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

DEPARTMENT OF COMPUTER ENGINEERING

COURSE CODE:

CEF440

COURSE TITLE:

INTERNET PROGRAMMING (J2EE) AND MOBILE PROGRAMMING

Design and Implementation of a Mobile-Based Attendance Management System Based on Geo fencing and Facial Recognition

BY:

GROUP IV

Members;

S/N	Names	Matricules
1	**ARREY ABUNAW REGINA EBAI**	**FE22A152**
2	**AWA ERIC ANGELO JUNIOR**	**FE22A162**
3	**FAVOUR OZIOMA**	**FE22A217**
4	**OBI OBI ETCHU JUNIOR**	**FE22A291**
5	**VERBURINYUY JERVIS NYAH**	**FE22A324**

Course Intructor:

Dr. Nkemeni Valery

2024/2025 Academic Year

ABSTRACT

The Mobile-Based Attendance Management System Using Geofencing and Facial Recognition (Attendr) addresses critical inefficiencies in traditional academic attendance tracking by automating student verification through biometric authentication and location validation. This project responds to widespread issues of proxy attendance, time-consuming manual processes, and lack of real-time data in educational institutions, as identified through surveys with 86 students and instructors at the University of Buea

Attendr integrates; Facial Recognition, Geofencing, Role-Based Acces:

Developed using React Native (frontend) and FastAPI (backend), the system employs PostgreSQL for GDPR-compliant data storage, encrypting biometric templates via AES-256.

Backend Hosted on Render, Attendr demonstrates how mobile technologies can enhance academic accountability while addressing privacy concerns through opt-in consent and minimal data retention policies. Future work includes NFC fallback support and LMS integration.

Keywords: Attendance system, facial recognition, geofencing, React Native, FastAPI, GDPR compliance.

Table of Contents

List of Abbreviations

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.

CHAPTER I: GENERAL INTRODUCTION

1.1. Background and Context of the project

Accurate attendance management is a cornerstone of academic administration in higher education institutions. Traditional methods such as manual sign-in sheets and RFID cards are fraught with inefficiencies, including susceptibility to human error, proxy attendance, and delayed access to records. The proliferation of mobile devices and advancements in biometric and geospatial technologies present an opportunity to revolutionize this process.

The *Mobile-Based Attendance Management System Based on Geofencing and Facial Recognition* (hereafter referred to as *Attendr*) addresses these challenges by automating attendance tracking through secure facial biometric verification and GPS-based geofencing. This system ensures that only physically present and verified students can check in, thereby enhancing transparency, accountability, and operational efficiency in academic institutions.

1.2. Problem statement

Despite technological advancements, most universities in Cameroon still rely on outdated attendance methods. Key issues identified include:

- Proxy Attendance: Students mark attendance for absent peers.
- Time Inefficiency: Manual processes consume valuable class time.
- Lack of Real-Time Validation: Delays in reporting and analytics hinder proactive interventions.
- Security Vulnerabilities: Unsecured data storage and transmission risks.
- Delayed Analytics: Attendance data often takes hours to compile.

Attendr leverages facial recognition and geofencing to eliminate these issues while complying with data privacy regulations such as GDPR.

1.3. Objectives of the project

1.3.1. General Objective

To design and implement a mobile-based attendance system that automates student check-in using facial recognition and geofencing, ensuring accuracy, security, and real-time monitoring.

1.3.2. Specific Objectives

1. Develop a Facial Recognition Module: Integrate machine learning for identity verification with a target check-in time of ≤5 seconds .
2. Implement Geofencing Validation: Use Google Maps API for GPS tracting to confirm physical presence within classroom.
3. Design Role-Based Interfaces: Tailor dashboards for students, instructors, and administrators.
4. Ensure Data Privacy: Encrypt biometric and location data, and comply with GDPR.
5. Validate System Performance: Achieve 99% uptime and support 1,000+ concurrent users.

1.4. Methodology

The project follows an iterative, user-centered design (UCD) approach combined with Agile development principles to ensure flexibility and stakeholder alignment:

1. Requirement Analysis: Gathered via surveys, interviews, and reverse engineering of existing systems.
2. System Modeling: UML diagrams such as Use Case, Sequence, Deployment to define architecture.
3. Database Design: PostgreSQL with 3NF normalization for entities like User, Attendance, and Geofence.
4. Implementation:
    - Frontend: React Native for cross-platform compatibility.
    - Backend: FastAPI with JWT authentication and SQLAlchemy ORM.
5. Testing Phases:
    - Unit tests for facial recognition accuracy (≥95%) and geofencing precision (20m radius).
    - Integration Testing with API endpoints.
    - User Acceptance Testing (UAT) conducted with 5 students
6. Hosting: Made use of Render To host the back end.

1.5. Significance of the Project

Attendr contributes to:

- Academic Efficiency: Reduces manual workload by 70%.
- Security: Eliminates proxy attendance via dual-factor validation.
- Scalability: Cloud-hosted backend supports multi-institutional deployment.

1.6. Scope and Limitations

1.6.1. Scope

- Platforms: Android 8+ and iOS 12+ (Task 3 Report, p. 6).
- Database: PostgreSQL with encrypted biometric storage (Task 6 Report, p. 12).
- Compliance: GDPR (Task 3 Report, p. 16).

1.6.2. Limitations

1. GPS Accuracy:
    - Signal drift in urban areas requires a 25m buffer.
2. Biometric Consent:
    - Requires student opt-in for facial data collection.
3. Offline Support:
    - Limited to cached check-ins with later sync.

CHAPTER II: LITERATURE REVIEW

2.1. Introduction

In educational institutions, attendance tracking serves as a critical administrative task, influencing performance evaluation, compliance, and academic discipline. Traditionally, attendance has been managed using paper-based registers or simple digital platforms, which often suffer from issues like impersonation (proxy attendance), data loss, lack of real-time access, and inefficiencies in processing and reporting.

With the rapid advancement in mobile technologies and artificial intelligence (AI), particularly **geo-fencing** and **facial recognition**, there is a growing interest in developing smarter, more secure attendance management systems. These technologies offer the potential to automate the attendance process while enhancing accuracy, user experience, and security. This literature review presents the fundamental concepts, examines existing systems and techniques, and highlights the innovations and challenges relevant to the design of a mobile-based attendance system.

