Unit Test For POS system

Team_5

201211355 손지웅
201611303 조정익
201610401 손하영

CONTENTS >>

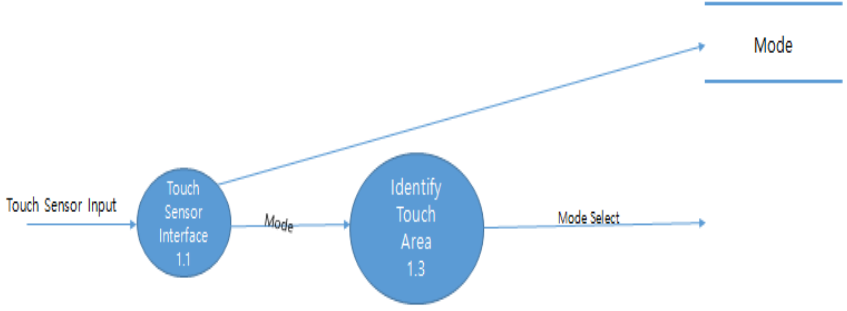
CHAPTER 01

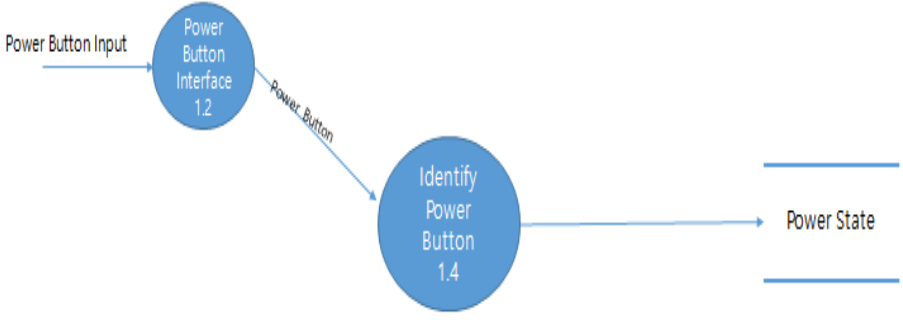
Revision

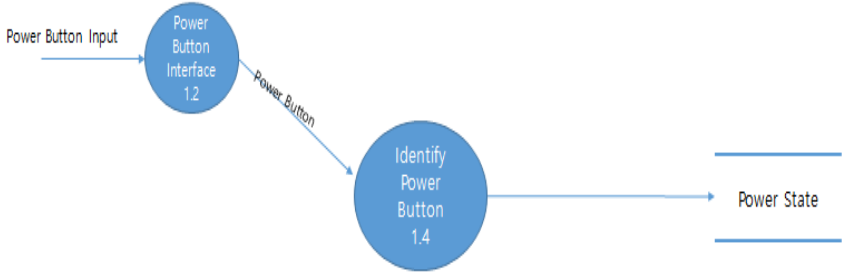
1-1. Revised SDA

1-2. Revised SDS

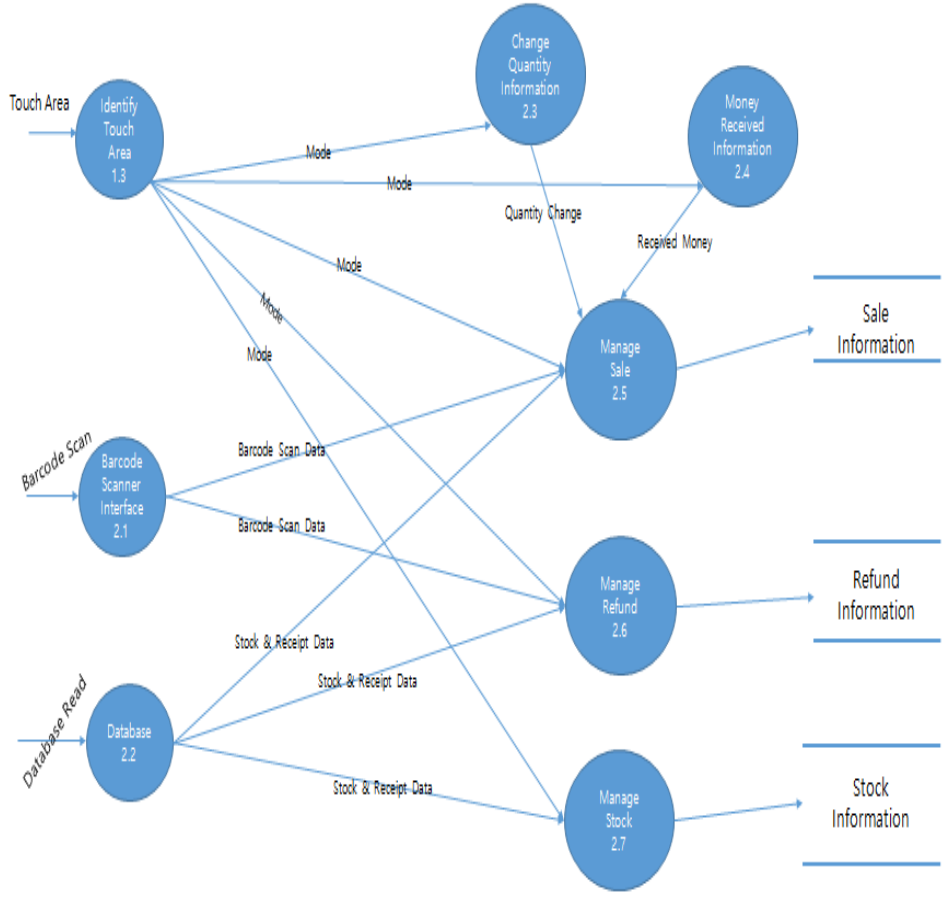
1-1. Revised SDA

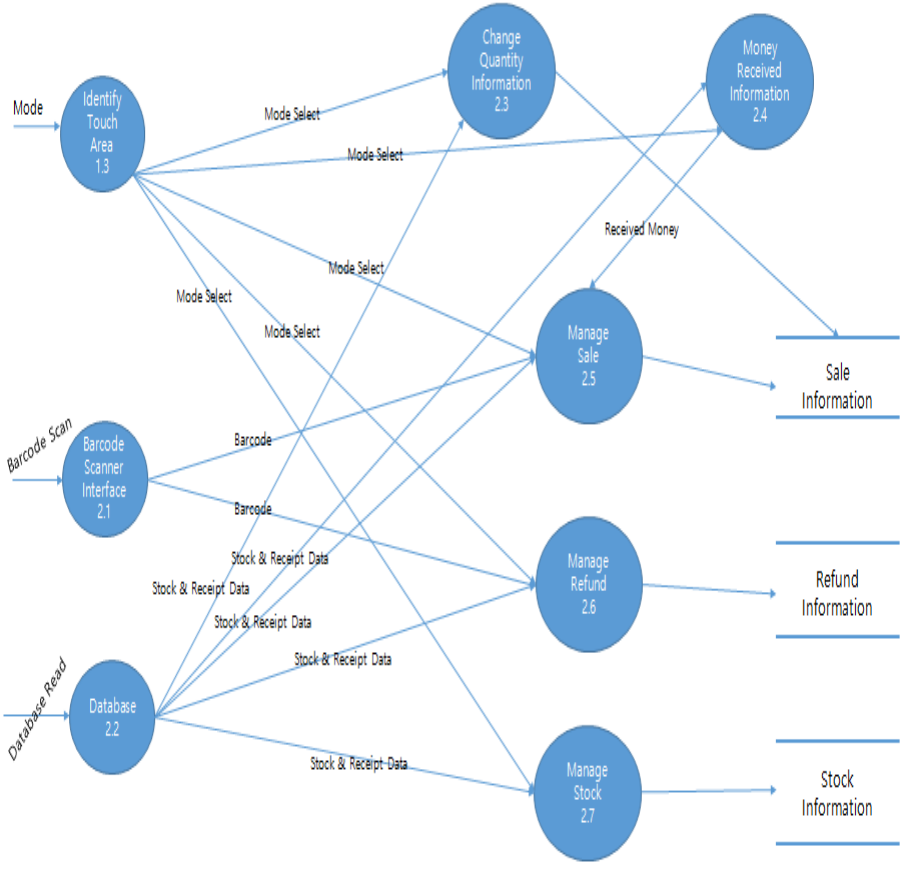




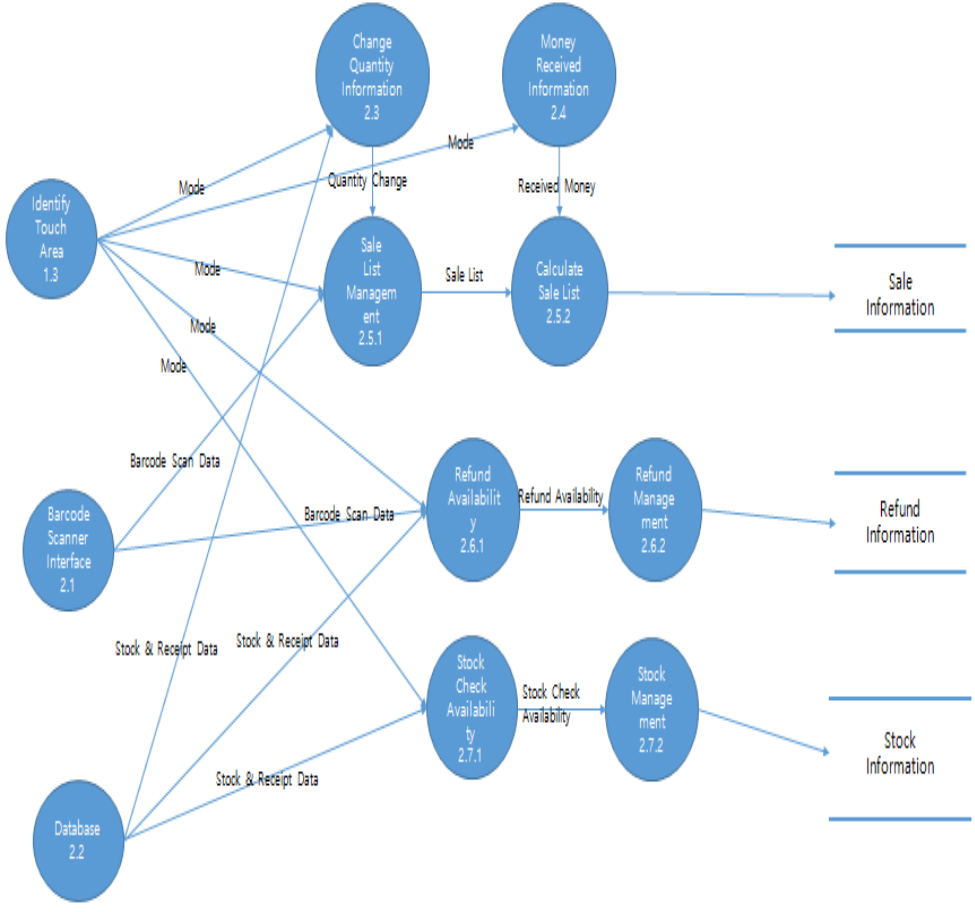


