Software Requirement Analysis for Point Of Sale System

Project Team
Team 6

Date
2017.11.06

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

오류! 책갈피가 정의되어 있지 않습니다.

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

오류! 책갈피가 정의되어 있지 않습니다.

3.2.1.2 Process Specification 9

오류! 책갈피가 정의되어 있지 않습니다.

3.2.1.3 Data Dictionary 9

오류! 책갈피가 정의되어 있지 않습니다.

3.2.2 DFD Level 1 11

3.2.2.1 DFD 11

3.2.3 DFD Level 2 12

3.2.3.1 DFD 12

3.2.3.2 Process Specification 13

3.2.3.2.1 Process 1.1 13

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 14

오류! 책갈피가 정의되어 있지 않습니다.

3.2.3.2.5 Process 1.5 14

3.2.3.2.6 Process 2.1 15

3.2.3.2.7 Process 2.2 16

3.2.3.2.7 Process 2.2 오류! 책갈피가 정의되어 있지 않습니다.

오류! 책갈피가 정의되어 있지 않습니다.

3.2.3.2.8 Process 2.3 16

오류! 책갈피가 정의되어 있지 않습니다.

3.2.3.2.9 Process 2.4 16

오류! 책갈피가 정의되어 있지 않습니다.

3.2.3.3 Data Dictionary 17

오류! 책갈피가 정의되어 있지 않습니다.

3.2.4 DFD Level 3 19

3.2.4.1 DFD 19

3.2.4.2 State Transition Diagram 20

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

모든 시스템은 SW 만으로 구현하고 HW 가 필요한 부분은 SW 모듈을 만들어 가상의 HW 를 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 프린터기를 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 프린터기를 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.
POST는 매일 24시(자정)에 정산이 이루어지며 재고 서버에 있는 하루 동안 있었던 판매, 환불 정보를 프린터기를 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

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

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

POST 의 프린터를 사용한 모든 출력은 텍스트 파일 생성으로 대체한다.

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

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

- 201709111843

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

POST 에서 다루는 상품은 총 7 종류(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 2 개의 텍스트 파일로 대체한다.

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

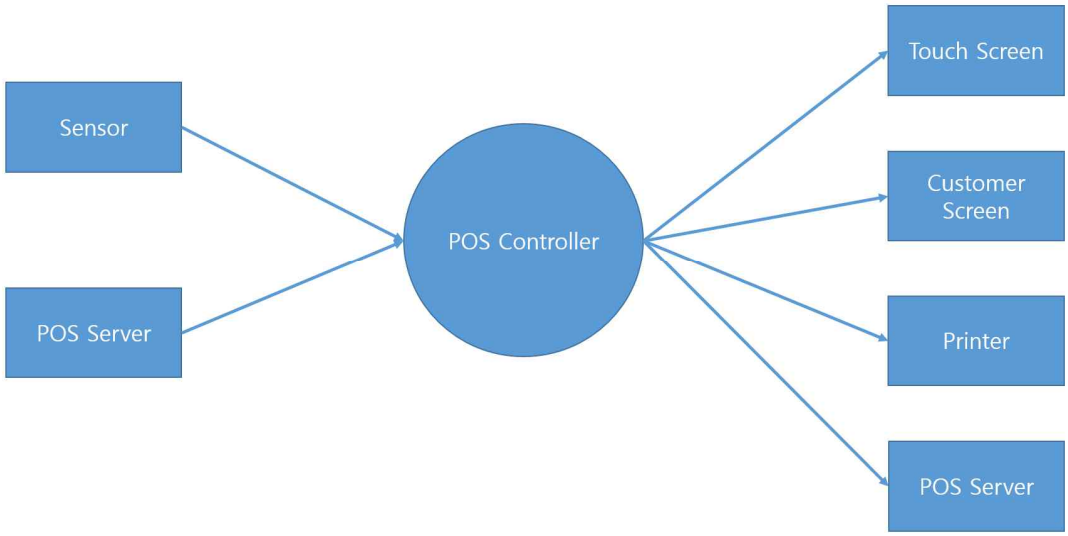
하루가 지나면 재고 서버의 상품 파일, 판매 관리 파일은 다음 날짜로 새로 생성이 되며 상품 파일의 모든 상품은 100 개로 초기화한다.

POST 의 전원이 켜지면 자동으로 재고 서버와 상품을 동기화한다.

