UNIVERSITY OF BUEA

Buea, South West Region

Cameroon

P.O. Box 63,

Tel: (237) 3332 21 34/3332 26

REPUBLIC OF CAMEROON

PEACE - WORK - FATHERLAND

Task3:

REQUIREMENT ANALYSIS FOR A SMART ATTENDANCE MONITORING SOLUTION USING FACIAL RECONITION AND GEOFENCING

Instructor: **Dr Nkemeni Valery**

Course: **CEF440: Internet Programming and Mobile Programming**

By Group7

NAMES	MATRICULE
FONYUY VERENA MONYUYTA-AH	FE22A220
KENFACK DONJIO ABEL BRUNEL	FE22A380
NSONDO MIRELLE NYISEKINYI	FE22A283
TATA THECLAIRE GHALANYUY	FE22A310
UNJI STEPHEN UKU	FE22A323

2024/2025 ACADERMIC YEAR

Table of content

Introduction

Requirement analysis is a critical phase in the software development lifecycle (SDLC). It serves as the foundation for successful system design and implementation by evaluating the expectations, constraints, and dependencies of a project.

Before diving into the analysis, here's a quick recap of the project features:

Component	**Description**
Platform	Mobile-based (likely Android, iOS)
Core Features	Facial recognition, geofencing, real-time check-in
Users	Students, Instructors (and possibly Admins)
Supporting Modules	Attendance history, dashboard, course filtering
Objective	Accurate, quick, and tamper-proof attendance collection
Technologies Mentioned	Machine learning libraries, GPS, mobile cameras

I. Review and analysis of the requirements gathered

This section provides a foundational overview of the project to establish the technical and functional context in which the requirement analysis is based. It defines key components, user roles, objectives, and enabling technologies that frame the system's scope.

In this section, we rigorously analyze the requirements of a mobile-based attendance management system that integrates **facial recognition** and **geofencing**, focusing on:

- **Completeness**: Are all necessary requirements captured?
- **Clarity**: Are the requirements unambiguous and measurable?
- **Technical Feasibility**: Can the requirements be implemented given current technology?
- **Dependency Relationships**: What interdependencies exist among features or components?

1. Completeness Analysis

Completeness ensures that all the functional and non-functional expectations of the stakeholders are identified and captured in the requirement document in the previous stage (require , leaving no critical functionality unaddressed.

Completeness refers to whether the requirements fully describe **all system functionalities**, **boundary conditions**, **user scenarios**, and **constraints**.

1.1 Present Functional Requirements

The current documentation mentions the following explicit requirements:

Requirement	**Present?**	**Notes**
Facial recognition check-in	yes	Clearly stated
Geofencing validation	yes	Clearly stated
Student check-in process (< 5s)	yes	Quantified requirement
Instructor attendance dashboard	yes	Included
Filtering by course/date/student	yes	Important for management
Student view of attendance history	yes	Included

1.2 Missing Functional Requirements

While core features are covered, several critical system functions and constraints are not mentioned:

A. User Authentication

- No mention of:
  - Student login/logout
  - Instructor/admin authentication
  - Password reset, account recovery

B. Role Management

- Undefined user roles: Can instructors register students or courses?
- Is there a Super Admin for overall system control?

C. Course and Schedule Management

- How are courses created?
- Is the check-in tied to a course schedule (date/time)? Or is it open-ended?

D. Session Control

- When is check-in enabled or disabled?
- Is attendance only possible during class time?

E. Error Handling

- What happens if:
  - Face not recognized?
  - Student is at location but GPS is inaccurate?
  - Face matches multiple records?

F. Notification System

- No alerts for:
  - Missed attendance
  - Successful/failed check-in
  - Instructor notifications for low attendance

G. Reporting and Analytics

- No mention of:
  - Attendance statistics over time
  - Export features (PDF/CSV)
  - Institutional performance metrics

H. Administrative Controls

- Not clear how users or data are managed:
  - Who creates student accounts?
  - Can instructors modify records?
  - How are students registered to courses?

1.3 some resolutions

System Resolutions:

- Student management modules will be developed to handle registrations, password recovery, and role-based access.
- A course management feature will be introduced, allowing instructors to create and manage course schedules tied to attendance sessions.
- Session control logic will restrict check-in strictly to scheduled class times and locations.
- Error handling mechanisms will be incorporated to manage face recognition failures and GPS inaccuracies, including retry options.
- Notification services will send real-time alerts on successful/failed check-ins and missed sessions to students and instructors.
- Reporting modules will generate real-time attendance analytics and allow data exports in formats such as CSV and PDF.
- Administrative tools will enable authorized personnel to create and manage user accounts and system settings securely.
- Student and instructor authentication will be implemented through secure login and multi-factor authentication.

2. Clarity Analysis

This section measures the preciseness of requirement definitions, ensuring that they are expressed in an unambiguous, testable, and measurable manner to guide the system's development.

Clarity ensures each requirement is **specific**, **unambiguous**, and **measurable**, leaving no room for misinterpretation by developers, testers, or stakeholders.

2.1 Clear Requirements

Some requirements are well-defined and measurable:

- "**Check-in must take no more than 5 seconds per student**" is a clearly measurable performance benchmark.
- "**Students must be within the geofenced classroom boundary before check-in is permitted**" clearly defines an access constraint.

