

Software Requirement Analysis for Point Of Sale System

Project Team

Team 6

Date

0000-00-00

Team Information

201311264 김병식

201610379 김나연

201611248 강병성

201610070 김지우

Table of Contents

1	Introduction _____	4
1.1	Purpose _____	4
	_____	4
1.2	Scope _____	4
1.3	Definition, acronyms, and abbreviations _____	4
1.4	Reference _____	5
1.5	Overview _____	5
2	Overall Description _____	5
2.1	Product Perspective _____	5
2.2	Product functions _____	5
2.3	User characteristics _____	5
2.4	Constraints _____	6
3	Structured Analysis _____	7
3.1	System Context Diagram _____	7
3.1.1	Basic System Context Diagram _____	7
3.1.2	Event List _____	7
	_____	8
3.1.3	The System Context Diagram _____	8
3.2	Data Flow Diagram _____	8
3.2.1	DFD level 0 _____	8
3.2.1.1	DFD _____	8
	_____	9
3.2.1.2	Process Specification _____	9
	_____	10
3.2.1.3	Data Dictionary _____	10

	_____	11
3.2.2	DFD Level 1 _____	11
3.2.2.1	DFD _____	11
3.2.3	DFD Level 2 _____	11
3.2.3.1	DFD _____	12
3.2.3.2	Process Specification _____	12
3.2.3.2.1	Process 1.1 _____	12
3.2.3.2.2	Process 1.2 _____	13
3.2.3.2.3	Process 1.3 _____	13
3.2.3.2.4	Process 1.4 _____	13
	_____	14
3.2.3.2.5	Process 1.5 _____	14
3.2.3.2.6	Process 2.1 _____	14
3.2.3.2.7	Process 2.2 _____	15
3.2.3.2.7	Process 2.2 _____	16
	_____	16
3.2.3.2.8	Process 2.3 _____	16
	_____	16
3.2.3.2.9	Process 2.4 _____	17
	_____	17
3.2.3.3	Data Dictionary _____	17
	_____	18
3.2.4	DFD Level 3 _____	19
3.2.4.1	DFD _____	19
3.2.4.2	State Transition Diagram _____	19
3.2.5	Overall DFD _____	20

1 Introduction

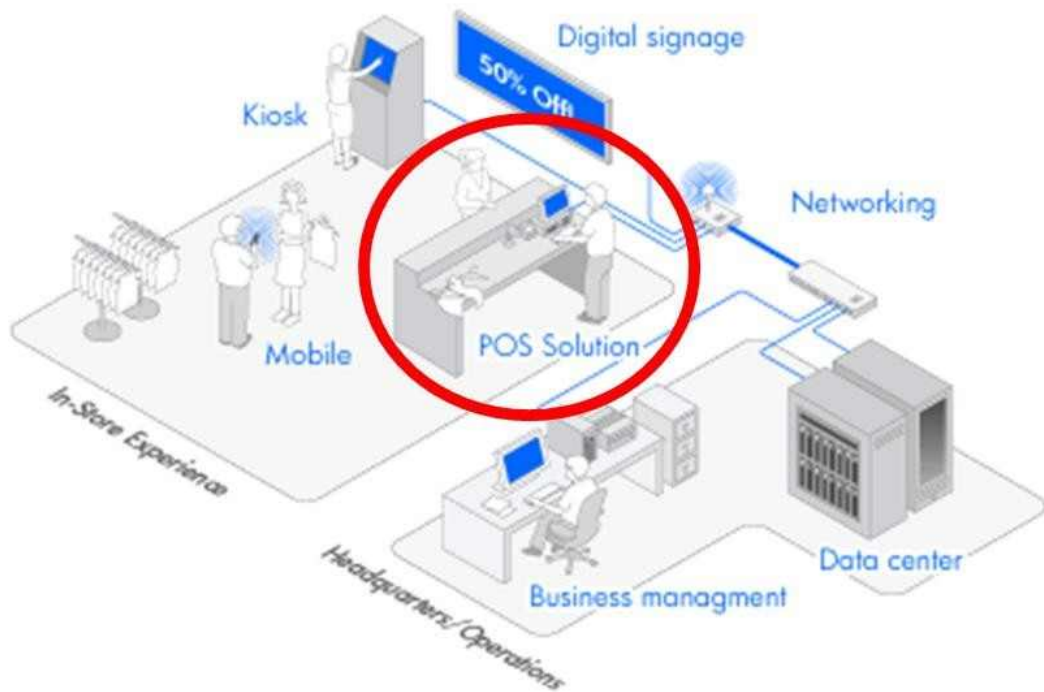
1.1 Purpose

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

1.2 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.

모든 시스템은 SW만으로 구현하고 HW가 필요한 부분은 SW 모듈을 만들어 가상의 HW를 구현한다.



1.3 Definition, acronyms, and abbreviations

HW: Hardware

SW: Software

POS: Point Of Sale

POST: Point Of Sale Terminal

캐셔: Cashier POST 사용자

1.4 Reference

1.5 Overview

2장 개발 대상에 대한 설명; 3장 세부 기능 명세

2 Overall Description

2.1 Product Perspective

SW 로 개발하는 가상의 POS 시스템

2.2 Product functions

POST 는 캐셔 화면(터치 스크린)을 이용해 판매, 환불, 재고 확인을 할 수 있다.

POST 는 바코드 스캐너를 통해 상품 또는 영수증의 바코드를 스캔하여 정보를 입력 받는다.

POST 는 판매 상황에서 상품의 바코드를 바코드 스캐너로 스캔하여 판매 목록에 상품을 추가하고 상품의 수량을 증가시킬 수 있다.

POST 는 판매 상황에서 캐셔 화면(터치 스크린)의 입력을 통해 판매할 상품의 수량을 증가, 감소시키거나 상품을 목록에서 제거할 수 있다.

POST 는 판매 상황에서 고객 화면을 통해 판매할 상품의 총액을 보여 준다.

POST 는 판매를 완료한 후 고객 화면과 캐셔 화면에 각각 판매 정보를 출력한다.

