High-Level Design (HLD) Document

Project Title: Project X – Automated Attendance System

Version: 1.0

Prepared By: Albert&Friends

Date: May 2025

# 1. Introduction

## 1.1 Purpose

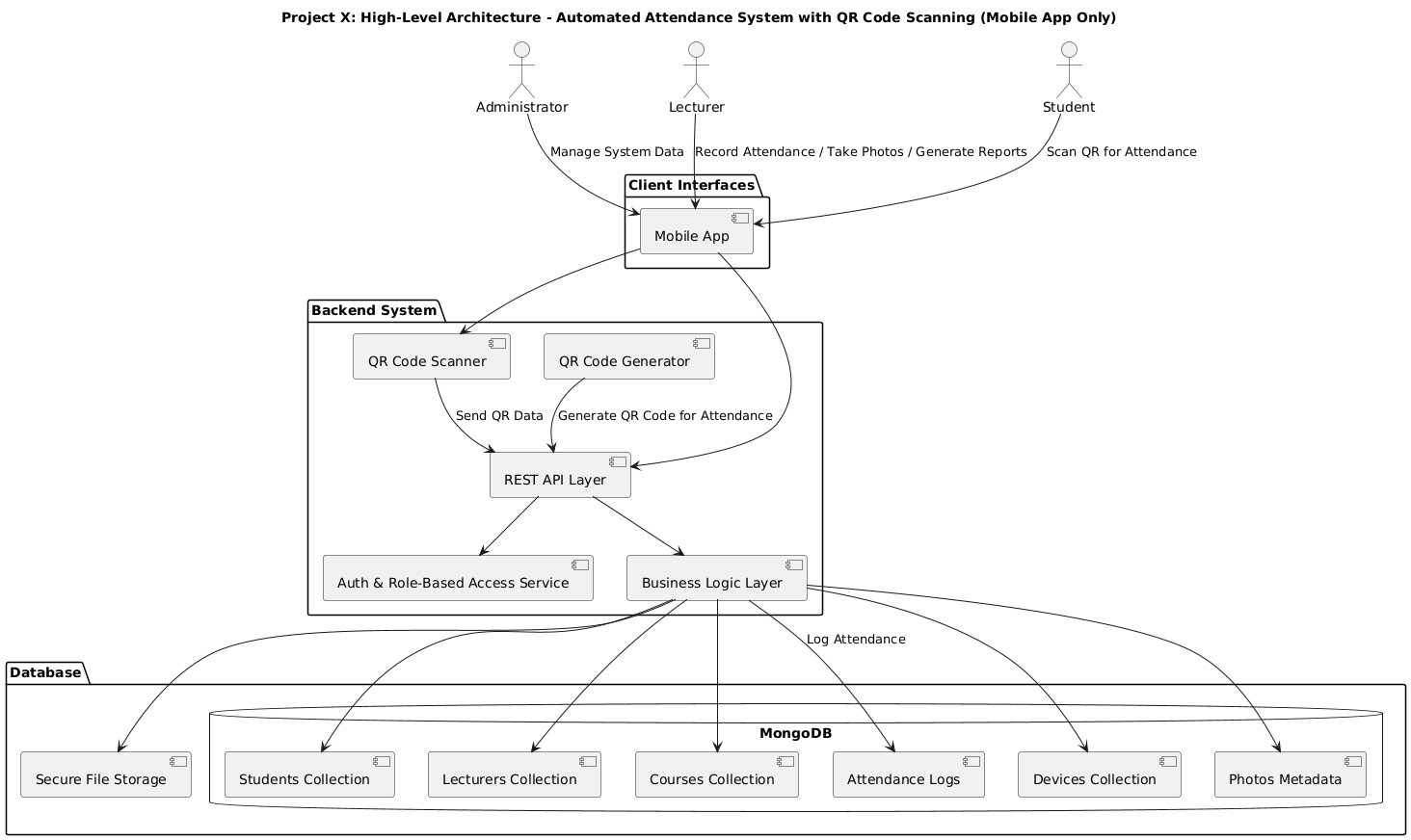
The purpose of this document is to provide a high-level design overview of the Automated Attendance System (Project X), outlining the system components, user roles, architecture, workflows, and interactions. This document acts as a blueprint for system developers and stakeholders, ensuring a shared understanding of the system’s structure, key components, workflows, and responsibilities. It sets the foundation for implementation, deployment, and testing.

