UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE)

KONAM, NAGERCOIL – 629 004

RECORD NOTE BOOK

CCS356-OBJECT ORIENTED SOFTWARE ENGINEERING

REGISTER NO :

NAME :

YEAR/SEMESTER :

DEPARTMENT :

UNIVERSITY COLLEGE OF ENGINEERINGNAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE)

KONAM, NAGERCOIL – 629 004

Register No:

Certified that, this is the bonafide record of work done by ***Mr./Ms.*** *…… of VI Semester in Computer Science and Engineering of this college, in the* ***CCS356 – OBJECT ORIENTED SOFTWARE ENGINEERING*** *During the academic year 2023-2024 in partial fulfillment of the requirements of the B.E Degree course of the Anna University Chennai.*

Staff-in-charge **Head of the Department**

This record is submitted for the University Practical Examination held on .....

Internal Examiner **External Examiner**

INDEX

Exp No	Date	Title	Page	Sign
1.		Identify a software system that needs to be developed		
2.		Document the Software Requirements Specification (SRS) for the Student Information System		
3.		Identify the use cases and the develop the Use Case model		
4.		Identify the conceptual classes and develop a Domain Model and also derive a Class diagram for the Student Information system		
5.		Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration diagram		
6.		Draw relevant State Chart and Activity Diagram		
7.		Implement the system as per the detailed design		
8.		Test the software system for all the scenarios identified as per the use case diagram		
9.		Improve the reusability and maintainability of thesoftware system by applying appropriate design patterns		

Ex no:01 DATE:	**To identify a software system that needs to be developed – Student Information System**

Aim:

To identify a software system that needs to be developed –Student Information System

Introduction:

A Student Information System (SIS) is a System that manages the records of student regarding admission and examination part. A Student Information System(SIS) is designed to help collages for management of dental student. Extensive information is available at your fingertips though this System. Viewing student data, managing admission and reshuffling, managing seats, quota, board, semester, faculty, category and for examination, block allocation, subject management, scheduling exam, result and related issues are made simple and easy. There are custom search capabilities to aid in finding student information and working on student records. This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources. SMS allows the keeping of personnel data in a form that can be easily accessed and analyzed in a consistent way.

Features of Student Information System:

1. **Performance :**

   The Student Management System shall be built upon the web development technique and put on the web server online. The system and the server must be capable of handling the real-time error functionality occurs by the defined users. In addition, the system must be safety critical. All failures reported by the server side must be handled instantaneously to allow for user and system safety.

2. **Reliability :**

   The system is safety critical. If it moves out of normal operation mode, the requirement to drop or down the server and fix it as soon as possible and open it again. This emergency behaviour shall not occur without reason.

3. **Availability** :

   When in normal operating conditions, request by a user for an online system shall be handled within 1 second. Immediate feedback of the systems activities shall be communicated to the user by clearing the system and giving space n speed to their hospitality.

4. **Security** :

   There shall be a strong security mechanism should be place in the server side of the system to keep unwanted users to hack or damage the system. However, all users of the system

give and store the details of privacy related to 14 personal information and many other. However, our system can be accessed online so we need very secured system as far as security is concerned.

5. **Maintainability:**

   There shall be design documents describing maintenance of the software and database used to save the user details as well as the daily updated and modification done in system. There shall be an access on the control system by the admin to maintained it properly at the front end as well as at back end.

6. **Portability:**

   There is portability requirement as far as our system is concern because it is an online as well as offline (local server based) system so we can access it from anywhere through the internet connection. And we have to maintain the copy of stored data into our database.

7. **Relational Database Schema:**

   The relational database schema effectively organizes and manages student-related data, including personal information, academic records, course details, and administrative data.

Result:

Thus the software system that is need to be developed for Student Information system was executed successfully.

EX no:02 DATE:	DOCUMENT THE SOFTWARE REQUIREMENTS SPECIFICATION(SRS) FOR THE IDENTIFIED SYSTEM

Aim:

To implement a software Requirement Specification for Student Information System (SIS).

Problem Statement:

1. The Student Information System (SIS) currently in use struggles with manual processes and resource-intensive tasks, hindering the efficient management of student data across educational institutions.

2.The central objective of the system is to streamline the students information management process by enabling online registration students fill out registration form containing personal details ,academic information and other relevant data.The system verifies the authenticity of this information against existing records in the database.

3.The initial verification is a crucial step in the student context,the system validates the provided information against the database to ensure accuracy and reliability.

4.Upon successful verification, the system follows the student information to the respective academic departments for further manual processing ,Administrator use the systems report to evaluate the applications and any discrepancies may result in necessary actions or penalties in accordance with academic policies .

5.Once all verification processes are complete and all criteria are met ,the students information is permanently added to the database subsequently ,the student information system facilitates the insurances of necessary documents, such as identification cards or transcripts to the student.

6.The problem statement outlines the challenges faced by the current student information system and emphazies the need for a more efficient and automated solution to enhance the management of student data across various academic processes.

Software Requirement Specification:

Introduction:

A Student Information System (SIS) is a software application that manages student data and academic administration in educational institutions. It handles tasks such as student records management, course management, academic administration, instructor/staff management, communication, reporting, and integration with other systems. SIS streamlines processes, enhances student experiences, and supports institutional effectiveness.

Purpose:

The project aims at the following matters:

- Automation of admission and enrolment as per board, quota, category and available seats.
- Assistance in decision-making.
- To manage information of student, faculty and courses.
- Consistently update information of all the students.

Document Convention:

	Font	**Style**	**Size**
Heading	Times New Roman	Bold	18
Sub-Heading	Times New Roman	Bold	14
Other's	Times New Roman	Regular	13

Intended Audience and Reading Suggestions:

This SRS is mainly developed for the student information. In this there are Administrators ,Teachers ,Students and Parents.

Administrators: Manage overall system functionality and access all features.

Teachers: Input and update student grades, attendance, and communicate with students.

Students: Access and update personal information, view academic records, and communicate with faculty.

Parents: Monitor student performance and receive communication from the system.

Product Scope:

- The Student Information System (SIS) needs to manage student data and academic records effectively.
- It should support multiple educational institutions managing their student population.
- Data security measures must be implemented to prevent unauthorized access to student records.
- The system should facilitate efficient handling of student inquiries and requests, ensuring quick response times.
- Clear process flows should be incorporated to track the status of student records, such as admissions, enrollment, grades, and financial aid, at any given time.

Definitions, Acronyms and the Abbrevations:

References:

SIS: Student Information System

UI: User Interface

API: Application Programming Interface

Technologies to be used:

- HTML
- JSP
- Javascript
- Java

Tools to be used:

- Star UML

Overview:

- SRS includes two sections overall description and specific requirements –
- Overall Description will describe major role of the system components and inter-connections.
- Specific Requirements will describe roles & functions of the actors.

Overall Description:

Product Perspective:

The various system tools that have been used in developing both the front end, back end and other tools of the project are being discussed in this section.

Software Interface:

- **Front End Client** -The back end is implemented using MYSQL which is used to design the databases.
- **Back End**- MySQL is the world's second most widely used open source relational database management system (RDMS). The SQL phrase stands for structured query.

Hardware Interface:

All components able000 to be executed on personal computers with Windows OS platforms and other platforms like Linux, Unix.

System Function:

• The Student Information System (SIS) is designed as a comprehensive web-based solution tailored to the needs of educational institutions.

• It allows institutions to manage student data and academic records efficiently.

• Students can access the system to view their academic information, register for courses, and interact with administrative processes.

• The system supports multiple campuses or locations of the educational institution.

• Features include automated document storage, where administrative documents are automatically uploaded to the system's server.

• An automatic uploader module allows for scheduled uploads of documents from various locations.

• Documents are routed to specific users, such as administrators or academic advisors, based on predefined rules set by the system administrator.

• Data entry personnel and quality check users can access documents through a web-based interface to perform their respective tasks efficiently.

User Characteristics:

The user profiles identified to have interaction with Student

Management System that anyone can register and login into system and us the required resources. The students can easily fill up the registration form online and submit it. And the admin will check the details that is the student is eligible as per the admission criteria. After the student will successfully registered he can use college/school system environments as per their limits decided by admin.

Constraints:

Student Information System can be accessed successfully by any client location and it's not necessary that every registration is genuine, so there is chances of fake registration that may reflect some errors. So the system is designed such a way that the database will keep updated by administrator and there is better security options available on the server that can prevent fake IP addresses to access system

Assumptions and Dependencies:

- Students, faculty, and administrative staff are assumed to have basic computer literacy and proficiency in the English language to interact with the Student Information System (SIS).
- The system must prioritize privacy and security measures to safeguard student information and sensitive data from unauthorized access or breaches.

System Features:

Student Registration:

Allow new students to fill out registration forms with personal and academic information.

User Authentication and Authorization:

Implement secure user authentication and role-based access control.

Student Profile Management:

Enable students to update personal information, contact details, and profile pictures.

Academic Record Tracking

Record and update student grades, courses, and transcripts.

Attendance Management

Track and manage student attendance for various courses.

Communication Platform

Facilitate communication between students, faculty, and parents through announcements and messages.

Enrollment and Registration

Simplify the enrollment process, allowing students to register for courses and pay fees online.

Reporting and Analytics

Generate reports and analytics for administrators to assess student performance and system usage.

System Requirements:

System requirements for the Student Information System (SIS) specify the capabilities and constraints necessary to meet the needs of users and stakeholders.

