UNIVERSITY OF BUEA

Buea, South West Region

Cameroon

P.O. Box 63,

Tel: (237) 3332 21 34/3332 26 90

REPUBLIC OF CAMEROON

PEACE - WORK - FATHERLAND

Task 4:
System Modeling and Design

Instructor: **Dr Nkemeni Valery**

Course Code: **CEF440**

Course Title: **Internet Programming and Mobile Programming**

By Group 7:

NAME	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 Academic Year

Table Of Content

Introduction

System modeling and design is a critical phase in software development that defines the structure, behavior, and interactions within a system. This report presents comprehensive models for a mobile-based attendance management system that leverages facial recognition and geofencing. Through diagrams such as context, data flow, use case, sequence, class, and deployment, we illustrate the system's architecture, data processing, and operational flow to ensure secure, automated, and real-time attendance tracking.

I. Context Diagram Analysis of the SMART Facial Recognition Attendance System

Executive Summary

The SMART Attendance System as represented in the context diagram below illustrates the system boundaries and data flows between the core system and five external entities: Students, Instructors, GPS Service, Database and the Facial Recognition Service. This document outlines the logical system architecture from a high-level view, analyzes interaction patterns, and identifies security and privacy considerations.

1. Purpose and Scope

The SMART Facial Recognition Attendance System represents a technological solution to automate attendance tracking in educational institutions. This report analyzes the context diagram that depicts the system's boundaries and interactions with its external entities, while all the internal entities are.

The scope encompasses:

- System boundaries definition
- Data flow analysis
- External entity interactions
- Security and privacy considerations
- Implementation considerations

2. System Overview

The SMART Facial Recognition Attendance System automates traditional attendance tracking by implementing biometric technology, location verification, and centralized data management. The system aims to eliminate manual attendance processes, reduce administrative burdens, prevent proxy attendance, and provide real-time attendance records for both students and instructors.

3. Context Diagram Analysis

3.1 Context Diagram Overview

The context diagram provides a logical (what actions are happening) in high-level visual representation of the SMART Facial Recognition Attendance System and its interactions with its external entities. Logical or conceptual diagrams have different levels of abstractions, with context diagram placed at the highest level ,making the SMART Facial Recognition Attendance System to be seen as a single process and placed at the center , with all external entities arranged around it, connected by data flows that indicate the information exchanged between them.

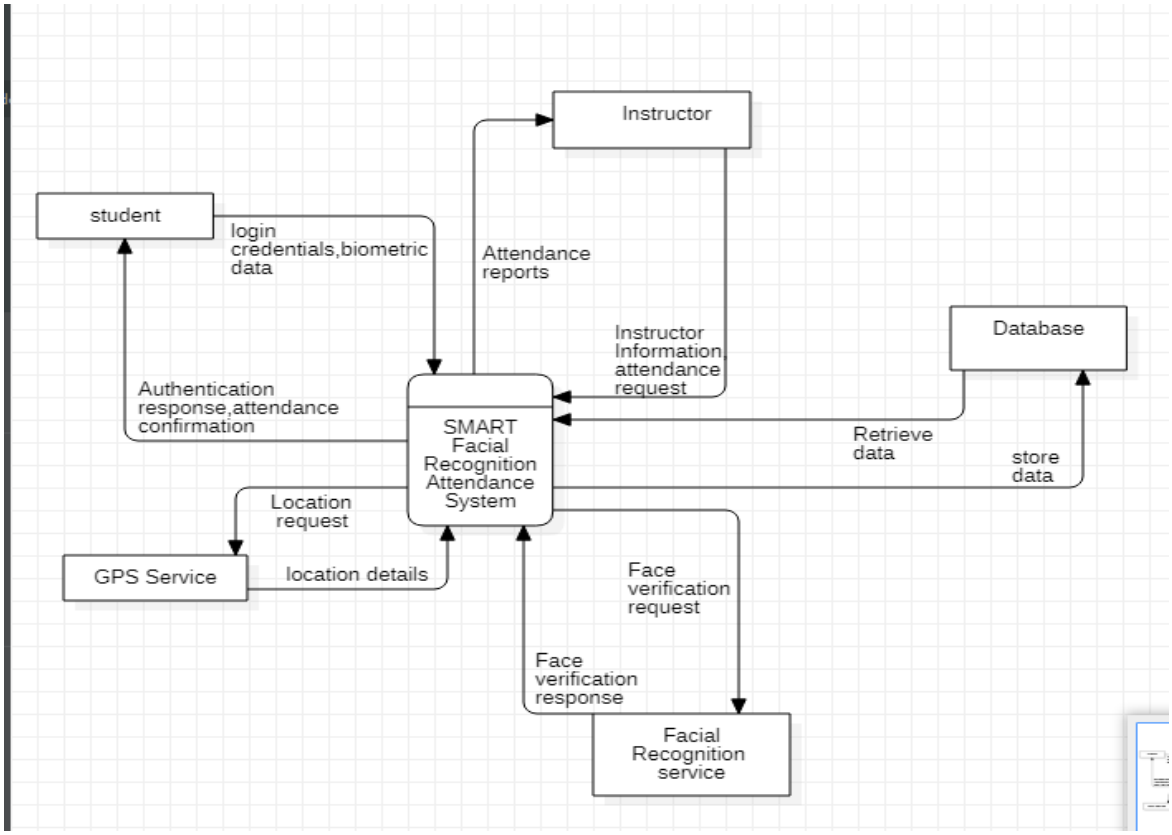

fig 1: Context Diagram

The diagram illustrates the system boundary, clearly represents which components are part of the external entities. The directional arrows represent data flows, showing what information enters and exits the system, and which external entity is the source or destination of that information.

3.2 System Boundary Definition

The context diagram establishes clear system boundaries or limits by positioning the SMART Facial Recognition Attendance System as the central process or main process surrounded by five external entities.

4. Data flow analysis between the external entities and the central process or system

4.1 Student Interactions

- Inputs to System: Login credentials, biometric data
- Outputs from System: Authentication response ,attendance confirmation.

4.2 Instructor Interactions

- Inputs to System: Instructor information, attendance requests
- Outputs from System: Attendance reports

4.3 GPS Service Interactions

- Inputs to System: Location details
- Outputs from System: location request

4.4 Database Interactions

- Inputs to System: Data retrieval requests
- Outputs from System: Stored data

4.5 Facial Recognition Service Interactions

- Inputs to System: Face verification response
- Outputs from System: Face verification requests

5. Security and Privacy Considerations

5.1 Biometric Data Handling

The context diagram reveals that biometric data flows from Students to the System raises several considerations:

- Secure transmission protocols must be implemented for biometric data transfer.

- Data at rest should be encrypted in the database.
- Compliance with data protection regulations must be ensured.
- Clear data retention policies should be established.

5.2 Location Data Processing

GPS location data processing introduces privacy implications:

- Location data should be used solely for attendance verification.
- Transparency regarding location tracking must be maintained with users.

5.3 Access Control Requirements

Multiple data flows indicate the need for robust access control:

- Role-based access controls for instructors and administrators.
- Student access limited to personal attendance records.
- Authentication required for all system interactions.

6. Implementation Considerations

6.1 System Integration Points

The context diagram identifies key integration points that require special attention:

- GPS Service data consumption.
- Database connections for data retrieval and storage.
- User authentication services.

6.2 Real-time Processing Requirements

Several data flows suggest real-time processing needs:

- Face verification requests and responses.
- Attendance confirmation to students.

6.3 Scalability Considerations

The context diagram implies potential scalability requirements:

- Supporting multiple simultaneous student verifications.
- Processing attendance for multiple classes concurrently.

