

SoftwareEngineering Term Project 1,2

1팀

201512087 김상훈

201512109 서석빈

201512116 안원석

201512135 정상우

201512142 주민종

담당교수님 : 최선한 교수님

제출일자 : 2019.11.03

**Console 기반의 POS(Point Of Sales Systems, 판매시점정보관리 시스템) 개발 요구사항 명세서**

목차

1) 서론

2) 목표

3) 기능 기술

4) 동적 기술

5) 정보 기술

6) 검증 기준

7) 유스케이스 다이어그램

8) 클래스 다이어그램

9) 시퀸스 다이어그램

10) 테스트 결과

**1. 서론**

POS(Point of sale system) 프로젝트의 목표는 다음과 같다. POS 시스템에 대한 요구사항을 사용자의 입장에서 도출하여 요구사항 명세서를 작성한다. 그리고 이를 바탕으로 객체지향 프로그래밍으로 구현하는 것을 목표로 한다. 본 문서는 위 프로젝트에 대한 요구사항 명세서이며, 이는 프로젝트 평가의 척도가 된다. 이후 수정 시 프로젝트 팀원 모두의 동의가 필요하다.

**2. 목표**

**2.1 독자**

본 문서는 독자를 다음과 같이 규정하며, 따로 범위가 설정되어 있지 않은 경우 문서 전체를 지정하는 것으로 한다.

2.1.1 소프트웨어 공학 교과목 교수님 및 조교님

2.1.2 해당 프로젝트 팀원

2.1.3 소프트웨어 공학 교과목 수강 인원

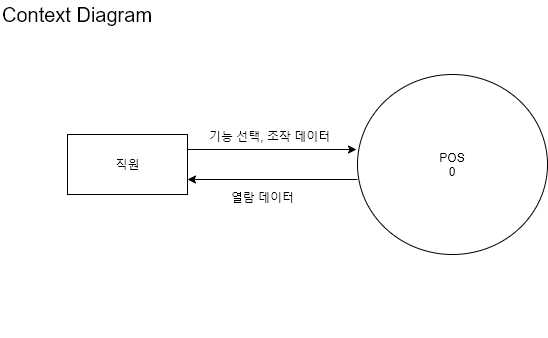
**2.2 전체 기술**

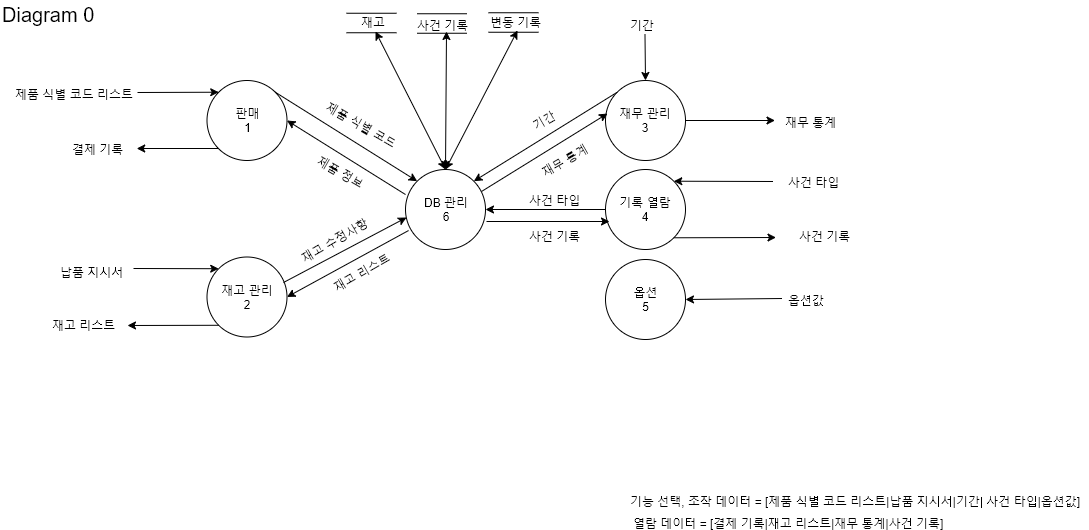
아래에 도표의 형태로 기술하였다.

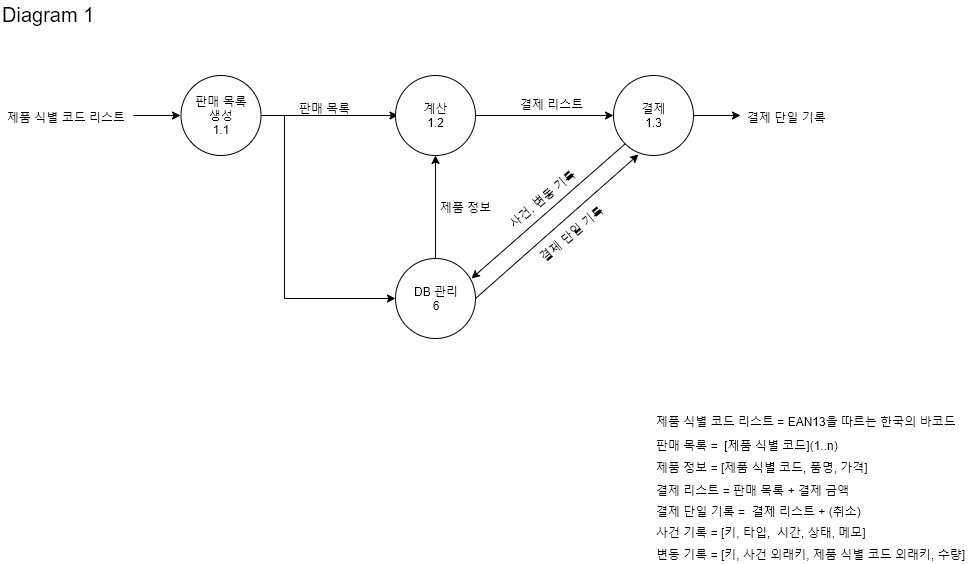
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **영역** | **사용자** | **화면** | **요구사항** | **제약조건** |
| **로컬 서버** | **전체** | **백** | **로컬 서버 내의 모든 데이터는 데이터 베이스 혹은 파일에 저장한다.** | **읽기와 쓰기는 실시간으로 이루어져야 한다.** |
| **백** | **여러 계산용 단말에 연결 가능해야 한다.** | **다중 연결을 지원해야 한다.** |
| **백** | **모든 제품의 상세 데이터를 가지고 있다.** | **제품명, 수량, 바코드, 금액** |
| **백** | **제품의 상세 데이터를 조작 가능해야 한다.** | **위에 상기한 항목과 추가, 삭제 기능을 구현하여야 한다.** |
| **판매원** | **판매** | **판매할 물건은 바코드나 코드로 입력 받을 수 있다.** | **콘솔 입력도 가능하여야 한다.** |
| **판매** | **데이터베이스에 존재하는 물건에 대해서 판매가 가능하여야 한다.** | **없는 물품일 경우 None을 추가하고, 이를 삭제할 수 있어야 한다.** |
| **판매** | **결제를 대기중인 품목 리스트는 수정이 가능하다.** |  |
| **판매** | **위의 리스트에서 품목의 수량은 수정 가능하다.** | **+ <숫자> : 제품의 개수를 <숫자>만큼 증가, 연속해서 누를 시 두번째부터 1개씩 증가** |
| **\* <숫자> : 제품의 개수를 <숫자>로 변경** |
| **- <숫자> : 제품의 개수를 <숫자>만큼 감소, 연속해서 누를 시 두번째부터 1개씩 감소** |
| **수량이 0이 될 경우, 리스트에서 삭제한다.**  **단 음수일 경우 보존한다.** |
| **결제** | **계산이 가능하여야 한다.** | **1초 내에 이루어져야 한다.**  **재고가 결제 수량보다 적을 경우 음수의 가상재고를 가지고**  **이는 금액도 마찬가지이다.** |
| **결제 기록** | **영수증이 저장되어야 한다.** |  |
| **결제 단일 기록** | **영수증의 상세내용을 확인할 수 있어야 한다.** |  |
| **납품업자** | **재고관리** | **모든 제품의 상세 데이터를 조작할 수 있다.** |  |
| **재고관리** | **납품관리 기능을 수행할 수 있다.** | **제품 추가, 삭제 기능을 구현하여야 한다.** |
| **사장** | **재무관리** | **매출액을 표시한다.** | **기간별 매출액 표시를 지원한다. (연, 월, 일)** |
| **전체** | **콘솔창** | **비상 상황을 대비하여**  **서버의 모든 기능을 간단히 구현한다.** | **결제, 결제기록, 납품, 납품기록, 통계 기능을 간단히 구현한다.** |
| **계산용 단말** | **전체** | **백** | **모바일 기기를 사용할 수 있다.** |  |
| **백** | **이전에 연결한 단말은 자동연결을 지원해야한다.** |  |
| **백** | **IP/PORT - PASSWORD로 접속하며, 사용자는 분류하지 않는다.** |  |
| **옵션** | **텍스트 크기를 변경할 수 있다.** | **단 0은 되지 않아야 한다.** |
| **판매원** | **판매** | **자체적으로 계산이 가능하여야 한다.** | **로컬 서버의 기능 구현** |
| **실시간으로 계산(결제) 내역이 로컬서버에 반영되어야 함** |
| **결제를 완료하기 전까지 계산은 끝나지 않음** |
| **납품업자** | **재고관리** | **로컬 서버의 기능을 구현한다.** |  |
| **사장** | **재무관리** | **로컬 서버의 기능을 구현한다.** |  |

