AuctionBay

Project Phase Three

CSCI3060U, Software Quality Assurance

Miljanovic, M.
Ontario Tech University, Faculty of Science
March 06, 2023

Tuesday Lab (CRN: 72674):

Rija Baig (100746674)
Preet Panchal (100707094)
Eihab Syed (100707448)
Nathaniel Tai (100662284)

AUCTIONBAY

TABLE OF CONTENTS

AUCTIONBAY

1.0. INTRODUCTION

AuctionBay is an Auction-style Sales Service. The system consists of two parts: the Front End, a point of purchase terminal for bidding and advertising items and the Back End, an overnight batch processor to maintain and update a master auctions file intended to run each day at midnight. This is Phase 3 of the project which contains the front end requirements testing for AuctionBay. This document will include the testing log table which shows and explains pass and fail in different test cases.

2.0. METHOD OF SCRIPTING

For the automation of running all of our test cases, our team is using a Bash script that runs directly from the UNIX shell prompt on the terminal. Our script is able to access the src folder to run the makefile before each script run, as well as run the program using the same command from Phase 2. The script presets four users in the current_user_accounts.txt file and four items in the available_items.txt file. The script runs and accesses the ‘testcases/inputs’ directory to run the inputs for each test case in the program. This is run via test.sh script. Our test.sh calls the compare.sh script to compare the expected outputs found in ‘testcases/outputs/expected’ with the actual generated outputs found in ‘testcases/outputs/actual’ and creates files that represent the differences between the outputs found in ‘testcases/outputs/differences’. The compare.sh script also compares the daily transaction files for each test case from ‘testcases/outputs/expected’ with the actual generated transaction files found in ‘testcases/outputs/actual’.

AUCTIONBAY

3.0. PHASE 2 TESTING LOG TABLE

Test Name (Pass/Fail)	Test Description	Nature of Failure	Error in Code	Actions to Resolution
login01	User attempts to enter an operation other than 'login' before logging in.	NA	NA	NA
login02	User attempts to login after already logging into an active session.	NA	NA	NA
login03	Any non-admin user (FS, BS, SS) attempts to enter a privileged operation.	Incorrect error is displayed in actual test output.	Conditional for checking the user validation is not handled correctly.	Fix the IF-ELSE statements and put them in the correct order to ensure it results in the expected test output.
login04	Invalid user login; username does not exist in user-accounts file.	Extra newline character resulting in the expected output being different from the actual test output.	There is an additional newline character added to the error message.	Remove the newline character from the error and test the program to see if it matches with the expected test output.
logout01	User attempts to logout before logging in.	Test fails as we misunderstood this testcase.	There is no error in code. When a user enters 'logout' before logging in, the program will simply	Remove this test case from the list of test cases for Phase 3.

			terminate as logout is out program exit operation as well.	
logout02	User successfully logs out after logging in.	NA	NA	NA
create01	Any non-admin user (FS, BS, SS) attempts to enter ‘create’.	Incorrect error is displayed in actual test output.	Conditional for checking the user validation is not handled correctly.	Fix the IF-ELSE statements and put them in the correct order to ensure it results in the expected test output.
create02	Admin attempts to create a user with a username with more than 15 characters in length.	The error that is displayed is correct, however, it shows up at the wrong time.	Conditional for checking the username length is working after it asks the users the other field.	Fix the IF-ELSE to ensure that the error is handled right after the user enters an invalid username.
create03	Admin attempts to create a user with a user type other than AA, FS, BS, and SS.	NA	NA	NA
create04 (07)	Admin attempts to create a user that already exists in the user-accounts file.	NA	NA	NA
create05 (08)	Admin successfully creates a new user.	NA	NA	NA

delete01	Any non-admin user (FS, BS, SS) attempts to delete a user.	Incorrect error is displayed in actual test output.	Conditional for checking the user validation is not handled correctly.	Fix the IF-ELSE statements and put them in the correct order to ensure it results in the expected test output.
delete02	Admin attempts to delete itself.	NA	NA	NA
delete03	Admin attempts to delete a user that does not exist in the user-accounts file.	NA	NA	NA
delete04	Admin deletes the user that matches the username provided.	For this testcase, the program kept on crashing and it was unable to read 'login' as an input.	There does not seem to be any visible error at this time in the code or the expected test output as it runs fine without the script.	Complete all the cases for delete and complete the code. Check in script to ensure it is doing everything correctly. Change expected output accordingly before Phase 3 script run.
advertise01	A buy-standard (BS) user attempts to put an item for auction.	NA	NA	NA
advertise02	User successfully puts an item up for auction.	Item name is not displayed when item is put up for auction successfully.	There is no string input for item name in the printf statement.	Add the itemname record to the printf to print out the item

