SOFTWARE ENGINEERING CONCEPTS (LAB MANUAL)

Software Concepts & Engineering - Lab Manual

OVERVIEW

An On-Duty Management System (ODMS) is designed to manage, track, and optimize the scheduling and duties of students who are on duty whether it is inter-college events or intra-college events.

Key components:

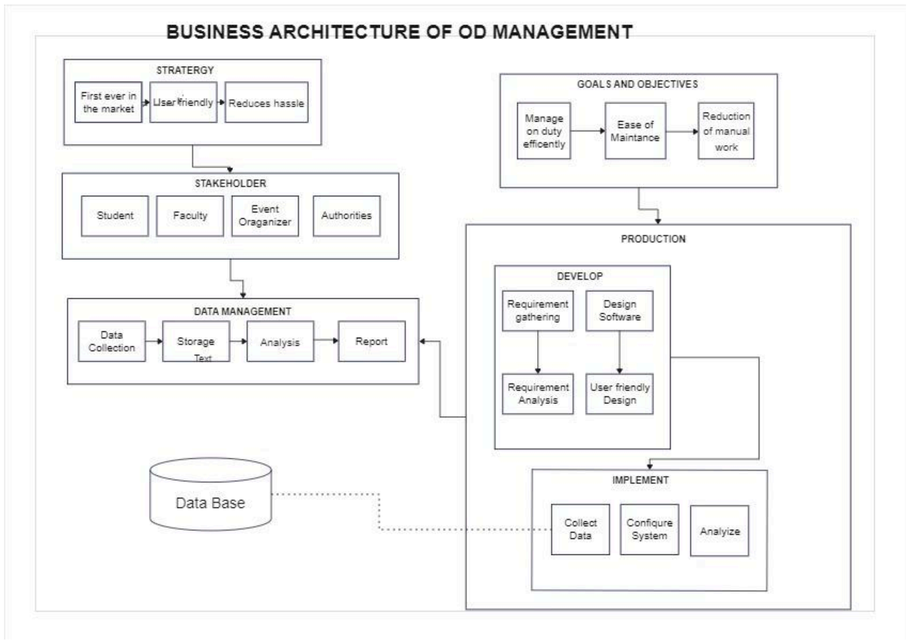
User roles:
*Faculty : adds/ deletes events , manages events , accepts/ denies OD request sent by student
*Student :request OD

OD type:
*Inter college
*Intra college

OD Request process:
*Submission: Students submit OD requests through the system.
*Approval of OD: Requested are routes to the faculty for approval.
*Notification: Students and faculties receive notification about OD request status.
*Tracking: OD taken is tracked and balanced are upload accordingly.

Benefits :
*Efficiency : Streamlines the OD request and approval process, reducing paperwork and manual errors.
*Transparancy : Provides clear visibility into OD balances and minimizing misunderstandings.

BUSINESS ARCHITECTURE



Requirements as user stories

As a student, I should be able to log in to the website using my university credentials so that I can access the website
As a student, I must be able to reset my password so that I can log in even if I forget the password
As a Faculty, I should be able to log in to the website using my credentials so that I can access the website
As a Faculty, I must be able to add or remove events from the list so that students can avail for OD for those events
As a student,I must be able to select events from the list and attach proof so that the faculty can verify and grant me od
As a student, I must be able to track my OD so that I can check the progress of my od request
As a faculty, I must be able to view the students OD request with proof so that i can verify and provide od for the students
As a faculty, I must be able to send an od letter as a pdf to the students so that they can use it for proof to other faculty members
As a student, I must be able to view if my OD request is approved or denied so that I can take appropriate action
As a student, I should receive notifications about the status of my OD request so that I am informed of any updates.

Poker Planning Estimation For User stories:

Student login: 3 story points

Password reset: 5 story points

Faculty login: 3 story points

Add/remove events: 8 story points

Select events and attach proof: 5 story points

Track OD request: 5 story points

View OD requests with proof: 5 story points

Send OD letter as PDF: 8 story points

View OD request status: 3 story points

Non Functional Requiremnts:

Performance:

The system should respond to user requests within 2 seconds for 95% of the requests.

