

System Test Report for Point Of Sale System

- Test Cases Specification
- Test Summary Report

Project Team

T6

Latest update on:

2017-11-19

Team Information

201311264 김병식

201610379 김나연

201611248 강병성

201610070 김지우

Table of Contents

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

1 Introduction

1.1 Objectives

본 문서는 System Test Plan에서 추출한 Test Case를 통해 System Test의 결과를 기록한 문서이다. 결과의 여부는 PASS와 FAIL로 표기하였다.

1.2 References

2017SE-POS-STP-1.0

2 System test case specification

2.1 Test case specification identifier

POS_STC_Num.Num

2.2 Test items

Identifier	Input Specification	Output Specification
POS_STC_000.000	종료 버튼을 누름	포스기 프로세스와 포스기 시계 프로세스가 죽음 (포스기 시각 기록 YYYYMMDDHHmm)
POS_STC_000.001	전원 버튼을 누름	포스기 프로세스와 포스기 시계 프로세스가 생성되고 포스기 시계는 종료된 시각부터 이어짐 (포스기 시각 : YYYYMMDDHHmm, 서버 파일의 날짜 확인으로 대체한다.)
POS_STC_001.000	포스기 프로그램 실행	홈 화면에 제공되는 메뉴에 판매, 환불, 재고 확인이 출력
POS_STC_001.001	바코드 번호 001 → 010 → 011 → 100 → 101 → 110 → 111 입력	판매 리스트에 7가지 상품의 정보가 추가되고 수량은 1로 출력
POS_STC_001.002	바코드 번호 001 → 010 → 001 → 100 → 100 → 001 → 111 입력	판매 리스트에 과자(3), 아이스크림(1), 물(2), 커피(1) 상품의 정보가 하나의 Tick 발생마다 실시간으로 추가되어 출력 * 괄호는 상품의 개수임
POS_STC_001.003	'+' → 001 → 10 '+' → 010 → 20	판매 리스트에 과자(10), 아이스크림(20), 과일(30), 물(40), 라면(50), 음료수(60), 커피(70)가 출력

	'+' → 011 → 30 '+' → 100 → 40 '+' → 101 → 50 '+' → 110 → 60 '+' → 111 → 70 * 매뉴얼 지침에 따름 * 이전에 판매 기록이 없음을 전제, 즉 상품의 개수는 모두 100개	피(70) 상품의 정보가 하나의 Tick 발생마다 실시간으로 추가되어 출력 * 괄호는 상품의 개수임
POS_STC_001.004	'+' → 101 → 50 '-' → 101 → 30 '+' → 111 → 70 '-' → 111 → 20 * 매뉴얼 지침에 따름 * 이전에 판매 기록이 없음을 전제, 즉 상품의 개수는 모두 100개	판매 리스트에 라면(20), 커피(50) 상품의 정보가 하나의 Tick 발생마다 실시간으로 추가되어 출력 * 괄호는 상품 개수임
POS_STC_001.005	판매 완료에 대해 'y' 입력 *POS_STC_001.001의 결과를 이용한다.	POS_STC_001.001의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력
POS_STC_001.006	판매 완료에 대해 'y' 입력 *POS_STC_001.002의 결과를 이용한다.	POS_STC_001.002의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력
POS_STC_001.007	판매 완료에 대해 'y' 입력 *POS_STC_001.003의 결과를 이용한다.	POS_STC_001.003의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력
POS_STC_001.008	판매 완료에 대해 'y' 입력 *POS_STC_001.004의 결과를 이용한다.	POS_STC_001.004의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력
POS_STC_001.009	POS_STC_001.005 → POS_STC_001.006 수행 후 재고 확인 요청 * 3분을 넘기지 않음을 전제	과자(96), 아이스크림(98), 과일(99), 물(97), 라면(99), 음료수(99), 커피(98)의 재고 상태로 양식에 맞춰 출력 * 괄호는 상품 개수임