2.2 Ambiguous or Vague Requirements

Requirement	Ambiguity
"Uses geofencing…"	No details on the boundary radius (e.g., 20m? 50m?). What about GPS drift?
"Facial recognition using ML"	Unclear: which algorithm? CNN? Haar Cascades? Will models be pre-trained or custom-trained? Will it run on-device or in the cloud?
"Supports mobile devices"	Android only? iOS? be cross-platform (iOS and Android)? Minimum version support?
"Secure storage of biometric data"	Local or cloud storage? Encryption standard (e.g., AES-256)?
"Real-time attendance"	What defines real-time? Instant DB update? < 1 second latency?
Unclear error boundaries	what if a student is in the geofence but not recognized by the camera?

SOME RESOLUTIONS

- Geofence radius: e.g., 30 meters with a ±5m tolerance.
- Recognition threshold: 85% similarity score for match.
- Device support: Android 9+, camera ≥ 8MP, GPS access.
- Security standard: AES encryption, SSL transport.

3. Technical Feasibility Analysis

Technical feasibility assesses whether the envisioned system can be realistically built and deployed given available technology frameworks, device capabilities, security standards, and resource constraints.

Technical feasibility analyzes whether the proposed system can realistically be implemented with current technology, performance limits, and integration complexity, within project constraints such as time, cost, and device capabilities.

3.1 Feasibility of Facial Recognition

- **Mobile Device Capability:** Most modern smartphones support real-time camera input and image processing.
- **Available Frameworks:** Libraries such as **OpenCV**, **MediaPipe**, **TensorFlow Lite**, and **Google ML Kit** provide APIs for facial recognition that can run efficiently on-device.
- **Challenges:** Varying lighting conditions, face occlusions (e.g., masks, hats), and camera quality can affect accuracy.

3.2 Feasibility of Geofencing

- **Technological Availability:** Android and iOS natively support geofencing APIs via **Google Location Services** and **Core Location** respectively.
- **Feasibility Issues:**
  - GPS signal is weak indoors or in dense urban environments.
  - Typical GPS accuracy on mobile devices ranges from 3–10 meters.
  - Dependence on background location permissions can affect user adoption.

3.3 Feasibility of Real-Time Processing

- Check-in completion within 5 seconds is feasible if:
  - Facial recognition is done on-device (to avoid network latency).
  - Geofencing is pre-initialized and processed via cached coordinates.
- Requires efficient multi-threaded programming and device resource management.

3.4 Security and Privacy Feasibility

- Facial biometric data is highly sensitive.
- Storing and processing such data requires compliance with data protection standards (e.g., **GDPR**, **HIPAA**).
- Encrypted storage (AES, SSL/TLS in transit) must be enforced.

4. Dependency Relationships

Dependencies define how different system components rely on each other. Understanding these relationships is critical to identifying potential bottlenecks or failure points during system integration.

Interdependencies among components define the critical path for development and testing.

Understanding the interdependence of various system components is crucial for project scheduling, risk mitigation, and system integration.

Component	**Depends On**	**Impact**
Face Recognition	Camera, image preprocessing, ML inference	Failure in any = invalid check-in
Geofencing	GPS, network, user permissions, OS power settings	May fail silently if disabled
Check-in Timing	Both Face + Geofence success	All must sync within time limit
Dashboard	Backend data sync, role-based access	Requires real-time DB
Notification System	Background tasks, OS support	Background restrictions on Android 10+
App Launch	Auth + Role check	Cannot proceed without valid login
Attendance Logging	Secure, fast DB write	Delay here affects real-time promise

Risks

- **Permission Denial:** Users may decline camera/GPS access, rendering features inoperable.
- **Battery Optimization Policies:** May prevent background services (like geofencing) from functioning properly on certain devices.
- **Model Drift:** Facial recognition models may lose accuracy over time without periodic retraining.

5. Potential Risks and Mitigation

Risk	Impact	Mitigation
Low-light or occluded face	Recognition failure	Use pre-processing (e.g., histogram equalization)
GPS spoofing or drift	False check-in	Validate with Wi-Fi + cell tower + GPS fusion
Permissions denied	App unusable	Prompt clearly with rationale + fallback options
Device variance	Inconsistent experience	Set minimum hardware requirements
Network dependency	Offline unavailability	Allow offline caching with delayed sync
Privacy violation	Legal consequences	Anonymize data, store embeddings not photos

Every system faces inherent risks during its development and deployment. This section identifies these potential risks early on and outlines clear strategies for mitigating their impacts.

6. Summary of Analysis

Criteria	Evaluation
Completeness	Core features included, but missing critical sub-features (auth, roles, error handling, session logic, notifications)
Clarity	Several ambiguities exist; needs quantification and technical specificity
Technical Feasibility	Achievable with current tools and libraries on modern devices
Dependencies	Multiple real-time dependencies; must be rigorously tested together

A detailed synthesis of how well the system's requirements meet the project's completeness, clarity, feasibility, and interdependency standards, laying the groundwork for the design phase.

7. Final Thoughts for this section

Final thoughts summarize the overarching importance of thoroughly addressing all requirement-related gaps and ambiguities to ensure a successful system outcome.

This system concept is timely and technologically grounded. However, to ensure a robust implementation:

- **Complete the functional scope** (especially user management, access control, error scenarios).
- **Clarify technical parameters** to avoid ambiguities.
- **Pre-plan fallback logic** for GPS/camera failures.
- **Prioritize privacy and performance** from day one.

The requirement analysis for the mobile-based attendance management system identified key functionalities such as authentication, user management, session scheduling, notifications, and reporting as essential for robust implementation. Gaps in geofence definitions, model specifications, and device compatibility were noted, with strategies proposed for clarification. The system is technically feasible with current mobile technologies, assuming careful management of resources, permissions, and background services, especially for integrating facial recognition and geofencing modules.

