FACULTY
OF
INFORMATION
AND
COMMUNICATION TECHNOLOGY

NATIONAL DIPLOMA: COMPUTER SYSTEMS ENGINEERING

PROJECT DESIGN III: PJD 301B

DESIGN DOCUMENTATION

PROJECT TITLE:.....Student Attendance Management System……

LECTURER NAME……Muwanguzi Mark Ntume and Chunling Du……

STUDENTS NAME:……MALOGADIHLARE TEBOGO MARVIN.....

STUDENTS NUMBER……218161693……

EMAIL ADDRESS……218161693@tut4life.ac.za……

CONTACT NUMBERS ……0633675240……

SUBMISSION DATE ……06 SEPTEMBER 2024……

TABLE OF CONTENTS

Introduction

SAMS is an innovative solution developed for automating recording student attendance in educational institutions. Traditional attendance methods eat into precious time, are prone to errors, and cannot accomodate many people at once. SAMS incorporates RFID and facial recognition into a web-based system that is efficient, secure, and has a lot of features.

The system lets students mark their attendance either by scanning RFID tags or through facial recognition, making the attendance records quite tamper-proof. Such technologies make the whole process smooth and add a layer of security and reduces the chances for false attendances.

This SAMS design document covers all aspects of the system, including system architecture, detailed design concerns, diagrams and methodologies applied during its development. Furthermore, this document identifies goals, assumptions, and dependencies critical to success in implementing such a system.

Systems overview

SAMS will be an integrated system, including hardware parts: RFID readers, ESP32 microcontroller, and a camera for face recognition, with strong software support via Spring Boot and MySQL. The system will work in a web environment, accessible with standard web browsers, hosted on a local server.

The major user groups of SAMS are students, instructors, and administrators. Different users perform distinct functions and have different levels of access within the system:
- Student: can log in to view attendance records and mark attendance through RFID tags or face recognition.
Instructor: Maintain attendance lists, view statistics of attendance and generate reports.
Admins: Have full access in the system, like managing users, creating courses or classes, and assigning instructors to classes.

The system architecture is modular, which will help with easy maintenance and scalability of the system. Subsystems involved are as follows:
1. User Management: Controls user authentication, assignment of roles, and management of profile.
2. Attendance Management: It deals with recording, storing, and retrieving the attendance information by using RFID and face identification.
3. Reporting: Its tools provide generation and export of attendance reports.

Design considerations

Different key factors have been considered in designing SAMS so that the system achieves the following objectives:
Scalability: The system is designed to allow an increase in users and data records without performance degradation. This design leverages the principles of modularity, while scalable technologies such as MySQL are adopted for this purpose.
Security: Attendance data is sensitive information. Since all users have access to the system, encryption of data transmission will be used along with secure authentication of the users and restriction of access by roles to prevent unauthorized access.
Integration: SAMS integrates hardware components like RFID readers and cameras with software modules. These shall be integrated in such a way that data processing will take place at constant realistic times, latency will be minimized.
- User Experience: This system is planned to be very usable. Its interface is simple and straight-forward to use, allowing great navigation paths and responsive design for different devices.
Reliability: SAMS should work under any conditions without break and with as little downtime as possible. The system provides data redundancy and error handling for the continuity of work.

Assumptions and dependencies

A number of assumptions underpin the design of SAMS:
- User Competence: It is assumed users have basic knowledge of web browsers and how RFID tags or facial recognition systems work.
Hardware Availability: The project design assumes that hardware componenents will be available and that they work, such as the RFID readers, ESP32 microcontrollers, and cameras.
- Internet Access: While SAMS has been designed to run on the local server, it assumes stable internet access for its remote access and possible IDE/pom.xml file updates, etc.
Operating System Environment: The system should be installed on a server with Java, MySQL, and Tomcat, and should be serviced periodically for proper functioning.

Dependencies include:
- Spring Boot: It provides the framework to build the backend services.
- MySQL: Used to handle the database of recorded attendance and user information.
- OpenCV: Facial recognition library used within the system to extend security.
- ESP32: It deals with reading RFID tags and sends the communication to the backend to register the attendance.

Goals and guidelines

The main goal of SAMS is to provide a reliable, scalable, and secure system for managing the attendance of students. The system shall:
- Automate Attendance: No more manual attendance marking, let RFID and facial recognition technologies do their jobs.
- Better Accuracy: Ensure that the attendance records are accurate, tamper-proof with minimal false attendance issues.
- Accessibility Improvement: The web-based platform shall be made accessible to students, instructors, and administrators from any place with internet access.
- Attendance Reporting: Provide instructors and the administration the ability to generate detailed attendance reports for further analysis and record-keeping purposes.