The context diagram for the SMART Facial Recognition Attendance System provides a clear high-level view of system boundaries and interactions. The analysis reveals the system with multiple integration points, significant security and privacy considerations, and real-time processing requirements. The system design demonstrates a comprehensive approach to automating attendance tracking while incorporating verification mechanisms to ensure attendance authenticity.

II- DATA FLOW DIAGRAM (DFD)-LEVEL 1

The Data Flow Diagram (DFD) provides a graphical representation of how data flows through the "Mobile-Based Attendance Management System Based on Geofencing and Facial Recognition." It describes the various processes, external entities, data stores, and data flows involved in the system. This section presents a detailed Level 1 DFD which captures the core functionalities of the system and illustrates how data is input, processed, stored, and output.

1. Purpose of the DFD

The DFD aims to visually map the key operations within the attendance system to:

- Clarify system requirements
- Provide a foundation for system design
- Enhance communication among stakeholders
- Serve as documentation for system developers and auditors

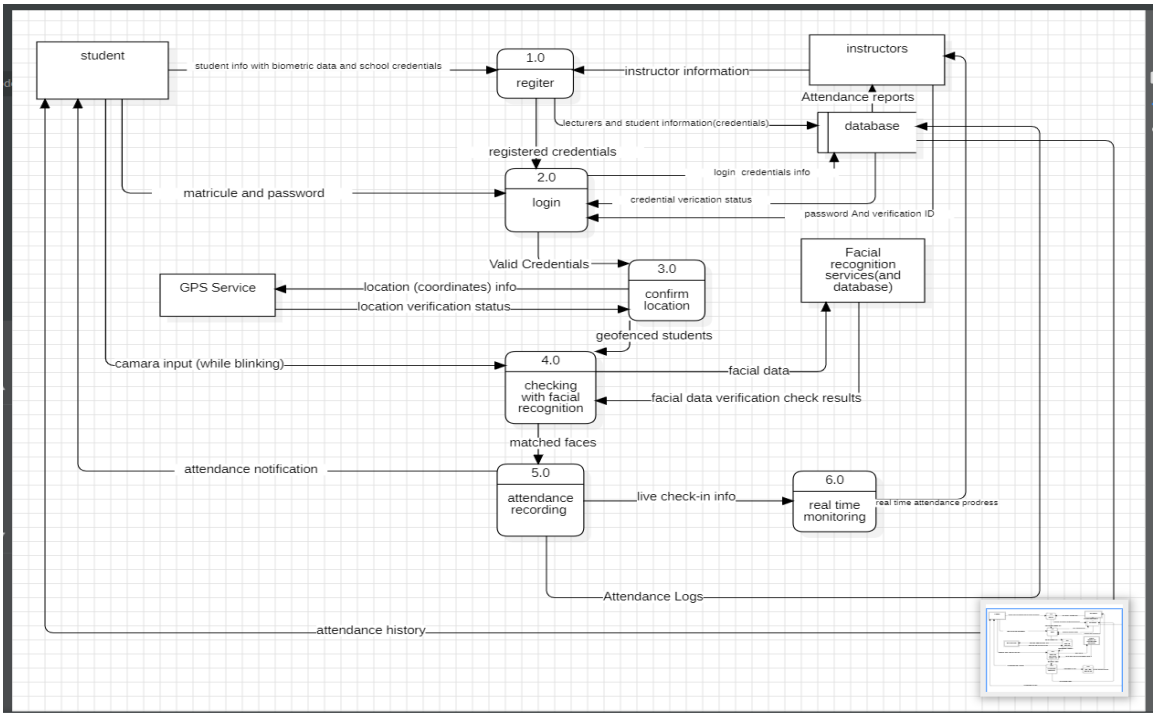

fig 2: Level 1 DFD

2. DFD Components Overview

2.1 External Entities

An **entity** is an external object (person, system, or service) that interacts with the system by providing input or receiving output. There are four external entities which include

1. **Student**: Provides biometric data and school credentials, receives attendance history and notifications.
2. **Instructors**: Receive real-time attendance reports and logs.
3. **GPS Service**: Supplies geolocation data used for geofencing.
4. **Facial Recognition Services and Database**: Used to verify facial input against stored biometric records.

2.2 Processes

A **process** is an operation or task within the system that transforms input data into output data. The processes

1.0 Register: Collects student and instructor data, including biometric information and credentials.

2.0 Login: Authenticates users based on matricule number and password.

3.0 Confirm Location: Validates the physical location of a user through GPS services and geofencing.

4.0 Checking with Facial Recognition: Verifies the student's identity using facial recognition algorithms.

5.0 Attendance Recording: Logs verified attendance entries.

6.0 Real-Time Monitoring: Provides live check-in data to instructors for attendance monitoring.

Summary table

Process	Description	Inputs	Outputs	Data Stores
1.0 Register	Handles new user enrolment	Personal & biometric data	Registered credentials	Main Database
2.0 Login	Authenticates user	Login credentials	Login status	Main Database
3.0 Confirm Location	Verifies geolocation	GPS coordinates	Location status	-
4.0 Facial Recognition	Verifies identity	Live facial image	Match result	Biometric Database
5.0 Record Attendance	Logs successful check-ins	Verification status	Attendance record	Attendance Logs
6.0 Real-Time Monitoring	Tracks and reports attendance	Live data stream	Alerts, Reports	Attendance History

2.3 Data Stores

A **data store** is a place where data is held for processing or future use. **Example:** Database, Attendance Logs, Facial Recognition Database.

- **Database**: Central repository for user credentials, attendance logs, and report data.

2.4 Data Flows

- "Student info with biometric data and school credentials" flows from Student to Register.
- "Instructor information" flows to Register.
- "Registered credentials" are passed to the Login process.
- "Matricule and password" are submitted by Student to Login.
- "Credential verification status" and "Valid credentials" are exchanged with the database and sent to Confirm Location.

- "Location (coordinates) info" is sent from the student to GPS Service and forwarded to Confirm Location.
- "Location verification status" is returned and students are geofenced.
- "Camera input (while blinking)" is submitted by Student to Facial Recognition process.
- "Facial data" is sent to the Facial Recognition Services and Database.
- "Facial data verification check results" are returned.
- "Matched faces" are sent to the Attendance Recording process.
- "Attendance notification" and "Attendance history" are sent to Student.
- "Live check-in info" flows from Attendance Recording to Real-Time Monitoring.
- "Real-time attendance progress" and "Attendance Reports" are shared with Instructors.
- "Attendance Logs" are stored in the database.

3. Detailed Process Descriptions

1.0 Register

This process handles the initial data entry from students and instructors. It captures:

- Biometric data (e.g., facial images)
- Institutional credentials (e.g., matricule, department, etc.)

2.0 Login

This process authenticates users based on submitted credentials. It communicates with the central database to:

- Validate usernames and passwords
- Forward valid credentials to location verification

3.0 Confirm Location

Utilizes GPS coordinates to verify that the student is within an approved geofenced area. It involves:

- Retrieving GPS data from the device
- Comparing with stored geofenced coordinates
- Marking the student as location-verified

4.0 Checking with Facial Recognition

This process ensures the person logging in is physically present. It involves:

- Capturing live image input (e.g., blinking)
- Sending data to facial recognition service
- Matching with stored biometric templates

5.0 Attendance Recording

Once facial identity and location are verified, this process logs the attendance:

- Stores attendance data in the database
- Sends real-time check-in information
- Notifies the student of the record

6.0 Real-Time Monitoring

Enables instructors to:

- Monitor live check-in status
- Review attendance logs
- Receive real-time updates on attendance

4. Data Stores Description

- **Database**: A relational and NoSQL data store that maintains the following:
    - Student and instructor credentials
    - Biometric templates
    - Attendance logs
    - Geofence boundaries
    - Audit trails and system logs

This Level 1 DFD outlines the critical data interactions and flows in the Mobile-Based Attendance System. It provides a clear picture of how various modules collaborate to enable secure, automated attendance recording using geofencing and facial recognition. The diagram ensures traceability and supports system design, evaluation, and audit.

