

CSCI 3060U/SOFE 3980U – Winter 2015
Course Project Assignment #1 – Front End Requirements
Due Date: Wednesday, Feb 11, 2015 (11pm)

AN EXTREME PROJECT

Phase 1: Front End Requirements

Group Name: Algorithmics 2.8
Group Members: Mohammad Ali, 100490152
Fuad Tareq, 100504146
Nicholas De Souza
Bradley Hamilton, 100481185

Part 1 - List of Test Cases and their Intention

Transaction	Test Name	Intention
-------------	-----------	-----------

Login	Login1	Test that the user will be able to enter the username and be accepted for login
	Login2	Test that no transaction other than Login should be accepted when not logged in
	Login3	Test that after username is accepted, it should read the available ticket file and start accepting transactions
	Login4	Test that no subsequent Login can be accepted for the currently logged-in user, until a logout
	Login5	Test that the user can Login as a non-admin
	Login6	Test that the user can Login as an admin
Logout	Logout1	Test that after the logout transaction is processed, a daily transaction file for the day is written, listing every transaction made in the session
	Logout2	Test that after User logout, no transaction, other than login, can be accepted until next login
Create	Create1	Test that user be able to enter the username as a text line and should be asked for type of user as admin or full-standard, buy standard, sell-standard

	Create2	Test that new username is different from all current users and is not more than 15 characters
	Create3	Test that the information is saved to the daily transaction file
	Create4	Test that max credit cannot be more than 999,999
	Create5	Test that only users logged in as admin can use the privileged transaction
Delete	Delete1	Test that user be able to enter the username as a text line and the username must be the name of an existing user but not the name of the current users
	Delete2	Test that it saves the information for the daily transaction file
	Delete3	Test that no further transaction should be accepted on a deleted user's available inventory of ticket for sale
	Delete4	Test that the transaction is only accepted when logged in as admin user
Sell	Sell1	Test that user is asked for the event title as a text line and sale price for the tickets in dollar and asked for the number of ticket for sale

	Sell2	Test that maximum price for a ticket sale does not exceed 999.99, the maximum length of an event title is 25 characters, and the maximum number of ticket sale is 100
	Sell3	Test that transaction is only accepted when logged in any type of account except standard-buy
	Sell4	Test that the information is saved to the daily transaction file and no further transaction should be accepted on a new ticket for sale until the next session.
Buy	Buy1	Test that it asks for the event title, number of tickets and the seller's username
	Buy2	Test that event title must be a current title with enough tickets available from the specified seller
	Buy3	Test that at most 4 ticket can be purchased in one buy transaction unless it is in privileged mode where there is no limit on all remaining ticker
	Buy4	Test that it displays the cost per ticket and the total cost to the user and ask for confirmation in the form of yes or no
	Buy5	Test that it should subtract the number of tickets from the seller's inventory

	Buy6	Test that transaction is only accepted when logged in any type of account except standard-sell
	Buy7	Test that the information is saved to the daily transaction file
Refund	Refund1	Test that it asks for the buyer's username, the seller's username and the amount of credit to transfer
	Refund2	Test that buyer and seller are both current users
	Refund3	Test that it transfers the specified amount of credit from the seller's balance to the buyer's credit balance
	Refund4	Test that the information is saved to the daily transaction file
	Refund5	Test that the transaction is only accepted when logged in as admin user
AddCredit	AddCredit1	Test that in the admin mode, it asks for the amount of credit to add and the username of the account to which credit is being added
	AddCredit2	Test that in the admin mode, the username has to be an existing username in the system
	AddCredit3	Test that in a standard account, it should ask for the amount of credit

	AddCredit4	Test that the amount of credit of more than \$1000.00 cannot be added in a given session
	AddCredit5	Test that the information is saved to the daily transaction file
General Requirements	General Requirement 1	Test that the Front End should never crash or stop except as directed by the transactions
	General Requirement 2	Test that Front end does not depend on valid inputs and should handle bad inputs of all kinds (assume input is at least line of text)

Part 2- Printout of the actual test inputs and expected outputs.

Login1 input file Login1.inp

Login

Username

Login1 output file Login1.txt

User has successfully logged in.

Login2 input file Login2.inp

Transaction

Login2 output file Login2.txt

Transaction cannot be accepted as user needs to be logged in.