II. Identification of Inconsistencies, Ambiguities, and Missing Information

This section aims to identify and analyze potential inconsistencies, ambiguities, and missing information in the proposed mobile-based attendance management system that integrates geofencing and facial recognition technologies. The objective is to ensure a comprehensive understanding of the system requirements and to highlight areas that require clarification or further development.

1. Identified Inconsistencies

1.1. System Architecture Definition

- **Issue:** The system architecture (cloud-based, on-premises, or hybrid) is not clearly defined.
- **Implication:** Lack of architectural clarity may lead to improper infrastructure planning, scalability issues, and inconsistent system performance.

- **Resolution:** Implement a hybrid architecture with core authentication processing on secure cloud servers while allowing for basic offline functionality on devices. Document the complete architecture with component diagrams.

1.2. Geofencing Parameters

- **Issue:** The specific parameters defining the geofence boundaries and accuracy requirements are not clearly outlined.
- **Implication:** Without precise geofence definitions, the system may inaccurately determine whether a user is within the designated area, leading to erroneous attendance records.
- **Resolutions:** acceptable radius will be defined for the geofence, with configurable parameters based on facility size, and implement a buffer zone system that provides warnings when users are near boundaries. Also environmental factors that may affect GPS accuracy, such as urban canyons or indoor settings are also considered.

1.3. Facial Recognition Accuracy

- **Issue:** The system's expected accuracy rate for facial recognition and integration with geofencing is not specified.
- **Implication:** Lack of defined accuracy metrics can result in unreliable attendance verification, especially in diverse real-world scenarios.
- **Resolutions:** Establish minimum 95% confidence threshold for positive identification with false positive rate below 0.1%, and create a clear workflow where facial recognition is triggered only after successful geofence verification.

1.4. Handling Multiple Attendance Events within Geofence

- **Issue:** The system needs to capture each and every user attendance during a single day without any delay. Since the whole day is taken into account the time each student takes attendance should also be mentioned.
- **Resolutions:** Need to ensure that there is time period allotted to avoid such complications.

2. Identified Ambiguities

2.1. User Consent and Data Privacy

- **Issue:** The process for obtaining user consent for collecting and processing biometric data is not detailed.
- **Implication:** Ambiguity in consent procedures may lead to non-compliance with data protection regulations and potential user distrust.
- **Resolutions:** Implement a transparent consent mechanism that informs users about data collection, usage, storage, and their rights. Ensure compliance with relevant data protection laws like GDPR, CCPA, and BIPA.

2.2. Relationship Between Authentication Methods

- **Issue:** The relationship between geofencing and facial recognition components is not clearly defined.
- **Implication:** Without a clear authentication workflow, the system may implement inconsistent verification procedures, leading to security gaps or unnecessarily complex user experiences.
- **Resolutions:** Document a sequential verification process where geofencing triggers facial recognition requirement with clear workflow diagrams.

2.3. Data Storage and Security

- **Issue:** The system's approach to storing and securing sensitive biometric data is not clearly defined.
- **Implication:** Unclear data storage practices may expose the system to security vulnerabilities and data breaches.
- **Resolutions:** secure on-server processing with encrypted transmission of biometric data, and data classification system with appropriate encryption levels for each type will be implemented (e.g., AES-256 for biometric templates).

2.4. Offline Functionality Requirements

- **Issue:** Whether the system works offline or requires continuous internet connectivity is ambiguous.
- **Implication:** Unclear connectivity requirements may result in system failures in areas with poor network coverage, affecting user experience and reliability.
- **Resolutions:** a hybrid system with essential functions available offline and background synchronization when connectivity is restored.

2.5. User Roles and Permissions

- **Issue**: While the system mentions students, instructors, and administrators, it does not fully define the permissions and access levels for each role. For example, can instructors view the attendance records of other instructors? How are new users added and assigned roles?
- **Resolutions:** Provide a detailed role-based access control matrix specifying which functions and data are accessible to each user role.

3. Missing Information

3.1. Detailed System Architecture Diagram:

- **Observation:** The available documentation lacks a clear system architecture diagram that illustrates the components of the system (mobile app, server, database, APIs), their interactions, and the flow of data.
- **Resolutions:** Develop Create a system architecture diagram to provide a high-level overview of the system's structure and components.

3.2. Integration with Existing School Systems

- **Observation:** It is unclear how the new attendance management system will integrate with existing school systems (e.g., student information systems, grading systems).
- **Resolutions:** Define the integration requirements with existing school systems. Specify the data exchange formats and APIs to be used.

3.3. Handling of Exceptional Scenarios

- **Observation:** There is no mention of how the system will handle scenarios such as facial recognition failures, device unavailability, or network issues.
- **Resolutions:** Implement a tiered fallback system with alternative authentication methods and supervisor override, and develop adaptive functionality that gracefully degrades features in poor connectivity.

4.4 Scalability and Performance Requirements

- **Observation:** There are no specified scalability requirements. How many users will the system support? What are the performance targets for attendance capture and data retrieval?
- **Resolutions:** specific scalability and performance requirements for the system will be designed, including the maximum number of concurrent users, attendance capture time, and data retrieval latency.

4.5 Error Handling and Reporting Mechanisms

- **Observation:** The system does not describe how it will handle errors (e.g., GPS signal loss, facial recognition failure, server downtime) or how errors will be reported to users and administrators.
- **Resolutions:** Describe the error handling and reporting mechanisms of the system. Provide examples of error messages and reporting procedures.

4.6. Backup and Recovery Plan

- **Observation:** The system lacks a backup and recovery plan in case of data loss or system failure.
- **Resolutions:** Document the backup and recovery plan, including backup frequency, storage location, and recovery procedures.

