REPUBLIC OF CAMEROON
****
Peace-Work-Fatherland
****
MINISTER OF HIGHER EDUCATION
****
UNIVERSITY OF BUEA

REPUBLIQUE DU CAMEROON
****
PAIX-Travail-Patrie
****
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
****
UNIVERSITE DE BUEA

FACULTY OF ENGINEERING AND TECHNOLOGY

COURSE TITLE:
INTERNET PROGRAMMING (J2EE) AND MOBILE PROGRAMMING
COURSE CODE:
CEF440

TASK 3 REPORT
Requirement Analysis

Course Intructor: Dr. Nkemeni Valery

GROUP IV

S/N	Names	Matricules
1	ARREY ABUNAW REGINA EBAI	FE22A142
2	AWA ERIC ANGELO JUNIOR	FE22A162
3	FAVOUR OZIOMA	FE22A217
4	OBI OBI ETCHU JUNIOR	FE22A291
5	VERBURINYUY JERVIS NYAH	FE22A324

Table of Contents

ABSTRACT

This document presents a comprehensive requirement analysis for the ***Mobile-Based Attendance Management System using Geofencing and Facial Recognition***. The primary objective is to automate attendance recording in educational institutions by leveraging mobile technologies, GPS-based geofencing, and facial recognition for identity verification. The analysis covers both functional and non-functional requirements, detailing user interactions, system constraints, and operational goals. Functional requirements specify the system's capabilities, such as user authentication, attendance marking within a defined geographical boundary, and real-time reporting. Non-functional requirements address performance, security, reliability, and usability standards. Additionally, the document outlines user roles, including administrators, lecturers, and students, and describes how each interacts with the system. This requirement analysis serves as a foundational step in the system design, ensuring alignment with user needs and institutional policies while promoting efficient and secure attendance management.

I - INTRODUCTION

1. Purpose

The purpose of this document is to define the software requirements and conduct a thorough requirement analysis for the Mobile-Based Attendance System. This system leverages facial recognition and geo-fencing to ensure accurate, secure, and real-time attendance tracking for educational institutions. The document will serve as a foundation for the design, development, testing, and deployment phases of the project.

2. Scope

This mobile-based attendance system will allow students to check in to classes using facial recognition and GPS validation to confirm their physical presence within a predefined geofenced area (e.g., a classroom or campus). Instructors and administrators will be able to monitor, view, and manage attendance records in real-time. The system aims to eliminate proxy attendance, streamline monitoring, and ensure accountability with biometric and location data. Key features include:

- Facial recognition-based student authentication.
- Geo-fencing for location validation during attendance.
- Real-time check-in and attendance marking.
- Dashboard for instructors to view attendance logs.
- Secure handling of biometric and location data.

3. Overview of the document

This document begins with an introduction to the system and proceeds to the requirement analysis phase, including requirement review, issue identification, prioritization, and classification. It then elaborates on functional and non-functional requirements, interface specifications, and system constraints.

II - OVERALL DESCRIPTION

1. Product Perspective

The Mobile-Based Attendance System is a standalone mobile application that integrates facial recognition and geo-fencing technologies to automate the process of recording student attendance. It is designed to work with existing mobile device features such as GPS and the camera, while optionally integrating with institutional databases for syncing student and class records. The system is intended to replace or enhance traditional attendance methods by adding biometric verification and location awareness for higher accuracy and fraud prevention.

This system will be the first of its kind within the institution and will operate independently of existing learning management systems (LMS), although future integration is possible.

2. Product Functions

Major functionalities of the system are;

- Student Authentication using facial recognition.
- Location Validation through GPS-based geo-fencing to ensure students are within the class/campus.
- Attendance Recording once both identity and location are validated.
- Real-Time Dashboard for instructors to view and monitor attendance logs.
- Student Attendance History viewable through their mobile app.
- Admin Tools to manage users, view statistics, and configure geo-fence parameters.

3. User Classes and Characteristics

- **Student -** Uses the mobile app to check in to class. Must register and validate face during onboarding.
- **Instructor -** Views attendance in real-time, manually overrides errors, and accesses reports.

- **Admin -** Sets geo-fence coordinates, manages users, and maintains the system.

4. Operating Environment

Mobile Platform: Android 8+ and iOS 12+

Camera: Front-facing camera with at least 720p resolution.

