REPUBLIQUE DU CAMEROUN	REPUBLIC OF CAMEROON
PAIX – TRAVAIL – PATRIE	PEACE – WORK – FATHERLAND
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE	MINISTRY OF HIGHER EDUCATION

FACULTY OF ENGINEERING AND TECHNOLOGY

PO BOX 63

BUEA, SOUTH WEST REGION

DEPARTMENT OF COMPUTER ENGINEERING

COURSE CODE: CEF440

COURSE TITLE: INTERNET PROGRAMMING AND MOBILE DEVELOPMENT

COURSE INSTRUCTOR: Dr. NKEMENI VALERY

TASK 2 REPORT

Presented by:

NAMES	MATRICLE
1. MEKOLLE ASHLEY ARREYMANYOR	**FE22A247**
2. AKO RUTH ACHERE	**FE22A144**
3. NGUIMENANG ZEUFACK KEREINE	**FE22A266**
4. TUMUTOH LYDIE-KASEY BIHNUI	**FE22A321**
5. TIWA DELAN NDIKA	**FE22A319**

APRIL 2025

Table of Contents

Abstract

This report documents the development of a Mobile-Based Attendance Management System using **Facial Recognition and Geofencing,** case study at the **University of Buea, Cameroon**. It outlines the **stakeholder identification process**, **requirement gathering techniques** (including surveys and interviews), **data collection and cleaning methods**, and **user reluctance assessment and mitigation ways.** based on feedback from 39 students and faculty.

Key findings reveal:

- 68% of users rely on manual attendance methods prone to errors.
- 51% have been wrongly marked absent due to current system flaws
- Privacy concerns (55%) and technical barriers (Internet availability, 90%) are major adoption hurdles
- Critical features demanded: real-time notifications (78%), LMS integration (59%)

The report provides a blueprint for addressing these challenges, also verifying if this system is feasible on the real world, case study the University of Buea.

1. Introduction

1.1 Background

Traditional attendance systems in academic institutions face critical challenges such as:

- **Human errors**: 51% wrongful absences reported
- **Cheating:** 80% of lecturers that use manual paper sheets faces **impersonification** where other students sign in for their friends that are absent
- **Time inefficiency**: More than 10 - 15 minutes/lecture wasted)
- **Lack of analytics**: Administrators and Lecturers desire a final reporting of students

This project proposes an automated solution combining:

- **Facial recognition** for identity verification
- **Geofencing** to validate physical presence
- **Mobile integration** for accessibility

1.2 Objectives

1. Identify stakeholders and their needs
2. Gather requirements through empirical data (surveys, interviews)
3. Ideate system specifications based on gathered data
4. Analyze user reluctance and mitigation strategies

2. Stake Holder Identification

The main stake holders in this system are:

1. Students

Role: Primary system users
Key Details:

- ➢ Use the app to log in their attendance via facial recognition and geofencing
- ➢ Receive attendance confirmations and alerts
- ➢ Can dispute errors through the app

Pain Points: Privacy concerns, Accuracy concerns in poor lighting and other technical conditions

2. Lecturers

Role: Attendance validators
Key Details:

- ➢ Initiate attendance sessions in class
- ➢ Verify flagged cases such as student recognition failures
- ➢ Access real-time report

Pain Points: Time wasted on manual processes, Attendance cheating by students, Overall Performance of their students results.

3. IT Administrators

Role: System maintainers
Key Details:

- ➢ Manage device
- ➢ Troubleshoot technical issues
- ➢ Ensure data security

Pain Points: Compatibility with existing campus systems

4. Academic Administrators

Role: Decision-makers
Key Details:

- Set attendance policies geofence boundaries for each class venue,
- Approve system-wide adoption

Pain Points: Meeting regulatory requirements

3. Requirement Gathering

After identifying stakeholders, the next thing was to get the requirements from them and the following method was adopted.

- **Survey Forms**: Survey forms were created via **Google Forms** and shared to students and lecturers with about Thirty-nine (39) responses received from students and One (1) response from lecturers according to Google Form's Response sheet.

- **User Interviews:** This was conducted on both students and Lecturers where Three (3) students and Lecturers from the Faculty of Engineering and Technology were interviewed. Only a single lecturer who is as well an **administrator** was interviewed.

- **Brainstorming**: Research was carried out by some team members on the technology to be used in the Design and Implementation of the Front-End and Back-End of this system. Other requirements were brought forth from the analysis made on the survey and Interview after understanding the goals and fraustrations of the Stake holders, applying logic to ensure solutions

- **Competitive Analysis and SWOT analysis**: Other geofence-based and facial recognition-based attendance management softwares were reviewed.

3.1. Competitive Analysis

A competitive analysis, also called competitor analysis and competition analysis, is the process of examining similar brands in your industry to gain insight into their offerings, branding, sales, and marketing approaches. Knowing your competitors in business analysis is important if you're a business owner, marketer, start-up founder, or product developer.