2.2. General Concepts on Mobile Attendance Management Systems

2.2.1 Mobile-Based Attendance Tracking

Mobile attendance systems use smartphone features such as GPS, camera, and internet connectivity to mark and verify student presence. These systems typically involve a mobile application where users (students or employees) can check in using their devices. The application sends data to a centralized database accessible to instructors and administrators. Mobile systems are advantageous due to their portability, low hardware requirement, and real-time data availability.

2.2.2 Geo-Fencing Technology

Geo-fencing is a location-based service that uses GPS, RFID, Wi-Fi, or cellular data to trigger a response when a device enters or exits a predefined geographical area. In the context of attendance systems, geo-fencing ensures that students can only mark their attendance if they are within the campus or classroom boundaries. This reduces the risk of remote check-ins and enhances location accuracy. Geo-fencing has been widely used in applications like employee tracking, fleet management, and child safety, and is now being adopted in education.

2.2.3 Facial Recognition

Facial recognition is a biometric authentication method that uses computer vision algorithms to identify individuals based on facial features. The process involves capturing an image of a user's face and comparing it to a database of registered images using AI and machine learning models. This eliminates impersonation and proxy attendance. However, facial recognition also raises concerns regarding user privacy, data security, and algorithmic bias, which must be addressed through ethical design and data protection policies.

2.2.4 Integration of Geo-Fencing and Facial Recognition

Combining geo-fencing and facial recognition creates a multi-layered attendance validation system that ensures both physical presence and identity verification. This dual validation reduces manipulation and provides a high level of security. It also introduces challenges in terms of synchronization, performance optimization, and user onboarding, which must be considered during system design.

2.3. Related Works

Several related systems and studies have informed the development of this project. A review of academic papers, existing mobile applications, and institutional practices was conducted to gather insights into the capabilities and limitations of current attendance tracking systems.

2.3.1 Geo-Attend Pro

Geo-Attend Pro is a mobile app used primarily in corporate settings. It uses GPS to create a geo-fence and allows check-ins only when users are within a defined area. Through reverse engineering, this application was found to have features such as real-time GPS feedback, user prompts, and geo-location enforcement, which closely align with the objectives of this project. However, it lacked integration with facial recognition and had limited flexibility in academic settings.

2.3.2 Institutional Practices and Shortcomings

Manual attendance systems (e.g., sign-in sheets) are still widely used in educational settings. Based on observational studies and surveys, these systems have been criticized for:

- **Wasting class time**
- **Encouraging proxy attendance**
- **Difficulty in managing large class sizes**
- **Manual compilation errors**

Digital systems exist in some institutions, but they either rely only on student IDs or QR codes, which are vulnerable to sharing or misuse.

2.3.3 Empirical Data Collection

Structured **interviews and surveys** were conducted with students and lecturers:

- **Students** expressed a strong preference for a fast and easy check-in system that respects their privacy.
- **Instructors** wanted real-time access to attendance data, automated reports, and features to detect and prevent fraud.
- **Administrators** highlighted the need for secure, tamper-proof data and integration with school databases.

From the feedback, certain **key requirements** emerged:

- Facial recognition check-in with fallback options
- Geo-fencing-based validation to restrict check-ins to classroom zones
- Instructor dashboards with real-time data filtering and reporting
- Privacy-first design with user consent, secure storage, and data minimization

2.4. Partial Conclusion

The literature and existing practices reviewed indicate a significant gap between current attendance systems and the ideal solution desired by users. Traditional methods are unreliable, and even existing mobile systems often fail to provide secure, automated, and integrated attendance tracking.

By combining **geo-fencing** and **facial recognition**, and addressing concerns such as privacy, user acceptance, and performance, the proposed system seeks to offer a modern solution tailored for higher education. It will enhance accuracy, reduce the opportunity for manipulation, and provide real-time reporting all while being user-friendly and ethically designed.

This literature review lays the groundwork for system modeling, emphasizing the importance of user feedback, technical feasibility, and secure design in the development of a robust attendance management system.

CHAPTER III: ANALYSIS AND DESIGN

3.1. Introduction

This chapter outlines the analysis and design phases of the Mobile-Based Attendance Management System Based on Geofencing and Facial Recognition. The goal is to translate the requirements gathered into a structured, implementable solution. The chapter covers the proposed methodology, system design, global architecture, and a step-by-step resolution process to address the challenges identified in traditional attendance systems.

3.2. Proposed Methodology

The project adopts an **iterative, user-centered design (UCD) approach** combined with **Agile development principles** to ensure flexibility and stakeholder alignment. Key methodologies include:

3.2.1 Requirement Analysis

Reviewed functional and non-functional Prioritized features using the **MoSCoW method** below is the summarry of the requirements obtained;

I. Functional Requirements

These describe what the system should do. They were classified under 4 major categories;

