

REPORT

“ Spring POS 프로그램 보고서 ”



과 목 명	스프링프레임워크
담당교수	신우창 교수님
학 과	소프트웨어 학과
학 번	2017301082
이 름	최지윤
제 출 일	2023-06-13

목차

1. 사용자 매뉴얼
2. 개발자 매뉴얼
3. 기타 개발에 관련하여 노력한 내용

사용자 매뉴얼 목차

1. 메인페이지

2. 직원 관리 페이지

2-1. 직원 등록

2-2. 직원 정보 수정

2-3. 직원 정보 삭제

3. 재고관리

3-1. 상품 등록

3-2. 정보 수정

3-3. 입고

3-4. 정보 삭제

3-5. 입고 기록

3-5-1 날짜 범위 검색

4. 판매 기능

4-1 판매 물품 추가

4-1-1 수량 수정 +

4-1-2 수량 수정 -

4-2 판매 물품 계산

4-3 판매 물품 초기화

5. 통계 기능

5-1 월별 통계

5-2 주간 통계

5-3 일별 통계

1.사용자 매뉴얼

1. 메인페이지

POS 웹

환영합니다.

계정 기능

재고 관리

판매 기능

통계 기능

해당 페이지는 핵심 기능 사용하는 페이지로 이동할 수 있는 메인화면입니다.

계정 기능은 편의점에서 일하는 직원들의 정보를 관리하는 페이지로 이동하고

재고관리는 편의점 물품들을 관리하는 페이지로 이동합니다.

판매기능의 경우는 편의점 판매기능을 하는 페이지로 이동하고

통계 기능의 경우 편의점 매출을 관리하는 페이지로 이동합니다.

2. 직원관리 페이지

Worker List

직원 목록					메인화면으로	신규 등록
index	직원 ID	직원 이름	등록 날짜	직급	정보 수정	정보 삭제
1	aaa	나점장	2023-06-05T08:43:07	manager	수정	삭제
2	bbb	김알바	2020-01-15T22:57:17	worker	수정	삭제
3	ccc	박알바	2020-01-21T05:18:45	worker	수정	삭제
4	ddd	이알바	2020-02-11T19:14:45	worker	수정	삭제
5	eee	최알바	2020-03-27T18:58:23	worker	수정	삭제
6	fff	장알바	2020-04-13T23:59:53	worker	수정	삭제
7	ggg	조알바	2023-06-04T04:53:09	worker	수정	삭제
8	hhh	허알바	2023-06-04T05:04:23	worker	수정	삭제
9	iii	윤알바	2023-06-05T07:49:25	worker	수정	삭제
10	jjj	사알바	2023-06-05T22:31:06	worker	수정	삭제

해당 페이지에서는 편의점 직원들의 정보를 관리하는 페이지입니다.

표에서 편의점에 등록된 직원들의 정보들을 한눈에 볼 수 있습니다.

직원 ID는 고유의 번호 임으로 한번 등록이 되면 바꿀 수 없는 정보입니다.

등록 날짜의 경우는 직원의 새로 등록된 시기를 의미합니다.

직급의 경우 manager/worker 두 종류가 존재하며 매니저분들은 manager, 그 외 분들은 worker로 구분할 수 있습니다.

“메인화면으로” 버튼을 눌러서 메인화면으로 돌아갈 수 있습니다.

“신규등록” 버튼을 눌러서 새로운 직원을 등록시킬 수 있습니다.

해당 직원의 정보와 같은 새로 줄에 수정 버튼을 눌러 정보를 수정/삭제를 할 수 있습니다.

2-1 직원 등록

Worker List

[처음 화면으로](#)

직원 목록					메인화면으로	신규 등록
index	직원 ID	직원 이름	등록 날짜	직급	정보 수정	정보 삭제
1	aaa	나점장	2023-06-05T08:43:07	manager	수정	삭제
2	bbb	김알바	2020-01-15T22:57:17	worker	수정	삭제
3	ccc	박알바	2020-01-21T05:18:45	worker	수정	삭제
4	ddd	이알바	2020-02-11T19:14:45	worker	수정	삭제
5	eee	최알바	2020-03-27T18:58:23	worker	수정	삭제
6	fff	장알바	2020-04-13T23:59:53	worker	수정	삭제
7	ggg	조알바	2023-06-04T04:53:09	worker	수정	삭제
8	hhh	허알바	2023-06-04T05:04:23	worker	수정	삭제
9	iii	윤알바	2023-06-05T07:49:25	worker	수정	삭제
10	jjj	사알바	2023-06-05T22:31:06	worker	수정	삭제
11	test	tester	2023-06-11T19:16:46	manager	수정	삭제

2. 직원 관리 페이지에서 오른쪽 상단의 '신규 등록'을 눌러 직원 등록 페이지로 이동합니다.

직원 등록

해당 페이지는 '신규등록'을 누를 시 이동되는 직원등록페이지입니다.

직원 ID:

해당 페이지에서 빈칸에 직원 정보를 기입 하여 새로운 직원들을 등록할 수 있습니다.

이름:

직원ID의 경우 직원 고유 코드로 한번 설정하면 변경할 수 없습니다.

비밀번호:

비밀번호 확인:

비밀번호의 경우 해당 로그인을 위해 필요한 비밀번호를 입력해야 하며 직급의 경우 해당 리스트를 눌러 manager, worker 중 직급 하나를 설정해야 합니다. '가입 완료' 버튼을 누르면 직원등록 기능이 실행이 됩니다.

직급:

가입 완료

직원 등록이 성공적으로 끝났습니다. 그림과 같은 페이지로 이동해야 직원 등록이 성공적으로 완료됩니다.

[처음 화면으로](#)

2-2 직원 정보 수정

Worker List

직원 목록					메인화면으로	신규 등록
index	직원 ID	직원 이름	등록 날짜	직급	정보 수정	정보 삭제
1	aaa	나점장	2023-06-05T08:43:07	manager	수정	삭제
2	bbb	김알바	2020-01-15T22:57:17	worker	수정	삭제
3	ccc	박알바	2020-01-21T05:18:45	worker	수정	삭제
4	ddd	이알바	2020-02-11T19:14:45	worker	수정	삭제
5	eee	최알바	2020-03-27T18:58:23	worker	수정	삭제
6	fff	장알바	2020-04-13T23:59:53	worker	수정	삭제
7	ggg	조알바	2023-06-04T04:53:09	worker	수정	삭제
8	hhh	허알바	2023-06-04T05:04:23	worker	수정	삭제
9	iii	윤알바	2023-06-05T07:49:25	worker	수정	삭제
10	jjj	사알바	2023-06-05T22:31:06	worker	수정	삭제
11	test	tester	2023-06-11T09:35:53	manager	수정	삭제

2. 직원 관리 페이지에서 수정을 원하는 직원의 '수정' 버튼을 눌러 수정 페이지로 이동합니다.

직원 정보 수정

직원 ID:

"test"

이름:

tester

비밀번호:

비밀번호 확인:

직급:

수정 완료

취소

해당 화면에서 수정 사항을 입력합니다. 이때 ID는 변경하실 수 없습니다.

변경사항을 입력한 후 '수정완료' 눌러 변경사항을 적용 시킵니다.

정보 수정이 성공적으로 끝났습니다. 해당 화면이 나와야 변경된 사항이 성공적으로 적용이 됩니다.

[처음 화면으로](#)

2-3 직원 정보 삭제

Worker List

[처음 화면으로](#)

직원 목록					메인화면으로	신규 등록
index	직원 ID	직원 이름	등록 날짜	직급	정보 수정	정보 삭제
1	aaa	나점장	2023-06-05T08:43:07	manager	수정	삭제
2	bbb	김알바	2020-01-15T22:57:17	worker	수정	삭제
3	ccc	박알바	2020-01-21T05:18:45	worker	수정	삭제
4	ddd	이알바	2020-02-11T19:14:45	worker	수정	삭제
5	eee	최알바	2020-03-27T18:58:23	worker	수정	삭제
6	fff	장알바	2020-04-13T23:59:53	worker	수정	삭제
7	ggg	조알바	2023-06-04T04:53:09	worker	수정	삭제
8	hhh	허알바	2023-06-04T05:04:23	worker	수정	삭제
9	iii	윤알바	2023-06-05T07:49:25	worker	수정	삭제
10	jjj	사알바	2023-06-05T22:31:06	worker	수정	삭제
11	test	tester	2023-06-11T19:16:46	manager	수정	삭제

2. 직원 관리 페이지에서 수정을 원하는 직원의 '삭제' 버튼을 눌러 삭제 페이지로 이동합니다.

ID: test

버튼 위에 자신이 삭제하고자 하는 직원이 맞는지 확

name: tester 님의 정보를 삭제하시겠습니까? 인 후 '예' 버튼을 눌러 삭제 기능을 진행시킵니다.

예

취소

삭제를 원하지 않는다면 '취소' 버튼을 누릅니다.

정보 삭제가 성공적으로 끝났습니다. 해당 화면이 나와야 정상적으로 삭제 기능이 적용됩

[처음 화면으로](#)

니다.

3. 재고 관리

Items List

[처음 화면으로](#)

상품 목록						상품 등록	입고 기록
index	상품 ID	상품 이름	상품 가격	상품 갯수	정보 수정	제품 입고	정보 삭제
1	aaa	바나나우유	1500	10	수정	입고	삭제
2	bbb	초코우유	1200	1	수정	입고	삭제
3	ccc	참치마요 삼각김밥	1000	17	수정	입고	삭제
4	ddd	햇바	1000	14	수정	입고	삭제
5	eee	참이슬	1200	20	수정	입고	삭제
6	fff	신라면	1050	10	수정	입고	삭제

해당 화면은 재고관리 페이지입니다. 해당 페이지에서 입고관리 및 편의점 물품 관리를 할 수가 있습니다.