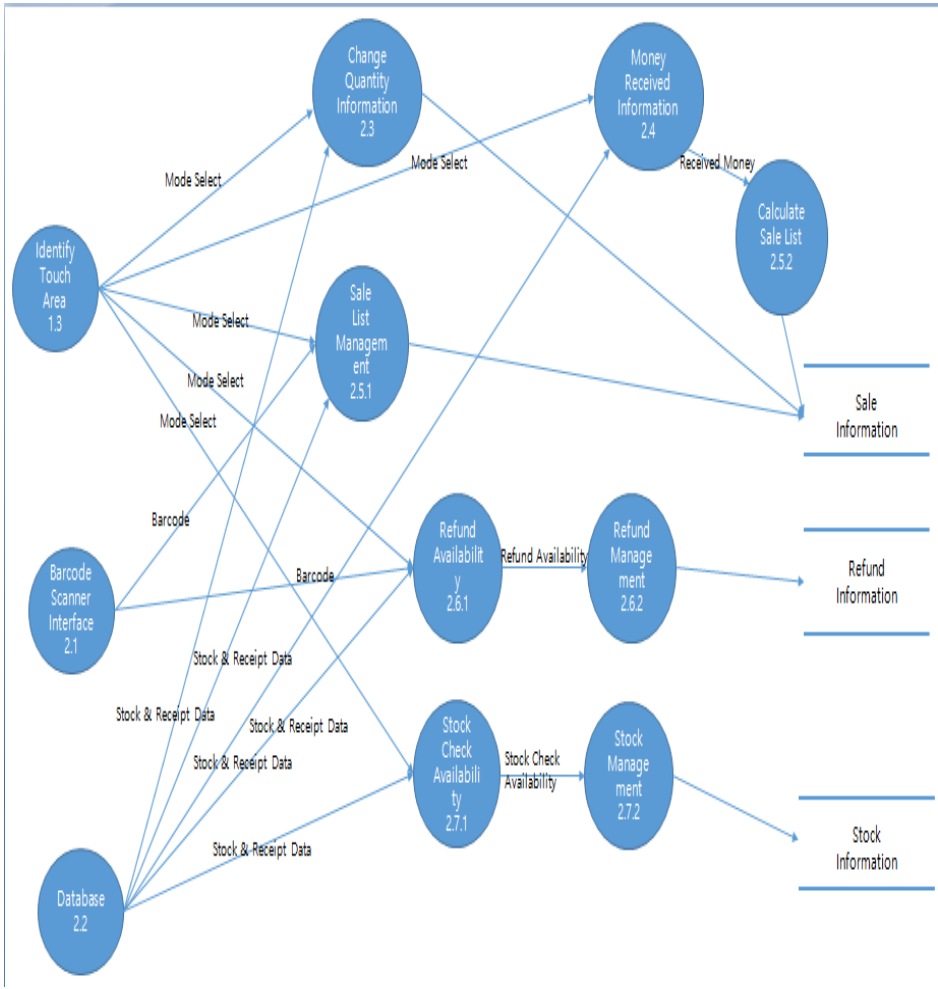
1-1. Revised SDA



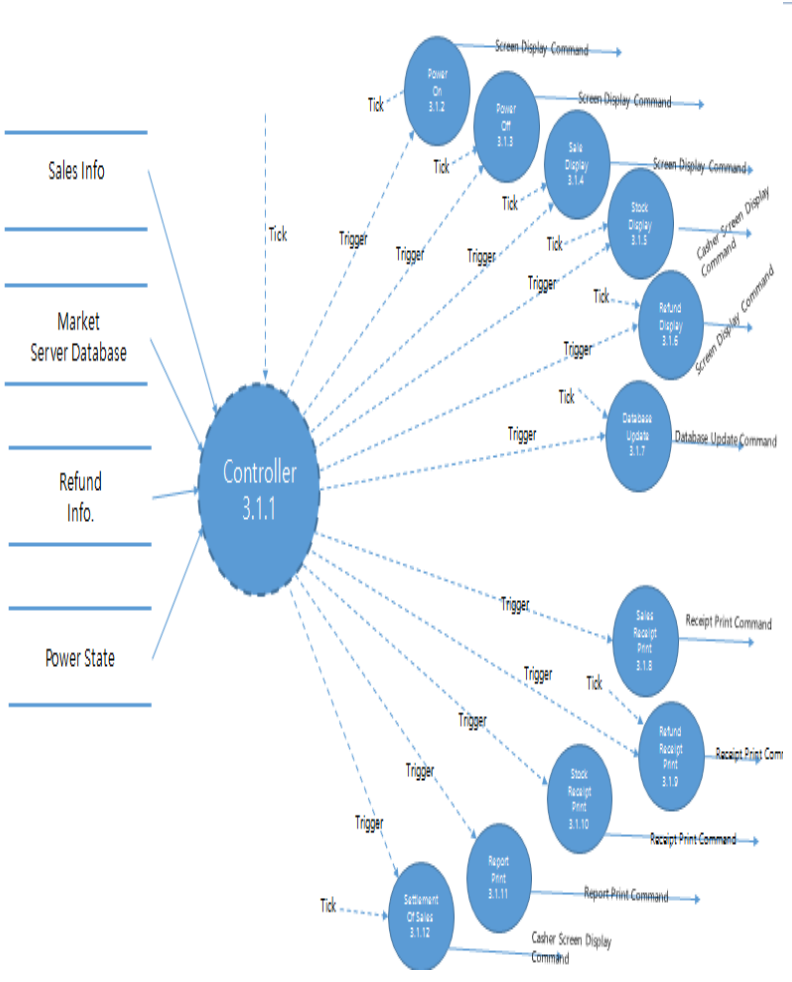


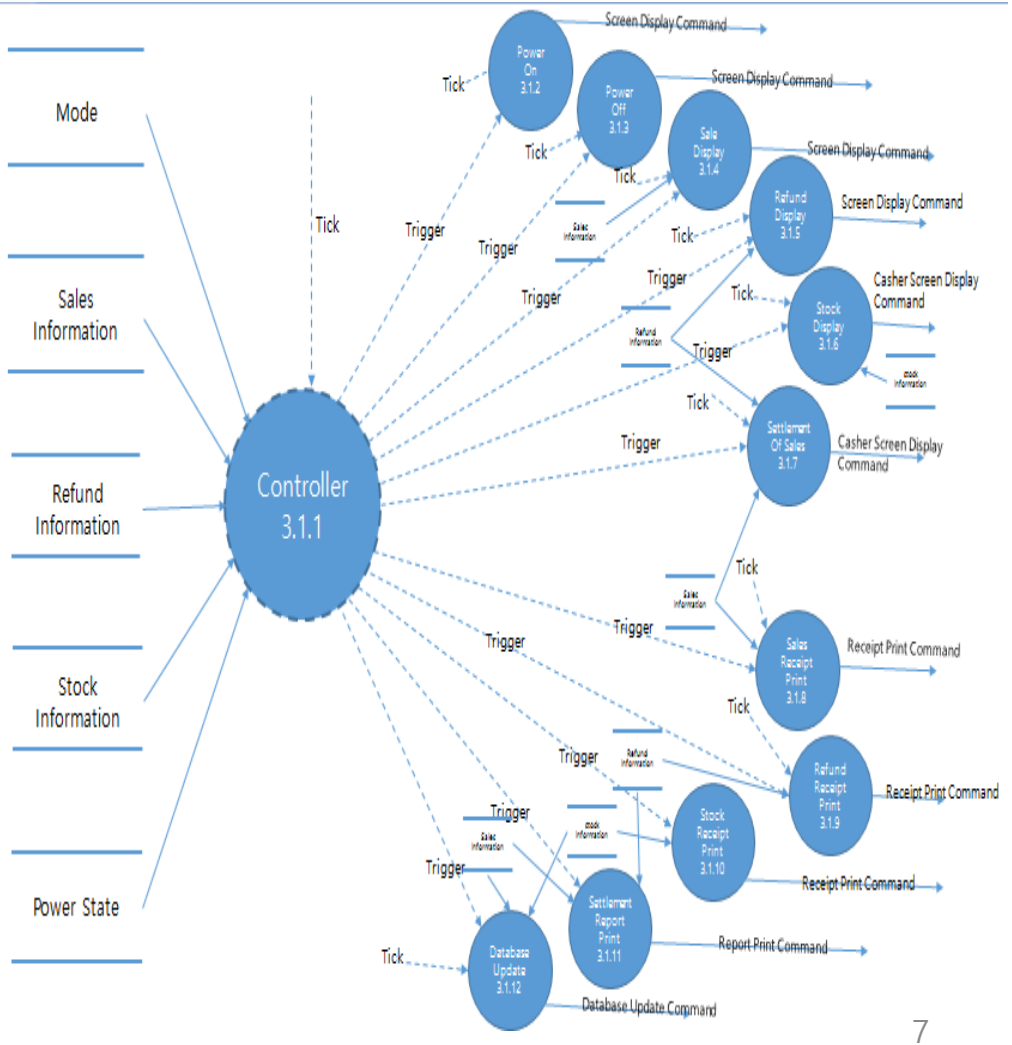
1-1. Revised SDA



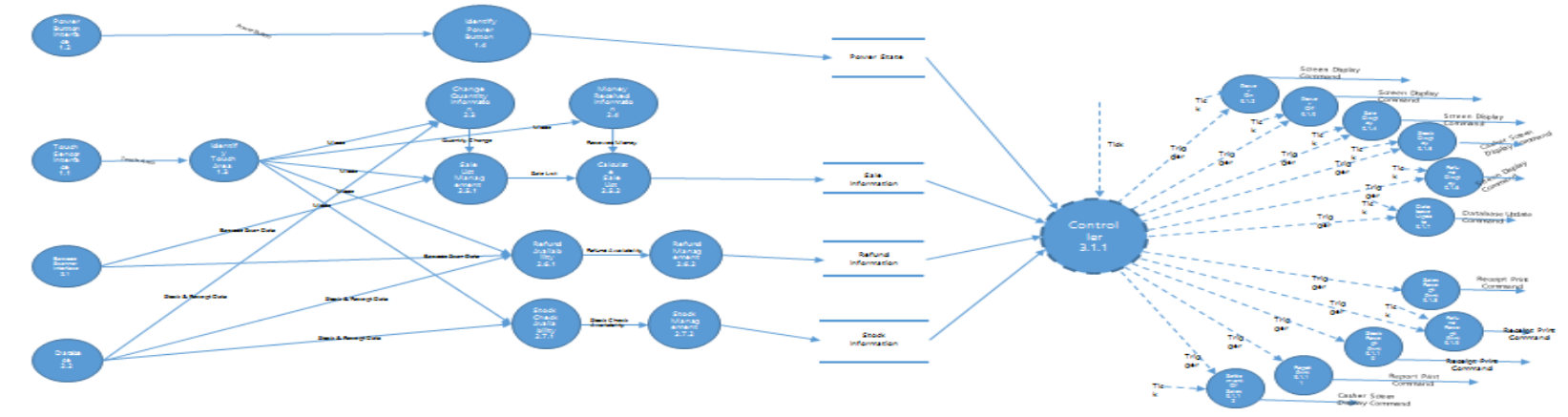


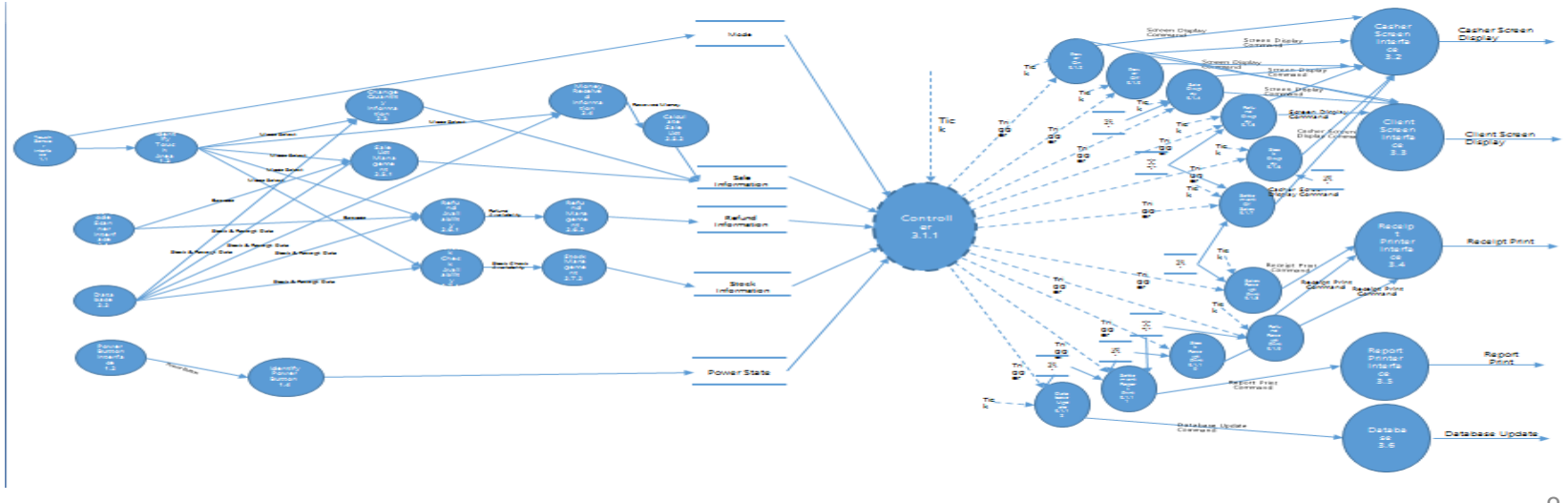
1-1. Revised SDA



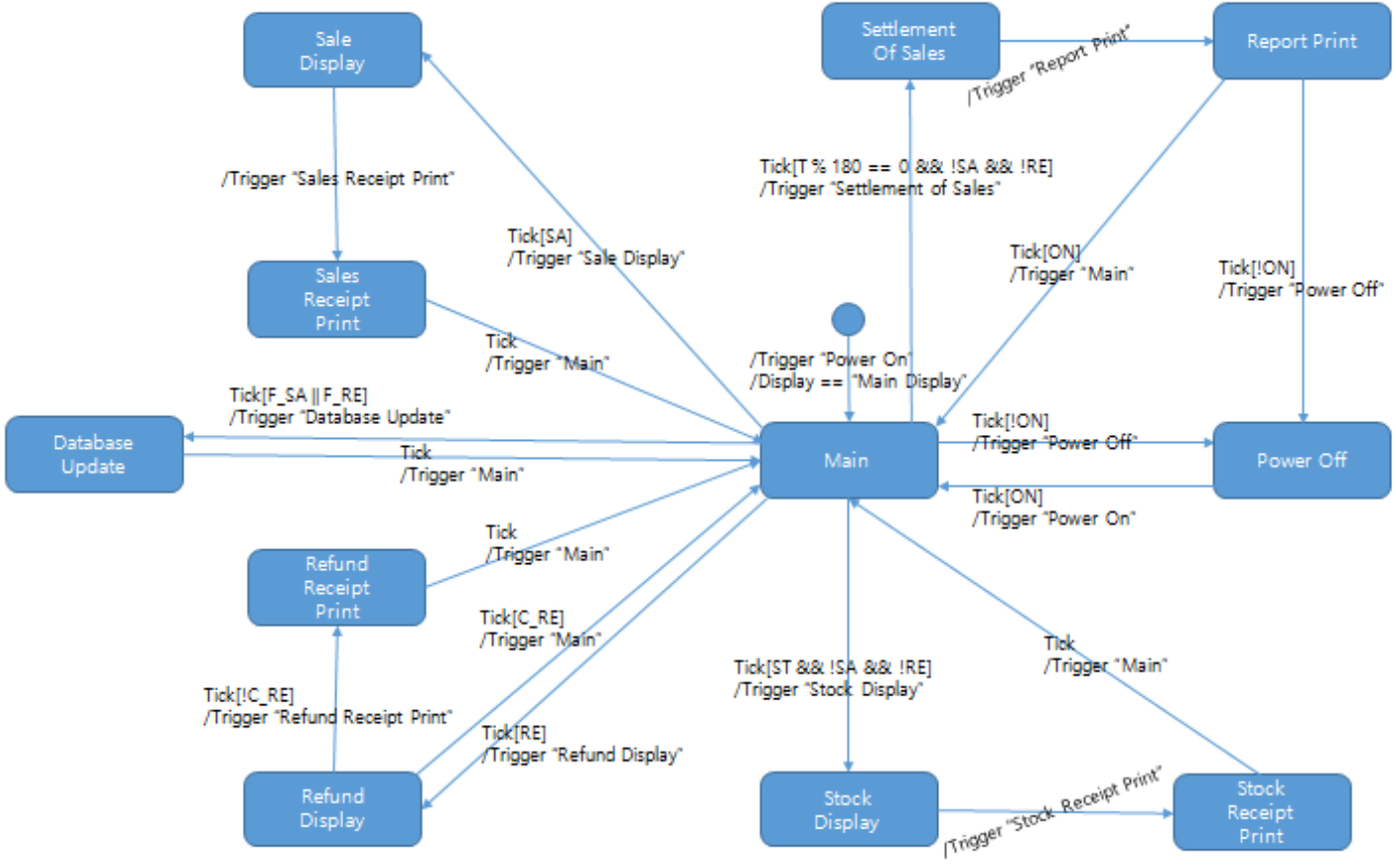


1-1. Revised SDA

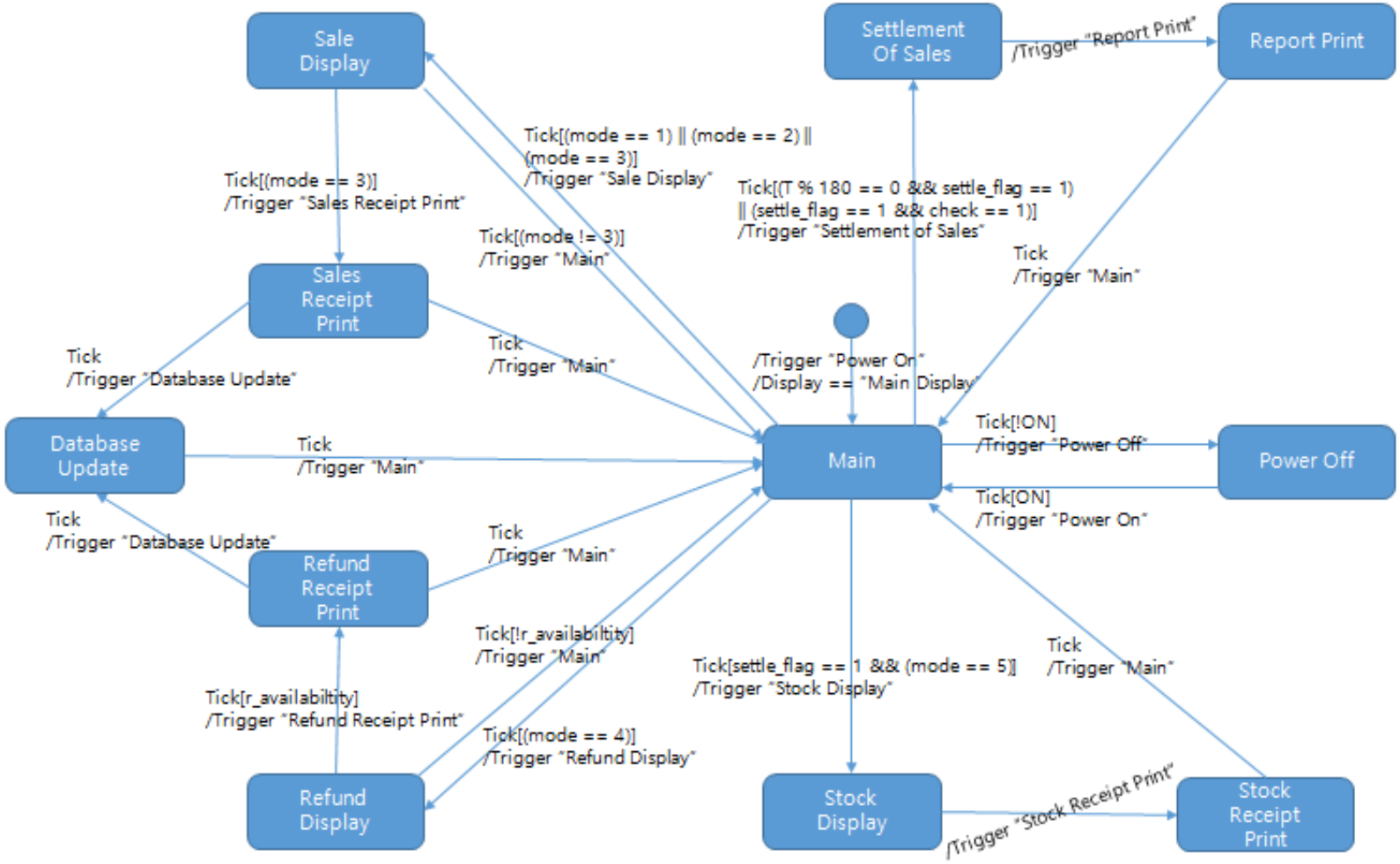




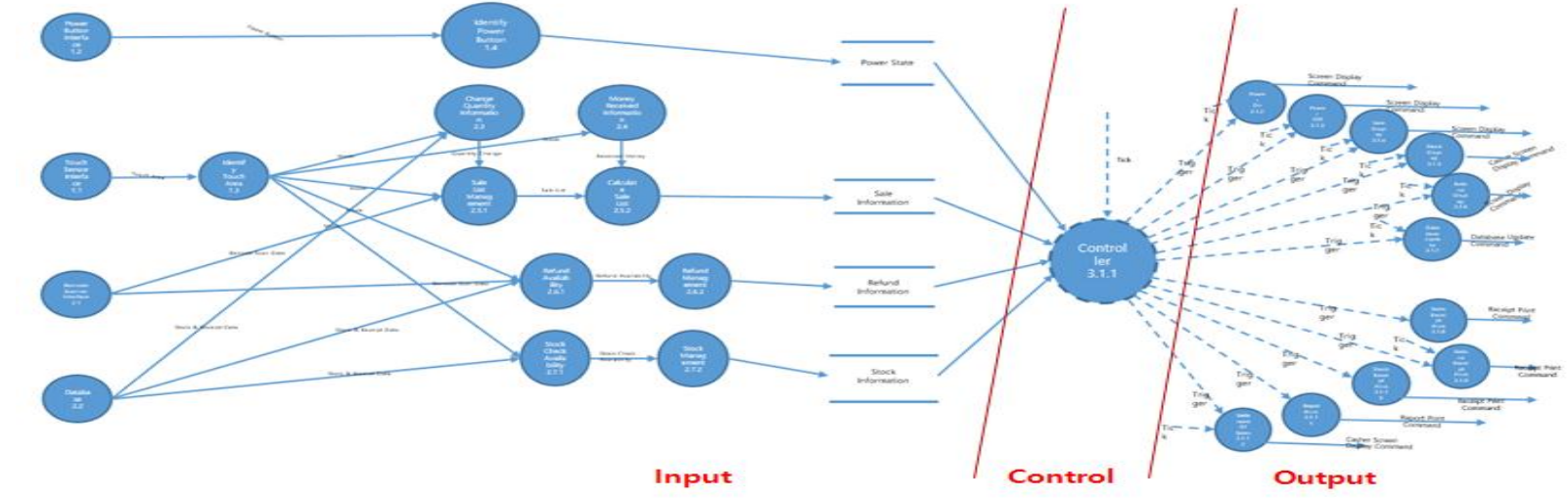
