

한국스마트정보교육원

51 기 1 팀 프로젝트 완료 보고서

프로젝트명	
계약 주문 통합 관리 솔루션 : 약수터	
프로젝트 팀명	
길중식	
팀 원	박중연
	길범진
	최성식
작성일	2024-07-25

호스팅	https://yaksutor.duckdns.org/
GitHub	https://github.com/ksmart51th-team1/ks51team01

목차

1 . 프로젝트 기본정보

1.1 시스템 흐름도

1.2 주요 기능별 흐름도

2 . 요구사항 정의서

2.1 기능적 요구사항

2.2 상세기능 정의

3 . 업무분담(담당업무)

4 . 설계사양서

4.1 ERD 물리모드

4.2 테이블 구조

4.3 네이밍 규칙

4.4 디렉토리 구조

4.5 패키지 구조

5 . 개발환경

6. 어플리케이션

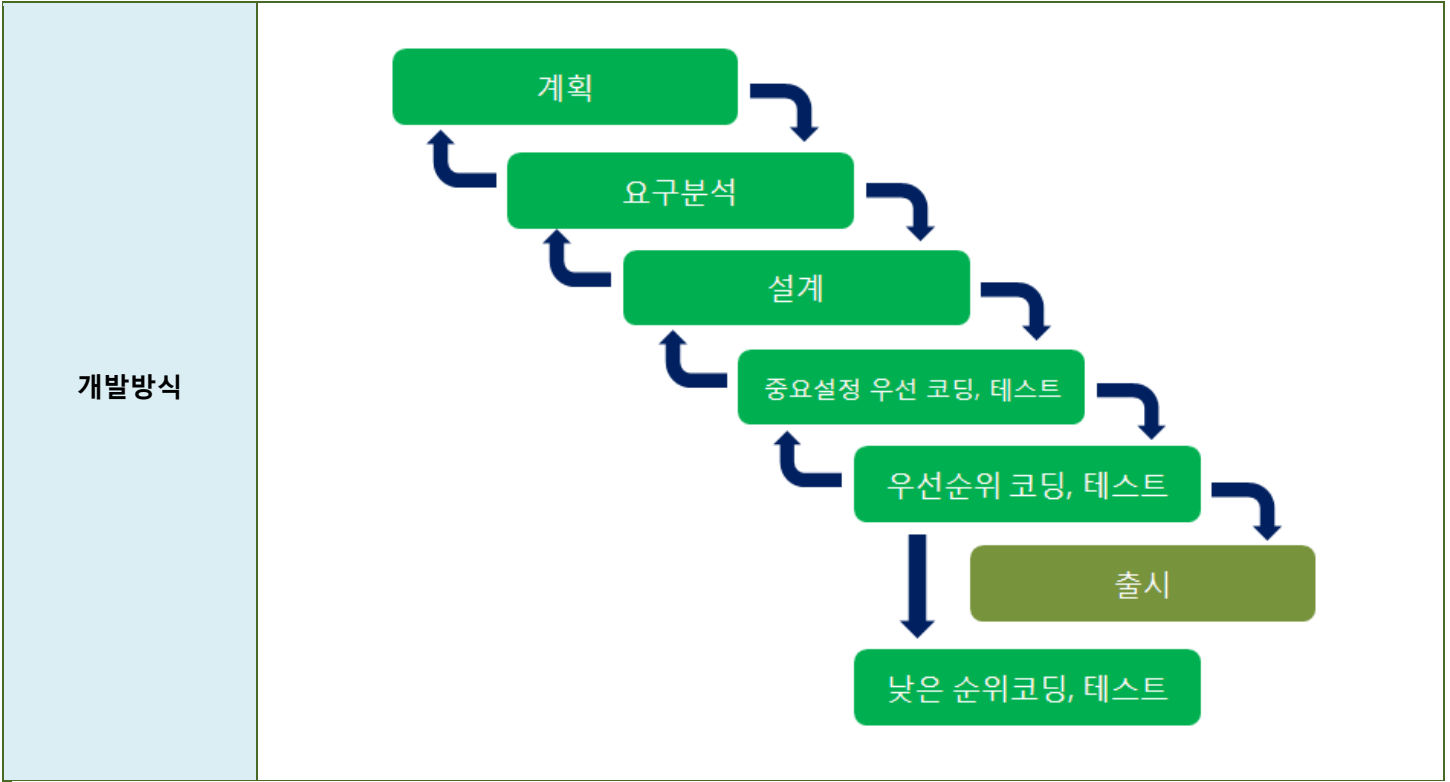
6.1 어플리케이션 프로시저

6.2 서버 및 기타 설정

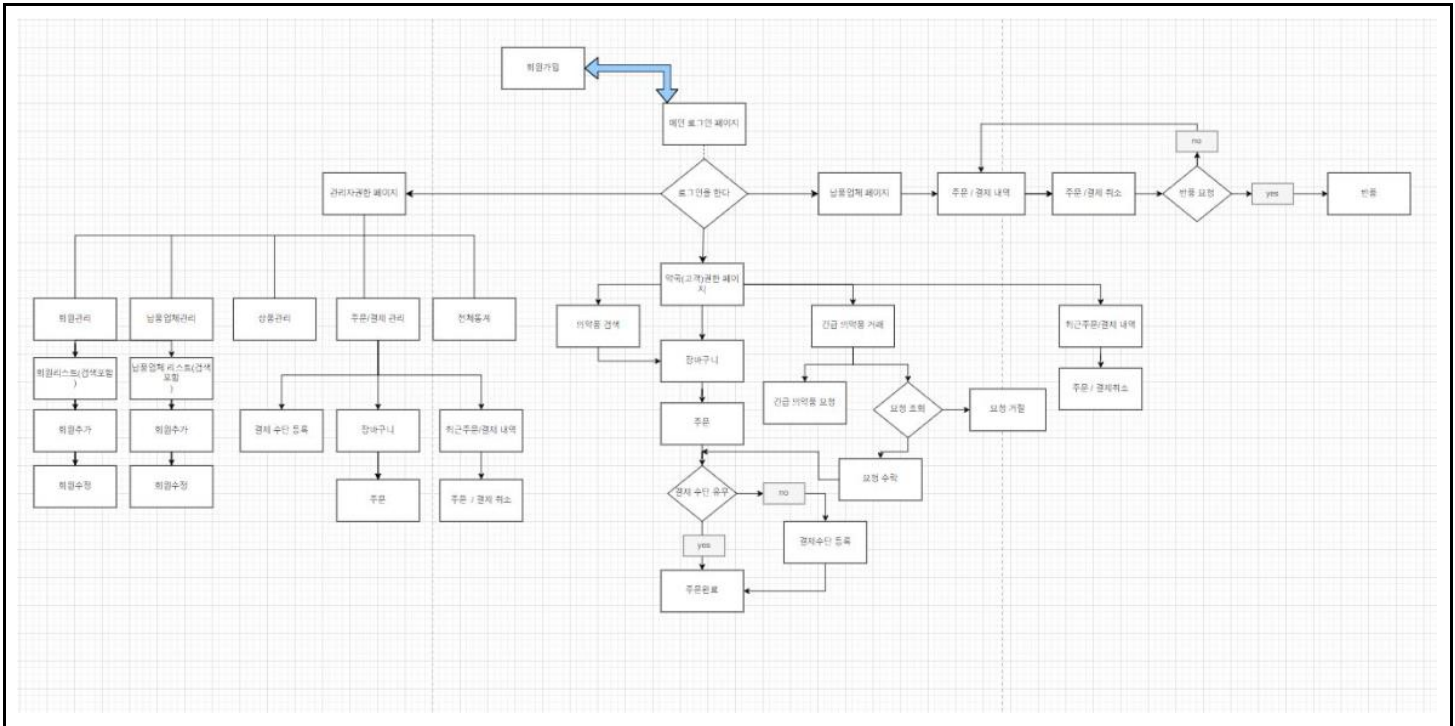
6.3 주요 기능별 정리

1. 프로젝트 기본정보

<p>프로젝트명</p>	<p>제약 주문 통합 관리 솔루션 : 약수터</p>	
<p>개발기간</p>	<p>■ 2024.02.19~2024.08.01(113 일)</p>	
	<p>2024.04.01~ 2024.04.15(15 일)</p>	<p>프로젝트 주제 선정 및 사전조사</p>
	<p>2024.04.16~ 2024.05.17(32 일)</p>	<ul style="list-style-type: none"> - 시스템 구조도 작성 - 설계자료 취합, 수정 및 보완 - ERD 작업 및 메서드, 패키지 정리
	<p>2024.05.18~ 2024.05.25(7 일)</p>	<ul style="list-style-type: none"> - 상세자료 입력 / 기능정의서 작성 - 개발환경 설정 - Hosting 설정
	<p>2024.05.20~2024.05.29(9 일)</p>	<ul style="list-style-type: none"> - 각 기능별 상세기준 및 보완사항 확인 / 수정
	<p>2024.05.29~2022.07.12(45 일)</p>	<ul style="list-style-type: none"> - 테스트, 오류 수정 및 보완 - 미완성 기능 구현 - 상세테이블 및 자료 구현 / 수정
	<p>2024.06.18~2024.08.01(52 일)</p>	<ul style="list-style-type: none"> - 완료보고서 / 최종 배포파일 작성
<p>개발목적</p>	<ol style="list-style-type: none"> 1. 재고 관리 효율화 - 재고 수준을 실시간으로 추적하여 품목의 부족 또는 과다를 방지하고, 효율적인 재고 회전을 유지 2. 발주 프로세스 간편화 - 자동화된 프로세스를 통해 주문서 작성, 납품업체와의 커뮤니케이션, 주문 승인을 간편하게 처리 3. 약국간 부족한 의약품 교환 - 긴급하게 의약품이 부족한 경우 가맹약국으로 부터 의약품을 대신 납품 받을 수 있는 기능 4. 비용 절감 및 수익성 향상 - 불필요한 재고 비용을 줄이고, 시간을 절약하여 약국의 운영 효율성을 높임 	

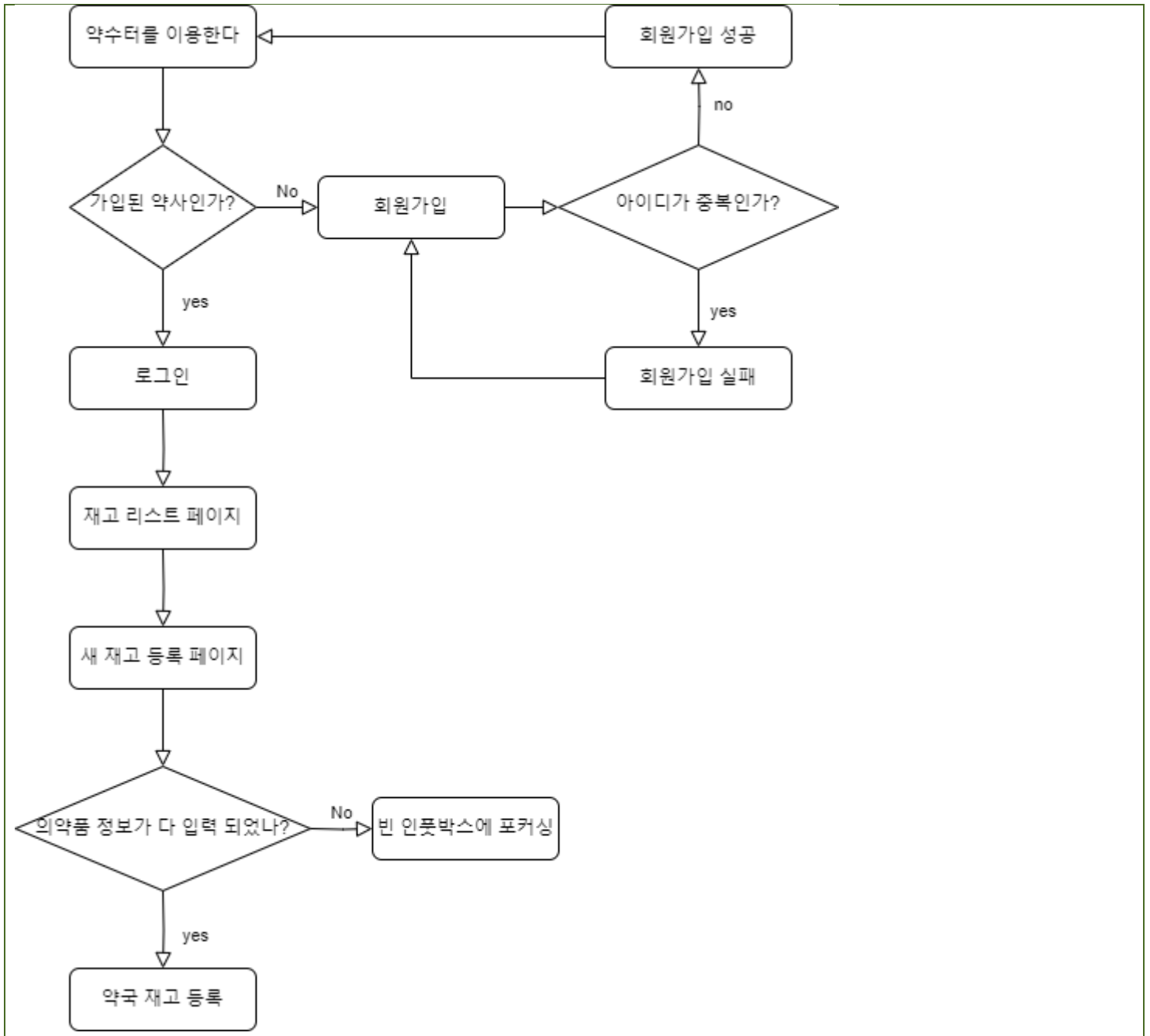


1.1 시스템 구조도

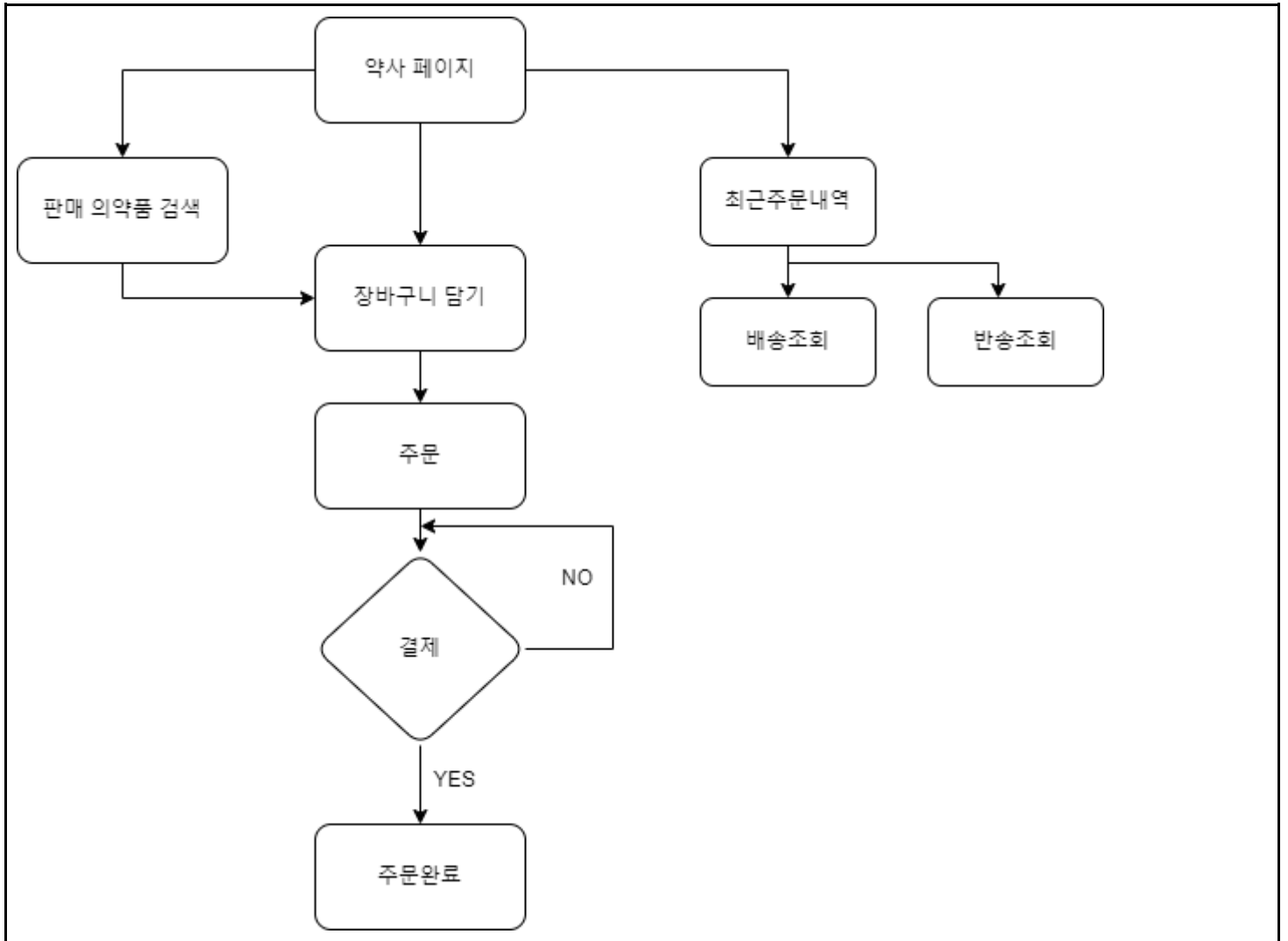


1.2 주요 기능별 흐름도

<재고등록 프로세스 서비스 구조도>



<주문 프로세스 서비스 구조도>



2. 요구사항 정의서

2.1 기능적 요구사항

1.회원관리	1.회원관리	1.회원관리	1.회원관리
1.1 회원등록	1.1 회원등록	1.1 회원등록(본인)	1.1 회원등록(본인)
1.2 회원수정	1.2 회원수정	1.2 회원수정(본인)	1.2 회원수정(본인)
1.3 회원탈퇴	1.3 회원탈퇴	1.3 회원탈퇴(본인)	1.3 회원탈퇴(본인)
1.4 개인회원조회	1.4 개인회원조회	1.4 개인회원조회(본인)	1.4 개인회원조회(본인)
1.6 전체회원검색	1.6 전체회원검색		
2.로그인관리	2.로그인관리	2.로그인관리	2.로그인관리
2.1 로그인	2.1 로그인	2.1 로그인	2.1 로그인
2.2 로그아웃	2.2 로그아웃	2.2 로그아웃	2.2 로그아웃
2.3 아이디찾기	2.3 아이디찾기	2.5 아이디찾기(본인)	2.5 아이디찾기(본인)
2.4 비밀번호 찾기	2.4 비밀번호 찾기	2.4 비밀번호 찾기(본인)	2.4 비밀번호 찾기(본인)
2.5 로그인 이력 조회(검색)	2.5 로그인 이력 조회(검색)		
2.6 로그인 이력 삭제	2.6 로그인 이력 삭제		
3. 판매의약품등록	3.의약품관리	3.의약품관리	3.의약품관리
3.1 의약품등록	3.1 의약품등록	3.1 의약품등록(본인)	
3.2 판매의약품등록 수정	3.2 판매의약품등록 수정	3.2 판매의약품등록 수정(본인)	
3.3 판매의약품삭제	3.3 판매의약품삭제	3.3 판매의약품삭제(본인)	
3.4 판매의약품검색	3.4 판매의약품검색	3.4 판매의약품검색(본인)	3.4 판매의약품검색
4. 결제(주문)관리	4.결제(주문)관리	4.결제(주문)관리	4.결제(주문)관리
4.1 장바구니에 의약품 추가	4.1 장바구니에 의약품 추가		4.1 장바구니에 의약품 추가
4.2 주문 생성	4.2 주문 생성		4.2 주문 생성
4.3 주문 취소	4.3 주문 취소	4.3 주문 취소(본인)	4.3 주문 취소(본인)
4.4 주문 조회	4.4 주문 조회	4.4 주문 조회(본인)	4.4 주문 조회(본인)
4.5 결제 수단 등록	4.5 결제 수단 등록		4.5 결제 수단 등록(본인)
4.6 결제 확인	4.6 결제 확인	4.6 결제 확인(본인)	4.6 결제 확인(본인)
5. 반품/환불 관리	5. 반품/환불 관리	5. 반품/환불 관리	5. 반품/환불 관리
5.1 반품 신청	5.1 반품 신청		5.1 반품 신청(본인)
5.2 반품 신청 내역 조회	5.2 반품 신청 내역 조회	5.2 반품 신청 내역 조회(본인)	5.2 반품 신청 내역 조회(본인)
5.3 반품 취소	5.3 반품 취소	5.3 반품 취소(본인)	5.3 반품 취소(본인)
5.4 반품 요청 수락	5.4 반품 요청 수락	5.4 반품 요청 수락(본인)	
5.5 반품 상태 조회	5.5 반품 상태 조회	5.5 반품 상태 조회(본인)	5.5 반품 상태 조회(본인)

6. 배송 관리	6. 배송 관리	6. 배송 관리	6. 배송 관리
6.1 배송 등록	6.1 배송 등록	6.1 배송 등록(본인)	
6.2 배송 취소	6.2 배송 취소	6.2 배송 취소(본인)	
6.3 배송 상태 추적	6.3 배송 상태 추적	6.3 배송 상태 추적 (본인)	
6.4 배송 상태 확인	6.4 배송 상태 확인	6.4 배송 상태 확인(본인)	6.4 배송 상태 확인(본인)
7. 약국 재고	7. 약국 재고 관리	7. 약국 재고 관리	7. 약국 재고 관리
7.1 재고 등록	7.1 재고 등록		7.1 재고 등록
7.2 재고 수정	7.2 재고 수정		7.2 재고 수정
7.3 재고 조회	7.3 재고 조회		7.3 재고 조회
7.4 재고 삭제	7.4 재고 삭제		7.4 재고 삭제
8. 약국 입고고 이력	8. 약국 입고고 이력	8. 약국 입고고 이력	8. 약국 입고고 이력
8.1 출고 등록	8.1 입/출고 등록		
8.2 출고 수정	8.2 입/출고 수정		
8.3 출고 조회	8.3 출고 조회		8.3 출고 조회
8.4 출고 삭제	8.4 출고 삭제		
9. 약국 정상 재고조사	9. 약국 정상 재고조사	9. 약국 정상 재고조사	9. 약국 정상 재고조사
9.1 재고 조회	9.1 재고 조회		9.1 재고 조회
9.2 재고조사 후 실수량 입력	9.2 재고조사 후 실수량 입력		9.2 재고조사 후 실수량 입력
9.3 재고 현황 업데이트	9.3 재고 현황 업데이트		9.3 재고 현황 업데이트
10. 약국 비정상 재고조사	10. 약국 비정상 재고조사	10. 약국 비정상 재고조사	10. 약국 비정상 재고조사
10.1 재고 조회	10.1 재고 조회		10.1 재고 조회
10.2 재고조사 후 실수량 입력	10.2 재고조사 후 실수량 입력		10.2 재고조사 후 실수량 입력
10.3 비정상 사유 입력	10.3 비정상 사유 입력		10.3 비정상 사유 입력
10.4 재고 현황 업데이트	10.4 재고 현황 업데이트		10.4 재고 현황 업데이트
11. 약국 간 긴급 의약품 거래	11. 약국 간 긴급 의약품 거래	11. 약국 간 긴급 의약품 거래	11. 약국 간 긴급 의약품 거래
11.1 긴급 의약품 요청	11.1 긴급 의약품 요청		11.1 긴급 의약품 요청
11.2 긴급 의약품 요청 조회	11.2 긴급 의약품 요청 조회		11.2 긴급 의약품 요청 조회
11.3 긴급 의약품 요청 수락/거절	11.3 긴급 의약품 요청 수락/거절		11.3 긴급 의약품 요청 수락/거절
11.4 긴급 의약품 요청 취소	11.4 긴급 의약품 요청 취소		11.4 긴급 의약품 요청 취소
12. 문의하기	12. 문의하기	12. 문의하기	12. 문의하기
12.1 문의 등록	12.1 문의 등록	12.1 문의 등록	12.1 문의 등록
12.2 문의 수정	12.2 문의 수정	12.2 문의 수정	12.2 문의 수정
12.3 문의 삭제	12.3 문의 삭제	12.3 문의 삭제	12.3 문의 삭제

12.4 문의 내역 조회	12.4 문의 내역 조회	12.4 문의 내역 조회	12.4 문의 내역 조회
12.5 문의 답변 등록	12.5 문의 답변 등록		
12.6 문의 답변 수정	12.6 문의 답변 수정		
12.7 문의 답변 삭제	12.7 문의 답변 삭제		
13. 커뮤니티	13. 커뮤니티	13. 커뮤니티	13. 커뮤니티
13.1 커뮤니티 글 등록	13.1 커뮤니티 글 등록	13.1 커뮤니티 글 등록	13.1 커뮤니티 글 등록
13.2 커뮤니티 글 수정	13.2 커뮤니티 글 수정	13.2 커뮤니티 글 수정(본인)	13.2 커뮤니티 글 수정(본인)
13.3 커뮤니티 글 삭제	13.3 커뮤니티 글 삭제	13.3 커뮤니티 글 삭제(본인)	13.3 커뮤니티 글 삭제(본인)
13.4 커뮤니티 글 조회	13.4 커뮤니티 글 조회	13.4 커뮤니티 글 조회	13.4 커뮤니티 글 조회
13.5 커뮤니티 댓글 등록	13.5 커뮤니티 댓글 등록	13.5 커뮤니티 댓글 등록	13.5 커뮤니티 댓글 등록
13.6 커뮤니티 댓글 수정	13.6 커뮤니티 댓글 수정	13.6 커뮤니티 댓글 수정(본인)	13.6 커뮤니티 댓글 수정(본인)
13.7 커뮤니티 댓글 삭제	13.7 커뮤니티 댓글 삭제	13.7 커뮤니티 댓글 삭제(본인)	13.7 커뮤니티 댓글 삭제(본인)
14. 공지사항	14.공지사항	14. 공지사항	14. 공지사항
14.1 공지사항 조회	14.1 공지사항 조회	14.1 공지사항 조회	14.1 공지사항 조회
14.2 공지사항 등록	14.2 공지사항 등록		
14.3 공지사항 수정	14.3 공지사항 수정		
14.4 공지사항 삭제	14.4 공지사항 삭제		
15. 자주하는 질문	15. 자주하는 질문	15. 자주하는 질문	15. 자주하는 질문
15.1 자주하는 질문 조회	15.1 자주하는 질문 조회	15.1 자주하는 질문 조회	15.1 자주하는 질문 조회
15.2 자주하는 질문 등록	15.2 자주하는 질문 등록		
15.3 자주하는 질문 수정	15.3 자주하는 질문 수정		
15.4 자주하는 질문 삭제	15.4 자주하는 질문 삭제		

2.1.1 기능 보완사항

No.	기능적 요구사항	관리자	일반회원
1	회원 가입시 주소검색	다음카카오 API 를 활용하여 검색	다음카카오 API 를 활용하여 검색
2			

2.2 상세기능 정의

4) 판매의약품등록

No.	요구사항명	요구사항 상세 (처리 process)
3.1	판매의약품등록	의약품정보+의약품효능+의약품성분+의약품이미지 입력
3.2	판매의약품수정	판매의약품수정
3.3	판매의약품삭제	판매의약품삭제
3.4	판매의약품검색	판매의약품검색

5) 결제(주문)관리

No.	요구사항명	요구사항 상세 (처리 process)
4.1	장바구니에 의약품 추가	판매중인 의약품을 장바구니에 추가
4.2	주문 생성	주문하기 버튼 클릭시 새로운 주문과 주문상세를 생성
4.3	주문 취소	주문취소시 주문상세삭제후 주문 삭제
4.4	주문 조회	전체주문, 판매자별주문, 구매자별주문 조회
4.5	결제 확인	결제 API 를 활용하여 pg 사에 연결하여 결제 기능 수행

3. 업무분담(담당 업무)

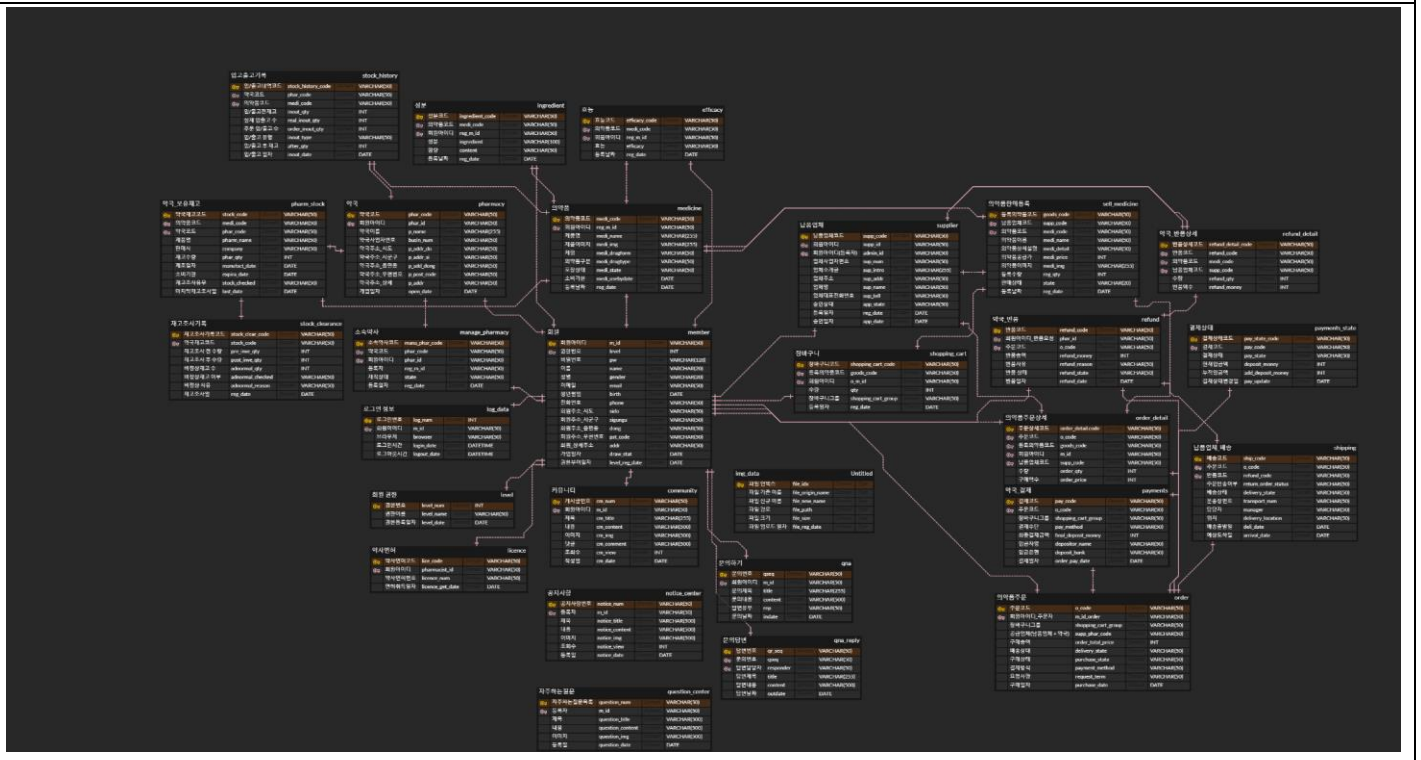
이름	담당 업무	업무 내용
박중연	팀장	회의 진행, 일정 관리, 업무 조율
	회원관리 기능	회원 정보 입력시 회원가입 기능, 각 권한별 권한부여기능, 약국 추가 기능
	로그인관리 기능	Spring Security 6.3 을 이용한 로그인 기능, 로그인 로그 조회 기능
	주문관리 기능	장바구니에 담긴 의약품으로 주문을 생성하는 기능, 주문 취소, 주문 수정 기능
	결제관리 기능	주문완료 이후 포트원 API 를 활용하여 카카오페이 결제 기능
	배송관리 기능	결제 이후 배송 기능,
	Github 코드 관리	팀원별 branch 에서 PR 하는 Deveop, 배포용 master 브랜치 관리
	운영 환경 배포	오라클 클라우드에서 nginx 와 jenkins 를 활용하여 웹 어플리케이션 배포
	네이밍 규칙 작성	특정한 매소드나 클래스, 인터페이스의 네이밍 규칙 작성
	개발문서 작성	전체 개발 문서 작성
	테이블 DB 작성	말은 임무 및 상세테이블 컬럼 작성
	ERD 작성	eXERD 를 이용하여 팀프로젝트 DB 테이블 ERD 작성
	URI 매핑	각 기능 내의 URI 정리
	화면 경로	말은 각 클래스의 화면 경로 확인 및 오류사항 수정
	프로세스흐름도작성	팀 프로젝트의 주제의 이해도를 높이기 위한 프로세스 흐름도 작성
	사이트맵작성	사이트맵 작성
	기능정 의서 작성	기능별 상세한내용을 작성
	데이터사전 작성	데이터 사전 작성
	완료 보고서 작성	완료 보고서 작성
길범진	판매 의약품 관리 기능	판매 의약품을 crud 할 수 있는 기능

