

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

**(AN AUTONOMOUS INSTITUTION,
AFFILIATED TO ANNA UNIVERSITY)**

Rajiv Gandhi Salai (OMR), Kalavakkam - 603 110.

LABORATORY RECORD

NAME : SARIIKA ESACK IMUTHU, PADMA POOJA, C., SHRI KESHAVINEER
Reg. No. : 2015001094, 201501072, 201501101.....
Dept. : CSE..... Sem. : VI..... Sec. : B.....

ssn

**SRI SIVASUBRAMANIYA NADAR
COLLEGE OF ENGINEERING, CHENNAI**
(AN AUTONOMOUS INSTITUTION, AFFILIATED TO ANNA UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this is the bonafide record of the practical work done in the

U.C.S.I.6.I.7 - Mini Project Laboratory by

Name ..Saxika Esackimuthu, Padma Padja A.C, Shrikeshavineer

Register Number205001094, 205001072, 205001101.....

Semester ..VI.....

Branch ..Computer Science.....

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam.

During the Academic year ..2022-2023.....


Faculty


Head of the Department

Submitted for the.....Practical Examination held at SSNCE
on.....

Internal Examiner

External Examiner

INDEX

Name : SARIKA ESACKIMUTHU..... Reg. No. 20SD01094.....

Sem : VI..... Sec : B.....

Ex. No.	Date of Expt.	Title of the Experiment	Marks	Page No.	Signature of the Faculty	Remarks
1	9/2/23	Identification of Problem statement		1	Rony	
2	9/2/23	Software requirements Specification		4	Rony	
3	15/2/23	Identify use cases and develop use case model		15	Rony	
4	26/2/23	UML class diagram and domain Model		21	Rony	
5	10/3/23	sequence and collaboration diagram		30	Rony	
6	19/3/23	Activity diagram		39	Rony	
7	26/3/23	State Machine Diagram		45	Rony	
8	1/4/23	Component, Deployment and package diagrams		49	Rony	
9	5/4/23	Implementation of UI and domain layers		54	Rony	

INDEX

Name : SARIKALESHACKIMUTHU Reg. No. 201001094

Sem : VI Sec : B

**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 1

IDENTIFICATION OF PROBLEM STATEMENT

Students must come up with a problem statement which will address the needs of the user.
[CO1, K2 & K3]

Identify an activity from your day-to-day life that needs to be automated. You may select any activity for which a software solution will provide an efficient and effective alternative. You may follow the steps given below to identify the problem and prepare a brief problem statement for the same.

- Identify the common issues in daily life.
- Think about designing a software application for one of these issues, that will automate its workflow.
- Identify the scope and users of the application.
- Analyze the requirements of the customers (users).
- Write a brief problem statement.

Sample List of Projects

1. Result analysis
2. PROS – Permission and Request Online System
3. EMS - Electoral Management System
4. EAS – Elective (PE/OE) Allotment System
5. HMS – Hostel Management System
6. Resource Management System
7. TMS-Transport Management System
8. Professional Elective
9. Tours and Travels
10. System for Co-Curricular Attendance Management (SCAM)
11. Choice Based Elective Selection System
12. Research Publications
13. Query site for college students
14. SSN Eats – Food Ordering System
15. Department Resource Management System
16. Lost and Found Management System
17. Hostel Management System
18. Funded Projects Tracking System
19. Result Analysis
20. Movie Ticket Booking System

Attendance Management System (AMS) Problem Statement

Ex No :

Date :

Attendance management is an essential task for colleges to monitor student attendance, performance, and discipline. Accurate and up-to-date attendance records help colleges to make informed decisions and ensure that students attend classes regularly. However, the traditional manual attendance management system followed by colleges has several problems.

First, the manual attendance management system is time-consuming and prone to errors, which can result in inaccurate attendance records. It is also difficult to store and maintain attendance records, especially for larger colleges. In addition, the manual attendance system requires a lot of paperwork, making it difficult to keep track of attendance records. Second, there is a lack of transparency, as attendance data is not readily accessible to all relevant parties. This can result in students missing classes, poor performance, and low academic progress. In addition, it is difficult for the faculty to monitor and analyze the attendance data of students, leading to delays in decision-making. Third, the traditional attendance management system does not provide real-time data, which means that the attendance data is not updated in real time. This makes it difficult for colleges to track student attendance, especially when students skip classes.

To overcome the aforementioned problems, the proposed automated attendance management system (AMS) aims to eliminate manual data entry, provide accurate and up-to-date attendance records, and make it easier to maintain attendance data. In addition, it can provide transparency and accountability, enabling the faculty to track student attendance patterns and make informed decisions. The admin can register students and teachers and generate login credentials. Essentially, for a professor or a student to enter the web app, they have to sign in using their credentials which will be validated and access will be denied in the case of incorrect credentials.

The professor can mark attendance which will update the overall attendance record. The student can view the attendance course-wise, check leave history and even raise a ticket claiming attendance if the professor has not marked attendance. Moreover, a warning mail will be sent to the student if the attendance percentage goes below the 75% mark. Furthermore, a student can apply for OD by filling and uploading the necessary documents as proof and if the professor accepts the request, attendance will be automatically updated.

Therefore, the problem statement for an attendance management system for colleges is to create an automated system that is accurate, up-to-date, and easy to use. The system should eliminate paperwork, provide real-time attendance data, improve transparency, and enhance accountability. Thus, the AMS aims to relax the burden of attendance maintenance by making the process user-friendly for both the students and the professors.

**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 2

SOFTWARE REQUIREMENTS SPECIFICATION

Prepare a Software Requirements Specification (SRS) document to capture detailed requirements of the system proposed. [CO1, K2, K3 & K4]

The SRS should include the following.

- Abstract (Problem Statement Description)
- Introduction
- Purpose
- Scope
- Intended Audience
- Overall Description
 - Users and Characteristics (Actors)
 - Functional Requirements
 - Sub Functionalities
- Requirements Specification
 - User Interfaces
 - Non- Functional Requirements
- System Flow Diagram.
- Operating Environment
- Hardware Requirements
- Technologies.
- Design and Implementation Constraints
- Performance Requirements
- Security Requirements
- Software Quality Attributes

Ex No :

Date :

Attendance Management System

Software Requirements Specification Document

The Attendance Management System (AMS) for Colleges is designed to help colleges and universities manage student attendance more efficiently. The AMS will provide a user-friendly interface for instructors and administrators to manage student attendance and automate the manual process of attendance tracking. This system will be an essential tool for colleges and universities to manage student attendance and provide valuable insights for analysis.

Contents

1 INTRODUCTION.....	3
1.1 DOCUMENT PURPOSE.....	3
1.2 PRODUCT SCOPE.....	3
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	4
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	4
1.5 DOCUMENT CONVENTIONS.....	4
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	5
2 OVERALL DESCRIPTION.....	5
2.1 PRODUCT PERSPECTIVE.....	5
2.2 PRODUCT FUNCTIONALITY.....	5
2.3 USERS AND CHARACTERISTICS.....	5
2.4 OPERATING ENVIRONMENT.....	6
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	6
2.6 USER DOCUMENTATION.....	6
2.7 ASSUMPTIONS AND DEPENDENCIES.....	6
3 SPECIFIC REQUIREMENTS.....	6
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	6
3.2 FUNCTIONAL REQUIREMENTS.....	7
4 NON-FUNCTIONAL REQUIREMENTS.....	8
4.1 PERFORMANCE REQUIREMENTS.....	8
4.2 SAFETY AND SECURITY REQUIREMENTS.....	8
4.3 SOFTWARE QUALITY ATTRIBUTES.....	9

1. INTRODUCTION

1.1 DOCUMENT PURPOSE

The purpose of a Software Requirements Specification (SRS) for an attendance management system is to clearly define the functional and non-functional requirements of the software, as well as any constraints, assumptions, and dependencies that must be considered during the design, development, and testing of the system. This document serves as a contract between the college and the development team, providing a detailed description of what the software is expected to do, how it will perform, and what it will look like. The specific purposes of an SRS for an attendance management system in a college include:

1. Describing the System Requirements: The SRS document provides a comprehensive description of the system requirements, including the functional requirements (features, capabilities, user interfaces) such as the ability to record attendance for multiple courses, class sections, and instructors.
2. Facilitating Compliance with Attendance Policies: The SRS document outlines how the attendance management system will facilitate compliance with college attendance policies, such as the ability to track student attendance, monitor absences, and generate reports.
3. Supporting Communication with Stakeholders: The SRS document describes how the attendance management system will support communication with various stakeholders, such as instructors, students, and administrators, through features such as email notifications, messaging, and reporting.
4. Supporting Campus-wide Deployment: The SRS document describes how the attendance management system will be deployed across the entire college campus, including hardware and software requirements, as well as any network or security considerations.
5. Supporting Integration with Existing Systems: The SRS document also describes how the attendance management system will integrate the LMS of SSN.

Overall, the purpose of this document is to ensure that the software meets the needs of the college, is of high quality, and is specifically addressing the unique requirements of a college environment.

1.2 PRODUCT SCOPE

The Software Requirements Specification captures all the requirements in a single document. The Attendance Management software aims at helping the user to address issues from multi-disciplinary angles related to Attendance management and services. This software helps to organize events without any paperwork and has a wide variety of modules. The Attendance Management System is supposed to have the following features :

1. Login function: The application provides any user with three types of login. One for the administrator, one for the instructor and one for the student. Any user needs to login, in order to avail the features of the application.
2. View Attendance: The system provides the student to view attendance marked by instructors of multiple courses over time.
3. Raise ticket: The student is provided with an option called raise ticket. If the instructor has wrongly marked the attendance record, the student may raise a query related to the above.

4. Apply for OD: The system provides the student to apply for OD. Whenever the student takes leave in connection with extracurricular activities, he/she may request attendance if his absence is considered valid.
5. Mark Attendance: The system provides the instructor with a list of courses he handles and a list of students in each course. The instructor is responsible for marking the attendance of each student for the entire duration of his course.
6. Approve ticket: The instructor is given the option to approve or reject the ticket raised by the student based on reasonable grounds.
7. Approve OD: This system gives the instructor the option to approve the OD received from the students after carefully checking the documents attached to the OD. The instructor changes the marked attendance once the documents are verified.
8. Register users: The system provides the administrator to register new instructors and students. He is responsible for assigning the courses to instructors and taking care of the entire system.
9. Set Attendance policies: The system provides the administrator to set, alter or remove the attendance policies.
10. Mailing facility: End users of this system will receive an email if attendance goes below the minimum mark.

1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW

The intended audience of a Software Requirements Specification (SRS) for an attendance management system typically includes the development team, project stakeholders, and other interested parties, such as system administrators, end-users, and quality assurance personnel. The document provides a detailed description of the system's features, functions, and capabilities, as well as the technical, operational, and security requirements that must be met.

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

- SSN - Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam
- LMS - Learning Management System
- AMS - Attendance Management System
- SRS - Software Requirement Specification
- OD - On Duty
- Administrator: The in-charge who is responsible for setting attendance policies, adding new instructors and students, and handling attendance reports.
- Instructor: The in-charge who is responsible for marking attendance of the students, handling queries raised by the students and verifying the OD applications received from the students.
- Student: The end user who serves as a client views the attendance marked by the instructor, raises a query against marked attendance and applies for OD.
- Server: Machine that stores all the information and records.

1.5 DOCUMENT CONVENTIONS

The entire document is in Times New Roman font. The headings are numbered 1,2,3... and so on and sub-headings are numbered x.1,x.2.... and so on. Both headings and subheadings are in bold. Main title: Font Times New Roman and size 14

Subtitles: Font Times New Roman and size 14

Content: Font Times New Roman and size 12

1.6 REFERENCES AND ACKNOWLEDGMENTS

Software Engineering book written by Roger Pressman, Ian Sommerville.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The AMS is a web application to be used by the college management to improve the efficiency of management, instructors, and students. The AMS to be developed benefits greatly the instructors and students of the institution. The web application provides facilities to instructors to mark attendance of students during class & reduce manual work. It is used to track student's attendance, absentee record, attendance history & other related documents. It is aimed at replacing the tedious paper works that colleges currently use. The system will collect data and store it for fast and easy reference. The system will provide users with a complete record of their attendance. It is thus helpful in reducing the time and complexity of maintaining the records.

2.2 PRODUCT FUNCTIONALITY

The AMS allows the instructor to record, store, and monitor students' attendance history & manage the classroom efficiently. The Product functions are more or less same as described in the product perspective. The functions of the web app includes the system providing different types of services based on the types of users [Administrator/ Instructor/ Student].

- The student should be provided with a login page through which they can login into their respective accounts.
- The student should be provided with the attendance report for different courses over the duration of the same.
- Provisions for the students to raise a query if attendance is wrongly marked and apply for OD.
- The instructors will be able to mark and view attendance of the students who are enrolled in their courses.
- The instructors are provided with the interfaces to approve the raised tickets and OD and alter the marked attendance.
- The administrator can generate login credentials for instructors and students.
- The admin can also change the attendance policies (minimum attendance mark etc)

2.3 USERS AND CHARACTERISTICS

The users of the system are the instructors, students and administrators who maintain the system.

The instructors and students are assumed to have basic knowledge of computers and internet browsing. The administrators of the system are assumed to have more knowledge of the internals of the system and are able to rectify the minor problems that may arise due to high traffic, disk crashing and other catastrophes to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient on how to guide the users on how to use the web application without any problems.

2.4 OPERATING ENVIRONMENT

The Attendance Management System can run on desktop operating systems such as Windows and Linux. The system uses MySQL as a database server. The other software components include SpringBoot, ReactJS and Figma.

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

- The attendance management system will be designed to work with the existing infrastructure of the college, including hardware, software, and network. The institution's system must be compatible with the internet applications.
- The system will be scalable to accommodate the growing number of students and instructors.
- The system will also be secure and protect the privacy of student information. The information of all users must be stored in a database that is accessible by the AMS.
- The web application is connected to the institution's server and is running all 24 hours a day.
- The users access the web application from any computer that has internet connection and browsing capabilities.
- The users must have their correct usernames and passwords to enter into their corresponding accounts.

2.6 USER DOCUMENTATION

The system comes with a very easy user interface. It provides a printed user manual in pdf format. It also includes a forum where users can discuss and clarify their doubts regarding the issues faced by them while using the application.

2.7 ASSUMPTIONS AND DEPENDENCIES

- The users have sufficient knowledge of computers.
- The users have basic English knowledge, as the interface will be provided in English.
- The college's network infrastructure is sufficient to support the attendance management system with required network bandwidth, server capacity, and network security.
- The institution's server loads all the end users accessing the web application at the same time.
- The product can access the institution's student database.
- Assuming that all the information entered by the user will be correct. If any wrong information is found then the system will notify an alert.
- The system is required to save generated reports.
- The system is dependent on compliance with all applicable regulations and policies.

3. SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

The User Interface Screens are described in table 1.

Table 1: Attendance Management User Interface Screens

Screen Name	Description
Register	Create user ID (Student and Professor individually)
Login	Log into the system.
Professor	Mark attendance, approve/reject OD request, approve/reject ticket for attendance claim.
Student	Display attendance, raise ticket claiming attendance.
Leave records	Display to leave history.
Apply for OD	Application for OD with relevant documents as proof.
Approve/reject OD	If the Professor approves the OD request, records get updated automatically. If the professor rejects it, the records remain unchanged.

3.1.2 Hardware Interfaces

The system shall run on :

Operating system: Any OS.

Scripts which support CGI, HTML & Javascript.

Web Browser : Google Chrome , Mozilla firefox.

3.1.3 Software Interfaces

The system shall interface with MySql database.

To implement the project we have chosen React for frontend , Spring Boot for backend and Figma for UI for easy interaction.

3.2 Functional Requirements

- User Management: The system should allow the admin to create and manage user accounts for faculty, students, and staff. The admin should be able to assign roles and permissions to each user.
- Attendance Recording: The system should allow faculty to mark attendance for each student in their class. The faculty should be able to mark attendance by marking present/absent.
- Attendance Reporting: The system should allow faculty or admin to generate attendance reports for each student or the entire class. The reports should show the total number of classes held, the total number of classes attended, and the percentage of attendance for each student.

- Notifications: The system should send notifications to faculty and students regarding attendance, leaves and warning if attendance goes below 75%.
- Analytics and Insights: The system should provide analytics and insights on attendance patterns, such as the percentage of attendance for each course, the percentage of students with high or low attendance, and other relevant metrics.
- Integration with Other Systems: The system should be able to integrate with other systems like student information systems and learning management systems to provide a seamless experience for faculty and students.
- OD Application: The system should allow students to apply for OD with necessary documents and get approval from faculty. Faculty should be able to view the application and verify documents.

4. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

4.1 PERFORMANCE REQUIREMENTS

Performance requirements define acceptable response times for system functionality.

- Accuracy: The system should accurately record attendance data to ensure that the attendance records of students are reliable and trustworthy.
- Timeliness: The system should be able to capture attendance data in real-time or as close to real-time as possible to enable faculty and staff to make informed decisions based on the most up-to-date information.
- Scalability: The system should be able to handle a large volume of data and users, as colleges can have many students, faculty members, and staff who will use the system concurrently.
- Accessibility: The system should be accessible to all users, including those with disabilities or who may need special accommodations to access the system.
- Security: The system should be secure and protect the privacy of student data, preventing unauthorized access to attendance records.
- User-friendliness: The system should be easy to use and intuitive, allowing faculty, staff, and students to quickly navigate and use the system without extensive training or support.
- Integration: The system should integrate with other systems used by the college, such as the student information system, to streamline the attendance tracking process.
- Reporting: The system should generate reports that are useful for faculty, staff, and administrators to monitor attendance trends and identify areas for improvement.

4.2 SECURITY REQUIREMENTS

- Access Control: The system will have proper access controls, including password protection and user role-based permissions, to ensure that only authorized personnel can access attendance data.
- Data Encryption: Attendance data will be encrypted when in transit and at rest to protect it from

unauthorized access.

- Data Backup and Recovery: The system will have a backup and recovery mechanism to prevent the loss of data in case of system failure, disaster, or cyber-attacks.
- Audit Trails: The system will maintain audit trails to track any modifications to attendance data to detect and prevent any fraudulent activities.
- User Authentication: The system will have a robust user authentication mechanism to ensure that users are who they claim to be.
- System Monitoring: The system will have a continuous monitoring mechanism to detect and prevent any unauthorized access attempts, suspicious activities, or security breaches.
- System Updates and Patches: The system will be updated regularly with the latest security patches to prevent any vulnerabilities that may be exploited by attackers.

4.3 SOFTWARE QUALITY ATTRIBUTES

4.3.1 Standards Compliance

- Security: The system will be designed and developed in a way that protects user data from unauthorized access or breaches.
- Privacy: The system will be designed and developed in a way that collects and processes personal data in a transparent and lawful manner.
- Usability: The system will be designed and developed in a way that makes it easy for users to use and understand.
- Reliability: The system will be designed and developed in a way that ensures it performs consistently and without errors.
- Performance: The system will be designed and developed in a way that ensures it can handle the expected number of users and transactions.
- Maintainability: The system will be designed and developed in a way that makes it easy for developers to maintain and update it.

4.3.2 Reliability

- Accuracy: The system will accurately record and track attendance for each student, ensuring that the data is reliable and can be used for attendance reporting and analysis.
- Availability: The system will be available and accessible to users whenever they need to use it, without any downtime or service disruptions that could impact attendance tracking.
- Recovery: In the event of a system failure or other issue, the system will be able to recover quickly and without data loss, ensuring that attendance data is preserved and can be used for reporting and analysis.
- Consistency: The system will provide consistent and reliable results over time, regardless of changes in workload or usage patterns.
- Scalability: The system will be able to handle an increasing number of users and data volumes without compromising its reliability or performance.

4.3.3 Availability

The system will be constantly monitored for issues that could impact availability, such as system performance or server availability. This can help ensure that any issues are detected and resolved quickly, minimizing the impact on users. The system will have a disaster recovery plan in place to

ensure that critical data can be recovered in the event of a system failure or outage. This should include regular backups of system data and the ability to restore that data quickly and efficiently.

4.3.4 Maintainability

Maintainability is one of the important software quality attributes that refers to the ease with which software can be modified, enhanced, or adapted.

- **Modularity:** The system will be designed in a modular way so that it is easy to modify, enhance or replace a particular module without affecting the other modules. This will make it easier to maintain the system in the long run.
- **Code readability:** The code will be well-organized, well-documented, and easy to understand. This will make it easier for developers to maintain and modify the code when required.
- **Flexibility:** The system will be flexible enough to accommodate changes and updates as per the evolving requirements of the college. It should be easy to add new features, modify existing ones, and integrate with other systems.
- **Testability:** The system will be designed in a way that it can be easily tested for errors and bugs. This will make it easier to locate and fix any issues in the system, and ensure that the system continues to function smoothly.
- **Scalability:** The system will be scalable to accommodate changes in the size of the college, such as an increase in the number of students or courses. This will ensure that the system can continue to handle the growing needs of the college.
- **Maintainability tools:** The system will be developed using tools that support maintainability, such as version control systems, automated testing tools, and code analysis tools. These tools can help developers to identify and fix issues quickly, and ensure that the system is well-maintained.

