

# 편의점 POS 시스템 개발자 메뉴얼

2022. 06. 10

2019305059 이현수

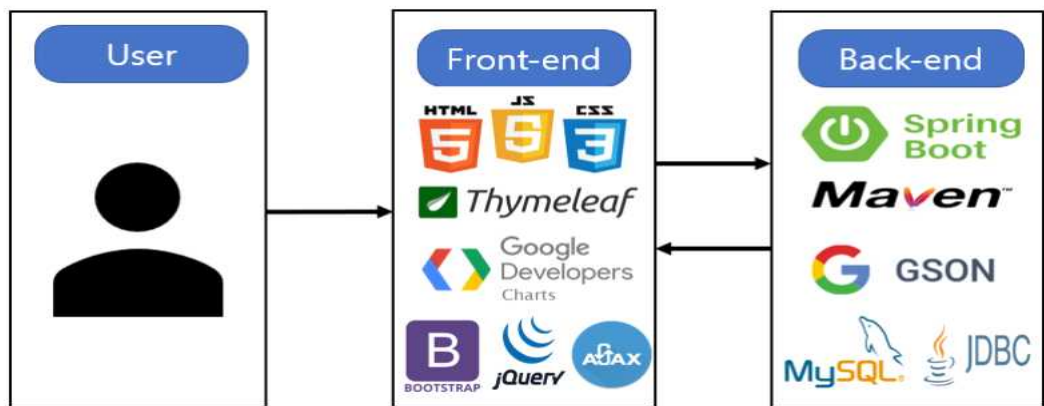
# 요 약 본

## 작품 정보

**프로젝트명** 편의점 POS(Point Of Sale) 시스템

**프로젝트 소개** 스프링부트 프레임워크를 이용한 편의점 POS 시스템 개발

### 작품 구성도



### 작품 기능

- 계정기능 : 사용자(직원, 매니저)의 ID/PASSWORD 관리
- 재고관리(입고) 기능 : 제품별 코드/가격 지정, 날짜별 제품입고 수량 등
- 판매기능 : 손님이 가져온 물건 종류와 수량입력하고 계산
- 통계기능 : 하루/일주일/한달 판매량/매출액 최다 판매 제품

### 요구 사항

- Spring Boot 사용
- Spring MVC를 이용하여 웹버전으로 개발
- 계정정보, 재고정보, 판매정보 등은 MySQL 사용(Spring JDBC 이용)
- Spring Context객체를 사용할 것. (xml 또는 자바설정 방식 선택)
- 주요 객체들간의 dependency를 spring 설정기능을 이용해 constructor/setter injection을 사용하여 연결
- Bean 객체 7개 이상 등록(setter, constructor injection 사용) - 17개 Bean 객체 사용
- 웹화면은 Thymeleaf 사용

# 본 문

## I. 작품 개요

### 1. 작품 소개

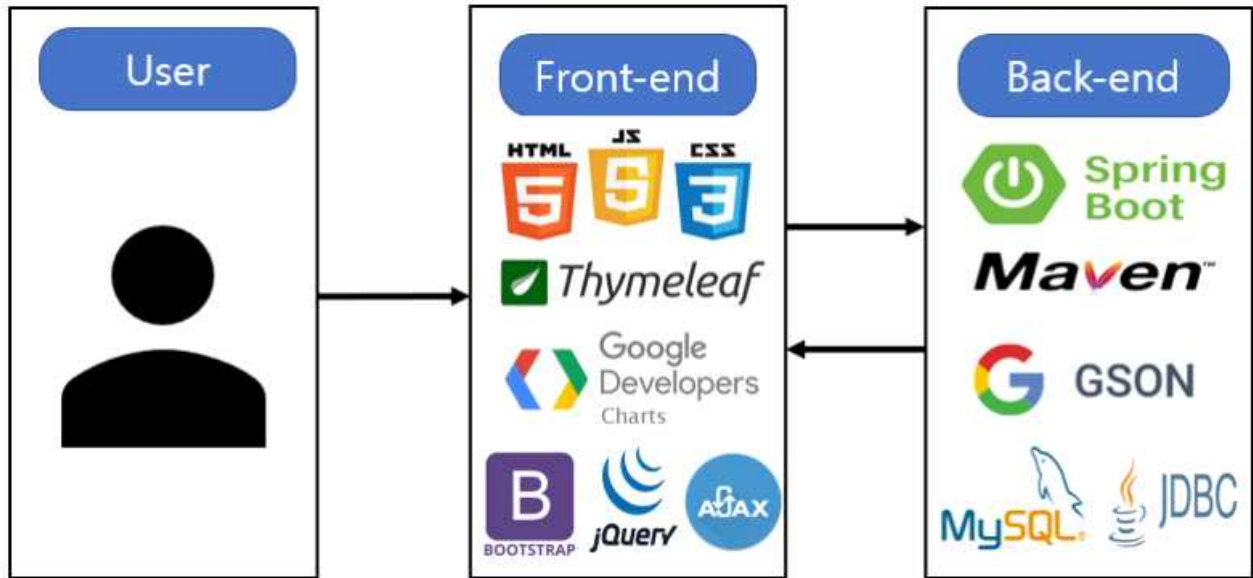
1) Spring MVC를 이용한 웹서비스 편의점 POS 시스템

- Spring Boot 사용
- 계정기능은 직원, 매니저의 아이디와 비밀번호, 이름을 관리함
- 계정을 만들고 로그인을 해야 POS 시스템의 기능을 사용할 수 있음
- 재고 관리 기능을 통해 상품 코드, 상품 이름, 상품 가격, 상품 수량을 관리할 수 있음
- 재고 관리 기능에서 날짜별 제품 입고 수량과 입고한 사용자 계정을 저장함
- 제품 판매를 통해 재고에 변동 발생 시에도 관련 기록이 저장됨
- 판매 기능에서 선택한 상품을 계산함
- 판매 기능을 통해 판매한 상품은 기록이 됨
- 저장된 판매 기록을 통해 하루/일주일/한달 판매량/매출액 최다 판매제품을 알려주는 통계서비스를 제공함

## II. 작품 내용

### 1. 작품 구성도

#### 1) 서비스 구성도



### 2. 작품 기능


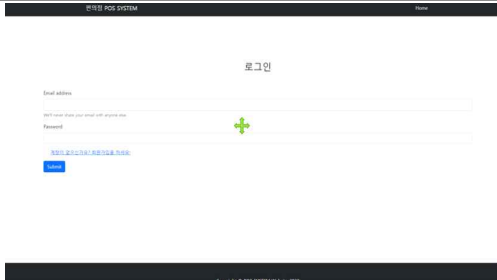
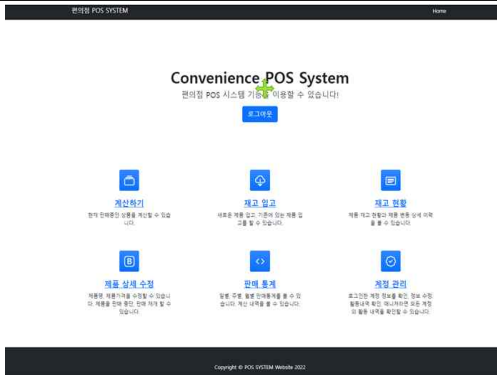