3 Structured Analysis

3.1 System Context Diagram

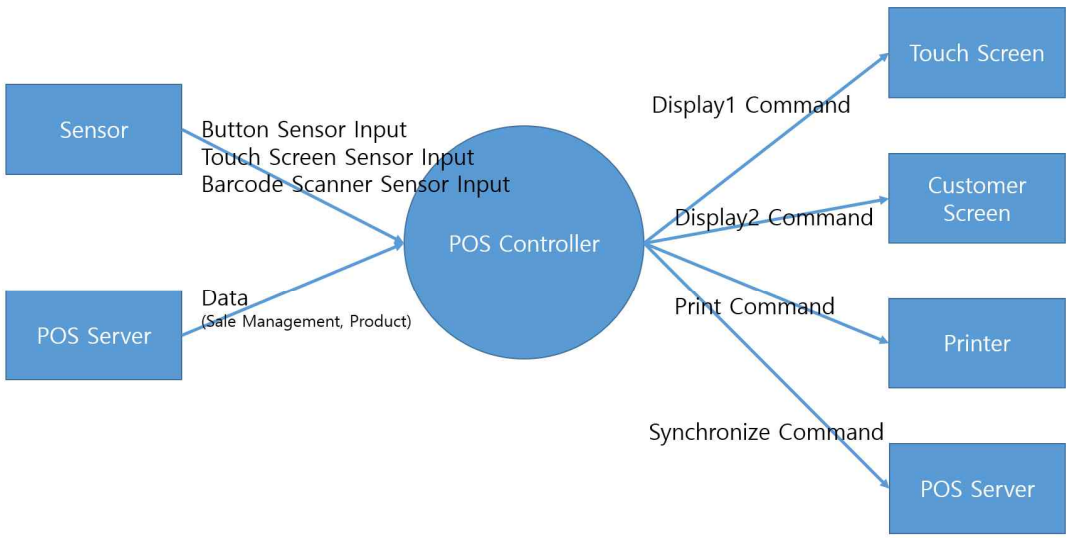
3.1.1 Basic System Context Diagram



3.1.2 Event List

Input / Output Event	Description
Button Sensor Input	POS 시스템의 전원 신호를 전달
Touch Screen Sensor Input	터치된 화면 좌표를 전달
Barcode Scanner Sensor Input	스캐너로 감지된 상품 및 영수증 바코드를 전달
Data (Sale Management, Product)	서버에 요청한 판매 관리 데이터, 상품 데이터
Current Time	디지털 시계로 시각을 전달
Display1 Command	캐셔가 보는 화면인 Touch Screen에 출력될 데이터
Display2 Command	손님이 보는 화면인 Customer Screen에 출력될 데이터
Print Command	Printer에 출력될 데이터
Synchronize Command	POS Server에 업데이트 되어야 할 데이터

3.1.3 The System Context Diagram

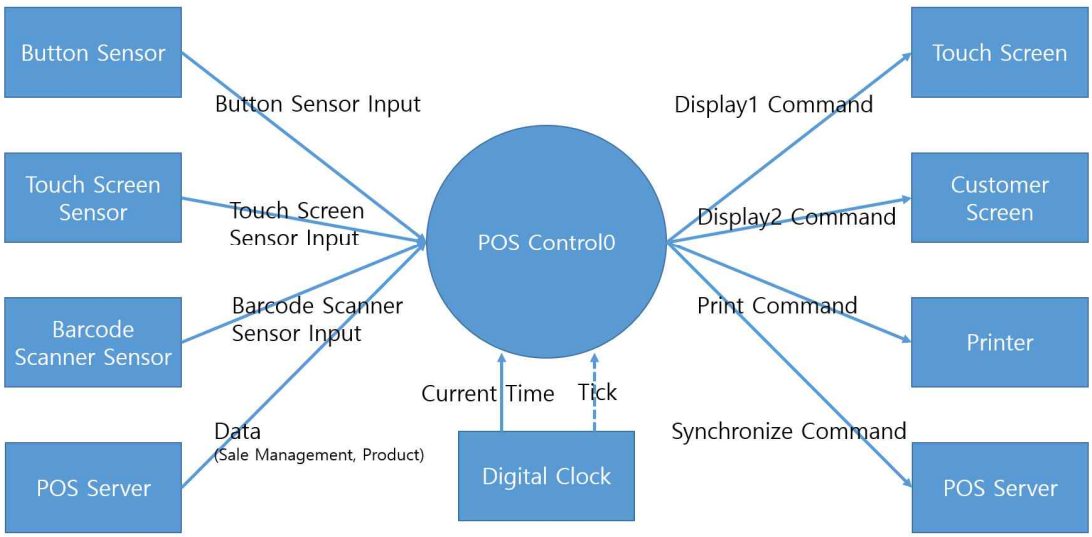


3.2 Data Flow Diagram

3.2.1 DFD level 0

3.2.1.1 DFD

DFD level0



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
Output	Display1,2 Command, Print Command, Synchronize Command
Process Description	1) 버튼 센서에서 전달 받은 신호에 따라 활성화, 비활성화 된다. 2) 터치스크린 센서로부터 전달 받은 화면 좌표에 대응되는 동작들을 수행하고 관련된 외부 동작을 수행할 수 있게 트리거한다. 3) 바코드 스캐너 센서에서 전달 받은 데이터를 통해 판매, 환불 동작을 수행한다. 4) 서버의 데이터를 받아 내부 스토리지와 동기화를 수행한다.

