



## ***UE21CS352B - Object Oriented Analysis & Design using Java***

### **Mini Project Report**

### **“Computer Based Test Management System”**

*Submitted by:*

<b>Srikrishna R Chitnis</b>	<b>PES1UG21CS619</b>
<b>Srinivas</b>	<b>PES1UG21CS624</b>
<b>Ramakrishnan</b>	
<b>Suraj Rao</b>	<b>PES1UG21CS647</b>
<b>Suvan Ashwin</b>	<b>PES1UG21CS651</b>

*6<sup>th</sup> Semester K Section*

**Prof. Bhargavi Mokashi**  
Assistant Professor

**January - May 2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
FACULTY OF ENGINEERING  
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)  
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

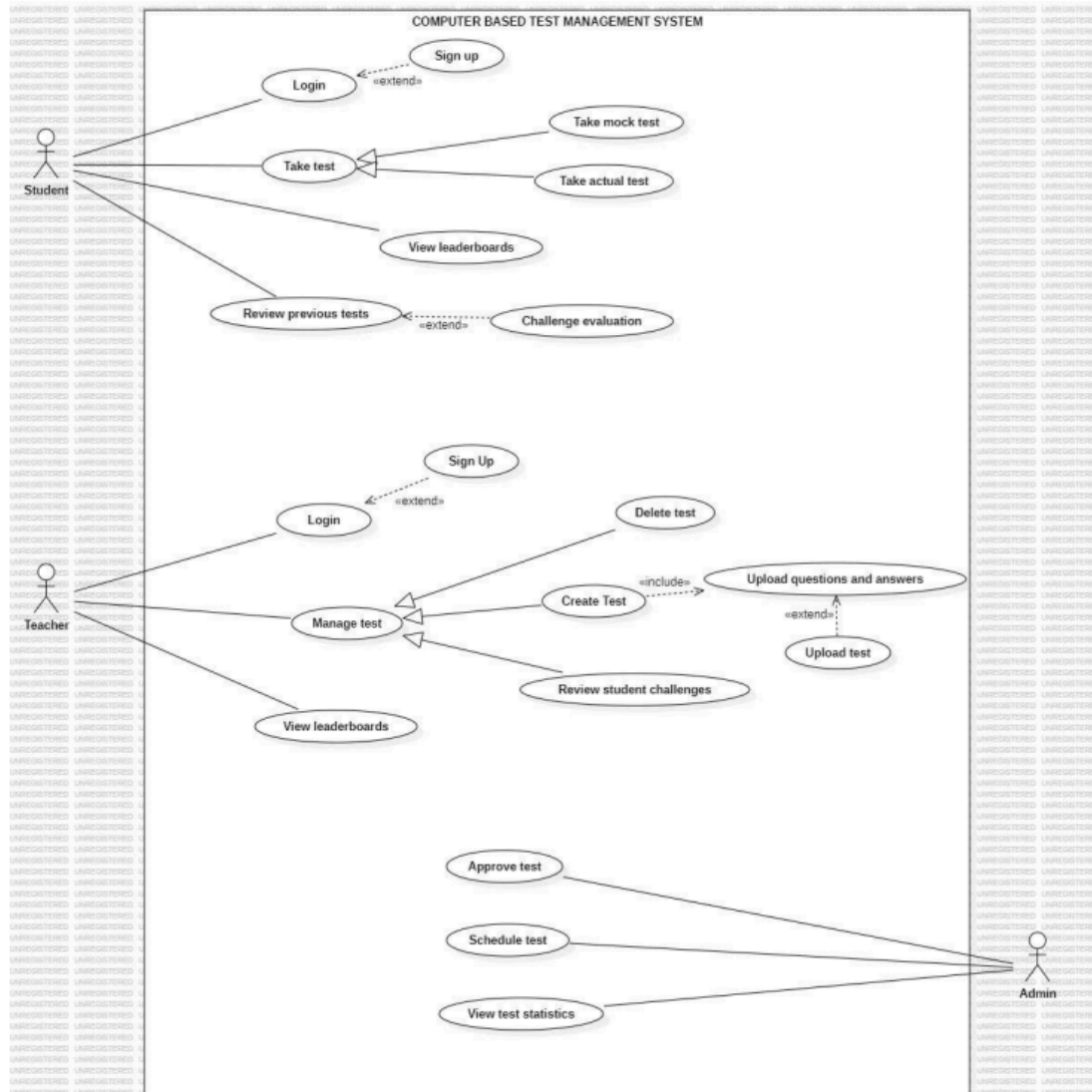
## **Problem Statement:**

**Design and develop a comprehensive Computer-Based Test Management System that facilitates the creation, scheduling, and administration of online tests/assessments. The system should cater to the needs of three distinct user roles: Students, Teachers, and Administrators. The primary objectives of the system are:**

