



System **T**esting for POS

Project Team 5 (손지웅, 조정익, 손하영)

Index

1. 요구사항 분석
 2. Features to be & not to be tested
 3. Test Case
 4. Team 5 Result
 5. Team 6 Result
-



1. 요구사항 분석

1.1 기능명세서 분석

1.2 Category 분석

1.1 기능명세서 분석

전원

출력

성능

모드

상품 입력

1. 판매 모드

2. 환불 모드

3. 재고 확인

4. 정산

1.2 Category 분석 (전원 ~ 성능)

전원

사용자는 언제든지 전원을 끌 수 있다.

출력

POST는 캐시 화면(터치 스크린)을 이용해 판매, 환불, 재고 확인을 할 수 있다.

해당 정보는 캐시 화면(터치 스크린)와 고객 화면에 각각 업데이트 된다.

POST의 모든 모니터의 출력은 Cygwin Console(Cygwin Bash Shell)로 대체한다.

POST의 프린터를 사용한 모든 출력은 텍스트 파일 생성으로 대체한다.

성능

버튼에 대한 반응 속도는 1ms 이하이다.

1.2 Category 분석 (모드 ~ 상품 입력)

모드

POST는 기본화면에서 판매를 진행할 수 있다.

POST는 사용자가 캐시 화면(터치 스크린)의 기본 화면에서 환불 버튼을 누를 시 환불을 진행할 수 있다.

정산은 POST가 기본 화면인 경우에서만 진행이 되며, 판매 또는 환불 중에는 완료가 된 후에 정산이 진행된다.

POST는 기본화면에서 재고 확인 버튼 입력 시 재고 확인을 할 수 있다.

POST는 판매 중이거나 환불 중에는 재고 확인을 할 수 없다.

상품 입력

POST는 캐시 화면(터치 스크린)을 이용해 판매, 환불, 재고 확인을 할 수 있다.

POST에서 다루는 상품은 총 7종류(과자, 아이스크림, 과일, 물, 라면, 음료수, 커피)로 한정한다.

1.2 Category 분석 (1. 판매 모드)

1. 판매 모드

POST에서 다루는 상품은 총 7종류(과자, 아이스크림, 과일, 물, 라면, 음료수, 커피)로 한정한다.

POST는 바코드로 읽은 상품이 판매 목록에 없는 경우 바코드로 읽은 상품을 판매 목록에 추가하고, 판매 목록에 있는 경우 해당 상품의 개수를 늘린다.

<- 1_000

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

<- 1_001

POST는 판매하려는 상품의 개수가 상품의 재고 개수보다 부족할 경우 캐시 화면(터치 스크린)에 "XX 개수가 부족합니다" 라는 메시지를 띄운다.

<- 1_002

POST는 캐시 화면(터치 스크린)을 통해 고객에게 받은 금액을 입력할 수 있고, 결제를 진행 할 수 있다.

<- 1_003

POST는 결제가 완료되면 캐시 화면(터치 스크린)와 고객 화면에 각각 판매 정보를 띄운다.

<- 1_004

POST는 판매가 완료되면 영수증 프린터기를 이용해 판매 영수증을 출력한다.

<- 1_005

(판매 영수증을 출력하는 것은 sale_YYYYMMDDhhmm.txt (년월일시분) 파일을 생성하는 것으로 대체한다.)

POST는 결제가 완료되면 재고 서버의 상품의 개수를 업데이트 한다.

<- 1_006

(재고 서버의 당일 상품 파일(YYYYMMDD_product.txt)의 상품의 개수를 업데이트 하는 것으로 대체한다.)

POST는 결제가 완료되면 재고 서버의 판매 정보를 추가한다.

<- 1_007

(재고 서버의 당일 판매 목록 파일(YYYYMMDD_sale_management.txt)에 현재 판매 정보를 추가하는 것으로 대체한다.)

1.2 Category 분석 (1. 판매 모드)

1. 판매 모드

캐셔 화면(터치 스크린)	
Display example	
총 판매액	20,000
현금	30,000
거스름돈	10,000
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500
...	...
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000

<- 1_008

고객 화면	
Display example	
총 판매액	20,000
현금	30,000
거스름돈	10,000

<- 1_009

영수증 양식
영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)
날짜 : YYYY.MM.DD (년.월.일)
판매 상품
상품, 단가, 수량, 판매 금액
...
상품, 단가, 수량, 판매 금액
총 판매액 : x

<- 1_010

1.2 Category 분석 (2. 환불 모드)

2. 환불 모드

고객이 환불 요청 시, 영수증 내역의 전체 상품에 대해 환불을 수행한다.

POST는 사용자가 캐시 화면(터치 스크린)의 기본 화면에서 환불 버튼을 누를 시 환불을 진행할 수 있다.

POST는 바코드 센서에서 영수증 바코드를 읽었을 때 바코드로 읽은 영수증이 이전에 발급 했던 영수증인지 재고 서버에 확인을 요청한다.

재고 서버에서 환불이 가능하다는 응답이 올 경우, 캐시 화면(터치 스크린)을 통해 환불을 완료할 수도 있고 환불을 취소할 수도 있다.

<- 2_000

재고 서버에서 환불이 불가능하다는 응답이 올 경우, 캐시 화면(터치 스크린)와 고객 화면에 각각 "해당 영수증은 환불이 불가능 합니다." 라는 메시지를 띄운다.

<- 2_001

POST는 환불이 완료되면 영수증 프린터기를 이용해 환불 영수증을 출력한다.

<- 2_002

(환불 영수증을 출력하는 것은 refund_YYYYMMDDhhmm.txt (년월일시분) 파일을 생성하는 것으로 대체한다.)

POST는 환불이 완료되면 재고 서버의 상품의 개수를 업데이트 한다.

<- 2_003

(재고 서버의 당일 상품 파일(YYYYMMDD_product.txt)의 상품의 개수를 업데이트 하는 것으로 대체한다.)

1.2 Category 분석 (2. 환불 모드)

2. 환불 모드

캐시 화면(터치 스크린)	
Display example	
판매 날짜	2017년 09월 07일 05시 20분
환불 금액	20,000
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500
...	...
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000

<- 2_004

고객 화면	
Display example	
판매 날짜	2017년 09월 07일 05시 20분
환불 금액	20,000

<- 2_005

영수증 양식
영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)
날짜 : YYYY.MM.DD (년.월.일)
환불 상품
상품, 단가, 수량, 판매 금액
...
상품, 단가, 수량, 판매 금액
환불 금액 : x

<- 2_006

1.2 Category 분석 (3. 재고 확인)

3. 재고 확인

재고 서버는 2개의 텍스트 파일로 대체한다.

(상품 파일: YYYYMMDD_product.txt, 판매 목록 파일: YYYYMMDD_sale_management.txt)

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

POST의 전원이 켜지면 자동으로 재고 서버와 상품을 동기화한다.

POST는 재고 서버에서 당일 재고 정보를 받아와 동기화를 하고 캐시 화면 (터치 스크린)에 재고 정보를 출력하고 영수증 프린터를 이용해 재고 정보를 출력한다.

(재고 확인을 출력하는 것은 stock_YYYYMMDD.txt (년월일) 파일을 생성하는 것으로 대체한다.)

POST는 판매 중이거나 환불 중에는 재고 확인을 할 수 없다.

재고 확인 양식
날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)
상품, 단가, 재고
...
상품, 단가, 재고

<- 3_000

<- 3_001

<- 3_002

<- 3_003

<- 3_004

1.2 Category 분석 (4. 정산)

4. 정산

POST 프로그램의 하루는 현실시간 3분으로 가정한다.(현실시간 1초 → 가상시간 8분) <- 4_000

POST 프로그램에서 시각 표현이 필요한 부분은 "YYYYMMDDhhmm"으로 "연월일시분"으로 표현한다.

- 201709111843

POST 프로그램 종료 후 재실행 시 기존에 진행하던 시각이 이어지도록 한다. <- 4_001

POST는 매일 12시가 되면 하루 동안 판매 했던 내역을 재고 서버에서 받아와 영수증 프린터기를 이용해 출력한다. <- 4_002

(정산을 출력하는 것은 settle_YYYYMMDD.txt (년월일) 파일을 생성하는 것으로 대체한다.)

정산은 POST가 기본 화면인 경우에서만 진행이 되며, 판매 또는 환불 중에는 완료가 된 후에 정산이 진행된다. <- 4_003

정산 영수증 양식
날짜 : YYYY.MM.DD (년.월.일)
판매 상품
상품, 단가, 수량, 금액
...
상품, 단가, 수량, 금액
판매 금액 :
환불 상품
상품, 단가, 수량, 금액
...
상품, 단가, 수량, 금액
환불 금액 :

<- 4_004

2. Features to be & not to be tested

2.1 Features to be tested

2.2 Features not to be tested

2.1 Features to be tested (전원)

Id	description
0_000	판매 모드, 환불 모드, 재고 확인 모드 중 마음대로 전원을 종료할 수 있어야 한다.

2.1 Features to be tested (1. 판매 모드)

Id	Description
1_000	바코드로 읽은 상품이 판매 목록에 없는 경우 판매 목록에 추가하고, 판매 목록에 있는 경우 해당 상품의 개수를 늘린다.
1_001	캐시 화면을 통해 판매 목록에 있는 상품의 개수를 추가/감소시키거나 상품을 목록에서 제거할 수 있다.
1_002	판매하려는 상품의 개수가 상품의 재고 개수보다 부족할 경우 캐시 화면에 "XX 개수가 부족합니다" 라는 메시지를 띄운다.
1_003	캐시 화면을 통해 고객에게 받은 금액을 입력할 수 있고, 결제를 진행할 수 있다.
1_004	결제가 완료되면 캐시화면과 고객 화면에 각각 판매 정보를 띄운다.
1_005	판매가 완료되면 판매 영수증을 출력한다. (sale_YYYYMMDDhhmm.txt 파일을 생성)
1_006	결제가 완료되면 재고 서버의 상품의 개수를 업데이트한다. (재고 서버의 당일 상품 파일(YYYYMMDD_product.txt)의 상품의 개수를 업데이트하는 것으로 대체)
1_007	결제가 완료되면 재고 서버의 판매 정보를 추가한다. (재고 서버의 당일 판매 목록 파일(YYYYMMDD_sale_management.txt)에 현재 판매 정보를 추가하는 것으로 대체)

