendance while preserving user flexibility.

Proposal for Improvement

It is proposed to use all detected devices based on their MAC addresses, rather than just the renamed devices (SAFE devices). This would ensure that every nearby device, regardless of whether it is a SAFE device, is treated as a node in the system.

6 Device Identification Strategy

6.1 Device Identification Models

For identifying and using Bluetooth devices as nodes:

- **Model 1:** Use only devices renamed in the format `SAFE_studentRollNumber_SAFE`.
- **Model 2:** Use all detected devices, identified by their unique MAC addresses.

Motivation for the Shift to Model 2

Concern	SAFE_studentRollNumber_SAFE	MAC Address
Proxy Attendance	A single student can mark attendance for friends by logging in to the respective SAFE accounts. One device can mark attendance for multiple students without them actually being present.	Each device is uniquely identified using its MAC address, preventing the same device from being used for multiple students. Ensures one student per device.
Scattered Seating Issues	Students sitting in small and separate groups may form disconnected clusters (islands) and get excluded from the main component.	Additional detected devices can act as bridges, helping connect isolated student groups into a single large component.
Noise Control	Only SAFE student devices are used, reducing interference from unrelated devices.	Some unrelated devices may be included, introducing noise, but this trade-off may improve connectivity.

Table 5: Comparison of Model 1 and Model 2

6.2 Concerns

- **Proxy Attendance via a Single Device**
  The current attendance system identifies students based on their renamed device names. This allows a student to log in from the same device using multiple SAFE accounts and mark attendance for themselves and their friends. Since the system treats each device name as a distinct entity, it mistakenly registers multiple students as present, even if only one device is physically in the classroom.
- **Proxy Attendance via Same Device**
  A single student can log into different SAFE accounts from the same device and mark attendance for multiple users, enabling proxy attendance.
- **Proxy Possibility Under Model 1** Attendance was marked for two users using the same device. The system did not detect the proxy, confirming the vulnerability.

Preventing Proxy via

- **Model 1:** Additionally, in Model 1, a modified approach was tested: tracking MAC addresses and disallowing duplicate usage within a session.
- **Model 2:**
  When MAC addresses are used for device identification, each device is uniquely recognized, preventing it from being reused across multiple student accounts. This ensures a one-student-per-device mapping and helps prevent proxy attempts. The system should flag and block any second attendance attempt from the same MAC address within the same session.

Impact of Seating Patterns and Inclusion of Non-Student Devices

Concern: Scattered Seating and Disconnected Clusters

When students are seated in small, dispersed clusters across a classroom, their devices may fail to form sufficient Bluetooth connections (edges) due to the physical distance between them. This can result in isolated nodes or disconnected subgraphs, causing some students to be excluded from the largest connected component and hence, incorrectly marked absent.

Inclusion of Additional Nearby Devices

To mitigate this issue, we propose including additional nearby devices — even if they are not registered SAFE devices — to act as bridging nodes. These devices help form a more connected graph, increasing the likelihood that all student devices become part of the main connected component.

Impact on Attendance Accuracy

We evaluate whether including all detected devices using their MAC addresses improves or deteriorates the accuracy of attendance marking. Specifically, we compare this approach (Model 2) with one that considers only verified SAFE-named student devices (Model 1).

Experiments to be Conducted

The following experiments will be conducted to analyze how varying classroom sizes affect connectivity and attendance marking under different configurations:

- **Classroom Size Variations:** Experiments will be conducted with different classroom sizes—40, 60, 80, and 100 students. For each size, seating orientations and densities will be adjusted to simulate realistic classroom conditions. The primary objective is to assess how the size of the largest connected component changes across different models and whether proxy prevention and accurate attendance marking are effectively maintained.

- **Model 1: Only SAFE-Named Devices:** In this model, only devices identified as SAFE-named student devices will be included. This serves as a control experiment to evaluate the trade-offs between attendance accuracy and connectivity. The size of the largest connected component will be recorded for each configuration.
- **Model 2: All Devices Included:** For this model, all detected devices (identified by MAC addresses) will be included in the experiment to enhance connectivity. Similar to Model 1, the size of the largest connected component will be recorded for each configuration.

7 Scenarios Where Attendance Verification May Fail

7.1 Attendance Marking with Overlapping Timestamps

This case examines the attendance system when two students sitting close to each other mark their attendance sequentially with a short time gap. The goal is to determine whether the time gap affects verification and whether both students remain in the same connected component.
Two students mark their attendance one after another with a specific time gap.

