

Test Case Descriptions

Test Name	Description
Login_1 - Success	The objective is to verify the initiation of a front-end session that fulfills all essential prerequisites. This involves selecting the login option and providing a valid username and password.
Login_2 - user is already logged in	To verify the initiation of a front-end session for a user who has already logged in and is currently running the session.
Login_3 - failed authentication	To verify the behavior of the front-end session when the user selects the login option but enters an incorrect or unrecognized username and/or password during login.
Login_4 - Non admin login	To verify that a front-end session initiated from a non-administrative account does not allow privileged transactions.
Login_5 - Admin login	To verify the start of a front end session from an admin account allows all transactions
Logout_1 - user is logged in	Verify the completion of a front end session where the logout option is selected and all necessary requirements are met.
Logout_2 - user is not logged in	To test the proper implementation of the logout feature when the user selects the logout option but is not currently logged into a front end session.
Logout_3 - Transaction after logout	To test the end of a front end session in which the user attempts to perform a transaction after having logged out of their account
Create_1 - Admin Success	To test the successful creation of a new account with bidding and/or selling privileges. A valid username is provided and account type (admin, full-standard... etc) is selected. The user creating the new account is currently logged into an admin account
create_2 - Admin fail (missing account info)	To test the creation of a new account by an admin, but with incomplete information provided. The username and/or account type are missing, and the system should reject the account creation until all required fields are filled out.

Create_3 - Username already exists	To verify the creation of a new account by an admin but has supplied a username that is already registered for another pre-existing account
Create_4 - Username exceeds character limit	To verify the creation of a new account by an admin but has supplied a username that exceeds the maximum character count of 15
Create_5 - Non-admin	To test the creation of a new account in which the user is a non-admin
Delete_1 - Success	To verify the deletion of an existing account by an admin user, ensuring that the account being deleted is verified as pre-existing and is not currently logged in by the requesting admin user.
Delete_2 - Not a pre-existing account	To test the deletion of an account in which the username provided does not belong to any pre-existing accounts. User requesting the deletion is also an admin.
Delete_3 - Can't be the same name as the current user	To test the deletion of an account in which the username provided belongs to the current user requesting the deletion
Advertise_1 - Success	To verify the successful addition of an item to the auction board, ensure the user is logged in with a valid account and that all the information related to the item and its bidding is accurate.
Advertise_2 - userType is standard-buy	To verify that the advertise function properly restricts the ability of a "standard-buy" account to put up an item for auction.
Advertise_3 - exceeds item character limit (25)	To test that the advertise function rejects an attempt to advertise an item that exceeds the character limit
bid_1 - userType is not standard-sell	To test a successful bid when the user is logged into a compatible account and enters a valid itemName, sellerName and bid
bid_2 - userType is standard-sell	To test that a standard-sell account cannot bid on items
bid_3 - missing info	To test that this function fails when the required information is not provided

bid_4 - item given does not exist	To test that this function fails when given a non-existent item name
refund_1 - Success	To test the successful refund of credit to a buyer's account from the seller's account
refund_2 - Missing user info	To test that the function rejects the refund if the buyer's username and/or the seller's username and/or the amount to transfer is missing
refund_3 - Buyer/Seller not in database	To test that the function ends if neither the given buyer or seller exist in the database
addCredit_1 - Success	To test the successful addition of credit from an admin account into the system for the purchase of accounts
addCredit_2 - Success, Non-Admin	To test that the successful addition of credit from a non-admin account into the system for the purchase of accounts
addCredit3 - Invalid Username	To test that the function rejects an attempt to add credit while in admin mode, if the username given does not

Side note: Not all constraints test cases are shown in the table but they will be tested. We will also test the daily transaction file and the user account file to verify that it updates after each user transaction.

Test Plan

How our tests are organized:

Our tests will be organized using the following directory layout: A folder named Tests which will contain folders named after our different transactions such as Login and AddCredit. Inside of each of those folders, there will be three additional folders that will hold the .txt files for inputs, outputs and daily transactions.

File Structure Example: --->Tests

```

--->Login
    ---> LoginInputs
        --->Login1.txt
    --->LoginOutputs
        --->Login1.txt
    --->LoginTransactions
        --->Login1.txt
--->AddCredit
    ---> AddCreditInputs

```

--->AddCredit1.txt
--->AddCreditOutputs
--->AddCredit.txt
--->AddCreditTransactions
--->AddCredit.txt

How Our Tests will be run:

We will be running our test cases with shell scripts.

How Our Test outputs will be stored and organized:

The output of our tests will be used for comparison for later runs will be stored and organized using the following directory layout:

Example: Results ---> AddCredit--->AddCreditResults ---> AddCreditout, AddCreditScript