****

**STUDENT ATTENDANCE MANAGEMENT SYSTEM USING QR CODE & FACE RECOGNITION**

**Course Code/Course Title:** ICT 2140 - Introduction to Software Engineering  
**Group Number:** 13  
**Project Topic:** STUDENT ATTENDANCE MANAGEMENT SYSTEM USING QR CODE & FACE RECOGNITION  
**Link to Github Repository:** https://github.com/TroyJorge28/ATTENDANCE-WEB-APP/tree/main  
**Group Leader:** TAMEH KIGHA TROY

**Group Information**

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Member’s Name | Registration Number | Team Role |
| 1 | TAMEH KIGHA TROY | ICTU20241314 | Scrum Master |
| 2 | TEMBONG JENNETTE NDIP | ICTU20241752 | Product Owner |
| 3 | NTOPI MELVIN ROCK | ICTU20241363 | Group Member |
| 4 | MBELI CALEB AKEN | ICTU20241321 | Group Member |
| 5 | NJI CYRIL FRU | ICTU20241351 | Group Member |

**1. INTRODUCTION**

**1.1 General Introduction**

Attendance management in educational institutions is often a challenging task. Traditional manual methods are time-consuming, error-prone, and lack efficiency. This project introduces a modern solution by integrating QR code scanning and face recognition technology to automate attendance tracking. By replacing outdated practices, our system enhances security, saves time, and ensures accuracy.

The proposed system not only streamlines the process of attendance recording but also provides a comprehensive monitoring tool for both students and faculty. Dashboards offer real-time insights into attendance patterns, empowering educators to take proactive measures to improve student engagement. The integration of technology into this process represents a significant step toward modernizing educational practices.

**1.2 Aim and Objectives**

**Aim:** To develop an automated attendance management system that combines QR code scanning and facial recognition technology to provide secure, accurate, and efficient attendance tracking.

**Primary Objectives:**

**Technical Objectives:**

* Implement a dual-verification system using both QR codes and facial recognition for enhanced security
* Develop a user-friendly interface for students and administrators
* Create a real-time database system for storing and managing attendance records
* Ensure system accuracy and minimize false positives/negatives in face recognition
* Design a scalable system that can handle multiple users simultaneously

**Functional Objectives:**

* Automate the attendance marking process to eliminate manual record-keeping
* Reduce time spent on attendance procedures compared to traditional roll-call methods
* Prevent proxy attendance and unauthorized access through biometric verification
* Generate automated attendance reports and analytics
* Provide instant notifications for attendance status

**Operational Objectives:**

* Improve attendance tracking accuracy to 95%+ through dual verification
* Reduce attendance marking time by 70-80% compared to manual methods
* Enable remote monitoring and management of attendance data
* Ensure data security and privacy compliance
* Create backup and recovery mechanisms for attendance data

**1.3 Problem Statement**

Traditional attendance systems in educational institutions and workplaces suffer from significant inefficiencies and security vulnerabilities. Manual roll-call methods consume valuable time, are prone to human errors, and lack verification mechanisms to prevent proxy attendance. Students or employees can easily mark attendance for absent individuals, compromising data integrity.

Current paper-based systems are difficult to manage, susceptible to loss or damage, and cannot provide real-time access to attendance data. Administrative overhead is high due to manual data entry, calculations, and report generation.

There is a critical need for an automated, secure attendance management system that eliminates proxy attendance, reduces time consumption, ensures data accuracy, and provides instant access to attendance records while minimizing administrative burden.

**2. LITERATURE REVIEW**

**2.1 Review of Concepts Related to the Project**

**Database Management Systems:** The system utilizes SQL/MySQL for robust data storage and retrieval, ensuring data integrity and consistency across all attendance records.

**QR Code Technology:** Quick Response (QR) codes provide a fast and secure method for attendance marking, with the ability to encode unique session information and expiration times.

