

ASSIGNMENT 3  
TEST REPORT  
CISC327 GROUP 13 (SPICE TESTS)

Note:

Many of the tests did not run initially due to small things such as inconsistent new lines.  
We fixed all the little issues and inconsistencies and then ran our first test.  
The second run of the test program was run after we ensured all of our input files were correct leaving us with only bugs in the code. All bugs were fixed and the test program was run once more with all successes.

To run tests:

You must have java installed on your linux system  
run `java -h` to check if it is installed

QIES can be built, or you can use the provided .class files.

To build QUIES navigate to `./src/` directory and execute:  
`javac QIESBase.java -d ../bin/`

To run QIES navigate to `./build/` directory and execute:

`java -cp ../bin "QIESBase" "vsf.txt"`

Once java is installed, and QIES is built...

you may simply run `./tests/run_tests.sh`  
and the tests will automatically be run using the  
class files found in the `/build/` directory.  
The outputs of the console and the TXN summary  
file from the `./tests/output/` and `./tests/expected/`  
are used for comparison.

On SUCCESS

Under the test name header SUCCESS will be displayed  
This means console and txn summary files are matching

On FAIL

A detailed report of the console output as well as  
differences between the txnsummary and console log  
are displayed.

On COMPLETE

Counters of tests passed and tests failed are  
displayed

| FAILURE TYPE | TEST CATEGORY | TEST NAME                                    | TEST PURPOSE   | ISSUE / SOLUTION  |
|--------------|---------------|--|--|---|
|              |               |  |  |   |
|              |               |  |  |   |
| TEST FAILURE | CANCEL_TICKET | cancel_10_tickets_service_agent              | ensure that agent can cancel 10 tickets in a session for a specific service                      | Expected transaction summary file mistakenly included a date on the cancel like. Removed date.  |
| TEST FAILURE | CANCEL_TICKET | cancel_20_tickets_session_agent              | ensure that agent can't cancel more than 10 tickets in a session for a specific service          | Expected transaction summary file mistakenly included a date on the cancel like. Removed date.  |
| SUCCESS      | CANCEL_TICKET | cancel_large_tickets_planner                 | ensure that planner can cancel unlimited tickets (more than 20)                                  |   |
| CODE FAILURE | CANCEL_TICKET | cancel_more_than_10_service_agent            | ensure that agent can't cancel more than 10 tickets per service when using one transaction       | Added code keep track of and limit how many tickets an agent can sell per service. Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead)    |
| CODE FAILURE | CANCEL_TICKET | cancel_more_than_10_service_multitrans_agent | ensure that agent can't cancel more than 10 tickets per service when using multiple transactions | Added code to keep track of and limit how many tickets an agent can sell per service. Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead) |
| CODE FAILURE | CANCEL_TICKET | cancel_more_than_20_session_agent            | ensure that an agent can't cancel more than 20 tickets total in a session                        | Added code keep track of and limit how many tickets an agent can sell per session. Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead)    |
| CODE FAILURE | CANCEL_TICKET | cancel_negative_tickets_agent                | ensure agent can't cancel a negative number of tickets   | Added code to check that the number entered was larger than 0   |
| CODE FAILURE | CANCEL_TICKET | cancel_negative_tickets_planner              | ensure planner can't cancel a negative number of tickets   | Added code to check that the number entered was larger than 0   |
| CODE FAILURE | CANCEL_TICKET | cancel_number_tickets_string                 | ensure that non numeric characters are not accepted for a number of tickets                      | Added a validation check to make sure all entered characters were numeric   |
| TEST FAILURE | CANCEL_TICKET | cancel_tickets_planner                       | ensure planner can successfully cancel tickets   | Console input mistakenly had multiple service numbers in it. Removed second service number.   |
| CODE FAILURE | CANCEL_TICKET | cancel_too_large_tickets_planner             | ensure planner can't cancel more than 1000 tickets (limit set by transaction summary file)       | Limit canceling tickets to 1000. Reprompt for valid number that is greater than 0 and less than 1001  |
| CODE FAILURE | CANCEL_TICKET | invalid_service_number                       | ensure given service number is valid   | Code did not reprompt for a valid service number. Instead it immediately terminated.  |

