

Software Requirement Analysis for Point Of Sale System

Project Team
Team 1

Date
2017-09-13

Team Information

201211337 김재현
201112052 방민석
201312259 백만일
201211383 조영래

Table of Contents

1.	개요 _____	5
1.1.	목적 _____	5
1.2.	범위 _____	5
1.3.	용어 _____	6
1.4.	참고 문헌 _____	6
1.5.	Overview _____	6
2.	개발 대상 설명 _____	7
2.1.	개발 대상 _____	7
2.2.	기능 (Statements of purpose) _____	7
2.3.	사용자 특징 _____	7
2.4.	제약 및 가정 사항 _____	8
3.	세부 요구 사항 _____	9
3.1.	외부 인터페이스 _____	9
3.1.1.	사용자 인터페이스 _____	9
3.1.2.	HW 인터페이스 _____	9
3.1.3.	SW 인터페이스 _____	9
3.1.4.	통신 인터페이스 _____	9
3.2.	기능 요구사항 _____	9
3.2.1.	판매 _____	9
3.2.1.1.	Function _____	9
3.2.1.2.	Input _____	11
3.2.1.3.	Output _____	11
3.2.2.	환불 _____	12
3.2.2.1.	Function _____	12
3.2.2.2.	Input _____	13
3.2.2.3.	Output _____	13
3.2.3.	정산 _____	14
3.2.3.1.	Function _____	14
3.2.3.2.	Input _____	14
3.2.3.3.	Output _____	14
3.2.4.	재고 확인 _____	15
3.2.4.1.	Function _____	15

3.2.4.2. Input	15
3.2.4.3. Output	15
3.3. 성능 요구사항	15
3.4. 설계 제약사항	15
3.5. 소프트웨어 속성	15
3.6. 기타 요구사항	15
4. Analysis	16
4.1. System Context Diagram	16
4.1.1. Basic System Context Diagram	16
4.1.2. Event List	16
4.1.3. The System Context Diagram	16
4.2. Data Flow Diagram	17
4.2.1. DFD level 0	17
4.2.1.1. DFD	17
4.2.1.2. Process Specification	17
4.2.1.2.1. Process 0	17
4.2.1.3. Data Dictionary	18
4.2.2. DFD Level 1	18
4.2.2.1. DFD	18
4.2.2.2. Process Specification	18
4.2.2.2.1. Process 1	18
4.2.2.2.2. Process 2	19
4.2.2.3. Data Dictionary	19
4.2.3. DFD Level 2	19
4.2.3.1. DFD	19
4.2.3.2. Process Specification	20
4.2.3.2.1. Process 1.1	20
4.2.3.2.2. Process 1.2	20
4.2.3.2.3. Process 1.3	20
4.2.3.2.4. Process 1.4	20
4.2.3.2.5. Process 2.1	21
4.2.3.2.6. Process 2.2	21
4.2.3.2.7. Process 2.3	21
4.2.3.3. Data Dictionary	21

4.2.4. DFD Level 3	22
4.2.4.1. DFD	22
4.2.4.2. Process Specification	23
4.2.4.2.1. Process 2.1.1	23
4.2.4.2.2. Process 2.1.2	23
4.2.4.2.3. Process 2.1.3	23
4.2.4.2.4. Process 2.1.4	23
4.2.4.2.5. Process 2.1.5	23
4.2.4.2.6. Process 2.1.6	24
4.2.4.3. Data Dictionary	24
4.2.4.4. State Transition Diagram (Controller)	24
4.2.5. Overall DFD	25

1. 개요

1.1. 목적

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

1.2. 범위

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 SW 만으로 구현하고 HW가 필요한 부분은 SW 모듈을 만들어 가상의 HW를 구현한다.