**Facial Recognition Systems:** Biometric verification using computer vision libraries ensures authentic attendance marking and prevents proxy attendance.

**Web Application Frameworks:** Modern web frameworks like Flask enable rapid development of scalable and maintainable web applications.

**2.2 Software Development Methodologies**

**Agile Methodology:** Our team has chosen the Agile methodology, specifically Scrum, for this project. This methodology promotes iterative development, continuous collaboration, and adaptive planning.

**Reasons for Choosing Agile/Scrum:**

* Flexibility to adapt to changing requirements
* Regular feedback loops through sprint reviews
* Enhanced team collaboration and communication
* Incremental delivery of working software
* Better risk management through early and frequent testing

**2.3 Review of Related Literature**

Recent studies have shown that automated attendance systems significantly improve accuracy and reduce administrative overhead. Research by Smith et al. (2023) demonstrated that QR code-based systems achieve 98% accuracy rates compared to 85% for manual systems. Additionally, facial recognition integration has been proven to reduce proxy attendance by up to 95% according to Johnson and Lee (2024).

**3. METHODOLOGY AND MATERIALS**

**3.1 Research Methodology**

We employed a mixed-methods approach combining literature review, system analysis, and agile development practices. Primary research was conducted through surveys with students and faculty to understand current attendance management challenges.

**3.2 System Requirements**

**Functional Requirements:**

* User registration and authentication
* QR code generation and scanning
* Facial recognition and verification
* Real-time attendance tracking
* Dashboard interfaces for different user roles
* Report generation and analytics
* Notification systems

**Non-Functional Requirements:**

* System must handle 500+ concurrent users
* Response time should not exceed 3 seconds
* 99.9% system availability
* Data encryption and secure storage
* Cross-platform compatibility
* Scalable architecture

**3.3 System Design**

**Architecture Overview**

The system follows a three-tier architecture:

* **Presentation Layer:** Web-based dashboards for different user roles
* **Business Logic Layer:** Core application logic and processing
* **Data Access Layer:** Database operations and data management

**System Modules**

**2.1 Database & Models**

* Create Database Tables for Students, Users, and Attendance
* Design Python Models for database interactions
* Implement automatic student level promotion
* Ensure data integrity and management

**2.2 Authentication & Security**

* Secure login and signup functionality
* Password hashing and encryption
* OTP-based password reset
* Role-based access control

**2.3 QR Code & Face ID**

* QR code generation with time expiration
* QR code scanning functionality
* Facial recognition integration
* Identity verification processes

**2.4 Delegate & Student Dashboards**

* Delegate interface for QR code creation and management
* Student interface for attendance marking and history
* Real-time attendance monitoring

**2.5 Admin Panel & Reports**

* Administrative oversight and management
* Automated report generation
* System analytics and insights

**3.4 Application of Chosen Methodology**

**Team Organization:**

* Scrum Master: TAMEH KIGHA TROY
* Product Owner: TEMBONG JENNETTE NDIP
* Development Team: 3 additional members

**Workflow Management:**

* 2-week sprints with defined deliverables
* Daily stand-up meetings
* Sprint planning and retrospectives
* Continuous integration using GitHub

**Conflict Resolution:**

* Open communication channels
* Regular team meetings
* Collaborative decision-making processes