3.2.1.3 Data Dictionary

Input / Output Event	Description	Format / Type
Button Sensor Input	POS 시스템의 전원 신호를 전달	
Touch Screen Sensor Input	터치된 화면 좌표를 전달	
Barcode Scanner Sensor Input	스캐너로 감지된 상품 및 영수증 바코드를 전달	

Data(Sale Management, Product)	서버에 요청한 판매 관리 데이터, 상품 데이터	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;

Current Time	디지털 시계로 시각을 전달	struct tm
		{
		int tm_sec;
		int tm_min;
		int im_hour;
		.....
		}; // library file 사용

Display1 Command	캐셔가 보는 화면인 Touch Screen에 출력될 데이터	Sale Info, Refund Info, Inventory Info, Error msg,
Display2 Command	손님이 보는 화면인 Customer Screen에 출력될 데이터	Sale Info, Refund Info
Print Command	Printer에 출력될 데이터	Sale Info, Refund Info, Stock Info, Settle Info

Sale Info	판매 상황, 판매 완료 상황에 실시간으로 업데이트 된 상품 판매 리스트 정보	typedef struct _sale_unit
		{
		Product * products[7];
		int amount;
		int money;
		int change;
		
		struct tm *t;
		}

Refund Info	환불 관련 정보	typedef struct _sale_unit Receipt;
		
		struct _refund_unit
		{
		Receipt receipt;
		struct tm *t;
		};

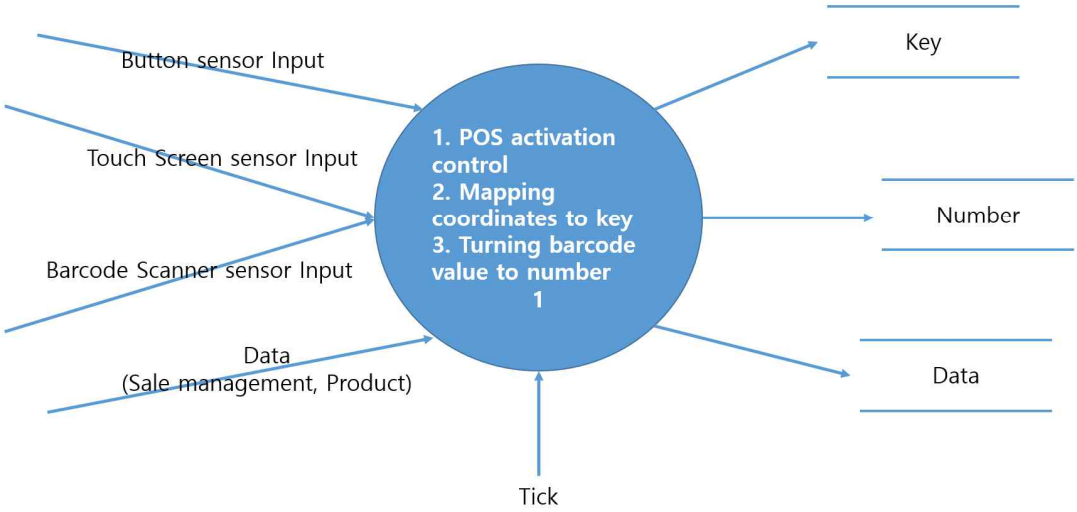
Inventory Info	내부 스토리지 정보	struct _inventory
		{
		Product list[7];
		struct tm *t;
		};

Error msg	판매, 환불 상황에서 발생할 수 있는 예외 처리 메시지	String
Synchronize Command	POS Server에 업데이트 되어야 할 데이터	Data