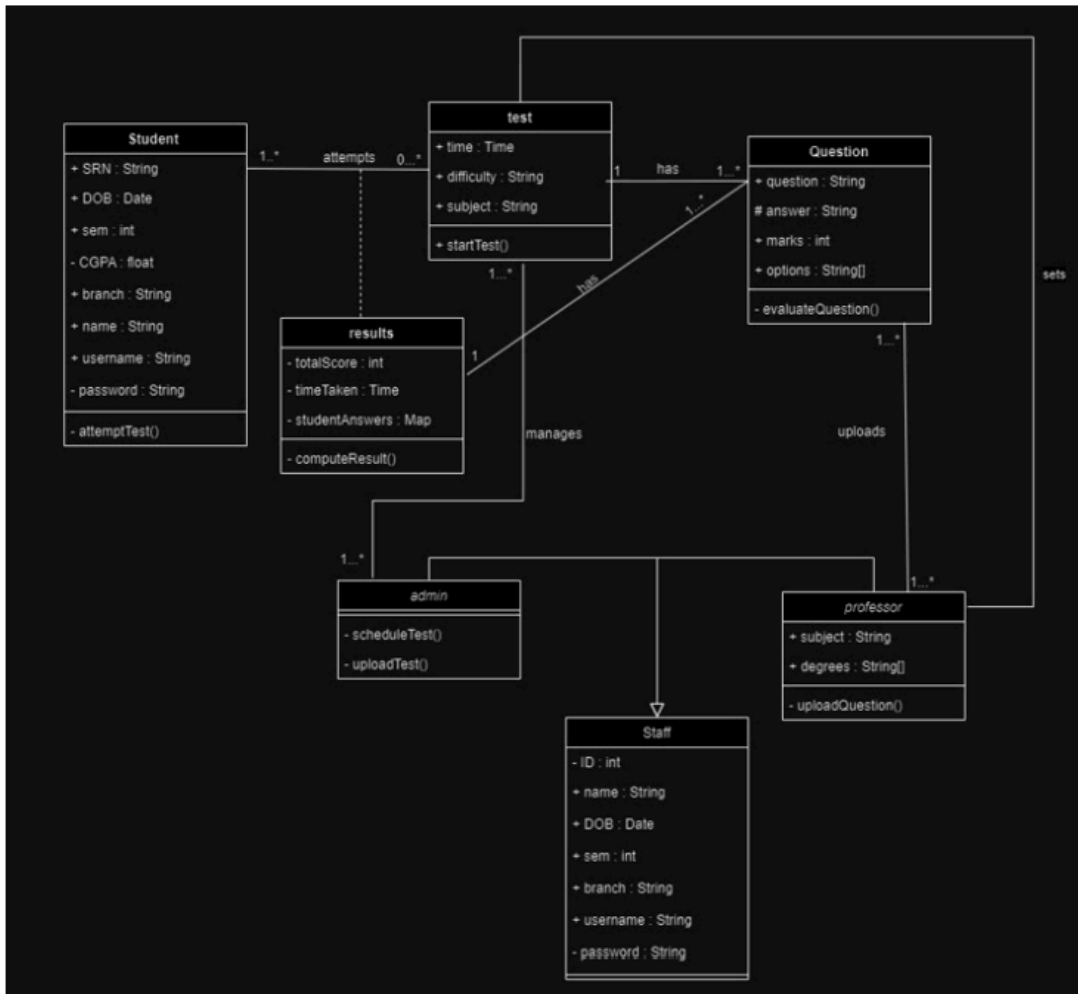
- **Student Module:**
  - **Enable students to view available and previously taken tests.**
  - **Provide a user-friendly interface for taking online tests.**
  - **Allow students to review their previous test submissions and challenge the evaluation if necessary.**
  - **Display a leaderboard to foster healthy competition and motivation among students.**
- **Teacher Module:**
  - **Empower teachers to create, update, and delete test questions and assemble comprehensive tests.**
  - **Facilitate the uploading of test questions.**
  - **Enable teachers to review and respond to student-initiated challenges regarding test evaluations.**
  - **Access the leaderboard for monitoring student performance.**
- **Administrator Module:**
  - **Grant administrators the authority to approve and schedule tests created by teachers.**
  - **Enable administrators to view comprehensive test statistics and analytics for performance monitoring and decision-making.**
  - **Implement a secure login/signup facility for students and teachers to access the system.**

## Models:

- USE CASE DIAGRAM :

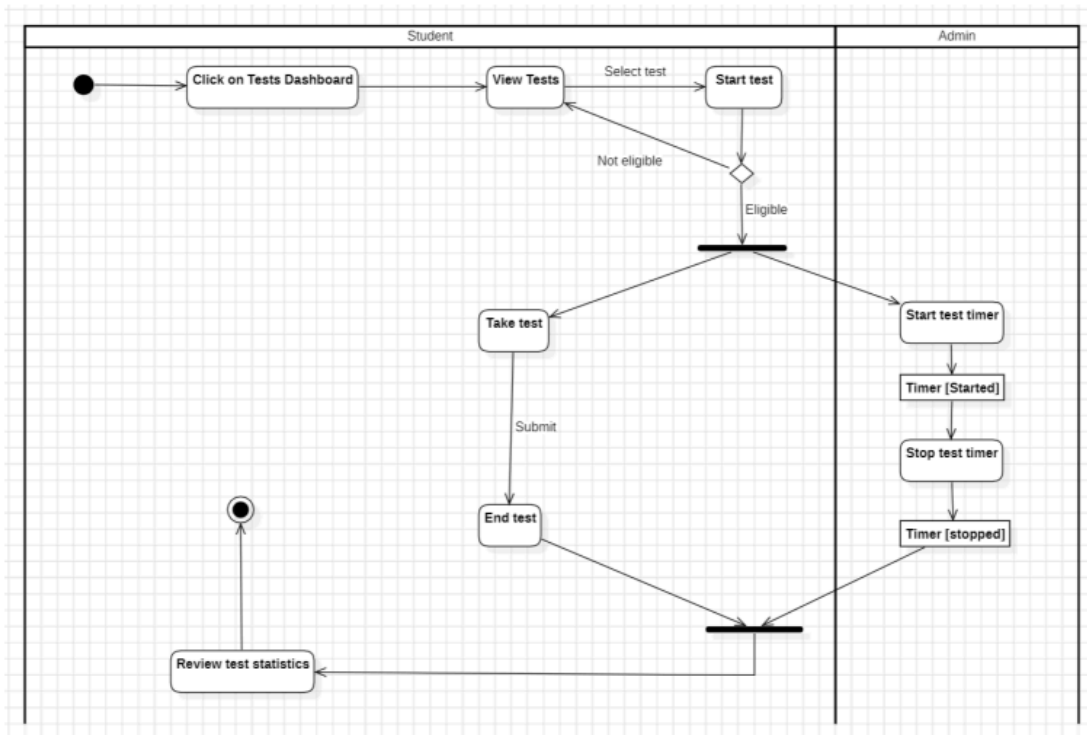


- CLASS DIAGRAM :