## 1.2 Scope

This system enables secure, real-time attendance tracking for students using QR code scanning via a mobile application. It supports lecturers, administrators, and students in managing and accessing attendance-related operations. Project X focuses on educational institutions that require a digital, contactless solution for attendance tracking. It covers user authentication, attendance logging via QR codes, photo capture, real-time reporting, data management, and secure role-based access.

# System Architecture Overview

## 2.1 Architectural Diagram



## 2.2 Key Components

|  |  |
| --- | --- |
| **Component** | **Description** |
| Mobile App | Interface used by students, lecturers, and admins to interact with the system. |
| REST API | Backend interface handling all business logic and data operations. |
| Business Logic Layer | Processes attendance, QR validation, role enforcement, etc. |
| MongoDB (Database) | Stores collections for users, attendance records, devices, and courses. |
| Secure File Storage | Holds student profile pictures securely. |

# 3. User Roles and Permissions

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| Administrator | Manage system records and access full reports. |
| Lecturer | Generate QR codes, take attendance, capture photos, manage courses. |
| Student | Enroll in courses and scan QR codes to record attendance. |

# 4. Client Interface

## 4.1 Mobile Application

Used by all user types. Lecturers generate and students scan QR codes for attendance. Admins manage users and view reports. The mobile application is developed using cross-platform technologies and designed with an intuitive UI/UX to cater to different user roles. The app includes modules such as login/authentication, attendance dashboard, QR code generator and scanner, report viewer, and photo capture interface. It communicates with the backend via secure RESTful APIs.

# 5. System Workflow

## 5.1 Attendance Process

1. Lecturer logs in and generates a QR code.
2. Student scans QR using their device.
3. Data is sent to REST API, validated, and stored in MongoDB.
4. Lecturers/Admins view reports and manage data.
5. Security measures ensure only authorized devices and users can access the system.

## 5.2 Device Registration Workflow

1. Lecturer logs in to the app and navigates to the Device Management section.
2. Lecturer registers a new device by providing device name and location.
3. Device ID is generated and stored in MongoDB.
4. Only registered devices can be used to generate or scan QR codes.

## 5.3 Reporting Workflow

1. Administrator or lecturer selects the Report module from the app.
2. Filters are applied (e.g., by course, student, or date).
3. System queries MongoDB and generates a formatted report.
4. Report is displayed in-app and available for export (PDF/CSV).

# 6. Security and Access Control

The system enforces authentication, role-based access, device registration, and secure API communication to protect data integrity and user privacy. Only verified users and registered devices can access sensitive operations.

# 7. Data Model Overview

|  |  |
| --- | --- |
| **Collection/Table** | **Key Fields** |
| Students | StudentID, Name, UniversityID, Profile Picture |
| Lecturers | LecturerID, Name, Device List |
| Courses | CourseID, Name, LecturerID |
| Attendance | StudentID, CourseID, Date, Time, Status |
| Devices | DeviceID, LecturerID, Device Info |
| Photos | PhotoID, StudentID, File Reference |

# 8. Reporting Module

The system supports generating reports based on students, courses, or date ranges. Reports are accessible via the mobile app and are exportable in PDF or CSV format.

# 9. Assumptions and Constraints

Only registered devices can record attendance. Mobile app requires internet connectivity. QR codes are session-specific and expire after use. The application assumes all users have valid credentials and compatible devices.

# 10. Error Handling and Logging

The system will implement a centralized logging mechanism to track system events, errors, and unauthorized access attempts. Logs will be categorized (info, warning, error) and stored in a secure cloud environment. Error messages will be user-friendly and actionable. Critical issues will trigger alerts for system administrators.

# 11. Performance and Scalability Considerations

The system is designed to handle concurrent users and large datasets with minimal latency. MongoDB is selected for its scalability and performance under read/write loads. Indexing will be applied to frequently queried fields. API endpoints will be optimized with pagination, caching, and asynchronous processing where applicable.