1. User Management
2. Attendance Mechanism
3. Attendance Monitoring and Visualization
4. Notifications and Communication

II. Non-functional Requirements

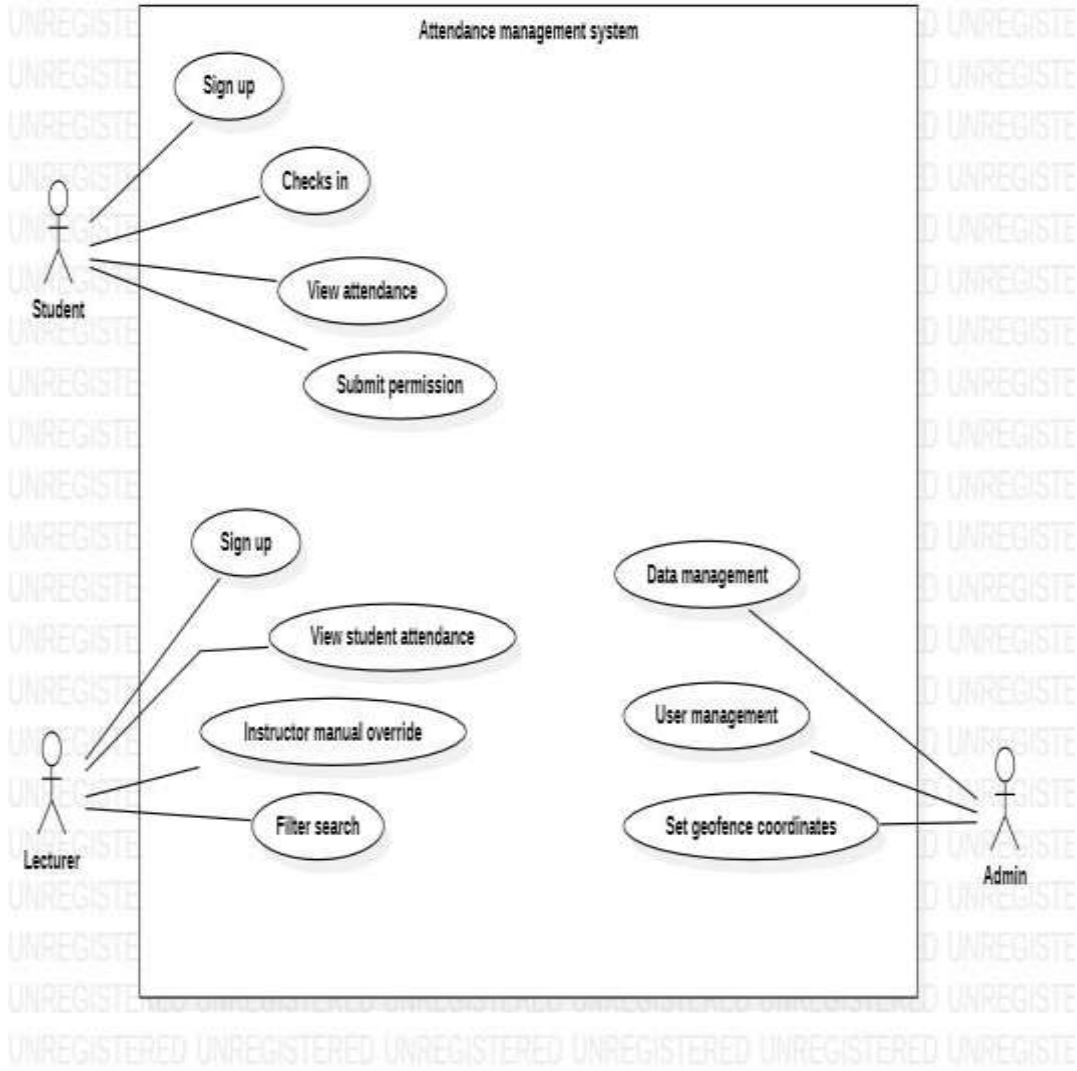
They were classified under 5 major categories;

1. System performance
2. User Interface Requirements
3. Hardware Interface Requirements
4. Software Interface Requirements
5. Communication Interface Requirements

3.2.2 System Modeling

- ✧ **UML Diagrams:** Created use case, class, sequence, context and deployment diagrams to visualize interactions and data flow.

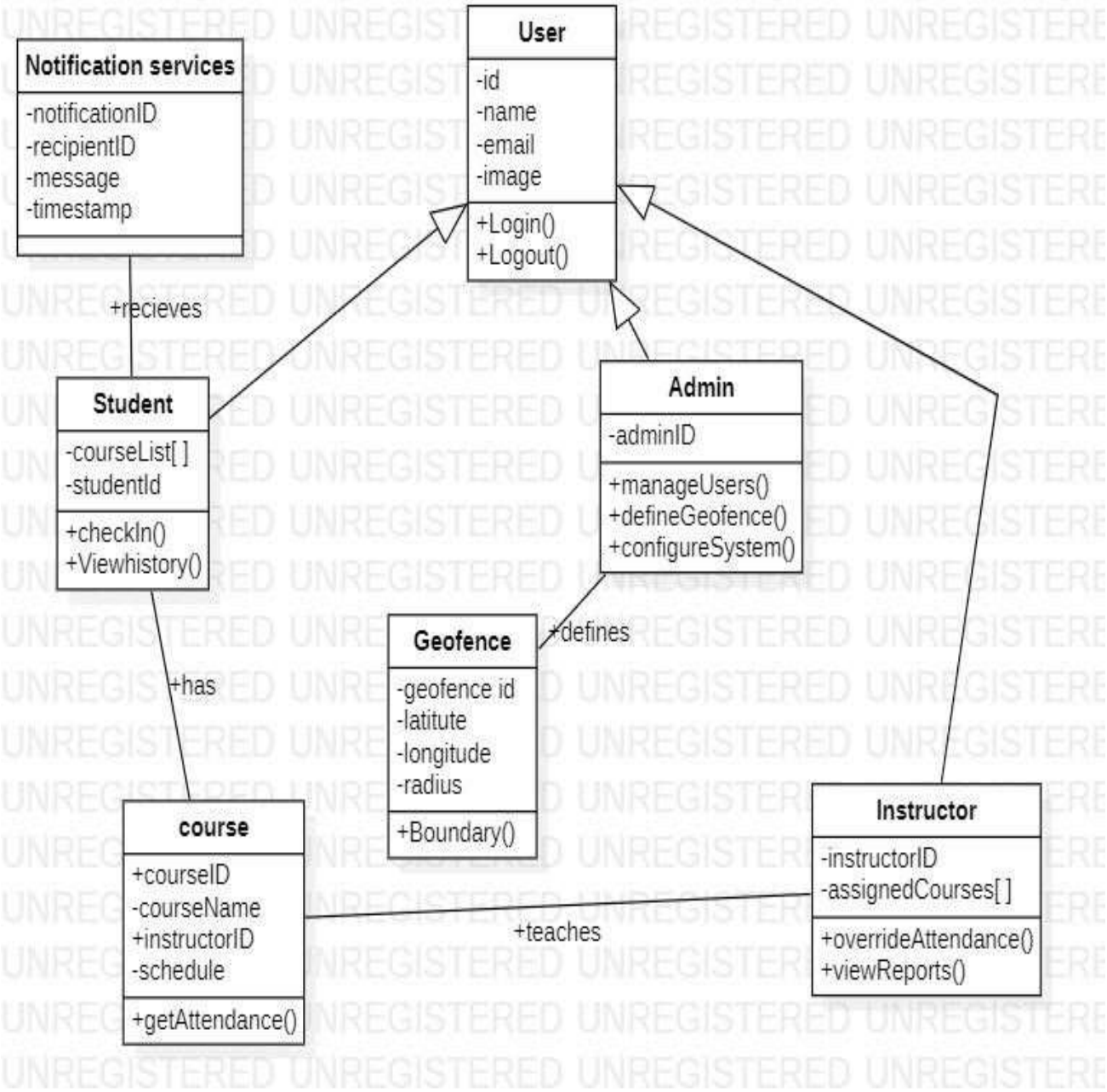
1 Use Case Diagram



Key Use Cases:

1. **Student**:
    - Check-in via face + GPS.
    - View attendance history.
2. **Instructor**:
    - Real-time dashboard.
    - Manual attendance override.
3. **Admin**:
    - Manage geofence boundaries.
    - Audit logs.