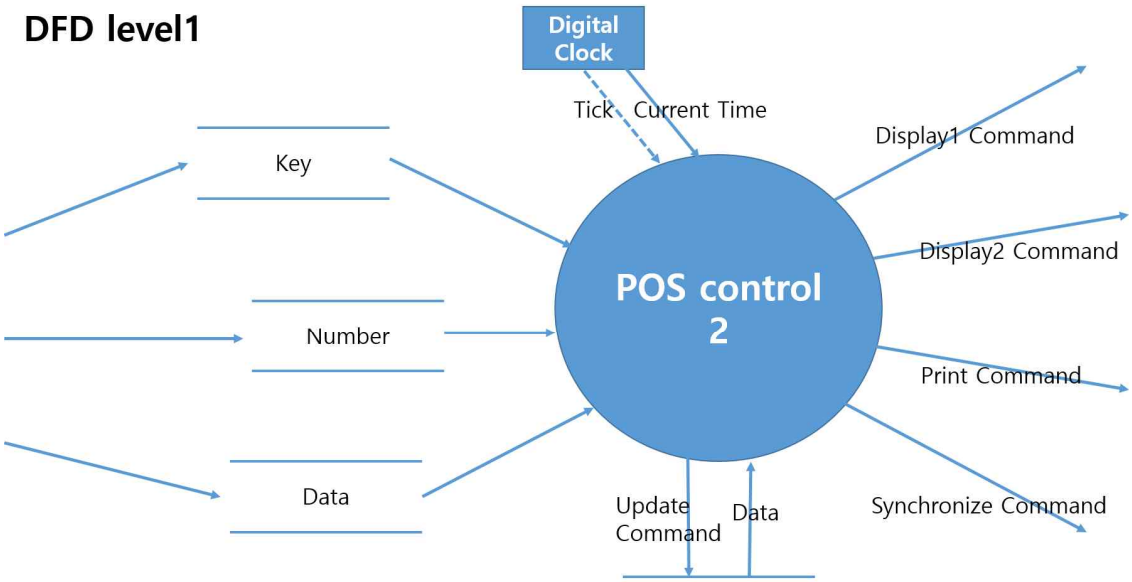
3.2.2 DFD Level 1

3.2.2.1 DFD

DFD level1



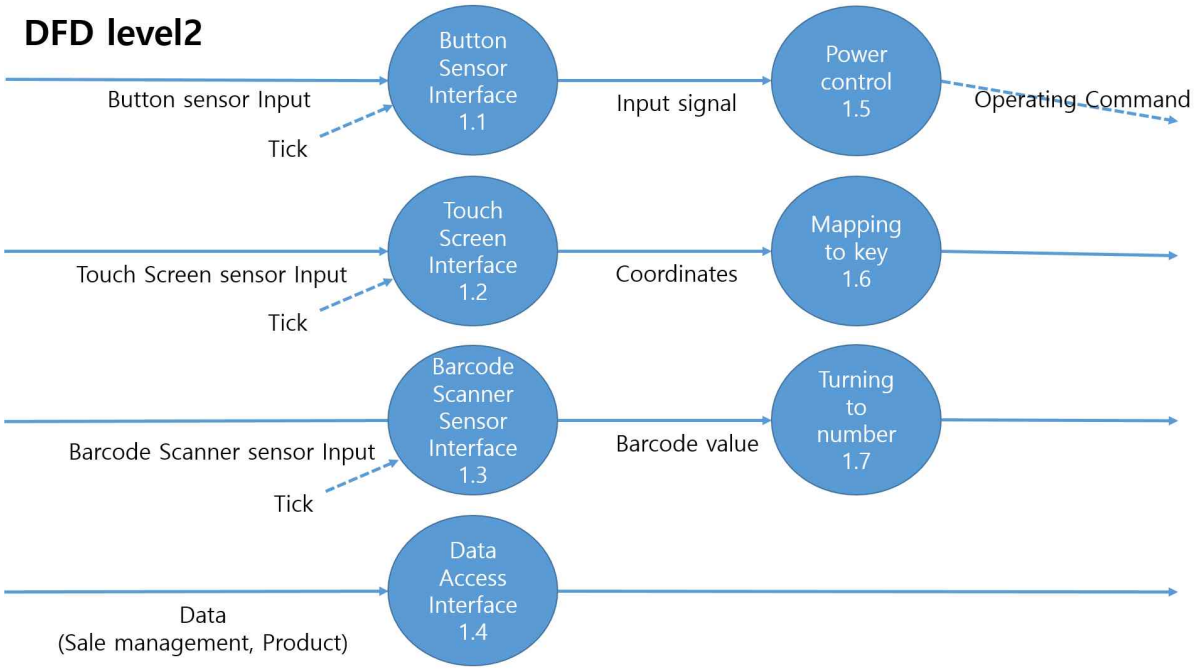
DFD level1

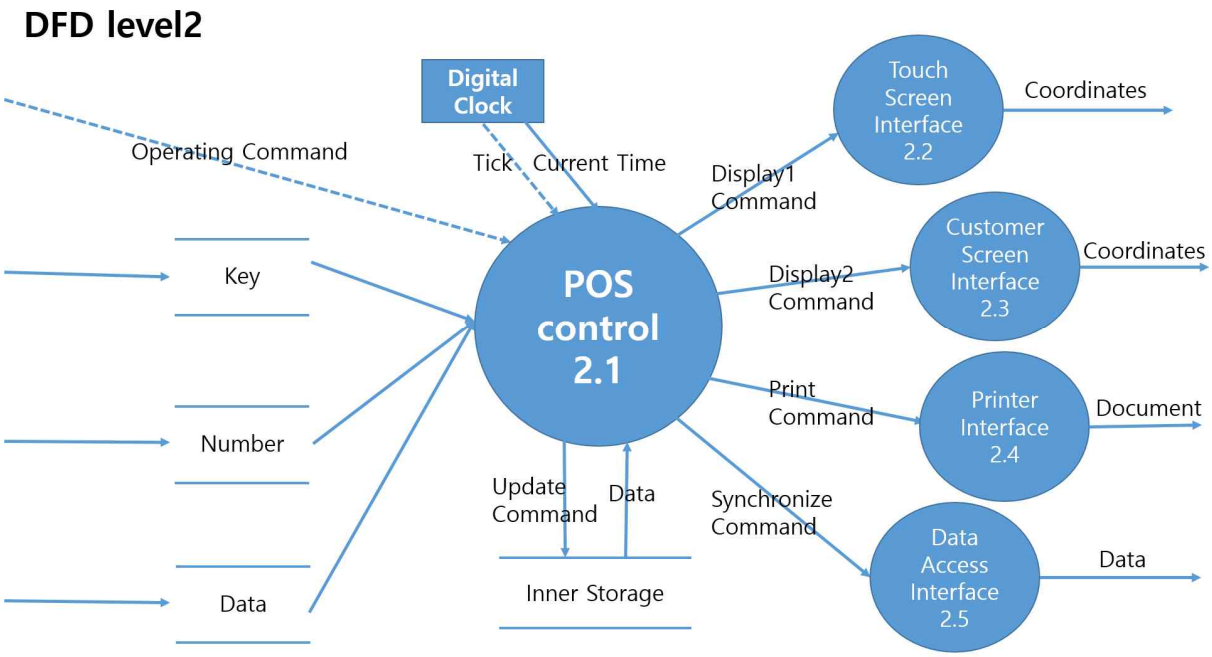


*Display1 : touch screen(캐셔가 보는 화면)
*Display2 : display screen(손님이 보는 화면)