Other Requirements:

Legal and Regulatory Requirements

Comply with data protection laws and regulations.

Ethical Requirements

Ensure the ethical use of student data and information.

Privacy and Data Protection

Implement measures to protect the privacy and confidentiality of student information.

Non-functional Requirements:

Performance Requirements

The system should respond to user actions within [define response time] seconds.

Security Requirements

Implement secure user authentication and data encryption to protect user information.

Reliability and Availability

The system should have a minimum uptime of 99% for availability.

Maintainability

Provide a mechanism for easy system updates and maintenance.

Portability

The system should be accessible across different devices and screen sizes.

RESULT:

Thus the implementation of software Requirement Specification for Student Information System (SIS) was implemented successfully

Ex no:03 DATE:	To Identify the Use Cases and Develop the Use Case Model

Aim:

To identify the use cases and develop use case model.

Introduction of Use Case Diagram:

A use case diagram is a type of behavioral diagram in the Unified Modeling Language (UML) that represents the interactions between a system and its actors (users or external systems) in terms of the system's behavior. It provides a graphical overview of the functionalities provided by a system and the actors involved in those functionalities. Use case diagrams are widely used in software engineering to capture the requirements of a system and to visualize the high-level system structure.

Purpose of Use Case Diagram:

The purpose of a use case diagram is to visually represent the functional requirements of a system from the perspective of its users or external systems. It helps in understanding the system's behavior by illustrating how users interact with the system to accomplish specific tasks or goals. Use case diagrams facilitate communication between stakeholders, including developers, designers, and clients, by providing a clear and concise representation of the system's functionality.

Components of Use Case Diagram:

Actors:

Actors represent the users or external systems interacting with the system being modeled. In the Student Information System, the actors include the student, Teacher and Admin.

Use Cases:

Use cases represent the functionalities or services provided by the system. Each use case describes a specific interaction between an actor and the system to achieve a particular goal. In the Student Information system, examples of use cases include "Fill admission" ,"Pay fees," "Get enrolled," "Mark attendance," "Prepare test paper," "Declare result," and "Payment."

Relationships:

Relationships, such as associations and dependencies. Associations depict the connections between actors and use cases. For example, students interacts with use cases like course registration, grade viewing, and transcript requests. Dependencies represent the reliance of one use case on another. For instances, course registration may depend on course availability, while grade calculation depends on completed coursework.

Benefits of Use Case Diagram:

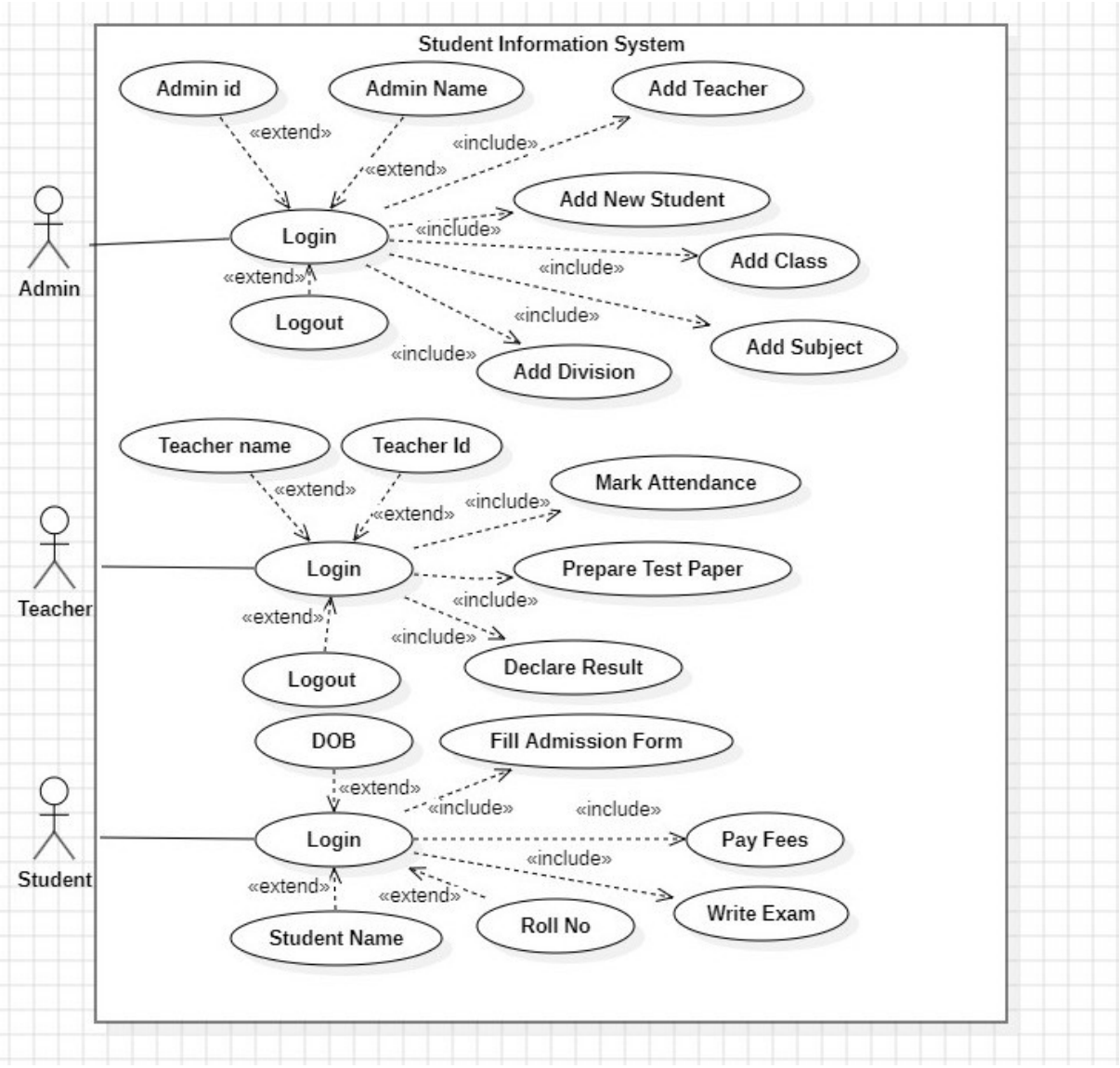
Requirements Elicitation: Use case diagrams help in eliciting and defining the functional requirements of a system by identifying the interactions between users and the system.

Visualization: Use case diagrams provide a visual representation of the system's functionality, making it easier for stakeholders to understand and analyze the system's behavior.

Communication: Use case diagrams serve as a communication tool between stakeholders, including developers, designers, and clients, by providing a common understanding of the system's requirements and behavior.

Requirements Verification: Use case diagrams can be used to verify whether all the necessary functionalities and interactions have been captured and addressed in the system design.

Use Case Diagram of Student Information System:



Modify/Delete:

- In a Student Information System (SIS) use case diagram, "Modify" and "Delete" actions represent key functionalities for managing student records.
- "Modify" allows authorized users, typically administrative staff, to update existing student information such as personal details or academic records.
- This ensures data accuracy and reflects any changes in student status. Conversely, "Delete" enables authorized users to remove student records from the system when necessary, such as when a student graduates or withdraws from the institution.
- These actions support efficient data management, ensuring the integrity and relevance of student information within the SIS.

Mark attendance:

- Once a suitable job opportunity is identified, negotiation with the client ensues to establish project parameters.
- Negotiation may involve discussions on project scope, deliverables, deadlines, pricing, and terms of service.
- BPO representatives may engage in communication with clients through meetings, emails, or video conferences to reach mutual agreements.
- The negotiation phase aims to ensure alignment between client expectations and the BPO organization's capabilities.

Marking Attendance:

- In a Student Information System (SIS) use case diagram, "Marking Attendance" allows instructors or designated staff members to record student attendance during classes or academic activities.
- This functionality enables users to access attendance lists for specific sessions, mark students as present, absent, or tardy, and update attendance records accordingly.
- For example, an instructor can quickly take attendance for a lecture, indicating each student's attendance status. This use case ensures accurate tracking of student engagement, supports academic performance evaluation, and helps enforce attendance policies efficiently within the SIS.

Preparing test paper:

- In a Student Information System (SIS) use case diagram, "Preparing Test Paper" allows instructors or designated staff to create assessments for students.

- This functionality enables users to access a test creation interface within the SIS, where they can define parameters such as test format, question types, and time limits.
- Users can then assemble questions from a question bank, customize the assessment, and finalize it for distribution to students.
- This use case streamlines the test creation process, enhances assessment efficiency, and ensures consistency in assessment design within the SIS.

Result declaration:

- In a Student Information System (SIS) use case diagram, "Declaring Result" allows instructors or administrative staff to record and publish student grades or examination results.
- This functionality enables users to input grades, verify accuracy, and finalize results for publication to students.
- For example, an instructor accesses the result management module, inputs grades, reviews entries for accuracy, and finalizes results for publication. This use case ensures efficient management of student assessment outcomes and facilitates timely communication of grades within the SIS.

Fill Admission form:

- In a Student Information System (SIS) use case diagram, "Filling Admission Form" enables prospective students to complete and submit application forms for admission to educational programs.
- This functionality allows users to access the admission application module within the SIS, input personal and academic details, and upload required documents.
- For instance, a prospective student visits the institution's website, fills out the online admission form, and submits it electronically through the SIS.
- This use case simplifies the admission process, enhances accessibility for applicants, and facilitates efficient handling of admission applications within the system.

