**Test Cases**

|  |  |  |
| --- | --- | --- |
| Transaction | Test Name | Intention |
| login | login1 | Test that login transaction is accepted as an admin. |
|  | login2 | Test that no other transaction is accepted before login |
|  | login3 | Test that we can log in as a full-standard user |
|  | login4 | Test that we can log in as a buy-standard user |
|  | login5 | Test that we can log in as an sell-standard user |
|  | login6 | Test that you cannot do another login transaction when user is already logged in |
|  | login7 | Test that login transaction is not accepted by a username that does not exist |
| logout | logout1 | Test that logout transaction is accepted when logged in |
|  | logout2 | Test that logout transaction is not accepted when user is not logged in |
|  | logout3 | Test that no other transaction except login is accepted after logout |
| create | create1 | Test that create transaction is accepted for admin to add a admin |
|  | create2 | Test that create transaction is accepted for admin to add a full-standard user |
|  | create 3 | Test that create transaction is accepted for admin to add a buy-standard user. |
|  | create4 | Test that create transaction is accepted for admin to add a sell-standard user |
|  | create5 | Test that this transaction cannot be made by a full-standard user |
|  | create6 | Test that this transaction cannot be made by a buy-standard user |
|  | create7 | Test that this transaction cannot be made by a sell-standard user |
|  | create8 | Test that this transaction does not accept input greater than 15 characters |
|  | create9 | Test that this transaction does not accept a duplicate username |
| delete | delete1 | Test that this transaction is accepted by admin on an existing user |
|  | delete2 | Test that this transaction is not accepted by a buy-standard user. |
|  | delete3 | Test that this transaction is not accepted by a sell-standard user. |
|  | delete4 | Test that the transaction is not accepted when an admin provides a non-existent username. |
|  | delete5 | Test that the transaction is not accepted for the name of the current user |
|  | delete6 | Test that the information is saved for the daily transaction file |
|  | delete7 | Test that no other transactions should be accepted for the deleted user |
| sell | sell1 | Test that the transaction is accepted for valid arguments types for event title, price, and number of tickets for sale. |
|  | sell2 | Test that the transaction is accepted when done as an admin user. |
|  | sell3 | Test that the transaction is accepted when done as a full-standard user. |
|  | sell4 | Test that the transaction is accepted when done as an sell-standard user. |
|  | sell5 | Test that the transaction is not accepted by a buy-standard user. |
|  | sell6 | Test that the transaction is accepted when the price of a ticket is 999.99. |
|  | sell7 | Test that the transaction is accepted when the price of the ticket is less than 999.99 |
|  | sell8 | Test that the transaction is not accepted when the price of the ticket is greater than 999.99 |
|  | sell9 | Test that the transaction is accepted when the word count for a title of a ticket is 25 characters long. |
|  | sell10 | Test that the transaction is accepted when the word count for a title of a ticket is less than 25 characters long. |
|  | sell11 | Test that the transaction is not accepted when the word count for a title of a ticket is more than 25 characters long. |
|  | sell12 | Test that the transaction is accepted when the number of tickets for sale is set to 100. |
|  | sell13 | Test that the transaction is accepted when the number of tickets for sale is less than 100. |
|  | sell14 | Test that the transaction is not accepted when the number of tickets for sale is greater than 100. |
|  | sell15 | Test that no transactions can be done until the next session. |
| buy | buy1 | Test that the transaction is accepted when logged in as an admin user or a buy-standard user. |
|  | buy2 | Test that the transaction is not accepted when logged in as a sell-standard user. |
|  | buy3 | Test that the transaction is accepted for valid data types of event title, number of tickets and the seller’s username. |
|  | buy4 | Test that the transaction is not accepted when one of the data types for event title, number of tickets, or the seller’s username is incorrect. |
|  | buy5 | Test that the transaction is accepted when the event title is a current title under the provided seller’s username. |
|  | buy6 | Test that the transaction is not accepted when the event title is does not exist under the provided seller’s username. |
|  | buy7 | Test that the transaction is accepted when buying 4 tickets under an event title that has 4 or more in stock. |
|  | buy8 | Test that the transaction is accepted when buying less than 4 tickets under an event that has 4 tickets in stock. |
|  | buy9 | Test that the transaction is not accepted when the user inputs 0 for the number of tickets to buy. |
|  | buy10 | Test that the transaction is not accepted when requesting to buy more tickets than how much is in stock for an event title. |
| refund | refund1 | Test that transaction is accepted when the username, seller and amount of credit to transfer are valid data types. |
|  | refund2 | Test that the transaction is not accepted when one of the input parameters is an invalid data type. |
|  | refund3 | Test that the transaction is accepted when the buyer’s username and the seller’s username are current users |
|  | refund4 | Test that the transaction is not accepted when the buyer’s username is not current and the seller’s username is current. |
|  | refund5 | Test that the transaction is not accepted when the buyer’s username is current but the seller’s username is not current. |
|  | refund6 | Test that the transaction is not accepted when neither the buyer and seller’s username are current users. |
|  | refund7 | Test that the transaction is accepted when the specified amount from the seller’s credit balance is transferred to the buyer’s credit balance |
|  | refund8 | Test that the transaction is not accepted when the seller’s credit is less than the specified amount of credit to transfer. |
|  | refund9 | Test that the transaction is not accepted when the amount of credit to transfer will result in the buyer’s credit to exceed the credit limit. |
|  | refund10 | Test that the transaction is not accepted if the user is a any account with no admin privileges. |
| addcredit | addcredit1 | Test that the transaction is accepted when an admin is adding credit to an account that exists |
|  | addcredit2 | Test that the transaction is not accepted when an admin is adding credit to a user that does not exist. |
|  | addcredit3 | Test that the transaction is accepted when an admin user asks for a maximum of $1000 in 1 session |
|  | addcredit4 | Test that the transaction is accepted when a full-standard user asks for a maximum of $1000 in 1 session |
|  | addcredit5 | Test that the transaction is accepted when a buy-standard user asks for a maximum of $1000 in 1 session |
|  | addcredit6 | Test that the transaction is accepted when a sell-standard user asks for a maximum of $1000 in 1 session |
|  | addcredit7 | Test that the transaction is not accepted when an admin user asks for an amount greater than $1000 in 1 session. |
|  | addcredit8 | Test that the transaction is not accepted when a full-standard user asks for an amount greater than $1000 in 1 session. |
|  | addcredit9 | Test that the transaction is not accepted when a buy-standard user asks for an amount greater than $1000 in 1 session. |
|  | addcredit10 | Test that the transaction is not accepted when an sell-standard user asks for an amount greater than $1000 for any given user in 1 session. |
|  | addcredit11 | Test that the transaction is not accepted when an admin user asks for an amount that will result in an account surpassing the maximum credit limit of 999,999 |
|  | addcredit12 | Test that the transaction is not accepted when a full-standard user asks for an amount that will result in their account surpassing the maximum credit limit of 999,999 |
|  | addcredit13 | Test that the transaction is not accepted when a buy-standard user asks for an amount that will result in their account surpassing the maximum credit limit of 999,999 |
|  | addcredit14 | Test that the transaction is not accepted when a sell-standard user asks for an amount that will result in their account surpassing the maximum credit limit of 999,999 |