3.2.3 DFD Level 2

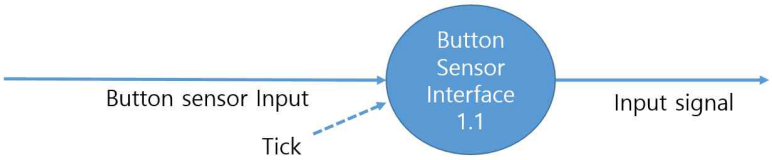
3.2.3.1 DFD





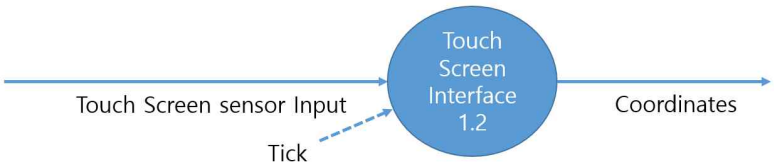
3.2.3.2 Process Specification

3.2.3.2.1 Process 1.1



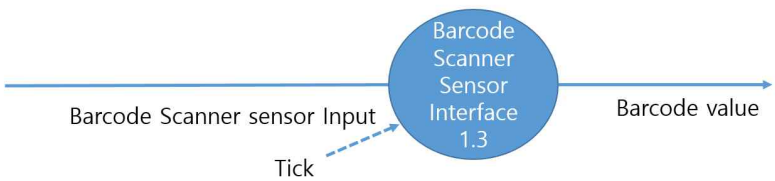
Reference No.	1.1
Name	Button Sensor Interface
Input	Button Sensor Input
Output	Input signal
Process Description	눌린 버튼에 대한 입력 신호를 전달

3.2.3.2.2 Process 1.2



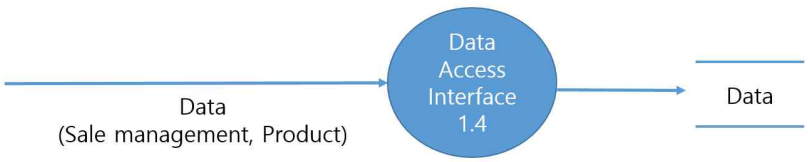
Reference No.	1.2
Name	Touch Screen Sensor Interface
Input	Touch Screen Sensor Input
Output	Coordinates on Touch Screen
Process Description	터치로 눌린 화면 상의 좌표 값을 전달

3.2.3.2.3 Process 1.3



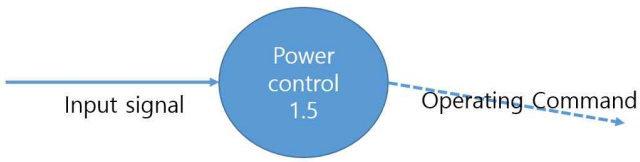
Reference No.	1.3
Name	Barcode Scanner Sensor Interface
Input	Barcode Scanner Sensor Input
Output	Barcode value
Process Description	바코드 스캐너로 읽은 바코드 데이터를 전달

3.2.3.2.4 Process 1.4



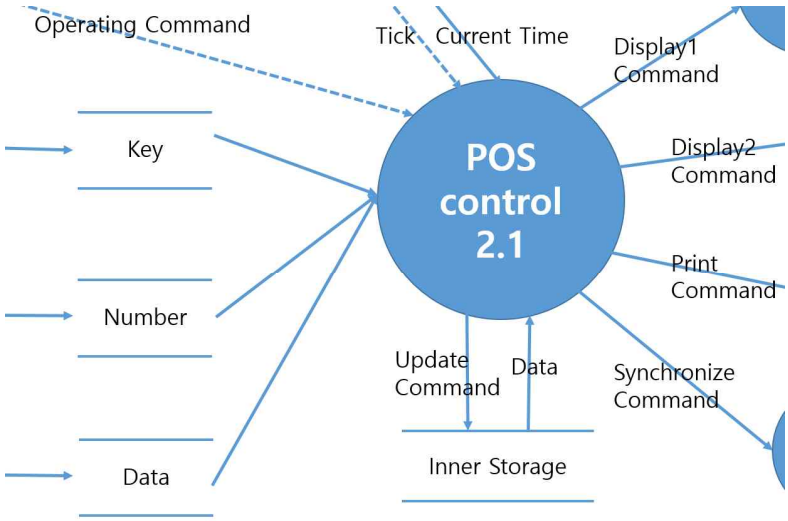
Reference No.	1.4
Name	Data Access Interface
Input	Touch Screen Sensor Input
Output	Data
Process Description	서버에 요청한 판매 관리 데이터, 상품 데이터

3.2.3.2.5 Process 1.5



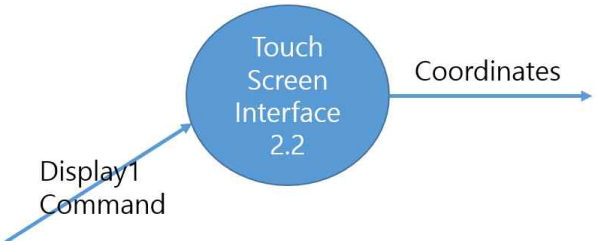
Reference No.	1.5
Name	Power control
Input	Input signal
Output	Operating Command
Process Description	Input signal에 따라 1(High voltage)인 경우 포스기 전원을 켜고 0(low voltage)인 경우 전원을 끈다.

