SOFTWARE ENGINEERING & CONCEPTS – LAB MANUAL

[ATTENDENCE MANAGEMENT SYSTEM]

<<SHAUN PAUL MOSES>> **<<220701266>>**

<<SHREYAS GOWD LC>> **<<220701272>>**

<<SIHABUTHEEN HAQ PS>> **<<220701278>>**

<<SOWMIYANARAYANNAN GK>> **<<220701284>>**

<<SUJITH SS>> **<<220701290>>**

<<SWEDHA J>> **<<220701296>>**

<<BE Computer science and engineering>>
<<2023-2024>> <<E SECTION>>

Software Concepts & Engineering - Lab Manual

Overview of the Project

Please explain the details of the project in terms of

- What is the problem it is trying to resolve

- When students have attendance shortage, they get to know it only a few dats before the end semester exam as the teachers forget to upload everyday's attendance. Due to this students face attendance shortage and are asked to attend the special classes or if they can't the student will not be permitted to attend the examination.
- Most of the professors take more than 10 minutes to take attendance. So just for taking attendance more than an hour is wasted every day.

- How it would help users when we implement the system

Efficiency: Users, both students, and faculty, will experience increased efficiency in managing attendance-related tasks. Requesting leave or updating attendance can be done quickly through the system, saving time compared to manual processes.
Convenience: Students can easily enter their attendance using their phones, which eliminates the need for manual attendance sheets or physical presence in a specific location. This flexibility is especially helpful for students who may have scheduling conflicts or who are unable to attend class physically.
Data Analysis: Over time, the system accumulates a wealth of attendance data that can be analyzed to identify patterns, trends, and insights. This information can be valuable for assessing student engagement, identifying areas for improvement, and making data-driven decisions.

Business Architecture Diagram

Explain the Business Need of the project, explain

- The current process (either manual or automatic). How the current process is working

Attendance Taking: In a traditional classroom setting, instructors use paper attendance sheets to record students' attendance. This involves calling out each student's name or checking off their names as they enter the classroom.

Leave Requests: If a student needs to take leave for any reason (illness, personal reasons, etc.), they typically have to inform the instructor verbally or through some written communication (such as email or a leave application form).

- Explain the same for the different personas the system would be used

- **Instructor/Class Incharge**:
  - **Attendance Taking**: With the new system, instructors access a digital interface where they can quickly view a list of enrolled students and mark their attendance with just a few clicks or taps. The system may also offer options for taking attendance through mobile devices, such as smartphones or tablets, making it more convenient for instructors to manage attendance even outside the classroom.

- **Leave Requests**: Students can submit leave requests through the system's interface, specifying the reason for leave and the duration of absence. Instructors receive notifications of these requests and can approve or deny them directly through the system. The system maintains a record of all leave requests and their status, providing a convenient way for instructors to manage student absences.

- **Student**:

- **Attendance Taking**: Students use the system's mobile app or web interface to mark their attendance when they enter the classroom. They may scan a QR code displayed in the classroom or check in through a geo-location feature to confirm their presence. This eliminates the need for manual roll call and ensures accurate and efficient attendance tracking.
- **Leave Requests**: Instead of approaching the instructor directly, students submit leave requests through the system, providing all necessary details electronically. They receive notifications about the status of their requests and can view any approved leave periods in their attendance records, ensuring transparency and clarity regarding their attendance status.

-

- Explain the business problems

1. **Initial Implementation Costs**: The initial setup and implementation of an automated attendance management system can require significant investment in terms of software development, hardware procurement (if any), and training for users. This initial cost might be a concern for educational institutions with limited budgets. **User Adoption and Training**: Despite its benefits, introducing a new system may face resistance from users who are accustomed to manual processes or who may perceive the system as complex or unnecessary.

Providing adequate training and support for users to understand and utilize the system effectively is crucial for successful adoption.

2. **Technical Issues and Downtime**: Like any software system, an automated attendance management system may experience technical issues, bugs, or downtime, which can disrupt operations and impact user satisfaction. Ensuring robust technical support and maintenance processes are in place to address these issues promptly is essential.

Requirements as User Stories

User Story 1: As a faculty member (if the internet is available), I want to enter the attendance for the students who are present for today's class.
User Story 2: As a faculty (if the internet is available), I want to enter the attendance for the students who are present for today's class.
User Story 3: As a faculty, I want to edit the given attendance details if I am marked wrong details (within the given short period of time).
User Story 4: As a student, I want to submit the on duty letter to the respective staffs.
User Story 5: As a student I want to submit my leave letter to the respective class incharge.

User Story 6: As a user, I want to know the holidays.

