**Project X - Automated Attendance System Documentation**

**1. Introduction**

**1.1 Project Overview**

The University requires an automated attendance system that identifies students via registered devices (phones, tablets, or computers). Attendance data is stored in a cloud-based MySQL database, accessible via a REST API. Features include CRUD operations, reports, photo capture, and device location tracking.

**1.2 Objectives**

* Automate attendance tracking.
* Store data securely in the cloud.
* Generate reports for students, lecturers, and courses.
* Implement photo storage and role-based access control.

**2. Functional Requirements**

**2.1 Device Registration & Attendance Tracking**

* Only registered devices can record attendance.
* Attendance records include timestamp, student ID, course, and lecturer.
* Data is accessible by student, course, and lecturer.

**2.2 Database & API Access**

* Cloud-based MySQL database with REST API.
* Role-based permissions ensure security.

**2.3 CRUD Operations & Reports**

* Manage students, lecturers, courses, and attendance records.
* Generate reports for attendance, student lists, and enrollment.

**2.4 Photo Capture & Location Tracking**

* Capture and store student photos.
* Track device location for verification.

**3. System Design & Architecture**

**3.1 High-Level Design**

* **Frontend:** Web/mobile app for attendance tracking.
* **Backend:** Node.js REST API for data processing.
* **Database:** Cloud-based for storage.
* **Authentication:** Role-based access control.

**3.2 Key Components**

* **User Interface:** Attendance tracking and reports.
* **API Layer:** Handles frontend-backend communication.
* **Database Layer:** Stores users, attendance, and courses.
* **Use Case:** Lecturer registers a device, marks attendance, and generates reports.

**4. Development & Tech Stack**

**4.1 Development Process**

**Frontend Development**

* Initialize React Native project.
* Implement state management using Redux.
* Design UI components and navigation using React Navigation.

**Backend Development**

* Set up a Node.js and Express.js server.
* Configure MySQL database
* Develop REST API endpoints for attendance tracking.

**Testing & Quality Assurance**

* Perform unit testing with individual components.
* Conduct end-to-end testing
* Validate API endpoints and database interactions.

**Deployment & Maintenance**

* Release mobile app using Expo for cross-platform compatibility.
* Monitor system performance and apply security updates.

**5. Conclusion**

This document outlines the system’s objectives, requirements, and architecture. The automated attendance system aims to enhance efficiency and security in tracking student attendance while ensuring reliable data storage and access. By utilizing a modern tech stack and agile development practices, the system is designed for scalability and ease of use. Future steps include refining system design, implementing core functionalities, and conducting rigorous testing to ensure reliability before deployment.

**Project X - Automated Attendance System Documentation**

**Members**

Richard Fuertes

Richard Jefferson O. Daganio

Jomari E. Gamao