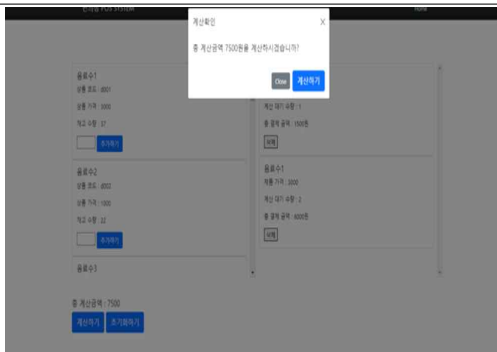
#### 1) 기능 구성도



## 2) 전체 기능 목록

구분	기능		설명
S/W	회원가입		회원가입 시 아이디, 비밀번호, 이름, 직책(매니저, 직원)을 입력받아 계정 정보를 저장함.
	로그인		회원가입을 통해 생성한 아이디, 비밀번호를 통해 로그인을 할 수 있음. 로그인을 해야 POS 시스템의 모든 기능을 이용할 수 있음.
	로그아웃		로그인 된 상태에서 로그아웃을 할 수 있음.
	계산하기	제품추가	판매중인 상품을 선택해 계산 대기 구역에 추가할 수 있음. 제품을 추가하면 총 계산금액이 업데이트됨.
		제품삭제	제품 추가 기능을 통해 추가한 제품을 계산 대기 구역에서 삭제할 수 있음.
		초기화	제품 추가 기능을 통해 추가한 모든 제품을 계산 대기 구역에서 삭제함.
		계산하기	제품 추가를 통해 추가한 모든 제품에 대해 계산을 할 수 있음. 계산하기 전에 계산확인 창이 뜸.
	재고입고	새로운 제품 추가	제품 코드, 제품명, 가격, 수량을 입력해 새로운 제품을 추가할 수 있음. 제품 코드는 유일해야 함.
		기존 제품 재고 입고	기존 제품 입고 수량을 입력해 제품을 입고 처리함.
	재고현황	재고 현황	데이터베이스에 등록된 모든 제품의 가격, 판매상태, 재고 수량을 볼 수 있음.
		수량 변동 내역 상세보기	제품 수량 변동 내역을 보여줌. BUY, SALE 두가지 상태로 구분되며, BUY는 제품 입고를 의미하고 SALE은 제품 판매를 의미함.
	제품 상세 수정	제품 이름, 가격 수정	제품의 제품명, 제품 가격을 수정할 수 있음. 제품 이름은 중복이 허용됨.
		제품 상태 수정	제품 상태는 판매중지, 판매중으로 나뉨. 판매중지인 제품은 계산할 수 없음.
		제품 수정 이력	제품 생성, 제품 상태 수정 내역을 볼 수 있음. CREATE, START, STOP 세가지 상태가 있음. CREATE는 처음 제품을 추가한 경우, START는 제품 상태를 판매중으로 변경한 경우, STOP은 제품 상태를 판매중지로 변경한 경우임.
	판매 통계	판매량 통계	사용자가 일별, 주별, 월별 날짜를 선택해 선택한 일, 주, 월별 판매량을 파이차트로 보여주고, 판매량 상위 5개 제품은 표로 표시됨.
		판매액 통계	사용자가 일별, 주별, 월별 날짜를 선택해 선택한 일, 주, 월별 판매액을 파이차트로 보여주고, 판매액 상위 5개 제품은 표로 표시됨.
		판매 내역	계산한 판매내역을 확인할 수 있음. 선택한 판매내역은 판매한 물품을 볼 수 있음.
	계정 관리	계정 정보 수정	로그인 중인 계정의 비밀번호, 이름, 직책을 변경할 수 있음.
		계정 활동 내역	로그인 중인 계정이 처리한 판매 계산 내역, 상품 변경 내역, 상품 상태 변경 내역을 볼 수 있음.
		타계정 활동 내역	로그인 중인 계정의 직책이 매니저인 경우 매니저를 포함한 다른 모든 계정의 상세정보, 활동 내역을 볼 수 있음.

### 3) S/W 주요 기능

기능		설명	작품실물사진
회원가입		회원가입 시 아이디, 비밀번호, 이름, 직책(매니저, 직원)을 입력받아 계정 정보를 저장함.	
로그인		회원가입을 통해 생성한 아이디, 비밀번호를 통해 로그인을 할 수 있음. 로그인을 해야 POS 시스템의 모든 기능을 이용할 수 있음.	
로그아웃		로그인 된 상태에서 로그아웃을 할 수 있음.	
계산하기	제품추가	판매중인 상품을 선택해 계산 대기 구역에 추가할 수 있음. 제품을 추가하면 총 계산금액이 업데이트됨.	
	제품삭제	제품 추가 기능을 통해 추가한 제품을 계산 대기 구역에서 삭제할 수 있음.	
	초기화	제품 추가 기능을 통해 추가한 모든 제품을 계산 대기 구역에서 삭제함.	
	계산하기	제품 추가를 통해 추가한 모든 제품에 대해 계산을 할 수 있음. 계산하기 전에 계산확인 창이 뜬다.	