POST 는 판매를 완료한 후 프린터를 이용해 판매 영수증을 출력한다.

POST 는 판매를 완료한 후 재고 서버의 상품 수량 정보를 업데이트 한다.

POST 는 환불 상황에서 영수증의 바코드를 바코드 스캐너로 스캔하여 환불 가능한 영수증인지 판단 할 수 있다.

POST 는 환불을 완료한 후 고객 화면과 캐셔 화면에 각각 환불 정보를 출력한다.

POST 는 환불을 완료한 후 프린터를 이용해 환불 영수증을 출력한다.

POST 는 환불을 완료한 후 재고 서버의 상품 수량 정보를 업데이트 한다.

POST 는 재고 확인 요청이 들어오면 현재 매장의 재고 정보를 화면에 출력한다.

POST 는 매일 24시(자정)에 정산이 이루어지며 재고 서버에 있는 하루 동안 있었던 판매, 환불 정보를 프린터를 이용해 정산 보고서를 출력한다.

2.3 User characteristics

POST 사용자는 캐셔이다.

사용자는 언제든지 전원을 끌 수 있다.

사용자는 바코드 스캐너를 이용해 상품, 영수증을 스캔할 수 있다.

2.4 Constraints

POST의 캐셔 화면(터치 스크린) 입력 및 스캐너 스캔 행위는 키보드 입력으로 대체한다.

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

POST의 프린터를 사용한 모든 출력은 텍스트 파일 생성으로 대체한다.

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

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

- 201709111843

POST 프로그램 종료 후 재실행 시 기존에 진행하던 시각이 이어지도록 한다.

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

- 상품의 정보는 하위 표 참고

상품	단가	바코드
과자	1,000	001
아이스크	1,500	010
과일	3,000	011
물	500	100
라면	800	101
음료수	1,200	110
커피	2,000	111

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

재고 서버는 2개의 텍스트 파일로 대체한다.

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

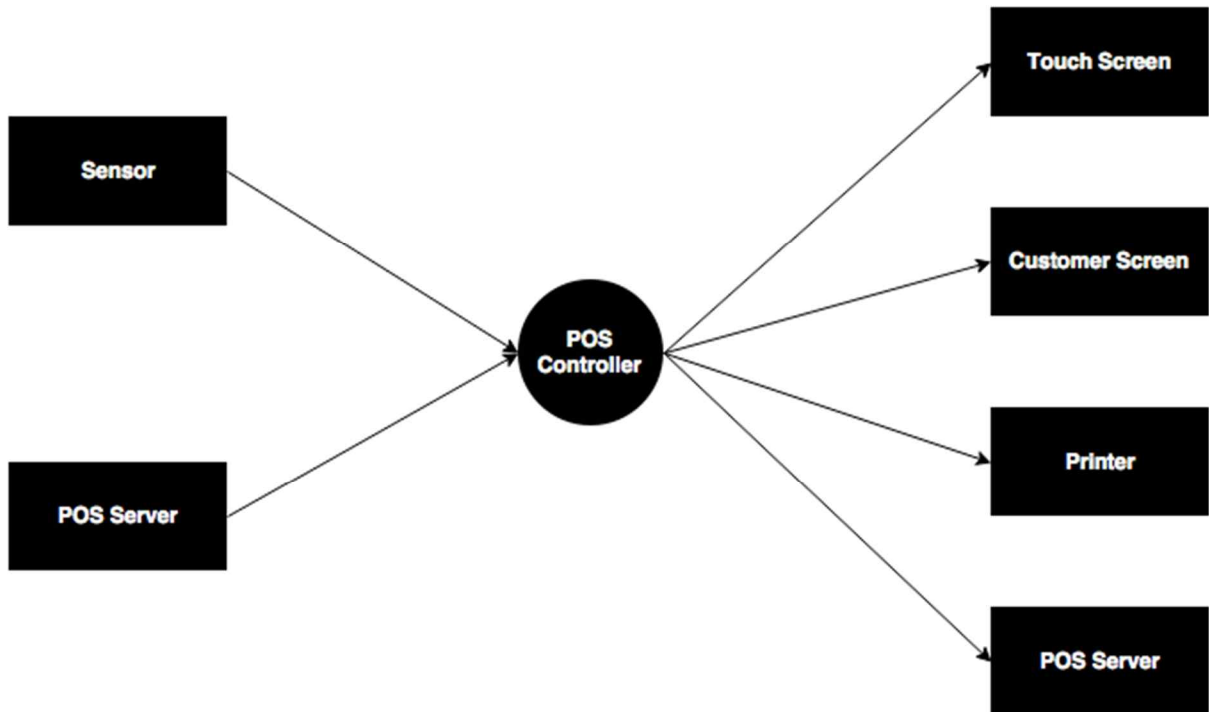
하루가 지나면 재고 서버의 상품 파일, 판매 관리 파일은 다음 날짜로 새로 생성이 되며 상품 파일의 모든 상품은 100개로 초기화한다.

POST의 전원이 켜지면 자동으로 재고 서버와 상품을 동기화한다.

