**Test Plan**

## CSCI 3060 Winter 2024 Phase I

Public Github Repository: <https://github.com/Linderwolf/SQA>

## **Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Name** | **Description** | **Test Step** | **Expected Result** |
| **Login** | | | |
| noSessionInvalid | When there is no front-end session, Input should accept only the ‘login’ transaction | Input any text other than ‘login’  Garbage Input:  Hello1234  Invalid Transaction code:  create  Valid Username:  admin | Error message. Return to prompting for the valid transaction code: “login” |
| noSessionValid | Entering the ‘login’ transaction code should read from the CurrentUserAccountsFile and prompt the user to input their username | Input text that reads, “login” | CurrentUserAccounts is read from.  Front-End session is started.  User is prompted to enter a username. |
| loginInvalidUsername | Any provided text that is not precisely listed in the User Accounts File should not grant access to the system. | Input any text that is not a stored Username  \*Invalid Transaction:  login  \*Not in the system:  Adminzzzzz  \*Multiple otherwise valid inputs in one line:  Admin User | A warning message is displayed, and the user is prompted to login again. |
| loginAdminValid | The system should be able to compare a username input to the ledger of Current User Accounts, and if they’re not currently logged in, start a session logging them in. | Input a Valid Username:  “Admin” | If the user is not currently logged in, they log in successfully and are provided with a list of valid transaction codes for their user type.  Transaction list should include:  logout  create  delete  sell  buy  refund  addcredit |
| loginAlreadyLoggedIn | The system should not allow a user to Login while they’re already logged in. | Input the invalid transaction code, ‘login’, while logged in. | Reprompt the user to enter a valid transaction code. |
| loginFullStandardValid |  | Input a Valid Username:  “FullStandard” | If the user is not currently logged in, they log in successfully and are provided with a list of valid transaction codes for their user type.  Transaction list should include:  logout  sell  buy  addcredit |
| loginBuyStandardValid |  | Input a Valid Username:  “BuyStandard” | If the user is not currently logged in, they log in successfully and are provided with a list of valid transaction codes for their user type.  Transaction list should include:  logout  buy  addcredit |
| loginSellStandardValid |  | Input a Valid Username:  “SellStandard” | If the user is not currently logged in, they log in successfully and are provided with a list of valid transaction codes for their user type.  Transaction list should include:  logout  sell  addcredit |
| Reading of Daily Transaction File | Since a user cannot log in if they’re already logged in,  Reading the Transaction File ensures that no previous login transaction was processed. (According to the Client) | Input the ‘login’ transaction code while not logged in, then log in to a user account listed in the Current User Accounts file.  Reading of the Daily Transaction File should require no input from the user. | The user is able to log in and a front-end session is started |
| User is now Logged in. System now accepts new transactions. | | | |
| Invalid Transaction for User’s Permission | Input should only accept “sell”, “buy”, or “addcredit” unless the account type is privileged: “create”, “delete”, “refund” | Input invalid transaction code | Error message and prompting for valid transaction code |
| Attempt to create new front end session while currently in one | Ensure no previous login transaction processed | Input “login” transaction code while logged in | Error message and prompting user to logout. |
|  | Incorrect Transaction Code | Input text that is not a valid transaction code | A warning message should be displayed.  The front-end should continue reading for input and not proceed to any other transaction. |
| **Logout** | “logout” transaction code | Inputs text that reads “logout” | Front End session ends. User is informed the session ended successfully |
| logoutBeforeLogInInvalid | Attempt to logout before starting a frontend session.  The user cannot log out while not logged in. | Input text that reads “logout” while not in a Front End session | A warning message should be displayed. The user should be prompted to login first. |
| logoutLoggedInValid | Attempt to logout with varying account types after logging into those accounts. | Input text that reads, “logout” after logging in as a Fullstandard account, BuyStandard account, SellStandard account, and then Admin account. | The system should write to the Daily Transaction File, logging each transaction performed by that user, according to the specifications.  This should be repeated for each pair of login/logout. |
| Write to the Daily Transaction File | After a successful logout, the system should write to the Daily Transaction File | No input should be required other than a “logout” transaction. | The System should write to the Daily Transaction File according to the listed specifications. |
| **Create** | “create” transaction code | “create” |  |