3.1.1. Benefits of conducting competitor analysis

A competitor analysis offers several benefits, including:

- Understanding industry standards so that you can meet and exceed them
- Differentiating products and services
- Fulfilling customers' desires and solving their problems better than competitors
- Distinguishing your brand
- Standing out in your marketing
- Measuring your growth

3.1.2. Competitive analysis of direct and indirect apps

App Name	Unique Features	Strengths	Weaknesses	Platform	References
TrackCC	- QR code-based self check-in - Customizable attendance types - Real-time data sharing with students and guardians	- User-friendly interface - Comprehensive features (attendance, behavior, grading) - Offline functionality with sync capabilities	- Limited to 10 students in free version - No facial recognition or geofencing features - Basic reporting tools	Mobile App & Website	TrackCC
MyAttendanceTracker (MyAT)	- Integrated gradebook - Parent messaging system	- Free to use - Simple and intuitive design	- Expensive for small schools -Limited automation	Mobile App & Website	MyAttendanceTracker

App Name	Unique Features	Strengths	Weaknesses	Platform	References
	- Customizable attendance reporting	- Suitable for various class or group settings	- Basic data export options		
Palgeo	- Facial recognition attendance - Geofencing with beacon technology - CCTV integration for attendance	- Multi-tech integration (biometric, QR code, facial recognition) - Automated check-in/out with location tracking - High security with AI technology	- May require additional hardware for full functionality- Potential privacy concerns with CCTV integration - Complex setup process	Mobile App & Website	Palgeo
AttenFace	- AI-powered facial recognition - Real-time attendance tracking - Geofencing-based location verification	- High accuracy in identification - Automated notifications to parents - Secure data handling	- High battery consumption - Limited device compatibility - Steep learning curve	Mobile App	AttenFace

3.1.3. Observations & Insights

- **TrackCC** and **MyAT** excel in **institutional integration**, offering a full suite of class tools including gradebooks, messaging, and basic attendance tracking—ideal for schools that want an all-in-one platform.
- **Palgeo** offers the most **advanced geolocation and surveillance tools**, making it a top pick for larger institutions with security concerns. However, the reliance on hardware and CCTV can increase cost and complexity.
- **AttenFace** stands out for its **facial recognition and geofencing precision**, targeting modern, small-to-mid-sized organizations that prioritize automation and tech-forward solutions. However, it lacks a web dashboard, which may hinder admin functionality.
- While **MyAT** markets itself as a free solution, its **advanced anti-spoofing and customization** come at a cost for institutions wanting premium features, making it less affordable for smaller schools.

3.2. SWOT Analysis

SWOT stands for **Strengths, Weaknesses, Opportunities, and Threats**. It is a simple yet powerful tool used to assess a project or business idea from all angles. It helps identify what you do well, what needs improvement, where growth is possible, and what risks might affect your plans. By carrying out a SWOT analysis early in the project development, especially during the requirements gathering stage, you can create a clear roadmap and make more informed decisions.

In this analysis, we focus on a mobile attendance system that uses geofencing (location tracking) and facial recognition to ensure students are physically present in a classroom before their attendance is marked. This system is designed to solve problems like proxy attendance and manual check-ins, which are common challenges in educational institutions across Cameroon.

Below is a detailed SWOT analysis of the project to guide further development:

Strengths (Internal, Positive)	Weaknesses (Internal, Negative)
Combines geofencing and facial recognition for high-accuracy attendance tracking.	Requires strong internet connectivity and precise GPS accuracy to function properly.
Reduces administrative workload by automating attendance processes.	Facial recognition can struggle in low-light environments or with poor camera quality.

Prevents proxy attendance, enhancing integrity and trust in the system.	Some students may not own smartphones or may use incompatible devices.
Allows real-time data access and monitoring of students by instructors or admins.	Concerns over biometric data privacy and security must be addressed carefully.

Opportunities (External, Positive)	Threats (External, Negative)
Expand usage to corporate settings, conferences, and events beyond schools.	Legal restrictions and school policies might limit the use of facial recognition technology.
Integrate with school information systems for automatic syncing of attendance records.	Presence of competitor applications offering similar features in the market.
Use AI for continuous improvement in facial recognition accuracy over time.	GPS or battery failure can lead to unsuccessful check-ins, affecting reliability.
Implement alert systems for parents or staff for real-time attendance updates.	Risks such as lost or shared devices could allow misuse of the check-in system.

This SWOT analysis provides a clear overview of the strengths, weaknesses, opportunities, and threats associated with the mobile-based attendance management system. By identifying these factors early in the requirements gathering phase, we can better prepare for challenges, leverage strengths, and explore opportunities for future growth.

3.3. Comparing Implementation Options for Geofencing and Facial Recognition

3.3.1. Key System Requirements

The key requirements that will influence our backend choices:

1. **Fast Facial Recognition** - Process and verify student identity within 5 seconds
2. **Accurate Geofencing** - Verify student presence within classroom boundaries
3. **Real-time Data Processing** - Immediate attendance validation and recording
4. **Secure Biometric Storage** - Protection of sensitive facial data

5. **User Dashboards -** For both instructors and students
6. **Scalability -** Handle multiple concurrent check-ins during class start times
7. **Integration Capabilities -** Connect with existing school management systems

3.3.2. Backend Technology Comparison

1. Node.js

Strengths:

- **Asynchronous processing -** Excellent for handling multiple concurrent requests
- **Real-time capabilities -** Socket.io integration enables instant attendance updates
- **JSON-native -** Seamless data exchange with mobile applications
- **NPM ecosystem -** Rich libraries for both facial recognition (face-api.js) and geolocation
- **Scalability -** Lightweight and high throughput for peak usage periods

Weaknesses:

- **CPU-intensive operations -** May require careful optimization for facial recognition processing
- **Security concerns -** Requires additional measures for storing sensitive biometric data
- **Learning curve -** Event-driven programming paradigm might be unfamiliar to some developers

Implementation considerations:

- Express.js for RESTful API development
- Socket.io for real-time attendance updates
- MongoDB or PostgreSQL for database solutions
- AWS Rekognition or Microsoft Face API integration for facial recognition
- JWT for secure authentication

2. Django (Python)

Strengths:

- **Python ML ecosystem -** Native integration with powerful facial recognition libraries (OpenCV, dlib)
- **Security features -** Built-in protections against common web vulnerabilities
- **ORM system -** Simplified database operations and migrations

- **Admin interface -** Ready-to-use dashboard for instructors
- **Scalability -** Can be deployed with Gunicorn and Nginx for high performance

Weaknesses:

- **Synchronous by default -** May require additional configurations for real-time operations
- **Performance -** Not as efficient as Node.js for handling concurrent connections
- **Mobile integration -** Requires more work to create efficient APIs for mobile clients

Implementation considerations:

- Django REST framework for API development
- Channels for WebSocket support
- PostgreSQL for relational data storage
- Face_recognition library or cloud services for facial recognition
- GeoDjango for geofencing functionality

3. Spring Boot (Java)

Strengths:

- **Enterprise-grade -** Robust, secure, and battle-tested framework
- **Performance -** Excellent handling of computational tasks like facial recognition
- **Mature ecosystem -** Comprehensive libraries and integrations
- **Multithreading -** Superior handling of CPU-intensive operations
- **Scalability -** Designed for high-throughput enterprise applications

Weaknesses:

- **Verbose -** More code required compared to Node.js or Django
- **Resource intensity -** Higher memory footprint
- **Startup time -** Slower initial loading compared to Node.js
- **Development speed -** Less rapid prototyping capability

Implementation considerations:

- Spring WebFlux for reactive programming
- JPA/Hibernate for database operations
- PostgreSQL or Oracle for enterprise-grade data storage
- Google Cloud Vision API or Amazon Rekognition for facial processing
- WebSockets for real-time communication

4. *Firebase*

Strengths:

- **Backend-as-a-Service -** Minimal server-side coding required
- **Real-time database** - Native support for instant attendance updates
- **Cloud Functions -** Serverless architecture for processing facial recognition
- **Authentication** - Built-in user management system
- **Scalability** - Automatic scaling with Google infrastructure

Weaknesses:

- **Vendor lock-in -** Dependency on Google's ecosystem
- **Limited customization -** Less flexibility for complex requirements
- **Pricing** - Can become expensive with high usage
- **Complex queries -** Less powerful than traditional SQL databases

Implementation considerations:

- Firebase Realtime Database or Firestore for attendance records
- Firebase Authentication for user management
- Cloud Functions for facial recognition processing
- Firebase Storage for storing reference facial data
- ML Kit for on-device facial recognition

5. *ASP.NET Core*

Strengths:

- **High performance** - Exceptional speed and efficiency
- **Cross-platform** - Runs on Windows, Linux, and macOS
- **Strong typing** - Fewer runtime errors and better code organization
- **SignalR** - Powerful library for real-time functionality
- **Security** - Comprehensive identity and protection features

Weaknesses:

- **Learning curve** - Steeper than some alternatives
- **Microsoft ecosystem** - Best performance with other Microsoft services
- **Community size** - Smaller than Node.js or Django communities
- **Resource requirements** - Higher than Node.js

Implementation considerations:

- Entity Framework Core for database operations
- SQL Server or PostgreSQL for data storage
- SignalR for real-time communication
- Azure Cognitive Services for facial recognition
- Spatial data types for geofencing implementation

3.3.3. Database Options

Database	Strengths	Weaknesses	Best For
PostgreSQL	Spatial data support, ACID compliance, JSON storage	Moderate learning curve	Comprehensive systems requiring both relational and spatial features
MongoDB	Schema flexibility, horizontal scaling, JSON native	Less robust transactions, no spatial indexing	Rapid development and prototype systems
Firebase Firestore	Real-time updates, serverless, easy mobile integration	Query limitations, pricing at scale	Mobile-first applications with simple data models
MySQL	Widespread adoption, good performance, solid tooling	Less advanced spatial features than PostgreSQL	Traditional applications with well-defined schemas
Redis	In-memory performance, pub/sub capabilities	Persistence limitations	Caching layer and real-time notifications

3.3.4. Facial Recognition Services

Service	Accuracy	Cost	Integration Complexity	Privacy Control
Amazon Rekognition	High	Pay-per-use	Moderate	Data stored in AWS
Microsoft Azure Face API	Very High	Tiered pricing	Low with .NET	Data stored in Azure
Google Cloud Vision	High	Pay-per-use	Low with Firebase	Data stored in GCP
OpenCV (self-hosted)	Moderate	Server costs only	High	Complete control
Face API.js (client-side)	Moderate	Free	Low	On-device processing

3.3.5. Geofencing Implementation

Approach	Accuracy	Battery Impact	Implementation Complexity
Native geofencing APIs	High	Moderate	Low (platform-specific)
Custom polygon calculation	Very High	Low to High (depends on implementation)	High
Third-party services (Google Geofence API)	Very High	Optimized	Low
Hybrid (GPS + WiFi/Bluetooth beacons)	Excellent	Moderate	High

3.3.6. Recommended Architecture

Based on the analysis, a recommended architecture would be:

1. **Primary Option: Node.js + MongoDB + face API.js**

Why this combination:

- **Performance** - Node.js excels at handling multiple concurrent connections, essential during class check-in periods
- **Real-time capabilities** - Native support for WebSockets enables instant attendance updates
- **face-api.js** - JavaScript-based facial recognition that integrates well with Node.js
- **Development speed** - Rapid prototyping and iteration with JavaScript across front and back ends
- **Scalability** - Easily scaled horizontally for handling varying loads

2. **Alternative Option: Django + PostgreSQL + Self-hosted Facial Recognition**

Why this combination:

- **Python ML ecosystem** - Excellent for institutions with existing data science expertise
- **Spatial data** - PostgreSQL with PostGIS provides powerful geofencing capabilities
- **Security** - Django's robust security features protect sensitive biometric data
- **Cost control** - Self-hosted facial recognition reduces operational costs
- **Admin capabilities** - Built-in admin interface speeds up development of instructor dashboards

3.3.7. Scalability and Performance Considerations

1. **Load balancing** - Essential for handling peak loads during class start times
2. **Caching strategies** - Implement Redis to cache student data and reduce database load
3. **Edge computing** - Process facial recognition on mobile devices when possible
4. **Database sharding** - For systems with thousands of students
5. **Asynchronous processing** - Queue non-critical operations during peak times

3.3.8. Security Considerations

1. **Biometric data protection** - Implement encryption at rest and in transit
2. **Limited data retention** - Store processed facial features rather than raw images
3. **User consent** - Clear opt-in process and data usage transparency
4. **Access controls** - Role-based permissions for viewing attendance data
5. **Audit logging** - Track all system accesses and changes

3.3.9. Our Conclusion

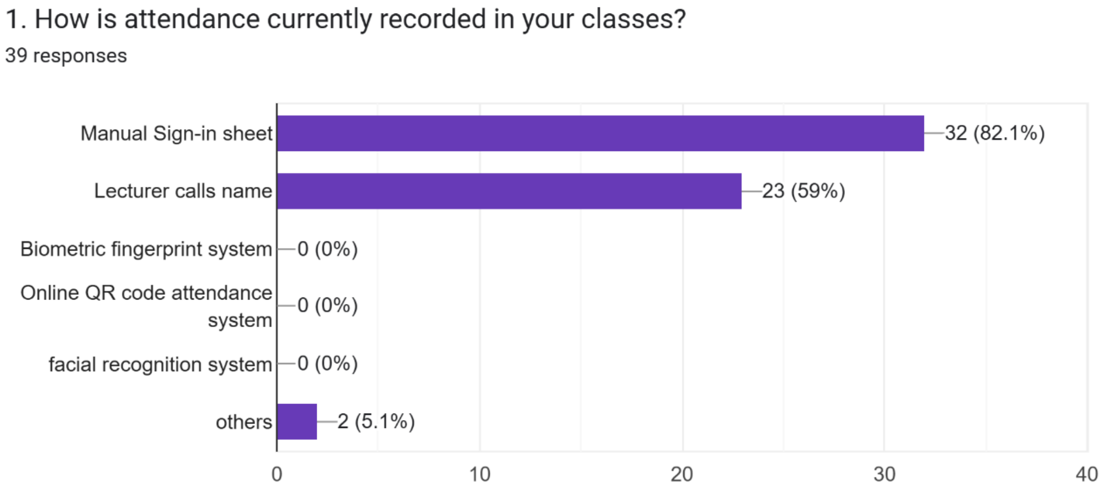
- The most balanced solution for the specified attendance management system would be a Node.js backend with MongoDB, supplemented by a self-hosted facial recognition (face API.js). This provides the optimal balance of development speed, performance, and scalability.
- For institutions with stricter data privacy requirements or existing Python expertise, the Django solution offers more control over sensitive biometric data while still delivering excellent performance.

Regardless of the chosen technology stack, implementing proper security measures and following data privacy best practices should be paramount given the sensitive nature of biometric data and student location information.

3.4. Survey and Interview Data Cleaning

3.4.1. Analysis of the Students Survey

1. "How is attendance currently recorded in your classes?"



Top Responses:

- Manual roll call (68%)
- Sign-up sheets (23%)

Implications for Development:

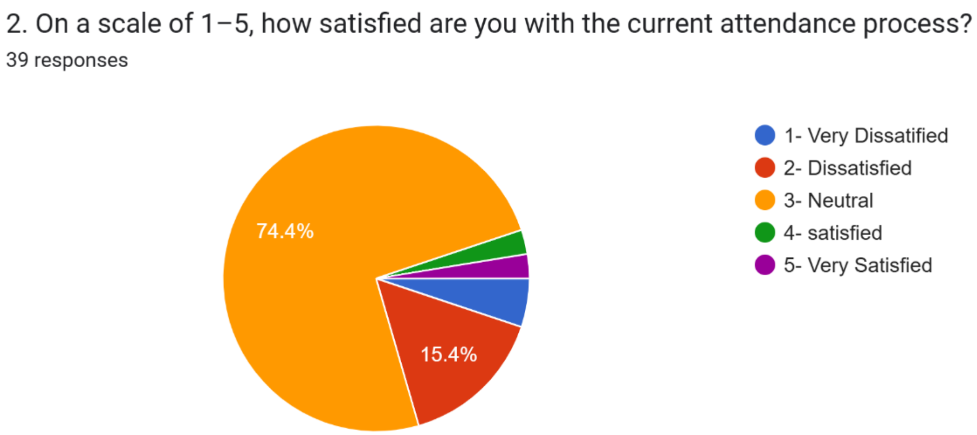
1. **Need for Automation**

- Since most use manual methods, the system must completely replace paper-based tracking with clear advantages (speed, accuracy).

2. **Transition Support**

- Users accustomed to paper may resist change. The system could:
  - Generate printable reports (for those who still want paper backups).
  - Include a simple, tutorial-driven UI or a training session on how the system works.
- **Risk:** Faculty may revert to old methods if the system is complex.
- **Mitigation:**
  - Conduct training workshops.
  - Assign "tech ambassadors" (students/faculty) to assist others.

2. "On a scale of 1–5, how satisfied are you with the current process?"