2.1 Features to be tested (1. 판매 모드)

Id	Description																
1_008	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">캐셔 화면(터치 스크린)</th> </tr> <tr> <th colspan="2" style="text-align: center;">Display example</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">총 판매액</td> <td style="text-align: right;">20,000</td> </tr> <tr> <td style="text-align: center;">현금</td> <td style="text-align: right;">30,000</td> </tr> <tr> <td style="text-align: center;">거스름돈</td> <td style="text-align: right;">10,000</td> </tr> <tr> <td style="text-align: center;">상품</td> <td style="text-align: right;">수량 : 5 / 단가 : 500 / 판매 금액 : 2,500</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> <tr> <td style="text-align: center;">상품</td> <td style="text-align: right;">수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000</td> </tr> </tbody> </table>	캐셔 화면(터치 스크린)		Display example		총 판매액	20,000	현금	30,000	거스름돈	10,000	상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500	상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000
캐셔 화면(터치 스크린)																	
Display example																	
총 판매액	20,000																
현금	30,000																
거스름돈	10,000																
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500																
...	...																
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000																
1_009	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">고객 화면</th> </tr> <tr> <th colspan="2" style="text-align: center;">Display example</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">총 판매액</td> <td style="text-align: right;">20,000</td> </tr> <tr> <td style="text-align: center;">현금</td> <td style="text-align: right;">30,000</td> </tr> <tr> <td style="text-align: center;">거스름돈</td> <td style="text-align: right;">10,000</td> </tr> </tbody> </table>	고객 화면		Display example		총 판매액	20,000	현금	30,000	거스름돈	10,000						
고객 화면																	
Display example																	
총 판매액	20,000																
현금	30,000																
거스름돈	10,000																
1_010	<table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="padding: 5px;"> 영수증 양식 영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분) 날짜 : YYYY.MM.DD (년.월.일) 판매 상품 상품, 단가, 수량, 판매 금액 ... 상품, 단가, 수량, 판매 금액 총 판매액 : x </td> </tr> </tbody> </table>	영수증 양식 영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분) 날짜 : YYYY.MM.DD (년.월.일) 판매 상품 상품, 단가, 수량, 판매 금액 ... 상품, 단가, 수량, 판매 금액 총 판매액 : x															
영수증 양식 영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분) 날짜 : YYYY.MM.DD (년.월.일) 판매 상품 상품, 단가, 수량, 판매 금액 ... 상품, 단가, 수량, 판매 금액 총 판매액 : x																	

2.1 Features to be tested (2. 환불 모드)

Id	Description
2_000	이전에 발급했던 영수증 바코드를 읽었을 때 환불 가능하다는 응답이 오고, 캐시 화면을 통해 환불을 완료할 수도 있고 환불을 취소할 수도 있다.
2_001	이전에 발급하지 않았던 영수증 바코드를 읽었을 때 환불 불가능하다는 응답이 오고, 캐시 화면과 고객 화면에 각각 "해당 영수증은 환불이 불가능합니다"라는 메시지를 띄운다.
2_002	환불이 완료되면 환불 영수증을 출력한다. (환불 영수증을 출력하는 것은 (refund_YYYYMMDDhhmm.txt) 을 생성하는 것으로 대체)
2_003	환불이 완료되면 재고 서버의 상품의 개수를 업데이트한다. (재고 서버의 당일 상품 파일(YYYYMMDD_product.txt)의 상품의 개수를 업데이트하는 것으로 대체)

2.1 Features to be tested (2. 환불 모드)

Id	Description														
2_004	<table border="1"> <thead> <tr> <th colspan="2" data-bbox="556 242 1425 284">캐시 화면(터치 스크린)</th> </tr> <tr> <th colspan="2" data-bbox="556 284 1425 326">Display example</th> </tr> </thead> <tbody> <tr> <td data-bbox="556 326 803 368">판매 날짜</td> <td data-bbox="803 326 1425 368">2017년 09월 07일 05시 20분</td> </tr> <tr> <td data-bbox="556 368 803 410">환불 금액</td> <td data-bbox="803 368 1425 410">20,000</td> </tr> <tr> <td data-bbox="556 410 803 452">상품</td> <td data-bbox="803 410 1425 452">수량 : 5 / 단가 : 500 / 판매 금액 : 2,500</td> </tr> <tr> <td data-bbox="556 452 803 494">...</td> <td data-bbox="803 452 1425 494">...</td> </tr> <tr> <td data-bbox="556 494 803 536">상품</td> <td data-bbox="803 494 1425 536">수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000</td> </tr> </tbody> </table>	캐시 화면(터치 스크린)		Display example		판매 날짜	2017년 09월 07일 05시 20분	환불 금액	20,000	상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500	상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000
캐시 화면(터치 스크린)															
Display example															
판매 날짜	2017년 09월 07일 05시 20분														
환불 금액	20,000														
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500														
...	...														
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000														
2_005	<table border="1"> <thead> <tr> <th colspan="2" data-bbox="556 575 1414 617">고객 화면</th> </tr> <tr> <th colspan="2" data-bbox="556 617 1414 659">Display example</th> </tr> </thead> <tbody> <tr> <td data-bbox="556 659 795 701">판매 날짜</td> <td data-bbox="795 659 1414 701">2017년 09월 07일 05시 20분</td> </tr> <tr> <td data-bbox="556 701 795 744">환불 금액</td> <td data-bbox="795 701 1414 744">20,000</td> </tr> </tbody> </table>	고객 화면		Display example		판매 날짜	2017년 09월 07일 05시 20분	환불 금액	20,000						
고객 화면															
Display example															
판매 날짜	2017년 09월 07일 05시 20분														
환불 금액	20,000														
2_006	<table border="1"> <tbody> <tr> <td data-bbox="556 780 1425 822">영수증 양식</td> </tr> <tr> <td data-bbox="556 822 1425 864">영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)</td> </tr> <tr> <td data-bbox="556 864 1425 906">날짜 : YYYY.MM.DD (년.월.일)</td> </tr> <tr> <td data-bbox="556 906 1425 948">환불 상품</td> </tr> <tr> <td data-bbox="556 948 1425 990">상품, 단가, 수량, 판매 금액</td> </tr> <tr> <td data-bbox="556 990 1425 1033">...</td> </tr> <tr> <td data-bbox="556 1033 1425 1075">상품, 단가, 수량, 판매 금액</td> </tr> <tr> <td data-bbox="556 1075 1425 1117">환불 금액 : x</td> </tr> </tbody> </table>	영수증 양식	영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)	날짜 : YYYY.MM.DD (년.월.일)	환불 상품	상품, 단가, 수량, 판매 금액	...	상품, 단가, 수량, 판매 금액	환불 금액 : x						
영수증 양식															
영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)															
날짜 : YYYY.MM.DD (년.월.일)															
환불 상품															
상품, 단가, 수량, 판매 금액															
...															
상품, 단가, 수량, 판매 금액															
환불 금액 : x															

2.1 Features to be tested (3. 재고 확인)

Id	Description					
3_000	(상품 파일: YYYYMMDD_product.txt, 판매 목록 파일: YYYYMMDD_sale_management.txt) 의 2개의 재고서버가 하루(3분)가 지나면 상품 파일, 판매 관리 파일은 다음 날짜로 새로 생성이 된다.					
3_001	하루(3분)가 지나면 상품 파일의 모든 상품은 100개로 초기화된다.					
3_002	재고 서버에서 당일 재고 정보를 받아와 동기화를 하고 캐시 화면에 재고 정보를 출력하고 영수증 프린터기를 이용해 재고 정보를 출력한다. ((stock_YYYYMMDD.txt)파일을 생성하는 것으로 대체)					
3_003	판매 중이거나 환불 중에는 재고 확인을 할 수 없다.					
3_004	<table border="1"><tr><td data-bbox="560 727 1553 775">재고 확인 양식</td></tr><tr><td data-bbox="560 780 1553 828">날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)</td></tr><tr><td data-bbox="560 833 1553 881">상품, 단가, 재고</td></tr><tr><td data-bbox="560 886 1553 934">...</td></tr><tr><td data-bbox="560 939 1553 987">상품, 단가, 재고</td></tr></table>	재고 확인 양식	날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)	상품, 단가, 재고	...	상품, 단가, 재고
재고 확인 양식						
날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)						
상품, 단가, 재고						
...						
상품, 단가, 재고						

2.1 Features to be tested (4. 정산)

Id	Description
4_000	프로그램 종료 후 재실행 시 기존에 진행하던 시각이 이어지도록 한다.
4_001	현실 시간 3분이 지나면 하루가 넘어가며 (settle_YYYYMMDD.txt)파일을 생성한다.
4_002	하루 동안 판매했던 내역을 재고 서버에서 받아와 영수증 프린터기를 이용해 출력한다. (정산을 출력하는 것은 (settle_YYYYMMDD.txt)파일을 생성하는 것으로 대체한다.)
4_003	정산은 POST가 기본 화면인 경우에서만 진행이 되며, 판매 또는 환불 중에는 완료가 된 후에 정산이 진행된다.
4_004	<div data-bbox="542 756 1470 1129" style="border: 1px solid black; padding: 5px;"><p>정산 영수증 양식</p><p>날짜 : YYYY.MM.DD (년.월.일)</p><p>판매 상품</p><p>상품, 단가, 수량, 금액</p><p>...</p><p>상품, 단가, 수량, 금액</p><p>판매 금액 :</p><p>환불 상품</p><p>상품, 단가, 수량, 금액</p><p>...</p><p>상품, 단가, 수량, 금액</p><p>환불 금액 :</p></div>

2.2 Features not to be tested

출력

POST는 캐시 화면(터치 스크린)을 이용해 판매, 환불, 재고 확인을 할 수 있다.

해당 정보는 캐시 화면(터치 스크린)와 고객 화면에 각각 업데이트 된다.

POST의 모든 모니터의 출력은 Cygwin Console(Cygwin Bash Shell)로 대체한다.

POST의 프린터를 사용한 모든 출력은 텍스트 파일 생성으로 대체한다.

성능

버튼에 대한 반응 속도는 1ms 이하이다.

=> 다른 테스트를 진행하면서 확인할 수 있는 사항들이다.

2.2 Features not to be tested

모드

POST는 기본화면에서 판매를 진행할 수 있다.

POST는 사용자가 캐시 화면(터치 스크린)의 기본 화면에서 환불 버튼을 누를 시 환불을 진행할 수 있다.

정산은 POST가 기본 화면인 경우에서만 진행이 되며, 판매 또는 환불 중에는 완료가 된 후에 정산이 진행된다.

POST는 기본화면에서 재고 확인 버튼 입력 시 재고 확인을 할 수 있다.

POST는 판매 중이거나 환불 중에는 재고 확인을 할 수 없다.

상품 입력

POST는 캐시 화면(터치 스크린)을 이용해 판매, 환불, 재고 확인을 할 수 있다.

=> 다른 테스트를 진행하면서 확인할 수 있는 사항들이다.

2.2 Features not to be tested

2. 환불 모드

고객이 환불 요청 시, 영수증 내역의 전체 상품에 대해 환불을 수행한다.