Fees payment:

- In a Student Information System (SIS) use case diagram, "Fees Payment" allows students to conveniently pay tuition fees or related charges through the system.
- This functionality enables students to access the fees payment module within the SIS, view their outstanding balances, select payment methods, and securely complete transactions.
- For example, a student logs into the SIS, views their invoice, selects a payment method, and completes the transaction. This use case streamlines the fee payment process, enhances user

convenience, and ensures timely processing of payments within the SIS.

Get enrolled:

- In a Student Information System (SIS) use case diagram, "Getting Enrolled for Exams" enables students to easily register for upcoming examinations or assessments.
- This functionality allows students to access the exam enrollment module within the SIS, view available exam schedules, select the exams they wish to take, and confirm their enrollment.
- For instance, a student logs into the SIS, selects the desired exams from the available list, and completes the enrollment process. This use case simplifies exam registration, provides students with visibility into available exam opportunities, and ensures efficient management of examination enrollment within the system.

RESULT:

Thus the use case diagram for Student Information System (SIS) was implemented and executed successfully.

Ex no:04 DATE:	**Identify the conceptual classes and develop a domain model and derive**

Aim:

To identify the conceptual classes and develop a domain model and derive class diagram for Student Information system.

Conceptual Classes:

1.Admin: Represents the administrative staff responsible for managing the Student Information System. Attributes may include admin ID, username, password, contact information, etc.

2.Class: Represents the academic classes or courses offered within the educational institution. Attributes may include class ID, title, description, schedule, room number, etc.

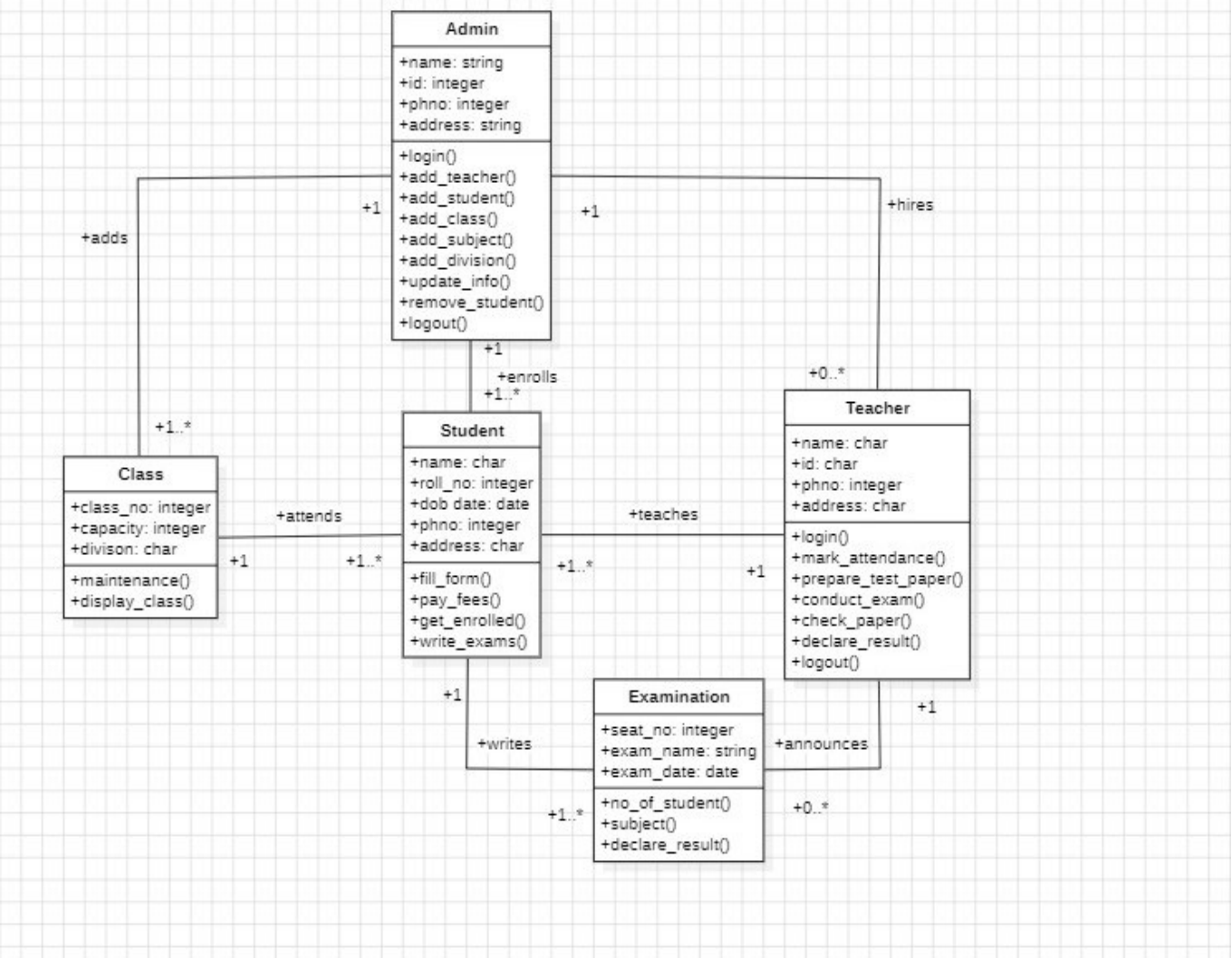
3.Student: Represents the individual students enolled in the educational institution. Attributes may include student ID, name, address, contact information, grade level, etc.

4.Teacher: Represents the instructors or educators responsible for teaching classes within the educational institution. Attributes may include teacher ID, name, subject expertise, contact information, etc.

5.Examination: Represents the assessments or examinations administered to students. Attributes may include exam ID, title, description, date, time, duration, etc.

The Student Information System (SIS) domain model encapsulates key entities and their attributes essential for managing student data and academic processes. This includes entities such as Student, Course, Enrollment, Teacher, Attendance, Grade, Exam, and Administrator. Students are identified by unique IDs and have associated personal details, while courses are defined with titles, descriptions, and assigned instructors. Enrollment records track students' participation in courses, attendance is logged for class sessions, and grades are recorded for assessments. Examinations are scheduled with specific details like dates and durations. Administrators oversee system operations with credentials and permissions. These entities and associated functionalities streamline student management, academic operations, and administrative tasks within the SIS.

Class Diagram for Student Information System:



DOMAIN MODEL:

A domain model for a student information system includes entities like Student, Course, Enrollment, Instructor, Department, Semester, Grade, Program, Faculty, Classroom, Assignment, and Attendance. These entities represent key concepts and their relationships within the system, such as students enrolling in courses taught by instructors, belonging to departments, attending classes in classrooms, completing assignments, and receiving grades.

Domain Model for Student Information System:

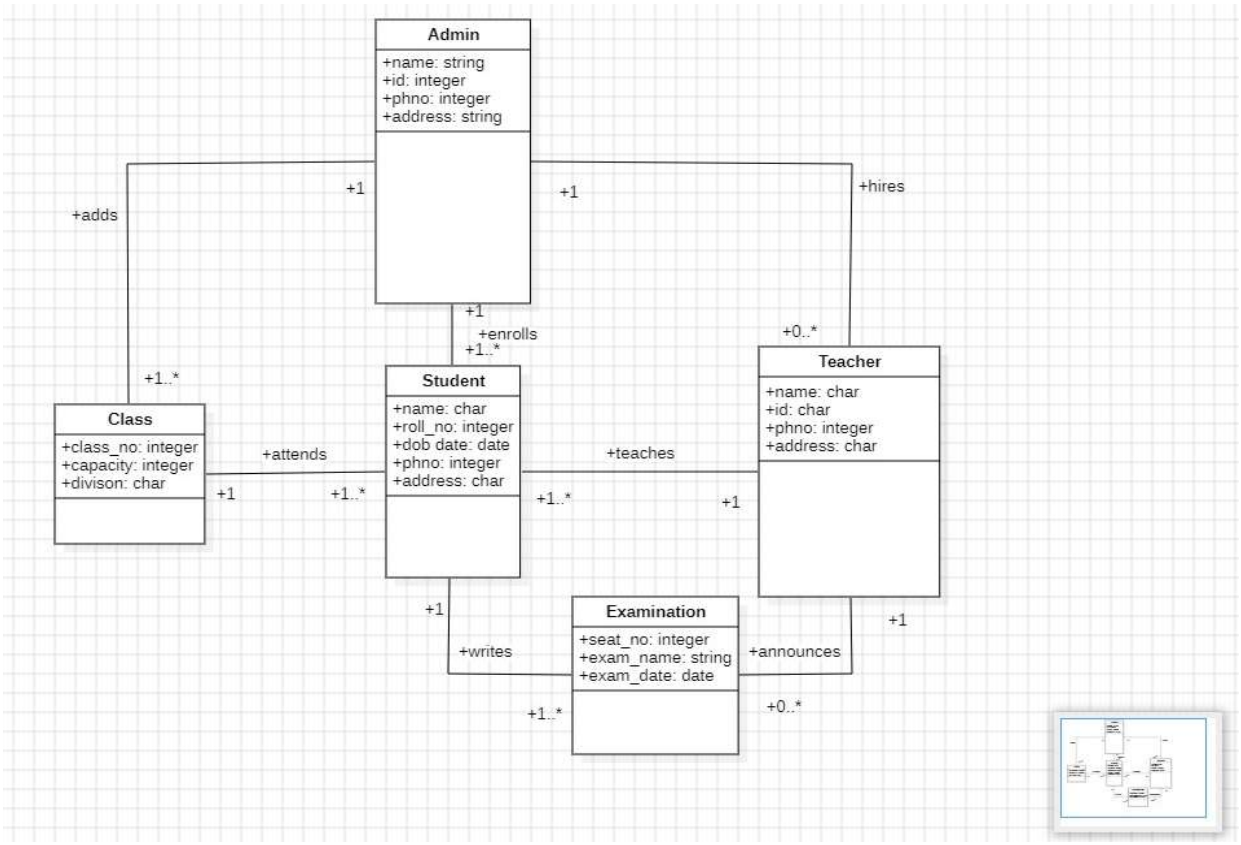
1.Student: The central entity in the system, representing individuals enrolled in educational programs. Attributes may include student ID, name, date of birth, contact information, etc.