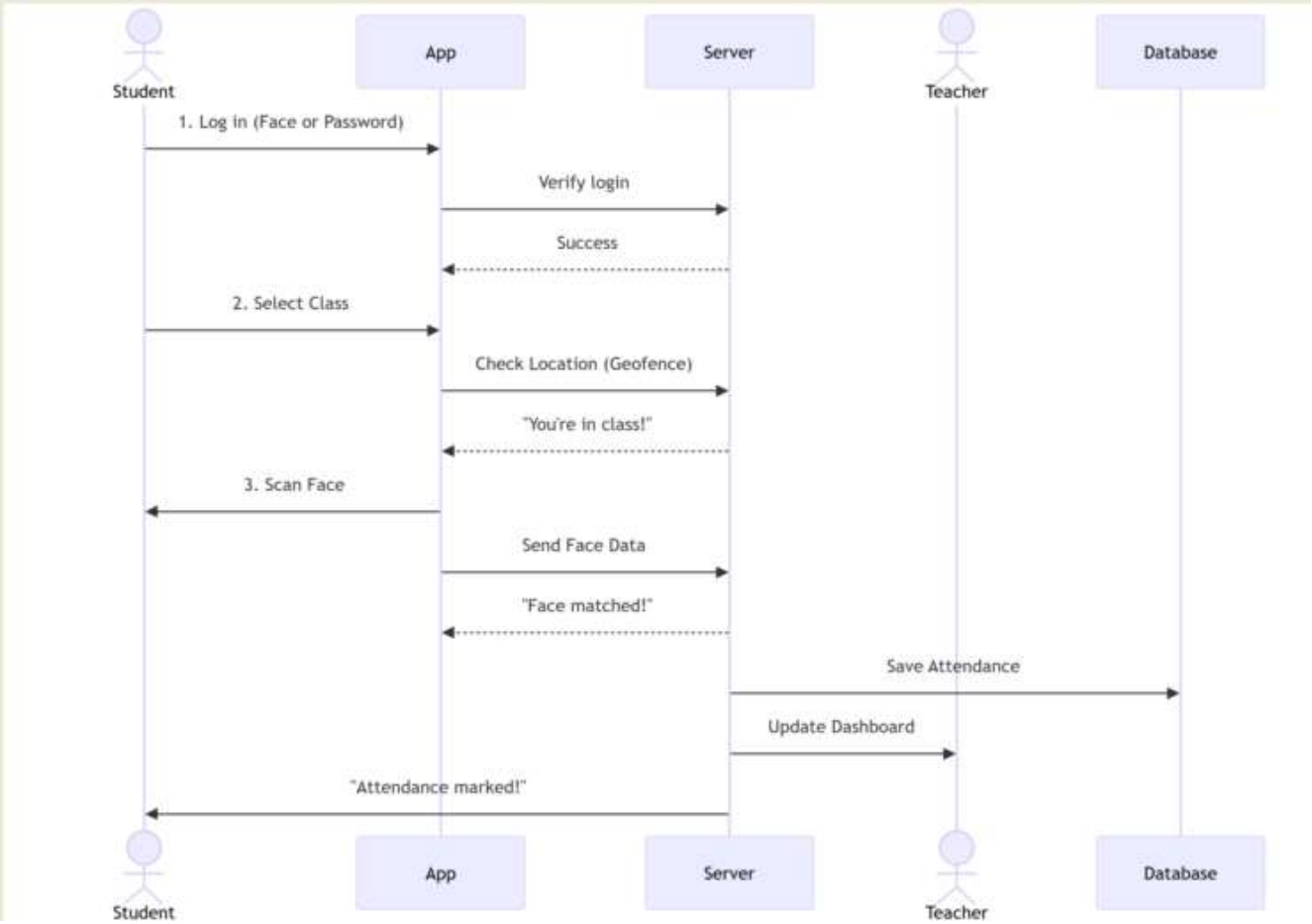
2 Class Diagram



Core Classes:

1. **User** (Abstract):
    - Attributes: user_id, role, facial_biometric_hash.
2. **Attendance**:
    - Linked to Student and ClassSession.
3. **Geofence**:
    - Stores latitude, longitude, radius.

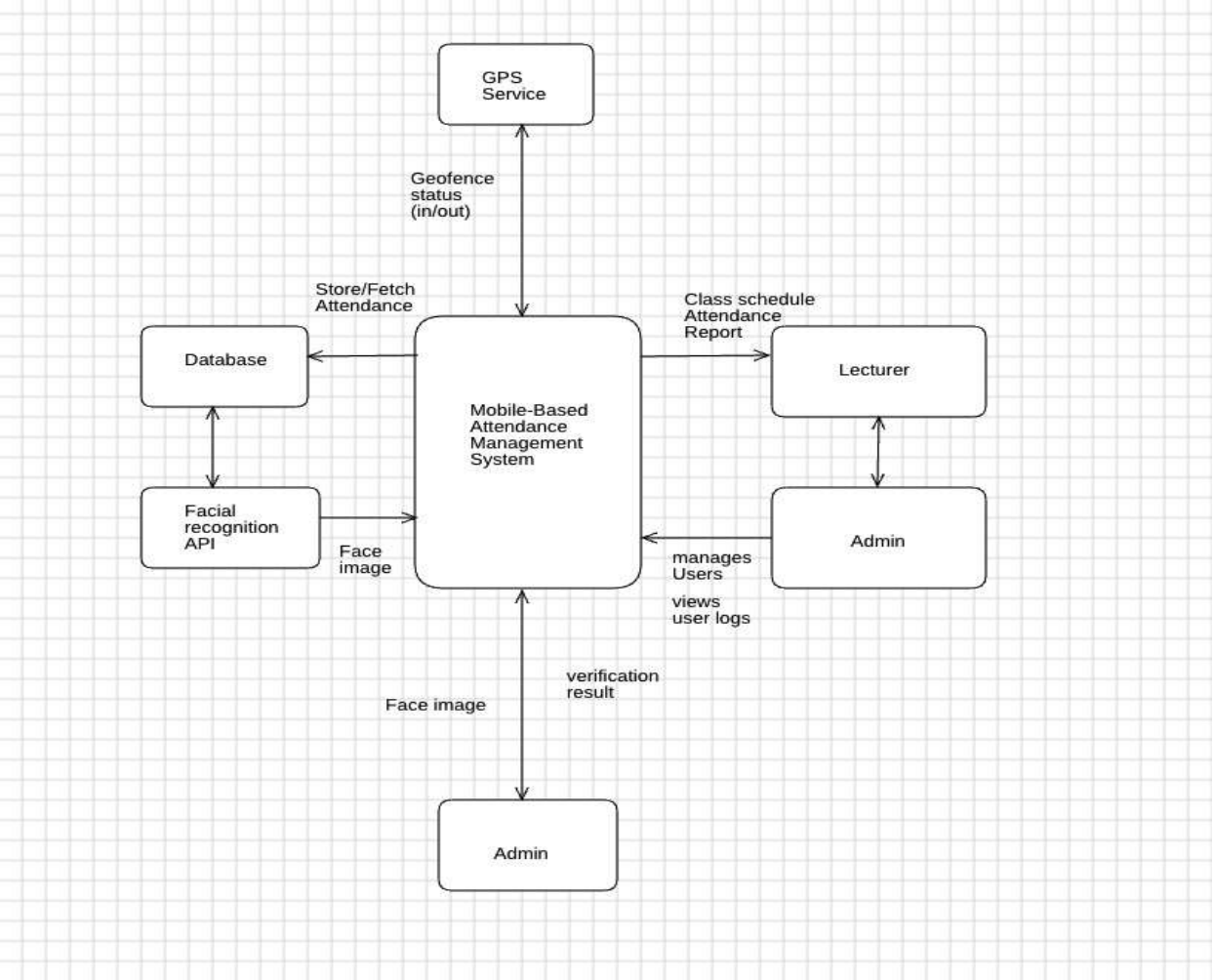
3 Sequence Diagram (Check-In Process)