POST는 사용자가 캐시 화면(터치 스크린)의 기본 화면에서 환불 버튼을 누를 시 환불을 진행할 수 있다.

POST는 바코드 센서에서 영수증 바코드를 읽었을 때 바코드로 읽은 영수증이 이전에 발급 했던 영수증인지 재고 서버에 확인을 요청한다.

=> 다른 테스트를 진행하면서 확인할 수 있는 사항들이다.

3. Test Case

3.1 전원

3.2 판매 모드

3.3 환불 모드

3.4 재고 확인 모드

3.5 정산

3.1 Test Case (전원)

Id	Input	Output (Expected)
0_000_000	판매 모드 진행 중 전원을 종료한다.	전원이 종료된다.
0_000_001	환불 모드 진행 중 전원을 종료한다.	전원이 종료된다.
0_000_002	재고 확인 모드 진행 중 전원을 종료한다.	전원이 종료된다.

3.1 Test Case (1. 판매 모드)

Id	Input	Output (Expected)
1_000_000	바코드 001 입력	과자(001)가 추가
1_000_001	바코드 010 입력	아이스크림(010)가 추가
1_000_002	바코드 011 입력	과일(011)이 추가
1_000_003	바코드 100 입력	물(100)이 추가
1_000_004	바코드 101 입력	라면(101)이 추가
1_000_005	바코드 110 입력	음료수(110)이 추가
1_000_006	바코드 111 입력	커피(111)가 추가
1_000_007	바코드 000 입력	예외 처리
1_000_008	바코드 001 입력 후 001 입력	과자(001)가 2개 추가된다.

3.1 Test Case (1. 판매 모드)

Id	Input	Output (Expected)
1_001_000	1_000_000상태에서 과자를 3개 추가	과자(001)가 5개가 된다
1_001_001	1_001_000상태에서 과자를 4개 감소	과자(001)가 1개가 된다
1_001_002	1_001_001상태에서 과자를 1개 감소시켜 제거	과자(001)이 목록에서 제거된다
1_001_003	1_001_002상태에서 과자를 1개 감소	예외 처리
1_002_000	하루가 지나기 전에 001 상품을 102개 판매하려고 한다.	"개수가 부족합니다"
1_003_000	1_000_000상태에서 판매 목록의 합계 < 고객에게 받은 돈	결제가 진행된다.
1_003_001	1_000_000상태에서 판매 목록의 합계 > 고객에게 받은 돈	결제가 진행되지 않는다.

3.1 Test Case (1. 판매 모드)

Id	Input	Output (Expected)																
1_004_000	1_003_000	캐셔 화면과 고객 화면에 판매 정보가 업데이트된다.																
1_005_000	1_003_000	sale_YYYYMDDhhmm.txt 파일이 생성된다.																
1_006_000	1_003_000	YYYYMMDD_product.txt의 상품의 개수가 업데이트된다.																
1_007_000	1_003_000	YYYYMMDD_sale_management.txt에 현재 판매 정보가 추가된다.																
1_008_000	1_003_000	<table border="1" data-bbox="1118 633 1680 811"> <thead> <tr> <th colspan="2">캐셔 화면(터치 스크린)</th> </tr> <tr> <th colspan="2">Display example</th> </tr> </thead> <tbody> <tr> <td>총 판매액</td> <td>20,000</td> </tr> <tr> <td>현금</td> <td>30,000</td> </tr> <tr> <td>거스름돈</td> <td>10,000</td> </tr> <tr> <td>상품</td> <td>수량 : 5 / 단가 : 500 / 판매 금액 : 2,500</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>상품</td> <td>수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000</td> </tr> </tbody> </table>	캐셔 화면(터치 스크린)		Display example		총 판매액	20,000	현금	30,000	거스름돈	10,000	상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500	상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000
캐셔 화면(터치 스크린)																		
Display example																		
총 판매액	20,000																	
현금	30,000																	
거스름돈	10,000																	
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500																	
...	...																	
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000																	
1_009_000	1_003_000	<table border="1" data-bbox="1118 852 1692 977"> <thead> <tr> <th colspan="2">고객 화면</th> </tr> <tr> <th colspan="2">Display example</th> </tr> </thead> <tbody> <tr> <td>총 판매액</td> <td>20,000</td> </tr> <tr> <td>현금</td> <td>30,000</td> </tr> <tr> <td>거스름돈</td> <td>10,000</td> </tr> </tbody> </table>	고객 화면		Display example		총 판매액	20,000	현금	30,000	거스름돈	10,000						
고객 화면																		
Display example																		
총 판매액	20,000																	
현금	30,000																	
거스름돈	10,000																	
1_010_000	1_003_000	<table border="1" data-bbox="1132 1012 1680 1140"> <tbody> <tr> <td>영수증 양식</td> </tr> <tr> <td>영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)</td> </tr> <tr> <td>날짜 : YYYY.MM.DD (년.월.일)</td> </tr> <tr> <td>판매 상품</td> </tr> <tr> <td>상품, 단가, 수량, 판매 금액</td> </tr> <tr> <td>...</td> </tr> <tr> <td>상품, 단가, 수량, 판매 금액</td> </tr> <tr> <td>총 판매액 : x</td> </tr> </tbody> </table>	영수증 양식	영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)	날짜 : YYYY.MM.DD (년.월.일)	판매 상품	상품, 단가, 수량, 판매 금액	...	상품, 단가, 수량, 판매 금액	총 판매액 : x								
영수증 양식																		
영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)																		
날짜 : YYYY.MM.DD (년.월.일)																		
판매 상품																		
상품, 단가, 수량, 판매 금액																		
...																		
상품, 단가, 수량, 판매 금액																		
총 판매액 : x																		

3.1 Test Case (2. 환불 모드)

Id	Input	Output (Expected)
2_000_000	1_003_000에서 발급했던 영수증 바코드를 입력하고 환불 취소를 입력한다.	환불이 되지 않는다.
2_000_001	2_000_000에 이어 1_003_000에서 발급했던 영수증 바코드를 입력하고 환불을 완료한다.	환불이 된다.
2_000_002	2_000_001에 이어 1_003_000에서 발급했던 영수증 바코드를 입력한다.	"환불 불가능"
2_000_003	1_000_006에 이어 판매를 종료한 후 발급했던 영수증 바코드를 입력하고 환불을 완료한다.	환불이 된다.
2_000_004	2_000_003에 이어 1_000_006에서 발급했던 영수증 바코드를 입력한다.	"환불 불가능"
2_001_000	발급하지 않았던 임의의 영수증 바코드를 입력한다.	"환불 불가능" 메시지가 나온다.

3.1 Test Case (2. 환불 모드)

Id	Input	Output (Expected)														
2_002_000	2_000_001	refund_YYYYMMDDhhmm.txt 을 생성														
2_002_001	2_000_003	refund_YYYYMMDDhhmm.txt 을 생성														
2_003_000	2_000_001	YYYYMMDD_product.txt의 상품의 개수를 업데이트한다.														
2_003_001	2_000_003	YYYYMMDD_product.txt의 상품의 개수를 업데이트한다.														
2_004_000	2_000_001	<table border="1" data-bbox="1136 575 1711 771"> <thead> <tr> <th colspan="2">캐서 화면(터치 스크린)</th> </tr> <tr> <th colspan="2">Display example</th> </tr> </thead> <tbody> <tr> <td>판매 날짜</td> <td>2017년 09월 07일 05시 20분</td> </tr> <tr> <td>환불 금액</td> <td>20,000</td> </tr> <tr> <td>상품</td> <td>수량 : 5 / 단가 : 500 / 판매 금액 : 2,500</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>상품</td> <td>수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000</td> </tr> </tbody> </table>	캐서 화면(터치 스크린)		Display example		판매 날짜	2017년 09월 07일 05시 20분	환불 금액	20,000	상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500	상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000
캐서 화면(터치 스크린)																
Display example																
판매 날짜	2017년 09월 07일 05시 20분															
환불 금액	20,000															
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500															
...	...															
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000															
2_005_000	2_000_001	<table border="1" data-bbox="1120 819 1754 934"> <thead> <tr> <th colspan="2">고객 화면</th> </tr> <tr> <th colspan="2">Display example</th> </tr> </thead> <tbody> <tr> <td>판매 날짜</td> <td>2017년 09월 07일 05시 20분</td> </tr> <tr> <td>환불 금액</td> <td>20,000</td> </tr> </tbody> </table>	고객 화면		Display example		판매 날짜	2017년 09월 07일 05시 20분	환불 금액	20,000						
고객 화면																
Display example																
판매 날짜	2017년 09월 07일 05시 20분															
환불 금액	20,000															
2_006_000	2_000_001	<table border="1" data-bbox="1120 970 1619 1155"> <tbody> <tr> <td>영수증 양식</td> </tr> <tr> <td>영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)</td> </tr> <tr> <td>날짜 : YYYY.MM.DD (년.월.일)</td> </tr> <tr> <td>환불 상품</td> </tr> <tr> <td>상품, 단가, 수량, 판매 금액</td> </tr> <tr> <td>...</td> </tr> <tr> <td>상품, 단가, 수량, 판매 금액</td> </tr> <tr> <td>환불 금액 : x</td> </tr> </tbody> </table>	영수증 양식	영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)	날짜 : YYYY.MM.DD (년.월.일)	환불 상품	상품, 단가, 수량, 판매 금액	...	상품, 단가, 수량, 판매 금액	환불 금액 : x						
영수증 양식																
영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)																
날짜 : YYYY.MM.DD (년.월.일)																
환불 상품																
상품, 단가, 수량, 판매 금액																
...																
상품, 단가, 수량, 판매 금액																
환불 금액 : x																

3.1 Test Case (3. 재고 확인)

Id	Input	Output (Expected)
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_001_000	과자(001)의 재고를 98개로 맞춰놓고 하루(3분)이 넘어갈 때를 기다린다.	상품파일의 모든 상품은 100개로 초기화한다.
3_001_001	1_000_006의 상태에서 판매를 하고 하루(3분)이 넘어갈 때를 기다린다.	상품파일의 모든 상품은 100개로 초기화한다.
3_002_000	하루(3분)이 지나기 전에 과자(001)의 재고를 98개로 맞춰놓고 전원을 껐다 키고 재고 확인을 한다.	캐시 화면에 재고 정보가 출력되고 stock_YYYYMMDD.txt파일을 이용해 재고 정보를 출력한다. 당일 재고 정보이므로 과자의 재고는 98개여야 한다.
3_002_001	하루(3분)이 지나기 전에 1_000_006의 상태에서 판매를 하고 전원을 껐다 키고 재고 확인을 한다.	캐시 화면에 재고 정보가 출력되고 stock_YYYYMMDD.txt파일을 이용해 재고 정보를 출력한다. 당일 재고 정보이므로 상품의 재고는 99개여야 한다.

