Test Cases

Black-Box Test Cases

These test cases are defined based on the available documentation and execution of the program. The code was not inspected.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	New user registration – validation check of username (only alphanumeric)	no	Click on the register button . Fill in the details – username should be non alphanumeric , rest all the fields should be filled correctly	The registration does not happen . A message should appear – Fill in all the fields correctly .	Pass
2	User Registration with correct fields .	no	Fill in the registration fields correctly – and click on the register button	Successful registration of the new user .	Pass
3	User Login	The user must already be registered .	Fill in the login credentials correctly – Username and the password and click on sign in button .	Successful login of the user .	Pass
4	New User Registration – Unique Username	A user with the same username should already exist .	Click on the register button , Fill in the credentials and the username should already have been registered .	Registration fails – User already exists , change the username	Pass
5	New User Registration – Password must be 6 characters long	no	Click on the register button , and fill in all the details , but the password must be less than 6 characters .	Registration fails – Password must be 6 characters long	Pass
6	New User Registration – Full name is required	no	Click on the register button and fill in all the fields , only the username field should be just one character long .	Registration fails – Enter full name	Fail – Even after filling just one character as name , the registration happens successfully .

7	New User Registration – Using an already registered email id	The email id should already be registered with some other username .	Click on the register button and fill in all the details just the email should be one that has already been registered.	Registration fails – Email is already registered.	Pass
8	User Login – Login with unregistered username	no	Fill in the details for login – use a new username which is not registered.	Login fails – User not registered.	Pass
9	User Login – Login with wrong password .	The user must already be registered	Fill in the login credentials – wrong password.	Login fails – Wrong Password.	Pass
10	User Login – Login with registered Email id	The user must already be registered	Fill in the login credentials	Successful Login	Pass
11	User Login – Login with invalid Email id	no	Fill in the login credentials – Fill invalid login id (without @ and .com)	Login fails – Enter valid Email ID	Pass
12	User Login – Missing fields	The user must already be registered	Fill in the one login credential correctly and leave other blank	Login fails –	Pass
13	User Login – Missing fields – suitable prompt message	The user must already be registered	Fill in the one login credential correctly and leave other blank	Login fails and prompt appears – Enter all the required fields	Fail – The prompt message that appears is not suitable.
14	Home Page – Functionality	Successful login	Search functionality to locate materials by author name, material name, or material ID.	The corresponding material is shown in the table .	Pass
15	Home Page – Functionality	Search the material in the search bar .	Reset button to clear search filters.	All the materials uploaded should be shown in the table .	Pass
16	Home Page – Functionality	Successful login	Ability to sort materials in ascending or descending order.	Availability of the option .	Fail – There is no option to arrange the material .
17	Upload Material – Functionality	Successful login	Upload the material by clicking on the upload option on the upload material page .	Material uploaded successfully .	Pass
18	Upload Material – Functionality	Successful login	Upload large material > 500 KB by clicking on the upload option on the upload material page .	Material is not uploaded because of large size .	Pass
19	Edit / Delete Material – Functionality	Material has already been uploaded by the user .	Click on the Edit / delete option to edit or delete the already uploaded material .	Users can delete their own uploaded materials but cannot	Pass

				delete materials uploaded by others.	
20	Edit User Profile – Password	Successful login	Go to Edit Profile option and enter the new password , then click on save button , then log out and login again with the new credentials .	Successful login with the new password .	Pass
21	Edit User Profile – Email Id	Successful login	Go to Edit Profile option and enter the new email id , then click on save button , then log out and login again with the new credentials .	Successful login with the new email id .	Pass
22	Edit User Profile – Username	Successful login	Go to Edit Profile option and enter the new username , then click on save button , then log out and login again with the new credentials .	Successful login with the new username .	Fail – This only changes the name of the user , but during login credentials the user have the use the old username only for logging in .
23	Logout	Successful login	Go to the Logout Tab , and click on the log out button	Successful Log out and the user is redirected to the login page	Pass

White-Box Test Cases

These additional test cases were defined during inspection of the code.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation (“pass” or failure description)
1	testRegisterUser_Success : Test if the User with all the valid input fields is successfully registered.	UserService initialized with userDao mocked	1. Mock UserDao.addUser() to return true. 2. Call userService.registerUser() with valid unique email, username, full name, and password.	Registration succeeds.	Test passes. User registered successfully message is printed.
2	testRegisterUser_Failure_UsernameTaken : Test if the user with already existing email or user name is allowed to register again or not .	UserService initialized with userDao mocked	1. Mock UserDao.addUser() to return false. 2. Call userService.registerUser() with an email or username that already exists	Registration fails due to username or email already taken	Test passes. Username or Email is already taken!! message is printed.
3	testRegisterUser_Exception : To test the Exception handling .	UserService initialized with userDao mocked	1. Mock UserDao.addUser() to throw SQLException. 2. Call userService.registerUser() with valid email, username, full name, and password	Registration fails due to SQLException	Test passes. SQLException is caught and handled correctly.

4	testLoginUser_Success : To test the successful user login	Mock ‘userDaoMock.getUserByUsername’ To return a ‘User’	Call ‘userService.loginUser’ with correct username and password.	User login should succeed (result is successful and the data of user should match with the expected user)	Pass
5	testLoginUser_IncorrectPassword : To test the User login failure due to the incorrect password	Mock userDaoMock.getUserByUsername to return a User	Call userService.loginUser with correct username and incorrect password.	User login should fail due to incorrect password (result is unsuccessful and user data is null).	Pass
6	testLoginUser_UserNotFound : User login failure due to user not found	Mock userDaoMock.getUserByUsername to return null	Call userService.loginUser with a username that does not exist.	User login should fail due to user not found (result is unsuccessful and user data is null).	Pass
7	testLoginUser_Exception : To test User login failure due to SQL Exception	Mock userDaoMock.getUserByUsername to throw SQLException	Call userService.loginUser with a valid username.	User login should fail due to an exception	Pass

				(result is unsuccessful and user data is null).	
8	testEditProfile_Success : tests user login edit	Mock userDaoMock.editProfile to return true	Call userService.editProfile with valid data. 2. Mock userDaoMock.getUserByUsername to return User with updated details.	Profile edit should succeed (result is successful and user data matches updated details).	Pass
9	testEditProfile_UsernameTaken : Tests user edit failure due to already existing username or email	Mock userDaoMock.editProfile to return false	Call userService.editProfile with an already taken username or email	Profile edit should fail due to username or email already taken (result is unsuccessful and user data is null).	Pass
10	testEditProfile_Exception : Tests user profile edit failure due to SQL exception	Mock userDaoMock.editProfile to throw SQLException	Call userService.editProfile with valid user details	Profile edit should fail due to an exception (result is unsuccessful and user	Pass

				data is null).	
11	Test Material Uploaded	userDaoMock.Material	Call Material ID authorName uploadedDate filleName	It should show the data of the uploaded image. i.e., the name of user who uploaded it, uploaded time stamp, filename.	Pass
12	testUser	userDaoMock.User	Call ID Email Username fullName password	It should get the ID, Email, Username, fullname and password of the new user to the old user	Pass