GPS Module: Enabled and accurate within 50 meters.

Cloud Server: Backend API hosted on a cloud platform (e.g., AWS, Firebase, or Render).

Database: Cloud-hosted database (e.g., PostgreSQL, Firebase Firestore, or MongoDB).

Internet: Required for syncing attendance data and user authentication.

5. Design and Implementation Constraints

Data Privacy: Must comply with data protection regulations regarding biometric data.

Mobile Hardware: Must work on mid-range smartphones with limited RAM and CPU.

Network Dependency: System should be tolerant of minor network delays.

Face Model Training: Pre-enrollment of student facial data is required.

Location Accuracy: Geo-fencing accuracy might vary due to GPS signal conditions.

6. User Documentation

The following documentation will be provided:

- Student User Guide: How to register, check-in, and view attendance.
- Instructor Manual: Monitoring dashboard usage and handling exceptions.
- Admin Manual: Setting up the system, geo-fence configuration, and data management.

III - REQUIREMENT ANALYSIS

i. Review and Analysis of Requirements

The first step involves reviewing all gathered requirements to ensure they are:

✧ **Clear:** Each requirement must be understandable and unambiguous

✧ **Complete**: All required details should be included

✧ **Feasible**: Technically and practically implementable within scope

✧ **Independent**: Should not unnecessarily overlap or conflict with others

S/N	Description	Clarity	Completeness	Feasibility
1	User Registration	✔	✔	✔
2	User Login	✔	✔	✔
3	System Check for Existing Users	✔	✔	✔
4	Facial Recognition Check-In	✔	✔	✔
5	Geo-Fencing Verification	✔	✔	✔
6	Real-Time Attendance View	✔	✔	✔
7	Attendance History View	✔	✔	✔
8	Parent Notification	✔	✔	✔
9	Instructor Manual Override	✔	✔	✔
10	Student Dashboards	✔	✔	✔
11	Reporting & Analytics Dashboard	✔	✔	✔
12	Data Privacy Handling	✔	✔	✔
13	Usability Enhancements	✔	✔	✔
14	High Availability	✔	✔	✔
15	Platform Compatibility	✔	✔	✔
16	Scalable & Responsive Design	✔	✔	✔
17	Secure Authentication	✔	✔	✔
18	Archive Attendance Data	✔	✔	✔
19	Attendance Analysis Post-Lecture	✔	✔	✔
20	Mark Absence in Red	✔	✔	✔
21	Confirm Instructor Override	✔	✔	✔
22	Update Class Location	✔	✔	✔
23	Notifications to Users	✔	✔	✔
24	Check-In Duration ≤ 5 Seconds	✔	✔	✔
25	Define Virtual Classroom Boundaries	✔	✔	✔
26	Request Permission from Lecturers	✔	✔	✔
27	Data Deletion Policies	✔	✔	✔
28	Filtering by Course, Date, Student	✔	✔	✔
29	Student Attendance History & Participation View	✔	✔	✔
30	Automatic Clock-In/Out	✔	✔	✔

ii. Identified Issues

After a thorough review of the gathered requirements, several ambiguities, overlaps, contradictions, and missing elements were identified. These issues are outlined below with corresponding suggestions for resolution.