2. On a scale of 1–5, how satisfied are you with the current attendance process?

39 responses

Top Responses:

- Neutral (3) to Dissatisfied (4)

Implications for Development:

1. **Focus on Pain Points**

- ➢ Users dislike time-wasting and errors. The system must:
  - ❖ Reduce attendance time to <1 minute.
  - ❖ Provide clear error correction tools such as the "undo" button.

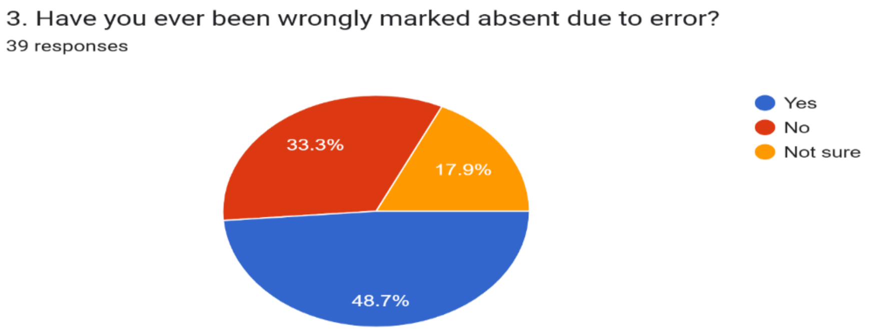
2. **Transparency**

- ➢ Show real-time status (e.g., "You’ve been marked present") to build trust.

Risk Assessment:

- ➢ **Risk:** High expectations may lead to disappointment if the system isn’t flawless.

3. "Have you ever been wrongly marked absent?"


3. Have you ever been wrongly marked absent due to error?

39 responses

Top Response:

- Yes (51%)

Implications for Development:

1. **Error Prevention**

➢ Use multi-factor checks:

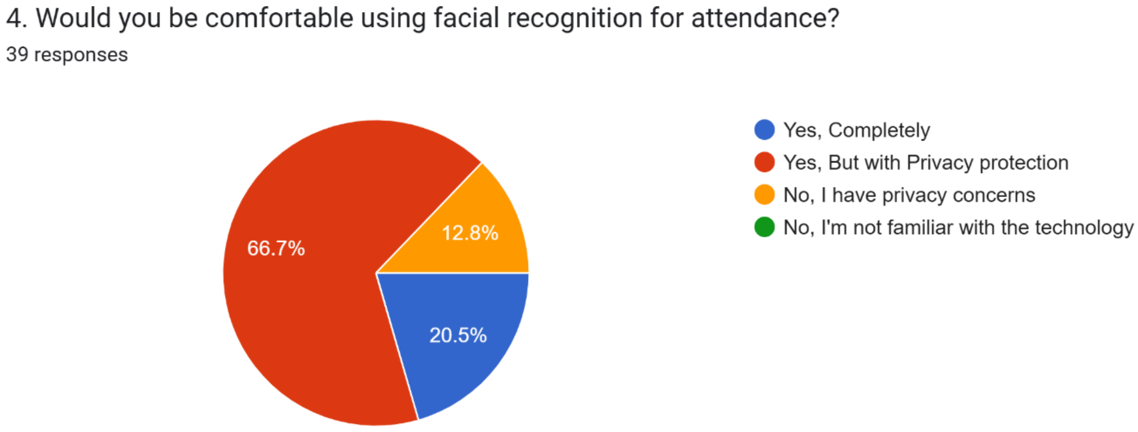
- ❖ Facial recognition + geofencing to confirm presence.
- ❖ Finger print as alternative in case of Poor Lighting.

Risk Assessment:

➢ **Risk:** Disputes could overwhelm staff if not handled efficiently.

➢ **Mitigation:** Provide Lecturers the Ability to carry out Multiple Facial Recognition log for students who registered the course.

4. "Would you be comfortable using facial recognition?"


4. Would you be comfortable using facial recognition for attendance?

39 responses

Top Responses:

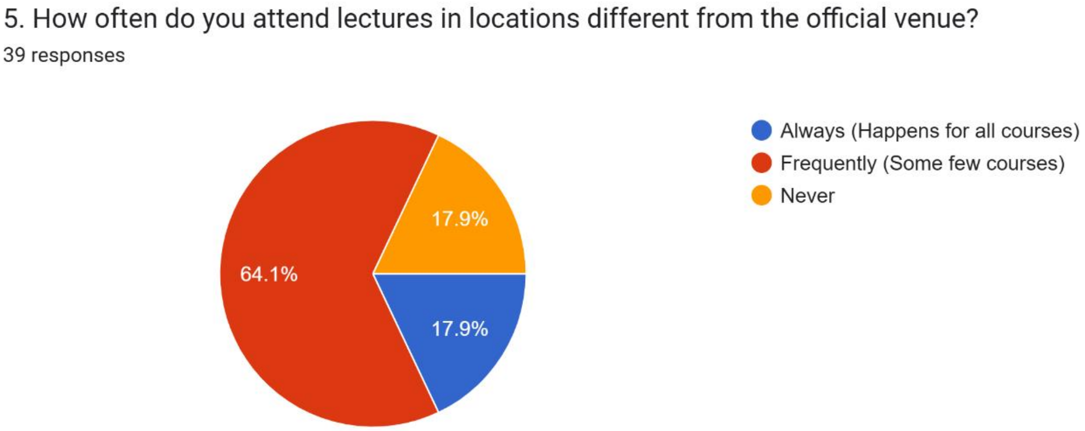
- Mixed (Yes, with privacy concern)

Implications for Development:

- **Privacy by Design**
  - Store facial data locally where possible.
  - Add an alternative QR code check-in or Finger print check in
  - Ensure the faces are compressed before saved to reduced space and ensure security.

