



What's in My cart?

목차



01

프로젝트 소개

Project
Team Member

02

프로그래밍 과정

개발 환경

03

Web 기획

Storyboard/ER- Diagram/Sequence Diagram/
Use Case/Class Diagram

04

홈페이지 구현

구현 영상
주요 기능



01. 프로젝트 소개

Project



What's in My Cart?

“ 코로나 19로 직접 접촉이 꺼려지는 환경에서 실시간 배송 서비스들이 많이 생기는 시기에 'E-mart, 홈플러스 등의 대형 오프라인 마트에서 어플리케이션을 이용하여 실시간 배송 서비스를 제공하면 어떨까?' 라는 생각으로 제작하게 되었습니다. ”



Client

위치기반으로 주변 마트의 물품을 구매하는 고객으로서 Shopper의 Delivery서비스를 제공받는 대상입니다.



Shopper

Client에게 배달서비스를 제공해주는 대상입니다. Client가 지불한 Delivery Cost를 받고 배달 서비스를 대행해줍니다.



Admin

Client에게 QnA를 통해 웹사이트 관련 문제를 해결하고, DashBoard를 통해 다양한 통계치를 활용합니다.



02. 프로그래밍 과정



개발 환경

BACK-END



BACK-END 개발환경 선택 이유

- Spring boot 는 내장 서버와 DB를 지원하기 때문에 개발 환경을 더 쉽게 설정할 수 있어서 선택했습니다.
- JPA는 Mapper를 이용할때보다 객체지향적인 개발을 할 수 있기 때문에 선택했습니다.





개발 환경

FRONT-END



FRONT-END 개발환경 선택 이유

- 쉽게 웹 디자인을 할 수 있어 사용하게 되었습니다.
- Spring boot에서 JSP보다 Thymeleaf 템플릿을 권장하여서 사용하였습니다.