Some of the prescribed guidelines followed during development include:
Modular Design: The system is broken down into independent modules that may be individually designed, tested, and maintained.
Adherence to standards means that coding standards and best practices are followed to ensure code readability, maintainability, and reusability.
User-Centric Design: Easy-to-use and responsive design in the user interface that would keep friendly user experiences on any device.

Development Methods

SAMS follows the Agile methodology-a software development approach pursued in an iterative way, continuous feedback, and responsiveness to requirements. Some of the main activities included in the process of development are:
- Requirement Gathering: Basic requirements are gathered in a group of students, instructors, and administrators.
- Design and Prototyping: This is the building of a prototype to validate the design of a system through showing it to potential users so as to gain feedback.
- Implementation: The system itself is developed iteratively, with each iteration building on a certain module or feature.
- Testing: Comprehensive test for each module, from postman testing to integration testing to manual testing.
Deployment: The system is deployed to a local server and made accessible to users. Ongoing support and maintenance post-deployment are provided.

Architectural Strategies

SAMS is designed based on a client-server architecture. The client interacts with the server using RESTful APIs via a web browser normally. Some of the important architectural strategies used in the design of SAMS are as follows.

- 3-tier Architecture: The system will be threetiered based on Presentation, Business Logic, and Data Access. Which are the user interface, logic/services and database.
- RESTful Services:RESTful APIs are exposed for all the back-end services that can be easily integrated into the front-end and other third-party systems.
- Security Layer: The security layer is one that provides an implementation for the control of authentication, authorization, and data encryption.
- Scalability: Horizontal scaling in this system incorporates an additional server with an increase in the user base.

