**Requirements for Project X – Automated Attendance System**

**1. System Overview**

The system shall automate student attendance tracking using mobile devices, ensuring secure, real-time data storage and retrieval via a cloud-based MySQL database with REST API access.

**2. Functional Requirements**

**2.1 User Roles & Access**

* The system shall support **three** user roles:
  1. **Administrator** – manages users, devices, courses, and reports.
  2. **Lecturer** – records student attendance, manages their own registered devices, and takes student photos.
  3. **Student** – is identified by the system and has their attendance recorded.

**2.2 Device Registration & Authentication**

1. A **lecturer** must register a device (phone, tablet, or computer) before it can be used for attendance tracking.
2. A **lecturer** may register **multiple** devices, but each device must be uniquely identifiable.
3. Only **registered** devices shall be allowed to access the attendance system.

**2.3 Attendance Tracking**

1. The system shall **automatically** or **manually** record student attendance when a student enters the classroom.
2. Each attendance record shall include the **student’s ID, name, timestamp, and course details**.
3. Attendance data shall be **immediately stored** in the cloud database.
4. The system shall provide an option for **manual override** in case of errors.

**2.4 Student Identification & Photo Capture**

1. The system shall support **photo capture** of students using the lecturer’s registered device.
2. The photo shall be stored as a **file**, while the student’s **ID and name** are stored in the database.
3. The system shall enforce **passport-style guidelines** for student photos.

**2.5 Location Tracking (Optional Feature)**

1. The system **may** allow lecturers to track the real-time location of their registered device in case it is lost.
2. Location data shall be **securely stored and accessible** only to authorized users.

**2.6 Reporting & Data Management**

1. The system shall generate **attendance reports**, listing:
   * Students present/absent per session
   * Overall attendance trends
   * Lecturers and their assigned courses
   * Students enrolled in each course
2. Users shall be able to **add, edit, delete, and view** all items in the system, including students, lecturers, courses, and attendance records.

**2.7 System Access & API Integration**

1. The system shall use a **REST API** for all database operations.
2. Access control shall be implemented using **secure authentication mechanisms**.
3. The database shall be **cloud-based using MySQL**.

**3. Testing Requirements**

Testing shall be performed at **three levels**:

1. **User Acceptance Testing (UAT):** Ensuring that the system meets business requirements and client expectations.
2. **System Testing:** Verifying system-wide functionality, including API interactions and database operations.
3. **Unit Testing:** Testing individual components where applicable to ensure proper functionality.
4. **System Development Life Cycle**

**Review Process at Each Stage in the Project Life Cycle**

**1. Requirements Review**

**Objective:**

Ensure that all system requirements are clearly defined, feasible, and aligned with project goals.

**Activities:**

1. **Stakeholder Consultation:** Gather input from administrators, lecturers, and students.
2. **Feasibility Analysis:** Verify technical and business feasibility.
3. **Requirements Validation:** Cross-check with university policies and IT infrastructure.
4. **Review Meeting:** Document feedback and refine requirements.

**Deliverables:**

1. Approved Requirements Document
2. Functional and Non-Functional Requirements Specification
3. Use Case Diagrams

**2. Design Review**

**Objective:**

Validate that the high-level and detailed designs meet the system requirements.

**Activities:**

1. **High-Level Design (HLD) Review:** Assess conceptual architecture and system components.
2. **Detailed Design Review:** Verify UML diagrams and data flow models.
3. **Security & Performance Assessment:** Ensure compliance with authentication and API security protocols.
4. **Design Walkthroughs:** Identify potential design flaws before implementation.

**Deliverables:**

1. Approved High-Level and Detailed Design Documents
2. UML Diagrams (Use Case, Class, Sequence, etc.)
3. Database Schema Design

**3. Code Review**

**Objective:**

Ensure that code follows best practices, meets requirements, and is free of major defects.

**Activities:**

1. **Peer Code Review:** Conduct systematic examination of written code.
2. **Automated Code Analysis:** Utilize static code analysis tools.
3. **Unit Testing:** Verify individual components work correctly.
4. **Security Review:** Identify and fix potential vulnerabilities.
5. **Coding Standards Compliance:** Ensure adherence to style guides and best practices.

**Deliverables:**

1. Code Review Reports
2. Unit Test Results
3. Updated Codebase with Fixes

**4. Testing Review**

**Objective:**

Verify that the system functions correctly and meets quality standards.

**Activities:**

1. **Unit Testing Review:** Assess coverage and effectiveness.
2. **System Testing Review:** Ensure API, database, and authentication mechanisms function correctly.
3. **User Acceptance Testing (UAT) Review:** Validate against business needs and user expectations.
4. **Performance & Security Testing:** Identify bottlenecks and vulnerabilities.

**Deliverables:**

1. Test Cases and Results
2. Bug Tracking Reports
3. UAT Approval Sign-off

**5. Deployment & Maintenance Review**

**Objective:**

Ensure smooth deployment and establish a maintenance plan.

**Activities:**

1. **Deployment Checklist Review:** Verify readiness for production release.
2. **Monitoring & Logging Setup:** Ensure logging mechanisms are in place.
3. **Incident Response Plan Review:** Define support and escalation processes.
4. **Post-Deployment Evaluation:** Collect user feedback and prioritize enhancements.

**Deliverables:**

1. Deployment Report
2. Maintenance & Support Plan
3. Post-Implementation Review Report

**Conclusion**

Following this structured review process ensures a robust, secure, and user-friendly Automated Attendance System. By integrating Agile methodologies, iterative improvements, and continuous validation, the project remains aligned with stakeholder