

Software Quality Assurance
CSCI 3060U
Phase 1: Front End Requirements

Group Name:

Jessica (100657595)

Terry Darmawan Hosea (100670290)

Pulkit Madan (100674520)

Teaching Assistant Name: **Jude Arokiam**

Phase 1 Requirements

*Bolted words are available transactions

Github: https://github.com/terryhosea/csci3060u-project/tree/master/phase_1

Transaction	Test Name	Intention	Input	Expected Output
login	login1	Valid login case - Test to read user account file first before accepting any user logins - Test to prompt the user to enter username after login transaction - Test to accept login transaction if user is in user account file - Test to read available items file after successful login and begin accepting transactions	login valid_user	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Item File Welcome valid_user!
	login2	Test to reject invalid user that is not in the user account file	login invalid_user	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>ERROR</i>] Invalid username
	login3	Test to check that no transaction is accepted before user is logged in.	bid	[<i>DONE</i>] Read User Account File Welcome to auction site! [<i>ERROR</i>] Not logged in
	login4	Test to check login is not accepted after successful login	login valid_user login	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File

[illegible]

		performed after a successful login - test to check that program is writing transaction the daily transaction file.	valid_user logout	Enter username: [<i>DONE</i>] Read Available Items File Welcome valid_user! [<i>DONE</i>] Write Daily Transaction File
	logout2	To test that user has to be logged in before logging out	logout	[<i>DONE</i>] Read User Account File Welcome to auction site! [<i>ERROR</i>] Not logged in
	logout3	Only login transaction accepted when logged out	login valid_user logout bid	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome valid_user! [<i>DONE</i>] Write Daily Transaction File Welcome to auction site! [<i>ERROR</i>] Not logged in
	logout4	To test if Daily Transaction File successfully written after logging out	login valid_user advertise item_name valid_bid valid_days	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter item name: Enter minimum bid: Enter days of auction:

			logout	<p>[<i>DONE</i>] Advertisement created</p> <p>[<i>WRITE</i>] Write to Daily Transaction File</p> <p>[<i>ERROR</i>] Error saving transaction log to Daily Transaction File</p>
create	create1	<p>Valid create case</p> <ul style="list-style-type: none"> - Test to prompt user for new username - After user enters username, test for prompt that asks for user type. - Test to prompt user to enter amount of initial credit - Test to store new user information which will be saved to daily transaction file when session ends 	<p>login admin</p> <p>create new_user</p> <p>SS 1</p>	<p>[<i>DONE</i>] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[<i>DONE</i>] Read Available Items File</p> <p>Welcome admin! Enter new username: User types AA : Admin FS : Full-standard BS : buy-standard SS : Sell Standard</p> <p>Enter user type: Enter credit value:</p> <p>[<i>DONE</i>] User Created</p>
	create2	Test that check that this transaction is valid only when the user type is admin.	<p>login reg_user</p> <p>create</p>	<p>[<i>DONE</i>] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[<i>DONE</i>] Read Available Items File</p> <p>Welcome reg_user!</p> <p>[<i>ERROR</i>] Account not privileged!</p>
	create3	To test that new username should be at most 15		[<i>DONE</i>] Read User Account File

		characters	login admin create 16_characters_username	Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter new username: [<i>ERROR</i>] Username has a 15 character limit
	create4	To test that new username is unique	login admin create existing_user	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter new username: [<i>ERROR</i>] Username has been taken
	create5	To test if enter user type in the create function is a valid user type.	login admin create new_user invalid_type	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter new username: User types AA : Admin FS : Full-standard BS : buy-standard SS : Sell Standard Enter user type:

[illegible]

		<ul style="list-style-type: none"> -Test to prompt the user to enter item name - Test to prompt the user to enter minimum bid - Test that the minimum bid is not negative - Test to prompt the user to enter the number of days until the auction ends - test that # of days entered is between in range [0. 100] (ends are inclusive). - Test to store new advertisement information which will be saved to daily transaction file when session ends 	login reg_user advertise item_name 100 10	Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome reg_user! Enter item name: Enter minimum bid: Enter days of auction: [DONE] Advertisement created
	advertise2	To test that advertise transaction is available for any user type except standard-buy	login existing_user advertise	[DONE] Read User Account File Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome stdbuy_user! [ERROR] Account not privileged!
	advertise3	To test if item name is at most 19 characters	login admin advertise 19_characters_items	[DONE] Read User Account File Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome admin! Enter item name: [ERROR] Item name has a 19 character limit