약국 재고관리 기능	약국 재고의 crud 할 수 있는 기능, 입출고 기록을 확인하는 기능
약국 재고조사 기능	약국 재고조사 시기에 맞춰 재고조사 및 등록하는 기능
반품 관리 기능	반품을 등록하고 취소하는 기능
배송 관리 기능	반품에 관한 배송을 등록하고 관리하는 기능
네이밍 규칙 작성	특정한 매소드나 클래스, 인터페이스의 네이밍 규칙 작성
ERD 작성	eXERD 를 이용하여 팀프로젝트 DB 테이블 ERD 작성
URI 매핑	각 기능 내의 URI 정리
화면 경로	말은 각 클래스의 화면 경로 확인 및 오류사항 수정
프로세스흐름도작성	팀 프로젝트의 주제의 이해도를 높이기 위한 프로세스 흐름도 작성
사이트맵작성	사이트맵 작성
기능정의서 작성	기능별 상세한내용을 작성
데이터사전 작성	데이터 사전 작성
완료 보고서 작성	완료 보고서 작성

최성식	메인화면	메인화면 index 및 사용자 메인화면 작성
	커뮤니티 기능	커뮤니티 기능 및 댓글 서비스
	1:1 문의 기능	1:1 문의 서비스
	공지사항 기능	공지사항 서비스
	자주하는 질문 기능	자주하는 질문 서비스
	장바구니 기능	장바구니 서비스
	주문하기 기능	주문하기 서비스
	약국 사용자 페이지	공통부분 레이아웃 화면 구현
	Github Readme 작성	Github readme 작성
	fragment 작성	header, aside, footer 작성
	ERD 작성	eXERD 를 이용하여 팀프로젝트 DB 테이블 ERD 작성
	URI 매핑	각 기능 내의 URI 정리
	화면 경로	말은 각 클래스의 화면 경로 확인 및 오류사항 수정
	프로세스흐름도작성	팀 프로젝트의 주제의 이해도를 높이기 위한 프로세스 흐름도 작성
	사이트맵작성	사이트맵 작성
	기능정의서 작성	기능별 상세한내용을 작성
	데이터사전 작성	데이터 사전 작성
완료 보고서 작성	완료 보고서 작성	

4. 설계사양서

4.1 ERD 물리 & 논리 모드

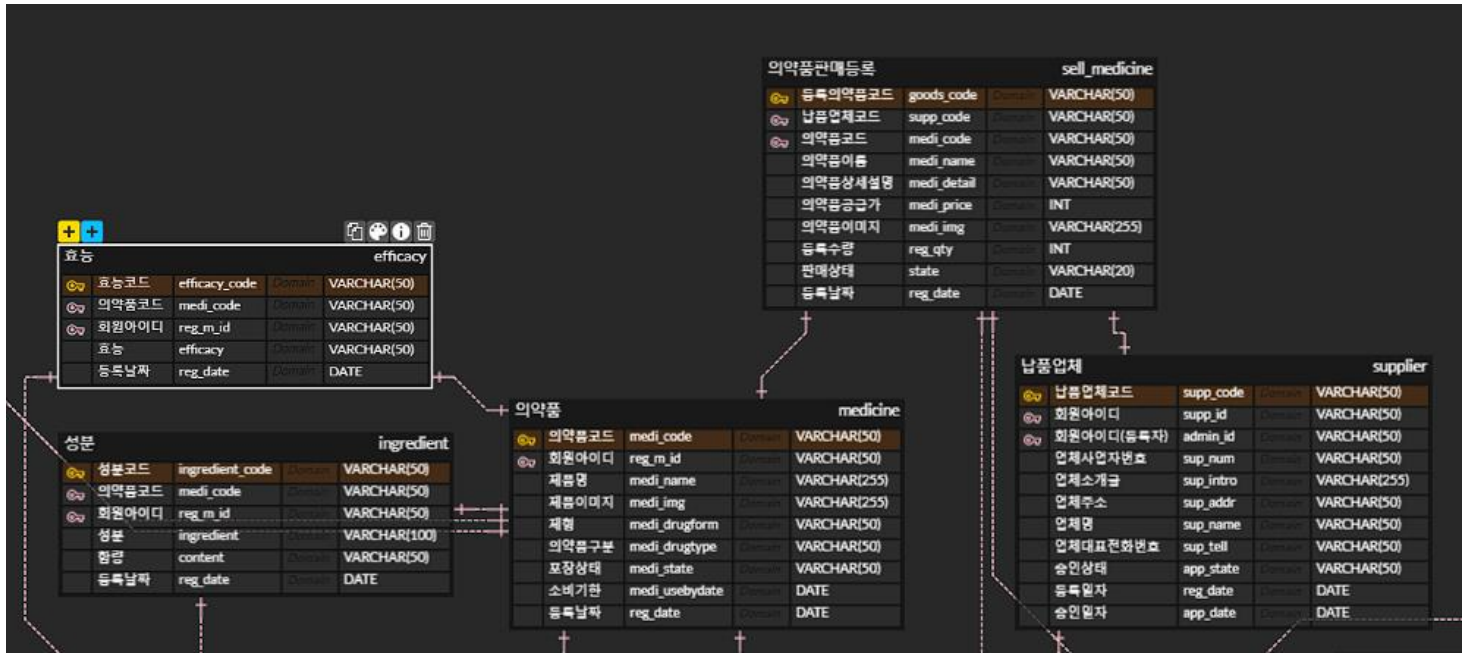


기능별 ERD

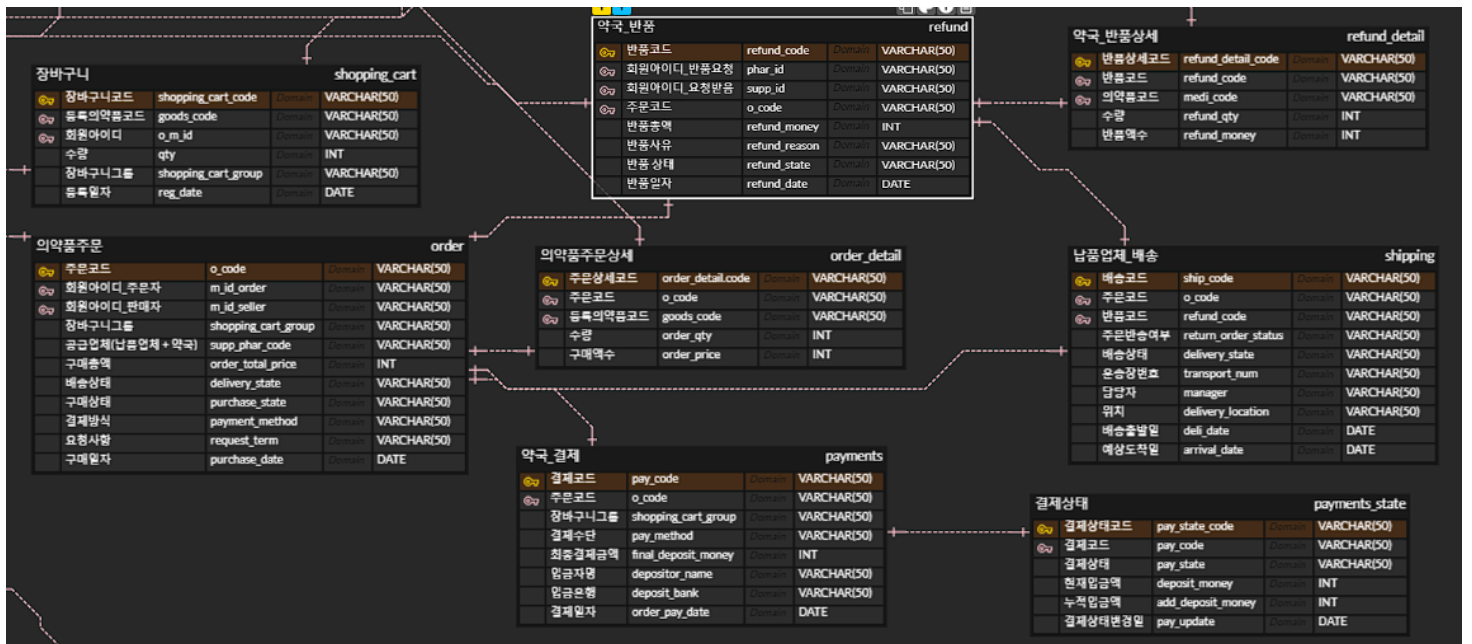
회원관리,로그인 ERD



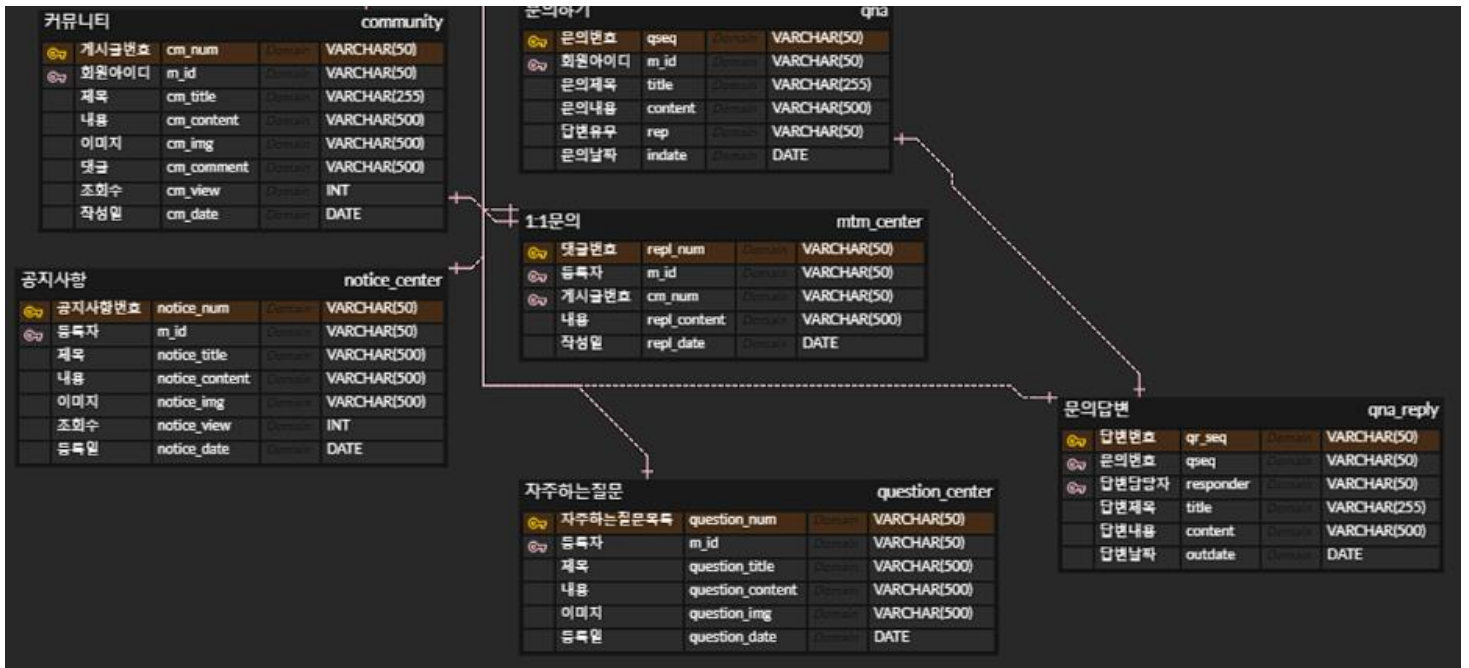
판매 의약품, 의약품 정보 ERD



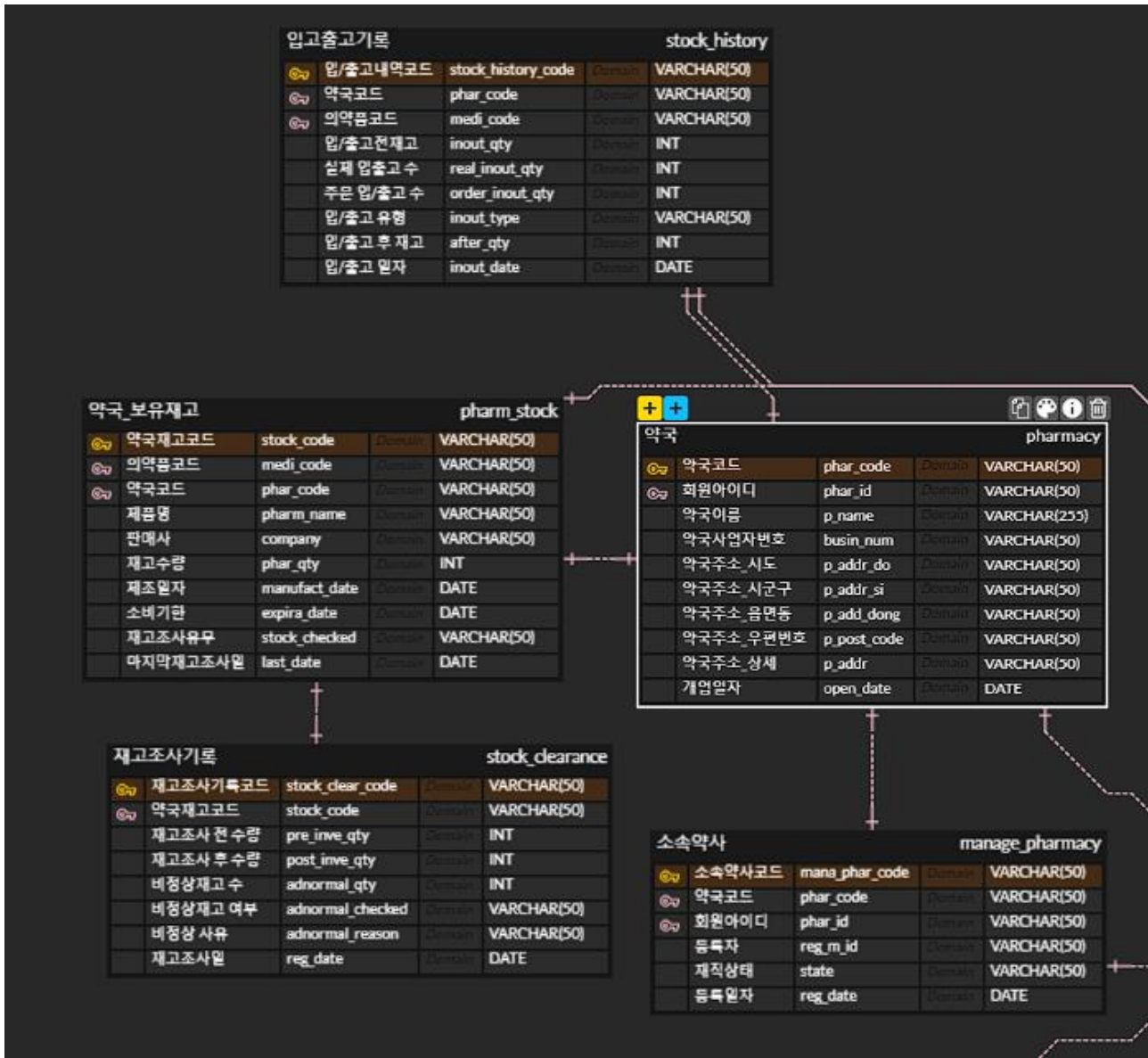
주문, 결제, 배송, 반품 ERD



커뮤니티 ERD



약국 재고 관리, 재고조사 ERD



4.2 테이블 구조

한글테이블	회원	영문테이블	member				
회원아이디(PK)	권한번호(FK)	비밀번호	이름	성별	이메일	생년월일	전화번호
varchar(50)	INT	vachar(128)	vachar(20)	vachar(20)	vachar(50)	date	vachar(50)
m_id	level	pw	name	gender	email	birth	phone
id001	1	pw001	홍01	남	1@1.com	97.10.03	010-0000-0001
id002	2	pw002	홍02	여	2@2.com	97.11.03	010-0000-0002
id003	3	pw003	홍03	남	3@3.com	97.12.03	010-0000-0003
id004	4	pw004	홍04	여	4@4.com	97.12.03	010-0000-0004
id005	5	pw005	홍05	남	5@5.com	97.12.03	010-0000-0005
회원주소_시도	회원주소_시군구	회원주소_읍면동	회원주소_우편번호	회원_상세주소	가입일자	권한부여일자	
vachar(50)	vachar(50)	vachar(50)	vachar(50)	vachar(50)	date	date	
sido	sigungu	dong	pst_code	addr	draw_stat	level_reg_date	
전북특별자치도	전주시 완산구	서신동	55001	빌딩	2023-05-05	2023-05-05	
전북특별자치도	전주시 완산구	서신동	22004	빌딩 301	2024-05-10	2024-05-10	
전북특별자치도	전주시 완산구	서신동	56174	새빌딩301	2024-05-13	2024-05-13	
전북특별자치도	전주시 완산구	서신동	56174	새빌딩302	2024-05-14	2024-05-13	
전북특별자치도	전주시 완산구	서신동	56174	새빌딩303	2024-05-15	2024-05-13	

한글테이블	납품업체	영문테이블	supplier	주요 정규과 예정(회원 테이블)						
납품업체코드(PK)	회원아이디(FK)	등록자(관리자)	업체 사업자번호	업체 소개글	업체주소	업체명	업체대표전화번호	승인 상태	등록일자	승인 일자
VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(255)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	DATE	DATE
supp_code(++)	supp_id	admin_id	sup_num	sup_intro	sup_addr	sup_name	sup_tell	app_state	reg_date	app_date
supp_1	id002	id001	1231	강한 약 강한 기업	서울특별시 강남구	(주)강한약품	02-123-1234	1. 납품업체 승인요청	2024-02-05	2024-02-05
supp_2	id006	id001	123546	웃음을 주는 약품	전주시 완산구	주식회사법진	063-000-0001	2. 관리자 승인	2024-03-01	2024-03-01
supp_3	id007	id001	489453	안녕하세요?	전주시완산구	한국스마트	063-000-0002	3. 관리자 승인 거부	2024-03-07	2024-03-07

한글테이블	약국	영문테이블	pharmacy							
약국코드(PK)	회원아이디(개국약사)(FK)	약국 이름	약국 사업자번호	약국주소_시도	약국주소_시군구	약국주소_읍면동	약국주소_우편번호	약국주소_상세	개업일자	운영상태
VARCAHR(50)	VARCAHR(50)	VARCAHR(255)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	DATE	VARCAHR(50)
phar_code(++)	phar_id	p_name	busin_num	p_addr_do	p_addr_si	p_addr_dong	p_post_code	p_addr	open_date	p_state
phar_code_1	id003	ㅇㅇ약국	1231	서울특별시	강서구	화곡1동	523-546	크리스탈빌딩 305호	2024-04-30	운영중
phar_code_2	id008	ㄹㅇ약국	123546	경기도	의정부시	의정부1동	532-565	의정부빌딩 102호	2024-05-01	운영중
phar_code_3	id009	ㅂㅇ약국	489453	전북특별자치도	전주시 완산구	서신동	356-568	M타워빌딩 203호	2024-05-02	폐업

한글테이블	소속 약사	영문테이블	manage_pharmacy		
소속약사코드(PK)	약국 코드(FK)	회원아이디(FK)	등록자	재직 상태	등록 일자
VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50),null허용	VARCAHR(50)	DATE
mana_phar_code(++)	phar_code	phar_id	reg_m_id	state	reg_date
mana_phar_code_1	phar_code_1	id004	id003	재직중	2024-04-30
mana_phar_code_2	phar_code_1	id010	id003	재직중	2024-05-01
mana_phar_code_3	phar_code_1	id011	id001	퇴사	2024-05-02

한글테이블	의약품	영문테이블	medicine					
의약품코드(PK)	등록자(FK)	제품명	제품이미지	제형	의약품 구분	포장상태	소비기한	등록날짜
VARCHAR(50)	VARCAHR(50)	VARCAHR(255)	VARCAHR(255)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	DATE	DATE
medi_code(++)	reg_m_id	medi_name	medi_img	medi_drugform	medi_drugtype	medi_state	medi_usebydate	reg_date
medi_code_1	id002	그날엔	day.img	알약	전문약품	박스	2024-05-31	2024-05-01
medi_code_2	id003	타이레놀	tylenol.img	알약	일반약품	개별	2025-04-28	2024-05-02
medi_code_3	id003	모가프렌	moga.img	가루형	전문약품	박스	2024-12-31	2024-05-03

한글테이블	의약품판매등록	영문테이블	sell_medicine						
등록의약품코드(PK)	납품업체코드(FK)	의약품코드(FK)	의약품이름	의약품상세설명	의약품 공급가	의약품 이미지	등록수량	판매상태	등록날짜
VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	INT	VARCAHR(255)	INT	VARCHAR(20)	DATE
goods_code(++)	supp_code	medi_code	medi_name	medi_detail	medi_price	medi_img	reg_qty	state	reg_date
goods_1	supp_1	medi_code_1	그날엔	두통 및 해열	3000	img1	4000	판매중	2024-05-01
goods_2	supp_1	medi_code_2	타이레놀	두통 및 해열	5000	img2	0	매진	2024-05-02
goods_3	supp_2	medi_code_3	모가프텐	해열제	3000	img3	9000	판매중단	2024-05-03

한글테이블	의약품주문	영문테이블	order						
주문코드(PK)(++)	회원아이디_주문자(FK)	장바구니그룹	구매 총액	배송 상태	구매 상태	결제 방식	요청 사항	거래타입	구매 일자
VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	INT	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	DATE
o_code(++)	m_id_order	shopping_cart_group	order_total_price	delivery_state	purchase_state	payment_method	request_term	order_type	purchase_date
o_code_1	id003	group_1	280000	배송대기	결제준비중	카드	고온주의	주문	23-04-09
o_code_2	id003	group_1	30000	배송대기	결제준비중	카드	저온배송	주문	23-04-09
o_code_3	id004	group_2	88750	배송완료	결제완료	계좌이체	저온배송	긴급의약품요청	24-02-04

한글테이블	납품업체_배송	영문테이블	shipping						
배송코드(PK)(++)	주문코드(FK)	반송코드(FK)	주문반송여부	배송상태	운송장번호	담당자	위치	배송 출발일	예상도착일
VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	DATE	DATE
ship_code	o_code	refund_code	return_order_state	delivery_state	transport_num	manager	delivery_location	deli_date	arrival_date
ship_1	o_code_1	null	주문	배송준비중	6525151544	홍길동	육전HUB	24-05-25	24-05-25
ship_2	o_code_2	null	주문	배송중	6525156522	이길동	연계SUB	24-02-27	24-05-27
ship_3	null	refund_code_1	반송	배송완료	6525198524	삼길동	육전HUB	24-05-28	24-05-28
ship_4	null	refund_code_2	반송	배송중	6525173515	사길동	OO집하장	24-05-29	24-05-29

한글테이블	약국_보유재고	영문테이블	pharm_stock	약국에서 보유중인 상품들이 DB에 쌓이는 테이블					
약국 재고코드(PK)	의약품 코드(FK)	약국코드(FK)	제품명	판매사	재고 수량	제조일자	소비기한	재고조사유무	마지막재고조사일
VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	INT	DATE	DATE	VARCAHR(50)	DATE
stock_code(++)	medi_code	phar_code(++)	pharm_name	company	phar_qty	manufact_date	expira_date	stock_checked	last_date(NULL)
stock_1	medi_code_1	phar_code_1	그날엔	경동제약	1020 --> 1000	2023-03-03	2025-07-03	무	2024-05-16
stock_2	medi_code_2	phar_code_2	타이레놀	동국제약	86	2024-02-01	2027-02-03	무	2027-02-03
stock_3	medi_code_3	phar_code_3	트라켄세미서방정	종근당	424	2024-03-07	2024-08-03	유	2024-08-03

한글테이블	약국_반품	영문테이블	refund				
반품코드(PK)(++)	회원아이디(반품요청)(FK)	주문코드(FK)	반품 총액	반품 사유	반품 상태	반품 일자	
VARCAHR(50)	VARCAHR(50)	VARCAHR(50)	INT	VARCAHR(50)	VARCAHR(50)	DATE	
refund_code(++)	phar_id	o_code	refund_money	refund_reason	refund_state	refund_date	
refund_code_1	id003	o_code_1	280000	구매실수	반품완료	2024-05-01	
refund_code_2	id003	o_code_1	30000	소비기한 초과	반품확인	2024-05-02	

4.3 네이밍 규칙

패키지 구조	<ul style="list-style-type: none"> ▶ Level 0 : 배포할 프로젝트 명을 뒤에서 부터 명칭 ▶ Level 1 : Web Application 구성 요소로 구분 (ex: controller, service, mapper, domain, config, interceptor) ▶ Level 2 : 실구현체 이름 + 접미사: 상위 패키지 네이밍 (파스칼 표기법) (ex: MemberController, CourseService, OperationMapper) 		
	Level 0	Level 1	Level 2

		controller	MemberController.java
		service	MemberService.java
		mapper	MemberMapper.java
		dto	Member.java
		config	WebConfig.java
		interceptor	LoginInterceptor.java
명명규칙	▶ 공통		
	모든 이름은 a-z, A-Z, 0-9 의 글자로 조합한다.		
	프로그램 내에서 사용하는 모든 이름은 64 자를 넘지 않는 것으로 한다.		
	정확한 의미 전달을 위해 약어의 사용은 피한다.		
	특수기호는 사용하지 않는 것을 원칙으로 한다.		
	약어 사용이 의미 전달에 도움이 되는 경우에는 표준용어사전에 포함되어있는 약속된 약어를 사용할 수 있으나, 그렇지 않은 경우에는 단어의 spelling 을 모두 적는 것으로 한다		
	▶ 파일명		
	Java	대소문자를 사용하며 단어의 시작부분은 대문자로 표기한다. (파스칼 표기법)	
		단어는 표준용어사전을 따른다.	
		명사 또는 동사 + 명사의 혼합된 형태로 이름을 구성한다.	
		첫 번째 문자는 항상 대문자를 사용한다.	
	mapper	클래스 유형에 따라 다음 표를 참고하여 구성한다.	
		Dao (java 파일)의 명명 규칙을 따른다. (ex: Dao MemberMapper -> mapper MemberMapper)	
	html	파일의 확장자는 .xml 로 한다.	
		단어의 시작부분은 소문자로 표기하고 혼합된 형태로 이름을 구성할 때에는 각 단어의 첫 글자는 대문자로 작성한다. (카멜표기법)	
css	파일의 확장자는 .html 로 한다.		
	변수명의 시작부분을 소문자로 표기하고 각 단어의 구분자는 ' - ' 으로 표기하여 작성한다.		
JavaScript	파일의 확장자는 .css 로 한다.		
	변수명의 시작부분을 소문자로 표기하고 각 단어의 구분자는 ' - ' 으로 표기하여 작성한다.		
	파일의 확장자는 .js 로 한다. 오픈소스 자바스크립트의 경우, 최초 배포된 이름 그대로 유지한다.		
▶ method			

동사 혹은 동사 + 명사로 작성하며, 혼합된 형태로 이름을 구성할 때에는 각 단어의 첫 글자는 대문자로 작성한다. (카멜표기법)		
상황에 따라 다음 표를 참고하여 일반적인 Prefix 를 적용하며, 나머지는 필요에 따른 적절한 이름을 사용한다.		
기능유형	대표용어	
get	조회	
add	등록	
modify	수정	
delete	삭제	
apply	신청	
recruit	모집	
search	검색	
manage	관리	
enter	입점	
result	결과	
▶ 변수명		
대소문자를 사용하여 변수명의 시작부분을 소문자로 표기한다.		
명사로 작성하며 각 단어의 첫 글자에는 대문자로 표기한다.		
변수명에는 기호나 특수문자를 사용하지 않는다.		
변수명은 의미 있는 단어를 사용하도록 한다.(표준 용어사전 참고)		
변수에 해당하는 컬럼이 테이블에 존재하는 경우에는 컬럼명을 기준으로 camel 표기법을 준수하여 사용한다		
변수명은 중복되는 이름이 나오지 않도록 하여 일련번호를 붙여 사용하는 경우가 없도록 한다.		
상수명은 공통코드를 제외한 변수명 전체를 대문자로 하고, 단어 연결에는 '_'를 사용한다		
예시		
java	변수	int orderCnt; / double deliveryAmt; / String userName; / StringBuffer sqlStr;
	상수	static final int WIDTH = 4; / static final long DEFAULT_EXPIRES_TIME = TimeUnit.MINUTES.toSeconds(30);

		사용자 정의 타입	private MemberService memberService;
	javaScript	변수	var orderCnt; / var deliveryAmt; / var userName;
		상수	const WIDTH = 4; / const DEFAULT_PER_PAGE = 10;

4.4 디렉토리 구조

index.html		메인화면		
pharmacy	pharmMain.html		약사 메인화면	
	fragments	header.html	헤더 조각	
		footer.html	푸터 조각	
	layout	default.html	기본 레이아웃 파일	
	member	memberOwnerInsert.html	개국약사 회원가입	
		memberStaffInsert.html	관리약사 회원가입	
	mypage	myDelivery.html	배송조회	
		myOrderDetailView.html	주문상세	
		myOrderList.html	주문목록	

		myPage.html	마이페이지 메인		
	order	cart.html	장바구니		
		orderMedicine.html	주문하기		
		orderMedicineDetailView.html	주문하기 상세		
		paymentMethod.html	결제수단 선택		
		boardList.html	커뮤니티 조회		
		faq.html	자주하는 질문 조회		
		board	notice.html	공지사항 조회	
			qna.html	1:1 문의등록	
			qnaList.html	1:1 문의조회	
	pharmstock	checkMedi.html	재고조사		
		myStockHistory.html	입출고기록		
		myStockInsert.html	재고등록		
		myStockRelease.html	출고		
		myStockSearchList.html	재고관리		

		stockDetailView.html	재고상세	
	refund	myRefundDetailView.html	환불상세	
		myRefundInsert.html	환불생성	
		myRefundSearchList.html	환불조회	
supplier	supplierMain.html		납품업체 메인화면	
	board	boardList.html	커뮤니티 조회	
		faq.html	자주하는 질문	
		notice.html	공지사항	
		qna.html	1:1 문의등록	
		qnaList.html	1:1 문의조회	
	layout	default.html	기본 레이아웃 파일	
	fragments	aside.html	사이드메뉴 조각	
		footer.html	푸터 조각	
		header.html	헤더 조각	
medicine	myMedicineDelete.html	내 의약품 삭제		

		myMedicineDetailView.html	내 의약품 상세		
		myMedicineInsert.html	내 의약품 등록		
		myMedicineModify.html	내 의약품 수정		
		myMedicineSearchList.html	내 의약품 조회		
	order	myOrderDelete.html	내 주문 삭제		
		myOrderDetailView.html	내 주문 상세		
		myOrderSearchList.html	내 주문 조회		
	refund	myRefundDetailView.html	내 환불 상세		
		myRefundSearchList.html	내 환불 조회		
	shipping	myRefundShipList.html	내 환불배송 조회		
		myShipSearchList.html	내 배송 조회		
	admin	adminMain.html		관리자 메인화면	
		fragments	aside.html	사이드메뉴 조각	
			footer.html	푸터 조각	
header.html			헤더 조각		

layout	default.html	기본 레이아웃 파일	
loginhistory	loginHistory.html	로그인 기록 조회	
medicine	medicineDelete.html	의약품 삭제	
	medicineDetailView.html	의약품 상세	
	medicineInsert.html	의약품 등록	
	medicineModify.html	의약품 수정	
	medicineSearchList.html	의약품 조회	
notice	noticeAdd.html	공지 등록	
	noticeList.html]	공지 조회	
order	addShoppingCart.html	장바구니 생성	
	orderDelete.html	주문 삭제	
	orderDetailView.html	주문 상세	
	orderSearchList.html	주문 목록	
pharmacy	memberInsert.html	회원 생성	

	memberSearchList.html	회원 목록	
	pharmacyInsert.html	약국 등록	
	pharmacySearchList.html	약국 목록	
pharmStock	stockDelete.html	재고 삭제	
	stockHistoryDelete.html	입출고기록 삭제	
	stockHistoryInsert.html	입출고기록 등록	
	stockHistoryList.html	입출고기록 조회	
	stockHistoryModify.html	입출고기록 수정	
	stockInsert.html	재고 등록	
	stockModify.html	재고 수정	
	stockSearchList.html	재고 조회	
qna	qnaAdd.html	문의 등록	
	qnaList.html	문의 목록	
questionCenter	faqAdd.html	자주하는 질문 등록	
	faqList.html	자주하는 질문 목록	

	refund	refundDelete.html'	환불 삭제	
		refundDetailView.html	환불 상세	
		refundRequest.html	환불 생성	
		refundSearchList.html	환불 목록	
	shipping	RefundShipList.html	환불 배송 목록	
		ShipSearchList.html	배송 목록	
	supplier	supplierInsert.html	납품업체 생성	
		supplierSearchList.html	납품업체 조회	

4.5 패키지 구조

no	패키지명	클래스 또는	메소드 명	설명
----	------	--------	-------	----

		인터페이스 명		
admin				
1	team01.yaksutor.admin.controller	AdLoginController	deleteHistory	로그인 이력 삭제
2			loginHistory	로그인 이력 조회
3		AdMedicineController	medicineInsert	의약품 등록 페이지 맵핑
4			medicineSearchList	의약품 목록 페이지 맵핑
5			medicineModify	의약품 수정 페이지 맵핑
6			medicineDelete	의약품 삭제 페이지 맵핑
7			medicineDetailView	의약품 상세 페이지 맵핑
8			checkLevel	판매자 아이디 확인 ajax 요청
9			medicineInsert	의약품 및 판매 의약품 등록 ajax 요청
10			medicineModify	의약품 수정 ajax 요청
11			medicineDelete	의약품 삭제 ajax 요청 기능-판매 의약품 판매상태 수정