1-1. Revised SDA

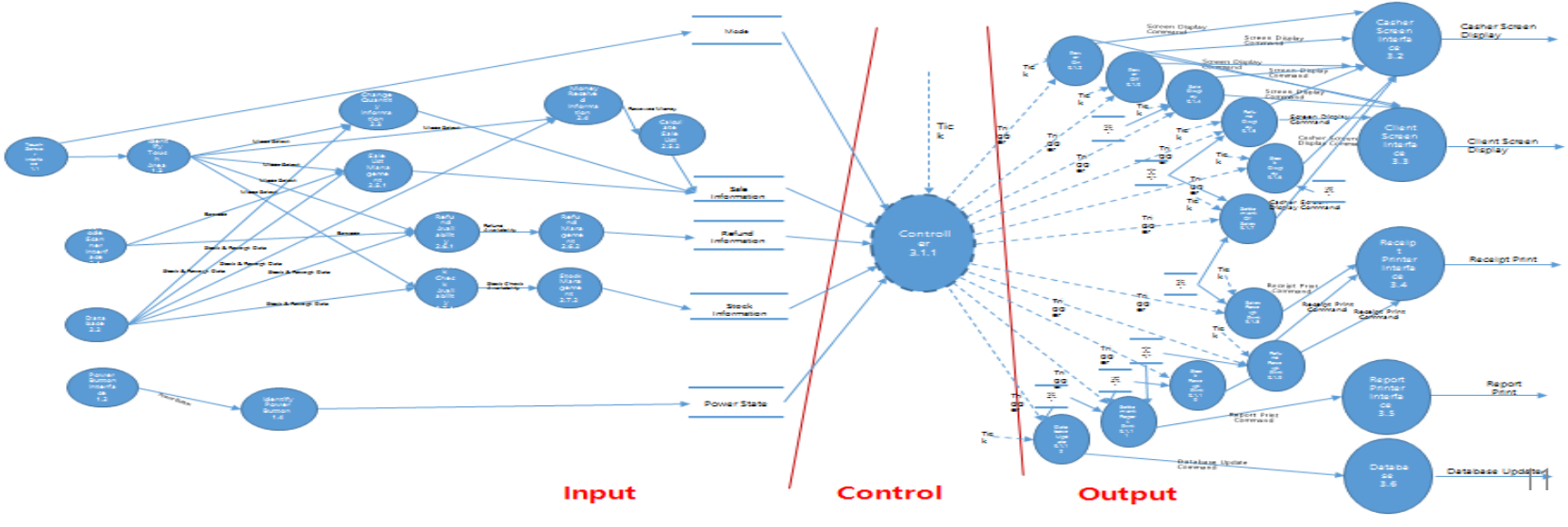


1-1. Revised SDA



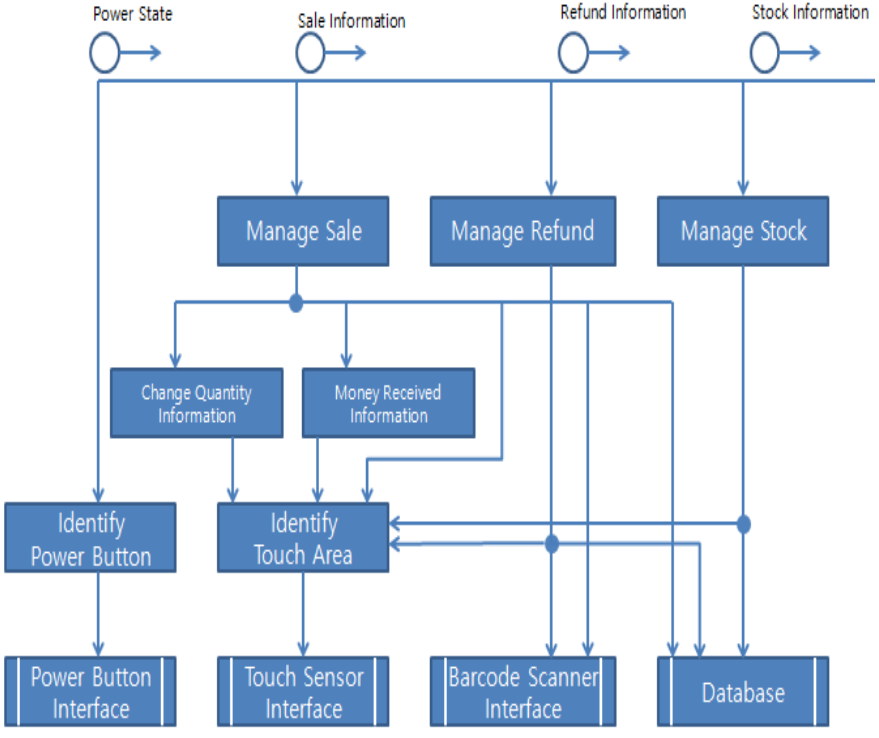
1-2. Revised SDS



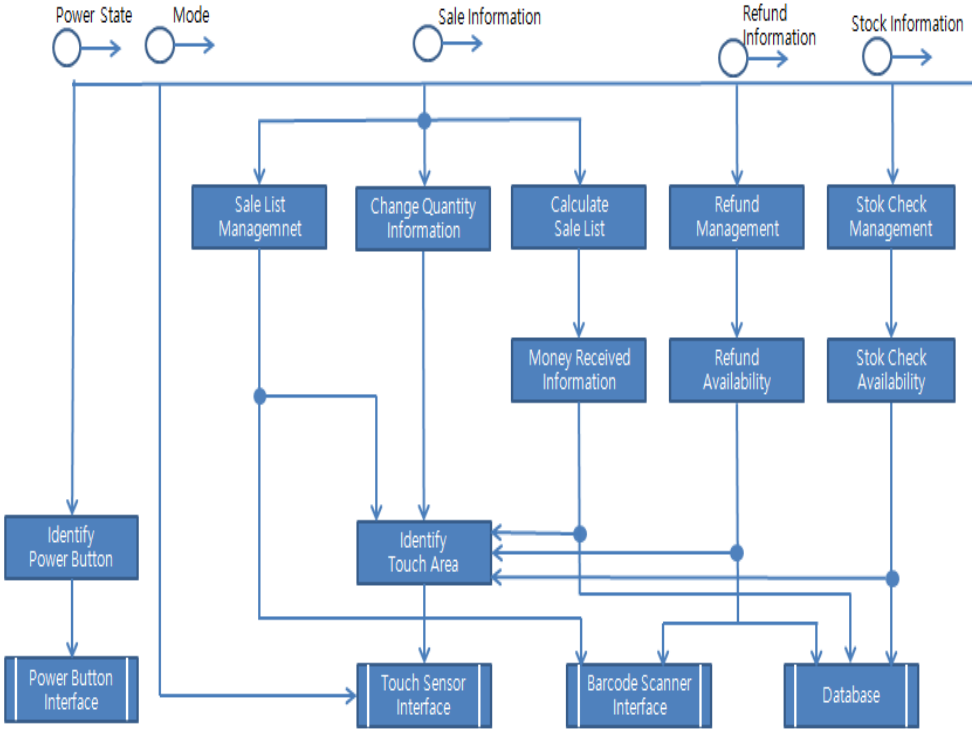


1-2. Revised SDS

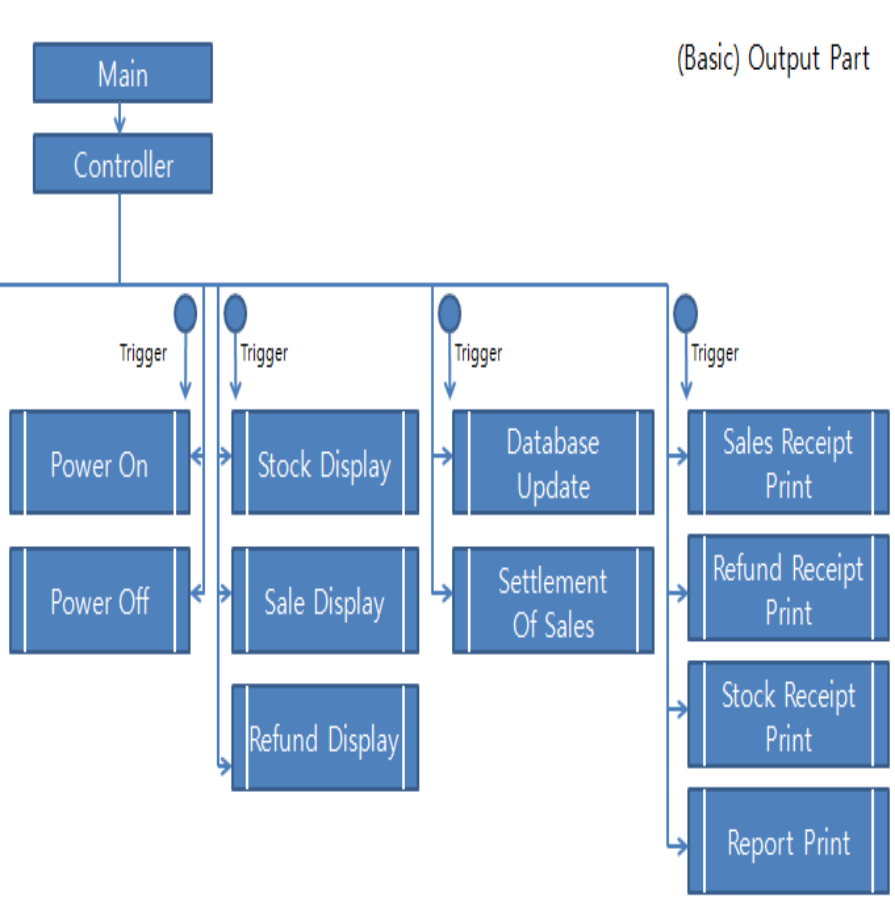
(Basic) Input Part

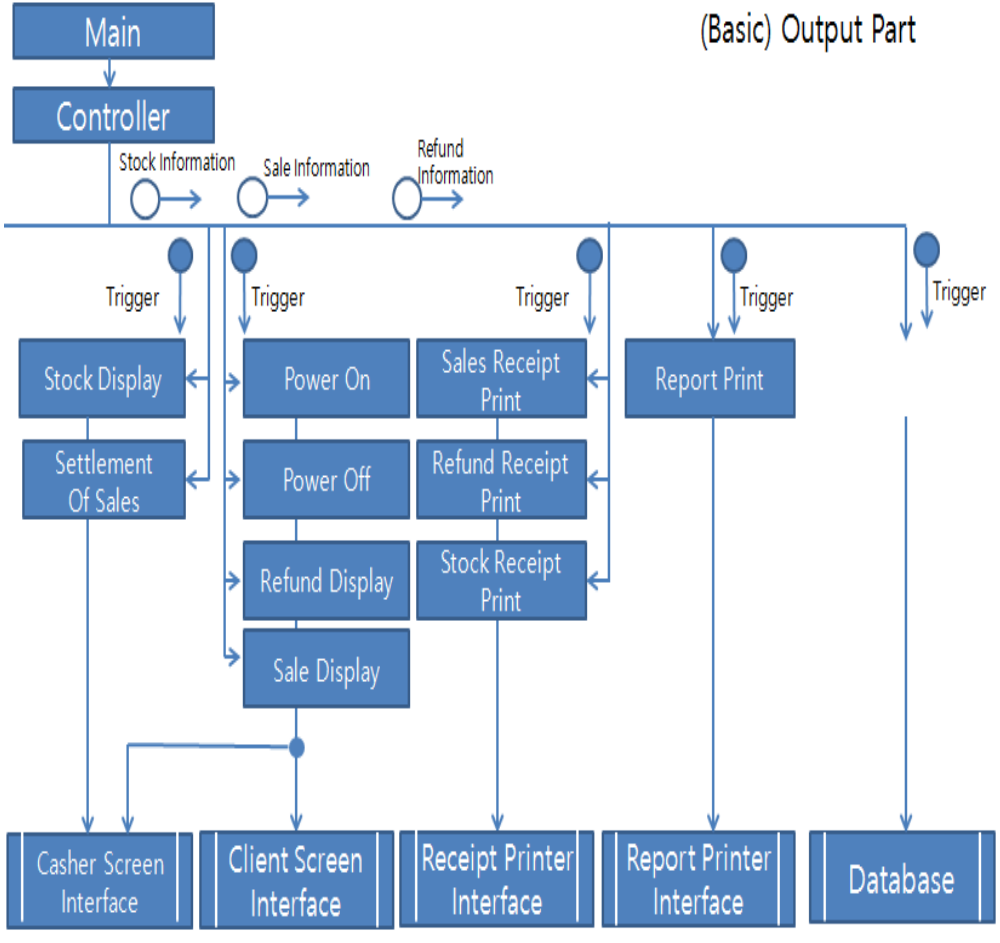


(Basic) Input Part

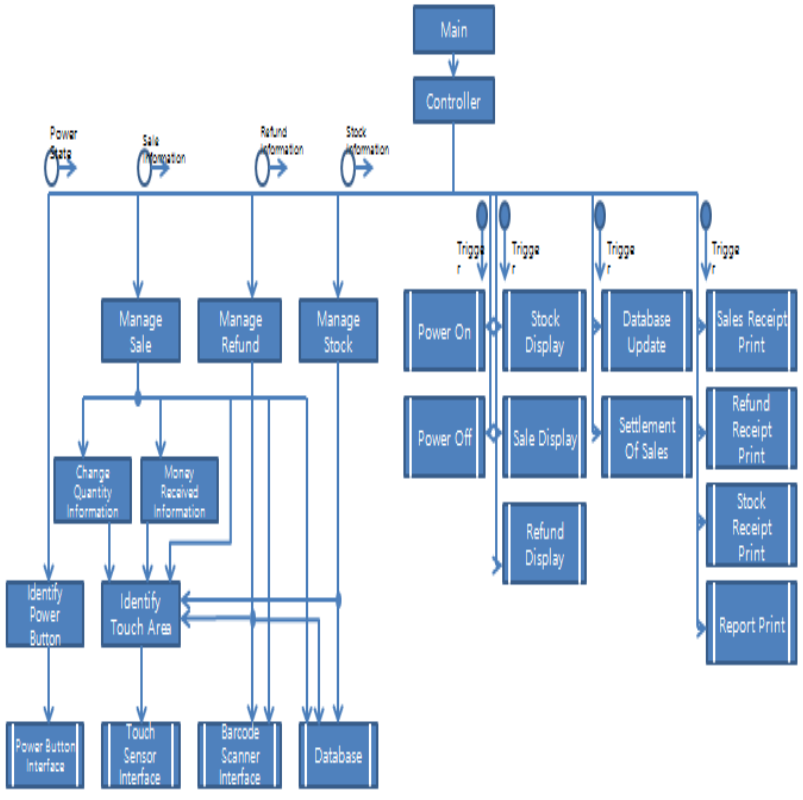


1-2. Revised SDS