User Story 7: As a student, I want to know whether my leave letter is accepted or not by the respective class in charge.

Poker Planning for User Stories:

1. **User Story 1: Enter Attendance for Present Students**

- This involves creating a user interface for faculty to enter attendance and updating the database.
- Estimation: 3 story points

2. **User Story 2: Enter Attendance for Present Students (Alternate)**

- This seems identical to User Story 1. If it's truly the same, it should have the same estimate.
- Estimation: 3 story points

3. **User Story 3: Edit Attendance Details**

- This involves implementing a feature for faculty to edit attendance within a short period.
- Estimation: 2 story points

4. **User Story 4: Submit On Duty Letter**

- This involves creating a feature for students to submit OD requests.
- Estimation: 3 story points

5. **User Story 5: Submit Leave Letter**

- This involves creating a feature for students to submit leave requests.
- Estimation: 3 story points

6. **User Story 6: View Holidays**

- This involves implementing a feature to display holiday information to users.
- Estimation: 2 story points

7. **User Story 7: Check Leave Letter Status**

- This involves creating a feature for students to check the status of their leave requests.
- Estimation: 2 story points

NFR's

1. **Performance:**

- The system should be able to handle concurrent attendance submissions and request processing efficiently, ensuring a smooth user experience even during peak usage times.

- Response times for attendance submission, request processing, and system operations should be within acceptable limits to prevent user frustration.

2. **Security:**

- Data transmission between the mobile application, portal, and database should be encrypted using industry-standard protocols (e.g., SSL/TLS) to ensure confidentiality and integrity.
- Access to student attendance records, OD, and Leave requests should be restricted to authorized users only, with appropriate authentication and access control mechanisms in place.