3.1 Test Case (3. 재고 확인)

Id	Input	Output (Expected)					
3_003_000	판매 중일 때 재고 확인이 되는지 확인한다.	불가능해야 한다.					
3_003_001	환불 중일 때 재고 확인이 되는지 확인한다.	불가능해야 한다.					
3_004_000	3_001_000	<table border="1"><tr><td data-bbox="1122 422 1754 452">재고 확인 양식</td></tr><tr><td data-bbox="1122 452 1754 482">날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)</td></tr><tr><td data-bbox="1122 482 1754 512">상품, 단가, 재고</td></tr><tr><td data-bbox="1122 512 1754 543">...</td></tr><tr><td data-bbox="1122 543 1754 573">상품, 단가, 재고</td></tr></table>	재고 확인 양식	날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)	상품, 단가, 재고	...	상품, 단가, 재고
재고 확인 양식							
날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)							
상품, 단가, 재고							
...							
상품, 단가, 재고							

3.1 Test Case (4. 정산)

Id	Input	Output (Expected)
4_000_000	프로그램 종료하기 전에 시간을 체크하고 재실행 시 기존 시각이 이어지는지 체크한다.	기존 시각이 이어져야 한다.
4_001_000	현실 시간 3분이 지나갔을 때 하루가 넘어가며 settle_YYYYMMDD.txt가 생성되는지 확인한다. (기본화면)	하루가 넘어가야 하며 settle_YYYYMMDD.txt가 생성되는지 확인한다.
4_001_001	현실 시간 6분이 지나가고 settle_YYYYMMDD.txt가 생성되는지 확인한다. (기본화면)	settle_YYYYMMDD.txt가 생성되는지 확인한다.
4_001_002	현실 시간 9분이 지나가고 settle_YYYYMMDD.txt가 생성되는지 확인한다. (기본화면)	settle_YYYYMMDD.txt가 생성되는지 확인한다.
4_002_000	하루가 지나가기 전에 1_003_000을 진행하고 4_001_000을 진행한다.	settle_YYYYMMDD.txt가 생성되며 1_003_000의 정보가 있어야 한다.

3.1 Test Case (4. 정산)

Id	Input	Output (Expected)
4_003_000	프로그램을 판매 화면으로 두고 3분을 기다린다.	정산이 진행되지 않는다.
4_003_001	4_003_000의 상태에서 판매 모드를 완료한다	정산이 진행된다.
4_003_002	프로그램을 환불 화면으로 두고 3분을 기다린다.	정산이 진행되지 않는다.
4_003_003	4_003_001의 상태에서 환불 모드를 완료한다	정산이 진행된다.
4_004_000	4_002_000	<div style="border: 1px solid black; padding: 5px;"> <p>정산 영수증 양식</p> <p>날짜 : YYYY.MM.DD (년.월.일)</p> <p>판매 상품</p> <p>상품, 단가, 수량, 금액</p> <p>...</p> <p>상품, 단가, 수량, 금액</p> <p>판매 금액 :</p> <p>환불 상품</p> <p>상품, 단가, 수량, 금액</p> <p>...</p> <p>상품, 단가, 수량, 금액</p> <p>환불 금액 :</p> </div>
4_004_001	4_003_003	'''

4. Team 5 Result

4.1 Test Result(0)

4.2 Test Result(1)

4.3 Test Result(2)

4.4 Test Result(3)

4.5 Test Result(4)

4.1 Test Result (0_000)

Id	Result
0_000_000	Passed
0_000_001	Passed
0_000_002	Passed

```
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
  snack    quantity :    1 | price : 1000 | sum_price : 1000
Total Sale Money      1000
Receive Money          0
Send Money             0
Client Screen
Total Sale Money      1000
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)
-1
1. Power On 2. Power Off : 2
POS OFF
$ 정보 @-----PC ~/team5
```

<0_000_000>

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
4
Refund Mode
Refund Receipt Barcode : 11
Casher Screen
Display
Can not Refund this Receipt
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
-1
1. Power On 2. Power Off : 2
POS OFF
$ 정보 @-----PC ~/team5
```

<0_000_001>

```
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 :    98 | 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 :   100 | price : 1200
  coffee   quantity :   100 | price : 2000
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
-1
1. Power On 2. Power Off : 2
POS OFF
$ 정보 @-----PC ~/team5
```

<0_000_002>

4.2 Test Result (1_000)

Id	Result
1_000_000	Passed
1_000_001	Passed
1_000_002	Passed
1_000_003	Passed
1_000_004	Passed
1_000_005	Passed
1_000_006	Passed
1_000_007	Passed
1_000_008	Passed

4.2 Test Result (1_000)

```
Sale Mode
Barcode Input
-> Barcode : 001

Casher Screen
  snack  quantity :    1 | price :   1000 | sum_price :   1000
Total Sale Money          1000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          1000
Receive Money              0
Send Money                 0
```

<1_000_000>

```
Sale Mode
Barcode Input
-> Barcode : 010

Casher Screen
  snack  quantity :    1 | price :   1000 | sum_price :   1000
  ice_cream quantity :    1 | price :   1500 | sum_price :   1500
Total Sale Money          2500
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          2500
Receive Money              0
Send Money                 0
```

<1_000_001>

```
Barcode Input
-> Barcode : 011

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
Total Sale Money          5500
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          5500
Receive Money              0
Send Money                 0
```

<1_000_002>

```
Sale Mode
Barcode Input
-> Barcode : 100

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
Total Sale Money          6000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          6000
Receive Money              0
Send Money                 0
```

<1_000_003>

```
Sale Mode
Barcode Input
-> Barcode : 101

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
Total Sale Money          6800
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          6800
Receive Money              0
Send Money                 0
```

<1_000_004>

```
Sale Mode
Barcode Input
-> Barcode : 110

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
Total Sale Money          8000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          8000
Receive Money              0
Send Money                 0
```

<1_000_005>

4.2 Test Result (1_000)

```
Sale Mode
Barcode Input
-> Barcode : 111

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_000_006>

```
Sale Mode
Barcode Input
-> Barcode : 000
Wrong Input!
```

<1_000_007>

```
Sale Mode
Barcode Input
-> Barcode : 001

Casher Screen
  snack quantity : 1 | price : 1000 | sum_price : 1000
Total Sale Money 1000
Receive Money 0
Send Money 0

Client Screen
Total Sale Money 1000
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)
1

Sale Mode
Barcode Input
-> Barcode : 001

Casher Screen
  snack quantity : 2 | price : 1000 | sum_price : 2000
Total Sale Money 2000
Receive Money 0
Send Money 0

Client Screen
Total Sale Money 2000
Receive Money 0
Send Money 0
```

<1_000_008>

4.2 Test Result (1_001)

Id	Result
1_001_000	Passed
1_001_001	Passed
1_001_002	Passed
1_001_003	Passed

4.2 Test Result (1_001)

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

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

Casher Screen
   snack   quantity :    4 | price :   1000 | sum_price :   4000
Total Sale Money          4000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          4000
Receive Money              0
Send Money                 0

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

Casher Screen
   snack   quantity :    5 | price :   1000 | sum_price :   5000
Total Sale Money          5000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          5000
Receive Money              0
Send Money                 0
```

<1_001_000>

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

Casher Screen
   snack   quantity :    4 | price :   1000 | sum_price :   4000
Total Sale Money          4000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          4000
Receive Money              0
Send Money                 0

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

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

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

Casher Screen
   snack   quantity :    2 | price :   1000 | sum_price :   2000
Total Sale Money          2000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          2000
Receive Money              0
Send Money                 0

Casher Screen
   snack   quantity :    1 | price :   1000 | sum_price :   1000
Total Sale Money          1000
Receive Money              0
Send Money                 0

Client Screen
Total Sale Money          1000
Receive Money              0
Send Money                 0
```

<1_001_001>

4.2 Test Result (1_001)

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

Casher Screen
Total Sale Money      0
Receive Money         0
Send Money            0

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

<1_001_002>

```
Casher Screen
Total Sale Money      0
Receive Money         0
Send Money            0

Client Screen
Total Sale Money      0
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. (-) : 2
Wrong Input!
```

<1_001_003>

4.2 Test Result (1_002)

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

```
Casher Screen
    snack      quantity :    102 | price :    1000 | sum_price : 102000
Total Sale Money                                102000
Receive Money                                    0
Send Money                                        0
```

```
Client Screen
Total Sale Money                                102000
Receive Money                                    0
Send Money                                        0
```

4.2 Test Result (1_002~1_003)

Id	Result
1_002_000	Passed
1_003_000	Passed
1_003_001	Passed

```
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. (-) : 1
Casher Screen
Display
snack is sold out
```

<1_002_000>

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
3
Money Received Mode
Received Money : 1000
Casher Screen
snack quantity : 1 | price : 1000 | sum_price : 1000
Total Sale Money 1000
Receive Money 1000
Send Money 0
Client Screen
Total Sale Money 1000
Receive Money 1000
Send Money 0
```

<1_003_000>

```
Money Received Mode
Received Money : 500
Received Money : 500
Received Money : 500
Received Money : 100
Received Money : 1000
Casher Screen
snack quantity : 1 | price : 1000 | sum_price : 1000
Total Sale Money 1000
Receive Money 1000
Send Money 0
Client Screen
Total Sale Money 1000
Receive Money 1000
Send Money 0
```

<1_003_001>

4.2 Test Result (1_004~1_010)

Id	Result
1_004_000	Passed
1_005_000	Passed
1_006_000	Passed
1_007_000	Passed
1_008_000	Passed
1_009_000	Passed
1_010_000	Passed

```
Casher Screen
  snack      quantity :      1 | price :   1000 | sum_price :   1000
Total Sale Money
Receive Money
Send Money

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

<1_004_000, 1_008_000, 1_009_000>

4.2 Test Result (1_004~1_010)

sale_201712142350.txt 2017-11-20 오후... 텍스트 문서

<1_005_000>

20171214_product.txt 2017-11-20 오후... 텍스트 문서

20171214_sale_management.txt 2017-11-20 오후... 텍스트 문서

<1_006_000, 1_007_000>

sale_201712142350.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Receipt Num : 2017.12.14.23.50 (년.월.일.시.분)

Date : 2017.12.14 (년.월.일)

Sale Product

Name, Price, Quantity, Sum_price

snack, 1000, 1, 1000

Total Sale Price : 1000

<1_010_000>

4.3 Test Result (2_000~2_001)

Id	Result
2_000_000	Fail (환불 Y/N 선택 불가능)
2_000_001	Passed
2_000_002	Passed
2_000_003	Passed
2_000_004	Passed
2_001_000	Passed