Usability:

User Interface: The system should have an intuitive and user-friendly interface, with clear navigation and helpful tooltips.

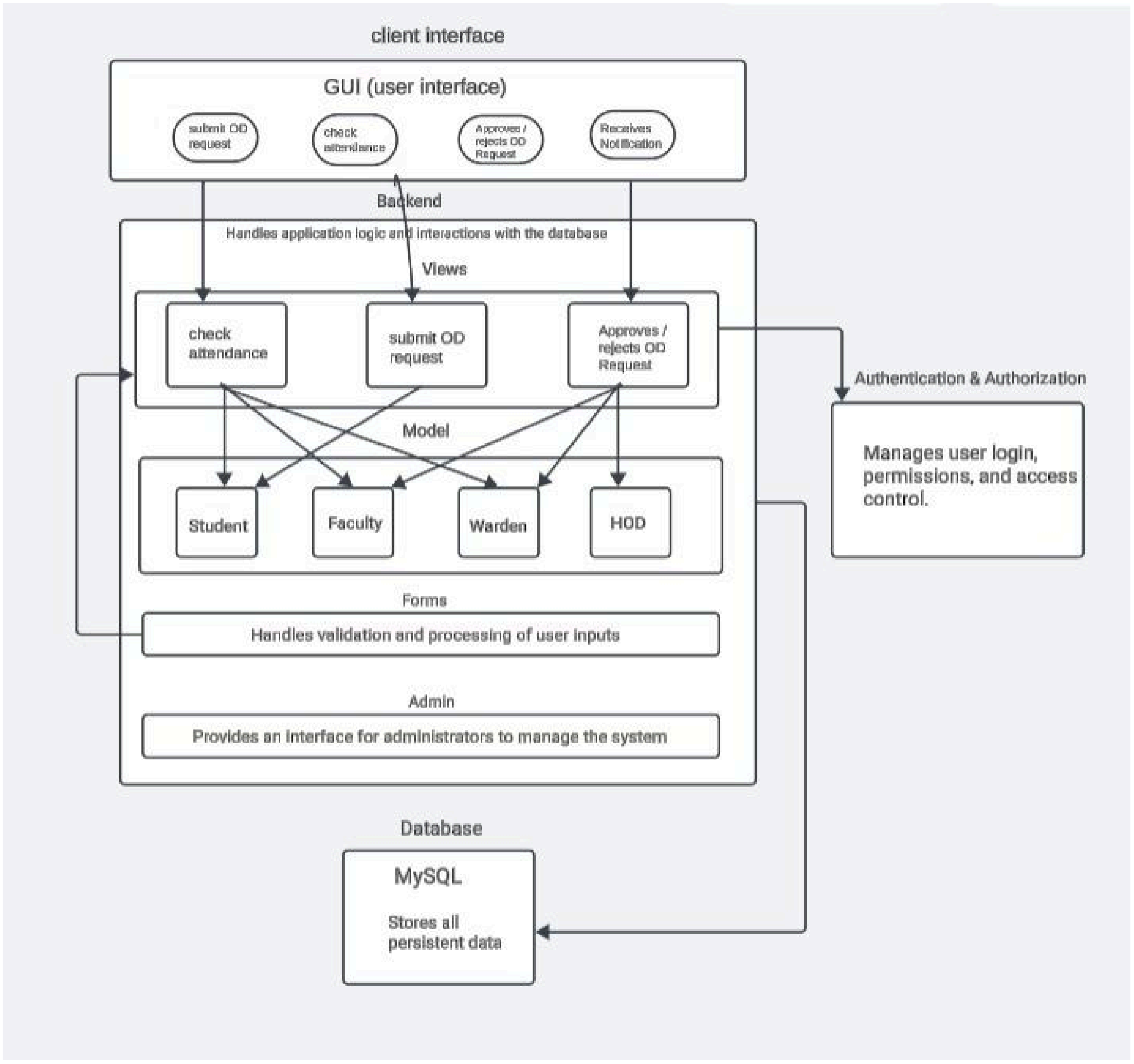
Reliability:

Error Handling: The system should handle errors gracefully and provide meaningful error messages to users.