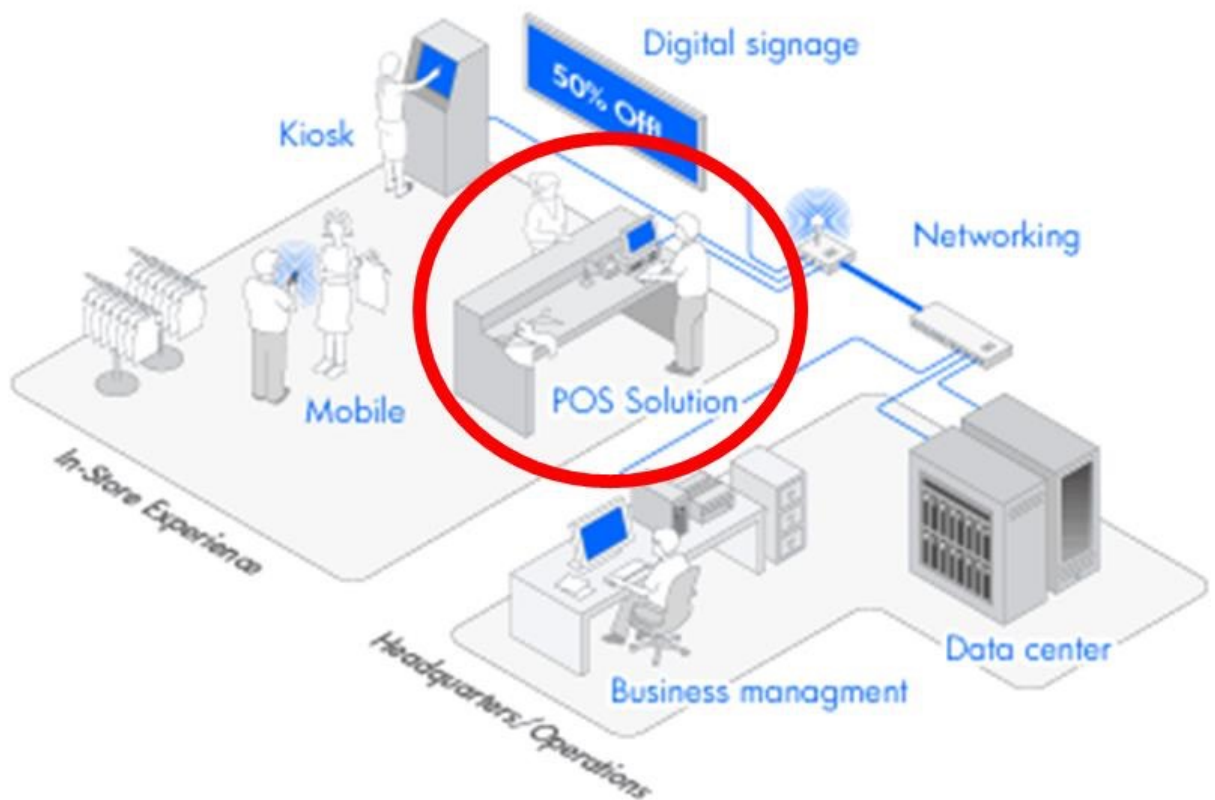


Figure 1 POS System

이미지 출처: <http://traudit.blogspot.com/2012/03/narrative-of-system-control-evaluation.html>

1.3. 용어

HW: Hardware

SW: Software

POS: Point Of Sale

POST: Point Of Sale Terminal

캐셔: Cashier POST 사용자

1.4. 참고 문헌

1.5. Overview

2장 개발 대상에 대한 설명 3장 세부기능 명세 4장 결과

2. 개발 대상 설명

2.1. 개발 대상

SW로 개발하는 가상의 POS 시스템

2.2. 기능 (Statements of purpose)

POST는 캐시 화면(터치 스크린)을 이용해 판매, 환불, 재고 확인을 할 수 있다.

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

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

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

POST는 판매 상황에서 고객 화면을 통해 판매할 상품의 총액을 보여 준다.

POST는 판매를 완료한 후 고객 화면과 캐시 화면에 각각 판매 정보를 출력한다. POST는 판매를 완료한 후 프린터기를 이용해 판매 영수증을 출력한다.

POST는 판매를 완료한 후 재고 서버의 상품 수량 정보를 업데이트 한다.

POST는 환불 상황에서 영수증의 바코드를 바코드 스캐너로 스캔하여 환불 가능한 영수증인지 판단할 수 있다.

POST는 환불을 완료한 후 고객 화면과 캐시 화면에 각각 환불 정보를 출력한다. POST는 환불을 완료한 후 프린터기를 이용해 환불 영수증을 출력한다.

POST는 환불을 완료한 후 재고 서버의 상품 수량 정보를 업데이트 한다.

POST는 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. 사용자 특징

POST 사용자는 캐시이다.

사용자는 언제든지 전원을 끌 수 있다.

사용자는 바코드 스캐너를 이용해 상품, 영수증을 스캔할 수 있다.

2.4. 제약 및 가정 사항

POST의 캐시 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 가

정한다.(현실시간 1초→가상시간 8분) POST 프로그램에서 시각 표현이 필요한 부분은 “YYYYMMDDhhmm”으로 “연월일시분” 으로 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
아이스크림	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 초기화 한다.

POST의 전원이 켜지면 자동으로 재고 서버와 상품을 동기화한다.

3. 세부 요구 사항

3.1. 외부 인터페이스

3.1.1. 사용자 인터페이스

입력 : 캐셔 화면(터치 스크린), 바코드 스캐너
출력 : 캐셔 화면(터치 스크린), 고객 화면, 영수증

3.1.2. HW 인터페이스

입력 : 바코드 스캐너 센서, 캐셔 화면(터치 스크린) 센서
출력 : 캐셔 화면(터치 스크린), 고객 화면, 영수증 프린터



3.1.3. SW 인터페이스

입력: 재고 정보
출력: 재고 정보

3.1.4. 통신 인터페이스

3.2. 기능 요구사항

3.2.1. 판매

3.2.1.1. Function

POST는 기본화면에서 판매를 진행할 수 있다.

POST는 바코드 센서에서 상품의 바코드를 읽을 수 있다.

POST는 바코드로 읽은 상품이 판매 목록에 없는 경우 바코드로 읽은 상품을 판매 목록에 추가하고, 판매 목록에 있는 경우 해당 상품의 개수를 늘린다.

해당 정보는 캐시 화면(터치 스크린)과 고객 화면에 각각 업데이트 된다.

POST는 캐시 화면(터치 스크린)을 통해 판매 목록에 있는 상품의 개수를 추가, 감소 시키거나 상품을 목록에서 제거 할 수 있다. v

해당 정보는 캐시 화면(터치 스크린)과 고객 화면에 각각 업데이트 된다.

POST는 판매하려는 상품의 개수가 상품의 재고 개수보다 부족할 경우 캐시 화면(터치 스크린)에 “XX 개수가 부족합니다” 라는 메시지를 띄운다.