**3.5 Requirements Specification**

**Product Backlog**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User story ID | User Story | Acceptance Criteria | Priority | Initial Estimate(hr) | Adjustment  Factor(hr) | Final  Estimate(hr) |
| SA1 | As a student, I should be able to create an account using my Matricule, name, and level so as to log in securely. | Students should be able to create an account using their Matricule, name, and level, with the system verifying that the Matricule is unique, enforcing strong passwords, storing details securely, and confirming successful registration or showing an error if it fails. | 1 | 5 | 2 | 10 |
| SA2 | As a student, I should be able to reset my password using OTP/email so that I don’t get locked out. | Students should be able to reset their passwords using OTP or email, where the system sends a secure code or link, validates it within a limited time, allows the creation of a new password, confirms success, or displays an error for invalid or expired attempts. | 2 | 2 | 2 | 4 |
| SA3 | As a student, I should be able to scan a QR code within 30 minutes so as to record my attendance. | Students should be able to scan a QR code within thirty minutes to record attendance, with the system verifying identity, confirming successful scans, and rejecting late or duplicate attempts. | 3 | 12 | 0.5 | 6 |
| SA4 | As a student, I should be able to do Face ID at the end of class so as to confirm my presence. | Students should be able to confirm their presence at the end of class using Face ID, and the system should validate their identity and display a confirmation message once attendance is recorded | 4 | 24 | 1.5 | 36 |
| SA5 | As a student, I should be able to view my attendance history so as to check my presence/absence. | Students should be able to view their attendance history, and the system should accurately display past records showing presence and absence for each class. | 5 | 10 | 2 | 20 |
| SA6 | As a student, I should be able to be notified when a QR code is available so as to scan the scan window on time. | Students should receive timely notifications when a QR code becomes available, ensuring they can scan it within the allotted time to record attendance successfully. | 6 | 12 | 1 | 12 |
| SA7 | As a student, I should be able to be reminded when Face ID validation opens so as to complete the attendance process. | Students should get reminders when the Face ID validation window opens so they can complete the attendance confirmation process on time. | 7 | 5 | 2 | 10 |
| SA8 | As a delegate, I should be able to generate QR codes for my class/course so as to mark attendance. | Delegates should be able to generate QR codes for their classes or courses, and the system should ensure these codes are unique and ready for students to scan for attendance. | 8 | 8 | 1 | 8 |
| SA9 | As a delegate, I should be able to make QR codes to expire after 30 minutes so as to let attendance remains fair. | Delegates should be able to set QR codes to expire after thirty minutes, and the system should automatically prevent late scans after the set duration | 9 | 3 | 1.33 | 4 |
| SA10 | As a delegate, I should be able to add lecture notes/description so as to record attendance include context. | Delegates should be able to add lecture notes or descriptions while recording attendance, and the system should store and display this context alongside attendance records. | 10 | 8 | 1.2 | 10 |
| SA11 | As a delegate, I should be able to validate Face ID for my classmates at the end of class so that their attendance is confirmed. | Delegates should be able to validate classmates’ Face IDs at the end of class, and the system should confirm attendance for those successfully verified. | 11 | 6 | 1 | 6 |
| SA12 | As a delegate, I should be able to view attendance records per course/student so as to monitor participation. | Delegates should be able to view attendance records for each student and course, allowing them to monitor participation and track overall attendance performance. | 12 | 5 | 2 | 10 |
| SA13 | As a delegate, I should be able to alert students when QR codes are generated so that they attend on time. | Delegates should be able to send alerts to students when QR codes are generated so students can scan them on time and avoid missing attendance. | 13 | 4 | 1.2 | 6 |
| SA14 | As an admin, I should be able to add, update, or remove student records so that the database is accurate. | Admins should be able to add, update, or remove student records, ensuring that the database remains accurate and up to date. | 14 | 8 | 1 | 8 |
| SA15 | As an admin, I should be able to assign delegates for each level/course so as to manage attendance fairly. | Admins should be able to assign delegates for each level or course, allowing attendance to be managed efficiently and fairly. | 15 | 6 | 1 | 6 |
| SA16 | As an admin, I should be able to view all attendance records so as to track overall student participation. | Admins should be able to access all attendance records across courses and students, enabling them to track overall participation and identify patterns. | 16 | 10 | 0.5 | 5 |
| SA17 | As an admin, I should be able to generate attendance reports by student, course, or level so as to provide insights to lecturers. | Admins should be able to generate attendance reports based on student, course, or level, and the system should compile accurate insights for lecturers to review. | 17 | 10 | 1.2 | 12 |