2.Course: Represents the courses offered by the educational institution. Attributes may include course code, title, description, credits, etc.

3.Enrollment: Connects students to the courses they are enrolled in. It may include additional attributes like enrollment date, grade, status, etc.

4.Instructor: Represents the instructors who teach courses. Attributes may include instructor ID, name, contact information, etc.

5.Department: Represents the departments within the institution responsible for offering courses. Attributes may include department code, name, description, etc.



RESULT:

Thus the class diagram for Student Information system is implemented and executed Successfully.

Ex no:05 DATE:	**Using the identified scenarios, find the interaction between using objects and represent them using UML Sequence and Collaboration Diagram for Student Information System**

Aim:

To represent the interaction between objects using UML Sequence diagram and Collaboration diagram for Student Information System

Sequence diagram:

An object is shows as an box and the top of dash vertical line. This vertical line is called object life line. The life line represents objects life during interaction this was give by "Jackupson". Each message is represent by an arrow between the lifeline of Two object.

The order of message is occurred from top to bottom of a page. Message contain Messages name, argument and some control information.

Self call is an message that an object sends to itself by sending messages arrow back to the same lifeline.

A sequence diagram illustrates a kind of format in which each object interacts via message. It is generalize between two or more specialized diagram.

This interactive behaviour is represented in UML by two diagrams known as Sequence diagram and Collaboration diagram. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

The purposes of interaction diagrams are to visualize the interactive behaviour of the system. Now visualizing interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction, that is why sequence and collaboration diagrams are used to capture dynamic nature but from a different angle.

The purposes of interaction diagram can be describes as:

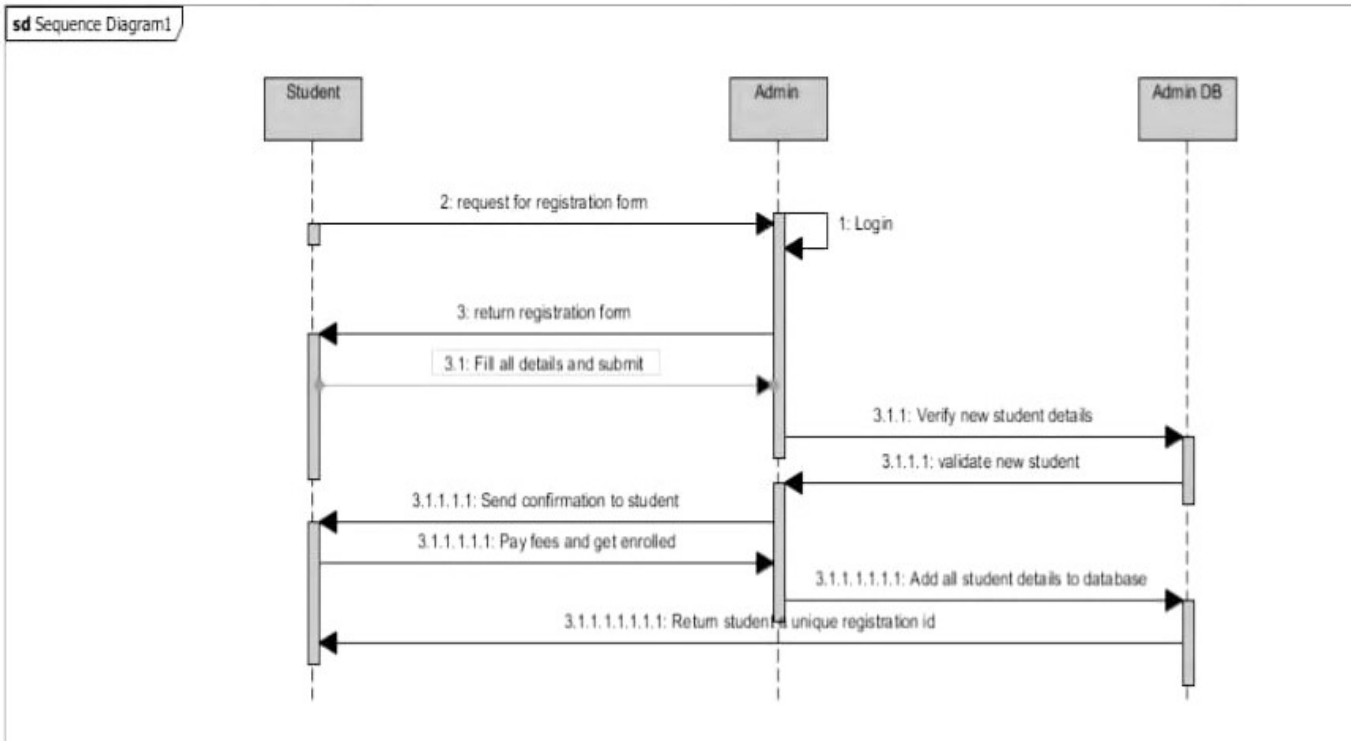
i) To capture dynamic behaviour of a system.

ii) To describe the message flow in the system.

iii) To describe structural organization of the objects.

iv) To describe interaction among objects.

Sequence diagram for Student Information System:



Collaboration Diagram:

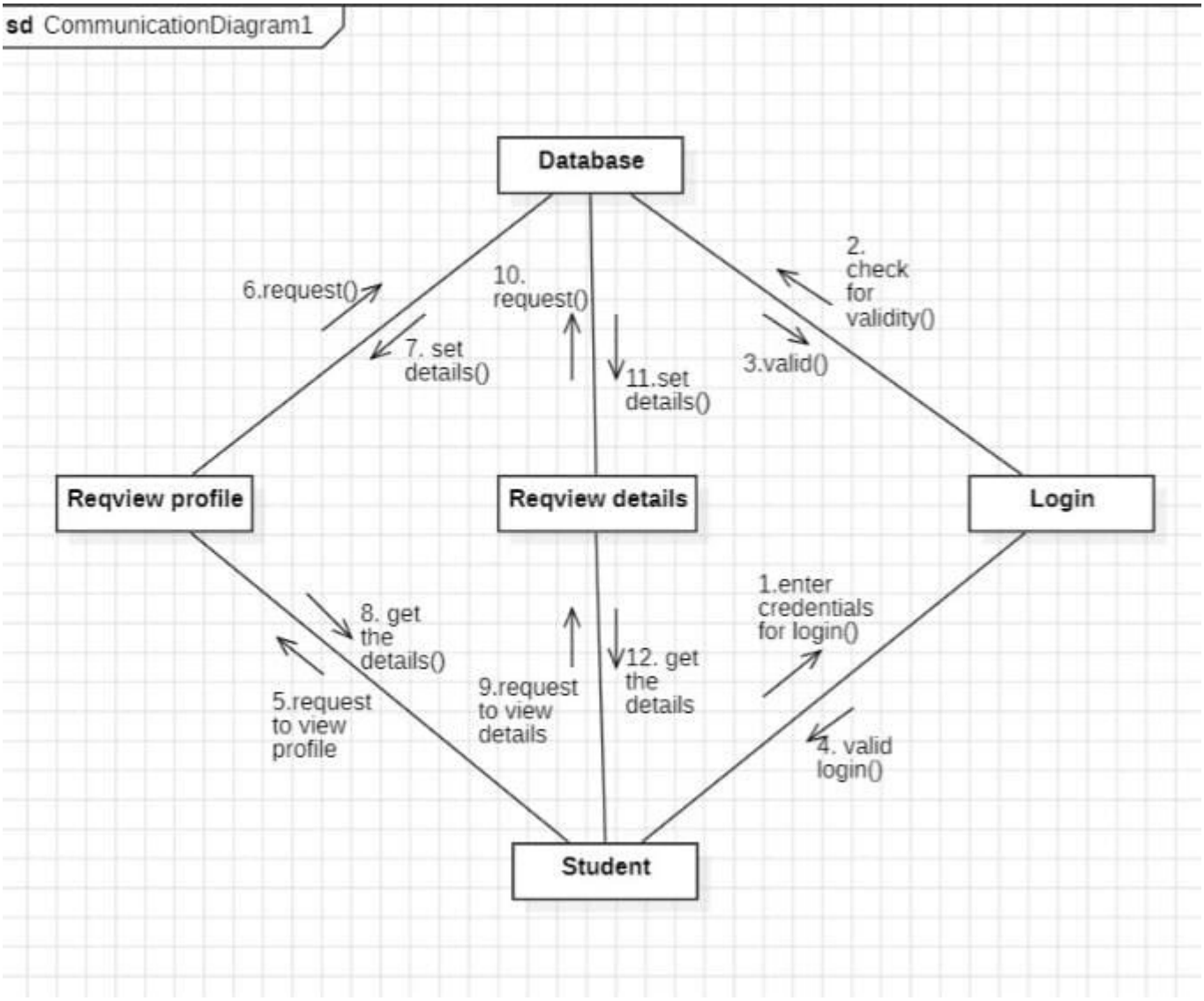
Communication diagram illustrate that object interact on a graph or network format in which object can be placed where on the diagram.