4.3.5 Portability

The Attendance Management System shall run in any web browser that contains Java Runtime and the MySQL Access Database.

Ex.No.3

Identify Use Cases and develop the Use Case model.

Aim:Creation of an UML Use case model for a problem domain.

Prerequisites:

some basic knowledge about business domain

some basic understanding of UML Use case models

Input: Problem Statement

Output :

1. Use case template
2. **Use case Modeling** using UML use case Diagram

Use case diagram must include the following elements:

1. Actor
2. Use case (process)
3. System boundary
4. Associations and relationship

<<include>>
<<extends>>
generalization

The following should be the flow of the assignment to be submitted

Aim

Notations

// Draw and Write the definition of all the Usecase diagram notation

Identification of Actors

// List the actors in your problem statement

Identification of Scenarios

// List all success and failure scenarios and the subfunctions in your problem statement.

Remember: You will draw Usecase diagrams for all these listed scenarios and subfunctions

Relating Usecases

Define Generalization, Association (include, extend, uses) with their respective notations

Draw Usecase diagrams for all the scenarios and subfunction identified above

Write the fully dressed Usecase description for

- a. Main success scenario
- b. Frequent Alternate scenario
- c. One important subfunction

Documentation

// You will document your observation on the Use case diagram of your system briefly in a paragraph

Use Case Description

Ex No :

Date :

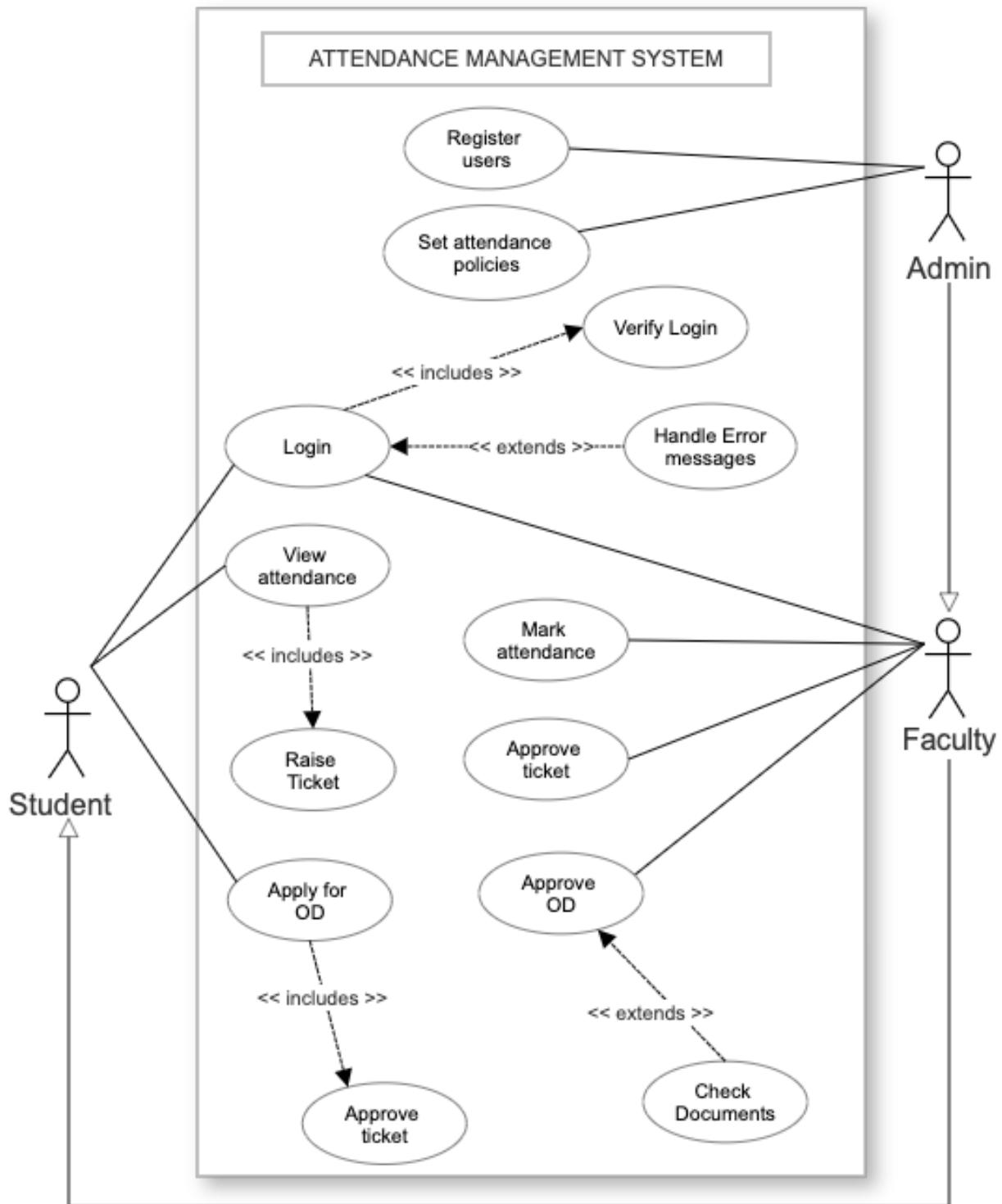
Aim:

To describe the use cases and draw the use case diagram for Attendance Management System.

Notations :

Diagram	Notation	Explanation
	Actor	Represents the various types of users of the system
	Relationship	Represents “includes” relationship between use cases
	Relationship	Represents “extends” relationship between use cases
	Relationship	Represents generalization between actors and between use cases
	Relationship	Represents association between use cases

Use case diagram:



Scope: Attendance Management System

Primary Actors: Faculties, Students

Stakeholders and Interests:

- **ADMIN** - wants a user-friendly application that can maintain attendance records of students. Admin can create login credentials for the users and view the attendance at any time. Admin can also change attendance policies like setting the minimum attendance benchmark.
- **STUDENT** - wants to view their attendance for each course. Students can also raise queries claiming attendance. Moreover, students can apply for OD by submitting the required documents.
- **INSTRUCTOR** - wants an automated application that manages the attendance records. Instructors can mark attendance, view attendance and accept/reject OD applications.

Main Success Scenario:

1. Admin generates the login credentials for instructors and students.
2. Admin sets attendance policies.
3. Instructor opens the AMS application.
4. Instructor tries to log into the application using his/her credentials.
5. Username/Email is validated with the password and access is granted.
6. Instructor marks the attendance and gets the consolidated record.
7. Students open the AMS application.
8. Student tries to log into the application using his/her credentials.
9. Username/Email is validated with the password and access is granted.
10. Student views his/her attendance record/leave history.
11. Student raises a ticket claiming attendance.
12. Instructor accepts the claim and the record gets updated.
13. Students fill out the application for OD.

14. Instructor accepts the application and the record gets updated.

Special Requirements:

1. Internet connectivity
2. System should be capable of handling a user load of around 100 students.
3. Robust recovery in case of any failure.

Ex.No. 4

UML Class Diagram

Aim:Creation of UML class diagram for problem domain.

Prerequisites :

some basic understanding of UML class models

Input: Problem Statement

Output : UML class Diagram

Apply the following concepts in class diagram

1. Singleton class
2. constraints
3. Active class
4. Association class
5. Template class
6. Keywords
7. Abstract and Final class
8. Stereotypes
9. Notes
10. Attribute test
11. Association Lines
12. Dependency
13. Interfaces
14. Composition and Aggregation
15. Qualified Association
16. User Defined Compartments

Domain Model and Class Diagram

Ex No :

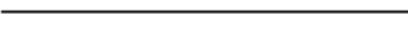
Date :

Aim:

To identify the classes and their associations and draw domain model and class diagram for Attendance Management System.

Notations for domain model :

Notation	Type	Explanation		
<table border="1"><tr><td>Classname</td></tr><tr><td>+ field: type</td></tr></table>	Classname	+ field: type	Conceptual Class	An object in the problem domain
Classname				
+ field: type				
<hr/> <hr/>	Association	Relationship between objects or the classes		
1..* 1	Multiplicity of the Role	Defines how many instances of a class A can be associated with one instance of a class B		
→	Generalization	The child class inherits the characteristics of base class		

Notation	Type	Explanation			
<table border="1"> <tr><td>Classname</td></tr> <tr><td>+ field: type</td></tr> <tr><td>+ method(type): type</td></tr> </table>	Classname	+ field: type	+ method(type): type	Conceptual Class	An object in the problem domain along with its attributes
Classname					
+ field: type					
+ method(type): type					
	Aggregation	It is a relationship between two classes where one class is a part of another where one class is meaningful even without the aggregate			
	Composition	It is a relationship between two classes where one class is a part of another where one class is not meaningful without the other			
	Generalization	The child class inherits the characteristics of base class			
	Association	Relationship between objects or the classes			
	Multiplicity of the Role	Defines how many instances of a class A can be associated with one instance of a class B			

Identification of Classes:

Conceptual Class Category	Conceptual Class
Physical or Tangible object	Attendance record, Course
Specifications	Course Description
Roles of People	User: Students, Teachers, Admin
Events	Marking attendance, Applying for OD, Raising Ticket.

Identification of Noun Phrases:

Initial list of noun phrases:

- Student
- StudentDescription
- Admin
- AdminDescription
- Teacher
- TeacherDescription
- Attendance
- OD
- Subject
- Subject Name
- Subject Code
- Student ID
- Teacher ID
- Email
- ODProof
- Date
- Period

Final List of Noun Phrases:

- Student
- StudentDescription
- Admin
- AdminDescription
- Teacher
- TeacherDescription
- Attendance
- OD
- Subject

Identification of Association:

a. Identification of Association

Category	Association	Between
A is logically contained in B	PartOf	Student and Subject
A is logically contained in B	PartOf	Teacher and Subject
A manages B	Manages	Admin and Teacher
A manages B	Manages	Admin and Student
A is an event related to B	Marks	Attendance and Teacher
A is an event related to B	Views	Attendance and Student
A is an event related to B	Applies	OD and Student
A is an event related to B	Verifies	OD and Teacher
A is item description of B	Describes	StudentDescription and Student
A is item description of B	Describes	TeacherDescription and Teacher
A is logically contained in B	Consists	Subject and Attendance

A is an event related to B	Updates	OD and Attendance
----------------------------	---------	-------------------

b. Definition of Association

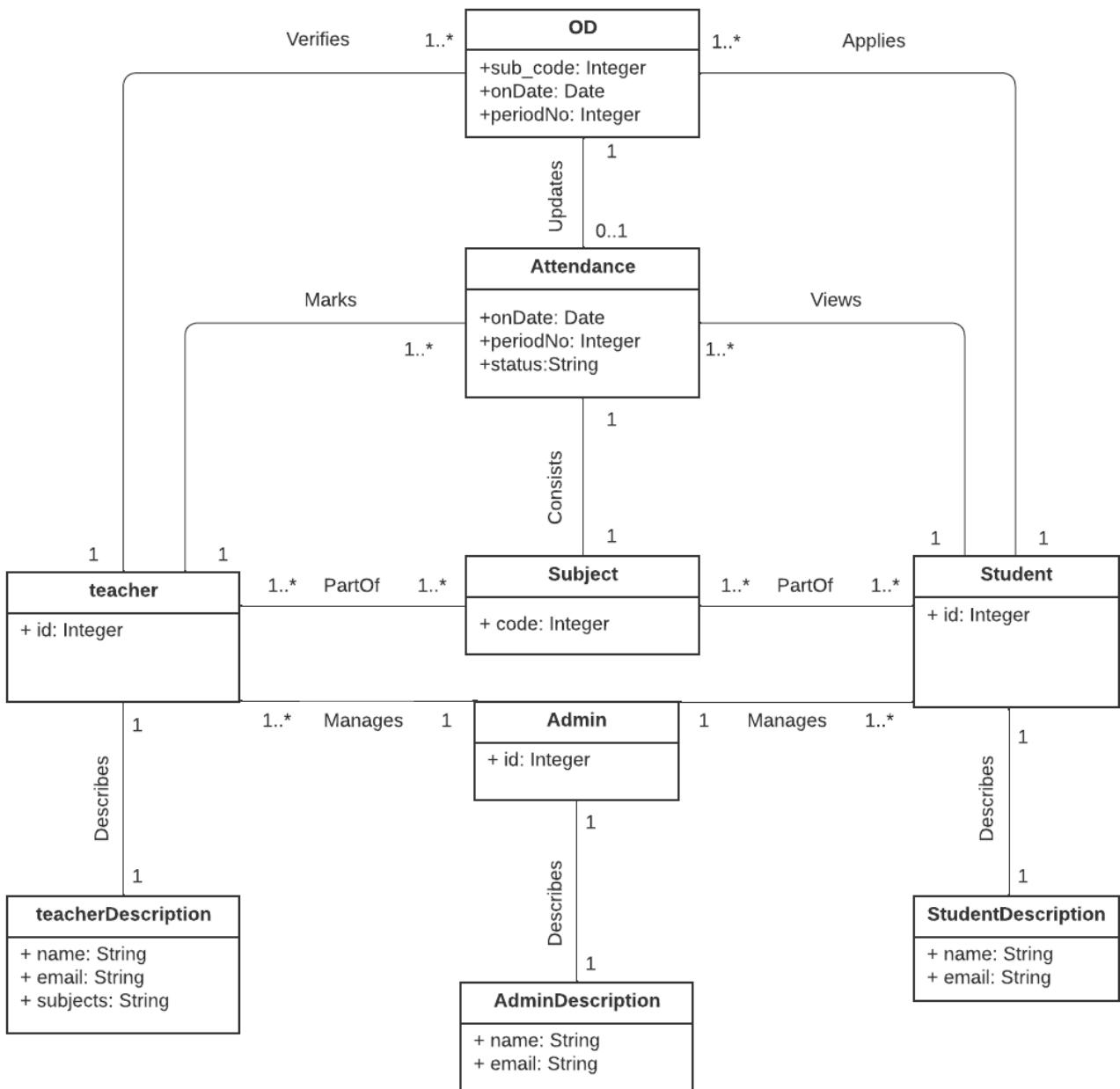
Association	Explanation
Generalization	The Generalization association ("is a") is the relationship between the base class that is named as "superclass" or "parent" and the specific class that is named as "subclass" or "child"
Aggregation	An aggregation is a relationship between classes which says one class 'is a part of' another class. In aggregation, the part (constituent) is meaningful without the whole (aggregate)
Composition	An aggregation is a relationship between classes which says one class 'is a part of' another class. In composition, the part (component) is not meaningful without the whole (container)

c. Multiplicity

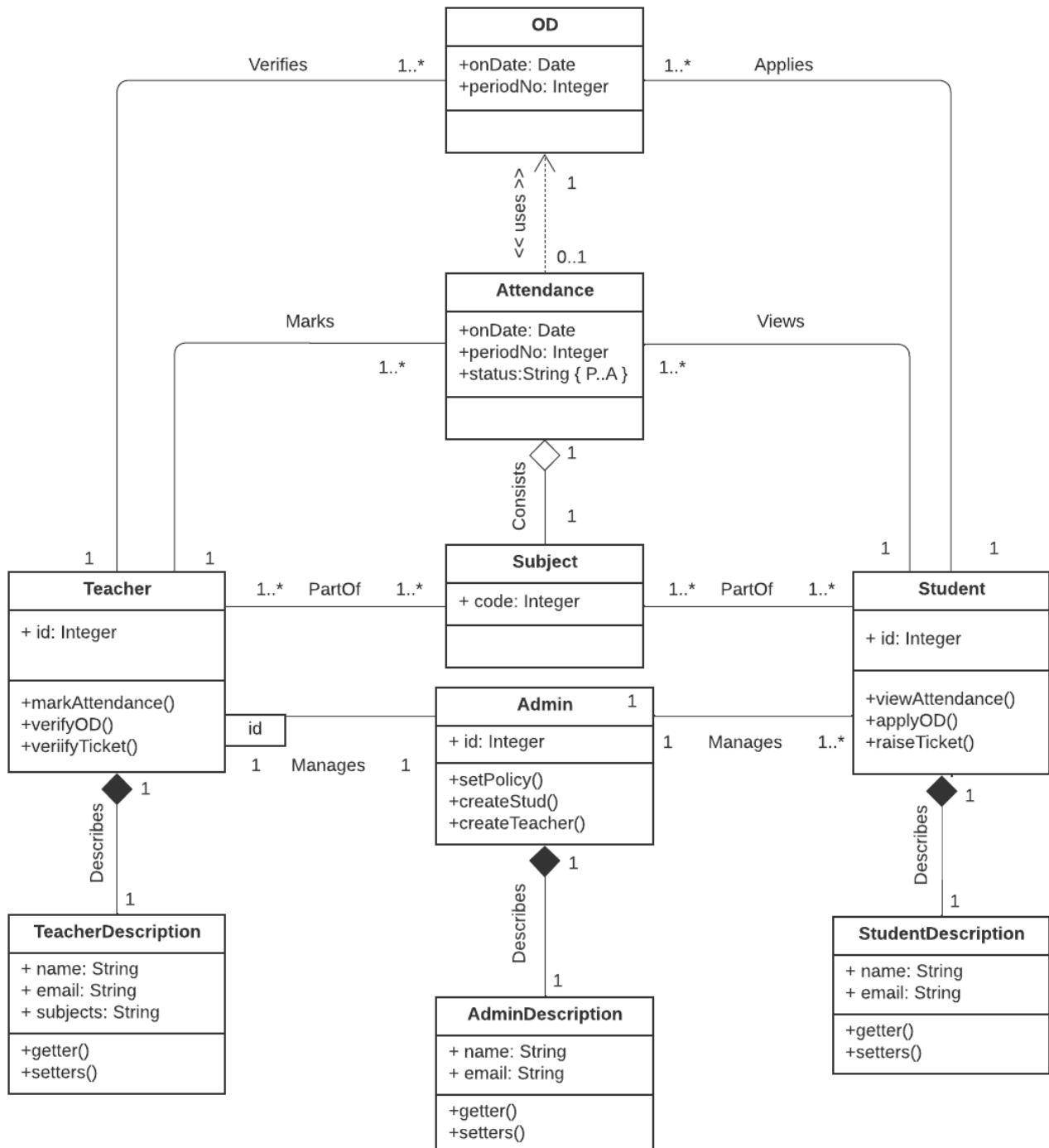
Relationship	Multiplicity
Student and Subject	One or more students (1..*) can be a part of one or more subjects (1..*)

Teacher and Subject	One teacher (1) can be a part of one or more subjects (1..*)
Admin and Teacher	One admin (1) can manage one or more teachers (1..*)
Admin and Student	One admin (1) can manage one or more students (1..*)
Attendance and Teacher	One teacher (1) can mark one or more attendances (1..*)
Attendance and Student	One student (1) can view one or more attendances (1..*)
OD and Student	One student (1) can apply for one or more ODs (1..*)
OD and Teacher	One teacher(1) can verify one or more ODs (1..*)
StudentDescription and Student	One student (1) can describe one student (1)
TeacherDescription and Teacher	One teacher (1) can describe one teacher (1)
Subject and Attendance	One subject (1) can consist one attendance (1)
OD and Attendance	One OD (1) may or may not update one attendance (0..1)

Domain Model Diagram:



Class Diagram:



**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 5

Interaction Diagrams

Design the interaction diagrams for the chosen application depicting the sequence of actions between the entities involved in the functionalities of the system proposed. [CO3, K2, K3 & K4]

The interaction diagrams include the following.

1. A Sequence diagram for the main success scenario in the use case diagram.
2. A Sequence diagram for each function (Use Case). (Number of sequence diagrams according to the main functions (use cases) you have in your proposed system).
3. Collaboration diagram for the main success scenario in the use case diagram.

Note:

Include the following notations in the sequence diagrams.

- Lifeline Boxes and Lifelines
- Lifeline Boxes and Lifelines
- Messages
- Focus of Control and Execution Specification Bars
- Illustrating Reply or Returns
- Messages to "self" or "this"
- Creation of Instances
- Object Lifelines and Object Destruction
- Diagram Frames in *UML* Sequence Diagrams
- Looping
- Conditional Messages
- Conditional Messages in *UML*
- Mutually Exclusive Conditional Messages
- Iteration Over a Collection
- Nesting of Frames
- Messages to Classes to Invoke Static (or Class) Methods
- Polymorphic Messages and Cases
- Asynchronous and Synchronous Calls

Include the following notations in the collaboration diagrams.