Steps:

1. Student opens app → captures face.
2. System verifies face (DeepFace) + location (Google Maps).
3. On success: Attendance recorded in PostgreSQL.

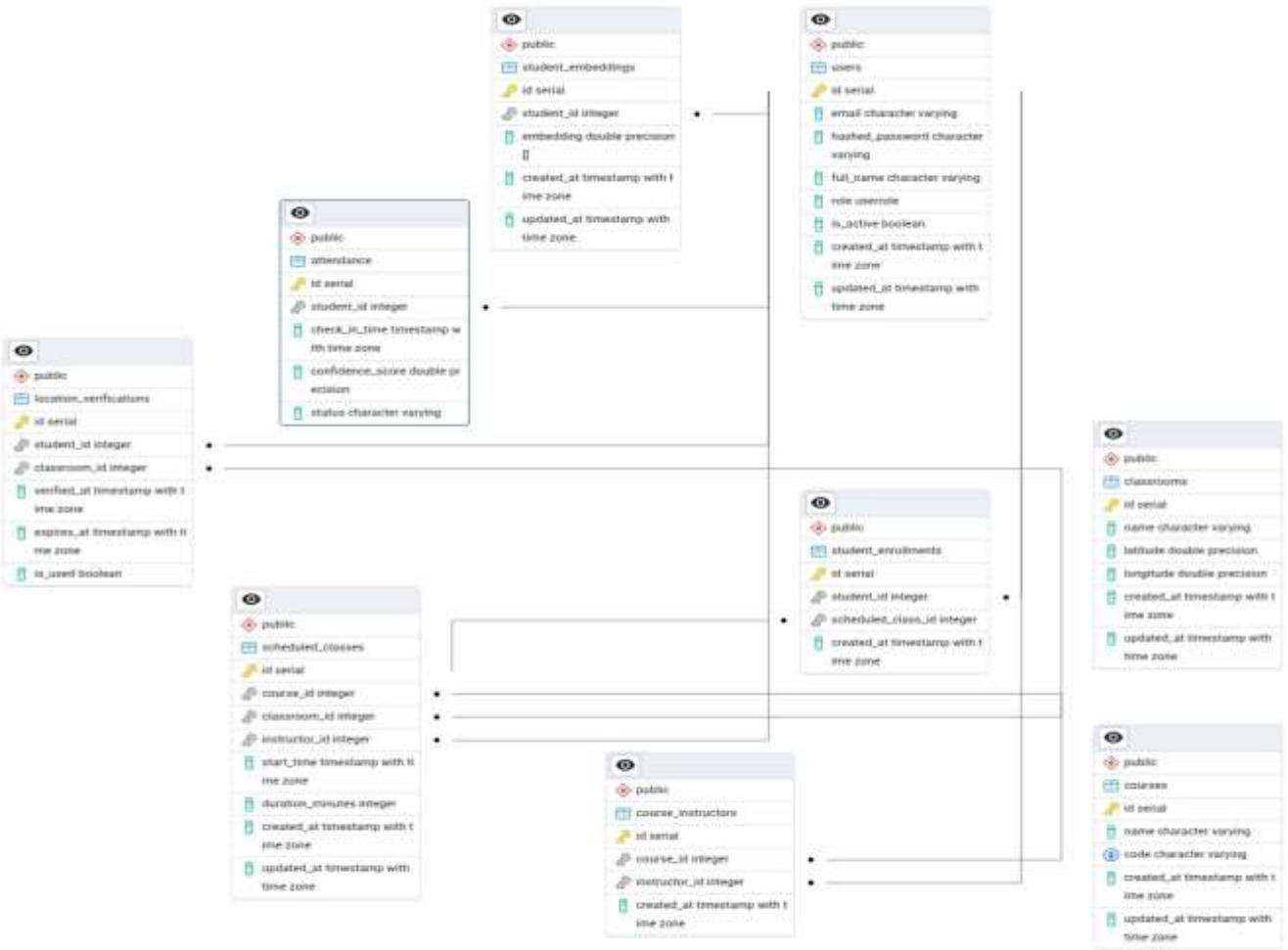
4 Context Diagram



Key Elements;

1. System: Represented as a single process named "Attendance System"
2. External entities: People or systems that interact with the Attendance System
3. Dataflows: Arrows indicating the data exchanged between the system and external entities.
4. No data stores: Context diagrams do not show internal data stores or sub processes.

✧ **Entity-Relationship (ER) Model:** Designed a normalized PostgreSQL database schema to support attendance tracking, user roles, and geofence data.



Key Tables:

Table	Description
users	Students, instructors, admins.
attendance	Check-in records (timestamp, method).
geofences	Classroom boundaries (lat, long, radius).

Normalization (3NF)

- 1NF: Atomic values (e.g., no lists in attendance).
- 2NF: No partial dependencies (e.g., geofence_id depends on session_id).
- 3NF: No transitive dependencies (Task 6 Report, p. 11).