The identification of these inconsistencies, ambiguities, and missing information is crucial for the successful development and implementation of the mobile-based attendance management system.

Addressing these areas will enhance system reliability, user trust, and compliance with legal standards. By implementing the recommended solutions, we can develop a robust attendance management system that effectively utilizes geofencing and facial recognition technologies while maintaining security, compliance, and positive user experience.

III. Prioritization of Requirements

Prioritizing requirements is fundamental to ensuring the system meets its objectives effectively. This process helps allocate resources efficiently while maintaining the integrity of core functionalities. We focused on the essential aspects of real-time attendance tracking, security, and accuracy, ensuring that the most critical features receive top priority.

To achieve this, we utilized various elicitation techniques, including surveys, interviews, focus groups, prototyping, and reverse engineering, to gather stakeholder input and system expectations. Once the requirements were compiled, we applied the MoSCoW prioritization method, which categorizes requirements into four levels based on importance and feasibility:

- ✧ **Must-Have**: These features are essential for the system's core functionality. Without them, the project cannot fulfill its objectives.
- ✧ **Should-Have**: These requirements are highly beneficial but not mandatory for the initial deployment. They improve usability and efficiency but are not strictly necessary. An example is the instructor dashboard, which simplifies attendance tracking but does not affect core system operations.
- ✧ **Could-Have**: These features enhance user experience but are not crucial for fundamental functionality. For instance, UI theming (dark mode) improves aesthetics but does not affect attendance tracking.
- ✧ **Won't-Have**: These requirements are either impractical due to current constraints or unnecessary within the current scope. Blockchain-based storage falls into this category because it introduces complexity without immediate benefits.

MoSCoW Prioritization

Requirement	MoSCoW Priority	Priority(Feasibility)	Rationale and Support for Project Goals
Facial-recognition check-in	Must Have	High	Core to automatic attendance capture; enables accurate, real-time identity verification. Identified by stakeholders as central to the concept. Requires camera/ML.
Geofencing with radius control	Must Have	High	Ensures students are within the classroom location at check-in, preventing fraudulent remote attendance and supporting accuracy and security.
Lecturer activates attendance tracking within the goefence area	Must Have	High	Defines when check-in opens; ensures only authorized sessions allow attendance capture, maintaining control and security of the process.
Face spoof (liveness) detection	Must Have	Medium	Protects against impersonation and unauthorized check-ins, directly supporting system security and integrity of attendance data.
End-to-end check-in latency <= 5s	Must Have	High	Ensures efficient check-in by limiting processing time for students

Secure user authentication and roles	Must Have	High	Requires all users (students/instructors) to log in with credentials. Ensures only authorized users can mark or manage attendance, aligning with security and privacy needs.
Data encryption (at rest and transit)	Must Have	High	Encrypting biometric and location data mitigates privacy risk and complies with legal standards, directly supporting security and stakeholder trust.
GDPR compliance and consent handling	Must Have	Medium	Addresses stakeholder privacy concerns by managing user consent and adhering to regulations; essential for ethical handling of sensitive data.
Attendance dashboard (instructor UI)	Should Have	High	Provides instructors a real-time view of attendance and alerts. Improves usability and instructor experience, but not strictly necessary in MVP.
Concurrent check-in performance	Should Have	Medium	Ensures system remains responsive under load (e.g. many students checking in simultaneously). Supports real-time performance and scalability.
Filter attendance by course/date/team	Should Have	High	Enables instructors to sort and view attendance by course, date, or individual student, enhancing data analysis and reporting capabilities.

Manual attendance override	Should Have	High	Allows lecturers to flag/unflag attendance records to handle discrepencies
Offline check-in mode	Could Have	Medium	Allows attendance marking without Internet (sync later). Useful for reliability but not mandatory for initial deployment.
User interface theming (UI themes)	Could Have	High	Improves user experience (e.g. dark mode), but does not affect core attendance function. Considered a lower priority UX enhancement.
Student view attendance history	Could Have	High	Allows students to see their own past attendance records, supporting transparency and self-monitoring.
Blockchain-based storage	Won't Have	Low	Innovative data storage was evaluated but deemed overkill for MVP. Not prioritized as it would add complexity without essential benefits in current scope.

This prioritization framework helps **streamline development efforts**, ensuring essential functionalities are addressed first while allowing room for future enhancements. By structuring features based on importance and feasibility, the project remains **efficient and adaptable to technological and institutional needs**.

IV.Classification of Requirements

To clarify scope and ensure comprehensive coverage, we separated all gathered requirements into functional and non-functional categories. Functional requirements define what the system must do (features and behaviors), while non-functional requirements define system qualities (performance,

security, usability, etc.). This distinction helps structure our development efforts and validates that both feature completeness and quality attributes meet our objectives.

Functional vs. Non-Functional Classification

Requirement	Classification	Description
Facial-recognition check-in	Functional	The app uses the front camera and ML model to verify a student's face for attendance in real time.
Geofencing (location-based verification)	Functional	The system checks device GPS to confirm the student is within the classroom boundary before allowing check-in.
Lecturer activates attendance tracking within the goefence area	Functional	Allows the lecturer to trigger the start of the attendance session by activating the attendance tracking within the virtual boundary
Attendance data storage and reporting	Functional	System must log each check-in and allow queries/reports, including database schemas for attendance records.
Instructor dashboard (attendance overview)	Functional	Displays real-time attendance lists and alerts for instructors based on recorded check-ins.
Offline mode for attendance	Functional	Allows check-in when offline by caching data locally and syncing when online.
Face spoof/liveness detection	Functional	Analyzes captured face for liveness (e.g. blinking) to prevent use of photographs.