- Link
- Messages
- Messages to "self" or "this"
- Creation of instances

- Message number sequencing
- Conditional Messages
- Mutually Exclusive Conditional Messages
- Iteration or Looping
- Iteration Over a Collection
- Messages to Classes to Invoke Static (or Class) Methods
- Polymorphic Messages and Cases
- Asynchronous and Synchronous Calls

SEQUENCE AND COLLABORATION DIAGRAMS

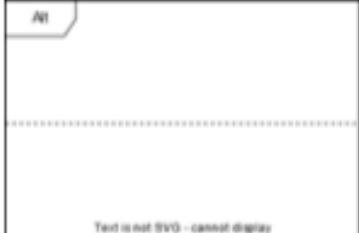
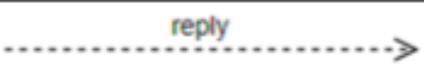
Ex No :

Date :

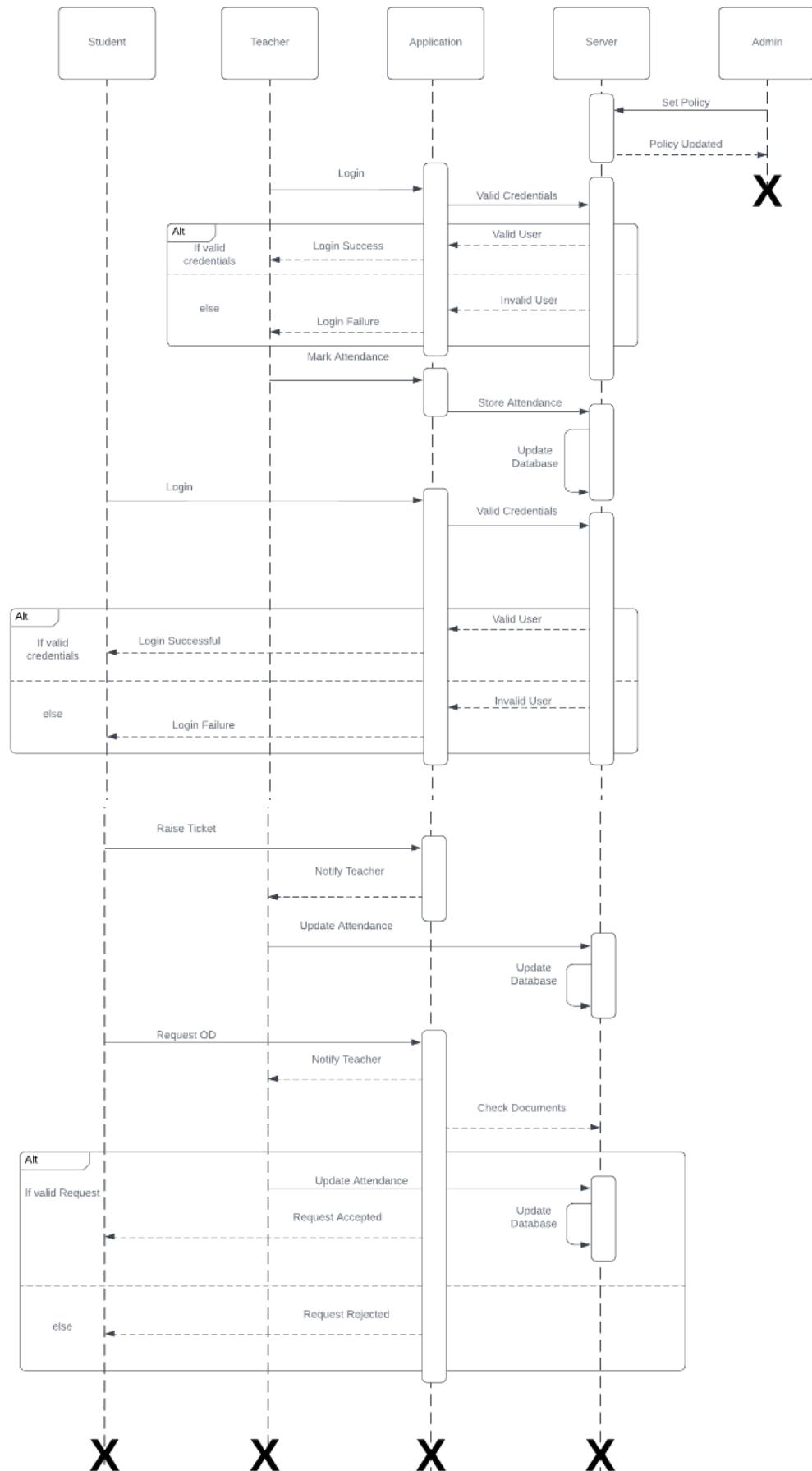
Aim:

To identify the objects and the sequences of interactions between the objects and to draw the sequence and collaboration diagrams.

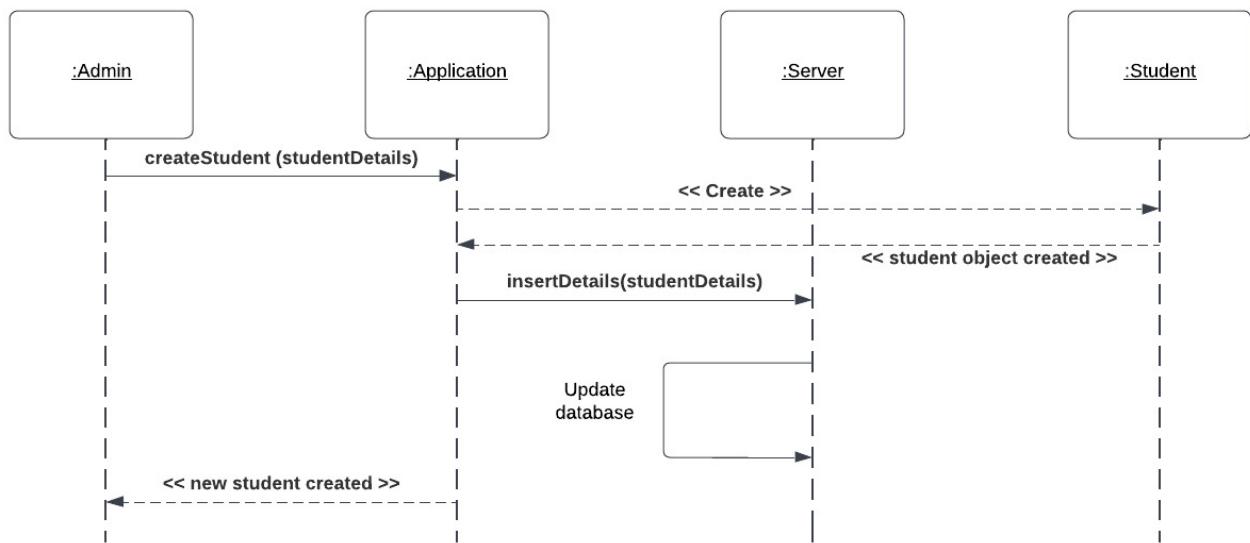
Notations for Sequence Diagram:

Notation	Type	Explanation
	Object	An object in the problem domain
	Activation frame	The time when an object is active during the interactions
	Alternate	The alternate options that an user can choose
	Reply	Reply from an object
	Request	Request to an object

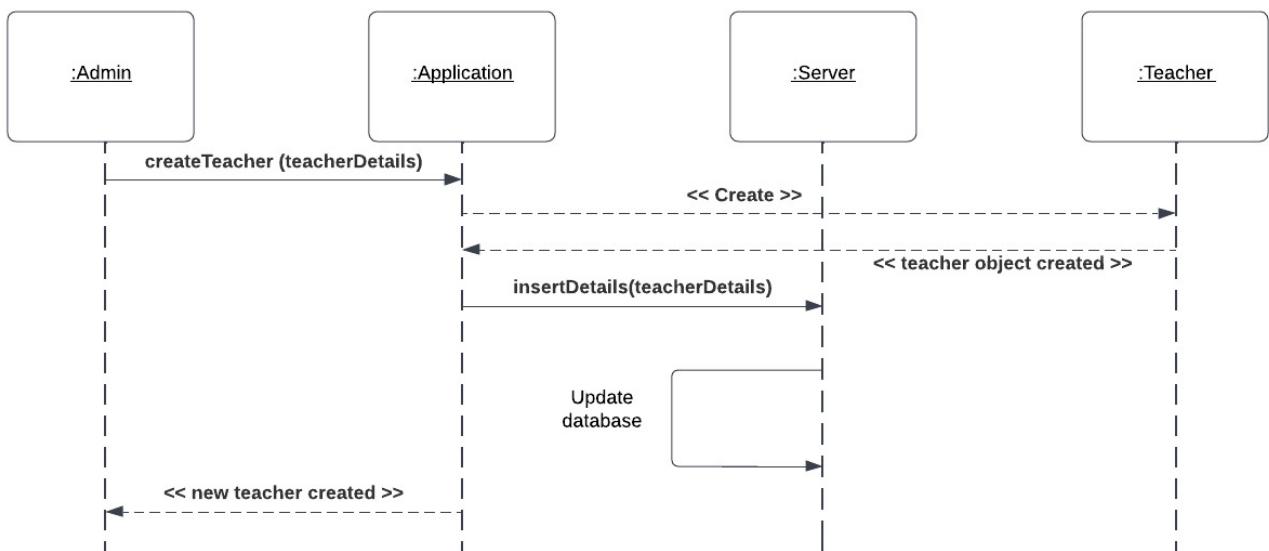
SEQUENCE DIAGRAM:



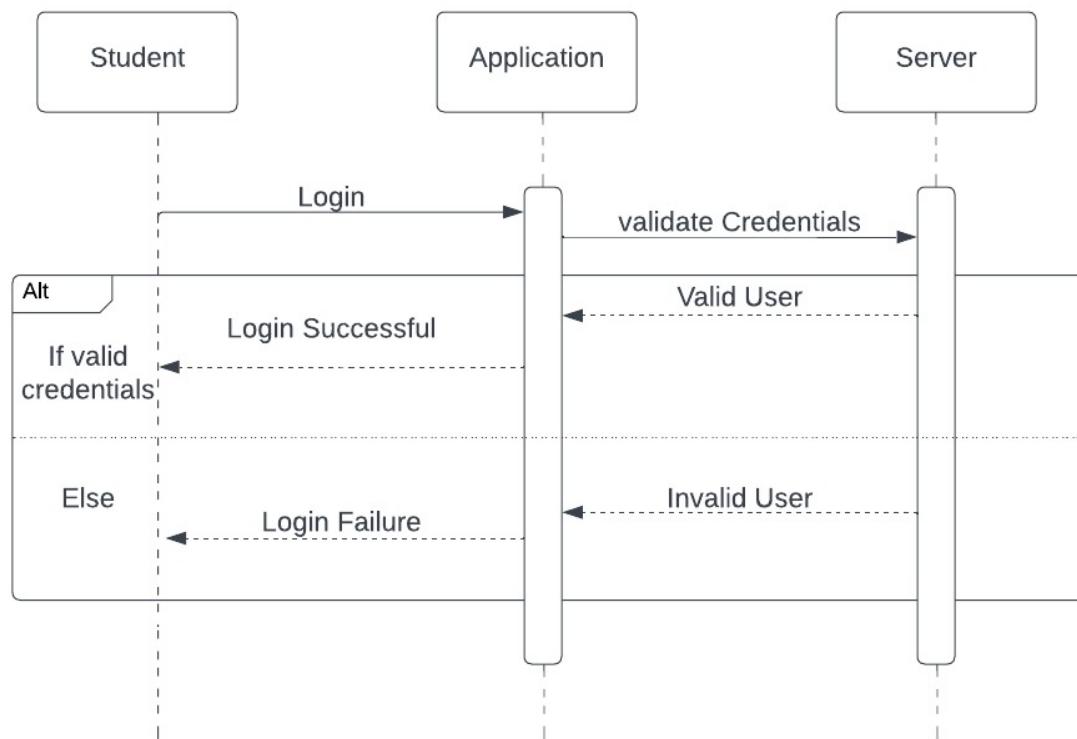
CreateStudent :



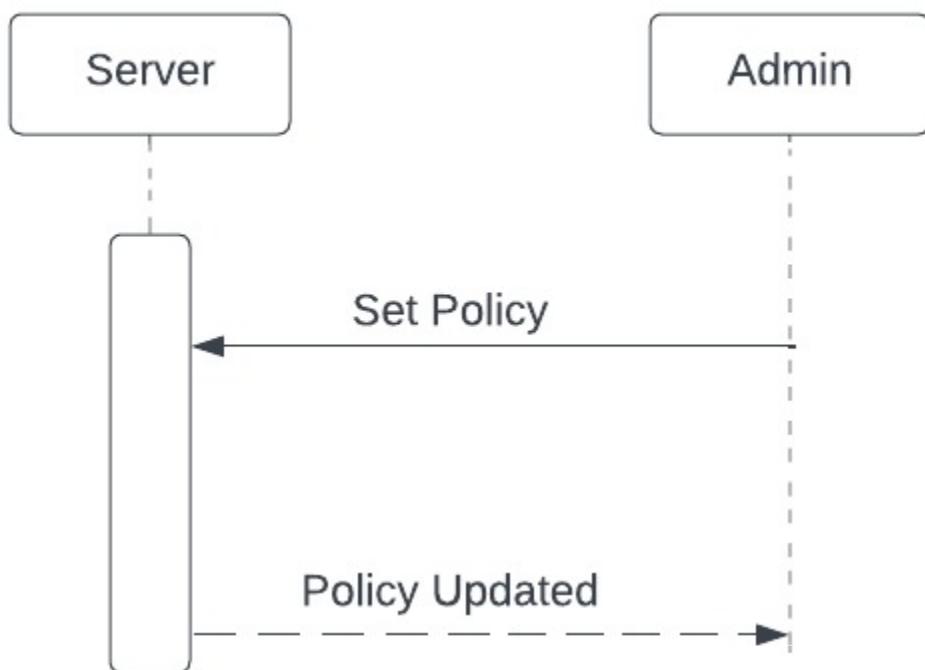
CreateTeacher :



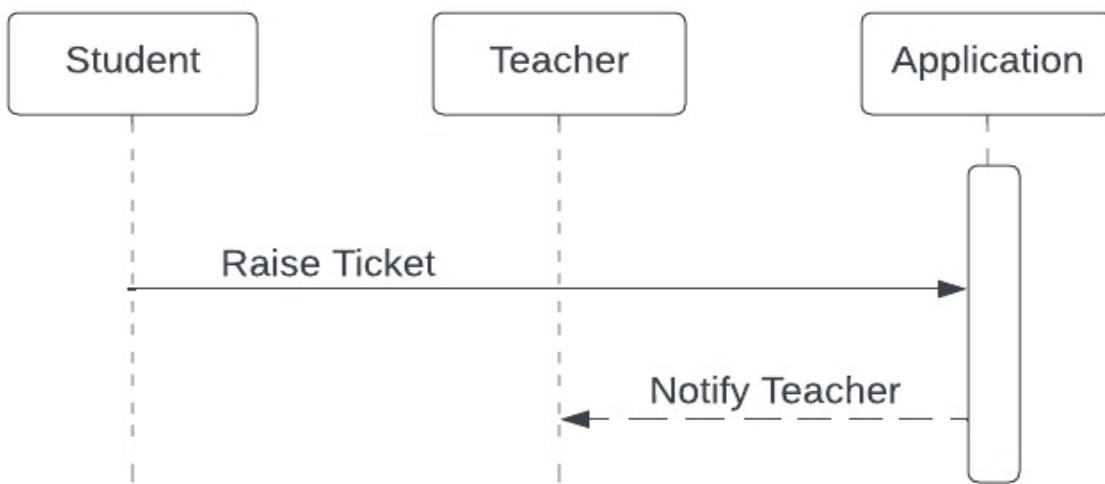
Login :



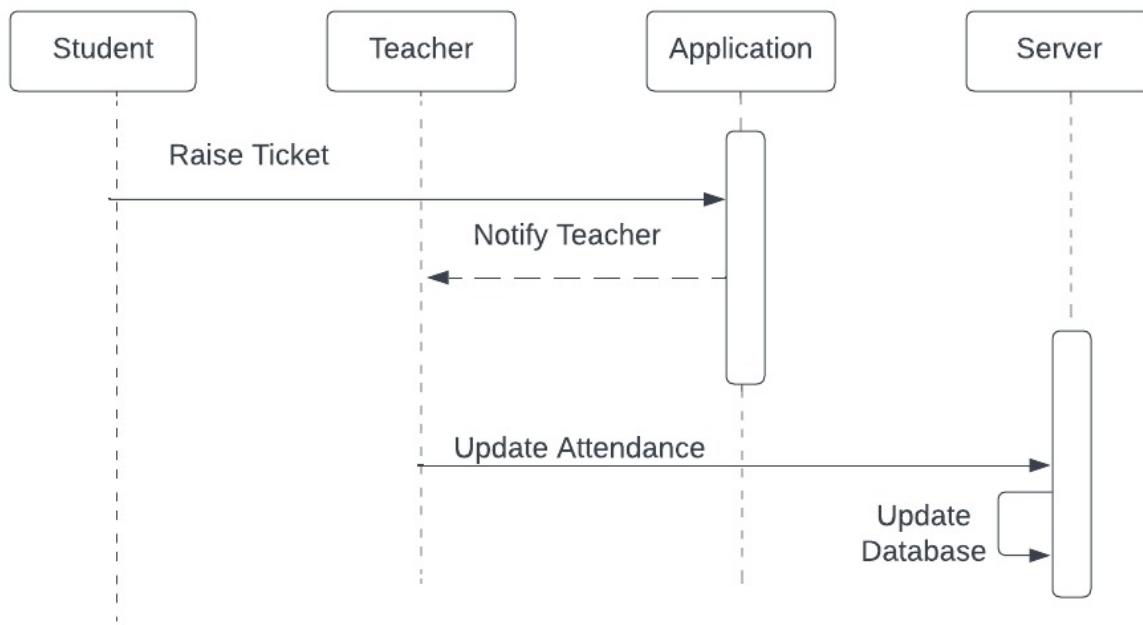
SetPolicy :



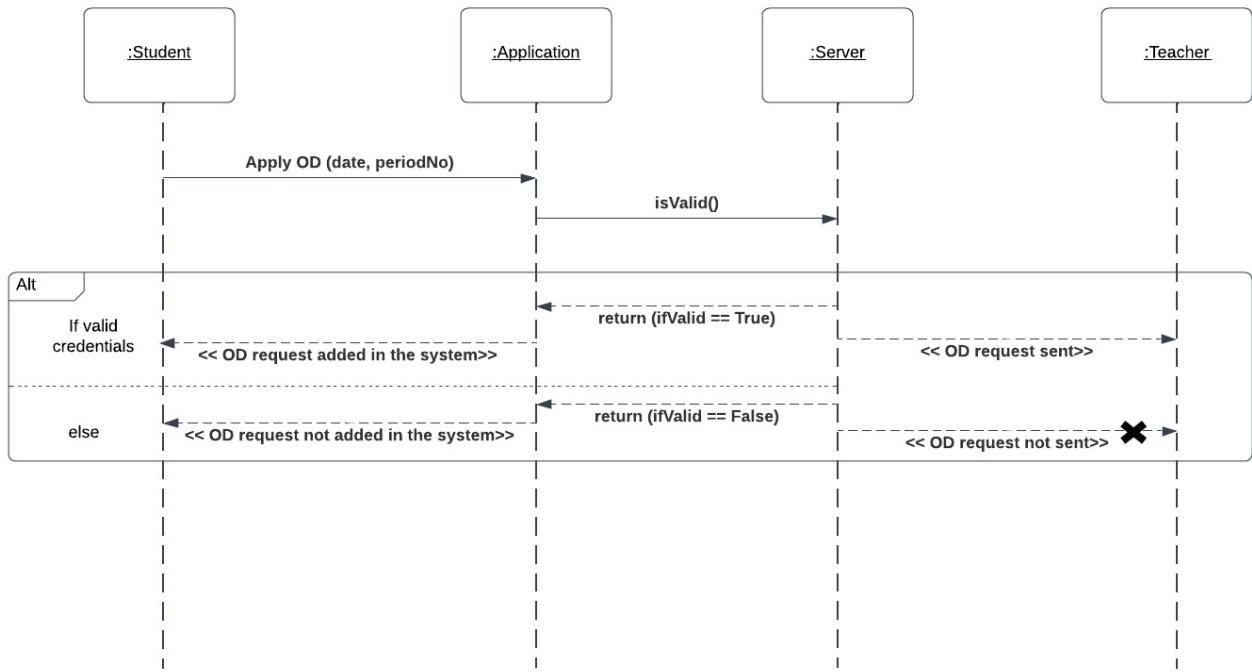
RaiseTicket :



