Unit Testing Plan

for Point of Sale System

- **Test Plan**
- **Test Design Specification**
- **Test Cases Specification**

Project Team

Team 4

Date

2017-11-04

Team Information
산업공학과 201211178 민경훈
산업공학과 201211187 배승현
컴퓨터공학과 201311283 송형선
컴퓨터공학과 201611299 정희승

Table of Contents

1 Introduction 4

1.1 Objectives 4

1.2 Background 4

1.3 Scope 4

1.4 Project plan 4

1.5 Configuration management plan 4

1.6 References 4

2 Test items 4

3 Features to be tested 5

4 Features not to be tested 5

5 Approach 6

6 Item pass/fail criteria 6

7 Unit test design specification 6

7.1 Test design specification identifier 6

7.2 Features to be tested 6

7.3 Approach refinements 6

7.4 Test identification 6

7.5 Feature pass/fail criteria 7

8 Unit test case specification 7

8.1 Test case specification identifier 7

8.2 Test items 12

8.3 Input specifications 12

8.4 Output specifications 12

9 Testing tasks 12

10 Environmental needs 12

11 Unit Test deliverables 13

12 Schedules 13

1 Introduction

1.1 Objectives

본 문서는 2017학년도 2학기 소프트웨어 공학 개론 수업의 Team 4가 개발한 Point of Sale System을 Unit Testing하기 위한 계획 문서이다. Team 4가 정의한 Unit Testing을 수행하기 위하여 Testing Pass/Fail Criteria를 정의하고 이를 수행하기 위한 Test Design & Test Cases를 제작한다.

1.2 Background

POS System은 사용자의 요청에 의해 판매, 환불, 재고확인 및 종료를 수행한다.
Unit Test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 Test이며, 시스템의 성 능을 좌우하는 요소들이 요구사항을 만족하는지를 확인할 수 있는 기본적인 Test approach이다

1.3 Scope

물품의 처리에 대한 Unit Test를 수행하기 위한 자원과 절차, Test approach와 Technique와 필요로 하는 환경 및 도구 등을 정의한다. Unit Test는 시스템 핵심 기능 관련 프로세스에 중점을 두며 전달 역할, 스크린, 버튼 인터페이스 등 단순 프로세스는 Test에서 제외한다

1.4 Project plan

1.5 Configuration management plan

1.6 References

[2017SE_B][TP#1]SRA_Team4_rev3
[2017SE_B][TP#2]SDS_Team4_rev2

2 Test items

Team 4가 SASD 기법을 이용하여 개발한 POS System을 Testing한다. SA에서 최소단위의 각 Process별로 요구사항을 만족하는지, 정상적인 결과가 나오는지, 잘못된 값 입력 시 예 외 처리가 동작하는지 Testing을 수행한다. **Figure 1 Overall DFD**는 SA를 이용하여 요구사항을 분석 한 결과를 DFD를 이용해 나타낸 그림이고, **Figure 2 Structural Chart**는 SD의 Structural Chart를 나타낸 그림이다. 각 그림을 참조하여 Unit을 지정하고, 지정한 Unit을 SRA에 명세 된 내용과 같은 동작을 하는지 확인한다