| 1\_success | User uses the “create” command while logged in as admin | Admin account inputs “create” transaction code | User is prompted to define “username” and “user type”. Info is saved to the “daily transaction file” |
| 2\_fullInvalid | User attempts to use “create” command while logged in as full-standard | Non-admin account input “create” transaction code | Error message informing user that “create” is a privileged command and to switch to an admin account before using. |
| 3\_buyInvalid | User attempts to use “create” command while logged in as buy-standard | Non-admin account input “create” transaction code | Error message informing user that “create” is a privileged command and to switch to an admin account before using. |
| 4\_sellInvalid | User attempts to use “create” command while logged in as sell-standard | Non-admin account input “create” transaction code | Error message informing user that “create” is a privileged command and to switch to an admin account before using. |
| 5\_usernameTaken | User cannot use already used username | Input username is found in Current User Accounts file | If the username is already being used, inform the user with a message and prompt to input a different username. |
| 6\_usernameLength | User cannot enter username with over 15 characters. | Input is more than 15 characters | If the username is longer than 15 characters, inform the user with a message and prompt to input a shorter username. |
| 7\_showPrivilegeBuy | Information about the new user should be displayed after creating a new buy-standard user. | The user finishes inputting new user information. | A message should show the new user’s privileges. |
| 8\_showPrivilegeSell | Information about the new user should be displayed after creating a new sell-standard user. | The user finishes inputting new user information. | A message should show the new user’s privileges. |
| 9\_showPrivilegeFull | Information about the new user should be displayed after creating a new full-standard user. | The user finishes inputting new user information. | A message should show the new user’s privileges. |
| **Delete** | “delete” transaction code | Input text that reads “delete” |  |
| 1\_success | Admin cancels any games for sale on a user’s account | Input “delete” then when requested a username, input a username that is an existing user and not the current user | A success message is displayed and the daily transaction file is updated |
| 2\_notAdmin | This is a privileged transaction so the user must be an admin to execute it | Input “delete” | An error message is displayed informing the user this is a privileged transaction |
| 3\_nonexistentUser | The username must be for an existing user for the command to be executed | Input “delete” then when requested a username, input a username that is not an existing user | An error message is displayed informing the admin that the username is not an existing user |
| 4\_currentUser | The username can’t be the current user for the command to be executed | Input “delete” then when requested a username, input the username of the current user | An error message is displayed informing the admin that the username is the current user |
| **Sell** | “sell” transaction code | Input text that reads “sell” | Prompts user to the “game name” and “game price”, then save the information to the “daily transaction file” |
| 1\_success | User uses “sell” command while logged in as admin | Input text that reads “sell” | User receives prompts for “sell” command |
| 2\_fullValid | User uses “sell” command while logged in full-standard | Input text that reads “sell” | User receives prompts for “sell” command |
| 3\_sellValid | User uses “sell” command while logged in as sell-standard | Input text that reads “sell” | User receives prompts for “sell” command |
| 4\_buyInvalid | User attempts to use “sell” while logged in as “buy-standard” | Input text that reads “sell” | An error message is displayed saying the user cannot use that command. |
| 5\_nameTaken | Name is not taken by another game | User inputs “game name” which is already used by another game | An error message is displayed saying the name has already been taken by another listing |
| 6\_nameLength | Name is not longer than 25 characters | User inputs “game name” longer than 25 characters | An error message is displayed informing the user of the max name length |
| 7\_priceIsNumber | Price should not have letters | User inputs words when prompted for price | An error message is displayed saying the price has to be a number. |
| 8\_priceDecimal | Price cannot have more than 2 decimal places | User inputs price that has an extra decimal | Error message informs user that prices cannot have more than 2 decimal places |
| 9\_priceUpperCheck | Price is supposed to be within $0 and $999.99 | User inputs price larger than $999.99. | An error message says the price has to be less than $999.99 |
| 10\_priceLowerCheck | Price is supposed to be within $0 and $999.99 | User inputs price less than $0 | An error message says the price has to be a positive number |
| 11\_dollarSign | Dollar sign input is ignored | User inputs price with ‘dollar’ symbol | System accepts price |
