**Kissy Faith M. Candia BSIT -3C**

**Bjorn Dimsey Gofredo TEAM C**

**Francis Lloyd Catog**

**Project X – Attendance System Requirements Document**

**1. Project Overview**

**Objective:** Develop a cloud-based automated attendance system for university lecturers and students.

**2. System Requirements**

1. **Functional Requirements:**
   1. **User Authentication:**
      1. Lecturers and students must log in via a registered device.
      2. Secure authentication using JWT tokens.
   2. **Attendance Marking:**
      1. Students check in using QR codes or NFC technology.
      2. System records the attendance status in real time.
   3. **Data Management:**
      1. Lecturers can add, edit, delete, and view student records.
      2. System stores student photos along with their university ID.
   4. **Reports & Analytics:**
      1. Generate attendance reports in PDF, CSV, or Excel.
      2. View course-wise and student-wise attendance logs.
   5. **Security Features:**
      1. Biometric authentication for lecturers.
      2. Data encryption and secure API communication.
   6. **Non-Functional Requirements:**
      1. **Scalability:** Supports multiple lecturers and courses.
      2. **Reliability:** Data backup and cloud storage to prevent data loss.
      3. **Performance:** Optimized database queries for fast response times.
2. **Sprint Plan & Deliverables**
3. **Sprint 1: User Authentication & System Setup**
4. **Deliverables:** ✅ Cloud-based MySQL database setup

✅ REST API for authentication

✅ Lecturer & student login/logout functionality

✅ Role-based access control

* + - * 1. ✅ Sprint Demonstration: Lecturer logs in using a registered device

1. **Sprint 2: Attendance Tracking Implementation**
2. **Deliverables:**
   * + - 1. ✅ QR code/NFC generation for students
         2. ✅ Student identity verification
         3. ✅ Real-time attendance status updates
         4. ✅ Sprint Demonstration: Lecturer marks student attendance via mobile device
3. **Sprint 3: Reports & Data Management**
4. **Deliverables:**
   * + - 1. ✅ CRUD operations for students, lecturers, and courses
         2. ✅ Attendance reports (PDF, CSV, Excel)
         3. ✅ Dashboard for analytics
         4. ✅ Sprint Demonstration: Lecturer generates an attendance report
5. **Sprint 4: Security Enhancements & Optimization**

**Deliverables:**

* + - * 1. ✅ Biometric authentication (fingerprint/face recognition)
        2. ✅ Data encryption

(attendance records, student photos)

* + - * 1. ✅ Secure API communication (TLS & certificate pinning)
        2. ✅ Sprint Demonstration: Lecturer logs in using biometric authentication

**Sprint 5: Final Integration & User Acceptance Testing**

**Deliverables:**

Unit

System

and user acceptance tests (UAT)

Bug fixes and UI/UX refinements

Cloud deployment

Final project documentation and presentation

Sprint Demonstration: Full system walkthrough with real-world test cases

**4. Development & Agile Methodology**

**Project Management Approach:**

* **Methodology:** Agile (Scrum-based)
* **Daily Standups:** Progress updates, blockers, and next steps
* **Tools:**
  + GitHub (Version Control)
  + Trello (Task Management)
  + Firebase/AWS (Cloud Services)

**Team Roles:**

* **Team Lead** (Rotates every sprint)
* **Backend Developer** (API & Database)
* **Frontend Developer** (Mobile UI/UX)
* **Database Administrator** (Data storage & security)

**5. Final Deliverables**

* Fully functional cloud-based attendance system
* Secure API with authentication and encryption
* Comprehensive attendance tracking with reporting features
* Complete project documentation and final presentation

**6. System Design & UML Diagrams**

**Entity-Relationship Diagram (ERD):**

The following ERD illustrates the relationships between key entities in the system:

**Entities:**

* **Students:** ID, Name, Email, Course ID
* **Lecturers:** ID, Name, Email
* **Courses:** ID, Name, Lecturer ID
* **Attendance Records:** ID, Student ID, Course ID, Date, Status
* **Reports:** ID, GeneratedAt, Content
* **Classes:** ID, Class Name
* **Use Cases:** ID, Description
* **Sequences:** ID, Sequence Name
* **Activities:** ID, Activity Name
* **Components:** ID, Name

The database is structured to optimize query efficiency and ensure referential integrity.

**Use Case Diagram:**

* Lecturer logs in and manages attendance.
* Student checks in via QR code/NFC.
* System records attendance and generates reports.

**Sequence Diagram:**

* Student scans QR code → System verifies → Attendance recorded.
* Lecturer logs in → Views attendance records → Generates report.

**Activity Diagram:**

* Login process
* Attendance verification
* Report generation

These UML diagrams provide a clear visual representation of the system’s workflow, ensuring a structured approach to development.

This structured sprint plan ensures a clear roadmap for development while integrating best practices for security and scalability.