12	AdMemberController	deletePharmacy	약국 삭제
13		modifyPharmacy	약국 수정
14		modifyMember	회원 수정
15		deleteMember	회원 삭제
16		pharmacySearchList	약국 리스트 조회
17		memberSearchList	회원 리스트 조회
18		AdPharmacyInsert	관리자화면 약국 추가
19		AdMemberInsert	관리자화면 회원 추가
20		idCheck	DB 에 저장된 회원의 ID 체크
21		pharmacyInsert	새로운 약국 추가
22		memberInsert	관리자 화면 새로운 회원 추가
23		isStaff	기존에 등록된 관리약사 권한의 회원인지 조회
24		AdminController	adminMain

25	AdRefundController	refundRequest	반품 신청 페이지 맵핑
26		refundSearchList	반품 목록 페이지 맵핑
27		refundDetailView	반품 상세 페이지 맵핑
28		myRefundShip	반품 배송 등록 ajax 요청
29		updateForClear	반품 완료 ajax 요청
30	AdShippingController	orderShip	주문 배송 목록 페이지 맵핑
31		refundShip	반품 배송 목록 맵핑
32		cancelRefund	반품 취소 ajax 요청
33	AdStockController	stockInsert	재고 등록 페이지 맵핑
34		stockSearchList	재고 목록 페이지 맵핑
35		stockModify	재고 수정 페이지 맵핑
36		stockHistoryList	입출고 기록 페이지 맵핑
37		insertStock	재고 추가 ajax 요청
38		modifyStock	재고 수정 ajax 요청

39		AdSupplierController	supplierModify	관리자페이지 납품업체 권한 회원 정보 수정
40	supplierDelete		관리자페이지 납품업체 권한 회원 정보 삭제	
41	supplierSearchList		관리자페이지 납품업체 권한 회원 정보 조회	
42	supplierInsert		관리자페이지 납품업체 권한 회원 추가	
44		OrderController	addShoppingCart	판매중인 의약품을 장바구니에 추가
45	orderSearchList		관리자화면에서 전체 주문 목록 조회	
46	deleteOrder		주문목록에서 선택한 주문 삭제	
47	orderDelete		주문목록에서 특정주문 삭제	
48	orderDetailView		특정 주문에 대한 주문 상세 조회	
49	team01.yaksutor.admin.service	AdLoginHistoryService	getLoginHistory	로그인 내역 조회 서비스 구현
50		AdMedicineService	getMedicineList	의약품 리스트 출력

51		insertMedicine	의약품 등록 트랜잭션
52		getMedicalInfo	수정할 의약품의 정보 검색 및 리턴
53		modifyMedicine	의약품 수정 트랜잭션
54		deleteMedicine	의약품 삭제 트랜잭션
55	AdMemberService	getPharmacyInfo	전체 약국 정보 조회
56		getMemberInfo	전체 회원 정보 조회
57		insertOwnerMember	개국약사 회원 추가
58		insertStaffMember	관리약사 회원 추가
59		staffToOwnerMember	기존 관리약사에서 개국약사로 회원권한 전환
60		memberDelete	회원 탈퇴
61		memberModify	회원의 정보 수정
62		pharmacyModify	약국 정보 수정
63		pharmacyDelete	약국 정보 삭제

64		AdOrderService	getOrderList	관리자 화면에서 전체 주문 목록 조회 서비스
65			deleteOrderByCode	관리자 화면에서 특정 주문 삭제
66			getOrderDetailListByCode	주문코드로 주문상세내역 조회
67		AdShippingService	cancelRefund	반품 취소 트랜잭션
68		AdStockService	insertStock	재고 등록 트랜잭션
69			modifyStock	재고 수정 트랜잭션
70		AdSupplierService	supplierInsert	납품업체 회원 추가
71	team01.yaksutor.admin.mapper	AdLoginHistoryMapper	insertLoginHistory	로그인 내역 저장
72			updateLogoutDate	로그아웃 시각 업데이트
73			findLatestLoginHistory	가장 마지막 로그인한 로그인정보 조회
74			getLoginHistory	로그인 내역 조회
75			getLoginHistoryCnt	로그에 저장된 로그인내역의 갯수 조회

76	AdMedicineMapper	checkLevel	판매자 아이디 확인
77		getMedicineList	판매 의약품 목록 검색
78		getMedicineListCnt	판매 의약품 수 검색
79		insertMedicine	의약품 등록
80		insertImg	이미지 등록
81		insertIngredient	의약품 성분 등록
82		insertEfficacy	의약품 효능 등록
83		insertSellMedicine	판매 의약품 등록
84		getSuppCode	판매자 아이디로 납품업체 코드 검색
85		getMedicineInfo	상품 코드로 의약품 정보 검색
86		getSuppName	상품 코드로 판매자 이름 검색
87	getSellMediInfo	상품 코드로 판매 의약품 정보 검색	

88		getAdIngredient	의약품 코드로 의약품 성분 리스트 검색
89		getAdEfficacy	의약품 코드로 의약품 효능 리스트 검색
90		updateMedicine	의약품 정보 수정
91		deleteIngredient	성분 삭제
92		deleteEfficacy	효능 삭제
93		updateSellMedicine	판매 의약품 정보 수정
94		getMedicineImg	의약품 코드로 이미지 검색
95		getMediCode	상품코드로 의약품 코드 검색
96		getMedicineName	의약품 코드로 의약품 이름 검색
97		deleteSellMedicine	판매 의약품 상태 '삭제'로 수정
98	AdMemberMapper	supplierInsert	납품업체 회원 추가
99		memberInsert	개국약사,관리약사 회원 추가

10 0		licenseInsert	약사 면허 정보 추가
10 1		pharmacyInsert	약국 정보 추가
10 2		managePharmacyInsert	관리약사 정보 추가
10 3		idCheck	ID 중복체크
10 4		getPharCode	개국약사 아이디로 약국코드 조회
10 5		deleteManagePharmacyById	관리약사에서 개국약사로 회원전환
10 6		updateMemberLevelNumById	회원의 아이디로 권한 업데이트
10 7		getPharmacyList	약국전체정보조회
10 8		getSupplierInfo	납품업체전체정보조회
10 9		deleteMemberById	회원 탈퇴
11		modifyMember	회원 수정

0			
11 1		modifyPharmacy	약국수정
11 2		deletePharmacy	약국삭제
11 3		deleteSupplier	납품업체삭제
11 4		modifySupplier	납품업체수정
11 5		getPharmacyById	회원 아이디로 약국 정보 조회
11 6	AdminMapper	medicineInsert	의약품 등록
11 7		autoIncrementMediCode	의약품 코드 자동증가
11 8	AdOrderMapper	getOrderList	주문 전체 내역 조회
11 9		deleteOrderDetailById	회원 아이디로 주문 상세내역 조회
12 0		getOrderDetailsByCode	주문코드로 주문상세조회

12 1	AdRefundMapper	getRefundList	반품 목록 검색
12 2		getRefundDetailList	반품 상세 정보 검색
12 3		getShippingDetail	배송 상세 검색
12 4		getRefundTotal	총 반품 금액 검색
12 5		getPaymentMethod	주문 시 결제방식 검색
12 6		insertShipping	반품 배송 등록
12 7		updateShipping	배송 상태 수정
12 8		updateRefund	반품 상태 수정
12 9	AdShippingMapper	getOrderShipList	주문 배송 목록
13 0		getRefundShipList	반품 배송 목록
13		deleteShipping	shipping 삭제

1			
13 2		deleteRefundDetail	refundDetail 삭제
13 3		deleteRefundDetail	refund 삭제
13 4		updateCancleRefund	order 상태 반품 -> 결제완료
13 5		getOCodeByRefundCode	반품 코드로 주문 코드 검색
13 6	AdStockMapper	getPharmStockList	재고 목록 조회
13 7		getPharIdByPharCode	약국코드로 약사아이디 조회
13 8		getPharNameByPharCode	약국코드로 약국이름 조회
13 9		getPharmStockByStockCode	재고코드로 약국 재고 조회
14 0		updateMedicine	의약품 정보 수정
14 1		deleteIngredient	성분 삭제

14 2			deleteEfficacy	효능 삭제
14 3			updateStock	재고 수정
14 4			getStockHistory	재고 입출고 기록 목록 조회
14 5			addStockHistory	재고 입출고 기록 추가

supplier

14 6	team01.yaksutor.supplier.controller	SuMedicineController	myMedicineInsert	판매 의약품 등록 페이지 맵핑
14 7			myMedicineSearchList	납품업자 별 판매 의약품 목록 페이지 맵핑
14 8			myMedicineModify	내 의약품 수정 페이지 맵핑
14 9			myMedicineDelete	내 의약품 판매상태 -> '삭제'로 수정
15 0			myMedicineDetailView	내 판매 의약품 상세 페이지 맵핑
15 1			insertMedicine	내 판매 의약품 등록 ajax 요청

15 2		modifyMedicine	내 판매 의약품 수정 ajax 요청
15 3		deleteMedicine	내 의약품 상태 -> '삭제'로 수정 ajax 요청
15 4	SuOrderController	deleteOrder	나의 주문 삭제
15 5		myOrderSearchList	나에게 들어온 주문 조회
15 6		myOrderDetailView	특정 주문에 대한 주문상세내역 조회
15 7		myOrderDelete	나에게 들어온 주문 삭제
15 8		SupplierController	supplierMain
15 9	SuRefundController	myRefundSearchList	내 반품 목록 페이지 맵핑
16 0		myRefundDetailView	내 반품 상세 페이지 맵핑
16 1		myRefundShip	내 반품 배송 등록 ajax 요청

16 2			updateForClear	내 반품 상태 -> '반품완료'로 변경 ajax 요청	
16 3		SuShippingController	shipSearchList	내 주문 배송 목록 페이지 맵핑	
16 4			refundShipList	내 반품 배송 목록 페이지 맵핑	
16 5	team01.yaksutor.supplier.service	SuMedicineService	getSellMediList	내 판매 의약품 목록 조회	
16 6				insertMedicine	내 판매 의약품 등록 트렌젝션
16 7				modifyMedicine	내 판매 의약품 수정 트렌젝션
16 8			SuOrderService	getOrderListById	회원의 ID 로 주문리스트 조회
16 9				deleteOrderByoCode	주문코드로 주문 삭제
17 0				getOrderDetailByOderCode	주문코드로 주문상세 정보 조회
17 1	team01.yaksutor.supplier.mapper	SuMedicineMapper	getListSellMedicine	판매 의약품 목록 조회	
17				getSuppCode	납품업체 코드 조회

2			
17 3		insertMedicine	의약품 정보 등록
17 4		insertImg	이미지 등록
17 5		insertIngredient	성분 등록
17 6		insertEfficacy	효능 등록
17 7		insertSellMedicine	판매 의약품 등록
17 8		getMedicineInfo	의약품 정보 조회
17 9		getSuppName	납품업체 이름 조회
18 0		getSellMedicineInfo	판매 의약품 정보 조회
18 1		getIngrList	성분 목록 조회
18 2		getEffiList	효능 목록 조회

18 3		getMediCode	의약품 코드 조회
18 4		updateMedicine	의약품 정보 수정
18 5		deleteIngredient	성분 삭제
18 6		deleteEfficacy	효능 삭제
18 7		updateSellMedicine	판매 의약품 수정
18 8		deleteSellMedicine	판매 의약품 삭제
18 9	SuOrderMapper	getOrderListById	판매자 회원의 주문리스트 조회
19 0		deleteOrderDetailById	판매자 ID 로 주문상세 삭제
19 1		getOrderDetailsBySellerIdAndOrderCode	판매자 ID 와 주문코드로 주문과 주문상세를 조회
19 2	SuRefundMapper	getRefundTotal	반품 총액 조회
19		getPaymentMethod	결제 방법 조회

3			
19 4		getRefundDetailByRefundCode	반품 코드로 반품 상세 조회
19 5		getSuppCodeById	회원 아이디로 납품업체 코드 조회
19 6		getRefundListBySuppCode	납품업체 코드로 반품 목록 조회
19 7		getShippingDetail	배송 상세 조회
19 8		insertShipping	배송 등록
19 9		updateShipping	배송 수정
20 0		updateRefund	반품 수정
20 1	SuShippingMapper	getSuppCode	납품업체 코드 조회
20 2		getShippingList	배송 목록 조회
20 3		getRefundShippingList	반품 배송 목록 조회

pharmacy

204	team01.yaksutor.pharmacy.controller	PharmacyController	pharmMain	약사 메인화면, 판매의약품 조회
205			contact	고객센터 조회
206			qna	1:1 문의조회
207			qnaList	1:1 문의목록
208			PostMapping	1:1 문의등록
209			deleteQna	1:1 문의삭제
210			myPage	마이페이지 조회
211			myOderList	나의 주문 조회
212			myDelivery	나의 배송목록조회
216			searchDeli	배송 조회

21 7	PhBoardController	getBoardList	커뮤니티 + 댓글 조회 (사용자)
21 8		getBoardList2	커뮤니티 + 댓글 조회 (관리자)
21 9		getBoardList3	커뮤니티 + 댓글 조회 (납품자)
22 0		addBoard	커뮤니티 등록 (사용자)
22 1		addBoard3	커뮤니티 등록 (납품자)
22 2		addBoard2	커뮤니티 등록 (관리자)
22 4		modifyBoard2	커뮤니티 수정 (사용자)
22 5		modifyBoard3	커뮤니티 수정 (납품자)
22 6		modifyBoard	커뮤니티 수정 (관리자)
22 7		deleteBoard2	커뮤니티 삭제 (사용자)
22		deleteBoard3	커뮤니티 삭제 (납품자)

8			
22 9		deleteBoard	커뮤니티 삭제 (관리자)
23 0		addRepl	커뮤니티 댓글 등록 (사용자)
23 1		addRepl2	커뮤니티 댓글 등록 (납품자)
23 2		addRepl3	커뮤니티 댓글 등록 (관리자)
23 3		modifyRepl	커뮤니티 댓글 수정 (관리자)
23 4		deleteRepl	커뮤니티 댓글 삭제 (사용자)
23 5		deleteRepl2	커뮤니티 댓글 삭제 (납품자)
23 6		deleteRepl3	커뮤니티 댓글 삭제 (관리자)
23 7		getQuestionCenterList	자주하는 질문 조회 (사용자)
23 8		getQuestionCenterList3	자주하는 질문 조회 (납품자)

23 9		getQuestionCenterList2	자주하는 질문 조회 (관리자)
24 0		addQuestionCenter	자주하는 질문 등록 (관리자)
24 2		modifyQuestionCenter	자주하는 질문 수정 (관리자)
24 3		deleteQuestionCenter	자주하는 질문 삭제 (관리자)
24 4		getNoticeList	공지사항 조회 (사용자)
24 5		getNoticeList3	공지사항 조회 (납품자)
24 6		getNoticeList2	공지사항 조회 (관리자)
24 7		addNotice	공지사항 추가 (관리자)
24 9		modifyNotice	공지사항 수정 (관리자)
25 0		deleteNotice	공지사항 삭제 (관리자)
25		qnaList	1:1 문의 조회 (사용자)

1			
25 2		qnaList2	1:1 문의 조회 (관리자)
25 3		addQna	1:1 문의 추가 (관리자)
25 4		qnaAdd	1:1 문의 추가 (사용자)
25 5		qnaAdd2	1:1 문의 추가 (납품자)
25 6		deleteQna3	1:1 문의 삭제 (사용자)
25 7		deleteQna	1:1 문의 삭제 (관리자)
25 8		deleteQna2	1:1 문의 삭제 (납품자)
25 9		addQnaReply	1:1 문의 답변 추가
26 0		deleteQnaReply	1:1 문의 답변 삭제
26 1	PhMedicineController	getSellMedicineList	판매의약품 조회

26 2		orderMedicineDetailView	판매의약품 상세
26 3	PhMemberController	insertStaff	관리약사의 회원가입 처리
26 4		insertOwner	개국약사의 회원가입 처리
26 5		memberStaffInsert	관리약사 가입페이지로 이동
26 6		memberOwnerInsert	개국약사 가입페이지로 이동
26 7	PhMyPageController	myAccountModify	나의 회원 정보 수정
26 8	PhOrderController	deleteOrderDetail	주문상세코드를 이용해 주문상세를 삭제하는 기능
26 9		OrderPay	장바구니로부터 주문을 생성
27 0		myOrderDetailView	주문상세 페이지로 이동
27 1		paymentMethod	결제 방법 선택

27 2	PhPaymentController	paymentComplete	결제 완료후 로직 수행
27 3	PhPharmStockController	pharmStrock	재고조사 페이지 맵핑
27 4		myStockInsert	재고 등록의 GetMapping
27 5		myStockSearchList	내 재고 목록 페이지 맵핑
27 6		myStockRelease	내 재고 출고 페이지 맵핑
27 7		myStockHistory	내 재고 입출고 기록 페이지 맵핑
27 8		stockDetailView	내 재고 상세페이지 맵핑
27 9		myStockInsert	재고 등록 ajax 요청을 받는 메소드
28 0		myStockRelease	내 재고 출고 submit 맵핑
28 1		checkMedi	내 재고 조사 ajax 요청
28		PhRefundController	myRefundInsert

2				맵핑
28 3			myRefundSearchList	내 반품 목록 페이지 맵핑
28 4			myRefundDetailView	내 반품 상세 페이지 맵핑
28 5			myRefundInsert	내 반품 등록 ajax 요청
28 6			cancelRefund	내 반품 취소 ajax 요청
28 7		PhShoppingCartCon troller	getShoppingCartList	조회
28 8			addShoppingCart	생성 (약사)
28 9			addShoppingCart2	생성 (관리자)
29 0			deleteShoppingCart	삭제 (사용자)
29 1			deleteShoppingCart2	삭제 (관리자)
29 2	team01.yaksutor.pharmacy.service	PhBoardService	getBoardList	커뮤니티 글 조회

29 3		addBoard	커뮤니티 글 추가
29 4		modifyBoard	커뮤니티 글 수정
29 5		deleteBoard	커뮤니티 글 삭제
29 6		getReplList	커뮤니티 댓글 조회
29 7		addRepl	커뮤니티 댓글 추가
29 8		modifyRepl	커뮤니티 댓글 수정
29 9		deleteRepl	커뮤니티 댓글 삭제
30 0		getQuestionCenterList	자주하는 질문 조회
30 1		addQuestionCenter	자주하는 질문 추가
30 2		modifyQuestionCenter	자주하는 질문 수정
30		deleteQuestionCenter	자주하는 질문 삭제

3			
30 4		getNoticeList	공지사항 조회
30 5		addNotice	공지사항 추가
30 6		modifyNotice	공지사항 수정
30 7		deleteNotice	공지사항 삭제
30 8		getQnaList	1:1 문의 조회
30 9		getQnaReplyList	1:1 문의 답변 조회
31 0		addQna	1:1 문의 추가
31 1		addQnaReply	1:1 문의 답변 추가
31 2		deleteQna	1:1 문의 삭제
31 3		deleteQnaReply	1:1 문의 답변 삭제

31 4	PhMedicineService	getSellMedicineList	판매의약품 조회
31 5	PhMemberService	insertStaffMember	관리약사 권한의 회원 추가
31 6		insertOwnerMember	개국약사 권한의 회원 추가
31 7		getPharmacyById	회원 ID 로 회원이 소속된 약국 정보 조회
31 8	phMyPageService	getMeberInfoById	회원 ID 로 회원정보 조회
31 9		modifyMyAccount	나의 계정 정보 수정
32 0	PhOrderService	submitOrder	주문 요청
32 1		getOrderListById	회원 Id 로 주문목록 조회
32 2		getOrderDetailListByOCode	사용자 화면 주문코드로 주문상세 조회
32 3		deleteOrderDetail	사용자 화면 주문상세 삭제
32		updateOrderStatus	사용자 화면 주문상태

4			업데이트
32 5	PhRefundService	insertRefund	반품 등록 트랜잭션
32 6		cancelRefund	반품 취소 트랜잭션
32 7	PhShoppingCartService	getShoppingCartList	장바구니 조회
32 8		addShoppingCart	장바구니 추가
32 9		deleteShoppingCart	장바구니 삭제
33 0		getCartItemsByMemberId	특정아이디로 장바구니 조회
33 1	PhStockService	stockInsert	medicine, PharmStock, Ingredient, Efficacy 에 값을 담아서 쿼리문을 실행하는 트랜잭션
33 2		stockRelease	재고 출고 트랜잭션
33 3		stockClearance	재고조사 후 등록 트랜잭션

33 4	team01.yaksutor.pharmacy.mapper	PhBoardMapper	getBoardList	커뮤니티 글 조회
33 5			getReplList	커뮤니티 댓글 조회
33 6			getQuestionCenterList	자주하는 질문 조회
33 7			getNoticeList	공지사항 조회
33 8			getQnaList	1:1 문의 조회
33 9			getQnaReplyList	1:1 문의 답변 조회
34 0			addBoard	커뮤니티 등록
34 1			modifyBoard	커뮤니티 수정
34 2			deleteBoard	커뮤니티 삭제
34 3			addRepl	커뮤니티 댓글 등록
34			modifyRepl	커뮤니티 댓글 수정

4			
34 5		deleteRepl	커뮤니티 댓글 삭제
34 6		addQuestionCenter	자주하는 질문 등록
34 7		modifyQuestionCenter	자주하는 질문 수정
34 8		deleteQuestionCenter	자주하는 질문 삭제
34 9		addNotice	공지사항 등록
35 0		modifyNotice	공지사항 수정
35 1		deleteNotice	공지사항 삭제
35 2		addQna	문의 등록
35 3		deleteQna	문의 삭제
35 4		addQnaReply	문의답변 등록

35 5		deleteQnaReply	문의답변 삭제
35 6		updateQna	문의 업데이트
35 7		getQnaListByKey	특정 QnA 리스트 조회
35 8	PhMedicineMapper	getSellMedicineList	판매의약품 리스트 조회
35 9	PhMemberMapper	modifyMyAccount	나의 계정 정보 수정
36 0	PhOrderMapper	getSellerInfoByGoodsCode	의약품 상품 코드로 판매자 정보 조회
36 1		addOrder	주문 추가
36 2		addOrderDetail	주문상세 정보 추가
36 4		updateOrderPrice	주문가격 업데이트
36 5		getOrderListById	회원의 Id 로 주문 리스트 조회
36		getOrderDetailListByOCode	주문코드로 주문상세

6			리스트 조회
36 7		deleteOrderDetail	주문상세 삭제
36 8		getOrderDetailByCode	주문코드로 주문상세조회
36 9		deleteOrderDetail	주문상세 정보 삭제
37 0		updateOrderStatus	주문상태 업데이트
37 1	PhRefundMapper	getRefundList	반품 목록 조회
37 2		getRefundDetailList	반품 상세 목록 조회
37 3		getRefundInfo	반품 정보 조회
37 4		getOrderDetailListByOCode	주문코드로 주문 상세 조회
37 5		getOrderByOCode	주문코드로 주문 조회
37 6		getRefundOrderList	주문 상세 조회

37 7		getPharId	약국 아이디 조회
37 8		insertRefund	반품 등록
37 9		getMediCodeByGoodsCode	판매 의약품 코드로 의약품 코드 조회
38 0		insertRefundDetail	반품 상세 등록
38 1		updateOrderState	주문 상태 수정
38 2		deleteShipping	shipping 삭제
38 3		deleteRefundDetail	refundDetail 삭제
38 4		deleteRefundDetail	refund 삭제
38 5		updateCancleRefund	order 상태 반품 -> 결제완료
38 6	PhShippingMapper	getShipInfoByNum	운송장 번호로 배송 정보 조회
38	PhShoppingCartMap	getShoppingCartList	조회

7	per		
38 8		addShoppingCart	추가
38 9		deleteShoppingCart	삭제
39 0		findByOMId	특정아이디로 장바구니 조회
39 1	PhStockMapper	getPharCodeById	아이디로 약국코드 조회
39 2		getPharmNameById	아이디로 약국 이름 조회
39 3		getPharmStockList	약국 재고 목록 조회
39 4		insertMedicine	의약품 정보 등록
39 5		insertPharmStock	약국 재고 등록
39 6		insertIngredient	성분 등록
39 7		insertEfficacy	효능 등록

39 8		getMedicineByCode	재고 코드로 의약품 정보 조회
39 9		getPharmStockByCode	재고 코드로 약국 재고 조회
40 0		getIngredientByCode	재고 코드로 성분 조회
40 1		getEfficacyByCode	재고 코드로 효능 조회
40 2		getStockQty	재고 개수 조회
40 3		stockRelease	약국 재고 출고
40 4		insertStockHistory	입출고 기록 등록
40 5		getStockInfo	재고 정보 조회
40 6		getStockHistory	재고 입출고 기록 조회
40 7		getOldStockCheck	6 개월 지난 재고 조회
40		insertStockClearance	재고 조사 이력 등록

8			
40 9			updateStockClearance 재고 조사 수정
41 0			updateStockClear 재고 삭제

5. 개발 / 서비스 환경

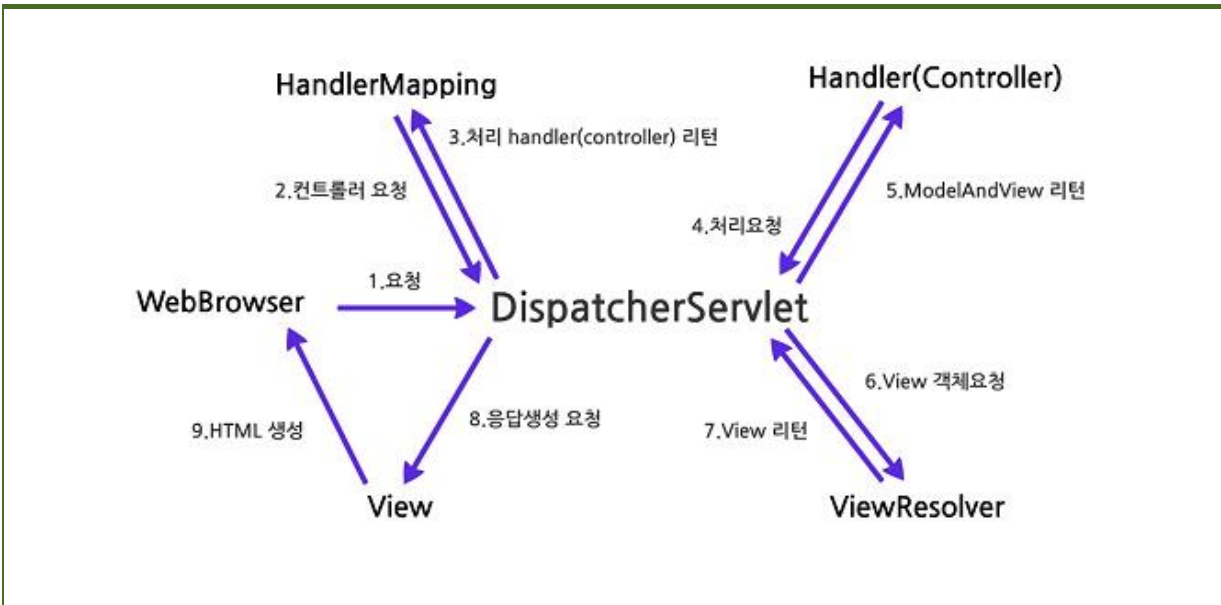
*	구분	개발 환경		서비스 환경
S / W	OS	Windows 10		Linux - CentOS Stream release 8
	웹브라우저	Chrome, Safari		Chrome, Safari, Microsoft Edge, Firefox
	WAS	apache-tomcat-10.x		apache-tomcat-10.x
	개발언어	서버	JDK17 / Servlet 6.x / UTF-8	
클라이언트		JavaScript / HTML5 / CSS / thymeleaf / jQuery		

	Framework	bootstrap 5 / Spring Framework 6.x / Mybatis 3.x	springframework 4.2.1 spring-webmvc 4.3.9 mybatis 3.3.1 mybatis-spring-boot 2.5.1 bootstrap 4.6
	DBMS	MySQL 8.0.26	MySQL 8.0.26
	Tools	Eclipse, Spring Tool Suite, IntelliJ IDEA, Visual Studio Code	STS 4.2.1

6. 어플리케이션

6.1 어플리케이션 프로시저

기본 어플리케이션 프로시저



6.2 서버 및 기타 설정

pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.1</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>team01</groupId>
  <artifactId>Yaksutor</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Yaksutor</name>
  <description>Yaksutor</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection/>
    <developerConnection/>
  </scm>
</project>

```

```

    <url/>
</scm>
<properties>
  <java.version>17</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jdbc</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>3.0.3</version>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>

```

```

    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter-test</artifactId>
    <version>3.0.3</version>
    <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.bgee.log4jdbc-log4j2/log4jdbc-log4j2-jdbc4.1 -->
<dependency>
    <groupId>org.bgee.log4jdbc-log4j2</groupId>
    <artifactId>log4jdbc-log4j2-jdbc4.1</artifactId>
    <version>1.16</version>
</dependency>
<!-- https://mvnrepository.com/artifact/nz.net.ultraq.thymeleaf/thymeleaf-layout-dialect -->
<dependency>
    <groupId>nz.net.ultraq.thymeleaf</groupId>
    <artifactId>thymeleaf-layout-dialect</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity6</artifactId>
</dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
                <excludes>

```

```

        <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
        </exclude>
    </excludes>
</configuration>
</plugin>
</plugins>
</build>

</project>

```

application.properties

```
spring.application.name=Yaksutor
```

```
# 서버 포트 설정
```

```
server.port=80
```

```
# Thymeleaf 캐시 비활성화
```

```
spring.thymeleaf.cache=false
```

```
# 데이터 소스 설정
```

```
# log4jdbc 를 사용하기 위해 1.드라이버 클래스를 설정하고, 2. url 앞에 log4jdbc 를 붙여야 합니다.
```

```
# spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
spring.datasource.driver-class-name=net.sf.log4jdbc.sql.jdbcapi.DriverSpy
```

```
#
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/ksmart51db?serverTimezone=UTC&characterEncoding=UTF8
```

```
spring.datasource.url=jdbc:log4jdbc:mysql://152.69.197.150:3306/ks51team01db?serverTimezone=UTC&characterEncoding=UTF8
```

```
spring.datasource.username=ks51team01id
spring.datasource.password=ks51team01pw

# MyBatis 설정
# classpath: -> src/main/resources/
mybatis.mapper-locations=classpath:mapper/**/*.xml

# Mapper 스캔할 type(package) 설정
# 여러 패키지를 설정할 경우 , 로 구분합니다.
mybatis.type-aliases-package=team01.yaksutor.*.dto,team01.yaksutor.dto

# 배너 설정
spring.banner.location=classpath:banner/banner.txt

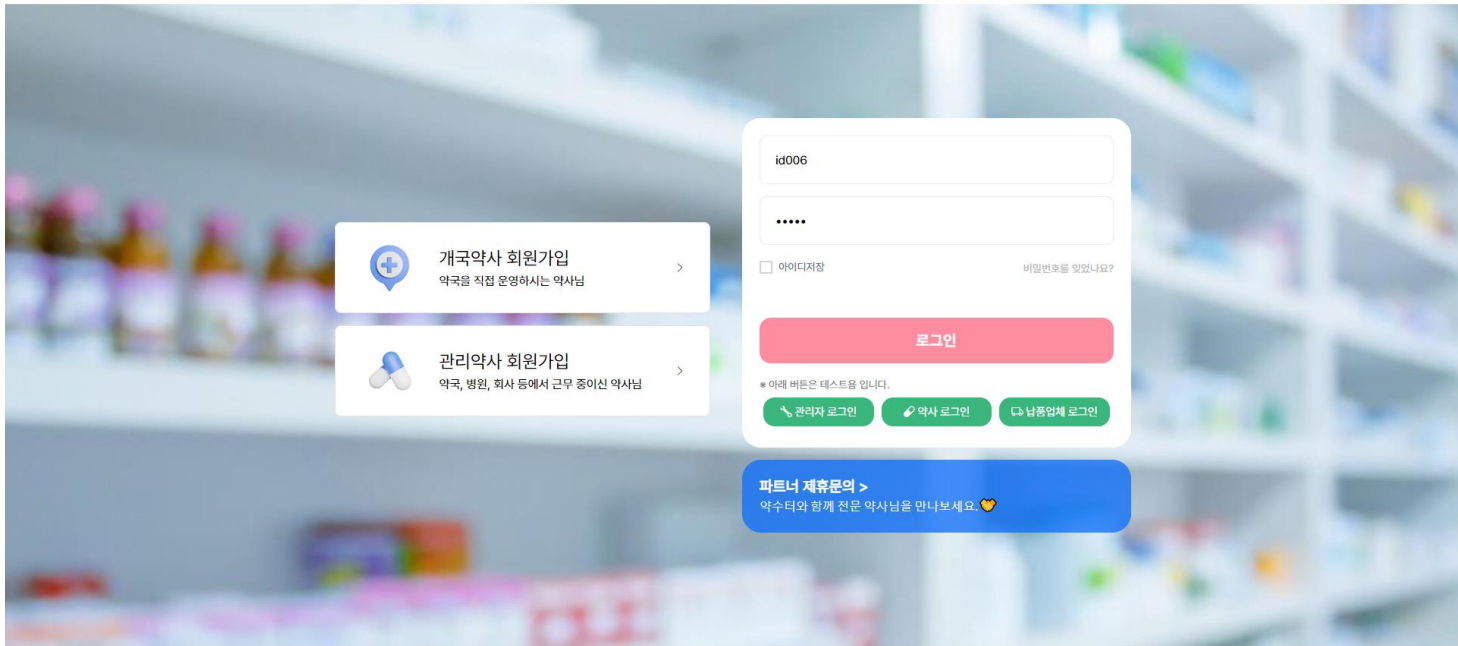
# 로거 설정 파일 위치
logging.config=classpath:logback-spring.xml

spring.servlet.multipart.max-file-size=50MB
spring.servlet.multipart.max-request-size=50MB

server.servlet.session.persistent=false
```

6.3 주요 기능별 정리

프로젝트 메인화면



길법진 팀원	박중연 팀장	최성식 팀원
본인 기능	본인 기능	메인화면(Index)
본인 기능	본인 기능	사용자 메인화면
본인 기능	본인 기능	커뮤니티 페이지
본인 기능	본인 기능	공지사항 페이지
본인 기능	본인 기능	1:1 문의 페이지
본인 기능	본인 기능	자주하는 질문 페이지
본인 기능	본인 기능	장바구니 페이지
본인 기능	본인 기능	주문하기 페이지
		레이아웃(fragment)



약수터 | 대표 김약수 | 사업자 등록 번호 : 123-34-34567 | 통신판매업 신고번호 : 제 2024-전북전주-000000호 [사업자 정보 확인]
 전라북도 전주시 덕진구 123 | 대표전화 : 1000-1234 | 문의: admin@yaksutor.com
 개인정보 처리방침 | 이용약관
 © 2024 YAKSUTOR



프로젝트 메인 페이지

- 팀 프로젝트의 메인페이지
- 개국약사와 관리약사 회원 가입이 가능하다
- 로그인 시 권한에 따라 각각 서로 다른 권한의 메인페이지로 이동한다.

■ Html(javascript)