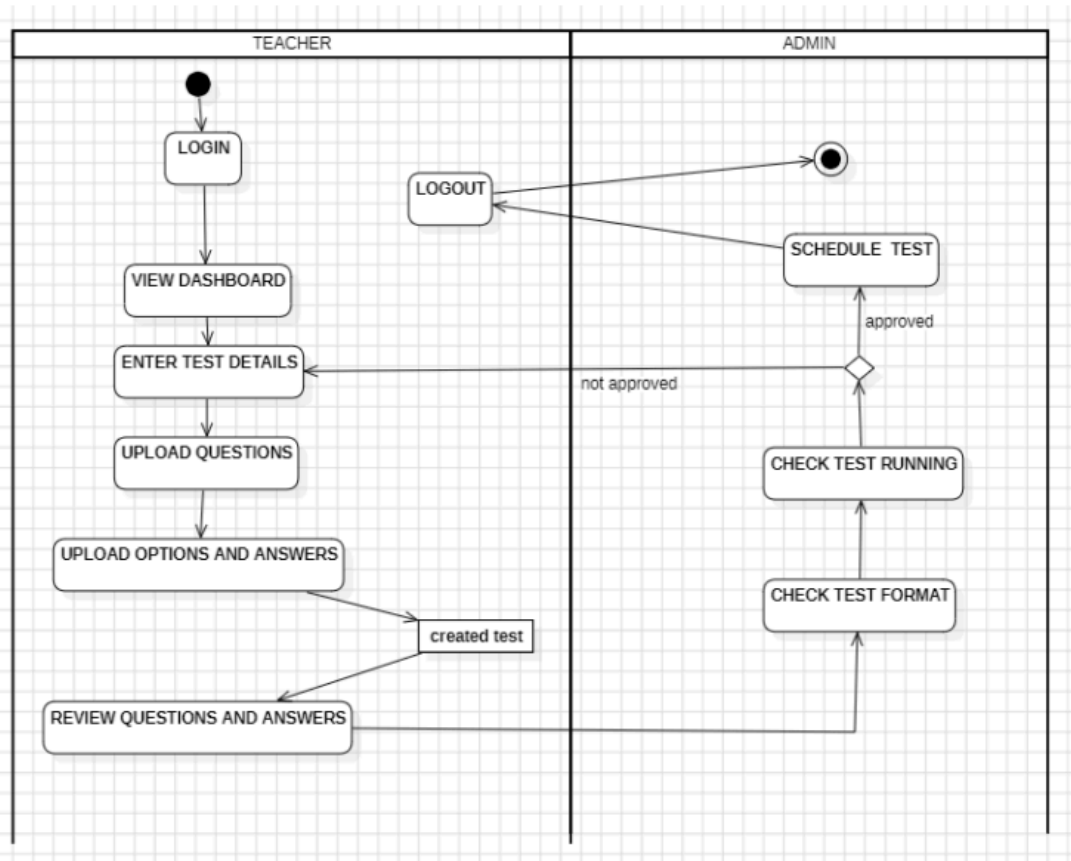


## Activity Diagrams:

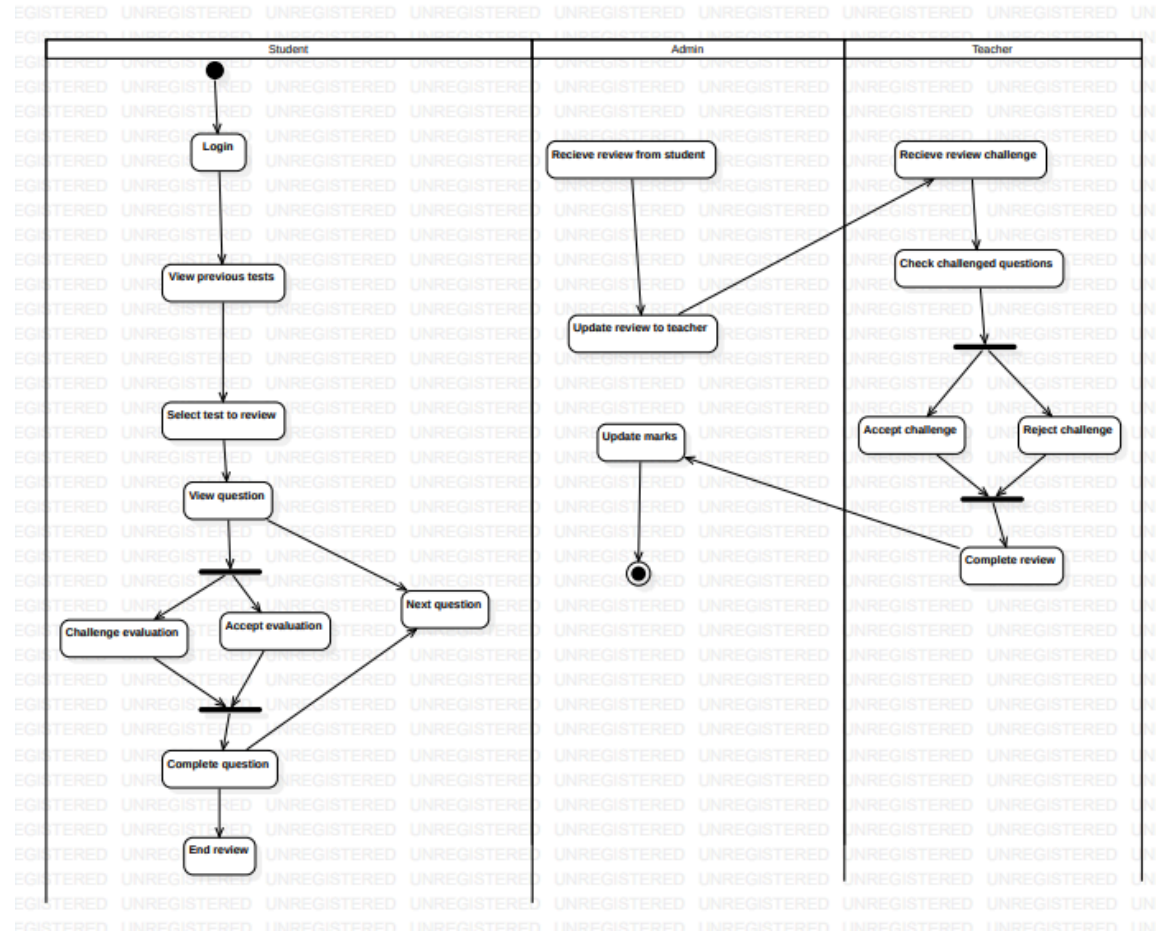
### 1) Activity Diagram for Taking Test



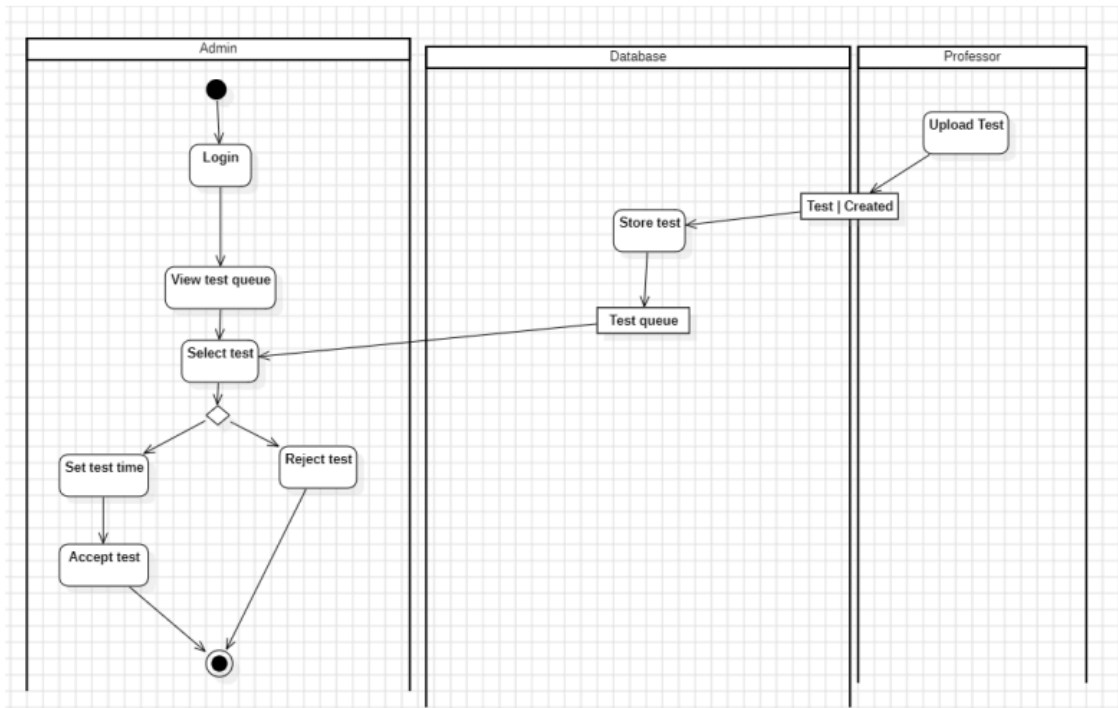