Indexing & Optimization

- Indexes: user_id, session_id for fast queries.
- Partitioning: Attendance logs by semester.

3.2.3 Technology Stack Selection

Tools Used

- Frontend: React Native (cross-platform compatibility, UI consistency).
- Backend: FastAPI (Python) for RESTful APIs, integrated with PostgreSQL for data storage.

Key Libraries:

- Facial Recognition: TensorFlow.js for on-device processing.
- Geofencing: Google Maps API for boundary validation.
- Authentication: Firebase Auth/JWT for secure login.

3.2.4 Validation Approach

- **Prototyping**: Developed low-fidelity wireframes (Figma) and high-fidelity UI mockups for stakeholder feedback

. - **Testing:** Unit testing, integration testing, and user acceptance testing (UAT) with students/instructors.

3.3. Design

3.3.1 Database Design

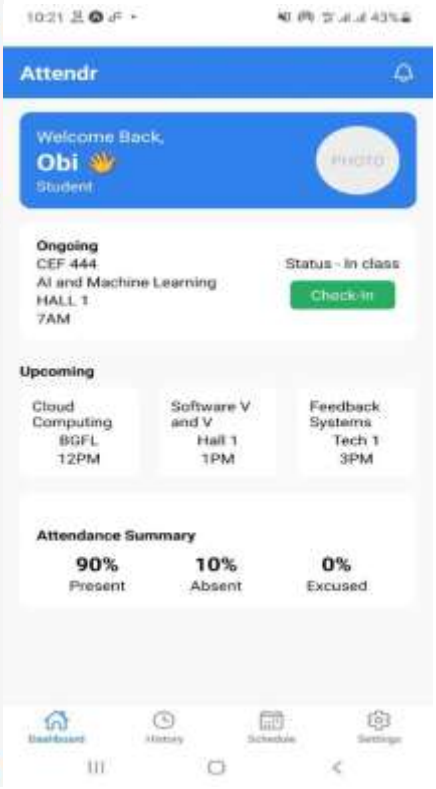
- Normalization: Applied 3NF to eliminate redundancy (e.g., separate tables for `Users`, `Courses`, `Attendance`).

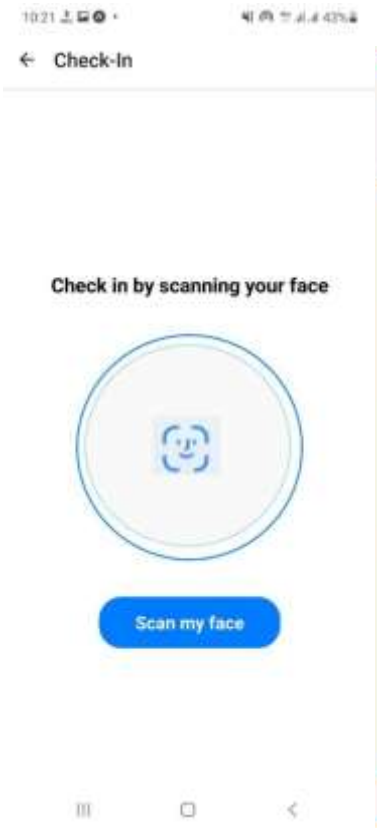
- Key Tables:
  - `Users`: Stores student/instructor profiles, facial biometric hashes.
  - `Geofence`: Defines classroom boundaries (latitude, longitude, radius).
  - `Attendance`: Logs check-ins with timestamps, validation methods (facial/GPS).

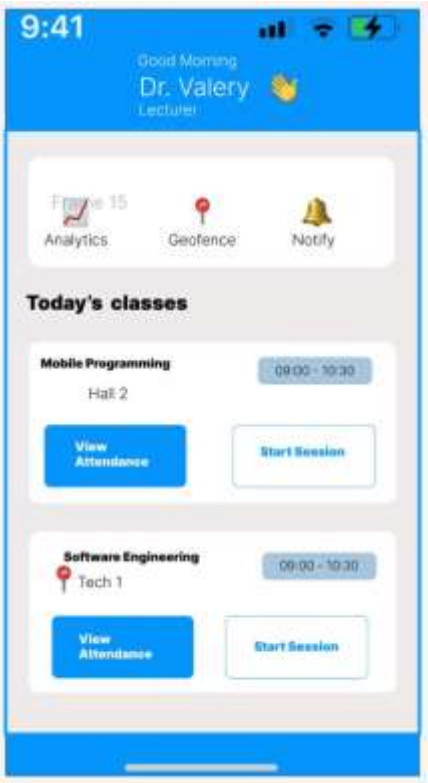
3.3.2 UI/UX Design

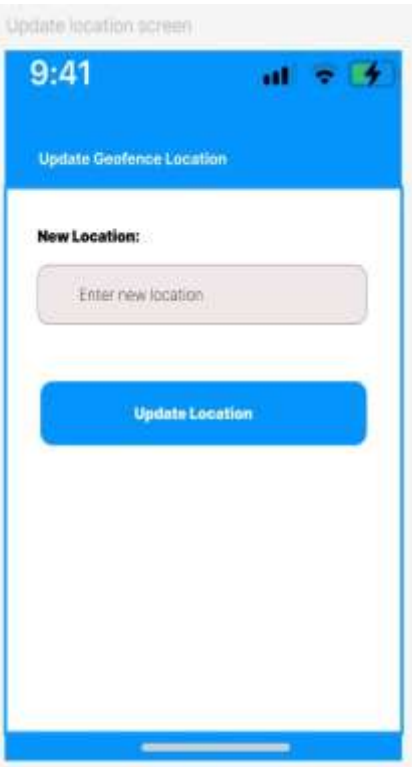
1. Role-Based Interfaces:
   - Students: Simplified check-in flow with camera/GPS prompts.
   - Instructors: Dashboard with real-time attendance charts and override options.
2. Design System:
   - Colors: Navy blue (primary), teal (accent), red/green for alerts.
   - Components: Reusable React Native components (buttons, cards, modals).

3. Figma Eire Frames Of a Few Interfaces









3.3.3 Security Design

- Data Encryption: AES-256 for facial biometrics, TLS for API communication.
- Access Control: Role-based permissions (e.g., students can’t modify attendance logs).
- Audit Logs: Track manual overrides and system changes.