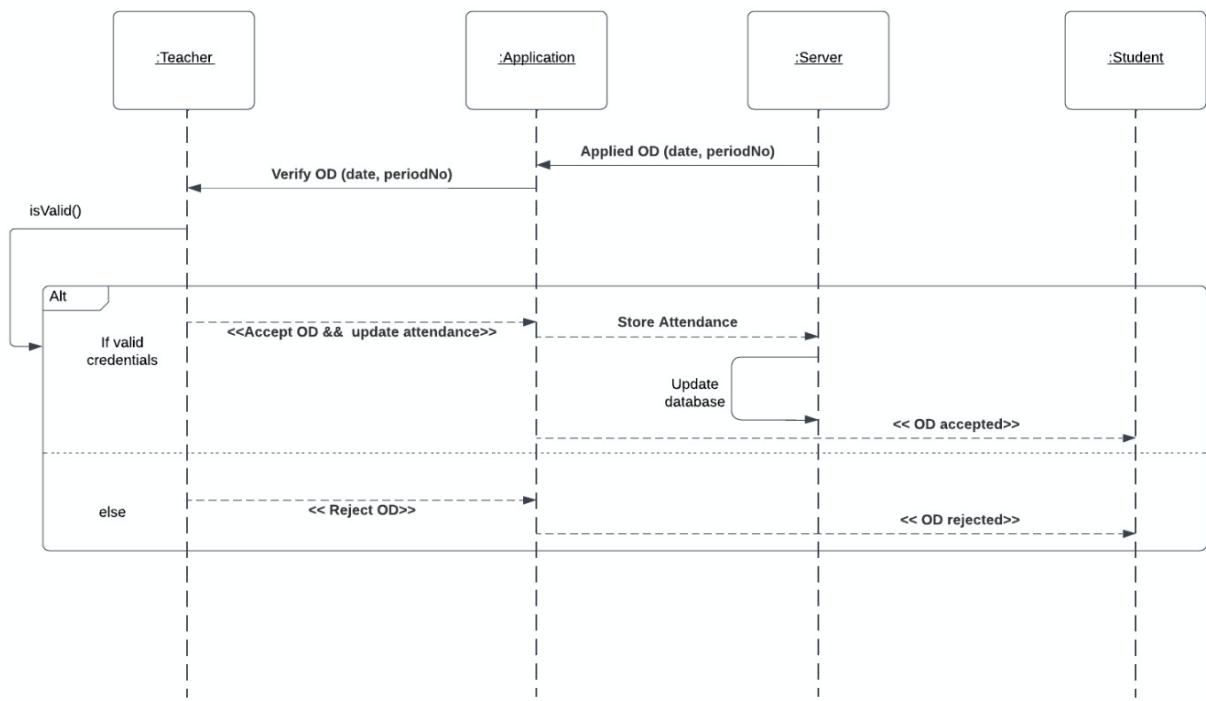
AcceptTicket :



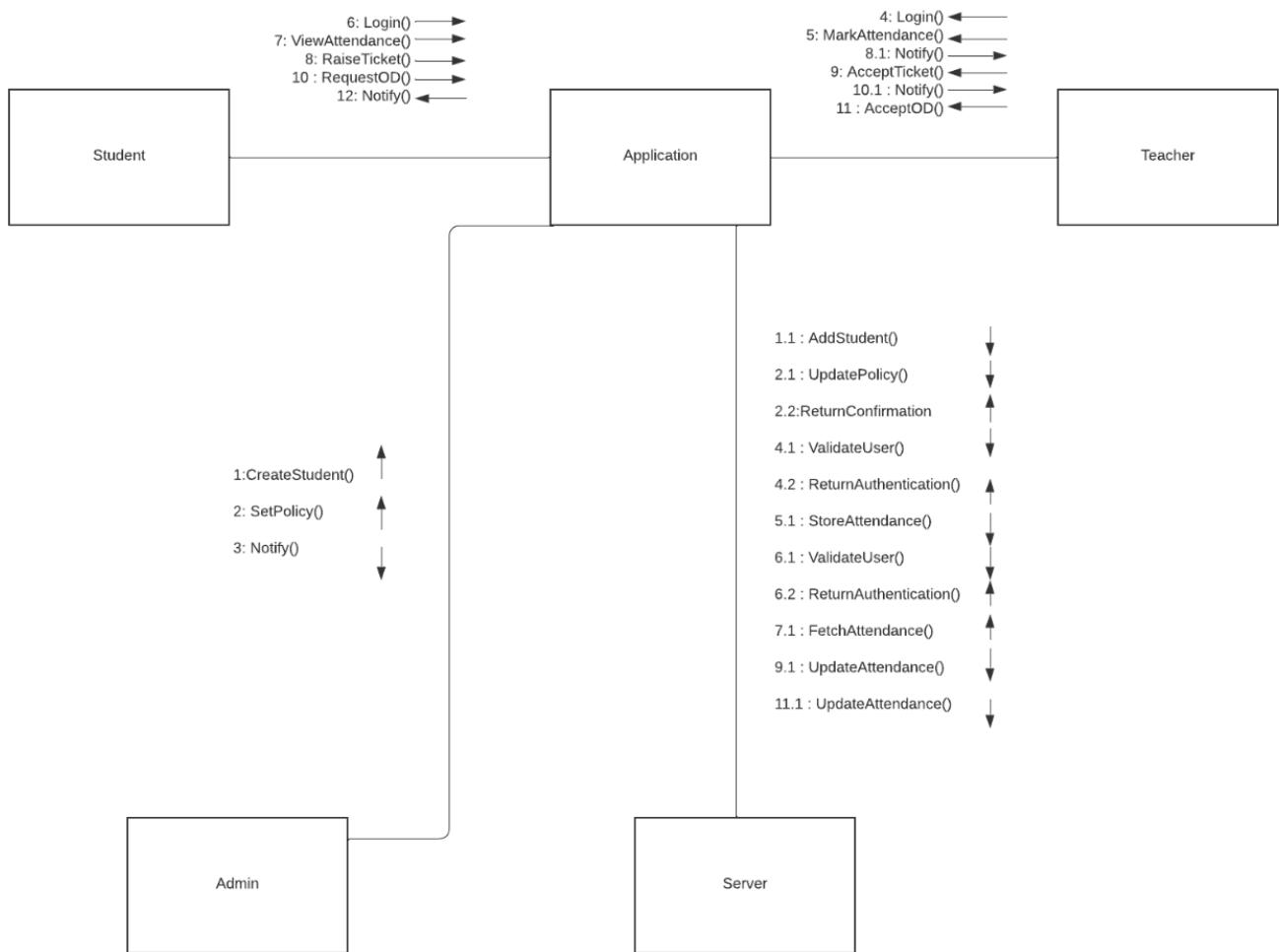
Request OD :



Accept/Reject OD :



Collaboration Diagram :



**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO.: 6

Activity Diagram

Design an activity diagram for the chosen application depicting the major activities involved in the application with control flow and object flow edges. [CO3, K2, K3 & K4]

The interaction diagrams include the following.

1. An activity diagram for the main success scenario in the use case diagram.
2. An activity diagram for each function (Use Case). (Number of activity diagrams according to the main functions (use cases) you have in your proposed system).

List all the major activities in the application.

Identify the partitions (which corresponds to each class)

Note:

Include the following notations in the Activity Diagram.

- Activity edge
- Activity edge with Guards
- Object flow edge
- Interrupting Edge
- Swimlane
- Control Nodes (Initial, Flow final, Activity final, Decision, Merge, Fork and join Nodes)
- Object Node (Pin, Central Buffer, Parameter, expansion nodes)
- Action

ACTIVITY DIAGRAM

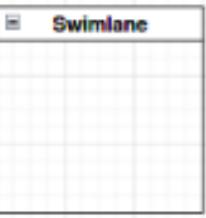
Ex No :

Date :

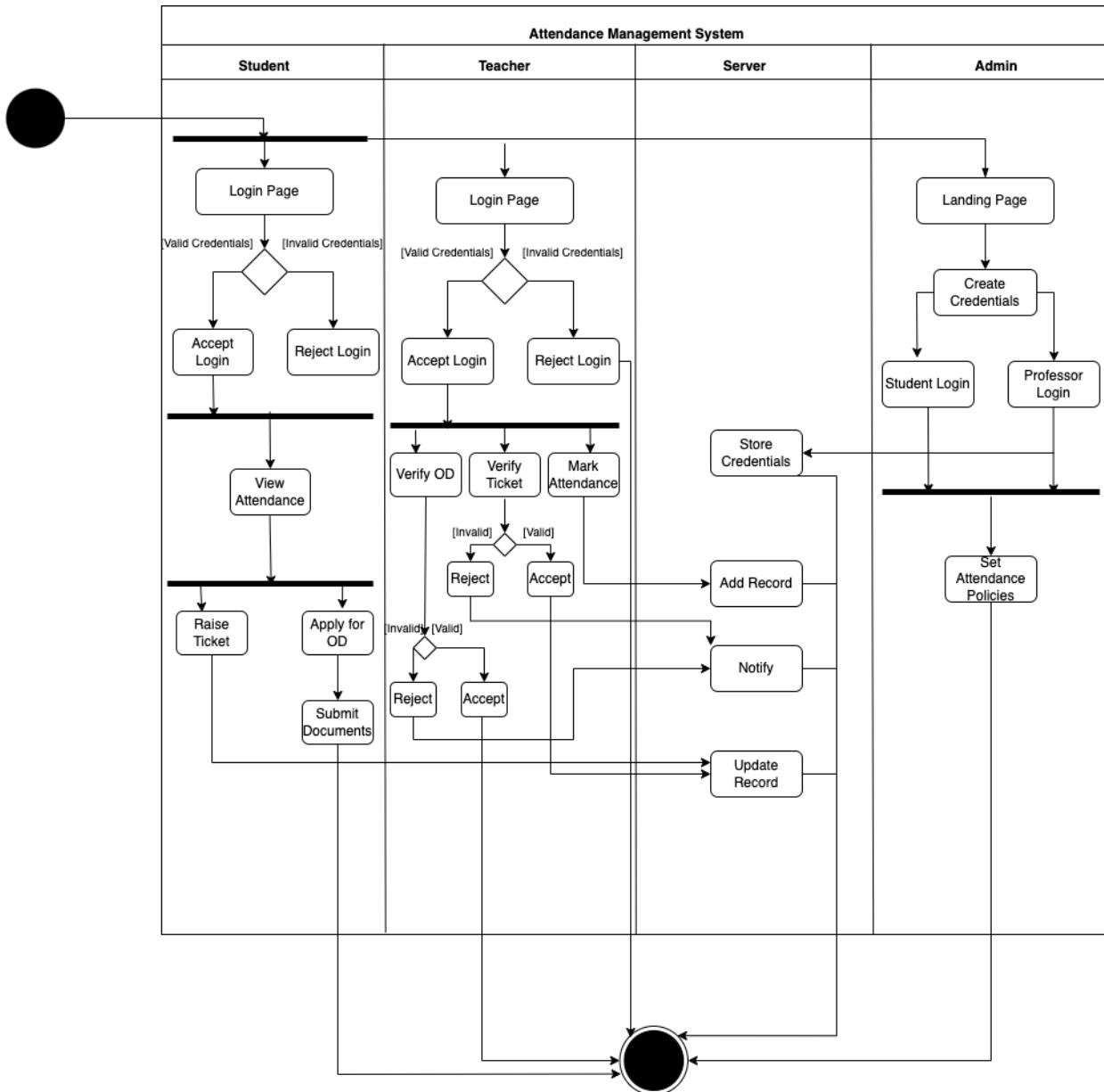
Aim:

To establish and define various states and define transitions between the defined states.

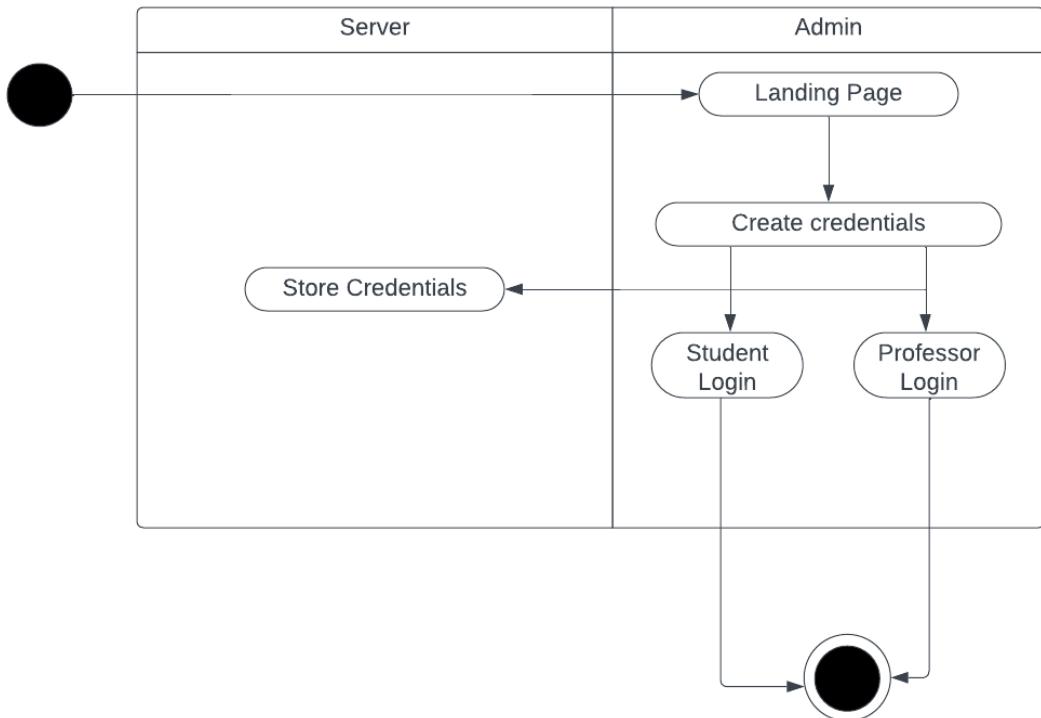
Notations for Activity Diagram:

Notation	Type	Explanation
	Start	It initiates the transition
	End	It marks the end if the transitions
	State	Each state in the state machine is represented using this rounded rectangle
	Decision	It represents the transitions based on the conditions
	Swim Lane	It is used for grouping same user's activities in one column
	Fork	It is used for concurrent executions

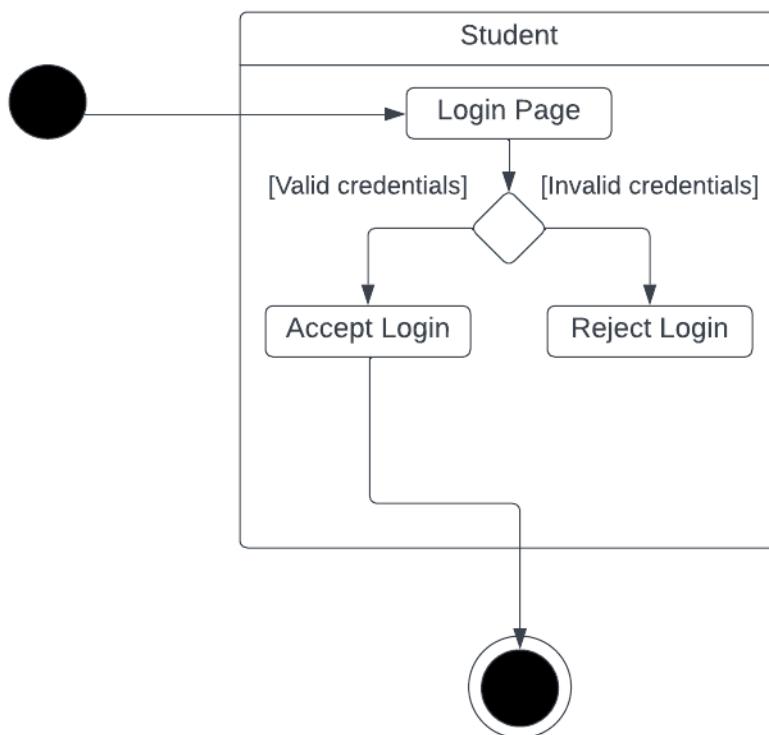
Activity Diagram :



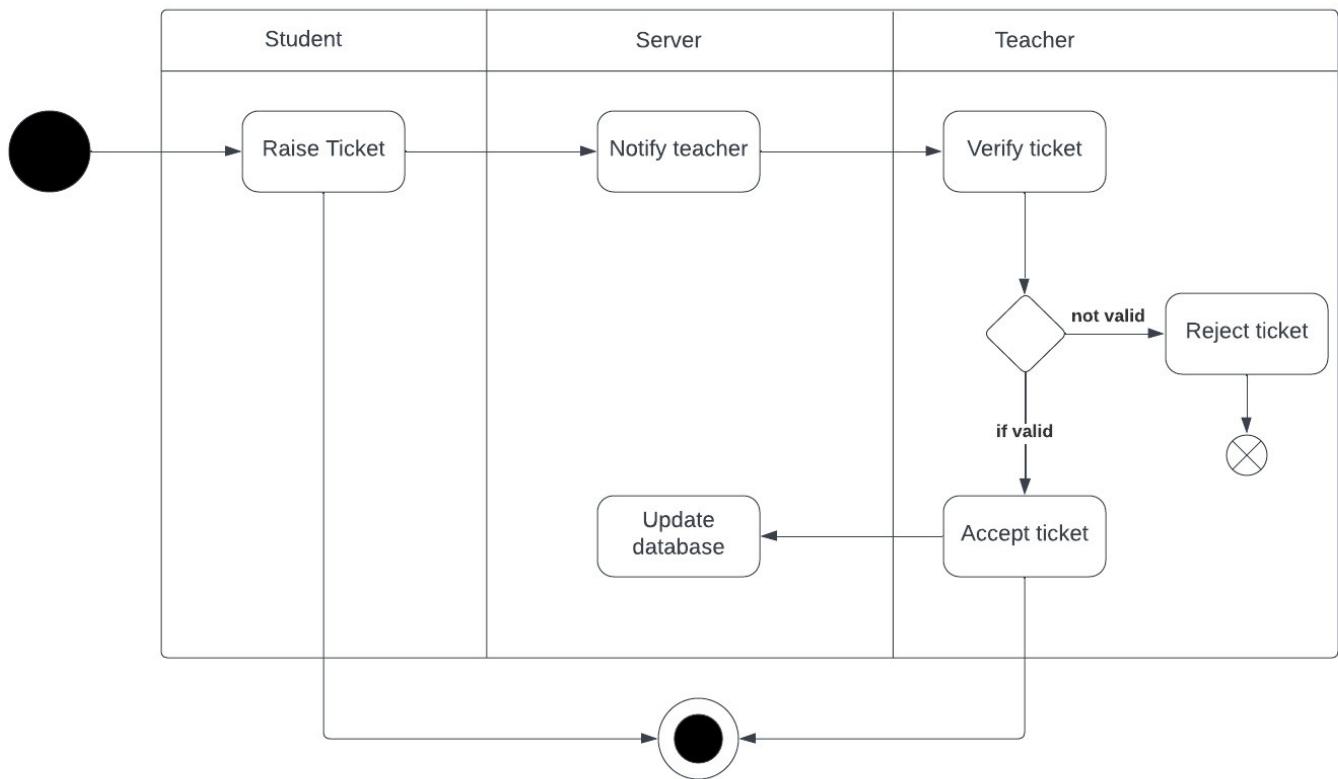
Registration and Policies Setup:



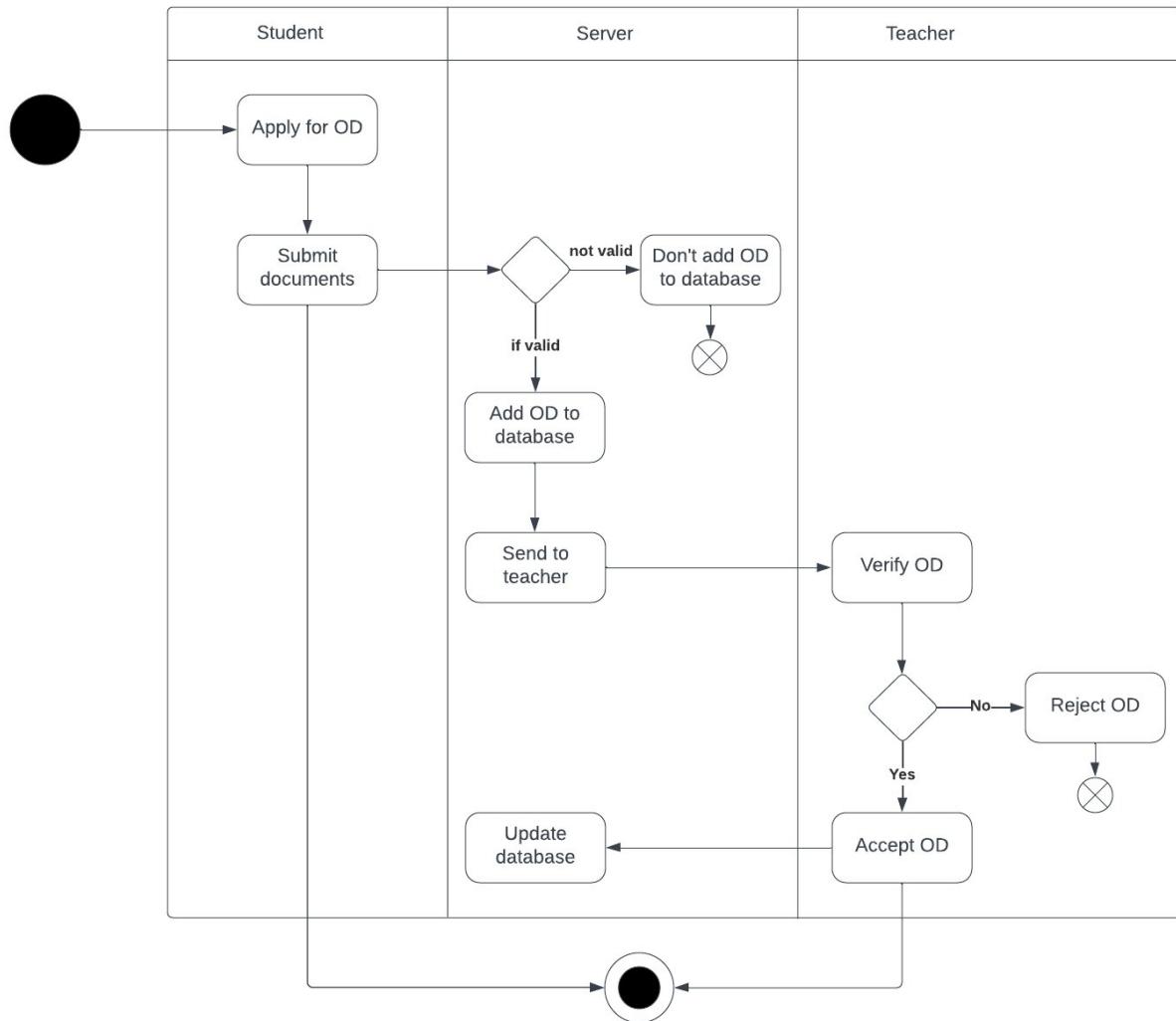
Login:



Raise Ticket and Verification:



OD application and Acceptance:



**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 7

State Machine Diagrams

Design a state machine diagram for every object involved in the chosen application depicting the states that an object will go through in the form of transition based on different events that would occur in the system and the resultant action. [CO3, K2, K3 & K4]

The interaction diagrams include the following.