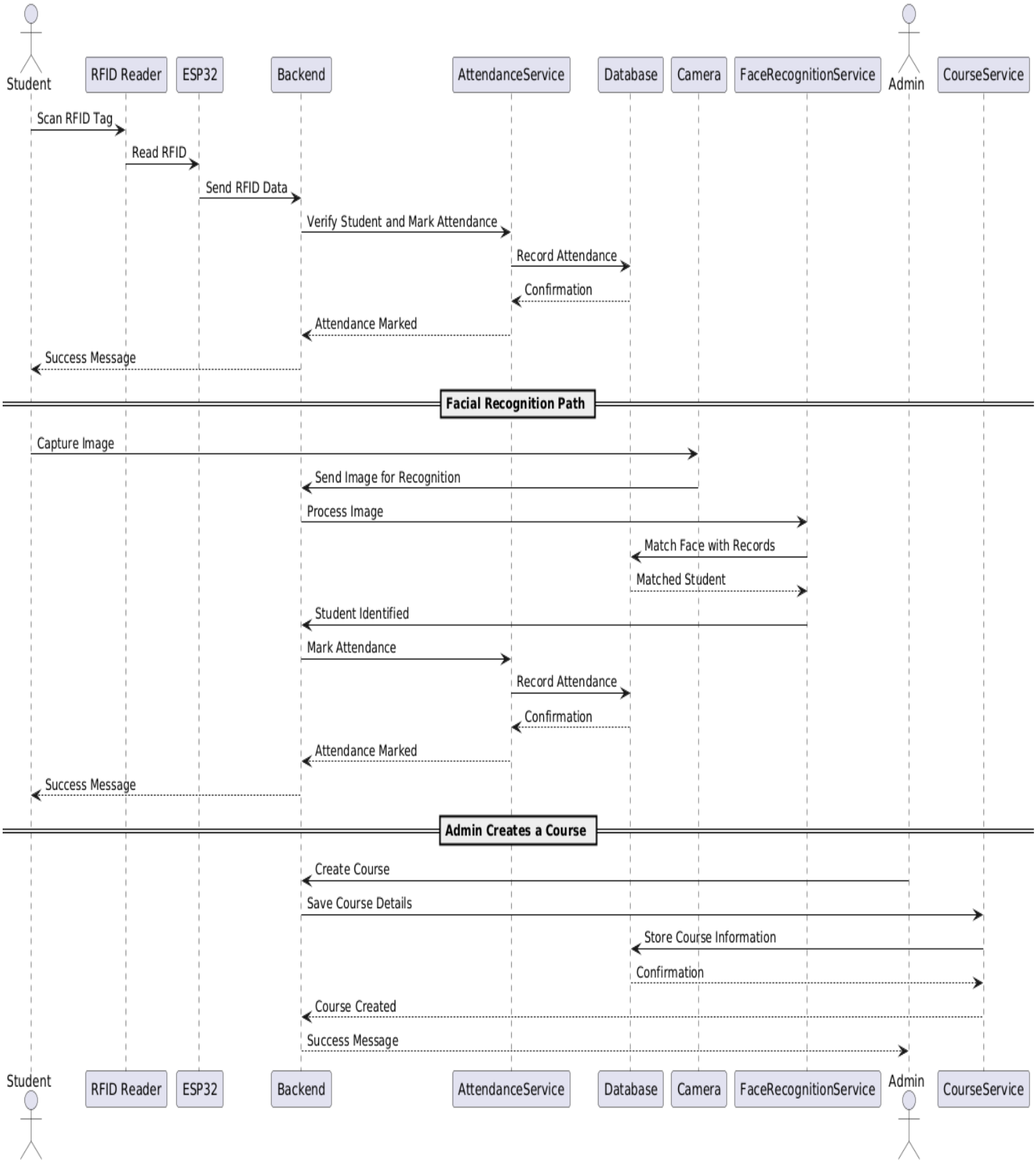
System Architecture

The major components of SAMS's system architecture include the following:

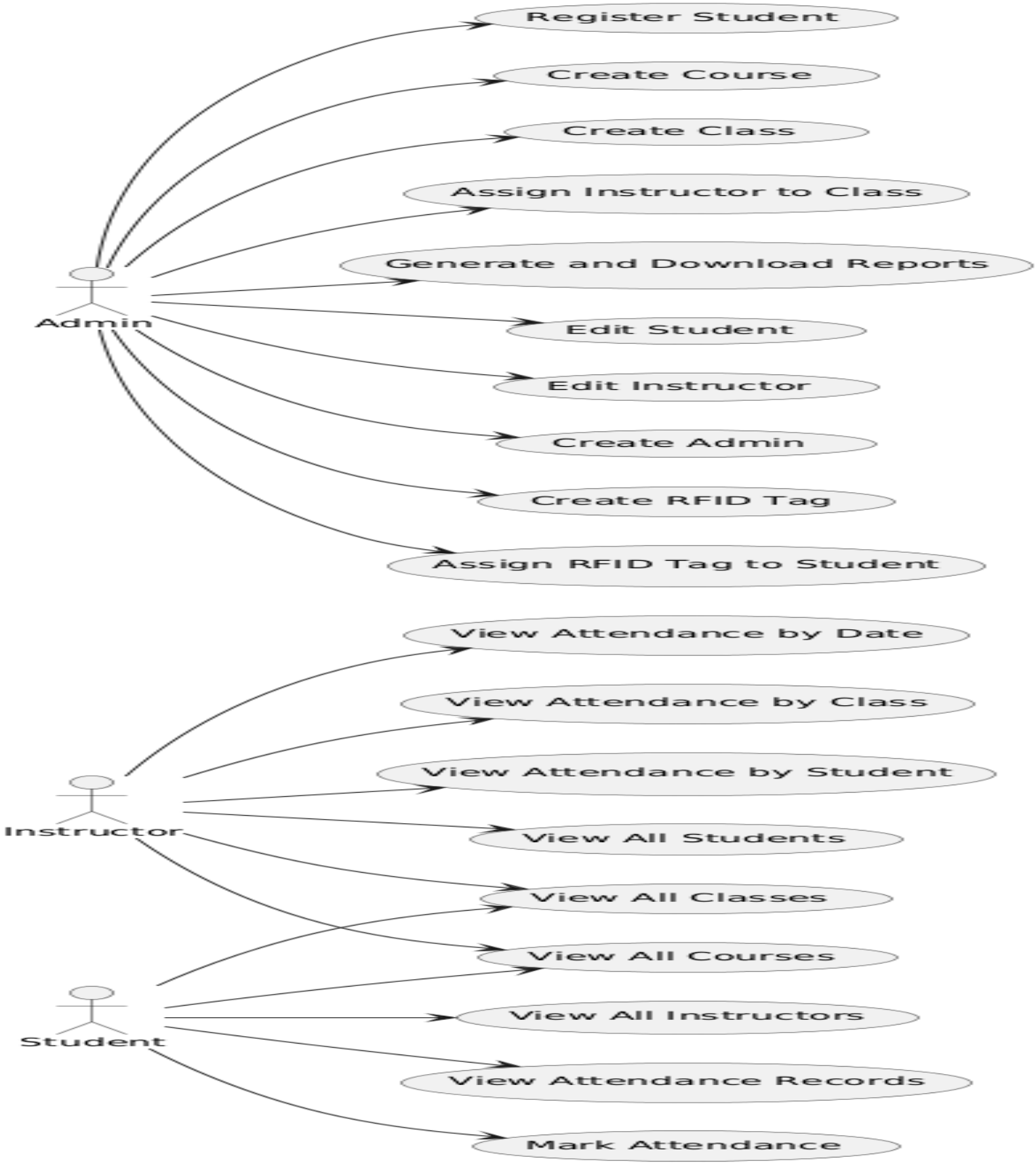
1. Frontend (Client): Front consists of an HTML, CSS, and JavaScript-built user interface for responsiveness and interactivity on the user's side. The front-end interacts with the back-end through AJAX calls over the RESTful APIs.
2. Backend - Server: Spring Boot develops or creates the server-side application, which maintains the business logic, processes requests, and communicates with the database. It also integrates OpenCV with the server to carry out face detection and process RFID information coming from the ESP32.
3. Database: The MySQL database persists all the information related to the users, records of attendance, course and classes, and system-wide settings. The schema of the database is designed by having data integrity in mind and also sustains complex queries which are needed by reporting.