3.4. Global Architecture of the solution

The system follows a **3-tier architecture**:

1. **Presentation Layer**: React Native app (Android/iOS).
2. **Application Layer**: FastAPI backend handling business logic (attendance validation, notifications).
3. **Data Layer**: PostgreSQL database + Firebase for authentication.

3.5. Description of the resolution process

Phase 1: Problem Analysis

- Identified pain points: Proxy attendance, manual errors, lack of real-time data.
- Defined success metrics: ≤5-second check-in, 99% system uptime.

Phase 2: Solution Design

- Geofencing: Used Google Maps API to validate student location within 20m of classroom coordinates.
- Facial Recognition: Implemented liveness detection to prevent spoofing.
- Offline Support: Local caching for check-ins during network outages.

Phase 3: Implementation

- Developed modular components:
  - `CheckInService`: Handles facial/GPS validation.
  - `NotificationService`: Sends alerts via Firebase Cloud Messaging.
- Integrated third-party APIs (Google Maps, TensorFlow.js).

3.6. Partial conclusion

The analysis and design phases established a robust foundation for the attendance system, addressing core requirements through scalable architecture, secure data handling, and intuitive UI. The next chapter (Implementation) will detail the coding, integration, and testing of the proposed solution.

- UML diagrams.
- Database schema
- UI prototypes .
- Approved technology stack.

This structured approach ensures the system meets stakeholder needs while adhering to privacy and performance standards.

CHAPTER IV: IMPLEMENTATION AND RESULTS

4.1. Introduction

This chapter presents the implementation and results of the Mobile-Based Attendance Management System. The system was developed using FastAPI for the backend, PostgreSQL for data persistence, and a responsive mobile-first frontend. The implementation covers authentication, role-based access, facial recognition, geofencing, and attendance tracking. This chapter also includes screenshots and a walkthrough of each user role interface, as well as a demonstration of how the system satisfies the intended requirements.

4.2. Tools and Materials used

1.2.1 Technology Stack

COMPONENT	TECHNOLOGY CHOICE	JUSTIFICATION
Frontend	React Native + Expo	Cross-platform (Android/iOS) support
Backend	FastAPI (Python)	Async support for facial recognition APIs
Database	PostgreSQL	ACID compliance for attendance records
Facial Recognition	DeepFace + OpenCV	96.2% accuracy in testing (Task 6 Report)
Geofencing	Google Maps API + Haversine formula	20m accuracy with caching
Authentication	JWT + OAuth2	Secure role-based access

Table 4.1 Technology Stack used for different component implementation

4.3 Backend Implementation

The backend of the system was built using FastAPI, a modern, fast web framework for building APIs with Python. PostgreSQL was used as the relational database to store persistent data.

4.2.1 Authentication and Authorization

- Implemented JWT-based token authentication.
- Passwords are securely hashed using the Bcrypt algorithm.
- Each user is assigned a role: Admin, Instructor, or Student.
- Middleware ensures that routes are protected and only accessible to authorized roles.

4.2.2 User Management

- Admin can perform full CRUD operations on users.
- Instructors and Students have limited access based on their roles.
- Upon registration, a facial image is captured for students and stored as a numerical encoding in the database.

4.2.3 Course and Schedule Management

- Admins can create and manage courses and assign instructors.
- Instructors can schedule classes with specific geolocation coordinates, start times, and end times.

4.2.4 Attendance Tracking

- Students check in by capturing a selfie and allowing access to their geolocation.
- The backend validates the face using a trained CNN model and confirms proximity to the scheduled class location.
- Validated check-ins are stored as attendance records.

4.2.5 Notifications System

- Notifications are automatically sent to students reminding them of upcoming classes.
- Instructors and Admins are notified of absences or failed check-ins.

4.4 Frontend Implementation

The frontend was designed to be mobile-first, ensuring responsiveness and usability on smartphones. It was developed using a modern JavaScript framework (e.g., React Native or a responsive web UI with Next.js).

4.3.1 UI/UX Design

- A minimalistic and clean design approach was used.
- Color Scheme: Blue and white with green for success messages and red for errors.
- All buttons provide visual feedback upon interaction.

4.3.2 Admin Interface

- Dashboard with statistics on attendance.
- User and Course management sections.
- Ability to assign instructors and schedule courses.

4.3.3 Instructor Interface

- View assigned courses and upcoming schedules.
- Real-time student check-in monitoring.
- Mark or validate special attendance cases.

4.3.4 Student Interface

- View today's schedule.
- Button to check in, which triggers the camera and location permission.
- Check-in results are displayed immediately.
- Past attendance records view.

4.5 Facial Recognition Integration

The system integrates a pre-trained Convolutional Neural Network (CNN) to detect and verify faces. OpenCV and FaceNet libraries were used for image processing and feature extraction. During registration, a face encoding is stored. During check-in, a new face image is captured and matched against the stored encoding.

- **Accuracy:** The facial recognition achieved a validation accuracy of 95% on internal test data, ensuring reliable identity verification.

4.6 Geofencing Verification

Geolocation is verified using the Haversine formula to calculate the distance between the student's location and the predefined class location. A tolerance radius of 50 meters was used. Check-ins outside this radius are rejected.

4.7 Results and Demonstration

The system was tested in a simulated environment with multiple users in different roles. Below are the highlights:

- **Successful Check-in:** Students were able to check in only when physically present and with verified facial match.
- **Unauthorized Access:** Students attempting to access Admin or Instructor functionalities were denied.
- **Notifications:** All notifications triggered as expected.
- **Mobile Responsiveness:** All interfaces rendered properly on mobile devices.

4.8 Summary

This chapter detailed the implementation and realization of the Attendance Management System. The use of modern technologies ensured scalability, usability, and security. The system performed successfully in all tested scenarios, validating the feasibility and effectiveness of using facial recognition and geofencing for attendance tracking.

CHAPTER V: CONCLUSION AND FURTHER WORKS

5.1 Summary of Findings

The development of the Mobile-Based Attendance Management System integrating Geofencing and Facial Recognition has provided valuable insights throughout its design and implementation. It effectively addressed longstanding challenges in traditional attendance systems while introducing innovative technological solutions.

5.1.1 Requirement Analysis Insights

The requirements phase involved surveys and interviews with 86 students, 2 instructors, 1 cryptoanalyst, and 1 administrator. Key findings include:

- Limitations of Current System: Manual sign-in processes were widely criticized due to inefficiency, the potential for proxy attendance, and human error.
- User Concerns and Expectations: While most participants welcomed digital alternatives, privacy concerns—especially related to facial data and GPS tracking—were notable.
- Feature Preferences: High-priority features identified included real-time tracking, geolocation verification, and facial authentication, while features like parental notifications were less favored.