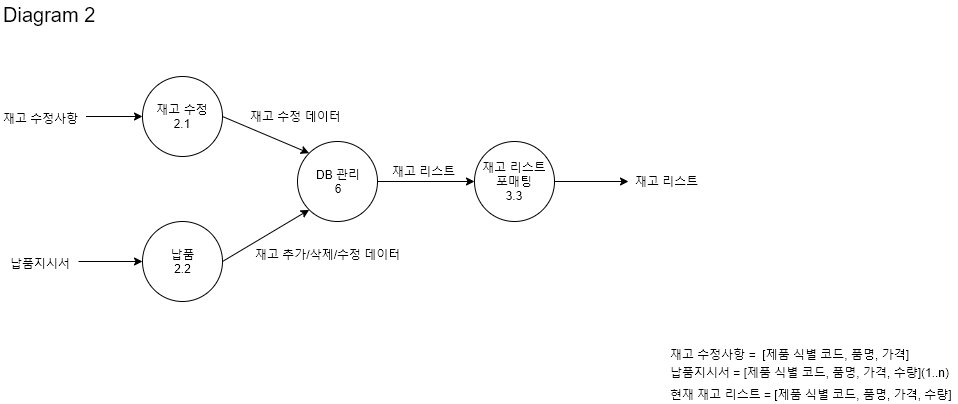
**3.기능 기술**

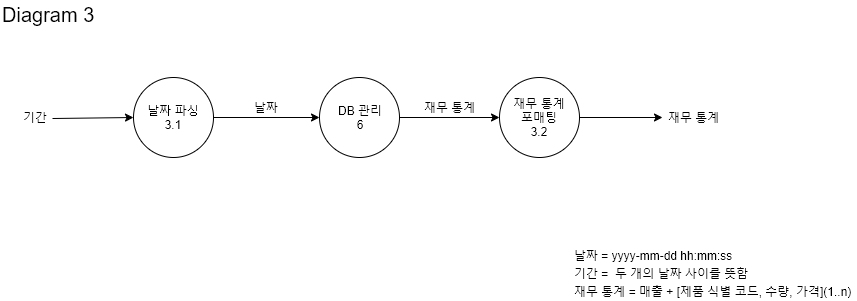
3.1 자료 흐름도

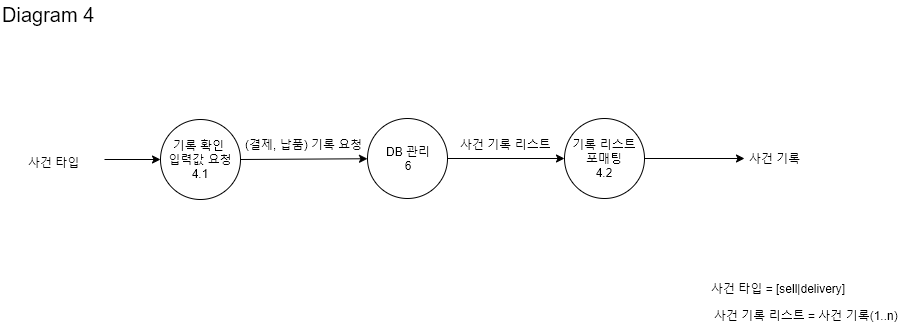


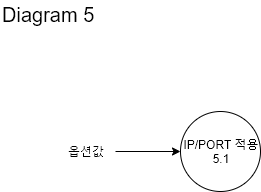


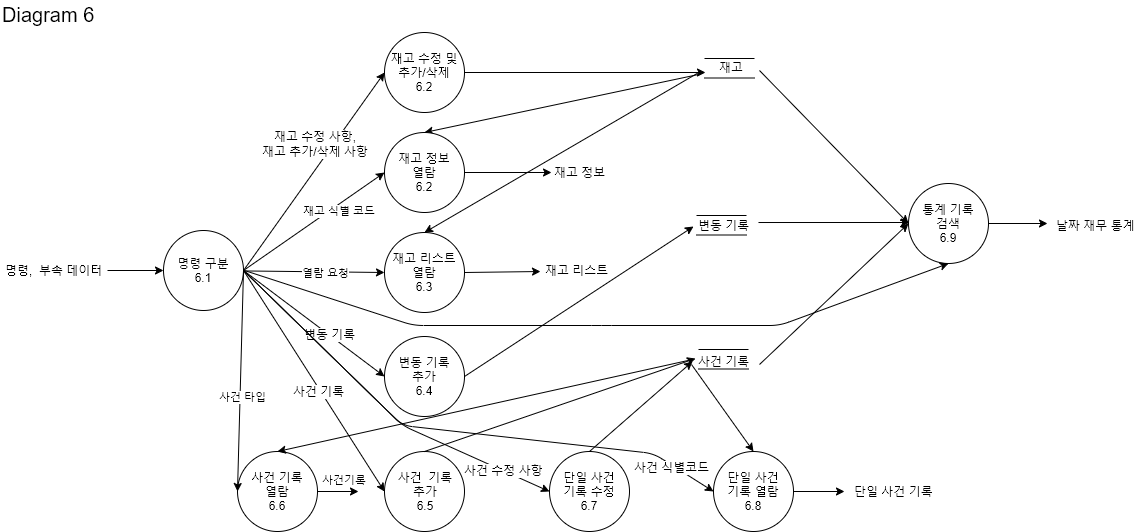












3.2 프로세스 명세서

|  |
| --- |
| **Name** |
| 1.1 |
| **TITLE** |
| 판매목록 생성 |
| **INPUT/OUTPUT** |
| stockKeyList  /  sellList |
| **BODY** |
| /\*  이 프로시저는 받은 재고 데이터로 판매목록을 생성한다.  \*/  do  if stockKey  stockInfomation = select \* from stock where stockKey  sellList append stockInfomation  while stockKey was inputted |

|  |
| --- |
| **Name** |
| 1.2 |
| **TITLE** |
| 계산 |
| **INPUT/OUTPUT** |
| sellList, stockInfomation  /  paymentList |
| **BODY** |
| /\*  이 프로시저는 판매목록의 금액을 합산하여 결제목록을 생성한다.  \*/  foreach sellList  sum = stockInfomation.price \* sellList.number  paymentList = sellList + paymentList |