|              |                |                                       |   |  |
|--------------|----------------|---------------------------------------|---|--|
| CODE FAILURE | CANCEL_TICKET  | invalid_service_number_string         | ensure non-numeric characters are not accepted as a service number  | Added validation check to make sure all entered characters were numeric  |
| SUCCESS      | CANCEL_TICKET  | valid_service_number                  | ensure valid service numbers are accepted   |  |
| CODE FAILURE | CHANGE_TICKET  | change_agent_ticket_limit             | ensure agent can't change more than 20 tickets in a single session while using a single transaction           | Added code keep track of and limit how many tickets an agent can change per session.   |
| CODE FAILURE | CHANGE_TICKET  | change_agent_ticket_limit_multi_trans | ensure agent can't change more than 20 tickets in a single session while using multiple transactions          | Added code keep track of and limit how many tickets an agent can change per session.   |
| CODE FAILURE | CHANGE_TICKET  | change_invalid_current_service_number | ensure that invalid service number is not accepted as the source service number                               | Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead)  |
| CODE FAILURE | CHANGE_TICKET  | change_invalid_new_service_number     | ensure that invalid service number is not accepted as the destination service number                          | Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead)  |
| CODE FAILURE | CHANGE_TICKET  | change_invalid_ticket_number_zero     | ensure that 0 is not accepted as a valid number of tickets  | Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead)  |
| SUCCESS      | CHANGE_TICKET  | change_large_planner                  | ensure planner can change more than 20 tickets in one session   |  |
| CODE FAILURE | CHANGE_TICKET  | change_too_large_planner              | ensure planner can't change more than 1000 tickets in one session (limit set by the transaction summary file) | Added validation loop to reprompt user for number of tickets to change if it is outside of the bounds of 1 and 1000<br>Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead) |
| TEST FAILURE | CHANGE_TICKET  | change_valid_to_valid_agent           | ensure agent can change a ticket from one valid service to another valid service                              | Test console input had a space inbetween change and ticket. Removed space.   |
| TEST FAILURE | CHANGE_TICKET  | change_valid_to_valid_planner         | ensure planner can change a ticket from one valid service to another valid service                            | Test console input had a space inbetween change and ticket. Removed space.   |
| CODE FAILURE | CREATE_SERVICE | assign_service_date_above_d_range     | ensure date entry above 31 is not accepted  | Added code to check that the entered date value was between 1 and 31 (inclusive)   |
| CODE FAILURE | CREATE_SERVICE | assign_service_date_above_m_range     | ensure month entry above 12 is not accepted   | Added code to check that the entered month value was between 1 and 12 (inclusive)  |
| CODE FAILURE | CREATE_SERVICE | assign_service_date_above_y_range     | ensure year entry above 2999 is not accepted  | Added code to check that the entered year value was between 1980 and 2999 (inclusive)  |
| CODE FAILURE | CREATE_SERVICE | assign_service_date_below_d_range     | ensure date entry below 1 is not accepted   | Added code to check that the entered date value was between 1 and 31 (inclusive)   |
| CODE FAILURE | CREATE_SERVICE | assign_service_date_below_m_range     | ensure month entry below 1 is not accepted  | Added code to check that the entered month value was between 1 and 12 (inclusive)  |