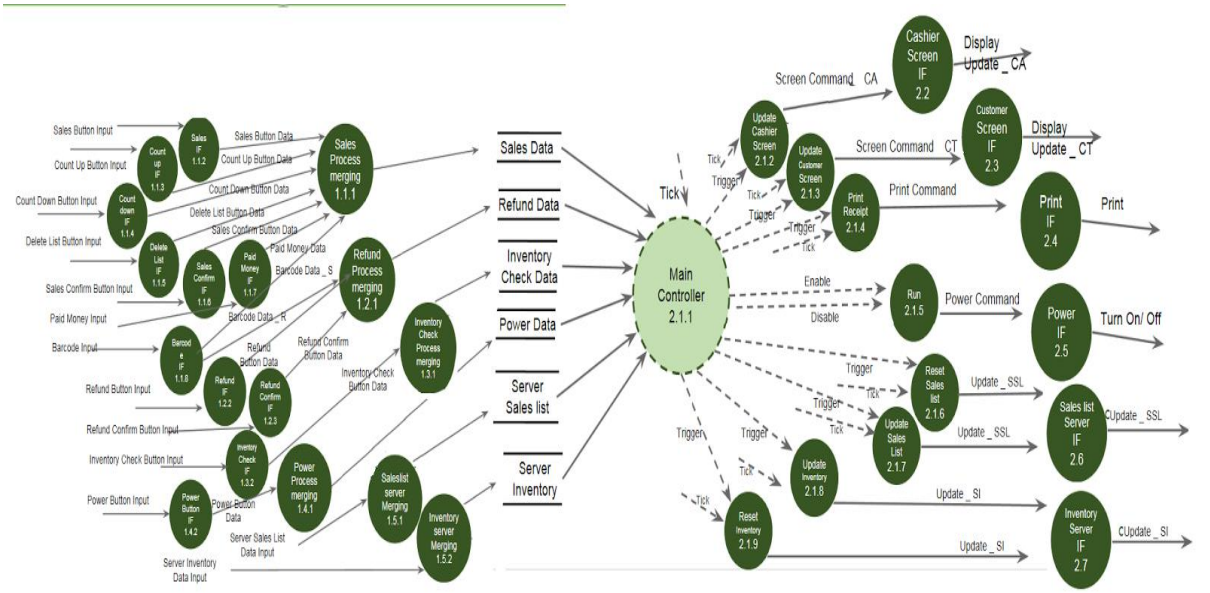

Figure 1 Overall DFD

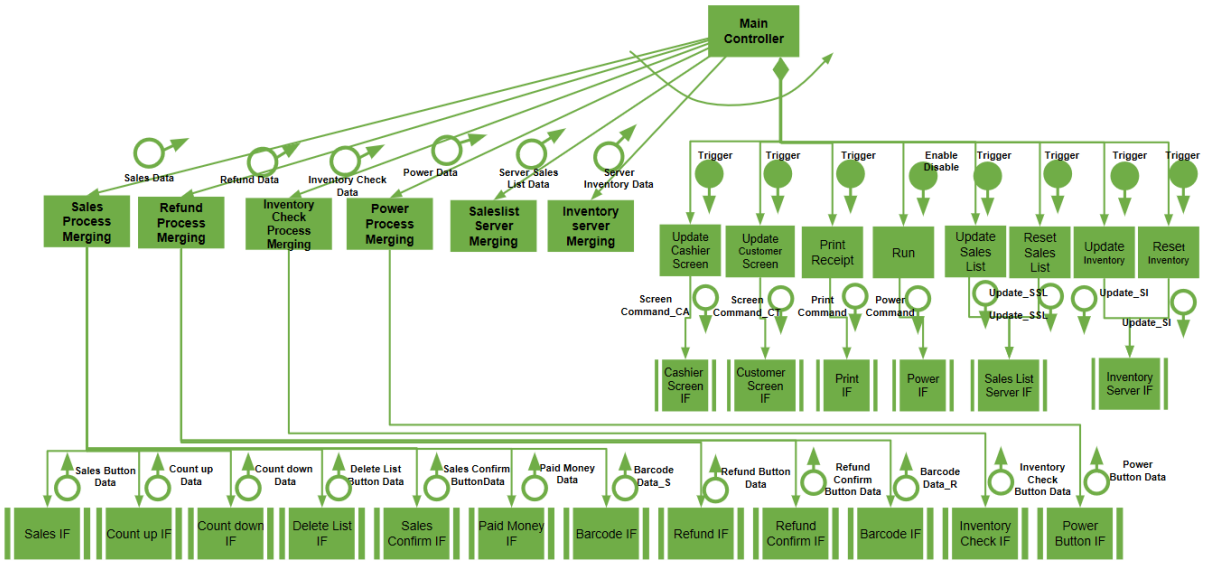

Figure 2 Structure Chart

3 Features to be tested

시스템 핵심 기능 관련 모듈 중 입력 담당 모듈 부분과 처리 담당 모듈 부분을 거쳐 요구사항 만족, 정상적인 동작과 잘못된 값 입력 시 동작에 중점을 두어 테스트 한다.

1) Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는지 테스트한다.
2) Modules in SDS : 각 모듈이 가지고 있는 데이터 인터페이스를 테스트 한다.

4 Features not to be tested

전달 역할, 단순한 프로세스, UI 구성 프로세스, 특수한 상황에서만 작동하는 프로세스는 테스

트에서 제외한다.

5 Approach

POS System의 Program source code와 Unit Tests는 Cygwin Linux Terminal 환경에서 이루어지며, Program code의 변경 및 수정사항은 지속적으로 통합되고 테스트 된다.