3 Structured Analysis

3.1 System Context Diagram

3.1.1 Basic System Context Diagram

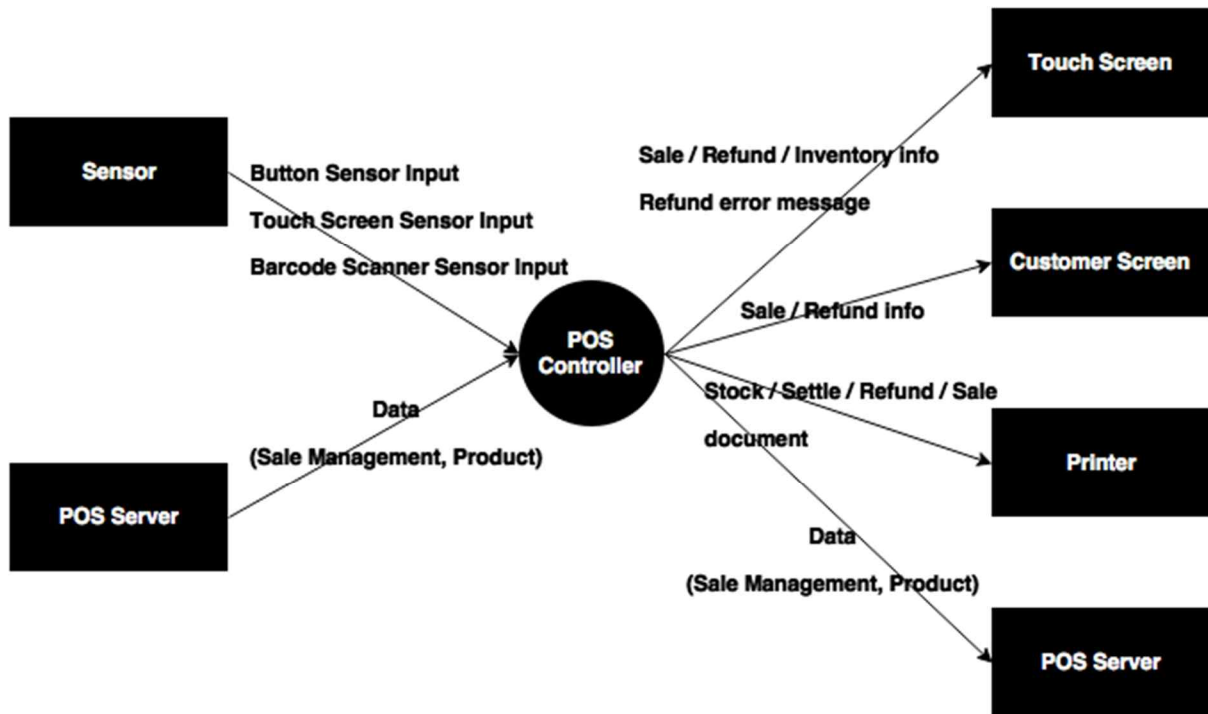


3.1.2 Event List

Input/Output Event	Description
Button Sensor Input	POS 시스템의 전원 신호를 전달한다.
Touch Screen Sensor Input	터치된 화면 좌표를 전달한다.
Barcode Scanner Sensor Input	스캐너로 감지된 상품의 바코드, 영수증 바코드를 전달한다.
Data(Sale Management, Product)	서버로부터 요청된 판매 관리 데이터와 상품 데이터이다.
Current Time	디지털 시계로 시각을 전달한다.
Sale / Refund / Inventory Info	1) 판매 상황 및 판매 완료 상황 실시간으로 업데이트 된 상품 판매 리스트 정보, 판매 영수증에 출력될 정보

	2) 환불 상황 환불 영수증에 출력될 정보 3) 재고 확인 상황 요청된 시점에서의 내부 스토리지 상태 정보
Refund error message	환불 상황에서 예외 발생으로 인해 정상적으로 서비스를 제공하지 못할 때 출력될 메시지이다.
Stock / Settle / Refund / Sale document	재고 확인, 정산, 환불, 판매 완료 상황에서 문서로 출력될 데이터