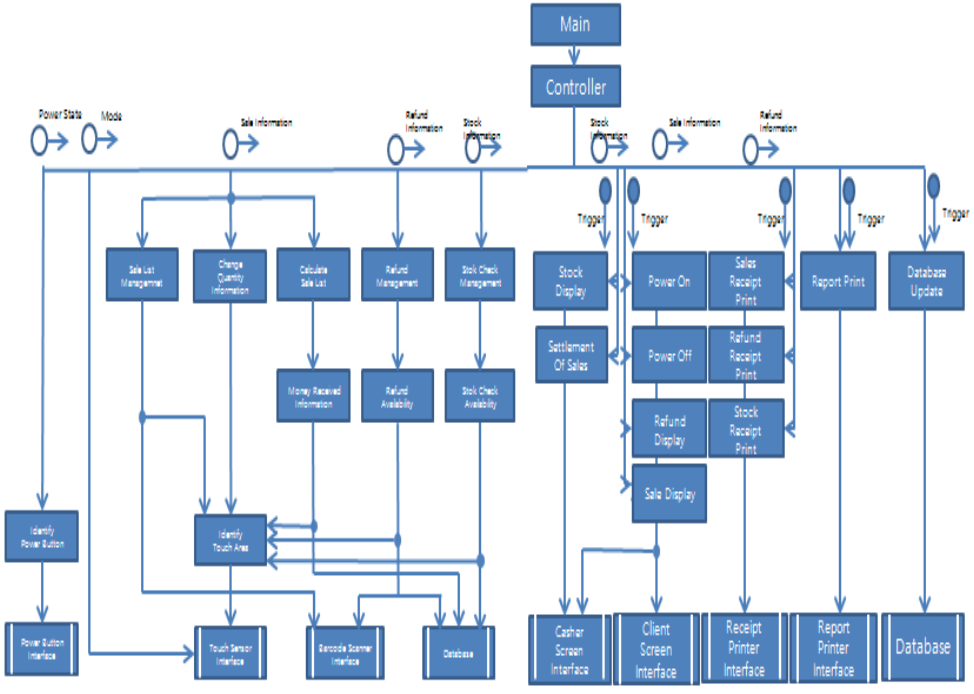




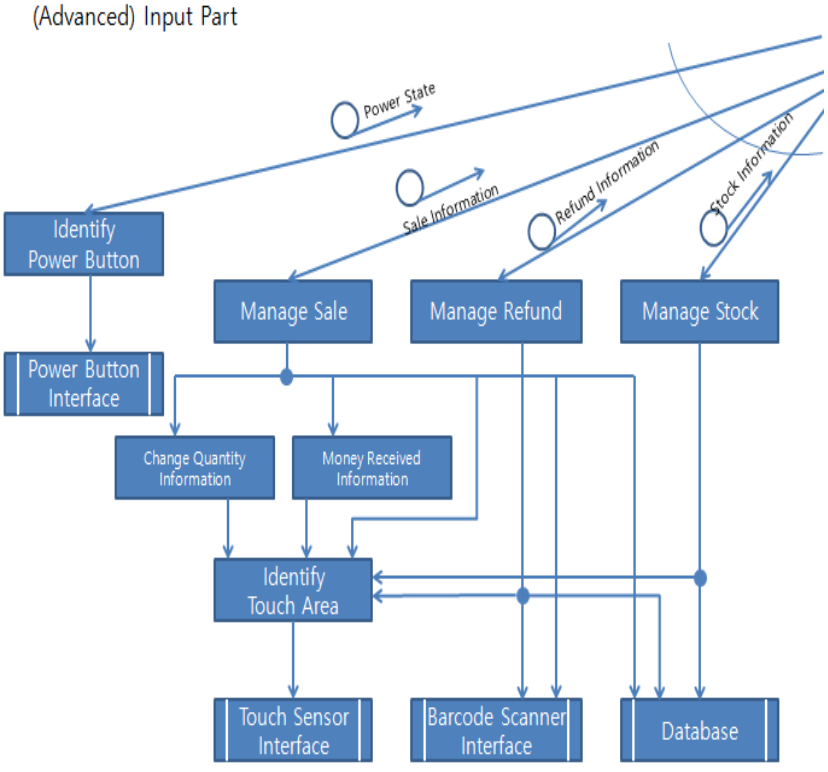
1-2. Revised SDS

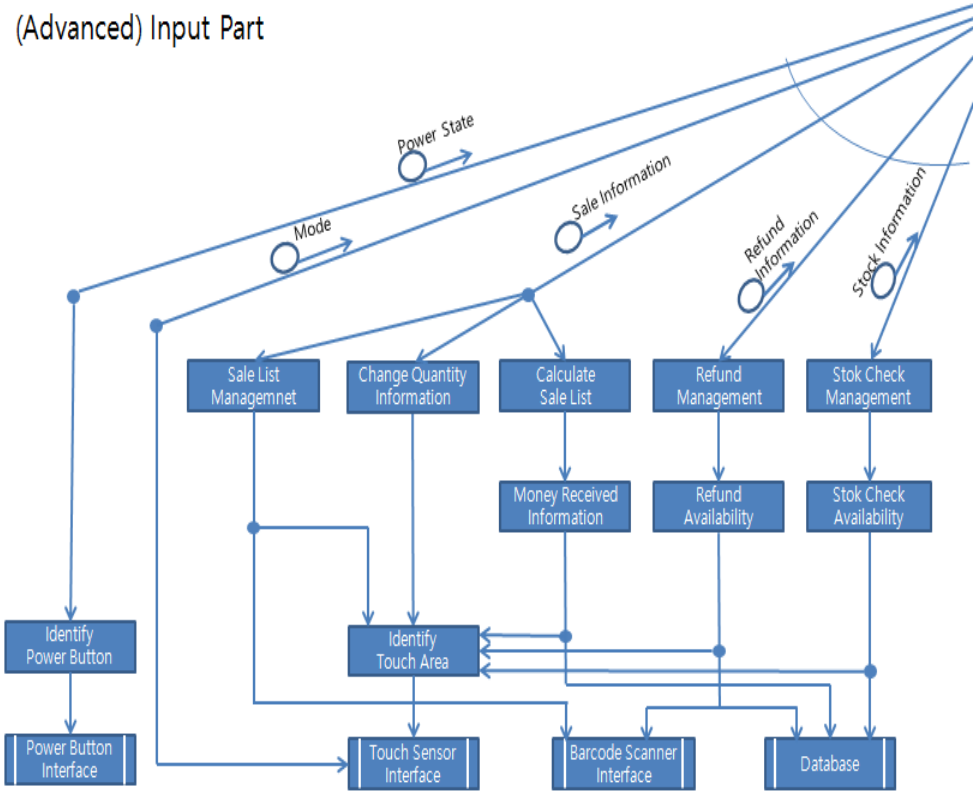


(Basic)

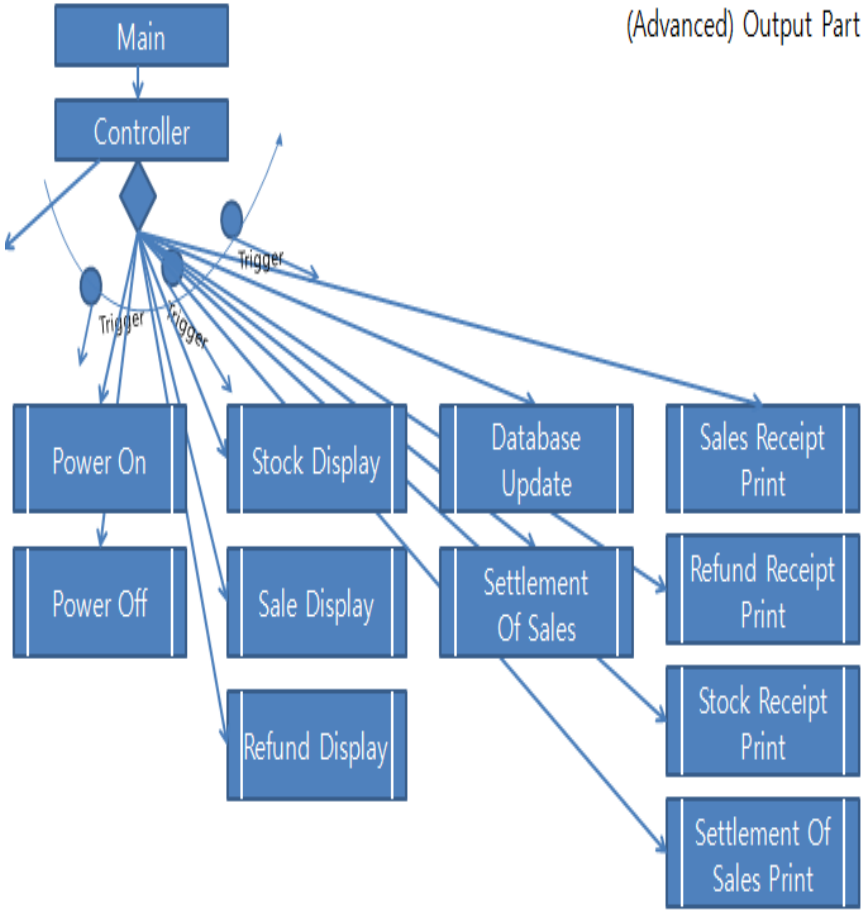


1-2. Revised SDS

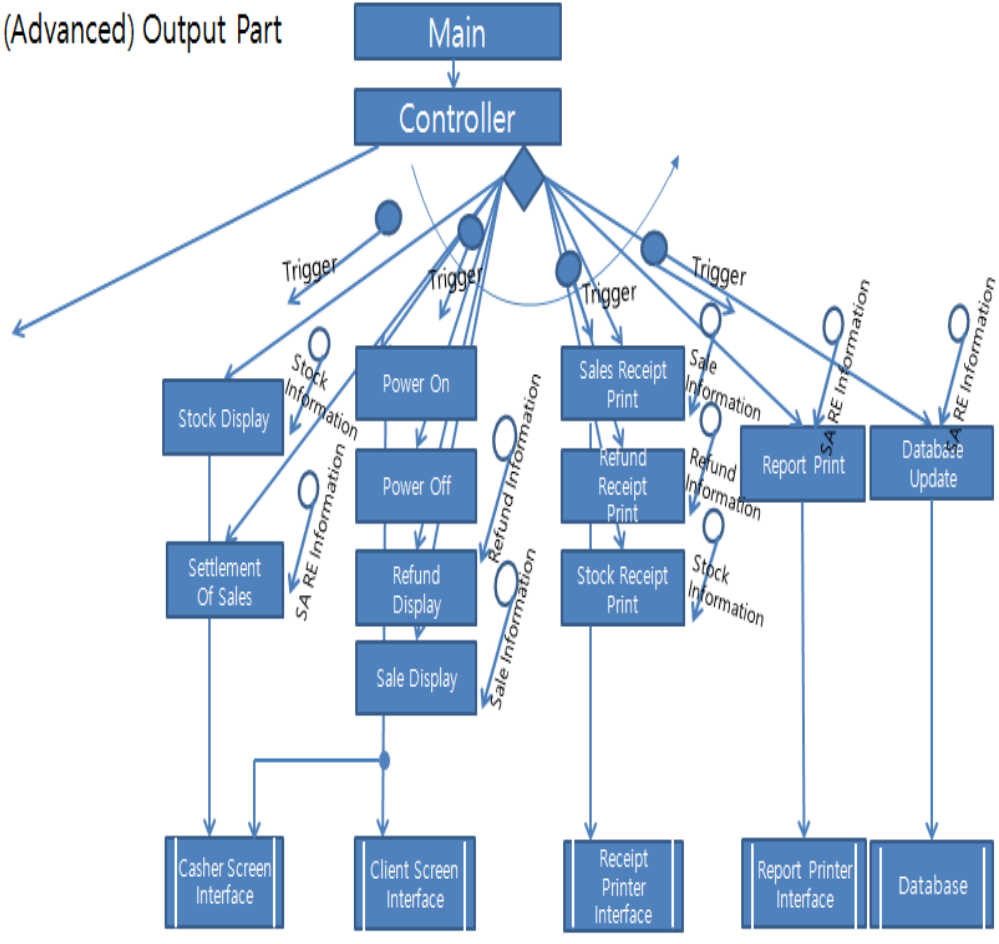




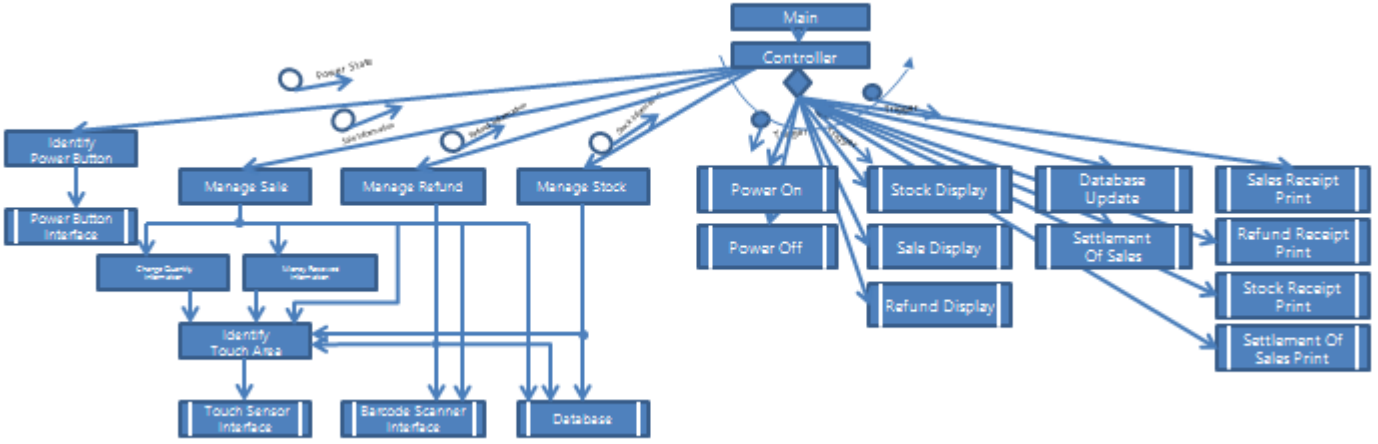
1-2. Revised SDS



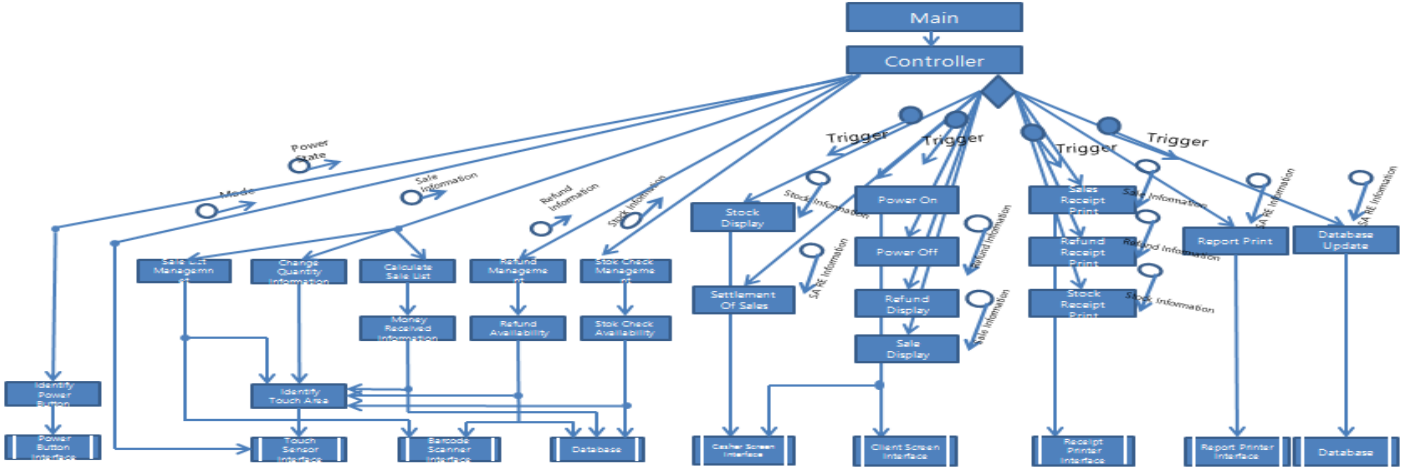
(Advanced) Output Part



1-2. Revised SDS



(Advanced)