III- Use Case Diagram and Analysis

This section presents a **focused technical analysis** of the **Use Case Diagram** for the **Mobile-Based Attendance Management System (MAMS)** a solution integrating **facial recognition** and **geofencing** to ensure secure, automated, and real-time attendance tracking. The diagram outlines key system functionalities, actor interactions, and process flows, while the accompanying **Use Case Specification Table** elaborates on each interaction's technical and operational logic. It excludes unrelated diagrams and aligns with formal software engineering documentation standards like **IEEE 830-1998.**

1. Use Case Diagram Overview

1.1. Actors and System Scope

Actor	Role
Student	Initiates facial recognition check-in, views attendance.
Lecturer	Creates sessions, manually overrides failed check-ins, generates reports.
Administrator	Configures geofences, audits attendance logs.
External Systems	GPS/Wi-Fi Services for geolocation; Biometric DB for face verification.

1.2. Use Cases and Interactions

The Use Case Diagram captures the following key functionalities:

- Facial Recognition Check-In
- Geofencing Validation
- Manual Override
- Retry Failed Check-In
- Generate Attendance Report
- Configure Geofence
- Fallback for Weak GPS Signal

1.3. Visual Diagram

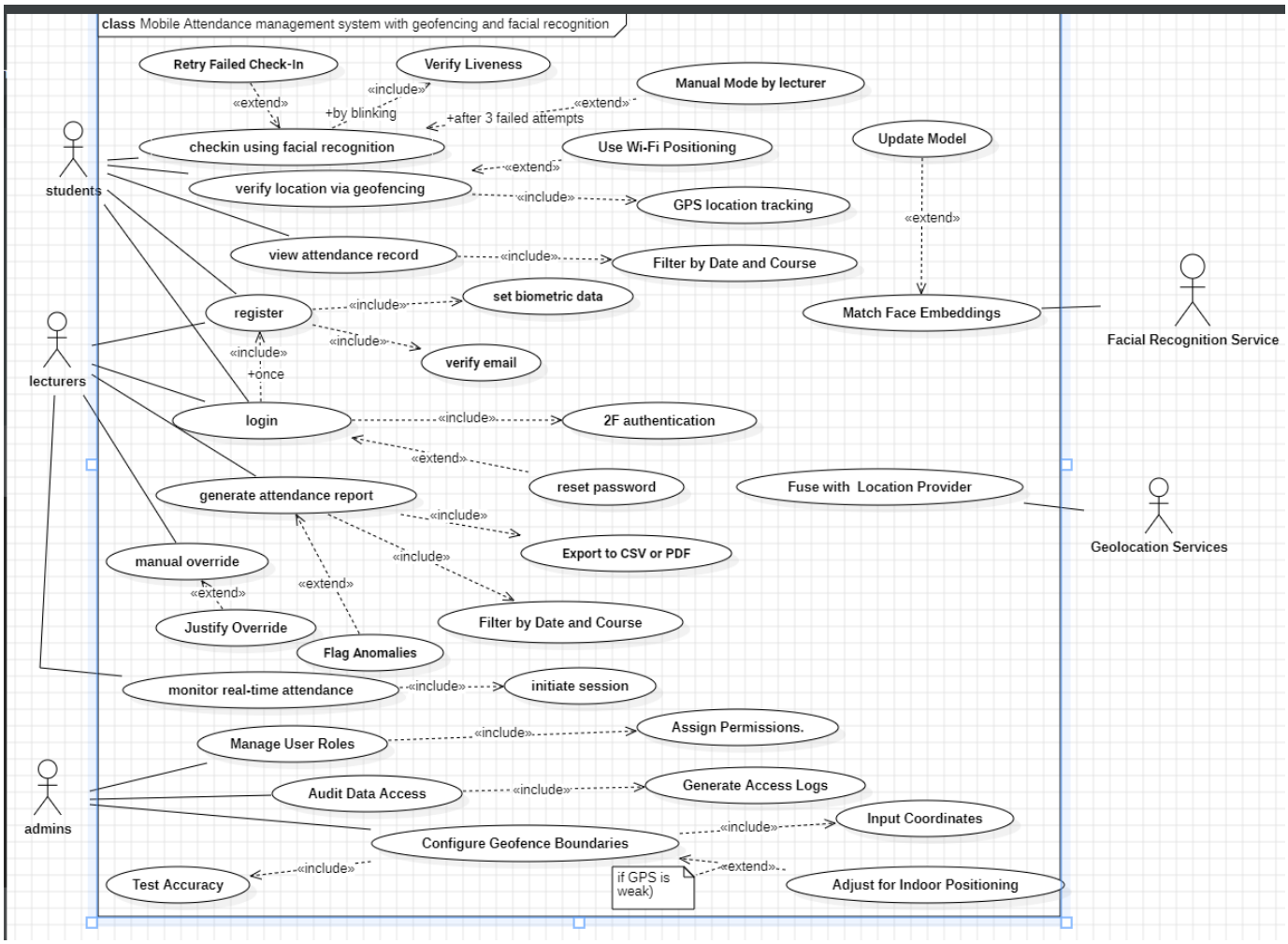

fig 3: Use Case Diagram

2. Use Case Specification Table

The **Use Case Specification Table** provides a structured and detailed description of each use case identified in the system. It outlines the **actor(s) involved**, **system behavior**, **preconditions**, **triggers**, **postconditions**, and **exception flows** associated with each use case. This table complements the Use Case Diagram by offering deeper insight into how the system responds to various user interactions and external conditions.

Each row in the table corresponds to a specific functionality, including its **normal operation** and how the system handles **errors or alternate flows**, ensuring both functional completeness and system reliability.

Use case specification table

Use Case ID	Use Case Name	Actor	Description	Preconditions	Postconditions	Triggers	Exceptions
UC-01	Facial Recognition Check-In	Student	Verifies student identity via facial biometrics.	1. Student is registered. 2. App is open.	1. Attendance recorded. 2. Log updated.	Student initiates check-in.	1. Face not recognized (UC-04). 2. Low light.
UC-01a	Liveness Detection <<include>>	System	Ensures real person is presenting their face (anti-spoofing).	1. Camera active.	Liveness confirmed.	Part of UC-01	Fake image/video detected.
UC-02	Geofencing Validation	GPS Service	Confirms student is within classroom boundaries.	1. GPS/Wi-Fi enabled. 2. Geofence set.	Location verified.	UC-01 triggers location check.	1. Weak GPS signal (UC-05).
UC-02a	Adjust for Weak GPS <<extend>>	System	Switches to Wi-Fi/indoor positioning if GPS is unreliable.	1. GPS signal lost.	Hybrid positioning activated.	GPS timeout.	No backup signal available.
UC-03	Manual Override	Lecturer	Lecturer manually marks attendance after biometric failure.	1. 3 failed attempts. 2. Lecturer logged in.	Attendance updated with justification.	System prompts override.	1. Unauthorized override attempt.

UC-04	Retry Failed Check-In <<extend>>	Student	Allows 3 retries if facial recognition fails.	1. Initial check-in failed.	Success or switch to UC-03.	Automatic after failure.	Max retries exceeded.
UC-05	Generate Attendance Report	Lecturer/Admin	Exports filtered reports (CSV/PDF).	1. Attendance data exists.	Report generated.	User requests report.	No data for selected filters.
UC-06	Configure Geofence	Admin	Sets GPS boundaries for classrooms.	1. Admin has privileges.	Geofence saved.	Admin inputs coordinates.	Invalid coordinates.