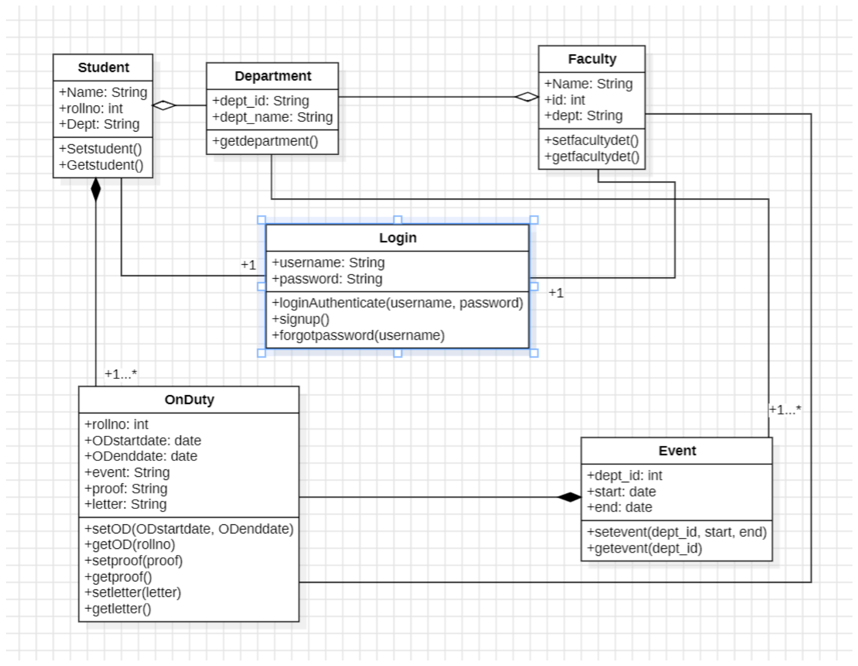
Security

Data Encryption: All sensitive data, such as login credentials and OD request details, should be encrypted both in transit and at rest.

ARCHITECTURE DIAGRAM

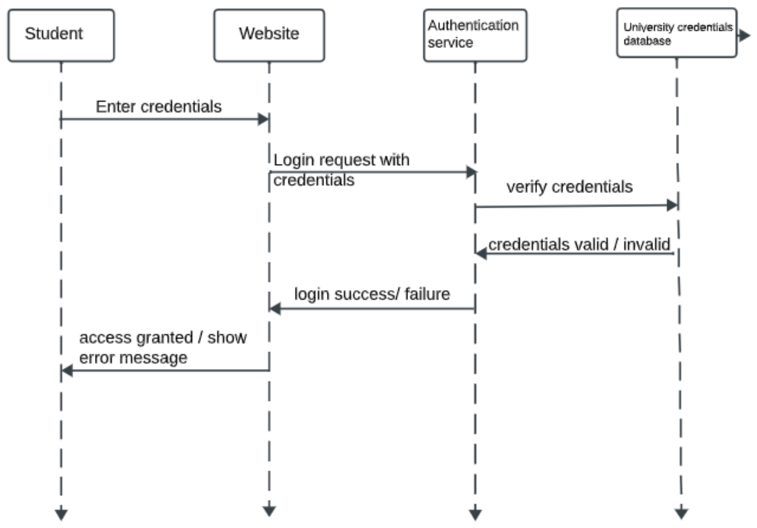

client interface
GUI (user interface)
submit OD request
check attendance
Approves / rejects OD Request
Receives Notification
Backend
Handles application logic and interactions with the database
Views
check attendance
submit OD request
Approves / rejects OD Request
Authentication & Authorization
Manages user login, permissions, and access control.
Model
Student
Faculty
Warden
HOD
Forms
Handles validation and processing of user inputs
Admin
Provides an interface for administrators to manage the system
Database
MySQL
Stores all persistent data