CHAPTER 02

Code

2-1. 구현된 함수

2-1. 구현된 함수

Id	Function Name
1.3	void identify_touch_area()
1.4	int identify_power_button()
2.1	int barcode_to_index()
2.3	void quantity_change()
2.4	void money_received()
2.5.1	void sale_list_manage()
2.5.2	void calculate_sale_info()
2.6.1	int refund_availability()
2.6.2	void refund_manage()

2-1. 구현된 함수

Id	Function Name
2.7.1	int stock_check_availability()
2.7.2	
3.1.4	
3.1.5	
3.1.6	
3.1.7	
3.1.8	
3.1.9	
3.1.10	

2-1. 구현된 함수

Id	Function Name
3.1.11	
3.1.12	

CHAPTER 03

Unit Test

3-1. Unit Test Tool

3-2. Features to be tested

3-2. Features not to be tested

3-1. Unit Test Tool

-Integrated Platform Environment : Cygwin + gcc

-Test Tool : Cunit

3-2. Features to be tested

ID	Name	Description
1.3	Identify Touch Area	Touch Area를 바탕으로 그에 맞는 Mode정보를 보낸다. Input : mode Output : 함수
1.4	Identify Power Button	Power On/Off 상태를 Power State로 보낸다. Input: temp Output: Power state
2.1	Barcode Scnner Interface	바코드 정보를 받는다. Input: 바코드 int[] Output: index
2.3	Change Quantity Information	Mode 가 수량 변경 모드일 때 이루어진다. 수량 변경된 것을 Manage Sale에 알린다. Input: index, stock_info->상품.quantity Output: sale_info->상품.quantity stock_info->상품.quantity sale_info->total_price
2.4	Money Received Information	Mode가 돈 수령 모드일 때 이루어진다. 수령한 돈을 Manage Sale에 알린다. Input: received money, total price, index Output: received money, total price, sended money

3-2. Features to be tested

2.5.1	Sale List Management	Mode가 Sale모드 때 혹은 돈 수령모드나 수량 변경모드가 이뤄지고 난 이후에 이루어진다. 현재 담겨진 전체 상품 정보들을 Calculate Sale List 로 보낸다. Input: index, sale_info->상품.quantity, stock_info->상품.quantity, sale_info->total_price Ouput: sale_info->상품.quantity, stock_info->상품.quantity, sale_info->total_price
2.5.2	Calculte Sale List	담겨 진 상품 정보들을 Sale List Management로 받고 Received Money를 통해 받은 금액으로 총 계산 된 Sale Information을 만든다. Input: index, sale_info_quantity =0, sale_info_total_price, s_r[0][index], s_r[0][7] Output: sale_display(), database_update(), receipt_list_file(), sale_receipt_print(), s_r[0][index], s_r[0][7]
2.6.1	Refund Availability	Mode가 Refund모드 때 이루어진다. 바코드를 통해 환불 가능 여부를 판단해서 Refund Management 로 넘겨준다. Input: index, list_data, check Output: Refund Availability
2.6.2	Refund Management	Refund Availablity 에 따라 처리 유무를 결정한다. 유효한 영수증이면 바코드 정보를 바탕으로 환불 정보를 생성한다.

3-2. Features to be tested

2.7.1	Stock Check Availablity	Mode가 Stock 모드 때 이루어진다. 판매나 환불상태인지 확인하고 재고 확인 가능 여부를 넘겨준다. Input: settle_flag Output: stock check Availablity
3.1.4	Sale Display	판매 정보를 보여준다. Input: index, sale_info_recv_money, sale_info_snd_money sale_info_s_arr[index].name, sale_info_s_arr[index].quantity sale_info_s_arr[index].price ,sale_info_total_price Output: 출력
3.1.5	Stock Display	재고 정보를 보여준다. 재고 확인 모드이고 판매 혹은 환불 모드가 아닐 때 수행된다. Input: index, stock_info_s_arr[index].quantity stock_info_s_arr[index].name, stock_info_s_arr[index].price

3-2. Features to be tested

3.1.6	Refund Display	환불 정보를 보여준다 Input: index I, year1, month1, day1, hour1, min1, t_price refund_info[i].name, refund_info[i].price refund_info[i].quantity Output: 출력
3.1.7	Database Update	데이터베이스를 업데이트한다. Input: index I, sale_info, stock_info_s_arr[i].name stock_info_s_arr[i].price, stock_info_s_arr[i].quantity year1, month1, day1, hour1, min1 Output: 출력
3.1.8	Sales Receipt Print	판매 영수증을 출력한다. Input: index I, sale_info_s_arr[i].name sale_info_s_arr[i].price, sale_info_s_arr[i].quantity sale_info_total_price, year1, month1, day1, hour1,min1 Output: 출력
3.1.9	Refund Receipt Print	환불 영수증을 출력한다. Input: index I, refund_info_quantity refund_info_name, refund_info_price year1, month1, day1, hour1, min1 Output: 출력

3-2. Features to be tested

3.1.10	Stock Receipt Print	재고 확인 영수증을 출력한다. Input: index I, stock_info_s_arr[i].name stock_info_s_arr[i].price, stock_info_s_arr[i].quantity year1, month1, day1, hour1, min1 Output: 출력
3.1.11	Report Print	정산 보고서를 출력한다. Input: index I, day1, month1, year1 sale_info_temp[i].name, sale_info_temp[i].price sale_info_temp[i].quantity refund_info_temp[i].name, refund_info_temp[i].price refund_info_quantity total_sale_price, total_refund_price Output: 출력
3.1.12	Settlement Of Sales	3분(T%180 == 0)마다 정산을 진행한다. 판매 혹은 환불 모드가 아닐 때 수행된다.

3-2. Features not to be tested

ID	Name	Description
1.1	Touch Sensor Interface	Touch Sensor로부터 입력 정보를 받는다. 입력 정보(좌표)를 바탕으로 어느 위치가 입력 되었는지 Identify Touch Area로 보낸다.
1.2	Power Button Interface	Power Button으로부터 입력을 받는다. Power Button의 입력 정보를 Identify Power Button으로 보낸다.
2.2	Database	Database로부터 재고 및 영수증 정보를 읽어 온다. 재고 및 영수증 정보를 동시에 보내고 받는 프로세스에서 해당 정보가 자신의 맞는 것인지 확인 하고 처리한다.
2.7.2	Stock Management	Stock Check Availability에 따라 처리 유무를 결정한다. 재고확인이 가능하면 재고 정보를 생성한다
3.1.1	Controller	Data Store의 data들을 바탕으로 출력 및 정산을 제어한다.
3.1.2	Power On	POS기의 전원을 킨다. (Main 상태)
3.1.3	Power Off	POS기의 전원을 끈다.

3-2. Features not to be tested

3.2	Casher Screen Interface	Main Control로부터 Command를 받고 Casher Screen을 출력한다.
3.3	Client Screen Interface	Main Control로부터 Command를 받고 Client Screen을 출력한다.
3.4	Receipt Printer Interface	Main Control로부터 Command를 받고 판매 영수증을 출력한다.
3.5	Report Printer Interface	Main Control로부터 Command를 받고 정산 보고서를 출력한다.
3.6	Database	Main Control로부터 Command를 받고 Database를 업데이트한다.

CHAPTER 04

Test Specification

4-1. Cunit Test

4-2. Display Test

4-1. Cunit Test

Identifier	Input Specification	Output Specification
POS_1.3_000	mode : 1 n: 0	settle_flag: 0 function: Sale Mode
POS_1.3_001	mode : 2 n : 0	settle_flag: 0 function: Quantity Change Mode
POS_1.3_002	mode: 3 n : 0	settle_flag: 1 function: Money Received Mode
POS_1.3_003	mode: 4 n: 1	settle_flag: 0 function: Refund Mode
POS_1.3_004	mode: 4 n: 2	settle_flag: 1 function: NULL
POS_1.3_005	mode: 5 n: 0	settle_flag: 0 function: "Can not check Stock"
POS_1.3_006	mode: 5 n:1	settle_flag = 1 function: Stock Check Mode
POS_1.3_007	mode: 6 n: 0	settle_flag = 0 function: Reselect Mode
POS_1.4_000	temp: 1	1 (Power_on)
POS_1.4_001	temp: 2	0 (Power_off)

4-1. Cunit Test

POS_2.1_000	barcode: 001	index: 1
POS_2.1_001	barcode: 010	index: 2
POS_2.1_002	barcode: 011	index: 3
POS_2.1_003	barcode: 100	index: 4
POS_2.1_004	barcode: 101	index: 5
POS_2.1_005	barcode: 110	index: 6
POS_2.1_006	barcode: 111	index: 7

4-1. Cunit Test

POS_2.3_000	index: 1 sale_info_quantity: 0 sale_info_total_price: 0 sale_info_price: 1000 stock_info_quantity: 0	sale_info_quantity: 0 sale_info_total_price: 0 stock_info_quantity: 0 "Sold out"
POS_2.3_001	index: 1 sale_info_quantity: 1 sale_info_total_price: 500 sale_info_price: 1000 stock_info_quantity: 100	sale_info_quantity: 2 sale_info_total_price: 1500 stock_info_quantity: 99 "Sale_display"
POS_2.3_002	index: 2 sale_info_quantity: 3 sale_info_total_price: 1000 sale_info_price: 1500 stock_info_quantity: 4	sale_info_quantity: 4 sale_info_total_price: 2500 stock_info_quantity: 3 "Sale_display"
POS_2.3_003	index: 3 sale_info_quantity: 0 sale_info_total_price: 0 sale_info_price: 3000 stock_info_quantity: 0	sale_info_quantity: 0 sale_info_total_price: 0 stock_info_quantity: 0 "Sold out"
POS_2.3_004	index: 4 sale_info_quantity: 5 sale_info_total_price: 2000 sale_info_price: 500 stock_info_quantity: 50	sale_info_quantity: 6 sale_info_total_price: 2500 stock_info_quantity: 49 "Sale_display"
POS_2.3_005	index: 5 sale_info_quantity: 0 sale_info_total_price: 0 sale_info_price: 800 stock_info_quantity: 0	sale_info_quantity: 0 sale_info_total_price: 0 stock_info_quantity: 0 "Sold out"
POS_2.3_006	index: 6 sale_info_quantity: 10 sale_info_total_price: 5000 sale_info_price: 1200 stock_info_quantity: 3	sale_info_quantity: 11 sale_info_total_price: 6200 stock_info_quantity: 2 "Sale_display"
POS_2.3_007	index: 7 sale_info_quantity: 0 sale_info_total_price: 0	sale_info_quantity: 0 sale_info_total_price: 0 stock_info_quantity: 0

4-1. Cunit Test

POS_2.4_000	index: 1 recv_money: 4000 sale_info_total_price: 3000	snd_money: 1000 "Calculate_sale"
POS_2.4_001	index: 2 recv_money: 4000 sale_info_total_price :5000	snd_money: 0 "Received Money deffeceincy"
POS_2.4_002	index: 3 recv_money: 4000 sale_info_total_price: 2500	snd_money: 1500 "Calculate_sale"
POS_2.4_003	index: 4 recv_money: 4000 sale_info_total_price: 6000	snd_money: 0 "Received Money deffeceincy"
POS_2.4_004	index: 5 recv_money: 9000 sale_info_total_price: 5000	snd_money: 4000 "Calculate_sale"
POS_2.4_005	index: 6 recv_money: 2000 sale_info_total_price: 8000	snd_money: 0 "Received Money defficeincy"
POS_2.4_006	index: 7 recv_money: 4500 sale_info_total_price: 3300	snd_money: 1200 "Calculate_sale"

4-1. Cunit Test

POS_2.5.1_000	index: 1 sale_info_quantity: 1 sale_info_total_price: 500 sale_info_price: 1000 stock_info_quantity: 100	sale_quantity: 2 sale_total_price: 1500 stock_quantity: 99 "Sale_display"
POS_2.5.1_001	index: 2 sale_info_quantity: 3 sale_info_total_price : 1000 sale_info_price: 1500 stock_info_quantity: 4	sale_quantity: 4 sale_total_price: 2500 stock_quantity: 3 "Sale_display"
POS_2.5.1_002	index: 3 sale_info_quantity: 0 sale_info_total_price : 0 sale_info_price: 3000 stock_info_quantity: 12	sale_quantity: 1 sale_total_price: 3000 stock_quantity: 11 "Sale_display"

4-1. Cunit Test

POS_2.5.1_003	index: 4 sale_info_quantity: 5 sale_info_total_price: 2000 sale_info_price: 500 stock_info_quantity: 50	sale_quantity: 6 sale_total_price: 2500 stock_quantity: 49 "Sale_display"
POS_2.5.1_004	index: 5 sale_info_quantity: 4 sale_info_total_price: 0 sale_info_price: 800 stock_info_quantity: 20	sale_quantity: 5 sale_total_price: 800 stock_quantity: 19 "Sale_display"
POS_2.5.1_005	index: 6 sale_info_quantity: 10 sale_info_total_price: 5000 sale_info_price: 1200 stock_info_quantity: 3	sale_quantity: 11 sale_total_price: 6200 stock_quantity: 2 "Sale_display"
POS_2.5.1_006	index: 7 sale_info_quantity: 8 sale_info_total_price: 0 sale_info_price: 2000 stock_info_quantity: 10	sale_quantity: 9 sale_total_price: 2000 stock_quantity: 9 "Sale_display"
POS_2.5.1_007	index: 0 sale_info_quantity: 9 sale_info_total_price: 0 sale_info_price: 4000 stock_info_quantity: 5	sale_quantity: 9 sale_total_price: 0 stock_quantity: 5 "Invalid Index"

4-1. Cunit Test

POS_2.5.2_000	index: 1 sale_info_quantity: 0 sale_info_total_price: 1000 s_r_0_i: 3 s_r_0_7_i: 1500	s_r_0: 3 s_r_0_7: 2500 "sale_display" "database_update" "receipt_list_file" "sale_receipt_print"
POS_2.5.2_001	index: 1 sale_info_quantity: 1 sale_info_total_price: 1000 s_r_0_i: 2 s_r_0_7_i: 2000	s_r_0: 3 s_r_0_7: 3000 "sale_display" "database_update" "receipt_list_file" "sale_receipt_print"
POS_2.6.1_000	check: 0	0

4-1. Cunit Test

	barcode: 001 t_barcode: 001 list_data: 0	
POS_2.6.1_001	check: 1 barcode: 001 t_barcode: 001 list_data: 0	0
POS_2.6.1_002	check: 0 barcode: 001 t_barcode: 001 list_data: 1	1
POS_2.6.1_003	check: 0 barcode: 001 t_barcode: 010 list_data: 1	2
POS_2.6.1_004	check: 1 barcode: 001 t_barcode: 001 list_data: 1	3
POS_2.6.1_005	check: 1 barcode: 001 t_barcode: 010 list_data: 1	4

4-1. Cunit Test

POS_2.6.1_005	check: 1 barcode: 001 t_barcode: 010 list_data: 1	4
POS_2.6.2_000	refund_i_quantity : 0 refund_i_name : "snack" stock_k_quantity : 2 stock_k_name: "water"	stock_k_quantity: 2 함수 호출
POS_2.6.2_001	refund_i_quantity : 1 refund_i_name : "snack" stock_k_quantity: 2 stock_k_name : "snack"	stock_k_quantity: 3 함수 호출
POS_2.6.2_002	refund_i_quantity : 1 refund_i_name "snack" stock_k_quantity: 2 stock_k_name "water"'	stock_k_quantity: 2 함수 호출
POS_2.7.1_000	settle_flag: 0	0
POS_2.7.1_001	settle_flag: 1	1
POS_3.1.4_000	index : 1	화면 출력