재고 입고	새로운 제품 추가	제품 코드, 제품명, 가격, 수량을 입력해 새로운 제품을 추가할 수 있음. 제품 코드는 유일해야 함.	<div>새로운 제품 입고</div> <div>제품 정보 입력</div> <div>코드</div> <div>제품명</div> <div>가격</div> <div>수량</div> <div>제품 추가</div>																																																																																							
	기존 제품 재고 입고	기존 제품 입고 수량을 입력해 제품을 입고 처리함.	<div>새로운 제품 입고</div> <div>수량 입고</div> <div>코드</div> <div>제품명</div> <div>가격</div> <div>현재 재고 수량</div> <div>추가 입고 수량</div> <div>제품 추가</div>																																																																																							
재고 현황	재고 현황	데이터베이스에 등록된 모든 제품의 가격, 판매상태, 재고 수량을 볼 수 있음.	<div>제품 재고 현황</div> <table><tr><th>Code</th><th>제품이름</th><th>가격</th><th>재고수량</th><th>판매상태</th><th>상태변경</th></tr><tr><td>0001</td><td>종로구1</td><td>1000</td><td>17</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0002</td><td>종로구2</td><td>1000</td><td>22</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0003</td><td>종로구3</td><td>2000</td><td>19</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0004</td><td>종로구4</td><td>2000</td><td>20</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0005</td><td>종로구5</td><td>1500</td><td>30</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0007</td><td>성동북동1</td><td>2000</td><td>9</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0008</td><td>성동북동2</td><td>1000</td><td>17</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0009</td><td>성동북동3</td><td>10000</td><td>19</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0004</td><td>성동북동4</td><td>20000</td><td>10</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0001</td><td>강가1</td><td>1500</td><td>40</td><td>ON</td><td>판매중지/판매중</td></tr></table>	Code	제품이름	가격	재고수량	판매상태	상태변경	0001	종로구1	1000	17	ON	판매중지/판매중	0002	종로구2	1000	22	ON	판매중지/판매중	0003	종로구3	2000	19	ON	판매중지/판매중	0004	종로구4	2000	20	ON	판매중지/판매중	0005	종로구5	1500	30	ON	판매중지/판매중	0007	성동북동1	2000	9	ON	판매중지/판매중	0008	성동북동2	1000	17	ON	판매중지/판매중	0009	성동북동3	10000	19	ON	판매중지/판매중	0004	성동북동4	20000	10	ON	판매중지/판매중	0001	강가1	1500	40	ON	판매중지/판매중																					
	Code	제품이름	가격	재고수량	판매상태	상태변경																																																																																				
0001	종로구1	1000	17	ON	판매중지/판매중																																																																																					
0002	종로구2	1000	22	ON	판매중지/판매중																																																																																					
0003	종로구3	2000	19	ON	판매중지/판매중																																																																																					
0004	종로구4	2000	20	ON	판매중지/판매중																																																																																					
0005	종로구5	1500	30	ON	판매중지/판매중																																																																																					
0007	성동북동1	2000	9	ON	판매중지/판매중																																																																																					
0008	성동북동2	1000	17	ON	판매중지/판매중																																																																																					
0009	성동북동3	10000	19	ON	판매중지/판매중																																																																																					
0004	성동북동4	20000	10	ON	판매중지/판매중																																																																																					
0001	강가1	1500	40	ON	판매중지/판매중																																																																																					
수량 변동 내역 상세보기	제품 수량 변동 내역을 보여줌. BUY, SALE 두가지 상태로 구분되며, BUY는 제품 입고를 의미하고 SALE은 제품 판매를 의미함.	<div>수량변동내역</div> <table><tr><th>이동일</th><th>제품이름</th><th>가격</th><th>수량</th><th>날짜 시간</th><th>방법</th></tr><tr><td>1</td><td>종로구1</td><td>1000</td><td>2</td><td>2022-05-17T15:09</td><td>SALE</td></tr><tr><td>1</td><td>종로구1</td><td>1000</td><td>10</td><td>2022-05-18T11:04:13</td><td>BUY</td></tr><tr><td>2</td><td>종로구1</td><td>1000</td><td>1</td><td>2022-05-18T05:11:46</td><td>SALE</td></tr><tr><td>1</td><td>종로구1</td><td>1000</td><td>4</td><td>2022-05-18T05:07:38</td><td>BUY</td></tr><tr><td>1</td><td>종로구1</td><td>1000</td><td>2</td><td>2022-05-18T13:08:02</td><td>SALE</td></tr><tr><td>1</td><td>종로구1</td><td>1000</td><td>1</td><td>2022-05-17T15:05</td><td>SALE</td></tr><tr><td>1</td><td>종로구1</td><td>1000</td><td>1</td><td>2022-05-18T08:06:26</td><td>SALE</td></tr><tr><td>1</td><td>종로구1</td><td>1000</td><td>30</td><td>2022-05-18T04:49</td><td>BUY</td></tr></table>	이동일	제품이름	가격	수량	날짜 시간	방법	1	종로구1	1000	2	2022-05-17T15:09	SALE	1	종로구1	1000	10	2022-05-18T11:04:13	BUY	2	종로구1	1000	1	2022-05-18T05:11:46	SALE	1	종로구1	1000	4	2022-05-18T05:07:38	BUY	1	종로구1	1000	2	2022-05-18T13:08:02	SALE	1	종로구1	1000	1	2022-05-17T15:05	SALE	1	종로구1	1000	1	2022-05-18T08:06:26	SALE	1	종로구1	1000	30	2022-05-18T04:49	BUY																																		
이동일	제품이름	가격	수량	날짜 시간	방법																																																																																					
1	종로구1	1000	2	2022-05-17T15:09	SALE																																																																																					
1	종로구1	1000	10	2022-05-18T11:04:13	BUY																																																																																					
2	종로구1	1000	1	2022-05-18T05:11:46	SALE																																																																																					
1	종로구1	1000	4	2022-05-18T05:07:38	BUY																																																																																					
1	종로구1	1000	2	2022-05-18T13:08:02	SALE																																																																																					
1	종로구1	1000	1	2022-05-17T15:05	SALE																																																																																					
1	종로구1	1000	1	2022-05-18T08:06:26	SALE																																																																																					
1	종로구1	1000	30	2022-05-18T04:49	BUY																																																																																					
제품 상세 수정	제품 이름, 가격 수정	제품의 제품명, 제품 가격을 수정할 수 있음. 제품 이름은 중복이 허용됨.	<div>제품 수정</div> <div>제품 정보 입력</div> <div>코드</div> <div>제품명</div> <div>가격</div> <div>제품 수정</div>																																																																																							
	제품 상태 수정	제품 상태는 판매중지, 판매중으로 나뉨. 판매중지인 제품은 계산할 수 없음.	<table><tr><th>Code</th><th>제품이름</th><th>가격</th><th>수량</th><th>수정가격</th><th>수정수량</th><th>현재상태</th><th>현재상태수정</th></tr><tr><td>0001</td><td>종로구1</td><td>1000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0002</td><td>종로구2</td><td>1000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0003</td><td>종로구3</td><td>2000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0004</td><td>종로구4</td><td>2000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0005</td><td>종로구5</td><td>1500</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0007</td><td>성동북동1</td><td>2000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0008</td><td>성동북동2</td><td>1000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0009</td><td>성동북동3</td><td>10000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0004</td><td>성동북동4</td><td>20000</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr><tr><td>0001</td><td>강가1</td><td>1500</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>판매중지/판매중</td></tr></table>	Code	제품이름	가격	수량	수정가격	수정수량	현재상태	현재상태수정	0001	종로구1	1000	ON	ON	ON	ON	판매중지/판매중	0002	종로구2	1000	ON	ON	ON	ON	판매중지/판매중	0003	종로구3	2000	ON	ON	ON	ON	판매중지/판매중	0004	종로구4	2000	ON	ON	ON	ON	판매중지/판매중	0005	종로구5	1500	ON	ON	ON	ON	판매중지/판매중	0007	성동북동1	2000	ON	ON	ON	ON	판매중지/판매중	0008	성동북동2	1000	ON	ON	ON	ON	판매중지/판매중	0009	성동북동3	10000	ON	ON	ON	ON	판매중지/판매중	0004	성동북동4	20000	ON	ON	ON	ON	판매중지/판매중	0001	강가1	1500	ON	ON	ON	ON
Code	제품이름	가격	수량	수정가격	수정수량	현재상태	현재상태수정																																																																																			
0001	종로구1	1000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0002	종로구2	1000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0003	종로구3	2000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0004	종로구4	2000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0005	종로구5	1500	ON	ON	ON	ON	판매중지/판매중																																																																																			
0007	성동북동1	2000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0008	성동북동2	1000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0009	성동북동3	10000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0004	성동북동4	20000	ON	ON	ON	ON	판매중지/판매중																																																																																			
0001	강가1	1500	ON	ON	ON	ON	판매중지/판매중																																																																																			
제품 수정 이력	제품 생성, 제품 상태 수정 내역을 볼 수 있음. CREATE, START, STOP 세가지 상태가 있음. CREATE는 처음 제품을 추가한 경우, START는 제품 상태를 판매중으로 변경한 경우, STOP은 제품 상태를 판매중지로 변경한 경우임.	<div>이력내역</div> <table><tr><th>이동일</th><th>제품이름</th><th>새이름</th><th>이동가격</th><th>새가격</th><th>상태</th><th>날짜</th></tr><tr><td>1</td><td>종로구1</td><td>종로구1</td><td>1000</td><td>1500</td><td>START</td><td>2022-05-18T09:44:19</td></tr><tr><td>1</td><td>종로구1</td><td>종로구1</td><td>1000</td><td>1500</td><td>STOP</td><td>2022-05-18T11:41:18</td></tr><tr><td>1</td><td>종로구1</td><td>종로구1</td><td>1000</td><td>1500</td><td>START</td><td>2022-05-18T11:41:18</td></tr><tr><td>1</td><td>종로구1</td><td>종로구1</td><td>1000</td><td>1500</td><td>UPGRADE</td><td>2022-05-18T12:16:08</td></tr><tr><td>1</td><td>종로구1</td><td>종로구1</td><td>1000</td><td>1500</td><td>UPGRADE</td><td>2022-05-18T12:16:14</td></tr><tr><td>1</td><td>종로구1</td><td>종로구11</td><td>1000</td><td>1500</td><td>UPGRADE</td><td>2022-05-18T12:18:01</td></tr><tr><td>1</td><td>종로구11</td><td>종로구11</td><td>1000</td><td>3000</td><td>UPGRADE</td><td>2022-05-18T12:18:07</td></tr><tr><td>1</td><td>종로구11</td><td>종로구1</td><td>3000</td><td>3000</td><td>UPGRADE</td><td>2022-05-18T12:18:06</td></tr><tr><td>2</td><td>종로구1</td><td>종로구1</td><td>3000</td><td>3000</td><td>STOP</td><td>2022-05-18T20:05:13</td></tr><tr><td>2</td><td>종로구1</td><td>종로구1</td><td>3000</td><td>3000</td><td>START</td><td>2022-05-18T20:05:15</td></tr></table>	이동일	제품이름	새이름	이동가격	새가격	상태	날짜	1	종로구1	종로구1	1000	1500	START	2022-05-18T09:44:19	1	종로구1	종로구1	1000	1500	STOP	2022-05-18T11:41:18	1	종로구1	종로구1	1000	1500	START	2022-05-18T11:41:18	1	종로구1	종로구1	1000	1500	UPGRADE	2022-05-18T12:16:08	1	종로구1	종로구1	1000	1500	UPGRADE	2022-05-18T12:16:14	1	종로구1	종로구11	1000	1500	UPGRADE	2022-05-18T12:18:01	1	종로구11	종로구11	1000	3000	UPGRADE	2022-05-18T12:18:07	1	종로구11	종로구1	3000	3000	UPGRADE	2022-05-18T12:18:06	2	종로구1	종로구1	3000	3000	STOP	2022-05-18T20:05:13	2	종로구1	종로구1	3000	3000	START	2022-05-18T20:05:15											
이동일	제품이름	새이름	이동가격	새가격	상태	날짜																																																																																				
1	종로구1	종로구1	1000	1500	START	2022-05-18T09:44:19																																																																																				
1	종로구1	종로구1	1000	1500	STOP	2022-05-18T11:41:18																																																																																				
1	종로구1	종로구1	1000	1500	START	2022-05-18T11:41:18																																																																																				
1	종로구1	종로구1	1000	1500	UPGRADE	2022-05-18T12:16:08																																																																																				
1	종로구1	종로구1	1000	1500	UPGRADE	2022-05-18T12:16:14																																																																																				
1	종로구1	종로구11	1000	1500	UPGRADE	2022-05-18T12:18:01																																																																																				
1	종로구11	종로구11	1000	3000	UPGRADE	2022-05-18T12:18:07																																																																																				
1	종로구11	종로구1	3000	3000	UPGRADE	2022-05-18T12:18:06																																																																																				
2	종로구1	종로구1	3000	3000	STOP	2022-05-18T20:05:13																																																																																				
2	종로구1	종로구1	3000	3000	START	2022-05-18T20:05:15																																																																																				

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



### III. 작품 개발 정보

#### 1. 작품 개발 환경

구분		상세내용
version : v1 / 개발기간 : 2022.05.09~2022.06.06		
S/W 개발환경	OS	Windows 11
	개발환경(IDE)	IntelliJ IDEA 2021.2.2
	개발도구	Spring Boot 2.6.7
	빌드관리도구	Maven
	개발언어	JAVA 11
	형상관리도구	Github – <a href="https://github.com/ehs2803/Convenience-POS-System">https://github.com/ehs2803/Convenience-POS-System</a>

#### 2. 의존성

groupId	artifactId	version
org.springframework.boot	spring-boot-starter-web	2.6.7
org.springframework.boot	spring-boot-starter-test	2.6.7
org.springframework.boot	spring-boot-starter-thymeleaf	2.6.7
org.springframework.boot	spring-boot-starter-data-jdbc	2.3.4
org.springframework.boot	spring-boot-starter-jdbc	2.6.7
org.springframework.boot	spring-boot-starter-validation	2.6.7
mysql	mysql-connector-java	8.0.28
org.apache.tomcat	tomcat-jdbc	9.0.62
nz.net.ultraq.thymeleaf	thymeleaf-layout-dialect	3.0.0
com.google.code.gson	gson	2.8.7
org.hibernate	hibernate-validator	5.2.4.Final