1. A state machine diagram for every object of a class in the application. (Refer class diagram and draw a state machine diagram for each class)

Note:

Consider identifying and designing the following notations in the state machine diagram.

- Behavioral state machine diagram
 - Simple state
 - Composite state
 - Orthogonal composite state
 - Submachine state
 - Pseudo state machine
- Protocol state machine diagram
 - State
 - Transition

STATE MACHINE DIAGRAM

Ex No :

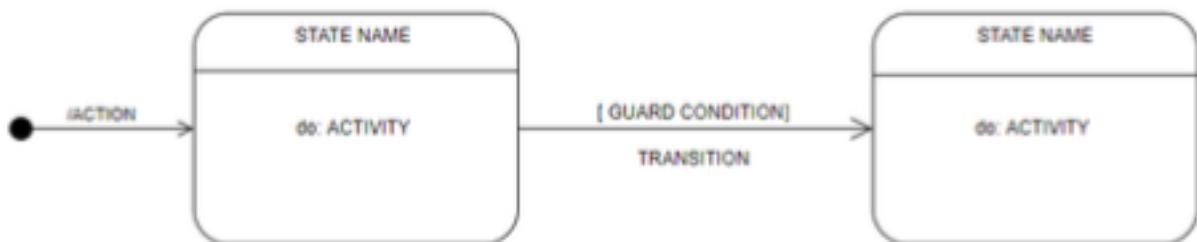
Date :

Aim :

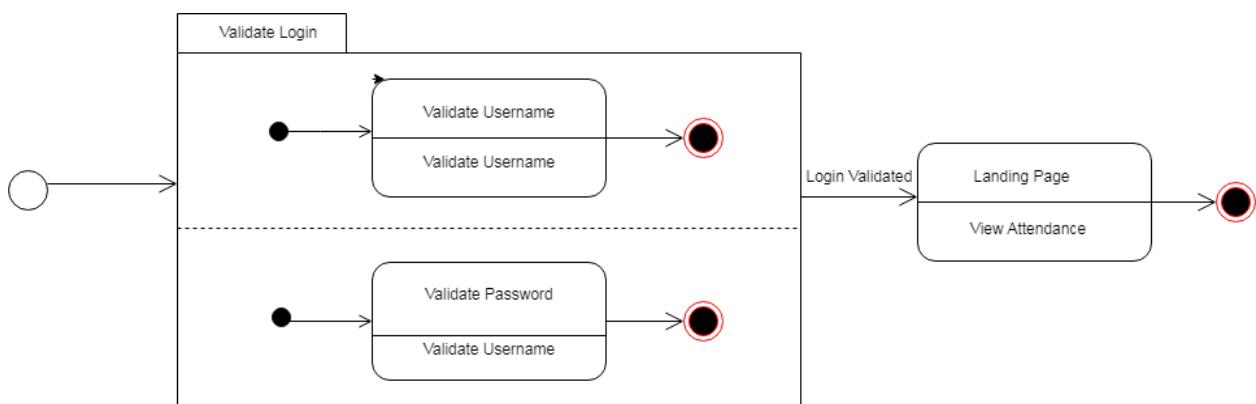
To establish and define various states and define transitions between the defined states.

UML Notations for state machine diagram:

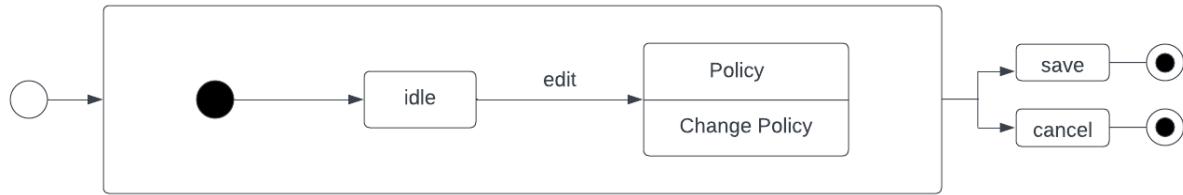
UML STATE MACHINE DIAGRAM NOTATIONS



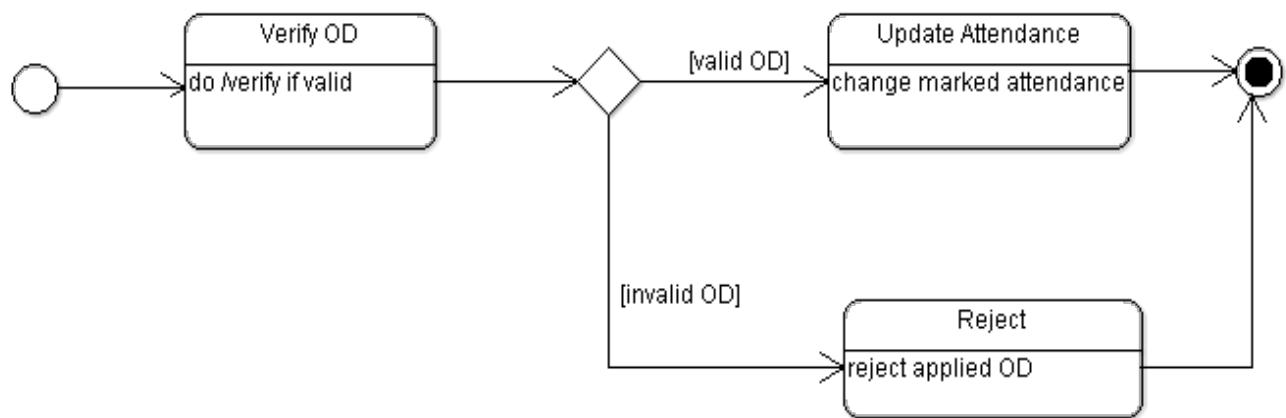
VALIDATE LOGIN



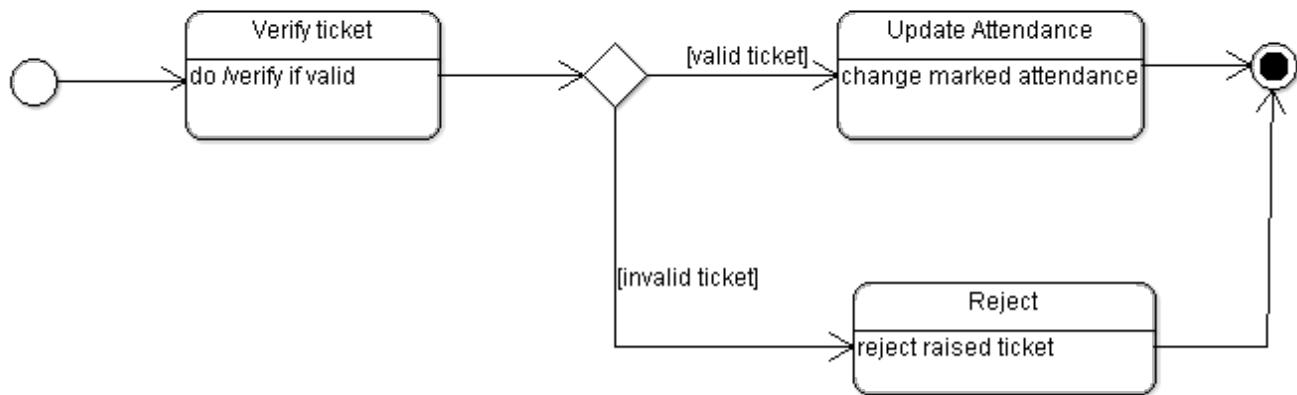
SET POLICIES



VERIFY OD



VERIFY TICKET



**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 8

Component, Deployment & Package Diagrams

Design Component, Deployment & Package Diagrams for the chosen application. [CO3, K2, K3 & K4]

Note:

In component diagram specify the following.

- components,
- interfaces,
- ports and
- relationships.

In deployment diagram specify artifacts mentioned below.

- executable files,
- libraries,
- archives,
- database servers and
- configuration files.

In package diagram specify the following.

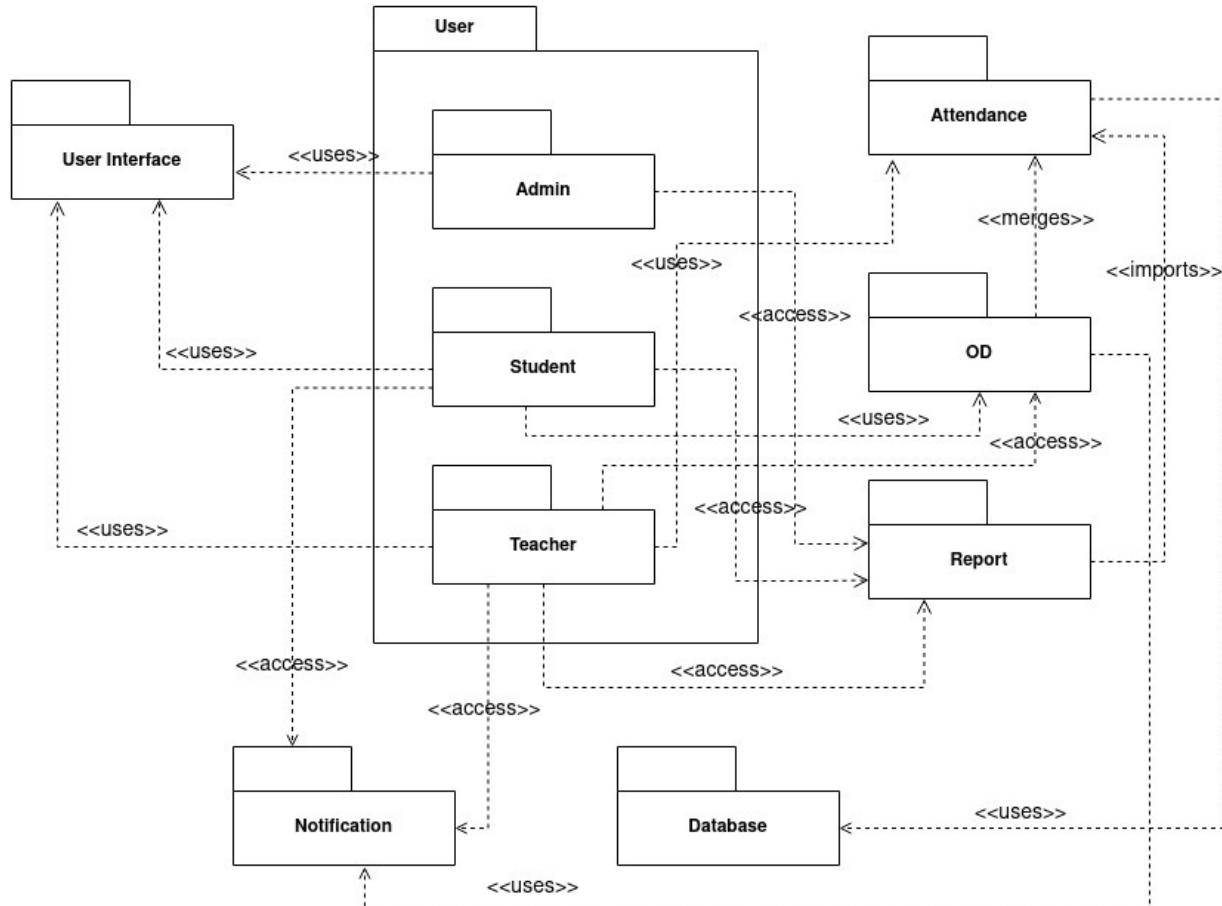
- Package
- Dependency
- Import
- Access
- Merge

Package Diagram, Deployment Diagram and Component Diagram

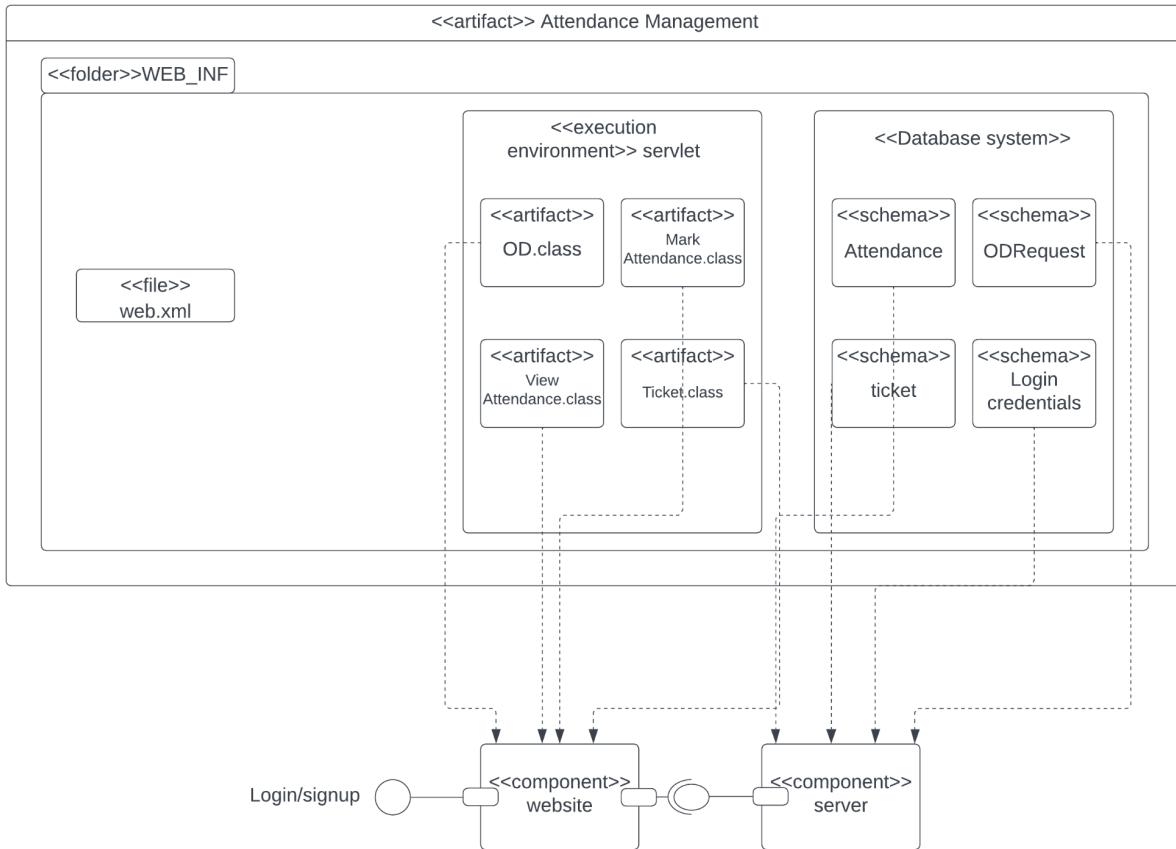
Ex No :

Date :

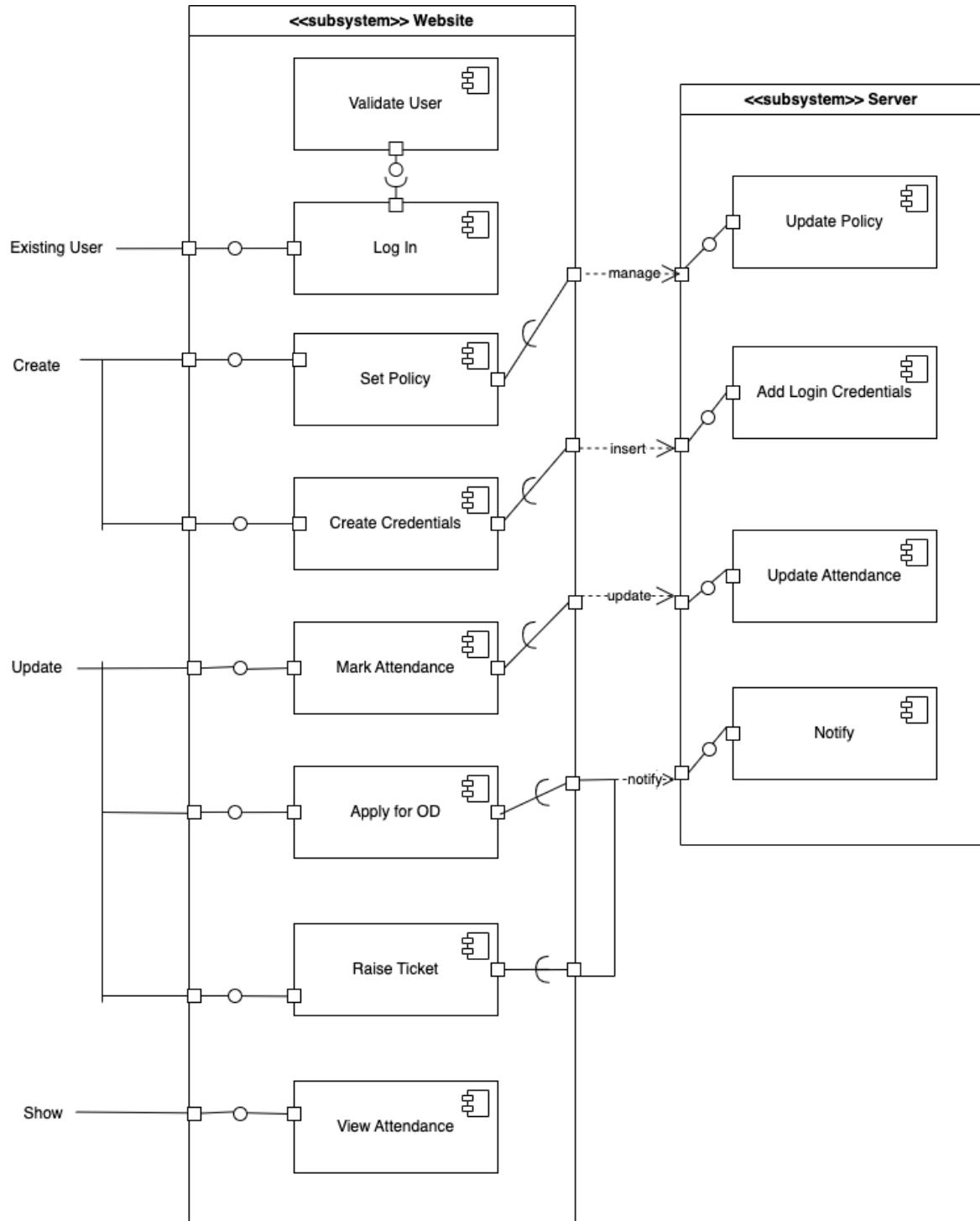
Package Diagram



Deployment Diagram



Component Diagram



**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 9

Implementation of UI & Domain Layers

Implement the UI layer according to the design in the deployment diagram. [CO3, K3 & K4]
Implement the Domain layer according to the class diagram. [CO3, K3 & K4]

UI Layer

HTML

CSS

Javascript

MySQL

Domain Layer

Map the class diagram to the code using OOP concepts.

Use Netbeans and Tomcat Server.

Implementation of UI & Domain Layers

Ex No :

Date :

Aim :

To implement the UI layer according to the design in the deployment diagram and the Domain layer according to the class diagram.

