Answer Report
for POST System

Project Team
T5

Date
2017-11-27

Team Information
201211355 손지웅
201611303 조정익
201610401 손하영

Table of Contents

1 Introduction

System test 결과 저희 팀에서 수행한 Test에서 3개의 fail이 T4에서 수행한 Test에서 3개의 fail이 나왔습니다. Fail된 결과를 분석해보니 저희 팀, T4에서 테스트의 공통적인 fail이 한가지 있었고 저희 팀에서 fail 된 나머지 두개의 결과는 동일한 부분에서 나와 한가지 문제로 묶을 수 있었습니다.

T4에서 fail 된 두가지 결과, T5에서 fail 된 한 가지 결과, 공통적으로 fail 된 결과에 대해 대응서를 작성하였습니다.

2 Fail Case

2.1 Case #1 (요구 사항 위반)

T4_POS.STP.000.001	바코드로 과자(001)추가 후 목록제거를 통해 0으로변경, 라면과 물을 수량1로 추가한다. 돈을 지불하고 판매 완료 뒤, 환불실시.	판매영수증에 과자가 없고 라면과 물만 있는지 확인. 영수증에도 과자가 없고 라면과 물만 1개씩인지 확인. 환불 후 환불되었는지 확인. 재고서버에 과자,물 라면 모두 수량이 100개인지 확인	Fail

POST는 캐셔 화면(터치 스크린)을 통해 판매 목록에 있는 상품의 개수를 추가, 감소시키거나 상품을 목록에서 제거 할 수 있다.

원인을 분석한 결과 목록 제거라는 부분에서 fail에 생겼습니다. 저희는 요구사항에 대해 저희는 수량 감소를 통한 목록 제거로 이해하였습니다. 요구사항을 보니 제거가 좀 더 맞는 것 같아 논란의 여지를 없애기 위해 목록을 제거하는 방식을 채택하였습니다.

2.2 Case #2 (요구 사항 위반)

3_000_000	과자(001)의 재고를 98개로 맞춰놓고 하루(3분)이 넘어갈 때를 기다린다.	상품 파일, 판매 관리 파일이 다음 날짜로 새로 생성된다.
3_000_001	1_000_006의 상태에서 판매를 하고 하루(3분)이 넘어갈 때를 기다린다.	상품 파일, 판매 관리 파일이 다음 날짜로 새로 생성된다.

3_000_000	Fail (판매 목록 파일이 날짜가 지나도 새로 생성안됨)
3_000_001	Fail('')

판매 관리 파일이 판매가 끝난 뒤에만 생성 및 업데이트가 되는 것이라고 이해하고 구현 하여서 생긴 문제였습니다. 이로 인해 하루가 지나도 판매 관리 파일이 새롭게 생성되지 않았습니다. 이에 매일 새롭게 업데이트 되는 방식을 추가하였습니다.

2.3 Case #3 (요구 사항 위반)

T4_POS.STP.003.001	메인화면에서 환불 입력한다. 그 다음, 환불되지 않은 영수증 입력후 환불확정을 입력한다.	환불기능이 제대로 작동하는지 확인	Fail

2_000_000	Fail (환불 Y/N 선택 불가능)

이 또한 요구 사항을 지키지 못한 부분으로 제대로 된 영수증 입력 시 환불 여부를 묻지 않고 바로 환불되게 하였습니다. 이에 환불 여부를 묻는 부분을 추가하였습니다.

2.4 Case #4

T4_POS.STP.000.001	바코드로 과자(001)추가 후 목록제거를 통해 0으로변경, 라면과 물을 수량1로 추가한다. 돈을 지불하고 판매 완료 뒤, 환불실시.	판매영수증에 과자가 없고 라면과 물만 있는지 확인. 영수증에도 과자가 없고 라면과 물만 1개씩인지 확인. 환불 후 환불되었는지 확인. 재고서버에 과자,물 라면 모두 수량이 100개인지 확인	Fail

이 부분은 이해가 되지 않던 부분입니다. 핵심은 재고수량을 넘겼을 시 판매가 불가하게 하는 부분이 fail 되었다는 건데 자체적으로 몇 번의 테스트를 해도 문제가 없었습니다. 그래서 T4에게 직접 물어보니 3분을 정확히 타이머로 정하여 수행하였다고 하였습니다.

프로그램 시작 시 현실시간이 3분을 나눈 기준으로 더하여서 반영되기 때문에 3분 내에서 어느 정도 지난 시점에서 하루가 지나 새롭게 재고가 업데이트 되고 그 재고를 바탕으로 판매가 이루어져서 일어났던 것으로 저희 프로그램은 이상이 없다고 판단하였습니다.