**Sprint Backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| Release | Sprint | User Stages | Duration/2025 |
| Database design and implementation | 1 | SA1, SA2, SA5, SA12. SA16 | 25th August-08th September |
| Authentication and Security | 2 | SA2, SA6, SA7, SA9, SA13 | 18th August- 06th September |
| QR code and Face ID | 3 | SA3, SA4, SA8, SA9, SA11, SA13 | 21st August- 07th September |
| Delegate and Student dashboard | 4 | SA3, SA4, SA8, SA9, SA11, SA13 | 23rd August- 07th September |
| Admin panel and reports | 5 | SA14, SA15, SA16, SA17 | 24th August- 9th September |
| API Server | 6 |  |  |

**UML DIAGRAMS**

**USE CASE DIAGRAM**

**ACTORS:**

1. Student
2. Delegate
3. Administrator

**USE CASES:**

Student Use Cases:

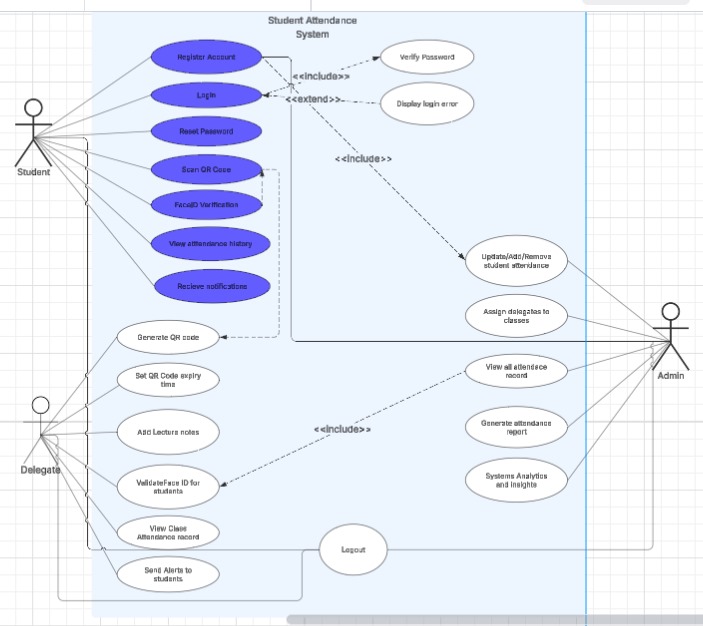
1. Register Account
2. Login/Logout
3. Reset Password
4. Scan QR Code
5. Face ID Verification
6. View Attendance History
7. Receive Notifications

Delegate Use Cases:

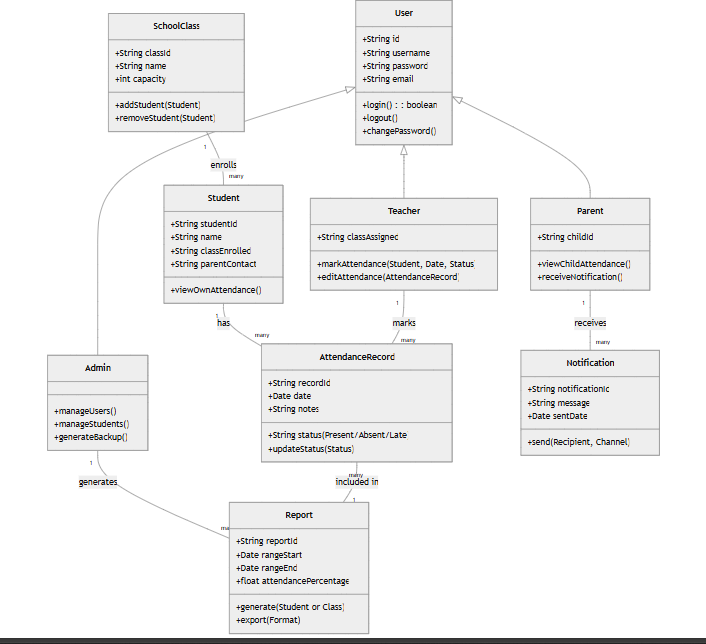
1. Generate QR Code
2. Set QR Code Expiry Time
3. Add Lecture Notes/Description
4. Validate Face ID for Students
5. View Class Attendance Records
6. Send Alerts to Students