4.3 Test Result (2_002~2_006)

Id	Result
2_002_000	Passed
2_002_001	Passed
2_003_000	Passed
2_003_001	Passed
2_004_000	Passed
2_005_000	Passed
2_006_000	Passed

4.3 Test Result (2_000)

sale_201711201535.txt

2017-11-20 오후... 텍스트 문서

1KB 2

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
4
Refund Mode
Refund Receipt Barcode : 201711201535

Casher Screen
Date                2017.11.20 20:24
snack  quantity :   1000 | price :    1 | sum_price :   1000
Total Refund Money                1000

Client Screen
Date                2017.11.20 20:24
Total Refund Money                1000
```

<2_000_001>

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
4
Refund Mode
Refund Receipt Barcode : 201711201535

Casher Screen
Display
Can not Refund this Receipt
```

<2_000_002>

sale_201711202211.txt

2017-11-20 오후... 텍스트 문서

1KB

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
4
Refund Mode
Refund Receipt Barcode : 201711202211

Casher Screen
Date                2017.11.21 10:08
snack  quantity :   1000 | price :    1 | sum_price :   1000
ice_cream quantity :   1500 | price :    1 | sum_price :   1500
fruit  quantity :   3000 | price :    1 | sum_price :   3000
water  quantity :    500 | price :    1 | sum_price :    500
ramen  quantity :    800 | price :    1 | sum_price :    800
beverage quantity :   1200 | price :    1 | sum_price :   1200
coffee quantity :   2000 | price :    1 | sum_price :   2000
Total Refund Money                10000

Client Screen
Date                2017.11.21 10:08
Total Refund Money                10000
```

<2_000_003>

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
4
Refund Mode
Refund Receipt Barcode : 201711202211

Casher Screen
Display
Can not Refund this Receipt
```

<2_000_004>

4.3 Test Result (2_001~2_003)

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
4
Refund Mode
Refund Receipt Barcode : 1101201

Casher Screen
Display
Can not Refund this Receipt
```

<2_001_000>

refund_201711202024.txt

2017-11-20 오후... 텍스트 문서

1KB

<2_002_000>

refund_201711211008.txt

2017-11-20 오후... 텍스트 문서

1KB

<2_002_001>

```
Date : 2017.11.20 (년.월.일)
Sale Product
Name, Price, Quantity
snack, 1000, 100
Name, Price, Quantity
ice_cream, 1500, 100
Name, Price, Quantity
fruit, 3000, 100
Name, Price, Quantity
water, 500, 100
Name, Price, Quantity
ramen, 800, 100
Name, Price, Quantity
beverage, 1200, 100
Name, Price, Quantity
coffee, 2000, 100
```

```
Date : 2017.11.21 (년.월.일)
Sale Product
Name, Price, Quantity
snack, 1000, 101
Name, Price, Quantity
ice_cream, 1500, 101
Name, Price, Quantity
fruit, 3000, 101
Name, Price, Quantity
water, 500, 101
Name, Price, Quantity
ramen, 800, 101
Name, Price, Quantity
beverage, 1200, 101
Name, Price, Quantity
coffee, 2000, 101
```

<2_003_000>

<2_003_001>

4.3 Test Result (2_004~2_006)

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
4
Refund Mode
Refund Receipt Barcode : 201711201535

Casher Screen
Date                2017.11.20 20:24
snack      quantity :   1000 | price :    1 | sum_price :   1000
Total Refund Money                                1000

Client Screen
Date                2017.11.20 20:24
Total Refund Money                                1000
```

<2_004_000, 2_005_000>

refund_201711202024.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Receipt Num : 2017.11.20.20.24 (년.월.일.시.분)

Date : 2017.11.20 (년.월.일)

Refund Product

Name, Price, Quantity, Sum_price

snack, 1000, 1, 1000





Total Refund Price : 1000





<2_006_000>

4.4 Test Result (3_000~3_003)






Id	Result
3_000_000	Fail (판매 목록 파일이 날짜가 지나도 새로 생성안됨)
3_000_001	Fail("")
3_001_000	Passed
3_001_001	Passed
3_002_000	Passed
3_002_001	Passed
3_003_000	Passed
3_003_001	Passed
3_004_000	Passed




4.4 Test Result (3_000)

 20171120_product.txt	2017-11-20 오후 3:59
 20171120_sale_management.txt	2017-11-20 오후 3:59
 sale_201711201558.txt	2017-11-20 오후 3:59
 sale_receipt_list.txt	2017-11-20 오후 3:59

 20171120_product.txt	2017-11-20 오후 3:59	텍스트 문서
 20171120_sale_management.txt	2017-11-20 오후 3:59	텍스트 문서
 20171121_product.txt	2017-11-20 오후 4:02	텍스트 문서
 sale_201711201558.txt	2017-11-20 오후 3:59	텍스트 문서

<3_000_000>

 20171120_product.txt	2017-11-20 오후 4:16
 20171120_sale_management.txt	2017-11-20 오후 4:16
 sale_201711202014.txt	2017-11-20 오후 4:16
 sale_receipt_list.txt	2017-11-20 오후 4:16
 stock_20171120.txt	2017-11-20 오후 4:16

 20171120_product.txt	2017-11-20 오후 4:16	텍스트 문서
 20171120_sale_management.txt	2017-11-20 오후 4:16	텍스트 문서
 20171121_product.txt	2017-11-20 오후 4:18	텍스트 문서

<3_000_001>

4.4 Test Result (3_001)

20171120_product.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 5

```
Date : 2017.11.20 (년.월.일)
Sale Product
Name, Price, Quantity
snack, 1000, 99
Name, Price, Quantity
ice_cream, 1500, 99
Name, Price, Quantity
fruit, 3000, 99
Name, Price, Quantity
water, 500, 99
Name, Price, Quantity
ramen, 800, 99
Name, Price, Quantity
beverage, 1200, 99
Name, Price, Quantity
coffee, 2000, 99
```

20171120_product.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 5

```
Date : 2017.11.20 (년.월.일)
Sale Product
Name, Price, Quantity
snack, 1000, 99
Name, Price, Quantity
ice_cream, 1500, 99
Name, Price, Quantity
fruit, 3000, 99
Name, Price, Quantity
water, 500, 99
Name, Price, Quantity
ramen, 800, 99
Name, Price, Quantity
beverage, 1200, 99
Name, Price, Quantity
coffee, 2000, 99
```

20171121_product.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
Date : 2017.11.21 (년.월.일)
Sale Product
Name, Price, Quantity
snack, 1000, 100
Name, Price, Quantity
ice_cream, 1500, 100
Name, Price, Quantity
fruit, 3000, 100
Name, Price, Quantity
water, 500, 100
Name, Price, Quantity
ramen, 800, 100
Name, Price, Quantity
beverage, 1200, 100
Name, Price, Quantity
coffee, 2000, 100
```

20171121_product.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
Date : 2017.11.21 (년.월.일)
Sale Product
Name, Price, Quantity
snack, 1000, 100
Name, Price, Quantity
ice_cream, 1500, 100
Name, Price, Quantity
fruit, 3000, 100
Name, Price, Quantity
water, 500, 100
Name, Price, Quantity
ramen, 800, 100
Name, Price, Quantity
beverage, 1200, 100
Name, Price, Quantity
coffee, 2000, 100
```

<3_001_000>

<3_001_001>

4.4 Test Result (3_002)

20171120_sale_management.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Receipt Num : 2017.11.20.17.21 (년.월.일.시.분)
Date : 2017.11.20 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 2, 2000
Total Sale price : 2000

stock_20171120.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Date : 2017.11.20.18.23 (년.월.일.시.분)
Name, Price, Quantity
snack, 1000, 98
Name, Price, Quantity
ice_cream, 1500, 100
Name, Price, Quantity
fruit, 3000, 100
Name, Price, Quantity
water, 500, 100
Name, Price, Quantity
ramen, 800, 100
Name, Price, Quantity
beverage, 1200, 100
Name, Price, Quantity
coffee, 2000, 100

```
snack quantity : 2 | price : 1000 | sum_price : 2000
Total Sale Money 2000
Receive Money 2000
Send Money 0

Client Screen
Total Sale Money 2000
Receive Money 2000
Send Money 0

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

1. Power On 2. Power Off : 2
POS OFF

호명보 @team5-PC ~/team5
$ ./POST_Main.exe
POST System

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 : 98 | 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 : 100 | price : 1200
coffee quantity : 100 | price : 2000
```

<3_002_000>

4.4 Test Result (3_002)

20171120_sale_management.txt - 메모장	stock_20171120.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)	파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
Receipt Num : 2017.11.20.19.48 (년.월.일) Date : 2017.11.20 (년.월.일) Sale Product Name, Price, Quantity, Sum_price snack, 1000, 1, 1000 Name, Price, Quantity, Sum_price ice_cream, 1500, 1, 1500 Name, Price, Quantity, Sum_price fruit, 3000, 1, 3000 Name, Price, Quantity, Sum_price water, 500, 1, 500 Name, Price, Quantity, Sum_price ramen, 800, 1, 800 Name, Price, Quantity, Sum_price beverage, 1200, 1, 1200 Name, Price, Quantity, Sum_price coffee, 2000, 1, 2000 Total Sale price : 10000	Date : 2017.11.20.23.51 (년.월.일.시.분) Name, Price, Quantity snack, 1000, 99 Name, Price, Quantity ice_cream, 1500, 99 Name, Price, Quantity fruit, 3000, 99 Name, Price, Quantity water, 500, 99 Name, Price, Quantity ramen, 800, 99 Name, Price, Quantity beverage, 1200, 99 Name, Price, Quantity coffee, 2000, 99

```
1. Power On 2. Power Off : 2  
POS OFF
```

```
순정보@-PC ~/team5  
$ ./POST_Main.exe  
POST System
```

```
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 : 99 | price : 1000  
ice_cream quantity : 99 | price : 1500  
fruit quantity : 99 | price : 3000  
water quantity : 99 | price : 500  
ramen quantity : 99 | price : 800  
beverage quantity : 99 | price : 1200  
coffee quantity : 99 | price : 2000
```

```
1. Sale Mode
```

<3_002_001>

4.4 Test Result (3_003~3_004)

```
Sale Mode
Barcode Input
-> Barcode : 001

Casher Screen
  snack    quantity :    1 | price :   1000 | sum_price :   1000
Total Sale Money                1000
Receive Money                    0
Send Money                       0

Client Screen
Total Sale Money                1000
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)
5
Can not check Stock
```

```
1. Sale Mode
2. Quantity Changed Mode
3. Money Received Mode
4. Refund Mode
5. Stock Check Mode
(-1 : Power Mode)
5
Refund Mode
Refund Receipt Barcode : 201711210000

Casher Screen
Date                                2017.11.21 05:04
  snack    quantity :   1000 | price :    1 | sum_price :   1000
Total Refund Money                1000

Client Screen
Date                                2017.11.21 05:04
Total Refund Money                1000
```