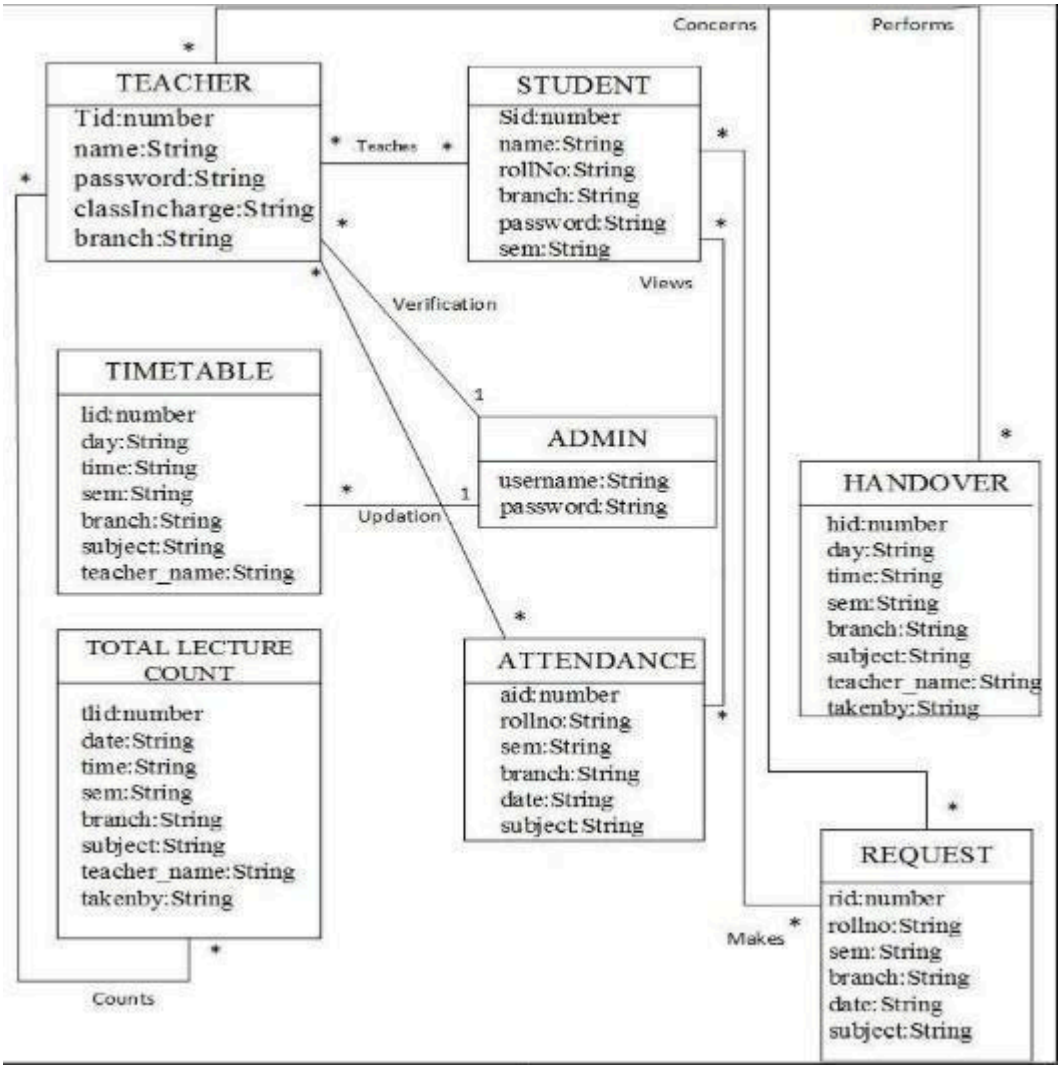
3. **Scalability:**

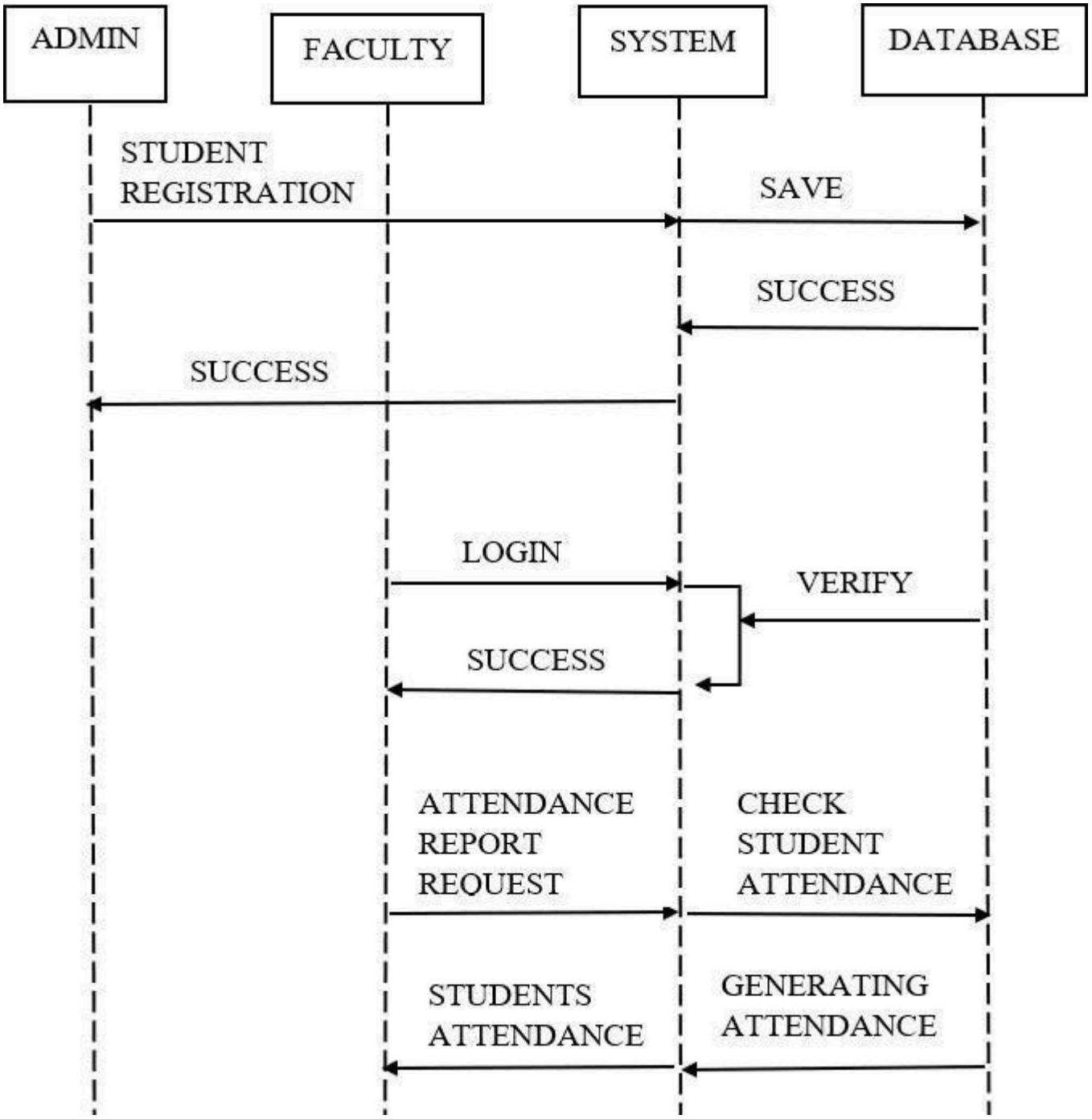
- The system should be designed to accommodate future growth in user base and data volume without significant performance degradation or architectural overhaul.
- The architecture should be scalable both vertically (adding more resources to existing servers) and horizontally (adding more servers to distribute load) to handle increased demand.