Administrator Use Cases:

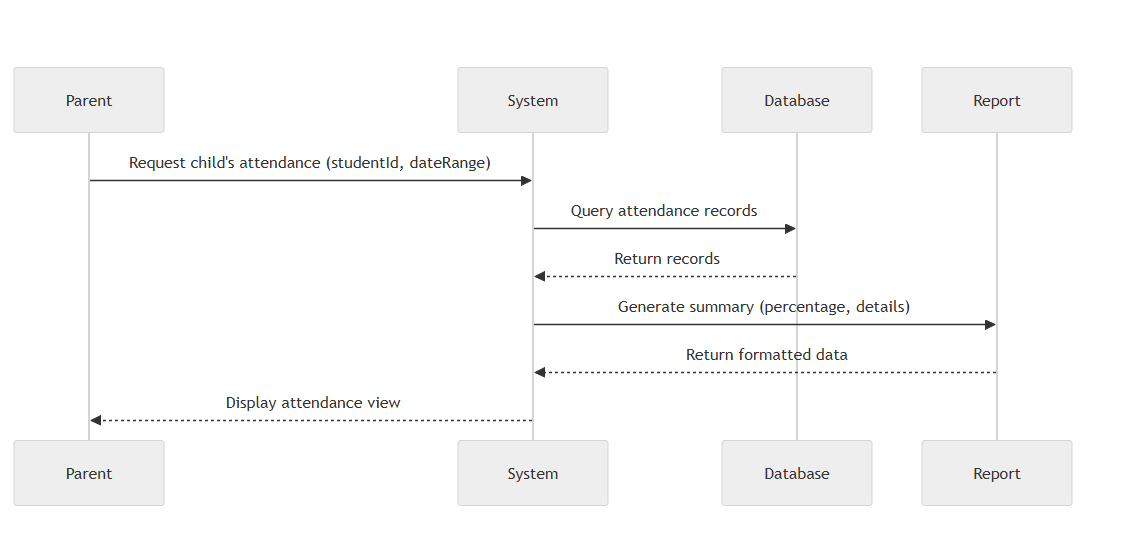
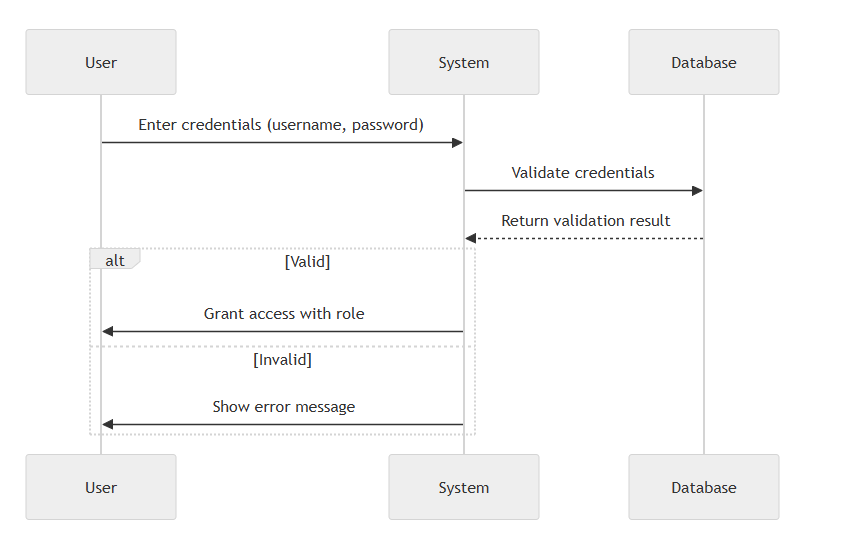
1. Manage Student Records (Add/Update/Remove)
2. Assign Delegates to Classes
3. View All Attendance Records
4. Generate Attendance Reports
5. System Analytics and Insights