3.2.3.2.6 Process 2.1



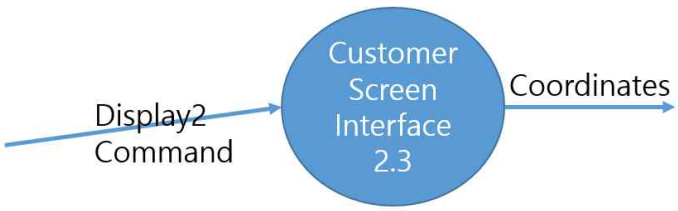
Reference No.	2.1
Name	POS Controller
Input	Operating Command, Key, Number, Data, Current Time
Output	Display1, 2 Command, Print Command, Update Command, Synchronize Command
Process Description	1) Operating Command를 입력 받으면 값에 따라 포스기 전원을 제어 2) Key를 입력 받으면 Key에 대응되는 동작을 수행하거나 시스템 조건에 의해 내부적으로 변경이 가능 • 판매[SALE] : 키가 PAYMENT 또는 ERROR가 되기 전까지 바코드 스캐너로 상품번호를 입력 받고 상품 판매 리스트를 업데이트하고 변경된 재고 상태를 Update Command로 전달 • 환불[REFUND] : 환불이 가능한 조건(유효한 바코드 코드)일 때 영수증 단위로 상품 수량이나 잔고를 Update Command로 전달하여 수정함 • 재고확인[STOCK] : 내부 스토리지 상태를 가져옴 • 정산[SETTLE] : 포스기 내부 시스템 시간으로 자정이 되었을 때 자동적으로 키 값이 바뀌고 판매 실적을 낸 후 내부스토리지를 초기값을 Update Command로 전달 • 취소[CANCEL] : 환불 상황에서 제공되는 키 값이며 동작 수행 이전의 내부스토리지 상태를 Update Command로 전달 • 증가/감소/지불금액[ADD/MINUS/SUM] : 판매 상황에서 제공되는 키 값이며 수량을 조절하거나 손님이 지불한 금액을 의미 • 에러[ERROR] : 판매, 환불 상황에서 발생할 수 있는 예외 조건에 해당되면 키 값을 변경함 • 종료[TERMINATE] : Operating Command 값이 0이면 모든 자원을 정리하고 포스기 전원을 종료함 3) 외부 entity와의 interaction • SALE : 상품 판매 리스트와 총액을 Display1,2 Command로 전달 • PAYMENT : 최종적인 상품 판매 리스트를 Display1 Command, Print Command로 전달 • REFUND : 환불이 가능한 경우 환불된 상품 판매 리스트를 Display 1,2 Command로 전달 • STOCK : 내부 스토리지 상태를 Diaplay1 Command, Print Command로 전달 • SETTLE : 판매 실적을 Print Command로 전달 • ERROR : 에러 메시지를 Display1 Command로 전달 4) 서버와 동기화 : Key 값에 따른 동작을 수행하기 전 내부 스토리지 상태를 Synchronize Command로 전달하여 동기화를 수행함

3.2.3.2.7 Process 2.2



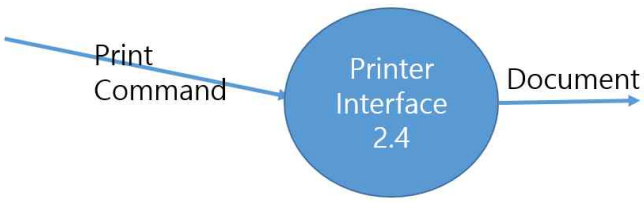
Reference No.	2.2
Name	Touch Screen Interface
Input	Display1 Command
Output	Coordinates
Process Description	Display1 Command의 값 중 활성화 데이터가 1이 되었을 때 터치스크린에 출력할 데이터의 좌표를 반환

3.2.3.2.8 Process 2.3



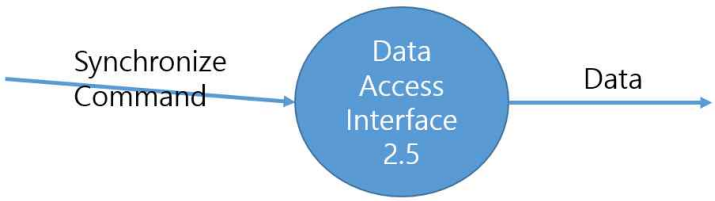
Reference No.	2.3
Name	Customer Screen Interface
Input	Display2 Command
Output	Coordinates
Process Description	Display2 Command의 값 중 활성화 데이터가 1이 되었을 때 손님용 스크린에 출력할 데이터의 좌표를 반환