S.No	Time Gap	Student 1	Student 2
1	$<$ 24 seconds	VERIFIED	VERIFIED
2	$>$ 24 seconds	VERIFIED	VERIFIED

Table 6: Case 1: Attendance verification based on time gap

```
[2025-04-02 18:24:14,891: WARNING/ForkPoolWorker-2] These are the components of the undirected graph
[2025-04-02 18:24:14,891: WARNING/ForkPoolWorker-2] [['bhavana_student@nomail.none', 'srinithya_student@noemail.none']]
[2025-04-02 18:24:14,891: WARNING/ForkPoolWorker-2]
This is maximum connected component in our graph
[2025-04-02 18:24:14,891: WARNING/ForkPoolWorker-2] ['bhavana_student@nomail.none', 'srinithya_student@noemail.none']
```

```
[2025-04-02 18:42:56,407: WARNING/ForkPoolWorker-2] These are the components of the undirected graph
[2025-04-02 18:42:56,407: WARNING/ForkPoolWorker-2] [['bhavana_student@nomail.none', 'srinithya_student@noemail.none']]
[2025-04-02 18:42:56,407: WARNING/ForkPoolWorker-2]
This is maximum connected component in our graph
[2025-04-02 18:42:56,408: WARNING/ForkPoolWorker-2] ['bhavana_student@nomail.none', 'srinithya_student@noemail.none']
```

Figure 5: Attendance marking with a time gap of more than 24 seconds

Observations:

- Regardless of whether the time gap is less or more than 24 seconds, an edge forms between both students, ensuring that they remain in the same connected component and their attendance is successfully marked.

7.2 Attendance Failure Due to Sequential App Usage

This case examines the attendance system's failure when two students mark attendance sequentially, but one closes the app before the other attempts to mark their presence.

Scenario:

Two students are sitting close to each other and mark their attendance one after another. However, Student 1 submits their attendance and closes the app before Student 2 opens it.

Problem:

- When Student 1 marks attendance, their device sends data to the server.
- At that moment, Student 2's device is not detected.
- After marking attendance, Student 1 closes the app.
- Later, Student 2 opens the app and marks attendance.
- Since Student 1 is no longer in the app, they cannot detect Student 2's presence.

Consequence:

- Student 2 is not included in the maximum component and is marked absent.

S.No	Student 1 Status	Student 2 Status	Attendance Outcome
1	App Open	App Open	Both Verified
2	App Closed	App Open	Student 2 Marked Absent

Table 7: Case 2: Attendance failure due to sequential app usage

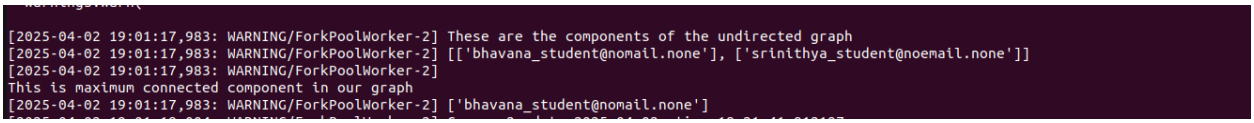


Figure 6: Student 1 marks attendance while Student 2's app is closed

Observations:

- If both students keep their apps open during attendance marking, both are verified.
- If Student 1 closes the app before Student 2 marks attendance, Student 2 remains undetected and is marked absent.

Algorithm 2: Attendance Marking via Bluetooth Proximity (Current Algorithm)

Input: List of devices with discovered Bluetooth devices and RSSI values
Output: Each device is assigned a flag: 'F' (Present) or 'T' (Absent)
Initialize:
visited_devices $\leftarrow \emptyset$ ▷ Tracks checked devices
graph_components $\leftarrow \emptyset$ ▷ Stores connected components
proximity_map $\leftarrow \emptyset$ ▷ Stores adjacency relationships
mac_address_to_user $\leftarrow \{\}$ ▷ Tracks users per MAC
user_to_mac_address $\leftarrow \{\}$ ▷ Tracks user to MAC
Construct Adjacency List:
for *each user* U *in device list* **do**
each (MAC, info) in scan results of U
user_name $\leftarrow$ info["user"], rssi $\leftarrow$ info["rssi"] **if** *user_name* $\neq \emptyset$ ***and*** *starts with "SAFE"* ***and*** *user_name[4:]* $\in$ *marked_users* **then**
mac_address_to_user[MAC] $\leftarrow$ mac_address_to_user[MAC] $\cup$ {user_name[4:]}
user_to_mac_address[user_name[4:]] $\leftarrow$ MAC **if** $RSSI > -55$ **then**
adj_list[U] $\leftarrow$ adj_list[U] $\cup$ {user_name[4:], MAC}
else
RSSI > -55
adj_list[U] $\leftarrow$ adj_list[U] $\cup$ {MAC}
Add missing keys to the adjacency list:
DFS Function:
DFSD, *component*
visited_devices $\leftarrow$ visited_devices $\cup \{D\}$
component $\leftarrow$ component $\cup \{D\}$ **for** *each* D' *in proximity_map*[D] **do**
$D' \notin$ visited_devices
DFSD', *component*
Identify Connected Components: for *each device* D *in device list* **do**
$D \notin$ visited_devices
component $\leftarrow \emptyset$
DFSD, *component*
graph_components $\leftarrow$ graph_components $\cup \{component\}$
Find Largest Connected Component (LCC):
$LCC \leftarrow \arg\max_{component \in \text{graph\_components}} |component|$
Assign Attendance Status: for *each device* D *in device list* **do**
$D \in LCC$ **and** mac_address_to_user[D].size < 1
Mark D as 'F' (Present) **else**
Mark D as 'T' (Absent)