#### 3. View

##### 1) 부트스트랩 무료 템플릿

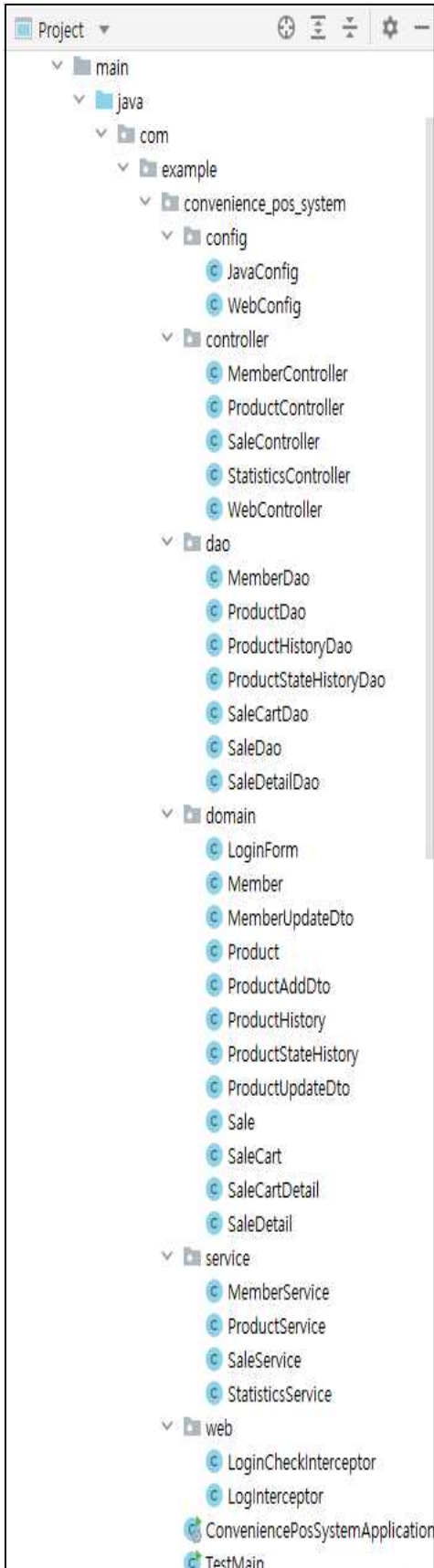
<https://startbootstrap.com/template/heroic-features>

##### 2) CSS, JS 라이브러리

구분	이름	버전	사용방식
CSS	Bootstrap icons	1.4.1	CDN
JS	jQuery	3.5.0	CDN
JS	gstatic - charts(google chart)	-	CDN

## 4. 프로그램 구조

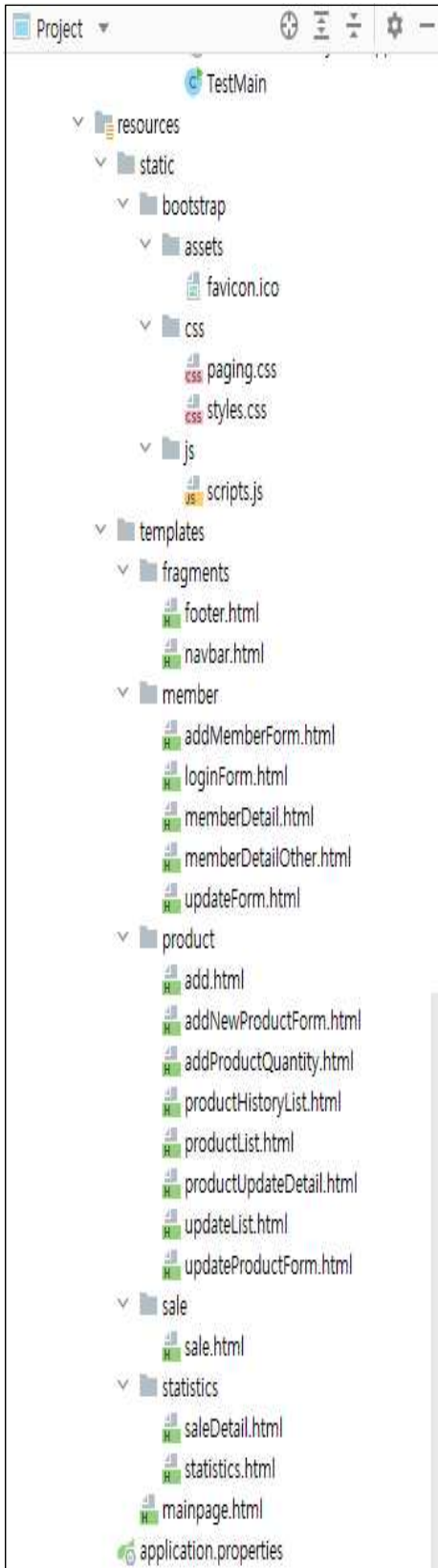
### 1) java 소스



#### 패키지 구조

- convenience\_pos\_system
  - config **[config 코드]**
    - JavaConfig.java (Bean 객체 생성)
    - WebConfig.java (인터셉터 등록)
  - controller **[controller 코드]**
    - MemberController.java (멤버서비스controller)
    - ProductController.java (제품서비스controller)
    - SaleController.java (판매서비스controller)
    - StatisticsController.java (통계서비스controller)
    - WebController.java (메인페이지 서비스controller)
  - dao **[DAO 코드]**
    - MemberDao.java (member\_tb 접근)
    - ProductDao.java (product\_tb 접근)
    - ProductHistoryDao.java (product\_history\_tb 접근)
    - ProductStateHistoryDao.java (product\_state\_history\_tb 접근)
    - SaleCartDao.java (sale\_cart\_tb 접근)
    - SaleDao.java (sale\_tb 접근)
    - SaleDetailDao.java (sale\_detail\_tb 접근)
  - domain **[DB테이블 Dto 코드]**
    - LoginForm.java (로그인 form DTO)
    - Member.java (member\_tb DTO)
    - MemberUpdateDto.java (멤버수정 Form DTO)
    - Product.java (product\_tb DTO)
    - ProductAddDto.java (제품수량 추가 Form DTO)
    - ProductHistory.java (product\_history\_tb DTO)
    - ProductStateHistory.java (product\_state\_history\_tb DTO)
    - ProductUpdateDto.java (제품수정 Form DTO)
    - Sale.java (sale\_tb DTO)
    - SaleCart.java (sale\_cart\_tb DTO)
    - SaleCartDetail.java (계산페이지 제품 DTO)
    - SaleDetail.java (sale\_detail\_tb DTO)
  - service **[Service 코드]**
    - MemberService.java (멤버 서비스)
    - ProductService.java (제품 서비스)
    - SaleService.java (계산, 판매 서비스)
    - StatisticsService.java (통계 서비스)
  - web **[인터럽트 설정 코드]**
    - LoginCheckInterceptor.java (로그인 확인 인터셉터)
    - LogInterceptor.java (로그 인터셉터)