	Issue Type	Description	Suggestion
10 & 22	Redundancy / Overlap	Instructor Manual Override (FR10) and Confirm Instructor Override (FR22) appear to be overlapping functionalities.	Merge both or clarify distinction. Rephrase FR22 as: “Require confirmation step for Instructor Manual Override.”
4	Ambiguity	Facial recognition check-in lacks information on fallback methods in case of failure or unavailability.	Add a fallback option such as password or PIN-based check-in.
5	Ambiguity	Geo-fencing rule does not specify how strict the location radius is or if it is configurable.	Define geo-fence radius range and mention configurability per location or class.
23	Ambiguity	Notification system is unclear — what types, who receives them, and when?	Specify events (e.g., absence alerts, clock-ins), target users (student, parent, admin), and medium (email, app).
25	Ambiguity	Defining virtual classroom boundaries may be overlapping with geo-fencing, and its intent is unclear.	Clarify whether this refers to GPS boundaries or digital constraints like IP/geolocation policies.
10 & 27	Conflict / Contradiction	Manual instructor override (FR10) vs. student request for permission (FR27) may cause role conflicts.	Clarify role authority — is instructor override independent, or must permission always be requested?
—	Missing Requirement	Authentication Method not clearly specified.	Detail if OAuth, 2FA, biometric, or custom auth is supported.
—	Missing Requirement	Device and platform specifications are not outlined.	Include minimum device specs (e.g., Android 9+, iOS 12+) and supported platforms (mobile, web).
—	Missing Requirement	Admin roles and permissions are not clearly defined.	Add requirements for admin user management, dashboard access, and data control.
—	Missing Requirement	Error handling scenarios are not covered (e.g., GPS/facial recognition failures).	Define fallback flows and error messages for common failure cases.
—	Missing Requirement	No mention of data audit trails or logs despite features like override and deletion.	Add audit logging requirement to track sensitive user actions.
—	Missing Requirement	Security practices (e.g., encryption, session timeout, rate limiting) are under-specified.	Include requirements for encryption of data at rest and in transit, as well as session handling.

iii. Requirement Prioritization

Use the MoSCoW prioritization method to categorize requirements as:

- Must Have
- Should Have
- Could Have
- Won’t Have (for now)

This will help our team focus on critical features for initial development.

i. Must Have (Critical for MVP or Legal/Compliance needs)

User Authentication – Fundamental to access any functionality, prevents duplication, ensures integrity,.

Facial Recognition Check-In – Core innovation feature; essential for attendance validation.

Geo-Fencing Verification – Key attendance validation constraint; ensures physical presence.

Attendance History View – Critical for transparency and record-keeping.

Secure Authentication – Mandatory for protecting personal data.

Data Privacy – Legal requirement (e.g., GDPR compliance).

Platform Compatibility – Ensures users across Android/iOS can access the system.

Check-In Duration ≤5 Seconds – Affects usability and class flow.

Scalable & Responsive Design – Supports usage in real-world school environments.

Availability (99%+) – Ensures system is usable during school hours.

Attendance Analysis Post-Lecture – Provides essential insight after sessions.

ii. Should Have (Important but not necesary for MVP)

Parent Notification – Important for engagement, especially in schools.

Instructor Manual Override – Allows fallback in case of system issues.

Request Permission from Lecturers – Edge use-case, can be added later.

Student Dashboards – Improves user experience for students.

Reporting & Analytics Dashboard – Valuable for institutional insight.

Mark Absence in Red – Helps visual tracking of attendance quickly.

Confirm Instructor Override – Prevents unauthorized attendance changes.

Define Virtual Classroom Boundaries – Ensures precision in geolocation logic.

Data Deletion Policies – Needed for compliance but not urgent for MVP.

Update Class Location – Useful but rare; can be m\anaged manually initially.

Filtering by Course, Date, or Student – Improves usability, not a blocker.

iii. Could Have (Nice to include, low risk if left out initially)

Update Class Location – Useful but rare; can be managed manually initially.

Notifications (to all roles) – Useful for engagement, but not MVP-critical.

Archive Attendance Data – Useful for long-term records, not urgent.

Automatic Clock-In/Out – Adds convenience, but needs thorough testing.

Usability Enhancements – Important, but initial version can launch with a basic UI.

iv. Won’t Have (For now)

Biometric input beyond facial recognition – e.g finger prints or Iris scan

In Application chat or messages – better handle via existing communication platforms.

However, Automatic Clock-In/Out and Request Permissions may be deferred beyond MVP if development resources are limited.

iv.Requirement Classification

I. Functional Requirements

These describe what the system should do. They were classified under 7 major categories;

1. User Management

- □FR1: User Registration
- □FR2: User Login

2. Attendance Mechanism

- □FR3: Facial Recognition Check-In
- □FR4: Geo-Fencing Verification
- □FR5: Instructor Manual Override
- FR6: Filtering by Course, Date, Student

3. Attendance Monitoring and Visualization

- FR7:Student dashboard
- FR8:instructor dashboard

4. Notifications and Communication

- □FR9: Parent Notification
- □FR10: Notifications to Users

II. Non-functional Requirements

- □NFR1: Data privacy compliance
- □NFR2: Usability
- □NFR3: Reliability
- □NFR4: Compatibility
- □NFR5: Scalability
- □NFR6: Security
- □NFR7: Performance
- □NFR8: Data deletion

