

T1 Presentation

소프트웨어공학개론

발표자 : 장진서

목차

- 매뉴얼 작성
- STP
- STR (T1, T2)

Manual to T6

팀1 프로그램 매뉴얼

처음화면은 전원이 꺼진 상태입니다. "1"을 입력하고 진행합니다.

인터페이스는 캐쉬화면(상단), 고객화면(중단) 그리고 프로그램 사용자로부터 입력받는 부분이 하단에 위치해있습니다.

전원이 켜지고 나서 초기 상태를 홈 상태라 했을 때, 홈 상태에서 바코드입력(1), 수량변경(2), 판매목록제거(3), 판매(4), 환불(5), 재고확인(6), 시간확인(7), 전원종료(8)를 입력받고 수행합니다.

- 바코드 입력 (string 값)

001 : orange

010 : beer

100 : wine

101 : bread

110 : cheese

111 : pokachip

011 : ramyun

010 : gun

- 수량변경

먼저, 판매목록의 맨 위에서부터 시작하는 n번째 목록을 숫자형태로 입력받습니다. 맨 위 목록은 1입니다.

Testing Plan

- SRS를 보고 요구사항을 만족하는지 확인
- 범위 : 2.4 제약 및 가정사항 → 이것도 요구사항이라 판단
3.2 기능 요구사항
3.3 성능 요구사항

2.4 제약 및 가정 사항

- POST의 캐시 화면(터치 스크린) 입력 및 스캐너 스캔 행위는 키보드 입력으로 대체한다.
- POST의 모든 모니터의 출력은 Cygwin Console(Cygwin Bash Shell)로 대체한다.
- POST의 프린터를 사용한 모든 출력은 텍스트 파일 생성으로 대체한다.
- POST 프로그램의 하루는 현실시간 3분으로 가정한다.(현실시간 1초 → 가상시간 8분)
- POST 프로그램에서 시각 표현이 필요한 부분은 "YYYYMMDDhhmm"으로 "연월일시분"으로 표현한다.
- 201709111843
- POST 프로그램 종료 후 재실행 시 기존에 진행하던 시각이 이어지도록 한다.
POST에서 다루는 상품은 총 7종류(과자, 아이스크림, 과일, 물, 라면, 음료수, 커피)로 한정한다.

SRS 요구사항 분석

- 2.4 제약 및 가정사항 12 (11+ 1)
 - 3.2 기능요구사항 22
 - 판매 10
 - 환불 7
 - 정산 2
 - 재고확인 3
 - 3.3 성능 요구사항 1
- 총 35

- 하루가 지나면 재고 서버의 상품 파일, 판매 관리 파일은 다음 날짜로 새로 생성이 되며
- 상품 파일의 모든 상품은 100개로 초기화 한다.

To be / Not to be Tested

- To be Tested - 27
- Not to be Tested – 8
- Total – 35

To be Tested

- 2.4 제약 및 가정사항 6
 - 3.2 기능요구사항 21
 - 판매 9
 - 환불 7
 - 정산 2
 - 재고확인 3
- 총 27

Not to be Tested

가정사항 : 6 / 12 not to be / (to be + not to be)

- (1) 시각표현이 필요한 부분은 "YYYYMMDDhhmm"으로 "연월일시분"으로 표현한다.
- (2) 캐쉬화면 입력 및 스캐너 스캔행위는 키보드 입력으로 대체한다.
- (3) 모든 모니터의 출력은 Cygwin console 로 대체한다.
- (4) 모든 출력은 텍스트 파일 생성으로 대체한다.
- (5) 재고 서버는 2개의 텍스트 파일로 대체한다.
- (6) 고객이 환불 요청 시, 영수증 내역의 전체 상품에 대해 환불을 수행한다.

Not to be Tested

판매 요구사항 : 1 / 10

(7) 바코드 센서에서 상품의 바코드를 읽을 수 있다.

성능 요구사항 : 1 / 1

(8) 버튼에 대한 반응 속도가 1ms 이하

1단계 : ID와 설명 생성

7.1 Test design specification identifier↵

<Table : 1 Testing List>↵

ID↵	Description↵	↵
POS_STC.1↵	프로그램 종료 후 재실행 시 기존에 진행하던 시각이 이어진다↵	↵
POS_STC.2↵	프로그램의 하루는 현실시간 3분으로 가정한다↵	↵
POS_STC.3↵	상품 총 7종류를 모두 다룰 수 있는지 확인한다↵	↵
POS_STC.4↵	하루가 지나면 재고서버의 상품 파일(YYYYMMDD_product.txt)↵ 판매 관리 파일(YYYYMMDD_sale_management.txt)은 다음 날짜로 새로 생성 된다↵	↵
POS_STC.5↵	하루가 지나면 재고서버의 상품 파일(YYYYMMDD_product.txt)의 모든 상품은 100개로 초기화 한다↵	↵
POS_STC.6↵	전원이 켜지면 자동으로 재고서버의 상품을 동기화한다↵	↵