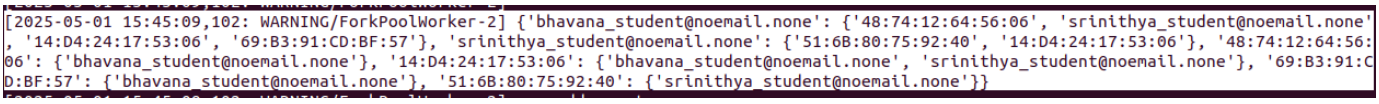
8 Field Evaluation

System Setup for Attendance Marking

For attendance marking via Bluetooth proximity, we implemented an algorithm that identifies each device using a unique "SAFE_rollnumber_SAFE" identifier and its corresponding MAC address, which serves as the node identity. Extra devices, such as those used for bridging isolated components, are also treated as nodes within the system. This setup ensures that previously isolated devices can be connected, enabling more reliable attendance tracking. The following algorithm, 2, outlines the process we employed:

Proximity-Based Adjacency Mapping

The algorithm constructs an adjacency list representing Bluetooth proximity connections between devices. Each key in the adjacency list corresponds to a device (either identified by email or MAC address), and its associated set contains the devices it was in close proximity with (as determined by signal strength thresholds).
The structure of the adjacency list is shown below:



An excerpt of the raw log representation:

```
{'bhavana_student@noemail.none': {'48:74:12:64:56:06', 'srinithya_student@noemail.none',
                                  '14:D4:24:17:53:06', '69:B3:91:CD:BF:57'},
 'srinithya_student@noemail.none': {'51:6B:80:75:92:40', '14:D4:24:17:53:06'},
 '48:74:12:64:56:06': {'bhavana_student@noemail.none'},
 '14:D4:24:17:53:06': {'bhavana_student@noemail.none', 'srinithya_student@noemail.none'}
 '69:B3:91:CD:BF:57': {'bhavana_student@noemail.none'},
 '51:6B:80:75:92:40': {'srinithya_student@noemail.none'}}
```

'bhavana_student@noemail.none' is directly connected (in proximity) to the following devices:
- 'srinithya_student@noemail.none'
- MAC address '14:D4:24:17:53:06'
- MAC address '69:B3:91:CD:BF:57'
- MAC address '48:74:12:64:56:06'

These connections are used by the algorithm to build a proximity graph, which helps identify connected components for reliable attendance inference.

Issue Encountered:

Some students marked attendance without updating the app, and instead, attendance was recorded using Wi-Fi, as the algorithm had crashed. Due to this, some fields were missing in the old version, causing errors. Consequently, after the issue was resolved, the remaining students' attendance was not recorded.

9 Limitations and Mitigation Strategies

Limitations in Preventing Proxy Attendance

- In LA rooms with staircase layouts, students can position themselves beneath the seats and still mark attendance, as Bluetooth (BT) waves can pass through thin walls with minimal power loss, enabling proxy attendance.
- In classrooms like the LHC, students can stand behind the server, away from the classroom, and still mark attendance. Stricter RSSI thresholds may block distant students, but they can also mistakenly prevent legitimate students inside the classroom from being recognized.

Strategies to Reduce Proxy Attendance Risk

- Encouraging students to sit close together in a single group can help avoid isolated clusters or "islands," improving the reliability of attendance detection.
- While preventing all proxy attendance may not be entirely feasible, reducing its likelihood remains important. Despite the limitations of Bluetooth, the system aims to minimize opportunities for misuse as much as possible.

10 Future Work

10.1 Threshold RSSI Value

The threshold RSSI value for the edge between two devices is currently set to -55. However, this value fluctuates based on the distance and environment. Therefore, further testing is required to determine the most appropriate threshold. We suggest using a value of -75, which could provide more stable results across varying distances and environments.

10.2 Attendance Window

Currently, a single graph is constructed for all attendance windows. This approach does not separate the attendance data based on different time windows. We propose modifying the current system to construct individual graphs for each attendance window, which will allow for more accurate analysis of attendance patterns over time.

10.3 In-Class Experiments

The current implementation has been tested only on a small set of devices. Further experimentation is needed in a real classroom setting to validate the approach at scale and to assess its effectiveness in live scenarios.

11 Code

11.1 Server

Branch: safe_server-nithya-bhavana-bluetoothAttendance-changes
Final Commit ID: 7c5a1a1db86c8b457d16bacaad6537a75059e22e
Merged Commit ID with safe v2 server: b257168485fca5ff36dc0521f7b55899280b4e49

11.2 Safe App

Branch: safe-nithya-bhavana-ui-changes
Final Commit ID: b76e9523e549efcb425a6a8e11547a19b1ad0593
Merged Commit ID with safe app: b76e9523e549efcb425a6a8e11547a19b1ad0593 (same as above)