Republic of the Philippines
DAVAO ORIENTAL STATE UNIVERSITY
Guang-guang, Dahican, City of Mati, Davao Oriental
Faculty of Computing, Data Sciences, Engineering and Technology
Information Technology Program

ITC 130 – Applications Development in Emerging Technologies

PROJECT X: Automated Attendance Tracking System: Low Level Design

PRESENTED BY:

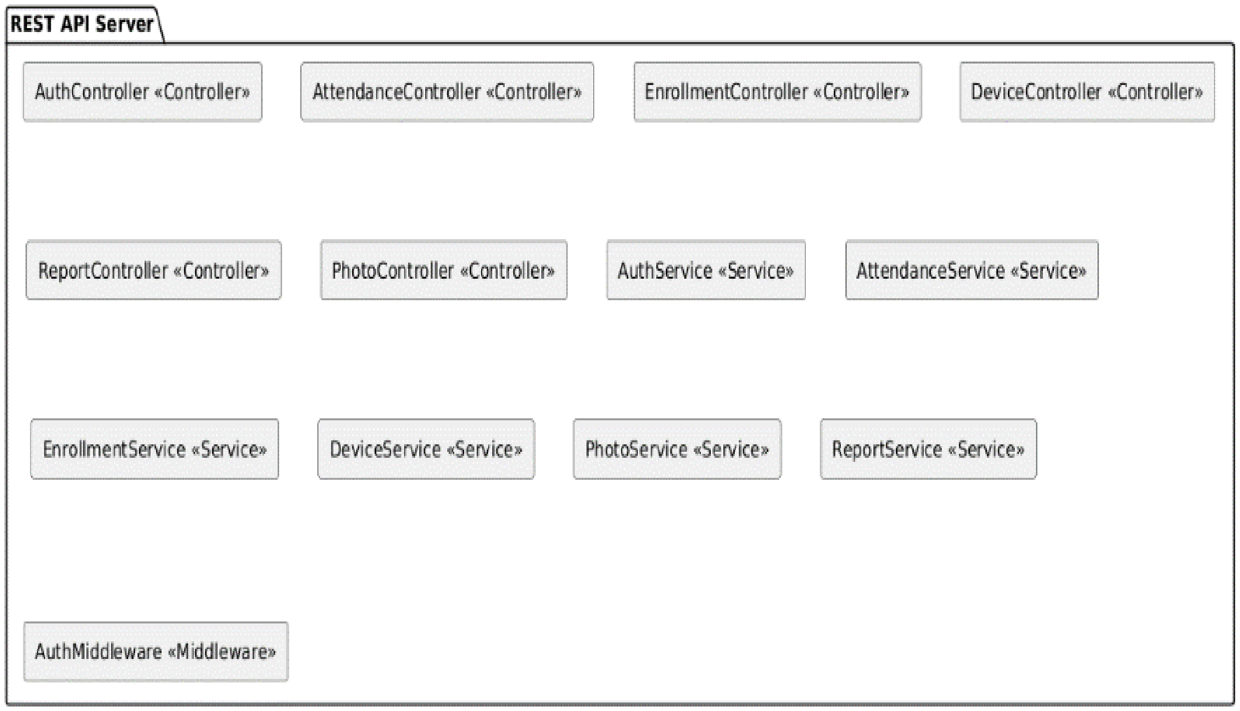
NEIL ROY G. OMONGOS

WENONA MARIE M. MONTEMAYOR

SHAMAIAH LEE CADUT

System Architecture Overview

This document provides a detailed breakdown of the architectural components of the REST API server, including controllers, services, repositories, database integration, and external dependencies.


REST API Server
AuthController «Controller»
AttendanceController «Controller»
EnrollmentController «Controller»
DeviceController «Controller»
ReportController «Controller»
PhotoController «Controller»
AuthService «Service»
AttendanceService «Service»
EnrollmentService «Service»
DeviceService «Service»
PhotoService «Service»
ReportService «Service»
AuthMiddleware «Middleware»

Overview

The **component structure** defines the core functionalities of the REST API, ensuring modular interaction between controllers, services, and middleware for efficient request processing.

Controllers (Request Handlers)

These act as entry points for handling HTTP requests and delegating operations to the service layer:

- AuthController – Manages user authentication and authorization.
- AttendanceController – Handles attendance tracking operations.
- EnrollmentController – Oversees user enrollment processes.

- DeviceController – Manages registered device interactions.
- ReportController – Generates structured reports for users.
- PhotoController – Processes image-related functionalities.