3-1. 상품 등록

Items List

[처음 화면으로](#)

상품 목록						상품 등록	입고 기록
index	상품 ID	상품 이름	상품 가격	상품 갯수	정보 수정	제품 입고	정보 삭제
1	aaa	바나나우유	1500	10	수정	입고	삭제
2	bbb	초코우유	1200	1	수정	입고	삭제
3	ccc	참치마요 삼각김밥	1000	17	수정	입고	삭제
4	ddd	햇바	1000	14	수정	입고	삭제
5	eee	참이슬	1200	20	수정	입고	삭제
6	fff	신라면	1050	10	수정	입고	삭제

3. 재고관리 페이지에서 '상품등록' 버튼을 눌러 입고 기능을 실행하기 위한 페이지로 이동합니다.

상품 등록

해당 페이지에서 상품의 정보들을 입력 후 등록 완료를 누릅니다.

상품 ID:

상품 이름:

상품 가격:

입고 수량:

등록 완료

취소

이때 기존에 등록된 상품ID를 중복된 ID로 등록을 한다면 등록에 실패하게 되므로 주의하셔야 합니다.

기존에 있는 상품을 입고를 원한다면 3-2. 정보 수정 및 입고 페이지의 설명을 보시면 됩니다.

등록을 원치 않을 시 취소 버튼을 누릅니다.

상품 등록이 성공적으로 끝났습니다.

해당 페이지가 나와야 상품 등록이 성공적으로 끝이 납니다.

[처음 화면으로](#)

3-2. 정보 수정

Items List

[처음 화면으로](#)

상품 목록						상품 등록	입고 기록
index	상품 ID	상품 이름	상품 가격	상품 갯수	정보 수정	제품 입고	정보 삭제
1	aaa	바나나우유	1500	10	수정	입고	삭제
2	bbb	초코우유	1200	1	수정	입고	삭제
3	ccc	참치마요 삼각김밥	1000	17	수정	입고	삭제
4	ddd	핫바	1000	14	수정	입고	삭제
5	eee	참이슬	1200	20	수정	입고	삭제
6	fff	신라면	1050	10	수정	입고	삭제

3. 재고관리 페이지에서 '수정' 버튼을 눌러 정보 수정 기능을 실행하기 위한 페이지로 이동합니다.

등록 상품 정보 수정

상품 ID:
"fff"

상품 이름:

상품 가격:

상품 수량:

해당 페이지에서 제품의 변경사항을 입력합니다. 이때 상품의 ID
는 변경하실 수 없습니다. 상품ID를 수정하기 위해서는 해당 상품
ID의 정보를 삭제(3-4 정보 삭제 참조) 후 다시 등록(3-1 상품등록
참조) 해야 합니다.

해당 페이지가 나와야 상품 정보 수정 기능이 성공적
으로 끝이 납니다.

[처음 화면으로](#)

3-3 입고

Items List

[처음 화면으로](#)

상품 목록						상품 등록	입고 기록
index	상품 ID	상품 이름	상품 가격	상품 갯수	정보 수정	제품 입고	정보 삭제
1	aaa	바나나우유	1500	10	수정	입고	삭제
2	bbb	초코우유	1200	1	수정	입고	삭제
3	ccc	참치마요 삼각김밥	1000	17	수정	입고	삭제
4	ddd	햇바	1000	14	수정	입고	삭제
5	eee	참이슬	1200	20	수정	입고	삭제
6	fff	신라면	1200	10	수정	입고	삭제

3. 재고관리 페이지에서 '입고' 버튼을 눌러 제품 입고 기능을 실행하기 위한 페이지로 이동합니
다.

입고

상품 ID: "fff" 해당 페이지에서 해당 제품의 입고 수량을 입력해줍니다.

상품 이름:
신라면

상품 가격:
1200

상품 입고 수량:

상품 입고가 성공적으로 끝났습니다. 해당 페이지가 나와야 상품 정보 수정 기능이 성공적으로 끝이 납니다.

상품 ID: fff

입고 수량: 5

[처음 화면으로](#)

3-4 정보 삭제

Items List

[처음 화면으로](#)

상품 목록						상품 등록	입고 기록
index	상품 ID	상품 이름	상품 가격	상품 갯수	정보 수정	제품 입고	정보 삭제
1	aaa	바나나우유	1500	10	<input type="button" value="수정"/>	<input type="button" value="입고"/>	<input type="button" value="삭제"/>
2	bbb	초코우유	1200	1	<input type="button" value="수정"/>	<input type="button" value="입고"/>	<input type="button" value="삭제"/>
3	ccc	참치마요 삼각김밥	1000	17	<input type="button" value="수정"/>	<input type="button" value="입고"/>	<input type="button" value="삭제"/>
4	ddd	햇바	1000	14	<input type="button" value="수정"/>	<input type="button" value="입고"/>	<input type="button" value="삭제"/>
5	eee	참이슬	1200	20	<input type="button" value="수정"/>	<input type="button" value="입고"/>	<input type="button" value="삭제"/>
6	fff	신라면	1200	15	<input type="button" value="수정"/>	<input type="button" value="입고"/>	<input type="button" value="삭제"/>

3. 재고관리 페이지에서 '삭제' 버튼을 눌러 제품 삭제 기능을 실행하기 위한 페이지로 이동합니다.

상품ID: fff

상품name: 신라면 제품의 정보를 삭제하시겠습니까?

예

취소

버튼 위에 자신이 삭제하고자 하는 상품 정보가 맞는지 확인 후 '예' 버튼을 눌러 삭제 기능을 진행시킵니다.

삭제를 원하지 않는다면 '취소' 버튼을 누릅니다.

정보 삭제가 성공적으로 끝났습니다.

해당 페이지가 나와야 삭제 기능이 성공적으로 끝납니다.

[처음 화면으로](#)

3-5 입고 기록

Items List

[처음 화면으로](#)

상품 목록						상품 등록	입고 기록
index	상품 ID	상품 이름	상품 가격	상품 갯수	정보 수정	제품 입고	정보 삭제
1	aaa	바나나우유	1500	10	수정	입고	삭제
2	bbb	초코우유	1200	1	수정	입고	삭제
3	ccc	참치마요 삼각김밥	1000	17	수정	입고	삭제
4	ddd	햇바	1000	14	수정	입고	삭제
5	eee	참이슬	1200	20	수정	입고	삭제

3. 재고관리 페이지에서 '입고 기록' 버튼을 눌러 제품 삭제 기능을 실행하기 위한 페이지로 이동합니다.

Receiving Date List

[처음 화면으로](#)

[재고 관리 화면으로](#)

날짜별 입고일					
입고일 검색:	연도-월-일	~	연도-월-일	검색	
index	상품 ID	상품 이름	상품 가격	수량	입고일
1	aaa	바나나우유	1500	10	2023-06-11T22:25:47
2	bbb	초코우유	1200	1	2023-06-11T22:26:14
3	ccc	참치마요 삼각김밥	1000	17	2023-06-11T22:26:37
4	ddd	핫바	1000	14	2023-06-11T22:26:49
5	eee	참이슬	1200	20	2023-06-11T22:27:05

해당 페이지에서 입고된 상품의 입고일과 얼마만큼 들어왔는지 볼 수 있는 페이지입니다. 날짜 범위를 설정해서 설정된 날짜 범위에서 입고 정보를 확인을 할 수가 있습니다.

3-5-1 날짜 범위 검색

Receiving Date List

[처음 화면으로](#)

[재고 관리 화면으로](#)

날짜별 입고일					
입고일 검색:	연도-월-일	~	연도-월-일	검색	
index	상품 ID	상품 이름	상품 가격	수량	입고일
1	aaa	바나나우유	1500	10	2023-06-11T22:25:47
2	bbb	초코우유	1200	1	2023-06-11T22:26:14
3	ccc	참치마요 삼각김밥	1000	17	2023-06-11T22:26:37
4	ddd	핫바	1000	14	2023-06-11T22:26:49
5	eee	참이슬	1200	20	2023-06-11T22:27:05

해당 페이지에서 붉은 네모칸으로 표시된 부분을 눌러 날짜 설정을 해줍니다.

Receiving Date List

[처음 화면으로](#)

[재고 관리 화면으로](#)

날짜별 입고일					
입고일 검색:	2023-06-06	~	2023-06-11	검색	
index	상품 ID	상품 이름	상품 가격	수량	입고일
1	aaa	바나나우유	1500	10	2023-06-11T22:25:47
2	bbb	초코우유	1200	1	2023-06-11T22:26:14
3	ccc	참치마요 삼각김밥	1000	17	2023-06-11T22:26:37
4	ddd	핫바	1000	14	2023-06-11T22:26:49
5	eee	참이슬	1200	20	2023-06-11T22:27:05

검색 버튼을 눌러 해당 날짜 범위 안에 입고 기록을 확인합니다.

Receiving Report

[처음 화면으로](#)

[재고 관리 화면으로](#)

날짜별 입고 기록					
입고일 검색:	연도-월-일	~	연도-월-일	검색	
2023-06-06 ~ 2023-06-11 입고 기록					전체
index	상품 ID	상품 이름	상품 가격	수량	입고일
1	aaa	바나나우유	1500	10	2023-06-11T22:25:47
2	bbb	초코우유	1200	1	2023-06-11T22:26:14
3	ccc	참치마요 삼각김밥	1000	17	2023-06-11T22:26:37
4	ddd	핫바	1000	14	2023-06-11T22:26:49
5	eee	참이슬	1200	20	2023-06-11T22:27:05

해당화면은 날짜 범위 검색의 결과 화면입니다. 해당페이지에서 검색 날짜 범위와 함께 해당 범위 안에 입고 기록 보여줍니다.

