

Unit Testing Classes

1. UT_AddUser.cs

Handles the addition of users to ensure only valid users are added while preventing duplicates.

- **Key Methods:**
 - **AddUser_UserDoesNotExist_ConfirmUserAddition:**
Verifies a new user is added successfully if they do not already exist in the system.
Process: Sets up valid user data, calls the **AddUser** method, and asserts that the method returns **true** and the user is now in the database.
 - **AddUser_UserAlreadyExists_RejectUserAddition:**
Ensures the system rejects adding duplicate users.
Process: Adds a user to the database, attempts to add the same user again, and asserts the method returns **false** without any changes to the database.
 - **AddUser_CheckUserExistence_BeforeAddition:**
Ensures the system checks for user existence before adding them.
Process: Verifies the user does not exist before calling **AddUser**, adds the user, and verifies they are in the database.

2. UT_Login.cs

Validates login functionality, ensuring proper authentication and handling of errors.

- **Key Methods:**
 - **Authenticate_ValidCredentials_ReturnsTrue:**
Ensures login succeeds when valid credentials are provided.
Process: Sets up valid username and password, calls the **Authenticate** method, and asserts it returns **true**.
 - **Authenticate_InvalidPassword_ReturnsFalse:**
Ensures login fails with an incorrect password.
Process: Sets up a valid username and invalid password, calls **Authenticate**, and asserts it returns **false**.
 - **Authenticate_ExceedsAllowedAttempts_ThrowsException:**
Ensures the system locks out users after too many failed login attempts.
Process: Simulates multiple failed login attempts and asserts an exception is thrown on the fourth attempt.

3. UT_Logout.cs

Tests the logout process and ensures proper session termination.

- **Key Methods:**
 - **Logout_EndsSession_SessionIsInactive:**
Verifies the session becomes inactive after logout.
Process: Logs in a user to activate a session, calls the **Logout** method, and asserts the session is inactive.
 - **Logout_WithoutActiveSession_DoesNotThrowException:**
Ensures calling logout without an active session does not result in errors.
Process: Calls **Logout** without logging in first and asserts no exceptions are thrown.

4. UT_RemoveStudent.cs

Tests the removal of students, ensuring proper validation and admin confirmation.

- **Key Methods:**
 - **RemoveStudent_StudentExists_ConfirmAndDelete:**
Ensures a student can be removed if they exist in the database.
Process: Adds a student, calls **RemoveStudent** with their ID, and asserts they are no longer in the database.
 - **RemoveStudent_StudentDoesNotExist_ReturnsFalse:**
Verifies the system returns **false** when trying to remove a non-existent student.
Process: Calls **RemoveStudent** with a non-existent ID and asserts the method returns **false**.
 - **RemoveStudent_ConfirmRemoval_DeletesStudent:**
Ensures admin confirmation is required before a student is removed.
Process: Adds a student, simulates admin confirmation, calls **RemoveStudent**, and verifies the student is deleted.

5. UT_RemoveUser.cs

Tests the removal of users with similar functionality to student removal.

- **Key Methods:**
 - **RemoveUser_UserExists_RemovedSuccessfully:**
Verifies a user can be removed if they exist.
Process: Adds a user, calls **RemoveUser**, and asserts the user is no longer in the database.

- **RemoveUser_UserDoesNotExist_ReturnsFalse:**
Ensures the system returns `false` when attempting to remove a non-existent user.
Process: Calls `RemoveUser` with a non-existent ID and asserts the method returns `false`.

6. UT_SearchStudent.cs

Validates student search functionality for various input conditions.

- **Key Methods:**
 - **SearchStudent_ValidCriteria_ReturnsResults:**
Ensures valid search criteria return matching students.
Process: Adds multiple students, calls `SearchStudent` with valid criteria (e.g., name), and asserts only matching results are returned.
 - **SearchStudent_InvalidCriteria_ReturnsEmptyResults:**
Ensures invalid criteria return no results.
Process: Searches for a non-existent name, asserting the result is an empty list.
 - **SearchStudent_EmptyCriteria_ReturnsAllStudents:**
Verifies all students are returned if no criteria are provided.
Process: Calls `SearchStudent` with empty criteria and asserts all students are returned.

7. UT_UpdateCourse.cs

Tests updating course details, including validation and permissions.

- **Key Methods:**
 - **UpdateCourse_ValidData_UpdatesSuccessfully:**
Verifies valid data updates the course.
Process: Updates a course's name and credits, and asserts the changes are reflected in the database.
 - **UpdateCourse_InvalidData_ReturnsFalse:**
Ensures invalid updates are rejected.
Process: Attempts to update a course with invalid data (e.g., empty name) and asserts the method returns `false`.
 - **UpdateCourse_NonExistentCourse_ReturnsFalse:**
Prevents updates for non-existent courses.
Process: Calls `UpdateCourse` with a non-existent ID and asserts the method returns `false`.
 - **UpdateCourse_UnauthorizedAccess_ThrowsException:**
Ensures updates are restricted to authorized users.

Process: Calls `UpdateCourse` without admin privileges and asserts an exception is thrown.

8. UT_UpdateUser.cs

Validates user credential updates, including authentication and validation.

- **Key Methods:**
 - **UpdateUser_ValidData_ConfirmUpdate:**
Verifies valid user details are updated successfully.
Process: Updates a user's name or email, and asserts the changes are reflected in the database.
 - **UpdateUser_InvalidData_RejectUpdate:**
Ensures invalid updates are rejected.
Process: Attempts to update a user with invalid data (e.g., empty fields) and asserts the method returns `false`.
 - **UpdateUser_UserNotAuthenticated_ThrowsException:**
Prevents updates for unauthenticated users.
Process: Calls `UpdateUser` without logging in and asserts an exception is thrown.
 - **UpdateUser_ValidationFails_RejectUpdate:**
Ensures updates fail if validation rules are not met (e.g., duplicate email).
Process: Updates a user with duplicate data and asserts the method returns `false`.