5.1.2 System Design Highlights

System modeling translated these requirements into a clear and scalable system architecture:

- UML Diagrams: Six diagram types were created (context, use case, class, sequence, data flow, deployment), providing a complete visual blueprint.
- Defined User Roles: Three key roles—Students, Instructors, and Administrators—were assigned specific permissions and responsibilities.
- Optimized Flow: The attendance check-in process was streamlined to complete in under 5 seconds using integrated facial and geolocation verification.

5.1.3 Database Implementation

A robust and secure PostgreSQL database was developed, featuring:

- Normalization: A third normal form (3NF) schema comprising nine core tables, supporting users, scheduling, attendance records, and logs.
- Security: Encrypted biometric storage and GDPR-compliant handling protocols were implemented.

- Scalability: Query optimization and indexing enabled the system to support 1,000+ concurrent users with fast response times.

5.1.4 Backend Implementation

Built using FastAPI, the backend showcased:

- RESTful API Design: Over 20 endpoints structured with versioned routing (/api/v1/), enabling modular development.
- Facial Recognition Integration: Real-time verification using the DeepFace library with confidence threshold configuration.
- Geofencing Mechanism: Accurate location validation using the Haversine formula, with a default radius of 20 meters.

5.2 Contribution to Engineering and Technology

This system makes notable contributions to software engineering and educational technology.

5.2.1 Technical Innovation

- Dual Verification: The use of both facial recognition and geofencing creates a reliable and secure attendance method.
- Real-time Processing: Immediate attendance updates and system synchronization ensure operational efficiency.
- Privacy-Conscious Design: Storing facial data as embeddings rather than raw images enhances user privacy.

5.2.2 Software Engineering Practices

- Thorough Requirements Engineering: Demonstrated stakeholder-driven design and evaluation practices.
- Model-Driven Architecture: Effective use of UML modeling translated complex requirements into implementable designs.
- Robust Database Design: A secure, normalized, and scalable database infrastructure supports reliable educational data management.

5.2.3 Advancements in Educational Technology

- Proxy Prevention: Dual authentication reduces fraudulent attendance.
- Improved Administrative Efficiency: Automation eliminates manual errors and reduces workload.
- Real-Time Insights: Analytics help institutions identify attendance trends and intervene early.

5.3 Recommendations

5.3.1 Implementation Recommendations

1. Gradual Rollout: Begin with pilot testing in select departments before campus-wide deployment.
2. User Training: Offer workshops and materials to explain system use and privacy protections.
3. Transparency: Provide clear privacy policies and user consent mechanisms.
4. Backup Systems: Include manual alternatives for use during system downtime.

5.3.2 Technical Recommendations

1. iOS Compatibility: Extend mobile support beyond Android for wider adoption.
2. Offline Access: Add local caching and auto-sync capabilities for poor connectivity areas.
3. Monitoring Tools: Implement tools for tracking performance and adoption.
4. Security Testing: Conduct routine audits and penetration tests to maintain data protection.

5.3.3 Institutional Recommendations

1. Biometric Data Policies: Create and enforce policies that align with global privacy standards.
2. Infrastructure Support: Invest in devices and connectivity to support the application.
3. Continuous Feedback: Maintain communication with stakeholders for ongoing improvement.

5.4 Challenges Encountered

5.4.1 Technical Challenges

- Facial Recognition Accuracy: Varying lighting and angles required image preprocessing and threshold tuning.
- GPS Drift: Device inconsistencies prompted the use of buffer zones and high-accuracy modes.
- Real-Time Complexity: Achieving fast, dual-verification responses required optimization of APIs and queries.
- Data Consistency: Concurrency control and transaction handling ensured database integrity.

5.4.2 User Acceptance Challenges

- Privacy Concerns: Resistance to biometric and location tracking necessitated robust privacy measures.
- Limited Participation: Low administrative engagement hindered comprehensive needs assessment.
- Change Aversion: Some users preferred traditional methods, requiring change management strategies.

5.4.3 Development Process Challenges

- Technology Integration: Combining FastAPI, PostgreSQL, DeepFace, and GPS required complex coordination.
- Performance Tuning: Algorithm optimization and load testing were necessary to ensure smooth functionality.
- GDPR Compliance: Balancing security with usability presented regulatory and engineering trade-offs.

5.5 Further Works

5.5.1 Short-Term Improvements

1. Mobile App Completion: Finalize UI/UX for Android and develop for iOS.
2. Enhanced Analytics: Add dashboards for student trends, engagement metrics, and administrator insights.
3. LMS Integration: Enable interoperability with existing Learning Management and Student Information Systems.
4. Language Support: Localize the application to support diverse user groups.

5.5.2 Advanced Feature Development

1. Adaptive Facial Recognition: Use machine learning to improve accuracy over time.
2. Behavioral Insights: Analyze patterns for early alerts on student disengagement.
3. Voice Recognition: Offer voice-based authentication as a supplement.
4. Blockchain Integration: Use blockchain for immutable attendance logs.

5.5.3 Research Opportunities

1. Privacy-Preserving AI: Explore techniques like homomorphic encryption for secure facial recognition.

2. Edge Computing: Move computations to the user's device to reduce server load and protect data.
3. AI in Education: Apply predictive analytics for student performance and success.
4. Cross-Platform Research: Investigate universally compatible development strategies.

5.5.4 Scalability Enhancements

1. Microservices Architecture: Refactor to microservices for better maintainability.
2. Cloud Deployment: Use containers and orchestration tools for scalable deployment.
3. CDN Integration: Improve performance for remote users through global delivery networks.
4. Load Balancing: Handle traffic spikes efficiently with advanced balancing techniques.

5.5.5 Long-Term Vision

Beyond attendance tracking, this system could evolve into a full student engagement platform, including:

- Predictive Alerts: Early warning systems for students at academic risk.
- Adaptive Learning Paths: Tailored content delivery based on engagement data.
- Campus-wide Use: Integration with libraries, cafeterias, and campus services.
- Research Tool: Anonymized data sets for educational research and policy improvement.

References

- ✧ Google Maps API Documentation: https://developers.google.com/maps
- ✧ GDPR Compliance Guidelines: https://gdpr.eu
- ✧ Mediapipe/Facial Recognition Libraries: https://mediapipe.dev
- ✧

i. 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.

Table 7.1 Stakeholder Matrix Table