

# Unit Testing Plan for POS System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

**1 Team**

Date

**2017-11-04**

---

**Team Information**

201311272 나경수

201411276 서수빈

201411313 장진서

201611293 전다운

## 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.....	7
4	Features not to be tested .....	7
5	Approach.....	8
6	Item pass/fail criteria .....	8
7	Unit test design specification.....	8
7.1	Test design specification identifier .....	8
7.2	Features to be tested .....	8
7.3	Approach refinements.....	9
7.4	Test identification .....	9
7.5	Feature pass/fail criteria .....	9
8	Unit test case specification.....	9
8.1	Test case specification identifier .....	9
8.2	Test items .....	11

8.3	Input specifications.....	12
8.4	Output specifications.....	12
9	Testing tasks .....	12
10	Environmental needs .....	12
11	Unit Test deliverables.....	12
12	Schedules .....	12

## 1 Introduction

### 1.1 Objectives

본 문서는 2017년 건국대학교의 소프트웨어공학 개론 강의의 실습과제를 설명한다. 실습과제는 Point Of Sale (POS) System을 소프트웨어만을 이용한 가상의 시스템으로 구현 하는 것이다

### 1.2 Background

2017년 2학기 소프트웨어공학개론 수업에서 개발하는 SASD기법을 이용한다. Test가 필요한 기능끼리 나누어 C-unit test를 한다.

C unit test는 c를 기반으로 한 시스템에서 최소 모듈단위로 test를 하여 시스템이 제대로 작동하는지 test한다.

### 1.3 Scope

POS System이란 판매와 관련한 데이터를 일괄적으로 관리하고, 고객정보를 수집하여 부가가치를 향상시키는 시스템이다. 본 프로젝트는 전체 POS System 중 POS 단말기만을 대상으로 구현하는 것으로 규모를 제한한다

### 1.4 Project plan

### 1.5 Configuration management plan

POS System의 unit test를 위한 test는 Cygwin환경에서 이루어진다.