Class Diagram

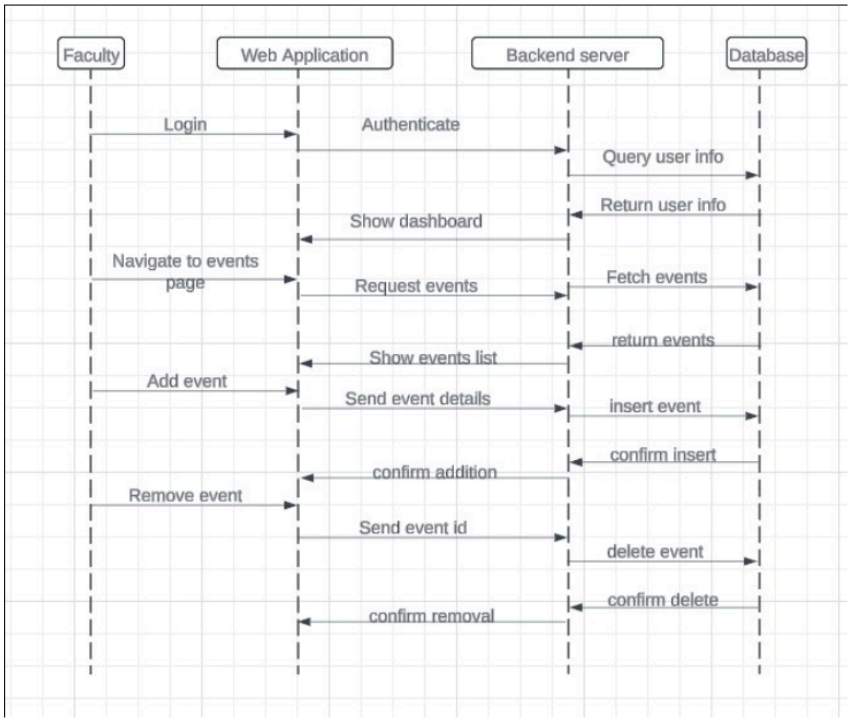


Sequence Diagram

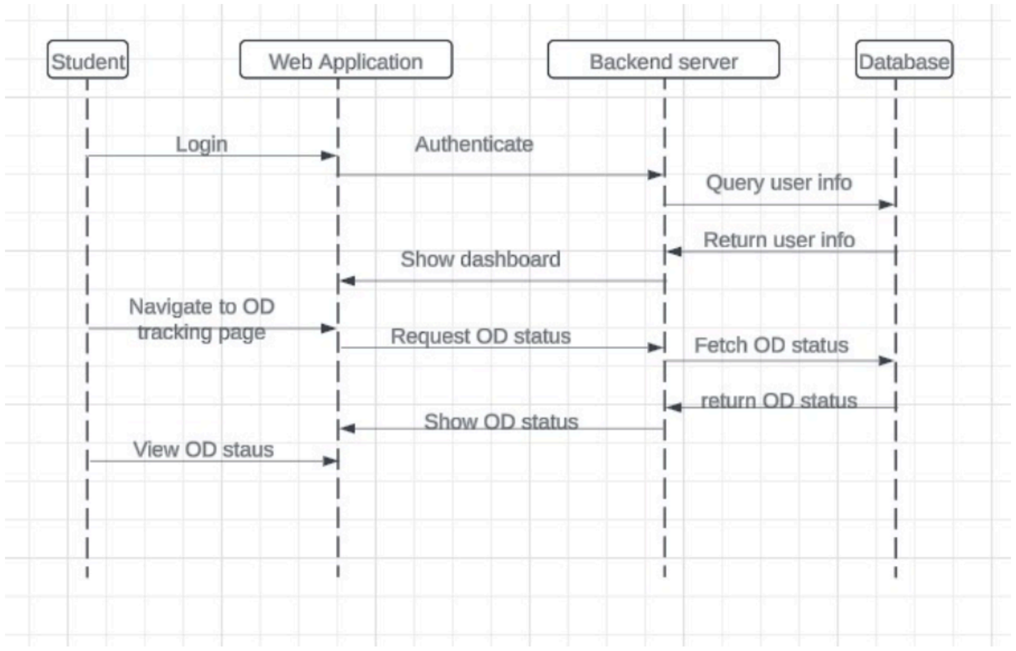
1)User story:As a student, I should be able to log in to the website using my university credentials so that I can access the website