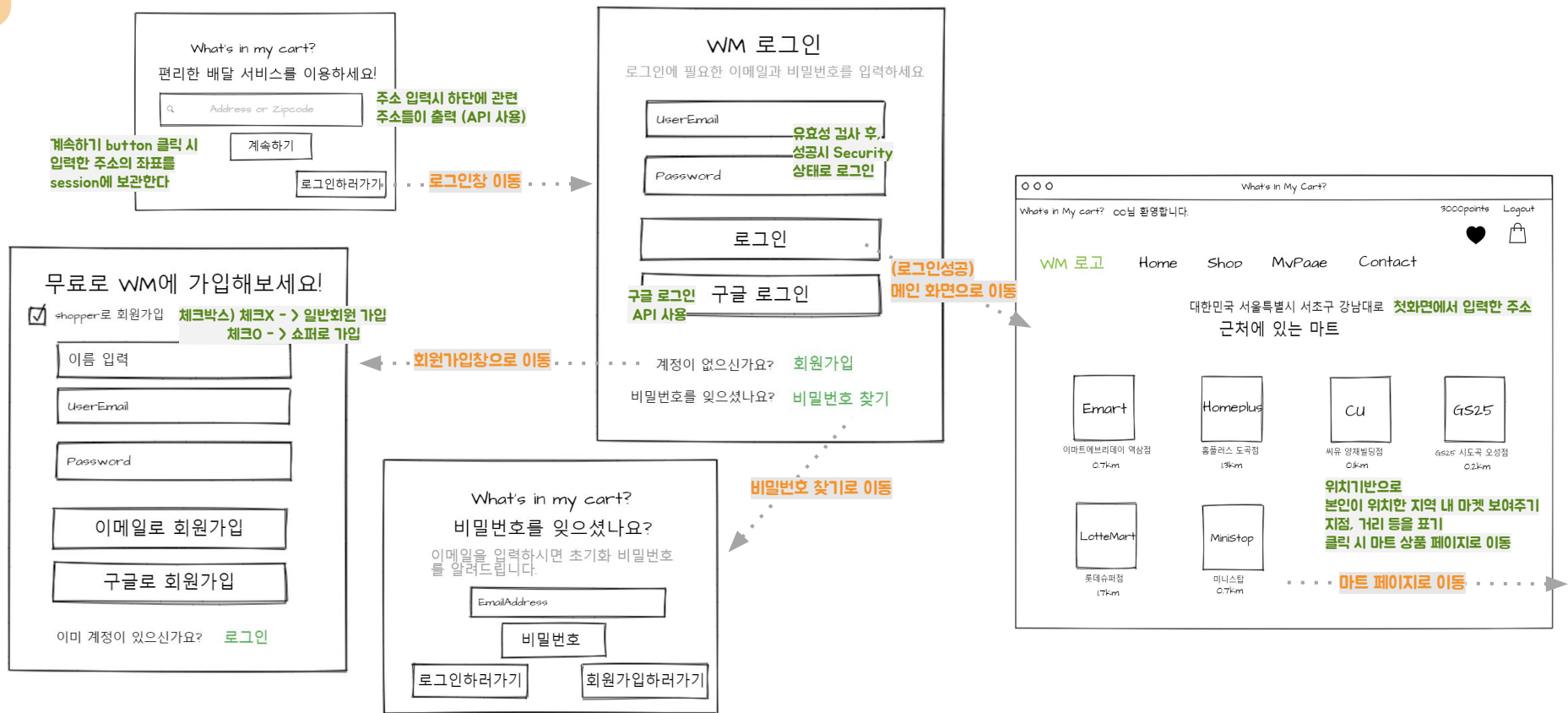




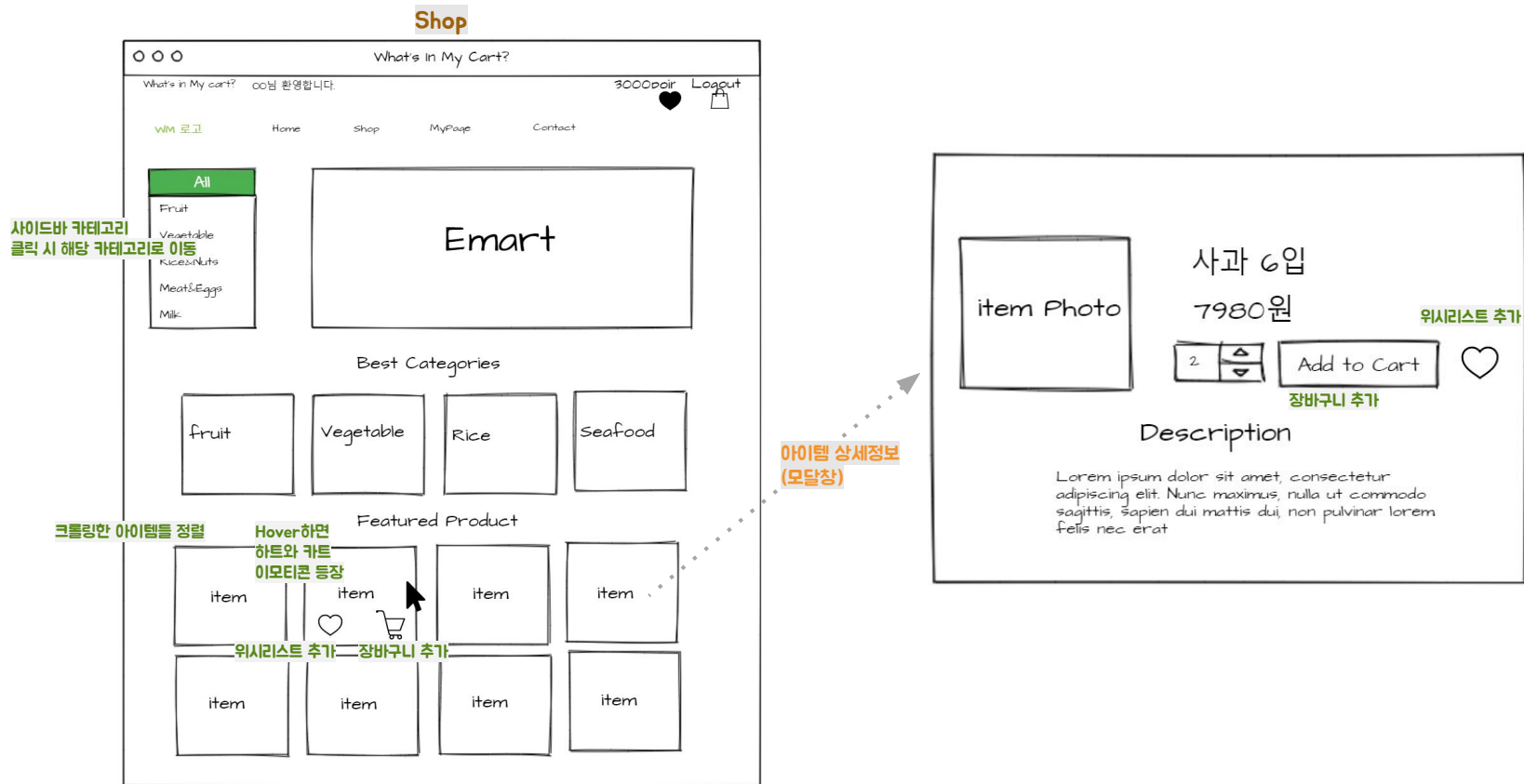
03. Web 기획

Storyboard - Client

... ➡ click

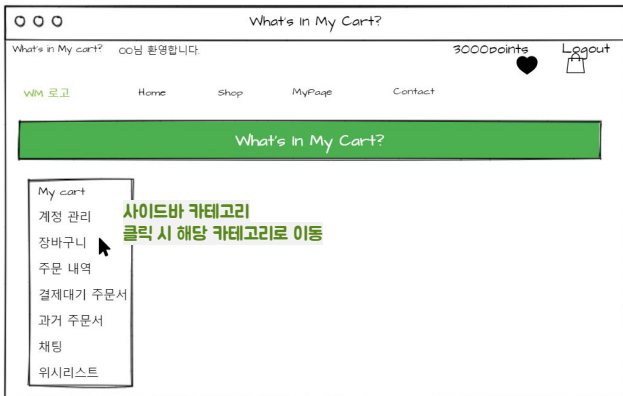


Storyboard - Client



Storyboard - Client

My Page



마트페이지로 이동

계정관리

My Info

이름

이메일 이메일은 readonly

주소

비밀번호 입력 후 수정 및 탈퇴 기능

주문서 (주문내역)

주문서가 null일때 표시되는 화면

주문하신 상품이 없으시네요!

(if문 조건)
주문서 매칭X -> 매칭되지 않았습니니다!
주문서 매칭O -> ~님과 매칭되었습니다!

장바구니

장바구니 상품 count 표기

총 1개의 상품

emart 바나멜리 프레스쉬 바나나 1송이 장바구니에서 빼기 2980원

장바구니 상품 정보 정렬