Receiving Report

[처음 화면으로](#)

[재고 관리 화면으로](#)

날짜별 입고 기록					
입고일 검색:	연도-월-일	□ ~	연도-월-일	□	검색
2023-06-06 ~ 2023-06-11 입고 기록					전체
index	상품 ID	상품 이름	상품 가격	수량	입고일
1	aaa	바나나우유	1500	10	2023-06-11T22:25:47
2	bbb	초코우유	1200	1	2023-06-11T22:26:14
3	ccc	참치마요 삼각김밥	1000	17	2023-06-11T22:26:37
4	ddd	햇바	1000	14	2023-06-11T22:26:49
5	eee	참이슬	1200	20	2023-06-11T22:27:05

해당 결과 화면에서 전체를 누르면 전체 기간의 입고 기록을 보여주는 페이지로 이동합니다.

4. 판매 기능

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정

해당 페이지는 판매기능을 하는 페이지입니다. 해당 페이지에서 손님이 구매하고자 하는 물품의 ID를 입력하여 리스트에 추가하고 수량 수정 후 결제까지 하는 페이지입니다.

4-1 판매 물품 추가

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정

붉은색으로 표시한 네모칸에 손님이 구매하고자 하는 상품의 ID를 입력합니다.

Wish List

[처음 화면으로](#)

판매 물품 리스트					
aaa		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정

'물품 추가' 버튼을 눌러서 네모칸에 입력된 상품의 ID 아이디를 가진 상품 정보를 리스트에 추가를 합니다.

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정
1	aaa	바나나우유	1500	1	<div>+ -</div>

해당 화면은 판매 물품 리스트에 추가된 화면입니다.

4-1-1 수량 수정 +

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정
1	aaa	바나나우유	1500	1	<div>+ -</div>

해당 '+' 버튼을 눌러 수량을 +1 추가합니다.

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정
1	aaa	바나나우유	1500	2	<div><div>+</div><div>-</div></div>

해당 화면은 4-1-1의 결과화면입니다.

4-1-2 수량 수정 -

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정
1	aaa	바나나우유	1500	2	<div><div>+</div><div>-</div></div>

해당 '-'버튼을 눌러 수량을 1 줄어들게 합니다.

수량이 1일 때 '-'버튼을 누르면 판매 물품리스트에 삭제됩니다.

4-2 판매 물품 계산

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정
1	aaa	바나나우유	1500	2	<div>+ -</div>
2	ccc	참치마요 삼각김밥	1000	1	<div>+ -</div>
3	ddd	햇바	1000	2	<div>+ -</div>

손님이 구매할 물품을 해당 리스트에 추가(4-1 판매물품 추가)가 완료가 되면 계산 버튼을 눌러 물품 계산을 진행합니다.

총 금액은 6000 입니다. 해당 페이지에서는 총 가격이 나오고 '계산' 버튼을 누르면 계산 기능을 진행합니다.

계산 하시겠습니까?

계산을 원하지 않는다면 '취소' 버튼을 눌러 이전 화면으로 돌아갑니다.

구매가 성공적으로 끝났습니다. 해당 페이지가 나와야 상품 계산 기능이 성공적으로 끝이 납니다

[처음 화면으로](#)

4-3 판매 물품 초기화

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정
1	ccc	참치마요 삼각김밥	1000	4	<div>+ -</div>
2	aaa	바나나우유	1500	3	<div>+ -</div>
3	ddd	햇바	1000	5	<div>+ -</div>

손님이 구매할 물품을 해당 리스트에 추가(4-1 판매물품 추가)가 된 상태에서 해당 리스트를 비우기 위해 '초기화'를 눌러 물품 리스트를 비웁니다.

Wish List

[처음 화면으로](#)

판매 물품 리스트					
		물품 추가	계산	초기화	
index	상품 ID	상품 이름	상품 가격	수량	수량 수정

5. 통계 기능

Statistics

[처음 화면으로](#)

판매 내역						
판매 시간 검색	월별 통계	----년 --월 <input type="checkbox"/>	검색	주간 통계	----, --번째 주 <input type="checkbox"/>	검색
일별 통계	연도-월-일 <input type="checkbox"/>	검색				
총 판매량	7			총 매출액	9000	
최다 판매제품:	ID: aaa	Name: 바나나우유		최다 판매제품 판매량:	4	
index	상품 ID	상품 이름	상품 가격	수량	총 가격	판매 시기
1	aaa	바나나우유	1500	2	3000	2023-06-11T23:24:38
2	ccc	참치마요 삼각김밥	1000	1	1000	2023-06-11T23:24:38
3	ddd	핫바	1000	2	2000	2023-06-11T23:24:38
4	aaa	바나나우유	1500	2	3000	2023-06-11T23:27:38

해당 페이지는 매출표를 보여주는 페이지입니다. 총 판매량과 총 매출액 그리고 최대 판매제품의 ID와 이름 그리고 판매량을 보여줍니다. 가장 처음 보여주는 매출은 전체 기간의 매출을 보여주며 한달,일주일,하루 기준으로 검색을 해서 판매량을 볼 수 있습니다.

5-1 월별 통계

Statistics

[처음 화면으로](#)

판매 내역						
판매 시간 검색	월별 통계	----년 --월 <input type="checkbox"/>	검색	주간 통계	----, --번째 주 <input type="checkbox"/>	검색
일별 통계	연도-월-일 <input type="checkbox"/>	검색				
총 판매량	7			총 매출액	9000	
최다 판매제품:	ID: aaa	Name: 바나나우유		최다 판매제품 판매량:	4	
index	상품 ID	상품 이름	상품 가격	수량	총 가격	판매 시기
1	aaa	바나나우유	1500	2	3000	2023-06-11T23:24:38
2	ccc	참치마요 삼각김밥	1000	1	1000	2023-06-11T23:24:38
3	ddd	핫바	1000	2	2000	2023-06-11T23:24:38
4	aaa	바나나우유	1500	2	3000	2023-06-11T23:27:38

해당 페이지에서 붉은 네모칸으로 표시된 부분을 눌러 월(month) 설정을 해줍니다.

Statistics

[처음 화면으로](#)

판매 내역						
판매 시간 검색	월별 통계	2023년 06월 <input type="checkbox"/>	검색	주간 통계	----, --번째 주 <input type="checkbox"/>	검색
일별 통계	연도-월-일 <input type="checkbox"/>	검색				
총 판매량	7			총 매출액	9000	
최다 판매제품:	ID: aaa	Name: 바나나우유		최다 판매제품 판매량:	4	
index	상품 ID	상품 이름	상품 가격	수량	총 가격	판매 시기
1	aaa	바나나우유	1500	2	3000	2023-06-11T23:24:38
2	ccc	참치마요 삼각김밥	1000	1	1000	2023-06-11T23:24:38
3	ddd	핫바	1000	2	2000	2023-06-11T23:24:38
4	aaa	바나나우유	1500	2	3000	2023-06-11T23:27:38

'검색'을 눌러 설정한 기간의 매출표를 보여줍니다.

Statistics

[처음 화면으로](#)

[전체 기간 통계](#)

2023-06 판매 내역							
판매 시간 검색	월별 통계	----년 --월	검색	주간 통계	----, --번째 주	검색	일별 통계
연도-월-일	검색						
총 판매량	7			총 매출액	9000		
최다 판매제품	ID: aaa	Name: 바나나우유		최다 판매제품 판매량	4		
index	상품 ID	상품 이름	상품 가격	수량	총 가격	판매 시기	
1	aaa	바나나우유	1500	2	3000	2023-06-11T23:24:38	
2	ccc	참치마요 삼각김밥	1000	1	1000	2023-06-11T23:24:38	
3	ddd	햇바	1000	2	2000	2023-06-11T23:24:38	
4	aaa	바나나우유	1500	2	3000	2023-06-11T23:27:38	

해당 화면은 5-1 월별 통계의 결과화면입니다.

5-2 주간 통계

Statistics

[처음 화면으로](#)

[전체 기간 통계](#)

2023-06 판매 내역							
판매 시간 검색	월별 통계	----년 --월	검색	주간 통계	----, --번째 주	검색	일별 통계
연도-월-일	검색						
총 판매량	7			총 매출액	9000		
최다 판매제품	ID: aaa	Name: 바나나우유		최다 판매제품 판매량	4		
index	상품 ID	상품 이름	상품 가격	수량	총 가격	판매 시기	
1	aaa	바나나우유	1500	2	3000	2023-06-11T23:24:38	
2	ccc	참치마요 삼각김밥	1000	1	1000	2023-06-11T23:24:38	
3	ddd	햇바	1000	2	2000	2023-06-11T23:24:38	
4	aaa	바나나우유	1500	2	3000	2023-06-11T23:27:38	

해당 페이지에서 붉은 네모칸으로 표시된 부분을 눌러 주(week) 설정을 해줍니다.

Statistics

[처음 화면으로](#)

판매 내역							
판매 시간 검색	월별 통계	----년 --월	검색	주간 통계	2023, 24번째 주	검색	일별 통계
연도-월-일	검색						
총 판매량	7			총 매출액	9000		
최다 판매제품	ID: aaa	Name: 바나나우유		최다 판매제품 판매량	4		
index	상품 ID	상품 이름	상품 가격	수량	총 가격	판매 시기	
1	aaa	바나나우유	1500	2	3000	2023-06-11T23:24:38	
2	ccc	참치마요 삼각김밥	1000	1	1000	2023-06-11T23:24:38	
3	ddd	햇바	1000	2	2000	2023-06-11T23:24:38	
4	aaa	바나나우유	1500	2	3000	2023-06-11T23:27:38	

'검색'을 눌러 설정한 기간의 매출표를 보여줍니다.