These NFRs ensure that the Attendance Management System not only functions correctly but also meets the performance, security, and scalability requirements essential for its successful deployment and operation.

Architecture Diagram

Class diagram




ADMIN
FACULTY
SYSTEM
DATABASE
STUDENT REGISTRATION
SAVE
SUCCESS
SUCCESS
LOGIN
VERIFY
SUCCESS
ATTENDANCE REPORT REQUEST
CHECK STUDENT ATTENDANCE
STUDENTS ATTENDANCE
GENERATING ATTENDANCE

m:

Test Strategy

A well-defined test strategy is crucial for ensuring the functionality, reliability, and security of your attendance management system. Here's a breakdown of key aspects to consider:

1. Test Objectives:

- Define clear objectives for your testing efforts. What are you trying to achieve with the testing process? Some examples include:
    - Verifying the system accurately captures and stores attendance data.
    - Ensuring different attendance marking methods (fingerprint, RFID, mobile app) function properly.
    - Validating user access controls and authorization levels.
    - Testing report generation functionality for attendance summaries and data analysis.
    - Evaluating system performance and scalability under various user loads.
    - Identifying and addressing potential security vulnerabilities.

2. Test Scope:

- Determine the functionalities and features that will be included in the testing process. This might involve prioritizing critical features first and then progressively testing additional functionalities.

3. Test Levels:

- Implement a multi-level testing approach to comprehensively evaluate the system:

- **Unit Testing:** Focuses on testing individual software units (modules, functions) to ensure they operate as designed.
- **Integration Testing:** Tests how different system components interact and exchange data with each other.
- **System Testing:** Evaluates the entire system as a whole, simulating real-world user scenarios and workflows.
- **User Acceptance Testing (UAT):** Involves actual users testing the system to ensure it meets their expectations and is user-friendly.

4. Test Techniques:

- Utilize various testing techniques to cover different aspects of the system:
  - **Functional Testing:** Verifies if the system behaves as per its specifications.
  - **Non-Functional Testing:** Evaluates non-functional aspects like performance, security, usability, and scalability.
  - **Regression Testing:** Re-executes critical test cases after code modifications to ensure no regressions are introduced.
  - **Security Testing:** Identifies and mitigates potential security vulnerabilities in the system.

5. Test Environment:

- Set up a dedicated test environment that mirrors the production environment as closely as possible. This ensures tests accurately reflect how the system will behave in real-world use.

6. Test Data Management:

- Prepare realistic test data that simulates various attendance scenarios (normal working hours, overtime, leaves, exceptions).
- Implement mechanisms to manage test data effectively, including data creation, modification, and cleanup.

7. Test Reporting and Defect Management:

- Document test results comprehensively, including passed and failed test cases.
- Utilize a defect management tool to track identified issues, their severity, and their resolution process.

8. Test Automation:

- Consider automating repetitive test cases to improve efficiency and reduce manual testing efforts. Tools like Selenium can be used for web application testing, while tools like JUnit can be used for unit testing.

Additional Considerations:

- **Security Testing:** Pay particular attention to security testing, including penetration testing to identify vulnerabilities and ensure user data is protected.
- **Performance Testing:** Conduct performance testing to evaluate system behavior under various user loads and ensure it can handle anticipated usage.
- **Usability Testing:** Involve potential users in usability testing to identify any difficulties or confusing aspects of the user interface.

By following these guidelines and tailoring them to your specific attendance management system, you can create a comprehensive and effective testing strategy that ensures its quality and reliability.

