REQUIREMENTS DOCUMENT FOR PROJECT X ATTENDANCE SYSTEM

TEAM BURDANGBALL
JOOMER JAKE CABALLERO
CHRISTOPHER BURGOS
LEOPOLDO DUMADANGON

1. Introduction

The Project x Attendance System is an automated attendance tracking solution designed for university lecturers. It supports multiple registered devices, stores attendance records in a cloud-hosted MySQL database, and provides interfaces for students, lecturers, and administrators. The system uses REST API for data access and modification.

2. System Overview

2.1 Functional Requirements

2.1.1 Automated Attendance Tracking: The system should automatically track attendance for university lecturers.

2.1.2 Multi-Device Support: The system should support multiple registered devices (phones, tablets, computers).

2.1.3 Cloud-Hosted Database: Attendance records should be stored in a cloud-hosted MySQL database.

2.1.4 User Interfaces: The system should provide interfaces for students, lecturers, and administrators.

2.1.5 REST API: The system should use REST API for data access and modification.

2.2 Non-Functional Requirements

2.2.1 Scalability: The system should be scalable to accommodate a growing number of users and devices.

2.2.2 Performance: The system should perform efficiently with real-time data sync and quick response times.

3. User Roles and Permissions

3.1 Lecturers

3.1.1 Device Registration: Lecturers should be able to register one or more devices.

3.1.2 Attendance Recording: Lecturers should be able to record attendance via QR code, GPS, or biometric authentication.

3.1.3 Student Management: Lecturers should be able to manage students enrolled in their courses.

3.1.4 Report Generation: Lecturers should be able to generate attendance reports.

3.2 Administrators

3.2.1 System Management: Administrators should be able to manage lecturers, students, courses, and attendance records.

3.2.2 Device Management: Administrators should be able to manage device access and tracking.

3.3 Students

3.3.1 Attendance Viewing: Students should be able to view their own attendance records.

3.4 Access Control

3.4.1 Admin Privileges: Only administrators should be able to modify system-wide data.

3.4.2 Lecturer Privileges: Lecturers should be able to manage their own courses and students.

4. Attendance Tracking

4.1 Attendance Recording

4.1.1 Methods: Attendance can be recorded via QR code, GPS, or biometric authentication.

4.1.2 Validation: The system should validate physical presence before marking students present.

4.1.3 Real-Time Sync: Attendance data should be synced in real-time.

5. Device Registration & Tracking

5.1 Device Registration

5.1.1 Multiple Devices: Lecturers should be able to register one or more devices.

5.1.2 Access Control: Only registered devices should be used for attendance tracking.

5.2 Device Management

5.2.1 Admin Control: Administrators should be able to manage device access and tracking.

6. Student Management

6.1 Student Profiles

6.1.1 Profile Management: Administrators should be able to add, update, delete, and view student profiles.

6.1.2 Profile Information: Student profiles should include name, university ID, and passport-style photo.

6.1.3 Photo Storage: Photos should be stored as files, linked to student IDs in the database.

6.2 Course Enrollment

6.2.1 Enrollment: Lecturers should be able to enroll students in their courses.

7. Course Management

7.1 Course Creation

7.1.1 Course Management: Administrators should be able to create, update, and delete courses.

7.1.2 Course Information: Each course should have a unique course ID, name, and assigned lecturer(s).

7.2 Student Assignment

7.2.1 Assignment: Lecturers should be able to assign students to courses.

8. Reporting & Data Management

8.1 Report Generation

8.1.1 Report Types: The system should generate reports on attendance records, student lists, courses, and registered devices.

8.1.2 Export Formats: Reports should be exportable as CSV or PDF.

9. Security & Access Control

9.1 Authentication

9.1.1 Secure Login: The system should use OAuth 2.0 / JWT Authentication for secure login.

9.1.2 Encryption: Database connections should be encrypted for security.

9.2 Audit Logs

9.2.1 Tracking: Audit logs should track modifications to attendance records.

10. System Architecture

10.1Microservices Architecture

10.1.1 REST API: The system should use a microservices architecture with REST API.

10.1.2 Frontend: The frontend should be developed using React Native.

10.1.3 Backend: The backend should be developed using Spring Boot or Node.js with Express.

11. Testing & Development Process

11.1 Testing Levels

11.1.1 Unit Testing: Unit testing should be conducted for individual components.

11.1.2 System Testing: System testing should be conducted for database and API integration.

11.1.3 User Acceptance Testing (UAT): UAT should be conducted to ensure the system meets user requirements.

11.2 Agile Development

11.2.1 Sprints: Development should follow a 2-week sprint cycle.

11.2.2 Daily Standups: Daily standups should be conducted to track progress.

11.2.3 Client Demonstrations: Sprint client demonstrations should be held at the end of each sprint.

12. Conclusion

The Project x Attendance System aims to provide a robust, secure, and scalable solution for automated attendance tracking in a university setting. By adhering to the outlined requirements, the system will ensure efficient attendance management, user-friendly interfaces, and secure data handling. The system leverages modern technologies, such as cloud-hosted MySQL databases and REST APIs, to provide real-time data synchronization and seamless integration with other university systems. Its support for multiple devices, including smartphones, tablets, and computers, ensures flexibility and convenience for lecturers.

Security is a top priority, with OAuth 2.0 and JWT authentication for secure logins, encrypted database connections, and audit logs to track changes. Advanced attendance tracking methods, such as QR codes, GPS, and biometric authentication, ensure accuracy and prevent fraud. The system's role-based interfaces cater to students, lecturers, and administrators, offering tailored functionalities like attendance viewing, student management, and report generation.

Built on a microservices architecture with React Native for the frontend and Spring Boot or Node.js for the backend, the system is scalable and reliable. Agile development practices, including 2-week sprints and Test-Driven Development (TDD), ensure a high-quality, user-focused solution. With these features, the Project x Attendance System is designed to meet the needs of modern educational institutions effectively.