Statistics

[처음 화면으로](#)

전체 기간 통계

202324 판매 내역															
판매 시간 검색	월별 통계		----년 --월 <input type="checkbox"/>		검색	주간 통계		----, --번째 주 <input type="checkbox"/>		검색	일별 통계		연도-월-일 <input type="checkbox"/>		검색
총 판매량	7					총 매출액					9000				
최다 판매제품:	ID: aaa	Name: 바나나우유				최다 판매제품 판매량:					4				
index	상품 ID	상품 이름			상품 가격			수량			총 가격		판매 시기		
1	aaa	바나나우유			1500			2			3000		2023-06-11T23:24:38		
2	ccc	참치마요 삼각김밥			1000			1			1000		2023-06-11T23:24:38		
3	ddd	핫바			1000			2			2000		2023-06-11T23:24:38		
4	aaa	바나나우유			1500			2			3000		2023-06-11T23:27:38		

해당 화면은 5-2 주간 통계 화면입니다.

5-3 일별 통계

Statistics

[처음 화면으로](#)

판매 내역											
판매 시간 검색	월별 통계		----년 --월 <input type="checkbox"/>	검색	주간 통계		----, --번째 주 <input type="checkbox"/>	검색	일별 통계	연도-월-일 <input type="checkbox"/>	검색
총 판매량	7				총 매출액	9000					
최다 판매제품:	ID: aaa	Name: 바나나우유			최다 판매제품 판매량:	4					
index	상품 ID	상품 이름		상품 가격	수량		총 가격				판매 시기
1	aaa	바나나우유		1500	2		3000				2023-06-11T23:24:38
2	ccc	참치마요 삼각김밥		1000	1		1000				2023-06-11T23:24:38
3	ddd	핫바		1000	2		2000				2023-06-11T23:24:38
4	aaa	바나나우유		1500	2		3000				2023-06-11T23:27:38

해당 페이지에서 붉은 네모칸으로 표시된 부분을 눌러 일(date) 설정을 해줍니다.

Statistics

[처음 화면으로](#)

판매 내역											
판매 시간 검색	월별 통계		----년 --월 <input type="checkbox"/>	검색	주간 통계		----, --번째 주 <input type="checkbox"/>	검색	일별 통계	2023-06-11 <input type="checkbox"/>	검색
총 판매량	7				총 매출액	9000					
최다 판매제품:	ID: aaa	Name: 바나나우유			최다 판매제품 판매량:	4					
index	상품 ID	상품 이름		상품 가격	수량		총 가격	판매 시기			
1	aaa	바나나우유		1500	2		3000	2023-06-11T23:24:38			
2	ccc	참치마요 삼각김밥		1000	1		1000	2023-06-11T23:24:38			
3	ddd	핫바		1000	2		2000	2023-06-11T23:24:38			
4	aaa	바나나우유		1500	2		3000	2023-06-11T23:27:38			

'검색'을 눌러 설정한 기간의 매출표를 보여줍니다.

Statistics

[처음 화면으로](#)

[전체 기간 통계](#)

2023-06-11 판매 내역													
판매 시간 검색	월별 통계		----년 --월 <input type="checkbox"/>		검색	주간 통계		----, --번째 주 <input type="checkbox"/>		검색	일별 통계	연도-월-일 <input type="checkbox"/>	검색
총 판매량	7					총 매출액	9000						
최다 판매제품:	ID: aaa	Name: 바나나우유				최다 판매제품 판매량:	4						
index	상품 ID	상품 이름			상품 가격	수량			총 가격		판매 시기		
1	aaa	바나나우유			1500	2			3000		2023-06-11T23:24:38		
2	ccc	참치마요 삼각김밥			1000	1			1000		2023-06-11T23:24:38		
3	ddd	햇바			1000	2			2000		2023-06-11T23:24:38		
4	aaa	바나나우유			1500	2			3000		2023-06-11T23:27:38		

해당 화면은 5-3 일별 통계 결과화면입니다.

2. 개발자 매뉴얼

1. 직원 관리

2. 재고 관리

3. 판매 기능

4. 통계 기능

5. config 및 controller

6. jsp

7. DB

1. 직원관리

직원 관리의 핵심 기능: 직원 조회, 직원 등록, 직원 정보 수정, 직원 정보 삭제

packge: Worker

구성 class

class 이름	기능
Worker	직원 정보 생성자
WorkerDao	직원관리시 필요한 DB 연산 메소드 포함
WorkerRegisterRequest	직원 등록 정보 요구사항
WorkerRegisterService	직원 등록
WorkerModifyService	직원 정보 수정

1-1. Worker

직원 정보 생성자 클래스 직원들의 정보 DB의 기입사항을 포함합니다.

-private 변수

private 변수	Type	기능
ID	String	직원 ID
password	String	직원 비밀번호
name	String	직원 이름
start_date	LocalDateTime	직원 등록일
title	String	직원 직급

-생성자

private 변수	Type	파라미터	파라미터Type	기능
ID	String	ID	String	직원 ID
password	String	password	String	직원 비밀번호
name	String	name	String	직원 이름
start_date	LocalDateTime	start_date	LocalDateTime	직원 등록일
title	String	title	String	직원 직급

-setter getter 메소드

메소드 이름	getter	setter
ID	getID()	setID(String ID)
password	getPassword()	setPassword(String password)
name	getName()	setName(String name)
start_date	getStart_date()	setStart_date(LocalDateTime start_date)
title	getTitle()	setTitle(String Title)

1-2 WorkerDao

직원 관리를 위해 DB 명령어를 사용하는 메소드들을 포함

-private 변수

private 변수 이름	Type	기능
jdbcTemplate	JdbcTemplate	jdbcTemplate 연산

-생성자

파라미터	파라미터Type	기능
dataSource	DataSource	dataSource를 이용하여 jdbcTemplate객체 를 구함

-메소드

메소드 이름	Type	파라미터	파라미터Type	기능
selectAll()	List<Worker>	-	-	DB에서 worker 테이블의 모든 튜플을 구함
selectByID()	Worker	ID	String	파라미터로 받은 ID와 같은 worker테이블의 튜플을 구합니다.
insert()	void	worker	Worker	파라미터로 받은 worker의 정보를 worker 테이블에 저장
update()	void	worker	Worker	파라미터로 받은 worker의 정보를

				통해 worker 테이블의 튜플 정보 수정
delete()	void	ID	String	파라미터로 받은 ID와 같은 worker테이블의 튜플을 삭제합니다.

1-3 WorkerRegisterRequest

직원 등록을 위한 정보들을 포함하는 클래스 setter와 getter 메소드

-private 변수

private 변수	Type	기능
ID	String	직원 ID
password	String	직원 비밀번호
confirmPassword	String	직원 비밀번호 확인
name	String	직원 이름
title	String	직원 직급

-setter getter 메소드

메소드 이름	getter	setter
ID	getID()	setID(String ID)
password	getPassword()	setPassword(String password)

confirmPassword	getConfirmPassword()	setConfirmPassword(String password)
name	getName()	setName(String name)
title	getTitle()	setTitle(String Title)

1-4 WorkerRegisterService

직원 등록 기능을 하는 메소드를 가지고 있는 클래스

-private 변수

private 변수 이름	Type	기능
workerDao	WorkerDao	직원 등록 기능을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
workerDao	WorkerDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	regist
Type	void
파라미터	req
파라미터Type	WorkerRegisterRequest
기능	파라미터로 받은 직원의 정보를 DB에 넣기 위해 ID 중복 검사를 진행 후 입력이 가능하면 workerDao클래스의 insert()를 이용하여 DB에 기입

	만약 중복된 ID가 존재한다면 Exception을 발생시킵니다.
--	-------------------------------------

1-5 WorkerModifyService

직원 정보 수정 기능을 하는 메소드를 가지고 있는 클래스

-private 변수

private 변수 이름	Type	기능
workerDao	WorkerDao	직원 등록 기능을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
workerDao	WorkerDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	modify
Type	void
파라미터	req
파라미터Type	Worker
기능	파라미터로 받은 수정된 직원 정보의 변경 사항을 업데이트를 위해 workerDao의 update()메소드를 실행 함으로 직원 정보 수정

2. 재고 관리

재고 관리의 핵심 기능: 상품조회, 상품등록, 상품 정보 수정, 상품 입고, 상품 정보 삭제, 입고 기록, 날짜별 입고 기록 조회

packge: Inventory, receivng

구성 class

class 이름	기능
Inventory	상품 정보 클래스
InventoryDao	상품관리 시 필요한 DB 연산 메소드 포함
InventoryRegisterRequest	상품 등록 정보 요구사항
InventoryRegisterService	상품 등록
InventoryModifyService	상품 정보 수정
InventoryReceivingItemService	상품 입고
Receiving	입고 기록 정보 클래스
ReceivingDao	입고 기록 관리시 필요한 DB연산 메소드 포함
ReceivingRegisterService	입고 기록 등록
ReceivingSearchService	날짜별 입고 기록 검색

2-1. Inventory

상품 정보 생성자 클래스 상품 정보 DB의 기입사항을 포함합니다.