```
<!DOCTYPE html>
<html class="js sizes websockets customelements history postmessage webworkers picture pointerevents
webanimations webgl srcset flexbox cssanimations csscolumns csscolumns-width csscolumns-span csscolumns-fill
```

```

csscolumns-gap csscolumns-rule csscolumns-rulecolor csscolumns-rulestyle csscolumns-rulewidth csscolumns-
breakbefore csscolumns-breakafter csscolumns-breakinside"
  lang="en" xmlns:th="http://www.thymeleaf.org" >
<head>
  <meta charset="utf-8">
  <title th:text="{title}"> </title>
  <meta content="ie=edge" http-equiv="x-ua-compatible">
  <meta content="" name="description">
  <meta content="width=device-width, initial-scale=1" name="viewport">
  <meta content="" property="og:title">
  <meta content="" property="og:type">
  <meta content="" property="og:url">
  <meta content="" property="og:image">

  <!-- Favicon -->
  <link rel="shortcut icon" th:href="@{/resource/user/assets/imgs/theme/favicon.svg}" type="image/x-icon">
  <!-- Template CSS -->
  <link rel="stylesheet" th:href="@{/resource/user/assets/css/main.css?v=5.6}">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-icons/1.10.5/font/bootstrap-icons.min.css"
rel="stylesheet">

  <link rel="stylesheet" th:href="@{/resource/user/assets/css/footer.css}" />
</head>

<body style="overflow: visible;"> <div class="body-overlay-1"> </div>

<!--End header-->
<main class="main pages">
  <div style="height: 70px;">
    <div class="logo logo-width-1">
      
    </div>
  </div>

  <div class="page-content pt-150 pb-150 indexBackground">
    <div class="container">
      <div class="row">
        <div class="col-xl-8 col-lg-10 col-md-12 m-auto">
          <div class="row align-items-center">
            <div class="col-lg-6 pr-30 d-lg-block">
              <div class="guide-banner">
                <div class="list-banner-box">
                  <div class="logo logo-width-1 mb-3 regForm hover-up"
th:onclick="|location.href='@{/member/memberInsert/owner}'|">
                    <div class="img-box">
                      
                    </div>
                    <div class="text-box">
                      <span class="name-text">개국약사 회원가입</span>
                      <span class="introduce-text">약국을 직접 운영하시는 약사님</span>
                    </div>
                    <i class="bi bi-chevron-right"> </i>
                  </div>
                  <div class="logo logo-width-1 mb-3 regForm hover-up"
th:onclick="|location.href='@{/member/memberInsert/staff}'|" >
                    <div class="img-box">

```

```

        
    </div>
    <div class="text-box">
        <span class="name-text">관리약사 회원가입</span>
        <span class="introduce-text">약국, 병원, 회사 등에서 근무 중이신
약사님</span>
    </div>
    <i class="bi bi-chevron-right"></i>
</div>
</div>
</div>
</div>
<div class="col-lg-6 col-md-8">
    <div class="login_wrap widget-taber-content background-white" >
        <div class="padding_eight_all bg-white" style="padding: 23px; border-radius: 25px; font-
family: 'Pretendard-Regular';">

            <form method="post" th:action="@{/login}">
                <input name="secret_key" th:value="secret" type="hidden" />
                <div class="form-group">
                    <input id="memberId" name="memberId" placeholder="회원아이디를
입력하세요. *" type="text" value="id006">
                </div>
                <div class="form-group">
                    <input id="memberPw" name="memberPw" placeholder="비밀번호를
입력하세요 *" type="password" value="pw006">
                </div>

                <div class="login_footer form-group mb-50">
                    <div class="chek-form">
                        <div class="custome-checkbox">
                            <input class="form-check-input" id="exampleCheckbox1"
name="checkbox" type="checkbox" value="">
                            <label class="form-check-label"
for="exampleCheckbox1"><span>아이디저장</span></label>
                        </div>
                    </div>
                    <a class="text-muted" href="#">비밀번호를 잊었나요?</a>
                </div>

                <div class="form-group" th:if="{error}">
                    <label class="form-label text-danger" th:text="{exception}"></label>
                </div>

                <div class="form-group">
                    <button class="btn btn-heading btn-block" id="indexLoginBtn" name="login"
style="width: 100%; background: rgb(255,140,159); font-family: 'Pretendard-Regular'; border-radius: 15px; font-
size: 20px; font-weight: 600"
                    type="submit">로그인</button>
                </div>
                ※ 아래 버튼은 테스트용 입니다.
                <div style="display: flex">
                    <a class="shortCut hover-up" data-login-id="admin" data-login-
pw="admin1234"><i class="bi bi-wrench"></i> 관리자 로그인</a>
                    <a class="shortCut hover-up" data-login-id="id007" data-login-
pw="pw007"><i class="bi bi-capsule"></i> 약사 로그인</a>
                    <a class="shortCut hover-up" data-login-id="id001" data-login-
pw="pw001"><i class="bi bi-truck"></i> 납품업체 로그인</a>

```



```

        </div>
    </form>
</div>
    <div class="padding_eight_all mt-3 hover-up"
onclick="location.href='https://forms.gle/1VRRReEqFk24heS8q8';" style="padding: 23px; border-radius: 25px;
background-color: rgba(13,110,253,0.76); cursor:pointer">
    <h5 style="color: white">파트너 제휴문의 > </h5>
    <p style="color: white">약수터와 함께 전문 약사님을 만나보세요.<i class="fs-
5">□</i> </p>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div style="background: #fafafa;">
<div th:replace="~{user/pharmacy/fragments/footer :: teamIntroFragment}"> </div>
<div th:replace="~{user/pharmacy/fragments/footer :: footerFragment}"> </div>
</div>
</main>

<!-- Preloader Start -->
<div id="preloader-active" style="display: none;">
    <div class="preloader d-flex align-items-center justify-content-center">
        <div class="preloader-inner position-relative">
            <div class="text-center">
                
            </div>
        </div>
    </div>
</div>

<!-- Vendor JS-->
<script th:src="@{/resource/user/assets/js/vendor/modernizr-3.6.0.min.js}"> </script>
<script th:src="@{/resource/user/assets/js/vendor/jquery-3.6.0.min.js}"> </script>
<script th:src="@{/resource/user/assets/js/vendor/jquery-migrate-3.3.0.min.js}"> </script>
<script th:src="@{/resource/user/assets/js/vendor/bootstrap.bundle.min.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/slick.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/jquery.syotimer.min.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/wow.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/perfect-scrollbar.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/magnific-popup.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/select2.min.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/waypoints.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/counterup.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/jquery.countdown.min.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/images-loaded.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/isotope.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/scrollup.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/jquery.vticker-min.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/jquery.theia.sticky.js}"> </script>
<script th:src="@{/resource/user/assets/js/plugins/jquery.elevatezoom.js}"> </script>
<!-- Template JS -->
<script th:src="@{/resource/user/assets/js/main.js?v=5.6}"> </script> <a href="#top" id="scrollUp" style="position:
fixed; z-index: 2147483647; display: none;"> <i class="fi-rs-arrow-small-up"> </i> </a>
<script th:src="@{/resource/user/assets/js/shop.js?v=5.6}"> </script>

```

```

<script>
    $('.shortCut').click(e => {
        e.preventDefault();
        const target = e.target;
        const $loginForm = $('<form> </form>');
        const $memberId = $('<input />', {name: 'memberId', value: $(target).data('loginId')});
        const $memberPw = $('<input />', {name: 'memberPw', value: $(target).data('loginPw')});
        const $secret = $('<input />', {name: 'secret_key', value: $('input[name="secret_key"]').val()});
        $loginForm.attr('action', '/login');
        $loginForm.attr('method', 'post');
        $loginForm.append($memberId, $memberPw, $secret);
        $('body').append($loginForm);
        $loginForm.submit();
    });
</script>

</body>
</html>

```

■ Controller

```

package team01.yaksutor.security.controller;

import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lombok.RequiredArgsConstructor;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.annotation.AuthenticationPrincipal;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import team01.yaksutor.common.mapper.MemberMapper;
import team01.yaksutor.common.service.MemberService;
import team01.yaksutor.dto.Member;

@Controller
@RequiredArgsConstructor
public class LoginController {

    @GetMapping(value="/")
    public String login(@RequestParam(value = "error", required = false) String error,
        @RequestParam(value = "exception", required = false) String exception, Model model){
        model.addAttribute("error",error);
        model.addAttribute("exception",exception);
    }
}

```

```

    return "index";
}

@GetMapping(value = "/logout")
public String logout(HttpServletRequest request, HttpServletResponse response) {
    Authentication authentication = SecurityContextHolder.getContextHolderStrategy().getContext().getAuthentication();
    if (authentication != null) {
        new SecurityContextLogoutHandler().logout(request, response, authentication);
    }

    return "redirect:/";
}

@GetMapping(value="/denied")
public String accessDenied(@RequestParam(value = "exception", required = false) String exception,
    @AuthenticationPrincipal Member member, Model model) {

    model.addAttribute("memberId", member.getMemberId());
    model.addAttribute("exception", exception);

    return "login/denied";
}
}

```

■ 로그인 성공시 Handler

```

package team01.yaksutor.security.handler;

import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lombok.extern.slf4j.Slf4j;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.web.DefaultRedirectStrategy;
import org.springframework.security.web.RedirectStrategy;
import org.springframework.security.web.authentication.SimpleUrlAuthenticationSuccessHandler;
import org.springframework.security.web.savedrequest.HttpSessionRequestCache;
import org.springframework.security.web.savedrequest.RequestCache;
import org.springframework.security.web.savedrequest.SavedRequest;
import org.springframework.stereotype.Component;
import org.springframework.web.servlet.handler.HandlerMappingIntrospector;
import team01.yaksutor.dto.AccountContext;
import team01.yaksutor.dto.Member;

```

```

import java.io.IOException;

@Component
@Slf4j
public class FormAuthenticationSuccessHandler extends SimpleUrlAuthenticationSuccessHandler {
    private final RequestCache requestCache = new HttpSessionRequestCache();
    private final RedirectStrategy redirectStrategy = new DefaultRedirectStrategy();

    @Override
    public void onAuthenticationSuccess(final HttpServletRequest request, final HttpServletResponse response, final
Authentication authentication) throws IOException {

        Member sessionMember = (Member) authentication.getPrincipal();
        System.out.println(sessionMember);
        request.getSession().setAttribute("S_ID", sessionMember.getMemberId());
        request.getSession().setAttribute("S_LEVEL", sessionMember.getMemberLevelName());
        request.getSession().setAttribute("S_NAME", sessionMember.getMemberName());

        System.out.println("1. " + request.getSession().getAttribute("S_ID"));
        System.out.println("2. " + request.getSession().getAttribute("S_LEVEL"));
        System.out.println("3. " + request.getSession().getAttribute("S_NAME"));

        String memberInfo = authentication.getName();

        int start = memberInfo.indexOf("memberLevelName=") + "memberLevelName=".length();
        String memberLevel = memberInfo.substring(start);
        memberLevel = memberLevel.substring(0, memberLevel.indexOf(","));

        String defaultPath = "/";

        switch (memberLevel){
            case "관리자" -> defaultPath = "/admin/adminMain";
            case "개국약사", "관리약사" -> defaultPath = "/pharm/main";
            case "납품업체" -> defaultPath = "/supplier/supplierMain";
        }
        setDefaultTargetUrl(defaultPath);

        SavedRequest savedRequest = requestCache.getRequest(request, response);

        if(savedRequest!=null) {
            redirectStrategy.sendRedirect(request, response, defaultPath);
        }
    }
}

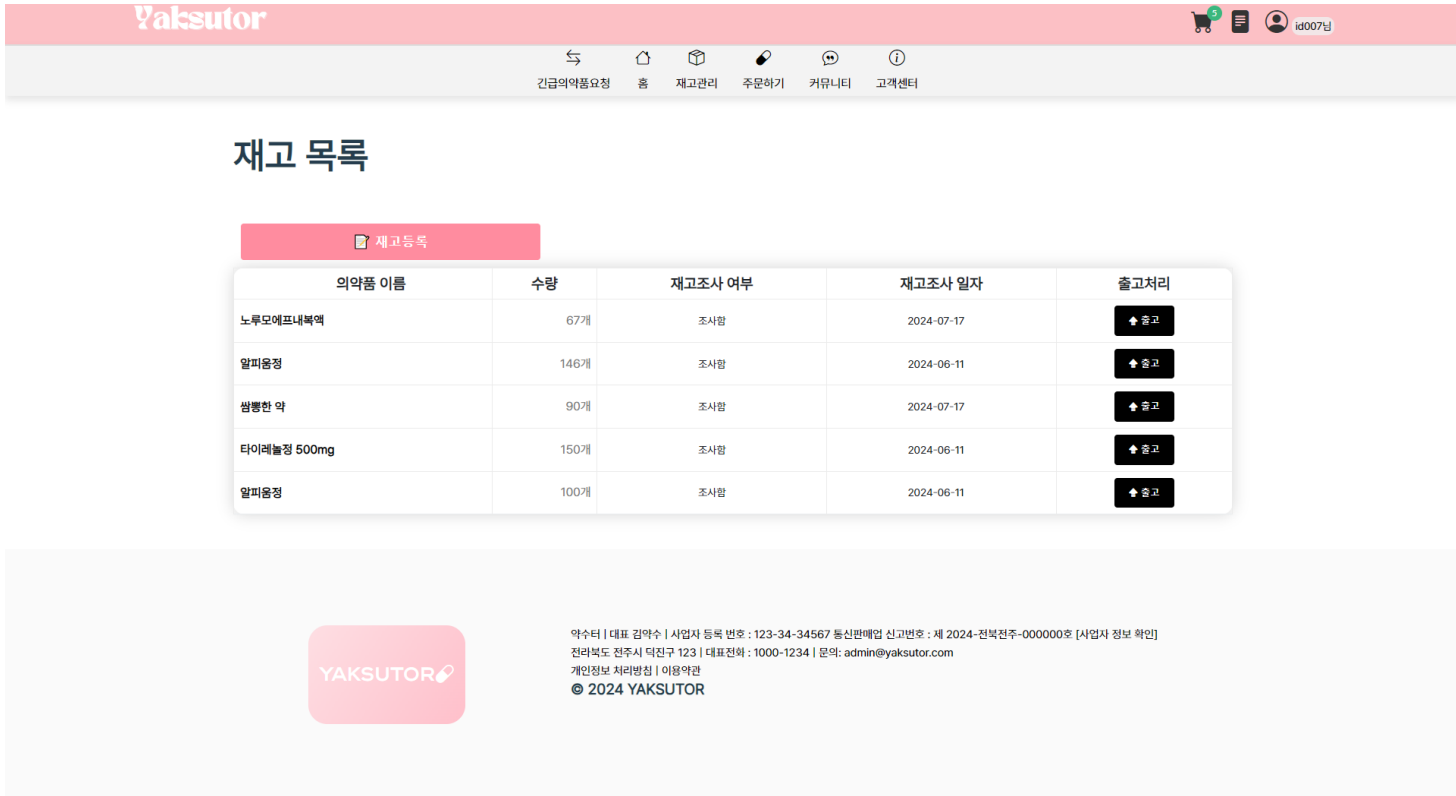
```

```

else {
    redirectStrategy.sendRedirect(request, response, getDefaultTargetUrl());
}
}
}
}

```

약국 사용자 - 재고 조회 페이지



약국 사용자 - 재고 조회 페이지

- 약국에서 보유중인 의약품의 재고 리스트를 화면에 출력한다
- 재고등록 버튼을 클릭하면 재고를 새로 등록할 수 있는 페이지로 이동한다.
- 출고 버튼을 클릭하면 출고처리를 수행하는 페이지로 이동한다.

■ Html(javascript)

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"

```

```

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
layout:decorate="~{user/pharmacy/layout/default}">
<head>
  <title th:text="{title}"> </title>
</head>

<th:block layout:fragment="customContent">
  <main class="main">
    <div class="container mb-30 mt-50 font-1">
      <div class="row">
        <div class="col-xl-10 col-lg-12 m-auto">
          <div class="mb-50">
            <h1 class="heading-2 mb-10" th:text="{content}"> </h1>
          </div>
          <a class="btn btn-sm" style="width: 30%; margin: 10px; font-size: 17px;background: #ff8c9f"
th:href="@{/pharm/myStockInsert}"> 재고등록 </a>
          <table class="table table-wishlist" style="font-family: 'Pretendard-Regular'">
            <thead>
              <tr class="text-center fs-5">
                <th scope="col" class="custome-checkbox start pl-30">의약품 이름 </th>
                <th scope="col">수량 </th>
                <th scope="col" style="text-align: center">재고조사 여부 </th>
                <th scope="col" style="text-align: center">재고조사 일자 </th>
                <th scope="col" class="end">출고처리 </th>
              </tr>
            </thead>
            <tbody>
              <tr class="pt-30" th:each="s : {stockList}" th:if="{ stockList != null}">
                <td class="product-des product-name">
                  <h6> <a class="mb-10" style="color: black"
th:href="@{/pharm/stockDetailView(stockCode={s.stockCode},mediCode={s.mediCode})}"
th:text="{s.medicine.mediName}"> </a> </h6>
                </td>
                <td class="price text-end" data-title="Price">
                  <p th:text="{s.pharQty + '개'}"> </p>
                </td>
                <td class="text-center detail-info" data-title="Stock">
                  <span th:text="{s.stockChecked}"> </span>
                </td>
                <td class="text-center detail-info" data-title="Stock">
                  <div class="detail-extralink mr-15" th:text="{s.lastDate}">
                </div>

```

```

                </td>
                <td data-title="Cart" style="text-align: center">
                    <a class="btn btn-sm" style="background: black;"
th:href="@{/pharm/myStockRelease(stockCode=${s.stockCode},mediName=${s.medicine.mediName})}" > <i
class="bi bi-capslock-fill"> </i> 출고 </a>
                </td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</div>
</main>

</th:block>

<th:block layout:fragment="customJsFile">
</th:block>

<th:block layout:fragment="customJs">
    <script th:inline="javascript">
        $('.stockChecked').each(function(){
            const value = $(this).text();
            if(value == '조사함'){
                $(this).addClass('in-stock');
            } else{
                $(this).addClass('out-stock')
            }
        });
    </script>
</th:block>
</html>

```

■ Controller

```
package team01.yaksutor.pharmacy.controller;
```

```

import com.fasterxml.jackson.databind.util.JSONPObject;
import jakarta.servlet.http.HttpServletRequest;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import team01.yaksutor.dto.*;
import team01.yaksutor.pharmacy.mapper.PhStockMapper;
import team01.yaksutor.pharmacy.service.PhStockService;

import java.util.List;

@Controller
@RequestMapping("/pharm")
@RequiredArgsConstructor
@Slf4j
public class PhPharmStockController {
    private final PhStockMapper phStockMapper;
    private final PhStockService phStockService;
    private final HttpServletRequest request;

    /**
     * 재고조사 페이지 맵핑
     * 작성자: 길범진
     * @param model
     * @return
     */
    @GetMapping("/checkMedi")
    public String pharmStrock(Model model) {
        String sid = request.getSession().getAttribute("S_ID").toString();
        String pharCode = phStockMapper.getPharCodeById(sid);
        List<PharmStock> oldStockList = phStockMapper.getOldStockCheck(pharCode);
        log.info("여기보세요: {}", oldStockList);

        model.addAttribute("oldStockList", oldStockList);
        model.addAttribute("title", "재고조사");
        return "user/pharmacy/pharmstock/checkMedi";
    }

    /**
     * 재고 등록의 GetMapping
     * @param model
     * @return
     */
    @GetMapping("/myStockInsert")
    public String myStockInsert(Model model) {

```



```

model.addAttribute("title", "재고 추가");
model.addAttribute("content", "재고 추가");

return "user/pharmacy/pharmstock/myStockInsert";
}

/**
 * 내 재고 목록 페이지 맵핑
 * 작성자: 길범진
 * @param model
 * @return
 */
@GetMapping("/myStockSearchList")
public String myStockSearchList(Model model) {
    String sid = request.getSession().getAttribute("S_ID").toString();
    log.info("sid: {}", sid);
    String pharCode = phStockMapper.getPharCodeById(sid);
    List<PharmStock> stockList = phStockMapper.getPharmStockList(pharCode);

    model.addAttribute("stockList", stockList);
    model.addAttribute("title", "재고 목록");
    model.addAttribute("content", "재고 목록");

    return "user/pharmacy/pharmstock/myStockSearchList";
}

/**
 * 내 재고 출고 페이지 맵핑
 * 작성자: 길범진
 * @param model
 * @param stockCode
 * @param mediName
 * @return
 */
@GetMapping("/myStockRelease")
public String myStockRelease(Model model,
                             @RequestParam(value="stockCode") String stockCode,
                             @RequestParam(value="mediName") String mediName) {

    String thisStockCode = stockCode;
    String stockQty = phStockMapper.getStockQty(stockCode);
    String thisMedi = mediName;

    model.addAttribute("thisStockCode", thisStockCode);
    model.addAttribute("stockQty", stockQty);
    model.addAttribute("mediName", thisMedi);
    model.addAttribute("title", "출고");
    model.addAttribute("content", "출고");

    return "user/pharmacy/pharmstock/myStockRelease";
}

```

```

}

/**
 * 내 재고 입출고 기록 페이지 맵핑
 * 작성자: 길범진
 * @param model
 * @return
 */
@GetMapping("/myStockHistory")
public String myStockHistory(Model model) {
    String sid = request.getSession().getAttribute("S_ID").toString();
    String pharCode = phStockMapper.getPharCodeById(sid);
    List<StockHistory> stockHistoryList = phStockMapper.getStockHistory(pharCode);

    model.addAttribute("historyList", stockHistoryList);
    model.addAttribute("title", "입출고 기록");
    model.addAttribute("content", "입출고 기록");

    return "user/pharmacy/pharmstock/myStockHistory";
}

/**
 * 내 재고 상세페이지 맵핑
 * 작성자: 길범진
 * @param model
 * @param mediCode
 * @return
 */
@GetMapping("/stockDetailView")
public String stockDetailView(Model model,
                              @RequestParam(value="mediCode") String mediCode) {

    StockInfo stockInfo = new StockInfo();
    String sid = request.getSession().getAttribute("S_ID").toString();
    String pharCode = phStockMapper.getPharCodeById(sid);
    stockInfo.setMedicine(phStockMapper.getMedicineByCode(mediCode));
    stockInfo.setPharmStock(phStockMapper.getPharmStockByCode(mediCode, pharCode));
    List<Ingredient> ingrList = phStockMapper.getIngredientByCode(mediCode);
    stockInfo.setIngrList(ingrList);
    stockInfo.setEffiList(phStockMapper.getEfficacyByCode(mediCode));

    log.info("stockInfo: {}", stockInfo);

    model.addAttribute("stockInfo", stockInfo);
    model.addAttribute("title", "재고 상세");
    model.addAttribute("content", "재고 상세");

    return "user/pharmacy/pharmstock/stockDetailView";
}

```

```

}

/**
 * 재고 등록 ajax 요청을 받는 메소드
 * @param stockInfo
 * @return
 */
@PostMapping("/myStockInsert")
@ResponseBody
public String myStockInsert(@RequestParam(value="stockInfo") StockInfo stockInfo){
    String sid = request.getSession().getAttribute("S_ID").toString();
    log.info("stockInfo: {}", stockInfo);
    // 세션 아이디와 화면에서 받은 정보를 트랜잭션 실행을 위해 서비스에 정의된 메소드에 전달
    phStockService.stockInsert(stockInfo, sid);

    return "user/pharmacy/pharmstock/myStockInsert";
}

/**
 * 내 재고 출고 submit 맵핑
 * 작성자: 길범진
 * @param qty
 * @param stockCode
 * @return
 */
@PostMapping("/myStockRelease")
public String myStockRelease(@RequestParam(value="qty") String qty,
                             @RequestParam(value="stockCode") String stockCode){
    phStockService.stockRelease(qty, stockCode);

    return "user/pharmacy/pharmstock/myStockSearchList";
}

/**
 * 내 재고 조사 ajax 요청
 * 작성자: 길범진
 * @param stockClearance
 * @return
 */
@PostMapping("/checkMedi")
@ResponseBody
public String checkMedi(@RequestBody StockClearance stockClearance){
    log.info("checkMedi: {}", stockClearance);
    int oldQty = stockClearance.getPreInveQty();
    int newQty = stockClearance.getPostInveQty();
    int abnormalQty = oldQty - newQty;
    abnormalQty = Math.abs(abnormalQty);
    stockClearance.setAdnormalQty(abnormalQty);
    if(abnormalQty == 0) {
        stockClearance.setAdnormalReason("없음");
        stockClearance.setAdnormalChecked("정상");
    } else {
        stockClearance.setAdnormalChecked("비정상");
    }
}

```

```

    }
    log.info("new checkMedi: {}", stockClearance);
    phStockService.stockClearance(stockClearance);

    return "redirect:user/pharmacy/pharmstock/checkMedi";
}
}

```

■ Service

```

package team01.yaksutor.pharmacy.service;

import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import team01.yaksutor.dto.*;
import team01.yaksutor.pharmacy.mapper.PhStockMapper;

import java.util.List;

@Service
@RequiredArgsConstructor
@Transactional
@Slf4j
public class PhStockService {
    private final PhStockMapper phStockMapper;

    /**
     * medicine, PharmStock, Ingredient, Efficacy 에 값을 담아서 쿼리문을 실행하는 트랜잭션
     * 작성자: 길범진
     * @param stockInfo
     * @param sid
     */
    @Transactional
    public void stockInsert(StockInfo stockInfo,
                           String sid) {
        String pharCode = phStockMapper.getPharCodeById(sid);
        String pharmName = phStockMapper.getPharmNameById(sid);

        Medicine medicine = stockInfo.getMedicine();
        medicine.setRegMId(sid);
        phStockMapper.insertMedicine(medicine);

        PharmStock pharmStock = stockInfo.getPharmStock();
        pharmStock.setMediCode(medicine.getMediCode());
        pharmStock.setPharCode(pharCode);
        pharmStock.setPharmName(pharmName);
        pharmStock.setExpiraDate(medicine.getMediUsebydate());
        phStockMapper.insertPharmStock(pharmStock);

        List<Ingredient> ingrList = stockInfo.getIngrList();
        for (Ingredient ingr : ingrList) {
            ingr.setMediCode(pharmStock.getMediCode());
            ingr.setRegMId(sid);
            phStockMapper.insertIngredient(ingr);
        }

        List<Efficacy> effiList = stockInfo.getEffiList();
        for (Efficacy eff : effiList) {
            eff.setMediCode(pharmStock.getMediCode());
            eff.setRegMId(sid);
            phStockMapper.insertEfficacy(eff);
        }
    }

    /**
     * 재고 출고 트랜잭션
     * 작성자: 길범진
     * @param qty
     * @param stockCode
     */
    @Transactional
    public void stockRelease(String qty,
                             String stockCode) {
        log.info("qty: {}", qty);
        log.info("stockCode: {}", stockCode);

        int newQty = Integer.parseInt(qty);
    }
}

```

```

PharmStock beforeStock = phStockMapper.getStockInfo(stockCode);

StockHistory sh = new StockHistory();
int inout = beforeStock.getPharQty() - Integer.parseInt(qty);
sh.setInoutQty(beforeStock.getPharQty());
sh.setRealInoutQty(inout);
sh.setOrderInoutQty(inout);
sh.setInoutType("출고");
sh.setAfterQty(newQty);
sh.setPharCode(beforeStock.getPharCode());
sh.setMediCode(beforeStock.getMediCode());

PharmStock pharmStock = new PharmStock();
pharmStock.setStockCode(stockCode);
pharmStock.setPharQty(newQty);
phStockMapper.stockRelease(pharmStock);
phStockMapper.insertStockHistory(sh);

}

/**
 * 재고조사 후 등록 트랜잭션
 * 작성자: 김범진
 * @param stockClearance
 */
@Transactional
public void stockClearance(StockClearance stockClearance){
    phStockMapper.insertStockClearance(stockClearance);
    String stockCode = stockClearance.getStockCode();
    int isUpdate = stockClearance.getAdnormalQty();
    int pharQty = stockClearance.getPostInveQty();
    PharmStock pharmStock = new PharmStock();
    pharmStock.setStockCode(stockCode);
    pharmStock.setPharQty(pharQty);
    if(isUpdate != 0){
        phStockMapper.updateStockClearance(pharmStock);
    } else{
        phStockMapper.updateStockClear(pharmStock);
    }
}
}
}

```

■ Mapper(SQL)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="team01.yaksutor.pharmacy.mapper.PhStockMapper">

    <resultMap id="medicineListMap" type="Medicine">
        <id column="medi_code" property="mediCode" />
        <result column="reg_m_id" property="regMId" />
        <result column="medi_name" property="mediName" />
        <result column="medi_img" property="mediImg" />
        <result column="medi_drugform" property="mediDrugform" />
        <result column="medi_drugtype" property="mediDrugtype" />
        <result column="medi_state" property="mediState" />
        <result column="medi_usebydate" property="mediUsebydate" />
        <result column="reg_date" property="regDate" />

        <association property="imgData" javaType="ImgData">
            <id column="file_idx" property="fileIdx"/>
            <result column="file_origin name" property="fileOriginName"/>
            <result column="file_new name" property="fileNewName"/>
            <result column="file_path" property="filePath"/>
            <result column="file_size" property="fileSize"/>
            <result column="file_reg_date" property="fileRegDate"/>
        </association>
    </resultMap>

    <resultMap id="medicalInfoMap" type="Medicine">
        <id column="medi_code" property="mediCode" />
        <result column="reg_m_id" property="regMId" />
        <result column="medi_name" property="mediName" />
        <result column="medi_img" property="mediImg" />
        <result column="medi_drugform" property="mediDrugform" />
        <result column="medi_drugtype" property="mediDrugtype" />
        <result column="medi_state" property="mediState" />
        <result column="medi_usebydate" property="mediUsebydate" />
        <result column="reg_date" property="regDate" />

        <association property="imgData" javaType="ImgData">
            <id column="file_idx" property="fileIdx"/>
            <result column="file_origin name" property="fileOriginName"/>
            <result column="file_new name" property="fileNewName"/>
            <result column="file_path" property="filePath"/>
            <result column="file_size" property="fileSize"/>
        </association>
    </resultMap>

```

```

        <result column="file_reg_date"          property="fileRegDate"/>
    </association>
</resultMap>

<resultMap id="ingrListMap" type="Ingredient">
    <id      column="ingredient_code"  property="ingredientCode"/>
    <result  column="medi_code"        property="mediCode"/>
    <result  column="reg_m_id"         property="regMid"/>
    <result  column="ingredient"       property="ingredient"/>
    <result  column="content"          property="content"/>
    <result  column="reg_date"         property="regDate"/>
</resultMap>

<resultMap id="effiListMap" type="Efficacy">
    <id      column="efficacy_code"    property="efficacyCode"/>
    <result  column="medi_code"        property="mediCode"/>
    <result  column="reg_m_id"         property="regMid"/>
    <result  column="efficacy"         property="efficacy"/>
    <result  column="reg_date"         property="regDate"/>
</resultMap>