3 Answer

3.1 Case #1 (목록 제거)

```
while (1) {
    printf("1. (+) 2. (-) : 3. Delete : ");
    scanf("%d", &sel);

    if(sel < 1 || sel > 3) {
        printf("Wrong Input!\n\n");
        continue;
    }
    else {
        break;
    }
}
```

```
else {
    if (sale_info->s_arr[index - 1].quantity == 0) {
        // message passing 부분
        printf("Wrong Input!\n\n");
        return;
    }
    int p_num = 0;
    p_num = sale_info->s_arr[index - 1].quantity;
    sale_info->s_arr[index - 1].quantity = 0;
    sale_info->total_price -= (sale_info->s_arr[index - 1].price * p_num);
    stock_info->s_arr[index - 1].quantity += p_num;
    sale_display(sale_info);
}
```

3번 목록제거를 추가 하여 주고 예외처리 범위도 변경하였습니다.

이후 3번 else{} 부분으로 들어가서 일단 상품목록의 개수가 0개일 때는 목록 제거가 이루어 지면 안된다고 생각하여 에러를 출력하고 수량변경모드가 실패하고 메인으로 돌아가게 하였습니다.

제대로 목록제거가 이루어질 때 따로 변수에 현재 판매목록의 개수를 추가하고 현재 판매목록은 0개의 개수와, 전체 금액의 감소를, 재고에서는 미리 담아둔 변수를 통한 재고 수량증가를 하였습니다.

```
Casher Screen
     snack      quantity :       4 | price :    1000 | sum_price :    4000
 ice_cream      quantity :       1 | price :    1500 | sum_price :    1500
     water      quantity :       1 | price :     500 | sum_price :     500
     ramen      quantity :       1 | price :     800 | sum_price :     800
  beverage      quantity :       2 | price :    1200 | sum_price :    2400
    coffee      quantity :       2 | price :    2000 | sum_price :    4000
Total Sale Money                                                      13200
Receive Money                                                             0
Send Money                                                                0


Client Screen
Total Sale Money                                                      13200
Receive Money                                                             0
Send Money                                                                0


1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 6
1. (+) 2. (-) : 3. Delete : 3


Casher Screen
     snack      quantity :       4 | price :    1000 | sum_price :    4000
 ice_cream      quantity :       1 | price :    1500 | sum_price :    1500
     water      quantity :       1 | price :     500 | sum_price :     500
     ramen      quantity :       1 | price :     800 | sum_price :     800
    coffee      quantity :       2 | price :    2000 | sum_price :    4000
Total Sale Money                                                      10800
Receive Money                                                             0
Send Money                                                                0


Client Screen
Total Sale Money                                                      10800
Receive Money                                                             0
Send Money                                                                0
```

음료를 제거했을 시 리스트와, 총 금액의 변화입니다.

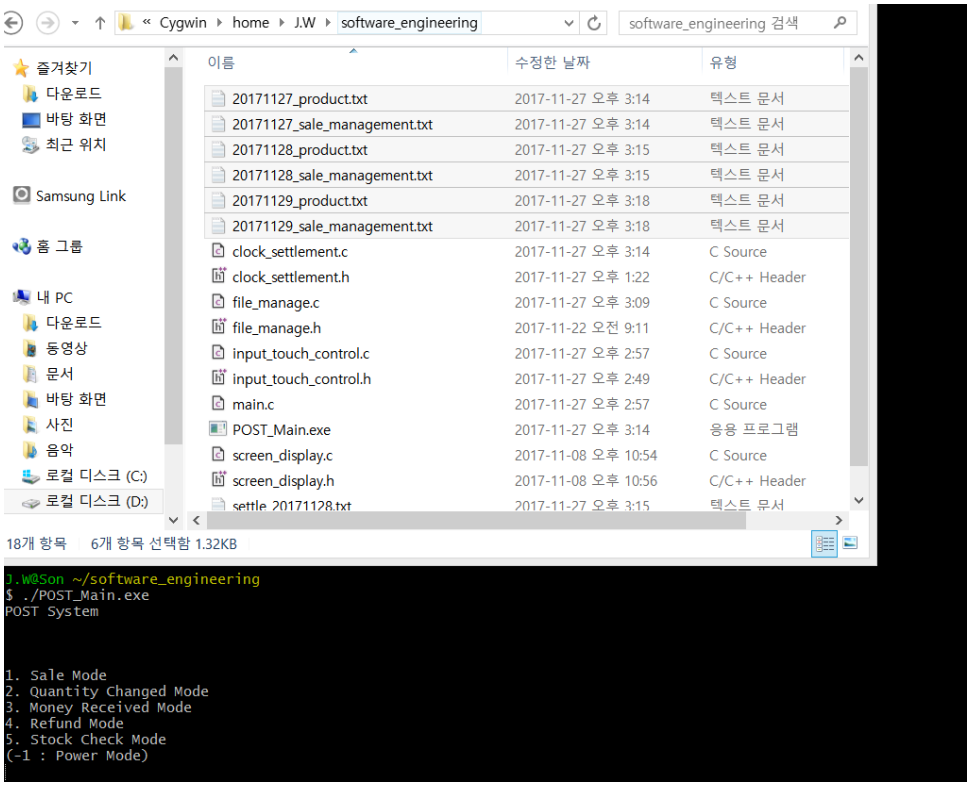
```
Money Received Mode

Received Money : 11000




Casher Screen
     snack      quantity :       4 | price :    1000 | sum_price :    4000
 ice_cream      quantity :       1 | price :    1500 | sum_price :    1500
     water      quantity :       1 | price :     500 | sum_price :     500
     ramen      quantity :       1 | price :     800 | sum_price :     800
    coffee      quantity :       2 | price :    2000 | sum_price :    4000
Total Sale Money                                                      10800
Receive Money                                                         11000
Send Money                                                              200


Client Screen
Total Sale Money                                                      10800
Receive Money                                                         11000
Send Money                                                              200


1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
5
Stock Check Mode


Casher Screen
     snack      quantity :      96 | price :    1000
 ice_cream      quantity :      99 | price :    1500
     fruit      quantity :     100 | price :    3000
     water      quantity :      99 | price :     500
     ramen      quantity :      99 | price :     800
  beverage      quantity :     100 | price :    1200
    coffee      quantity :      98 | price :    2000
```

