

Unit Test Report for Point Of Sale System

- Test Cases Specification
- Test Summary Report

Project Team

Team 1

Latest update on:

2017-11-06

Team Information

201211337 김재현

201112052 방민석

201312259 백만일

201211383 조영래

Table of Contents

1	Introduction.....	3
1.1	Objectives	3
1.2	References.....	3
2	Unit test case specification.....	3
2.1	Test case specification identifier	3
2.2	Test items.....	6
2.3	Input specifications.....	9
2.4	Output specifications.....	9
3	Environmental needs.....	9
4	Unit test summary report.....	10
4.1	Test summary report identifier.....	10
4.2	Evaluation.....	11

1 Introduction

1.1 Objectives

본 문서는 2017 학년도 2학기 소프트웨어 공학 개론 수업의 Team1이 개발한 POS system(이하 POS) Unit Test의 결과를 위한 문서이다.

1.2 References

2 Unit test case specification

2.1 Test case specification identifier

< Table1 Test Design Identification >

Identifier	Feature (Process DFD)	Valid / Invalid value
POS_BAR_DATA_01	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "001", -1}
POS_BAR_DATA_02	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "010", -1}
POS_BAR_DATA_03	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "011", -1}
POS_BAR_DATA_04	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "100", -1}
POS_BAR_DATA_05	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "101", -1}
POS_BAR_DATA_06	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "110", -1}
POS_BAR_DATA_07	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "111", -1}
POS_BAR_DATA_08	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "000", -1}

POS_BAR_DATA_09	1.3 Determine Barcode	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "r", barcode, -1}
POS_COMMAND_DATA_01	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'b', &server, &list, -1}
POS_COMMAND_DATA_02	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'s', &server, &list, 1}
POS_COMMAND_DATA_03	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'r', &server, &list, 2}
POS_COMMAND_DATA_04	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'C', &server, &list, 3}
POS_COMMAND_DATA_05	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'i', &server, &list, -1}
POS_COMMAND_DATA_06	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'c', &server, &list, -1}
POS_COMMAND_DATA_07	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'e', &server, &list, 0}
POS_COMMAND_DATA_08	1.4 Determine Command	determine_command(COMMAND, SERVER, LIST, state) : {'x', &server, &list, -1}
POS_SERVER_DATA_01	2.2 Convert Server Data	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num, 0}
POS_SERVER_DATA_02	2.2 Convert Server Data	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num, 1}
POS_SERVER_DATA_03	2.2 Convert Server Data	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num, 2}
POS_SERVER_DATA_04	2.2 Convert Server Data	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list,

		receipt_num,2}
POS_Controller_01	3.1.1 Controller	selected_state {-1}
POS_Controller_02	3.1.1 Controller	selected_state {0}
POS_Controller_03	3.1.1 Controller	selected_state {1}
POS_Controller_04	3.1.1 Controller	selected_state {2}
POS_Controller_05	3.1.1 Controller	selected_state {3}
POS_Controller_06	3.1.1 Controller	selected_state {4}
POS_Controller_07	3.1.1 Controller	selected_state {int_max}
POS_Display_Main_01	3.1.2 Display Main	display_main(DISPLAY) : {&display}
POS_SALE_DATA_01	3.1.3 Sale	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, list, server, sale, 1, display}
POS_SALE_DATA_02	3.1.3 Sale	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {0, 0, server, sale, 1, display}
POS_SALE_DATA_03	3.1.3 Sale	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, 0, server, sale, 1, display}
POS_SALE_DATA_04	3.1.3 Sale	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, 0, 0, sale, 1, display}
POS_REFUND_DATA_01	3.1.4 Refund	Refund(LIST, SERVER, REFUND, state, DISPLAY) : {&list, &server, ,&refund, 2, display}
POS_CHECK_DATA_01	3.1.5 Check Inventory	Check(SERVER, state, DISPLAY) : {&server, 3, &display}
POS_SETTLEMENT_01	3.1.6 Settlement	settlement(LIST, SERVER, SETTLEMENT, state) : {&list, &server, &settlement, 4}
POS_SETTLEMENT_02	3.1.6 Settlement	settlement(LIST, SERVER, SETTLEMENT, state) : {0, &server, &settlement, 4}
POS_Update_Display_01	3.1.7 Update Display	update_display(SALE,REFUND, SERVER, DISPLAY) : {1, 0, 0, 0}
POS_Update_Display_02	3.1.7 Update Display	update_display(SALE,REFUND, SERVER, DISPLAY) : {0, 1, 0, 0}
POS_Update_Display_03	3.1.7 Update Display	update_display(SALE,REFUND, SERVER, DISPLAY) : {0, 0, 1, 0}
POS_Update_Server_01	3.1.8 Update Server	update_server(SERVER, LIST) : {&server, &list}