-private 변수

private 변수	Type	기능
------------	------	----

ID	String	제품 ID
name	String	제품 이름
price	int	제품 가격
count	int	제품 수량

-생성자

private 변수	Type	파라미터	파라미터Type	기능
ID	String	ID	String	제품 ID
name	String	name	String	제품 이름
price	int	price	int	제품 가격
count	int	count	int	제품 수량

-setter getter 메소드

메소드 이름	getter	setter
ID	getID()	setID(String id)
name	getName()	setName(String name)
price	getPrice()	setPrice(int price)
count	getCount()	setCount(int count)

2-2 InventoryDao

상품 관리를 위해 DB 명령어를 사용하는 메소드들을 포함

-private 변수

private 변수 이름	Type	기능
---------------	------	----

jdbcTemplate	JdbcTemplate	jdbcTemplate 연산
--------------	--------------	-----------------

-생성자

파라미터	파라미터Type	기능
dataSource	DataSource	dataSource를 이용하여 jdbcTemplate객체 를 구함

-메소드

메소드 이름	Type	파라미터	파라미터Type	기능
selectAll()	List<Inventory>	-	-	DB에서 inventory 테이블의 모든 튜플을 구함
selectByID()	Inventory	ID	String	파라미터로 받은 ID와 동일한 ID를 가진 상품 정보를 Inventory테이블 에서 구합니다.
insert()	void	inventory	Inventory	파라미터로 받은 inventory의 정보를 inventory 테이블에 저장
update()	void	inventory	Inventory	파라미터로 받은 inventory의 정보를 통해

				inventory 테이블의 튜플 정보 수정
delete()	void	inventory	Inventory	파라미터로 받은 inventory의 정보를 통해 같은 inventory테이블 의 튜플을 삭제합니다.

2-3 InventoryRegisterRequest

상품 등록을 위한 정보들을 포함하는 클래스 setter와 getter 메소드

-private 변수

private 변수	Type	기능
ID	String	제품 ID
name	String	제품 이름
price	int	제품 가격
count	int	제품 수량

-setter getter 메소드

메소드 이름	getter	setter
ID	getID()	setID(String id)

name	getName()	setName(String name)
price	getPrice()	setPrice(int price)
count	getCount()	setCount(int count)

2-4 InventoryRegisterService

상품 등록 기능을 하는 메소드를 가지고 있는 클래스

-private 변수

private 변수 이름	Type	기능
inventoryDao	InventoryDao	상품 등록 기능을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
inventoryDao	InventoryDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	regist
Type	void
파라미터	req
파라미터Type	InventoryRegisterRequest
기능	파라미터로 받은 상품의 정보를 DB에 넣기 위해 상품ID 중복 검사를 진행 후 입력이 가능하면 InventoryDao클래스의 insert()를 이용하여 DB에 기입

	만약 중복된 ID가 존재한다면 Exception을 발생시킵니다.
--	-------------------------------------

2-5 InventoryModifyService

상품 정보 수정 기능을 하는 메소드를 가지고 있는 클래스

-private 변수

private 변수 이름	Type	기능
inventoryDao	InventoryDao	상품 등록 기능을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
inventoryDao	InventoryDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	modify
Type	void
파라미터	req
파라미터Type	Inventory
기능	파라미터로 받은 수정된 제품 정보의 변경 사항을 업데이트를 위해 InventoryDao의 update()메소드를 실행 함으로 제품 정보 수정

2-6 InventoryReceivingItemService

상품 입고 기능을 하는 메소드를 가지고 있는 클래스

-private 변수

private 변수 이름	Type	기능
inventoryDao	InventoryDao	상품 등록 기능을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
inventoryDao	InventoryDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	receiving
Type	void
파라미터	req
파라미터Type	InventoryRegisterRequest
기능	파라미터로 받은 입고 수량이 기입된 제품 정보중 ID를 이용하여 기존에 존재하는 상품을 selectByID()메소드를 통해 상품 수량을 얻고 파라미터로 기입된 입고 수량과 합쳐서 다시 수량을 inventoryDao안에 update()메소드를 통해 상품 수량을 수정합니다. 이때 기입된 제품의 입고 수량이 음수일 경우와 기입된 제품 정보의ID를 통해 selectByID()메소드 실행시 일치하는 제품이 존재하지않을 때 각각의 Exception을 발생시킵니다.

2-7 Receiving

입고 기록 정보 생성자 클래스 입고기록의 DB의 기입사항을 포함합니다.

-private 변수

private 변수	Type	기능
ID	String	제품 ID
name	String	제품 이름
price	int	제품 가격
count	int	제품 수량
receiving_date	LocalDateTime	제품 입고일

-생성자

private 변수	Type	파라미터	파라미터Type	기능
ID	String	ID	String	제품 ID
name	String	name	String	제품 이름
price	int	price	int	제품 가격
count	int	count	int	제품 수량
receiving_date	LocalDateTime	receiving_date	LocalDateTime	제품 입고일

-setter getter 메소드

메소드 이름	getter	setter
ID	getID()	setID(String iD)
name	getName()	setName(String name)

price	getPrice()	setPrice(int price)
count	getCount()	setCount(int count)
receiving_date	getReceiving_date()	setReceiving_date(LocalDate Time receiving_date)

2-8 ReceivingDao

입고 기록 관리를 위해 DB 명령어를 사용하는 메소드들을 포함

-private 변수

private 변수 이름	Type	기능
jdbcTemplate	JdbcTemplate	jdbcTemplate 연산

-생성자

파라미터	파라미터Type	기능
dataSource	DataSource	dataSource를 이용하여 jdbcTemplate객체 를 구함

-메소드

메소드 이름	Type	파라미터	파라미터Type	기능
ItemSelectAll()	List<Receiving>	-	-	DB에서 Receiving 테이블의 모든 튜플을 구함
DateSelectAll()	Receiving	start_date	LocalDateTime	파라미터로 받은

		end_date	LocalDateTime	두 변수 값 범위의 receiving_date의 값을 가지는 Receiving 테이블의 튜플을 구함
insert()	void	receiving	Receiving	파라미터로 받은 receiving의 정보를 receiving 테이블에 저장

2-9 ReceivingRegisterService

입고 기록을 등록 하는 메소드를 가지고 있는 클래스

-private 변수

private 변수 이름	Type	기능
receivingDao	ReceivingDao	입고 기록 등록 기능을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
receivingDao	ReceivingDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	regist
Type	void

파라미터	req
파라미터Type	InventoryRegisterRequest
기능	<p>상품 정보를 파라미터로 받아서 입고 기록에 저장을 하기 위해 파라미터로 받는 정보와 저장되는 현재 시간을 Receiving 클래스 변수로 만든 후 receivingDao의 Insert() 메소드를 통해 Receiving 테이블에 입고기록을 등록합니다.</p> <p>해당 파라미터의 타입을 InventoryRegisterRequest로 설정한 이유는 inventoryReceivingItemService의 regist와, inventoryRegisterService의 regist와의 호환성을 위해 설정하였습니다.</p>

2-10 ReceivingSearchService

날짜별 입고 기록 검색

-private 변수

private 변수 이름	Type	기능
receivingDao	ReceivingDao	날짜별 입고 기록 검색 기능을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
receivingDao	ReceivingDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	search
Type	List<Receiving>
파라미터	start_date, end_date

파라미터Type	LocalDate, LocalDate
기능	<p>파라미터로 받은 start_date와 end_date을 LocalDateTime 타입 변수에 넣으면서 시간 부분을 추가하여 넣어주고 receivingDao의 DateSelectAll() 메소드의 두 파라미터로 변환한LocalDateTime 타입 변수에 넣어 날짜범위 안의 입고 기록을 검색합니다.</p>

3 판매 기능

판매 기능의 핵심기능: 상품 검색, 상품판매 리스트(장바구니) 추가, 상품판매 리스트(장바구니) 수정, 상품판매 리스트(장바구니) 초기화, 계산

packge: Sell

구성 class

class 이름	기능
Sell	판매 상품 정보 클래스
SellDao	판매 기능 시 필요한 DB 연산 메소드 포함
SellFunctionService	계산 시 상품의 재고량 수정
SellSearchService	상품 판매 리스트(장바구니)에 추가할 상품 검색 기능

3-1 Sell

판매 기능에 필요한 생성자

-private 변수

private 변수	Type	기능
ID	String	제품 ID
name	String	제품 이름
price	int	제품 가격
count	int	제품 수량

-생성자

private 변수	Type	파라미터	파라미터Type	기능
------------	------	------	----------	----

ID	String	ID	String	제품 ID
name	String	name	String	제품 이름
price	int	price	int	제품 가격
count	int	count	int	제품 수량

-setter getter 메소드

메소드 이름	getter	setter
ID	getID()	setID(String iD)
name	getName()	setName(String name)
price	getPrice()	setPrice(int price)
count	getCount()	setCount(int count)

3-2 SellDao

판매 기능 구현을 위해 DB 명령어를 사용하는 메소드들을 포함

-private 변수

private 변수 이름	Type	기능
jdbcTemplate	JdbcTemplate	jdbcTemplate 연산

-생성자

파라미터	파라미터Type	기능
dataSource	DataSource	dataSource를 이용하여

		jdbcTemplate객체를 구함
--	--	--------------------

-메소드

메소드 이름	Type	파라미터	파라미터Type	기능
ItemSelectAll()	List<Sell>	-	-	DB에서 Inventory 테이블의 모든 튜플을 구함
SellSelectAll()	Sell	ID	String	파라미터로 받은 ID값과 동일한 통해 Inventory 테이블의 튜플을 검색 후 Sell 클래스 변수로 반환
InventorySelectById()	Inventory	ID	String	파라미터로 받은 ID값과 동일한 통해 Inventory 테이블의 튜플을 검색 후 Inventory 클래스 변수로 반환
itmeCountMinus())	void	inventory	Inventory	제품 구매가 이뤄지면 해당 제품의 수량을 수정하여 inventory 테이블의 튜플 값 수정