The total amount of
물품 가격 2980원
배송비 5000
총 결제금액 7980원

장바구니로 이동

결제대기로 이동

Order List (결제대기/과거)

1번 주문서 주문서가 존재할때, 주문서를 표시

Total 7980원

My Order List

1번 주문서
아직 shopper와 매칭되지 않았습니니다!

2번 주문서
shopper5@naver.com님과 매칭 되셨어요!

배송이 완료되면 거래종료 클릭
클릭 시 Shopper 평가

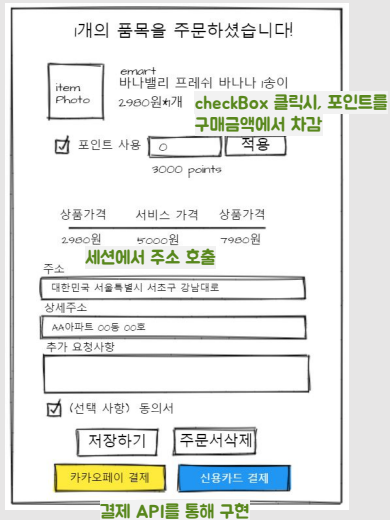
채팅창으로 이동

Storyboard - Client

My Page



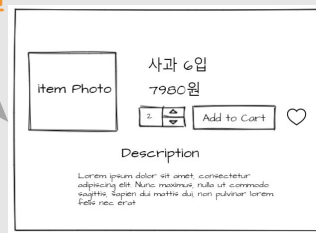
주문서 (결제대기)



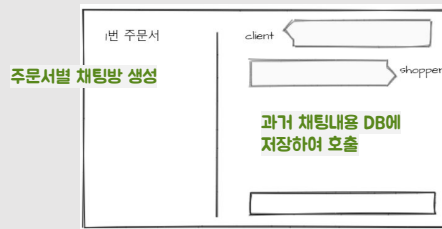
위시리스트



아이템 상세정보 (모달창)



채팅



Storyboard - Client

QnA

What's In My Cart?