|                                |                |                                    |  |  |
|--------------------------------|----------------|------------------------------------|--|--|
| <b>CODE FAILURE</b>            | CREATE_SERVICE | assign_service_date_below_y_range  | ensure year entry below 1980 is not accepted   | Added code to check that the entered year value was between 1980 and 2999 (inclusive)  |
| <b>TEST FAILURE</b>            | CREATE_SERVICE | assign_service_date_invalid_format | ensure date is entered in a valid format   | Console input lines were in incorrect order. Program expected day month year, and then reprompted for invalid inputs. Input lines expected an immediate reprompt.  |
| <b>TEST FAILURE</b>            | CREATE_SERVICE | assign_service_date_nondigit       | ensure non-numeric characters are not accepted as date entries   | Console input lines were in incorrect order. Program expected day month year, and then reprompted for invalid inputs. Input lines expected an immediate reprompt.  |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_name_begin_space    | ensure a service name can't begin with a space   | Date format of output did not 0 pad months or days. This caused a difference between output and expected txn summary files. Added padding to fix.  |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_name_end_space      | ensure a service name can't end with a space   | Date format of output did not 0 pad months or days. This caused a difference between output and expected txn summary files. Added padding to fix.  |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_name_large          | ensure service name entry is 39 characters or less   | Added code to check that the entered service name value was between 3 and 39 characters (inclusive). Date format of output did not 0 pad months or days. This caused a difference between output and expected txn summary files. Added padding to fix. |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_name_nonalpha       | ensure non-letter characters are not accepted in the service name                                      | Add regular expression check to ensure only alpha characters and apostrophes are accepted. Date format in console input was in the incorrect format.   |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_name_small          | ensure service name is >=3 characters  | Added code to check that the entered service name value was between 3 and 39 characters. Date format in console input was in the incorrect format. (inclusive)   |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_number_0_start      | ensure system disallows service numbers beginning with 0   | Date format in console input was in the incorrect format. Program was not padding single digit days and months. Added padding for txn summary output.  |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_number_duplicate    | ensure system disallows repeated service numbers   | Date format in console input was in the incorrect format. Program was not padding single digit days and months. Added padding for txn summary output.  |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_number_large        | ensure that the system disallows service numbers longer than 5 digits, and then prompts for a re-entry | Date format in console input was in the incorrect format. Program was not padding single digit days and months. Added padding for txn summary output.  |
| <b>CODE &amp; TEST FAILURE</b> | CREATE_SERVICE | assign_service_number_non_digit    | ensure system disallows non-digit service numbers  | Date format in console input was in the incorrect format. Program was not padding single digit days and months. Added padding for txn summary output.  |

|                                    |                |                              |   |  |
|------------------------------------|----------------|------------------------------|---|--|
| <b>CODE &amp;<br/>TEST FAILURE</b> | CREATE_SERVICE | assign_service_number_small  | ensure that the system disallows service numbers smaller than 5 digits, and then prompts for a re-entry | Date format in console input was in the incorrect format. Program was not padding single digit days and months. Added padding for txn summary output.  |
| <b>CODE &amp;<br/>TEST FAILURE</b> | CREATE_SERVICE | create_new_service_valid     | ensure new service, with proper inputs, can be created  | Date format in console input was in the incorrect format. Program was not padding single digit days and months. Added padding for txn summary output.  |
| <b>SUCCESS</b>                     | CREATE_SERVICE | create_service_agent         | ensure agent cannot execute createservice   |  |
| <b>TEST FAILURE</b>                | DELETE_SERVICE | agent_deleteservice          | ensure agent cannot execute deleteservice   | Input code caused code not to terminate. No txn summary file outputted. Ensuring code termination fixed the issue.   |
| <b>CODE FAILURE</b>                | DELETE_SERVICE | cancelticket_deleted_service | ensure that cancelticket cannot be executed on a deleted service  | Entering invalid service number ended command instead of reprompting for another input.  |
| <b>SUCCESS</b>                     | DELETE_SERVICE | deleteservice_agent          | ensure that agent cannot delete a service   |  |
| <b>CODE FAILURE</b>                | DELETE_SERVICE | invalid_service_number       | ensure that invalid service number cannot be cancelled  | Entering invalid service number ended command instead of reprompting for another input.  |
| <b>CODE FAILURE</b>                | DELETE_SERVICE | sellticket_deleted_service   | ensure that sellticket cannot be executed on a deleted service  | Transaction summary file included the deleted service number despite program reprompting for new service number. (Old number was kept in separate variable and used instead)   |
| <b>TEST FAILURE</b>                | DELETE_SERVICE | valid_service_number         | ensure that provided service number is valid  | Expected txn summary file had a date hardcoded into the DEL line. Removed and replaced with single 0.  |
| <b>CODE FAILURE</b>                | LOGIN          | cancelticket_before_login    | ensure that cancelticket is not accepted before login   | Added check in cancelticket code to make sure user is logged in first.   |
| <b>CODE FAILURE</b>                | LOGIN          | createservice_before_login   | ensure that createservice is not accepted before login  | Added check in createservice code to make sure user is logged in first.  |
| <b>CODE FAILURE</b>                | LOGIN          | deleteservice_before_login   | ensure that deleteservice is not accepted before login  | Added check in deleteservice code to make sure user is logged in first.  |
| <b>TEST FAILURE</b>                | LOGIN          | invalid_session_type_empty   | ensure that empty input is not accepted as login type   | Input code caused code not to terminate. No txn summary file outputted. Ensuring code termination fixed the issue.   |
| <b>CODE &amp;<br/>TEST FAILURE</b> | LOGIN          | invalid_session_type_numeric | ensure that numeric input is not accepted as login type   | Input code caused code not to terminate. No txn summary file outputted. Fixed input code. Entering an incorrect user type did not reprompt for a correct one instead it just terminated the command. Added a reprompt. |
| <b>CODE &amp;<br/>TEST FAILURE</b> | LOGIN          | invalid_session_type_string  | ensure that strings other than agent and planner are not accepted as login                              | Input code caused code not to terminate. No txn summary file outputted. Fixed input code. Entering an incorrect user type did not reprompt for a correct one instead it just terminated the command. Added a reprompt. |