6 Item pass/fail criteria

Identifier	Feature
POS.UTC.111	SALE = TRUE/False
POS.UTC.121	REFUND = TRUE/False
POS.UTC.131	CHECK = TRUE/False
POS.UTC.141	POWER_OFF = True/False
POS.UTC.151	Saleslist_server_Merging = 0 or True
POS.UTC.152	Inventory_server_Merging = 0 or 1
POS.UTC.211	통합적인 부분(전달 프로세스)이므로 Test 제외
POS.UTC.212	Update_Cashier_Screen=0 or 1
POS.UTC.213	Update_Cashier_Screen=0 or 1
POS.UTC.214	Print_Receipt=0 or 1
POS.UTC.215	Run = 0 or 1
POS.UTC.216	Reset_Saleslist = 0 or 1
POS.UTC.217	Update_Sales_List = 0 or 1
POS.UTC.218	Update_Inventory = 0 or 1
POS.UTC.219	Reset_Inventory = 0 or 1

7 Unit test design specification

7.1 Test design specification identifier

7.2 Features to be tested

7.3 Approach refinements

각 Process Specification에 명시된 내용을 기반으로 Test Design 및 Test Cases를 생성해 낸다.

7.4 Test identification

Identifier	Feature
POS.UTC.111	Sale Process 가 흐르는지 확인
POS.UTC.121	Refund Process 가 흐르는지 확인
POS.UTC.131	Inventory Check Process가 흐르는지 확인
POS.UTC.141	Power Process가 흐르는지 확인
POS.UTC.151	SalesList Merging 되는지 확인
POS.UTC.152	Inventory Merging 되는지 확인
POS.UTC.211	Main Controller가 제대로 수행되는지 확인
POS.UTC.212	Update_Cashier_Screen가 수행 되어지는지 확인
POS.UTC.213	Update_Customer_Screen가 수행 되어지는지 확인
POS.UTC.214	Print_Receipt가 수행 되어지는지 확인
POS.UTC.215	Run이 작동하는지 확인. 잘못된 데이터시 Run =0
POS.UTC.216	Reset_Saleslist가 수행 되어지는지 확인
POS.UTC.217	Update_Sales_List가 수행 되어지는지 확인
POS.UTC.218	Update_Inventory가 수행 되어지는지 확인
POS.UTC.219	Reset_Inventory가 수행 되어지는지 확인

Table 1 Test Design Identification

7.5 Feature pass/fail criteria

POS의 각 모듈은 SRA에 정의되어 있는 요구사항을 만족해야 한다. 각 모듈의 입/출력 및 동작은 SRA의 Process description 항목 및 STD를 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

Identifier	Feature	Pass / Fail Criteria
POS.UTC.111	Sale Process 가 흐르는지 확인	SALE = TRUE/False
POS.UTC.111.00	데이터를 전달하는 모듈이므로 테스트 하지 않음	
POS.UTC.121	Refund Process 가 흐르는지 확인	REFUND = TRUE/False
POS.UTC.121.00	데이터를 전달하는 모듈이므로 테스트 하지 않음	
POS.UTC.131	Inventory Check Process 가 흐르는지 확인	CHECK = TRUE/False

POS.UTC.131.00	데이터를 전달하는 모듈이므로 테스트 하지 않음	
POS.UTC.141	Power Process 가 흐르는지 확인	POWER_OFF = True/False
POS.UTC.141.00	데이터를 전달하는 모듈이므로 테스트 하지 않음	
POS.UTC.151	SalesList Merging 되는지 확인	Saleslist_server_Merging = 0 or True
POS.UTC.151.00	input: char saleFileTestServer[] = "test_sale_management.txt"; expected : 0	Saleslist_server_Merging = 0
POS.UTC.151.01	input: char saleFileTestServer[] = "test_sale_management.txt"; char receipt_code[20] = "2017.11.04.09.36"; SoldData * sold_data = (SoldData *)malloc(sizeof(SoldData) * total_item); strcpy((sold_data)->item, "과자"); (sold_data)->money = 1000; (sold_data)->quantity = 2; ((sold_data)->pay) = 2000; expected : 1	Saleslist_server_Merging = 1
POS.UTC.151.02	input: char saleFileTestServer[] = "test_sale_management.txt"; char receipt_code[20] = "2017.11.04.09.36"; SoldData * sold_data = (SoldData *)malloc(sizeof(SoldData) * total_item); strcpy((sold_data)->item, "과자"); (sold_data)->money = 1000; (sold_data)->quantity = 2; ((sold_data)->pay) = 2000;	Saleslist_server_Merging = 2