POST는 캐시 화면(터치 스크린)을 통해 고객에게 받은 금액을 입력할 수 있고, 결제를 진행 할 수 있다.

POST는 결제가 완료되면 캐시 화면(터치 스크린)과 고객 화면에 각각 판매 정보를 띄운다.

캐시 화면(터치 스크린)	
Display example	
총 판매액	20,000
현금	30,000
거스름돈	10,000
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500
...	...
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000

고객 화면	
Display example	
총 판매액	20,000
현금	30,000
거스름돈	10,000

POST는 판매가 완료되면 영수증 프린터기를 이용해 판매 영수증을 출력한다.

영수증 양식
영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)
날짜 : YYYY.MM.DD (년.월.일)
판매 상품
상품, 단가, 수량, 판매 금액
...
상품, 단가, 수량, 판매 금액
총 판매액 : x

(판매 영수증을 출력하는 것은 sale_YYYYMMDDhhmm.txt (년월일시분) 파일을 생성하는 것으로 대체한다.)

POST는 결제가 완료되면 재고 서버의 상품의 개수를 업데이트 한다.
(재고 서버의 당일 상품 파일(YYYYMMDD_product.txt)의 상품의 개수를 업데이트 하는 것으로 대체한다.)

POST는 결제가 완료되면 재고 서버의 판매 정보를 추가한다.
(재고 서버의 당일 판매 목록 파일(YYYYMMDD_sale_manage.txt)에 현재 판매정보를 추가하는 것으로 대체한다.)

3.2.1.2. Input

캐시 화면(터치 스크린)
상품의 재고
바코드 스캐너

3.2.1.3. Output

화면 출력
영수증 판매
상품 정보

3.2.2. 환불

3.2.2.1. Function

POST는 사용자가 캐시 화면(터치 스크린)의 기본 화면에서 환불 버튼을 누를 시 환불을 진행할 수 있다.

POST는 바코드 센서에서 영수증 바코드를 읽었을 때 바코드로 읽은 영수증 이 이전에 발급 했던 영수증인지 재고 서버에 확인을 요청한다.

재고 서버에서 환불이 가능하다는 응답이 올 경우, 터치스크린을 통해 환불 을 완료할 수 도 있고 환불을 취소할 수도 있다.

재고 서버에서 환불이 불가능하다는 응답이 올 경우, 캐시 화면(터치 스크린) 와 고객 화면에 각각 “해당 영수증은 환불이 불가능 합니다.” 라는 메시지를 띄운다.

POST는 환불이 완료되면 캐시 화면(터치 스크린)와 고객 화면에 각각 환불 정보를 띄운다.

캐시 화면(터치 스크린)	
Display example	
판매 날짜	2017년 09월 07일 05시 20분
환불 금액	20,000
상품	수량 : 5 / 단가 : 500 / 판매 금액 : 2,500
...	...
상품	수량 : 4 / 단가 : 1,000 / 판매 금액 : 4,000

고객 화면	
Display example	
판매 날짜	2017년 09월 07일 05시 20분
환불 금액	20,000

POST는 환불이 완료되면 영수증 프린터기를 이용해 환불 영수증을 출력한다.

영수증 양식
영수증 번호 : YYYY.MM.DD.hh.mm (년.월.일.시.분)
날짜 : YYYY.MM.DD (년.월.일)
환불 상품
상품, 단가, 수량, 판매 금액
...
상품, 단가, 수량, 판매 금액
환불 금액 : x

(환불 영수증을 출력하는 것은 refund_YYYYMMDDhhmm.txt (년월일시분) 파일을 생성하는 것으로 대체한다.)

POST는 환불이 완료되면 재고 서버의 상품의 개수를 업데이트 한다.
(재고 서버의 당일 상품 파일(YYYYMMDD_product.txt)의 상품의 개수를 업데이트 하는 것으로 대체한다.)

3.2.2.2. Input

캐셔 화면(터치 스크린)
판매 정보
바코드 스캐너

3.2.2.3. Output

화면 출력
영수증
환불 상품 정보

3.2.3. 정산

3.2.3.1. Function

POST는 매일 12시가 되면 하루 동안 판매 했던 내역을 재고 서버에서 받아와 영수증 프린터기를 이용해 출력한다.

정산 영수증 양식
날짜 : YYYY.MM.DD (년.월.일)
판매 상품
상품, 단가, 수량, 금액
...
상품, 단가, 수량, 금액
판매 금액 :
환불 상품
상품, 단가, 수량, 금액
...
상품, 단가, 수량, 금액
환불 금액 :

(정산을 출력하는 것은 settle_YYYYMMDD.txt (년월일) 파일을 생성하는 것으로 대체한다.)

정산은 POST가 기본 화면인 경우에서만 진행이 되며, 판매 또는 환불 중에는 완료가 된 후에 정산이 진행된다.

3.2.3.2. Input

판매 정보

3.2.3.3. Output

정산 영수증

3.2.4. 재고 확인

3.2.4.1. Function

POST는 기본화면에서 재고 확인 버튼 입력 시 재고 확인을 할 수 있다.

POST는 재고 서버에서 당일 재고 정보를 받아와 동기화를 하고 캐시 화면 (터치 스크린)에 재고 정보를 출력하고 영수증 프린터기를 이용해 재고 정보를 출력한다.