|  |
| --- |
| **Name** |
| 1.3 |
| **TITLE** |
| 결제 |
| **INPUT/OUTPUT** |
| paymentList  /  singlePaymentHistory |
| **BODY** |
| /\*  이 프로시저는 결제목록을 받아 결제를 진행하고 결과를 사건 기록에 저장한다.  \*/  if payment amount  singlePaymentHistory = paymentList    sellList = null  paymentList = null |

|  |
| --- |
| **Name** |
| 2.1 |
| **TITLE** |
| 재고 수정 |
| **INPUT/OUTPUT** |
| changeStockInformation  /  parsedChangeStockInformation |
| **BODY** |
| /\*  재고 수정 데이터를 받아 DB 쿼리에 맞도록 파싱한다.  \*/  parsedChangeStockInformation.name = parse name in changeStockInformaion  parsedChangeStockInformation.price = parse price in changeStockInformaion  parsedChangeStockInformation.key = parse key in changeStockInformaion |

|  |
| --- |
| **Name** |
| 2.2 |
| **TITLE** |
| 납품 |
| **INPUT/OUTPUT** |
| delivery order  /  parsedStockInformation |
| **BODY** |
| /\*  납품지시서 안의 재고 추가/삭제 및 수정 데이터를 파싱한다.  .  \*/  foreach delivery order  parsedChangeStockInformation.name = parse name in changeStockInformaion  parsedChangeStockInformation.price = parse price in changeStockInformaion  parsedChangeStockInformation.key = parse key in changeStockInformaion  parsedChangeStockInformation.amount = parse amount in changeStockInformation |

|  |
| --- |
| **Name** |
| 3.3 |
| **TITLE** |
| 재고 리스트 포매팅 |
| **INPUT/OUTPUT** |
| currentStockList  /  formatted stockList |
| **BODY** |
| /\*  재고 리스트를 화면 출력 UI에 맞게 포매팅한다.  \*/ |

|  |
| --- |
| **Name** |
| 3.1 |
| **TITLE** |
| 날짜 파싱 |
| **INPUT/OUTPUT** |
| duration  /  date |
| **BODY** |
| /\*  기간을 입력받아 DB 쿼리에 맞게 파싱한다.  \*/  if duration is day  date = today  else if duration is month  date = this month  else if duration is year  date = this year  else if duration is day between day  date = day concat day |

|  |
| --- |
| **Name** |
| 3.2 |
| **TITLE** |
| 재무통계 포매팅 |
| **INPUT/OUTPUT** |
| date financial statistic  /  Date financial statistic chart |
| **BODY** |
| /\*  재무 통계를 화면 출력 UI에 맞게 포매팅한다.  \*/ |

|  |
| --- |
| **Name** |
| 4.1 |
| **TITLE** |
| 기록 확인 입력값 요청 |
| **INPUT/OUTPUT** |
| event type  /  parsed event type |
| **BODY** |
| /\*  이벤트 타입을 추출하여 해당 타입 이벤트 검색을 호출한다.  \*/  Parsed event type = parse event type |

|  |
| --- |
| **Name** |
| 4.2 |
| **TITLE** |
| 기록 리스트 포매팅 |
| **INPUT/OUTPUT** |
| event list  /  Formatted event list |
| **BODY** |
| /\*  사건 기록 리스트를 화면 출력 UI에 맞게 포매팅한다.  \*/ |

|  |
| --- |
| **Name** |
| 5.1 |
| **TITLE** |
| 옵션 |
| **INPUT/OUTPUT** |
| option value  / |
| **BODY** |
| /\*  입력 받은 옵션값을 적용한다.  \*/  ip = optionValue.ip  port = optionValue.port |

|  |
| --- |
| **Name** |
| 6.1 |
| **TITLE** |
| 명령 구분 |
| **INPUT/OUTPUT** |
| Instruction, data  /  call funtion |
| **BODY** |
| /\*  명령을 파싱하여 해당 프로세스를 호출한다.  \*/  if getStock  getStock  else if editStock  editStock(data)  else if getStocks  getStocks  else if getEvent  getEvent(data)  else if tryChangeEvent  tryChangeEvent(data)  else if getEventList  getEventList(data)  else if getSelling  getSelling(data)  else if addEvent  addEvent(data)  else if addChange  addChange(data) |

|  |
| --- |
| **Name** |
| 6.2 |
| **TITLE** |
| 재고 수정 및 추가/삭제 |
| **INPUT/OUTPUT** |
| changeStockInformation,  createOrDeleteStockInformation  / |
| **BODY** |
| /\*  재고 수정 및 추가 삭제를 DB에 적용  \*/  key = parse key in changeStockInformation  name = parse name in changeStockInformaion  price = parse price in changeStockInformaion  select insert values key, name, price or duplicate key update |

|  |
| --- |
| **Name** |
| 6.3 |
| **TITLE** |
| 재고 정보 열람 |
| **INPUT/OUTPUT** |
| stockKey  /  stockInformation |
| **BODY** |
| /\*  재고 정보를 DB에 요청  \*/  Select \* from stock where stock.key = key |

|  |
| --- |
| **Name** |
| 6.3 |
| **TITLE** |
| 재고 리스트 열람 |
| **INPUT/OUTPUT** |
| request  / |
| **BODY** |
| /\*  재고리스트 열람을 DB에 요청  \*/  select \* from stock |

|  |
| --- |
| **Name** |
| 6.4 |
| **TITLE** |
| 변동 기록 추가 |
| **INPUT/OUTPUT** |
| changeHistory  / |
| **BODY** |
| /\*  변동 기록을 DB에 추가  \*/  eventKey = parse eventKey in changeHistory  changeKey = parse changeKey in changeHistory  amount = parse amount in changeHistory  insert into change values eventKey, changeKey, amount |

|  |
| --- |
| **Name** |
| 6.5 |
| **TITLE** |
| 사건 기록 추가 |
| **INPUT/OUTPUT** |
| event  / |
| **BODY** |
| /\*  사건을 DB에 추가  \*/  type = parse type in changeStockInformation  time = parse time in changeStockInformaion  status = parse status in changeStockInformation  memo = parse memo in changeStockaInformation  insert into event values type, time, status, memo |

|  |
| --- |
| **Name** |
| 6.6 |
| **TITLE** |
| 사건 기록 열람 |
| **INPUT/OUTPUT** |
| eventType  /  eventHistory |
| **BODY** |
| /\*  요청된 타입의 사건 기록을 열람  \*/  select \* from event where type = eventType |

|  |
| --- |
| **Name** |
| 6.7 |
| **TITLE** |
| 단일 사건 기록 수정 |
| **INPUT/OUTPUT** |
| changeSingleEventInformation  / |
| **BODY** |
| /\*  단일 사건 수정을 DB에 적용  \*/  status = parse status in changeSingleEventInformation  key = parse status in changeSingleEventInformation  update event set status where key |

|  |
| --- |
| **Name** |
| 6.8 |
| **TITLE** |
| 단일 사건 기록 열람 |
| **INPUT/OUTPUT** |
| eventKey  /  singleEventHistory |
| **BODY** |
| /\*  단일 사건 기록을 DB에 요청  \*/  select \* from event where key= eventKey |

|  |
| --- |
| **Name** |
| 6.9 |
| **TITLE** |
| 통계 기록 검색 |
| **INPUT/OUTPUT** |
| /  financial statistic |
| **BODY** |
| /\*  DB에서 받은 기록들로 통계 계산  \*/  select key, name, sum(amount) from (event join change) join stock |

3.3 자료 사전

배경흐름도에 기입함.