User account management (login, roles)	Functional	Allows students and staff to register, log in, and have different permissions.
UI/UX design and ease of use	Functional	The interface will be intuitive with minimal clicks for check-in and clear feedback.
Manual attendance overide	Functional	Allow lecturers to flag/unflag attendance to correct errors
Data encryption (biometric & GPS data)	Non-Functional	All sensitive data at rest or in transit will be encrypted, protecting privacy.
Access control and authentication security	Non-Functional	Robust authentication and session management to prevent unauthorized access.
Data protection and compliance (GDPR, etc.)	Non-Functional	System design must comply with privacy laws by managing user consent and data usage.
Recognition accuracy (≥95%)	Non-Functional	The face recognition model should achieve a high accuracy rate to minimize errors.
System response time (low latency)	Non-Functional	Ensures near real-time check-in processing to meet performance goals.
End-to-end check-in latency <= 5s	Non-Functional	Limits processing time for attendance check-in to improve system responsiveness
Concurrency/performance (scalability)	Non-Functional	Supports many simultaneous check-ins without delay.
Reliability/availability (uptime target)	Non-Functional	Highly available during class hours with fallback when connectivity is poor.
Platform compatibility (Android)	Non-Functional	The app should support major mobile platforms for broad accessibility.

Power/battery optimization	Non-Functional	Minimizes battery drain from GPS/camera use for user convenience.
Maintainability/modularity	Non-Functional	Clean, modular code structure to facilitate updates and feature expansions.

Defining these categories helps **structure system development efficiently**, allowing developers to focus on **feature completeness and long-term reliability**. This classification ensures that both functional and non-functional aspects of the system are well-integrated, contributing to a **robust and user-friendly attendance management solution**.

V. Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the Mobile-Based Attendance Management System. It details both functional and non-functional requirements, system constraints, and design specifications for an attendance tracking solution that leverages facial recognition and geofencing technologies. This document serves as a definitive reference for stakeholders, developers, and quality assurance teams throughout this development lifecycle.

1.2 Scope

The Mobile-Based Attendance Management System aims to revolutionize traditional attendance tracking methods in educational institutions by leveraging cutting-edge technologies. The system will:

- Enable real-time attendance marking through GPS geofencing
- Provide secure verification using facial recognition technology

- Offer cross-platform accessibility through mobile (Android/iOS)
- Implement role-based access control for students, instructors, and administrators
- Generate comprehensive attendance reports and analytics
- Deliver timely notifications through email and push notifications
- Integrate seamlessly with existing institutional information systems through RESTful APIs
- Support offline functionality for areas with limited connectivity
- Ensure data privacy and security compliant with relevant regulations

1.3 Definitions, Acronyms, and Abbreviations

Term/Acronym	**Definition**
SRS	Software Requirements Specification
GPS	Global Positioning System
FR	Facial Recognition
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
UI	User Interface
UX	User Experience
ML	Machine Learning
DBMS	Database Management System
RDS	Relational Database Service (AWS)
OTP	One-Time Password
GDPR	General Data Protection Regulation
FERPA	Family Educational Rights and Privacy Act
JWT	JSON Web Token
MVC	Model-View-Controller
SLA	Service Level Agreement
CI/CD	Continuous Integration/Continuous Deployment

1.4 Overview

This SRS document is structured to provide a comprehensive understanding of the Mobile-Based Attendance Management System. It begins with an introduction and general description of the system, followed by detailed functional and non-functional requirements. The document includes use case scenarios for each requirement to illustrate system behavior, and appendices containing supplementary information.

The intended audience includes:

- Development team members
- Quality assurance testers
- Project managers
- Educational institution stakeholders
- System administrators

2. Overall Description

2.1 Product Perspective

The Mobile-Based Attendance Management System is designed as a self-contained solution with integration capabilities for existing educational information systems. It operates within the broader ecosystem of educational technology, complementing student information systems, learning management systems, and administrative tools.

The system architecture follows a client-server model with:

- Client-side applications built using Flutter for cross-platform compatibility
- Server-side components providing RESTful API services
- Cloud-based database and authentication services
- Integration of ML services for facial recognition
- GPS and camera hardware integration for location and biometric verification

2.2 Product Functions

2.3. Operating Environment

The Mobile-Based Attendance Management System operates within the following technical environment:

2.3.1 Client Applications

- **Mobile Application:**
  - Android
  - iOS
  - Flutter
  - Access to camera and GPS hardware

2.3.2 Server Infrastructure

- **Backend Services:** Cloud-based
- **Database:** Firebase Firestore
- **Authentication:** Firebase Authentication
- **Storage:** Firebase Storage
- **ML Services:** Firebase ML Kit

2.3.3 Network Requirements

- Internet connectivity
- GPS capability on mobile devices
- Firewall configurations allowing necessary ports/protocols

2.4 Design and Implementation Constraints

The development and deployment of this Mobile-Based Attendance Management System are subject to the following constraints:

2.4.1 Regulatory Constraints

- Must comply with GDPR for data privacy
- Must adhere to FERPA requirements
- Must follow local data protection laws in operating regions
- Special consideration for biometric data storage and processing
- Attendance data must be retained according to institutional policies

2.4.2 Technical Constraints

- Frontend development limited to Flutter and Dart
- Facial recognition must work with various lighting conditions
- Database design must support both SQL and NoSQL options
- System must function in environments with intermittent connectivity
- Mobile app must minimize battery consumption during GPS and camera usage
- Facial recognition check-in must complete within a specified amount of time
- GPS accuracy can vary by device and location