재고 확인 양식
날짜 : YYYY.MM.DD.hh.mm (년.월.일.시.분)
상품, 단가, 재고
...
상품, 단가, 재고

(재고 확인을 출력하는 것은 stock_YYYYMMDD.txt (년월일) 파일을 생성하는 것으로 대체한다.)

POST는 판매 중이거나 환불 중에는 재고 확인을 할 수 없다.

3.2.4.2. Input

터치스크린

재고 정보

3.2.4.3. Output

화면 출력

영수증

3.3. 성능 요구사항

버튼에 대한 반응 속도는 1ms 이하이다.

3.4. 설계 제약사항

SASD 개발 방법론을 이용하여 분석 및 설계한다.

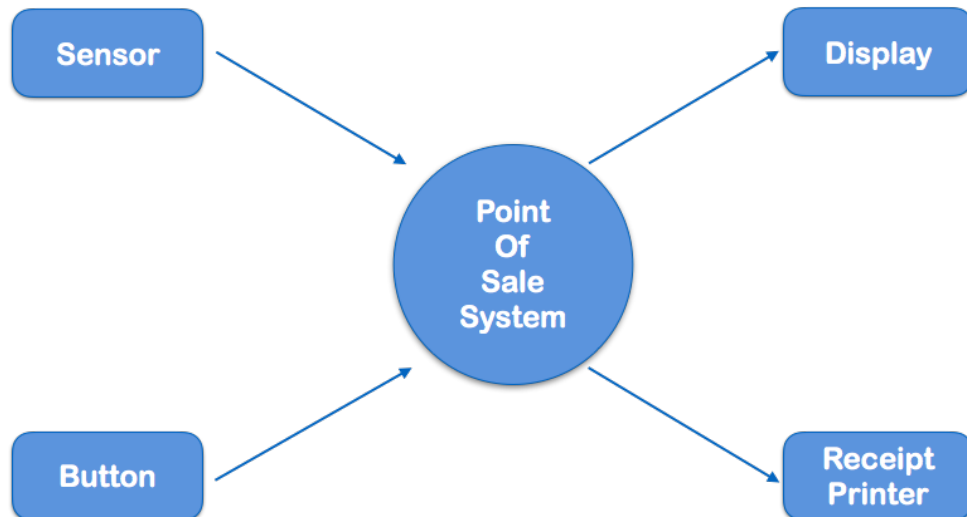
3.5. 소프트웨어 속성

3.6. 기타 요구사항

4. Analysis

4.1. System Context Diagram

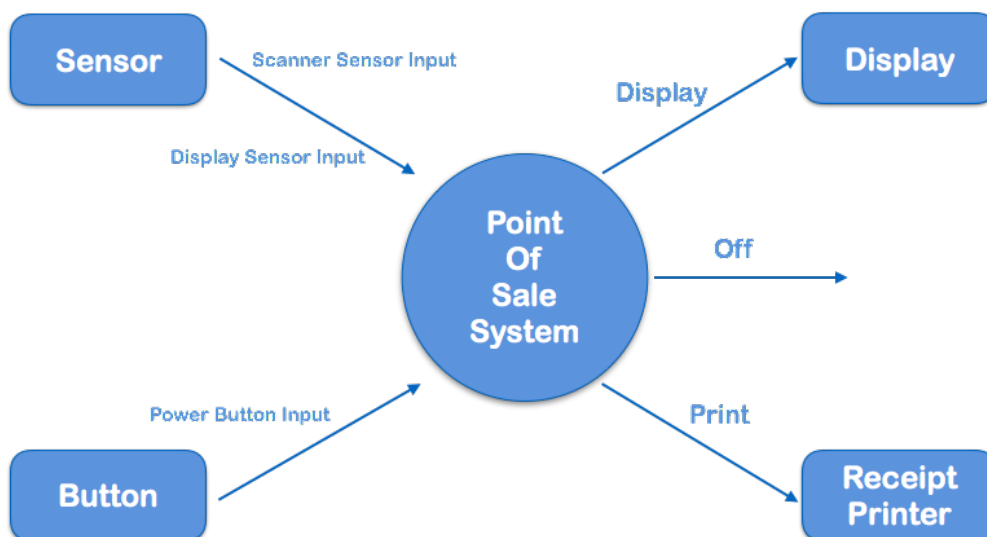
4.1.1. Basic System Context Diagram



4.1.2. Event List

Input/ Output Event	Description
Scanner Sensor Input	바코드 정보를 읽는다.(상품 바코드, 영수증 바코드)
Display Sensor Input	캐셔 화면 입력 정보(상품 개수, 상품 목록제거, 받은 금액, 환불, 재고확인)
Power Button Input	전원 버튼으로 POST를 종료한다.
Print	영수증(판매, 환불, 정산)을 프린터에서 파일로 출력한다.
Display	캐셔 화면, 고객 화면을 출력한다.
Off	Turn Off