**4. 동적 기술**

4.1 시나리오

- 일반적인 경우

|  |
| --- |
| <전체>  (옵션)  메인 화면이 출력된다.  옵션 화면에 들어간다.  옵션의 상태가 값을 입력한다.  수정된 옵션 화면이 출력된다.  확인 코드를 입력한다.  메인 화면이 출력된다. |

|  |
| --- |
| <점원> (판매)  메인 화면이 출력된다.  화면에 판매 화면을 열람한다.  물품의 바코드/코드를 입력한다.  화면에 판매 목록이 출력된다.  결제금액을 받아서 결제를 한다.  잔금과 영수증이 출력된다.  화면에 확인창이 출력된다.  확인 코드를 입력한다.  메인 화면이 출력된다.  (결제 기록 확인)  메인 화면이 출력된다.  결제 기록을 열람한다.  화면에 결제 기록이 출력된다.  단일 기록을 열람한다.  화면에 단일 기록이 출력된다.  + (환불)  환불 신청을 한다.  화면에 환불창이 출력된다.  확인 코드를 입력한다.  메인 화면이 출력된다. |

|  |
| --- |
| <사장>  (재무통계)  메인 화면이 출력된다.  재무통계 화면에 들어간다.  화면에 당일 시간별로 통계표가 출력된다.  기간을 입력한다.  화면에 기간에 따른 통계표가 출력된다.  확인 코드를 입력한다.  메인 화면이 출력된다. |

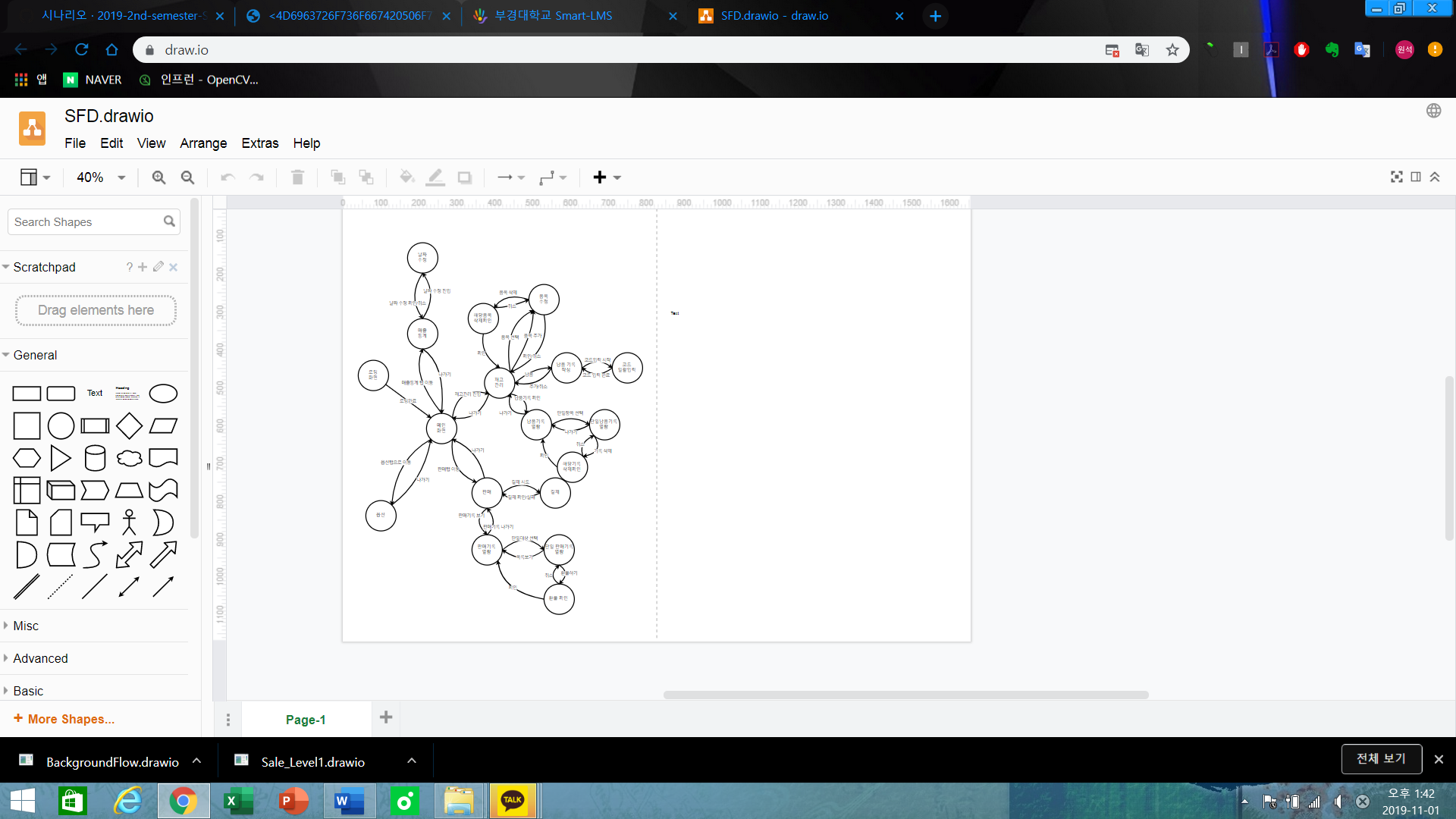
|  |
| --- |
| <납품업자>  (납품)  메인 화면이 출력된다.  납품 관리에 들어간다.  납품 지시서를 입력한다.  재고 리스트가 수정되어 화면에 출력된다.  결과를 확인하고 확인 코드를 입력한다.  메인 화면이 출력된다. |

- 예외적인 경우

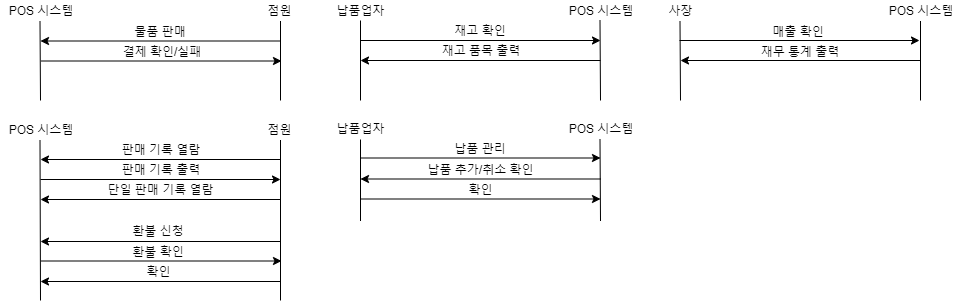
|  |
| --- |
| <점원>  (판매 물품 입력 중, 수량 수정할 때)  메인 화면이 출력된다.  판매 화면에 들어간다.  물품의 바코드/코드를 입력한다.  화면에 판매 목록이 출력된다.  출력된 판매 목록의 수정하고자 하는 제품의 순서 태그를 입력한다.  수량 수정을 입력한다.(\* <숫자>, + <숫자>, - <숫자>)  수정된 판매 목록이 출력된다.  (판매 물품 입력 중, 취소할 때)  메인 화면이 출력된다.  판매 화면에 들어간다.  물품의 바코드/코드를 입력한다.  취소 코드를 입력한다.  메인 화면이 출력된다.  (환불할 때)  메인 화면이 출력된다.  판매 화면에 들어간다.  결제 기록 화면에 들어간다.  환불하고자 하는 단일 기록 화면에 들어간다.  환불 코드를 입력한다.  환불창이 출력된다.  확인 코드를 입력한다.  메인 화면이 출력된다. |