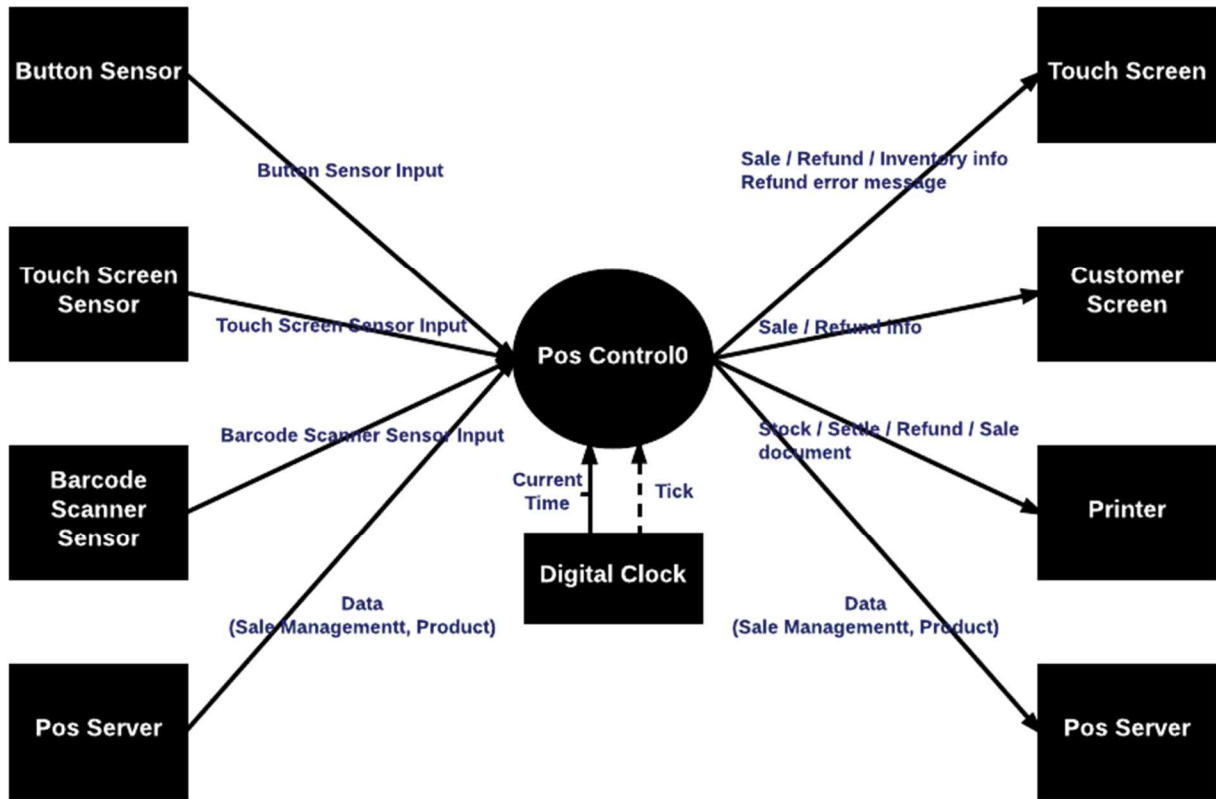
3.1.3 The System Context Diagram



3.2 Data Flow Diagram

3.2.1 DFD level 0

3.2.1.1 DFD



3.2.1.2 Process Specification

Reference No.	0
Name	Pos Control0
Input	Button Sensor Input, Touch Screen Sensor Input, Barcode Scanner Sensor Input, Data(Sale Management, Product)
Output	Sale info, Refund info, Inventory info, Refund error message, Stock document, Settle document, Refund document, Sale document, Data(Sale Management, Product)
Process Description	<p>1) 버튼 센서에서 전달받은 신호에 따라 활성화, 비활성화된다.</p> <p>2) 터치 스크린 센서가 전달한 화면 좌표에 따른 서비스를 제공한다. (화면에 결과를 출력하거나 프린터기로 문서를 출력하는 것 모두 포함)</p>

- 3) 바코드 스캐너 센서에서 전달한 바코드 값을 통해 판매 서비스 수행할 때에는 판매되는 상품을 실시간으로 업데이트하고 환불 상황에서는 영수증 단위로 환불을 수행한다.
- 4) 서버의 데이터로 동기화를 우선적으로 수행한 후 서비스 동작 완료되었을 때 변경된 사항을 서버에 반영한다.

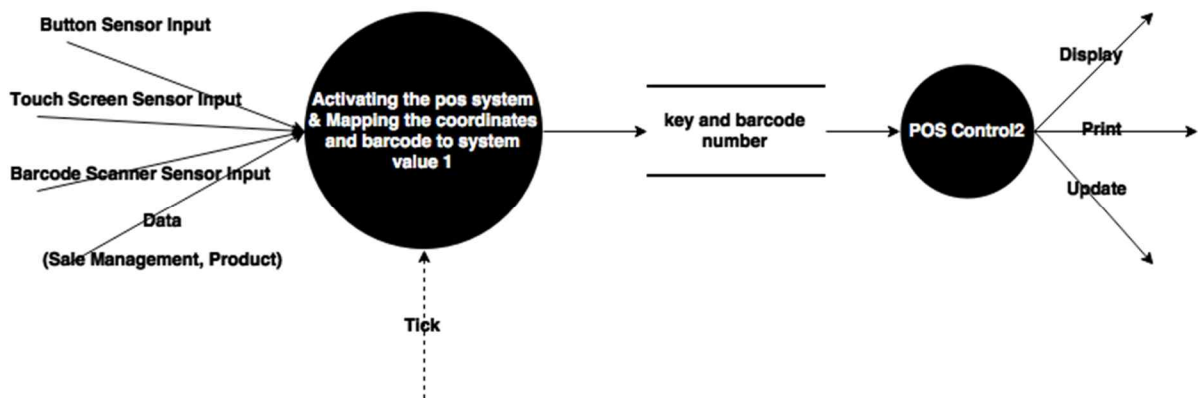
3.2.1.3 Data Dictionary

Input/Output Event	Description	Format/Type
Button Sensor Input	POS 시스템의 전원 신호를 전달한다.	
Touch Screen Sensor Input	터치된 화면 좌표를 전달한다.	
Barcode Scanner Sensor Input	스캐너로 감지된 상품의 바코드, 영수증 바코드를 전달한다.	
Data(Sale Management, Product)	서버로부터 요청된 판매 관리 데이터와 상품 데이터이다.	<pre> struct _inventory { Product list[7]; struct tm * time; }; struct product_code { unsigned char b1 : 1; unsigned char b2 : 1; }; typedef struct _product { struct product_code key; unsigned int volume; unsigned int price; } Product; struct _management { Record * _list; int count; } typedef struct { struct _sale_unit sale; struct _refund_unit refund; } Record; </pre>
Current Time	디지털 시계로 시각을 전달한다.	<pre> struct tm { int tm_sec; /* Seconds */ int tm_min; /* Minutes */ int tm_hour; /* Hour (0--23) */ int tm_mday; /* Day of month (1--31) */ int tm_mon; /* Month (0--11) */ int tm_year; /* Year (calendar year minus 1900) */ int tm_wday; /* Weekday (0--6; Sunday = 0) */ int tm_yday; /* Day of year (0--365) */ int tm_isdst; /* 0 if daylight savings time is not in effect) */ } </pre>

Sale Info	판매 상황 및 판매 완료 상황에서 실시간으로 업데이트 된 상품 판매 리스트 정보, 판매 영수증에 출력될 정보와 동일하다.	<pre> struct _sale_unit { Product * products[7]; int amount; int money; int change; struct tm * t; } </pre>
Refund Info	환불상황에서 환불 영수증에 출력될 정보와 동일하다.	<pre> typedef struct _sale_unit Receipt; struct _refund_unit { Receipt receipt; struct tm * t; } </pre>
Inventory Info	재고 확인 상황에서 요청된 시점에서의 내부 스토리지 상태를 의미한다.	<pre> struct _inventory { Product list[7]; struct tm * time; }; </pre>
Refund error message	환불 상황에서 예외 발생으로 인해 정상적으로 서비스를 제공하지 못할 때 출력될 메시지이다.	String
Stock document		text file
Settle document	재고 확인, 정산, 환불, 판매 완료 상황에서 문서로 출력될 데이터이다.	text file
Refund document		text file
Sale document		text file