**Test Plan**

**Test Organization**

Tests for each transaction are named with their respective transaction name and their corresponding numbers that are ordered consecutively. Ex: buy1, buy2, buy3 etc.

And are put in their respective folders. Ex. buy, login, logout, etc.

Input files

**.inp files** contains the input into the console for each test case

**.cuaf files** contains a list of current user accounts at the start of the test cases’ execution.

Expected output files

**.bto files** contains the expected output dialog of the console during the session of the test case

**.etf files** contains the expected daily transactions when a test case is completed

**.eatf files** contains the expected available tickets after buy/sell transactions are executed.

**.cuafo files** contains a list of current user accounts after testing the create transactions

Actual output files

**.ato files** contains the actual output dialog of the console after the session of the test case is completed

**.atf files** contains the actual daily transactions

**.aatf files** contains the actual available tickets

**.acuafo files** contains the actual user account file

**Test Run Plan**

For the front end, the test program will read **.inp** andthe **.cuaf** file and run it against the system. It will then read the expected output files, with the actual outputs then compare the two results. Tests will be run in order by the transaction type (in the order of login, logout, create, delete, sell, buy, refund, addcredit) and the test number in ascending order.

**Requirements Problems**

The requirements doesn’t specify how users know what type of transactions (create, buy, sell, etc) are available when the system starts or after they logged in. We assume it should display a message such as, “Choose a transaction to perform (logout, create, ….):” The display message will only list the transactions that are available to the user type. For example, some user types can only buy/sell tickets while the admin can do all the transaction types.

Similar to above, it doesn’t specify how users know the details of the event and tickets that can be bought, sold, or refunded. We assume this information should be printed out.