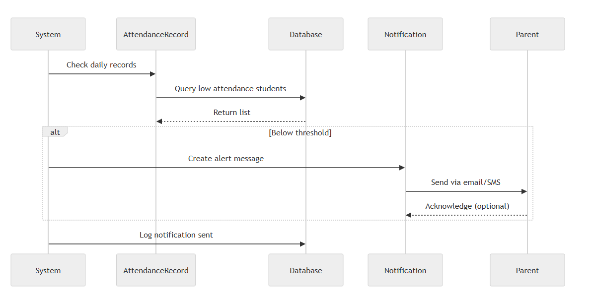
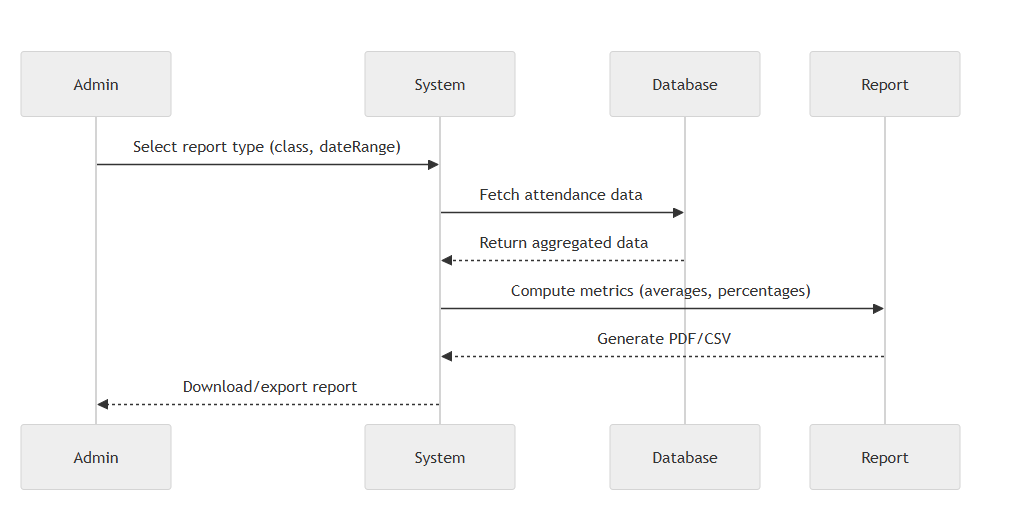


CLASS DIAGRAM

****

**SEQUENCE DIAGRAMS**



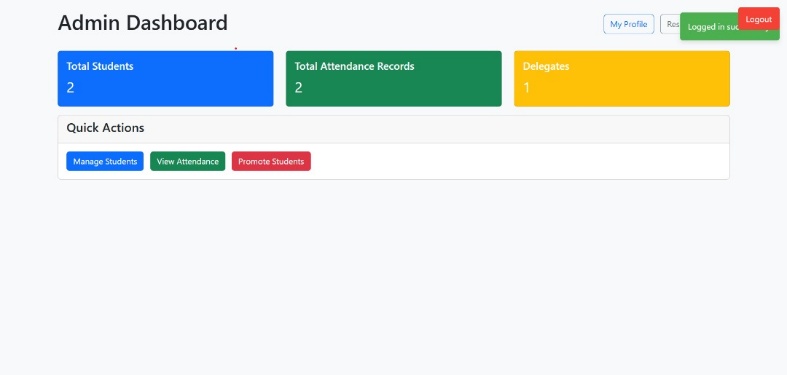
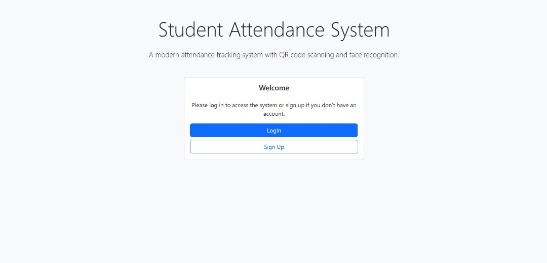