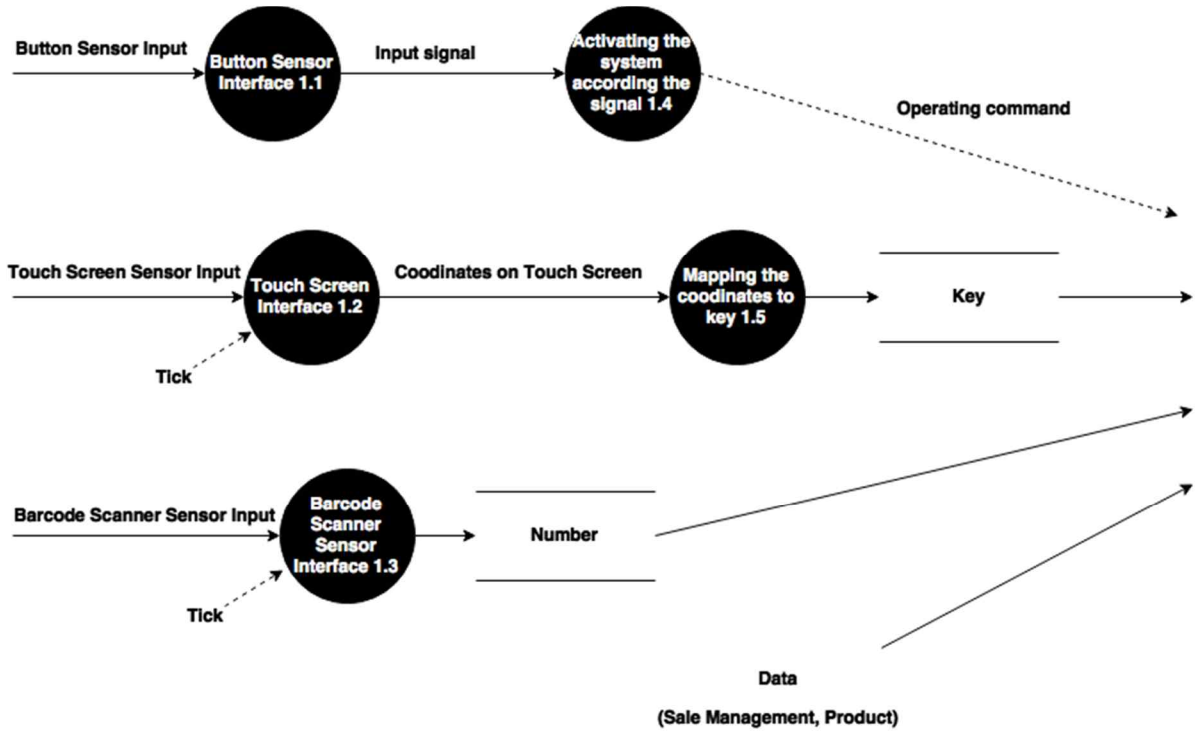
3.2.2 DFD Level 1

3.2.2.1 DFD



3.2.3 DFD Level 2

3.2.3.1 DFD



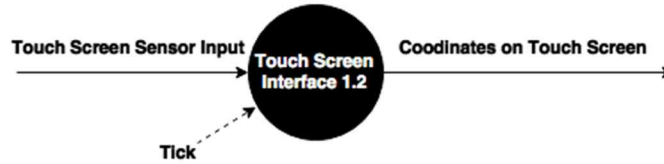
3.2.3.2 Process Specification

3.2.3.2.1 Process 1.1



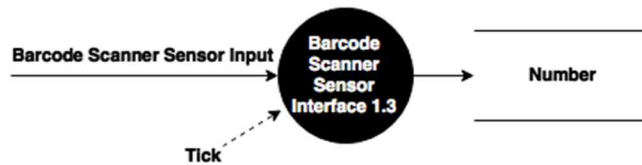
Reference No.	1.1
Name	Button Sensor Interface
Input	Button Sensor Input
Output	Input signal
Process Description	눌러진 버튼에 대한 입력신호를 전달한다.

3.2.3.2.2 Process 1.2



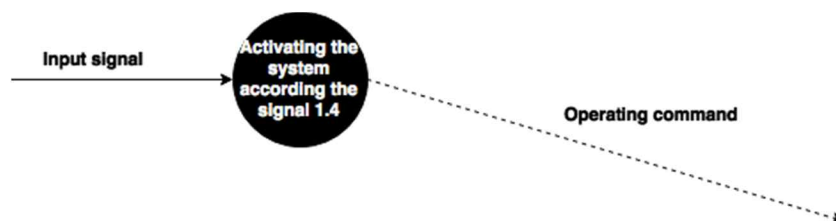
Reference No.	1.2
Name	Touch Screen Interface
Input	Touch Screen Sensor Input
Output	Coordinates on Touch Screen
Process Description	터치로 눌린 화면 상의 좌표 값을 전달한다.

3.2.3.2.3 Process 1.3



Reference No.	1.3
Name	Barcode Scanner Sensor Interface
Input	Barcode Scanner Sensor Input
Output	Number
Process Description	입력된 바코드 값을 시스템이 처리할 수 있는 바코드 번호로 변형시켜서 내보낸다.

3.2.3.2.4 Process 1.4



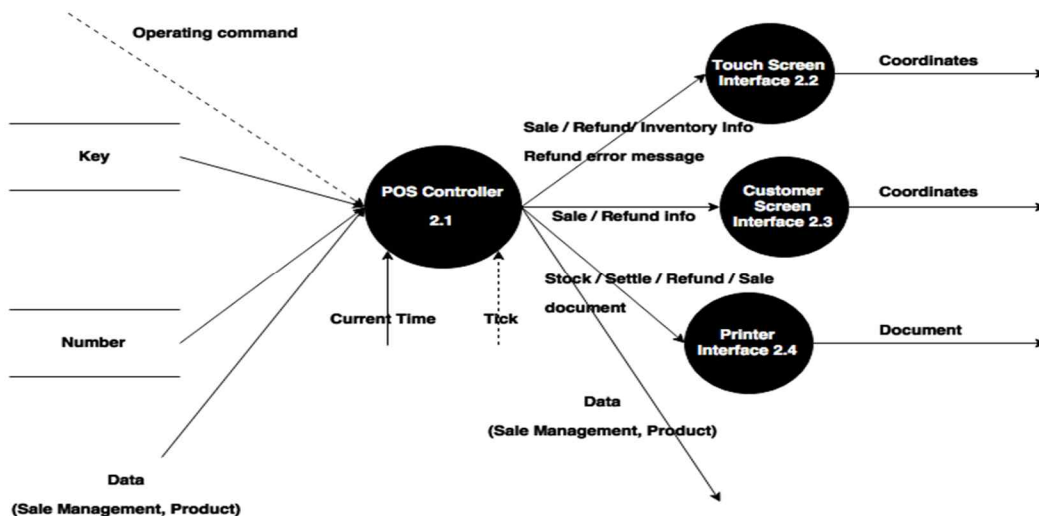
Reference No.	1.4
Name	Activating the system according the signal
Input	Input signal
Output	Operating command
Process Description	전달받은 신호에 따라 high voltage인 경우 포스기를 활성화시키고 low voltage인 경우 포스기를 비활성화 시킨다.