<3_003_000>

```
Date : 2017.11.20.18.23 (년.월.일.시.분)
Name, Price, Quantity
snack, 1000, 98
Name, Price, Quantity
ice_cream, 1500, 100
Name, Price, Quantity
fruit, 3000, 100
Name, Price, Quantity
water, 500, 100
Name, Price, Quantity
ramen, 800, 100
Name, Price, Quantity
beverage, 1200, 100
Name, Price, Quantity
coffee, 2000, 100
```

<3_003_001>

<3_004_000>

4.5 Test Result (4_000~4_004)

Id	Result
4_000_000	Passed
4_001_000	Passed
4_001_001	Passed
4_001_002	Passed
4_002_000	Passed
4_003_000	Passed
4_003_001	Passed
4_003_002	Passed
4_003_003	Passed
4_004_000	Passed
4_004_001	Passed

4.5 Test Result (4_000)

```
Casher Screen
  snack  quantity :    1 | price :   1000 | sum_price :   1000
Total Sale Money
Receive Money
Send Money

Client Screen
Total Sale Money
Receive Money
Send Money

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

1. Power On 2. Power Off : 2
POS OFF

명 정보 @-----PC ~/team5
$ ./POST_Main.exe
POST System

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 :    99 | price :   1000
  ice_cream quantity :   100 | price :   1500
  fruit   quantity :   100 | price :   3000
  water   quantity :   100 | price :    500
  ramen   quantity :   100 | price :    800
```

sale_201711201748.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
Receipt Num : 2017.11.20.17.48 (년.월.일.시.분)
Date : 2017.11.20 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Sale Price : 1000
```

stock_20171120.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
Date : 2017.11.20.18.12 (년.월.일.시.분)
Name, Price, Quantity
snack, 1000, 99
Name, Price, Quantity
ice_cream, 1500, 100
Name, Price, Quantity
fruit, 3000, 100
Name, Price, Quantity
water, 500, 100
Name, Price, Quantity
ramen, 800, 100
Name, Price, Quantity
beverage, 1200, 100
Name, Price, Quantity
coffee, 2000, 100
|
```

20171120_product.txt	2017-11-20 오후 4:53
20171120_sale_management.txt	2017-11-20 오후 4:53
sale_201711201748.txt	2017-11-20 오후 4:53
sale_receipt_list.txt	2017-11-20 오후 4:53
save_data.txt	2017-11-20 오후 4:54
stock_20171120.txt	2017-11-20 오후 4:53


<4_000_000>

4.5 Test Result (4_001)

 20171120_product.txt	2017-11-20 오후 5:00
 settle_20171121.txt	2017-11-20 오후 5:01
 settle_20171122.txt	2017-11-20 오후 5:04
 settle_20171123.txt	2017-11-20 오후 5:07

 settle_20171121.txt

2017-11-20 오후 5:01

 settle_20171121.txt - 메모장

<4_001_000>

파일(F) 편집(E) 서식(O) 보기(V)

Date : 2017.11.21 (년.월.일)

Sale Product


Total Sale Price : 0

Refund Product

Total Refund Price : 0

 settle_20171122.txt

2017-11-20 오후 5:04

 settle_20171122.txt - 메모장

<4_001_001>

파일(F) 편집(E) 서식(O) 보기(V)

Date : 2017.11.22 (년.월.일)

Sale Product


Total Sale Price : 0

Refund Product

Total Refund Price : 0

 settle_20171123.txt

2017-11-20 오후 5:07

 settle_20171123.txt - 메모장

<4_001_002>

파일(F) 편집(E) 서식(O) 보기(V)

Date : 2017.11.23 (년.월.일)

Sale Product

Total Sale Price : 0

Refund Product

Total Refund Price : 0

4.5 Test Result (4_002)

20171120_product.txt 2017-11-20 오후 5:12
20171120_sale_management.txt 2017-11-20 오후 5:12
sale_201711201750.txt 2017-11-20 오후 5:12
sale_receipt_list.txt 2017-11-20 오후 5:12

sale_201711201750.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Receipt Num : 2017.11.20.17.50 (년.월.일.시.분)
Date : 2017.11.20 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Sale Price : 1000

20171120_product.txt 2017-11-20 오후 5:12
20171120_sale_management.txt 2017-11-20 오후 5:12
20171122_product.txt 2017-11-20 오후 5:16
sale_201711201750.txt 2017-11-20 오후 5:12
sale_receipt_list.txt 2017-11-20 오후 5:12
save_data.txt 2017-11-20 오후 5:16
settle_20171121.txt 2017-11-20 오후 5:13
settle_20171122.txt 2017-11-20 오후 5:16

settle_20171121.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Date : 2017.11.21 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Sale Price : 1000

Refund Product
Total Refund Price : 0

settle_20171122.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Date : 2017.11.22 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Sale Price : 1000

Refund Product
Total Refund Price : 0

<4_002_000>

4.5 Test Result (4_003)

20171120_product.txt

2017-11-20 오후 5:22

```
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 :
```

<4_003_000>

```
Casher Screen
  snack  quantity :    1 | price :   1000 | sum_price :   1000
Total Sale Money                1000
Receive Money                    1000
Send Money                        0

Client Screen
Total Sale Money                1000
Receive Money                    1000
Send Money                        0
```

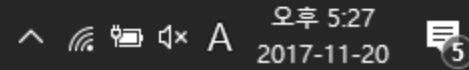
<4_003_001>

POST_Main.exe

2017-11-20 오전 10:53

20171120_product.txt

2017-11-20 오후 5:22



20171120_product.txt	2017-11-20 오후 5:22
20171122_product.txt	2017-11-20 오후 5:28
20171122_sale_management.txt	2017-11-20 오후 5:28
sale_201711221638.txt	2017-11-20 오후 5:28
sale_receipt_list.txt	2017-11-20 오후 5:28
settle_20171122.txt	2017-11-20 오후 5:28

4.5 Test Result (4_003)

📄 settle_20171122.txt

2017-11-20 오후 5:28

📄 settle_20171123.txt

2017-11-20 오후 5:29

📄 settle_20171122.txt

2017-11-20 오후 5:28

📄 settle_20171123.txt

2017-11-20 오후 5:29

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



<4_003_002>

```
Casher Screen
Date                2017.11.24 21:09
snack  quantity : 1000 | price : 1 | sum_price : 1000
Total Refund Money 1000

Client Screen
Date                2017.11.24 21:09
Total Refund Money 1000
```

📄 refund_201711242109.txt

2017-11-20 오후 5:35

텍스트 문서

📄 settle_20171122.txt

2017-11-20 오후 5:28

텍스트 문서

📄 settle_20171123.txt

2017-11-20 오후 5:29

텍스트 문서

📄 settle_20171124.txt


2017-11-20 오후 5:35

텍스트 문서

<4_003_003>

4.5 Test Result (4_004)

<4_004_000>

 settle_20171121.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Date : 2017.11.21 (년.월.일)

Sale Product

Name, Price, Quantity, Sum_price


snack, 1000, 1, 1000

Total Sale Price : 1000

Refund Product

Total Refund Price : 0

<4_004_001>

 settle_20171124.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Date : 2017.11.24 (년.월.일)

Sale Product

Total Sale Price : 0

Refund Product

Name, Price, Quantity, Sum_price

snack, 1000, 1, 1000

Total Refund Price : 0

5. Team 6 Result

5.1 Test Result(0)

5.2 Test Result(1)

5.3 Test Result(2)

5.4 Test Result(3)

5.5 Test Result(4)

5.1 Test Result (0_000)

Id	Result
0_000_000	Passed
0_000_001	Passed
0_000_002	Passed

```
Touch Screen
amount : 1000
money : 1000
change : 0

product name | price | count
snack        | 1000 | 1

SALE[0] REFUND[1] STOCK[2] TERMINATE[3]
command: 3
[POS] terminate...
```

<0_000_000>

```
Touch Screen
receipt code : 201711202239
date : 2017.11.20
product list - refund
refund : 1000

product name | price | count
snack        | 1000 | 1

SALE[0] REFUND[1] STOCK[2] TERMINATE[3]
command: 3
[POS] terminate...
```

<0_000_001>

```
Touch Screen
date : 2017.11.20.18.25

product name | price | volume
snack        | 1000 | 100
ice-cream    | 1500 | 100
fruit        | 3000 | 100
water        | 500  | 100
ramen        | 800  | 100
drink        | 1200 | 100
coffee      | 2000 | 100

SALE[0] REFUND[1] STOCK[2] TERMINATE[3]
command: 3
[POS] terminate...
```

<0_000_002>

5.2 Test Result (1_000)

Id	Result
1_000_000	Passed
1_000_001	Passed
1_000_002	Passed
1_000_003	Passed
1_000_004	Passed
1_000_005	Passed
1_000_006	Passed
1_000_007	Passed
1_000_008	Passed

5.2 Test Result (1_000)

```
Touch Screen
amount : 1000

product name | price | count
snack        | 1000 | 1

[ADD[+]      MINUS[-]      BARCODE[code]
command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_000>

```
Touch Screen
amount : 2500

product name | price | count
snack        | 1000 | 1
ice-cream    | 1500 | 1

[ADD[+]      MINUS[-]      BARCODE[code]
command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_001>

```
Touch Screen
amount : 5500

product name | price | count
snack        | 1000 | 1
ice-cream    | 1500 | 1
fruit        | 3000 | 1

[ADD[+]      MINUS[-]      BARCODE[code]
command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_002>

```
Touch Screen
amount : 6000

product name | price | count
snack        | 1000 | 1
ice-cream    | 1500 | 1
fruit        | 3000 | 1
water        | 500  | 1

[ADD[+]      MINUS[-]      BARCODE[code]
command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_003>

```
Touch Screen
amount : 6800

product name | price | count
snack        | 1000 | 1
ice-cream    | 1500 | 1
fruit        | 3000 | 1
water        | 500  | 1
ramen        | 800  | 1

[ADD[+]      MINUS[-]      BARCODE[code]
command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_004>

```
Touch Screen
amount : 8000

product name | price | count
snack        | 1000 | 1
ice-cream    | 1500 | 1
fruit        | 3000 | 1
water        | 500  | 1
ramen        | 800  | 1
drink        | 1200 | 1

[ADD[+]      MINUS[-]      BARCODE[code]
command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_005>