User Interface Requirements

- □NFR9: Mobile app
- □NFR10: Facial recognition
- □NFR11: Dashboard

Hardware Interface Requirements

- □NFR12: Camera
- □NFR13: GPS
- □NFR14: Sensors

Software Interface Requirements

- □NFR15: Facial recognition engine
- □NFR16: GPS and Mapping API
- □NFR17: Notification service
- □NFR18: Authentication Service
- □NFR19: Database Integration

Communication Interface Requirements

- □NFR20: Internet connectivity
- □NFR21: API communication
- □NFR22: Push notifications

IV. FUNCTIONAL REQUIREMENTS

This section provides an in-depth specification of the system's functional requirements. Each requirement defines a specific function the system must perform to achieve the desired behavior and ensure a robust, efficient, and user-friendly attendance management experience.

FR1 – User Registration

- The system allows students to create an account by providing required personal details, academic information, and biometric data (e.g., facial image). This enables secure future check-ins and identification.
- The system also checks for existing users

FR2 – User Login

- Registered users can log in to the application using traditional credentials (username/password) or biometric authentication such as facial recognition for enhanced security. This is to ensure secure authentication

FR3 – Facial Recognition Check-In

- During class check-in, the system uses facial recognition to authenticate the student's identity and log attendance. The face is matched against stored biometric data for verification.

FR4 – Geo-Fencing Verification

- Check-in is allowed only if the student is physically present within the geo-fenced boundary of the classroom. The system uses GPS to confirm the student's location.

FR5 – Instructor Manual Override

- Instructors have the authority to manually mark attendance for students in case of system failure or valid justifications provided by students.

FR6 – Student Dashboards

- Each student has a personalized dashboard showing attendance stats, upcoming classes, check-in history, and performance indicators.
- Students have access to a timeline of their past attendance records, categorized by course, date, and status, to monitor their participation.
- Students can submit requests for permission and upload viable proof for absence.

FR7 – Lecturer Dashboard

- The system provides analytics to instructors and administrators, including graphs and charts on student attendance trends, averages, and anomalies.
- Students who miss a class are visually flagged in red in the attendance interface to allow easy identification by instructors.
- Administrators can modify the GPS coordinates for virtual classroom boundaries if class locations change.
- Admins can set and adjust GPS boundaries for classrooms to define the physical space where check-ins are valid.

FR8 – Filtering by Course, Date, Student

- The system enables detailed filtering of attendance data for reports and dashboards by course, date, or student.

FR9 – Parent Notification

- Automated notifications are sent to parents or guardians to inform them when a student fails to check in or when they are marked absent.

FR10 – Notifications to Users

Users receive real-time notifications for check-ins, overrides, warnings, and system alerts via push notifications or in-app messages.

V - NON-FUNCTIONAL REQUIREMENTS

NFR1 – **Data privacy compliance:** System must comply with data privacy laws such as GDPR.

NFR2 – **Usability:** User interface should be user-friendly and intuitive.

NFR3 – **Reliability:** System must ensure 99% uptime.

NFR4 – **Compatibility:** Supports Android 8+ and iOS 12+ mobile devices.

NFR5 – **Scalability:** System should handle 1000+ concurrent users.

NFR6 – **Security:** Secure user authentication using OAuth 2.0 or JWT.

NFR7 – **Performance:** Check-in process should complete in ≤5 seconds.

NFR8 – **Data deletion:** Users should control and request deletion of their personal data.

NFR9 – **Mobile app:** Mobile app must allow intuitive navigation for students, instructors, and admins with role-based access.

NFR10 – **Facial recognition:** UI should guide users for proper alignment and show clear success/failure check-in messages.

NFR11 – **Dashboard:** Dashboards must show visual trends, filters, and summary stats for instructors/admins.

NFR12 – **Camera:** Mobile device must have at least a 5MP camera accessible by the app with necessary permissions.

NFR13 – **GPS:** Device GPS must have accuracy within 20 meters for geo-fencing checks.

NFR14 – **Sensors:** Mobile motion/environment sensors should be optionally supported for additional validation.

NFR15 – **Facial recognition engine:** Must support real-time face embeddings and identity verification within 1 second using on-device or cloud-based tools.