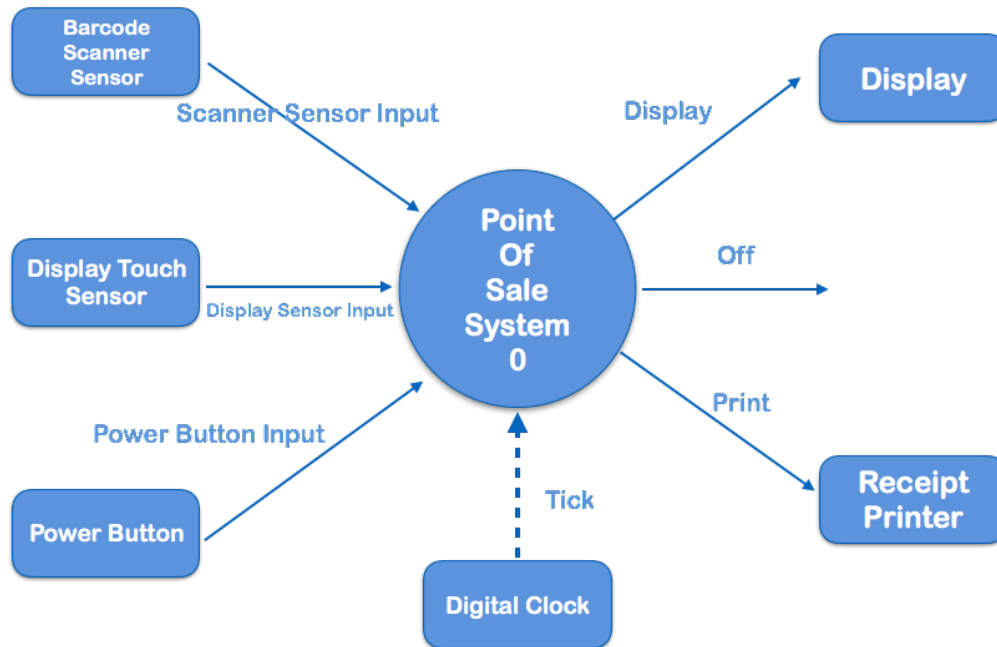
4.1.3. The System Context Diagram



4.2. Data Flow Diagram

4.2.1. DFD level 0

4.2.1.1. DFD



4.2.1.2. Process Specification

4.2.1.2.1. Process 0

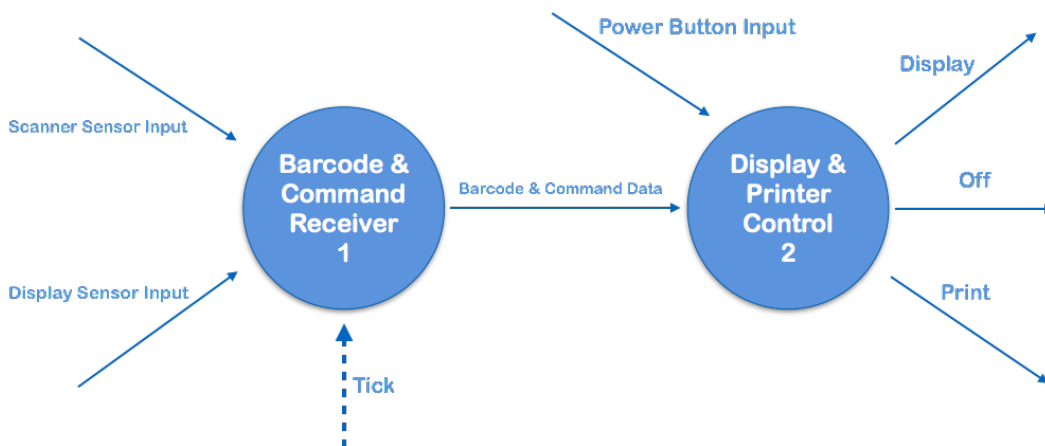
Reference No.	0
Name	Print Of Sale System
Input	Scanner Sensor Input Display Sensor Input Power Button Input Tick
Output	Display Print Off
Process Description	Scanner 와 Display Sensor들에서 받아온 데이터를 가지고 Print Of Sale System안에서 처리하여 Display에 화면을 출력하고 Receipt Printer에서 영수증을 출력한다. 또한, Power Button Input을 받아서 종료 한다.

4.2.1.3. Data Dictionary

Input/ Output Event	Description	Format / Type
Scanner Sensor Input	바코드 정보를 읽는다.	String
Display Sensor Input	캐시 화면 입력 정보(상품 개수, 상품 목록제거, 받은 금액, 환불, 재고확인)	Command / String
Power Button Input	전원 버튼으로 POST를 종료한다.	True/ False
Print	영수증(판매,환불,정산)을 프린터로 출력한다.	File
Display	캐시 화면, 고객 화면을 출력한다.	Structure
Off	Turn Off	Off

4.2.2. DFD Level 1

4.2.2.1. DFD



4.2.2.2. Process Specification

4.2.2.2.1. Process 1

Reference No.	1
Name	Barcode & Command Receiver
Input	Scanner Sensor Input Display Sensor Input
Output	Barcode & Command Data
Process Description	Scanner 와 Display Sensor들에서 받아온 Data를 가지고 Barcode & Command Receiver 안에서 처리하여 Display & Printer Control에 Barcode & Command Data를 전달한다.