## 2) view 소스



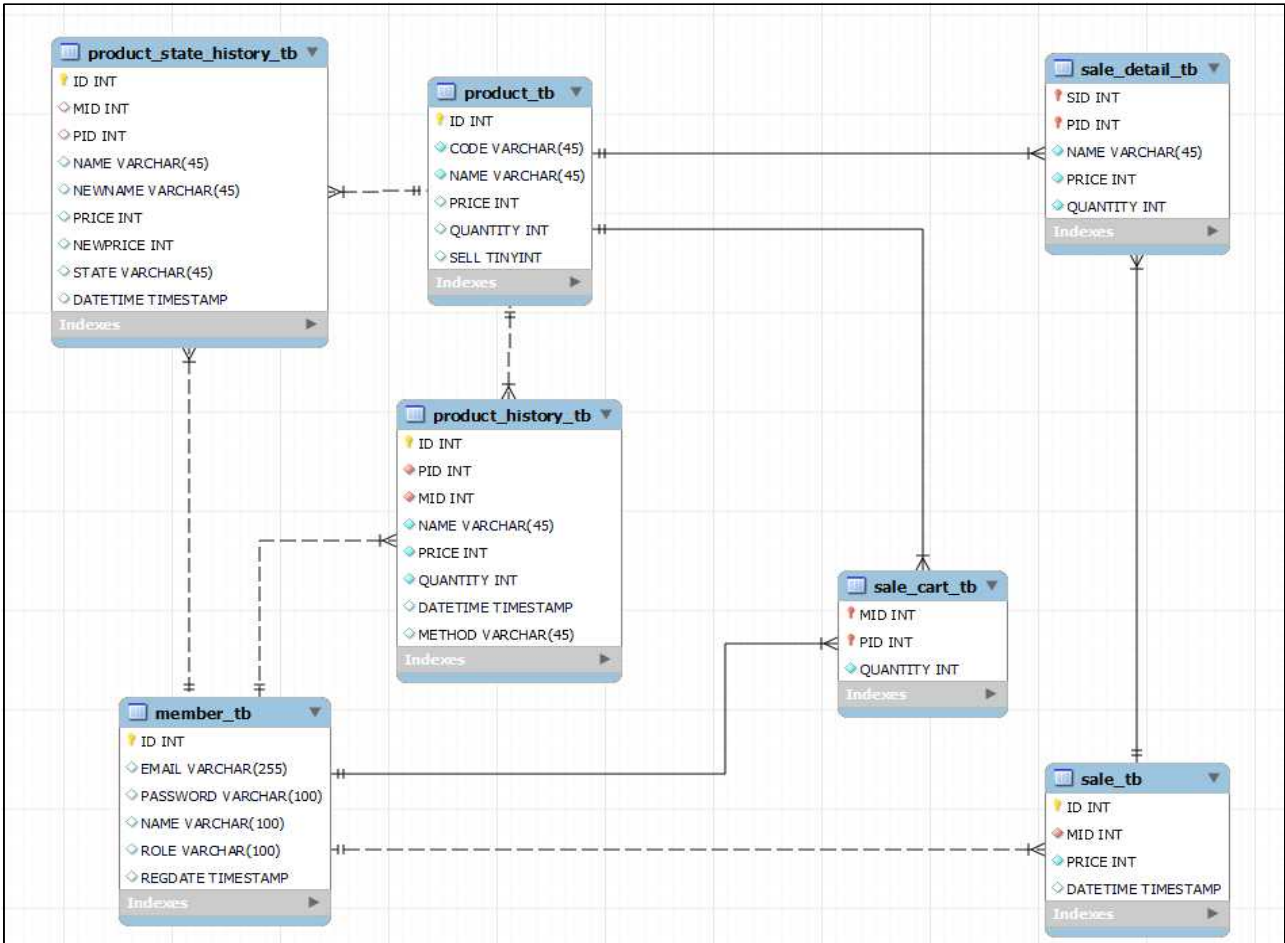
### 패키지 구조

#### resources

- static
  - bootstrap [부트스트랩 이용]
    - assets
      - favicon.ico
    - css
      - paging.css (테이블 페이지징을 위한 css)
      - styles.css (부트스트랩 무료템플릿 css)
    - js
      - scripts.js (부트스트랩 무료템플릿 js)
  - templates
    - fragments
      - footer.html (footer)
      - navbar.html (navbar)
    - member
      - addMemberForm.html (회원가입 Form)
      - loginForm.html (login Form)
      - memberDetail.html (계정관리페이지)
      - memberDetailOther.html (타계정활동확인)
      - updateForm.html (계정 정보 수정 Form)
    - product
      - add.html (제고 입고 페이지)
      - addNewProductForm.html (제품추가Form)
      - addProductQuantity.html (제품수량추가Form)
      - productHistoryList.html (제품변동내역)
      - productList.html (제고 현황 페이지)
      - productUpdateDetail.html (제품 변동 상세)
      - updateList.html (제품 수정 페이지)
      - updateProductForm.html (제품수정 Form)
    - sale
      - sale.html (계산하기 페이지)
    - statistics
      - saleDetail.html (판매 내역 상세보기)
      - statistics.html (통계 페이지)
    - mainpage.html (메인 화면)

## 5. 데이터베이스 구조

### 1) ER Diagram



테이블은 총 7개입니다.

**member\_tb** : 회원가입 시 회원의 정보를 저장하는 테이블입니다. 회원의 이메일, 비밀번호, 회원 이름, 회원 역할, 회원 등록 날짜가 저장됩니다.

**product\_tb** : 상품 정보가 저장되는 테이블입니다. 상품코드, 상품명, 상품가격, 상품 재고 수량, 현재 판매여부 정보가 저장됩니다.

**product\_history\_db** : 상품 수량 변동 내역 기록이 저장되는 테이블입니다. member\_tb와 product\_tb를 참조하고, 상품명, 상품가격, 상품 변동 수량, 날짜, 변동 방법이 저장됩니다.

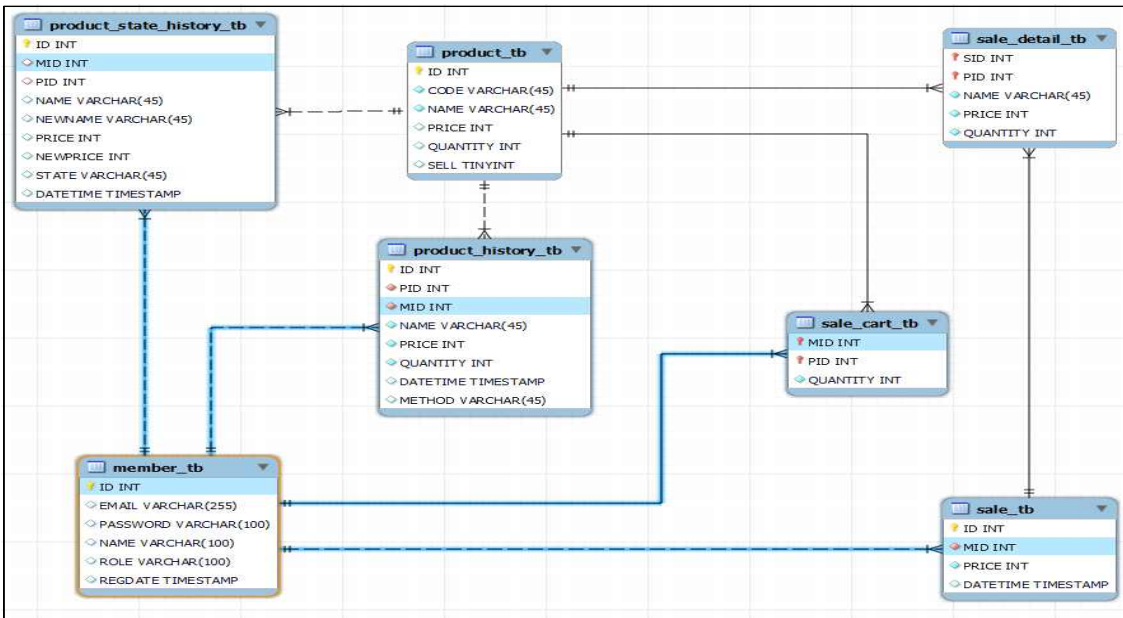
**product\_state\_hlstory\_db** : 상품 상태 변동 내역 기록이 저장되는 테이블입니다. member\_tb와 product\_tb를 참조하고, 기존 상품명, 새로운 상품명, 기존 상품 가격, 새로운 상품 가격, 상태 변동 구분, 날짜가 저장됩니다.

**sale\_tb** : 판매 내역이 저장되는 테이블입니다. member\_tb를 참조하고, 총 판매가격, 날짜가 저장됩니다.

**sale\_detail\_tb** : 판매 상세 내역이 저장되는 테이블입니다. sale\_tb, product\_tb 테이블을 참조하고, 상품명, 상품 가격, 판매 수량이 저장됩니다.

**sale\_cart\_tb** : 계산 대기 상품을 저장하는 테이블입니다. member\_tb, product\_tb를 참조하고, 계산 할 대기 상품 수량을 의미합니다.

## 2) 테이블 관계



### member\_tb : 회원 정보 저장 테이블

회원가입 시 회원의 정보를 저장하는 테이블입니다. 회원의 이메일, 비밀번호, 회원 이름, 회원 역할, 회원 등록 날짜가 저장됩니다.

### member\_tb - sale\_db 관계

sale\_tb는 member\_tb의 primary key를 외래키로 참조합니다. sale\_tb는 계산 내역 기록을 저장하는 테이블로 어느 직원이 계산했는지를 알 수 있습니다. 한명의 직원의 복수의 계산을 할 수 있기 때문에 1:N 관계입니다.