Student
Website
Authentication service
University credentials database
Enter credentials
Login request with credentials
verify credentials
credentials valid / invalid
login success/ failure
access granted / show error message

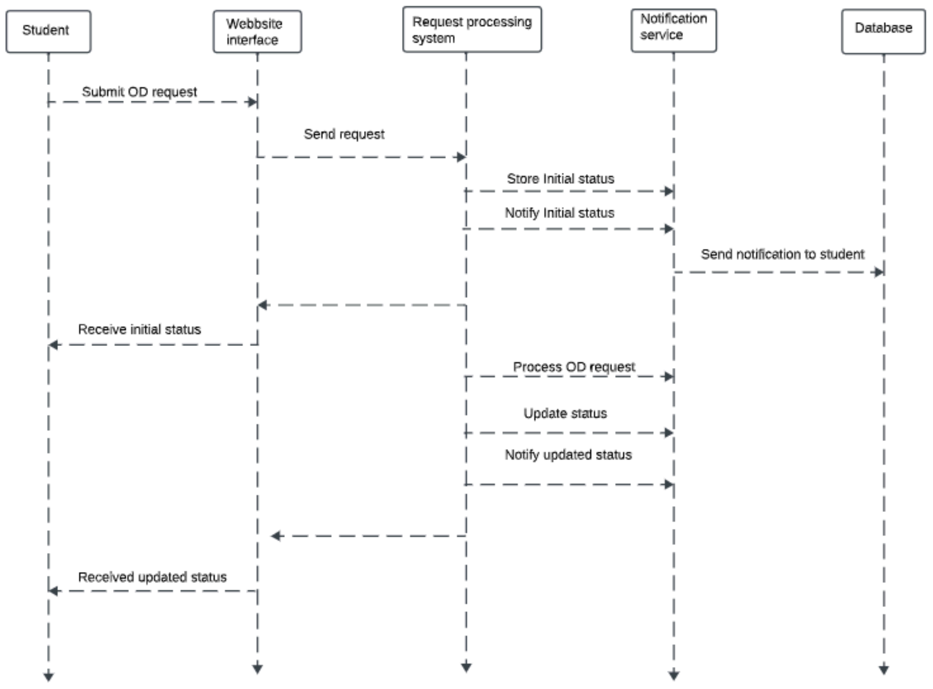
2) User story :As a Faculty, I must be able to add or remove events from the list so that students can avail for OD for those events



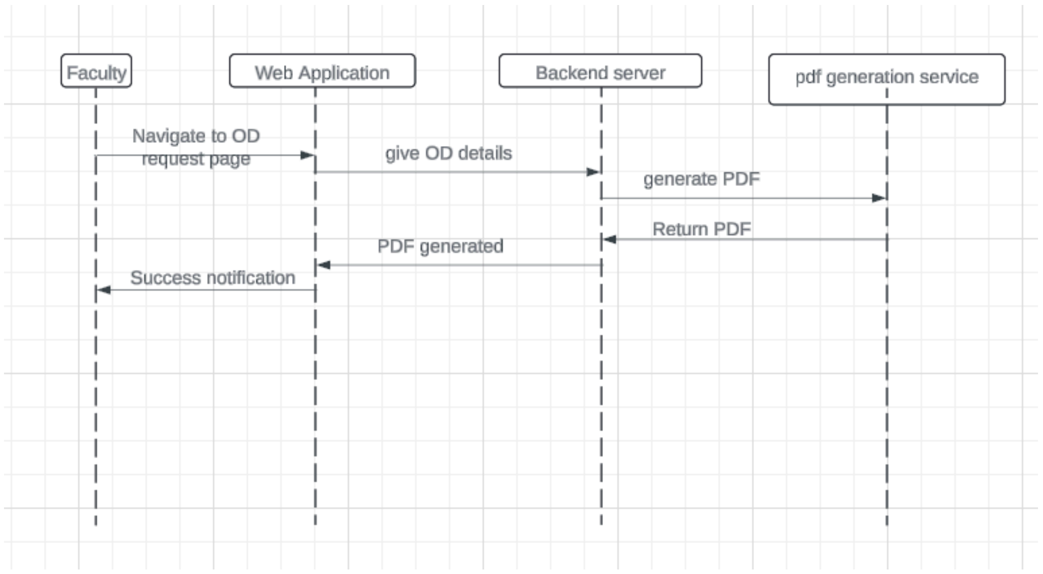
3)User story:As a student, I must be able to track my OD so that I can check the progress of my OD request