5.2 Test Result (1_000)

```
Touch Screen
amount : 10000

product name | price | count
snack        | 1000 | 1
ice-cream    | 1500 | 1
fruit        | 3000 | 1
water        | 500  | 1
ramen        | 800  | 1
drink        | 1200 | 1
coffee      | 2000 | 1

ADD[+]      MINUS[-]      BARCODE[code]

command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_006>

```
Touch Screen
amount : 10000

product name | price | count
snack        | 1000 | 1
ice-cream    | 1500 | 1
fruit        | 3000 | 1
water        | 500  | 1
ramen        | 800  | 1
drink        | 1200 | 1
coffee      | 2000 | 1

ADD[+]      MINUS[-]      BARCODE[code]

command:
[ERROR] Invalid value
**help** : if you stop this action, enter 'quit'
```

<1_000_007>

```
Touch Screen
amount : 2000

product name | price | count
snack        | 1000 | 2

ADD[+]      MINUS[-]      BARCODE[code]

command: |
**help** : if you stop this action, enter 'quit'
```

<1_000_008>

5.2 Test Result (1_001)

Id	Result
1_001_000	Passed
1_001_001	Passed
1_001_002	Passed
1_001_003	Passed

5.2 Test Result (1_001)

Touch Screen			Customer Screen		
amount : 5000			amount : 5000		
product name	price	count	product name	price	count
snack	1000	5	snack	1000	5
ADD[+]	MINUS[-]	BARCODE[code]			
command:					
help : if you stop this action, enter 'quit'					

<1_001_000>

Touch Screen			Customer Screen		
amount : 1000			amount : 1000		
product name	price	count	product name	price	count
snack	1000	1	snack	1000	1
ADD[+]	MINUS[-]	BARCODE[code]			
command:					
help : if you stop this action, enter 'quit'					

<1_001_001>

5.2 Test Result (1_001)

Touch Screen			Customer Screen		
amount : 0			amount : 0		
product name		price		count	
ADD[+]	MINUS[-]	BARCODE[code]			
command:					
help : if you stop this action, enter 'quit'					

<1_001_003>

Touch Screen			Customer Screen		
amount : 0			amount : 0		
product name		price		count	
ADD[+]	MINUS[-]	BARCODE[code]			
command:					
[ERROR] Invalid value					
help : if you stop this action, enter 'quit'					

<1_001_004>

5.2 Test Result (1_002~1_003)

Id	Result
1_002_000	Passed
1_003_000	Passed
1_003_001	Passed



<1_002_000>

5.2 Test Result (1_002~1_003)

```
Touch Screen
amount : 1000

product name | price | count
snack       | 1000 | 1

ADD[+]      MINUS[-]      BARCODE[code]

money :
[ERROR] : Insufficient amount
```

<1_003_000>

```
Touch Screen
amount : 1000
money : 1000
change : 0

product name | price | count
snack       | 1000 | 1

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command:
```

<1_003_001>

5.2 Test Result (1_004~1_010)

Id	Result
1_004_000	Passed
1_005_000	Passed
1_006_000	Passed
1_007_000	Passed
1_008_000	Passed
1_009_000	Passed
1_010_000	Passed

```
Touch Screen
amount : 1000
money : 1000
change : 0
product name | price | count
snack       | 1000 | 1
SALE[0]     REFUND[1] STOCK[2] TERMINATE[3]
command:
```

```
Customer Screen
amount : 1000
money : 1000
change : 0
product name | price | count
snack       | 1000 | 1
```

<1_004_000, 1_008_000, 1_009_000>

5.2 Test Result (1_000)

sale_201711281131.txt

2017-11-20 오후 6:26

<1_005_000>

20171120_product.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

[product]	[price]	[code]	[volume]
snack	1000	001	99
ice-cream	1500	010	100
fruit	3000	011	100
water	500	100	100
ramen	800	101	100
drink	1200	110	100
coffee	2000	111	100

total sales : 1000
total refund : 0

<1_006_000>

20171128_sale_management.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
[sale record]
|| code || amount || money || change ||
| 201711281131 | 1000 | 1000 | 0 |
[list] snack(1) ice-cream(0) fruit(0) water(0) ramen(0) drink(0) coffee(0)
[refund record]
```

<1_007_000>

sale_201711281131.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
receipt code : 2017.11.28.11.31
date : 2017.11.28
sale product list
[prduct name] [price] [count]
snack          1000    1
sale amount : 1000
```

<1_010_000>

5.3 Test Result (2_000~2_001)

Id	Result
2_000_000	Fail (특정 경우에서 실패)
2_000_001	Fail (영수증을 취소했음에도 불구하고 Invalid value로 뜬다)
2_000_002	Passed
2_000_003	Passed
2_000_004	Passed
2_001_000	Passed

5.3 Test Result (2_000~2_001)

```
Touch Screen
amount : 1000
money : 1000
change : 0

product name | price | count
snack        | 1000 | 1

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

[POS] Are you sure that you wanna this?(y, n) : n
Segmentation fault (core dumped)
$ 정보 @ ~-PC ~/team6
```

```
Touch Screen
amount : 3000
money : 4000
change : 1000

product name | price | count
fruit        | 3000 | 1

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command:
[POS] cancel success!
```

<2_000_000>

```
Touch Screen
amount : 1000
money : 1000
change : 0

product name | price | count
snack        | 1000 | 1

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

[POS] Are you sure that you wanna this?(y, n) : n
Segmentation fault (core dumped)
$ 정보 @ ~-PC ~/team6
```

5.3 Test Result (2_000~2_001)

```
Touch Screen
amount : 1000
money : 1000
change : 0

product name | price | count
snack       | 1000 | 1

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command: |
[ERROR] Invalid value
```

<2_000_001>

```
Touch Screen
receipt code : 201711210555
date : 2017.11.21
product list - refund
refund : 1000

product name | price | count
snack       | 1000 | 1

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command: |
[ERROR] Invalid value
```

<2_000_002>

5.3 Test Result (2_000~2_001)

Touch Screen				Customer Screen			
receipt code : 201711222043				receipt code : 201711222043			
date : 2017.11.22				date : 2017.11.22			
product list - refund				product list - refund			
refund : 10000				refund : 10000			
product name	price	count		product name	price	count	
snack	1000	1		snack	1000	1	
ice-cream	1500	1		ice-cream	1500	1	
fruit	3000	1		fruit	3000	1	
water	500	1		water	500	1	
ramen	800	1		ramen	800	1	
drink	1200	1		drink	1200	1	
coffee	2000	1		coffee	2000	1	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]				
command:							

<2_000_003>

Touch Screen			
receipt code : 201711222043			
date : 2017.11.22			
product list - refund			
refund : 10000			
product name	price	count	
snack	1000	1	
ice-cream	1500	1	
fruit	3000	1	
water	500	1	
ramen	800	1	
drink	1200	1	
coffee	2000	1	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			
[ERROR] Invalid value			

<2_000_004>

Touch Screen			
amount : 1000			
money : 1000			
change : 0			
product name	price	count	
snack	1000	1	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			
[ERROR] Invalid value			

<2_000_005>

5.3 Test Result (2_002~2_006)

Id	Result
2_002_000	Passed
2_002_001	Passed
2_003_000	Fail (001만 판매하고 환불했을 경우 total refund가 반영안됨)
2_003_001	Passed
2_004_000	Passed
2_005_000	Passed
2_006_000	Passed

5.3 Test Result (2_002~2_006)

refund_201711210141.txt

2017-11-20 오후 7:33

<2_002_000>

refund_201711221157.txt

2017-11-20 오후 7:38

<2_002_001>

20171121_product.txt - 메모장

파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)
[product]		[price]	[code]	[volume]
snack		1000	001	100
ice-cream	1500		010	100
fruit		3000	011	100
water		500	100	100
ramen	800		101	100
drink		1200	110	100
coffee		2000	111	100
total sales : 0				
total refund : 0				

<2_003_000>

20171122_product.txt - 메모장

파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)
[product]		[price]	[code]	[volume]
snack		1000	001	101
ice-cream	1500		010	101
fruit		3000	011	101
water		500	100	101
ramen	800		101	101
drink		1200	110	101
coffee		2000	111	101
total sales : 0				
total refund : 10000				

<2_003_001>

5.3 Test Result (2_002~2_006)

Touch Screen				Customer Screen			
receipt code : 201711222043				receipt code : 201711222043			
date : 2017.11.22				date : 2017.11.22			
product list - refund				product list - refund			
refund : 10000				refund : 10000			
product name	price	count		product name	price	count	
snack	1000	1		snack	1000	1	
ice-cream	1500	1		ice-cream	1500	1	
fruit	3000	1		fruit	3000	1	
water	500	1		water	500	1	
ramen	800	1		ramen	800	1	
drink	1200	1		drink	1200	1	
coffee	2000	1		coffee	2000	1	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]				
command:							

<2_004_000, 2_005_000>

 refund_201711210141.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
receipt code : 2017.11.20.21.17
date : 2017.11.21
refund product list
[prduct name] [price] [count]
snack          1000    1
refund amount : 1000
```

<2_006_000>

5.4 Test Result (3_000~3_003)

Id	Result
3_000_000	Fail (상품 파일, 판매 목록 파일이 날짜가 지나도 새로 생성안됨)
3_000_001	Fail("")
3_001_000	Passed
3_001_001	Passed
3_002_000	Fail (동기화가 안됨)
3_002_001	Fail (동기화가 안됨)
3_003_000	Passed
3_003_001	Passed
3_004_000	Passed

5.4 Test Result (3_000)

```

Touch Screen
date : 2017.11.20.22.16

product name | price | volume
snack       | 1000 | 98
ice-cream   | 1500 | 100
fruit       | 3000 | 100
water       | 500  | 100
ramen       | 800  | 100
drink       | 1200 | 100
coffee     | 2000 | 100

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command:
    
```

```

20171120_product.txt      2017-11-20 오후 7:44
20171120_sale_management.txt 2017-11-20 오후 7:44
stock_20171120.txt       2017-11-20 오후 7:44
sale_201711202200.txt    2017-11-20 오후 7:44
-----
20171120_product.txt      2017-11-20 오후 7:44
20171120_sale_management.txt 2017-11-20 오후 7:44
stock_20171120.txt       2017-11-20 오후 7:44
sale_201711202200.txt    2017-11-20 오후 7:44
posSystem.exe             2017-11-15 오전 11:22
    
```