### member\_tb - sale\_cart\_db 관계

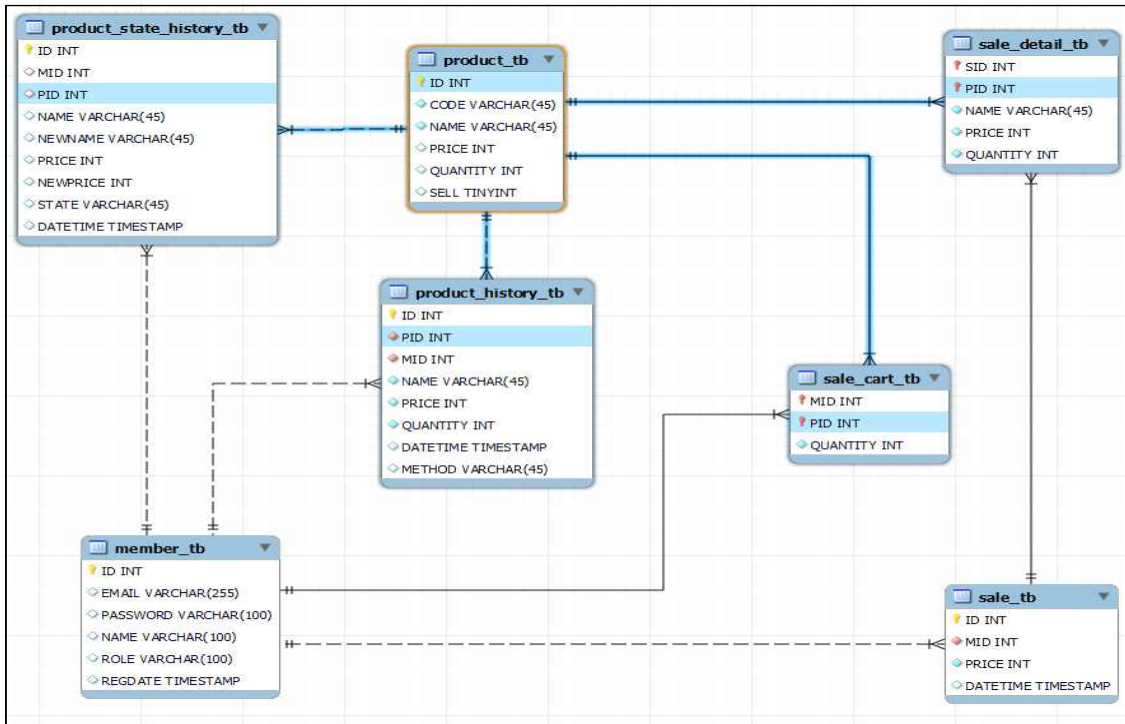
sale\_cart\_tb는 member\_tb의 primary key를 외래키로 참조합니다. sale\_cart\_tb 테이블은 계산 대기 상품을 임시로 저장하는 테이블로 어느 계정의 직원이 계산 대기 구역에 상품을 추가했는지 알 수 있습니다. 이렇게 함으로써 서로 다른 계정에서 로그인 후 계산하기 페이지에 들어가면 로그인 중인 계정이 추가한 상품만 계산 대기 구역에 표시됩니다. 한 직원이 복수의 상품을 계산 대기 구역에 저장할 수 있기 때문에 1:N 관계입니다.

### member\_tb - product\_history\_tb 관계

product\_history\_tb는 member\_tb의 primary key를 외래키로 참조합니다. product\_history\_tb는 상품 수량 변동 내역 기록을 저장하는 테이블로, 제품을 입고하거나 제품을 판매할 경우 제품 수량이 늘어나거나 줄어들게 되는데 이것을 어느 직원이 처리 했는지를 저장합니다. 한 직원이 복수의 상품, 혹은 여러 번의 입고, 계산을 할 수 있으므로 1:N 관계입니다.

### member\_tb - product\_state\_history\_tb 관계

product\_state\_history\_tb는 member\_tb의 primary key를 외래키로 참조합니다. 이 테이블은 상품 상태 변동을 저장하는 테이블로 상품의 이름, 가격, 판매상태를 수정한 경우 어느 직원이 이 작업을 처리했는지 확인할 수 있습니다. 한 직원이 특정 제품을 여러번 혹은 여러개의 제품에 대해 상태 수정을 할 수 있기 때문에 1:N 관계입니다.



## product\_tb : 상품 정보 저장 테이블

### product\_tb - product\_state\_history\_tb 관계

product\_state\_history\_tb는 product\_tb의 primary key를 참조합니다. 이 테이블은 상품 상태 변동을 저장하는 테이블로 상품의 이름, 가격, 판매상태를 수정한 경우 어떤 상품이 수정 대상인지를 확인할 수 있습니다. 한 상품에 대해 여러번의 상태 수정이 발생할 수 있기 때문에 1:N 관계입니다.

### product\_tb - product\_history\_tb 관계

product\_history\_tb는 product\_tb의 primary key를 참조합니다. product\_history\_tb는 상품 수량 변동 내역 기록을 저장하는 테이블로, 제품을 입고하거나 제품을 판매할 경우 제품 수량이 늘어나거나 줄어들게 되는데 어떤 상품의 수량이 변동됐는지를 확인할 수 있습니다. 한 상품에 대해 복수의 수량 변동이 발생할 수 있기 때문에 1:N 관계입니다.

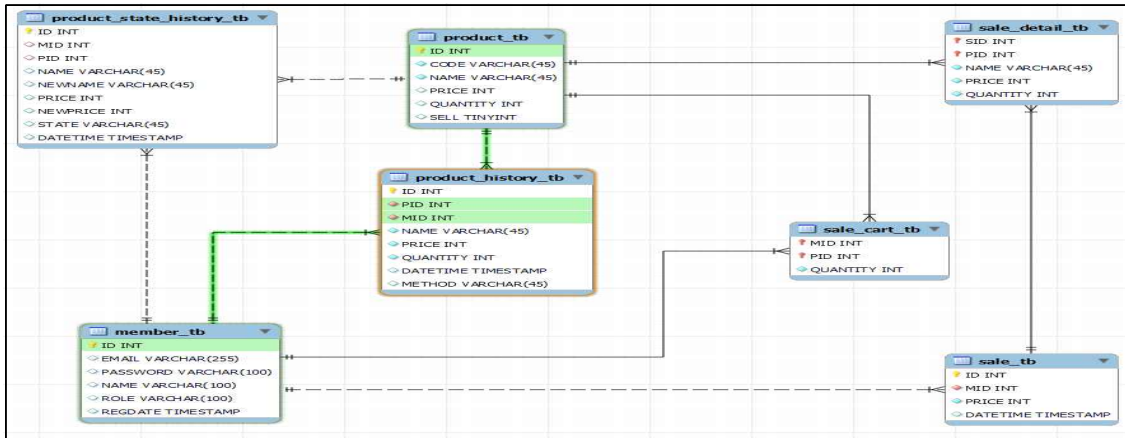
### product\_tb - sale\_cart\_tb 관계

sale\_cart\_tb는 product\_tb의 primary key를 참조합니다. sale\_cart\_tb 테이블은 계산 대기 상품을 임시로 저장하는 테이블로 어떤 상품이 계산 대기 구역으로 이동했는지 확인할 수 있습니다. 두명 이상의 직원이 동시에 로그인 해 동일한 상품을 계산 대기 구역에 추가할 수 있으므로 1:N 관계입니다.

### product\_tb - sale\_detail\_tb 관계

sale\_detail\_tb는 product\_tb의 primary key를 참조합니다. sale\_detail\_tb는 특정 판매건에 대해 계산한 상품명, 상품가격, 상품 수량을 저장합니다. 이때 어떤 상품을 계산했는지 확인할 수 있습니다. 한 상품에 대해 복수의 계산이 가능하기 때문에 1:N 관계입니다.





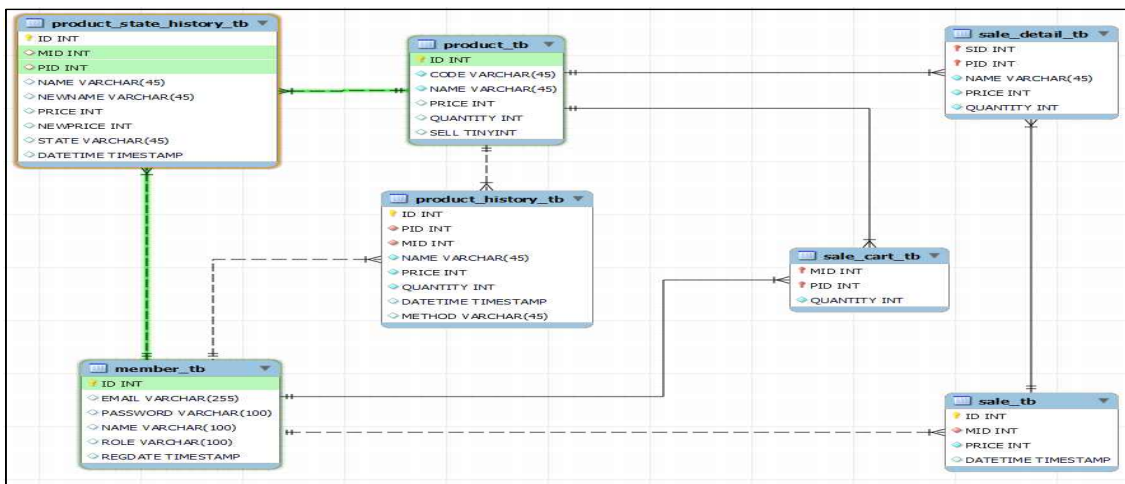
**product\_history\_tb : 상품 수량 변동 내역 저장 테이블**

### product\_history\_tb - product\_tb 관계

product\_history\_tb는 product\_tb의 primary key를 참조합니다. 상품 수량 변동 내역의 한 row는 한 개의 상품에 대응하므로 N:1 관계입니다.

