**Project X: Automated University Attendance System**

**Refined and Structured Requirements**

**System Overview**

The university requires an **Automated Attendance System** that allows instructors to record students' attendance using registered devices (phones, tablets, or computers). The system should store attendance data on a cloud-based **MySQL relational database** accessed via a **REST API**. Additionally, the system should support **student image capture**, device **location tracking**, and **comprehensive reporting** functionalities.

**Functional Requirements**

**2.1 User Management**

* The system shall allow **lecturers** to register their devices.
* Only **registered devices** shall be used to mark attendance.
* Each lecturer shall have **one or more** registered devices.
* There shall be multiple lecturers in the system.
* The system shall allow adding, editing, and deleting **lecturers**.

**2.2 Student Management**

* The system shall allow adding, updating, viewing, and deleting **students**.
* Each student shall have a **University ID**, **name**, and **passport-style photo** stored in the database.
* The system shall enable capturing a **photo** of the student during attendance marking.

**2.3 Course and Enrollment Management**

* The system shall allow adding, updating, viewing, and deleting **courses**.
* The system shall allow enrolling students into **courses**.
* The system shall maintain a record of **which students are in which courses**.

**2.4 Attendance Management**

* The system shall allow instructors to **mark attendance** using their registered devices.
* Attendance data shall be **stored in the cloud database**.
* The system shall allow viewing and exporting **attendance reports**.

**2.5 Device Tracking**

* The system shall track the **location of registered devices** in case the instructor is lost.

**2.6 Reporting and Data Management**

* The system shall allow generating reports on:
  + **Student attendance**
  + **List of students**
  + **Lecturers**
  + **Courses and enrollments**
* The system shall allow exporting reports in **CSV or PDF format**.

**3. Non-Functional Requirements**

* **Cloud-Based Storage:** MySQL relational database hosted in the cloud.
* **API Accessibility:** Attendance data shall be accessed through a **REST API**.
* **Security:** Authentication and authorization mechanisms for users.
* **Scalability:** The system should support **multiple concurrent users**.

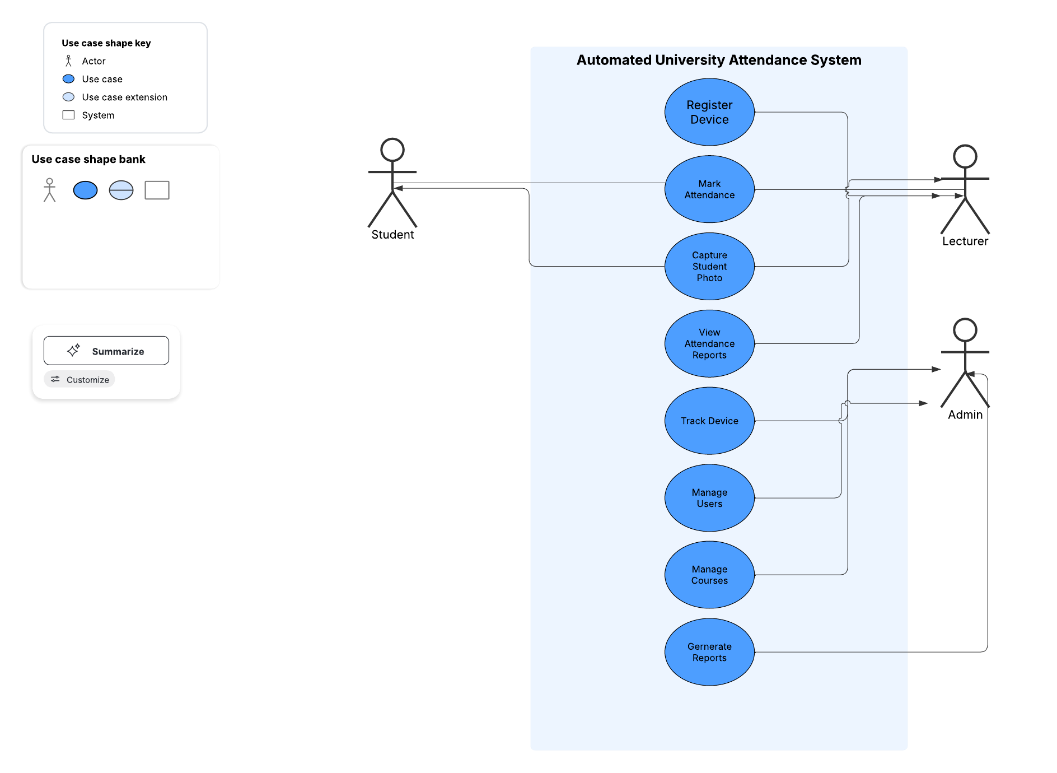
**4. Testing Requirements**

The system shall undergo the following testing phases:

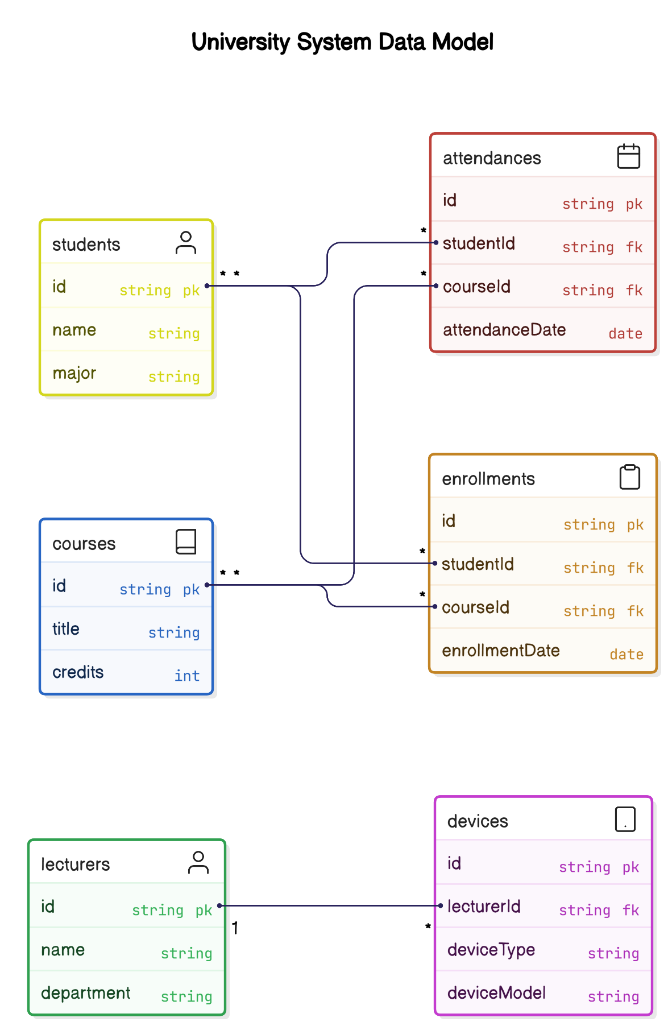
1. **User Acceptance Testing (UAT):** Ensuring the system meets end-user expectations.
2. **System Testing:** Validating the entire system workflow.
3. **Unit Testing:** Testing individual components where applicable.

**5. UML Diagrams**

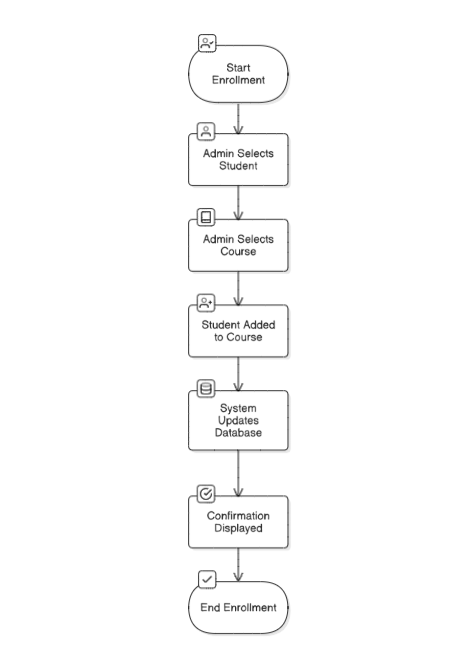
**5.1 Use-Case Diagram**



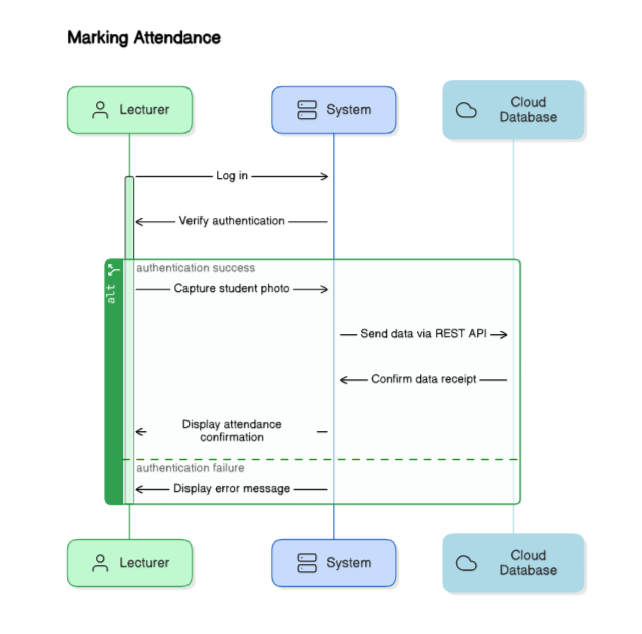
**5.2 Class Diagram**

****

**5.3 Activity Diagram**

**Student Enrollment Process**

**5.4 Sequence Diagram**

****

**6. Agile Development Plan**

**6.1 Project Team Structure**

* **Team Size:** 3 members
* **Rotating Team Leader** for each phase

**6.2 Development Stages and Sprint Plan**

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Task** | **Deliverables** |
| Sprint 1 | Requirements & Use Cases | Finalized Requirements & Use-Case Diagrams |
| Sprint 2 | High-Level Design | Conceptual & Component Architecture |
| Sprint 3 | Development | Core functionalities implementation |
| Sprint 4 | Testing | Unit, System, and UAT |
| Sprint 5 | Final Demonstration | Working System & Documentation |

**7. Evaluation Criteria**

|  |  |  |
| --- | --- | --- |
| **Stage** | **Criteria** | **Weight** |
| 1 | Requirements Elicitation & Documentation | 10% |
| 2 | High-Level Design (UML & Architecture) | 15% |
| 3 | Detailed Design (Class, Sequence, Activity Diagrams) | 15% |
| 4 | Review Process (Design, Code) | 10% |
| 5 | Agile Practices (Sprints, Standups, Demos) | 10% |
| 6 | TDD Demonstration | 10% |
| 7 | Final Presentation & Demo | 30% |
| **Total** | **100%** |  |

**8. Conclusion**

The project aims to deliver a **professional, scalable, and secure** university attendance system. It will follow agile practices, use **cloud-based storage**, and ensure **robust testing** at every stage. The focus is on developing a **functional prototype** while ensuring students gain **real-world software development experience**.