				name as a string to match the expected output.
advertise03	User attempts to add a minimum bid type other than a double.	NA	NA	NA
advertise04	User attempts to put an item for auction that is listed with a minimum bid amount greater than $999.99.	NA	NA	NA
advertise05	User attempts to put an item for auction with an item name of more than 25 characters in length.	NA	NA	NA
advertise06	User attempts to put up an item for auction with an expiration period of more than 100 days.	NA	NA	NA
advertise07	An admin (AA) or full-standard (FS) user attempts to place a bid on a item right after putting up for an auction. After adding an item no other transaction	NA	NA	NA

	should be accepted on a new ticket for sale until the next session.			
advertise08	User attempts to advertise an item with the same name as an item they already have listed.	NA	NA	NA
bid01	A sell-standard (SS) user attempts to place a bid.	NA	NA	NA
bid02	User attempts to place a bid on an item that does not exist in the available-items file.	NA	NA	NA
bid03	User attempts to place a bid on an item where the seller's username entered does not exist in the user-accounts file.	NA	NA	NA
bid04	User attempts to enter a bid amount that is not an appropriate type.	Incorrect error is displayed in actual test output.	There is no conditional handling the validation of the bid amount, so the program reads the next possible error.	Fix the IF-ELSE in the code and handle the error to check whether the bid amount entered is a number.
bid05	User's new bid placed is successfully	The new bid does not update in the	There is no variable assignment in	Replace the new bid amount over

	added to the item's record of the ‘Current Highest Bid’ amount.	available_items.txt file.	the code for the current highest bid amount.	the current highest bid variable which is read by the available_items .txt file so it saves to the overall item record struct.
bid06	User’s new bid placed is not greater than the current highest bid amount if the user type is admin (AA).	Incorrect error is displayed in actual test output.	There is no conditional handling the validation of the bid amount, so the program reads the next possible error.	Implement a conditional to check whether the bid amount is greater than current highest bid.
bid07	User’s new bid placed is not at least 5% greater than the current highest bid amount if the user type is buy-standard (BS).	Program ignores the 5% bid amount constraint and continues normally.	There is no conditional to validate whether the bid amount entered by FS or BS is atleast 5%.	Implement a conditional to handle the 5% bid constraint and display an error to match the expected test output.
bid08	User’s new bid placed is not at least 5% greater than the current highest bid amount if the user type is full-standard (FS).	Program ignores the 5% bid amount constraint and continues normally.	There is no conditional to validate whether the bid amount entered by FS or BS is atleast 5%.	Implement a conditional to handle the 5% bid constraint and display an error to match the expected test output.
addcredit01	Admin attempts to add credit to a username that does not exist in the user-accounts file.	NA	NA	NA

addcredit02	Any user attempts to add with a credit type other than a double.	Incorrect error is displayed in actual test output.	There is no conditional handling the validation of the credit amount, so the program reads the next possible error.	Fix the IF-ELSE in the code and handle the error to check whether the credit amount entered is a number.
addcredit03	If the user type is full-standard (FS) only credit amount is entered.	Program asks user for a username and credit instead of just asking for credit.	There is no conditional checking the user type to ensure that it is not AA.	Implement a IF-ELSE to check whether the user type is not AA, if so, do not ask user for username.
addcredit04	If the user type is buy-standard (BS) only a credit amount is entered.	Program asks user for a username and credit instead of just asking for credit.	There is no conditional checking the user type to ensure that it is not AA.	Implement a IF-ELSE to check whether the user type is not AA, if so, do not ask user for username.
addcredit05	If the user type is sell-standard (SS) only a credit amount is entered.	Program asks user for a username and credit instead of just asking for credit.	There is no conditional checking the user type to ensure that it is not AA.	Implement a IF-ELSE to check whether the user type is not AA, if so, do not ask user for username.
addcredit06	Any credit amount entered is no more than $1000.00 in any given session.	Program allows user to enter a credit amount over 999.99.	There is no conditional checking the credit amount range.	Implement an IF-ELSE to check if credit amount entered is between 1-999.
listall01	User attempts to view all items available for	NA	NA	NA