3.2.3.2.5 Process 1.5



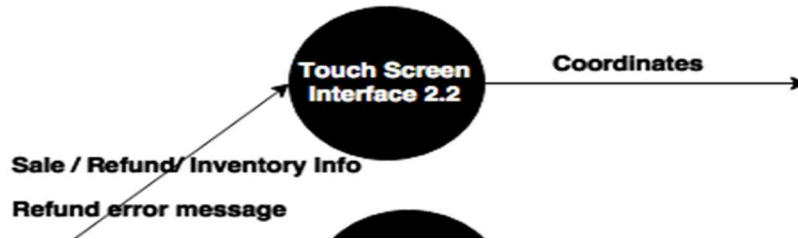
Reference No.	1.5
Name	Mapping the coordinates to key
Input	Coordinates on Touch Screen
Output	Key
Process Description	터치 스크린 상의 화면 좌표에 대응되는 메뉴를 시스템 내부에서 처리 가능한 Key 값으로 매핑시킨다.

3.2.3.2.6 Process 2.1



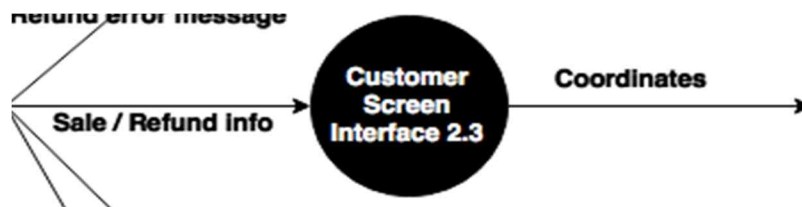
Reference No.	2.1
Name	POS Controller
Input	Operating command, Key, Number, Data(Sale management, Product), Current Time
Output	Sale / Refund Inventory Info, Refund error message, Sale / Refund Info, Stock / Settle / Refund / Sale document, Data(Sale management, Product)
Process Description	<p>1) 활성화 제어를 받으면 그에 따라 포스기 전원을 키거나 끈다.</p> <p>2) Key를 받으면 Key에 대한 값에 따라 포스기 서비스를 수행하는데 시스템 조건에 따라 자동적으로 Key 값이 변경될 수도 있다.</p> <ul style="list-style-type: none"> ▪ 판매[SALE] : 키 값이 FINISH가 되기 전까지 바코드 스캐너로 상품 번호를 입력 받고 상품 판매 리스트를 업데이트 해준다. ▪ 환불[REFUND] : 유효한 영수증인지를 먼저 확인한 후 유효하지 않은 영수증이면 에러 메시지를 터치스크린에 출력하도록 하고 유효한 영수증이면 영수증 단위로 환불을 수행하여 재고 상태를 업데이트 시킨다. ▪ 재고 확인[STOCK] : 요청된 시점에서의 내부 스토리지 상태를 스냅샷 찍어서 출력한다. ▪ 정산[SETTLE] : 디지털 시계로부터 전달받은 Current Time이 자정을 가리킬 때 키 값은 자동적으로 SETTLE로 바뀌고 판매 실적에 대한 정산을 내린다. 정산 후에 내부 스토리지 초기화 및 서버의 재고를 초기화하라고 요청한다. ▪ 취소[CANCEL] : 환불이나 판매 서비스 도중에 키 값이 CANCEL이 되면 서비스 수행하기 전에 찍어놓은 내부 스토리지 스냅샷 상태로 복원시킨다. ▪ 증가/감소[ADD/MINUS] : 상품 판매 서비스를 진행 중에 활성화되는 키 값으로 이에 따라 판매 상품 수량을 조절한다. <p>1) 바코드 번호를 받으면 상품 번호인지 영수증 번호인지 식별하고 그에 따른 동작을 수행한다.</p> <ul style="list-style-type: none"> • 상품 번호 -> 판매 시작 • 영수증 번호 -> 영수증 식별 -> 환불 시작 <p>2) 서버로부터 Data를 받으면 포스기 내부 스토리지와 동기화한다.</p> <p>3) 외부 기기 제어</p> <ul style="list-style-type: none"> • SALE : 실시간으로 업데이트 되는 판매 리스트를 Touch Screen과 Customer Screen에 출력하도록 정보를 전달한다. • FINISH : 판매 완료에 대한 판매 영수증에 대한 데이터를 Touch Screen, Customer Screen, Printer에 전달한다. • REFUND : 환불이 불가능한 경우, 에러 메시지를 Touch Screen에 출력하도록 정보를 제공하고 환불이 정상적으로 완료된 경우에는 환불 영수증에 대한 데이터를 Touch Screen, Customer Screen, Printer에 전달한다. • STOCK : 요청된 시점에서의 내부 스토리지 상태에 대한 데이터를 Touch Screen, Printer에 전달한다. • SETTLE : 정산 영수증에 대한 데이터를 Printer에 전달한다.

3.2.3.2.7 Process 2.2



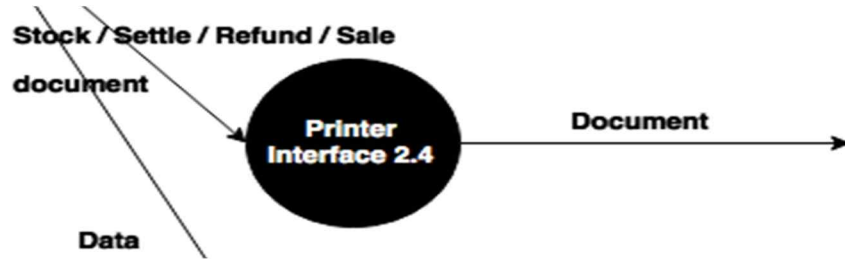
Reference No.	2.2
Name	Touch Screen Interface
Input	Sale / Refund / Inventory Info, Refund error message
Output	Coordinates
Process Description	포스기로부터 출력하라는 명령을 받고 현재 출력해야 할 데이터에 대한 화면(터치 스크린) 상의 좌표 값을 설정 후 반환한다.