POS_Update_Server_02	3.1.8 Update Server	update_server(SERVER, LIST) : {&server, 0}
POS_PRINT_DATA_01	3.1.9 Print Data	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {&sale, 0, 0, 0}
POS_PRINT_DATA_02	3.1.9 Print Data	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, &refund, 0, 0}
POS_PRINT_DATA_03	3.1.9 Print Data	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, &server, 0}
POS_PRINT_DATA_04	3.1.9 Print Data	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, 0, &settlemet}

2.2 Test items

< Table2 Test Case Identification >

Identifier	Input Specification	Output Specification
POS_BAR_DATA_01	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "001", -1}	Barcode Data(product)
POS_BAR_DATA_02	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "010", -1}	Barcode Data(product)
POS_BAR_DATA_03	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "011", -1}	Barcode Data(product)
POS_BAR_DATA_04	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "100", -1}	Barcode Data(product)
POS_BAR_DATA_05	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "101", -1}	Barcode Data(product)
POS_BAR_DATA_06	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "110", -1}	Barcode Data(product)
POS_BAR_DATA_07	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "111", -1}	Barcode Data(product)
POS_BAR_DATA_08	determine_barcode(LIST, SERVER,	Error(Exception Handling)

	COMMAND, barcode, state) : {&list, &server, "b", "000", -1}	
POS_BAR_DATA_09	determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "r", barcode, -1}	Barcode Data(receipt)
POS_COMMAND_DATA_01	determine_command(COMMAND, SERVER, LIST, state) : {'b', &server, &list, -1}	Command = Input Barcode
POS_COMMAND_DATA_02	determine_command(COMMAND, SERVER, LIST, state) : {'s', &server, &list, 1}	Command = Sale
POS_COMMAND_DATA_03	determine_command(COMMAND, SERVER, LIST, state) : {'r', &server, &list, 2}	Command = Refund
POS_COMMAND_DATA_04	determine_command(COMMAND, SERVER, LIST, state) : {'C', &server, &list, 3}	Command = Check
POS_COMMAND_DATA_05	determine_command(COMMAND, SERVER, LIST, state) : {'i', &server, &list, -1}	Command = Input Cash
POS_COMMAND_DATA_06	determine_command(COMMAND, SERVER, LIST, state) : {'c', &server, &list, -1}	Command = product_amt change
POS_COMMAND_DATA_07	determine_command(COMMAND, SERVER, LIST, state) : {'e', &server, &list, 0}	Command = Exit
POS_COMMAND_DATA_08	determine_command(COMMAND, SERVER, LIST, state) : {'x', &server, &list, -1}	Error(Exception Handling)
POS_SERVER_DATA_01	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,0}	Server Data
POS_SERVER_DATA_02	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,1}	Server Data
POS_SERVER_DATA_03	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,2}	Server Data