4-1. Cunit Test

POS_3.1.12_000	settle_flag: 2 get_time_flag: 0 check: 0 year: 2017 month: 11 day: 6 hour: 3 min: 47 start: 1 end: 100	check : 0 get_time_flag: 0 update_flag: 0 return int: 0
POS_3.1.12_001	settle_flag: 1 get_time_flag: 1 check: 1	check = 0; get_time_flag :0 update_flag 0

	year: 2017 month: 11 day: 6 hour: 3 min: 47 start: 0 end: 100	return int 2

4-2. Display Test

	sale_info_recv_money: 1000 sale_info_snd_money: 2000 sale_info_s_arr[index].name: "snack" sale_info_s_arr[index].quantity: 0 sale_info_s_arr[index].price: 1000 sale_info_total_price: 5000	
POS_3.1.4_001	index : 1 sale_info_recv_money: 1000 sale_info_snd_money: 2000 sale_info_s_arr[index].name: "snack" sale_info_s_arr[index].quantity: 1 sale_info_s_arr[index].price : 1000 sale_info_total_price: 5000	화면 출력
POS_3.1.5_000	index: 1 stock_info_s_arr[index].quantity: 0 stock_info_s_arr[index].name: "snack" stock_info_s_arr[index].price: 1000	화면 출력
POS_3.1.5_001	index: 1 stock_info_s_arr[index].quantity: 1 stock_info_s_arr[index].name:"snack" stock_info_s_arr[index].price: 1000	화면 출력
POS_3.1.6_000	index l : 1 year1: 2017 month1: 11 day1: 6	화면 출력

4-2. Display Test

POS_3.1.6_000	index I : 1 year1: 2017 month1: 11 day1: 6 hour1: 12 min1: 7 t_price: 4500 refund_info[i].name: "snack" refund_info[i].price: 3000 refund_info[i].quantity: 0	화면 출력
POS_3.1.6_001	index I : 1 year1 : 2017 month1: 11 day1: 6 hour1: 12 min1: 7	화면 출력

	t_price: 4500 refund_info[i].name: "snack" refund_info[i].price: 3000 refund_info[i].quantity: 1	

4-2. Display Test

POS_3.1.7_001	index I : 1 sale_info: 1 sale_info_s_arr[i].quantity: 2 sale_info_s_arr[i].name: "snack" sale_info_s_arr[i].price: 1000 sale_info_s_arr[i].total_price : 3500 stock_info_s_arr[i].name: "snack" stock_info_s_arr[i].price: 1000 stock_info_s_arr[i].quantity: 0 year1: 2017 month1: 11 day1: 6 hour1: 12 min1: 7	화면 출력
POS_3.1.7_002	index I =1 sale_info: 1 sale_info_s_arr[i].quantity: 2 sale_info_s_arr[i].name: "snack" sale_info_s_arr[i].price: 1000 sale_info_s_arr[i].total_price: 3500 stock_info_s_arr[i].name: "snack"	화면 출력

	t_price: 4500 refund_info[i].name: "snack" refund_info[i].price: 3000 refund_info[i].quantity: 1	

4-2. Display Test

POS_3.1.7_000	index I :1 sale_info: 0 sale_info_s_arr[i].quantity: 2 sale_info_s_arr[i].name: "snack" sale_info_s_arr[i].price: 1000 sale_info_s_arr[i].total_price: 3500 stock_info_s_arr[i].name: "snack" stock_info_s_arr[i].price: 1000 stock_info_s_arr[i].quantity: 0 year1: 2017 month1: 11 day1: 6 hour1: 12 min1: 7	화면 출력
POS_3.1.7_001	index I : 1 sale_info: 1 sale_info_s_arr[i].quantity: 2 sale_info_s_arr[i].name: "snack" sale_info_s_arr[i].price: 1000 sale_info_s_arr[i].total_price : 3500 stock_info_s_arr[i].name: "snack" stock_info_s_arr[i].price: 1000 stock_info_s_arr[i].quantity: 0 year1: 2017 month1: 11 day1: 6 hour1: 12 min1: 7	화면 출력
POS_3.1.7_002	index I =1 sale_info: 1 sale_info_s_arr[i].quantity: 2 sale_info_s_arr[i].name: "snack" sale_info_s_arr[i].price: 1000 sale_info_s_arr[i].total_price: 3500 stock_info_s_arr[i].name: "snack"	화면 출력

4-2. Display Test

	stock_info_s_arr[i].price: 1000 stock_info_s_arr[i].quantity: 1 year1: 2017 month1: 11 day1: 6 hour1: 12 min1: 7	
POS_3.1.8_000	index I :1 sale_info_s_arr[i].name: "snack" sale_info_s_arr[i].price: 1000 sale_info_s_arr[i].quantity: 0 sale_info_total_price: 5000 year1 : 2017 month1: 11 day1: 6 hour1: 12 min1: 15	화면 출력
POS_3.1.8_001	index I : 1 sale_info_s_arr[i].name: "snack" sale_info_s_arr[i].price: 1000 sale_info_s_arr[i].quantity: 1 sale_info_total_price: 5000 year1: 2017 month1: 11 day1: 6	화면 출력

4-2. Display Test

POS_3.1.9_000	index I :1 refund_info_quantity: 0 refund_info_name: "snack" refund_info_price: 1000 year1: 2017 month1: 11 day1: 6 hour1: 12 min1: 17	화면 출력
POS_3.1.9_001	index I :1 refund_info_quantity: 1 refund_info_price: 1000 year1: 2017 month1: 11 day1: 6 hour1: 12 min1:17	
POS_3.1.10_000	index I : 1 stock_info_s_arr[i].name: "snack" stock_info_s_arr[i].price: 1000 stock_info_s_arr[i].quantity: 4 year1 : 2017 month1: 11 day1: 6 hour1: 12 min1: 19	화면 출력
POS_3.1.11_000	index I : 1 day1 : 6 month1: 11 year1: 2017 sale_info_temp[i].name: "snack" sale_info_temp[i].price: 1000 sale_info_temp[i].quantity: 0 refund_info_temp[i].name: "snack" refund_info_temp[i].price: 1000 refund_info_quantity: 0	화면 출력

4-2. Display Test

	sale_info_temp[i].price: 1000 sale_info_temp[i].quantity: 0 refund_info_temp[i].name: "snack" refund_info_temp[i].price: 1000 refund_info_quantity: 0 total_sale_price: 2000 total_refund_price: 1000	
POS_3.1.11_001	index I : 1 day1: 6 month1: 11 year1: 2017 sale_info_temp[i].name: "snack" sale_info_temp[i].price: 1000 sale_info_temp[i].quantity: 0 refund_info_temp[i].name: "snack" refund_info_temp[i].price: 1000 refund_info_quantity: 1 total_sale_price: 2000	화면 출력
POS_3.1.11_002	index I : 1 day1: 6 month1: 11 year1: 2017 sale_info_temp[i].name: "snack" sale_info_temp[i].price: 1000 sale_info_temp[i].quantity: 1 refund_info_temp[i].name: "snack" refund_info_temp[i].price: 1000 refund_info_quantity: 0 total_sale_price: 2000 total_refund_price: 1000	화면 출력
POS_3.1.11_003	index I = 1 day1: 6 month1 : 11 year1: 2017 sale_info_temp[i].name: "snack" sale_info_temp[i].price: 1000 sale_info_temp[i].quantity: 1 refund_info_temp[i].name: "snack" refund_info_temp[i].price: 1000 refund_info_quantity: 1 total_sale_price:2000 total_refund_price:1000	화면 출력