What's in My cart? 00님 환영합니다. 3000points Logout

WM 로고 Home Shop MyPage Contact

QnA

문의하기

나의 QnA

QnA / 나의 QnA

QnA는 전체 질문글,
나의 QnA는 내가 한 질문글이 나옴

No	title	name	date
	문의합니다	김회원	20/08/12

Dropbox 제목, 이를 선택가능

제목 Q search 문의하기

제목, 이를 기준으로 QnA 검색 가능

문의하기로 이동

문의하기

Leave Message

name email 세션에서 이름, 이메일 호출 Read only

제목

문의 사항을 입력하세요

문의하기

Storyboard - Shopper

Shopper Page

What's In My Cart? Shopper

내 계정
메세지
주문 관리
길 찾기
로그아웃

Welcome!

해당 홈페이지는 shopper 전용 홈페이지입니다.
What's in My cart의 딜리버리 서비스를 지원해보
세요

Drop down 형식

내 계정
마이페이지
회원탈퇴
메세지
채팅
주문 관리
올라온 주문서
받은 주문서
처리한 주문서
길 찾기
로그아웃

마이페이지

Client page와 동일

My info

이름

이메일 이메일은 readonly

주소

회원탈퇴

비밀번호를 입력해 주세요.

비밀번호 입력 후 수정 및 탈퇴 가능

채팅

1번 주문서

client

shopper

매칭된 주문서별
채팅방 생성

과거 채팅내용 DB에
저장하여 호출

올라온 주문서

Order Sheets Client가 주문한 내역 전체 확인

1번 주문서

☒ Check를 누르면 매칭 완료

받은 주문서

Order Sheets 매칭 완료된 주문서 내역

1번 주문서

by 000@gmail.com

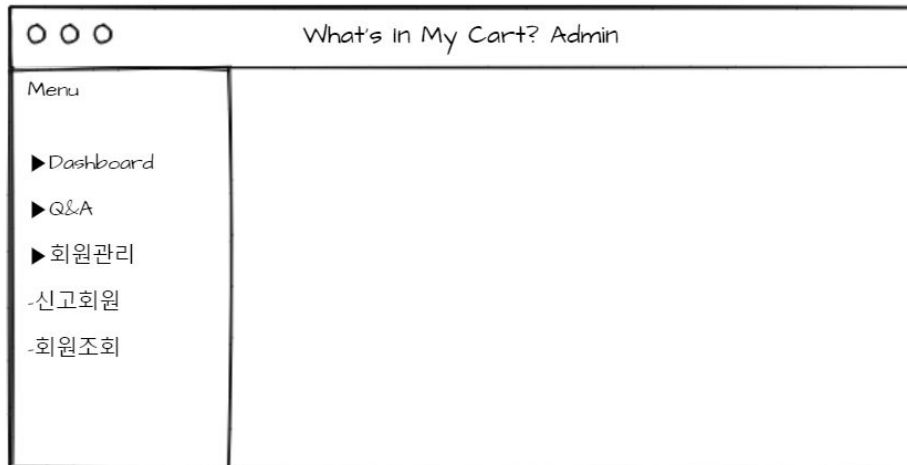
처리한 주문서

Order Sheets 배송 완료된 주문서 내역

1번 주문서

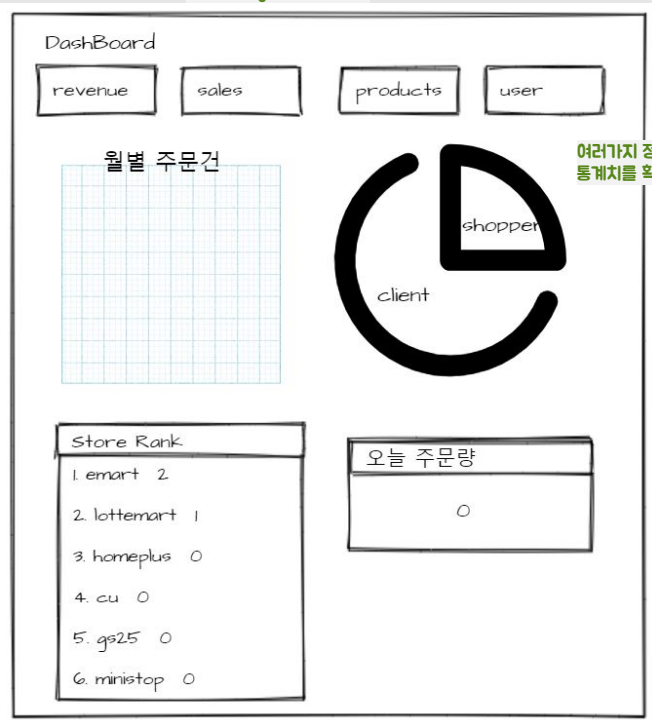
Storyboard - Admin

Admin Page



Dashboard

Admin Page의 main 화면