In collaboration diagram the object can be placed in anywhere on the diagram.

The collaboration comes from sequence diagram. The collaboration diagram represents the collaboration which is a set of object related to achieve and decide outcome.

In collaboration the sequence is indicated by numbering the messages several numbering schemes are available.

Collaboration diagram for Student Information System:



Result:

Thus the UML Sequence diagram and Collaboration diagram for Student Information System was developed successfully.

Ex no:06 DATE:	**Draw relevant state chart and Activity Diagram for the same system**

Aim:

To draw Activity diagram for Student Information System

State chart diagram:

A State Chart Diagram, also known as a State Machine Diagram, is a type of behavioral diagram in the Unified Modeling Language (UML). It represents the various states that an object or system can be in and the transitions between those states triggered by events.

In a State Chart Diagram:

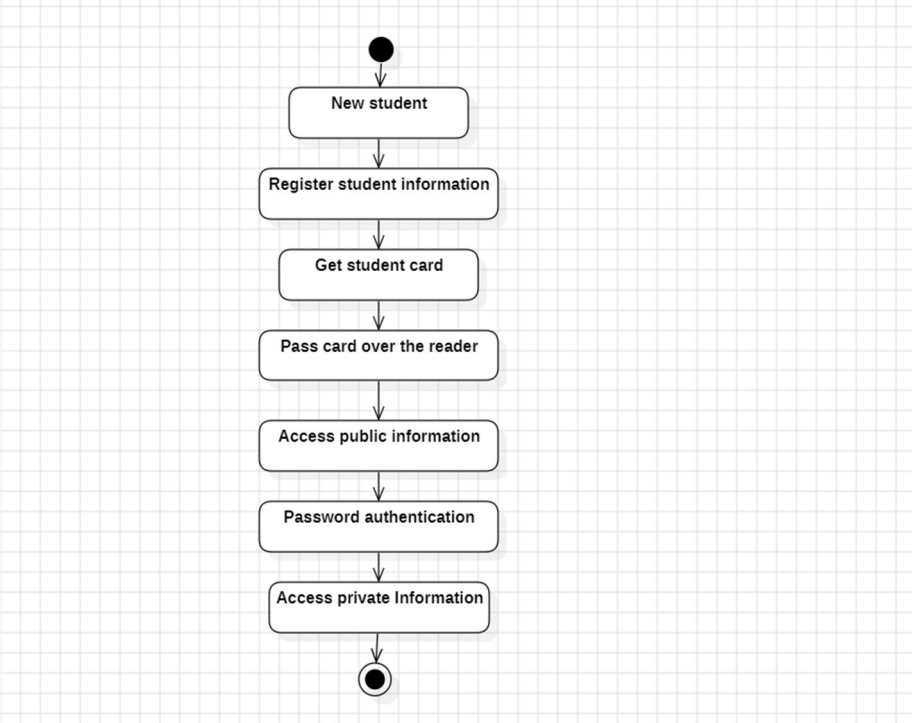
1. States: These represent the different conditions or situations that an object or system can be in at any given time. For example, in a traffic light system, states could include "Green," "Yellow," and "Red."

2. Transitions:These indicate the changes from one state to another. Transitions are triggered by events or conditions. For instance, in the traffic light system, the transition from "Green" to "Yellow" might be triggered by a timer, while the transition from "Yellow" to "Red" could occur when the timer expires.

3. Events: These are the stimuli or occurrences that cause a transition from one state to another. Events can be internal or external to the system. Examples of events in a student information system could include "Student Enrollment," "Course Registration," or "Grade Submission."

State Chart Diagrams are useful for modeling the behavior of systems that have a finite number of possible states and well-defined transitions between those states. They help to visualize the dynamic behavior of a system and can aid in requirements analysis, system design, and implementation.

State Chart diagram for Student Information System:



Activity diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be usedto describe the business and operational step-by-step workflows of components in a system.

An activity diagram shows the overall flow of control. An activity is shown as an rounded boxcontaining the name of the operation. This activity diagram describes the behaviour of the system.

An activity diagram is variation or special case or a state machine, in which the states are activitiesrepresenting a performance of operation and the transition or triggered by the completion of operation.

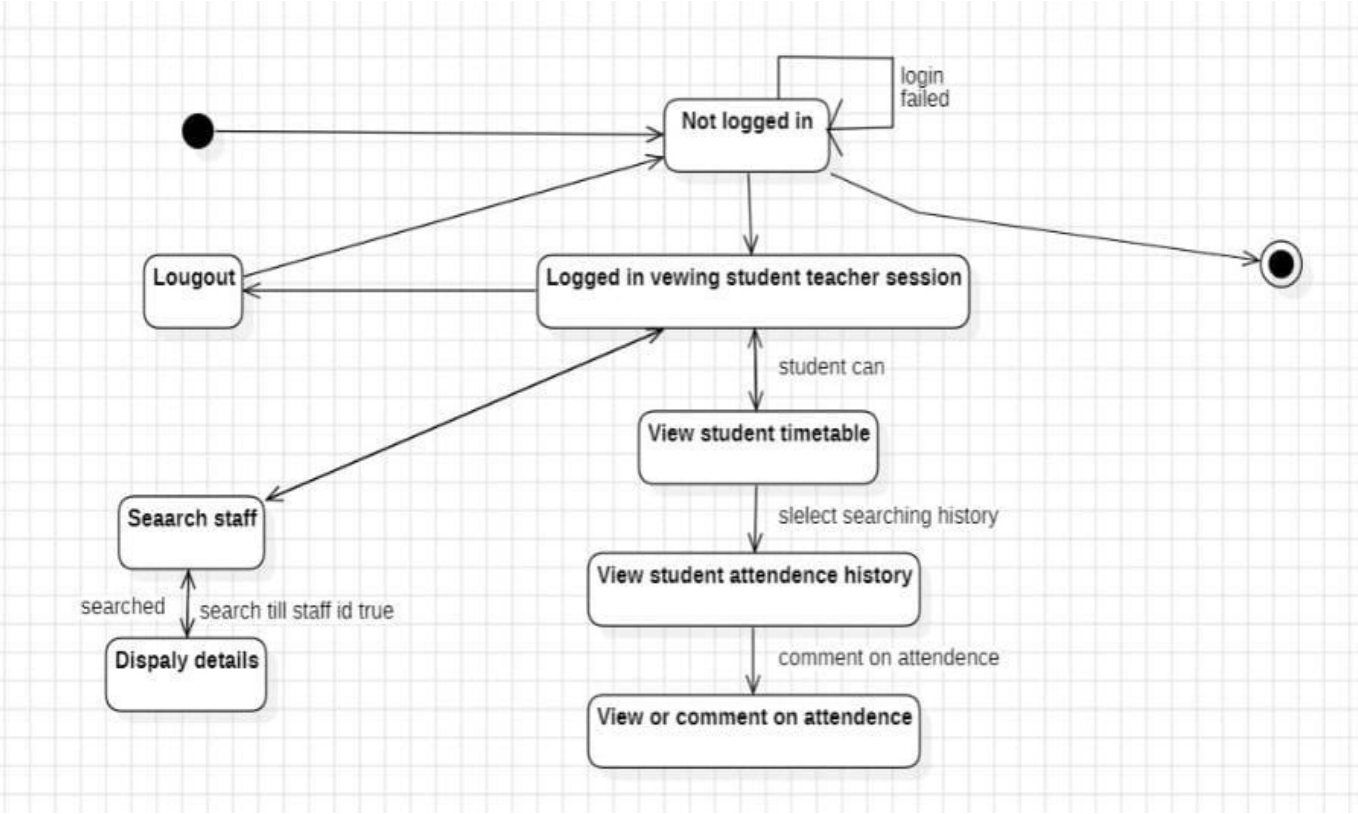
Activity diagrams deals with all type of flow control by using different elements like fork, join etc. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagramis used to show message flow from one activity to another.

Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flowchart but it is not. It shows different flow like parallel, branched, concurrent and single.

The purposes can be described as:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

Activity diagram for Student Information System:



Result:

Thus the activity diagram for Student Information System was developed successfully.

Ex no:07 DATE	Implement the system as per the detailed design

Aim:

To implement the Student Information System as per the detailed design.