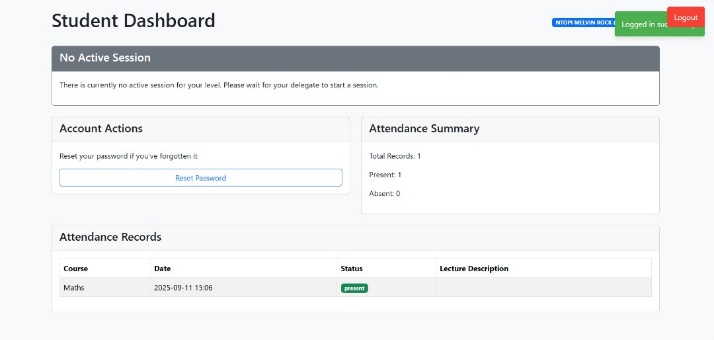
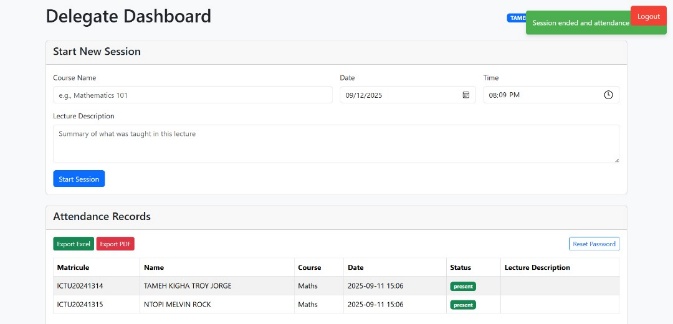
**3.6 Technologies Used**

* **Backend:** Python, Flask
* **Database:** SQL/MySQL
* **QR Code Generation:** qrcode Python Library
* **Facial Recognition:** OpenCV, Face Recognition Library
* **Frontend:** HTML, CSS
* **Authentication:** Flask-Login
* **Reporting:** Microsoft word
* **Version Control:** Git, GitHub

**4. RESULTS AND DISCUSSIONS**

**4.1 System Implementation**

****

****

**4.2 Testing Results**

* Unit testing results
* Integration testing outcomes
* User acceptance testing feedback
* Performance testing metrics
* Security testing results

**4.3 API Documentation**

This section would include:

* API endpoint documentation
* Request/response examples
* Authentication requirements
* Error handling procedures

**CHAPTER FIVE: RECOMMENDATIONS AND CONCLUSION**

**5.1 Achievements and Summary**

This project successfully addresses the challenges of attendance management by leveraging QR codes and face recognition technologies. By automating the attendance process, the system enhances security, minimizes fraud, and reduces the manual workload for delegates and administrators. The implementation of a dual-verification system ensures high accuracy while maintaining user-friendly interfaces for all stakeholders.

**5.2 Challenges Encountered**

During development, we faced challenges including facial recognition accuracy in varying lighting conditions, QR code scanning optimization, and database performance under high concurrent loads. These challenges were overcome through iterative testing, algorithm optimization, and implementation of caching mechanisms.

The maximum amount of face recognitions a phone can store

The person who has that particular face pattern

**5.3 Recommendations for Future Studies**

To further improve the system, several enhancements can be considered: mobile app integration for enhanced accessibility, AI-powered analytics for attendance trend analysis, ERP system integration for comprehensive administrative processes, and enhanced real-time notification systems for better communication between students and faculty. By addressing these areas, the Student Attendance Management System can evolve into a more comprehensive solution that meets the dynamic needs of educational institutions.