4.2 상태 변화도, 사건 추적도

상태 변화도

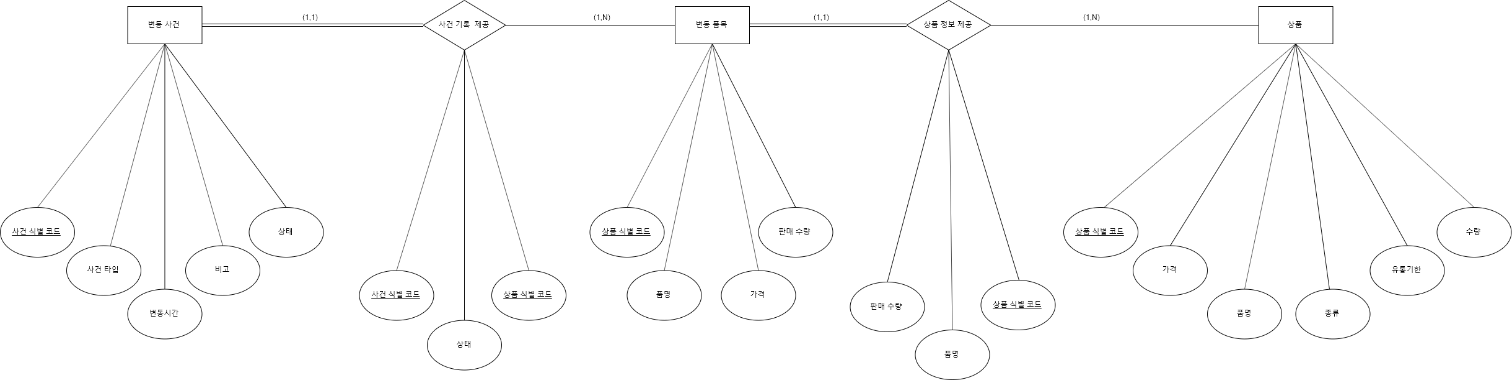


사건 추적도

****

**5. 정보 기술**

5.1 EER



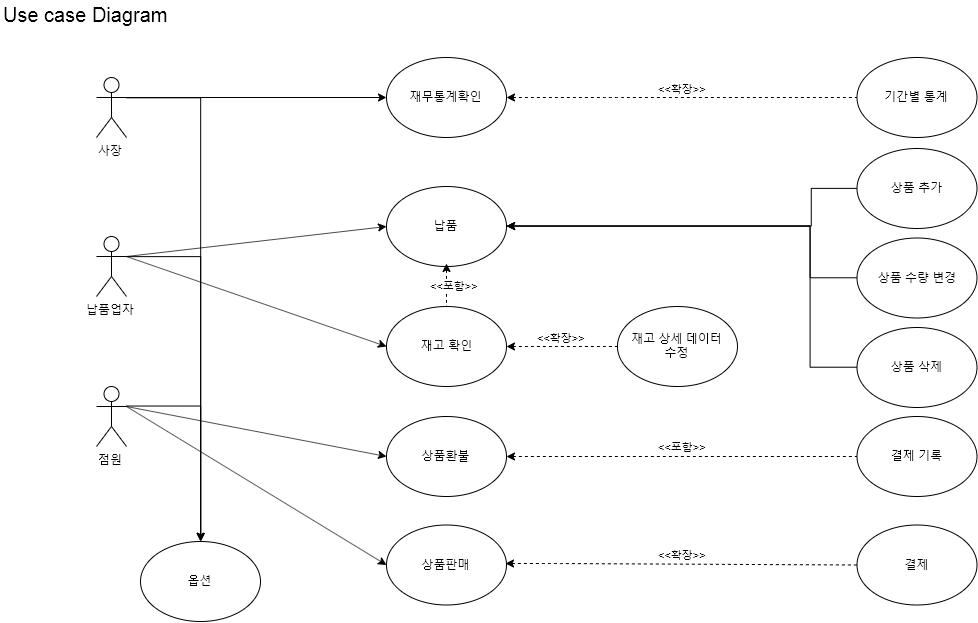
5.2 자료사전

|  |  |  |  |
| --- | --- | --- | --- |
| 이름 | 타입 | 제약조건 | 비고 |
| Event | | | 변동사건 테이블 |
| eKey | VARCHAR(25) | PRIMARY KEY | 사건 식별 코드 |
| eType | BYTE | NOT NULL | 변동타입 |
| eTime | DATETIME | NOT NULL | 변동시간 |
| eState | VARCHAR(3) |  | 상태 |
| eNote | VARCHAR(200) |  | 비고 |
| Change | | | 변동품목 테이블 |
| cKey | VARCHAR(25) | PRIMARY KEY | 키 |
| cEvent | VARCHAR(25) | foreign to 상품 | 사건 외래키 |
| cStock | VARCHAR(25) | foreign to 변동사건 | 상품 외래키 |
| cNumber | INT |  | 갯수변동 |
| Stock | | | 상품 테이블 |
| sKey | VARCHAR(25) | PRIMARY KEY | 상품 식별 코드 |
| sName | VARCHAR(50) |  | 상품명 |
| sPrice | INT |  | 상품가격 |

**6.검증 기준**

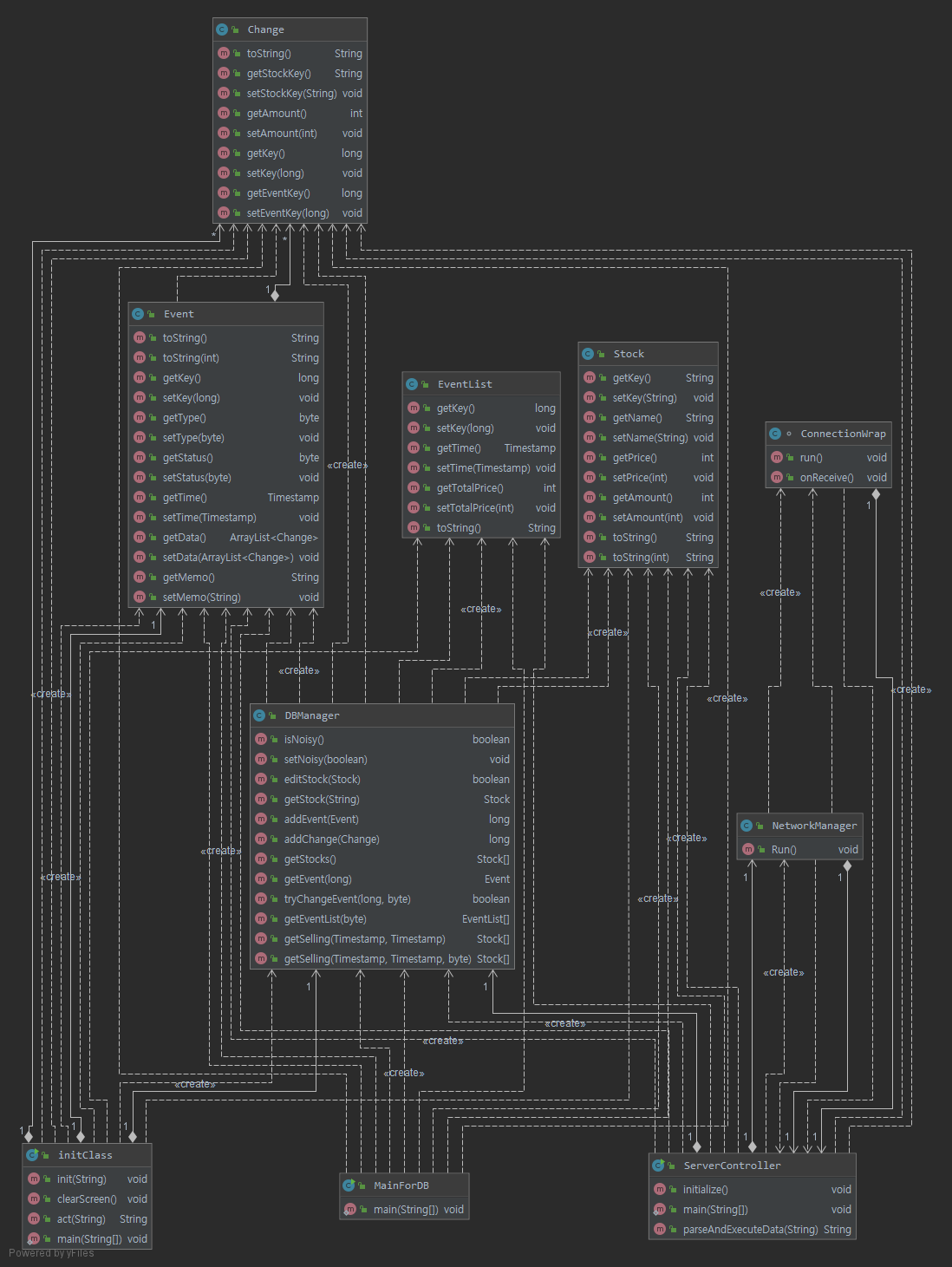
|  |  |  |  |
| --- | --- | --- | --- |
| **Program Checklist** | | | |
| **검증 기준** | **Yes** | **No** | **비 고** |
| 프로그램이 정상적으로 수행되는가? |  |  |  |
| 이미 결제된 품목들을 환불할 수 있는가? |  |  |  |
| 새로운 품목을 추가할 수 있는가? |  |  |  |
| 재고품 리스트를 출력 가능한가? |  |  |  |
| 계산을 완료하고 영수증을 출력 가능한가? |  |  |  |
| 기간을 설정하여 매출통계를 확인할 수 있는가? |  |  |  |
| 프로그램 종료 후 재시작했을 경우 이전 데이터들을 그대로 저장하고 있는가? |  |  |  |
| 바코드를 통해 물건을 정상적으로 인식할 수 있는가? |  |  |  |
| 재고품의 정보를 수정할 수 있는가? |  |  |  |
| 모바일 기기와 연동하여 사용 가능한가? |  |  |  |
| 지정되어 있는 IP/Password를 통해서만 로그인 가능한가? |  |  |  |
| 버튼을 통하여 화면간의 이동이 가능한가? |  |  |  |
| 서버와의 연동이 정상적으로 가능한가? |  |  |  |
| 데이터베이스를 통해 데이터들을 관리하는가? |  |  |  |
| 프로그램 기능상 오류가 있는 부분은 없는가? |  |  |  |