2.4.3 Business Constraints

- Development timeline of 2 months
- Budget limitations affecting hosting and third-party service options
- Must provide migration path from existing attendance systems
- System must be maintainable by institution's IT staff

2.4.4 Security Constraints

- Biometric templates must be securely encrypted
- Data encryption for all personal information
- Secure authentication with multi-factor options
- Regular security audits and penetration testing
- Comprehensive access logging and monitoring

2.5 Assumptions and Dependencies

2.5.1 Assumptions

- Students possess GPS-enabled smartphones with cameras (Android or iOS)
- Educational institutions have reliable internet connectivity
- Faculty members can access computers or smartphones during classes
- IT support is available for system deployment and maintenance
- Attendance policies are clearly defined by the institution
- Users consent to biometric data collection

2.5.2 Dependencies

- Availability of Firebase or AWS cloud services
- Flutter SDK compatibility with target platforms
- ML libraries for facial recognition processing
- Institution's ability to provide necessary server infrastructure
- Availability of technical resources for integration with existing systems
- Cooperation from stakeholders for requirements validation and testing

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Attendance Management

ID	Requirement	Priority	Description
FR2.1	Facial Recognition	High	Students shall be able to mark attendance using facial recognition verification.
FR2.2	GPS Attendance	High	Students shall be able to mark attendance only when physically present within the configured geofence of the class location.
FR2.3	Manual override	Medium	The lectures will manually be able to mark student present if they are not able to .

FR2.4	Attendance Verification	High	The system shall verify and confirm attendance marking with a success notification to the user.
FR2.5	Attendance Window	High	The system shall enforce configurable time windows for marking attendance (e.g. 30 minuets to class end).
FR2.6	Attendance History	Medium	Students shall be able to view their personal attendance history with filtering options.
FR2.7	Attendance Correction	Low	Faculty shall be able to manually correct or overrides attendance records with justification notes.

3.1.2 Course and Class Management

ID	Requirement	Priority	Description
FR3.1	Course Creation	High	Administrators shall be able to create, update, and courses with relevant details.
FR3.2	Class Scheduling	High	Administrators shall be able to schedule classes with date, time, duration, and location information.
FR3.3	Student Enrollment	High	Administrators shall be able to enroll students in courses individually or via batch upload.
FR3.4	Timetable Management	Medium	The system shall provide timetable views for students and faculty based on their enrolled/assigned courses.
FR3.5	Location Management	High	Administrators shall be able to define and manage location geofences for attendance marking.

3.1.3 Reporting and Analytics

ID	Requirement	Priority	Description
FR4.1	Attendance Reports	High	Administrators shall be able to generate attendance reports by class, course, or student.

FR4.2	Export Functionality	Medium	The system shall allow exporting reports in multiple formats (PDF, CSV, Excel).
FR4.3	Attendance Statistics	Medium	The system shall provide statistical analysis of attendance patterns with visual representations.
FR4.4	Absence Tracking	Medium	The system shall identify and highlight students with attendance below configurable thresholds.
FR4.5	Custom Reports	Low	Administrators shall be able to create and save custom report templates with selected parameters.

3.1.4 Notification System

ID	Requirement	Priority	Description
FR5.1	Attendance Confirmation	High	The system shall send push notifications confirming successful attendance marking.
FR5.2	Absence Alerts	Medium	The system shall notify students about missed classes at configurable intervals.
FR5.3	Attendance Reminders	Medium	The system shall send reminders before scheduled classes based on user preferences.
FR5.4	System Announcements	Low	Administrators shall be able to send system-wide announcements to all users or specific groups.
FR5.5	Notification Preferences	Medium	Users shall be able to configure their notification preferences by type.

3.1.6 System Administration

ID	Requirement	Priority	Description
FR6.1	User Management	High	Administrators shall be able to create, update, deactivate, and delete user accounts.

FR6.2	Role Management	High	Administrators shall be able to define and assign roles with specific permissions.
FR6.3	System Configuration	Medium	Administrators shall be able to configure system parameters and thresholds.
FR6.4	Audit Logging	Medium	The system shall maintain audit logs of critical actions for security and troubleshooting.
FR6.5	Data Backup	High	The system shall support scheduled backups of all critical data.
FR6.6	Biometric Template Management	High	Administrators shall be able to manage and reset biometric templates when necessary.

3.1.7 Integration Capabilities

ID	Requirement	Priority	Description
FR7.1	API Access	Medium	The system shall provide RESTful API endpoints for integration with external systems.
FR7.2	Data Import	Medium	The system shall support importing user and course data from CSV or Excel files.
FR7.3	Single Sign-On	Low	The system shall support integration with institutional SSO solutions.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

ID	Requirement	Description	Metric
NFR1.1	Concurrent Users	The system shall support at least 500 concurrent users without performance degradation.	Response time < 3 seconds
NFR1.2	Response Time	The system shall provide attendance confirmation within 3 seconds under normal network conditions.	95% of requests

NFR1.3	Database Performance	The system shall handle at least 100 database transactions per second.	Latency < 500ms
NFR1.4	Mobile Application Launch	The mobile application shall launch within 5 seconds on supported devices.	90% of launches
NFR1.5	Report Generation	The system shall generate standard reports within 10 seconds.	For reports covering up to 1000 records
NFR1.6	Facial Recognition Speed	The facial recognition process shall complete within 5 seconds.	90% of verification attempts

