

Contents**Filter****Description**[Field Summary](#)[Constructor Summary](#)[Method Summary](#)[Field Details](#)[db](#)[Constructor Details](#)[TestCases\(\)](#)[Method Details](#)[setupDatabase\(\)](#)[tearDownAll\(\)](#)[testStudentCanSubmitReview](#)[testInstructorCanFetchPending](#)[testInstructorCanApproveRequ](#)[testReviewerCanCreateReview](#)[testRetrieveAllReviewsForAns](#)

Class TestCases

[java.lang.Object](#)
[application.TestCases](#)

```
public class TestCases
extends Object
```

TestCases contains unit and integration tests for database operations within the CSE360 application. These tests verify the main user work flows:

- Student reviewer request submission
- Instructor reviewing and approving requests
- Reviewer creating and retrieving answer reviews

All tests use a shared DatabaseHelper instance \ The tests are designed for use with an in-memory H2 database or a local test database.

At this time clearing memory those not work. A conflict on User creation makes deleting dependents problematic.

Tables tested:

- cse360users
- reviewer_requests
- questions
- answers
- answer_reviews

Author:

Omar Diaz

Field Summary

Fields

Modifier and Type	Field	Description
private static databasePart1.Data	db	Shared database helper instance used for all tests.

Constructor Summary

Constructors

Constructor	Description
--------------------	--------------------

TestCases()

Method Summary

All Methods	Static Methods
--------------------	-----------------------

Instance Methods	Concrete Methods
-------------------------	-------------------------

Modifier and Type	Method	Description
(package private) static void	setupDatabase()	Initializes the test database connection and clears key tables before tests run.
(package private) static void	tearDownAll()	Closes the database connection after all tests have completed.
(package private) void	testInstructorCan	Verifies that an instructor can approve a pending reviewer request.
(package private) void	testInstructorCan	Verifies that an instructor can retrieve all pending reviewer requests.
(package private) void	testRetrieveAllRe	Tests that all reviews for a specific answer can be retrieved in order.
(package private) void	testReviewerCanCr	Tests that a reviewer can successfully insert a new

review for an answer.

```
(package  
private) void
```

testStudentCanSub Verifies that a student can successfully submit a reviewer request.



Methods inherited from class java.lang.Object

```
clone , equals , finalize , getClass , hashCode ,  
notify , notifyAll , toString , wait , wait ,  
wait
```

Field Details

db

```
private static databasePart1.DatabaseHelper db
```

Shared database helper instance used for all tests.

Constructor Details

TestCases

```
public TestCases()
```

Method Details

setupDatabase

```
@BeforeAll  
static void setupDatabase()  
throws SQLException
```

Initializes the test database connection and clears key tables before tests run. Unsure if clearing helps with testing.

Throws:

`SQLException` - if a database connection or statement fails

tearDownAll

```
@AfterAll  
static void tearDownAll()
```

Closes the database connection after all tests have completed.

testStudentCanSubmitReviewerRequest

```
@Test  
void testStudentCanSubmitReviewerRequest()  
throws  
SQLException
```

Verifies that a student can successfully submit a reviewer request.

- Ensures user accounts exist
- If user dose not exist, create them.
- Submits a reviewer request via `DatabaseHelper.submitReviewerRequest(java.lang.String)`
- Checks that the request is stored with status "Pending"

Throws:

`SQLException` - if a database access error occurs



testInstructorCanFetchPendingRequests

```
@Test  
void testInstructorCanFetchPendingRequests()  
throws  
SQLException
```

Verifies that an instructor can retrieve all pending reviewer requests.

- Ensures both instructor and user accounts exist
- If they don't exist, create them.
- Submits a reviewer request for the student
- Uses `DatabaseHelper.getPendingReviewerRequests()` to fetch pending requests
- Asserts that the student appears in the result list

Throws:

`SQLException` - if a database access error occurs

testInstructorCanApproveRequest

```
@Test  
void testInstructorCanApproveRequest()  
    throws  
SQLException
```

Verifies that an instructor can approve a pending reviewer request.

- Ensures both instructor and user accounts exist
- If they don't exist, create them.
- Student submits a reviewer request
- Instructor approves it using
`DatabaseHelper.processReviewerRequest(boolean, java.lang.String)`
- Confirms that the request is marked "Approved" and the user gains the "reviewer" role

Throws:

`SQLException` - if a database access error occurs

testReviewerCanCreateReview

```
@Test  
void testReviewerCanCreateReview()  
    throws SQLException
```

Tests that a reviewer can successfully insert a new review for an answer.

- Ensures reviewer accounts exist
- If they don't exist, create them.
- Creates a test question, and an answer
- Inserts a Review
- Validates that the review appears in the `answer_reviews` table

Throws:

`SQLException` - if a database access error occurs

testRetrieveAllReviewsForAnswer

```
@Test  
void testRetrieveAllReviewsForAnswer()
```

throws**SQLException**

Tests that all reviews for a specific answer can be retrieved in order.

- Ensures reviewer accounts exist
- If they don't exist, create them.
- Creates a question, answer, and two reviews
- Calls
`DatabaseHelper.getReviewsForAnswer(java.lang.String)`
- Verifies both reviews are returned in ascending creation order

Throws:

SQLException - if a database access error occurs