## 2) Activity Diagram for teacher creating test



### 3) Activity Diagram for reviewing test



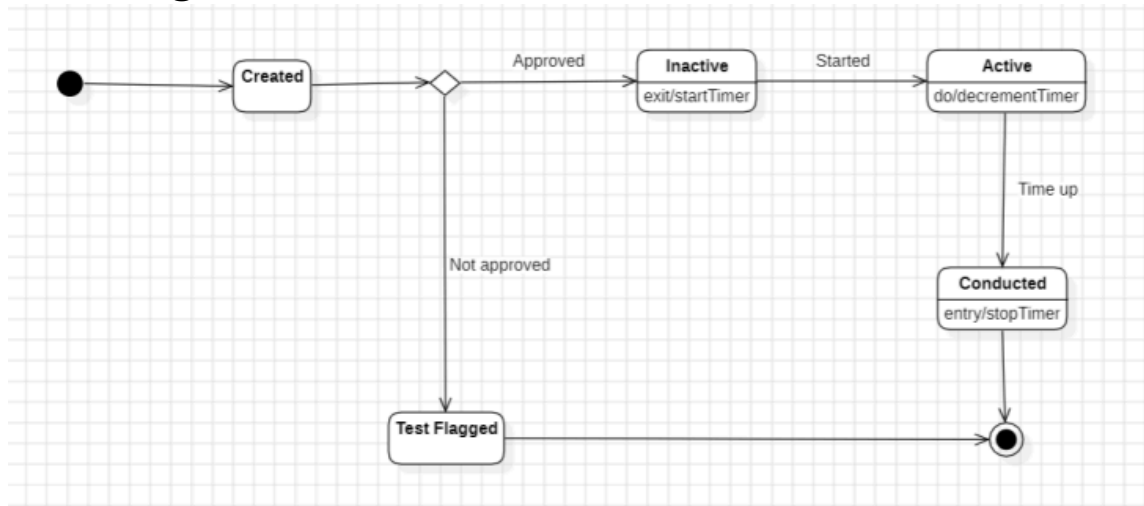
#### 4)Activity Diagram for Admin schedules test