### product\_history\_tb - member\_tb 관계

product\_history\_tb는 member\_tb의 primary key를 참조합니다. 상품 수량 변동 내역의 한 row는 한 멤버에 대응하므로 N:1 관계입니다.



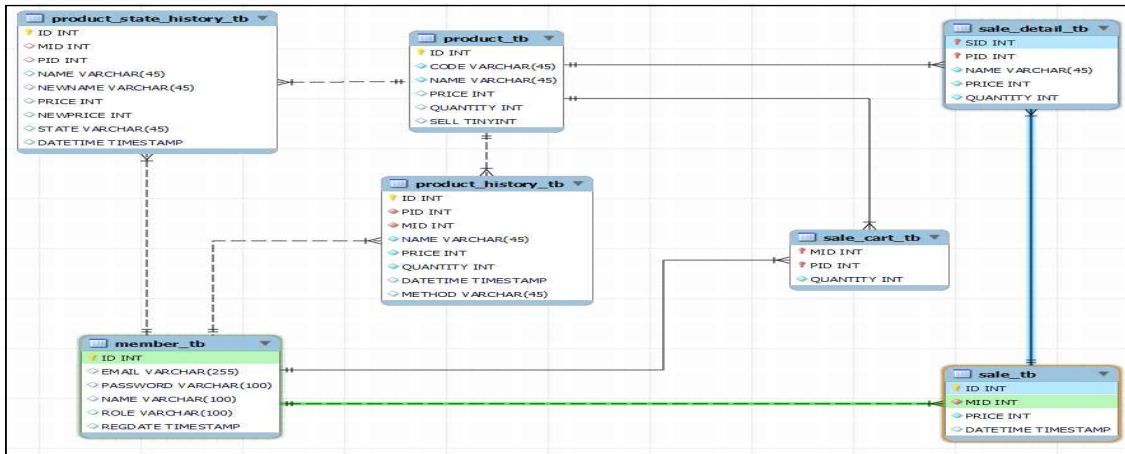
**product\_state\_history\_tb : 상품 상태 변동 내역 저장 테이블**

### product\_state\_history\_tb - product\_tb 관계

product\_state\_history\_tb는 product\_tb의 primary key를 참조합니다. 상품 상태 변동 내역의 한 row는 한 개의 상품에 대응하므로 N:1 관계입니다.

### product\_state\_history\_tb - member\_tb 관계

product\_state\_history\_tb는 member\_tb의 primary key를 참조합니다. 상품 수량 변동 내역의 한 row는 한 멤버에 대응하므로 N:1 관계입니다.



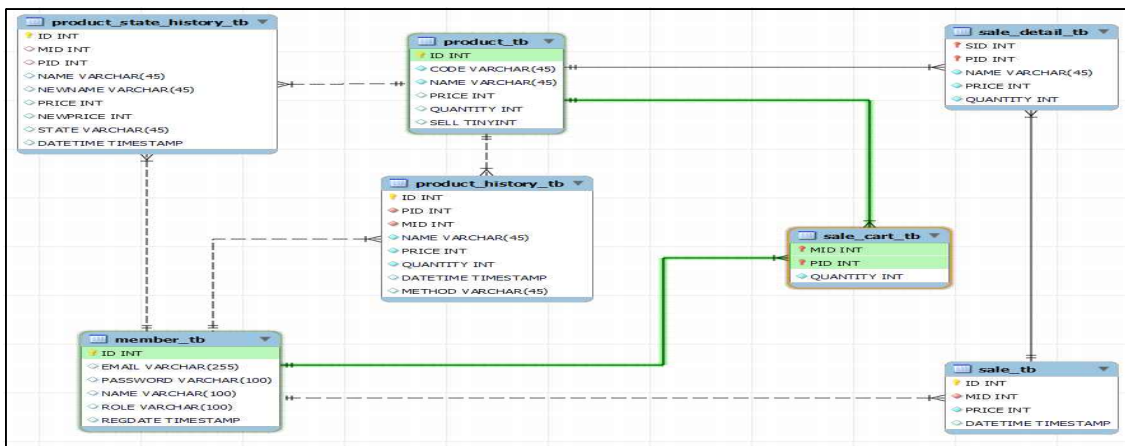
## sale\_tb : 계산 내역 저장 테이블

### sale\_tb - member\_tb 관계

sale\_tb는 member\_tb의 primary key를 참조합니다. 계산을 어떤 직원이 처리했는지 알 수 있습니다. 한 건의 계산 내역에 대해서 한명의 직원이 처리하므로 N:1 관계입니다.

### sale\_tb - sale\_detail\_tb 관계

sale\_tb는 sale\_detail\_tb의 primary key를 참조합니다. 어떤 한 건의 계산 내역에 대해서 어떤 제품을 구매했는지 알 수 있습니다. 한 건의 계산 내역에 대해서 복수개의 제품을 구매할 수 있기 때문에 1:N 관계입니다.



## sale\_cart\_tb : 대기 계산 상품 임시 저장 테이블

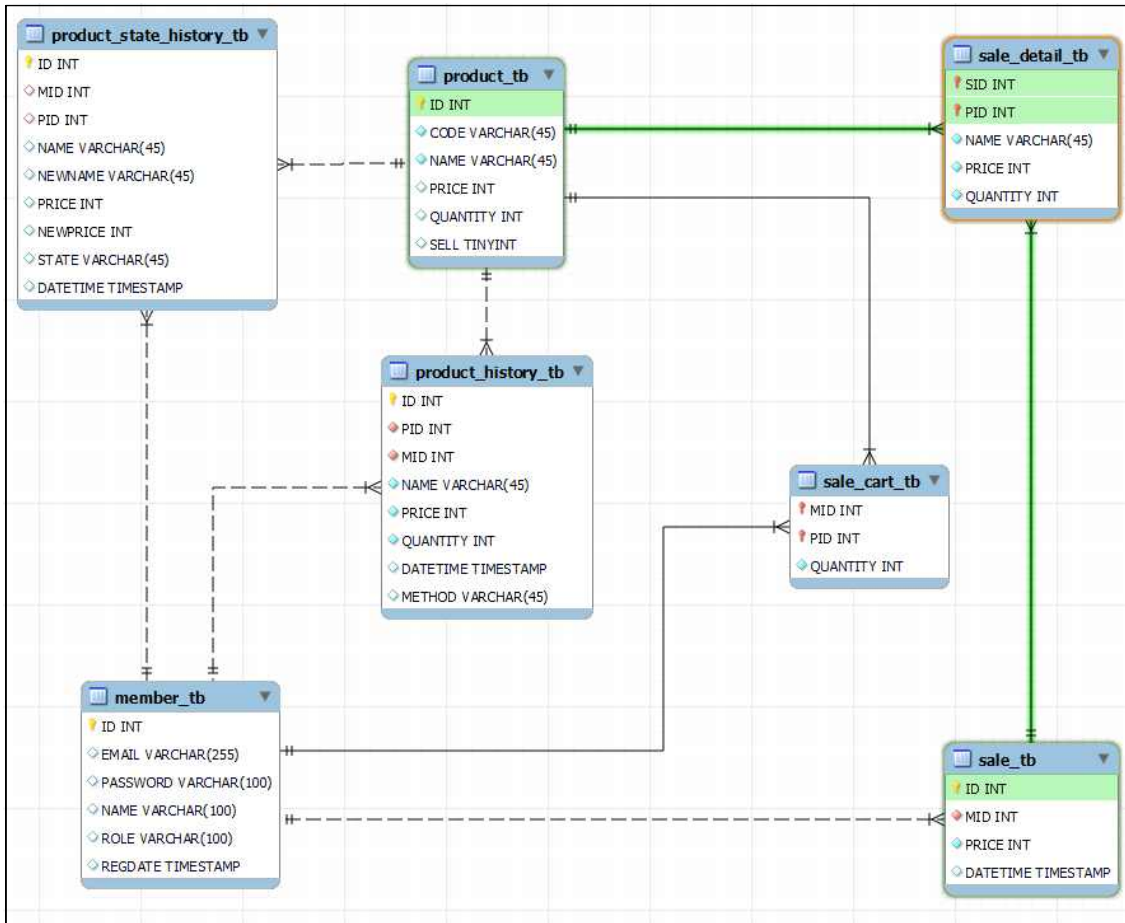
### sale\_cart\_tb - member\_tb 관계

sale\_cart\_tb는 member\_tb의 primary key를 참조합니다. 계산 대기 내역 추가를 어떤 직원이 처리했는지 알 수 있습니다. 한 건의 계산 대기 내역 추가는 한명의 직원이 처리하므로 N:1 관계입니다.