| 12\_buySameSessionInvalid | Game cannot be bought within the same session it was created in | After using “sell”, user attempts “buy” command on the new game. | Error message is displayed telling user to login and try again. |
| 13\_buyNewSessionValid | User makes new session to buy the game added by “sell” | User logs out and logs back in to buy the new game | Game can be added to user library after “buy” test cases |
| **Buy** | “buy” transaction code | Input text that reads “buy” | Prompts user to input “game name” and the “seller’s username”.  Credit equal to the game’s price should be added to the seller’s account. Credit equal to the game’s price should be deducted from the user’s account.  Log to daily transaction file |
| 2\_noSessionInvalid | “buy” transaction code | Input “buy” without starting a session (logging in) | Inform user that is not a valid command and to log in first |
| 3\_noPrivilegeInvalid | “buy” transaction code | Input “buy” while logged in a Sell Standard account type | Error message informing user that they don’t have permission to use that command |
| 4\_gameNameValid | Ensure the game name entered by user is in the Available Games File | Input a valid game name | Prompts user to input the seller’s name |
| 5\_gameNameInvalid | User inputs a game name not in the Available Games File | Invalid game name input (not found in Available Games File) | Error message informs that is not a valid game name |
| 6\_sellerNameValid | Ensure the seller’s name entered is associated with the previously entered Game Name by reading Available Games File | Correctly input the associated seller’s name | Output that the game from the seller is found |
| 7\_sellerNameInvalid | Read Available Games File and find the user specified game does not have such a seller | Invalid seller’s name input | Error message informs user that the game does not have that seller |
| 8\_noSellerInvalid | User specified seller does not exist | Invalid seller’s name input | Error message informs user that there is no such seller |
| 9\_userSellerInvalid | Ensure user is not buying their own game | User input their own username as the seller | Error message informing user they cannot buy their own game |
| 10\_notHaveGameValid | Ensure user does not have the specified game in their  collection | User does not already have the specified game | Can buy specified game |
| 11\_haveGameInvalid | User has the specified game in their collection | The game is in the user’s collection | Error message informs user the specified game is already in their collection |
| 12\_creditSufficient | User has sufficient credit to buy the selected game | User’s credit is more than or equal to the game price. | Purchase is successful. Write to daily transaction file |
| 13\_creditInsufficient | User does not have sufficient credit to buy the selected game | User’s credit is less than the game price | Error message informs user that they do not have sufficient credits |
| **Refund** | “refund” transaction code | Input text that reads “refund” | Prompt for buyer’s username and seller’s username, game name, then the amount of credit to transfer.  Should transfer the specified amount of credit from the seller’s credit balance to the buyer’s credit balance.  Log to daily transaction file. |
| 2\_noSessionInvalid | “refund” transaction code | Input “refund” without starting a session (logging in) | Inform user that is not a valid command and to log in first |
| 3\_FSPrivilegeInvalid | User attempts to use “refund” command while logged in as full standard account | Full standard account input “refund” transaction code | Error message informing user that “refund” is a privileged command and to switch to an admin account before using. |
| 4\_BSPrivilegeInvalid | User attempts to use “refund” command while logged in as buy standard account | Buy standard account input “refund” transaction code | Error message informing user that “refund” is a privileged command and to switch to an admin account before using. |
| 5\_SSPrivilegeInvalid | User attempts to use “refund” command while logged in as sell standard account | Sell standard account input “refund” transaction code | Error message informing user that “refund” is a privileged command and to switch to an admin account before using. |
| 6\_privilegeValid | User uses the “refund” command while logged in as admin | Admin account inputs “refund” transaction code | Admin is prompted to enter the buyer’s username, seller’s username, name of game to refund, and amount of credit to transfer. |
| 7\_buyerNotCurrentInvalid | Buyer username not in Current User Accounts File | Reading of Current User Accounts File | Error message showing buyer is not a current user |
| 8\_sellerNotCurrentInvalid | Seller username not in Current User Accounts File | Reading of Current User Accounts File | Error message showing seller is not a current user |
| 9\_usersNotCurrentValid | Buyer and Seller usernames in Current User Accounts File | Reading of Current User Accounts File | Prompt for amount to refund |