<resultMap id="getStockList" type="PharmStock">
    <id      column="stock_code"       property="stockCode"/>
    <result  column="medi_code"        property="mediCode"/>
    <result  column="phar_code"        property="pharCode"/>
    <result  column="pharm_name"       property="pharmName"/>
    <result  column="company"          property="company"/>
    <result  column="phar_qty"         property="pharQty"/>
    <result  column="manufact_date"    property="manufactDate"/>
    <result  column="expira_date"      property="expiraDate"/>
    <result  column="stock_checked"    property="stockChecked"/>
    <result  column="last_date"        property="lastDate"/>

    <association property="medicine" javaType="Medicine">
        <id      column="medi_code"    property="mediCode" />
        <result  column="reg_m_id"     property="regMid" />
        <result  column="medi_name"    property="mediName" />
        <result  column="medi_img"     property="mediImg" />
        <result  column="medi_drugform" property="mediDrugform" />
        <result  column="medi_drugtype" property="mediDrugtype" />
        <result  column="medi_state"   property="mediState" />
        <result  column="medi_usebydate" property="mediUsebydate" />
        <result  column="reg_date"     property="regDate" />
    </association>
</resultMap>

<resultMap id="getStockInfo" type="PharmStock">
    <id      column="stock_code"       property="stockCode"/>
    <result  column="medi_code"        property="mediCode"/>
    <result  column="phar_code"        property="pharCode"/>
    <result  column="pharm_name"       property="pharmName"/>
    <result  column="company"          property="company"/>
    <result  column="phar_qty"         property="pharQty"/>
    <result  column="manufact_date"    property="manufactDate"/>
    <result  column="expira_date"      property="expiraDate"/>
    <result  column="stock_checked"    property="stockChecked"/>
    <result  column="last_date"        property="lastDate"/>
</resultMap>

<resultMap id="getStockHistoryList" type="StockHistory">
    <id      column="stock_history_code" property="stockHistoryCode"/>
    <result  column="phar_code"          property="pharCode"/>
    <result  column="medi_code"         property="mediCode"/>
    <result  column="inout_qty"         property="inoutQty"/>
    <result  column="real_inout_qty"    property="realInoutQty"/>
    <result  column="order_inout_qty"   property="orderInoutQty"/>
    <result  column="inout_type"        property="inoutType"/>
    <result  column="after_qty"         property="afterQty"/>
    <result  column="inout_date"        property="inoutDate"/>

    <association property="medicine" javaType="Medicine">
        <id      column="medi_code"    property="mediCode" />
        <result  column="reg_m_id"     property="regMid" />
        <result  column="medi_name"    property="mediName" />
        <result  column="medi_img"     property="mediImg" />
        <result  column="medi_drugform" property="mediDrugform" />
        <result  column="medi_drugtype" property="mediDrugtype" />
        <result  column="medi_state"   property="mediState" />
        <result  column="medi_usebydate" property="mediUsebydate" />
        <result  column="reg_date"     property="regDate" />
    </association>
</resultMap>

<select id="getPharCodeById" parameterType="string" resultType="string">
    SELECT
        p.phar_code
    FROM
        pharmacy AS p
    WHERE
        p.phar_id = #{sid};
</select>

<select id="getPharmNameById" parameterType="string" resultType="string">
    SELECT
        p.p name

```

```

FROM
  pharmacy AS p
WHERE
  p.phar_id = #{sid};
</select>

<select id="getPharmStockList" parameterType="string" resultMap="getStockList">
  SELECT
    p.stock_code,
    p.medi_code,
    p.phar_code,
    p.pharm_name,
    p.phar_qty,
    p.stock_checked,
    p.last_date,
    m.medi_name
  FROM
    pharm_stock AS p
    left join
    medicine AS m
    on
    p.medi_code = m.medi_code
  WHERE
    p.phar_code = #{pharCode};
</select>

<insert id="insertMedicine" parameterType="Medicine">
  <selectKey keyProperty="mediCode" resultType="String" order="BEFORE">
    /* 상품 자동 증가 코드 */
    SELECT
      case
      when COUNT(medi_code) = 0 then 'medi_code_1'
      ELSE
      CONCAT('medi_code ',
      max(cast(SUBSTRING_INDEX(medi_code, 'medi_code_', -1) AS UNSIGNED))+1
      )
      END AS mediCode
    FROM
      medicine;
  </selectKey>
  INSERT INTO medicine
  (medi_code, reg_m_id, medi_name, medi_drugform, medi_drugtype, medi_state, medi_usebydate,
  reg_date)
  VALUES (#{mediCode}, #{regMId}, #{mediName}, #{mediDrugform}, #{mediDrugtype}, #{mediState},
  #{mediUsebydate}, CURDATE())
</insert>

<insert id="insertPharmStock" parameterType="PharmStock">
  <selectKey keyProperty="stockCode" resultType="String" order="BEFORE">
    /* 상품 자동 증가 코드 */
    SELECT
      case
      when COUNT(stock_code) = 0 then 'stock_code_1'
      ELSE
      CONCAT('stock_code ',
      max(cast(SUBSTRING_INDEX(stock_code, 'stock_code_', -1) AS UNSIGNED))+1
      )
      END AS stockCode
    FROM
      pharm_stock;
  </selectKey>
  INSERT INTO pharm_stock
  (stock_code, medi_code, phar_code, pharm_name, company, phar_qty, expira_date, stock_checked,
  last date)
  VALUES (#{stockCode}, #{mediCode}, #{pharCode}, #{pharmName}, #{company}, #{pharQty}, #{expiraDate}, '무', CURDATE())
</insert>

<insert id="insertIngredient" parameterType="Ingredient">
  <selectKey keyProperty="ingredientCode" resultType="String" order="BEFORE">
    /* 상품 자동 증가 코드 */
    SELECT
      case
      when COUNT(ingredient_code) = 0 then 'ingr_1'
      ELSE
      CONCAT('ingr ',
      max(cast(SUBSTRING_INDEX(ingredient_code, 'ingr_', -1) AS UNSIGNED))+1
      )
      END AS ingredientCode
    FROM
      ingredient;
  </selectKey>
  INSERT INTO ingredient
  (ingredient_code, medi_code, reg_m_id, Ingredient, Content, reg_date)
  VALUES (#{ingredientCode}, #{mediCode}, #{regMId}, #{ingredient}, #{content}, CURDATE())
</insert>

<insert id="insertEfficacy" parameterType="Efficacy">
  <selectKey keyProperty="efficacyCode" resultType="String" order="BEFORE">
    /* 상품 자동 증가 코드 */
    SELECT
      case
      when COUNT(efficacy_code) = 0 then 'effi_1'
      ELSE
      CONCAT('effi ',
      max(cast(SUBSTRING_INDEX(efficacy_code, 'effi_', -1) AS UNSIGNED))+1
      )
      END AS efficacyCode
    FROM

```

```

        efficacy;
    </selectKey>
    INSERT INTO efficacy
    (efficacy_code, medi_code, reg_m_id, efficacy, reg_date)
    VALUES ({TefficacyCode}, {medICode}, {regMId}, {Tefficacy}, CURDATE())
</insert>

<select id="getMedicineByCode" parameterType="string" resultMap="medicalInfoMap">
    SELECT
        m.*,
        i.*
    FROM
        medicine AS m
        left join
        img_data AS i
        on
        m.medi_img = i.file_idx
    WHERE
        m.medi_code = #{mediCode};
</select>

<select id="getPharmStockByCode" parameterType="string" resultMap="getStockList">
    SELECT
        p.*
    FROM
        pharm_stock AS p
    WHERE
        p.medi_code = #{mediCode} AND p.phar_code = #{pharCode};
</select>

<select id="getIngredientByCode" parameterType="string" resultMap="ingrListMap">
    SELECT
        i.*
    FROM
        ingredient AS i
    WHERE
        i.medi_code = #{mediCode};
</select>

<select id="getEfficacyByCode" parameterType="string" resultMap="effiListMap">
    SELECT
        e.*
    FROM
        efficacy AS e
    WHERE
        e.medi_code = #{mediCode};
</select>

<select id="getStockQty" parameterType="string" resultType="string">
    SELECT
        p.phar_qty
    FROM
        pharm_stock AS p
    WHERE
        p.stock_code = #{stockCode};
</select>

<update id="stockRelease" parameterType="PharmStock">
    UPDATE pharm_stock
    SET
        phar_qty=#{pharQty}
    WHERE stock_code=#{stockCode};
</update>

<insert id="insertStockHistory">
    <selectKey keyProperty="stockHistoryCode" resultType="String" order="BEFORE">
        /* 상품 자동 증가 코드 */
        SELECT
            case
            when COUNT(stock_history_code) = 0 then 'sh_code_1'
            ELSE
            CONCAT('sh code ',
            max(cast(SUBSTRING_INDEX(stock_history_code, 'sh_code_', -1) AS UNSIGNED))+1
            )
            END AS stockHistoryCode
        FROM
            stock_history;
    </selectKey>
    INSERT INTO stock_history
    (stock_history_code, phar_code, medi_code, inout_qty, real_inout_qty, order_inout_qty,
    inout_type, after_qty, inout_date)
    VALUES ({stockHistoryCode}, #{pharCode}, #{mediCode}, #{inoutQty}, #{realInoutQty},
    #{orderInoutQty}, #{inoutType}, #{afterQty}, CURDATE())
</insert>

<select id="getStockInfo" parameterType="String" resultMap="getStockInfo">
    SELECT
        p.*
    FROM
        pharm_stock AS p
    WHERE
        p.stock_code = #{stockCode};
</select>

<select id="getStockHistory" parameterType="string" resultMap="getStockHistoryList">

```



```

SELECT
    s.stock_history_code,
    s.phar_code,
    s.medi_code,
    s.inout_qty,
    s.real_inout_qty,
    s.inout_type,
    s.after_qty,
    s.inout_date,
    m.medi_name
FROM
    stock_history AS s
    INNER join
    medicine AS m
    on
    s.medi_code = m.medi_code
WHERE
    s.phar_code = #{pharCode};
</select>

<select id="getOldStockCheck" parameterType="string" resultMap="getStockList">
    <![CDATA[
        SELECT
            p.*,
            m.*
        FROM
            pharm_stock AS p
            INNER join
            medicine AS m
            on
            m.medi_code = p.medi_code
        WHERE
            p.last_date <= DATE_SUB(CURDATE(), INTERVAL 6 MONTH) AND p.phar_code = #{pharCode};
    ]]>
</select>






<insert id="insertStockClearance" parameterType="StockClearance">
    <selectKey keyProperty="stockClearCode" resultType="String" order="BEFORE">
        /* 상품 자동 증가 코드 */
        SELECT
            case
            when COUNT(stock_clear_code) = 0 then 'sc_code_1'
            ELSE
            CONCAT('sc_code ',
                max(cast(SUBSTRING_INDEX(stock_clear_code, 'sc_code_', -1) AS UNSIGNED))+1
            )
            END AS stockClearCode
        FROM
            stock_clearance;
    </selectKey>
    INSERT INTO stock_clearance
    (stock_clear_code, stock_code, pre_inve_qty, post_inve_qty, adnormal_qty, adnormal_checked,
    adnormal_reason, reg_date)
    VALUES (#{stockClearCode}, #{stockCode}, #{preInveQty}, #{postInveQty}, #{adnormalQty},
    #{adnormalChecked}, #{adnormalReason}, CURDATE())
</insert>

<update id="updateStockClearance" parameterType="PharmStock">
    UPDATE pharm_stock
    SET
        phar_qty=#{pharQty},
        stock_checked='조사함',
        last_date=CURDATE()
    WHERE stock_code=#{stockCode}
</update>

<update id="updateStockClear" parameterType="PharmStock">
    UPDATE pharm_stock
    SET
        stock_checked='조사함',
        last_date=CURDATE()
    WHERE stock_code=#{stockCode}
</update>
</mapper>

```

주문 상세

상품명	단가	주문수량	금액	삭제
 알로페데정 1mg	3500	1	3500	🗑️
 도페린정 5mg	2700	2	5400	🗑️
 후시딘연고	3000	5	15000	🗑️
 우루사정 100mg	3500	5	17500	🗑️
 타이레놀정 500mg	3000	7	21000	🗑️

← 돌아가기

합계	62400 원
배송비	무료
도착지	전북특별자치도 전주시 중화산동 중화빌딩123호
결제요금	62400 원
결제하기 →	

현재 카카오페이로만 결제가 가능합니다 추후 더많은 결제 서비스가 지원 될 예정입니다.



약수터 | 대표 김약수 | 사업자 등록 번호 : 123-34-34567 통신판매업 신고번호 : 제 2024-전북전주-000000호 [사업자 정보 확인]
 전라북도 전주시 덕진구 123 | 대표전화 : 1000-1234 | 문의: admin@yaksutor.com
 개인정보 처리방침 | 이용약관
 © 2024 YAKSUTOR

약국 사용자 - 주문 상세 페이지

- 나의 주문 목록에서 원하는 주문을 클릭시 해당 주문 상세 페이지로 이동한다.
- 화면에 하나의 주문에 담긴 모든 의약품들의 정보가 표시된다.
- 화면 우측에 합계 주문 금액과 배송비가 표시된다.
- 결제하기 버튼을 클릭시 카카오간편결제 창이 팝업된다.

■ Html(javascript)

```
<!DOCTYPE html>
```

```

<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorate="~/user/pharmacy/layout/default">
<head>
<!-- Add necessary head elements here -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

</head>

<th:block layout:fragment="customContent">
  <div class="container mb-80 mt-50">
    <div class="row">
      <div class="col-lg-8 mb-40">
        <h1 class="heading-2 mb-10" th:text="{content}"></h1>
      </div>
    </div>
    <div class="row">
      <div class="col-lg-8">
        <div class="table-responsive shopping-summery">
          <table class="table table-wishlist">
            <thead>
              <tr class="main-heading text-center">
                <th class="custome-checkbox start pl-30"></th>
                <th scope="col" colspan="2">상품명</th>
                <th scope="col">단가</th>
                <th scope="col">주문수량</th>
                <th scope="col">금액</th>
                <th scope="col" class="end">삭제</th>
              </tr>
            </thead>
            <tbody>
              <tr class="pt-30" th:each="od : ${orderDetailList}"
                th:if="{orderDetailList != null}"
                th:attr="data-price=${od.sellMedicine.mediPrice}, data-quantity=${od.orderQty}, data-order-detail-

```

```

code=${od.orderDetailCode}" >

    <td class="custome-checkbox pl-30" > </td>

    <td class="image product-thumbnail pt-40" >  </td>

    <td class="product-des product-name text-center" >

        <h6 class="mb-5" > <a class="product-name mb-10 text-heading"
th:text="${od.sellMedicine.mediName}" > </a> </h6>

        </td>

    <td class="price text-center" data-title="Price" >

        <h4 class="text-body" th:text="${od.sellMedicine.mediPrice}" > </h4>

    </td>

    <td class="price text-center" data-title="Stock" >

        <h4 class="text-body" th:text="${od.orderQty}" > </h4>

    </td>

    <td class="price text-center" data-title="Price" >

        <h4 class="text-brand" th:text="${od.orderPrice}" > </h4>

    </td>

    <td class="action text-center" data-title="Remove" >

        <a href="#" class="text-body delete-order-detail" >

            <i class="fi-rs-trash" > </i> 삭제

        </a>

    </td>

</tr>

</tbody>

</table>

</div>

<div class="divider-2 mb-30" > </div>

<div class="cart-action d-flex justify-content-between" >

    <a class="btn" th:href="@{/pharm/myOrderList}" style="background-color: rgba(0,0,0,0.85)" > <i class="fi-rs-
arrow-left mr-10" > </i> 돌아가기 </a>

    <a th:if="${purchaseState == '결제완료'}" class="btn mr-10 mb-sm-15"
th:href="@{/pharm/myRefundInsert(orderCode=${orderCode})}" style="background-color: rgba(0,0,0,0.85)" > <i class="fi-rs-
refresh mr-10" > </i> 반품하기 </a>

</div>

</div>

<div class="col-lg-4" >

```

```

<div class="border p-md-4 cart-totals ml-30">
  <div class="table-responsive">
    <table class="table no-border">
      <tbody>
        <tr>
          <td class="cart_total_label">
            <h6 class="text-muted">합계</h6>
          </td>
          <td class="cart_total_amount">
            <h4 class="text-muted text-end" id="totalSum"></h4>
          </td>
        </tr>
        <tr>
          <td scope="col" colspan="2">
            <div class="divider-2 mt-10 mb-10"></div>
          </td>
        </tr>
        <tr>
          <td class="cart_total_label">
            <h6 class="text-muted">배송비</h6>
          </td>
          <td class="cart_total_amount">
            <h6 class="text-muted text-end">무료</h6>
          </td>
        </tr>
        <tr>
          <td class="cart_total_label">
            <h6 class="text-muted">도착지</h6>
          </td>
          <td class="cart_total_amount">
            <h6 class="text-muted text-end" th:each="m : ${memberInfo}" th:if="${memberInfo}"
              th:utext="${m.memberAddrSido} + ' ' + ${m.memberAddrSigungu} + '<br>' +
                ${m.memberAddrDong} + ' ' + ${m.memberAddr}"></h6>
          </td>
        </tr>
      </tbody>
    </table>
  </div>

```

```

        <tr>
            <td scope="col" colspan="2">
                <div class="divider-2 mt-10 mb-10"> </div>
            </td>
        </tr>
    </tr>
    <tr>
        <td class="cart_total_label">
            <h6 class="text-muted">결제요금 </h6>
        </td>
        <td class="cart_total_amount">
            <h4 class="text-end" id="totalSum2"> </h4>
        </td>
    </tr>
</tbody>
</table>
</div>
<a class="btn mb-20 w-100" style="background-color: rgba(0,0,0,0.85)" id="paymentButton" th:href="@{#}"
>결제하기 <i class="fi-rs-sign-out ml-15"> </i> </a>
<p>현재 카카오페이로만 결제가 가능합니다 추후 더많은 결제 서비스가 지원 될 예정입니다.</p>
</div>
</div>
</div>
</div>
</th:block>
<th:block layout:fragment="customJsFile">
    <!-- Add necessary JS files here -->
    <script src="https://cdn.portone.io/v2/browser-sdk.js"> </script>
</th:block>
<th:block layout:fragment="customJs">
    <script th:inline="javascript">
        $(document).ready(function() {
            const orderCode = /*[[${orderCode}]]*/;
            console.log(orderCode);
        });
    </script>

```

```

// 삭제 버튼 클릭 이벤트 핸들러 설정
$(document).on('click', '.delete-order-detail', function(event) {
    event.preventDefault(); // 링크 기본 동작 방지

    const row = $(this).closest('tr.pt-30');
    const orderDetailCode = row.data('order-detail-code');

    if (!orderDetailCode) {
        console.error('Order detail code not found');
        return;
    }

    $.ajax({
        url: `/pharm/deleteOrderDetail/${orderDetailCode}`,
        type: 'DELETE',
        success: function(result) {
            row.remove();
            updateTotalSum();
        },
        error: function(xhr, status, error) {
            console.error('Failed to delete order detail:', error);
        }
    });
});

// 합계 업데이트 함수
function updateTotalSum() {
    let totalSum = 0;
    $('tr.pt-30').each(function() {
        const price = parseInt($(this).data('price'), 10);
        const quantity = parseInt($(this).data('quantity'), 10);
        if (!isNaN(price) && !isNaN(quantity)) {
            totalSum += price * quantity;
        }
    });
}

```

```

    });

    $('#totalSum').text(`${totalSum} 원`);
    $('#totalSum2').text(`${totalSum} 원`);
}

// 초기 합계 계산
updateTotalSum();

$('#paymentButton').on('click', function(event) {
    event.preventDefault();

    const totalAmount = parseInt($('#totalSum2').text().replace(' 원', ''), 10);
    const orderName = '약수터 의약품 구매'; // 실제 주문명으로 변경 필요
    const merchantUid = orderCode; // model 에서 받은 orderCode 사용

    // 포트원 결제 데이터 설정
    const data = {
        storeId: 'store-f4751201-4959-40a4-9738-f58651a60e01',
        channelKey: 'channel-key-75b2b89c-d62f-4285-bbc7-d378c03b94dc',
        name: orderName, // 주문명
        totalAmount: totalAmount, // 결제 금액
        paymentId: merchantUid, // 고유 주문 ID
        orderName: orderName, // 주문명 추가
        currency: 'KRW', // 통화 단위
        payMethod: "EASY_PAY",
        easyPay: {
            easyPayProvider: "KAKAOPAY"
        }
    };

    requestPayment(data);
});

async function requestPayment(data) {
    try {

```



```

        console.log("Requesting payment with data: ", data);

        const response = await PortOne.requestPayment(data);
        console.log("Payment response: ", response);
        if (response.code != null) {
            // 오류 발생
            return alert(response.message);
        }
        const postData = {
            merchantUid: data.paymentId,
            paidAmount: data.totalAmount
        };

        const validation = $.ajax({
            async : false,
            url: "/pharm/paymentComplete",
            method: "POST",
            contentType: "application/json",
            data: JSON.stringify(postData),
            dataType: 'json'
        });
    } catch (error) {
        console.error("Error during payment process: ", error);
        alert("결제 처리 중 오류가 발생했습니다.");
    }
}
});
</script>
</th:block>
</html>

```

■ Controller

```

package team01.yaksutor.pharmacy.controller;

import jakarta.servlet.http.HttpServletRequest;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;

```

```

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import team01.yaksutor.common.mapper.MemberMapper;
import team01.yaksutor.dto.Member;
import team01.yaksutor.dto.OrderDetail;
import team01.yaksutor.dto.ShoppingCart;
import team01.yaksutor.pharmacy.service.PhOrderService;
import java.util.List;

@Controller
@RequestMapping("/pharm")
@RequiredArgsConstructor
@Slf4j
public class PhOrderController {

    private final PhOrderService phOrderService;
    private final MemberMapper memberMapper;

    /**
     * 주문상세코드를 이용해 주문상세를 삭제하는 기능
     * @param orderDetailCode
     * @return ResponseEntity
     */
    @DeleteMapping("/deleteOrderDetail/{orderDetailCode}")
    public ResponseEntity<Void> deleteOrderDetail(@PathVariable String orderDetailCode) {
        boolean isDeleted = phOrderService.deleteOrderDetail(orderDetailCode);
        if (isDeleted) {
            return ResponseEntity.ok().build();
        } else {
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).build();
        }
    }

    /**
     * 장바구니로부터 주문을 생성
     * @param shoppingCart
     * @return ResponseEntity
     */
    @PostMapping("/OrderPay")
    public ResponseEntity<String> OrderPay(@RequestBody List<ShoppingCart> shoppingCart) {
        log.info(shoppingCart.toString());

        try {
            phOrderService.submitOrder(shoppingCart);
            return ResponseEntity.ok("주문이 성공적으로 등록되었습니다.");
        } catch (Exception e) {
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("정상적인 주문경로가
아닙니다.");
        }

    }

    /**
     * 주문상세 페이지로 이동

```

```

* @param oCode
* @param purchaseState
* @param model
* @return url 주소
*/
@GetMapping("/myOrderDetailView")
public String myOrderDetailView(@RequestParam("oCode") String oCode,
                               @RequestParam("purchaseState") String purchaseState,
                               HttpServletRequest request,
                               Model model) {

    String sId = (String) request.getSession().getAttribute("S_ID");
    Member memberInfo = memberMapper.getMemberInfoById(sId);
    List<OrderDetail> orderDetailList = phOrderService.getOrderDetailListByOCode(oCode);
    String orderCode = oCode;

    model.addAttribute("purchaseState", purchaseState);
    model.addAttribute("orderCode", orderCode);
    model.addAttribute("title", "주문 상세");
    model.addAttribute("content", "주문 상세");
    model.addAttribute("orderDetailList", orderDetailList);
    model.addAttribute("memberInfo", memberInfo);

    return "user/pharmacy/myPage/myOrderDetailView";
}

@GetMapping("/paymentMethod")
public String paymentMethod(Model model) {

    model.addAttribute("title", "결제수단 등록");
    model.addAttribute("content", "결제수단 등록");

    return "user/pharmacy/order/paymentMethod";
}
}

```

■ Service

```

package team01.yaksutor.pharmacy.service;

import jakarta.servlet.http.HttpServletRequest;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import team01.yaksutor.dto.Order;
import team01.yaksutor.dto.OrderDetail;
import team01.yaksutor.dto.SellerInfo;
import team01.yaksutor.dto.ShoppingCart;

```

```

import team01.yaksutor.pharmacy.mapper.PhOrderMapper;

import java.util.List;

@Service
@RequiredArgsConstructor
@Slf4j
public class PhOrderService {

    private final PhOrderMapper phOrderMapper;
    private final HttpServletRequest request;

    @Transactional
    public void submitOrder(List<ShoppingCart> shoppingCart) {

        //주문생성
        String sessionId = (String) request.getSession().getAttribute("S_ID");
        Order order = new Order();
        order.setMIdOrder(sessionId);
        order.setShoppingCartGroup("group1");
        order.setDeliveryState("배송대기");
        order.setPurchaseState("결제준비중");
        order.setPaymentMethod("카카오페이");
        order.setRequestTerm("파손주의");
        order.setOrderType("주문");

        //생성한 주문 DB 에 등록
        phOrderMapper.addOrder(order);
        String oCode = order.getOCode();
        //화면으로 부터 넘겨받은 장바구니 데이터를 돌면서 주문상세에 저장
        int totalPrice = 0;
        for (ShoppingCart cart : shoppingCart) {
            SellerInfo sellerInfo = phOrderMapper.getSellerInfoByGoodsCode(cart.getGoodsCode());
            OrderDetail orderDetail = new OrderDetail();
            orderDetail.setOCode(oCode);
            orderDetail.setGoodsCode(cart.getGoodsCode());
            orderDetail.setOSellerId(sellerInfo.getSuppld());
            orderDetail.setSuppCode(sellerInfo.getSuppCode());
            orderDetail.setOrderQty(cart.getSumQty());
            orderDetail.setOrderPrice((cart.getSumQty() * cart.getMediPrice()));
            System.out.println(orderDetail);
            phOrderMapper.addOrderDetail(orderDetail);
        }
    }
}

```

```

        totalPrice = totalPrice + cart.getSumQty() * cart.getMediPrice();
    }

    order.setOrderTotalPrice(totalPrice);
    System.out.println(order);
    phOrderMapper.updateOrderPrice(order);

}

public List<Order> getOrderListById() {
    String sessionId = (String) request.getSession().getAttribute("S_ID");
    return phOrderMapper.getOrderListById(sessionId);
}

public List<OrderDetail> getOrderDetailListByOCode(String oCode) {
    return phOrderMapper.getOrderDetailListByOCode(oCode);
}

@Transactional
public boolean deleteOrderDetail(String orderDetailCode) {
    try {
        // 삭제할 OrderDetail 을 먼저 조회
        OrderDetail orderDetail = phOrderMapper.getOrderDetailByCode(orderDetailCode);
        if (orderDetail == null) {
            return false;
        }
        // OrderDetail 을 삭제
        phOrderMapper.deleteOrderDetail(orderDetailCode);

        // 해당 주문 코드(oCode)로 남아 있는 OrderDetail 목록을 조회
        List<OrderDetail> remainingOrderDetails =
phOrderMapper.getOrderDetailListByOCode(orderDetail.getOCode());

        // 남아 있는 OrderDetail 의 총 금액을 다시 계산
        int totalPrice = remainingOrderDetails.stream()
            .mapToInt(OrderDetail::getOrderPrice)
            .sum();

        if (remainingOrderDetails.isEmpty()) {
            // 남아 있는 OrderDetail 이 없으면 Order 를 삭제

```

```

        phOrderMapper.deleteOrder(orderDetail.getOCode());
    } else {
        // Order 객체를 업데이트
        Order order = new Order();
        order.setOCode(orderDetail.getOCode());
        order.setOrderTotalPrice(totalPrice);

        // Order 테이블의 총 주문 금액을 업데이트
        phOrderMapper.updateOrderPrice(order);
    }
    return true;
} catch (Exception e) {
    return false;
}
}

@Transactional
public void updateOrderStatus(String merchantUid, String status) {
    phOrderMapper.updateOrderStatus(merchantUid, status);
}
}

```

■ Mapper(SQL)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="team01.yaksutor.pharmacy.mapper.PhOrderMapper">

    <resultMap id="OrderResultMap" type="Order">
        <id column="o_code" property="oCode" />
        <result column="m_id_order" property="mIdOrder" />
        <result column="shopping_cart_group" property="shoppingCartGroup" />
        <result column="order_total_price" property="orderTotalPrice" />
        <result column="delivery_state" property="deliveryState" />
        <result column="purchase_state" property="purchaseState" />
        <result column="payment_method" property="paymentMethod" />
        <result column="request_term" property="requestTerm" />
        <result column="order_type" property="orderType" />
        <result column="purchase_date" property="purchaseDate" />
    </resultMap>

    <resultMap id="OrderDetailResultMap" type="OrderDetail">
        <id column="order_detail_code" property="orderDetailCode"/>
        <result column="o_code" property="oCode"/>
        <result column="goods_code" property="goodsCode"/>
        <result column="o_seller_id" property="oSellerId"/>
        <result column="supp_code" property="suppCode"/>
        <result column="order_qty" property="orderQty"/>
        <result column="order_price" property="orderPrice"/>
    </resultMap>

```

```

<association property="sellMedicine" javaType="SellMedicine">
  <id column="goods_code" property="goodsCode"/>
  <result column="supp_code" property="suppCode"/>
  <result column="medi_code" property="mediCode"/>
  <result column="medi_name" property="mediName"/>
  <result column="medi_detail" property="mediDetail"/>
  <result column="medi_price" property="mediPrice"/>
  <result column="medi_img" property="mediImg"/>
  <result column="reg_qty" property="regQty"/>
  <result column="state" property="state"/>
  <result column="reg_date" property="regDate"/>
</association>

<association property="imgData" javaType="ImgData">
  <id column="file_idx" property="fileIdx"/>
  <result column="file_origin_name" property="fileOriginName"/>
  <result column="file_new_name" property="fileNewName"/>
  <result column="file_path" property="filePath"/>
  <result column="file_size" property="fileSize"/>
  <result column="file_reg_date" property="fileRegDate"/>
</association>
</resultMap>

<!-- 결제완료시 결제상태 업데이트 -->
<update id="updateOrderStatus" parameterType="String">
  UPDATE `order`
  SET
  purchase_state=#{status}
  WHERE
  o_code = #{merchantUid}
</update>
<!-- 주문코드로 주문삭제 -->
<delete id="deleteOrder" parameterType="String">
  DELETE FROM `order` WHERE o_code=#{oCode}
</delete>
<!-- 주문코드로 주문상세 조회 -->
<select id="getOrderDetailByCode" parameterType="String" resultMap="OrderDetailResultMap">
  SELECT
  order_detail_code,
  o_code,
  goods_code,
  o_seller_id,
  supp_code,
  order_qty,
  order_price
  FROM
  order_detail
  WHERE
  order_detail_code = #{orderDetailCode};
</select>
<!-- 주문상세 삭제 -->
<delete id="deleteOrderDetail" parameterType="String">
  DELETE FROM order_detail
  WHERE order_detail_code = #{orderDetailCode}
</delete>

<!-- 주문코드로 주문상세조회 -->
<select id="getOrderDetailListByOCode" parameterType="String" resultMap="OrderDetailResultMap">
  SELECT
  i.file_path,
  sm.medi_name,
  sm.medi_price,
  od.order_detail_code,

```

```

od.o_code,
od.goods_code,
od.o_seller_id,
od.supp_code,
od.order_qty,
od.order_price

FROM order_detail as od
INNER JOIN
sell_medicine AS sm
ON
od.goods_code = sm.goods_code
INNER join
img_data AS i
ON
sm.medi_img = i.file_idx
WHERE
od.o_code = #{oCode}
</select>

<!-- 주문자 ID 로 주문내역을 조회 -->
<select id="getOrderListById" parameterType="String" resultMap="OrderResultMap">
SELECT
o_code,
m_id_order,
shopping_cart_group,
order_total_price,
delivery_state,
purchase_state,
payment_method,
request_term,
order_type,
purchase_date
FROM `order` AS o
WHERE o.m_id_order = #{sessionId}
and
o.purchase_state != '반품'
ORDER BY purchase_date desc;
</select>

<!-- 총주문금액 업데이트 -->
<update id="updateOrderPrice" parameterType="Order">
/*order 테이블 총주문금액 업데이트*/
UPDATE
`order`
SET
order_total_price=#{orderTotalPrice}
WHERE
o_code = #{oCode}
</update>

<!-- 주문상세 추가 -->
<insert id="addOrderDetail" parameterType="OrderDetail">
<selectKey keyColumn="orderDetailCode" keyProperty="orderDetailCode" resultType="String" order="BEFORE">
/* order_detail_code 자동증가 코드 */
SELECT
CASE
WHEN COUNT(order_detail_code) = 0 THEN 'od_code_1'
ELSE CONCAT(
'od_code_',
MAX(CAST(SUBSTRING_INDEX(order_detail_code, '_', -1) AS UNSIGNED)) + 1
)
END AS orderDetailCode
FROM

```



```

        order_detail
    </selectKey>
    INSERT INTO order_detail
    (order_detail_code, o_code, goods_code, o_seller_id, supp_code, order_qty, order_price)
    VALUES ({orderDetailCode}, #{oCode}, #{goodsCode}, #{oSellerId}, #{suppCode}, #{orderQty}, #{orderPrice})
</insert>

<!-- 주문추가 -->
<insert id="addOrder" parameterType="Order">
    <selectKey keyProperty="oCode" resultType="String" order="BEFORE">
        /* o_code 자동증가 코드 */
        SELECT
        CASE
        WHEN COUNT(o_code) = 0 THEN 'o_code_1'
        ELSE CONCAT(
        'o_code_',
        MAX(CAST(SUBSTRING_INDEX(o_code, '_', -1) AS UNSIGNED)) + 1
        )
        END AS oCode
        FROM
        `order`;
    </selectKey>
    INSERT INTO `order`
    (o_code, m_id_order, shopping_cart_group, order_total_price, delivery_state, purchase_state, payment_method,
    request_term, order_type, purchase_date)
    VALUES ({oCode}, #{mIdOrder}, #{shoppingCartGroup}, 0, #{deliveryState}, #{purchaseState}, #{paymentMethod},
    #{requestTerm}, #{orderType}, CURDATE())
</insert>

<!-- 상품코드로부터 판매자정보 불러오기 -->
<select id="getSellerInfoByGoodsCode" resultType="SellerInfo">
    SELECT
    s.suppl_id AS 'suppld',
    sm.supp_code AS 'suppCode'
    FROM
    sell_medicine AS sm
    INNER JOIN
    supplier AS s
    ON
    sm.supp_code = s.supp_code
    WHERE
    sm.goods_code = #{goodsCode};
</select>
</mapper>

```

커뮤니티

총 14 개의 글이 있습니다.