3-3 SellFunctionService

계산 시 상품의 재고량 수정기능

-private 변수

private 변수 이름	Type	기능
SellDao	SellDao	제품 검색 및 수량 수정을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
sellDao	SellDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	itemCountMinus
Type	int
파라미터	req
파라미터Type	Sell
기능	<p>파라미터로 받은 Sell 클래스 변수의 getID를 이용해sellDao의 InventorySelectByID()메소드의 파라미터로 넣어 재고량을 확인하고 계산 요청 받은 수량 보다 현재 재고량이 더 많은지 확인을 합니다.</p> <p>만약 재고량이 더 적다면 바로 음수의 값을 반환시켜 재고량 계산 및 변경사항 DB에 저장되는 연산이 안되었음을 알립니다.</p> <p>재고량이 더 많다면 inventory 테이블에 계산 후 바뀌는 재고량을 계산하고, inventory 테이블에서 바뀐 재고량을 적용하는 sellDao의 ItemCountMinus()메소드를 통해 바뀐 재고량을 적용시켜줍니다. 해당 기능이 제대로 작동을 하면 1을 반환시켜 제대로 작동이 된걸</p>

	표시해줍니다.
--	---------

3-4 SellSearchService

상품 판매 리스트(장바구니)에 물품 추가 시 필요한 검색기능

-private 변수

private 변수 이름	Type	기능
SellDao	SellDao	제품 검색을 수행하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
sellDao	SellDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	search
Type	Sell
파라미터	ID
파라미터Type	String
기능	<p>파라미터로 받은 값을 통해서 sellDao의 SellSelectByID()메소드를 이용해 inventory 테이블에 물품을 가져와 Sell 클래스 변수에 저장합니다. 이때 받은 ID의 값으로 SellSelectByID()메소드 실행시켰으나 물품을 가져오지 못한다면 Exception을 발생시킵니다.</p> <p>가져온 정보를 다시 새로운 Sell 클래스 변수로 저장하는데 count의 값을 1로 설정한 다음 새로운 Sell 클래스 변수를 반환합니다.</p>

4 통계 기능

통계 기능의 핵심기능: 매출확인, 총 판매량, 총 매출액, 최다 판매량 제품 정보, 월별통계,주간통계,일별통계

packge: Statistics

구성 class

class 이름	기능
Statistics	매출 정보 클래스
StatisticsDao	통계 기능 시 필요한 DB 연산 메소드 포함
StatisticsRegisterService	Statistics 테이블에 튜플 저장
StatisticsCountMaxRequest	최다 판매량 제품 정보 클래스
StatisticsCountMaxService	최다 판매량 제품 구하는 클래스

4-1 Statistics

통계 기능에 필요한 매출 정보 클래스

-private 변수

private 변수	Type	기능
ID	String	제품 ID
name	String	제품 이름
price	int	제품 가격
count	int	제품 수량
sellPrice	int	판매 가격

sellDate	LocalDateTime	판매일
----------	---------------	-----

-생성자

private 변수	Type	파라미터	파라미터Type	기능
ID	String	ID	String	제품 ID
name	String	name	String	제품 이름
price	int	price	int	제품 가격
count	int	count	int	제품 수량
sellPrice	int	sellPrice	int	판매 가격
sellDate	LocalDateTime	sellDate	LocalDateTime	판매일

-setter getter 메소드

메소드 이름	getter	setter
ID	getID()	setID(String iD)
name	getName()	setName(String name)
price	getPrice()	setPrice(int price)
count	getCount()	setCount(int count)
sellPrice	getSellPrice()	setSellPrice(int sellPrice)
sellDate	getSellDate()	LocalDateTime sellDate(LocalDate)

4-2 StatisticsDao

판매 기능 구현을 위해 DB 명령어를 사용하는 메소드들을 포함

-private 변수

private 변수 이름	Type	기능
jdbcTemplate	JdbcTemplate	jdbcTemplate 연산

-생성자

파라미터	파라미터Type	기능
dataSource	DataSource	dataSource를 이용하여 jdbcTemplate객체 를 구함

-메소드

메소드 이름	Type	파라미터	파라미터Type	기능
selectAll()	List<Statistics>	-	-	DB에서 Statistics 테이블의 모든 튜플을 구함
ItemelectALLID()	List< Statistics CountMax Request>	-	-	제품 ID 전체 조회
InventorySelect CountSumByID()	List<	Sreq	Statistics	제품ID를 통해 통계창

	Statistics CountMax Request>		CountMax Request	count조회
ItemSelect Price()	int	ID	String	제품 ID를 통해 price 값 전체 조회
Month SelectAll()	List<Statistics>	year month	String String	월별 통계 구하기
WeekSelectAll()	List<Statistics>	week	String	주간통계 구하기
DateSelectAll()	List<Statistics>	date	LocalDate	일별 통계 구하기
insert()	void	sell	Statistics	통계 내역 추가

4-3 StatisticsRegisterService

통계 내역(매출) 등록

-private 변수

private 변수 이름	Type	기능
statisticsDao	StatisticsDao	매출 통계를 등록하는 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
statisticsDao	StatisticsDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	regist
Type	void
파라미터	req
파라미터Type	Sell
기능	판매된 제품의 정보들을 Statistics 테이블에 등록하기 위해 Statistics 클래스 변수로 만들어서 Statistics 테이블에 등록합니다.

4-4 StatisticsCountMaxRequest

최대 판매량 제품 정보 클래스

-private 변수

private 변수	Type	기능
ID	String	제품 ID
name	String	제품 이름
count	int	제품 수량

-생성자

private 변수	Type	파라미터	파라미터Type	기능
ID	String	ID	String	제품 ID
name	String	name	String	제품 이름
count	int	count	int	제품 수량

-setter getter 메소드

메소드 이름	getter	setter
--------	--------	--------

ID	getID()	setID(String iD)
name	getName()	setName(String name)
count	getCount()	setCount(int count)

4-5 StatisticsCountMaxService

최다 판매량 제품 구하는 클래스

-private 변수

private 변수 이름	Type	기능
statisticsDao	StatisticsDao	최다 판매량 제품을 구하기 위한 메소드의 DB연산

-생성자

파라미터	파라미터 Type	기능
statisticsDao	StatisticsDao	DB 연산 접근을 위한 @Bean 객체 생성

-메소드

메소드 이름	countMax
Type	List<StatisticsCountMaxRequest>
파라미터	-
파라미터Type	-

기능	<p>최다 판매량을 구하기 위해 가장 먼저 물품의 전체 ID를 구한 뒤 List인 IDs에 넣어줍니다. 그런 다음 최대 판매량을 기록한 물품의ID를 저장할 List와 최대 판매량을 비교하며 찾기 위한 변수 maxCount를 선언 합니다. IDs에 있는 ID를 통해 ItemSelecounSumByID()메소드를 이용하여 판매량을 가져오고 가져온 매출량을 maxCount의 값과 비교하여 더 큰값이면 maxCount값을 갱신 후 maxID를 초기화 후 해당 ID를 추가합니다. 만약 maxCount의 값과 같다면 maxID를 비우지 않고 ID값을 추가합니다. 이 작업을 IDs 길이 만큼 루프를 돕니다. 그런 뒤 마지막으로 최대판매량을 기록한 물품의 ID가 저장된 maxID를 반환합니다.</p>
----	--

5 config 및 controller

5-1 DBconfig

JDBC를 이용하여 mysql에 접근하여 프로젝트간 DB사용하도록 해주는 configuration 클래스

@Bean 객체	기능
DataSource	Mysql DB 연결 관리
WorkerDao	worker 테이블의 관리를 위한 연산을 위한 DB 접근
WorkerRegisterService	직원 등록기능을 위한 DB 접근
WorkerModifyService	직원 정보 수정기능을 위한 DB 접근
InventoryDao	물품 관련 연산을 위한 DB 접근
InventoryRegisterService	물품 등록 기능을 위한 DB 접근
InventoryModifyService	물품 정보 수정 기능을 위한 DB 접근
InventoryReceivingItemService	상품 입고 기능을 위한 DB 접근
ReceivingDao	입고 관련 연산을 위한 DB 접근
ReceivingRegisterService	입고 기록 등록 기능을 위한 DB접근
ReceivingSearchService	날짜별 입고 기록 검색기능을 위한 DB접근
SellDao	판매 기능 시 필요한 DB 연산
SellSearchService	계산 시 상품의 재고량 수정을 위한 DB접근
SellFunctionService	상품 판매 리스트(장바구니)에 추가할 상품 검색 기능을 위한 DB접근
StatisticsDao	통계 기능 시 필요한 DB 연산 메소드 포함

StatisticsRegisterService	Statistics 테이블에 튜플 저장하기 위한 DB접근
StatisticsCountMaxService	최다 판매량 제품 구하기 위한 DB 접근