Deployment Architecture of the application

Deployment architecture for an attendance management system, along with a corresponding diagram:

Components:

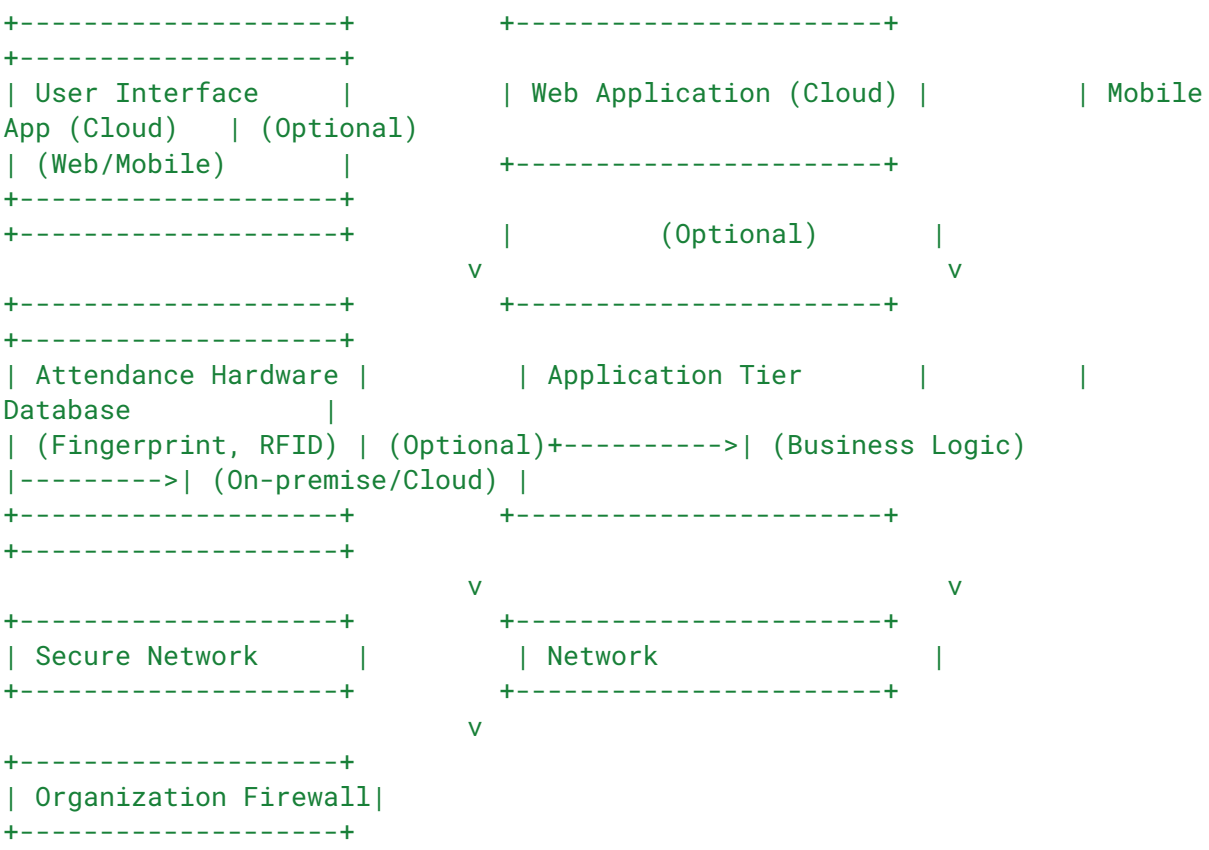
- Presentation Tier:
  - Web Application (optional): Users access the system through a web browser to view attendance records, reports, etc. This can be deployed on-premise or in the cloud.
  - Mobile App (optional): Users can mark attendance using features like QR code scanning, facial recognition, or geolocation.
  - Hardware (optional): Depending on the chosen attendance method (fingerprint scanner, RFID reader), specific hardware might be deployed at designated locations.
- Application Tier (Business Logic):
  - This tier handles core functionalities like processing attendance data, managing user accounts, and generating reports. It can be deployed on a web server (on-premise or cloud).
- Data Tier:
  - Database: Stores attendance data, user information, and other relevant system data. This can be a relational database management system (MySQL, PostgreSQL) deployed on a separate server (on-premise or cloud).

- Network:
  - A secure network connects all components, allowing communication between the user interface, application tier, and database.

Deployment Options:

- On-Premise Deployment
- Cloud Deployment
- Hybrid Deployment (combination of on-premise and cloud)

Deployment Architecture Diagram:



Explanation:

- The diagram depicts three possible user interfaces: a web application, a mobile app, and optional attendance hardware.
- Users can interact with the system through their chosen interface.
- The application tier processes attendance data and interacts with the database.
- The database stores all system data.
- A secure network connects all components, with the organization's firewall providing an additional security layer.

Choosing a Deployment Option:

Consider factors like scalability, security, IT expertise, and cost when selecting a deployment option.

Additional Considerations:

- Implement strong security measures.
- Ensure data backup and recovery.
- Consider offline functionality for mobile apps (if applicable).
- Adhere to data privacy regulations.

By following these guidelines, you can create a secure and efficient deployment architecture for your attendance management system.