

**Automated Attendance System**

**Faculty of Computing Data Sciences Engineering and Technology**

**Christian Dave B. Motlog**

**Jelaie G. Talara**

**Rikki Mae Alvar**

**Bachelor of Science and Information Technology**

**May 2025**

**Table of Content**

[1. Introduction 1](#_Toc199254811)

[1.1 Purpose 1](#_Toc199254812)

[1.2 Scope 1](#_Toc199254813)

[1.3 Definitions, Acronyms, and Abbreviations 1](#_Toc199254814)

[1.4 References 1](#_Toc199254815)

[2. System Overview 2](#_Toc199254816)

[2.1 System Description 2](#_Toc199254817)

[2.2 User Roles and Personas 2](#_Toc199254818)

[3. Architecture Design 2](#_Toc199254819)

[3.1 System Architecture Diagram 2](#_Toc199254820)

[3.2 Component Diagram 2](#_Toc199254821)

[4. Module Description 3](#_Toc199254822)

[4.1 QR Code Generator 3](#_Toc199254823)

[4.2 Attendance Module 3](#_Toc199254824)

[4.3 User Management 3](#_Toc199254825)

[4.4 Reporting Module 3](#_Toc199254826)

[4.5 Authentication Module 3](#_Toc199254827)

[5. Data Design 4](#_Toc199254828)

[5.1 Data Model Diagram 4](#_Toc199254829)

[5.2 Data Flow 4](#_Toc199254830)

[6. Technology Stack 4](#_Toc199254831)

[7. Security Considerations 4](#_Toc199254832)

[8. Scalability & Performance 5](#_Toc199254833)

[9. Integration 5](#_Toc199254834)

[10. Assumptions & Constraints 5](#_Toc199254835)

[11. Appendices 6](#_Toc199254836)

# Introduction

## Purpose

The **Automated Attendance System** is designed to streamline and digitize the process of tracking student attendance in educational institutions. By utilizing QR code technology, it eliminates manual logging, minimizes errors, and provides real-time attendance data to instructors and administrators.

## Scope

The system includes the following core functionalities:

* **User Role Management:** Supports Admin, Instructor, and Student roles.
* **Unique QR Code Assignment:** Each student is issued a unique QR code linked to their identity.
* **Attendance Scanning:** Instructors scan student QR codes to instantly record attendance.
* **Admin Dashboard:** Admin users manage accounts, generate reports, and configure system settings.
* **Cross-Platform Access:** The system is accessible via web and/or mobile platforms.

## Definitions, Acronyms, and Abbreviations

* **QR Code:** Quick Response Code, a two-dimensional barcode storing student info.
* **Admin:** User who manages the system.
* **Instructor:** User who records attendance by scanning QR codes.
* **Student:** User who possesses a unique QR code for attendance.

## References

<https://www.mongodb.com/docs/manual/>

[https://www.researchgate.net/publication/342343678\_Smart\_Attendance\_System\_using](https://www.researchgate.net/publication/342343678_Smart_Attendance_System_using_QR_Code)

[\_QR\_Code](https://www.researchgate.net/publication/342343678_Smart_Attendance_System_using_QR_Code)

<https://nodejs.org/docs/latest/api/>

# System Overview

## System Description

The system consists of three main user roles interacting with a centralized backend database. Students are registered and assigned unique QR codes, which serve as their digital identity for attendance. Instructors use scanning devices (mobile app or web camera) to scan QR codes during class, which instantly records attendance. Admin monitors attendance data and manages users.

## User Roles and Personas

**Role Description Key Actions**

Admin Manages users and system settings Create/update users, view reports

Instructor Takes attendance by scanning QR codes Scan QR code, view attendance history Student Receives unique QR code to be scanned View own attendance records

# Architecture Design

## System Architecture Diagram

*(Here, insert a diagram showing: Student devices → QR code generation → Instructor scanning device → Backend server → Admin dashboard)*

## Component Diagram

**Component Description**

QR Code Generator Generates unique QR codes for each student

Attendance Module Processes QR code scans, validates and logs attendance

**Component Description**

User Management Handles creation and management of Admin, Instructor, Student

Reporting Module Generates attendance reports accessible by Admin Authentication Manages login and access control for all users

# Module Description

## QR Code Generator

* **Description:** Generates unique QR codes linked to student IDs.
* **Responsibilities:** Create, store, and distribute QR codes securely.
* **Interfaces:** Student profile creation UI, QR code download/print options.

## Attendance Module

* **Description:** Scans and validates QR codes, marks attendance.
* **Responsibilities:** Real-time attendance recording, duplicate scan prevention.
* **Interfaces:** QR scanner interface for instructors, backend API for attendance logging.

## User Management

* **Description:** Admin module for user account lifecycle.
* **Responsibilities:** CRUD operations for Admin, Instructor, and Student accounts.

## Reporting Module

* **Description:** Provides attendance statistics and reports.
* **Responsibilities:** Generate daily, weekly, monthly reports; export to CSV/PDF.

## Authentication Module

* **Description:** Ensures secure login and access control.
* **Responsibilities:** Role-based authentication, session management.

# Data Design

## Data Model Diagram

Entities:

* **User (Admin, Instructor, Student)**
* **StudentQR** (StudentID, QRCodeData)
* **AttendanceRecord** (AttendanceID, StudentID, InstructorID, Timestamp, Status)

## Data Flow

* Student data → QR Code generated → Student receives QR code
* Instructor scans QR code → System verifies and logs attendance → Admin views reports

# Technology Stack

|  |  |
| --- | --- |
| **Layer** | **Technology/Tools** |
| Frontend | React (Web) or React Native (Mobile) |
| Backend | Node.js |
| Database | MongoDB |
| QR Code | QR code generation libraries (e.g., qrcode.js) |

Authentication JWT-based authentication

# Security Considerations

* Secure storage of QR codes to prevent forgery.
* Authentication and role-based access control.
* SSL/TLS encryption for all communications.
* Prevent replay attacks on QR code scans by timestamp validation.

# Scalability & Performance

* Use efficient indexing on student IDs for quick attendance logging.
* Support multiple concurrent instructors scanning simultaneously.
* Optionally, deploy backend with load balancing for high availability.

# Integration

* Integration with existing student information systems (SIS) for syncing student data (optional).
* API endpoints for mobile apps and web clients.

# Assumptions & Constraints

* Students have access to their unique QR codes (printed or digital).
* Instructors have devices capable of scanning QR codes.
* Network connectivity is available during attendance scanning.

# Appendices

* Sample QR code format specification
* User interface wireframes (optional)