3.2.3.2.8 Process 2.3



Reference No.	2.3
Name	Customer Screen Interface
Input	Sale / Refund Info
Output	Coordinates
Process Description	포스기로부터 출력하라는 명령을 받고 현재 출력해야 할 데이터에 대한 화면(고객용 스크린) 상의 좌표 값을 설정 후 반환한다.

3.2.3.2.9 Process 2.4



Reference No.	2.4
Name	Printer Interface
Input	Stock / Settle / Refund / Sale document
Output	Document
Process Description	포스기로부터 출력하라는 명령을 받고 프린터가 현재 출력해야 할 데이터에 대한 문서들을 반환한다.

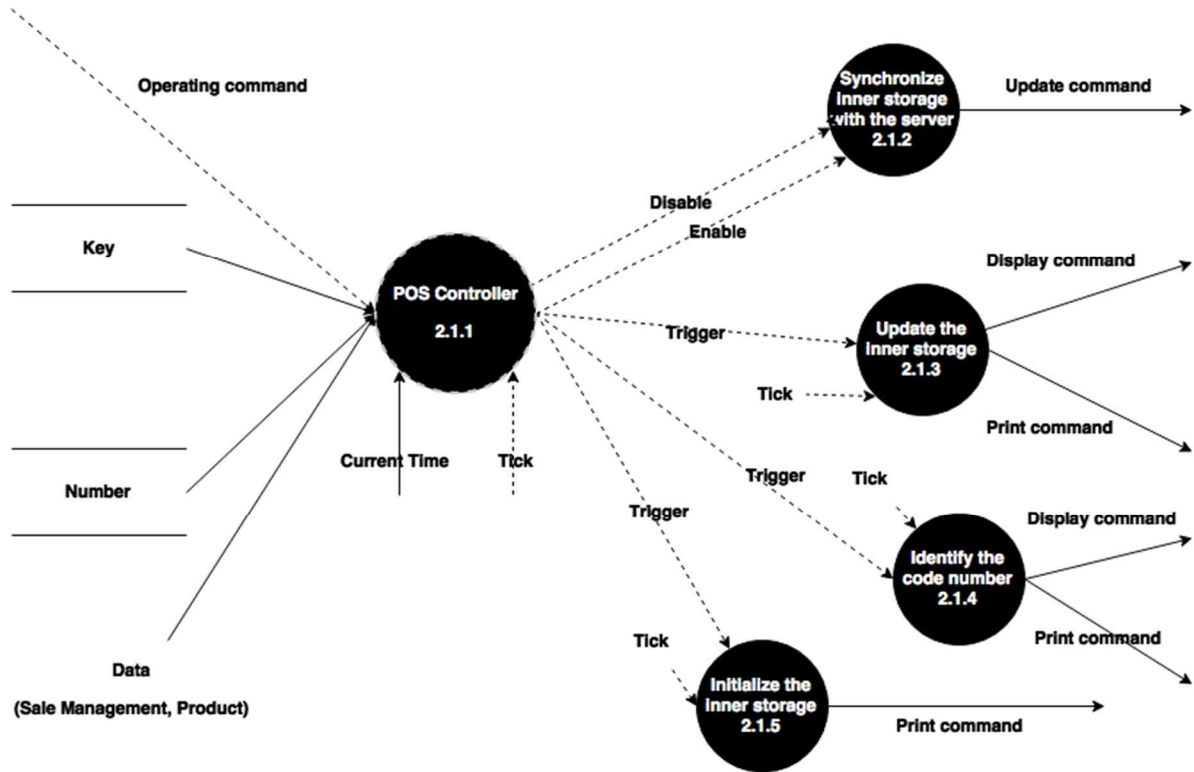
3.2.3.3 Data Dictionary

Input/Output Event	Description	Format/Type
Button Sensor Input	POS 시스템의 전원 신호를 전달한다.	
Touch Screen Sensor Input	터치된 화면 좌표를 전달한다.	
Barcode Scanner Sensor Input	스캐너로 감지된 상품의 바코드, 영수증 바코드를 전달한다.	
Input signal	버튼에 따른 전원 입력 신호이다.	
Coordinates on Touch Screen	터치가 발생한 화면 상의 좌표 값이다.	
Operating command	활성화 신호에 따라 포스기 전원을 제어하는 명령이다.	
Number	바코드 스캐너로부터 전달된 바코드 값이 시스템 내부에서 이용 가능하도록 변형된 정수 형태이다.	Integer
Key	터치가 감지된 화면 좌표에 대응하는 작업 키 값이다.	enum { SALE, REFUND, STOCK, SETTLE, CANCEL, FINISH} enum { ADD, MINUS} Integer (상품 구매에 따른 입력 금액을 위해)
		struct _inventory { Product list[7]; struct tm * time; }; struct product_code { unsigned char b1 : 1; unsigned char b2 : 1; };

