Group Name: BOYS AT THE BACK
Members:

Rudelito Dongiapon

Jhon Ryan Pagantian

Christian Villanueva

Francis Al Capińa

Project X: Automated Attendance System

SYSTEM REQUIREMENTS

R1. User Roles and Access

R1.1. The system has three main roles: Lecturer, Student, and Administrator.

R1.2. Only registered lecturers can record attendance.

R1.3. Attendance can only be recorded using authorized devices.

R1.4. Administrators have full control over system data, including managing students, lecturers, devices, and courses.

R2. Attendance Recording & Data Storage

R2.1. Lecturers can record attendance via registered devices (mobile, tablet, or computer).

R2.2. Attendance records are stored in a cloud-based MySQL database.

R2.3. Each attendance entry is linked to a course and includes details such as student, lecturer, date, and time.

R2.4. Attendance records can be accessed in real-time.

R3. Device Registration & Tracking

R3.1. Lecturers can register multiple devices for attendance tracking.

R3.2. The system can track device locations for assistance.

R3.3. Only registered devices can interact with the attendance system.

R4. Student Enrollment & Management

R4.1. Students can enroll in courses through the system.

R4.2. The system maintains student details, including name, university ID, and profile picture (captured via the system).

R4.3. Administrators can add, update, and delete student records.

R5. Reporting & Data Access

R5.1. Reports available include:

R5.1.1. Attendance reports by student, course, or date range.

R5.1.2. Student enrollment per course.

R5.1.3 Lecturer and registered device lists.

R5.2. Reports are accessible to lecturers and administrators.

R6. Photo Capture & Storage

R6.1. Lecturers can capture student photos for attendance verification.

R6.2. Photos are stored securely, with IDs and names recorded in the database.

R7. System Architecture & API Access

R7.1. The system operates via a REST API for database interactions.

R7.2. All actions, including attendance, enrollment, and reporting, are managed through the API.

R7.3. API requests are authenticated and authorized.

R8. Testing Criteria

R8.1. **Unit Testing:** Each function (e.g., attendance logging, student enrollment) is tested individually.

R8.2. **System Testing:** End-to-end testing ensures all workflows function correctly.

R8.3. **User Acceptance Testing (UAT):** The system is tested against user requirements.

SYSTEM COMPONENTS

Frontend (Client-Side)

- A web and mobile app (built using React or Flutter) provides interfaces for lecturers, students, and administrators.
- A camera module captures student photos.

Backend (Server-Side)

- **REST API (Node.js/Express):** Handles requests from the frontend. .

Authentication Module: Manages user logins.

- **Attendance Processing Module:** Records attendance.
- **Report Generation Module:** Creates attendance reports. .

Device Tracking Module: Tracks registered devices.

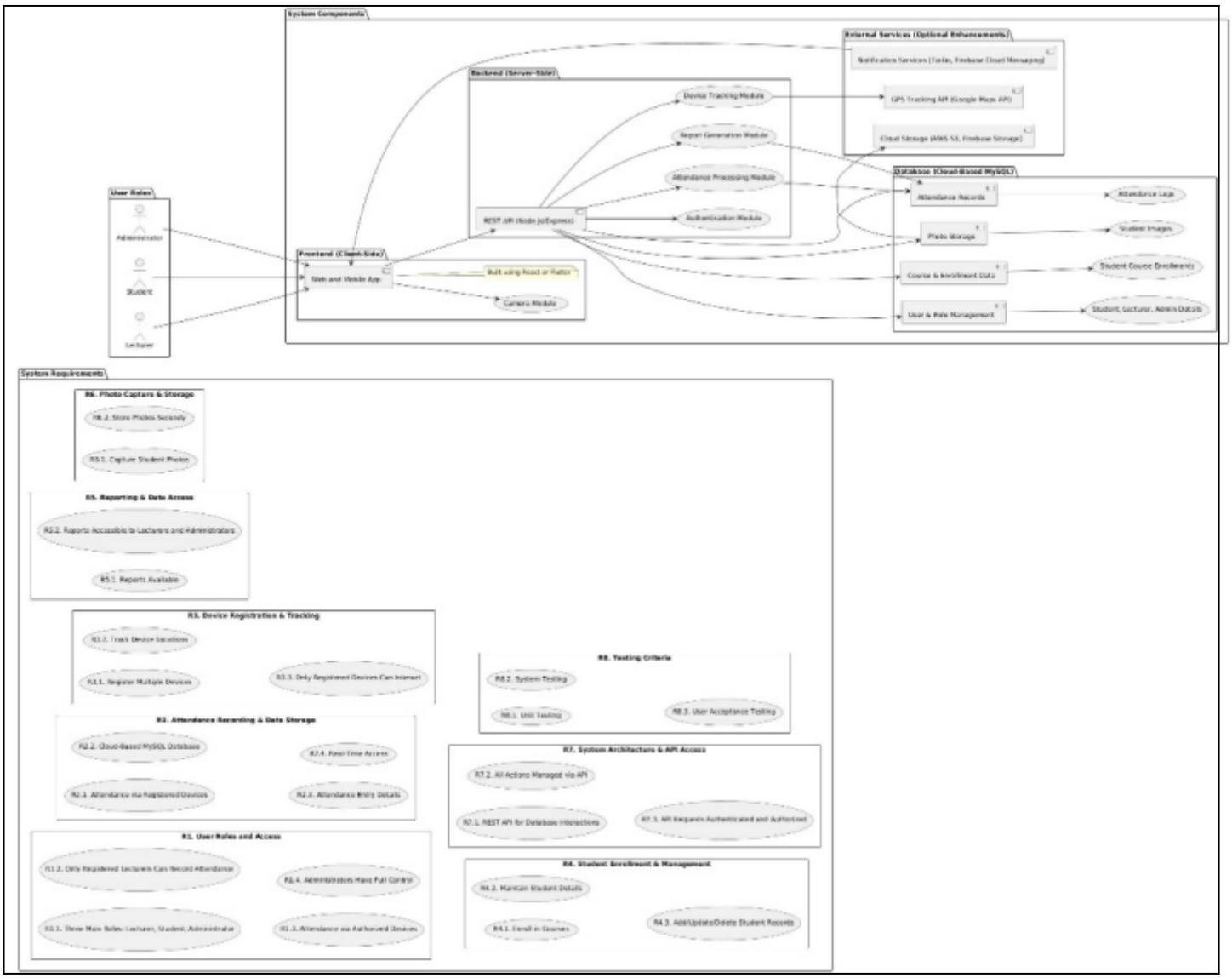
Database (Cloud-Based MySQL)

- **User & Role Management:** Stores student, lecturer, and admin details. .
- **Course & Enrollment Data:** Manages student-course enrollments.
- **Attendance Records:** Maintains attendance logs.
- **Photo Storage:** Saves student images for verification.

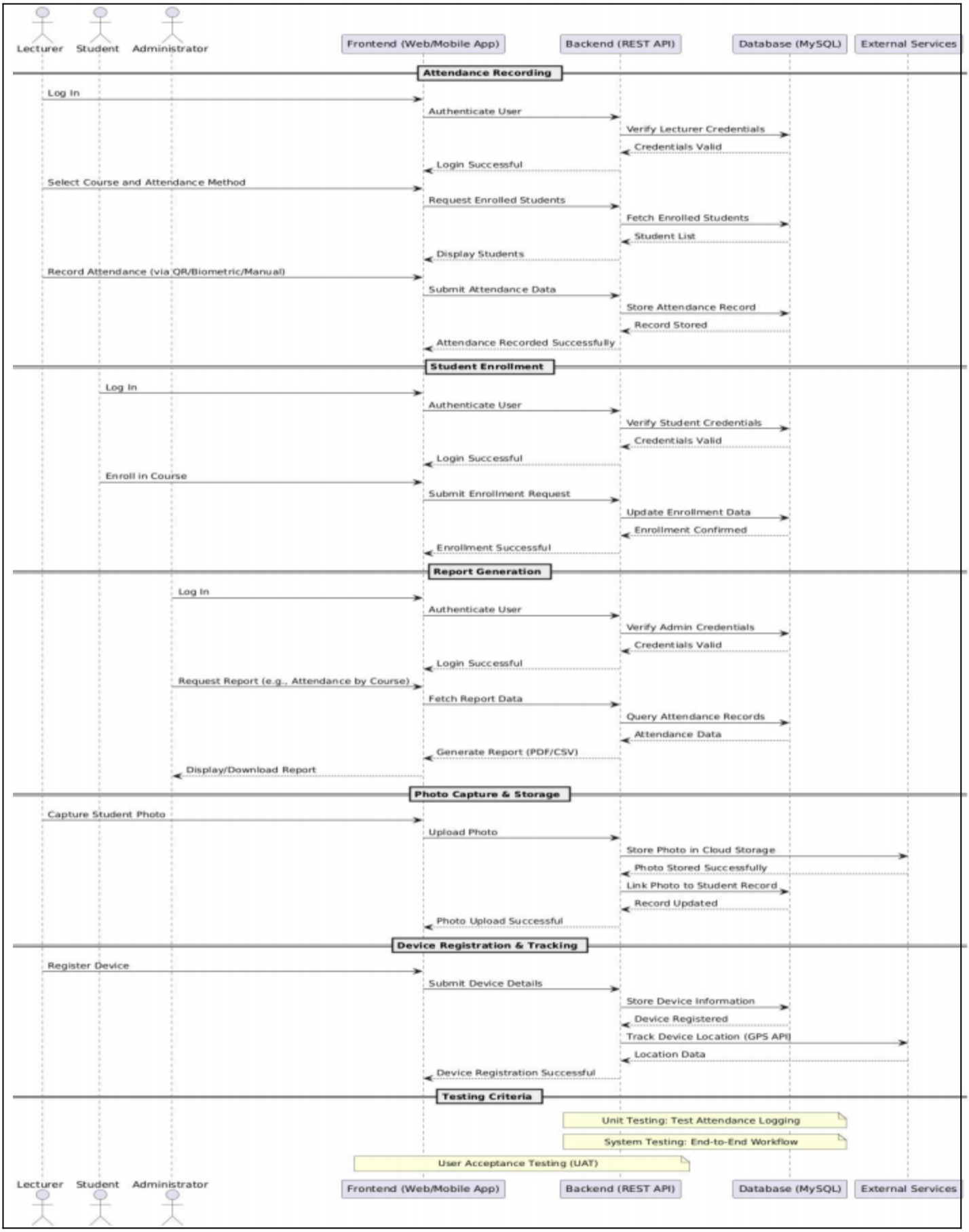
External Services (Optional Enhancements)

- **Cloud Storage (AWS S3, Firebase Storage):** Stores student images. .
- **GPS Tracking API (Google Maps API):** Locates lecturer devices.
- **Notification Services (Twilio, Firebase Cloud Messaging):** Sends attendance reminders.

CONCEPTUAL MODEL



HIGH LEVEL SEQUENCE DIAGRAM



USE CASE DIAGRAM

1. Mark Student Attendance

Actors:

- Primary: Lecturer
- Secondary: Student, Administrator

Preconditions:

- Lecturer is logged in.
- A class session is scheduled. .

Students are registered.

Main Flow:

1. Lecturer logs in and selects a course.
2. The system displays enrolled students.
3. Lecturer selects an attendance method (QR code, biometric, manual).
4. Students mark their attendance.
5. The system validates student presence and updates records.
6. Lecturer submits attendance.
7. Data is stored in the database, and reports are generated.

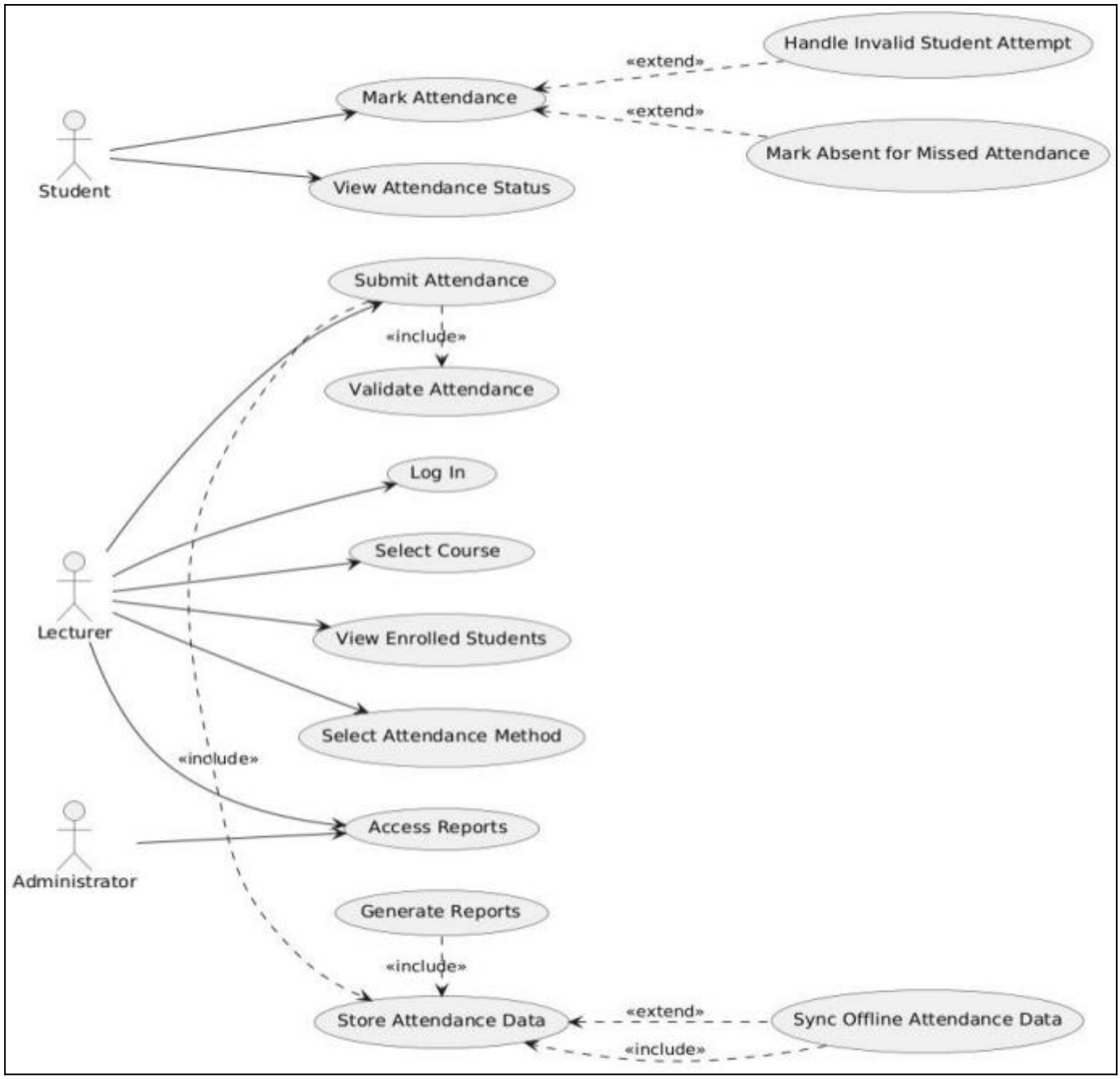
Alternate Flows:

- **Invalid Student Attempt:** Unauthorized students cannot mark attendance.
- **Missed Attendance:** Students who do not mark attendance within the timeframe are marked absent.
- **Offline Mode:** Attendance data is stored locally and synced later.

Postconditions:

- Attendance is recorded in the database.
- Lecturers and administrators can access reports. .

Students can view their attendance status.



---

2. Generate Attendance Report

Actors:

. Primary: Administrator .

Secondary: Lecturer

Preconditions:

. The user must be logged in.

. Attendance records must exist.

Main Flow:

1. The user logs in and navigates to the Reports section.
2. Filters are applied (by course, date range, student, or lecturer).
3. The report is generated in table, PDF, or CSV format.
4. The user can download or print the report.

Alternate Flows:

- **Invalid Date Range:** If no records exist for the selected range, the user is prompted to adjust filters.
- **Export Options:** Reports can be exported in multiple formats.
- **Access Restriction:** Lecturers can only generate reports for their assigned classes, while administrators have full access.

Postconditions:

- Reports are generated and ready for download.
- Users can use reports for record-keeping or analysis.