3. Technical Narratives

UC-01: Facial Recognition Check-In

Process Flow:

1. Student launches app and logs in with 2FA.
2. Camera initiates face capture and compares image with Biometric DB.
3. On success: Attendance is logged.
4. On failure: UC-04 is triggered.

Technologies To Be Used:

- **OpenCV, TensorFlow Lite**
- Liveness detection through blink/motion recognition.

UC-02: Geofencing Validation

Process Flow:

1. App fetch's current location.
2. Compares coordinates against classroom geofence.
3. On match: UC-01 proceeds.
4. On mismatch or error: UC-05 is triggered.

Technologies Used:

- **Google Fused Location API**, fallback via Wi-Fi triangulation.
- Tolerance radius: 30m ± 5m.

UC-03: Manual Override

Process Flow:

1. After three failed attempts, lecturer selects student manually.
2. Provides justification (e.g., low light, face partially obstructed).
3. Attendance marked, logs updated.

Security Measures:

- Role-Based Access Control (RBAC)
- Timestamp, user ID, and justification stored for audit.

4. Exception Handling Summary

Scenario	Fallback or Action	Outcome
Face not recognized (3x)	Trigger UC-03 (Manual Override)	Attendance decision deferred to lecturer
Weak or missing GPS signal	Trigger UC-05, use Wi-Fi-based location	Indoor fallback positioning enabled
Unauthorized override attempt	Reject + alert admin	Blocked and logged for review
Biometric DB not accessible	Use local cache + queue for sync	Offline check-in mode activated

The **Use Case Diagram** and detailed **Specification Table** collectively establish a **clear, modular, and secure** framework for the Mobile-Based Attendance Management System. They highlight:

These models serve as a foundational artifact for system prototyping, iterative development, and stakeholder validation.

IV. Sequence Diagram

Sequence diagrams are a vital tool in modeling the dynamic behavior of our mobile-based attendance management system that incorporates geofencing and facial recognition technologies. These diagrams visually represent the complex interactions between system components, helping to clarify the timing and order of operations that occur during attendance verification processes.

This section presents a detailed sequence diagram and narrative for the attendance marking process using geofencing and facial recognition within the mobile application.

1. System Overview

The Smart Attendance Management System utilizes geofencing and facial recognition technologies to automate university attendance tracking.

Key components include:

- Course-specific attendance sessions
- Two-factor student verification (location + biometrics)
- Real-time reporting dashboards

2.Participants Involved (in the sequence diagram)

Student: The individual whose attendance is being tracked.

Lecturer: The person responsible for initiating a session, track attendance records and generate attendance report.

Geofencing API: The API responsible for determining students location and verifying if they are within the designated geofence area.
Facial Recognition API: The API responsible for recognizing and verifying student faces.

System: The mobile-based attendance management system.

Database: Storage system for our attendance management system.

3. Sequence Diagrams

3.1 Student Sequence Diagrams

3.1.1 Student Signup

Objective: Register a new student with biometric (facial) data.

Participants: Student, System, Database, Facial Recognition API

Process Flow:

- Student submits personal details (name, matricule, department, etc.).
- System requests facial scan via API.
- Encrypted face data is stored in the database.
- System confirms successful registration.

Key Features:

- ✓ Facial data is encrypted before storage.
- ✓ Matricule serves as a unique identifier.

Diagram:

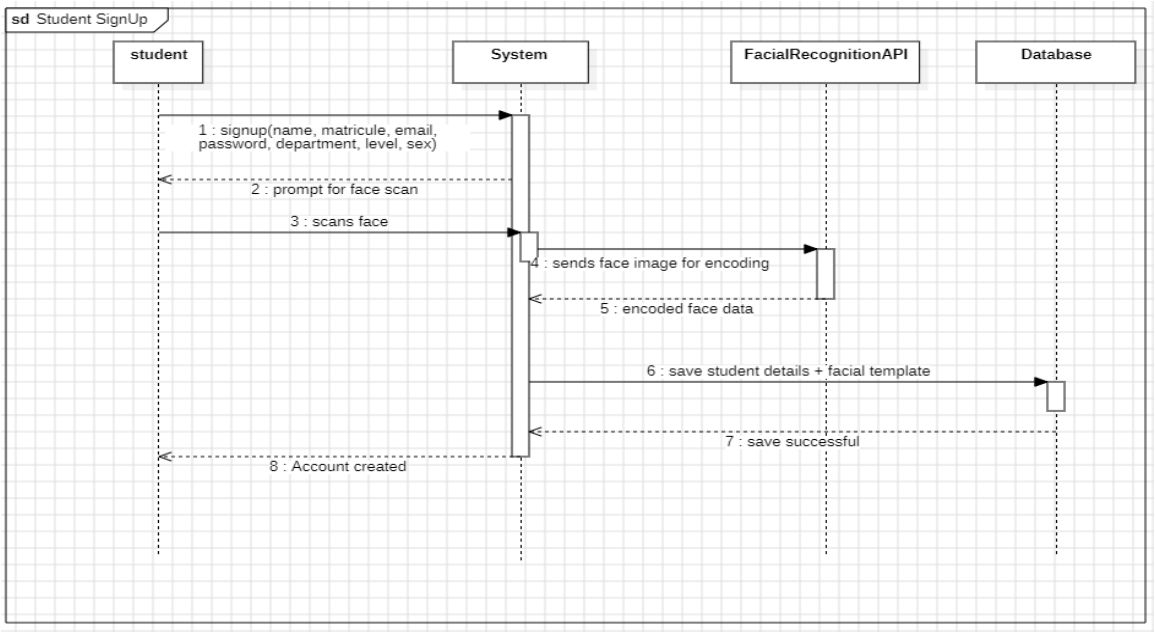

fig 4.1.1: Student Sign-up

3.1.2 Student Login

Objective: Authenticate a student using matricule and password.

Actors: Student, System, Database

Process Flow:

- ❖ Student enters matricule and password.
- ❖ System verifies credentials against the database.
- ❖ On success, redirects to the dashboard.

Exception Cases:

- o *Invalid credentials*: Shows error message, login not successful.

Key Features:

No facial recognition required for login (only for attendance check-in).

Matricule is the primary key.

Diagram:

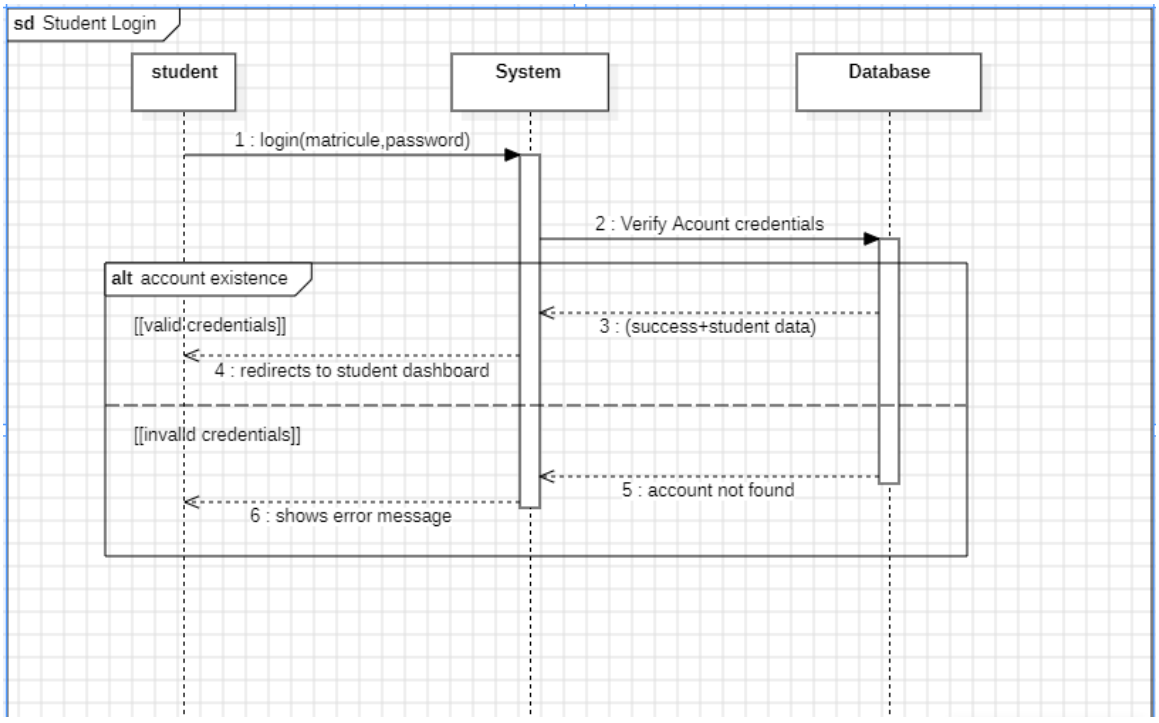

Fig 4.1.2: Student login

3.1.3 Attendance Check-In

Objective: Record attendance via geofencing + facial recognition.

Actors: Student, System, Geofence API, Facial Recognition API, Database

Process Flow:

- Student selects course for check-in
- System verifies:
    - Active session exists (initiated by lecturer)
    - Device location within classroom geofence
- Facial recognition compares live scan with enrolled template

Exception Cases:

No active session: "Attendance not currently being taken"

Location mismatch: "Move to designated classroom"

Biometric failure: Fallback to manual verification by lecturer

Key Features:

- Dual verification (location + biometrics).
- Fallback mechanism not shown (e.g., manual override).

Diagram:

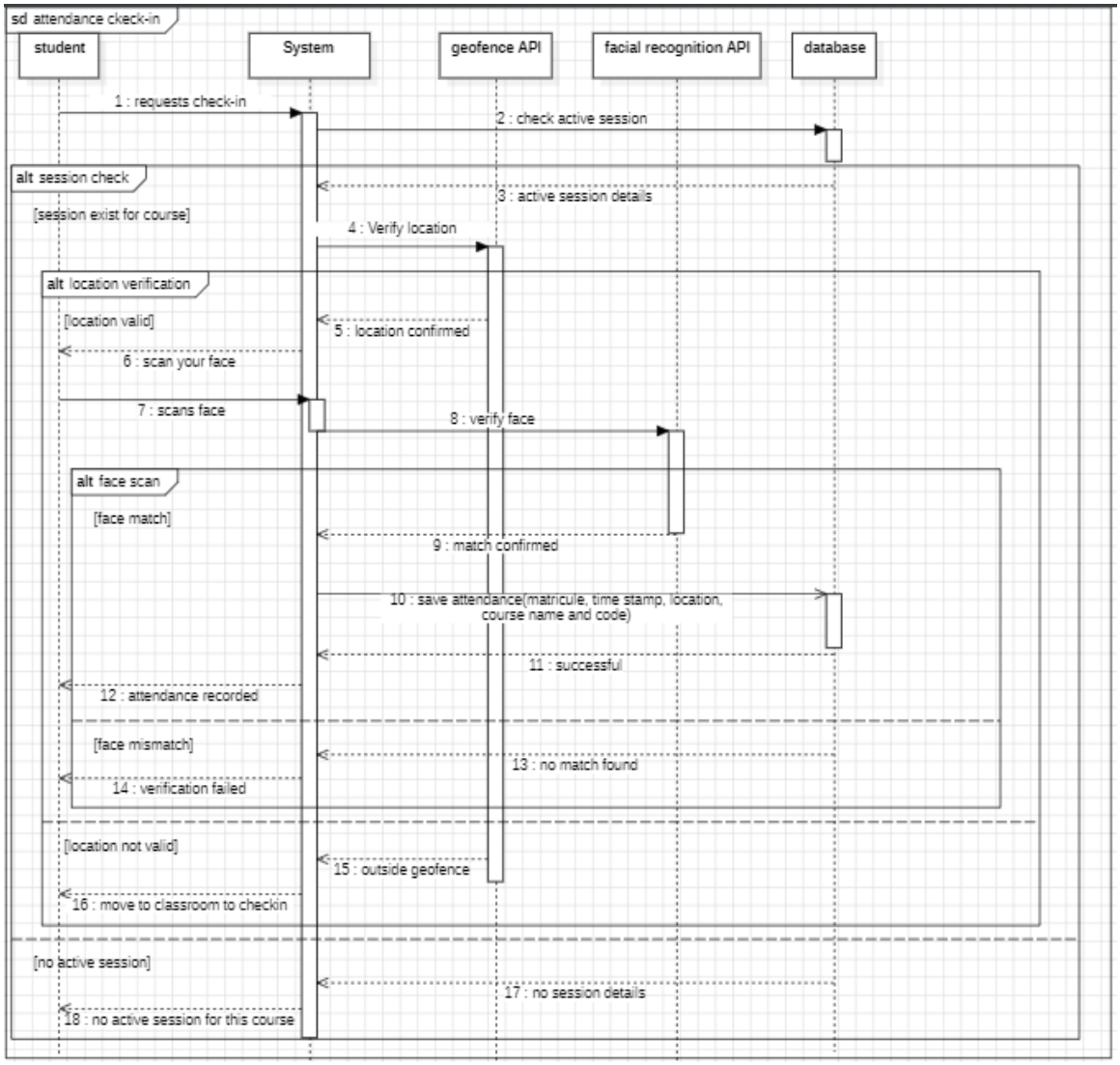

Fig 4.1.3: Student Class Check-in

3.2 Lecturer Sequence Diagrams

3.2.1 Lecturer Signup

Objective: Register a new lecturer (no biometric data needed).

Actors: Lecturer, System, Database

Process Flow:

- The lecturer initiates the sign-up process by providing personal and professional details.
- The system validates the input data.
- Upon successful validation, the system stores the lecturer's information in the database.

- The system confirms the successful creation of the account to the lecturer

Key Features:

Lecturer ID is the unique identifier.

Simpler than student signup (no facial scan).

Diagram:

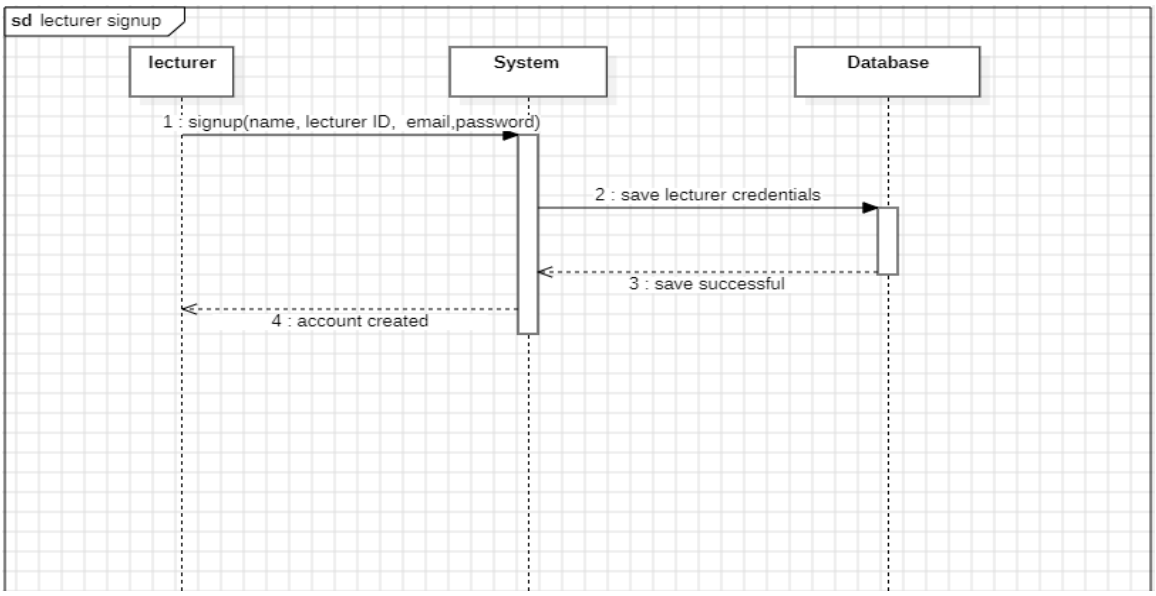

fig 4.2.1: Lecturer Signup

3.2.2 Lecturer Login

Objective: Authenticate a lecturer using ID and password.

Actors: Lecturer, System, Database

Process Flow:

- ❖ Lecturer enters ID and password.
- ❖ System validates credentials.
- ❖ Upon successful verification, the lecturer is granted access to the attendance dashboard.

Exception Cases:

- *Invalid credentials*: Shows error message, login not successful.

Diagram:

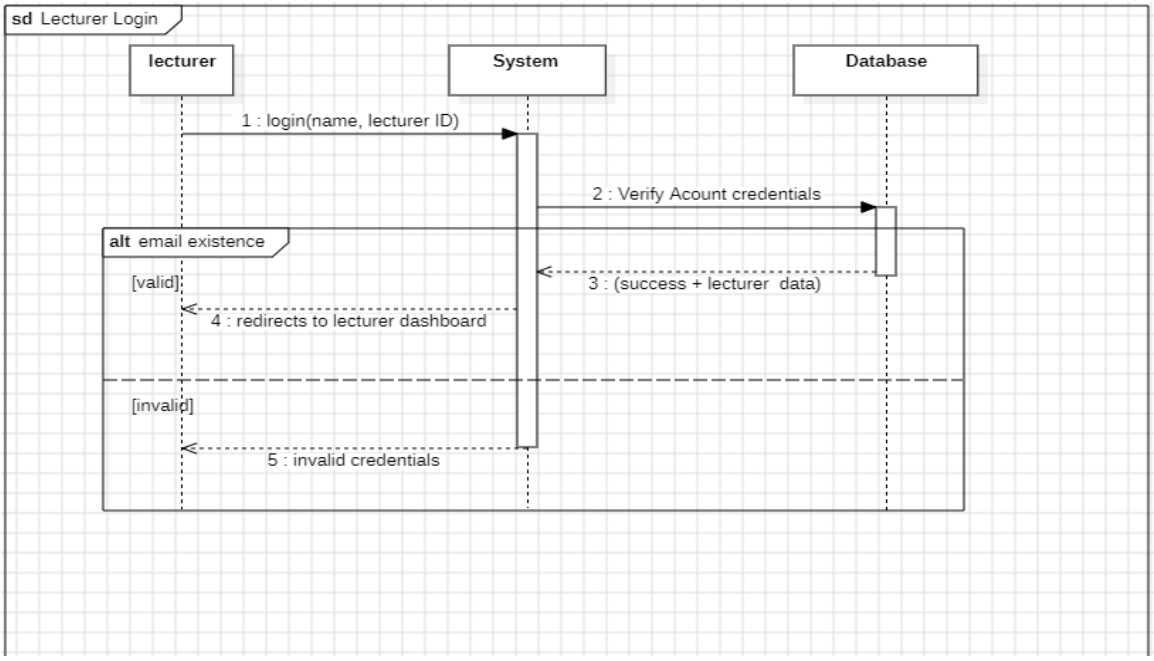

fig 4.2.2: Lecturer Login

3.2.3 Session Iniation and Attendance Tracking

Actors: Lecturer, System, Classroom Database

- The lecturer selects a course and initiates a new session.
- The system verifies the course details and creates a new session record in the database.
- The system confirms the successful initiation of the session to the lecturer.
- The lecturer tracks student attendance in real-time.
- The system displays attendance log.

Diagram:

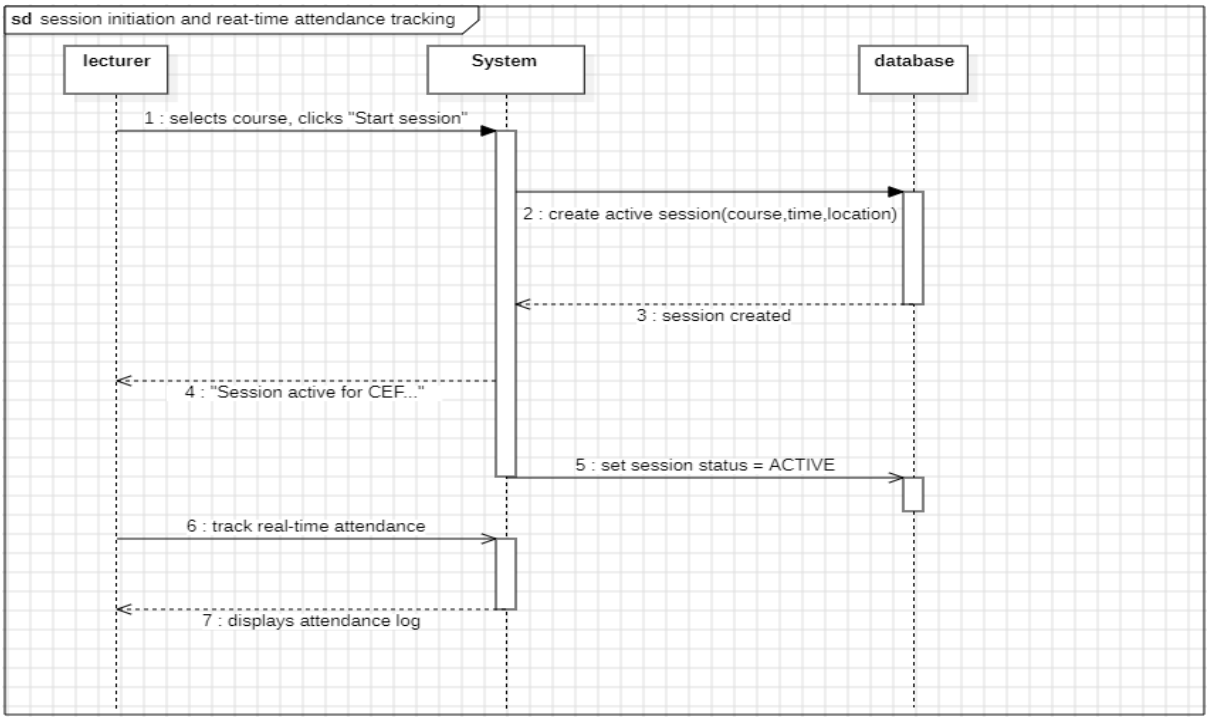

fig 4.2.3: Session Initiation And Real-time Attendance Tracking By Lecturer

3.2.4 Attendance Report Generation

Actors: Lecturer, System, Database

- The lecturer requests the generation of an attendance report for a specific course and date.
- The system retrieves relevant attendance data from the database.
- The system compiles the data into a structured report format.
- The system presents the generated report to the lecturer.