### sale\_cart\_tb - product\_tb 관계

sale\_cart\_tb는 product\_tb의 primary key를 참조합니다. 어떤 제품을 계산 대기 내역에 추가했는지 알 수 있습니다. 한 건의 계산 대기 내역의 row는 한 개의 제품을 처리하므로 N:1 관계입니다.





## sale\_detail\_tb : 계산 상세정보 저장 테이블

### sale\_detail\_tb - sale\_tb 관계

sale\_detail\_tb는 sale\_tb의 primary key를 참조합니다. 제품 상세 테이블은 어떤 계산 내역에 연결되는지 알 수 있습니다. 한 건의 계산 상세 내역의 row는 한 개의 계산 내역과 일치하므로 N:1 관계입니다.

### sale\_detail\_tb - product\_tb 관계

sale\_detail\_tb는 product\_tb의 primary key를 참조합니다. 제품 상세 테이블은 어떤 상품을 구매한건지 알 수 있습니다. 한 건의 계산 상세 내역의 row는 한 개의 제품과 일치하므로 N:1 관계입니다.

### 3) 테이블 DDL

#### ● member\_tb

```
CREATE TABLE `member_tb` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `EMAIL` varchar(255) DEFAULT NULL,  
  `PASSWORD` varchar(100) DEFAULT NULL,  
  `NAME` varchar(100) DEFAULT NULL,  
  `ROLE` varchar(100) DEFAULT NULL,  
  `REGDATE` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`ID`),  
  UNIQUE KEY `EMAIL` (`EMAIL`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb3
```

#### ● product\_tb

```
CREATE TABLE `product_tb` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `CODE` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `NAME` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `PRICE` int DEFAULT NULL,  
  `QUANTITY` int DEFAULT NULL,  
  `SELL` tinyint DEFAULT '0',  
  PRIMARY KEY (`ID`),  
  UNIQUE KEY `CODE` (`CODE`)  
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8mb3  
COLLATE=utf8_unicode_ci
```

#### ● product\_history\_tb

```
CREATE TABLE `product_history_tb` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `PID` int NOT NULL,  
  `MID` int NOT NULL,  
  `NAME` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `PRICE` int NOT NULL,  
  `QUANTITY` int NOT NULL,  
  `DATETIME` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `METHOD` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `ph_ibfk_1` (`PID`),  
  KEY `ph_ibfk_2` (`MID`),  
  CONSTRAINT `ph_ibfk_1` FOREIGN KEY (`PID`) REFERENCES `product_tb` (`ID`),  
  CONSTRAINT `ph_ibfk_2` FOREIGN KEY (`MID`) REFERENCES `member_tb` (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=53 DEFAULT CHARSET=utf8mb3  
COLLATE=utf8_unicode_ci
```

### ● product\_state\_history\_tb

```
CREATE TABLE `product_state_history_tb` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `MID` int DEFAULT NULL,  
  `PID` int DEFAULT NULL,  
  `NAME` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `NEWNAME` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `PRICE` int DEFAULT NULL,  
  `NEWPRICE` int DEFAULT NULL,  
  `STATE` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,  
  `DATETIME` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`ID`),  
  KEY `psh_ibfk_1` (`PID`),  
  KEY `psh_ibfk_2` (`MID`),  
  CONSTRAINT `psh_ibfk_1` FOREIGN KEY (`PID`) REFERENCES `product_tb` (`ID`),  
  CONSTRAINT `psh_ibfk_2` FOREIGN KEY (`MID`) REFERENCES `member_tb` (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=36 DEFAULT CHARSET=utf8mb3  
COLLATE=utf8_unicode_ci
```

### ● sale\_tb

```
CREATE TABLE `sale_tb` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `MID` int NOT NULL,  
  `PRICE` int NOT NULL,  
  `DATETIME` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`ID`),  
  KEY `or_ibfk` (`MID`),  
  CONSTRAINT `or_ibfk` FOREIGN KEY (`MID`) REFERENCES `member_tb` (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8mb3  
COLLATE=utf8_unicode_ci
```

### ● sale\_cart\_tb

```
CREATE TABLE `sale_cart_tb` (  
  `MID` int NOT NULL,  
  `PID` int NOT NULL,  
  `QUANTITY` int NOT NULL,  
  PRIMARY KEY (`MID`,`PID`),  
  KEY `ca_ibfk_1` (`MID`),  
  KEY `ca_ibfk_2` (`PID`),  
  CONSTRAINT `ca_ibfk_1` FOREIGN KEY (`MID`) REFERENCES `member_tb` (`ID`),  
  CONSTRAINT `ca_ibfk_2` FOREIGN KEY (`PID`) REFERENCES `product_tb` (`ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8_unicode_ci
```

● sale\_detail\_tb

```
CREATE TABLE `sale_detail_tb` (  
  `SID` int NOT NULL,  
  `PID` int NOT NULL,  
  `NAME` varchar(45) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,  
  `PRICE` int NOT NULL,  
  `QUANTITY` int NOT NULL,  
  PRIMARY KEY (`SID`,`PID`),  
  KEY `od_ibfk_1` (`SID`),  
  KEY `od_ibfk_2` (`PID`),  
  CONSTRAINT `od_ibfk_1` FOREIGN KEY (`SID`) REFERENCES `sale_tb` (`ID`),  
  CONSTRAINT `od_ibfk_2` FOREIGN KEY (`PID`) REFERENCES `product_tb` (`ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8_unicode_ci
```

## 6. 사용 기술

### 1) AJAX

통계 페이지에서 일별, 주별, 월별 판매량, 판매액을 조회할 수 있다. 이때 사용자는 일, 주, 월 날짜를 선택할 수 있다. 사용자가 날짜를 조회하기 버튼을 클릭하면 Ajax 통신을 통해 사용자가 선택한 날짜 정보가 서버로 전송된다. 서버에서는 이를 기반으로 데이터베이스에서 관련 데이터를 조회, 가공해 json 형식으로 전송한다. 클라이언트 자바스크립트에서는 이 데이터를 전송 받아 차트를 그린다.

### 2) Google chart API

통계 서비스에서 일별, 주별, 월별 판매량, 판매액을 조회할 때, 구글 차트의 파이차트와 테이블을 이용해 시각화해 사용자에게 통계 정보를 보여준다.

### 3) 스프링 인터셉터

모든 HTTP 요청을 로그로 남기는 기능, 인증 처리를 위한 기능을 스프링 인터셉터로 구현해 모든 HTTP 요청에 대해 logs 폴더 내에 log 파일이 남는다. 또 로그인하지 않은 상태에서 사용자가 url을 이용해 특정 페이지에 접근하려고 한다면 인터셉터를 통해 차단, 로그인 페이지로 이동한다.

## 7. 오픈소스


### 1) 테이블 페이징 구현

<https://codepen.io/jaehee/pen/mRmNEX>

위 사이트의 소스코드를 기반으로 테이블 페이징을 구현했습니다.

## 8. 기타

### 1) 로그



File	Line	Timestamp	Level	Message
logs	17	2022-06-01 11:26:25.137	INFO	25036 --- [http-nio-8080-exec-1] c.e.c.web.LogInterceptor : postHandle [ModelAttributeView [v
POS.log	18	2022-06-01 11:26:25.705	INFO	25036 --- [http-nio-8080-exec-1] c.e.c.web.LogInterceptor : RESPONSE [b0197f81-a9b1-4b7
POS.log.2022-05-29-0.log	19	2022-06-01 11:26:26.840	INFO	25036 --- [http-nio-8080-exec-4] c.e.c.web.LogInterceptor : REQUEST [62422584-0765-439b
POS.log.2022-05-30-0.log	20	2022-06-01 11:26:26.891	INFO	25036 --- [http-nio-8080-exec-4] c.e.c.web.LogInterceptor : postHandle [ModelAttributeView [v
POS.log.2022-05-30-1.log	21	2022-06-01 11:26:26.954	INFO	25036 --- [http-nio-8080-exec-4] c.e.c.web.LogInterceptor : RESPONSE [62422584-0765-439
POS.log.2022-05-31-0.log	22	2022-06-01 11:26:27.027	INFO	25036 --- [http-nio-8080-exec-5] c.e.c.web.LogInterceptor : REQUEST [db747bca-4ba7-4cde
POS.log.2022-06-01-0.log	23	2022-06-01 11:26:27.027	INFO	25036 --- [http-nio-8080-exec-5] c.e.c.web.LogInterceptor : REQUEST [db747bca-4ba7-4cde

스프링 인터셉터를 이용해 로그를 기록하게 함. logs 폴더에 일 단위로 로그 파일이 생성됨. 로그파일의 최대크기는 1MB임.