4.2.2.2.2. Process 2

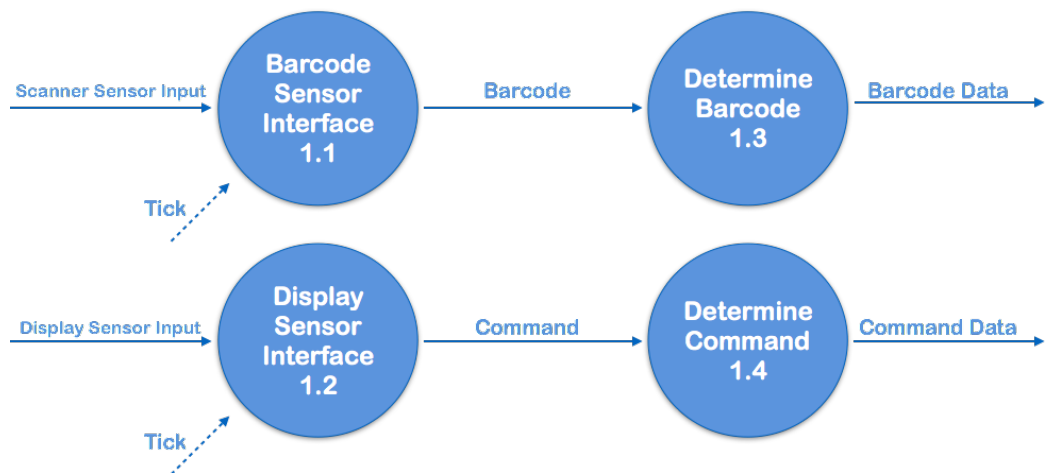
Reference No.	2
Name	Display & Printer Control
Input	Barcode & Command Data Power Button Input
Output	Display Print Off
Process Description	Barcode & Command Receiver 부터 받아온 Data를 가지고 처리하여 Display와 Printer에 명령을 내리고 Power Button Input을 받아 Off 할 수 있다.

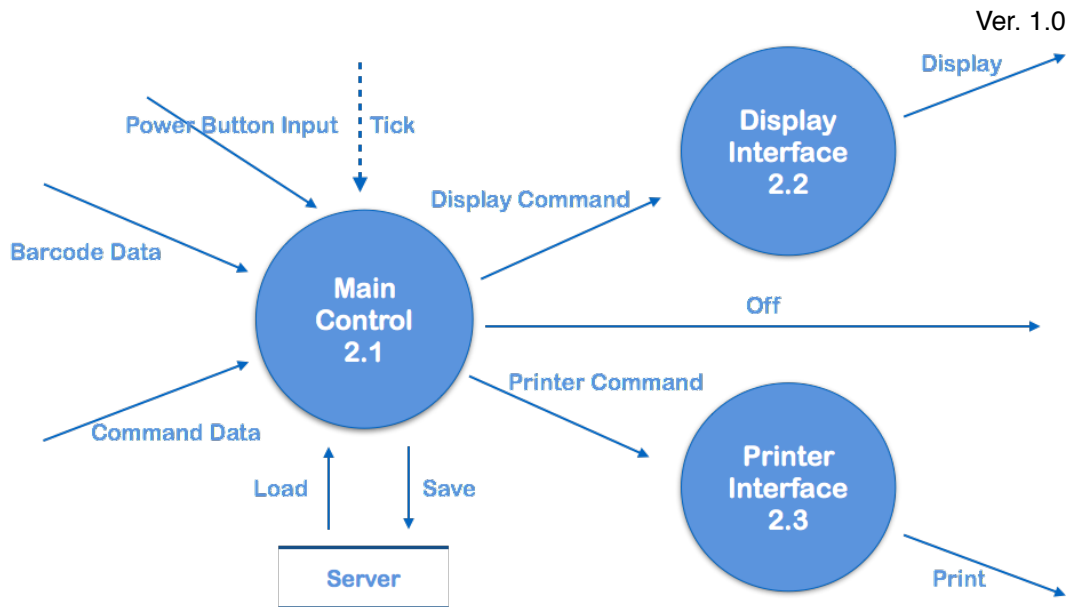
4.2.2.3. Data Dictionary

Input/ Output Event	Description	Format / Type
Barcode Data	입력 받은 Barcode 가 영수증 정보인지 상품정보인지 결정된 Data # char kind : 바코드의 종류 (0 = 상품 , 1 = 영수증) # int b_num : 바코드의 번호 (1 = 과자, 2 = 아이스크림, 3 = 과일, 4 = 물, 5 = 라면, 6 = 음료수, 7 = 커피, 그 외에는 영수증의 바코드 번호)	Structure
Command Data	입력 받은 Command 가 상품개수를 변경하는지 상품 목록제거를 하는지, 받은 금액을 입력하는지, 환불하는지, 재고확인 하는지 결정된 Data # int cmd : 명령어의 종류 (0 = 판매, 1 = 환불, 2 = 재고확인, 3 = 상품개수 변경 , 4 = 금액) # int p_num : cmd가 3인경우 변경될 상품개수 # int price : cmd가 4인경우 받은 금액	Structure

4.2.3. DFD Level 2

4.2.3.1. DFD





4.2.3.2. Process Specification

4.2.3.2.1. Process 1.1

Reference No.	1.1
Name	Barcode Sensor Interface
Input	Scanner Sensor Input Tick
Output	Barcode
Process Description	Scanner Sensor Input이 String변수 형태로 들어왔는지 확인한다.

4.2.3.2.2. Process 1.2

Reference No.	1.2
Name	Display Sensor Interface
Input	Display Sensor Input Tick
Output	Command
Process Description	Display Sensor Input이 String변수 형태로 들어왔는지 확인한다.

4.2.3.2.3. Process 1.3

Reference No.	1.3
Name	Determine
Input	Barcode
Output	Barcode Data
Process Description	입력 받은 정보를 해석해서 상품정보인지 영수증정보인지 판단하여 Data로 변환한다.