### 1.6 References

[2017SE\_B][TP#1]SRA\_Team1(2.0).docx

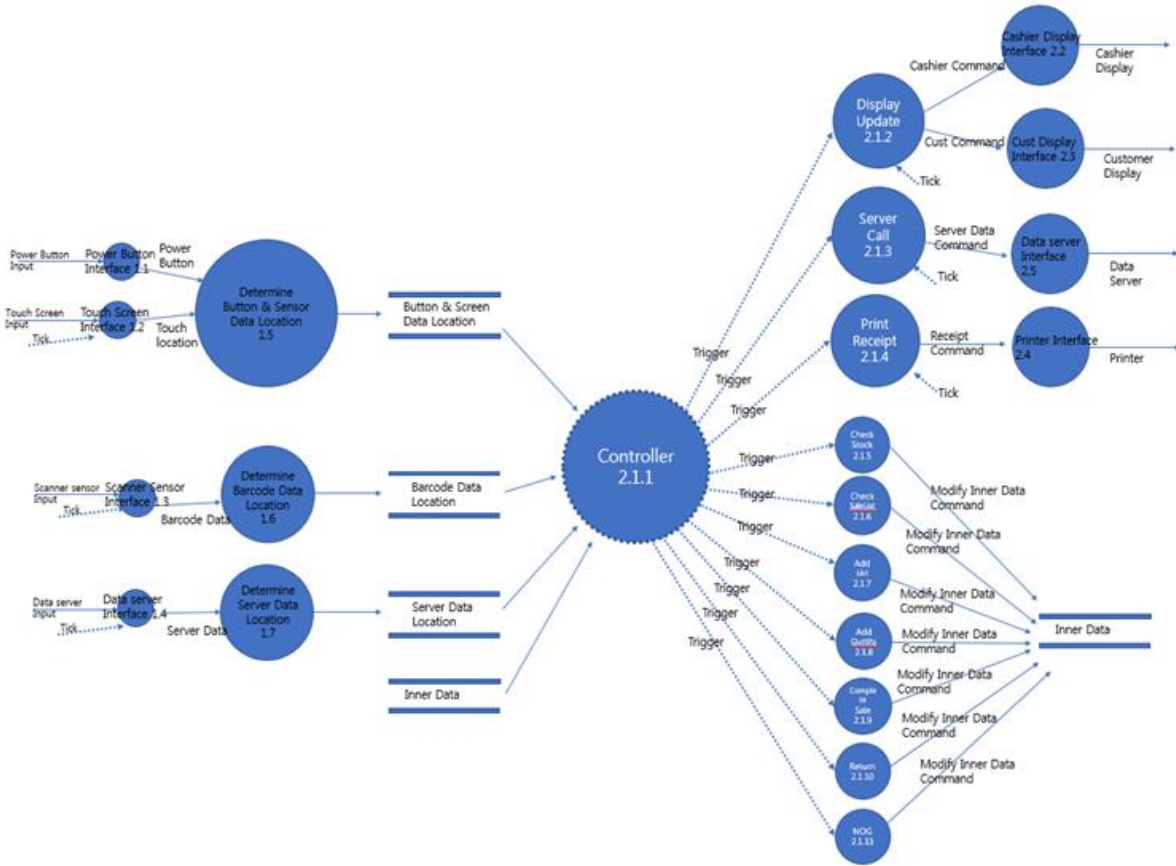
[2017SE\_B][TP#2]SDS\_Team1(2.0).docx

## 2 Test items

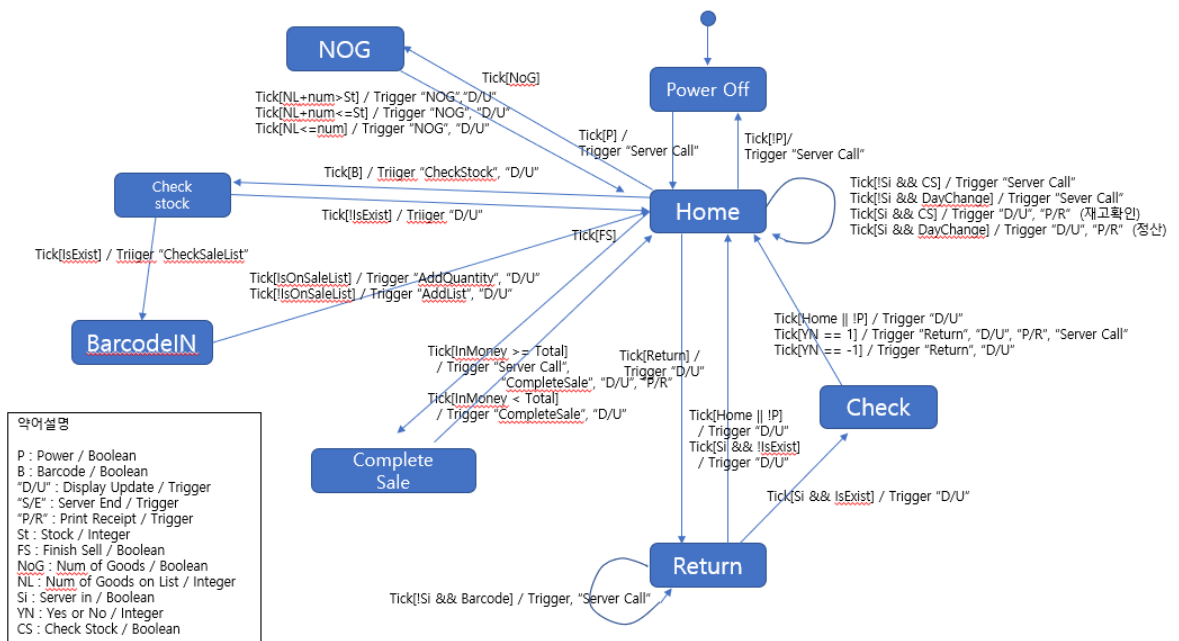
POS system의 모든 최소 모듈들이 test대상이다.

Test item은 아래의 DFD , State Transition Diagram , Structured Charts을 통해 작성되었다.