🔍 글 검색

- 안녕하세요 날짜별로 정렬되는지 확인해볼래요** 2024-07-19

k0007

확인되었는지?

👍 0 🗨️ 0
- 저는 남용업체입니다.** 2024-07-18

k0003

안녕하세요.

👍 1 🗨️ 0
- adsasdad** 2024-07-19

admin

dsasdad

👍 0 🗨️ 0
- 무료라니 믿을 수 없어요** 2024-07-08

k0007

감사합니다.

👍 1 🗨️ 0
- 날이 흐리네요** 2024-07-08

k0007

비도 살짝 내렸어

👍 0 🗨️ 0
- 텍스트** 2024-07-05

k0007

2222

👍 1 🗨️ 0
- 안녕하세요? 반기워요?** 2024-07-04

admin

네 반가워요?

👍 1 🗨️ 0
- CSS** 2024-07-04

admin

CSS

👍 0 🗨️ 0
- 수정된 커뮤니티 글...** 2024-07-02

admin

이 글에 보인다면 수정완료.....

👍 1 🗨️ 0
- 남용업체 파이팅** 2024-06-18

k0001

불쌍하지만 믿는 죽어갑니다.

👍 1 🗨️ 0
- 판매량 높은 의약품** 2024-06-14

k0015

어떤 약종의 세입 많이 팔지나요?

👍 1 🗨️ 0
- 남용하기 너무 편해요** 2024-06-13

k0003

감사합니다.

👍 0 🗨️ 0
- 긴급 의약품 거래** 2024-06-12

k0008

내일 필요한 거 구매할수있나?

👍 2 🗨️ 0
- 약수터 너무 좋아요** 2024-06-11

k0006

좋은 정보들이 많이 나와서 정말 유용합니다.....

👍 5 🗨️ 0



약국 사용자 페이지 - 커뮤니티

- 플랫폼을 이용하는 모든 사용자가 커뮤니티를 이용할 수 있다.
- 사용자는 글쓰기 버튼을 통해 글을 등록 할 수 있다..
- 화면에 커뮤니티에 등록된 글들이 표시된다
- 해당 게시글에 댓글을 등록하고 수정 삭제 할 수 있다.
- 글과 댓글은 해당 세션의 ID 와 일치해야만 수정이나 삭제가 가능하다

■ Html(javascript)

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout" xmlns="http://www.w3.org/1999/html"
      layout:decorate="~{user/pharmacy/layout/default}">
<head>
  <style>
    .content-hidden {
      display: none;
    }
    .content-hidden2 {
      display: none;
    }
  </style>
</head>

<th:block layout:fragment="customContent">
  <div class="container mb-30 mt-50 font-1">
    <div class="row">
      <div class="col-xl-10 col-lg-12 m-auto">
        <div class="mb-50">
          <h1 class="heading-2 mb-10 font-1">커뮤니티</h1>
          <h2 class="text-body" style="font-family: 'Pretendard-Regular';">총
            <span class="text-brand text-info"
              th:each="b, i : ${boardList}"
              th:if="${i.index} < 1"
              th:text="${i.size}">
            </span> 개의 글이 있습니다.</h2>
          <a class="btn btn-md mt-3" data-toggle="modal" data-target="#writeBoardModal" style="background-color:
            #ff8c9f; font-family: 'Pretendard-Regular'; border-radius: 15px;">
            □글쓰기
          </a>
        </div>
      <div class="card mb-4">
        <div class="card-body">
          <article class="itemlist m-3 border-bottom" th:if="${boardList != null}" th:each="b : ${boardList}">
```

```

<div class="row align-items-center">
  <div class="col-lg-4 col-sm-4 col-8 flex-grow-1 col-name">
    <div class="col-lg-12 comments-review">
      <div class="user-comments-profile">
        <div style="display: inline-block;">
          
        </div>
        <div style="display: inline-block; width: auto; height: 100%; color: black; font-weight: bold;
margin-left: 15px; top: -20px; position: relative">
          <span class="font-1 board-title" th:text="{b.boardTitle}" > </span> <i class="board-date"
th:text="{b.boardDate}" > </i>
        </div>
        <br>
        <div style="display: inline-block; width: auto; height: 100%; color: black; font-weight: bold;
top: 10%">
          <p th:text="{b.mld}" > </p>
        </div>
      </div>
    <div class="col-lg-12" style="height: 10px"> </div>
    <div style="background: #f9fafb; border-radius: 25px; padding: 10px; min-height: 60px; margin-
bottom: 10px;">
      <div>
        <div class="donation-comments" style="color: black; font-size: 16px;"
th:text="{b.boardContent}" >
        </div>
      </div>
      </div>
      <div class="reply" style="display: inline-block; cursor: pointer">
        <i class="bi bi-heart" style="color: red; margin-right: 10px; position: relative; top: 2px; font-
size: 23px"> </i>
        <span class="bi bi-chat" th:text="'' + {b.replSize}" > </span>
        <mark class="bi bi-three-dots" style="color: black; font-size: 23px; background:none;
margin-left: 10px" > </mark>
      </div>
      <div class="comlist-menu">
        <a class="comlist-edit" style="color: black" data-toggle="modal" data-
target="#editBoardModal"
          th:data-board-id="{b.boardCode}"
          th:data-board-title="{b.boardTitle}"
          th:data-board-content="{b.boardContent}"
          th:data-board-mid="{b.mld}"
          th:data-session-id="{session.S_ID}" > <i class="bi bi-pen"> </i> 수정 </a>
        <div style="border-bottom: 1px solid #f0f5f5"> </div>
        <a class="comlist-delete" style="color:red" data-toggle="modal" data-
target="#deleteBoardModal"

```

```

th:data-board-id="{b.boardCode}"
th:data-session-id="{session.S_ID}"
th:data-board-mid="{b.mld}" > <i class="bi bi-trash" > </i> 삭제 </a>
</div>

<div class="comments-reply" style="width: 100%; display: none">
  <div style="display: flex; margin-bottom: 10px; margin-top: 12px">
    <div style="display: inline-block; top: 0; margin-right: 5px; margin-bottom: 5px;">
      
    </div>

    <form id="addReplForm" th:action="@{/pharm/replAdd}" method="post" style="display:
flex">
      <textarea id="replContent" name="replContent" class="replyArea" minlength="1"
maxlength="200" th:placeholder="{b.mld} + ' 님에게 댓글 추가...' required"> </textarea>
      <input type="text" class="form-control" id="mld" name="mld"
th:value="{session.S_ID}" hidden="hidden" >
      <input type="text" class="form-control" id="cmNum" name="cmNum"
th:value="{b.boardCode}" hidden="hidden">
      <button type="submit" class="btn btn-success" id="replyBtn" > <i class="bi bi-cursor
replyCursor" > </i> </button>
    </form>
  </div>

  <div class="reply" style="display: flex; margin-bottom: 15px" th:each="r : ${replList}"
th:if="{r.cmNum} == {b.boardCode}">
    <div style="top: 0; margin-right: 5px;">
      
      <br>
      <p th:text="{r.mld}" style="font-size: 0.78rem;" > </p>
    </div>
    <div style="background: #f9fafb; border-radius: 25px; padding: 10px; width: 40%">
      <div>
        <div class="reply-user-id" style="font-size: 12px" th:text="{r.replDate}" > </div>
        <div class="donation-comments" style="color: black; font-size: 15px"
th:text="{r.replContent}" > </div>
        <div th:if="{r.mld} == {session.S_ID}" > <a class="replDelete" style="color: red"
data-toggle="modal" data-target="#deleteReplModal"
th:data-repl-id="{r.replNum}" > 삭제 </a> </div>
      </div>
    </div>
  </div>
</div>

```

```

        <div class="col-lg-12" style="height: 20px"> </div>
    </div>
</div>
</div>
</div>
<!-- row ./ -->
</article>
</div>
<!-- card-body end// -->
</div>
</div>
</div>
</div>
<!-- 작성 모달 -->
<div class="modal fade" id="writeBoardModal" tabindex="-1" aria-labelledby="writeBoardModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content community-modal">

            <div class="modal-body">
                <h5 class="modal-title font-1 m-1" id="writeBoardModalLabel">글 남기기</h5>

                <form id="addBoardForm" method="post" th:action="@{/pharm/boardAdd}">
                    <textarea class="form-control commodal-title mb-3" id="boardTitle" name="boardTitle"
placeholder="제목을 입력하세요."
                    required type="text"> </textarea>
                    <textarea class="form-control commodal-content" id="boardContent" name="boardContent"
placeholder="내용을 입력하세요." required
                    rows="4"> </textarea>
                    <input class="form-control" hidden="hidden" id="mId" name="mId" th:value="{session.S_ID}">
                    <input hidden="hidden" id="boardView" name="boardView" value="0">
                    <div class="mt-3 commodal-info">
                        <p class="font-1" style="font-size: 14px">
                            □최근 제 3 자에 대한 허위사실 유포, 비방 목적의 댓글로 인해 당사자간 법적분쟁이 발생한 사례가
증가하고 있습니다.
                            약의적 댓글 작성자는 명예훼손, 모욕 등으로 법적 책임을 부담하게 될 수 있다는 점을 유의하여 주시기
바랍니다.
                        </p>
                    </div>
                    <button class="btn btn-success mt-3" id="commodal-btn" type="submit">글쓰기</button>
                </form>
            </div>
        </div>
    </div>
</div>
<!-- 글 수정 모달 -->
<div class="modal fade" id="editBoardModal" tabindex="-1" aria-labelledby="editBoardModalLabel" aria-hidden="true">

```

```

<div class="modal-dialog modal-dialog-centered">
  <div class="modal-content community-modal">

    <div class="modal-body">
      <!-- 권한 없음 메시지 -->
      <div id="editBoardError" class="content-hidden" role="alert">
        <p style="color: black"> □작성자인 <i style="color: blue">수정</i>할 수 있습니다.</p>
        <button type="button" class="btn-modal1 m-1" data-dismiss="modal" aria-label="Close" style="width:
100%">알겠어요.</button>
      </div>

      <form id="editBoardForm" method="post" th:action="@{/pharm/board/modify}">
        <h5 class="modal-title font-1 m-1" id="editBoardModalLabel">글 수정하기</h5>

        <input type="hidden" id="editBoardId" name="boardCode">
        <input type="hidden" id="editBoardmId" name="mId">
        <input type="hidden" id="sessionId">

        <textarea class="form-control commodal-title mb-3" id="editBoardTitle" name="boardTitle"
placeholder="제목을 입력하세요."
          required type="text"> </textarea>
        <textarea class="form-control commodal-content" id="editBoardContent" name="boardContent"
placeholder="내용을 입력하세요." required
          rows="4"> </textarea>
        <div class="mt-3 commodal-info">
          <p class="font-1" style="font-size: 14px">
            □최근 제 3 자에 대한 허위사실 유포, 비방 목적의 댓글로 인해 당사자간 법적분쟁이 발생한 사례가
증가하고 있습니다.
            약의적 댓글 작성자는 명예훼손, 모욕 등으로 법적 책임을 부담하게 될 수 있다는 점을 유의하여 주시기
바랍니다.
          </p>
        </div>
        <button class="btn btn-success mt-3" id="editCommodalBtn" type="submit">수정하기</button>
      </form>
    </div>

  </div>

</div>
<!-- 커뮤니티 삭제 확인 모달 -->
<div class="modal fade" id="deleteBoardModal" tabindex="-1" aria-labelledby="deleteBoardModalLabel" aria-
hidden="true">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content community-modal">

      <div class="modal-body">
        <!-- 권한 없음 메시지 -->

```

```

<div id="deleteBoardError" class="content-hidden2" role="alert">
  <p style="color: black"> □작성자만 <i style="color: red">삭제</i> 할 수 있습니다. </p>
  <button type="button" class="btn-modal1 m-1" data-dismiss="modal" aria-label="Close" style="width:
100%">알겠어요.</button>
</div>

<form id="deleteBoardForm" method="post" th:action="@{/pharm/board/delete}">
  <h5 class="modal-title font-1 m-1" id="deleteBoardModalLabel">정말로 삭제할까요?</h5>
  <div class="m-3 commodal-info">
    <p class="font-1" style="font-size: 14px">
      □삭제된 글은 복구할 수 없습니다.
    </p>
  </div>
  <input type="hidden" id="deleteBoardId" name="boardCode">
  <input type="hidden" id="sessionId2">
  <input type="hidden" id="deleteBoardmId">

  <button type="submit" class="deleteBtnOk" style="background-color: #ff8c9f; border:
none">삭제</button> <br>
  <button type="button" class="deleteBinNo" data-dismiss="modal" aria-label="Close">취소</button>
</form>
</div>

</div>
</div>
</div>
<!-- 댓글 삭제 확인 모달 -->
<div class="modal fade" id="deleteReplModal" tabindex="-1" aria-labelledby="deleteReplModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content community-modal">

      <div class="modal-body">
        <form id="deleteReplForm" method="post" th:action="@{/pharm/replDelete}">
          <h5 class="modal-title font-1 m-1" id="deleteReplModalLabel">정말로 삭제할까요?</h5>
          <div class="m-3 commodal-info">
            <p class="font-1" style="font-size: 14px">
              □삭제된 글은 복구할 수 없습니다.
            </p>
          </div>
          <input type="hidden" id="deleteReplId" name="replNum">

          <button type="submit" class="deleteBtnOk" style="background-color: #ff8c9f; border:
none">삭제</button> <br>
          <button type="button" class="deleteBinNo" data-dismiss="modal" aria-label="Close">취소</button>
        </form>
      </div>
    </div>
  </div>
</div>

```



```

    </div>
</div>

</th:block>

<th:block layout:fragment="customJsFile">
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</th:block>

<th:block layout:fragment="customJs">
    <script>
        $(document).ready(function () {
            $('.bi-chat').on('click', function () {
                $(this).closest('.comments-review').find('.comments-reply').slideToggle();
            });
        });

        var i = 0;
        $('.reply span').on('click',function(){
            if(i==0){
                $(this).attr('class','bi bi-chat-fill');
                i++;
            }else if(i==1){
                $(this).attr('class','bi bi-chat');
                i--;
            }
        });

        $('.reply i').on('click',function(){
            if(i==0){
                $(this).attr('class','bi bi-heart-fill');
                i++;
            }else if(i==1){
                $(this).attr('class','bi bi-heart');
                i--;
            }
        });
        $(document).ready(function() {
            $('.reply mark').on('click', function() {
                $(this).closest('.comments-review').find('.comlist-menu').slideToggle();
            });
        });
    </script>

```

```

// 글 수정 버튼 클릭 시 모달 창에 데이터 채우기
$('.comlist-edit').on('click', function() {
    const boardId = $(this).data('board-id');
    const boardTitle = $(this).data('board-title');
    const boardContent = $(this).data('board-content');
    const boardMid = $(this).data('board-mid');
    const sessionId = $(this).data('session-id');
    if(boardMid === sessionId) {
        $('#editBoardId').val(boardId);
        $('#editBoardTitle').val(boardTitle);
        $('#editBoardContent').val(boardContent);
        $('#editBoardMid').val(boardMid);
        $('#sessionId').val(sessionId);
        $('#editBoardError').addClass('content-hidden');
        $('#editBoardForm').show();
    } else{
        $('#editBoardError').removeClass('content-hidden');
        $('#editBoardForm').hide();
    }
});

// 글 삭제 버튼 클릭 시 모달 창에 데이터 채우기
$('.comlist-delete').on('click', function() {
    const boardId2 = $(this).data('board-id');
    const sessionId2 = $(this).data('session-id');
    const boardMid2 = $(this).data('board-mid');
    if(boardMid2 === sessionId2) {
        $('#deleteBoardId').val(boardId2);
        $('#sessionId2').val(sessionId2);
        $('#deleteBoardMid').val(boardMid2);
        $('#deleteBoardError').addClass('content-hidden2');
        $('#deleteBoardForm').show();
    } else{
        $('#deleteBoardError').removeClass('content-hidden2');
        $('#deleteBoardForm').hide();
    }
});

// 댓글 삭제 버튼 클릭 시 모달 창에 데이터 채우기
$('.replDelete').on('click', function() {
    const replNum = $(this).data('repl-id');

    if(replNum != null) {
        $('#deleteReplId').val(replNum);
        $('#deleteReplForm').show();
    } else{
        $('#deleteReplForm').hide();
    }
});

```

```

    }
    });

});
</script>
</th:block>
</html>

```

■ Controller

```

package team01.yaksutor.pharmacy.controller;

import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import team01.yaksutor.pharmacy.dto.*;
import team01.yaksutor.pharmacy.service.PhBoardService;

import java.util.List;

@Controller
@RequiredArgsConstructor
@Slf4j
public class PhBoardController {

    private final PhBoardService phBoardService;

    /* -----
    *           커뮤니티 + 댓글
    * -----*/

    // 커뮤니티 + 댓글 조회 (사용자)
    @GetMapping("/pharm/board")
    public String getBoardList(Model model) {
        List<Board> boardList = phBoardService.getBoardList();
        List<Repl> replList = phBoardService.getReplList();
        model.addAttribute("boardList", boardList);
        model.addAttribute("replList", replList);

        return "user/pharmacy/board/boardList";
    }
    // 커뮤니티 + 댓글 조회 (관리자)
    @GetMapping("/admin/boardList")
    public String getBoardList2(Model model) {
        List<Board> boardList = phBoardService.getBoardList();

```

```

List<Repl> replList = phBoardService.getReplList();
model.addAttribute("boardList", boardList);
model.addAttribute("replList", replList);

return "admin/board/boardList";
}
// 커뮤니티 + 댓글 조회 (납품자)
@GetMapping("/supplier/boardList")
public String getBoardList3(Model model) {
    List<Board> boardList = phBoardService.getBoardList();
    List<Repl> replList = phBoardService.getReplList();
    model.addAttribute("boardList", boardList);
    model.addAttribute("replList", replList);

    return "user/supplier/board/boardList";
}
// 커뮤니티 등록 (사용자)
@PostMapping("/pharm/boardAdd")
public String addBoard(Board board){
    phBoardService.addBoard(board);
    return "redirect:/pharm/board";
}
// 커뮤니티 등록 (납품자)
@PostMapping("/supplier/boardAdd")
public String addBoard3(Board board){
    phBoardService.addBoard(board);
    return "redirect:/supplier/boardList";
}
// 커뮤니티 등록 (관리자)
@GetMapping("/admin/boardAdd")
public String addBoard(Model model) {
    List<Board> boardList = phBoardService.getBoardList();

    model.addAttribute("boardList", boardList);
    return "admin/board/boardAdd";
}
@PostMapping("/admin/boardAdd")
public String addBoard2(Board board){
    phBoardService.addBoard(board);
    return "redirect:/admin/boardList";
}
// 커뮤니티 수정 (사용자)
@PostMapping("/pharm/board/modify")
public String modifyBoard2(Board board){
    phBoardService.modifyBoard(board);
    return "redirect:/pharm/board";
}
// 커뮤니티 수정 (납품자)
@PostMapping("/supplier/board/modify")
public String modifyBoard3(Board board){
    phBoardService.modifyBoard(board);
    return "redirect:/supplier/boardList";
}
// 커뮤니티 수정 (관리자)
@PostMapping("/admin/boardList/modify")
public String modifyBoard(Board board){

```

```

    phBoardService.modifyBoard(board);
    return "redirect:/admin/boardList";
}
// 커뮤니티 삭제 (사용자)
@PostMapping("/pharm/board/delete")
public String deleteBoard2(@RequestParam String boardCode){
    phBoardService.deleteBoard(boardCode);
    return "redirect:/pharm/board";
}
// 커뮤니티 삭제 (납품자)
@PostMapping("/supplier/board/delete")
public String deleteBoard3(@RequestParam String boardCode){
    phBoardService.deleteBoard(boardCode);
    return "redirect:/supplier/boardList";
}
// 커뮤니티 삭제 (관리자)
@PostMapping("/admin/boardList/delete")
public String deleteBoard(@RequestParam String boardCode){
    phBoardService.deleteBoard(boardCode);
    return "redirect:/admin/boardList";
}
// 커뮤니티 댓글 등록 (사용자)
@PostMapping("/pharm/replAdd")
public String addRepl(Repl repl){
    phBoardService.addRepl(repl);
    return "redirect:/pharm/board";
}
}
// 커뮤니티 댓글 등록 (납품자)
@PostMapping("/supplier/replAdd")
public String addRepl2(Repl repl){
    phBoardService.addRepl(repl);
    return "redirect:/supplier/boardList";
}
// 커뮤니티 댓글 등록 (관리자)
@PostMapping("/admin/replAdd")
public String addRepl3(Repl repl){
    phBoardService.addRepl(repl);
    return "redirect:/admin/boardList";
}
}
// 커뮤니티 댓글 수정 (관리자)
@PostMapping("/admin/boardList/modifyRepl")
public String modifyRepl(Repl repl){
    phBoardService.modifyRepl(repl);
    return "redirect:/admin/boardList";
}
// 커뮤니티 댓글 삭제 (사용자)
@PostMapping("/pharm/replDelete")
public String deleteRepl(@RequestParam String replNum){
    phBoardService.deleteRepl(replNum);
    return "redirect:/pharm/board";
}
}
// 커뮤니티 댓글 삭제 (납품자)
@PostMapping("/supplier/replDelete")
public String deleteRepl2(@RequestParam String replNum){
    phBoardService.deleteRepl(replNum);
    return "redirect:/supplier/boardList";
}
}

```

```

}
// 커뮤니티 댓글 삭제 (관리자)
@PostMapping("/admin/boardList/deleteRepl")
public String deleteRepl3(@RequestParam String replNum){
    phBoardService.deleteRepl(replNum);
    return "redirect:/admin/boardList";
}

/* -----
*           자주하는 질문
* -----*/

// 자주하는 질문 조회 (사용자)
@GetMapping("/pharm/faq")
public String getQuestionCenterList(Model model) {
    List<QuestionCenter> questionCenterList = phBoardService.getQuestionCenterList();

    model.addAttribute("questionCenterList", questionCenterList);
    return "user/pharmacy/board/faq";
}
// 자주하는 질문 조회 (납품자)
@GetMapping("/supplier/faq")
public String getQuestionCenterList3(Model model) {
    List<QuestionCenter> questionCenterList = phBoardService.getQuestionCenterList();

    model.addAttribute("questionCenterList", questionCenterList);
    return "user/supplier/board/faq";
}
// 자주하는 질문 조회 (관리자)
@GetMapping("/admin/faqList")
public String getQuestionCenterList2(Model model) {
    List<QuestionCenter> questionCenterList = phBoardService.getQuestionCenterList();

    model.addAttribute("questionCenterList", questionCenterList);
    return "admin/questionCenter/faqList";
}
// 자주하는 질문 등록 (관리자)
@GetMapping("/admin/faqAdd")
public String addQuestionCenter(Model model) {
    List<QuestionCenter> questionCenterList = phBoardService.getQuestionCenterList();

    model.addAttribute("questionCenterList", questionCenterList);
    return "admin/questionCenter/faqAdd";
}
@PostMapping("/admin/faqAdd")
public String addQuestionCenter(QuestionCenter questionCenter){
    phBoardService.addQuestionCenter(questionCenter);
    return "redirect:/admin/faqList";
}
// 자주하는 질문 수정 (관리자)
@PostMapping("/admin/faqList/modify")
public String modifyQuestionCenter(QuestionCenter questionCenter){
    phBoardService.modifyQuestionCenter(questionCenter);
    return "redirect:/admin/faqList";
}
// 자주하는 질문 삭제 (관리자)
@PostMapping("/admin/faqList/delete")

```

```

public String deleteQuestionCenter(@RequestParam String questionNum){
    phBoardService.deleteQuestionCenter(questionNum);
    return "redirect:/admin/faqList";
}

```

```

/* -----공지사항-----*/

```

```

// 공지사항 조회 (사용자)
@GetMapping("/pharm/notice")
public String getNoticeList(Model model) {
    List<Notice> noticeList = phBoardService.getNoticeList();
    model.addAttribute("noticeList", noticeList);
    return "user/pharmacy/board/notice";
}

```

```

// 공지사항 조회 (납품자)
@GetMapping("/supplier/notice")
public String getNoticeList3(Model model) {
    List<Notice> noticeList = phBoardService.getNoticeList();
    model.addAttribute("noticeList", noticeList);
    return "user/supplier/board/notice";
}

```

```

// 공지사항 조회 (관리자)
@GetMapping("/admin/noticeList")
public String getNoticeList2(Model model) {
    List<Notice> noticeList = phBoardService.getNoticeList();
    model.addAttribute("noticeList", noticeList);
    return "admin/notice/noticeList";
}

```

```

// 공지사항 추가 (관리자)
@GetMapping("/admin/noticeAdd")
public String addNotice(Model model){
    List<Notice> noticeList = phBoardService.getNoticeList();
    model.addAttribute("noticeList", noticeList);
    return "admin/notice/noticeAdd";
}

```

```

@PostMapping("/admin/noticeAdd")
public String addNotice(Notice notice){
    phBoardService.addNotice(notice);
    return "redirect:/admin/noticeList";
}

```

```

// 공지사항 수정 (관리자)
@PostMapping("/admin/noticeList/modify")
public String modifyNotice(Notice notice){
    phBoardService.modifyNotice(notice);
    return "redirect:/admin/noticeList";
}

```

```

// 공지사항 삭제 (관리자)
@PostMapping("/admin/noticeList/delete")
public String deleteNotice(@RequestParam String noticeNum){
    phBoardService.deleteNotice(noticeNum);
    return "redirect:/admin/noticeList";
}

```

```

/*문의*/
// 조회

```

```

@GetMapping("/admin/qnaList")
public String qnaList(Model model) {
    List<Qna> qnaList = phBoardService.getQnaList();
    List<QnaReply> qnaReplyList = phBoardService.getQnaReplyList();
    model.addAttribute("qnaList", qnaList);
    model.addAttribute("qnaReplyList", qnaReplyList);
    model.addAttribute("title", "문의내역");

    return "admin/qna/qnaList";
}
@GetMapping("/supplier/qnaList")
public String qnaList2(Model model) {
    List<Qna> qnaList = phBoardService.getQnaList();
    List<QnaReply> qnaReplyList = phBoardService.getQnaReplyList();
    model.addAttribute("qnaList", qnaList);
    model.addAttribute("qnaReplyList", qnaReplyList);
    model.addAttribute("title", "문의내역");

    return "user/supplier/board/qnaList";
}
// 추가 (관리자)
@PostMapping("/admin/qna/add")
public String addQna(Qna qna){
    phBoardService.addQna(qna);
    return "redirect:/admin/qnaList";
}
@GetMapping("/admin/qnaAdd")
public String qnaAdd(){
    return "admin/qna/qnaAdd";
}
// 추가 (납품자)
@GetMapping("/supplier/qnaAdd")
public String qnaAdd2(){
    return "user/supplier/board/qna";
}
@PostMapping("/supplier/qna/add")
public String addQna2(Qna qna){
    phBoardService.addQna(qna);
    return "redirect:/supplier/qnaList";
}

// 삭제 (관리자)
@PostMapping("/admin/qna/delete")
public String deleteQna(@RequestParam String qseq){
    phBoardService.deleteQna(qseq);
    return "redirect:/admin/qnaList";
}
// 삭제 (납품자)
@PostMapping("/supplier/qna/delete")
public String deleteQna2(@RequestParam String qseq){
    phBoardService.deleteQna(qseq);
    return "redirect:/supplier/qnaList";
}

/*문의 답글 */
// 추가

```



```

@PostMapping("/admin/qnaReply/add")
public String addQnaReply(QnaReply qnaReply){
    phBoardService.addQnaReply(qnaReply);
    return "redirect:/admin/qnaList";
}
// 삭제
@PostMapping("/admin/qnaReply/delete")
public String deleteQnaReply(@RequestParam String qrSeq){
    phBoardService.deleteQnaReply(qrSeq);
    return "redirect:/admin/qnaList";
}
}
}

```

■ Service

```

package team01.yaksutor.pharmacy.service;

import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import team01.yaksutor.pharmacy.dto.*;
import team01.yaksutor.pharmacy.mapper.PhBoardMapper;

import java.util.List;

@Service("phBoardService")
@Transactional
@RequiredArgsConstructor
@Slf4j
public class PhBoardService {
    private final PhBoardMapper phBoardMapper;

    /* -----커뮤니티-----*/
    // 조회
    public List<Board> getBoardList() {
        List<Board> boardList = phBoardMapper.getBoardList();
        return boardList;
    }
    // 추가
    public void addBoard(Board board){phBoardMapper.addBoard(board);
    }
    // 수정
    public int modifyBoard(Board board){
        return phBoardMapper.modifyBoard(board);
    }
}

```

```

}
// 삭제
public void deleteBoard(String boardCode){
    phBoardMapper.deleteBoard(boardCode);
}

/* -----커뮤니티 댓글-----*/
// 조회
public List<Repl> getReplList() {
    List<Repl> replList = phBoardMapper.getReplList();
    return replList;
}
// 추가
public void addRepl(Repl repl) {
    phBoardMapper.addRepl(repl);
}
// 수정
public int modifyRepl(Repl repl) {
    return phBoardMapper.modifyRepl(repl);
}
// 삭제
public void deleteRepl(String replNum) {
    phBoardMapper.deleteRepl(replNum);
}

/* -----자주하는 질문-----*/
// 조회
public List<QuestionCenter> getQuestionCenterList(){
    List<QuestionCenter> questionCenterList = phBoardMapper.getQuestionCenterList();
    return questionCenterList;
}
// 추가
public void addQuestionCenter(QuestionCenter questionCenter) {
    log.info("questionCenter :{}", questionCenter);
    phBoardMapper.addQuestionCenter(questionCenter);
}
// 수정
public int modifyQuestionCenter(QuestionCenter questionCenter){
    return phBoardMapper.modifyQuestionCenter(questionCenter);
}
// 삭제
public void deleteQuestionCenter(String questionNum){
    phBoardMapper.deleteQuestionCenter(questionNum);
}
}

```

```

/* -----공지사항-----*/
// 조회
public List<Notice> getNoticeList(){
    List<Notice> noticeList = phBoardMapper.getNoticeList();
    return noticeList;
}
// 추가
public void addNotice(Notice notice){
    phBoardMapper.addNotice(notice);
}
// 수정
public int modifyNotice(Notice notice){
    return phBoardMapper.modifyNotice(notice);
}
// 삭제
public void deleteNotice(String noticeNum){
    phBoardMapper.deleteNotice(noticeNum);
}

/* -----1:1 문의-----*/
// 조회
public List<Qna> getQnaList(){
    List<Qna> qnaList = phBoardMapper.getQnaList();
    return qnaList;
}
public List<QnaReply> getQnaReplyList(){
    List<QnaReply> qnaReplyList = phBoardMapper.getQnaReplyList();
    return qnaReplyList;
}
// 문의 추가
public void addQna(Qna qna) {
    phBoardMapper.addQna(qna);
}
// 답변 추가
@Transactional
public void addQnaReply(QnaReply qnaReply) {
    phBoardMapper.addQnaReply(qnaReply);
    String qseq = qnaReply.getQseq();
    phBoardMapper.updateQna(qseq);
}

// 삭제
@Transactional
public void deleteQna(String qseq) {
    //qseq 로 조회를 해 먼저 근데 조인을해서 답글이랑 조인을해서

```

```

String qrseq = phBoardMapper.getQnaListByKey(qseq);
if(!qrseq.isEmpty()){
    phBoardMapper.deleteQnaReply(qrseq);
    phBoardMapper.deleteQna(qseq);
}
}
public void deleteQnaReply(String qrSeq){
    phBoardMapper.deleteQnaReply(qrSeq);
}
}
}

```

■ Mapper(SQL)

```

<!-- =====커뮤니티===== -->

<!-- 커뮤니티 조회 -->
<resultMap type="Board" id="boardResultMap">
    <id column="cm_num" property="boardCode" />
    <result column="m_id" property="mId"/>
    <result column="cm_title" property="boardTitle"/>
    <result column="cm_content" property="boardContent"/>
    <result column="cm_view" property="boardView"/>
    <result column="cm_date" property="boardDate"/>
    <result column="replSize" property="replSize"/>
</resultMap>