- **Education and Validation**
  - Explain how data is used
  - Provide clarity to lecturers on the overall performance of students who attended their Lectures.

5. "How often do you attend lectures in unofficial locations?"



5. How often do you attend lectures in locations different from the official venue?

39 responses

Top Response:

Occasionally (28%)

Implications for Development:

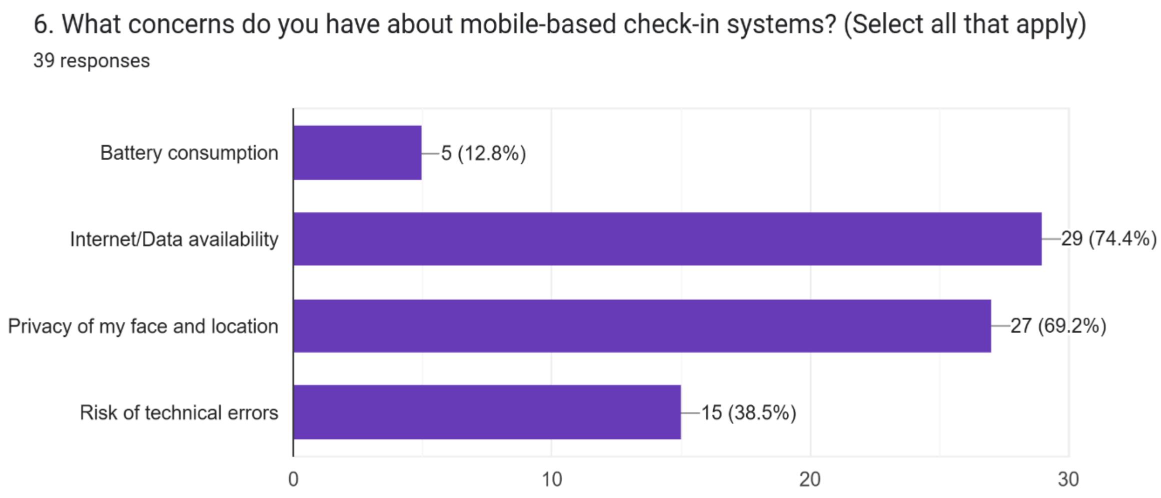
1. **Flexible Geofencing**

- ➢ Allow Lecturers to add temporary locations such as a catch-up class which is not scheduled on the school’s Time table. However, the Venue of the class must be a known venue to the system.

Risk Assessment:

- ➢ **Risk:** Students in unofficial locations may be excluded.
- ➢ **Mitigation:**
  - ➢ Provide a notification system to indicate students of last minutes changes in class venues or additional catch-up classes

6. "What concerns do you have about mobile check-in?"



6. What concerns do you have about mobile-based check-in systems? (Select all that apply)

39 responses

Top Responses:

- ➢ Internet dependency (74.4%)
- ➢ Privacy (69.2%)

Implications for Development:

1. **Optimize Performance**

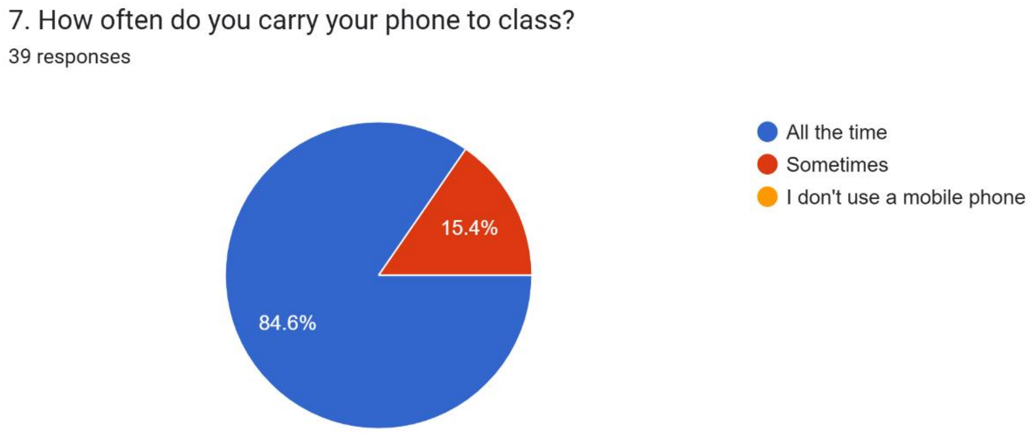
- ➢ Use low-power mode for geofencing.

- Cache data for offline use.

Risk Assessment:

- **Risk:** Students without reliable phones or internet that are present, may be excluded.
- **Mitigation:**
    - Provide access to lecturers to perform multiple student sign in using a reliable mobile device offered by the school for their personal attendance.

7. "How often do you carry your phone?"