Services (Business Logic Layer)

Responsible for executing system processes, interacting with repositories, and implementing security:

- AuthService – Handles token authentication and credential validation.
- AttendanceService – Processes attendance records.
- EnrollmentService – Manages enrollment and associated dependencies.
- DeviceService – Ensures device configuration and authentication.
- ReportService – Aggregates and formats reports for stakeholders.
- PhotoService – Manages image storage and retrieval.

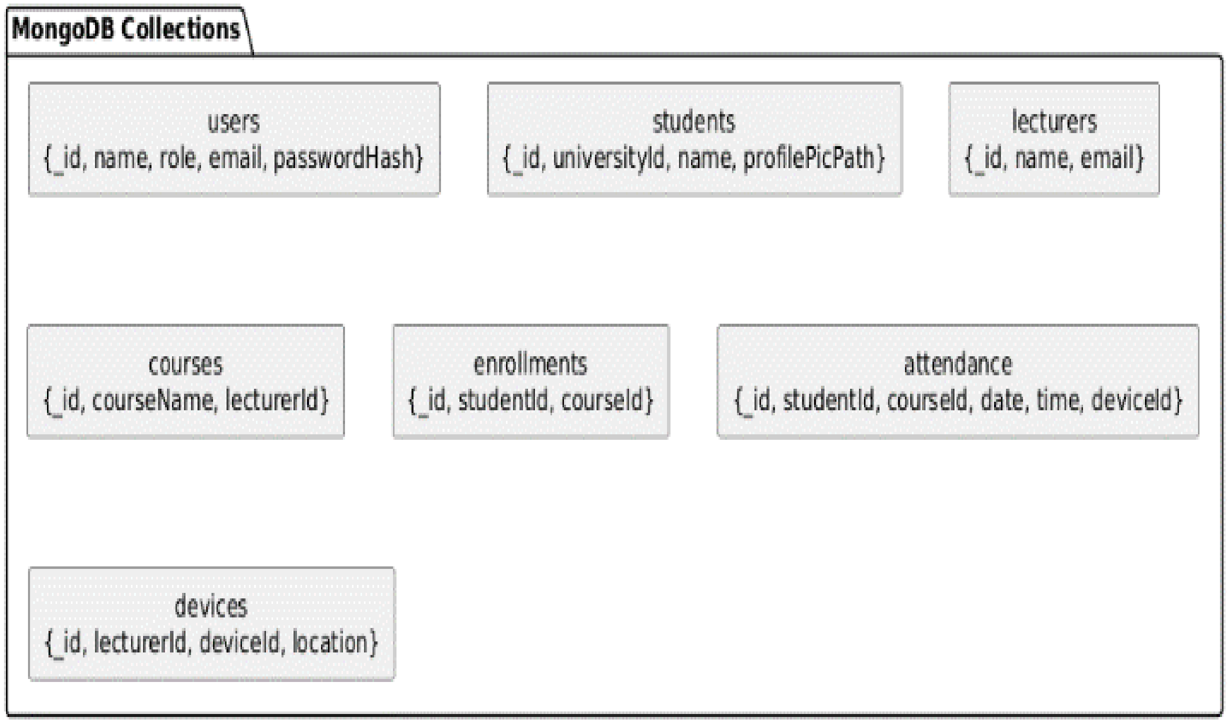
Middleware (Intermediary Processes)

Ensures security measures and request handling:

- AuthMiddleware – Verifies authentication tokens before granting access

Key Considerations

- **Modular Development:** Separation of concerns allows independent scaling of components.
- **Security & Authentication:** Implemented through middleware layers ensuring controlled access.
- **Error Handling:** Structured logging and validation mechanisms ensure system robustness.


MongoDB Collections
users
{_id, name, role, email, passwordHash}
students
{_id, universityId, name, profilePicPath}
lecturers
{_id, name, email}
courses
{_id, courseName, lecturerId}
enrollments
{_id, studentId, courseId}
attendance
{_id, studentId, courseId, date, time, deviceId}
devices
{_id, lecturerId, deviceId, location}

Overview

The **persistence layer** handles structured data storage, ensuring efficient retrieval and transaction integrity.

Repositories (Data Access Layer)

Abstracts direct interactions with the database and provides structured query execution:

- UserRepository – Manages authentication and role-based access control (RBAC).
- AttendanceRepository – Handles attendance record storage and retrieval.
- EnrollmentRepository – Stores enrollment details linked to user profiles.
- DeviceRepository – Maintains registered device data.
- ReportRepository – Facilitates structured reporting queries.
- PhotoRepository – Stores and retrieves image data.