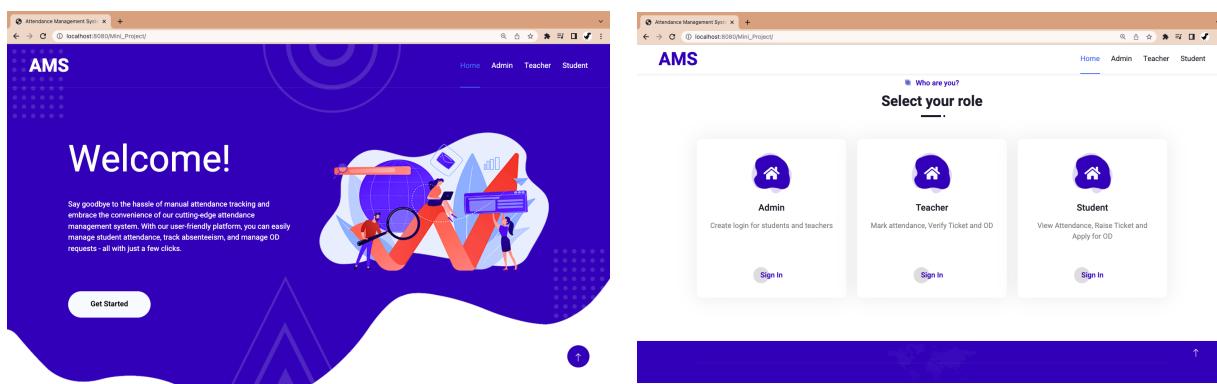
Tech Stack :

UI Layer

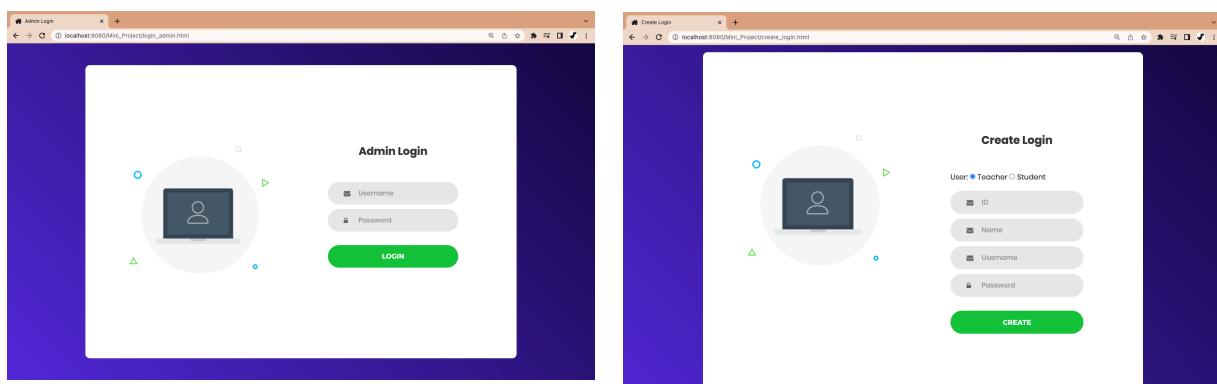
- HTML
- CSS
- Javascript
- MySQL

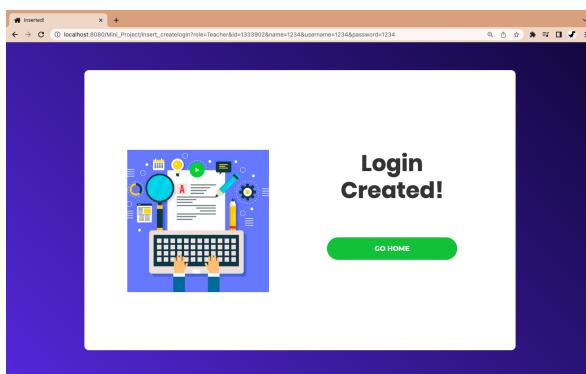
Domain Layer

- Tomcat Server

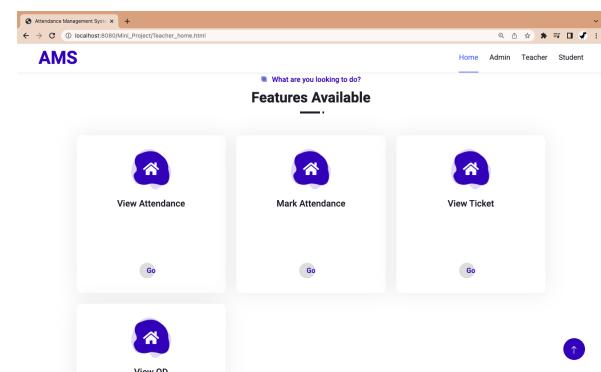
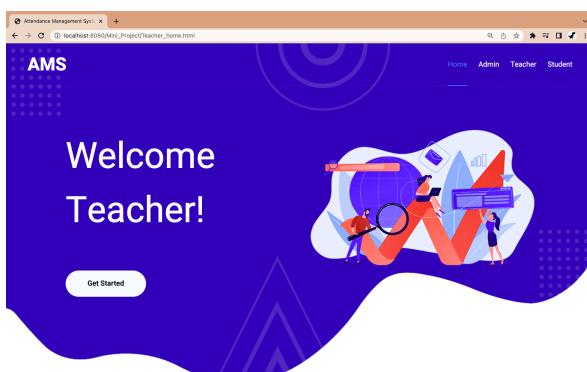
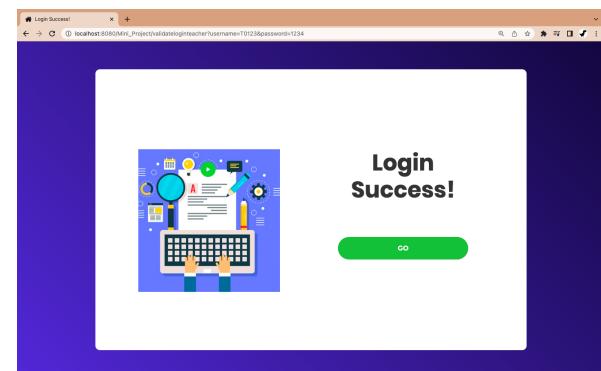
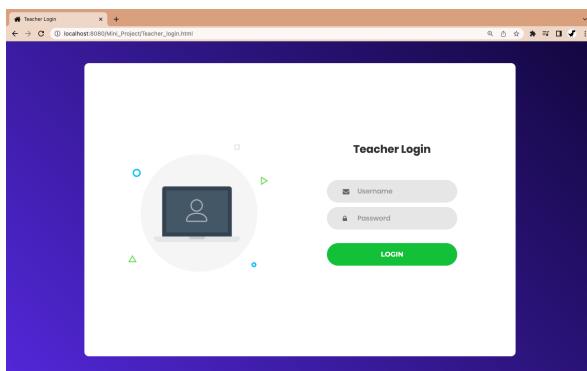


ADMIN





TEACHER



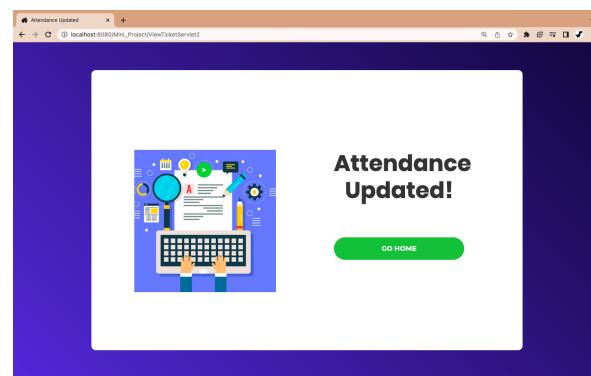
Attendance			
Date	ID	Name	Present
2023-03-10	20001	null	No
2023-04-01	20001	null	Yes
2023-04-14	20001	null	Yes
2023-05-01	20001	null	No
2023-05-02	20001	null	Yes
2023-03-10	20002	null	Yes
2023-04-01	20002	null	Yes
2023-04-14	20002	null	Yes
2023-05-01	20002	null	No
2023-05-02	20002	null	Yes
2023-03-10	20009	Pooja	Yes
2023-04-01	20009	Pooja	No

Mark Attendance:		
Student ID	Name	Present or Absent
20001	null	<input type="radio"/> Yes <input type="radio"/> No
20002	null	<input type="radio"/> Yes <input type="radio"/> No
20009	Pooja	<input type="radio"/> Yes <input type="radio"/> No
23314	null	<input type="radio"/> Yes <input type="radio"/> No
122290	Ashley Simpson	<input type="radio"/> Yes <input type="radio"/> No
123390	Alisha Denvers	<input type="radio"/> Yes <input type="radio"/> No

Verify Ticket

Student ID	Name	Date	Accept or Reject
163790	Mary Simone	2023-04-01	<input checked="" type="radio"/> Accept <input type="radio"/> Reject

SUBMIT



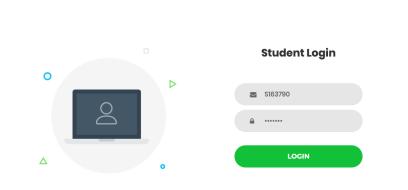
Verify OD

Student ID	Name	Date	Reason	Accept or Reject
163790	Mary Simone	2023-04-14	Culturals	<input type="radio"/> Accept <input checked="" type="radio"/> Reject

SUBMIT

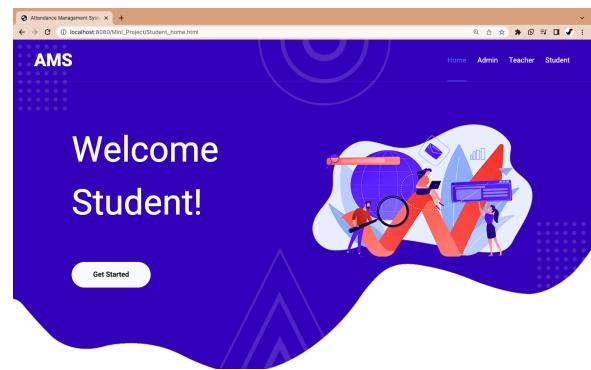
STUDENT

Student Login



Student Login

STUDENT ID: 163790
PASSWORD:
LOGIN



AMS

What are you looking to do?

Features Available

- View Attendance**
- Raise Ticket**
- Apply for OD**
- Check OD Status**
- Check Ticket Status**

My Attendance

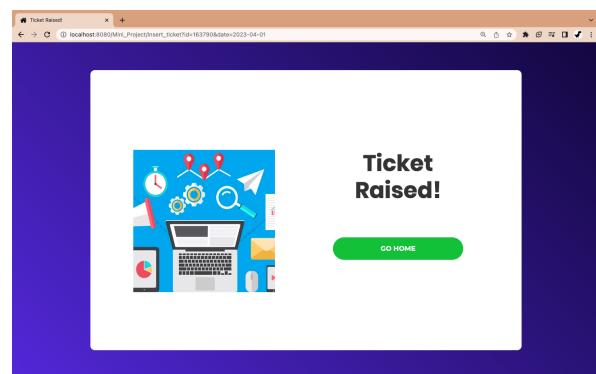
Date	Present
2023-03-10	No
2023-04-01	No
2023-04-14	No
2023-05-01	No
2023-05-02	Yes

GO HOME

Raise Ticket

163790
01/04/2023

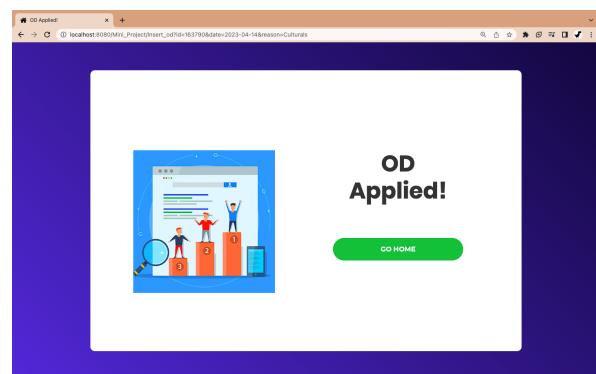
RAISE



Apply OD

163790
14/04/2023
Culturals

APPLY



Your OD Status

Date	Reason	Status
2023-04-14	Culturals	Rejected
2023-05-01	Culturals	Rejected

GO HOME

Your Ticket Status

Date	Status
2023-04-01	Accepted

GO HOME

**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 10

Implementation of Technical Service Layer

Implement the major functionalities of the application. [CO3, K3 & K4]
Integrate the modules (concepts) involved in the class diagram. [CO3, K3 & K4]

UI Layer

HTML
CSS
Javascript
MySQL

Domain Layer

Map the class diagram to the code using OOP concepts.
Use Netbeans and Tomcat Server.

Technical Service Layer

Major functionalities
Establish DB connectivity, store and retrieve data related to the application.
Integrate the modules together.

UCS1617- Mini Project **Implementation of Technical Layer**

Ex No :

Date :

Aim :

To implement the major functionalities of the application and integrate the modules (concepts) involved in the class diagram.

Tech Stack :

UI Layer

- HTML
- CSS
- Javascript
- MySQL

Domain Layer

- Tomcat Server

Technical Service Layer

- DB - MySQL

```
[mysql]> show tables;
+-----+
| Tables_in_mini_project |
+-----+
| attendance
| login
| od
| ticket
+-----+
4 rows in set (0.04 sec)
```

```
[mysql> describe login;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| role  | enum('Teacher','Student') | YES  |     | NULL    |       |
| id    | int(11)          | NO   | PRI | NULL    |       |
| name  | varchar(30)        | YES  |     | NULL    |       |
| username | varchar(30)        | YES  |     | NULL    |       |
| password | varchar(30)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

[mysql> describe od;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11)          | NO   | PRI | NULL    |       |
| od_date | date            | NO   | PRI | NULL    |       |
| reason | varchar(200)      | YES  |     | NULL    |       |
| status | enum('Applied','Accepted','Rejected') | YES  |     | Applied |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

[mysql> describe ticket;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11)          | NO   | PRI | NULL    |       |
| ticket_date | date            | NO   | PRI | NULL    |       |
| status | enum('Applied','Accepted','Rejected') | YES  |     | Applied |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

[mysql> describe attendance;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11)          | NO   | PRI | NULL    |       |
| curr_date | date            | NO   | PRI | NULL    |       |
| name  | varchar(30)        | YES  |     | NULL    |       |
| present | enum('Yes','No') | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

DatabaseAccess_ViewAttendance

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class DatabaseAccess_ViewAttendance extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
```

```

// Database credentials
static final String USER = "root";
static final String PASS = "mysql";
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    // Set response content type
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "Attendance";

    out.println("<!doctype html>\n" +
               "<html>\n" +
               "  <head><title>View Attendance</title><meta charset=\"UTF-8\"><meta name=\"viewport\""
content="width=device-width, initial-scale=1"><link rel=\"icon\" type=\"image/png\""
href="images/icons/favicon.ico"/><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/bootstrap/css/bootstrap.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="fonts/fontawesome-4.7.0/css/fontawesome.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/animate/animate.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/css-hamburgers/hamburgers.min.css"/><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/select2/select2.min.css"/><link rel=\"stylesheet\" type=\"text/css\" href="css/util.css"/><link
rel=\"stylesheet\" type=\"text/css\" href="css/main.css"/>" +
               "</head>\n<body>\n" +
               "<div class=\"limiter\"><div class=\"container-login100\"><div class=\"card\" align=\"center;\""
>" +

    "<span class=\"login100-form-title\" style=\"font-size:
30px;text-align:center;\">Attendance</span>\n");

    try {

        //out.println("<p>Driver Access Initialising....</p>");
        // Register JDBC driver
        Class.forName(JDBC_DRIVER);
        //Class.forName("com.mysql.cj.jdbc.Driver");

        //out.println("<p>Driver Access Succesful....</p>");
        // Open a connection
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
        //out.println("<p>Database Connection Successful....</p>");

        // Execute SQL query
        Statement stmt = conn.createStatement();
        String sql;
        sql = "SELECT * FROM attendance";
        ResultSet rs = stmt.executeQuery(sql);
        out.print("<table>");
        out.print("<tr><th>Date</th>");
        out.print("<th>ID</th>");
        out.print("<th>Name</th>");
        out.print("<th>Present</th>");
        out.print("</tr>");


```

```

// Extract data from result set
while (rs.next()) {
    //Retrieve by column name
    String curr_date = rs.getString("curr_date");
    int id = rs.getInt("id");
    String name = rs.getString("name");
    String present = rs.getString("present");

    //Display values
    out.print("<tr><td style=\"padding:10px 10px 10px 10px;\">" + curr_date +
    "</td>");
    out.print("<td style=\"padding:10px 10px 10px 10px;\">" + id + "</td>");
    out.print("<td style=\"padding:10px 10px 10px 10px;\">" + name + "</td>");
    out.print("<td style=\"padding:10px 10px 10px 10px;\">" + present + "</td></tr>");
}

// Clean-up environment
rs.close();
stmt.close();
conn.close();
out.println("</table>");
out.println("<div class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
onclick="location.href='Teacher_home.html'" type="button">Go Home</button>");
out.println("</div></div></div><script"
src="vendor/jquery/jquery-3.2.1.min.js"></script><script src="vendor/bootstrap/js/popper.js"></script><script"
src="vendor/bootstrap/js/bootstrap.min.js"></script><script src="vendor/select2/select2.min.js"></script><script"
src="vendor/tilt/tilt.jquery.min.js"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script"
src="js/main1.js"></script></body></html>");

} catch (SQLException se) {
    //Handle errors for JDBC
    se.printStackTrace();
} catch (Exception e) {
    //Handle errors for Class.forName
    e.printStackTrace();
}
}
}
}

```

DatabaseAccess_ViewAttendanceStud

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class DatabaseAccess_ViewAttendanceStud extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";

```

```

// Database credentials
static final String USER = "root";
static final String PASS = "mysql";
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    // Set response content type
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "My Attendance";
    Integer id = 0;
    HttpSession session = request.getSession();
    String user = (String)session.getAttribute("username");

    out.println("<!doctype html>\n" +
               "<html>\n" +
               "  <head><title>My Attendance</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/fontawesome-4.7.0/css/fontawesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\">" +
               "  </head>\n<body>\n" +
               "  <div class=\"limiter\"><div class=\"container-login100\"><div class=\"card\" align=\"center;\">" +
               "    <span class=\"login100-form-title\" style=\"font-size: 30px;text-align:center;\">My
Attendance</span>\n");
    try {

        //out.println("<p>Driver Access Initialising....</p>");
        // Register JDBC driver
        Class.forName(JDBC_DRIVER);
        //Class.forName("com.mysql.cj.jdbc.Driver");

        //out.println("<p>Driver Access Succesful....</p>");
        // Open a connection
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
        //out.println("<p>Database Connection Successful....</p>");

        // Execute SQL query
        Statement stmt = conn.createStatement();
        String sql;

        sql = "SELECT id FROM login where username = " + user + "";
        ResultSet rs1 = stmt.executeQuery(sql);
        if (rs1.next()) {
            //Retrieve by column name

```

```

        id = rs1.getInt("id");
    }
    rs1.close();

    sql = "SELECT * FROM attendance where id=" + id;
    ResultSet rs = stmt.executeQuery(sql);
    out.print("<table style = \"width:80%;margin:auto;\" border = \"1\">");
    out.print("<tr><th style=\"width:50%;padding:10px 10px 10px 10px;\">Date</th>");
    //out.print("<th style=\"width:50%;padding:10px 10px 10px 10px;\">ID</th>");
    //out.print("<th style=\"width:50%;padding:10px 10px 10px 10px;\">Name</th>");
    out.print("<th style=\"width:50%;padding:10px 10px 10px 10px;\">Present</th>");
    out.print("</tr>");

    // Extract data from result set
    while (rs.next()) {
        //Retrieve by column names
        String curr_date = rs.getString("curr_date");
        //int ids = rs.getInt("id");
        //String name = rs.getString("name");
        String present = rs.getString("present");

        //Display values
        out.print("<tr><td style=\"padding:10px 10px 10px 10px;\">" + curr_date +
        "</td>");
        //out.print("<td style=\"padding:10px 10px 10px 10px;\">" + ids + "</td>");
        //out.print("<td style=\"padding:10px 10px 10px 10px;\">" + name + "</td>");
        out.print("<td style=\"padding:10px 10px 10px 10px;\">" + present + "</td></tr>");
    }

    // Clean-up environment
    rs.close();
    stmt.close();
    conn.close();
    out.println("</table>");
    out.println("<div class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
    onclick="location.href='Student_home.html'" type="button">Go Home</button>");
    out.println("</div></div></div><script"
    src="vendor/jquery/jquery-3.2.1.min.js"></script><script src="vendor/bootstrap/js/popper.js"></script><script"
    src="vendor/bootstrap/js/bootstrap.min.js"></script><script src="vendor/select2/select2.min.js"></script><script"
    src="vendor/tilt/tilt.jquery.min.js"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script"
    src="js/main1.js"></script></body></html>");