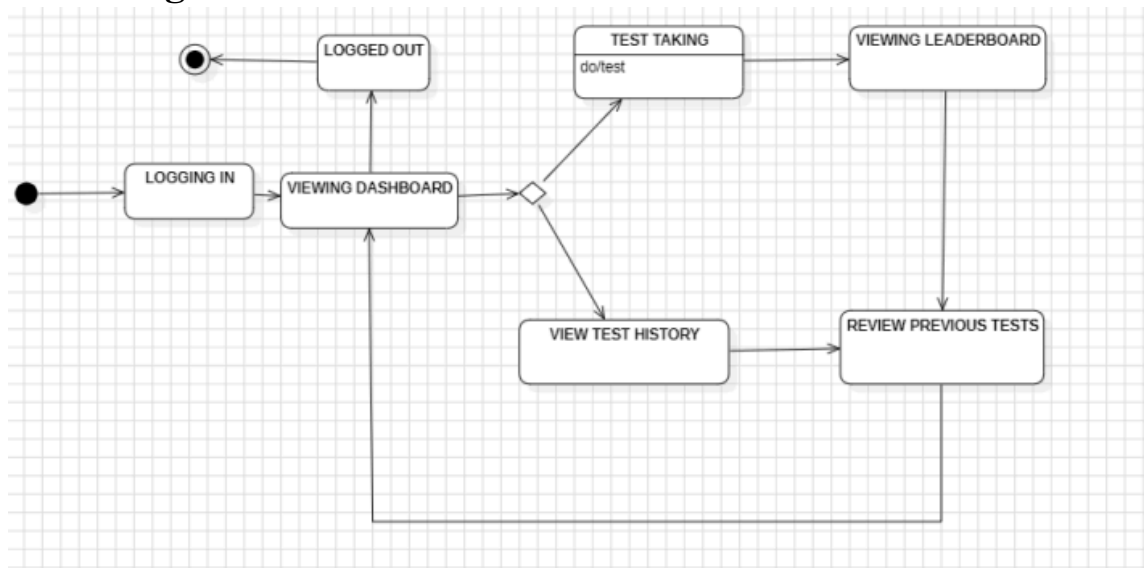


## State Diagrams:

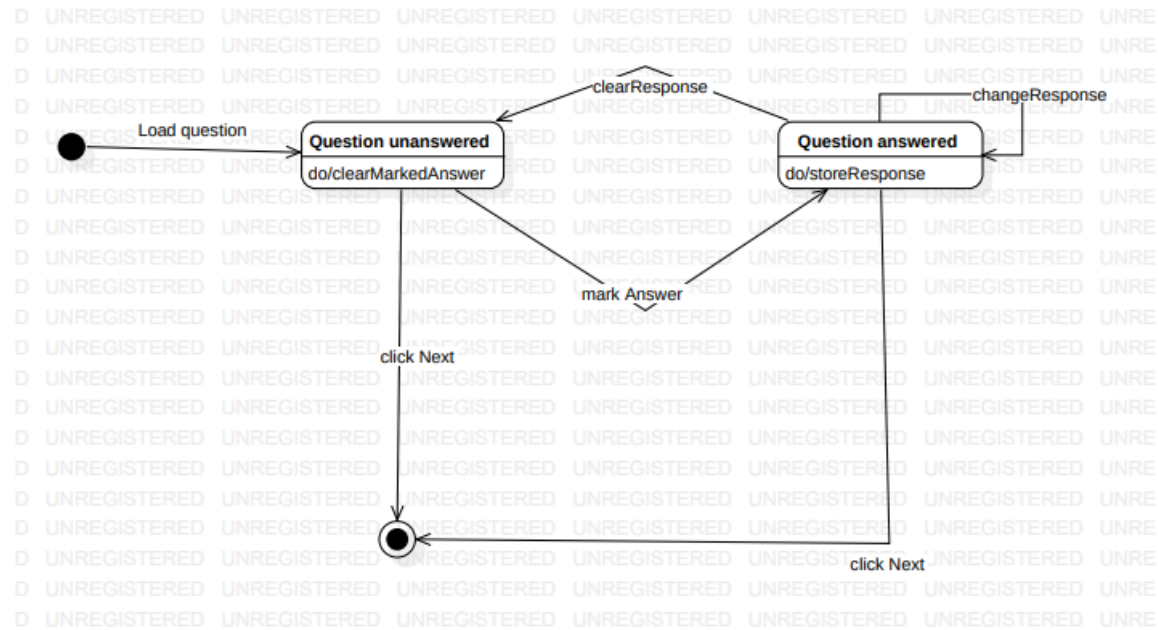
### State Diagram for Test Creation:



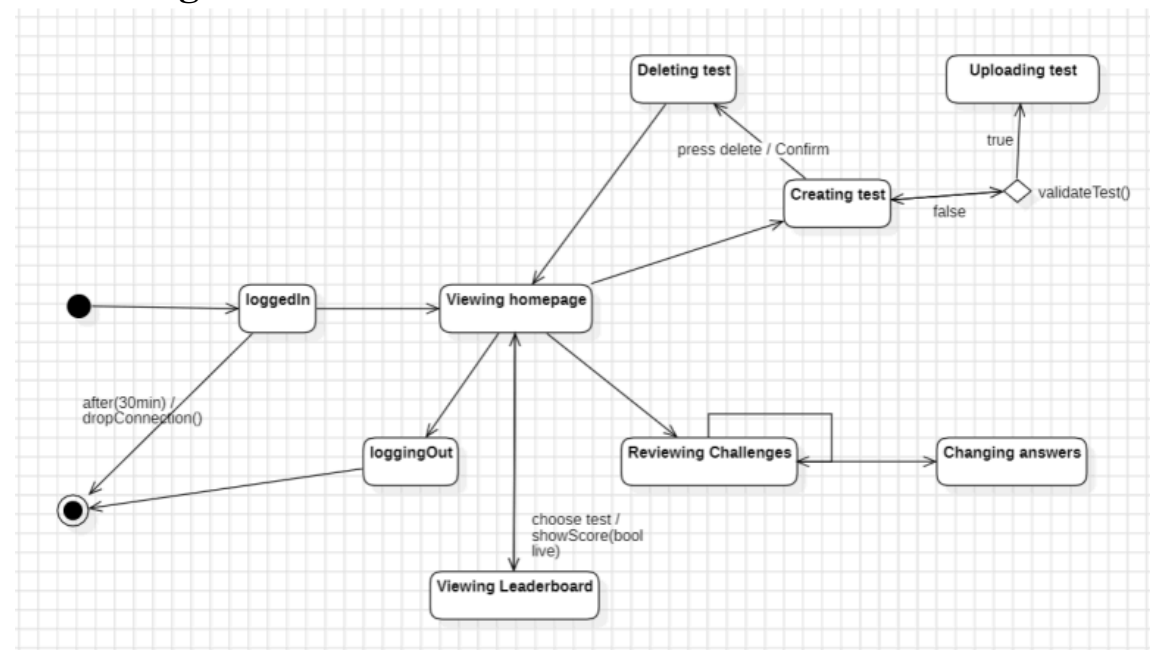
### State diagram for Student:



## State Diagram for Test taking:



## State Diagram for Professor:



## Architecture Patterns used:

- **Singleton Pattern (Database Connection):**

- The Singleton pattern is applied in the database connection module to ensure that only one instance of the database connection is created throughout the application's lifecycle.
- This ensures efficient resource utilization and prevents multiple connections to the database, which could lead to performance issues.
- By encapsulating the database connection logic within a Singleton class, we maintain a centralized point of access to the database, enhancing maintainability and scalability.

- **Factory Pattern (Registration Module):**

- The Factory pattern is utilized in the registration module to facilitate the creation of different types of user objects based on user input.
- By abstracting the object creation process into a factory class, we promote code flexibility and extensibility.
- For instance, the registration factory class dynamically creates instances of student or teacher objects based on the user's role selection during the registration process.
- This decouples the client code from the specific implementations of user objects, allowing for easy addition of new user types in the future.

- **Command Pattern (Admin Module):**

- The Command pattern is employed in the admin module to encapsulate requests as objects, thereby allowing the parameterization of clients with queues, requests, and operations.
- Each admin command corresponds to a specific action that an administrator can perform, such as approving tests, scheduling tests, or viewing statistics.
- By encapsulating these actions within command objects, we promote loose coupling between the invoker (administrator) and the receiver (system components).
- This enables us to easily extend the system with new admin functionalities without modifying existing code.

### **Design Principles used:**

- **Single Responsibility Principle (SRP):**

- The SRP is adhered to by designing classes with a single responsibility, focusing on specific use cases or functionalities.
- For example, separate classes are created for user authentication, test creation, test scheduling, and result generation.
- This promotes code clarity, maintainability, and reusability, as each class is responsible for a specific task, minimizing the impact of changes and facilitating easier debugging and testing.

- **Open-Closed Principle (OCP):**

- The OCP is observed by designing abstract classes and interfaces that are open for extension but closed for modification.
- For instance, the admin module consists of an abstract admin command class that defines the interface for executing admin actions.
- Concrete command classes extend this abstract class to implement specific admin functionalities.
- This design allows for the addition of new admin commands without altering existing code, promoting code stability and scalability.

- **Github link:**

[https://github.com/SinkableVirus/Online\\_Test\\_Management\\_System](https://github.com/SinkableVirus/Online_Test_Management_System)

## **INDIVIDUAL CONTRIBUTIONS :**

- Srinivas - Teacher Use case
- Srikrishna - Student Use case
- Suvan - Review Use case and Login In Parts
- Suraj - Admin Use case and leaderboard

# Screenshots :

## STUDENT :

### Student Registration :

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/studentRegister'. The page has a green header bar with the text 'Online Test Management System' and navigation links: 'Student Registration', 'Student Login', 'Teacher Registration', 'Teacher Login', and 'Admin Login'. The main content area features a 'Student Registration' form with the following fields: 'Enter Full Name', 'Enter SRN', 'Enter Branch', and 'Enter Password'. Each field is represented by a text input box. Below these fields is a blue 'Register' button. The browser's status bar at the bottom shows 'localhost:8080/studentRegister'.

localhost:8080/studentRegister

Online Test Management System Student Registration Student Login Teacher Registration Teacher Login Admin Login

Student Registration

Enter Full Name

Enter SRN

Enter Branch

Enter Password

Register

localhost:8080/studentRegister

## Student login :

The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The page title is 'Online Test Management System'. The navigation bar includes links for 'Student Registration', 'Student Login', 'Teacher Registration', 'Teacher Login', and 'Admin Login'. The main content area is titled 'Student Login' and contains a form with two input fields: 'Enter SRN' and 'Enter Password'. Below these fields is a blue 'Login' button.