Storyboard - Admin

Admin Page

○○○

What's In My Cart? Admin

Menu

▶ Dashboard
▶ Q&A
▶ 회원관리
- 신고회원
- 회원조회

QnA

Q&A 관리 상세글을 클릭하면 답변 가능

No	title	name	date
1	문의합니다	김회원	20/08/12

제목

Q

search

Dropbox
제목, 이름 선택가능

제목, 이름 기준으로 QnA 검색 가능

신고회원

회원 신고 내역 이메일 또는 아이디로 검색 가능

Q

search

이메일

번호	신고자	신고당한 회원	내용	날짜	처리버튼
----	-----	---------	----	----	------

회원조회

회원조회

Q

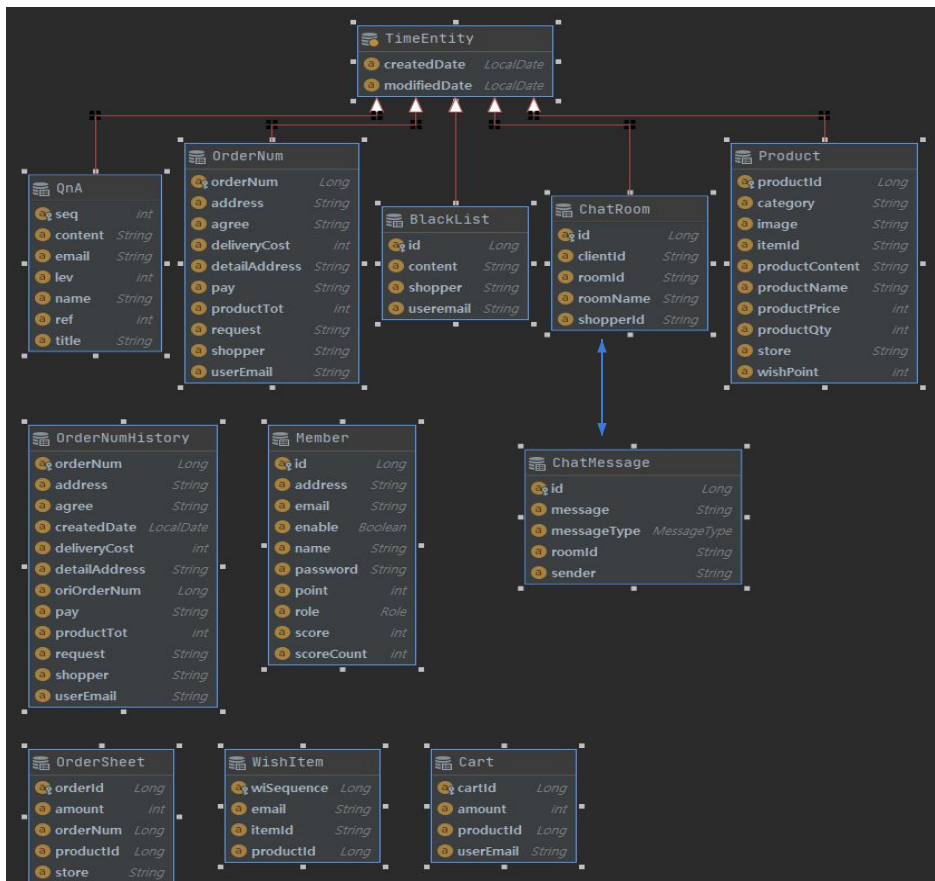
search

이메일

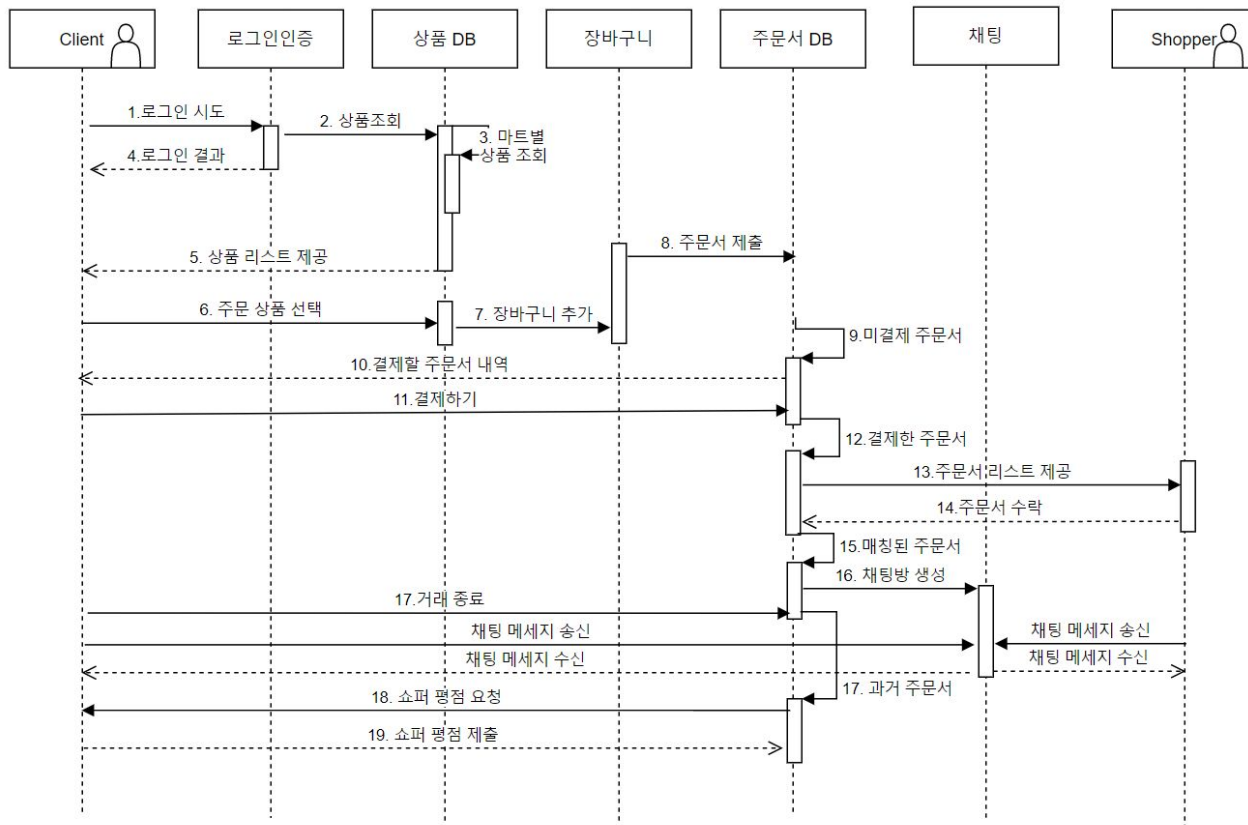
Email	Name	Address	Role	Enable	Setting
-------	------	---------	------	--------	---------