Diagram:

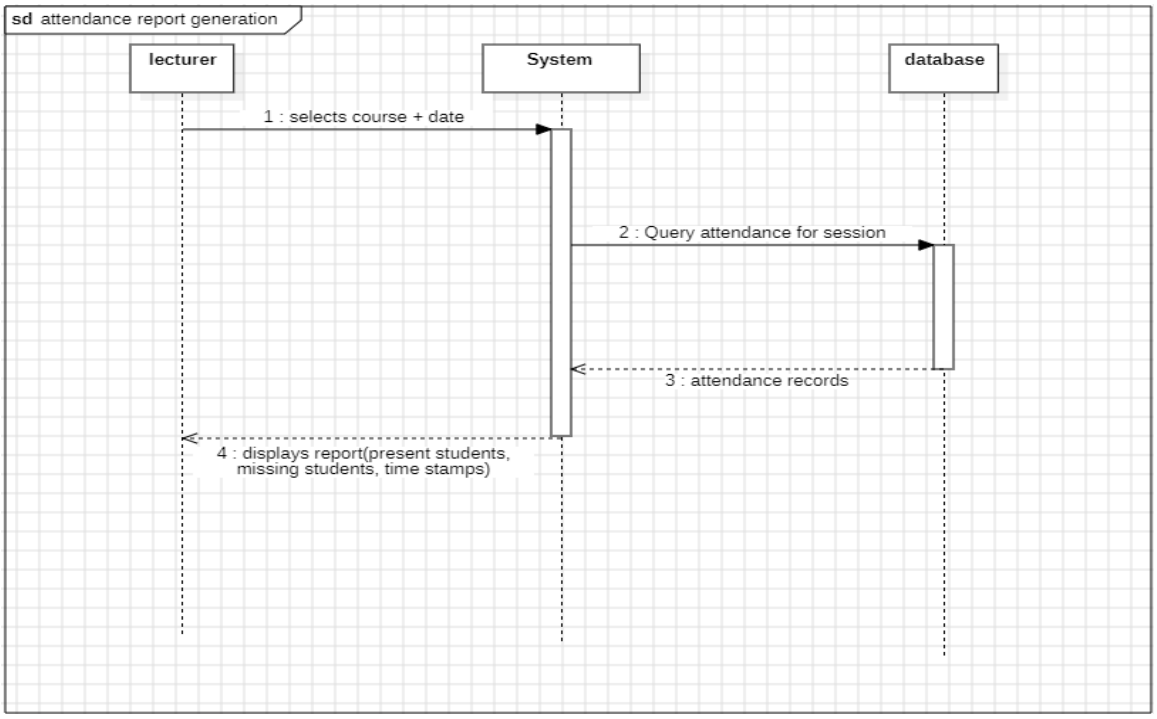

fig 4.2.4: Attendance Report Generation By Lecturer

This sequence diagram helps in understanding the flow and interaction between the components involved in the attendance process. It ensures proper communication, system validation, and user acknowledgment at each step.

V. CLASS DIAGRAM

1. Purpose of the Class Diagram

The class diagram models the static structure of the attendance management system by defining the system's key classes, their attributes, methods, and the relationships among them. It serves as a blueprint for both the database schema and the backend logic that governs the interaction between students, lecturers, courses, and attendance sessions

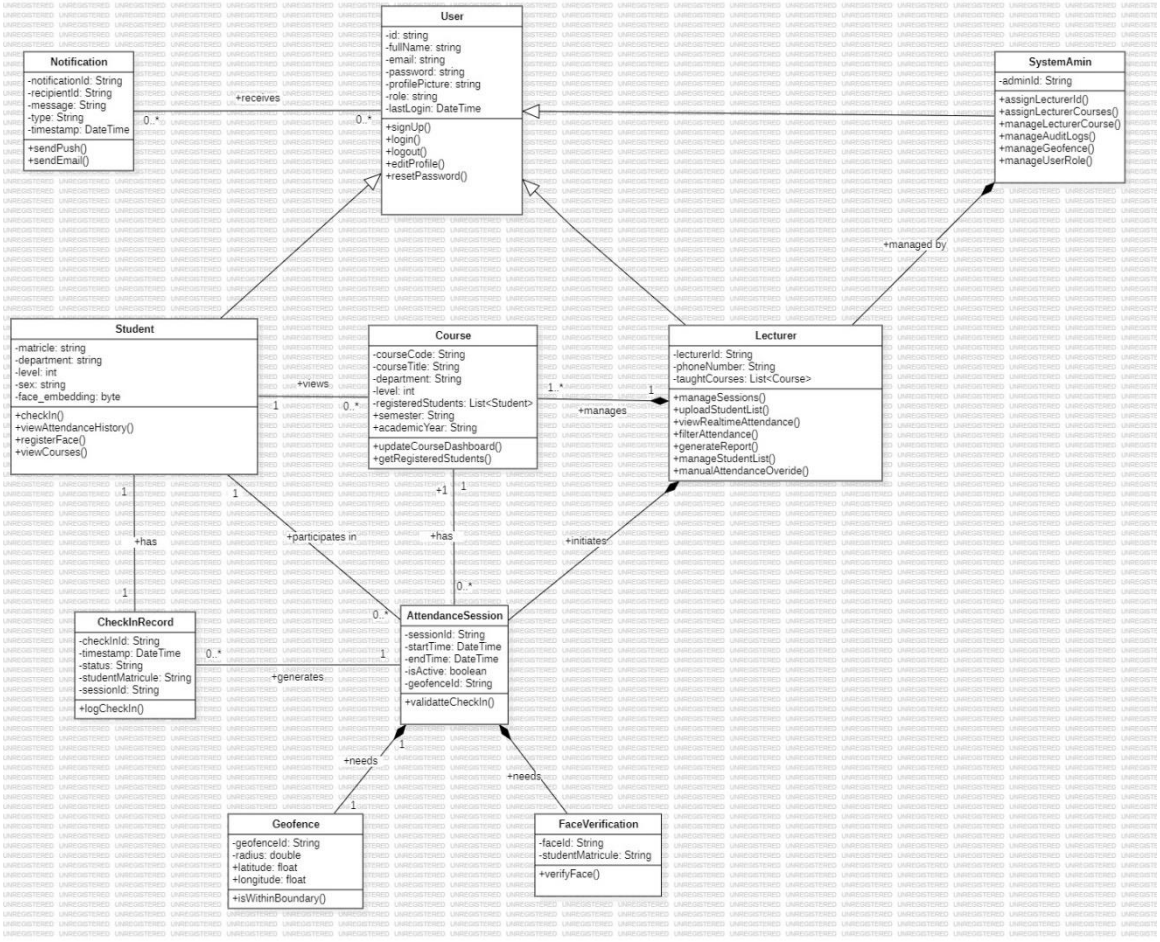

fig 5: Class Diagram

2. Design Objectives

The main objective of the class diagram is to provide a clear abstraction of all the major entities involved in the attendance system and how they interact. The design captures roles such as students, lecturers, and system administrators, and integrates attendance logic via facial recognition and geolocation enforcement. It also highlights how users are linked to check-in sessions, how courses are managed, and how facial data and geofencing data are utilized to validate presence.

3. Key Design Highlights

3.1 Role-Based Structure

The system separates concerns by defining distinct user roles with clear responsibilities. Students interact with the system primarily to perform check-ins and view their attendance history. Lecturers handle attendance

session management and student registration via course lists. A system administrator oversees role assignment, geofence creation, and course setup.

3.2 Session and Attendance Tracking

Attendance sessions are central to the system, with each session linked to a geofence that enforces physical presence during check-in. The diagram captures the real-time validation logic through associated classes like AttendanceSession, CheckInRecord, and FaceVerification.