Top Response:

- Always (94%)

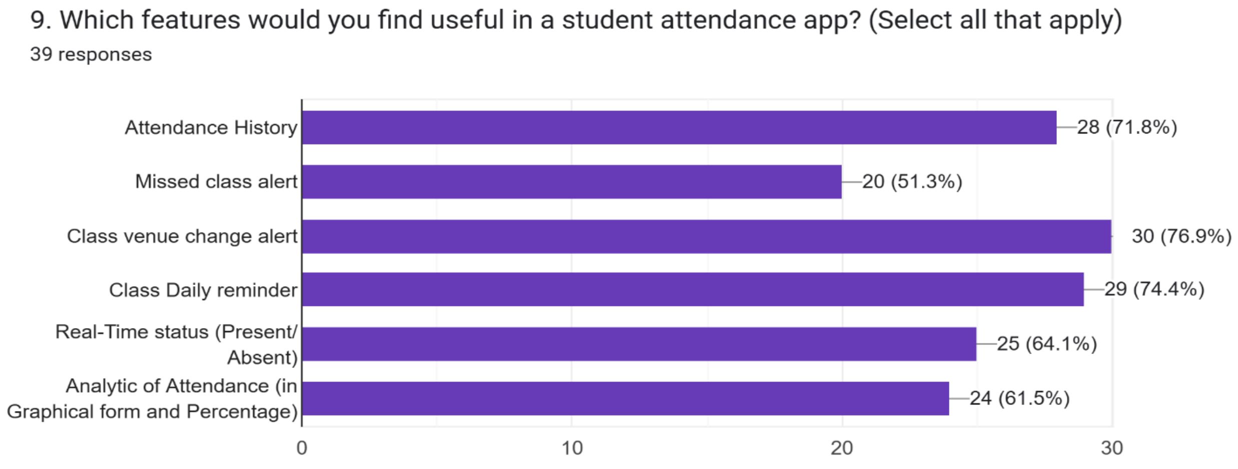
Implications for Development:

- Mobile-first design is viable, but include a web portal for rare cases.
- All Students own a mobile device meaning the system can be adopted

Risk Assessment:

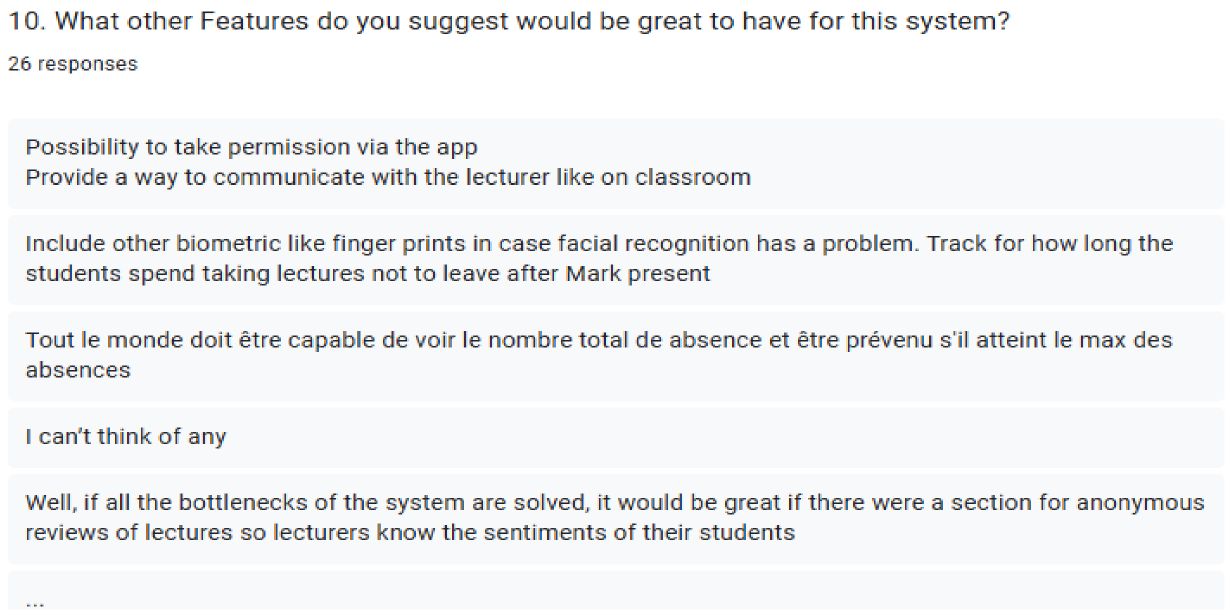
- Only a few don’t bring their phone

9–10. "Desired features?"



9. Which features would you find useful in a student attendance app? (Select all that apply)

39 responses



10. What other Features do you suggest would be great to have for this system?

26 responses

Possibility to take permission via the app
Provide a way to communicate with the lecturer like on classroom

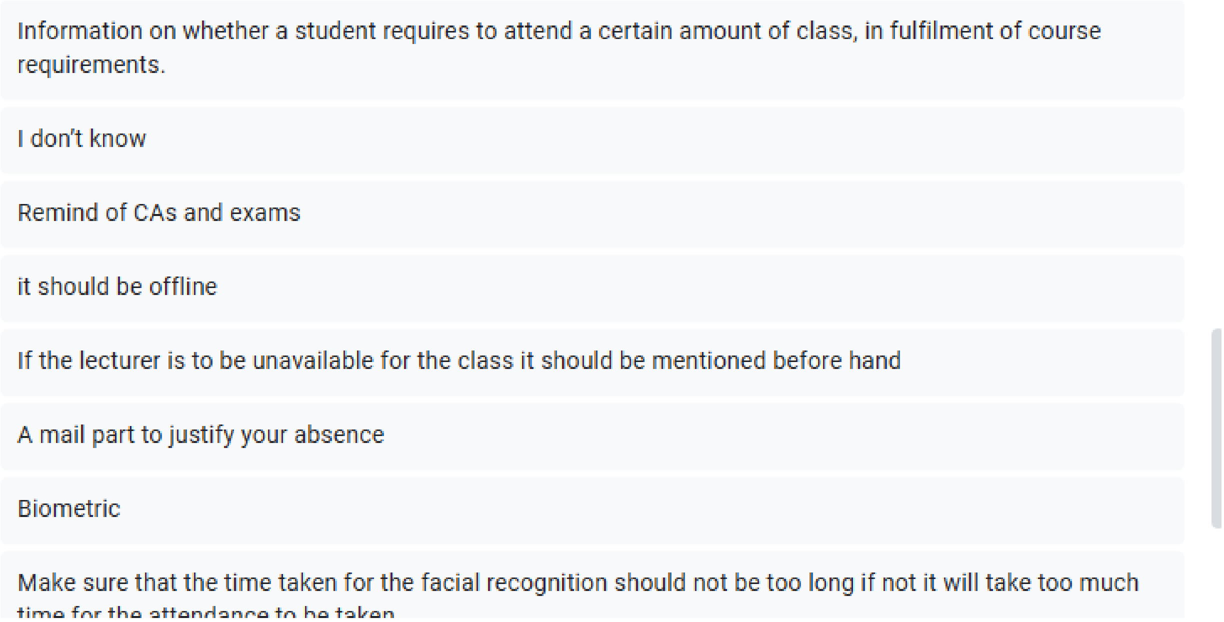
Include other biometric like finger prints in case facial recognition has a problem. Track for how long the students spend taking lectures not to leave after Mark present

