

CSCI 3060U

Phase 5

Aqib Alam, Noor Alam, Shane Francis, Ethan
Jacalan

Our Main Root directory for Phase5 is called CSCI3060UProject\SQAPhase5

The Directory all the main files are in :

CSCI3060UProject\SQAPhase5\app\src\main\java\SQAPhase4

and the Test files are all in :

CSCI3060UProject\SQAPhase5\app\src\test\java\SQAPhase4

Test Cases:

Test Names	Descriptions
test_Bid	It makes sure that the system can make bids on an item
test_Advertise	Make sure advertise can add items to the available items.
test_AddUser	Making sure that the user function adds a new user with their information into the user accounts file
test_DeleteUser	Ensure that the delete user function removes a user's account from the user account file.
test_addCredit	Tests the add credit function will add credit to the user account.
test_DecAuctionDays	Make sure that the decAuctionFunction decreases the number of days left for the item.
test_ParseLine	Make sure that each parses each transaction correctly.
test_Refund	Makes sure that the refund function returns credits from the seller to the buyer.

Changes:

Test Name	Description	Fix
test_addUser	Test if a new user is added with valid data and fails with invalid data.	Verify input validation, correct handling of input data, and error messages in addUser().
test_deleteUser	Test if a user is deleted with a valid user ID and fails with an invalid ID.	Check user ID validation, proper handling of non-existent IDs, and error messages in deleteUser().
test_addCredit	Test if adding credit works with valid data and fails with invalid data.	Confirm proper credit handling, input data validation, and error messages in addCredit().
test_refund	Test if a refund is processed with valid data and fails with invalid data.	Review refund processing logic, data validation, and error messages in refund().
test_bid	Test if placing a bid works with valid data and fails with invalid data.	Inspect bid handling, input data validation, and error messages in bid().
test_advertise	Test if creating an advertisement works with valid data and fails with invalid.	Examine ad creation logic, data validation, and error messages in advertise().
test_addSpaces	Test if spaces are added correctly for formatting and fail with invalid data.	Analyze string formatting logic, input validation, and error messages in addSpaces().
test_formatAmount	Test if formatting an amount works with valid data and fails with invalid data.	Assess amount formatting logic, input validation, and error messages in formatAmount().

Path Coverage Testing:

Test Name	Description	Fix
test_credituserExists1	Add credit to an existing user without exceeding the credit limit	Ensure the method “addCredit” correctly updates the user's credit
test_credituserExists1	Add credit to an existing user, but exceed the credit limit	Add a check in the “addCredit” method to prevent credit limit violations
test_AddCreditUserNotExists	Attempt to add credit to a non-existent user	Verify that the “addCredit” method returns -1 when the user does not exist

Inside our tests folder, we create a file called `CurrentUserAccountTest.java`, where there is a set of J-unit test which is responsible for path coverage for the `addCredit` method was created where the following possible execution paths within the method was accounted for:

1. When the user exists and adding credit does not exceed the limit.
2. When the user exists and adding credit exceeds the limit.
3. When the user does not exist.