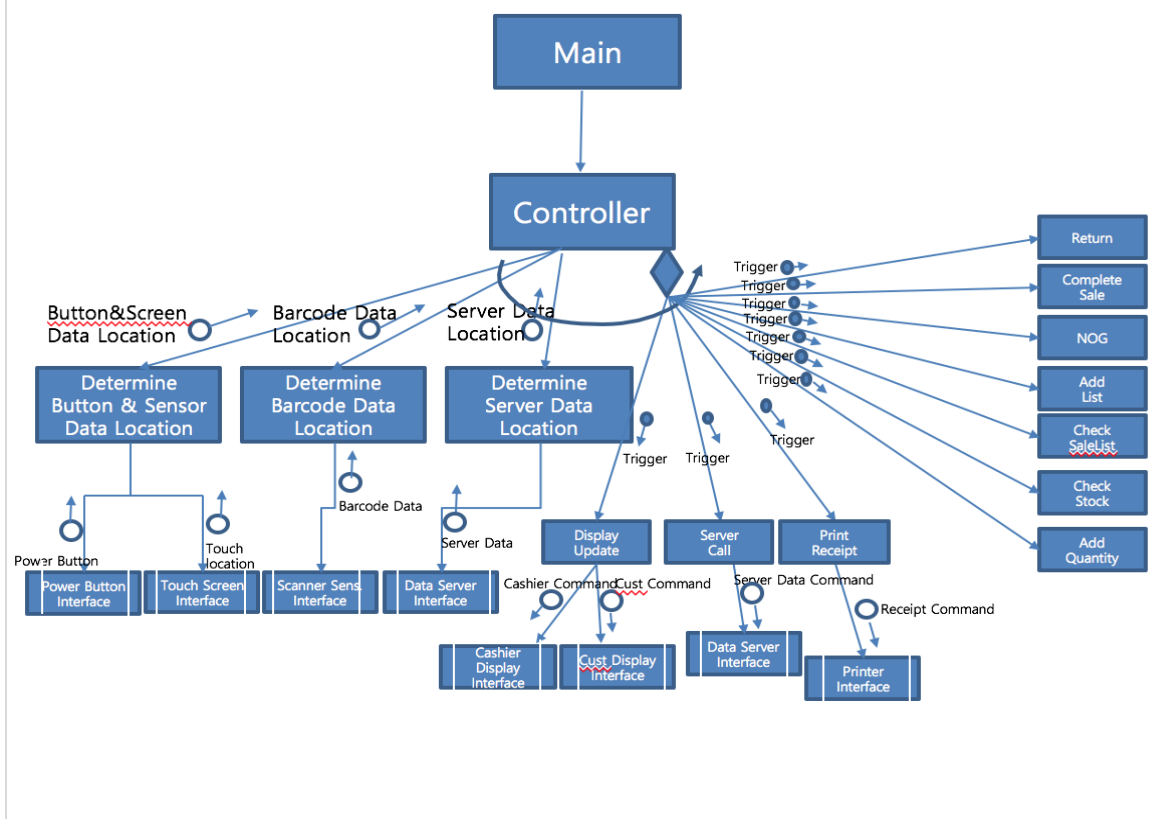
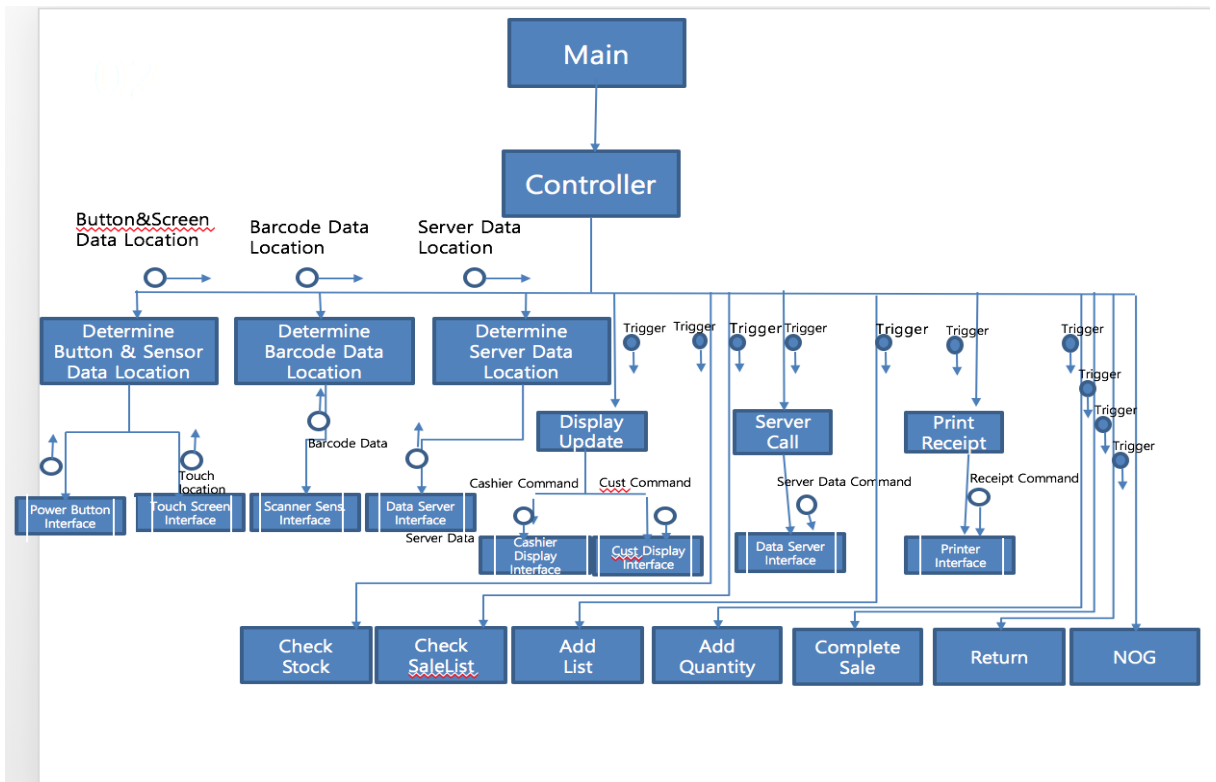
(1) POS System Overall DFD