4.2.3.2.4. Process 1.4

Reference No.	1.4
Name	Determine Command
Input	Command
Output	Command Data
Process Description	Display에서 받은 명령 정보를 가지고 상품수량을 늘리거나 상품목록을 삭제, 환불, 재고확인, 현금 입력인지를 판단하기 위한 Command Data로 변환한다.

4.2.3.2.5. Process 2.1

Reference No.	2.1
Name	Main Control
Input	Barcode Data Command Data Power Button Input Load Tick
Output	Display Command Printer Command Off Save
Process Description	입력 받은 정보들을 변환하여서 각각 적절한 Command를 보내고 Server에 Save를 한다. 또한 Power Button Input을 받아 Off 한다.

4.2.3.2.6. Process 2.2

Reference No.	2.2
Name	Display Interface
Input	Display Command
Output	Display
Process Description	Display Command를 받아 캐시 화면과 고객 화면에 정보를 표시한다.

4.2.3.2.7. Process 2.3

Reference No.	2.3
Name	Printer Interface
Input	Printer Command
Output	Print
Process Description	받은 정보들을 처리해서 영수증/재고확인표를 출력한다.

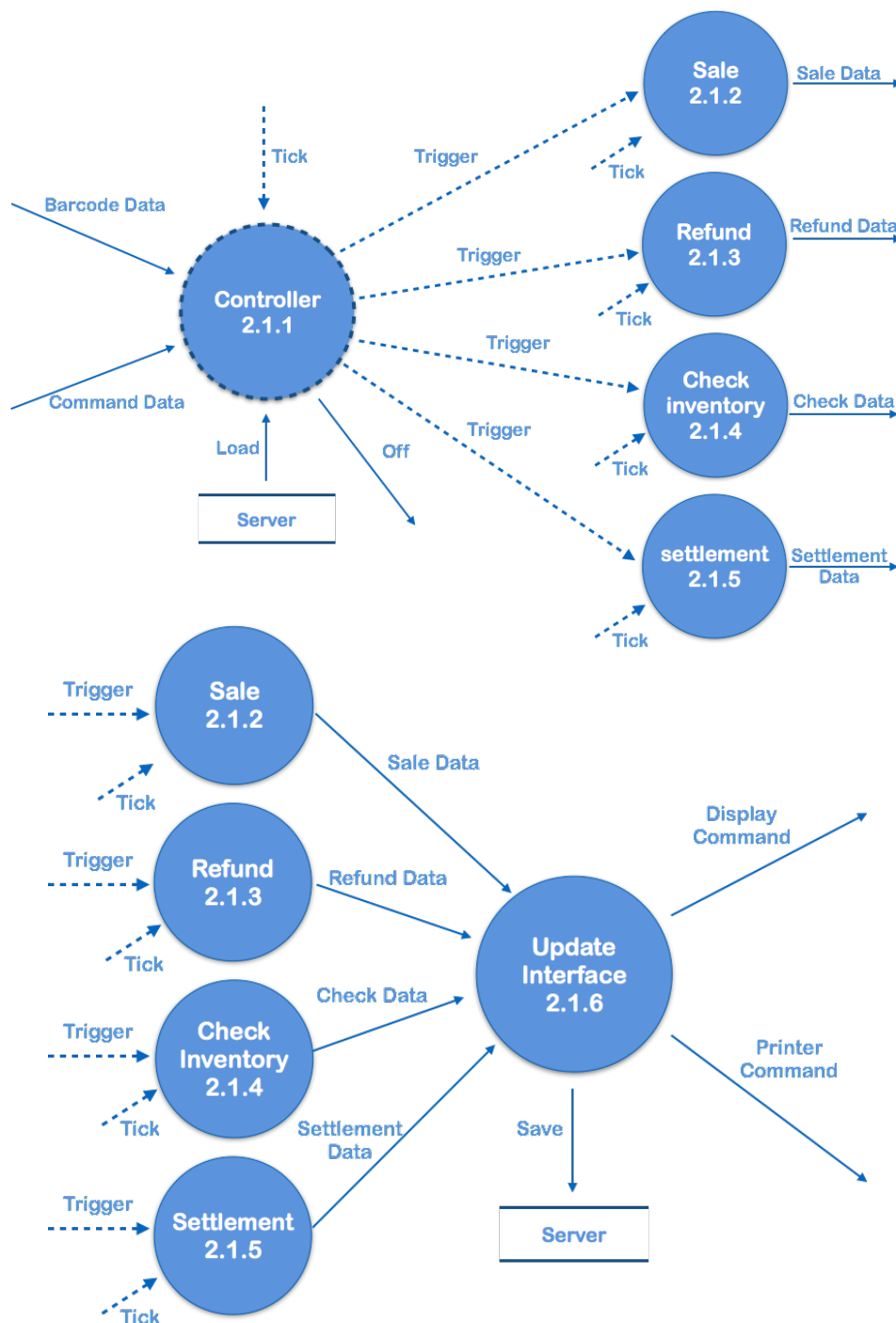
4.2.3.3. Data Dictionary

Input/ Output Event	Description	Format / Type
Barcode	Scanner Sensor Input을 Barcode Sensor Interface에서 알맞은 입력인지 판단된 Input	String
Command	Display Sensor Input을 Display Sensor Interface에서 알맞은 입력인지 판단된 Input	String
Display Command	캐시, 고객 화면에 표시하게 될 정보 # int total_amt : 총 판매액 # int price : 받은 금액 # int change : 거스름돈 # struct* Product (linked list, 상품명, 상품개수,가격, 상품의 판매금액) : 판매정보	Structure
Printer Command	Printer로 표시하게 될 정보 # char* rec_n: 영수증 번호 (날짜,시간) # char* data : 날짜 # struct* Product : 판매정보 # int total_amt : 총 판매액	Structure