5-2 POS Controller

프로젝트의 controller클래스

메소드 이름	주소	파라미터	전송방식	기능
main	"/"/main"	-	GET	main화면

-변수

접근자	Type	변수 이름	어노테이션	기능
private	WorkerDao	workerDao	@Autowired	worker 테이블의 관리를 위한 연산을 위한 DB 접근
private	WorkerRegisterService	workerRegisterService	@Autowired	직원 등록기능을 위한 DB 접근
private	WorkerModifyService	workerModifyService	@Autowired	직원 정보 수정기능을 위한 DB 접근
private	InventoryDao	inventoryDao	@Autowired	물품 관련 연산을 위한 DB 접근
private	InventoryRegisterService	inventoryRegisterService	@Autowired	물품 등록 기능을 위한 DB 접근
private	InventoryModifyS	inventoryModifyS	@Autowired	물품 정보 수정

	ervice	ervice		기능을 위한 DB 접근
private	InventoryReceivin gltemService	inventoryReceivin gltemService	@Autowired	상품 입고 기능을 위한 DB 접근
private	ReceivingDao	receivingDao	@Autowired	입고 관련 연산을 위한 DB 접근
private	ReceivingRegister Service	receivingRegisterS ervice	@Autowired	입고 기록 등록 기능을 위한 DB접근
private	ReceivingSearchS ervice	receivingSearchSe rvice	@Autowired	날짜별 입고 기록 검색기능을 위한 DB접근
private	SellDao	sellDao	@Autowired	판매 기능 시 필요한 DB 연산
private	SellSearchService	sellSearchService	@Autowired	계산 시 상품의 재고량 수정을 위한 DB접근
private	SellFunctionServic e	sellFunctionServic e	@Autowired	상품 판매 리스트(장바구니) 에 추가할 상품 검색 기능을 위한 DB접근
private	StatisticsDao	statisticsDao	@Autowired	통계 기능 시 필요한 DB 연산 메소드 포함
private	StatisticsRegisterS ervice	statisticsRegisterS ervice	@Autowired	Statistics 테이블에 튜플 저장하기 위한

				DB접근
private	StatisticsCountMa xService	statisticsCountMa xService	@Autowired	최다 판매량 제품 구하기 위한 DB 접근
default	wishList	List<Sell>	-	상품 판매 리스트(장바구니)
default	totalPrice	int	-	(handleSellAgree ment 메소드 안) 총합계산 값을 구하기 위한 변수
default	str	ArrayList<String>	-	(handleSellAgree ment 메소드 안) 구매할 수 없는 상품 ID와 이름을 저장하는 문자열 ArrayList

5-2-a Worker 관련 handle 메소드

메소드 이름	RequestMapping 주소	파라미터	model 식별키	전송 방식	기능
	return 주소				
handleWorker	/worker/worker_opt ion	Model model	workers	POST	직원 관리 화면 DB 연산을 통해 직원 정보를 가져와 list로 받고 해당 정보를 view로 전달
	/worker/worker_opt ion				
handleRegisterWor	/worker/register_w	Model model	registerR	POST	직원 등록 누를시

ker	orker		equeset		신규 직원의 정보를 저장할 새로운 클래스 변수를 생성해 view로 전달
	/worker/register_w orker				
handleRegisterSucc ess	/worker/register_su ccess	WorkerRegisterReq uest WregReq, Model model	worker error	POST	등록할 직원 정보를 파라미터로 받아서 DB에 넣는 연산까지 진행 후 등록한 직원의 정보를 view로 전달 DB연산 시 오류 발생 시 오류 사항 view로 전달
	/worker/register_su ccess 예외 발생시: /worker/register_fail				
handleModify	/worker/modify_wo rker	Worker worker, Model model	worker	POST	수정 버튼을 누를시 해당 직원의 정보를 파라미터로 받아 view로 전달
	/worker/modify_wo rker				
handleModifySucce ss	/worker/modify_su ccess	Worker WregReq, Model model	worker	POST	DB연산을 통해 정보를 수정하고 수정한 정보를 다시 view로 전달
	/worker/modify_su ccess				
handleDelete_agre ement	/worker/delete_agr eement	Worker worker, Model model	worker	POST	삭제 버튼을 누를 시 해당 직원의 정보를 파라미터로 받아 view로 전달
	/worker/delete_agr eement				
handleDelete	/worker/delete_suc cess	Worker worker, Model model	worker	POST	DB연산을 통해 해당 직원 정보 삭제 후 삭제한
	/worker/delete_suc				

	cess				직원 정보 view로 전달
--	------	--	--	--	----------------

5-2-b Inventory 관련 handle 메소드

메소드 이름	RequestMapping 주소	파라미터	model 식별키	전송 방식	기능
	return 주소				
handleInventory	/inventory/inventory_management	Model model	items	GET	DB연산을 통해 제품 정보를 모두 가져와 view로 전달
	/inventory/inventory_management				
handleRegisterItem	/inventory/register_item	Model model	registerRequest	POST	상품 등록 누를 시 신규 물품의 정보를 저장할 새로운 클래스 변수를 생성해 view로 전달
	/inventory/register_item				
handleRegisterSuccess	/inventory/register_success	InventoryRegisterRequest IregReq, Model model	inventory error	POST	입력 받은 제품의 정보를 DB연산을 통해 inventory테이블, receiving테이블에 등록 후 해당 정보 view로 전달 오류 발생 시 오류 사항 view로 전달
	/inventory/register_success 예외 발생 시: /inventory/register_fail				
handleReceivingItem	/inventory/receiving_item	Inventory inventory, Model model	inventory	POST	입고 버튼 누를 시 해당 제품의 정보를 파라미터로 받아 view로 전달
	/inventory/receiving_item				

	g_item				
handleReceivingSuccess	/inventory/receiving_success	InventoryRegisterRequest lregReq, Model model	inventory error	POST	재고량이 수정된 제품의 정보를 파라미터로 받고 해당 파라미터를 이용하여 Receiving테이블과 inventory테이블의 정보를 DB연산을 하고 입력받은 제품의 바뀐 정보를 view로 전달 오류 발생 시 오류 사항 view로 전달
	/inventory/receiving_success 예외 발생 시: /inventory/receiving_fail				
handleModifyItem	/inventory/modify_item	Inventory inventory, Model model	inventory	POST	수정 버튼을 누를 시 해당 제품의 정보를 파라미터로 받아 view로 전달
	/inventory/modify_item				
handleModifyItemSuccess	/inventory/modify_success	Inventory Inventory, Model model	inventory	POST	제품의 수정사항을 파라미터로 받고 변경사항을 DB연산 후 view로 전달
	/inventory/modify_success				
handleDeleteItemAgreement	/inventory/delete_agreement	Inventory Inventory, Model model	inventory	POST	삭제 버튼을 누를 시 해당 제품의 정보를 파라미터로 받아 view로 전달
	/inventory/delete_agreement				
handleDeleteItem	/inventory/delete_success	Inventory Inventory, Model model	inventory	POST	제품의 정보를 파라미터로 받아 해당 정보를 DB연산 후 view로 전달
	/inventory/delete_success				

handleReceivingList	/receiving/receiving_item_list	Model model	items	POST	DB연산을 통해 입고 기록을 모두 가져와 view로 전달
	/receiving/receiving_item_list				
handleReceivingSearch	/receiving/receiving_search	@RequestParam("start_date") @DateTimeFormat (pattern="yyyy-MM-dd") LocalDate start_date,	start_date end_date items	POST	해당 핸들러를 부르는 jsp에서 날짜를 받아와 yyyy-MM-dd 형식의 LocalDate타입으로 파라미터로 받고 해당 파라미터로 DB연산 후 결과와 2개의 파라미터를 view로 전달
	/receiving/receiving_searchItem	@RequestParam("end_date") @DateTimeFormat (pattern="yyyy-MM-dd") LocalDate end_date, Model model			

5-2-C Sell 관련 handle 메소드

메소드 이름	RequestMapping 주소	파라미터	model 식별키	전송 방식	기능
	return 주소				
handleSell	"/sell/sell_manage ment"	-	-	GET	판매 기능 페이지 이동
	"/sell/sell_manage ment"				
handleSellSearch	/sell/sell_search	String search, Model model	items, error	POST	파라미터로 상품 ID를 받아서 DB연산을 통해 정보를 얻고 해당 정보를 상품 판매 리스트(장바구니) 에 추가 한 뒤 view에 전달 오류 발생 시 오류 사항 view로 전달
	/sell/sell_manage ment 에러 발생 시: /sell/sell_search_fail				
handleSellResetList	/sell/reset	Model model	items	POST	초기화 버튼 누를 시 상품 판매 리스트(장바구니) 초기화 하고 view로 초기화된 리스트 값 전달
	/sell/sell_manage ment				
handleSellAgreement	/sell/sell_agreement	Model model	totalPrice, fails	POST	총 합 가격을 구할 수 있는 변수와 구매할 수 없는 상품의ID와 이름을
	/sell/sell_agreement				

					포함한 문자열 list 선언 후 모든 상품 구매 리스트(장바구니)의 물품을 재고량과 계산하여 구매할 수 있는 지 없는 지 확인 후 구매 할 수 없다면 list에 문자열을 포함하고 그렇지 않다면 총합 계산금액을 구함 문자열이 존재하지 않는다면 총합계산금액을 view전달 그렇지 않다면 구매 할 수 없으므로 구매 실패
handleSellSuccess	/sell/sell_success	-	-	POST	상품 판매 리스트(장바구니)에 있는 상품의 정보를 DB연산을 통해 등록 후 상품 판매 리스트(장바구니) 비우기
	/sell/sell_success 구매 불가 시 /sell/sell_fail				
handleSellCountM odifyPlus	/sell/sell_countMod ifyPlus	@RequestParam("c lone_index") int index,	items	POST	상품 판매 리스트(장바구니)의 수량을 1 증가시키고 view로
	/sell/sell_managem				

	ent	Model model			전달
handleSellCountM odifyMinus	/sell/sell_countMod ifyMinus	@RequestParam("c lone_index") int index,Model model	items	POST	상품 판매 리스트(장바구니)의 수량을 1 감소시키고 view로 전달
	/sell/sell_managem ent				