<3_000_000>



```

Touch Screen
date : 2017.11.21.05.33

product name | price | volume
snack       | 1000 | 99
ice-cream   | 1500 | 99
fruit       | 3000 | 99
water       | 500  | 99
ramen       | 800  | 99
drink       | 1200 | 99
coffee     | 2000 | 99

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command:
    
```

```

20171121_product.txt      2017-11-20 오후 7:58
20171121_sale_management.txt 2017-11-20 오후 7:58
sale_201711210525.txt    2017-11-20 오후 7:58
stock_20171121.txt       2017-11-20 오후 7:58
-----
20171121_product.txt      2017-11-20 오후 7:58
20171121_sale_management.txt 2017-11-20 오후 7:58
stock_20171121.txt       2017-11-20 오후 7:58
sale_201711210525.txt    2017-11-20 오후 7:58
sale_201711210221.txt    2017-11-20 오후 7:58
settle_20171121.txt      2017-11-20 오후 7:58
    
```

<3_000_001>



5.4 Test Result (3_001)

Touch Screen			
date : 2017.11.20.22.16			
product name	price	volume	
snack	1000	98	
ice-cream	1500	100	
fruit	3000	100	
water	500	100	
ramen	800	100	
drink	1200	100	
coffee	2000	100	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

Touch Screen			
date : 2017.11.23.06.56			
product name	price	volume	
snack	1000	100	
ice-cream	1500	100	
fruit	3000	100	
water	500	100	
ramen	800	100	
drink	1200	100	
coffee	2000	100	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

<3_001_000>

Touch Screen			
date : 2017.11.21.05.33			
product name	price	volume	
snack	1000	99	
ice-cream	1500	99	
fruit	3000	99	
water	500	99	
ramen	800	99	
drink	1200	99	
coffee	2000	99	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

Touch Screen			
date : 2017.11.23.18.45			
product name	price	volume	
snack	1000	100	
ice-cream	1500	100	
fruit	3000	100	
water	500	100	
ramen	800	100	
drink	1200	100	
coffee	2000	100	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

<3_001_001>

5.4 Test Result (3_002)

Touch Screen			
date : 2017.11.24.02.40			
product name	price	volume	
snack	1000	98	
ice-cream	1500	100	
fruit	3000	100	
water	500	100	
ramen	800	100	
drink	1200	100	
coffee	2000	100	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

<3_002_000>

Touch Screen			
date : 2017.11.20.22.48			
product name	price	volume	
snack	1000	99	
ice-cream	1500	99	
fruit	3000	99	
water	500	99	
ramen	800	99	
drink	1200	99	
coffee	2000	99	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

<3_002_001>

Touch Screen			
date : 2017.11.20.20.10			
product name	price	volume	
snack	1000	100	
ice-cream	1500	100	
fruit	3000	100	
water	500	100	
ramen	800	100	
drink	1200	100	
coffee	2000	100	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

Touch Screen			
date : 2017.11.20.20.17			
product name	price	volume	
snack	1000	100	
ice-cream	1500	100	
fruit	3000	100	
water	500	100	
ramen	800	100	
drink	1200	100	
coffee	2000	100	
SALE[0]	REFUND[1]	STOCK[2]	TERMINATE[3]
command:			

5.4 Test Result (3_003~3_004)

```
ADD[+]          MINUS[-]          BARCODE[code]
command: |
[ERROR] Invalid value
**help** : if you stop this action, enter 'quit'
```

<3_003_000>

```
SALE[0]          REFUND[1]          STOCK[2]          TERMINATE[3]
receipt code :
[ERROR] Invalid value
```

<3_003_001>

stock_20171121.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
date : 2017.11.21.11.13
[product]          [price]          [code]          [volume]
snack              1000            001            100
ice-cream          1500            010            100
fruit              3000            011            100
water              500             100            100
ramen              800             101            100
drink              1200            110            100
coffee            2000            111            100
```

<3_004_000>

5.5 Test Result (4_000~4_004)

Id	Result
4_000_000	Fail (동기화가 안됨)
4_001_000	Fail (자동으로 생성되지 않으며 판매 후 재고 확인을 해야 생성됨)
4_001_001	Fail ("
4_001_002	Fail ("
4_002_000	Fail (정산 파일이 자동으로 생성되지 않음)
4_003_000	Passed
4_003_001	Passed
4_003_002	Passed
4_003_003	Passed
4_004_000	Fail (형식 맞지 않음)
4_004_001	Fail ("

5.5 Test Result (4_000)

```
Touch Screen
date : 2017.11.23.14.49

product name | price | volume
snack        | 1000 | 100
ice-cream    | 1500 | 100
fruit        | 3000 | 100
water        | 500  | 100
ramen        | 800  | 100
drink        | 1200 | 100
coffee      | 2000 | 100

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command: |
[ERROR] Invalid value
```

```
Touch Screen
date : 2017.11.20.20.34

product name | price | volume
snack        | 1000 | 100
ice-cream    | 1500 | 100
fruit        | 3000 | 100
water        | 500  | 100
ramen        | 800  | 100
drink        | 1200 | 100
coffee      | 2000 | 100

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command: |
```

<4_000_000>

5.5 Test Result (4_001)

```
Touch Screen
date : 2017.11.20.20.37

product name | price | volume
snack        | 1000 | 100
ice-cream    | 1500 | 100
fruit        | 3000 | 100
water        | 500  | 100
ramen        | 800  | 100
drink        | 1200 | 100
coffee      | 2000 | 100

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command:
```

20171120_product.txt	2017-11-20 오후 8:21
20171120_sale_management.txt	2017-11-20 오후 8:21
stock_20171120.txt	2017-11-20 오후 8:21
20171120_product.txt	2017-11-20 오후 8:21
20171120_sale_management.txt	2017-11-20 오후 8:21
stock_20171120.txt	2017-11-20 오후 8:21
posSystem.exe	2017-11-15 오전 11:22

<4_001_000>

20171120_product.txt	2017-11-20 오후 8:21
20171120_sale_management.txt	2017-11-20 오후 8:21
stock_20171120.txt	2017-11-20 오후 8:21
posSystem.exe	2017-11-15 오전 11:22

20171120_product.txt	2017-11-20 오후 8:21
20171120_sale_management.txt	2017-11-20 오후 8:21
stock_20171120.txt	2017-11-20 오후 8:21
posSystem.exe	2017-11-15 오전 11:22

<4_001_001>

<4_001_002>

5.5 Test Result (4_002)

```
Touch Screen
date : 2017.11.20.21.58

product name | price | volume
snack       | 1000 | 99
ice-cream   | 1500 | 100
fruit       | 3000 | 100
water       | 500  | 100
ramen       | 800  | 100
drink       | 1200 | 100
coffee     | 2000 | 100

SALE[0]      REFUND[1]      STOCK[2]      TERMINATE[3]

command:
```

20171120_product.txt	2017-11-20 오후 8:38
20171120_sale_management.txt	2017-11-20 오후 8:38
stock_20171120.txt	2017-11-20 오후 8:38
sale_201711202150.txt	2017-11-20 오후 8:38
20171120_product.txt	2017-11-20 오후 8:38
20171120_sale_management.txt	2017-11-20 오후 8:38
stock_20171120.txt	2017-11-20 오후 8:38
sale_201711202150.txt	2017-11-20 오후 8:38
posSystem.exe	2017-11-15 오전 11:22

System tray: ^, Wi-Fi, Battery, Mute, Speaker, 오후 8:46, 2017-11-20

<4_002_000>

5.5 Test Result (4_003)



20171120_product.txt	2017-11-20 오후 8:50
20171120_sale_management.txt	2017-11-20 오후 8:50
posSystem.exe	2017-11-15 오전 11:22

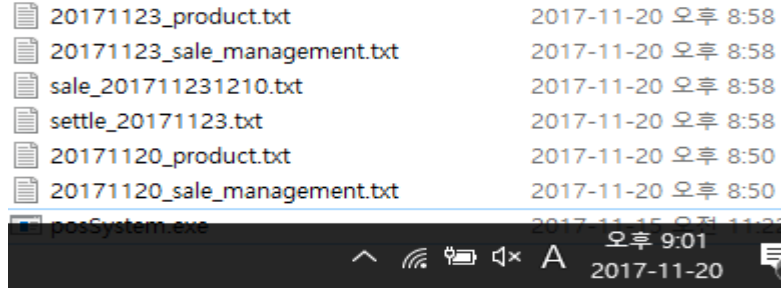
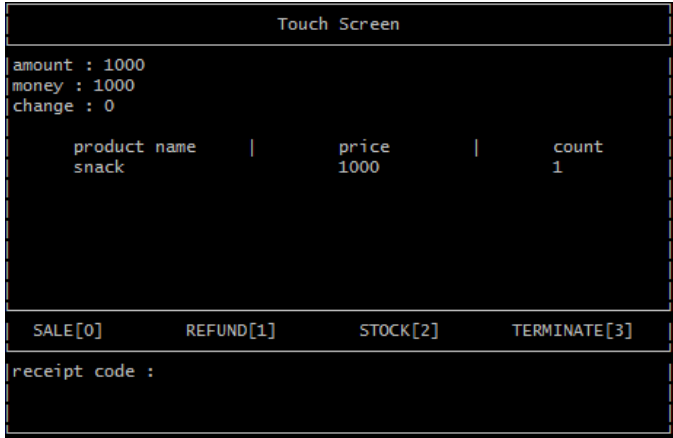


<4_003_000>

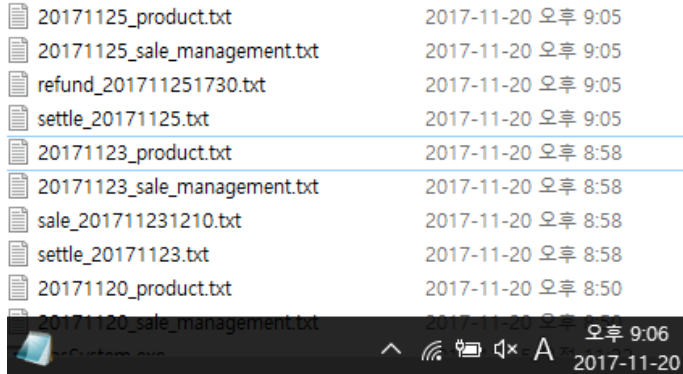
20171123_product.txt	2017-11-20 오후 8:58
20171123_sale_management.txt	2017-11-20 오후 8:58
sale_201711231210.txt	2017-11-20 오후 8:58
settle_20171123.txt	2017-11-20 오후 8:58
20171120_product.txt	2017-11-20 오후 8:50
20171120_sale_management.txt	2017-11-20 오후 8:50

<4_003_001>

5.5 Test Result (4_003)



<4_003_002>



<4_003_003>

5.5 Test Result (4_004)

 settle_20171123.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
date : 2017.11.23.12.10
sale list
s_code : 201711231210
total sales : 1000, count : 1

refund list
total refund : 0, count : 0
```

<4_004_000>

 settle_20171125.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
date : 2017.11.25.17.46
sale list
total sales : 0, count : 0

refund list
r_code : 201711251730
total refund : 1000, count : 1
```

<4_004_001>

Q & A