POS_SERVER_DATA_04	convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,2}	Error(Exception Handling)
POS_Controller_01	selected_state {-1}	Input Barcode
POS_Controller_02	selected_state {0}	Wait
POS_Controller_03	selected_state {1}	Sale
POS_Controller_04	selected_state {2}	Refund
POS_Controller_05	selected_state {3}	Check
POS_Controller_06	selected_state {4}	Settlement
POS_Controller_07	selected_state {int_max}	Error(Exception Handling)
POS_Display_Main_01	display_main(DISPLAY): {&display}	Display Data
POS_SALE_DATA_01	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, list, server, sale, 1, display}	Sale Data
POS_SALE_DATA_02	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {0, list, server, sale, 1, display}	Error(Exception Handling) r_price < total_price
POS_SALE_DATA_03	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, 0, server, sale, 1, display}	Error(Exception Handling) List x
POS_SALE_DATA_04	sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, list, 0, sale, 1, display}	Error(Exception Handling) p_amt > server_stock
POS_REFUND_DATA_01	Refund(LIST, SERVER, REFUND, state, DISPLAY) : {&list, &server, &refund, 2, display}	Refund Data
POS_CHECK_DATA_01	Check(SERVER, state, DISPLAY) : {&server, 3, &display}	Check Data
POS_SETTLEMENT_01	settlement(LIST, SERVER, SETTLEMENT, state) : {list, &server, &settlement, 4}	Settlement Data
POS_SETTLEMENT_02	settlement(LIST, SERVER, SETTLEMENT, state) : {0, &server, &settlement, 4}	Error(Exception Handling) List x
POS_Update_Display_01	update_display(SALE,REFUND, SERVER, DISPLAY) : {1, 0, 0, 0}	Display Command
POS_Update_Display_02	update_display(SALE,REFUND, SERVER, DISPLAY) : {0, 1, 0, 0}	Display Command

POS_Update_Display_03	update_display(SALE,REFUND, SERVER , DISPLAY) : {0, 0, 1, 0}	Display Command
POS_Update_Server_01	update_server(SERVER, LIST) : {&server, &list}	Save
POS_Update_Server_02	update_server(SERVER, LIST) : {&server, 0}	Error(Exception Handling) list x
POS_PRINT_DATA_01	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {&sale, 0, 0, 0}	Printer Command
POS_PRINT_DATA_02	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, &refund, 0, 0}	Printer Command
POS_PRINT_DATA_03	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, &server, 0}	Printer Command
POS_PRINT_DATA_04	print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, 0, &settlemet}	Printer Command

2.3 Input specifications

< Table2 Test Case Identification > 참조

2.4 Output specifications

< Table2 Test Case Identification > 참조

3 Environmental needs

POS system Unit Test를 위한 환경적 요구사항은 다음과 같다.

(1) Hardware & platform

GCC compiler/linker

(2) Unit test framework

CUnit

(3) Integrated Platform Cygwin

Cygwin

4 Unit test summary report

4.1 Test summary report identifier

Test Case Identifier	Result
POS_BAR_DATA_01	Passed
POS_BAR_DATA_02	Passed
POS_BAR_DATA_03	Passed
POS_BAR_DATA_04	Passed
POS_BAR_DATA_05	Passed
POS_BAR_DATA_06	Passed
POS_BAR_DATA_07	Passed
POS_BAR_DATA_08	Passed
POS_BAR_DATA_09	Passed
POS_COMMAND_DATA_01	Passed
POS_COMMAND_DATA_02	Passed
POS_COMMAND_DATA_03	Passed
POS_COMMAND_DATA_04	Passed
POS_COMMAND_DATA_05	Passed
POS_COMMAND_DATA_06	Passed
POS_COMMAND_DATA_07	Passed
POS_COMMAND_DATA_08	Passed
POS_SERVER_DATA_01	Passed
POS_SERVER_DATA_02	Passed
POS_SERVER_DATA_03	Passed
POS_SERVER_DATA_04	Passed
POS_Controller_01	Passed
POS_Controller_02	Passed
POS_Controller_03	Passed
POS_Controller_04	Passed
POS_Controller_05	Passed
POS_Controller_06	Passed
POS_Controller_07	Passed
POS_Display_Main_01	Passed
POS_SALE_DATA_01	Passed

POS_SALE_DATA_02	Passed
POS_SALE_DATA_03	Passed
POS_SALE_DATA_04	Passed
POS_REFUND_DATA_01	Passed
POS_CHECK_DATA_01	Passed
POS_SETTLEMENT_01	Passed
POS_SETTLEMENT_02	Passed
POS_Update_Display_01	Passed
POS_Update_Display_02	Passed
POS_Update_Display_03	Passed
POS_Update_Server_01	Passed
POS_Update_Server_02	Passed
POS_PRINT_DATA_01	Passed
POS_PRINT_DATA_02	Passed
POS_PRINT_DATA_03	Passed
POS_PRINT_DATA_04	Passed

4.2 Evaluation

Total test case : 46개

Passed : 46개

Failed : 0개