NFR16 – **GPS and Mapping API:** Use third-party services (Google Maps, OpenStreetMap) for geo-boundary definitions and verifications.

NFR17 – **Notification service:** System should support real-time push alerts using Firebase Cloud Messaging or equivalent.

NFR18 – **Authentication Service:** Secure login/session management must be implemented using OAuth2, JWT, or Firebase Auth.

NFR19 – **Database Integration:** System must work with PostgreSQL, Firebase Realtime DB, or MongoDB for user and attendance data.

NFR20 – **Internet connectivity:** Internet is required for sync and cloud-based facial recognition; offline fallback with sync is required.

NFR21 – **API communication:** REST or GraphQL APIs must be used, with all data sent over HTTPS with TLS encryption.

NFR22 – **Push notifications:** Mobile devices should receive real-time alerts via push notifications.

VI. CONCLUSION

The requirement analysis for the Mobile-Based Attendance Management System using Geofencing and Facial Recognition has established a clear understanding of the system's goals, user expectations, and operational needs.

By defining both functional and non-functional requirements, this document ensures that the system will not only perform its core tasks effectively such as secure user authentication and accurate attendance recording but will also meet standards for performance, reliability, and user-friendliness. The roles and interactions of administrators, lecturers, and students have been carefully considered to guarantee a seamless user experience.

This analysis provides a solid foundation for the subsequent design and implementation phases, helping to minimize risks, align development efforts with stakeholder needs, and ultimately deliver a robust solution for modernizing attendance management in educational institutions.

VII. APPENDIX

i. Glossary

Term/Acronym	Definition
API	Application Programming Interface – a set of functions allowing communication between software applications.
GPS	Global Positioning System – satellite-based navigation system used to determine precise location.
Geo-fencing	A virtual boundary defined by GPS coordinates used to validate user location.
JWT	JSON Web Token – a secure method for transferring claims between parties for authentication.
MVP	Minimum Viable Product – a basic version of the product with core functionality to meet key requirements.
UI	User Interface – the graphical layout of the application through which users interact with the system.
FR/NFR	Functional/Non-Functional Requirements – system behaviors and quality attributes, respectively.
GDPR	General Data Protection Regulation – a regulation on data protection and privacy.
OCR	Optical Character Recognition – technology used to convert textual images into machine-readable text.

ii. References

- ❖ IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- ❖ Google Maps API Documentation: https://developers.google.com/maps
- ❖ Firebase Authentication Documentation: https://firebase.google.com/docs/auth
- ❖ Mediapipe/Facial Recognition Libraries: https://mediapipe.dev
- ❖ GDPR Compliance Guidelines: https://gdpr.eu

iii. Stakeholder Information

Stakeholder	Role	Responsibilities
Project Team System	Students/Developers	Develop, test, and deploy the system.
Lecturer/Supervisor	Project Supervisor	Oversee the project progression and ensure requirements are met.
Students	End users	Register, check-in for classes, and view attendance status.
Instructors	System Users	Monitor student attendance, manage overrides, and generate reports.
Administrators	System Admins	Manage user accounts, define geo-fences, and oversee system maintenance.
Parents/Guardians	Secondary Users	Receive notifications regarding student attendance.

iv. Assumptions and Dependencies

- All students and instructors have access to mobile devices with working cameras and GPS functionality.
- Institutional support for integrating and hosting the backend infrastructure.
- Reliable internet connectivity for real-time attendance syncing.
- Student and instructor cooperation for initial registration and system onboarding.
- Compliance with data privacy regulations (GDPR or local policies) for biometric and location data handling.

v. Some System Diagrams

a. System Flow Diagram (Textual Representation)

[Start]

↓

[User Login]

↓

[Select Course/Session]

↓

[Geofence Validation]

↓ (If within geofence)

[Facial Recognition & Liveness Detection]

↓ (If matched)

[Attendance Recorded]

↓

[Confirmation & Real-Time Dashboard Update]

↓

[User Views Reports/History]

↓

[Logout/End]

vi. Additional notes

- Future versions may include additional features like NFC-based check-ins or deeper LMS integration.
- All third-party APIs and services must be evaluated for security and privacy compliance before implementation.
- Continuous user feedback will be gathered post-deployment for iterative improvements.