3.2.2 Security Requirements

ID	Requirement	Description	Verification
NFR2.1	Data Encryption	All sensitive data shall be encrypted both in transit and at rest.	Security audit
NFR2.2	Authentication Security	The system shall enforce password complexity rules and account lockout after failed attempts.	Penetration testing
NFR2.3	Session Management	User sessions shall expire after 30 minutes of inactivity.	User acceptance testing
NFR2.4	Authorization	The system shall implement role-based access control for all resources.	Security audit
NFR2.5	Security Auditing	The system shall log all authentication attempts and critical operations.	Log review
NFR2.6	Biometric Security	Biometric templates shall be stored using industry-standard encryption (AES-256).	Security audit

3.2.3 Reliability Requirements

ID	Requirement	Description	Metric

NFR3.1	Availability	The system shall maintain 99.9% uptime during academic hours.	Monthly uptime report
NFR3.2	Data Backup	The system shall perform daily backups with retention for 30 days.	Backup verification
NFR3.3	Failure Recovery	The system shall recover from failures within few hours say 4	Disaster recovery testing
NFR3.4	Offline Operation	The mobile application shall support offline attendance marking with synchronization when connectivity is restored.	User acceptance testing
NFR3.5	Fault Tolerance	The system shall handle input errors gracefully with appropriate user feedback.	Error handling review

3.2.4 Usability Requirements

ID	Requirement	Description	Verification
NFR4.1	User Interface	The user interface shall follow Material Design guidelines for consistency.	UI review
NFR4.2	Accessibility	The system shall comply with WCAG 2.1 Level AA standards.	Accessibility testing
NFR4.3	Learnability	New users shall be able to use core functions without training.	Usability testing
NFR4.4	Documentation	The system shall provide context-sensitive help and documentation.	Documentation review

3.2.5 Maintainability Requirements

ID	Requirement	Description	Verification
NFR5.1	Code Quality	The codebase shall follow industry-standard coding conventions.	Code review

NFR5.2	Documentation	All code shall be documented with inline comments and API documentation.	Documentation review
NFR5.3	Modularity	The system shall be designed with modular components for easier maintenance.	Architecture review
NFR5.4	Configurability	System parameters shall be configurable without code changes.	Configuration testing
NFR5.5	Versioning	The system shall maintain proper versioning for all components.	Version control audit

3.2.6 Scalability Requirements

ID	Requirement	Description	Metric
NFR6.1	Horizontal Scaling	The system shall support horizontal scaling for increased load.	Performance testing
NFR6.2	Multi-Campus Support	The system shall support multiple campuses with distinct configurations.	Multi-tenancy testing
NFR6.3	Database Scaling	The database shall scale to support at least 500 students.	Database performance test
NFR6.4	Growth Support	The system shall accommodate 20% annual growth without architectural changes.	Capacity planning
NFR6.5	API Scalability	API endpoints shall handle at least 100 requests per second.	Load testing

3.3 External Interface Requirements

3.3.1 User Interfaces

ID	Requirement	Description	Priority
UI1.1	Mobile Application	Native-feeling mobile application with responsive design for various screen sizes.	High

UI1.2	Administrator Dashboard	Comprehensive dashboard for system administration and analytics.	Medium
UI1.3	Faculty Portal	Specialized interface for faculty to manage classes and attendance.	High
UI1.4	Accessibility	All interfaces shall be accessible to users with disabilities.	Medium

3.3.2 Hardware Interfaces

ID	Requirement	Description	Priority
HI1.1	Camera Integration	Integration with device camera for facial recognition and QR code scanning.	High
HI1.2	GPS Integration	Integration with device GPS sensors for location verification.	High
HI1.3	Biometric Sensors	Integration with facial recognition.	High

3.3.3 Software Interfaces

ID	Requirement	Description	Priority
SI1.1	ML API	Integration with machine learning APIs for facial recognition.	High
SI1.2	SIS Integration	Integration with Student Information Systems through API.	Medium
SI1.3	Email Services	Integration with SMTP services for email notifications.	High
SI1.4	Push Notification	Integration with FCM (Firebase Cloud Messaging)	High
SI1.5	Calendar Systems	Integration with institutional calendars.	Low

SI1.6	Authentication Services	Integration with OAuth	Medium

VI. Validate Requirements with Stakeholders

Following the completion of the requirements gathering phase for the Mobile-Based Attendance Management System Based on Geofencing and Facial Recognition, a structured stakeholder validation exercise was conducted to ensure that the identified requirements accurately reflect user needs, are technically and ethically feasible, and align with the project's objectives.

1. Aim of Validate Requirements with Stakeholders

Validation Requirements with Stakeholders session in the **Requirement Analysis** aims to ensure that all gathered and documented requirements are:

- **Accurate**: They truly reflect what the stakeholders (clients, users, business owners) need.
- **Complete**: No critical requirements are missing.
- **Understandable:** The requirements are clear to all stakeholders, with no ambiguities or confusion.
- **Feasible:** Stakeholders confirm that the requirements are achievable within the available resources, budget, and timeline.
- **Agreed Upon:** There is mutual consensus and formal approval (sign-off), confirming the development team can proceed.

This section helps to **prevent costly misunderstandings** later in the project by aligning the expectations of stakeholders with what will actually be built.

Before we proceed, let us have a reminder on what were the **expected outcome** and what are our **requirements** in the Implementation of a **Mobile-Based Attendance Management System Based on Geofencing and Facial Recognition**:

2. Expected Outcomes:

A fully functional mobile application that:

- Supports real-time attendance check-in using facial recognition.
- Uses geofencing to validate that student are within a specified classroom location before check-in is permitted.
- Ensures that the entire check-in process takes no more than 5 seconds per student.
- Integrates a face capture and recognition module using machine learning libraries
- Provides secure storage and comparison of facial biometric data.
- Integrates GPS services to define virtual classroom boundaries.
- Restricts check-in functionality to students within this boundary.
- Allows instructors to view real-time attendance data.
- Offers functionalities such as filtering by course, date, or student.
- Enables students to view their attendance history and participation status for each registered course.

