**High Level Design**

**Project Name:** Project X: Automated Attendance System

**Version:** 1.0

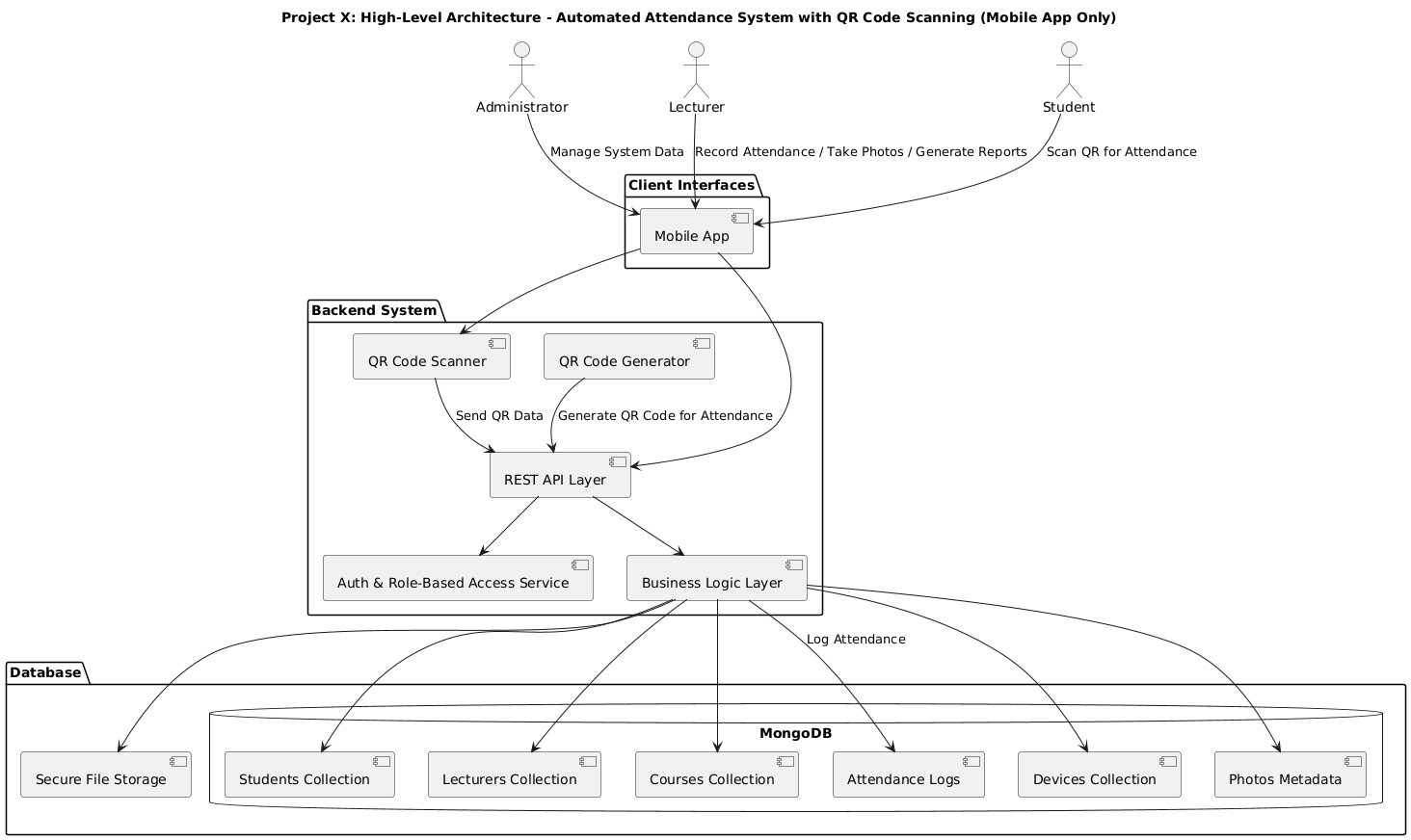
**Author:** Albert&Friends

**Date**: May 2025

1. **Introduction**

The Automated Attendance System, referred to as **Project X**, is a mobile-based application designed to provide **secure, real-time attendance tracking** for educational institutions. It integrates **QR code scanning**, **photo capturing**, and **cloud-based data storage** using **MongoDB**, while enforcing strict **role-based access control** for students, lecturers, and administrators.

1. **High Level Diagram**



### ****User Roles and Responsibilities****

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| Admin | Manage all data (students, lecturers, devices, courses). |
| Access all reports and perform system-level actions. |
| Lecturer | Generate QR codes and take student attendance. |
| Capture student photos and manage course-related data. |
| View attendance reports. |
| Student | Scan QR codes to mark attendance. |
| Enroll in available courses. |

### ****Client Interface: Mobile App****

* Used by **all user roles**.
* **Lecturers**: Generate QR codes, capture photos, and monitor attendance.
* **Students**: Scan QR codes to log attendance.
* **Admins**: Manage users, devices, and view reports.

### ****Backend System Components****

|  |  |
| --- | --- |
| **Component** | **Description** |
| Authentication & Role Control | Validates user login and authorizes access based on roles. |
| REST API Layer | Handles all communication between app and backend (e.g., attendance logging). |
| Business Logic Layer | Contains all core logic (e.g., QR validation, data processing, role enforcement) |
| QR Code Generator | Creates unique QR codes with course/session info |
| QR Code Scanner | Built-in scanner in mobile app to mark attendance |

### ****Database and Storage****

|  |  |
| --- | --- |
| Storage | Details |
| MongoDB | Stores data collections: students, lecturers, attendance logs, devices, etc. |
| Secure File Storage | Stores student photos (linked to student records via metadata) |

### ****System Workflow Summary****

* **Lecturer logs in** via the mobile app and generates a QR code for the attendance session.
* **Student scans** the QR code using the app.
* The scanned data is sent to the **REST API**, which:
  + Validates the QR code.
  + Verifies the student's enrollment.
  + Logs the attendance entry into **MongoDB**.
* **Lecturers/Admins** can view reports and manage data from the mobile app.
* **Security measures** ensure:
  + Only registered devices are used for attendance logging.
  + Only authorized users access specific data and features.

### ****Security Considerations****

* **Role-based access** strictly enforced.
* **Device registration** required for attendance logging.
* **Photo metadata** is securely linked and stored.
* **API authentication** required for every request.