**7.Use case Diagram**

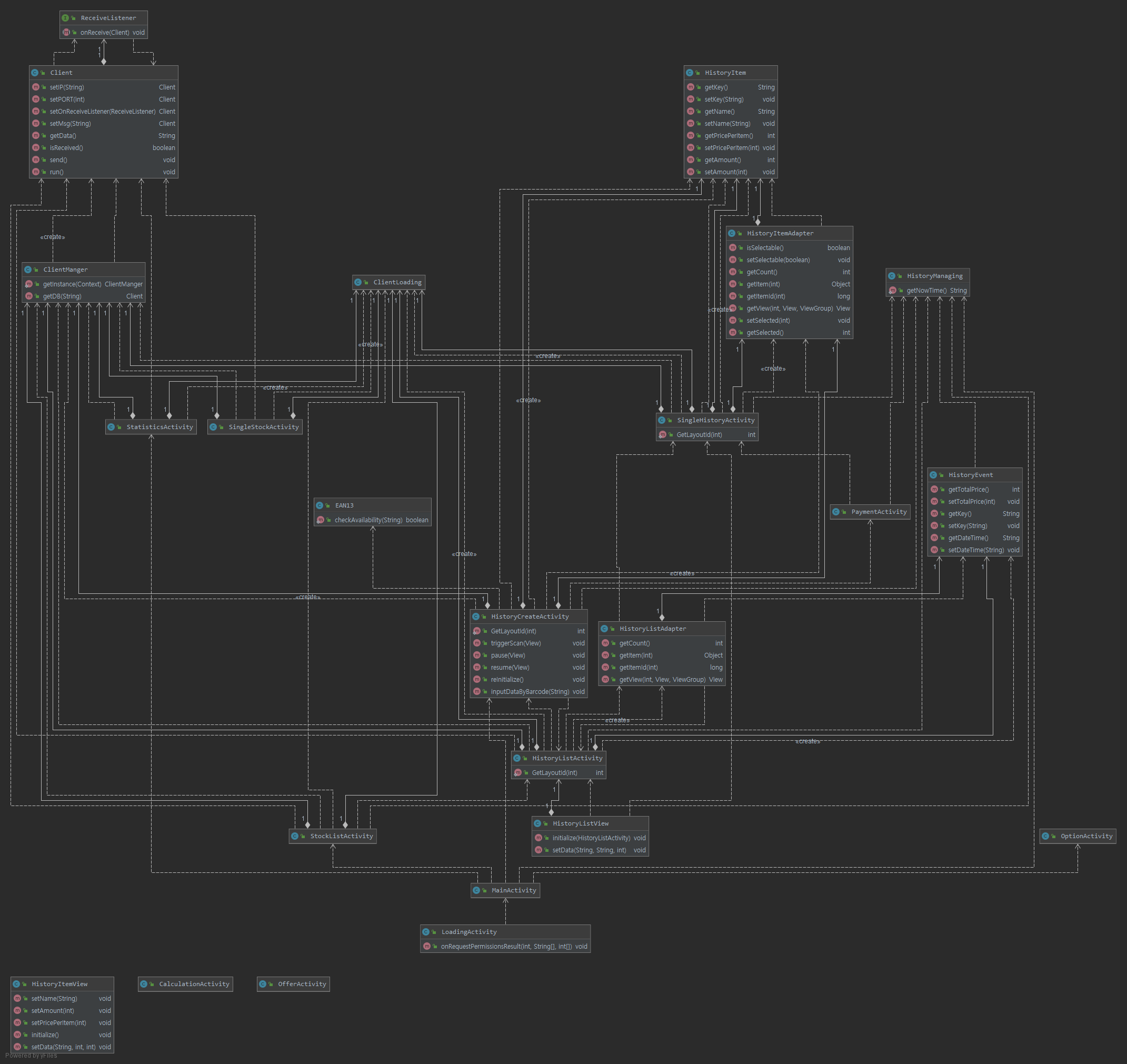


**8..Class Diagram**

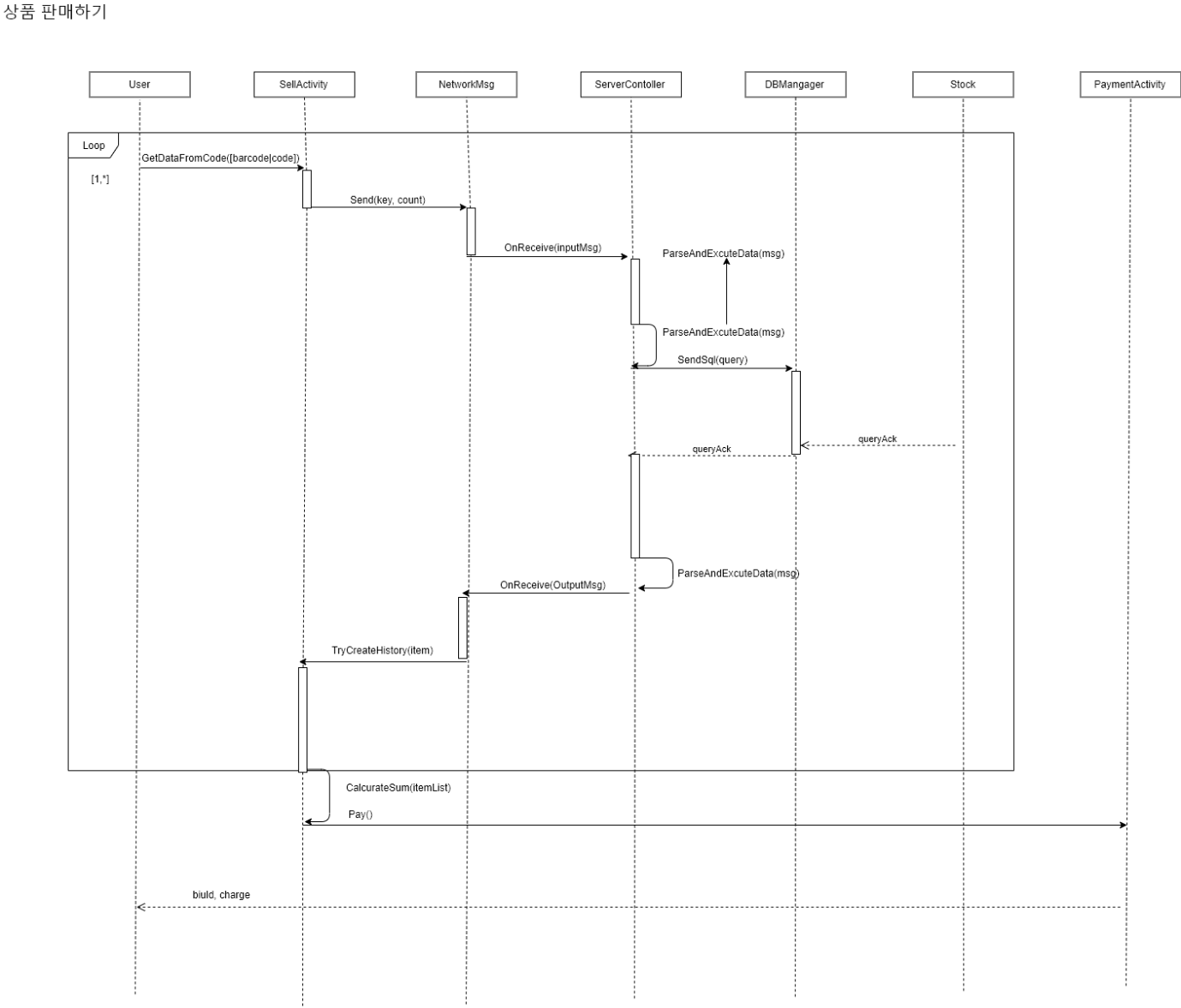
**서버**

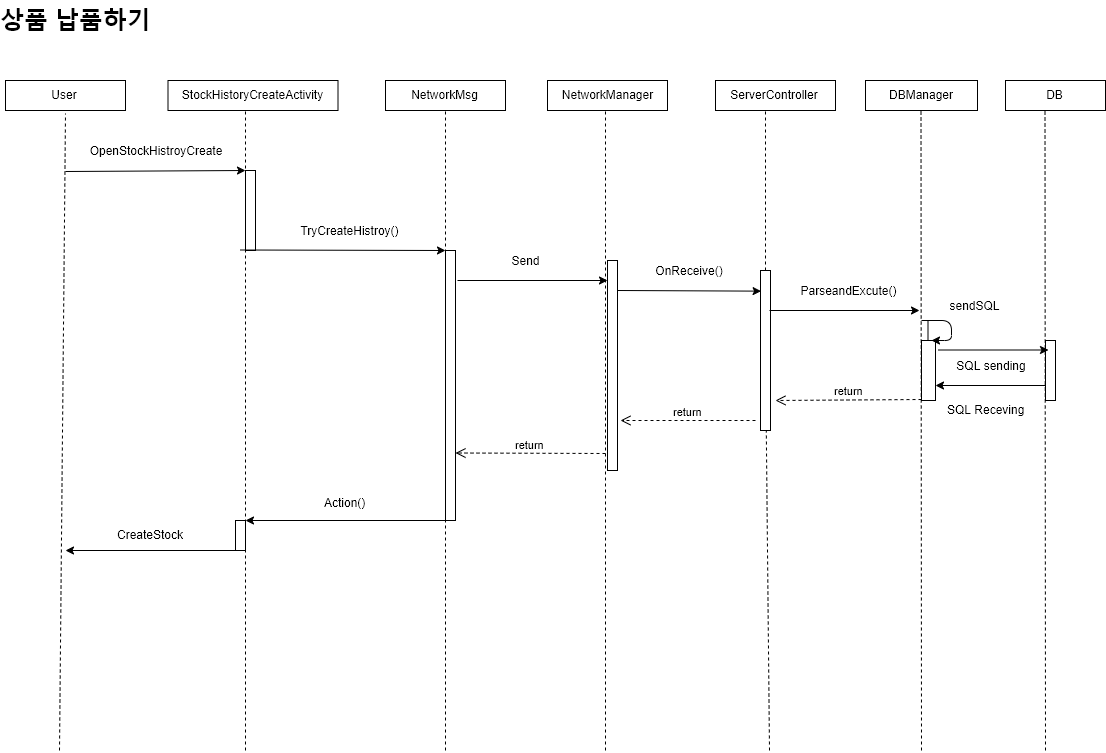
****

**클라이언트**

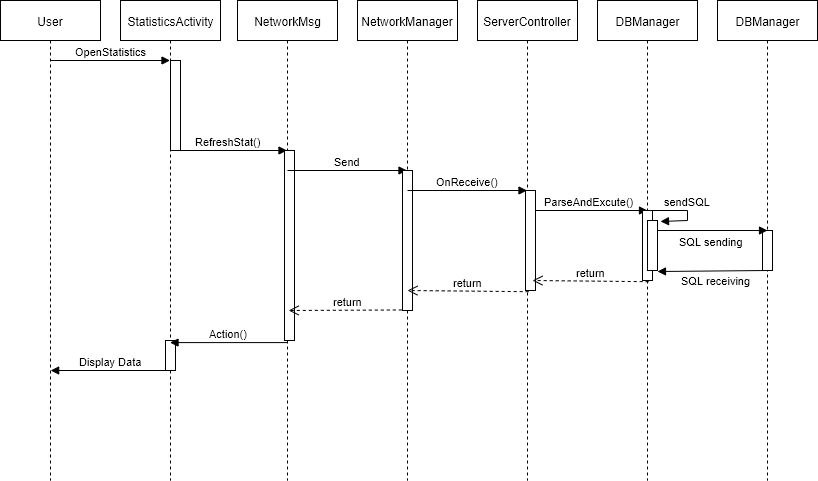
****

**9.Sequence Diagram**

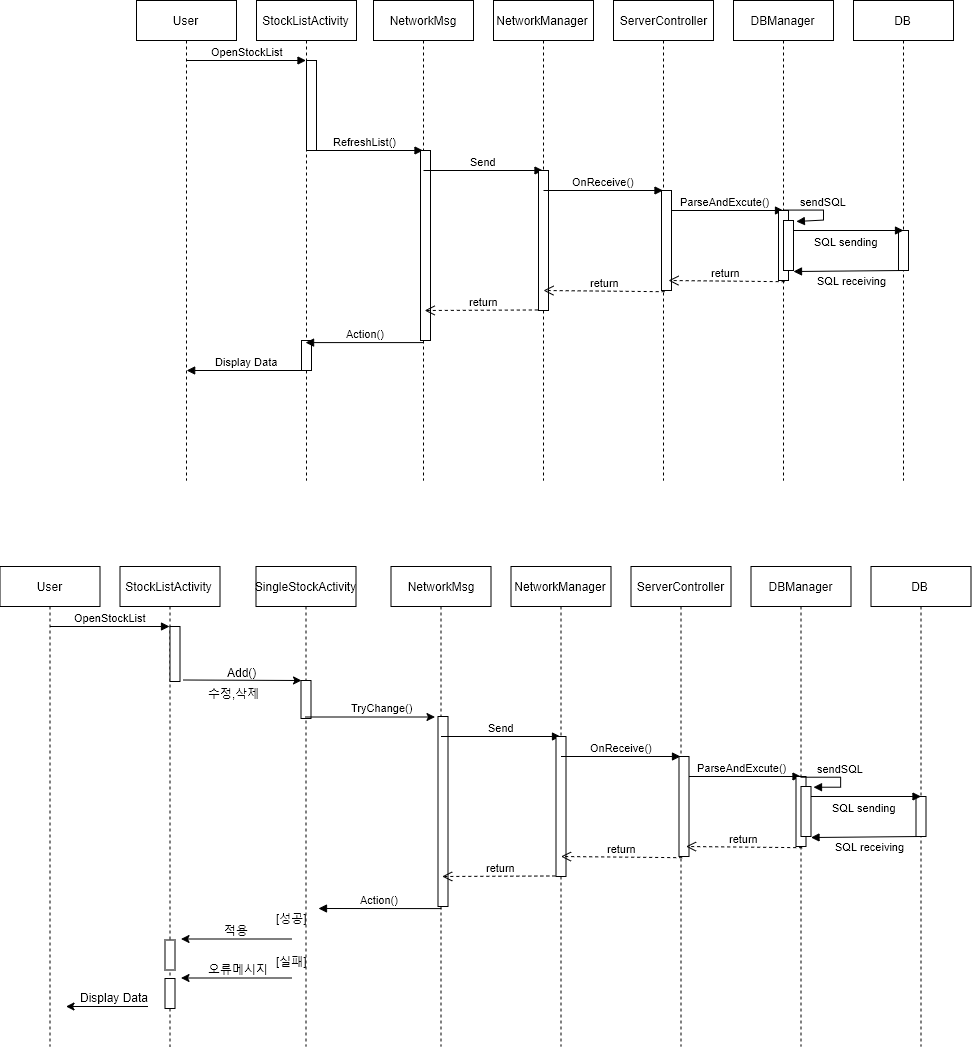


****

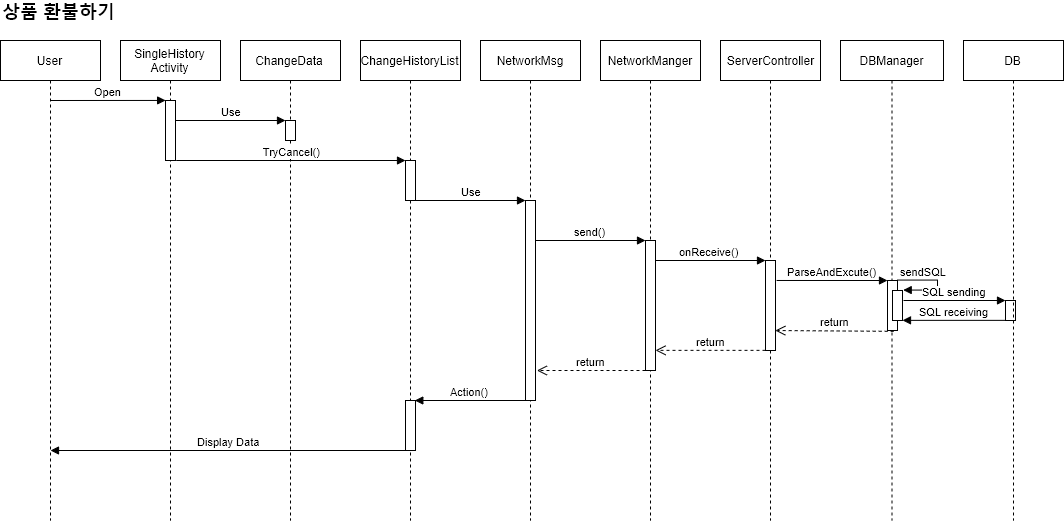
**매출 확인하기**

****

**재고 확인하기**

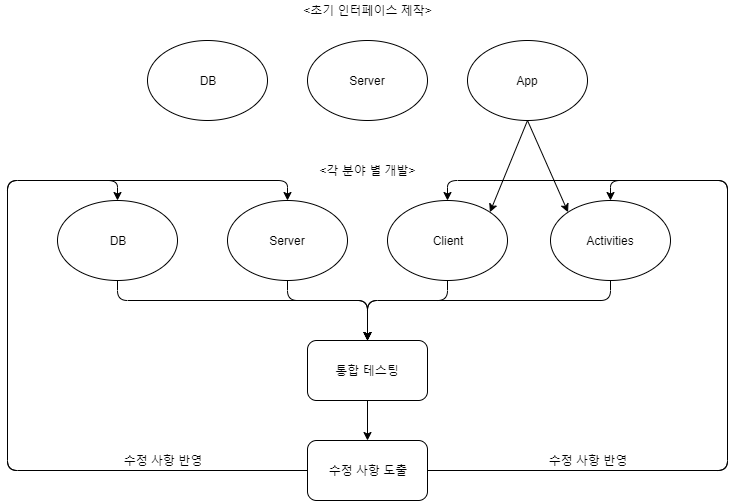
****

**상품 환불하기**

****

**10.테스트**

**<소프트웨어 테스팅 절차>**



**<테스팅 결과>**

|  |  |
| --- | --- |
| **1차 통합 테스팅** | |