| 10\_creditValueInvalid | Amount of credit to transfer is not equivalent to what the buyer spent | Read Daily Transaction File and find that credit amount specified is not equivalent to the game price in transaction | Error message that the amount specified to refund is not what the buyer spent on the game |
| 11\_creditValueValid | Amount of credit to transfer is equivalent to what the buyer spent | Read Daily Transaction File and find that credit amount specified is equivalent to the game price in transaction | Should transfer the specified amount of credit from the seller’s credit balance to the buyer’s credit balance.  Log to daily transaction file. |
| 12\_buyerNoGame | Buyer doesn’t have the specified game in their game collection | Read Game Collection File and find the user does not have the file in their collection | Error message indicating the buyer doesn’t have the specified game in their game collection and thus the transaction couldn’t be processed |
| 13\_sellerInvalid | Buyer did not buy game from the seller account specified | Read Daily Transaction Files and find that the buy transaction had different seller than the one specified by the admin | Error message indicating the seller specified is not the one part of the buy transaction |
| 14\_sellerValid | Buyer has the specified game in their game collection and the specified seller is correct | Read Daily Transaction Files and Game Collection File | Should transfer the specified amount of credit from the seller’s credit balance to the buyer’s credit balance.  Log to daily transaction file. |
| 15\_maxCredits | Buyer account’’s available credit is at the max limit (999,999) | Read Current User Accounts File and find the available credit for the buyer account is already at max (999,999) | Warning that the buyer’s credits are at the max and no more credits can be added. Ask for confirmation to continue. |
| 16\_nearMax | Buyer account’’s available credit will go beyond  the max limit (999,999) after refund and thus the surplus credits will be lost | Read Current User Accounts File and find the available credit for the buyer account will surpass the max after adding the refund amount | Warning that the buyer’s credits will reach the max and that the remaining credits will be lost. Ask for confirmation to continue. |
| 17\_fundsValid | Seller has enough credits to fully refund the buyer | Admin inputs buyer’s username, seller’s username, and amount of credit to transfer. Current User Accounts File is read. | Transfer the specified amount of credit from the seller’s credit balance to the buyer’s credit balance.  Log to daily transaction file |
| 18\_fundsInvalid | Seller does not have the amount of credits required to fully refund the buyer | Admin inputs buyer’s username, seller’s username, and amount of credit to transfer. Current User Accounts File is read. | Error message indicates that the seller does not have enough credits to fully refund the buyer. |
| 19\_priceDecimal | Price cannot have more than 2 decimal places | User inputs price that has an extra decimal | Error message informs user that prices cannot have more than 2 decimal places |
| 20\_dollarSign | Dollar sign input is ignored | User inputs price with ‘dollar’ symbol | System accepts price |
| **Add Credit** | “addcredit” transaction code | Input text that reads “addcredit” while logged into | If a front-end session has started, starts an Add Credit transaction,  prompting the user based on their current account type. |
| 1\_addCreditStandard | A standard account starts an Add Credit transaction. | Input text that reads “addcredit” as a standard account. | Prompts for the dollar amount of credit to add to the user’s account. |
| 2\_addCreditStandardAmountInvalid | The value of credit input is not valid. | Input text that is not a positive dollar amount less than or equal to $1000.00. | Display an error message and prompt the user to enter a valid amount. |
| 3\_addCreditStandardAmountValid | The value of credit input is valid | Input text that is a positive dollar amount below or equal to $1000.00 | Accept the amount.  Write to Daily Transaction File. |
| 4\_addCreditAdmin | An admin account starts an Add Credit Transaction | Input text that reads “addcredit” as an admin account. | Prompts the admin for a username |
| 5\_addCreditAdminUsernameInvalid | An admin account doing an Add Credit Transaction inputs an invalid username. | Input a username that is not an existing user. | Display an error message and prompt the admin to input the valid username of a user in the system. |
| 6\_addCreditAdminUsernameValid | An admin account doing an Add Credit Transaction inputs a valid username. | Input a username that is an existing user. | State that the amount input will be added to the provided account name.  Prompt the admin to input a dollar amount. |
| 7\_addCreditAdminAmountInvalid | The value of credit input is not valid. | Input text that is not a positive dollar amount less than $1000.00. | Display an error message and prompt the user to enter a valid amount. |
| 8\_addCreditAdminAmountValid | The value of credit input is valid | Input text that is a positive dollar amount below $1000.00 | Accept the amount.  Write to Daily Transaction File. |