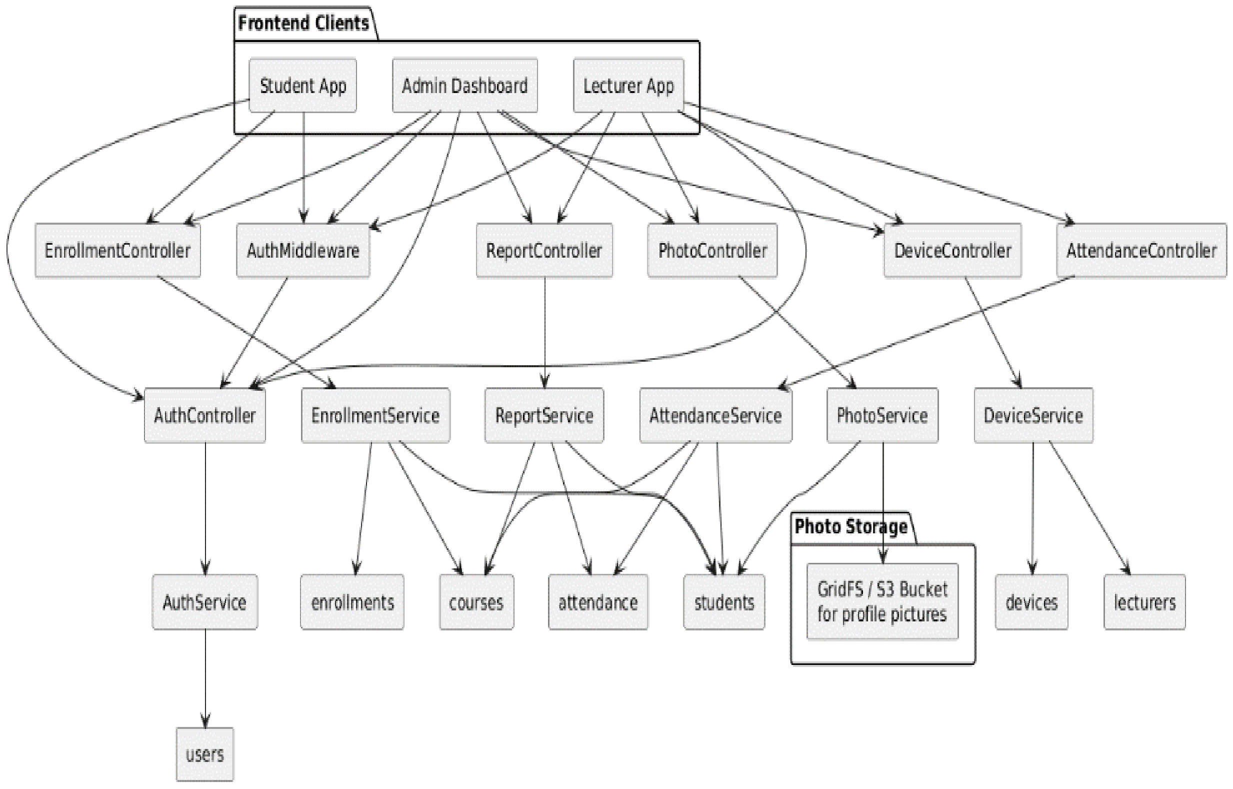
Database Schema (Entity Structure)

Defines the relational tables used for system persistence:

- Users Table – Stores user credentials and roles.
- Attendance Table – Maintains timestamped attendance entries.
- Enrollment Table – Contains user enrollment details.
- Devices Table – Manages device registration and interaction data.
- Reports Table – Stores structured reports.
- Photos Table – Handles image storage metadata.

Key Considerations

- **ACID Compliance:** Ensures transaction integrity and consistency.
- **Indexing & Query Optimization:** Implements indexing for enhanced performance.
- **Security Measures:** Includes encryption of sensitive fields



Overview

This section details the integrations between the REST API and third-party services, supporting authentication, storage, real-time messaging, and logging.

Third-Party Services

- Google OAuth – Provides secure authentication through Google accounts.
- AWS S3 – Enables cloud-based storage for images and documents.
- SMTP Server – Supports email notifications.

Client Applications

- Web Client (React.js) – Frontend interface for managing system functionalities.
- Mobile Client (React Native) – Mobile application for real-time tracking.

Messaging & Logging

- WebSocket – Implements real-time communication.
- Logger Service – Captures system events and debug logs.

Key Considerations

- **Authentication & Security:** OAuth improves security while maintaining accessibility.
- **Scalability:** AWS S3 supports efficient document storage.
- **Reliability:** Logging mechanisms ensure transparency and debugging efficiency.

Conclusion

This low-level design document provides a structured breakdown of system components, ensuring clarity in implementation and scalability in future enhancements.