Online Test Management System   Student Registration   Student Login   Teacher Registration   Teacher Login   Admin Login

### Student Login

Enter SRN

Enter Password

Login

## Student dashboard :

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/studentDashboard?SRN=123'. The page title is 'Student Dashboard'. On the left side, there are two green buttons: 'Logout' and 'View Previous Tests'. The main content area displays two test cards. Each card shows the difficulty level, start time, end time, and test description, followed by a green 'Take Test' button. The first card has a difficulty level of 'Medium' and a test description of 'Test description 1'. The second card has a difficulty level of 'Medium' and a test description of 'Test description 3'. Below the test cards, there is a section titled 'Java4' with a difficulty level of 'Medium'.

Logout   View Previous Tests

### Student Dashboard

Difficulty: Medium  
Start Time: 2024-04-12 08:00:00  
End Time: 2024-04-12 11:00:00  
Test description 1  
Take Test

Difficulty: Medium  
Start Time: 2024-04-12 09:00:00  
End Time: 2024-04-12 11:00:00  
Test description 3  
Take Test

**Java4**  
Difficulty: Medium



A screenshot of a web browser displaying a 'Questions' page. The browser's address bar shows 'localhost:3080/takeTest?SRN=SRN001&testId=TEST15'. The page has a dark header with navigation icons and a timer in the top right corner showing '0:4:58'. The main content area is light gray and contains a single question labeled '1) Question' with four radio button options: a, b, c, and d. A green 'Submit' button is located below the question options.

localhost:3080/viewPreviousTests?TSRN=SRIN001

Previous Tests

Test ID	Test Name	Marks Secured	Action
TEST001	Java	80	<a href="#">Review Test</a>
TEST002		85	<a href="#">Review Test</a>
TEST003		78	<a href="#">Review Test</a>
TEST004		92	<a href="#">Review Test</a>
TEST15		0	<a href="#">Review Test</a>

# REVIEW :

## Review Test :

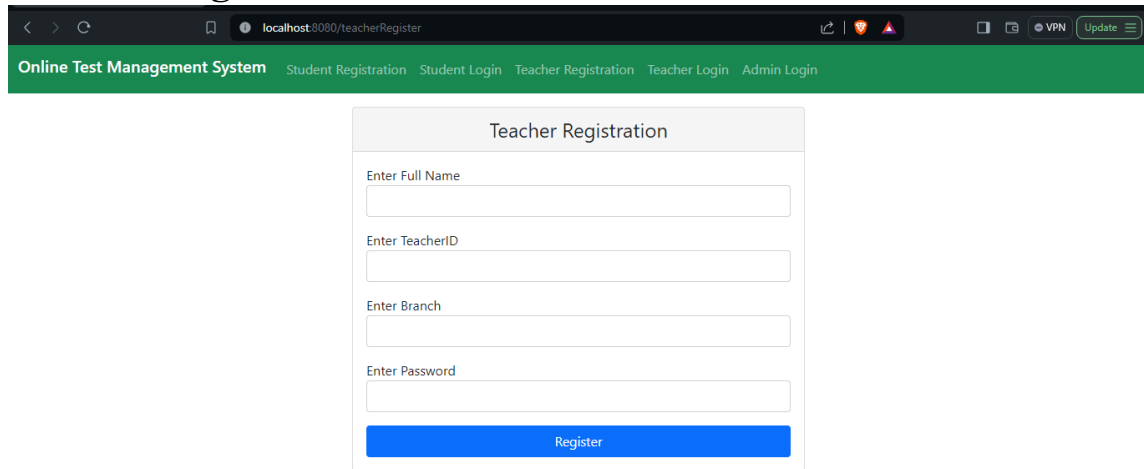


### Test Review

Question	Option 1	Option 2	Option 3	Option 4	Marked Answer	Correct Answer	Review Comment
What is the result of "9 + 3 * 2"?	15	18	24	12	a	a	<input type="text" value="None"/>
Which of the following is a valid way to create an object in Java?	ClassName obj = new ClassName();	ClassName obj = ClassName();	ClassName obj = new ClassName;	None of the above	a	a	<input type="text" value="None"/>
Which of the following is a Java code editor?	Eclipse	NetBeans	IntelliJ IDEA	All of the above	a	d	<input type="text" value="scam"/>
What is the purpose of the "static" keyword in Java?	To create a method that can be called without creating an object	To create a variable that is shared among all instances of a class	Both A and B	None of the above	c	c	<input type="text" value="None"/>
Which of the following is a Java collection interface?	List	Set	Map	All of the above	d	d	<input type="text" value="None"/>

## TEACHER :

### Teacher Registration :



The screenshot shows a web browser window with the address bar displaying "localhost:8080/teacherRegister". The page has a green header bar with the text "Online Test Management System" and a navigation menu with links: "Student Registration", "Student Login", "Teacher Registration", "Teacher Login", and "Admin Login". The main content area is a white box titled "Teacher Registration" containing four input fields: "Enter Full Name", "Enter TeacherID", "Enter Branch", and "Enter Password". A blue "Register" button is at the bottom.