회원 정지 유무 확인
회원정지 취소 버튼

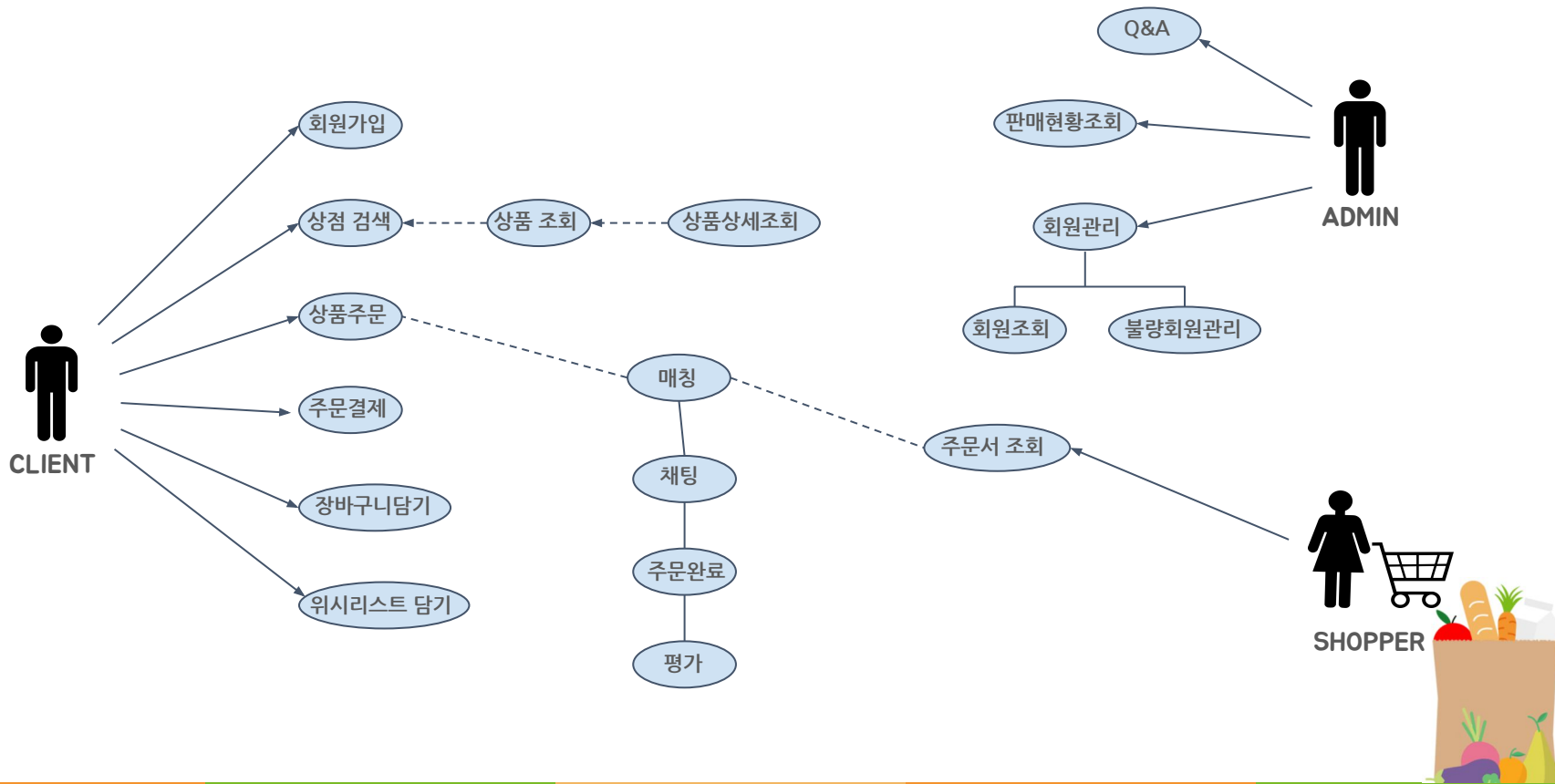
ER-Diagram



Sequence Diagram



Use Case





04. 홈페이지 구현



주요 기능

- ✓ Spring Security로 로그인, 회원가입 및 불량회원 관리
- ✓ SMTP를 이용한 메일 발송
- ✓ 사용자(Client) 위치 기반 마트 정보
- ✓ 사용자(Shopper) 위치 기반 주문서 리스트
- ✓ QnA 게시판 구현
- ✓ 마트별 상품에 대한 crawling
- ✓ 위시리스트, 장바구니
- ✓ 카카오 페이, 신용카드 결제 API
- ✓ STOMP를 이용한 1 : 1 채팅
- ✓ Chart.js를 이용한 Admin Dashboard





Spring Security

- ✓ Spring Security로 로그인, 회원가입 및 불량회원 관리
- ✓ SMTP를 이용한 메일 발송

웹사이트 접속 시 우편번호를 먼저 입력받고 로그인 및 회원가입을 진행한다.
입력받은 우편번호는 위치기반 서비스 제공 시 사용된다.