(2) POS System State Transition Diagram



(3) POS System Structured Charts



## 3 Features to be tested

ID	Name	Description
2.1.5	Check Stock	남은 재고 수량을 check 하고 Inner Data를 바꾼다.
2.1.6	CheckSaleList	찍은 바코드가 현재 판매목록에 있는지 check하고 Inner Data를 바꾼다.
2.1.7	AddList	새로운 제품이 찍히면 현재 판매 목록에 추가시킨다. Inner Data에 판매 목록 바꾸기.
2.1.8	AddQuantity	현재 판매 목록의 제품 수량을 증감시키기 위해 Inner Data를 바꾼다.
2.1.9	CompleteSale	판매 완료를 하기위해 현재 판매목록을 지우고 재고 수량을 고친다.
2.1.10	Return	환불 기능을 하기 위해 Inner Data의 재고 수량을 바꾼다.
2.1.11	NOG	현재 판매목록의 제품 수량을 입력된 수 만큼 증감시키기 위해 Inner Data를 바꾼다.

## 4 Features not to be tested

ID	Name	Description
1.1	Power Button Interface	"Power Button Input" 과정은 아날로그 신호를 읽어 True/False와 같은 디지털 신호로 바꿔준다. 그 신호를 "Power Button"로 Output 한다.
1.2	Touch Screen Interface	"Touch Screen Input" 과정은 캐시의 터치스크린에서 받은 위치를 디지털 신호를 주기적으로 읽고, 읽은 위치를 Touch location으로 보낸다
1.3	Scanner Sensor Interface	"Scanner Sensor Input" 과정은 Scanner에서 받은 디지털 신호를 주기적으로 읽고, 디지털 신호를 String으로 변환해 "Barcode Data"으로 Output 한다.
1.4	Data Server Interface	"Data Server Input" 과정은 Data Sever에서 보낸 Sever Data 를 보낸다.
1.5	Determine Button & Sensor Data Location	Power Button 에서 True/False 정보를 받아 Button Data Location으로 보낸다. Touch Home, Touch Sale , Touch Return , Touch Stock , Touch Total 에서 True/False 정보를 받아 Sensor Data Location로 보낸다. Touch Check, Touch NoG , Touch In-Money에서 Integer정보를 받아 Sensor Data Location로 보낸다

1.6	Determine Barcode Data Location	Barcode Data에서 String정보를 받아 Barcode Data Location로 보낸다.
1.7	Determine Server Data Location	Server Data에서 File정보를 받아 Server Data Location로 보낸다.
1.8	Cashier Display Interface	Cashier Command 를 받아 Cahier Display (터치스크린)의 화면을 바꾼다.
1.9	Cust Display Interface	Cust Command 를 받아 Customer Display (터치스크린)의 화면을 바꾼다.
1.10	Printer Interface	Receipt Command 를 받아 Printer가 영수증을 출력하도록 한다.
1.11	Data Server Interface	Server Data Command 를 받아 Data Server에 보낸 정보를 추가/ 삭제 하도록 한다.
1.12	Display Update	Display Update(D/U) 가 발생하면 cashier display 또는 Cust display를 바꾸는 command 가 발생한다.
1.13	Server Call	Server Data Command를 통해 판매완료 시, Receipt Data, Remain Data 동기화 요청을 하고, 환불을 원할 시, 환불가능 여부와 상품정보 Data를 요청하며, 시간 정보를 원할 시, 시간정보에 대한 요청을 Server에 보낸다.
1.14	Print Receipt	Print명령을 받을 시, Receipt Command Data를 Printer로 전송한다

## 5 Approach

CMS의 Program source code와 Unit Tests는 Cygwin 환경에서 이루어지며, Program code의 변경 및 수정사항은 지속적으로 통합되고 테스트 된다.

## 6 Item pass/fail criteria

Functional Test Pass/Fail Criteria : 각각의 입력에 대해 예상한 결과값과 일치 여부.