Implementation for student Information System:

Class.java

```
import java.util.*;
/**
 *
 */
public class Class {
    /**
     * Default constructor
     */
    public Class() {
    }
    /**
     *
     */
    public integer class_no;
    /**
     *
     */
    public integer capacity;
    /**
     *
     */
    public char divison;
    /**
     *
     */
    public void maintenance() {
        // TODO implement here
    }
    /**
     *
     */
    public void display_class() {
        // TODO implement here
    }
}
```

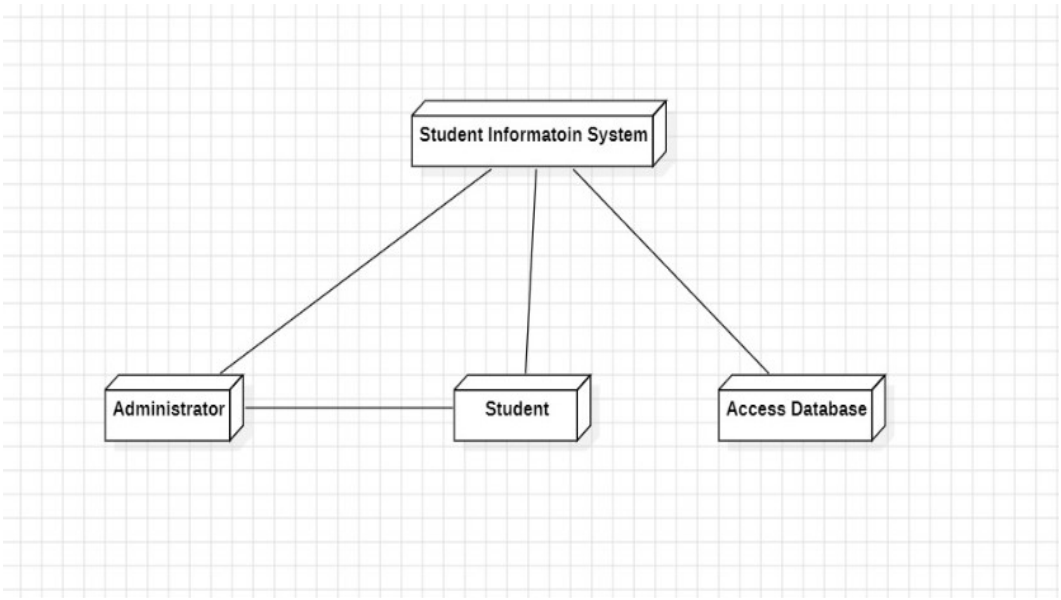
Admin.java

```
import java.util.*;
/**
 *
 */
public class Admin {
    /**
     * Default constructor
     */
    public Admin() {
    }
    /**
     *
     */
    public string name;
    /**
     *
     */
    public integer id;
    /**
     *
     */
    public integer phno;
    /**
     *
     */
    public string address;
    /**
     *
     */
    public void login() {
        // TODO implement here
    }
    /**
     *
     */
    public void add_teacher() {
        // TODO implement here
    }
    /**
     *
     */
    public void add_student() {
        // TODO implement here
    }
    /**
```

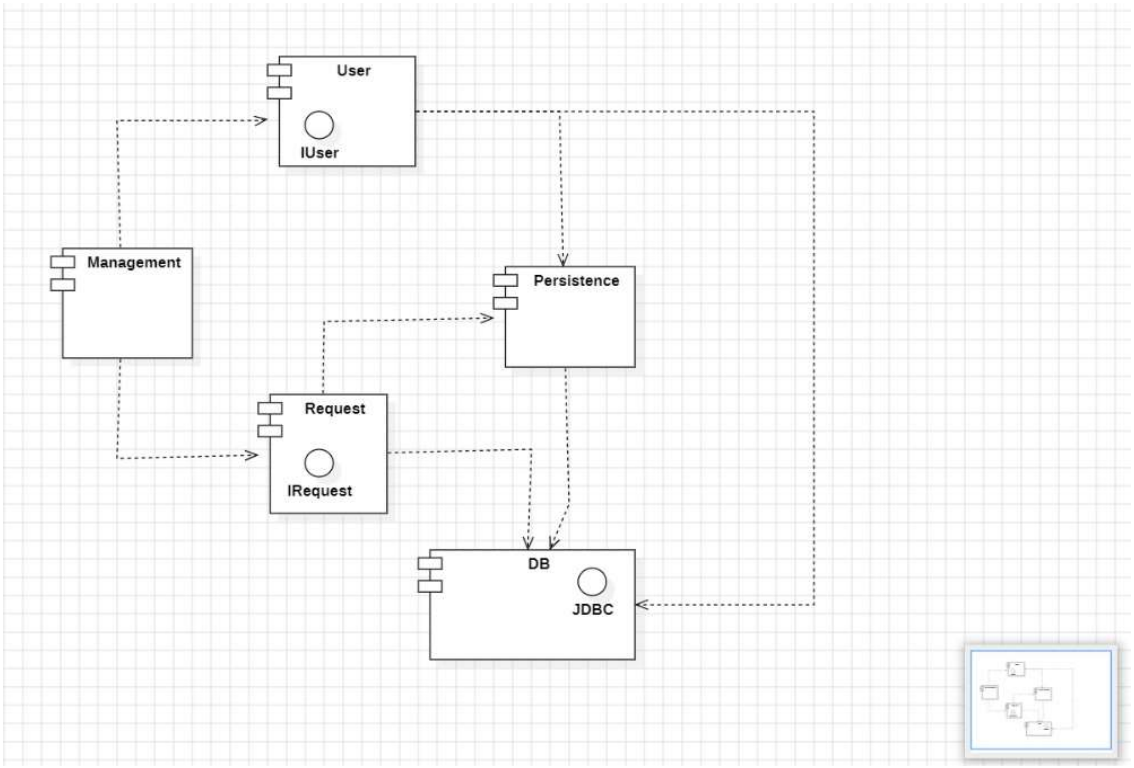
```
     *
     */
    public void add_class() {
        // TODO implement here
    }
    /**
     *
     */
    public void add_subject() {
        // TODO implement here
    }
    /**
     *
     */
    public void add_division() {
        // TODO implement here
    }
    /**
     *
     */
    public void update_info() {
        // TODO implement here
    }

    /**
     *
     */
    public void remove_student() {
        // TODO implement here
    }
    /**
     *
     */
    public void logout() {
        // TODO implement here
    }
}
     */
    public void get_enrolled() {
        // TODO implement here
    }
    /**
     *
     */
```

Deployment diagram:



Component diagram:



RESULT:

Thus the implementation of Student Information System was executed and the codes were generated successfully.

Ex no:08 DATE:	**Test the software system for all scenarios identified as per the use case diagram.**

Aim:

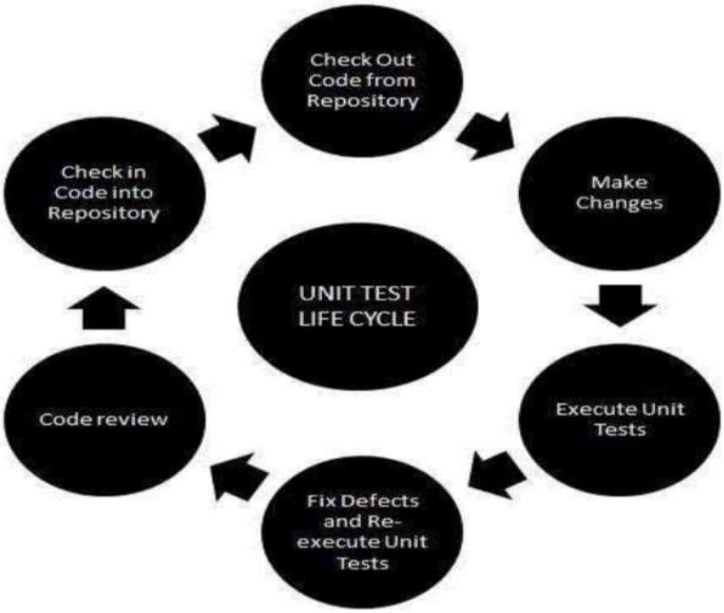
To test the software system for all scenarios identified in the use case diagram.

Student Information System:

The Student Information System (SIS) currently in use struggles with manual processes and resource-intensive tasks, hindering the efficient management of student data across educational institutions.

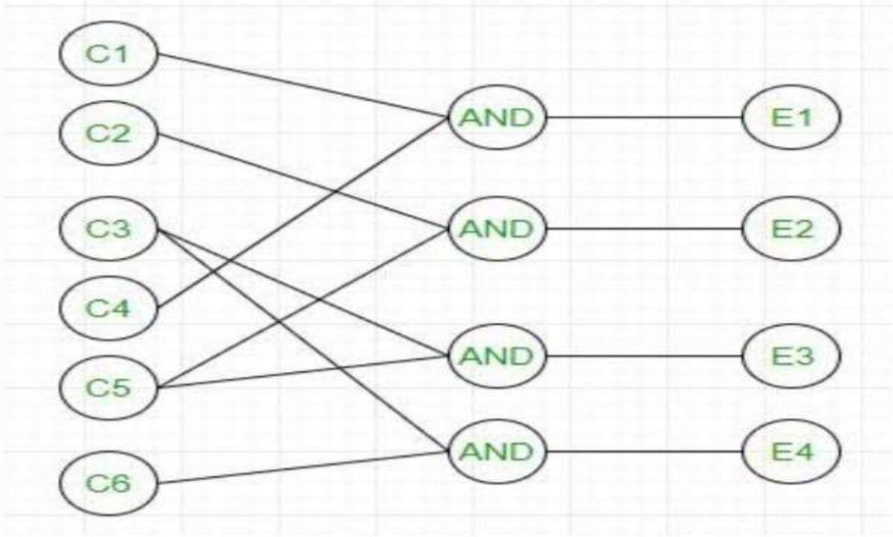
.**Unit Testing:**

- Unit testing involves testing individual components or modules of the software to ensure they function correctly in isolation.
- Each module, such as client management, project management, employee management, etc., would undergo unit testing.
- Example: Testing the "addClient()" function to ensure that a new client is successfully added to the system.