판매도 잘 이루어 지며 재고에서도 음료의 판매가 반영되지 않은 것을 볼 수 있습니다.

3.2 Case #2 (판매 관리 목록)

```
init_arr(s_r);
init_stock_info(&stock_info);
make_stock_file(&stock_info, year, month, day);
sale_list_file(&sale_info, year, month, day, hour, min);
```

하루가 지나면 수행하는 것들 중 맨 마지막에 파일목록을 새로 생성하는 함수를 넣어서 하루가 지나면 새롭게 생기도록 하였습니다.



아무 명령을 실행하지 않아도 계속 판매관리 목록파일이 생기는 것을 볼 수 있습니다.

3.3 Case #3 (환불 유무)

```
while (1) {
    printf("1. Refund Ok 2. Refund Cancel : ");
    scanf("%d", &mode);
    if (mode == 2) {
        return 0;
    }
    else if (mode == 1) {
        break;
    }
    else {
        printf("Wrong Input!\n\n");
        continue;
    }
}
```

간단하게 환불을 할 지 안 할지 인풋을 받아 그에 따라 환불을 원래 대로 실행 시 반복문을 빠져나가고 아닐 시 바로 0을 리턴하여 환불이 불가능하게 하였습니다. Else{}부분은 예외처리 부분입니다.

3.4 Case #4 (판매 시 재고 부족)

빠른 처리를 위해 시작할 때 불러오는 save_data.txt파일에서 재고를 임의로 수정하여 실행하였습니다.

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
5
Stock Check Mode


Casher Screen
     snack      quantity :       3 | price :    1000
 ice_cream      quantity :     100 | price :    1500
     fruit      quantity :     100 | price :    3000
     water      quantity :     100 | price :     500
     ramen      quantity :     100 | price :     800
  beverage      quantity :      33 | price :    1200
    coffee      quantity :      42 | price :    2000
```

Save_data.txt를 임의로 변경하여 만든 재고를 확인한 모습입니다.

```
Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 1
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
     snack      quantity :       3 | price :    1000 | sum_price :    3000
Total Sale Money                                                       3000
Receive Money                                                             0
Send Money                                                                0


Client Screen
Total Sale Money                                                       3000
Receive Money                                                             0
Send Money                                                                0


1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 1
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
snack is sold out



1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
1

Sale Mode
Barcode Input
-> Barcode : 001


Casher Screen
Display
snack is sold out
```

이후 3개의 스낵만 남은 상태에서 3개까지 현재 장바구니에 추가한 뒤 수량 증가 혹은 바코드 입력을 해도 재고가 부족하여 sold out이라고 표기하는 부분입니다.

더 정확한 결과를 위해 한번 더 실행해보았습니다.

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
5
Stock Check Mode


Casher Screen
     snack      quantity :       1 | price :    1000
 ice_cream      quantity :       1 | price :    1500
     fruit      quantity :       1 | price :    3000
     water      quantity :       1 | price :     500
     ramen      quantity :       1 | price :     800
  beverage      quantity :       1 | price :    1200
    coffee      quantity :       1 | price :    2000
```