<select id="getBoardList" resultMap="boardResultMap">
    SELECT
        c.cm_num,
        c.m_id,
        c.cm_title,
        c.cm_content,
        c.cm_view,
        c.cm_date,
        (
            SELECT COUNT(1) FROM repl AS r WHERE c.cm_num = r.cm_num
        ) AS replSize
    FROM
        community AS c
    order by c.cm_date desc;
</select>

<!-- 커뮤니티 등록 -->
<insert id="addBoard" parameterType="Board">
    <selectKey keyProperty="boardCode" resultType="String" order="BEFORE">
        SELECT
            CASE
                WHEN COUNT(cm_num) = 0 THEN 'comm_1'
            ELSE
                CONCAT(
                    'comm_',
                    MAX(CAST(SUBSTRING_INDEX(cm_num, 'comm_', -1) AS UNSIGNED)) + 1)
            END AS boardCode
        FROM
            community;
    
```

```

</selectKey>
INSERT INTO community
  (cm_num, m_id, cm_title, cm_content, cm_view, cm_date)
  values
  ({boardCode}, {mId}, #{boardTitle}, #{boardContent},#{boardView}, CURDATE());
</insert>

<!--커뮤니티 수정-->
<update id="modifyBoard" parameterType="Board">
  UPDATE community
  <set>
    <if test="boardTitle != null and boardTitle != "">
      cm_title = #{boardTitle},
    </if>
    <if test="boardContent != null and boardContent != "">
      cm_content = #{boardContent},
    </if>
  </set>
  WHERE
  cm_num = #{boardCode}
</update>

<!--커뮤니티 삭제-->
<delete id="deleteBoard" parameterType="String">
  DELETE FROM community
  WHERE cm_num = #{boardCode}
</delete>

<!-- =====커뮤니티 댓글===== -->

<!--커뮤니티 댓글 조회-->
<resultMap type="Repl" id="replResultMap">
  <id column="repl_num" property="replNum" />
  <result column="m_id" property="mId"/>
  <result column="cm_num" property="cmNum"/>
  <result column="repl_content" property="replContent"/>
  <result column="repl_date" property="replDate"/>
</resultMap>
<select id="getReplList" resultMap="replResultMap">
  SELECT
    r.repl_num,
    r.m_id,
    r.cm_num,
    r.repl_content,
    r.repl_date
  FROM
    repl AS r;
</select>
<!--커뮤니티 댓글 등록-->
<insert id="addRepl" parameterType="Repl">
  <selectKey keyProperty="replNum" resultType="String" order="BEFORE">
    SELECT
    CASE
    WHEN COUNT(repl_num) = 0 THEN 'repl_num_1'
    ELSE
    CONCAT(
    'repl_num_',
    MAX(CAST(SUBSTRING_INDEX(repl_num, 'repl_num_', -1) AS UNSIGNED)) + 1)
    END AS replNum
    FROM
    repl;
  </selectKey>
  INSERT INTO repl
  (repl_num, m_id, cm_num, repl_content, repl_date)

```

```

values
({replNum}, #{mId}, #{cmNum}, #{replContent}, CURDATE());
</insert>
<!--커뮤니티 댓글 수정-->
<update id="modifyRepl" parameterType="Repl">
    UPDATE repl
    <set>
        <if test="replContent != null and replContent != "">
            repl_content = #{replContent},
        </if>
    </set>
    WHERE
        repl_num = #{replNum}
</update>

<!--커뮤니티 댓글 삭제-->
<delete id="deleteRepl" parameterType="String">
    DELETE FROM repl
    WHERE repl_num = #{replNum}
</delete>

```

납품업체 회원 추가 (관리자 화면)

납품업체 추가 등록 취소 회원 등록

1. 회원정보 입력

회원아이디
아이디를 입력해주세요

회원비밀번호
비밀번호를 입력해주세요

회원이름
이름을 입력해주세요

이메일
이메일을 입력해주세요

성명
연도-월-일

핸드폰
이메일을 입력해주세요

성별 남 여

회원주소입력 주소검색

우편번호

도도도주소

상세주소

2. 업체 정보 입력

업체명
업체명을 입력해주세요

업체 사업자번호
업체사업자번호입력

업체전화번호
업체전화번호입력

사업장주소입력 주소검색

우편번호

도도도주소

상세주소

업체 소개
업체 소개 문 입력

2024 © Next - HTML, Ecommerce Template. All rights reserved

납품업체 회원 추가 (관리자 화면)

- 관리자가 사용자의 정보를 입력하여 납품업체 권한의 회원을 등록하는 기능을 수행하는 페이지
- 필수정보를 입력하지 않고 회원등록을 하게 되면 해당 필드가 비어있다는 메시지와 함께 해당 입력 필드로 포커싱 된다.
- 이미 사용중인 id 의 경우 이미 사용중인 아이디라는 메시지가 출력된다.
- 주소검색 버튼을 클릭하면 다음카카오 주소 API 를 호출하여 주소를 검색 후 주소를 추가 할 수 있다.

■ Html(javascript)

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorate="~{admin/layout/default}">
<head>
</head>

<th:block layout:fragment="customContent">
  <section class="content-main">
    <form id="addSupplier" th:action="@{/admin/supplier/supplierInsert}" method="post">
      <div class="row">
        <div class="col-6">
          <div class="content-header">
            <h2 class="content-title" th:text="${content}">새로운 회원 등록</h2>
            <div>
              <button id="resetBtn" class="btn btn-light rounded font-sm mr-5 text-body hover-up">등록
취소</button>
              <input type="button" id="submitBtn" class="btn btn-md rounded font-sm hover-up" value="회원
등록" />
            </div>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-lg-6">
          <div class="card mb-4">
            <div class="card-body">
              <div class="row">
                <div class="col-md-3">
                  <h6 class="fs-5">1. 회원정보입력</h6>
                </div>
                <div class="col-md-9">
                  <div class="mb-4">
                    <label class="form-label fs-6" for="memberId">회원아이디</label>
                    <input type="text" placeholder="아이디를 입력해주세요" name="memberId" id="memberId"
class="form-control w-50" >
                    <span id="check-message" class="form-text"></span>
                  </div>
                  <div class="mb-4">
                    <label class="form-label fs-6" for="memberPw">회원비밀번호</label>
                    <input type="text" placeholder="비밀번호를 입력해주세요." name="memberPw" id="memberPw"
class="form-control w-50" >
                  </div>

                  <div class="mb-4">
                    <label class="form-label fs-6 " for="memberName">회원이름</label>
                    <input type="text" placeholder="이름을 입력해주세요" name="memberName"
id="memberName" class="form-control w-50">

```

```

</div>
<div class="mb-4">
  <label class="form-label fs-6" for="memberEmail">이메일</label>
  <input type="text" placeholder="이메일을 입력해주세요" name="memberEmail"
id="memberEmail" class="form-control w-50">
</div>
<div class="mb-4">
  <label class="form-label fs-6" for="memberBirth">생일</label>
  <input type="date" name="memberBirth" id="memberBirth" class="form-control w-50">
</div>
<div class="mb-4">
  <label class="form-label fs-6" for="memberPhone">핸드폰</label>
  <input type="text" placeholder="이메일을 입력해주세요" name="memberPhone"
id="memberPhone" class="form-control w-50">
</div>
<div class="mb-4">
  <label class="form-label mr-30 fs-6" >성별</label>
  <input class="mr-5 fs-5" type="radio" name="memberGender" value="남"/> 남
  <input class="ml-5 mr-5 fs-5" type="radio" name="memberGender" value="여"/> 여
</div>
<div >
  <div class="mb-4 d-flex">
    <label class="form-label mr-30 fs-6" >회원주소입력</label>
    <button type="button" id="addrBtn" class="btn btn-md rounded font-sm hover-up ms-auto
me-3">주소검색</button>
  </div>
  <input class="form-control mb-3 w-25" type="text" name="memberPostCode"
id="memberPostcode" placeholder="우편번호" />
  <input class="form-control mb-3" type="text" name="targetAddr" id="targetAddr"
placeholder="도로명주소" />
  <input class="form-control mb-3" type="text" name="targetAddrDetail" id="targetAddrDetail"
placeholder="상세주소" />
  <input type="hidden" name="memberAddrSido" id="memberAddrSido" />
  <input type="hidden" name="memberAddrSigungu" id="memberAddrSigungu" />
  <input type="hidden" name="memberAddrDong" id="memberAddrDong" />
  <input type="hidden" name="memberAddr" id="memberAddr" />
</div>
</div> <!-- col.// -->
</div> <!-- row.// -->
<hr class="mb-4 mt-0">
<div class="row">
  <div class="col-md-3">
    <h6 class="fs-5">2. 업체 정보 입력</h6>
  </div>
  <div class="col-md-9">
    <div class="mb-4">
      <label class="form-label fs-6 " for="supName">업체명</label>
      <input type="text" placeholder="업체명을 입력해주세요" name="supName" id="supName"
class="form-control w-50">
    </div>
    <div class="mb-4">
      <label class="form-label fs-6" for="supNum">업체 사업자번호</label>
      <input type="text" placeholder="업체사업자번호입력" name="supNum" id="supNum"
class="form-control w-50">
    </div>
    <div class="mb-4">
      <label class="form-label fs-6" for="supTell">업체전화번호</label>
      <input type="text" placeholder="업체전화번호입력" name="supTell" id="supTell" class="form-

```



```

control w-50">
    </div>
    <div >
        <div class="mb-4 d-flex">
            <label class="form-label mr-30 fs-6" >사업장주소입력</label>
            <button type="button" id="supAddrBtn" class="btn btn-md rounded font-sm hover-up ms-auto
me-3">주소검색</button>
        </div>
        <input class="form-control mb-3 w-25" type="text" name="supPostcode" id="supPostcode"
placeholder="우편번호" />
        <input class="form-control mb-3" type="text" name="targetSupAddr" id="targetSupAddr"
placeholder="도로명주소" />
        <input class="form-control mb-3" type="text" name="targetSupAddrDetail"
id="targetSupAddrDetail" placeholder="상세주소" />
        <input type="hidden" name="supAddr" id="supAddr" />
    </div>
    <div class="mb-4">
        <label class="form-label fs-6" for="supIntro">업체 소개</label>
        <textarea type="text" placeholder="업체 소개 글 입력" name="supIntro" id="supIntro"
class="form-control w-50" ></textarea>
    </div>

</div> <!-- col.// -->
</div> <!-- row.// -->
</div>
</div>
</div>
</div>
</div>
</form>
</section> <!-- content-main end// -->

</th:block>

<th:block layout:fragment="customJsFile">
</th:block>

<th:block layout:fragment="customJs">
<script src="//t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"> </script>
<script th:inline="javascript">
function searchAddress(searchObj) {
    new daum.Postcode({
        oncomplete: function(data) {
            // 팝업에서 검색결과 항목을 클릭했을때 실행할 코드를 작성하는 부분.
            // 각 주소의 노출 규칙에 따라 주소를 조합한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var addr = ''; // 주소 변수

            //사용자가 선택한 주소 타입에 따라 해당 주소 값을 가져온다.
            if (data.userSelectedType === 'R') { // 사용자가 도로명 주소를 선택했을 경우
                data.address = data.roadAddress;
            } else { // 사용자가 지번 주소를 선택했을 경우(J)
                data.address = data.jibunAddress;
            }

            // 우편번호와 주소 정보를 해당 필드에 넣는다.

            if(searchObj != null){
                const valueArr = searchObj.value;

```

```

    for(const key in searchObj){
        if(key === 'focus'){
            document.getElementById(searchObj[key]).focus();
            continue;
        }
        document.getElementById(searchObj[key]).value = data[key];
    }
}

/*
document.getElementById('memberPostcode').value = data.zonecode;
document.getElementById("memberAddress").value = addr;
// 커서를 상세주소 필드로 이동한다.
document.getElementById("memberDetailAddress").focus();

*/
}
}).open();
}

//회원 주소 객체
$('#addrBtn').click(function(){
    const searchObj = {
        zonecode : 'memberPostcode',
        address: 'targetAddr',
        focus : 'targetAddrDetail'
    }
    searchAddress(searchObj);
});
//납품 업체주소객체
$('#supAddrBtn').click(function(){
    const searchObj = {
        zonecode : 'supPostcode',
        address: 'targetSupAddr',
        focus : 'targetSupAddrDetail'
    }
    searchAddress(searchObj);
});

$('#resetBtn').click(e => {
    e.preventDefault();
    location.href = '/admin/adminMain';
});

$('#submitBtn').click(function(){

    let isSubmit = true;

    // 회원의 주소 파싱 및 숨겨진 필드에 값 설정
    const fullAddress = $('#targetAddr').val().trim();
    console.log(fullAddress);
    const addrDetail = $('#targetAddrDetail').val().trim();
    console.log(addrDetail);
    if (fullAddress) {
        try {
            const parsedAddress = parseAddress(fullAddress);
            $('#memberAddrSido').val(parsedAddress.sido);

```

```

$('#memberAddrSigungu').val(parsedAddress.sigungu);
$('#memberAddrDong').val(parsedAddress.dong);
$('#memberAddr').val(parsedAddress.detail + ' ' + addrDetail);
} catch (e) {
  alert('잘못된 회원 주소 형식입니다.');
```

```

//납품업체의 주소 파싱 및 숨겨진 필드에 값 설정
const fullSupAddress = $('#targetSupAddr').val();
const addrSupDetail = $('#targetSupAddrDetail').val();
const postcode = $('#supPostcode').val();
```

```

if (fullSupAddress && postcode) {
  // Hidden input 에 '(우편번호) 전체주소 상세주소' 형식으로 저장
  $('#supAddr').val(`${postcode} ${fullSupAddress} ${addrSupDetail}`);
} else {
  if (!postcode) {
    alert('우편번호는 필수 입력 항목입니다.');
```

```

$('#addSupplier input:not([type="hidden"])').each((idx, element) => {
  const id = element.id;
  const value = element.value;
  if(!value){
    alert(`${id}는(은) 필수 입력 항목입니다.');
```

```

if(isSubmit) $('#addSupplier').submit();
```

```
});
```

```
//주소 파싱하는 함수 정의
```

```

function parseAddress(address) {
  const pattern = /^(?<sid>[가-힣]+특별자치도|[가-힣]+시|[가-힣]+도|[가-힣]+북|[가-힣]+남|[가-힣]+기|[가-
  힣]+산|[가-힣]+구|[가-힣]+울|[가-힣]+전)\Ws+(?<sigungu>[가-힣]+시\Ws+[가-힣]+구|[가-힣]+시|[가-힣]+군|[가-
  힣]+구)\Ws+(?<detail>.+)/;
  const match = address.match(pattern);
  if (match) {
```

```

const detailPattern = /(?!<dong>[가-힣]+[등|읍|면|리])?Ws*(?<detail>.+)?/;
const detailMatch = match.groups.detail.match(detailPattern);
return {
  sido: match.groups.sido,
  sigungu: match.groups.sigungu,
  dong: detailMatch.groups.dong || "",
  detail: detailMatch.groups.detail || match.groups.detail
};
} else {
  throw new Error('Invalid address format');
}
}

//회원가입시 ID 체크
$(document).ready(function() {
  $('#memberId').on('blur', function() {
    var memberId = $(this).val();
    var $checkMessage = $('#check-message');

    if (memberId.length === 0) {
      $checkMessage.text("").hide();
      return;
    }

    $.ajax({
      url: '/admin/member/idCheck',
      method: 'POST',
      data: { memberId: memberId },
      dataType: 'json',
      success: function(response) {
        if (response) {
          $checkMessage.text('이미 사용 중인 아이디입니다.').css('color', 'red').show();
        } else {
          $checkMessage.text('사용 가능한 아이디입니다.').css('color', 'green').show();
          $('#submitBtn').prop('disabled', false);
        }
      }
    });
  });
});
</script>
</th:block>
</html>

```

■ Controller

```

package team01.yaksutor.admin.controller;

import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.security.core.parameters.P;

```

```

import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import team01.yaksutor.admin.mapper.AdMemberMapper;
import team01.yaksutor.admin.service.AdMemberService;
import team01.yaksutor.common.mapper.MemberMapper;
import team01.yaksutor.dto.License;
import team01.yaksutor.dto.ManagePharmacy;
import team01.yaksutor.dto.Member;
import team01.yaksutor.dto.Pharmacy;
import team01.yaksutor.pharmacy.dto.QuestionCenter;

import java.util.List;

```

```

@Controller
@RequestMapping("/admin/member")
@Slf4j
@RequiredArgsConstructor
public class AdMemberController {

    private final AdMemberMapper admembermapper;
    private final PasswordEncoder passwordEncoder;
    private final AdMemberService adMemberService;
    private final MemberMapper memberMapper;

    /*약국 삭제*/
    @PostMapping("/pharmacyDelete")
    public String deletePharmacy(Pharmacy pharmacy){
        adMemberService.pharmacyDelete(pharmacy);
        return "redirect:/admin/member/pharmacySearchList";
    }

    /*약국 수정*/
    @PostMapping("/pharmacyModify")
    public String modifyPharmacy(Pharmacy pharmacy){
        adMemberService.pharmacyModify(pharmacy);
        return "redirect:/admin/member/pharmacySearchList";
    }
}

```

```

/*회원 수정*/
@PostMapping("/memberModify")
public String modifyMember(Member member){
    adMemberService.memberModify(member);
    return "redirect:/admin/member/memberSearchList";
}

/*회원 삭제*/
@PostMapping("/memberDelete")
public String deleteMember(@RequestParam String memberId){
    adMemberService.memberDelete(memberId);
    return "redirect:/admin/member/memberSearchList";
}

@GetMapping("/pharmacySearchList")
public String pharmacySearchList(Model model) {
    List<Pharmacy> pharmacyList = adMemberService.getPharmacyInfo();
    model.addAttribute("title", "약국 조회");
    model.addAttribute("content", "약국 조회");
    model.addAttribute("pharmacyList", pharmacyList);
    return "admin/pharmacy/pharmacySearchList";
}

@GetMapping("/memberSearchList")
public String memberSearchList(Model model) {
    List<Member> memberList = adMemberService.getMemberInfo();
    model.addAttribute("title", "전체회원정보");
    model.addAttribute("content", "전체회원정보");
    model.addAttribute("memberList", memberList);

    return "admin/pharmacy/memberSearchList";
}

@PostMapping("/adPharmacyInsert")
public String AdPharmacyInsert(Member member, Pharmacy pharmacy) {
    log.info("member {}", member);
    log.info("pharmacy {}", pharmacy);
    adMemberService.staffToOwnerMember(member, pharmacy);
    return "redirect:/admin/adminMain";
}

```

```

}

@PostMapping("/AdmemberInsert")
public String AdMemberInsert(Member member, Pharmacy pharmacy, License license, Model model) {

    //license 와 pharmacy 에 meberId 주입
    license.setPharmacistId(member.getMemberId());

    log.info("member {}", member); //애네가 html 에서 name 에 해당되는 DTO 에 매핑되어서 저장
    log.info("pharmacy {}", pharmacy);
    log.info("license {}", license);

    //서비스
    int memberLevel = member.getMemberLevelNum();
    member.setMemberPw(passwordEncoder.encode(member.getMemberPw()));
    log.info("member>> {}", member);
    if (memberLevel == 3) {
        pharmacy.setPharId(member.getMemberId());
        adMemberService.insertOwnerMember(member, license, pharmacy);
    } else if (memberLevel == 4) {
        ManagePharmacy managePharmacy = new ManagePharmacy();
        //pharCode 조회
        String pid = pharmacy.getPharId();
        String pname = pharmacy.getPName();
        String pharCode = admembermapper.getPharCode(pid);
        log.info("adadpharCode {}", pharCode);
        //managePharmacy 에 기본값 셋팅
        managePharmacy.setPharCode(pharCode);
        managePharmacy.setPharId(member.getMemberId());
        log.info("managePharmacy: {}", managePharmacy);
        adMemberService.insertStaffMember(member, license, managePharmacy);
    }
    return "redirect:/admin/member/memberInsert";
}

```

```

@PostMapping("/idCheck")
@ResponseBody
public boolean idCheck(@RequestParam(value = "memberId") String memberId) {
    log.info("아이디중복체크: memberId {}", memberId);
    boolean isMember = false;
    isMember = admembermapper.idCheck(memberId);
    return isMember;
}

```

```

}

@GetMapping("/pharmacyInsert")
public String pharmacyInsert(Model model) {

    model.addAttribute("title", "기존 관리약사 개국약사로 변경");
    model.addAttribute("content", "기존 관리약사 개국약사로 변경");

    return "admin/pharmacy/pharmacyInsert";
}

@GetMapping("/memberInsert")
public String memberInsert(Model model) {
    model.addAttribute("title", "약사회원추가");
    model.addAttribute("content", "약사회원등록");
    return "admin/pharmacy/memberInsert";
}

@PostMapping("/isStaff")
@ResponseBody
public boolean isStaff(@RequestParam(value = "memberId") String memberId) {
    boolean isStaff = false;

    // memberMapper 를 통해 memberId 로 회원 정보를 가져옵니다.
    Member member = null;
    try {
        member = memberMapper.getMemberInfoById(memberId);
    } catch (Exception e) {
        // 예외 발생 시 로그를 기록합니다.
        System.err.println("Error retrieving member info: " + e.getMessage());
        e.printStackTrace();
        throw new RuntimeException("Error retrieving member info", e);
    }

    // member 가 null 인 경우 예외를 던집니다.
    if (member == null) {
        throw new IllegalArgumentException("Member not found for ID: " + memberId);
    }

    // member 의 레벨을 확인합니다.
    int memberLevel = member.getMemberLevelNum();
    System.out.println("Member Level: " + memberLevel);
}

```



```

// memberLevel 이 4 인 경우 isStaff 를 true 로 설정합니다.
if (memberLevel == 4) {
    isStaff = true;
} else {
    isStaff = false;
}

System.out.println("isStaff: " + isStaff);
return isStaff;
}
}

```

■ Service

```

package team01.yaksutor.admin.service;

import jakarta.servlet.http.HttpServletRequest;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import team01.yaksutor.admin.mapper.AdMemberMapper;
import team01.yaksutor.dto.Member;
import team01.yaksutor.dto.Supplier;

import java.util.List;

@Service
@RequiredArgsConstructor
@Slf4j
public class AdSupplierService {

    private final HttpServletRequest request;
    private final PasswordEncoder passwordEncoder;
    private final AdMemberMapper adMemberMapper;

```

```
@Transactional
```

```
public void supplierInsert(Member member, Supplier supplier) {
```

```
    member.setMemberPw(passwordEncoder.encode(member.getMemberPw())); //비밀번호 암호화
```

```
    member.setMemberLevelNum(2); //납품업체 권한 셋팅
```

```
    String sessionId = (String) request.getSession().getAttribute("S_ID"); //현재 로그인중인 관리자의 세션 ID
```

```
    supplier.setSuppld(member.getMemberId()); // 납품업체 id 에 memberId 주입
```

```
    supplier.setAdminId(sessionId); // 세션 ID 를 등록자 ID 에 주입
```

```
    supplier.setAppState("관리자승인");
```

```
    adMemberMapper.memberInsert(member);
```

```
    adMemberMapper.supplierInsert(supplier);
```

```
}
```

```
public List<Supplier> getSupplierInfo() {
```

```
    List<Supplier> supplierInfo = adMemberMapper.getSupplierInfo();
```

```
    return supplierInfo;
```

```
}
```

```
//납품업체 폐업
```

```
public int supplierDelete(Supplier supplier) {
```

```
    return adMemberMapper.deleteSupplier(supplier);
```

```
}
```

```
public int supplierModify(Supplier supplier) {
```

```
    return adMemberMapper.modifySupplier(supplier);
```

```
}
```

```
}
```

■ Mapper(SQL)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="team01.yaksutor.admin.mapper.AdMemberMapper">

  <!-- AS 매번 하기 귀찮으니까 아래 resultMap 을 사용함 -->
  <resultMap id="memberResultMap" type="team01.yaksutor.dto.Member">
    <!-- PK 칼럼과 일반칼럼을 주의해야한다. -->
    <id column="m_id" property="memberId" />
    <result column="level" property="memberLevelNum"/>
    <result column="level_name" property="memberLevelName"/>
    <result column="pw" property="memberPw"/>
    <result column="name" property="memberName"/>
    <result column="gender" property="memberGender"/>
    <result column="email" property="memberEmail"/>
    <result column="birth" property="memberBirth"/>
    <result column="phone" property="memberPhone"/>
    <result column="sido" property="memberAddrSido"/>
    <result column="sigungu" property="memberAddrSigungu"/>
    <result column="dong" property="memberAddrDong"/>
    <result column="pst_code" property="memberPostCode"/>
    <result column="addr" property="memberAddr"/>
  </resultMap>

  <resultMap id="pharmacyResultMap" type="team01.yaksutor.dto.Pharmacy">
    <id column="phar_code" property="pharCode" />
    <result column="phar_id" property="pharId"/>
    <result column="p_name" property="pName"/>
    <result column="busin_num" property="businNum"/>
    <result column="p_addr_do" property="pAddrDo"/>
    <result column="p_addr_si" property="pAddrSi"/>
    <result column="p_addr_dong" property="pAddrDong"/>
    <result column="p_addr" property="pAddr"/>
    <result column="p_post_code" property="pPostCode"/>
    <result column="open_date" property="openDate"/>
    <result column="p_state" property="pState"/>
  </resultMap>

  <resultMap id="LicenseResultMap" type="team01.yaksutor.dto.License">
    <id column="lice_code" property="liceCode" />
```

```

<result column="pharmacist_id" property="pharmacistId"/>
<result column="license_num" property="licenseNum"/>
<result column="license_get_date" property="licenseGetDate"/>
</resultMap>

<resultMap id="memberLevelResultMap" type="team01.yaksutor.dto.Level">
  <id column="level_num" property="levelNum"/>
  <result column="level_name" property="levelName"/>
  <result column="level_date" property="levelDate"/>
</resultMap>

<resultMap id="managePharmacyResultMap" type="team01.yaksutor.dto.ManagePharmacy">
  <id column="mana_phar_code" property="manaPharCode"/>
  <result column="phar_code" property="pharCode"/>
  <result column="phar_id" property="pharId"/>
  <result column="reg_m_id" property="regMID"/>
  <result column="state" property="state"/>
  <result column="reg_date" property="regDate"/>
</resultMap>

<resultMap id="supplierResultMap" type="team01.yaksutor.dto.Supplier">
  <id column="supp_code" property="suppCode"/>
  <result column="supp_id" property="suppld"/>
  <result column="admin_id" property="adminId"/>
  <result column="sup_num" property="supNum"/>
  <result column="sup_intro" property="supIntro"/>
  <result column="sup_addr" property="supAddr"/>
  <result column="sup_name" property="supName"/>
  <result column="sup_tell" property="supTell"/>
  <result column="app_state" property="appState"/>
  <result column="reg_date" property="regDate"/>
  <result column="app_date" property="appDate"/>
</resultMap>

<sql id="memberSearch">
  SELECT
  m.m_id,
  m.`level`,
  l.level_name,
  m.pw,
  m.`name`,
  m.gender,
  m.email,
  m.birth,
  m.phone,
  m.sido,
  m.sigungu,
  m.dong,

```

```

m.pst_code,
m.addr
FROM `member` AS m
INNER JOIN `level` AS l
ON
m.`level` = l.level_num
</sql>

```

```

<select id="getPharmacyById" parameterType="String" resultMap="pharmacyResultMap">
/* 회원아이디로 약국 조회*/
SELECT
p.phar_code,
p.phar_id,
p.p_name,
p.busin_num,
p.p_addr_do,
p.p_addr_si,
p.p_addr_dong,
p.p_post_code,
p.p_addr,
p.open_date,
p.p_state
FROM
pharmacy AS p
INNER join
manage_pharmacy AS mp
on
p.phar_code = mp.phar_code
WHERE
p.phar_id = 'id013'
OR
mp.phar_id = 'id013'
GROUP BY p.phar_code;
</select>

```

```

<update id="modifySupplier" parameterType="Supplier">
/* 납품업체 수정 */
UPDATE supplier
<trim prefix="SET" suffixOverrides=",">
<if test="supNum != null and supNum != "">
sup_num=#{supNum},
</if>
<if test="supIntro != null and supIntro != "">
sup_intro=#{supIntro},
</if>
<if test="supAddr != null and supAddr != "">
sup_addr=#{supAddr},
</if>

```

```

    <if test="supName != null and supName != "">
        sup_name=#{supName},
    </if>
    <if test="supTell != null and supTell != "">
        sup_tell=#{supTell},
    </if>
    <if test="appState != null and appState != "">
        app_state=#{appState},
    </if>
</trim>
WHERE
supp_code=#{suppCode}
</update>

<update id="deleteSupplier" parameterType="Supplier">
    /*납품업체 삭제*/
    UPDATE supplier
    SET
    app_state='폐업'
    WHERE
    supp_code=#{suppCode}
</update>

<update id="deletePharmacy" parameterType="Pharmacy">
    /* 약국 삭제 */
    UPDATE pharmacy
    SET
    p_state='폐업'
    WHERE
    phar_code=#{pharCode}
</update>

<update id="modifyPharmacy" parameterType="Pharmacy">
    /* 약국 수정 */
    UPDATE pharmacy
    <trim prefix="SET" suffixOverrides=",">
        <if test="pName != null and pName != "">
            p_name=#{pName},
        </if>
        <if test="businNum != null and businNum != "">
            busin_num=#{businNum},
        </if>
        <if test="pAddrDo != null and pAddrDo != "">
            p_addr_do=#{pAddrDo},
        </if>
        <if test="pAddrSi != null and pAddrSi != "">
            p_addr_si=#{pAddrSi},
        </if>
    </trim>

```

```

<if test="pAddrDong != null and pAddrDong != "">
    p_addr_dong=#{pAddrDong},
</if>
<if test="pAddr != null and pAddr != "">
    p_addr=#{pAddr},
</if>
<if test="pPostCode != null and pPostCode != "">
    p_post_code=#{pPostCode},
</if>
<if test="openDate != null and openDate != "">
    open_date=#{openDate}
</if>
</trim>
WHERE
phar_code=#{pharCode};
</update>

<update id="modifyMember" parameterType="Member">
/* 회원수정 */
UPDATE member
<trim prefix="SET" suffixOverrides=",">
    <if test="memberName != null and memberName != "">
        name=#{memberName},
    </if>
    <if test="memberGender != null and memberGender != "">
        gender=#{memberGender},
    </if>
    <if test="memberEmail != null and memberEmail != "">
        email=#{memberEmail},
    </if>
    <if test="memberPhone != null and memberPhone != "">
        phone=#{memberPhone}
    </if>
</trim>
WHERE
m_id=#{memberId};
</update>

<select id="getSupplierInfo" resultMap="supplierResultMap">
SELECT
supp_code,
supp_id,
admin_id,
sup_num,
sup_intro,
sup_addr,
sup_name,
sup_tell,

```

```

app_state,
reg_date,
app_date
FROM
supplier
</select>

<update id="deleteMemberById" parameterType="Member">
UPDATE member AS m
SET
m.`level`=5
WHERE
m.m_id = #{memberId}
</update>

<select id="getPharmacyList" resultMap="pharmacyResultMap">
SELECT
phar_code,
phar_id,
p_name,
busin_num,
p_addr_do,
p_addr_si,
p_addr_dong,
p_post_code,
p_addr,
open_date,
p_state
FROM
pharmacy
</select>

<delete id="deleteManagePharmacyById" parameterType="String">
DELETE
FROM
manage_pharmacy AS mp
WHERE
mp.phar_id = #{staffId}
</delete>

<update id="updateMemberLevelNumById" parameterType="String">
UPDATE member AS m
SET
m.`level`=3
WHERE
m.m_id = #{staffId}
</update>

<insert id="supplierInsert" parameterType="Supplier">

```



```

<selectKey keyProperty="suppCode" resultType="String" order="BEFORE">
  /* supp_code 자동증가 코드 */
  SELECT
  CASE
  WHEN COUNT(supp_code) = 0 THEN 'supp_code_1'
  ELSE CONCAT(
  'supp_code_',
  MAX(CAST(SUBSTRING_INDEX(supp_code, '_', -1) AS UNSIGNED)) + 1
  )
  END AS suppCode
  FROM
  supplier;
</selectKey>

```

```

INSERT INTO supplier
(supp_code, supp_id, admin_id, sup_num, sup_intro, sup_addr, sup_name, sup_tell, app_state, reg_date, app_date)
VALUES ({suppCode}, {suppId}, {adminId}, {supNum}, {supIntro}, {supAddr}, {supName}, {supTell}, {appState},
CURDATE(), CURDATE())
</insert>

```

```

<insert id="memberInsert" parameterType="Member">
  INSERT INTO member
  (m_id, `level`, pw, name, gender, email, birth, phone, sido, sigungu, dong, pst_code, addr, draw_stat, level_reg_date)
  VALUES
  ({memberId}, {memberLevelNum}, {memberPw}, {memberName}, {memberGender}, {memberEmail},
  {memberBirth}, {memberPhone}, {memberAddrSido},
  {memberAddrSigungu}, ", {memberPostCode}, {memberAddr}, CURDATE(), CURDATE())
</insert>

```