POS_STC_001.010	POS_STC_001.007에 대한 환불 요청 환불 완료에 대해 'y' 입력	POS_STC_001.007에서 생성된 판매 영수증 단위로 환불 처리하고 영수증 번호, 날짜, 환불 상품 리스트, 환불 금액을 양식에 맞춰 출력
POS_STC_001.011	POS_STC_001.006에 대한 환불 요청 환불 완료에 대해 'y' 입력	POS_STC_001.006에서 생성된 판매 영수증 단위로 환불 처리하고 영수증 번호, 날짜, 환불 상품 리스트, 환불 금액을 양식에 맞춰 출력
POS_STC_002.000	바코드 번호 001 → 010 → 011 → 100 → 101 → 110 → 111 입력	판매 리스트에 7가지 상품의 정보가 추가되고 수량은 1로 출력
POS_STC_002.001	바코드 번호 001 → 010 → 001 → 100 → 100 → 001 → 111 입력	판매 리스트에 과자(3), 아이스크림(1), 물(2), 커피(1) 상품의 정보가 하나의 Tick 발생마다 실시간으로 추가되어 출력 * 괄호는 상품의 개수임
POS_STC_002.002	'+' → 001 → 10 '+' → 010 → 20 '+' → 011 → 30 '+' → 100 → 40 '+' → 101 → 50 '+' → 110 → 60 '+' → 111 → 70 * 매뉴얼 지침에 따름 * 이전에 판매 기록이 없음을 전제, 즉 상품의 개수는 모두 100개	판매 리스트에 과자(10), 아이스크림(20), 과일(30), 물(40), 라면(50), 음료수(60), 커피(70) 상품의 정보가 하나의 Tick 발생마다 실시간으로 추가되어 출력 * 괄호는 상품의 개수임
POS_STC_002.003	'+' → 101 → 50 '-' → 101 → 30 '+' → 111 → 70 '-' → 111 → 20 * 매뉴얼 지침에 따름 * 이전에 판매 기록이 없음을 전제, 즉 상품의 개수는 모두 100개	판매 리스트에 라면(20), 커피(50) 상품의 정보가 하나의 Tick 발생마다 실시간으로 추가되어 출력 * 괄호는 상품 개수임
POS_STC_002.004	판매 완료에 대해 'y' 입력 *POS_STC_002.000의 결과를 이용한다.	POS_STC_002.000의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력

POS_STC_002.005	판매 완료에 대해 'y' 입력 *POS_STC_002.001의 결과를 이용한다.	POS_STC_002.001의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력
POS_STC_002.006	판매 완료에 대해 'y' 입력 *POS_STC_002.002의 결과를 이용한다.	POS_STC_002.002의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력
POS_STC_002.007	판매 완료에 대해 'y' 입력 *POS_STC_002.003의 결과를 이용한다.	POS_STC_002.003의 상품 판매 리스트와 영수증 번호, 날짜, 총액, 지불 금액, 거스름 돈 정보를 양식에 맞춰 출력
POS_STC_002.008	POS_STC_002.001에 대한 환불 요청 환불 완료에 대해 'y' 입력	POS_STC_002.001에서 생성된 판매 영수증 단위로 환불 처리하고 영수증 번호, 날짜, 환불 상품 리스트, 환불 금액을 양식에 맞춰 출력
POS_STC_003.000	바코드 번호 001 → 010 → 011 → 100 → 101 → 110 → 111 입력	캐시 화면으로 상품 판매 리스트에 정의된 7가지 상품이 추가되는 것으로 확인
POS_STC_003.001	유효하지 않은 바코드 입력 00293, 2038, 44930	에러 메시지 발생
POS_STC_003.002	'+' → 101 → 101 '+' → 111 → 2002 * 매뉴얼 지침에 따름 * 이전에 판매 기록이 없음을 전제, 즉 상품의 개수는 모두 100개	에러 메시지 발생
POS_STC_003.003	'+' → 101 → 50 '-' → 101 → 60 '+' → 111 → 70 '-' → 111 → 75 * 매뉴얼 지침에 따름 * 이전에 판매 기록이 없음을 전제, 즉 상품의 개수는 모두 100개	에러 메시지 발생
POS_STC_003.004	POS_STC_002.000에 대한 상품 판매 완료 요청	거스름 돈으로 2000원을 판매 영수증에 기록하는지 캐시 화면/고객 화면으로 확