4)User story:As a student, I should receive notifications about the status of my OD request so that I am informed of any updates.



5)User story:As a faculty, I must be able to send an od letter as a pdf to the students so that they can use it for proof to other faculty members



TEST STRATEGY FOR ON-DUTY MANAGEMENT SYSTEM :

1. Test Scope and Objectives:
   - Define the scope of testing, including functional areas (e.g., duty scheduling, assignment, reporting).
   - Specify the objectives: validate system functionality, ensure data accuracy, and verify compliance with business rules.

2. Test Levels:
   - Unit Testing: Validate individual components (e.g., duty assignment algorithms).
   - Integration Testing: Verify interactions between modules (e.g., shift handovers).
   - System Testing: Validate end-to-end scenarios (e.g., duty creation, assignment, and reporting).

3. Test Data Strategy:
   - Generate synthetic data for duties, shifts, personnel, and roles.
   - Cover various scenarios (e.g., overlapping shifts, different duty types).

4. Test Environment:
   - Set up test environments mirroring production (e.g., databases, servers).
   - Ensure data privacy and security, especially when handling sensitive information.

5. Functional Testing:
   - Test the following scenarios:
     - Duty creation and assignment.
     - Shift handovers and continuity.
     - Duty swaps or replacements.
     - Reporting and analytics.
     - Compliance with labor laws (e.g., maximum working hours).
     - User roles (e.g., admin, supervisor, staff).

6. Non-Functional Testing:
   - Performance Testing: Evaluate system responsiveness under varying loads.
   - Security Testing: Verify access controls, data encryption, and user authentication.
   - Usability Testing: Assess user-friendliness and navigation.

7.Regression Testing:
   - Ensure that new features or bug fixes do not impact existing functionality.

SAMPLE TEST CASES:

1. User Story: Create a New Duty:
   - Happy Path:
     - Verify that a new duty can be created with valid details (date, time, location).
     - Confirm that the duty appears in the system after creation.
   - Error Scenarios:
     - Attempt to create a duty with missing or invalid information (e.g., blank date).
     - Check error messages for proper validation.

2. User Story: Assign Duty to Staff:
   - Happy Path:
     - Assign a duty to a staff member.
     - Validate that the staff member's schedule reflects the assigned duty.
   - Error Scenarios:
     - Try to assign a duty to an unavailable staff member (already assigned elsewhere).
     - Verify error handling for conflicting assignments.