2단계 : 특징/기대값 분리

< Table 2 Test Design Identification >

Test Case Identifier	Feature	Valid value
POS_STC.1	POST 프로그램 종료 후 재실행 시 기존에 진행하던 시각이 이어진다	프로그램 종료 후 재실행 시 그 전의 시각과 연결
POS_STC.2	프로그램의 하루는 현실시간 3분으로 가정한다	현실시간 3분이 지났을 때, 프로그램에서 하루가 지남
POS_STC.3	상품 7종류를 모두 다룰 수 있는지 확인한다	상품 7종류가 모두 재고에 존재
POS_STC.4	하루가 지나면 재고서버의 상품 파일(YYYYMMDD_product.txt), 판매 관리 파일(YYYYMMDD_sale_management.txt)은 다음 날짜로 새로 생성 된다	하루가 지나고 "YYYYMMDD_product.txt", "YYYYMMDD_sale_management.txt" 새로 생성

7.5 Feature pass/fail criteria

위의 7.4의 Valid value 경우를 도출해 냈을 때, Pass이며 Valid value 경우 이외의 다른 값이 나오는 경우 Fail로 판단한다.

3단계 : input/output 구체화

< Table 2 Test Design Identification >

Test Case Identifier	Feature	Valid value
POS_STC.1	POST 프로그램 종료 후 재실행 시 기존에 진행하던 시각이 이어진다	프로그램 종료 후 재실행 시 그 전의 시각과 연결
POS_STC.2	프로그램의 하루는 현실시간 3분으로 가정한다	현실시간 3분이 지났을 때, 프로그램에서 하루가 지남



Identifier	Input Specification	Output specification
POS_STC_001_000	프로그램 종료 후 재실행	프로그램 종료 직전 날짜/시간과 재실행 했을 때 시간과 비교
POS_STC_002_000	대기	3분이 지났을 때, 날짜를 확인하여 하루가 지났는지 확인

3단계 : input/output 구체화

POS_STC.11↵	POST는 캐시 화면(터치 스크린)을 통해 고객에게 받은 금액을 <u>입력</u> , 받은 금액이 총 판매액보다 클 경우, 결제를 진행↵	POST는 캐시 화면(터치 스크린)을 통해 고객에게 받은 금액을 <u>입력</u> , 받은 금액이 총 판매액보다 클 경우, 결제를 진행 ↵
-------------	--	---



POS_STC_011_000↵	캐시 화면(터치 스크린)을 통해 금액 입력↵	금액이 캐시 화면에 제대로 입력되는지 확인↵
POS_STC_011_001↵	총 판매액보다 작은 금액 입력↵	결제 불가로 아무 일도 일어나지 않는다.↵
POS_STC_011_002↵	총 판매액보다 큰 금액 입력↵	거스름돈이 캐시 화면에 제대로 출력 되고 판매 완료 <u>정보</u> , 영수증, 상품파일 업데이트 ↵

To be Tested → 최종 List

- 2.4 제약 및 가정사항 6 → 6
- 3.2 기능요구사항 21 → 26 (+5)
 - 판매 9 → 11 (+2)
 - 환불 7 → 7
 - 정산 2 → 3 (+1)
 - 재고확인 3 → 5 (+2)
- 총 27 → 32 (+5)

STR Pass/Fail

Identifier↵	Input Specification↵	Output specification↵	Result↵
POS_STC_001_000↵	프로그램 종료 후 재 실행↵	프로그램 종료 직전 날짜/시간과 재실행 했을 때 시간과 비교↵	↵
POS_STC_002_000↵	대기↵	3분이 지났을 때, 날짜를 확인하여 하루가 지났는지 확인↵	↵
POS_STC_003_000↵	대기화면에서 재고확 인버튼 입력↵	재고 텍스트 파일에 상품 7종류 존 재하는지 확인↵	↵
POS_STC_004_000↵	하루가 지남↵	"YYYYMMDD_product.txt", "YYYYMMDD_sale_management.txt" 새로 생성 확인↵	↵

STR Pass/Fail

POS_STC_017_000	환불 가능한 영수증 입력 후 환불 취소	환불 완료 안됨	F
-----------------	--------------------------	----------	---

- Input 단계적으로 수행

기본화면 -> 환불모드 -> 영수증입력 -> 환불가능할때 취소입력

Ex)

(home) -> 4 -> 201711081315 -> -1

T1 / T2 Testing

T1

Identifier↵	Input Specification↵	Output specification↵	Result↵
POS_STC_001_000↵	프로그램 종료 후 재 실행↵	프로그램 종료 직전 날짜/시간과 재실행 했을 때 시간과 비교↵	pass↵
POS_STC_002_000↵	대기↵	3분이 지났을 때, 날짜를 확인하여 하루가 지났는지 확인↵	Pass↵
POS_STC_003_000↵	대기화면에서 재고확 인버튼 입력↵	재고 텍스트 파일에 상품 7종류 존 재하는지 확인↵	Fail↵
POS_STC_004_000↵	하루가 지남↵	"YYYYMMDD_product.txt", "YYYYMMDD_sale_management.txt" 새로 생성 확인↵	Fail↵