5-2-C Statistics 관련 handle 메소드

메소드 이름	RequestMapping 주소	파라미터	model 식별키	전송 방식	기능
	return 주소				
handleStatistic	/statistics/statistics	Model model	items maxID	GET	DB연산을 통해 매출 통계, 최다판매량 정보를 얻어 view로 전달
	/statistics/statistics				
handleStatisticMonth	/statistics/statistics_ month	@RequestParam("month") String month, Model model	month items maxID	POST	DB연산을 통해 월별매출 통계, 최다판매량 정보를 얻어 view로 전달
	/statistics/statistics_ month				
handleStatisticWeek	/statistics/statistics_ week	@RequestParam("week") String week, Model model	week items maxID	POST	DB연산을 통해 주간매출 통계, 최다판매량 정보를 얻어 view로 전달
	/statistics/statistics_ week				
handleStatisticDate	/statistics/statistics_ date	@RequestParam("date") LocalDate date, Model model	date items maxID	POST	DB연산을 통해 일일매출 통계, 최다판매량 정보를 얻어 view로 전달
	/statistics/statistics_ date				

6. jsp

jsp	기능
main	메인 페이지 해당 페이지에서 대표적인 기능을 실행 가능

6-1 worker 관련 jsp

jsp	기능
worker_option	직원 관리 기능의 메인페이지 등록된 직원들의 정보를 표에 표시
register_worker	worker_option에서 신규 등록 버튼을 누르면 나오는 페이지 새로운 직원을 등록 할 수 있는 페이지
register_success	직원 등록이 성공하면 나오는 페이지
register_fail	직원 등록이 실패하면 나오는 페이지 오류 사항도 같이 표시
modify_worker	worker_option에서 수정 버튼을 누르면 나오는 페이지 등록되어있는 직원의 정보를 수정 가능
modify_success	직원 정보 수정이 성공하면 나오는 페이지
delete_agreement	worker_option에서 삭제 버튼을 누르면 나오는 페이지 해당 직원의 ID가 나오고 삭제하는지 재차 질문

delete_success	delete_agreement에서 확인 버튼을 누를 시 직원 정보가 삭제되고 삭제에 성공하면 나오는 페이지
----------------	---

6-2 Inventory 관련 jsp

디렉토리	jsp	기능
/inventory/	inventory_management	재고 관리의 메인 페이지 등록된 상품의 정보를 표에 표시
/inventory/	register_item	inventory_management에서 상품 등록 버튼을 누르면 나오는 페이지 기존에 존재하지 않는 제품의 정보를 입력하여 상품 등록 가능
/inventory/	register_success	상품 등록이 성공하면 나오는 페이지
/inventory/	register_fail	상품 등록이 성공하면 나오는 페이지
/inventory/	modify_item	inventory_management에서 수정 버튼을 누르면 나오는 페이지 등록된 상품의 정보를 수정 가능
/inventory/	modify_success	상품 수정이 성공하면 나오는 페이지
/inventory/	delete_agreement	inventory_management에서 삭제 버튼을 누르면 나오는 페이지 등록된 상품 ID가 나오고 해당 상품의 정보를 삭제할지 재차 질문
/inventory/	delete_success	delete_agreement에서 확인 버튼을 누를 시 상품 정보가 삭제되고 삭제에 성공하면 나오는

		페이지
/inventory/	receiving_item	inventory_management에서 입고 버튼을 누르면 나오는 페이지 등록된 상품의 재고를 추가 가능
/inventory/	receiving_success	상품 입고가 성공하면 나오는 페이지 상품 입고를 진행한 상품의 ID와 입고수량을 표시
/inventory/	receiving_fail	상품의 입고가 실패하면 나오는 페이지 상품 입고가 실패 사유를 함께 표시
/receiving/	receiving_item_list	inventory_management에서 입고 기록 버튼을 누르면 나오는 페이지 입고 기록을 표에 표시
/receiving/	receiving_searchItem	receiving_item_list에서 날짜를 입력받아 해당 날짜 범위 안에 존재하는 입고 기록을 보여주는 페이지 날짜 범위와 입고 기록을 표에 표시

6-3 Sell 관련 jsp

디렉토리	jsp	기능
/sell/	sell_management	<p>판매 기능의 메인 페이지</p> <p>상품 판매 리스트(장바구니)를 표시</p> <p>상품 ID를 기입하고 물품 검색을 하면 리스트에 추가 되면서 추가된 상품을 볼 수 있음</p> <p>초기화 버튼을 누르면 상품 판매리스트(장바구니)의 모든 정보가 초기화</p>
/sell/	sell_search_fail	<p>sell_management에서 상품 ID를 기입하고 물품 검색 버튼을 눌렀을 때 검색에 실패하면 나오는 페이지</p> <p>검색 실패 사유를 함께 표시</p>
/sell/	sell_agreement	<p>sell_management에서 계산 버튼을 누르면 나오는 페이지</p> <p>상품 판매 리스트(장바구니)에 등록된 상품들의 가격을 계산하여 총합금액을 표시</p> <p>확인을 누르면 계산 진행</p> <p>취소를 누르면 직전화면으로 뒤로가기</p>
/sell/	sell_success	<p>계산이 성공하면 나오는 페이지</p>
/sell/	sell_fail	<p>sell_management에서 계산 버튼을 누르고 계산이 불가능한 상품이 존재할 경우 나오는 페이지</p> <p>계산 불가능한 사유도 같이 표시</p>

6-4 Statistics 관련 jsp

디렉토리	jsp	기능
/statistics/	statistics	<p>통계 기능의 메인페이지</p> <p>전체 기간의 매출 정보 표시</p> <p><%%>태그를 통해 총 매출액과 총 판매량을 표시</p> <p>최다 판매 제품 정보, 최다 판매 제품 판매량 표시</p>
/statistics/	statistics_month	<p>statistics에서 월별통계의 날짜를 입력 후 검색 버튼을 누르면 나오는 페이지</p> <p>statistics의 페이지에서 표시하는 정보를 입력한 날짜(단위: 월)의 맞춰서 정보를 표시</p>
/statistics/	statistics_week	<p>statistics에서 주간통계의 날짜를 입력 후 검색 버튼을 누르면 나오는 페이지</p> <p>statistics의 페이지에서 표시하는 정보를 입력한 날짜(단위: 주)의 맞춰서 정보를 표시</p>
/statistics/	statistics_date	<p>statistics에서 일별통계의 날짜를 입력 후 검색 버튼을 누르면 나오는 페이지</p> <p>statistics의 페이지에서 표시하는 정보를 입력한 날짜(단위: 일)의 맞춰서 정보를 표시</p>

7 DB

데이터베이스 이름: spring_pos

-table 구성

table이름	기능
worker	직원 정보 데이터 베이스
inventory	제품 정보 데이터 베이스
receiving	입고 기록 데이터 베이스
statistics	판매 내역 및 통계 데이터베이스

7-1 worker 스키마

스키마 이름	스키마 타입	기능
ID	varchar(255)	직원 ID -기본키
password	varchar(255)	직원 password
name	varchar(255)	직원 이름
start_date	datetime	등록일
title	varchar(255)	직급

7-2 inventory 스키마

스키마 이름	스키마 타입	기능
ID	varchar(255)	제품 ID -기본키
name	varchar(20)	제품 이름
price	int	제품 가격
count	int	제품 수량

7-3 receiving 스키마

스키마 이름	스키마 타입	기능
num	int	새로운 튜플이 생성 될 시 자동으로 1씩 증가 -기본키
ID	varchar(255)	제품 ID
name	varchar(20)	제품 이름
price	int	제품 가격
count	int	제품 수량
receiving_date	datetime	입고일

7-4 statistics 스키마

스키마 이름	스키마 타입	기능
num	int	새로운 튜플이 생성 될 시 자동으로 1씩 증가 -기본키 -자동으로 +1
ID	varchar(255)	제품 ID
name	varchar(20)	제품 이름
price	int	제품 가격
count	int	제품 수량
sellPrice	int	판매 가격
sellDate	datetime	판매일

3. 기타 개발에 관련하여 노력한 내용

이번 프로젝트를 하면서 신경 쓰면서 했던 부분은 핵심적인 기능과 함께 정보의 등록, 수정, 삭제가 확실하게 구현하는 것에 초점을 맞추면서 진행하였습니다. 그리고 사용자의 잘못된 값 입력으로 오류가 발생하면 Exception을 발생시키고 왜 오류가 났는지 표시하는 페이지를 만들어 사용자가 잘못된 값을 입력하여 프로그램 오작동을 최대한 줄일 수 있게 개발하였습니다.

최대한 클래스, 메소드, 변수 이름을 보고 어떤 역할을 하는지를 감이 잡힐 수 있도록 설정하였고 프로그램을 코딩할 때 또한 최대한 간단하면서 직관적으로 코딩을 하였습니다. 그리고 패키지 및 디렉토리를 핵심 기능에 맞게 정리하여 개발을 진행하여 프로그램 유지보수가 쉬울 수 있게 신경써서 코딩하였습니다.

효율적인 프로그램의 동작을 위해 중첩 반복문을 최대한 배제하였습니다. 그리고 프로젝트를 진행하면서 최대한 배운 내용 안에서 프로그램을 구현하는 것을 목표로 하였고 정보를 보기 좋게 하기 위해서 html의 여러 태그들을 조금 더 복습하였습니다. 자바 부분에서는 다양한 날짜 타입을 알아보았고 형변환시 필요한 값들을 자동으로 넣어주는 메소드 들도 새롭게 알았습니다. 기본적인 sql문을 복습하는 계기가 되었으며 sql 연산 중 datetime 칼럼의 튜플을 검색할 때 사용할 수 있는 함수, month(), yearweek(), date()를 새롭게 알았습니다.

그리고 화이트라벨 오류코드 3자리 숫자의 의미를 새롭게 알아서 개발하는데 오류 난 부분을 조금 더 쉽게 고칠 수 있었습니다.