3. Validation Activities Conducted

To achieve this, the following methods were used:

- **Stakeholder meetings** with lecturers and students to present the documented functional and non-functional requirements.
- **Walkthroughs of use cases and interface mockups** to make the requirements understandable to non-technical participants.
- **Feedback forms and discussions** to collect acceptance, rejections, and suggestions for improvement.
- **Consensus sessions** to prioritize features and finalize requirement lists.

4. Key Validation Results

Stakeholders' Expectations	Validated Requirements	Stakeholder Concerns/Feedback
Real-time facial recognition for attendance	Facial-recognition check-in	Students raised **privacy concerns** about biometric data usage. A consent-based approach was recommended.
Geofencing to confirm physical presence	Geofencing location verification	Students questioned **GPS accuracy** in some campus areas. Lecturers requested a **manual override option**.
Fast attendance check-in (<5s)	Low-latency processing	Fully supported by all stakeholders. Seen as critical for practical use.
Use of machine learning for face recognition	ML facial recognition module	Admins requested **bias mitigation** strategies and **model transparency**.
Secure biometric data storage	Data encryption, GDPR compliance	Strong support, with emphasis on **data retention policies** and **student data rights**.
Instructor access to real-time attendance	Instructor dashboard	Fully endorsed by lecturers for improving class management.
Filtering and reporting features	Filtering by course/date/student	Approved; suggestions made to include **data export (CSV/PDF)** options.
Student access to attendance records	Student dashboard	Supported by students; requests made to **include absence notifications**.

5. Final Validation Summary

- **Accuracy**: Requirements accurately capture the expectations and real-world needs of end users.

- **Completeness**: All core functionalities and constraints are well-represented in the documented requirements.
- **Clarity**: Use of mockups and walkthroughs helped clarify all system features for stakeholders.
- **Feasibility**: Some technical concerns (e.g., GPS accuracy, facial recognition bias) were noted and mitigation strategies proposed.
- **Agreement**: All primary stakeholder groups formally approved the validated requirements with minor adjustments integrated.

5. Adjustments Made Based on Stakeholder Feedback

- Implementing a **user consent flow** for biometric data usage.
- Adding a **manual override function** for attendance in exceptional cases.
- Incorporating **bias monitoring and fairness auditing** in the face recognition system.
- Allowing **data export** and **student notifications** for improved usability.

The validation process successfully confirmed that the requirements for the attendance system are both stakeholder-approved and technically viable. Incorporating feedback from this phase ensures that the system will not only meet user needs but also gain trust and adoption across all intended user groups.

Conclusion

The requirement analysis phase serves as the backbone of our mobile-based attendance management system project. Through this task, we have systematically examined both functional and non-functional requirements to ensure they align with user expectations and project objectives. By clearly defining system capabilities, user roles, and constraints, this analysis provides a well-informed foundation for design, development, and testing. It also helps to preempt implementation issues and ensures that all stakeholders have a shared understanding of the system's goals and scope

VII. References

- Trackobit. "Key Features & Benefits of a Geofencing Attendance System." https://trackobit.com/blog/geofencing-attendance-system-work-and-benefits
- ResearchGate. "Survey of Evaluation Metrics in Facial Recognition Systems." https://www.researchgate.net/publication/372232460_Survey_of_Evaluation_Metrics_in_Facial_Recognition_Systems
- Outside GC. "Biometric Privacy Laws." https://www.outsidegc.com/blog/biometric-data-protection-a-growing-trend-in-state-privacy-legislation
- Information Commissioner's Office (ICO). "How do we keep biometric data secure?" https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/lawful-basis/biometric-data-guidance-biometric-recognition/how-do-we-keep-biometric-data-secure/
- ZohoPeople."AttendanceManagementSystem." https://www.zoho.com/people/attendance-management-system.html
- Clockgogo. "How Time Attendance Systems can Aid Management during Unexpected Events."https://clockgogo.com/2023/09/13/how-time-attendance-systems-can-aid-management-during-unexpected-events
- NIST."FaceRecognitionVendorTest (FRVT)."https://www.nist.gov/programsprojects/face-recognition-vendor-test-frvt
- ISO. "ISO/IEC 24745:2022 Information security, cybersecurity and privacy protection — Biometric information protection." https://www.iso.org/standard/75302.html
- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications https://standards.ieee.org/standard/830-1998.html (Also available at: https://ieeexplore.ieee.org/document/720574)
- Android SDK Documentation https://developer.android.com/docs
- Firebase ML Kit Documentation https://firebase.google.com/docs/ml-kit
- OpenCV Documentation https://docs.opencv.org/
- GDPR (EU) 2016/679, General Data Protection Regulation https://gdpr.eu/tag/gdpr/ (Official text: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679)

- FERPA (20 U.S.C. 1232g; 34 CFR Part 99), Family Educational Rights and Privacy Act https://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html (Full text: https://www.ecfr.gov/current/title-34/subtitle-A/part-99)
- Flutter & Dart Official Documentation
  - Flutter: https://flutter.dev/docs
  - Dart: https://dart.dev/guides
- Firebase Documentation https://firebase.google.com/docs
- AWS Documentation https://docs.aws.amazon.com
- ISO/IEC 25010:2011, Systems and software Quality Requirements and Evaluation (SQuaRE) https://www.iso.org/standard/35733.html (Preview available at: https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en)
- Material Design Guidelines https://m3.material.io