Black Box Testing:

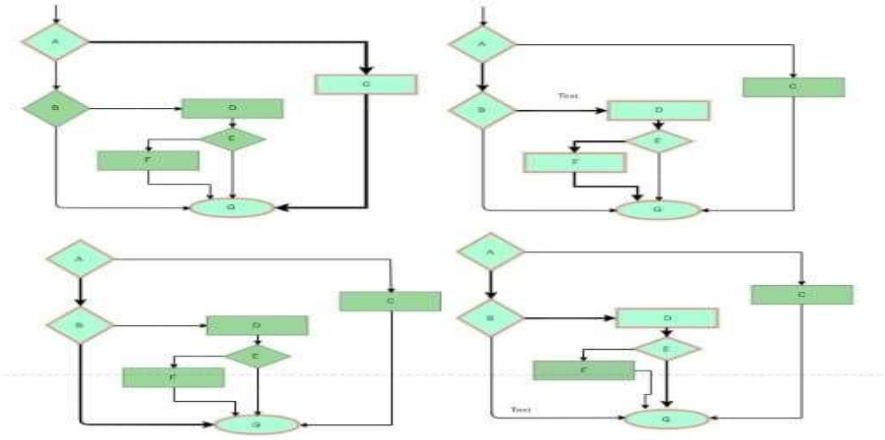
- Black box testing is a technique where the internal workings of the system are not known to the tester. The tester only tests the system's functionality based on its specifications.
- Testers would input various sets of data into the system and verify that the expected output is produced.
- Example: Testing the "searchForJob()" functionality to ensure that it returns the expected results based on different search criteria.



		1	2	3	4
CAUSES	C1	1	0	0	0
	C2	0	1	0	0
	C3	0	0	1	1
	C4	1	0	0	0
	C5	0	1	1	0
	C6	0	0	0	1
EFFECTS	E1	X	-	-	-
	E2	-	X	-	-
	E3	-	-	X	-
	E4	-	-	-	X

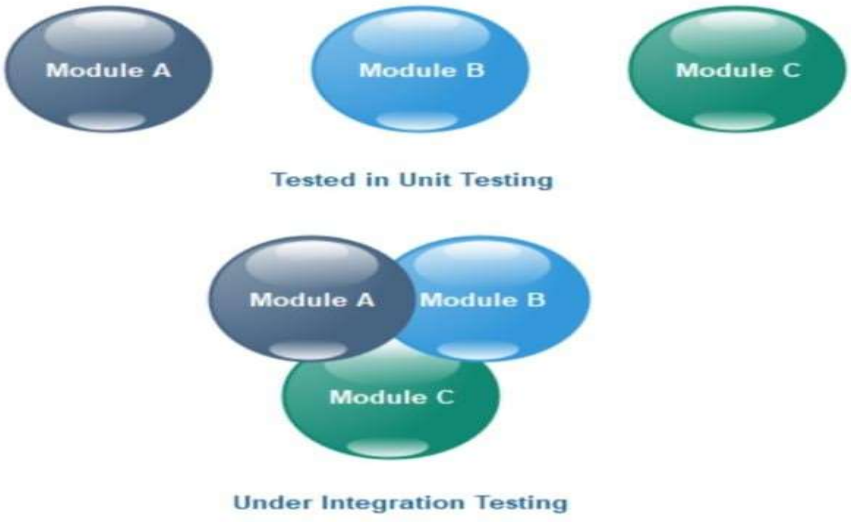
White Box Testing:

- White box testing involves testing the internal logic and structure of the software code.
- Testers would examine the code of individual modules to ensure that all code paths are tested.
- Example: Testing the "performQC()" function to ensure that it adequately checks the quality of the processed data.



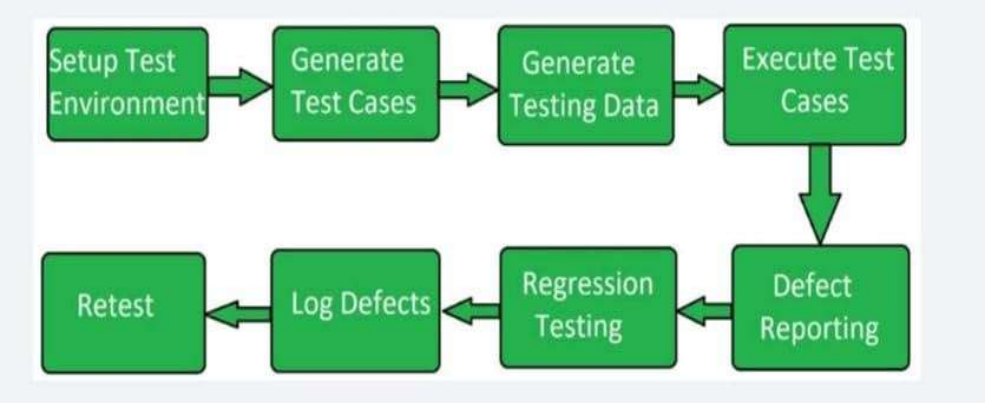
Integration Testing:

- Integration testing verifies that different modules of the software work together as expected.
- Testers would test the interaction between different modules, such as client management, project management, and employee management.
- Example: Testing the interaction between the "addClient()" and "addProject()" functions to ensure that a project can be associated with a client successfully.



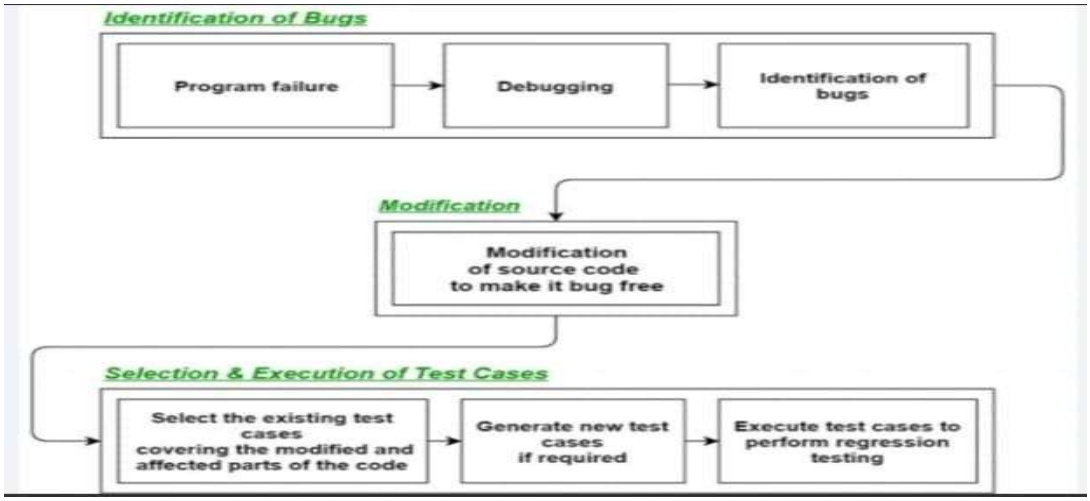
System Testing:

- System testing involves testing the entire system as a whole to verify that it meets the specified requirements.
- Testers would perform end-to-end testing of the entire system, including all modules and their interactions.
- Example: Testing the entire process from client negotiation to project delivery and payment to ensure that it functions as expected.



Regression Testing:

- Regression testing ensures that new changes or additions to the system do not adversely affect existing functionalities.
- Testers would re-run previously conducted tests after new changes or additions are made to the system to ensure that existing functionalities are not affected.
- Example: After adding a new feature to the system, testers would re-run all existing tests to ensure that the new feature did not introduce any bugs or errors.



Result:

Thus the software system for all scenarios identified in the use case diagram was tested successfully.

Ex no:09 DATE:	**Improve the reusability and maintainability of the software by applying appropriate design patterns**

Aim:

To improve the reusability and maintainability of the software system by applying appropriate design patterns.

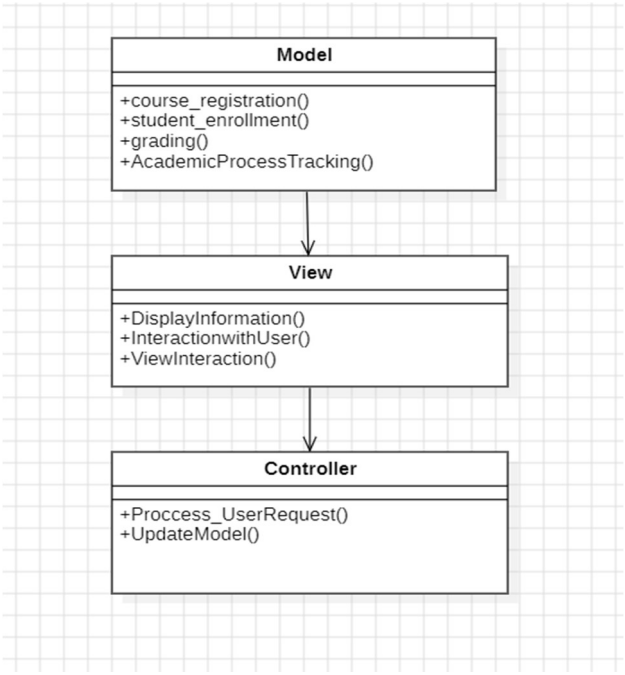
Student Information System (SIS):

To bolster the reusability and maintainability of a Student Information System (SIS), integrating design patterns is paramount. By employing the Factory Method Pattern, diverse student-related objects can be created uniformly, fostering adaptability and scalability. The Observer Pattern ensures seamless updates across modules when student data changes, enhancing real-time synchronization. Additionally, the Decorator Pattern facilitates the augmentation of student profiles and courses with additional features, promoting flexibility. Leveraging the Composite Pattern enables efficient representation of hierarchical structures like departmental arrangements, simplifying management tasks. Lastly, the Singleton Pattern ensures critical system components maintain singular instances, optimizing resource management and consistency throughout the SIS.