    } catch (SQLException se) {
        //Handle errors for JDBC
        se.printStackTrace();
    } catch (Exception e) {
        //Handle errors for Class.forName
        e.printStackTrace();
    }
}
}

```

DatabaseAccess_ViewOD

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class DatabaseAccess_ViewOD extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Set response content type
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String title = "OD Status";
        out.println("<!doctype html>\n" +
                   "<html>\n" +
                   "  <head><title>View OD Status</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/fontawesome-4.7.0/css/fontawesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\">" +
                   "</head>\n<body>\n" +
                   "  <div class=\"limiter\"><div class=\"container-login100\"><div class=\"card\" align=\"center;\">\n" +
                   "    <span class=\"login100-form-title\" style=\"font-size: 30px;text-align:center;\">Your OD\nStatus</span>\n" +
                   "    <table style = \"width:80%;margin:auto;\" border = \"1\">\n" +
                   "      <tr>\n" +
                   "        <th style=\"width:50%;padding:10px 10px 10px 10px;\">Date</th>\n" +
                   "        <th style=\"width:50%;padding:10px 10px 10px 10px;\">Reason</th>\n" +
                   "        <th style=\"width:50%;padding:10px 10px 10px 10px;\">Status</th>\n" +
                   "      </tr>\n" +
                   "    );
    try {
        // Register JDBC driver
        Class.forName(JDBC_DRIVER);
```

```

// Open a connection
Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);

// Execute SQL query
Statement stmt = conn.createStatement();

String sql;
Integer id = 0;
HttpSession session = request.getSession();
String user = (String)session.getAttribute("username");
sql = "SELECT id FROM login where username = " + user + "";

ResultSet rs1 = stmt.executeQuery(sql);
if (rs1.next()) {
    //Retrieve by column name
    id = rs1.getInt("id");
}
rs1.close();

sql = "SELECT * FROM od where id = " + id;
ResultSet rs2 = stmt.executeQuery(sql);

// Extract data from result set
while (rs2.next()) {
    //Retrieve by column name
    String curr_date = rs2.getString("od_date");
    String reason = rs2.getString("reason");
    String status = rs2.getString("status");

    //Display values
    out.print("<tr><td style=\"padding:10px 10px 10px 10px;\">" + curr_date +
    "</td>");
    out.print("<td style=\"padding:10px 10px 10px 10px;\">" + reason + "</td>");
    out.println("<td style=\"padding:10px 10px 10px 10px;\">" + status + "</td></tr>");
}
out.println("</table>");
out.println("<div class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
onclick="location.href='Student_home.html'" type=\"button\">Go Home</button>");
out.println("</div></div></div><script"
src="vendor/jquery/jquery-3.2.1.min.js"></script><script src="vendor/bootstrap/js/popper.js"></script><script
src="vendor/bootstrap/js/bootstrap.min.js"></script><script src="vendor/select2/select2.min.js"></script><script
src="vendor/tilt/tilt.jquery.min.js"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src="js/main1.js"></script></body></html>");

// Clean-up environment
rs2.close();
stmt.close();
conn.close();
} catch (SQLException se) {
    //Handle errors for JDBC
    se.printStackTrace();
}

catch (Exception e) {

```

```
        //Handle errors for Class.forName  
        e.printStackTrace();  
    }  
}
```

DatabaseAccess_ViewTicket

```

// Open a connection
Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);

// Execute SQL query
Statement stmt = conn.createStatement();

String sql;
Integer id = 0;
HttpSession session = request.getSession();
String user = (String)session.getAttribute("username");
sql = "SELECT id FROM login where username = " + user + """;

ResultSet rs1 = stmt.executeQuery(sql);
if (rs1.next()) {
    //Retrieve by column name
    id = rs1.getInt("id");
}
rs1.close();

sql = "SELECT * FROM ticket where id = " + id;
ResultSet rs2 = stmt.executeQuery(sql);

// Extract data from result set
while (rs2.next()) {
    //Retrieve by column name
    String curr_date = rs2.getString("ticket_date");
    String status = rs2.getString("status");

    //Display values
    out.print("<tr><td style=\"padding:10px 10px 10px 10px;\">" + curr_date +
"</td>");
    out.println("<td style=\"padding:10px 10px 10px 10px;\">" + status + "</td></tr>");
}
out.println("</table>");
out.println("<div class=\"container-login100-form-btn\"><button class=\"login100-form-btn\" onclick=\"location.href='Student_home.html' type=\"button\">Go Home</button></div>");
out.println("</div></div></div><script src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script src=\"js/main1.js\"></script></body></html>");

// Clean-up environment
rs2.close();
stmt.close();
conn.close();
} catch (SQLException se) {
    //Handle errors for JDBC
    se.printStackTrace();
}

catch (Exception e) {
    //Handle errors for Class.forName
}

```

```
        e.printStackTrace();
    }
}
```

Insert_createlogin

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class Insert_createlogin extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";

    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Set response content type
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        //out.println("<!DOCTYPE html><html lang='en'><head><title>Inserted!</title><meta charset='UTF
- 8'><meta name='viewport' content='width = device - width, initial - scale = 1'><link rel='icon' type='image / png'
href='images / icons / favicon.ico'/><link rel='stylesheet' type='text / css' href='vendor / bootstrap / css /
bootstrap.min.css'><link rel='stylesheet' type='text / css' href='fonts / font - awesome - 4.7.0 / css / font -
awesome.min.css'><link rel='stylesheet' type='text / css' href='vendor / animate / animate.css'><link rel='stylesheet'
type='text / css' href='vendor / css - hamburgers / hamburgers.min.css'><link rel='stylesheet' type='text / css'
href='vendor / select2 / select2.min.css'><link rel='stylesheet' type='text / css' href='css / util.css'><link rel='stylesheet'
type='text / css' href='css / main.css'></head><body>"); 
        //out.println("<!DOCTYPE html><html lang='en'><head><title>Inserted!</title><meta charset='UTF
- 8'><meta name='viewport' content='width = device - width, initial - scale =
1'><!--=====
=====--> <link rel='icon' type='image / png' href='images / icons /
favicon.ico'/><!--=====
=====--><link rel='stylesheet' type='text / css' href='vendor / bootstrap / css /
bootstrap.min.css'><!--=====
=====--><link rel='stylesheet' type='text / css' href='fonts / font - awesome - 4.7.0 / css / font -
awesome.min.css'><!--=====
=====--><link rel='stylesheet' type='text / css' href='vendor / animate / animate.css'><!--=====
=====--><link rel='stylesheet' type='text / css' href='vendor / css - hamburgers / hamburgers.min.css'><!--=====
=====--><link rel='stylesheet' type='text / css' href='vendor / select2 / select2.min.css'><!--=====
=====--><link rel='stylesheet' type='text / css' href='css / util.css'><!--=====
=====--><link rel='stylesheet' type='text / css' href='css / main.css'>
```

```

main.css'"><!--=====
=====--></head><body>");
try {
    // Register JDBC driver
    //Class.forName("com.mysql.cj.jdbc.Driver");
    Class.forName(JDBC_DRIVER);

    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    //out.println(Integer.valueOf(request.getParameter("id")) + request.getParameter("name")
+ Integer.valueOf(request.getParameter("age")));
    //out.println(request.getParameter("phone-no") + request.getParameter("address") +
request.getParameter("gender") + request.getParameter("marit-stat") + request.getParameter("dov"));
    // Execute SQL query
    PreparedStatement st = conn
        .prepareStatement("insert into login values(?, ?, ?, ?, ?, ?)");

    st.setString(1, request.getParameter("role"));
    st.setInt(2, Integer.valueOf(request.getParameter("id")));
    st.setString(3, request.getParameter("name"));
    st.setString(4, request.getParameter("username"));
    st.setString(5, request.getParameter("password"));
    st.executeUpdate();

    // Close all the connections
    st.close();
    conn.close();

    // Get a writer pointer
    // to display the successful result

    //out.println(
    // "<div class='limiter'><div class='container - login100'><div class='wrap - login100'><div
class='login100 - pic js - tilt' data-tilt><img src='img / portfolio - 3.jpg' alt='IMG'></div><form class='login100 - form
validate - form' action="><span class='login100 - form - title' style='font - size: 50px;'>Login Created!</span><div
class='container - login100 - form - btn'><button class='login100 - form - btn' onclick='location.href = 'index.html'' type='button'>Go Home</button></div><br><br><br><br><br><br></form></div></div><script src='vendor
/ jquery / jquery - 3.2.1.min.js'></script> <script src='vendor / bootstrap / js / popper.js'></script><script src='vendor /
bootstrap / js / bootstrap.min.js'></script><script src='vendor / select2 / select2.min.js'></script><script
src='vendor/tilt/tilt.jquery.min.js'></script><script >$('js-tilt').tilt({scale: 1.1})</script><script
src='js/main1.js'></script>");

    //out.println("<div class='limiter'><div class='container - login100'><div class='wrap -
login100'><div class='login100 - pic js - tilt' data-tilt><img src='img / portfolio - 3.jpg' alt='IMG'></div><form
class='login100 - form validate - form' action="><span class='login100 - form - title' style='font - size: 50px'>Login
Created!</span><div class='container - login100 - form - btn'><button class='login100 - form - btn'
onclick='location.href = 'index.html'' type='button'>Go
Home</button></div><br><br><br><br><br><br></form></div></div><!--=====
=====--> <script
src='vendor / jquery / jquery -
3.2.1.min.js'></script><!--=====
=====--><script src='vendor / bootstrap / js / popper.js'></script><script src='vendor /
bootstrap / js /
bootstrap.min.js'></script><!--=====
=====--><script src='vendor / select2 /

```

```

select2.min.js'></script><!--=====--><script src='vendor / tilt / tilt.jquery.min.js'></script><script >$('.js-tilt').tilt({scale: 1.1})</script><!--=====--><script src='js / main1.js'></script>");
        out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Inserted!</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img src=\"img/portfolio-3.jpg\" alt=\"IMG\"/></div><form class=\"login100-form validate-form\" action=\"\"><span class=\"login100-form-title\" style=\"font-size: 50px!>Login Created!</span><div class=\"container-login100-form-btn\"><button class=\"login100-form-btn\" onclick=\"location.href='index.html'\" type=\"button\">Go Home</button></div><br><br><br><br><br><br></form></div></div><script src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script src=\"js/main1.js\"></script>");
    } catch (Exception e) {
        out.print("<p>Error</p>");
        e.printStackTrace();
    } finally {
        //out.println("<form action='http://localhost:8080/PMS/ID' method='get'><button type='submit'>View</button>\"");
        out.println("</body></html>");
    }
}
}

```

Insert_od

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class Insert_od extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";

    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Set response content type
        response.setContentType("text/html");

```

```

PrintWriter out = response.getWriter();
//out.println("<html><head><title>Insert OD</title></head><body>");

try {
    // Register JDBC driver
    //Class.forName("com.mysql.cj.jdbc.Driver");
    Class.forName(JDBC_DRIVER);

    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    //out.println(Integer.valueOf(request.getParameter("id")) + request.getParameter("name")
+ Integer.valueOf(request.getParameter("age")));
    //out.println(request.getParameter("phone-no") + request.getParameter("address") +
request.getParameter("gender") + request.getParameter("marit-stat") + request.getParameter("dov"));
    // Execute SQL query
    PreparedStatement st = conn
        .prepareStatement("insert into od values(?, ?, ?, ?)");

    st.setInt(1, Integer.valueOf(request.getParameter("id")));
    st.setString(2, request.getParameter("date"));
    st.setString(3, request.getParameter("reason"));
    st.setString(4, "Applied");
    st.executeUpdate();

    // Close all the connections
    st.close();
    conn.close();

    // Get a writer pointer
    // to display the successful result

    out.println("<!DOCTYPE html><html lang=\"en\"><head><title>OD Applied!</title><meta
charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\"
type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/bootstrap/css/bootstrap.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="fonts/fontawesome-4.7.0/css/fontawesome.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/animate/animate.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/css-hamburgers/hamburgers.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/select2/select2.min.css"><link rel=\"stylesheet\" type=\"text/css\" href="css/util.css\"><link
rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div
class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img
src=\"img/portfolio-2.jpg\" alt=\"IMG!\"/></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px!\">OD Applied!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
onclick=\"location.href='Student_home.html'\" type=\"button\">Go
Home</button></div><br><br><br><br><br><br></form></div></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");
} catch (Exception e) {
    out.print("<p>Error</p>");
    e.printStackTrace();
} finally {
}

```

```

        // out.println("<form action='http://localhost:8080/PMS/ID' method='get'><button
type='submit'>View</button>");
        out.println("</body></html>");
    }
}
}

```

Insert_ticket

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class Insert_ticket extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";

    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";
    public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

        // Set response content type
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        //out.println("<html><head><title>Insert ticket</title></head><body>");

        try {
            // Register JDBC driver
            //Class.forName("com.mysql.cj.jdbc.Driver");
            Class.forName(JDBC_DRIVER);

            // Open a connection
            Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
            //out.println(Integer.valueOf(request.getParameter("id")) + request.getParameter("name")
+ Integer.valueOf(request.getParameter("age")));
            //out.println(request.getParameter("phone-no") + request.getParameter("address") +
request.getParameter("gender") + request.getParameter("marit-stat") + request.getParameter("dov"));
            // Execute SQL query
            PreparedStatement st = conn
                .prepareStatement("insert into ticket values(?, ?, ?)");

            st.setInt(1, Integer.valueOf(request.getParameter("id")));
            st.setString(2, request.getParameter("date"));
            st.setString(3, "Applied");
            st.executeUpdate();

            // Close all the connections
            st.close();
            conn.close();
        }
    }
}

```

```

        // Get a writer pointer
        // to display the successful result

        out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Ticket Raised!</title><meta
charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\"
type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"fonts/fontawesome-4.7.0/css/fontawesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link
rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div
class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img
src=\"img/portfolio-1.jpg\" alt=\"IMG\"></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px!\">Ticket Raised!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\"
onclick=\"location.href='Student_home.html'\" type=\"button\">Go
Home</button></div><br><br><br><br><br><br><br></form></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");

        } catch (Exception e) {
            out.print("<p>Error</p>");
            e.printStackTrace();
        } finally {
            // out.println("<form action='http://localhost:8080/PMS/ID' method='get'><button
type='submit'>View</button>\"");
            out.println("</body></html>");

        }
    }
}

```

MarkAttendanceServlet1

```

// Loading required libraries
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class MarkAttendanceServlet1 extends HttpServlet {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";

    public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

```

```

out.println("<!doctype html>\n" +
    "<html>\n" +
    "<head><title>MarkAttendance</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\">" +
    "</head>\n<body>\n" +
    "<div class=\"limiter\"><div class=\"container-login100\"><div class=\"card\" align=\"center;\" >" +
    "<form align=\"center\" style = \"width:80%;margin:auto;\" class=\"login100-form validate-form\" action=\"http://localhost:8080/Mini_Project/MarkAttendanceServlet2\" method=\"post\">" +
    "<!--<label for=\"date\">Date:</label>-->" +
    "<input class=\"input100\" type=\"date\" id=\"atddate\" name=\"atddate\"><br>\n" +
    "<span class=\"login100-form-title\" style=\"font-size: 30px;text-align:center;\">Mark Attendance:</span>\n" +
    "<table border = \"1\">\n" +
    "<tr>\n" +
    "<th style=\"width:50%;padding:10px 10px 10px 10px;\">Student ID</th>" +
    "<th style=\"width:50%;padding:10px 10px 10px 10px;\">Name</th>" +
    "<th style=\"width:50%;padding:10px 10px 10px 10px;\">Present or Absent</th>\n" +
    "</tr>\n");
}

// JDBC driver name and database URL

try {
    // Register JDBC driver
    Class.forName(JDBC_DRIVER);
    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    // Execute SQL query
    Statement stmt = conn.createStatement();
    String sql;

    sql = "Select id,name from login where role='Student'";
    ResultSet rs = stmt.executeQuery(sql);
    int i = 0;
    // Extract data from result set
    while (rs.next()) {
        //Retrieve by column name
        int id = rs.getInt("id");
        String name = rs.getString("name");

        //Display values
        out.println("<tr>\n" +
                    "<td style=\"padding:10px 10px 10px 10px;\">" + id + "</td>" +
                    "<td style=\"padding:10px 10px 10px 10px;\">" + name + "</td>" +
                    "<td style=\"padding:10px 10px 10px 10px;\">" +

```

```

        "<label><input type=\"radio\" name=\"id\" + i + \" value=\"yes\" required> Yes </label><br>" +
        "<label><input type=\"radio\" name=\"id\" + i + \" value=\"no\"> No </label>" +
        "</td>\n" +
        "</tr>\n");
    i = i + 1;
}
out.println("</table>\n");
out.println("<div class=\"container-login100-form-btn\"><input class=\"login100-form-btn\" type=\"submit\" value=\"Submit\"></div>");
out.println("</form></div></div></div><script src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script src=\"js/main1.js\"></script></body></html>");

// Clean-up environment
out.close();
rs.close();
stmt.close();
conn.close();
} catch (SQLException se) {
    //Handle errors for JDBC
    se.printStackTrace();
}
catch (Exception e) {
    //Handle errors for Class.forName
    e.printStackTrace();
}
}
}

```

MarkAttendanceServlet2

```
// Loading required libraries
import java.io.*;
import java.text.SimpleDateFormat;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class MarkAttendanceServlet2 extends HttpServlet {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String atddate = request.getParameter("atddate");
        response.setContentType("text/html");

```

```

PrintWriter out = response.getWriter();
// JDBC driver name and database URL

try {
    // Register JDBC driver
    Class.forName(JDBC_DRIVER);
    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    // Execute SQL query
    Statement stmt = conn.createStatement();
    String sql;

    sql = "Select id,name from login where role='Student'";
    ResultSet rs = stmt.executeQuery(sql);
    PreparedStatement st;

    int i = 0;
    // Extract data from result set
    while (rs.next()) {
        //Retrieve by column name
        int id = rs.getInt("id");
        String name = rs.getString("name");

        String pa = request.getParameter("id" + i);

        st = conn.prepareStatement("insert into attendance values(?, ?, ?, ?)");

        st.setString(2, atddate);
        st.setInt(1, id);
        st.setString(3, name);
        st.setString(4, pa);

        st.executeUpdate();

        i = i + 1;
    }
    // Clean-up environment
    rs.close();
    stmt.close();
    //st.close();
    conn.close();
    out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Attendance Marked</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img src=\"img/portfolio-3.jpg\" alt=\"IMG\"/></div><form class=\"login100-form validate-form\" action=\"\"><span class=\"login100-form-title\" style=\"font-size: 50px!>Attendance Recorded!</span><div"

```

```
class="container-login100-form-btn"><button class="login100-form-btn"
onclick="location.href='Teacher_home.html'" type="button">Go
Home</button></div><br><br><br><br><br><br><br><br><br><br></form></div></div></div><script
src="vendor/jquery/jquery-3.2.1.min.js"></script><script src="vendor/bootstrap/js/popper.js"></script><script
src="vendor/bootstrap/js/bootstrap.min.js"></script><script src="vendor/select2/select2.min.js"></script><script
src="vendor/tilt/tilt.jquery.min.js"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src="js/main1.js"></script>");

} catch (SQLException se) {
    //Handle errors for JDBC
    se.printStackTrace();
}

catch (Exception e) {
    //Handle errors for Class.forName
    e.printStackTrace();
}

}
```

Validateloginstudent

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class validateLoginStudent extends HttpServlet {
    private static final long serialVersionUID = 1L;
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    static final String USER = "root";
    static final String PASS = "mysql";

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
```

```

// Connect to database
Class.forName(JDBC_DRIVER);
// Open a connection
Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);

// Prepare statement to select user from database
PreparedStatement ps = conn.prepareStatement("SELECT * FROM login WHERE role='Student' AND
username=? AND password=? ");
ps.setString(1, request.getParameter("username"));
ps.setString(2, request.getParameter("password"));

// Execute query and get result set
ResultSet rs = ps.executeQuery();

// If user is found, create session and redirect to home page
if (rs.next()) {
    PrintWriter out = response.getWriter();
    HttpSession session = request.getSession();
    session.setAttribute("username", request.getParameter("username"));
    session.setAttribute("id", request.getParameter("id"));

    out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Login Success!</title><meta
charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\"
type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/bootstrap/css/bootstrap.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="fonts/font-awesome-4.7.0/css/font-awesome.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/animate/animate.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/css-hamburgers/hamburgers.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/select2/select2.min.css"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link
rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div
class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img
src=\"img/portfolio-3.jpg\" alt=\"IMG\"></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px!\">Login Success!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
onclick=\"location.href='Student_home.html'\"
type=\"button\">Go</button></div><br><br><br><br><br><br></form></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");

} else {
    // If user is not found, display error message
    PrintWriter out = response.getWriter();
    out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Login Failed!</title><meta
charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\""
type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/bootstrap/css/bootstrap.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="fonts/font-awesome-4.7.0/css/font-awesome.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/animate/animate.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/css-hamburgers/hamburgers.min.css"><link rel=\"stylesheet\" type=\"text/css\""
href="vendor/select2/select2.min.css"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link
rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div
class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img
src=\"img/portfolio-3.jpg\" alt=\"IMG\"></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px!\">Login Failed!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
onclick=\"location.href='Student_home.html'\"
type=\"button\">Go</button></div><br><br><br><br><br><br></form></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");

}

```

```

src=\"img/portfolio-3.jpg\" alt=\"IMG\"></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px\>Login Failed!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\"
onclick=\"location.href='Student_login.html'\" type=\"button\">Try
Again</button></div><br><br><br><br><br><br></form></div></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");

}

// Close resources
rs.close();
ps.close();
conn.close();

} catch (ClassNotFoundException | SQLException e) {
    e.printStackTrace();
}
}

}

}