3.2.3.2.9 Process 2.4



Reference No.	2.4
Name	Printer Interface
Input	Print Command
Output	Document
Process Description	Print Command의 값 중 활성화 데이터가 1이 되었을 때 프린터기가 출력해야 할 문서를 반환

3.2.3.2.10 Process 2.5



Reference No.	2.5
Name	Data Access Interface
Input	Synchronize Command
Output	Data
Process Description	서버에 업데이트할 판매 관리 데이터, 상품 데이터

3.2.3.3 Data Dictionary

Input / Output Event	Description	Format / Type
Button Sensor Input	**POS 시스템의 전원 신호를 전달**	
Touch Screen Sensor Input	**터치된 화면 좌표를 전달**	
Barcode Scanner Sensor Input	**스캐너로 감지된 상품 및 영수증 바코드를 전달**	
Operating Command	**POS 시스템 전원 제어**	**On(1), Off(0) / Integer**
		struct _inventory **{** **Product list[7];** **struct tm * time;** **};** **struct product_code** **{** **unsigned char b1 : 1;** **unsigned char b2 : 1;** **……** **};**

Data(Sale Management, Product)	서버에 요청한 판매 관리 데이터, 상품 데이터	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;

Inner Storage	POS 내부스토리지	Data

Current Time	디지털 시계로 시각을 전달	struct tm
		{
		int tm_sec;
		int tm_min;
		int im_hour;
		.....
		}; // library file 사용

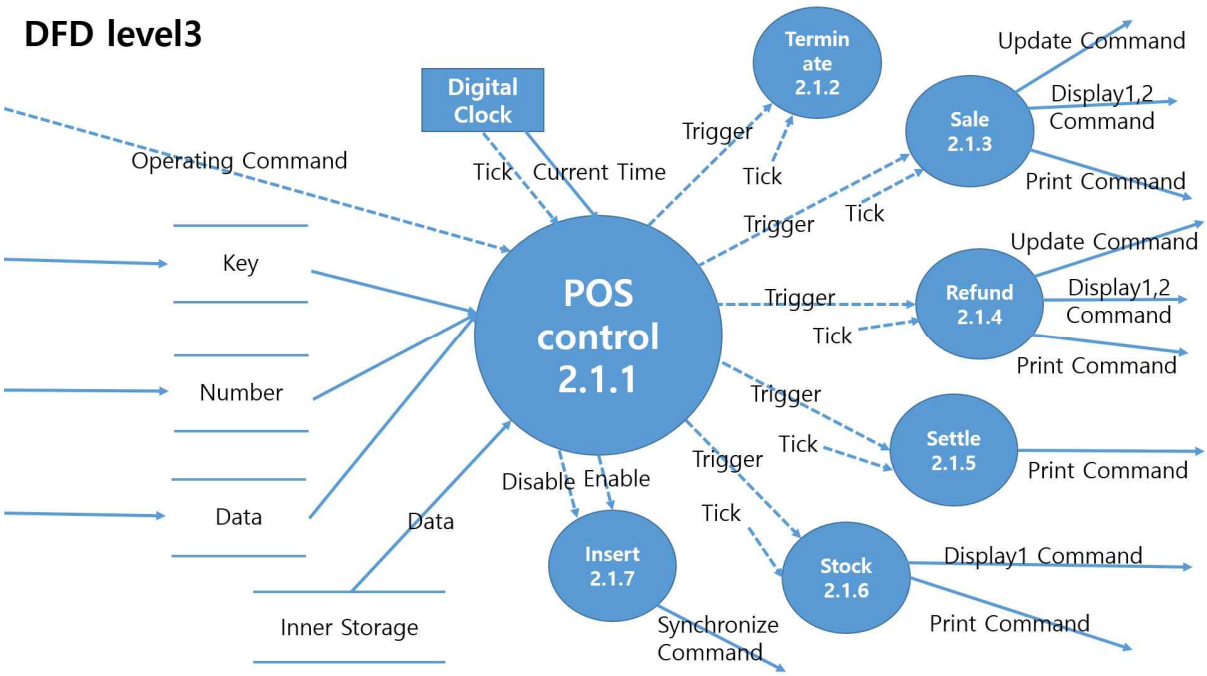
Key	포스기 동작 수행을 위한 키 값	SALE(0), REFUND(1), STOCK(2), SETTLE(3), CANCEL(4), PAYMENT(5), ERROR(6) ADD(0), MINUS(1) /Integer(Enum) SUM(손님이 지불하는 금액) / Integer
Number	바코드 번호를 정수로 변형시킨 값	Integer
Display1 Command	캐셔가 보는 화면인 Touch Screen에 출력될 데이터	Sale Info, Refund Info, Inventory Info, Error msg / struct On(1), Off(0) / Integer
Display2 Command	손님이 보는 화면인 Customer Screen에 출력될 데이터	Sale Info, Refund Info / struct On(1), Off(0) / Integer
Print Command	Printer에 출력될 데이터	Sale Info, Refund Info, Stock Info, Settle Info / struct On(1), Off(0) / Integer
Update Command	POS 내부스토리지에 업데이트할 데이터	Data