T2

Identifier↵	Input Specification↵	Output specification↵	Result↵
POS_STC_001_000↵	프로그램 종료 후 재 실행↵	프로그램 종료 직전 날짜/시간과 재실행 했을 때 시간과 비교↵	Pass↵
POS_STC_002_000↵	대기↵	3분이 지났을 때, 날짜를 확인하여 하루가 지났는지 확인↵	Pass↵
POS_STC_003_000↵	대기화면에서 재고확 인버튼 입력↵	재고 텍스트 파일에 상품 7종류 존 재하는지 확인↵	Fail↵
POS_STC_004_000↵	하루가 지남↵	"YYYYMMDD_product.txt", "YYYYMMDD_sale_management.txt" 새로 생성 확인↵	Pass↵

T1 Testing

```
/home/seproject2
-----
      Casher View
-----
number |name          |price          |quantity      |totalprice    |
-----
-----
Customer View
-----
Total Buy : 0
Cash : 0
Change : 0
-----
1.InBarcode 2:NumOfGoods 3:RemoveList 4:Sell 5:Refund 6:Stock 7:Time 8:Power Off
1
insert barcode :
010
```

(기본화면) -> 1(바코드입력모드) -> 010(바코드입력)

T1 Testing

```
/home/seproject2
-----
Casher View
-----
number |name      |price  |quantity |totalprice |
1      |beer     |310    |1         |310        |
-----

Customer View
-----
Total Buy : 310
Cash : 0
Change : 0

-----
1.InBarcode 2:NumOfGoods 3:RemoveList 4:Sell 5:Refund 6:Stock 7:Time 8:Power Off
|
```

판매목록에 없는 바코드 입력 시 목록에 추가 확인.(캐셔화면)

Pass

T1 Testing

```
/home/seproject2
-----
Casher View
-----
number |name      |price  |quantity |totalprice |
1      |beer     |310    |1        |310        |
-----

Customer View
-----
Total Buy : 310
Cash : 0
Change : 0

-----
1.InBarcode 2:NumOfGoods 3:RemoveList 4:Sell 5:Refund 6:Stock 7:Time 8:Power Off
file not opened! - sale_management
file not opened! - refund_management
Next Day, refresh stock!
```

정산은 기본화면인 경우에만 진행, but 판매종일때 정산이 됨

Fail

T2 Testing

```
~/2017se-master/src
-----
|          CASHIER SCREEN          |          CUSTOMER SCREEN          |
-----
*****
*          SELL STATE          *
*****
*  1. Increase Quantity  *
*  2. Decrease Quantity  *
*  3. BARCODE            *
*  4. RECEIPT           *
*****
Action : 3
Barcode : 001
```

(기본화면) -> 2(판매모드) -> 3(바코드입력모드) -> 001(바코드입력)

Fail 판매목록 추가가 화면에 출력되지않음

Pass 영수증 파일출력 가능

T2 Testing

<pre> ~/2017se-master/src ----- CASHIER SCREEN ----- ***** * REFUND STATE * ***** * 1. BARCODE * * 2. RECEIPT * * 3. CANCEL * ***** Action : 1 Barcode : 201711060736 해당 영수증은 환불이 불가능 합니다 </pre>	POS_STC_017_000↵	환불 가능한 영수증 입력 후 환불 취소↵	환불 완료 안됨↵	Fail Fail
	POS_STC_017_001↵	환불 가능한 영수증 입력 후 환불↵	환불 완료↵	Pass
	POS_STC_018_000↵	환불 불가능한 영수증 입력↵	"해당 영수증은 환불이 불가능 합 니다." 라는 메시지를 띄운다↵	Fail Fail
	POS_STC_019_000↵	환불 완료↵	<u>캐시</u> 화면(터치 스크린)와 고객 화 면에 각각 환불 정보를 띄운다↵	Fail
	POS_STC_020_000↵	환불 완료↵	환불 영수증(refund_ YYYYMMDDhhmm.txt)을 생성↵	Fail
	POS_STC_021_000↵	환불 완료↵	당일 상품 파일↵ (YYYYMMDD_product.txt)의 상품 의 개수를 업데이트↵	Fail

환불이 불가능하여 확인이 불가능한경우 Fail 처리

Testing 결과

- T1
25 Pass
7 Fail
통과율 : 25/32(78%)
- T2
12 Pass
20 Fail
통과율 : 12/32(38%)

List cheking → 기능 구현을 확인

- 가정사항 (제외)

판매 11

환불 7

정산 3

재고확인 5

총 26

Testing 결과분석 T1

- T1
 - 가정사항 - 제외
 - 판매기능 - 11/11(100%)
 - 환불기능 - 6/7(85%)
 - 정산기능 - 1/3(33%)
 - 재고확인기능 - 3/5(80%)

21/26(81%) 만족

Testing 결과분석 T2

- T2
 - 가정사항 - 제외
 - 판매기능 - 2/11(18%)
 - 환불기능 - 2/7(29%)
 - 정산기능 - 0/3(0%)
 - 재고확인기능 - 3/5(80%)

7/26(27%) 만족

QnA