```

Validateloginteacher

```

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class validateloginteacher extends HttpServlet {
    private static final long serialVersionUID = 1L;
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    static final String USER = "root";
    static final String PASS = "mysql";

    protected void doGet(HttpServletRequest request, HttpServletResponse response)

```

```

throws ServletException, IOException {

try {
    // Connect to database
    Class.forName(JDBC_DRIVER);
    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);

    // Prepare statement to select user from database
    PreparedStatement ps = conn.prepareStatement("SELECT * FROM login WHERE role='Teacher' AND
username=? AND password=? ");
    ps.setString(1, request.getParameter("username"));
    ps.setString(2, request.getParameter("password"));

    // Execute query and get result set
    ResultSet rs = ps.executeQuery();

    // If user is found, create session and redirect to home page
    if (rs.next()) {
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Login Success!</title><meta
charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\"
type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link
rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div
class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img
src=\"img/portfolio-3.jpg\" alt=\"IMG\"></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px;\">>Login Success!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
onclick=\"location.href='Teacher_home.html'\"
type=\"button\">Go</button></div><br><br><br><br><br><br></form></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$( '.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");

    } else {
        // If user is not found, display error message
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Login Failed!</title><meta
charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\""
type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\""
href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link
rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div
class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img
src=\"img/portfolio-3.jpg\" alt=\"IMG\"></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px;\">>Login Failed!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\""
onclick=\"location.href='Teacher_home.html'\"
type=\"button\">Go</button></div><br><br><br><br><br><br></form></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$( '.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");

    }
}

```

```
src="img/portfolio-3.jpg" alt="IMG"></div><form class="login100-form validate-form" action=""><span class="login100-form-title" style="font-size: 50px">Login Failed!</span><div class="container-login100-form-btn"><button class="login100-form-btn" onclick="location.href='Teacher_login.html'" type="button">Try Again</button></div><br><br><br><br><br><br><br></form></div></div><script src="vendor/jquery/jquery-3.2.1.min.js"></script><script src="vendor/bootstrap/js/popper.js"></script><script src="vendor/bootstrap/js/bootstrap.min.js"></script><script src="vendor/select2/select2.min.js"></script><script src="vendor/tilt/tilt.jquery.min.js"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script src="js/main1.js"></script>">
```

}

```
// Close resources
```

```
rs.close();
```

```
ps.close();
```

```
conn.close();
```

```
} catch (ClassNotFoundException | SQLException e) {
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

ViewOdServlet1

```
// Loading required libraries
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.sql.Date;

public class ViewOdServlet1 extends HttpServlet {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("<!doctype html>\n" +
                   "<html>\n" +
                   "  <head><title>Mark OD</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"css/style.css\"/>");    }
}
```

```

href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\">" +
        "</head>\n<body>\n" +
        "<div class=\"limiter\"><div class=\"container-login100\"><div class=\"card\" align=\"center;\" >" +
        "<form align=\"center\" style = \"width:80%;margin:auto;\" class=\"login100-form validate-form\" action=\"http://localhost:8080/Mini_Project/ViewOdServlet2\" method=\"post\">" +
        "    <span class=\"login100-form-title\" style=\"font-size: 30px;text-align:center;\">Verify OD</span>\n" +
        "    <table border = \"1\">\n" +
        "        <tr>\n" +
        "            <th style=\"width:50%;padding:10px 10px 10px 10px;\">Student ID</th>" +
        "            <th style=\"width:50%;padding:10px 10px 10px 10px;\">Name</th>" +
        "            <th style=\"width:50%;padding:10px 10px 10px 10px;\">Date</th>" +
        "            <th style=\"width:50%;padding:10px 10px 10px 10px;\">Reason</th>" +
        "            <th style=\"width:50%;padding:10px 10px 10px 10px;\">Accept or Reject</th>\n" +
        "        </tr>\n" +
    );
}

// JDBC driver name and database URL

try {
    // Register JDBC driver
    Class.forName(JDBC_DRIVER);
    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    // Execute SQL query
    Statement stmt = conn.createStatement();
    String sql;

    sql = "Select * from od where status=\"Applied\"";
    ResultSet rs = stmt.executeQuery(sql);

    int i = 0;
    // Extract data from result set
    while (rs.next()) {
        //Retrieve by column name
        int id = rs.getInt("id");
        Date date = rs.getDate("od_date");
        String reason = rs.getString("reason");

        Statement stmt2 = conn.createStatement();
        ResultSet rss = stmt2.executeQuery("Select name from login where id = " + id);

        if (rss.next()) {
            String name = rss.getString("name");
            //Display values
            out.println("<tr>\n" +
                "<td style=\"padding:10px 10px 10px 10px;\">" + id + "</td>" +
                "<td style=\"padding:10px 10px 10px 10px;\">" + name + "</td>" +
                "<td style=\"padding:10px 10px 10px 10px;\">" + date + "</td>" +
                "<td style=\"padding:10px 10px 10px 10px;\">" + reason + "</td>" +
                "<td style=\"padding:10px 10px 10px 10px;\">" +

```

```

        "<label><input type=\"radio\" name=\"id\" + i + \" value=\"Yes\" required>Accept</label>" +
        "<label><input type=\"radio\" name=\"id\" + i + \" value=\"No\">Reject</label>" +
        "</td>\n" +
        "</tr>"
    );
}

rss.close();
i = i + 1;
}
out.println("</table>\n");
out.println("<div class=\"container-login100-form-btn\"><input class=\"login100-form-btn\" type=\"submit\" value=\"Submit\"></div>");
out.println("</form></div></div></div><script src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script src=\"js/main1.js\"></script></body></html>");

// Clean-up environment
out.close();
rs.close();
stmt.close();
conn.close();
} catch (SQLException se) {
    //Handle errors for JDBC
    se.printStackTrace();
}
}

catch (Exception e) {
    //Handle errors for Class.forName
    e.printStackTrace();
}
}
}
}

```

ViewOdServlet2

```

// Loading required libraries
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.sql.Date;

public class ViewOdServlet2 extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

```

```

response.setContentType("text/html");
PrintWriter out = response.getWriter();
try {
    // Register JDBC driver
    Class.forName(JDBC_DRIVER);
    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    // Execute SQL query
    Statement stmt = conn.createStatement();
    String sql;

    sql = "Select id,od_date from od where status='Applied'";
    ResultSet rs = stmt.executeQuery(sql);
    PreparedStatement st, sto;

    int i = 0;
    // Extract data from result set
    while (rs.next()) {
        //Retrieve by column name
        int id = rs.getInt("id");
        Date date = rs.getDate("od_date");

        String pa = request.getParameter("id" + i);

        st = conn.prepareStatement("UPDATE attendance SET present = ? WHERE id = ? AND curr_date = ?");
        st.setString(1, pa);
        st.setInt(2, id);
        st.setDate(3, date);

        st.executeUpdate();

        sto = conn.prepareStatement("UPDATE od SET status = ? WHERE id = ? AND od_date = ?");
        if (pa.equals("Yes") == true) {
            sto.setString(1, "Accepted");
            Mailer.send("padmapooja2010291@ssn.edu.in", "Regarding OD", "Accepted");
        }
        else {
            sto.setString(1, "Rejected");
            Mailer.send("padmapooja2010291@ssn.edu.in", "Regarding OD", "Rejected");
        }

        sto.setInt(2, id);
        sto.setDate(3, date);

        sto.executeUpdate();

        i = i + 1;
    }
    // Clean-up environment
    rs.close();
    stmt.close();
    conn.close();
}

```

```

        out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Attendance Updated</title><meta
charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\"
type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\"
href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\"
href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\"
href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\"
href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\"
href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link
rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"></head><body><div class=\"limiter\"><div
class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img
src=\"img/portfolio-3.jpg\" alt=\"IMG\"/></div><form class=\"login100-form validate-form\" action=\"\"><span
class=\"login100-form-title\" style=\"font-size: 50px!>Attendance Updated!</span><div
class=\"container-login100-form-btn\"><button class=\"login100-form-btn\"
onclick=\"location.href='Teacher_home.html'\" type=\"button\">Go
Home</button></div><br><br><br><br><br><br></form></div></div></div><script
src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script
src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script
src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script
src=\"js/main1.js\"></script>");

    } catch (SQLException se) {
        //Handle errors for JDBC
        se.printStackTrace();
    }

    catch (Exception e) {
        //Handle errors for Class.forName
        e.printStackTrace();
    }
}
}
}

```

ViewTicketServlet1

```

// Loading required libraries
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.sql.Date;

public class ViewTicketServlet1 extends HttpServlet {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    // Database credentials
    static final String USER = "root";
    static final String PASS = "mysql";

    public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

```

```

out.println("<!doctype html>\n" +
    "<html>\n" +
    "<head><title>Verify Ticket</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/font-awesome-4.7.0/css/font-awesome.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/animate/animate.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\">" +
    "</head>\n<body>\n" +
    "<div class=\"limiter\"><div class=\"container-login100\"><div class=\"card\" align=\"center;\" >" +
    "<form align=\"center\" style = \"width:80%;margin:auto;\" class=\"login100-form validate-form\" action=\"http://localhost:8080/Mini_Project/ViewTicketServlet2\" method=\"post\">" +
        "<span class=\"login100-form-title\" style=\"font-size: 30px;text-align:center;\">Verify Ticket</span>\n" +
        "<table border = \"1\">\n" +
        "<tr>\n" +
        "<th style=\"width:50%;padding:10px 10px 10px 10px;\">Student ID</th>" +
        "<th style=\"width:50%;padding:10px 10px 10px 10px;\">Name</th>" +
        "<th style=\"width:50%;padding:10px 10px 10px 10px;\">Date</th>" +
        "<th style=\"width:50%;padding:10px 10px 10px 10px;\">Accept or Reject</th>\n" +
        "</tr>\n" +
    );
}

// JDBC driver name and database URL

try {
    // Register JDBC driver
    Class.forName(JDBC_DRIVER);
    // Open a connection
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
    // Execute SQL query
    Statement stmt = conn.createStatement();
    String sql;

    sql = "Select * from ticket where status=\"Applied\"";
    ResultSet rs = stmt.executeQuery(sql);

    int i = 0;
    // Extract data from result set
    while (rs.next()) {
        //Retrieve by column name
        int id = rs.getInt("id");
        Date date = rs.getDate("ticket_date");

        Statement stmt2 = conn.createStatement();
        ResultSet rss = stmt2.executeQuery("Select name from login where id = " + id);

        if (rss.next()) {
            String name = rss.getString("name");
            //Display values

```

```

        out.println("<tr>\n" +
            "<td style=\"padding:10px 10px 10px 10px;\">" + id + "</td>" +
            "<td style=\"padding:10px 10px 10px 10px;\">" + name + "</td>" +
            "<td style=\"padding:10px 10px 10px 10px;\">" + date + "</td>" +
            "<td style=\"padding:10px 10px 10px 10px;\">" +
            "<label><input type=\"radio\" name=\"id\" " + i + "\" value=\"Yes\" required>Accept</label>" +
            "<label><input type=\"radio\" name=\"id\" " + i + "\" value=\"No\">Reject</label>" +
            "</td>\n" +
            "</tr>"
        );
    }

    rss.close();
    i = i + 1;
}
out.println("</table>\n");
out.println("<div class=\"container-login100-form-btn\"><input class=\"login100-form-btn\" type=\"submit\" value=\"Submit\"/></div>");
out.println("</form></div></div></div><script src=\"vendor/jquery/jquery-3.2.1.min.js\"></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script src=\"js/main1.js\"></script></body></html>");
// Clean-up environment
out.close();
rs.close();
stmt.close();
conn.close();
} catch (SQLException se) {
    //Handle errors for JDBC
    se.printStackTrace();
}
}

catch (Exception e) {
    //Handle errors for Class.forName
    e.printStackTrace();
}
}
}
}

```

ViewTicketServlet2

```

// Loading required libraries
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.sql.Date;

public class ViewTicketServlet2 extends HttpServlet {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mini_project";
    // Database credentials
    static final String USER = "root";

```

```

static final String PASS = "mysql";

public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    try {
        // Register JDBC driver
        Class.forName(JDBC_DRIVER);
        // Open a connection
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
        // Execute SQL query
        Statement stmt = conn.createStatement();
        String sql;

        sql = "Select id,ticket_date from ticket where status='Applied'";
        ResultSet rs = stmt.executeQuery(sql);
        PreparedStatement st, sto;

        int i = 0;
        // Extract data from result set
        while (rs.next()) {
            //Retrieve by column name
            int id = rs.getInt("id");
            Date date = rs.getDate("ticket_date");

            String pa = request.getParameter("id" + i);

            st = conn.prepareStatement("UPDATE attendance SET present = ? WHERE id = ? AND curr_date = ?");
            st.setString(1, pa);
            st.setInt(2, id);
            st.setDate(3, date);

            st.executeUpdate();

            sto = conn.prepareStatement("UPDATE ticket SET status = ? WHERE id = ? AND ticket_date = ?");

            if (pa.equals("Yes") == true) {
                sto.setString(1, "Accepted");
                // Mailer.send("padmapooja2010291@ssn.edu.in", "Regarding OD", "Accepted");
            }
            else {
                sto.setString(1, "Rejected");
                // Mailer.send("padmapooja2010291@ssn.edu.in", "Regarding OD", "Rejected");
            }

            sto.setInt(2, id);
            sto.setDate(3, date);

            sto.executeUpdate();

            i = i + 1;
        }
    }
}

```

```

        }
        // Clean-up environment
        rs.close();
        stmt.close();
        conn.close();
        out.println("<!DOCTYPE html><html lang=\"en\"><head><title>Attendance Updated</title><meta charset=\"UTF-8\"><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"><link rel=\"icon\" type=\"image/png\" href=\"images/icons/favicon.ico\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/bootstrap/css/bootstrap.min.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"fonts/fontawesome-4.7.0/css/fontawesome.min.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/animate/animate.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/css-hamburgers/hamburgers.min.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"vendor/select2/select2.min.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"css/util.css\"/><link rel=\"stylesheet\" type=\"text/css\" href=\"css/main.css\"/></head><body><div class=\"limiter\"><div class=\"container-login100\"><div class=\"wrap-login100\"><div class=\"login100-pic js-tilt\" data-tilt><img src=\"img/portfolio-3.jpg\" alt=\"IMG\"/></div><form class=\"login100-form validate-form\" action=\"\"><span class=\"login100-form-title\" style=\"font-size: 50px!\">Attendance Updated!</span><div class=\"container-login100-form-btn\"><button class=\"login100-form-btn\" onclick=\"location.href='Teacher_home.html'\" type=\"button\">Go Home</button></div><br><br><br><br><br><br><br></form></div></div></div><script src=\"vendor/jquery/jquery-3.2.1.min.js\"/></script><script src=\"vendor/bootstrap/js/popper.js\"></script><script src=\"vendor/bootstrap/js/bootstrap.min.js\"></script><script src=\"vendor/select2/select2.min.js\"></script><script src=\"vendor/tilt/tilt.jquery.min.js\"></script><script >$('.js-tilt').tilt({scale: 1.1})</script><script src=\"js/main1.js\"></script>");

    } catch (SQLException se) {
        //Handle errors for JDBC
        se.printStackTrace();
    }

    catch (Exception e) {
        //Handle errors for Class.forName
        e.printStackTrace();
    }
}
}

```

**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 11

Developing Test Plan & Test Cases

Develop Test Plan and Test Cases for the chosen application. [CO4, K2 & K3]

A sample Test Plan and tabulated Test Cases for Agritech System – Leafage application are attached for your reference.

Developing Test Plan & Test Cases

Ex No :

Date :

Aim:

To develop test plans and test cases for the Attendance Management System (AMS) and try to improve the design based on the results of testing.

Identification of Testing Scenarios:

1. Check if admin can log in successfully
2. Check if the admin can add a new student to the system
3. Check if the admin can add a new teacher to the system
4. Check if a student can be authenticated based on login credentials
5. Check if a teacher can be authenticated based on login credentials
6. Check if a student can view attendance
7. Check if a student can apply OD
8. Check if a student can apply ticket
9. Check if a student can view OD status
10. Check if a student can view ticket status
11. Check if a teacher can mark attendance
12. Check if a teacher can view attendance
13. Check if a teacher can accept/ reject OD
14. Check if a teacher can accept/ reject ticket

Tabulate the Test Cases:

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Outcome	Actual Outcome	Pass/Fail
T01	Admin authentication	Go to admin page and enter login credentials: username & password	Username: root Password: 123456789	Valid	Valid	Pass
T02	Admin authentication	Go to admin page and enter login credentials: username & password	Username: root Password: 987654321	Invalid (incorrect password)	Invalid	Pass
T03	Create new student	Admin should be able to add student info	User: Student ID: 20001 Name: Keshavinee Username: Kesh Password: 5485@Ram	Valid	Valid	Pass
T04	Create new student	Admin should be able to add student info	User: Student ID: 20002 Name: Keshavinee Username: Kesh Password: 5485@Ram	Invalid (Username already exists)	Valid	Fail
T05	Create new teacher	Admin should be able to add teacher info	User: Teacher ID: 50001 Name: Kamala Ramesh Username:	Valid	Valid	Pass

			Kamala Password: kamal#23			
T06	Create new teacher	Admin should be able to add teacher info	User: Teacher ID: 50002 Name: Kamala Ramesh Username: Kamala Password: kamal#23	Invalid (Username already exists)	Valid	Fail
T07	Student authentication	Go to student page and enter login credentials: username & password	Username: Kesh Password: 5485@Ram	Valid	Valid	Pass
T08	Student authentication	Go to student page and enter login credentials: username & password	Username: Keshav Password: 5485@Ram	Invalid (incorrect Username)	Invalid	Pass
T09	Teacher authentication	Go to teacher page and enter login credentials: username & password	Username: Kamala Password: kamal#23	Valid	Valid	Pass
T10	Teacher authentication	Go to teacher page and enter login credentials: username & password	Username: Kamala Password: kamala#23	Invalid (incorrect password)	Invalid	Pass
T11	View Attendance	Student should be able to view	Click View Attendance button	Valid	Valid	Pass

		his/ her attendance				
T12	Apply OD	Student should click Apply for OD button & enter OD credentials	ID: 20001 Date: 05/05/2023 Reason: Culturals	Valid	Valid	Pass
T13	Apply OD	Student should click Apply for OD button & enter OD credentials	ID: 20001 Date: 20/06/2024 Reason: Culturals	Invalid (OD applied for date where student is already present)	Valid	Fail
T14	Apply Ticket	Student should click Raise Ticket button & enter Ticket credentials	ID: 20001 Date: 06/05/2023	Valid	Valid	Pass
T15	Apply Ticket	Student should click Raise Ticket button & enter Ticket credentials	ID: 20001 Date: 20/06/2024	Invalid (Ticket applied for date where attendance is not taken)	Valid	Fail
T16	View OD	Student should be able to view his/ her OD status	Click Check OD Status button	Valid	Valid	Pass
T17	View Ticket	Student should be able to view his/ her ticket status	Click Check Ticket Status button	Valid	Valid	Pass
T18	Mark	Teacher	Date: 05/05/2023	Valid	Valid	Pass

	Attendance	should be able to mark attendance	Student ID: 20001 Name: Keshavinee Present or Absent: Yes			
T19	Mark Attendance	Teacher should be able to mark attendance	Date: 20/06/2024 Student ID: 20001 Name: Keshavinee Present or Absent: Yes	Invalid (Marked attendance for date exceeding current date)	Valid	Fail
T20	View Attendance	Teacher should be able to view entire attendance	Click View Attendance button	Valid	Valid	Pass
T21	Verify OD	Teacher should be able to click View OD	Accept/ Reject OD	Valid	Valid	Pass
T22	Verify Ticket	Teacher should be able to click View Ticket	Accept/ Reject Ticket	Valid	Valid	Pass

**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
UCS1617- Mini Project**

A. NO. : 12

Applying Suitable Design Patterns

Apply the suitable design patterns for the chosen application to refine the implementation.
[CO5, K2 & K3]

Consider the following design patterns for refinement.

- Responsibility Driven Design (RDD)
- GRASP
 - Creator
 - Information Expert
 - Controller
 - Low coupling
 - High Cohesion
- GOF Patterns
 - Creational
 - Factory
 - Structural
 - Bridge
 - Adapter
 - Behavioral
 - Strategy
 - Observer

Applying Suitable Design Patterns

Ex No :

Date :

Aim:

To apply the suitable design patterns for the Attendance Management System (AMS) to refine the implementation.

Factory Pattern:

- Problem: Who should be responsible for creating families of related objects
- Solution: Use a factory object to handle all creation responsibility
- This pattern provides one of the best ways to create an object.
- Defines an interface for creating objects but let sub-classes decide which of those instantiate.
- Enables the creator to defer Product creation to a subclass.
- If some domain object (for example Sale) creates them as the creator pattern suggests, we will encounter the following problems:
 - The domain objects Sale must be aware of external systems (coupling)
 - Adding or removing an external calculator will affect the sale
 - Change in rules about adaptor usage will affect sale
 - This responsibility lowers the cohesion of the domain object because connectivity with external components is not its main job (separation of concerns)

Refined Class Diagram:

