# Phase 1 - Front End Requirements - Genesis

Test Cases

- Login
- Logout
- Create
- Delete
- Advertise
- Bid
- Refund
- Addcredit

Test Plan

- Structure
- Running Tests and Storing Results
- Test Log File Format
  - Header Chunk
  - Test Chunk

## Test Cases

### Login

| Test Name | Description | Input | Output (to file) | Output (terminal) |
|---|---|---|---|---|
| Login 1 | If the username exists, | login USER001 | [File: daily transaction] | Welcome USER001! |

|  |  |  |  |  |
|---|---|---|---|---|
| (admin) | and is an admin, logon successful and append admin privileges | logout | 10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | Account: admin (AA) |
| Login 2 (full-standard) | If the username exists, and is full-standard, logon successful | login USER002<br>logout | [File: daily transaction]<br>10 USER002 FS *CCCCCCCCC*<br>00 USER002 FS *CCCCCCCCC* | Welcome USER002!<br>Account: full-standard (FS) |
| Login 3 (buy-standard) | If the username exists, and is buy-standard, logon successful | login USER003<br>logout | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | Welcome USER003!<br>Account: buy-standard (BS) |
| Login 4 (sell-standard) | If the username exists, and is sell-standard, logon successful | login USER004<br>logout | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>00 USER004 SS *CCCCCCCCC* | Welcome USER004!<br>Account: sell-standard (SS) |
| Login 5 (unsuccessful) | If a username does not exist, logon failed. | login USER005 | [NIL] | ERROR: Login failed, username does not exist |
| Login 6 (already logged in) | If a user attempts to login while a session is in progress, login fails. | login USER001<br>login USER002<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Session already in progress. It must be ended before another user can login. |

## Logout

| **Test Name** | **Description** | **Input** | **Output (to file)** | **Output (terminal)** |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Logout 1 (Successful) | User typed logout, transaction files made, successfully logged out | login USER001<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | Login successful! |
| Logout 2 (Access Error) | User typed logout, user does not have write privileges to the transaction file | login USER001<br>logout | [NIL] | ERROR: Logout failed, user does not have write privileges to transaction file. |
| Logout 3 (Forced Exit) | User clicked Close window rather than logout | [Close window] | [NIL] | |
| Logout 4 (Not logged in) | User is not logged in or has already logged out | logout | [NIL] | ERROR: Logout failed, no user is currently logged in. |

## Create

| Test Name | Description | Input | Output (to file) | Output (terminal) |
|---|---|---|---|---|
| Create 1 (Admin: Create User) | User successfully created | login USER001<br>create<br>USER010<br>BS<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>01 USER010 BS *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC*<br>[File: user accounts]<br>USER010 BS *CCCCCCCCC* | User created successfully! |

|  |  |  |  |  |
|---|---|---|---|---|
| Create 2 (Admin: Create Admin) | User successfully created, append administrator privileges onto account | login USER001<br>create<br>USER011<br>AA<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>01 USER011 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC*<br>[File: user accounts]<br>USER011 AA *CCCCCCCCC* | User created successfully! |
| Create 3 (No privileges to create) | User attempted to create an account with valid permissions | login USER002<br>create<br>USER012<br>BS<br>logout | [File: daily transaction]<br>10 USER002 FS *CCCCCCCCC*<br>00 USER002 FS *CCCCCCCCC* | ERROR: Create failed, only admins can create user accounts. |
| Create 4 (Failed to create) | User attempted to create a new user that already exists | login USER001<br>create<br>USER001<br>BS<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Create failed, the user ‘USER001’ already exists |
| Create 5 (Not logged in) | User is not logged in or has already logged out | create | [NIL] | ERROR: Create failed, no user is currently logged in. |

## Delete

| Test Name | Description | Input | Output (to file) | Output (terminal) |
|---|---|---|---|---|
| Delete 1 (Admin: | User successfully deleted | login USER001<br>delete | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC* | User successfully deleted! |

| | | | | |
|---|---|---|---|---|
| Delete User) | | USER002<br>logout | 02 USER002 FS *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC*<br>[from file: user accounts; DELETE:]<br>USER002 FS *CCCCCCCCC* | |
| Delete 2 (Admin: delete itself) | User attempted to delete themself. Don't let them. | login USER001<br>delete USER001<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Delete failed, the user cannot delete oneself. |
| Delete 3 (No privileges to delete) | User attempted to delete an account without valid permissions | login USER002<br>delete USER001<br>logout | [File: daily transaction]<br>10 USER002 FS *CCCCCCCCC*<br>00 USER002 FS *CCCCCCCCC* | ERROR: Delete failed, the user does not have permission to delete other users. |
| Delete 4 (Failed to delete) | Read error, account doesn't exist | login USER001<br>delete USER013<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Delete failed, the user does not exist. |
| Delete 5 (Failed to delete) | Write error, cannot save changes | login USER001<br>delete USER002<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Delete failed, could not delete user. |
| Delete 6 (Not logged in) | User is not logged in or has already logged out | delete | [NIL] | ERROR: Delete failed, no user is currently logged in. |

## Advertise

| Test Name | Description | Input | Output (to file) | Output (terminal) |
|---|---|---|---|---|
| Advertise 1 (Success; non-buy-standard) | User successfully advertises an item. | login USER004<br>advertise<br>Monopoly<br>30<br>14<br>logout | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>03 Monopoly USER004 014 030.00<br>00 USER004 SS *CCCCCCCCC*<br>[File: available items]<br>Monopoly USER004 USER004 014 000.00 | Monopoly added to auction successfully! |
| Advertise 2 (Admin success) | Immediately approve bids | login USER001<br>advertise<br>Game of Life<br>25<br>21<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>03 Game of Life USER001 021 025.00<br>00 USER001 AA *CCCCCCCCC*<br>[File: available items]<br>Game of Life USER001 USER001 021 000.00 | Game of Life added to auction successfully! |
| Advertise 3 (buy-standard) | Non sell-standard tried to create a listing, improper permissions | login USER003<br>advertise<br>logout | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | ERROR: Advertise failed, user does not have permission to advertise. |
| Advertise 4 (Error, overpriced) | Sell price > 999.99, not allowed. Fail to list | login USER004<br>advertise<br>Checkers<br>1000<br>logout | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>00 USER004 SS *CCCCCCCCC* | ERROR: Advertise failed, sell price is over 999.99. |
| Advertise 5 (Error, name too long) | Name > 25 characters, not allowed. Fail to list. | login USER004<br>advertise<br>Supercalifragilisticexpialidocious | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>00 USER004 SS *CCCCCCCCC* | ERROR: Advertise failed, item name is longer than 25 characters. |

| | | | | |
|---|---|---|---|---|
| | | logout | | |
| Advertise 6 (Error, too long of an auction) | Days > 100, not allowed. Fail to list. | login USER004<br>advertise<br>Chess<br>50<br>101<br>logout | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>00 USER004 SS *CCCCCCCCC* | ERROR: Advertise failed, auction is longer than 100 days. |
| Advertise 7 (One item missing entry) | Incomplete advertisement, fail to list. | login USER004<br>advertise<br>Jenga<br>15<br>logout | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>00 USER004 SS *CCCCCCCCC* | ERROR: Advertise failed, one or more entry is missing to complete transaction. |
| Advertise 8 (Item already exists) | User tried to list an item that's already for sale by the same user for the same price, etc | login USER004<br>advertise<br>Monopoly<br>30<br>logout | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>00 USER004 SS *CCCCCCCCC* | ERROR: Advertise failed, item currently exists in the auction. |
| Advertise 9 (Not logged in) | User is not logged in or has already logged out | advertise | [NIL] | ERROR: Advertise failed, no user is currently logged in. |

## Bid

| Test Name | Description | Input | Output (to file) | Output (terminal) |
|---|---|---|---|---|
| Bid 1 | A completely | login USER003 | [File: daily transaction] | 30.00 |

|  |  |  |  |  |
|---|---|---|---|---|
| (Success!) | successful bid! | bid<br>Monopoly<br>14<br>USER001<br>35.00<br>logout | 10 USER003 BS *CCCCCCCCC*<br>04 Monopoly USER001 USER003 035.00<br>00 USER003 BS *CCCCCCCCC*<br>[File: available items]<br>Monopoly USER001 USER003 035.00 | You have successfully bidded on Monopoly for 35.00! |
| Bid 2 (Incomplete submission) | Submission data is missing or incomplete. Unable to set a bid. | login USER003<br>bid<br>Monopoly<br>14<br>40.00 | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | ERROR: Bid failed, one or more entry is missing to complete transaction. |
| Bid 3 (Improper permissions) | A non-buy standard account is trying to bid. Improper permissions. | login USER004<br>bid<br>logout | [File: daily transaction]<br>10 USER004 SS *CCCCCCCCC*<br>00 USER004 SS *CCCCCCCCC* | ERROR: Bid failed, user does not have permission to bid. |
| Bid 4 (Item does not exist) | Fail to find item listing. | login USER003<br>bid<br>Monopoly<br>14<br>USER002<br>logout | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | ERROR: Bid failed, item does not exist. |
| Bid 5 (New bid < current bid) | Bid is too low, cannot accept the new listing. | login USER003<br>bid<br>Monopoly<br>14<br>USER001<br>25.00<br>logout | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | ERROR: Bid failed, bid is lower than the current bid. |

| | | | | |
|---|---|---|---|---|
| Bid 6 (New bid < 1.05x current bid, with no permissions) | Bid is too low, cannot accept the new listing | login USER003<br>bid<br>Monopoly<br>14<br>USER001<br>36.00<br>logout | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | ERROR: Bid failed, bid is too low. |
| Bid 7 (Insufficient Funds) | User tried to bid more than they own, we can’t have that, that’s our money we’re losing now | login USER003<br>bid<br>Monopoly<br>14<br>USER001<br>100.00 | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | ERROR: Bid failed, user has insufficient funds. |
| Bid 8 (Seller will have > max amount of credits) | Seller lists a bid that when on success, results in over the maximum allowed credits. We will delete the extra credits to cap it at the max | login USER003<br>bid<br>Monopoly<br>14<br>USER001<br>999 | [File: daily transaction]<br>10 USER003 BS *CCCCCCCCC*<br>00 USER003 BS *CCCCCCCCC* | 14.00<br><br>You have successfully bidded on Monopoly for 35.00! |
| Bid 9 (Not logged in) | User is not logged in or has already logged out | bid | [NIL] | ERROR: Bid failed, no user is currently logged in. |

## Refund

| Test Name | Description | Input | Output (to file) | Output (terminal) |
|---|---|---|---|---|
| Refund 0 (Success!) | Successfully gave someone their money back! :) | login USER001<br>refund<br>USER003<br>USER001<br>40<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>05 USER003 USER001 000040.00<br>00 USER001 AA *CCCCCCCCC* | Successful refund! |
| Refund 1 (No permissions) | Non admin user trying to refund, reject it. | login USER002<br>refund<br>logout | [File: daily transaction]<br>10 USER002 FS *CCCCCCCCC*<br>00 USER002 FS *CCCCCCCCC* | ERROR: Refund failed, user does not have permission to refund. |
| Refund 2 (Input incomplete, missing item) | Form not fully completed, fail to give refund to a user | login USER001<br>refund<br>USER003<br>40<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Refund failed, one or more entry is missing to complete transaction. |
| Refund 3 (Negative refund money) | Improper input, cannot take away money from an account on refund | login USER001<br>refund<br>USER003<br>USER001<br>40<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Refund failed, user does not have sufficient funds for return transaction. |
| Refund 4 (One user does not exist) | Cannot refund money to user, buyer or seller does not exist. | login USER001<br>refund<br>USER040<br>USER01<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Refund failed, buyer or seller does not exist. |

| | | | | |
|---|---|---|---|---|
| Refund 5 (Not logged in) | User is not logged in or has already logged out | refund | [NIL] | ERROR: Refund failed, no user is currently logged in. |

## Addcredit

| Test Name | Description | Input | Output (to file) | Output (terminal) |
|---|---|---|---|---|
| Add Credit 1 (non admin account success) | User added credit onto their own account | login USER002<br>addcredit<br>100<br>logout | [File: daily transaction]<br>10 USER002 FS *CCCCCCCCC*<br>06 USER002 FS *CCCCCCCCC*<br>00 USER002 FS *CCCCCCCCC* | Credit added successfully! |
| Add Credit 2 (Admin access) | Admin added credits to a user account | login USER001<br>addcredit<br>100<br>USER003<br>logout | [File: daily transaction]<br>10 USER002 FS *CCCCCCCCC*<br>06 USER003 BS *CCCCCCCCC*<br>00 USER002 FS *CCCCCCCCC* | Credit added successfully! |
| Add Credit 3 (No username , admin) | Failed to give a null user credits | login USER001<br>addcredit<br>100<br>USER040<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Add credit failed, user does not exist. |
| Add | Fail to add credits, | login USER001 | [File: daily transaction] | ERROR: Add credit failed, credit is too |

| | | | | |
|---|---|---|---|---|
| Credit 4 (Added > 1000$) | amount is too high | addcredit<br>2000<br>USER003<br>logout | 10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | high. |
| Add Credit 5 (Added > user's max amount) | We tried to add credits, but you're already at the cap, I guess we have to cut off the extra and keep it at maxed | login USER001<br>addcredit<br>400<br>USER003<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Add credit failed, unable to complete transaction. |
| Add Credit 6 (User attempted to add letters) | Reject it, it's just flat out wrong. Sheeesh, go back to the main menu with you all | login USER001<br>addcredit<br>AAA<br>USER003<br>logout | [File: daily transaction]<br>10 USER001 AA *CCCCCCCCC*<br>00 USER001 AA *CCCCCCCCC* | ERROR: Add credit failed, credit value is invalid. |
| Add Credit 7 (Not logged in) | User is not logged in or has already logged out | addcredit | [NIL] | ERROR: No user is currently logged in. |

# Test Plan

## Structure

The project is structured as follows:
documents

```
|___ Front End
|          |____ [Front-end test documents and info]
|___ Back End
           |____ [Back-end test documents and info]
```

```
src
|___ frontend
|          |____ [Front-end C++ code]
|___ backend
           |____ [Back-end Java code]
```

```
tests
|___ frontend
|          |____ [Front-end C++ test scripts and test outputs]
|          |____ logs
|                    |___ [Historical test logs and master output file]
|___ backend
           |____ [Back-end Java tests]
           |____ logs
                     |___ [Historical test logs]
```

In the tests folder, it will have a frontend folder with all the relevant scripts and output files from those scripts.

## Running Tests and Storing Results

For the front end, there will be a PowerShell script (*frontend_master.ps1*) that will test all the above test cases. It will generate a folder to store the test results (/logs) and create a test log file with the format *YYYY_MM_DD_HOUR_MINUTE*_FE.log (see the *Test Log File Format* section for complete format details).
Where:

- YYYY: year of test run (e.g. 2017)
- MM: month of test run (e.g. 01)

- DD: day of test run (e.g. 30)
- HOUR: hour of test run (e.g. 15)
- MINUTE: minute of test run (e.g. 51)

This will allow multiple test output files to be generated so that it can be compared to at a later date. To limit the number of tests run, the *frontend_master.ps1* script can optionally take a comma separated list of arguments of transactions to test (e.g. login,logout). Omitting the argument will cause all tests to be run.

When the script is running, it will check a master output file with expected outputs and compare it to the actual program output (both the terminal and file output). In the master output file, it will have chunks for each transaction (e.g. login, logout, etc.). Each test case will have a unique test ID that corresponds to the sample output in the master output file in a given chunk.

## Test Log File Format

The test log file will be broken into chunks and will consist of a header chunk with test chunks. Each chunk will have a chunk identifier that identifies the type of chunk and the number of lines (not including the identifier line) that follow as part of the chunk. A chunk identifier is a single line and looks like:
[<ID><LEN><NAME>]

With:

<ID>: the identifier for the type of chunk

- 4 bytes
- HEAD or TEST

<LEN>: the number of lines that follow as part of this chunk (not including the chunk identifier line)

- 3 bytes
- Specifically, 3 digits (between 0-9)
- For example, 026

<NAME>: the name of the chunk

- Variable length, can be 0 length (e.g. in the header chunk)

## Header Chunk

This chunk is not optional, and must be the first in the file. Additionally, there can only be one header chunk in a log file. The ID for the header chunk is *HEAD*. The full chunk identifier is: [HEAD005]

The header chunk consists of 5 lines:

1. The test file description
    - Variable-length
2. The test start date/time
    - Format: YYYY-MM-DD HOUR:MINUTE:SECOND
    - Hour is 00 - 23 (2 bytes)
    - Minute is 00 - 59 (2 bytes)
    - Second is 00 - 59 (2 bytes)
    - E.g. 2017-12-20 22:31:02
3. The test end date/time
    - Same format as start date/time
4. The number of other chunks that follow
    - Variable length
    - Only characters allowed are: 0123456789
    - E.g. ‘10’; means there are 10 other chunks in this file
5. The number of tests that failed
    - Variable length
    - Only characters allowed are: 0123456789
    - If no tests failed, the line will be a single character (0) indicating success

## Test Chunk

A test chunk is used to test a certain transaction (e.g. login). All tests relevant to this transaction should be in this chunk. The ID for a test chunk is *TEST*. The full chunk identifier is: [TEST<LEN><NAME>]

The first line is as follows:

- The number of tests that failed
    - Variable length

    - Only characters allowed are: 0123456789
    - If no tests failed, the line will be a single character (0) indicating success

Each line that follows is in the following format:

<TEST ID>:<SUCCESS?>:<TEST NAME>:<MSG>

Where:

- <TEST ID>: 3 character/digit test identifier
- <SUCCESS?>: 1 character (1 for success/0 for failed)
- <TEST NAME>: A variable length string with a short description of the test case
    - Must not contain newline or colon characters
- <MSG>: The message (readable) that indicates what happened
    - This can be anything (e.g. success/failed/input does not match etc.)
    - It must not contain any newline characters