Login3 input file Login3.inp

Login

Username

Ticket file

Transaction

Login3 output file Login3.txt

Username Is accepted and you are logged in.

Ticket file information

Transaction accepted

Login4 input file Login4.inp

Login

2nd Login

Login4 output file Login4.txt

Login cannot be accepted as there is currently Logged in user.

Login5 input file Login5.inp

Login

A non-admin Username

Login5 output file Login5.txt

Username is accepted and logged in as a non-admin

Login6 input file Login6.inp

Login

Admin – Username

Login6 output file Login6.txt

Username is accepted and logged in as an admin

Logout1 input file Logout1.inp

Logout

Logout1 output file Logout1.txt

User has successfully logged out.

Daily transaction file Information

Logout2 input file Logout2.inp

Logout

Transaction

Logout2 output file Logout2.txt

User needs to be logged in for the Transaction to be accepted

Create1 input file Create1.inp

Create

Create1 output file Create1.txt

Please enter your Username:

Please select Type of user: admin or full-standard

buy standard

sell-standard

Create2 input file Create2.inp

Existed Username

Non-Existed Username but more than 15 characters

Create2 output file Create2.txt

Need a different Username, the Username already exists

Username cannot be more than 15 characters

Create3 input file Create3.inp

Create user

Create3 output file Create3.txt

Daily transaction file holding user information

Create4 input file Create4.inp

Max credit as 1000000

Max credit as 999999

Create4 output file Create4.txt

Max credit cannot be more than 999999

Max credit is accepted

Create5 input file Create5.inp

Create

User type – admin or full standard

Create5 output file Create5.txt

Admin users are allowed full transaction privileges

Delete1 Input file Delete1.inp

Non-Existed Username

Exited Username

Delete1 output file Delete1.txt

Username is a non-existent user

Username is accepted

Delete2 Input file Delete2.inp

Delete Transaction

Delete2 output file Delete2.txt

Daily Transaction file with Information of Deleted Transaction

Delete3 Input file Delete3.inp

Deleted User

Transaction on Deleted user's inventory of ticket for sale

Delete3 output file Delete3.txt

The user's account is deleted and no further transaction can be made.

Delete4 Input file Delete4.inp

Logged in as Non-admin Username

Delete Transaction

Logged in as Admin–Username

Delete Transaction

Delete4 output file Delete4.txt

Transaction denied; must be admin user to access

Transaction accepted

Sell1 Input file Sell1.inp

Sell

Sell1 output file Sell1.txt

The event title:

Sale price in \$:

Number of ticket for Sale:

Sell2 Input file Sell2.inp

Price for ticket sale as 1000

Price for ticket sale as 999.99

Event title as 26 characters

Event title as 25 characters

Maximum number of ticket sale as 101

Maximum number of ticket sale as 100

Sell2 output file Sell2.txt

Price for ticket sale cannot be more than 999.99

Price for ticket sale accepted

Event title cannot be more than 25 characters

Event title is accepted

Maximum number of ticket sale cannot be more than 100

Maximum number of Ticket sale accepted

Sell3 Input file Sell3.inp

Logged in as admin-user

Logged in as sell-standard user

Logged in as buy-standard user

Sell3 output file Sell3.txt

User can make Sell transaction

User can make Sell transaction

User cannot make Sell transaction

Sell4 Input file Sell4.inp

Sell transaction created

Transaction on Sell transaction created

Sell4 output file Sell4.txt

Daily transaction file with information on created Sell transaction

Transaction cannot be made on a new ticket for Sale until next session

Buy1 Input file Buy1.inp

Buy transaction

Buy1 output file Buy1.txt

Event title:

Number of tickets:

Sellers: Username

Buy2 Input file Buy2.inp

Not Current Event title

Not enough Number of Tickets

Current Event title

Enough Number of Tickets

Buy2 output file Buy2.txt

There is not a Current Event

There is not enough number of Tickets

There is a current event

There is number of tickets

Buy3 Input file Buy3.inp

Unprivileged User – number of tickets as 5

Unprivileged User – number of tickets as 4

Privileged User – number of tickets as 6

Buy3 output file Buy3.txt

Need to be a privileged user to buy tickets more than 4

Number of Tickets allowed

Number of Tickets allowed

Buy4 Input file Buy4.inp

Select the Event

Buy4 output file Buy4.txt

Cost/ticket:

Do u confirm the transaction: YES

NO

Buy5 Input file Buy5.inp

Number of tickets left: 8

Buy transaction Completed: 2 tickets sold

Check the number of tickets after transaction

Buy5 output file Buy5.txt

Number of tickets left: 6

Buy6 Input file Buy6.inp

Logged in as admin-user

Logged in as sell-standard user

Logged in as buy-standard user

Buy6 output file Buy6.txt

User can make Buy transaction

User cannot make Buy transaction

User can make Buy transaction

Buy7 Input file Buy7.inp

Buy transaction completed

Buy7 output file Buy7.txt

Daily transaction file with information of completed but Transaction

Refund1 input file Refund1.inp

Refund Transaction

Refund1 output file Refund1.txt

Buyer's Username:

Seller's Username:

Amount of credit to Transfer:

Refund2 input file Refund2.inp

Not Current Buyer

Not Current Seller

Current Buyer

Current Seller

Refund2 output file Refund2.txt

This is not a current buyer

This is not a current seller

This is the current buyer

This is a current Seller

Refund3 input file Refund3.inp

Amount of Credit to transfer: 1000

Refund3 output file Refund3.txt

The amount of credit of 1000 has been transferred

Refund4 input file Refund4.inp

Refund Transaction completed

Refund4 output file Refund4.txt

Daily transaction file with information of completed Refund Transaction

Refund5 input file Refund5.inp

Logged in as admin-user

Refund Transaction completed

Logged in as sell-standard user

Refund Transaction completed

Logged in as buy-standard user

Refund Transaction completed

Refund5 output file Refund5.txt

Refund Transaction accepted

Refund Transaction cannot be accepted

Refund Transaction cannot be accepted

AddCredit1 input file Addcredit1.inp

Logged in as admin

Add Credit

AddCredit1 output file Addcredit1.txt

Amount of Credit:

Username of the account:

AddCredit2 input file Addcredit2.inp

Logged in as admin

Non – Existed Username – the account for the credit being added

Existed Username – the account for the credit being added

AddCredit2 output file Addcredit2.txt

The Username does not exist

The Username exist

AddCredit3 input file Addcredit3.inp

Logged in as a Standard account

AddCredit3 output file Addcredit3.txt

Amount of credit:

AddCredit4 input file Addcredit4.inp

Amount of Credit as 1001

Amount of credit as 1000

AddCredit4 output file Addcredit4.txt

The amount of Credit cannot be more than 1000

The amount of credit is allowed

AddCredit5 input file Addcredit5.inp

Add credit transaction completed

AddCredit5 output file Addcredit5.txt

Daily transaction file with information of completed Add credit transaction

GeneralRequirements1 input file GeneralRquirements1.inp

Running transaction

Completing transaction

GeneralRequirements1 output file GeneralRquirements1.txt

No crash on Running Transaction

No crash on Completing Transaction

GeneralRequirements2 input file GeneralRquirements2.inp

Bad inputs

GeneralRequirements2 output file GeneralRquirements2.txt

Not allowed, please enter valid input

Part 3 - Test Plan

Test Organization

Tests are organized into a single directory organized by file name. Tests for each transaction are named starting with the transaction name and numbered sequentially, for example Login1, Login2, Login3 and so on. Test inputs are named ending in ".inp", for example "login1.inp". Corresponding expected File outputs are named with the same name ending in ".txt", for example "login1.txt". When tests are run, actual test run outputs for the actual output File will be named ending in ".atf" (e.g. "input1.atf") in the same directory.

Test Run Plan

Tests will be run using a tester application script that runs the Front End for each of the test input files in the directory. After each is run, the script will save the output file and compare it to the expected output file, and reporting if there is a difference.

Part 4 -Requirement Problems

The requirements do not properly explain the privileges of Admin, whether the Admin user has any constraints, how much control does he have and what the exact differences are between Sell-Standard and Buy-standard. We assume that the privileges of Admin combines of the privileges of Sell-Standard and Buy-standard and also being able to refund, add credit and delete transaction whereas Sell-Standard can only sell tickets and Buy-Standard can only buy tickets.