| 도출 사항 | 서버와 앱 간의 메시지 수신 오류 수정  클라이언트에 수신된 데이터의 액티비티 파싱 오류 수정  액티비티 커스텀 뷰로 인한 앱 속도 저하로 리스트 어댑터로 변경 |
| **2차 통합 테스팅** | |
| 도출 사항 | 수신 대기로 인한 액티비티 정지로 클라이언트를 런어블 쓰레드와 리스너 구조로 변경  수신 오류 시 무한대기로 인한 클라이언트 연결 및 수신 타임아웃 설정 |
| **3차 통합 테스팅** | |
| 도출 사항 | DB 판매 개수 음수 표기로 인한 통계 부분과 판매 기록 액티비티 음수 출력 수정  서버에 연결 실패 시 아이피, 포트 재설정 오류로 인한 불필요 전역변수 제거 |
| **최종 통합 테스팅** | |
| 도출 사항 | DB, 서버, 클라이언트 띄어쓰기 구분자로 인한 오류로 프로토콜 구분자 변경  서버 콘솔 부분 예외처리 부족으로 인한 오류 수정 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Program Checklist** | | | |
| **검증 기준** | **Yes** | **No** | **비 고** |
| 프로그램이 정상적으로 수행되는가? |  |  |  |
| 이미 결제된 품목들을 환불할 수 있는가? |  |  |  |
| 새로운 품목을 추가할 수 있는가? |  |  |  |
| 한번에 여러 품목을 추가할 수 있는가? |  |  |  |
| 재고품 리스트를 출력 가능한가? |  |  |  |
| 재고품의 정보를 수정할 수 있는가? |  |  |  |
| 판매/납품 정보를 확인할 수 있는가? |  |  |  |
| 판매/납품 정보를 수정할 수 있는가? |  |  |  |
| 기간을 설정하여 매출통계를 확인할 수 있는가? |  |  |  |
| 바코드로 물건을 정상적으로 인식할 수 있는가? |  |  |  |
| 여러 기기와 다중 연결 및 사용 가능한가? |  |  |  |
| 모든 액티비티 버튼들의 기능이 잘 작동하는가? |  |  |  |
| 서버와의 연동이 정상적으로 가능한가? |  |  |  |
| 데이터베이스를 통해 데이터들을 관리하는가? |  |  |  |
| 프로그램 종료 후 재시작할 경우 이전 데이터들을 그대로 저장하고 있는가? |  |  | DataBase  Shared Preference |
| 프로그램 기능상 오류가 있는 부분은 없는가? |  |  |  |

**<최종 테스팅 체크리스트>**