**Auction System - Test Failure Documentation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name** | **What was being tested** | **Error Message/Failure** | **Cause** | **Fix** |
| account\_creation1 | Testing to see if transaction writes the information given to the text files | Creates the file but it doesn’t write into the output file | transactionCode variable was not being set and was therefore bypassing the log transaction function. | Updated program to assign transactionCode with the correct value when an account is created. |
| delete3 | Test to see if you can delete another admin using an admin account | Error message says you can’t delete another admin but still deletes the user in the text file | Validation check for deleting admins was incomplete. | Completed admin code validation to prevent admin accounts from being deleted. |
| advertise1 | Test to make sure the advertisement transactions works with ideal input | Menu is printed twice when advertise transaction is successful | Program was receiving newline character as new input and did not recognize it | Added line of code to ignore the newline character after an item is put up for auction. |
| add\_credit1 | Test to make sure the add credit function works with ideal input | Math error: Adds the credit amount twice | Program code to add credit was repeated twice. | Removed the duplicate line of code that was adding the credit twice. |
| add\_credit1 | Test to make sure the add credit function works with ideal input and passes maximum session limit | Comparison error: a credit amount lower than the maximum still resolves to “exceeding” the session limit | Program code validation incorrectly compared a greater than value rather than a smaller than value | Replaced the “>” comparison with a “<” comparison in the add credit validation |
| add\_credit1 | Test to make sure the add credit function works with ideal input and passes maximum credit limit | Comparison error: a credit amount which would resolve to lower than the maximum incorrectly resolved to “exceeding” the maximum limit | Program code validation incorrectly compared a greater than value rather than a smaller than value | Replaced the “>” comparison with a “<” comparison in the add credit validation |
| add\_credit2 | Checking to see if you can’t add more than the current credit constraint ($1000) | You’re able to exceed the credit/session limit (which is $1000) | No credit session limit validation in code. | Added validation for $1000 session limit |
| add\_credit3 | Test if you can exceed the maximum credit limit of 999,999 | You’re able to enter a credit that exceed the session limit AND exceed the maximum limit of 999,999 | No maximum limit credit validation in code. | Added validation for $999,999 account credit limit. |
| add\_credit4 | Test to see what happens if you input a non-numeric value for adding credit | terminate called after throwing an instance of 'std::invalid\_argument'  what(): stof  Aborted (core dumped) | If a string was entered into the amount field it would cause an invalid conversion from float to string. | Added validation check to ensure that string to float conversion only happens on valid entries. |
| add\_credit5 | Test to see if you can add credit to another user | Program doesn’t ask which user they want to add credit | Program code did not contain a prompt for a username for admin accounts | Added a username prompt for users that are logged in with an admin account |
| refund1 | Test to see if the refund transaction works with ideal input | Refund function doesn’t refund amount to the user. | String to float comparison was comparing incorrect variable causing the statement to never return true | Corrected the if statement to compare the correct variable. |
| refund7 | Test to see if you can refund a value that exceeds the constraint of 999,9999 | Admin can enter a refund amount that can exceed the constraint of 999,999 | Program was missing validation for constraint. | Added validation check to ensure refund amount is less than the constraint. |
| bid7 | Test to see what happens when you enter a non-number value in the bid amount | Program will loop infinitely when entering a non-numeric value in the bid amount section of the code. | Program was missing validation for invalid duration input (string inputted). | Added validation check to ensure duration was entered as an integer. |