Sale Info	판매 상황, 판매 완료 상황에 실시간으로 업데이트 된 상품 판매 리스트 정보	typedef struct _sale_unit
		{
		Product * products[7];
		int amount;
		int money;
		int change;
		
		struct tm *t;
		}

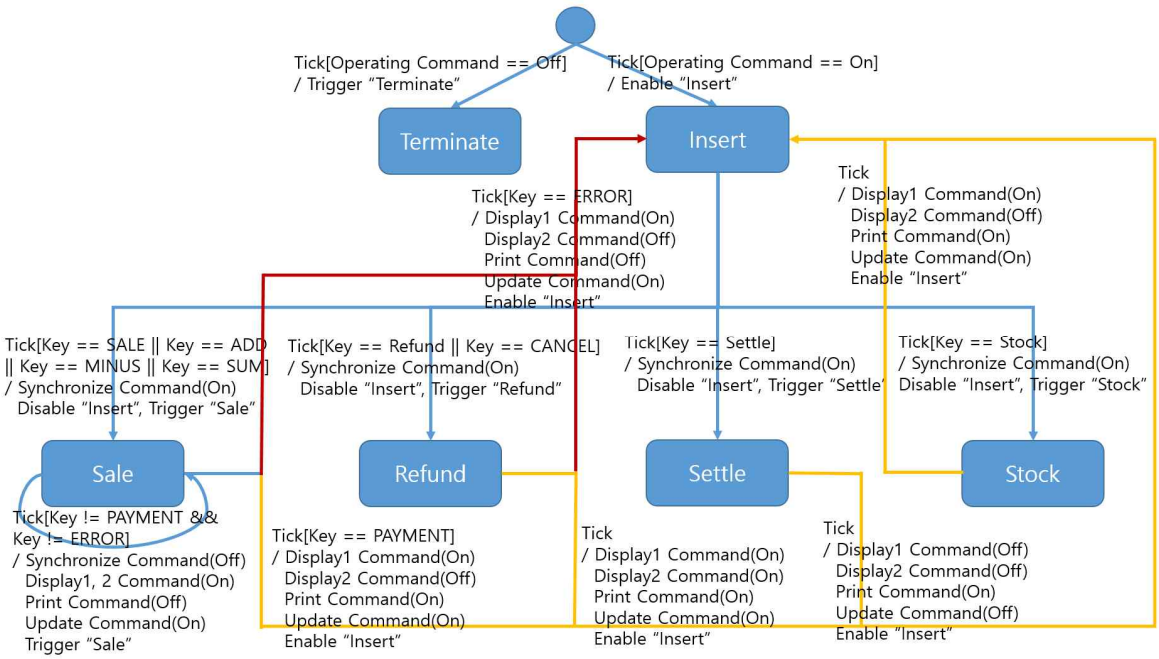
Refund Info	**환불 관련 정보**	**typedef struct _sale_unit Receipt;** **struct _refund_unit** **{** **Receipt receipt;** **struct tm *t;** **};**
Inventory Info	**내부 스토리지 정보**	**struct _inventory** **{** **Product list[7];** **struct tm *t;** **};**
Error msg	**판매, 환불 상황에서 발생할 수 있는 예외 처리 메시지**	**String**
Synchronize Command	**POS Server에 업데이트 되어야 할 데이터**	**Data**

3.2.4 DFD Level 3

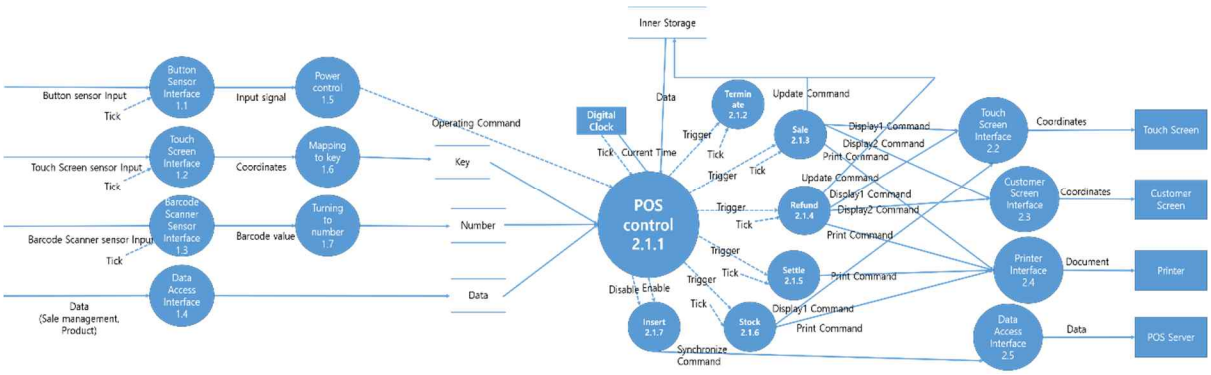
3.2.4.1 DFD



3.2.4.2 State Transition Diagram



* 빨간 선은 예외 상황이 발생 시 전이되는 흐름을 나타냄

* 노란 선은 기능 수행을 완료 후 작업(Key) 삽입 상태로 전이되기 전 내부스토리지 및 외부 entity에 동작 명령을 내리는 흐름을 나타냄

3.2.5 Overall DFD