Model-View-Controller (MVC):

- The Model represents student data and business logic in the SIS. It manages operations such as student enrollment, course registration, grading, and academic progress tracking.
- The View displays student information and provides interfaces for interaction within the SIS. It presents screens or pages where users (students, faculty, administrators) can view and interact with the system.
- The Controller processes user requests and coordinates actions between the View and Model in the SIS. It interprets user input, initiates corresponding actions, and updates the Model based on user interactions.

  This component facilitates tasks such as course enrollment, grade viewing, personal information updates, and report generation.



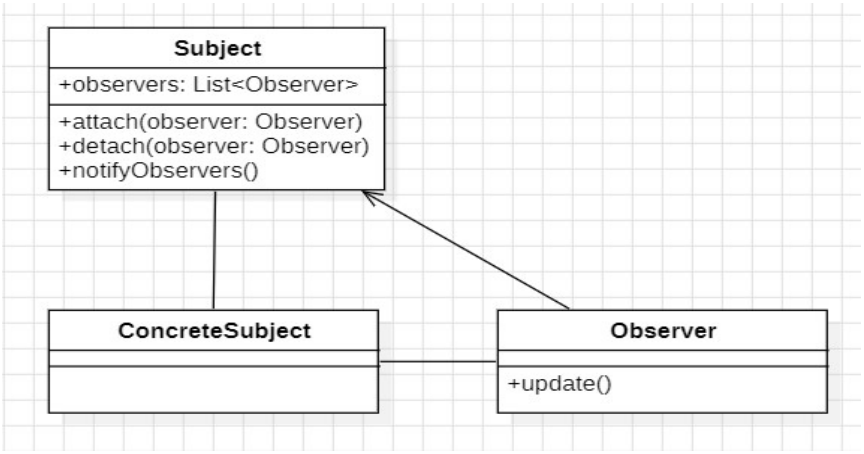
Observer Pattern:

- The Observer Pattern notifies components about system changes, enhancing flexibility and scalability. It enables easy addition of new notifications, improving system maintainability.

❖ Context: Used when objects need to be notified of changes in another object's state.

❖ Problem: Establishes a one-to-many relationship without tightly coupling objects, ensuring automatic updates.

❖ Solution: Define a subject interface with methods for attaching, detaching, and notifying observers. Observers register with subjects to receive updates on state changes.

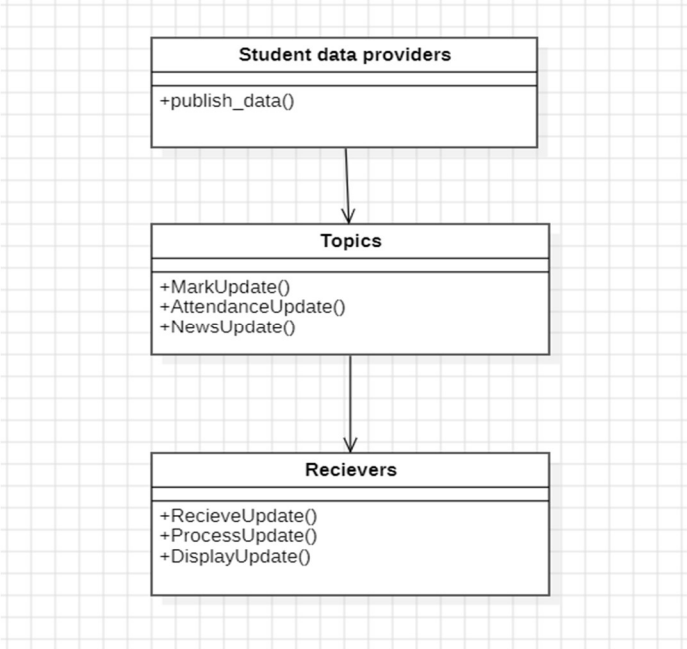


Publish-Subscribe Pattern:

The Publish-Subscribe Pattern facilitates real-time updates and notifications across various modules within a Student Information System.

Employing this pattern enriches scalability, adaptability, and responsiveness within the architecture of the student information system.

- Context: Real-time updates are paramount for administrative and academic operations within educational institutions.
- Problem: Distributing real-time student information without the Publish-Subscribe Pattern results in data latency and operational inefficiencies.
- Solution: By adopting the pattern, efficient dissemination of student data becomes achievable, allowing relevant updates to be broadcasted to subscribing modules, thereby bolstering system scalability and responsiveness.



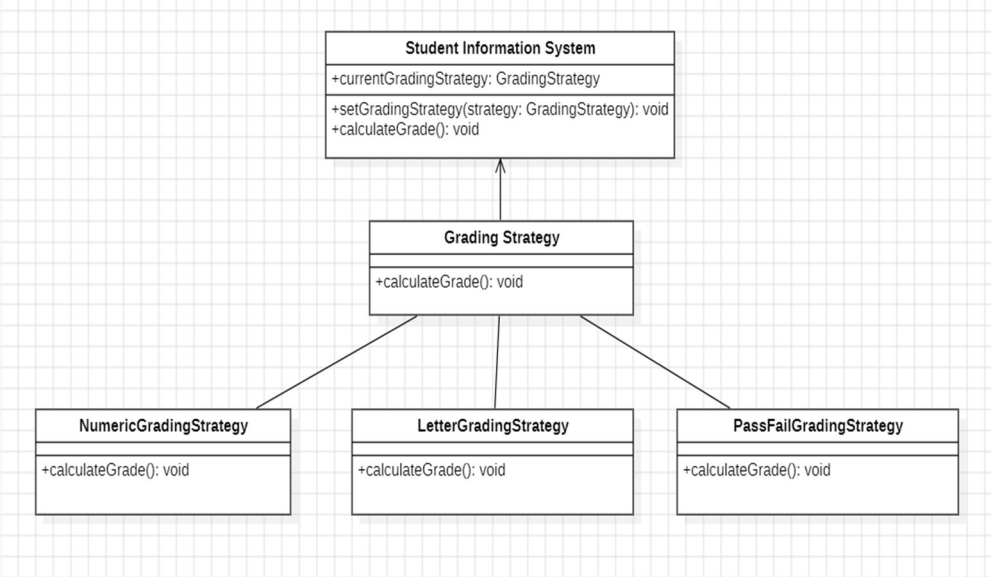
Strategy Pattern:

The Strategy Pattern proves effective in managing various algorithms for academic performance evaluation within a Student Information System.

Through the implementation of the Strategy Pattern, the system accommodates interchangeable evaluation algorithms, granting flexibility to users in selecting the most suitable assessment strategy.

- Context: Utilized to manage a set of algorithms interchangeably for academic evaluation purposes.
- Problem: Facilitates dynamic selection and switching of evaluation algorithms, fostering code flexibility and reusability.
- Solution: Define an interface for evaluation strategies with algorithmic methods. Implement

concrete strategy classes for each evaluation method and allow for runtime switching between them, enabling adaptability and customization in academic assessment processes.



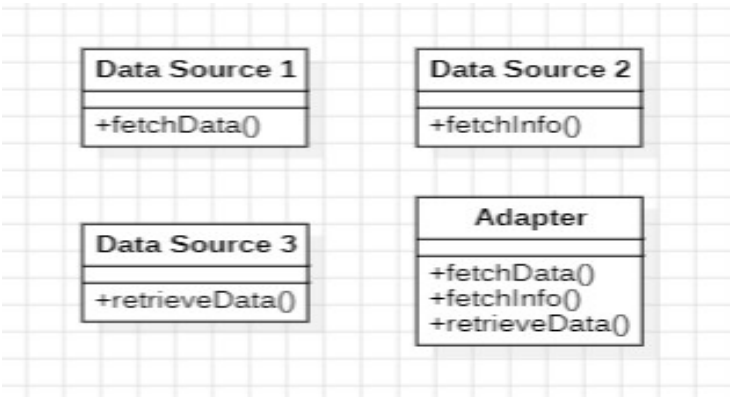
Adapter Pattern:

- The Adapter Pattern proves invaluable in seamlessly integrating various data sources into the Student Information System (SIS).
- It resolves the challenge of incompatible interfaces between data sources and the SIS by offering a wrapper that translates different data formats and protocols.
- Employing this pattern ensures smooth communication and data exchange, thereby enriching interoperability and flexibility within the student information ecosystem.

❖ Context:Integrating diverse data sources with varying formats and protocols is a common requirement within student information systems.

❖ Problem:The existence of incompatible interfaces between these data sources and the SIS results in communication obstacles and data parsing discrepancies.

❖ Solution: The Adapter Pattern presents a solution by furnishing a wrapper that translates data formats and protocols, facilitating seamless integration without compromising data integrity or communication efficiency.



Result:

Thus the reusability and maintainability of the software system by applying appropriate design patterns was executed successfully.