**Group Name:** Team Man-E

**Members:**

Jomari E. Gamao

Richard Fuertes

Richard Daganio

**Project X: Automated Attendance System**

**R1. Roles and Permissions Management**

R1.1. The platform will accommodate three core user roles: Instructor, Student, and Admin.

R1.2. Only validated instructors will have the ability to record attendance.

R1.3. Attendance logging will only be permitted from authorized devices.

R1.4. Administrators will have unrestricted access to manage all system data, including the ability to create, read, update, and delete information related to students, instructors, devices, and courses.R1.2. Only registered lecturers shall be able to record attendance.

**R2. Attendance Logging and Data Management**

R2.1. Instructors will be able to register student attendance via any authorized mobile device, tablet, or computer.

R2.2. The attendance information will be securely stored in a cloud-hosted MySQL database.

R2.3. Each attendance entry will be associated with a specific course, and it will include details like the student, instructor, date, and time of the session.

R2.4. Attendance records will be immediately available for retrieval.

**R3. Device Registration and Monitoring**

R3.1. Instructors will have the ability to register multiple devices for attendance tracking.

R3.2. The platform will track the locations of registered devices to assist instructors when necessary.

R3.3. Only registered devices will be authorized to access the attendance system.

**R4. Enrollment and Student Record Management**

R4.1. Students will be able to enroll in courses through the platform.

R4.2. The system will maintain and organize student information, including:

R4.2.1. Name

R4.2.2. University ID Number

R4.2.3. Profile Picture (captured via the platform)

R4.3. The platform will allow for the addition, modification, and deletion of student data.

**R5. Report Generation and Data Accessibility**

R5.1. The platform will generate various types of reports, including:

R5.1.1. Attendance records (by individual student, course, or specific date range).

R5.1.2. A roster of students enrolled in each course.

R5.1.3. A list of instructors and the devices they have registered.

R5.2. Reports will be available to both instructors and administrators.

**R6. Photo Capture and Storage**

R6.1. The platform will enable instructors to capture student photos for attendance records.

R6.2. These photos will be stored as image files, with corresponding student IDs and names stored in the database.

R6.3. The platform will implement secure access controls to protect these images.

**R7. System Infrastructure and API Interaction**

R7.1. The platform will use a RESTful API to manage interactions with the underlying database.

R7.2. All user interactions (e.g., attendance recording, course enrollment, report generation) will be handled through API endpoints.

R7.3. API requests will be subject to proper authentication and authorization processes.

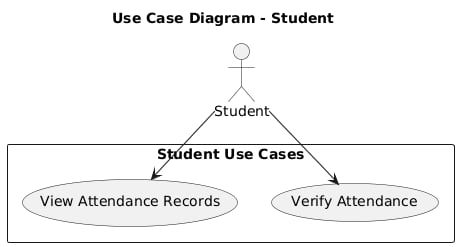
**8. Testing Procedure**

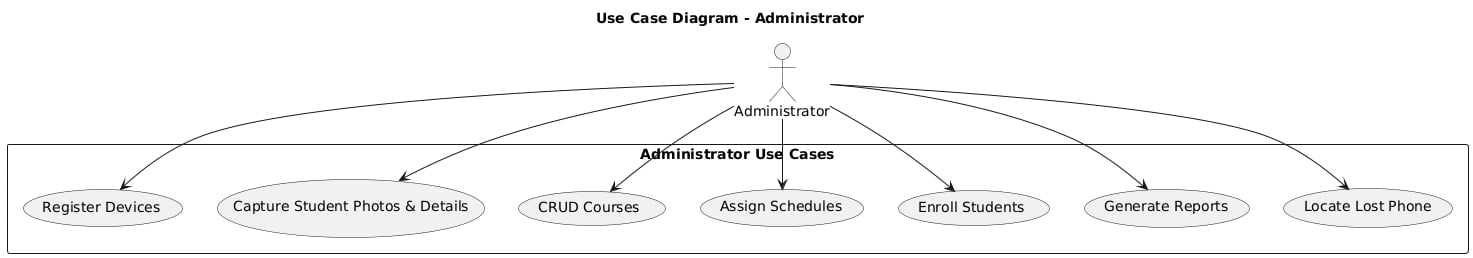
R8.1. Unit Testing: Individual functions (e.g., logging attendance, enrolling students) will be tested in isolation.

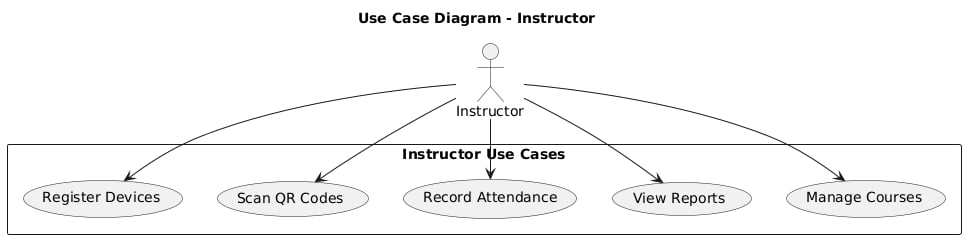
R8.2. System Testing: The entire system will be tested end-to-end to ensure that all workflows and features work as expected.

R8.3. User Acceptance Testing (UAT): The platform will undergo testing to ensure it meets user requirements and expectations, based on the above criteria.

**Use Case Diagram**



****

****

### ****Use Case Scenario: Automated Attendance System****

**Actors:**

1. **Lecturer**:  
   The Lecturer is the primary user who interacts with the system to perform tasks related to managing attendance, selecting courses, and generating attendance reports.
2. **Student:**The Student is the secondary actor who marks their own attendance, either manually or using some automated method (like QR codes or biometrics). They can also view their attendance record.
3. **Administrator:**The Administrator is the user with the highest level of control in the system. They can manage user roles (Lecturers and Students), register devices, manage courses, and perform system configurations such as adding new lecturers, students, or devices to the system.

**Use Cases:**

1. **Login (UC1):**
   * Actors Involved: Lecturer, Student, Administrator
   * Every user (Lecturer, Student, Administrator) must log in to the system to access its features. This use case represents the login process where users provide their credentials to access the system.
2. **Select Course & Session (UC2):**
   * Actors Involved: Lecturer
   * After logging in, the Lecturer selects the course and class session for which they will mark attendance. This is essential as the system needs to know which class and session the lecturer is recording attendance for.
3. **Mark Attendance (UC3):**
   * Actors Involved: Lecturer, Student
   * The Lecturer records attendance for each student in the selected course and session. The Student marks their own attendance by interacting with the system, possibly through QR codes, biometrics, or manually. Once a student marks attendance, it gets stored in the system, and the Lecturer can verify and submit it.
4. **Generate Attendance Report (UC4):**
   * Actors Involved: Lecturer, Administrator
   * Once attendance is recorded, the Lecturer or Administrator can generate a report showing the attendance of students for a given course, session, and date range. This is useful for tracking attendance trends, generating certificates, or addressing student concerns.
5. **View Attendance (UC5):**
   * Actors Involved: Student, Lecturer
   * Students can view their own attendance records through the system, allowing them to track their attendance over time. Lecturers can also access this data to monitor student attendance in their courses.
6. **Register Student (UC6):**
   * Actors Involved: Administrator
   * The Administrator is responsible for registering Students in the system. This is done when a new student joins a course, or a new student needs to be added to the system for attendance tracking.
7. **Register Lecturer (UC7):**
   * Actors Involved: Administrator
   * The Administrator registers Lecturers in the system, enabling them to access their assigned courses and record attendance.
8. **Register Device (UC8):**
   * Actors Involved: Administrator
   * The Administrator registers devices (e.g., mobile devices, tablets, or computers) that will be used to mark attendance. Only registered devices can interact with the attendance system.
9. **Manage Courses (UC9):**
   * Actors Involved: Administrator
   * The Administrator manages the list of courses offered, which includes adding, editing, or deleting courses. This is crucial because students need to be enrolled in courses, and lecturers need to assign attendance to the correct courses.
10. **Manage Devices (UC10):**

* Actors Involved: Administrator
* The Administrator manages the devices registered in the system. This includes keeping track of which devices are registered and ensuring they are available for attendance recording.

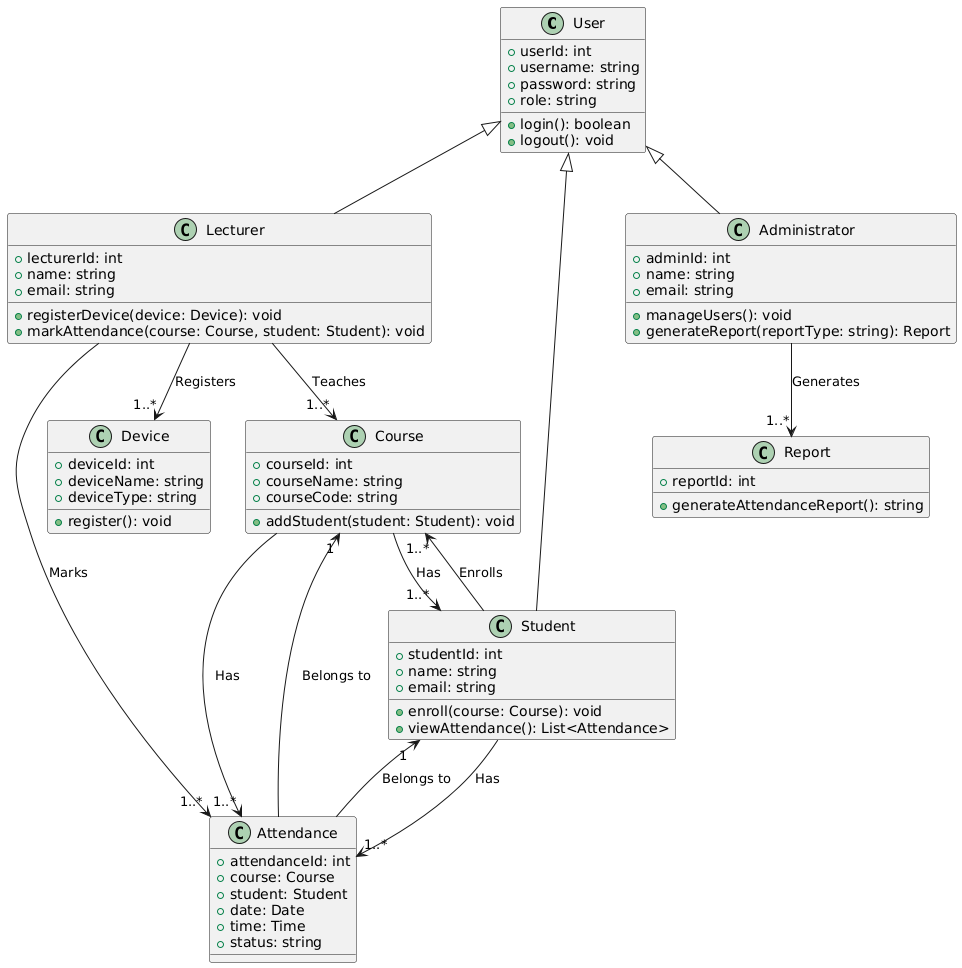
**Relationships:**

1. Lecturer - Login, Select Course & Session, Mark Attendance, Generate Attendance Report, View Attendance:  
   The Lecturer is the primary user responsible for logging in, selecting courses, marking attendance, and generating attendance reports. The Lecturer interacts with these use cases regularly during their teaching duties.
2. Student - Mark Attendance, View Attendance:  
   The Student has a more limited role in this system. They can mark their own attendance (using methods like QR code scans or biometrics) and view their attendance record.
3. Administrator - Register Student, Register Lecturer, Register Device, Manage Courses, Manage Devices:  
   The Administrator has full administrative rights over the system. They can manage users (students and lecturers), register devices, and manage courses, which are essential functions for maintaining the system.

**Include and Extend Relationships:**

* Include (UC1 → UC2):  
  The Login use case is included in the Select Course & Session use case. This means that before the Lecturer can select a course or session, they must first log in. The login is an essential step to access the system’s features.
* Extend (UC3 → UC4):  
  The Mark Attendance use case extends the Generate Attendance Report use case. This means that once attendance is marked for a class, generating the attendance report is a logical next step, although it's optional and could happen at a later time.

**Conceptual UML Class Diagram**

****

**Frontend (Client Side)**

* **Web & Mobile App (React/Flutter)**: This is the interface through which lecturers, students, and administrators interact with the system. The app allows users to mark attendance, view reports, and manage student data, depending on their role.
* **Camera Module**: This feature allows lecturers to capture passport-style photos of students for attendance records and profile management.

**Backend (Server Side)**

* **REST API (Node.js/Express)**: The API serves as the main bridge between the frontend and the backend, handling all requests from the client-side. It processes actions like logging attendance, enrolling students, and generating reports.
* **Authentication Module**: This module is responsible for managing user authentication, such as logging in, logging out, and validating user roles (Lecturer, Student, Admin) based on the credentials provided.
* **Attendance Processing Module**: This module processes the attendance data, including recording attendance and linking it to the relevant student, course, and date.
* **Report Generation Module**: This module handles the creation of various reports such as attendance by student, course, and date range. It generates downloadable reports for lecturers and administrators.
* **Device Tracking Module**: This module tracks the location of registered devices used by lecturers for attendance recording. It helps in managing devices and ensures that only authorized devices can access the system.

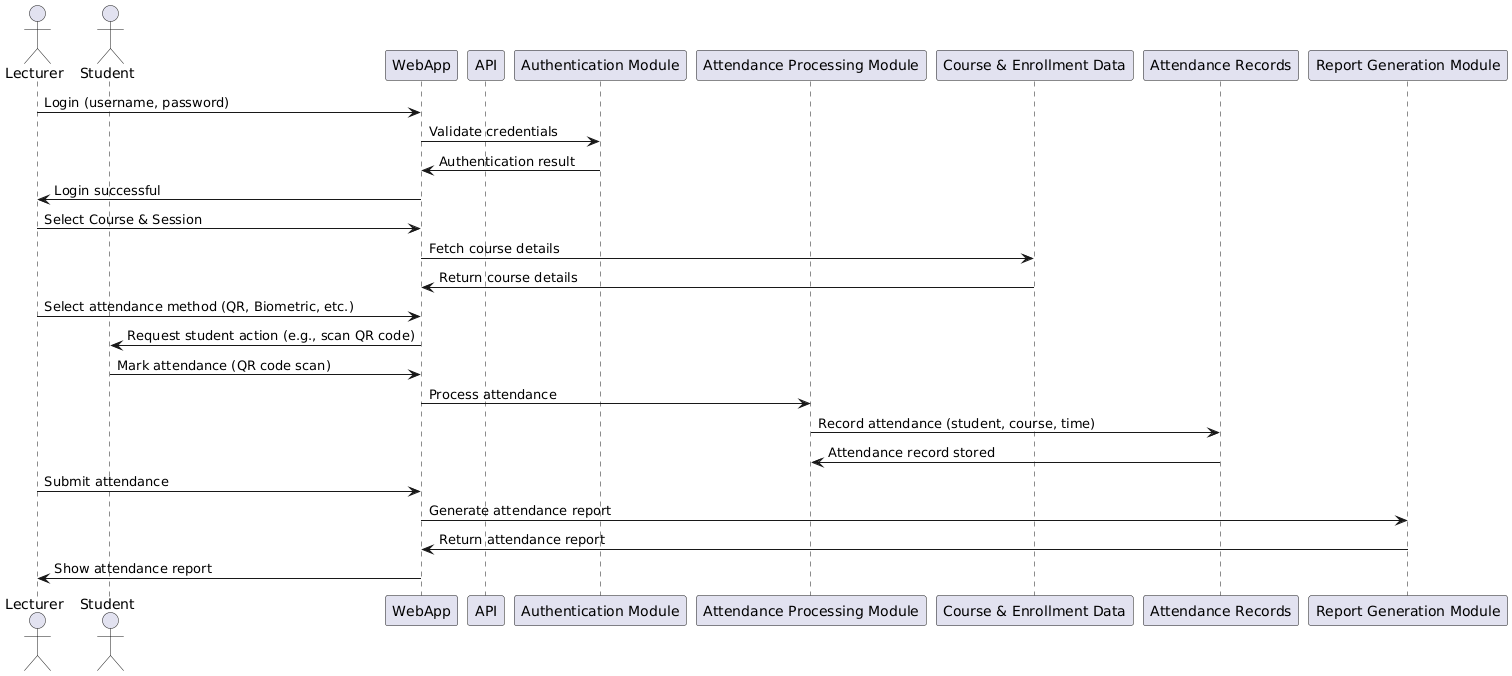
**Database (Cloud-Based MySQL)**

* **User & Role Management**: This part of the database stores all user data, including students, lecturers, and administrators. It also manages user roles, ensuring that each user has the appropriate permissions.
* **Course & Enrollment Data**: This stores information on all courses offered in the system, and tracks which students are enrolled in which courses.
* **Attendance Records**: This database component stores all attendance data, including student names, dates, courses, and attendance status (e.g., present, absent, late).
* **Photo Storage**: This stores the student profile photos captured by the system for attendance verification.

**External Services (Optional Enhancements)**

* **Cloud Storage (AWS S3, Firebase Storage)**: This external service can be used to store student photos in a scalable and secure manner.
* **GPS Tracking API (Google Maps API)**: This service helps track the physical location of the registered devices used by lecturers. It can be used to verify the lecturer's presence at a specific location or assist them if needed.
* **Notification Service (Twilio, Firebase Cloud Messaging)**: This external service sends notifications to users (lecturers, students, and administrators). For example, it could send reminders for attendance or notify users of system updates.

**Sequence Diagrams**

****

### ****Sequence Scenario: Automated Attendance System****

**Actors:**

* Primary Actor: Lecturer
* Secondary Actors: Student, Administrator

**Preconditions:**

* The Lecturer must be logged into the system.
* The class session for which attendance is being recorded must already be scheduled in the system.
* All students must be registered in the system and linked to the course.

**Main Flow:**

1. **Lecturer Logs In:**
   * The Lecturer opens the WebApp and enters their login credentials.
   * The WebApp sends the credentials to the Authentication Module for validation.
   * The Authentication Module authenticates the credentials and returns the login result to the WebApp, which allows the Lecturer to access the system.
2. **Select Course and Session:**
   * After successful login, the Lecturer selects the course and class session for which they wish to mark attendance.
   * The WebApp sends a request to the Course & Enrollment Data to retrieve details for the selected course.
   * The Course & Enrollment Data responds with the course details, and the WebApp displays them to the Lecturer.
3. **Select Attendance Marking Method:**
   * The Lecturer chooses the method by which students will mark their attendance (e.g., QR code, Biometric, Manual).
   * The WebApp then prompts the Student to mark attendance using the selected method.
4. **Student Marks Attendance:**
   * The Student interacts with the WebApp (e.g., scans a QR code or provides a biometric identifier) to mark their attendance.
   * The WebApp sends the attendance data to the Attendance Processing Module for validation and recording.
5. **System Validates Student's Presence:**
   * The Attendance Processing Module checks if the Student is registered in the selected course.
   * If the Student is valid, their attendance is recorded in the Attendance Records database.
6. **Real-Time Update:**
   * The Attendance Records database updates in real-time with the student's attendance status (present, late, or absent).
   * The WebApp continuously updates the attendance record for the Lecturer to monitor.
7. **Lecturer Submits Attendance Record:**
   * Once all students have marked their attendance, the Lecturer submits the final attendance record.
   * The WebApp sends the completed data to the Attendance Processing Module, which stores it in the Attendance Records database.
8. **Generate Attendance Report:**
   * After submitting the attendance record, the WebApp requests the Report Generation Module to generate the attendance report.
   * The Report Generation Module formats the report based on the selected course, date, and student details.
9. **Display and Download Report:**
   * The WebApp displays the generated attendance report to the Lecturer, who can review the data.
   * **The Lecturer may download or print the report for record-keeping.**

**Alternate Flows:**

* **Invalid Student Attempt:** 
  + If a Student attempts to mark attendance but is not registered for the selected course, the system will deny access and notify the Lecturer of the issue.
* **Missed Attendance:** 
  + If a Student fails to mark attendance within the given timeframe (e.g., during the class session), the system will automatically mark them as absent.
* **Offline Mode:** 
  + If the internet connection is lost during the attendance process, the WebApp stores the attendance data locally. Once the connection is restored, the system will synchronize the stored data with the server.

**Postconditions:**

* The Attendance Records are successfully stored in the database.
* The Lecturer and Administrator can access and view the attendance data in the generated report.
* Students can view their attendance status and track their attendance history.