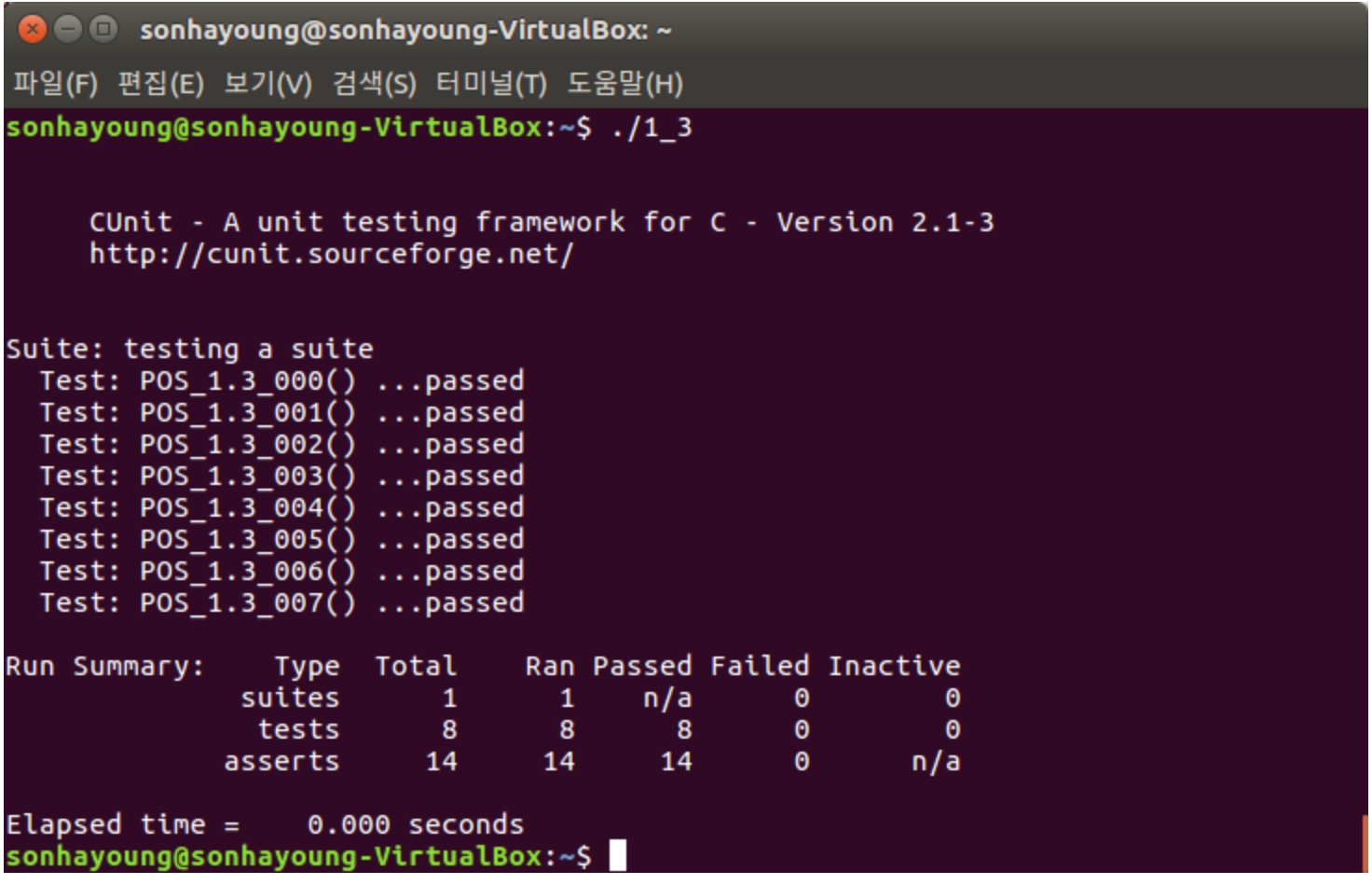
CHAPTER 05

Result

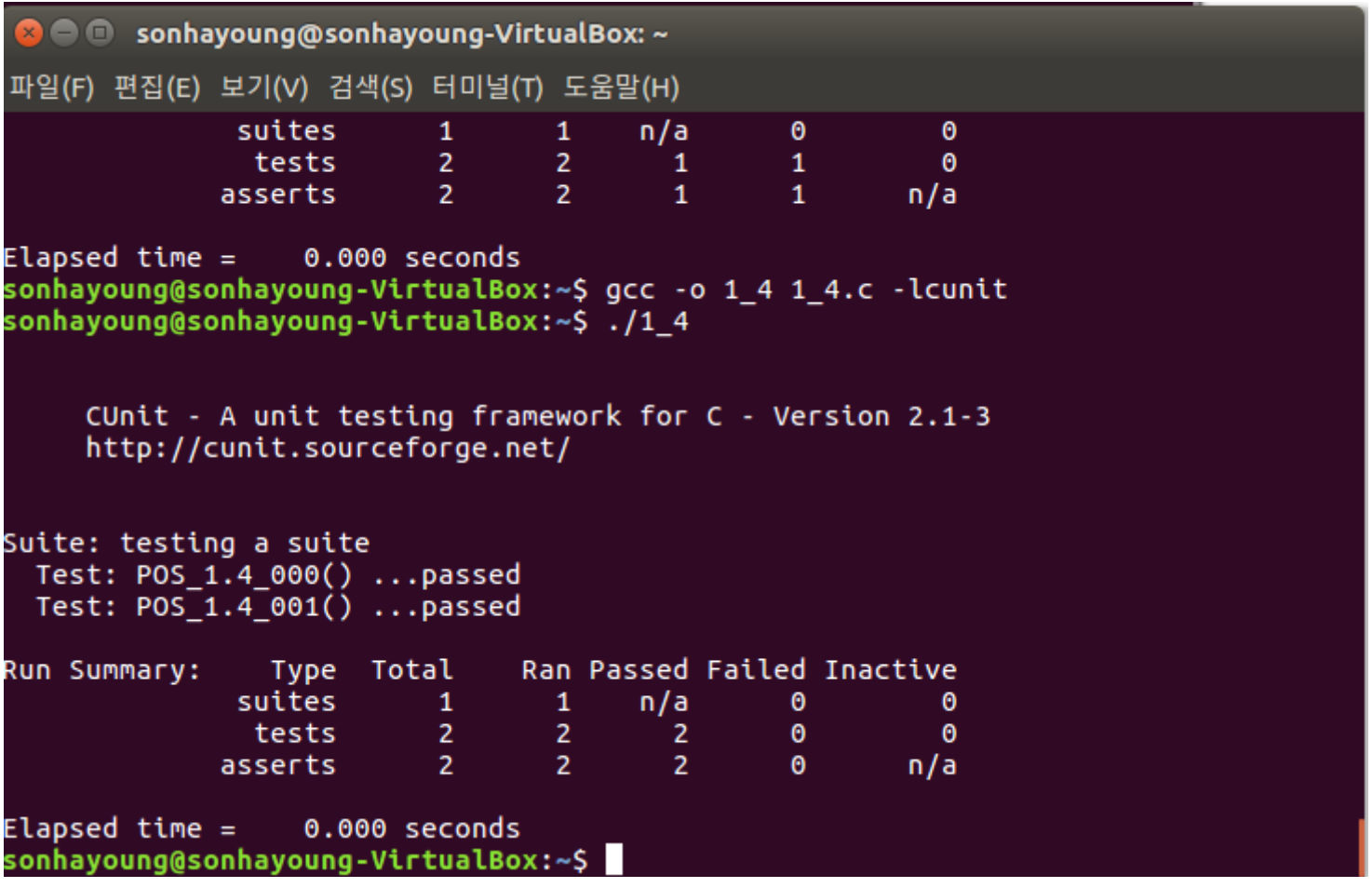
5-1. Cunit Result

5-2. Display Result

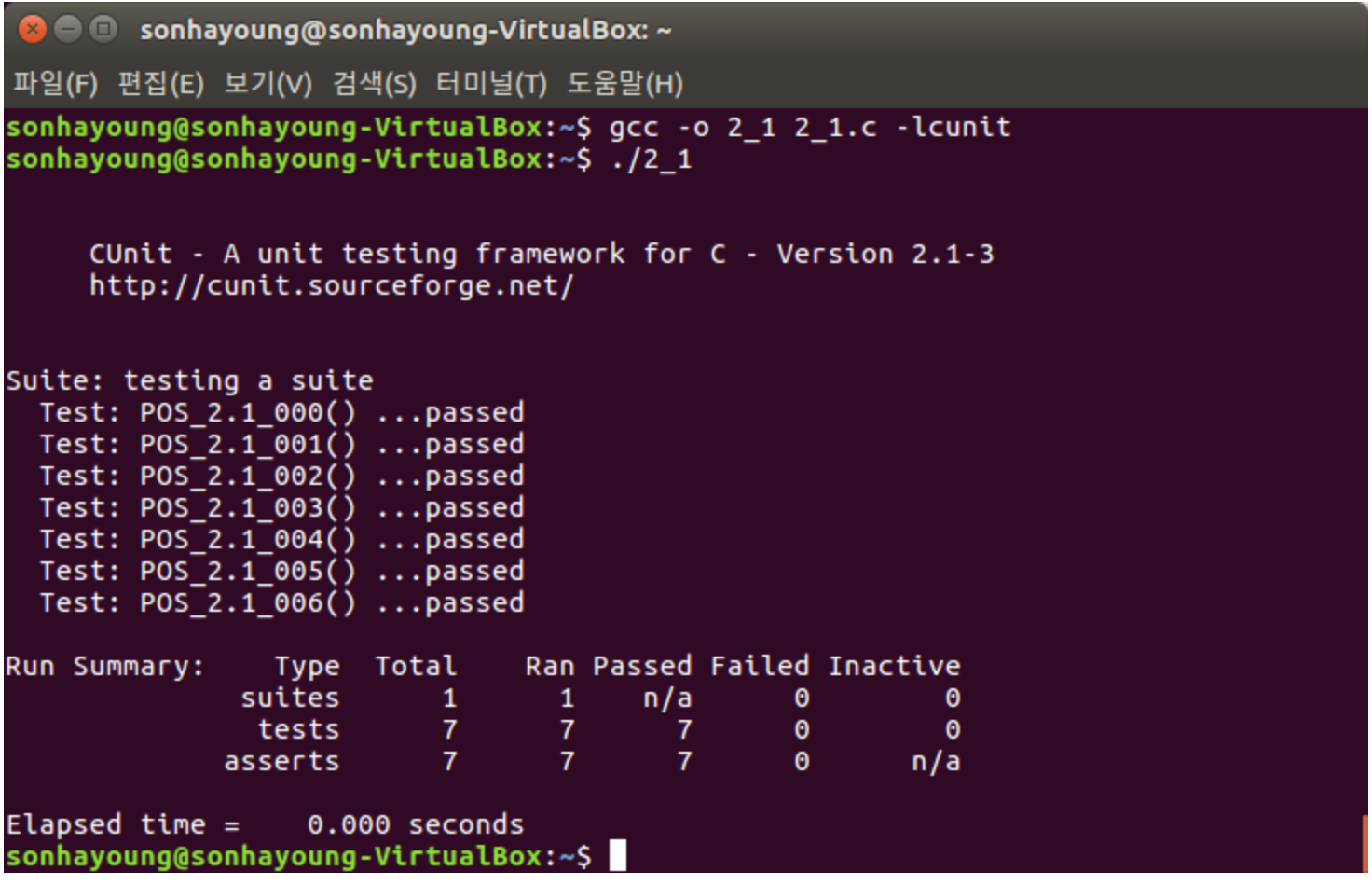
5-1. Cunit Result (1.3)



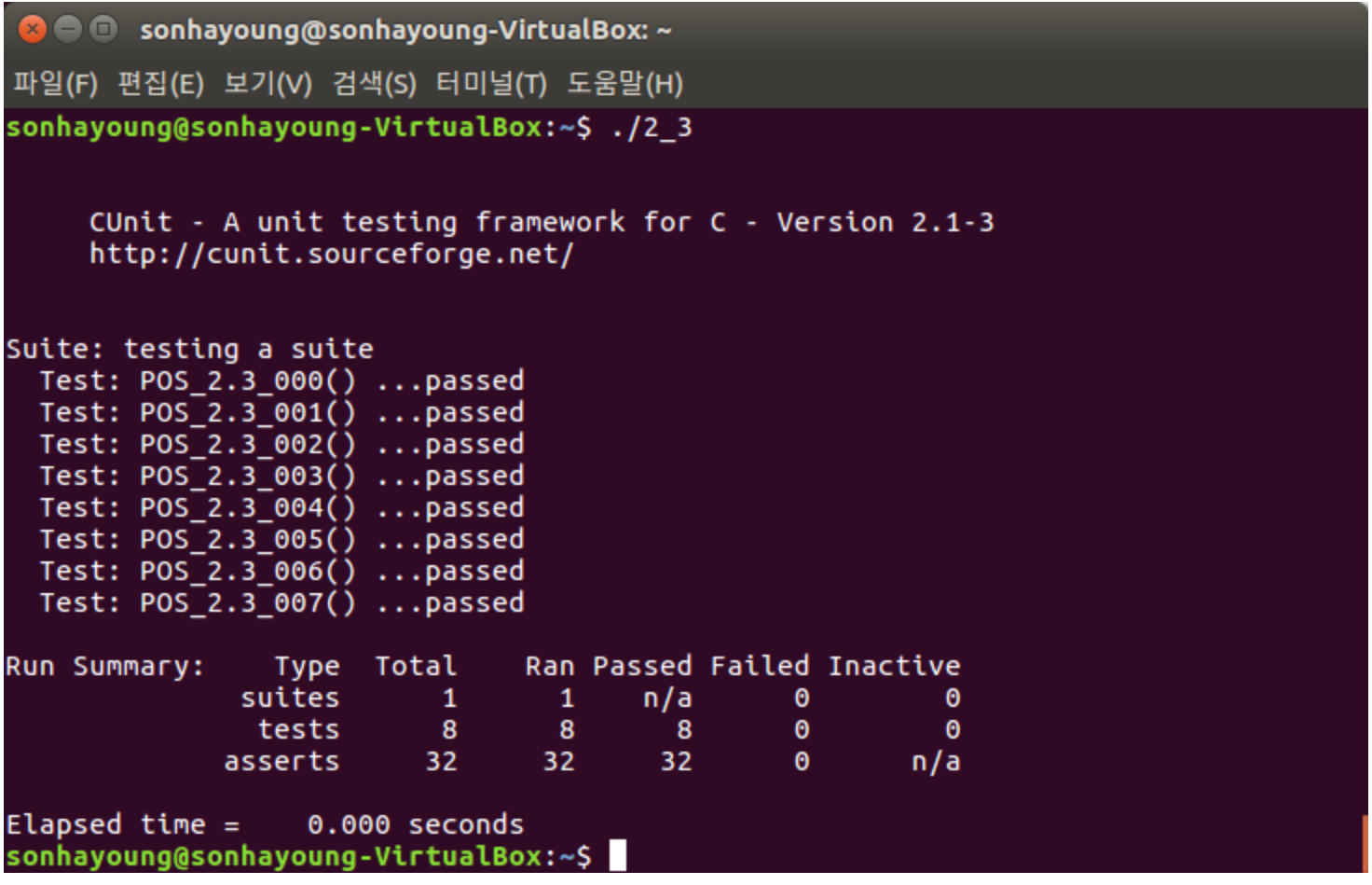
5-1. Cunit Result (1.4)



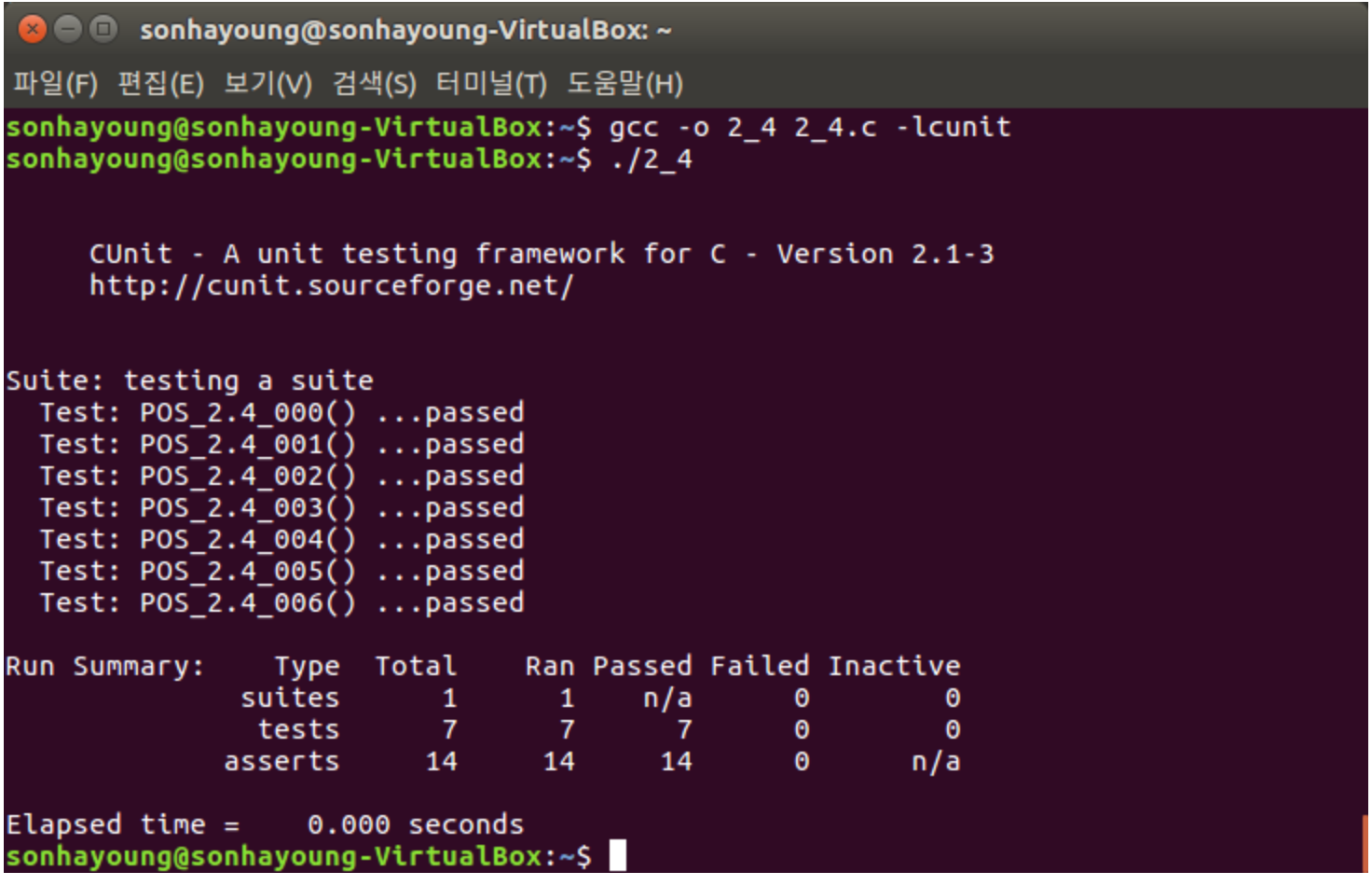
5-1. Cunit Result (2.1)



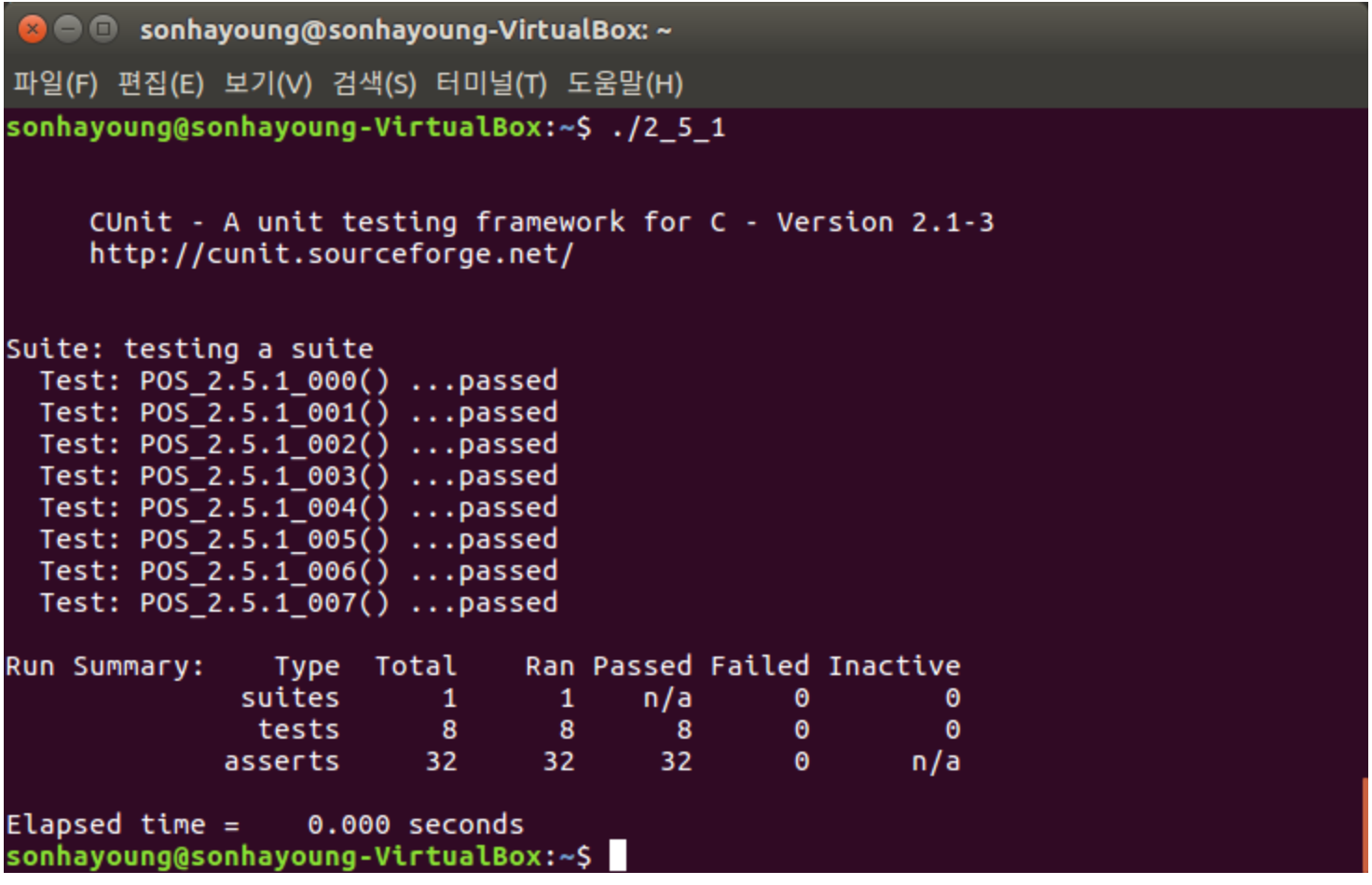
5-1. Cunit Result (2.3)



5-1. Cunit Result (2.4)