3.3 Face and Location Validation Integration

Biometric verification and geofencing are implemented through specific classes responsible for face verification and geofence logic. This ensures that check-ins are only accepted when both face and location match the expected criteria.

3.4 Notification and Feedback Mechanism

A notification system is integrated to provide timely alerts to students and lecturers. These notifications may include session activation prompts, attendance confirmation, or reminders.

3.5 Extensibility and Maintainability

The class diagram is modular, allowing future expansion such as multi-faculty support, advanced analytics, or integration with other university systems. Each class is designed with maintainability and scalability in mind, ensuring a clean and logical architecture.

4. Class Relationships Overview

Class Relationship	**Description**
Student ↔ Course	A course maintains a list of eligible students based on uploaded data. Students don’t register manually in this system.
Lecturer ↔ Course	A lecturer is assigned to one or more courses and manages session activations and student uploads.

AttendanceSession ↔ Geofence	Every session is mapped to a specific geofence to validate physical presence during check-in.
CheckInRecord ↔ AttendanceSession ↔ Student	Logs attendance by associating a student and a session, including timestamp and attendance status.
FaceVerification → Student	Verifies student identity through live face scan before check-in, using stored biometric data.
Notification → User (Student/Lecturer)	Sends alerts (e.g. session started, missed check-in) to relevant users in real time.

VI. DEPLOYMENT DIAGRAM

The deployment diagram illustrates the system's hardware and software components, their interactions, and their distribution across different nodes. This session analyses the **Deployment Diagram** of a **"Mobile-Based Attendance Management System Based on Geofencing and Facial Recognition."**

The aim of a Deployment Diagram is to model the physical deployment of software components on hardware nodes in a system.

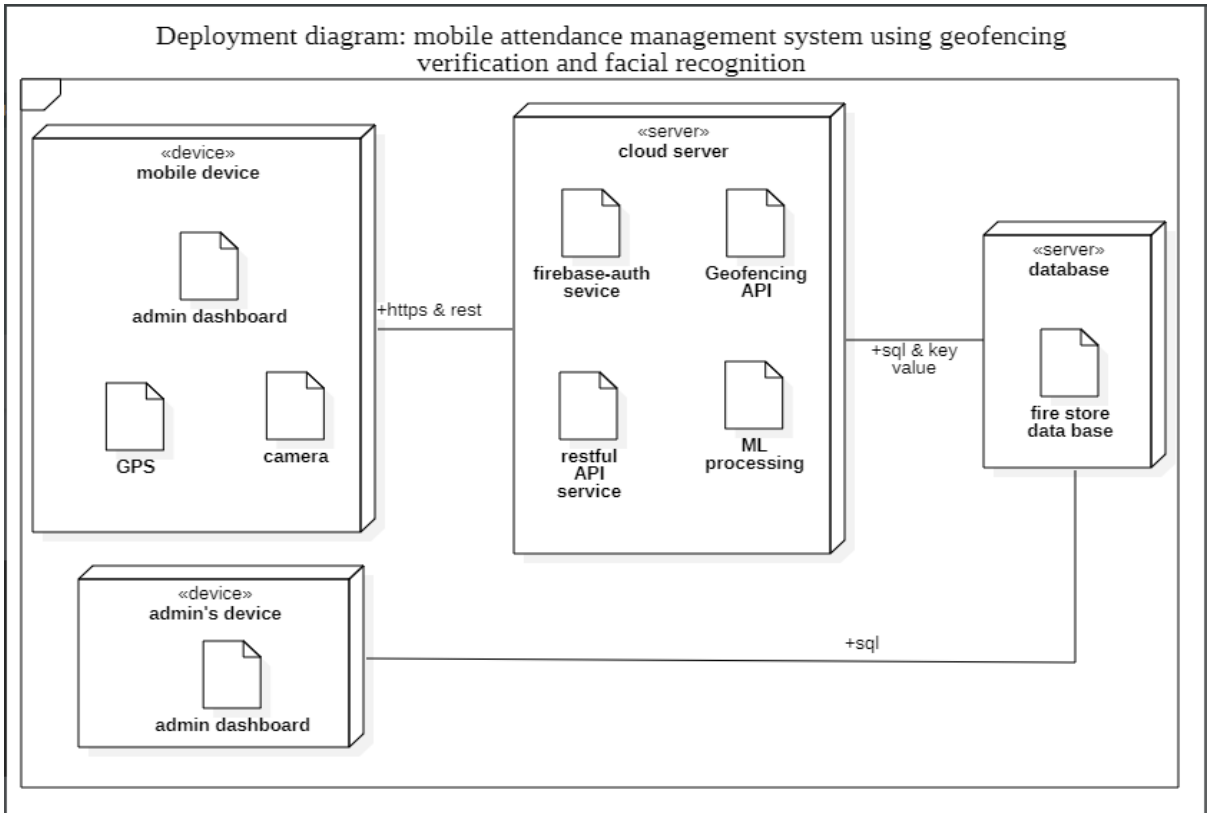

Fig: Deployment diagram

1. Aim of a deployment diagram

- ❖ Show how software (artifacts) is deployed across hardware (nodes).
- ❖ Illustrate the communication paths between devices.
- ❖ Map software components (e.g., executables, databases) to the machines they run on.
- ❖ Help in system design, especially for distributed systems, by showing runtime configuration.

In short, it visualizes the physical architecture of a system.

2. Components Description

a. Mobile Device (User Side)

- ❖ Mobile App: Installed on the user's smartphone, it provides the interface for attendance check-in.
- ❖ GPS: Used for geofencing to determine if the user is within the permitted location.
- ❖ Camera: Captures facial data for recognition.

b. Admin's Device

- Mobile App: Allows the admin to create lecturer's ID, monitor and manage attendance.

c. Cloud Server

Hosts the following services:

- Firebase Auth Service: Handles user authentication.
- Geofencing API: Validates location boundaries.
- RESTful API Service: Manages requests and responses between mobile clients and the cloud backend.
- ML Processing: Executes facial recognition using machine learning algorithms.

d. Database Server

- Firestore Database: Stores user profiles, attendance logs, GPS coordinates, and facial data metadata.

3. Deployment interaction

- Mobile Device ↔ Cloud Server: Users interact with the mobile app, which communicates with the cloud for location and identity verification.
- Cloud Server ↔ Database Server: Server-side services fetch and update records from the Firestore database.
- Admin's Device ↔ Cloud Server: The admin app communicates with the RESTful API for system control and monitoring.

4. Functionality Flow

- User opens the mobile app and attempts to check in.
- GPS verifies if the user is within the predefined geofence.
- Camera captures facial data for recognition.
- Firebase authenticates the user, and the ML service verifies the face.
- Upon successful validation, attendance is logged in Firestore.
- Admin can view attendance and assign lecturer ID via their app.

This deployment setup ensures, real-time attendance tracking, secure authentication and data storage, scalable cloud-based processing, efficient communication between components. The use of cloud services, geofencing,

and facial recognition enhances both accuracy and automation, making the system reliable for modern attendance management.

Conclusion

The system modeling and design process has provided a clear structural and functional blueprint for the SMART attendance system. Each model captures essential components, data interactions, and user behaviors, ensuring the system is scalable, secure, and efficient. These models form the foundation for implementation and guide the development of a reliable, technology-driven attendance solution

Appendix: Glossary

- **SMART** Facial Recognition **Attendance System**: A system utilizing biometric technology for attendance tracking.
- **Biometric Data**: Unique physical characteristics used for identification, such as facial features.
- **Context Diagram**: A high-level view showing a system and its interactions with external entities.
- **Data Flow**: Movement of data between the system and external entities.
- **External Entity**: Any person or system outside the system boundary that interacts with the system.