## 7 Unit test design specification

### 7.1 Test design specification identifier

T1\_POS.number

### 7.2 Features to be tested



3번의 Features to be tested 참조

### 7.3 Approach refinements

각 모듈의 요구사항을 만족하는 지를 확인하기 위해, 요구사항의 정의된 내용을 바탕으로 하여 Test Code를 작성하며, 각 모듈을 최소한의 Unit으로 나누어서 Unit test를 실시한다

### 7.4 Test identification

Identifier	Feature	Valid/Invalid value
T1_POS_000_000	2.1.5 check stock	valid
T1_POS_000_001	2.1.5 check stock	valid
T1_POS_000_002	2.1.5 check stock	valid
T1_POS_000_003	2.1.5 check stock	invalid
T1_POS_001_000	2.1.8 AddQuantity	valid
T1_POS_001_001	2.1.8 AddQuantity	valid
T1_POS_002_000	2.1.11 NOG	valid
T1_POS_002_001	2.1.11 NOG	invalid
T1_POS_002_002	2.1.11 NOG	invalid
T1_POS_003_000	2.1.9 Complete Sale	valid
T1_POS_003_001	2.1.9 Complete Sale	valid
T1_POS_004_000	2.1.10 Return	valid
T1_POS_004_001	2.1.10 Return	valid
T1_POS_005_000	2.1.6 CheckSaleList	valid
T1_POS_005_001	2.1.6 CheckSaleList	valid
T1_POS_006_000	2.1.7 AddList	valid
T1_POS_006_001	2.1.7 AddList	invalid

### 7.5 Feature pass/fail criteria

입력에 대해 pass/fail이 나오는 지 보고 그 뒤 범위 같은 것은 고려하지 않는다.

## 8 Unit test case specification

### 8.1 Test case specification identifier

Test case Identifier	Input Specification (input // 주어진 환경 )	Output Specification
T1_POS_000_000	stockList, "001", &a, name (재고struct,바코드,제품가격,제품명) // stockList has Item("001") ,Item("010")	True
T1_POS_000_001	stockList, "010", &a, name // 위와 동일환경	True
T1_POS_000_002	stockList, "010", &a, name // 위와 동일환경	True
T1_POS_000_003	stockList, "011", &a, name // 위와 동일환경	False
T1_POS_001_000	saleList, price, "001" (현재판매목록, 상품가격, 바코드) // saleList has Item("001")*2 saleList has Item("010")*1	3
T1_POS_001_001	saleList, price, "010" // 위와 동일환경	2
T1_POS_002_000	1,2, saleList,stockList (current fruit 1, increment currnet fruit by 2) // stockList has Item("fruit")*20 saleList has Item("fruit")*1	3
T1_POS_002_001	1,1000, saleList,stockList // 위와 동일환경	-1

T1_POS_002_002	1,-500, saleList,stockList // 위와 동일환경	0
T1_POS_003_000	saleList, stockList // saleList has Item("001")*1 saleList has Item("001")*1  StockList has Item("001")*100 StockList has Item("010")*100	99
T1_POS_003_001	stockList, "010" // 위와 동일환경	99
T1_POS_004_000	stockList, "001" (product current quantity, product barcode) // StockList has Item("fruit")*100 ReturnList has Item("fruit")*3	103
T1_POS_004_001	stockList, "010" // 위와 동일환경	102
T1_POS_005_000	saleList, "001" (현재판매목록struct, 바코드) SaleList has Item("001")*1 // saleList has Item("fruit")*1	1
T1_POS_005_001	saleList, "002"	-1
T1_POS_006_000	"001" (추가한 상품 바코드"001") // productList has Item("001")*2	True"
T1_POS_006_001	"000" // 위와 동일환경	False

## 8.2 Test items

7.4 Test Design Identification 참조

8.3 Input specifications

8.1 Test Case Identification 참조

8.4 Output specifications

8.1 Test Case Identification 참조

9 Testing tasks

Task	Processor tasks	Special skill	Effort	Finish date
Unit Test Plan 작성	[2017SE_B][TP#1]UTP_Team1작성			2017-11-04
Test design Specification		POS이해		
Test Case Specification		POS 이해		
Test Execution				
Test result report	[2017SE_B][TP#1]UTR_Team1작성			2017-11-04
개발팀에 report전달				

10 Environmental needs

Pc , Cygwin C unit Framework 필요

11 Unit Test deliverables

12 Schedules

1. Test item 모두 구하기
2. Features to be tested, not to be tested 나누기
3. id부여
4. test case input/output type 찾기
5. test case input/output 작성
6. test case code 작성
7. test
8. 에러가 뜨면 에러가 뜬 원인 찾기