- 이름, 이메일, 비밀번호 유효성 검사
- 비밀번호 암호화(BcryptPasswordEncoder)
- 주소입력창 자동완성(Google 주소 API 사용)
- 비밀번호 찾기 JavaMailSender로 email 전송
- 소셜 로그인으로 로그인 시 기존회원인지 확인 후 update 또는 save





위치 기반

- ✓ 사용자(Client) 위치 기반 마트 정보
- ✓ 사용자(Shopper) 위치 기반 주문서 리스트

Client가 처음 입력한 주소정보(우편번호)를 기반으로 주변 마트 정보를 나타낸다.

Shopper의 현재 위치로부터 Client가 선택한 마켓간의 위치를 계산하여 주문 정보를 나타낸다.

- 특정 위치로부터 마트까지 떨어져있는 거리 계산
- 정규식표현으로 정확하지 않은 마트들 제외
- Client가 여러개의 마켓에서 상품을 골랐을 경우, 마켓들 중 쇼퍼에게 가까운 마켓과의 거리를 보여줌
- 렌더링 속도에 대한 고민





QnA 게시판

✓ QnA 게시판 구현

QnA 게시판을 통해 Client와 Admin이 소통할 수 있다.
불만사항이나 문의글을 올릴 수 있으며, 답변받을 수 있다.

- 게시판 구현
- 게시글 검색 가능
- 내가 쓴 게시글 수정 및 삭제 가능
- Client가 모든 문의글을 볼 수 있으며, 내가 쓴 게시글도 따로 볼 수 있음





Crawling

✓ 마트별 상품에 대한 **crawling**

마트 / 카테고리별로 상품을 crawling하여 상품정보를 조회할 수 있도록 한다.

- Java에서 제공하는 Jsoup 사용
- 마트별로 상품정보를 가져옴
- 상품 클릭 시 상품 상세보기 가능
- 카테고리별로 상품정보를 가져옴
- 마트별, 카테고리별 상품검색 가능
- 위시리스트, 장바구니에 추가 가능
- crawling 데이터 및 서버관리에 대한 고민
- crawling 속도에 대한 고민





위시리스트, 장바구니, 결제

✓ 위시리스트, 장바구니

✓ 카카오 페이, 신용카드 결제 API

구매하고자 하는 상품을 장바구니 / 위시리스트에 담아 관리할 수 있다.

My Cart Page나 장바구니 / 위시리스트 아이콘을 클릭하면 상품목록을 확인할 수 있다.

장바구니에 있는 상품들은 카카오페이, 신용카드로 결제 가능하다.

결제가 완료된 주문서는 Shopper가 확인할 수 있어, Shopper 매칭 후 서비스를 제공받을 수 있다.

- 물건 장바구니에 담기 및 찜하기 가능
- 장바구니의 상품 정보(구매갯수, 배송비 등) 수정 가능
- API를 이용한 결제 기능 구현
- 주문서의 정보를 Shopper에게 제공
- Client가 원하는 상품이 품절되었을 시 대체 상품에 대한 고민을 마일리지 차감방식으로 해결





채팅

✓ STOMP를 이용한 1 : 1 채팅

채팅 기능을 통해 Client와 Shopper가 소통할 수 있다.

- STOMP를 이용한 채팅 구현
- Client와 Shopper간 커뮤니케이션 가능
- Shopper가 서비스를 제공할 주문서를 클릭했을 때 채팅방 생성
- Client가 배송완료 버튼을 누르면 채팅방 삭제





Dashboard

✓ Chart.js를 이용한 Admin **Dashboard**

현재 판매현황과 회원수를 보여주는 Admin Dashboard를 구현하였다.

- Chart.js를 이용해 현재 판매현황과 회원정보를 보여줌
- 월별 주문량 관리
- 어떤 지표들을 중심으로 보여주는게 적합한지 고민



구현 영상