Detailed System Design

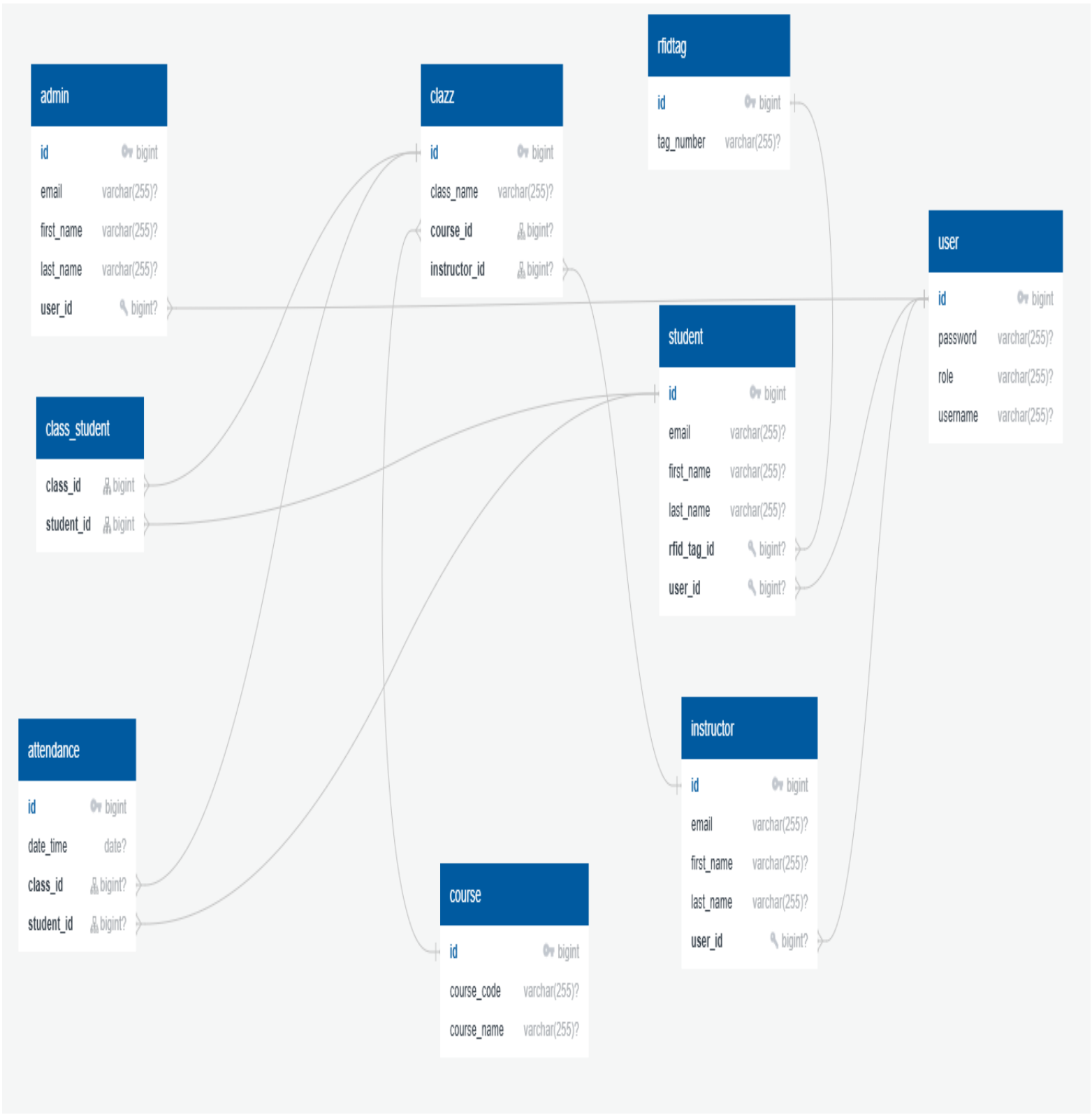
Sequence Diagram



Use Case Diagram



Class Diagram



The following section elaborates on every component of the system in detail:
-User Profile Management Module: This is for user registrations, logins, role assignments, and other profile-related managements. It contains UserController and UserService, both of which interact with -UserRepository to manage the data of users.
-Attendance Management Module: This module handles the recording of attendance by RFID and face recognition. It includes the AttendanceController, RFIDService, and FaceRecognitionService. It is the responsibility of this module to verify if the attendance is valid, and it should be stored in the database.
-Reporting Module: Issues reports regarding the attendance data. The ReportingService, which requests the database for reports, may export them to various formats, such as PDF and Excel.
-Facial Recognition Module: This interface integrates OpenCV to capture images and process them for student identification. In turn, the system uses either Haar Cascades or deep learning models to perform face detection and recognition, storing results in the database.
-RFID Module: This reads RFIDs with the help of a microcontroller-ESP 32. Then, the RFID data is used for processing and matching against the database of students in order to mark the attendance of students.

Functional Flow

The functional flow of SAMS is highlighted next through detailed process diagrams:
- Login Process: Depicts the process of a user logging in and all their credentials getting checked, along with the role type and routing to the respective dashboard.
- Attendance Marking: This flow will depict the path from scanning the RFID tag to face recognition, recording in the database, and error handling for unidentified tags/face detection.
Attendance Report Generation: The procedure for generating an attendance report by date range, class, and student, and how to export the report.
Each flow diagram would also include a description of the steps involved, the data being processed, and the interaction among system components.

1. Login Process

Process Flow Table:

Process	Description
Input Username and Password	User enters their username and password into the login form.
Verify Credentials	System checks if the entered username and password match the records in the database.
Check User Role	After validation, the system checks the user's role (Student, Instructor, Admin) to determine routing.
Route to Dashboard	Based on the role, the system redirects the user to the respective dashboard.

Status Flow Table:

Status	Description
Login Form Open	The login form is displayed, awaiting user input.
Credentials Verified	The system verifies the entered credentials against the database.
Routing	The user is routed to the appropriate dashboard based on their role (Student, Instructor, Admin).
Error	If credentials are invalid, an error message is displayed, and the login form is reset.

2. Attendance Marking

Process Flow Table:

Process	Description
Select Attendance Method	Student selects the method for attendance (RFID or Facial Recognition).
Scan RFID / Face Capture	Student scans their RFID tag or uses facial recognition.
Verify RFID/Face Data	The system checks if the scanned RFID tag or captured face matches the stored records.
Mark Attendance	If valid, attendance is marked and stored in the database.
Display Success or Error	System displays a success or error message depending on the validity of the RFID tag or facial recognition.

Status Flow Table:

Status	Description
Select Attendance Method	The system is waiting for the student to select either RFID or facial recognition.
Validating RFID/Face	The system is verifying the scanned RFID or captured facial data against the records in the database.
Attendance Marked	Attendance is successfully marked and recorded in the database.
Error	If the RFID or facial data is invalid, an error message is displayed.

Conclusion

The main philosophy in mind when designing SAMS is the reliability, scalability, and centrality to the users. The system leverages modern technologies such as RFID and face recognition in coming up with a formidable attendance management solution for students in educational institutions. This modular architecture makes SAMS downward-compatible with future requirements and thus a long-term solution for managing attendance.

This design document has to be carefully implemented, since the success of the system lies here. If the development team(Me) follows through with these detailed plans, SAMS will surely achieve its objectives and prove a worthy tool for its intended users.

Glossary

- RFID (Radio-Frequency Identification): A technology that uses electromagnetic fields to automatically identify and track tags attached to objects.
- ESP32: A low-cost, low-power system on a chip with Wi-Fi and Bluetooth capabilities, used in IoT applications.
- RESTful API: An architectural style for designing networked applications, relying on stateless, client-server communication via HTTP.
- OpenCV: An open-source computer vision and machine learning software library used for face detection and recognition.
- MySQL: An open-source relational database management system based on SQL (Structured Query Language).
- Spring Boot: A Java-based framework used to create stand-alone, production-grade Spring-based applications.

Bibliography

- OpenCV Documentation: https://docs.opencv.org/
- Spring Boot Reference Guide: https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
- MySQL Reference Manual: https://dev.mysql.com/doc/refman/8.0/en/
- RFID Technology Overview: https://www.rfidjournal.com/rfid-overview