	지불 금액 12000원 입력	인
POS_STC_003.005	POS_STC_002.003에 대한 상품 판매 완료 요청 지불 금액 116000원 입력	거스름 돈으로 0원을 판매 영수증에 기록하는지 캐시 화면/고객 화면으로 확인
POS_STC_003.006	'+' → 001 → 52 '+' → 101 → 68 바코드 번호 101 입력에 대한 상품 판매 완료 요청 * 매뉴얼 지침에 따름 * 이전에 판매 기록이 없음을 전제, 즉 상품의 개수는 모두 100개	서버 재고 파일에 과자(48), 아이스크림(100), 과일(100), 물(100), 라면(31), 음료수(100), 커피(100)의 정보가 기록 * 괄호는 상품 개수임
POS_STC_004.000	POS_STC_003.004에 대한 환불 요청	정상적인 환불 수행
POS_STC_004.001	POS_STC_003.004에 대한 환불 요청 *POS_STC_004.001 테스트 후에 수행하는 케이스임	에러 메시지 발생
POS_STC_004.002	영수증 바코드에 비정상적인 값, 203834747 입력	에러 메시지 발생
POS_STC_004.003	POS_STC_004.001에 대한 환불 처리 요청을 전제	판매한 7가지 상품 모두를(판매 영수증 단위로 환불함을 확인하기 위함) 환불 상품으로 출력하는지 확인
POS_STC_004.004	POS_STC_001.003에 대한 환불 처리 요청 환불 완료에 대해 'y'를 입력	과자(10), 아이스크림(20), 과일(30), 물(40), 라면(50), 음료수(60), 커피(70) 상품 모두를 환불 상품으로 출력하는지 확인
POS_STC_004.005	POS_STC_004.003의 환불 완료 상황을 전제	서버 재고 파일에 7가지 상품의 수량이 1증가했는지 확인
POS_STC_004.006	POS_STC_004.004의 환불 완료 상황을 전제	서버 재고 파일에 환불한 상품인 과자 수량 → 과자 수량 + 10 아이스크림 수량 → 아이스크림 수량 + 20

		<p>과일 수량 → 과일 수량 + 30</p> <p>물 수량 → 물 수량 + 40</p> <p>라면 수량 → 라면 수량 + 50</p> <p>음료수 수량 → 음료수 수량 + 60</p> <p>커피 수량 → 커피 수량 + 70</p> <p>으로 수량이 업데이트되었는지 확인</p>
POS_STC_004.007	POS_STC_002.005에 대한 상품 환불을 요청 환불 완료에 대해 'n'를 입력	재고 상태에 변함이 없음을 서버 재고 파일로 확인
POS_STC_004.008	POS_STC_002.005에 대한 상품 환불을 요청 환불 완료에 대해 'y'를 입력	POS_STC_002.005에서 판매한 상품에 대한 환불로 인한 재고 상태 업데이트가 반영된 서버 재고 파일 확인
POS_STC_005.000	POS_STC_001.009에 대한 재고 확인 요청을 전제	과자(96), 아이스크림(98), 과일(99), 물(97), 라면(99), 음료수(99), 커피(98)의 재고 상태로 양식에 맞춰 출력되고 서버 재고 상태와 동일한지 확인 * 괄호는 상품 개수임
POS_STC_005.001	포스기 시계로 다음 날 짜가 되었을 때 재고 확인 요청	모든 상품의 수량이 100개로 양식에 맞춰 출력하고 서버 재고 상태와 동일한지 확인
POS_STC_006.000	포스기 시계로 다음 날 짜가 되었을 때 홈 화면에 포스기를 위치	아래 프린터기 테스트 결과로 확인함
POS_STC_007.000	POS_STC_001.005, POS_STC_001.006, POS_STC_001.008에 대한 판매 상황을 전제 (* 단, POS_STC_001.005 이전에 판매 기록이 없으며 3개의 판매 동작 사이에 환불 동작을 요	3개의 상품 판매 파일인, sale_YYYYMMDDhhmm.txt가 실행파일이 위치한 디렉토리 내부에 생성됨
		3개의 판매 기록이 서버 파일인, YYYYMMDD_sale_mangement.txt에 기록됨