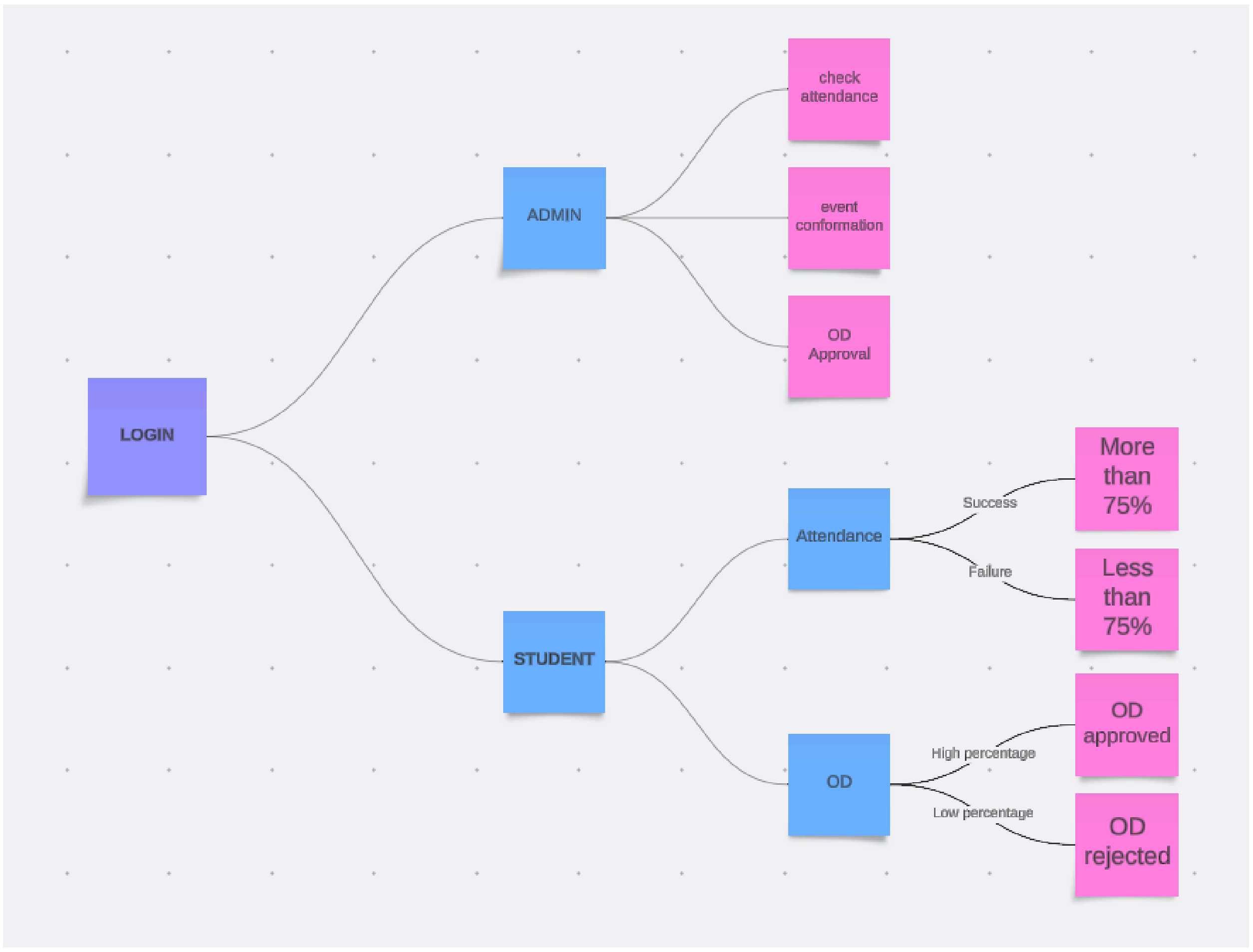
3. User Story: Generate Duty Reports:
   - Happy Path:
     - Generate duty reports for a specific date range.
     - Ensure accurate representation of assigned duties.
   - Error Scenarios:
     - Attempt to generate a report with invalid date ranges.
     - Check for missing or incorrect data in the report.

4. User Story: Swap Duties:
   - Happy Path:
     - Allow staff members to request duty swaps.
     - Validate that swapped duties are reflected correctly.
   - Error Scenarios:
     - Verify handling of invalid swap requests (e.g., requesting a swap with an incompatible duty).

5. User Story: Compliance with Working Hours:
   - Happy Path:
     - Assign duties while adhering to maximum working hours per day/week.
     - Confirm that the system prevents overloading staff.

- Error Scenarios:
  - Assign duties that exceed legal working hour limits.
  - Check for warnings or violations.

DEPLOYMENT ARCHITECTURE


LOGIN
ADMIN
check attendance
event conformation
OD Approval
STUDENT
Attendance
Success
Failure
More than 75%
Less than 75%
OD
High percentage
Low percentage
OD approved
OD rejected