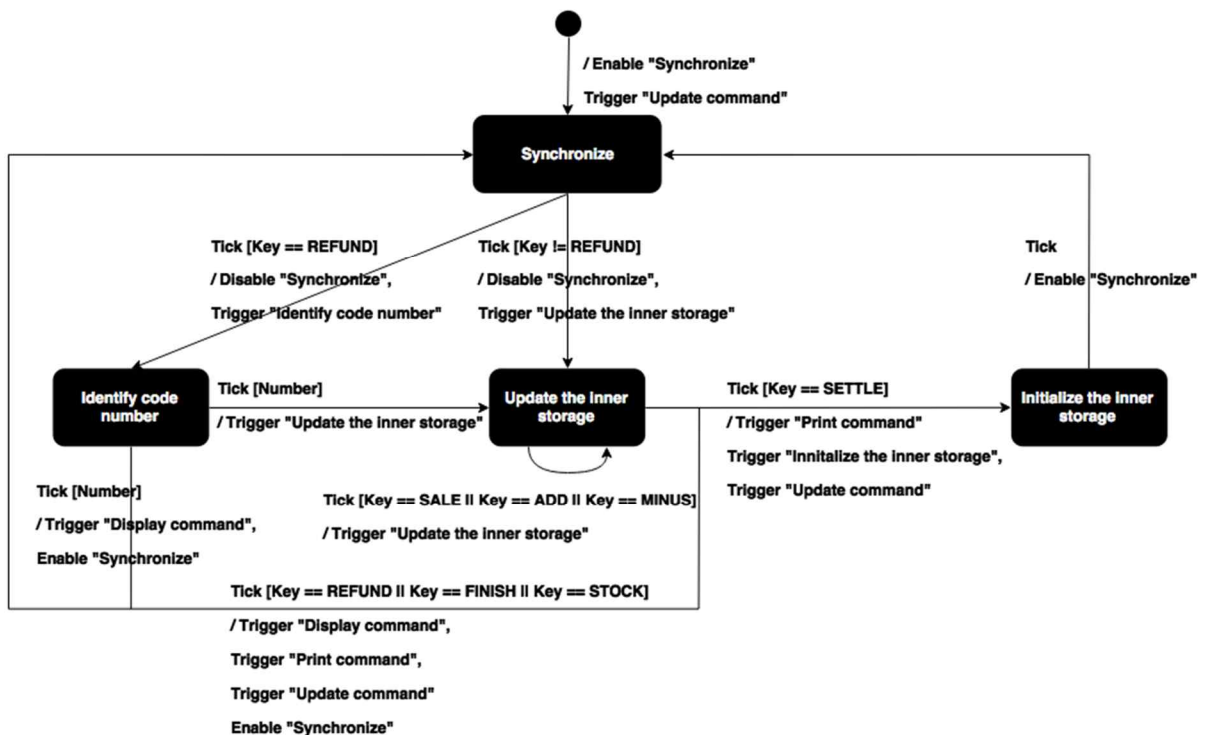
Data(Sale Management, Product)	서버로부터 요청된 판매 관리 데이터와 상품 데이터이다.	<pre>typedef struct _product { struct product_code key; unsigned int volume; unsigned int price; } Product; struct _management { Record * _list; int count; } typedef struct { struct _sale_unit sale; struct _refund_unit refund; } Record;</pre>
Current Time	디지털 시계로 시각을 전달한다.	<pre>struct tm { int tm_sec; /* Seconds */ int tm_min; /* Minutes */ int tm_hour; /* Hour (0--23) */ int tm_mday; /* Day of month (1--31) */ int tm_mon; /* Month (0--11) */ int tm_year; /* Year (calendar year minus 1900) */ int tm_wday; /* Weekday (0--6; Sunday = 0) */ int tm_yday; /* Day of year (0--365) */ int tm_isdst; /* 0 if daylight savings time is not in effect) */ }</pre>
Sale Info	판매 상황 및 판매 완료 상황에서 실시간으로 업데이트 된 상품 판매 리스트 정보, 판매 영수증에 출력될 정보와 동일하다.	<pre>struct _sale_unit { Product * products[7]; int amount; int money; int change; struct tm * t; }</pre>
Refund Info	환불상황에서 환불 영수증에 출력될 정보와 동일하다.	<pre>typedef struct _sale_unit Receipt; struct _refund_unit { Receipt receipt; struct tm * t; }</pre>
Inventory Info	재고 확인 상황에서 요청된 시점에서의 내부 스토리지 상태 정보를 의미한다.	<pre>struct _inventory { Product list[7]; struct tm * time; };</pre>
Refund error message	환불 상황에서 예외 발생으로 인해 정상적으로 서비스를 제공하지 못할 때 출력될 메시지이다.	String
Stock document		text file
Settle document	재고 확인, 정산, 환불, 판매 완료 상황에서 문서로 출력될 데이터이다.	text file
Refund document		text file
Sale document		text file

3.2.4 DFD Level 3

3.2.4.1 DFD



3.2.4.2 State Transition Diagram



* STD 추가적인 내용

- Key

Key는 터치 스크린로부터 입력 받은 위치 값을 터치 스크린 센서 인터페이스가 SALE, REFUND, INVEN, FINISH, SETTLE, CANCEL, ADD, MINUS로 각각 맵핑시킨다.

- Status Box

1. Synchronize : 포스기가 서비스를 수행하기 전 포스기 서버와 내부 스토리지(재고 상태, 포스기 사용 내역 등)와 상태를 동기화 시킨다.

2. Identify code number : 환불 영수증이 입력으로 들어왔을 때(바코드 스캐너 센서 인터페이스가 숫자 형식으로 바꿔줌) 해당 영수증이 유효한 영수증인지 검사한다. Tick 조건에 있는 Number는 유효한 영수증일 때 영수증 번호가 정상적으로 들어가 있고 유효하지 못한 영수증인 경우에만 0이 들어간다.

3. Update the inner storage : 작업 Key 값이 SALE, REFUND, SETTLE, INVEN (부속적인 동작인 CANCEL, FINISH, ADD, MINUS를 제외하고)일 때 각 작업에 맞는 서비스 제공하면서(SALE일 경우) 또는 제공 후에 내부 스토리지 상태를 업데이트시킨다. 작업 키가 판매 상황에서 발생할 수 있는 SLAE, ADD, MINUS일 때에만 지속적으로 Update the inner storage 상태를 가진다. 작업 키에 이외의 값이 들어온 경우에만 다음 상태로 전이한다.

4. Initialize the inner storage : 자정이 되었을 때 시스템에서 작업 키를 SETTLE로 변경함에 따라 내부 스토리지를 초기화하고 포스기 서버도 상품 재고를 초기화하도록 요청한다.

3.2.5 Overall DFD