	advertise4	To test if bidding price is more than maximum allowed (999.99)	login admin advertise Item_name 1000	[DONE] Read User Account File Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome admin! Enter item name: Enter minimum bid: [ERROR] Maximum bidding is 999.99
	advertise5	To test if advertise duration is less than 100 days	login admin advertise item_name 1000 101	[DONE] Read User Account File Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome admin! Enter item name: Enter minimum bid: Enter days of auction: [ERROR] Maximum auction days is 100
	advertise6	To test if advertise duration is at least 1 day	login admin advertise item_name 1000 101 0	[DONE] Read User Account File Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome admin! Enter item name: Enter minimum bid: Enter days of auction: [ERROR] Minimum auction

				days is 1
bid	bid1	Valid bid case - Test to prompt the user to enter the item's name - Test to prompt the user to enter the seller's name - Test to display the current highest bid and prompt the user for a new bid amount - Test to store new bid information which will be saved to daily transaction file when session ends	login admin bid item_name reg_user 11	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter item name: Enter seller's username: Current highest bid is \$10 Please enter new bid amount: [<i>DONE</i>] Bid placed
	bid2	To test that bid transaction is available for any user type except standard-sell	login reg_user bid	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome stdsell_user! [<i>ERROR</i>] Account not privileged!
	bid3	To test if item name is valid/available	login admin bid invalid_item reg_user	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter item name: Enter seller's username: [<i>ERROR</i>] Item unavailable

	bid4	To test if seller name is valid/available	<p>login admin</p> <p>bid item_name invalid_seller</p>	<p>[DONE] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[DONE] Read Available Items File</p> <p>Welcome admin! Enter item name: Enter seller's username:</p> <p>[ERROR] Invalid seller username</p>
	bid5	To test whether new bid is greater than the previous highest bid	<p>login admin</p> <p>bid item_name reg_user</p> <p>9</p>	<p>[DONE] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[DONE] Read Available Items File</p> <p>Welcome admin! Enter item name: Enter seller's username:</p> <p>Current highest bid is \$10 Please enter new bid amount:</p> <p>[ERROR] New bid should be greater than old bid</p>
	bid6	To test whether new bid is at least 5% higher than the previous highest bid for non-admin users	<p>login existing_user</p> <p>bid item_name reg_user</p>	<p>[DONE] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[DONE] Read Available Items File</p> <p>Welcome reg_user! Enter item name: Enter seller's username:</p>

			10.25	<p>Current highest bid is \$10 Please enter new bid amount:</p> <p>[<i>ERROR</i>] New bid should be at least 5% higher than the previous highest bid</p>
	bid7	To test whether buyer has sufficient balance to bid	<p>login admin</p> <p>bid item_name reg_user</p> <p>11</p>	<p>[<i>DONE</i>] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[<i>DONE</i>] Read Available Items File</p> <p>Welcome admin! Enter item name: Enter seller's username:</p> <p>Current highest bid is \$10 Please enter new bid amount: [<i>ERROR</i>] Insufficient funds</p>
	bid8	To test whether new bidding price exceeds 999.99 after new bid is placed	<p>login admin</p> <p>bid item_name reg_user</p> <p>2</p>	<p>[<i>DONE</i>] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[<i>DONE</i>] Read Available Items File</p> <p>Welcome admin! Enter item name: Enter seller's username:</p> <p>Current highest bid is \$998.99 Please enter new bid amount: [<i>ERROR</i>] Item price limit reached</p>
	bid9	To test whether new bid price is a float and non negative		[<i>DONE</i>] Read User Account File

			login admin bid item_name reg_user one / -1	Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter item name: Enter seller's username: Current highest bid is \$998.99 Please enter new bid amount: [<i>ERROR</i>] Invalid input
	bid10	To test that a seller can't bid on their own items	login admin bid item_name reg_user	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter item name: Enter seller's username: [<i>ERROR</i>] Can't bid on your own item
refund	refund1	Valid refund case -Test to prompt the user to enter buyer's username - Test to prompt the user to enter seller's username - Test to prompt the user to enter the amount of credit to transfer - Test that transferred a specified amount from the seller's credit to the buyer's credit - Test to store refund information which will be saved to daily transaction file when session ends	login admin refund reg_user existing_user 100	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter seller's username: Enter buyer's username: Enter amount of credit to be transferred: [<i>DONE</i>] Funds refunded