	strcpy((sold_data+1)->item, "물"); (sold_data+1)->money = 500; (sold_data+1)->quantity = 2; ((sold_data+1)->pay) = 1000; expected : 2	
POS.UTC.152	Inventory Merging 되는지 확인	Inventory_server_Merging = 0 or 1
POS.UTC.152.00	input: char productTestServer[] = "test_product.txt"; ItemData* itd = (ItemData*)malloc(sizeof(ItemData)); expected : 0	Inventory_server_Merging = 0
POS.UTC.152.01	input: char productTestServer[] = "test_product.txt"; ItemData* itd = (ItemData*)malloc(sizeof(ItemData) * 8); expected : 1	Inventory_server_Merging
POS.UTC.211	Main Controller 가 제대로 수행되는지 확인	통합적인 부분(전달 프로세스)이므로 Test 제외
POS.UTC.211.00	데이터를 전달하는 모듈이므로 테스트 하지 않음	
POS.UTC.212	Update_Cashier_Screen 가 수행 되어지는지 확인	Update_Cashier_Screen =0 or 1
POS.UTC.212.00	input : char ca[1000]; sprintf(ca,""); expected : 0	Update_Cashier_Screen =0
POS.UTC.212.01	input : char ca[1000]; sprintf(ca,"₩n%-15s %-15s %-15s₩n",	Update_Cashier_Screen = 1

	"상품", "수량", "가격"); expected : 1	
POS.UTC.213	Update_Customer_Screen 가 수행 되어지는지 확인	Update_Customer_Screen=0 or 1
POS.UTC.213.00	input: char ct[1000]; sprintf(ct,""); expected : 0	Update_Customer_Screen = 0
POS.UTC.213.01	input: char ct[1000]; sprintf(ct,"₩n 판매 날짜 : %d 년 %d 월 %d 일 %d 시 %d 분₩n", 2017, 1, 1, 1, 1); expected: 1	Update_Customer_Screen = 1
POS.UTC.214	Print_Receipt 가 수행 되어지는지 확인	Print_Receipt=0 or 1
POS.UTC.214.00	input: char testReceipt[] = "stock_test.txt"; char str[1000]; sprintf(str, ""); expected: 0	Print_Receipt = 0
POS.UTC.214.01	input: char testReceipt[] = "stock_test.txt"; char str[1000]; sprintf(str, "%-15s %-15s %-15s₩n", "상품", "단가", "수량"); expected: 1	Print_Receipt = 1
POS.UTC.215	Run 이 작동하는지 확인. 잘못된 데이터시 Run =0	Run = 0 or 1
POS.UTC.215.00	Main 과 마찬가지로 데이터를 전달하는 모듈이므로 테스트 하지 않음	
POS.UTC.216	Reset_Saleslist 가 수행 되어지는지 확인	Reset_Saleslist = 0 or 1
POS.UTC.216.00	input: char saleFileTestServer[] =	Reset_Saleslist = 0

	"test_sale_management.txt"; char receipt_code[20] = "2017.11.04.01.28"; SoldData * sold_data = (SoldData *)malloc(sizeof(SoldData) * (1+1)); strcpy((sold_data)->item, "물"); (sold_data)->money = 500; (sold_data)->quantity = 2; ((sold_data)->pay) = 1000; len = 0 expected: 0	
POS.UTC.216. 01	input: char saleFileTestServer[] = "test_sale_management.txt"; char receipt_code[20] = "2017.11.04.01.28"; SoldData * sold_data = (SoldData *)malloc(sizeof(SoldData) * (1+1)); strcpy((sold_data)->item, "물"); (sold_data)->money = 500; (sold_data)->quantity = 2; ((sold_data)->pay) = 1000; len = 1 expected: 1	Reset_Saleslist = 1
POS.UTC.217	Update_Sales_List 가 수행 되어지는지 확인	Update_Sales_List = 0 or 1
POS.UTC.217. 00	input: char productTestServer[] = "test_product.txt"; len = 0; expected: 0	Update_Sales_List = 0
POS.UTC.217. 01	input: char productTestServer[] = "test_product.txt"; len = 1;	Update_Sales_List = 1

	expected: 1	
POS.UTC.218	Update_Inventory 가 수행 되어지는지 확인	Update_Inventory = 0 or 1
POS.UTC.218.00	input: char productTestServer[] = "test_product.txt"; len = 0; expected: 0	Update_Inventory = 0
POS.UTC.218.01	input: char productTestServer[] = "test_product.txt"; len = 1; expected: 1	Update_Inventory = 1
POS.UTC.219	Reset_Inventory 가 수행 되어지는지 확인	Reset_Inventory = 0 or 1
POS.UTC.219.00	input: char productTestServer[] = ""; expected: 0	Reset_Inventory = 0
POS.UTC.219.01	input: char productTestServer[] = "reset_product.txt"; expected: 1	Reset_Inventory = 1

Table 2 Test Case Identification

8.2 Test items

Table 2 Test Case Identification 참고

8.3 Input specifications

Table 2 Test Case Identification 참고

8.4 Output specifications

Table 2 Test Case Identification 참고

9 Testing tasks

10 Environmental needs

Linux (Cygwin)

C Language Compiler/Linker(GCC)

11 Unit Test deliverables

12 Schedules