**Project Name: Task Management System**

**Sprint Details:**

* **Total Sprints:** 2
* **Testing Phase:** Unit Testing, Integration Testing, and Security Testing
* **Objective:** Ensure functionality, integration, and security compliance of the Task Management System.

**Test Cases:**

**1. Unit Tests**

**Tested Components:**

* **Service Layer:** TaskService
* **Repository Layer:** TaskRepository

**File Structure Tree:**

A white background with black text

Description automatically generated

**Test Case Report Summary:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name** | **Test File** | **Description** | **Expected Result** | **Actual Result** |
| createTask\_ShouldSaveAndReturnTask | TaskServiceTest.java | Tests if the service can save and return a new task correctly | Task is saved and returned with correct details | Pass |
| getTaskById\_ShouldReturnTask\_WhenTaskExists | TaskServiceTest.java | Verifies retrieval of a task by ID when it exists | Task is retrieved with correct details | Pass |
| getTaskById\_ShouldThrowException\_WhenTaskDoesNotExist | TaskServiceTest.java | Checks exception handling when a task ID does not exist | Throws RuntimeException for non-existent task | Pass |
| updateTask\_ShouldModifyAndReturnUpdatedTask | TaskServiceTest.java | Tests the ability to modify an existing task | Task is updated with new details | Pass |
| deleteTask\_ShouldRemoveTask\_WhenTaskExists | TaskServiceTest.java | Verifies successful deletion of a task | Task is successfully deleted | Pass |
| saveTask\_ShouldPersistAndReturnTask | TaskRepositoryTest.java | Tests repository's ability to save and persist a task | Task is saved and persisted | Pass |
| findById\_ShouldReturnTask\_WhenTaskExists | TaskRepositoryTest.java | Verifies correct retrieval of a task by ID from the repository | Task is retrieved correctly | Pass |
| findById\_ShouldReturnEmptyOptional\_WhenTaskDoesNotExist | TaskRepositoryTest.java | Checks that the repository returns empty Optional for non-existent tasks | Returns empty Optional for non-existent task | Pass |
| deleteTask\_ShouldRemoveTask\_WhenTaskExists | TaskRepositoryTest.java | Tests repository deletion of a task | Task is removed and no longer exists | Pass |

**2. Integration Tests**

**Tested Components:**

* **Controller Layer:** TaskController
* **Repository Layer:** TaskRepository

**File Structure Tree:**

A computer screen shot of a code

Description automatically generated

**Test Case Report Summary:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name** | **Test File** | **Description** | **Expected Result** | **Actual Result** |
| createTask\_ShouldReturnCreatedTask | TaskControllerIntegrationTest.java | Tests API endpoint for creating a new task for a user | Returns created task with correct details | Pass |
| getAllTasksForUser\_ShouldReturnTaskList | TaskControllerIntegrationTest.java | Tests API endpoint to retrieve all tasks for a user | Returns list of tasks for the user | Pass |
| updateTask\_ShouldReturnUpdatedTask | TaskControllerIntegrationTest.java | Tests API endpoint for updating an existing task | Returns updated task with new details | Pass |
| deleteTask\_ShouldReturnSuccessMessage | TaskControllerIntegrationTest.java | Tests API endpoint for deleting a task | Confirms task deletion with success message | Pass |
| saveTask\_ShouldPersistAndReturnTask | TaskRepositoryIntegrationTest.java | Verifies repository's ability to persist and retrieve tasks | Persists and retrieves saved task | Pass |

**3. Security Tests**

**Tested Components:**

* Authentication and Authorization
* Input Validation and Data Security

**File Structure Tree:**

A screen shot of a computer program

Description automatically generated

**Test Case Report Summary:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name** | **Test File** | **Description** | **Expected Result** | **Actual Result** |
| accessWithoutToken\_ShouldReturnUnauthorized | AuthenticationTest.java | Tests access control when no authentication token is provided | Request is denied with 401 status | Pass |
| accessOtherUserTasks\_ShouldReturnForbidden | AuthorizationTest.java | Verifies authorization rules for accessing another user's tasks | Request is denied with 403 status | Pass |
| sqlInjectionAttempt\_ShouldReturnBadRequest | SqlInjectionTest.java | Simulates SQL injection attempt to test input validation | Request is rejected as bad input | Pass |
| response\_ShouldNotExposeSensitiveData | SensitiveDataExposureTest.java | Checks that sensitive data is not exposed in API responses | Sensitive data is not exposed in responses | Pass |
| invalidData\_ShouldReturnBadRequest | DataValidationTest.java | Verifies that the system rejects invalid input data | Request fails with 400 status due to invalid data | Pass |

**Security Tests Dependencies:**

* Ensure you have spring-boot-starter-test in your pom.xml or build.gradle for Spring Boot testing.
* Use MockMvc for HTTP request simulation.
* Use Jackson ObjectMapper for JSON serialization.

**Test Cases Output Log (using mvn test):**

A screenshot of a computer

Description automatically generated