	청하지 않음을 명시함. 또한, 3분 내로 3개의 상품 판매는 완료됨) 각각 판매 완료에 대해 'y'를 입력	
POS_STC_007.001	POS_STC_007.000에 판매 완료한 영수증 단위로 환불을 요청 각각 환불 완료에 대해 'y'를 입력	3개의 환불 영수증 파일인, refund_YYYYMMDDhhmm.txt가 실행파일이 위치한 디렉토리 내부에 생성됨
POS_STC_007.002	POS_STC_007.001 테스트 이후를 전제 재고 확인 요청 (* 재고 확인 전에 3분을 초과하지 않음을 명시함)	요청 시점에서의 재고 상태(과자(96), 아이스크림(98), 과일(99), 물(97), 라면(79), 음료수(99), 커피(48))로 stock_YYYYMMDD.txt가 실행파일이 위치한 디렉토리 내부에 생성
POS_STC_007.003	포스기 시계로 다음 날짜가 되었을 때 홈 화면에 포스기를 위치 POS_STC_007.001 테스트 이후의 상황을 전제	하루 단위로 누적된 판매 내역을 settle_YYYYMMDD.txt에 출력되고 실행파일이 위치한 디렉토리 내부에 생성 (3개의 판매 기록과 3개의 환불 기록이 출력)
		재고 서버의 상품 파일과 판매 관리 파일은 다음 날짜로 새로 생성
		재고 서버의 상품 파일에 있어 7가지의 상품 수량은 모두 100개로 초기화

2.3 Input specifications

상위 표 참조

2.4 Output specifications

상위 표 참조

3 Environmental needs

-gcc 5.4.0버전

-cygwin64, 9pt

4 System test summary report

4.1 Test summary report identifier

Identifier	Result
POS_STC_000.000	PASS
POS_STC_000.001	FAIL
POS_STC_001.000	PASS
POS_STC_001.001	PASS
POS_STC_001.002	PASS
POS_STC_001.003	PASS
POS_STC_001.004	PASS
POS_STC_001.005	PASS
POS_STC_001.006	PASS
POS_STC_001.007	PASS
POS_STC_001.008	PASS
POS_STC_001.009	PASS
POS_STC_001.010	PASS
POS_STC_001.011	PASS
POS_STC_002.000	PASS
POS_STC_002.001	PASS
POS_STC_002.002	PASS
POS_STC_002.003	PASS
POS_STC_002.004	PASS
POS_STC_002.005	PASS
POS_STC_002.006	PASS
POS_STC_002.007	PASS
POS_STC_002.008	PASS
POS_STC_003.000	PASS
POS_STC_003.001	PASS
POS_STC_003.002	PASS
POS_STC_003.003	PASS
POS_STC_003.004	PASS
POS_STC_003.005	PASS
POS_STC_003.006	PASS

POS_STC_004.000	PASS
POS_STC_004.001	PASS
POS_STC_004.002	PASS
POS_STC_004.003	PASS
POS_STC_004.004	PASS
POS_STC_004.005	PASS
POS_STC_004.006	PASS
POS_STC_004.007	PASS
POS_STC_004.008	PASS
POS_STC_005.000	PASS
POS_STC_005.001	PASS
POS_STC_006.000	FAIL
POS_STC_007.000	PASS
	PASS
POS_STC_007.001	PASS
POS_STC_007.002	PASS
POS_STC_007.003	PASS
	PASS
	PASS

4.2 Evaluation

POS_STC_000.001과 POS_STC_006.000에서 FAIL로 평가되었다. 첫 번째 테스트 케이스는 요구사항 분석 미스에 해당된다. 사용자는 원할 때에 포스기 시스템을 종료하거나

실행시킬 수 있는데 포스기 종료 버튼을 메뉴 버튼으로 제공함으로써 포스기 시스템 상황에 제한을 받는다. 독립적인 프로세스로 생성되어야 한다. 두 번째 테스트 케이스는 포스기 시스템 시계로 처음 전원을 켜고 다음 날짜로 넘어갈 때에 1분도 안되어서 업데이트된다. 그러나 그 이후로는 3분을 맞춰서 업데이트 되는 걸 확인하였으며 추가적으로 문제되는 것은 홈 화면에서 작업 키를 input 함수로 받기 때문에 블로킹된다. 블로킹으로 인해 홈 화면에 있어도 input 함수로 흐름이 넘어가버리면 정산과 새로운 날짜에 대한 서버 파일 생성이 늦어진다. 이 또한 추가적인 시그널 변수를 사용해서 수정할 계획이다.