	refund2	To test if non privileged account called refund transaction	<p>login reg_user</p> <p>refund'</p>	<p>[DONE] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[DONE] Read Available Items File</p> <p>Welcome reg_user! [ERROR]: Account is not privileged!</p>
	refund3	To test if seller username exists	<p>login admin</p> <p>refund invalid_seller existing_user 100</p>	<p>[DONE] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[DONE] Read Available Items File</p> <p>Welcome admin! Enter seller's username: Enter buyer's username: Enter amount of credit to be transferred: [ERROR] Seller does not exist</p>
	refund4	To test if buyer username exists	<p>login admin</p> <p>refund reg_user existing_user 100</p>	<p>[DONE] Read User Account File</p> <p>Welcome to auction site! Enter username:</p> <p>[DONE] Read Available Items File</p> <p>Welcome admin! Enter seller's username: Enter buyer's username: Enter amount of credit to be transferred: [ERROR] Buyer does not exist</p>

	refund5	To test whether seller credit is sufficient to process refund to the buyer	login admin refund reg_user existing_user 100	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter seller's username: Enter buyer's username: Enter amount of credit to be transferred: [<i>ERROR</i>] Seller has insufficient funds
	refund6	To test whether buyer's credit will hit the limit after refund is issued	login admin refund reg_user existing_user 100	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter seller's username: Enter buyer's username: Enter amount of credit to be transferred: [<i>ERROR</i>] Buyer's credit has hit the limit
addcredit	addcredit1	Valid addCredit case for an admin user -Test to prompt the user to enter the amount of credit - Test to prompt the user to enter the username of the account to which credit is being added - Test to write to transaction stream file - Test to store addcredit information which will be	login admin addcredit 100 reg_user	[<i>DONE</i>] Read User Account File Welcome to auction site! Enter username: [<i>DONE</i>] Read Available Items File Welcome admin! Enter amount: Enter target username:

	addcredit5	To test if account's credit will hit limit after credit added	login admin addcredit 999 reg_user	[DONE] Read User Account File Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome admin! Enter amount: Enter target username: [ERROR] Hit credit limit
	addcredit6	Valid addcredit case for a non admin user -Test to prompt the user to enter the amount of credit - Test to write to transaction stream file - Should save this information to the daily transaction file	login reg_user addcredit 100	[DONE] Read User Account File Welcome to auction site! Enter username: [DONE] Read Available Items File Welcome reg_user! Enter amount: [DONE] Credit added

Test Plan

Test Input and Output Organization

We will be testing our program against our test cases which is organized under directory called "phase1". Under the directory, we have four main folders which are input, actual_output, expected_output, and result.

Where everything under input folder is the input, actual_output folder will contain all the files that is generated after the test is conducted against the program and record the terminal output of it, expected outputs contains all the output files we expected to be compared with actual_output, and results folder contains all the result files after comparing actual_output and expected_output.

Directory Structure:

```
phase_1
- input
  - addcredit1.txt
  - addcredit2.txt
  - login1.txt
  - login2.txt
  .
  .
  .
- expected_output
  - advertise1.dat
  - advertise2.dat
  - addcredit1.dat
  - addcredit2.dat
  .
  .
  .
- actual_output
  - advertise1.dat
  - advertise2.dat
  - addcredit1.dat
  .
  .
  .
- results
  - result_login1.dat
  - .....
```

Test Run Plan

The test will be run using shell scripts where it will open the front-end and start going through the test cases by giving inputs e.g. *program < login1.txt* this will more or less the formats we will do during the testing.

The test case will run the commands from the input file and generate an actual-output file (at later phase when we have the code). The test case will then compare the actual-output file and the expected-output file, display the result and generate a *result_testCaseName.dat* placed under results folder. This file, will have the result of the testcase

The original program will resume once the testing command is completed and the result is displayed.

We will test all scenarios both the valid cases and invalid cases, i.e, where we are expecting error(s). The test will be run against our program during development phase and another run once the front-end and more elements of the program is setup.

Requirements Problems

The program should not allow blank usernames. Transaction codes should be excluded from eligible usernames. Password is not required for login. Newly created account will be assigned only one account type and amount of credit greater than 0 by admin. When user input is rejected, transaction will be cancelled.

An account cannot have negative balance and maximum credit can contain is 999,999. User has to have enough credit to make a bid. Seller cannot make a bid on their own items. In order to do refund, seller has to have enough amount of credit.

There is no constraint about how long a session can be as well so we are never logging out a user. There is also no constraint if there are multiple sessions opened by multiple users which should be included in the consideration (especially later on) to avoid problem writing transactions user has done. The maximum length of username for both buyer and seller is 15 characters for both all files and transaction. The maximum length for an item is 19 characters for all files and transactions.

In case there is an invalid transaction/operation, the output is not written to the daily transaction file when the user logouts.