Tout le monde doit être capable de voir le nombre total de absence et être prévenu s'il atteint le max des absences

I can't think of any

Well, if all the bottlenecks of the system are solved, it would be great if there were a section for anonymous reviews of lectures so lecturers know the sentiments of their students

...



Information on whether a student requires to attend a certain amount of class, in fulfilment of course requirements.

I don't know

Remind of CAs and exams

it should be offline

If the lecturer is to be unavailable for the class it should be mentioned before hand

A mail part to justify your absence

Biometric

Make sure that the time taken for the facial recognition should not be too long if not it will take too much time for the attendance to be taken

A Few Responses:

- Real-time notifications
- Direct communication with Lecturer
- Feedback for Technical issues
- Notifications and daily reminders for classes
- Status update on the Attendance for a current class
- Analytics of Attendance per class, per semesters, in an understandable (graphs) form.

Implications for Development:

- Prioritize these features to drive adoption.

Risk Assessment:

- **Risk:** Feature overload could delay launch.
- **Mitigation:** Launch with core features first, add others later.

3.5. Technology Best for the Front End

For the **Facial Recognition and Geofencing-based Attendance System**, React Native provides the ideal balance between performance, development efficiency, and access to native device features.

3.5.1. Key Advantages for the Attendance System

1. Cross-Platform Compatibility

- **Single Codebase**: Developers write one set of code that works on both iOS and Android, reducing development time and cost.
- **Consistent UI/UX**: Ensures a uniform experience across devices, important for students and lecturers using different phones.

2. Native Performance for Critical Features

- **Facial Recognition**: React Native allows integration with native libraries (like ML Kit or OpenCV) for fast, accurate face detection.
- **Geofencing**: Uses device GPS natively for precise location tracking within classrooms.
- **Background Processing**: Supports running geofencing checks even when the app is minimized.

3. Strong Ecosystem for Required Features

- **Pre-Built Libraries**: Ready-made solutions exist for:
  - Camera access and face detection such as React Native Vision Camera.
  - Location tracking such as React Native Geolocation.
  - Offline data storage such as WatermelonDB for local attendance logs.
- **community Support**: Large developer community ensures quick troubleshooting.

4. Addressing Survey Concerns

1. **Battery Drain (67% Concerned)**:

- React Native allows optimization of location updates (e.g., lower frequency when idle).

2. **Privacy (74% Hesitant)**:

- Supports on-device face processing (no need to upload images to a server).

3. **Offline Use (80% Need It)**:

- Local storage options (e.g., SQLite) enable attendance logging without internet.

4.Conclusion

The data collection methods provided a robust mix of quantitative and qualitative insights, ensuring alignment with stakeholder needs. Cleaning and categorization transformed raw data into actionable requirements, directly informing the system's design. This structured approach ensures the project is both user-centric and technically viable.

5.References

Facial Recognition Technologies

1. Geitgey, A. (2023). Face recognition with Python. https://github.com/ageitgey/face_recognition
2. Vincent, M. (2023). Face-api.js: JavaScript API for face detection and recognition. https://github.com/justadudewhohacks/face-api.js
3. OpenCV Team. (2023). OpenCV: Open Source Computer Vision Library. https://opencv.org/

Geofencing and Location Services

4. Bhattacharya, S., & Singh, K. (2019). Geofencing: Confining the data in limited geographical boundaries. Information Systems Design and Intelligent Applications, 397-406.
5. Geofencing API Documentation. (2023). Google Developers. https://developers.google.com/location-context/geofencing

Backend Technologies

6. Node.js Documentation. (2023). https://nodejs.org/en/docs/
7. Firebase Documentation. (2023). https://firebase.google.com/docs
8. Django Documentation. (2023). https://docs.djangoproject.com/
9. MongoDB Documentation. (2023). https://docs.mongodb.com/
10. PostgreSQL Documentation. (2023). https://www.postgresql.org/docs/

Attendance Systems Research

11. Lukas, S., et al. (2016). Student attendance system in classroom using face recognition technique. International Conference on Information and Communication Technology Convergence, 1032-1035.
12. Hussain, S., et al. (2019). Mobile-based attendance tracking system for educational institutions. International Conference on Frontiers of Information Technology, 319-324.

Security Considerations

13. Alhejaili, M., et al. (2021). Biometric authentication in mobile apps: Security challenges and best practices. International Journal of Advanced Computer Science and Applications, 12(6), 89-97.