Teacher Registration

Enter Full Name

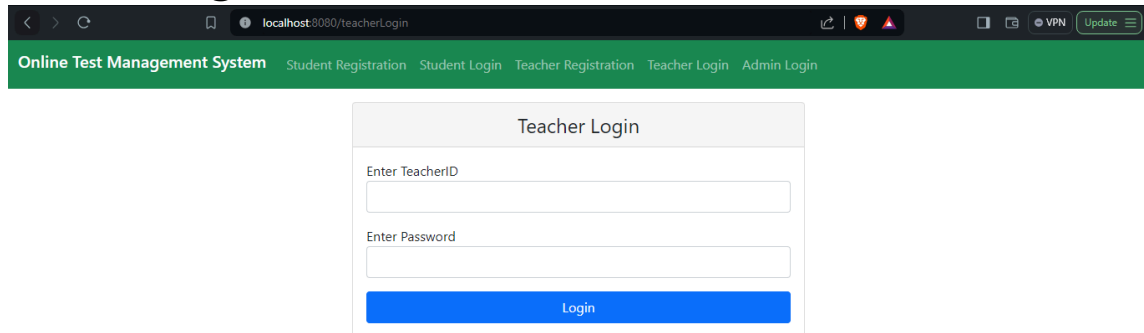
Enter TeacherID

Enter Branch

Enter Password

Register

### Teacher Login



The screenshot shows a web browser window with the address bar displaying "localhost:8080/teacherLogin". The page has a green header bar with the text "Online Test Management System" and a navigation menu with links: "Student Registration", "Student Login", "Teacher Registration", "Teacher Login", and "Admin Login". The main content area is a white box titled "Teacher Login" containing two input fields: "Enter TeacherID" and "Enter Password". A blue "Login" button is at the bottom.

Teacher Login

Enter TeacherID

Enter Password

Login

## Teacher Dashboard :

Logout

Upload Test

View Leaderboard

TEST002

See Reviews

TEST004

See Reviews

TEST007

See Reviews

## Teacher uploading test :

Back

Subject

Java

Difficulty

Medium

Start Date and Time

2024-04-22 11:05:00

End Date and Time

2024-04-22 11:10:00

Description

Test demo

1) Question

a) a

b) b

c) c

d) d

Marks: 3

Delete

Back

Marks: 3

Delete

Question:

Enter the question

a)

Enter option A

b)

Enter option B

c)

Enter option C

d)

Enter option D

Correct Option:

Enter the correct option

Marks:

Enter marks

Add

Submit

## Test reviewing for teacher :

Back

Test ID: TEST007

Question 1

What is the result of "9 + 3 \* 2"?

a) 15

b) 18

c) 24

d) 12

Student Reviews:

SRN	Review
123	None

Question 2

Which of the following is a valid way to create an object in Java?

a) `ClassName obj = new ClassName();`

b) `ClassName obj = ClassName();`

c) `ClassName obj = new ClassName;`

# General Leaderboard :

Test ID	Subject	Action
TEST001		<a href="#">View Details</a>
TEST002		<a href="#">View Details</a>
TEST003		<a href="#">View Details</a>
TEST004		<a href="#">View Details</a>
TEST006	Java1	<a href="#">View Details</a>
TEST007	Java2	<a href="#">View Details</a>
TEST008	Java3	<a href="#">View Details</a>
TEST009	Java4	<a href="#">View Details</a>
TEST010	Java5	<a href="#">View Details</a>

# Specific test Leaderboard :

Test Scores	
<b>Test Information</b>	
Test ID: TEST007	
Mean Score: 9.0	
Median Score: 9.0	
<b>Top Scores</b>	
Name	Score
first	9

**ADMIN :**

## Admin login :

Online Test Management System

[Student Registration](#) [Student Login](#) [Teacher Registration](#) [Teacher Login](#) [Admin Login](#)

Admin Login

Enter Admin ID

Enter Password

Login

### Pending tests for admin to approve :

localhost:3080/adminDashboard?adminId=AD001

Update

## Pending Tests

Test ID	Teacher ID	Description	Start Time	End Time	Difficulty	Update	Accept
TEST002	T002	Test description 2	2024-04-12 10:00:00	2024-04-12 12:00:00		Update	Accept
TEST004	T002	Test description 4	2024-04-12 11:00:00	2024-04-12 13:00:00		Update	Accept
TEST006	T001	Java Basics Test	2024-04-25 09:00:00	2024-04-25 11:00:00	Medium	Update	Accept
TEST008	T002	Collections and Generics Test	2024-04-29 10:00:00	2024-04-29 12:00:00	Easy	Update	Accept
TEST010	T002	Multithreading Test	2024-05-04 09:00:00	2024-05-04 11:00:00	Hard	Update	Accept
TEST012	T001	JDBC Test	2024-05-08 10:00:00	2024-05-08 12:00:00	Medium	Update	Accept
TEST014	T001	Servlets and JSP Test	2024-05-12 09:00:00	2024-05-12 11:00:00	Easy	Update	Accept

Undo Previous Update

# Tests accepted by admin :

Undo Previous Update

Accepted Tests

Test ID	Teacher ID	Description	Start Time	End Time	Difficulty	Undo
TEST001	T001	Test description 1	2024-04-12 08:00:00	2024-04-12 11:00:00		Undo
TEST003	T001	Test description 3	2024-04-12 09:00:00	2024-04-12 11:00:00		Undo
TEST007	T002	Object-Oriented Programming Test	2024-04-18 13:14:00	2024-04-18 14:00:00	Hard	Undo
TEST009	T002	Exception Handling Test	2024-05-02 14:00:00	2024-05-02 14:02:00	Medium	Undo
TEST011	T001	Input/Output Test	2024-05-06 13:00:00	2024-05-06 15:00:00	Easy	Undo
TEST013	T001	Networking Test	2024-05-10 14:00:00	2024-05-10 16:00:00	Hard	Undo
TEST015	T002	Java EE Test	2024-05-14 13:00:00	2024-05-14 15:00:00	Medium	Undo

Update Form with ID TEST006

Difficulty:

Hard

Start Date and Time:

09 / 07 / 2024 , 05 : 10 pm

End Date and Time:

10 / 07 / 2024 , 05 : 10 pm

Submit

TEST006	T001	Java Basics Test	2024-07-09 17:10:00	2024-07-10 17:10:00	Hard	Update	Accept
---------	------	------------------	---------------------	---------------------	------	--------	--------