|              |             |                                    |   |  |
|--------------|-------------|------------------------------------|---|--|
| TEST FAILURE | LOGIN       | login_session_agent                | ensure that agent login is accepted   | Input code caused code not to terminate. No txn summary file outputted. Ensuring code termination fixed the issue.   |
| TEST FAILURE | LOGIN       | login_session_planner              | ensure that planner login is accepted   | Input code caused code not to terminate. No txn summary file outputted. Ensuring code termination fixed the issue.   |
| SUCCESS      | LOGIN       | login_twice_agent                  | ensure that no subsequent login should be accepted after a login until after a logout |  |
| SUCCESS      | LOGIN       | login_twice_planner                | ensure that no subsequent login should be accepted after a login until after a logout |  |
| TEST FAILURE | LOGIN       | logout_before_login                | ensure that logout is not accepted until logged in                                    | Input code caused code not to terminate. No txn summary file outputted. Ensuring code termination fixed the issue.   |
| CODE FAILURE | LOGIN       | sellticket_before_login            | ensure that transactions not accepted unless logged in                                | Added check in sellticket code to make sure user is logged in first.   |
| SUCCESS      | LOGOUT      | login_accepted_after_logout        | ensure that login is accepted after logout  |  |
| SUCCESS      | LOGOUT      | logout_not_logged_in               | ensure that logout is not accepted unless already logged in                           |  |
| SUCCESS      | LOGOUT      | only_login_accepted_after_logout   | ensure no command other than login is accepted post logout                            |  |
| TEST FAILURE | SELL_TICKET | sell_existing_service_one          | ensure one ticket for a valid service can be sold                                     | Expected txn summary file included dates in the SEL line. These dates were removed and problem was resolved.   |
| TEST FAILURE | SELL_TICKET | sell_existing_service_two          | ensure two tickets can be sold for a valid service                                    | Expected txn summary file included dates in the SEL line. These dates were removed and problem was resolved.   |
| CODE FAILURE | SELL_TICKET | sell_invalid_service_number        | ensure that invalid service number is not accepted for ticket sales                   | Program did not reprompt for service number after entering an incorrect one. Added loop to do so.  |
| CODE FAILURE | SELL_TICKET | sell_invalid_service_number_string | ensure that string is not accepted as a valid service number for ticket sales         | Program did not reprompt for service number after entering an incorrect one. Added loop to do so.  |
| SUCCESS      | SELL_TICKET | sell_ticket_number_large           | ensure 1000 tickets can be sold in a single transaction                               |  |
| SUCCESS      | SELL_TICKET | sell_ticket_number_string          | ensure a string is not accepted as a number of tickets sold                           |  |
| CODE FAILURE | SELL_TICKET | sell_ticket_number_too_large       | ensure that no more than 1000 tickets can be sold                                     | Added loop to reprompt for number of tickets sold if it is less than 1 or greater than 1000. Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead) |

|              |             |                         |                                    |  |
|--------------|-------------|-------------------------|------------------------------------|--|
| CODE FAILURE | SELL_TICKET | sell_ticket_number_zero | ensure zero tickets cannot be sold | Added loop to reprompt for number of tickets sold if it is less than 1 or greater than 1000. Transaction summary file included the invalid ticket amount despite program reprompting for new service number. (Old number was kept in separate variable and used instead) |
|--------------|-------------|-------------------------|------------------------------------|--|