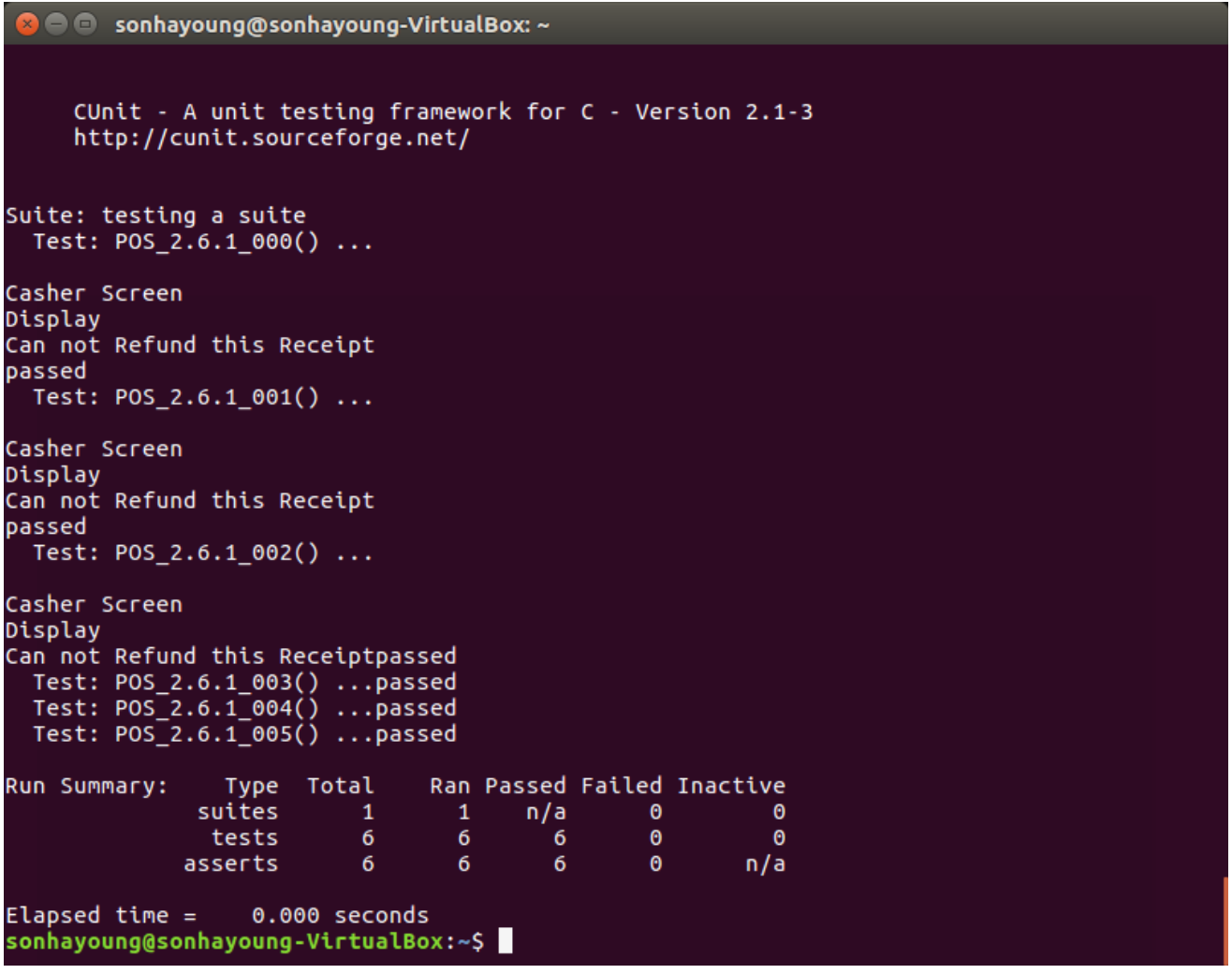
5-1. Cunit Result (2.5.1)



5-1. Cunit Result (2.5.2)

```
sonhayoung@sonhayoung-VirtualBox: ~
파일(F)  편집(E)  보기(V)  검색(S)  터미널(T)  도움말(H)
              suites      1      1    n/a      0        0
               tests      2      2      1      1        0
             asserts     12     12     11      1      n/a

Elapsed time =    0.000 seconds
sonhayoung@sonhayoung-VirtualBox:~$ gcc -o 2_5_2 2_5_2.c -lcunit
sonhayoung@sonhayoung-VirtualBox:~$ ./2_5_2


     CUnit - A unit testing framework for C - Version 2.1-3
     http://cunit.sourceforge.net/


Suite: testing a suite
  Test: POS_2.5.2_000() ...passed
  Test: POS_2.5.2_001() ...passed

Run Summary:    Type  Total    Ran Passed Failed Inactive
              suites      1      1    n/a      0        0
               tests      2      2      2      0        0
             asserts     12     12     12      0      n/a

Elapsed time =    0.000 seconds
sonhayoung@sonhayoung-VirtualBox:~$
```

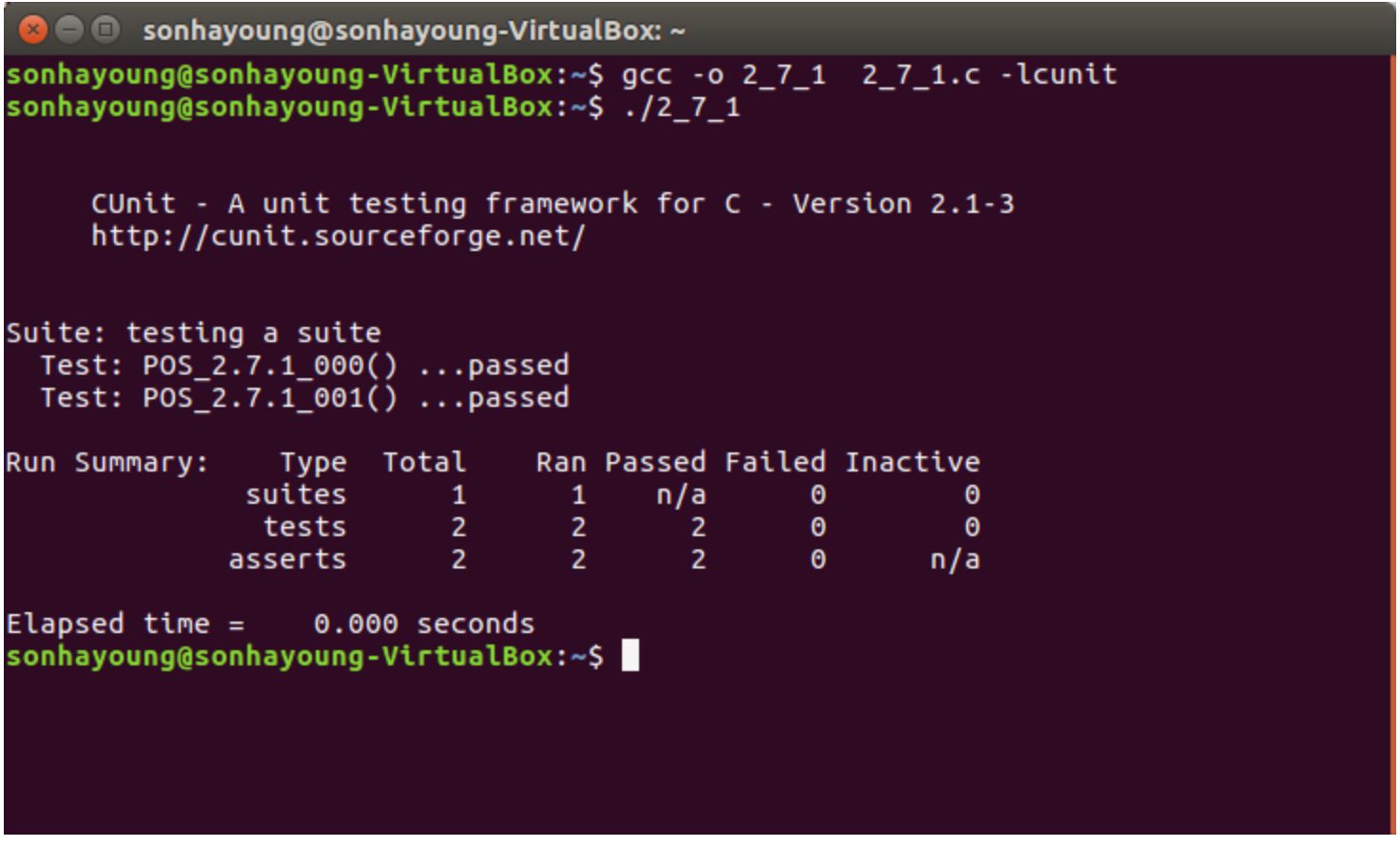
5-1. Cunit Result (2.6.1)



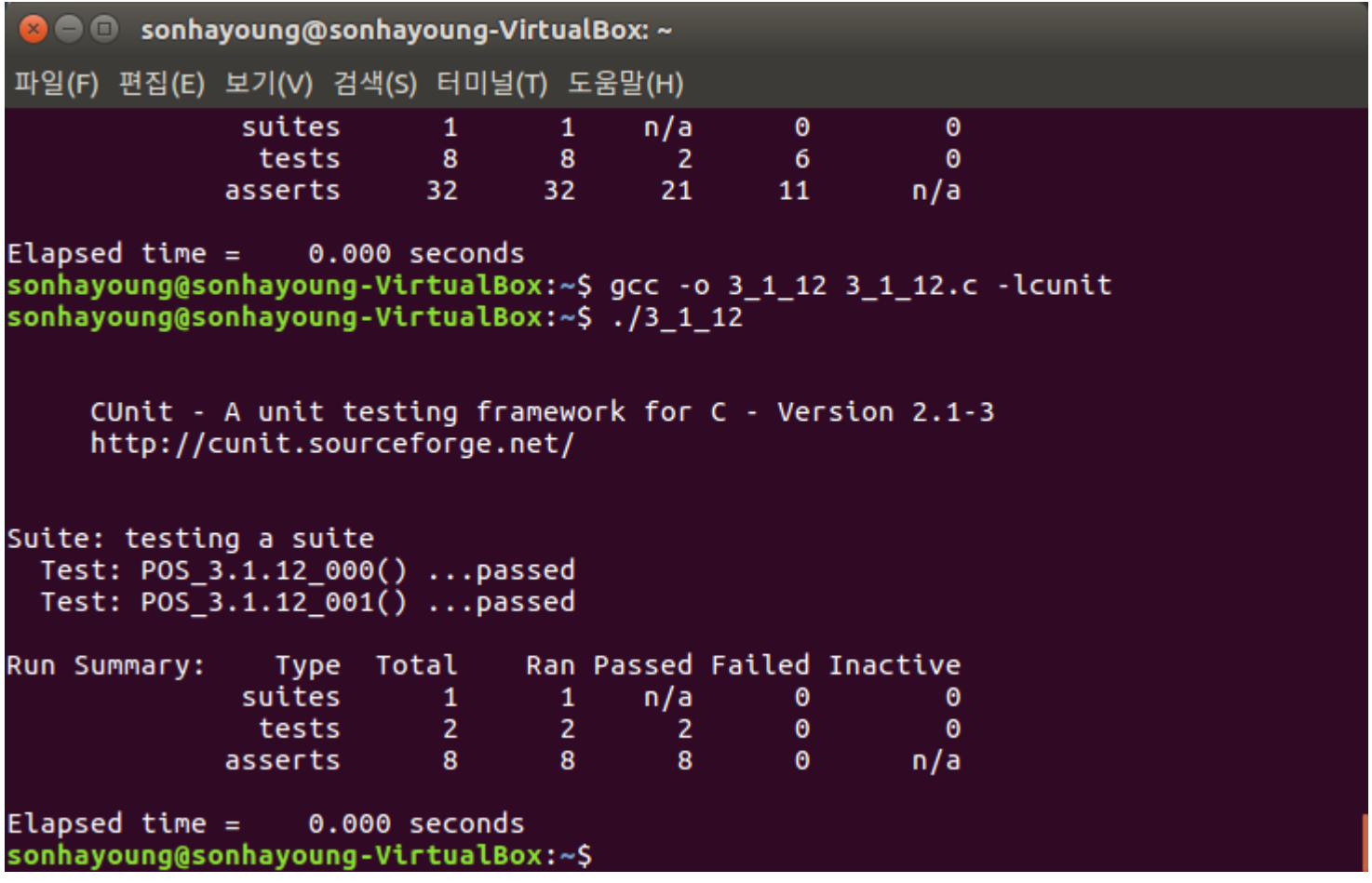
5-1. Cunit Result (2.6.2)

```
sonhayoung@sonhayoung-VirtualBox: ~
              tests      3      3      2      1        0
            asserts      6      6      5      1      n/a

Elapsed time =    0.000 seconds
sonhayoung@sonhayoung-VirtualBox:~$ gcc -o 2_6_2 2_6_2.c -lcunit
sonhayoung@sonhayoung-VirtualBox:~$ ./2_6_2


     CUnit - A unit testing framework for C - Version 2.1-3
     http://cunit.sourceforge.net/


Suite: testing a suite
  Test: POS_2.6.2_000() ...passed
  Test: POS_2.6.2_001() ...passed
  Test: POS_2.6.2_002() ...passed

Run Summary:    Type  Total    Ran Passed Failed Inactive
              suites      1      1    n/a      0        0
              tests      3      3      3      0        0
            asserts      6      6      6      0      n/a

Elapsed time =    0.000 seconds
sonhayoung@sonhayoung-VirtualBox:~$
```

5-1. Cunit Result (2.7.1)



5-1. Cunit Result (3.1.12)



5-2. Display Result (3.1.4)

```
:1


Casher Screen
Display
Total Sale Money                5000
Receive Money           1000
Send Money              2000


Client Screen
Display
Total Sale Money                5000
Receive Money           1000
Send Money              2000
```

```
:2


Casher Screen
Display
Total Sale Money                5000
Receive Money           1000
Send Money              2000
snack           quantity : 1 / price : 1000 / sum_price : 1000


Client Screen
Display
Total Sale Money                5000
Receive Money           1000
Send Money              2000
```

5-2. Display Result (3.1.5)

```
:3


Casher Screen
Display
```

```
Casher Screen
Display
snack           quantity : 1 / price : 1000
```

5-2. Display Result (3.1.6)

```
:5


Casher Screen
Display
Date            2017.11.06 12:07
Total Refund Money              4500


Client Screen
Display
Date            2017.11.06 12:07
Total Refund Money              4500
```

```
:6


Casher Screen
Display
Date            2017.11.06 12:07
Total Refund Money              4500
snack           quantity : 3000 / price : 1 / sum_price : 3000


Client Screen
Display
Date            2017.11.06 12:07
Total Refund Money              4500
```

5-2. Display Result (3.1.7)

```
:7
make_stock_file part
20171106_product.txtDate : 2017.11.06 (년 .월 .일 )
Sale Product
Name, Price, Quantity
snack, 1000, 0
```

```
:8
make_stock_file part
20171106_product.txtDate : 2017.11.06 (년 .월 .일 )
Sale Product
Name, Price, Quantity
snack, 1000, 0
sale_list_file part
20171106_sale_management.txtReceipt Num : 2017.11.06.12.07 (년 .월 .일 .시 .분 )
Date : 2017.11.06 (년 .월 .일 )
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 2, 2000
Total Sale price : 3500
```

```
:9
make_stock_file part
20171106_product.txtDate : 2017.11.06 (년 .월 .일 )
Sale Product
Name, Price, Quantity
snack, 1000, 1
sale_list_file part
20171106_sale_management.txtReceipt Num : 2017.11.06.12.07 (년 .월 .일 .시 .분 )
Date : 2017.11.06 (년 .월 .일 )
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 2, 2000
Total Sale price : 3500
```

5-2. Display Result (3.1.8)

```
:10
sale_201711061215.txtReceipt Num : 2017.11.06.12.15 (년.월.일.시.분)
Date : 2017.11.06 (년.월.일)
Sale Product
Total Sale Price : 5000
```

```
:11
sale_201711061215.txtReceipt Num : 2017.11.06.12.15 (년.월.일.시.분)
Date : 2017.11.06 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Sale Price : 5000
```

5-2. Display Result (3.1.9)

```
:12
refund_201711061217.txtReceipt Num : 2017.11.06.12.17 (년 .월 .일 .시 .분 )
Date : 2017.11.06 (년 .월 .일 )
Refund Product
Total Refund Price : 0
```

```
:13
refund_201711061217.txtReceipt Num : 2017.11.06.12.17 (년 .월 .일 .시 .분 )
Date : 2017.11.06 (년 .월 .일 )
Refund Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Refund Price : 1000
```

5-2. Display Result (3.1.10)

```
:14
stock_20171106.txtDate : 2017.11.06.12.19 (년 .월 .일 .시 .분 )
Name, Price, Quantity
snack, 1000, 4
```

5-2. Display Result (3.1.11)

```
:15
settle_20171106.txtDate : 2017.11.06 (년.월.일)
Sale Product
Total Sale Price : 2000

Refund Product
Total Refund Price : 1000
```

```
:16
settle_20171106.txtDate : 2017.11.06 (년.월.일)
Sale Product
Total Sale Price : 2000

Refund Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Refund Price : 1000
```

```
:17
settle_20171106.txtDate : 2017.11.06 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Sale Price : 2000

Refund Product
Total Refund Price : 1000
```

```
:18
settle_20171106.txtDate : 2017.11.06 (년.월.일)
Sale Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Sale Price : 2000

Refund Product
Name, Price, Quantity, Sum_price
snack, 1000, 1, 1000
Total Refund Price : 1000
```

Q & A