	auction via ‘listall’.			
listall02	Item deleted prior to a user entering the ‘listall’ operation does not show up as available for auction.	Item deleted is still being displayed when a user enters ‘listall’ after a item deletion (ie. auction done, or item bought).	There is no case so far to check when an item gets deleted from the available_items.txt file.	For future phases, be prepared to decrement auction days by 1 after every 24 hours and include a function for accepting highest bid. Note: This change will not be made for Phase 3.
listall03	Item advertised prior to a user entering the ‘listall’ operation shows up as available for auction.	NA	NA	NA
refund01	Any non-admin user (FS, BS, SS) attempts to enter ‘refund’.	NA	NA	NA
refund02	Admin attempts to enter a buyer’s username that does not exist in the user-accounts file.	NA	NA	NA
refund03	Admin attempts to enter a seller’s username that does not exist in the user-accounts file.	NA	NA	NA

refund04	Admin attempts to refund with a credit type other than a double.	NA	NA	NA
refund05	Admin attempts to refund a credit amount greater than the seller's credit balance.	NA	NA	NA
refund06	Admin attempts to refund a credit amount to the buyer's account which results in the buyer's credit balance surpassing 999,999.00.	NA	NA	NA

4.0. PHASE 3 CHANGES

*All of the following test cases **PASS** in Phase 3 testing script.*

4.1. FINAL LIST OF TEST CASES

Login:

1. User enters username that does not exist in current_user_accounts file.
2. AA login failed due to incorrect password.
3. AA logs in successfully.
4. FS login failed due to incorrect password.
5. FS logs in successfully.
6. BS login failed due to incorrect password.
7. BS logs in successfully.
8. SS login failed due to incorrect password.
9. SS logs in successfully.

Logout:

10. AA logs out successfully.
11. FS logs out successfully.
12. BS logs out successfully.
13. SS logs out successfully.

Create:

14. FS enters create.
15. BS enters create.
16. SS enters create.
17. AA enters create before logging in.
18. AA attempts to enter username more than 15 characters in length.
19. AA attempts to enter a username that already exists in the current_user_accounts file.
20. AA enters a username successfully, but enters an invalid user type.
21. AA enters a password more than 15 characters in length.
22. AA successfully creates a new user.

Delete:

23. FS enters delete.

24. BS enters delete.
25. SS enters delete.
26. AA enters delete before logging in.
27. AA attempts to enter its own username (ie. delete itself).
28. AA attempts to enter a username that does not exist in the current_user_accounts file.
29. AA successfully deletes an existing user.

Advertise:

30. BS enters advertise.
31. AA enters advertise before logging in.
32. AA attempts to enter item name more than 25 characters in length.
33. AA attempts to enter item name that already exists in available_items file.
34. AA attempts to set an invalid minimum bid price (not a number).
35. AA attempts to set a minimum bid price of more than 999.99.
36. AA attempts to set an invalid auction duration (not a number).
37. AA attempts to set an auction duration of more than 100 days.
38. AA successfully puts up an item for auction.
39. FS enters advertise before logging in.
40. FS attempts to enter item name more than 25 characters in length.
41. FS attempts to enter item name that already exists in available_items file.
42. FS attempts to set an invalid minimum bid price (not a number).
43. FS attempts to set a minimum bid price of more than 999.99.
44. FS attempts to set an invalid auction duration (not a number).
45. FS attempts to set an auction duration of more than 100 days.
46. FS successfully puts up an item for auction.
47. SS enters advertise before logging in.
48. SS attempts to enter item name more than 25 characters in length.
49. SS attempts to enter item name that already exists in available_items file.
50. SS attempts to set an invalid minimum bid price (not a number).
51. SS attempts to set a minimum bid price of more than 999.99.
52. SS attempts to set an invalid auction duration (not a number).
53. SS attempts to set an auction duration of more than 100 days.
54. SS successfully puts up an item for auction.

Bid:

55. SS enters bid after logging in.
56. AA enters bid before logging in.

57. AA attempts to enter an item name that does not exist in available_items file.
58. AA attempts to enter a seller's name that does not exist in current_user_accounts file.
59. AA attempts to bid on their own item.
60. AA attempts to enter an invalid new bid amount (not a number).
61. AA attempts to enter a new bid amount of less than the current bid amount.
62. AA successfully makes a new bid on an item available for auction.
63. FS enters bid before logging in.
64. FS attempts to enter item name that does not exist in available_items file.
65. FS attempts to enter a seller's name that does not exist in current_user_accounts file.
66. FS attempts to bid on their own item.
67. FS attempts to enter an invalid new bid amount (not a number).
68. FS attempts to enter a new bid amount of 5% less than the current bid amount.
69. FS successfully makes a new bid on an item available for auction.
70. BS enters bid before logging in.
71. BS attempts to enter item name that does not exist in available_items file.
72. BS attempts to enter a seller's name that does not exist in current_user_accounts file.
73. BS attempts to enter an invalid new bid amount (not a number).
74. BS attempts to enter a new bid amount of 5% less than the current bid amount.
75. BS successfully makes a new bid on an item available for auction.

Refund:

76. FS enters refund.
77. BS enters refund.
78. SS enters refund.
79. AA enters refund before logging in.
80. AA attempts to enter an invalid refund amount (not a number).
81. AA attempts to enter a buyer's name that does not exist in current_user_accounts file.
82. AA attempts to enter a seller's name that does not exist in current_user_accounts file.
83. AA attempts to enter a refund amount more than the seller’s credit balance.
84. AA successfully refunds an amount from the seller to the buyer.

Addcredit

85. AA enters addcredit before logging in.

86. AA attempts to enter an invalid credit amount (not a number).
87. AA attempts to enter a credit amount more than 1000.00.
88. AA attempts to enter a username that does not exist in current_user_accounts file.
89. AA successfully adds credit to the entered username.
90. FS enters addcredit before logging in.
91. FS attempts to enter an invalid credit amount (not a number).
92. FS attempts to enter a credit amount more than 1000.00.
93. FS successfully adds credit to itself.
94. BS enters addcredit before logging in.
95. BS attempts to enter an invalid credit amount (not a number).
96. BS attempts to enter a credit amount more than 1000.00.
97. BS successfully adds credit to itself.
98. SS enters addcredit before logging in.
99. SS attempts to enter an invalid credit amount (not a number).
100. SS attempts to enter a credit amount more than 1000.00.
101. SS successfully adds credit to itself.

List All Items:

102. AA enters listallitems before logging in.
103. AA enters listallitems.
104. AA enters listallitems after new item is put up for auction.
105. FS enters listallitems before logging in.
106. FS enters listallitems.
107. FS enters listallitems after new item is put up for auction.
108. BS enters listallitems before logging in.
109. BS enters listallitems.
110. SS enters listallitems before logging in.
111. SS enters listallitems.
112. SS enters listallitems after new item is put up for auction.

List All Users:

113. FS enters listallusers.
114. BS enters listallusers.
115. SS enters listallusers.
116. AA enters listallusers before logging in.
117. AA enters listallusers.

Resetpassword:

118. AA enters resetpassword before logging in.
119. AA attempts to enter a new password more than 15 characters in length.
120. AA successfully resets / creates a new password.
121. FS enters resetpassword before logging in.
122. FS attempts to enter a new password more than 15 characters in length.
123. FS successfully resets / creates a new password.
124. BS enters resetpassword before logging in.
125. BS attempts to enter a new password more than 15 characters in length.
126. BS successfully resets / creates new password.
127. SS enters resetpassword before logging in.
128. SS attempts to enter a new password more than 15 characters in length.
129. SS successfully resets / creates a new password.

5.0. INSTRUCTIONS TO RUN

5.1. PROGRAM

1) Open a new terminal window.
2) Go to folder: `cd AuctionBay/PHASE-3/src`
3) Run the makefile: `make`
4) Run: `./auction_system ./iofiles/current_users_accounts.txt ./iofiles/available_items.txt ./iofiles/daily_transaction.txt`

5.2. SCRIPT

1) Open a new terminal window.
2) Go to folder: `cd AuctionBay/PHASE-3/src/scripts`
3) Start Bash terminal: `bash`
4) Run: `./test.sh`