Load	서버에서 영수증 정보, 상품 재고 정보를 받아온다. # int inventory[10] : 상품의 재고 정보 (e.g. inventory[1] : 과자의 재고) # File receipt : 영수증 정보	Structure
Save	서버에 판매 정보, 상품 재고 변동 정보를 업데이트 한다. # struct *Product : 판매 정보	Structure

4.2.4. DFD Level 3

4.2.4.1. DFD



4.2.4.2. Process Specification

4.2.4.2.1. Process 2.1.1

Reference No.	2.1.1
Name	Controller
Input	Barcode Data Command Data Power Button Input Load Tick
Output	Trigger Off
Process Description	Command Data를 바탕으로 주기 마다 상황에 맞는 Trigger 신호를 보낸다.

4.2.4.2.2. Process 2.1.2

Reference No.	2.1.2
Name	Sale
Input	Trigger Tick
Output	Sale Data
Process Description	주기 마다 Trigger값이 바뀌면서 Triggering 될 때 Update Interface에 Sale Data를 보낸다.

4.2.4.2.3. Process 2.1.3

Reference No.	2.1.3
Name	Refund
Input	Trigger Tick
Output	Refund Data
Process Description	주기 마다 Trigger값이 바뀌면서 Triggering 될 때 Update Interface에 Refund Data를 보낸다.

4.2.4.2.4. Process 2.1.4

Reference No.	2.1.4
Name	Check inventory
Input	Trigger Tick
Output	Check Data
Process Description	주기 마다 Trigger값이 바뀌면서 Triggering 될 때 Update Interface에 Check Data를 보낸다.

4.2.4.2.5. Process 2.1.5

Reference No.	2.1.5
Name	Settlement
Input	Trigger Tick
Output	Settlement Command
Process Description	주기 마다 Trigger값이 바뀌면서 Triggering 될 때 Update Interface에 Settlement Data를 보낸다.

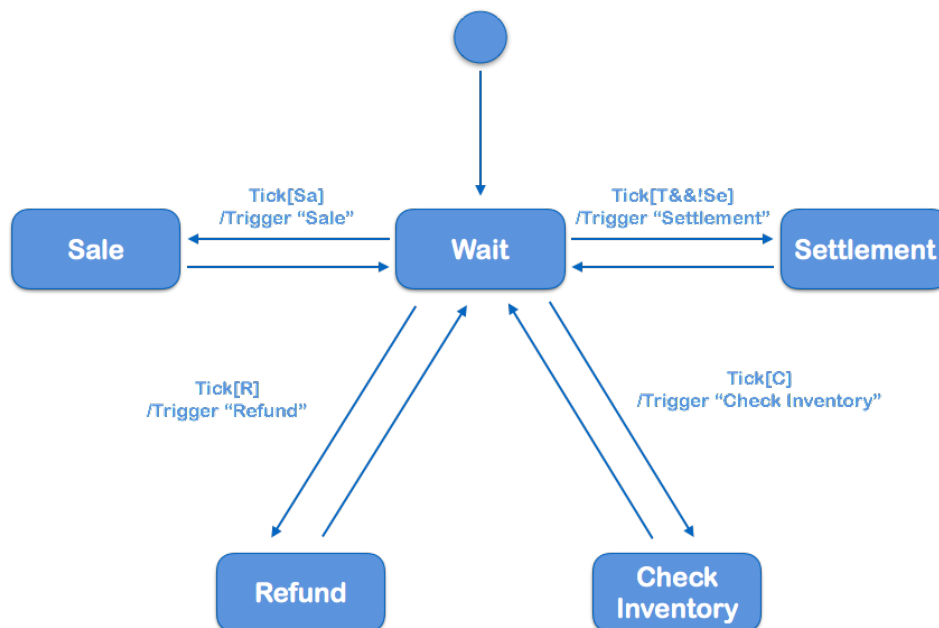
4.2.4.2.6. Process 2.1.6

Reference No.	2.1.6
Name	Update Interface
Input	Sale Data Refund Data Check Data Settlement Command
Output	Display Command Printer Command Save
Process Description	입력 받은 Data들을 종합 하여서 상품 재고 정보들을 Server에 Save를 하고 Display Interface와 Printer Interface에 명령을 내린다.

4.2.4.3. Data Dictionary

Input/ Output Event	Description	Format / Type
Sale Data	판매에서 처리된 Data (판매된 상품명, 수량, 판매금액) # Product *sale : 판매정보	Structure
Refund Data	환불에서 처리된 Data (환불여부, 환불 상품명, 수량, 환불금액) # int refund : 환불 여부 # Product *rec : 영수증의 판매 정보	Structure
Check inventory	재고 Data # int inventory[10] : 상품의 재고 정보 (e.g. inventory[1] : 과자의 재고)	Array
Settlement	하루동안 판매, 환불 된 Data # Product *total : 판매정보 + 환불정보	Structure

4.2.4.4. State Transition Diagram (*Controller*)



4.2.5. Overall DFD