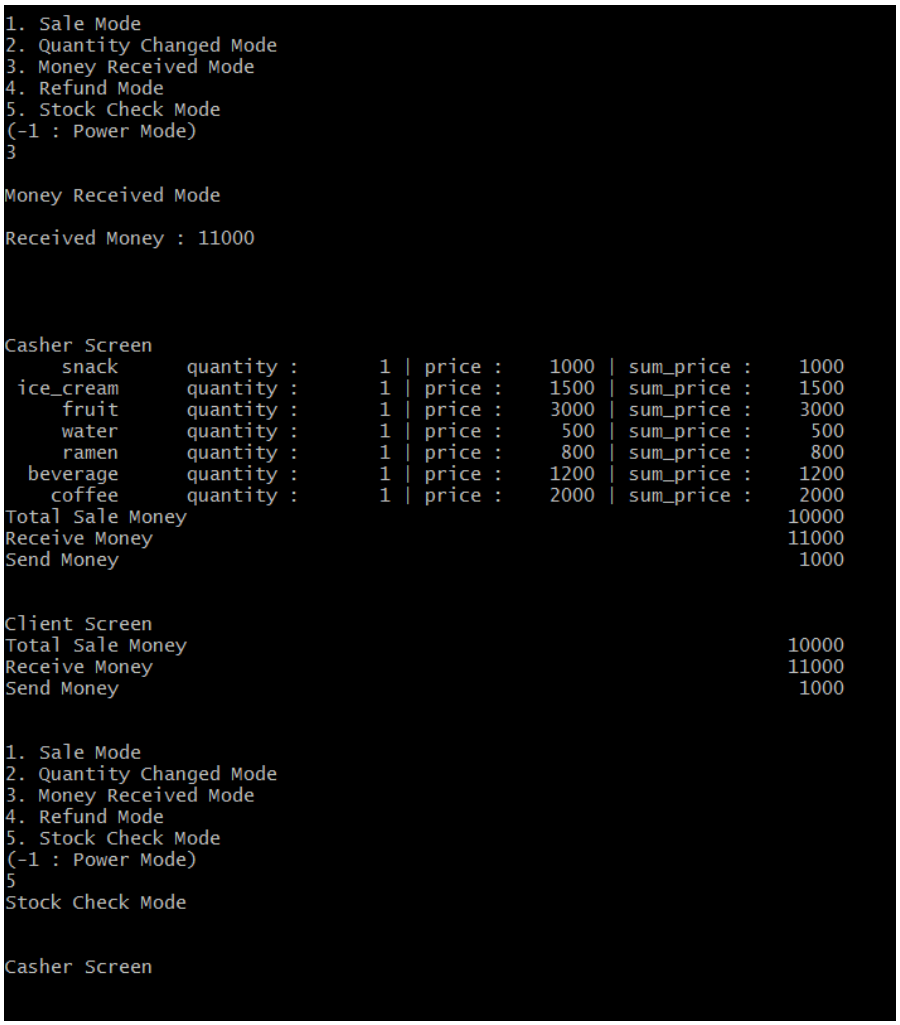
빠른 처리를 위해 시작할 때 불러오는 save_data.txt파일에서 재고를 전부 1개만 남게 수정하여 실행하였습니다.

```
Casher Screen
     snack      quantity :       1 | price :    1000 | sum_price :    1000
 ice_cream      quantity :       1 | price :    1500 | sum_price :    1500
     fruit      quantity :       1 | price :    3000 | sum_price :    3000
     water      quantity :       1 | price :     500 | sum_price :     500
     ramen      quantity :       1 | price :     800 | sum_price :     800
  beverage      quantity :       1 | price :    1200 | sum_price :    1200
    coffee      quantity :       1 | price :    2000 | sum_price :    2000
Total Sale Money                                                    10000
Receive Money                                                           0
Send Money                                                              0


Client Screen
Total Sale Money                                                    10000
Receive Money                                                           0
Send Money                                                              0


1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 1
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
snack is sold out



1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 2
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
ice_cream is sold out
```

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 3
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
fruit is sold out



1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 4
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
water is sold out



1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 5
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
ramen is sold out
```

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 6
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
beverage is sold out



1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
2

Quantity Change Mode
1. snack 2.ice cream 3.fruit 4.water 5.ramen 6.beverage 7.coffee
-> 7
1. (+) 2. (-) : 3. Delete : 1


Casher Screen
Display
coffee is sold out
```

이후 전부 수량을 증가해보았는데 모두 품절이 됐다고 출력되는 것을 볼 수 있었습니다.



이후 계산을 완료하고 재고를 확인했을 때까지 모두 정상적으로 동작하였습니다.

이후에도 다양하게 재고를 바꿔 진행해보았지만 아무 이상이 없는 것을 확인하였습니다.

결론적으로 T4의 테스트 시 판매를 한 개씩 증가하다가 재고가 중간에 채워지는 일이 발생하여 해당 케이스에서 Fail이 나왔다고 예상하고 프로그램 자체는 문제가 없다고 생각됩니다.