```

<insert id="licenseInsert" parameterType="License">
  <selectKey keyProperty="liceCode" resultType="String" order="BEFORE">
    /* lice_code 코드 자동증가 */
    SELECT
    CASE
    WHEN COUNT(lice_code) = 0 THEN 'lice_code_1'
    ELSE CONCAT(
    'lice_code_',
    MAX(CAST(SUBSTRING_INDEX(lice_code, '_', -1) AS UNSIGNED)) + 1
    )
    END AS liceCode
    FROM
    license;
  </selectKey>
  /* 약사면허등록 */
  INSERT INTO license
  (lice_code, pharmacist_id, license_num, license_get_date)
  VALUES ({liceCode}, {pharmacistId}, {licenseNum}, {licenseGetDate})

```

</insert>

<insert id="pharmacyInsert" parameterType="Pharmacy">

<selectKey keyProperty="pharCode" resultType="String" order="BEFORE">

/* phar_code 자동증가 코드 */

SELECT

CASE

WHEN COUNT(phar_code) = 0 THEN 'phar_code_1'

ELSE CONCAT(

'phar_code_',

MAX(CAST(SUBSTRING_INDEX(phar_code, '_', -1) AS UNSIGNED)) + 1

)

END AS pharCode

FROM

pharmacy;

</selectKey>

INSERT INTO pharmacy

(phar_code, phar_id, p_name, busin_num, p_addr_do, p_addr_si, p_addr_dong, p_post_code, p_addr, open_date)

VALUES ({pharCode}, {pharId}, {pName}, {businNum}, {pAddrDo}, {pAddrSi}, {pAddrDong}, {pPostCode},

{pAddr}, {openDate})

</insert>

<insert id="managePharmacyInsert" parameterType="ManagePharmacy">

<selectKey keyProperty="manaPharCode" resultType="String" order="BEFORE">

/* mana_phar_code 자동증가 코드 */

SELECT

CASE

WHEN COUNT(mana_phar_code) = 0 THEN 'mana_phar_code_1'

ELSE CONCAT(

'mana_phar_code_',

MAX(CAST(SUBSTRING_INDEX(mana_phar_code, '_', -1) AS UNSIGNED)) + 1

)

END AS manaPharCode

FROM

manage_pharmacy;

</selectKey>

INSERT INTO manage_pharmacy

(mana_phar_code, phar_code, phar_id, reg_m_id, state, reg_date)

VALUES ({manaPharCode}, {pharCode}, {pharId}, 'admin', '재직중', CURDATE())

</insert>

<select id="getPharCode" resultType="String">

/* pharId 와 pName 으로 Phar_code 조회 */

SELECT phar_code

FROM pharmacy

WHERE phar_id = {pharId}

</select>

```

<select id="getMemberInfoById" parameterType="String" resultMap="memberResultMap">
  <include refid="memberSearch">
  </include>
  WHERE
  m.m_id = #{memberId}
</select>

<select id="idCheck" parameterType="String" resultType="boolean">
  SELECT
  COUNT(m_id)
  FROM
  member
  WHERE
  m_id=#{memberId};

</select>

</mapper>

```

납품업체 사용자 - 내 판매 의약품 상세

내 판매 의약품 상세 페이지 (납품업체 사용자)

- 납품업체가 현재 판매중인 의약품의 정보를 볼 수 있습니다.
- 수정하기 버튼을 통해 수정 페이지로 갈 수 있다
- 제품의 이미지를 확인 할 수 있다
- 성분과 효능은 개 수에 따라 여러개가 출력된다

■ Html(javascript)

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorate="~{user/supplier/layout/default}">
<head>
</head>

<th:block layout:fragment="customContent">
  <section class="content-main">
    <div class="content-header">
      <h2 class="content-title" th:text="${content}"></h2>
    </div>
    <div class="card">
      <div class="card-body">
        <div class="row gx-5">
          <div class="col-lg-9">
            <section class="content-body p-xl-4" th:object="${medilInfo}">
              <div class="row">
                <div class="col-lg-8">
                  <div class="row gx-3">
                    <div class="col-6 mb-3">
                      <label class="form-label">의약품 이름</label>
                      <input class="form-control" type="text" readonly="readonly"
th:value="**{medicine.mediName}"/>
                    </div>
                    <!-- col ./ -->
                    <div class="col-lg-6 mb-3">
                      <label class="form-label">소비기한</label>
                      <input class="form-control" type="date"
th:value="**{medicine.mediUsebydate}"/>
                    </div>
                    <!-- col ./ -->
                    <div class="col-6 mb-3">
                      <label class="form-label">판매자 아이디</label>
                      <input class="form-control" type="text" readonly="readonly"
th:value="**{medicine.regMld}"/>
                    </div>
                    <!-- col ./ -->
                    <div class="col-lg-6 mb-3">
                      <label class="form-label">업체명</label>
                      <input class="form-control" type="text" readonly="readonly"
th:value="${suppName}"/>
                    </div>
                    <!-- col ./ -->
                    <div class="col-lg-6 mb-3">
                      <label class="form-label">제형</label>
                      <input class="form-control" type="text" readonly="readonly"
th:value="**{medicine.mediDrugform}"/>

```

```

        </div>
        <!-- col ./ -->
        <div class="col-lg-6 mb-3">
            <label class="form-label">의약품 구분</label>
            <input class="form-control" type="text" readonly="readonly"
th:value="**{medicine.mediDrugtype}"/>
        </div>
        <!-- col ./ -->
        <div class="col-lg-12 mb-3">
            <label class="form-label">상세 설명</label>
            <textarea class="form-control" rows="4" readonly="readonly"
th:text="**{sellMedicine.mediDetail}"> </textarea>
        </div>
        <!-- col ./ -->
        <div class="col-lg-6 mb-3">
            <label class="form-label">가격</label>
            <input class="form-control" type="text" readonly="readonly"
th:value="**{sellMedicine.mediPrice}"/>
        </div>
        <!-- col ./ -->
        <div class="col-lg-6 mb-3">
            <label class="form-label">판매 수량</label>
            <input class="form-control" type="text" readonly="readonly"
th:value="**{sellMedicine.regQty}"/>
        </div>
        <!-- col ./ -->
        <div class="row">
            <label class="col-lg-6 mb-3 form-label">성분</label>
            <label class="col-lg-6 mb-3 form-label">함량</label>
        </div>
        <div class="row" th:if="{medilInfo.ingrList != null}" th:each="i :
${medilInfo.ingrList}">
            <div class="col-lg-6 mb-3">
                <input class="form-control" type="text" readonly="readonly"
th:value="{i.ingredient}"/>
            </div>
            <!-- col ./ -->
            <div class="col-lg-6 mb-3">
                <input class="form-control" type="text" readonly="readonly"
th:value="{i.content}"/>
            </div>
        </div>
        <!-- col ./ -->
        <div>
            <div class="col-lg-12 mb-3">
                <label class="form-label">효능</label>
                <input class="form-control" type="text" readonly="readonly"
th:value="{effies}"/>
            </div>
        </div>
        <!-- col ./ -->
        <div class="col-lg-6 mb-3">
            <label class="form-label">등록일</label>
            <input class="form-control" type="text" readonly="readonly"
th:value="**{sellMedicine.regDate}"/>
        </div>
    </div>

```

```

        <!-- row.// -->
    </div>
    <!-- col.// -->
    <aside class="col-lg-4">
        <figure class="text-lg-center">
            
            <figcaption>
                <label class="form-label">제품 이미지</label>
            </figcaption>
        </figure>
    </aside>
    <!-- col.// -->
</div>
<!-- row.// -->
<br />
<a th:href="@{/supplier/myMedicineModify(goodsCode=*{sellMedicine.goodsCode})}"
class="btn btn-sm font-sm rounded btn-brand"> <i class="material-icons md-edit"></i> 수정하기 </a>
    <hr class="my-5" />
    <!-- row.// -->
</section>
<!-- content-body .// -->
</div>
<!-- col.// -->
</div>
<!-- row.// -->
</div>
<!-- card body end// -->
</div>
<!-- card end// -->
</section>

</th:block>

<th:block layout:fragment="customJsFile">
</th:block>

<th:block layout:fragment="customJs">
    <script th:inline="javascript">
        $('#resetBtn').click(e => {
            e.preventDefault();
            location.href = '/admin/adminMain';
        });

        $('#submitBtn').click(function(){
            let isSubmit = true;
            $('#addMedicine input, #addMedicine select').each((idx, element) => {
                const id = element.id;
                const value = element.value;
                if(!value){
                    alert(`${id}는(은) 필수 입력 항목입니다`);
                    $(element).focus();
                    isSubmit = false;
                    return false;
                }
            });
            if(isSubmit) $('#addMedicine').submit();
        });
    </script>

```

```

const createIngrRow = () => {
  const ingrContent = [
    { 'tagName' : 'input', 'type' : 'text', 'name' : 'Ingredient' },
    { 'tagName' : 'input', 'type' : 'text', 'name' : 'content' },
    { 'tagName' : 'button', 'type' : 'button' }
  ];
  const $tr = $('<tr></tr>');
  ingrContent.forEach(item => {
    const $td = $('<td></td>');
    const tagName = item.tagName;
    delete item.tagName;
    const $element = $('<' + tagName + ' />', item);
    $td.append($element);
    $tr.append($td);
  });
  return $tr;
}
// 3. 행의 추가 버튼
$(document).on('click', '.ingrAddBtn', function(){
  $('#ingrTbody button:last').addClass('ingrDelBtn')
  .removeClass('ingrAddBtn')
  .text('삭제');
  const $row = createIngrRow();
  $row.find('input').addClass('form-control');
  $row.find('button').addClass('btn btn-md rounded font-sm hover-up ingrAddBtn')
  .text('추가');
  $('#ingrTbody').append($row);
});
// 4. 행의 삭제 버튼
$(document).on('click', '.ingrDelBtn', function(){
  $(this).closest('tr').remove();
});

const createEffiRow = () => {
  const ingrContent = [
    { 'tagName' : 'input', 'type' : 'text', 'name' : 'efficacy' },
    { 'tagName' : 'button', 'type' : 'button' }
  ];
  const $tr = $('<tr></tr>');
  ingrContent.forEach(item => {
    const $td = $('<td></td>');
    const tagName = item.tagName;
    delete item.tagName;
    const $element = $('<' + tagName + ' />', item);
    $td.append($element);
    $tr.append($td);
  });
  return $tr;
}
// 3. 행의 추가 버튼
$(document).on('click', '.effiAddBtn', function(){
  $('#effiTbody button:last').addClass('effiDelBtn')
  .removeClass('effiAddBtn')
  .text('삭제');
  const $row = createEffiRow();
  $row.find('input').addClass('form-control');
  $row.find('button').addClass('btn btn-md rounded font-sm hover-up float-end effiAddBtn')

```

```

        .text('추가');
        $('#effiTbody').append($row);
    });
    // 4. 행의 삭제 버튼
    $(document).on('click', '.effiDelBtn', function(){
        $(this).closest('tr').remove();
    });
</script>
</th:block>
</html>

```

■ Controller

```

package team01.yaksutor.supplier.controller;

import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpSession;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.MultipartFile;
import team01.yaksutor.dto.*;
import team01.yaksutor.supplier.mapper.SuMedicineMapper;
import team01.yaksutor.supplier.service.SuMedicineService;

import java.util.List;

@Controller
@RequestMapping("/supplier")
@RequiredArgsConstructor
@Slf4j
public class SuMedicineController {
    private final HttpServletRequest request;
    private final SuMedicineService suMedicineService;
    private final SuMedicineMapper suMedicineMapper;

    /**
     * 판매 의약품 등록 페이지 맵핑
     * 작성자: 길범진
     * @param model
     * @return
     */
    @GetMapping("/myMedicineInsert")
    public String myMedicineInsert(Model model) {
        model.addAttribute("title", "내 의약품 등록");
        model.addAttribute("content", "내 의약품 등록");

        return "user/supplier/medicine/myMedicineInsert";
    }

    /**
     * 납품업자 별 판매 의약품 목록 페이지 맵핑
     * @param model
     * @return
     */

```



```

*/
@GetMapping("/myMedicineSearchList")
public String myMedicineSearchList(Model model) {
    String sid = request.getSession().getAttribute("S_ID").toString();
    log.info("sid: {}", sid);
    String suppCode = suMedicineMapper.getSuppCode(sid);
    List<SellMedicine> sellMediList = suMedicineService.getSellMediList(suppCode);

    model.addAttribute("sellMediList", sellMediList);
    model.addAttribute("title", "내 의약품 검색");
    model.addAttribute("content", "내 의약품 검색");

    return "user/supplier/medicine/myMedicineSearchList";
}

/**
 * 내 의약품 수정 페이지 맵핑
 * 작성자: 길범진
 * @param model
 * @param goodsCode
 * @return
 */
@GetMapping("/myMedicineModify")
public String myMedicineModify(Model model,
                               @RequestParam(value="goodsCode") String goodsCode) {
    String sid = request.getSession().getAttribute("S_ID").toString();
    String mediCode = suMedicineMapper.getMediCode(goodsCode);
    Medicine medicine = suMedicineMapper.getMedicineInfo(goodsCode);
    SellMedicine sellMedicine = suMedicineMapper.getSellMedicineInfo(goodsCode);
    List<Ingredient> ingrList = suMedicineMapper.getIngrList(mediCode);
    List<Efficacy> effiList = suMedicineMapper.getEffiList(mediCode);
    MedicineInfo mediInfo = new MedicineInfo();
    mediInfo.setMedicine(medicine);
    mediInfo.setSellMedicine(sellMedicine);
    mediInfo.setIngrList(ingrList);
    mediInfo.setEffiList(effiList);
    log.info("mediInfo: {}", mediInfo);
    model.addAttribute("mediInfo", mediInfo);
    model.addAttribute("title", "내 의약품 수정");
    model.addAttribute("content", "내 의약품 수정");

    return "user/supplier/medicine/myMedicineModify";
}

/**
 * 내 의약품 판매상태 -> '삭제'로 수정
 * @param model
 * @param goodsCode
 * @return
 */
@GetMapping("/myMedicineDelete")
public String myMedicineDelete(Model model,
                               @RequestParam(value="goodsCode") String goodsCode) {

    String mediCode = suMedicineMapper.getMediCode(goodsCode);
    Medicine medicine = suMedicineMapper.getMedicineInfo(goodsCode);
    String mediName = medicine.getMediName();

```

```

model.addAttribute("mediName", mediName);
model.addAttribute("mediCode", mediCode);
model.addAttribute("title", "내 의약품 삭제");
model.addAttribute("content", "내 의약품 삭제");

return "user/supplier/medicine/myMedicineDelete";
}

/**
 * 내 판매 의약품 상세 페이지 맵핑
 * @param model
 * @param goodsCode
 * @return
 */
@GetMapping("/myMedicineDetailView")
public String myMedicineDetailView(Model model,
    @RequestParam(value="goodsCode") String goodsCode) {
    String suppName = suMedicineMapper.getSuppName(goodsCode);
    String mediCode = suMedicineMapper.getMediCode(goodsCode);
    Medicine medicine = suMedicineMapper.getMedicineInfo(goodsCode);
    SellMedicine sellMedicine = suMedicineMapper.getSellMedicineInfo(goodsCode);
    List<Ingredient> ingrList = suMedicineMapper.getIngrList(mediCode);
    List<Efficacy> effiList = suMedicineMapper.getEffiList(mediCode);
    MedicineInfo mediInfo = new MedicineInfo();
    mediInfo.setMedicine(medicine);
    mediInfo.setSellMedicine(sellMedicine);
    mediInfo.setIngrList(ingrList);
    mediInfo.setEffiList(effiList);
    log.info("mediInfo: {}", mediInfo);
    String effies = "";
    for(int i = 0; i < effiList.size(); i++){
        effies += effiList.get(i).getEfficacy();
        if(i != effiList.size() - 1){
            effies += ", ";
        }
    }

    model.addAttribute("effies", effies);
    model.addAttribute("suppName", suppName);
    model.addAttribute("mediInfo", mediInfo);
    model.addAttribute("title", "내 의약품 상세");
    model.addAttribute("content", "내 의약품 상세");

    return "user/supplier/medicine/myMedicineDetailView";
}

/**
 * 내 판매 의약품 등록 ajax 요청
 * 작성자: 길범진
 * @param model
 * @param medicineInfo
 * @param multipartFile
 * @return
 */
@PostMapping("/insertMedicine")
@ResponseBody
public String insertMedicine(Model model,

```

```

        @RequestParam(value="medicineInfo") MedicineInfo medicineInfo,
        @RequestParam(value="img") List<MultipartFile> multipartFile) {
    log.info("medicineInfo: {}", medicineInfo);
    log.info("multipartFile: {}", multipartFile);
    suMedicineService.insertMedicine(medicineInfo, multipartFile);

    return "redirect:/supplier/myMedicineSearchList";
}

/**
 * 내 판매 의약품 수정 ajax 요청
 * 작성자: 길범진
 * @param model
 * @param medicineInfo
 * @param multipartFile
 * @return
 */
@PostMapping("/modifyMedicine")
@ResponseBody
public String modifyMedicine(Model model,
        @RequestParam(value="medicineInfo") MedicineInfo medicineInfo,
        @RequestParam(value="img") List<MultipartFile> multipartFile){
    log.info("medicineInfo: {}", medicineInfo);
    log.info("multipartFile: {}", multipartFile);

    suMedicineService.modifyMedicine(medicineInfo, multipartFile);

    return "redirect:/supplier/myMedicineSearchList";
}

/**
 * 내 의약품 상태 -> '삭제'로 수정 ajax 요청
 * 작성자: 길범진
 * @param mediCode
 * @return
 */
@PostMapping("/myMedicineDelete")
public String deleteMedicine(@RequestParam(value="mediCode") String mediCode){
    suMedicineMapper.deleteSellMedicine(mediCode);
    return "redirect:/supplier/myMedicineSearchList";
}
}

```

■ Service

```

package team01.yaksutor.supplier.service;

import jakarta.servlet.http.HttpServletRequest;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.web.multipart.MultipartFile;

```

```

import team01.yaksutor.admin.dto.RequestMedicine;
import team01.yaksutor.dto.*;
import team01.yaksutor.file.dto.FileRequest;
import team01.yaksutor.file.util.FileUtils;
import team01.yaksutor.supplier.mapper.SuMedicineMapper;

import java.util.List;

@Service
@Transactional
@RequiredArgsConstructor
@Slf4j
public class SuMedicineService {
    private final SuMedicineMapper suMedicineMapper;
    private final FileUtils fileUtils;
    private final HttpServletRequest request;

    /**
     * 내 판매 의약품 목록 조회
     * 작성자: 길범진
     * @param suppCode
     * @return
     */
    public List<SellMedicine> getSellMediList(String suppCode){
        List<SellMedicine> sellMediList = suMedicineMapper.getListSellMedicine(suppCode);

        return sellMediList;
    }

    /**
     * 내 판매 의약품 등록 트랜잭션
     * 작성자: 길범진
     * @param medicineInfo
     * @param multipartFile
     */
    @Transactional
    public void insertMedicine(MedicineInfo medicineInfo,
                               List<MultipartFile> multipartFile) {
        String sid = request.getSession().getAttribute("S_ID").toString();
        log.info("sid: {}", sid);
        Medicine medicine = medicineInfo.getMedicine();
        SellMedicine sellMedicine = medicineInfo.getSellMedicine();
        List<Ingredient> ingrList = medicineInfo.getIngrList();
    }
}

```

```

List<Efficacy> effiList = medicineInfo.getEffiList();
List<FileRequest> fileList = fileUtils.uploadFiles(multipartFile);
fileList.forEach( file -> {
    log.info("file:{}", file);
    medicine.setMediImg(file.getFileIdx());
    sellMedicine.setMediImg(file.getFileIdx());
});
for(FileRequest fileRequest : fileList){
    suMedicineMapper.insertImg(fileRequest);
}

medicine.setRegMId(sid);
suMedicineMapper.insertMedicine(medicine);
String mediCode = medicine.getMediCode();
sellMedicine.setSuppCode(suMedicineMapper.getSuppCode(sid));
sellMedicine.setMediName(medicine.getMediName());
sellMedicine.setMediCode(medicine.getMediCode());
suMedicineMapper.insertSellMedicine(sellMedicine);
ingrList.forEach( ingr -> {
    ingr.setMediCode(mediCode);
    ingr.setRegMId(sid);
    suMedicineMapper.insertIngredient(ingr);
});
effiList.forEach( effi -> {
    effi.setMediCode(mediCode);
    effi.setRegMId(sid);
    suMedicineMapper.insertEfficacy(effi);
});
}

/**
 * 내 판매 의약품 수정 트랜잭션
 * 작성자: 길범진
 * @param medicineInfo
 * @param multipartFile
 */
@Transactional
public void modifyMedicine(MedicineInfo medicineInfo,
    List<MultipartFile> multipartFile){
    Medicine medicine = medicineInfo.getMedicine();
    SellMedicine sellMedicine = medicineInfo.getSellMedicine();
    List<Ingredient> ingrList = medicineInfo.getIngrList();
    List<Efficacy> effiList = medicineInfo.getEffiList();

```

```

String mediCode = medicine.getMediCode();
String sid = request.getSession().getAttribute("S_ID").toString();
String suppCode = suMedicineMapper.getSuppCode(sid);

List<FileRequest> fileList = fileUtils.uploadFiles(multipartFile);
fileList.forEach( file -> {
    log.info("file:{}", file);
    medicine.setMediImg(file.getFileIdx());
    sellMedicine.setMediImg(file.getFileIdx());
});
for(FileRequest fileRequest : fileList){
    suMedicineMapper.insertImg(fileRequest);
}

medicine.setRegMId(sid);
suMedicineMapper.updateMedicine(medicine);

sellMedicine.setSuppCode(suppCode);
sellMedicine.setMediName(medicine.getMediName());

sellMedicine.setMediCode(medicine.getMediCode());
suMedicineMapper.updateSellMedicine(sellMedicine);

suMedicineMapper.deleteIngredient(mediCode);
suMedicineMapper.deleteEfficacy(mediCode);

ingrList.forEach( ingr -> {
    ingr.setMediCode(medicine.getMediCode());
    ingr.setRegMId(sid);
    suMedicineMapper.insertIngredient(ingr);
});

effiList.forEach( effi -> {
    effi.setMediCode(medicine.getMediCode());
    effi.setRegMId(sid);
    suMedicineMapper.insertEfficacy(effi);
});
}
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="team01.yaksutor.supplier.mapper.SuMedicineMapper">
  <resultMap id="medicineListMap" type="Medicine">
    <id column="medi_code"    property="mediCode" />
    <result column="reg_m_id"  property="regMId" />
    <result column="medi_name" property="mediName" />
    <result column="medi_img"  property="mediImg" />
    <result column="medi_drugform" property="mediDrugform" />
    <result column="medi_drugtype" property="mediDrugtype" />
    <result column="medi_state" property="mediState" />
    <result column="medi_usebydate" property="mediUsebydate" />
    <result column="reg_date"   property="regDate" />

    <association property="imgData" javaType="ImgData">
      <id column="file_idx"    property="fileIdx"/>
      <result column="file_origin_name" property="fileOriginName"/>
      <result column="file_new_name"    property="fileNewName"/>
      <result column="file_path"       property="filePath"/>
      <result column="file_size"      property="fileSize"/>
      <result column="file_reg_date"   property="fileRegDate"/>
    </association>
  </resultMap>

  <resultMap id="sellMedicineListMap" type="SellMedicine">
    <id column="goods_code"    property="goodsCode" />
    <result column="supp_code"  property="suppCode" />
    <result column="medi_code"  property="mediCode" />
    <result column="medi_name"  property="mediName" />
    <result column="medi_detail" property="mediDetail" />
  </resultMap>
</mapper>

```

```

<result column="medi_price" property="mediPrice" />
<result column="medi_img" property="medilmg" />
<result column="reg_qty" property="regQty" />
<result column="state" property="state" />
<result column="reg_date" property="regDate" />

<association property="imgData" javaType="ImgData">
  <id column="file_idx" property="fileIdx"/>
  <result column="file_origin_name" property="fileOriginName"/>
  <result column="file_new_name" property="fileNewName"/>
  <result column="file_path" property="filePath"/>
  <result column="file_size" property="fileSize"/>
  <result column="file_reg_date" property="fileRegDate"/>
</association>
</resultMap>

<resultMap id="medicalInfoMap" type="MedicalInfo">
  <id column="medi_code" property="mediCode" />
  <result column="reg_m_id" property="regMId" />
  <result column="medi_name" property="mediName" />
  <result column="medi_img" property="medilmg" />
  <result column="medi_drugform" property="mediDrugform" />
  <result column="medi_drugtype" property="mediDrugtype" />
  <result column="medi_state" property="mediState" />
  <result column="medi_usebydate" property="mediUsebydate" />
  <result column="reg_date" property="regDate" />

  <association property="imgInfo" javaType="AdlmgData">
    <id column="file_idx" property="fileIdx"/>
    <result column="file_origin_name" property="fileOriginName"/>
    <result column="file_new_name" property="fileNewName"/>
    <result column="file_path" property="filePath"/>
  </association>
</resultMap>

```



```

    <result column="file_size"      property="fileSize"/>
    <result column="file_reg_date"  property="fileRegDate"/>
</association>
</resultMap>

<resultMap id="ingrListMap" type="Ingredient">
    <id      column="ingredient_code"      property="ingredientCode"/>
    <result column="medi_code"      property="mediCode"/>
    <result column="reg_m_id"      property="regMId"/>
    <result column="ingredient"      property="ingredient"/>
    <result column="content"      property="content"/>
    <result column="reg_date"      property="regDate"/>
</resultMap>

<resultMap id="effiListMap" type="Efficacy">
    <id      column="efficacy_code"      property="efficacyCode"/>
    <result column="medi_code"      property="mediCode"/>
    <result column="reg_m_id"      property="regMId"/>
    <result column="efficacy"      property="efficacy"/>
    <result column="reg_date"      property="regDate"/>
</resultMap>

<select id="getListSellMedicine" parameterType="string" resultMap="sellMedicineListMap">
    SELECT
        s.*,
        i.file_path
    FROM
        sell_medicine AS s
        INNER join
        img_data i
    on
        s.medi_img = i.file_idx

```

```

WHERE
    s.supp_code = #{suppCode} AND s.state != '삭제';
</select>

<select id="getSuppCode" parameterType="string" resultType="string">
    SELECT
        supp_code
    FROM
        supplier AS s
    WHERE
        s.supp_id = #{sid};
</select>

<insert id="insertMedicine" parameterType="Medicine">
    <selectKey keyProperty="mediCode" resultType="String" order="BEFORE">
        /* 상품 자동 증가 코드 */
        SELECT
            case
            when COUNT(medi_code) = 0 then 'medi_code_1'
            ELSE
                CONCAT('medi_code_',
                    max(cast(SUBSTRING_INDEX(medi_code, 'medi_code_', -1) AS UNSIGNED))+1
                )
            END AS mediCode
        FROM
            medicine;
    </selectKey>
    INSERT INTO medicine
        (medi_code, reg_m_id, medi_name, medi_img, medi_drugform, medi_drugtype, medi_state, medi_usebydate,
reg_date)
        VALUES (#{mediCode}, #{regMId}, #{mediName}, #{mediImg}, #{mediDrugform}, #{mediDrugtype},
        #{mediState}, #{mediUsebydate}, CURDATE())

```

```
</insert>
```

```
<insert id="insertImg" parameterType="FileRequest">
```

```
INSERT INTO img_data
```

```
(file_idx, file_origin_name, file_new_name, file_path, file_size, file_reg_date)
```

```
VALUES ({fileIdx}, {fileOriginName}, {fileNewName}, {filePath}, {fileSize}, {fileRegDate})
```

```
</insert>
```

```
<insert id="insertIngredient" parameterType="Ingredient">
```

```
<selectKey keyProperty="ingredientCode" resultType="String" order="BEFORE">
```

```
/* 상품 자동 증가 코드 */
```

```
SELECT
```

```
case
```

```
when COUNT(ingredient_code) = 0 then 'ingr_1'
```

```
ELSE
```

```
CONCAT('ingr_',
```

```
max(cast(SUBSTRING_INDEX(ingredient_code, 'ingr_', -1) AS UNSIGNED))+1
```

```
)
```

```
END AS ingredientCode
```

```
FROM
```

```
ingredient;
```

```
</selectKey>
```

```
INSERT INTO ingredient
```

```
(Ingredient_code, medi_code, reg_m_id, Ingredient, Content, reg_date)
```

```
VALUES ({ingredientCode}, {mediCode}, {regMId}, {ingredient}, {content}, CURDATE())
```

```
</insert>
```

```
<insert id="insertEfficacy" parameterType="Efficacy">
```

```
<selectKey keyProperty="efficacyCode" resultType="String" order="BEFORE">
```

```
/* 상품 자동 증가 코드 */
```

```
SELECT
```

```
case
```

```

when COUNT(efficacy_code) = 0      then 'effi_1'

ELSE

CONCAT('effi_',

max(cast(SUBSTRING_INDEX(efficacy_code, 'effi_', -1) AS UNSIGNED))+1

)

END AS efficacyCode

FROM

efficacy;

</selectKey>

INSERT INTO efficacy

(efficacy_code, medi_code, reg_m_id, efficacy, reg_date)

VALUES ({efficacyCode}, #{mediCode}, #{regMId}, #{efficacy}, CURDATE())

</insert>

<insert id="insertSellMedicine" parameterType="SellMedicine">

<selectKey keyProperty="goodsCode" resultType="String" order="BEFORE">

/* 상품 자동 증가 코드 */

SELECT

case

when COUNT(goods_code) = 0      then 'goods_code_1'

ELSE

CONCAT('goods_code_',

max(cast(SUBSTRING_INDEX(goods_code, 'goods_code_', -1) AS UNSIGNED))+1

)

END AS goodsCode

FROM

sell_medicine;

</selectKey>

INSERT INTO sell_medicine

(goods_code, supp_code, medi_code, medi_name, medi_detail, medi_price, medi_img, reg_qty, state,

reg_date)

VALUES ({goodsCode}, {suppCode}, #{mediCode}, #{mediName}, #{mediDetail}, #{mediPrice}, #{mediImg},

```

```
#{regQty}, '판매중', CURDATE())
```

```
</insert>
```

```
<select id="getMedicineInfo" parameterType="String" resultMap="medicineListMap">
```

```
SELECT
```

```
    m.*,
```

```
    d.*
```

```
FROM
```

```
    medicine AS m
```

```
INNER join
```

```
sell_medicine AS s
```

```
on
```

```
m.medi_code = s.medi_code
```

```
INNER join
```

```
img_data AS d
```

```
on
```

```
m.medi_img = d.file_idx
```

```
WHERE
```

```
s.goods_code = #{goodsCode};
```

```
</select>
```

```
<select id="getSuppName" parameterType="string" resultType="string">
```

```
SELECT
```

```
su.sup_name
```

```
FROM
```

```
sell_medicine AS s
```

```
INNER join
```

```
supplier AS su
```

```
on
```

```
s.supp_code = su.supp_code
```

```
WHERE
```

```
s.goods_code = #{goodsCode};
```

```
</select>
```

```
<select id="getSellMedicineInfo" parameterType="String" resultMap="sellMedicineListMap">
```

```
    SELECT
```

```
    s.*
```

```
    FROM
```

```
    sell_medicine AS s
```

```
    WHERE
```

```
    s.goods_code = #{goodsCode};
```

```
</select>
```

```
<select id="getIngrList" parameterType="String" resultMap="ingrListMap">
```

```
    SELECT
```

```
    i.*
```

```
    FROM
```

```
    ingredient AS i
```

```
    WHERE
```

```
    i.medi_code = #{mediCode};
```

```
</select>
```

```
<select id="getEffiList" parameterType="String" resultMap="effiListMap">
```

```
    SELECT
```

```
    e.*
```

```
    FROM
```

```
    efficacy AS e
```

```
    WHERE
```

```
    e.medi_code = #{mediCode};
```

```
</select>
```

```
<select id="getMediCode" parameterType="string" resultType="string">
```

```
    SELECT
```

```
    m.medi_code
```

```

FROM
medicine AS m
INNER join
sell_medicine AS s
on
m.medi_code = s.medi_code
WHERE
goods_code = #{goodsCode};
</select>

<update id="updateMedicine" parameterType="Medicine">
UPDATE medicine
SET
    reg_m_id=#{regMId},
    medi_name=#{mediName},
    medi_img=#{mediImg},
    medi_drugform=#{mediDrugform},
    medi_drugtype=#{mediDrugtype},
    medi_state=#{mediState},
    medi_usebydate=#{mediUsebydate}
WHERE medi_code=#{mediCode}
</update>

<delete id="deleteIngredient" parameterType="string">
DELETE FROM ingredient WHERE medi_code=#{mediCode}
</delete>

<delete id="deleteEfficacy" parameterType="string">
DELETE FROM efficacy WHERE medi_code=#{mediCode}
</delete>

<update id="updateSellMedicine" parameterType="SellMedicine">

```

```
UPDATE sell_medicine

SET

    supp_code=#{suppCode},

    medi_code=#{mediCode},

    medi_name=#{mediName},

    medi_detail=#{mediDetail},

    medi_price=#{mediPrice},

    medi_img=#{mediImg},

    reg_qty=#{regQty}

WHERE goods_code=#{goodsCode}

</update>

<update id="deleteSellMedicine" parameterType="string">

    UPDATE sell_medicine

    SET

        state='삭제'

    WHERE

        medi_code=#{mediCode};

</update>

</mapper>
```